Filtering vs Convolution

16-385 Computer Vision

Filters we have learned so far …

The 'Box' filter

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

Gaussian filter

$$\frac{1}{16}$$

1	2	1
2	4	2
1	2	1

Sobel filter

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

Laplace filter

0	1	0
1	-4	1
0	1	0

Filtering vs Convolution

filtering
(cross-correlation)

$h = g \otimes f$

output filter image

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

What's the difference?

convolution

$h = g \star f$

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m-k, n-1]$$

Credit: Steve Seitz

Filtering vs Convolution

filtering
(cross-correlation)

$$h = g \otimes f$$

output — filter — image

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

filter flipped vertically and horizontally

convolution

$$h = g \star f$$

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m-k, n-1]$$

Filtering vs Convolution

output　　filter　　image

filtering
(cross-correlation)

$h = g \otimes f$

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

filter flipped vertically and horizontally

convolution

$h = g \star f$

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m-k, n-1]$$

Suppose g is a Gaussian filter.
How does convolution differ from filtering?

Recall...

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

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

can precompute this

Derivative Theorem of Convolution

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

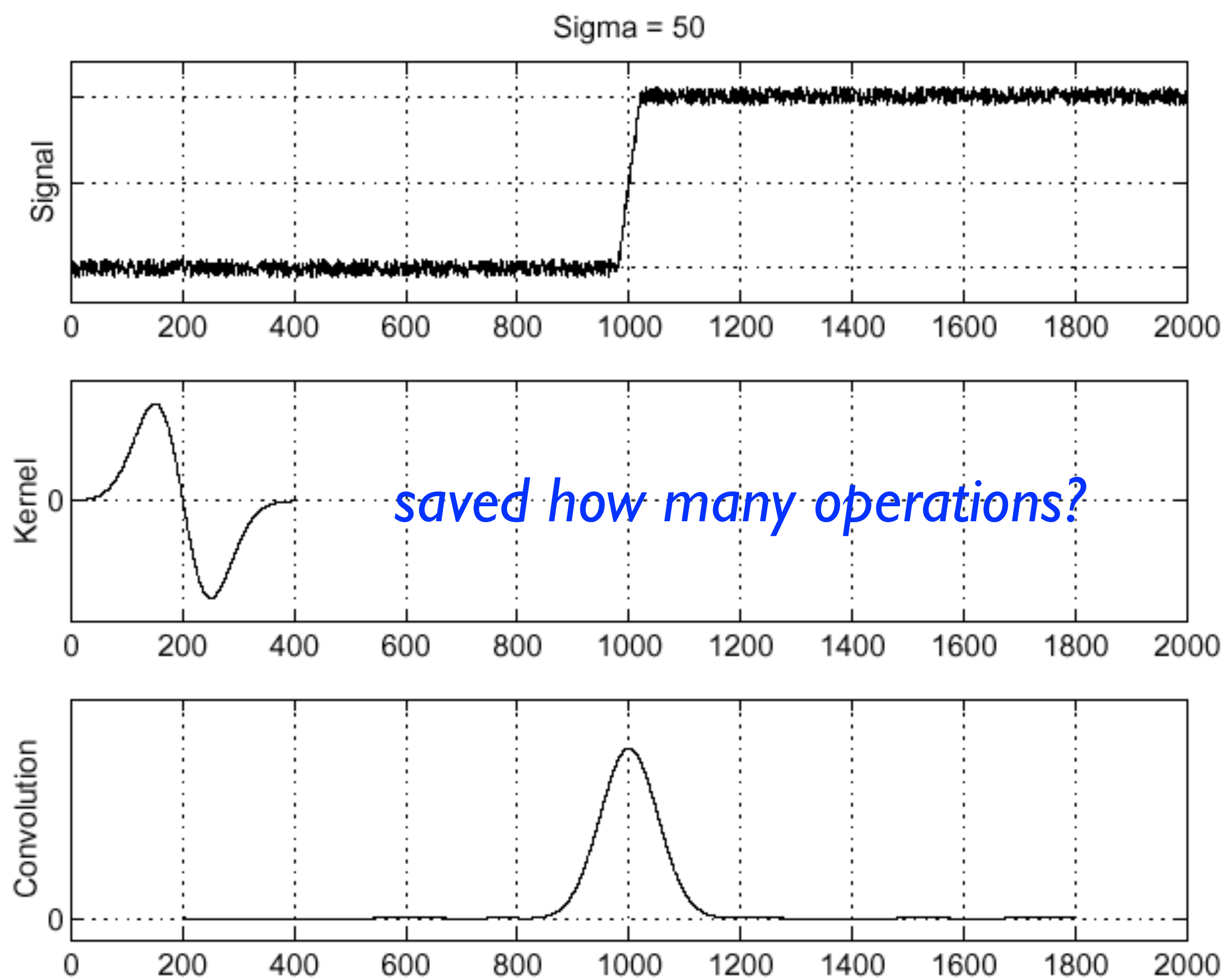
Input

Derivative of Gaussian

$\frac{\partial}{\partial x}h$

Output

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



Recall ...

Input

Gaussian

Smoothed input

Derivative

Output