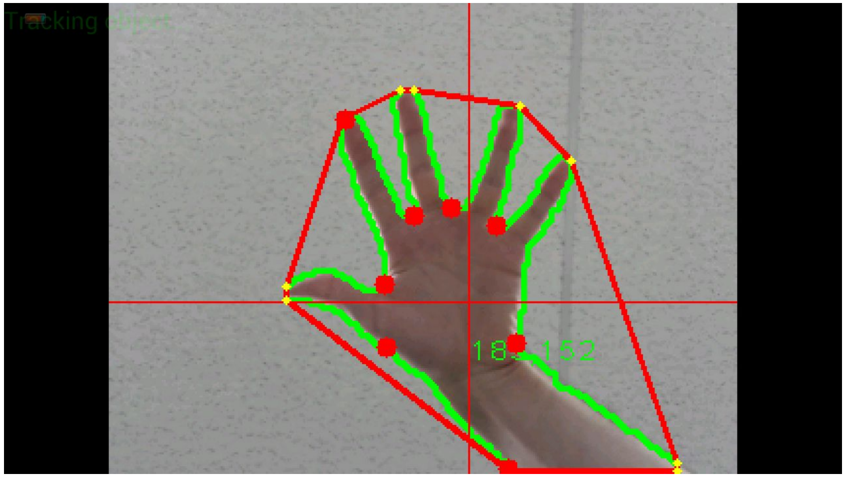
Practical AI: Classic computer vision. Models

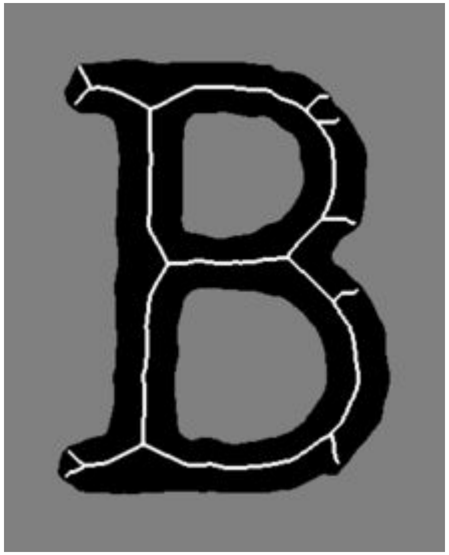
Stanislav Protasov for
Harbour.Space University

Agenda

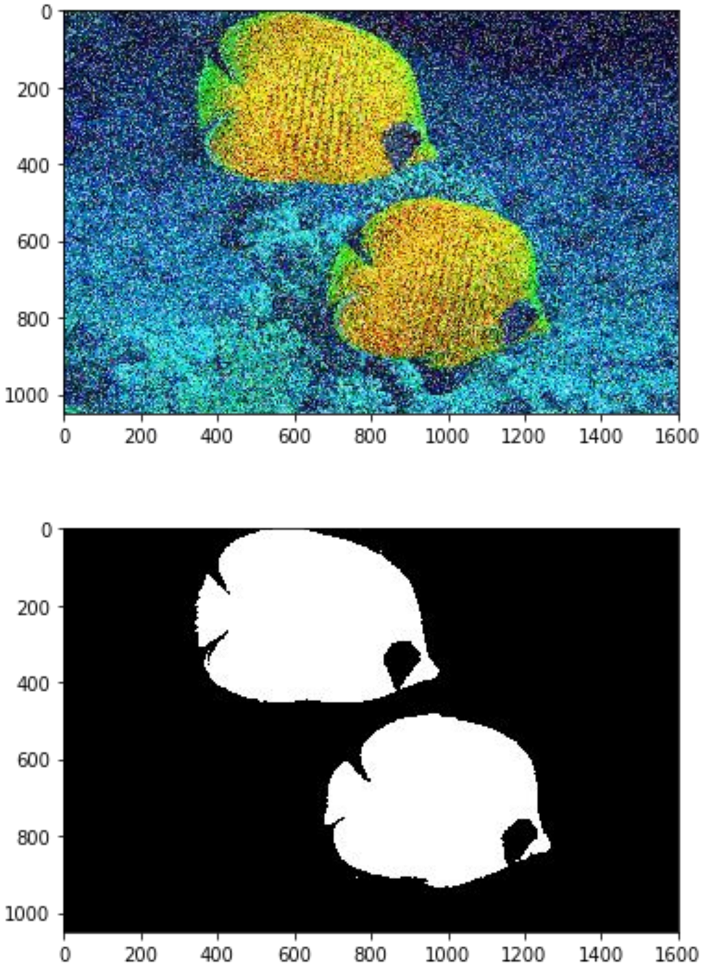
- Model
- Blobs, hulls and skeletons
- Parametric lines
- Primitive feature detectors
- Bag of features

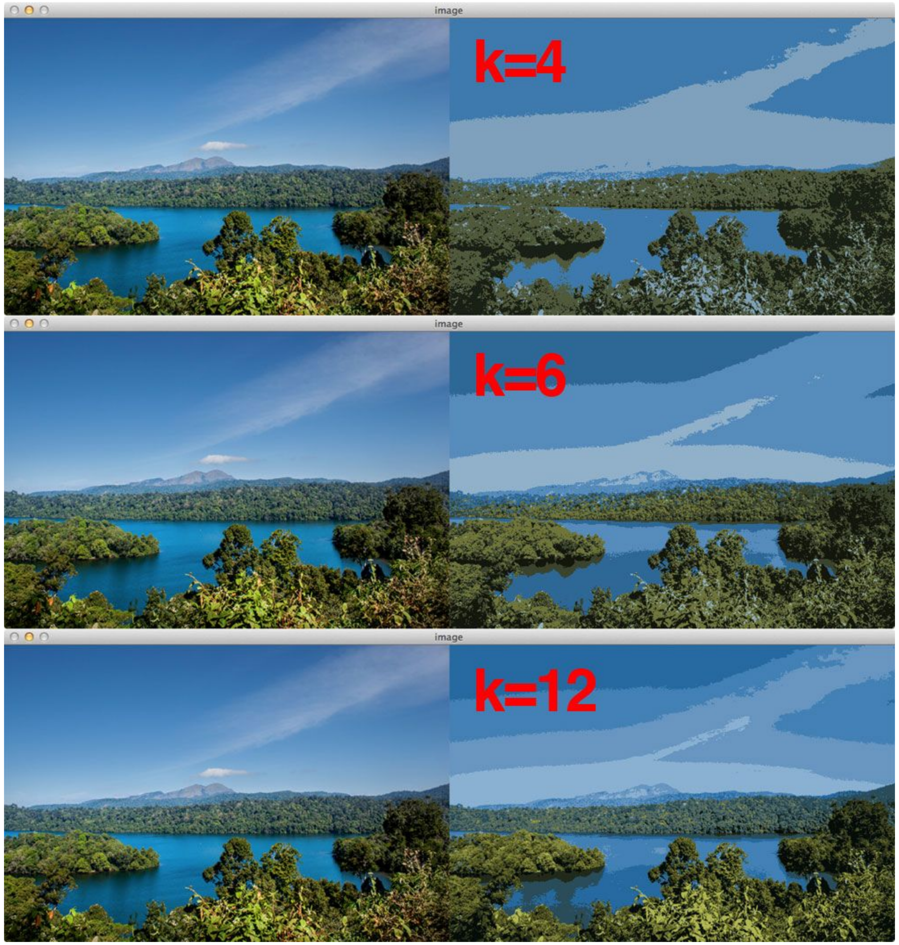
CV techniques: contours, hulls, areas and skeletons



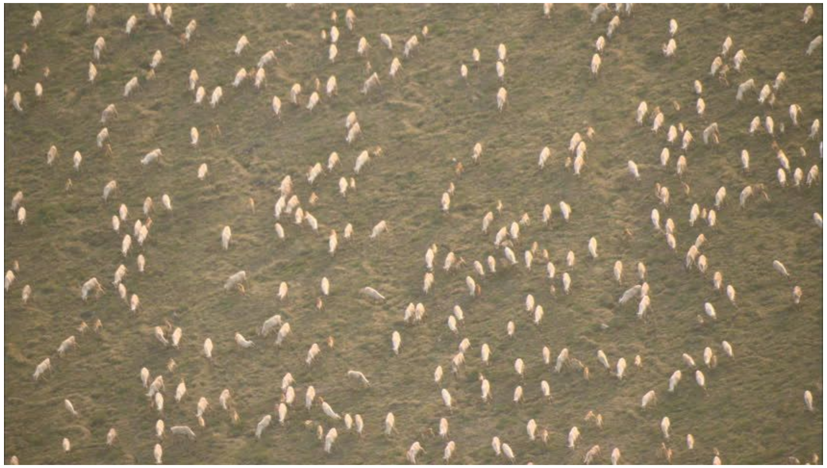


CV techniques: color thresholding and clustering





Lab #1: animal counting

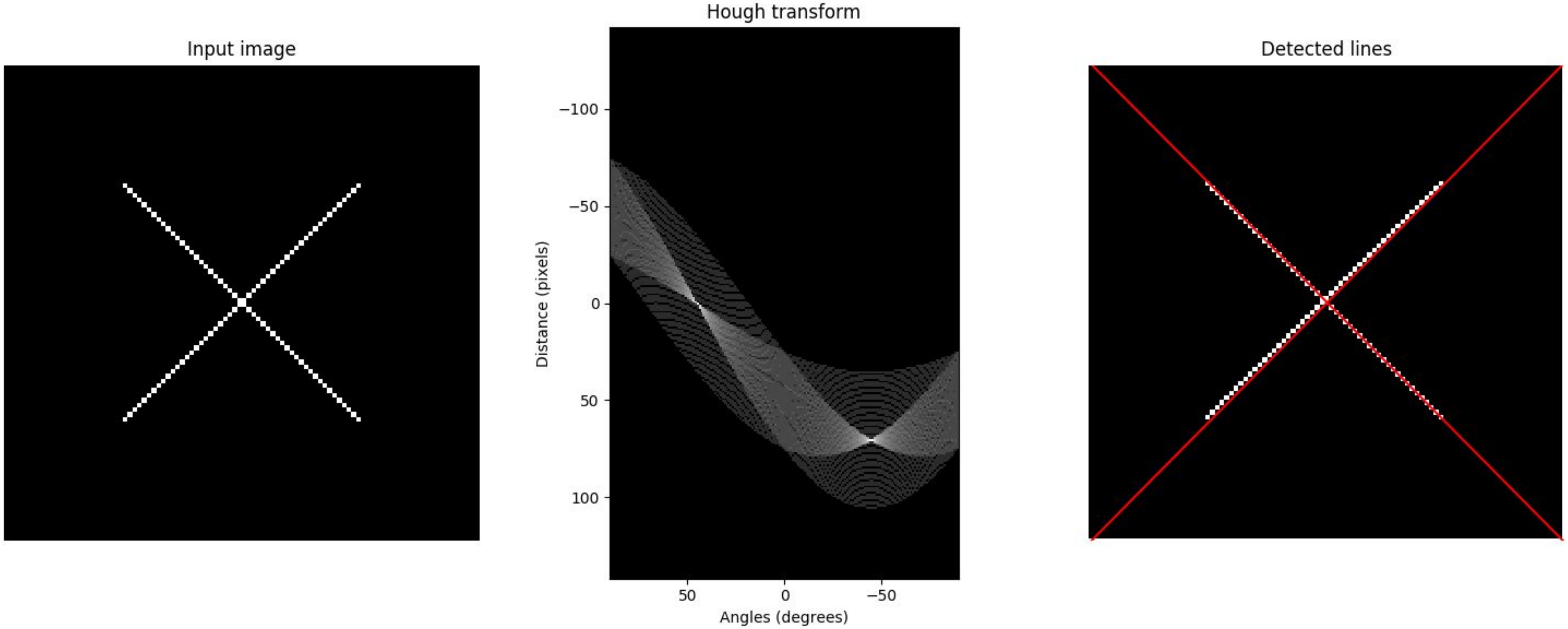


1) Write a script that automatically counts animals at image
   a) Binarize an image
   b) Find connected components
   c) Filter by size, form, color, ...
   d) Count!
2) (*) Write down parameters (number) of your solution. Think, which of them can be estimated automatically?

Parametric models

CV techniques: straight lines

Hough transform

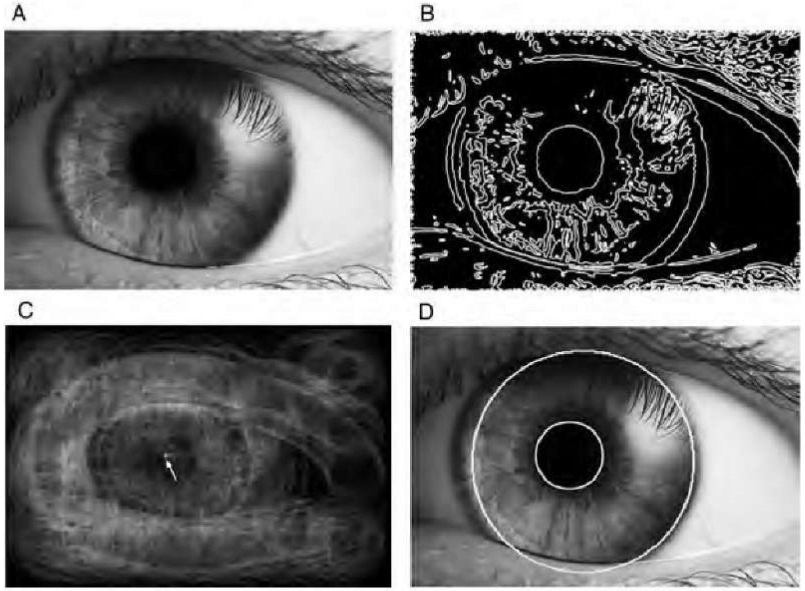


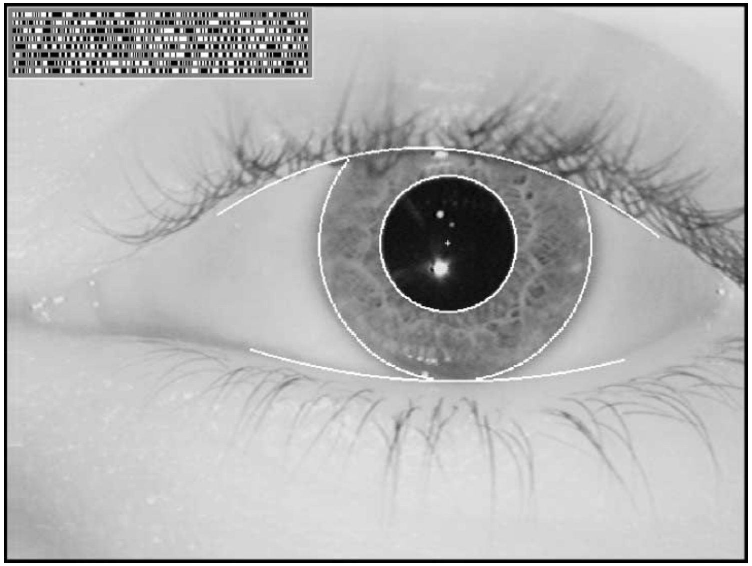
$$r = x\cos\theta + y\sin\theta$$

CV techniques: circles

Hough transform

DAugman detector





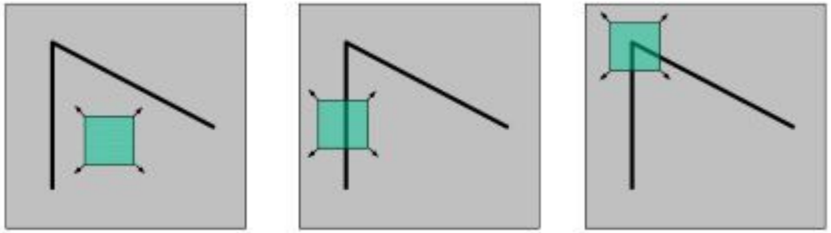
Lab #2. Restore a graph

You are given a drawing of the graph. Restore graph nodes and connections in the graph!

https://github.com/hsu-ai-course/hsu.ai/blob/master/code/09.%20Graph%20lab.ipynb

Features and keypoints

Harris corner detector



"flat" region:
no change in all
directions

"edge":
no change along the
edge direction

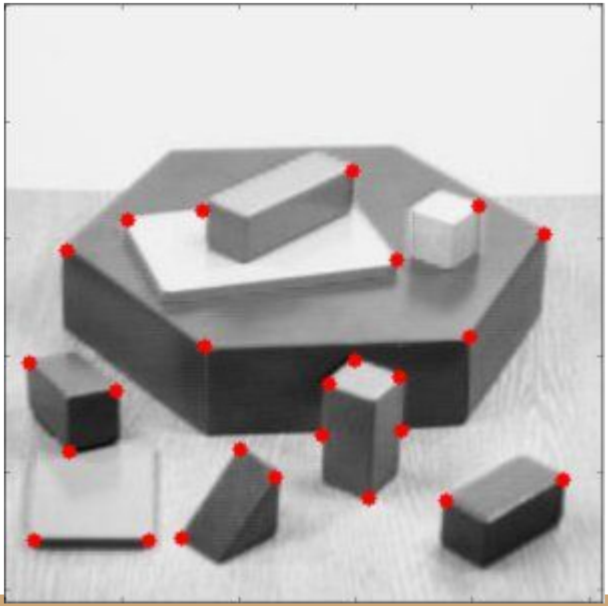
"corner":
significant change in
all directions

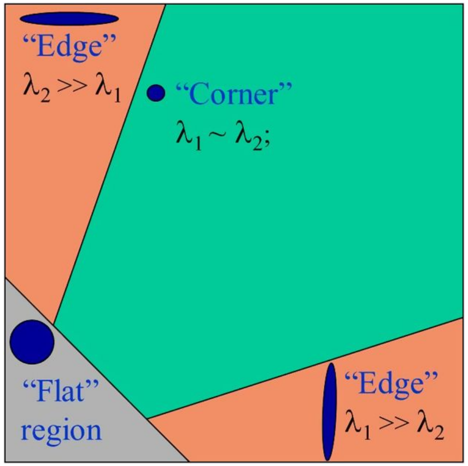
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

$$R = \det M - k(trace\, M)^2$$

$$\det M = \lambda_1 \lambda_2$$

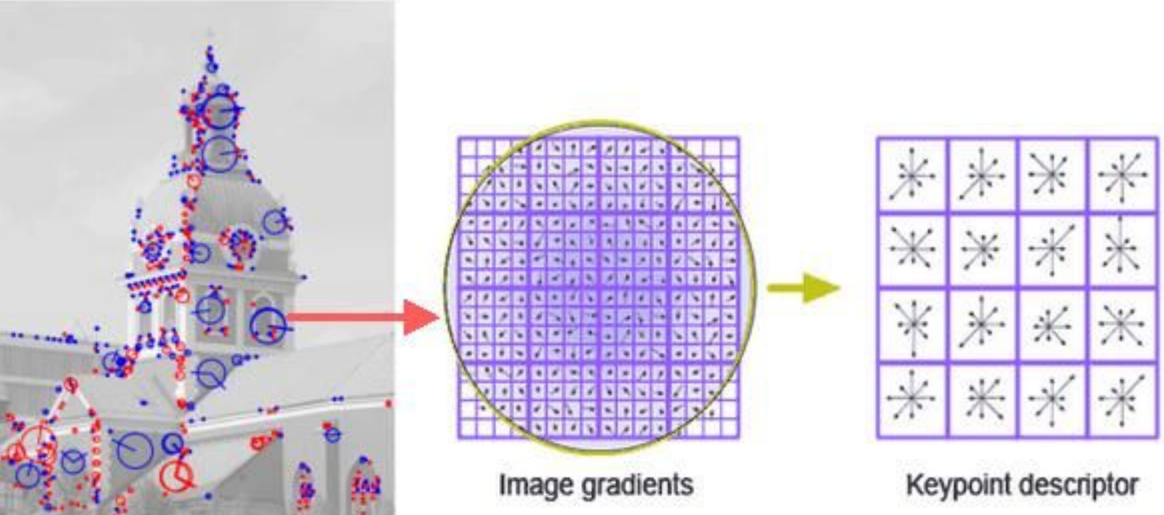
$$trace\, M = \lambda_1 + \lambda_2$$

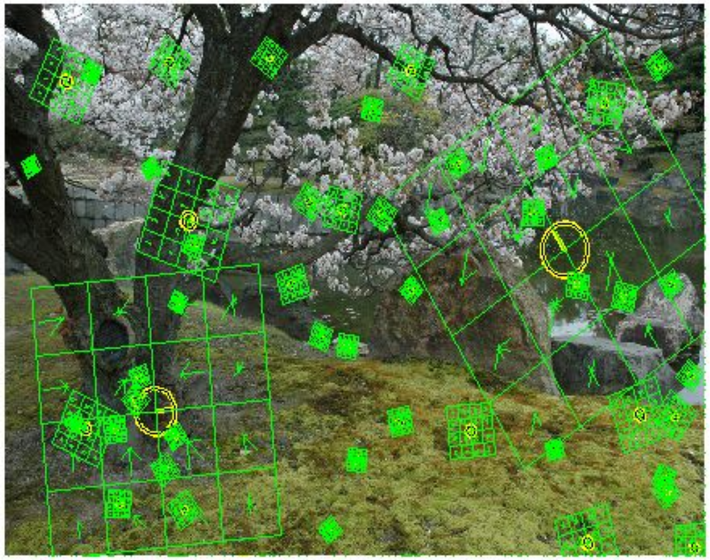




SIFT: Scale-invariant feature transform

1) Compute gradients for different images in *image pyramid* using difference of Gaussians (DoG). Image pyramid ~ Scale invariant
2) Search for local extrema in scale and space (*keypoints*)
3) Compute *direction* (*rotation invariant*)
4) Create descriptor: in 16x16 neighbourhood make 16 blocks, compute gradients (8 bins for angles) and make a vector.
5) Normalize (*intensity invariant*)





Checkers hometask

1) **Recognize field** position on image
    a) **Detect pieces** and colors
    b) Find out pieces **positions**

https://github.com/hsu-ai-course/hsu.ai/tree/master/homeworks/09