Feature matching

Digital Visual Effects, Spring 2005

Yung-Yu Chuang

2005/3/16

with slides by Trevor Darrell Cordelia Schmid, David Lowe, Darya Frolova, Denis Simakov, Robert Collins and Jiwon Kim

Announcements

- Project #1 is online, you have to write a program, not just using available software.
- Send me the members of your team.
- Sign up for scribe at the forum.

Blender

http://www.blender3d.com/cms/Home.2.0.html

Blender could be used for your project #3 matchmove.

In the forum

- Barycentric coordinate
- RBF

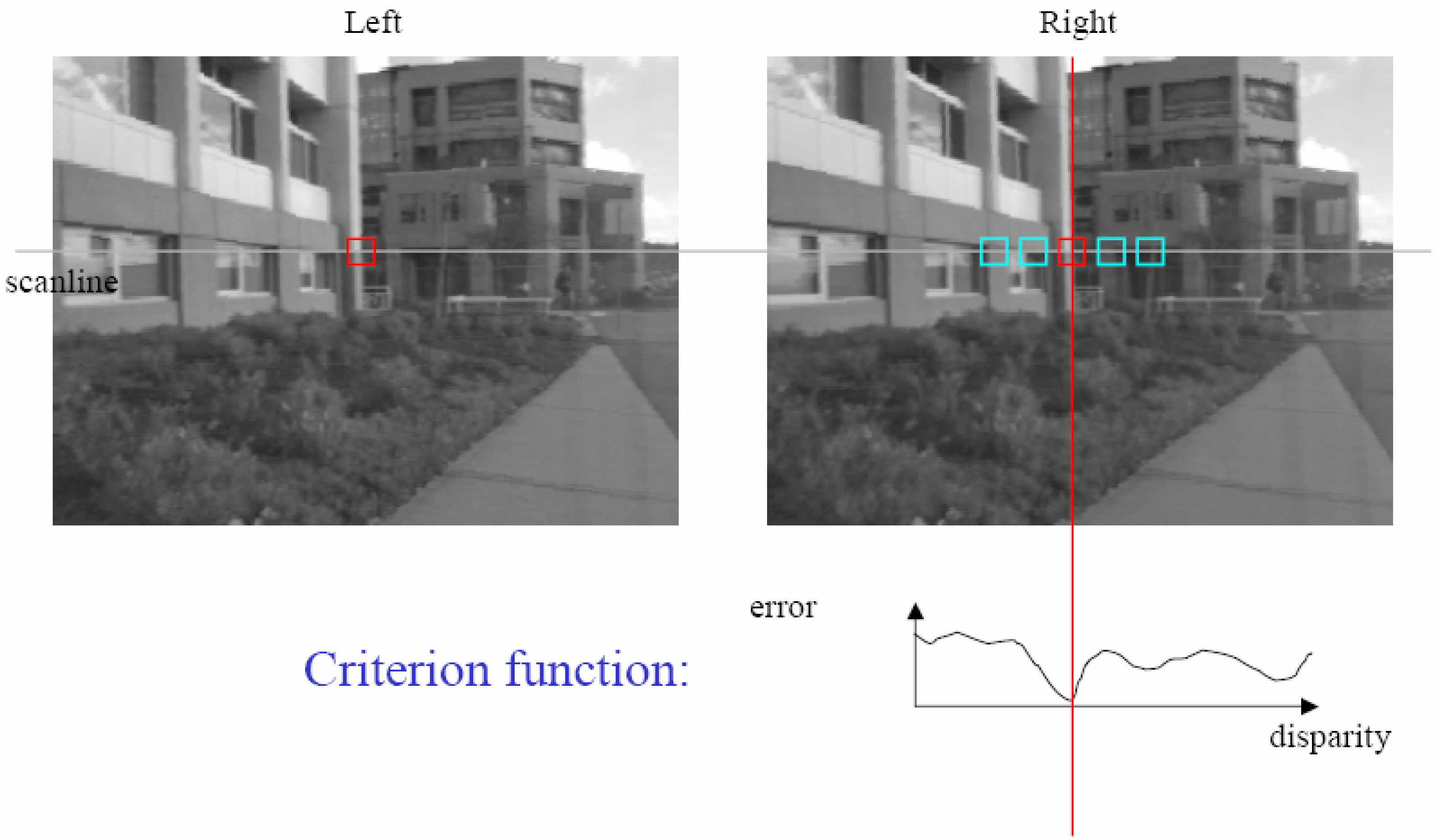
Outline

- Block matching
- Features
- Harris corner detector
- SIFT
- SIFT extensions
- Applications

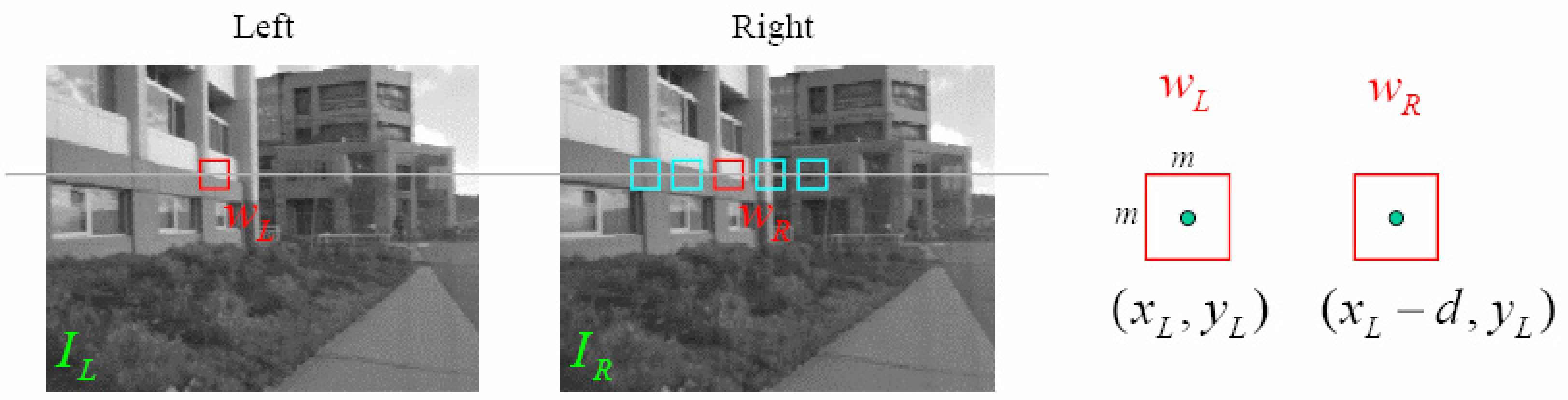
Correspondence by block matching

- Points are individually ambiguous
- More unique matches are possible with small regions of images


Correspondence by block matching



Sum of squared distance



w_L and w_R are corresponding m by m windows of pixels.

We define the window function :

$$W_m(x, y) = \{u, v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x, y, d) = \sum_{(u,v) \in W_m(x,y)} [I_L(u, v) - I_R(u - d, v)]^2$$

Image blocks as a vector

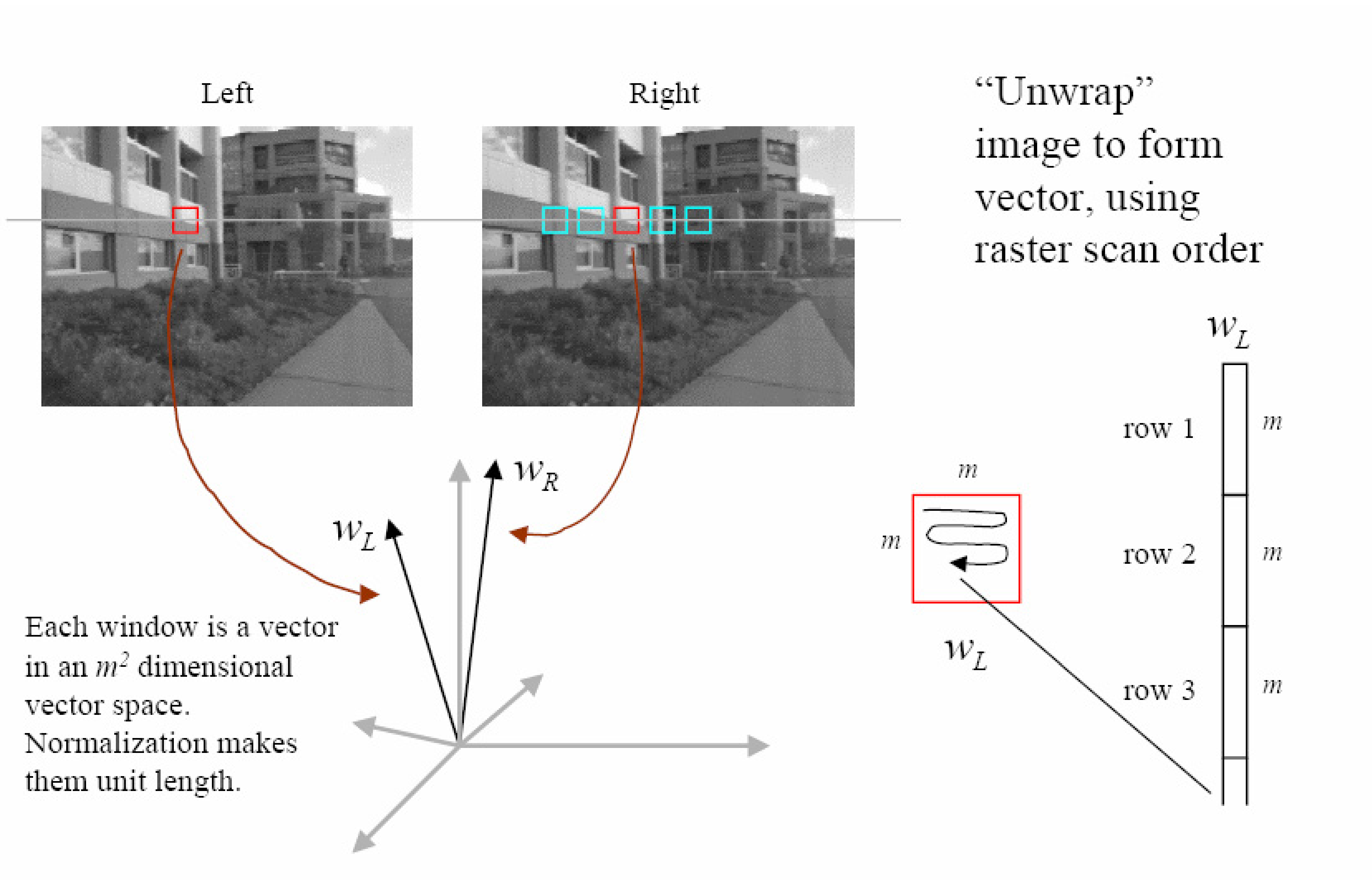
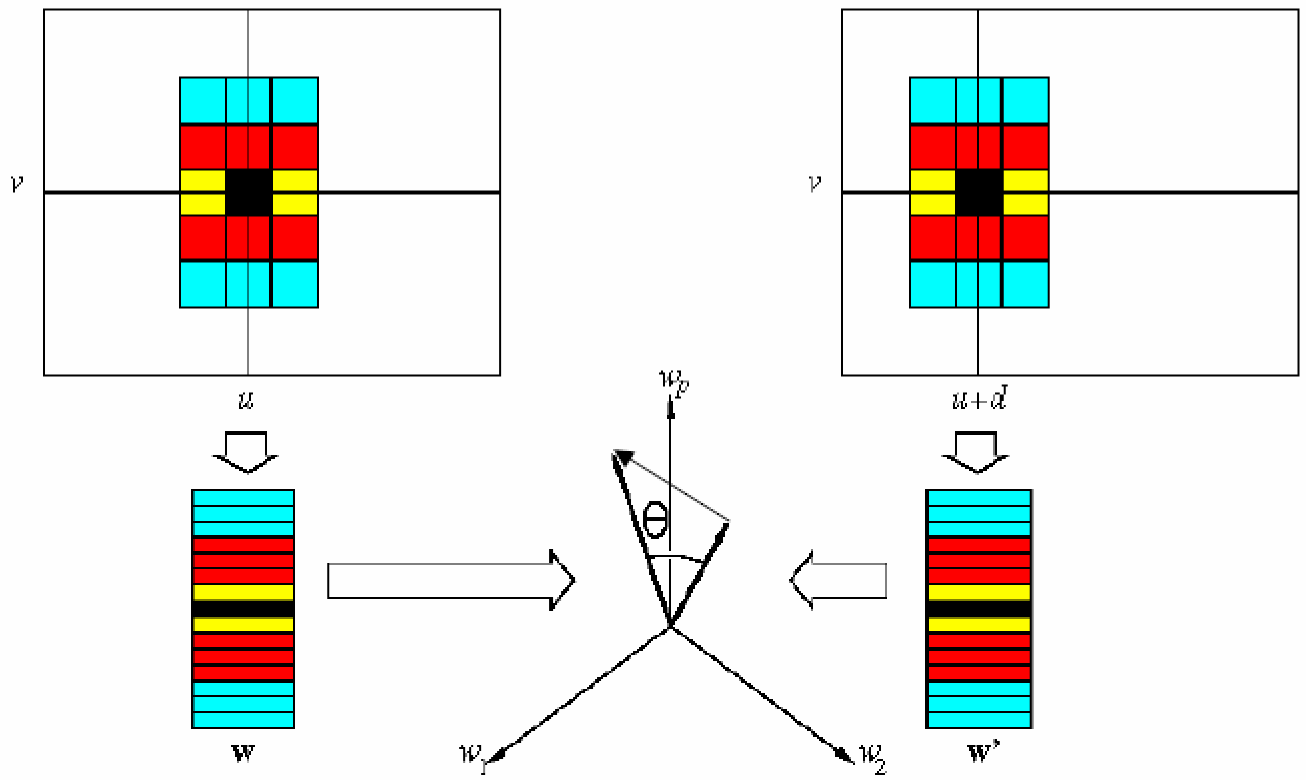
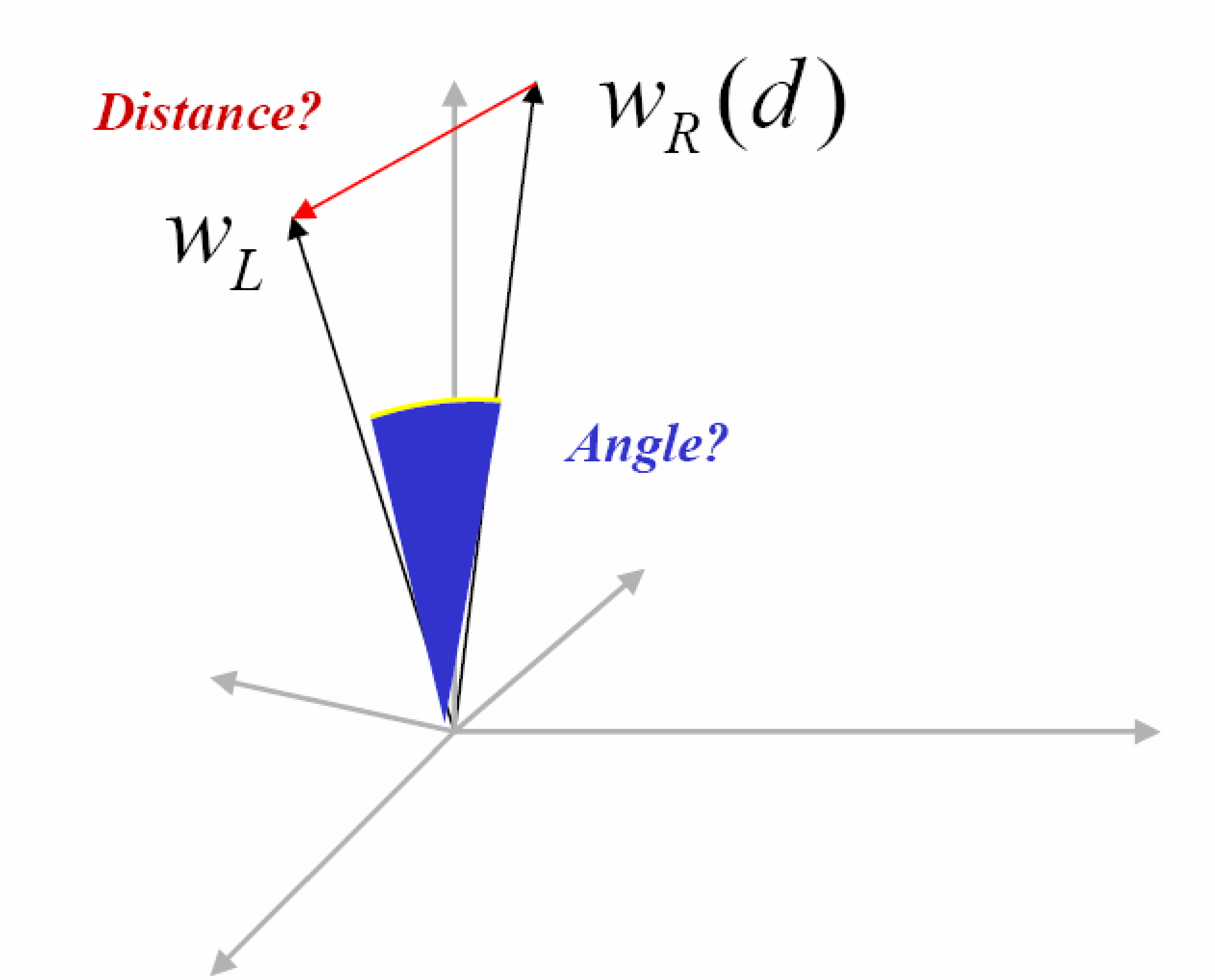


Image blocks as a vector



Matching metrics



Features

- Properties of features
- Detector: locates feature
- Descriptor and matching metrics: describes and matches features

- In the example for block matching:
  - Detector: none
  - Descriptor: block
  - Matching: distance

Desired properties for features

DigiVFX

- Invariant: invariant to scale, rotation, affine, illumination and noise for robust matching across a substantial range of affine distortion, viewpoint change and so on.
- Distinctive: a single feature can be correctly matched with high probability

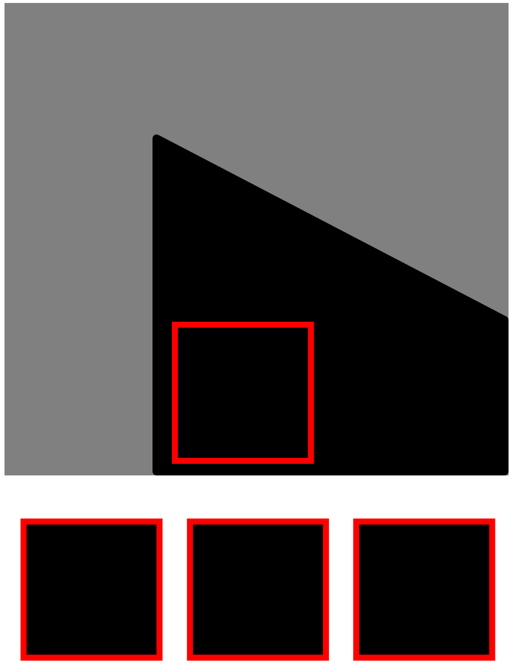
Harris corner detector

Moravec corner detector (1980)

DigiVFX

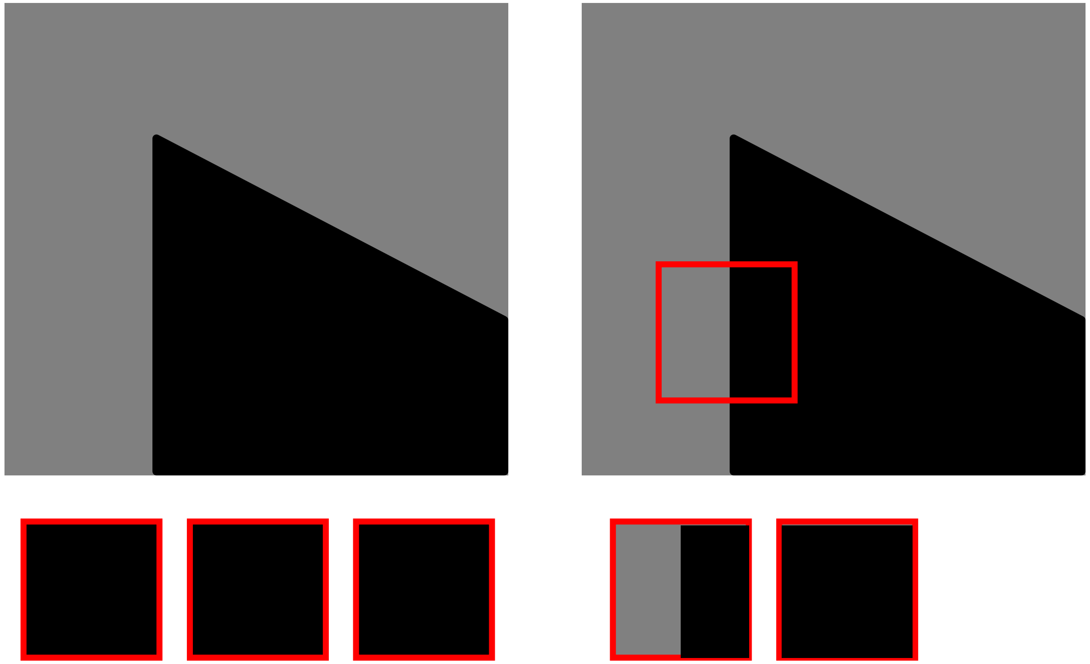
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity

Moravec corner detector



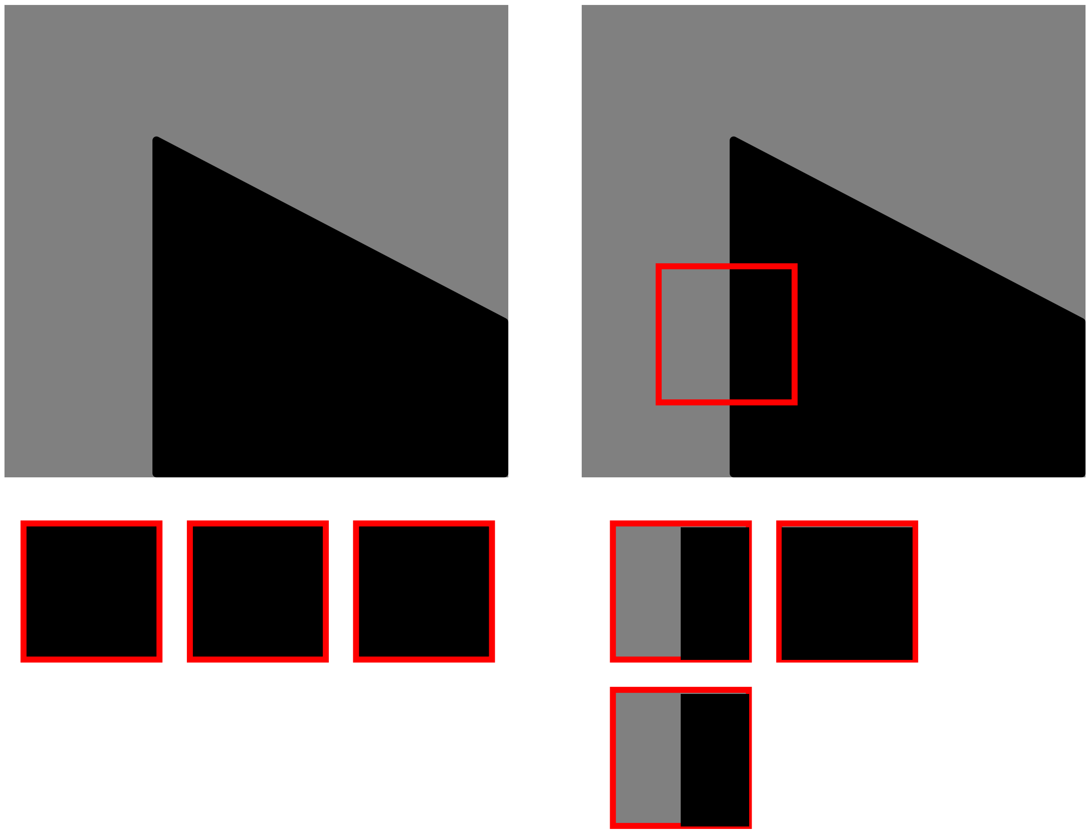
flat

Moravec corner detector



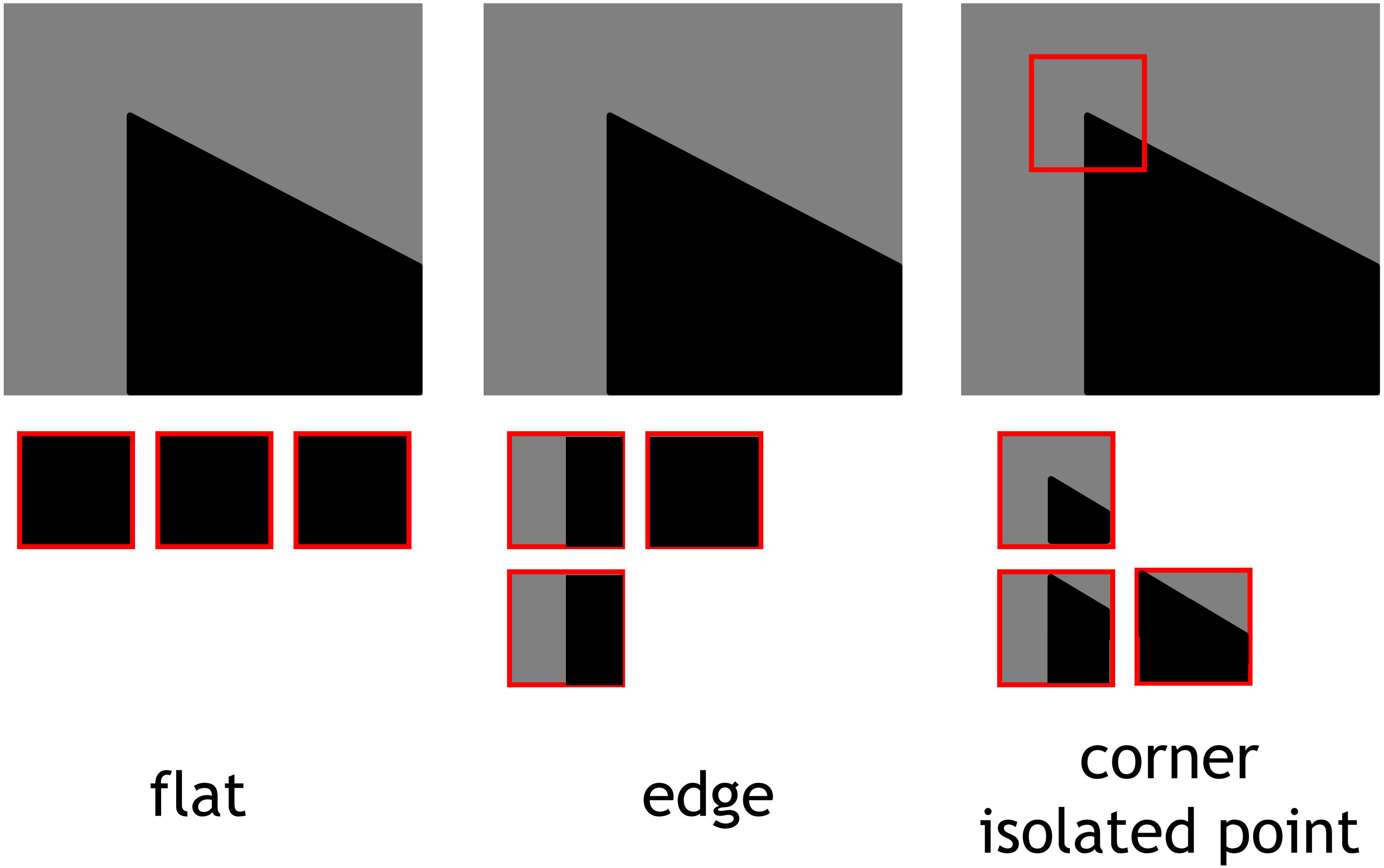
flat

Moravec corner detector



flat edge

Moravec corner detector

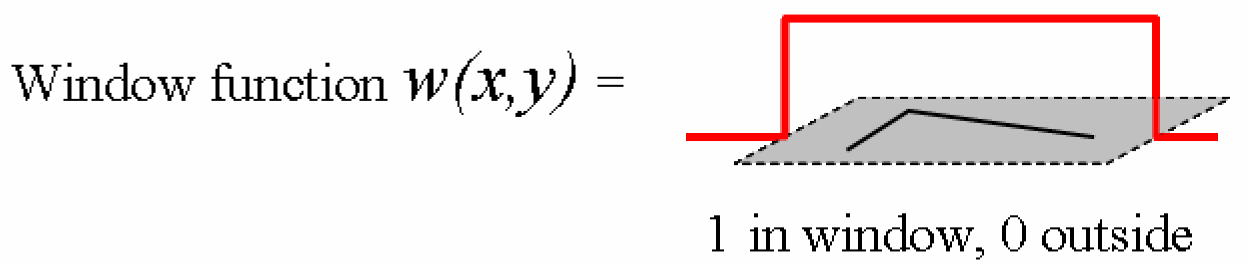


Moravec corner detector

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



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

Look for local maxima in *min{E}*

Problems of Moravec detector

- Noisy response due to a binary window function
- Only a set of shifts at every 45 degree is considered
- Responds too strong for edges because only minimum of E is taken into account

⇨ Harris corner detector (1988) solves these problems.

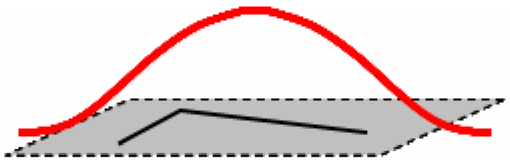
Harris corner detector

Noisy response due to a binary window function

- ➢ Use a Gaussian function

$$w(x,y) = \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

Window function $w(x,y)$ =



Gaussian

Harris corner detector

Only a set of shifts at every 45 degree is considered

- Consider all small shifts by Taylor's expansion

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

$$= \sum_{x,y} w(x,y)\left[I_x u + I_y v + O(u^2,v^2)\right]^2$$

$$E(u,v) = Au^2 + 2Cuv + Bv^2$$

$$A = \sum_{x,y} w(x,y) I_x^2(x,y)$$

$$B = \sum_{x,y} w(x,y) I_y^2(x,y)$$

$$C = \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y)$$

Harris corner detector

Equivalently, 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:

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

Harris corner detector

Responds too strong for edges because only minimum of E is taken into account

- ➢A new corner measurement

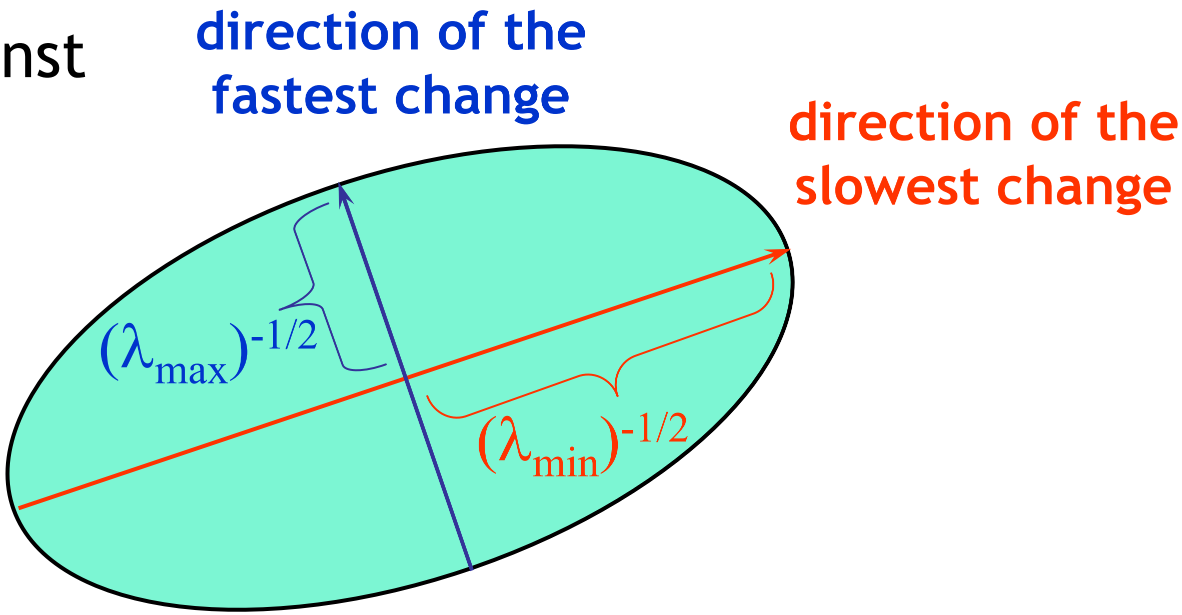
Harris corner detector

Intensity change in shifting window: eigenvalue analysis

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

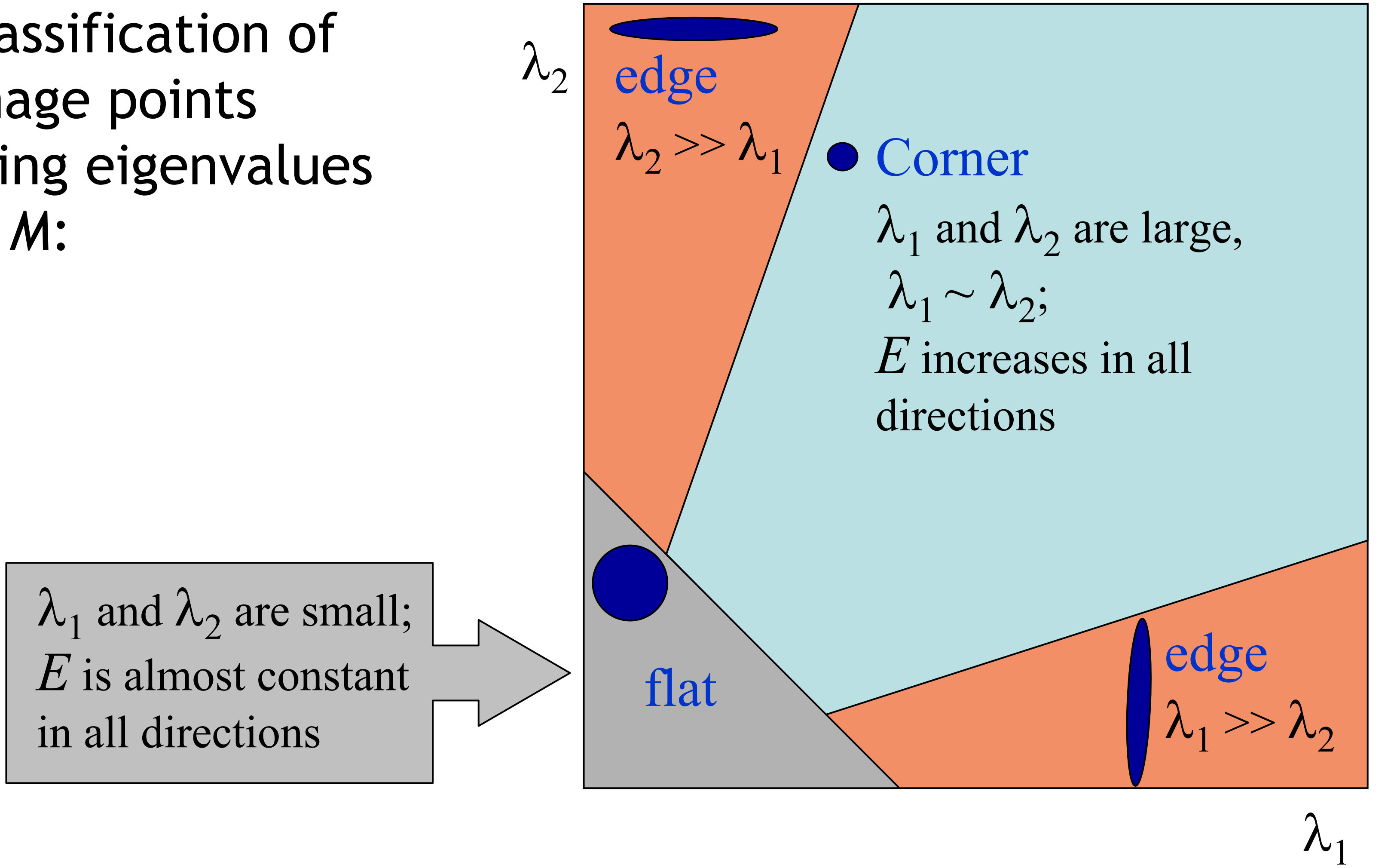
λ_1, λ_2 – eigenvalues of M

Ellipse *E(u,v)* = const



Harris corner detector

Classification of image points using eigenvalues of *M*:



Harris corner detector

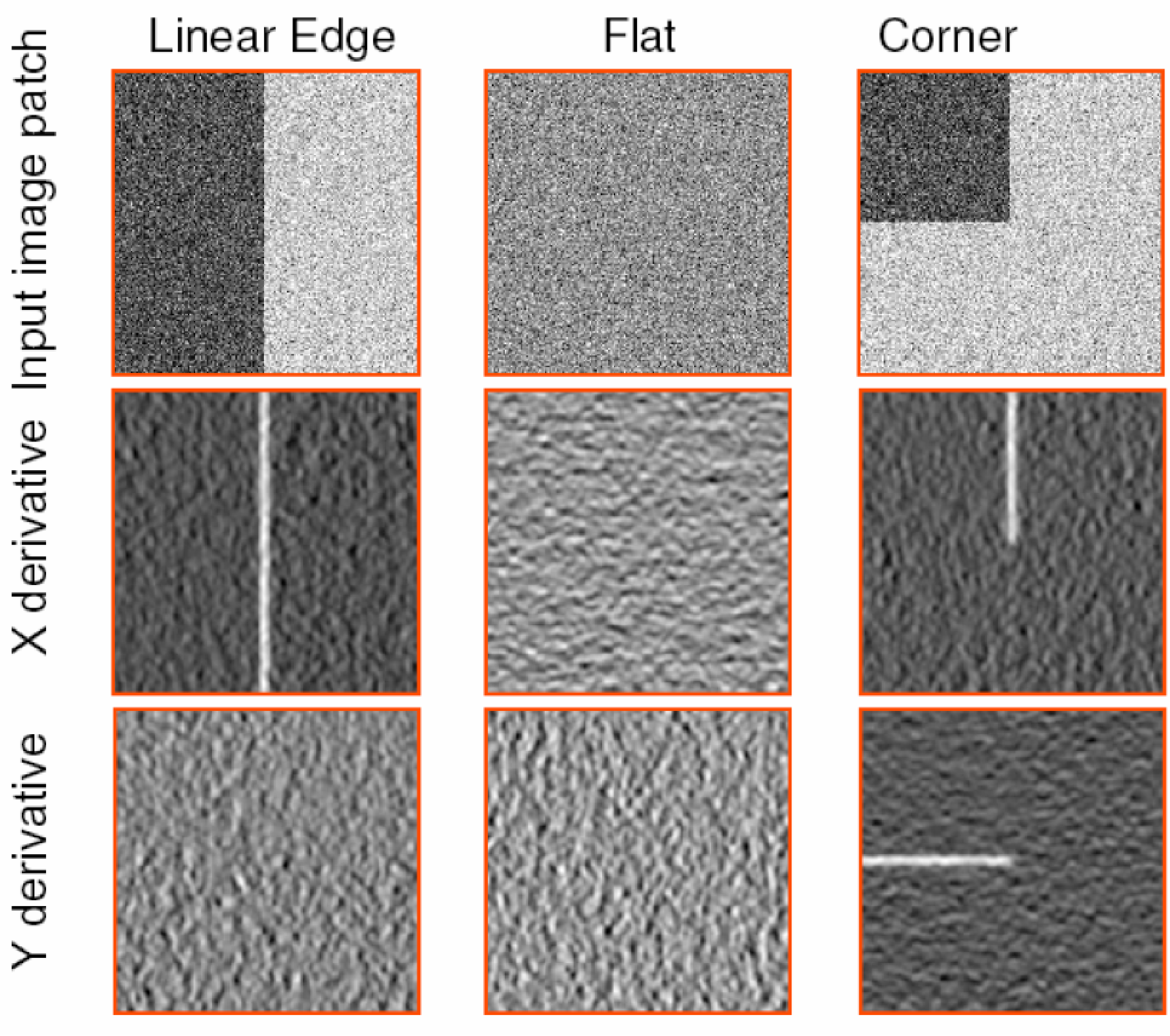
Measure of corner response:

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

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

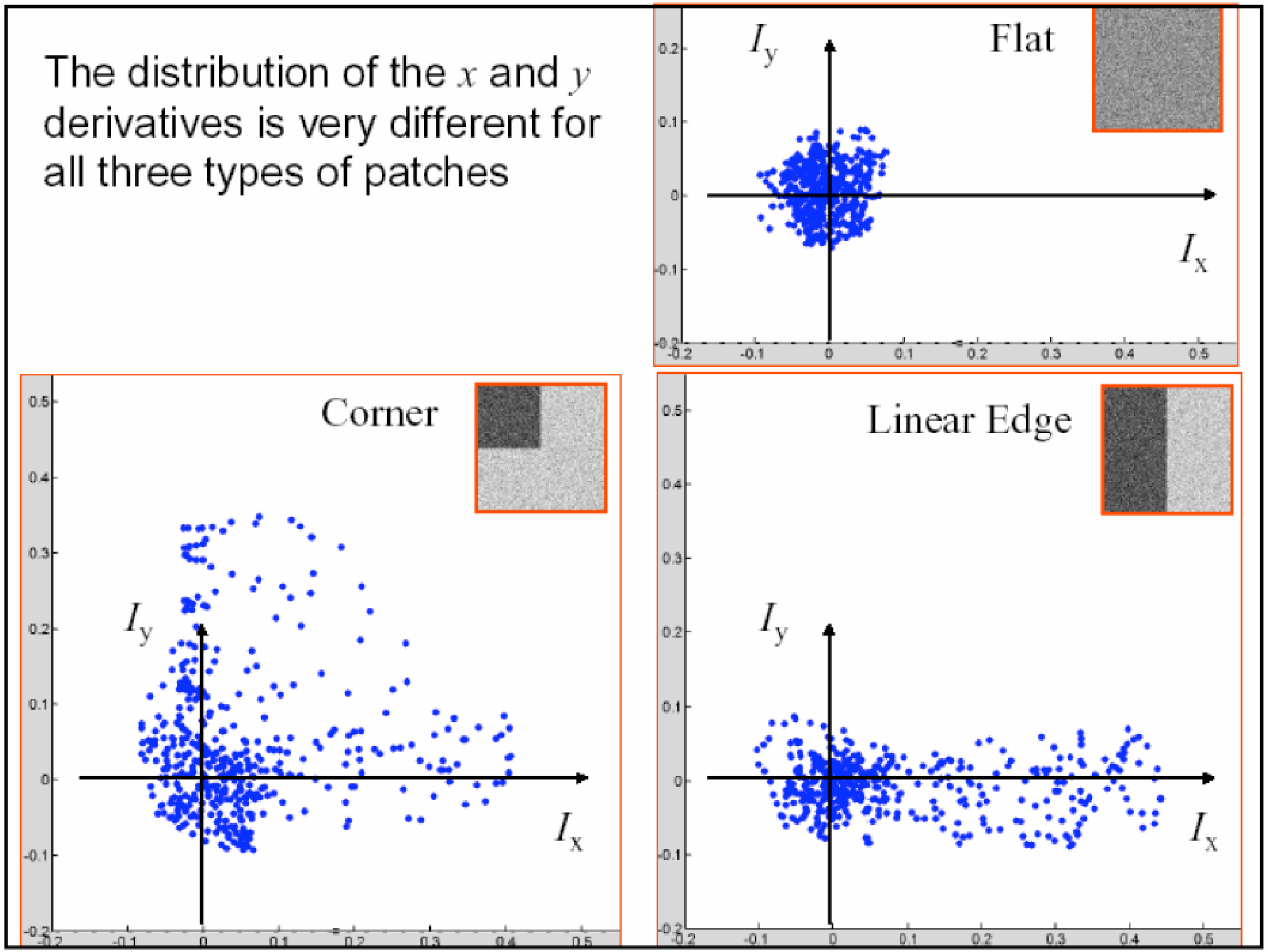
(*k* – empirical constant, *k* = 0.04-0.06)

Another view

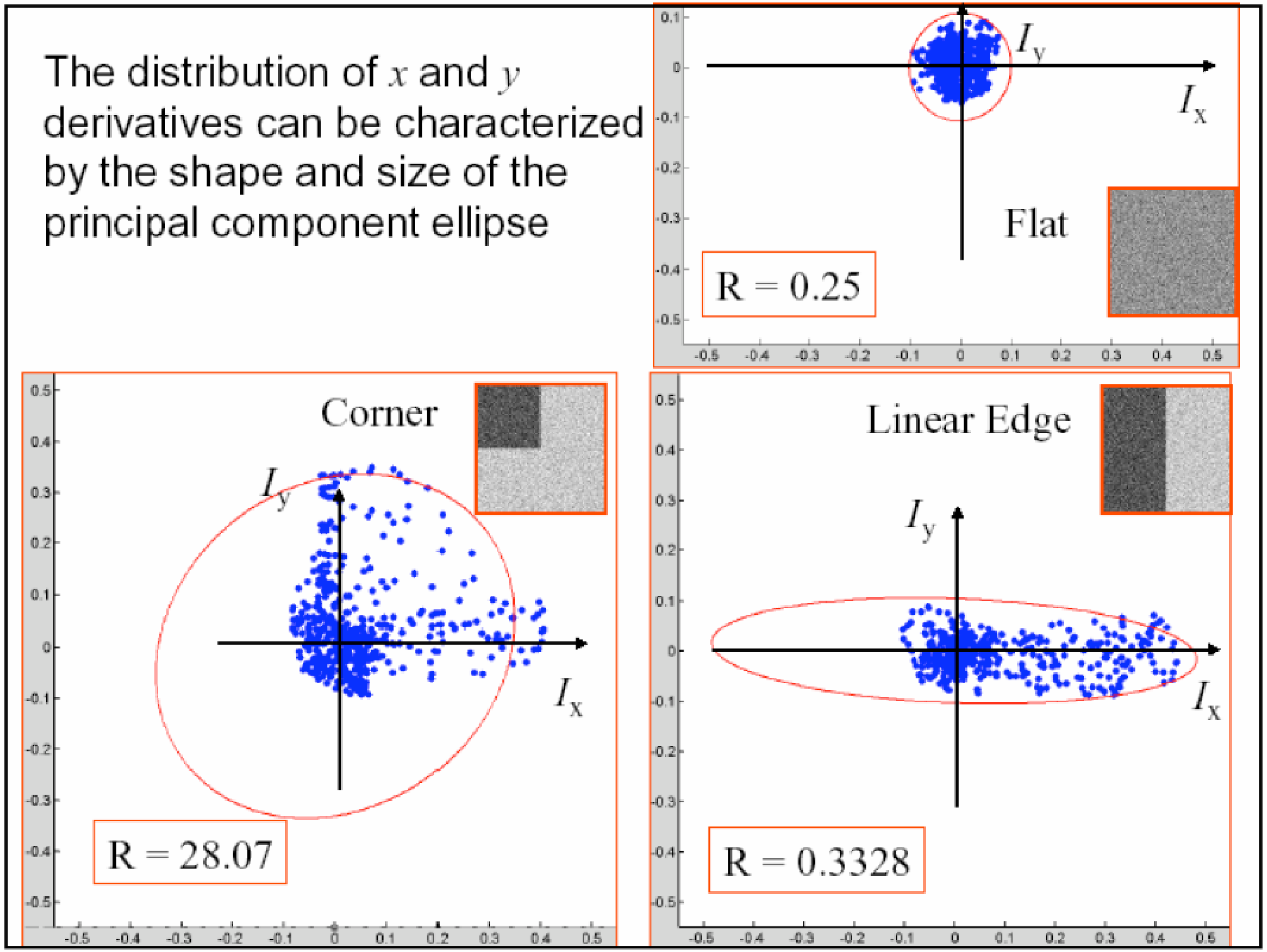


Another view

The distribution of the x and y derivatives is very different for all three types of patches



Another view



Summary of Harris detector

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x . I_x \quad I_{y2} = I_y . I_y \quad I_{xy} = I_x . I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_{\sigma'} * I_{x2} \quad S_{y2} = G_{\sigma'} * I_{y2} \quad S_{xy} = G_{\sigma'} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

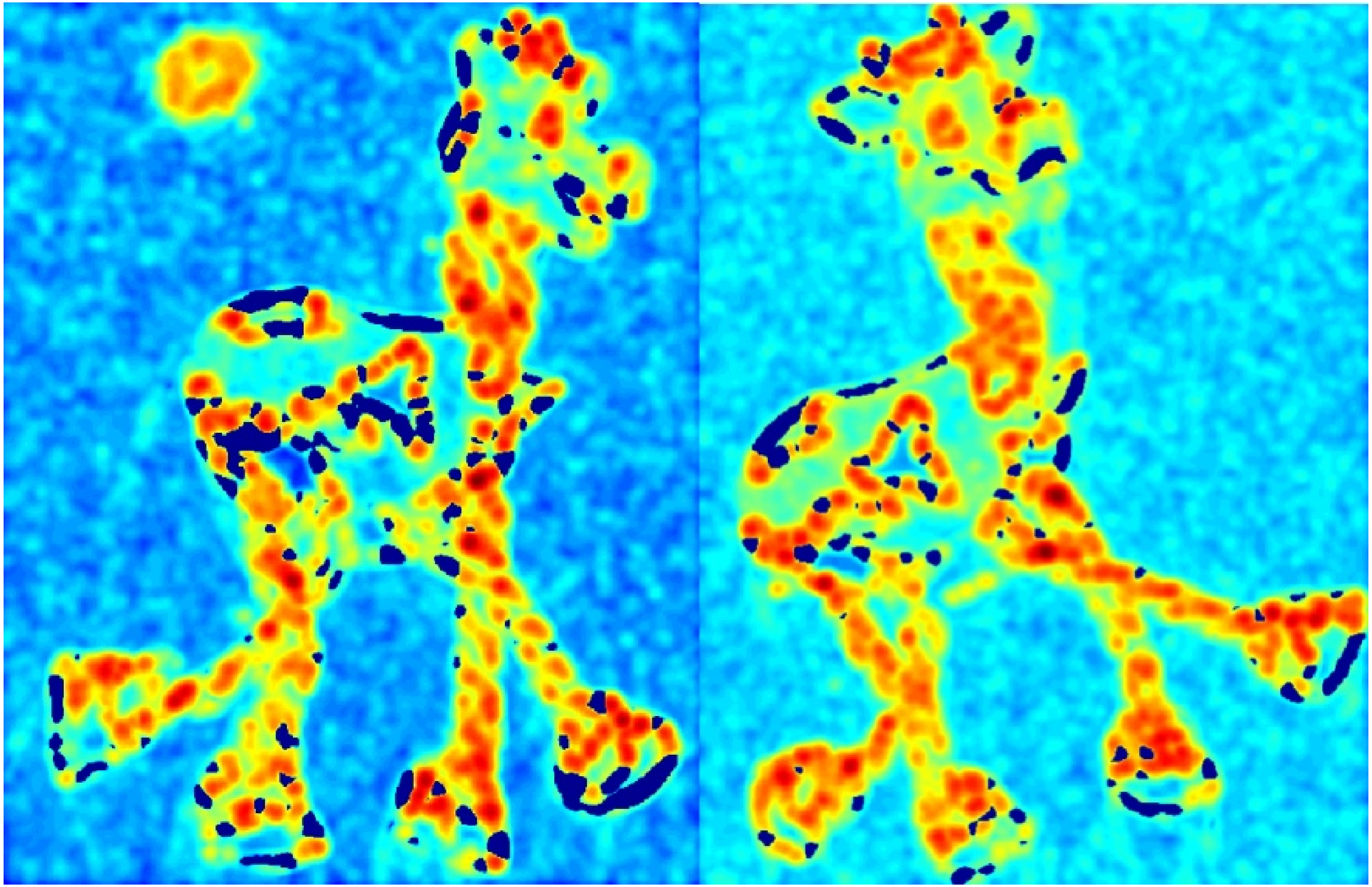
5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2$$

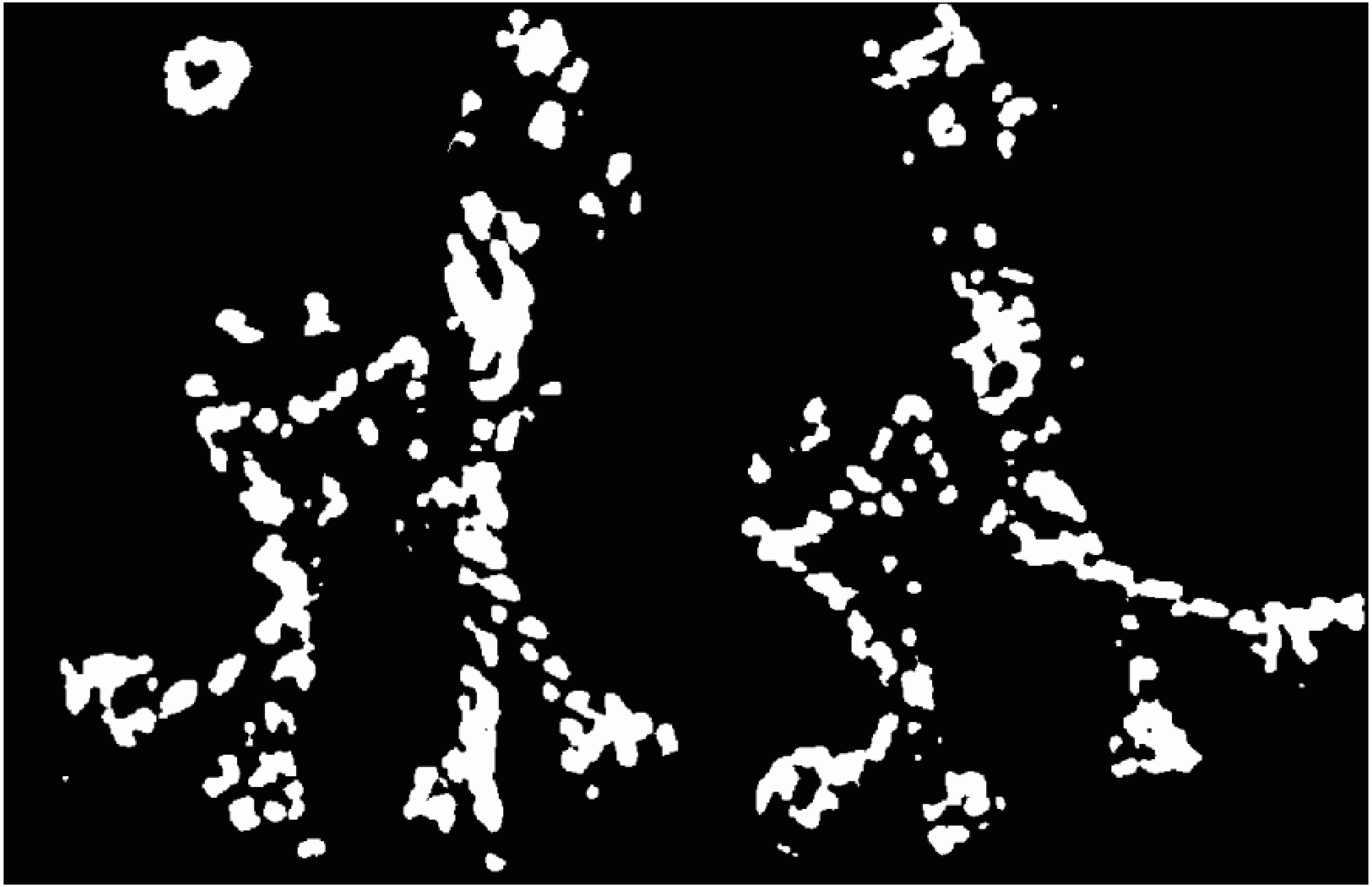
6. Threshold on value of R. Compute nonmax suppression.

Harris corner detector (input)

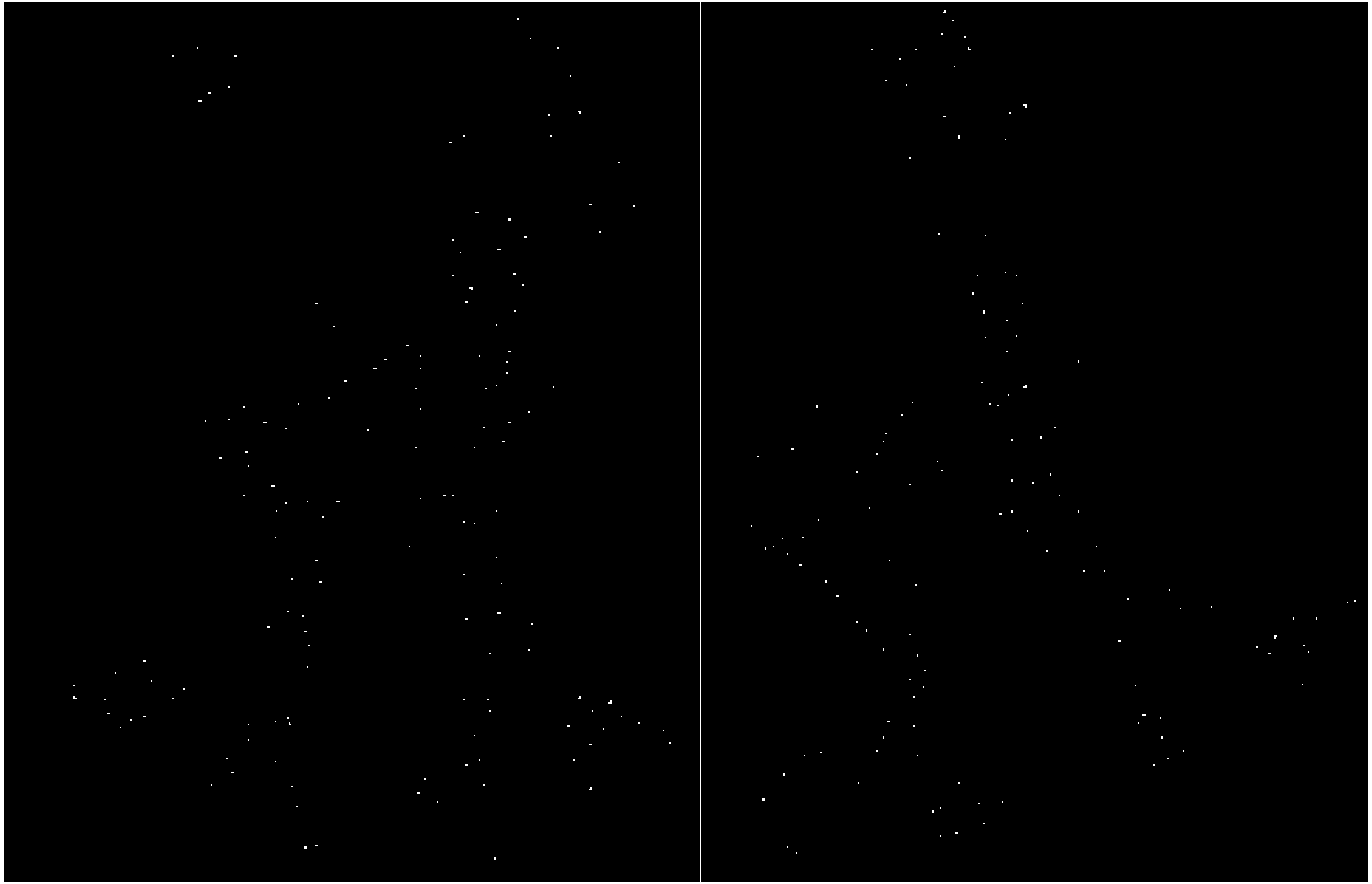
Corner response R



Threshold on R



Local maximum of R



Harris corner detector

Harris Detector: Summary

- 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*

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

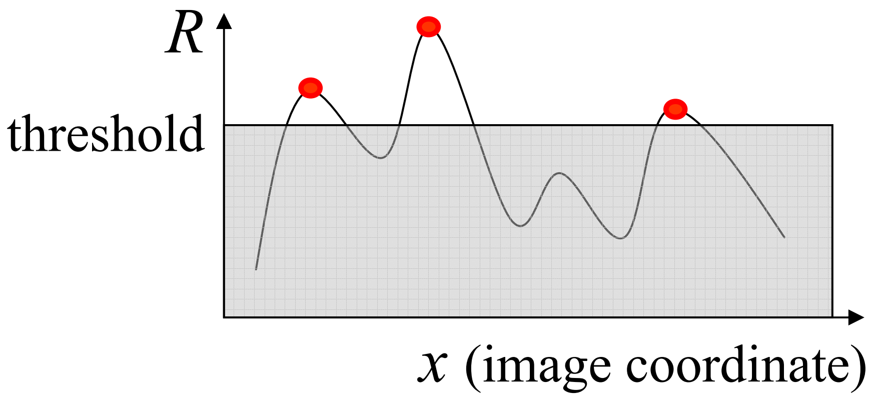
- A good (corner) point should have a *large intensity change* in *all directions*, i.e. *R* should be large positive

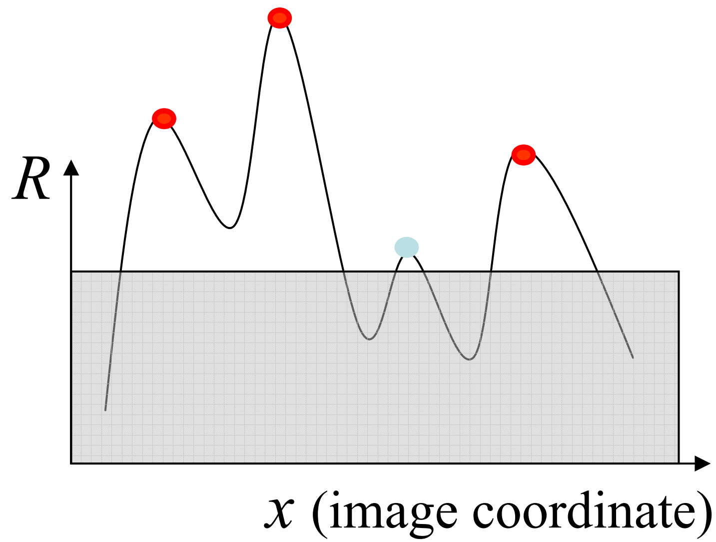
Harris Detector: Some Properties

- Partial invariance to *affine intensity* change

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

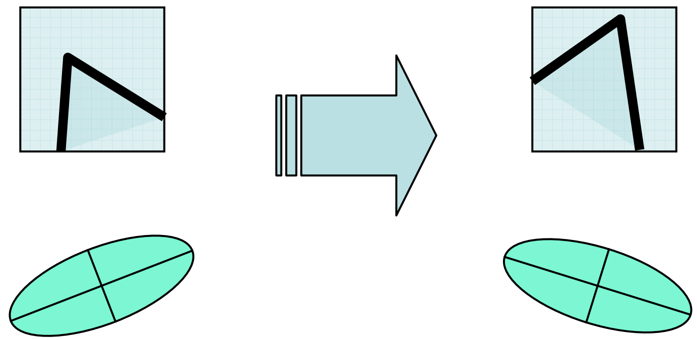
✓ Intensity scale: $I \rightarrow a\,I$





Harris Detector: Some Properties

- Rotation invariance



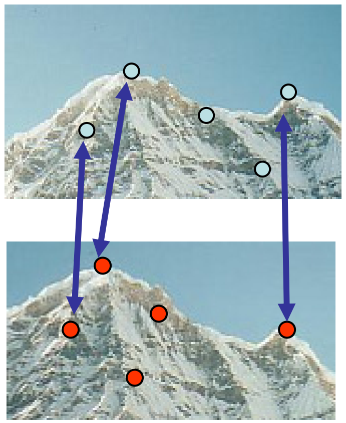
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

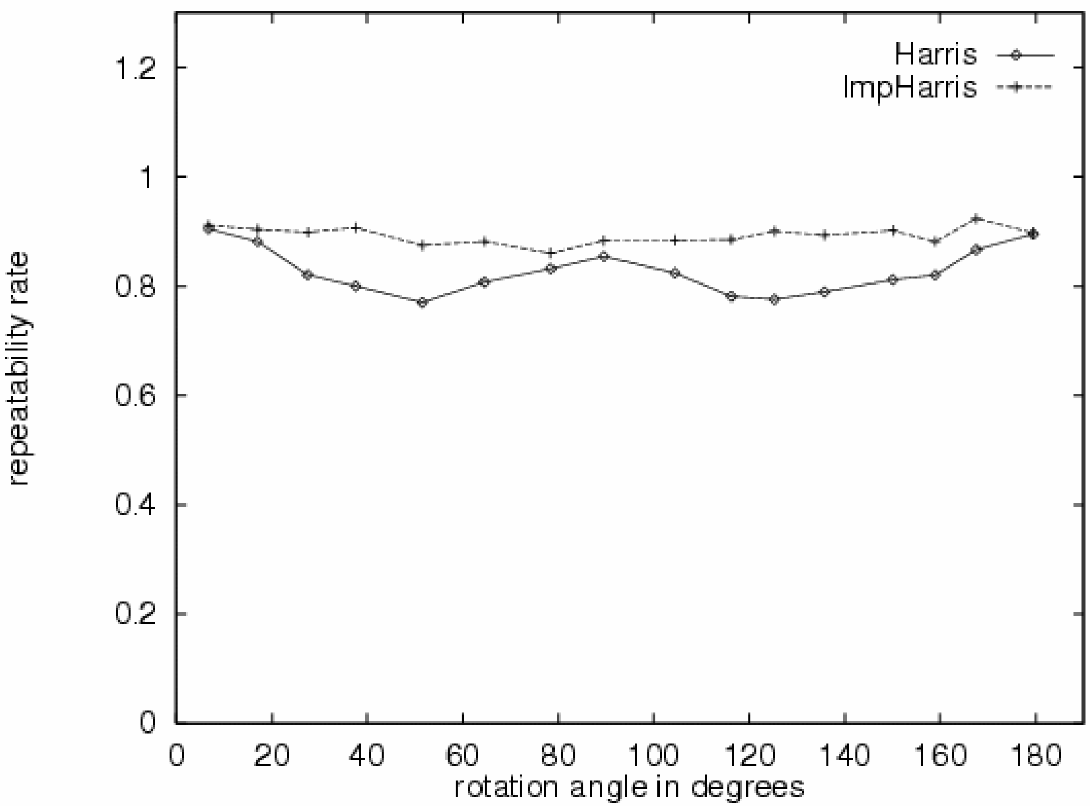
Corner response R is invariant to image rotation

Harris Detector is rotation invariant

Repeatability rate:

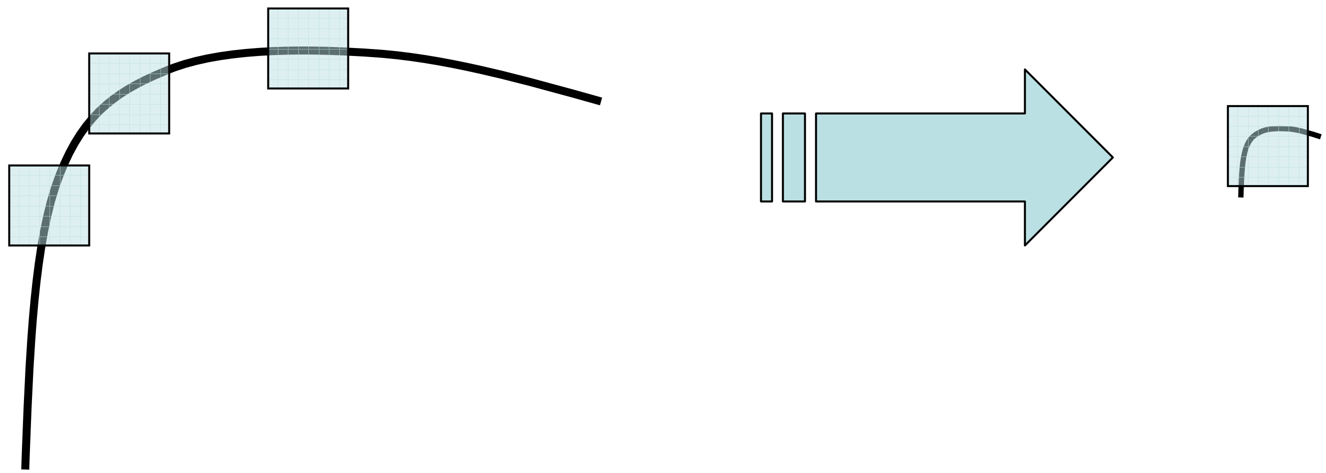
$$\frac{\text{\# correspondences}}{\text{\# possible correspondences}}$$





Harris Detector: Some Properties

- But: non-invariant to *image scale*!



All points will be classified as edges

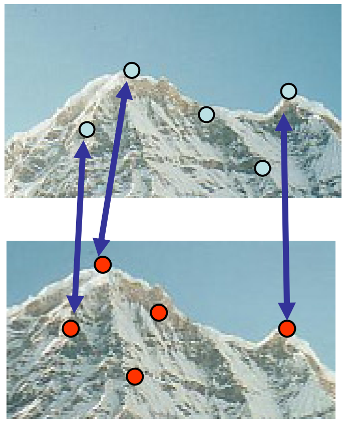
Corner !

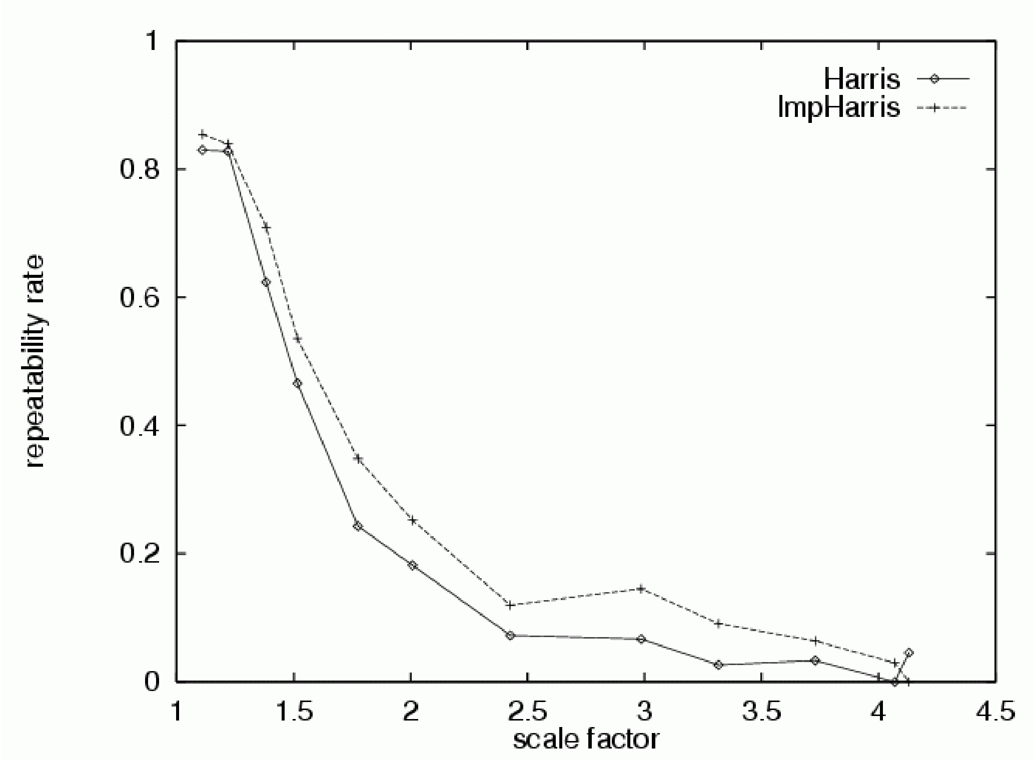
Harris Detector: Some Properties

- Quality of Harris detector for different scale changes

Repeatability rate:

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





SIFT
(Scale Invariant Feature Transform)

SIFT

- SIFT is an carefully designed procedure with empirically determined parameters for the invariant and distinctive features.

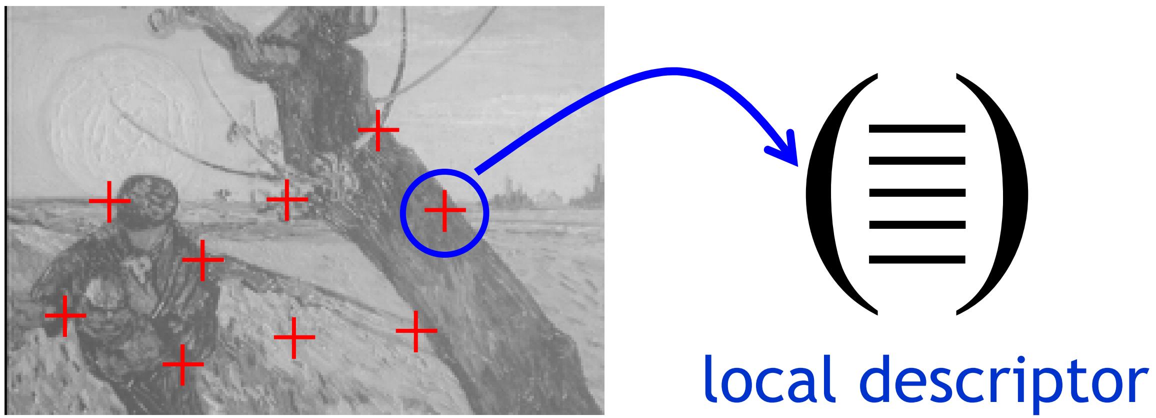
SIFT stages:

- Scale-space extrema detection
- Keypoint localization

detector

- Orientation assignment
- Keypoint descriptor

descriptor



A 500x500 image gives about 2000 features

1. Detection of scale-space extrema

- For scale invariance, search for stable features across all possible scales using a continuous function of scale, scale space.
- SIFT uses DoG filter for scale space because it is efficient and as stable as scale-normalized Laplacian of Gaussian.

DoG filtering

Convolution with a variable-scale Gaussian

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

$$G(x, y, \sigma) = 1/(2\pi\sigma^2) \exp^{-(x^2+y^2)/\sigma^2}$$

Difference-of-Gaussian (DoG) filter

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

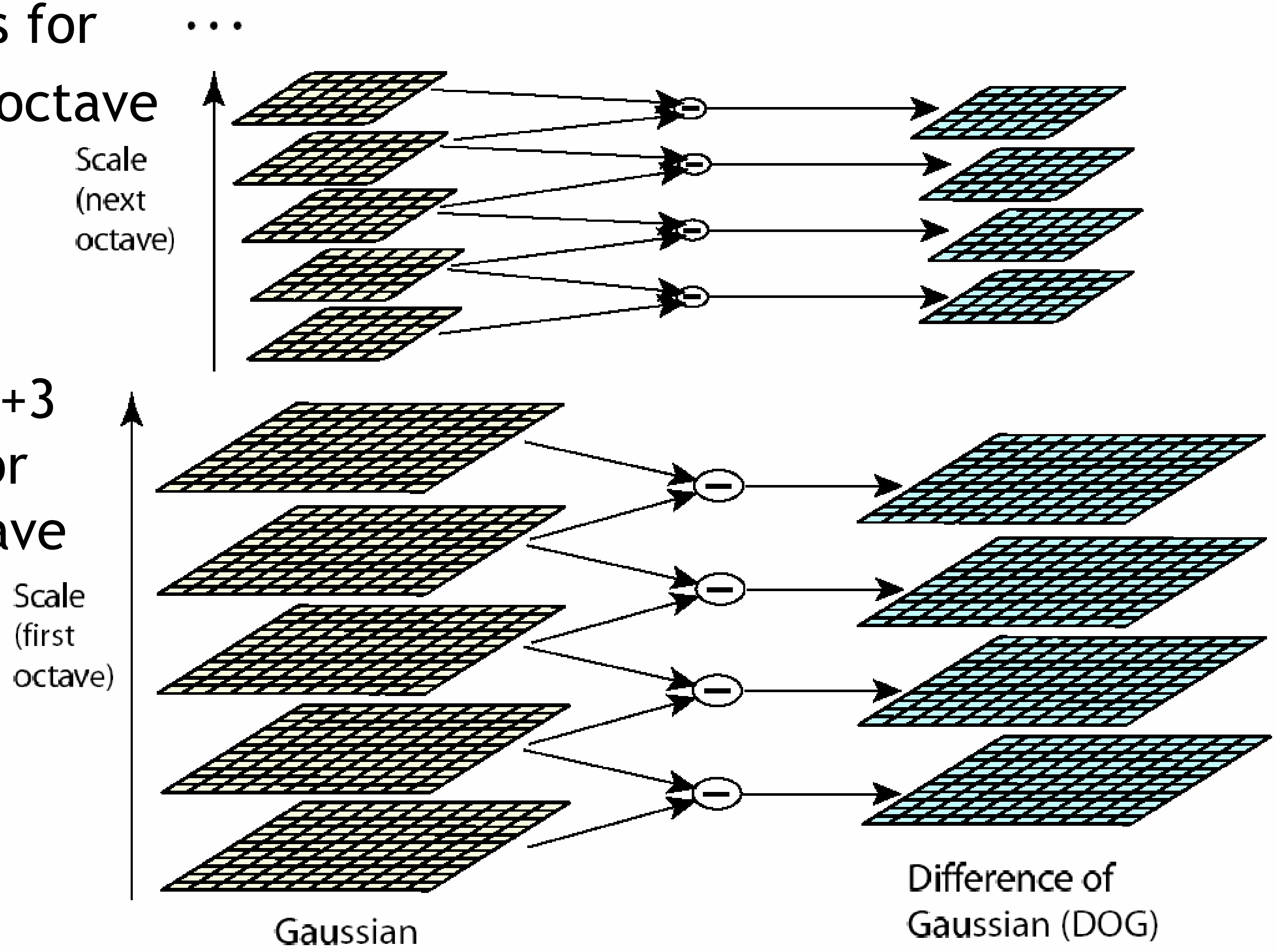
Convolution with the DoG filter

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

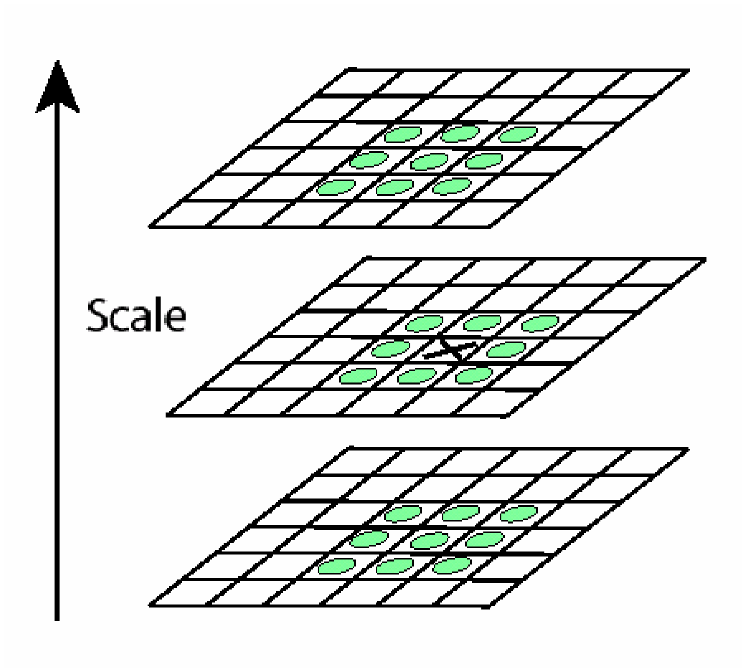
Scale space

σ doubles for the next octave

$K=2^{(1/s)}$, s+3 images for each octave



Keypoint localization

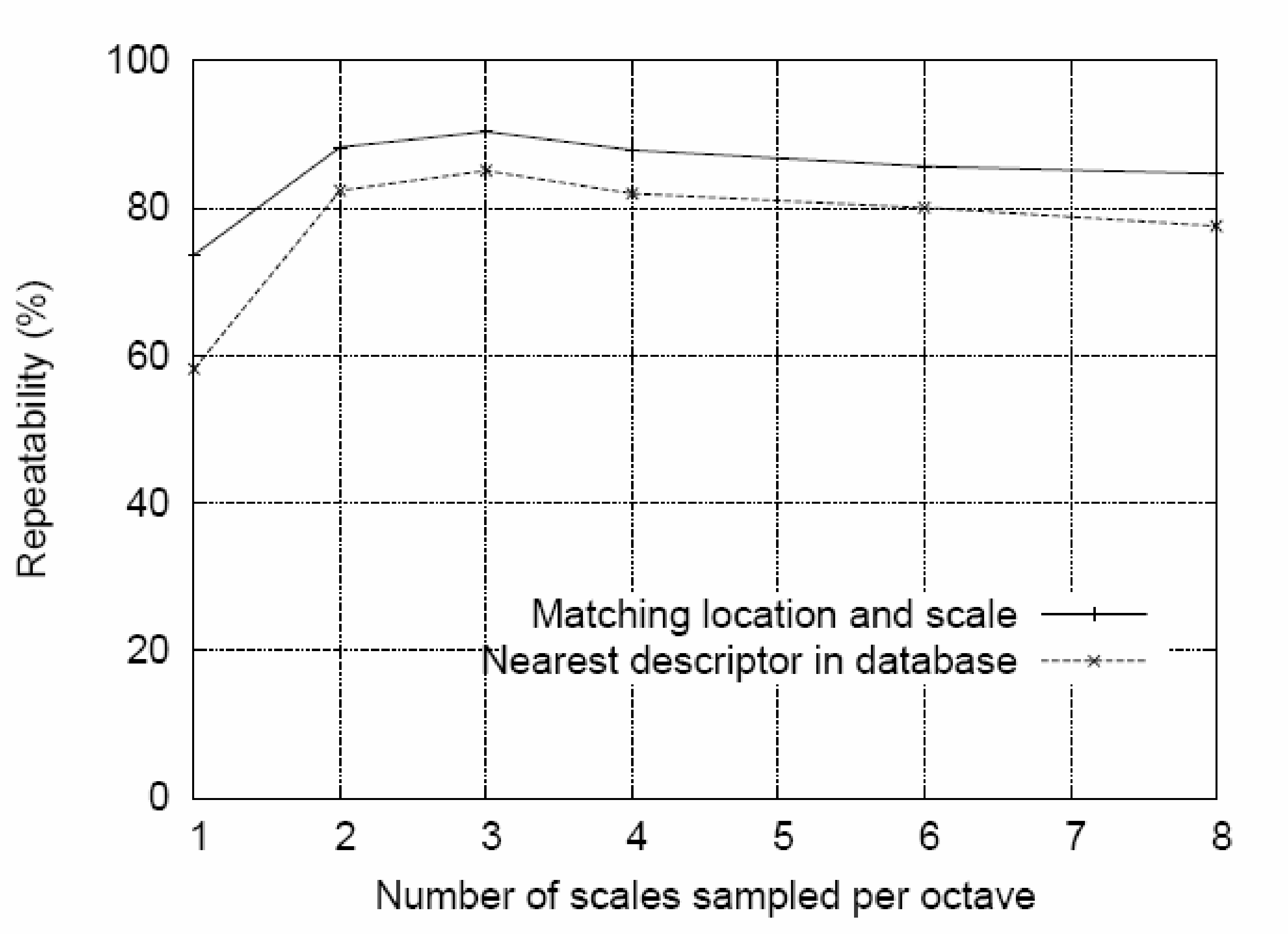


X is selected if it is larger or smaller than all 26 neighbors

Decide scale sampling frequency

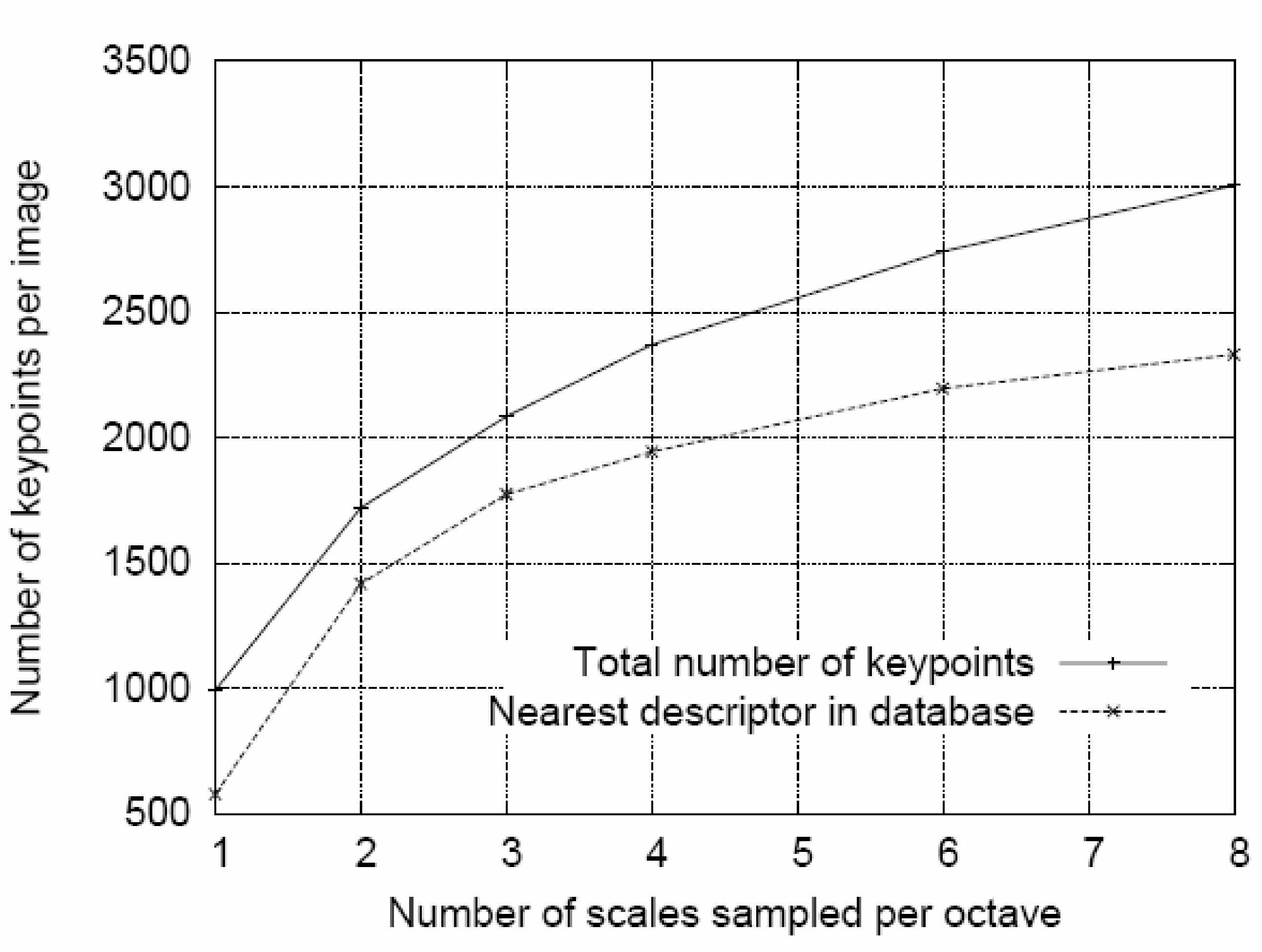
- It is impossible to sample the whole space, tradeoff efficiency with completeness.
- Decide the best sampling frequency by experimenting on 32 real image subject to synthetic transformations.

Decide scale sampling frequency

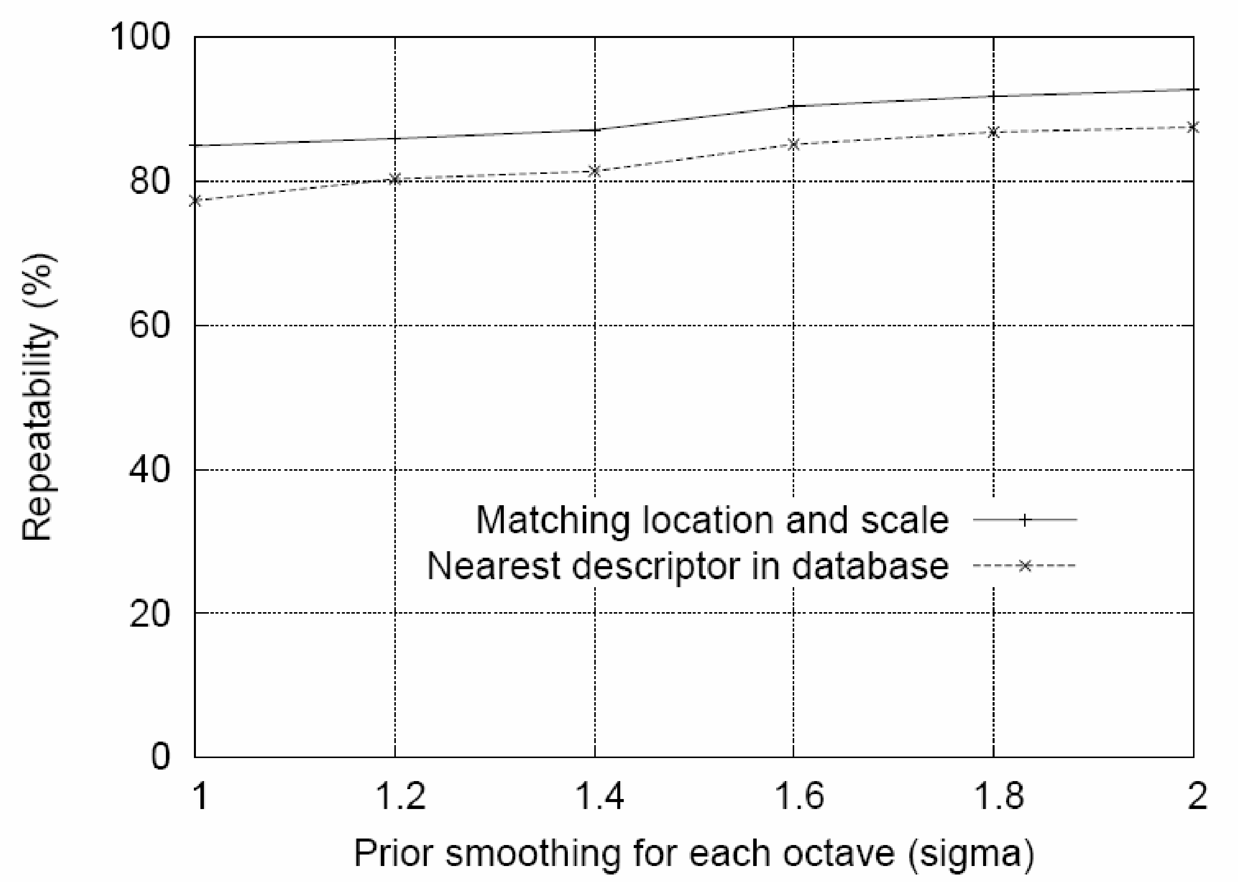


S=3, for larger s, too many unstable features

Decide scale sampling frequency

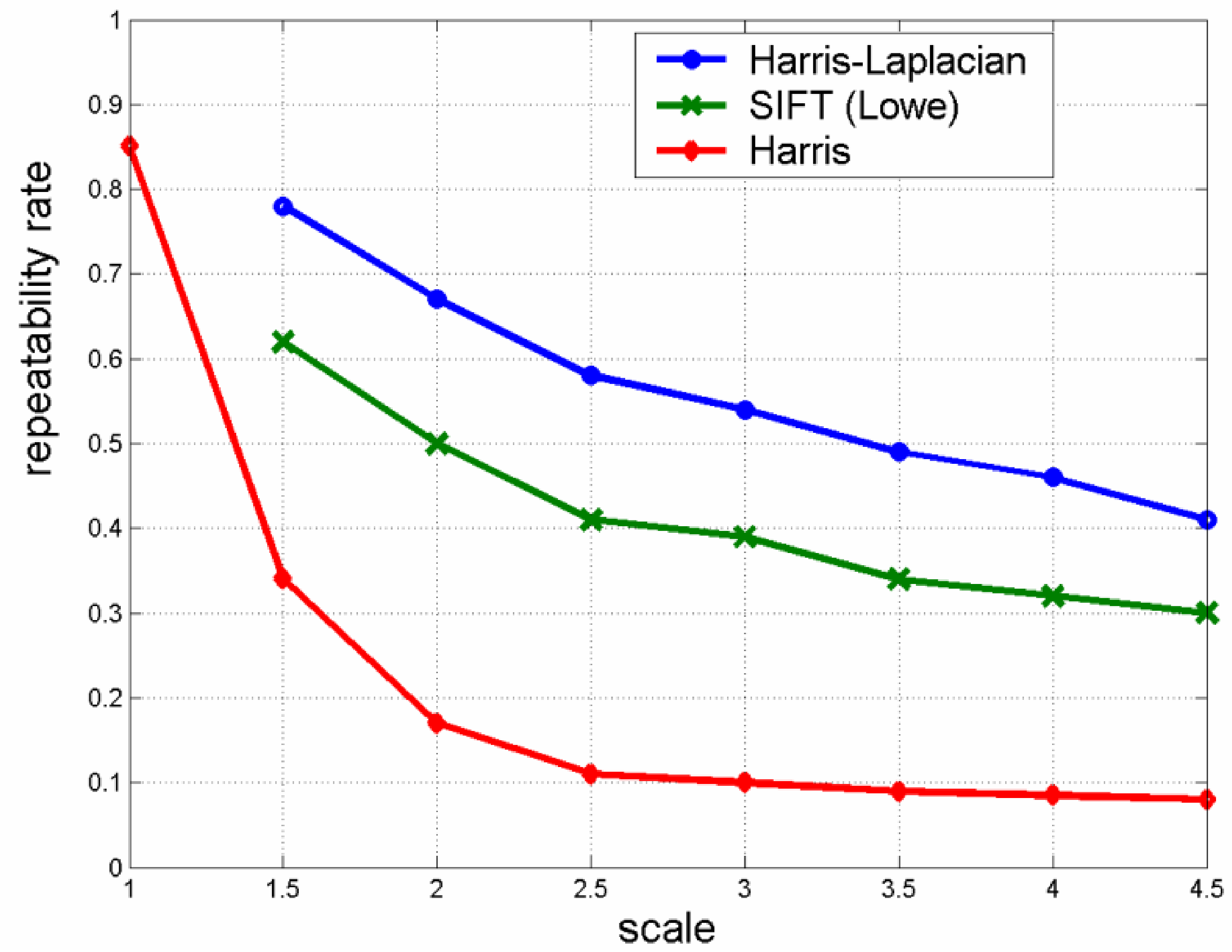


Pre-smoothing



σ =1.6, plus a double expansion

Scale invariance



2. Accurate keypoint localization

- Reject points with low contrast and poorly localized along an edge
- Fit a 3D quadratic function for sub-pixel maxima

Accurate keypoint localization

Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point:

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

where D and its derivatives are evaluated at the sample point and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of this function with respect to $\mathbf{x}$ and setting it to zero, giving

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}. \qquad (3)$$

If $\hat{\mathbf{x}}$ has offset larger than 0.5, sample point is changed.

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

If $|D(\hat{\mathbf{x}})|$ is less than 0.03 (low contrast), it is discarded.

Eliminating edge responses

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

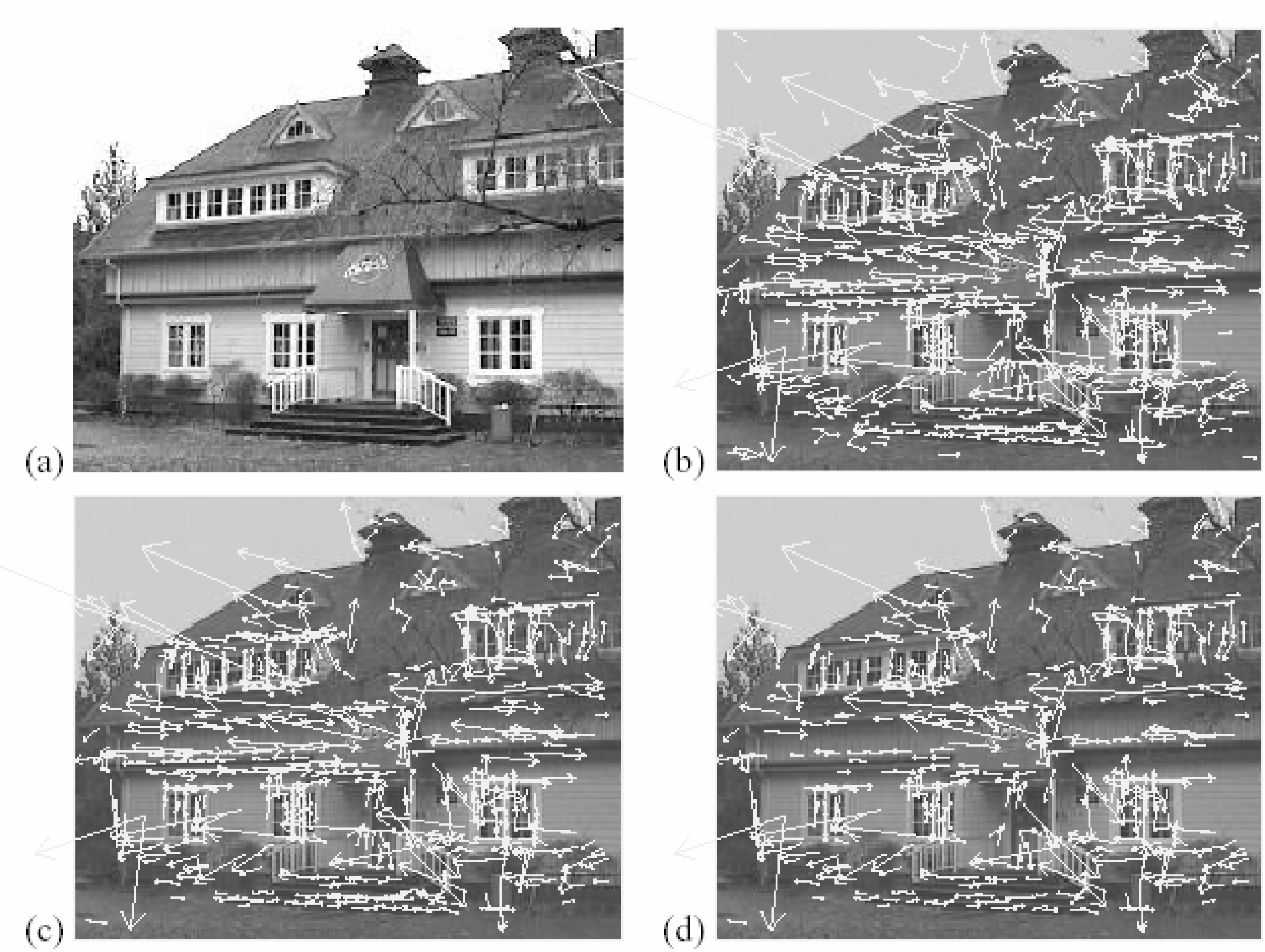
$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

Let $\alpha = r\beta$ $\quad \frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$

Keep the points with $\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}.$ r=10

Keypoint detector



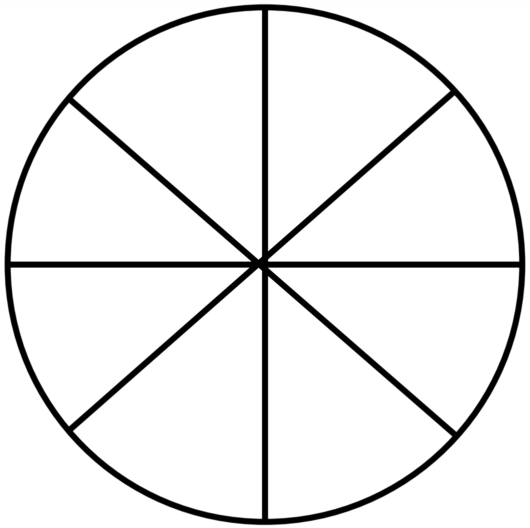
(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

3. Orientation assignment

- By assigning a consistent orientation, the keypoint descriptor can be orientation invariant.
- For a keypoint, L is the image with the closest scale,

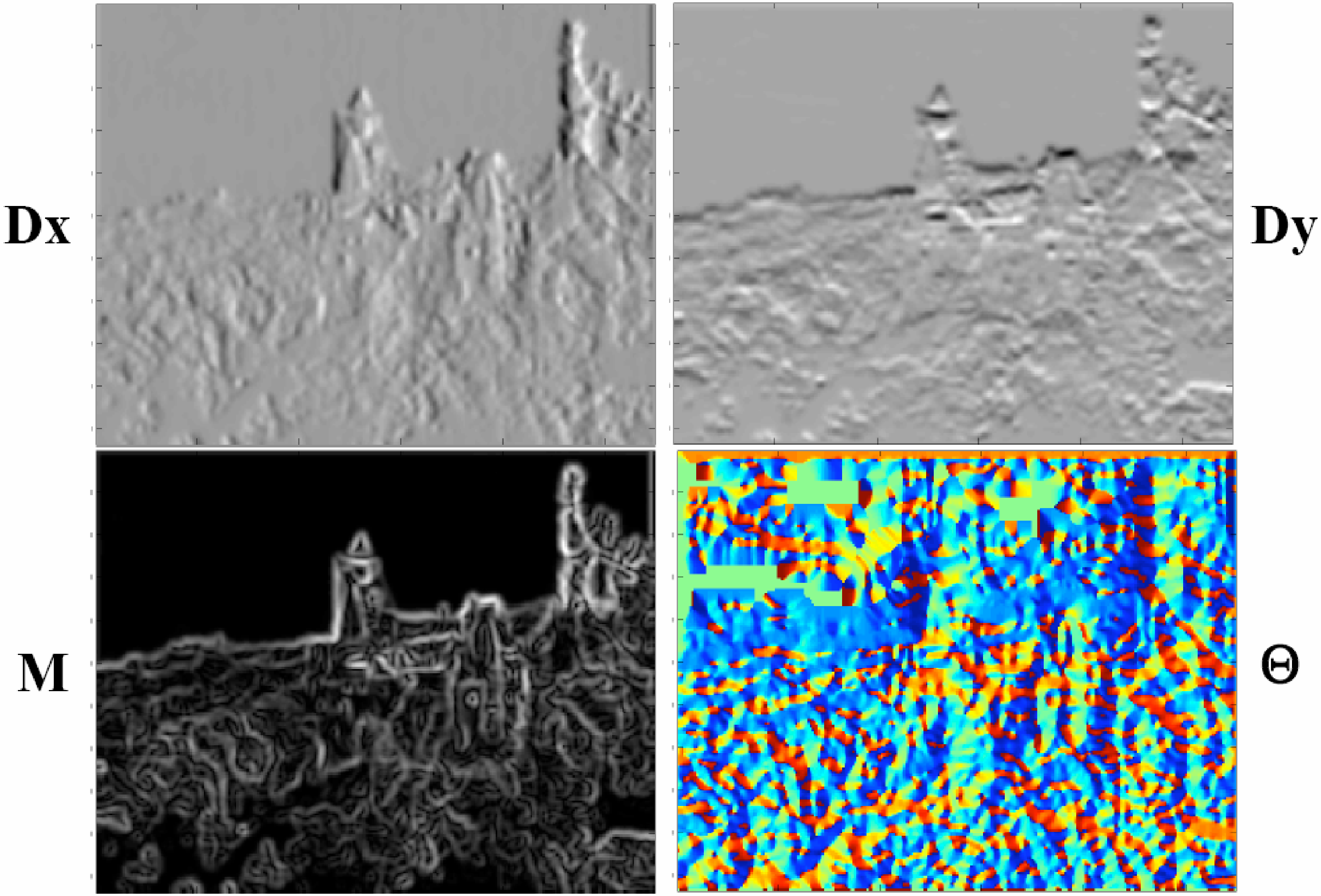
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

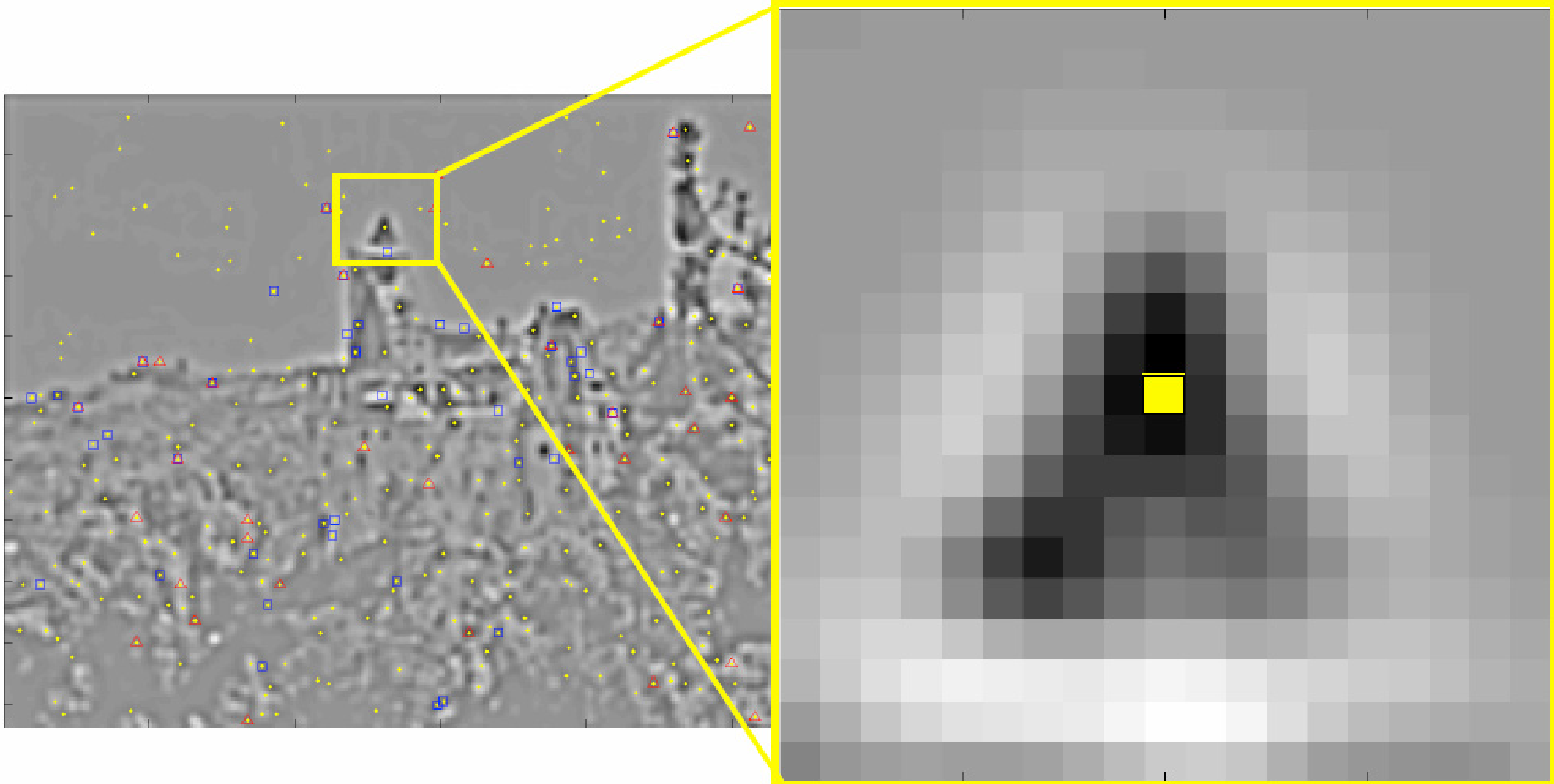


orientation histogram

Orientation assignment

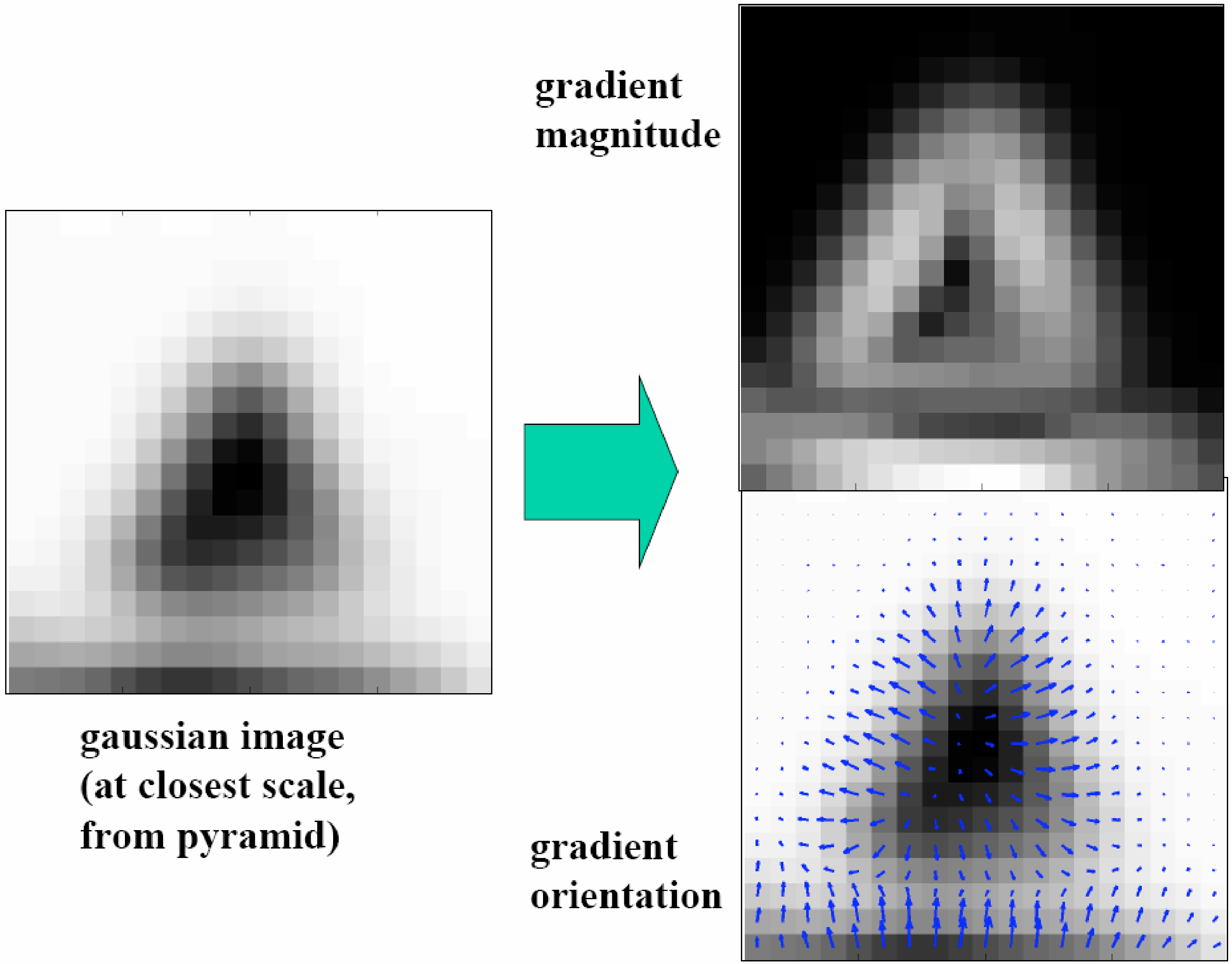


Orientation assignment

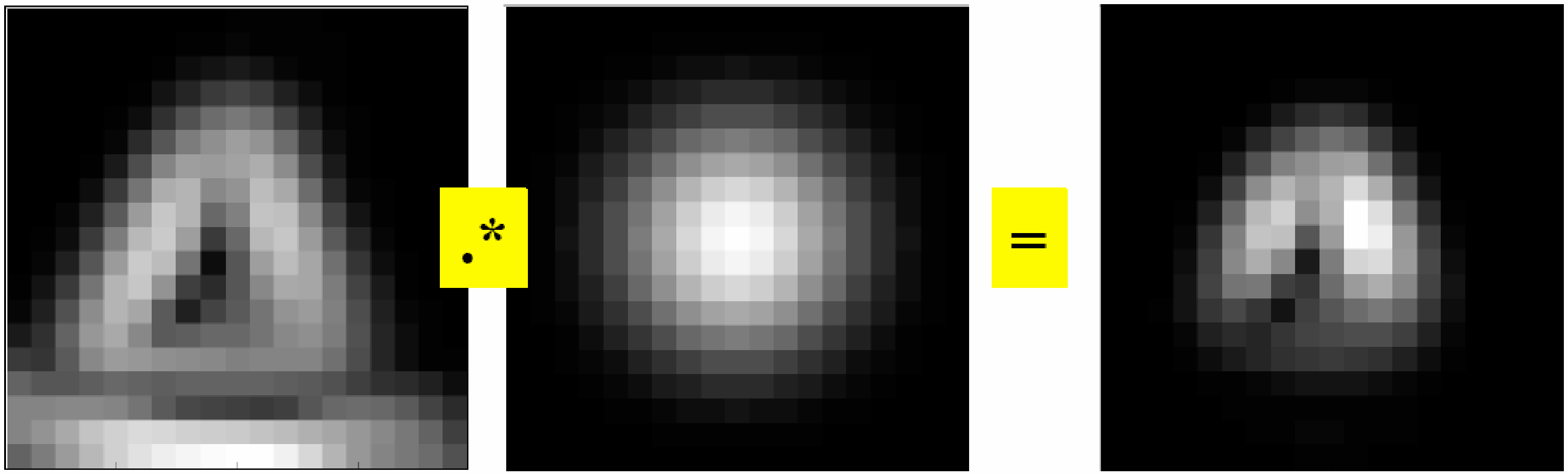


- Keypoint location = extrema location
- Keypoint scale is scale of the DOG image

Orientation assignment



Orientation assignment

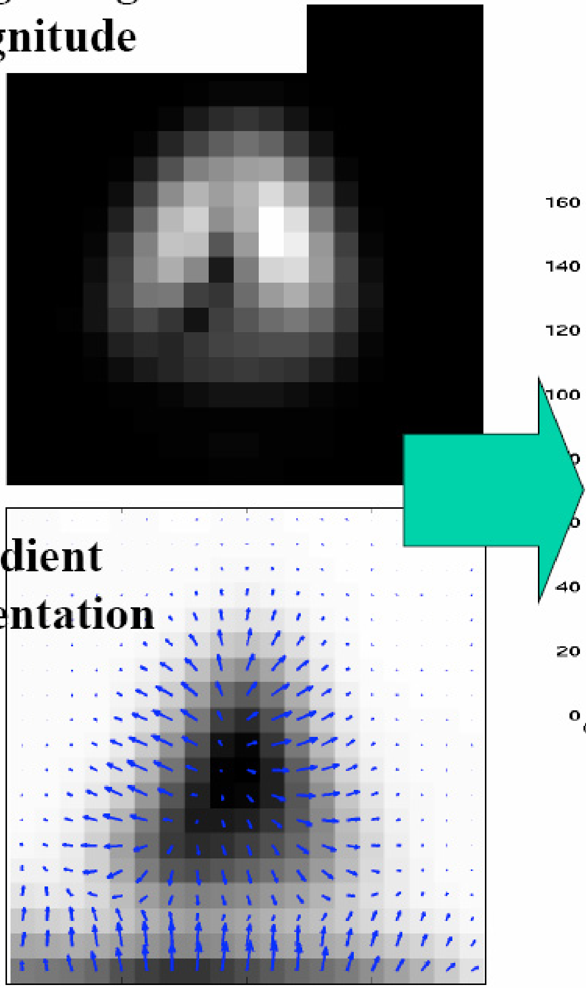


gradient magnitude | weighted by 2D gaussian kernel | weighted gradient magnitude

Orientation assignment

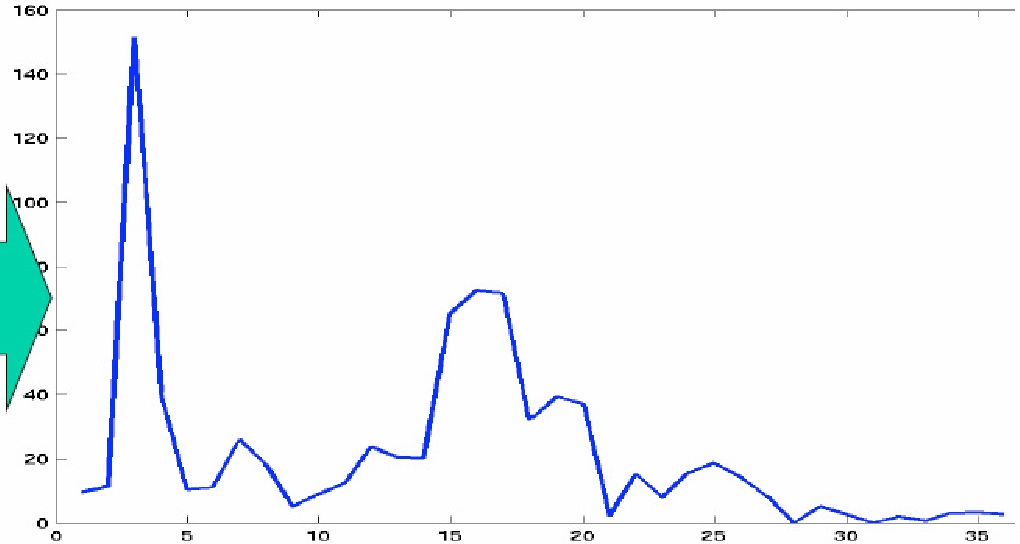
weighted gradient magnitude

gradient orientation



weighted orientation histogram.

Each bucket contains sum of weighted gradient magnitudes corresponding to angles that fall within that bucket.



36 buckets

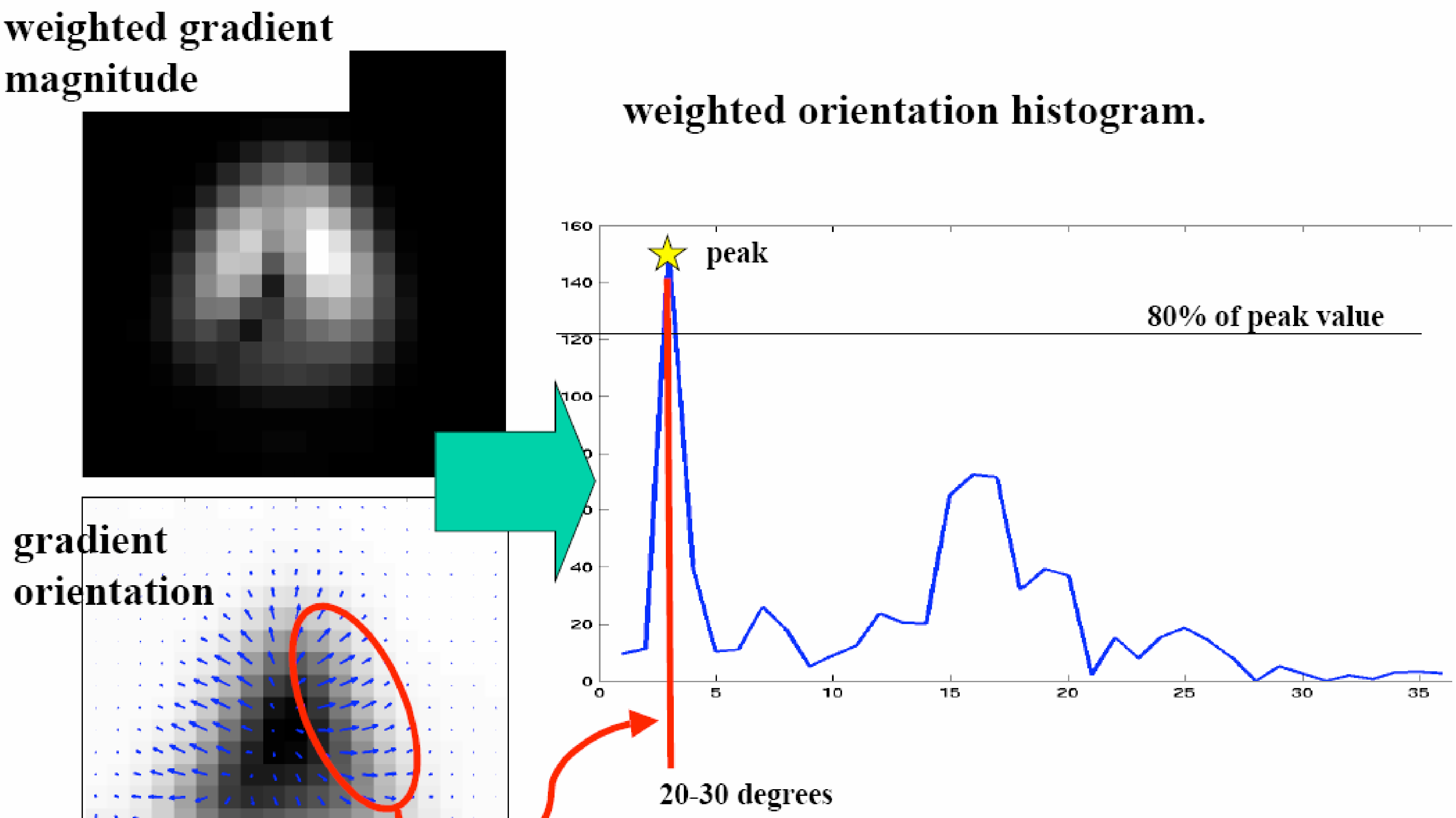
10 degree range of angles in each bucket, i.e.

0 <=ang<10 : bucket 1

10<=ang<20 : bucket 2

20<=ang<30 : bucket 3 …

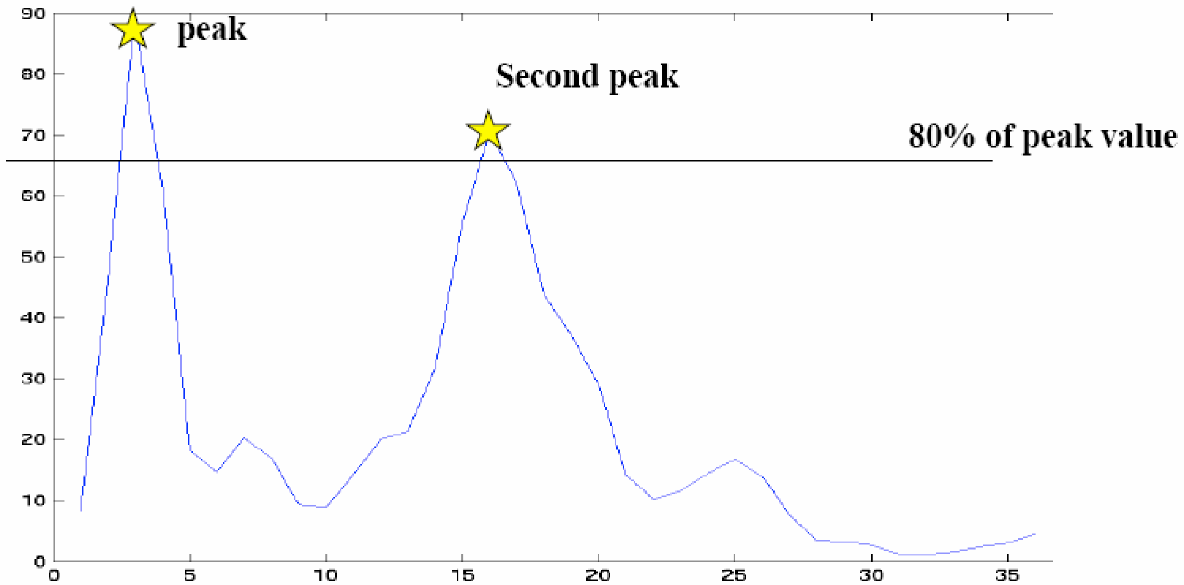
Orientation assignment



Orientation of keypoint is approximately 25 degrees

Orientation assignment

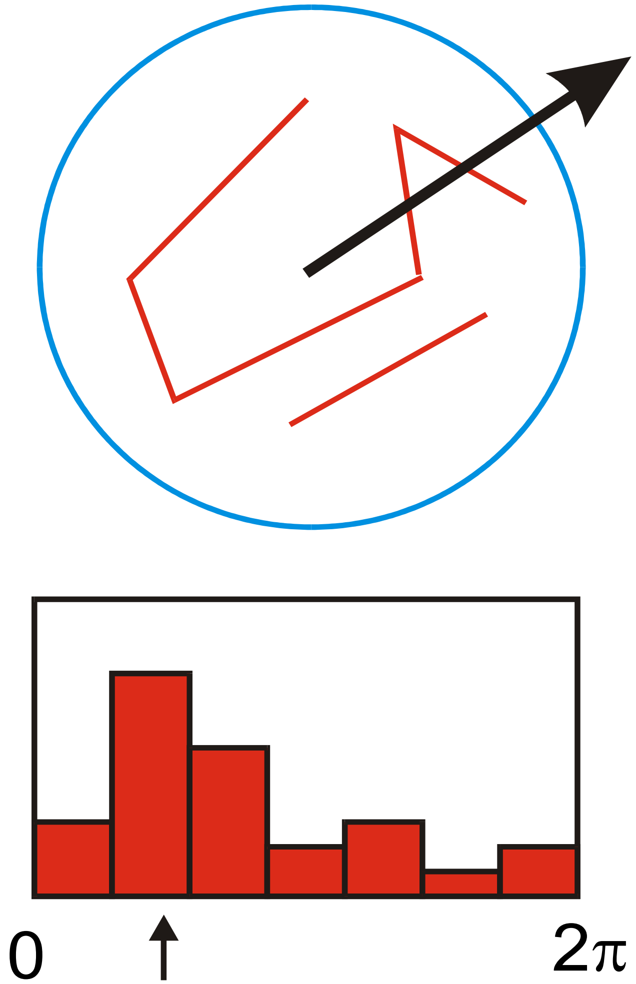
There may be multiple orientations.



In this case, generate duplicate keypoints, one with orientation at 25 degrees, one at 155 degrees.

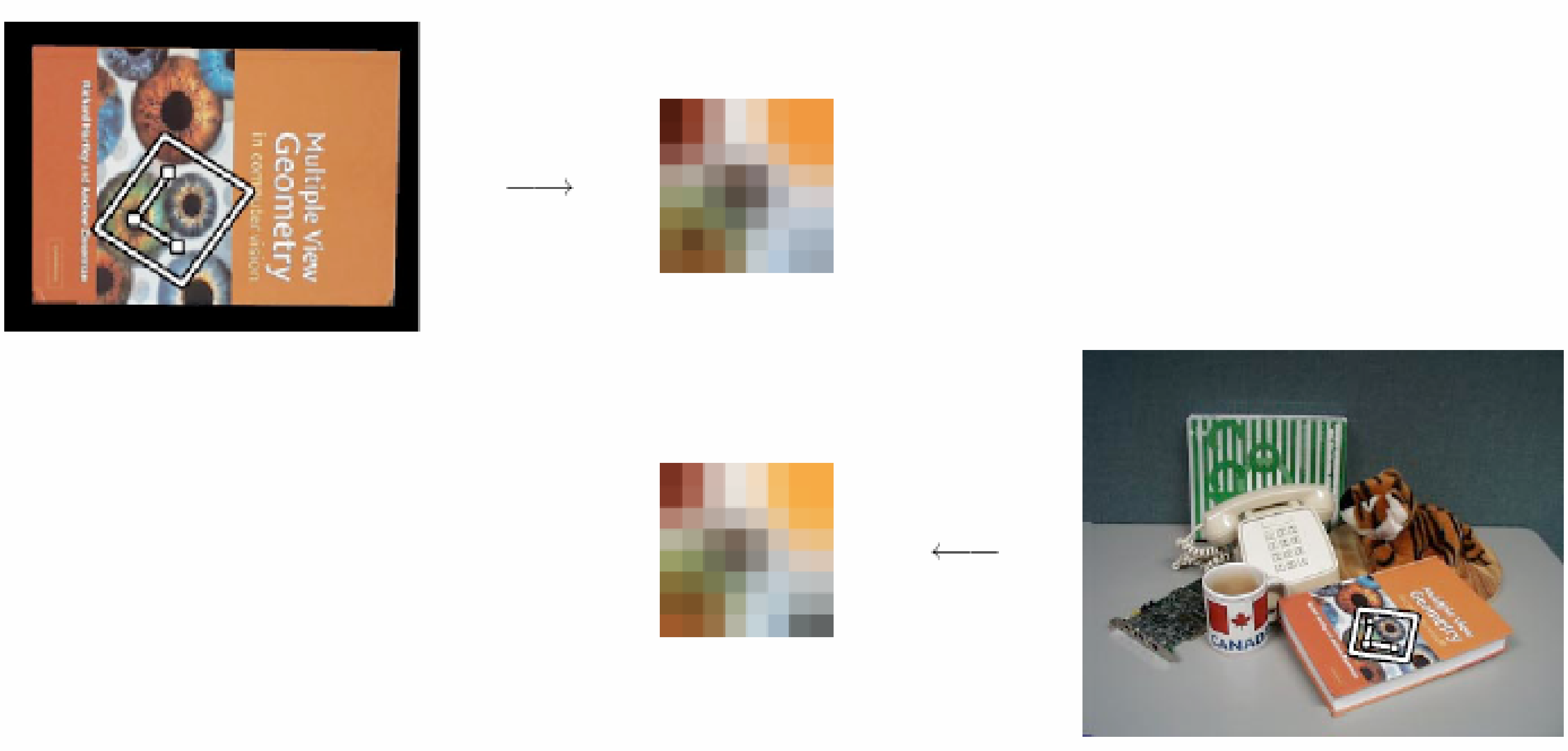
Design decision: you may want to limit number of possible multiple peaks to two.

Orientation assignment



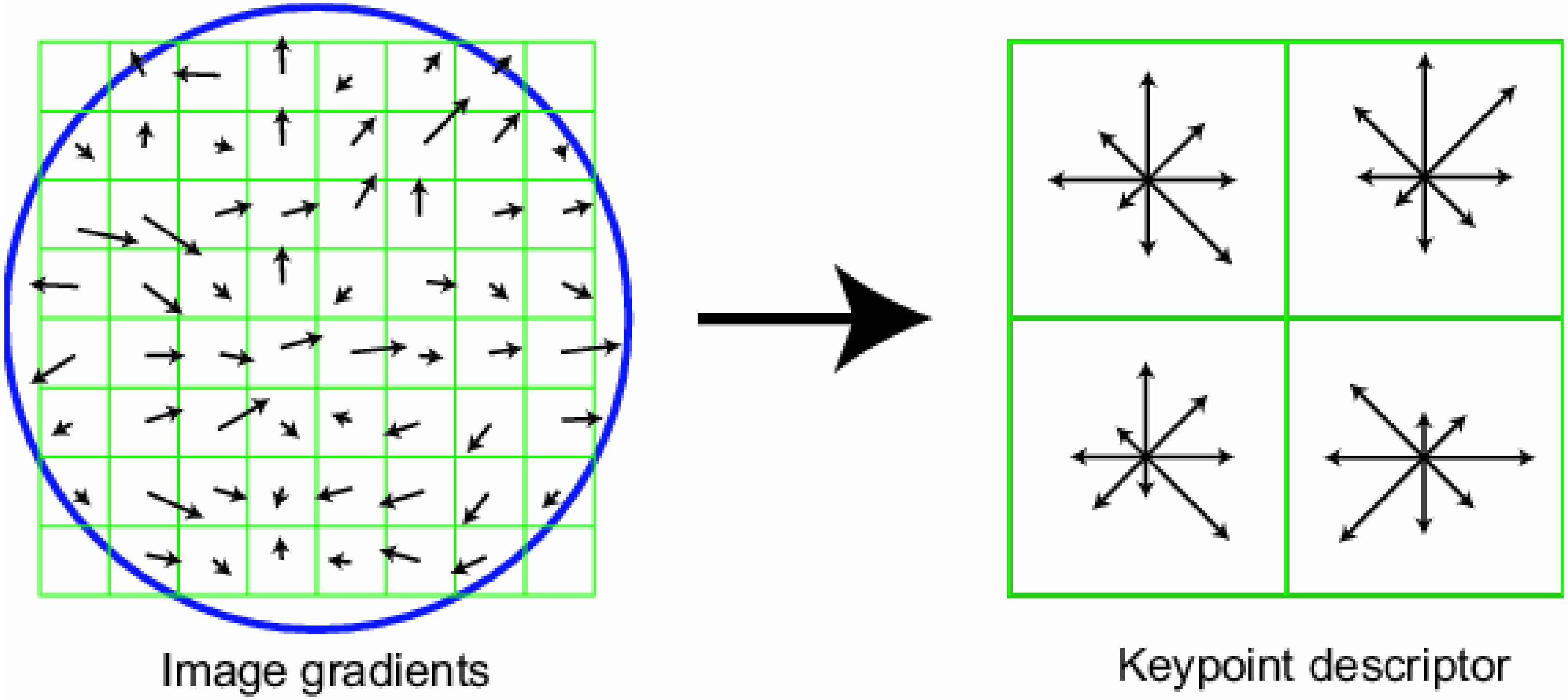
- 36-bin orientation histogram over 360°, weighted by m and 1.5*scale falloff
- Peak is the orientation
- Local peak within 80% creates multiple orientations
- About 15% has multiple orientations

Orientation invariance

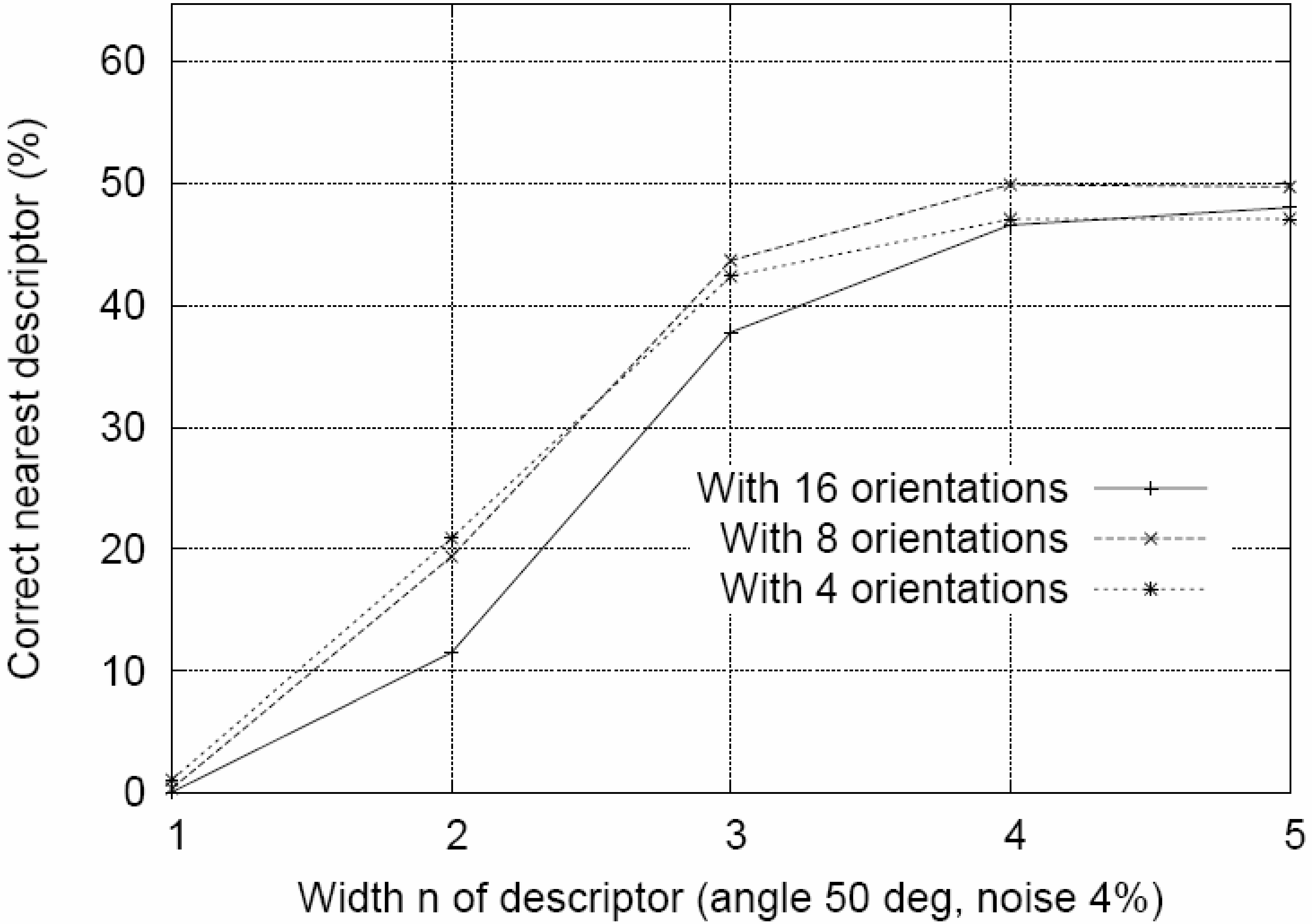


4. Local image descriptor

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions
- Normalized, clip the components larger than 0.2

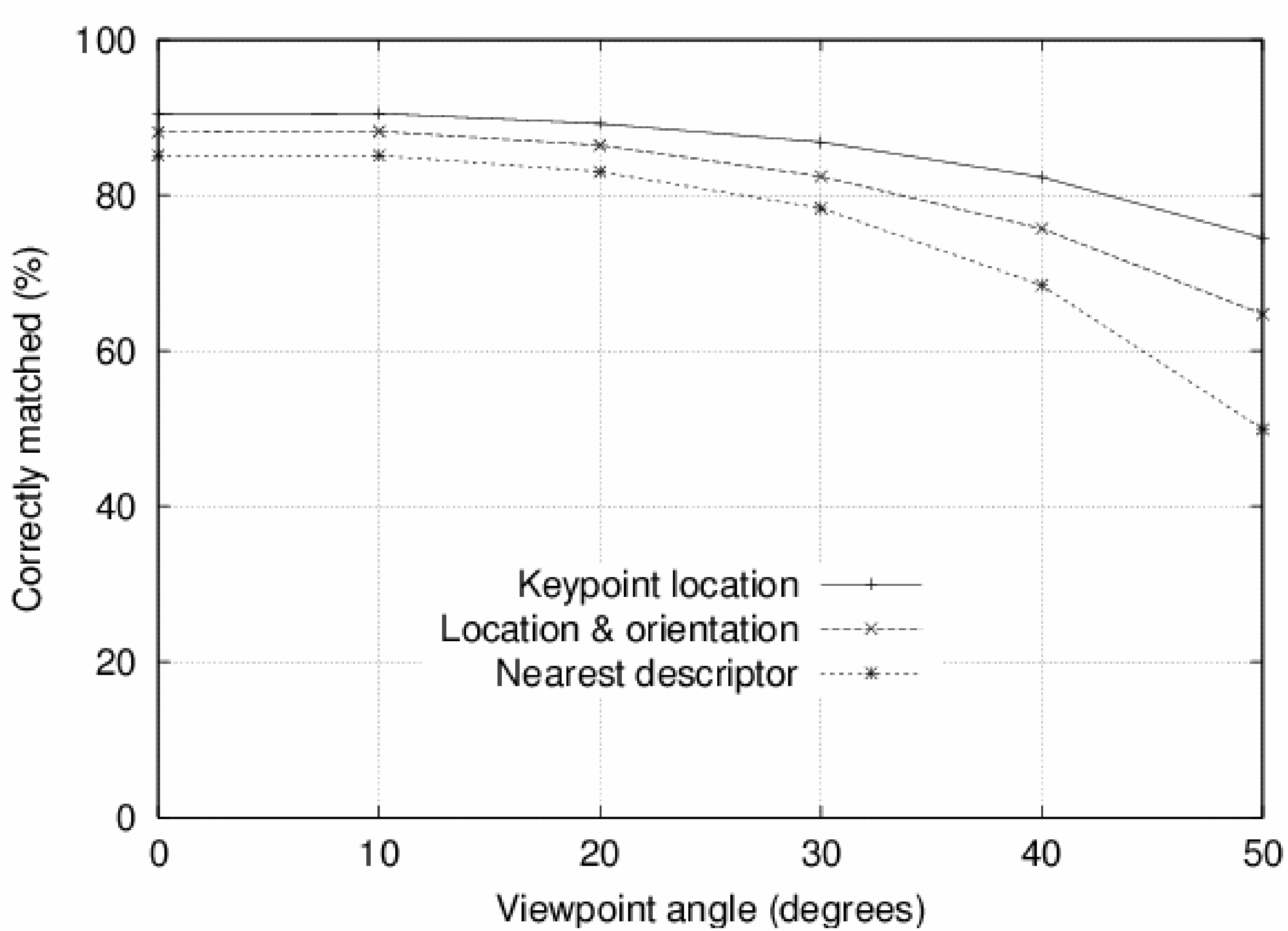


Why 4x4x8?

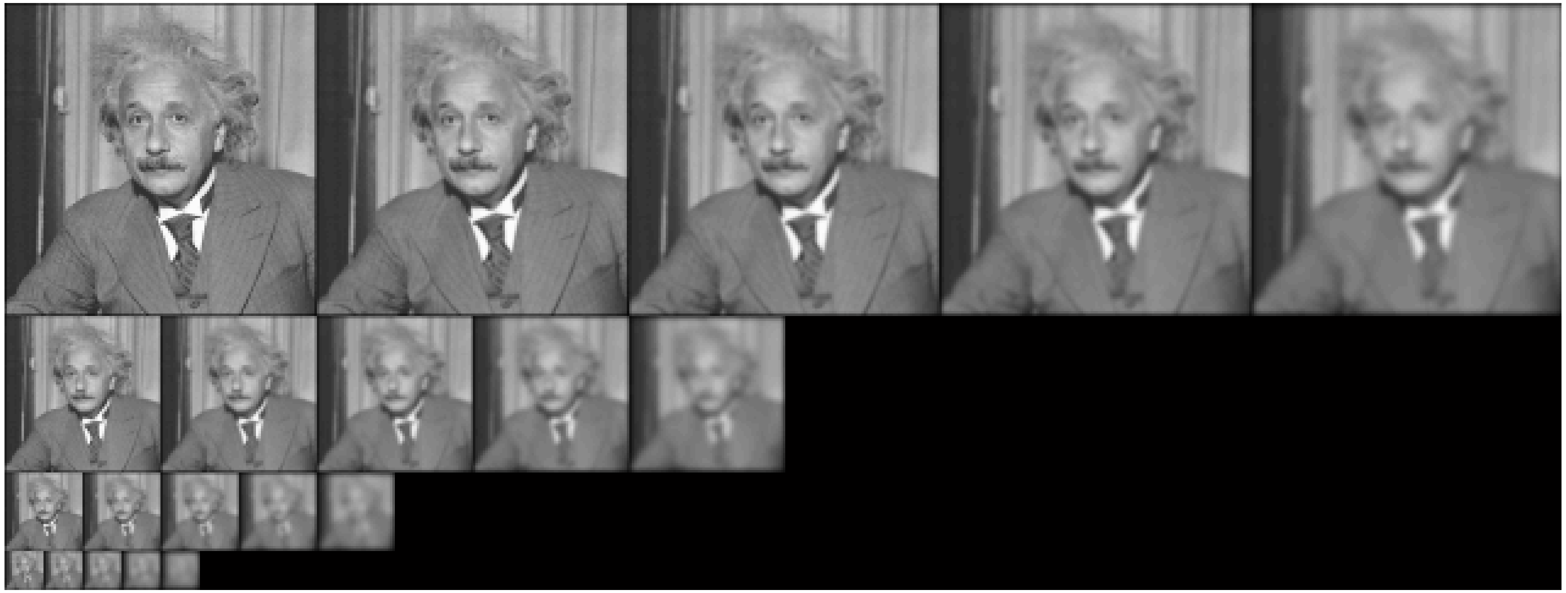


60
50
40
30
20
10
0
1
2
3
4
5
Correct nearest descriptor (%)
Width n of descriptor (angle 50 deg, noise 4%)
With 16 orientations
With 8 orientations
With 4 orientations

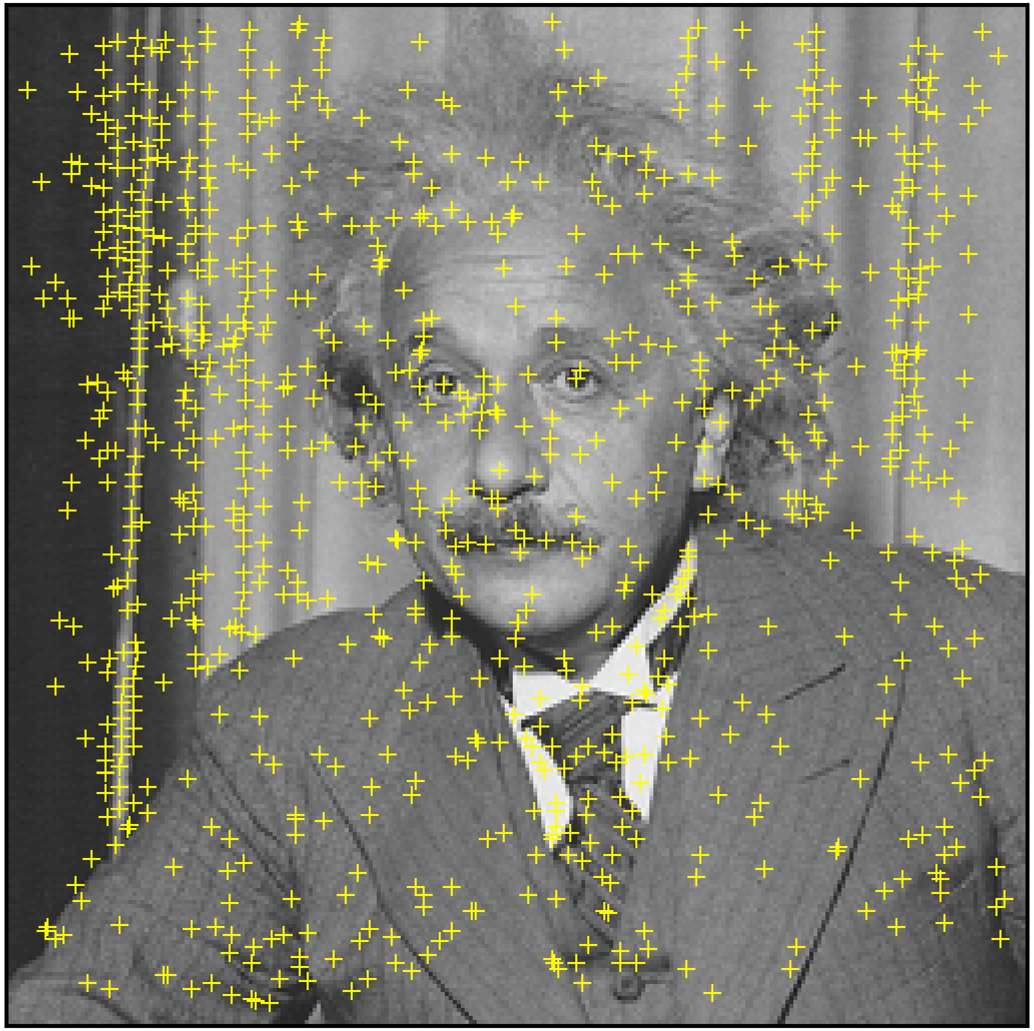
Sensitivity to affine change



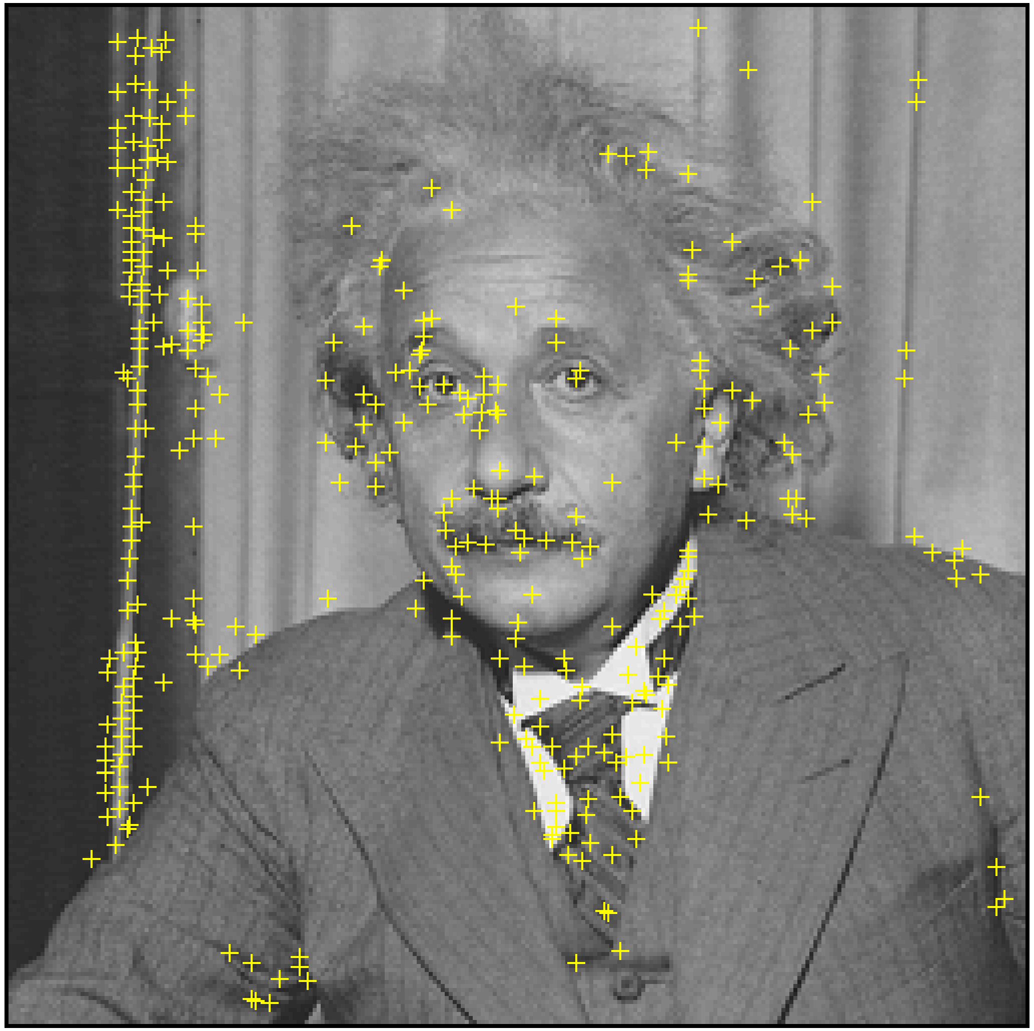
SIFT demo



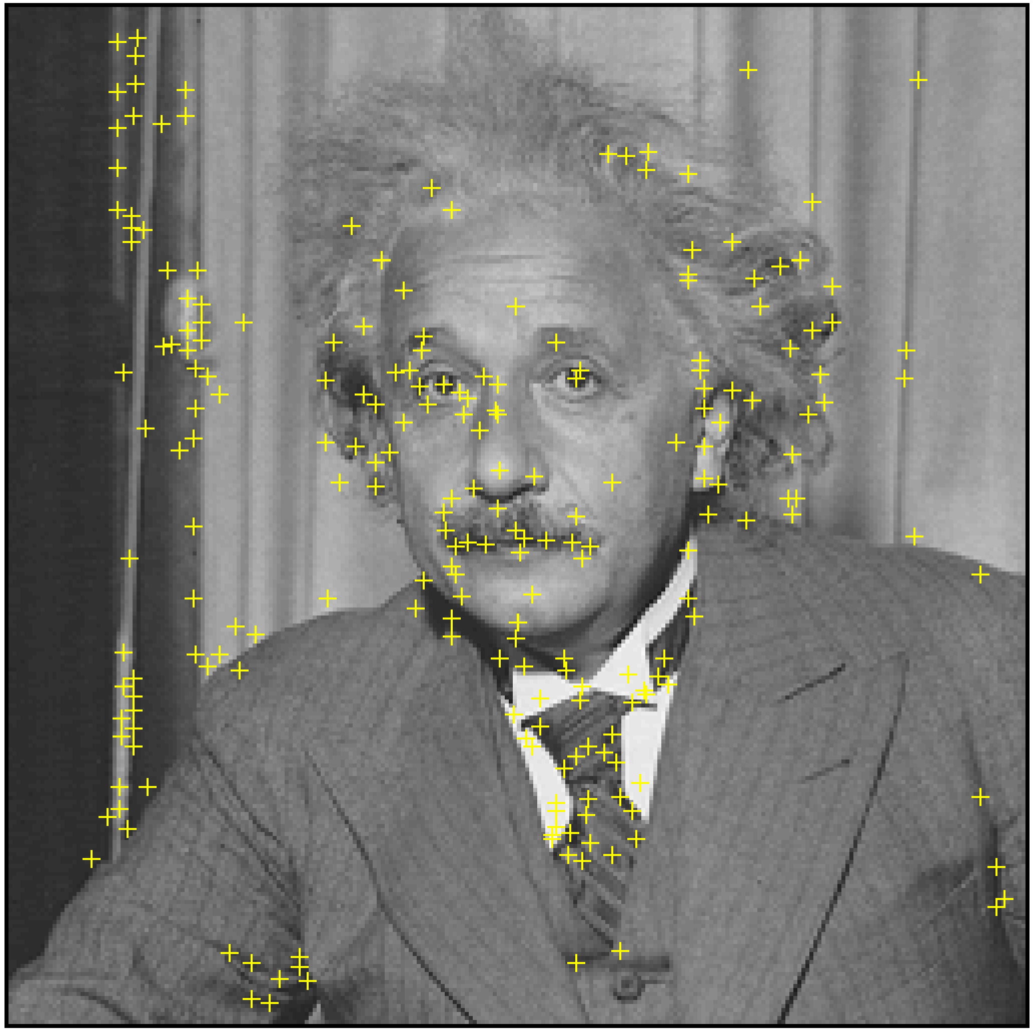
Maxima in D



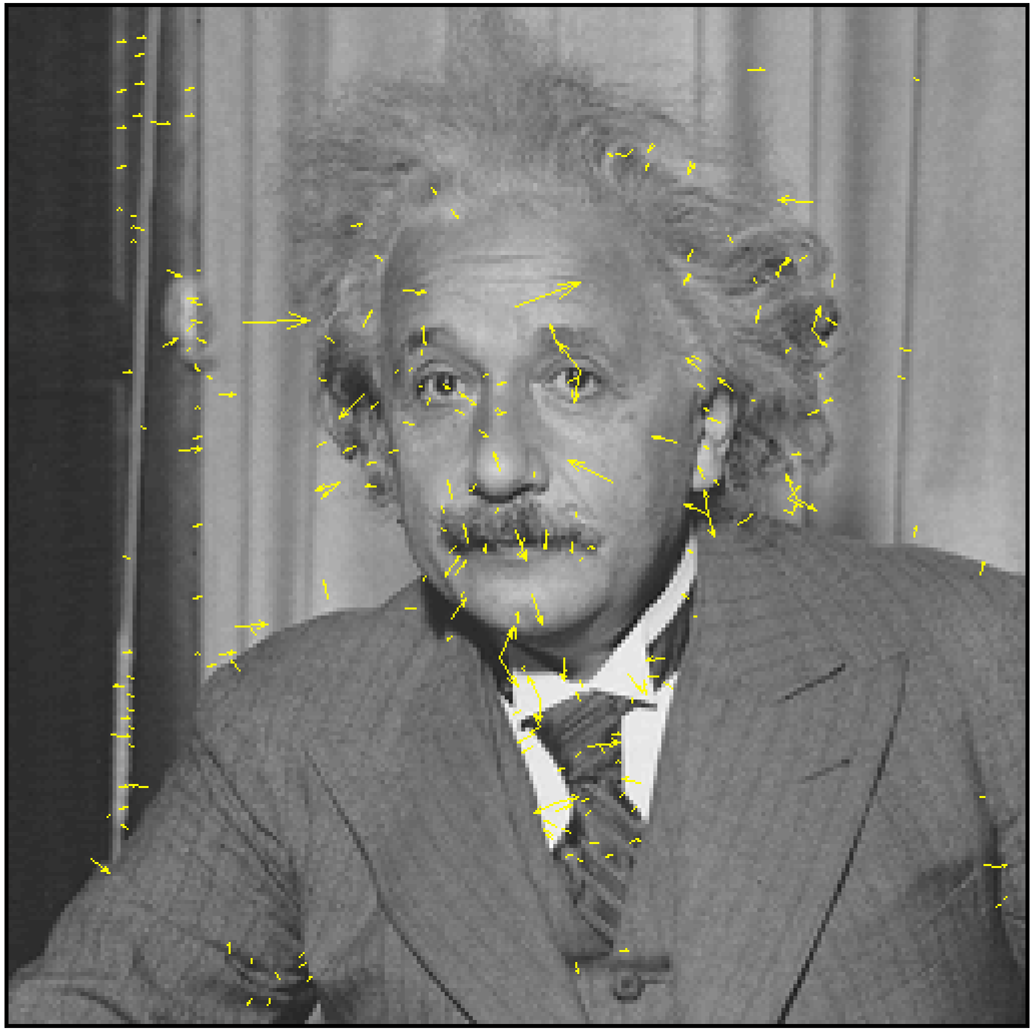
Remove low contrast



Remove edges



SIFT descriptor



Estimated rotation

- Computed affine transformation from rotated image to original image:

0.7060	-0.7052	128.4230
0.7057	0.7100	-128.9491
0	0	1.0000

- Actual transformation from rotated image to original image:

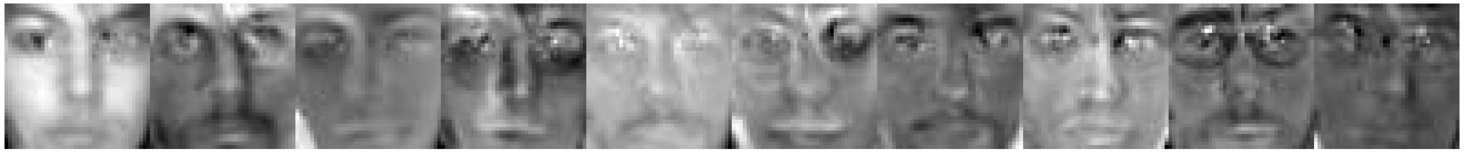
0.7071	-0.7071	128.6934
0.7071	0.7071	-128.6934
0	0	1.0000

SIFT extensions

PCA

Average face:

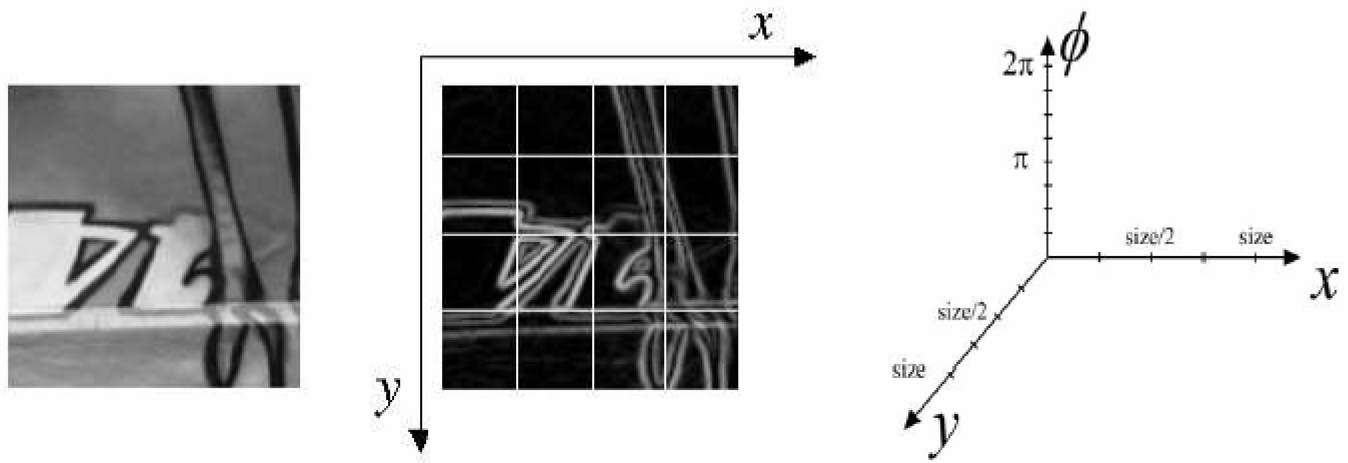
Top ten eigenfaces (left = highest eigenvalue, right = lowest eigenvalue):

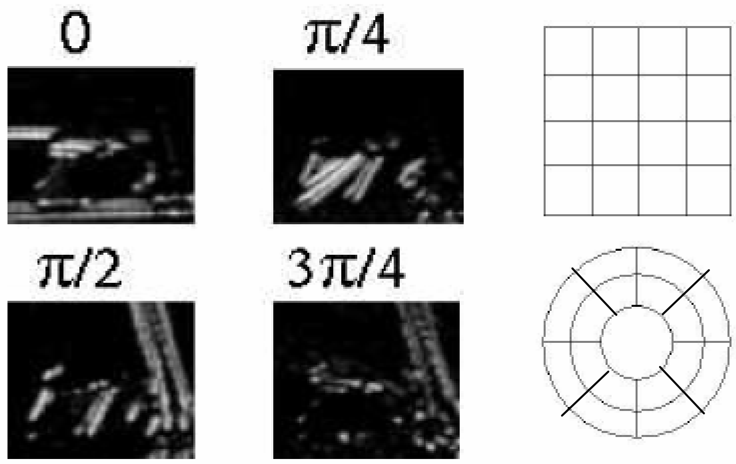


PCA-SIFT

- Only change step 4
- Pre-compute an eigen-space for local gradient patches of size 41x41
- 2x39x39=3042 elements
- Only keep 20 components
- A more compact descriptor

GLOH (Gradient location-orientation histogram)

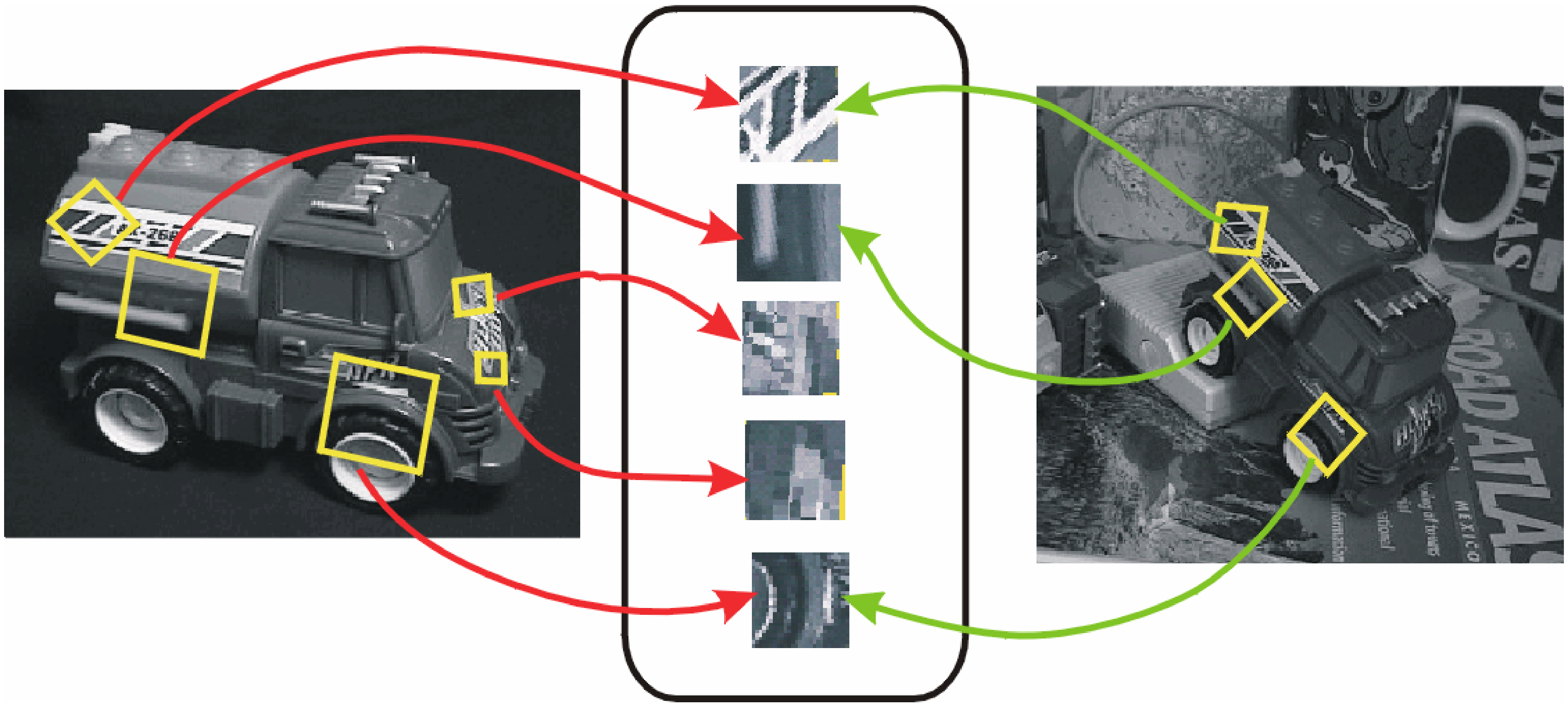




17 location bins
16 orientation bins
Analyze the 17x16=272-d eigen-space, keep 128 components

Applications

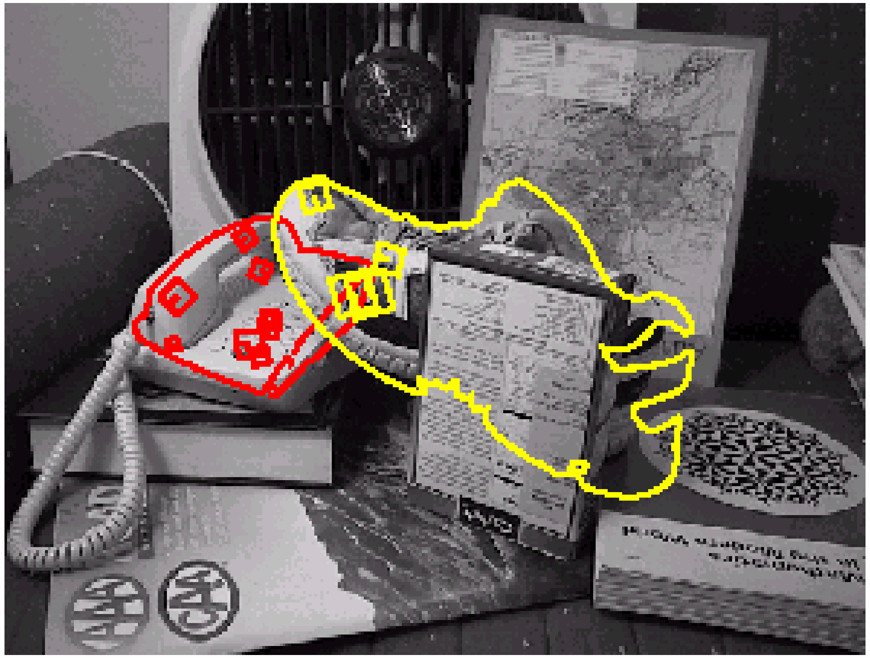
Recognition

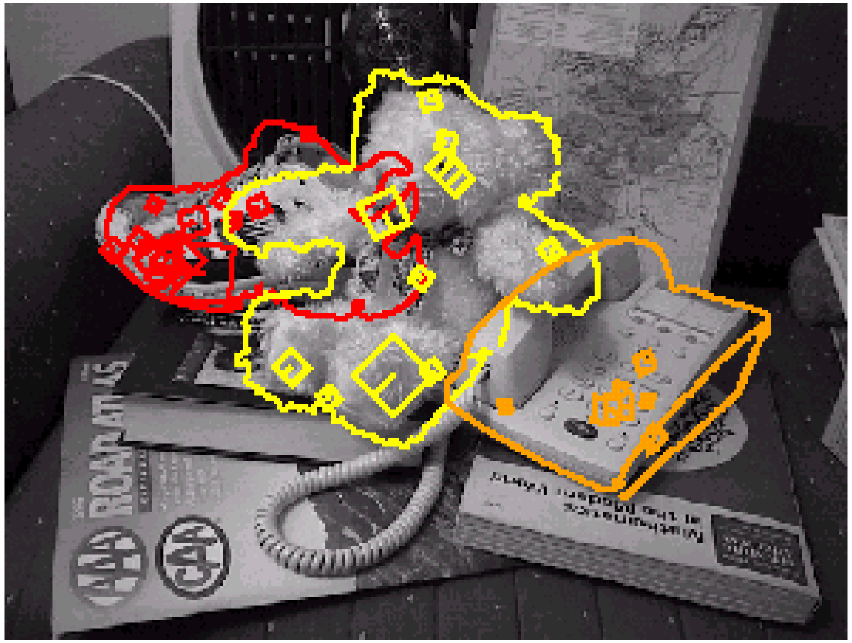


SIFT Features

3D object recognition

3D object recognition





Office of the past

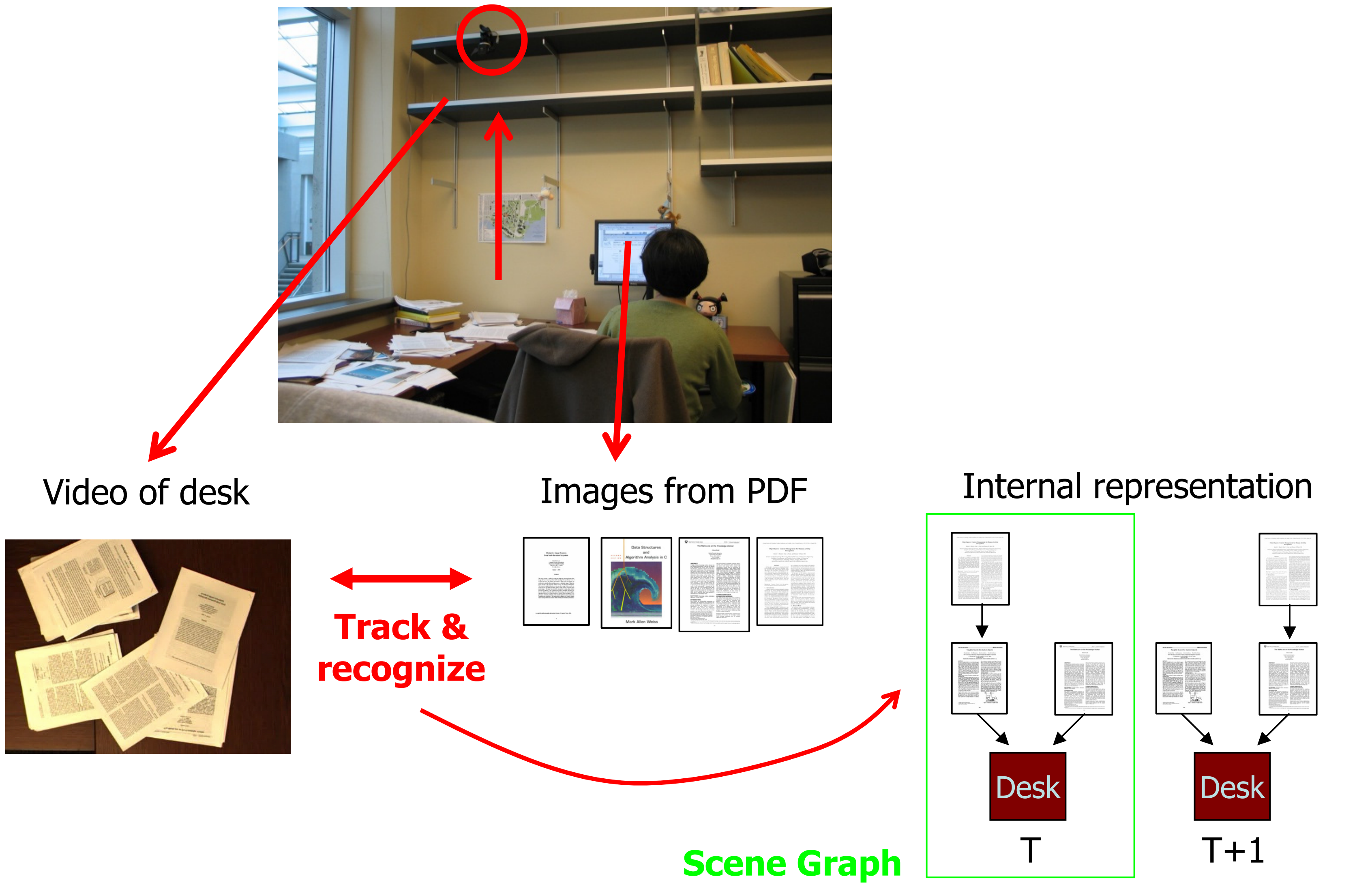
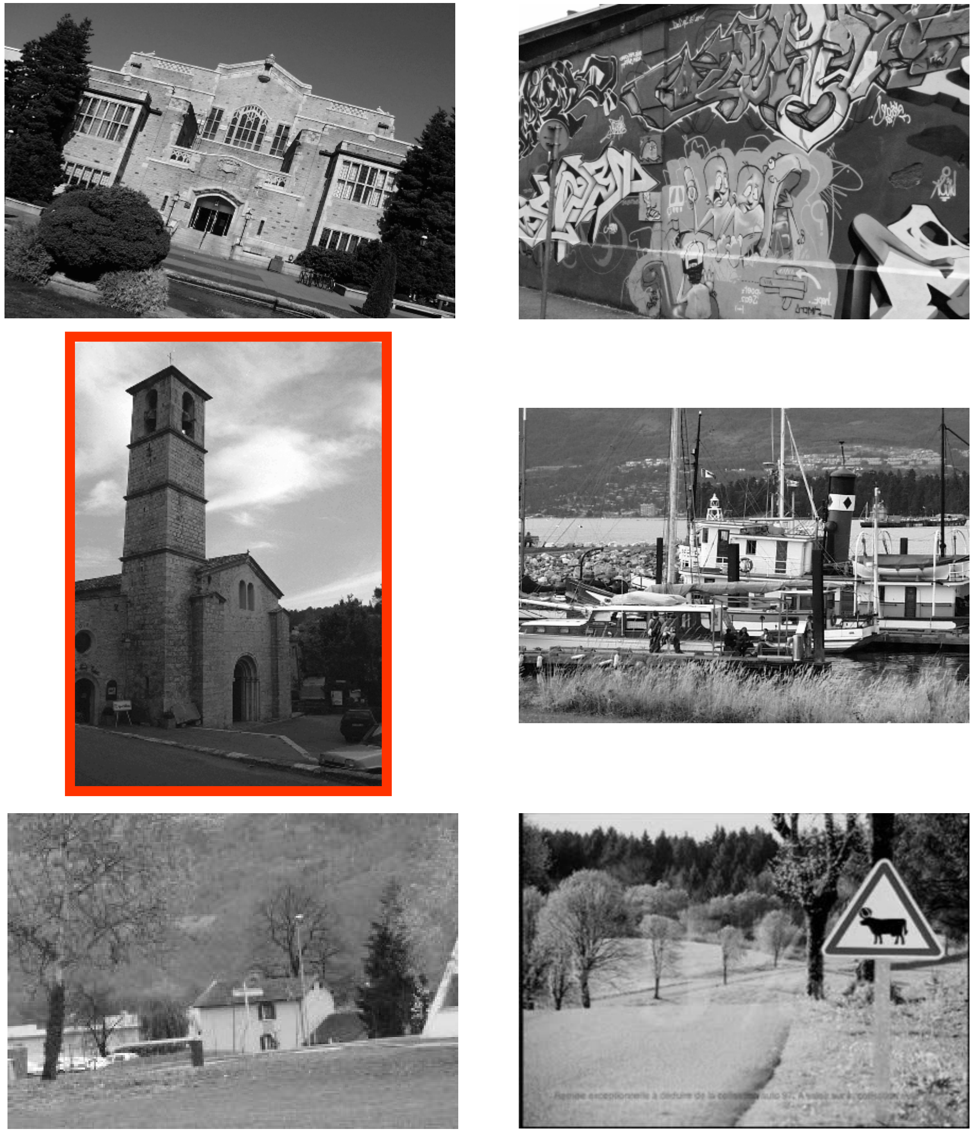


Image retrieval

change in viewing angle



• • •

> 5000 images

Image retrieval

22 correct matches

Image retrieval

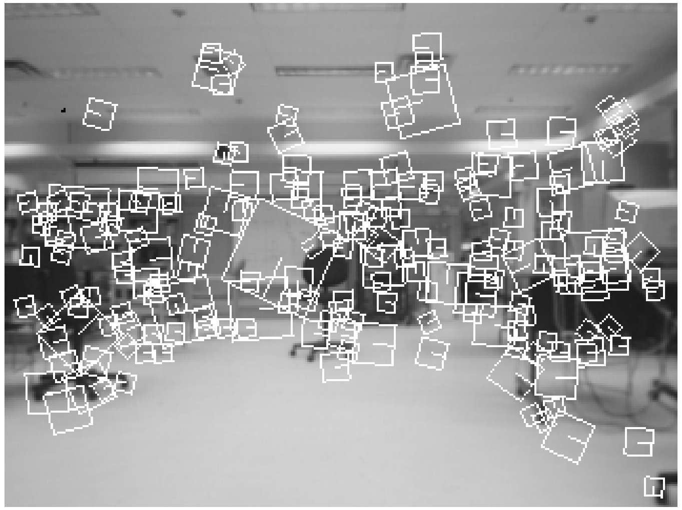
change in viewing angle

+ scale change

• • •

> 5000 images

Robot location

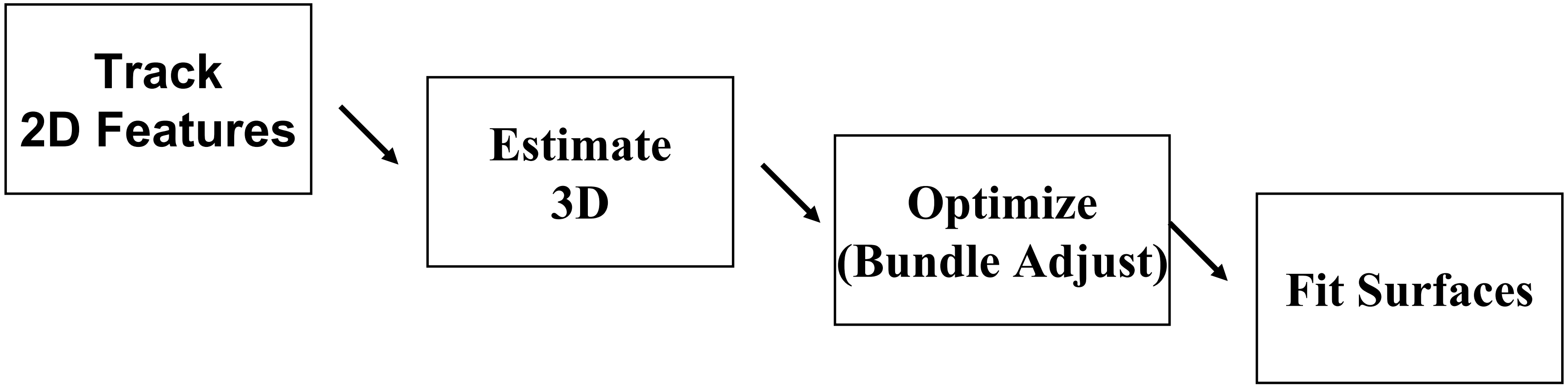


Robotics: Sony Aibo

SIFT is used for
- Recognizing charging station
- Communicating with visual cards
- Teaching object recognition
- soccer

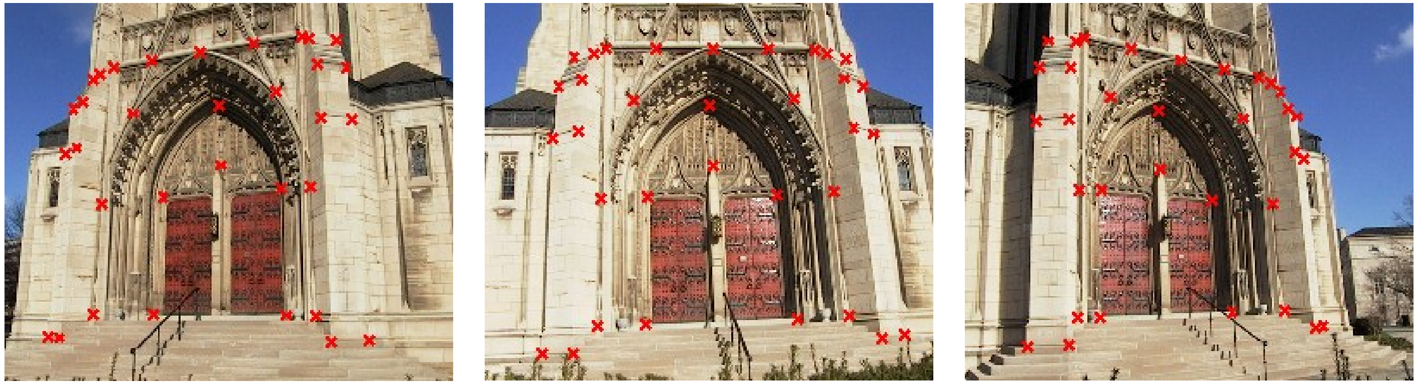
Structure from Motion

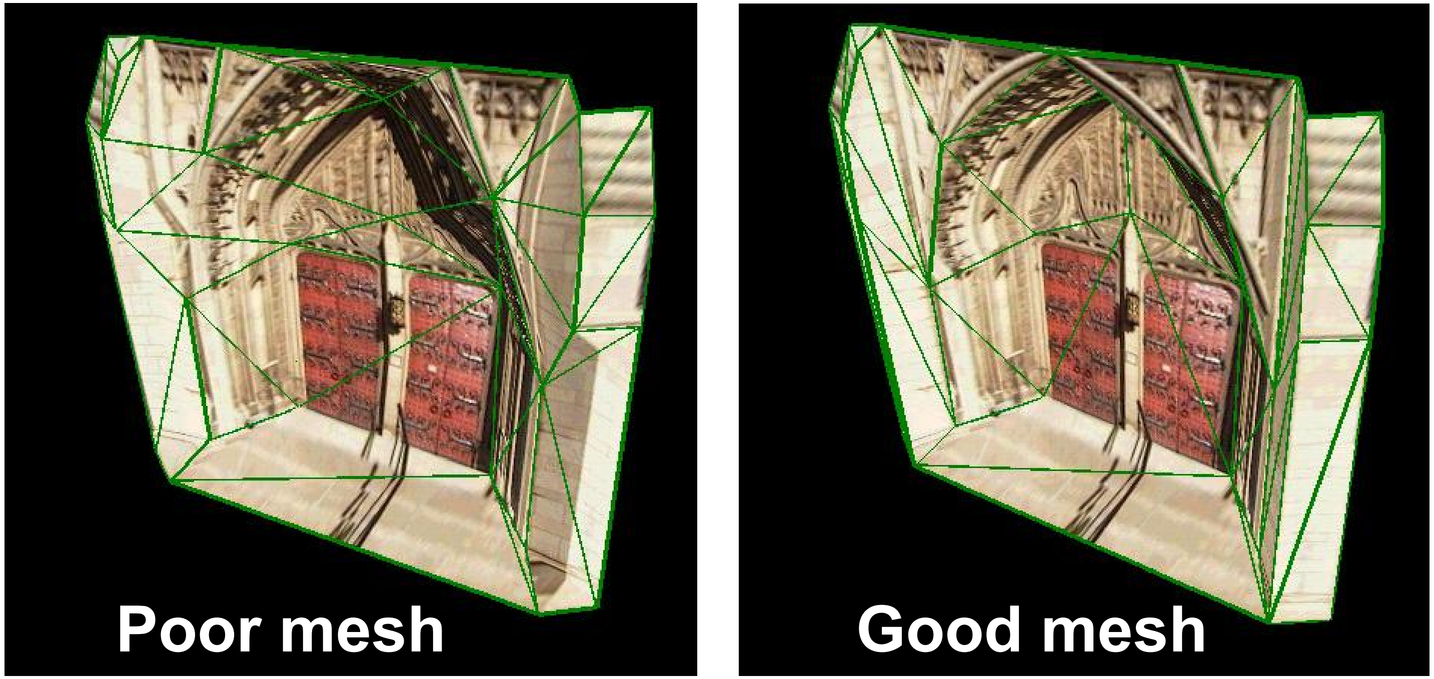
- The SFM Problem
  – Reconstruct scene geometry and camera motion from two or more images



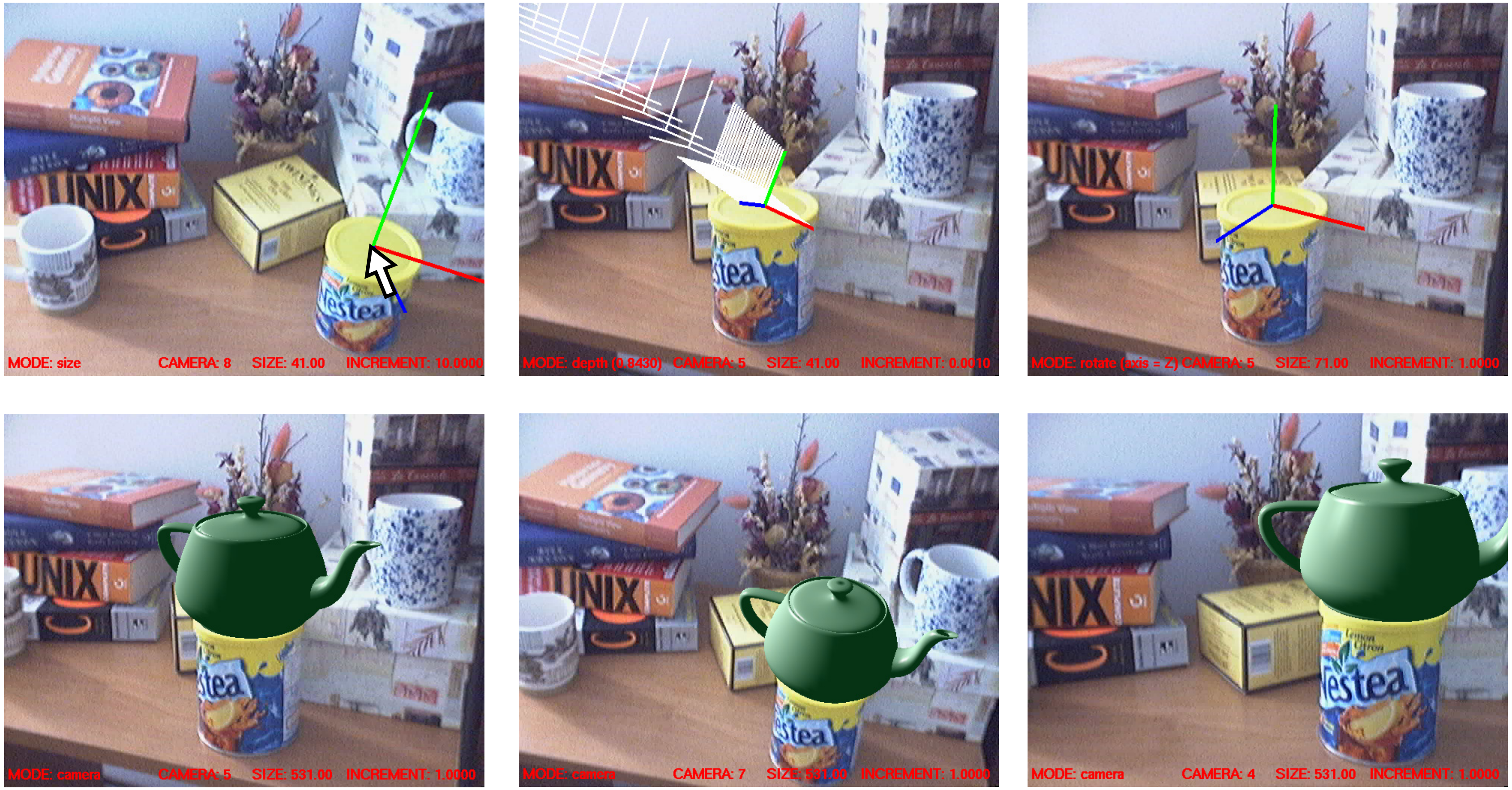
SFM Pipeline

Structure from Motion

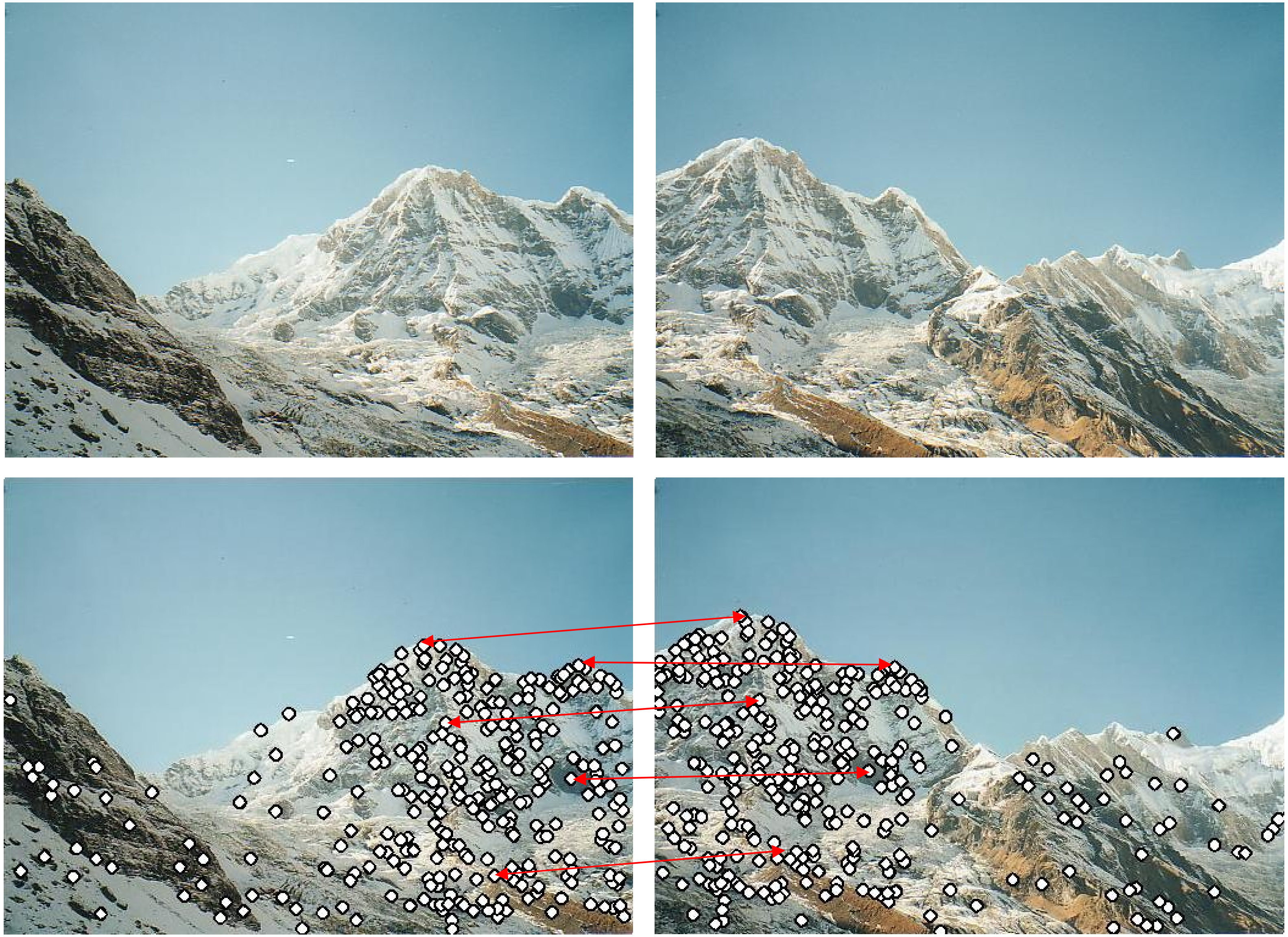




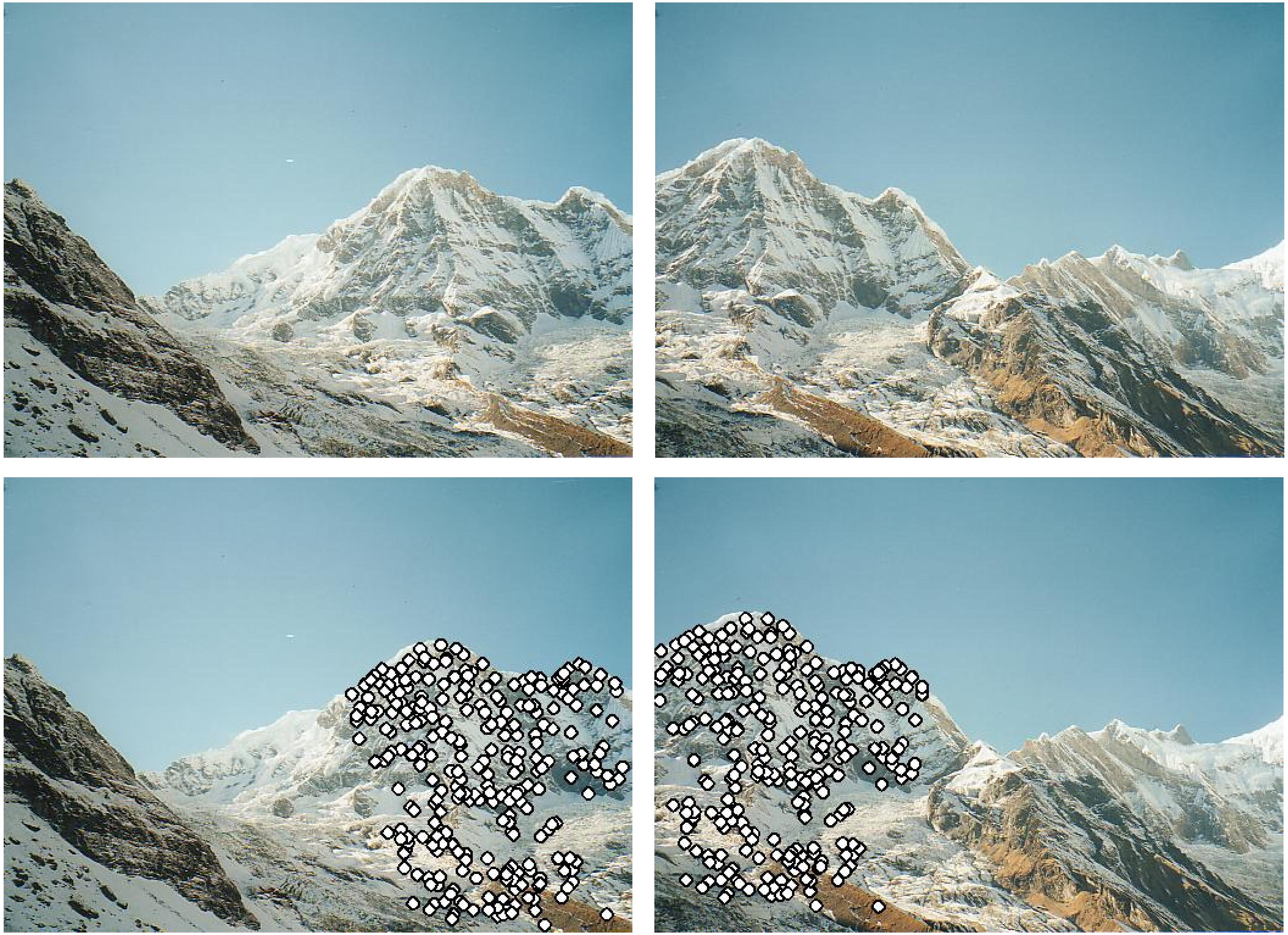
Augmented reality



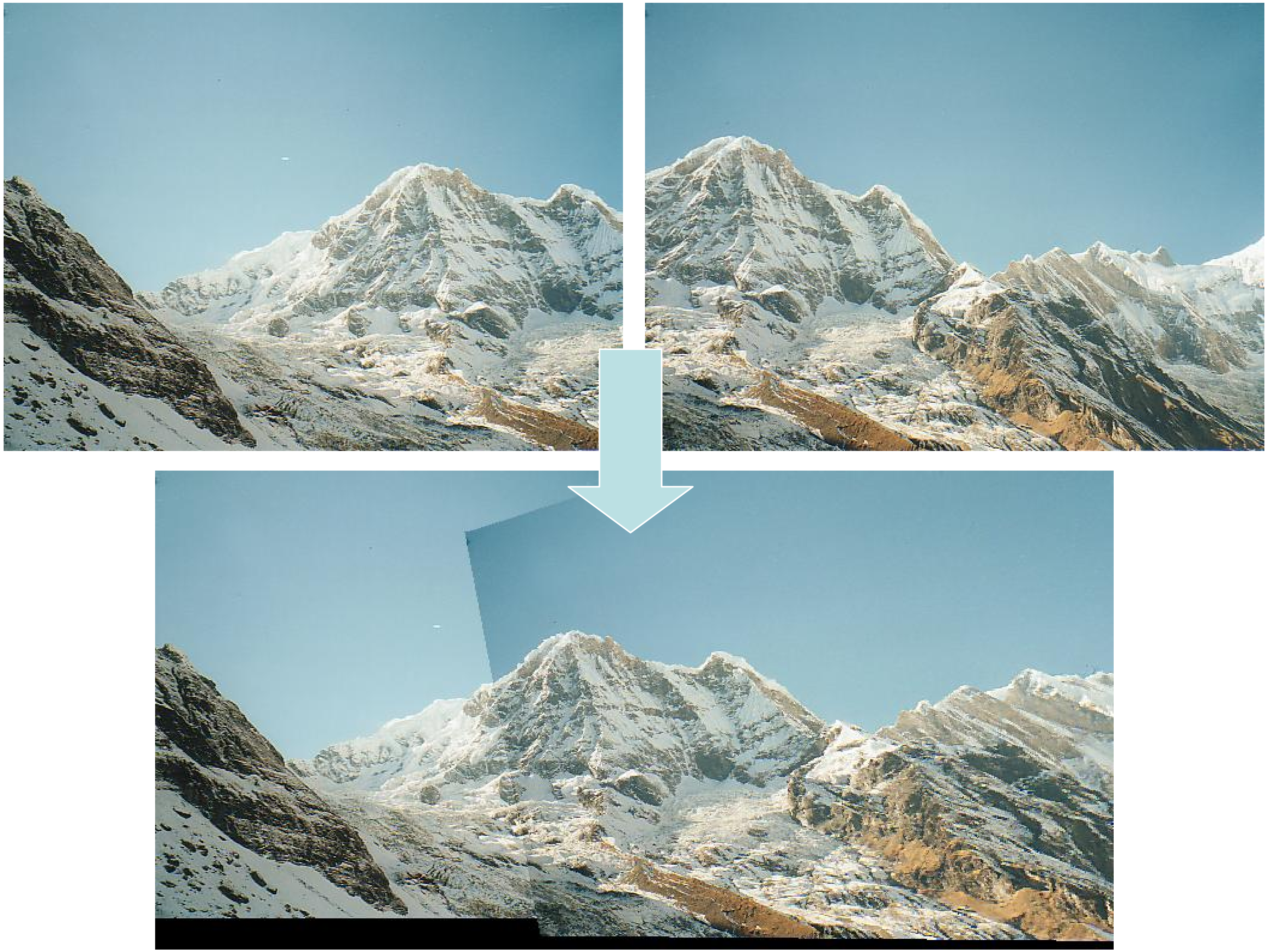
Automatic image stitching



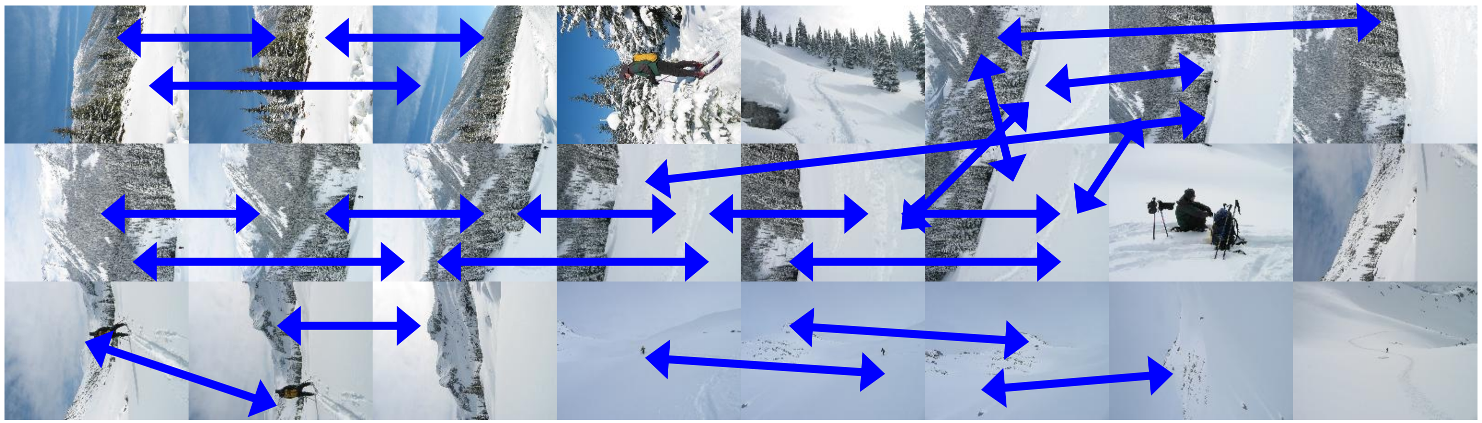
Automatic image stitching



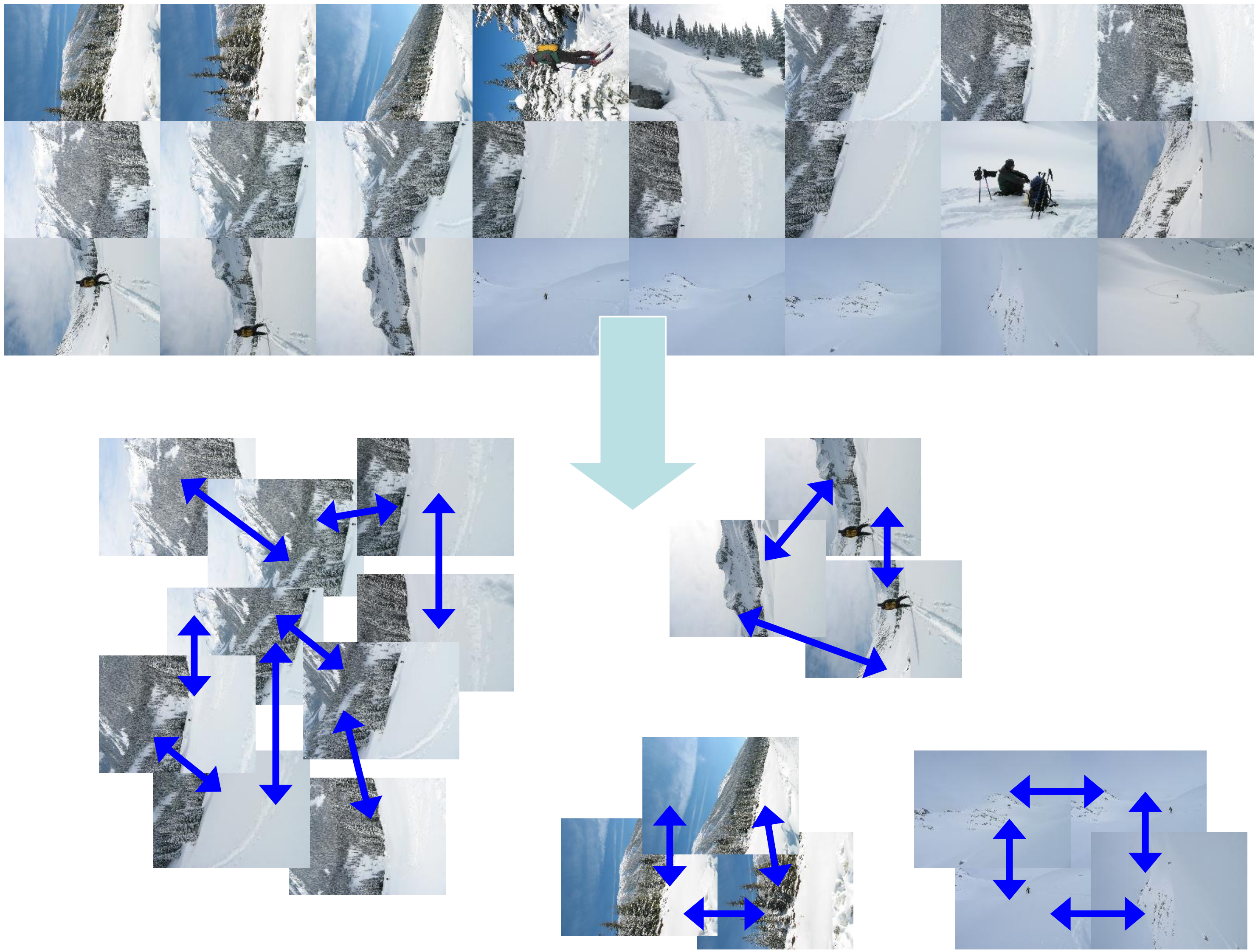
Automatic image stitching



Automatic image stitching



Automatic image stitching



Automatic image stitching

Reference

- Chris Harris, Mike Stephens, A Combined Corner and Edge Detector, 4th Alvey Vision Conference, 1988, pp147-151.
- David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 60(2), 2004, pp91-110.
- Yan Ke, Rahul Sukthankar, PCA-SIFT: A More Distinctive Representation for Local Image Descriptors, CVPR 2004.
- Krystian Mikolajczyk, Cordelia Schmid, A performance evaluation of local descriptors, Submitted to PAMI, 2004.
- SIFT Keypoint Detector, David Lowe.
- Matlab SIFT Tutorial, University of Toronto.