Image Filtering and Convolution

CS 4391 Introduction Computer Vision

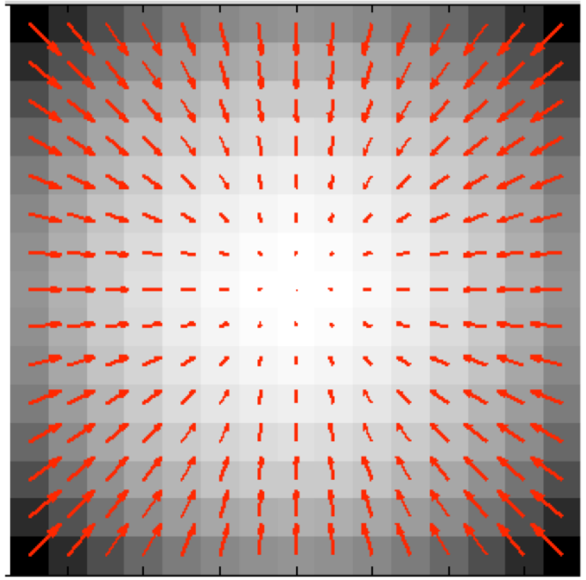
Professor Yu Xiang

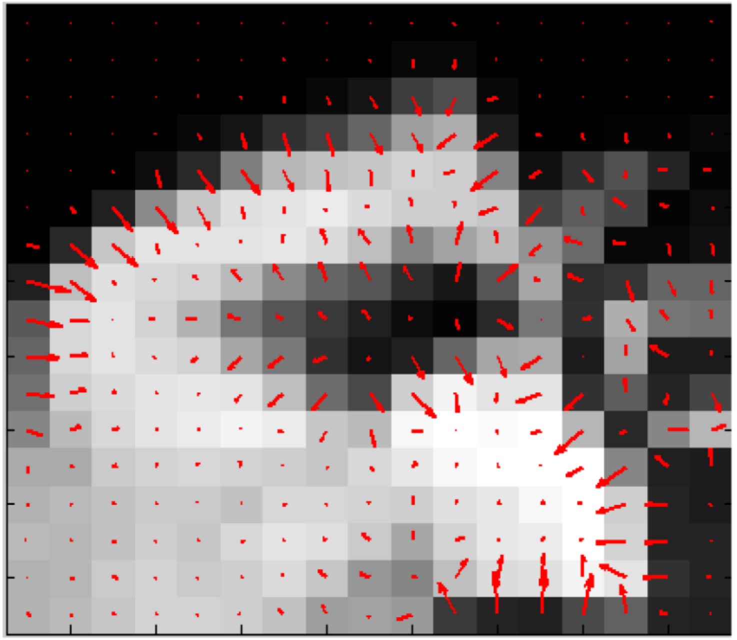
The University of Texas at Dallas

Recall: Image Gradient

- Gradient = Vector of partial derivatives of image

- Gradient direction is the steepest direction to increase the function value





Numerical Derivatives

Finite forward difference

$$\frac{f(x+h)-f(x)}{h} = f'(x) + O(h)$$

Finite backward difference

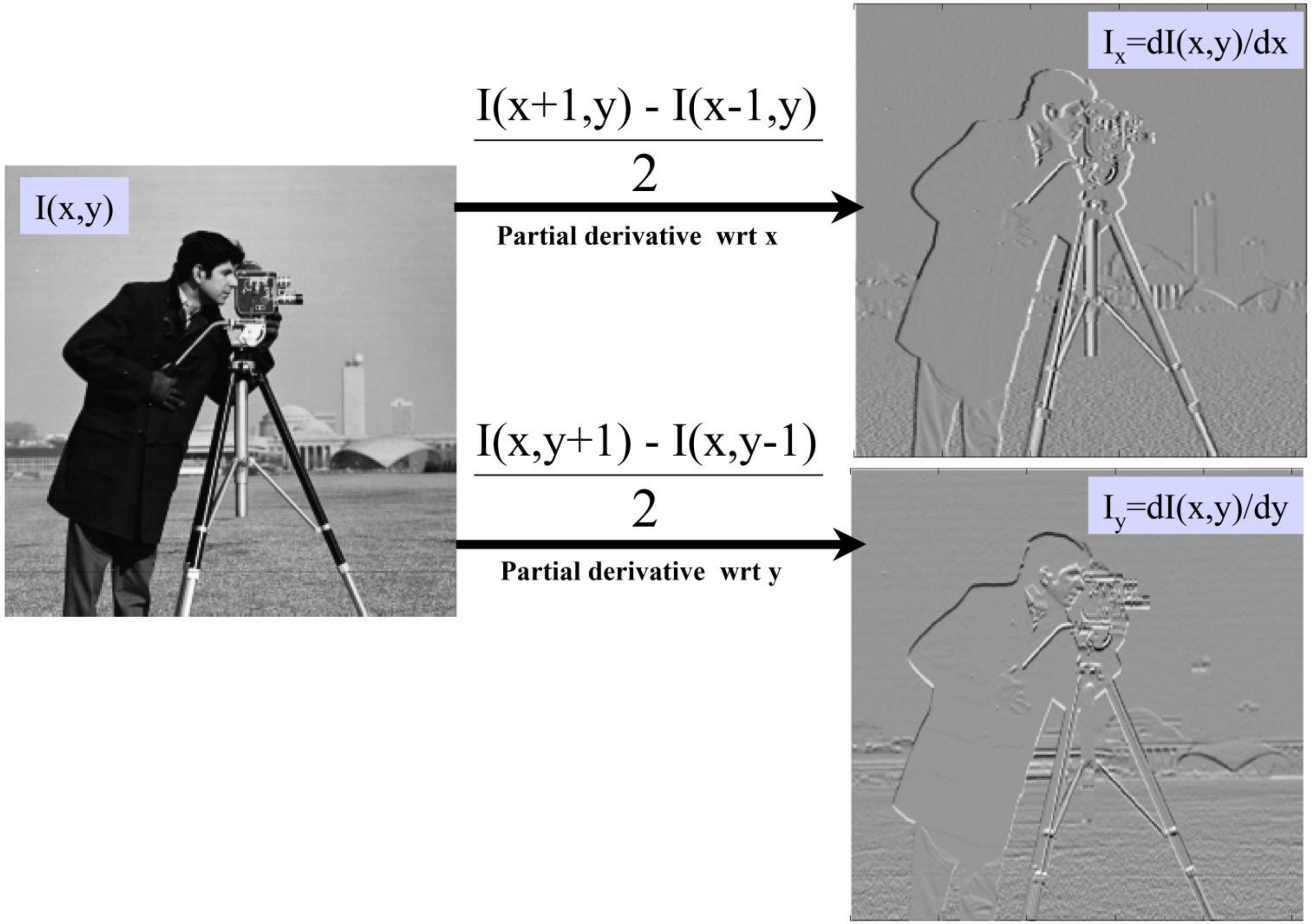
$$\frac{f(x)-f(x-h)}{h} = f'(x) + O(h)$$

Finite central difference

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + O(h^2)$$

More accurate

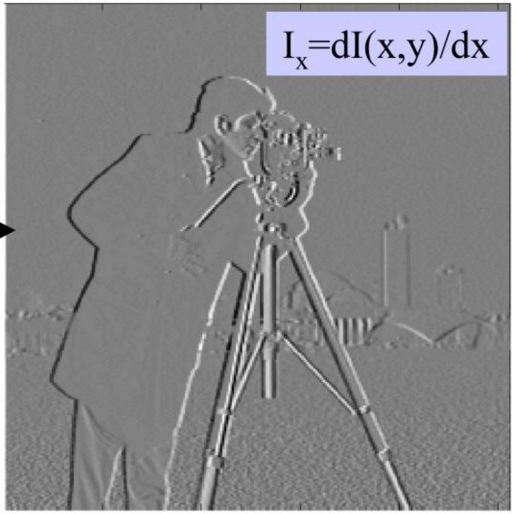
Example: Spatial Image Gradient



Example: Spatial Image Gradient

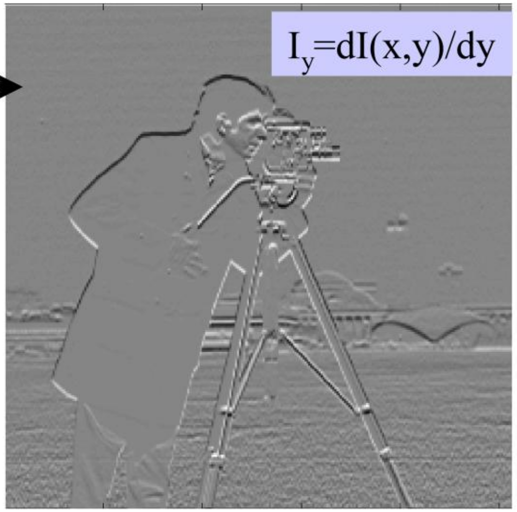
$$\frac{I(x+1,y) - I(x-1,y)}{2}$$

Partial derivative wrt x



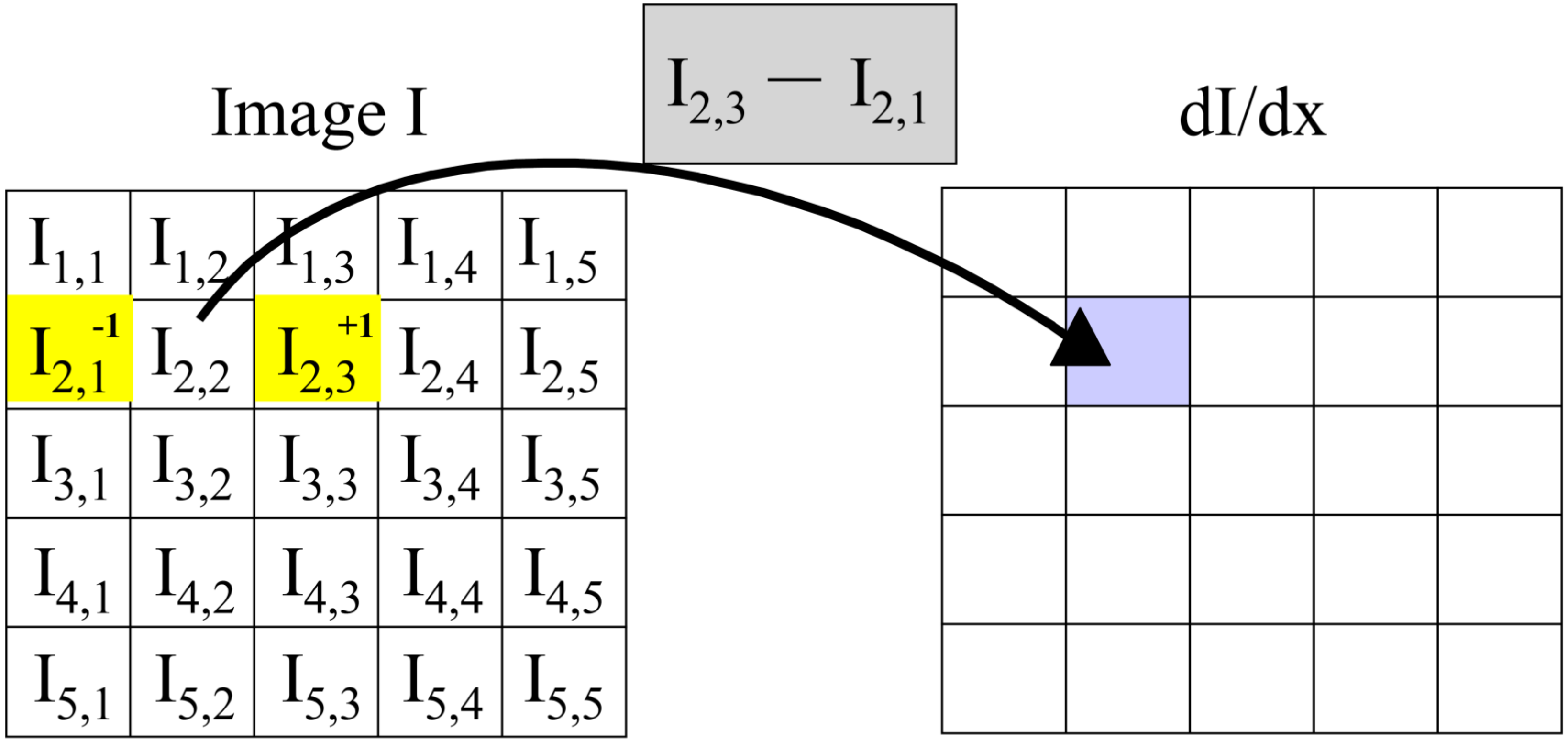
$$\frac{I(x,y+1) - I(x,y-1)}{2}$$

Partial derivative wrt y



Note: From now on we will drop the constant factor 1/2. We can divide by it later.

Example: Spatial Image Gradient



Example: Spatial Image Gradient

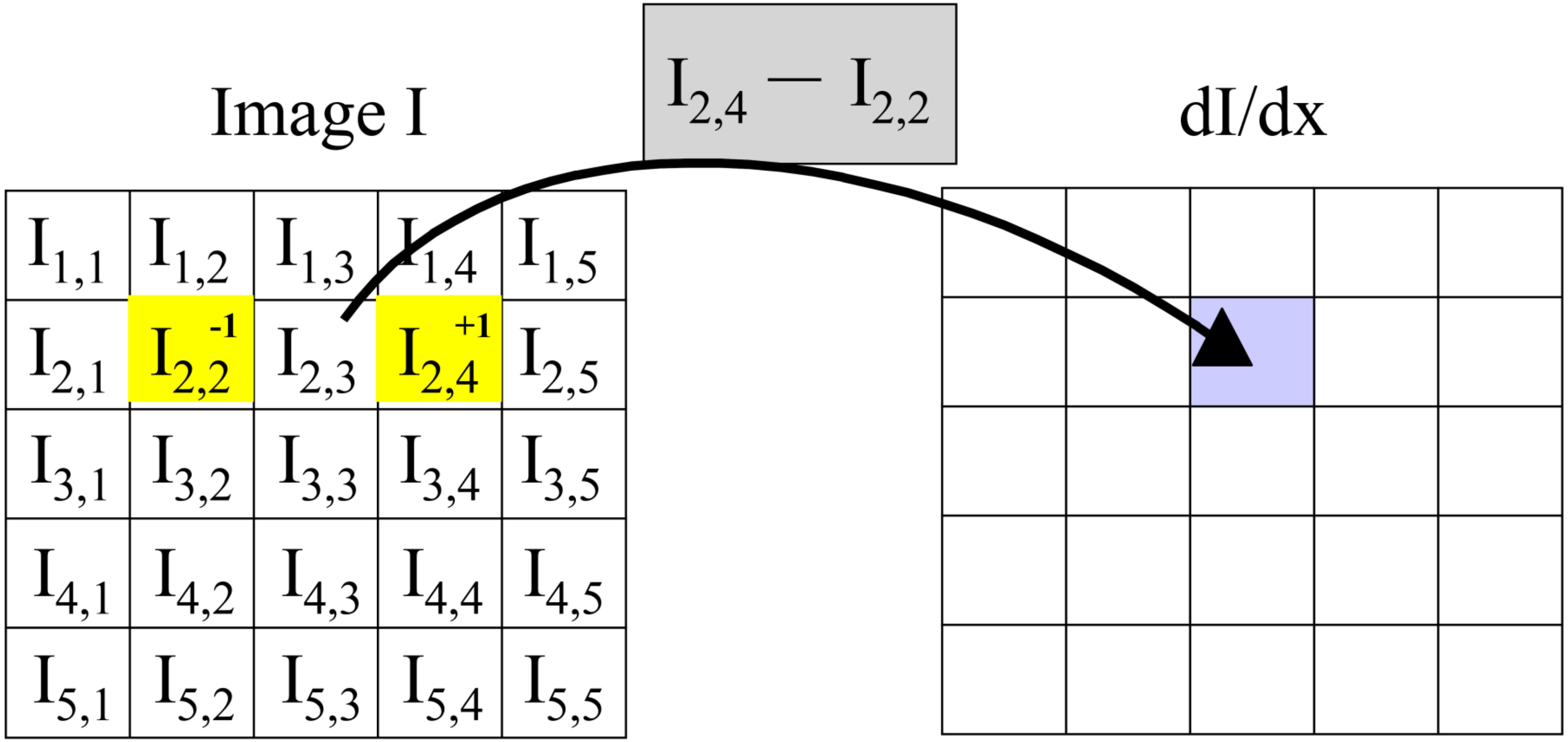
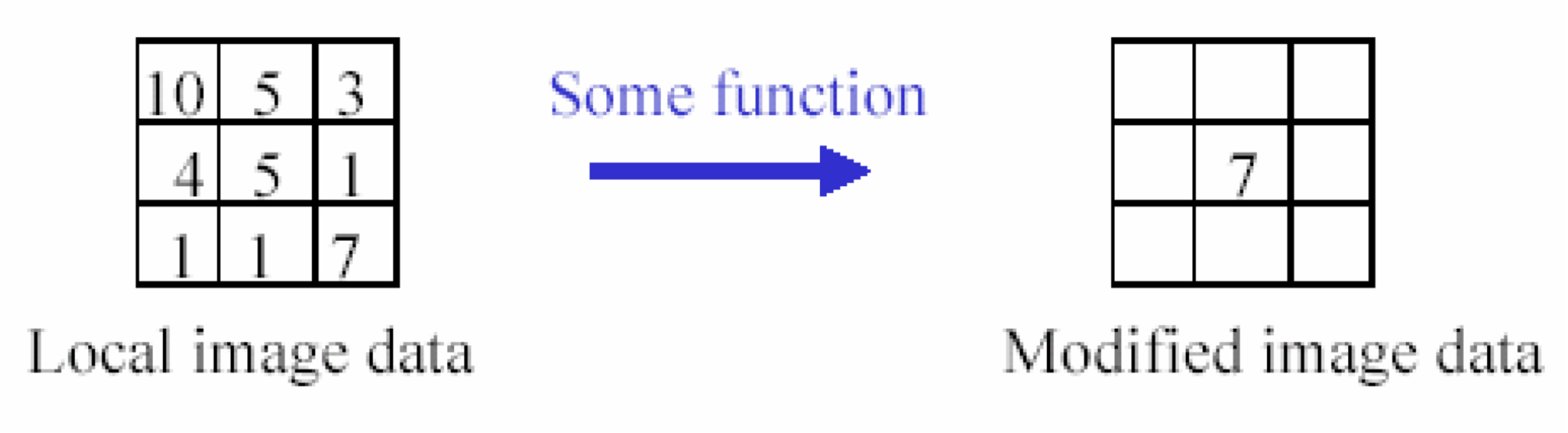


Image Filtering

- Modifying the pixels in an image based on some of function of a local neighborhood of pixels



Linear Filtering

- Using linear combination of the neighborhood of a pixels (weighted sum)

10	5	3
4	5	1
1	1	7

Local image data

0	0	0
0	0.5	0
0	1	0.5

kernel

	7	

Modified image data

$$10*0+5*0+3*0+4*0+5*.5+1*0+1*0+1*1+7*.5 = 7$$

Linear Filtering

$k \backslash l$	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

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

Kernel

$h(x,y)$

=

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

$g(x,y)$

Correlation $$g(i,j) = \sum_{k,l} f(i+k, j+l)h(k,l)$$

$$g = f \otimes h$$

Correlation vs. Convolution

$k \backslash l$	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

- Correlation $$g(i,j) = \sum_{k,l} f(i+k, j+l)h(k,l)$$

$$g = f \otimes h$$

What is the difference?

- Convolution $$g(i,j) = \sum_{k,l} f(i-k, j-l)h(k,l)$$

$$g = f * h$$

Filter flipped vertically and horizontally

$h(x,y)$

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

Properties of Convolution

Commutative

$$a \star b = b \star a$$

Associative

$$(((a \star b_1) \star b_2) \star b_3) = a \star (b_1 \star b_2 \star b_3)$$

Distributes over addition

$$a \star (b + c) = (a \star b) + (a \star c)$$

Scalars factor out

$$\lambda a \star b = a \star \lambda b = \lambda (a \star b)$$

Derivative Theorem of Convolution

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x} h) \star f$$

Convolution Example

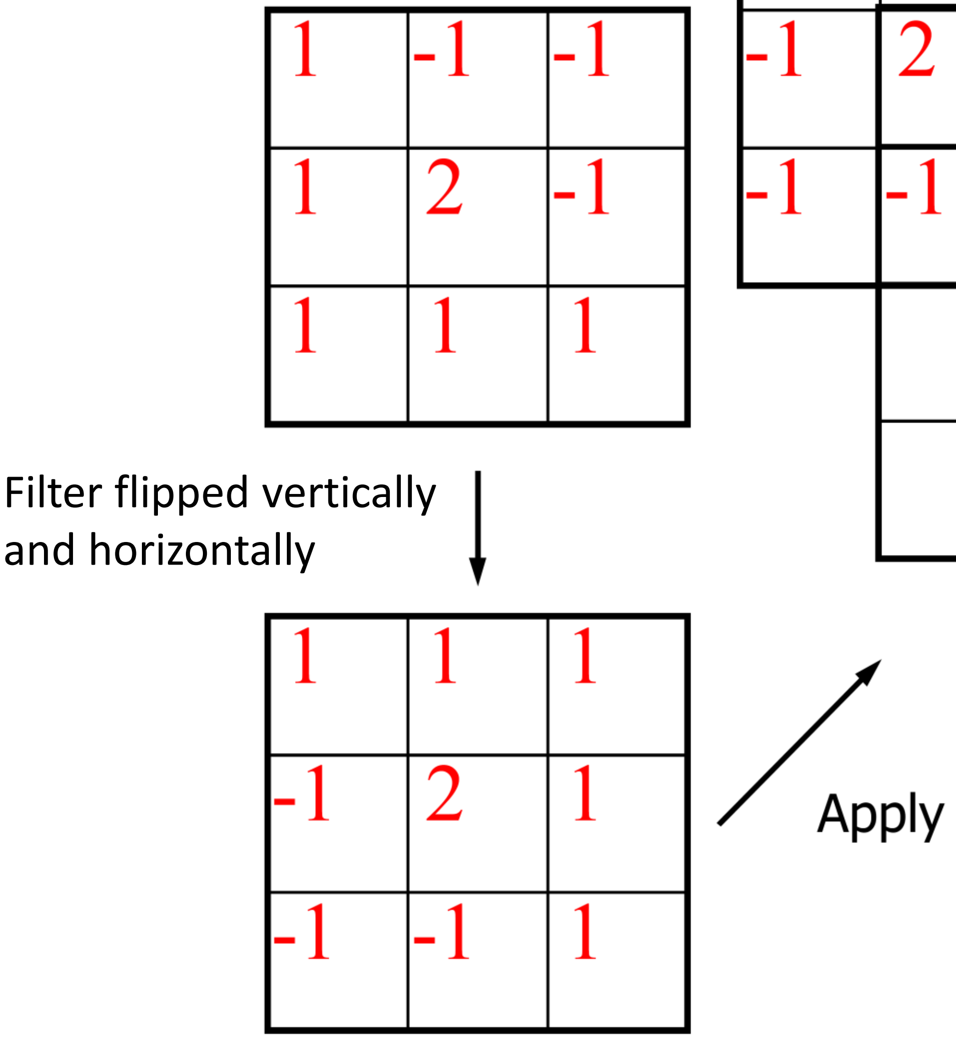
$$h = \begin{bmatrix} 1 & -1 & -1 \\ 1 & 2 & -1 \\ 1 & 1 & 1 \end{bmatrix} \qquad f = \begin{bmatrix} 2 & 2 & 2 & 3 \\ 2 & 1 & 3 & 3 \\ 2 & 2 & 1 & 2 \\ 1 & 3 & 2 & 2 \end{bmatrix} \qquad h * f = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

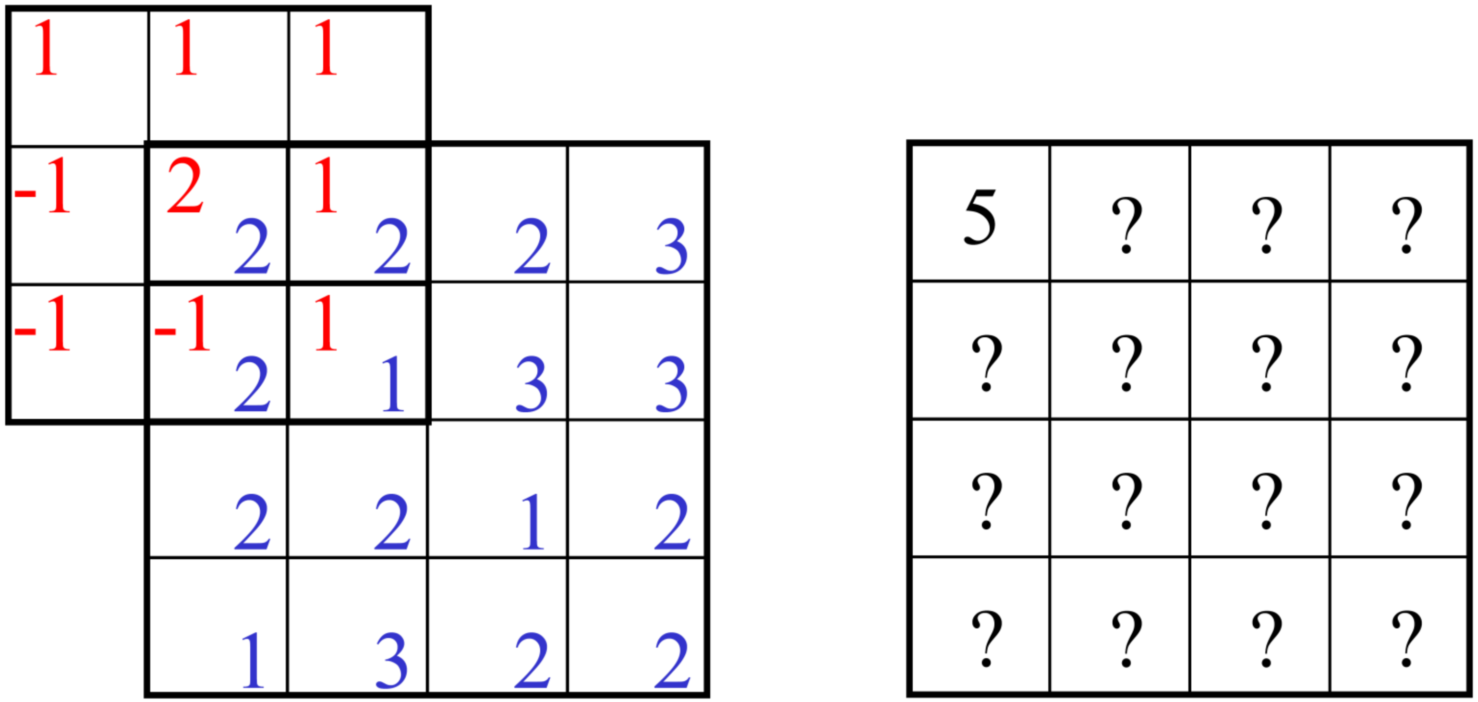
Filter flipped vertically and horizontally

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$$

Apply

adapted from C. Rasmussen, U. of Delaware


1 -1 -1
1 2 -1
1 1 1
Filter flipped vertically and horizontally
1 1 1
-1 2 1
-1 -1 1
Apply


5 ? ? ?
? ? ? ?
? ? ? ?
? ? ? ?

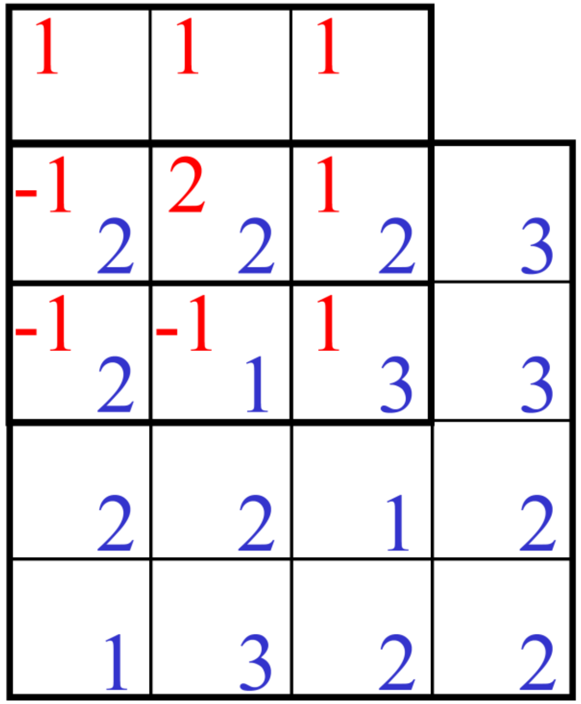
$$2*2+1*2+(-1)*2+1*1 = 5$$

1	-1	-1
1	2	-1
1	1	1

Filter flipped vertically
and horizontally

1	1	1
-1	2	1
-1	-1	1

Apply



5	4	?	?
?	?	?	?
?	?	?	?
?	?	?	?

$$2*2+1*2+(-1)*2+1*1 = 5$$

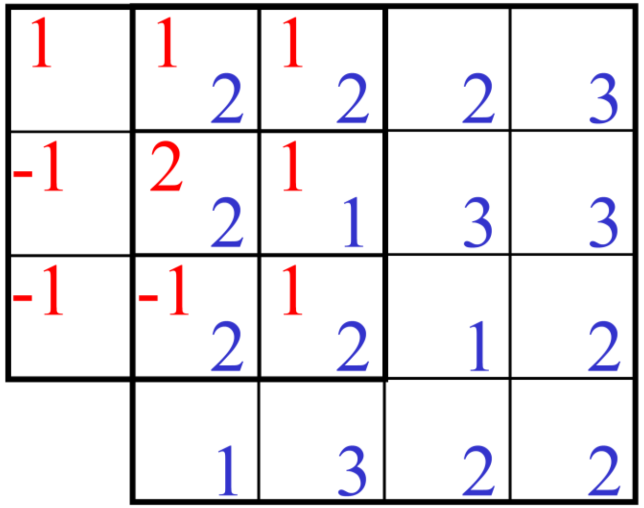
$$-1*2+2*2+1*2-1*2-1*1+1*3= 4$$

1	-1	-1
1	2	-1
1	1	1

Filter flipped vertically and horizontally

1	1	1
-1	2	1
-1	-1	1

Apply



5	4	4	-2
9	?	?	?
?	?	?	?
?	?	?	?

$2*2+1*2+(-1)*2+1*1 = 5$

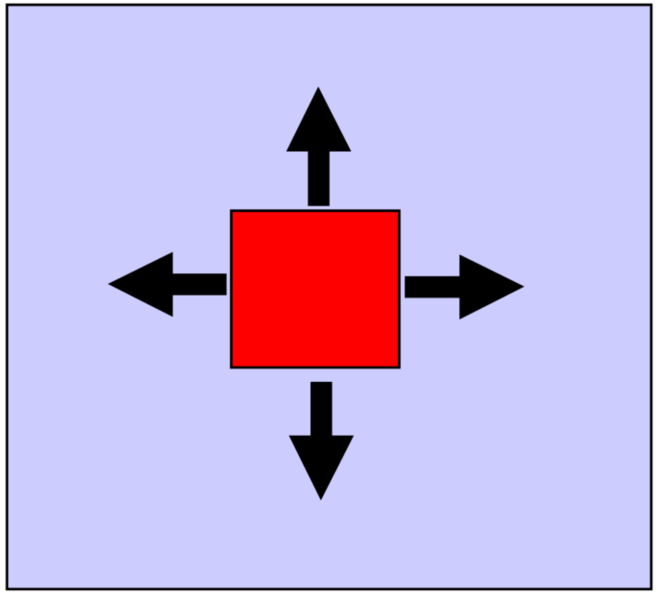
$-1*2+2*2+1*2-1*2-1*1+1*3= 4$

$-1*2+2*2+1*3-1*1-1*3+1*3= 4$

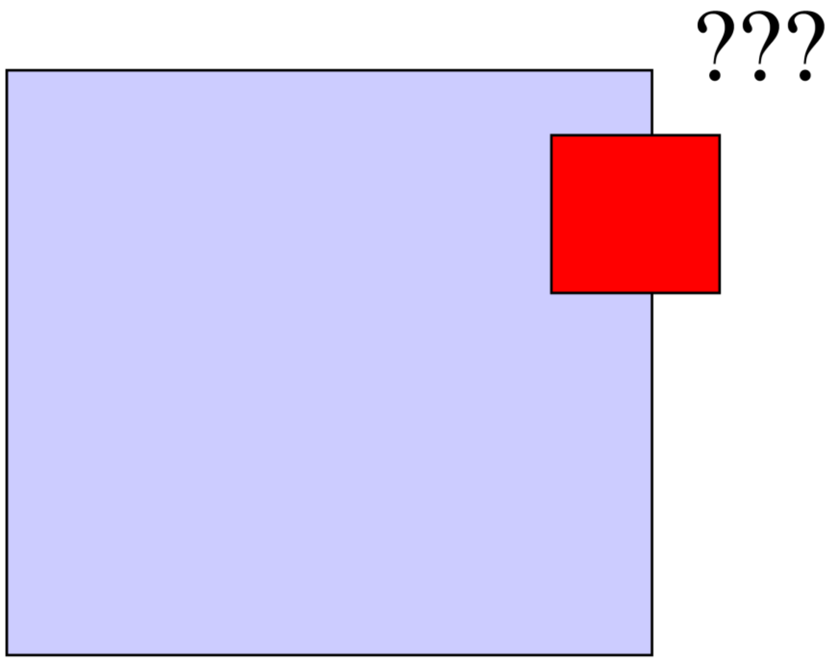
$-1*2+2*3-1*3-1*3 = -2$

$1*2+1*2+2*2+1*1-1*2+1*2= 9$

Border Handling



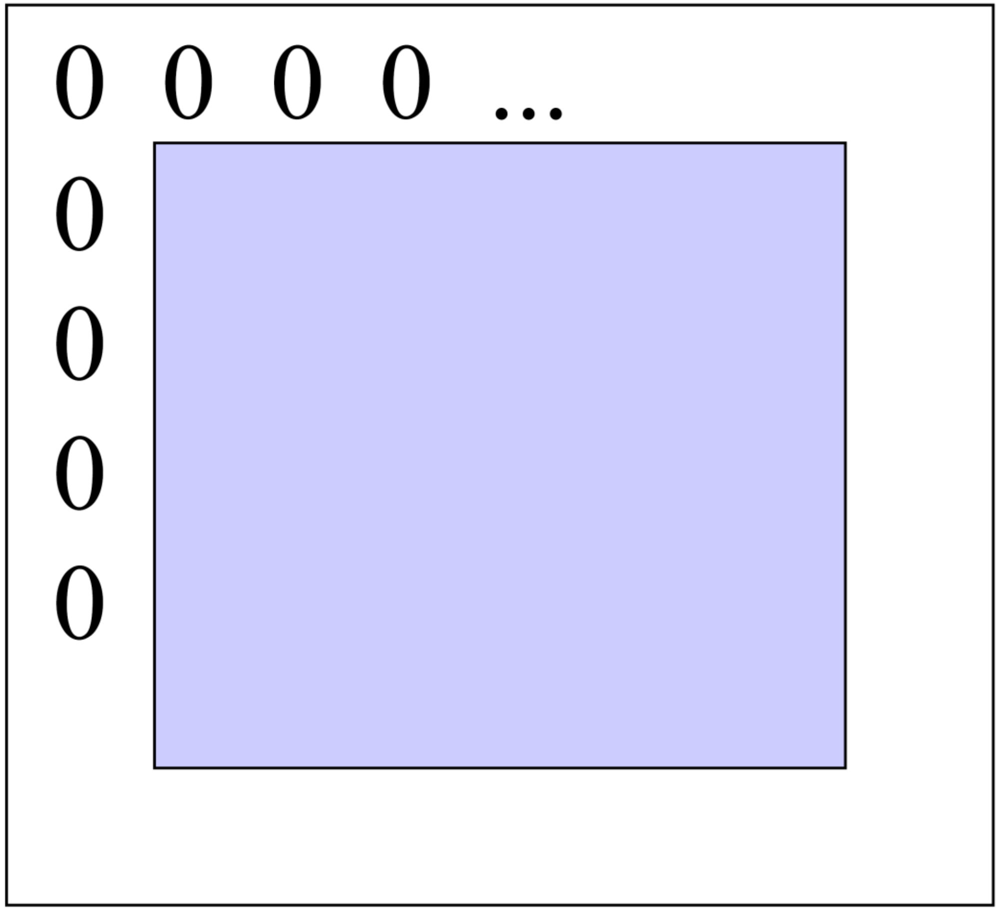
for interior pixels where there is full overlap, we know what to do.



but what values do we use for pixels that are “off the image” ?

Border Handling

- Zero padding



Border Handling

- Replicatior

1	2	3
4	5	6
7	8	9

			1	2	3			
	1s		1	2	3		3s	
			1	2	3			
1	1	1	1	2	3	3	3	3
4	4	4	4	5	6	6	6	6
7	7	7	7	8	9	9	9	9
			7	8	9			
	7s		7	8	9		9s	
			7	8	9			

Border Handling

- Reflection

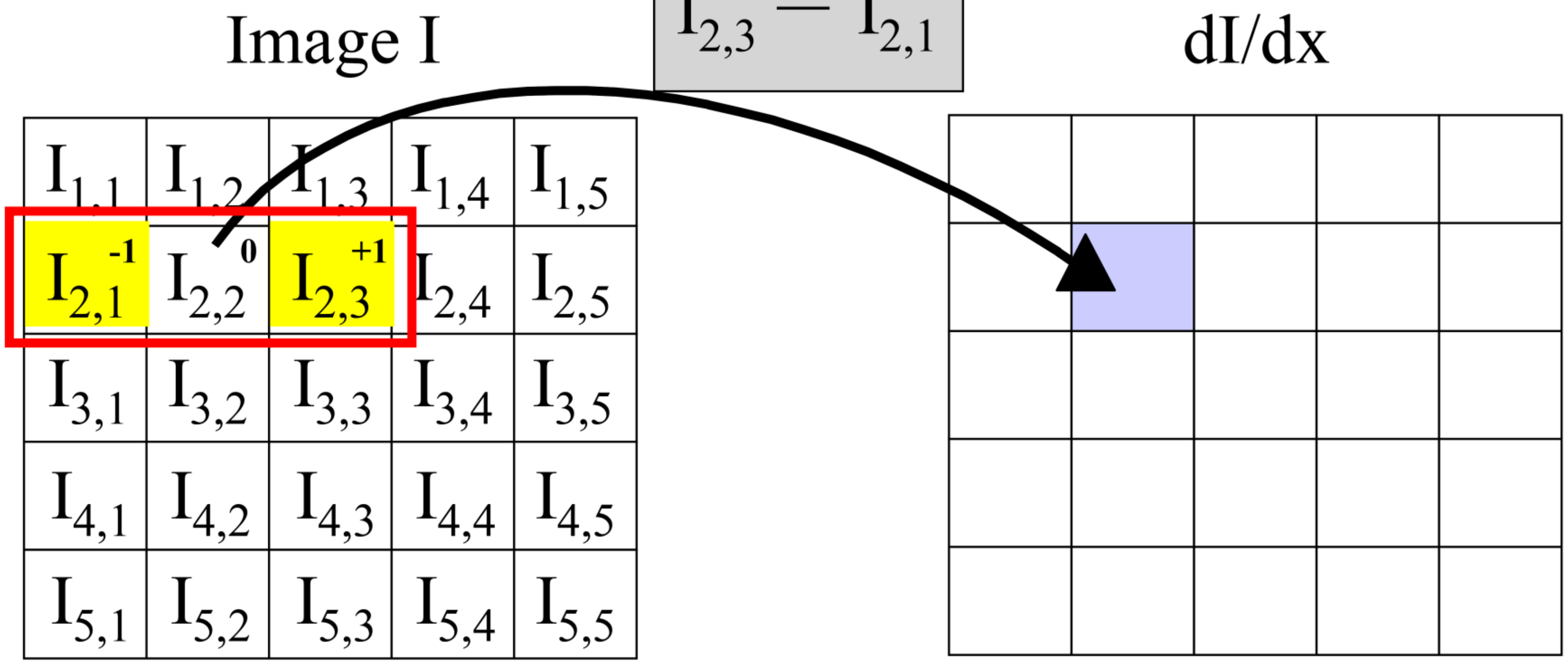
1	2	3
4	5	6
7	8	9

9	8	7	7	8	9	9	8	7
6	5	4	4	5	6	6	5	4
3	2	1	1	2	3	3	2	1
3	2	1	1	2	3	3	2	1
6	5	4	4	5	6	6	5	4
9	8	7	7	8	9	9	8	7
9	8	7	7	8	9	9	8	7
6	5	4	4	5	6	6	5	4
3	2	1	1	2	3	3	2	1

Back to Image Gradient

$$(-1) * I_{2,1} + (0) * I_{2,2} + (+1) * I_{2,3}$$

$$I_{2,3} - I_{2,1}$$

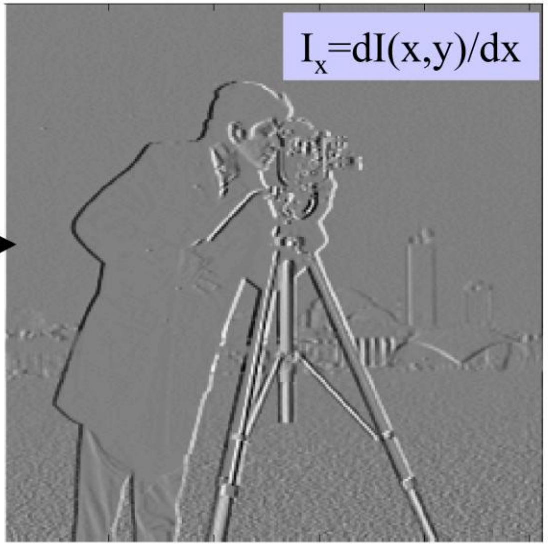


What filter is this?

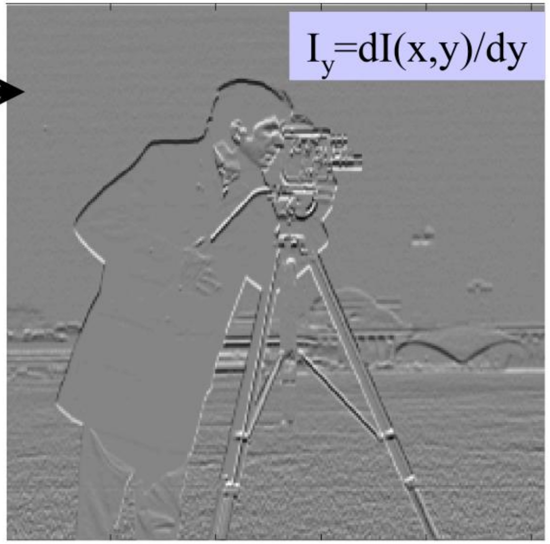
Back to Image Gradient

$$I_x = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} * I$$

Partial derivative wrt x



Partial derivative wrt y



$$I_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * I$$

Note that there
is a difference between
convolving with a 1xn row
filter and an nx1 col filter.

Further Reading

- Chapter 3.2, Richard Szeliski

- Carlo Tomasi, Image Correlation, Convolution and Filtering, https://courses.cs.duke.edu/fall15/cps274/notes/convolution-filtering.pdf