Lecture 7 Further Edge Detection

COMP3204 Computer Vision

What better ways are there to detect edges?

Book pp 154 - 170

Department of Electronics and Computer Science

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Content

1. How can we improve first-order edge detection?
2. How can we detect edges using second order differentiation/ differencing

Applying Sobel operator

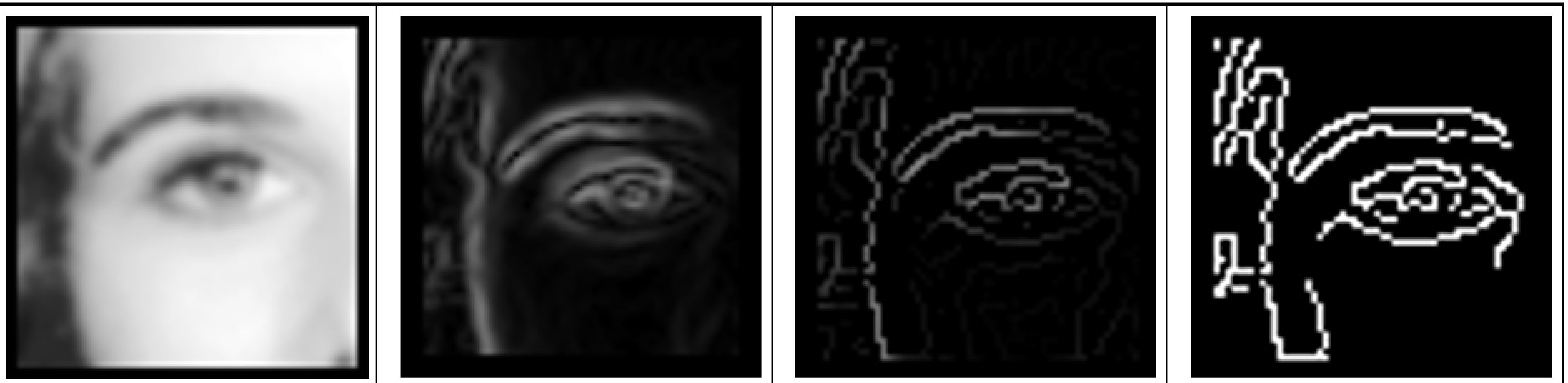
Sobel is a good basic operator

Blurred edges

(a) original image | (b) Sobel edge magnitude | (c) thresholded magnitude

Noisy edges

Stages in Canny edge detection operator



(a) Gaussian smoothing	**(b)** Sobel edge detection	**(c)** non-maximum suppression	**(d)** hysteresis thresholding

Stages in Canny Edge Detection

Canny gives thin edges in the right place, but is more complex

Canny edge detection operator

Formulated with three main objectives:

- optimal detection with no spurious responses;
- good localisation with minimal distance between detected and true edge position; and
- single response to eliminate multiple responses to a single edge.

Approximation

1. use Gaussian smoothing;
2. use the Sobel operator;

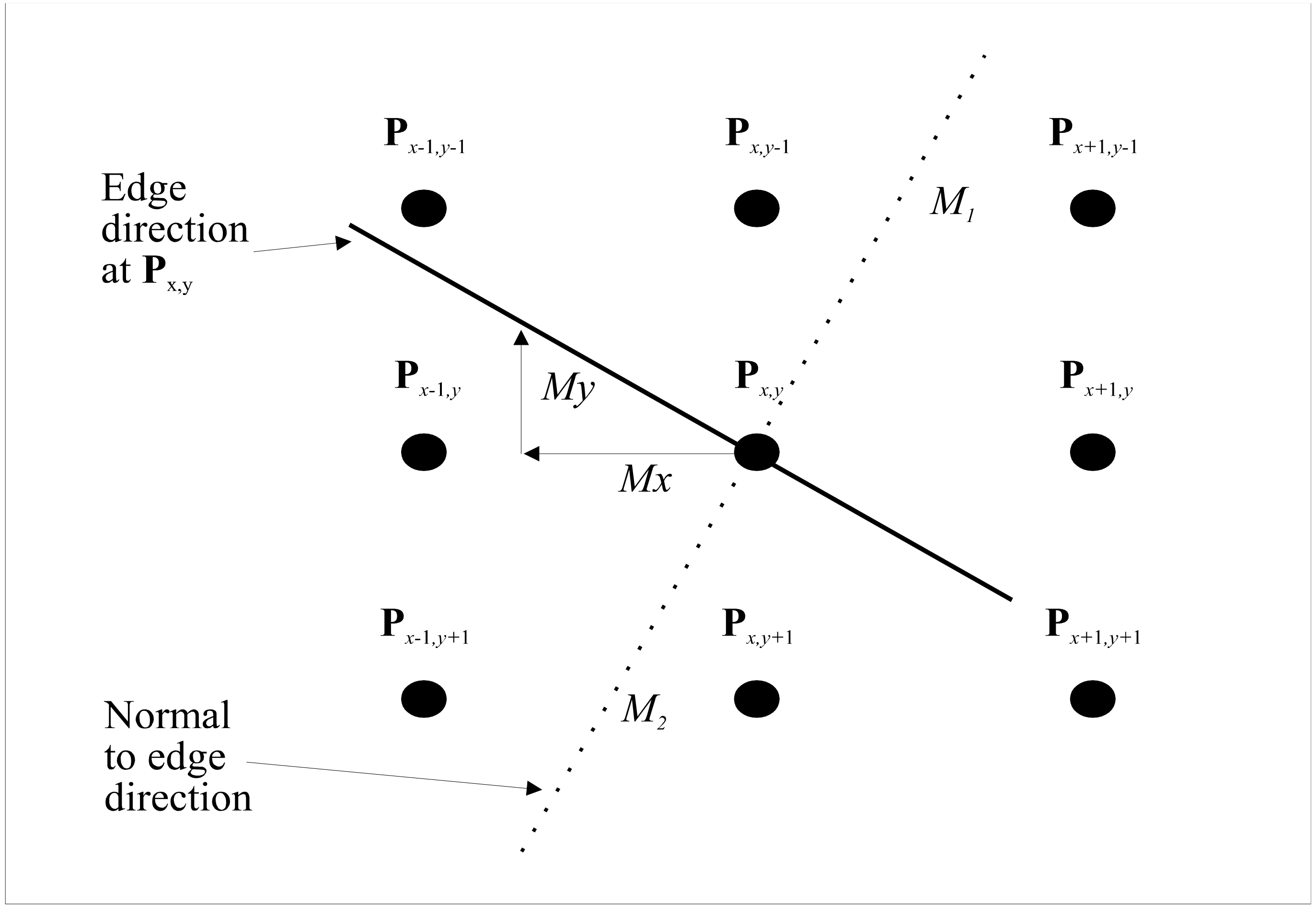
combine?

3. use non-maximal suppression; and
4. threshold with hysteresis to connect edge points.

Interpolation in non-maximum suppression

Need to use points which are not on the image grid

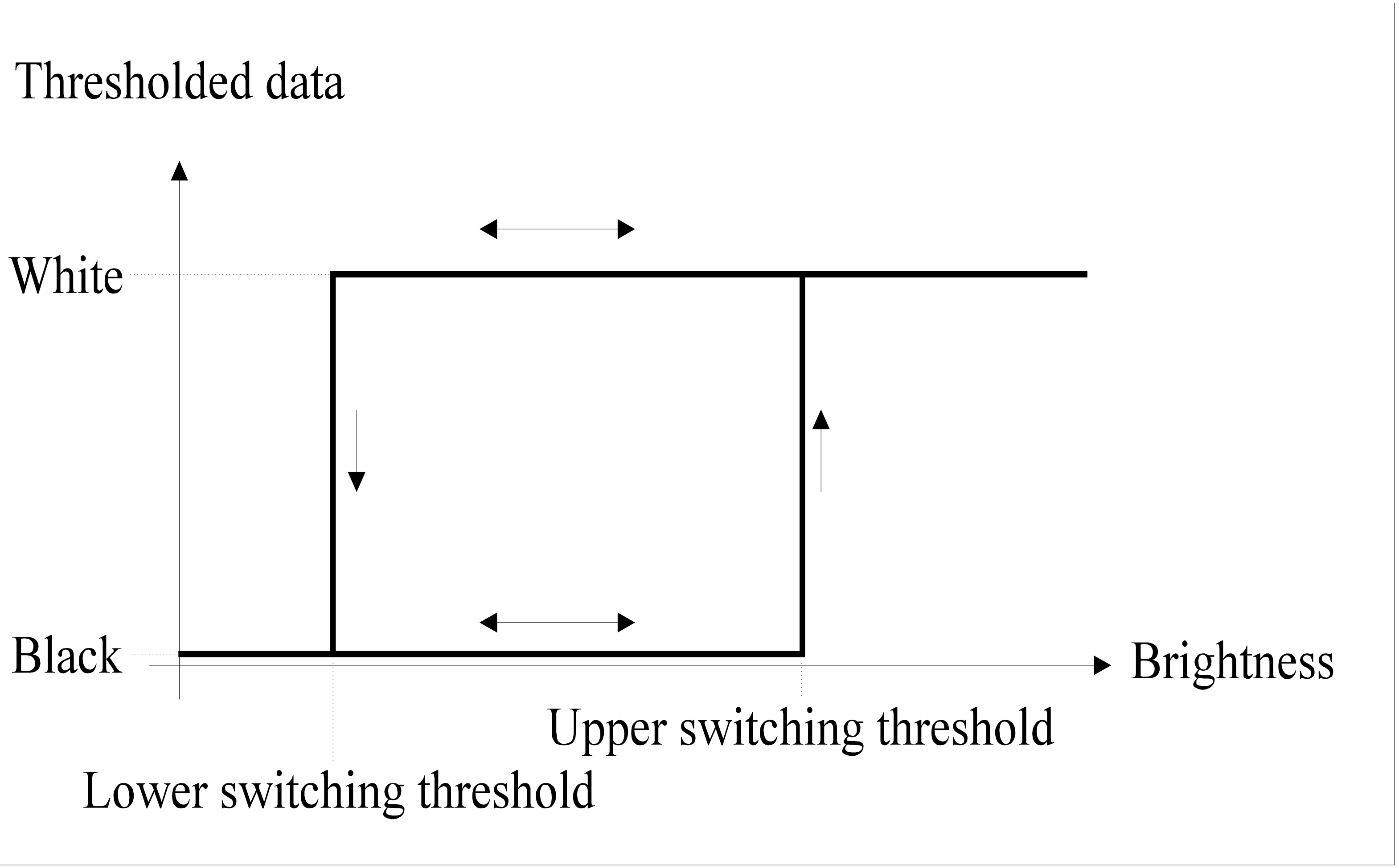
Uses linear interpolation



Hysteresis thresholding transfer function

Lower threshold = average noise

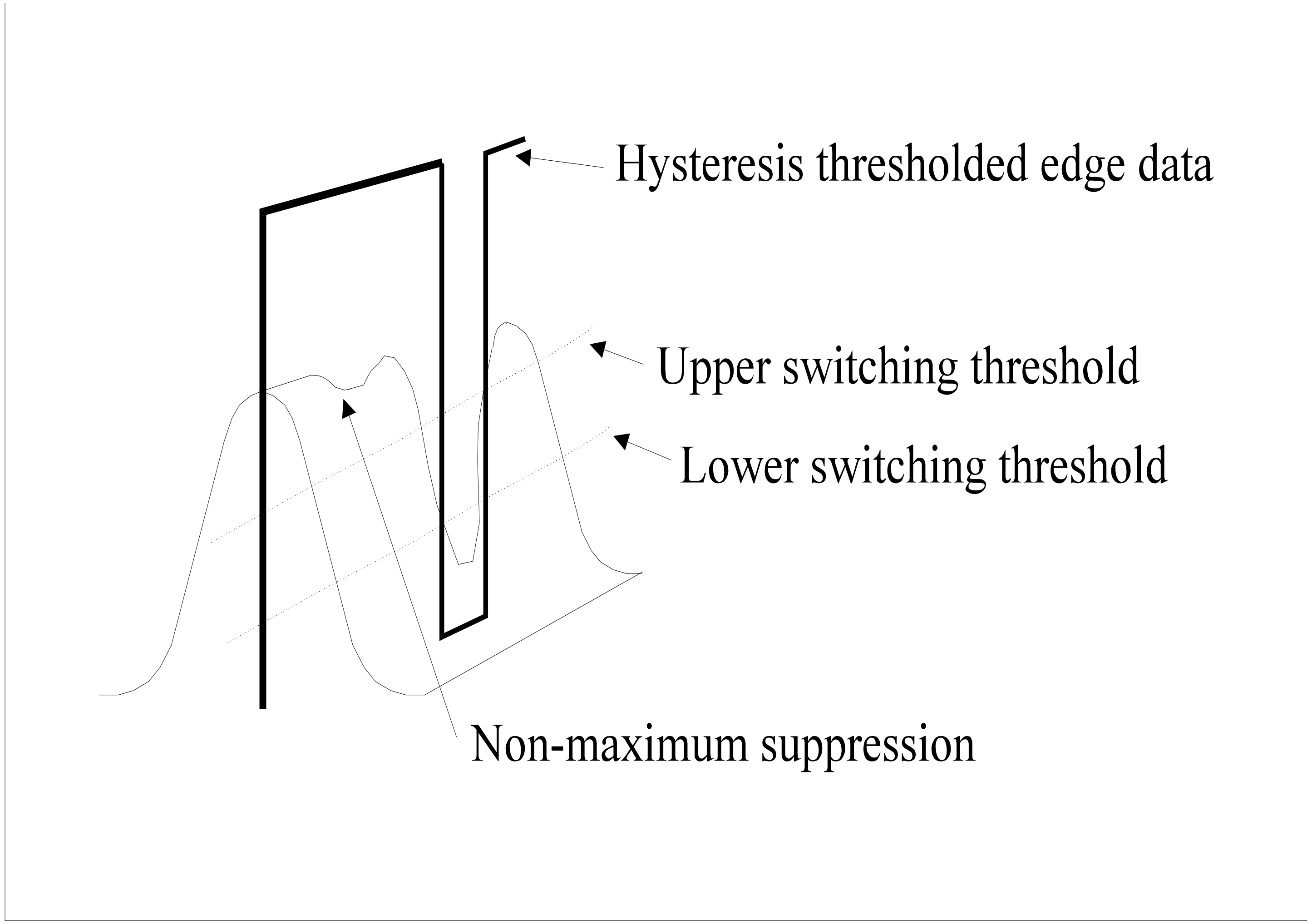
Upper threshold = average feature boundary



Action of non-maximum suppression and hysteresis thresholding

Walk along top of ridge

Gives thin edges in the right place



Comparing hysteresis thresholding with uniform thresholding

(a) hysteresis thresholding, upper level = 40, lower level = 10

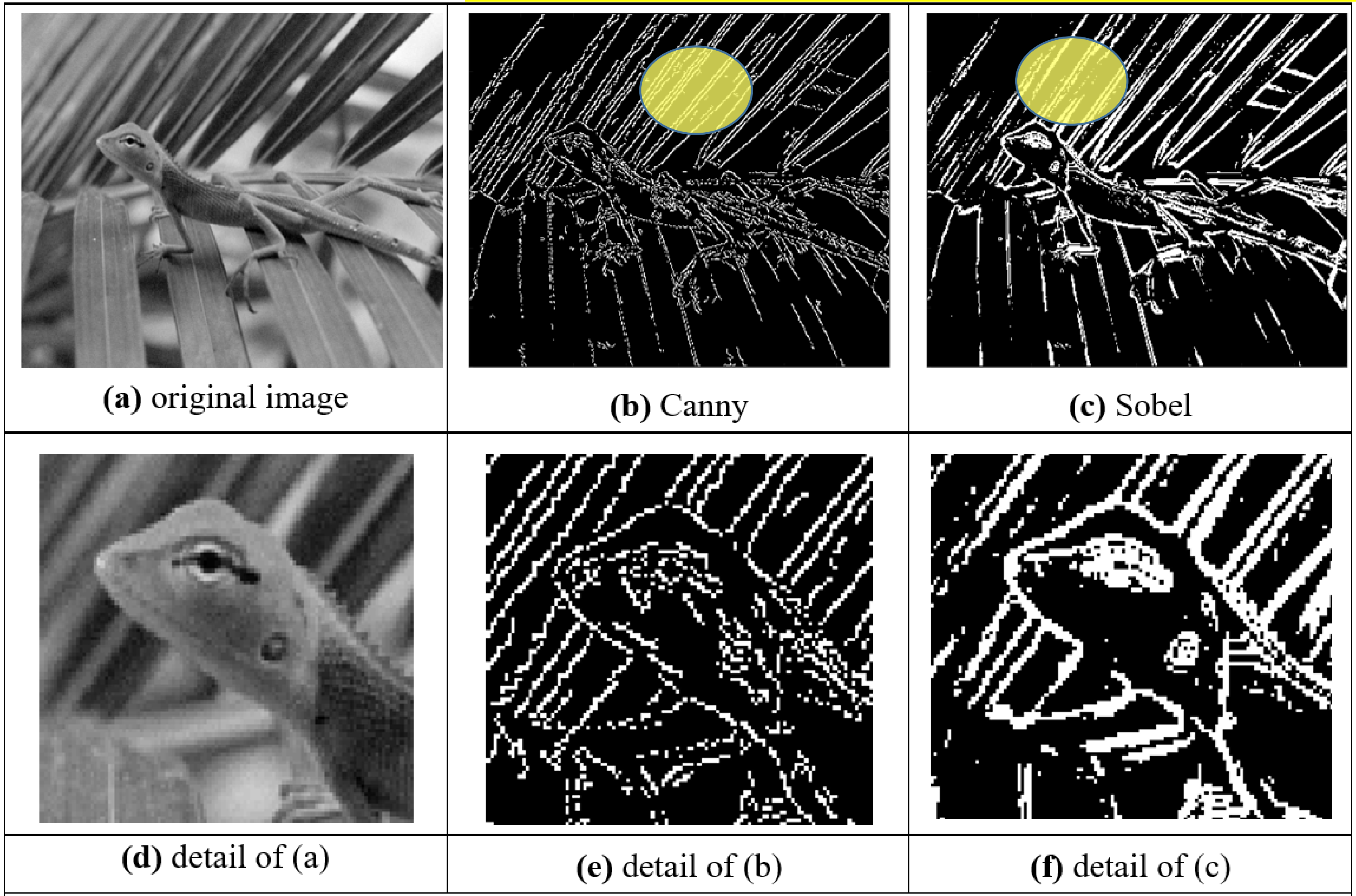
(b) uniform thresholding, level = 40

(c) uniform thresholding, level = 10

Hysteresis thresholding gives all points > upper threshold
plus any connected points > lower threshold

Comparing Canny with Sobel

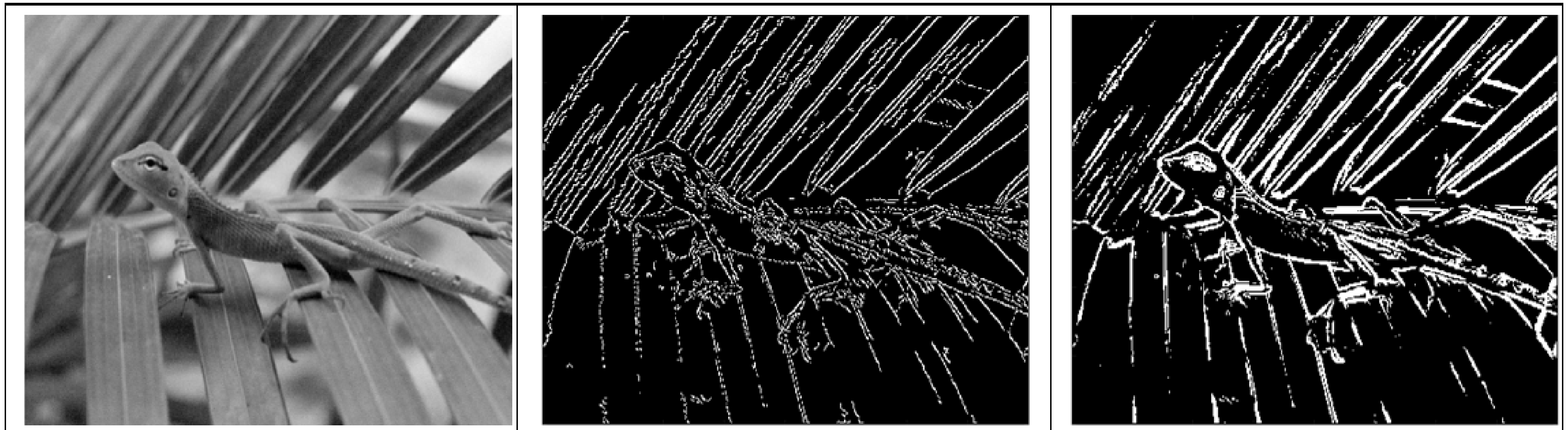
The lines are thinner here, making Sobel look better!



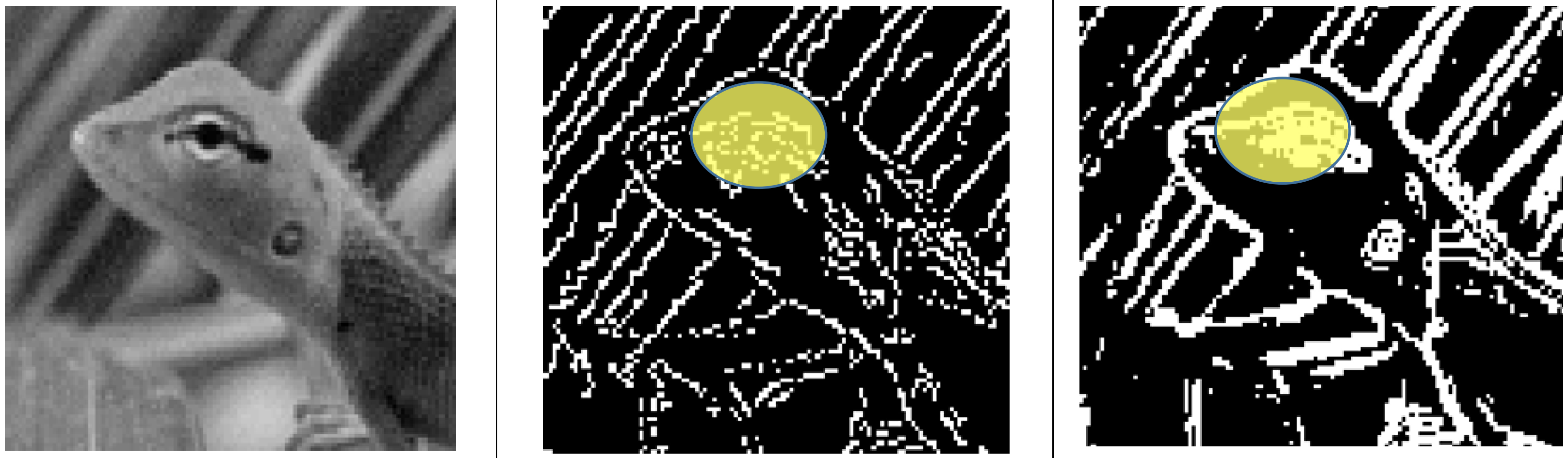
(a) original image
(b) Canny
(c) Sobel
(d) detail of (a)
(e) detail of (b)
(f) detail of (c)

Comparing Canny with Sobel

The lines are indeed thinner



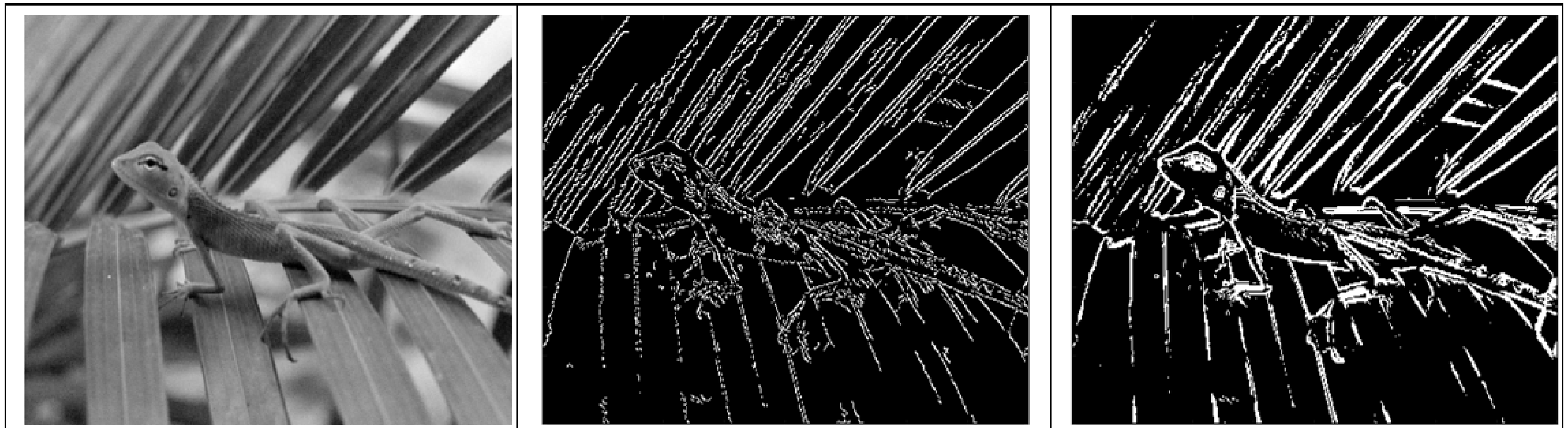
(a) original image | (b) Canny | (c) Sobel



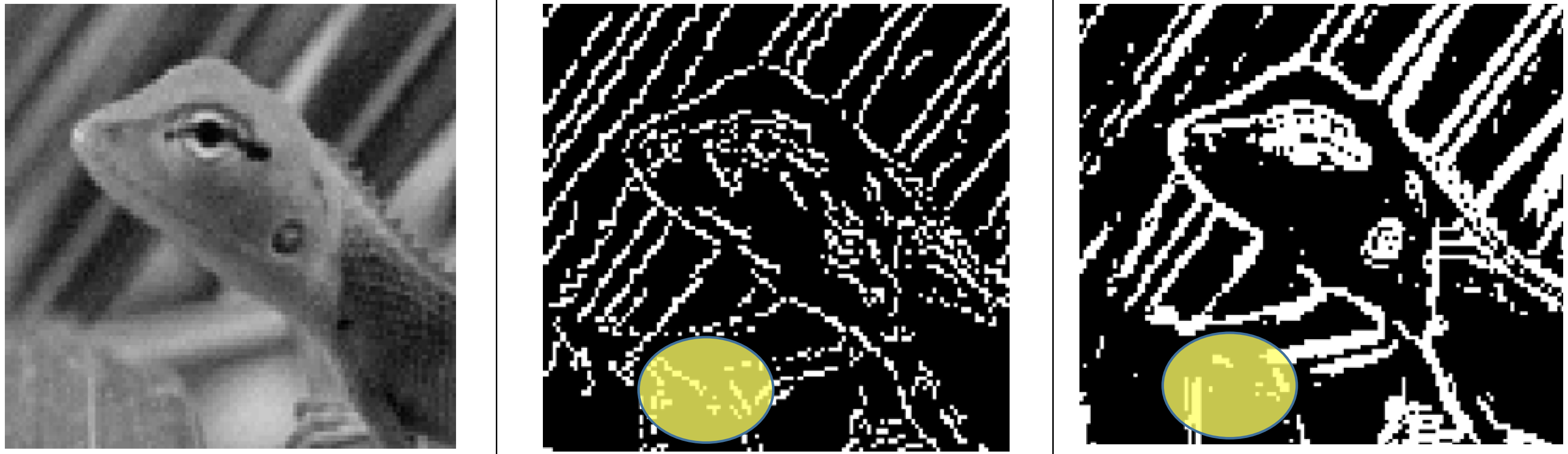
(d) detail of (a) | (e) detail of (b) | (f) detail of (c)

Comparing Canny with Sobel

The noise is less



(a) original image | **(b)** Canny | **(c)** Sobel

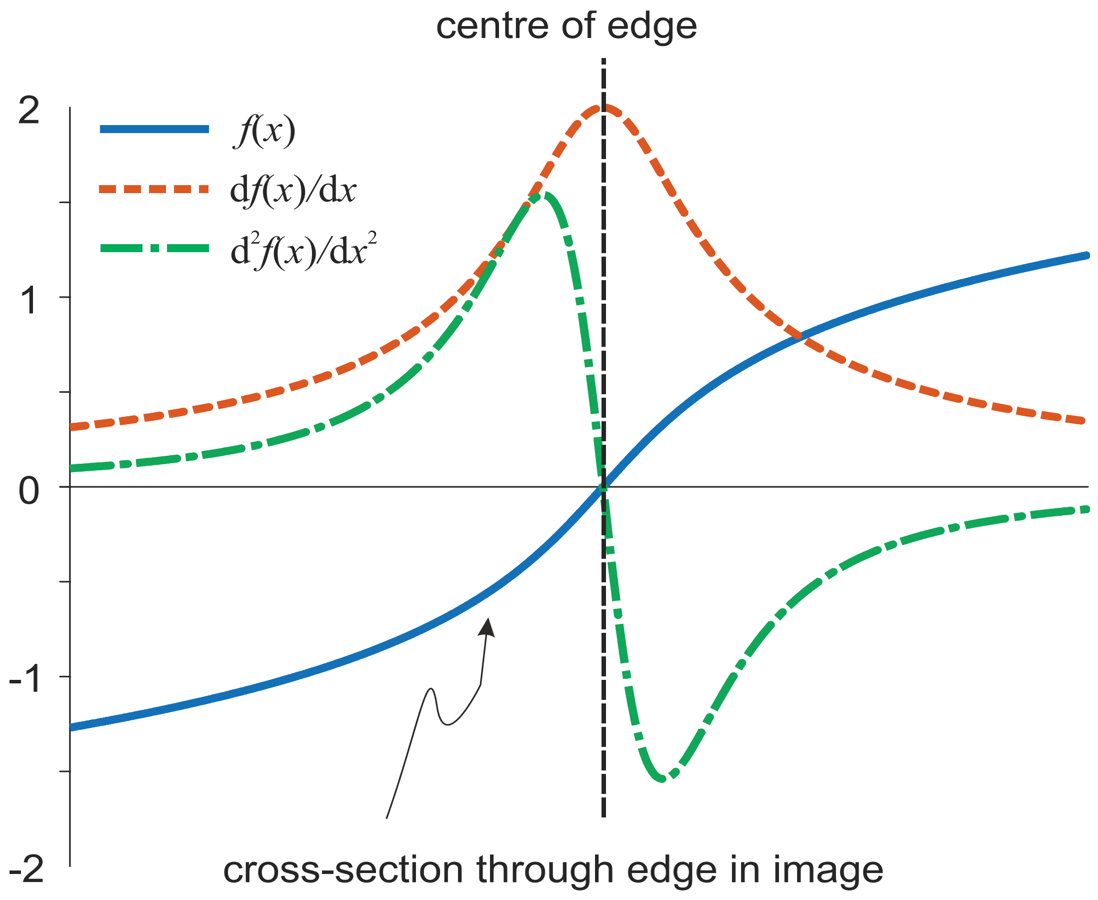


(d) detail of (a) | **(e)** detail of (b) | **(f)** detail of (c)

First and second order edge detection

First order = single differentiation
with thresholding

Second order = twice differentiation
with zero-crossing detection



Edge detection via the Laplacian operator

0	-1	0
-1	4	-1
0	-1	0

1	2	3	4	1	1	2	1
2	2	3	0	1	2	2	1
3	0	38	39	37	36	3	0
4	1	40	44	41	42	2	1
1	2	43	44	40	39	3	1
2	0	39	41	42	40	2	0
0	2	0	2	2	3	1	1
0	2	1	3	1	0	4	2

(a) image data

0	0	0	0	0	0	0	0
0	1	−31	−47	−36	−32	0	0
0	−44	70	37	31	60	−28	0
0	−42	34	12	1	50	−41	0
0	−37	47	8	−6	31	−32	0
0	−45	72	37	45	74	−36	0
0	6	−44	−38	−40	−31	−6	0
0	0	0	0	0	0	0	0

(b) result of the Laplacian operator

Simple, but unused!

Edge detection is about differentiation

Take a Gaussian function

$$g(x, y, \sigma) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

Differentiate once

$$\frac{\partial g(x,y,\sigma)}{\partial x} = -\frac{x}{\sigma^2}\, e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

And again

$$\frac{\partial^2 g(x,y,\sigma)}{\partial x^2} = \left(\frac{x^2}{\sigma^2} - 1\right)\frac{e^{\frac{-(x^2+y^2)}{2\sigma^2}}}{\sigma^2}$$

Mathbelts on…

Second order in x and y is

$$\nabla^2 g(x,y,\sigma) = \frac{\partial^2 g(x,y,\sigma)}{\partial x^2}U_x + \frac{\partial^2 g(x,y,\sigma)}{\partial y^2}U_y$$

By substitution

$$= \left(\frac{x^2}{\sigma^2}-1\right)\frac{e^{\frac{-(x^2+y^2)}{2\sigma^2}}}{\sigma^2} + \left(\frac{y^2}{\sigma^2}-1\right)\frac{e^{\frac{-(x^2+y^2)}{2\sigma^2}}}{\sigma^2}$$

So we get

$$= \frac{1}{\sigma^2}\left(\frac{x^2+y^2}{\sigma^2}-2\right)e^{\frac{-(x^2+y^2)}{\sigma^2}}$$

Why, oh why, have we done this ???

Second order = Laplacian of Gaussian = Marr Hildreth

Top 3 hits Google: “Laplacian of Gaussian”

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$LoG \triangleq \triangle G_\sigma(x,y) = \frac{\partial^2}{\partial x^2} G_\sigma(x,y) + \frac{\partial^2}{\partial y^2} G_\sigma(x,y) = \frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2}$$

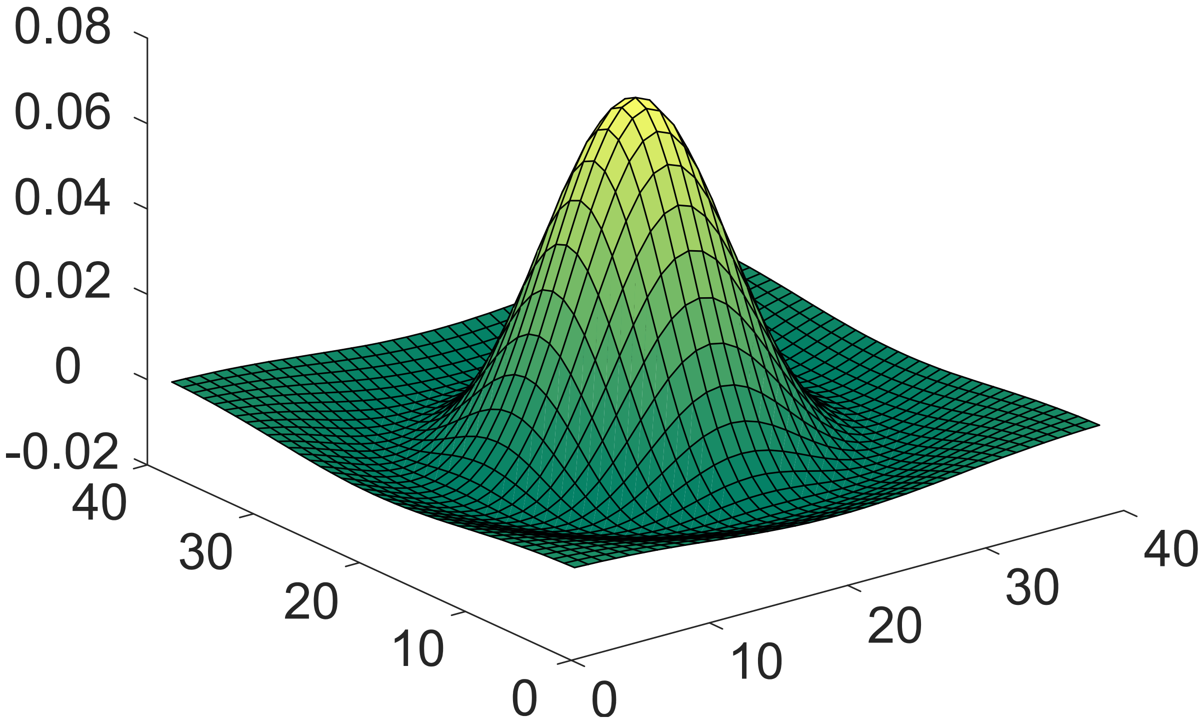
$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Two wrong, one right. Just one…..why?

(and two of them don’t even work!!)

http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm; http://fourier.eng.hmc.edu/e161/lectures/gradient/node8.html ; http://academic.mu.edu/phys/matthysd/web226/Lab02.htm

Shape of Laplacian of Gaussian operator



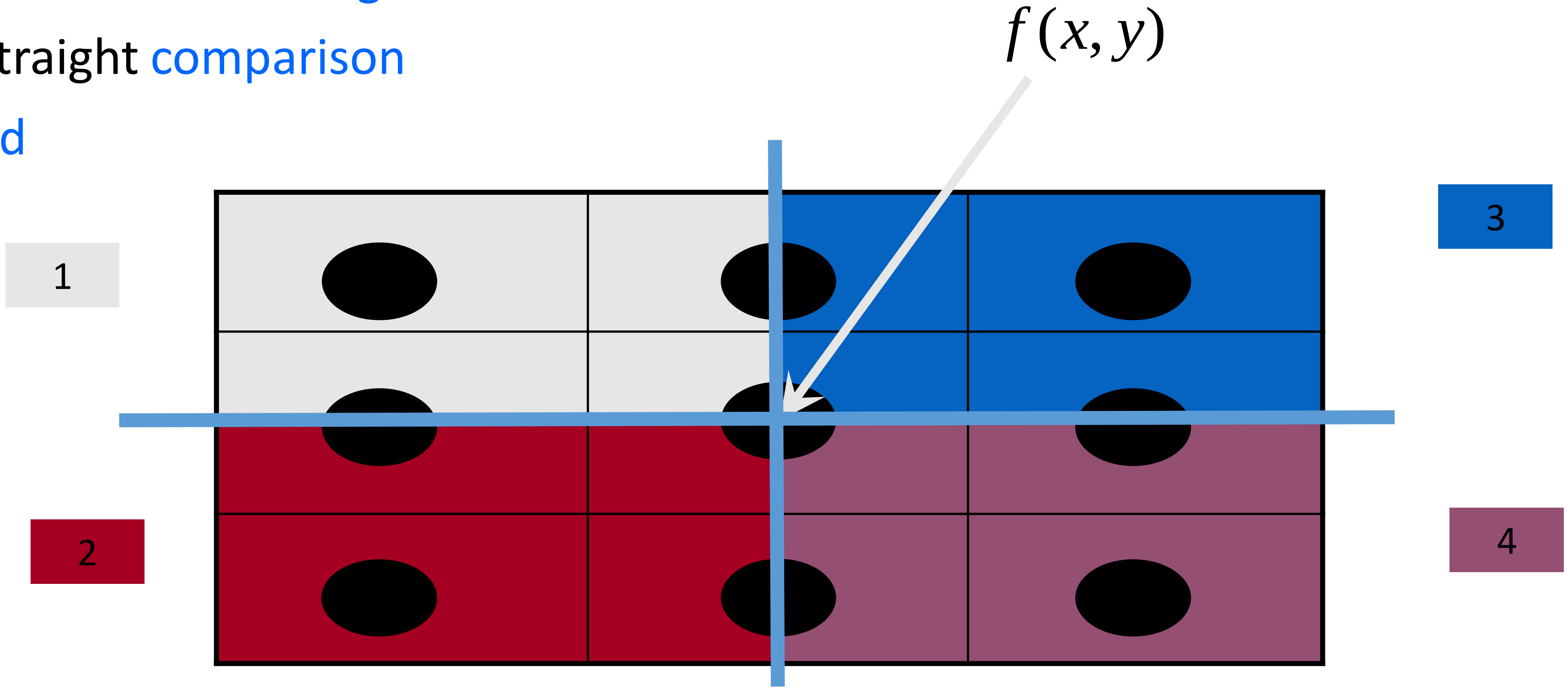
It's called the 'Mexican hat operator'

Zero crossing detection

Need to find zero-crossings in 2D

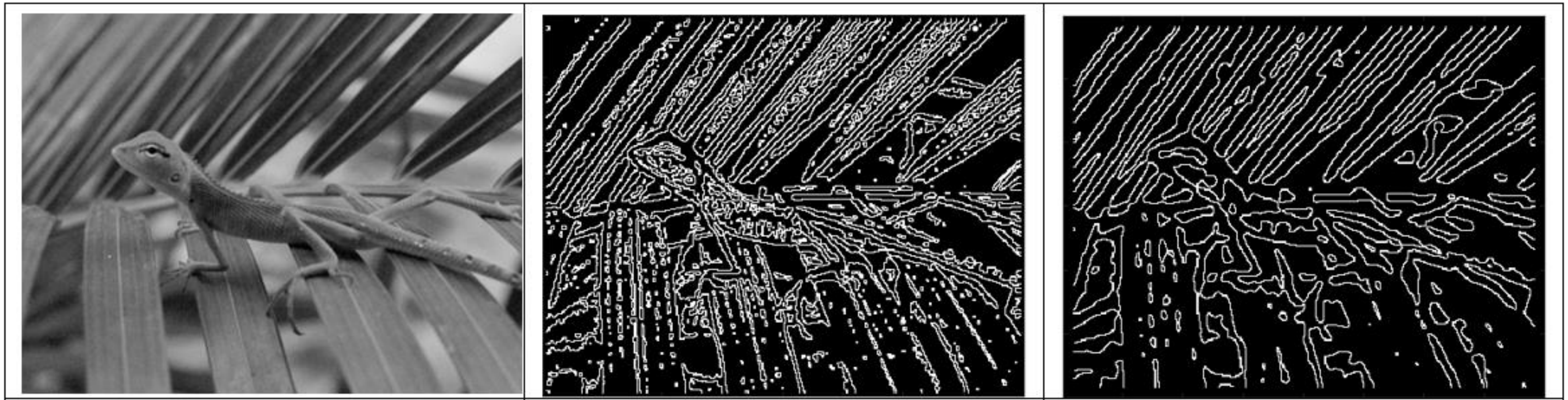
Basic – straight comparison

Advanced



$$IF\left(\max(1,2,3,4)>0 \wedge \min(1,2,3,4)<0\right) \quad THEN \quad f(x,y)=\text{edge}$$

Marr-Hildreth edge detection

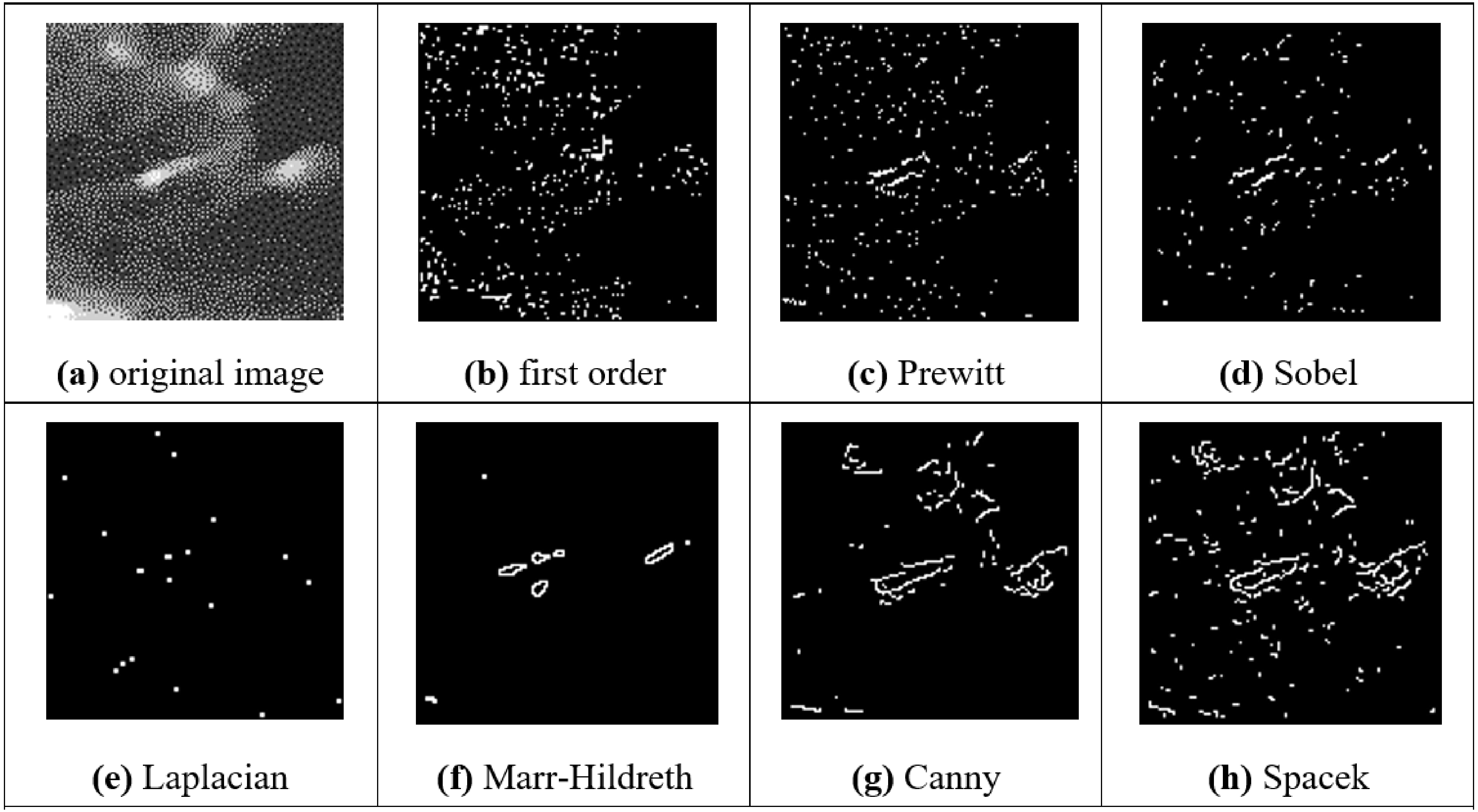


(a) image	(b) LoG ($9\times9; \sigma=1.4$)	(c) LoG ($19\times19; \sigma=3$)

Small template, small σ for local features

Large template, large σ for global features

Comparison of edge detection operators

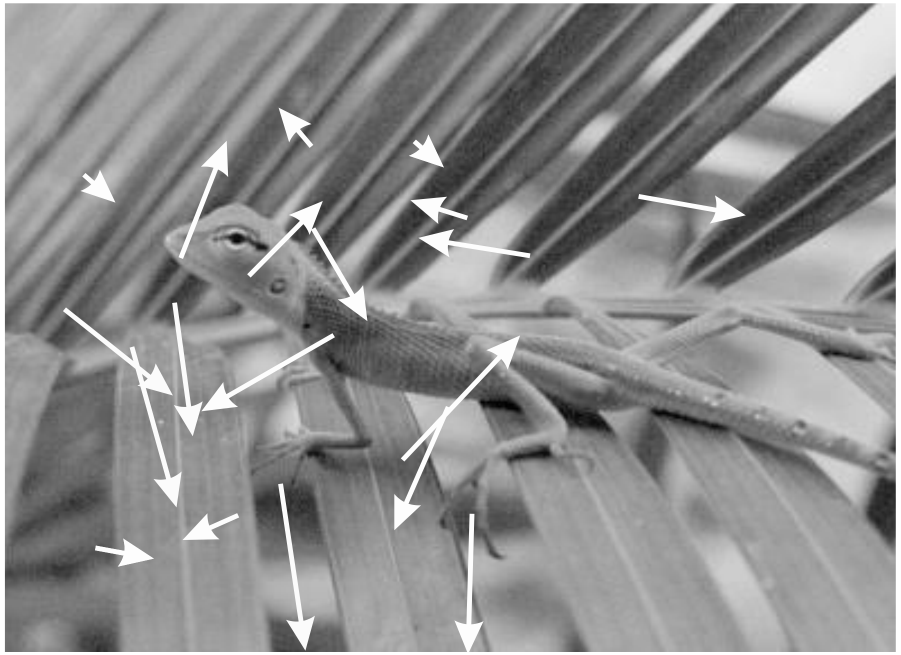


(a) original image | (b) first order | (c) Prewitt | (d) Sobel

(e) Laplacian | (f) Marr-Hildreth | (g) Canny | (h) Spacek

Newer stuff – interest detections

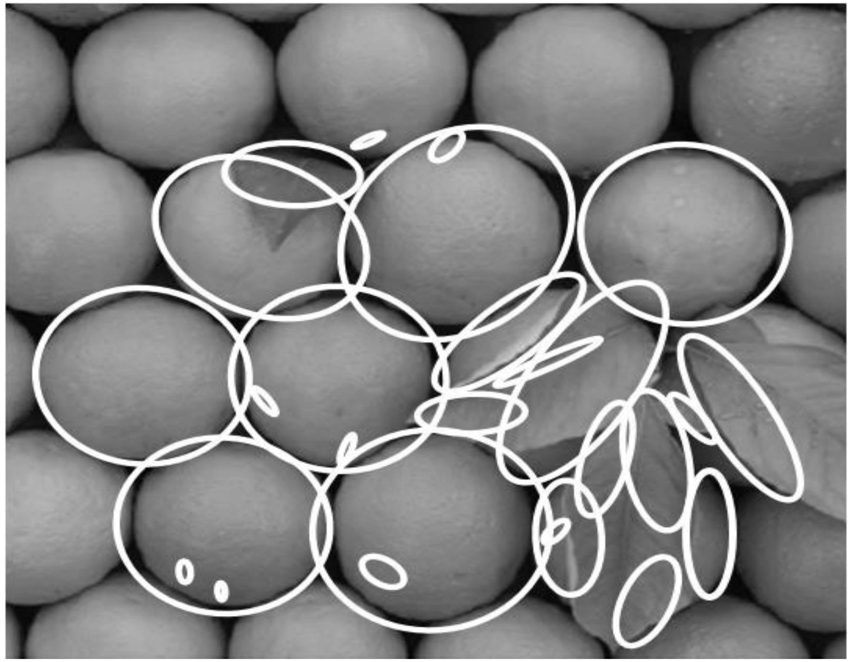
feature points



SIFT (mega famous)

(wait for Jon)

regions

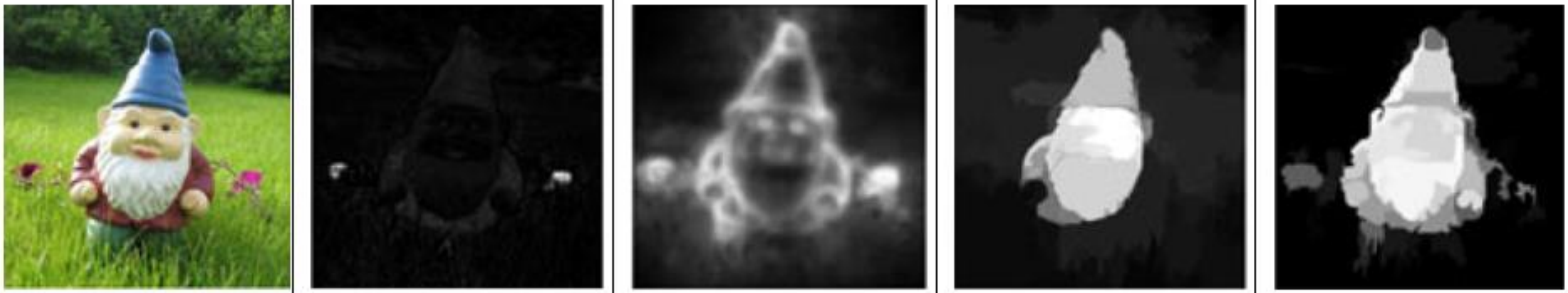


brightness clustering

(excellent, but confess its ours)

Lomeli-R. and Nixon and Carter, *Mach Vis Apps* 2016

Newer stuff – saliency



(a) image (b) [Achanta08] (c) context aware (d) [Jiang11] (e) region contrast

Comparison of State of Art Saliency Methods [Cheng15]

Takeaway time

1 – Canny provides thin edges in the right place

2 – second order (Marr-Hildreth) requires zero-crossing detection

3 – the results by Marr-Hildreth and Canny are well worth the extra computation

Now we need to collect the edges to find shape. Coming next…