SSY097 - Image Analysis

Lecture 2 - Filtering, gradients and scale

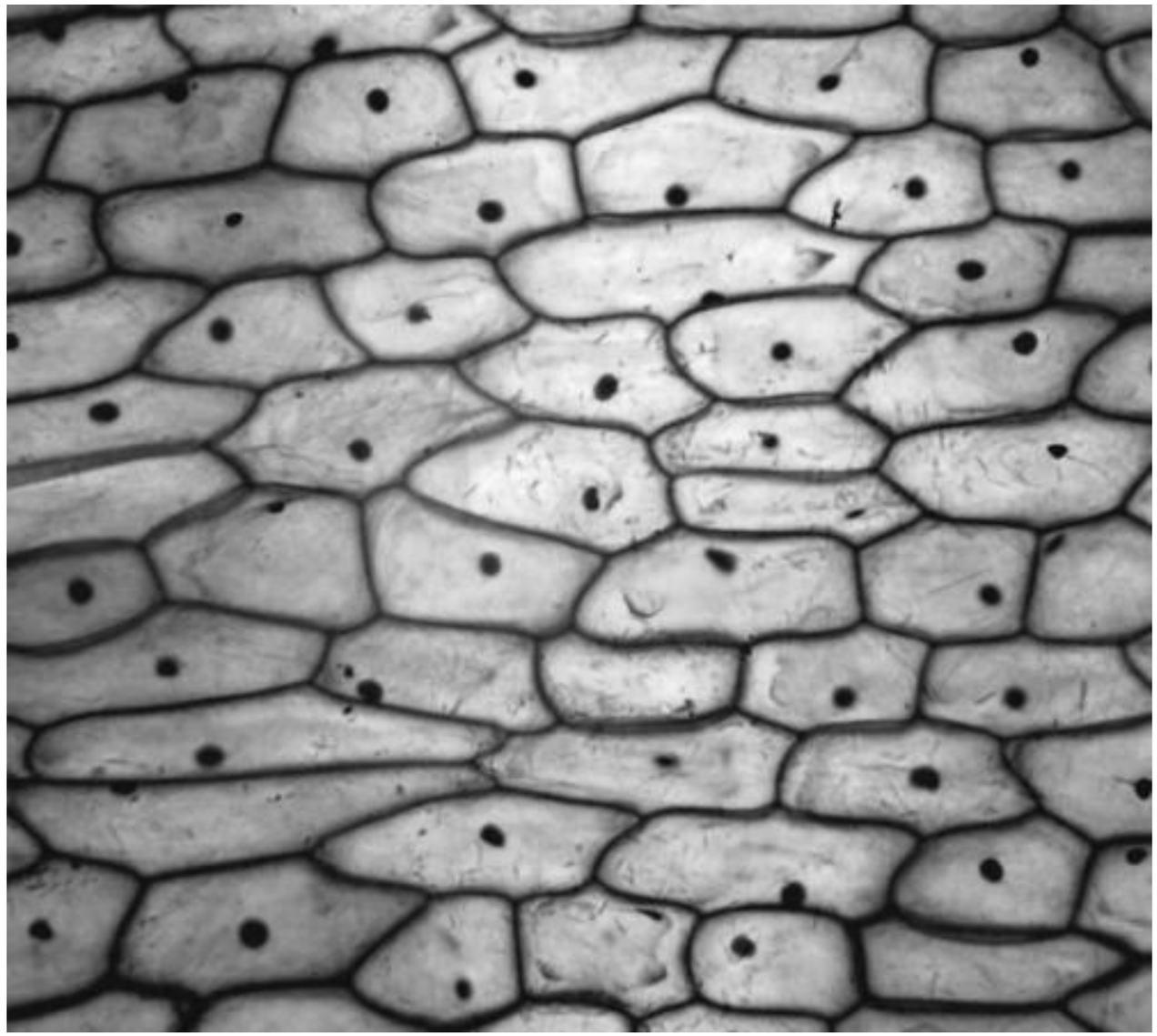
Torsten Sattler
(slides adapted from Olof Enqvist)

Last Lecture

Jan. 20	**Introduction, Linear classifiers and filtering**	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	Local features	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	Convolutional neural networks	
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	

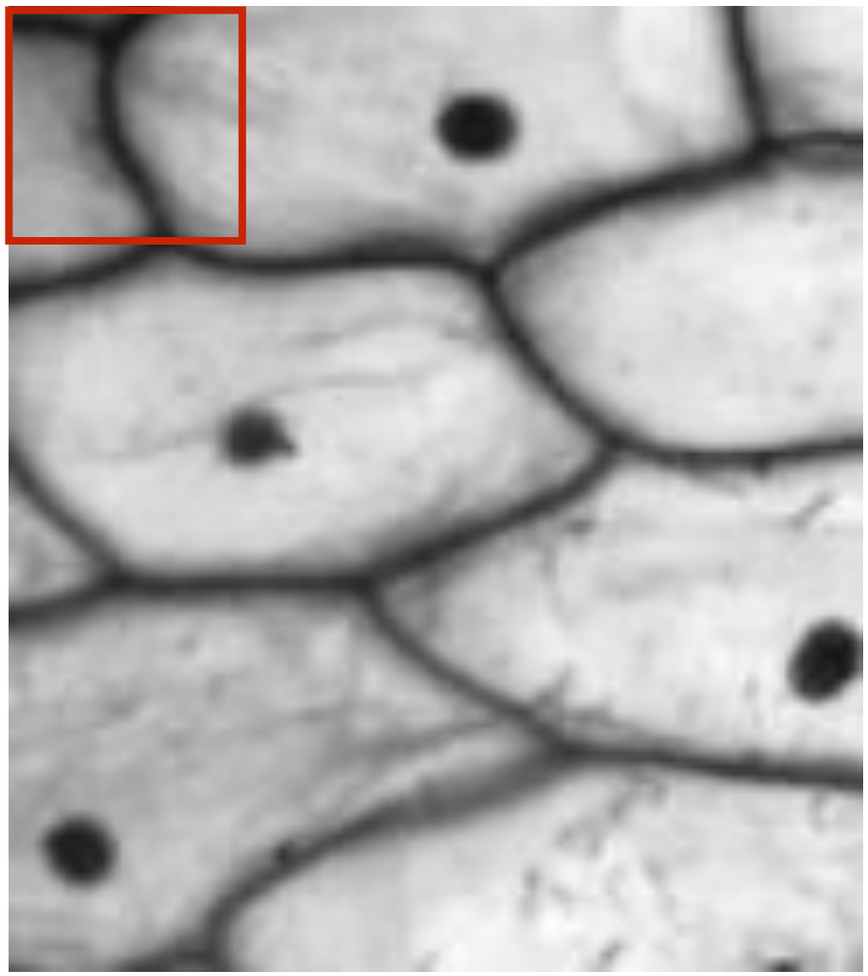
Last Lecture

Application: Counting of objects (nuclei)



Last Lecture

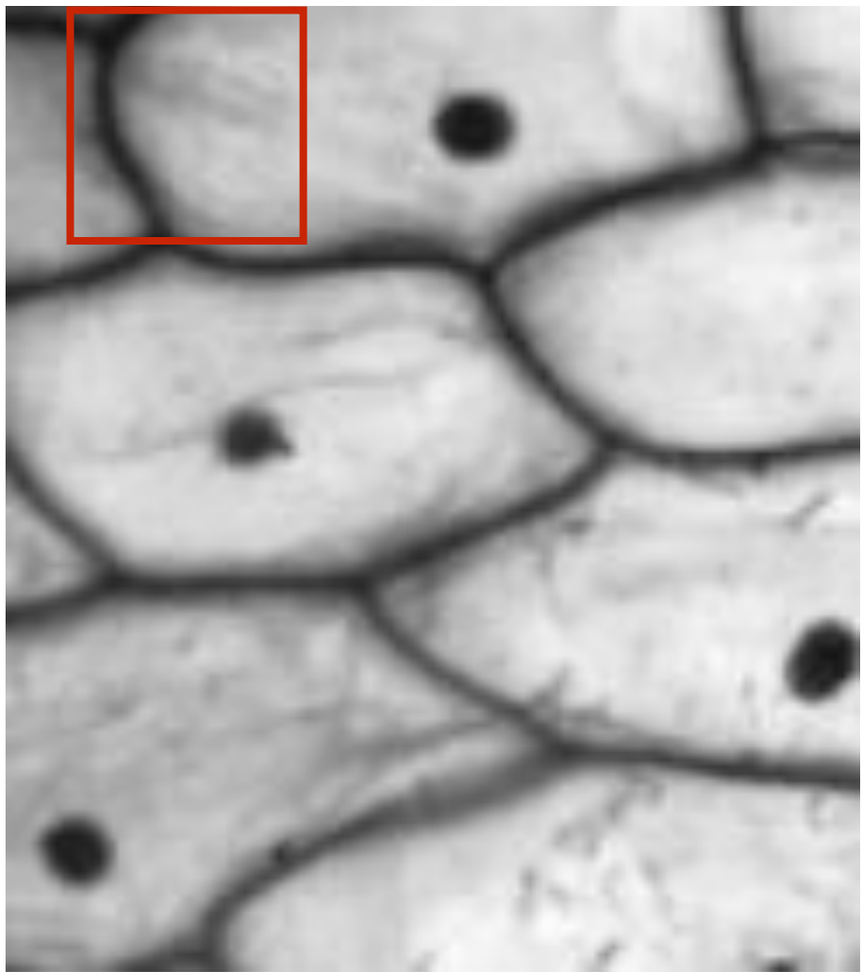
Sliding window classification



$$\cdot w < \tau$$

Last Lecture

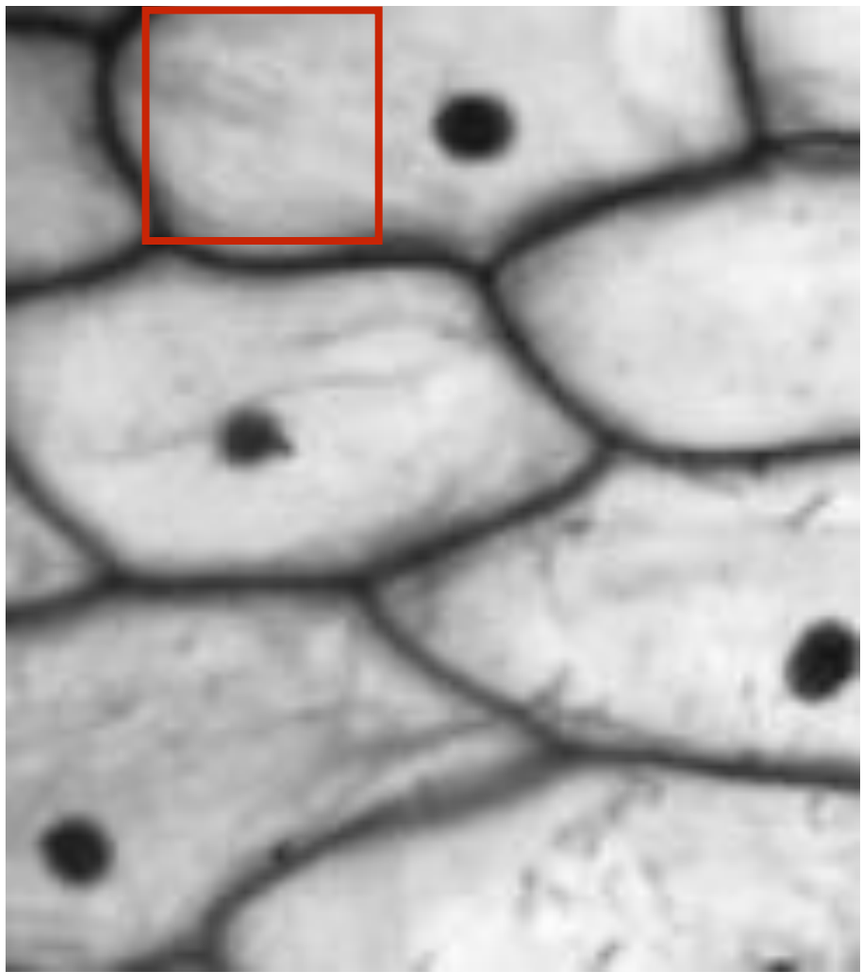
Sliding window classification



$\cdot\, w < \tau$

Last Lecture

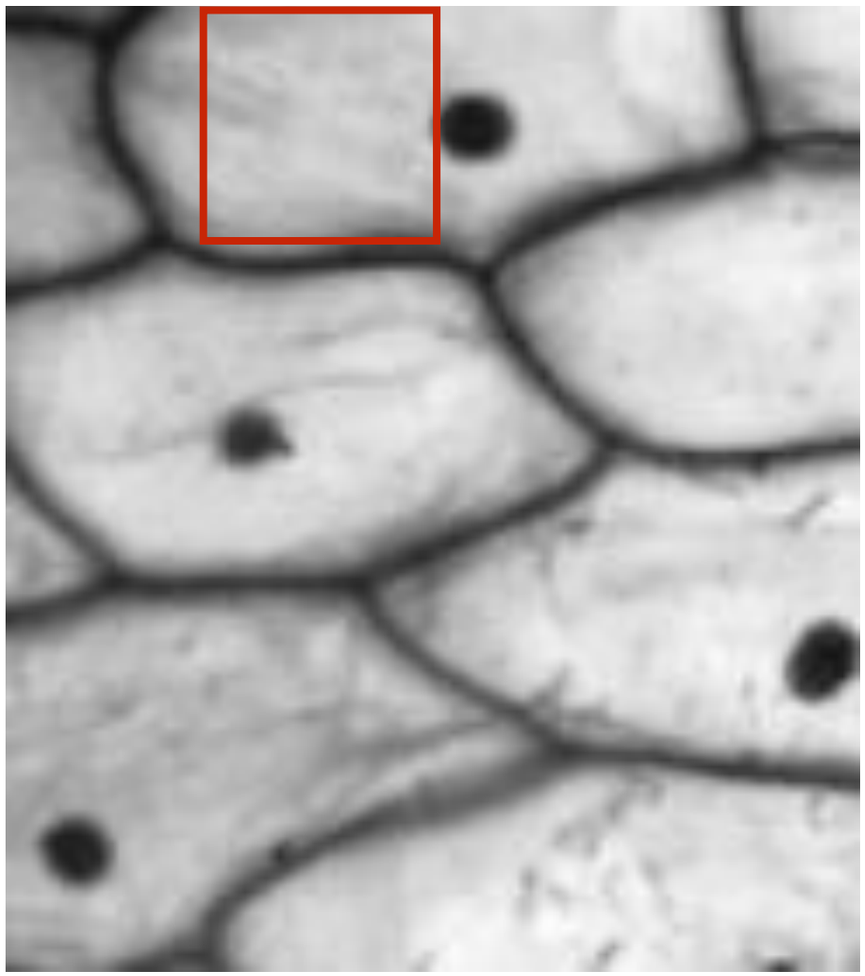
Sliding window classification



$\cdot\, w < \tau$

Last Lecture

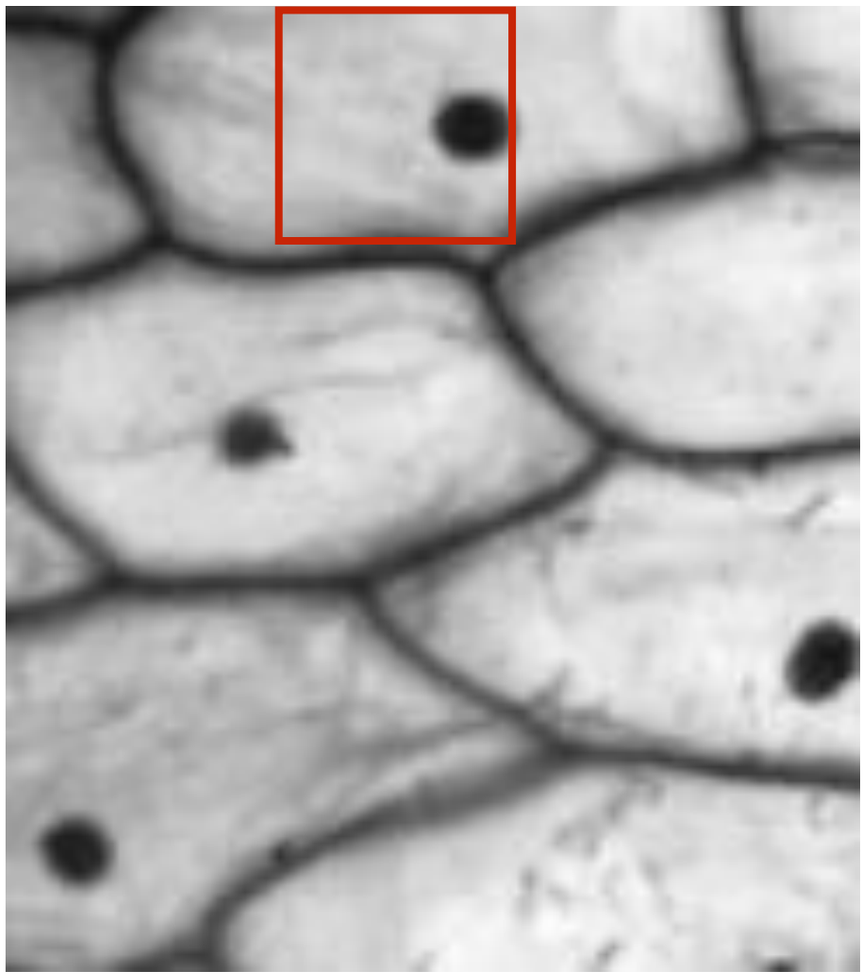
Sliding window classification



$\cdot w < \tau$

Last Lecture

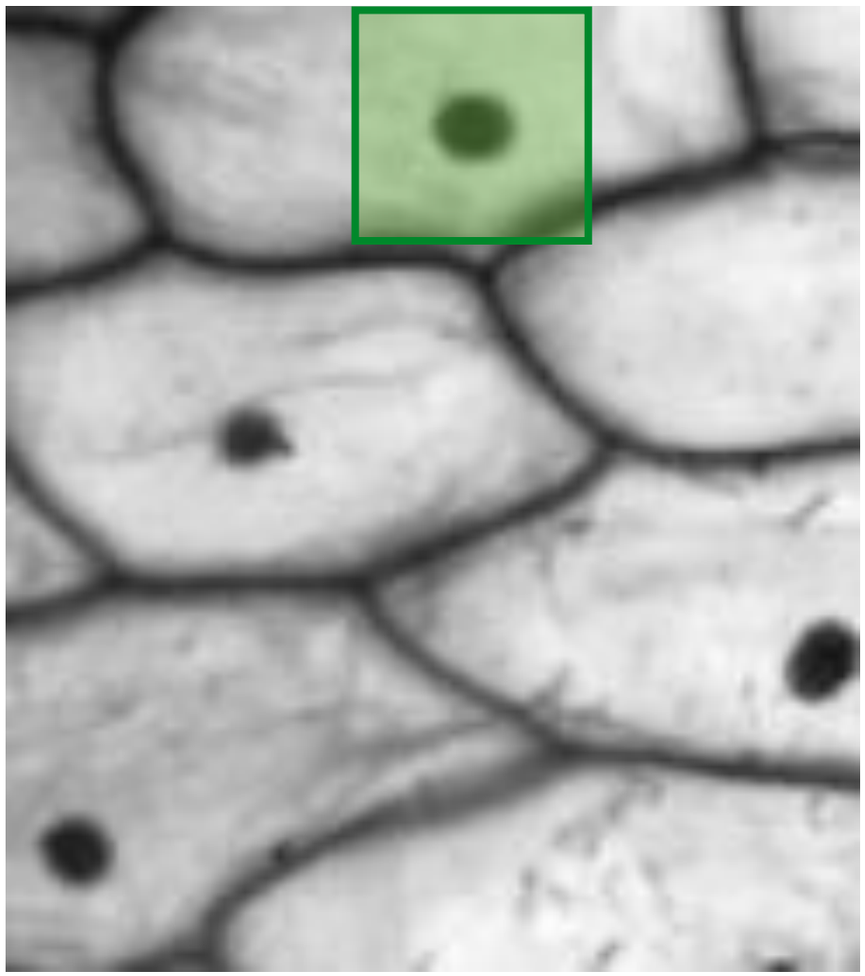
Sliding window classification



$\cdot\, w < \tau$

Last Lecture

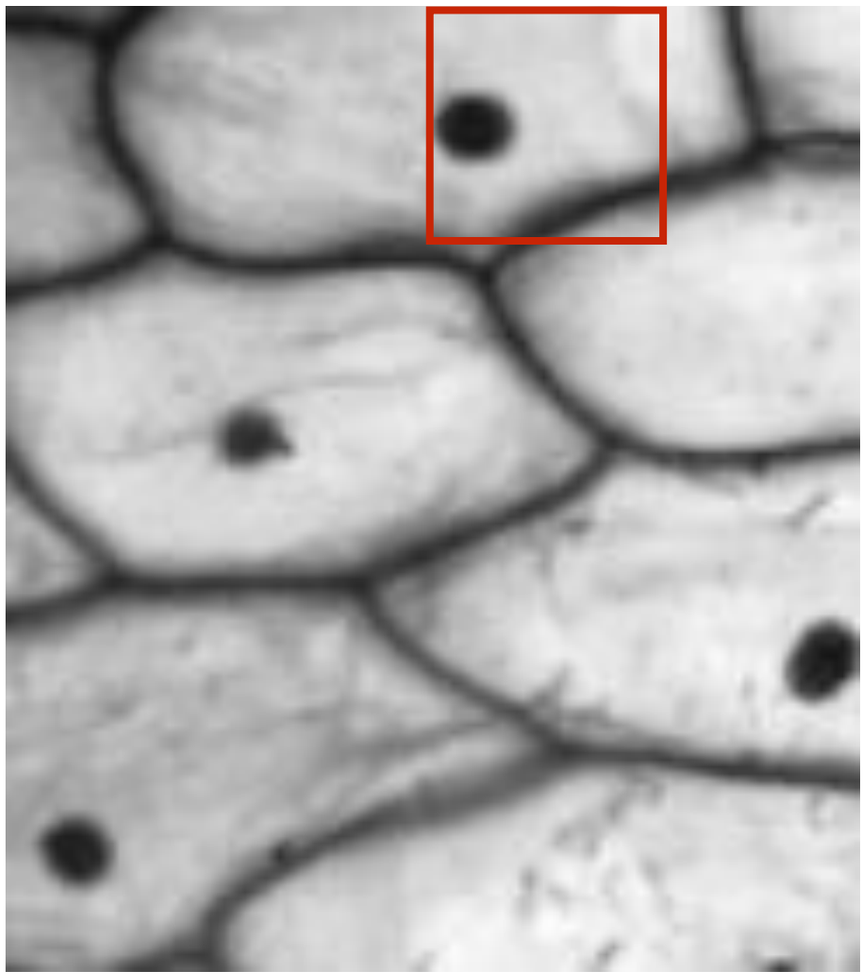
Sliding window classification



$\cdot w > \tau$

Last Lecture

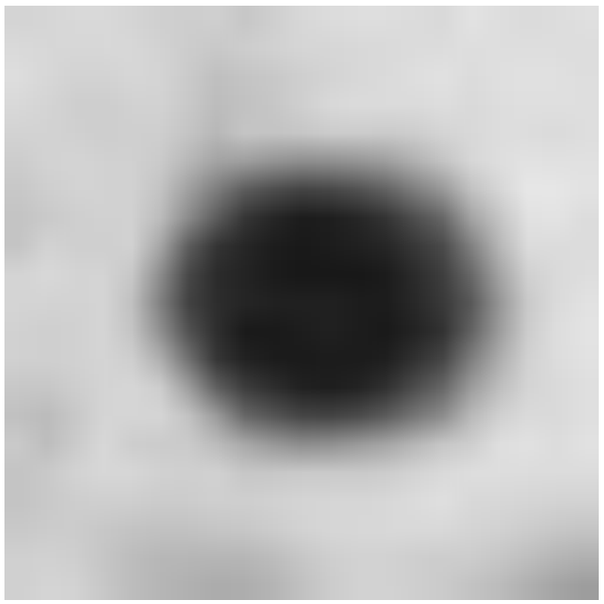
Sliding window classification



$\cdot w < \tau$

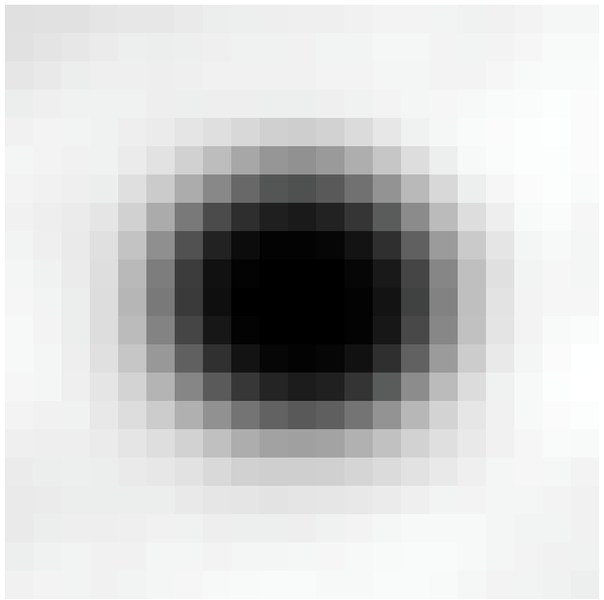
Last Lecture

Linear Filters

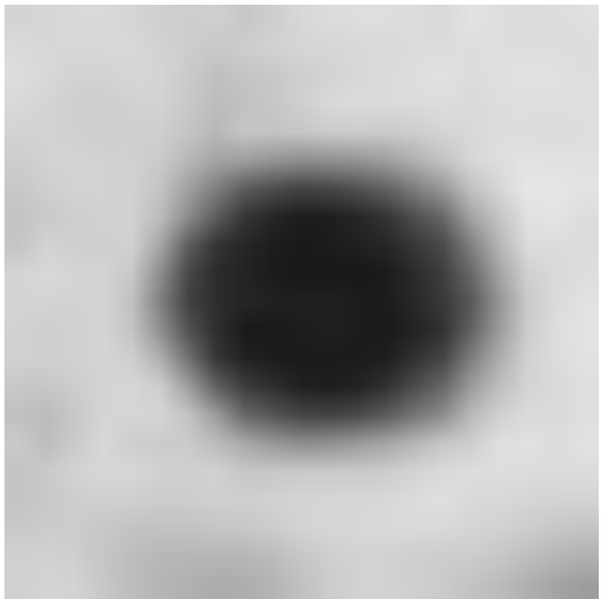
$$w \cdot$$  $$> \tau$$

Last Lecture

Linear Filters



$\cdot$

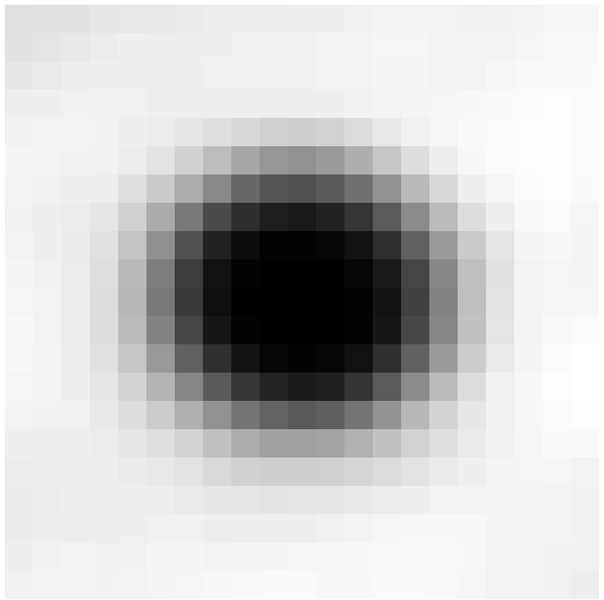


$> \tau$

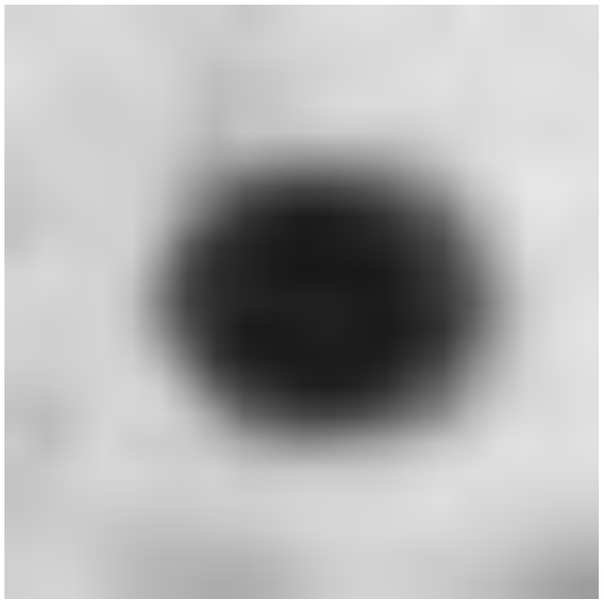
Mean positive / foreground patch

Last Lecture

Linear Filters



$\cdot$

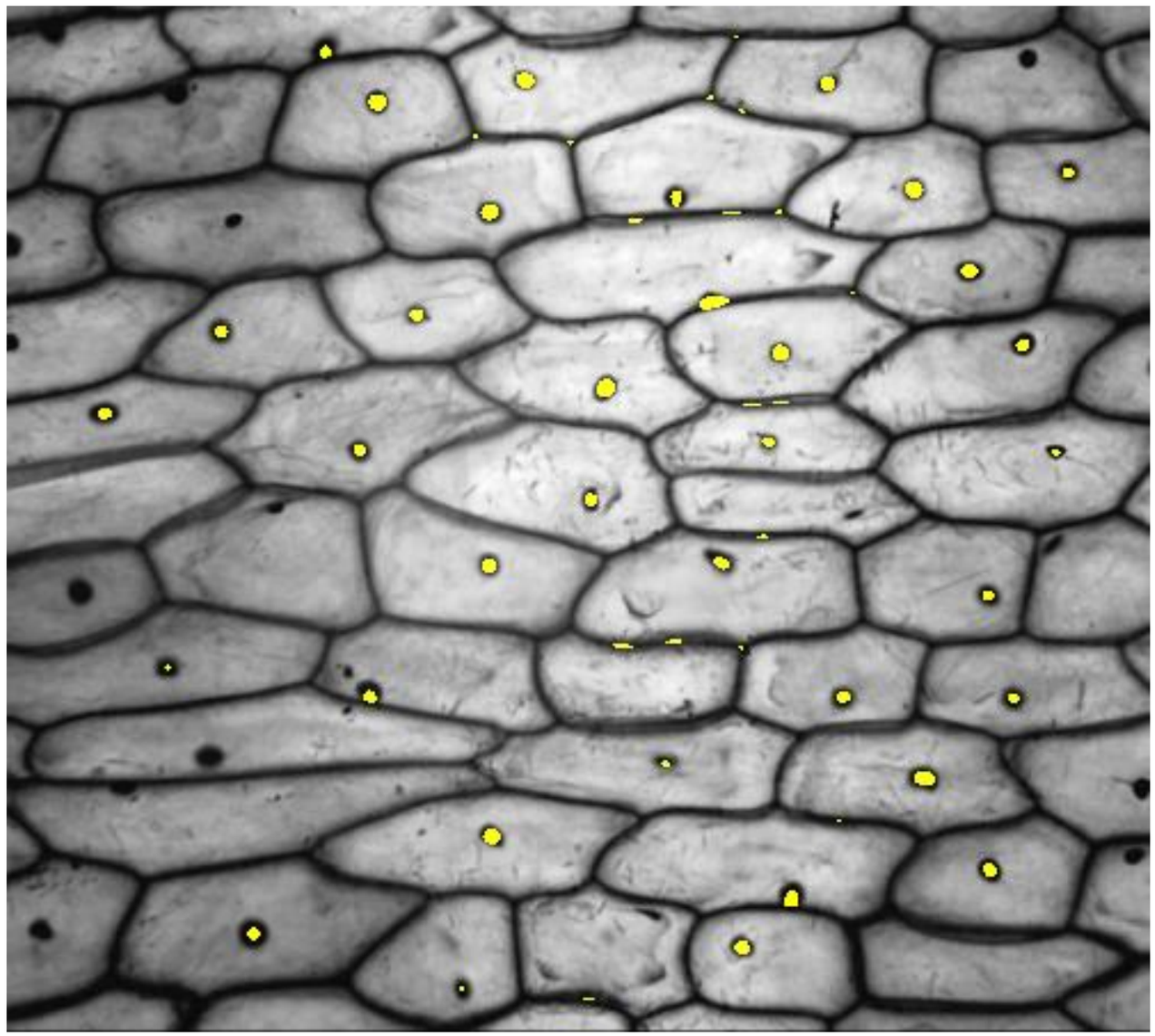


$> \tau$

$$\frac{1}{N_{\text{elements}}} (T - \mu_T \mathbf{1})$$

Last Lecture

Nonlinear filters: Non-maximum suppression



Last Lecture

Nonlinear filters: Non-maximum suppression

1	2	1	2	3	4	2
11	21	22	21	14	6	7
12	20	45	32	21	12	11
11	12	11	16	21	12	21
21	22	23	25	35	22	20
12	11	16	17	16	6	0
0	7	0	21	12	11	0

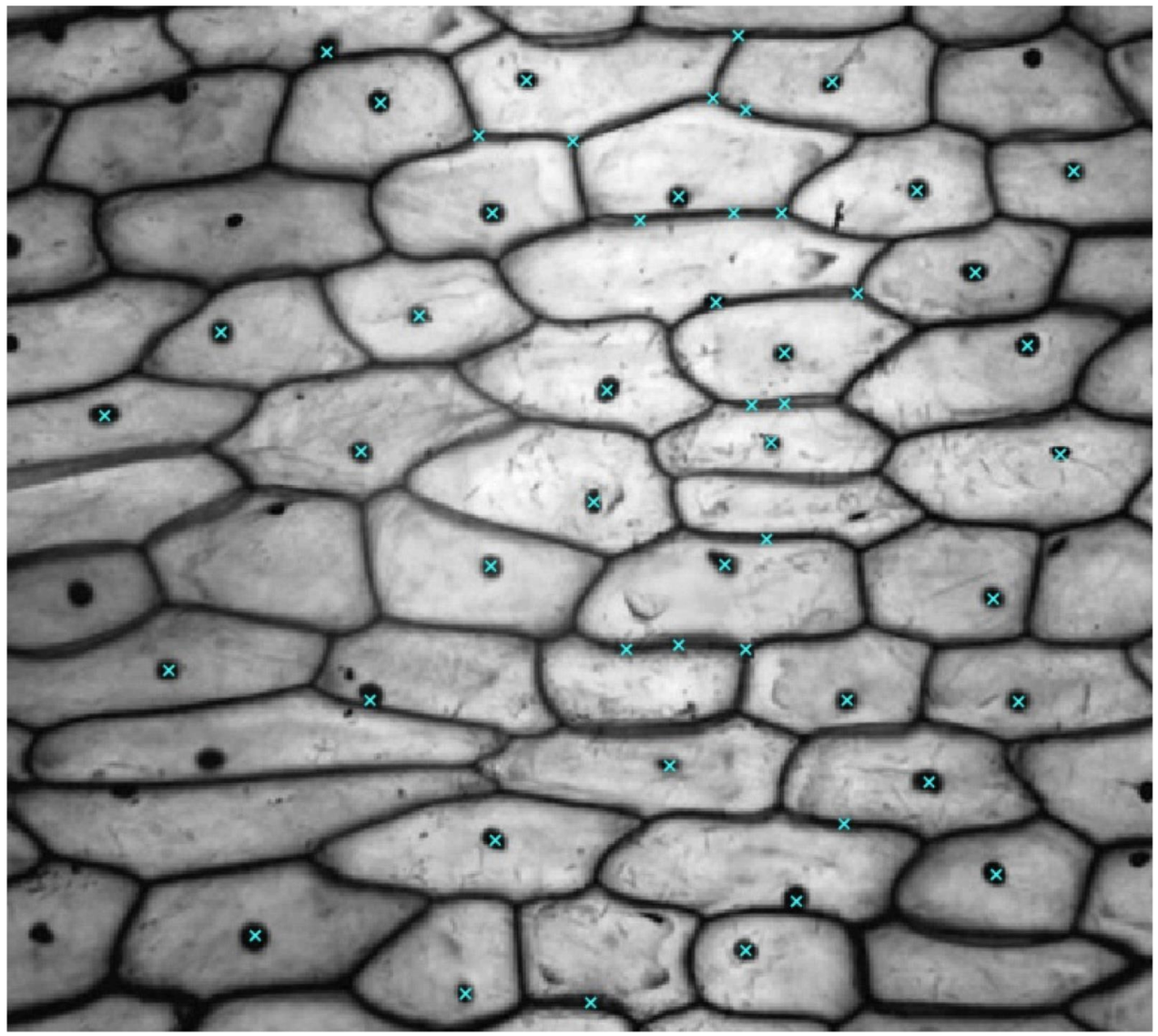
input

	0	0	0	0	0	
	0	1				

result

Last Lecture

Nonlinear filters: Non-maximum suppression



Today

- More Filters
- Similarity Measures
- Multi-Scale Processing

More Filters

Applying a Filter

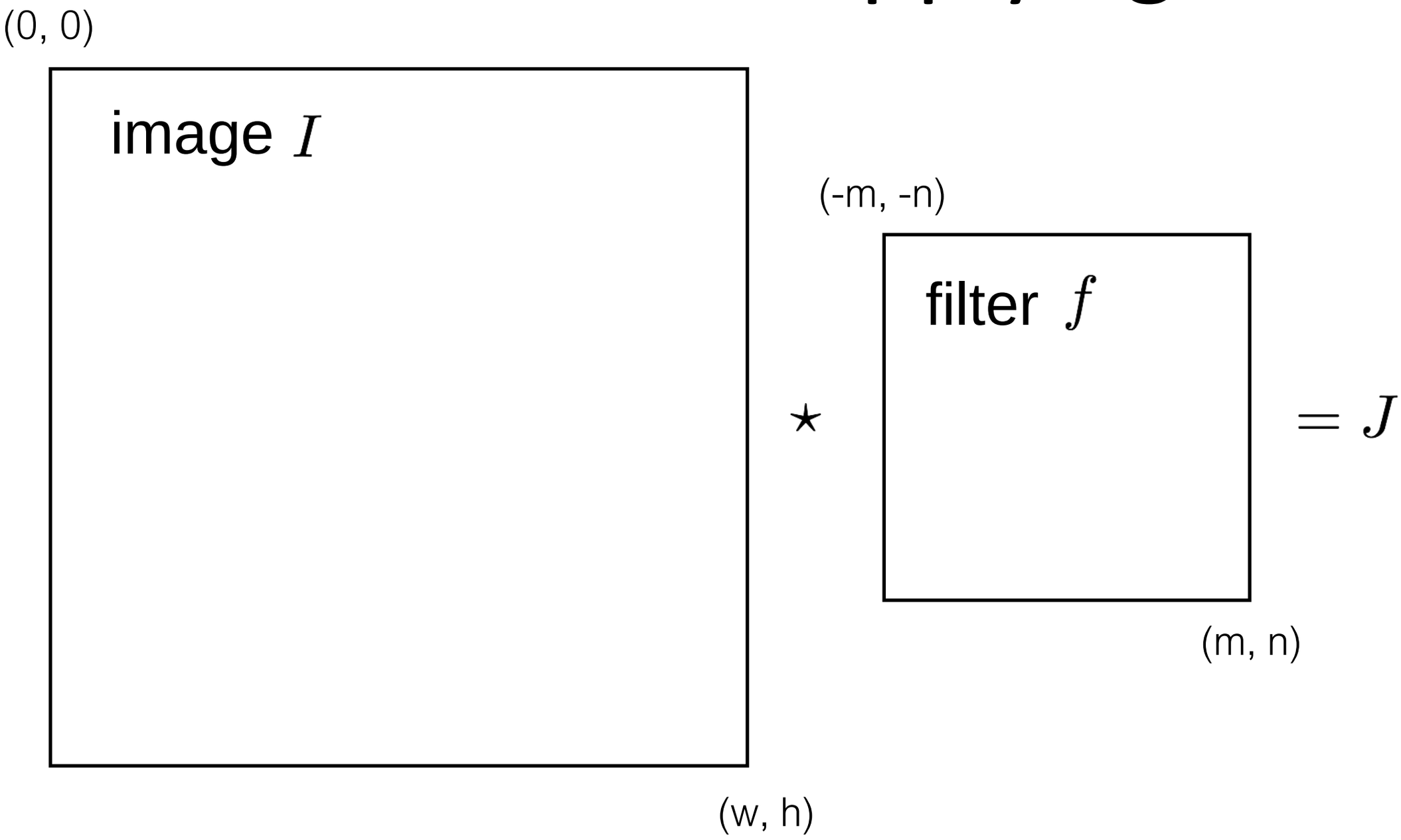
image I

$\star$

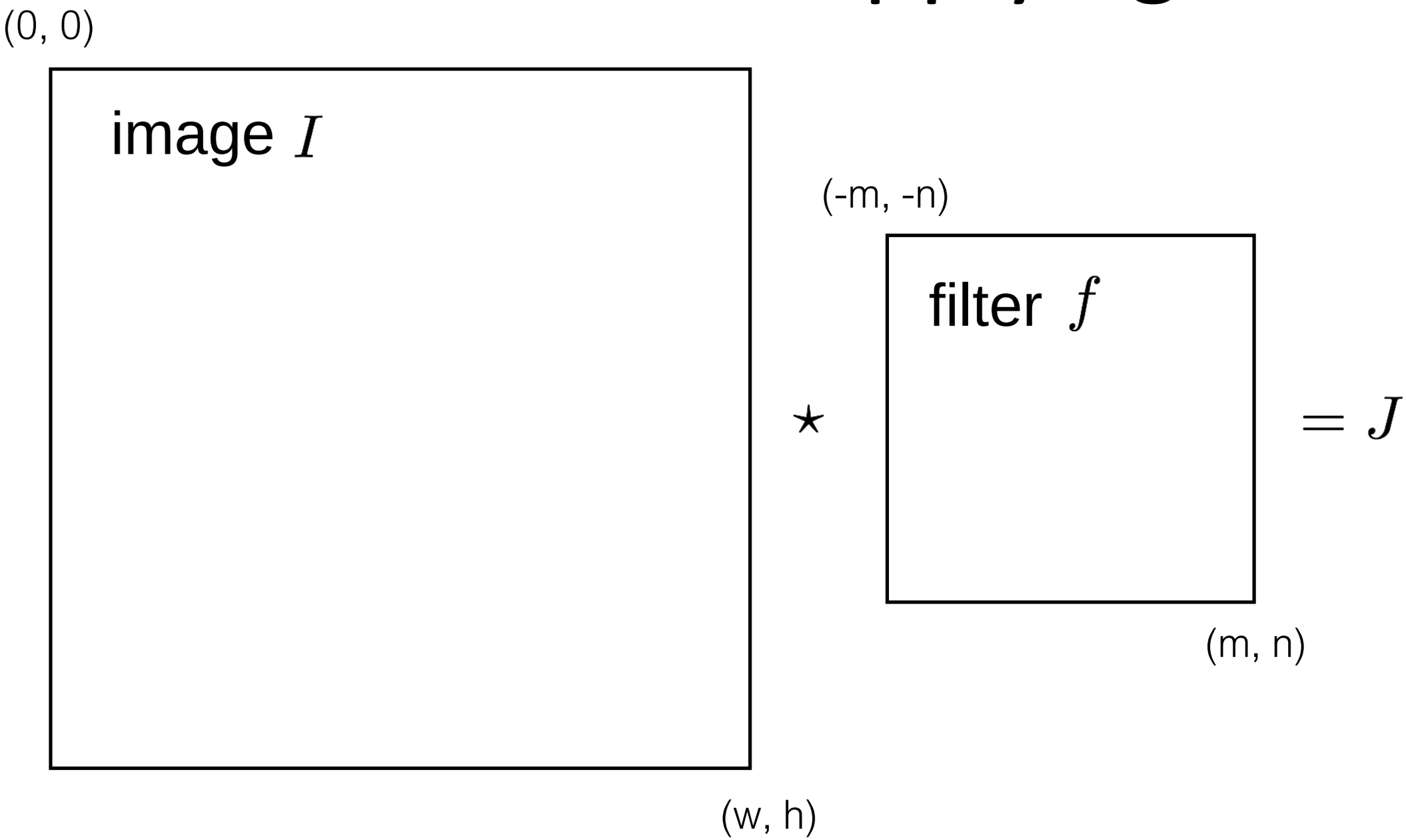
filter f

$= J$

Applying a Filter

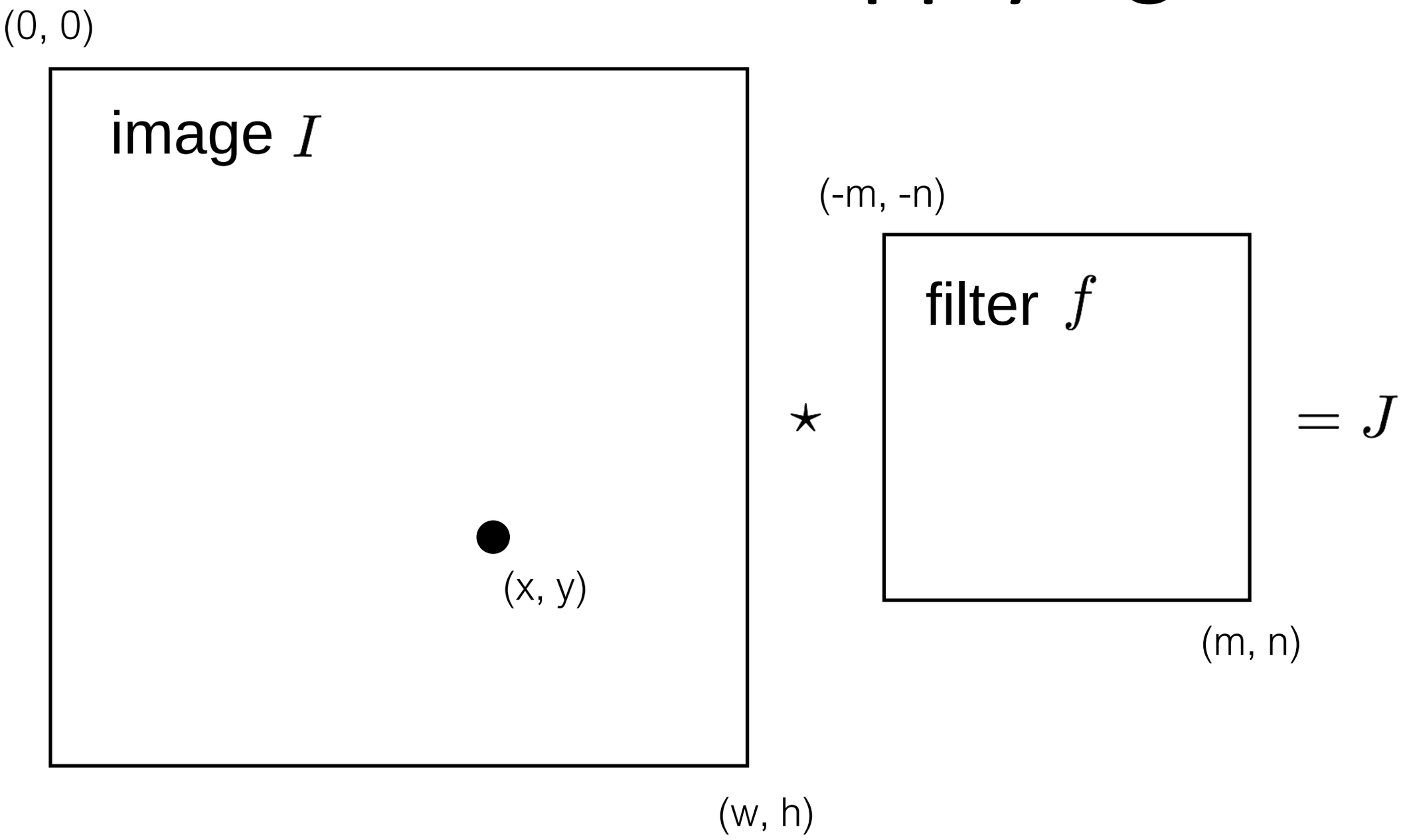


Applying a Filter



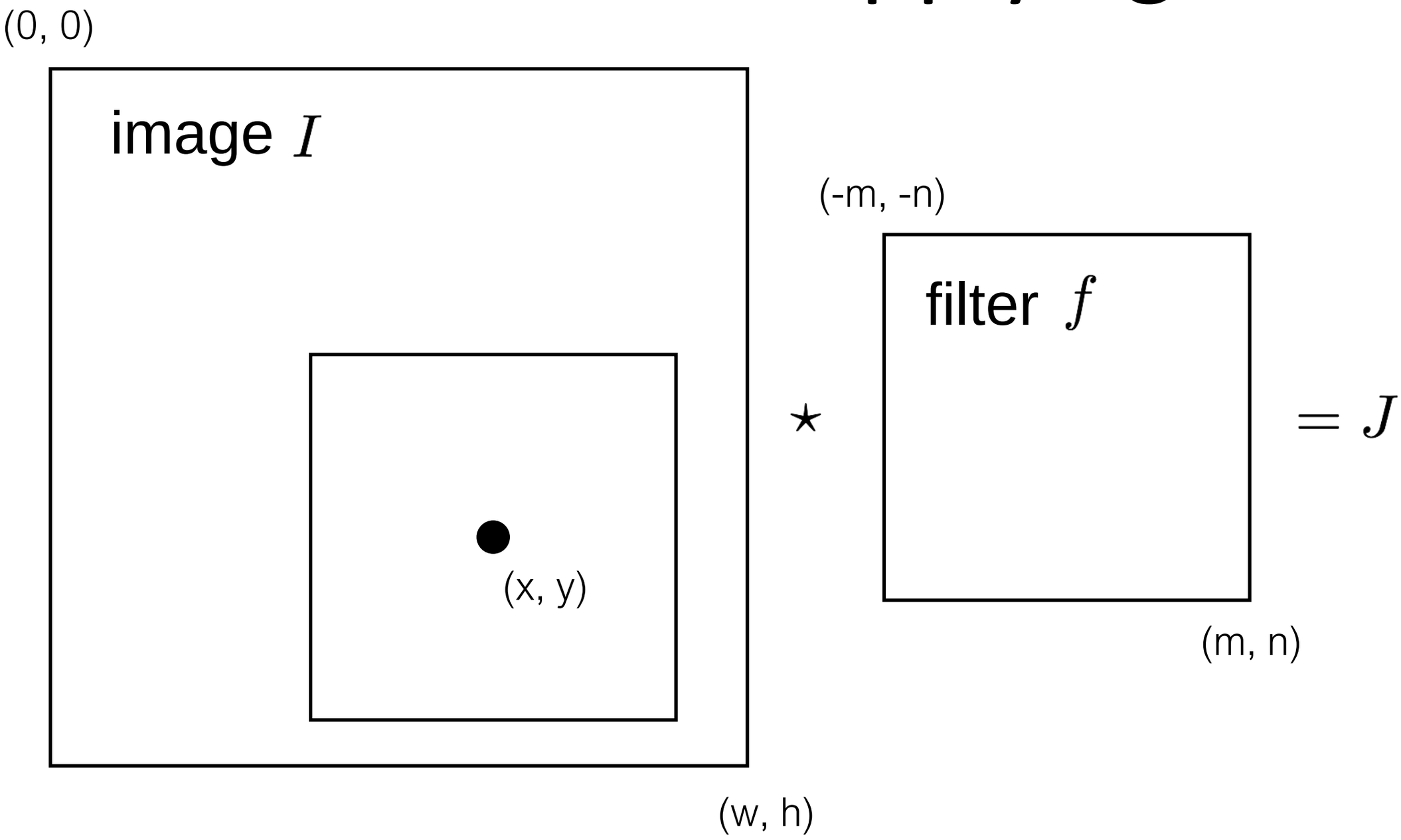
$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x+u, y+v) \cdot f(u,v)$$

Applying a Filter



$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x+u, y+v) \cdot f(u,v)$$

Applying a Filter



$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x+u, y+v) \cdot f(u,v)$$

Applying a Filter: A Warning

We are applying filters via cross-correlation:

$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x+u, y+v) \cdot f(u,v)$$

Often denotes as $J = I \otimes f$

Applying a Filter: A Warning

We are applying filters via cross-correlation:

$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x+u, y+v) \cdot f(u,v)$$

Often denotes as $J = I \otimes f$

Similar to mathematical definition of convolution:

$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x-u, y-v) \cdot f(u,v)$$

Applying a Filter: A Warning

We are applying filters via cross-correlation:

$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x+u, y+v) \cdot f(u,v)$$

Often denotes as $J = I \otimes f$

Similar to mathematical definition of convolution:

$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x-u, y-v) \cdot f(u,v)$$

Convolution typically denoted as $J = I \star f$

Applying a Filter: A Warning

We are applying filters via cross-correlation:

$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x+u, y+v) \cdot f(u,v)$$

Often denotes as $J = I \otimes f$

Similar to mathematical definition of convolution:

$$J(x,y) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} I(x-u, y-v) \cdot f(u,v)$$

Convolution typically denoted as $J = I \star f$

But, we will use $J = I \star f$ to denote correlation here!

Average Filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Average Filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

image source: http://graphics.stanford.edu/data/3Dscanrep/

Average Filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

image source: http://graphics.stanford.edu/data/3Dscanrep/

Average Filter

1		
1		
1		

image source: http://graphics.stanford.edu/data/3Dscanrep/

Average Filter

1		
1		
1		

image source: http://graphics.stanford.edu/data/3Dscanrep/

Gaussian Filter

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

Gaussian Filter

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

standard deviation

Gaussian Filter

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

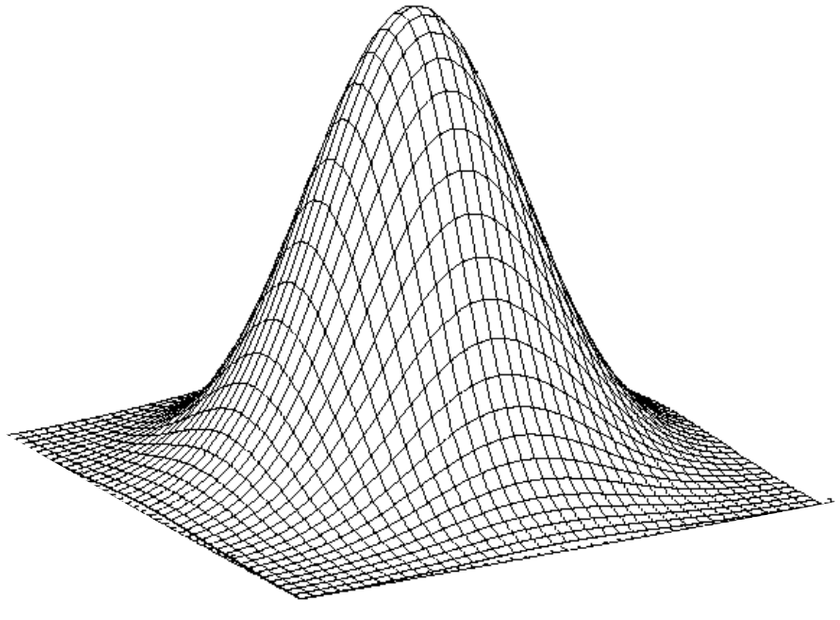
standard deviation

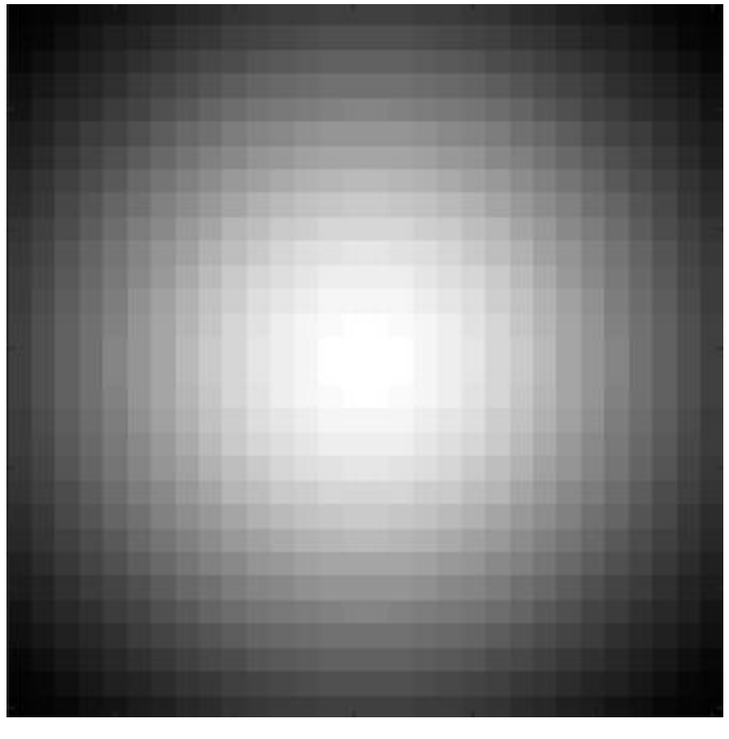
normalization factor

Gaussian Filter

$$G_\sigma(x,y) = \underbrace{\frac{1}{2\pi\sigma^2}}_{\text{normalization factor}} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

standard deviation

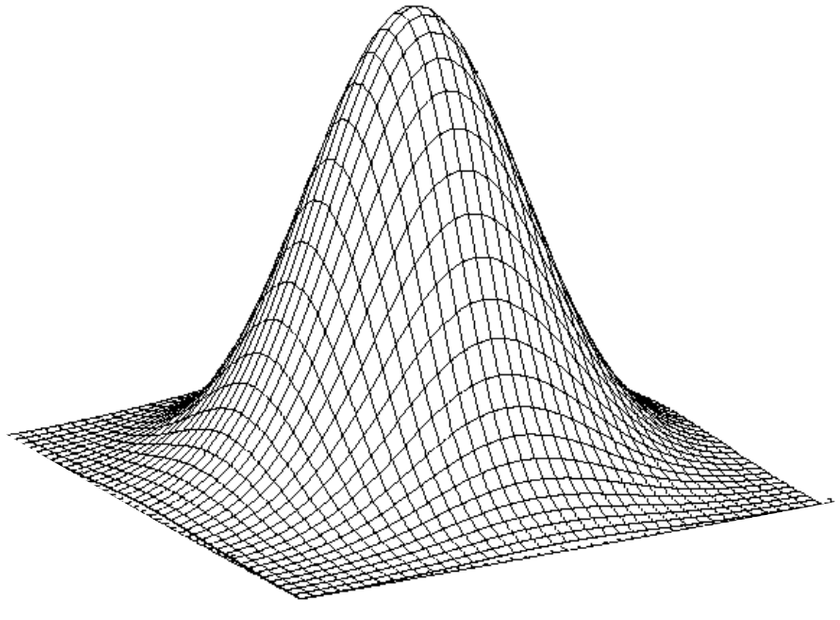


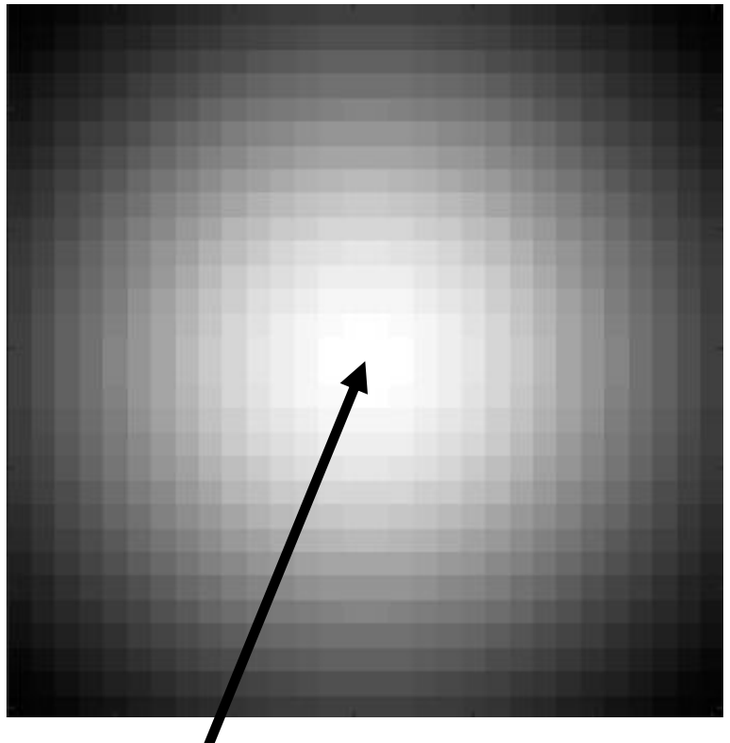


Gaussian Filter

$$G_\sigma(x,y) = \underbrace{\frac{1}{2\pi\sigma^2}}_{\text{normalization factor}} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

standard deviation





(0, 0)

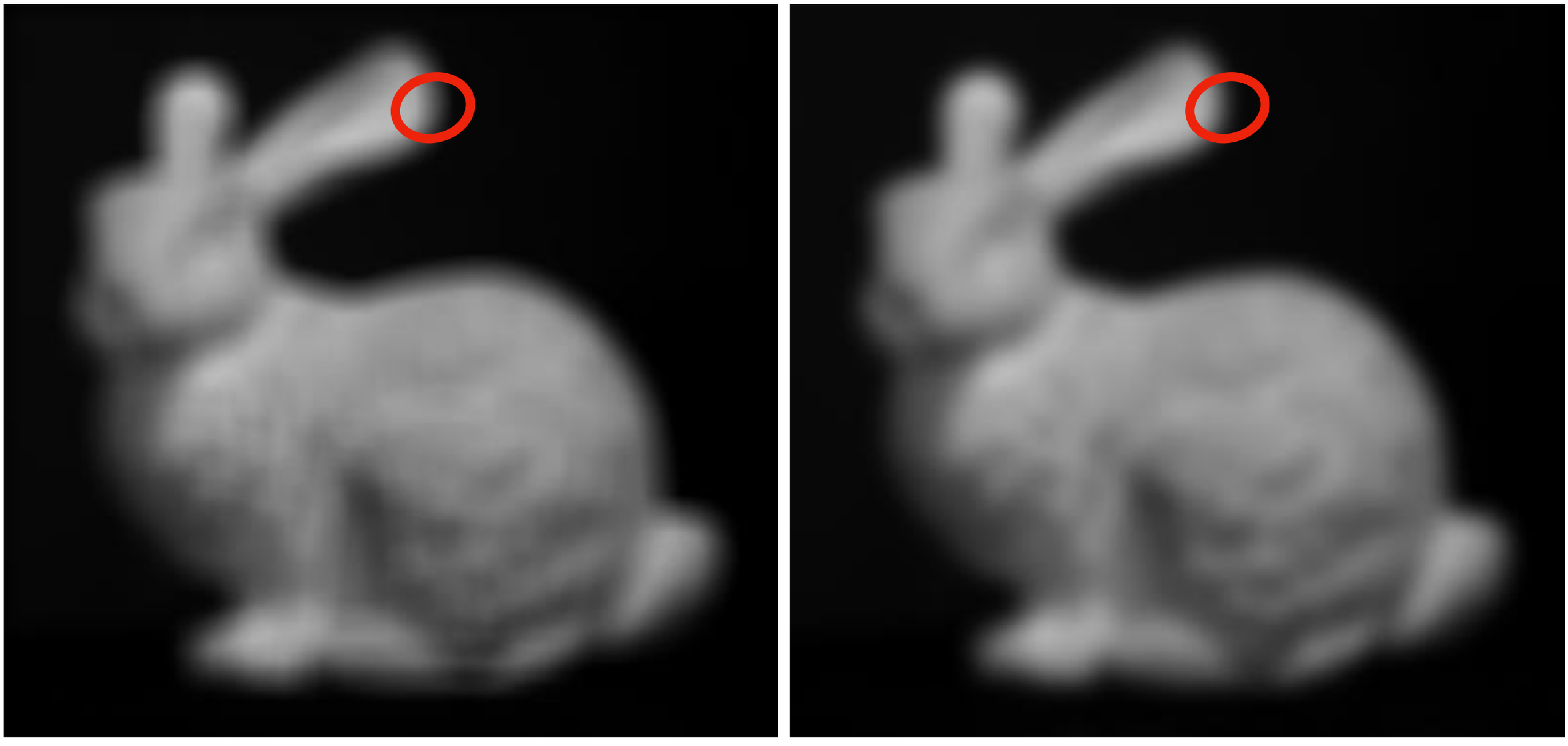
Gaussian Filter

Average filter

Gaussian filter

image source: http://graphics.stanford.edu/data/3Dscanrep/

Gaussian Filter

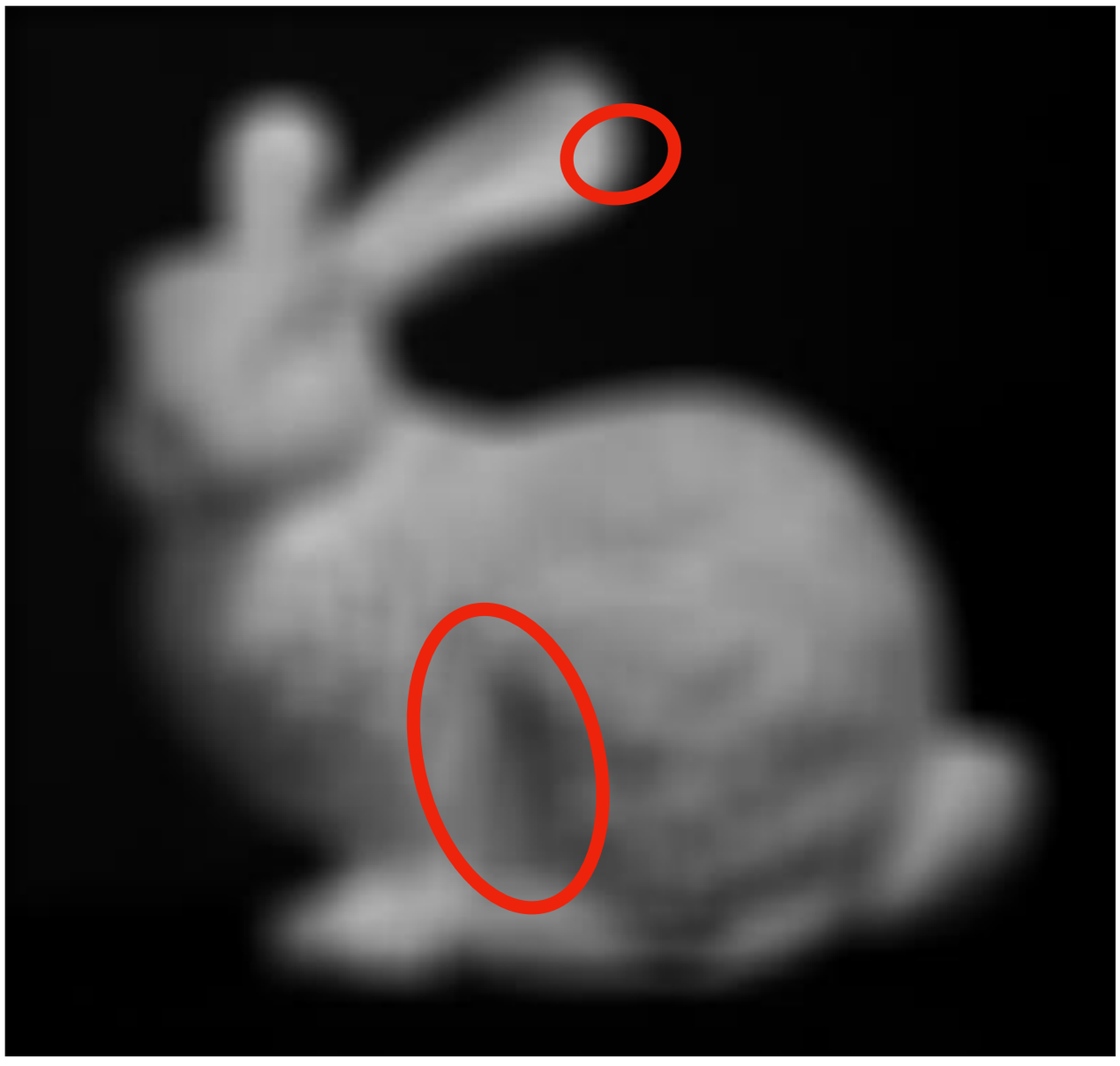


Average filter

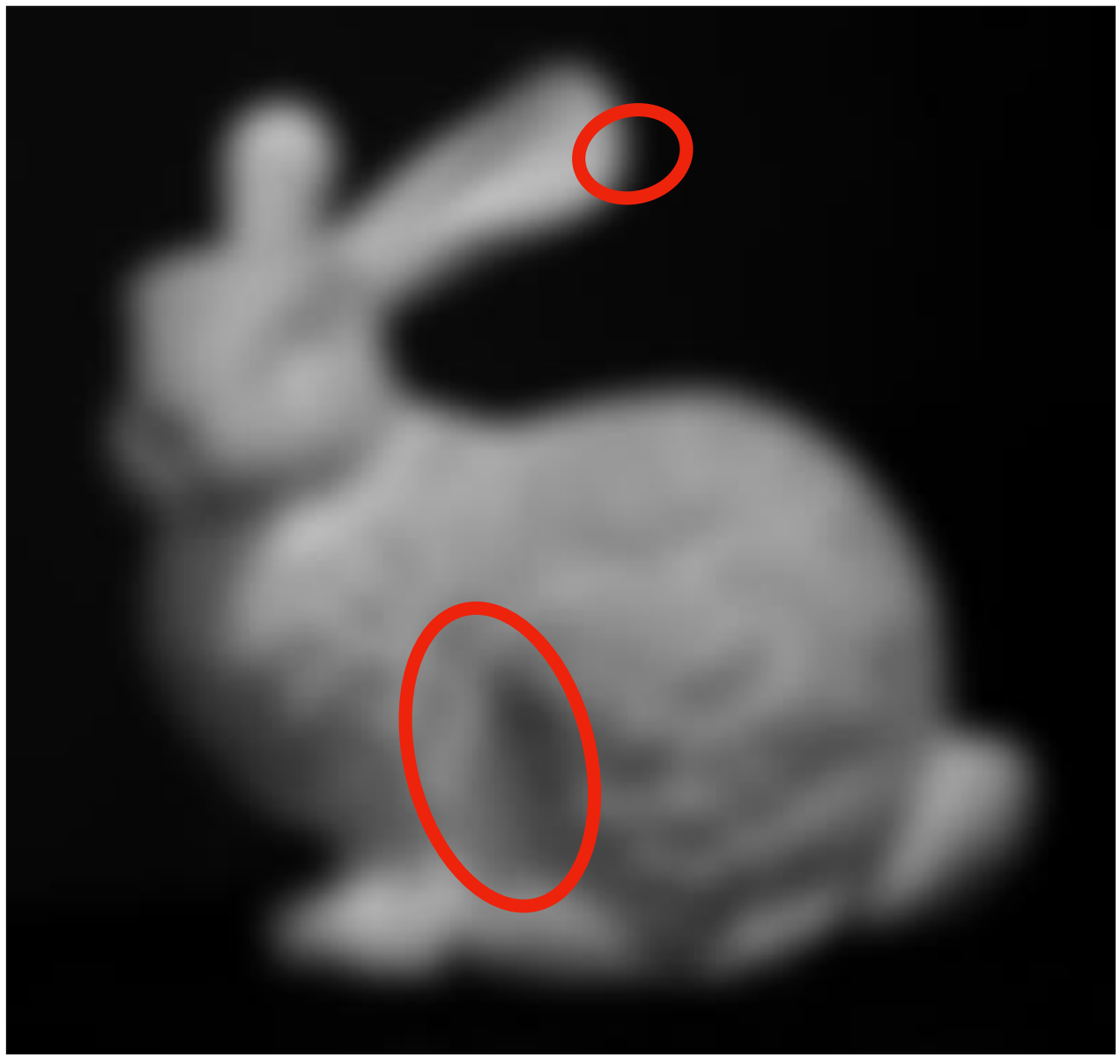
Gaussian filter

image source: http://graphics.stanford.edu/data/3Dscanrep/

Gaussian Filter



Average filter

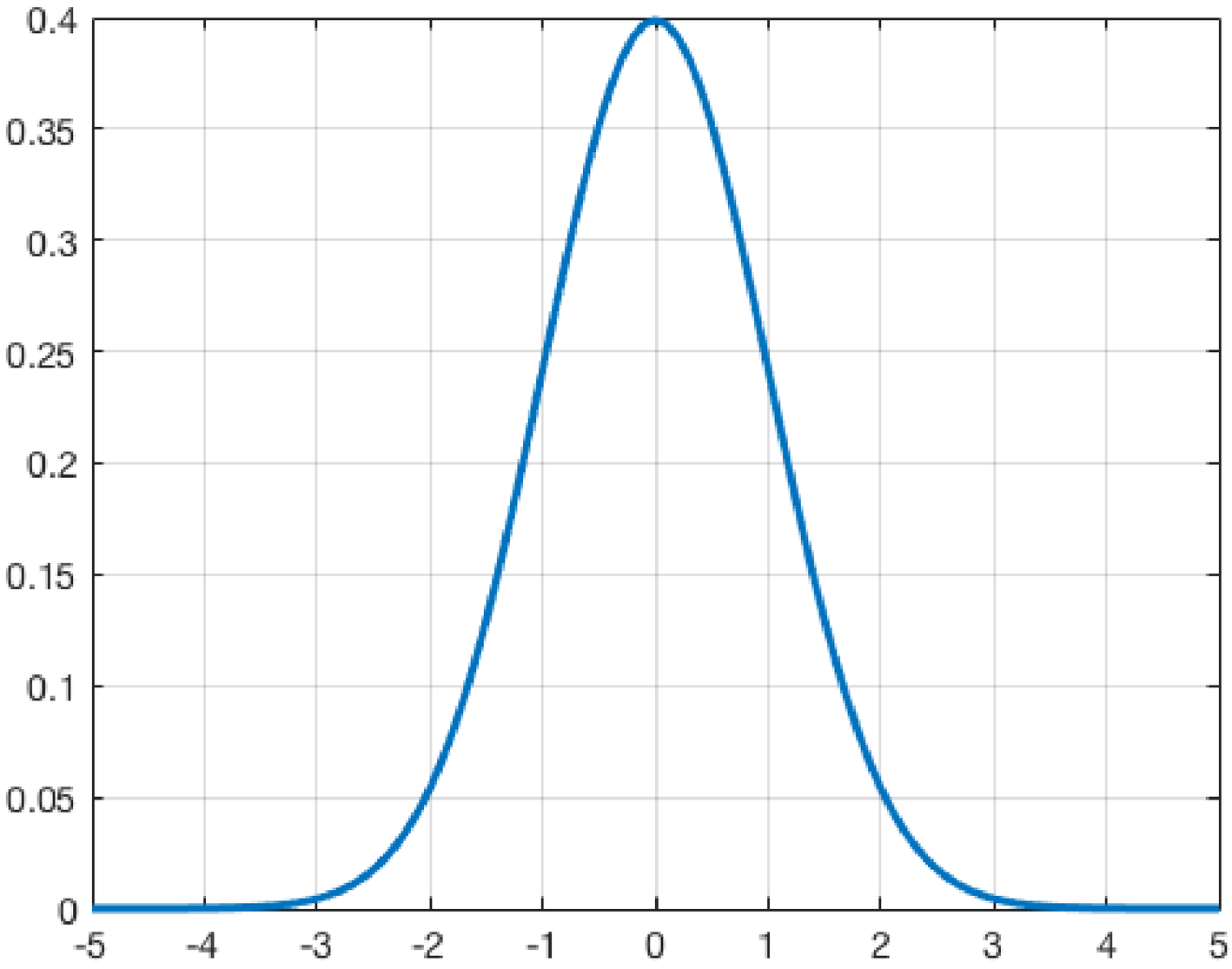


Gaussian filter

image source: http://graphics.stanford.edu/data/3Dscanrep/

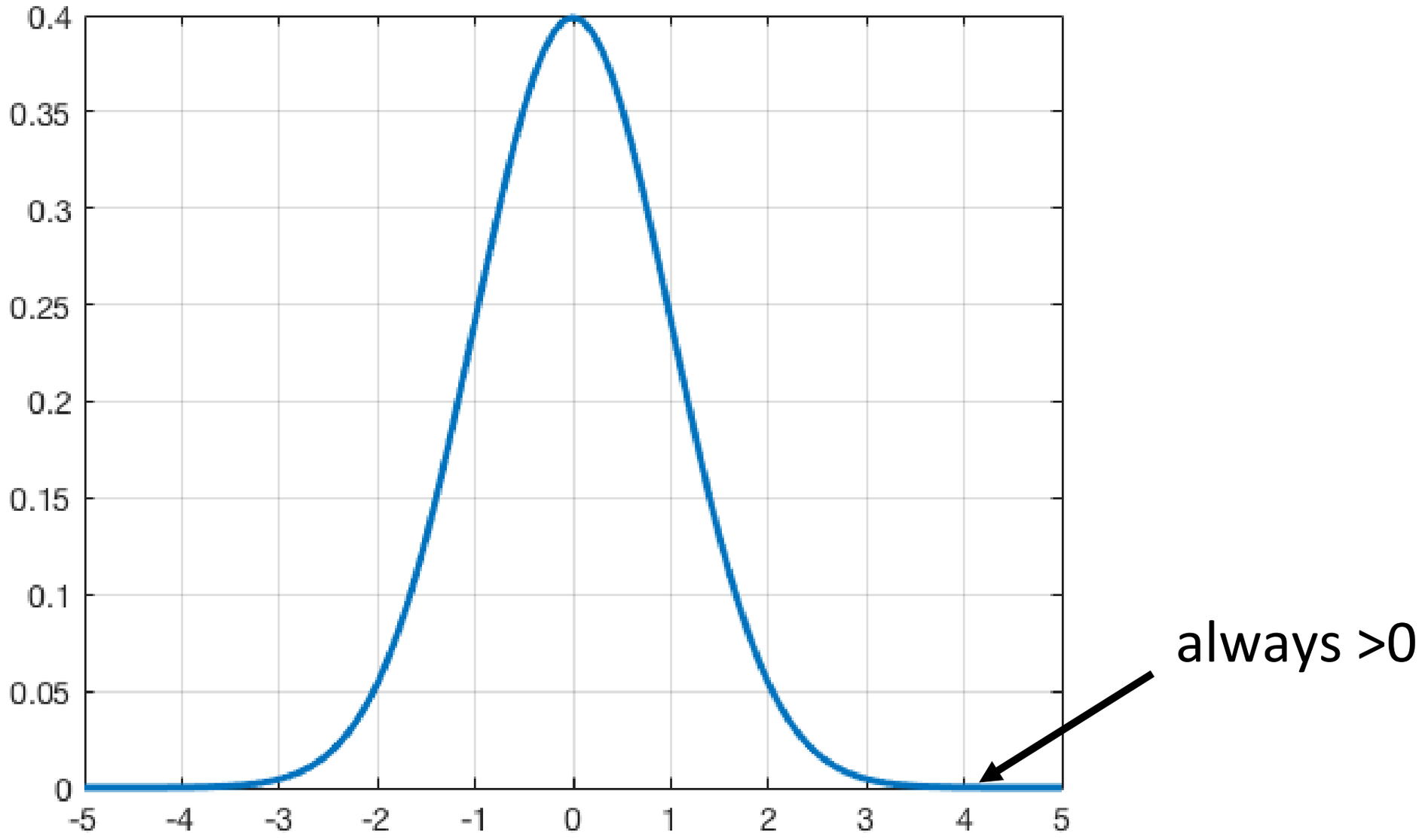
Gaussian Filter

Gaussian function has infinite support



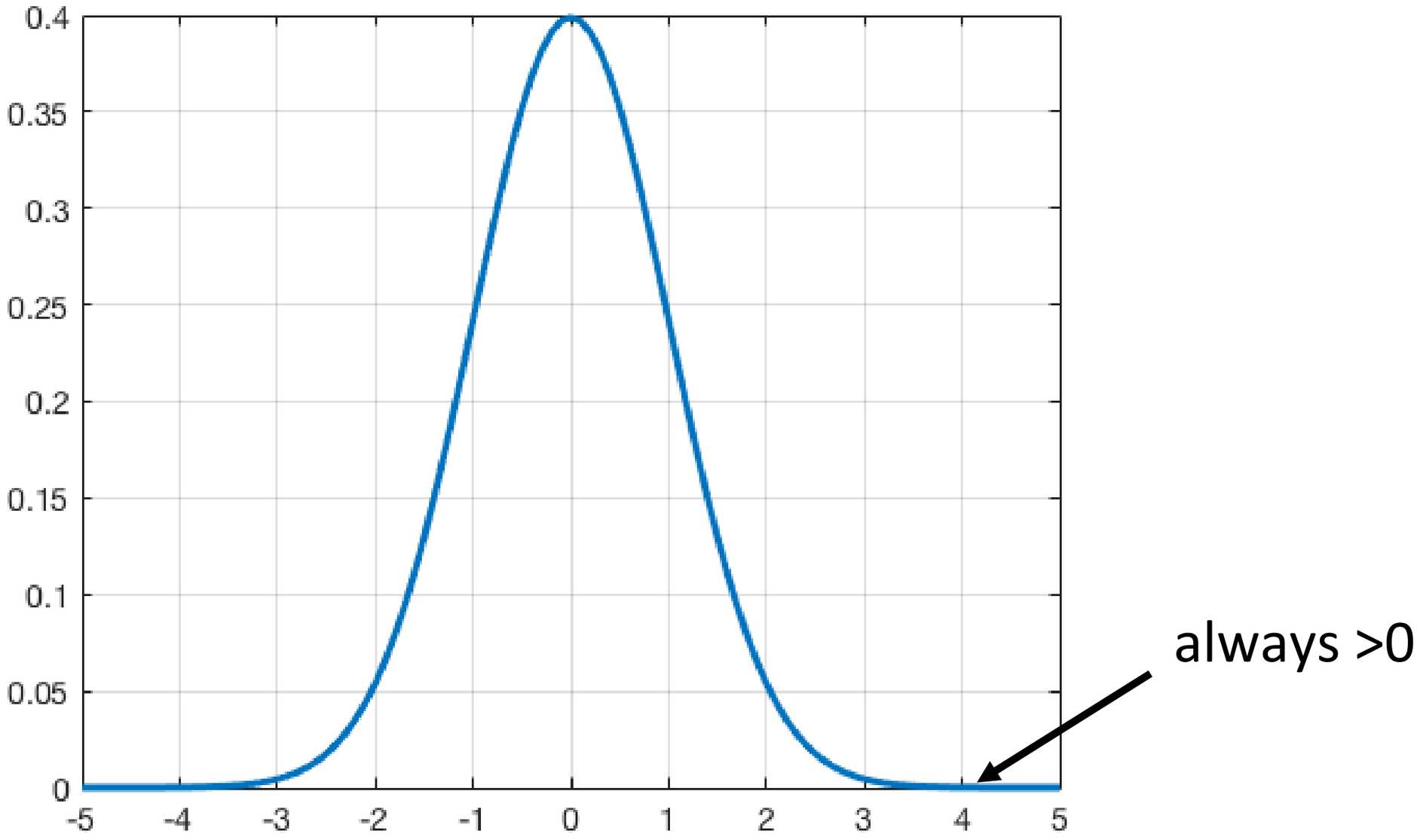
Gaussian Filter

Gaussian function has infinite support



Gaussian Filter

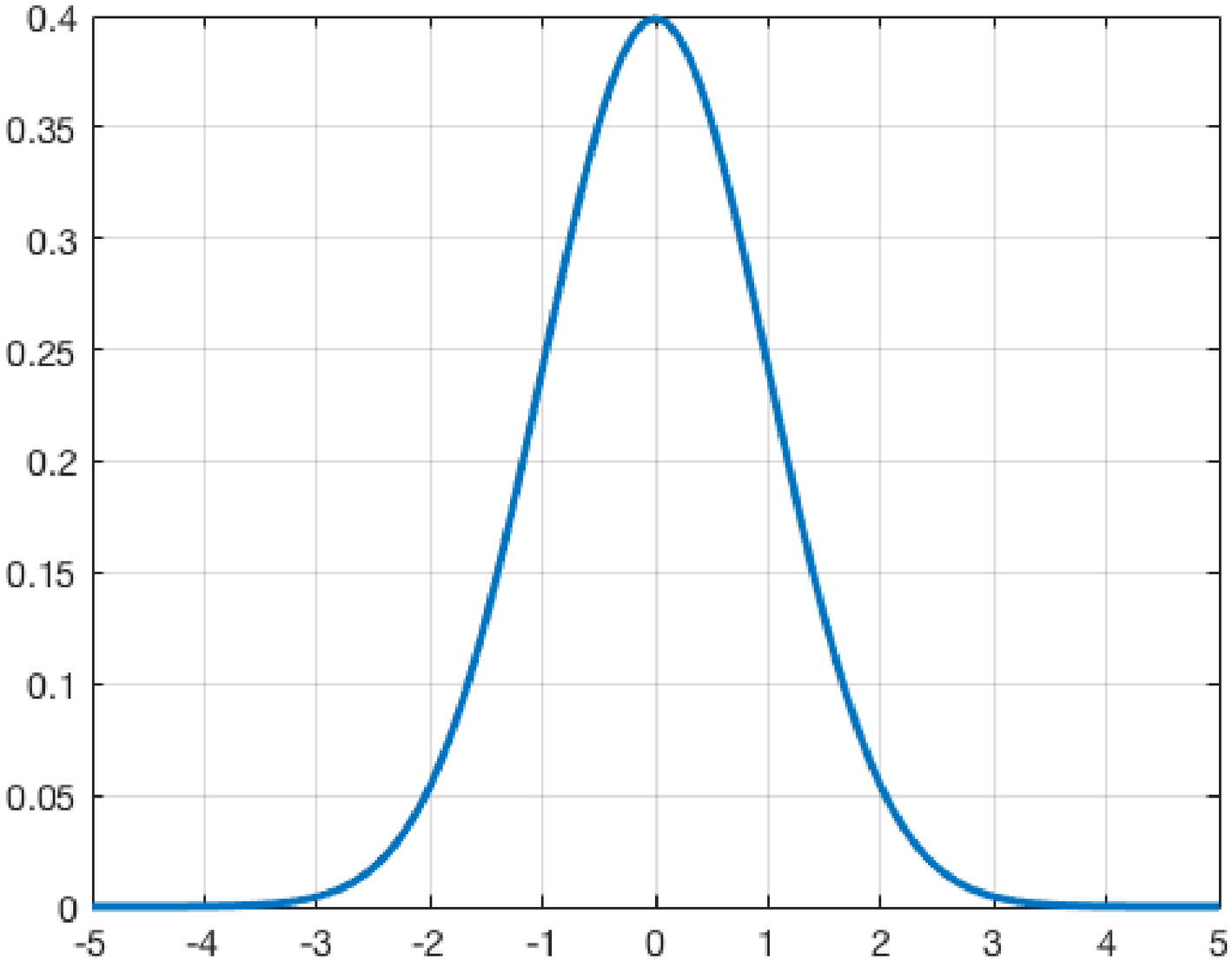
Gaussian function has infinite support



How to obtain a finite filter?

Gaussian Filter

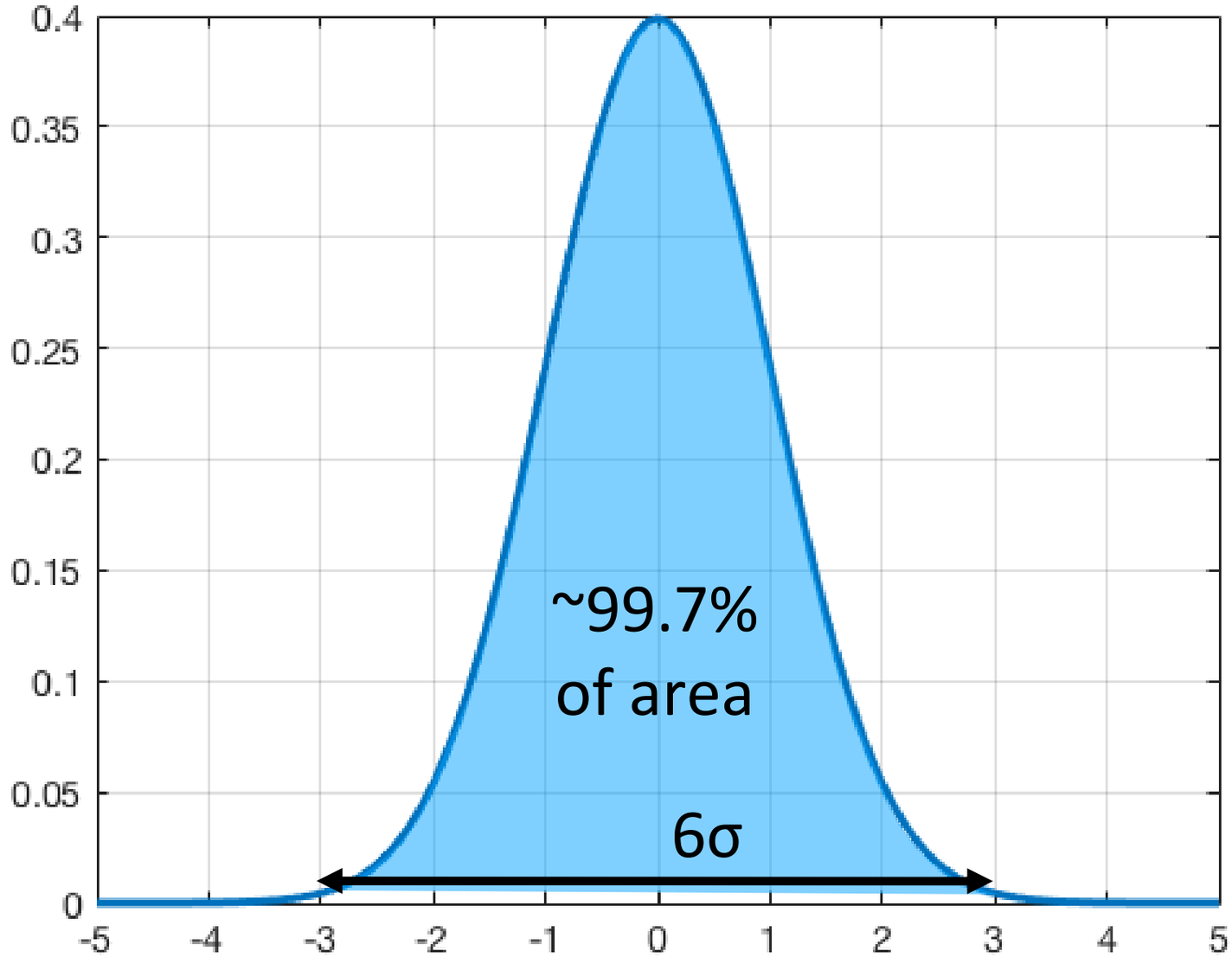
Gaussian function has infinite support



How to obtain a finite filter?

Gaussian Filter

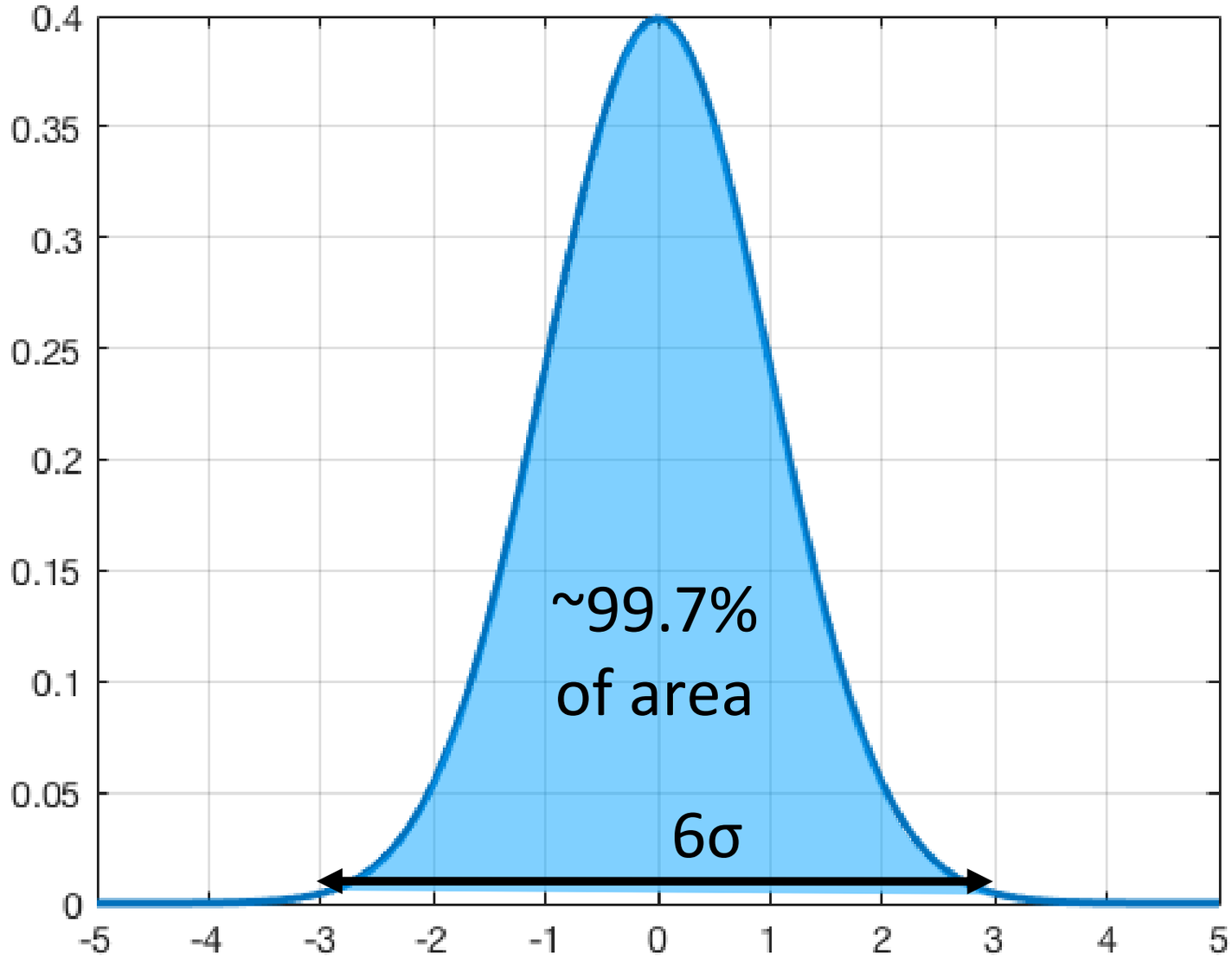
Gaussian function has infinite support



How to obtain a finite filter?

Gaussian Filter

Gaussian function has infinite support



Use filter size ~6σ

Gaussian Filter

Discrete Example for σ=1

$$\frac{1}{256}$$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

Scaled such that smallest value of $G_\sigma(x,y)$ corresponds to 1

Gaussian Filter

Gaussian filters are **separable**

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

2D Gaussian

=

1
4
6
4
1

1D Gaussian
y-direction

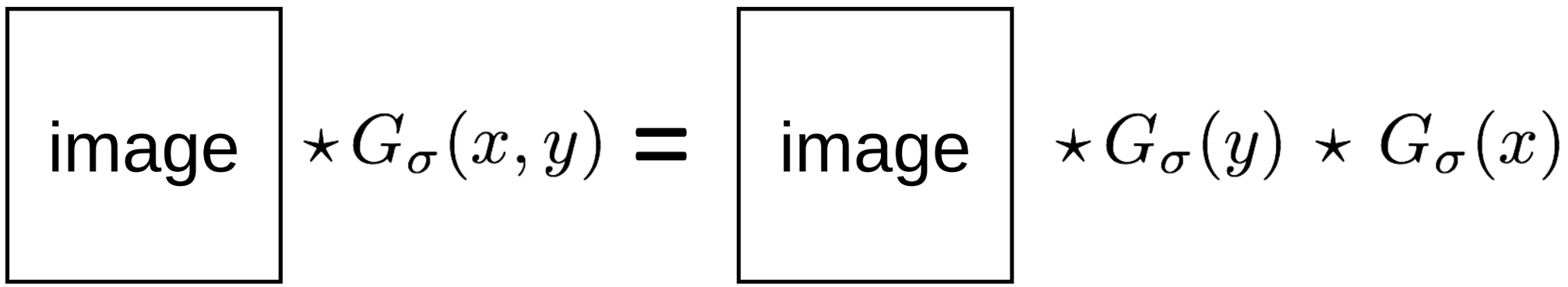
*

1	4	6	4	1

1D Gaussian
x-direction

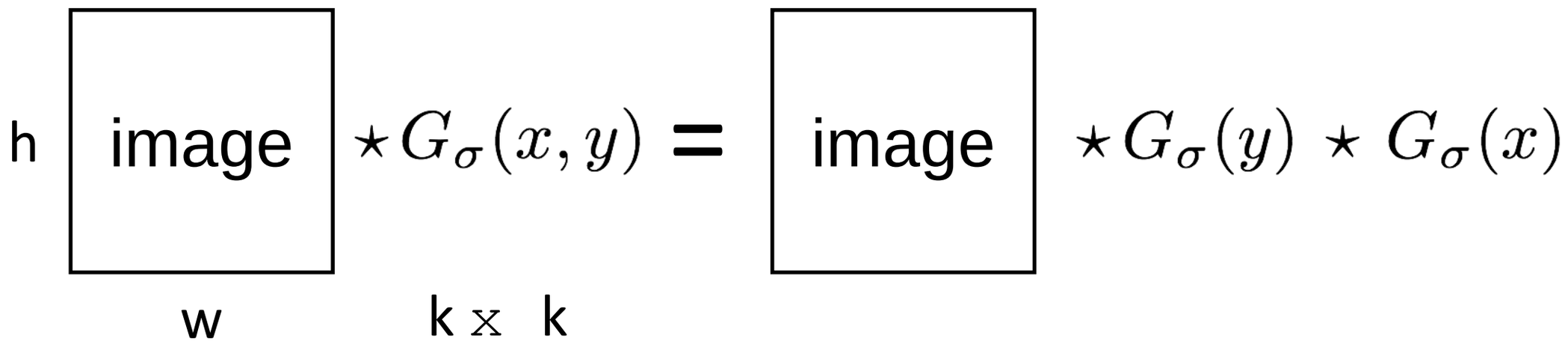
Gaussian Filter

Gaussian filters are **separable**



Gaussian Filter

Gaussian filters are **separable**



Gaussian Filter

Gaussian filters are **separable**

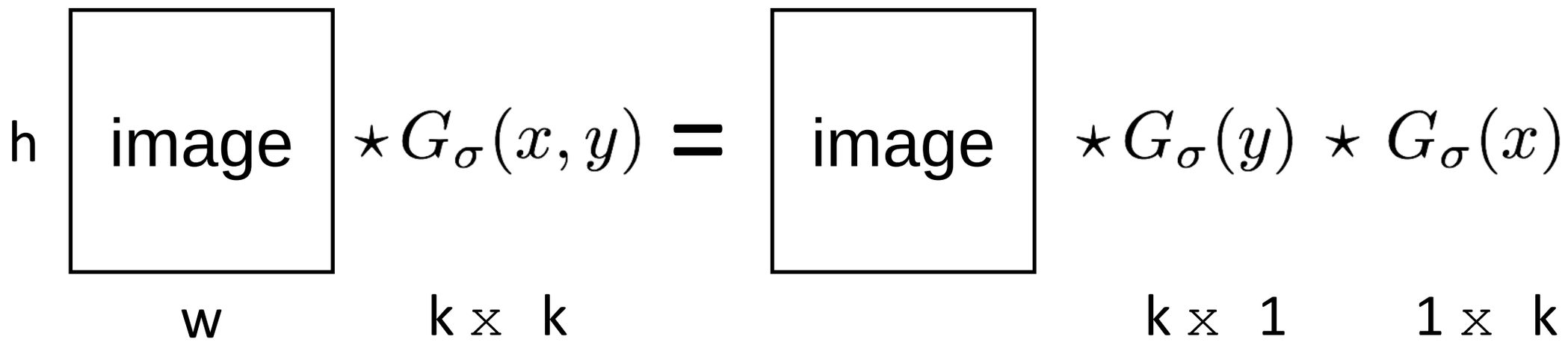
h image $\star G_\sigma(x, y) =$ image $\star G_\sigma(y) \star G_\sigma(x)$

w k x k

w * h * k * k operations

Gaussian Filter

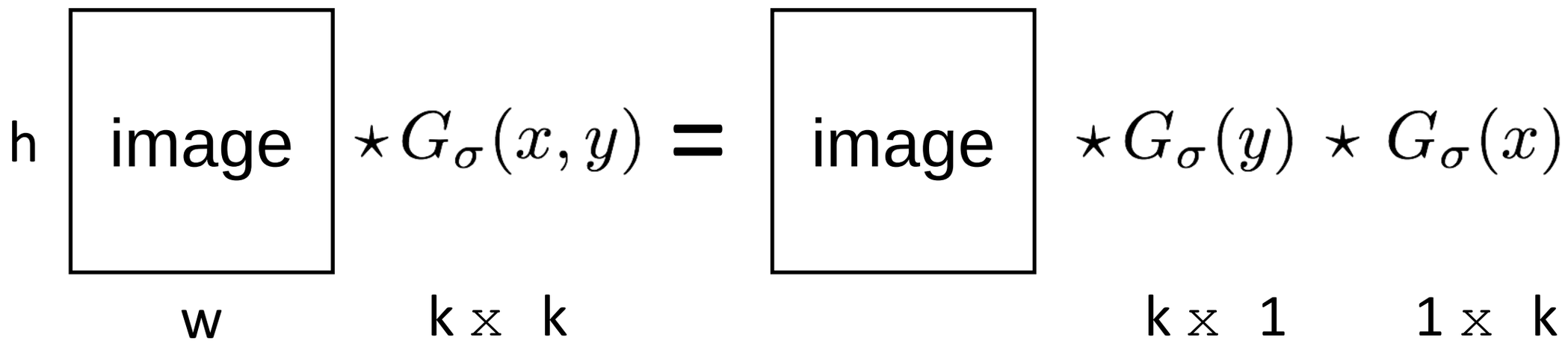
Gaussian filters are **separable**



w * h * k * k operations

Gaussian Filter

Gaussian filters are **separable**

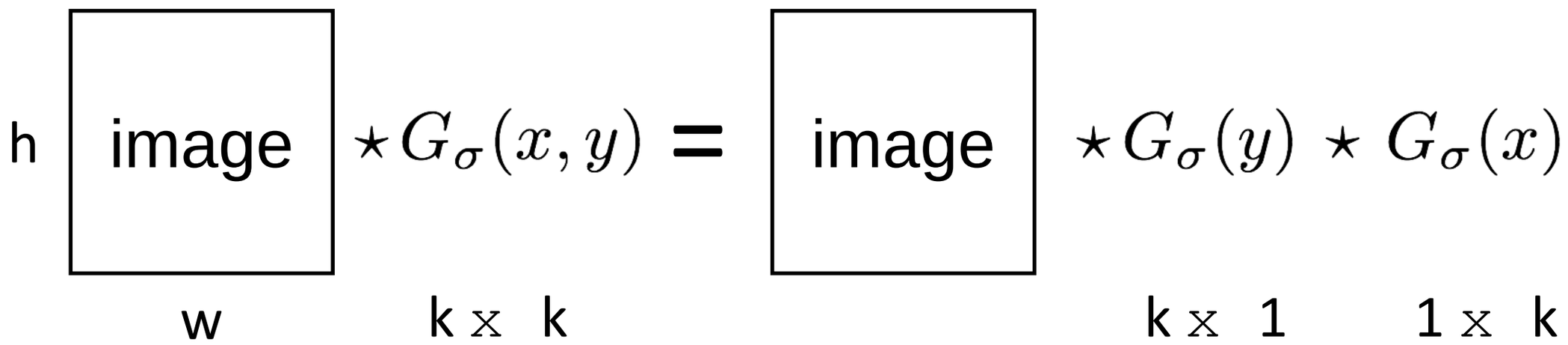


w * h * k * k operations

2 * w * h * k operations

Gaussian Filter

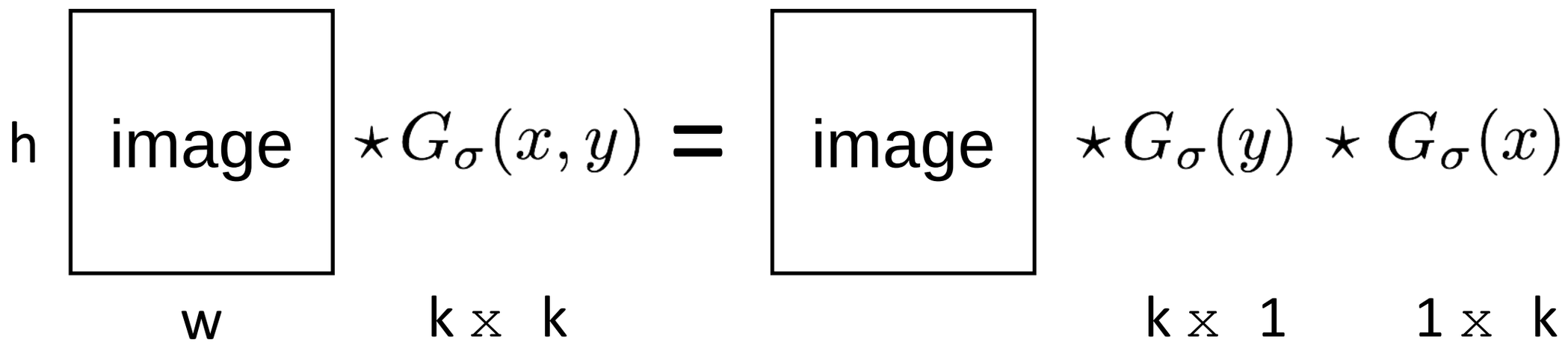
Gaussian filters are **separable**



w * h * k * k operations > 2 * w * h * k operations

Gaussian Filter

Gaussian filters are **separable**



w * h * k * k operations > 2 * w * h * k operations

Separable filter can be applied efficiently

Gaussian Filter

Iterative application

$$I \star G_{\sigma_1}(x,y) \star G_{\sigma_2}(x,y) = I \star G_{\sqrt{\sigma_1^2+\sigma_2^2}}(x,y)$$

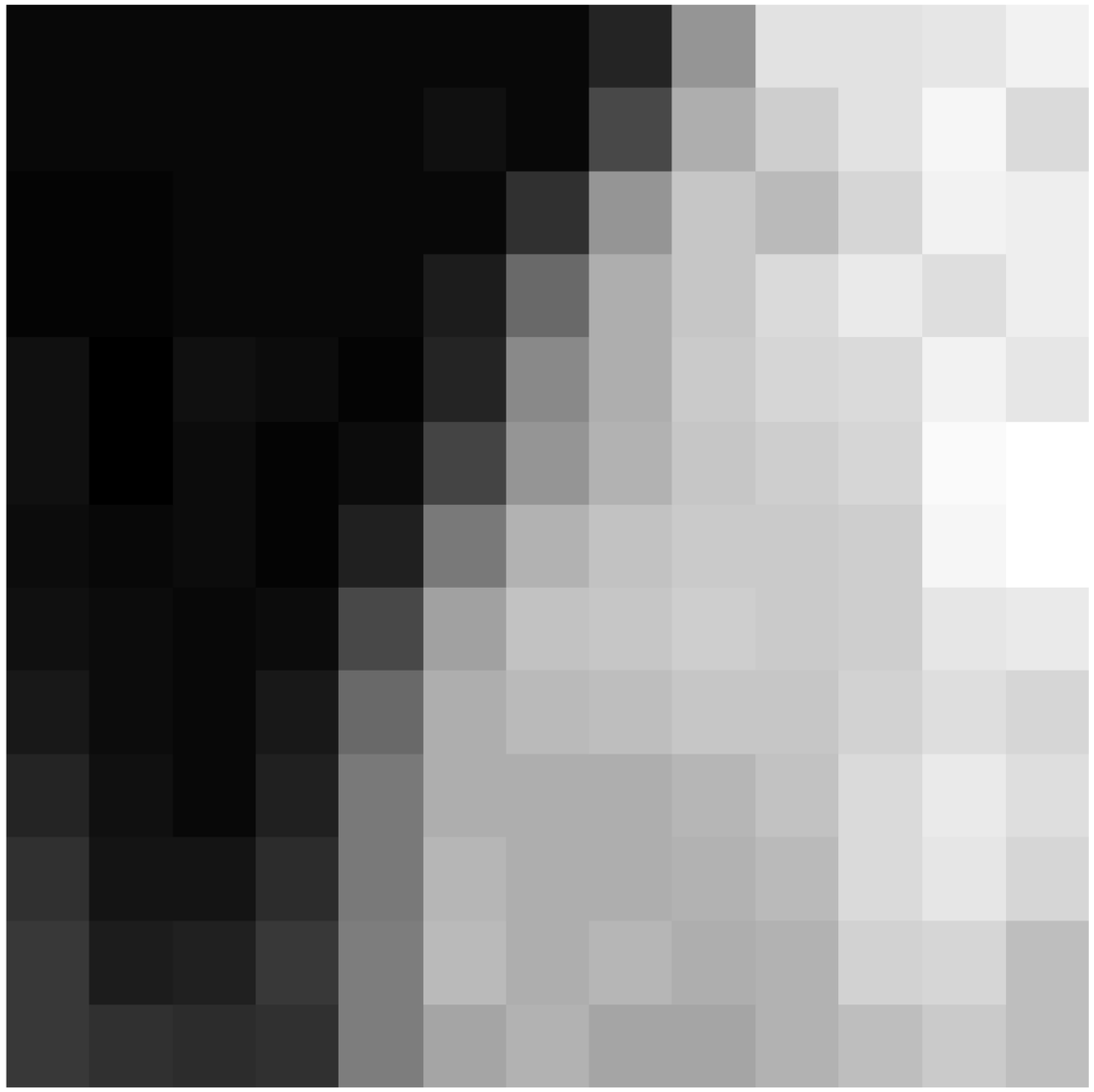
Gaussian Filter

Iterative application

$$I \star G_{\sigma_1}(x,y) \star G_{\sigma_2}(x,y) = I \star G_{\sqrt{\sigma_1^2+\sigma_2^2}}(x,y)$$

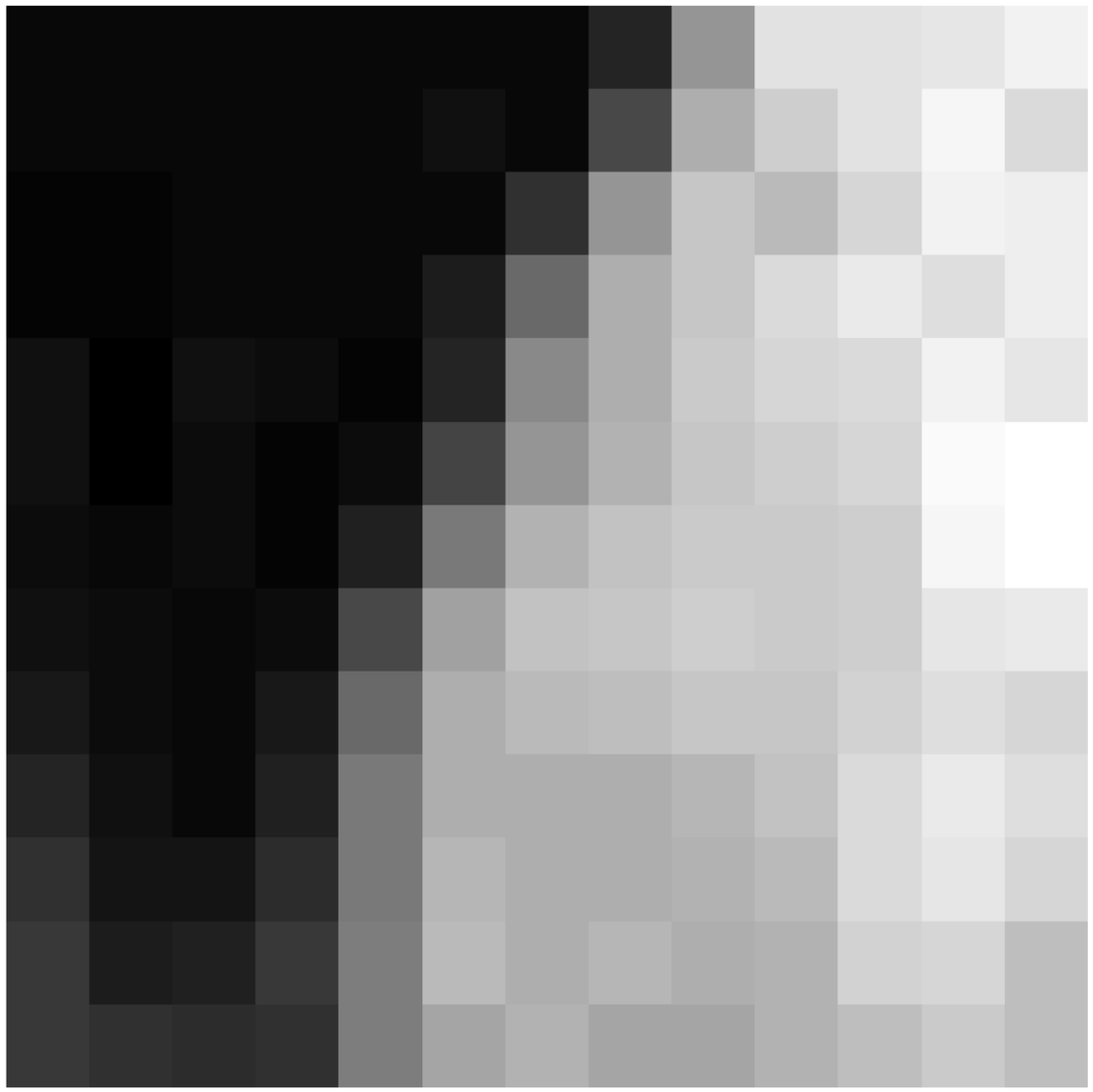
Useful if we need multiple smoothed version of an image
(more on this later)

Image Gradients



$$\nabla I(x,y) = \begin{pmatrix} I'_x \\ I'_y \end{pmatrix}$$

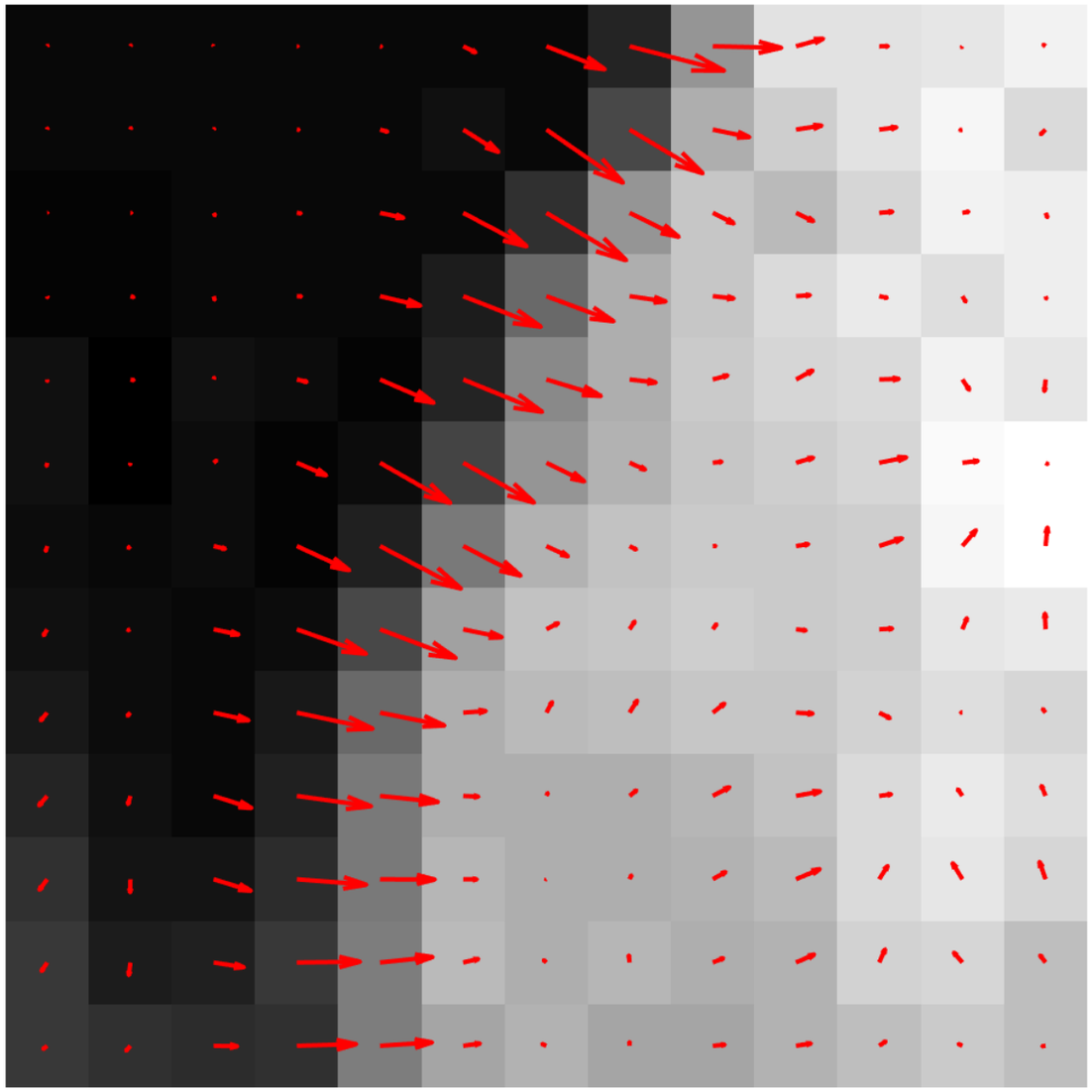
Image Gradients



$$\nabla I(x,y) = \begin{pmatrix} I'_x \\ I'_y \end{pmatrix}$$

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

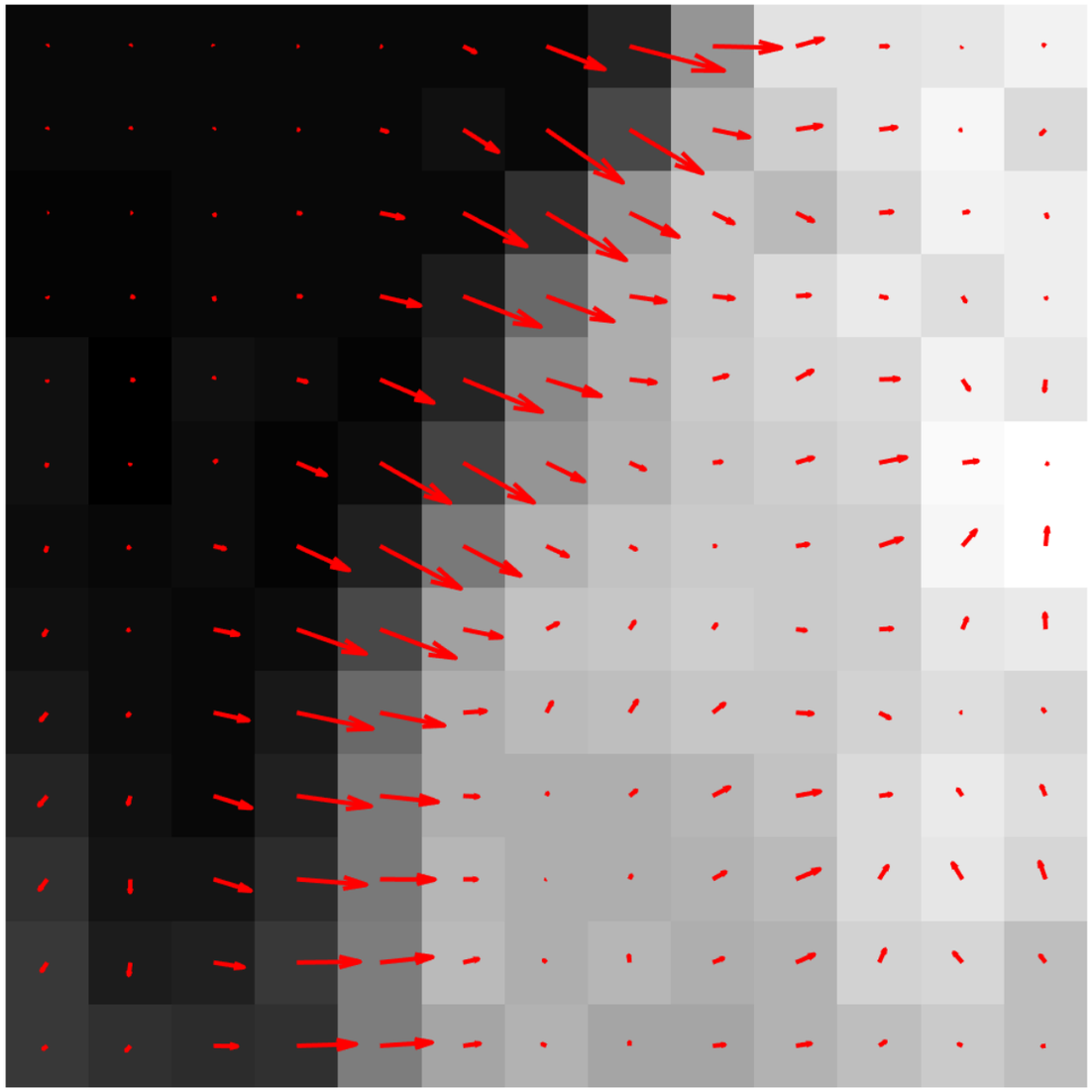
Image Gradients



$$\nabla I(x,y) = \begin{pmatrix} I'_x \\ I'_y \end{pmatrix}$$

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

Image Gradients

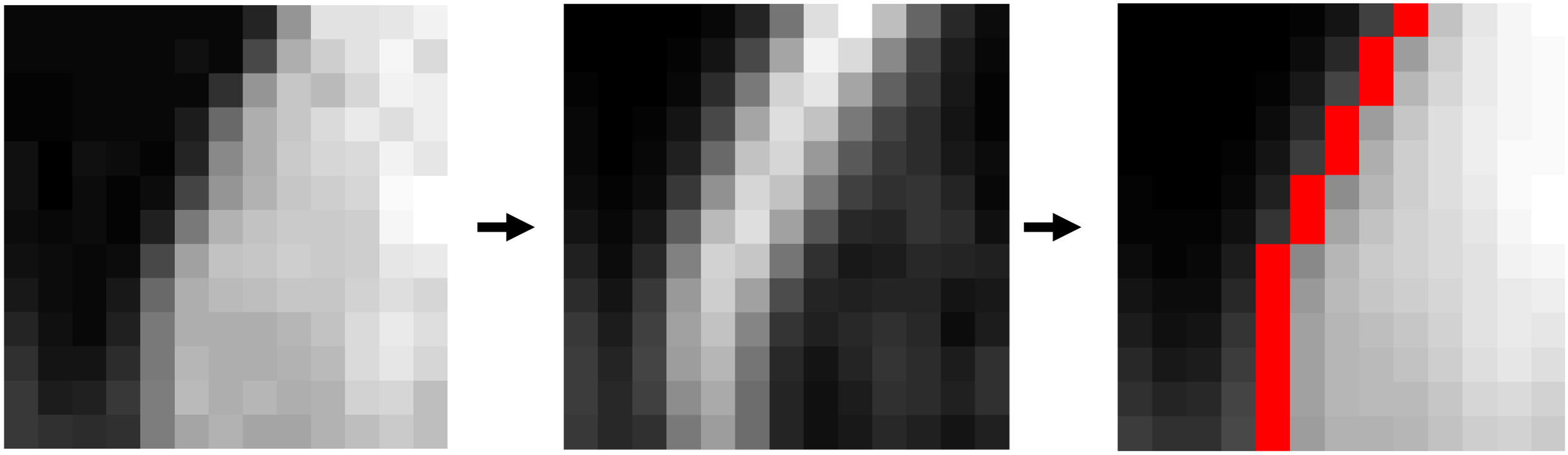


$$\nabla I(x,y) = \begin{pmatrix} I'_x \\ I'_y \end{pmatrix}$$

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

$$I'_x = I \star \begin{pmatrix} -0.5 & 0 & 0.5 \end{pmatrix}$$

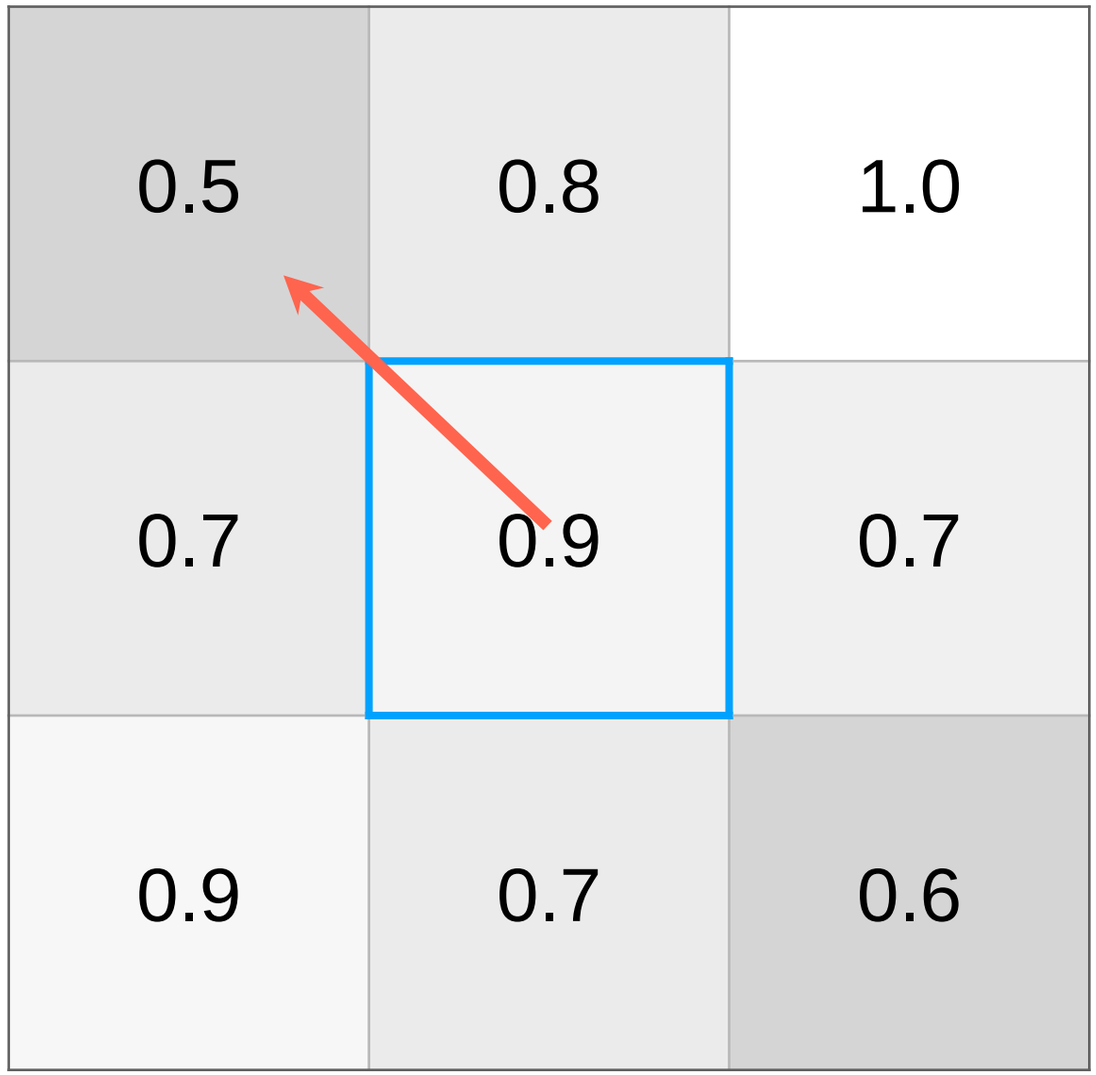
Application: Edge Detection



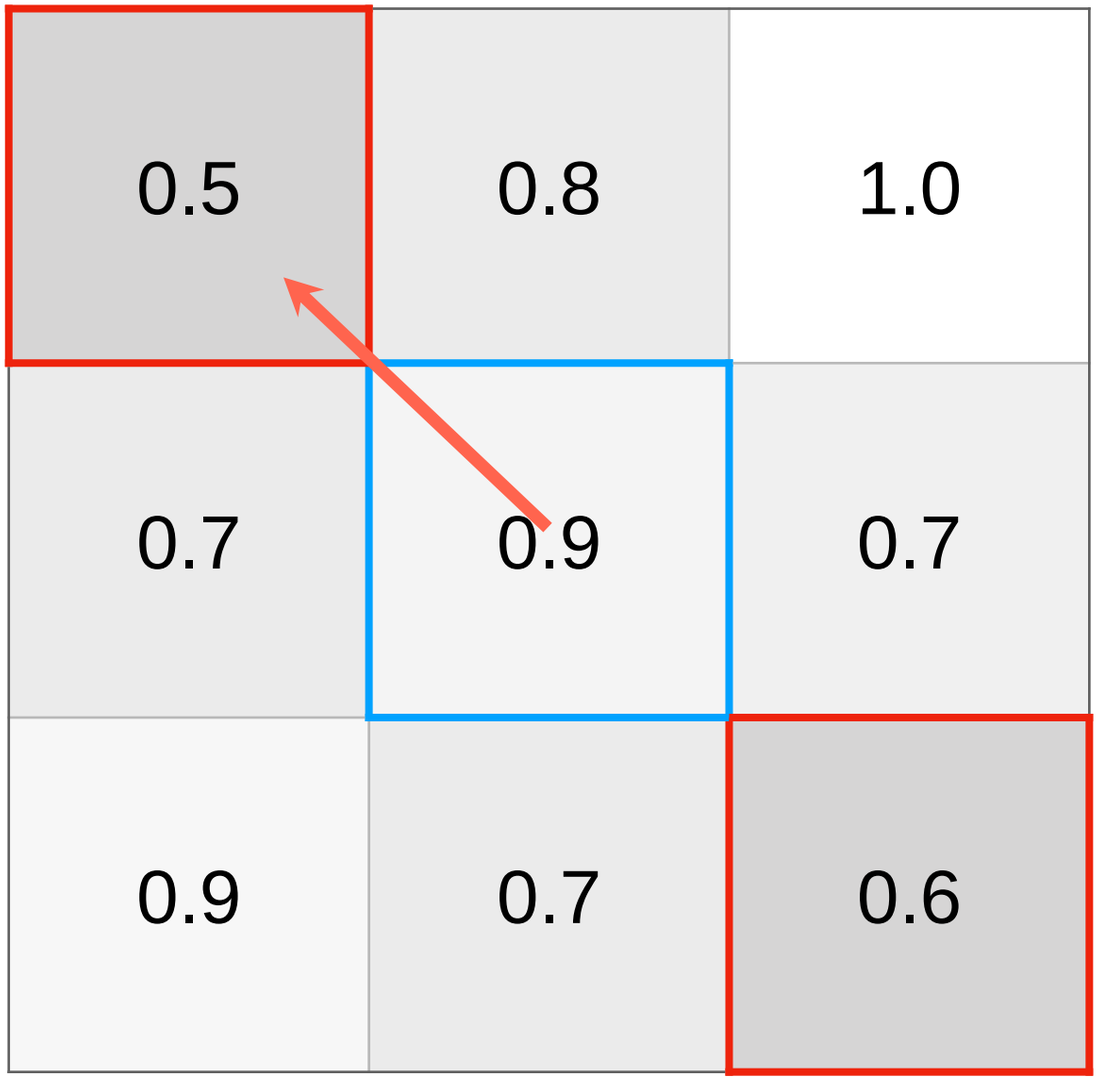
Magnitude of gradients

Thresholding and non-maximum suppression

Non-Maximum Suppression

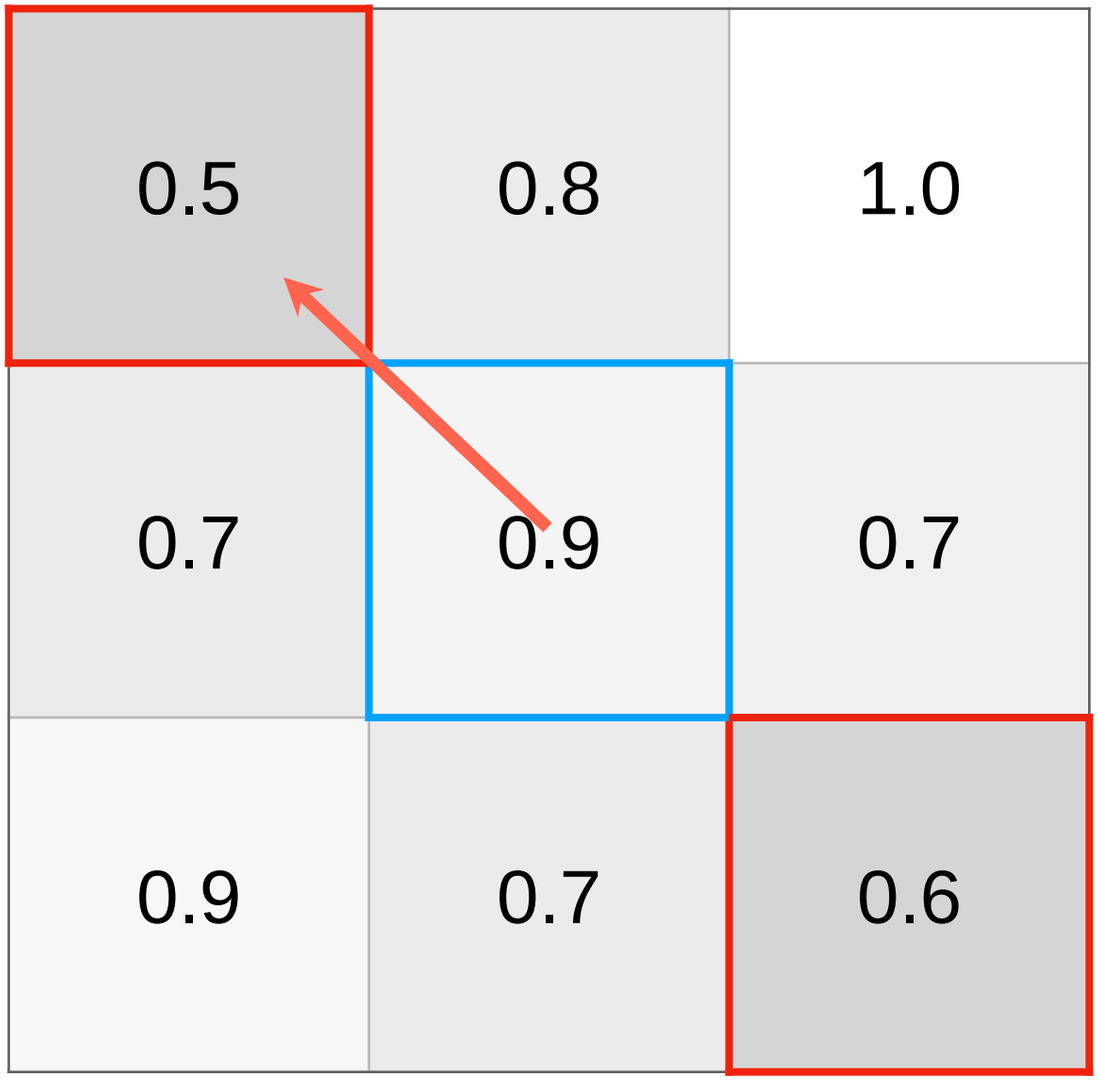


Non-Maximum Suppression



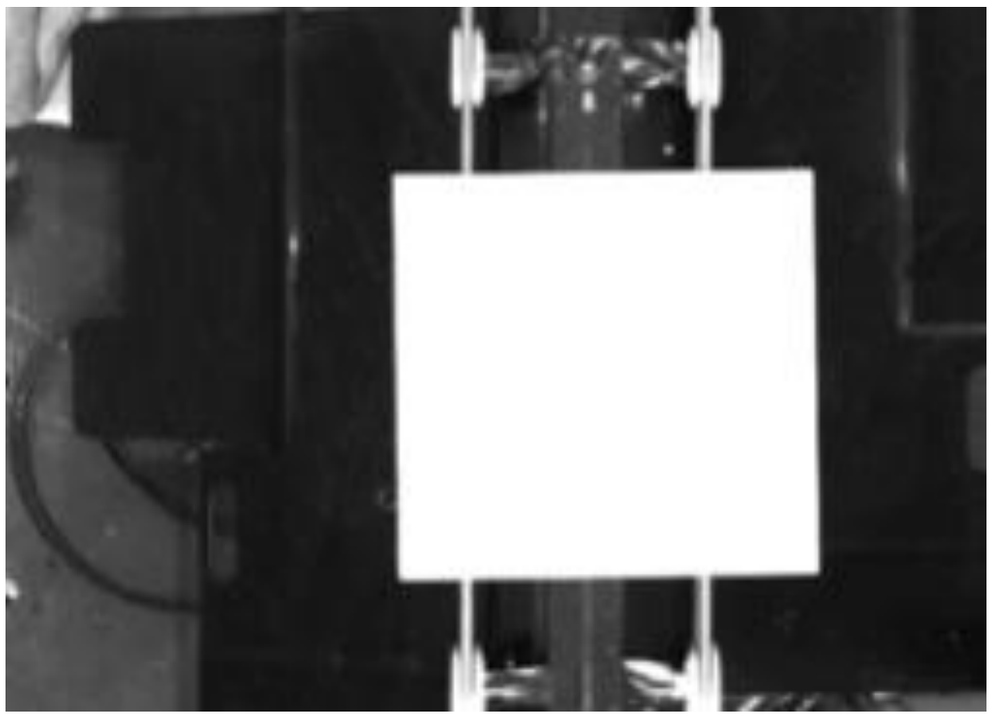
Only compare to neighbors along the gradient.

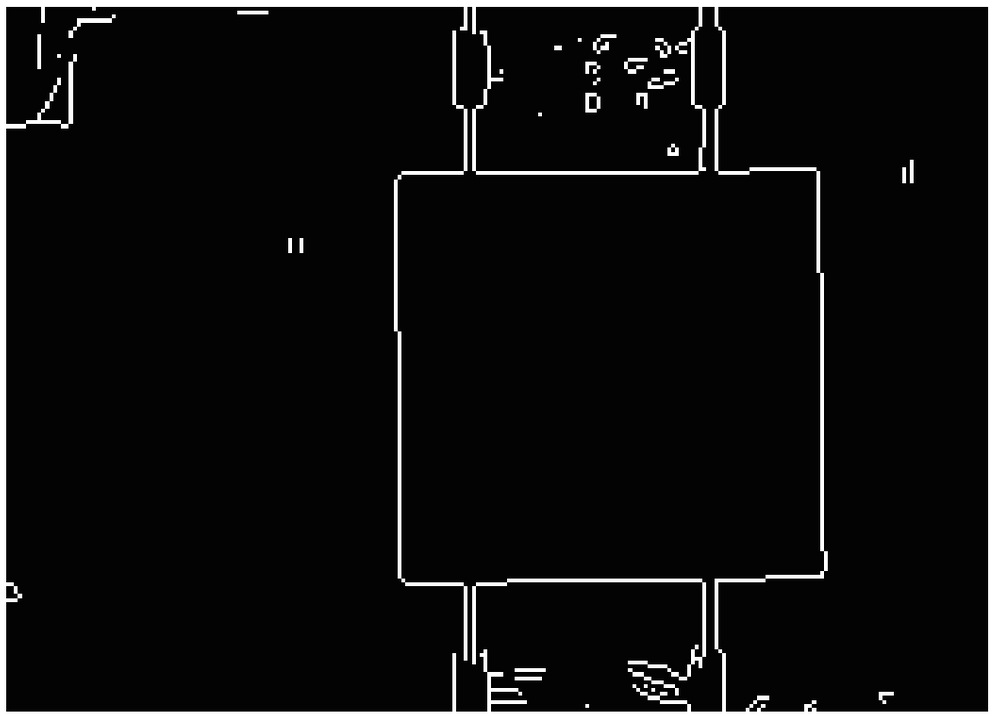
Non-Maximum Suppression



Only compare to neighbors along the gradient.
Keep if larger response than these.

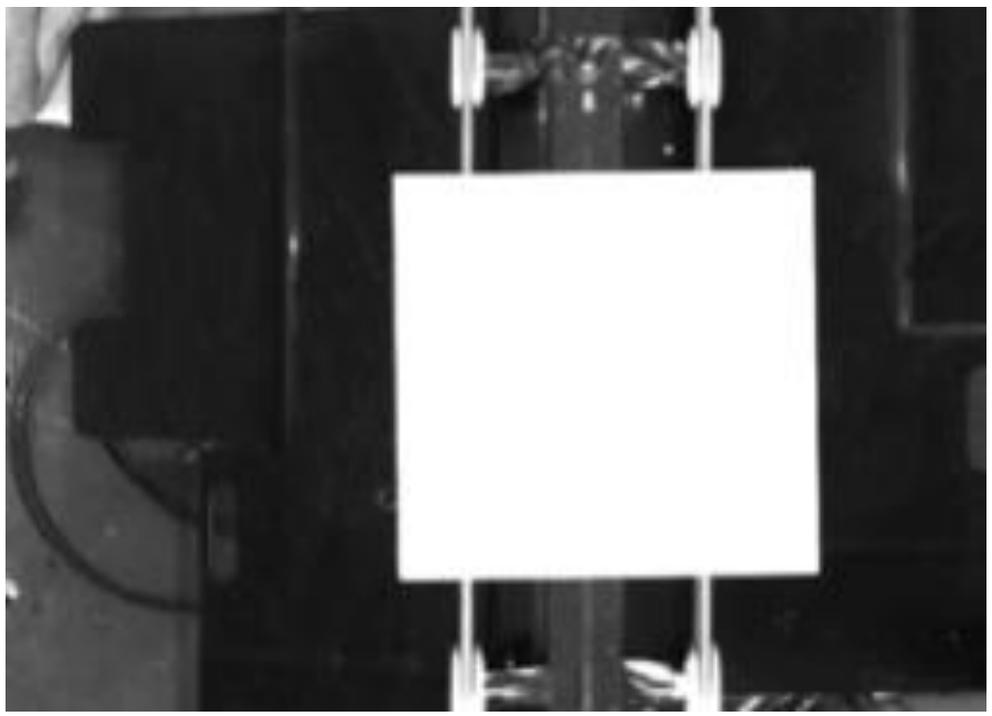
Application: Edge Detection

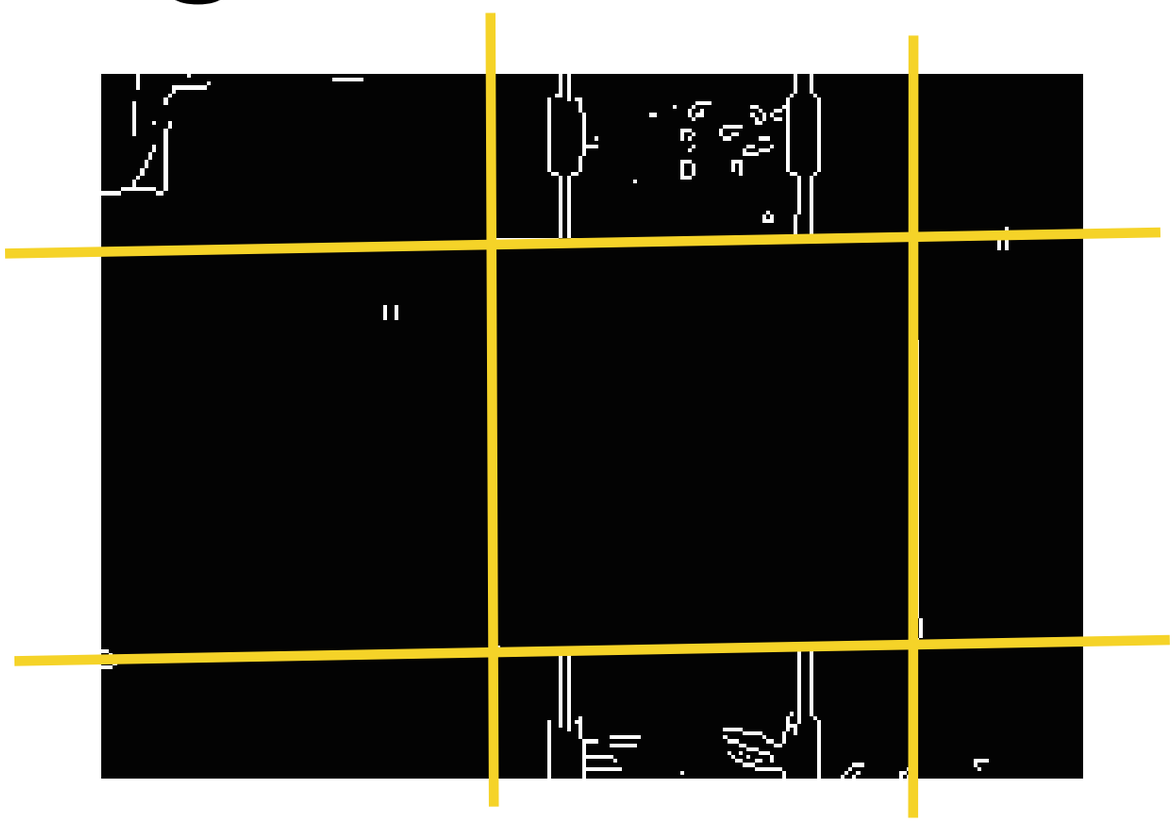




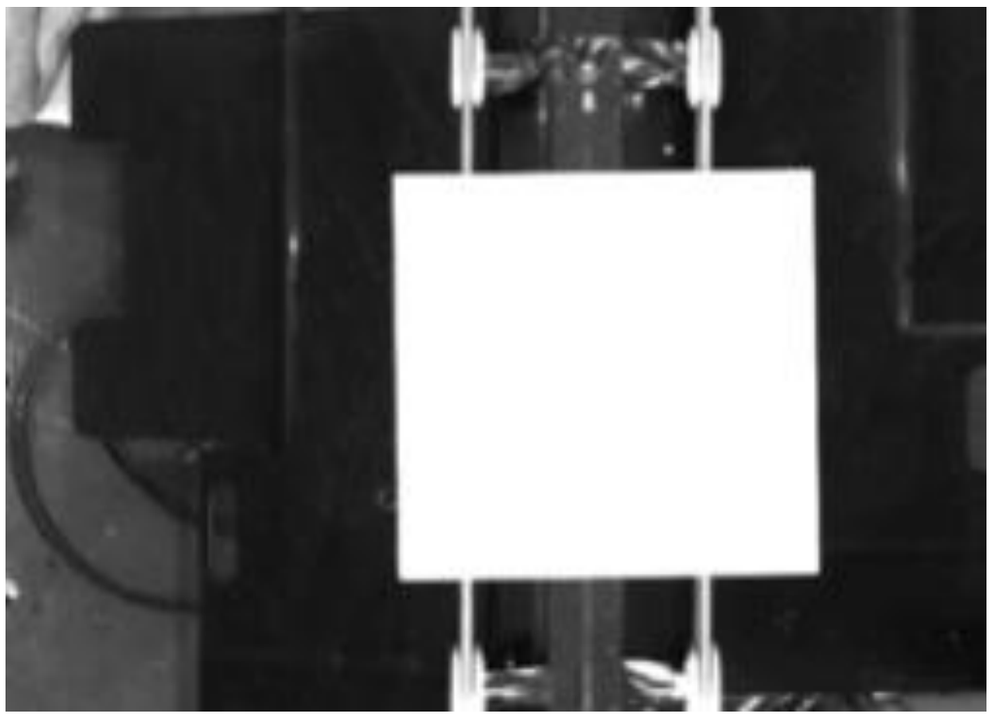
Using Matlab with default thresholds

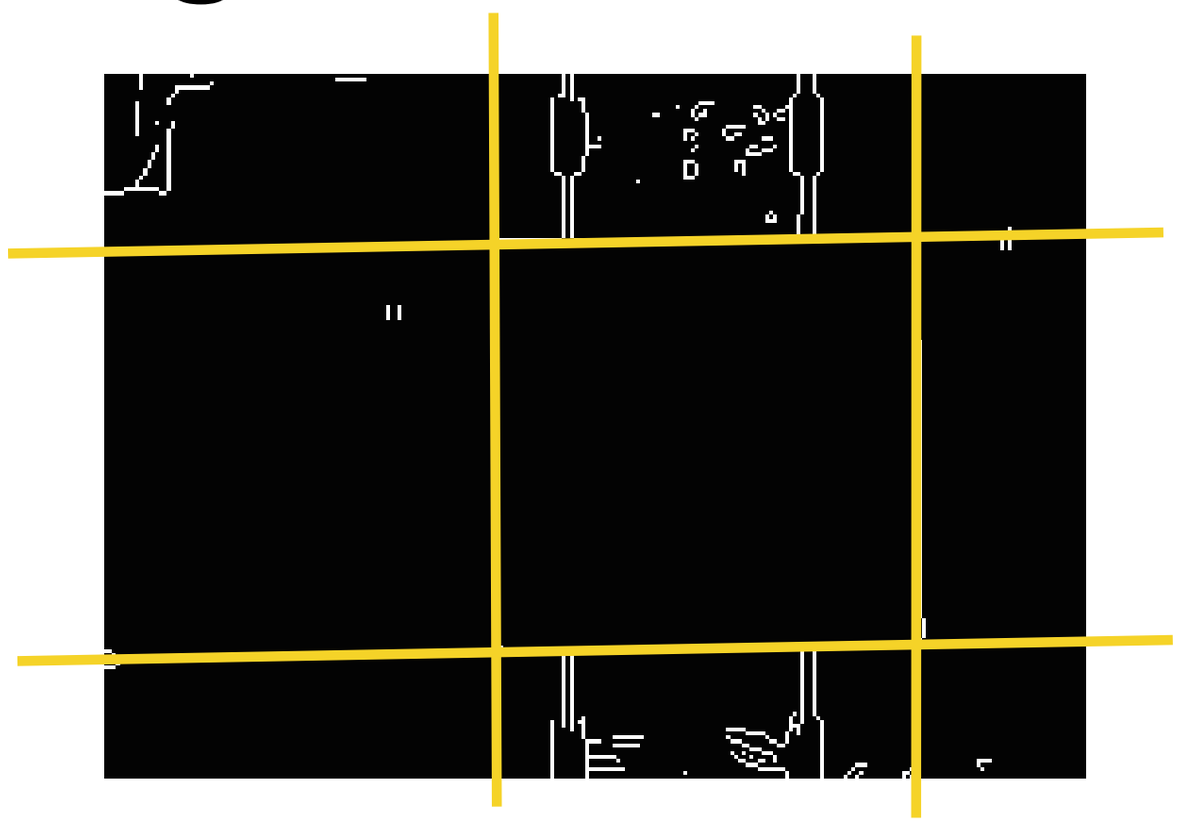
Application: Edge Detection





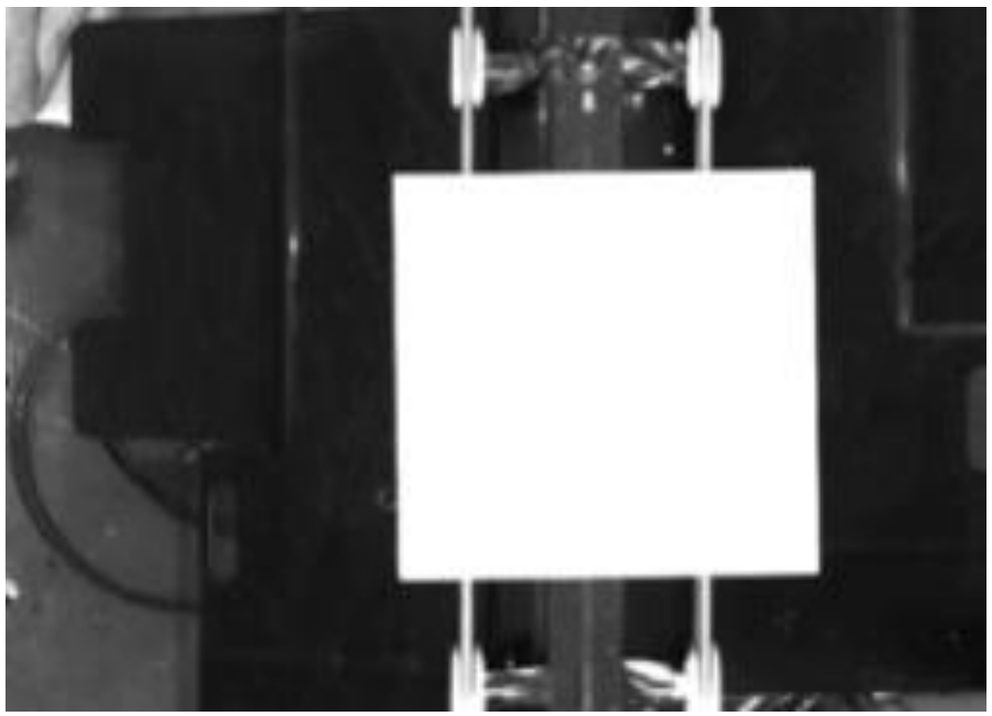
Application: Edge Detection

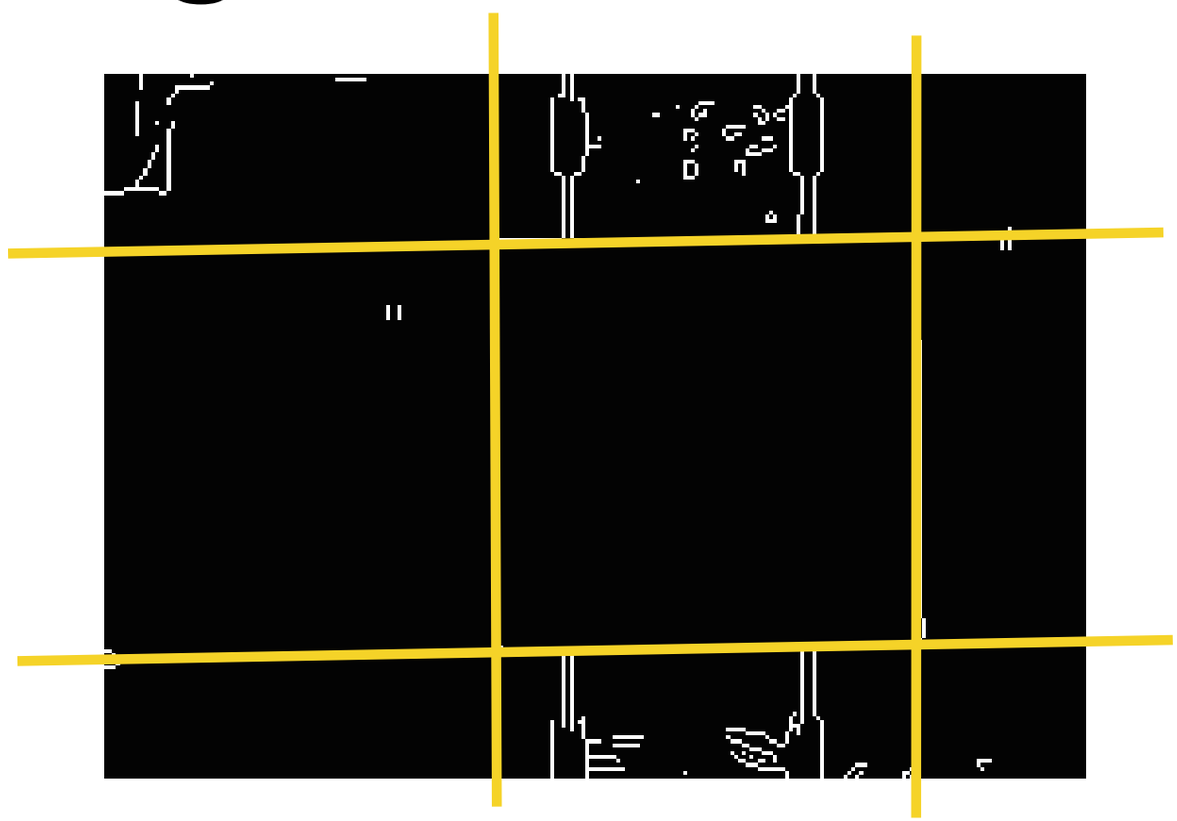


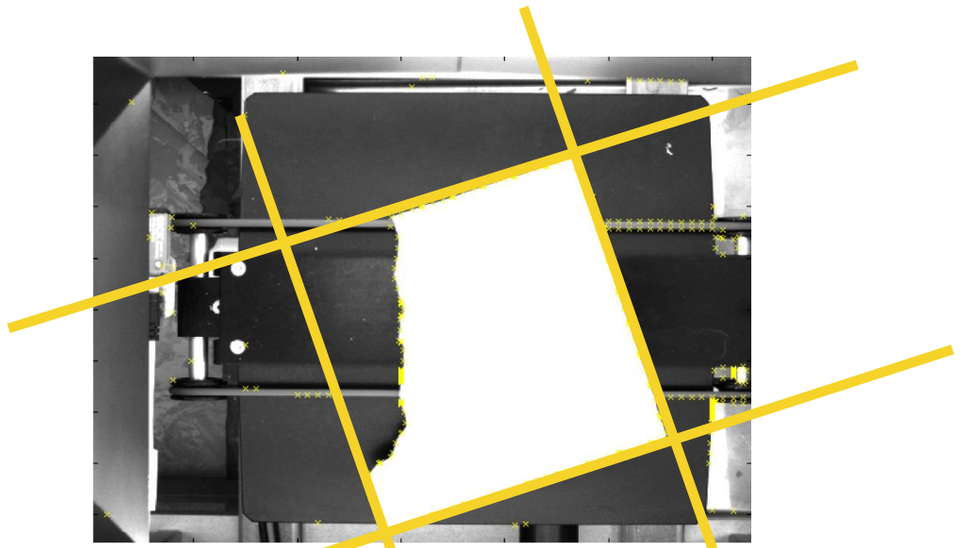


Fitting lines into the remaining pixels (see Lecture 7)

Application: Edge Detection







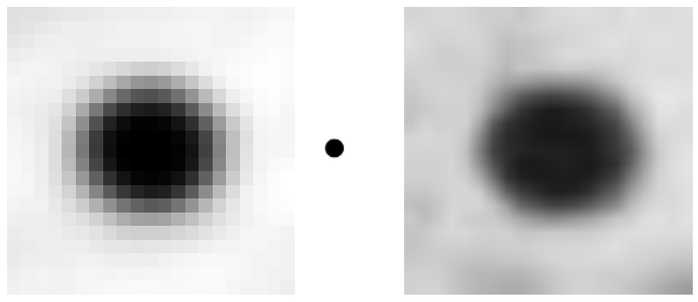
Similarity Measures

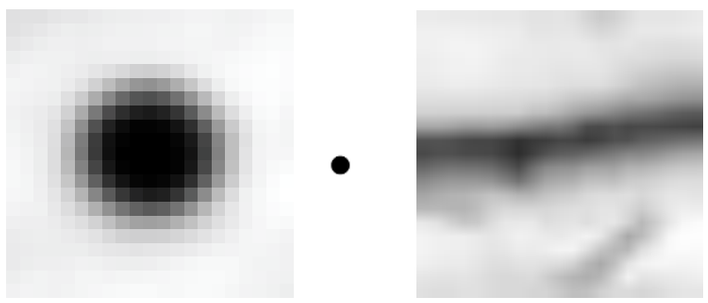
Measuring Similarity

Measuring similarity between images / patches central problem

Measuring Similarity

Measuring similarity between images / patches central problem





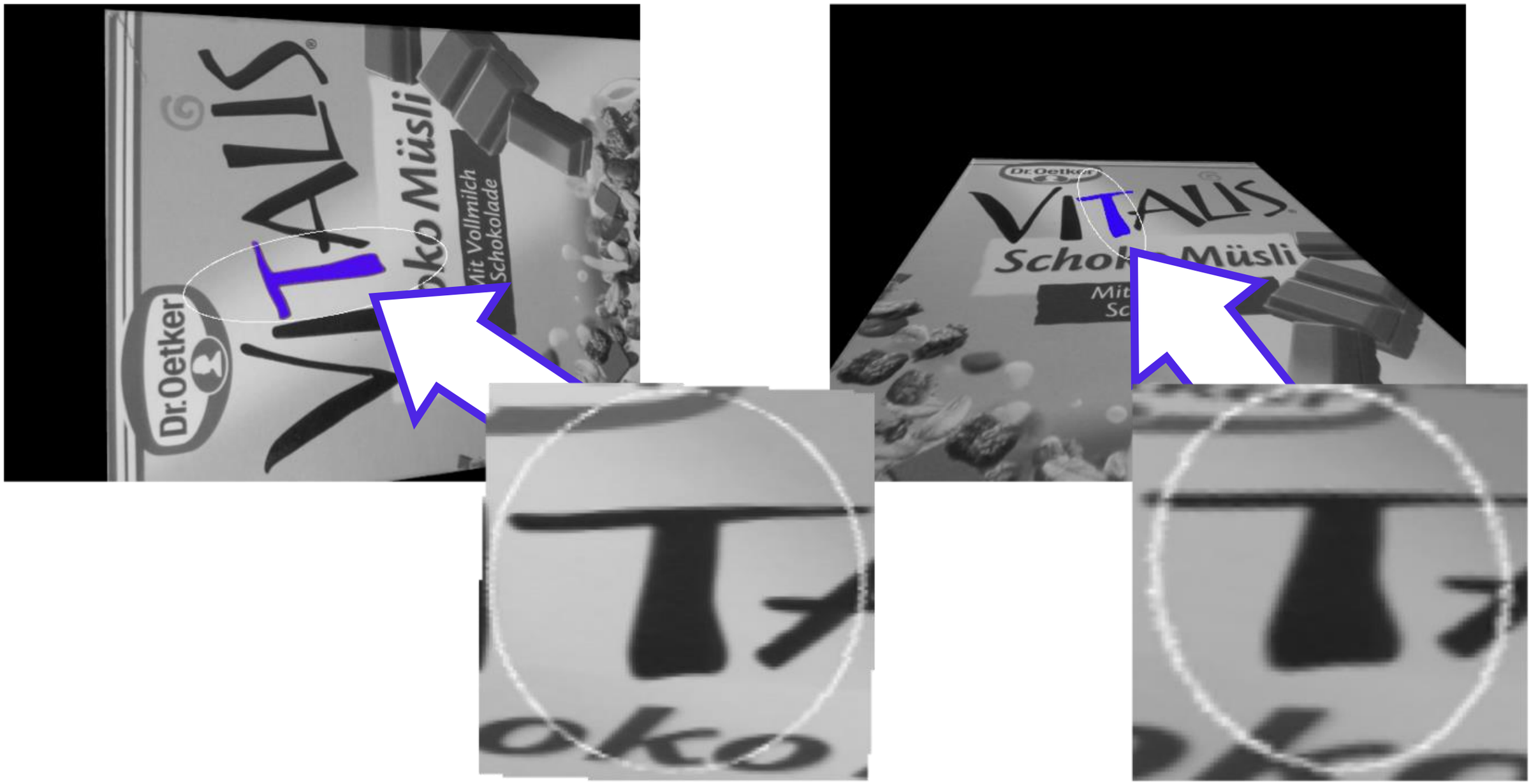
Measuring Similarity

Measuring similarity between images / patches central problem

slide credit: Marc Pollefeys, Kevin Köser

Measuring Similarity

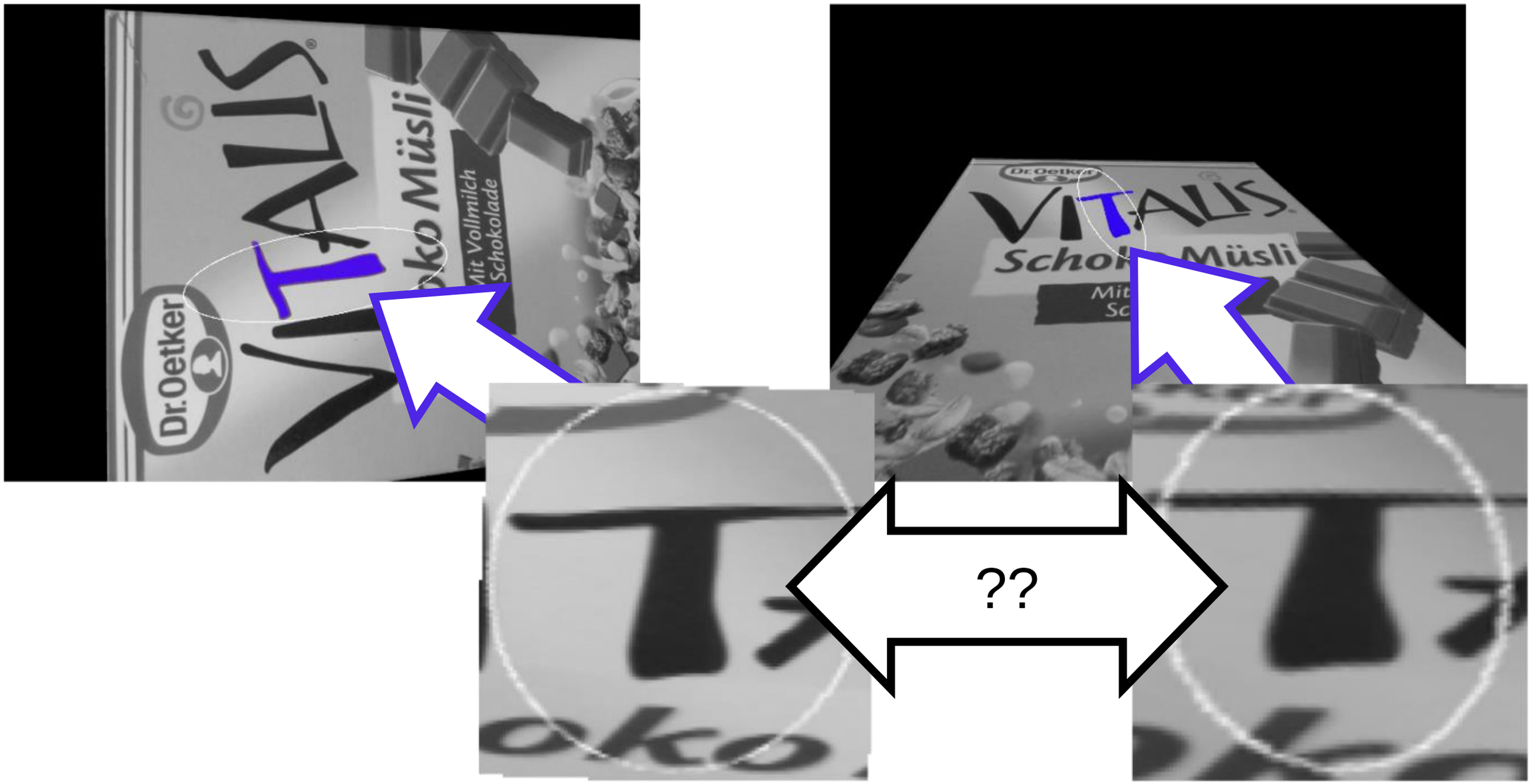
Measuring similarity between images / patches central problem



slide credit: Marc Pollefeys, Kevin Köser

Measuring Similarity

Measuring similarity between images / patches central problem



slide credit: Marc Pollefeys, Kevin Köser

Measuring Similarity

Covariance:

$P_1 =$ 

$P_2 =$ 

Measuring Similarity

Covariance:

$P_1 =$ 

$P_2 =$ 

$$\mathrm{Cov}\left(P_1, P_2\right) = \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(P_2 - \mu_2 \mathbb{1}\right)$$

Measuring Similarity

Covariance:

$P_1 =$ 

$P_2 =$ 

$$\begin{aligned}\operatorname{Cov}(P_1, P_2) &= \frac{1}{N_{\text{elements}}} (P_1 - \mu_1 \mathbb{1}) (P_2 - \mu_2 \mathbb{1}) \\ &= \frac{1}{N_{\text{elements}}} ((P_1 - \mu_1 \mathbb{1}) P_2 - \mu_2 (P_1 \mathbb{1} - \mu_1 \mathbb{1}\mathbb{1}))\end{aligned}$$

Measuring Similarity

Covariance:

$P_1 =$ 

$P_2 =$ 

$$
\begin{aligned}
\operatorname{Cov}(P_1, P_2) &= \frac{1}{N_{\text{elements}}} (P_1 - \mu_1 \mathbb{1}) (P_2 - \mu_2 \mathbb{1}) \\
&= \frac{1}{N_{\text{elements}}} ((P_1 - \mu_1 \mathbb{1}) P_2 - \mu_2 (P_1 \mathbb{1} - \mu_1 \mathbb{1}\mathbb{1})) \\
&= \frac{1}{N_{\text{elements}}} ((P_1 - \mu_1 \mathbb{1}) P_2 - \mu_2 (N_{\text{elements}} \cdot \mu_1 - \mu_1 \cdot N_{\text{elements}}))
\end{aligned}
$$

Measuring Similarity

Covariance:

$P_1 =$ 

$P_2 =$ 

$$
\begin{aligned}
\operatorname{Cov}(P_1, P_2) &= \frac{1}{N_{\text{elements}}} (P_1 - \mu_1 \mathbb{1}) (P_2 - \mu_2 \mathbb{1}) \\
&= \frac{1}{N_{\text{elements}}} ((P_1 - \mu_1 \mathbb{1}) P_2 - \mu_2 (P_1 \mathbb{1} - \mu_1 \mathbb{1}\mathbb{1})) \\
&= \frac{1}{N_{\text{elements}}} ((P_1 - \mu_1 \mathbb{1}) P_2 - \mu_2 (N_{\text{elements}} \cdot \mu_1 - \mu_1 \cdot N_{\text{elements}})) \\
&= \frac{1}{N_{\text{elements}}} (P_1 - \mu_1 \mathbb{1}) P_2
\end{aligned}
$$

Measuring Similarity

Covariance:

$P_1 =$ 

$P_2 =$ 

$$
\begin{aligned}
\mathrm{Cov}\left(P_1, P_2\right) &= \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(P_2 - \mu_2 \mathbb{1}\right) \\
&= \frac{1}{N_{\text{elements}}} \left(\left(P_1 - \mu_1 \mathbb{1}\right) P_2 - \mu_2 \left(P_1 \mathbb{1} - \mu_1 \mathbb{1}\mathbb{1}\right)\right) \\
&= \frac{1}{N_{\text{elements}}} \left(\left(P_1 - \mu_1 \mathbb{1}\right) P_2 - \mu_2 \left(N_{\text{elements}} \cdot \mu_1 - \mu_1 \cdot N_{\text{elements}}\right)\right) \\
&= \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) P_2
\end{aligned}
$$

We have used this before!

Measuring Similarity

Is Covariance a good similarity measure?

Measuring Similarity

Is Covariance a good similarity measure?

$P_1 =$ 

$P_2 =$

$= k \cdot P_1$

Measuring Similarity

Is Covariance a good similarity measure?

$$P_1 = \quad P_2 = \quad = k \cdot P_1$$

$$\mathrm{Cov}\left(P_1, P_2\right) \quad = \quad \frac{1}{N_{\mathrm{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(k \cdot P_1 - k \cdot \mu_1 \mathbb{1}\right)$$

Measuring Similarity

Is Covariance a good similarity measure?

$P_1 =$ 

$P_2 =$

$= k \cdot P_1$

$$\begin{aligned} \mathrm{Cov}\left(P_1, P_2\right) &= \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(k \cdot P_1 - k \cdot \mu_1 \mathbb{1}\right) \\ &= k \cdot \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(P_1 - \mu_1 \mathbb{1}\right) \end{aligned}$$

Measuring Similarity

Is Covariance a good similarity measure?

$P_1 =$ 

$P_2 =$

$= k \cdot P_1$

$$
\begin{aligned}
\mathrm{Cov}\left(P_1, P_2\right) &= \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(k \cdot P_1 - k \cdot \mu_1 \mathbb{1}\right) \\
&= k \cdot \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(P_1 - \mu_1 \mathbb{1}\right) \\
&= k \cdot \mathrm{Cov}\left(P_1, P_1\right)
\end{aligned}
$$

Measuring Similarity

Is Covariance a good similarity measure?

$P_1 =$ 

$P_2 =$

$= k \cdot P_1$

$$
\begin{aligned}
\mathrm{Cov}\left(P_1, P_2\right) &= \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(k \cdot P_1 - k \cdot \mu_1 \mathbb{1}\right) \\
&= k \cdot \frac{1}{N_{\text{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(P_1 - \mu_1 \mathbb{1}\right) \\
&= k \cdot \mathrm{Cov}\left(P_1, P_1\right) > \mathrm{Cov}\left(P_1, P_1\right)
\end{aligned}
$$

Measuring Similarity

Is Covariance a good similarity measure?

$P_1 =$ 

$P_2 =$

$= k \cdot P_1$

$$
\begin{aligned}
\mathrm{Cov}\left(P_1, P_2\right) &= \frac{1}{N_{\mathrm{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(k \cdot P_1 - k \cdot \mu_1 \mathbb{1}\right) \\
&= k \cdot \frac{1}{N_{\mathrm{elements}}} \left(P_1 - \mu_1 \mathbb{1}\right) \left(P_1 - \mu_1 \mathbb{1}\right) \\
&= k \cdot \mathrm{Cov}\left(P_1, P_1\right) > \mathrm{Cov}\left(P_1, P_1\right)
\end{aligned}
$$

This is not what we want!

Measuring Similarity

Correlation:

$$\mathrm{Corr}\,(P_1, P_2) = \frac{\mathrm{Cov}\,(P_1, P_2)}{\sqrt{\mathrm{Cov}\,(P_1, P_1)} \cdot \sqrt{\mathrm{Cov}\,(P_2, P_2)}}$$

Also known as **Zero-Mean Normalized Cross-Correlation** (ZNCC)

Measuring Similarity

When does ZNCC work?

slide credit: Marc Pollefeys, Kevin Köser

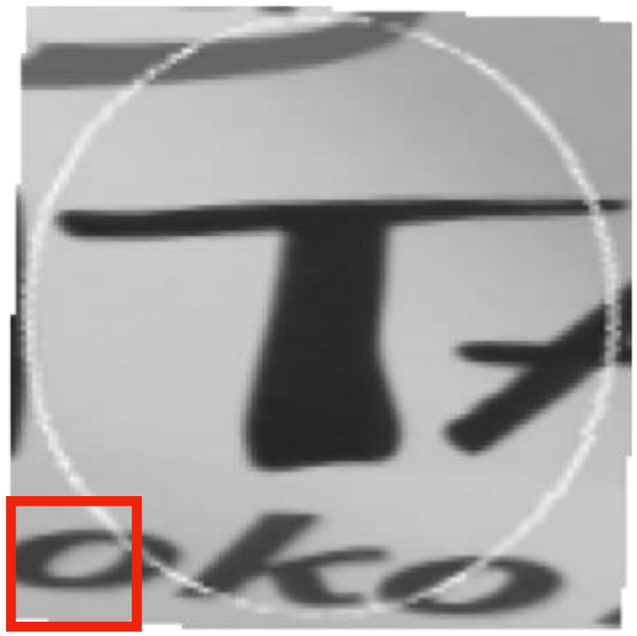
Measuring Similarity

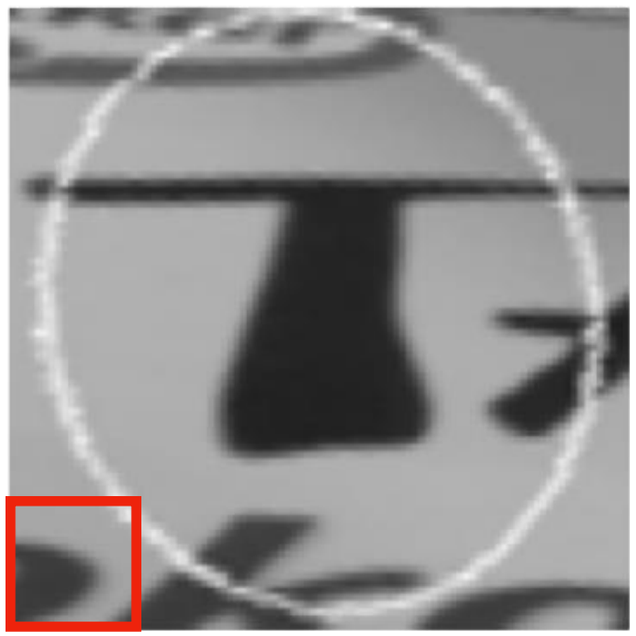
When does ZNCC work?

slide credit: Marc Pollefeys, Kevin Köser

Measuring Similarity

When does ZNCC work?

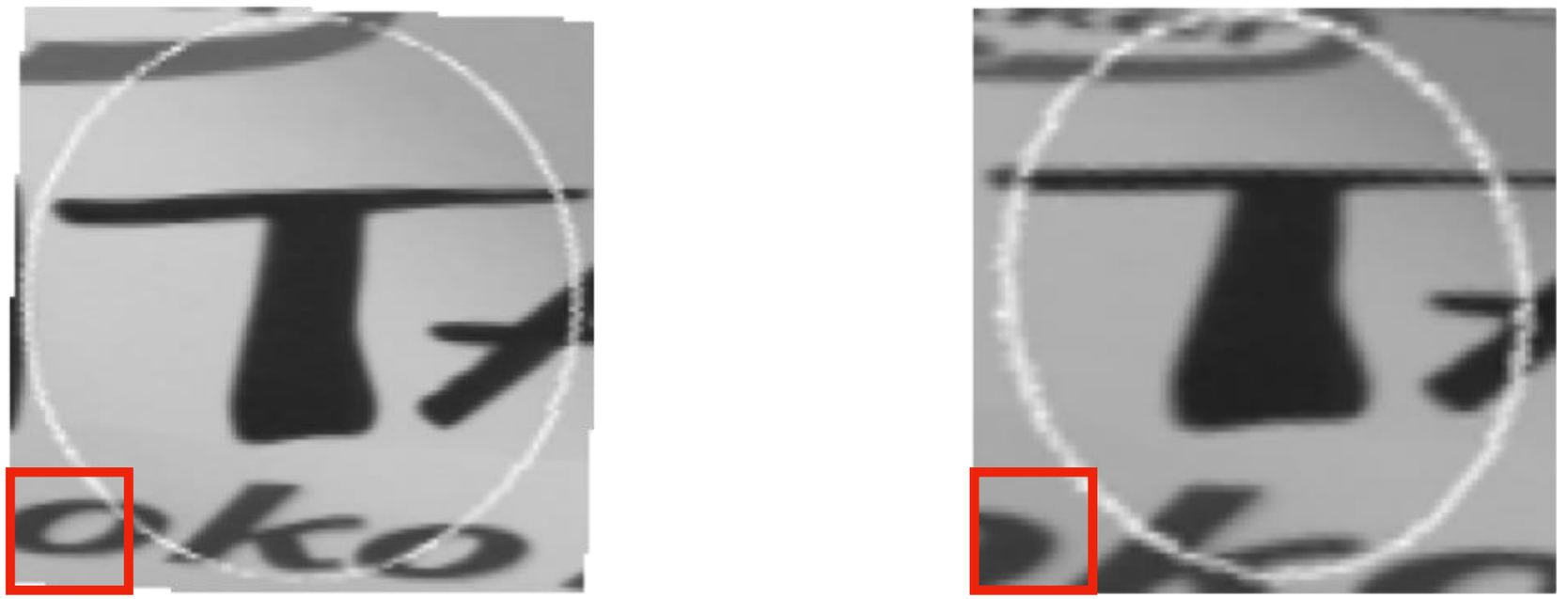




slide credit: Marc Pollefeys, Kevin Köser

Measuring Similarity

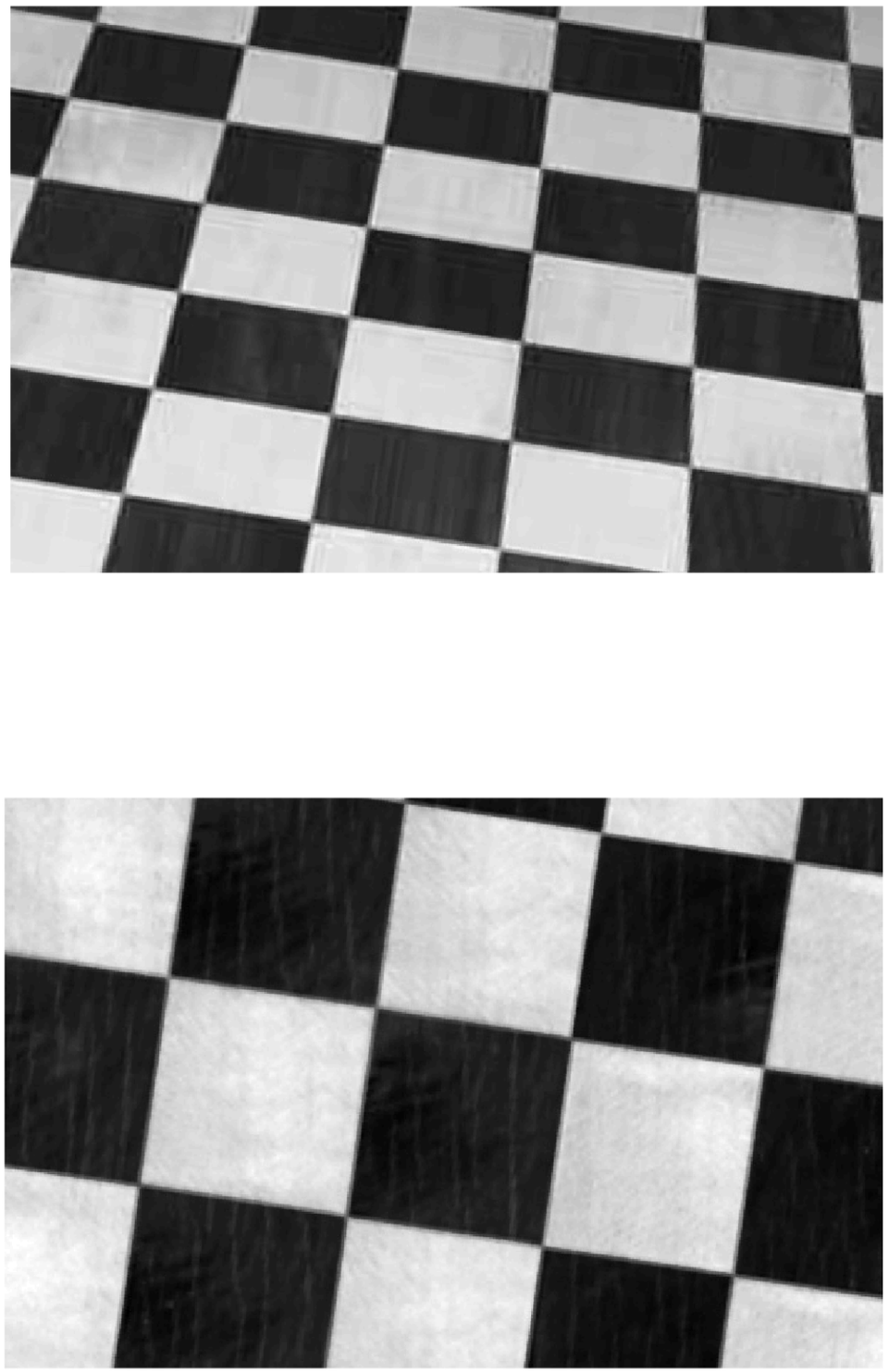
When does ZNCC work?



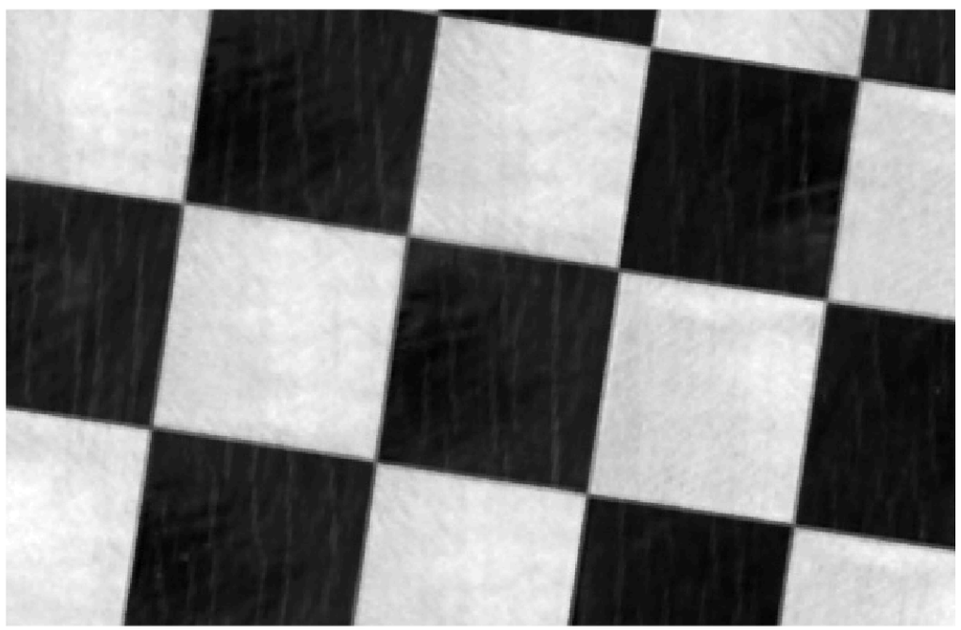
Pixels need to be (roughly) aligned!

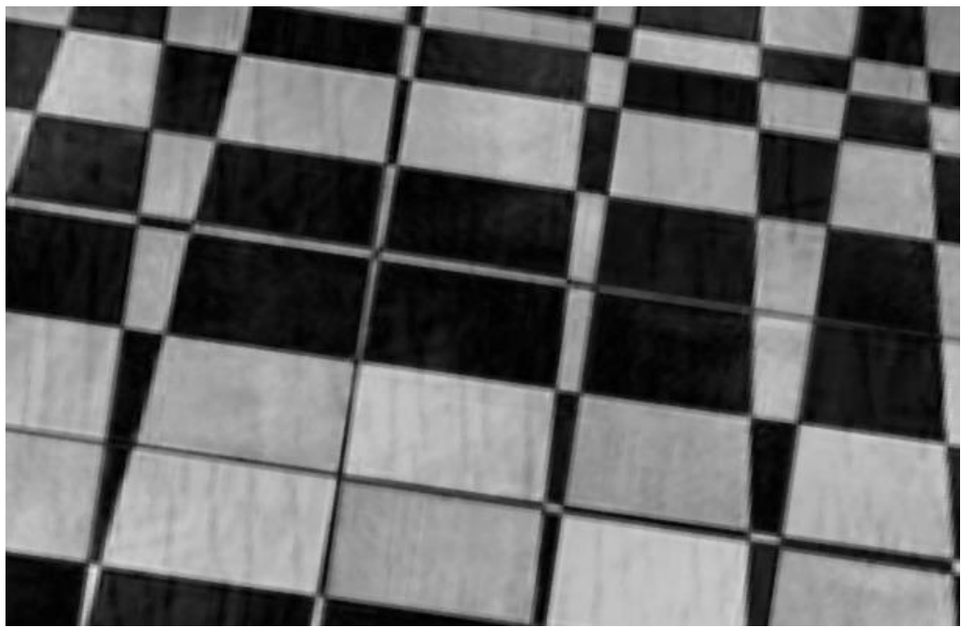
slide credit: Marc Pollefeys, Kevin Köser

Measuring Similarity



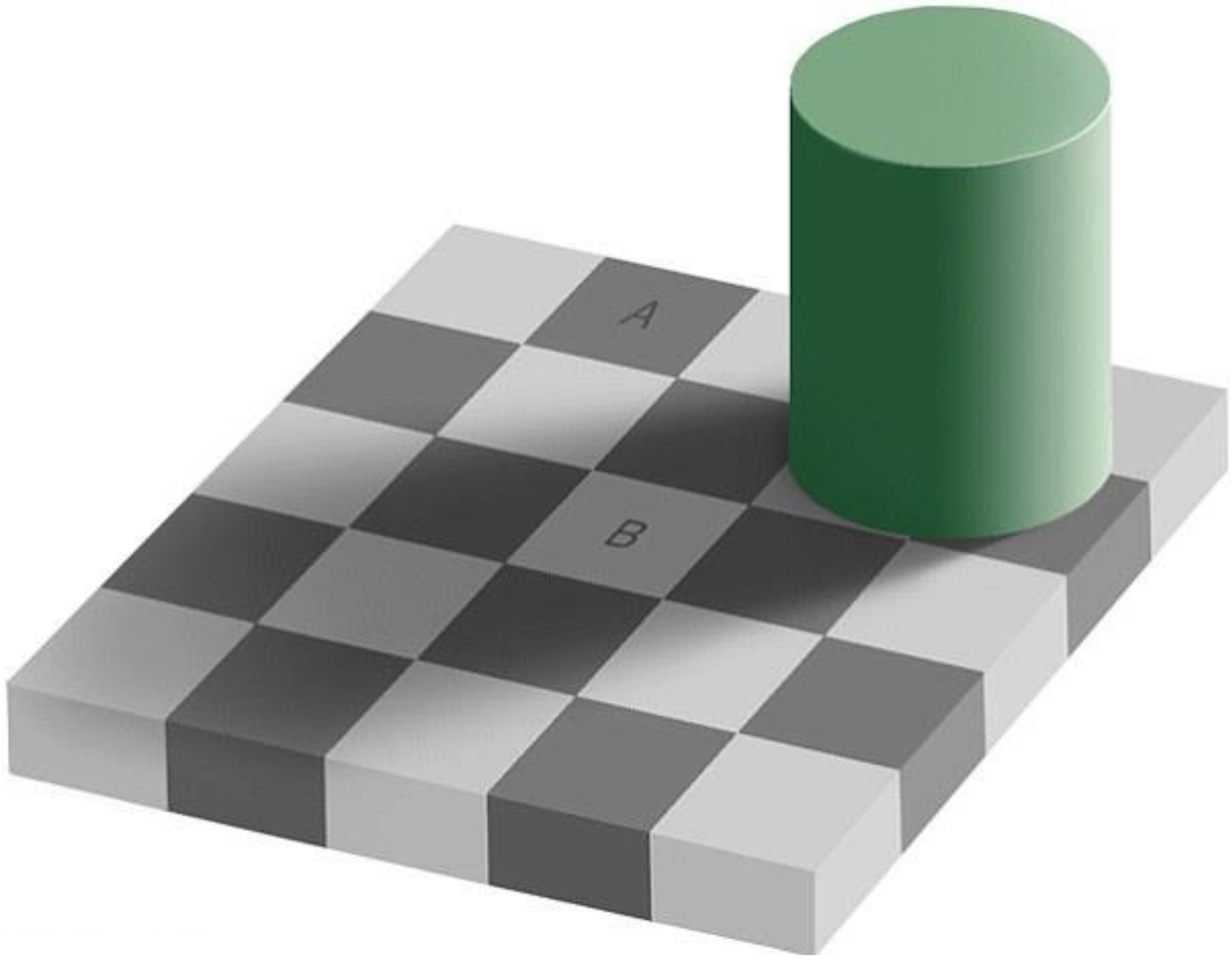
Measuring Similarity



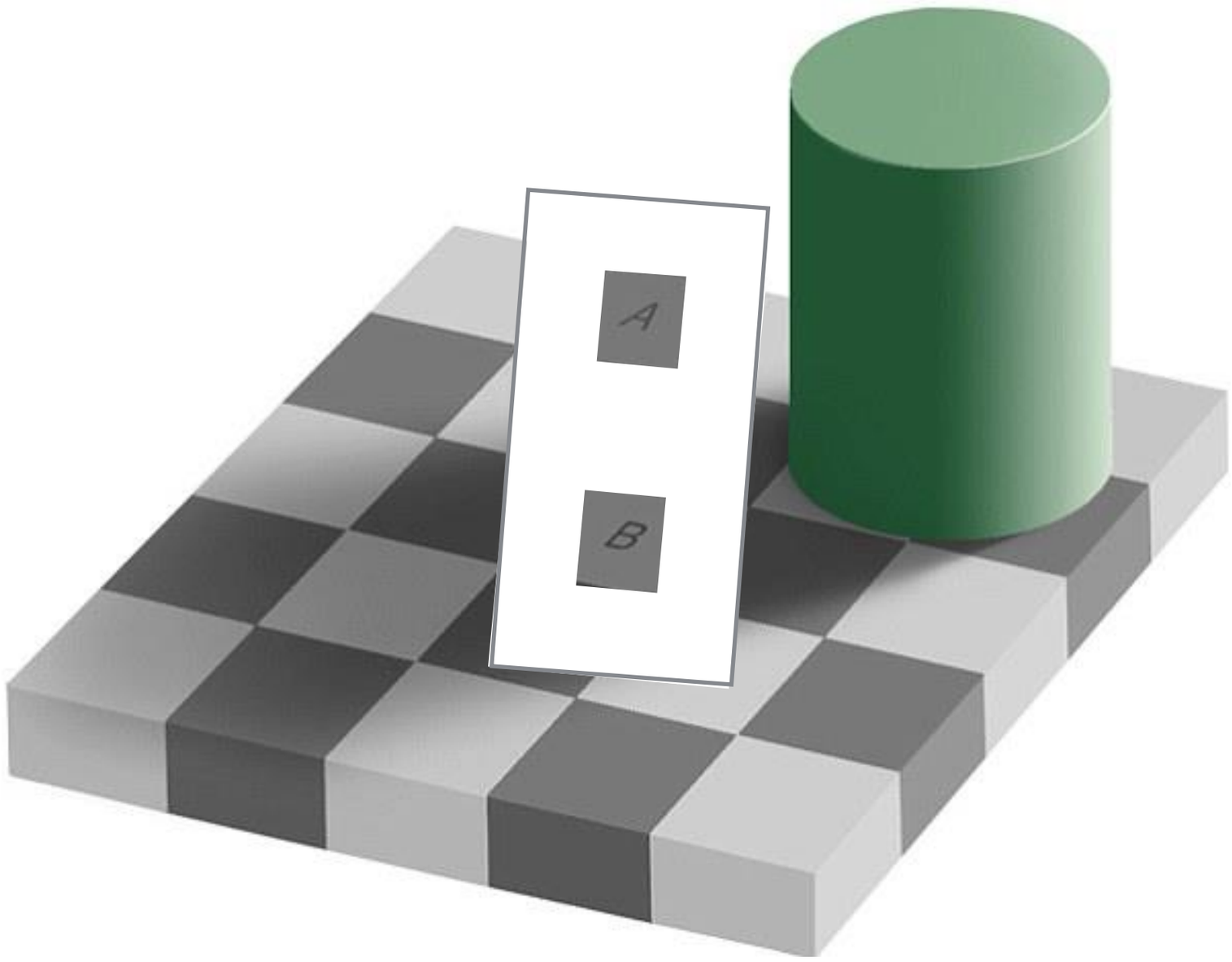


Difference image
black = no difference
white = difference

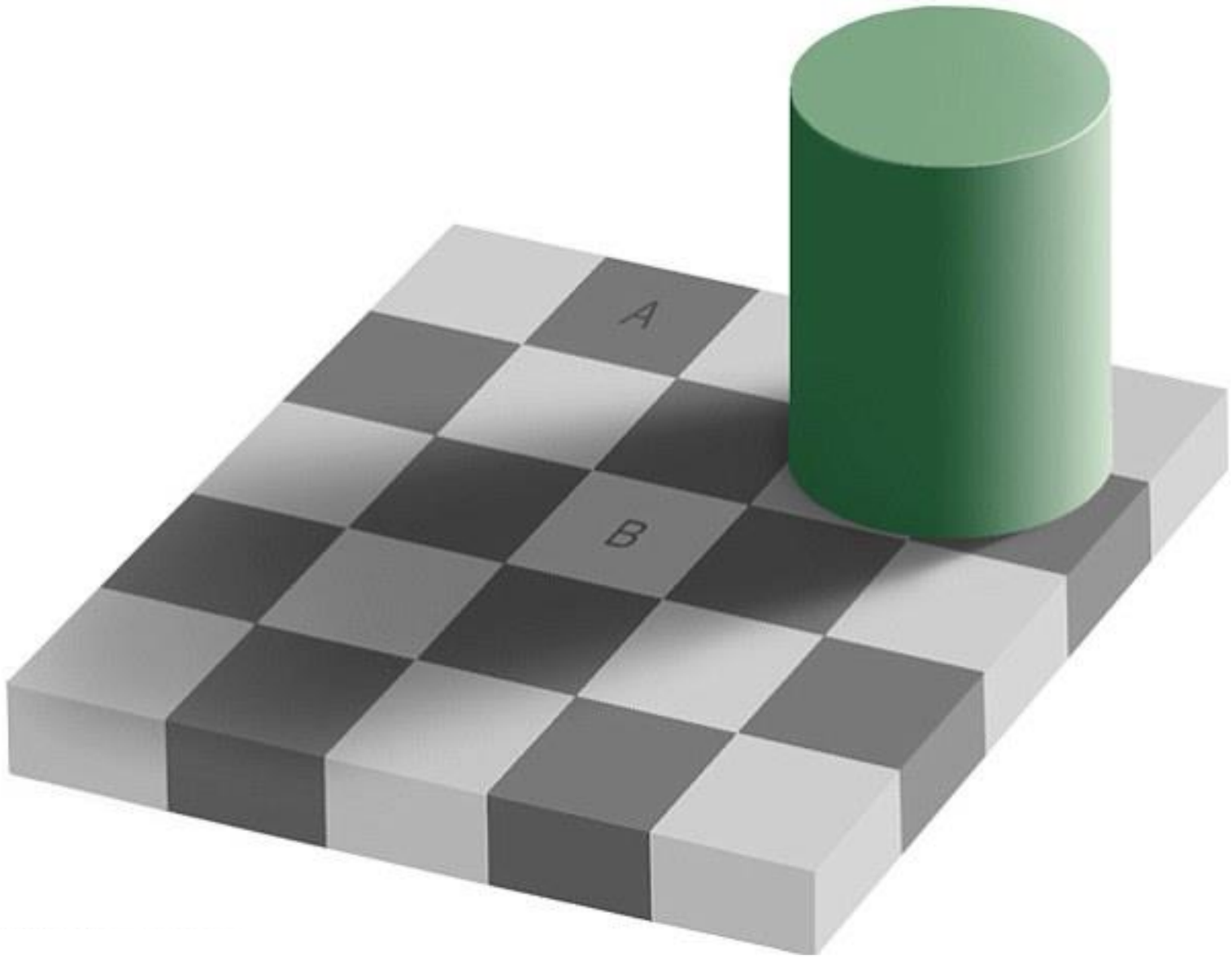
A Hint?



A Hint?

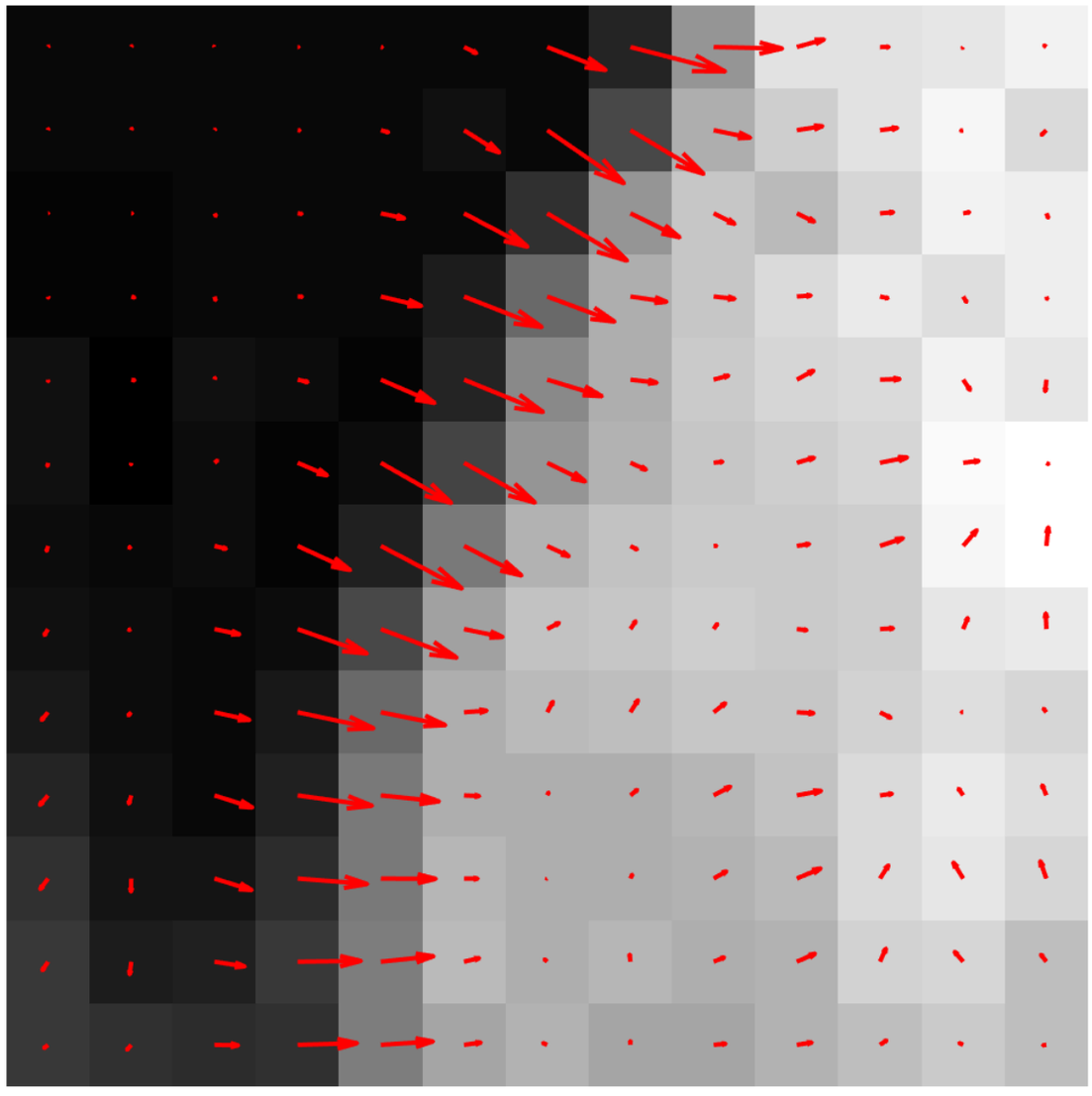


A Hint?

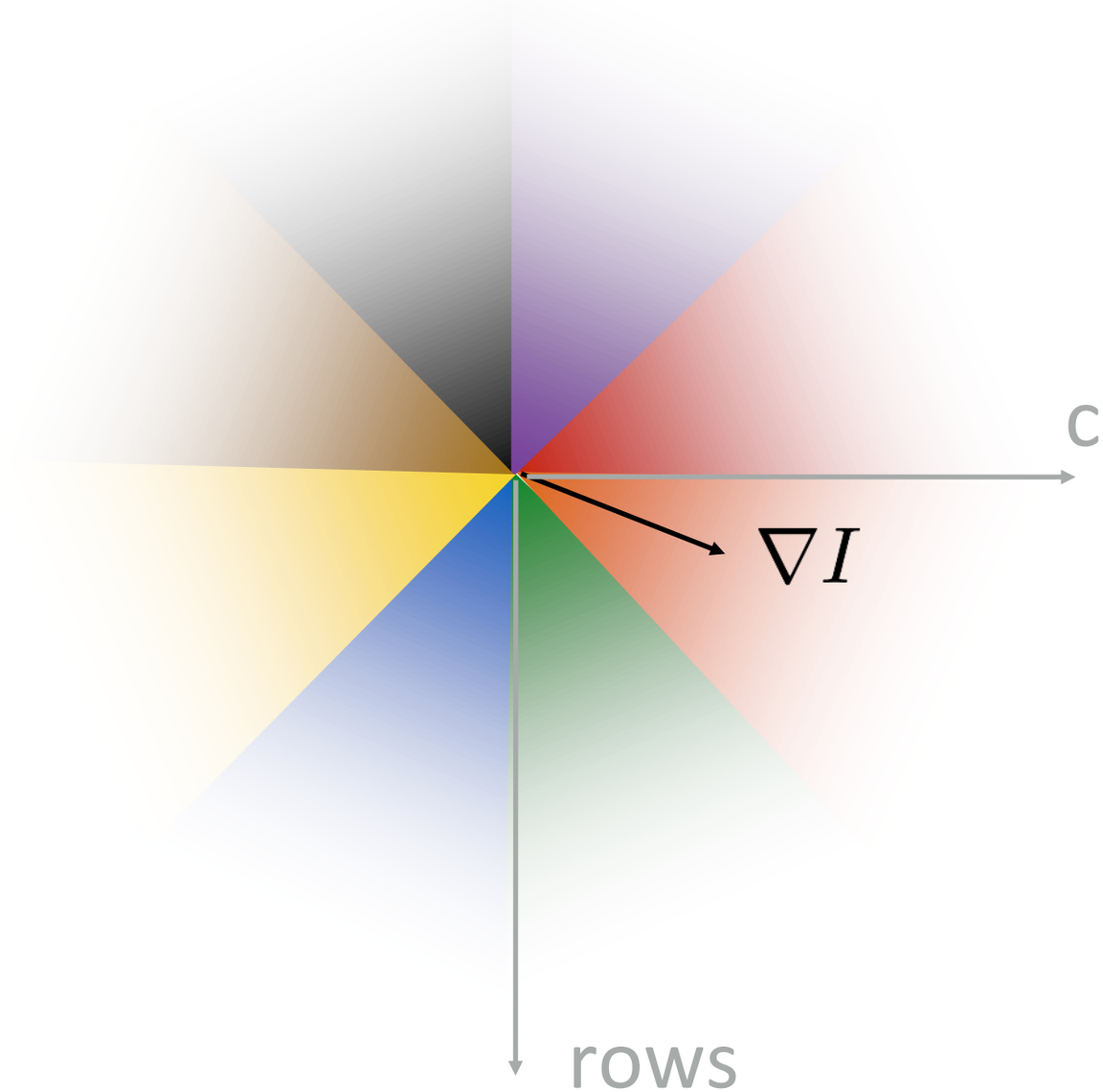


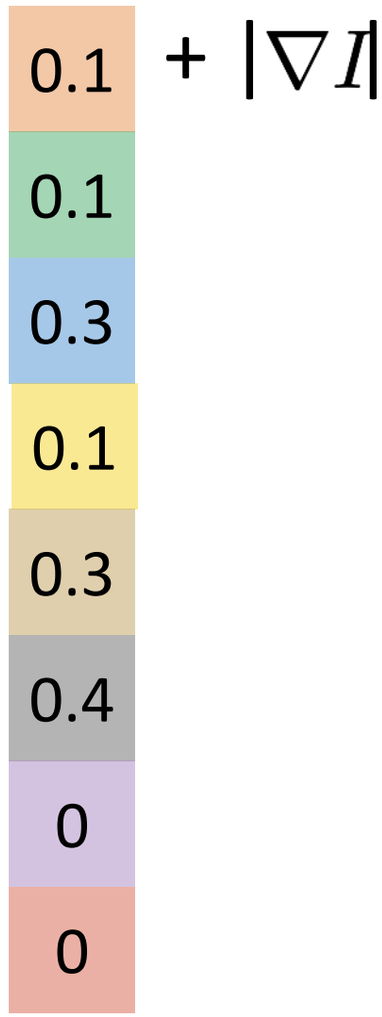
Humans seem to focus on gradients rather than absolute values.

Gradient Histogram

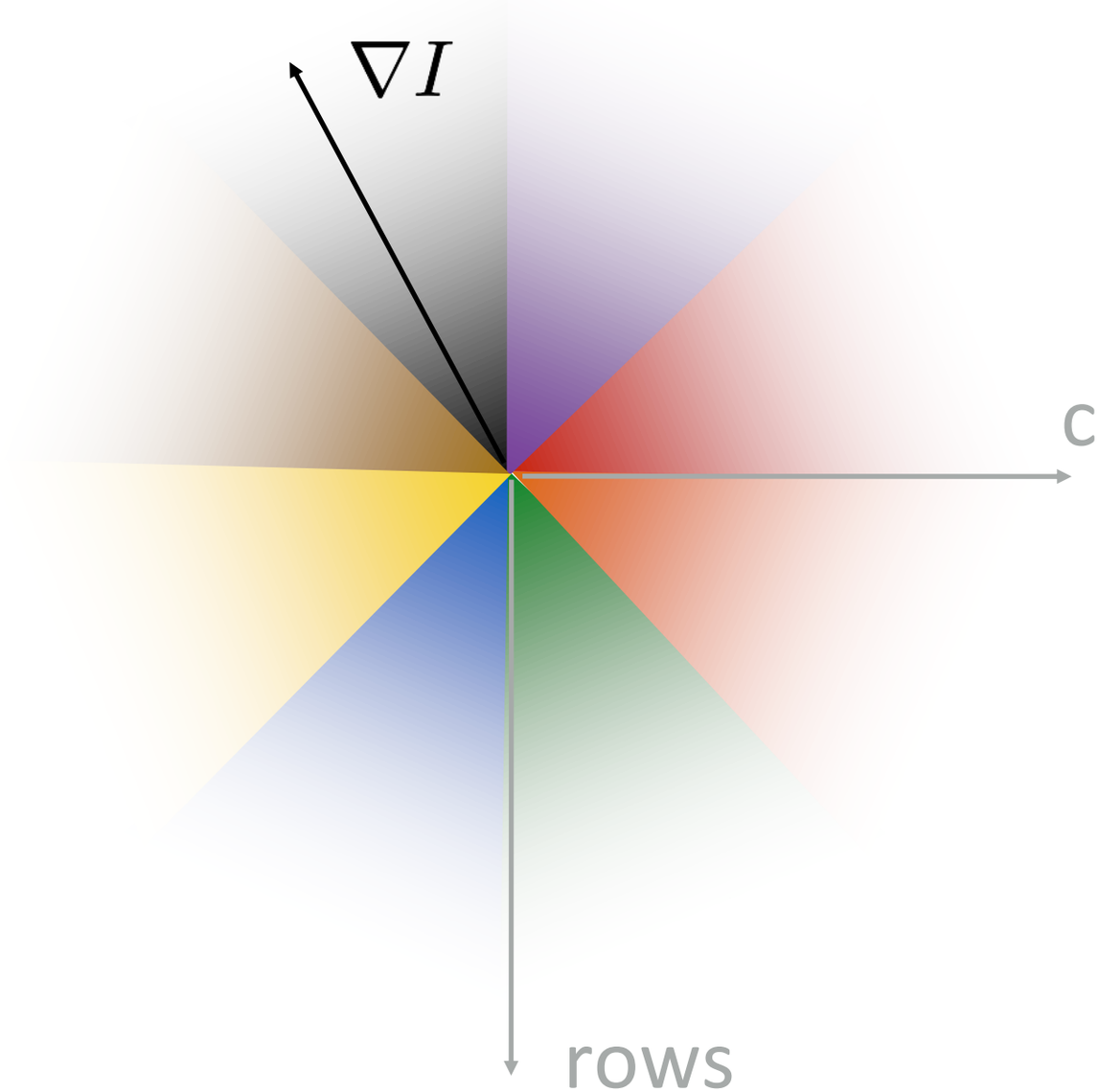


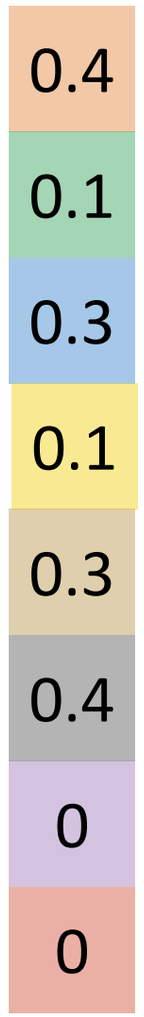
Gradient Histogram





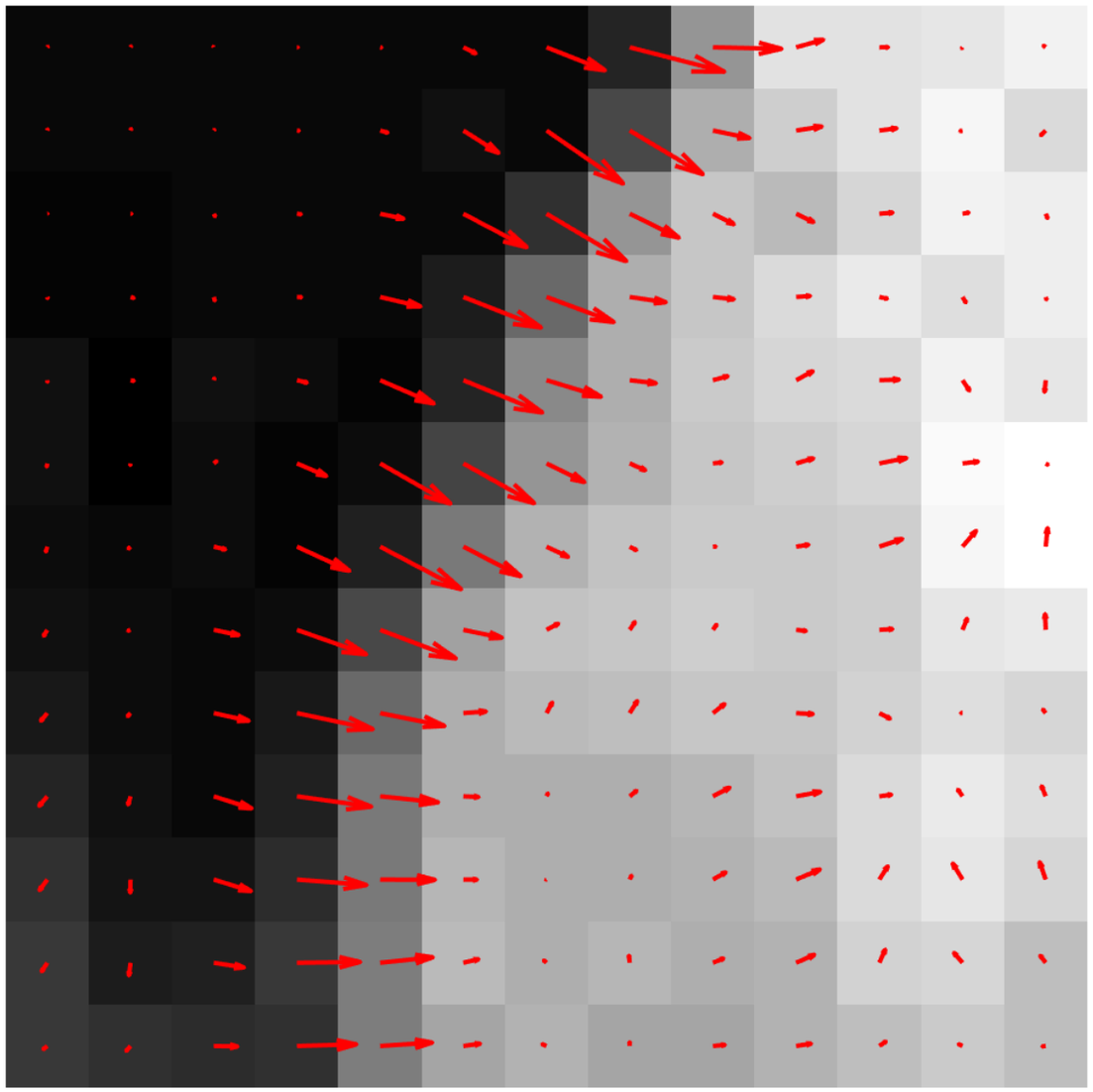
Gradient Histogram

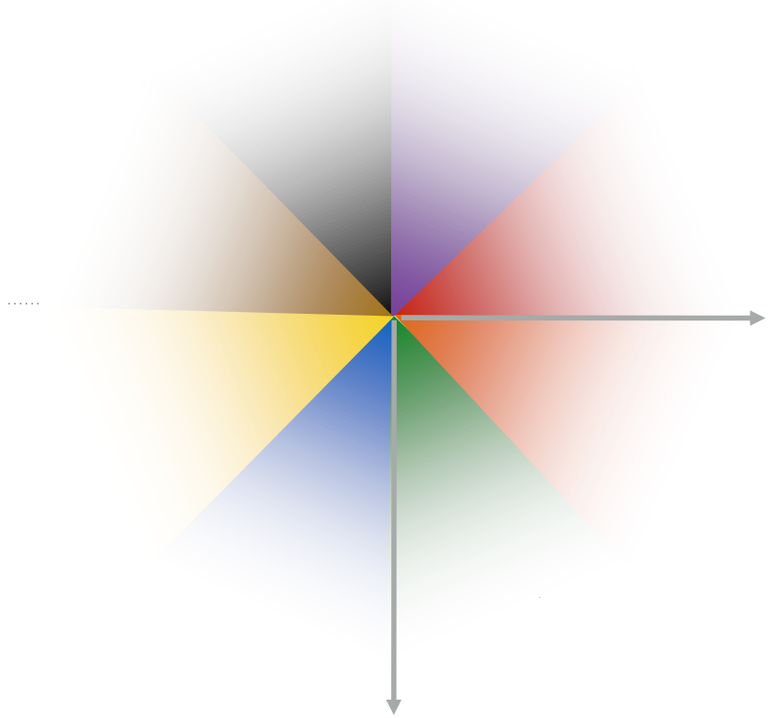




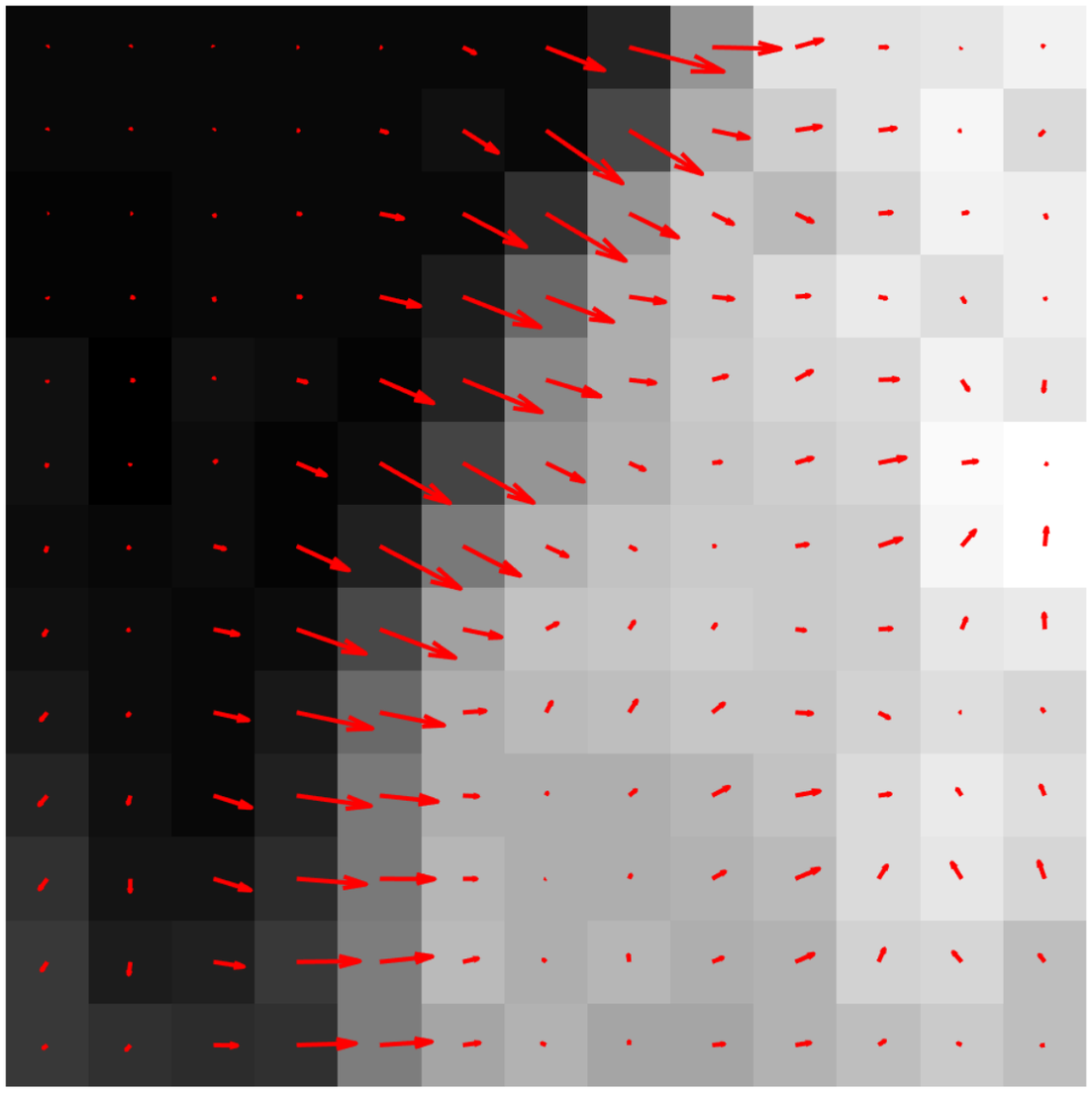
$+ |\nabla I|$

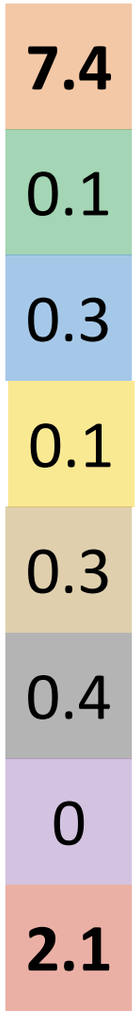
Gradient Histogram

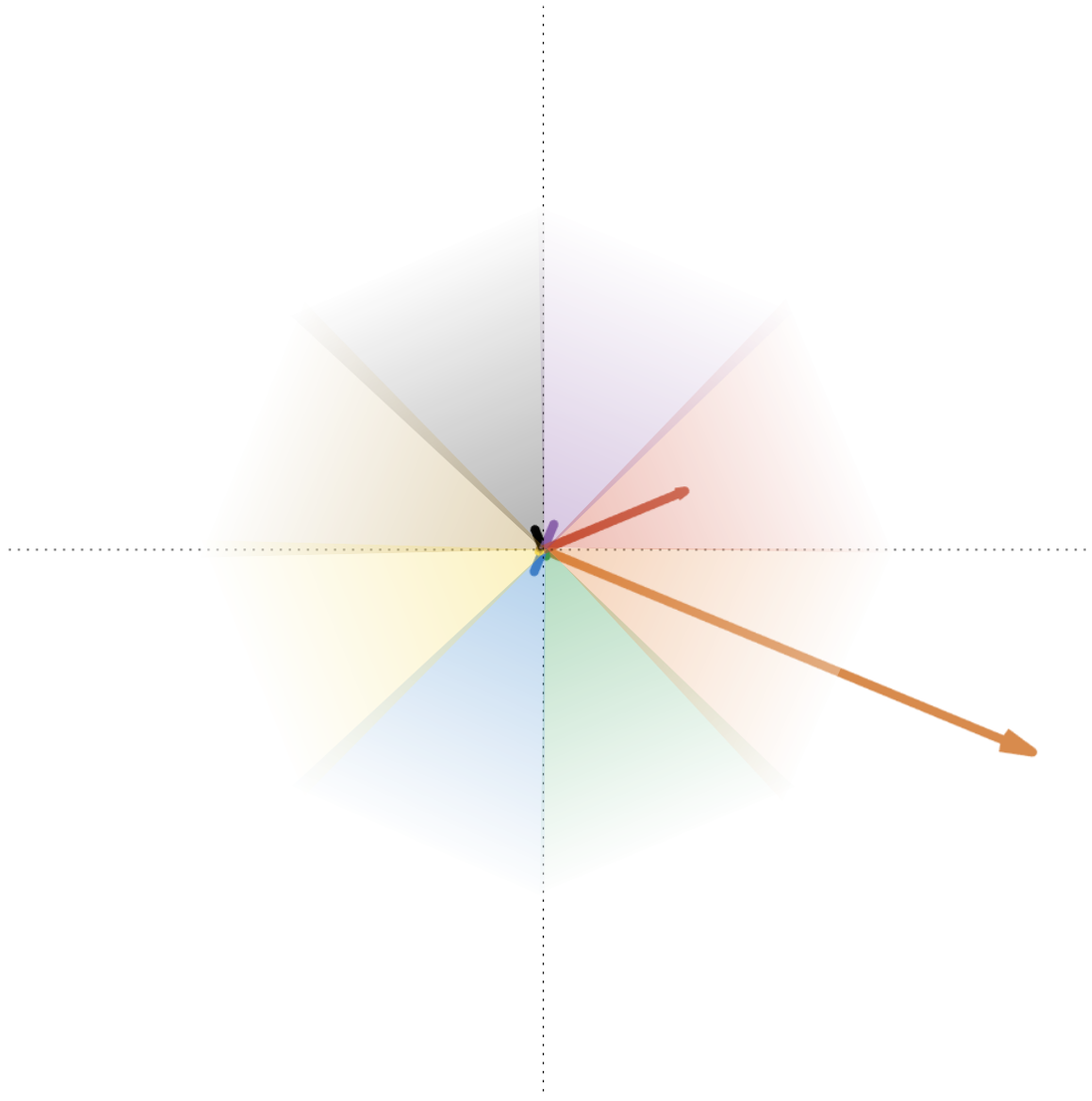




Gradient Histogram

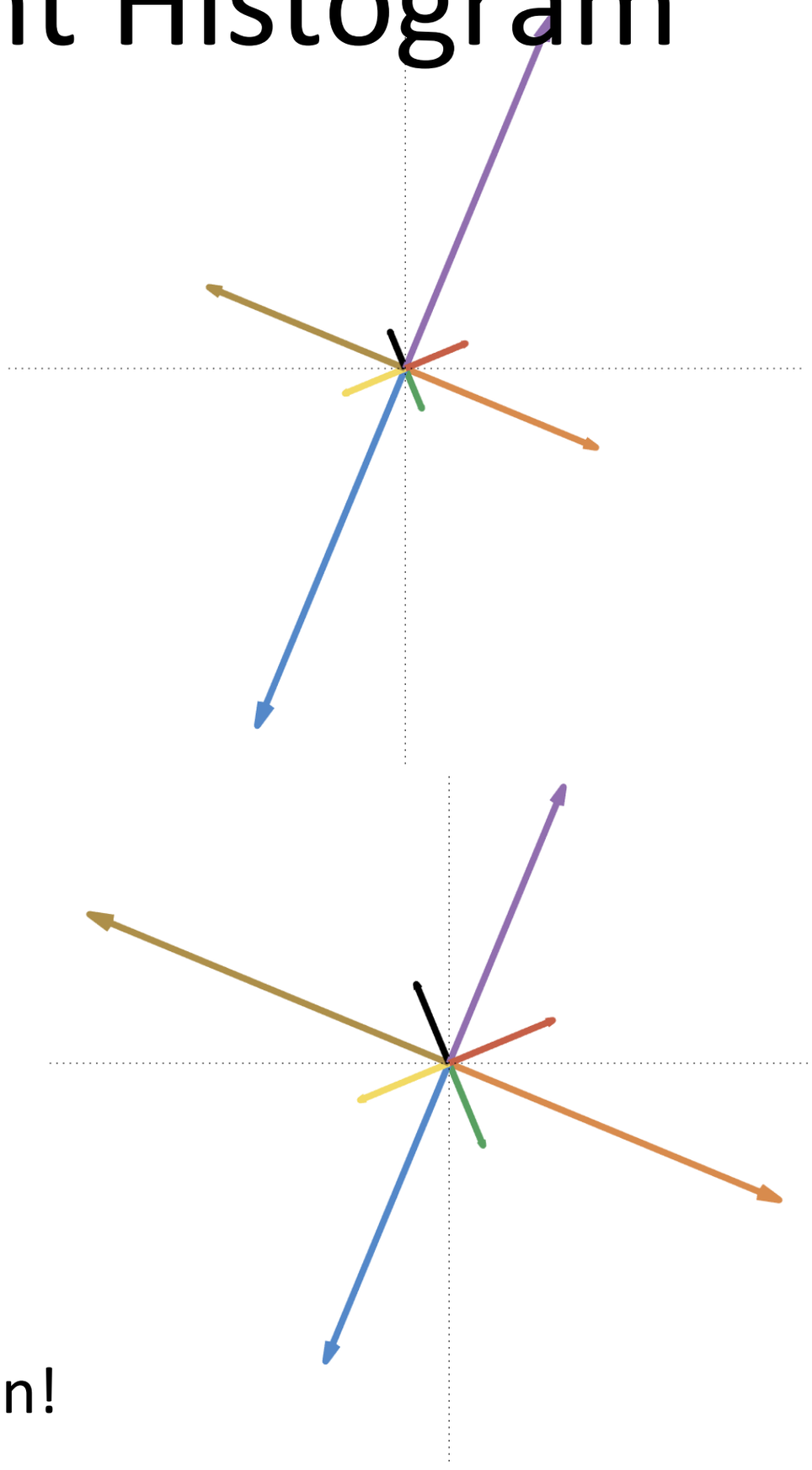


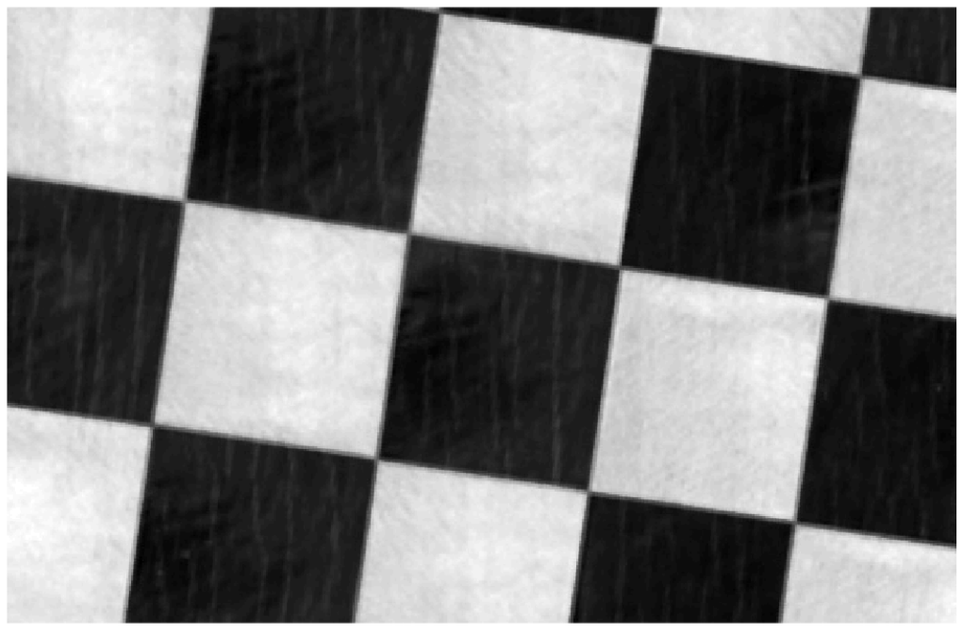




“bouquet of vectors”

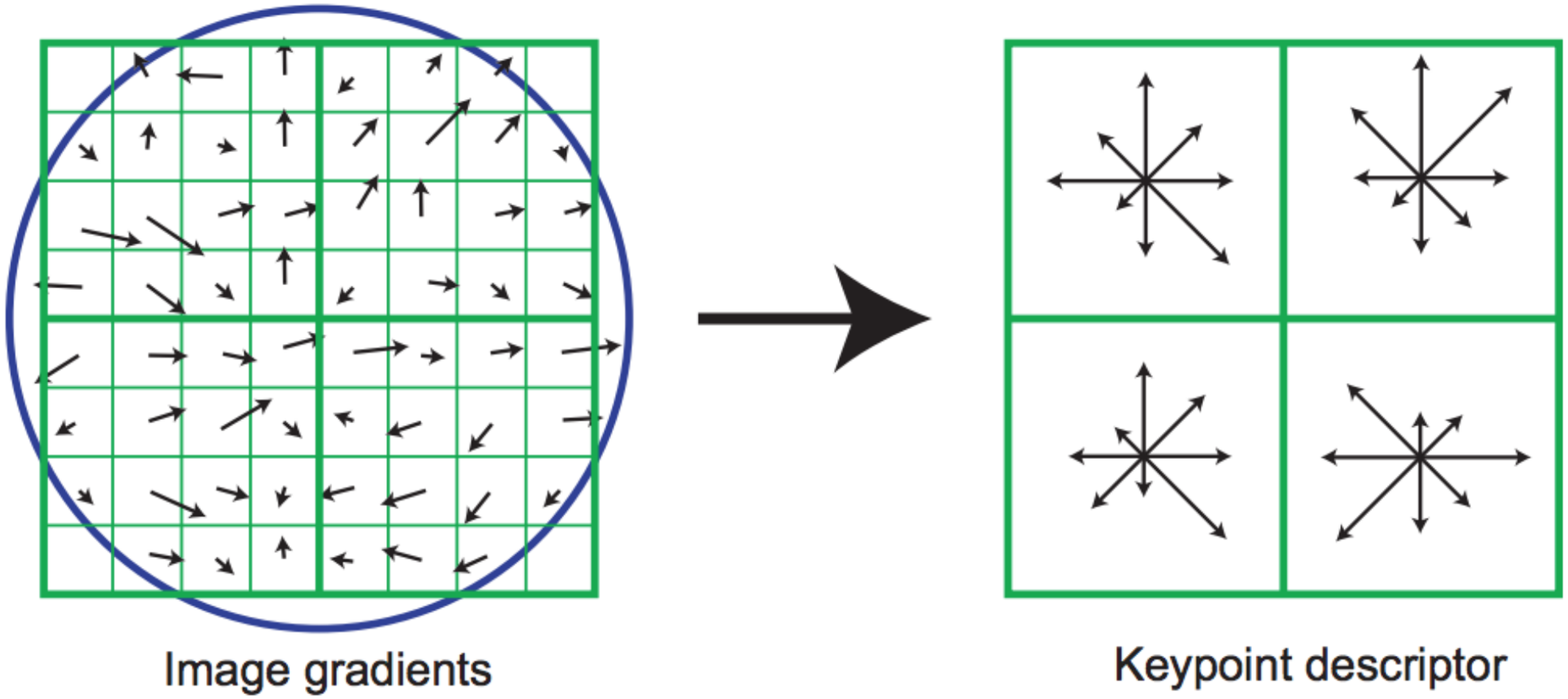
Gradient Histogram





Much more similar than pixel-wise comparison!

SIFT / HOG

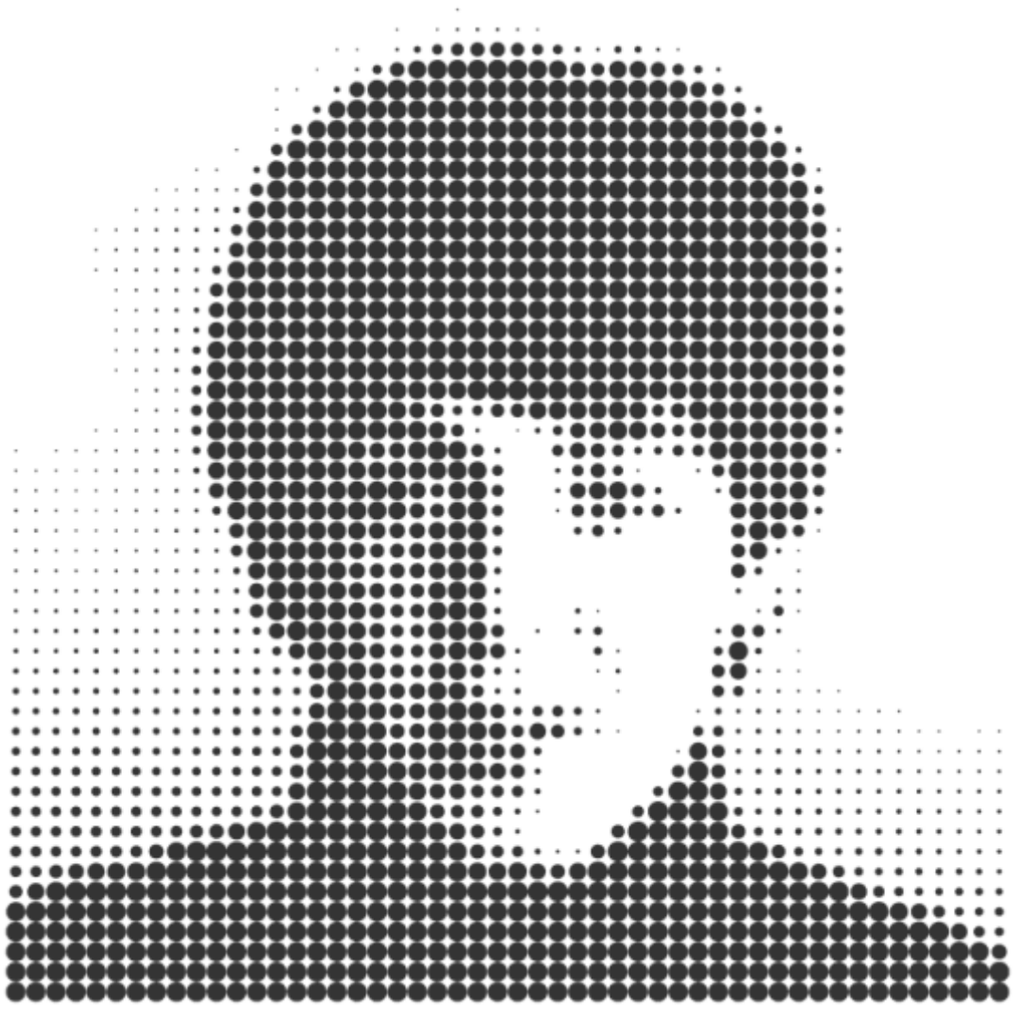


Known as **SIFT** or Histogram of Oriented Gradients (**HOG**)
More details in next lecture

slide credit: David G. Lowe

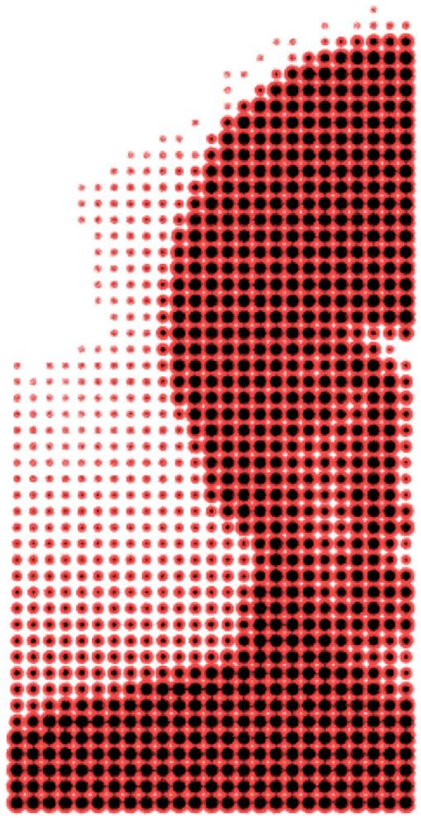
Multi-Scale Processing

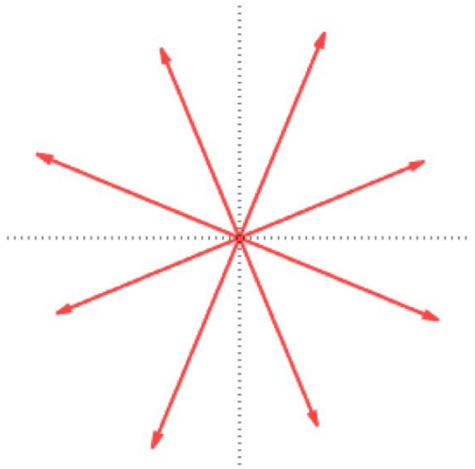
Scale

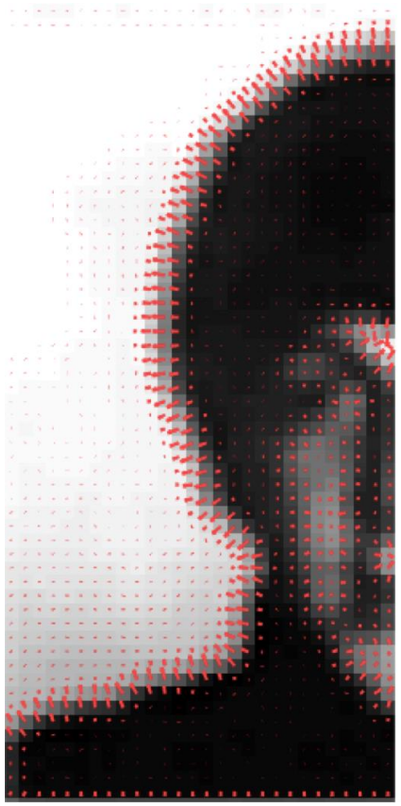


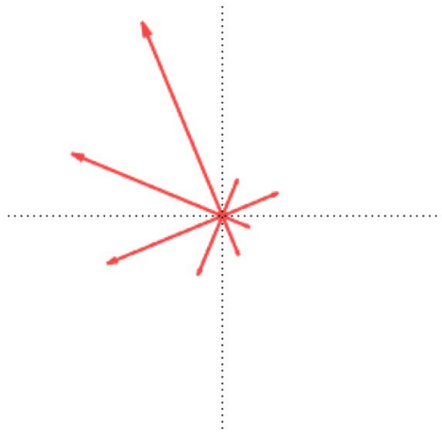
How to recognize objects at different distances?

Gradient histogram

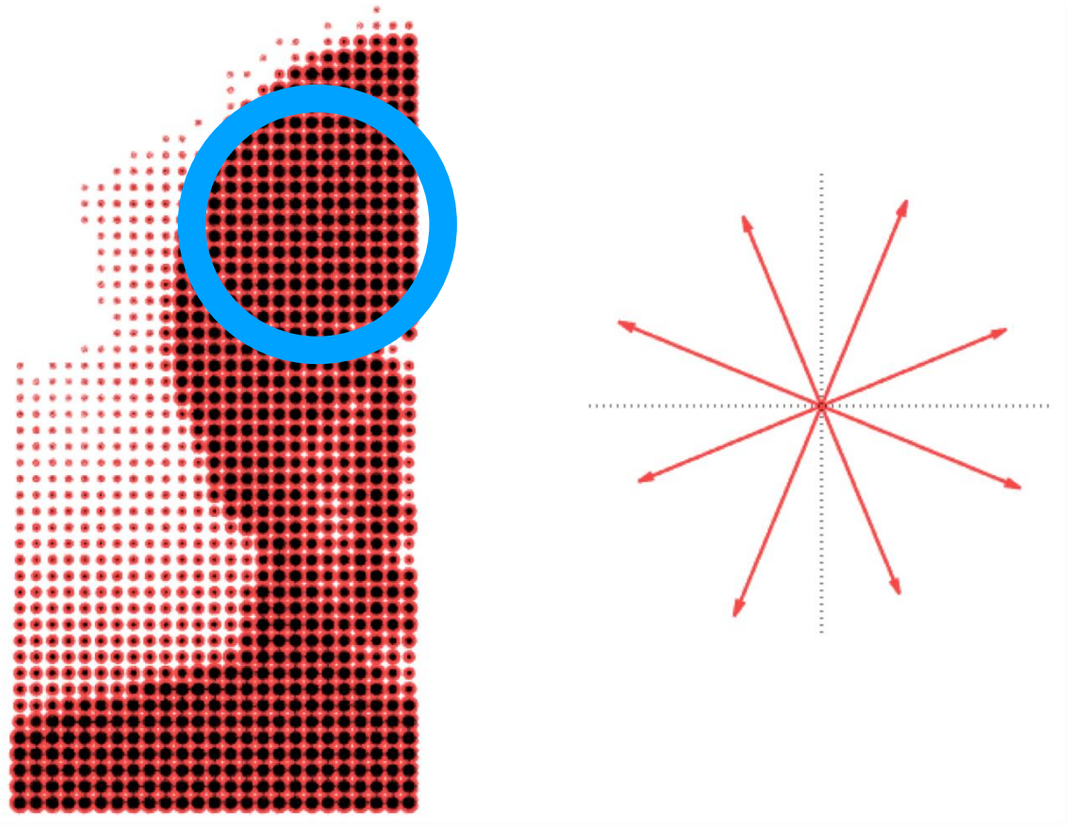


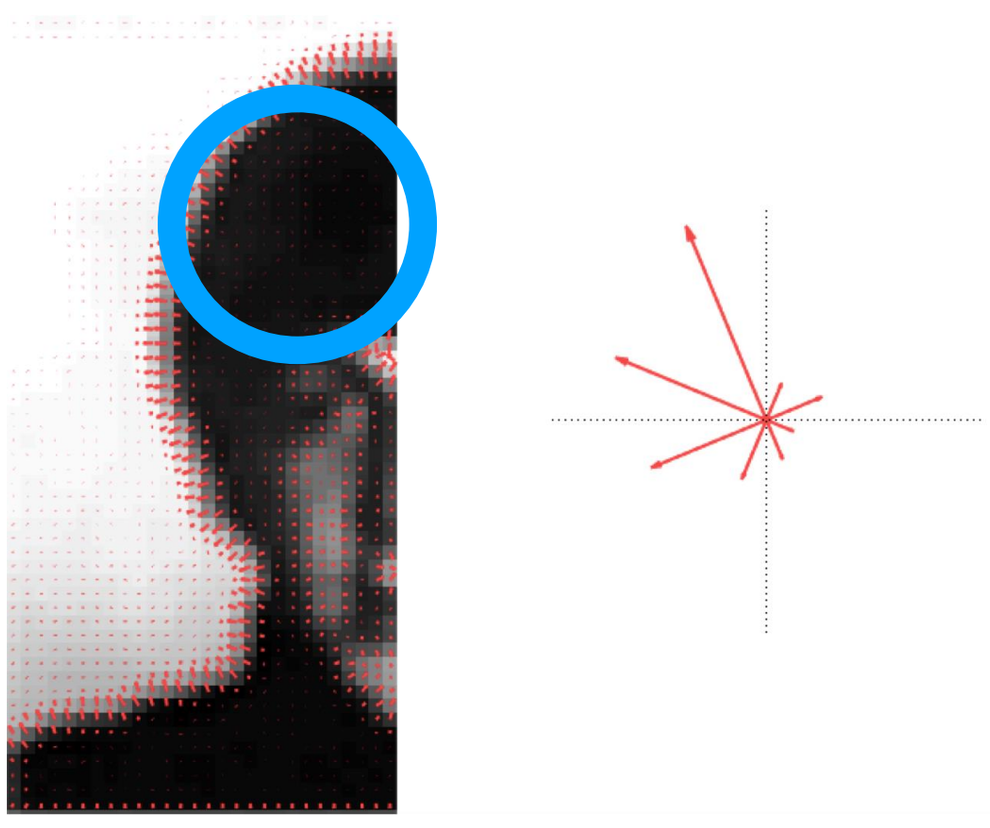


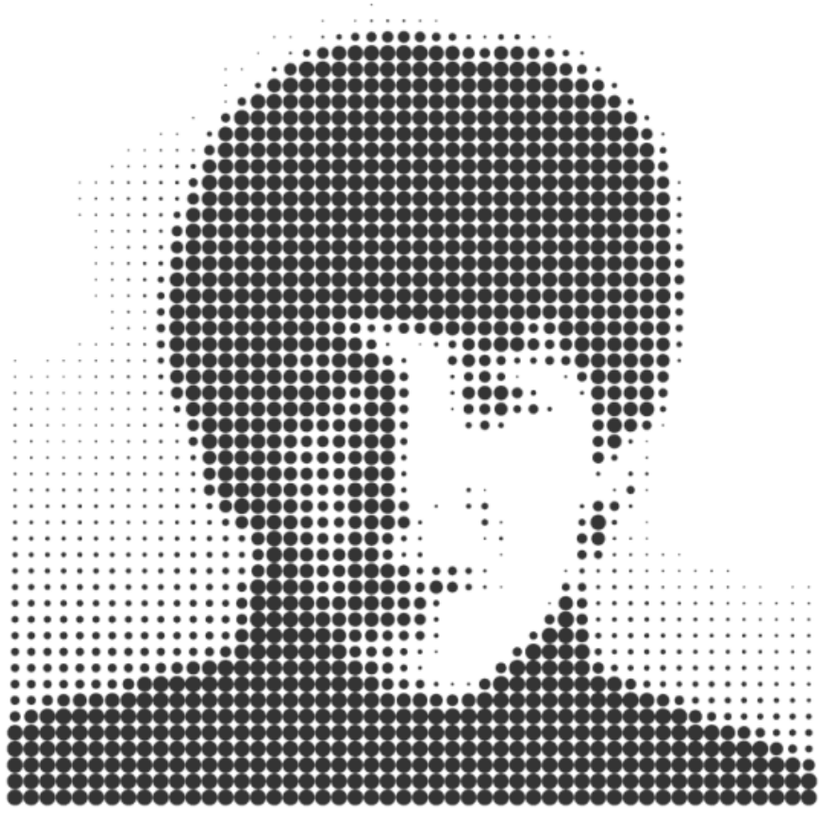


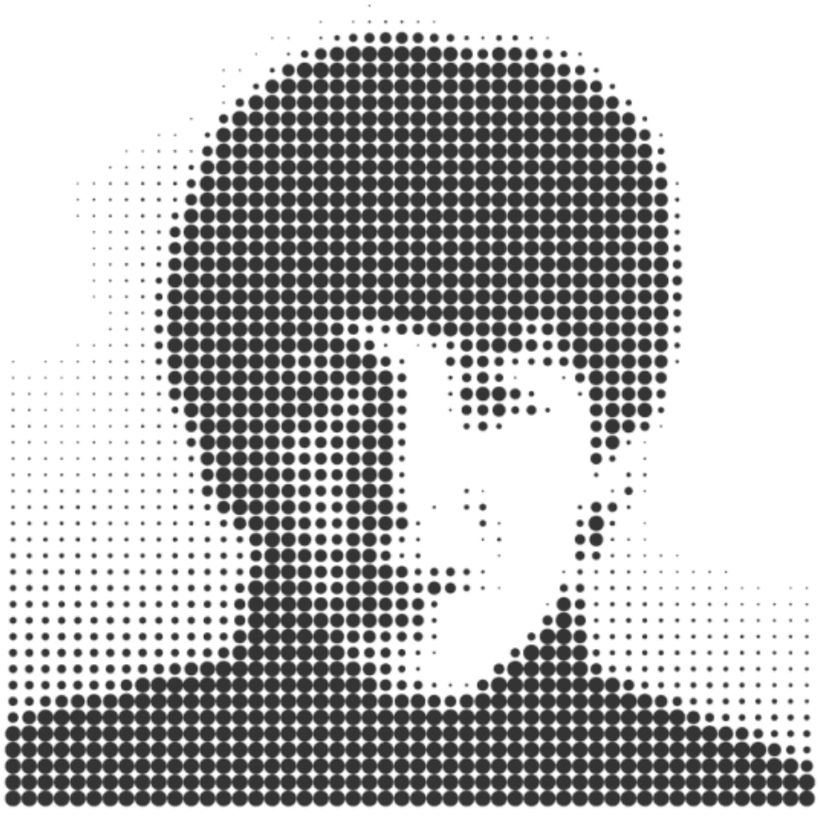


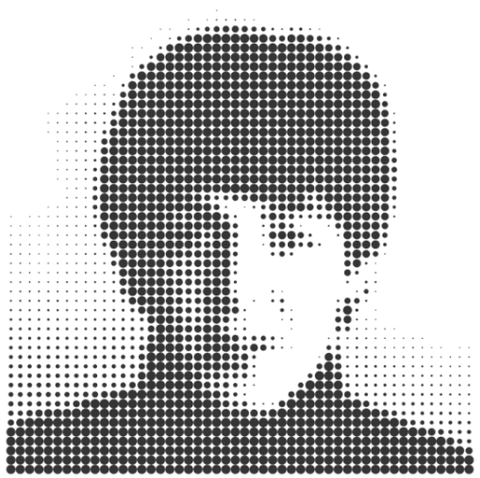
Gradient histogram

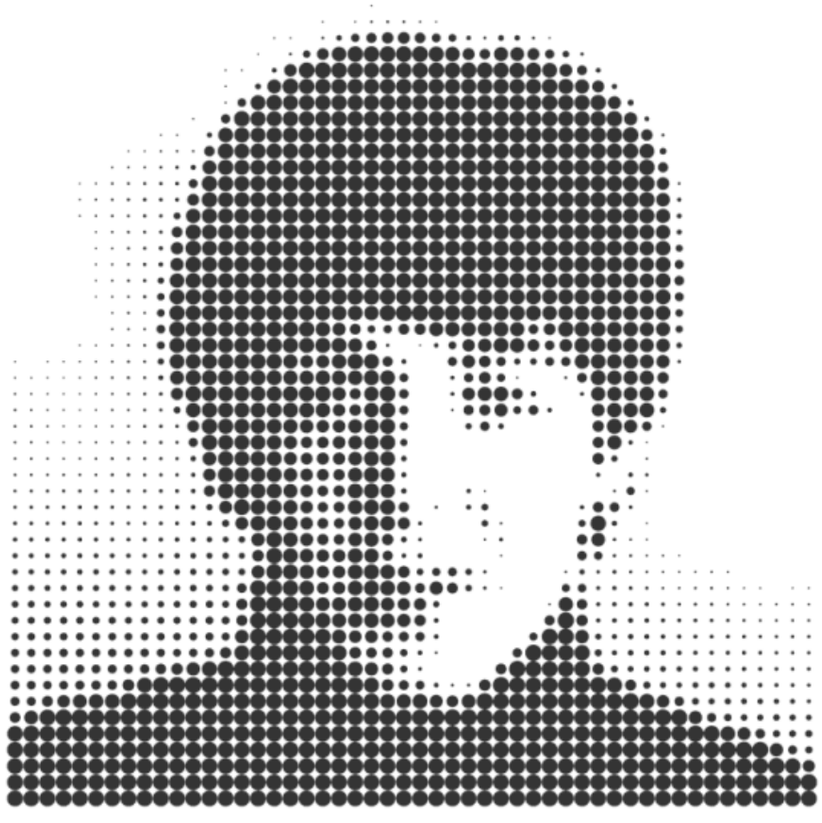


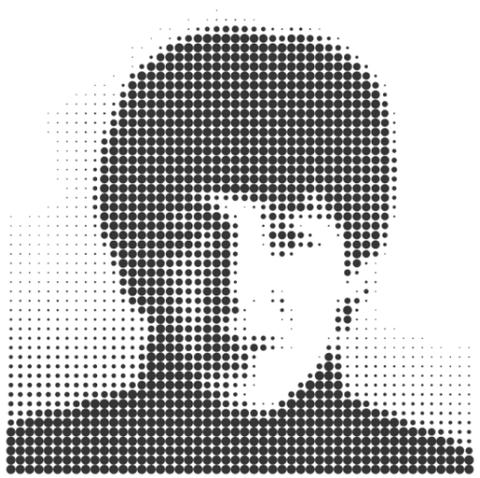


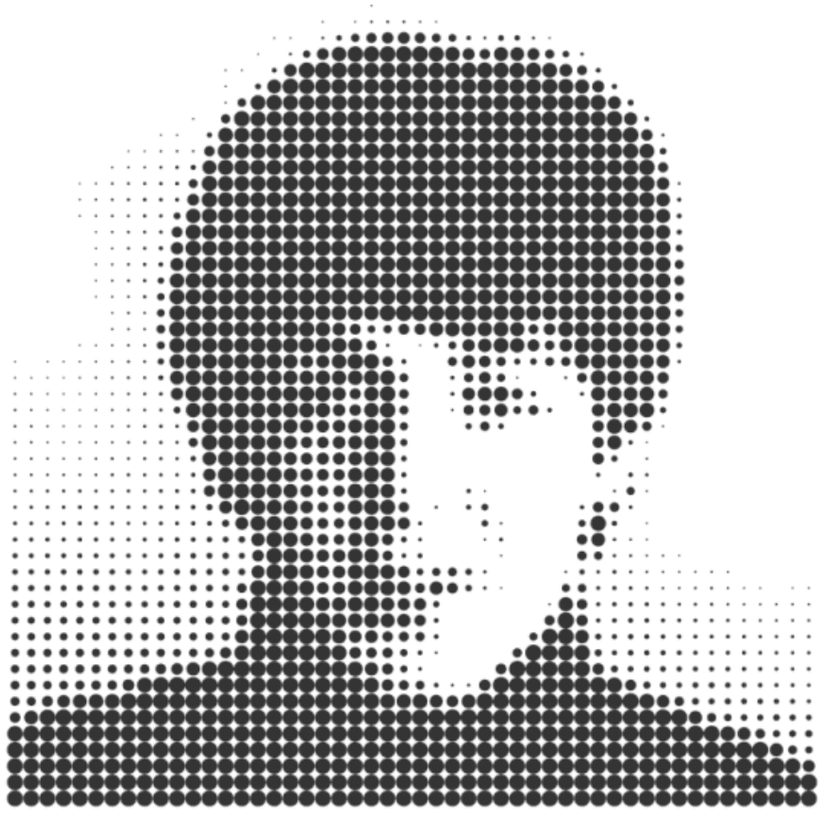


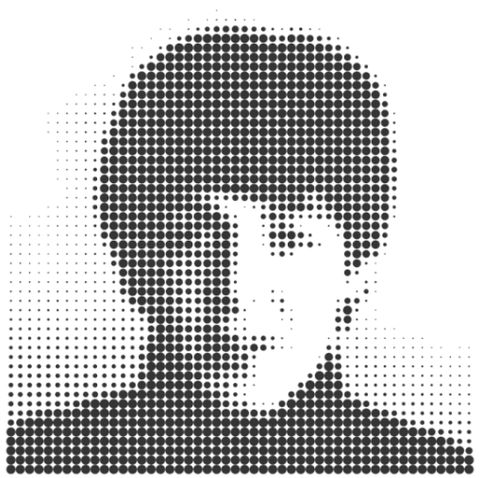


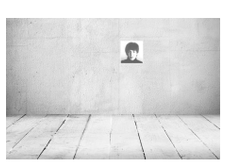


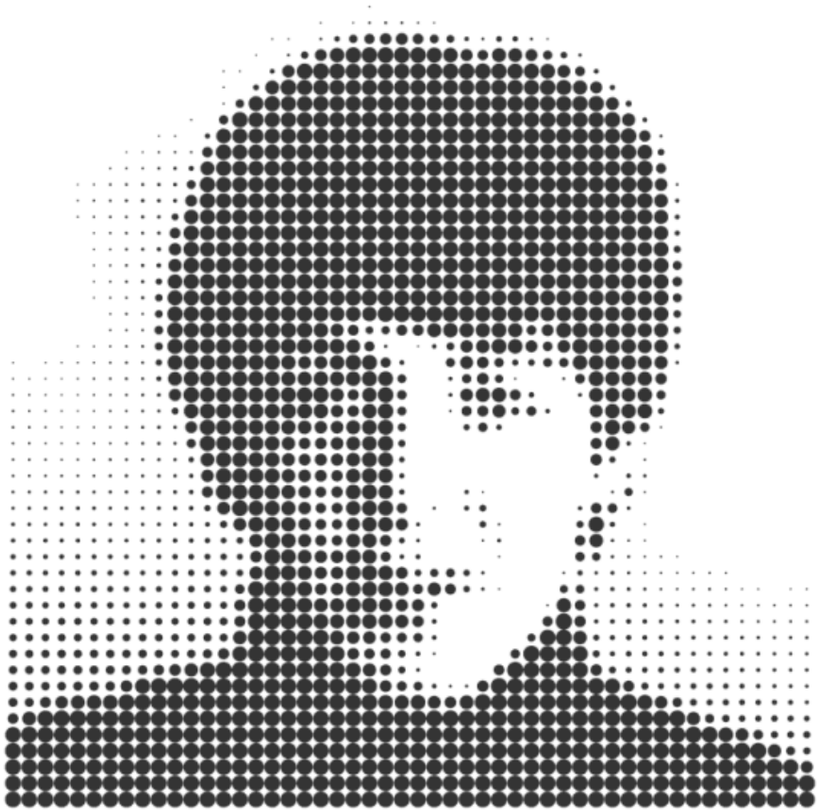


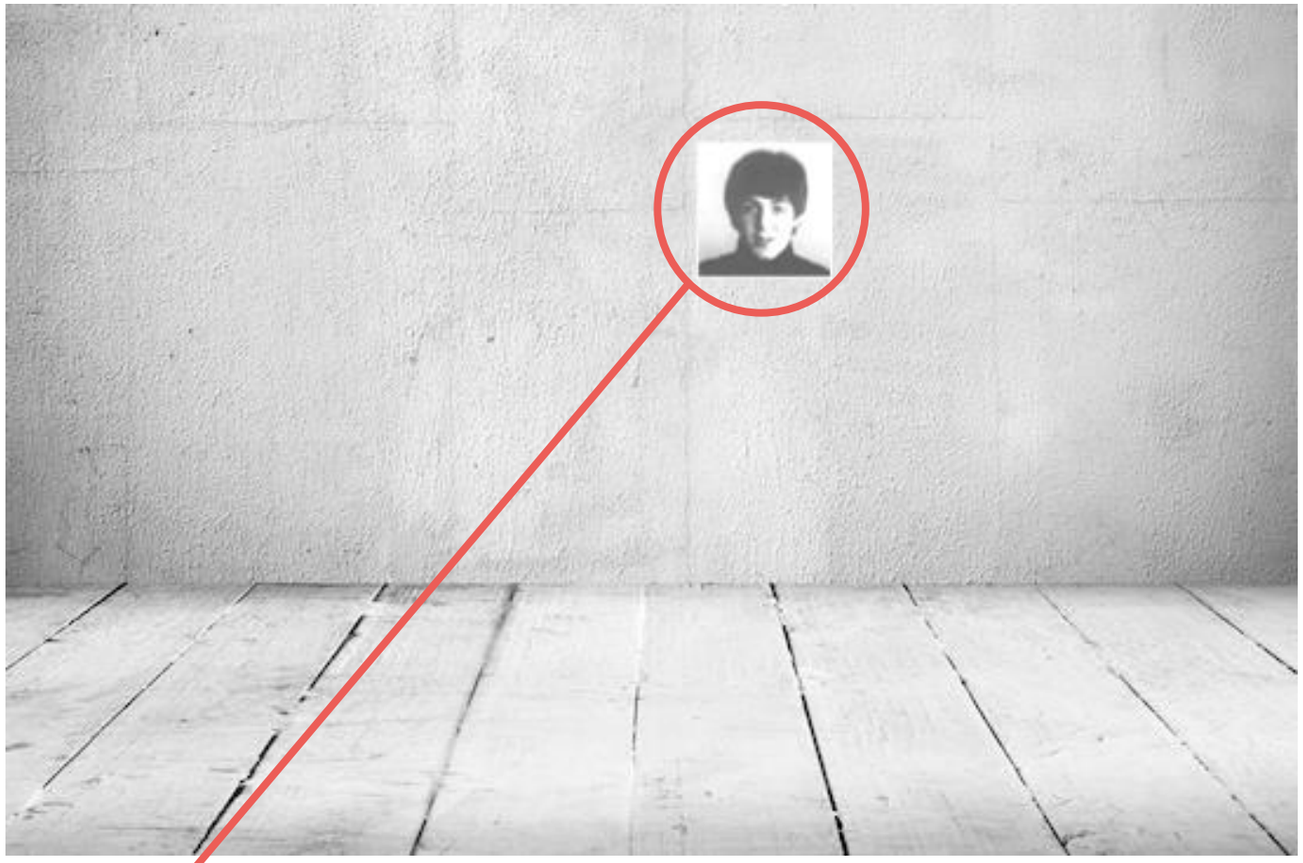


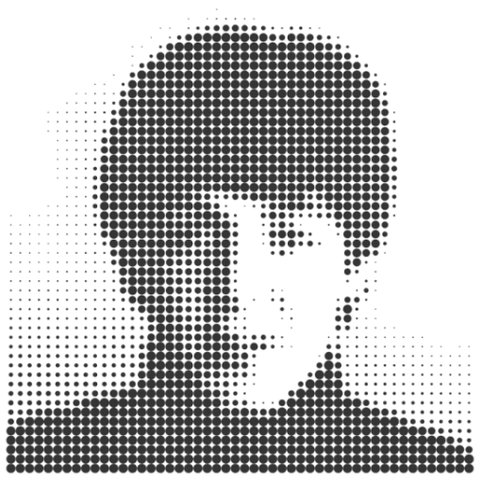


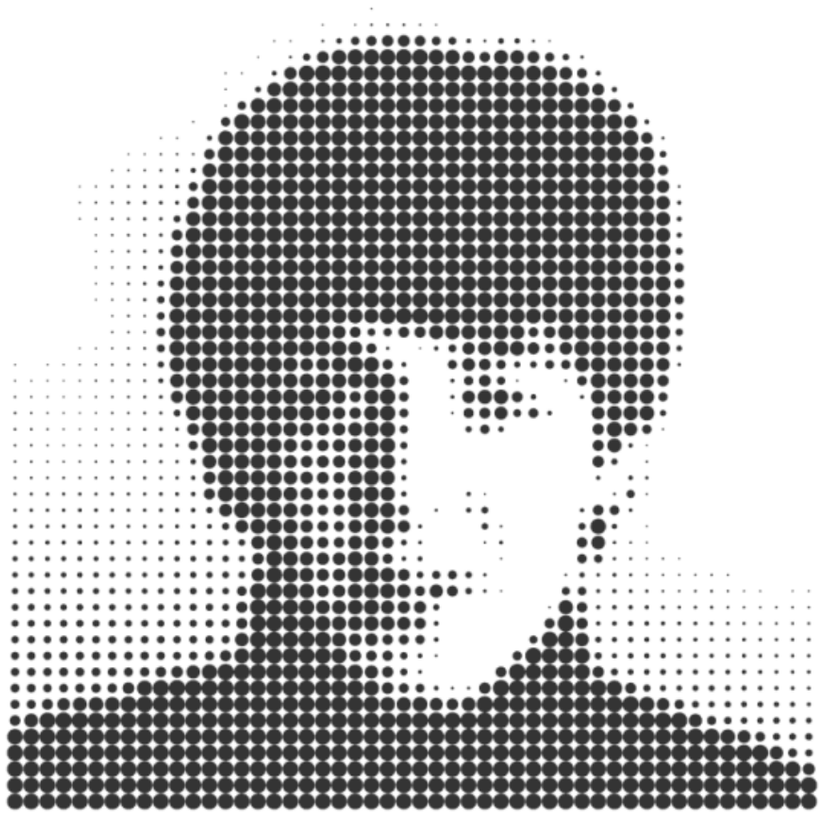


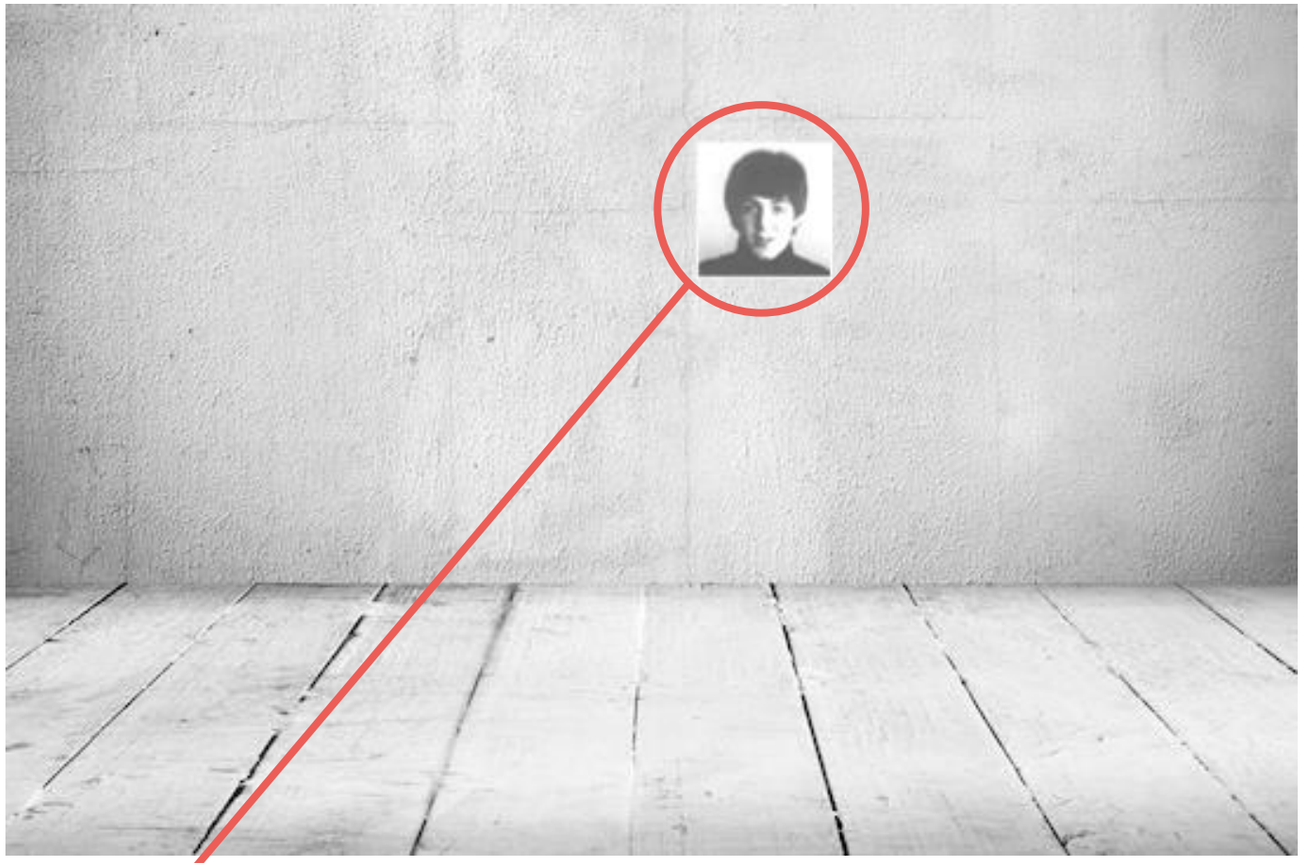


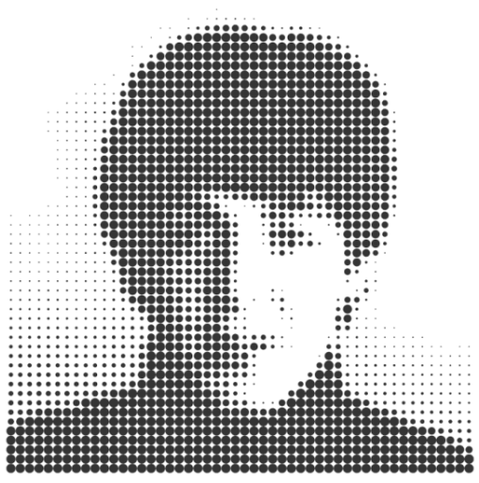


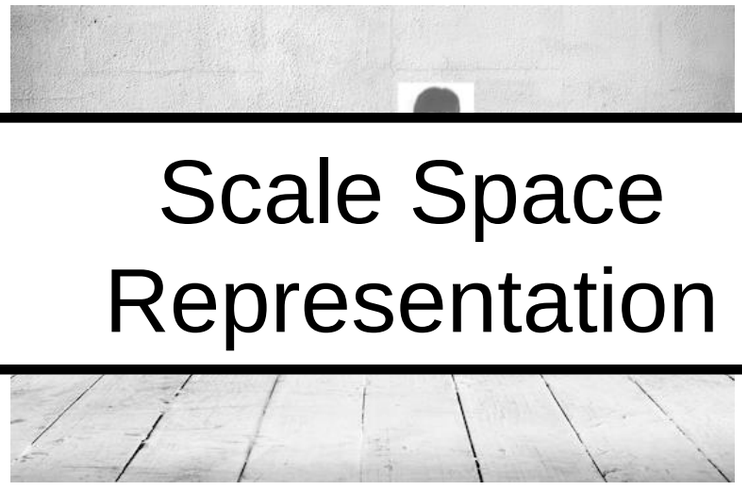








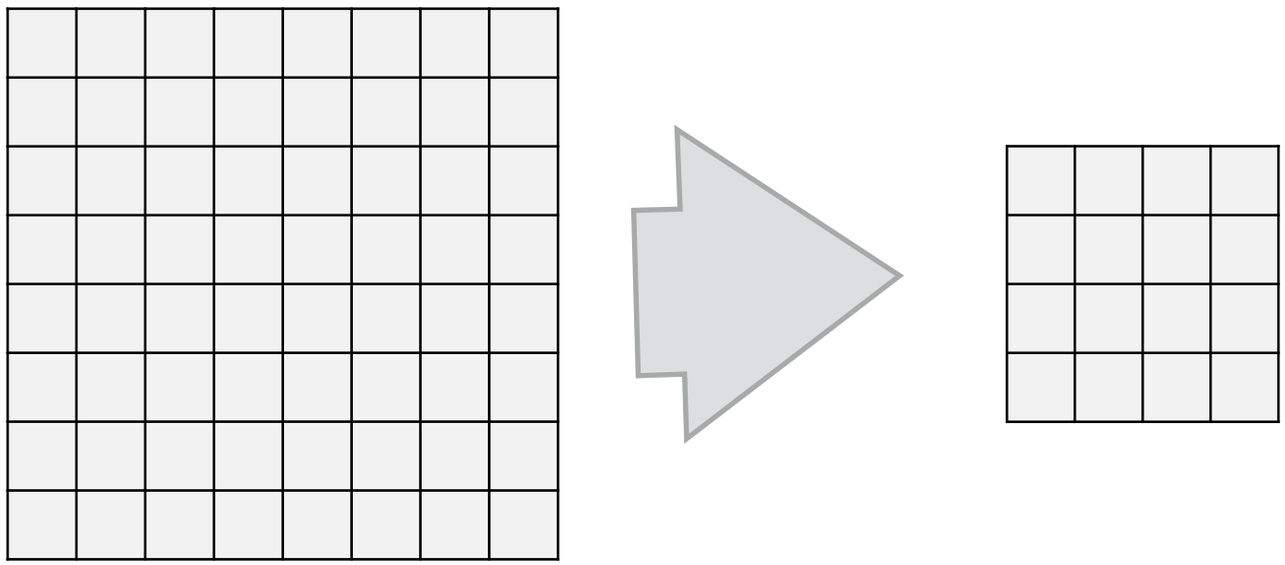




Scale Space Representation

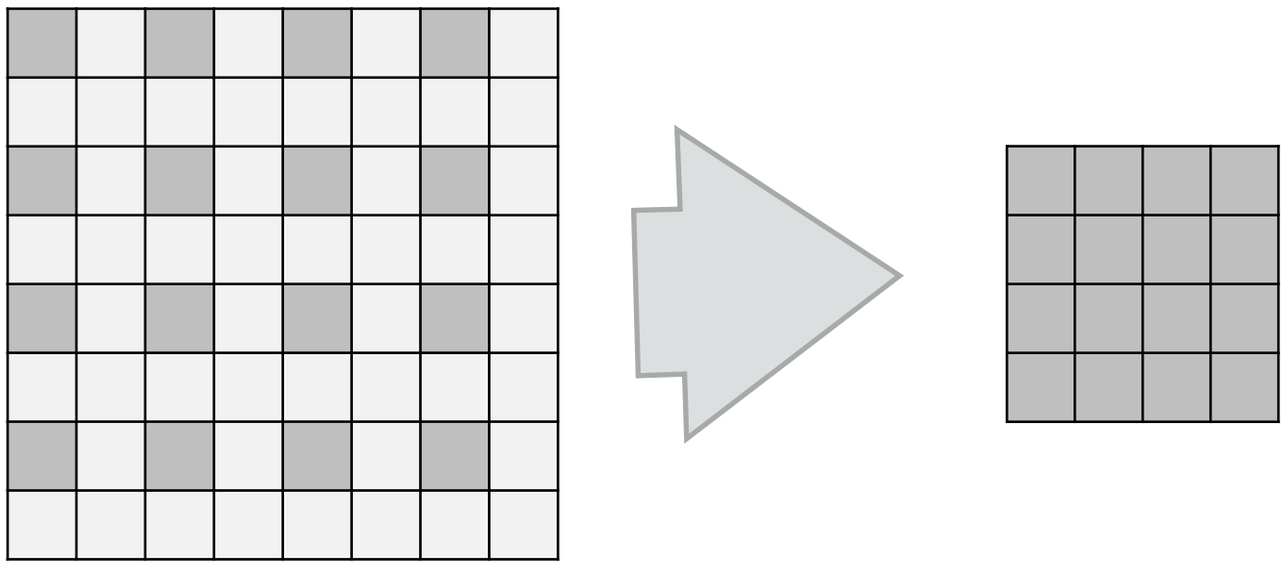
Building a Scale Space Representation

Downsampling:



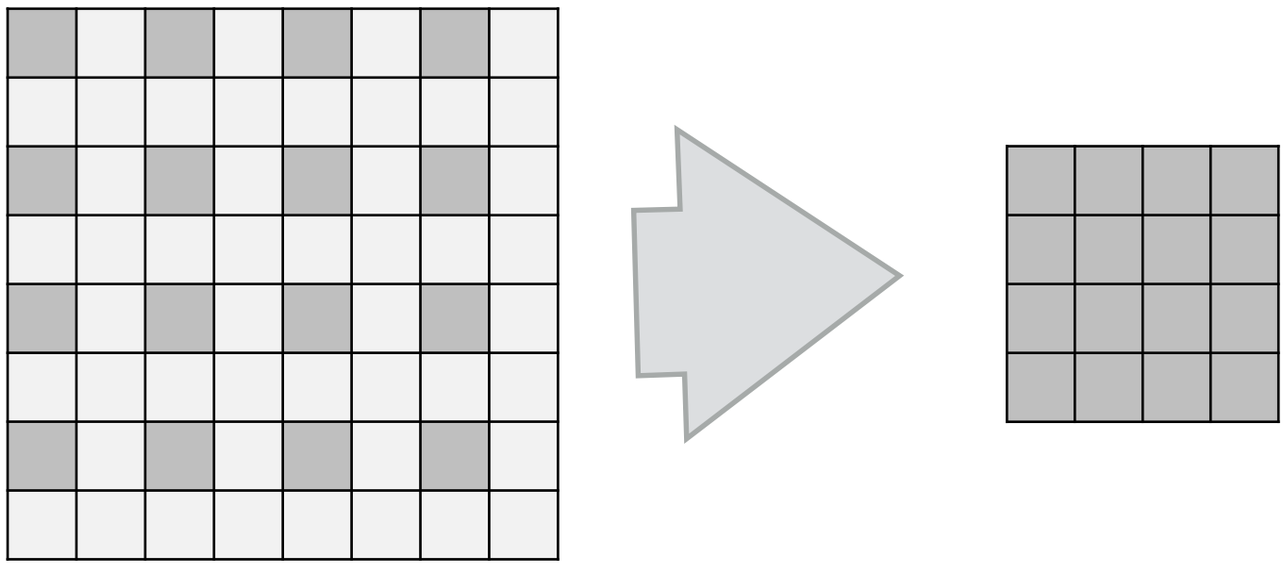
Building a Scale Space Representation

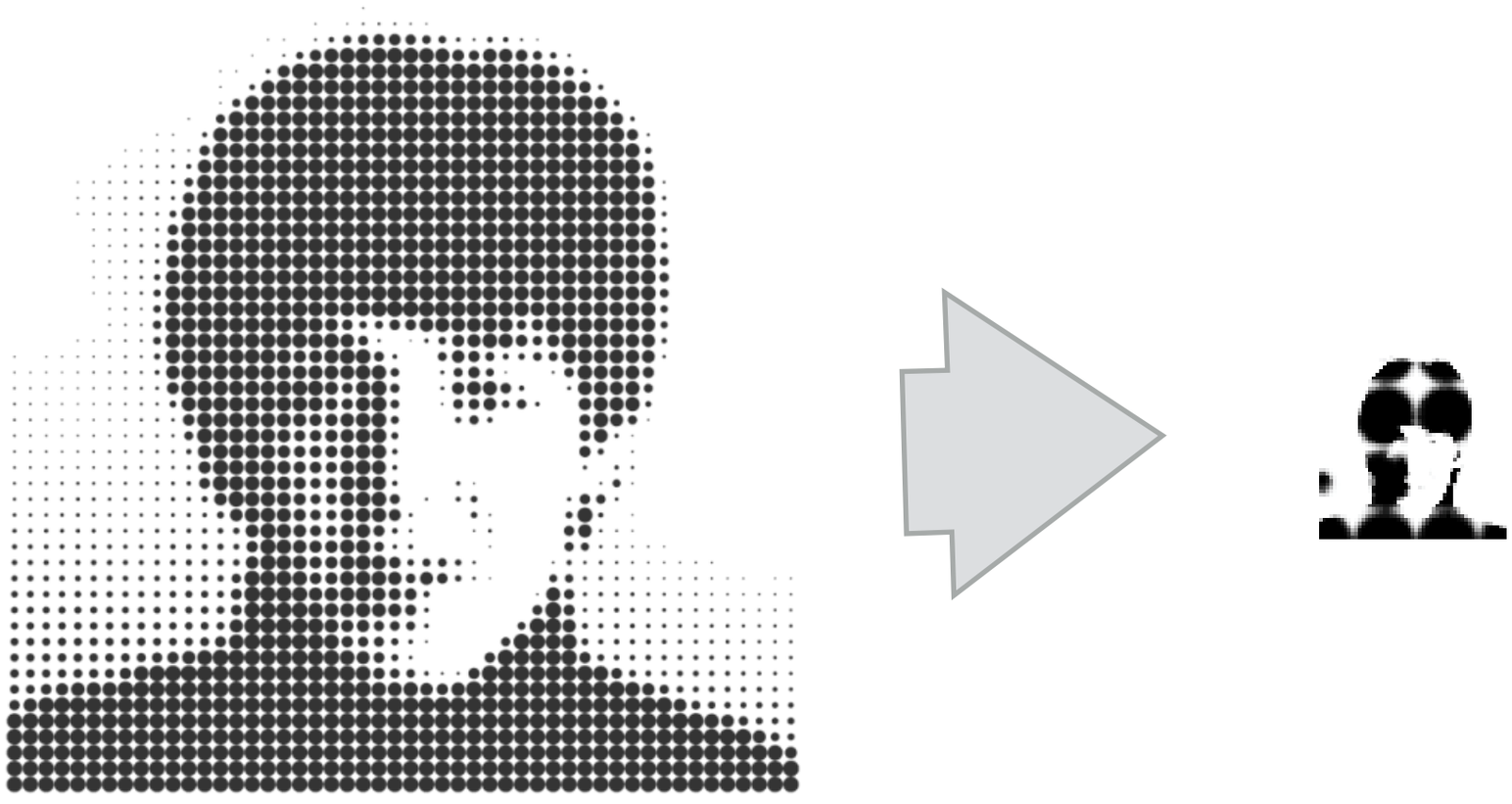
Downsampling:



Building a Scale Space Representation

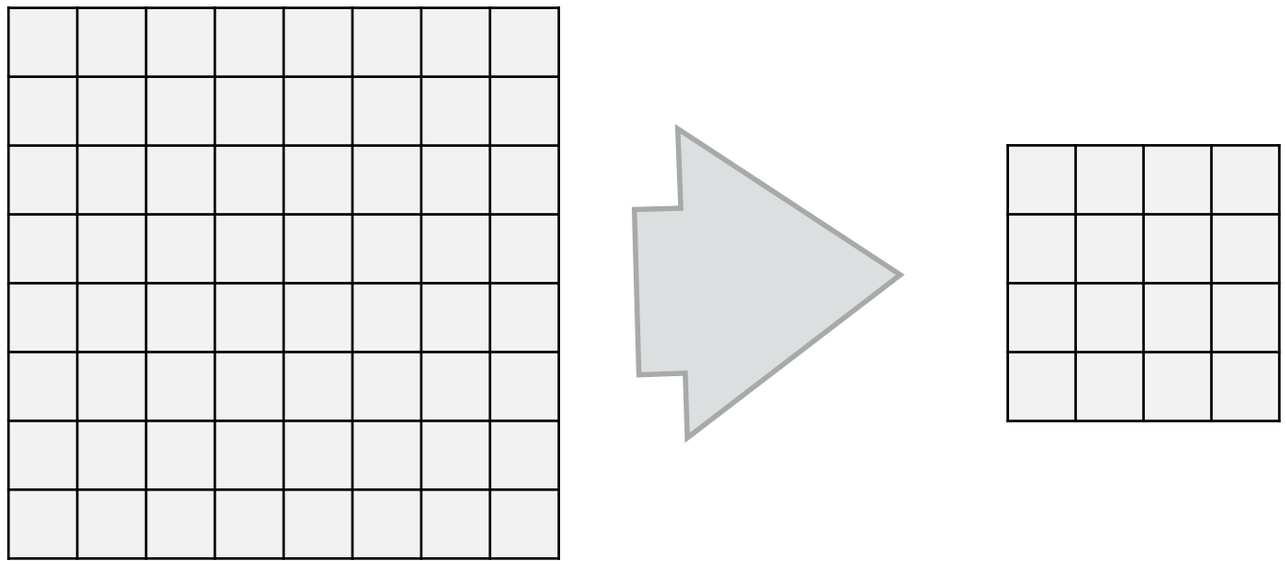
Downsampling:





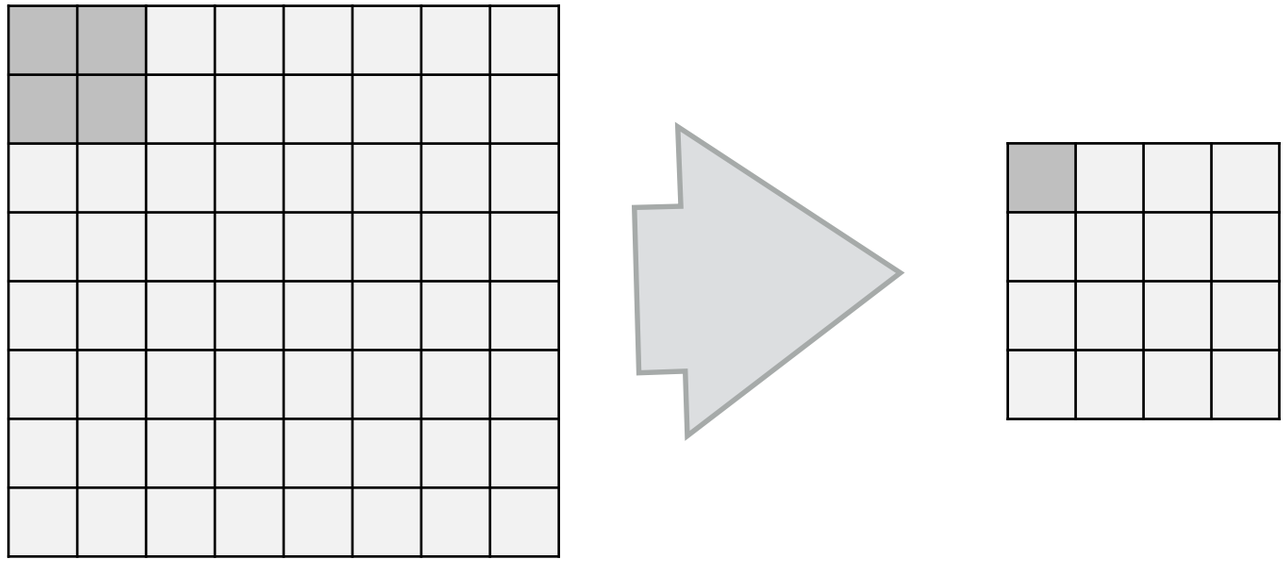
Building a Scale Space Representation

Better: Multiple pixels in larger image contribute to pixel in smaller image

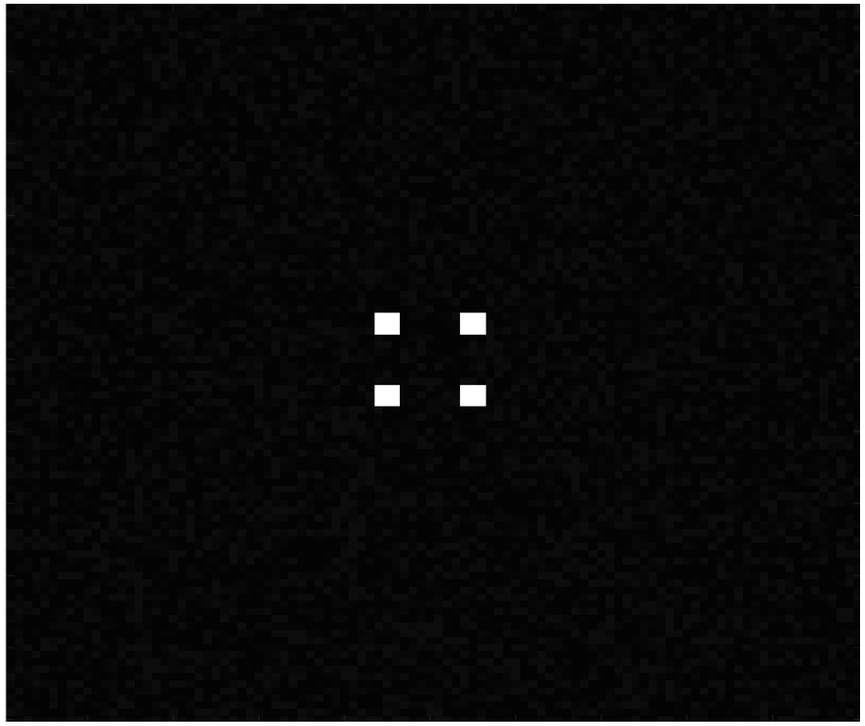


Building a Scale Space Representation

Better: Multiple pixels in larger image contribute to pixel in smaller image



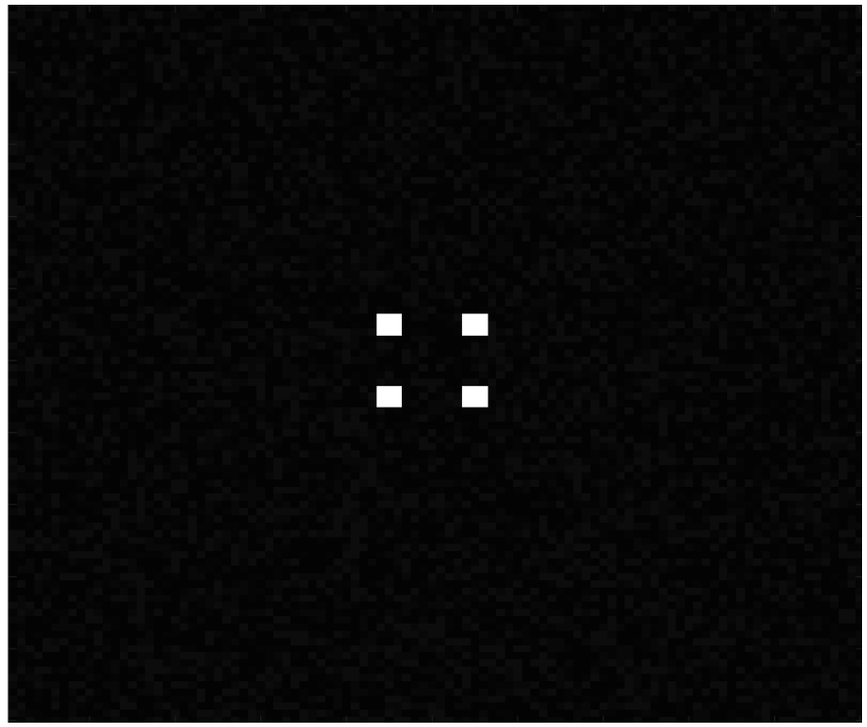
Averaging



$$* \quad \frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

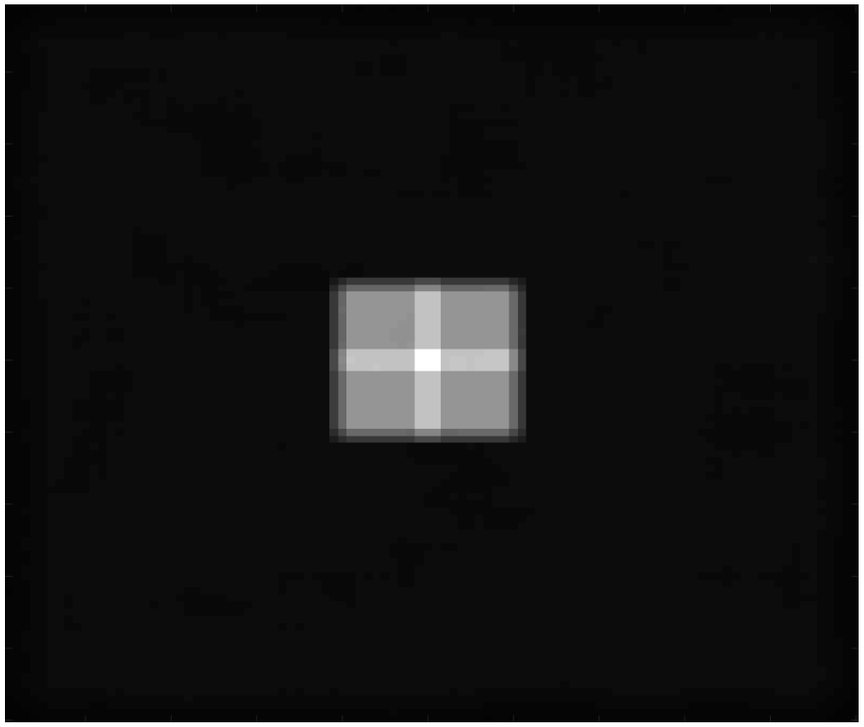
Averaging



* $\frac{1}{25}$

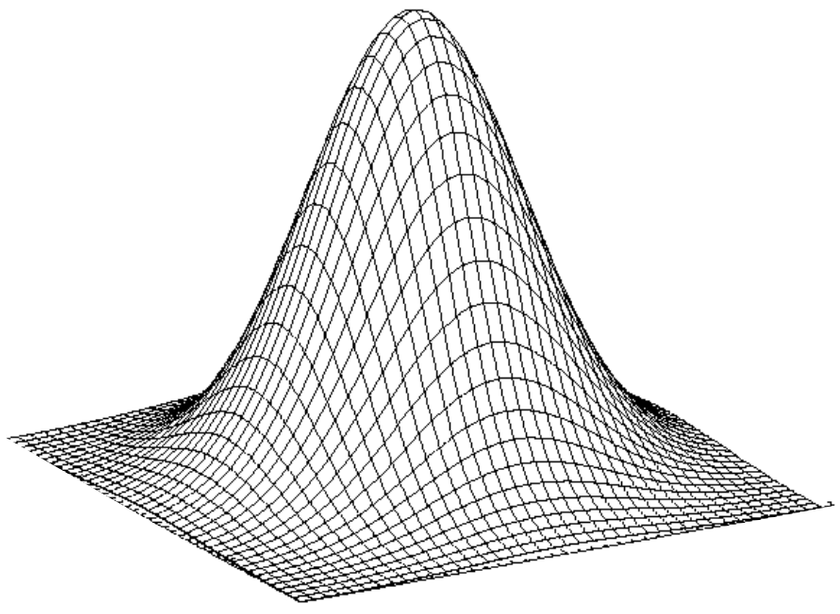
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

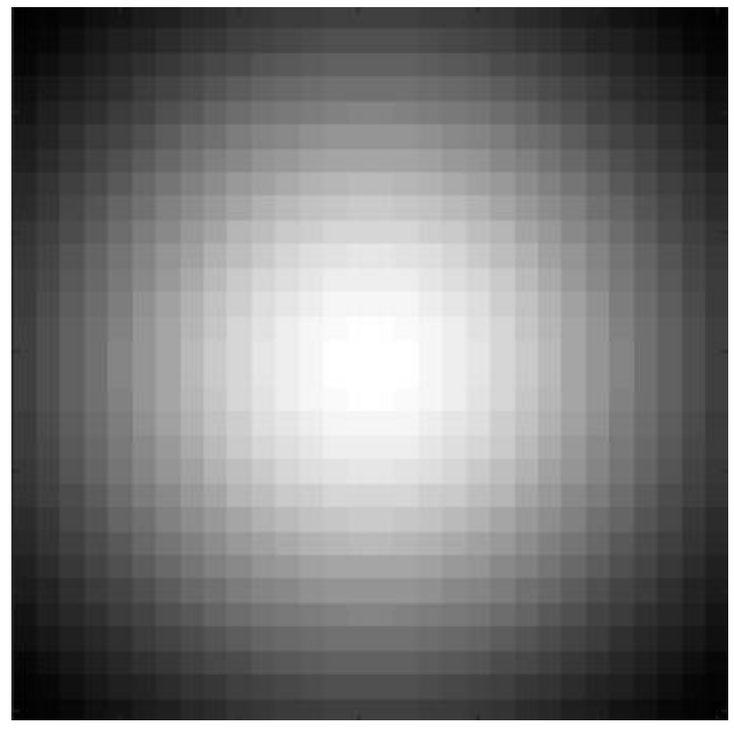
=



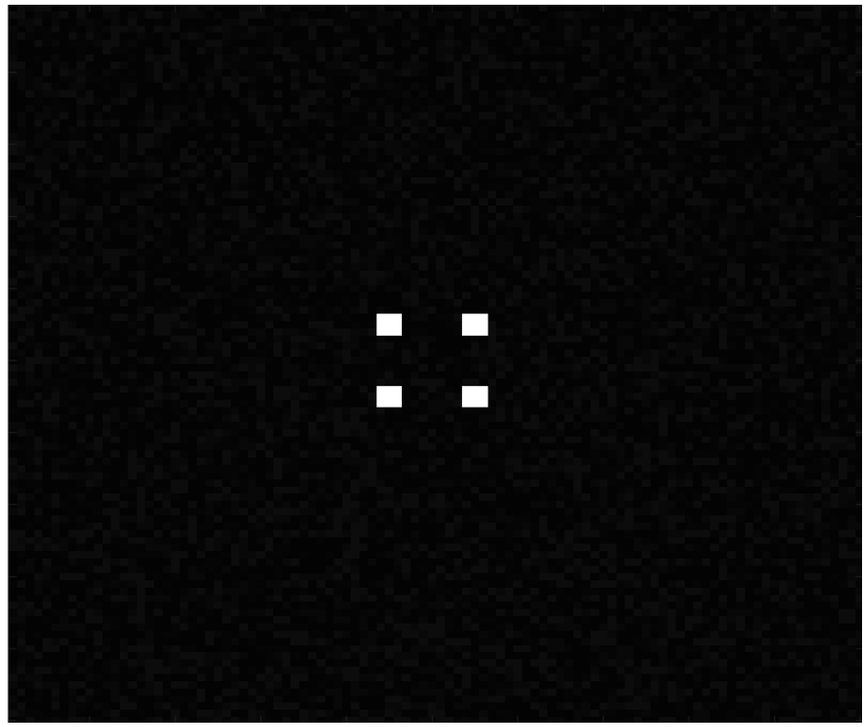
Gaussian Filter

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

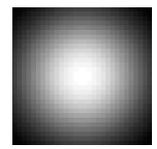




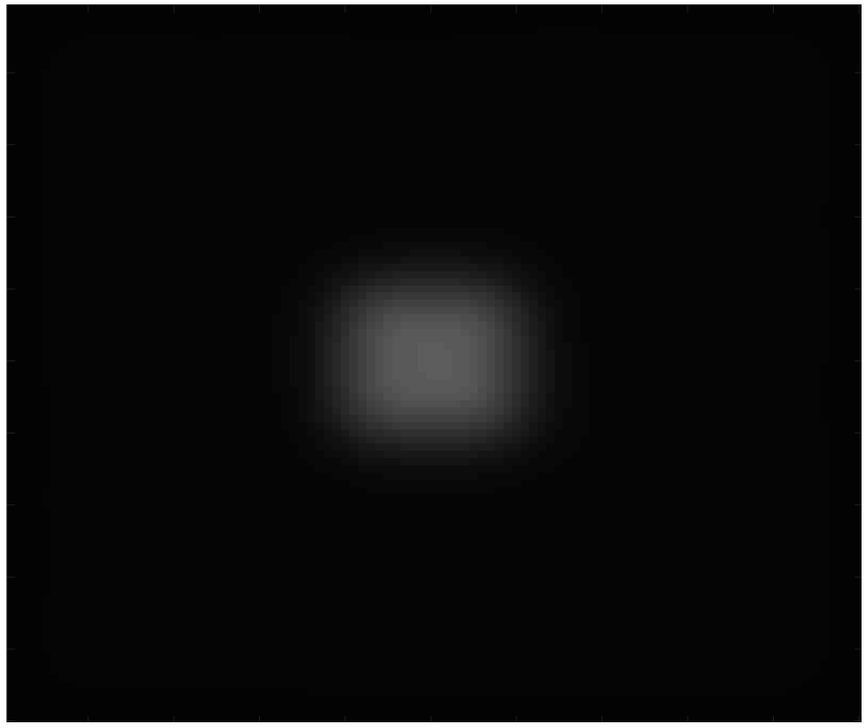
Gaussian Filter



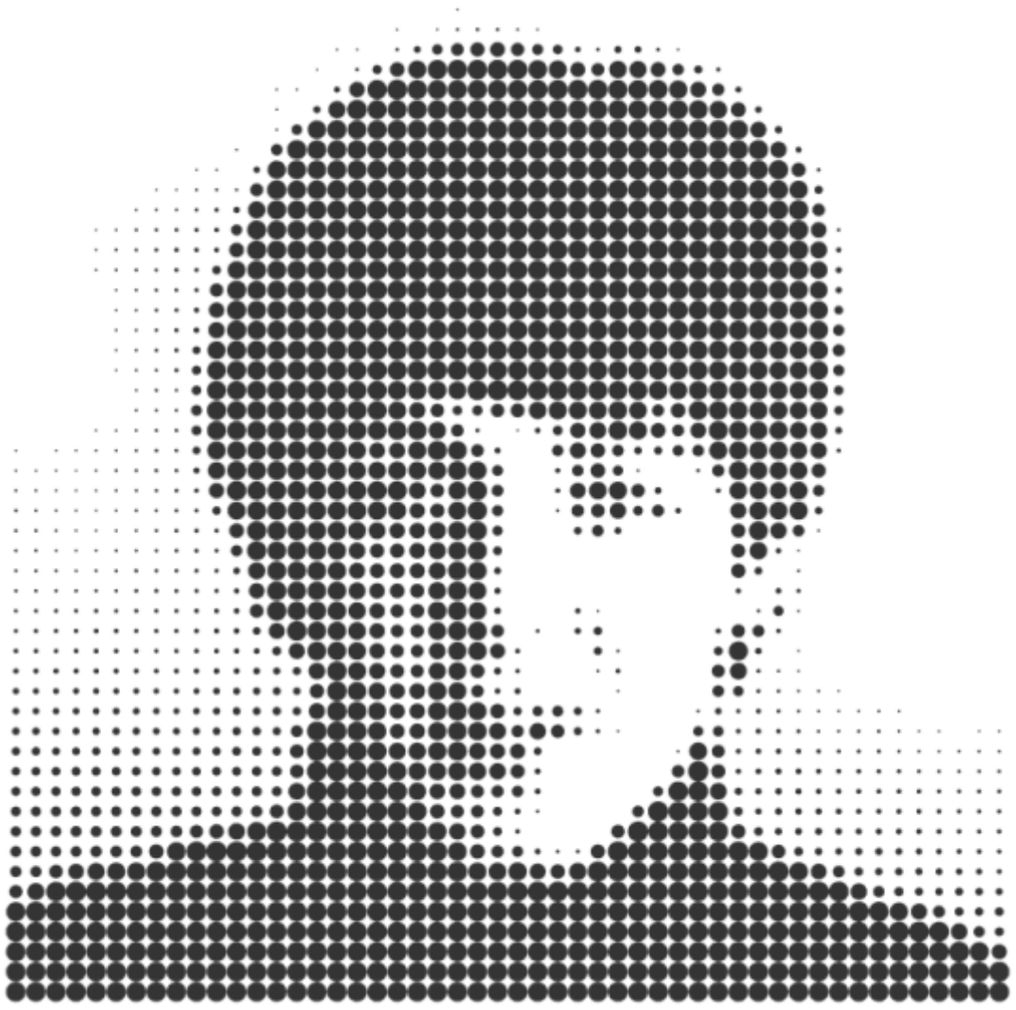
*



=



Scale



Scale

after Gaussian filtering

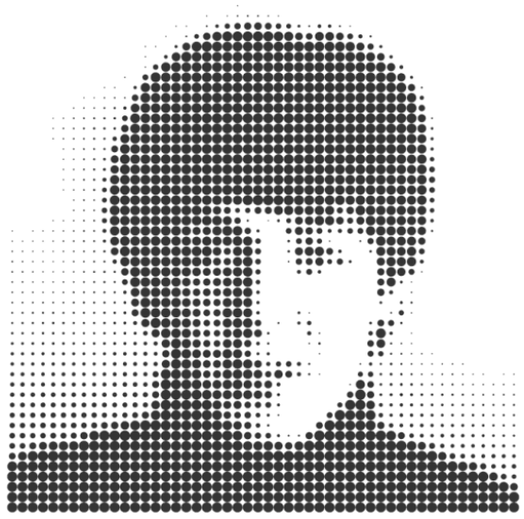
Scale Space Representation

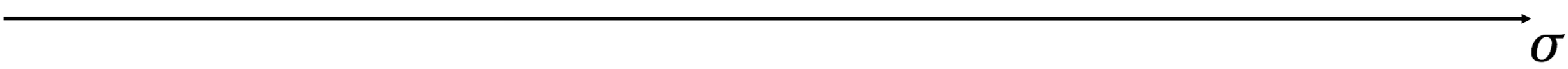
3D space defined by pixel position and blur-level: $L(x, y, \sigma^2)$

Scale Space Representation

3D space defined by pixel position and blur-level: $L(x, y, \sigma^2)$

$L(x, y, 1^2)$

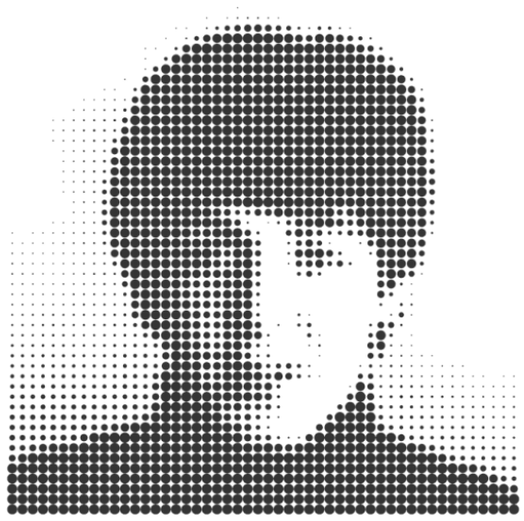




Scale Space Representation

3D space defined by pixel position and blur-level: $L(x, y, \sigma^2)$

$L(x, y, 1^2)$



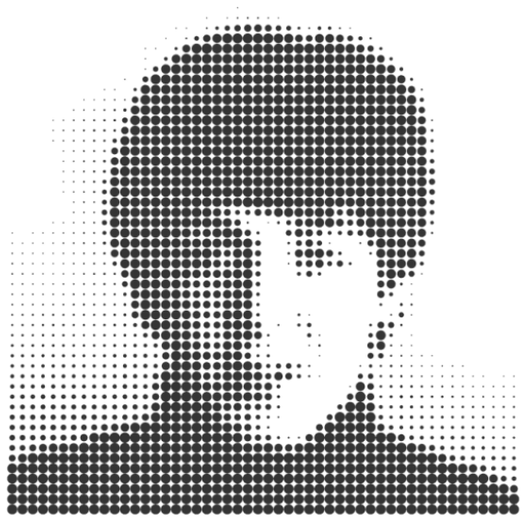
σ

$L(x, y, 2^2)$

Scale Space Representation

3D space defined by pixel position and blur-level: $L(x, y, \sigma^2)$

$L(x, y, 1^2)$



$L(x, y, 4^2)$

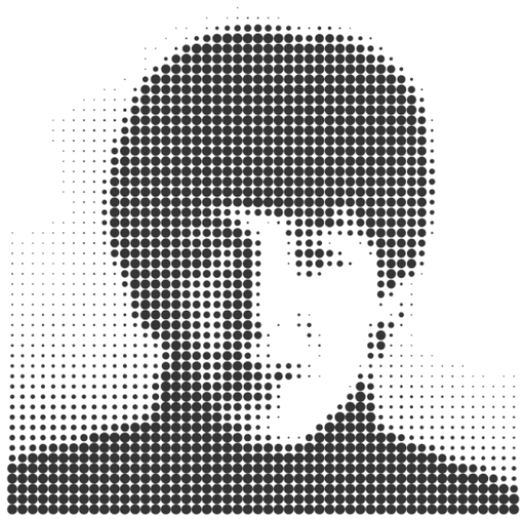
σ

$L(x, y, 2^2)$

Scale Space Representation

3D space defined by pixel position and blur-level: $L(x, y, \sigma^2)$

$L(x, y, 1^2)$



$L(x, y, 4^2)$

σ

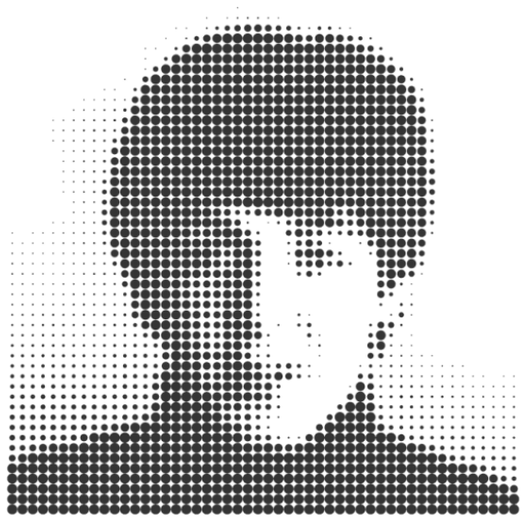
$L(x, y, 2^2)$

$L(x, y, 8^2)$

Scale Space Representation

3D space defined by pixel position and blur-level: $L(x, y, \sigma^2)$

$L(x, y, 1^2)$



$L(x, y, 4^2)$

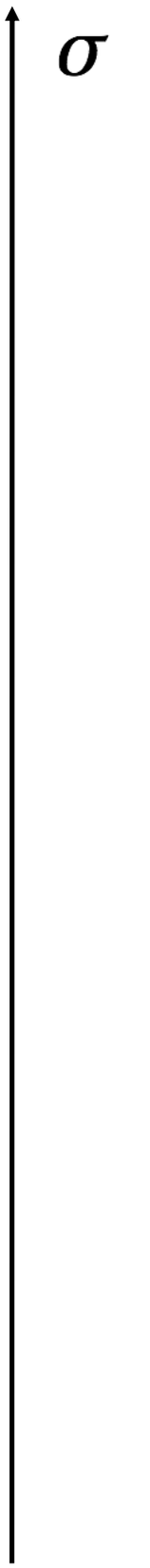
$L(x, y, 16^2)$

σ

$L(x, y, 2^2)$

$L(x, y, 8^2)$

Scale Space Pyramid



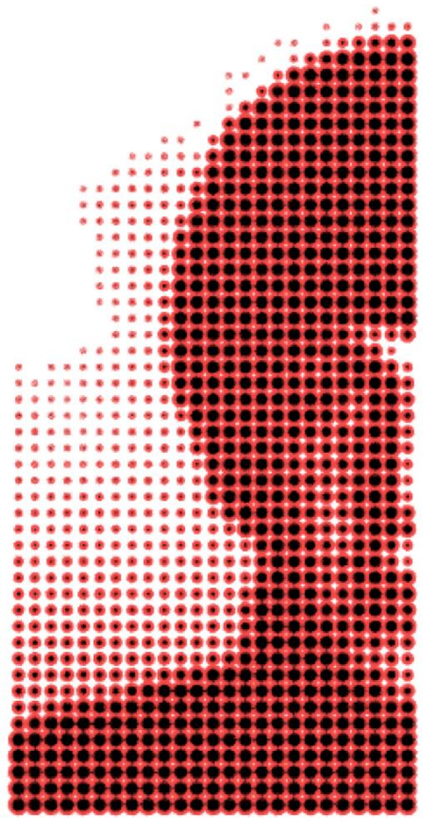
$L(x, y, 8^2)$

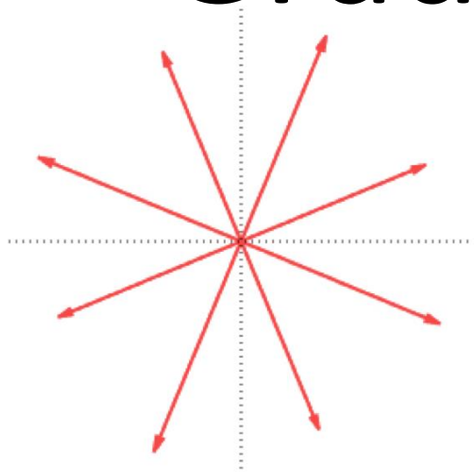
$L(x, y, 4^2)$

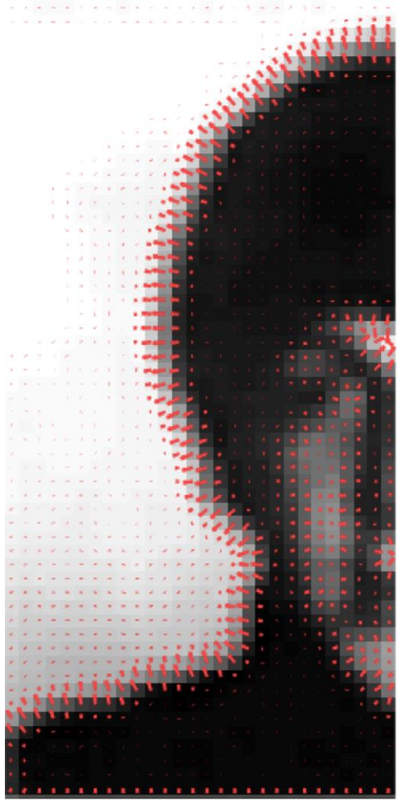
$L(x, y, 2^2)$

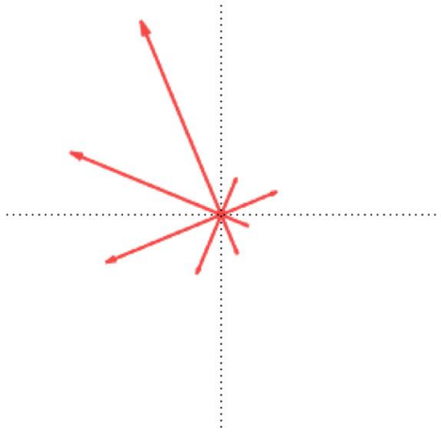
$L(x, y, 1^2)$

Gradient Histograms

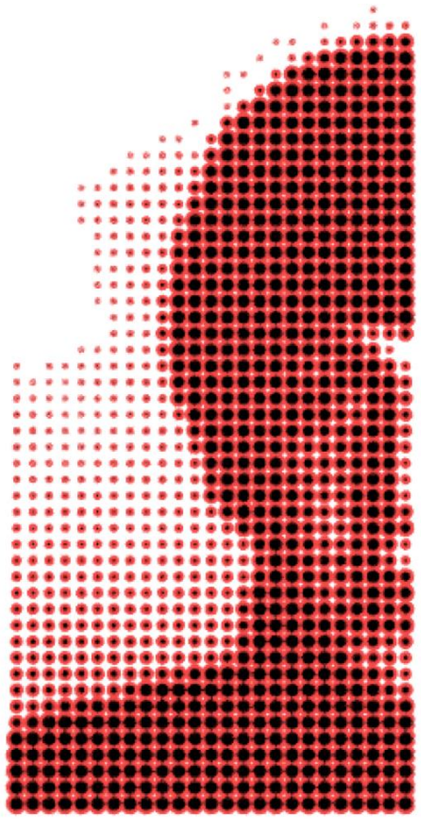


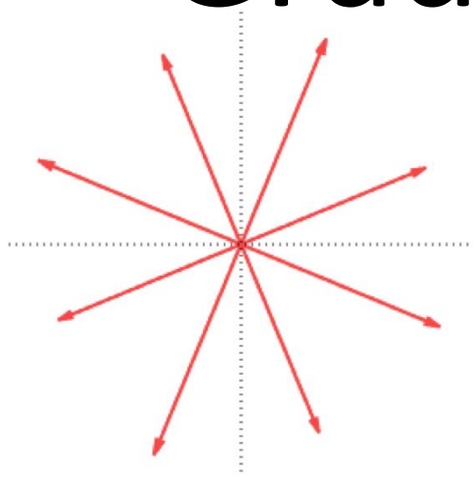


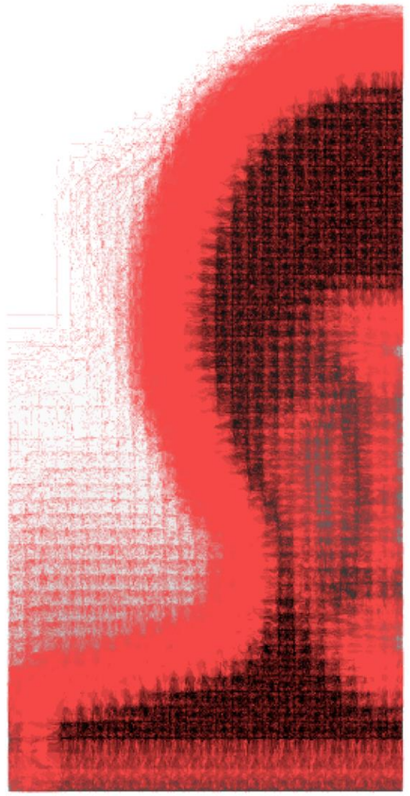


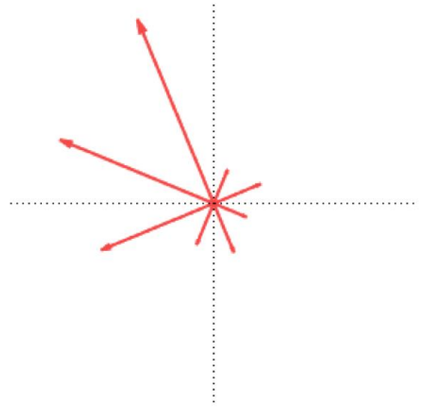


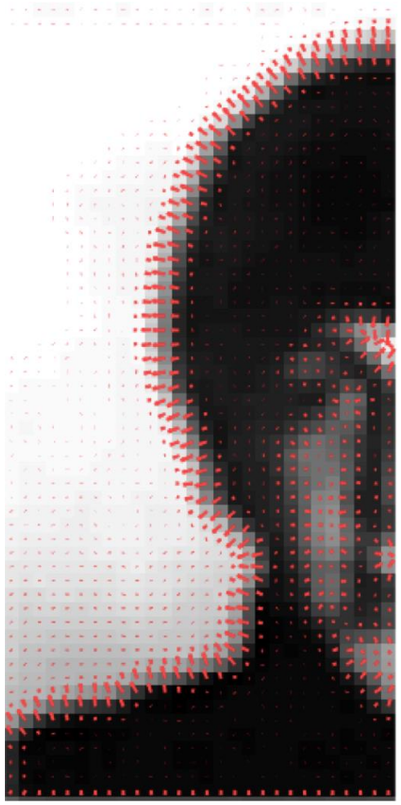
Gradient Histograms

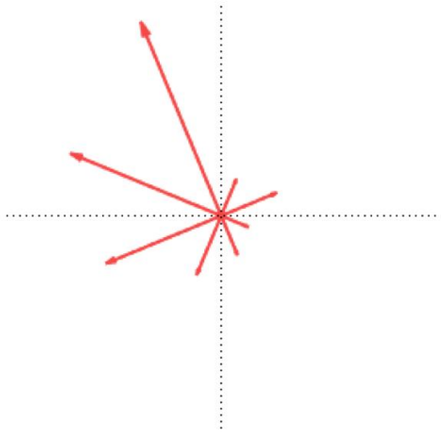




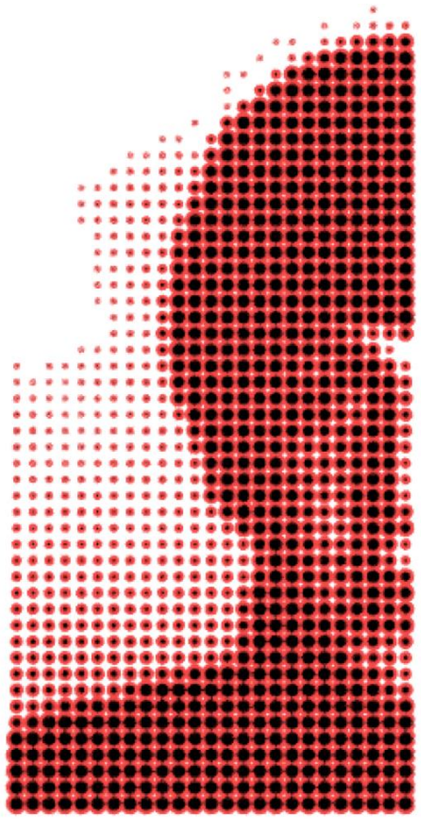


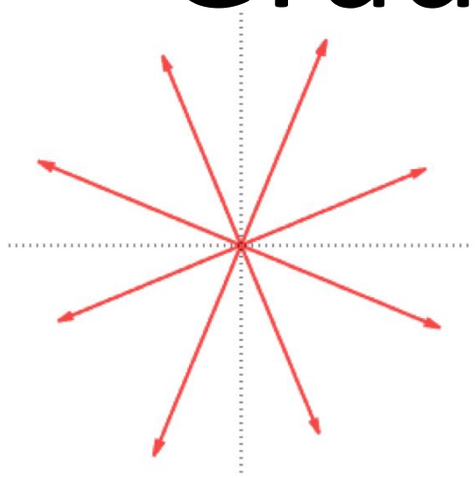


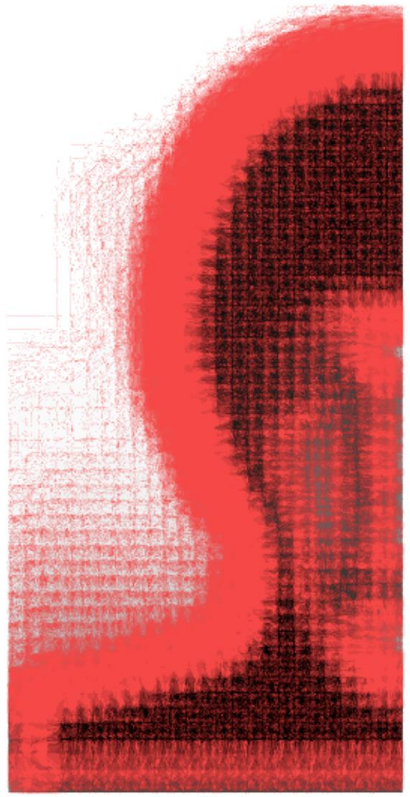


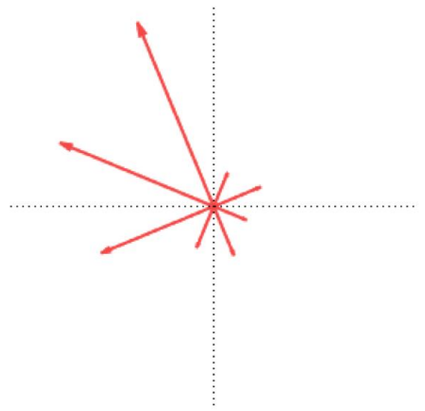


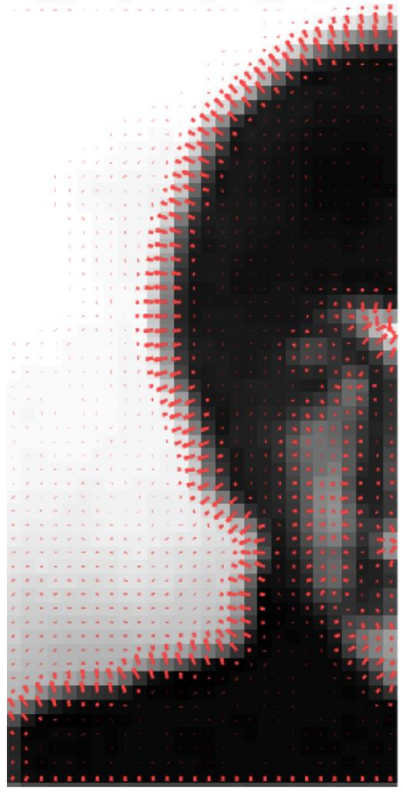
Gradient Histograms

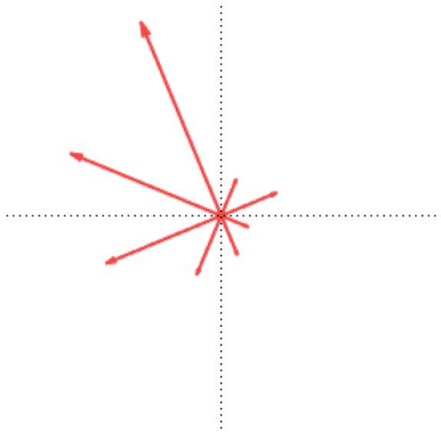






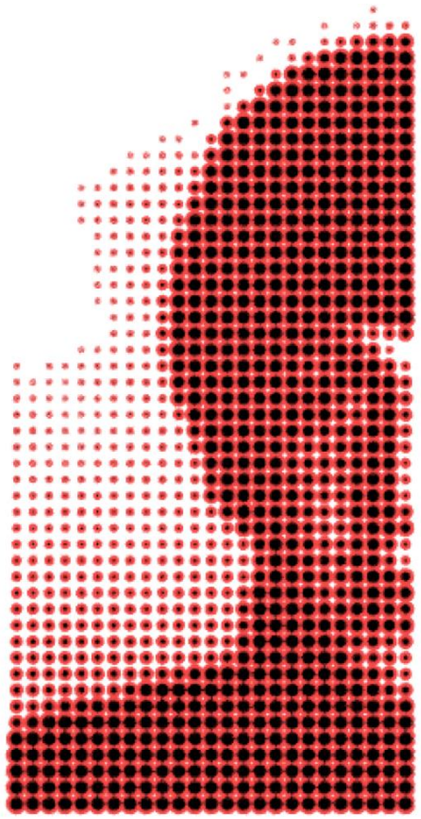


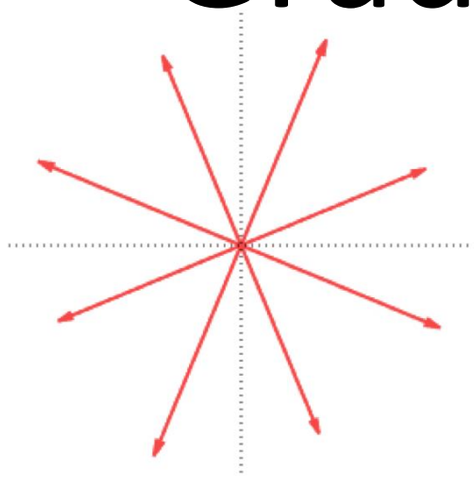


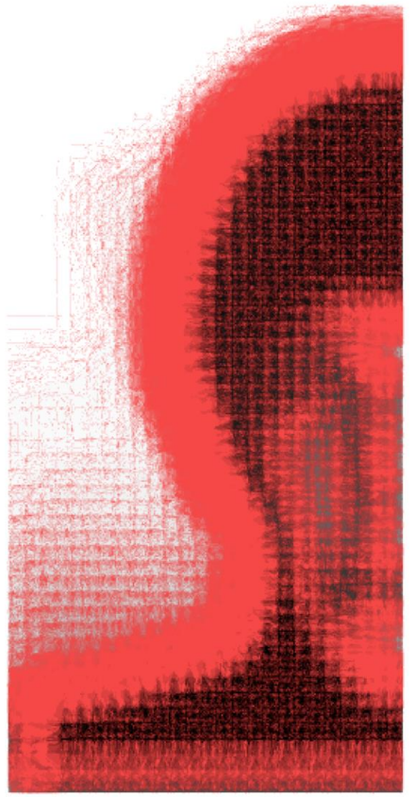


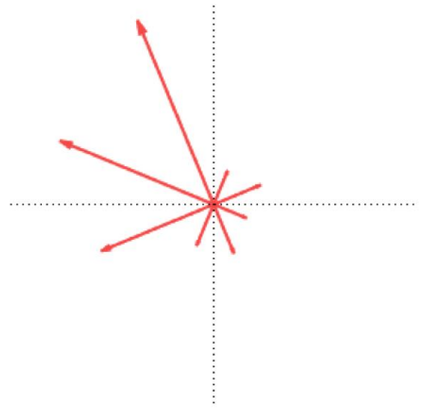
Much more similar!

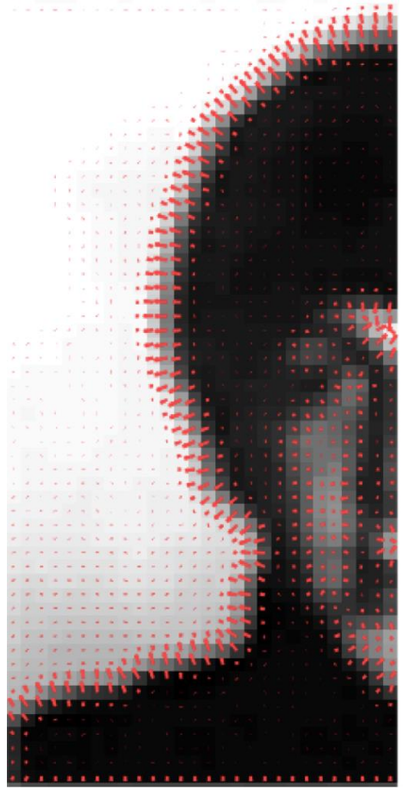
Gradient Histograms

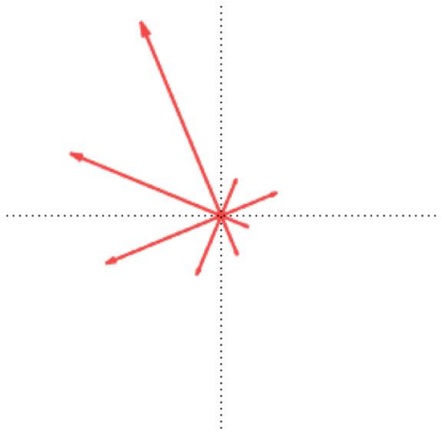












Much more similar!

… but how do we determine the correct scale?

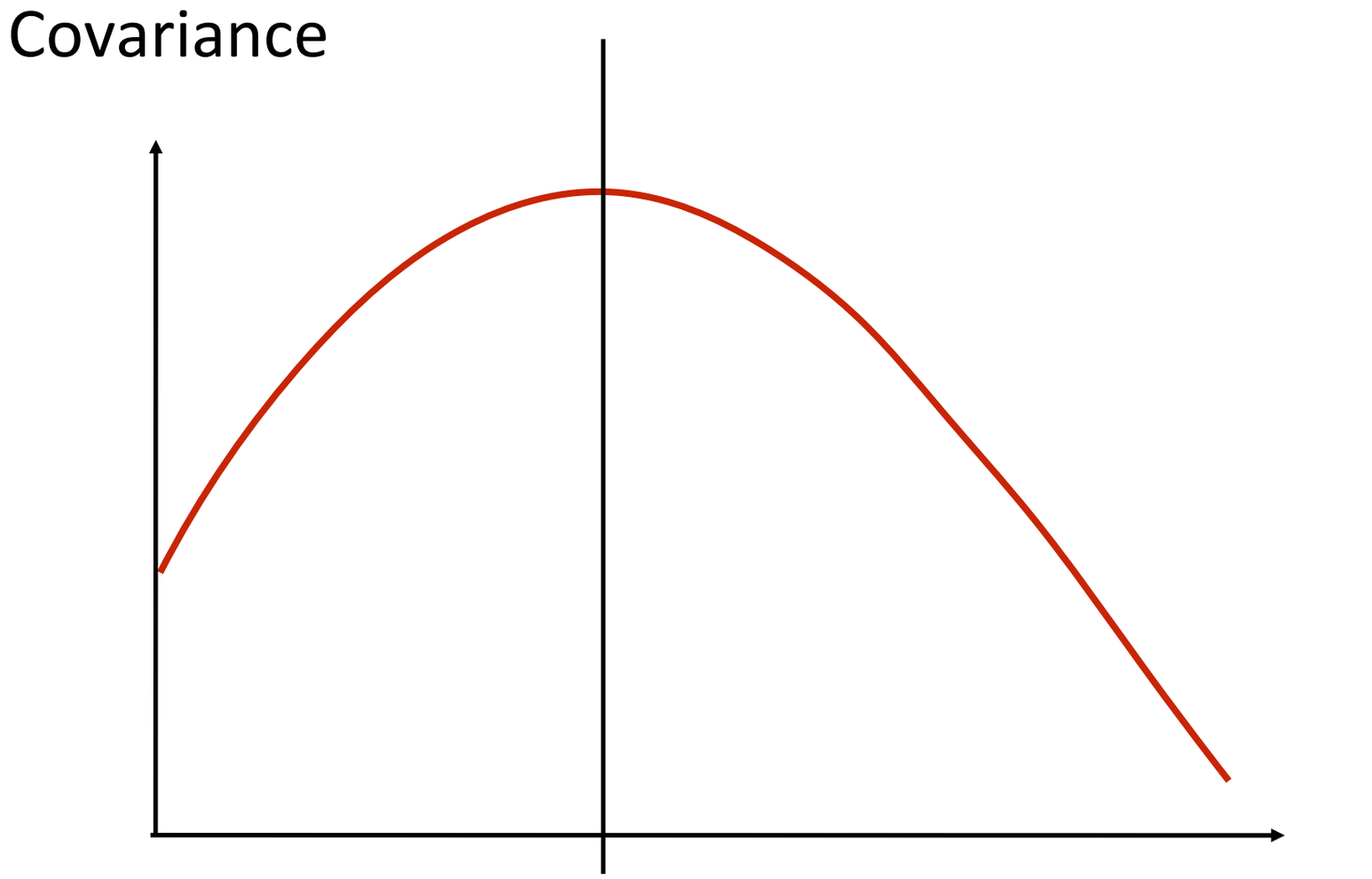
Estimating Scale

Query r =67

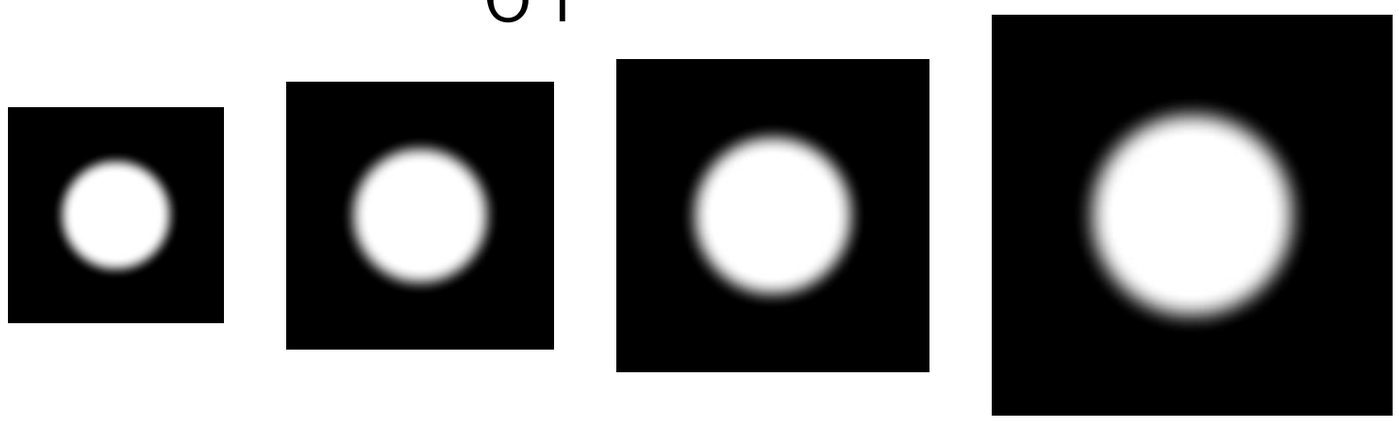
Templates:

Estimating Scale

Query r =67

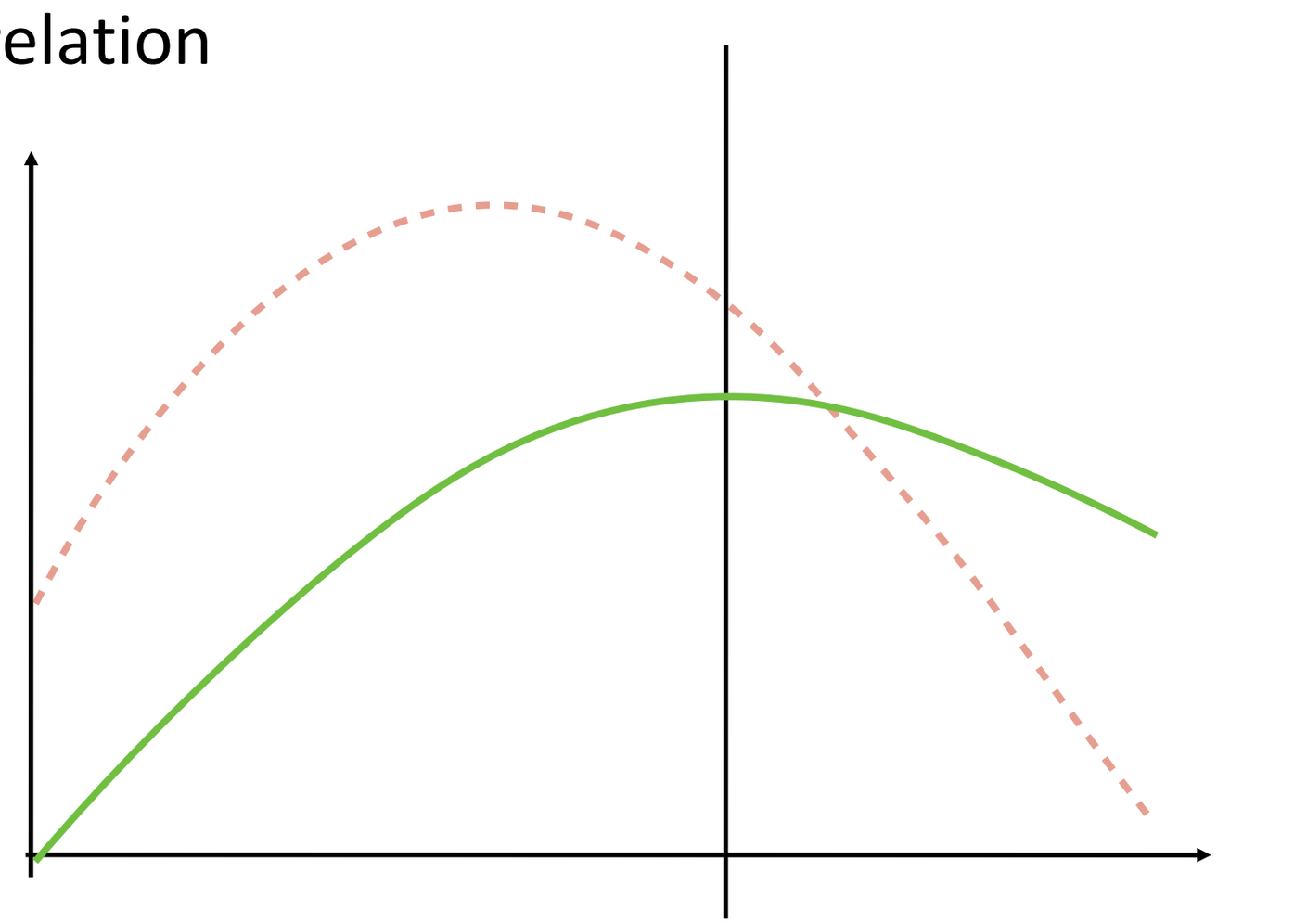


Templates:



Estimating Scale

Query. r =67



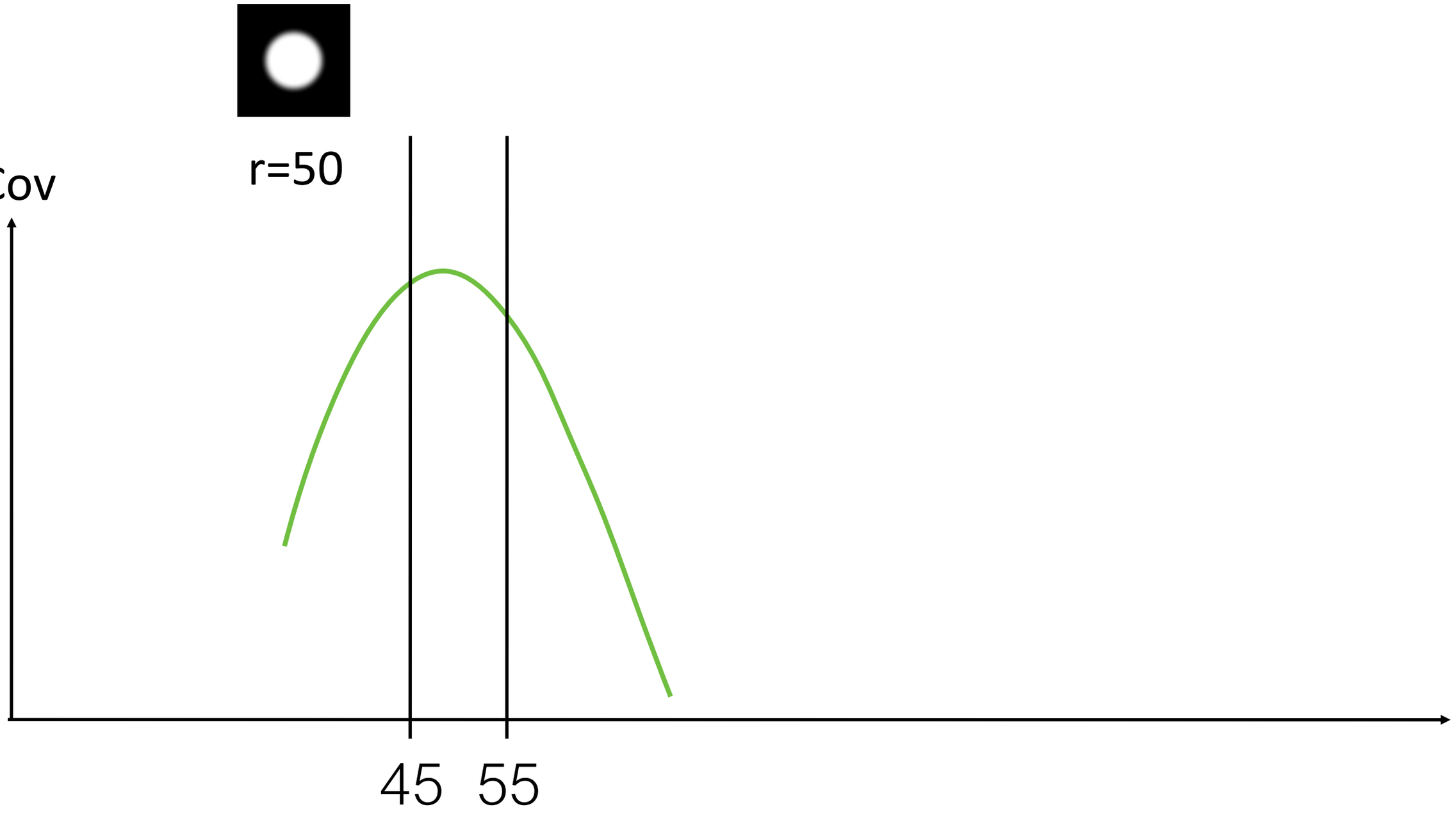
Templates:

Estimating Relative Scale

In practice: Discretized set of templates

Estimating Relative Scale

In practice: Discretized set of templates

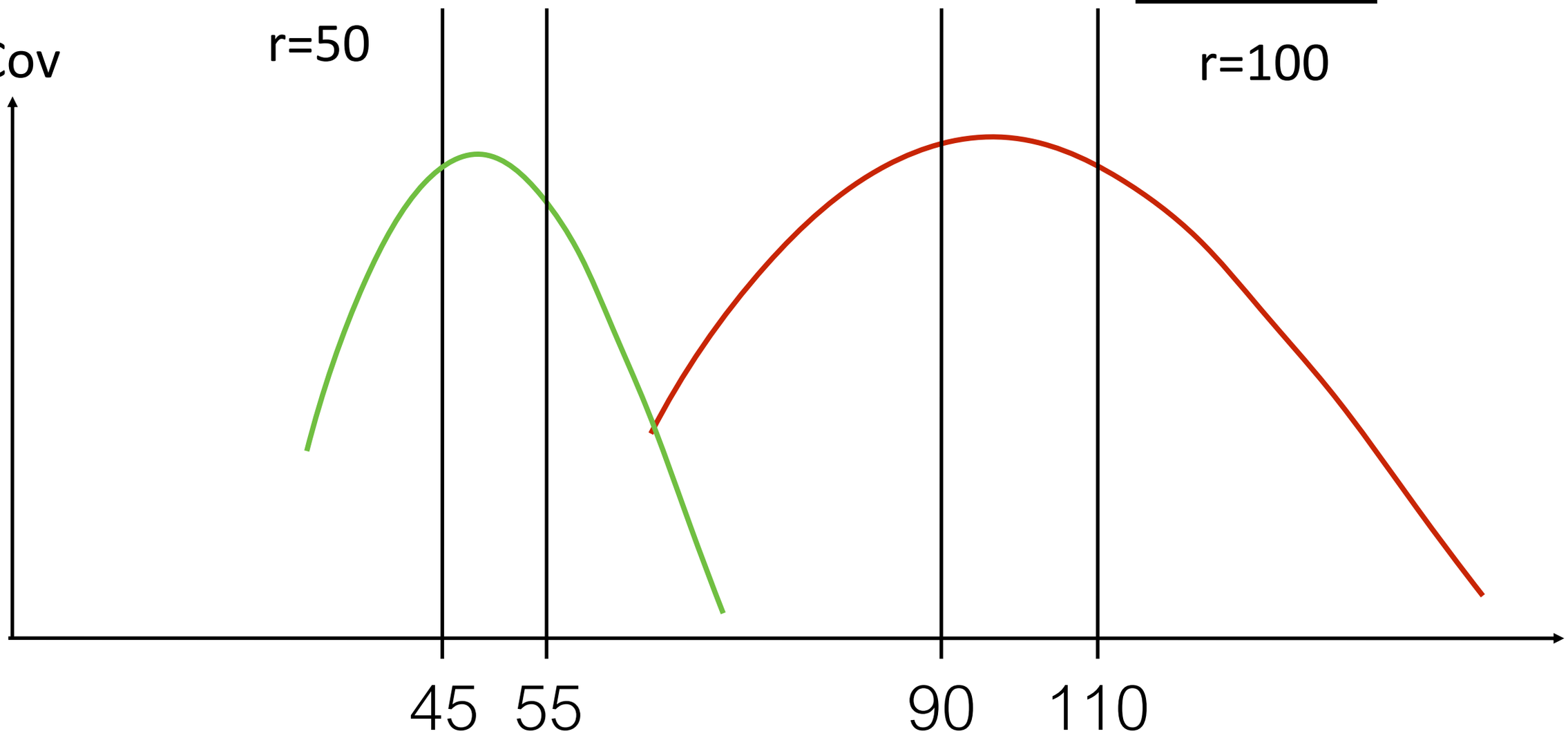


Estimating Relative Scale

In practice: Discretized set of templates

r=50

r=100

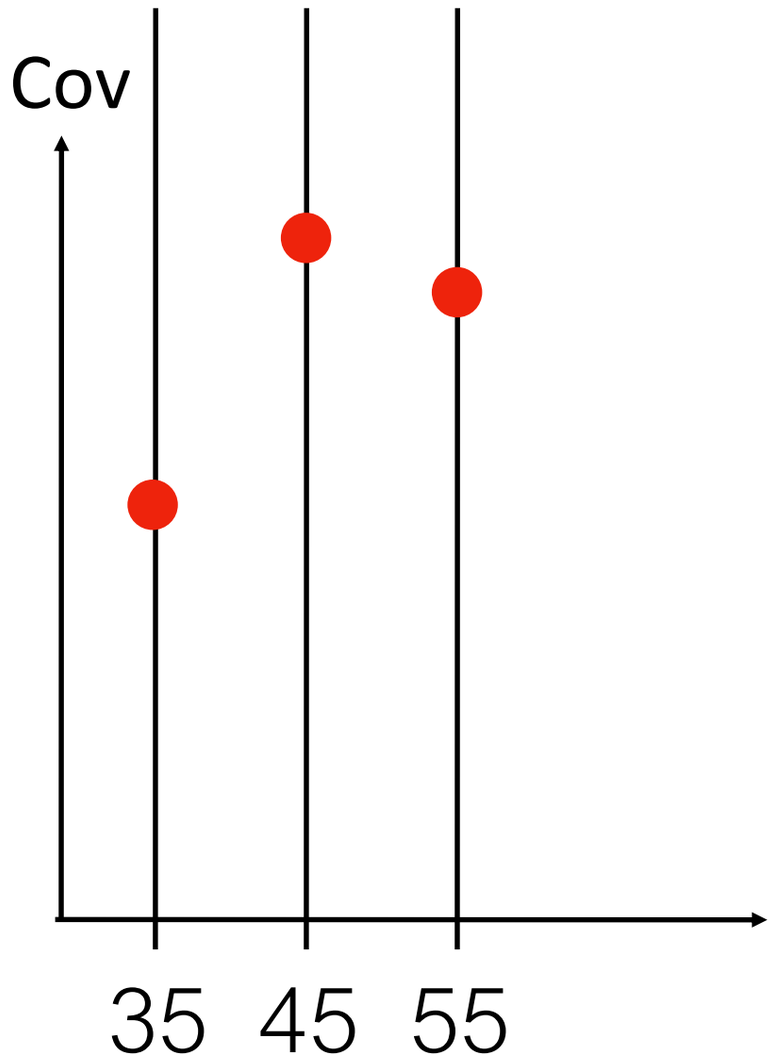


``Sub-Scale" Precision

How to determine scale more accurately?

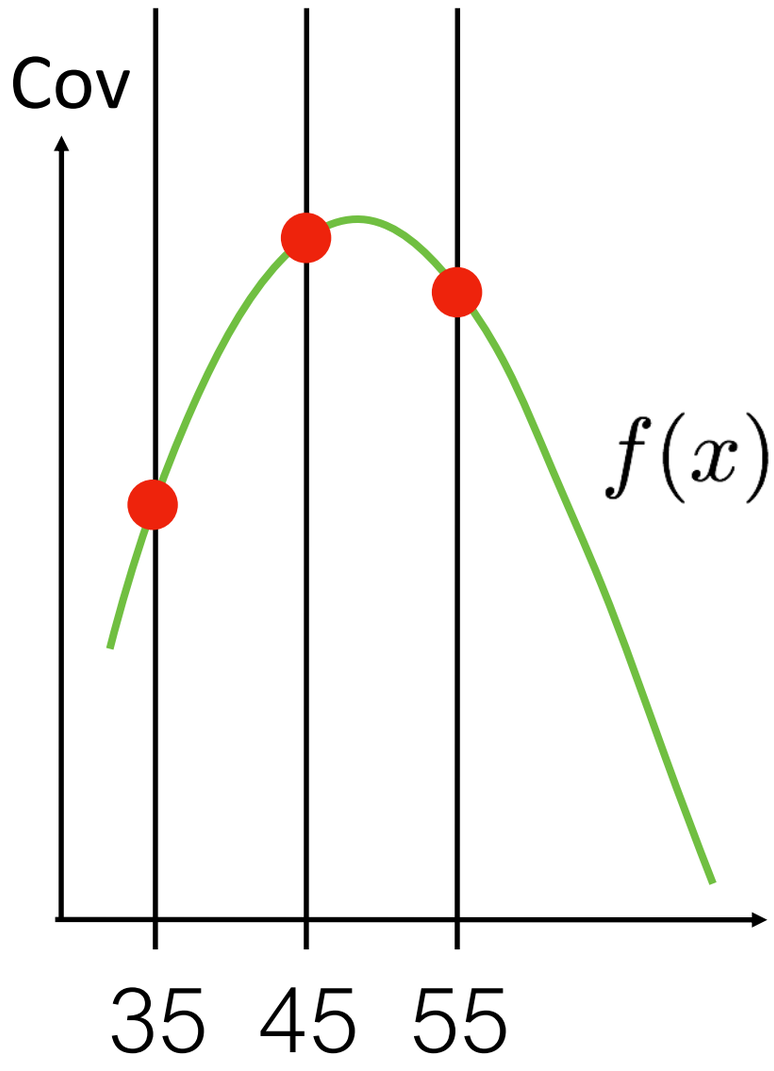
``Sub-Scale" Precision

How to determine scale more accurately?



``Sub-Scale" Precision

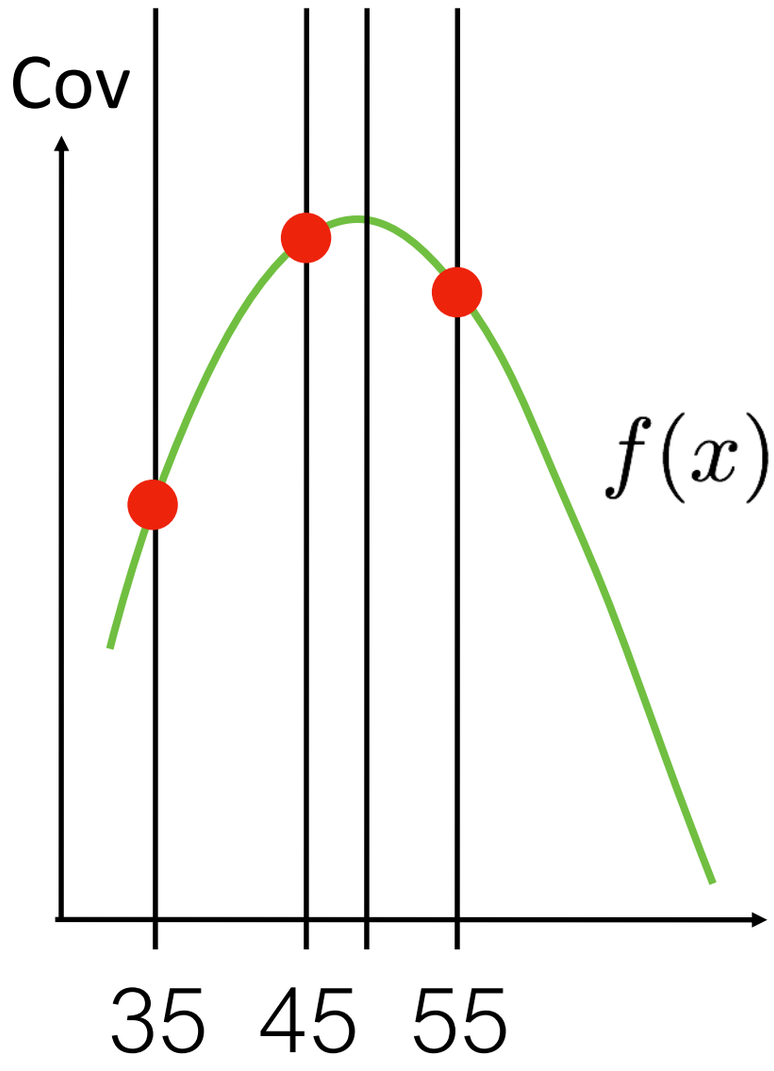
How to determine scale more accurately?



- Fit 2nd-order curve

``Sub-Scale" Precision

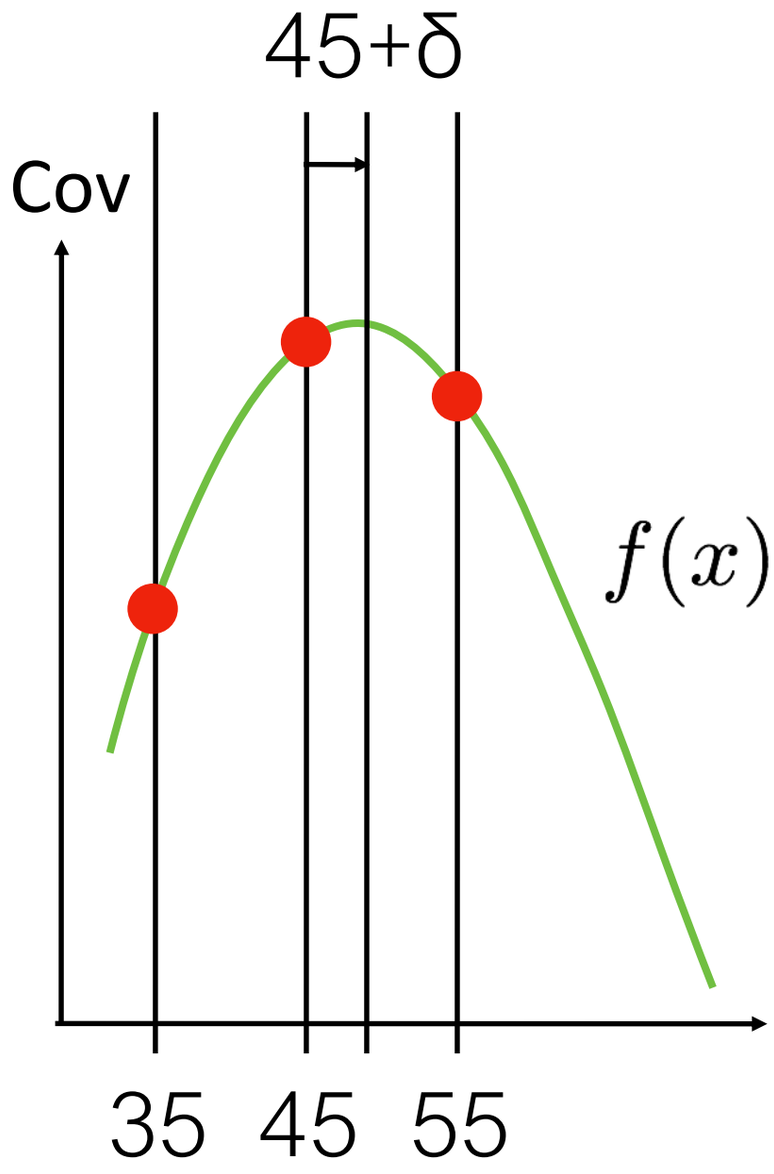
How to determine scale more accurately?



- Fit 2nd-order curve
- Determine position of maximum

``Sub-Scale" Precision

How to determine scale more accurately?



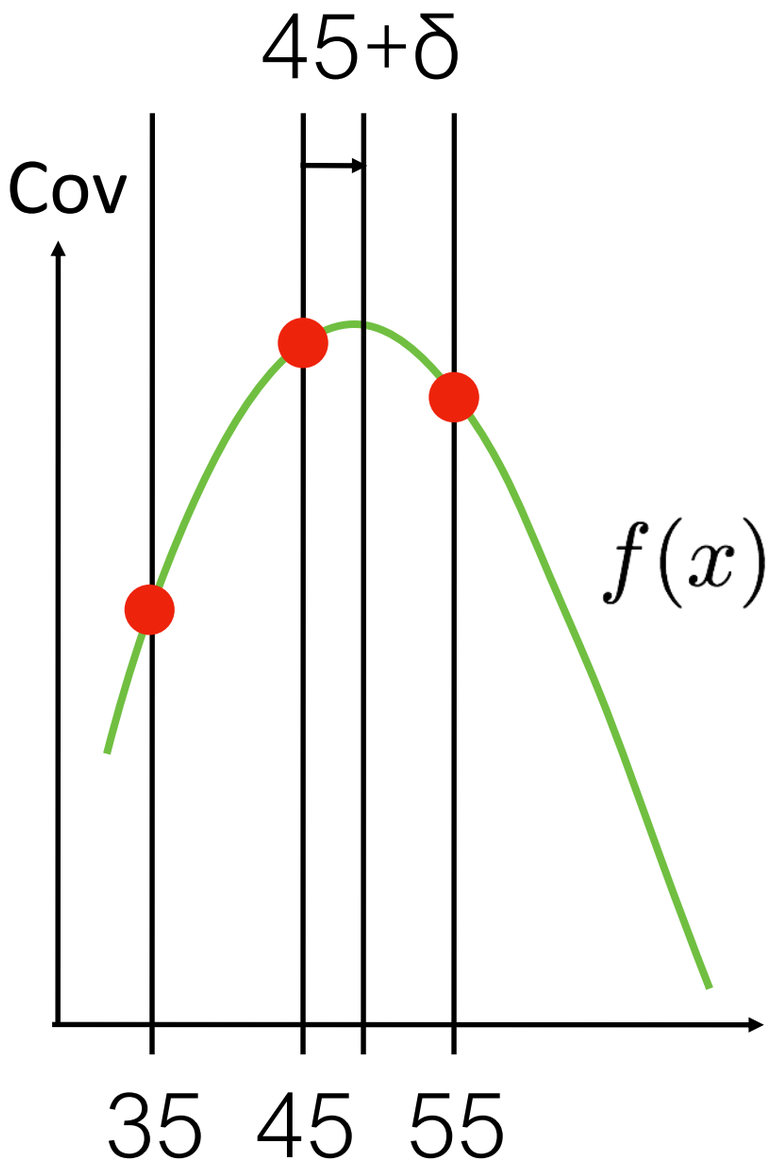
- Fit 2nd-order curve
- Determine position of maximum
- Obtain refinement of scale

``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion

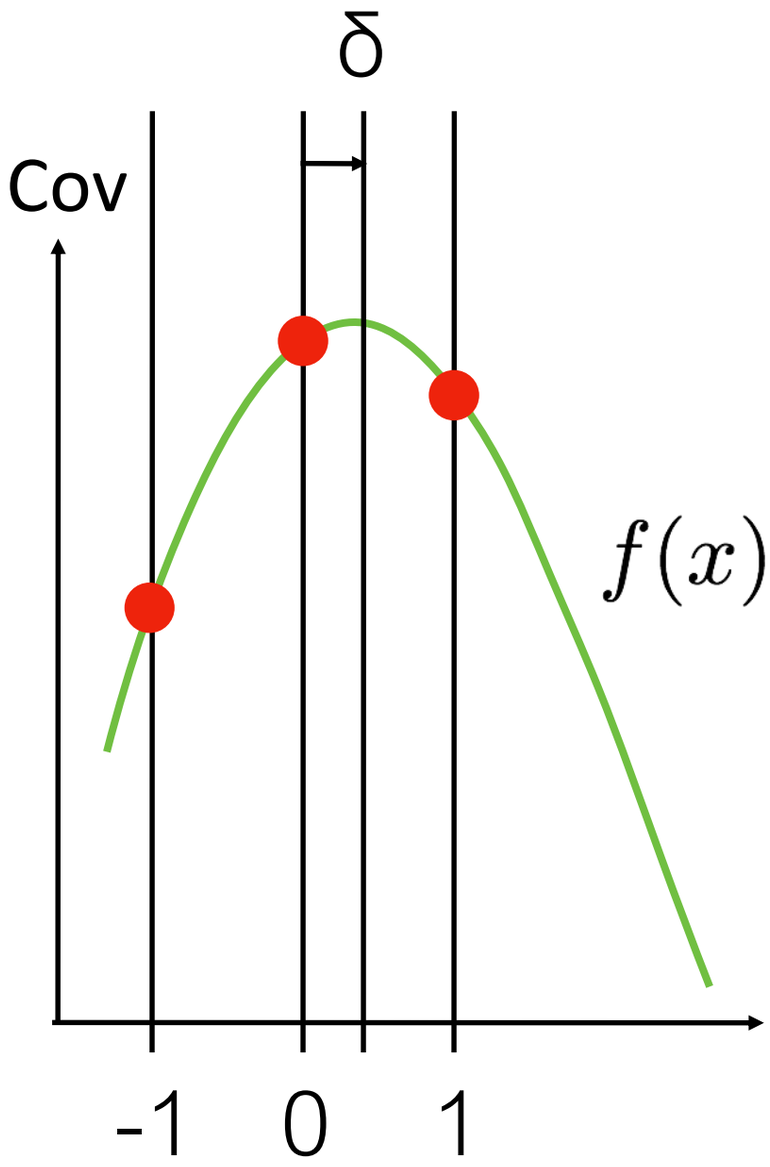
``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion



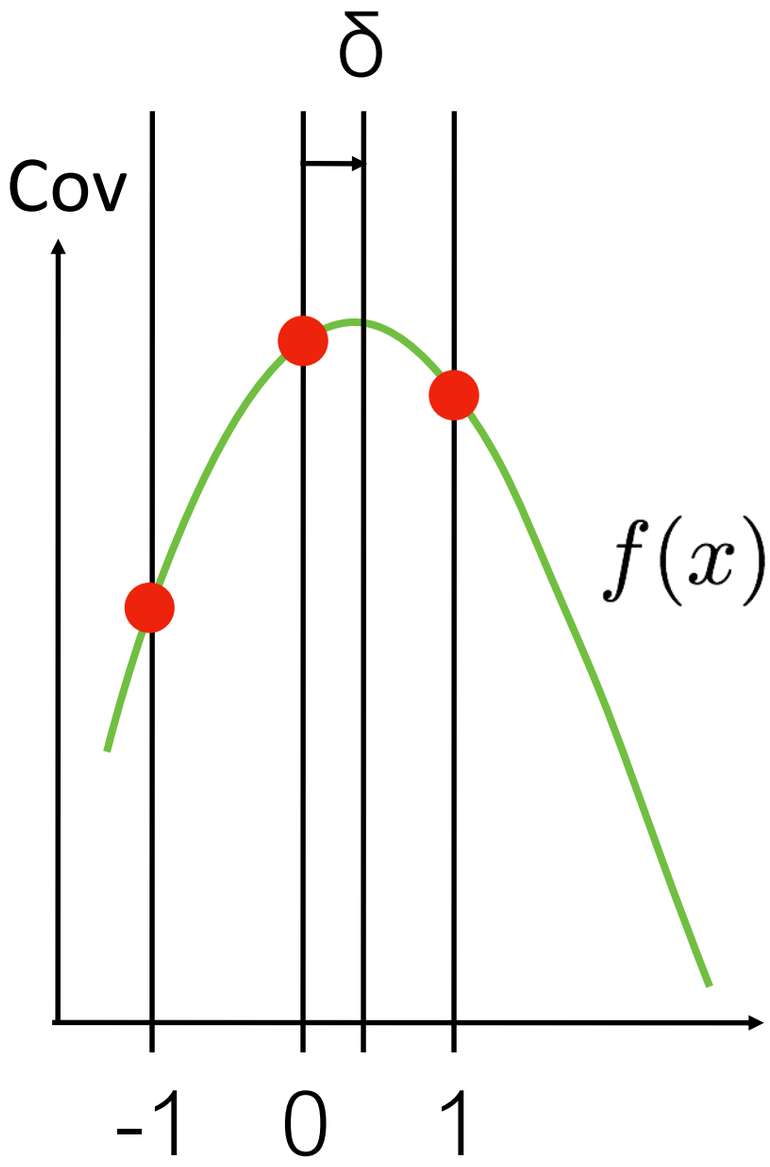
``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion



``Sub-Scale" Precision

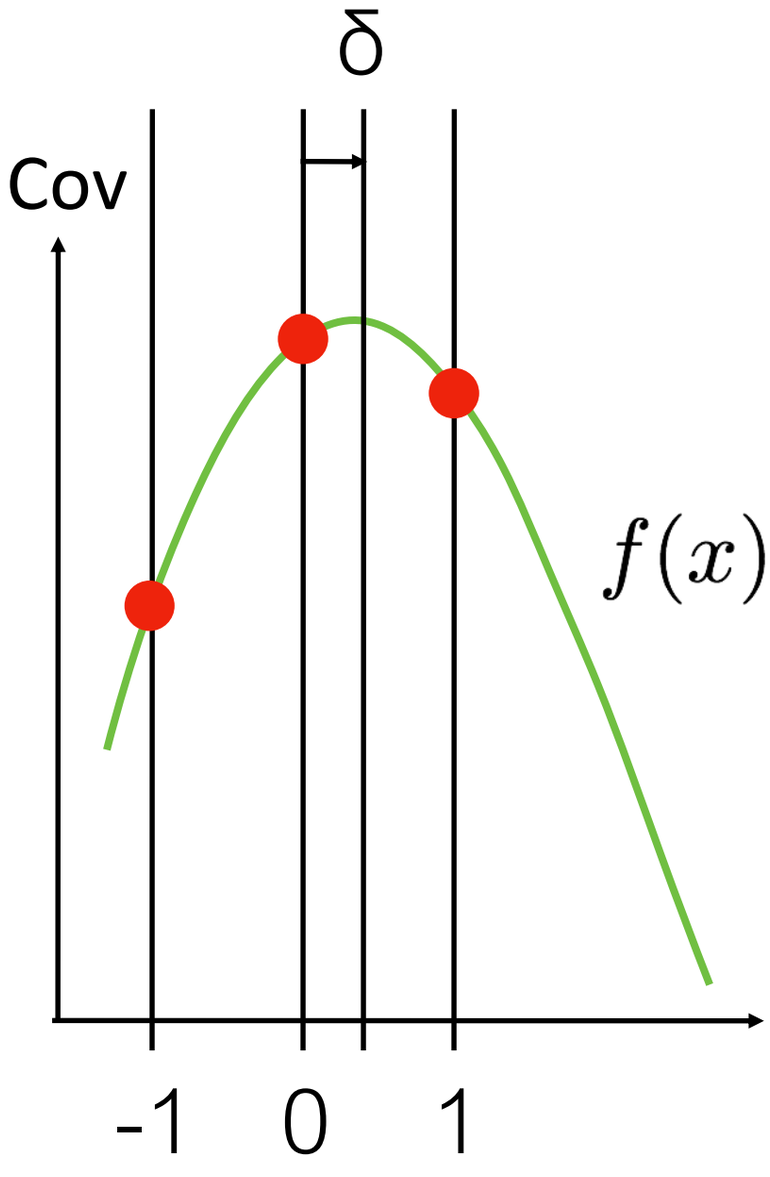
Curve fitting via 2nd-order Taylor expansion



$$f(\delta) = f(0) + \delta f'(0) + \frac{\delta^2}{2} f''(0)$$

``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion

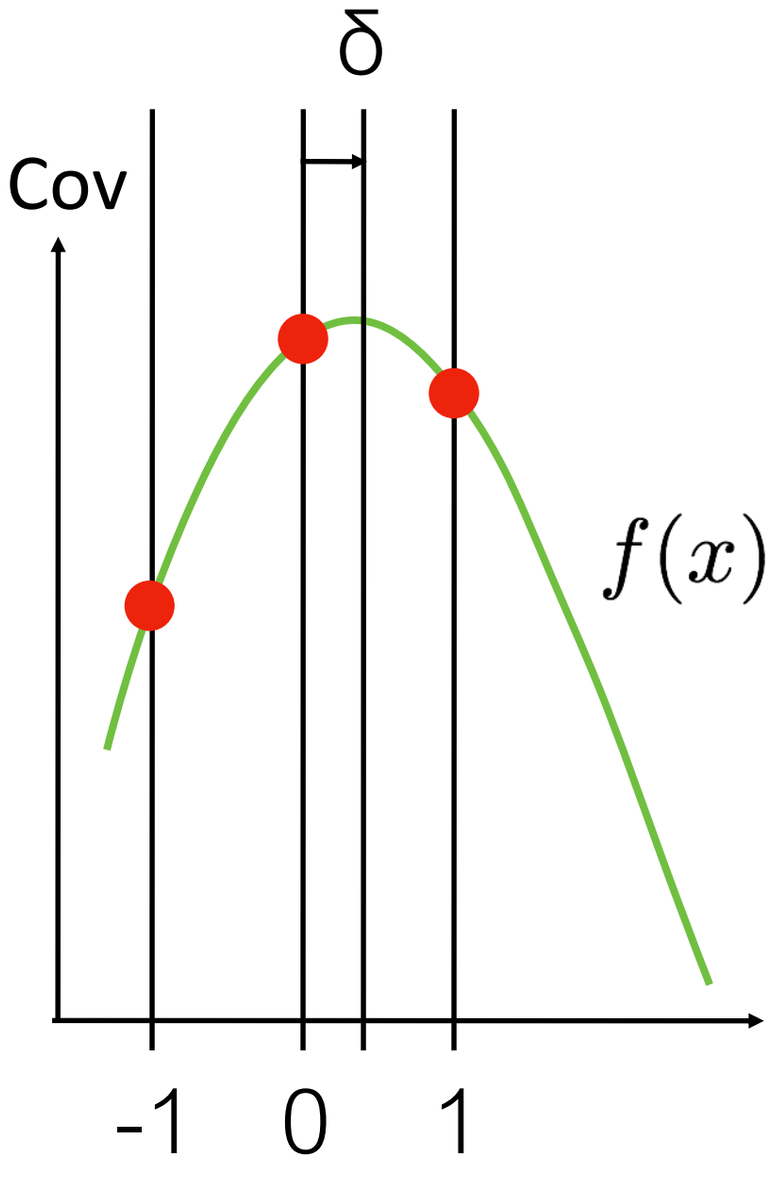


$$f(\delta) = f(0) + \delta f'(0) + \frac{\delta^2}{2} f''(0)$$

$$f'(0) = \frac{f(1) - f(-1)}{2} = a$$

``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion



$$f(\delta) = f(0) + \delta f'(0) + \frac{\delta^2}{2} f''(0)$$

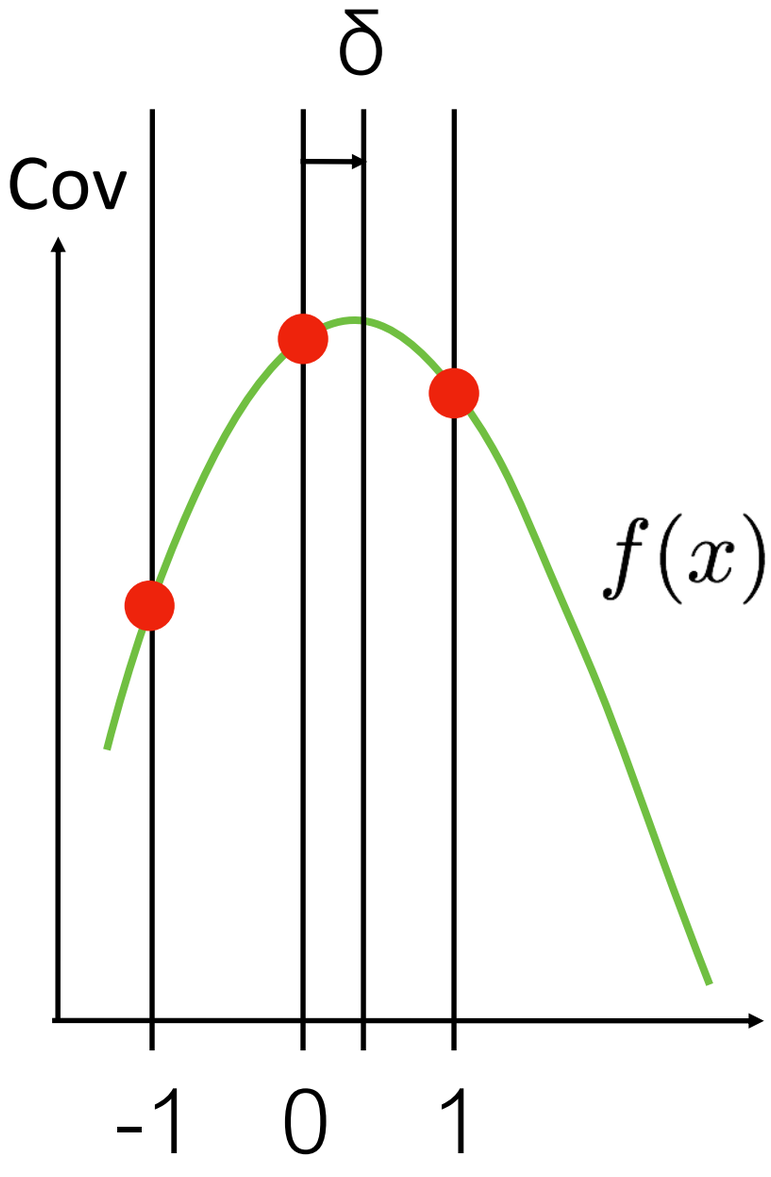
$$f'(0) = \frac{f(1) - f(-1)}{2} = a$$

$$f''(0) = f(1) - 2f(0) + f(-1) = b$$

discrete Laplacian

``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion



$$f(\delta) = f(0) + \delta f'(0) + \frac{\delta^2}{2} f''(0)$$

$$f'(0) = \frac{f(1) - f(-1)}{2} = a$$

$$f''(0) = f(1) - 2f(0) + f(-1) = b$$

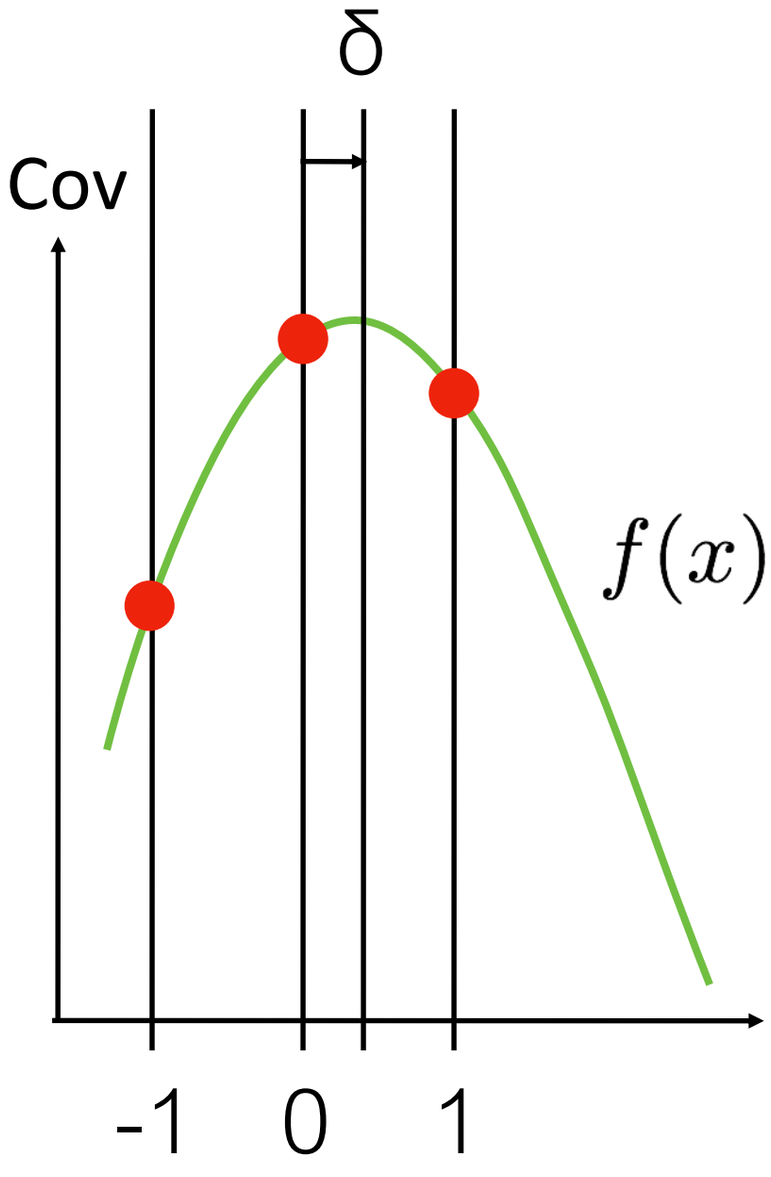
discrete Laplacian

Compute maximum:

$$f'(\delta) = a + b\delta$$

``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion



$$f(\delta) = f(0) + \delta f'(0) + \frac{\delta^2}{2} f''(0)$$

$$f'(0) = \frac{f(1) - f(-1)}{2} = a$$

$$f''(0) = f(1) - 2f(0) + f(-1) = b$$

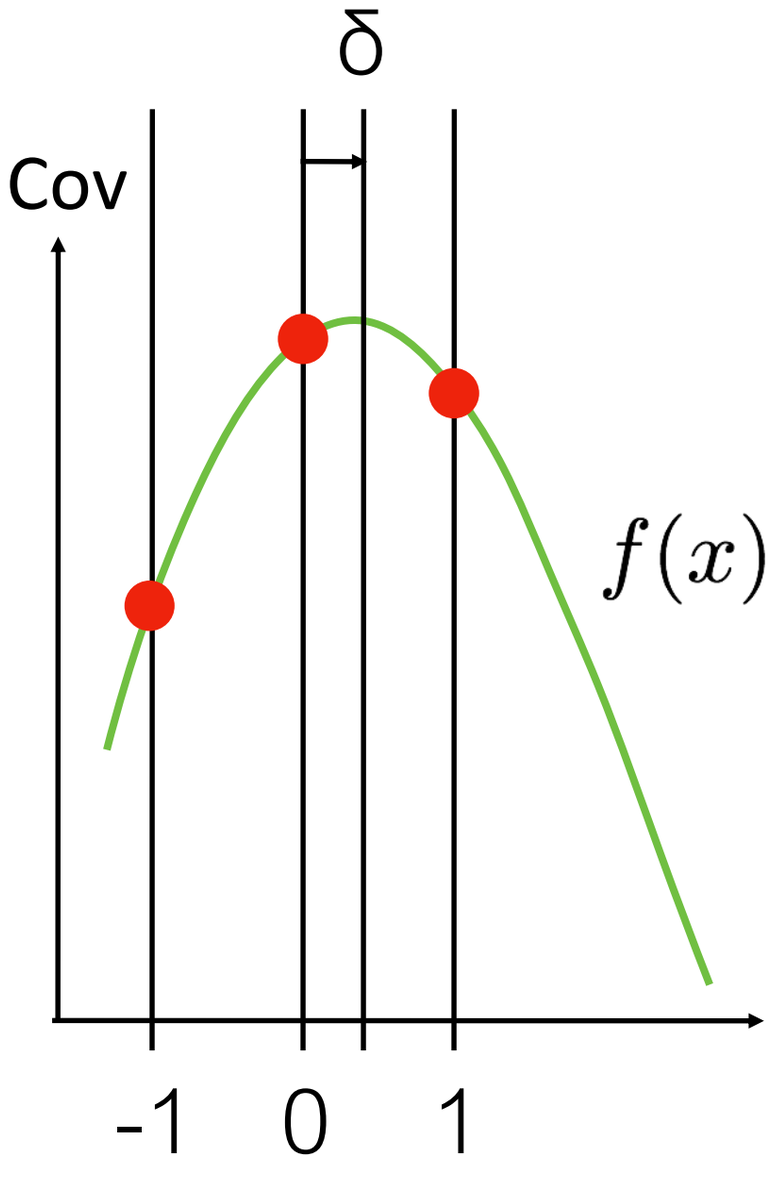
discrete Laplacian

Compute maximum:

$$f'(\delta) = a + b\delta \stackrel{!}{=} 0$$

``Sub-Scale" Precision

Curve fitting via 2nd-order Taylor expansion



$$f(\delta) = f(0) + \delta f'(0) + \frac{\delta^2}{2} f''(0)$$

$$f'(0) = \frac{f(1) - f(-1)}{2} = a$$

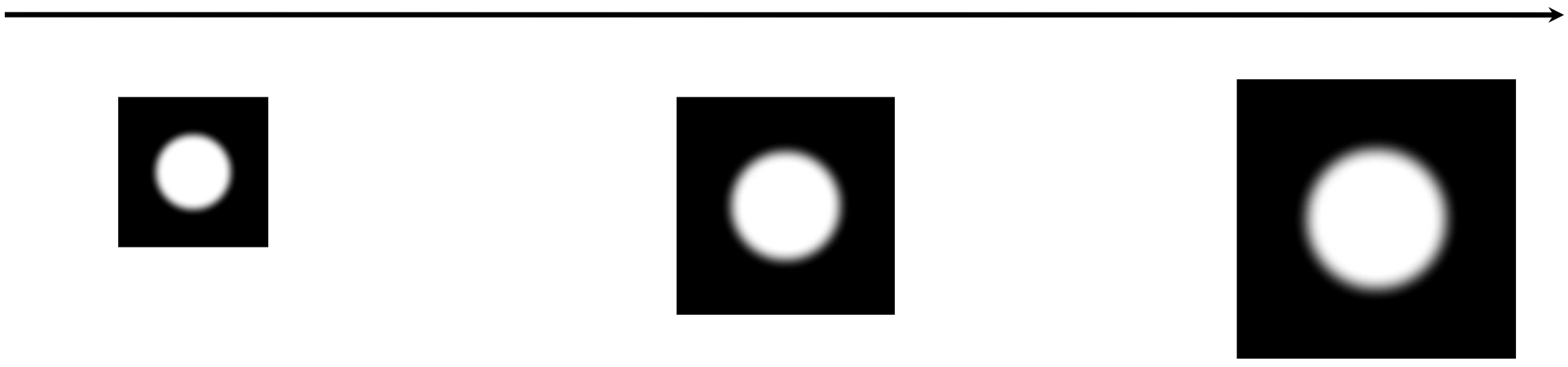
$$f''(0) = f(1) - 2f(0) + f(-1) = b$$

discrete Laplacian

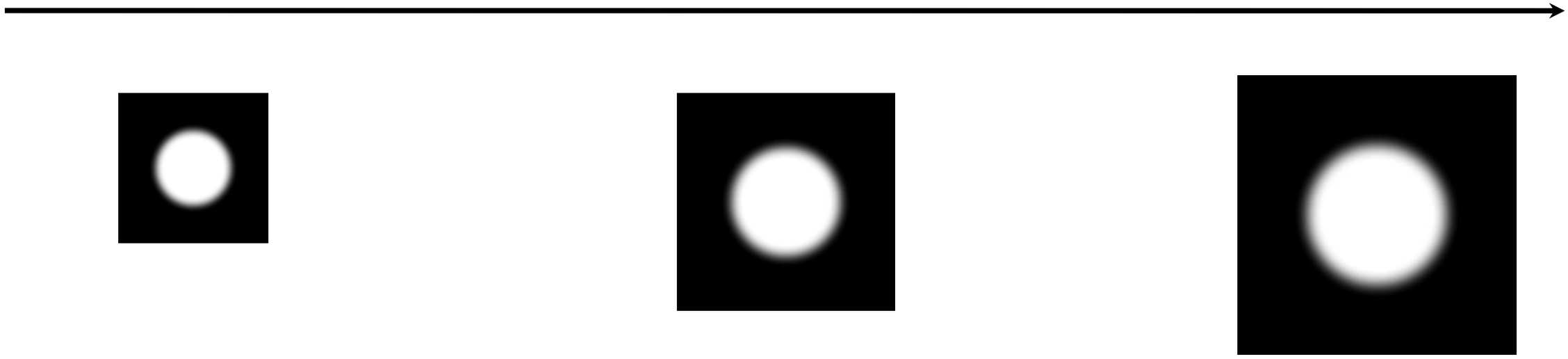
Compute maximum:

$$f'(\delta) = a + b\delta \stackrel{!}{=} 0 \Rightarrow \delta = \frac{-a}{b}$$

Detect Scale and Position



Detect Scale and Position



1	0	1	2	3	4	2
2	7	8	2	3	3	3
2	3	1	4	5	6	2
2	7	12	15	20	7	2
2	8	12	25	19	11	3
3	9	11	18	16	7	1
1	2	3	2	1	3	1

2	1	2	2	2	3	3
1	2	3	2	2	3	4
5	11	9	8	9	8	1
1	11	15	20	22	9	5
2	12	16	35	22	11	4
3	9	15	18	21	7	1
2	2	9	8	7	9	1

3	3	3	3	3	3	3
3	4	4	4	4	4	3
3	5	6	6	6	5	2
2	7	12	12	12	7	2
2	5	12	20	13	11	3
3	9	12	13	13	9	1
1	2	5	5	5	3	1

Detect Scale and Position

1	0	1	2	3	4	2
2	7	8	2	3	3	3
2	3	1	4	5	6	2
2	7	12	15	20	7	2
2	8	12	25	19	11	3
3	9	11	18	16	7	1
1	2	3	2	1	3	1

2	1	2	2	2	3	3
1	2	3	2	2	3	4
5	11	9	8	9	8	1
1	11	15	20	22	9	5
2	12	16	35	22	11	4
3	9	15	18	21	7	1
2	2	9	8	7	9	1

3	3	3	3	3	3	3
3	4	4	4	4	4	3
3	5	6	6	6	5	2
2	7	12	12	12	7	2
2	5	12	20	13	11	3
3	9	12	13	13	9	1
1	2	5	5	5	3	1

Larger than neighbors in image **and scale**!

Sub-Pixel Refinement

2nd order curve fitting (1D):

$$f(x) \approx f(0) + x \cdot f'(0) + \frac{x^2}{2} f''(0)$$

Sub-Pixel Refinement

2nd order curve fitting (1D):

$$f(x) \approx f(0) + x \cdot f'(0) + \frac{x^2}{2} f''(0)$$

2nd order surface fitting (3D):

$$f(x, y, \sigma) \approx f(0,0,0) + \begin{pmatrix} x & y & \sigma \end{pmatrix} \cdot \nabla f(0,0,0) + \frac{1}{2} \begin{pmatrix} x & y & \sigma \end{pmatrix} \mathtt{H}(0,0,0) \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

Sub-Pixel Refinement

2nd order curve fitting (1D):

$$f(x) \approx f(0) + x \cdot f'(0) + \frac{x^2}{2} f''(0)$$

2nd order surface fitting (3D):

$$f(x, y, \sigma) \approx f(0,0,0) + \begin{pmatrix} x & y & \sigma \end{pmatrix} \cdot \nabla f(0,0,0) + \frac{1}{2} \begin{pmatrix} x & y & \sigma \end{pmatrix} \mathtt{H}(0,0,0) \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

Hessian matrix:

$$\mathtt{H}(0,0,0) = \begin{pmatrix} f''_{xx}(0,0,0) & f''_{xy}(0,0,0) & f''_{x\sigma}(0,0,0) \\ f''_{xy}(0,0,0) & f''_{yy}(0,0,0) & f''_{y\sigma}(0,0,0) \\ f''_{x\sigma}(0,0,0) & f''_{y\sigma}(0,0,0) & f''_{\sigma\sigma}(0,0,0) \end{pmatrix}$$

Sub-Pixel Refinement

2nd order surface fitting (3D):

$$f(x,y,\sigma) \approx f(0,0,0) + \begin{pmatrix} x & y & \sigma \end{pmatrix} \cdot \nabla f(0,0,0) + \frac{1}{2} \begin{pmatrix} x & y & \sigma \end{pmatrix} \mathrm{H}(0,0,0) \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

Sub-Pixel Refinement

2nd order surface fitting (3D):

$$f(x,y,\sigma) \approx f(0,0,0) + \begin{pmatrix} x & y & \sigma \end{pmatrix} \cdot \nabla f(0,0,0) + \frac{1}{2} \begin{pmatrix} x & y & \sigma \end{pmatrix} \mathrm{H}(0,0,0) \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

Set derivative to 0:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \nabla f(0,0,0) + \mathrm{H}(0,0,0) \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

Sub-Pixel Refinement

2nd order surface fitting (3D):

$$f(x,y,\sigma) \approx f(0,0,0) + \begin{pmatrix} x & y & \sigma \end{pmatrix} \cdot \nabla f(0,0,0) + \frac{1}{2} \begin{pmatrix} x & y & \sigma \end{pmatrix} \mathrm{H}(0,0,0) \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

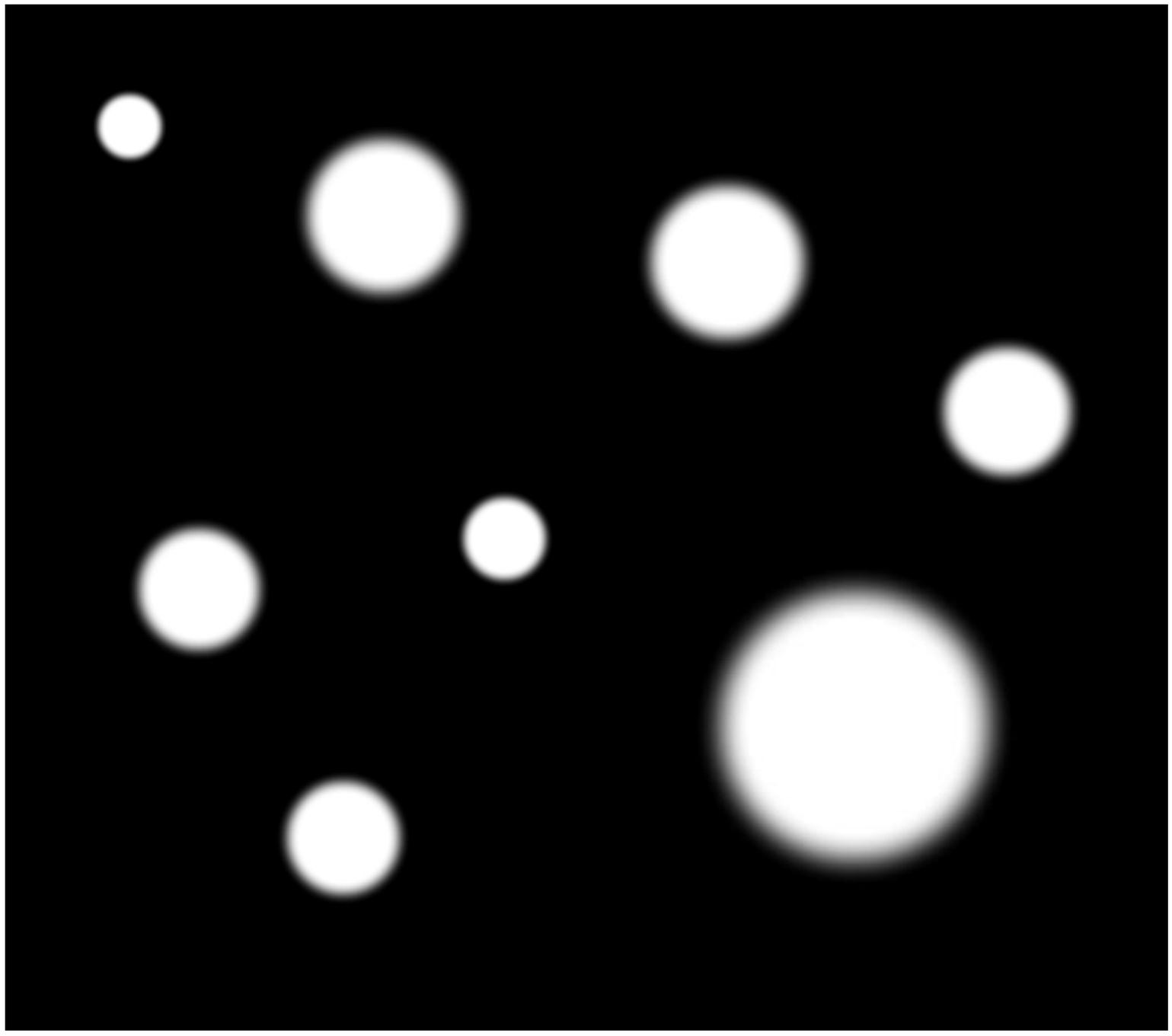
Set derivative to 0:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \nabla f(0,0,0) + \mathrm{H}(0,0,0) \begin{pmatrix} x \\ y \\ \sigma \end{pmatrix}$$

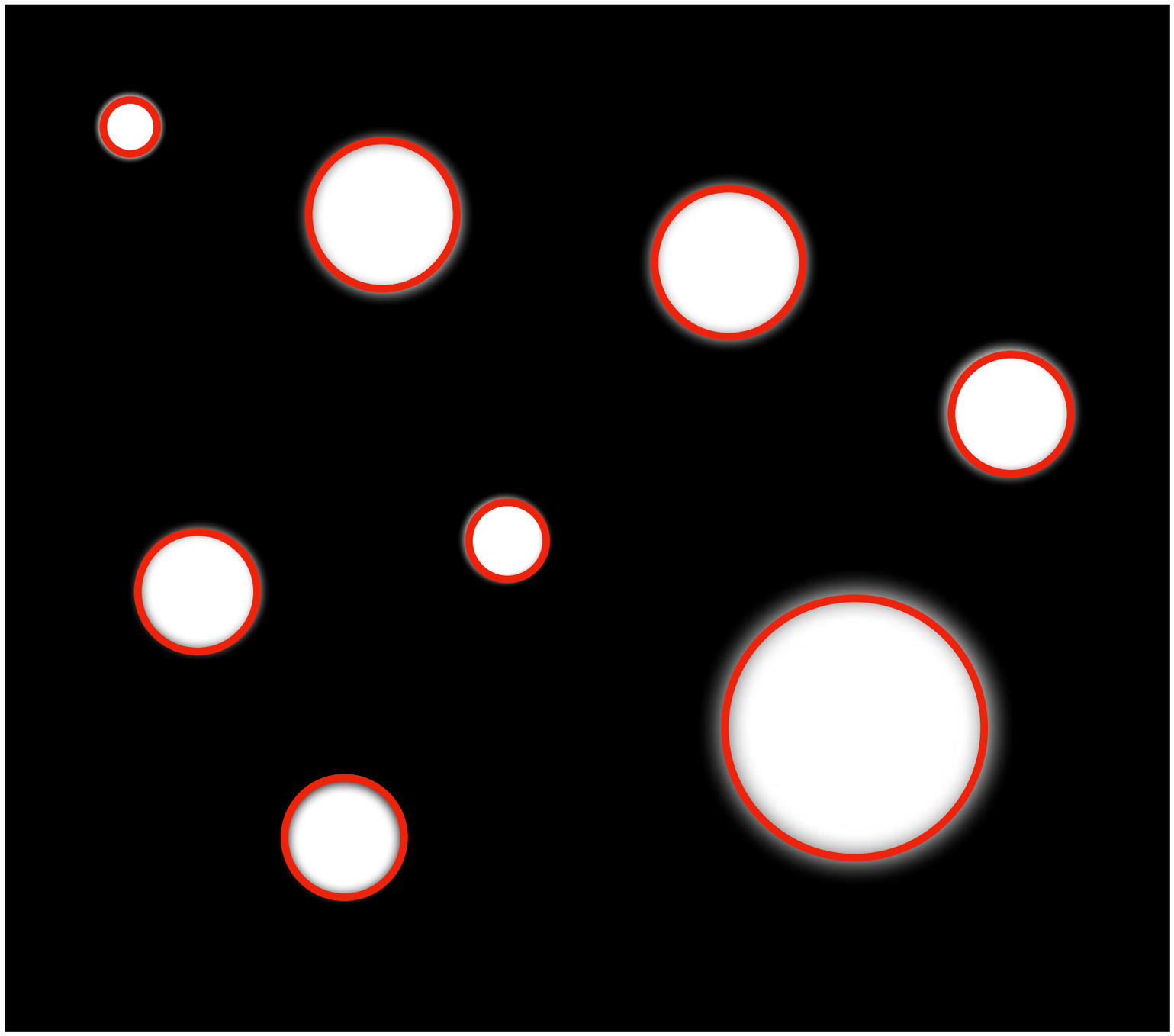
Solve small linear system:

$$\begin{pmatrix} x \\ y \\ \sigma \end{pmatrix} = -\mathrm{H}(0,0,0)^{-1} \nabla f(0,0,0)$$

Application



Application



Lessons Learned

- Main lessons from this lecture
  - Image filters: Gaussian, gradients (edge detection)
  - Similarity measures: ZNCC, histogram of gradients
  - Scale space: Gaussian pyramid by applying Gaussian Filter iteratively
  - Sub-pixel refinement: Fitting 2nd order surfaces
- Next lecture: Local features
  - Will use Gaussian filters, scale space representation, vector bouquets

Next Lecture

Jan. 20	Introduction, Linear classifiers and filtering	
Jan. 23	Filtering, gradients, scale	Lab 1
Jan. 27	**Local features**	
Jan. 30	Learning a classifier	Lab 2
Feb. 3	Convolutional neural networks	
Feb. 6	More convolutional neural networks	
Feb. 10	Robust model fitting and RANSAC	Lab 3
Feb. 13	Image registration	
Feb. 17	Camera Geometry	Lab 4
Feb. 20	More camera geometry	
Feb. 24	Generative neural networks	
Feb. 27	Generative neural networks	
Mar. 2	TBA	
Mar. 9	TBA	