EEE-6512: IMAGE PROCESSING AND COMPUTER VISION

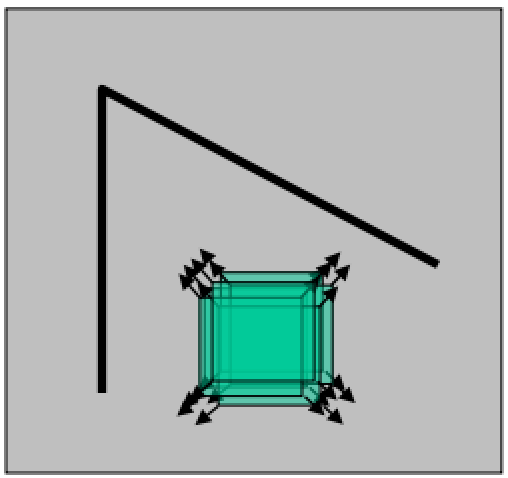
November 3, 2017

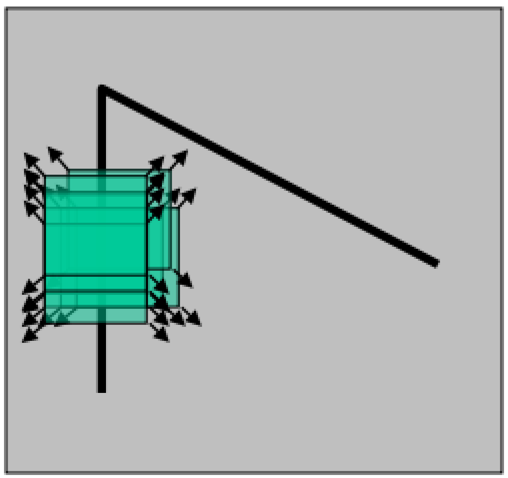
Lecture #9: Edges and Features

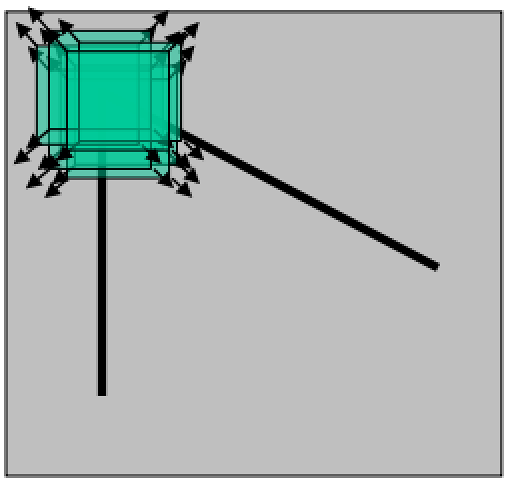
Dept. of Electrical and Computer Engineering

dwoodard@ece.ufl.edu

Harris Corner Detector







“flat” region:
no change in
all directions

“edge”:
no change along
the edge direction

“corner”:
significant change
in all directions

https://dsp.stackexchange.com/questions/14338/corner-detection-using-chris-harris-mike-stephens

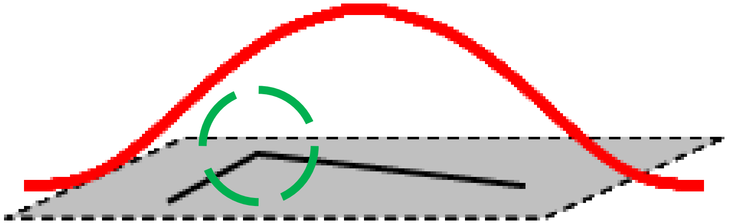
Harris corner detector

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 (cont.)

Need to quantify the change in pixel intensity given a small shift by u and v in the window w:

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

The resulting difference caused by the window shift, or the error

Window's position at x, y

Pixel intensity when the window is displaced by u, v

Pixel intensity at x, y

Harris corner detector (cont.)

Using the Taylor series expansion, we can simplify E:

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \ \ldots$$

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

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

$$E(u,v) = \sum_{x,y} w(x,y)\, u^2 I_x^2 + 2uvI_xI_y + \ v^2 I_y^2$$

Harris corner detector (cont.)

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 (cont.)

Only minimum of E is taken into account

➢A new corner measurement → minimum of second derivative

Intensity change in shifting window: eigenvalue analysis

$$\mathbf{A} \qquad \mathbf{Q} \qquad \mathbf{\Lambda} \qquad \mathbf{Q}^{-1}$$

$$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = \underbrace{\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix}}_{\substack{\text{Eigen vectors} \\ \text{of} \\ \mathbf{A}}} \underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}}_{\substack{\text{Eigen values} \\ \text{of} \\ \mathbf{A}}} \underbrace{\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix}^{-1}}_{\substack{\text{Eigen vectors} \\ \text{of} \\ \mathbf{A}}}$$

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

$$\lambda_1, \lambda_2 - \text{eigenvalues of } M$$

http://datinker.com/?p=211

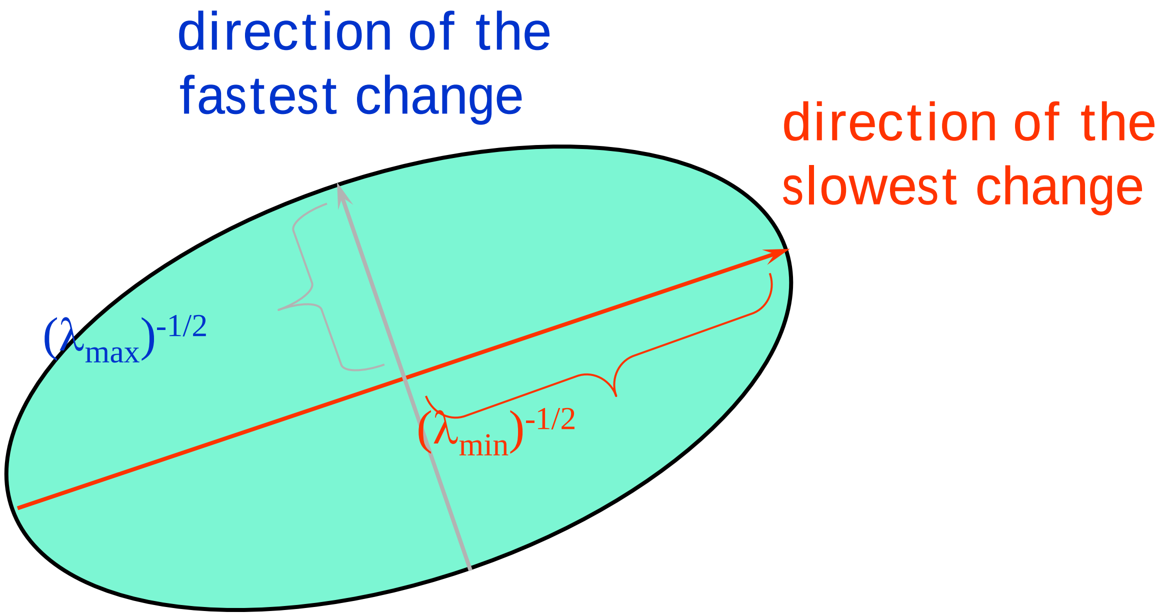
Harris corner detector (cont.)

Intensity change in shifting window: eigenvalue analysis

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

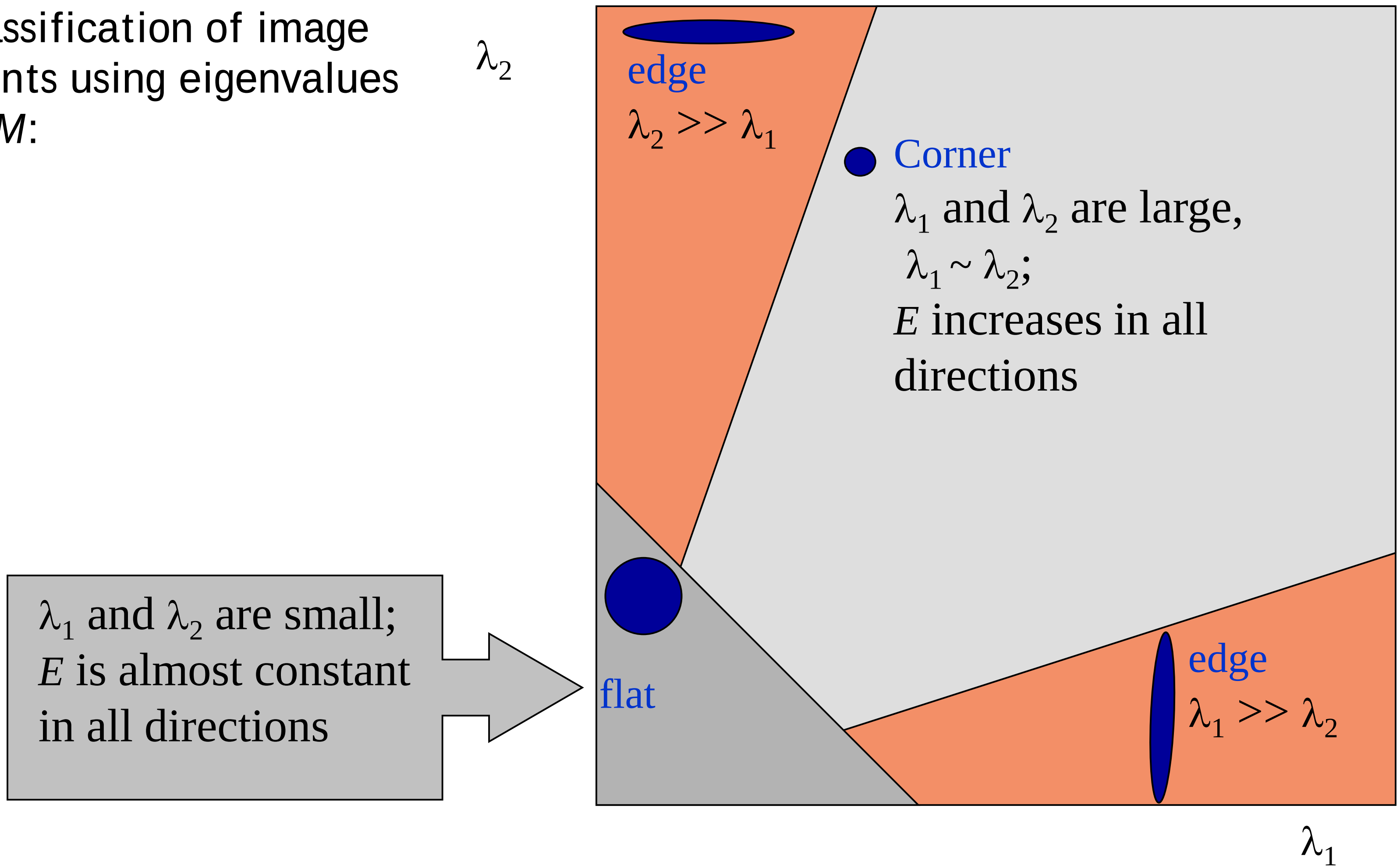
λ_1, λ_2 – eigenvalues of M

Ellipse *E(u,v)* = const



Harris corner detector (cont.)

Classification of image points using eigenvalues of *M*:



Harris corner detector (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)

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

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

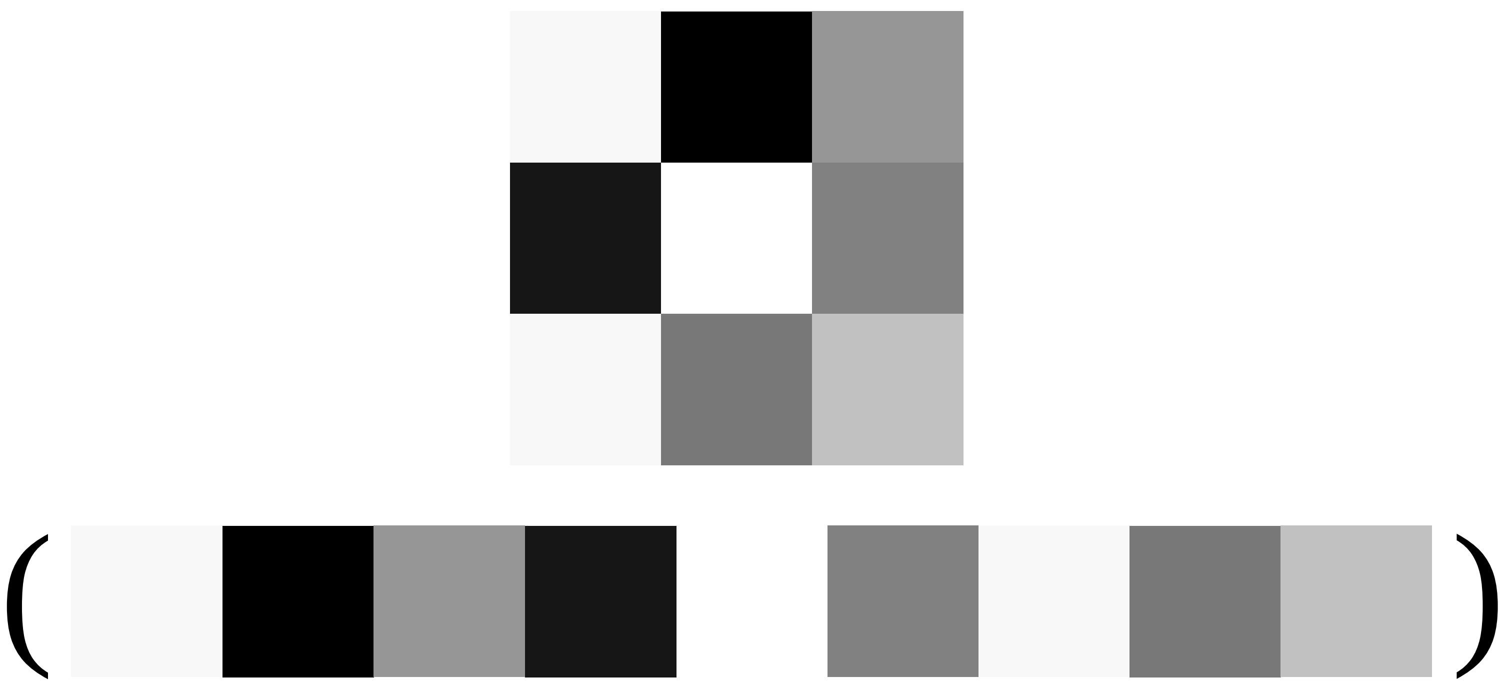
5. Compute the response of the detector at each pixel

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

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

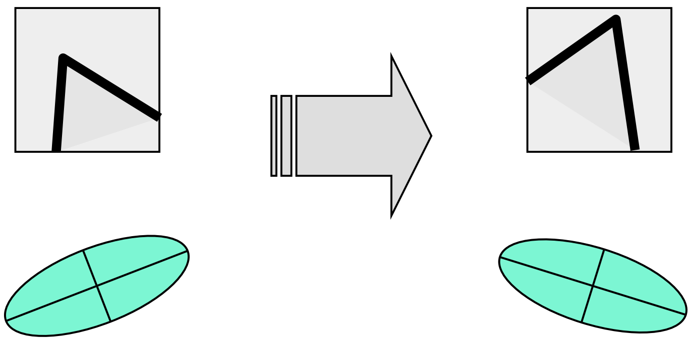
Now we know where features are

- But, how to match them?
- What is the descriptor for a feature? The simplest solution is the intensities of its spatial neighbors. This might not be robust to brightness change or small shift/rotation.



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

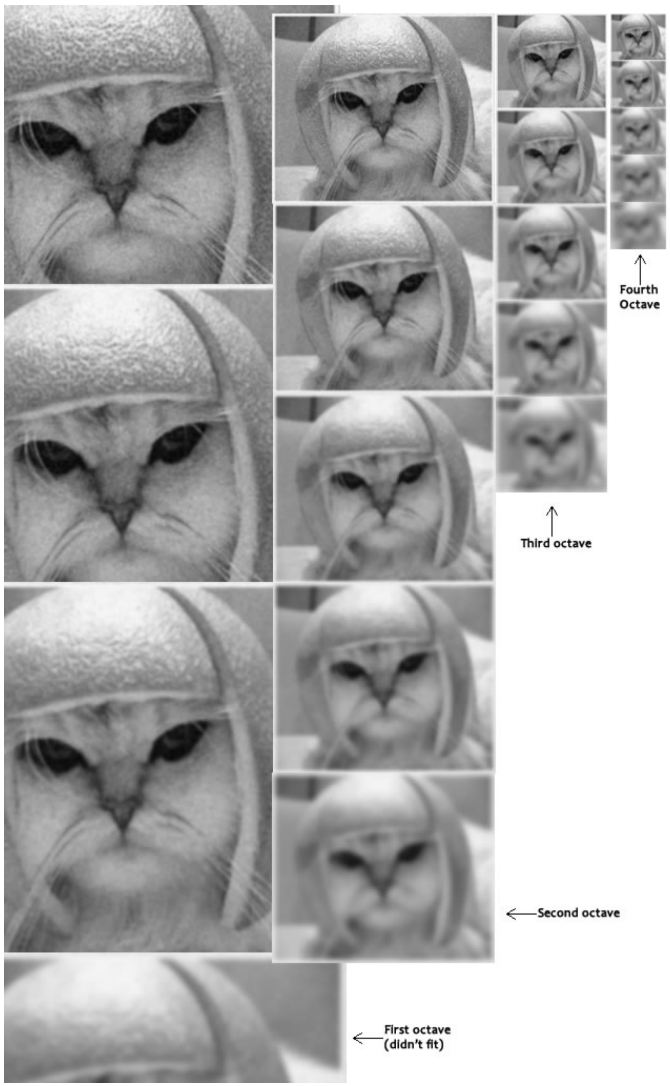
But non-invariant to image scale!

Scale Invariant Feature Transform (SIFT)

SIFT Feature Detection

- The first step of SIFT is to build a Laplacian pyramid of the image.
- The second step is to determine, for every pixel and for every scale, whether the pixel is a local maximum among its 26 neighbors.
- A final step discards pixels in untextured areas or along intensity edges.

SIFT Feature Detection (cont'd)
Step 1: Create scale space



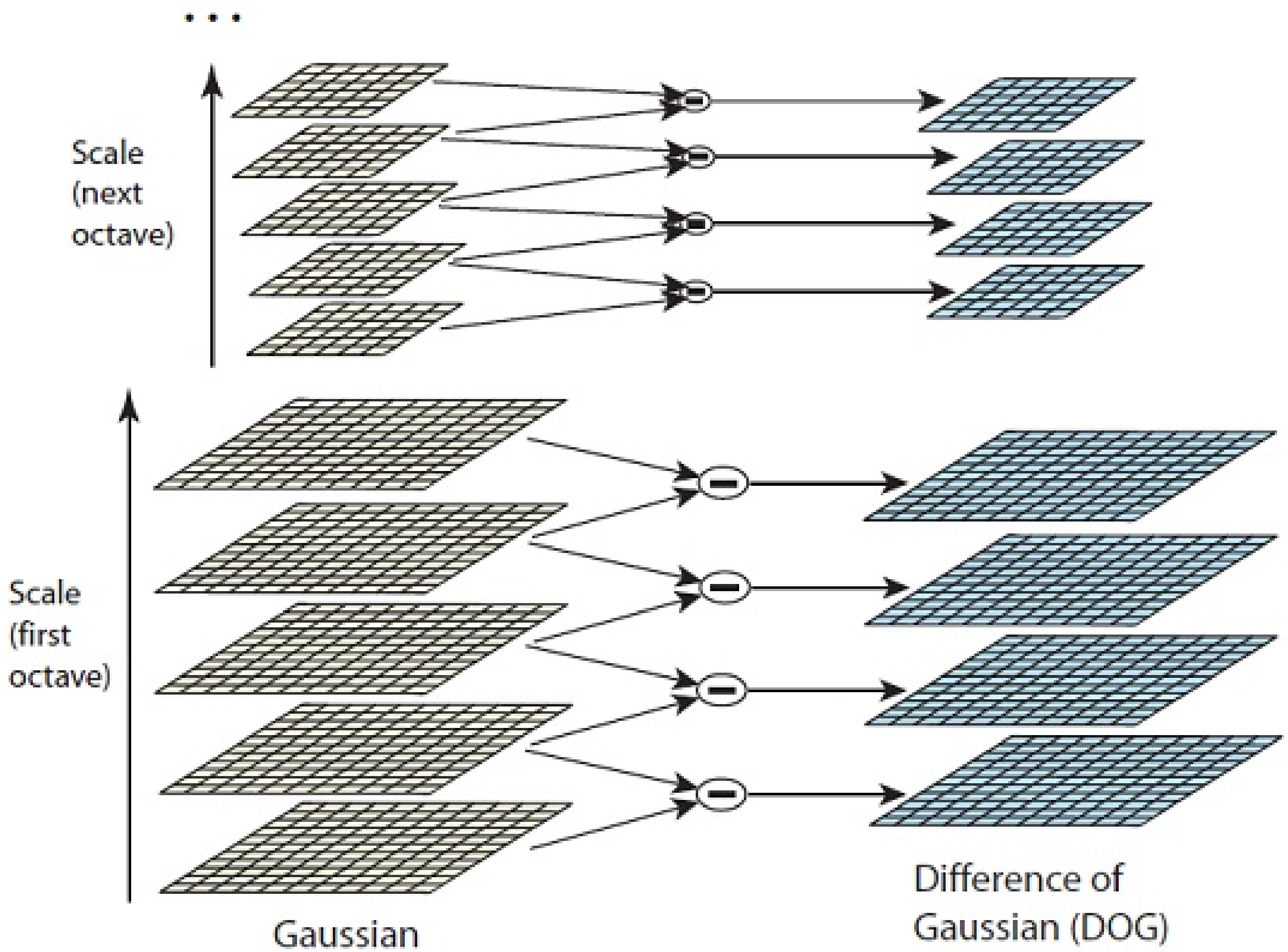
- Take the original image and blur it repetitively with a Gaussian operator

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

- Reduce its size by half and repeat
- Reduce *that* size by half and repeat and so on…
- Each size is referred to as an octave

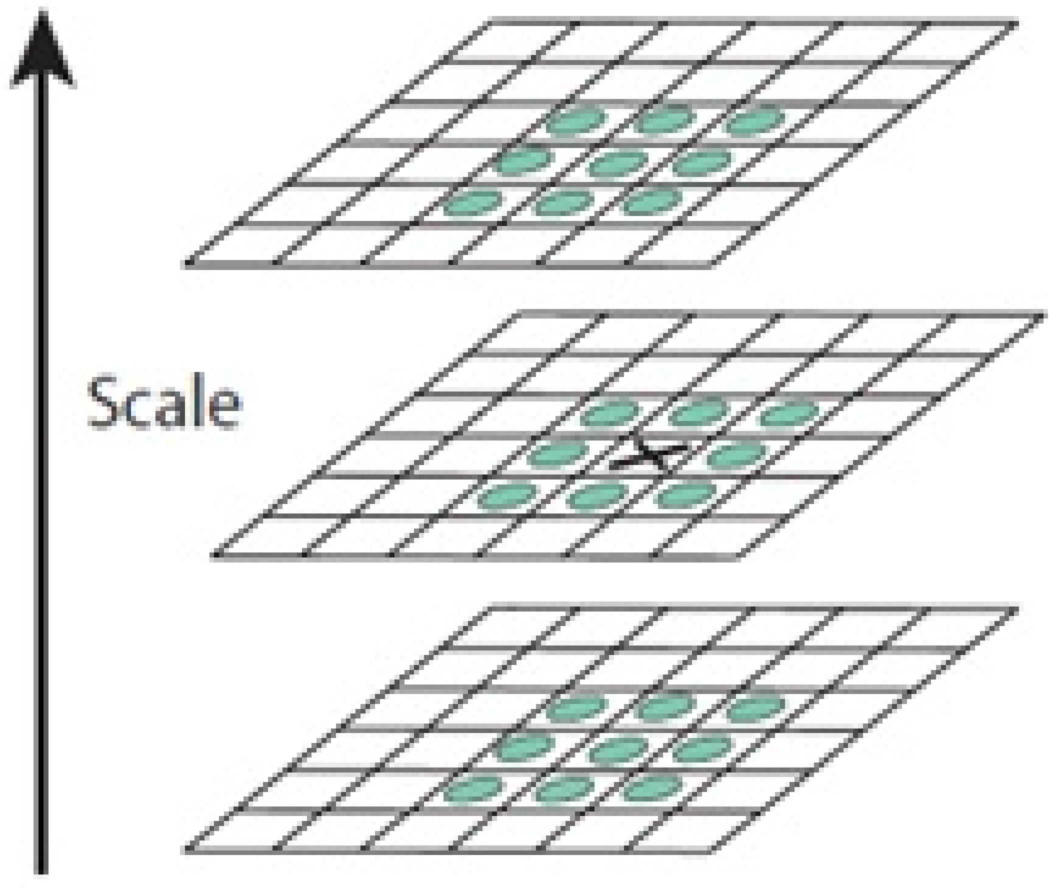
Recommended: http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/

SIFT Feature Detection (cont'd)
Step 2: Get DoG Images



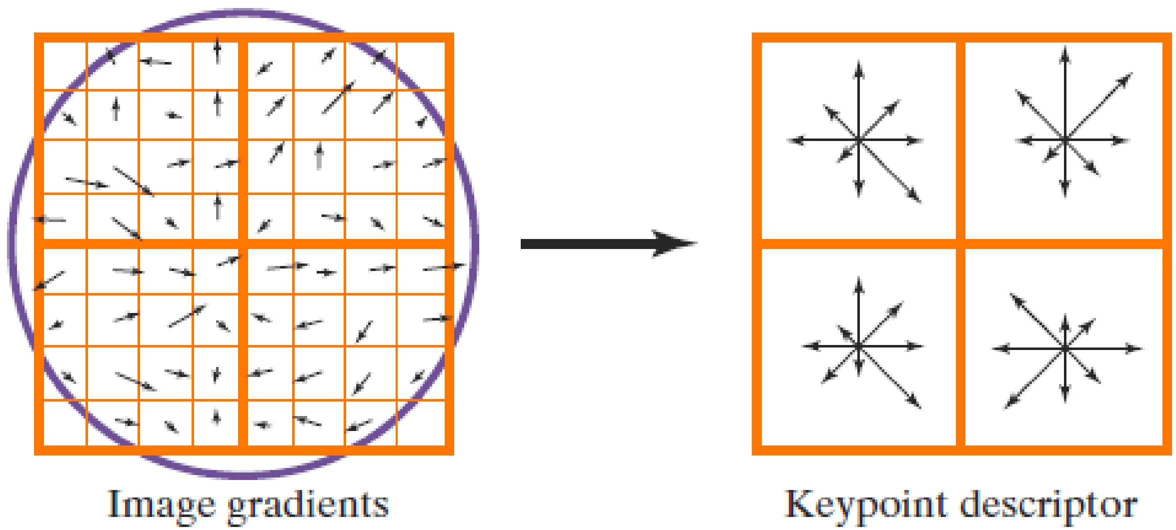
SIFT Feature Detection (cont'd)

Step 3: Find and filter through keypoints



SIFT Feature Descriptor

Figure 7.18 The SIFT feature descriptor is computed by accumulating the orientations of the gradient vectors in a neighborhood of the feature point into a 3D array over position and orientation.

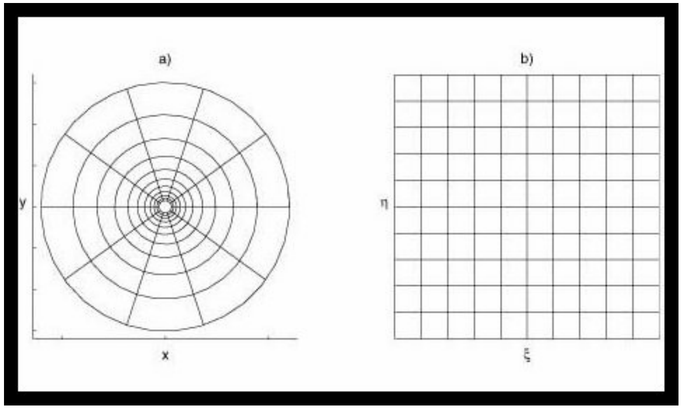


International Journal of Computer Vision, "Distinctive image features from scale-invariant keypoints," Volume 60(2), 2004, pages 91–110, D. G. Lowe, Copyright © 2004, Kluwer Academic Publishers. With permission of Springer.

Gradient Location and Orientation Histogram (GLOH)

- An extension of the SIFT descriptor is the **gradient location and orientation histogram (GLOH)**.

- Instead of using a rectangular grid of pixels, a log-polar grid is used to specify 17 spatial bins from 2 annuli and 8 orientations, in addition to one bin in the center.



http://users.isr.ist.utl.pt/~alex/Projects/TemplateTracking/logpolar.htm

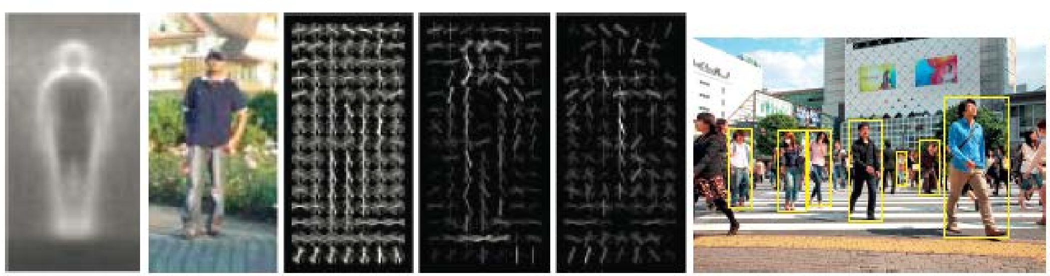
Gradient Location and Orientation Histogram (GLOH) (cont'd)

Figure 7.19 SIFT feature matching results. SIFT feature descriptors from the query image (middle) are matched against descriptors from the database (left) to detect objects at various poses and lighting conditions, and even with severe occlusion (right).

Histogram of Oriented Gradients (HOG)

- **Histogram of oriented gradients (HOG)**: a vector of concatenated histograms of gradient orientations.

- The term HOG is usually reserved for a descriptor computed over a dense rectangular region of the image rather than just at a feature point.



© 2005 IEEE. Reprinted, with permission, from N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2005; Tupungato/Shutterstock.com.

Figure 7.22 Histograms of oriented gradients (HOGs) are widely used for pedestrian detection.

Questions?

Slide Credits

Presentation slides include those of:

- Linda Shapiro, George Stockman, Simon Baker, Trevor Darrell, Cordelia Schmid, David Lowe, Darya Frolova, Denis Simakov, Robert Collins, Yung-Yu Chuang and Jiwon Kim