CE7453: Photogrammetric Computer Vision

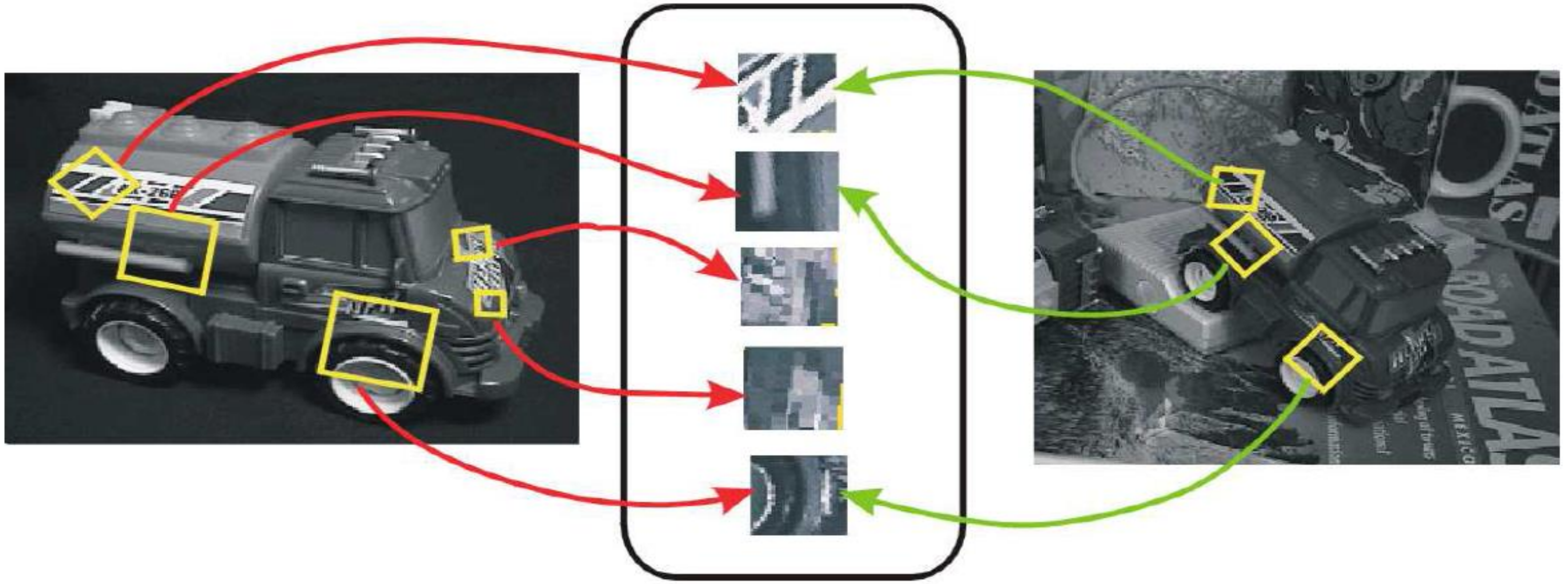
Lecture 4

Features

Acknowledgements: Most of the slides in this lecture come from Ping Tan, Yung-Yu Chuang. part of the materials of the all the lecture notes are from Cyrill Stachniss, Marc Pollefey, Wolfgang Foerstner, Bernhard Wrobel, James Hays, A. Dermanis, Armin Gruen, Alper Yilmaz.

Invariant Local Features

Features points/lines that are invariant to translation, rotation, scale and other projective distortions, are keys to push forward intelligent visual applications.

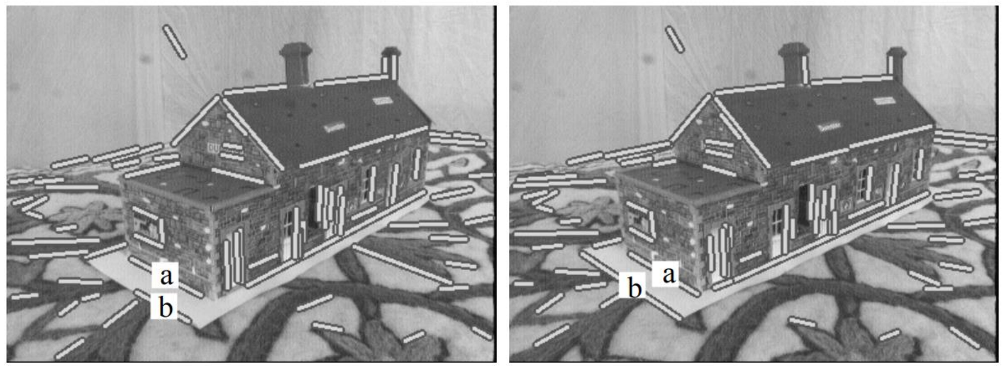


Features Descriptors

Features

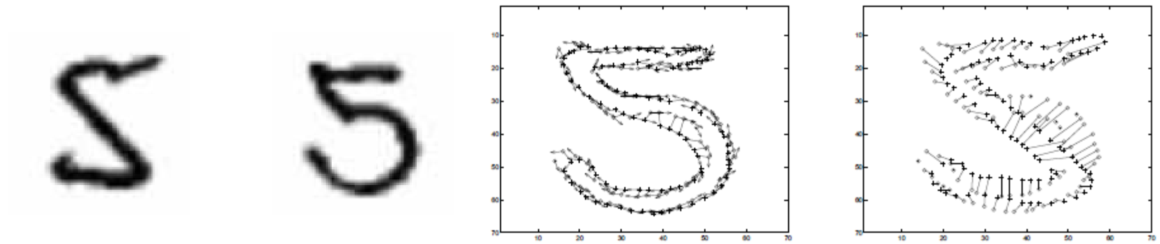
- It does not have to be only points!

- Line features



(Schmid 97 ICCV)

- Curves



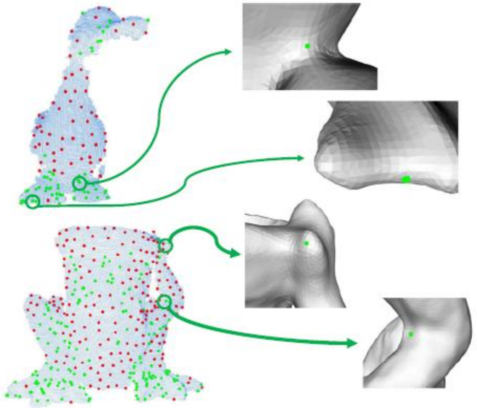
Belongie et al, 2002, PAMI

- Areas

Features – Cont.

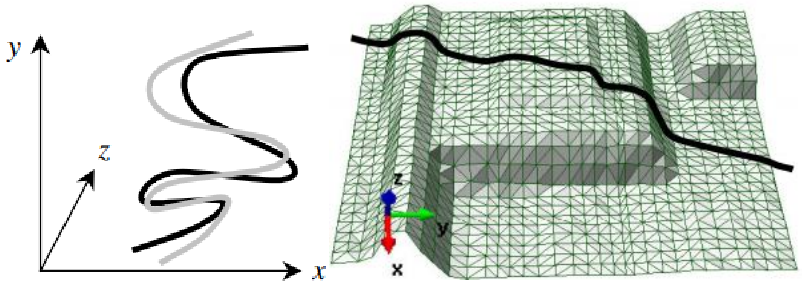
- It does not have to be only in 2D!

- 3D key points



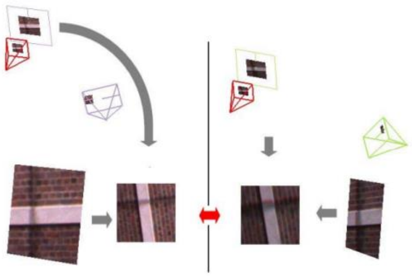
(Tonioni, et al, 2017, IJCV)

- 3D lines and Curves



Gruen and Acka, 2005, ISPRS Journal

- 3D patches



(Wu, et al, 2008, CVPR)

Features are keys to many fundamental problems

- Image alignment (3D, 2D registration)
- Image geo-referencing
- 3D reconstruction
- Motion tracking
- Object identification
- Image-based indexing and retrieval
- Navigation

Point Detectors

Moravec Corner Detector

Harris Corner Detector

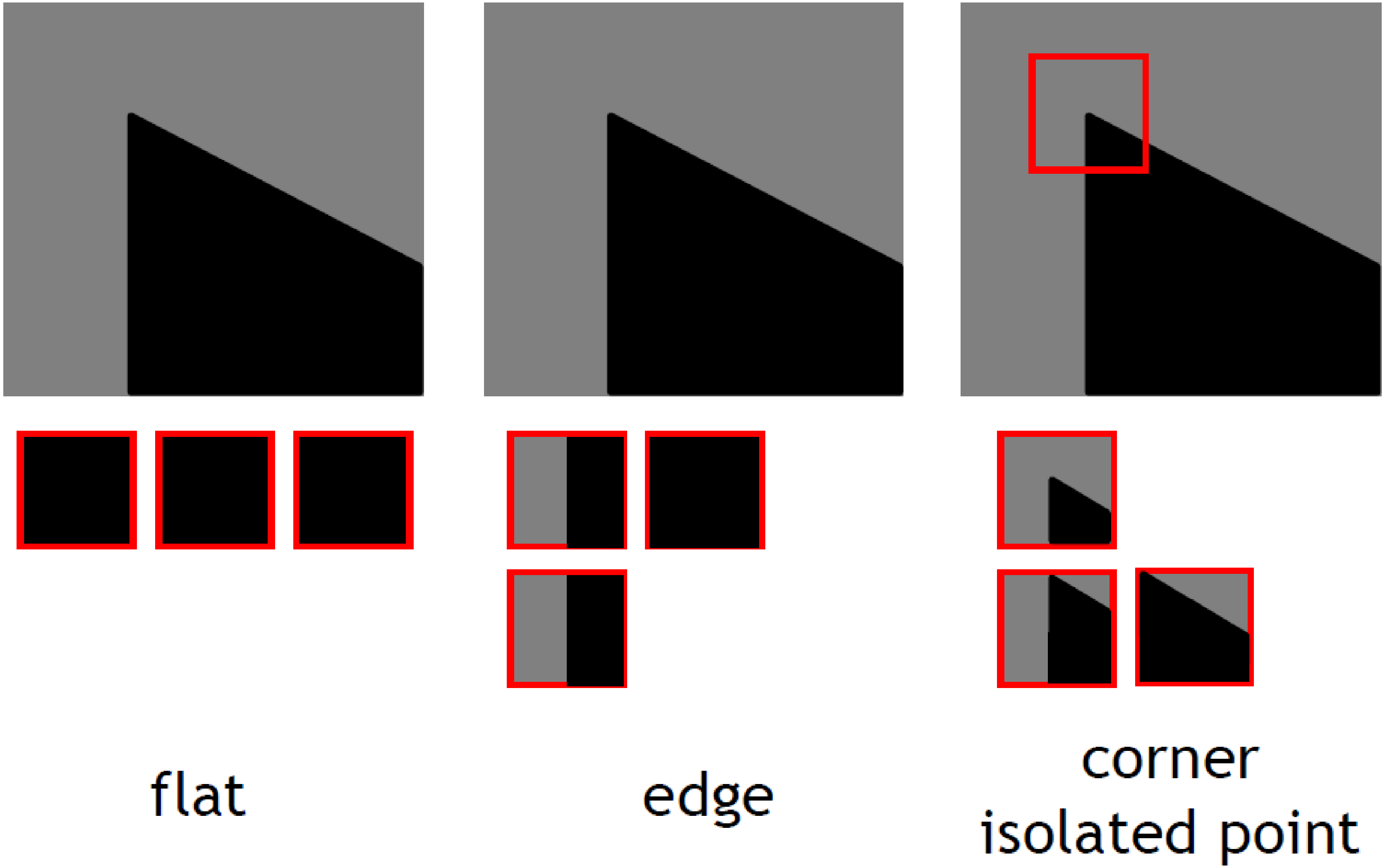
Sift Corner Detector

PCA-Sift

Affine-Sift

Learning-based detector

Moravec Corner Detector (1980)



Moravec Corner Detector – Cont.

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

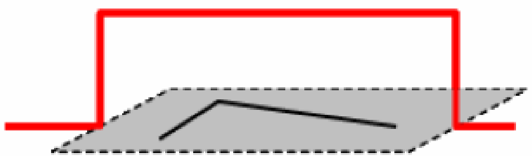
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

Four shifts: (u,v) = (1,0), (1,1), (0,1), (-1, 1)

Look for local maxima in *min{E}*

Moravec Corner Detector - Drawbacks

- Noisy response due to a binary window function
- Only a set of shifts at every 45 degree is considered
- Only Maximum of E is taken into account
- Computationally intensive

Harris Corner Detector

Window-averaged change of intensity induced by shifting the image data by [*u*,*v*]:

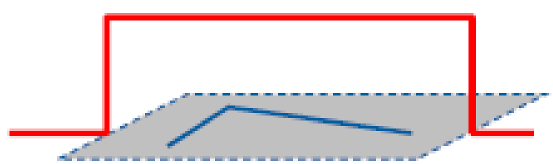
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Window function

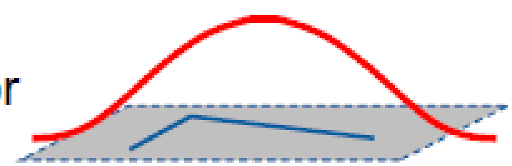
Shifted intensity

Intensity

Window function $w(x,y)$ =



or



1 in window, 0 outside

Gaussian

Harris Corner Detector – Cont.

First order Taylor Expansion

$$E(u,v) \approx \sum_{x,y} w(x,y)[I(x,y) + uI_x + vI_y - I(x,y)]^2$$

$$= \sum_{x,y} w(x,y)[uI_x + vI_y]^2$$

$$= (u \quad v) \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

Harris Corner Detector – Cont.

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

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

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

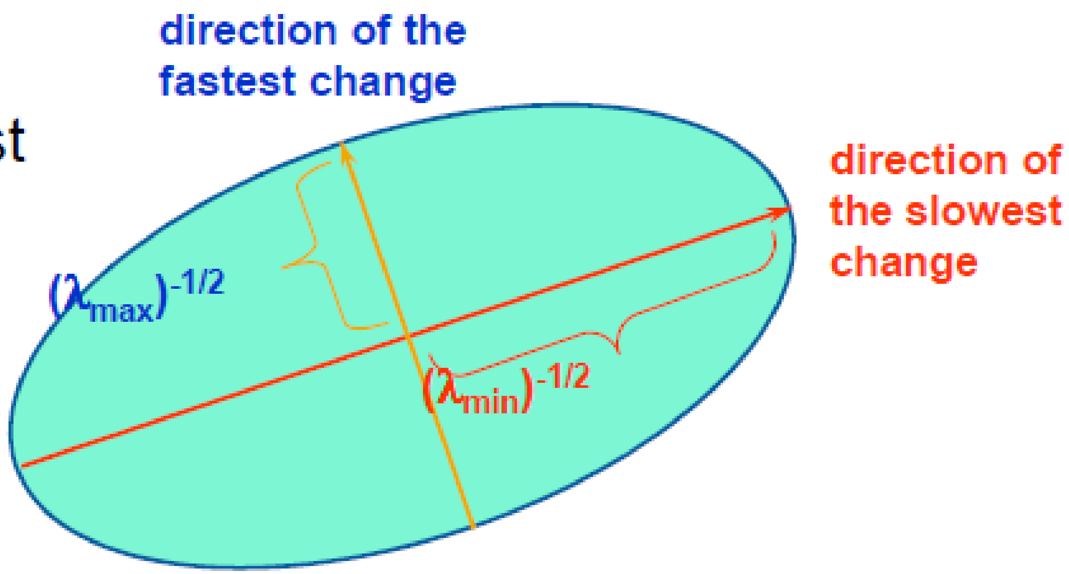
Harris Corner Detector – Cont.

Intensity change in shifting window: eigenvalue analysis

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

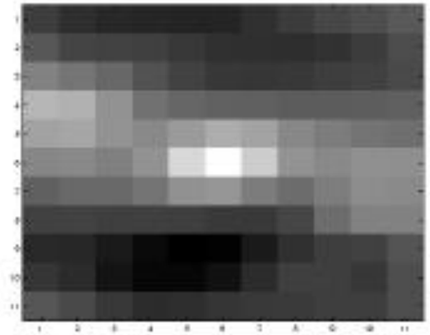
λ_1, λ_2 – eigenvalues of M

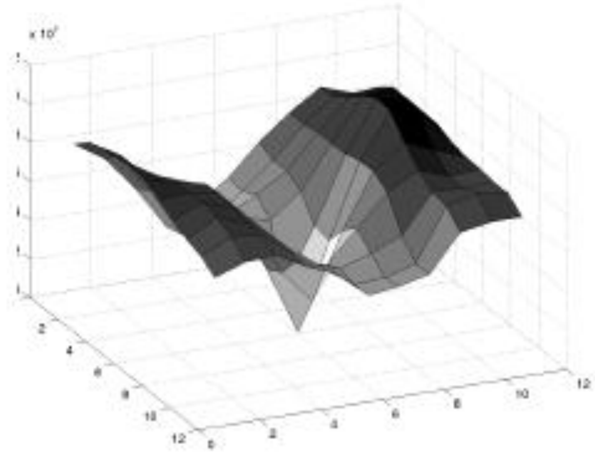
Ellipse $E(u,v)$ = const



Harris Corner Detectors – Cont.

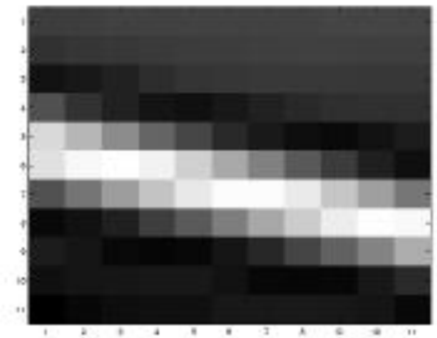
- Selecting Good Features

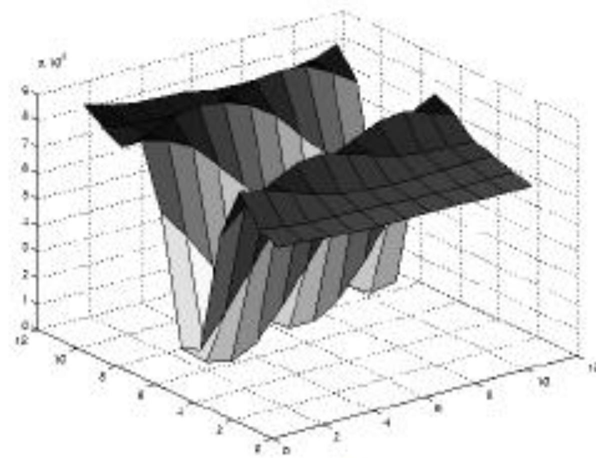




λ_1 and λ_2 are large

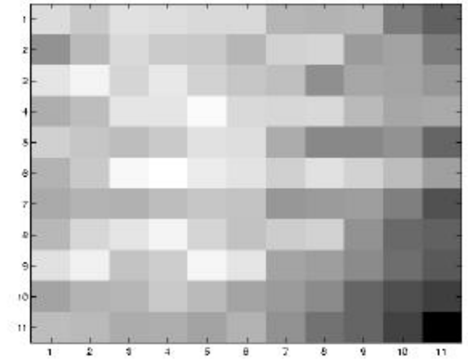
Harris Corner Detectors – Cont.

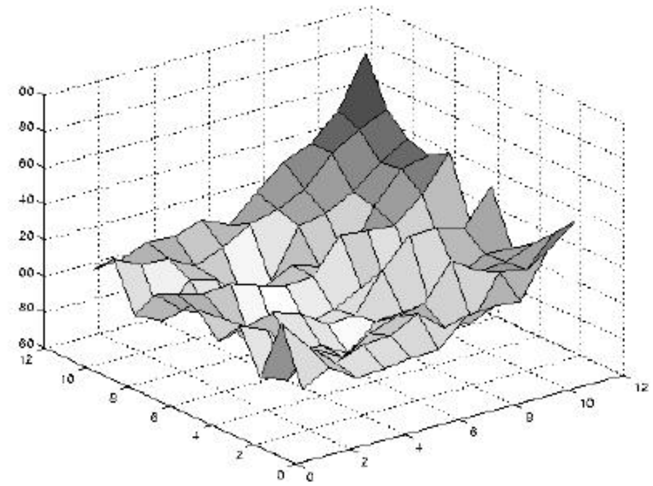




large λ_1, small λ_2

Harris Corner Detectors – Cont.

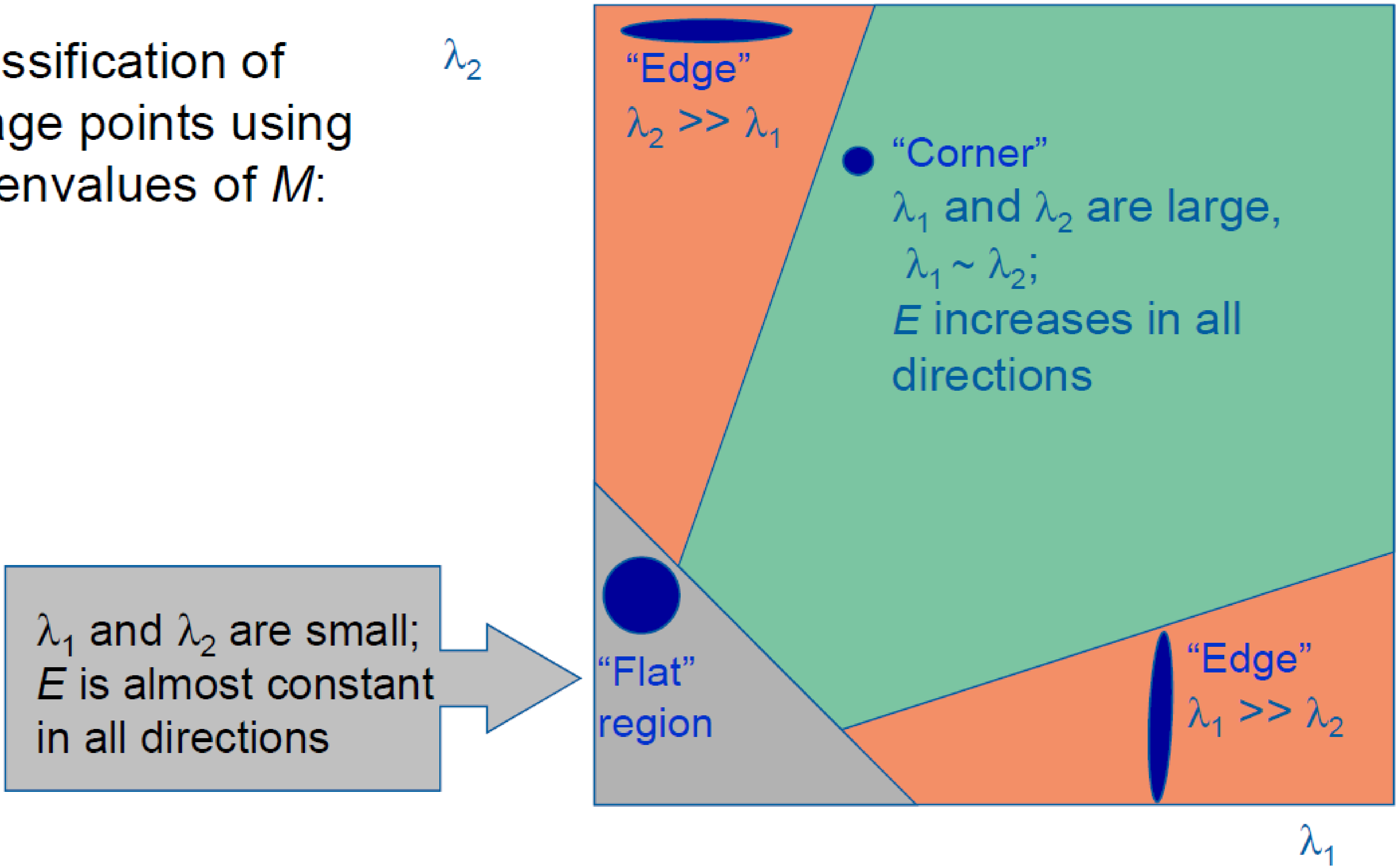




small λ_1, small λ_2

Harris Corner Detectors – Cont.

Classification of image points using eigenvalues of M:



Harris Corner Detectors – Cont.

- Measure of corner response

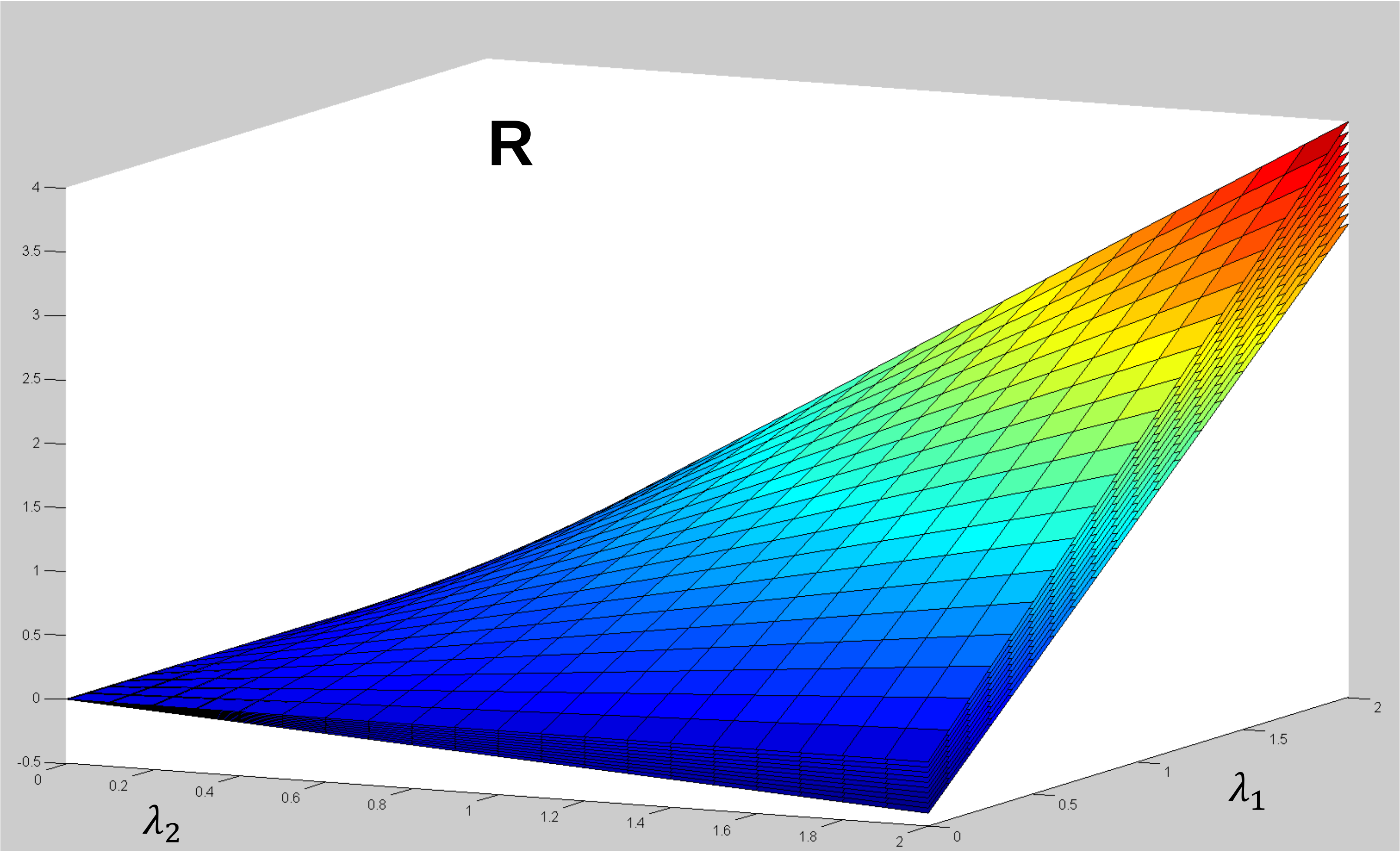
$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

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

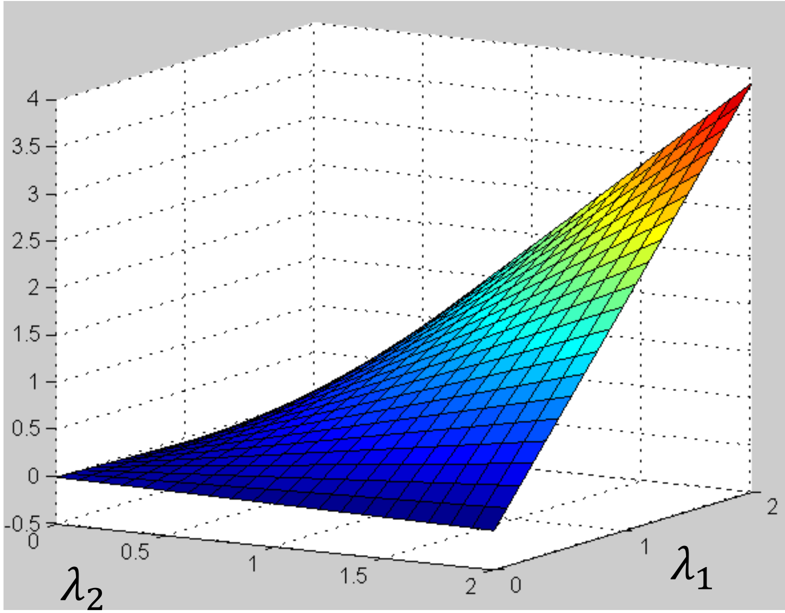
(k – empirical constant, k = 0.04-0.06)

Harris Corner Detectors – Cont.

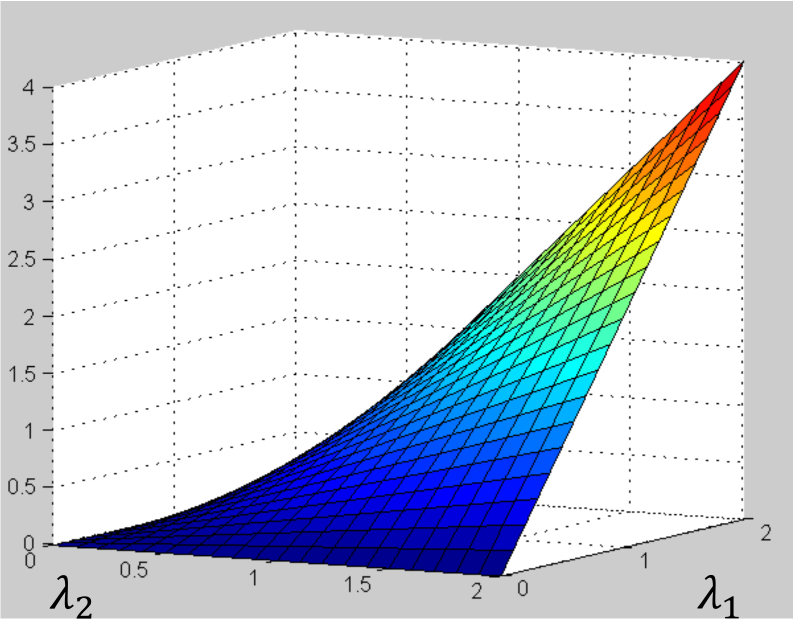
- Figure of **R,** k from 0 – 0.2



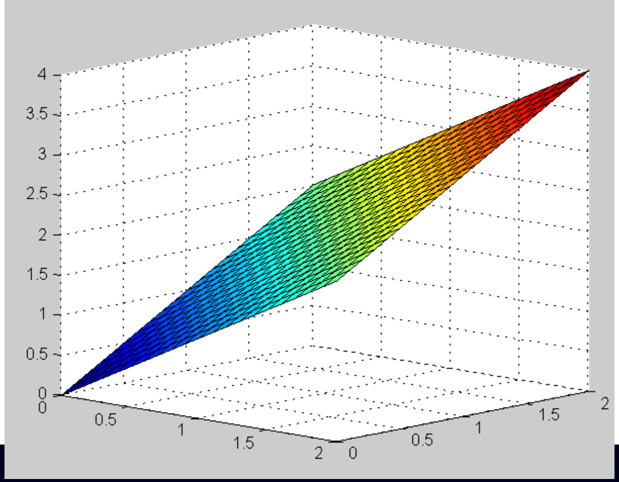
Harris Corner Detectors – Cont.



$R = \lambda_1\lambda_2 - 0.04(\lambda_1 + \lambda_2)$



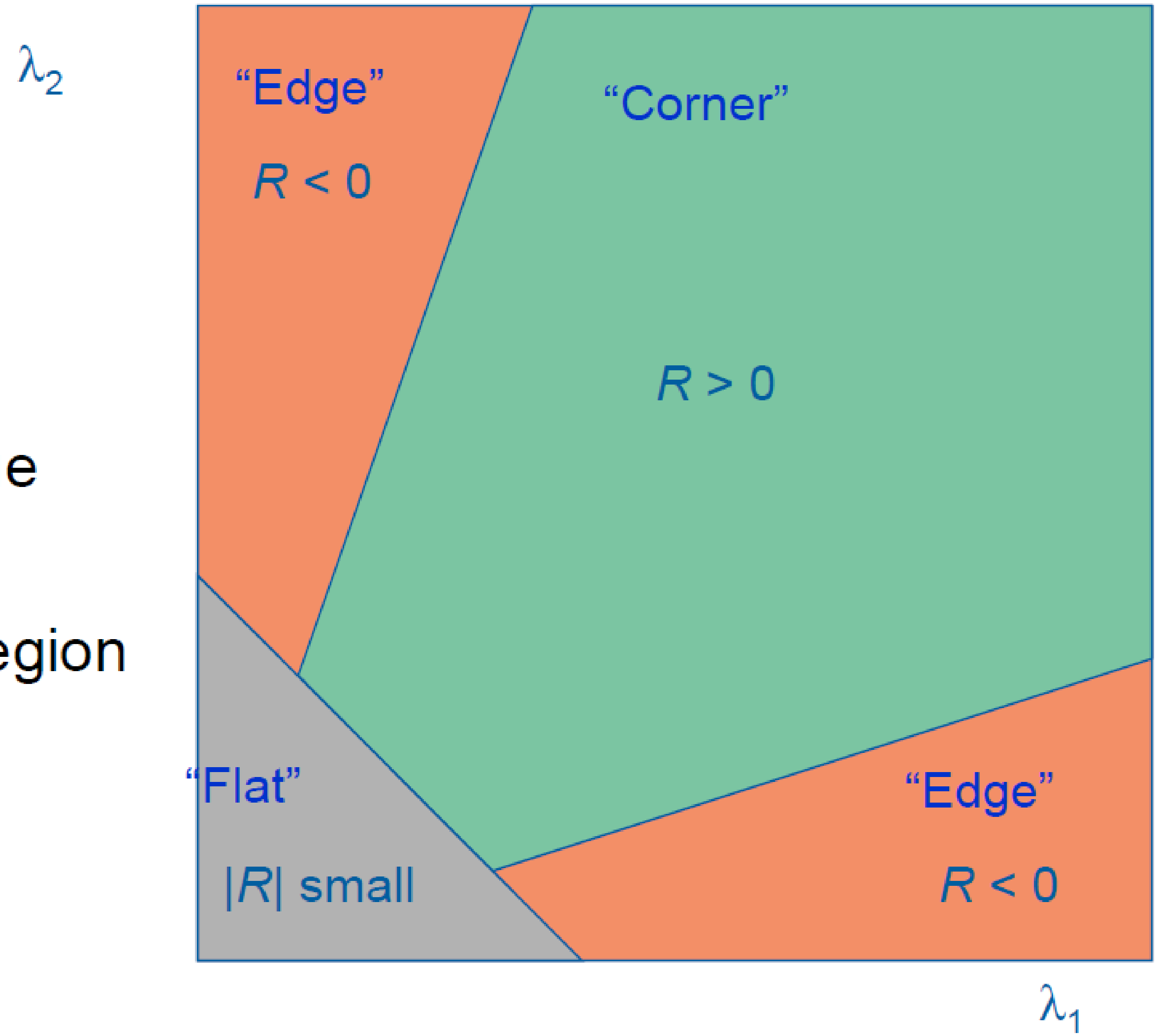
$R = \lambda_1\lambda_2$



$R = \lambda_1 + \lambda_2$

Harris Corner Detectors – Cont.

- R depends only on 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



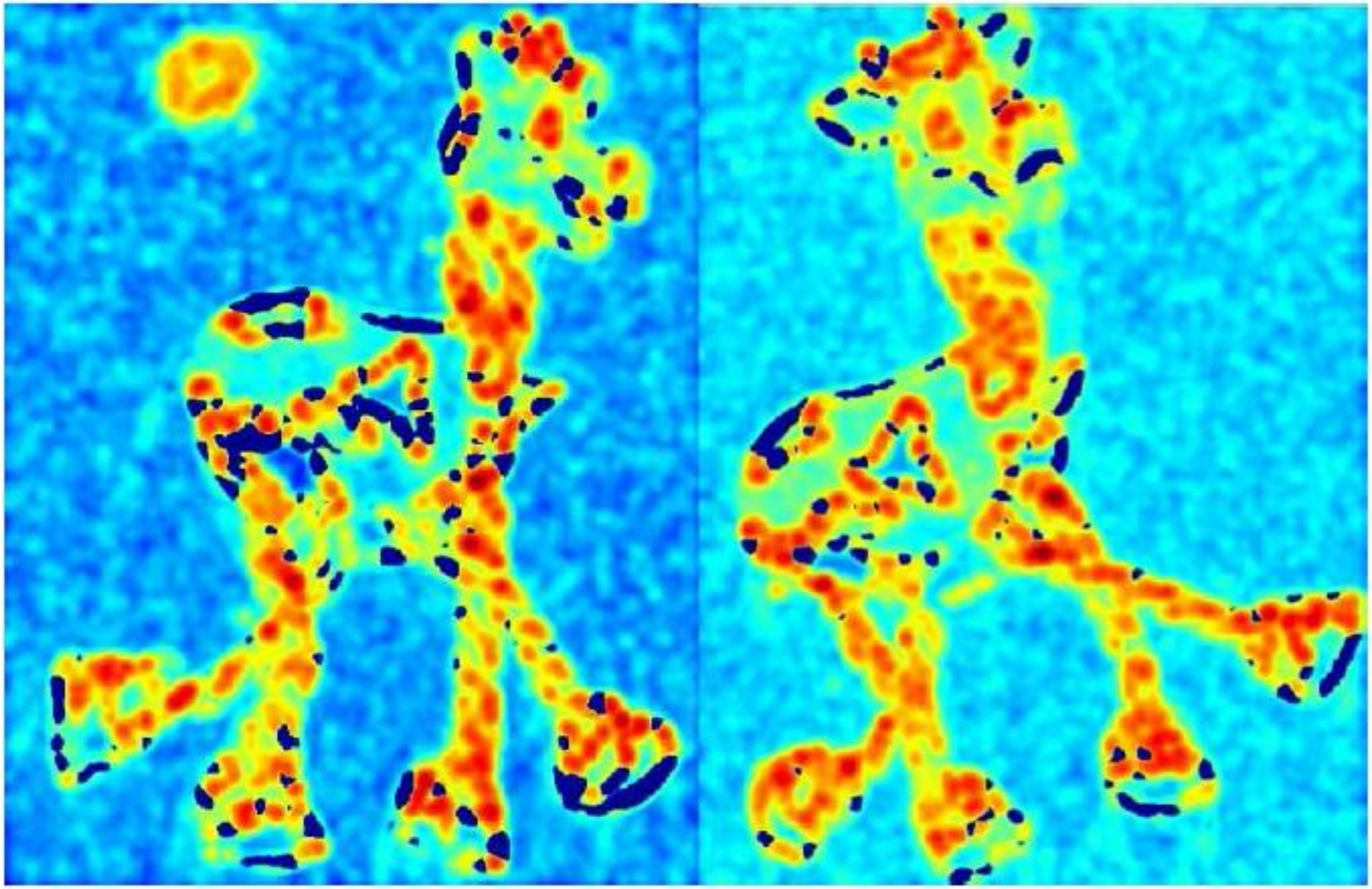
Harris Corner Detectors – Cont.

- The Algorithm:
  - Given a fixed window size, compute for each pixel the corner response function R
  - Find points with large corner response R (R > threshold)
  - Take the points of local maxima of R

Harris Corner Detectors – Cont.

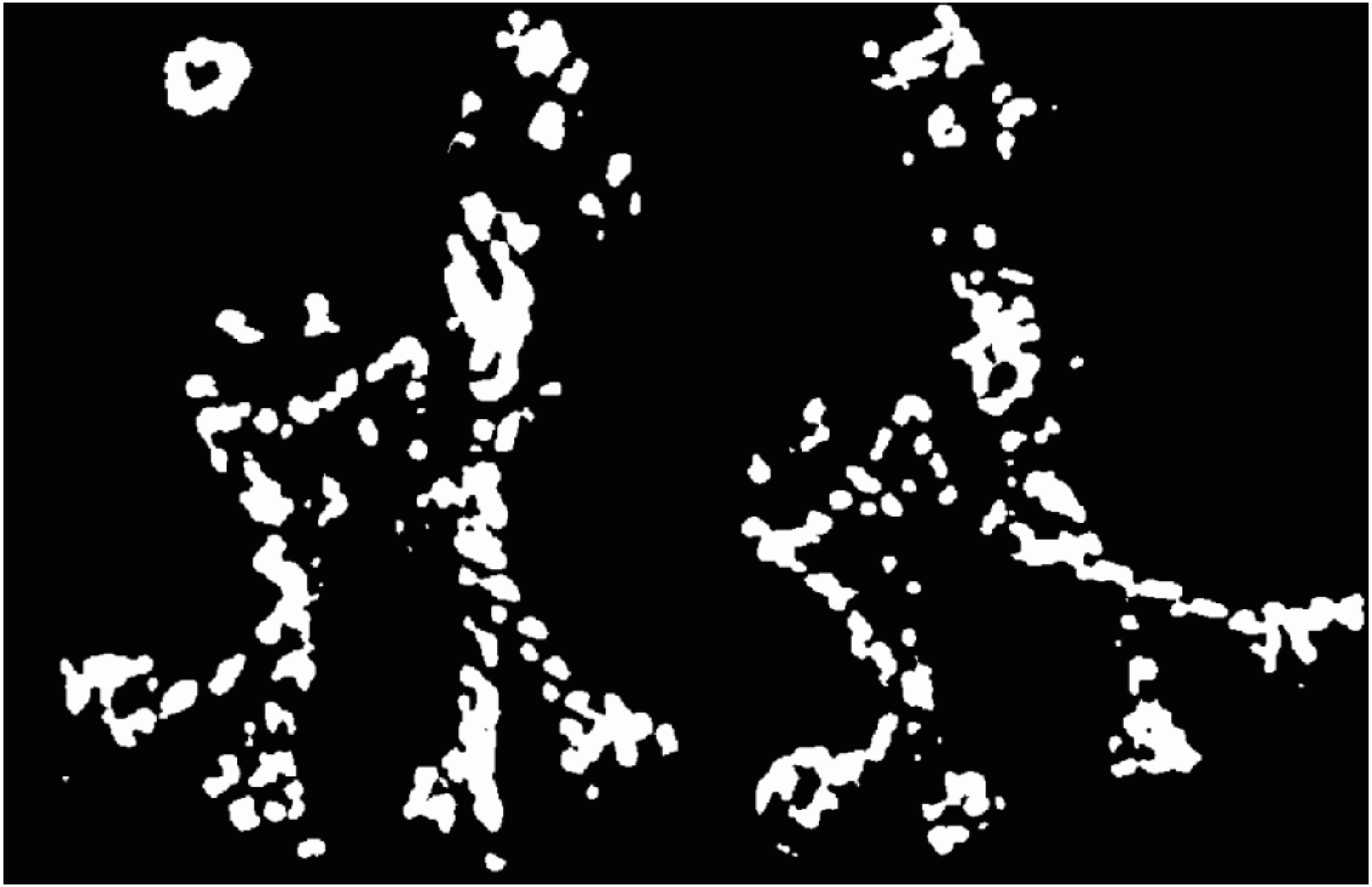
Harris Corner Detectors – Cont.

- Compute corner response R



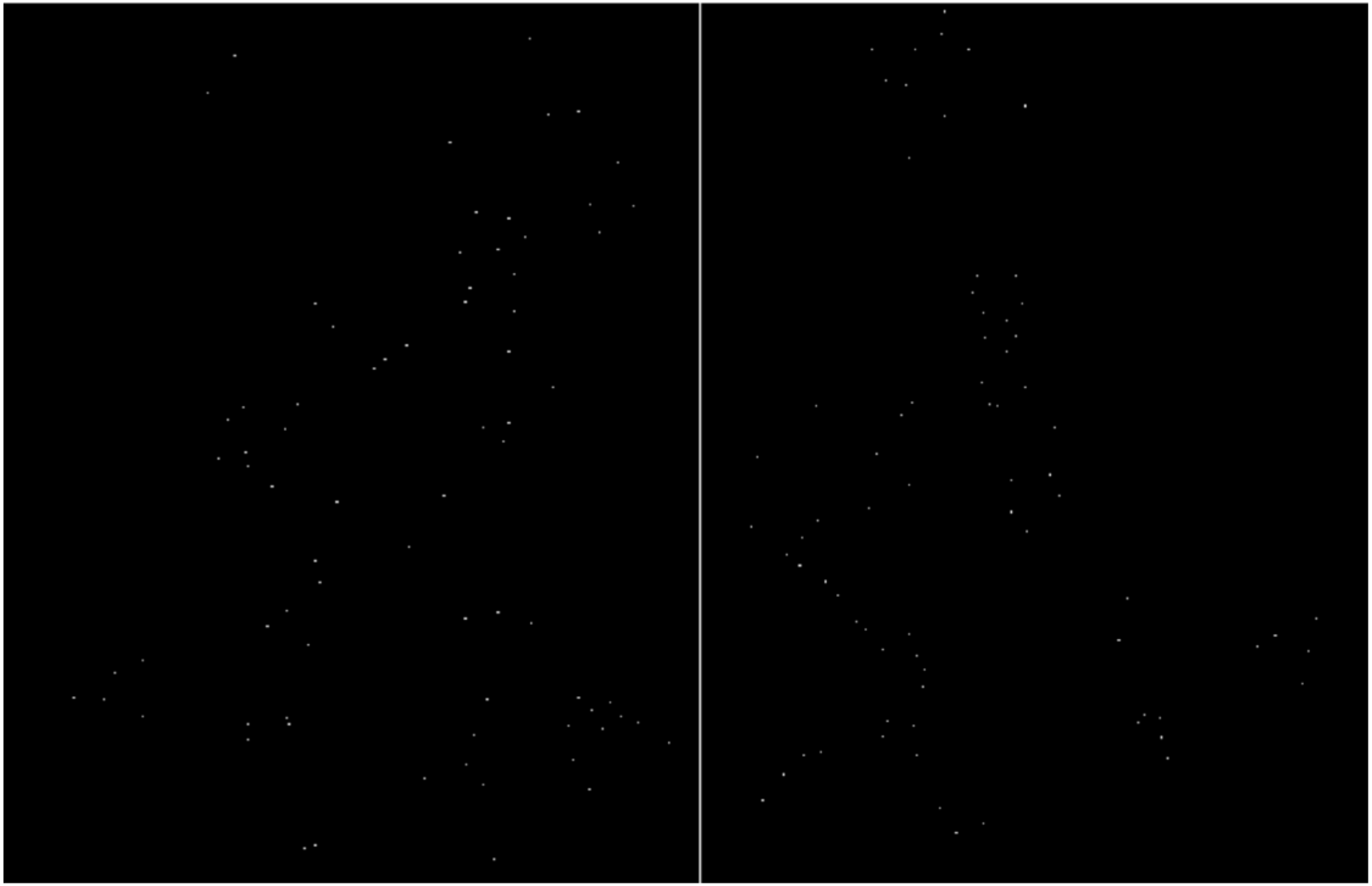
Harris Corner Detectors – Cont.

- Find points with large corner response R > threshold

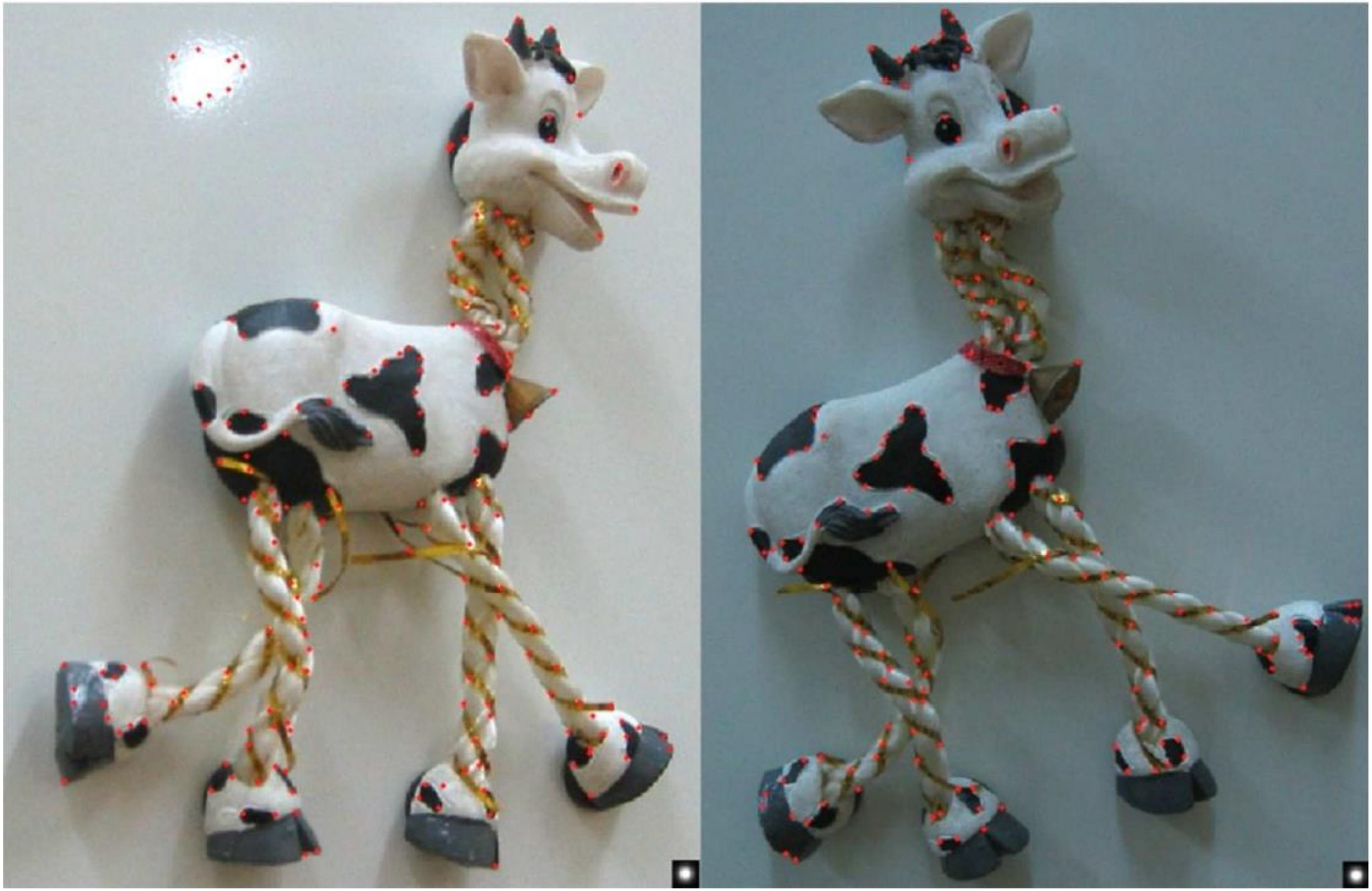


Harris Corner Detectors – Cont.

- Take only the points with local maxima of R (non-maximum suppression)



Harris Corner Detectors – Cont.



Harris detector - Summery

- Average intensity change in direction $[u,v]$ can be expressed as a bilinear form:

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

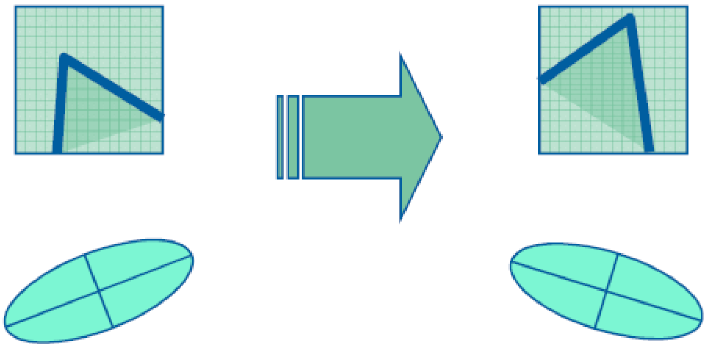
- Describe a point in terms of eigenvalues of M: *measure of corner response*

$$R = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2$$

- A good (corner) point should have a *large intensity change* in *all directions*, i.e. R should be large positive

Harris Detector – Properties

- Rotation invariance



Ellipse (iso-contour of the function E) rotates but its shape (i.e. eigenvalues) remains the same

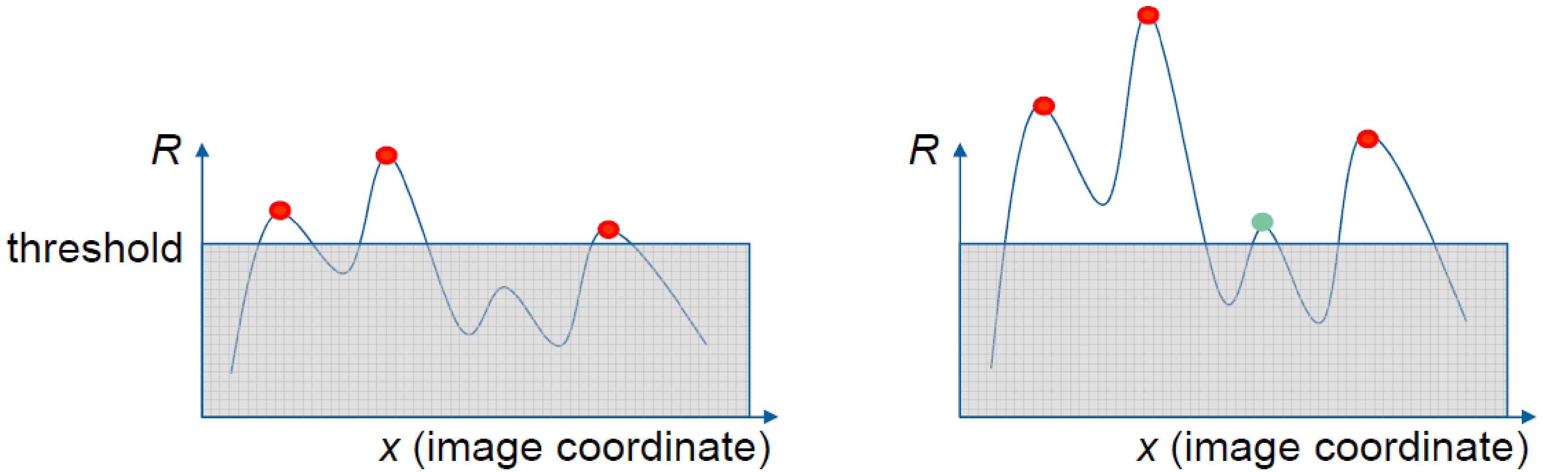
Corner response R is invariant to image rotation

Harris Detector – Properties – Cont.

- Partial invariance to additive and multiplicative intensity changes

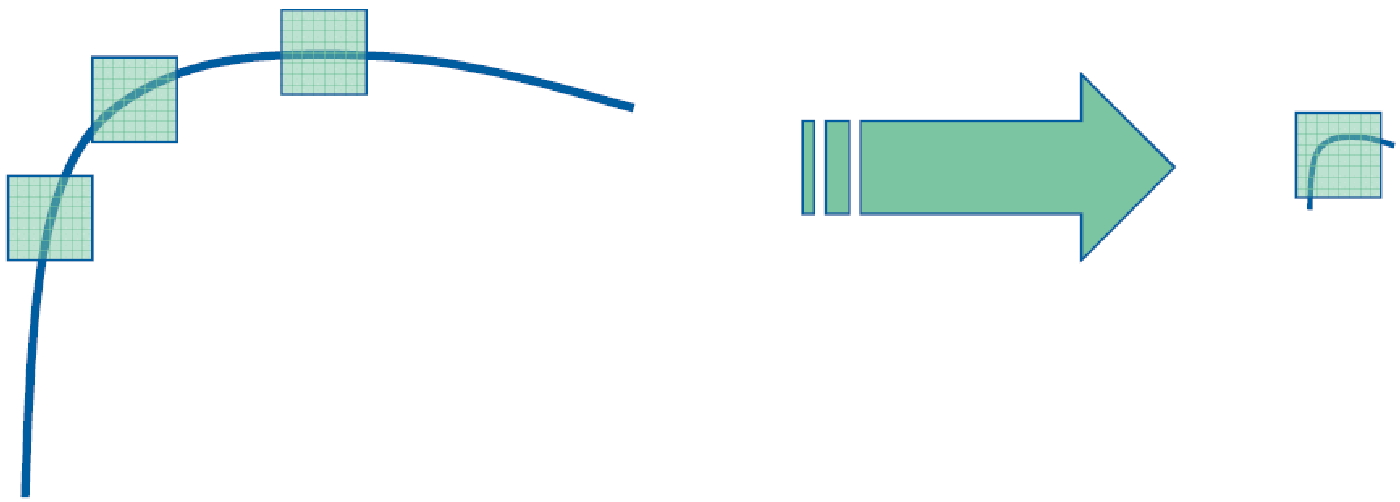
✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

✓ Intensity scale: $I \rightarrow a\,I$ Because of fixed intensity threshold on local maxima, only partial invariance to multiplicative intensity changes.



Harris Detector – Properties – Cont.

- How about scale?   -- No !



All points will be classified as edges

Corner !

How Do We Evaluate Detectors?

International Journal of Computer Vision 37(2), 151–172, 2000

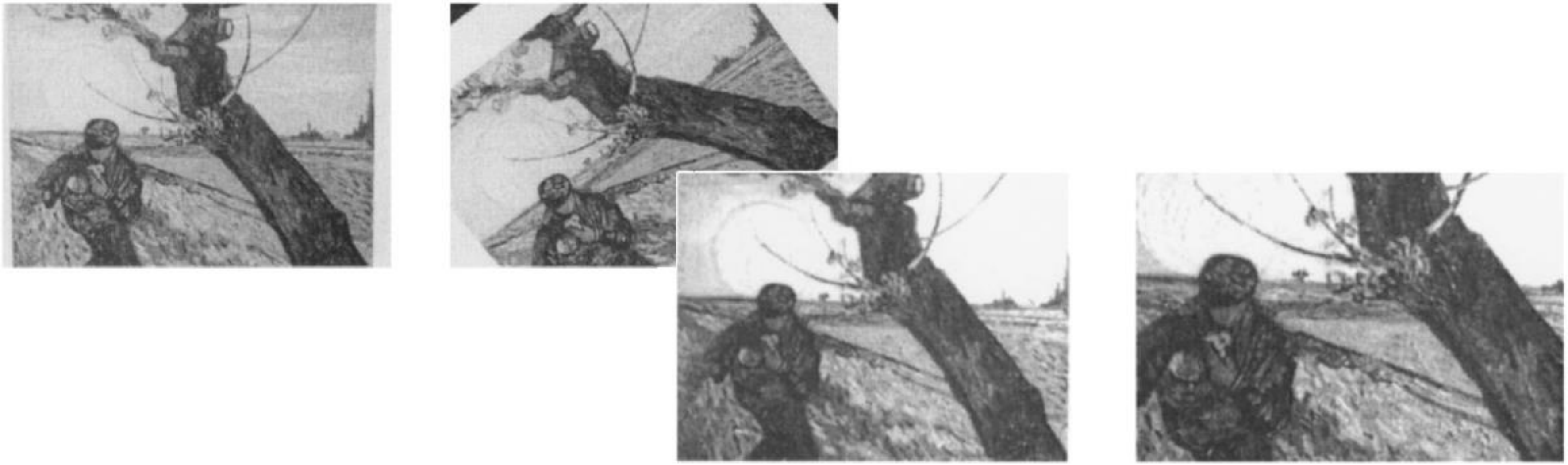
Evaluation of Interest Point Detectors

CORDELIA SCHMID, ROGER MOHR AND CHRISTIAN BAUCKHAGE
INRIA Rhône-Alpes, 655 av. de l'Europe, 38330 Montbonnot, France
Cordelia.Schmid@inrialpes.fr

Abstract. Many different low-level feature detectors exist and it is widely agreed that the evaluation of detectors is important. In this paper we introduce two evaluation criteria for interest points: repeatability rate and information content. Repeatability rate evaluates the geometric stability under different transformations. Information content measures the distinctiveness of features. Different interest point detectors are compared using these two criteria. We determine which detector gives the best results and show that it satisfies the criteria well.

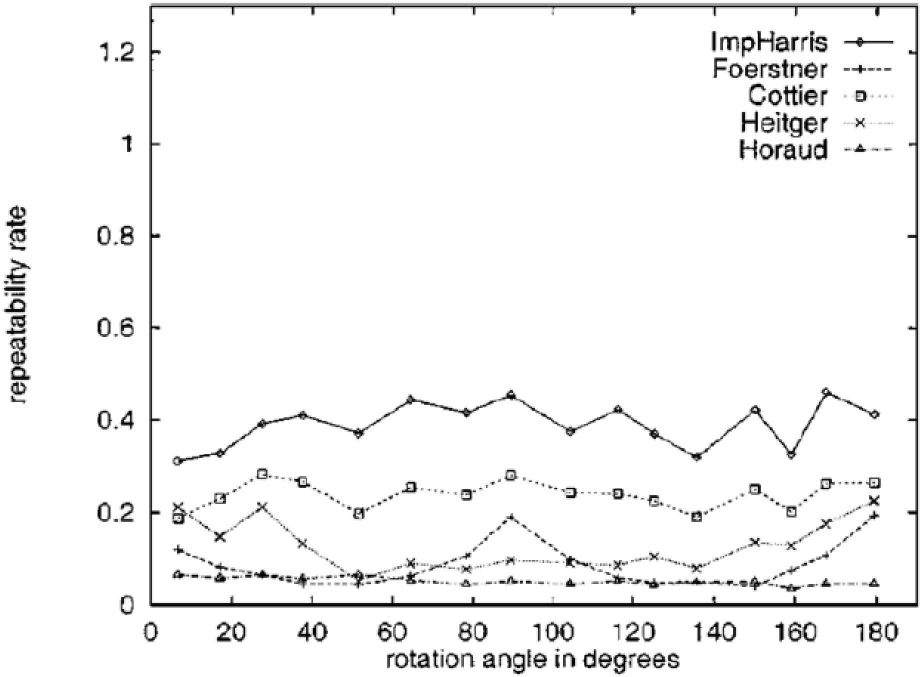
How Do We Evaluate Detectors?- Cont.

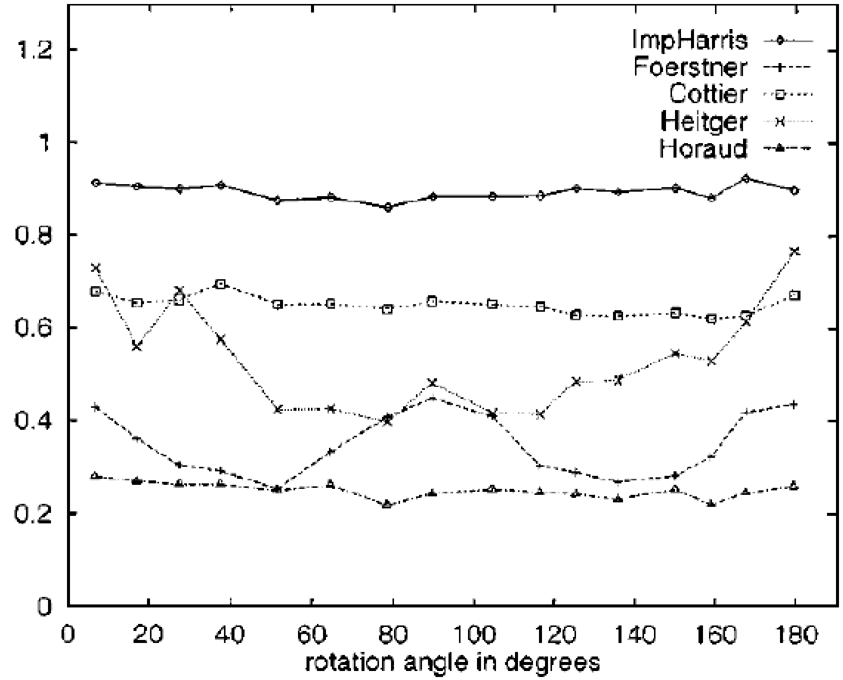
- Experimental setting:
  - Capture images of a planar scene with different rotation, zooming, noise and lighting configuration
  - Obtain perfect registration between these images separately
  - See if the same feature point can be detected in both images (within certain distance)
  - Quality is measured by $\text{Repeatability rate} = \dfrac{\#\ \text{correspondences}}{\#\ \text{possible correspondences}}$



Evaluation of Detectors

- Harris detector is invariant to rotations
  - The left and right are results when the threshold is 0.5 or 1.5 pixels
  - Note even when the rotation is 0, the repeatability ratio is not 1

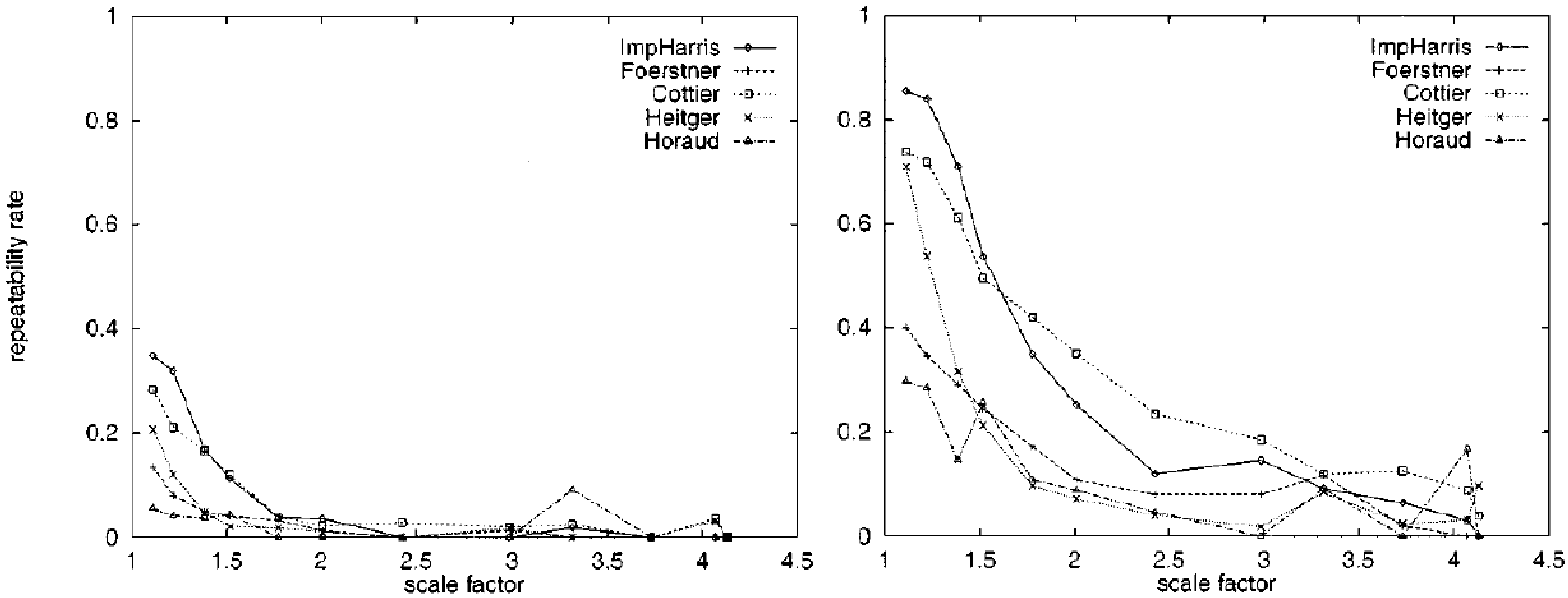




C.Schmid et.al. “Evaluation of Interest Point Detectors”. IJCV 2000

Harris Detector – Scale Variant

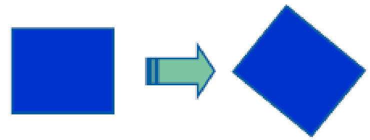
- Quality of Harris detector for different scale changes



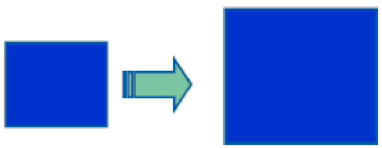
We want to detect the same points regardless of the images changes.

Model of Image Change

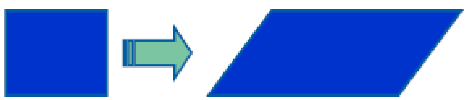
- Geometry
  - Rotation



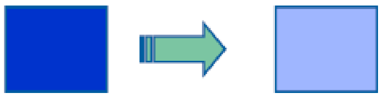
  - Similarity (rotation + uniform scale)



  - Affine (scale dependent on direction) valid for: orthographic camera, locally planar object



- Photometry
  - Affine intensity change ($I \rightarrow a\,I + b$)

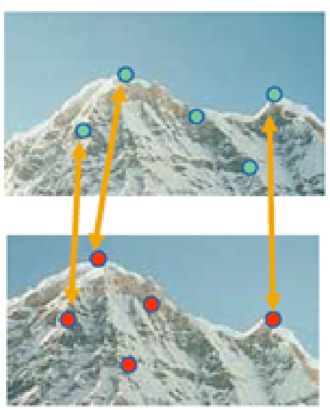


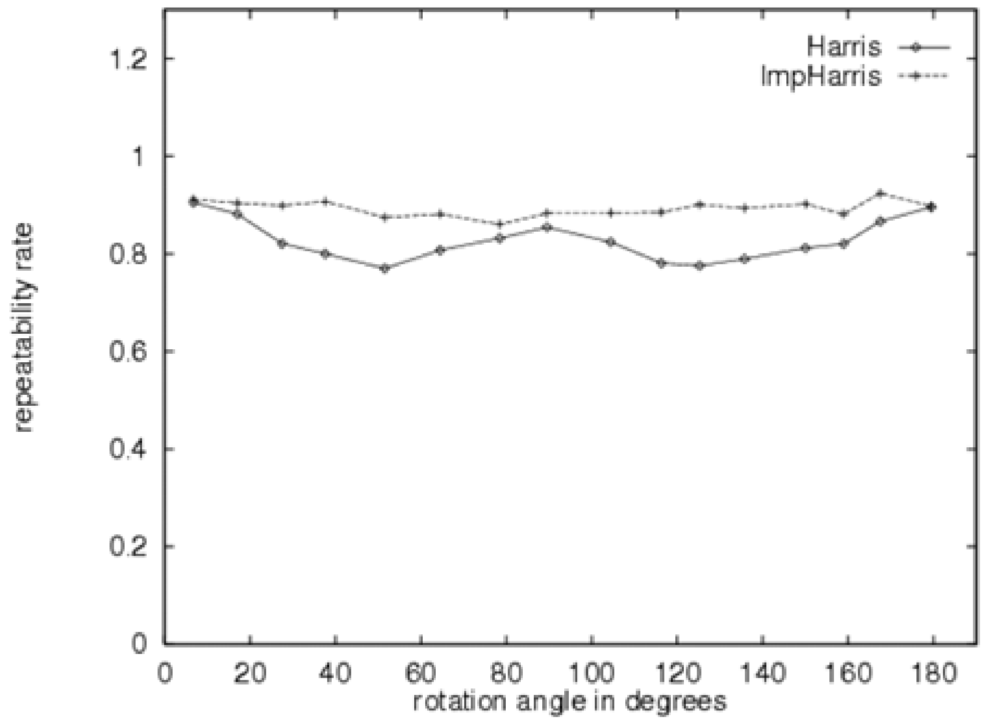
Rotation Invariance

- Harris detector is invariant to rotations
  - Note even when the rotation is 0, the repeatability ratio is not 1

Repeatability rate:

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



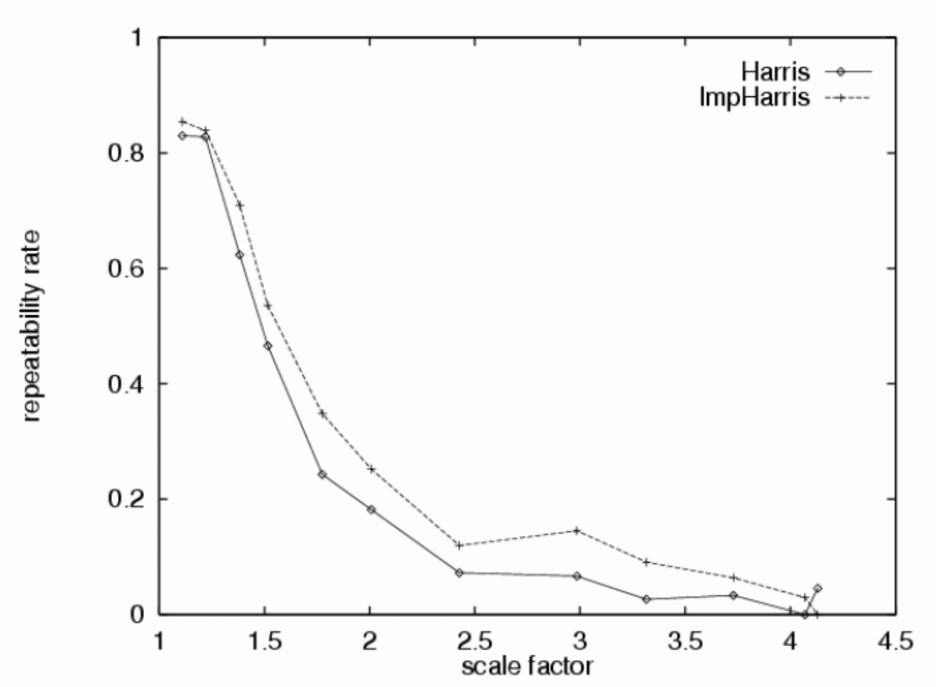


Imp.Harris uses derivative of Gaussian instead of standard template used by Harris et al.

C.Schmid et.al. “Evaluation of Interest Point Detectors”. IJCV 2000

Scale invariance

- Enlarge or shrink the image and see if the same points are being detected

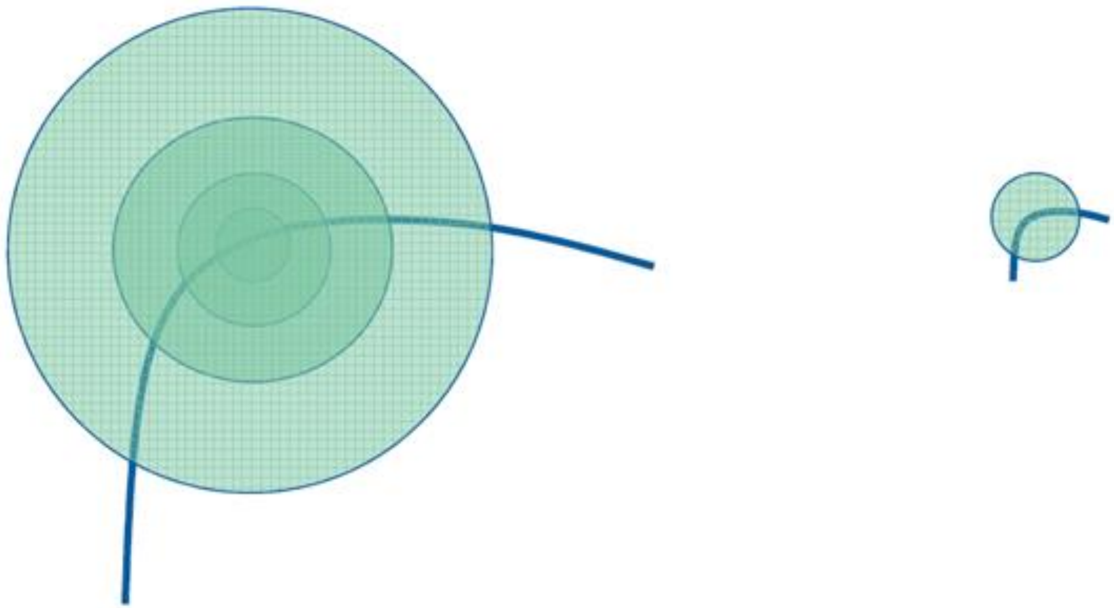


Harris detector gives poor results.

C.Schmid et.al. “Evaluation of Interest Point Detectors”. IJCV 2000

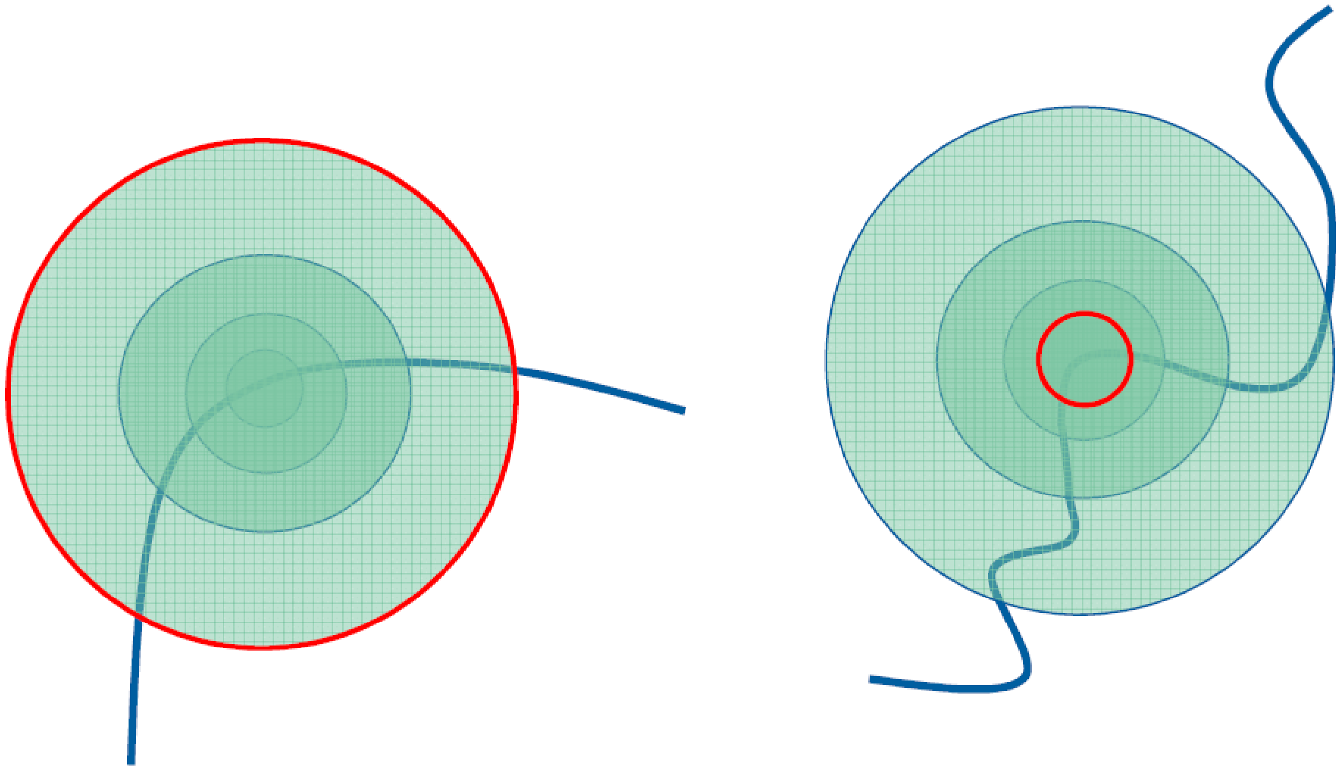
Scale-Invariant Detection

- 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 – Cont.

How to choose this scale independently? – note for point detector we do not know what is the scale of the other image.

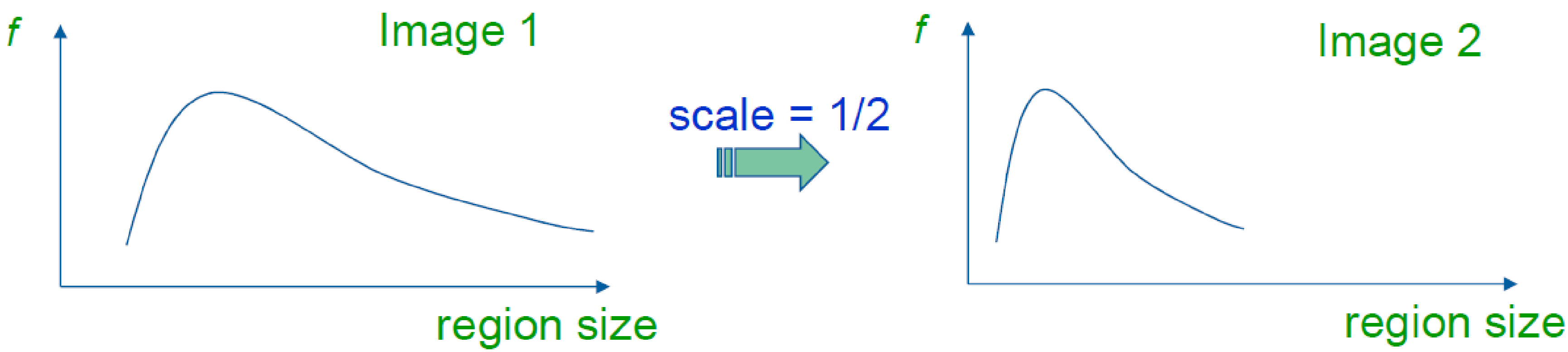


Scale-Invariant Detection – Cont.

- Solution: Design a function with respect to the size of the region (radius), find a distinct point on the function curve.

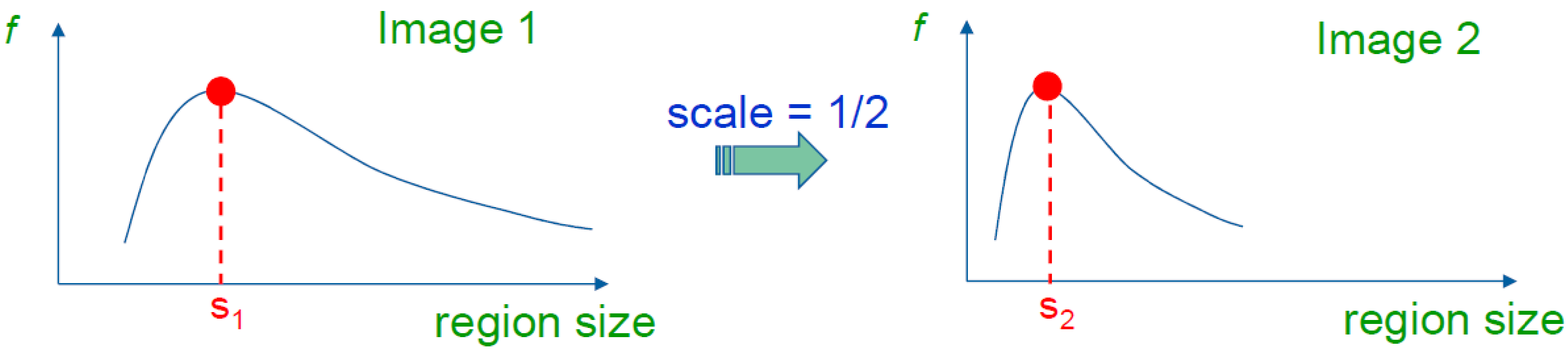
$$f_{I_1}(r), \quad f_{I_2}(r)$$

e.g. the average pixel value intensity within the circle



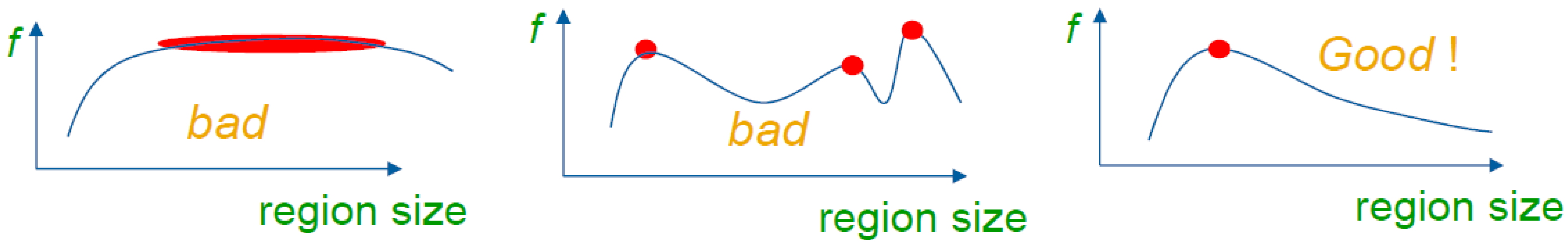
Scale-Invariant Detection – Cont.

- Take a local maximum of this function



Scale-Invariant Detection – Cont.

- A “good” function for scale detection:
  has one stable sharp peak



The basic rationale to have the function with distinctive peak such that it appears similarly in different images. Otherwise, if the function has multiple peaks with only slight difference, the maximum might change across different images (due to image noises and other factors.)

Scale-Invariant Detection – Cont.

Functions used in real images:

$$f(x,y) = (kernel * Image)(x,y)$$

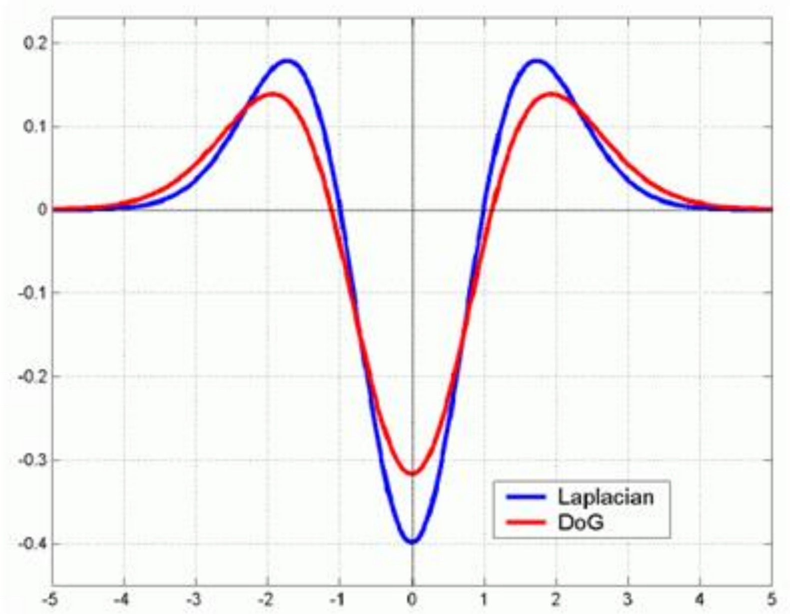
Kernels:

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

(Laplacian)

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

(Difference of Gaussians)



where Gaussian

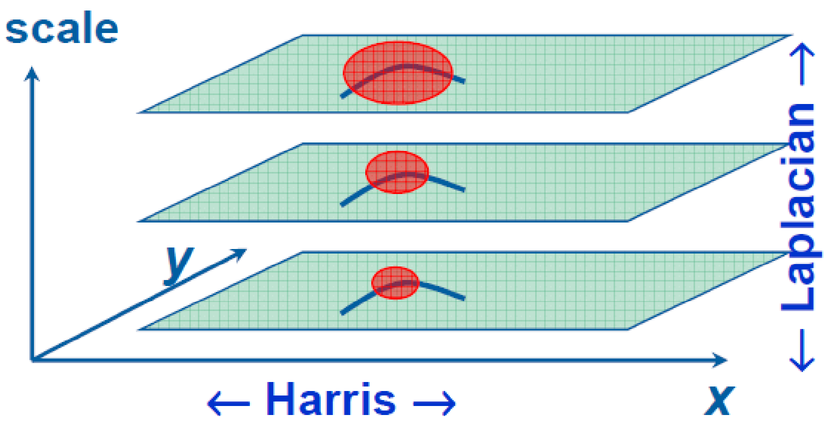
$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

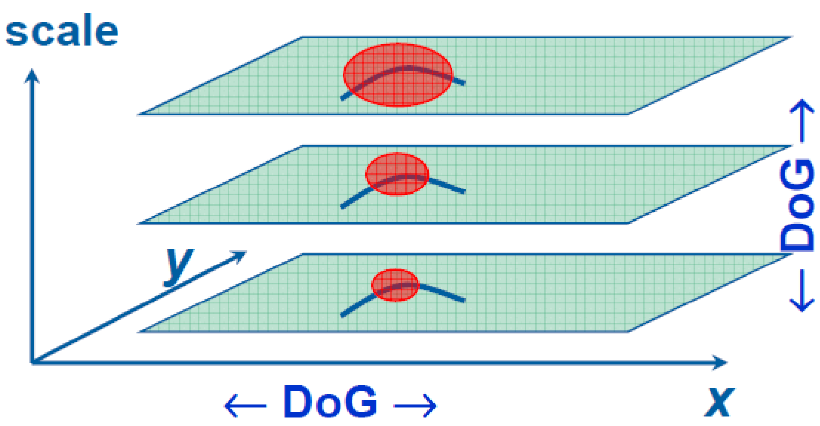
Note: both kernels are invariant to *scale* and *rotation*

Scale-Invariant Detection – Cont.

- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale
- SIFT (Lowe)[2]

  Find local maximum of:
  - *Difference of Gaussians in space and scale*





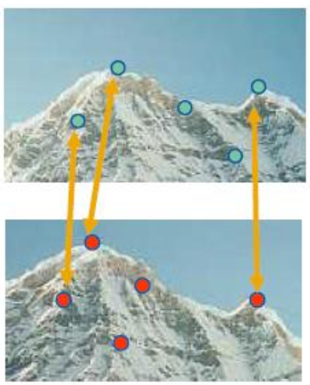
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004

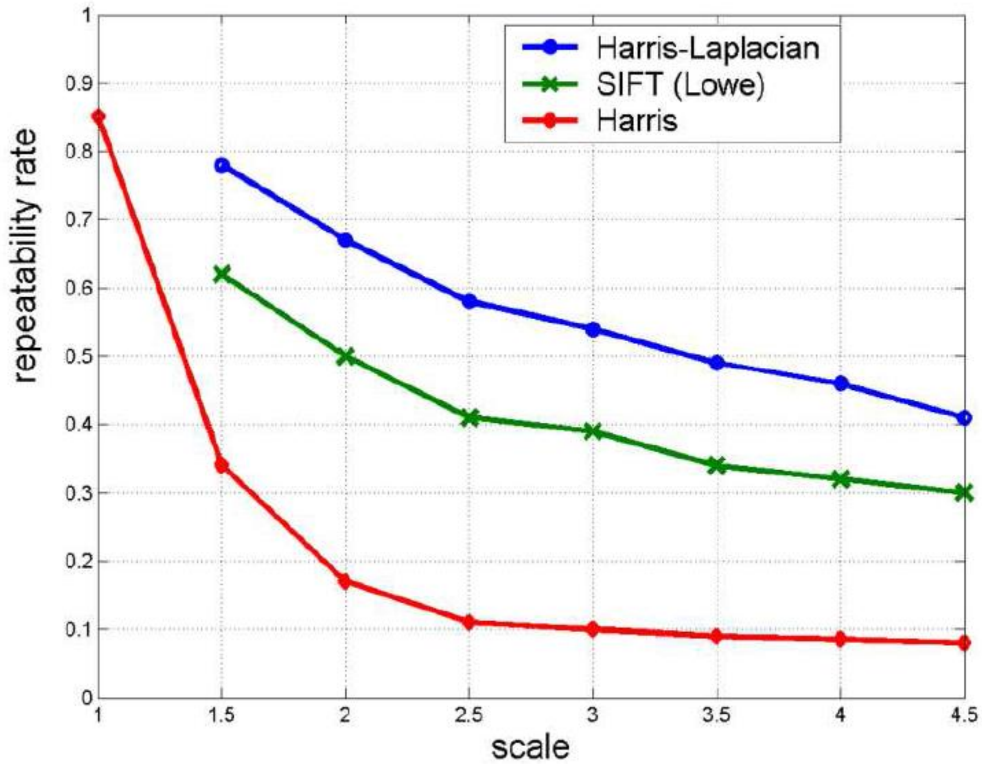
Scale-Invariant Detection – Cont.

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

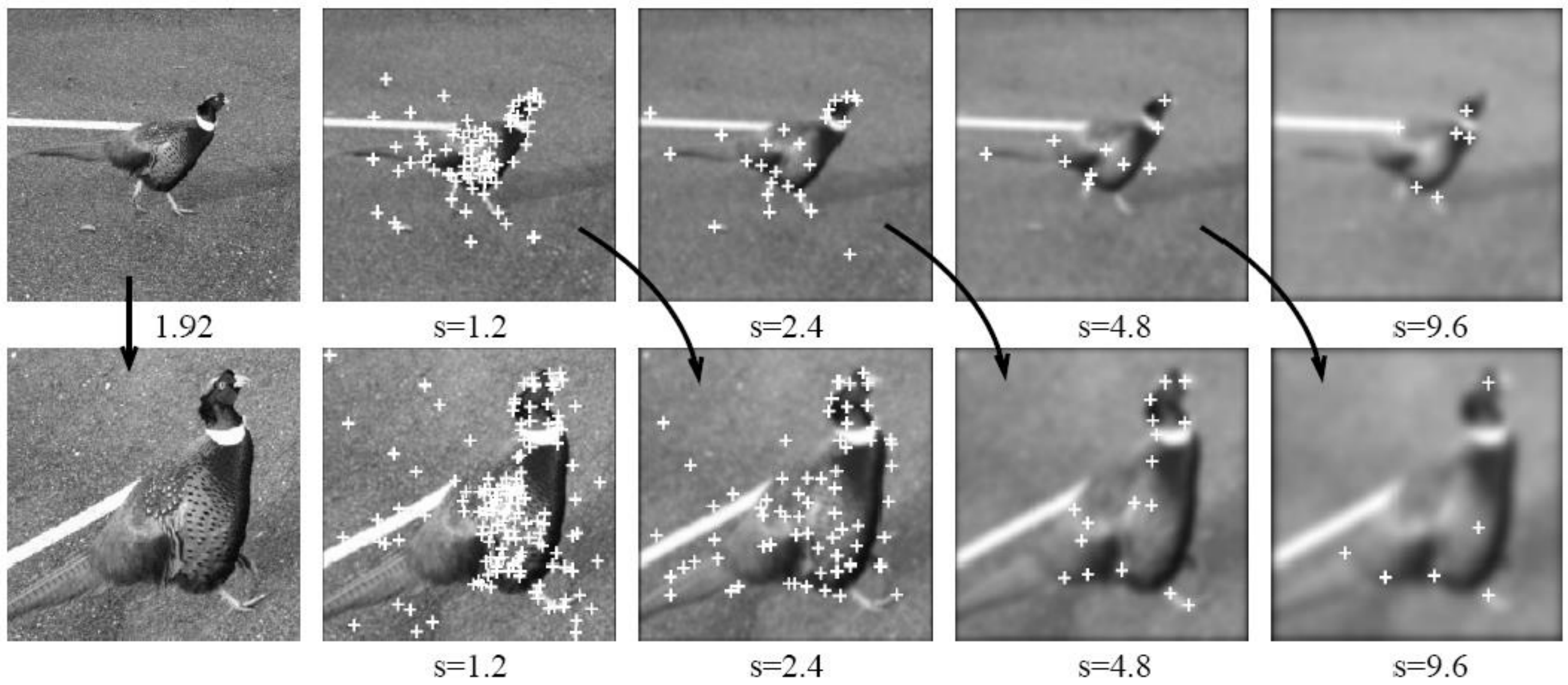
$$\frac{\text{\# correspondences}}{\text{\# possible correspondences}}$$





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

Examples of Laplacian Detectors



Scale-Invariant Detection – Summary

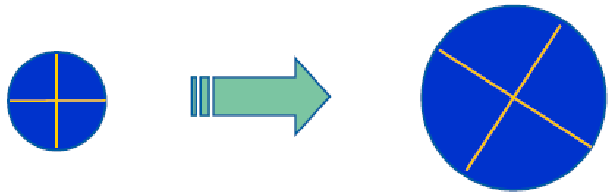
- Given: two images of the same scene with a large *scale difference* between them
- Goal: find *the same* interest points *independently* in each image
- Solution: search for *maxima* of suitable functions in *scale* and in *space* (over the image)

Methods:

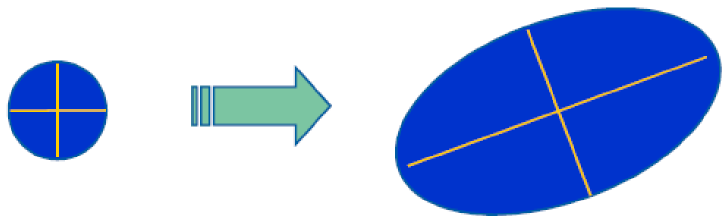
1. Harris-Laplacian [Mikolajczyk, Schmid]: maximize Laplacian over scale, Harris' measure of corner response over the image
2. SIFT [Lowe]: maximize Difference of Gaussians over scale and space

Affine-Invariant Detection

- Important when the image has perspective differences
  - Above we considered:
    Similarity transform (rotation + uniform scale)



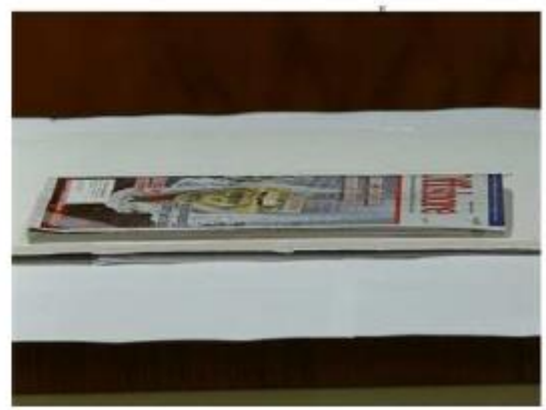
  - Now we go on to:
    Affine transform (rotation + non-uniform scale)



A circle is mapped to an ellipse now. We cannot just search in the scale space (the space of circle radius).

Affine-Invariant Detection

- Comparing to rotation-scale invariances, the affine distortion basically shows scale difference in orthogonal direction.

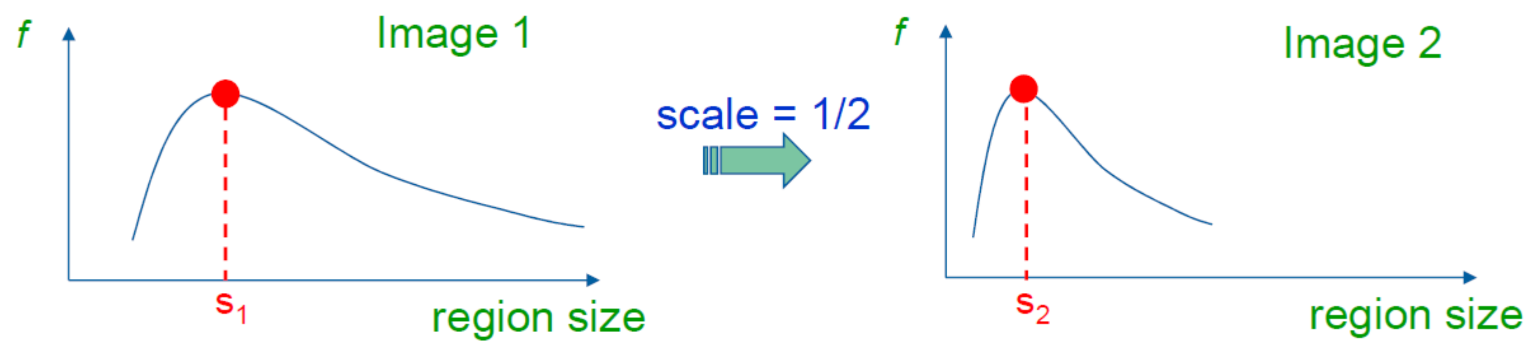


(Jean-Michel and Yu, 09)

Scale & Affine Invariant Interest Point Detectors

(Mikolajczyk and Schmid 04, IJCV)

Idea: Similar to the idea of finding the local maximum of a pre-defined function in the scale function:



Now, we have scale in two different directions, these two are orthogonal!

- So this should be a 3D function, 4D surface

Being, rotation of the major axis, length of the major axis, length of the minor axis

(Jean-Michel and Yu, 09)

Apparently defining a function over such a high-dimensional function is challenging

Scale & Affine Invariant Interest Point Detectors – Cont.

(Mikolajczyk and Schmid 04, IJCV)

Harris-scale space is represented as

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix}$$
$$= \sigma_D^2\, g(\sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}$$

$L = g(\sigma_D) * I,$ g can be non-uniform Gaussian kernel, the detection of harris corner points can be performed at different scale / non-uniform scale

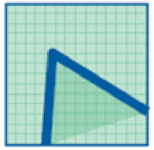
$$g(x, \Sigma_D) = \frac{1}{2\pi^k \sqrt{\det(\Sigma_D)}} e^{-\boldsymbol{x}^T \Sigma_D \boldsymbol{x}}$$

Therefore, $g(x, \Sigma_D)$ can non-uniformally sample the image, given the covariance matrix

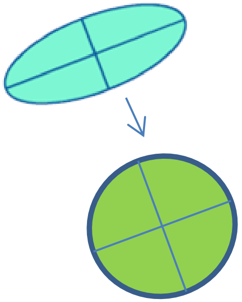
Scale & Affine Invariant Interest Point Detectors – Cont.

- We still want to calculate the gradient response as sift does, but instead of using the DOG, we use LOG.

how to deal with the scale differences in the orthogonal direction? – If we use circle, this will not address the affine-variations, we want to use a ellipse region that shows affine variation.



Recall that Harris detector find points with large eigenvalues of the second moment matrix, we want to find elliptical region that makes $\frac{\lambda_{min}}{\lambda_{max}}$ is close to 1 - Isotropy. This needs iterative solution of the following to get the optimal Σ_D



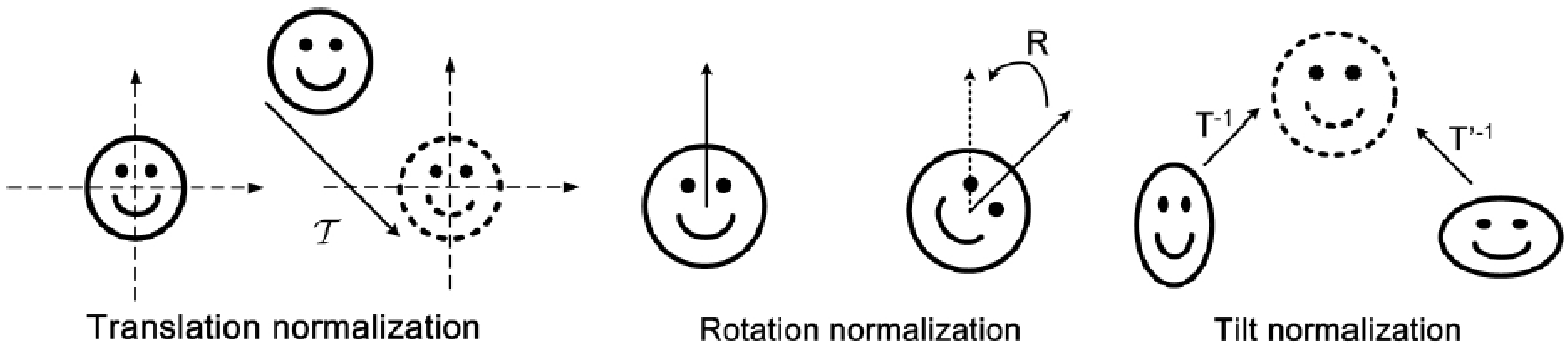
$$|\Sigma_D|^2 g(\Sigma_I) \begin{bmatrix} L_{xx}(\boldsymbol{x}, \Sigma_D) & L_{xy}(\boldsymbol{x}, \Sigma_D) \\ L_{xy}(\boldsymbol{x}, \Sigma_D) & L_{yy}(\boldsymbol{x}, \Sigma_D) \end{bmatrix}$$

$$\Sigma_I = s\Sigma_D$$

ASIFT – Affine SIFT

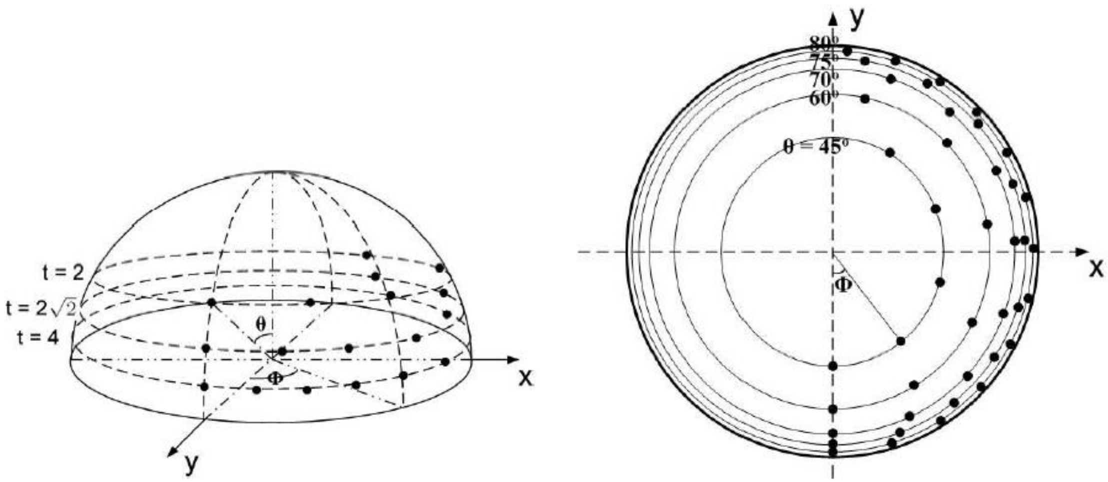
(Jean-Michel and Yu, 09, SIAM Journal)

Idea: since SIFT has gained a huge success, why not just artificially generate affine distorted image and extract points in many of the possible scenarios?



Affine

ASIFT – Affine SIFT



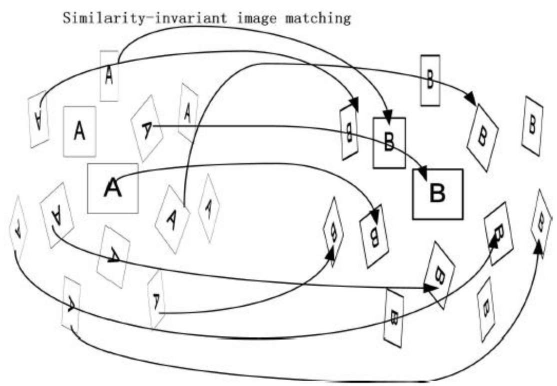
(Jean-Michel and Yu, 09, SIAM Journal)

Affine sampling

1. Each image is transformed by simulating all possible affine distortions caused by the change of camera optical axis orientation from a frontal position.

2. Extract SIFT points for all possible simulated images

3. Match all possible pairs when comparing two images.



ASIFT – Affine SIFT

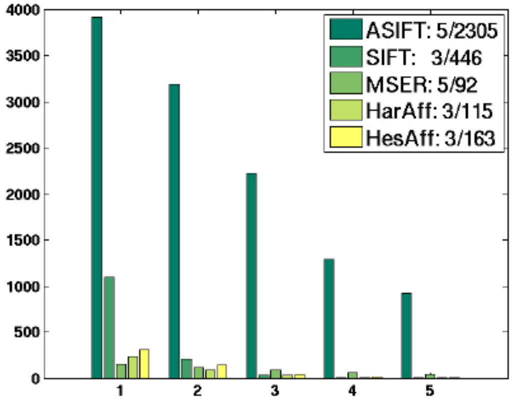
- Computation speed is a big issue – two resolution approach for matching

1. Subsample all simulated views.

2. Apply sift matching through all possible view pairs on low resolution images

3. Find M best pairs

4. Run sift matching on the original resolution images.



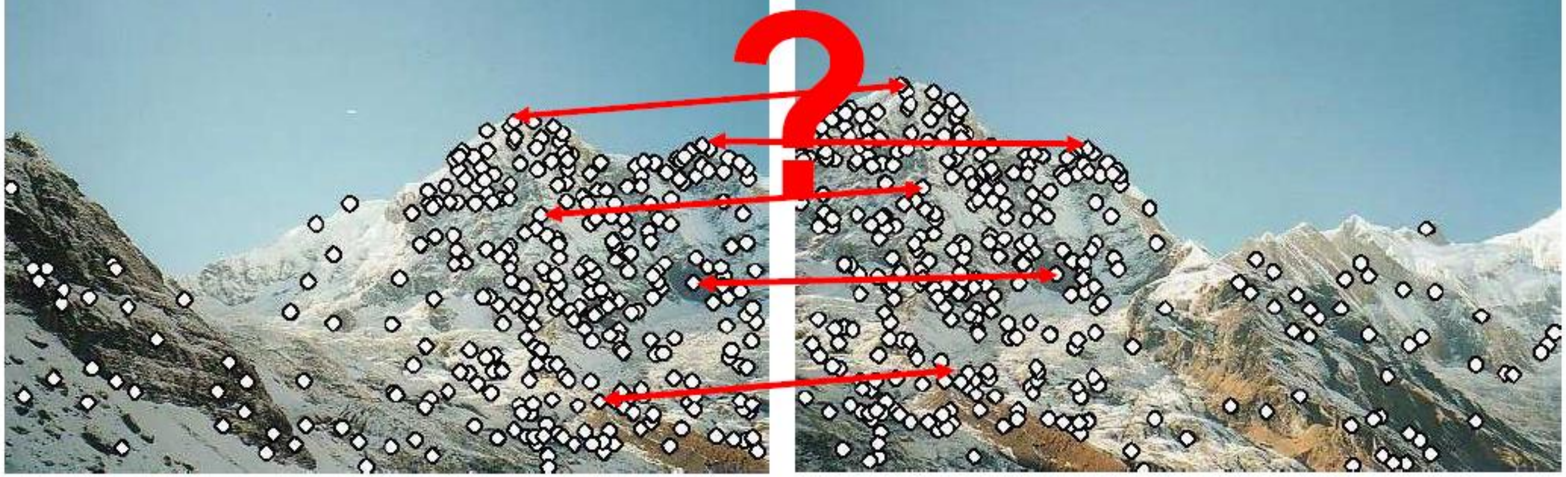
(Jean-Michel and Yu, 09, SIAM Journal)

Top-right corner m/n gives for each method the number of image pairs m on which more than 20 correct matches were detected, and the average number of matches n over these m pairs

Point Matching

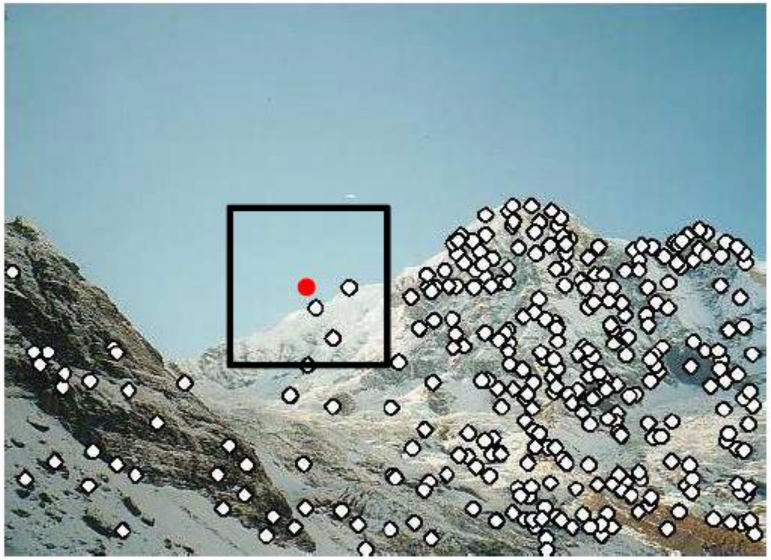
- We know how to detect points
- Next question:

How to match them?



Point Descriptors

- A simplest idea:
  - Take a local window of some pre-specified size at the detected feature point
  - Concatenate all pixel values within the window to form a vector – descriptor
  - Match features by evaluating the distance between these vectors



The descriptor can be seen as the signature of the point, for each point, we want to compare the points across different images and find those whose signature is similar as matches.

Point Matching Criteria

- Matching by thresholds: any two features are in correspondence if their distance is less than a threshold.
- Nearest neighbor matching: each feature is matched to its nearest neighbor if the distance between it and this neighbor is less than a threshold.
- Nearest neighbor distance ratio: find the nearest and the second nearest neighbor. If the ratio between the distance to the nearest neighbor and the second nearest neighbor is less than a threshold, the feature is matched to its nearest neighbor.

Point Descriptors

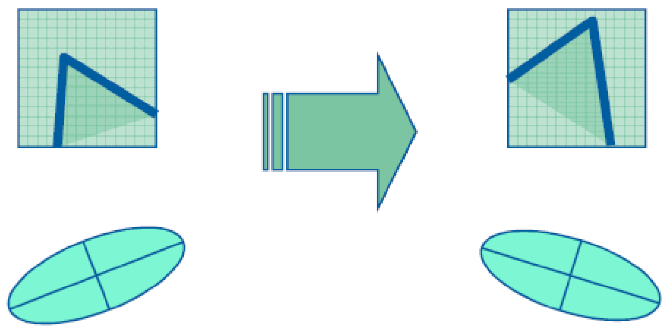
Like the detectors, we want the descriptors of the same points being rotation-invariant, scale-invariant.

Descriptors Invariant to Rotation

- (1) Harris corner response measure:
  depends only on the eigenvalues of the matrix *M*

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

$$R = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2$$

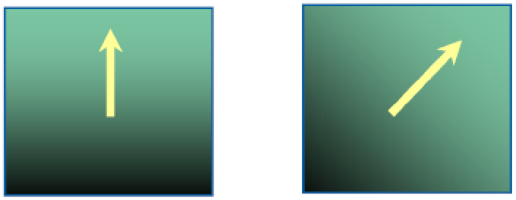


C.Harris, M.Stephens. "A Combined Corner and Edge Detector". 1988

Descriptors Invariant to Rotation – Cont.

- (2) Find local orientation

Dominant direction of gradient



- Compute image derivatives relative to this orientation

Example: SIFT descriptor – more details late

[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

Descriptors Invariant to Scale

- Use the scales determined by the detector and compute the descriptor in a normalized frame.

E.g. :

Moments integrated through an adapted window

Derivatives adapted to a scale

Affine-invariant Descriptor

- Affine invariant color moments

$$m_{pq}^{abc} = \int_{region} x^p y^q R^a(x,y) G^b(x,y) B^c(x,y)\, dx dy$$

Different combinations of these moments are fully affine invariant

Also invariant to affine transformation of intensity $I \rightarrow a\,I + b$

F.Mindru et.al. "Recognizing Color Patterns Irrespective of Viewpoint and Illumination". CVPR99

Affine-invariant Descriptor

- These need to be normalized of course:

Generic: $m^{abc}{}_{pq}/(m^{abc}{}_{00})^{p+q}$

More complex forms used in (Tuytelaars and Van Gool BMVC 2000)

$$inv[1] = S_{12}^{R} = \frac{\left\{ \begin{array}{l} M_{10}^{200} M_{01}^{100} M_{00}^{000} - M_{10}^{200} M_{00}^{100} M_{01}^{000} - M_{01}^{200} M_{10}^{100} M_{00}^{000} \\ + M_{01}^{200} M_{00}^{100} M_{10}^{000} + M_{00}^{200} M_{10}^{100} M_{01}^{000} - M_{00}^{200} M_{01}^{100} M_{10}^{000} \end{array} \right\}^2}{(M_{00}^{000})^2 \left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2 \right]^3}$$

$$inv[2] = S_{12}^{G} \qquad (similar)$$

$$inv[3] = S_{12}^{B} \qquad (similar)$$

$$inv[4] = D_{02}^{RG} = \frac{\left[M_{00}^{110} M_{00}^{000} - M_{00}^{100} M_{00}^{010} \right]^2}{\left[M_{00}^{200} M_{00}^{000} - (M_{00}^{100})^2 \right] \left[M_{00}^{020} M_{00}^{000} - (M_{00}^{010})^2 \right]}$$

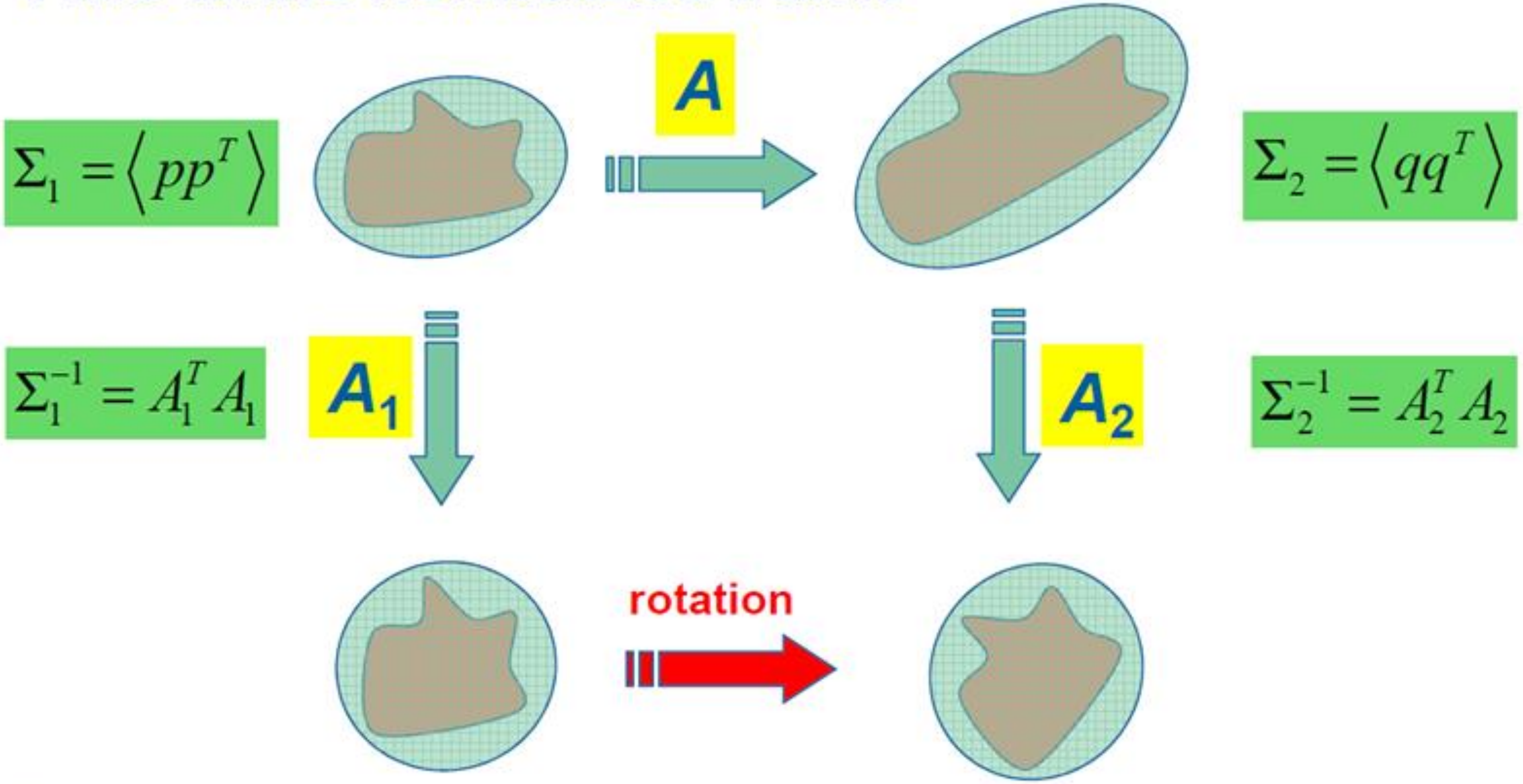
$$inv[5] = D_{02}^{GB} \qquad (similar)$$

$$inv[6] = D_{02}^{BR} \qquad (similar)$$

Different invariant combinations as the feature vector

Affine-invariant Descriptor – II

- Find affine normalized frame



Compute rotational invariant descriptor in this normalized frame

J.Matas et.al. "Rotational Invariants for Wide-baseline Stereo". Research Report of CMP, 2003

Affine-invariant Descriptor

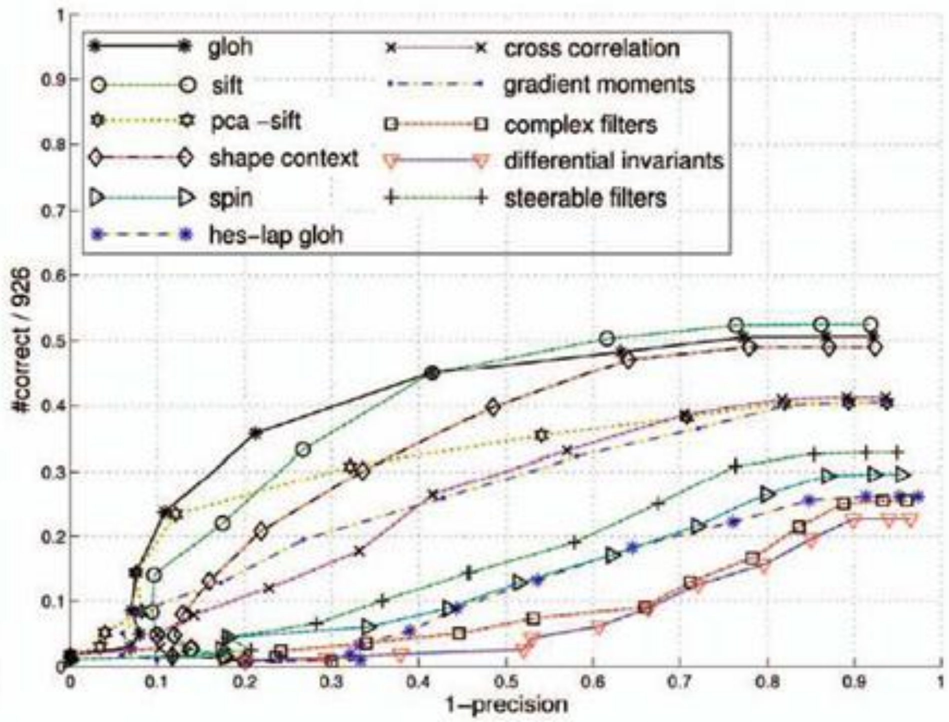
(Mikolajczyk and Schmid 04, IJCV) ‘s harris laplacian affine work use the similar idea.

They compute the gradient in the identified non-uniform scale region to normalize the overall gradient.

SIFT Descriptor – Scale-Invariant Feature Transform

- Empirically found[2] to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Rotation = 60^0



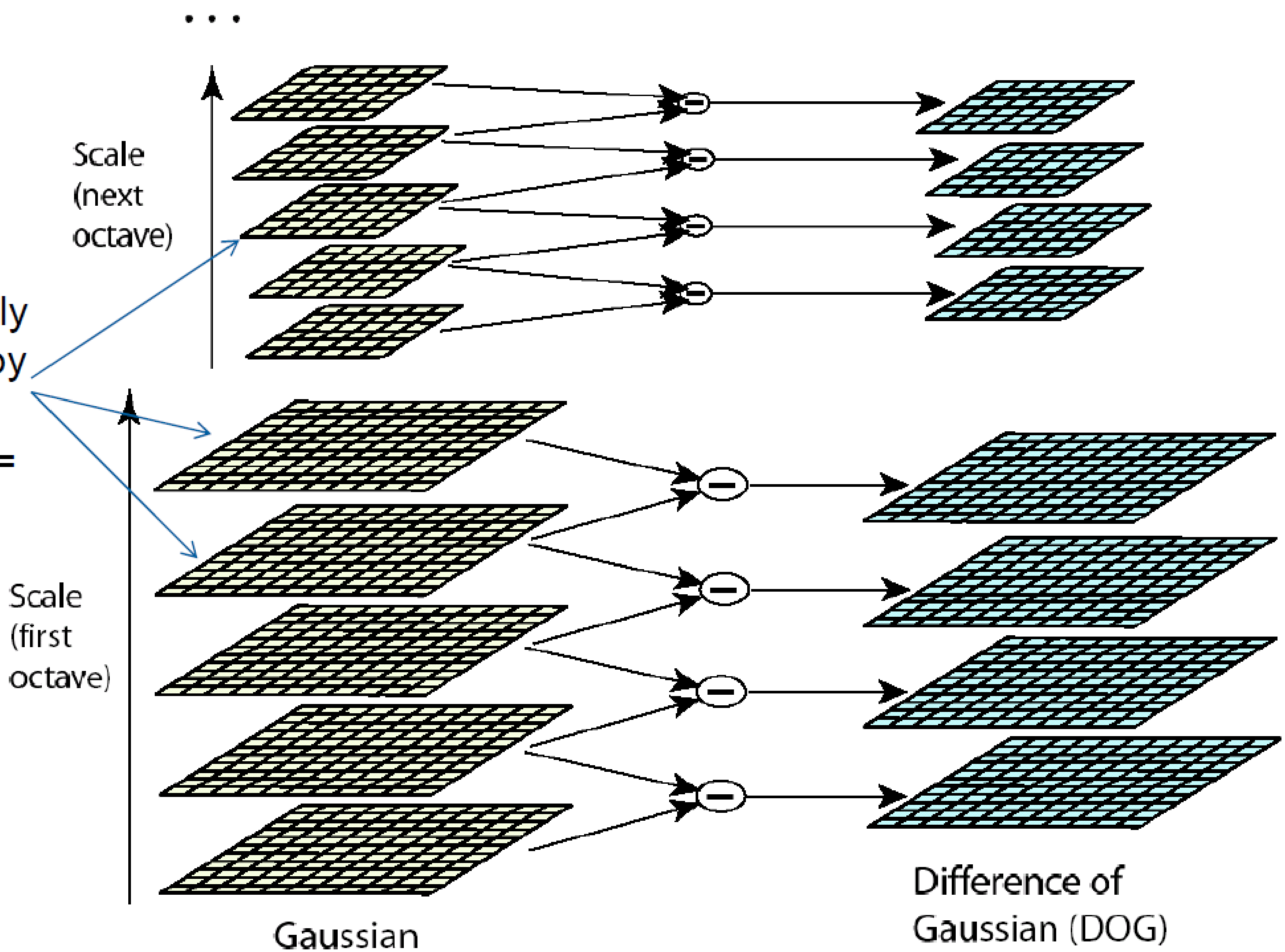
[1] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004
[2] K.Mikolajczyk, C.Schmid. "A Performance Evaluation of Local Descriptors". PAMI 2005

Distinguish Feature Detector and Descriptor

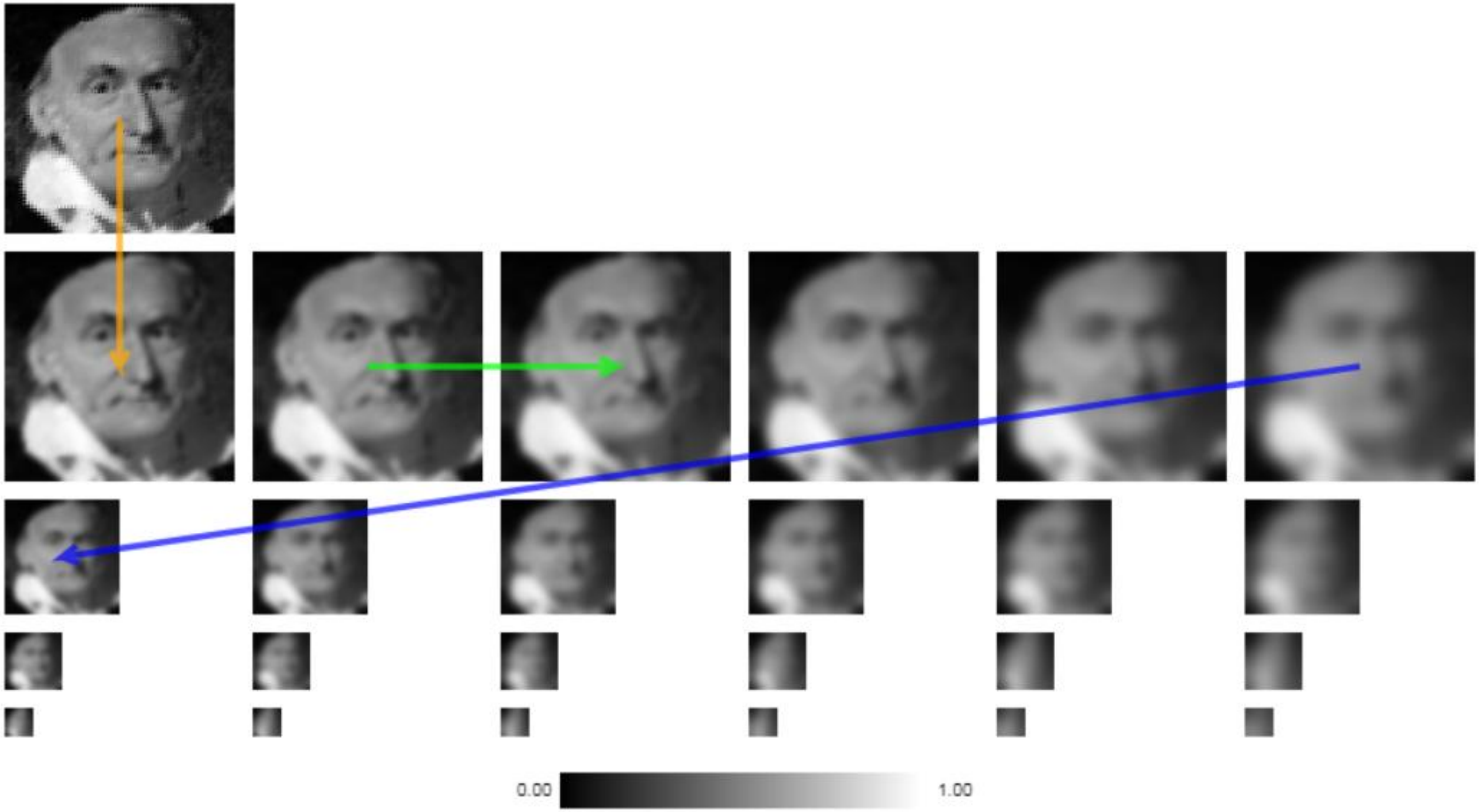
- Stable (repeatable) feature points can be detected regardless of image changes
  - Scale: search for correct scale as *maximum* of appropriate function
  - Affine: approximate regions with *ellipses* (this operation is affine invariant)
- Invariant and distinctive descriptors can be computed
  - Invariant *moments*
  - *Normalizing* with respect to scale and affine transformation

Scale Space – Process One Octave At a Time

Images incrementally convolved by Gaussian with sigma = 1.2



Scale Space – Process One Octave At a Time



Credit: http://weitz.de/sift/

Key Point Localization

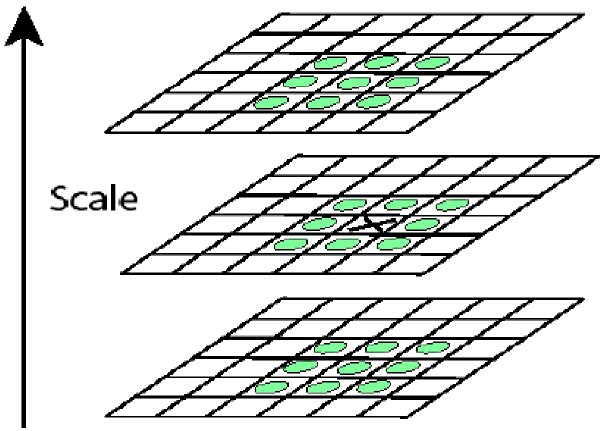
- Detect maxima and minima of difference-of-Gaussian in scale space
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)
- Taylor expansion around point:

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^\mathbf{T} \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

- Offset of extremum (use finite differences for derivatives):

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}$$

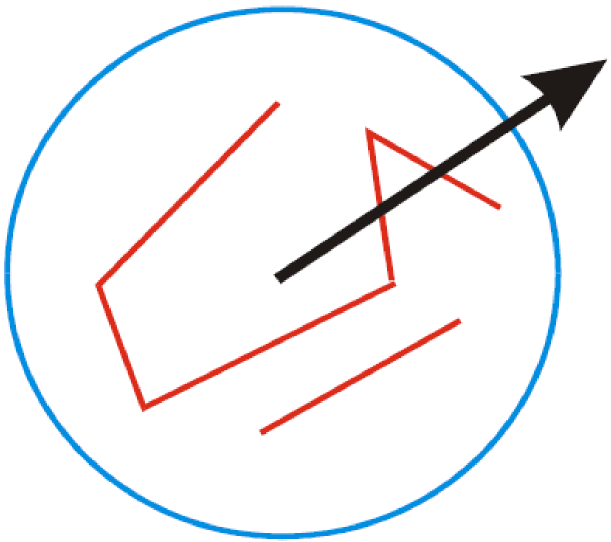
- Add $\hat{\mathbf{x}}$ back to the original detected position

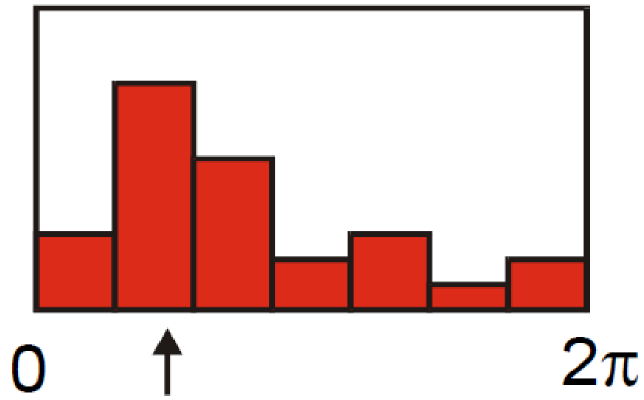


- Discard the feature point, if D(x) is too small.
- Discard the feature point, if it is an edge pt (similar to Harris Corner)

Select Canonical Orientation

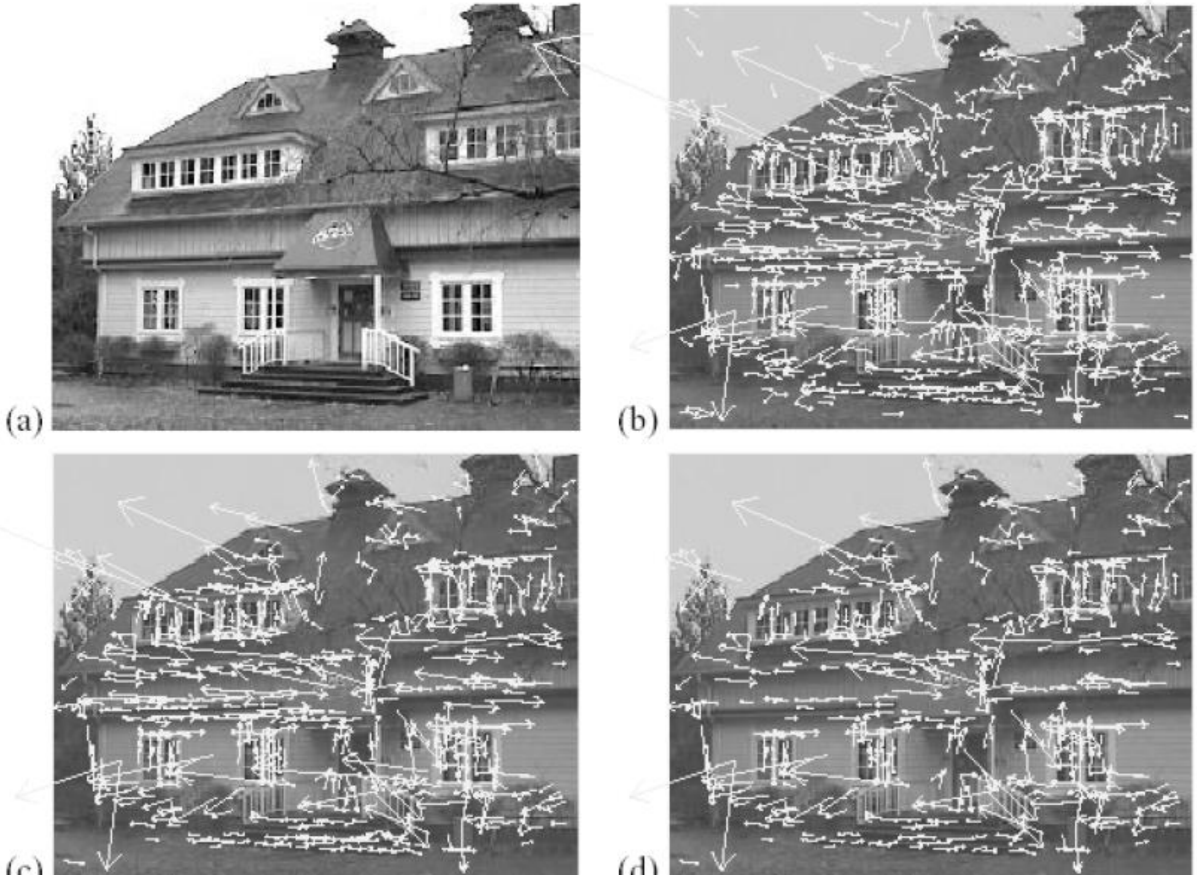
- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- If there are multiple peaks, create multiple features for each peak.
- Each key specifies stable 2D coordinates (x, y, scale, orientation)





Example of Key Point Refinement

Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach)



(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

SIFT Gradient Vector Computation

Each point with its region being rotated upright to achieve rotation invariant.

- Compute image gradients in the chosen region
- Weight the gradient magnitude by a Gaussian
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions

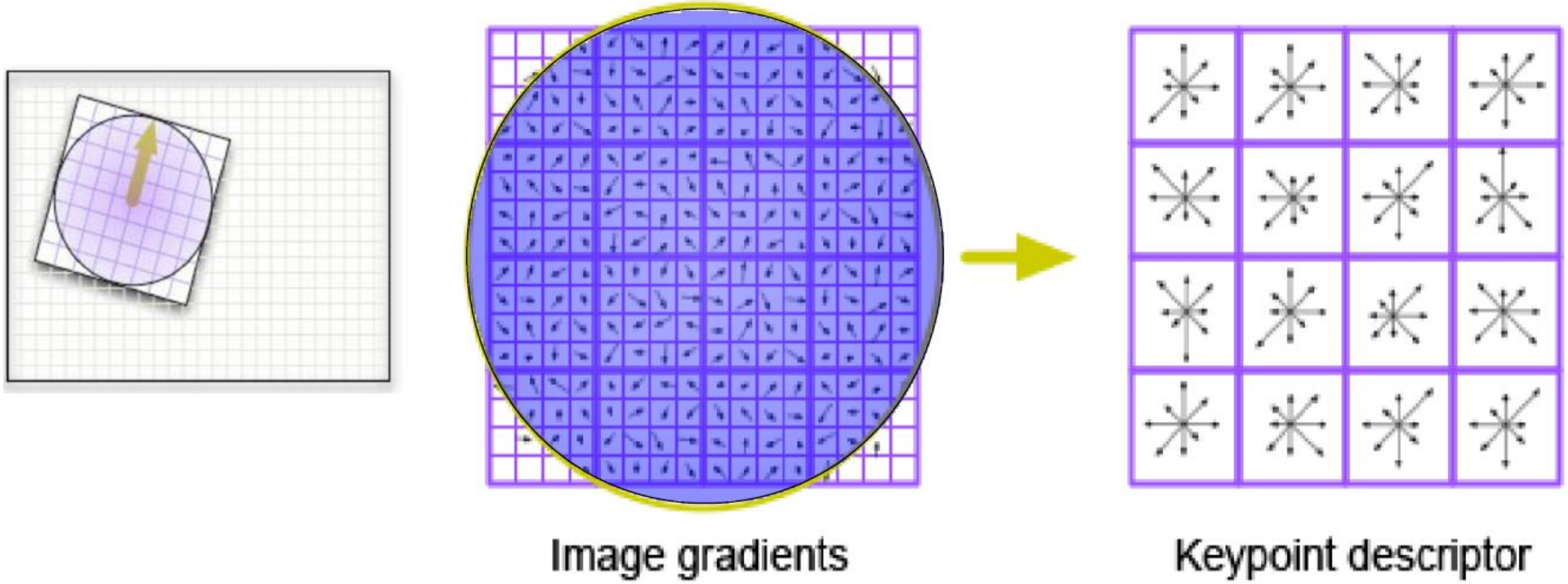
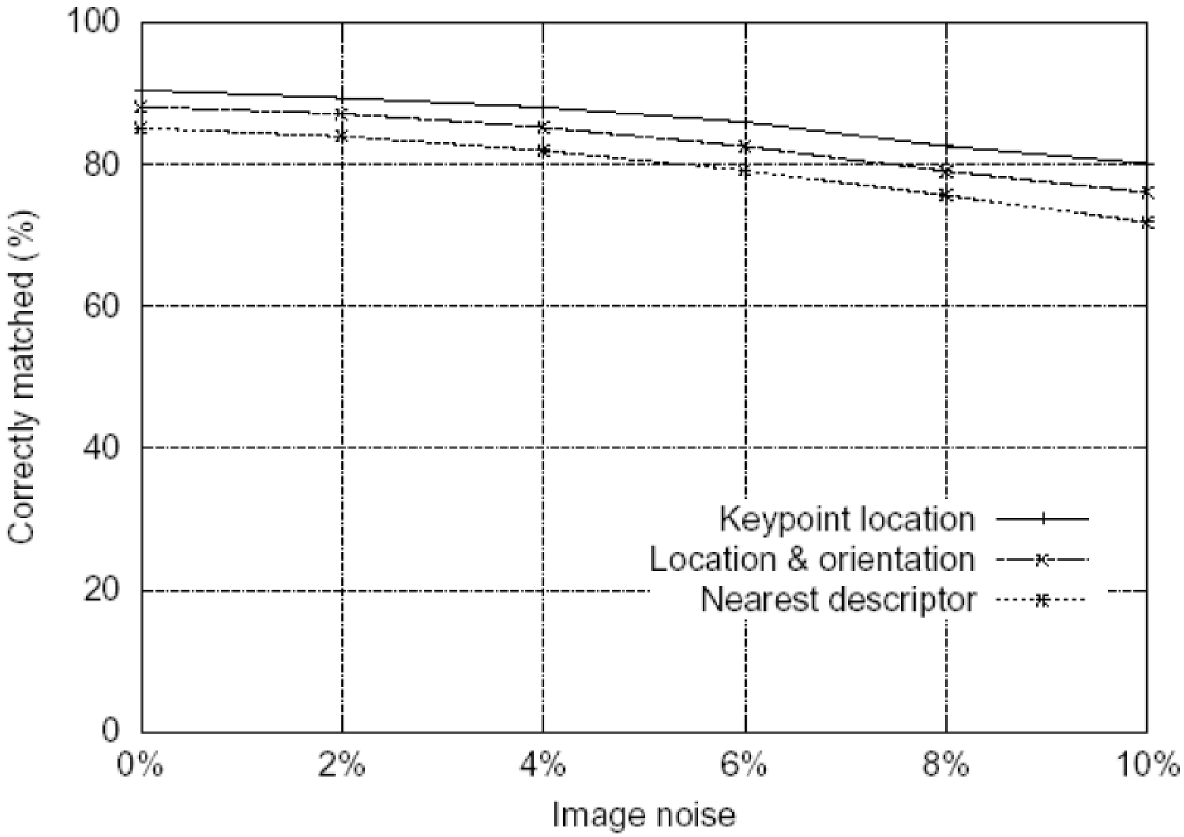


Image from: Jonas Hurreimann

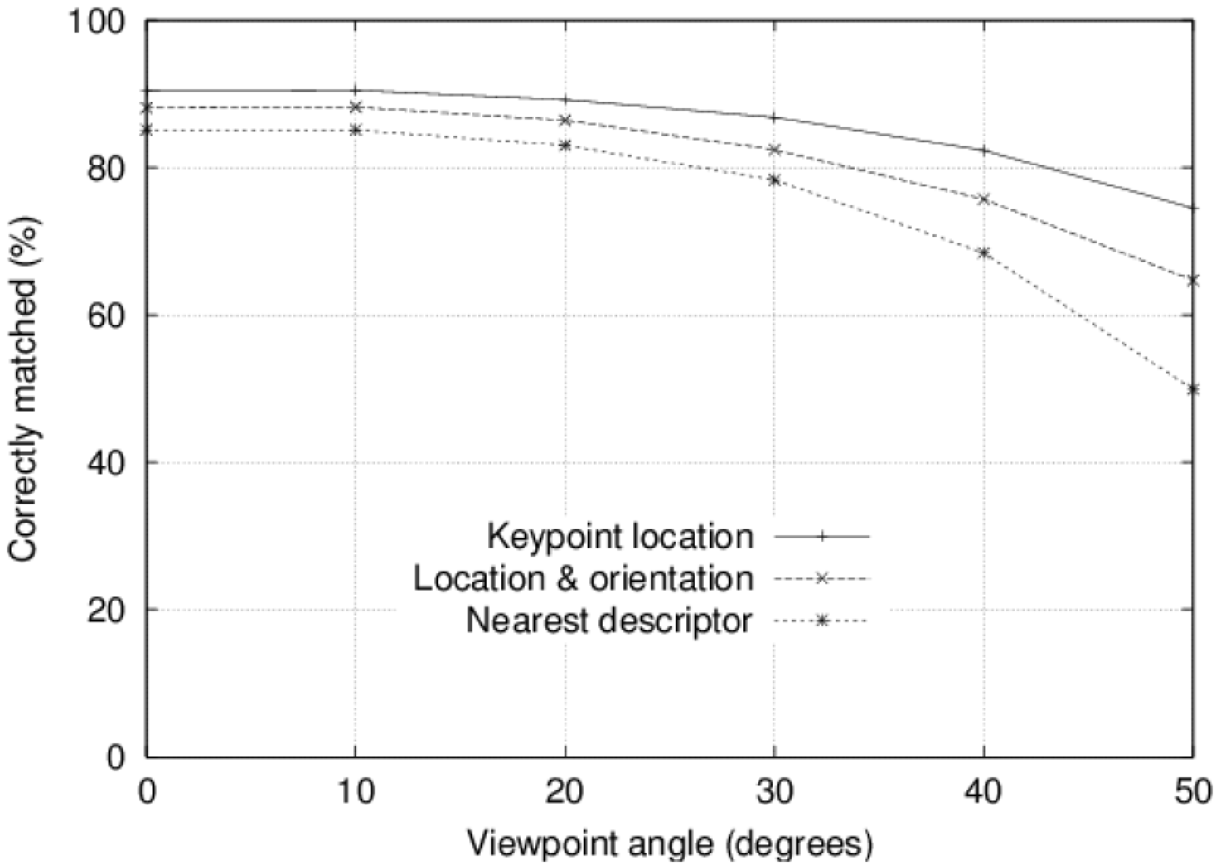
Feature Stability to Noise

- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features



Feature Stability to Affine Transformation

- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features



SIFT For Recognition

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affi ne transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

David Lowe, Recognition and Matching based on Local Invariant Features, 2003. CVPR Tutorial

SIFT For Recognition

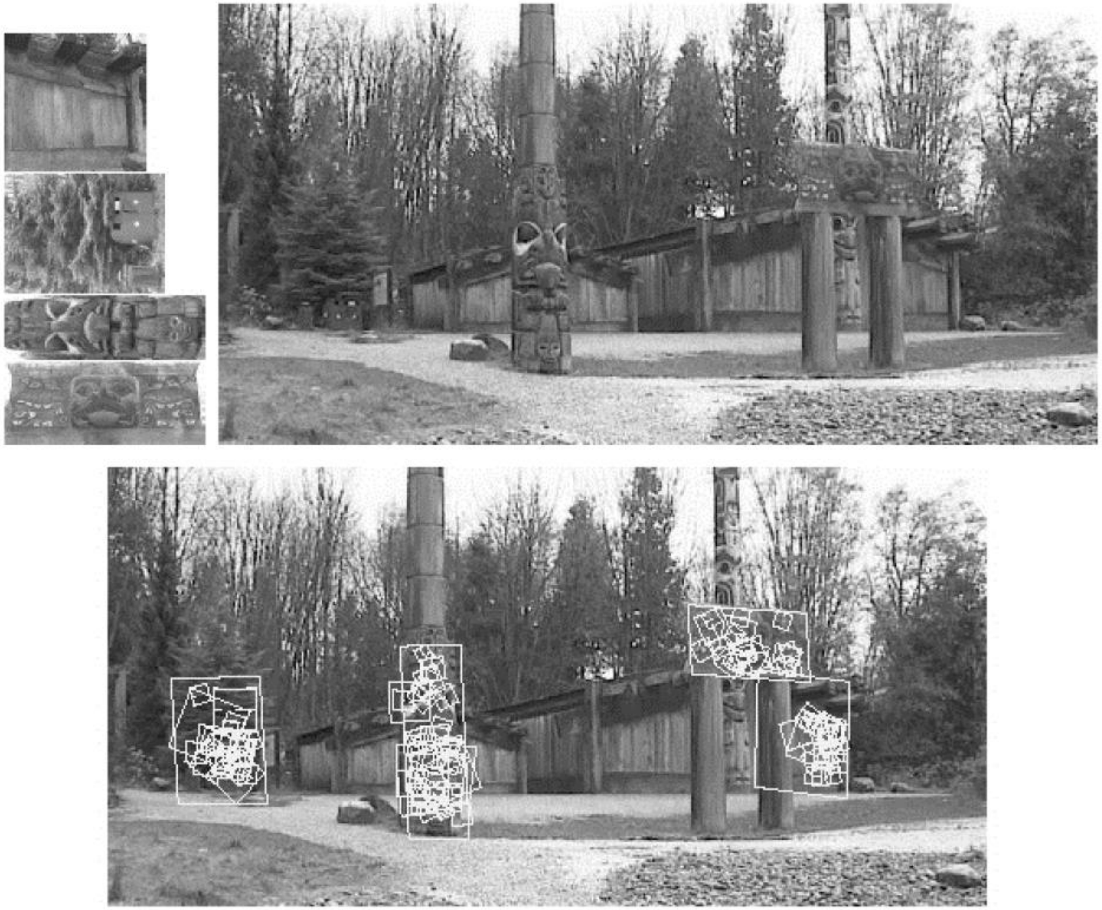


Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affi ne transform used for recognition.

David Lowe, Recognition and Matching based on Local Invariant Features, 2003. CVPR Tutorial

Speed up Robust Features

Bay et al, ECCV 2006

- Detector, Hessian Scale space

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}$$

Second order Gaussian derivatives are approximated with four box filter.

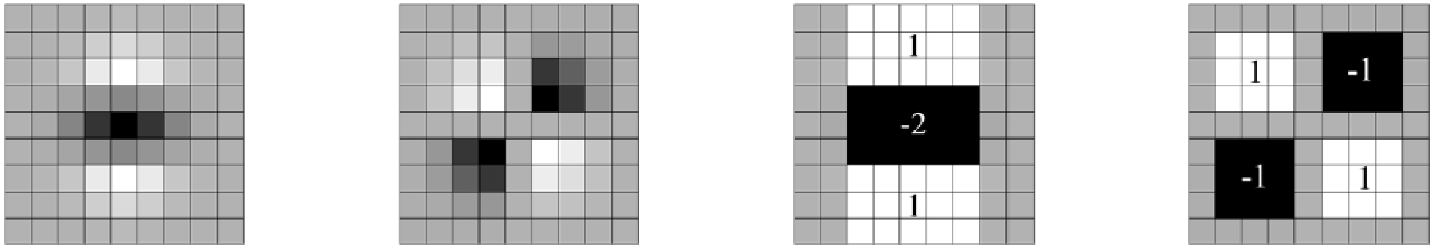
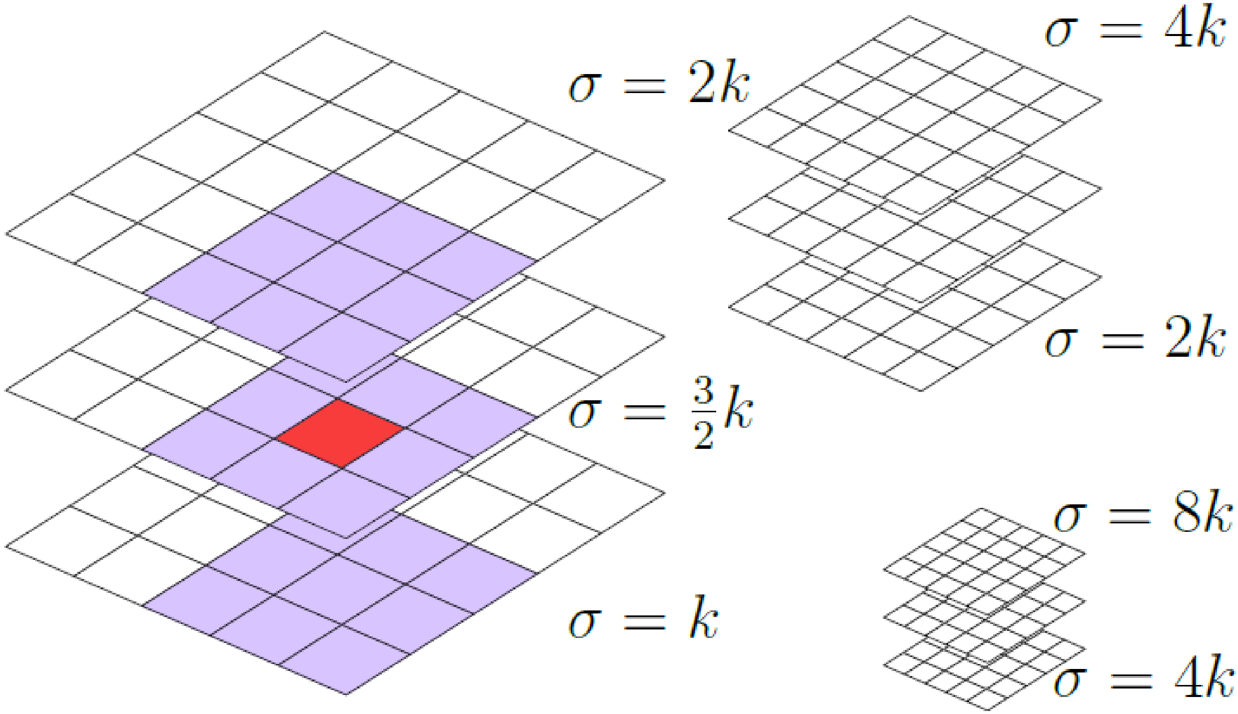


Fig. 1. Left to right: the (discretised and cropped) Gaussian second order partial derivatives in y-direction and xy-direction, and our approximations thereof using box filters. The grey regions are equal to zero.

Speed up Robust Features –Cont.

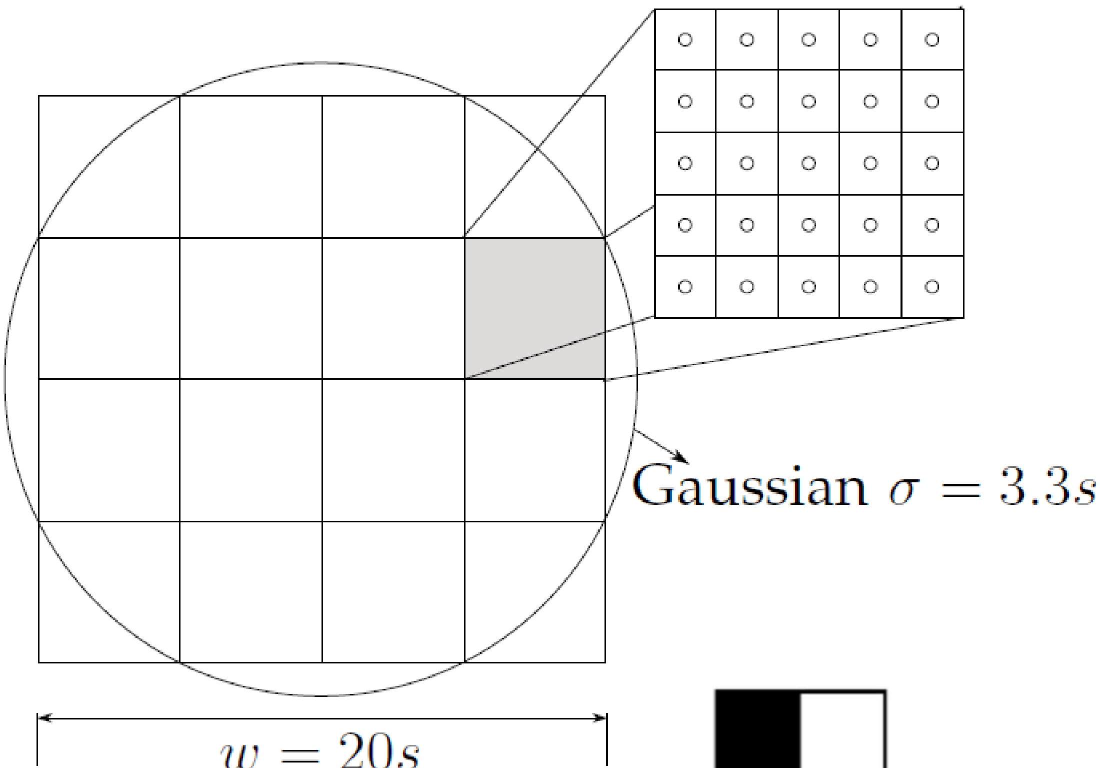


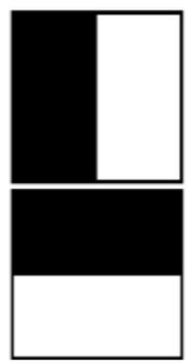
Less Octaves

Fig. 2. 3 octaves with 3 levels, The neighborhood for the $3 \times 3 \times 3$ non-maximum suppression used to detect features is highlighted.

(Pedersen, 2011)

Surf Descriptor





Harr kernel

Similar to SIFT
- Rotation to the dominant direction
- Sample region decomposition 4X4

Different from SIFT:
- Window size using 20 s (scale)
- Instead of using gradient, surf use Harr wavelet decomposition on in x and y direction, each cell with four dimensional vectors

$$\mathbf{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right).$$

Therefore in total 4 X 4 X 4 = 64 d Feature vector

Least Square Image Matching

- The descriptor-based matching matches the "signatures" to identify matches, this does not modify the locations the matches, i.e. if the detected points is not accurate in position due to image noise, then the matches may produce errors in geometric reconstruction.
- Idea: use the images itself to readjust the position of the matches to get the optimal correlation.

Least Square Image Matching (LSM)

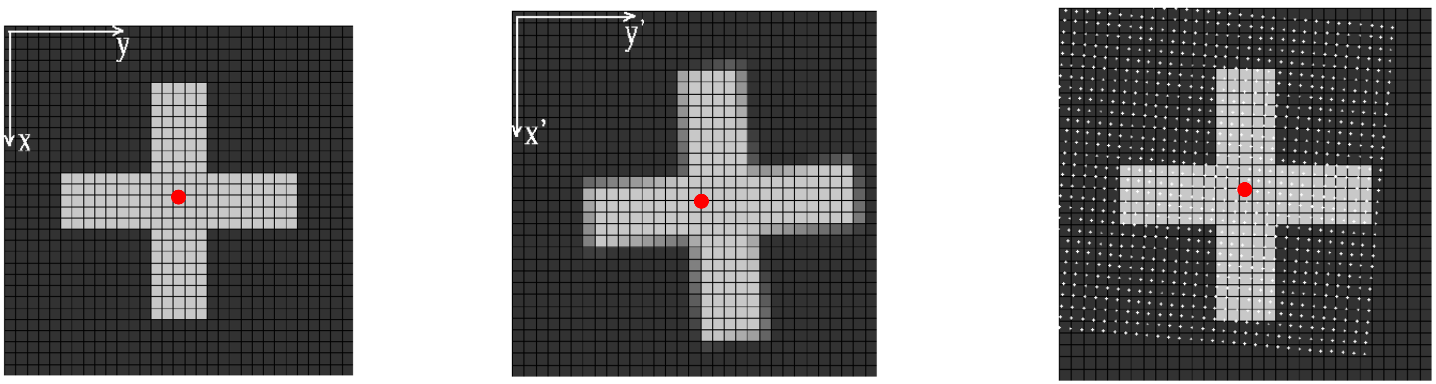
- Old technique but very useful for determine highly accurate point match – up 1/10 – 1/50 pixel accuracy

  (Förstner, W., 1982) (Gruen, 1985)

- Different from descriptor-based method, LSM will adjust the position of the points.

Least Square Image Matching (LSM)

Idea: Take two images patches centered at potential point matches, to minimize the squared errors of per-pixel difference under a geometrically affine and radiometrically additive & multiplicative transformation.



$g(x,y)$ $h(x,y)$ $h(x',y')$

$$x' = a_1x + a_2y + a_3$$
$$y' = b_1x + b_2y + b_3$$

Affine geometric

$$g(x,y) = k_1 h(x',y') + k_2$$

multiplicative additive

LSM – Cont.

- Minimization problem:

$F(x,y) = g(x,y) - k_1 h(x',y') - k_2 = 0,$ for every pixel in the patch

$$F(x,y) = g(x,y) - k_1 h(a_1 x + a_2 y + a_3, b_1 x + b_2 y + b_3) - k_2 = 0 \quad (1)$$

We have many pixels M rows, N cols, to calculate

$$\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{k}} E(\boldsymbol{a},\boldsymbol{b},\boldsymbol{k}) = \sum_{x,y}^{M,N} |F(x,y)|^2 \quad (2)$$

With $g(x,y)$ and $h(x,y)$ being non-linear, first order Taylor expansion of (1), do iterative solution:

$$F(x,y,\boldsymbol{B}) \approx F(x,y,\boldsymbol{B}^0) + \boldsymbol{\mathcal{J}}\left(F(x,y,\boldsymbol{B}^0)\right)^T \cdot \Delta\boldsymbol{B} = 0$$

$$\boldsymbol{\mathcal{J}}(F(x,y,\boldsymbol{B}^0)) = \left[\frac{\partial F}{\partial a_1}, \frac{\partial F}{\partial a_2}, \frac{\partial F}{\partial a_3}, \frac{\partial F}{\partial b_1}, \frac{\partial F}{\partial b_2}, \frac{\partial F}{\partial b_3}, \frac{\partial F}{\partial k_1}, \frac{\partial F}{\partial k_2}\right]_{|\boldsymbol{B}^0}^T$$

$$\Delta\boldsymbol{B} = [\Delta a_1, \Delta a_2, \Delta a_2, \Delta a_3, \Delta b_1, \Delta b_2, \Delta b_3, \Delta k_1, \Delta k_2]^{\boldsymbol{T}}$$

LSM – Cont.

e.g.

$$\frac{\partial F}{\partial a_1} = 0 - \frac{\partial h(x',y')}{\partial x}\frac{\partial(a_1x + a_2y + a_3)}{\partial a_1}$$
$$= -h_x(x',y')x$$

$$\frac{\partial F}{\partial k_1} = -h_x(x',y')$$

Therefore

$$\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{k}} E(\boldsymbol{a},\boldsymbol{b},\boldsymbol{k}) = \sum_{x,y}^{M,N} \left|F(x,y,\boldsymbol{B}^0) + \boldsymbol{\mathcal{J}}\left(F(x,y,\boldsymbol{B}^0)\right)^T \cdot \Delta\boldsymbol{B}\right|^2 \quad (3)$$

Now this linear

LSM – Cont.

- Given $\boldsymbol{B}^0 = [1,0,0,0,1,0,1,0]^T$, Compute $\Delta\boldsymbol{B}$ through (3) to get $\boldsymbol{B}^1 = \boldsymbol{B}^0 + \Delta\boldsymbol{B}$ until converge.

With respect to $\Delta\boldsymbol{B}$, equation (3) is a quadratic function, solution:

$$\Delta\boldsymbol{B} = \left[\sum_{x,y}^{M,N} \mathcal{J}\left(F(x,y,\boldsymbol{B}^0)\right) \mathcal{J}\left(F(x,y,\boldsymbol{B}^0)\right)^T\right]^{-1}\left[-\sum_{x,y}^{M,N} F(x,y,\boldsymbol{B}^0)\mathcal{J}\left(F(x,y,\boldsymbol{B}^0)\right)\right]$$

Approximate Hessian Matrix

Questions?