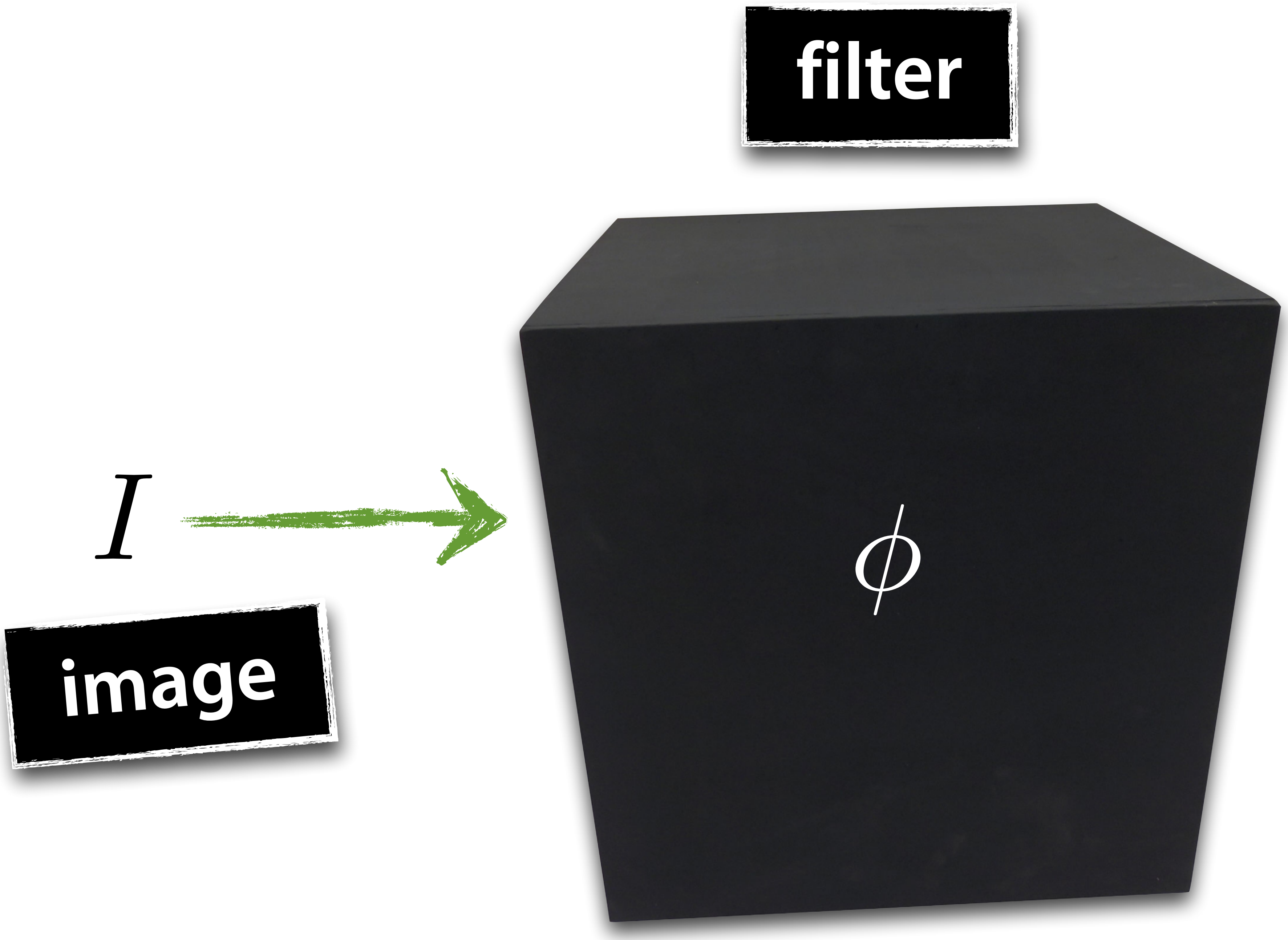
Intro to

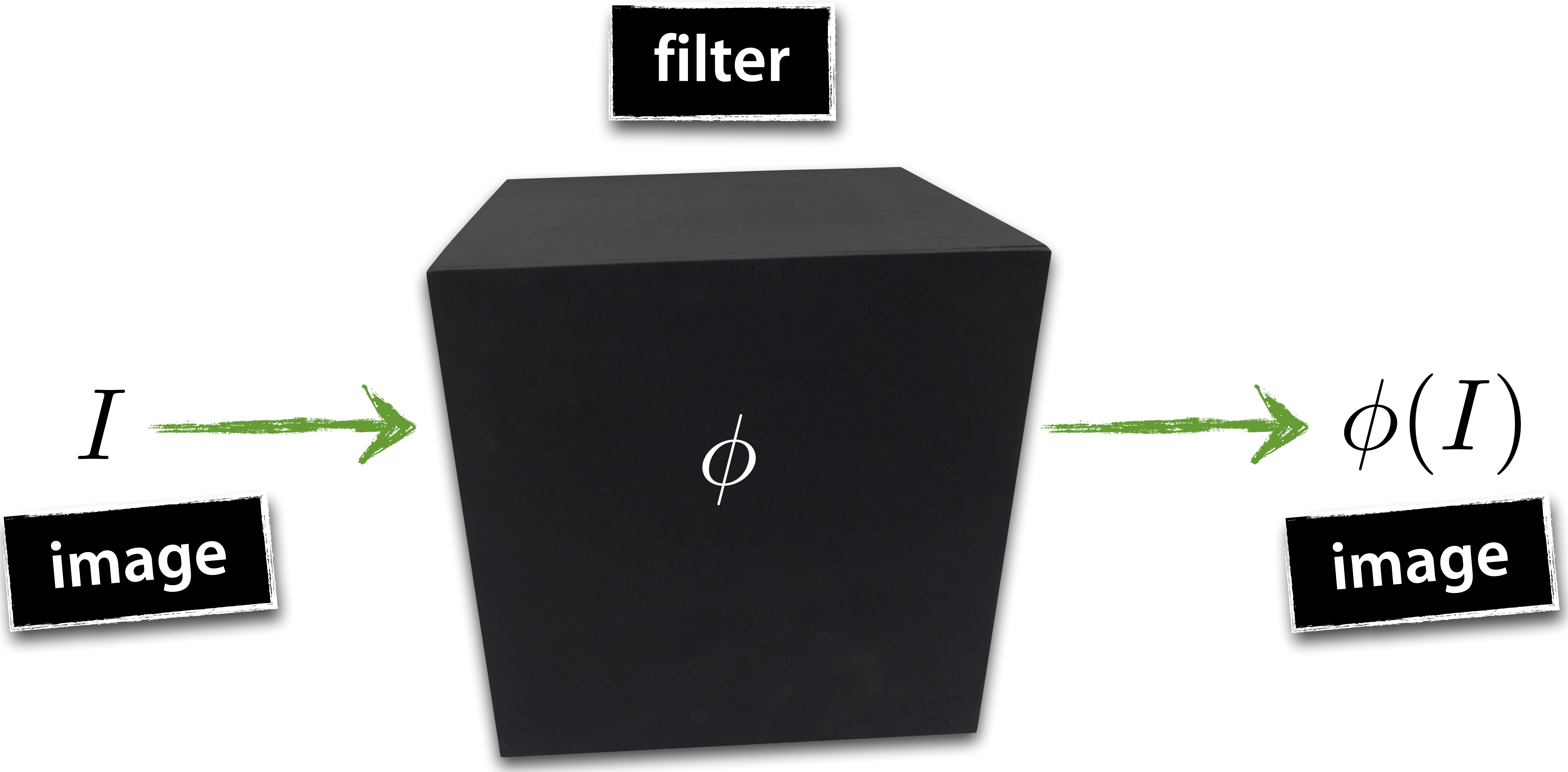
Computer Vision

Image filtering

Ryerson University

filter


filter
I
image
ϕ


filter
I
image
ϕ
$\phi(I)$
image

Image

NOISE

Image

NOISE


original


salt and pepper noise


Gaussian noise

image noise

$$I(x,y) = I_{\text{ideal}}(x,y) + \eta(x,y)$$

image noise

$$I(x,y) = I_{\text{ideal}}(x,y) + \eta(x,y)$$

where

$$\eta(x,y) \sim \mathcal{N}(\mu = 0, \sigma)$$

image noise

$$I(x,y) = I_{\text{ideal}}(x,y) + \eta(x,y)$$

where

$$\eta(x,y) \sim \mathcal{N}(\mu = 0, \sigma)$$

noise is assumed independent and identically distributed

image
noise

$$I(x, y) = I_{\text{ideal}}(x, y) + \eta(x, y)$$

where

$$\eta(x, y) \sim \mathcal{N}(\mu = 0, \sigma)$$

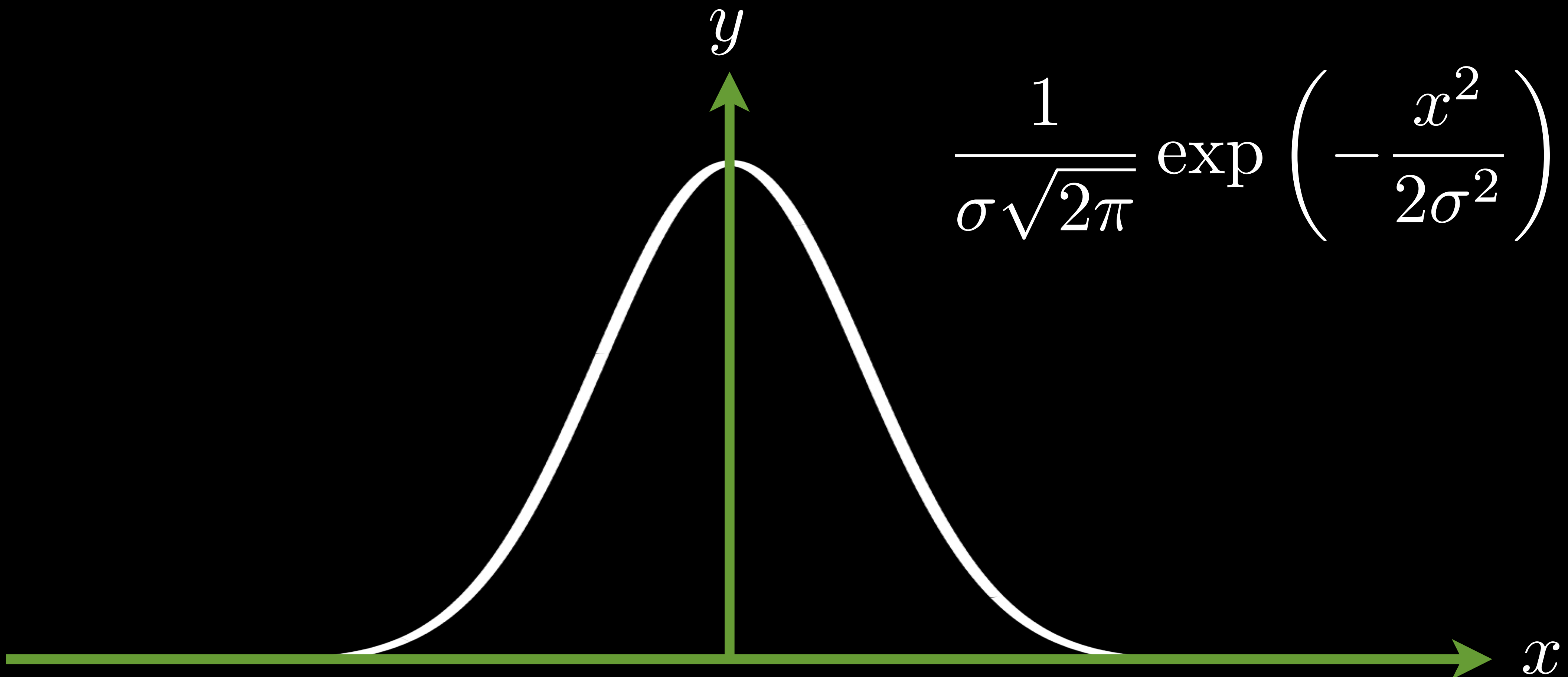
noise is assumed independent and identically distributed

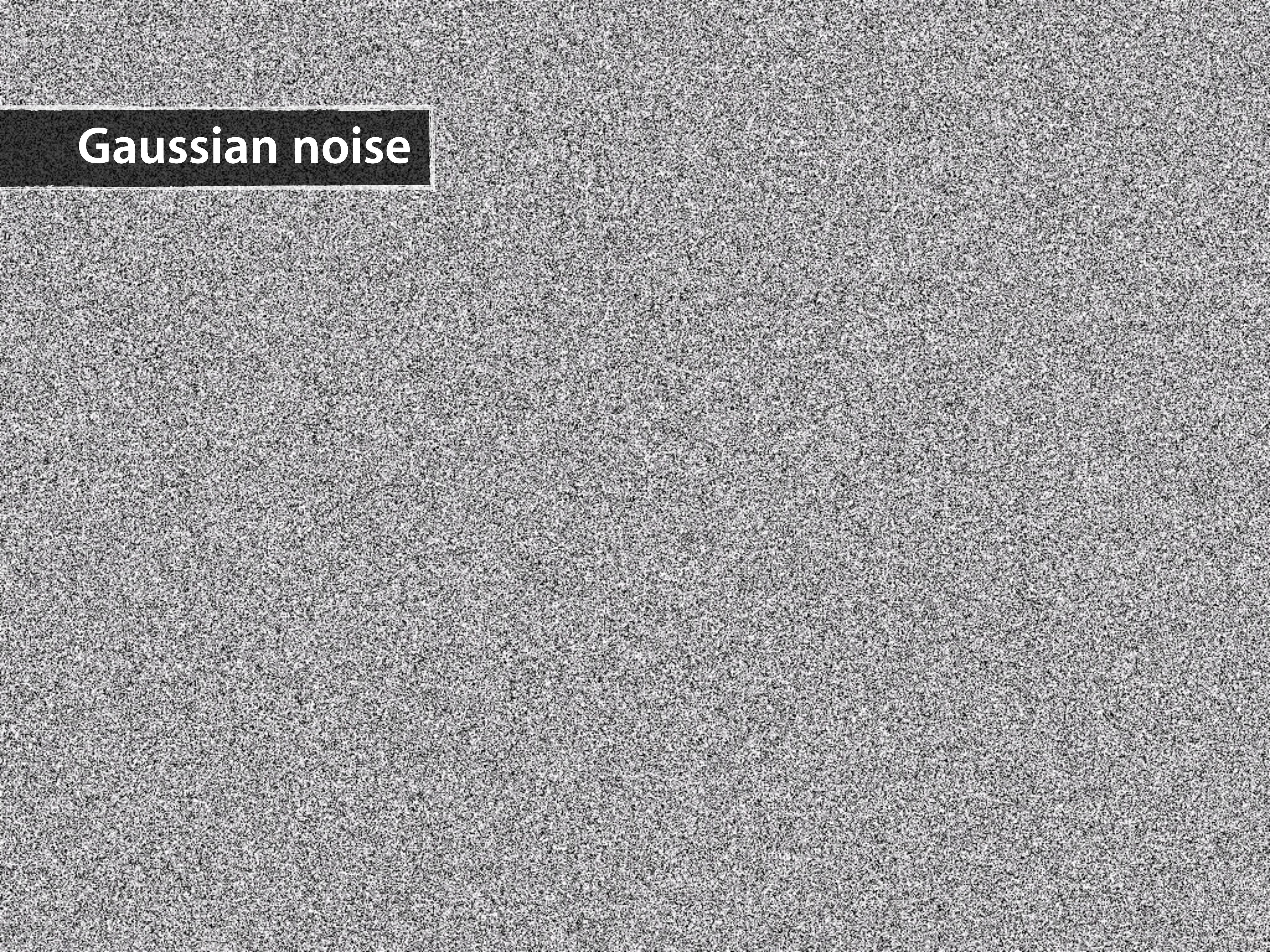
I.I.D.

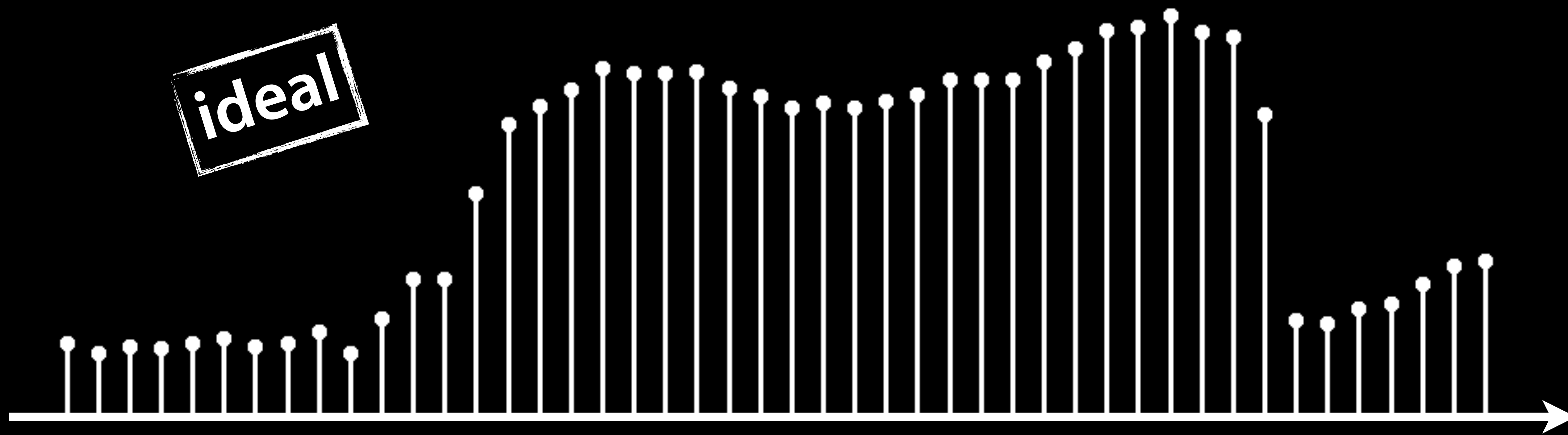
$$I(x,y) = I_{\text{ideal}}(x,y) + \eta(x,y)$$

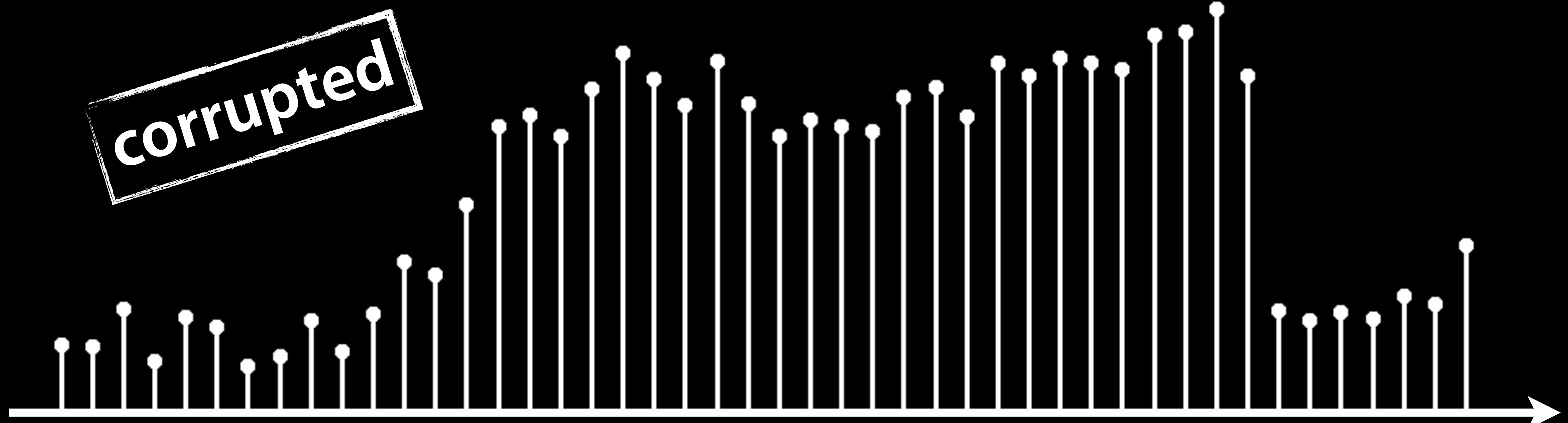
where

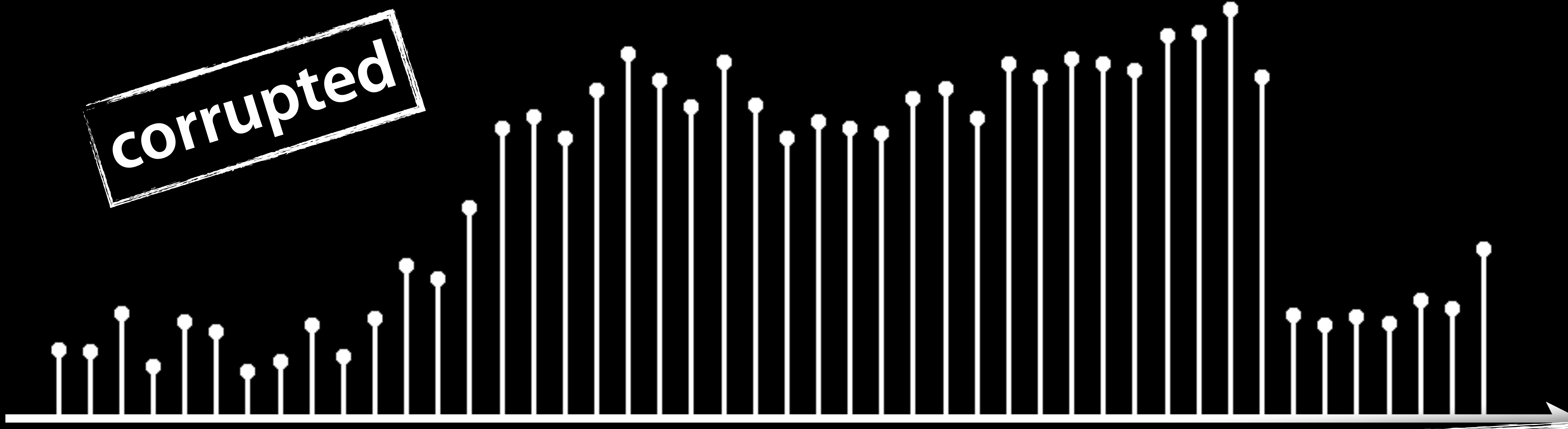
$$\eta(x,y) \sim \mathcal{N}(\mu = 0, \sigma)$$



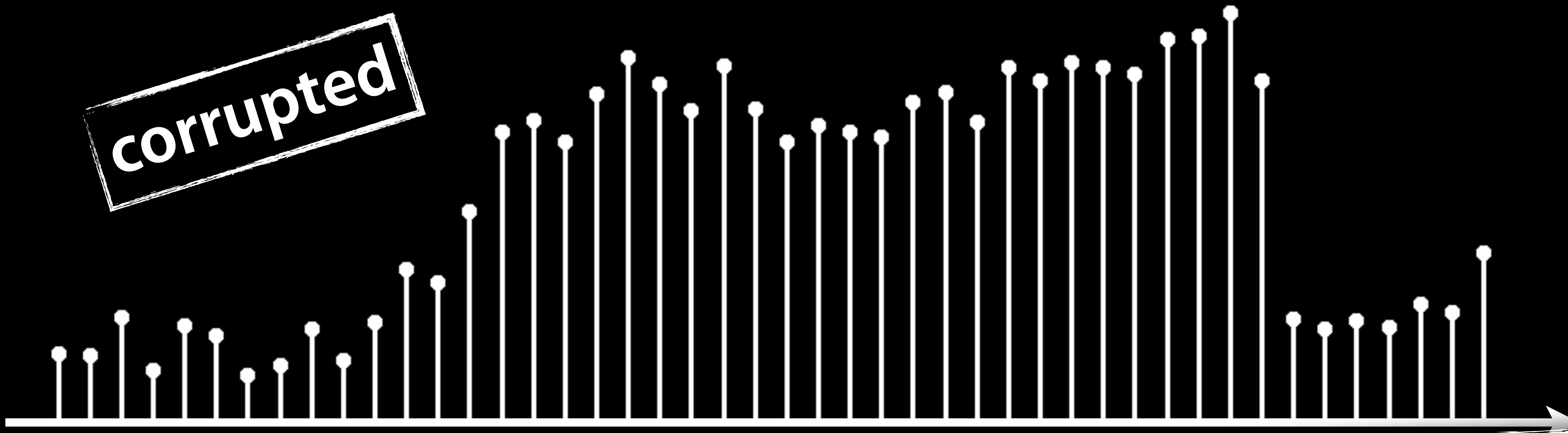

Gaussian noise


ideal

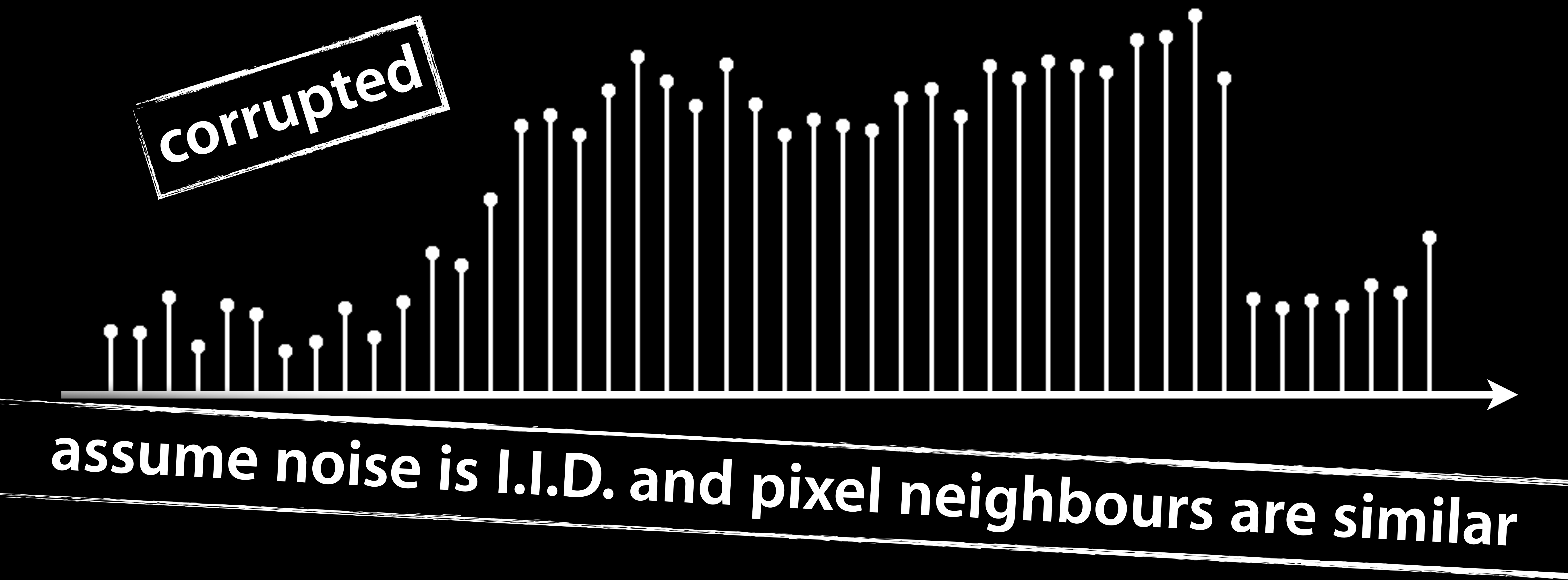

corrupted

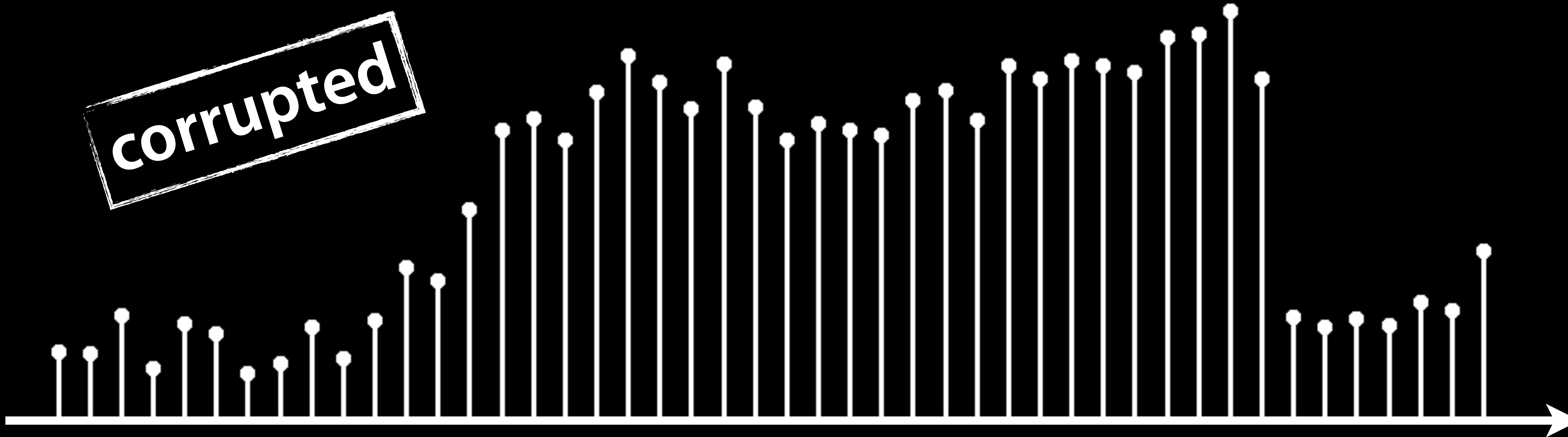


How can we remove the noise?

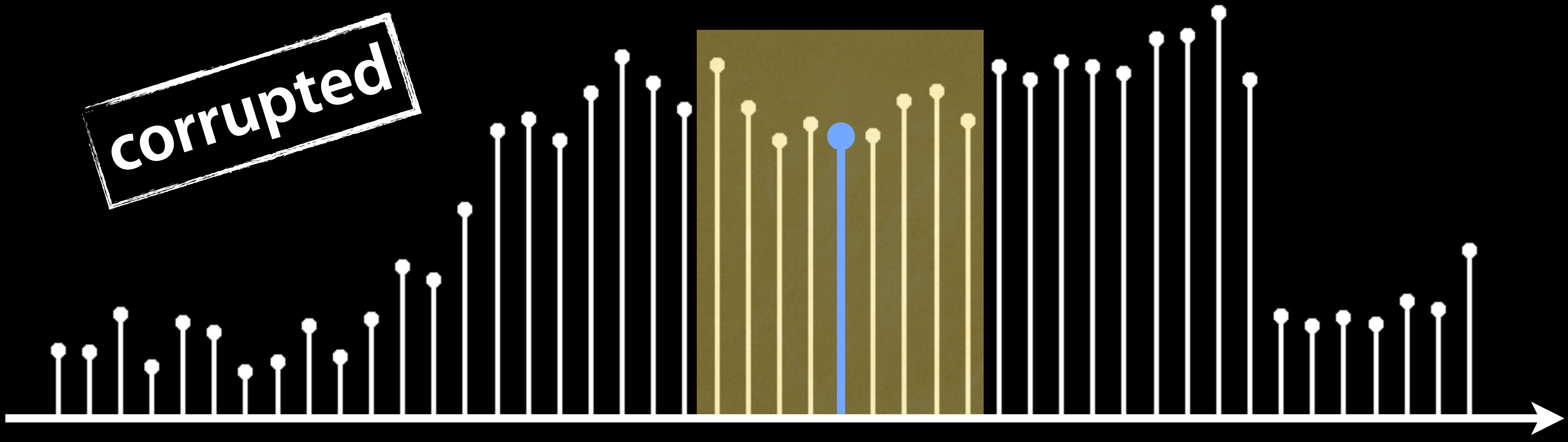


How can we remove the noise?

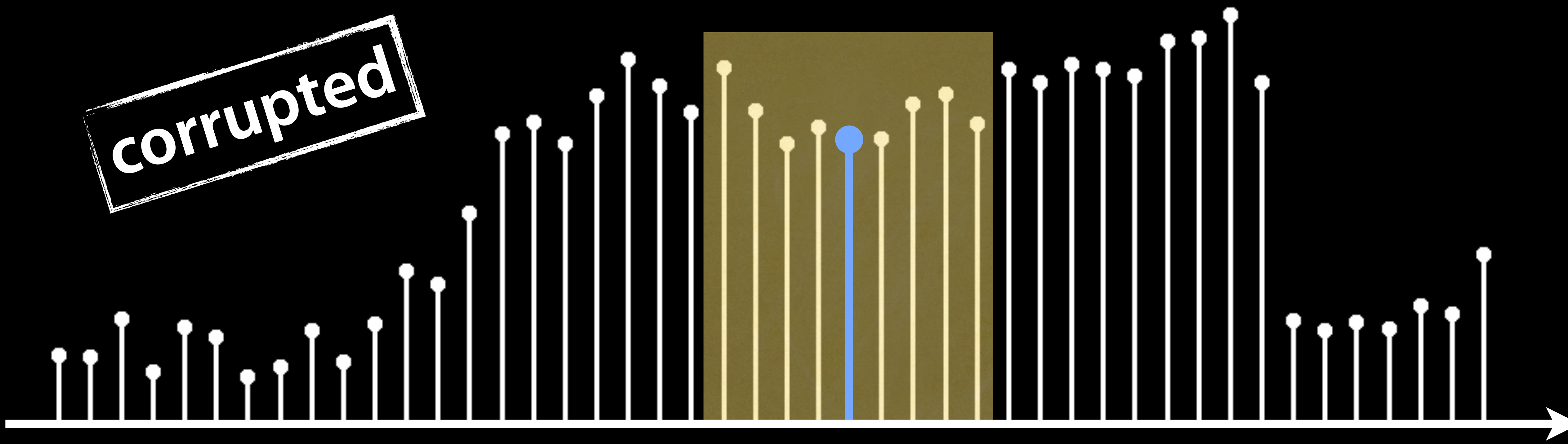

corrupted
assume noise is I.I.D. and pixel neighbours are similar



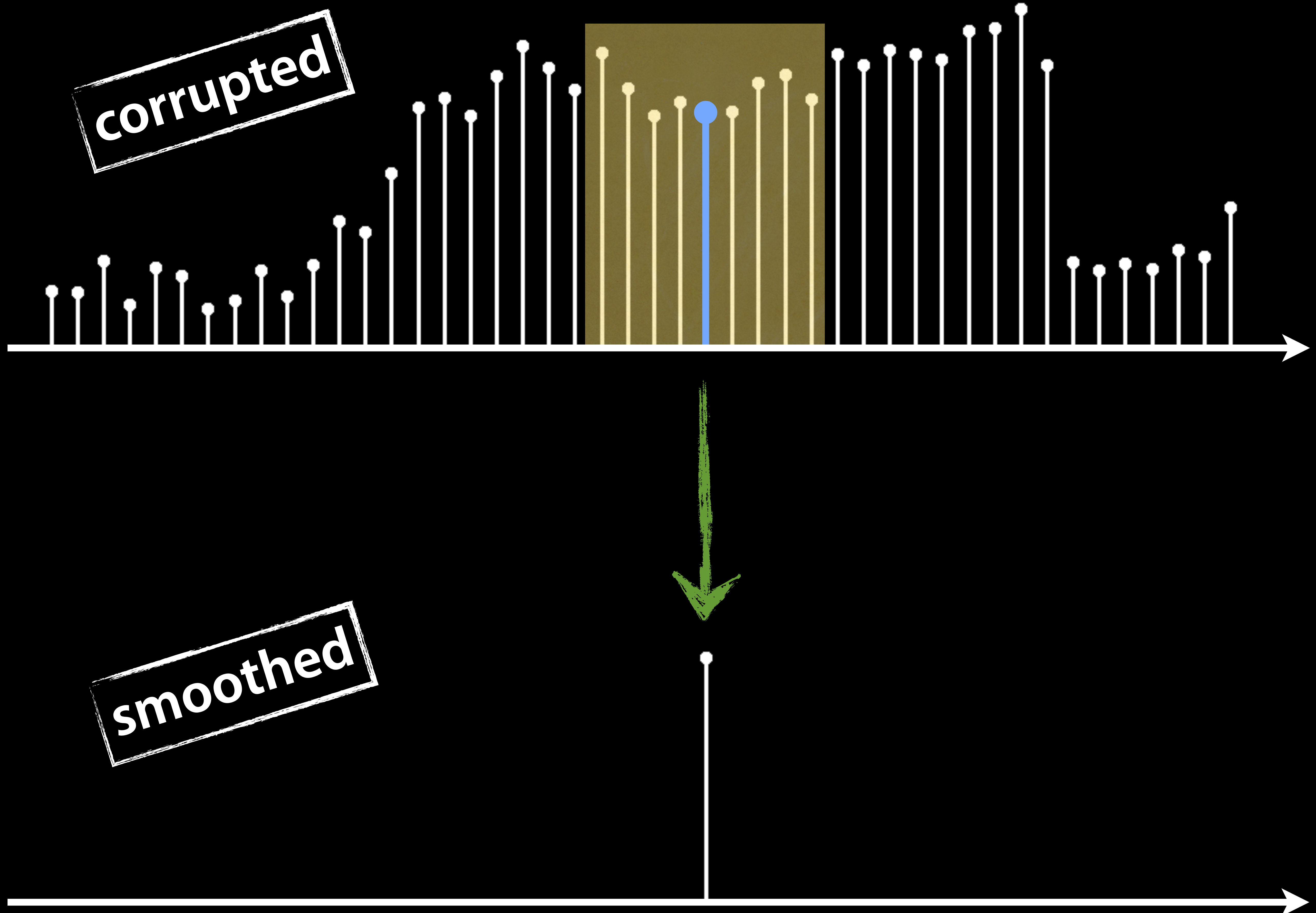
Let’s replace each pixel with an average of its neighours.

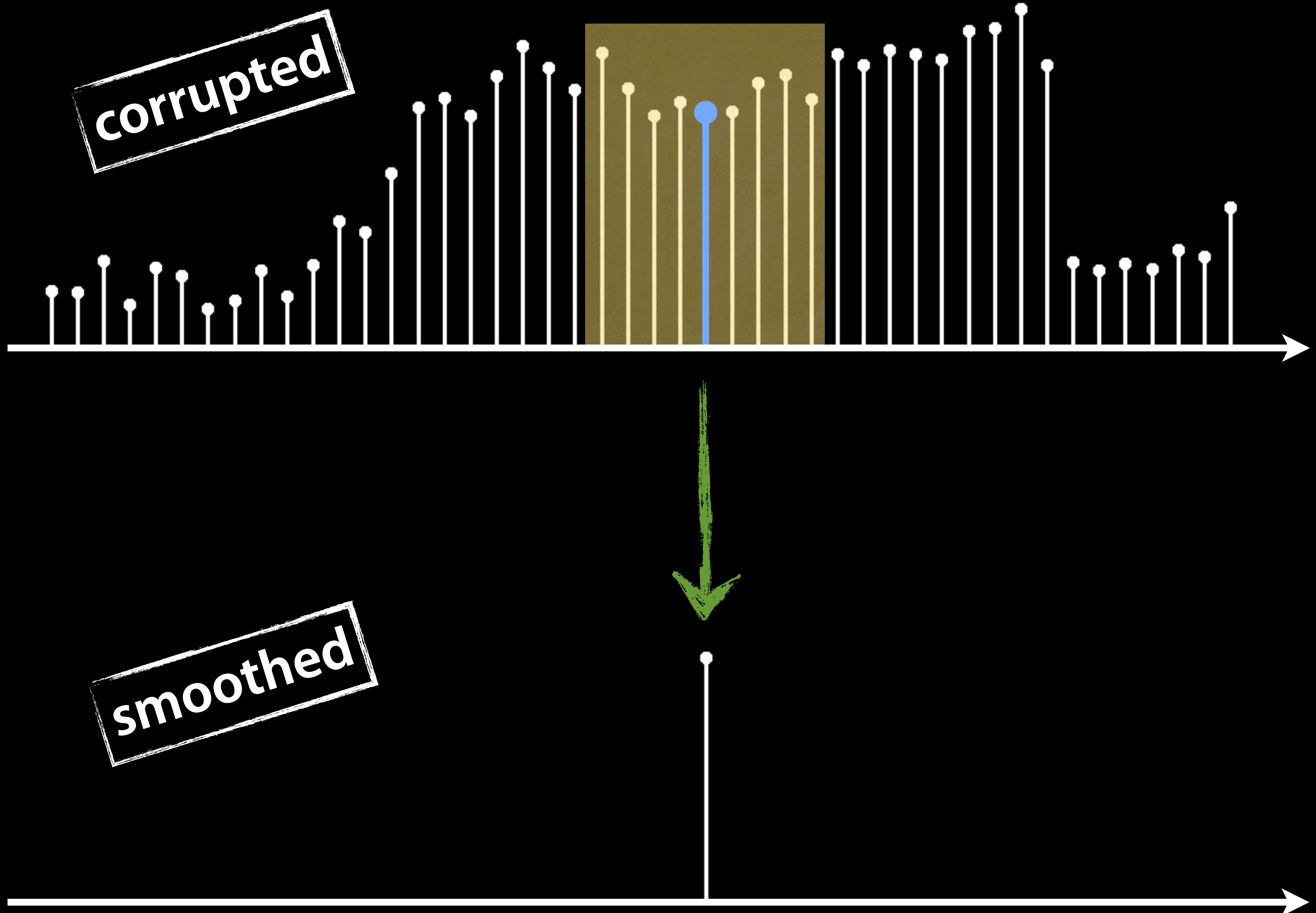


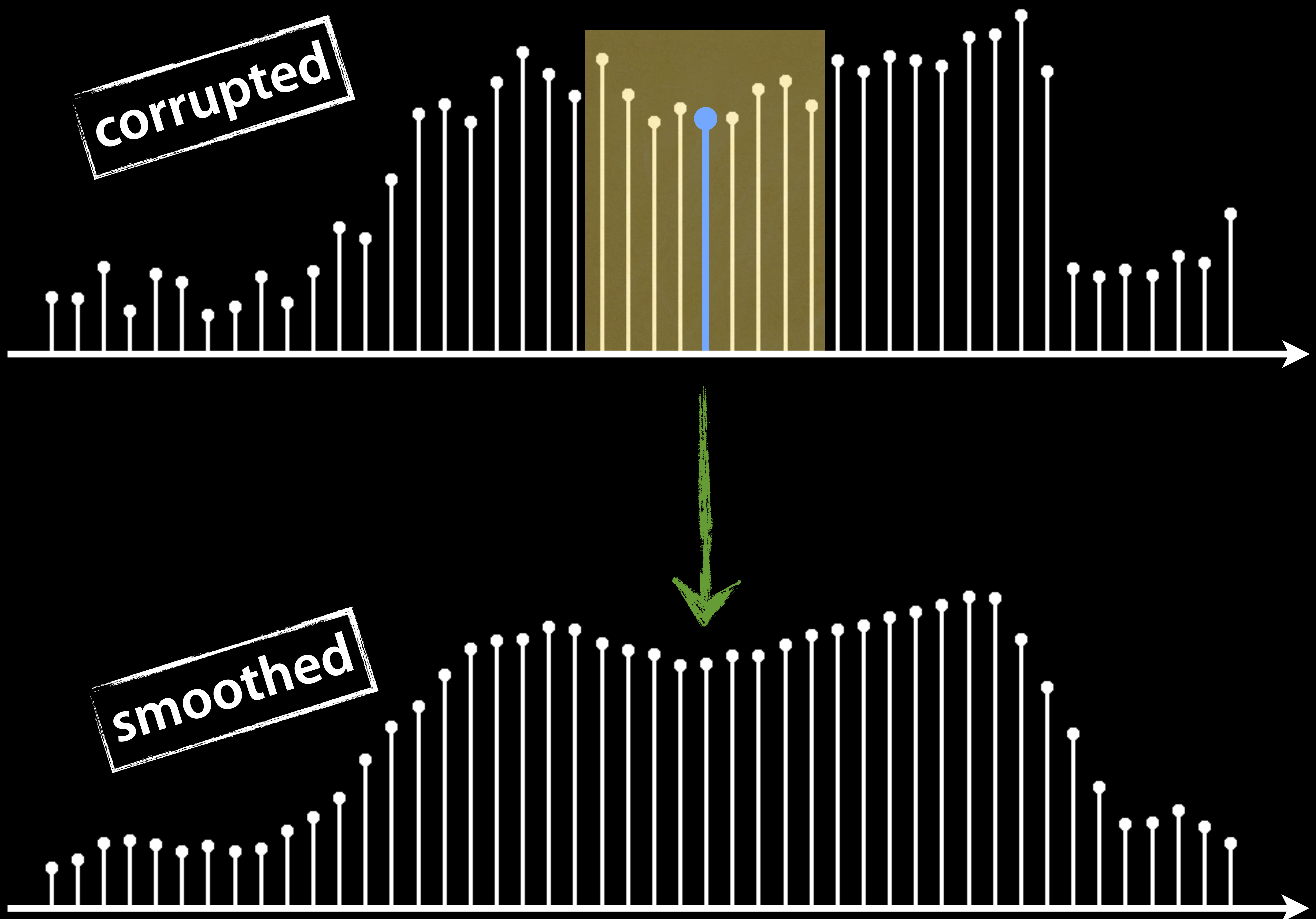
Let’s replace each pixel with an average of its neighbours.

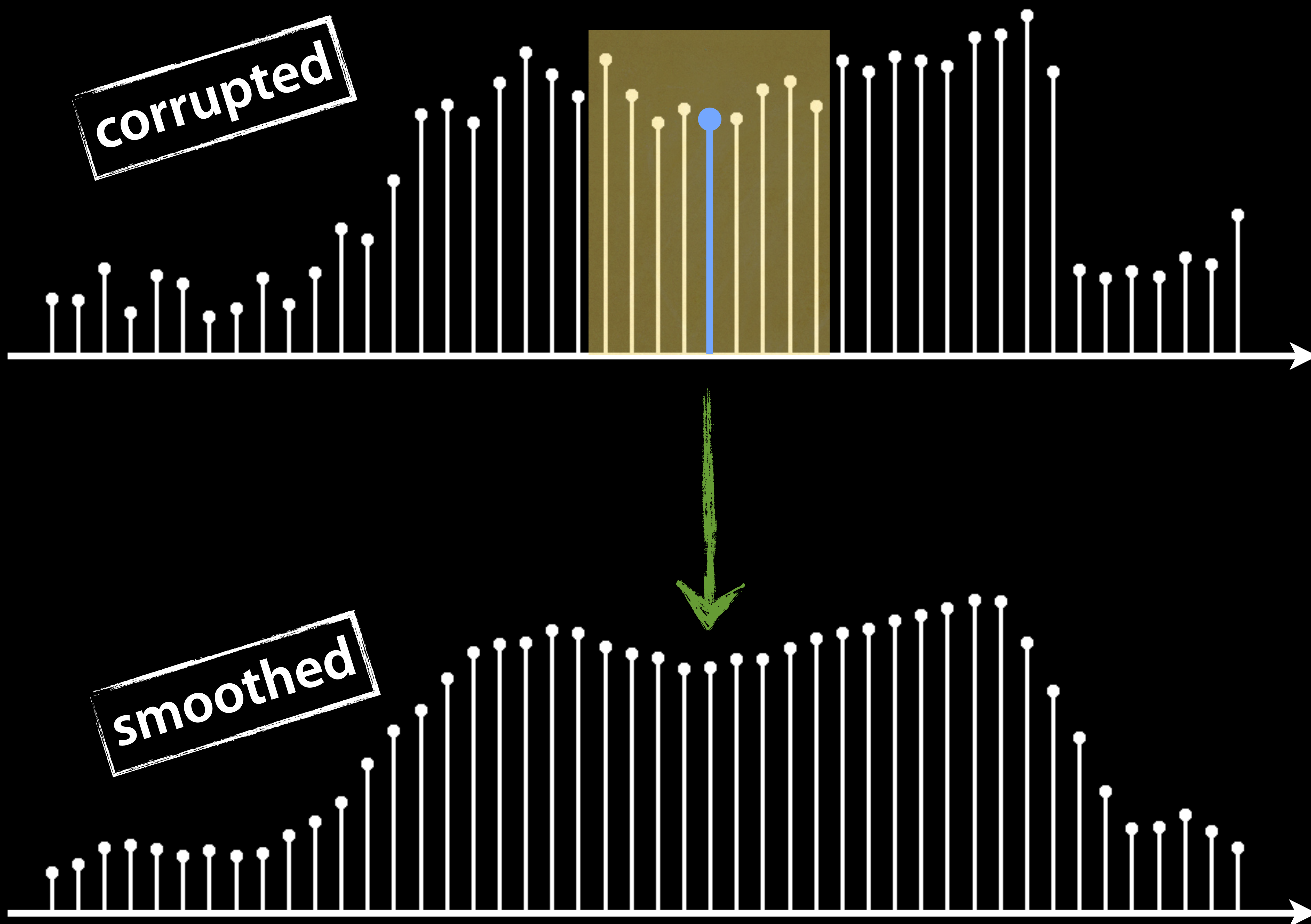


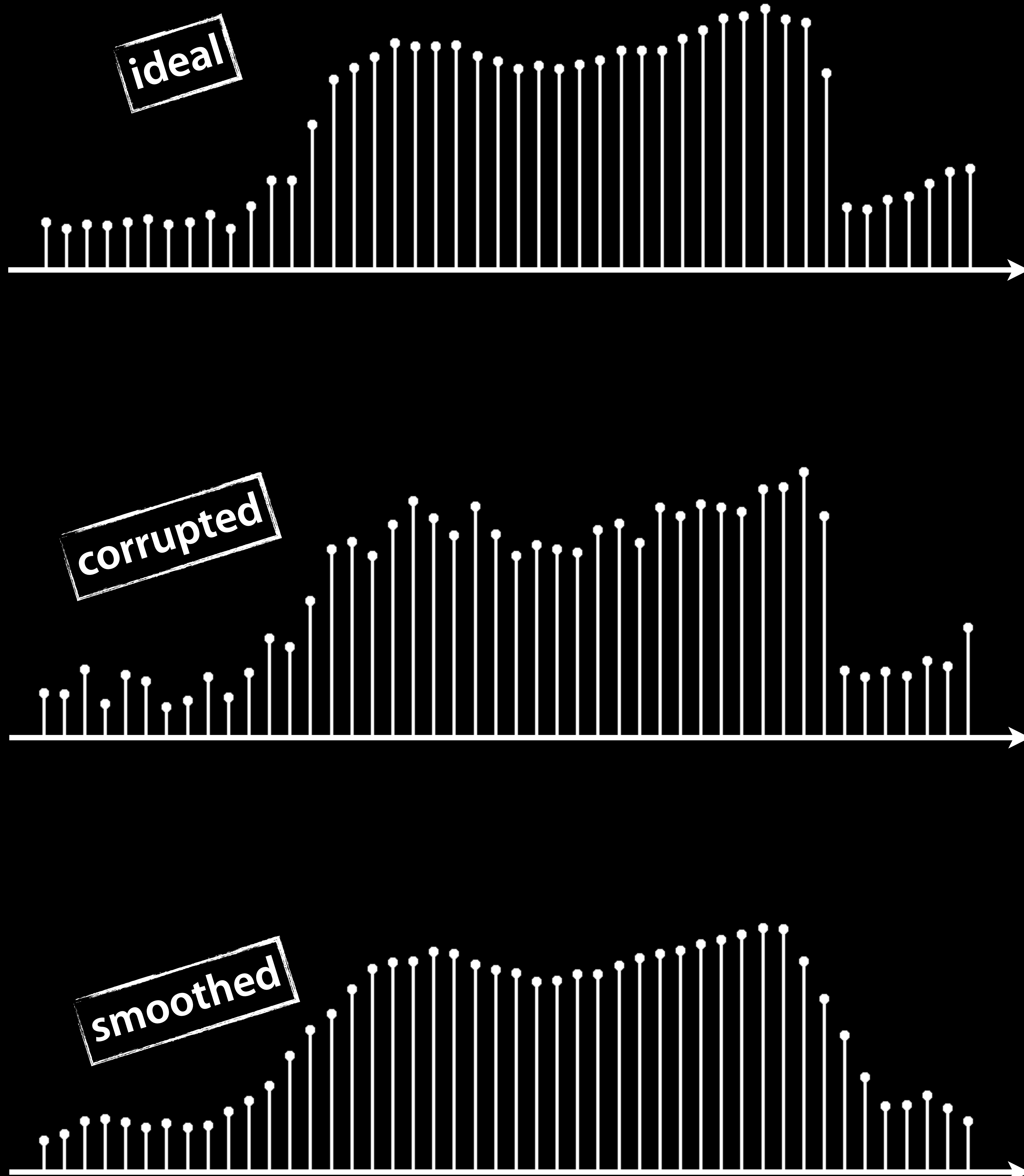
Let’s replace each pixel with an average of its neighbours.


corrupted
smoothed


corrupted
smoothed


corrupted
smoothed


corrupted
smoothed


ideal
corrupted
smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

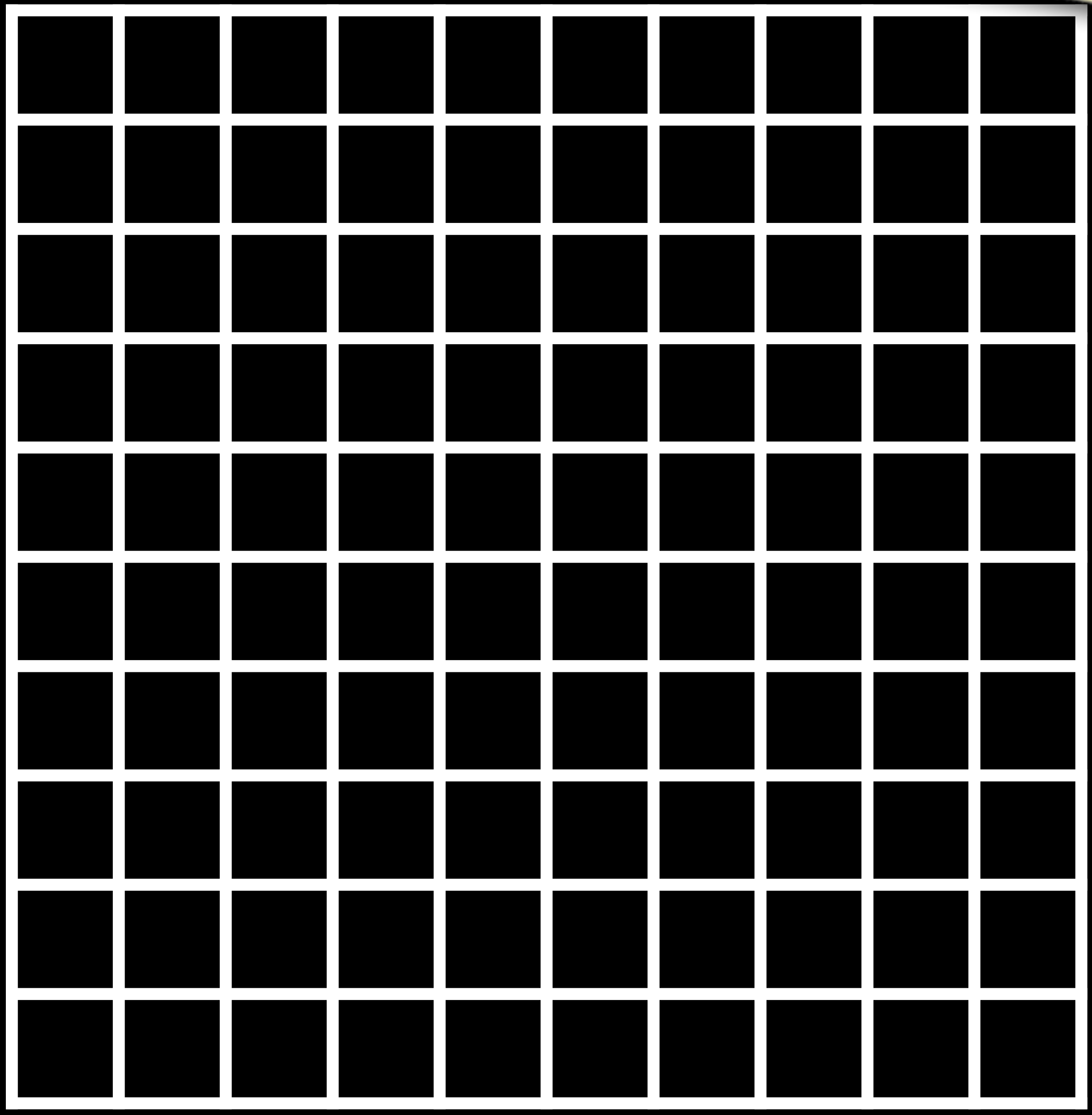
input

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

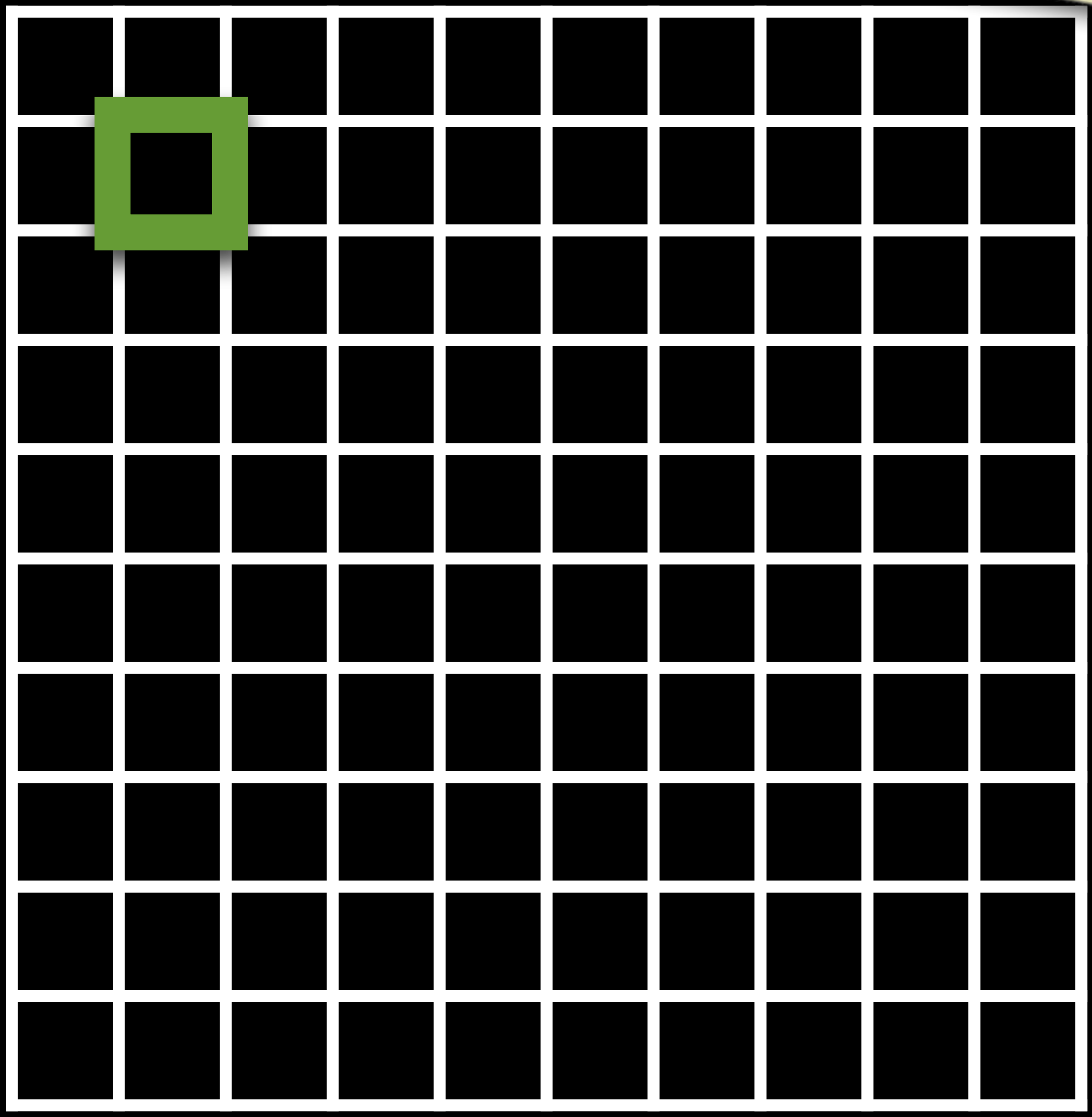


smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input



smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0								

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0								

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0								

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10							

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10							

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10							

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20						

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20						

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20						

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20	30					

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20	30					

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20	30					

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20	30	30				

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20	30	30				

smoothed

moving 2D average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

input

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

smoothed

Let the averaging window size be $(2K+1) \times (2K+1)$

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \frac{1}{(2K+1)^2} \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]$$

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \frac{1}{(2K+1)^2} \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]$$

output

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \frac{1}{(2K+1)^2} \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]$$

input

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \frac{1}{(2K+1)^2} \underbrace{\sum_{u=-K}^{K} \sum_{v=-K}^{K}}_{\text{loop over pixels in neighbourhood}} F[x+u, y+v]$$

loop over pixels in neighbourhood

Let the averaging window size be $(2K+1)\times(2K+1)$

$$H[x,y] = \underbrace{\frac{1}{(2K+1)^2}}_{\text{weight}} \sum_{u=-K}^{K}\sum_{v=-K}^{K} F[x+u,y+v]$$

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \frac{1}{(2K+1)^2} \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]$$

How can DIFFERENT WEIGHTS be included based on neighbourhood position?

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]\, G[u,v]$$

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]\, G[u,v]$$

mask, kernel or filter

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]\, G[u,v]$$

This is called cross correlation, denoted $H = G \otimes F$

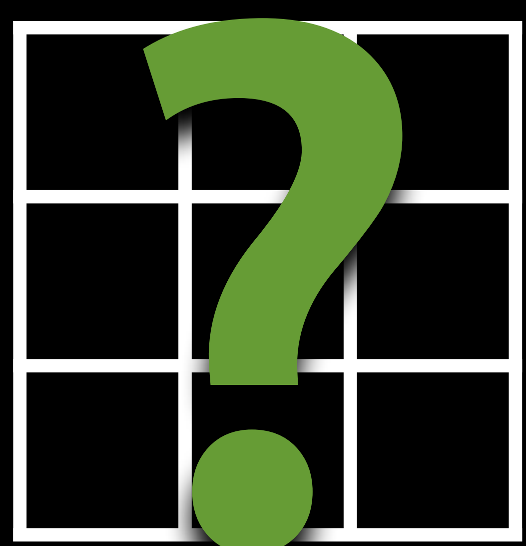
Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]\, G[u,v]$$

This is called cross correlation, denoted $H = G \otimes F$

Replace each pixel with a linear combination of its neighbours

average filtering



$G[x,y]$

$\otimes$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x,y]$

average filtering

$$
\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\otimes
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\
0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\
0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\
0 & 0 & 0 & 90 & 0 & 90 & 90 & 90 & 0 & 0 \\
0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 90 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

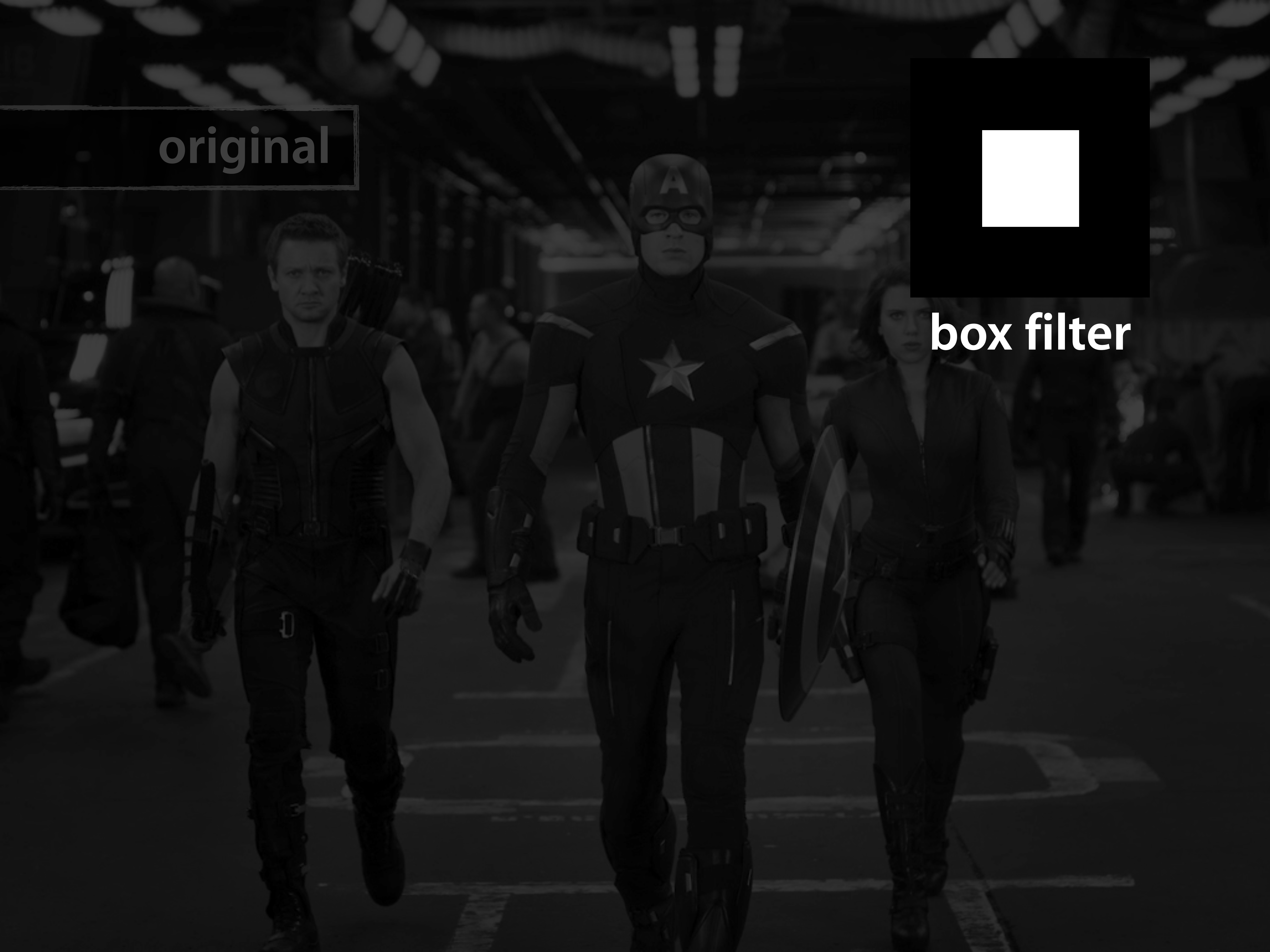
$G[x, y]$ $\qquad$ $F[x, y]$

average filtering

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \otimes \begin{bmatrix}0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&90&0&90&90&90&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&90&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\end{bmatrix}$$

$$G[x,y] \qquad\qquad F[x,y]$$


original


original
box filter


original


smooth by averaging

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \otimes \begin{bmatrix}0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&90&0&90&90&90&0&0\\ 0&0&0&90&90&90&90&90&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&90&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\end{bmatrix}$$

$$F[x,y]$$

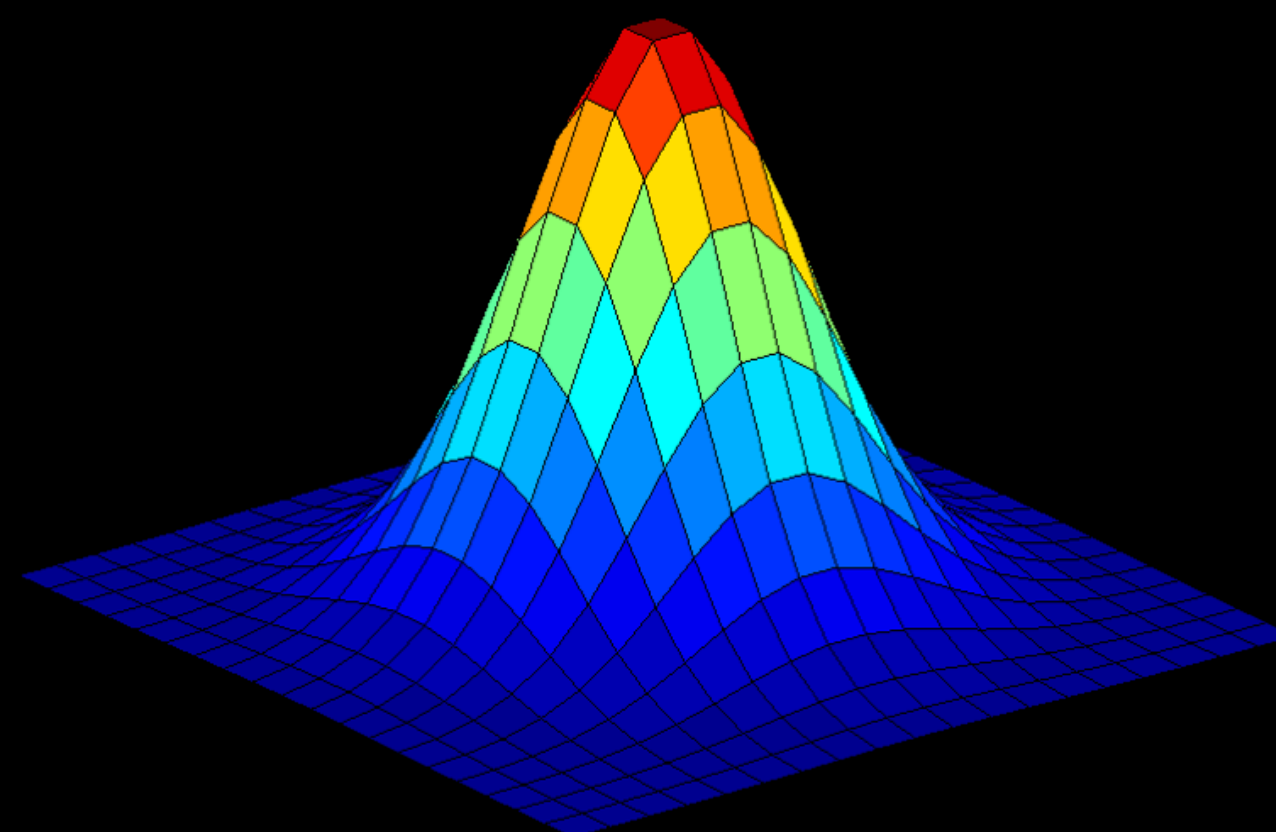
$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 0 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 90 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$G[x,y]$ $\qquad$ $F[x,y]$

Gaussian filtering

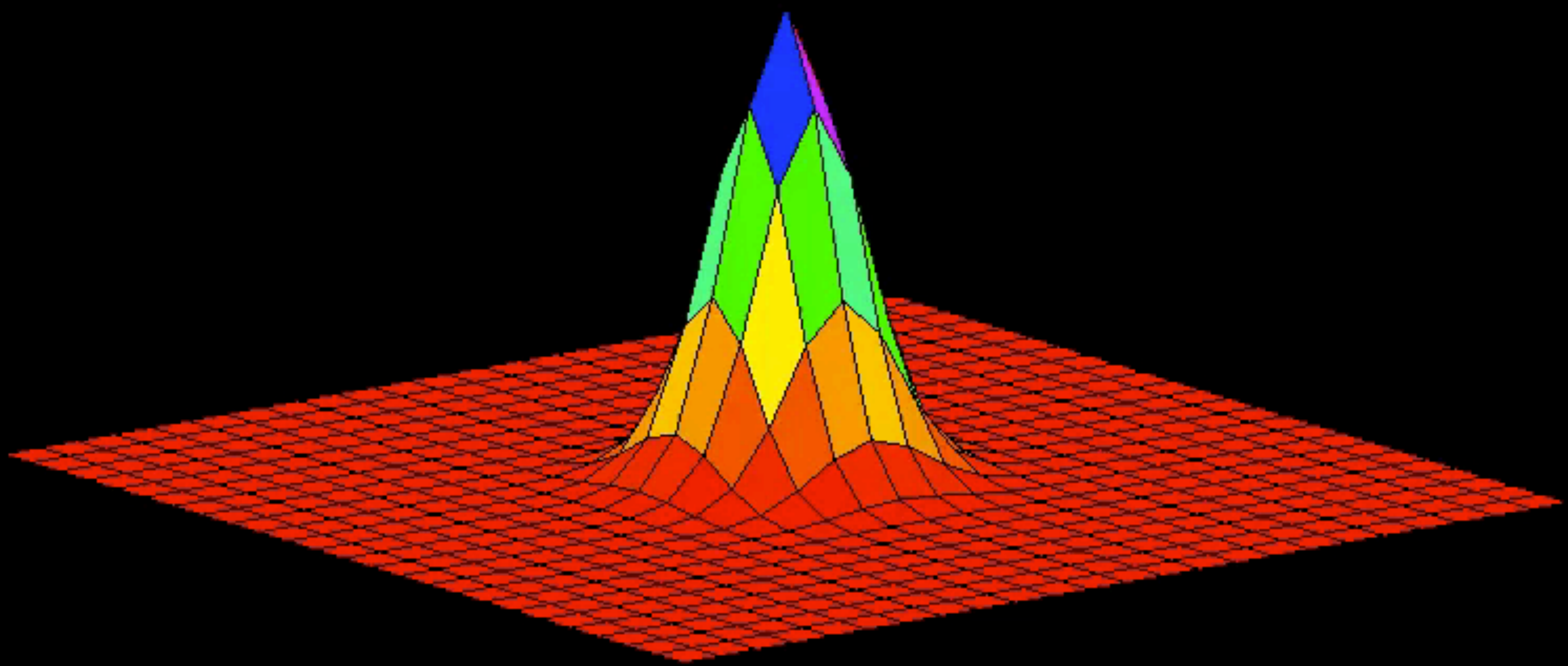
$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \approx$$

$$G[x, y]$$

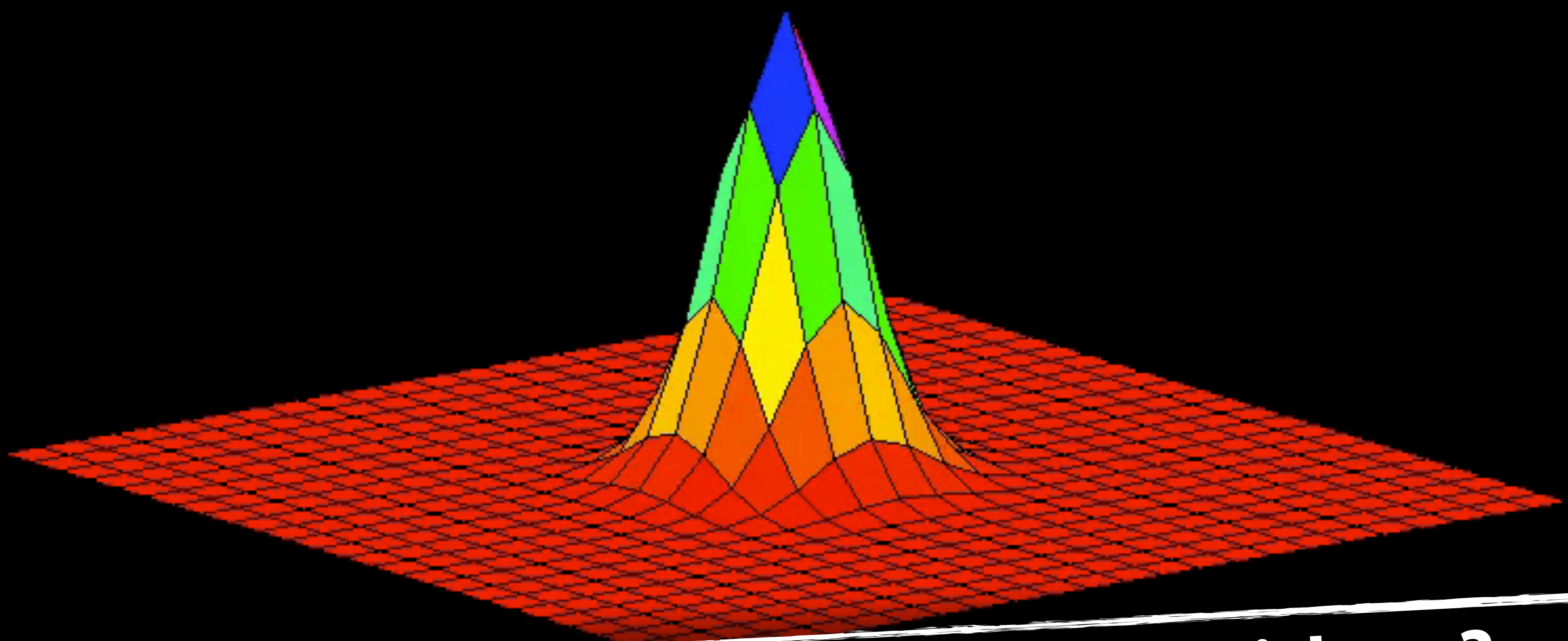


$$\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$



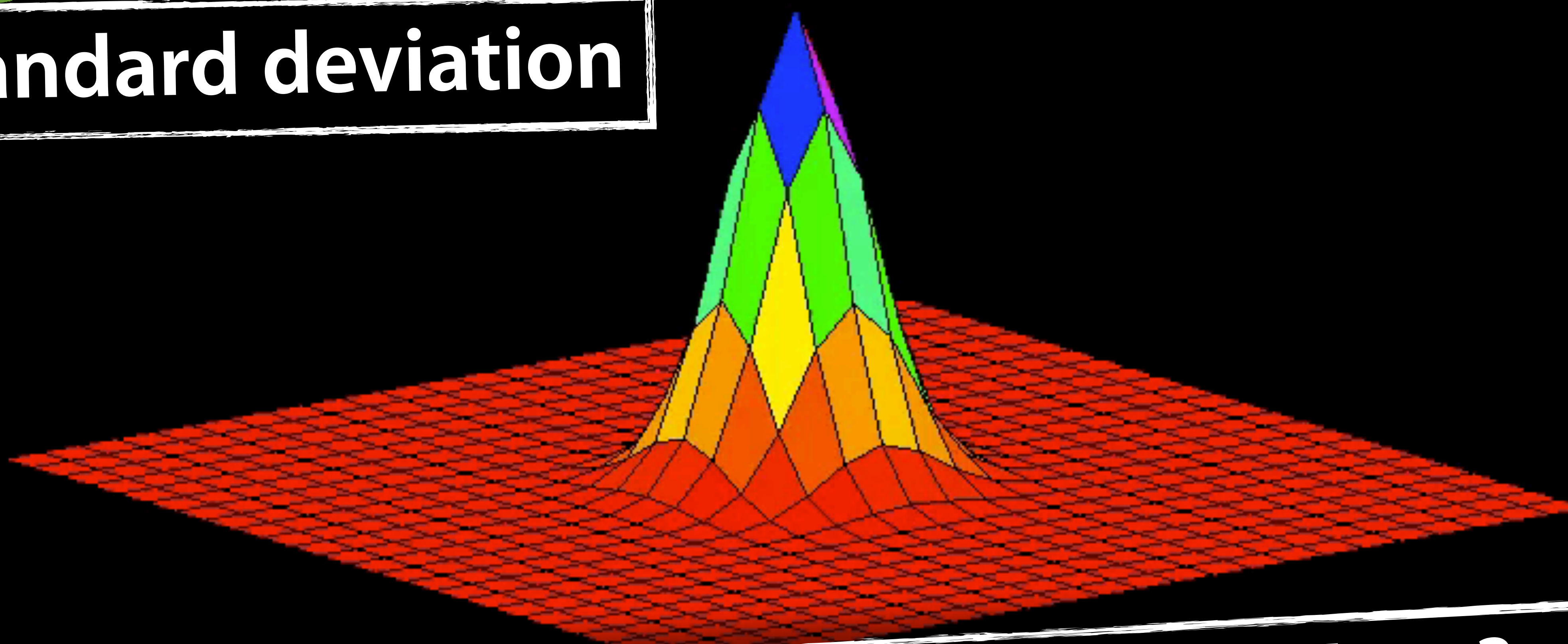
$$\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$



How does the Gaussian vary with σ ?

$$\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

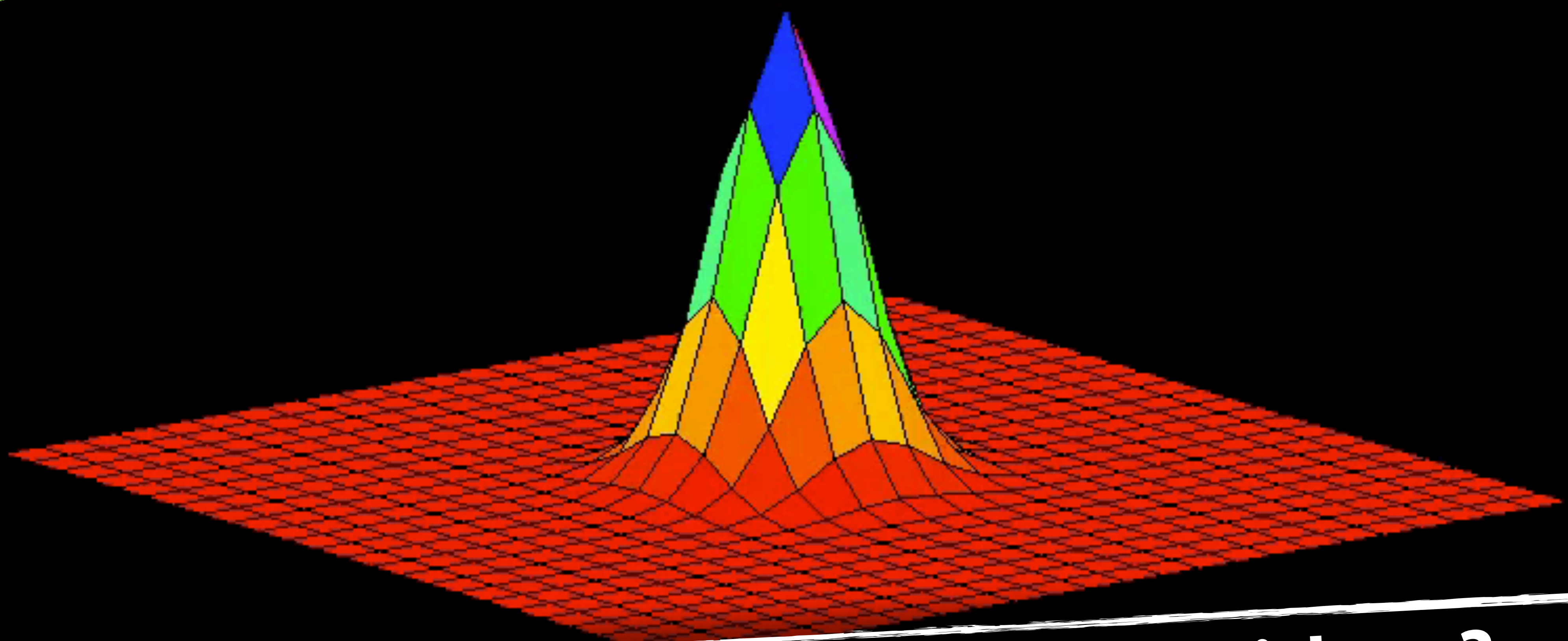
standard deviation



How does the Gaussian vary with σ ?

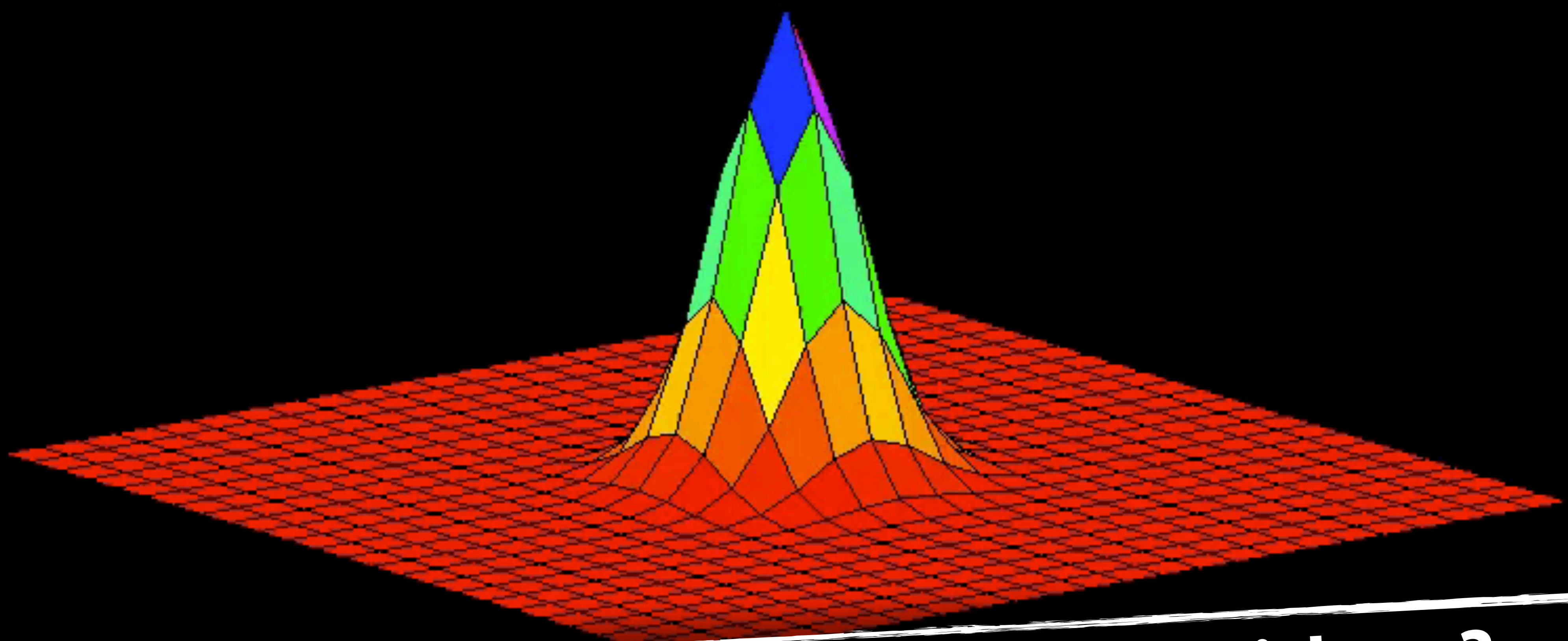
normalization

$$\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$



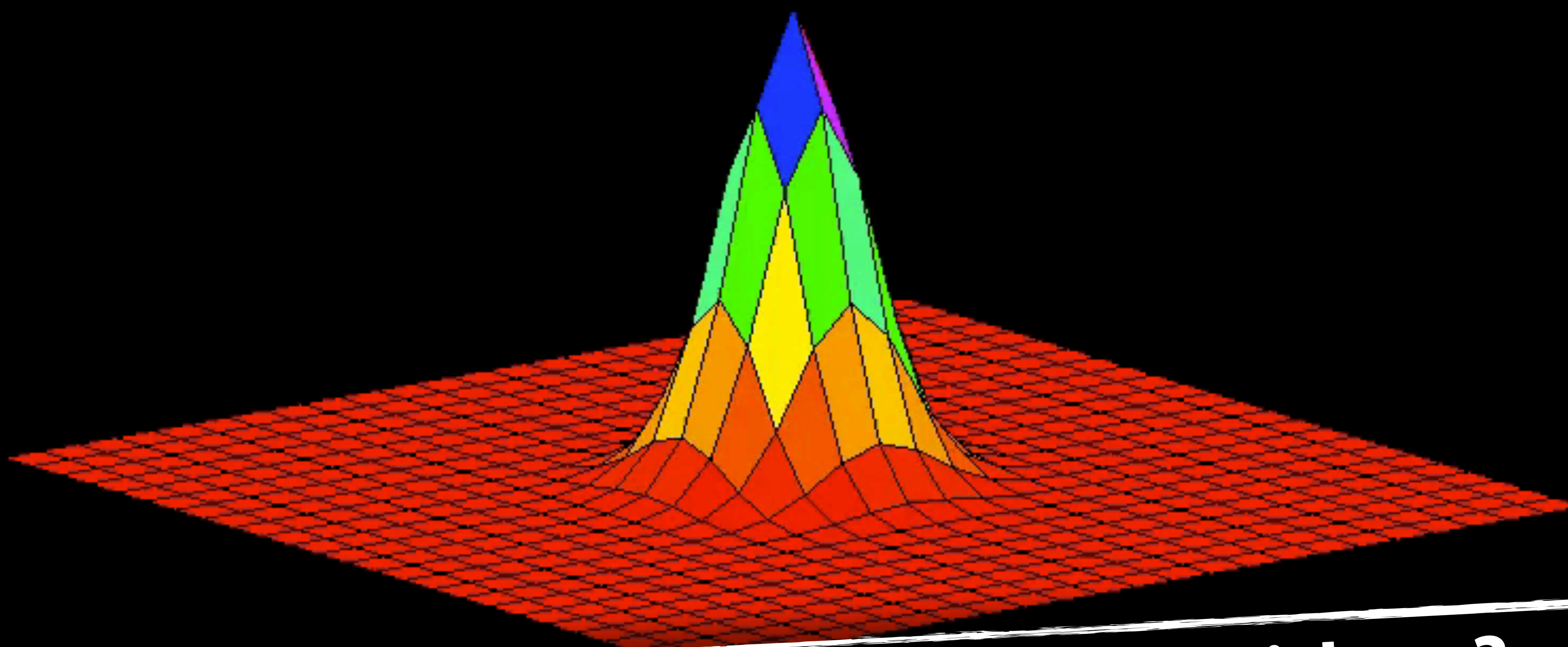
How does the Gaussian vary with σ ?

$$\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$



How does the Gaussian vary with σ ?

$$\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

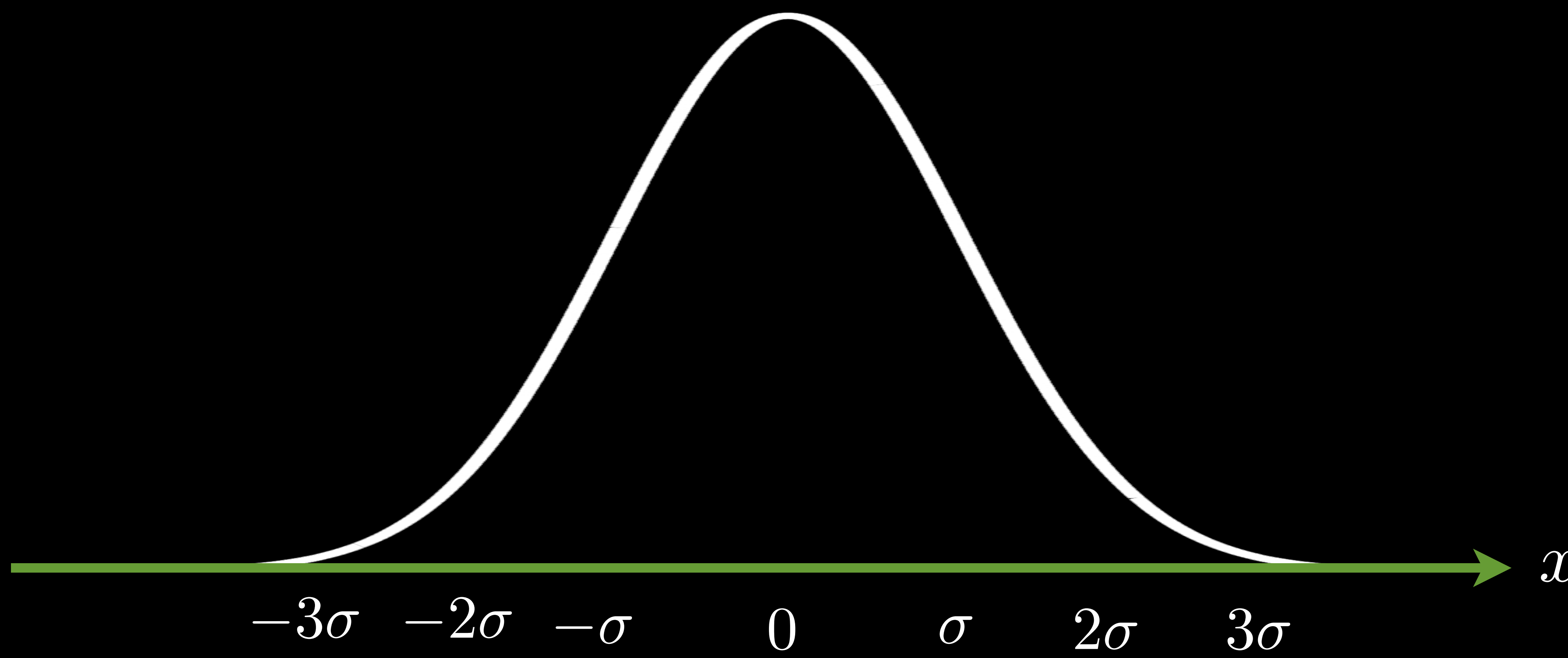


How does the Gaussian vary with σ ?

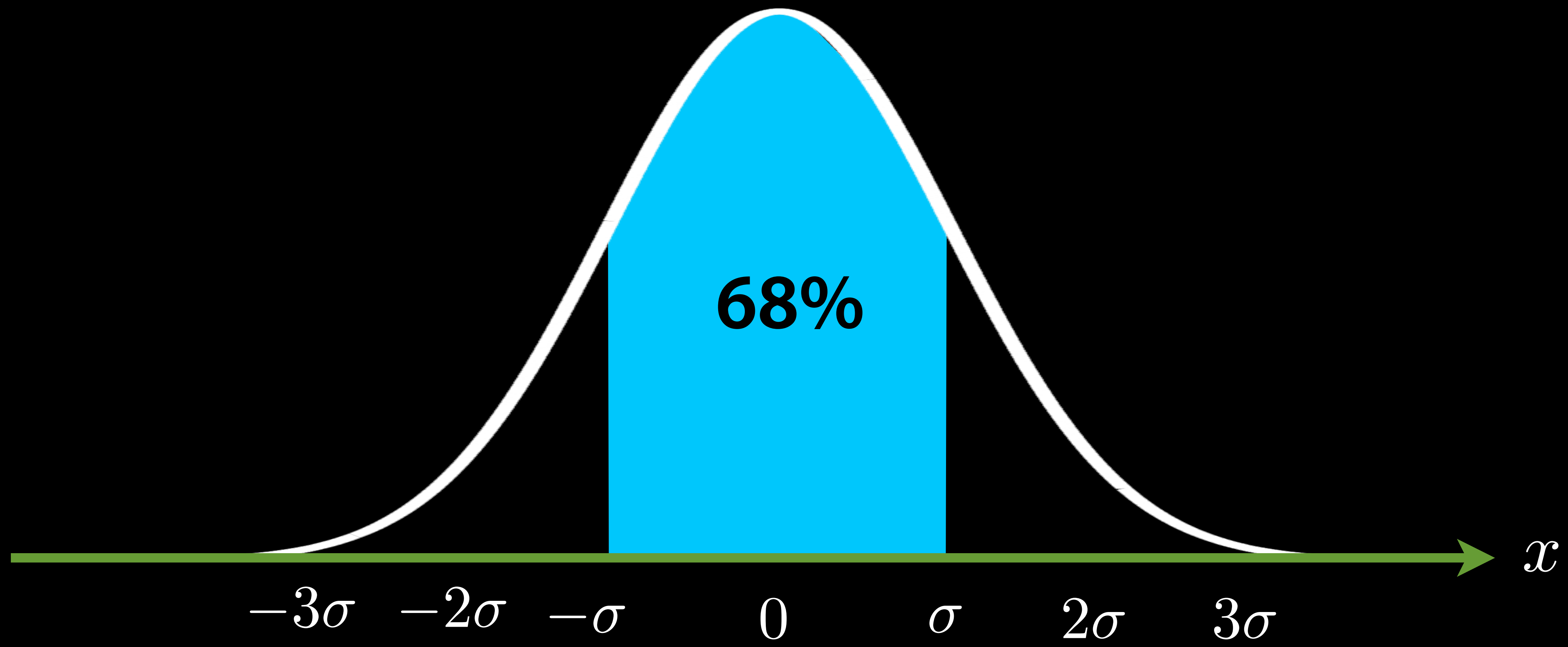
Gaussian filters have infinite support

Discrete filters use finite kernels

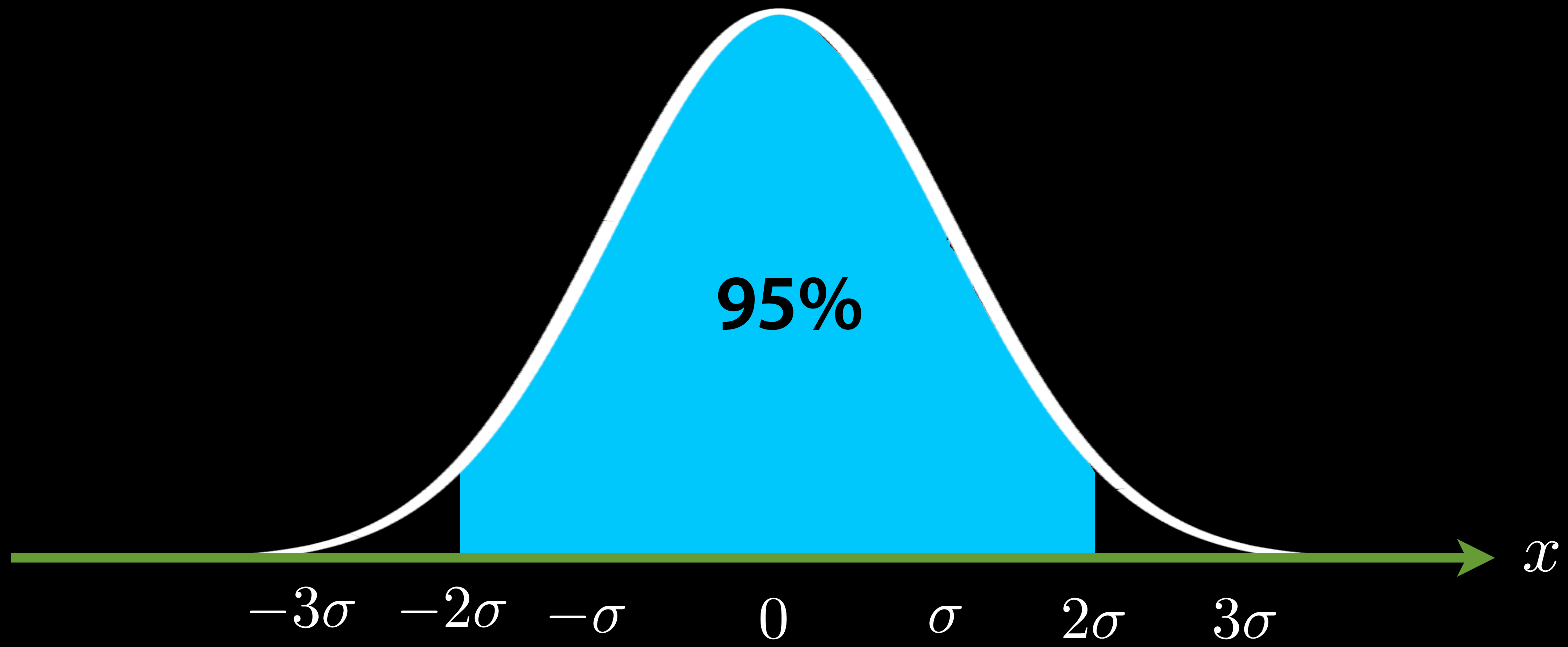
three-sigma rule



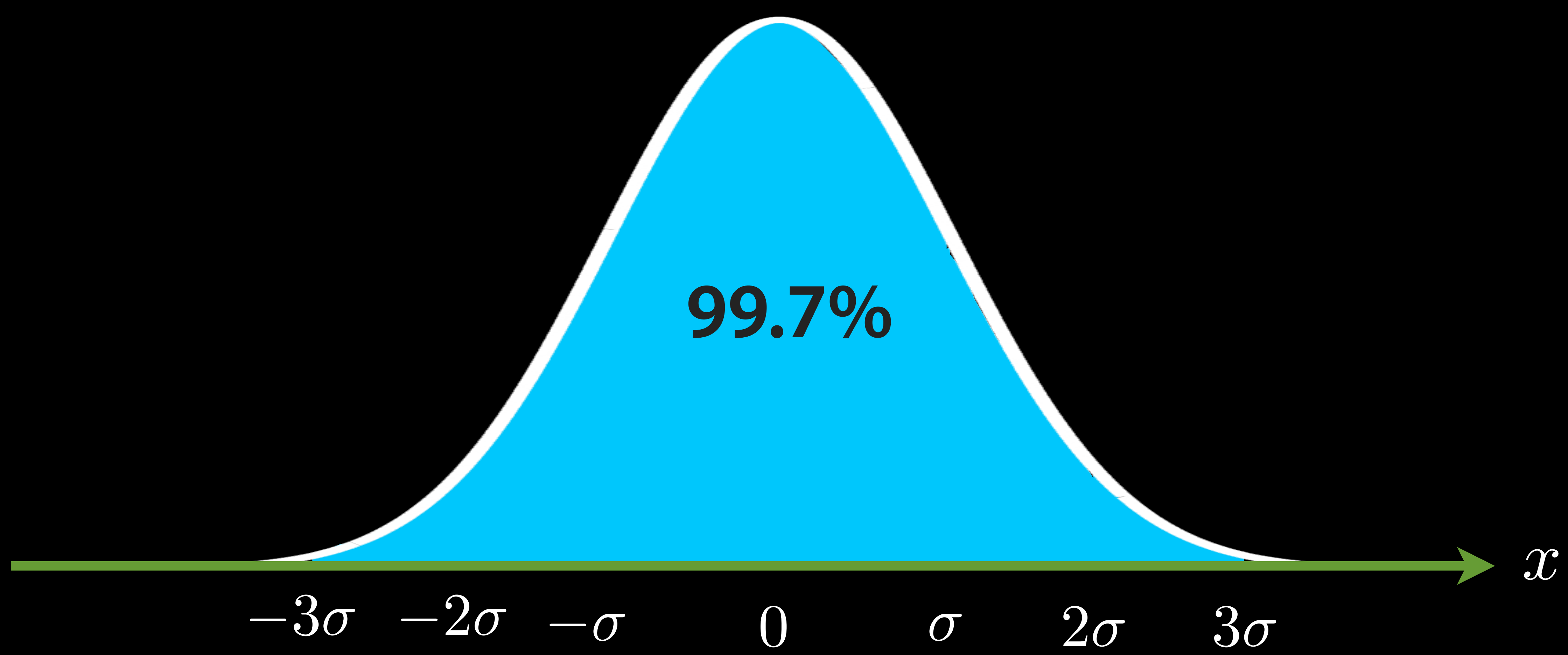
three-sigma rule



three-sigma rule



three-sigma rule




original


original
Gaussian filter


original


smooth by averaging


boundary
issues

boundary issues

boundary issues

boundary issues

clip filtering

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	0

0	0	0	0	0	0	0
0	3	3	2	1	0	0
0	0	0	1	3	1	0
0	3	1	2	2	3	0
0	2	0	0	2	2	0
0	2	0	0	0	0	0
0	0	0	0	0	0	0

boundary issues

clip filtering

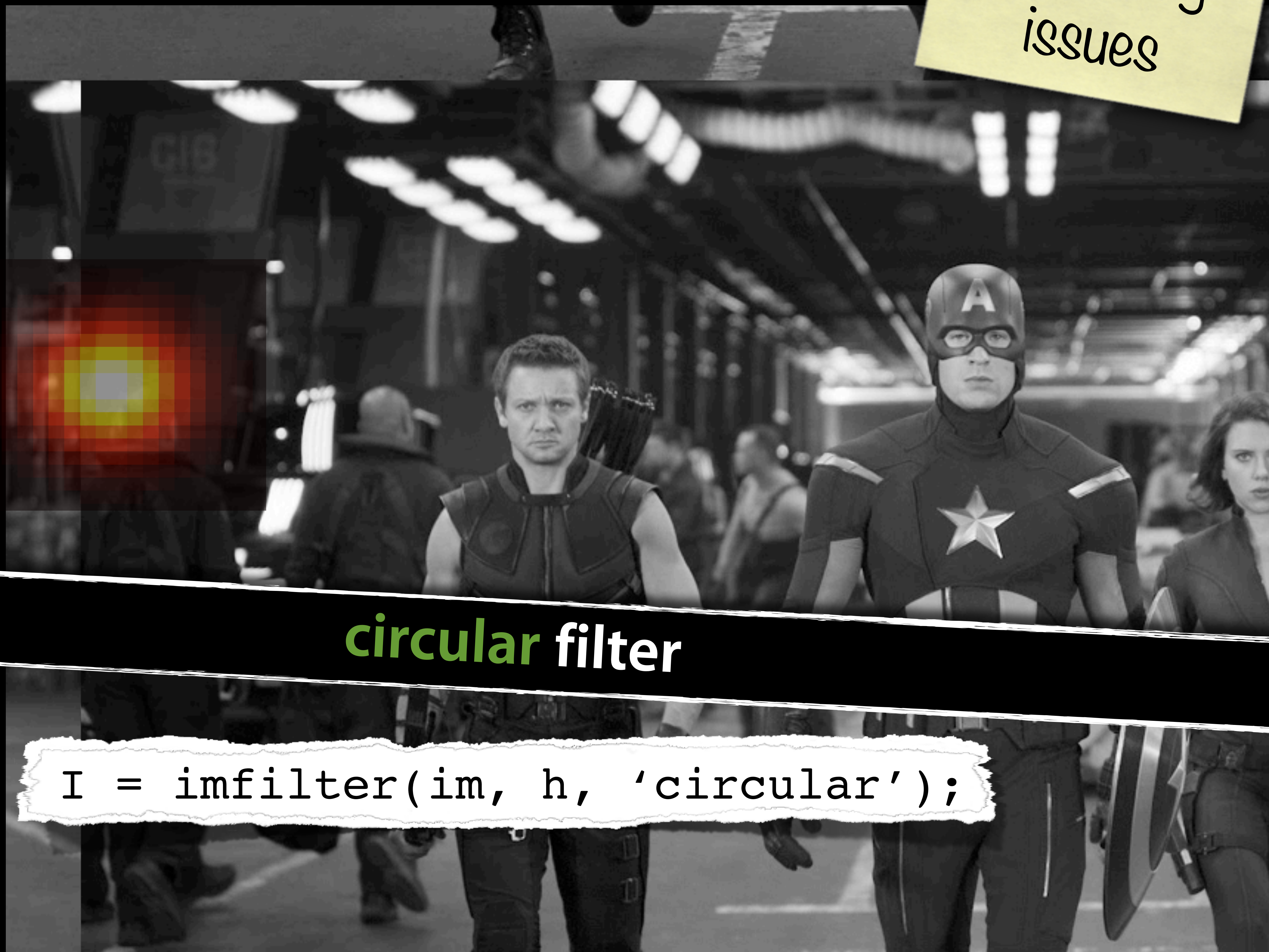
boundary issues

clip filtering

```
I = imfilter(im, h, 0);
```


boundary issues
circular filter

boundary issues

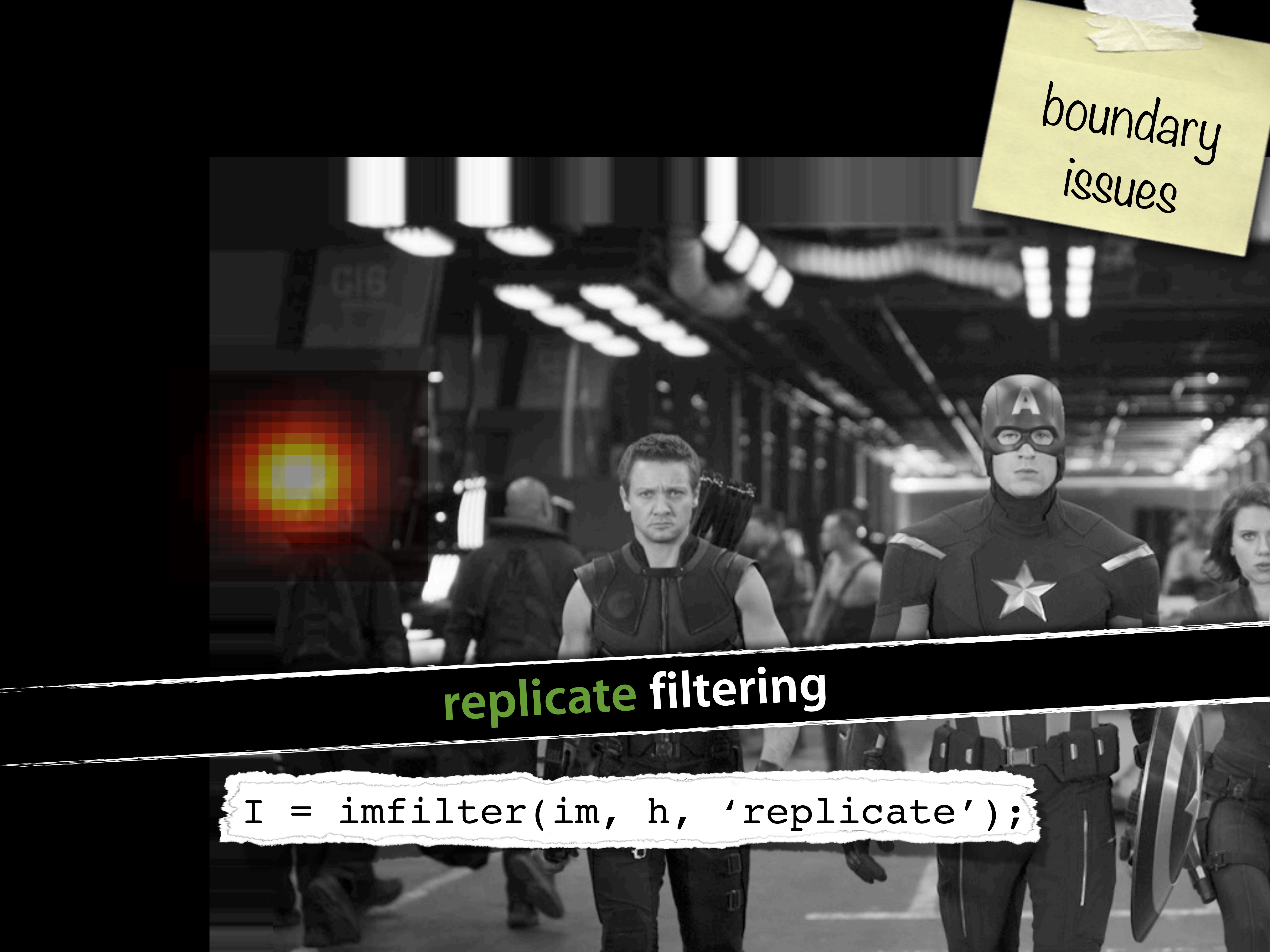


circular filter

```
I = imfilter(im, h, 'circular');
```

boundary issues

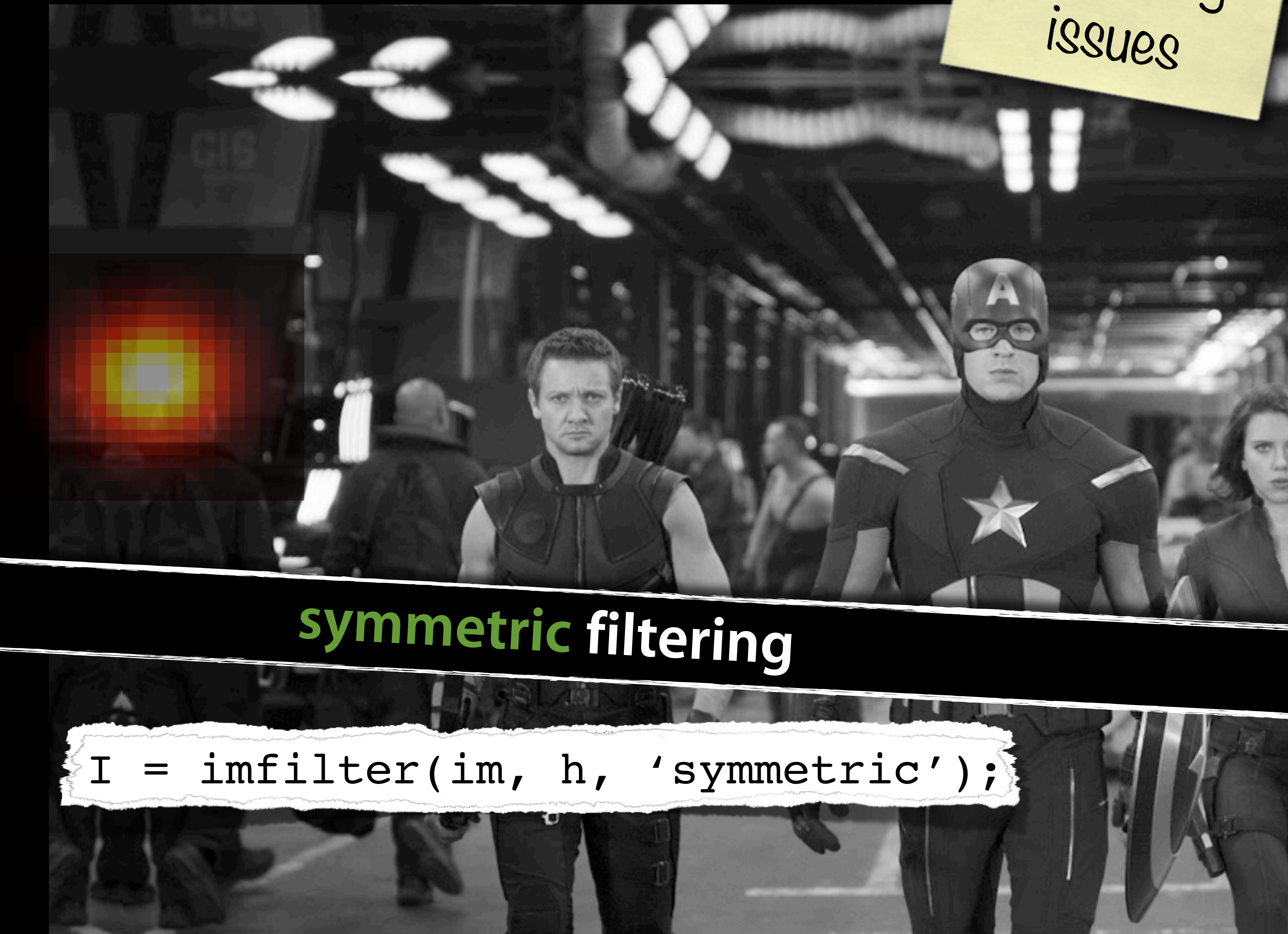
replicate filtering


boundary issues
replicate filtering
I = imfilter(im, h, 'replicate');

boundary issues

symmetric filtering

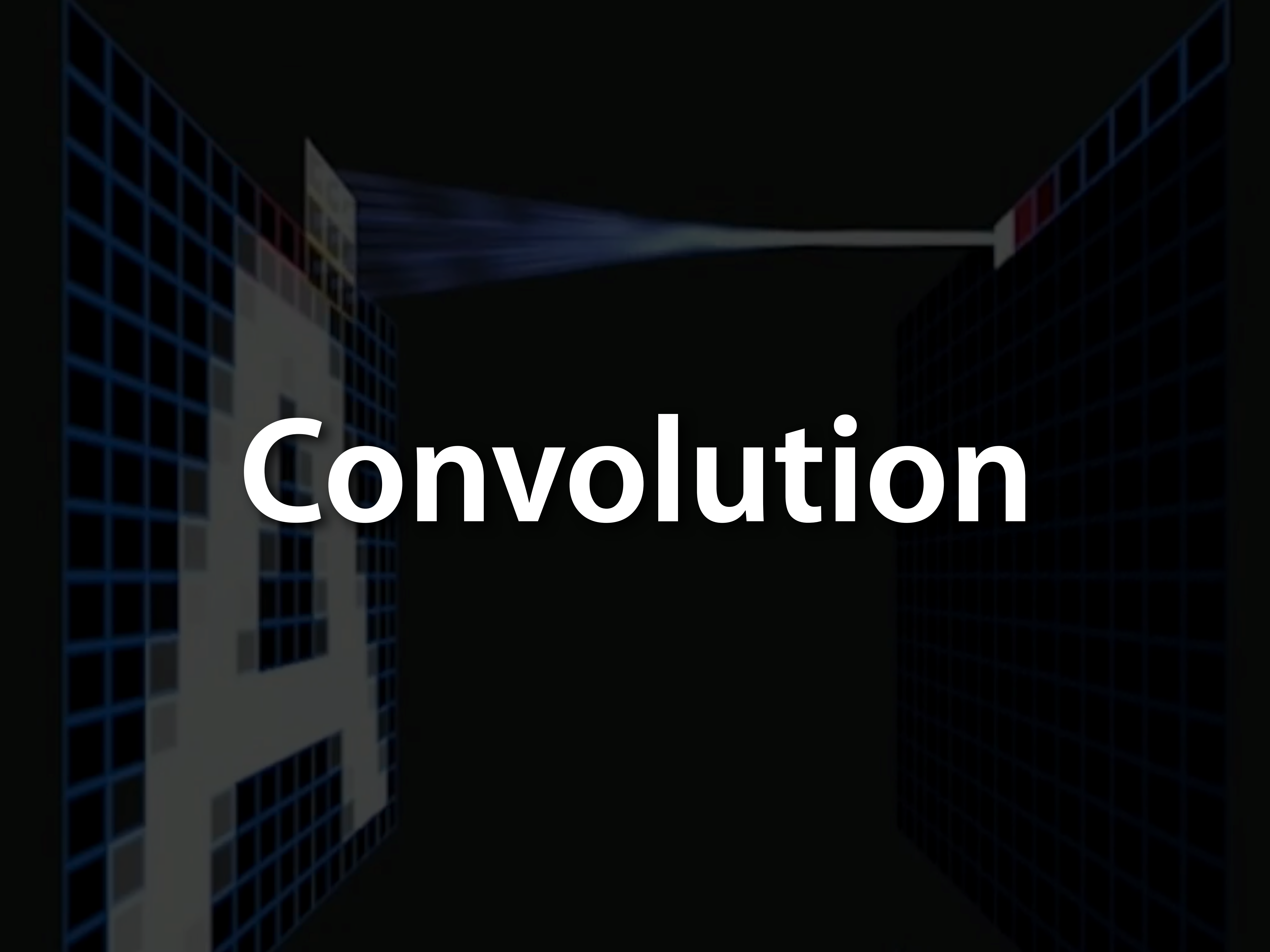
boundary issues



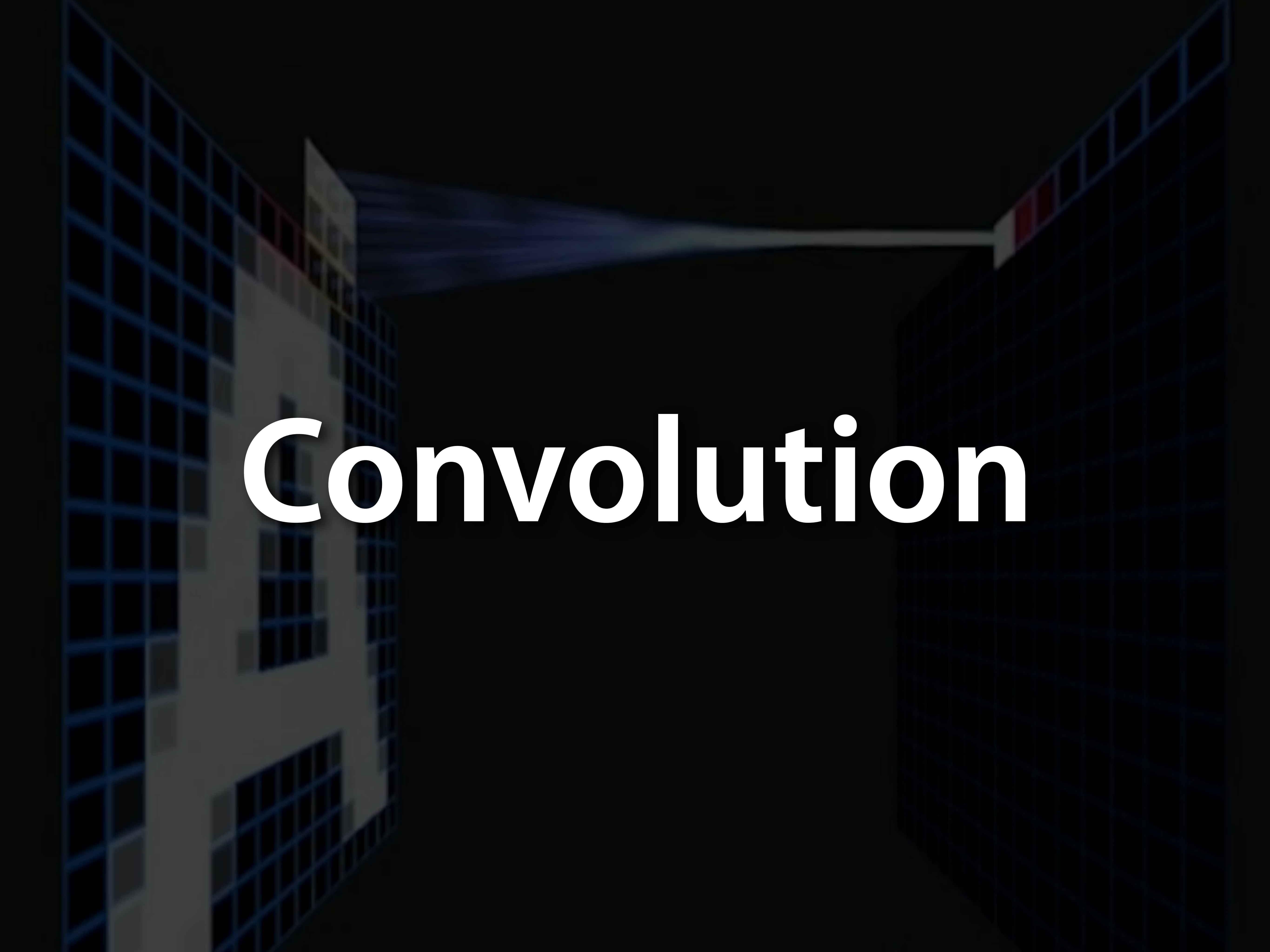
symmetric filtering

```
I = imfilter(im, h, 'symmetric');
```

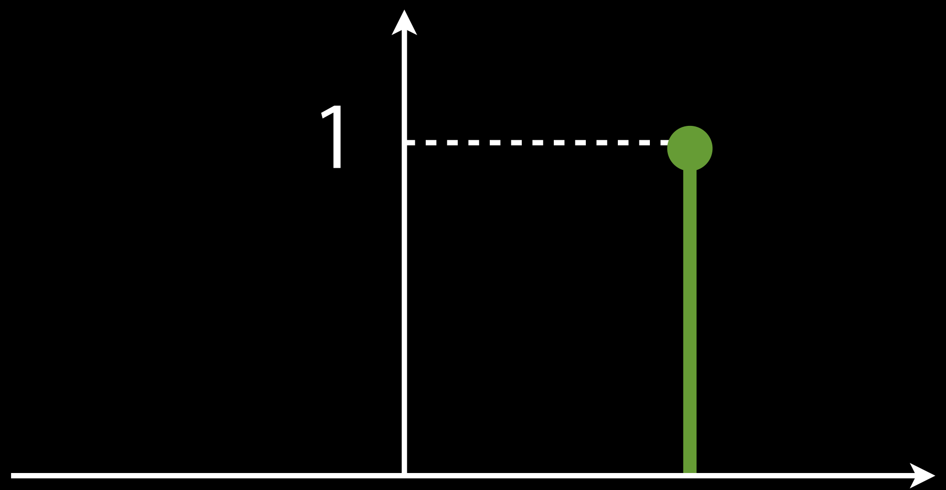
Convolution



Convolution

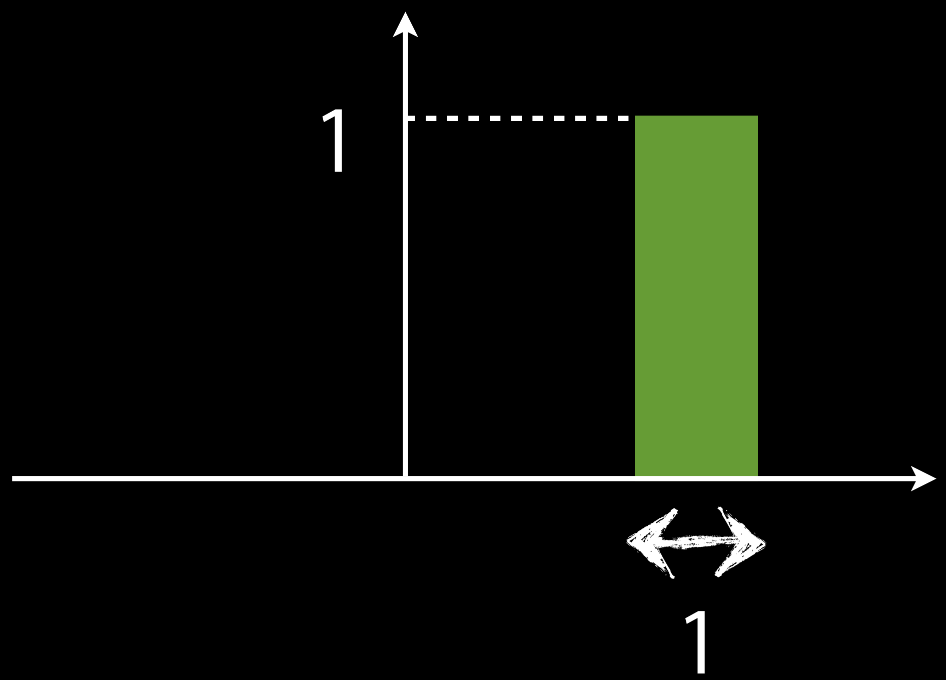


discrete
impulse

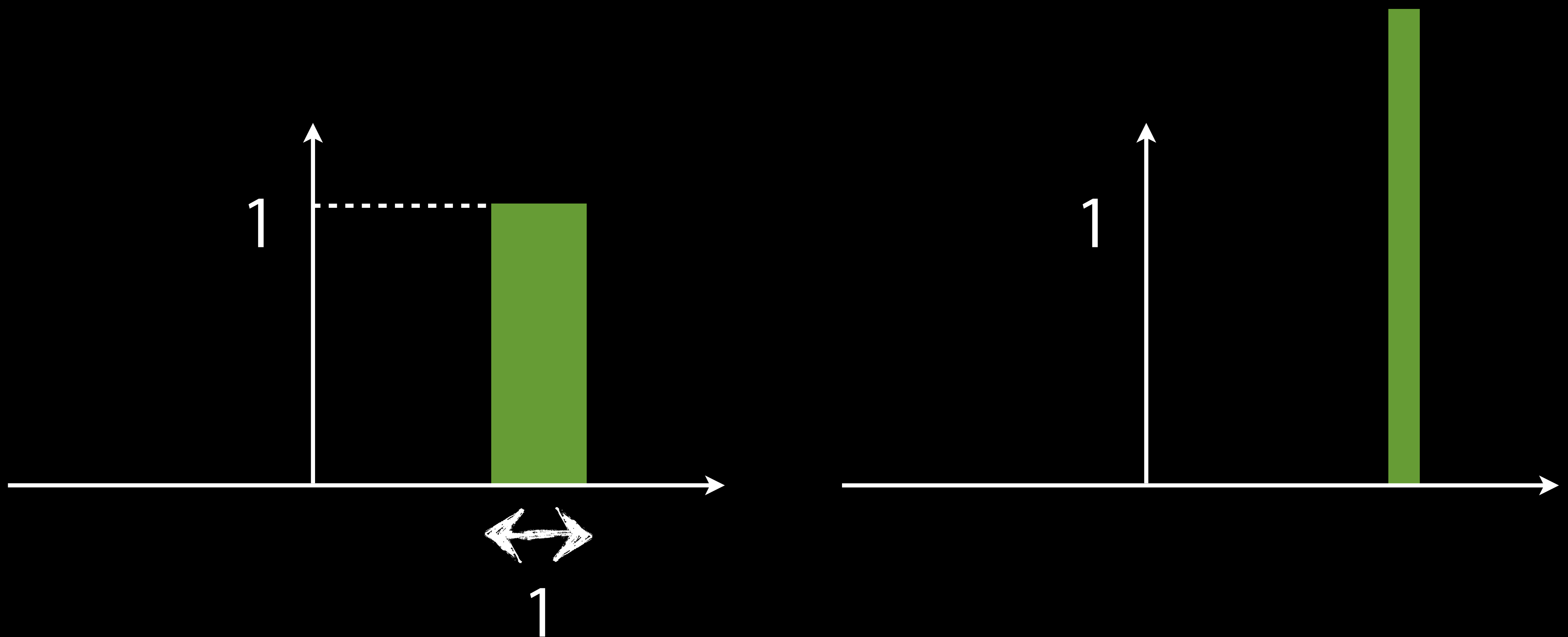


A function with a value of 1 at a single location

continuous impulse



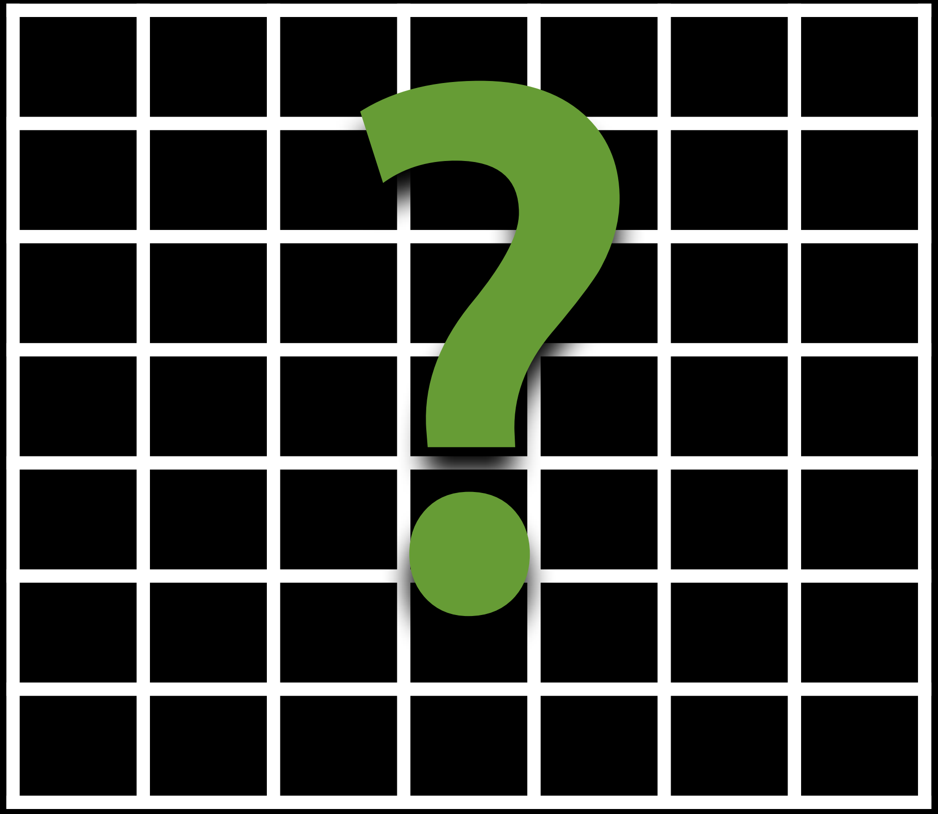
A function that is very narrow and very tall such that at the limit it has a unit area



A function that is very narrow and very tall such that at the limit it has a unit area

filtering an impulse

$$\underset{G[x,y]}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} \otimes \underset{F[x,y]}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}} =$$



$H[x,y]$

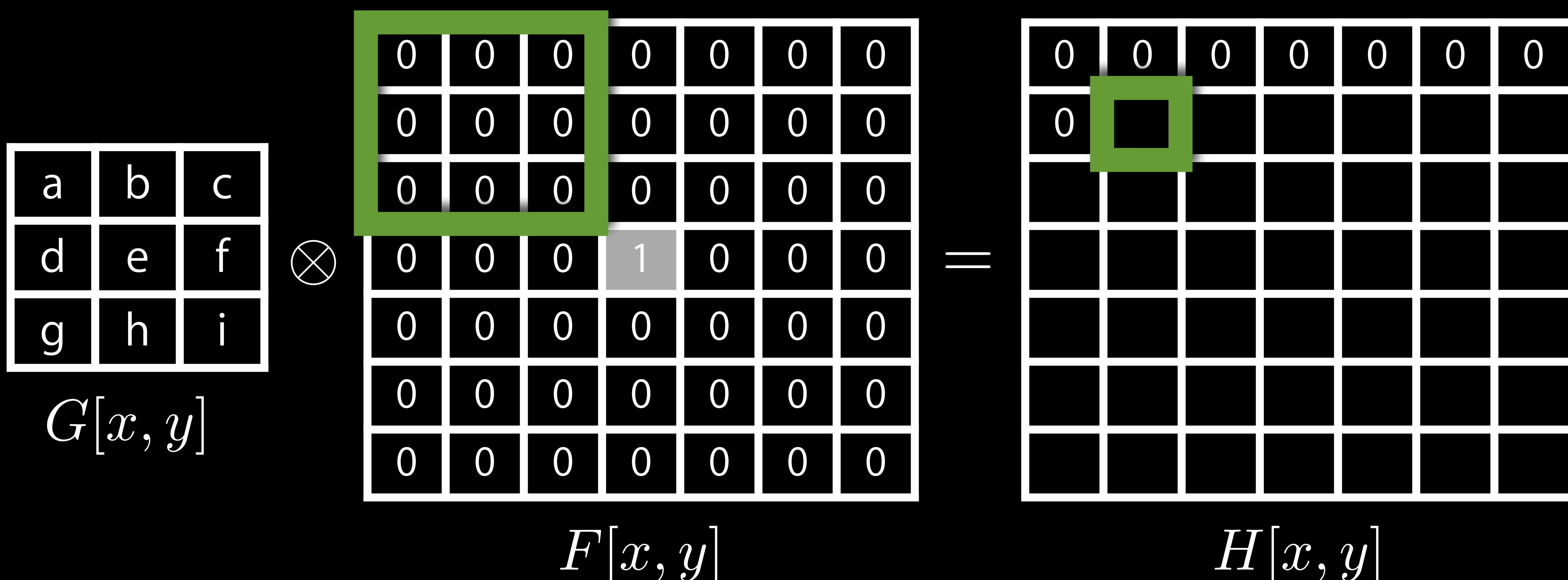
filtering an impulse

$$
\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}}_{G[x,y]}
\otimes
\underbrace{\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}_{F[x,y]}
=
\underbrace{\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & & & & & & \\
 & & & & & & \\
 & & & & & & \\
 & & & & & & \\
 & & & & & & \\
 & & & & & &
\end{bmatrix}}_{H[x,y]}
$$

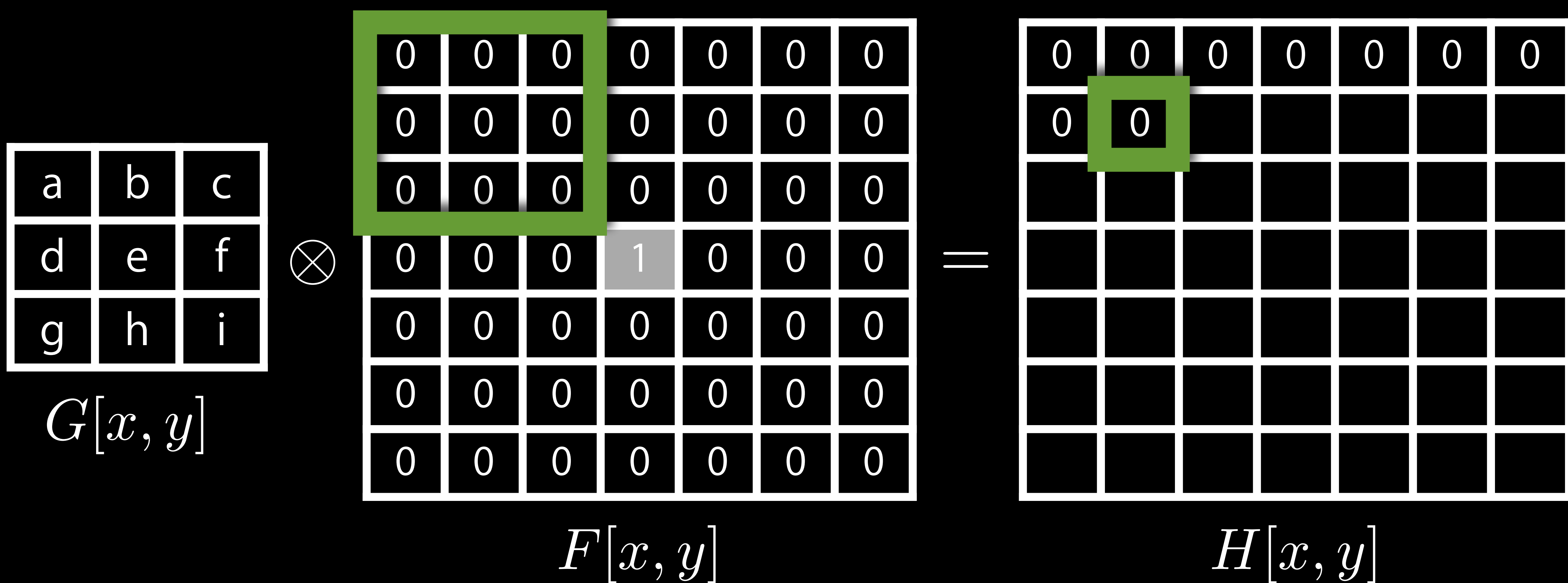
filtering an impulse

$$\underset{G[x,y]}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} \otimes \underset{F[x,y]}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}} = \underset{H[x,y]}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}}$$

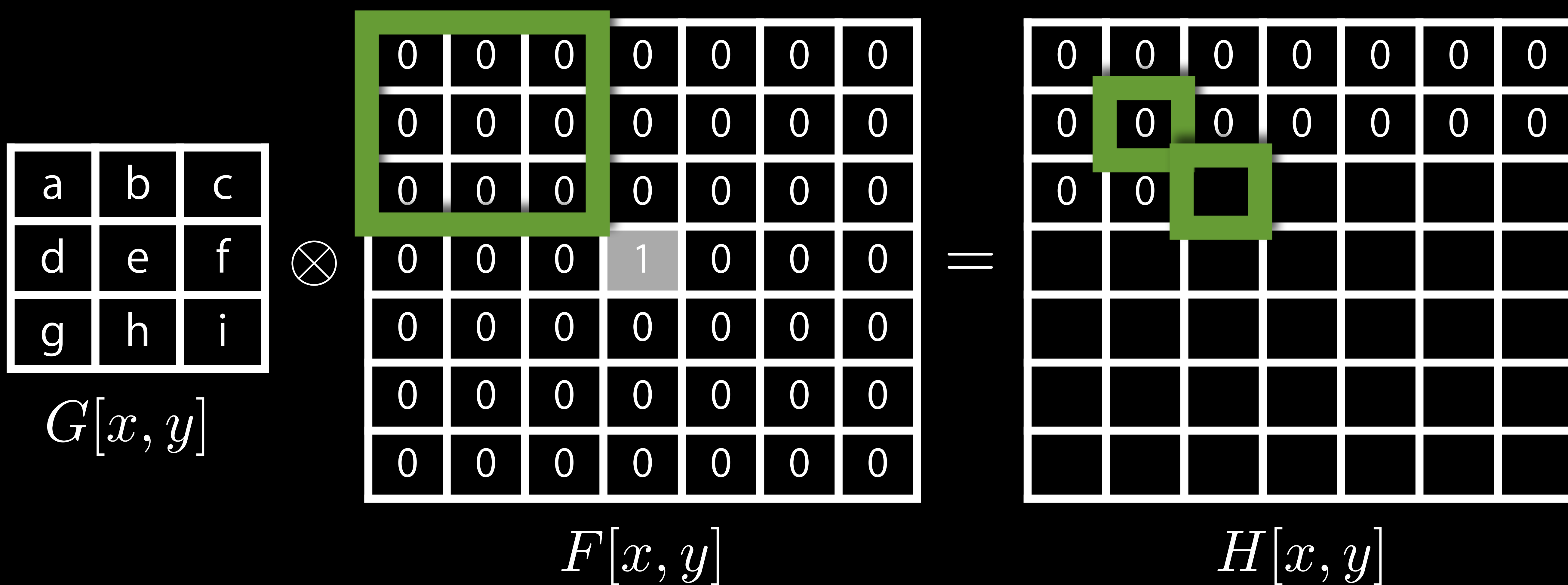
filtering
an impulse



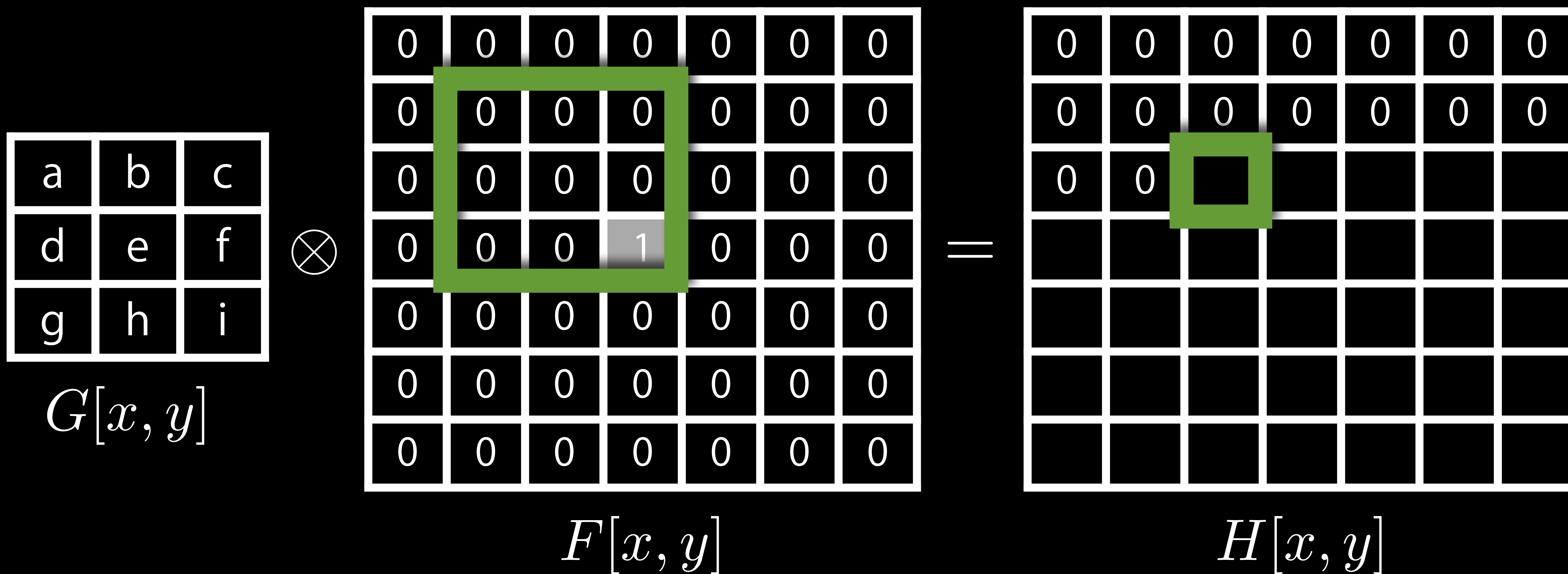
filtering an impulse



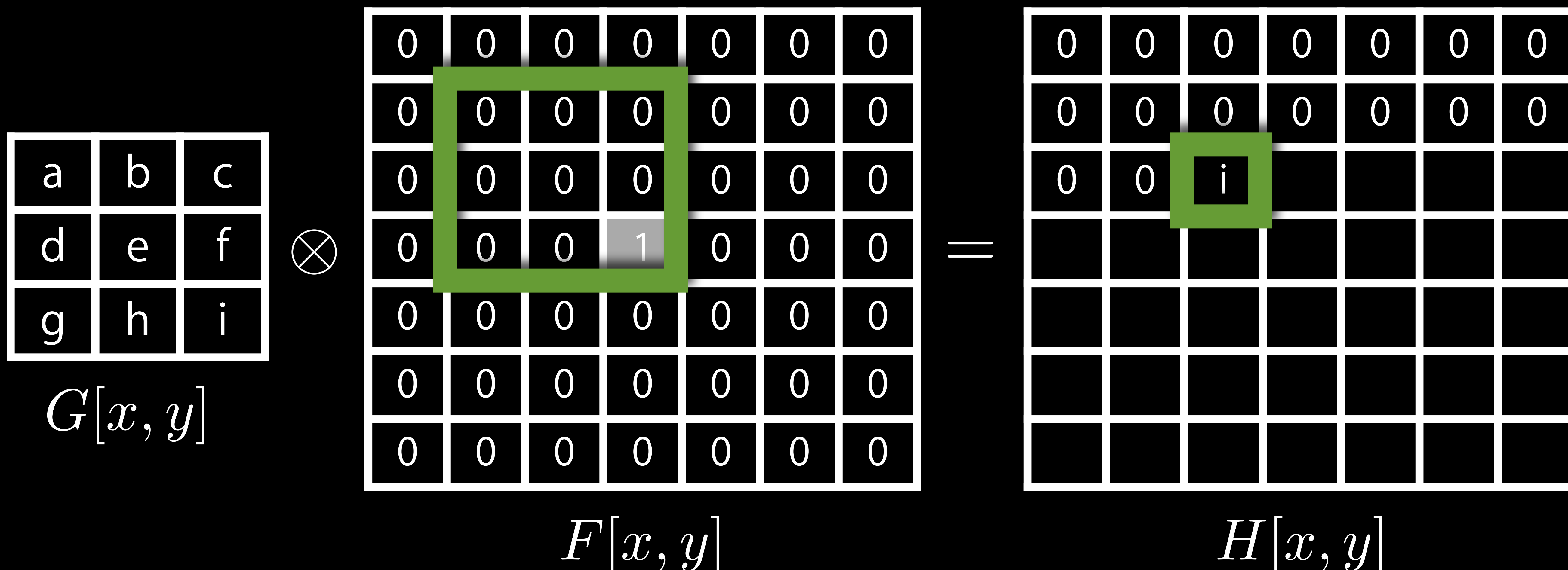
filtering
an impulse



filtering an impulse



filtering an impulse



filtering an impulse

a	b	c
d	e	f
g	h	i

$G[x, y]$

$\otimes$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

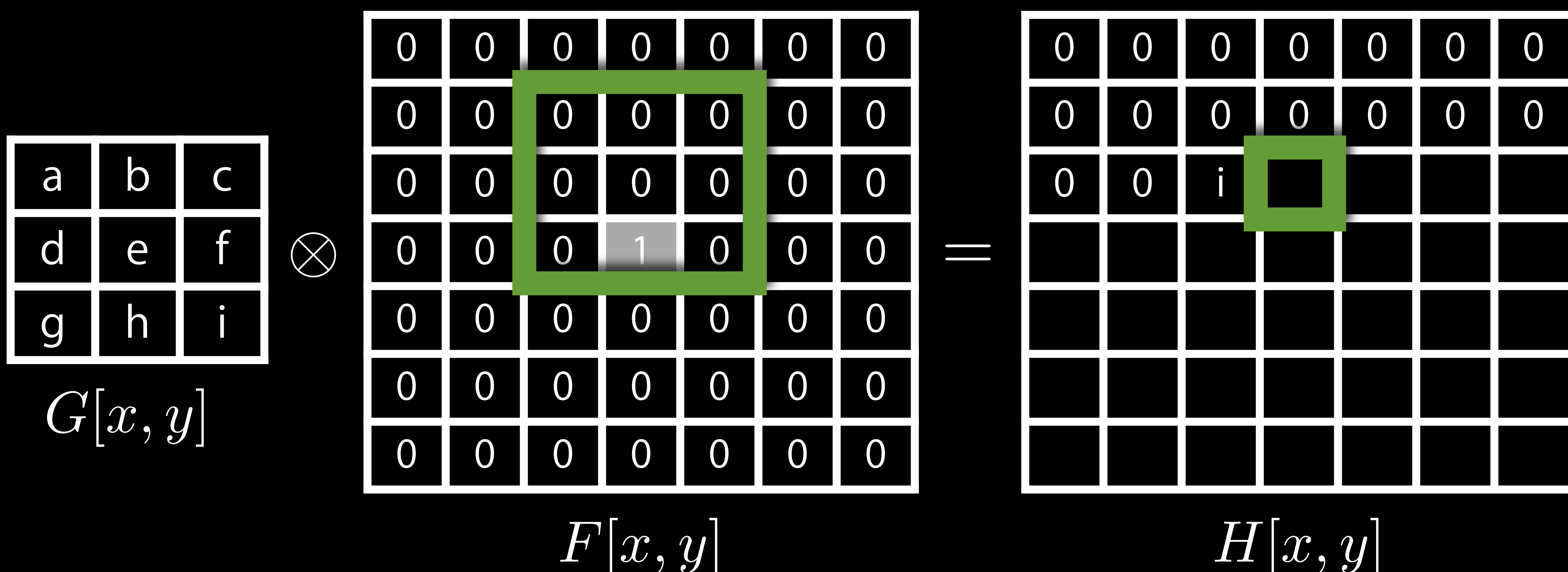
$F[x, y]$

$=$

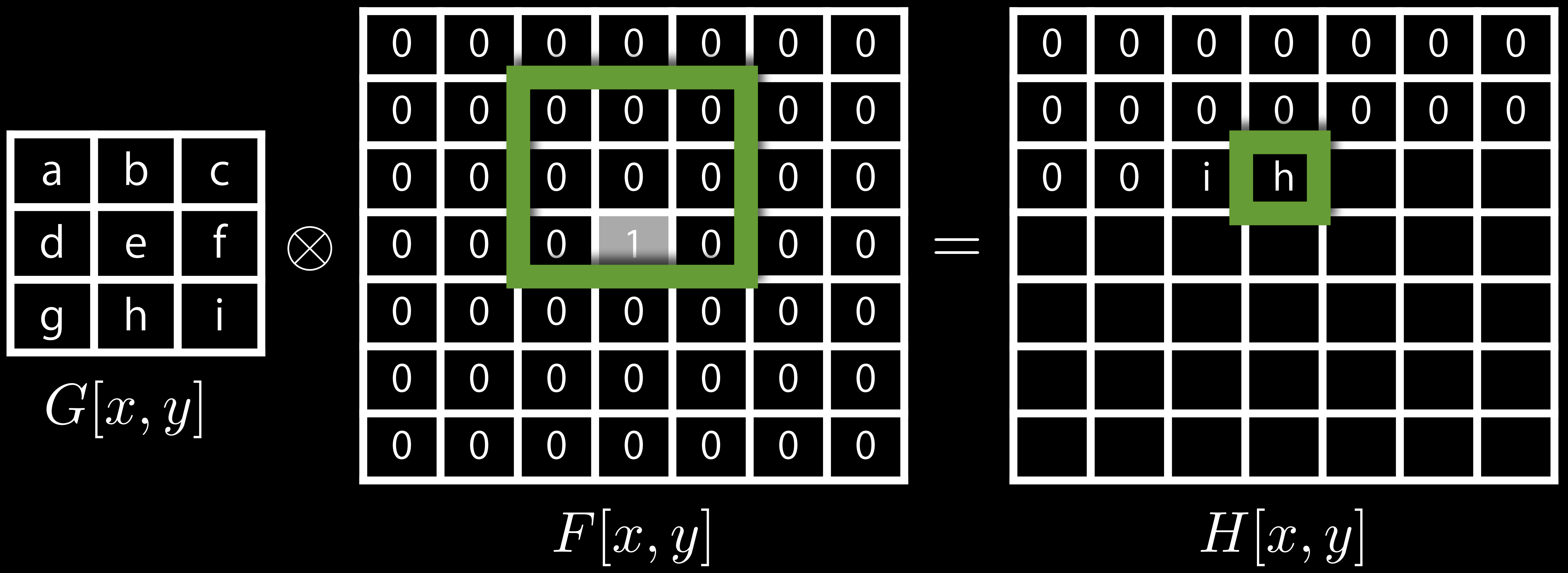
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	i				

$H[x, y]$

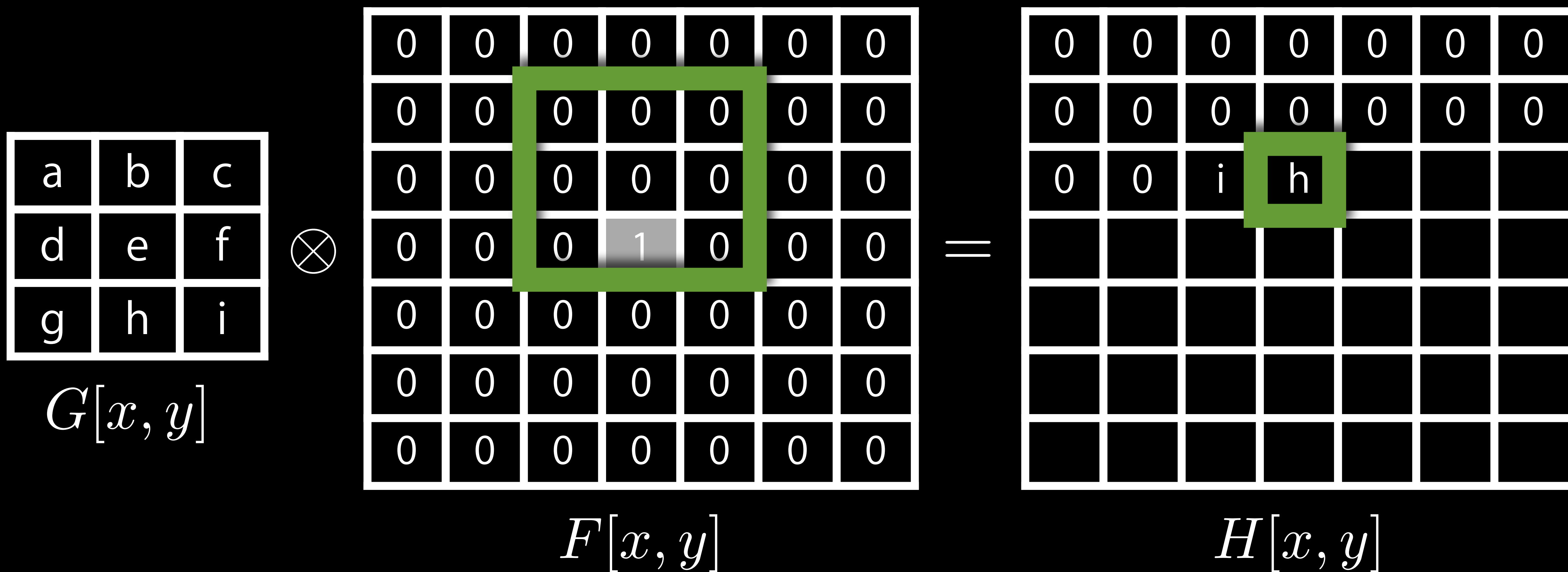
filtering an impulse



filtering an impulse



filtering an impulse



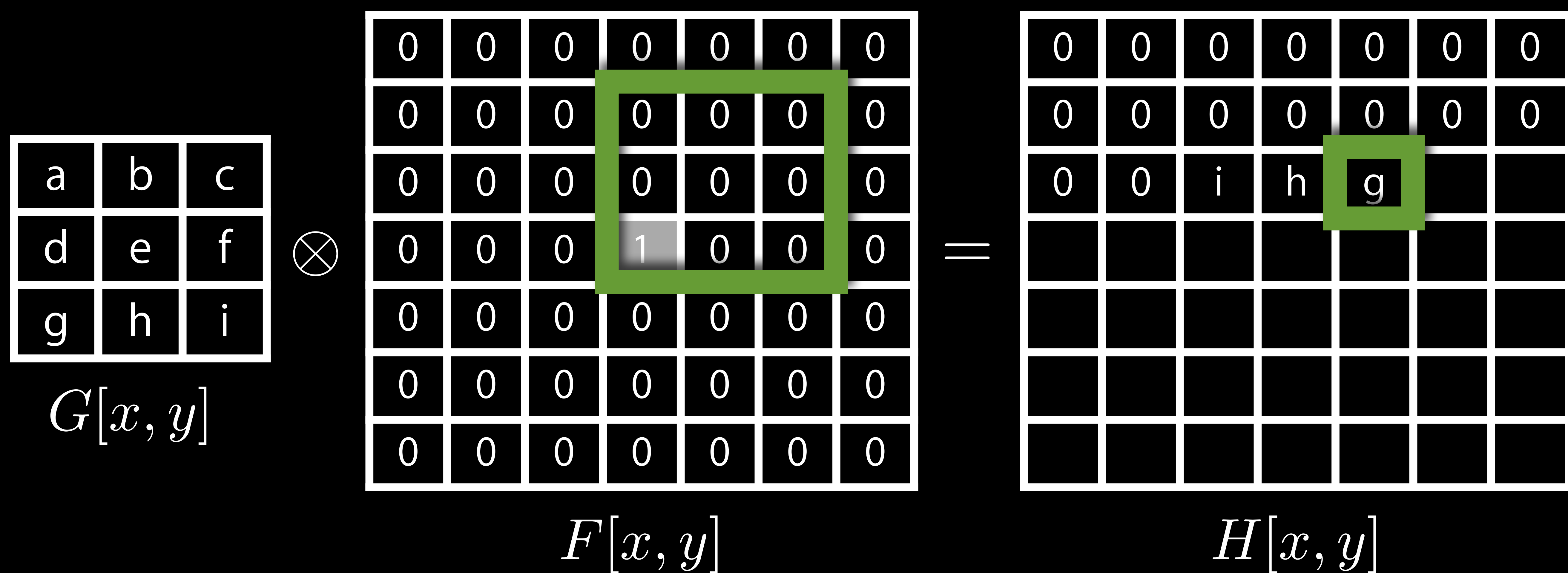
filtering
an impulse

$$\underset{G[x,y]}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} \otimes \underset{F[x,y]}{\begin{bmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{bmatrix}} = \underset{H[x,y]}{\begin{bmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&i&h&\boxed{}&&\\ &&&&&& \\ &&&&&& \\ &&&&&& \\ &&&&&& \end{bmatrix}}$$

filtering an impulse

$$\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}}_{G[x,y]} \otimes \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{F[x,y]} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & i & h & g & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}}_{H[x,y]}$$

filtering an impulse



filtering an impulse

$$\underset{G[x,y]}{\begin{array}{|c|c|c|}\hline a & b & c \\ \hline d & e & f \\ \hline g & h & i \\ \hline\end{array}} \otimes \underset{F[x,y]}{\begin{array}{|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0\\ \hline 0&0&0&1&0&0&0\\ \hline 0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0\\ \hline\end{array}} = \underset{H[x,y]}{\begin{array}{|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0\\ \hline 0&0&0&0&0&0&0\\ \hline 0&0&i&h&g&0&0\\ \hline 0&0&&&&&\\ \hline &&&&&&\\ \hline &&&&&&\\ \hline &&&&&&\\ \hline\end{array}}$$

filtering
an impulse

a	b	c
d	e	f
g	h	i

$G[x,y]$

$\otimes$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

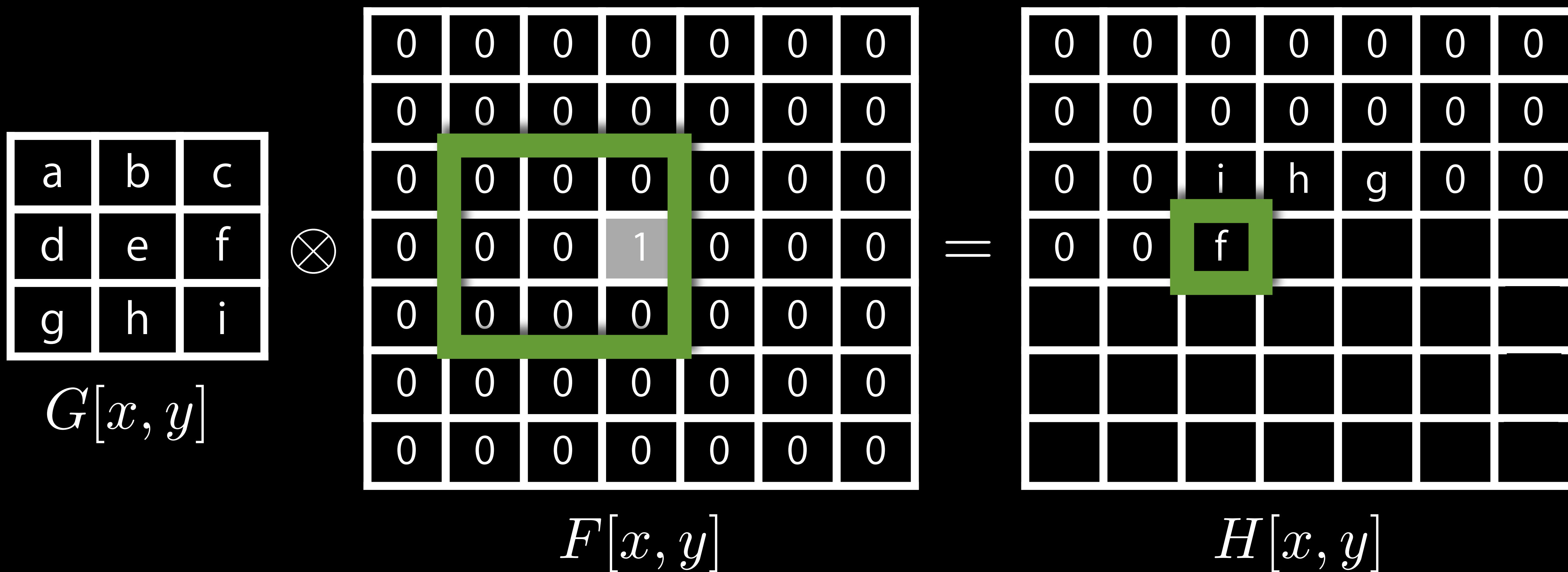
$F[x,y]$

$=$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	i	h	g	0	0
0	0	f				

$H[x,y]$

filtering an impulse



filtering an impulse

$$
\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}}_{G[x,y]}
\otimes
\underbrace{\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}_{F[x,y]}
=
\underbrace{\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & i & h & g & 0 & 0 \\
0 & 0 & f & e & d & 0 & 0 \\
0 & 0 & c & b & a & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}_{H[x,y]}
$$

filtering an impulse

$$\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}}_{G[x,y]} \otimes \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{F[x,y]} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & i & h & g & 0 & 0 \\ 0 & 0 & f & e & d & 0 & 0 \\ 0 & 0 & c & b & a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{H[x,y]}$$

Filter output is REVERSED!!!

filtering
an impulse

a	b	c
d	e	f
g	h	i

$G[x, y]$

$\otimes$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

$F[x, y]$

$=$

$H[x, y]$

filtering an impulse

$$\underset{G[x,y]}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} \otimes \underset{F[x,y]}{\begin{bmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{bmatrix}} = \underset{H[x,y]}{\begin{bmatrix} \, & \, & \, & \, & \, & \, & \, \\ \, & \, & \, & \, & \, & \, & \, \\ \, & \, & \, & \, & \, & \, & \, \\ \, & \, & \, & \, & \, & \, & \, \\ \, & \, & \, & \, & \, & \, & \, \\ \, & \, & \, & \, & \, & \, & \, \\ \, & \, & \, & \, & \, & \, & \, \end{bmatrix}}$$

How would you modify the kernel to avoid the flip?

filtering an impulse

$$\underset{G[x,y]}{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}} \otimes \underset{F[x,y]}{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}} = \underset{H[x,y]}{\begin{bmatrix} \; & \; & \; & \; & \; & \; & \; \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}}$$

How would you modify the kernel to avoid the flip?

Flip the kernel

filtering
an impulse

$$
\begin{bmatrix} c & b & a \\ f & e & d \\ i & h & g \end{bmatrix}
\otimes
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
=
\begin{bmatrix}
\; & \; & \; & \; & \; & \; & \; \\
\; & \; & \; & \; & \; & \; & \; \\
\; & \; & \; & \; & \; & \; & \; \\
\; & \; & \; & \; & \; & \; & \; \\
\; & \; & \; & \; & \; & \; & \; \\
\; & \; & \; & \; & \; & \; & \; \\
\; & \; & \; & \; & \; & \; & \;
\end{bmatrix}
$$

$G[x, y]$ $\quad$ $F[x, y]$ $\quad$ $H[x, y]$

How would you modify the kernel to avoid the flip?

Flip the kernel

filtering
an impulse

i	h	g
f	e	d
c	b	a

$G[x, y]$

$\otimes$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

$F[x, y]$

$=$

$H[x, y]$

How would you modify the kernel to avoid the flip?

Flip the kernel

filtering
an impulse

i	h	g
f	e	d
c	b	a

$G[x, y]$

$\otimes$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

$F[x, y]$

$=$

		a	b	c		
		d	e	f		
		g	h	i		

$H[x, y]$

How would you modify the kernel to avoid the flip?

Flip the kernel

Let the filter window size be $(2K+1) \times (2K+1)$

Let the filter window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v] F[x-u, y-v]$$

Let the filter window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x-u,y-v]$$

This is called convolution, denoted $H = G * F$

Let the filter window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v] F[x-u, y-v]$$

This is called convolution, denoted $H = G * F$

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v]\, G[u,v]$$

This is called cross correlation, denoted $H = G \otimes F$

Let the averaging window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} F[x+u, y+v] \, G[u,v]$$

This is called cross correlation, denoted $H = G \otimes F$

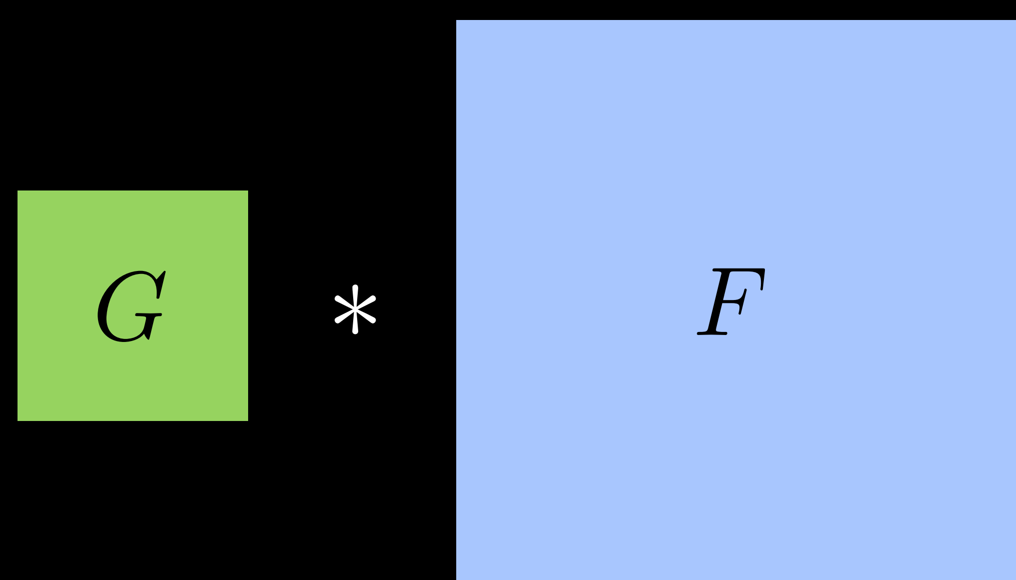
Let the filter window size be $(2K+1) \times (2K+1)$

$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v] F[x-u, y-v]$$

This is called convolution, denoted $H = G * F$

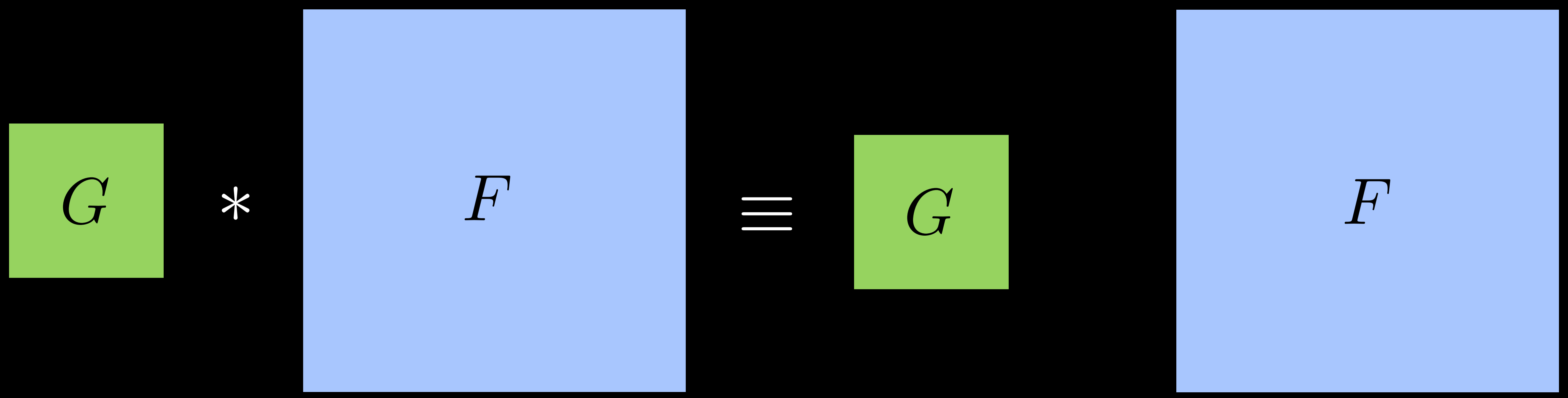
$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x-u, y-v]$$

This is called convolution, denoted $H = G * F$



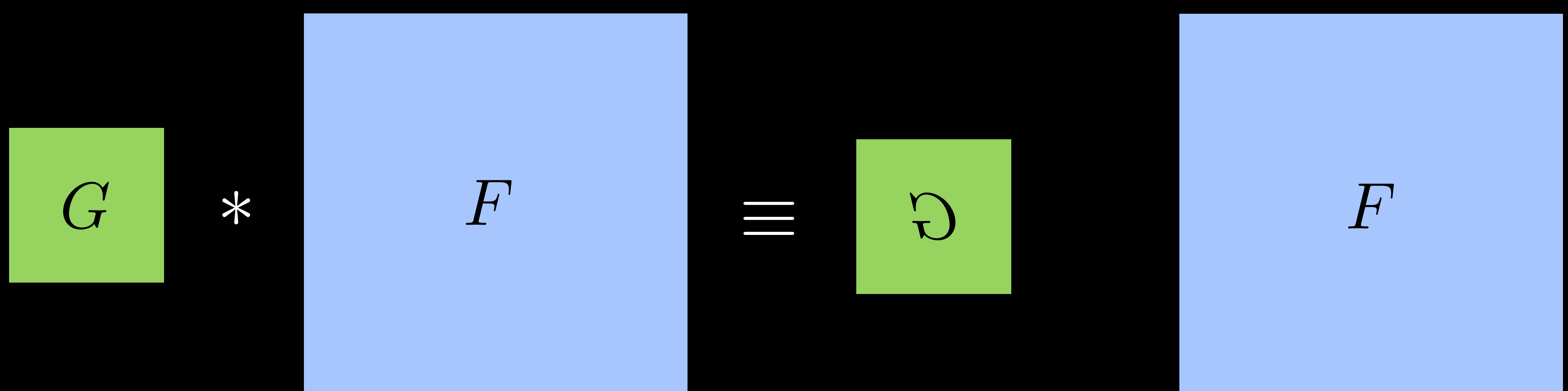
$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x-u, y-v]$$

This is called convolution, denoted $H = G * F$



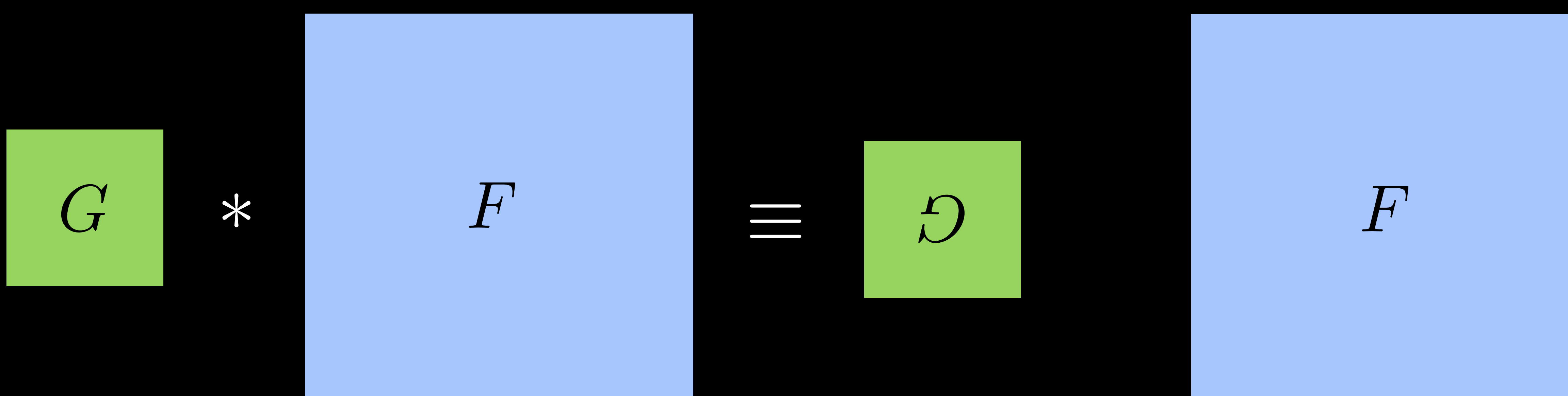
$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x-u, y-v]$$

This is called convolution, denoted $H = G * F$



$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v] F[x-u, y-v]$$

This is called convolution, denoted $H = G * F$



$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x-u,y-v]$$

This is called convolution, denoted $H = G * F$