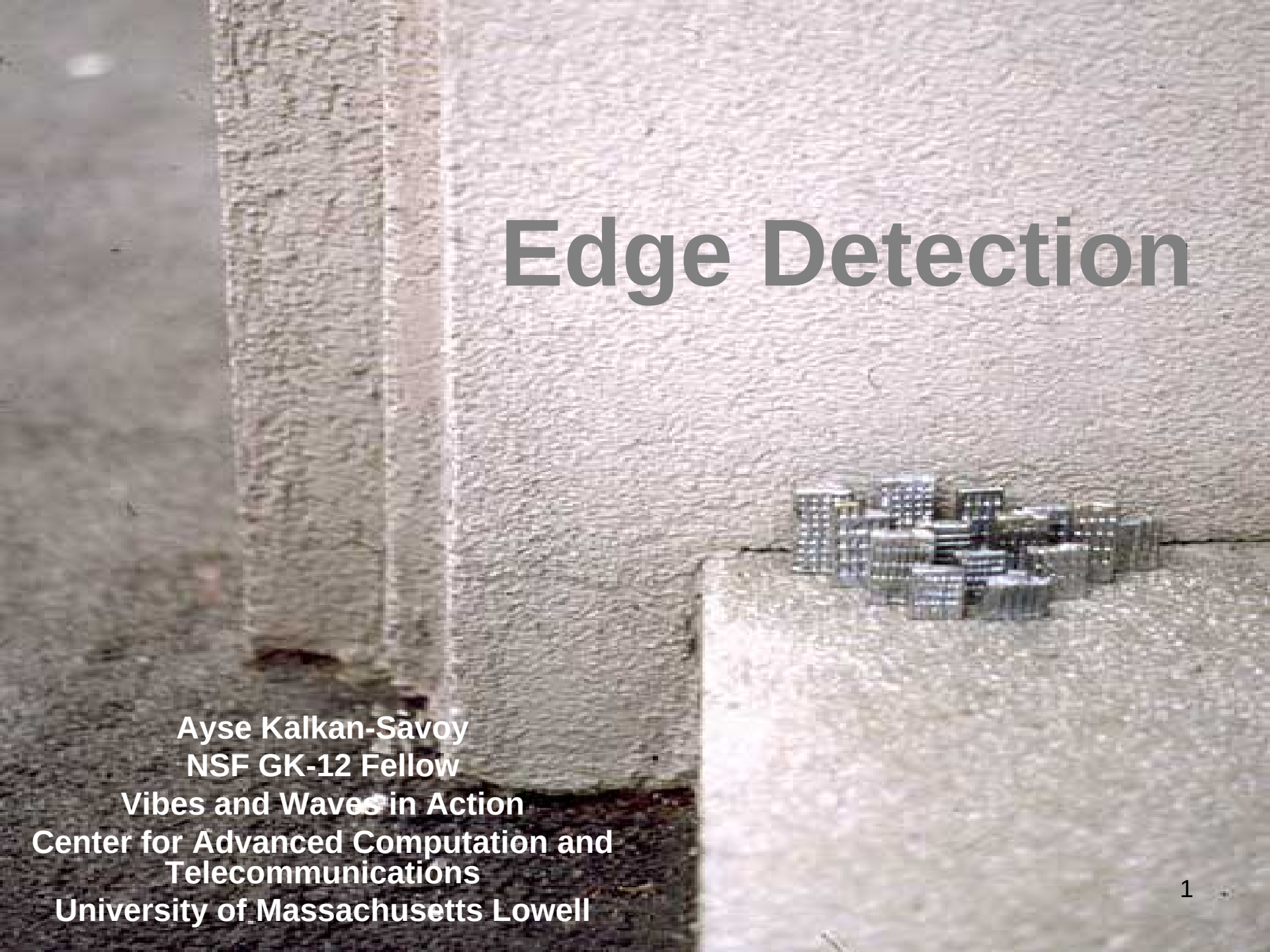
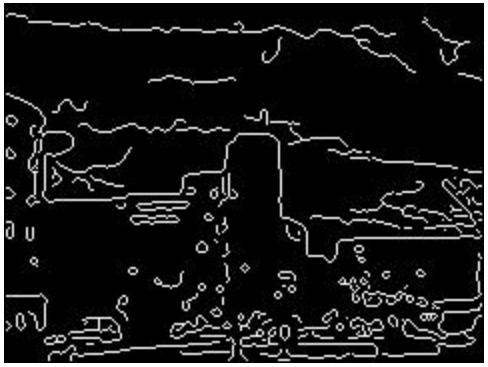
Edge Detection



Ayse Kalkan-Savoy
NSF GK-12 Fellow
Vibes and Waves in Action
Center for Advanced Computation and Telecommunications
University of Massachusetts Lowell

What is Edge Detection?

- Process to detect boundaries of objects in digital pictures.



Examples

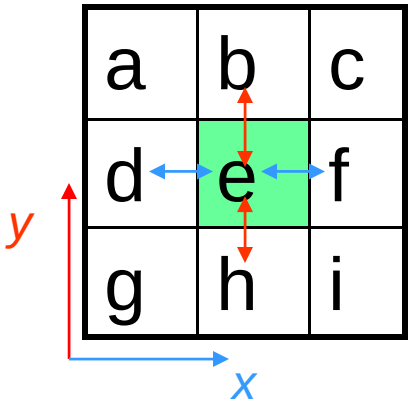
- A robot that has to pick up cubic parcels.
  - Edges of parcels needs to be recognized
- A face recognition program which detects faces first using edge detection
- In oil-sand exploration, to detect large lumps so as not to damage machinery
- To improve MRI and ultrasound imaging

How to find edges in 2-D Images?

- Digital Images are represented by a grid of ‘pixels’
  - Each pixel has an intensity representing color of the image at that location
- Edge is where a sudden change in color occurs.
- Intensity Values
  - Gray-scale image: usually an integer between 0-255.
  - RGB image: three integers each between 0-255.
- Basic edge detection: check rate of change in intensity values.
- Rate of intensity changes is calculated for each pixel comparing its intensity with its neighbors.

Rate of Intensity Change/Pixel

- 2D … Change in both horizontal (x) and vertical (y) direction.
- Square below is a 3x3 matrix. It is to represent a pixel and its surrounding pixels.
- a, b, c, d, .., i are intensity values.



Average of the sum of differences in x direction

$Ex = (e - d + f - e) / 2 = (f - d) / 2$

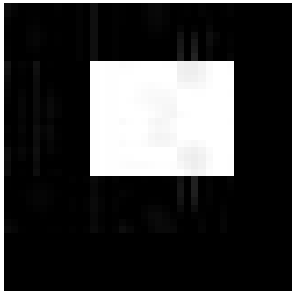
Average of the sum of differences in y direction

$Ey = (e - b + h - e) / 2 = (h - b) / 2$

Total magnitude for rate of intensity change:

$E = Ex^2 + Ey^2$

Example



Sample Image

This is a binary image.
Pixel values are either
1 or 0.
0: black
1: white

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

Intensity Values of Image

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

Ideal Results for
Edge Detected Image

Values of Ex and Ey

- **Ex** =

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	-0.5	-0.5	0	0	0	0.5	0.5	0
0	0	-0.5	-0.5	0	0	0	0.5	0.5	0
0	0	-0.5	-0.5	0	0	0	0.5	0.5	0
0	0	-0.5	-0.5	0	0	0	0.5	0.5	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

- **Ey** =

0	0	0	0	0	0	0	0	0	0
0	0	0	-0.5	-0.5	-0.5	-0.5	-0.5	0	0
0	0	0	-0.5	-0.5	-0.5	-0.5	-0.5	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0.5	0.5	0.5	0.5	0.5	0	0
0	0	0	0.5	0.5	0.5	0.5	0.5	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Values of E

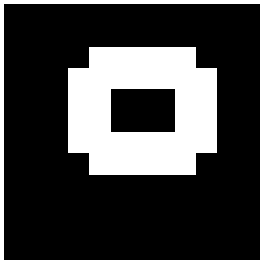
- Calculate $E = Ex^2+Ey^2$ for each pixel except boundaries.

0	0	0	0	0	0	0	0	0	0
0	0	0	0.25	0.25	0.25	0.25	0.25	0	0
0	0	0.25	0.50	0.25	0.25	0.25	0.50	0.25	0
0	0	0.25	0.25	0	0	0	0.25	0.25	0
0	0	0.25	0.25	0	0	0	0.25	0.25	0
0	0	0.25	0.50	0.25	0.25	0.25	0.50	0.25	0
0	0	0	0.25	0.25	0.25	0.25	0.25	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

- Select a threshold.
  If E < threshold, pixel intensity = 0, otherwise 1.

RESULT for Threshold = 0.20 :

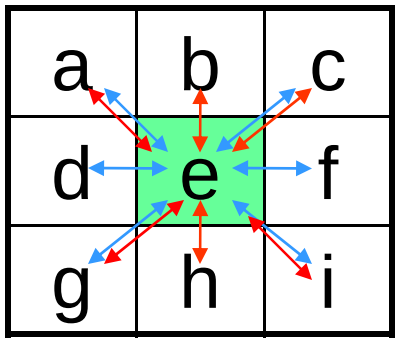
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0
0	0	1	1	0	0	0	1	1	0
0	0	1	1	0	0	0	1	1	0
0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



Edge Detected Image

Can we improve the result?

Enhance the computation by adding the diagonal neighbor pixel values.



Average of the sum of differences in x direction

Ex = (e-a + c-e + e-d + f-e + e-g + i-e)/6 = (c-a+f-d+i-g)/6

Average of the sum of differences in y direction

Ey = (e-a + e-b + e-c + g-e + h-e + i-e)/6 = (g+h+i-a-b-c)/6

Total magnitude for change in intensity:

$E = Ex^2 + Ey^2$

Values of Ex and Ey

- **Ex** =

0	0	0	0	0	0	0	0	0	0
0	0	-0.17	-0.17	0	0	0	0.17	0.17	0
0	0	-0.33	-0.33	0	0	0	0.33	0.33	0
0	0	-0.50	-0.50	0	0	0	0.50	0.50	0
0	0	-0.50	-0.50	0	0	0	0.50	0.50	0
0	0	-0.33	-0.33	0	0	0	0.33	0.33	0
0	0	-0.17	-0.17	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

- **Ey** =

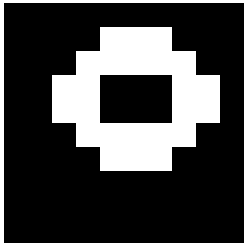
0	0	0	0	0	0	0	0	0	0
0	0	-0.17	-0.33	-0.50	-0.50	-0.50	-0.33	-0.17	0
0	0	-0.17	-0.33	-0.50	-0.50	-0.50	-0.33	-0.17	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0.17	0.33	0.50	0.50	0.50	0.33	0.17	0
0	0	0.17	0.33	0.50	0.50	0.50	0.33	0.17	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Results for Average Intensity Change (E)

0	0	0	0	0	0	0	0	0	0
0	0	0.06	0.14	0.25	0.25	0.14	0.06	0	0
0	0	0.14	0.22	0.25	0.25	0.25	0.22	0.14	0
0	0	0.25	0.25	0	0	0	0.25	0.25	0
0	0	0.25	0.25	0	0	0	0.25	0.25	0
0	0	0.14	0.22	0.25	0.25	0.25	0.22	0.14	0
0	0	0.06	0.14	0.25	0.25	0.25	0.14	0.06	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Results for Edges (Threshold=0.20)

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



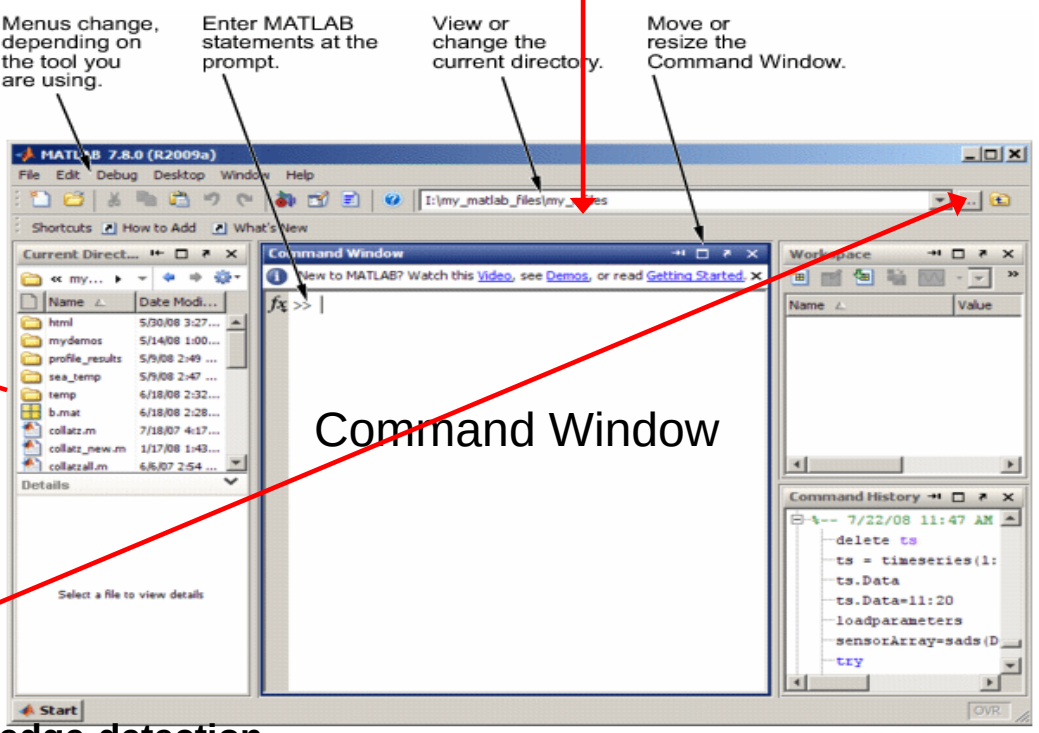
Edge Detected Image

Both algorithms resulted with an extra edge line around the real edges.

How can results of both methods be improved?

Matlab for Edge Detection

- Start Matlab. Q:\Matlab\MATLABR2009b
- Change current directory to: Q:/gk12/edge-detection



After clicking on this button,

select My Computer>Q>gk12>edge-detection

Matlab Image Commands

- Check your current directory window to see the image files.
- Read image using imread,
  x = imread(‘alaska.jpg’);

- Display the image.
  imshow(x)

- rgb2gray: convert color image to gray scale
  y = rgb2gray(x);

- Check image matrix size – if you like
  size(y)

- Display your gray scale image on a different window
  figure *->figure opens a new window*
  imshow(y)

Matlab Image Commands

- Edge detection command
  z = edge(y, ‘canny’); *canny: an edge detection method*

- Display result on a different window
  figure
  imshow(z)

- Edge Detection Methods you can try
  sobel, prewitt, roberts, log, zerocross, and canny

- Modify threshold to change edges. To see current threshold value
  [z threshold] = edge(y,’sobel’);
  threshold
  z = edge(y,’sobel’, *new-threshold*);
  figure
  imshow(z)