GNR602
Advanced Methods in Satellite Image Processing

Instructor: Prof. B. Krishna Mohan
CSRE, IIT Bombay
bkmohan@csre.iitb.ac.in

Slot 13

Lectures 14-15 Harris Corner Detector – HoG - SIFT

Feature Descriptors

1. Harris Corner Detector
2. Histogram of Oriented Gradients (HoG)
3. Scale Invariant Feature Transform (SIFT)

from Rick Szeliski's lecture notes,
and other sources…

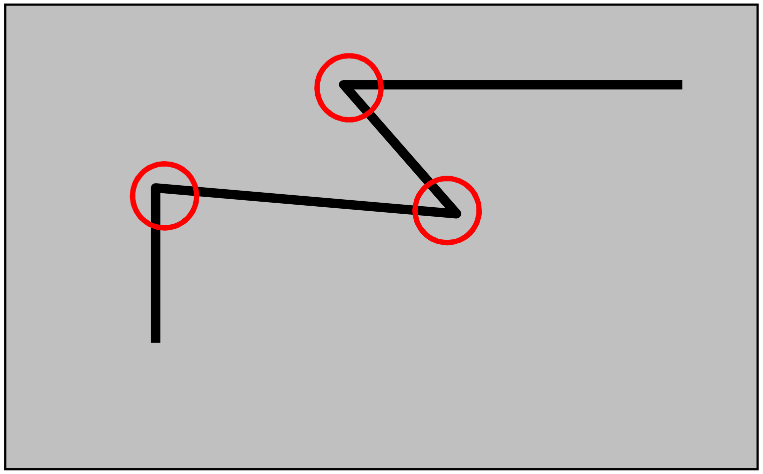
Feature Descriptors

- Images are recognized, matched using some key features that are perceived by humans invariant to rotation, translation, scale, illumination conditions, …
- Some of these abilities are captured in feature descriptors with varying capabilities
- Prominent among them are Harris Corner Detector (Harris-Stevenson algorithm), Histogram of Oriented Gradients (Navneet Dalal and Triggs) and Scale Invariant Feature Transform (David Lowe)

Harris Corner Detector

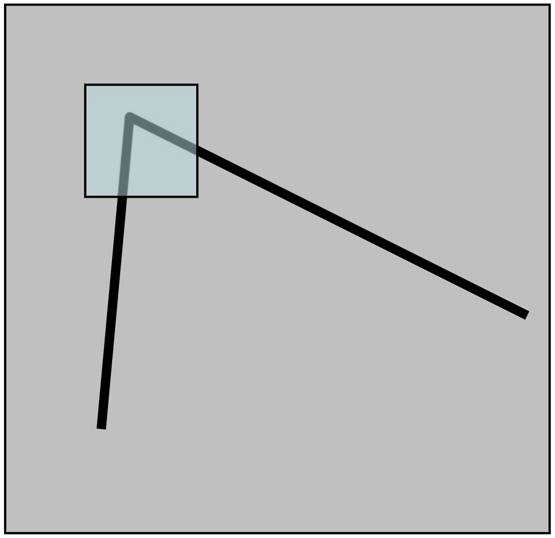
Harris corner detector

- C.Harris, M.Stephens. “A Combined Corner and Edge Detector”. 1988

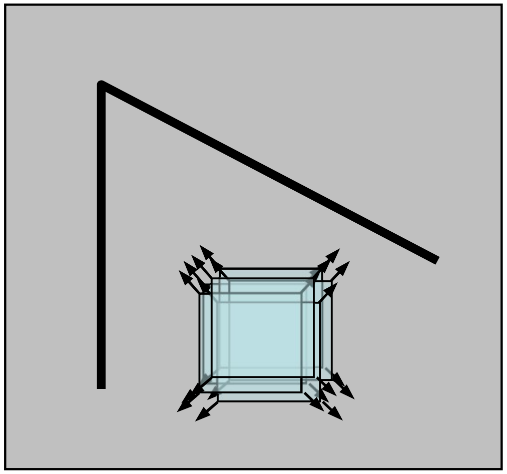


The Basic Idea

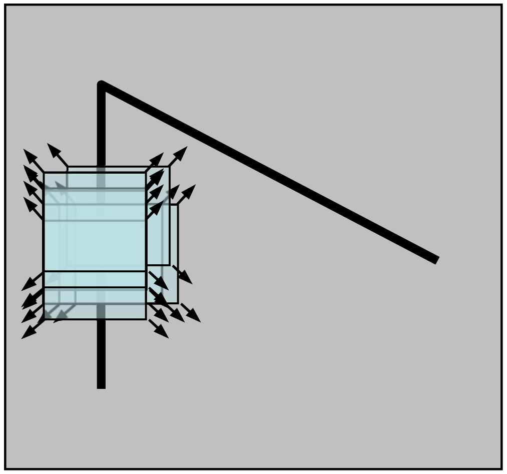
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



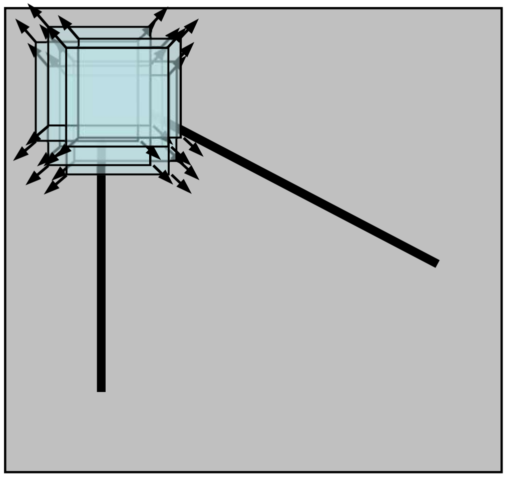
Harris Detector: Basic Idea



“flat” region:
no change in
all directions



“edge”:
no change along
the edge direction



“corner”:
significant change
in all directions

Harris Detector: Mathematics

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

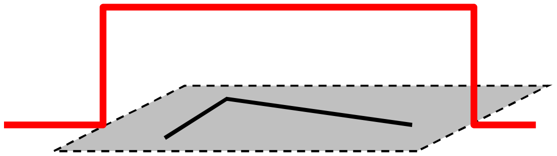
$$E(u,v) = \sum_{x,y} w(x,y) \left[I(x+u, y+v) - I(x,y) \right]^2$$

Window function

Shifted intensity

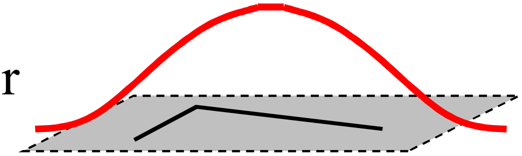
Intensity

Window function $w(x,y) =$



1 in window, 0 outside

or



Gaussian

Taylor series approximation to shifted image

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

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

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

Harris Detector: Mathematics

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

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

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

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

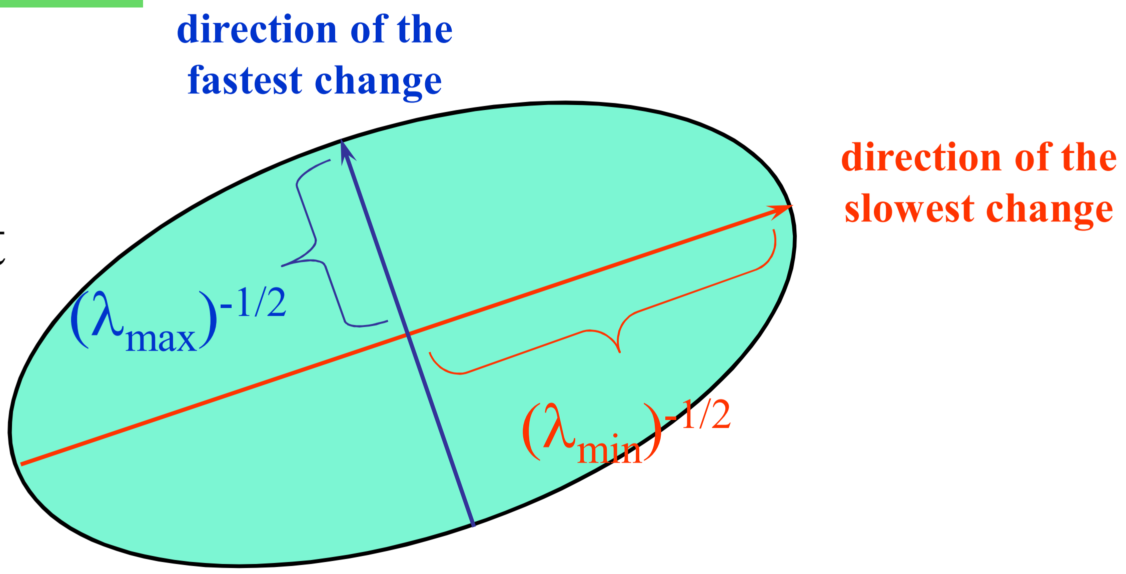
Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

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

λ_1, λ_2 – eigenvalues of M

Ellipse $E(u,v)$ = const



direction of the fastest change

direction of the slowest change

$(\lambda_{max})^{-1/2}$

$(\lambda_{min})^{-1/2}$

Example to show a polynomial as an ellipse

Let the polynomial be $4x^2 - 8x + y^2 + 4y = 8$

The standard form of an ellipse with centre at (p,q) and semi-axes given by a and b is:

$(x-p)^2 / a^2 + (y-q)^2 / b^2 = 1$

The above polynomial can be rewritten as:

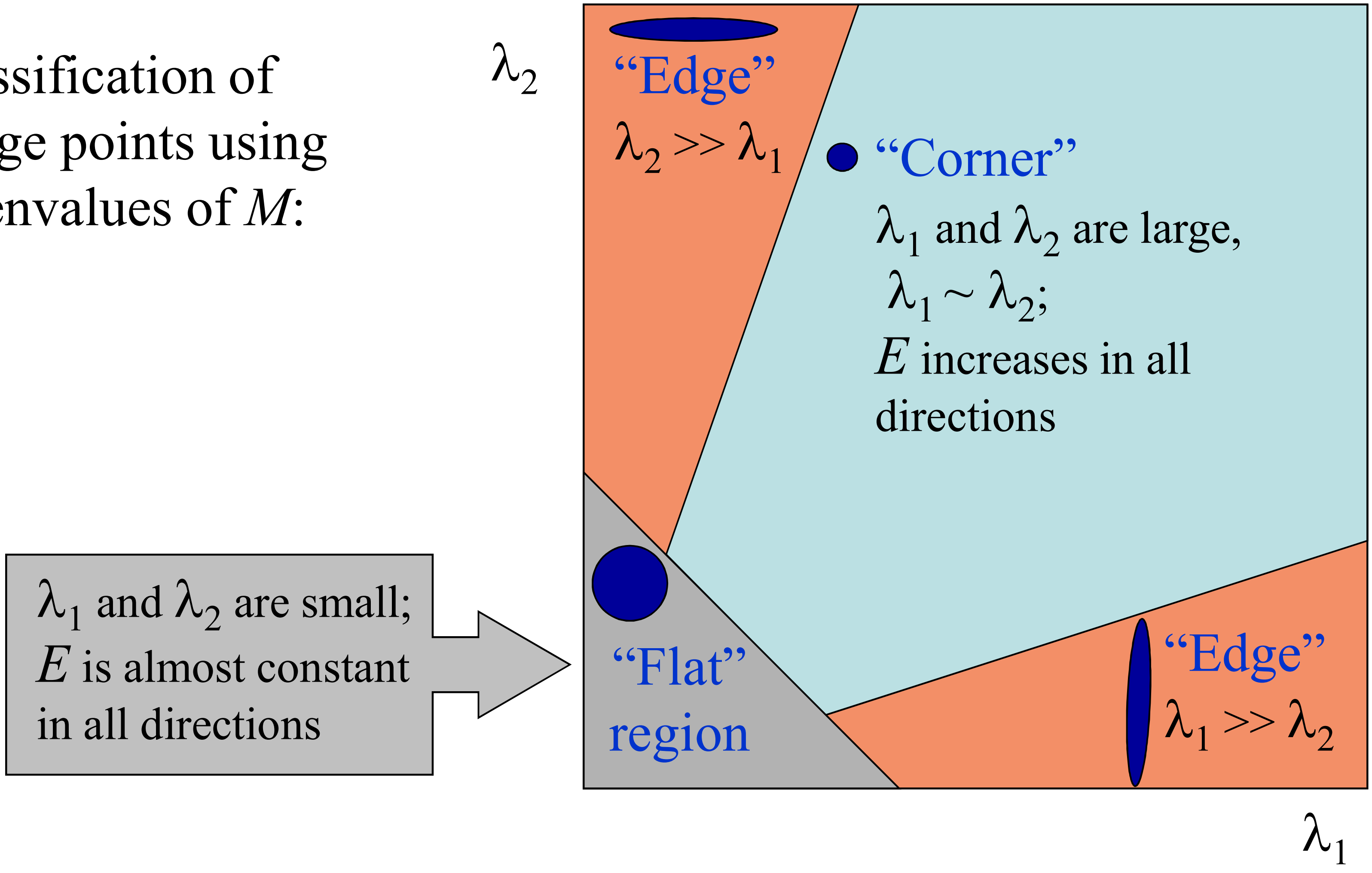
$4x^2 - 8x + 4 + y^2 + 4y + 4 = 8 + 4 + 4$

$4(x - 1)^2 + (y + 2)^2 = 16$ or

$(x - 1)^2 / 4 + (y + 2)^2 / 16 = 1$

Harris Detector: Threshold

Classification of image points using eigenvalues of *M*:



λ_2

"Edge"

$\lambda_2 >> \lambda_1$

"Corner"

λ_1 and λ_2 are large, $\lambda_1 \sim \lambda_2$; *E* increases in all directions

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

"Flat" region

"Edge"

$\lambda_1 >> \lambda_2$

λ_1

Harris Detector: Threshold

Measure of corner response:

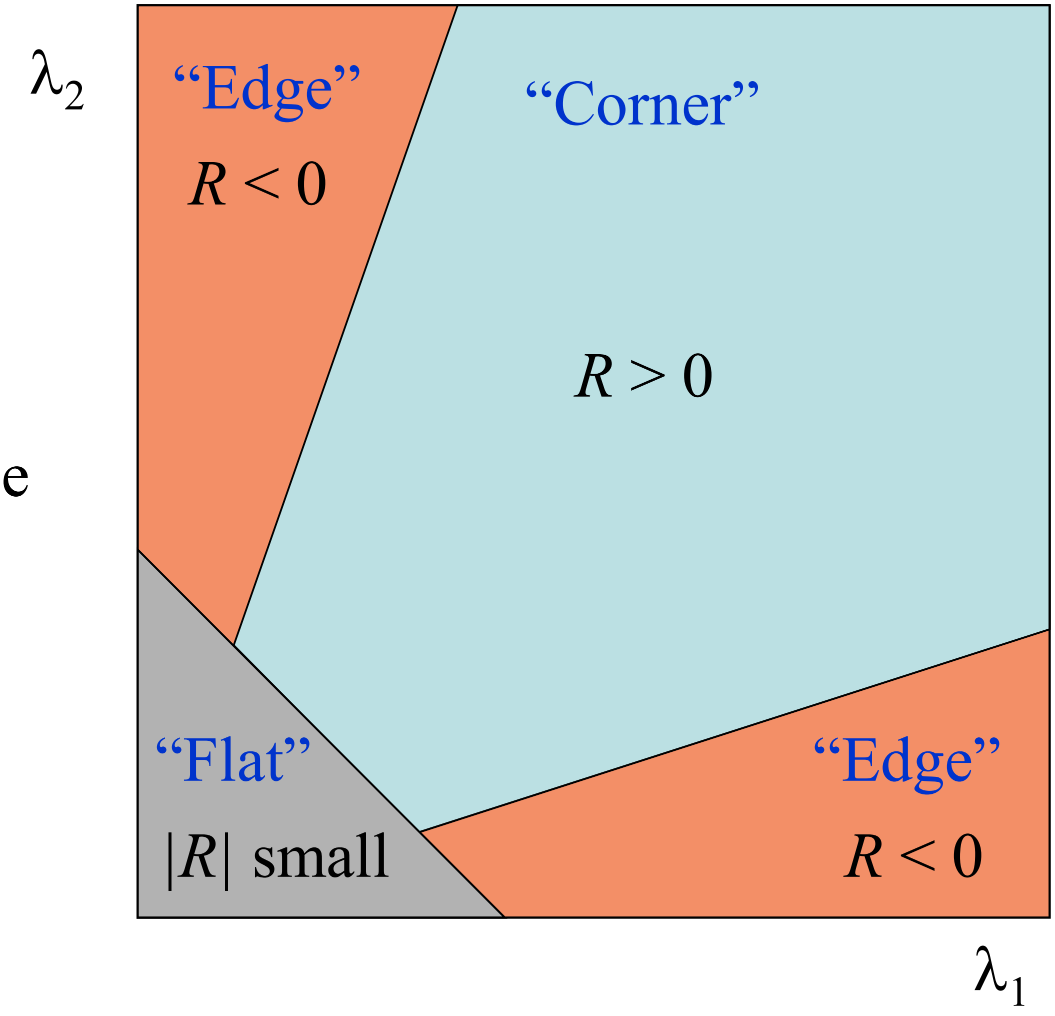
$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

$$\begin{aligned} \det M &= \lambda_1 \lambda_2 \\ \operatorname{trace} M &= \lambda_1 + \lambda_2 \end{aligned}$$

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

Harris Detector: Mathematics

- 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



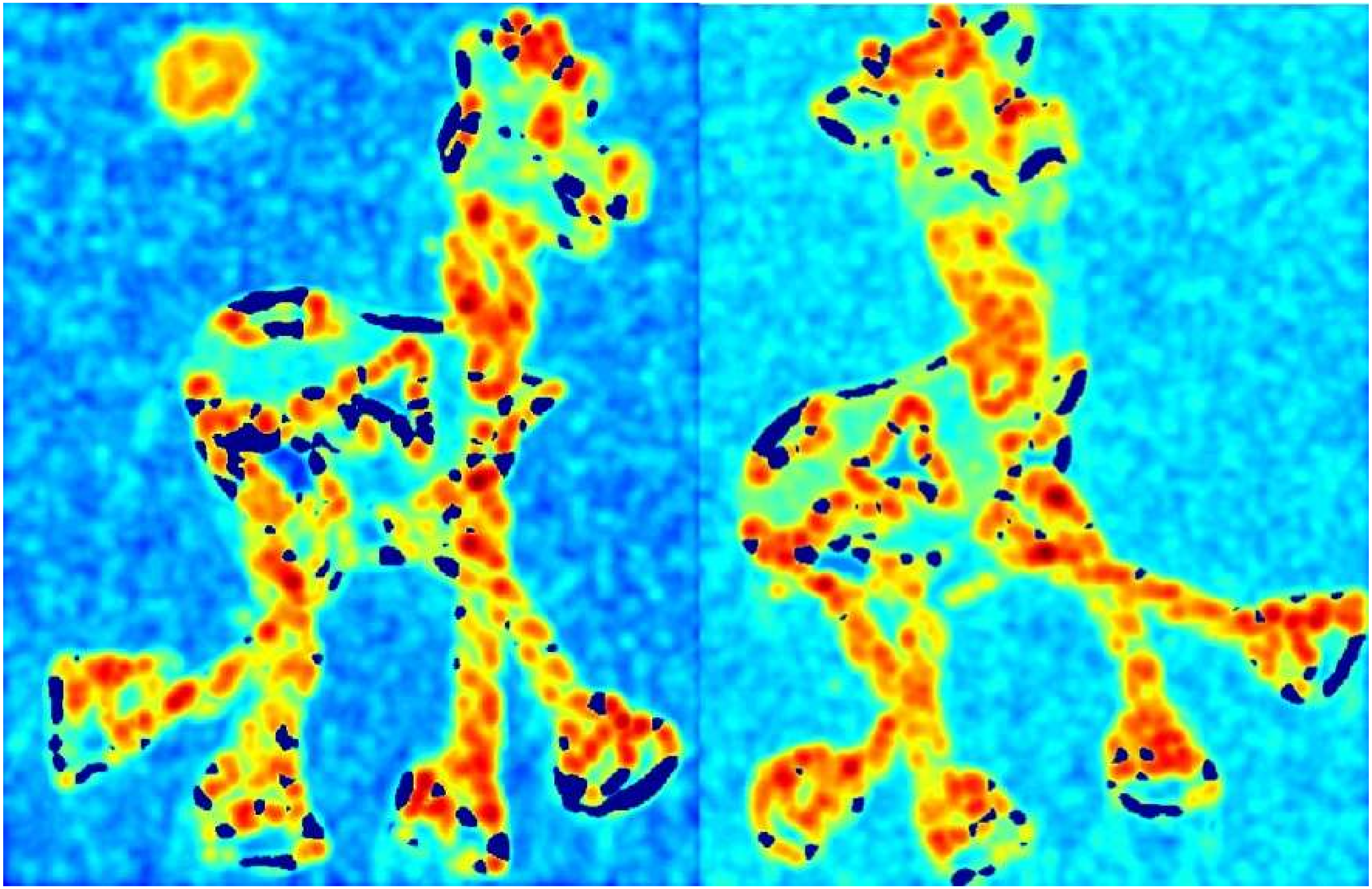
Harris Detector

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

Harris Detector: Workflow

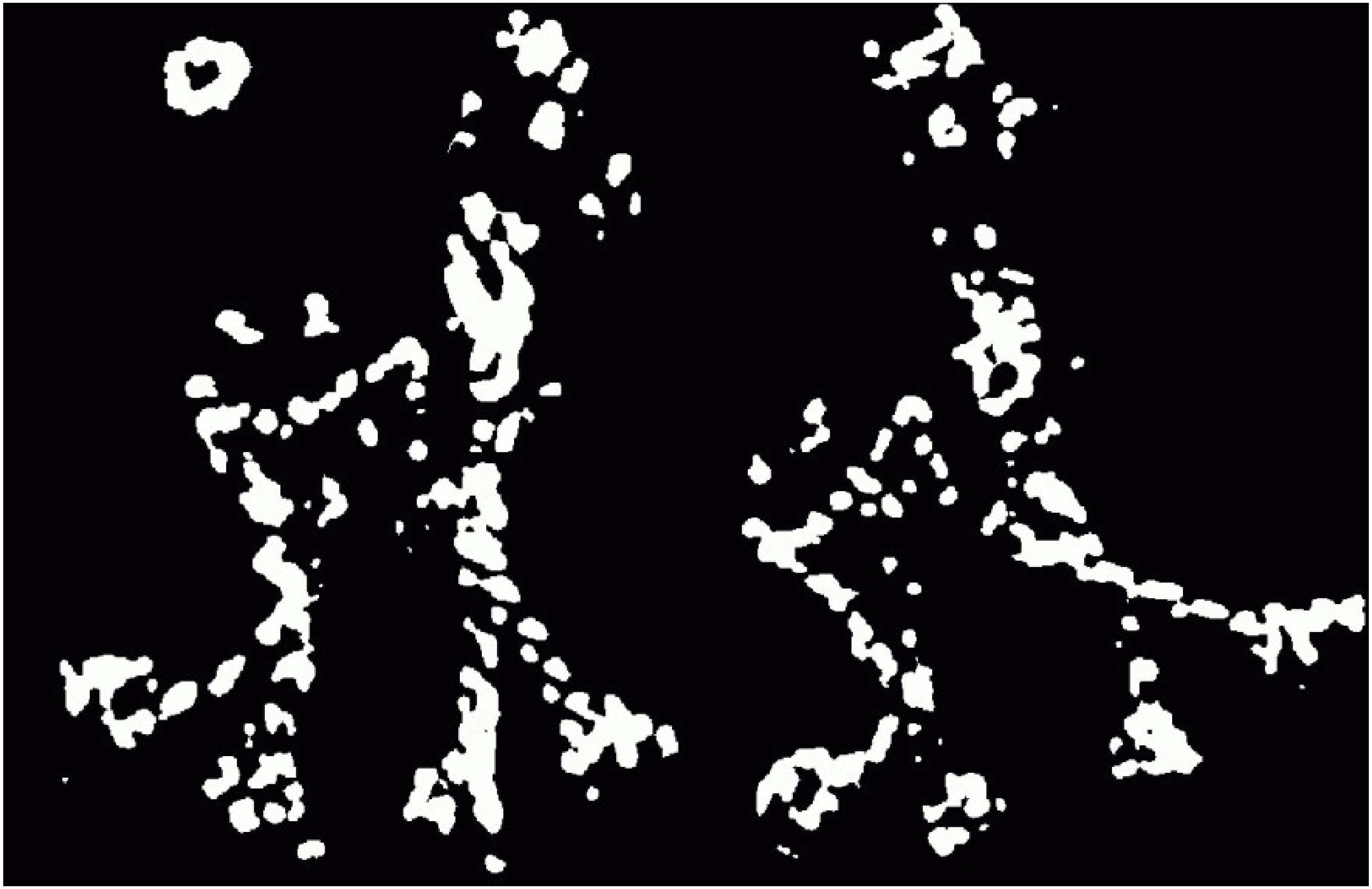
Harris Detector: Workflow

Compute corner response R



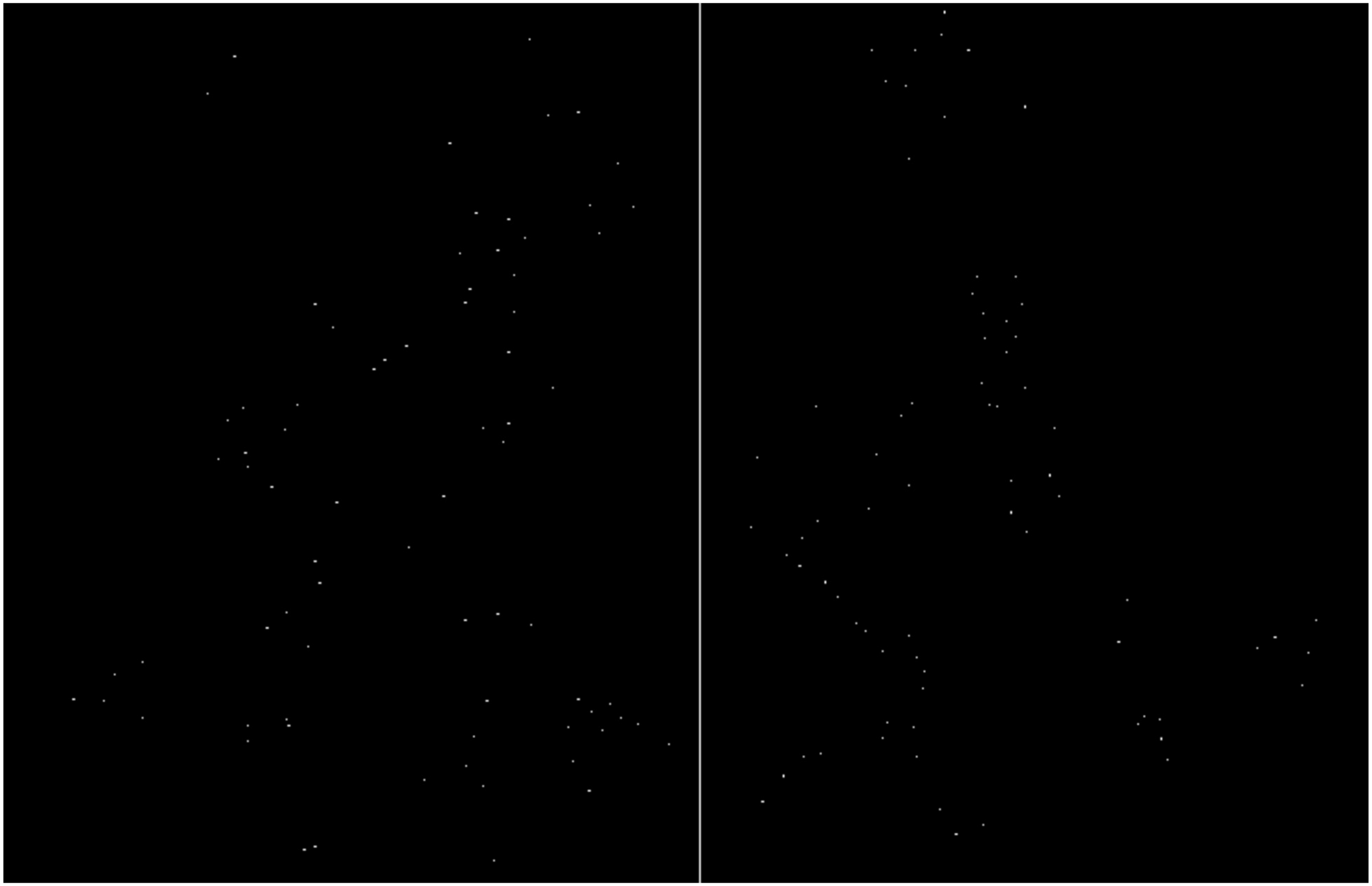
Harris Detector: Workflow

Find points with large corner response: R>threshold

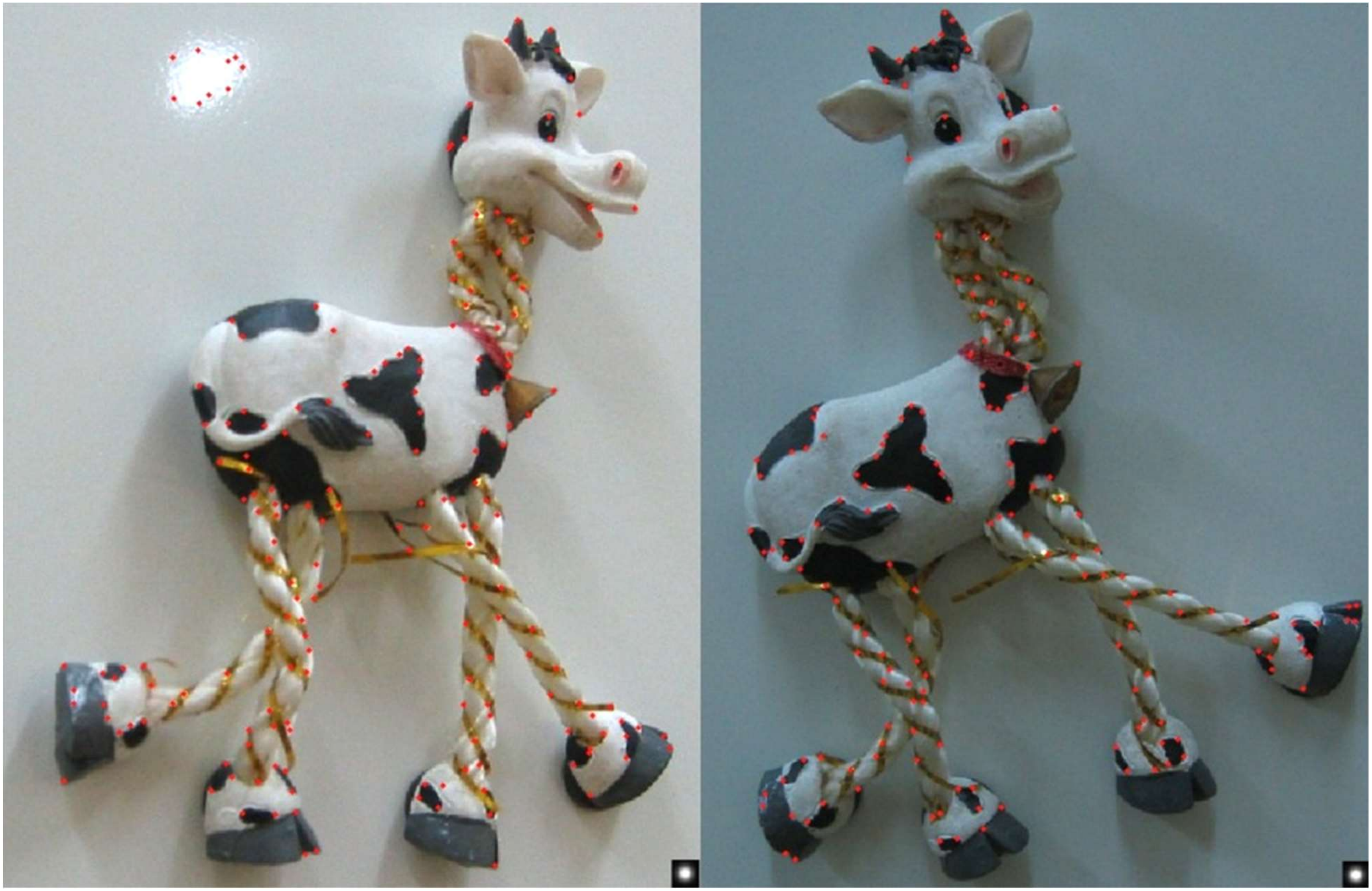


Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow



Harris Detector: Summary

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

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

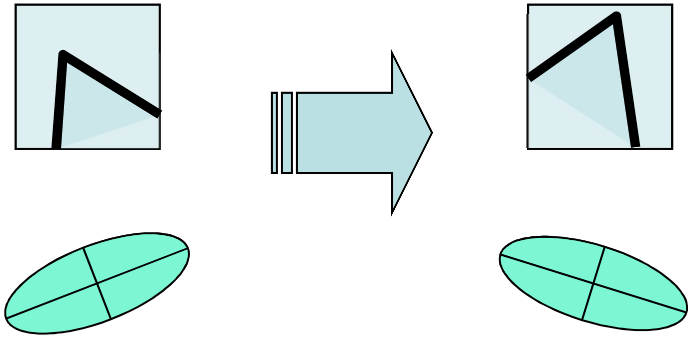
- Describe a point in terms of eigenvalues of *M*: *measure of corner response*

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

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

Harris Detector: Some Properties

- Rotation invariance



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

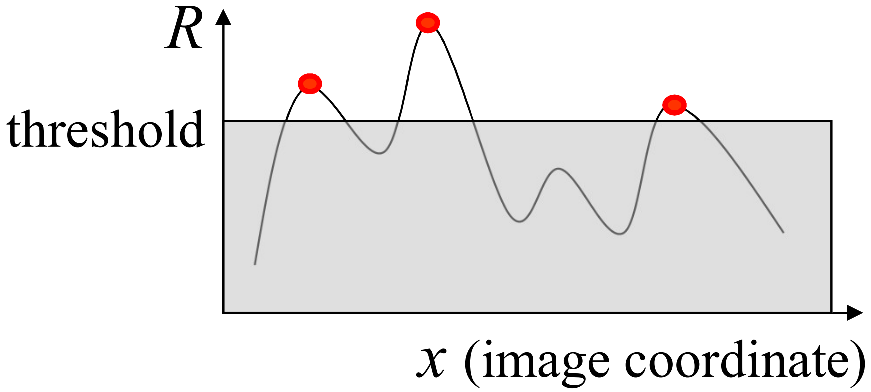
Corner response R is invariant to image rotation

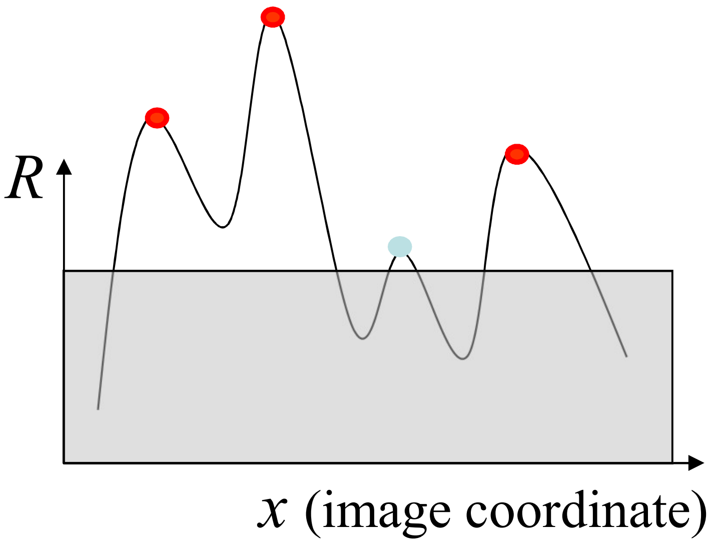
Harris Detector: Some Properties

- Partial invariance to *affine intensity* change

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

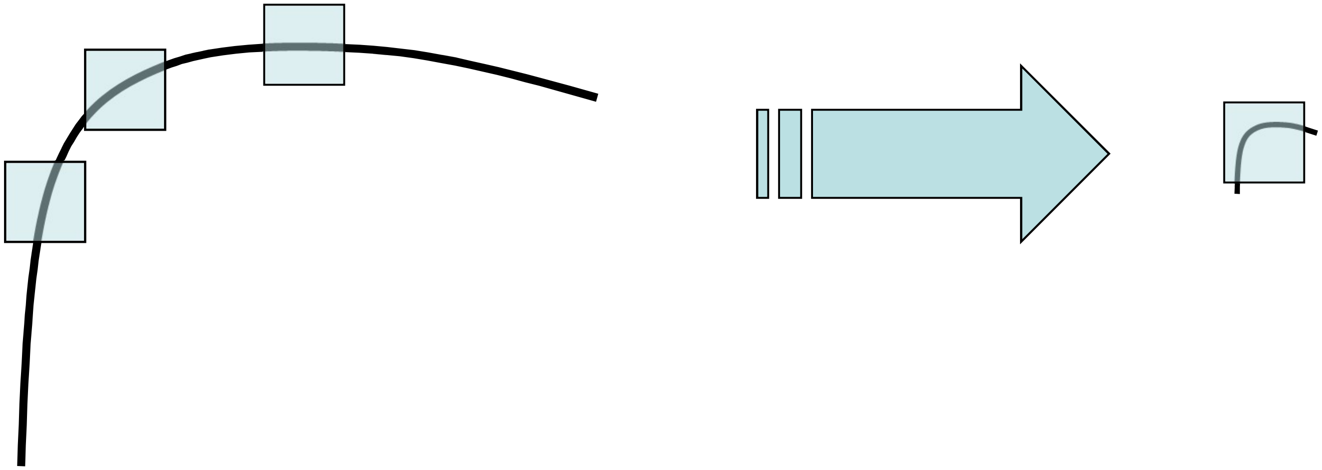
✓ Intensity scale: $I \rightarrow a\, I$





Harris Detector: Some Properties

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



All points will be classified as edges

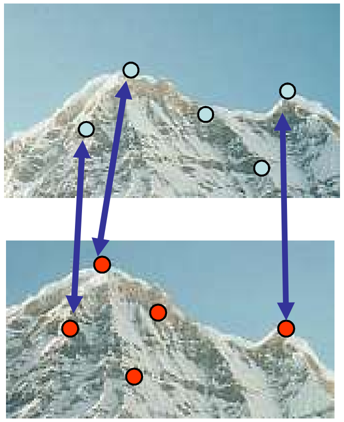
Corner !

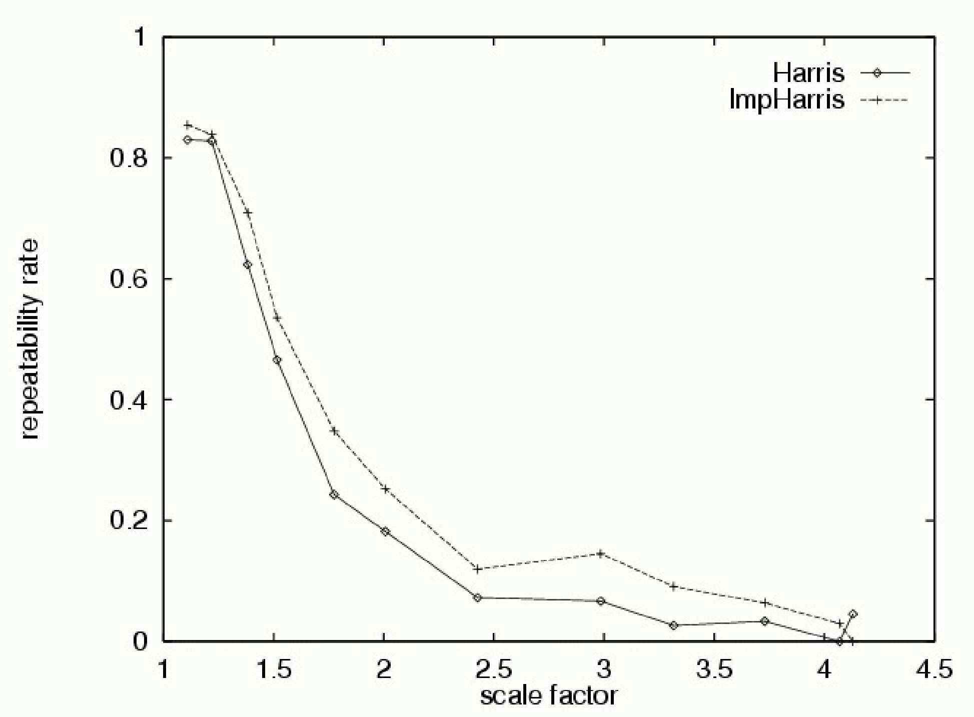
Harris Detector: Some Properties

- Quality of Harris detector for different scale changes

Repeatability rate:

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

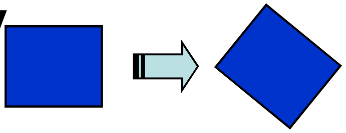




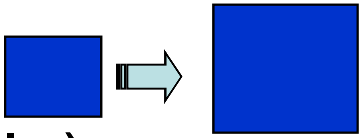
C.Schmid et.al. “Evaluation of Interest Point Detectors”. IJCV 2000

Models of Image Change

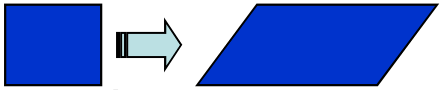
- Geometry



  - Rotation

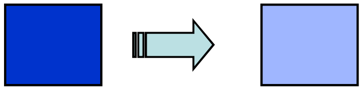


  - Similarity (rotation + uniform scale)



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

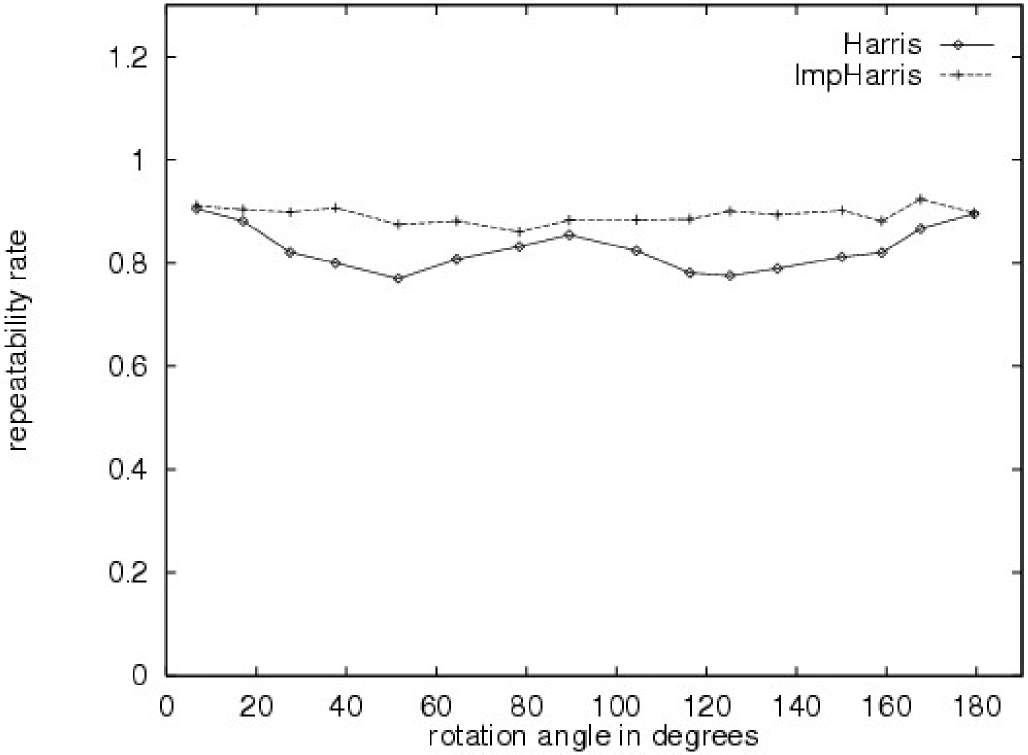
- Photometry



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

Rotation Invariant Detection

- Harris Corner Detector



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

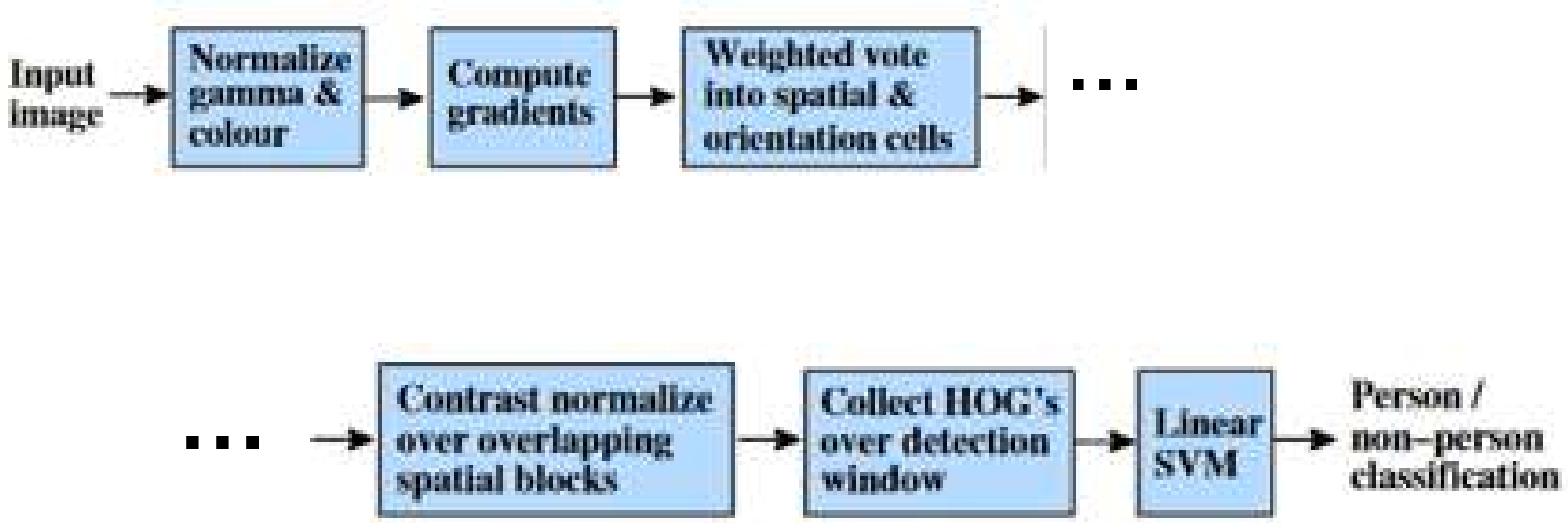
Histogram of Oriented Gradients (HoG)

What is HOG? (Histograms of Oriented Gradients)

HOG is an edge orientation histogram based descriptor, based on the orientation of the gradient in localized region that is called cells.

Therefore, it is easy to express the rough shape of the object and is robust to variations in geometry and illumination changes.

On the other hand, rotation and scale changes are not supported.



Input image

Normalize gamma & colour

Compute gradients

Weighted vote into spatial & orientation cells

...

...

Contrast normalize over overlapping spatial blocks

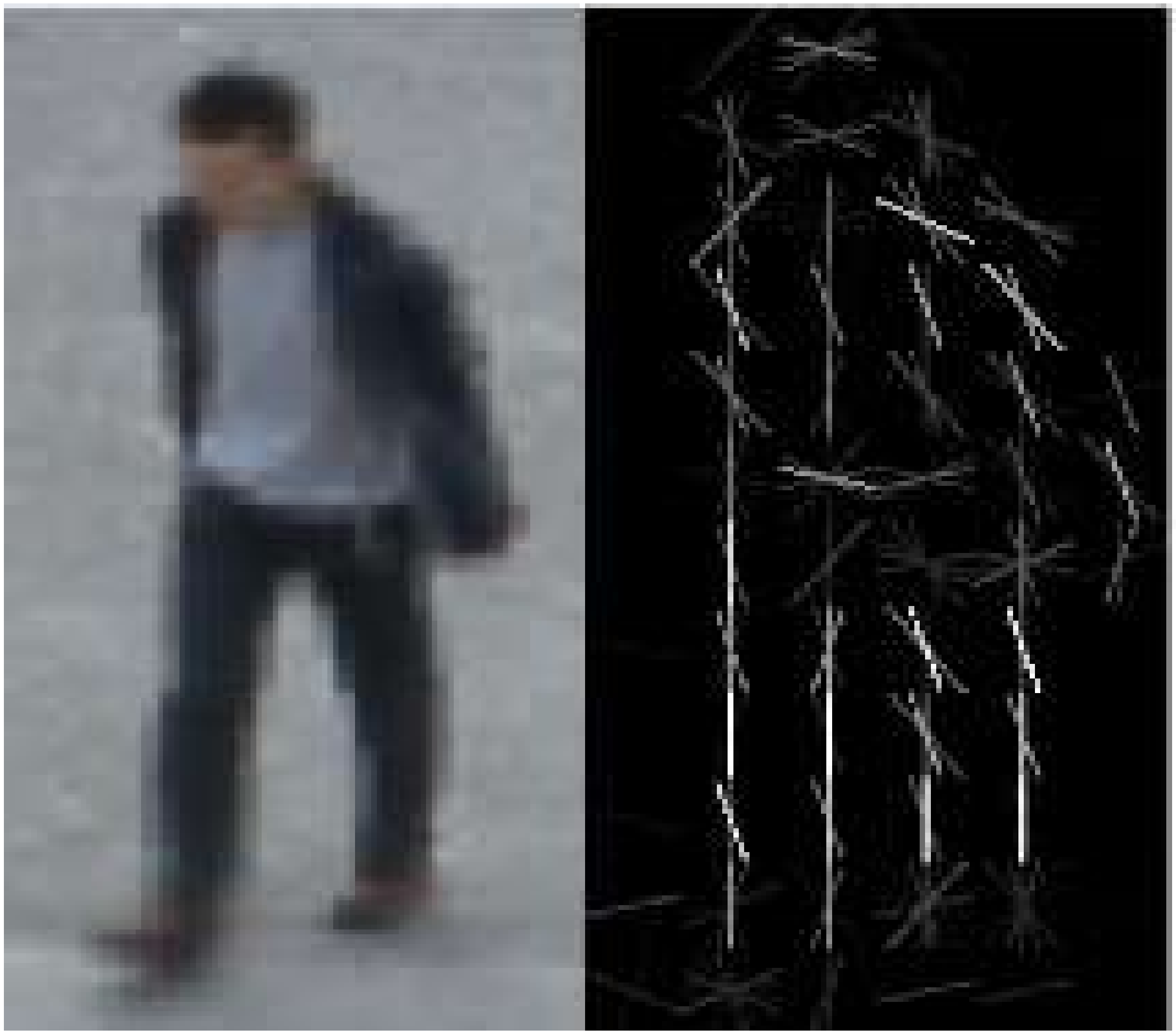
Collect HOG's over detection window

Linear SVM

Person / non–person classification

Overall Scheme for HoG based Person-Non-Person Classification

HOG image



HOG feature extraction algorithm

1. The color image is converted to grayscale
2. the luminance gradient is calculated at each pixel
3. To create a histogram of gradient orientations for each cell.
   - Feature quantity becomes robust to changes of form
4. Normalization and Descriptor Blocks
   - Feature quantity becomes robust to changes in illumination

HOG feature extraction algorithm(1)

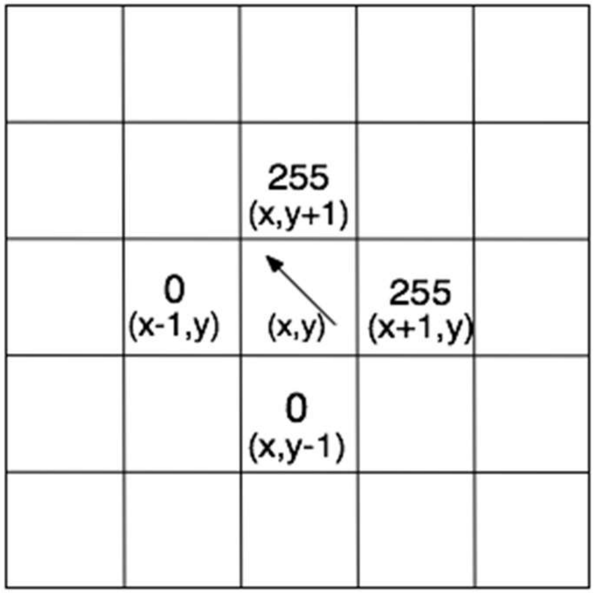
2. The luminance gradient is calculated at each pixel
   - The luminance gradient is a vector with magnitude m and orientation θ represented by the change in the luminance.

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

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

$$-\frac{\Pi}{2}<\theta<\frac{\Pi}{2}$$

※L is the luminance value of pixel

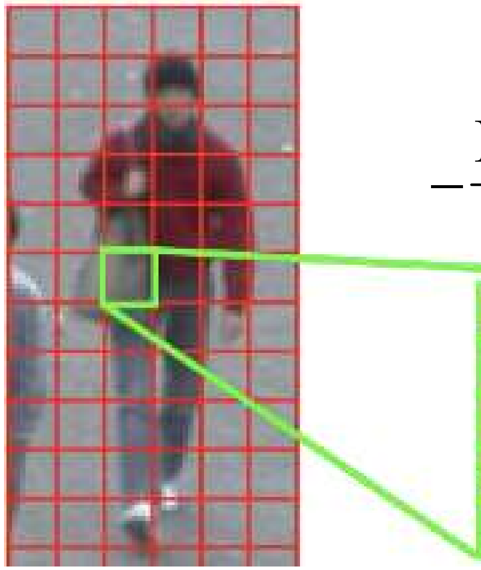


HOG feature extraction algorithm(2)

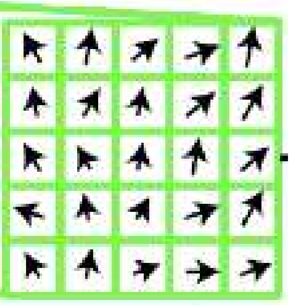
3. To create a histogram of gradient orientations for each cell(5 × 5pixel) using the gradient magnitude and orientation of the calculated.

   - The orientation bins are evenly spaced over 0° – 180° and are provided by nine of 20° . By adding the magnitude of the luminance gradient for each orientation, generate a histogram.

Image size = 60x30

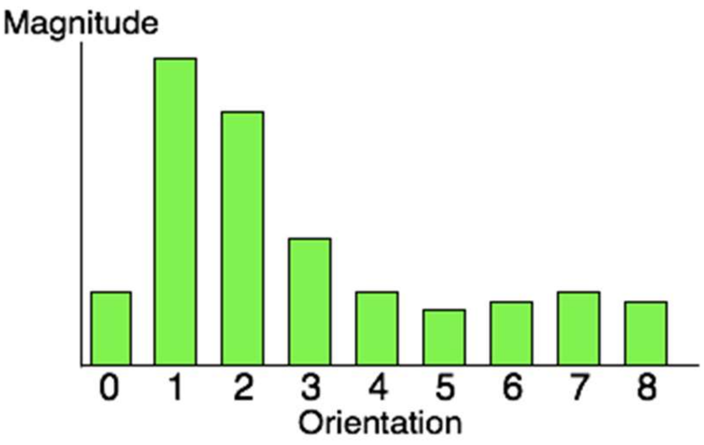


$$-\frac{\Pi}{2} < \theta < \frac{\Pi}{2}$$



Orientation num is

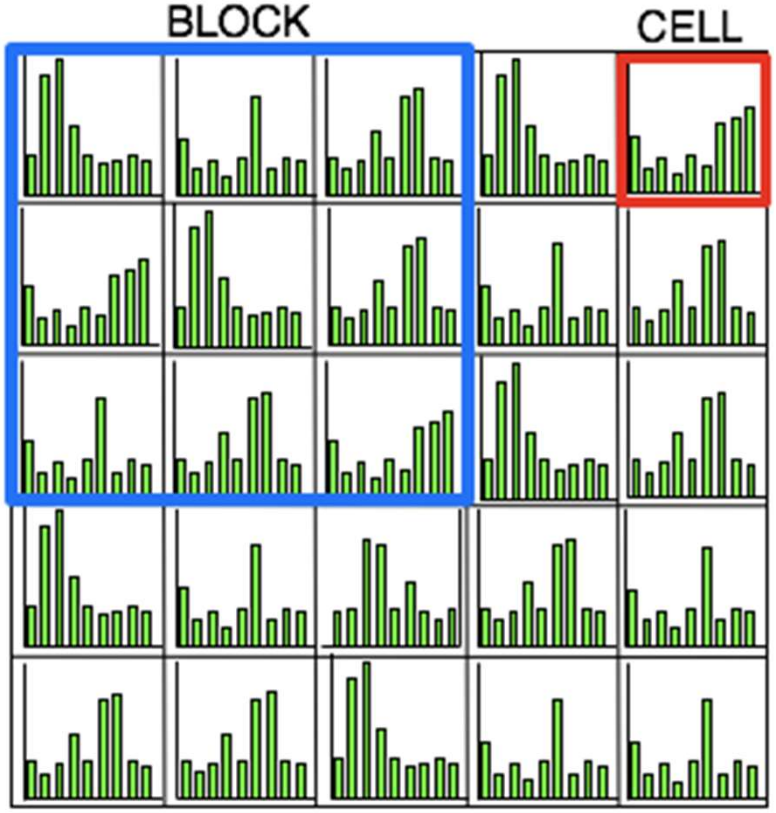
$$\left(\theta+\frac{\Pi}{2}\right)\div\Pi\times 9$$



HOG feature extraction algorithm(3)

4. **Normalization and Descriptor Blocks**
   - Normalization is performed using the following equation:

$$v(n) = \frac{v(n)}{\sqrt{\left(\sum_{k=1}^{3\times3\times9} v(k)^2\right)+1}}$$

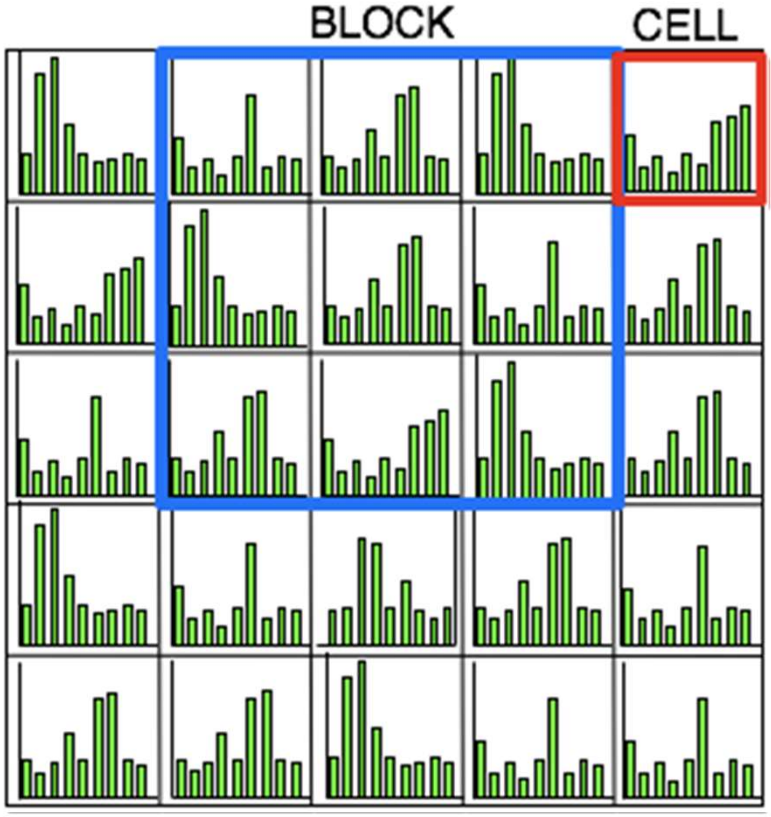


v(n) is the magnitude of each direction

HOG feature extraction algorithm(3)

4. **Normalization and Descriptor Blocks**
   - Normalization is performed using the following equation:

$$v(n) = \frac{v(n)}{\sqrt{\left(\sum_{k=1}^{3\times3\times9} v(k)^2\right)+1}}$$



v is the magnitude of each direction

Normalized vector is computed over Cells and 9 orientations

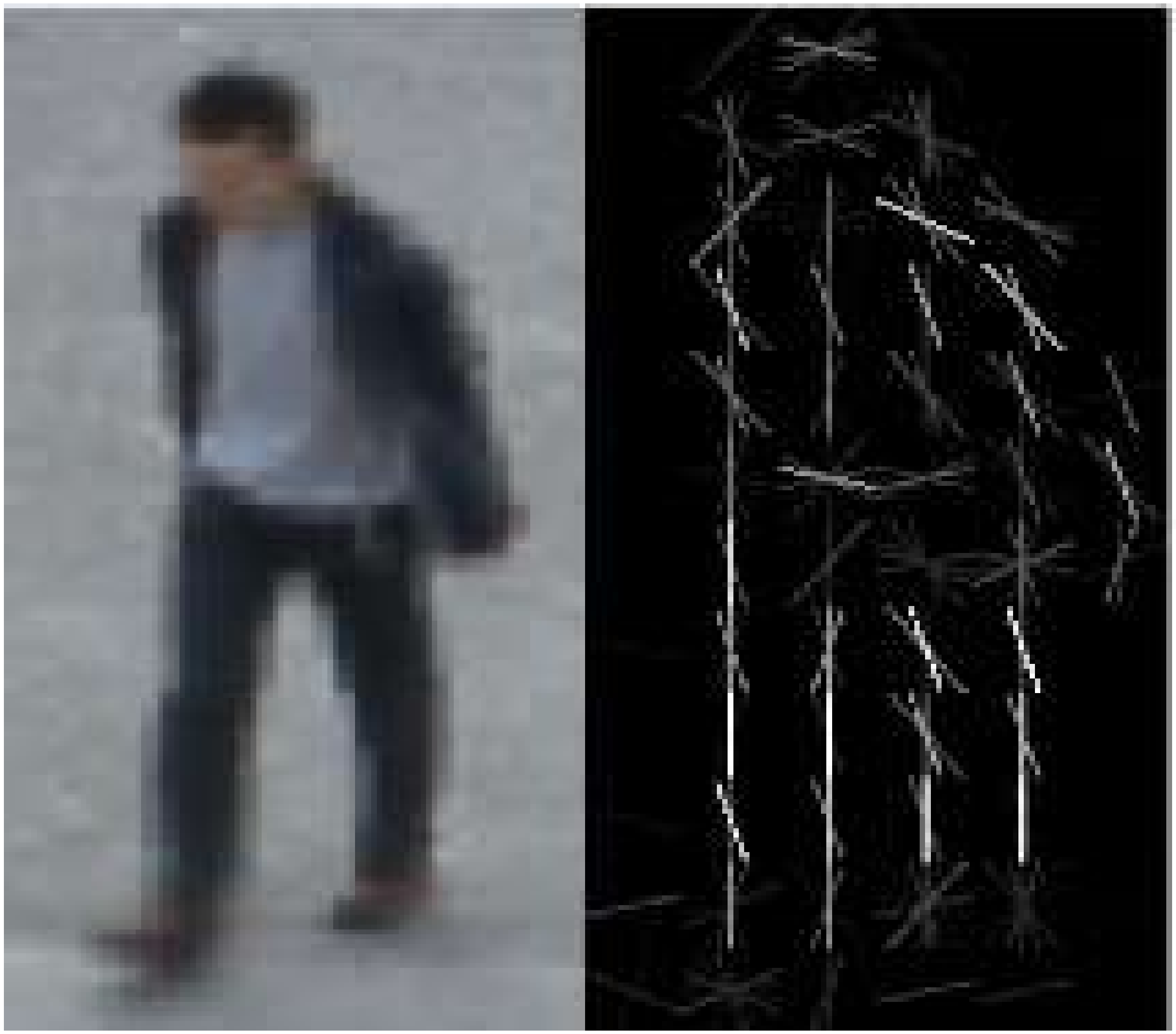
For a block, the vector size = 9x9=81

HOG image

Feature descriptor size

- 12x6 cells
- Number of orientations = 9
- Block size = 3x3=9
- Block moves 4 steps to right and 10 steps down

Descriptor size for total image = 10x4x9x9= 3240



Example of using HOG

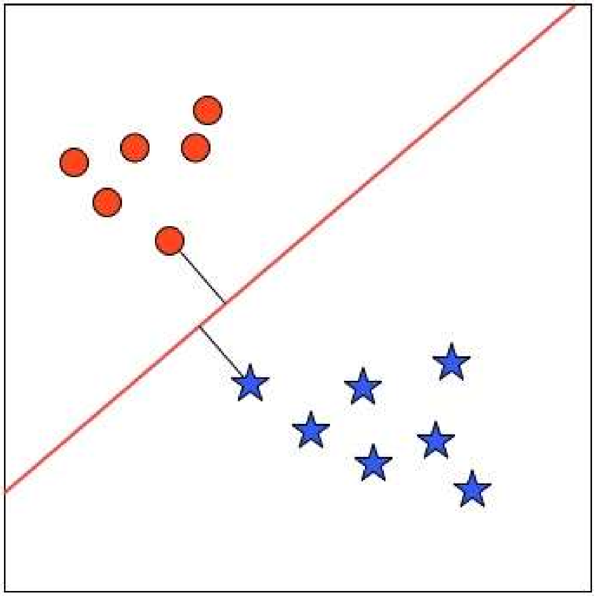
- HOG can represent a rough shape of the object, so that it has been used for general object recognition, such as people or cars.
- In order to achieve the general object recognition, the classifier (eg SVM) is be used.

  1. To teach the classifier, the correct image and the incorrect image.
  2. Scan the classifier to determine whether there are people in the detection window.

SVM based Classification

SVM divides space into two domains according to a teacher signal.

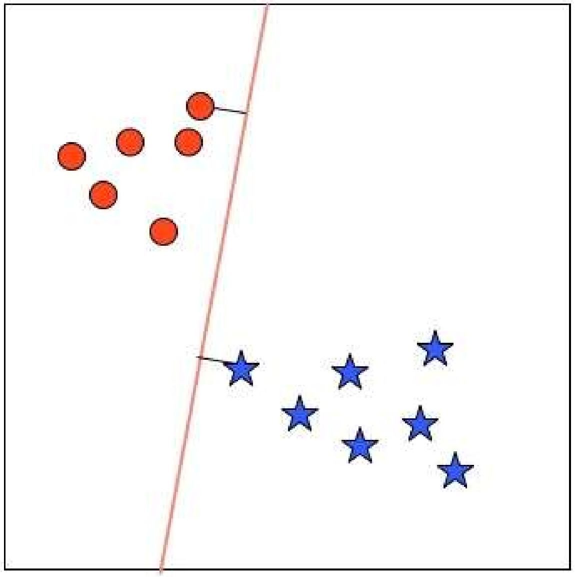
New examples are predicted to belong to a category based on which side of the gap domain.



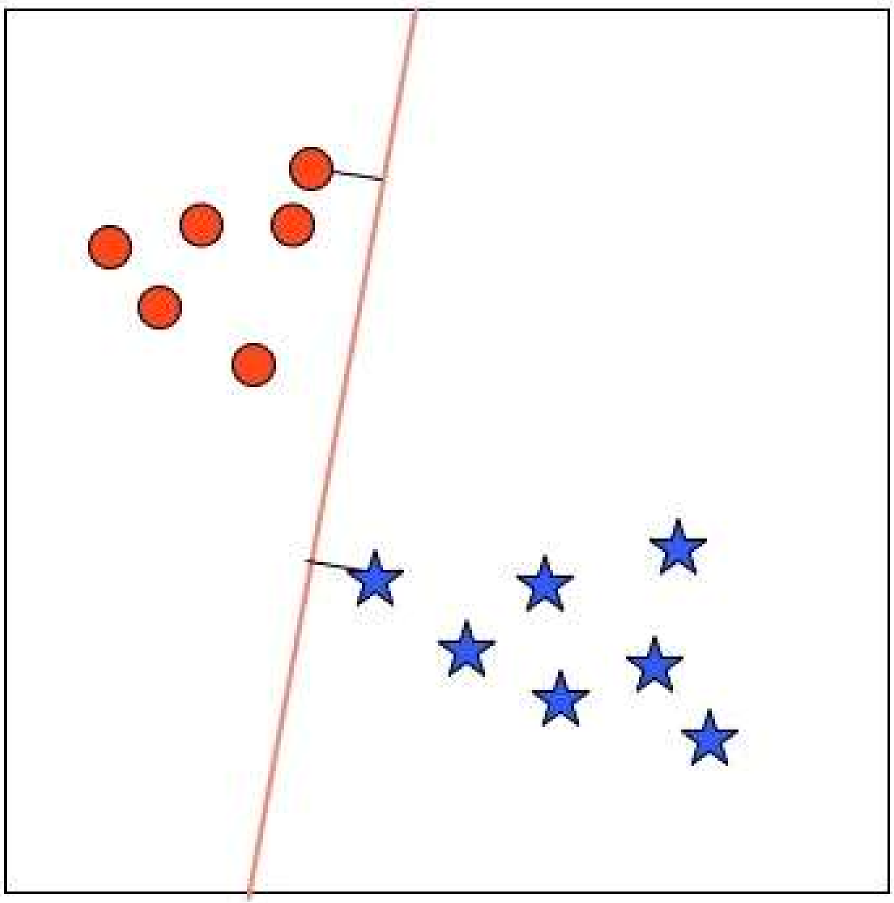
SVM Based Classification

SVM divides space into two domains according to a teacher signal.

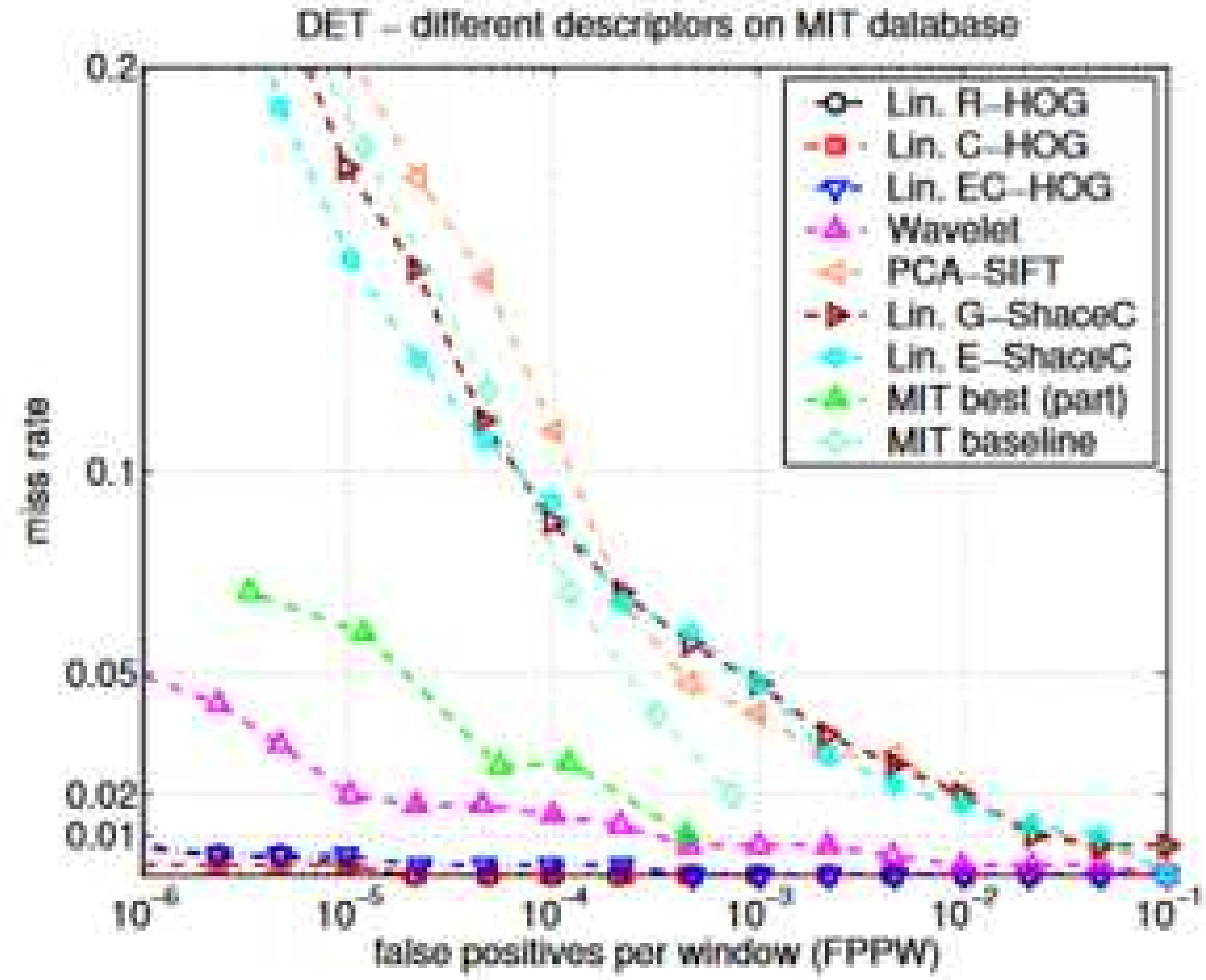
New examples are predicted to belong to a category based on which side of the gap domain.



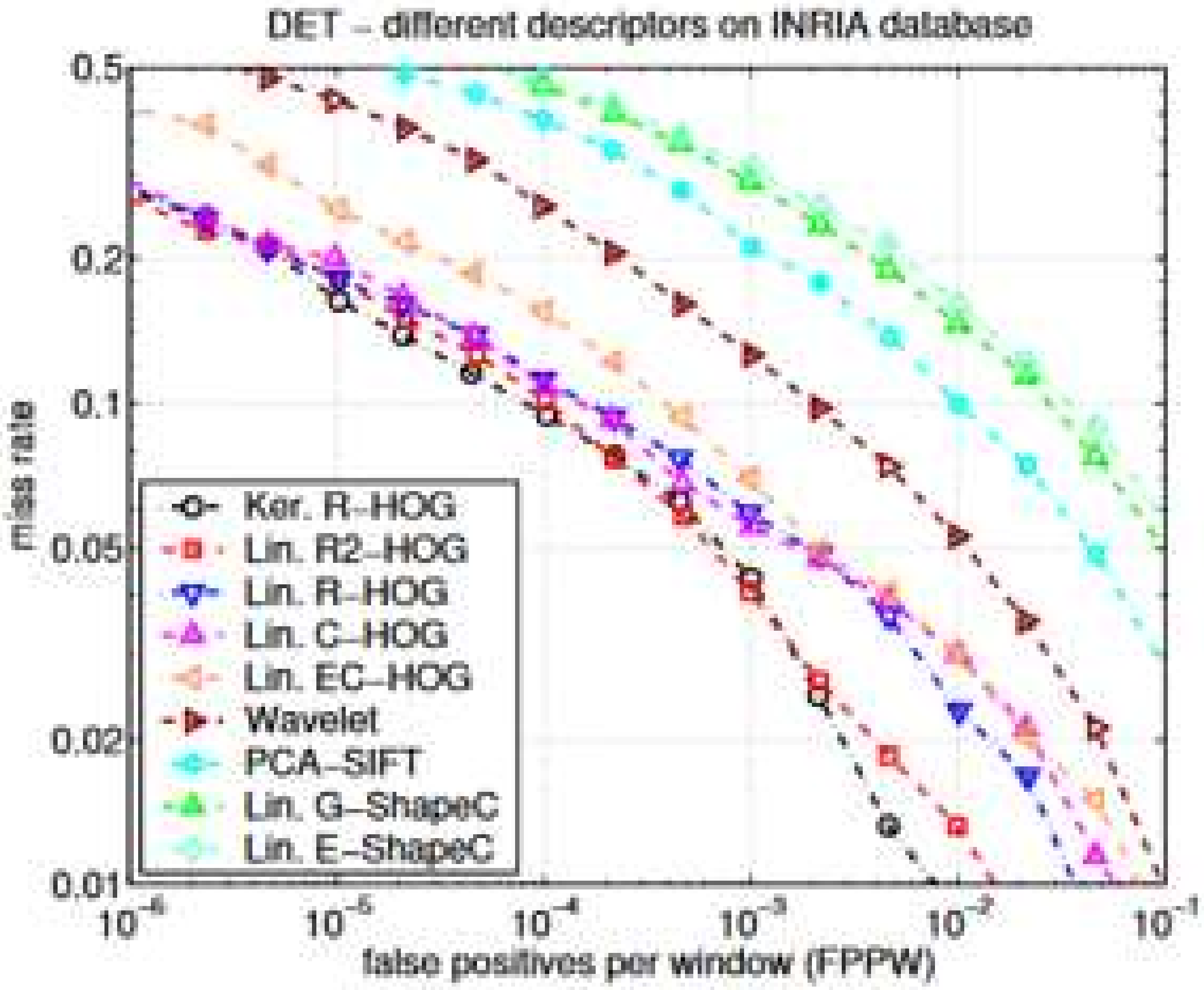
SVM Based Classification



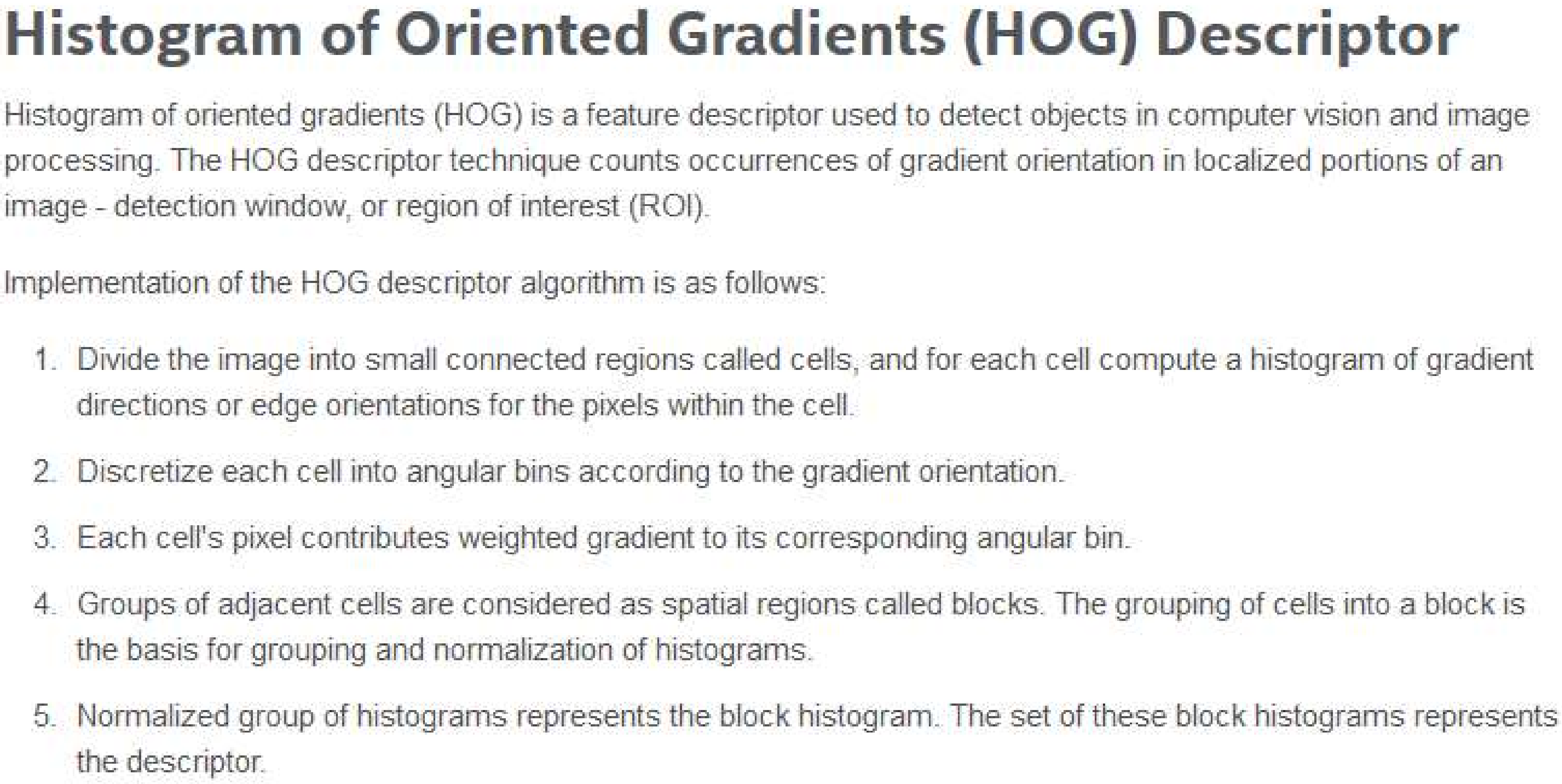
Comparison with Different Feature Descriptors



Comparison with Different Feature Descriptors



Summary



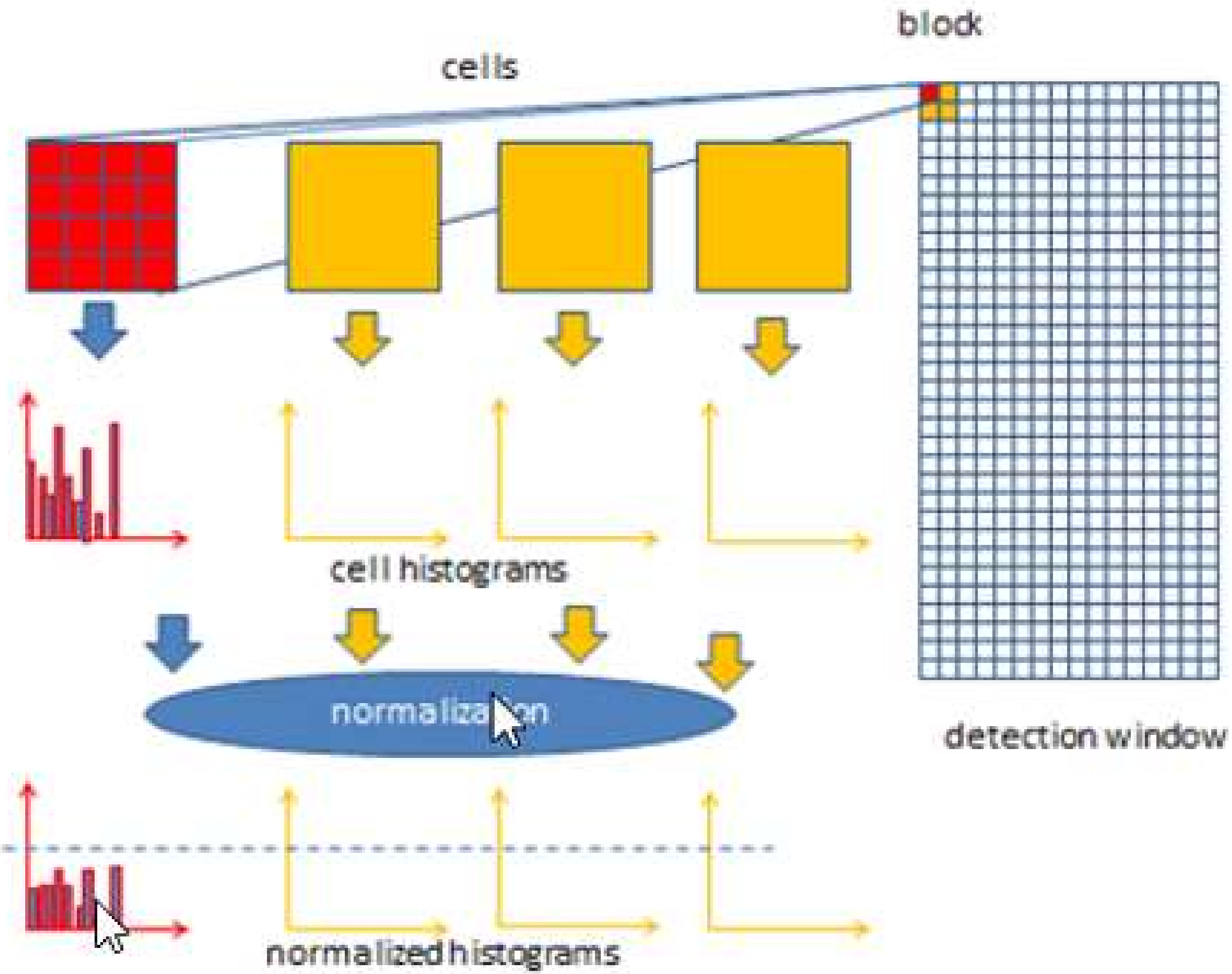
Histogram of Oriented Gradients (HOG) Descriptor

Histogram of oriented gradients (HOG) is a feature descriptor used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window, or region of interest (ROI).

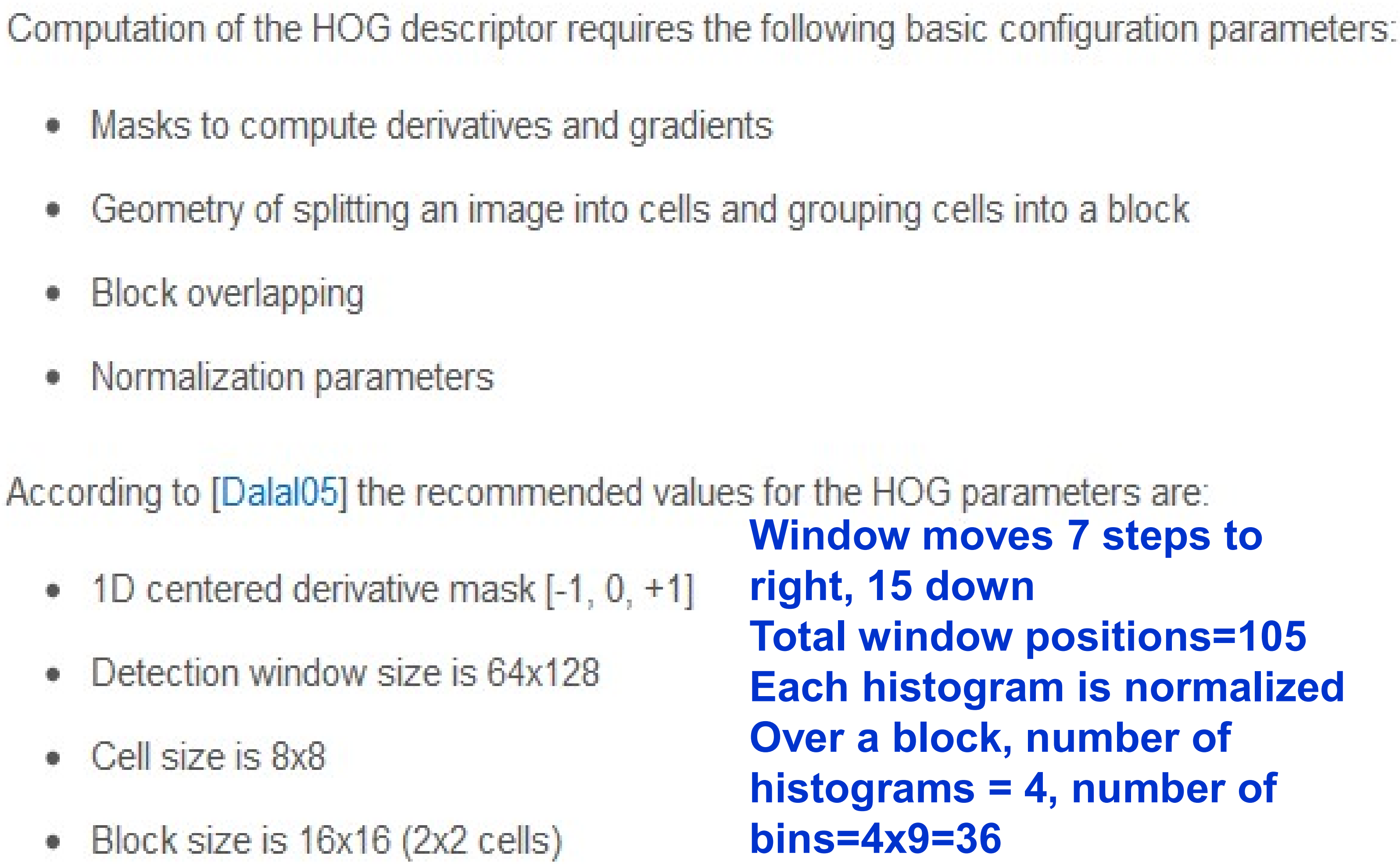
Implementation of the HOG descriptor algorithm is as follows:

1. Divide the image into small connected regions called cells, and for each cell compute a histogram of gradient directions or edge orientations for the pixels within the cell.
2. Discretize each cell into angular bins according to the gradient orientation.
3. Each cell's pixel contributes weighted gradient to its corresponding angular bin.
4. Groups of adjacent cells are considered as spatial regions called blocks. The grouping of cells into a block is the basis for grouping and normalization of histograms.
5. Normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor.

Steps


block
cells
cell histograms
normaliza on
detection window
normalized histograms

Parameters



Computation of the HOG descriptor requires the following basic configuration parameters:

- Masks to compute derivatives and gradients
- Geometry of splitting an image into cells and grouping cells into a block
- Block overlapping
- Normalization parameters

According to [Dalal05] the recommended values for the HOG parameters are:

- 1D centered derivative mask [-1, 0, +1]
- Detection window size is 64x128
- Cell size is 8x8
- Block size is 16x16 (2x2 cells)

Window moves 7 steps to right, 15 down
Total window positions=105
Each histogram is normalized
Over a block, number of histograms = 4, number of bins=4x9=36

Feature descriptor size = 15x7x9x4=3780 in Dalal’s original paper

Discussion

- Navneet Dalal and Bill Briggs initially developed this for person identification but it was shown that this could be used for other applications as well.

Scale Invariant Feature Transform (SIFT)

The SIFT (Scale Invariant Feature Transform) Detector and Descriptor

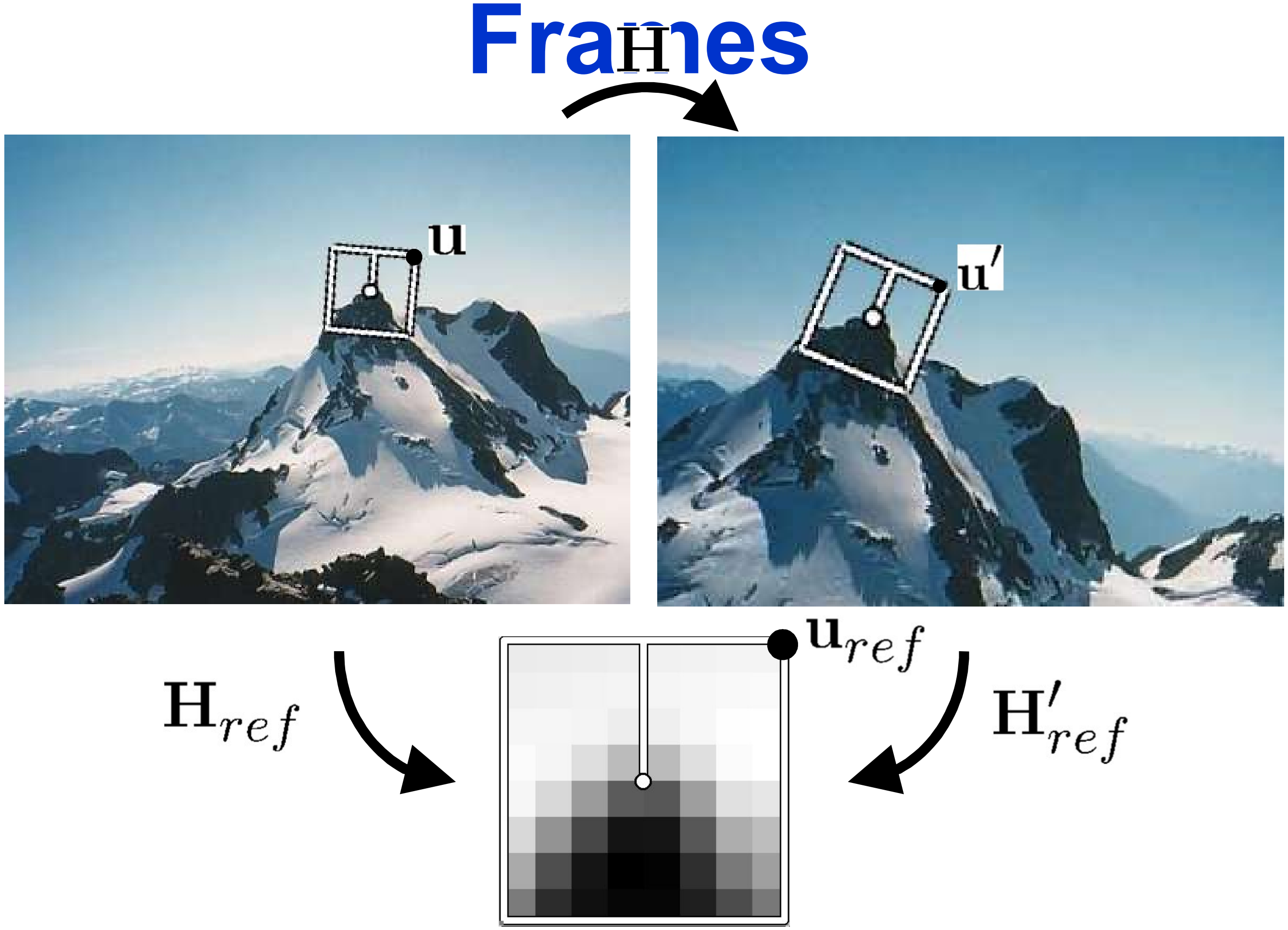
developed by David Lowe

University of British Columbia

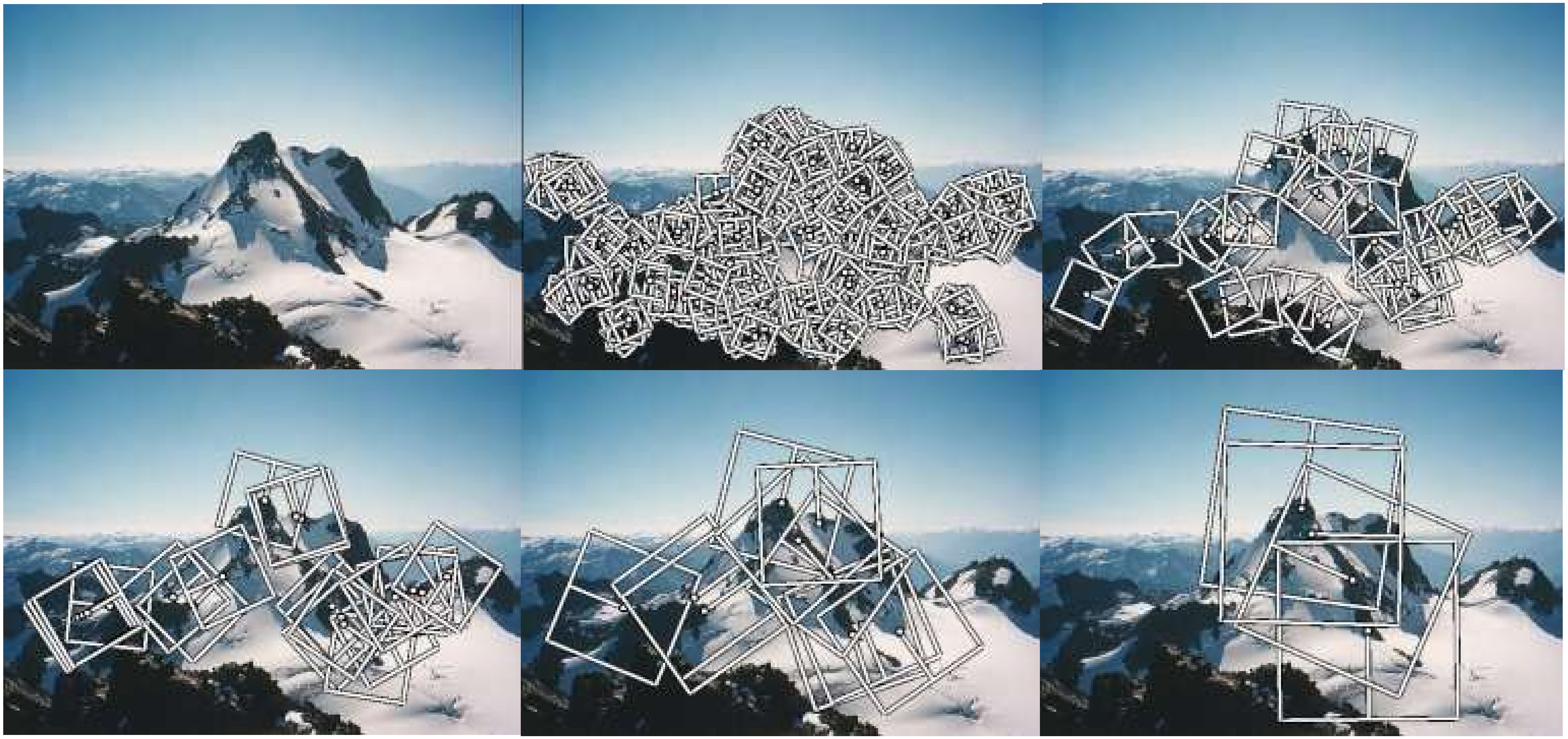
Initial paper ICCV 1999

Newer journal paper IJCV 2004

Review: Matt Brown's Canonical Frames



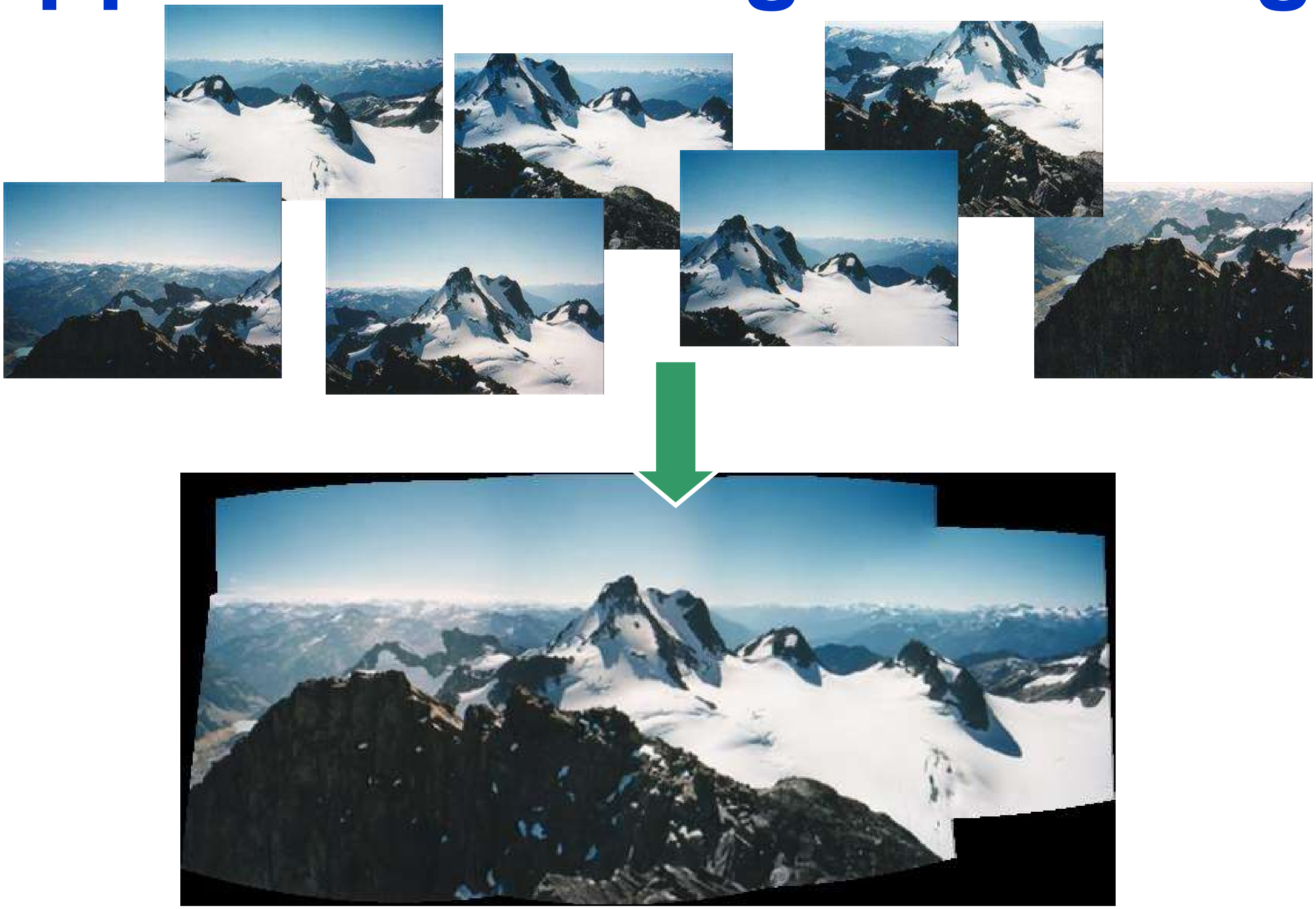
Multi-Scale Oriented Patches



- Extract oriented patches at multiple scales

[Brown, Szeliski, Winder CVPR 2005]

Application: Image Stitching



[Microsoft Digital Image Pro version 10]

Ideas from Matt’s Multi-Scale Oriented Patches

1. Detect an interesting patch with an interest operator. Patches are translation invariant.

2. Determine its dominant orientation.

3. Rotate the patch so that the dominant orientation points upward. This makes the patches rotation invariant.

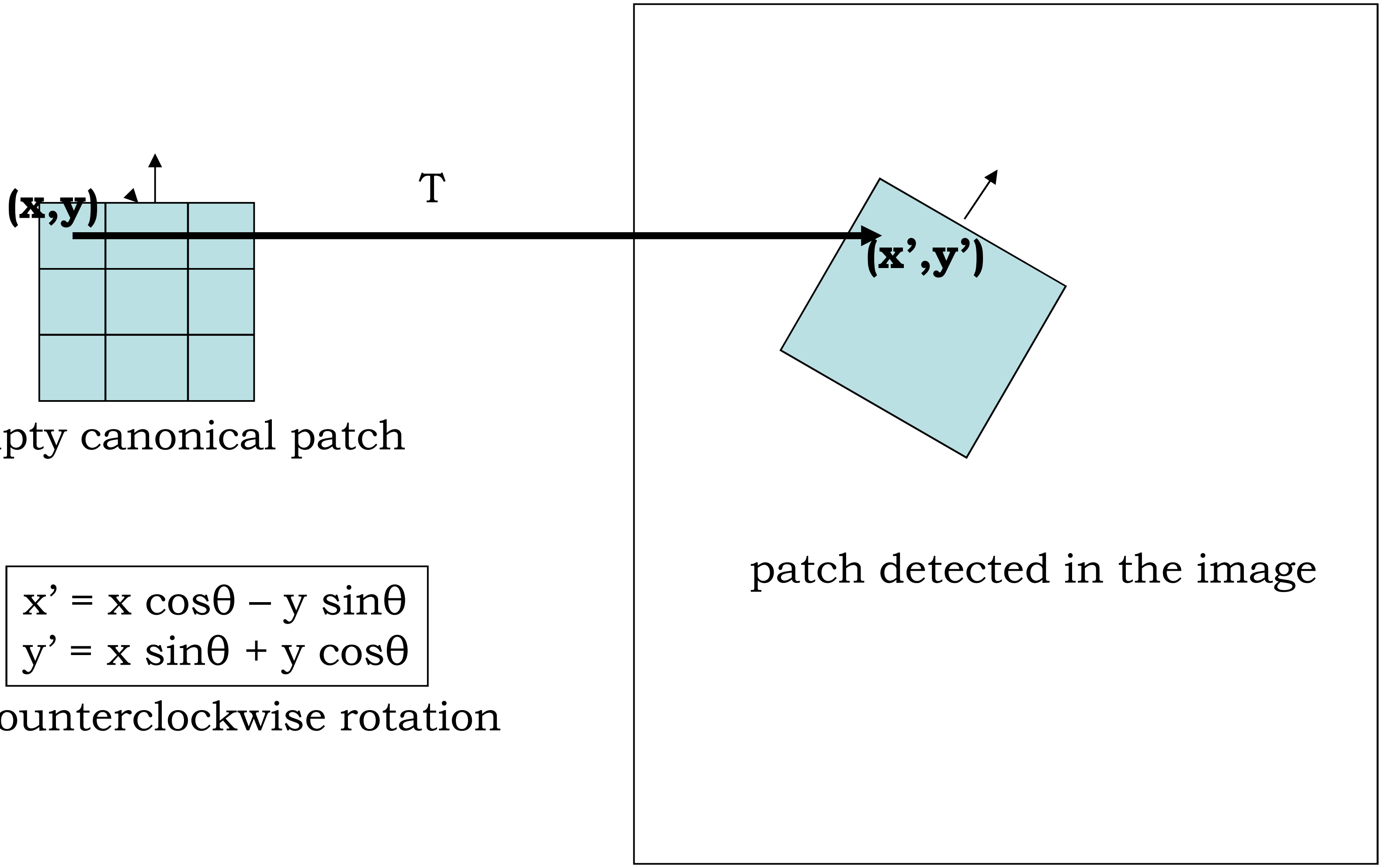
4. Do this at multiple scales, converting them all to one scale through sampling.

5. Convert to illumination “invariant” form

Implementation Concern: How do you rotate a patch?

- Start with an “empty” patch whose dominant direction is “up”.
- For each pixel in your patch, compute the position in the detected image patch. It will be in floating point and will fall between the image pixels.
- Interpolate the values of the 4 closest pixels in the image, to get a value for the pixel in your patch.

Rotating a Patch

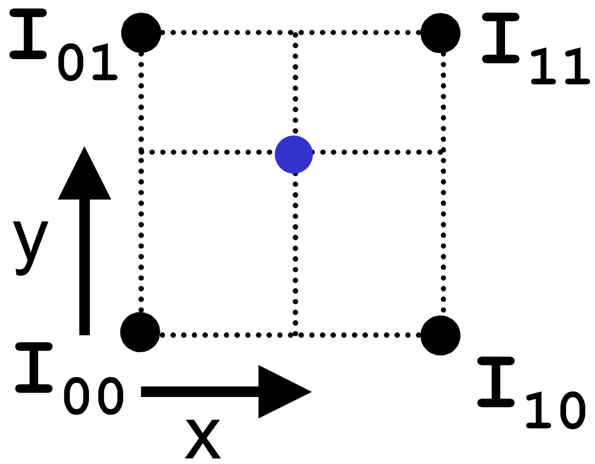


$$T \quad \boxed{\begin{array}{l} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \end{array}}$$

counterclockwise rotation

Using Bilinear Interpolation

- Use all 4 adjacent samples



SIFT: Motivation

- The Harris operator is not invariant to scale and correlation is not invariant to rotation[1].

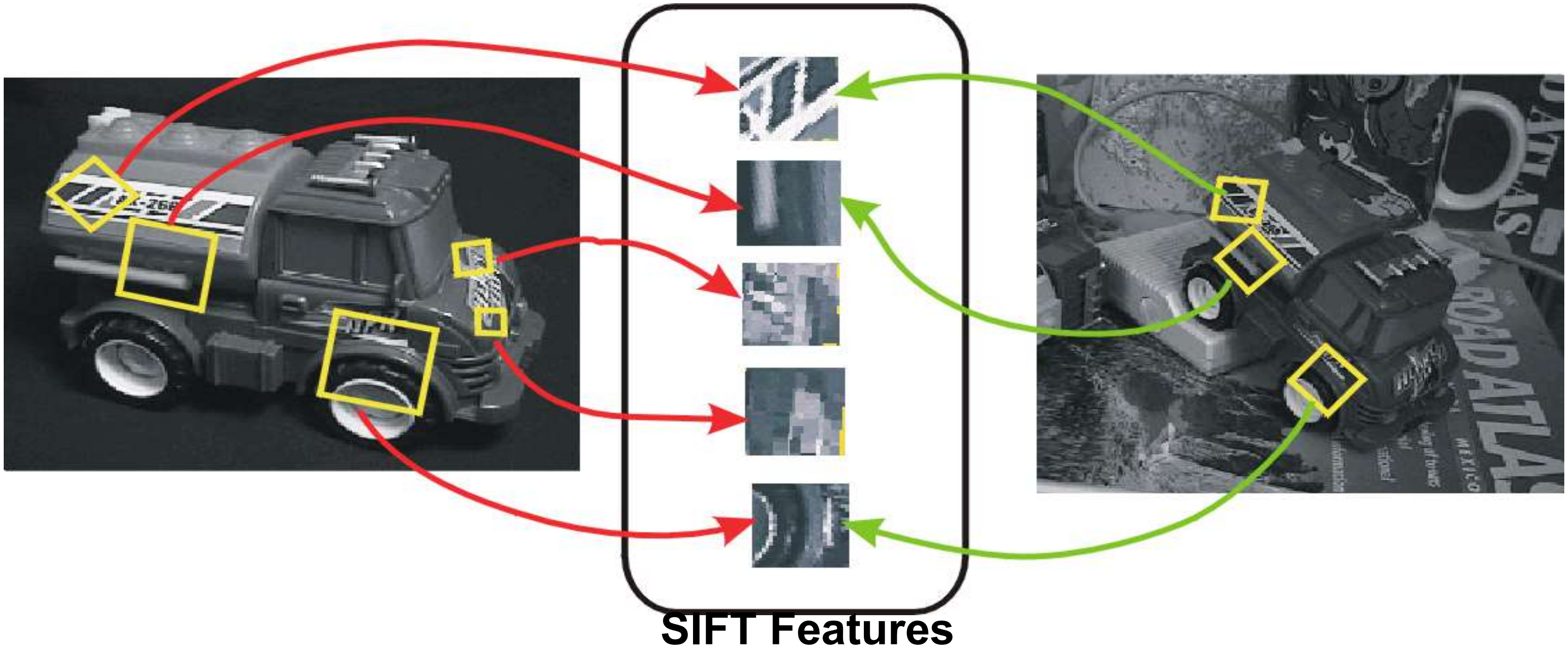
- For better image matching, Lowe's goal was to develop an interest operator that is invariant to scale and rotation.

- Also, Lowe aimed to create a descriptor that was robust to the variations corresponding to typical viewing conditions. The descriptor is the most-used part of SIFT.

[1] But Schmid and Mohr developed a rotation invariant descriptor for it in 1997.

Idea of SIFT

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



SIFT Features

Claimed Advantages of SIFT

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance
- **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

Overall Procedure at a High Level

1. Scale-space extrema detection

   Search over multiple scales and image locations.

2. Keypoint localization

   Fit a model to determine location and scale.
   Select keypoints based on a measure of stability.

3. Orientation assignment

   Compute best orientation(s) for each keypoint region.

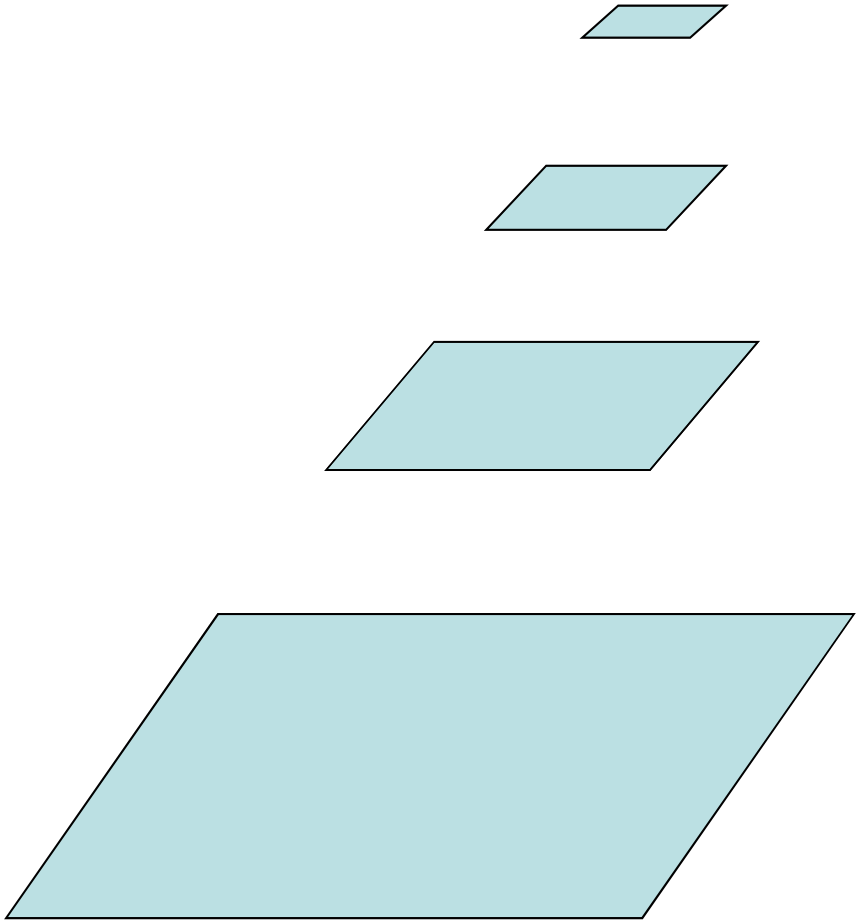
4. Keypoint description

   Use local image gradients at selected scale and rotation to describe each keypoint region.

1. Scale-space extrema detection

- Goal: Identify locations and scales that can be repeatably assigned under different views of the same scene or object.
- Method: search for stable features across multiple scales using a continuous function of scale.
- Prior work has shown that under a variety of assumptions, the best function is a Gaussian function.
- The scale space of an image is a function $L(x,y,\sigma)$ that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.

Aside: Image Pyramids



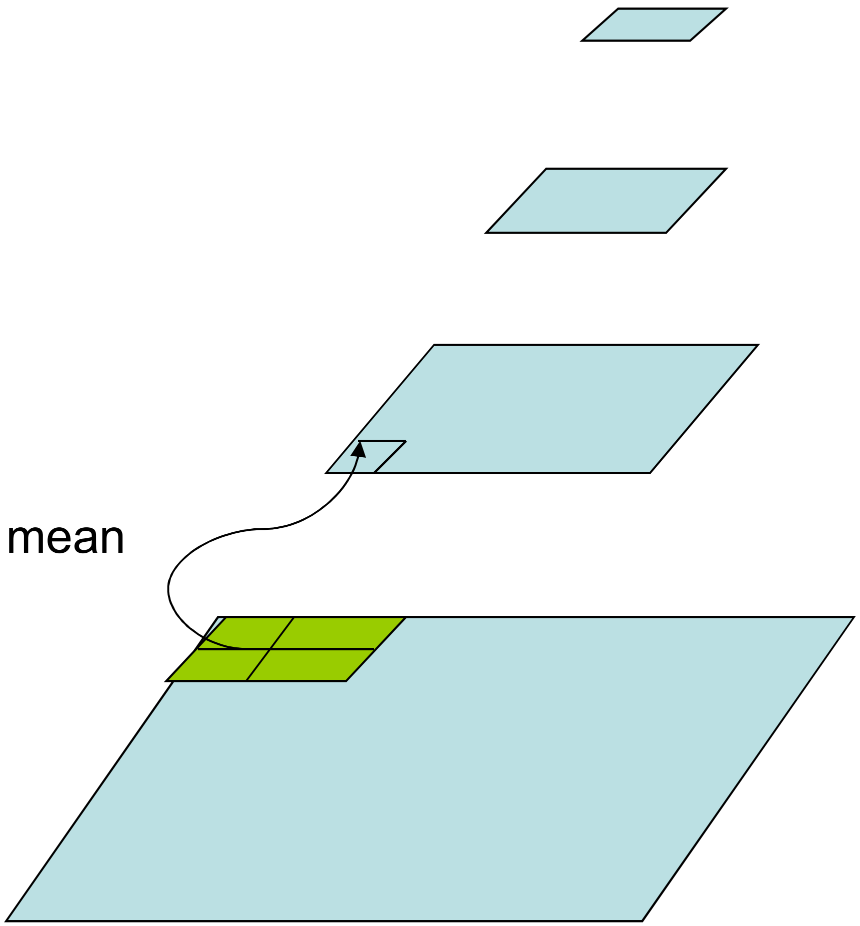
And so on.

3rd level is derived from the 2nd level according to the same funtion

2nd level is derived from the original image according to some function

Bottom level is the original image.

Aside: Mean Pyramid



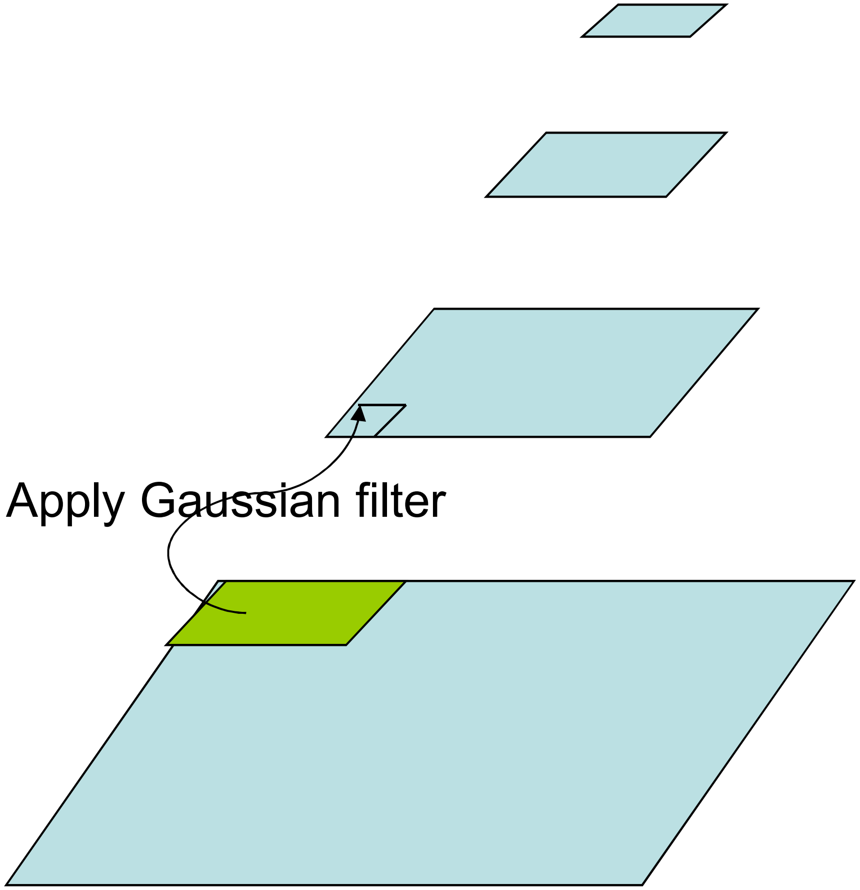
And so on.

At 3rd level, each pixel is the mean of 4 pixels in the 2nd level.

At 2nd level, each pixel is the mean of 4 pixels in the original image.

Bottom level is the original image.

Aside: Gaussian Pyramid At each level, image is smoothed and reduced in size.



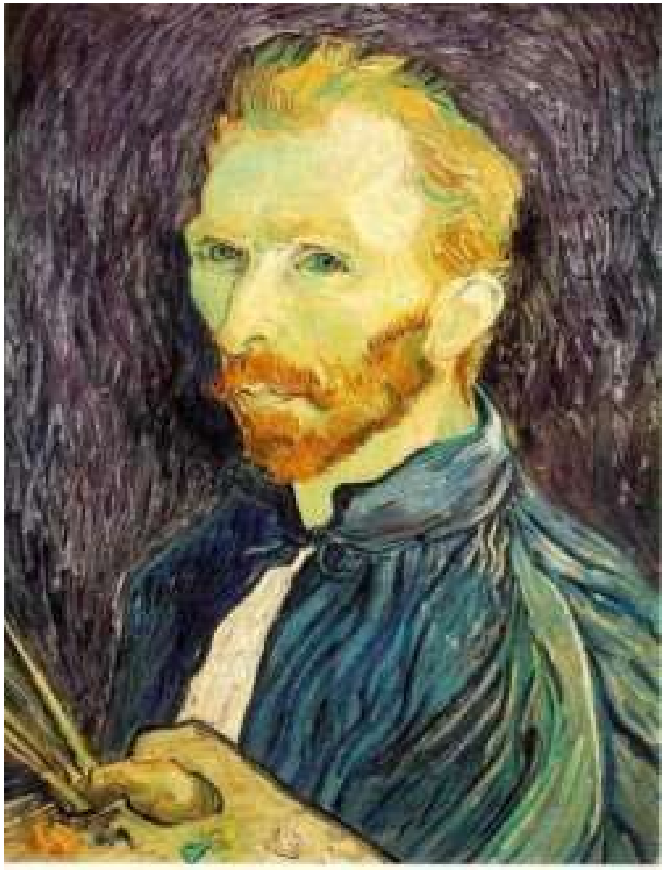
Apply Gaussian filter

And so on.

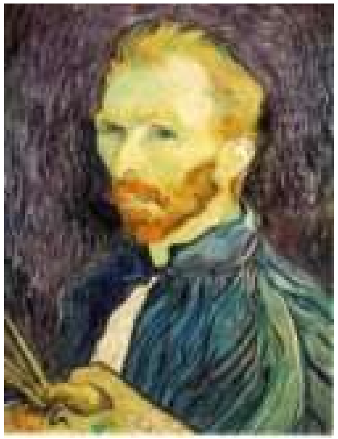
At 2nd level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.

Bottom level is the original image.

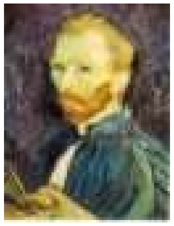
Example: Subsampling with Gaussian pre-filtering



Gaussian 1/2

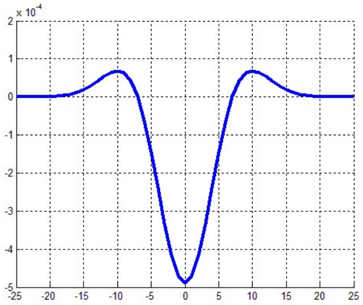


G 1/4

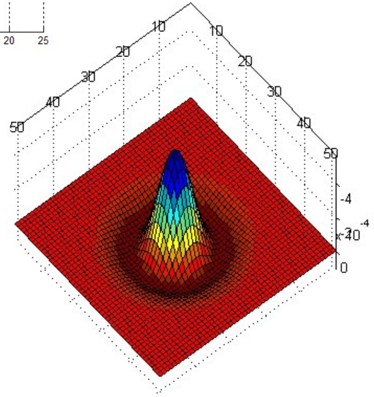


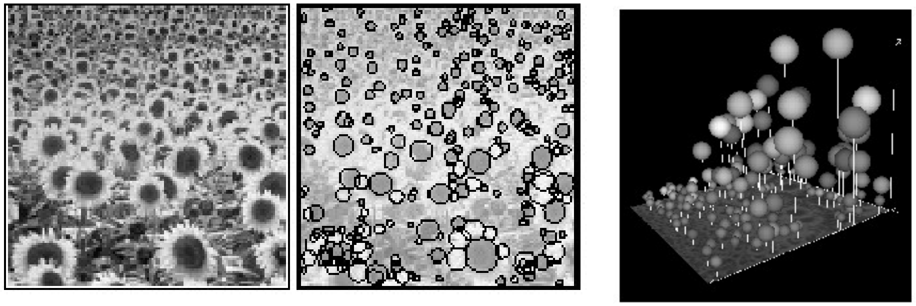
G 1/8

Lowe’s Scale-space Interest Points



- **Laplacian of Gaussian** kernel
  - Scale normalised (x by scale^2)
  - Proposed by Lindeberg
- Scale-space detection
  - Find local maxima across scale/space
  - A good “blob” detector



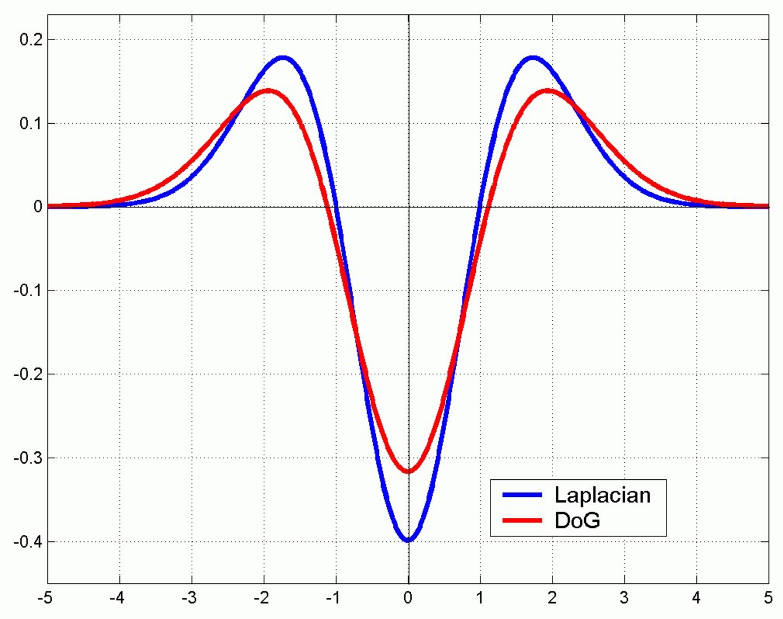


$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}}$$

$$\nabla^2 G(x, y, \sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

[T. Lindeberg IJCV 1998]

Lowe's Scale-space Interest Points: Difference of Gaussians



- Gaussian is an ad hoc solution of heat diffusion equation

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

- Hence

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G.$$

- k is not necessarily very small in practice

Lowe’s Pyramid Scheme

- Scale space is separated into octaves:
  - Octave 1 uses scale σ
  - Octave 2 uses scale 2σ
  - etc.

- In each octave, the initial image is repeatedly convolved with Gaussians to produce a set of scale space images.

- Adjacent Gaussians are subtracted to produce the DOG

- After each octave, the Gaussian image is down-sampled by a factor of 2 to produce an image ¼ the size to start the next level.

Lowe’s Pyramid Scheme

s+2 filters

$\sigma_{s+1}=2^{(s+1)/s}\sigma_0$

.

.

$\sigma_i=2^{i/s}\sigma_0$

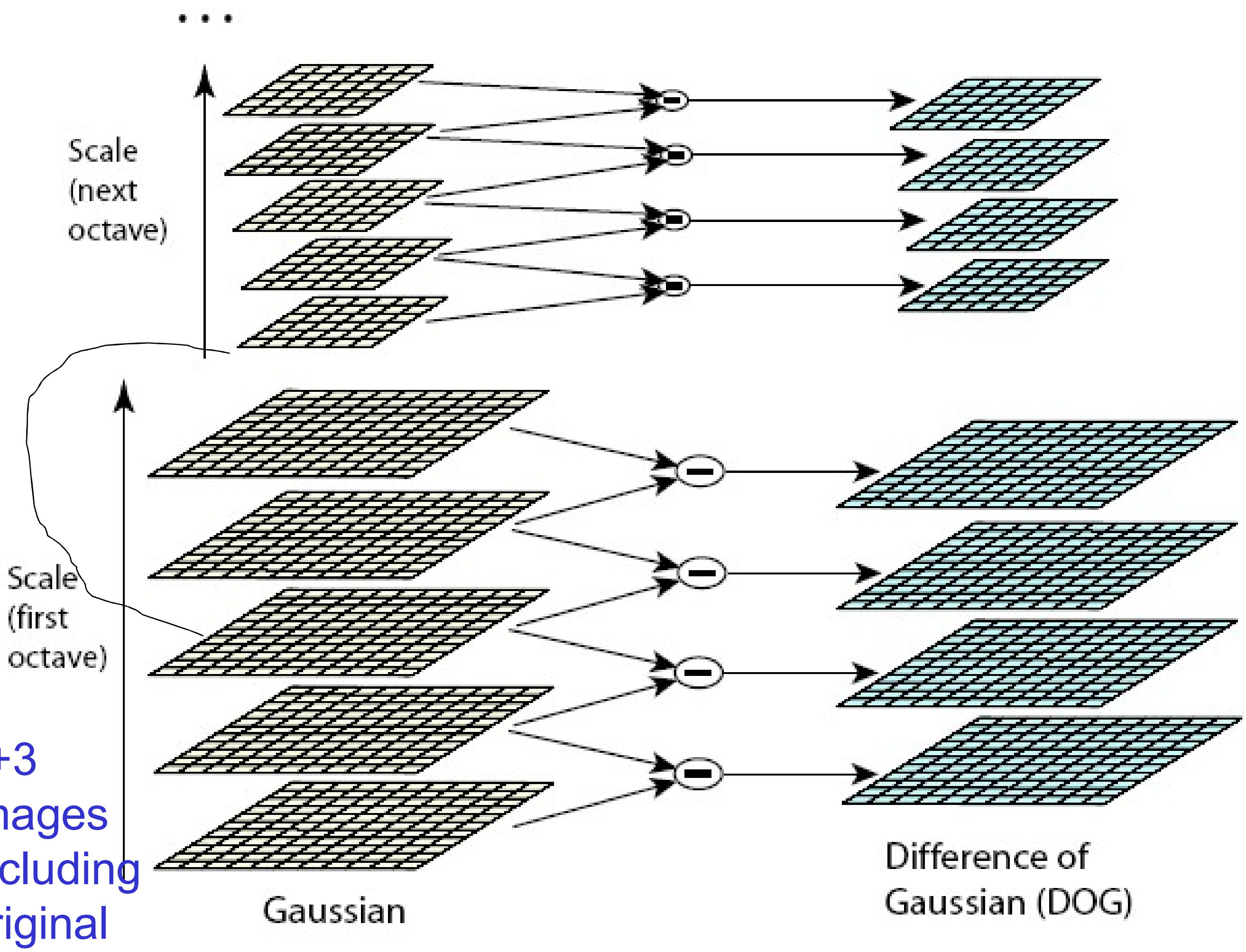
.

.

$\sigma_2=2^{2/s}\sigma_0$

$\sigma_1=2^{1/s}\sigma_0$

σ_0



s+3 images including original

s+2 differ-ence images

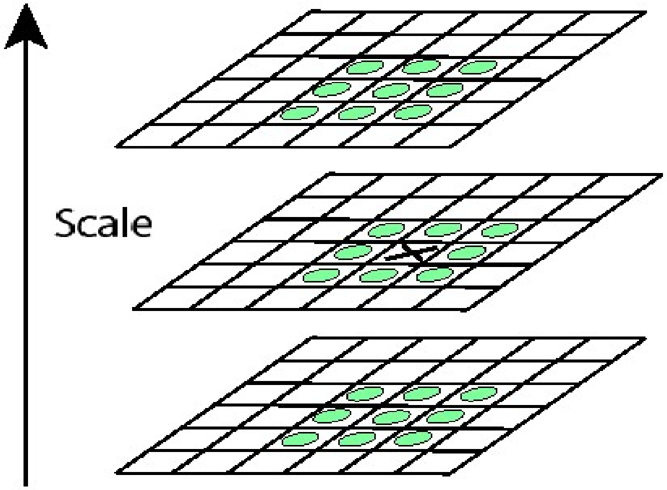
The parameter **s** determines the number of images per octave.

Key point localization

s+2 difference images.
top and bottom ignored.
s planes searched.

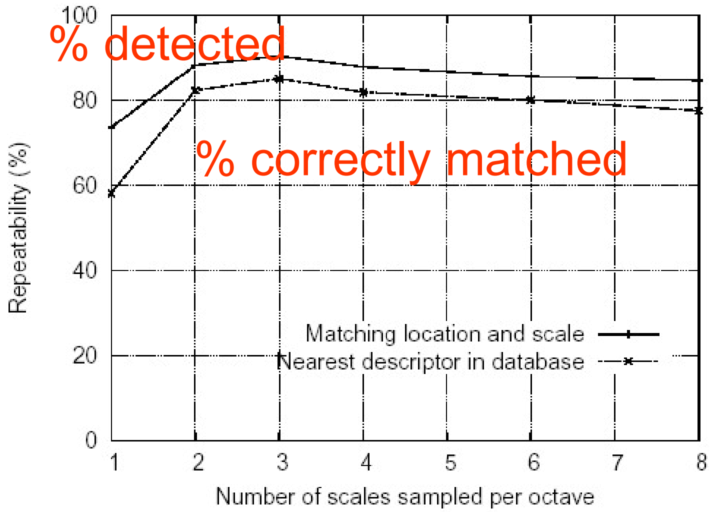
- Detect maxima and minima of difference-of-Gaussian in scale space

- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below



For each max or min found, output is the **location** and the **scale**.

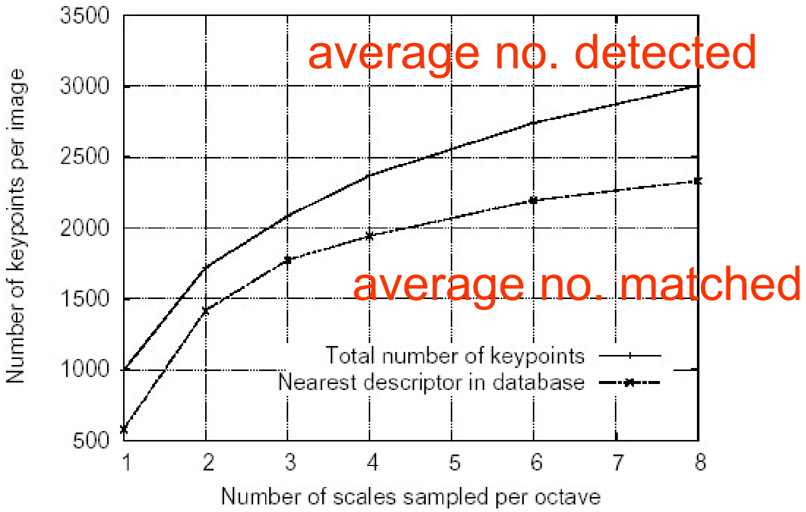
Scale-space extrema detection: experimental results over 32 images that were synthetically transformed and noise added.



% detected

% correctly matched

Stability



average no. detected

average no. matched

Expense

- **Sampling in scale for efficiency**
  - How many scales should be used per octave? S=?
    - More scales evaluated, more keypoints found
    - S < 3, stable keypoints increased too
    - S > 3, stable keypoints decreased
    - S = 3, maximum stable keypoints found

Keypoint localization

- Once a keypoint candidate is found, perform a detailed fit to nearby data to determine
  - location, scale, and ratio of principal curvatures
- In initial work keypoints were found at location and scale of a central sample point.
- In newer work, they fit a 3D quadratic function to improve interpolation accuracy.
- The Hessian matrix was used to eliminate edge responses.

Keypoint localization

- There are still a lot of points, some of them are not good enough.
- The locations of keypoints may be not accurate.
- Eliminating edge points.

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

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

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^{T} \hat{\mathbf{x}}. \tag{3}$$

Eliminating the Edge Response

- Reject flat areas (in terms of intensity):
  - □ $|D(\hat{\mathbf{x}})| < 0.03$
- Reject edges:

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

Let α be the eigenvalue with larger magnitude and β the smaller.

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

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

Let r = α/β.
So α = rβ

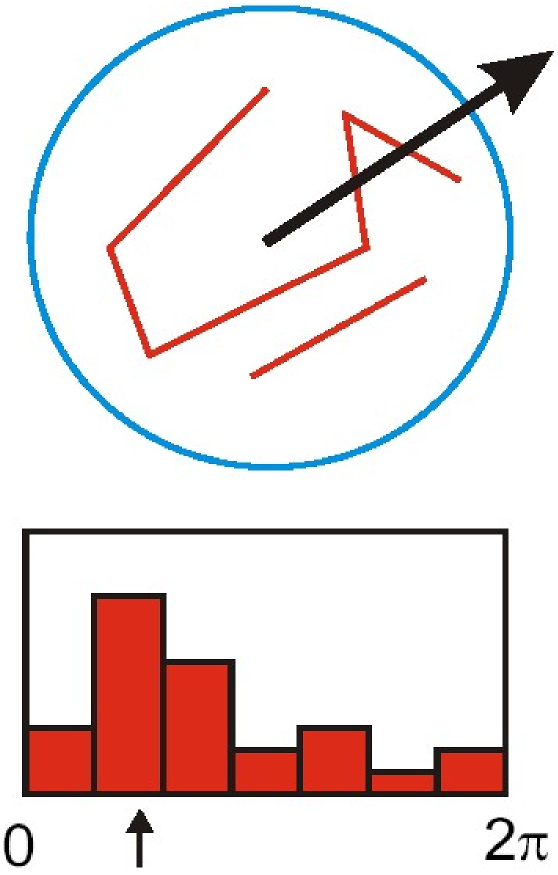
$$\frac{\operatorname{Tr}(\mathbf{H})^2}{\operatorname{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

$(r+1)^2/r$ is at a min when the 2 eigenvalues are equal.

  - □ r < 10

- **What does this look like?**

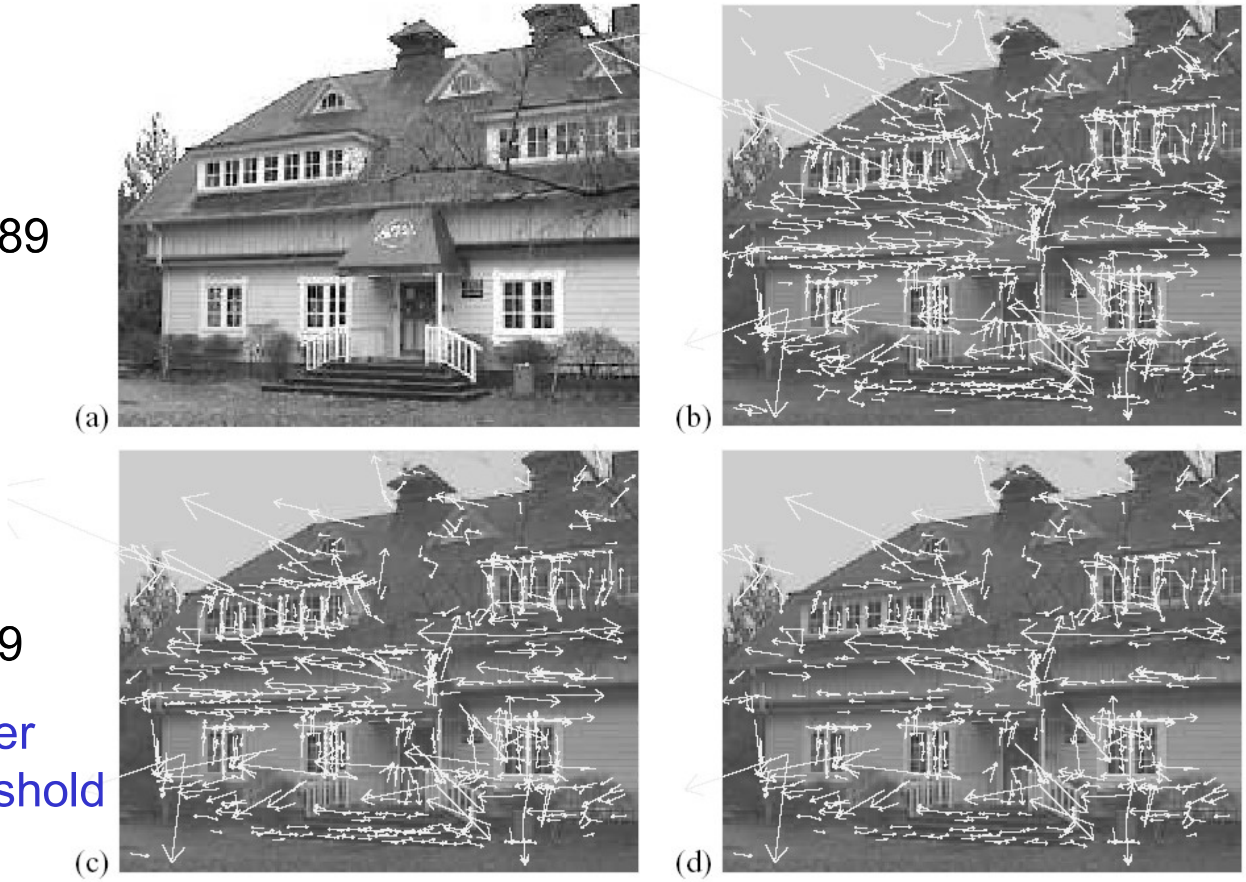
3. Orientation assignment



- Create histogram of local gradient directions at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale,orientation)

If 2 major orientations, use both.

Keypoint localization with orientation



233x189

832

initial keypoints

729

keypoints after gradient threshold

536

keypoints after ratio threshold

4. Keypoint Descriptors

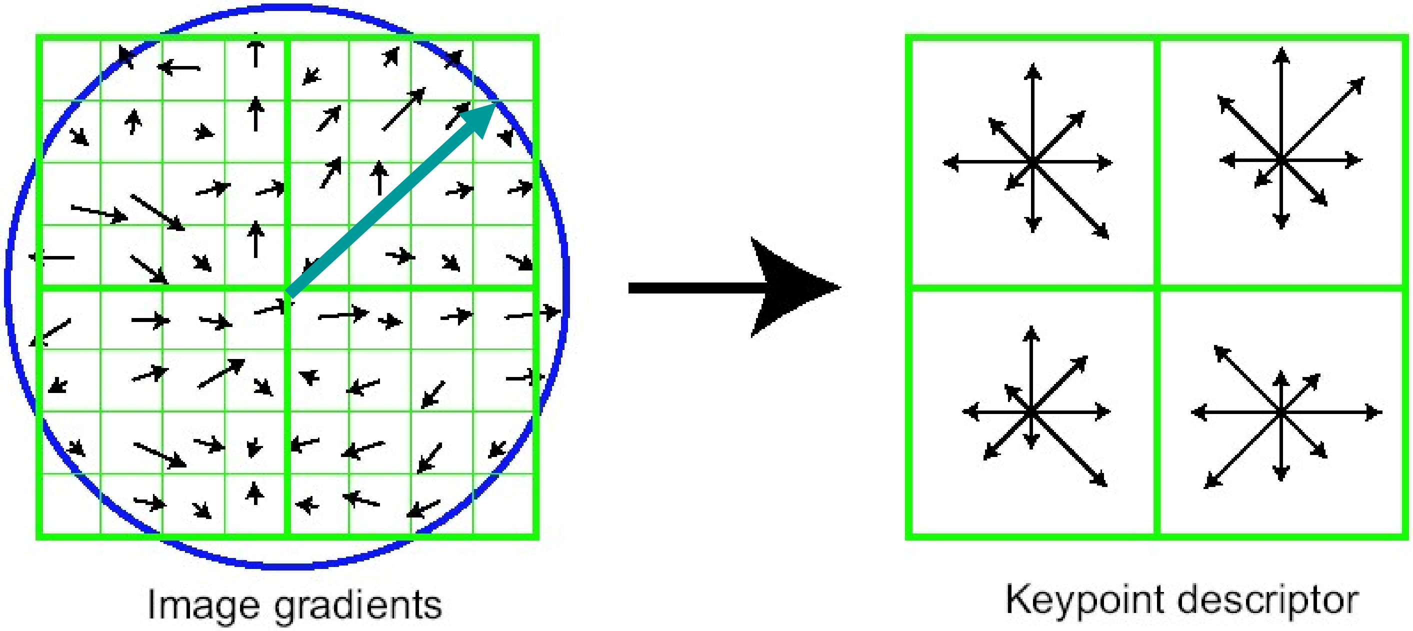
- At this point, each keypoint has
  - location
  - scale
  - orientation
- Next is to compute a descriptor for the local image region about each keypoint that is
  - highly distinctive
  - invariant as possible to variations such as changes in viewpoint and illumination

Normalization

- Rotate the window to standard orientation

- Scale the window size based on the scale at which the point was found.

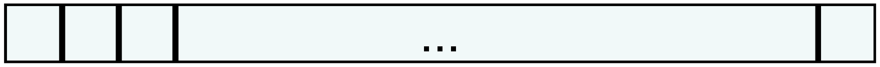
Lowe’s Keypoint Descriptor (shown with 2 X 2 descriptors over 8 X 8)



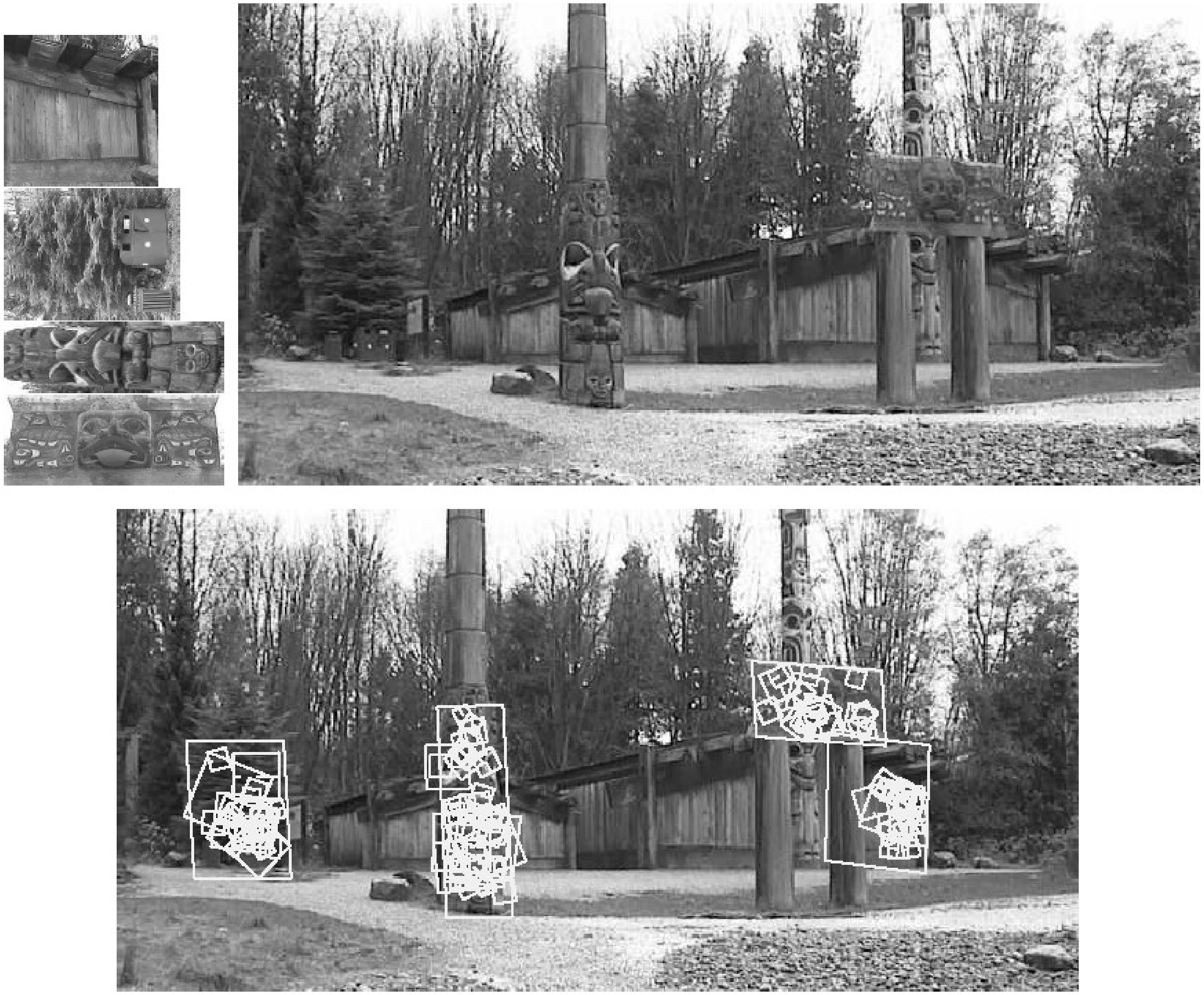
In experiments, 4x4 arrays of 8 bin histogram is used, a total of 128 features for one keypoint

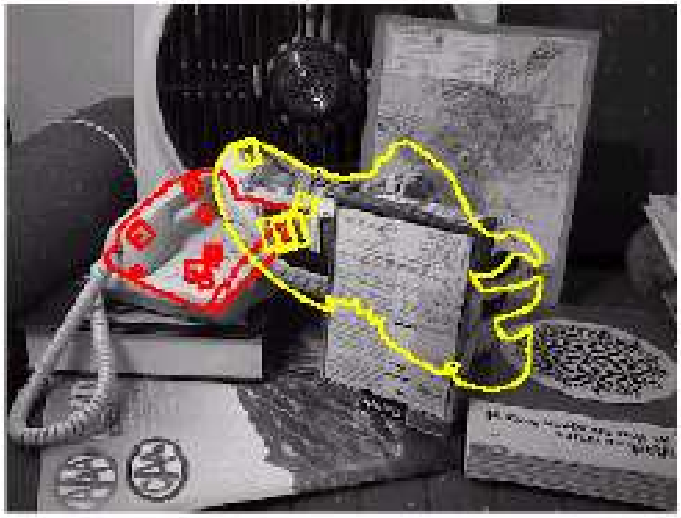
Lowe’s Keypoint Descriptor

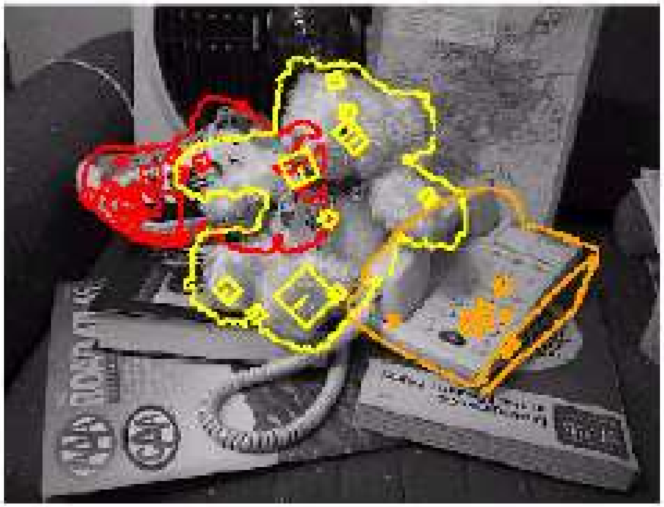
- use the normalized region about the keypoint
- compute gradient magnitude and orientation at each point in the region
- weight them by a Gaussian window overlaid on the circle
- create an orientation histogram over the 4 X 4 subregions of the window
- 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 times 8 directions gives a vector of 128 values.



Using SIFT for Matching “Objects”







Uses for SIFT

- Feature points are used also for:
  - Image alignment (homography, fundamental matrix)
  - 3D reconstruction (e.g. Photo Tourism)
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … many others

[Photo Tourism: Snavely et al. SIGGRAPH 2006]

Contd…