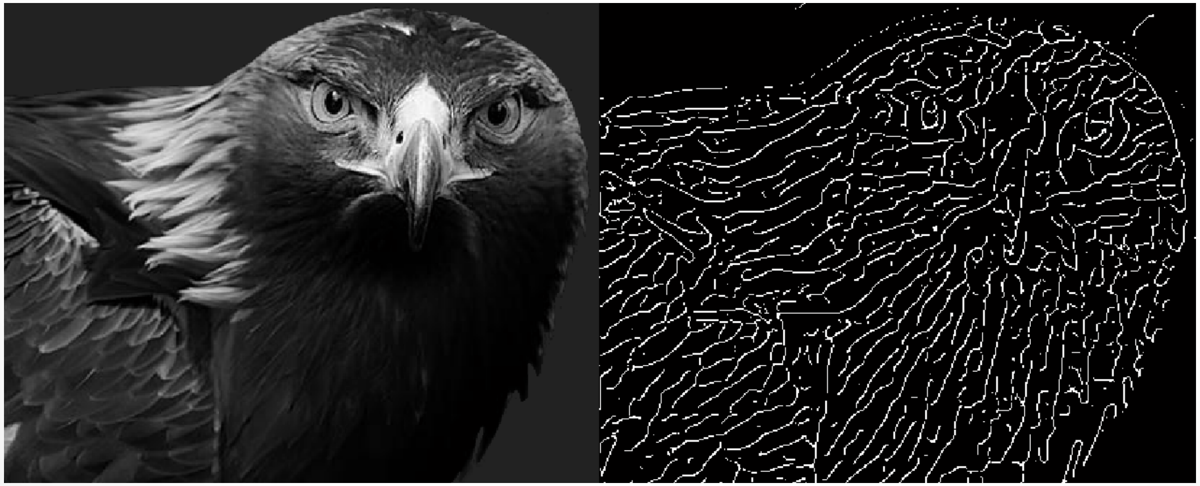
Edge Detection

COS791: Chapter 4

Dr. Mardé Helbig

First-order Edge Detectors

- Use grey-scale images
- Edge detection highlights image contrast
- Difference in intensity can highlight a boundary



First-order Edge Detectors
Basic Operators

- Can reveal a change in intensity by differencing adjacent points
- **Differencing horizontal** adjacent points reveal changes in vertical intensity => **horizontal edge detector**
- Applied to image *P*:

$$\mathbf{Ex}_{x,y} = \left|\mathbf{P}_{x,y} - \mathbf{P}_{x+1,y}\right| \qquad \forall x \in 1, N-1;\ y \in 1, N$$

Detects vertical edges

(a) Original image (b) Vertical edges

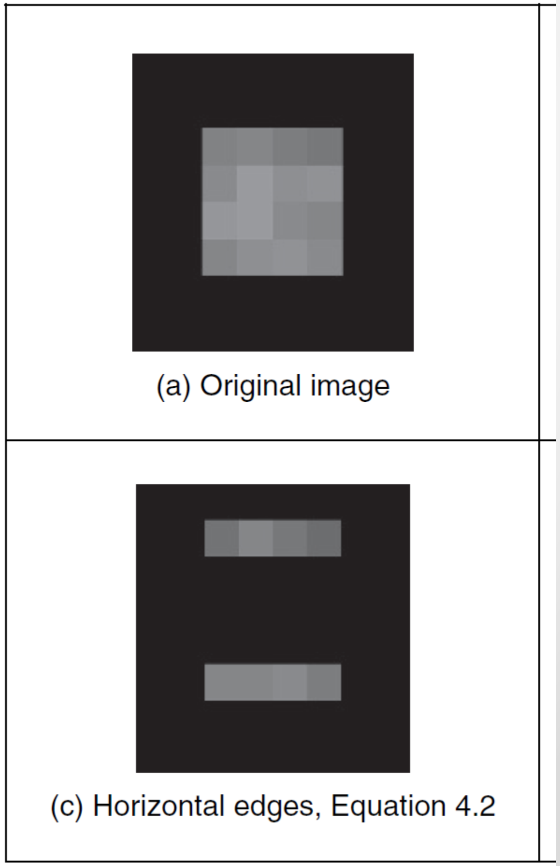
First-order Edge Detectors

Basic Operators

- Can reveal a change in intensity by differencing adjacent points
- **Differencing vertical** adjacent points reveal changes in horizontal intensity =>

  vertical edge detector
- Applied to image *P*:

$$\mathbf{Ey}_{x,y} = \left|\mathbf{P}_{x,y} - \mathbf{P}_{x,y+1}\right| \qquad \forall x \in 1, N;\ y \in 1, N-1$$


(a) Original image

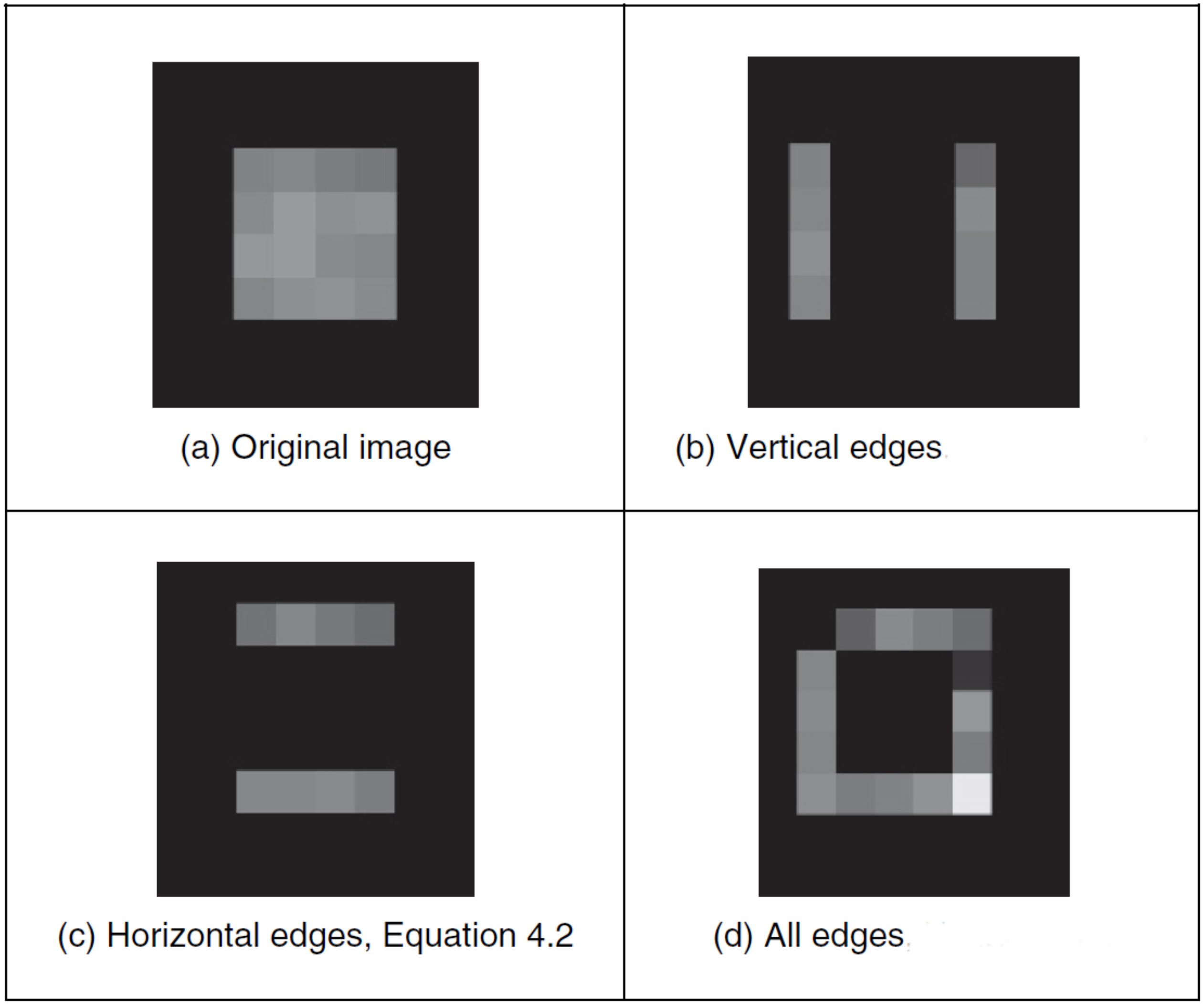
(c) Horizontal edges, Equation 4.2

Detects vertical edges

-

First-order Edge Detectors
Basic Operators

- Applying both to image P:



Horizontal Edge Detector

Vertical Edge Detector

Why is right lower corner bright and top left corner black?

First-order Edge Detectors

Basic Operators

- Combining the two operators produces an operator that can detect vertical **and** horizontal edges:

$$\mathbf{E}_{x,y} = \left|\mathbf{P}_{x,y} - \mathbf{P}_{x+1,y} + \mathbf{P}_{x,y} - \mathbf{P}_{x,y+1}\right| \qquad \forall x, y \in 1, N-1$$

which gives:

$$\mathbf{E}_{x,y} = \left|2 \times \mathbf{P}_{x,y} - \mathbf{P}_{x+1,y} - \mathbf{P}_{x,y+1}\right| \qquad \forall x, y \in 1, N-1$$

- Template and code for this first-order operator:

2	–1
–1	0

```
edge(pic):= | newpic←zero(pic)
            | for x∈0..cols(pic)-2
            |   for y∈0..rows(pic)-2
            |     newpic_{y,x}← |2·pic_{y,x}-pic_{y,x+1}-pic_{y+1,x}|
            | newpic
```

First-order Edge Detectors

Robert Cross Operator

- One of the earliest edge detector operators
- Implements basic first-order edge detection
- Uses **two** templates that differentiate pixel values **diagonally** and not along the axes
- Templates:

+1	0
0	–1

(a) M^{-}

0	+1
–1	0

(b) M^{+}

How do these templates differentiate diagonally?

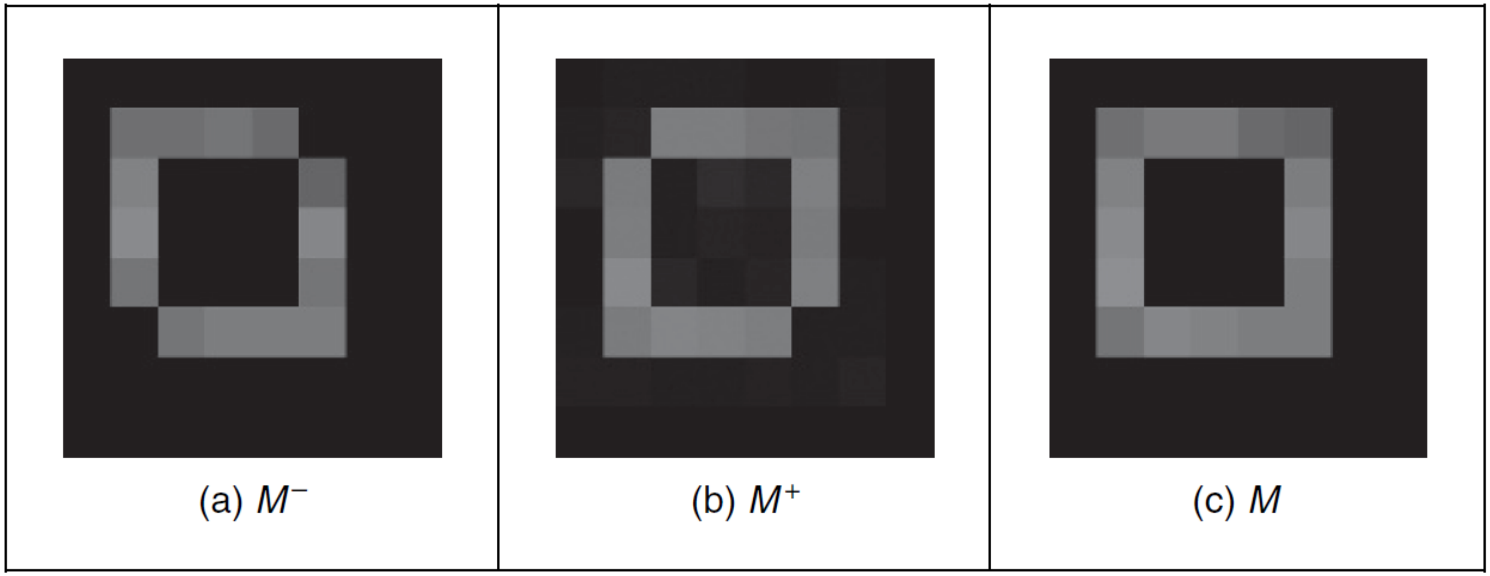
- Maximum value obtained by applying the templates is stored at the point

$$\mathbf{E}_{x,y} = \max\left\{\left|M^{+} * \mathbf{P}_{x,y}\right|, \left|M^{-} * \mathbf{P}_{x,y}\right|\right\} \qquad \forall x, y \in 1, N-1$$

First-order Edge Detectors

Robert Cross Operator

- Applying Robert Cross Operator to square image:

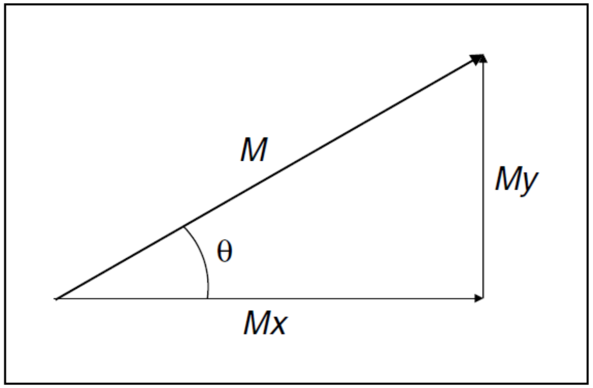


What different results are observed here from the previous example?

First-order Edge Detectors

Prewett Operator

- Edge detection operators use differentiation
- Will be affected by noise
- Can incorporate **averaging** within the process
- Two values, i.e. rate of change of brightness along each axis => vector
- Therefore:



$$M(x, y) = \sqrt{Mx(x, y)^2 + My(x, y)^2}$$ Magnitude

$$\theta(x, y) = \tan^{-1}\left(\frac{My(x, y)}{Mx(x, y)}\right)$$ Angle

First-order Edge Detectors
Prewett Operator

- Template:

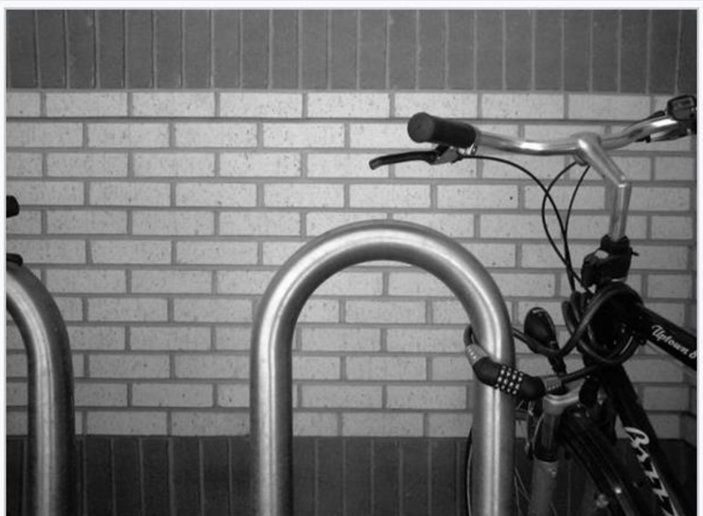
1	0	−1
1	0	−1
1	0	−1

(a) *Mx*

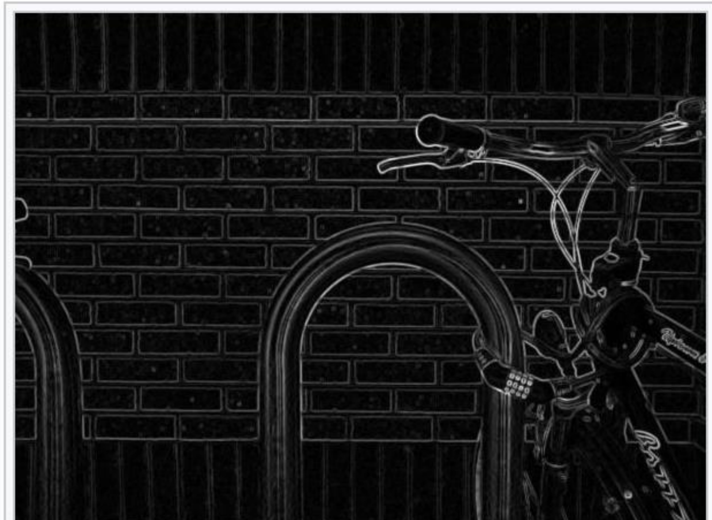
1	1	1
0	0	0
−1	−1	−1

(b) *My*

Do these values make sense?



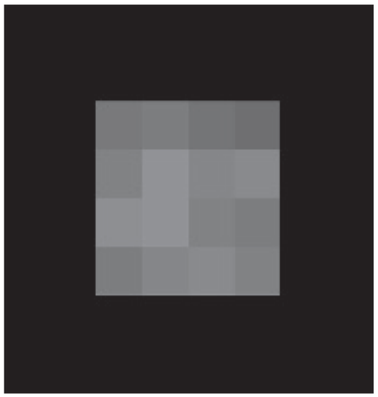
Grayscale image of a brick wall & a bike rack



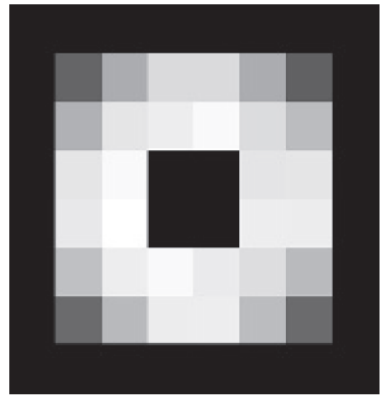
Gradient with Prewitt operator of grayscale image of a brick wall & a bike rack

First-order Edge Detectors

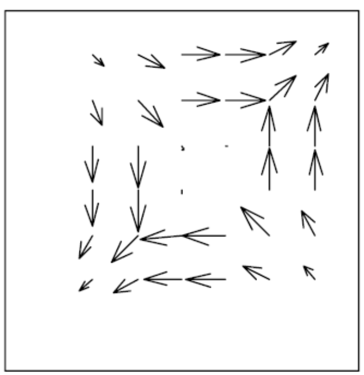
Prewett Operator



(a) Original image



(b) Edge magnitude



$prewitt\_vec_{0,1}, prewitt\_vec_{0,0}$

(c) Vector format

$$\text{dir} = \begin{bmatrix} 313 & 331 & 3 & 3 & 24 & 47 \\ 298 & 315 & 1 & 2 & 42 & 63 \\ 273 & 276 & 13 & 43 & 88 & 88 \\ 269 & 268 & 199 & 117 & 91 & 92 \\ 242 & 225 & 181 & 178 & 133 & 116 \\ 225 & 210 & 183 & 179 & 155 & 132 \end{bmatrix}$$

(d) Edge direction

Less defined at corners. Why?

Measured in degrees:

- 0 and 360: horizontal to the right
- 90: vertical upward

First-order Edge Detectors

Sobol Operator

- If you **double** the weight of the **center** pixels of the Prewitt templates, it produces the **Sobol** operator

1	0	–1
1	0	–1
1	0	–1

(a) *Mx*

1	1	1
0	0	0
–1	–1	–1

(b) *My*

Prewitt templates

1	0	–1
2	0	–2
1	0	–1

(a) *Mx*

1	2	1
0	0	0
–1	–2	–1

(b) *My*

Sobol templates

First-order Edge Detectors

Sobol Operator

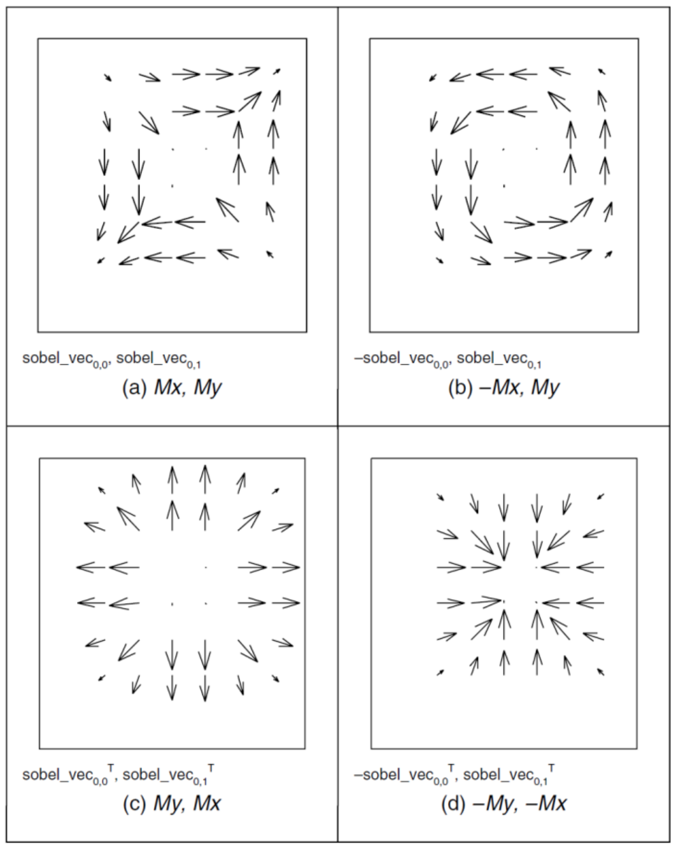
- But how can we apply templates of 5x5 or 7x7?
- Consider optimal forms of averaging and differencing
- Gaussian averaging produces optimal averaging => limit of the binomial expansion approximates the normal distribution
- Pascal's triangle produces a set of coefficients for a smoothing operator, where the limit approaches the coefficients of the Gaussian smoothing operator

Refer to pages 148 and 149 in Text book

First-order Edge Detectors

Sobol Operator

- Sobol edge-direction data can be arranged to point in different ways
- 4 Different ways:



Useful to speed up algorithm to search for specific shapes

First-order Edge Detectors

Canny Operator

- Currently most popular
- Three main objectives:
  1. **Optimal detection** with no spurious responses
  2. **Good localization** with minimal distance between detected and true edge position
  3. **Single response** to eliminate multiple responses on a single edge

First-order Edge Detectors

Canny Operator

- Three main objectives:
  1. **Optimal detection** with no spurious responses

- Can achieve the above by reducing noise
- Through **optimal smoothing** – Gaussian filtering

First-order Edge Detectors
Canny Operator

- Three main objectives:
  2. **Good localization** with minimal distance between detected and true edge position

- Aims for accuracy
- Can achieve this by process of **non-maximum suppression**
- Retains only those points at the top of a ridge of edge data, while suppressing others
- Results in thinning – output is thin lines of edge points, but accurate (in the right place)

Equivalent to peak detection

First-order Edge Detectors
Canny Operator

- Three main objectives:
  3. **Single response** to eliminate multiple responses on a single edge

- Location of a single edge point in response to a change in brightness
- More than one edge can be denoted as being present

First-order Edge Detectors

Canny Operator

- Derive an equation to apply non-maximum suppression for the operator
- But it is difficult to implement since you must estimate the normal direction
- Therefore, typically use an approximation

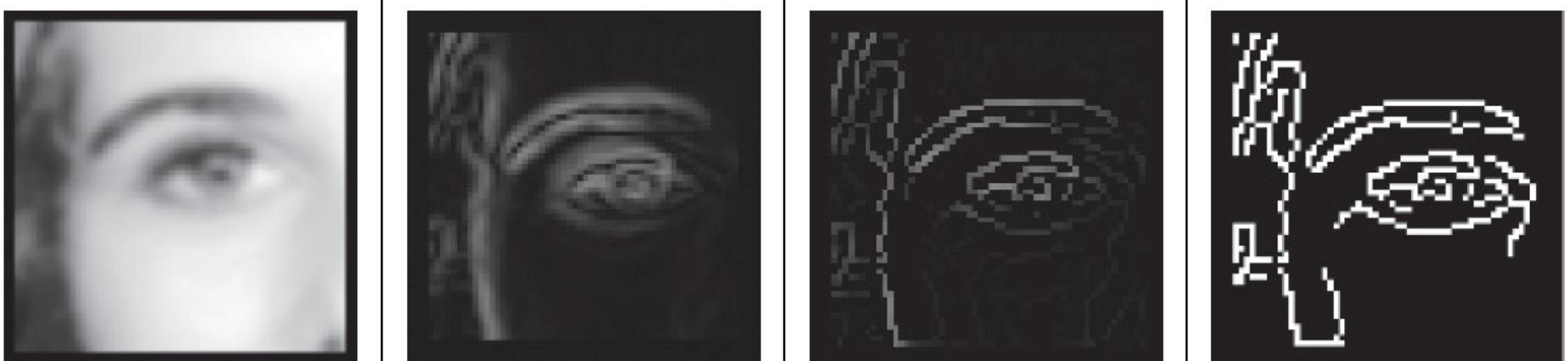
First-order Edge Detectors

Canny Operator: Approximation

Approximation has 4 steps/stages:

1. **Use Gaussian smoothing**
2. **Use Sobel operator**
3. Use non-maximal suppression
4. Threshold with hysteresis to connect edge points

Can combine these 2 steps



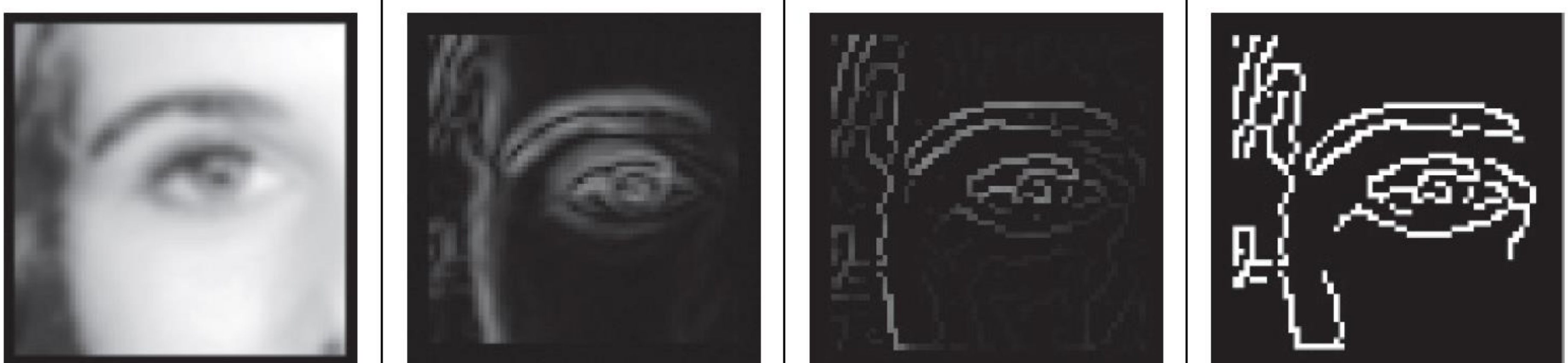
(a) Gaussian smoothing | (b) Sobel edge detection | (c) Non-maximum suppression | (d) Hysteresis thresholding

First-order Edge Detectors

Canny Operator: Approximation

Approximation has 4 steps/stages:

1. Use Gaussian smoothing
2. Use Sobel operator
3. **Use nonmaximal suppression**
4. Threshold with hysteresis to connect edge points



(a) Gaussian smoothing | (b) Sobel edge detection | (c) Non-maximum suppression | (d) Hysteresis thresholding

First-order Edge Detectors

Canny Operator: Nonmaximal Suppression

- **Nonmaximal suppression** locates the **highest** points in the edge magnitude data
- Use edge-direction information to check whether points are at the peak of a ridge
- Given a 3x3 region, a point is at a maximum if:
  - Gradient on either side of point < gradient at point

=> Need values of gradient along line normal to edge at point

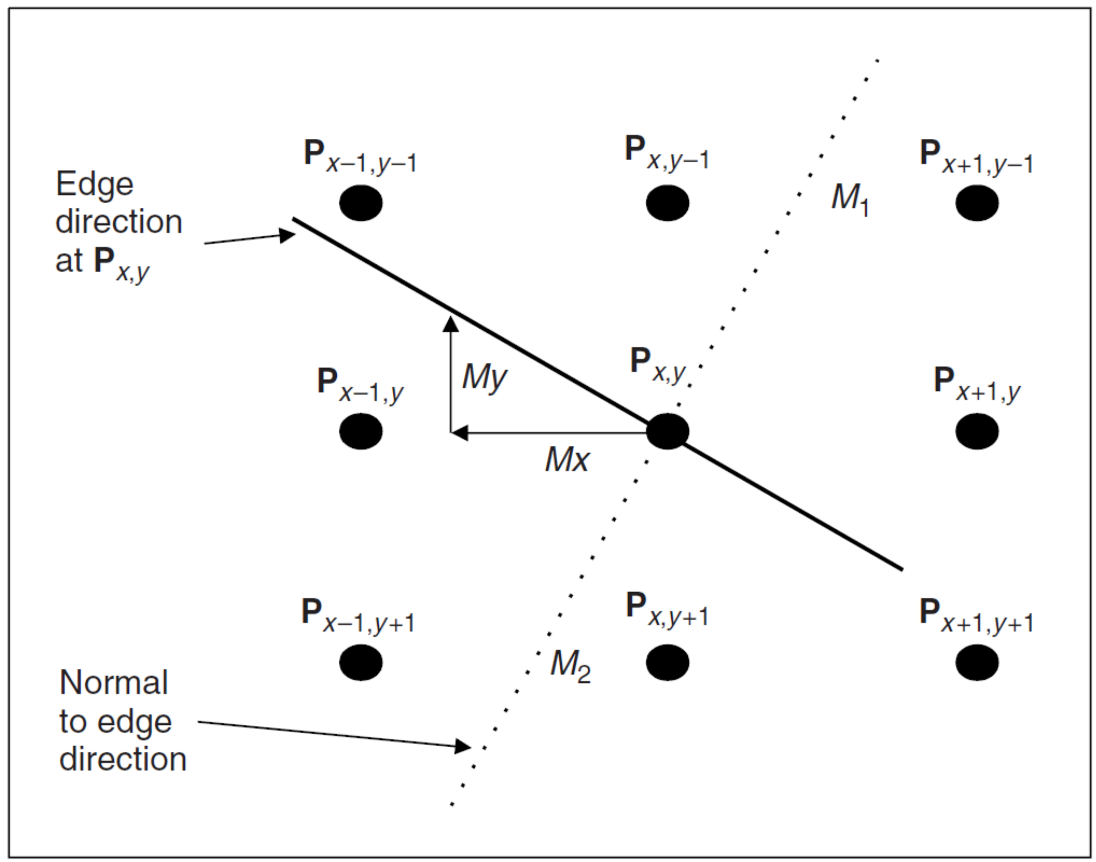
First-order Edge Detectors

Canny Operator: Nonmaximal Suppression

- Given a 3x3 region, a point is at a maximum if:
  - Gradient on either side of point < gradient at point

  => Need values of gradient along line normal to edge at point

$P_{x,y}$ is a maximum if:
It's gradient M(x,y) >
gradients at points M_1
and M_2 respectively



-

First-order Edge Detectors

Canny Operator: Nonmaximal Suppression

$P_{x,y}$ is a maximum if:
It's gradient M(x,y) > gradients at points M_1 and M_2 respectively

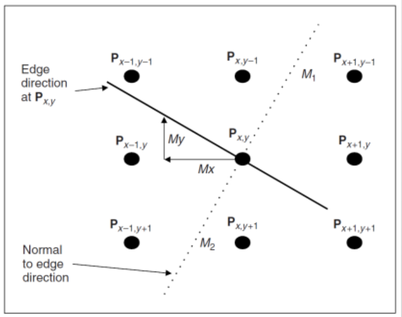
- Have a discrete neighbourhood => need interpolation
- First-order interpolation:

$$M_1 = \frac{My}{Mx} M(x+1,\ y-1) + \frac{Mx - My}{Mx} M(x,\ y-1)$$

and

$$M_2 = \frac{My}{Mx} M(x-1,\ y+1) + \frac{Mx - My}{Mx} M(x,\ y+1)$$

$P_{x,y}$ is a maximum if M(x,y) > gradients at points M_1 and M_2, else set to zero



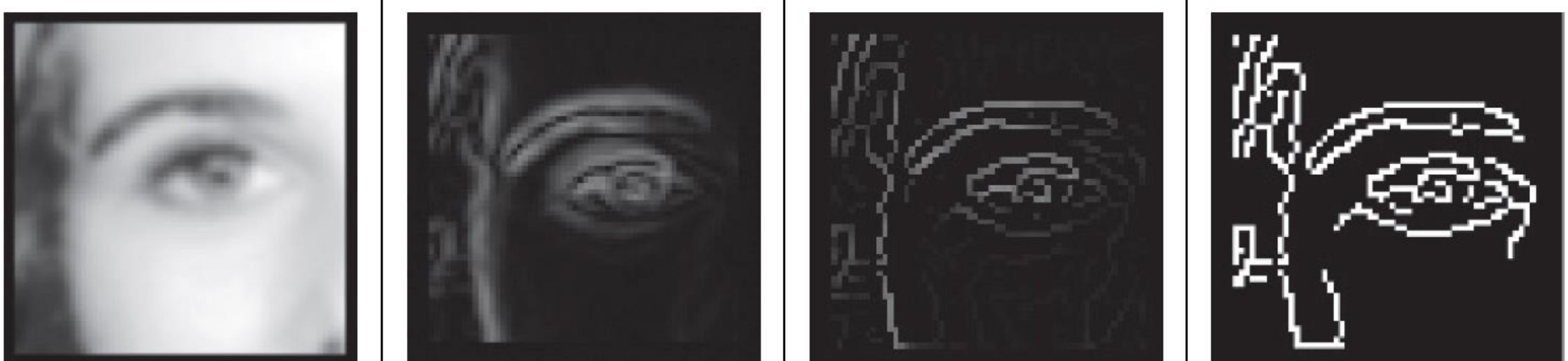
-

First-order Edge Detectors

Canny Operator: Approximation

Approximation has 4 steps/stages:

1. Use Gaussian smoothing
2. Use Sobel operator
3. Use nonmaximal suppression
4. **Threshold with hysteresis to connect edge points**

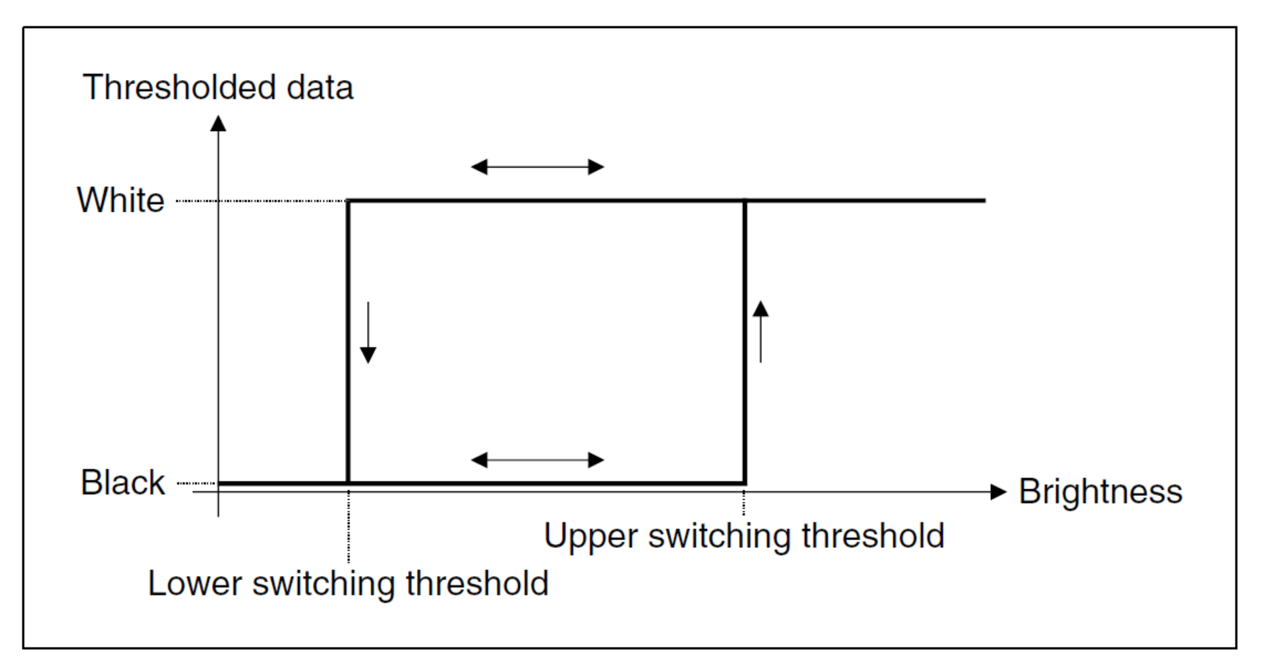


(a) Gaussian smoothing | (b) Sobel edge detection | (c) Non-maximum suppression | (d) Hysteresis thresholding

First-order Edge Detectors
Canny Operator: Hysteresis Thresholding

Hysteresis thresholding sets points to:

- White if upper threshold has been reached
- Black if lower threshold has been reached



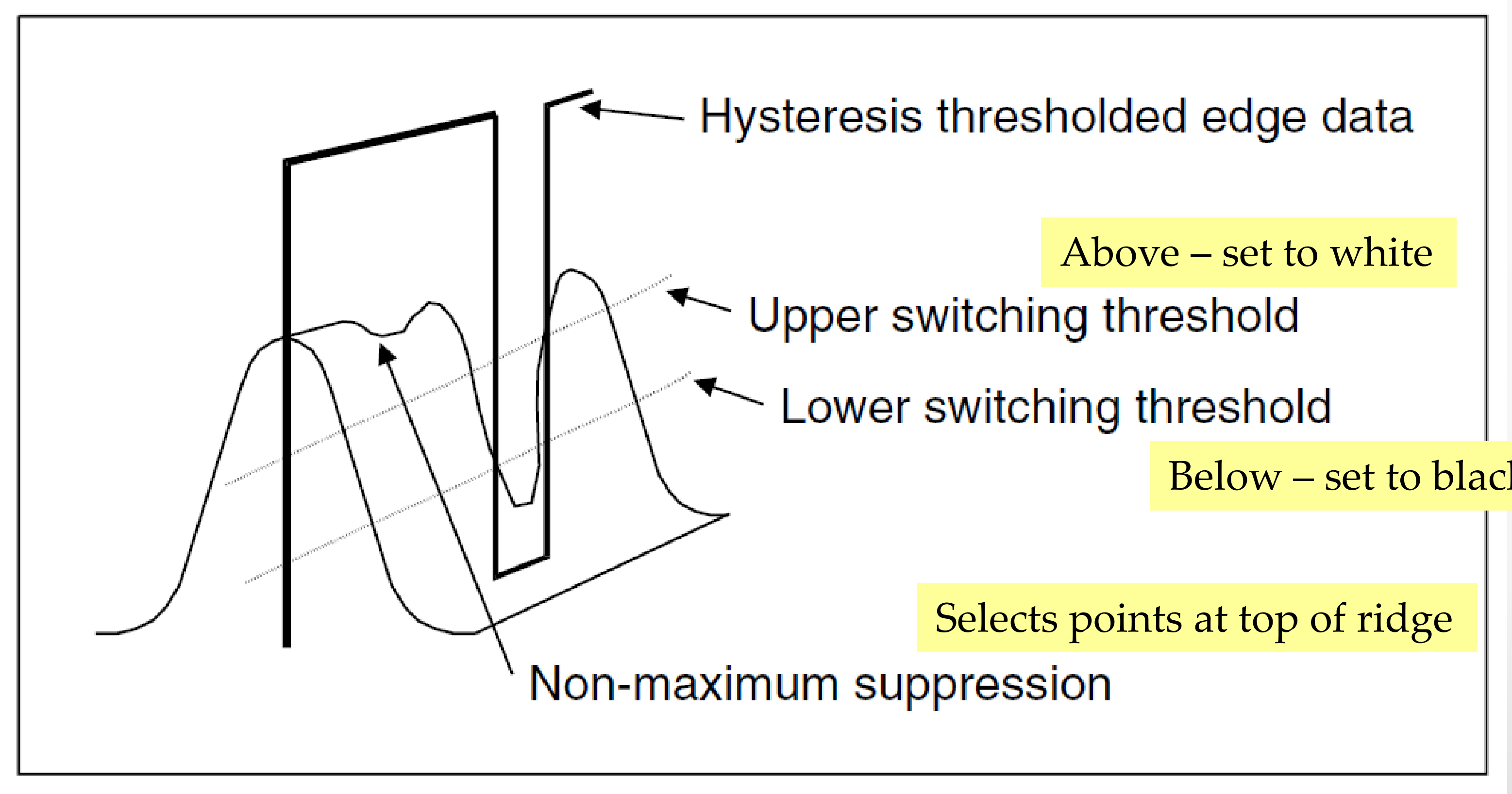
Arrows indicate possible movement: only 1 way to change from black to white and vice versa

First-order Edge Detectors

Canny Operator: Hysteresis Thresholding

Hysteresis thresholding sets points to:

- White if upper threshold has been reached
- Black if lower threshold has been reached



First-order Edge Detectors

Canny Operator: Hysteresis Thresholding

Hysteresis thresholding process:

1. Starts when an **edge point** from non-maximum suppression that exceeds upper threshold is found:
   - Labeled as edge point
   - Forms first point of a line of edge points
2. Neighbours of point are searched to determine whether they exceed the lower threshold
3. Any neighbour exceeding the lower threshold is labeled as an edge point and steps 2 and 3 are re-executed

First edge point becomes a seed point for a search.

It's neighbours become seed points if they exceed the lower threshold.

The search for each branch is terminated at points that have no neighbours above the lower threshold.

Requires recursion

First-order Edge Detectors

Canny Operator: Approximation

Which differences can you observe?

Hysteresis thresholding versus Uniform thresholding:

Sobol edge-detected eye

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

(b) Uniform thresholding, level = 40

(c) Uniform thresholding, level = 10

Too few points if threshold level is high

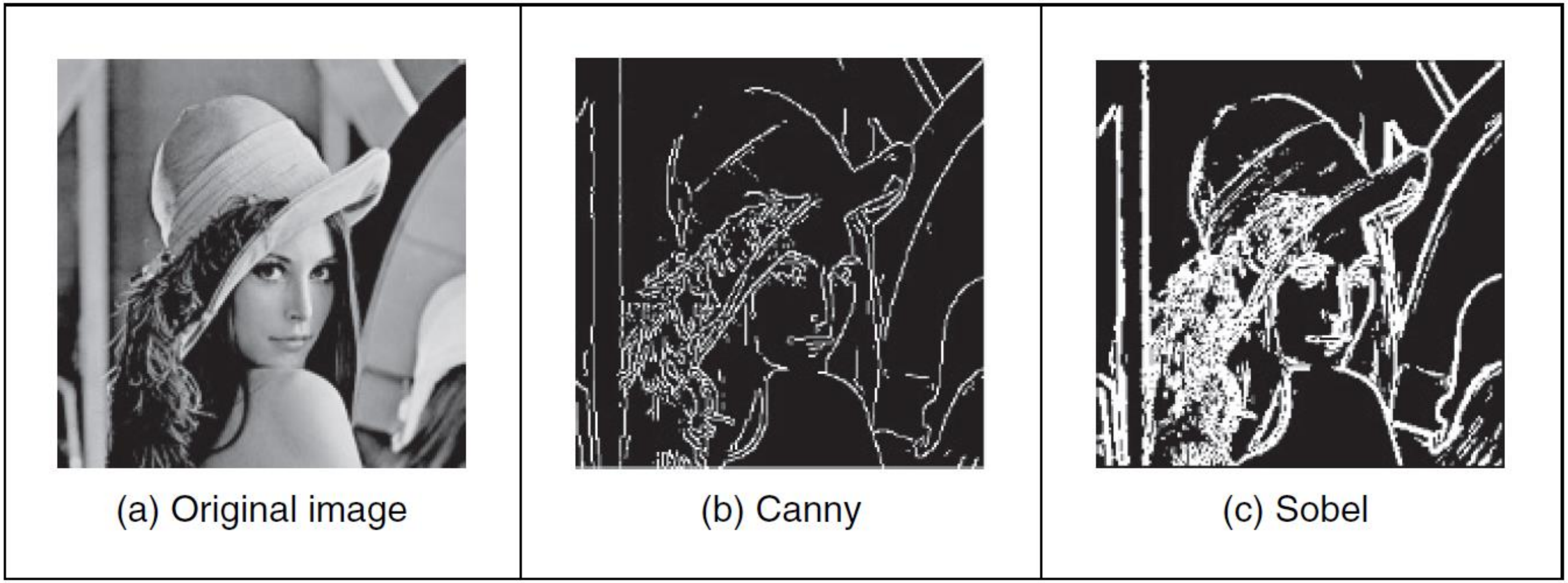
Too many points if threshold level is low

First-order Edge Detectors

Canny Operator: Approximation

Canny Operator versus Sobel Operator:

- Canny operator using *5*x*5* Gaussian operator
- *3*x*3* Sobol operator with uniform thresholding



(a) Original image (b) Canny (c) Sobel

Which differences can you observe?

First-order Edge Detectors
Review

Write out the equation for the following template:

2	–1
–1	0

First-order Edge Detectors

Review

Apply this template to the image on the right and write out the resulting matrix:

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

2	–1
–1	0

First-order Edge Detectors

Review

Can you apply the various operators discussed today to this image?

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

THE END