Feature detection, extraction, matching

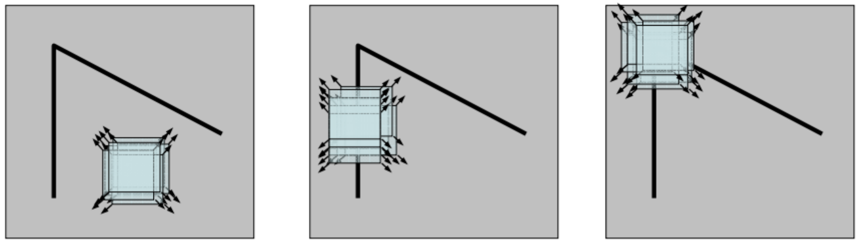
Features

- A vectorial representation of objects/patterns/areas in an image
- Features characterize color, shape, texture, etc.
- A good feature is robust
- **Feature detection** – find points of interests in the image: e.g.: corners
- **Feature extraction** – based on keypoints or features, give a numerical representation of the pixels-of-interest and their neighborhood.
  - Can be done globally at image level
  - Or divide image in patches
  - Use keypoints location as guide for localizing a neigborhood

Feature Detection

- A good feature is robust to changes in illumination, geometry (translation, rotation), viewpoint and scale.
- Algorithms:
  - Harris Corner Detection – scale-sensitive
  - Scale Invariant Feature Transformation (SIFT) – multiscale capabilities
    - There are many SIFT extensions: SURF, BRISK, ORB, KAZE…

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 $[u,v]$:

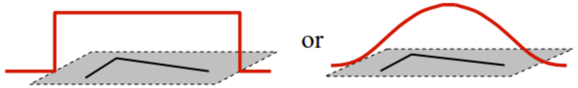
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

Window function $w(x,y) =$



1 in window, 0 outside

Gaussian

Harris Detector: Mathematics

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

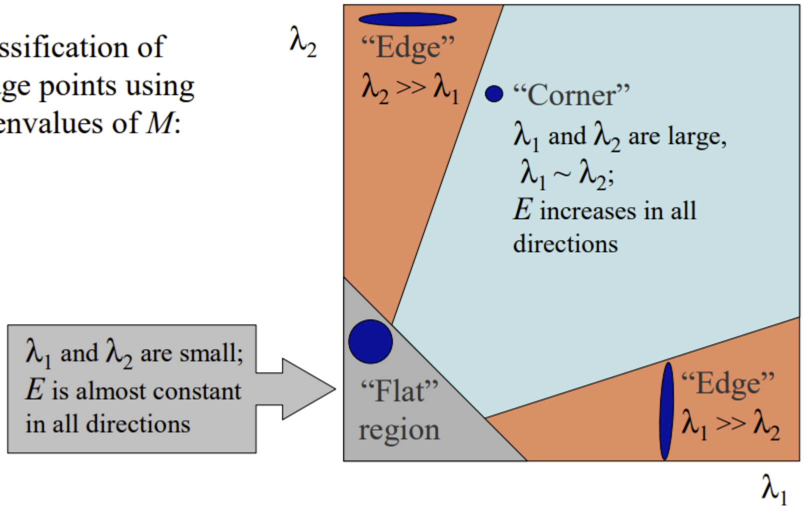
$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

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

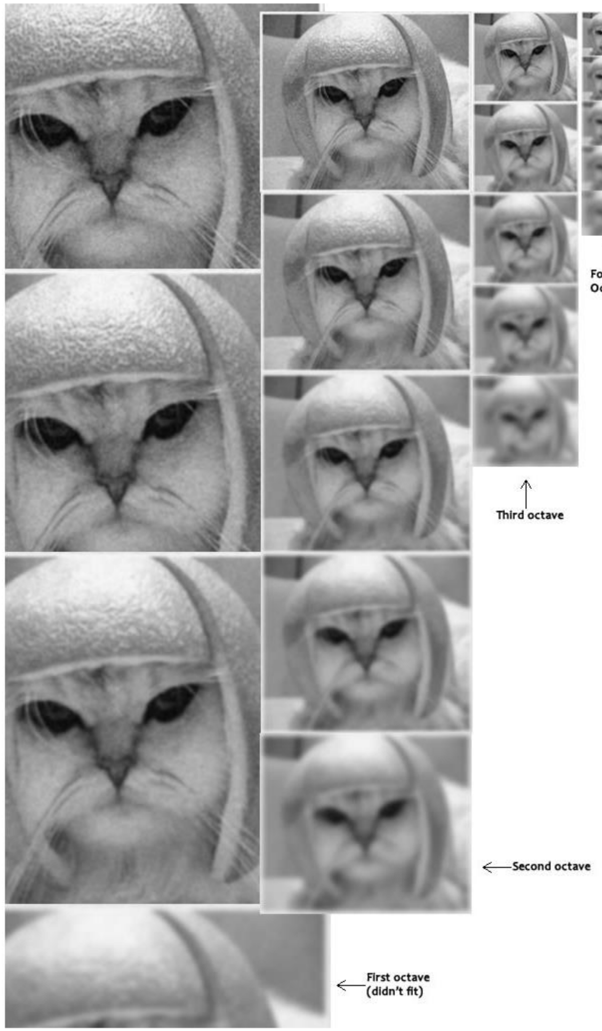
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

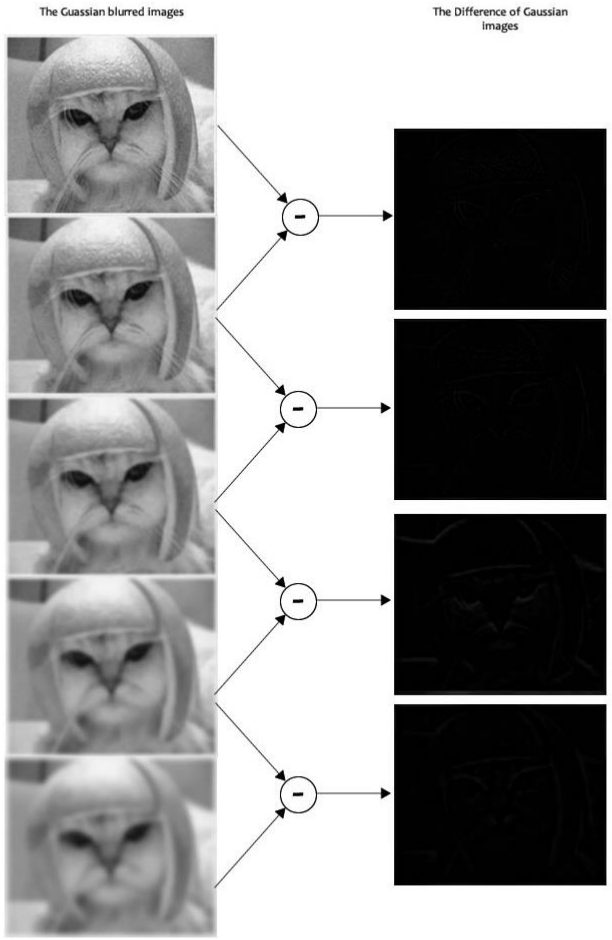
Harris Detector: Mathematics

Classification of image points using eigenvalues of M:



SIFT operates in a scale space



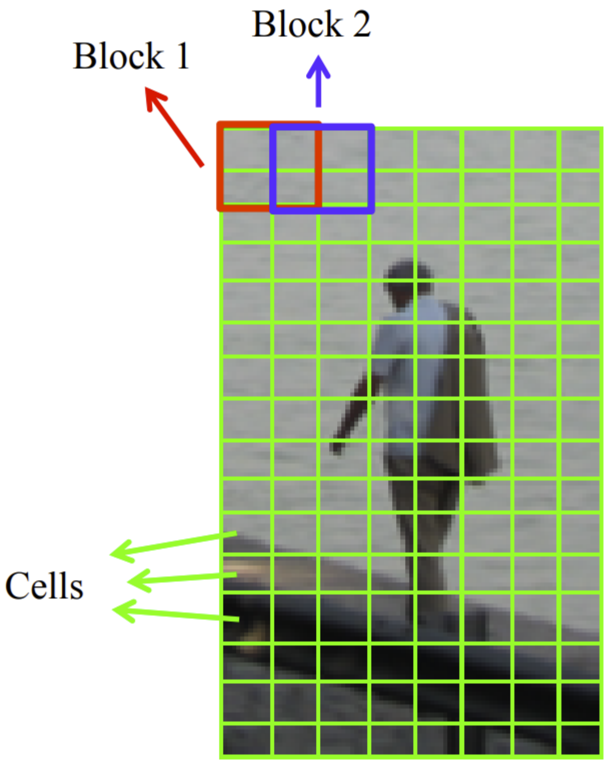


Feature extraction

- Compute a numerical representation around important points of interest and their neighborhood: pixel intensity value, gradient orientation, magnitude, histogram, etc
- Most of the feature detection techniques (SURF, ORB, etc.) have their own way of computing a descriptor.
- An important feature extraction technique is **Histogram of Oriented Gradients**.

Histogram of Oriented Gradients

- Proposed by Dalal and Triggs in 2005



HOG Steps

- HOG feature extraction
  - Compute centered horizontal and vertical gradients with no smoothing
  - Compute gradient orientation and magnitudes
    - For color image, pick the color channel with the highest gradient magnitude for each pixel.
  - For a 64x128 image,
  - Divide the image into 16x16 blocks of 50% overlap.
    - 7x15=105 blocks in total
  - Each block should consist of 2x2 cells with size 8x8.
  - Quantize the gradient orientation into 9 bins
    - The vote is the gradient magnitude
    - Interpolate votes bi-linearly between neighboring bin center.
    - The vote can also be weighted with Gaussian to downweight the pixels near the edges of the block.
  - Concatenate histograms (Feature dimension: 105x4x9 = 3,780)

Feature matching

- Given common objects in two scenes, map the correspondences between the same object.
- Application: image registration, image stitching (panoramic photos), template matching.

Classification

- Uses as well features.
- A classifier is a “function” that separates between features based on their similarity or dissimilarity.
- To be done next week.