Computer Vision Tutorial 2 - Convolution & Edge Detection

Manuel Heurich - Biorobotics Lab

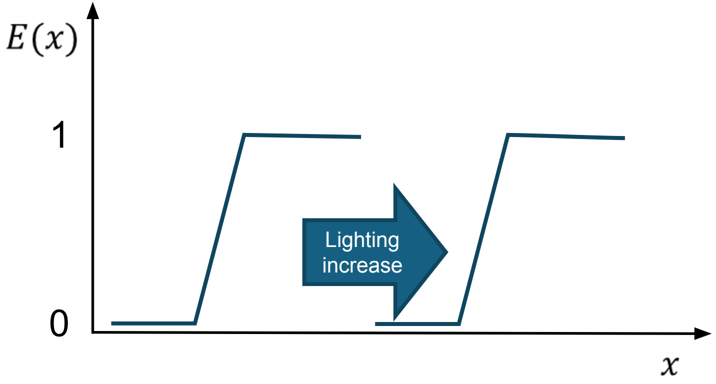
05.11.24

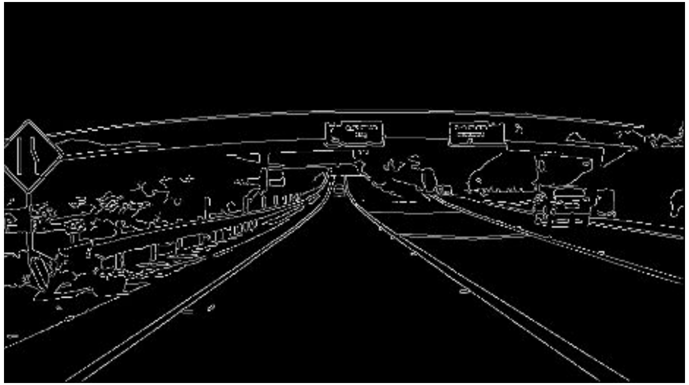
Edges are robust features

- global lighting changes introduce brightness shifts, but local brightness differences remain stable

- edge image = binary image

- detect local brightness changes!



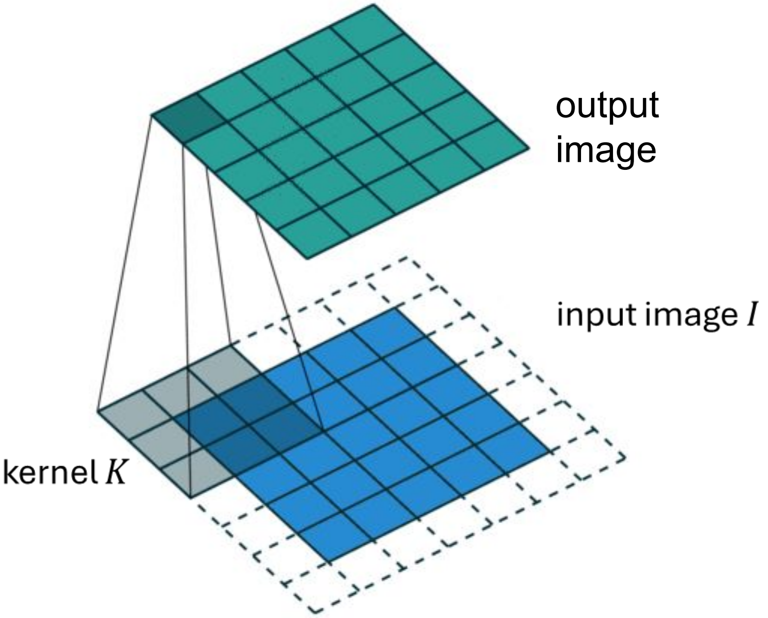


Computer Vision 101: Convolution

$$(I * K)_{x,y} = \sum_{-W}^{W} \sum_{-H}^{H} I_{x-w,y-h} K_{w,h}$$

$$= I_{N(x,y)}^{T} K$$

Notion 1: weighted sum



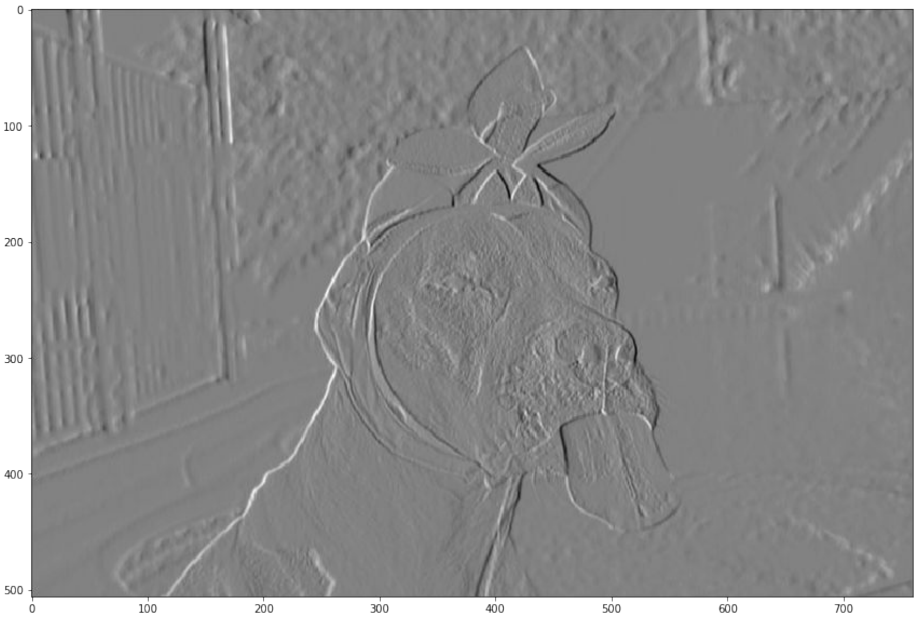
https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215

Convolution

```
K = [[1, 1, 1, 1, 1],
     [1, 1, 1, 1, 1],
     [1, 1, 1, 1, 1]]
```

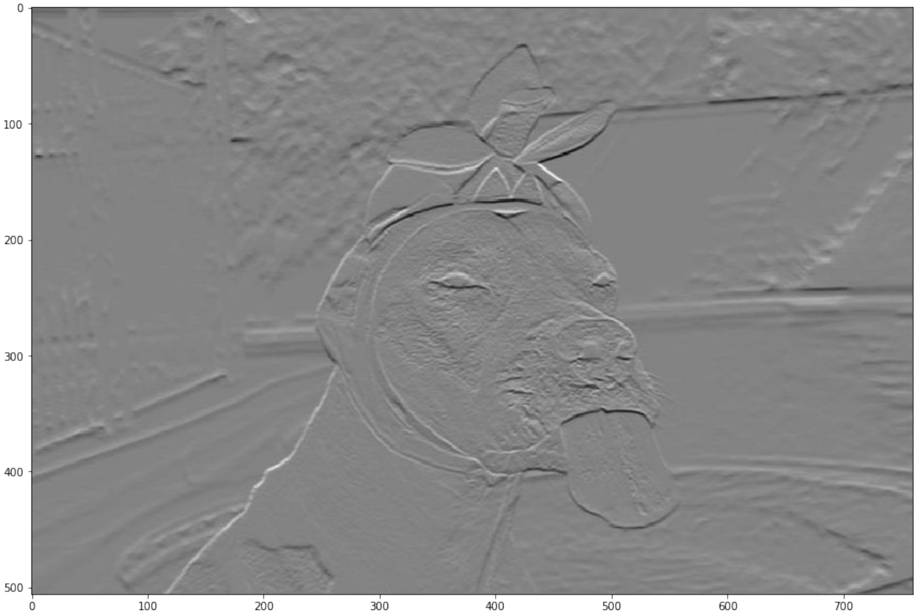
„Detect" edges with Convolution

```
K = [[-1, 0, 1],
     [-1, 0, 1],
     [-1, 0, 1]]
```

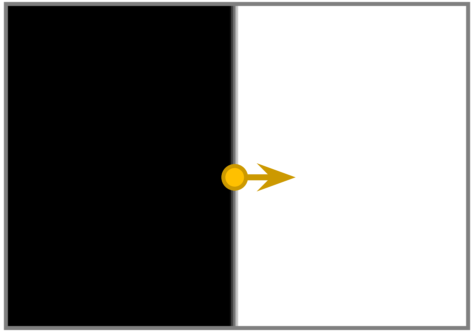


„Detect“ edges with Convolution

$$K = [[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]]^T$$



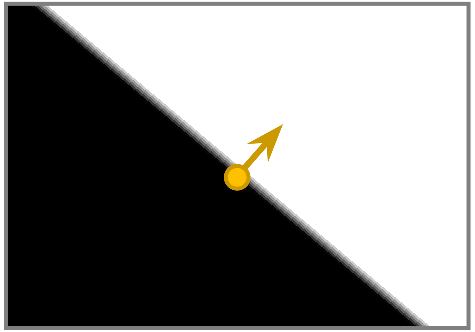
Computing the Image Gradient



$$\nabla I(x,y) = \begin{pmatrix} \frac{\delta I}{\delta x} \\ \frac{\delta I}{\delta y} \end{pmatrix}(x,y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Use Sobel filter to estimate the image gradient!

$$\nabla I(x,y) = \begin{pmatrix} G_x \\ G_y \end{pmatrix}(x,y)$$



$$\nabla I(x,y) = \begin{pmatrix} \frac{\delta I}{\delta x} \\ \frac{\delta I}{\delta y} \end{pmatrix}(x,y) = \begin{pmatrix} ? \\ ? \end{pmatrix}$$

Sobel Kernel, Sobel Filter

```
Sx =[[-1, 0, 1],
     [-2, 0, 2],
     [-1, 0, 1]]
```

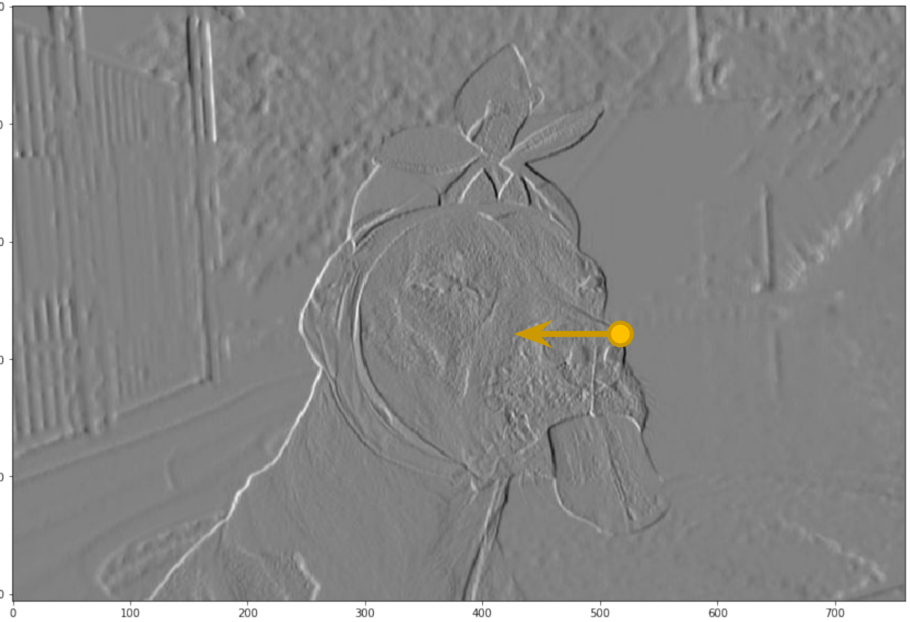
```
Sy =[[-1, -2, -1],
     [ 0,  0,  0],
     [ 1,  2,  1]]
```

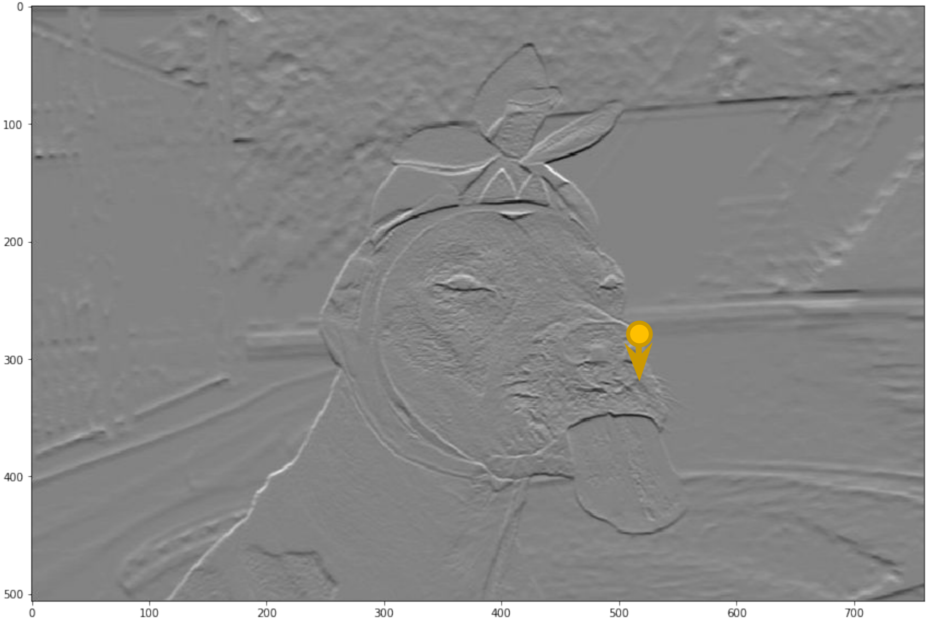
Convolve for derivative in x-direction

$$G_x = (I * S_x)$$

Convolve for derivative in y-direction

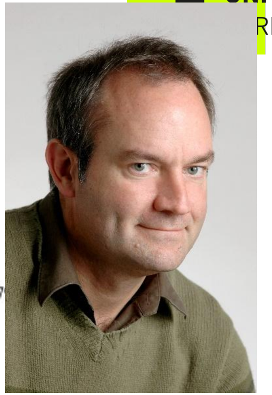
$$G_y = (I * S_y)$$


0
100
200
300
400
500
600
700


0
100
200
300
400
500
0
100
200
300
400
500
600
700

Canny‘s Edge Detection

[Cited by 29138]



IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986 67

A Computational Approach to Edge Detection

JOHN CANNY, MEMBER, IEEE

Abstract—This paper describes a computational approach to edge detection. The success of the approach depends on the definition of a *comprehensive* set of goals for the computation of edge points. These goals must be precise enough to delimit the desired behavior of the detector while making minimal assumptions about the form of the solution. We define detection and localization criteria for a class of edges, and present mathematical forms for these criteria as functionals on the operator impulse response. A third criterion is then added to ensure that the detector has only one response to a single edge. We use the criteria in numerical optimization to derive detectors for several common image features, including step edges. On specializing the analysis to step edges, we find that there is a natural uncertainty principle between detection and localization performance, which are the two main goals. With this principle we derive a single operator shape which is optimal at any scale. The optimal detector has a simple approximate implementation in which edges are marked at maxima in gradient magnitude of a Gaussian-smoothed image. We extend this simple detector using operators of several widths to cope with different signal-to-noise

detector as input to a program which could isolate simple geometric solids. More recently the model-based vision system ACRONYM [3] used an edge detector as the front end to a sophisticated recognition program. Shape from motion [29], [13] can be used to infer the structure of three-dimensional objects from the motion of edge contours or edge points in the image plane. Several modern theories of stereopsis assume that images are preprocessed by an edge detector before matching is done [19], [20]. Beattie [1] describes an edge-based labeling scheme for low-level image understanding. Finally, some novel methods have been suggested for the extraction of three-dimensional information from image contours, namely shape from contour [27] and shape from texture [31].

In all of these examples there are common criteria rel-

Outline of Canny's Edge Detection

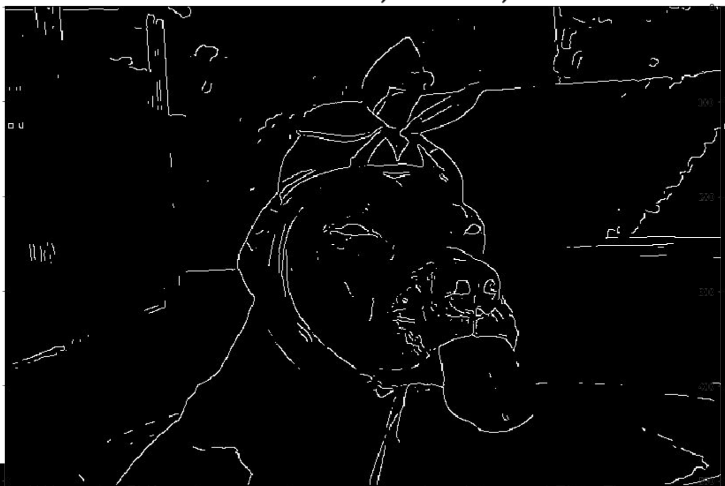
$$A = 1/16 \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

- **convolve** with Gaussian to smooth the image and remove noise
- **convolve** again with Sobel to compute the image gradient
- apply **non-maximum suppression**
- use **two thresholds**: 1) for strong and 2) weak edges
- **hysteresis**: track edge along edge direction and suppress weak edges not connected to strong edges

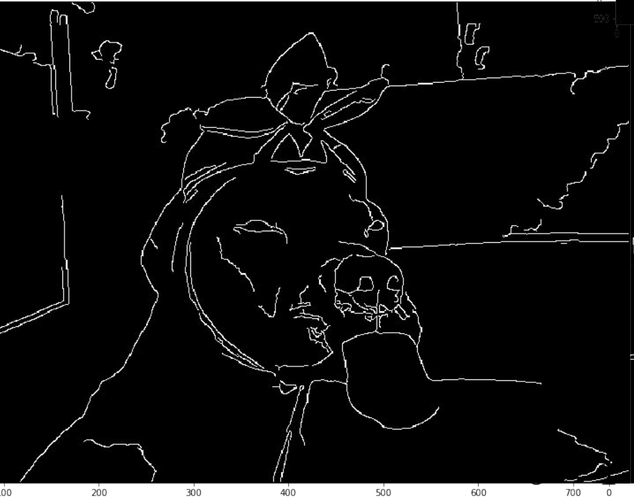
Effect of Low/High Thresholds

(using relative thresholds here:
0.4 means 40% of the maximum
gradient magnitude)

$\sigma = 1, l = 0.4, h = 0.4$



$\sigma = 1, l = 0.4, h = 0.6$



$\sigma = 1, l = 0.2, h = 0.6$

$\sigma = 1, l = 0.2, h = 0.4$

$\sigma = 1, l = 0.2, h = 0.2$