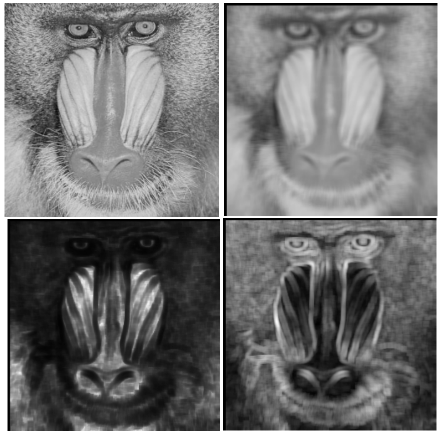
COMPSCI 373 S1C:
Computer Graphics and Image Processing

Part 3 – Image Processing

Moving Window Transform

Outline:

1. Examples of linear and non-linear filters
2. Moving Window Transform (MWT): main concepts
3. Gaussian linear filtering
4. Gradient-based edge detection



Local Means, St. Deviations, and Ratios

Window 3 x 3

Image $f(x,y)$	Means $\mu(x,y)$
Ratios $\rho(x,y)$	St.dev. $\sigma(x,y)$

$$\rho(x,y) = \frac{\mu(x,y)}{\sigma(x,y)}$$

Local Means, St.Deviations, and Ratios

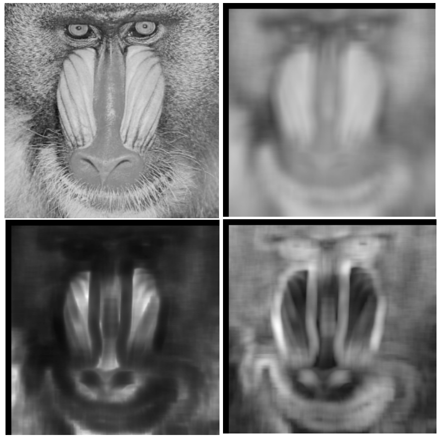
- Mean, standard deviation, and mean / deviation ratio of grey values in a rectangular window:

$$\mu(x,y) = \frac{1}{(2k+1)(2l+1)} \sum_{\xi=-k}^{k} \sum_{\eta=-l}^{l} f(x+\xi, y+\eta)$$

$$\sigma(x,y) = \sqrt{\frac{1}{(2k+1)(2l+1)} \sum_{\xi=-k}^{k} \sum_{\eta=-l}^{l} \left(f(x+\xi, y+\eta) - \mu(x,y)\right)^2}$$

$$\rho(x,y) = \mu(x,y) \Big/ \max\{\sigma(x,y),\, 1\}$$

  - ☐ Window centre (x,y) moves across an input image f



Local Means, St. Deviations, and Ratios

Window 7 x 7

Image $f(x,y)$	Means $\mu(x,y)$
Ratios $\rho(x,y)$	St.dev. $\sigma(x,y)$

$$\rho(x,y) = \frac{\mu(x,y)}{\sigma(x,y)}$$

Local Means, St. Deviations, and Ratios

Window 3 x 15

Image $f(x,y)$	Means $\mu(x,y)$
Ratios $\rho(x,y)$	St.dev. $\sigma(x,y)$

$$\rho(x,y) = \frac{\mu(x,y)}{\sigma(x,y)}$$

Moving Window Transform (MWT)

- Slides 2-5 exemplify moving window transform:
  - ☐ Local average (mean) linear filter
  - ☐ Local non-linear filter: scaled standard deviation
  - ☐ Local non-linear filter: scaled mean/deviation ratio
- Value of the transformed image, at pixel location (x,y) is a function of values of the original image in a $2k+1$ by $2l+1$ (e.g. $k=l=1$ for the 3×3 window) rectangle centred on pixel location (x,y)
- General MWT multiplies each kernel coefficient with the image value that lies directly beneath it

---

MWT – Example 1 (Linear Filter)

- Weighed mean (3 x 3 window):

$$\mu(x,y) = \frac{1}{\sum_{\xi=-1}^{1}\sum_{\eta=-1}^{1} s(\xi,\eta)} \sum_{\xi=-1}^{1}\sum_{\eta=-1}^{1} s(\xi,\eta) f(x+\xi,y+\eta)$$

kernel s

$\frac{1}{8}\times$

1	1	1
1	0	1
1	1	1

f

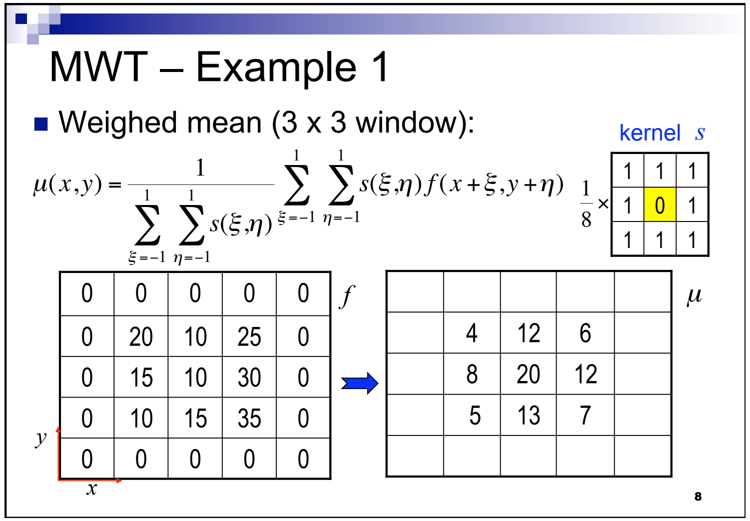
0	0	0	0	0
0	20	10	25	0
0	15	10	30	0
0	10	15	35	0
0	0	0	0	0

y

x

➡

μ



MWT – Example 1

- Weighed mean (3 x 3 window):

$$\mu(x,y)=\frac{1}{\sum_{\xi=-1}^{1}\sum_{\eta=-1}^{1}s(\xi,\eta)}\sum_{\xi=-1}^{1}\sum_{\eta=-1}^{1}s(\xi,\eta)f(x+\xi,y+\eta)$$

kernel s

$\frac{1}{8}\times$

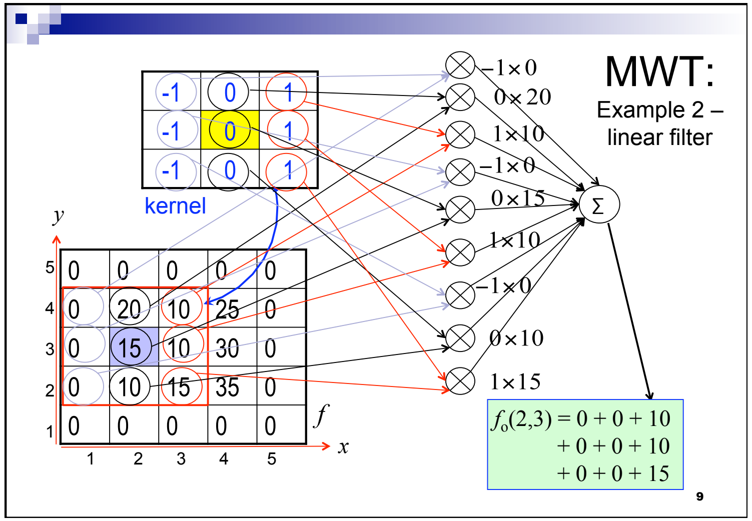
1	1	1
1	0	1
1	1	1

f

0	0	0	0	0
0	20	10	25	0
0	15	10	30	0
0	10	15	35	0
0	0	0	0	0

μ

	4	12	6	
	8	20	12	
	5	13	7	



MWT:

Example 2 – linear filter

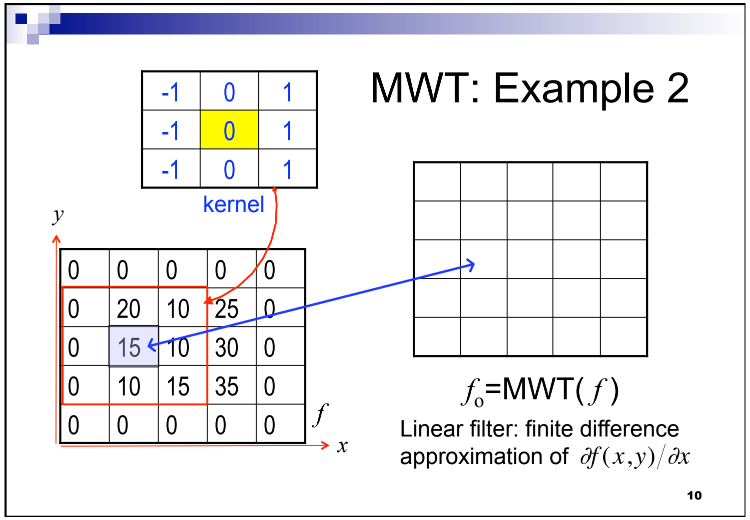
-1	0	1
-1	0	1
-1	0	1

kernel

f

y					
5	0	0	0	0	0
4	0	20	10	25	0
3	0	15	10	30	0
2	0	10	15	35	0
1	0	0	0	0	0
	1	2	3	4	5

$$f_o(2,3)=0+0+10+0+0+10+0+0+15$$



MWT: Example 2

-1	0	1
-1	0	1
-1	0	1

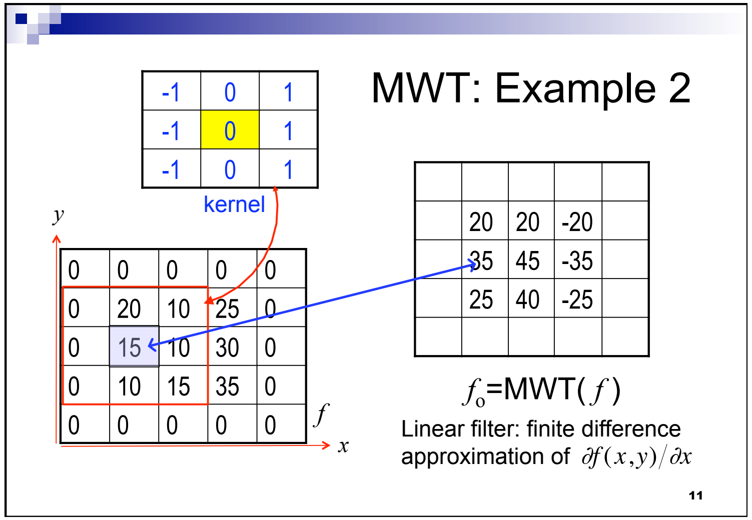
kernel

f

0	0	0	0	0
0	20	10	25	0
0	15	10	30	0
0	10	15	35	0
0	0	0	0	0

f_o=MWT(f)

Linear filter: finite difference approximation of $\partial f(x,y)/\partial x$



MWT: Example 2

-1	0	1
-1	0	1
-1	0	1

kernel

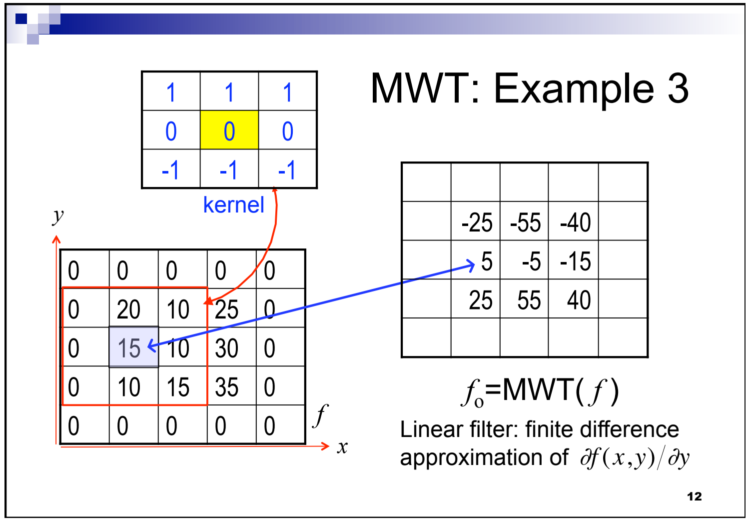
f

0	0	0	0	0
0	20	10	25	0
0	15	10	30	0
0	10	15	35	0
0	0	0	0	0

	20	20	-20	
	35	45	-35	
	25	40	-25	

f_o=MWT(f)

Linear filter: finite difference approximation of $\partial f(x,y)/\partial x$



MWT: Example 3

1	1	1
0	0	0
-1	-1	-1

kernel

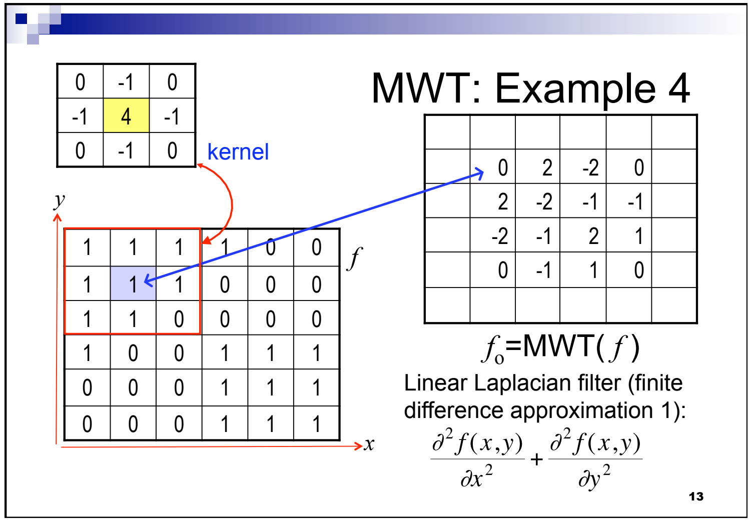
f

0	0	0	0	0
0	20	10	25	0
0	15	10	30	0
0	10	15	35	0
0	0	0	0	0

	-25	-55	-40	
	5	-5	-15	
	25	55	40	

f_o=MWT(f)

Linear filter: finite difference approximation of $\partial f(x,y)/\partial y$



MWT: Example 4

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

kernel

f

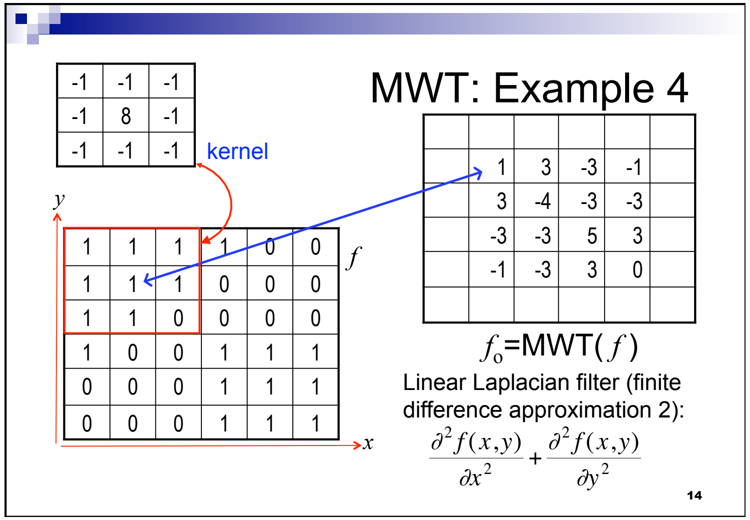
1	1	1	1	0	0
1	1	1	0	0	0
1	1	0	0	0	0
1	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

	0	2	-2	0	
	2	-2	-1	-1	
	-2	-1	2	1	
	0	-1	1	0	

f_o=MWT(f)

Linear Laplacian filter (finite difference approximation 1):

$$\frac{\partial^2 f(x,y)}{\partial x^2}+\frac{\partial^2 f(x,y)}{\partial y^2}$$



MWT: Example 4

-1	-1	-1
-1	8	-1
-1	-1	-1

kernel

f

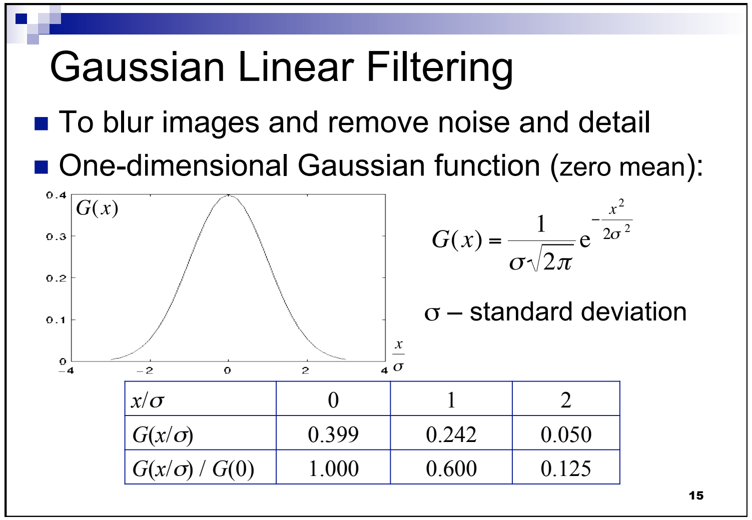
1	1	1	1	0	0
1	1	1	0	0	0
1	1	0	0	0	0
1	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

	1	3	-3	-1	
	3	-4	-3	-3	
	-3	-3	5	3	
	-1	-3	3	0	

f_o=MWT(f)

Linear Laplacian filter (finite difference approximation 2):

$$\frac{\partial^2 f(x,y)}{\partial x^2}+\frac{\partial^2 f(x,y)}{\partial y^2}$$



Gaussian Linear Filtering

- To blur images and remove noise and detail
- One-dimensional Gaussian function (zero mean):

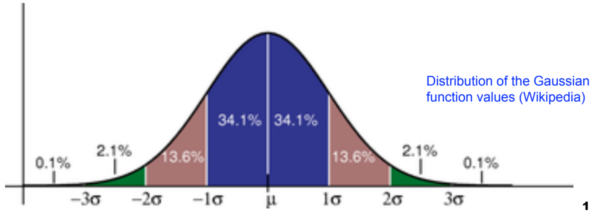
$$G(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}$$

σ – standard deviation

x/σ	0	1	2
$G(x/\sigma)$	0.399	0.242	0.050
$G(x/\sigma)$ / $G(0)$	1.000	0.600	0.125

Gaussian Linear Filtering

- Standard deviation of the Gaussian probability density function guides its behaviour:
  - ☐ 68% of the x-values: the range [mean – σ, mean + σ]
  - ☐ 95% of the x-values: [mean – 2σ, mean + 2σ]
  - ☐ 99.7% of the x-values: [mean – 3σ, mean + 3σ]

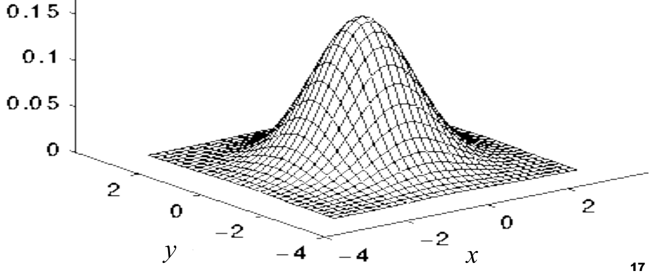


Distribution of the Gaussian function values (Wikipedia)

Gaussian Linear Filtering

- 2D Gaussian filter is built using the 2D Gaussian

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



Gaussian Linear Filtering

- Filter kernel s: an integer-valued approximation of digitised continuous 2D Gaussian function
  - ☐ Example: 5 × 5 window; $\sigma = 1$

$\frac{1}{273}$

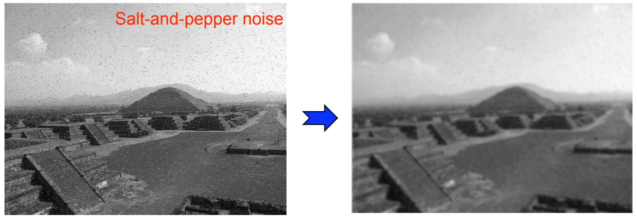
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

  - ☐ The larger the value of σ, the wider the peak of the Gaussian and the larger the blurring

---

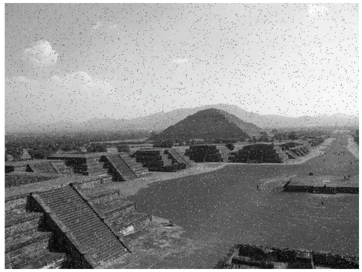
Gaussian Linear Filtering

- Non-uniform averaging: low pass filtering
- Rotational symmetry with no directional bias
- Fast computations due to separability (2D=1D×1D)
- Might not preserve image brightness



Gaussian Versus Median Filtering

- Gaussian filter: blurred edges; residual noise

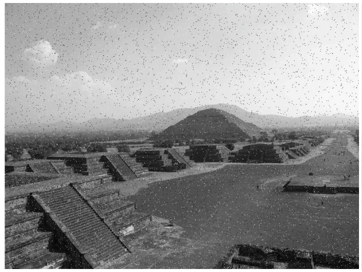


Salt-and-pepper noise

Filtered image

Gaussian Versus Median Filtering

- Median filter: preserved edges; removed noise

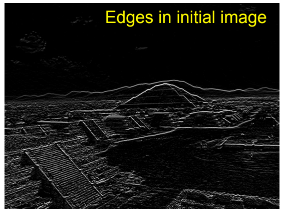


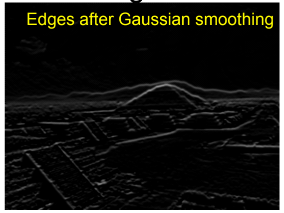
Salt-and-pepper noise

Filtered image

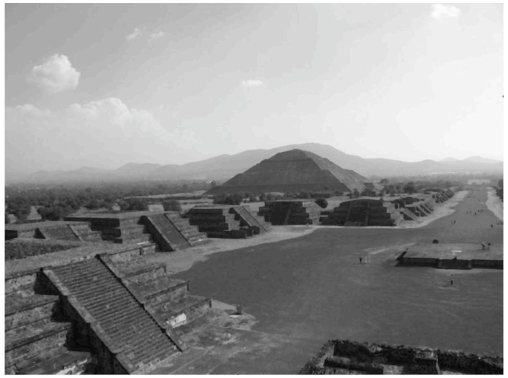
Gaussian Linear Filtering

- Inefficient for removing salt-and-pepper noise
  - ☐ Averaging is not robust to outliers (large deviations)
  - ☐ Median is much more robust with respect to outliers
- Efficient for image smoothing to more accurate approximation of derivatives in edge detection

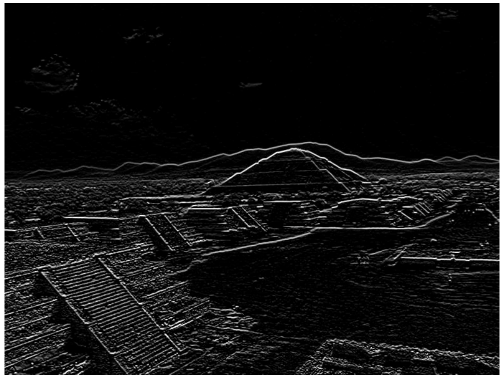




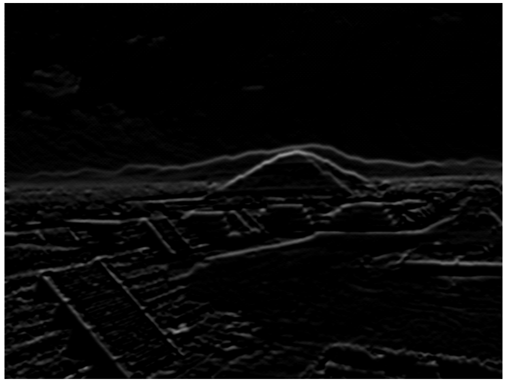
Initial Image to Detect Edges



Edge Enhancement



Edges Enhanced after Smoothing



MWT Based Edge Detection

- One of major applications of image filtering
- Formal definition of an edge
  - Locations of sudden grey level /colour changes
  - Transition between objects or object and background
  - Locations that attract visual attention
- **Problem**: Image noise has similar properties
- Conventional 3-step approach:

(*i*) Noise reduction (preserving edges as much as possible);
(*ii*) Edge enhancement; (*iii*) Edge localisation

MWT Based Edge Detection

- Estimation of the grey level gradient at pixels

$$g_x(x,y) = \frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y) - f(x-1,y)}{2}$$

$$g_y(x,y) = \frac{\partial f(x,y)}{\partial y} \approx \frac{f(x,y+1) - f(x,y-1)}{2}$$

- Gradient magnitude $g(x,y)$ and angle $a(x,y)$:

$$g(x,y) = \sqrt{g_x^2(x,y) + g_y^2(x,y)} \approx \left|g_x(x,y)\right| + \left|g_y(x,y)\right|$$

$$a(x,y) = \tan^{-1}\left({}^{g_y(x,y)}\!\big/_{g_x(x,y)} \right)$$

MWT Based Edge Detection

- Noise smoothing using a low-pass filter (mean, Gaussian, etc)
  - Separable Prewitt kernels (for the 3 x 3 averaging):

$$\frac{1}{6}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{1}{6}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

  - Separable Sobel kernels (for the 3 x 3 weighed mean):

$$\frac{1}{8}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Edge Detection Using Gradients

- Sign-alternate kernels:
  - Output images with positive and negative values
  - Display: by mapping zero gradient to mid-grey level
    - Positive gradient values appear brighter
    - Negative gradient values appear darker
- If meaningful edges are supposed to be strong, thresholded gradients form a binary edge map
  - Problem 1: Non-sharp edges for gradual transitions
  - Problem 2: Strong edges produced by noise
  - Problem 3: Edge localisation at ridges in the map

Edge Detection Using Gradients

- Initial image

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/gray/gm_2_4/130026.html

---

Edge Detection Using Gradients

- Automatically detected edges

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/gray/gm_2_4/130026.html

Edge Detection Using Gradients

- Human sketch of the meaningful boundaries

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/gray/gm_2_4/130026.html