Detecting intensity changes in a 2D image

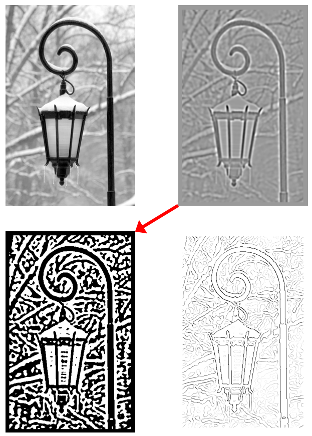
image

image after smoothing and second derivative

black = negative

white = positive

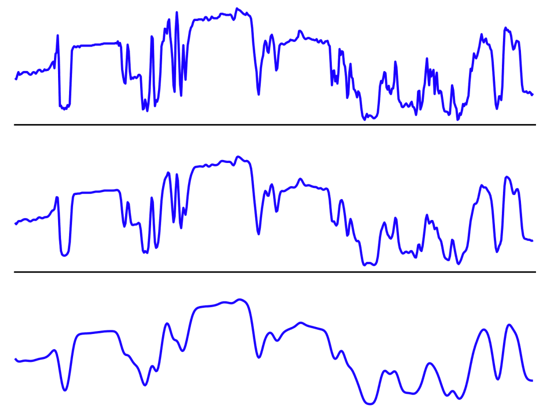
zero-crossing contours



"edge detection"

1

Smooth intensity to reduce noise + set scale



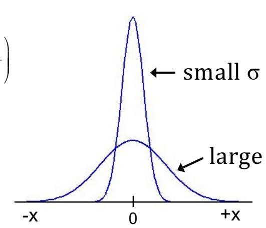
2

Smoothing the image intensities

Strategy 1: compute the *average* intensity in a neighborhood around each image location

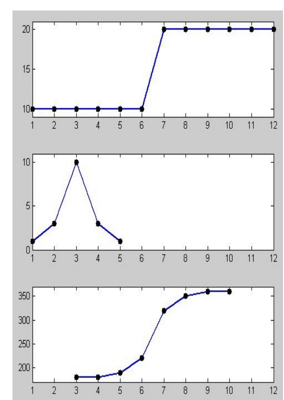
Strategy 2: combine intensities in the neighborhood using a smooth function that weighs nearby intensities more heavily, such as a *Gaussian function*

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



3

Convolution in one dimension



1	31	16	30	18	30	16	30	13	30	13	1
10	10	10	10	10	10	20	20	20	20	20	20

intensity

I(x)

convolution operator

G(x)

1	3	10	3	1
1/5	1/5	1/5	1/5	1/5

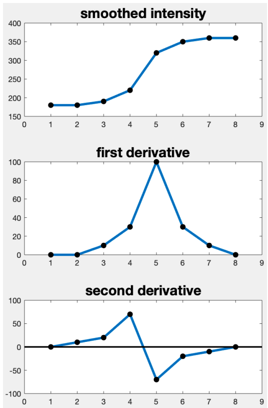
convolution result

0	0	180	180	190	220	320	350	360	360	0	0

G(x) * I(x)

4

Computing derivatives with convolution



+1 -1

smoothed intensity

0	0	180	180	190	220	320	350	360	360	0	0

↑ ↑ ...

1^{st} derivative

0	0	0	0	10	30	100	30	10	0	0	0

↑ ↑ ...

2^{nd} derivative

0	0	0	10	20	70	-70	-20	-10	0	0	0

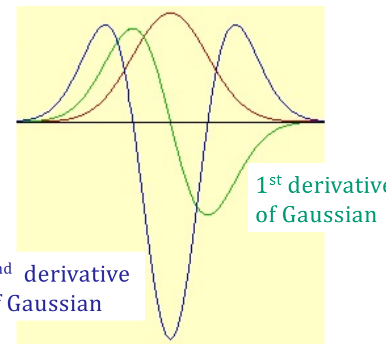
↑ ↑ ...

5

The derivative of a convolution

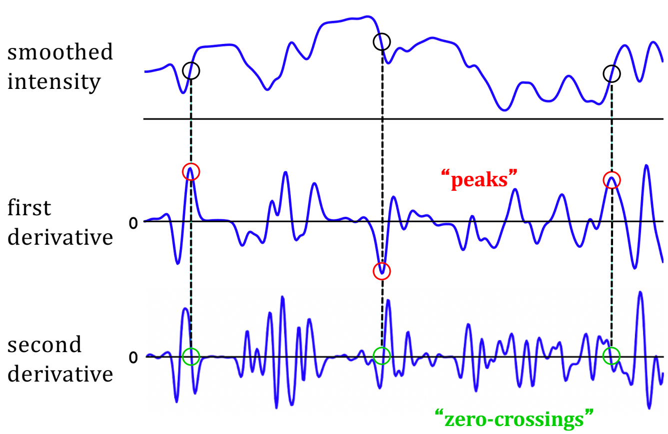
$$\frac{d}{dx}(G(x) * I(x)) = \left(\frac{d}{dx}G(x)\right) * I(x)$$

$$\frac{d}{dx}\left(\frac{d}{dx}G(x) * I(x)\right) = \left(\frac{d^2}{dx^2}G(x)\right) * I(x)$$



6

Derivatives of smoothed intensity



7

Convolution in two dimensions

1	3	1
3	8	3
1	3	1

2D convolution operator

*

1_1	1_3	1_1	8	8	8
1_3	1_8	1_3	8	8	8
1_1	1_3	1_1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8

image

=

convolution result

	24				

8

Convolution in two dimensions

1	3	1
3	8	3
1	3	1

2D convolution operator

*

1	1_{1}	1_{3}	8_{1}	8	8
1	1_{3}	1_{8}	8_{3}	8	8
1	1_{1}	1_{3}	8_{1}	8	8
1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8

image

=

convolution result

	24	59			

9

Convolution in two dimensions

1	3	1
3	8	3
1	3	1

2D convolution operator

*

1	1_{1}	1_{3}	8_{1}	8	8
1	1_{3}	1_{8}	8_{3}	8	8
1	1_{1}	1_{3}	8_{1}	8	8
1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8

image

=

convolution result

	24	59	157	192	

10

Convolution in two dimensions

1	3	1
3	8	3
1	3	1

2D convolution operator

*

1	1_{1}	1_{3}	8_{1}	8	8
1	1_{3}	1_{8}	8_{3}	8	8
1	1_{1}	1_{3}	8_{1}	8	8
1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8

image

=

convolution result

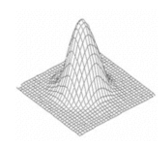
0	0	0	0	0	0
0	24	59	157	192	0
0	24	59	157	192	0
0	24	59	157	192	0
0	24	59	157	192	0
0	0	0	0	0	0

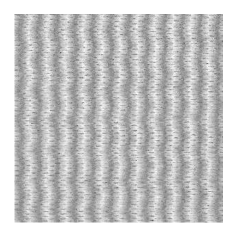
11

Smoothing a 2D image

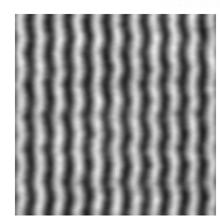
To smooth a 2D image *I(x,y)*, we convolve with a 2D Gaussian:

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





image



result of convolution

G(x,y) * I(x,y)

12

Convolution in two dimensions

1	3	1
3	8	3
1	3	1

2D convolution operator

*

1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8
1	1	1	8	8	8

image

=

convolution result

0	0	0	0	0	0
0	~~24~~	~~59~~	~~157~~	~~192~~	0
0	~~24~~	~~59~~	~~157~~	~~192~~	0
0	~~24~~	~~59~~	~~157~~	~~192~~	0
0	~~24~~	~~59~~	~~157~~	~~192~~	0
0	0	0	0	0	0

compute derivatives row-by-row, in the horizontal direction?

13

Detecting edges at all orientations

1	3	1
3	8	3
1	3	1

2D convolution operator

*

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8

image

=

convolution result

0	0	0	0	0	0
0	~~24~~	~~24~~	~~24~~	~~24~~	0
0	~~59~~	~~59~~	~~59~~	~~59~~	0
0	~~157~~	~~157~~	~~157~~	~~157~~	0
0	~~192~~	~~192~~	~~192~~	~~192~~	0
0	0	0	0	0	0

compute derivatives row-by-row, in the horizontal direction?

also need to compute derivatives in the vertical direction!

14

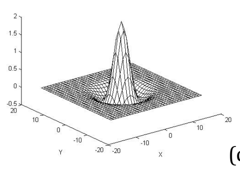
2D Laplacian derivative operator

To differentiate the smoothed image, we will use the *Laplacian* operator:

$$\nabla^2 = \left(\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \right)$$

We can again combine the smoothing and derivative operations:

$$\nabla^2 [G(x,y) * I(x,y)] = [\nabla^2 G(x,y)] * I(x,y)$$



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

(displayed with sign reversed)

15

Detect edges at all orientations

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	10	10	10	10	0	0	0
0	0	10	10	10	10	10	10	0	0
0	0	10	10	10	10	10	10	0	0
0	0	0	10	10	10	10	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

image

(with edges at many orientations)

Convolution operator

0	1	0
1	-4	1
0	1	0

Laplacian

0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	20	-20	-20	20	0	0	0
0	0	20	-20	0	0	-20	20	0	0
0	10	-20	0	0	0	0	-20	10	0
0	10	-20	0	0	0	0	-20	10	0
0	0	20	-20	0	0	-20	20	0	0
0	0	0	20	-20	-20	20	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	0	0	0	0	0	0	0

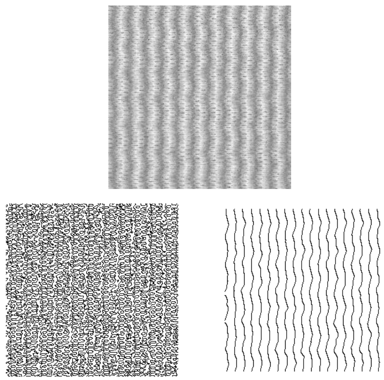
convolution result

(with zero-crossings highlighted)

16

Detecting intensity changes at multiple scales

small σ



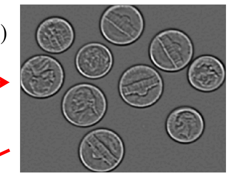
large σ

zero-crossings of convolutions of the image
with ∇^2G operators of different size

17

Computing the contrast of intensity changes

$[\nabla^2 G(x,y)] * I(x,y)$



zero-crossings with slope
displayed as darkness

-18	-12	13
-10	14	17
12	20	26

$$slope = \sqrt{dx^2 + dy^2}$$

18