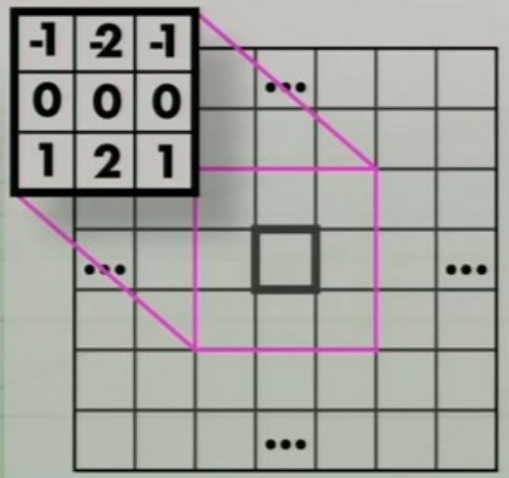
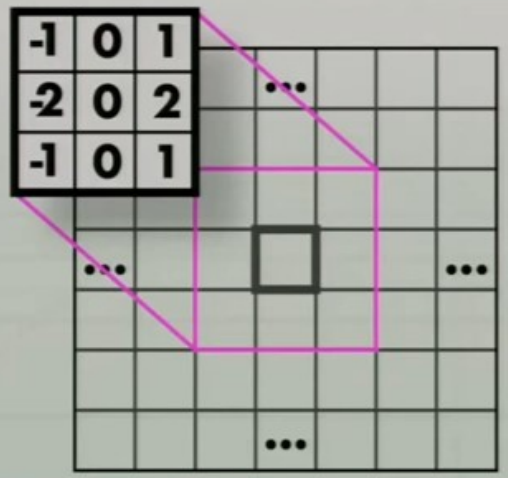
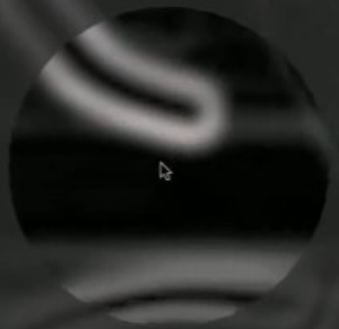
Another approach: gradient magnitude

- Don’t need 2nd derivatives
- Just use magnitude of gradient
- Are we done? No!







What we really want: line drawing

Canny Edge Detection

- Your first image processing pipeline!
  - Old-school CV is all about pipelines

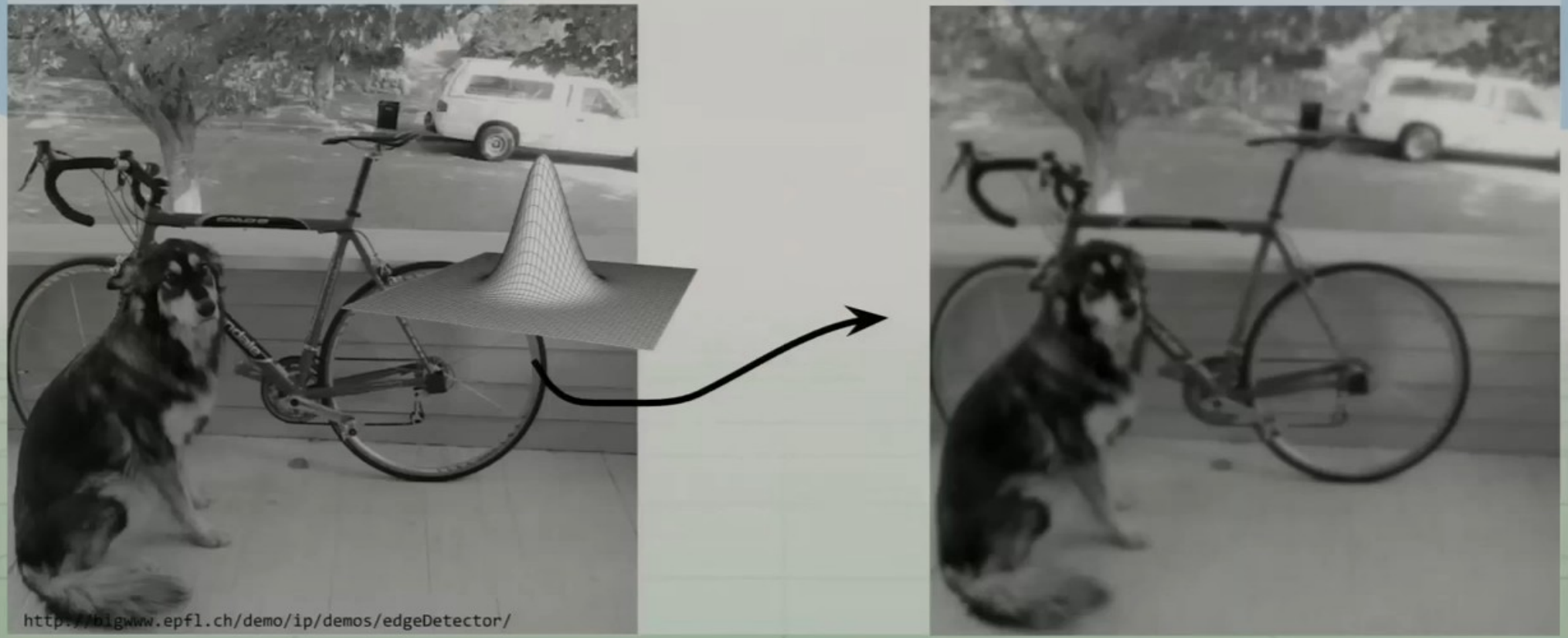
Algorithm:

- Smooth image (only want “real” edges, not noise)
- Calculate gradient direction and magnitude
- Non-maximum suppression perpendicular to edge
- Threshold into strong, weak, no edge
- Connect together components

http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/

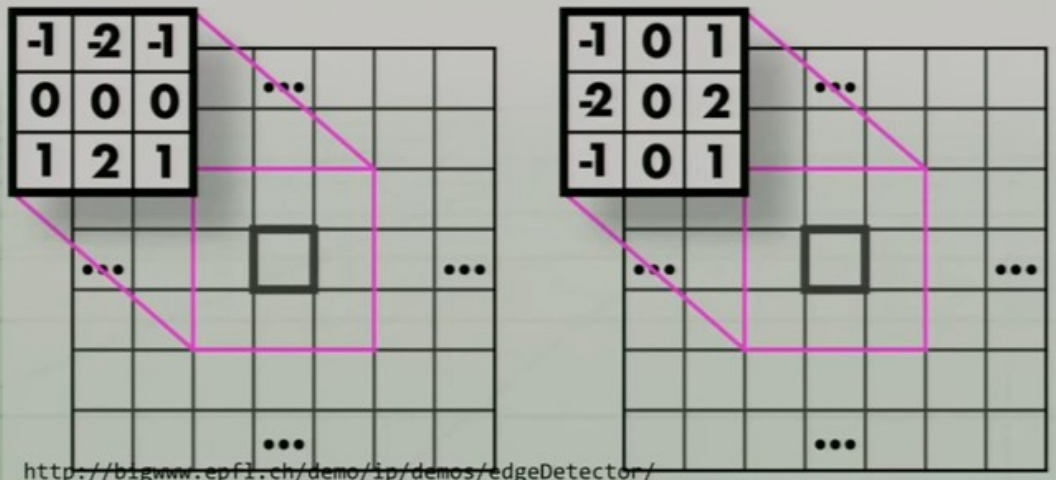
Smooth image

- You know how to do this, gaussians!



Gradient magnitude and direction

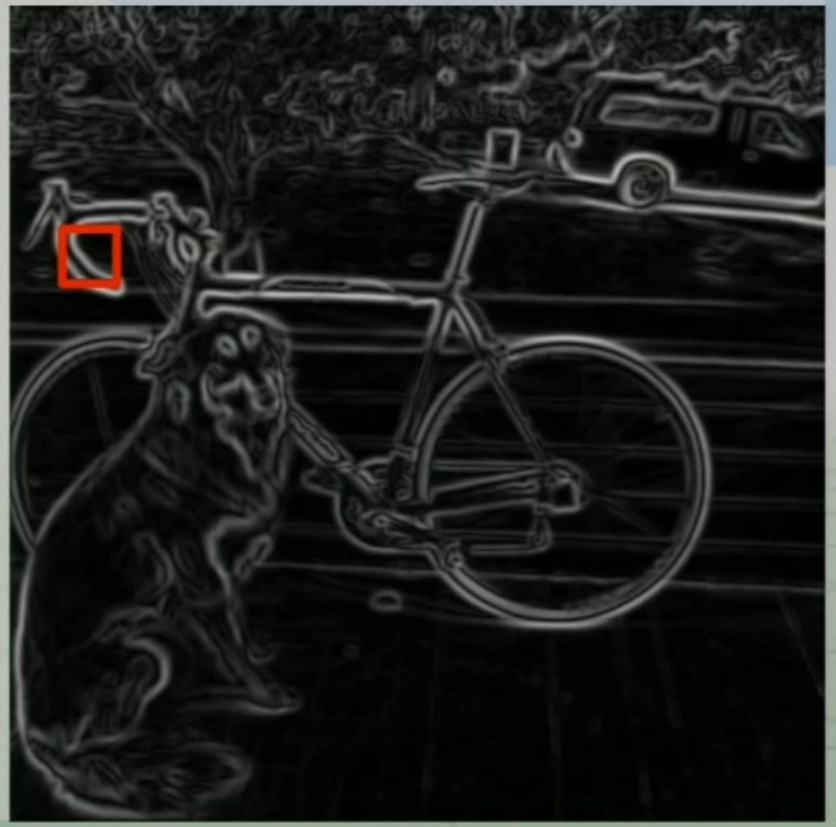
- Sobel filter



http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/

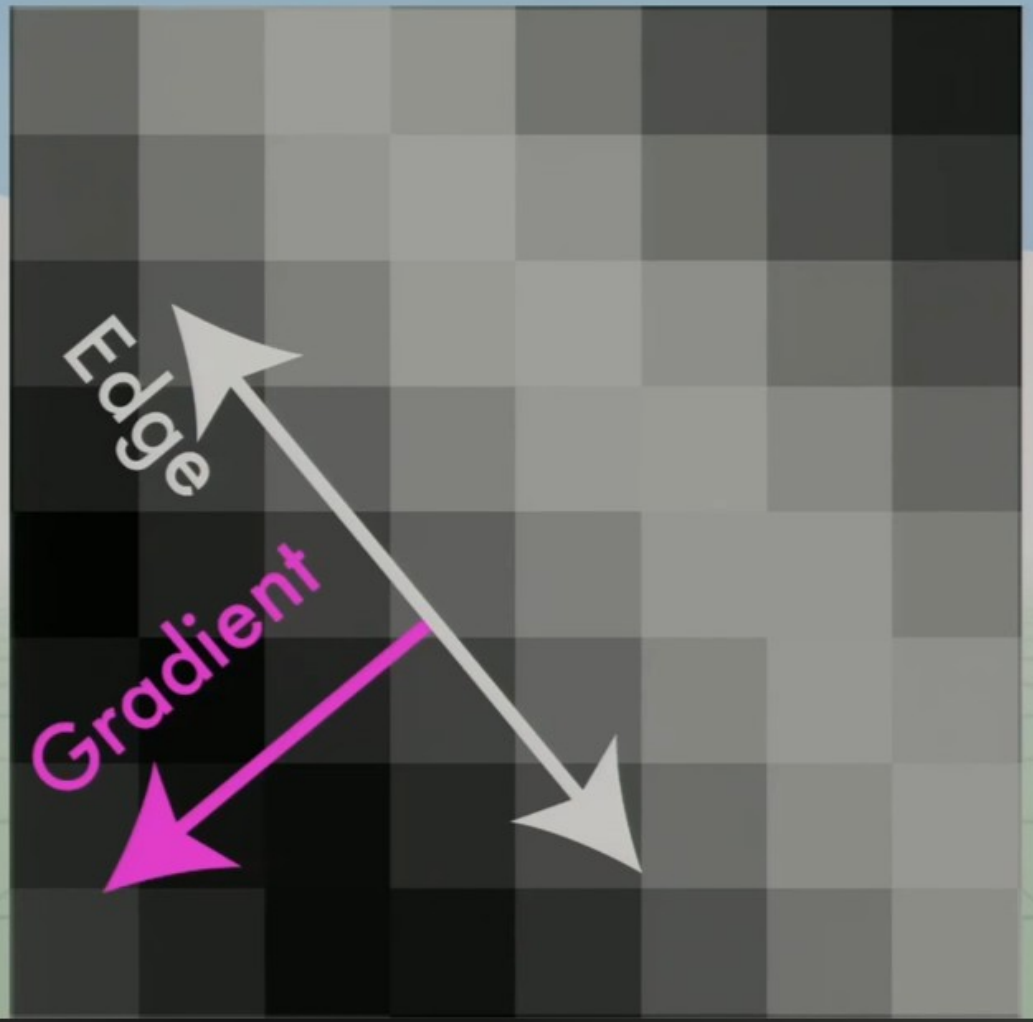
Non-maximum suppression

- Want single pixel edges, not thick blurry lines
- Need to check nearby pixels
- See if response is highest

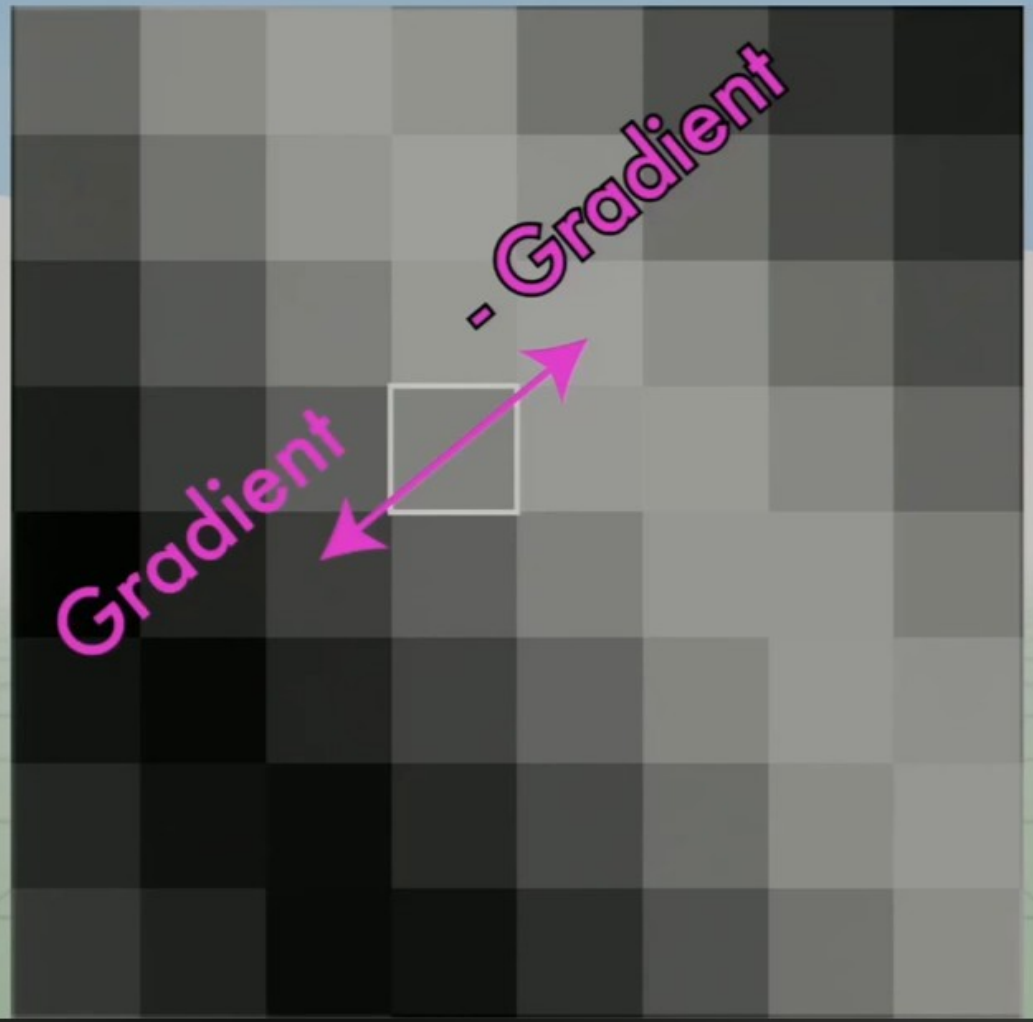


http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/

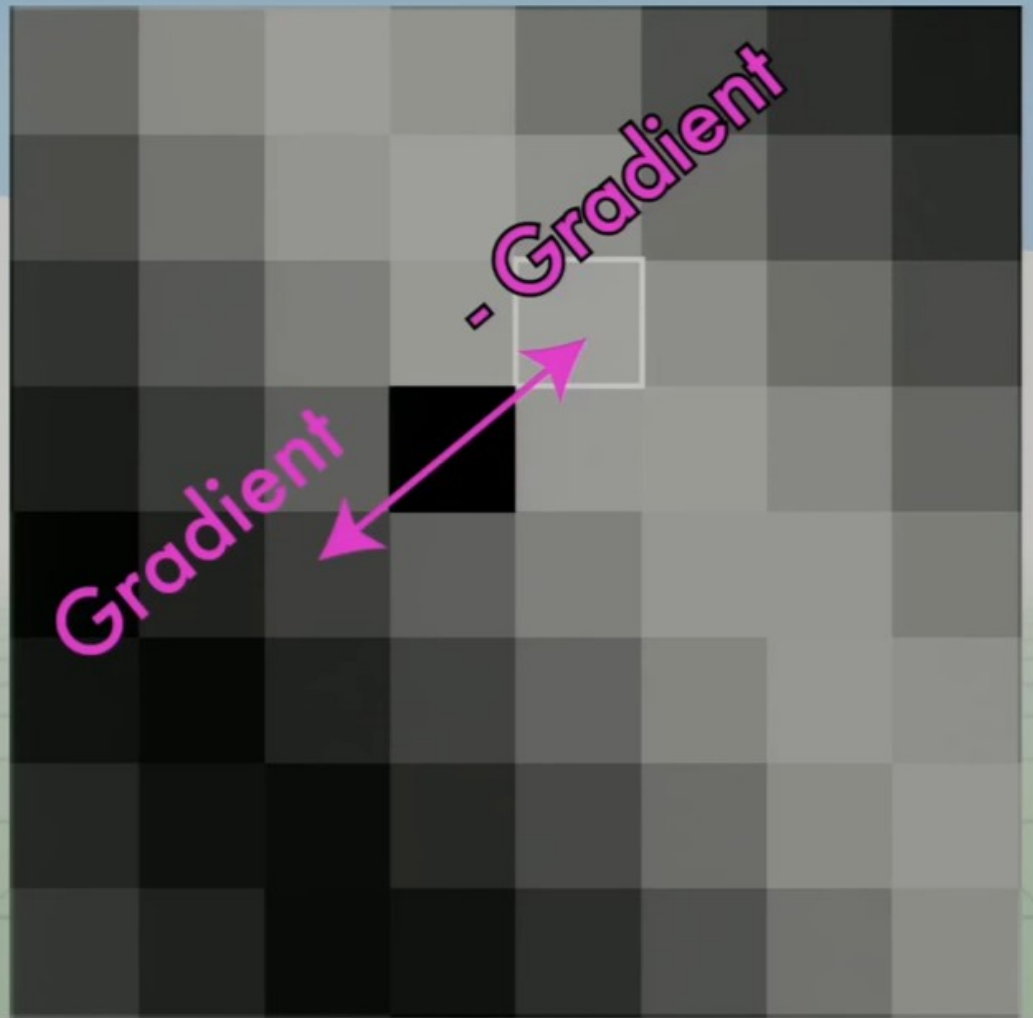
Non-maximum suppression



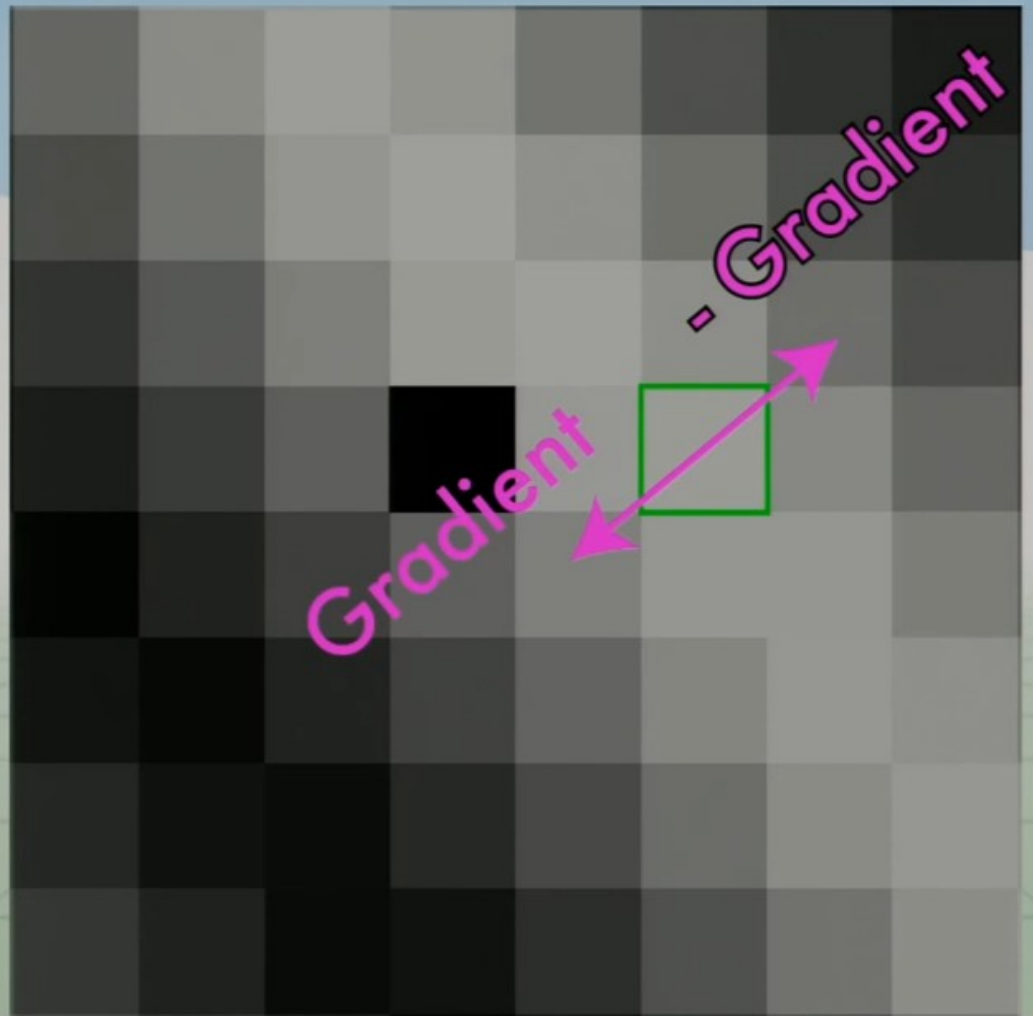
Non-maximum suppression



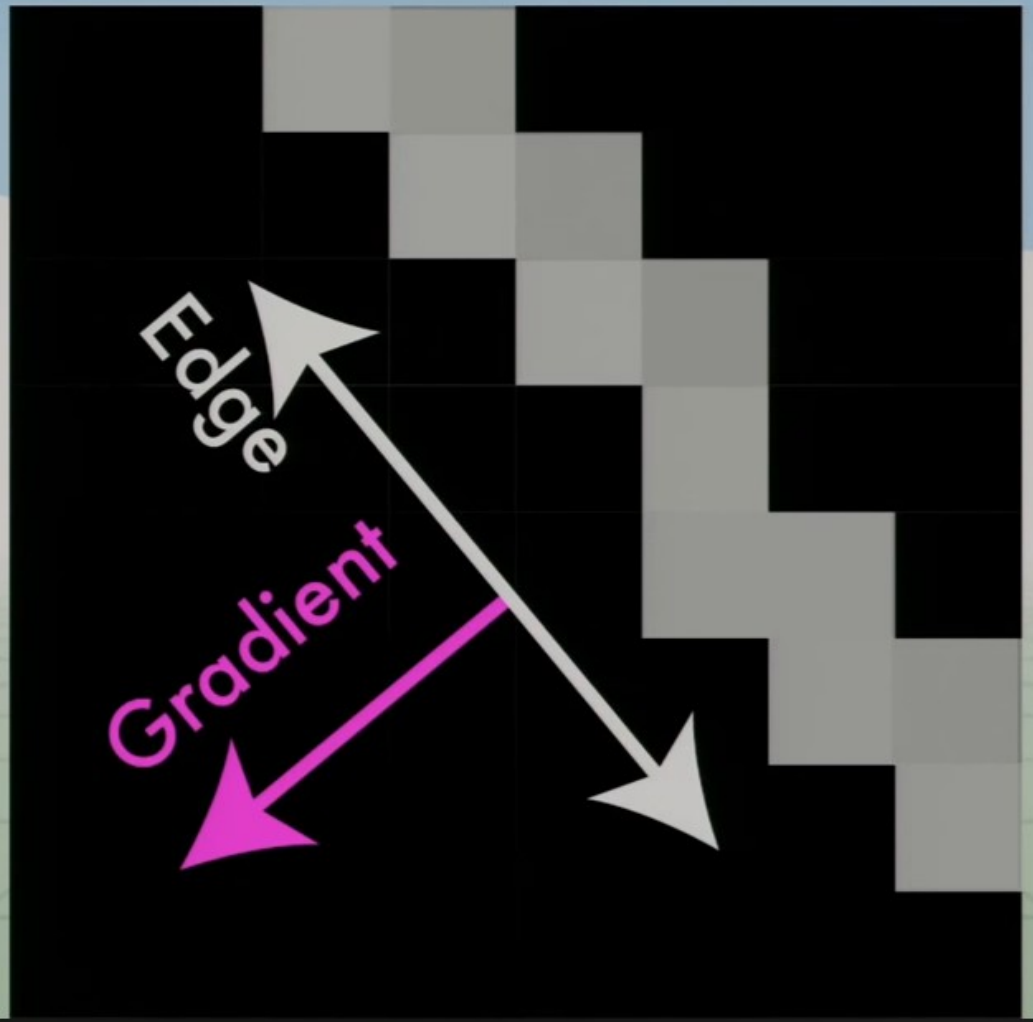
Non-maximum suppression



Non-maximum suppression



Non-maximum suppression



Non-maximum suppression

Threshold edges

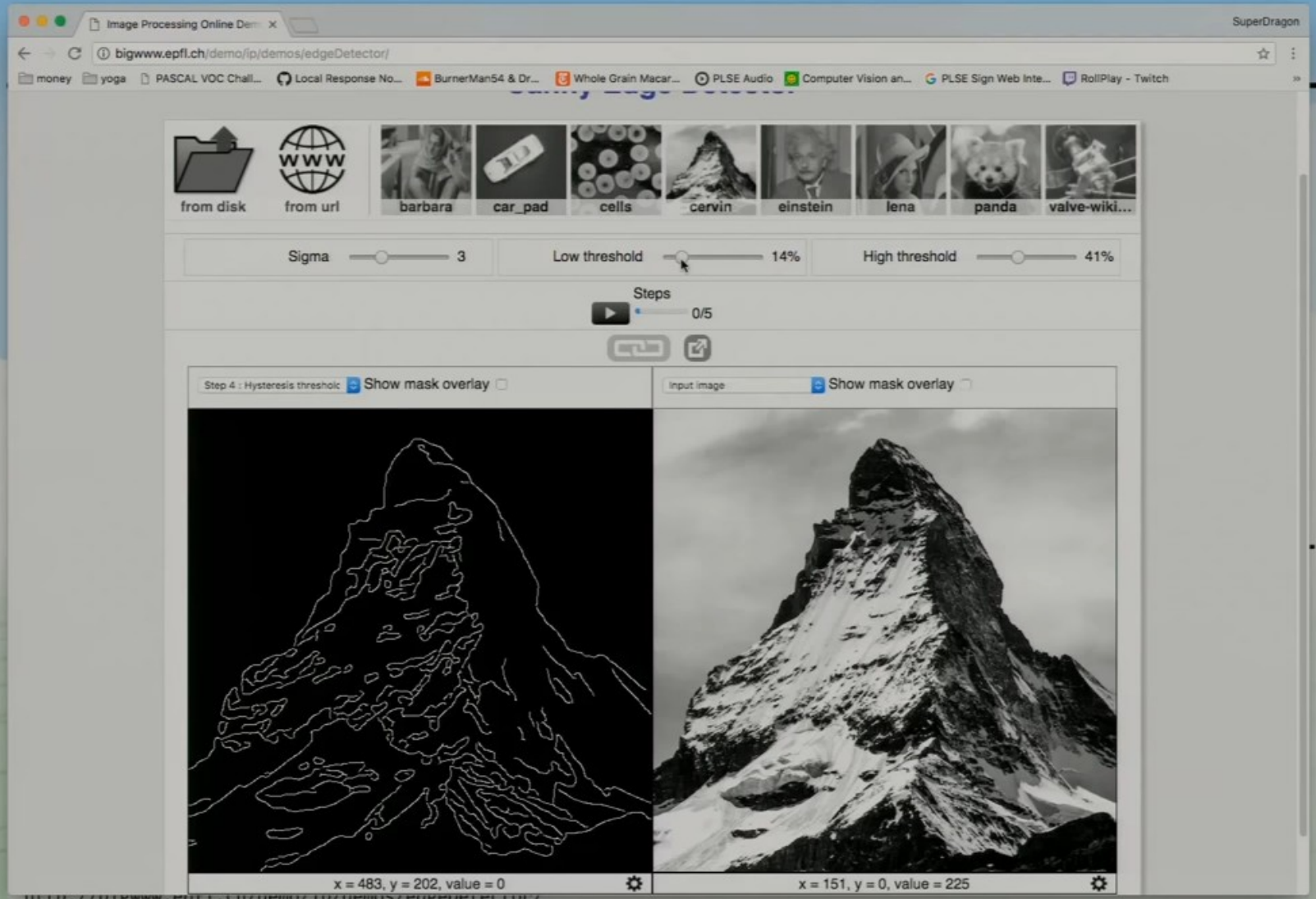
- Still some noise
- Only want strong edges
- 2 thresholds, 3 cases
  - R > T: strong edge
  - R < T but R > t: weak edge
  - R < t: no edge
- Why two thresholds?

http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/

Connect ‘em up!

- Strong edges are edges!
- Weak edges are edges
  iff they connect to strong
- Look in some neighborhood
  (usually 8 closest)

http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/



not noise)

.tude

to edge

Canny Edge Detection