CS 331
Computer Vision

09 / 12 / 2024

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- More Cross-correlation / convolution
- Second derivative masks explanation
- Using the horizontal and vertical edge filters to determine magnitude and angle of an edge
- Show edge detection results of prewitt masks on an image of Skidmore library
- In-class exercise on cross-correlation and edge magnitude and angle

Second derivative masks

- Examples of two dimensional second derivative masks:

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

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

Second derivative masks

- Example of a one dimensional second derivative mask:

-1 2 -1

Second derivative masks

```
original image:
   10  10  10  10  15  20  25  30  30  30  30


   notice there is a gradual edge (I will draw on board) there and if
you used a 1st derivative mask like:  1 0 -1    we get:


        0   0  -5 -10 -10 -10  -5   0   0


The first derivative mask seems to detect stronger edge pixels in the
3 middle ones and less strong edge pixels next to those.


original image again:
   10  10  10  10  15  20  25  30  30  30  30


   if you used a 2nd derivative mask like: -1  2 -1  we get:


        0   0  -5   0   0   0   5   0   0
```

Second derivative masks

The second derivative mask detects where the two "sides" of the edge are.

Here the second derivative mask is negative where the image function is **concave up** and is positive where the image function is **concave down**

You may have learned in calculus that this is the opposite. So a true second derivative mask really should be something like: 1 -2 1 rather than -1 2 -1 and that will get you a positive value of the second derivative when the function is concave up and a negative for concave down

(figure from Shapiro's Computer Vision book)

$$\textbf{mask } \mathbf{M} = [\mathbf{-1, 2, -1}]$$

S_1			12	12	12	12	12	24	24	24	24	24
S_1	$\otimes$	M	0	0	0	0	-12	12	0	0	0	0

(a) S_1 is an upward step edge

S_2			24	24	24	24	24	12	12	12	12	12
S_2	$\otimes$	M	0	0	0	0	12	-12	0	0	0	0

(b) S_2 is a downward step edge

S_3			12	12	12	12	15	18	21	24	24	24
S_3	$\otimes$	M	0	0	0	-3	0	0	0	3	0	0

(c) S_3 is an upward ramp

S_4			12	12	12	12	24	12	12	12	12	12
S_4	$\otimes$	M	0	0	0	-12	24	-12	0	0	0	0

(d) S_4 is a bright impulse or "line"

Figure 5.12: Cross correlation of four special signals with second derivative edge detecting mask $\mathbf{M} = [\mathbf{-1, 2, -1}]$; (a) upward step edge, (b) downward step edge, (c) upward ramp, and (d) bright impulse. Since the coordinates of $\mathbf{M}$ sum to zero, response on constant regions is zero. Note how a *zero-crossing* appears at an output position where different trends in the input signal join.

Convolution

- Convolution
  - is similar to cross-correlation except the upper-left value in the mask is multiplied by the lower-right value in the image neighborhood and so on until the lower-right value in the mask is multiplied by the upper-left value in the image.
  - think of the mask being flipped along the horizontal axis and then again along the vertical axis AND THEN doing cross-correlation with the flipped mask
- If the mask's values are symmetrical both vertically and horizontally, then cross-correlation and convolution yield the same result.

Convolution

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

$f(x,y)$

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

$h(x,y)$

=

69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

$g(x,y)$

Figure 3.10 *Neighborhood filtering (convolution): The image on the left is convolved with the filter in the middle to yield the image on the right. The light blue pixels indicate the source neighborhood for the light green destination pixel.*

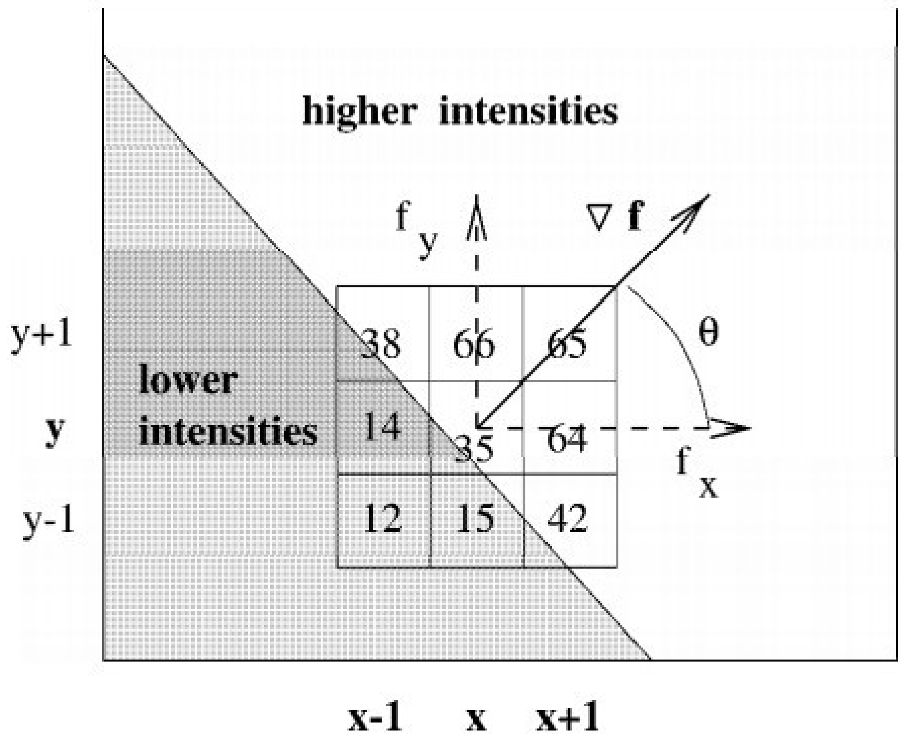
Estimating the gradient

- Often we want to find boundaries in images and know how sharp the boundaries are and in which direction the boundaries are.
- Given f(x,y) as our image function, we would like to compute the gradient at pixel [x,y].
- The magnitude of the gradient will give us an indication of how sharp a boundary is, and the angle of the gradient will give us the direction.
- If we want to estimate the magnitude and direction of the gradient (where the maximum change occurs) do the following:
  - the partial derivative of f w.r.t. x and the partial derivative of f w.r.t y make up the gradient.

Estimating the gradient

- The partial derivative of f w.r.t. x (f_x) and the partial derivative of f w.r.t y (f_y) make up the gradient.
- We can estimate these values in a discrete image at pixel x,y:
  - for f_x: take the difference between the pixel at x-1, y and x+1, y and divide by the change in x (2 pixels)
  - for f_y: take the difference between the pixel at x, y-1 and x, y+1 and divide by the change in y (2 pixels)
- We can get a better estimate if we average these calculations above and below and to the right and left of the pixel in question.
- The magnitude of the gradient is the square root of the sum of the squares of the partials
- The direction of the gradient is the angle which is arctan(f_y / f_x)

Estimating the gradient (figure from Shapiro & Stockman – Computer Vision book)



$$f_y = ((38-12)/2 + (66-15)/2 + (65-42)/2) / 3 = (13 + 25 + 11) / 3 = 16$$

$$f_x = ((65-38)/2 + (64-14)/2 + (42-12)/2) / 3 = (13 + 25 + 15) / 3 = 18$$

$$\theta = \tan^{-1} (16 / 18) = 0.727 \text{ rad} = 42 \text{ degrees}$$

$$|\nabla f| = (16^2 + 18^2)^{1/2} = 24$$

Figure 5.14: Estimating the magnitude and direction of contrast at I[x,y] by estimating the gradient magnitude and direction of picture function $f(x, y)$ using the discrete samples in the image array.

Estimating the gradient

- These lead to the Prewitt masks

$$\mathbf{G_x} = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * \mathbf{A}$$

- The Sobel masks come from the same idea but with twice the weight given to the row/col the pixel is in.

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

In class exercises

Cross-correlation with horizontal Prewitt mask

Determining angle and magnitude of an edge

Cross-correlation with a smoothing mask