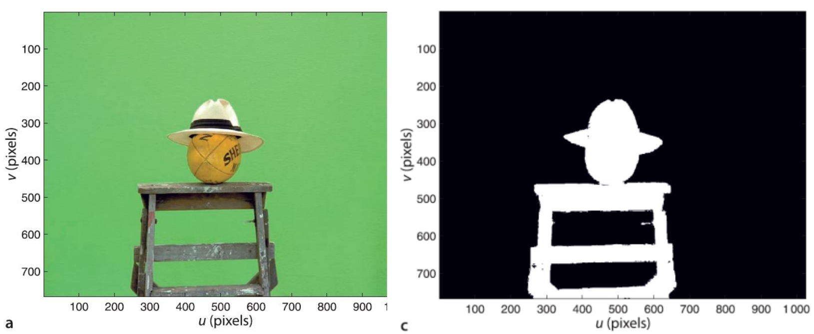
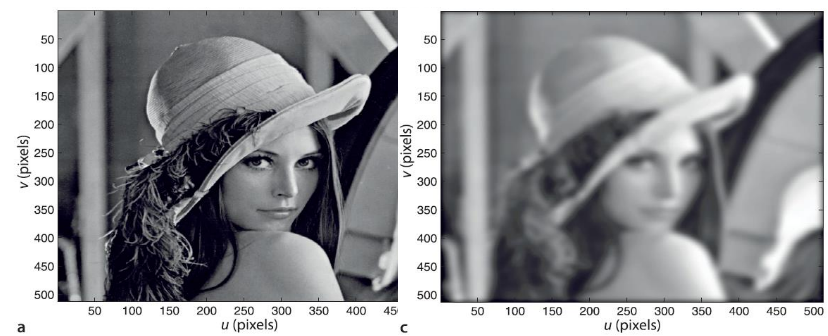
Image operations

- (Digital) images are discrete (both in time, space and information)
  - Processing is performed for each pixel
- The whole image is not processed (or available) at the same time (even though it might seem so)
  - It can be thought of as sliding over the image with a operational window (or looking glass)



Segmentation

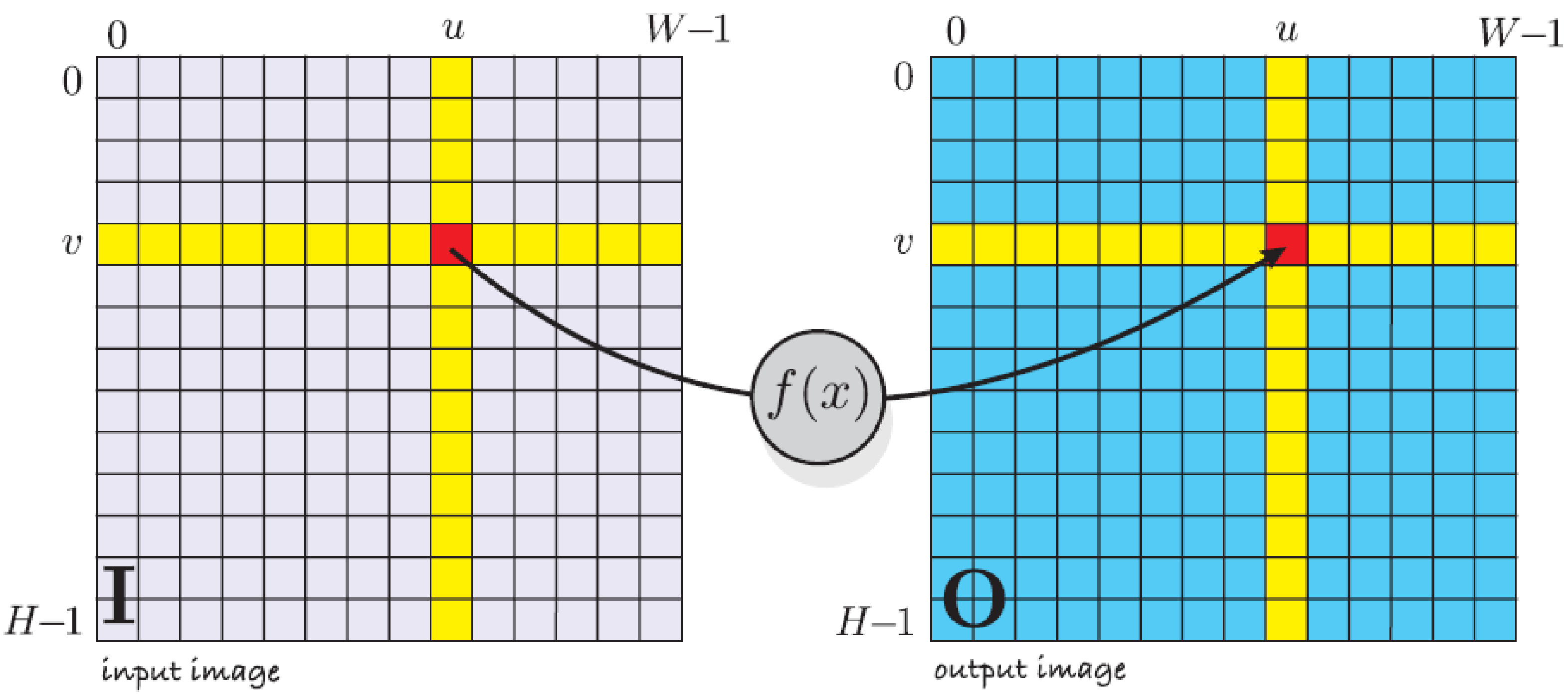


Smoothing

Monadic operators

- Output pixel is a function of the corresponding input pixel

$$\boldsymbol{O}[u,v] = f\big(\boldsymbol{I}[u,v]\big), \quad \forall (u,v) \in \boldsymbol{I}$$

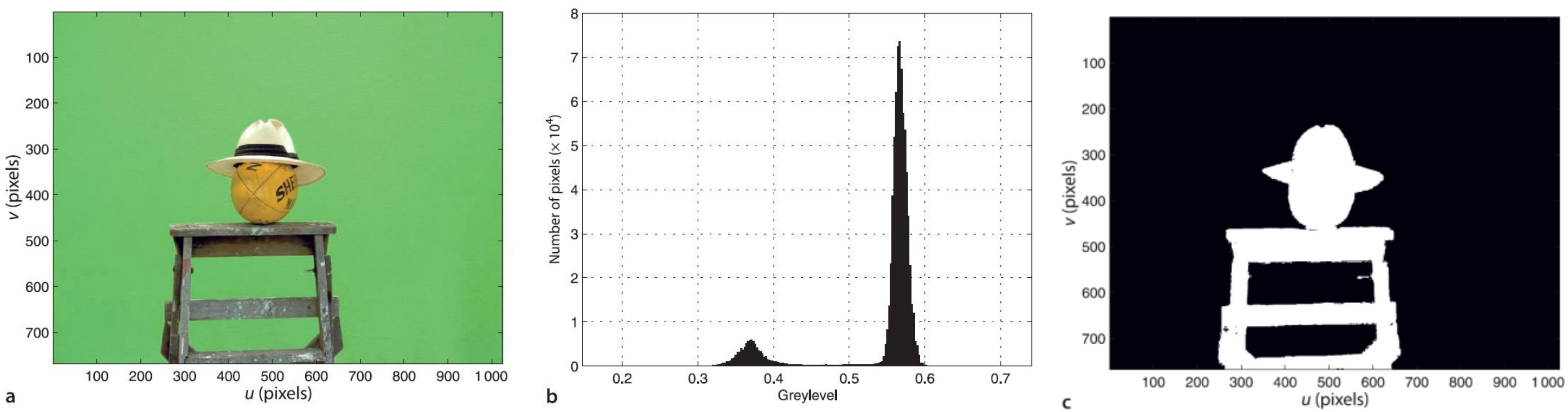


Monadic operators - Segmentation

- Single out single component (color)
  - Simple if distinct
  - Use the histogram to find threshold (T)
- Background removal
- Blob (object) extraction

$$f(x) = \begin{cases} 1 & x > T \\ 0 & otherwise \end{cases}$$

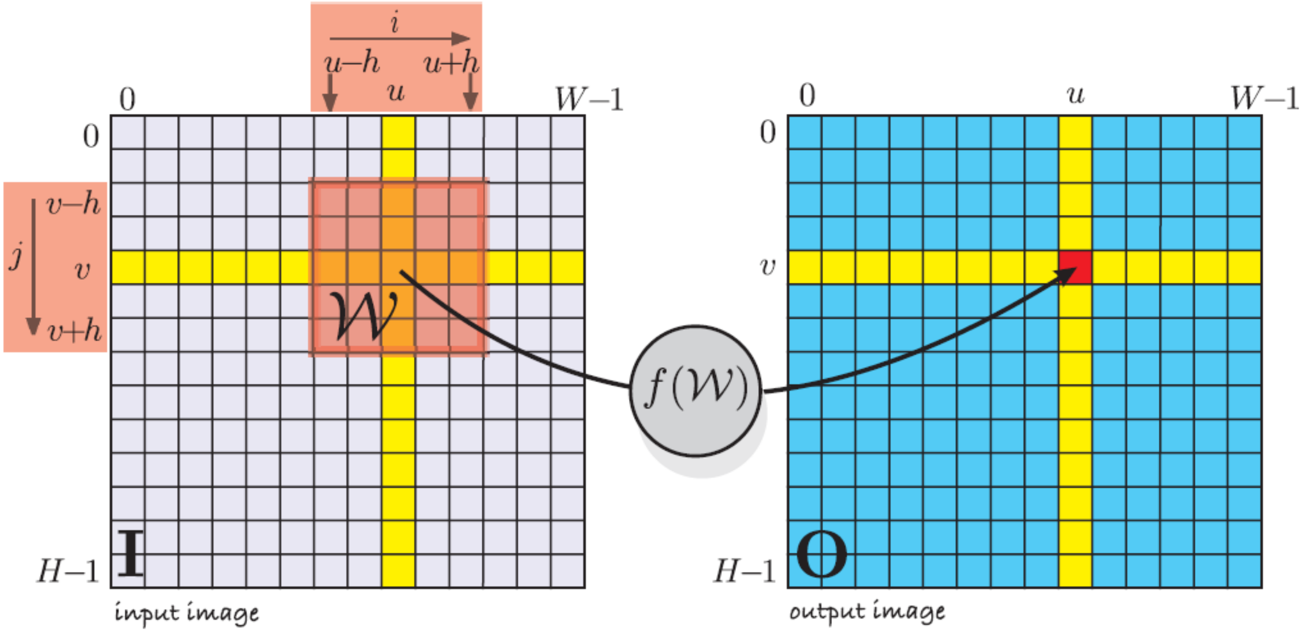
Assuming a color depth of 8-bits, what is the representation of red?



Spatial Operators

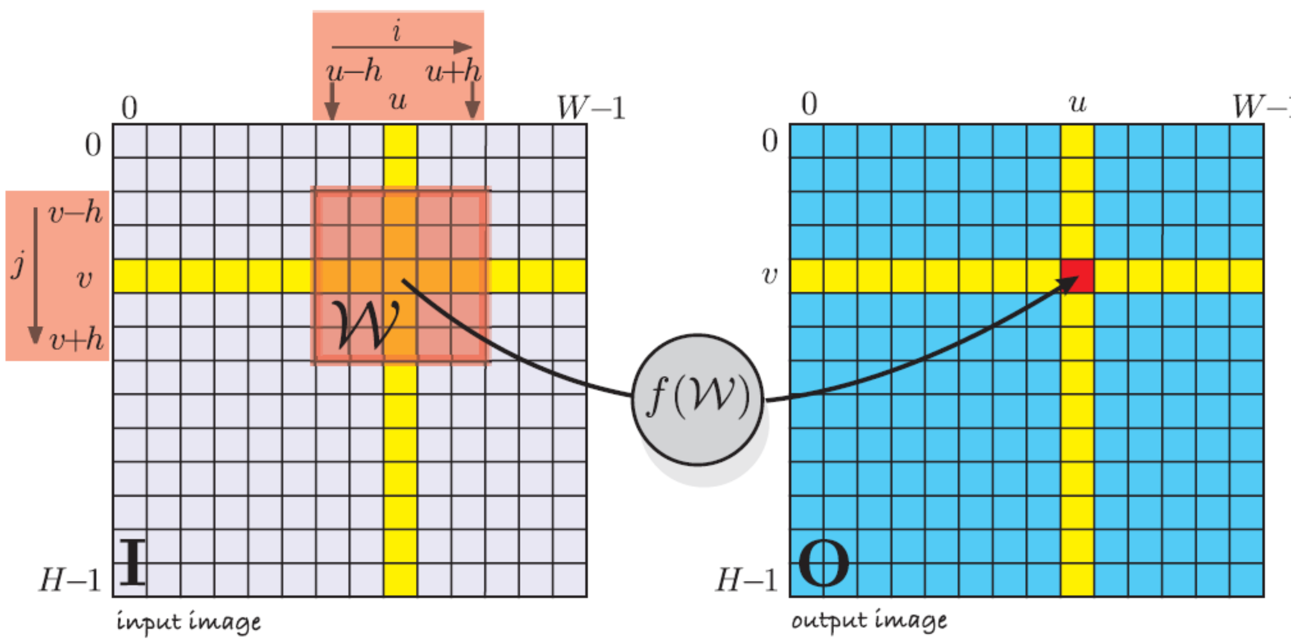
- Each pixel in the output image is a function of all pixels in a region surrounding the corresponding pixel in the input image.

$$\boldsymbol{O}[u,v] = f\left(\boldsymbol{I}[u+i, v+j]\right), \quad \forall (i,j) \in \mathcal{W}, \quad \forall (u,v) \in \boldsymbol{I}$$



Spatial Operators - Region

- Note that the region has odd side length (h=2)
- Image boundaries, W does not fit?
- Extensively used – smoothing, feature detection, filtering, matching, …



Spatial Operator Example - Filtering

Lowpass filtered image

Highpass filtered image

© R. Siegwart , D. Scaramuzza and M.Chli, ETH Zurich - ASL

Convolution

- 1D convolution from signal processing:

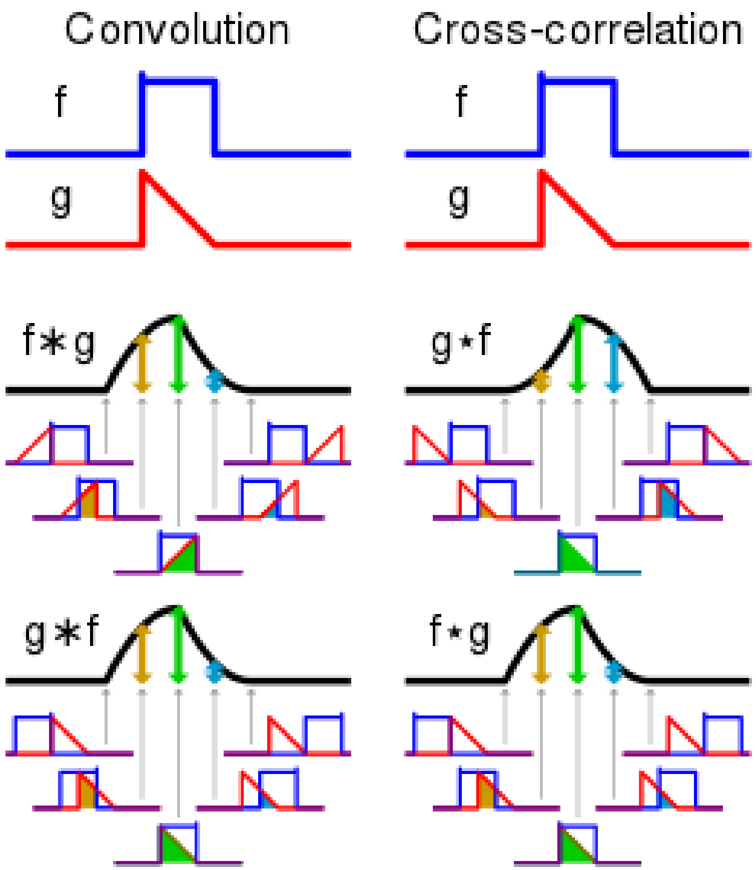
$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau$$

- Digital images are discrete

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

- Convolution vs Cross-correlation
  - Confusion: `conv2`, `imfilter`

- However, images are 2D

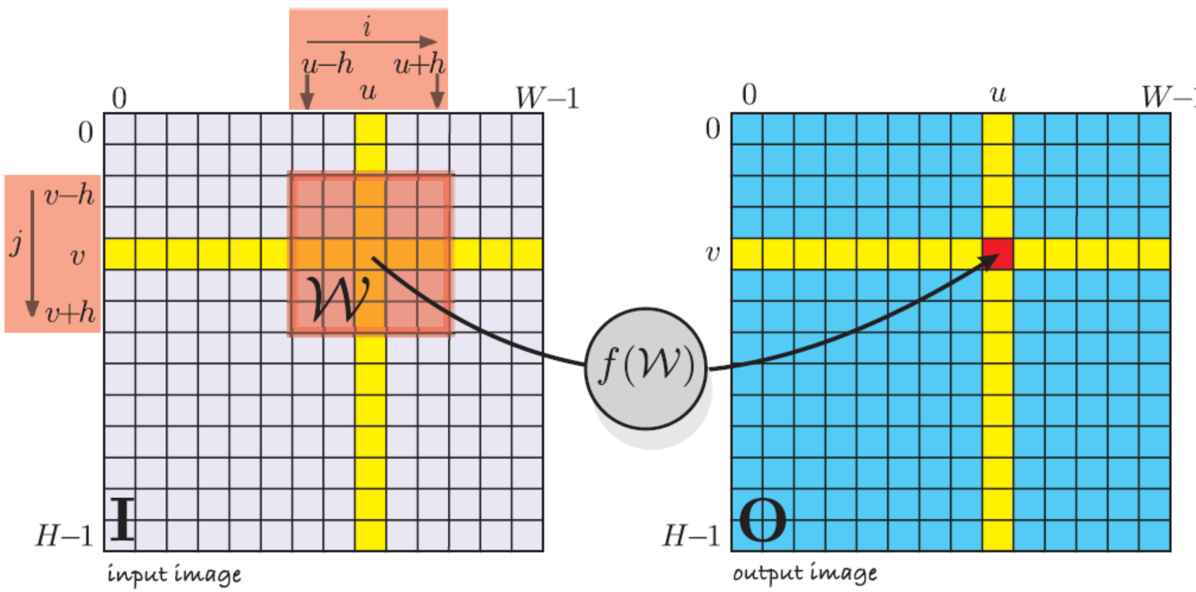


Convolution

$$\mathbf{O} = \boldsymbol{K} \otimes \mathbf{I}$$

$$\mathbf{O}[u,v] = \sum_{(i,j)\in\mathcal{W}} \boldsymbol{I}[u+i, v+j]\boldsymbol{K}[i,j], \quad \forall(u,v) \in \mathbf{I}$$

$\boldsymbol{K} \in \mathbb{R}^{w\times w}$ is the convolution kernel



- A window of pixels, from the input image, are multiplied element-wise with the kernel, to generate a single output pixel => loop over all pixels

Convolution

- A linear spatial operator
- The **kernel** is also referred to as a **filter**
- Use kernel of **odd** width and height
- Requires padding to produce same size output

- Examples:
  - Smoothing: Kernel of unit volume (sums to 1).

$$K = ones(N, N) \cdot \frac{1}{N^2}$$ ***Mean of the N by N neighborhood***

  - edges: Kernel weights sum to 0

- Convolution: A weighted sum of pixels where the weights and size are defined by the kernel.

Convolution Example

K

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

Rotate

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

Apply

I

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

$\mathbf{O} = K \otimes \mathbf{I}$

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

What is the value of O(0,0)? Consider pixels outside the image as 0.

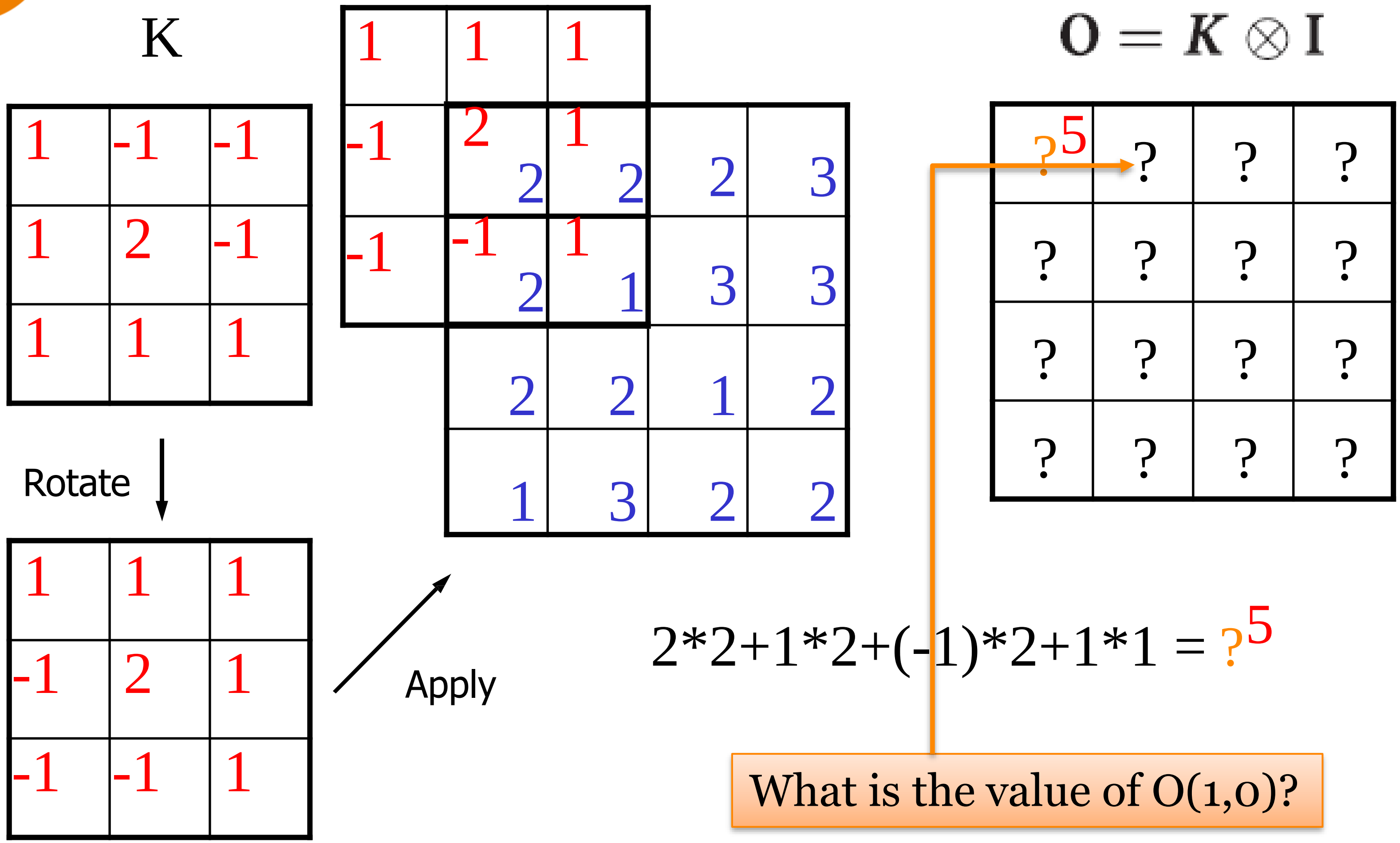
Remember:

$$\mathbf{O}[u,v] = \sum_{(i,j)\in\mathcal{W}} \boldsymbol{I}[u+i, v+j]\boldsymbol{K}[i,j], \quad \forall (u,v) \in \mathbf{I}$$

What is u, v?
What is i, j?

adapted from C. Rasmussen, U. of Delaware

Convolution Example



Robert Collins
CSE486, Penn State

Convolution Example

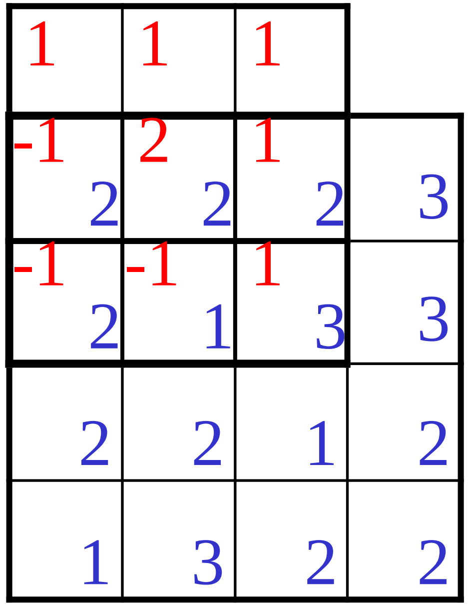
K

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

Rotate

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

Apply



$$\mathbf{O} = \boldsymbol{K} \otimes \mathbf{I}$$

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

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

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

Robert Collins
CSE486, Penn State

Convolution Example

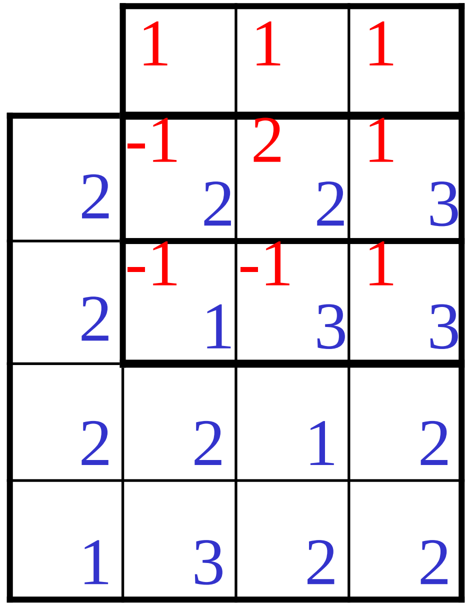
K

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

Rotate

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

Apply



$$\mathbf{O} = K \otimes \mathbf{I}$$

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

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

Robert Collins
CSE486, Penn State

Convolution Example

$$\mathbf{O} = K \otimes \mathbf{I}$$

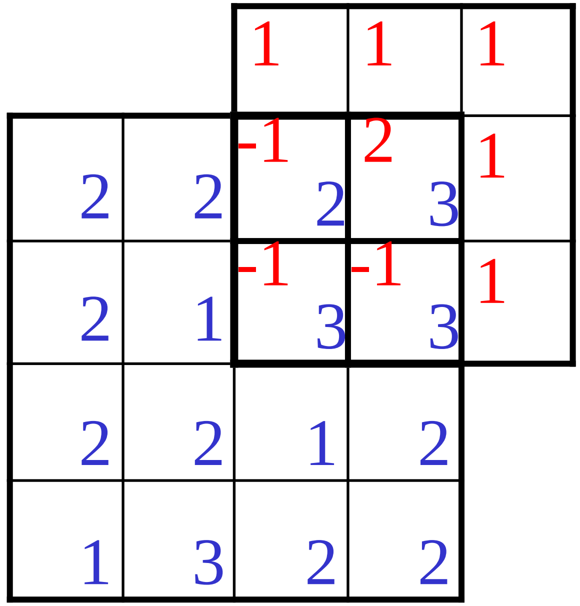
K

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

Rotate

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

Apply



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

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

Robert Collins
CSE486, Penn State

Convolution Example

K

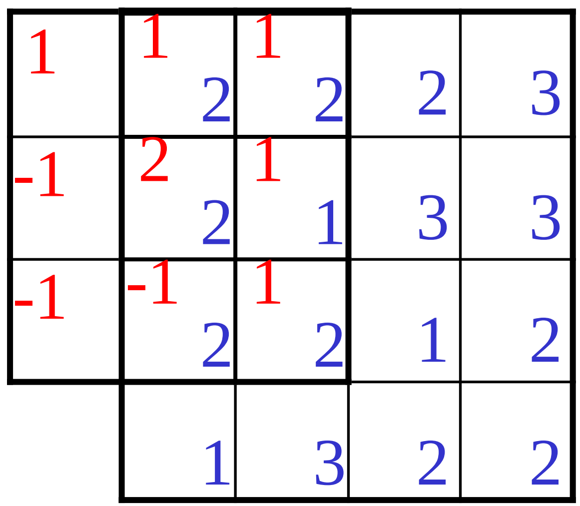
1	-1	-1
1	2	-1
1	1	1

Rotate

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

Apply

I



$\mathbf{O} = \boldsymbol{K} \otimes \mathbf{I}$

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

Robert Collins
CSE486, Penn State

Convolution Example

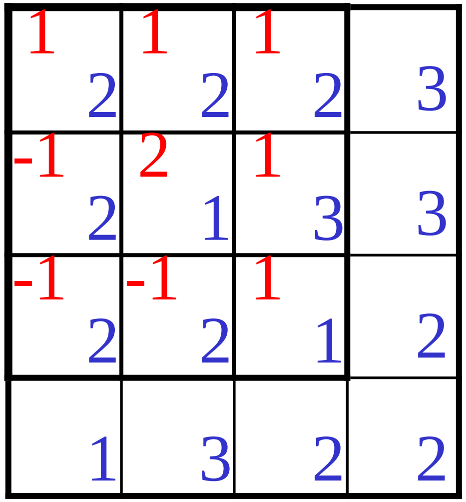
K

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

Rotate

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

Apply



$$\mathbf{O} = K \otimes \mathbf{I}$$

5	4	4	-2
9	6	?	?
?	?	?	?
?	?	?	?

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

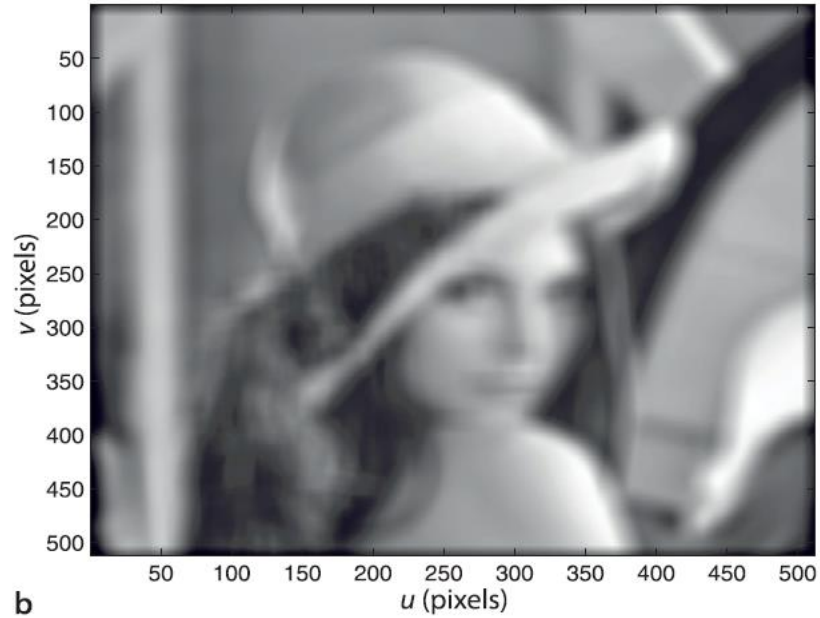
1*2+1*2+1*2-1*2+2*1+1*3-1*2-1*2+1*1= 6

Robert Collins
CSE486, Penn State

Convolution – Smoothing

Mean of neighborhood

a



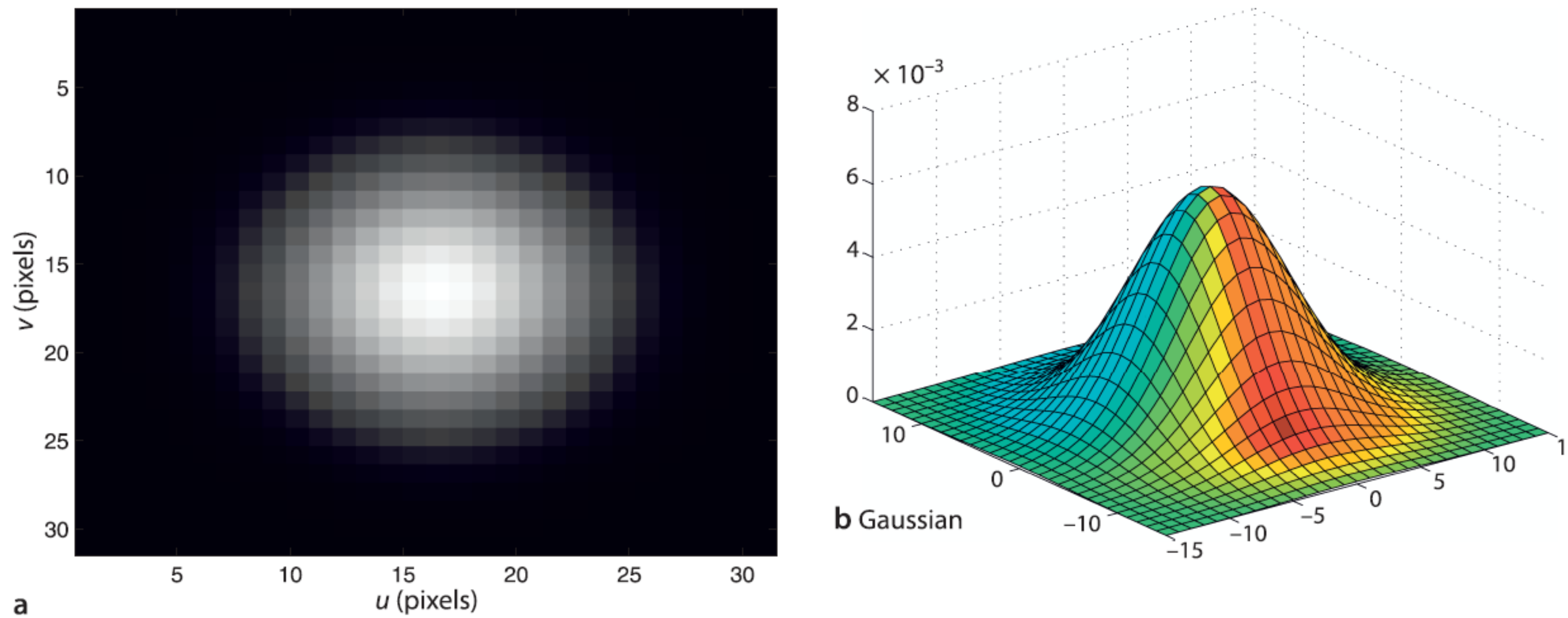
b

$$K = ones(N, N) \cdot \frac{1}{N^2}$$

1	1	1
1	1	1
1	1	1

$\frac{1}{9}$

Convolution maps – Gaussian kernel

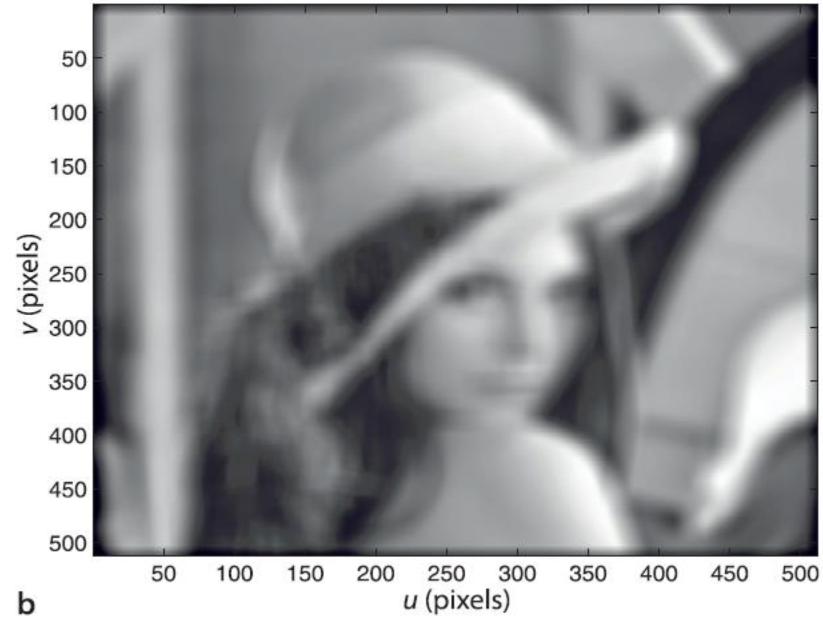


$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

The spread of the Gaussian function is controlled by the <u>standard deviation σ</u>

Convolution – Smoothing

Mean of neighborhood



Gaussian Kernel

Better smoothing: Central pixels weight more !!

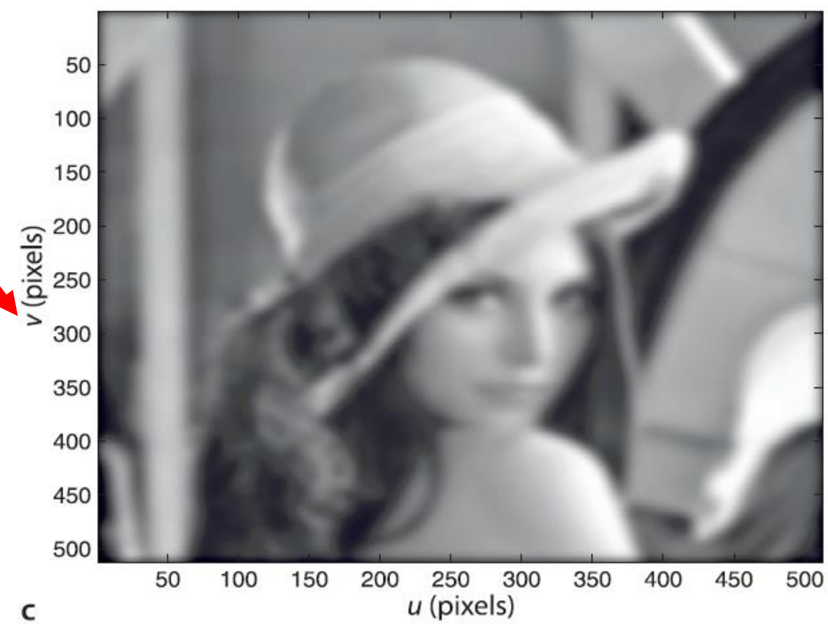


Image filtering | examples

- What does blurring take away?

original image

-

smoothed (5x5)

=

detail

Image filtering | examples

- Let's add it back:

original image

+ a

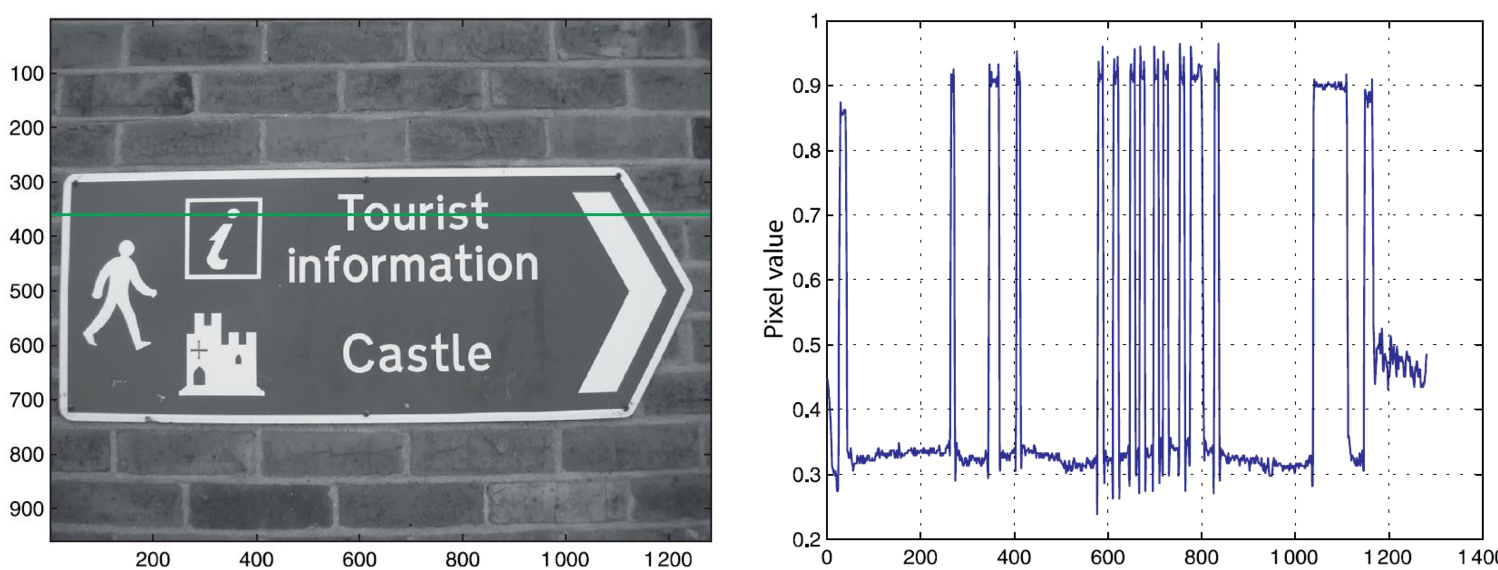
detail

=

sharpened

Edges

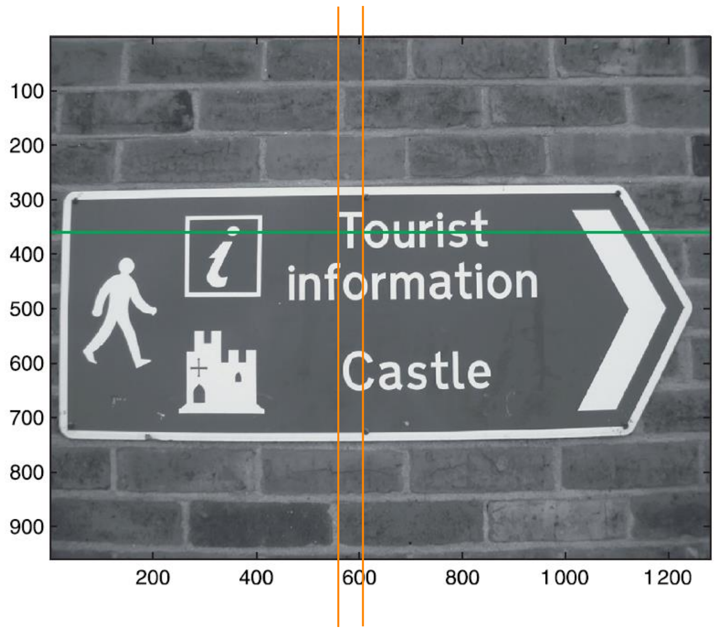
- Edges are useful characteristics of an image. Why?
  - They denote significant change; boundaries.
- Edge intensity profile

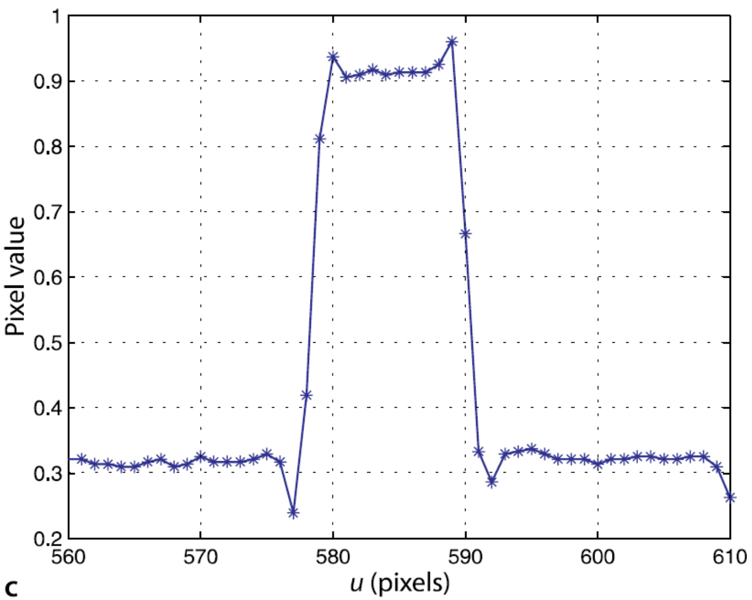


b

Edges

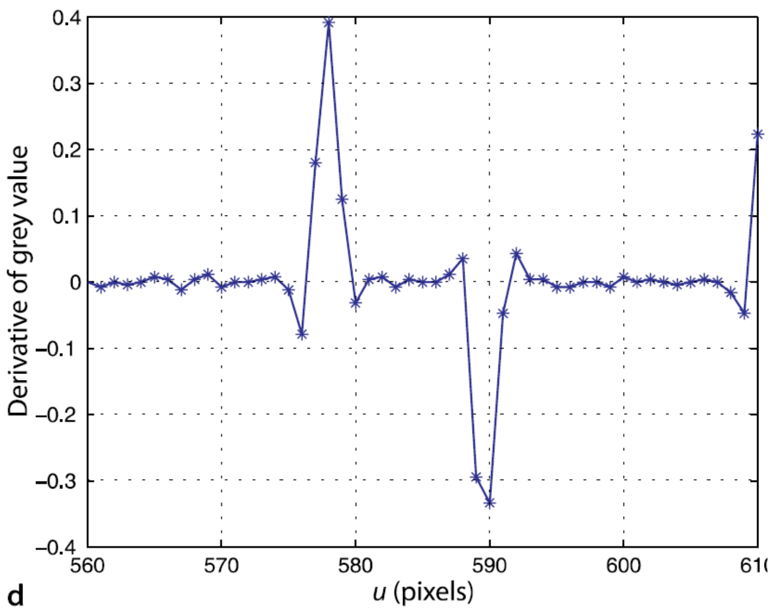
- Edges are useful characteristics of an image. Why?
  - They denote significant change; boundaries.
- Edge intensity profile
- Investigation on the T shows a clear edge
  - Or actually two edges

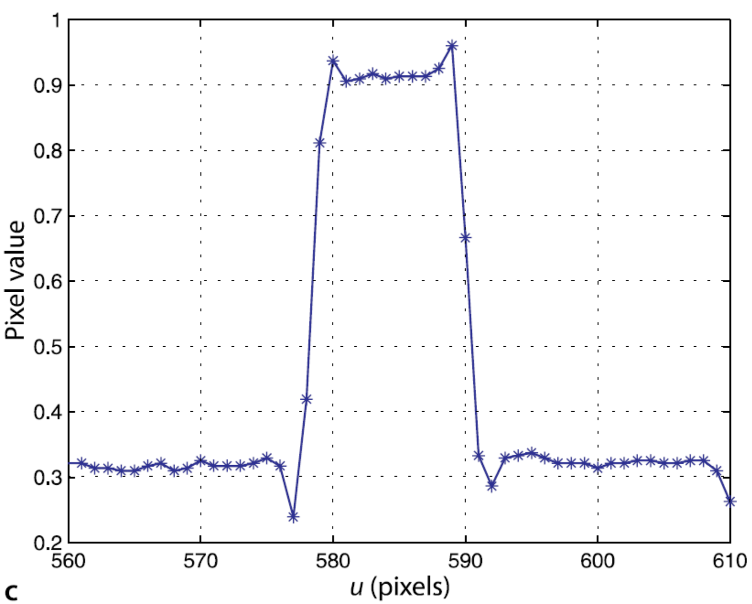




Edges

- Edges are useful characteristics of an image. Why?
  - They denote significant change; boundaries.
- Edge intensity profile
- Investigation on the T shows a clear edge
  - Or actually two edges
  - Which can be detected by… the derivative

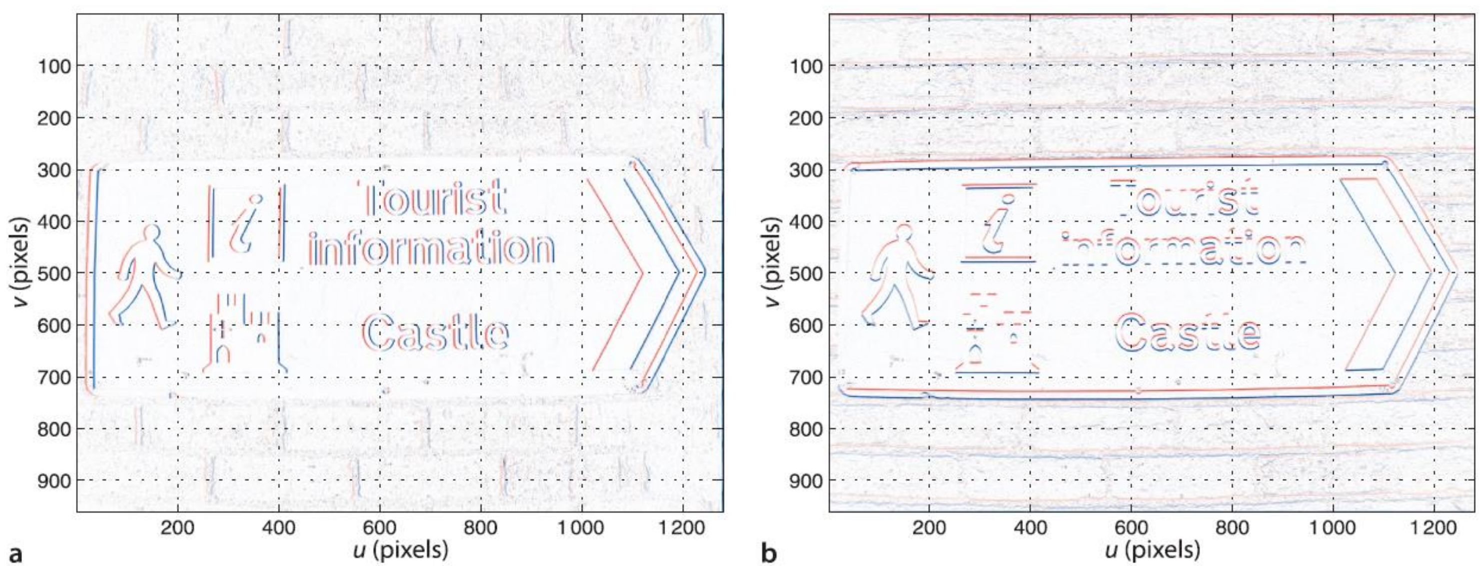




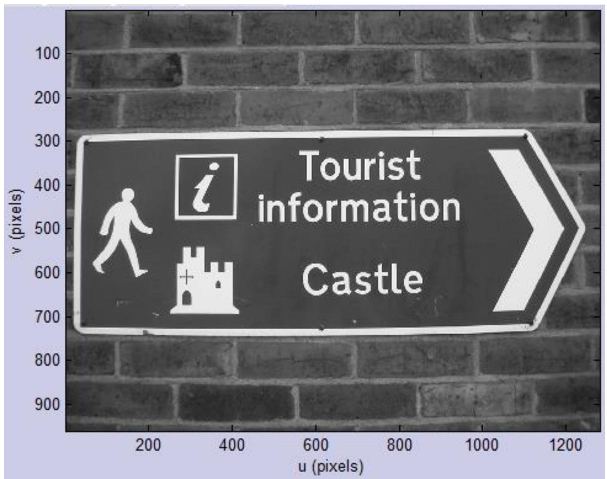
Derivative (1st order difference)

- The derivative is a very good indicator of an edge
- Convolving with a 1-directional kernel highlights changes in one direction (perpendicular edges)
- Use two separate kernels and combine the output

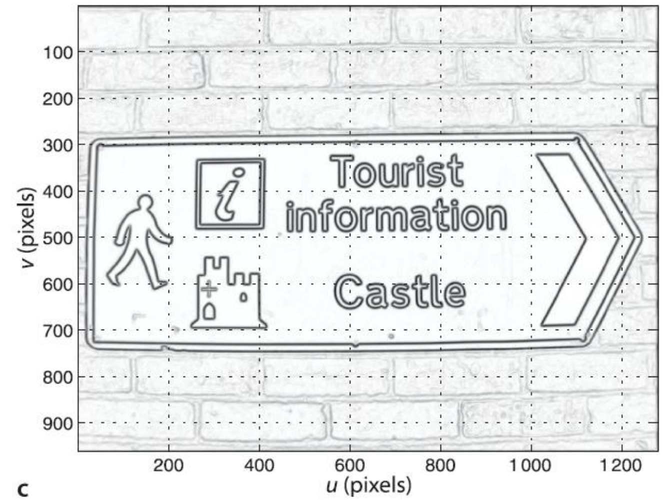
$$I_u = \frac{\partial I}{\partial u} \qquad I_v = \frac{\partial I}{\partial v}$$



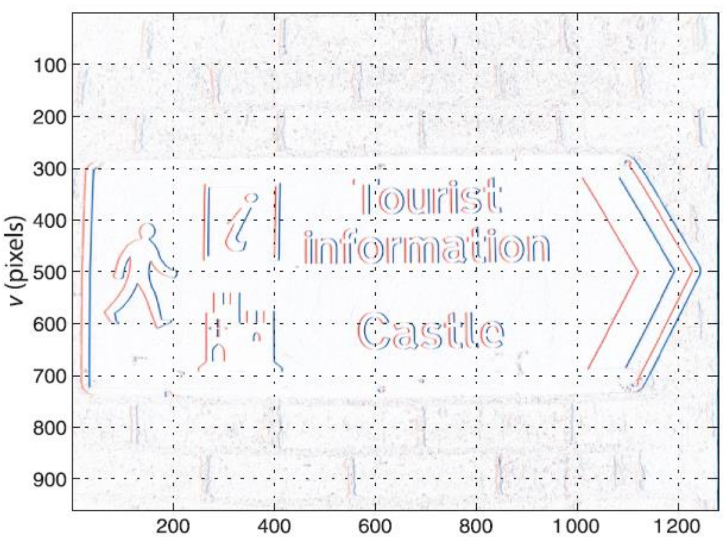
Derivative (1st order difference)



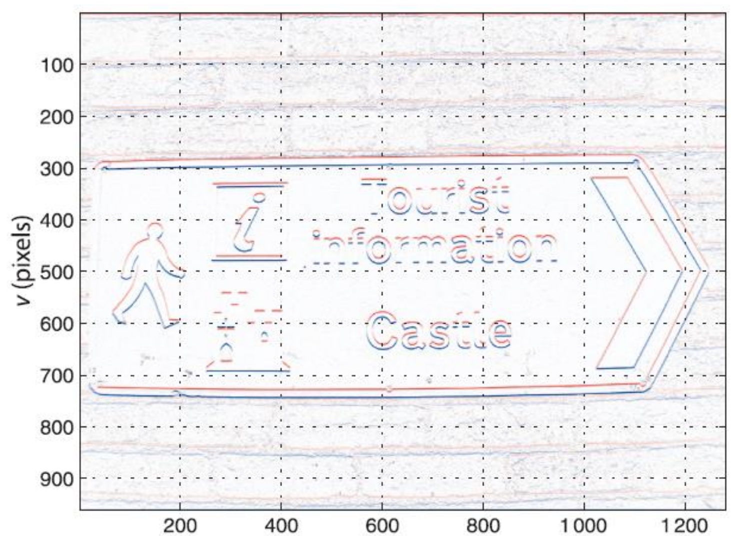
Gradient magnitude



c



a



b

Sobel kernel

first order difference

$$p'[v] = p[v] - p[v-1]$$

$$p'[v] = \frac{1}{2}\left(p[v+1] - p[v-1]\right)$$

symmetrical first order difference

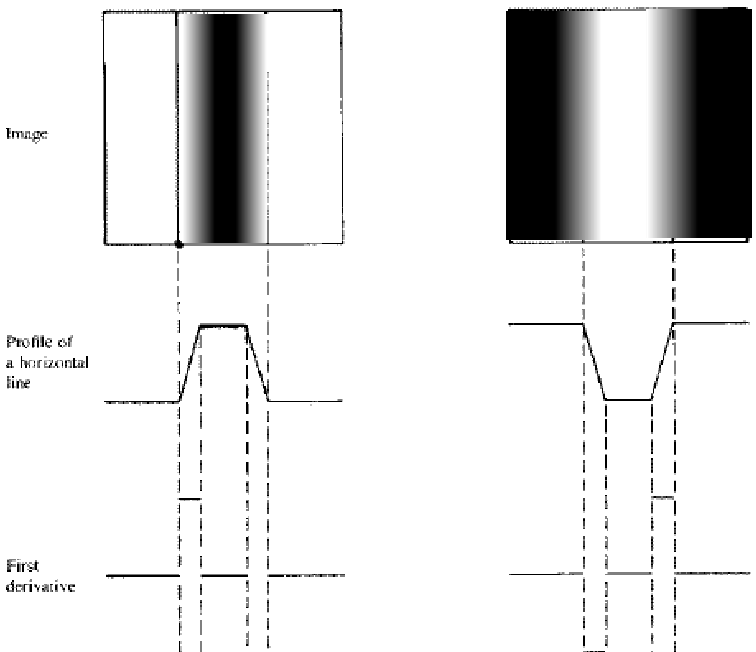
Derivate Operator: SOBEL kernel

$$D = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Transposing for vertical gradient

$$I_u = \frac{\partial I}{\partial u} = \nabla_u I = D \otimes I$$

$$I_v = \frac{\partial I}{\partial v} = \nabla_v I = D^T \otimes I$$



Edges

First Derivative

Sobel + G

However: gradient operators amplify noise: <u>high pass filters</u>

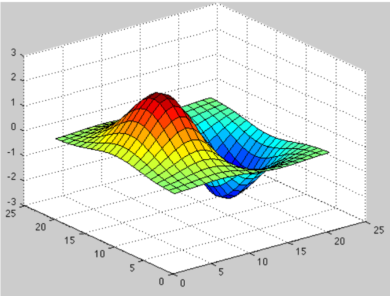
Solution: apply a Gaussian smoothing: <u>low pass filter</u> ⇒ ***<u>band pass filter</u>***

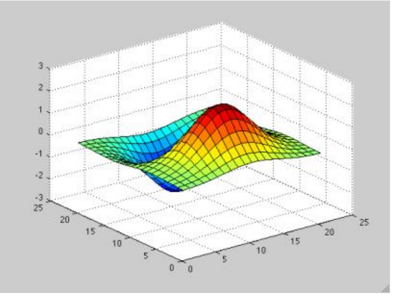
Instead of convolving the image with the Gaussian and *then* the derivative, we exploit the associative property of convolution to write

associativity

$$\nabla \mathbf{I} = \mathbf{D} \otimes \left(\mathbf{G}(\sigma) \otimes I\right) = \underbrace{\left(\mathbf{D} \otimes \mathbf{G}(\sigma)\right)}_{\text{DoG}} \otimes \mathbf{I}$$

$$A \otimes B \otimes C = (A \otimes B) \otimes C = A \otimes (B \otimes C)$$





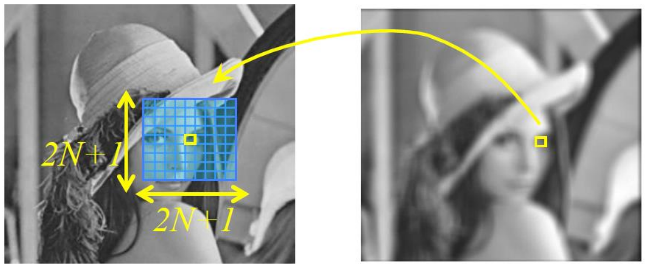
$$G_u = \frac{\partial G}{\partial u} = -\frac{u}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

$$G_v = \frac{\partial G}{\partial v} = -\frac{v}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Key points on smoothing + derivative masks

Smoothing masks

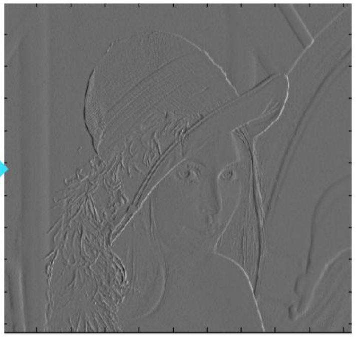
- Values positive
- Always **sum to 1** → constant regions same as input
- Amount of **smoothing proportional to mask size**



Derivative masks

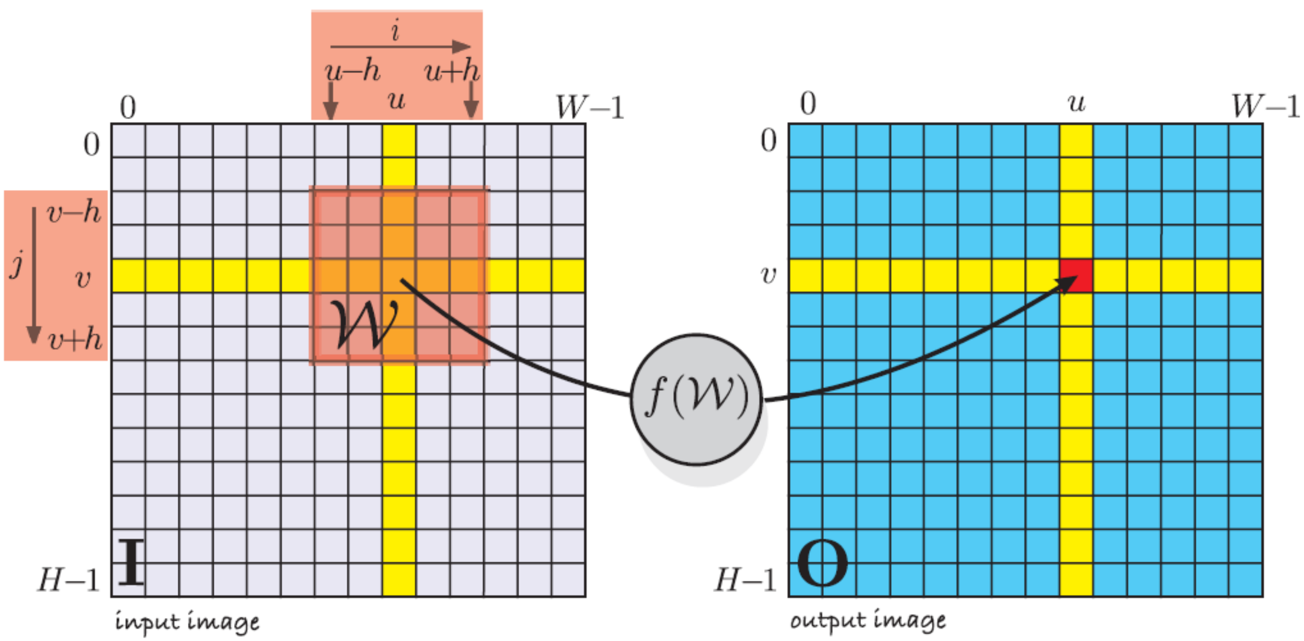
- Opposite signs used to get high response in regions of high contrast
- Always **sum to 0** → no response in constant regions
- High absolute value at points of high contrast

$$F_x = \begin{pmatrix} 1 & -1 \end{pmatrix}$$



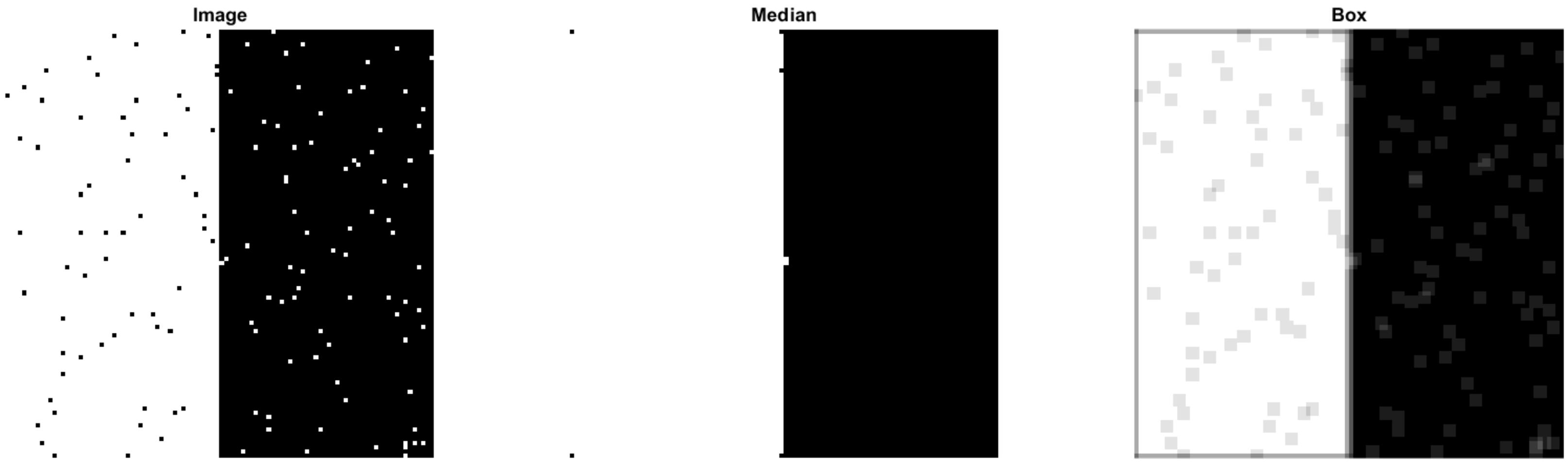
Spatial Operators – Median Filter

- f(W) = M(W)
- Non-linear – NOT convolution, non-associative
- Requires ordering/sorting = expensive



Median Filter examples

- Preserves edges & handles salt and pepper noise



- Expensive

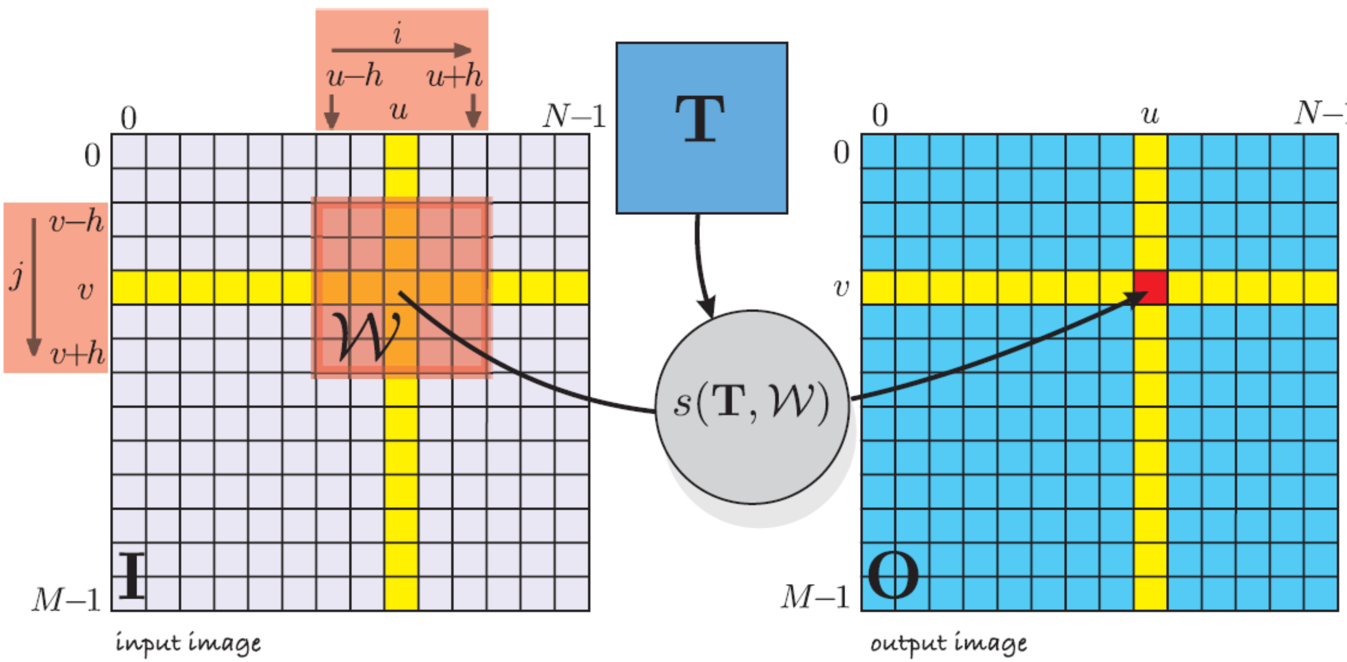
```
I = [ones(100,50),zeros(100,50)];
>> k = fspecial('average',15);
>> tic; If = conv2(I,k); toc;
Elapsed time is 0.004001 seconds.
>> tic; Im = medfilt2(I,[15,15]); toc;
Elapsed time is 0.017216 seconds.
```

Spatial Operations – Template matching

Goal: find which parts of the input image are more similar to the template

$$\mathbf{O}[u,v] = s(\mathrm{T}, \mathcal{W}), \quad \forall (u,v) \in \mathbf{I}$$

<u>s – SIMILARITY function, T – template</u>



Template matching – Similarity functions

$SAD: \geq 0$

$$s = \Sigma_{(u,v)\in I} \left| I_1[u,v] - I_2[u,v] \right|$$

sum of absolute differences

$SSD: \geq 0$

$$s = \Sigma_{(u,v)\in I} \left(I_1[u,v] - I_2[u,v] \right)^2$$

sum of square differences

$NCC: \in [0;1] \quad Good\ Match \approx 0.8$

normalized cross-correlation

$$s = \frac{\Sigma_{(u,v)\in I} I_1[u,v] \cdot I_2[u,v]}{\sqrt{\Sigma_{(u,v)\in I} I_1^2[u,v] \cdot \Sigma_{(u,v)\in I} I_2^2[u,v]}}$$

Invariant to brightness <u>gain</u>