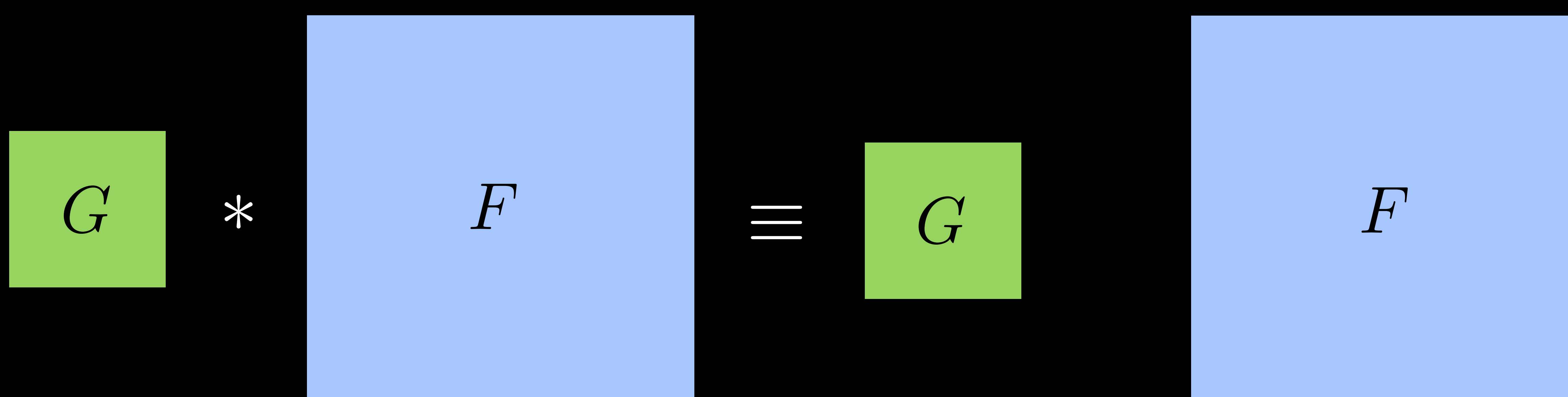
Intro to

Computer Vision

Image filtering

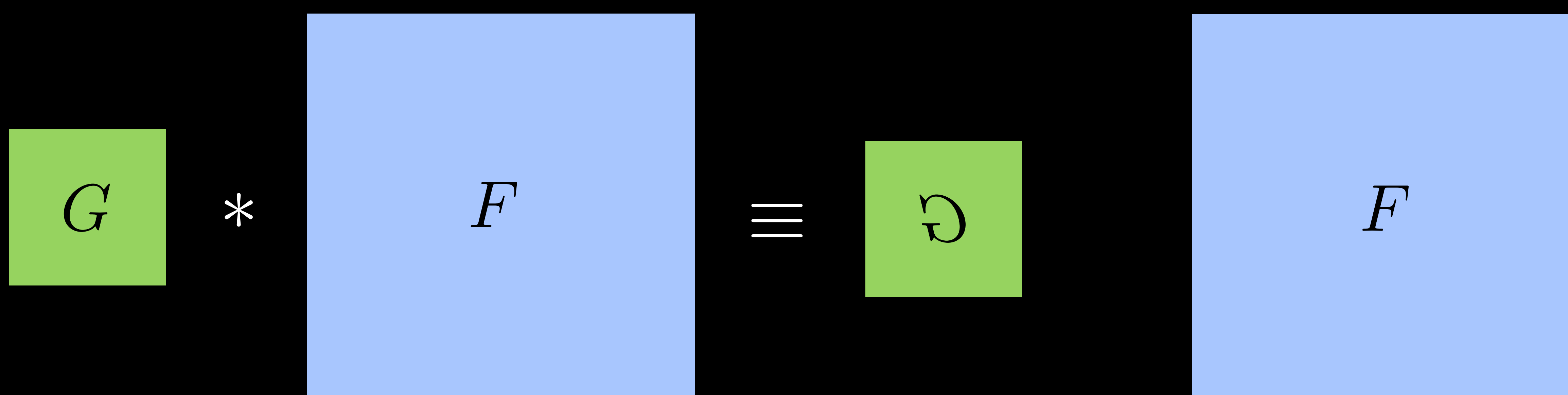
$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x-u, y-v]$$

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



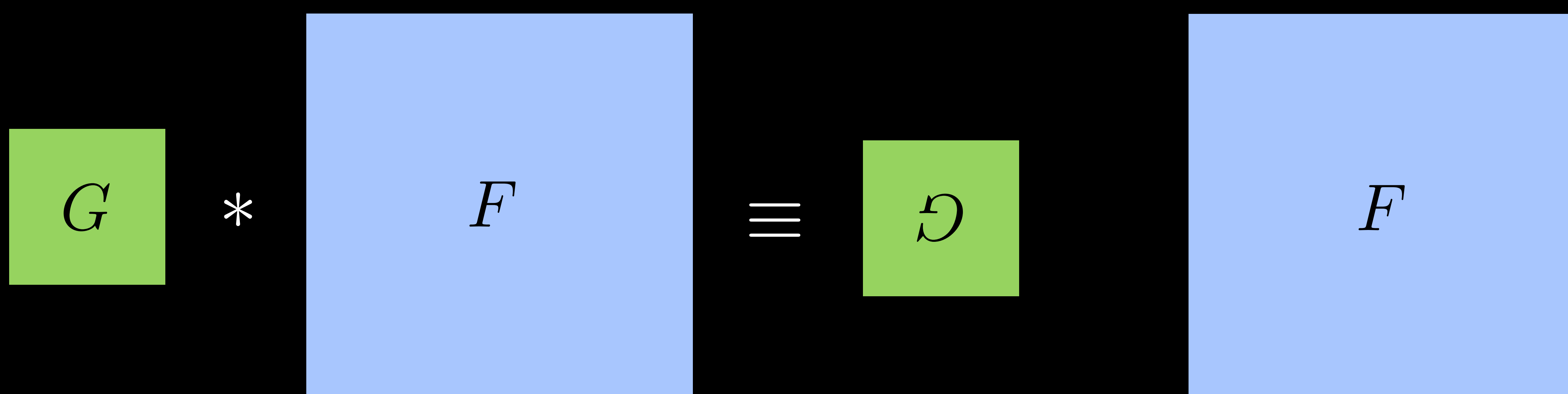
$$H[x, y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u, v] F[x - u, y - v]$$

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



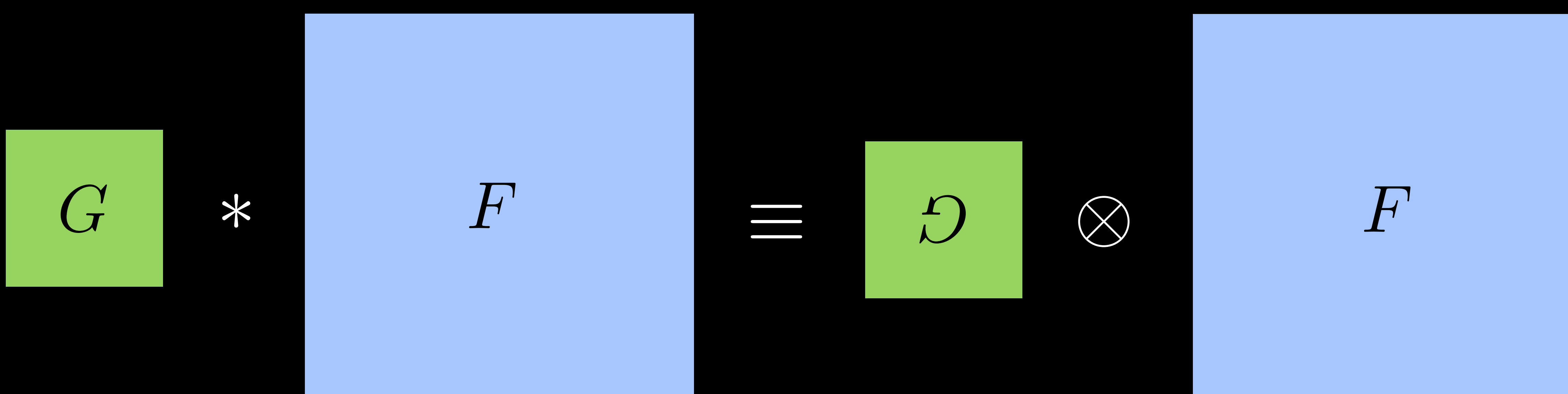
$$H[x, y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u, v] F[x-u, y-v]$$

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 $H = G * F$



Convolution Properties

identity

identity

$$F[x, y] * \delta[x, y]$$

identity

$$F[x, y] * \delta[x, y] = F[x, y]$$

linearity

$$G * (\alpha F_1[x, y] + \beta F_2[x, y])$$

linearity

$$G * (\alpha F_1[x,y] + \beta F_2[x,y])$$
$$= \alpha H_1[x,y] + \beta H_2[x,y]$$

shift invariant

$$G[x,y] * F[x-\alpha, y-\beta]$$

shift invariant

$$G[x, y] * F[x - \alpha, y - \beta] \\ = H[x - \alpha, y - \beta]$$

distributive

distributive

$$E[x,y] * (F[x,y] + G[x,y])$$

distributive

$$E[x,y] * (F[x,y] + G[x,y])$$
$$= (E[x,y] * F[x,y]) + (E[x,y] * G[x,y])$$

associative

$$(E[x,y] * F[x,y]) * G[x,y]$$

associative

$$(E[x,y] * F[x,y]) * G[x,y] \\ = E[x,y] * (F[x,y] * G[x,y])$$

associative

$$(E[x,y] * F[x,y]) * G[x,y]$$
$$= E[x,y] * (F[x,y] * G[x,y])$$

sequential filtering is equivalent to applying one filter

$$G(\mathbf{x}, \sigma_1) * G(\mathbf{x}, \sigma_2) * I(\mathbf{x})$$

$$G(\mathbf{x}, \sigma_1) * G(\mathbf{x}, \sigma_2)$$

$$G(\mathbf{x}, \sqrt{\sigma_1^2 + \sigma_2^2}) = G(\mathbf{x}, \sigma_1) * G(\mathbf{x}, \sigma_2)$$

$$G(\mathbf{x}, \sqrt{\sigma_1^2 + \sigma_2^2}) = G(\mathbf{x}, \sigma_1) * G(\mathbf{x}, \sigma_2)$$

convolution of Gaussians is a Gaussian

associative

$$(E[x,y] * F[x,y]) * G[x,y]$$
$$= E[x,y] * (F[x,y] * G[x,y])$$

sequential filtering is equivalent to applying one filter

commutative

$$F[x,y] * G[x,y]$$

commutative

$$F[x, y] * G[x, y] = G[x, y] * F[x, y]$$

commutativity
proof

commutativity proof

By definition

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

commutativity proof

By definition

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

Let $p = x - u$ and $q = y - v$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p, y-q] F[p, q]$$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p, y-q]F[p,q]$$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p,y-q]F[p,q]$$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p,y-q]F[p,q]$$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p, y-q]F[p,q]$$

rearrange

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p, y-q]F[p, q]$$

rearrange

$$= \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} F[p, q]H[x-p, y-q]$$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p, y-q]F[p,q]$$

rearrange

$$= \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} F[p,q]H[x-p, y-q]$$

$$= F * H$$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p,y-q]F[p,q]$$

rearrange

$$= \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} F[p,q]H[x-p,y-q]$$

$$= F * H$$

commutativity proof

By definition

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

Let $p = x - u$ **and** $q = y - v$

substitute

$$H * F = \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} H[x-p,y-q]F[p,q]$$

rearrange

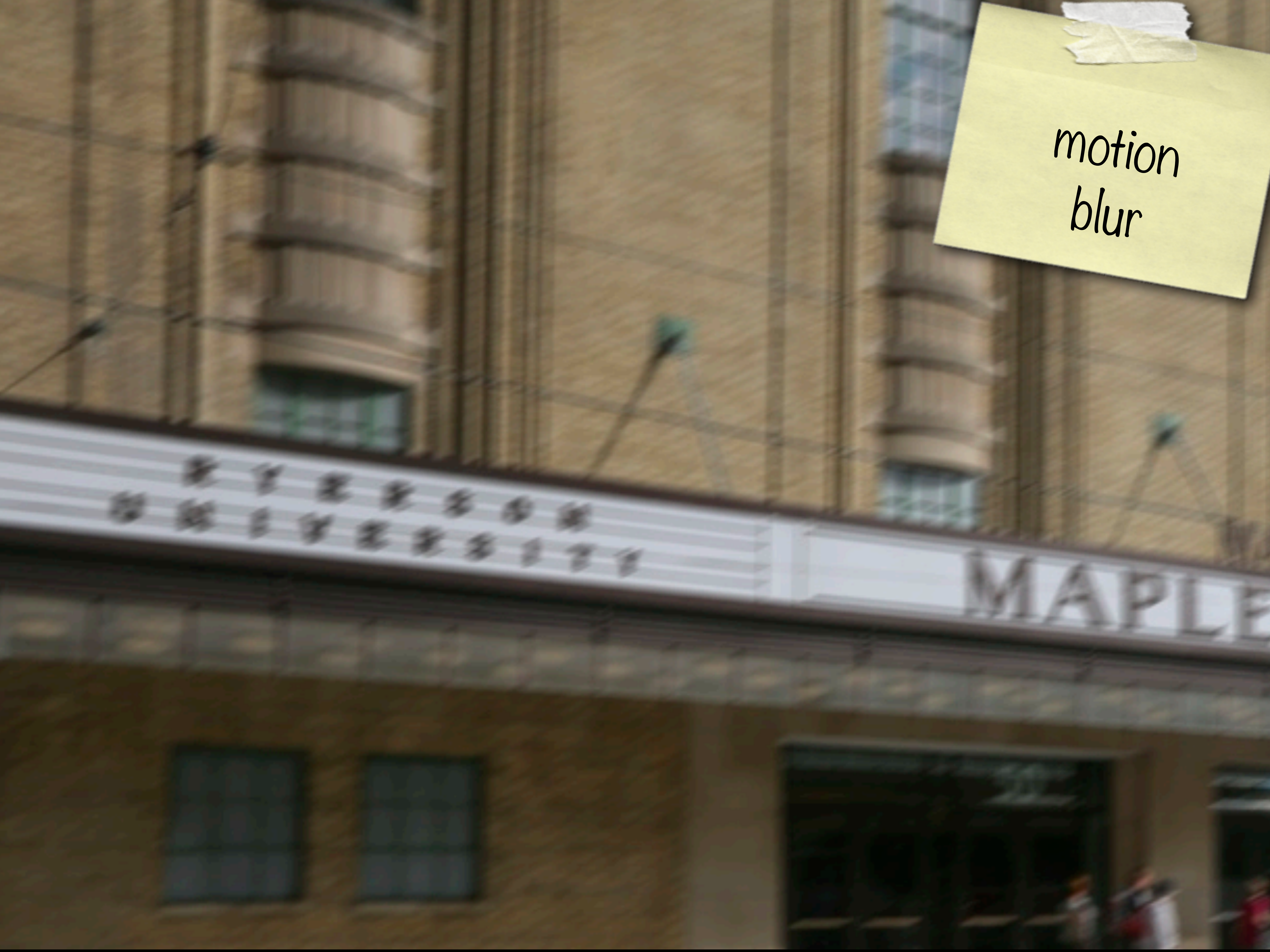
$$= \sum_{q=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} F[p,q]H[x-p,y-q]$$

$$= F * H$$


RYERSON
UNIVERSITY
MATTAMY ATHLETIC CENTRE
MAPLE LEAF GARDENS
50

OFF
ON
RAW
JPEG
START
STOP
AF-ON
MENU
INFO.
SET
LOCK
Canon
MATTAMY ATHLETIC CENTRE
PLE LEAF GARDENS
RYERSON
UNIVERSITY
MAPLE

MATTAMY ATHLETIC CENTRE
PLE LEAF GARDENS
RYERSON UNIVERSITY
MAPLE
OFF
ON
AF-ON
START STOP
RAW JPEG
MENU
INFO.
SET
LOCK
Canon


motion
blur

motion
blur

blurred image

motion blur

=

blurred image

motion
blur

=

blurred image **ideal image**

motion blur

blurred image

=

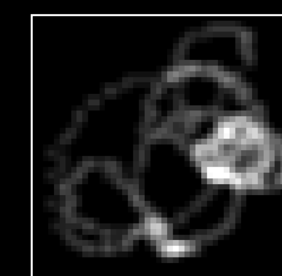
ideal image

*

motion blur

=

*



blurred image **ideal image** **kernel**

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

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

How many operations does convolution require per pixel?

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

How many operations does convolution require per pixel?

$$\boxed{(2K+1)^2}$$

separable filters

separable filters

$$F[x, y] = U[x]V[y]$$

separable filters

$$F[x,y] = U[x]V[y]$$

Leads to FASTER convolution

Assume the filter can be written as

$$F[x, y] = U[x]V[y]$$

Assume the filter can be written as

$$F[x, y] = U[x]V[y]$$

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v] \left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u] \right)$$

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v]\left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u]\right)$$

Assume the filter can be written as

$$F[x, y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u, v]U[x - u]V[y - v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y - v] \left(\sum_{u=-\infty}^{\infty} H[u, v]U[x - u] \right)$$

Does this look familiar?

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v] \left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u] \right)$$

1D horizontal convolution

Assume the filter can be written as

$$F[x, y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u, v]U[x - u]V[y - v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y - v] \left(\sum_{u=-\infty}^{\infty} H[u, v]U[x - u] \right)$$

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v] \left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u] \right)$$

Does this look familiar?

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v] \left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u] \right)$$

1D vertical convolution

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v] \left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u] \right)$$

Filter horizontally then vertically

Assume the filter can be written as

$$F[x, y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u, v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v]\left(\sum_{u=-\infty}^{\infty} H[u, v]U[x-u]\right)$$

How many operations does convolution require per pixel?

Assume the filter can be written as

$$F[x,y] = U[x]V[y]$$

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

insert filter definition

$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

factor out

$$= \sum_{v=-\infty}^{\infty} V[y-v] \left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u] \right)$$

How many operations does convolution require per pixel?

$$\boxed{2(2K+1)}$$

convolution

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

How many operations does convolution require per pixel?

$$\boxed{(2K+1)^2}$$

Assume the filter can be written as:

$$F[x,y] = U[x]V[y]$$

separable filtering

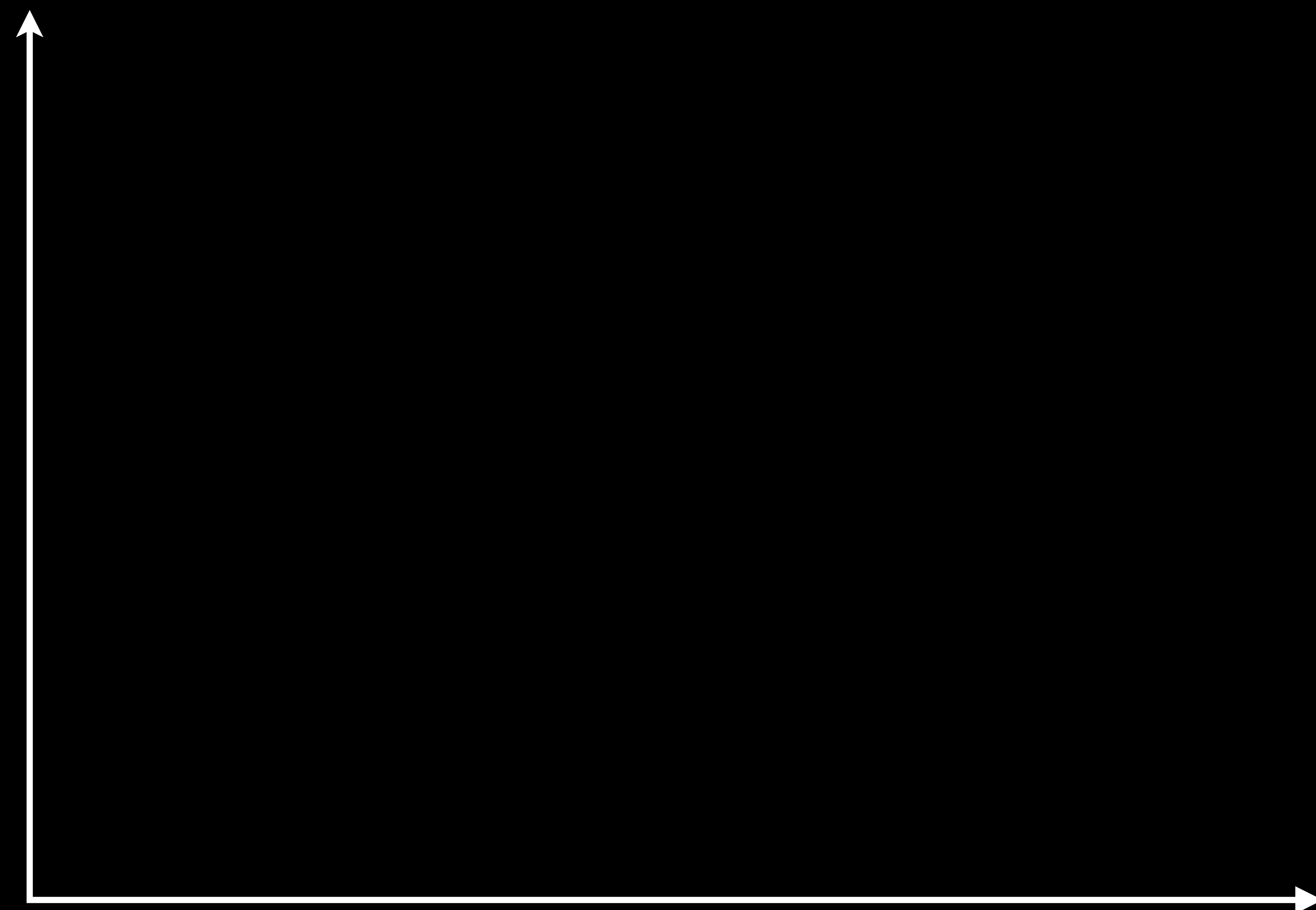
$$H * F = \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]F[x-u, y-v]$$

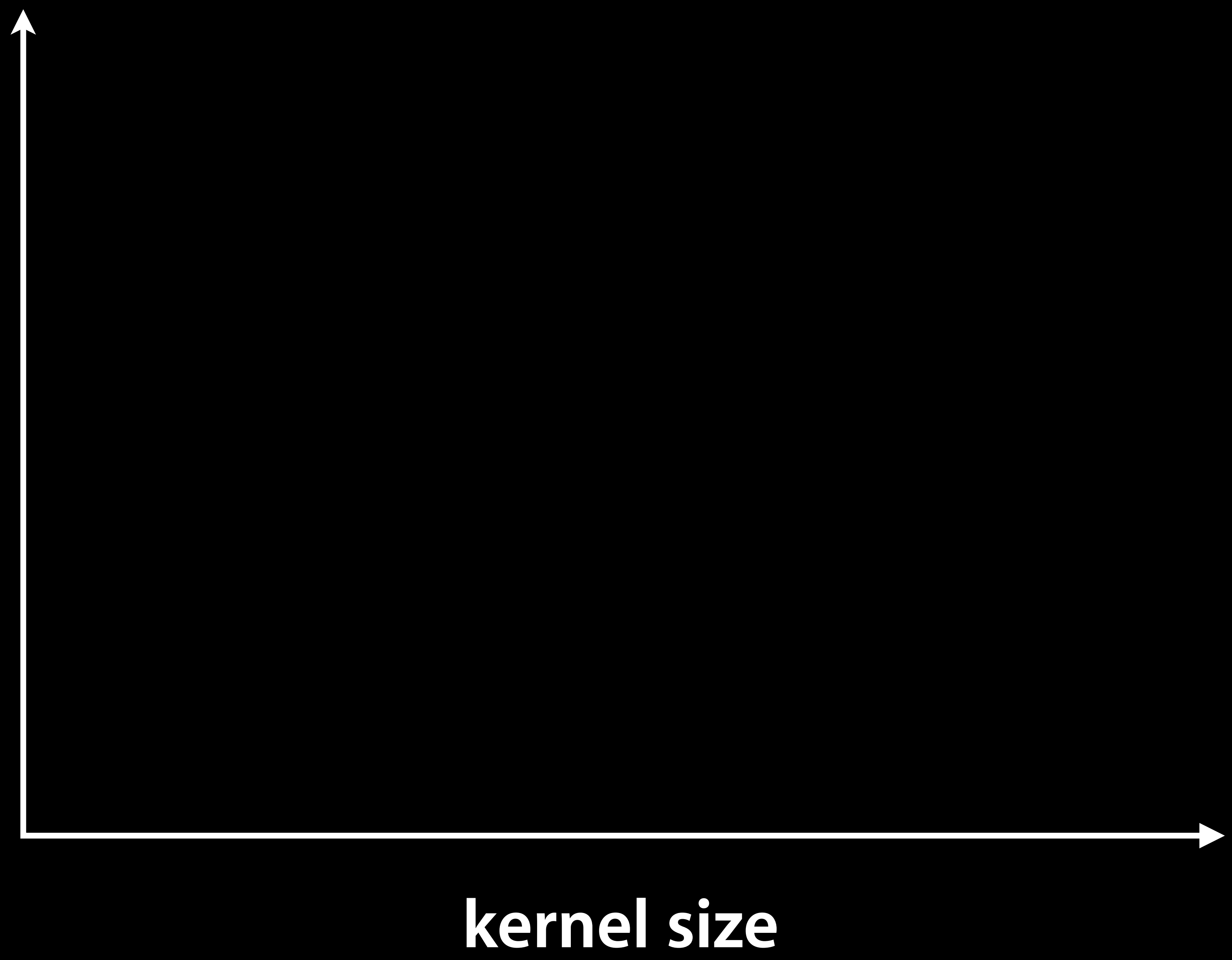
$$= \sum_{v=-\infty}^{\infty} \sum_{u=-\infty}^{\infty} H[u,v]U[x-u]V[y-v]$$

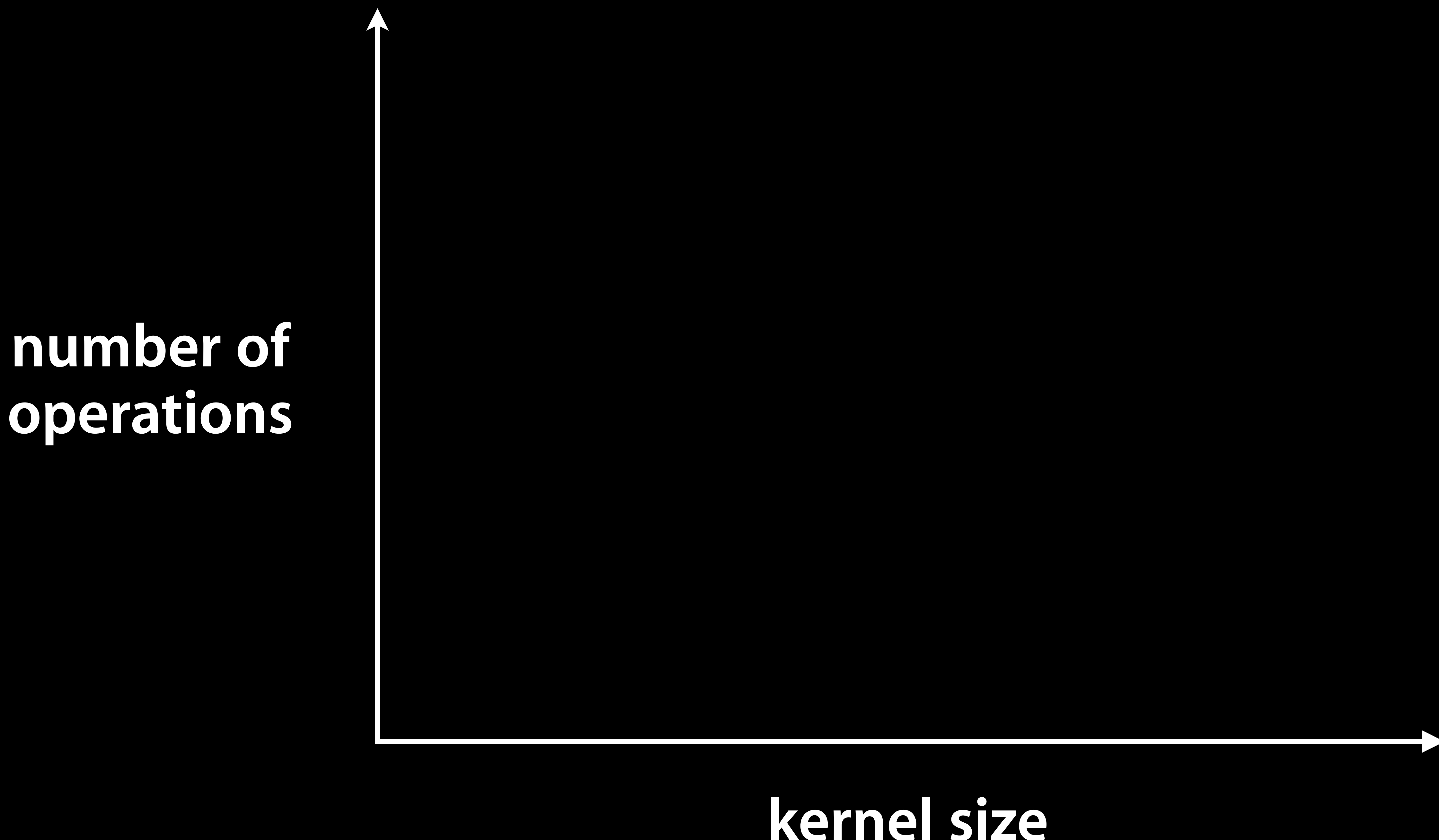
$$= \sum_{v=-\infty}^{\infty} V[y-v] \left(\sum_{u=-\infty}^{\infty} H[u,v]U[x-u] \right)$$

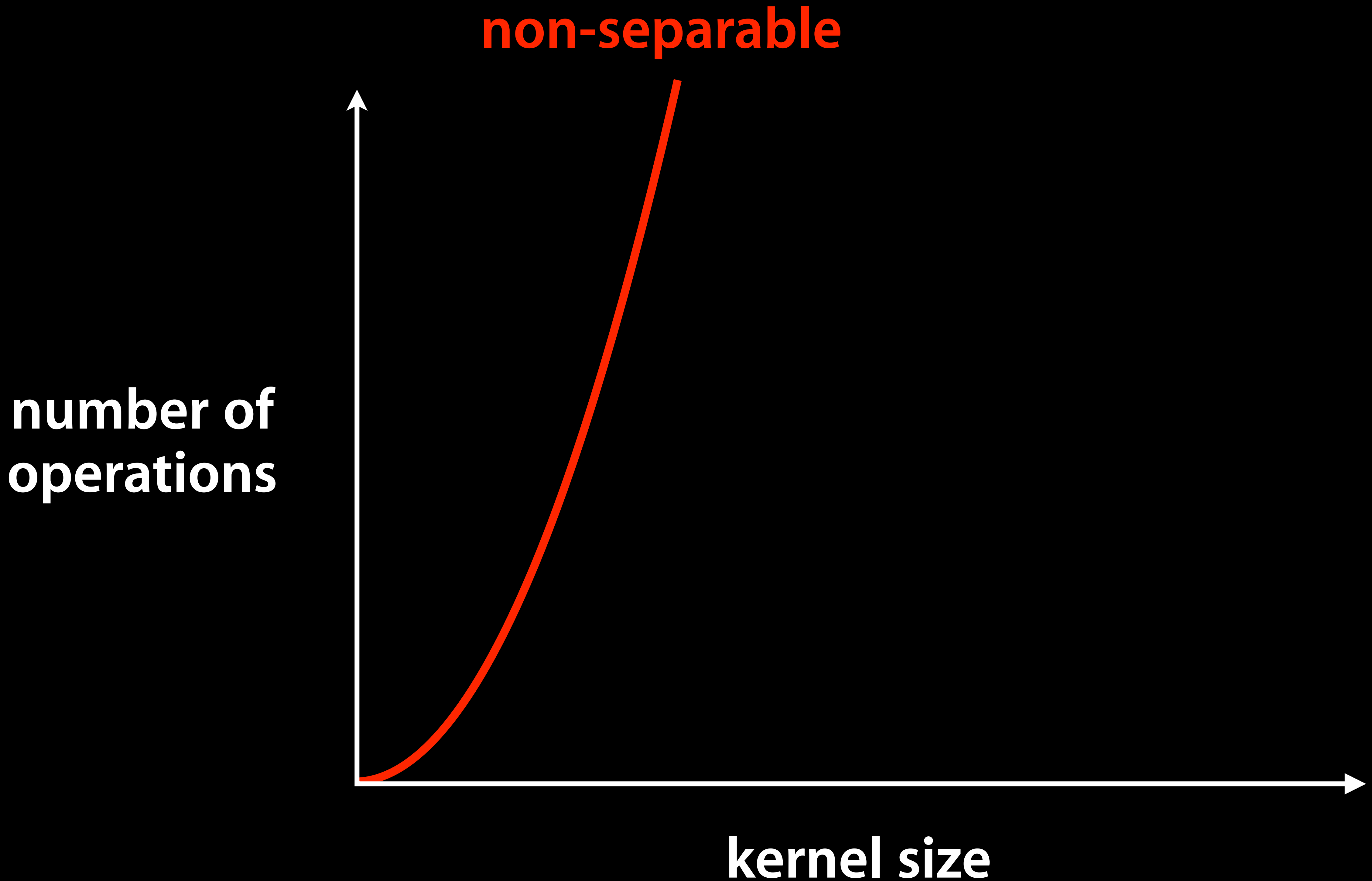
How many operations does convolution require per pixel?

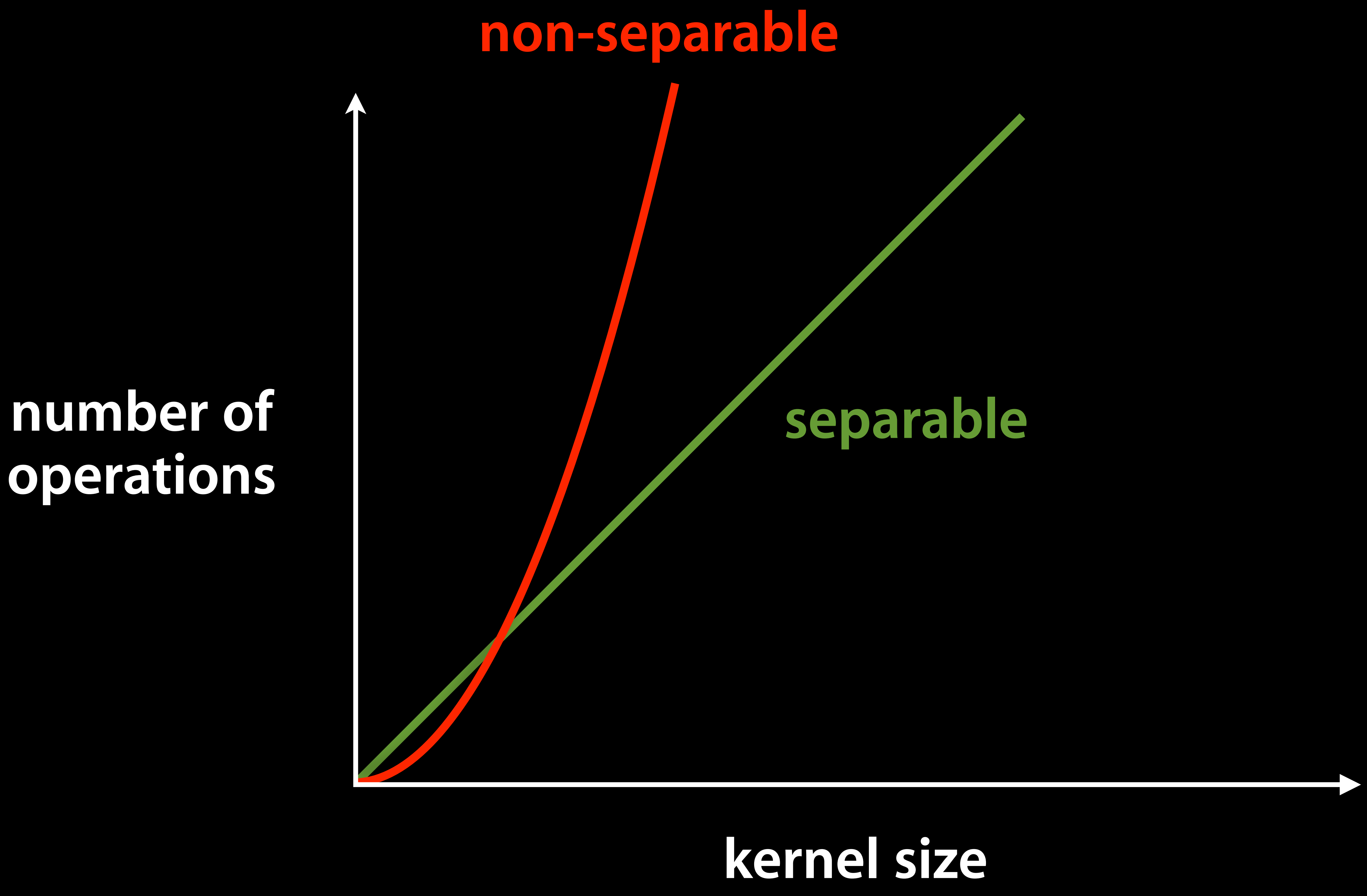
$$\boxed{2(2K+1)}$$

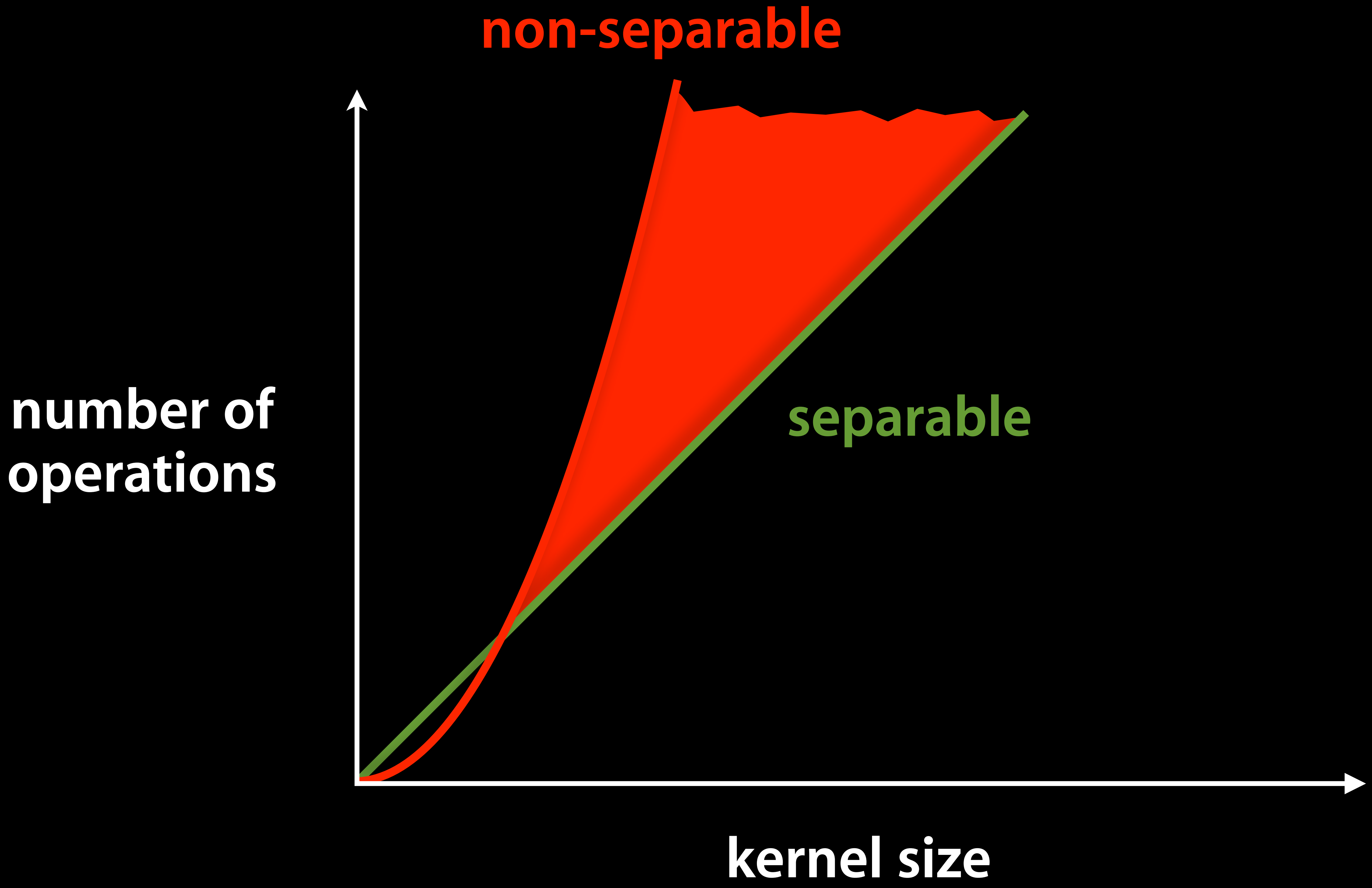



kernel size


number of
operations
kernel size


non-separable
number of operations
kernel size


non-separable
separable
number of
operations
kernel size


non-separable
separable
number of operations
kernel size

Show that the 2D Gaussian filter is separable.

Show that the 2D Gaussian filter is separable.

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

Show that the 2D Gaussian filter is separable.

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

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

Show that the 2D Gaussian filter is separable.

$$
\begin{aligned}
G(x, y) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \\
&= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp\left(-\frac{y^2}{2\sigma^2}\right) \\
&= \left[\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)\right] \left[\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{y^2}{2\sigma^2}\right)\right]
\end{aligned}
$$

Show that the 2D Gaussian filter is separable.

$$
\begin{aligned}
G(x,y) &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \\
&= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\exp\left(-\frac{y^2}{2\sigma^2}\right) \\
&= \left[\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma^2}\right)\right]\left[\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{y^2}{2\sigma^2}\right)\right] \\
&= G_1(x)G_1(y)
\end{aligned}
$$

review

review

Correlation $H = G \otimes F$

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

review

Correlation $H = G \otimes F$

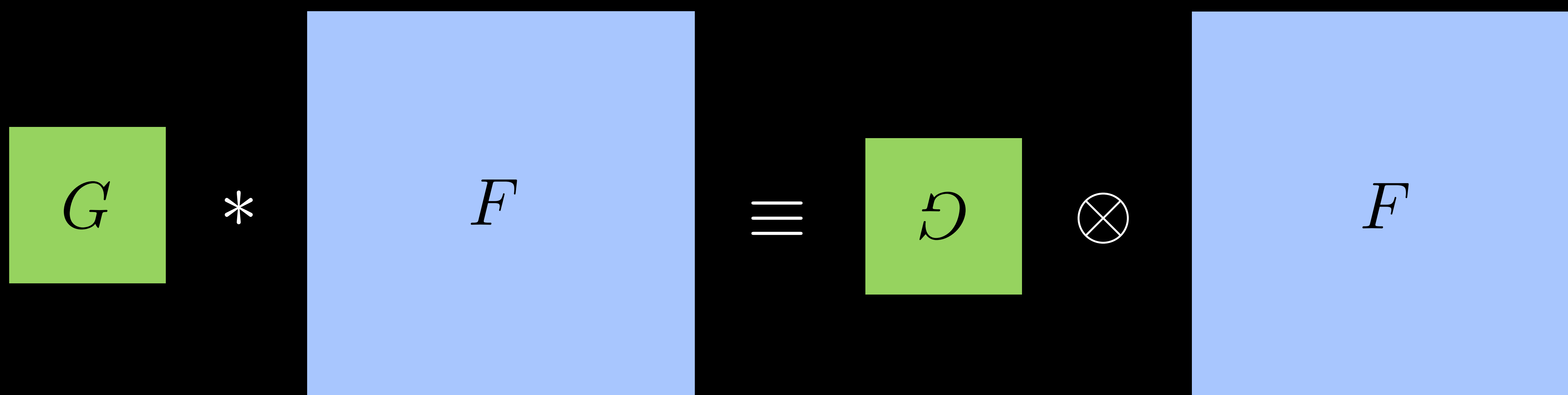
$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x+u, y+v]$$

Convolution $H = G * F$

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

Correlation $H = G \otimes F$

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



Convolution $H = G * F$

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

Correlation $H = G \otimes F$

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

review

For a symmetric filter how will the outputs differ?

Convolution $H = G * F$

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

review

Correlation $H = G \otimes F$

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

For a symmetric filter how will the outputs differ?

no difference

Convolution $H = G * F$

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

review

Correlation $H = G \otimes F$

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

For an impulse signal, how will the outputs differ?

Convolution $H = G * F$

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

review

Correlation $H = G \otimes F$

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

For an impulse signal, how will the outputs differ?

output flipped

Convolution $H = G * F$

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


Practical
use case

image sharpening

input

image sharpening

input

image sharpening

input

blurred

image sharpening

input

blurred

image sharpening


input

−


blurred

=

image sharpening

input

$-$

blurred

$=$

"sharp-stuff"

image sharpening

input

=

blurred

+

“sharp-stuff”

image sharpening

blurred

+

“sharp-stuff”

image sharpening

blurred

$+$

"sharp-stuff"

image sharpening

blurred

$+$

“sharp-stuff”

$=$

sharpened

input

sharpened

Predict

the filter outputs

input

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

?

RYERSON UNIVERSITY

input

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

RYERSON UNIVERSITY

no change

RYERSON UNIVERSITY

input

$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \; ?$$

$$\text{input} * \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \text{output}$$

input

shift one pixel to the right

input

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

?

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

input

blurred

Integral Image

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Abstract

This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new image representation called the

tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can

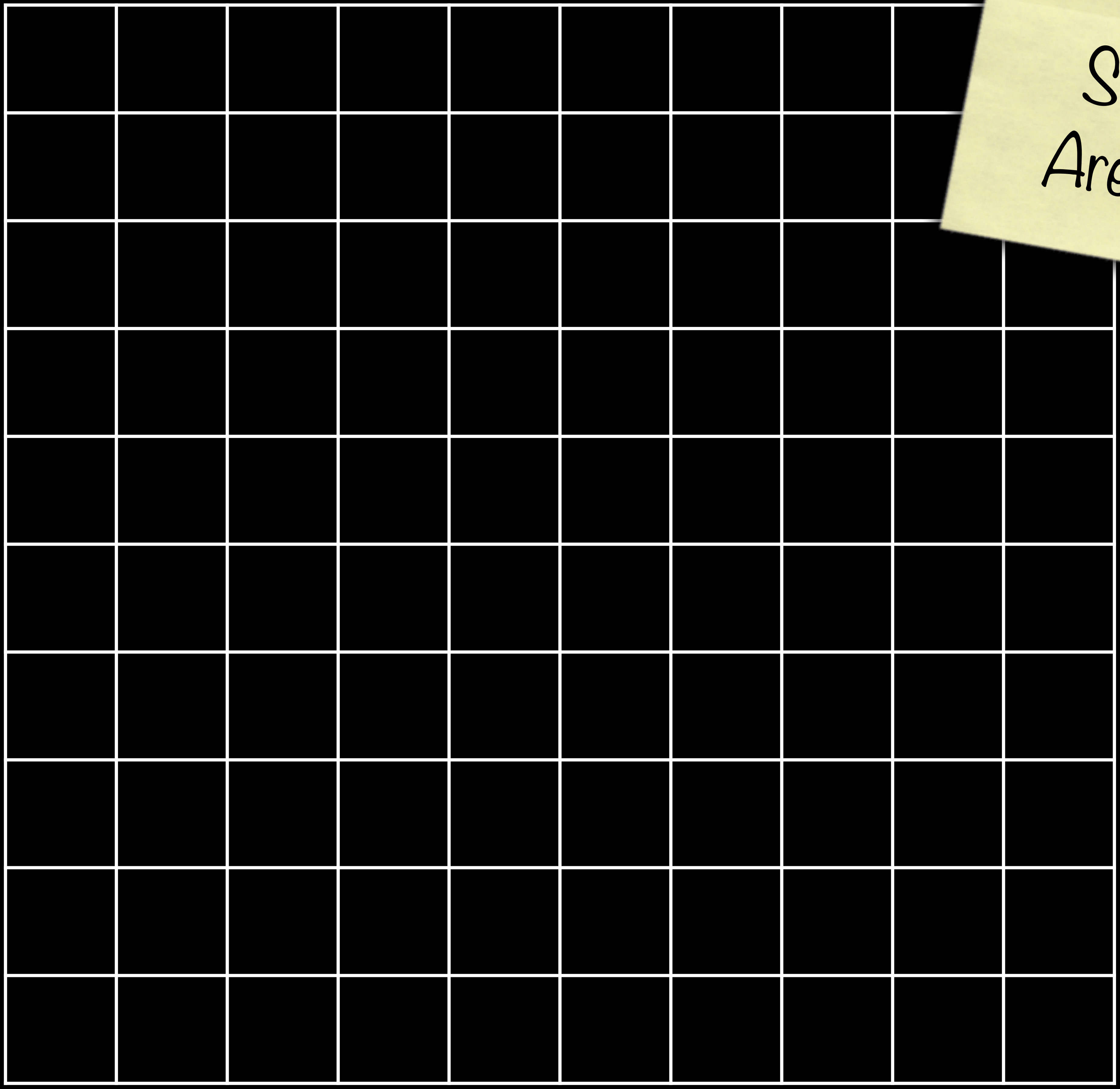
108	76	98	14	246	146	56	236	14	12
20	44	180	62	244	8	20	156	150	122
224	144	230	80	108	76	140	114	44	144
174	234	26	56	34	234	228	40	100	34
152	194	90	30	202	22	200	226	130	50
164	112	40	52	200	0	8	44	252	102
186	34	150	212	102	236	48	72	228	54
124	244	98	44	18	248	26	14	212	164
118	200	196	28	94	34	32	52	104	160
52	34	18	90	202	52	110	218	252	74

108	76	98	14	246	146	56	236	14	12
20	44	180	62	244	8	20	156	150	122
224	144	230	80	108	76	140	114	44	144
174	234	26	56	34	234	228	40	100	34
152	194	90	30	202	22	200	226	130	50
164	112	40	52	200	0	8	44	252	102
186	34	150	212	102	236	48	72	228	54
124	244	98	44	18	248	26	14	212	164
118	200	196	28	94	34	32	52	104	160
52	34	18	90	202	52	110	218	252	74

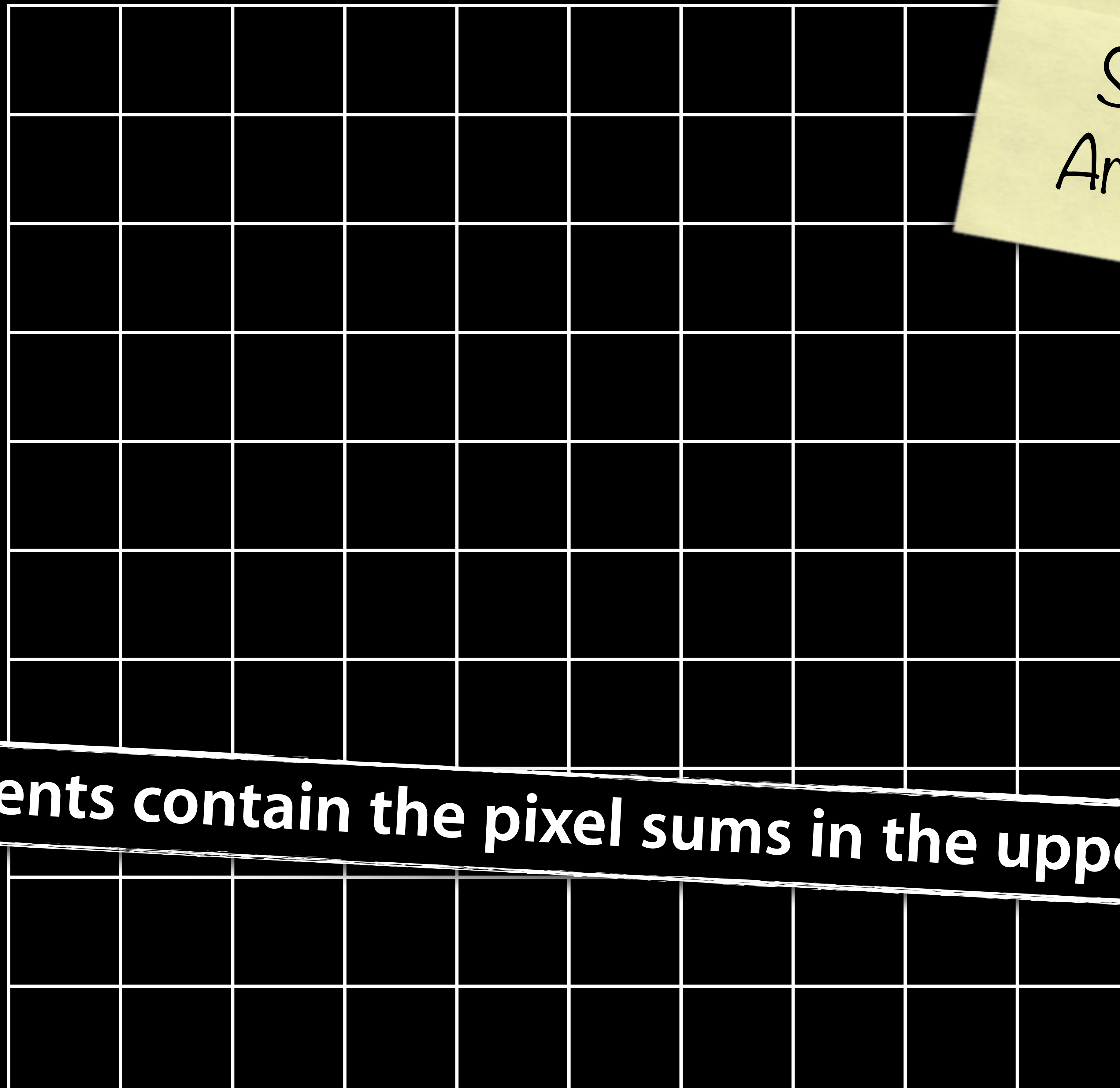
108	76	98	14	246	146	56	236	14	12
20	44	180	62	244	8	20	156	150	122
224	144	230	80	108	76	140	114	44	144
174	234	26	56	34	234	228	40	100	34
152	194	90	30	202	22	200	226	130	50
164	112	40	52	200	0	8	44	252	102
186	34	150	212	102	236	48	72	228	54
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
118	200	196	28	94	34	32	52	104	160
52	34	18	90	202	52	110	218	252	74

How can we compute the sum within the box?

Summed Area Table

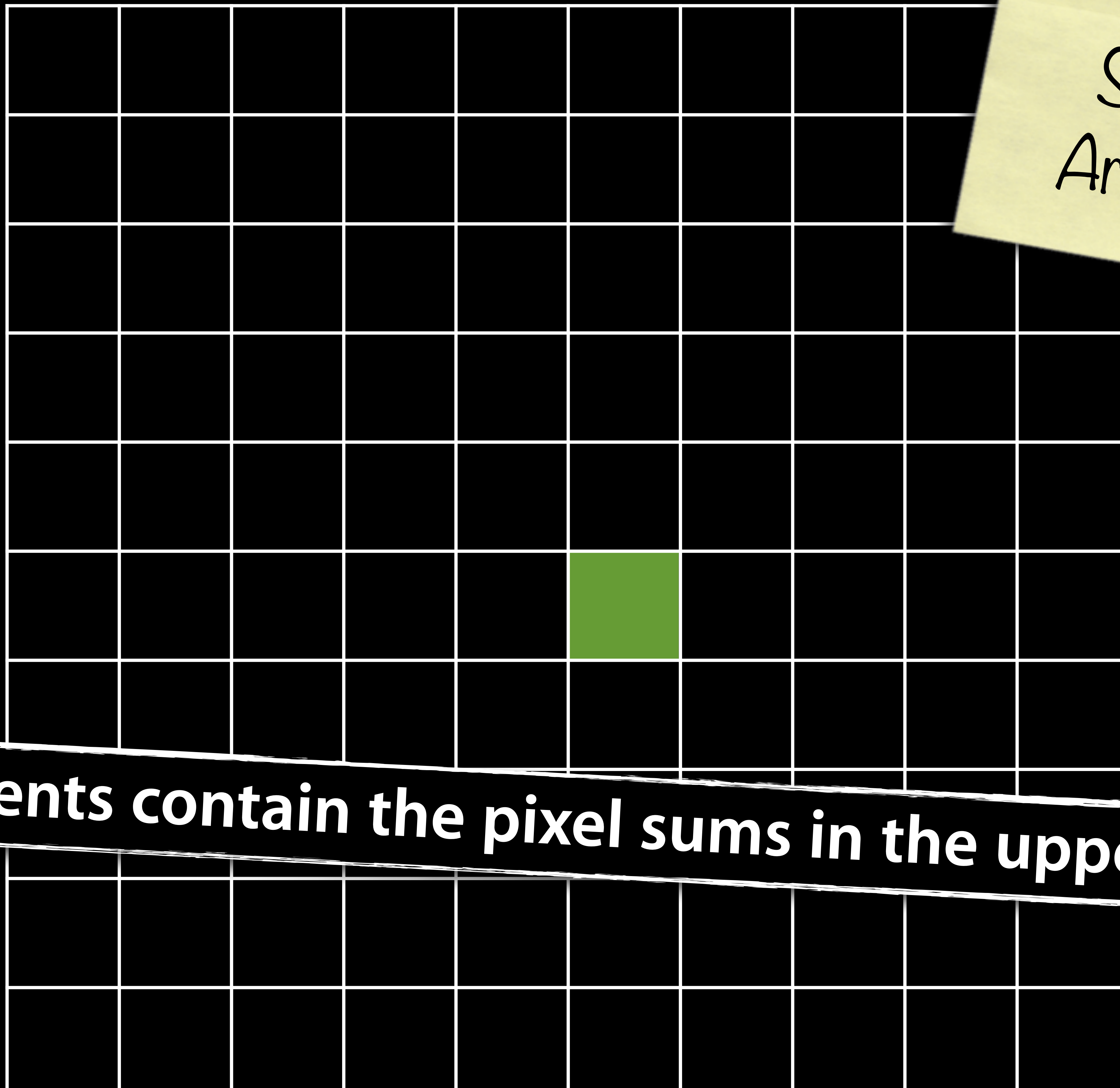


Summed Area Table

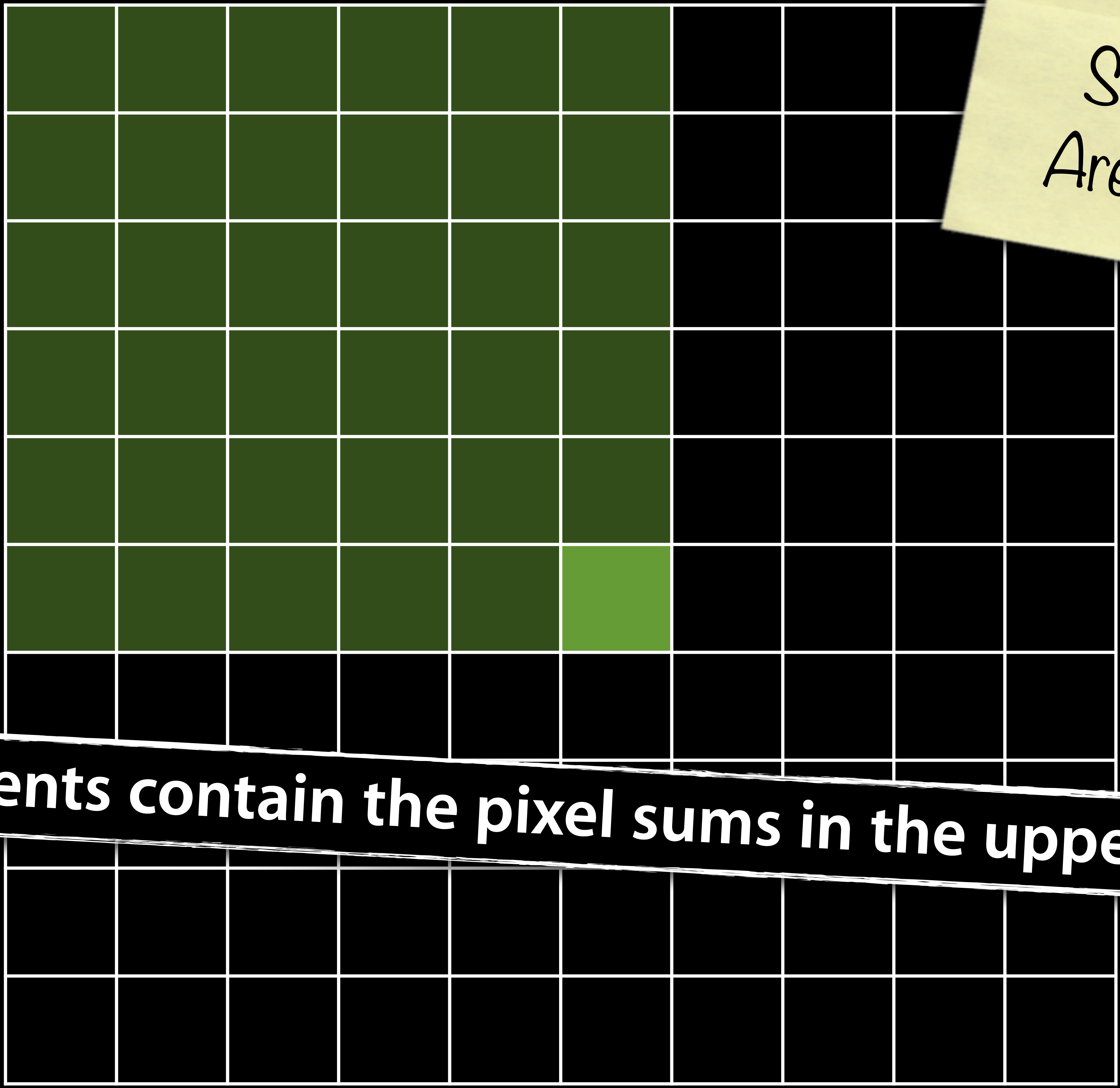


elements contain the pixel sums in the upper-left

Summed Area Table



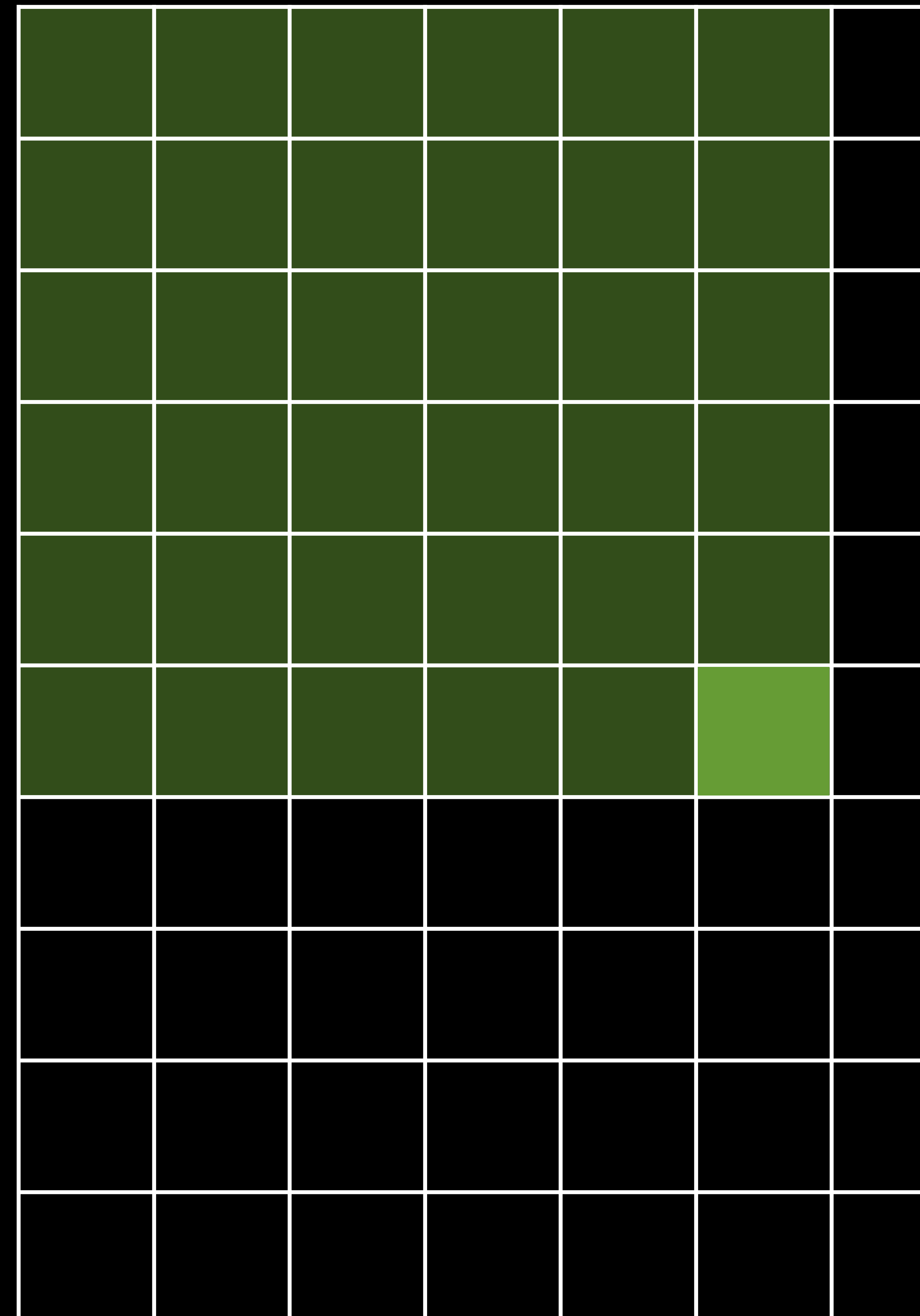
elements contain the pixel sums in the upper-left

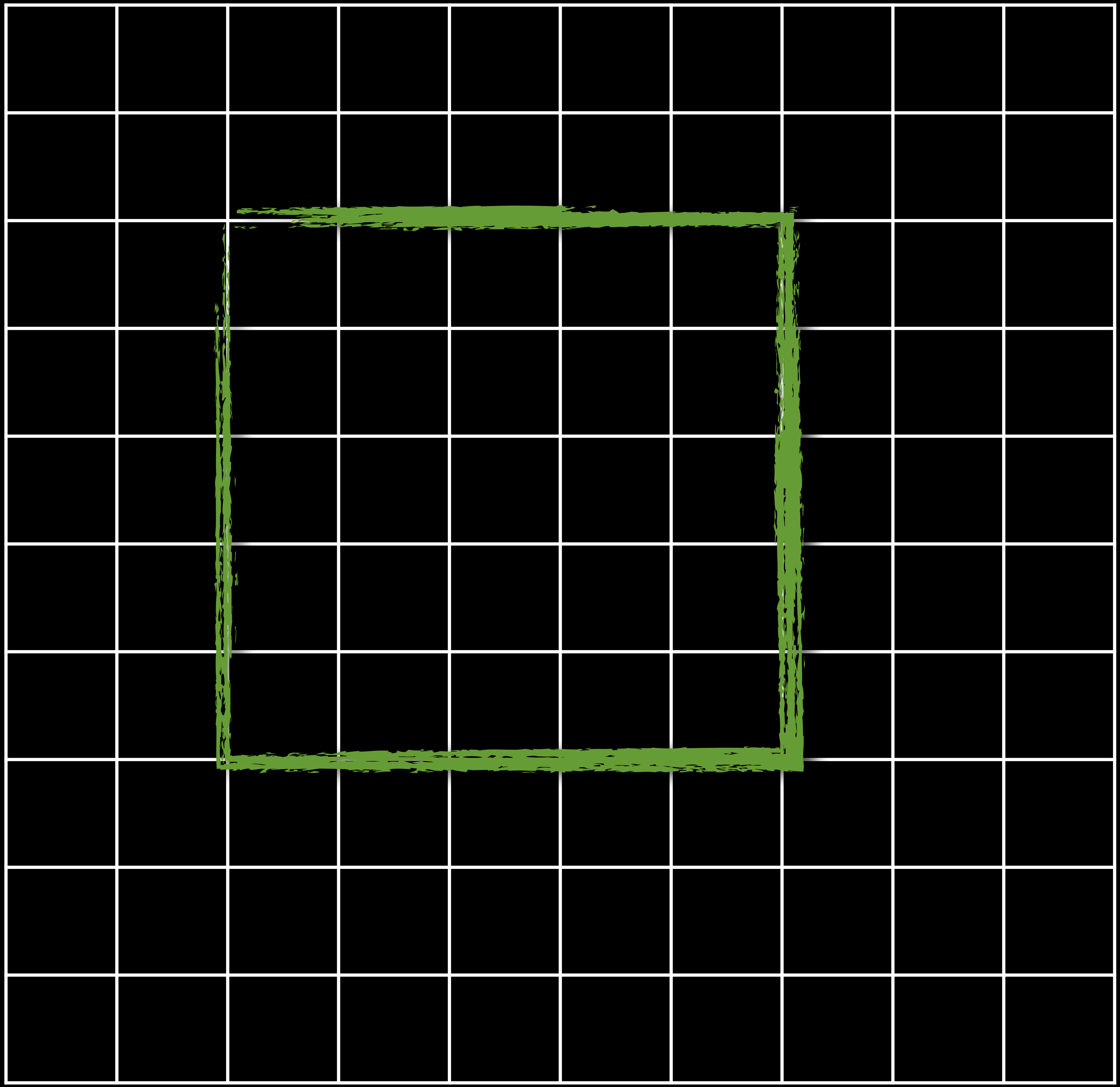


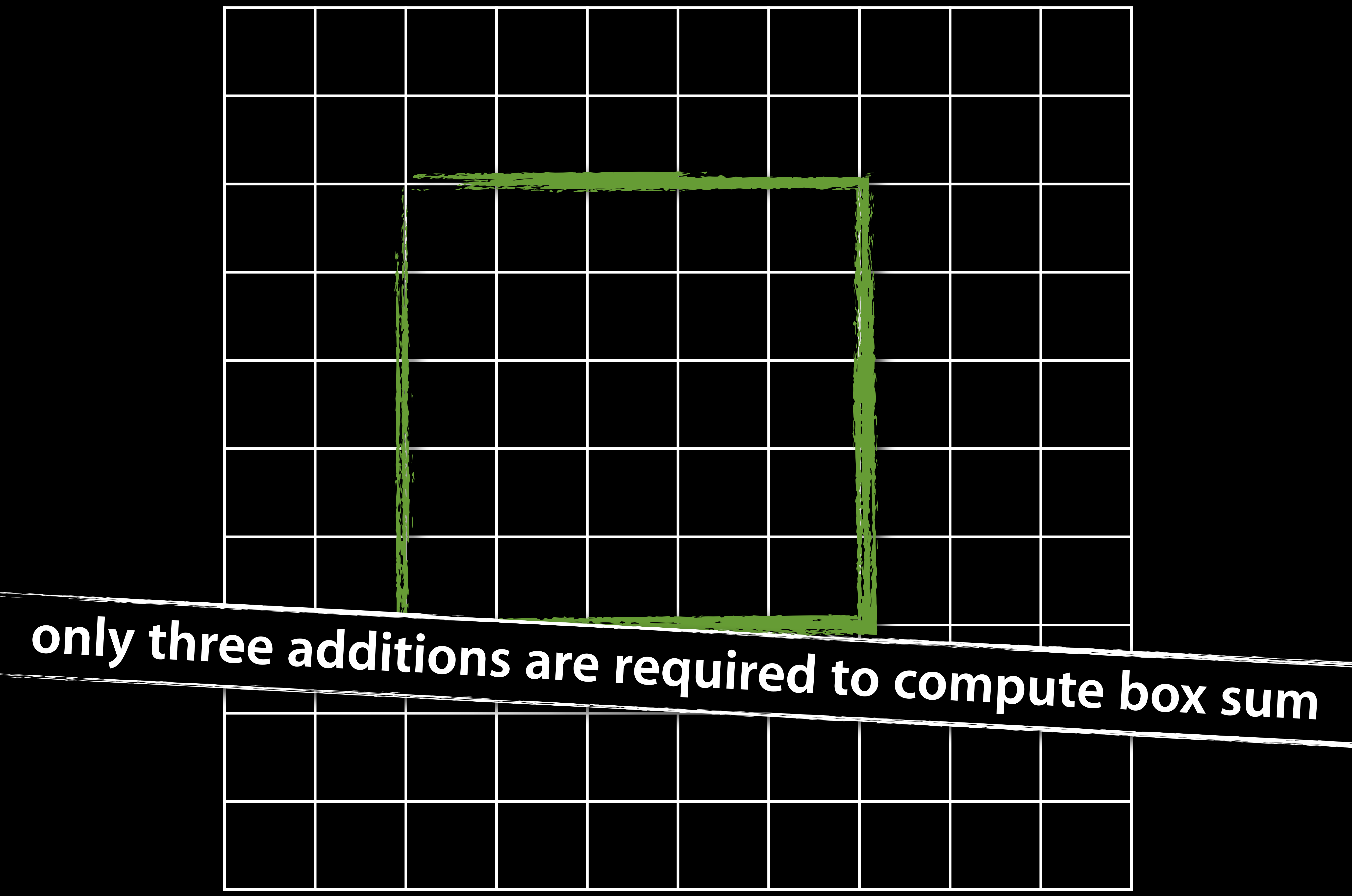
Summed Area Table

elements contain the pixel sums in the upper-left

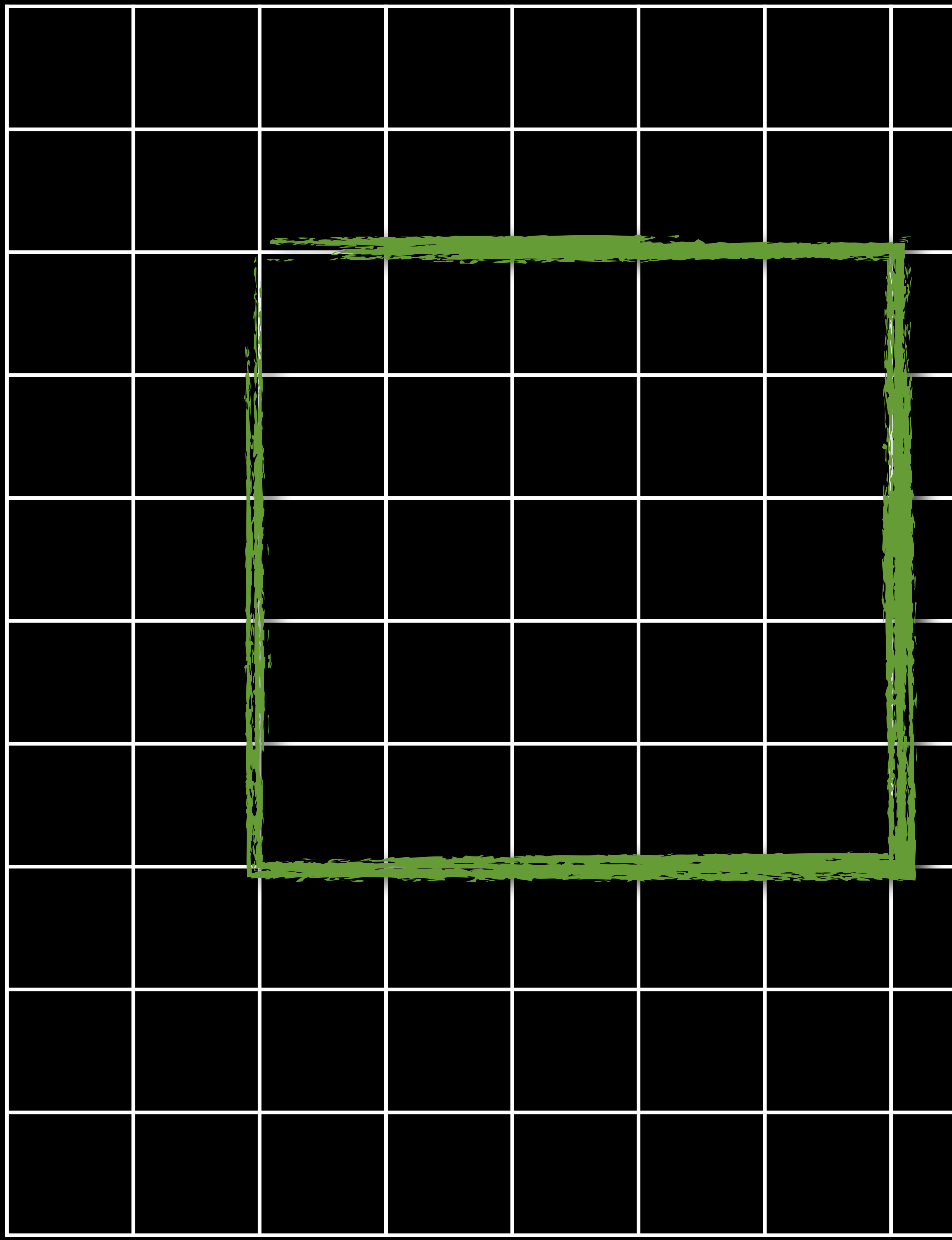
$$
\begin{aligned}
I_{\mathrm{S.A.T}}[x,y] = \\
&I[x,y]+ \\
&I_{\mathrm{S.A.T.}}[x-1,y]+ \\
&I_{\mathrm{S.A.T.}}[x,y-1]- \\
&I_{\mathrm{S.A.T.}}[x-1,y-1]
\end{aligned}
$$



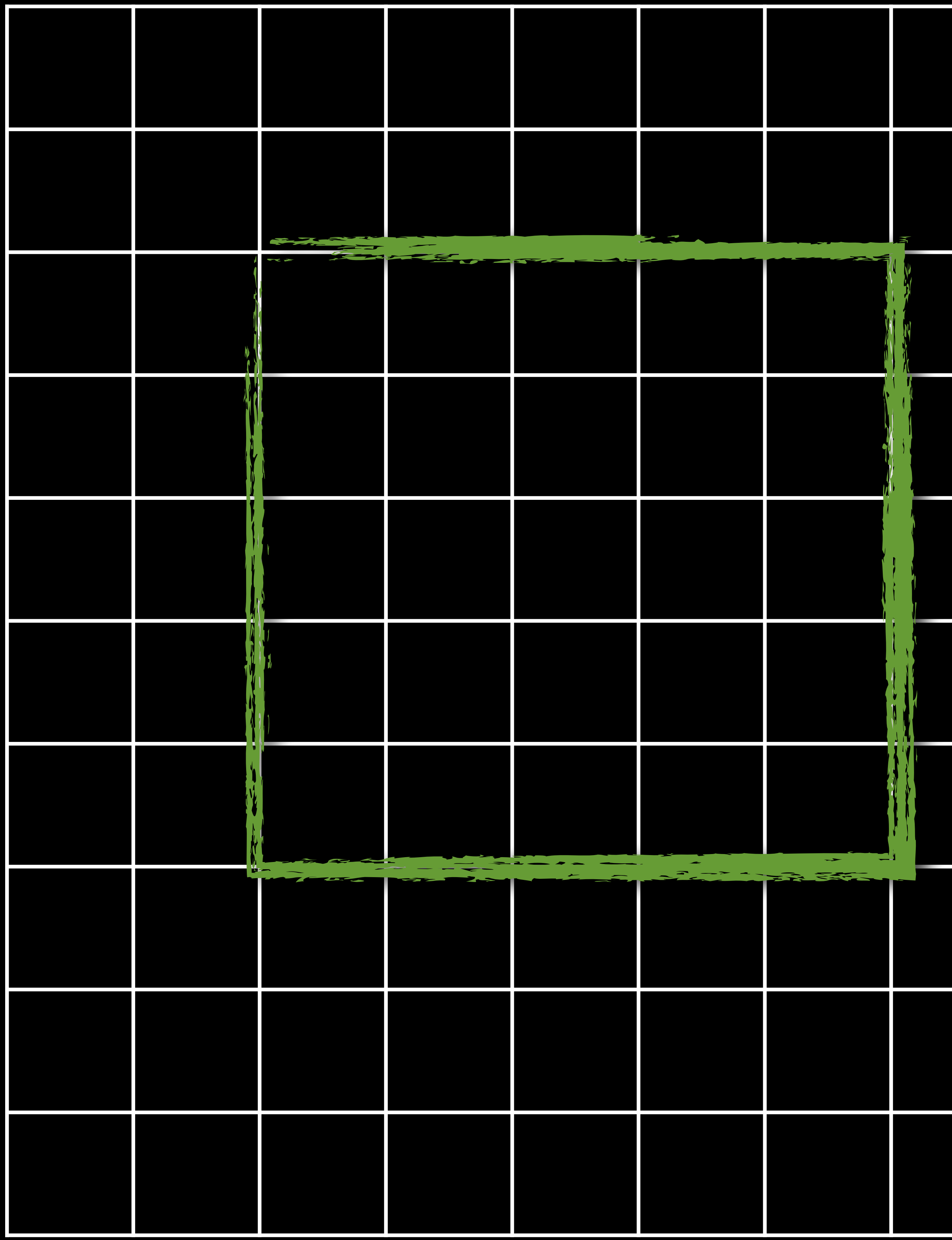



only three additions are required to compute box sum

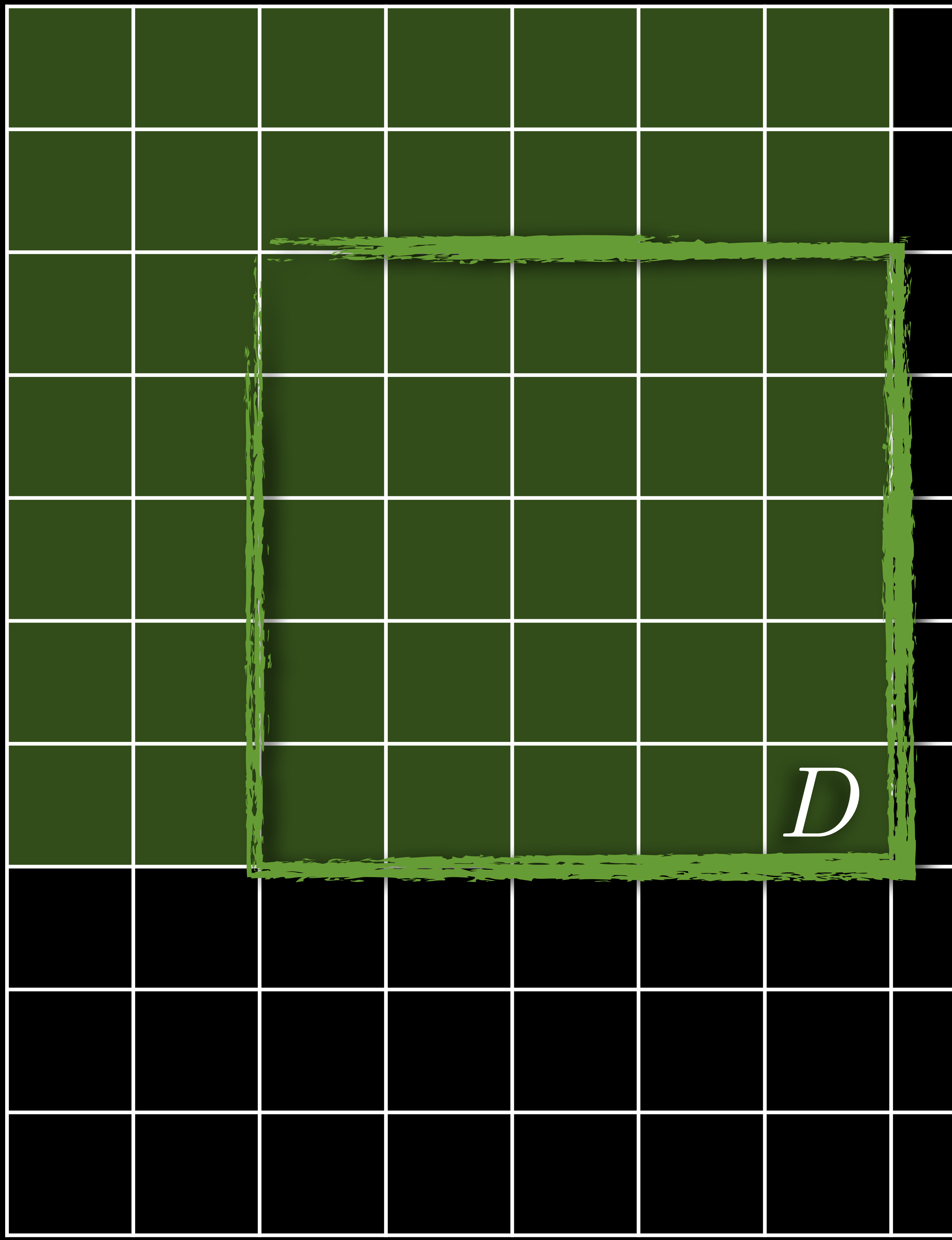
$$D - B - C + A$$



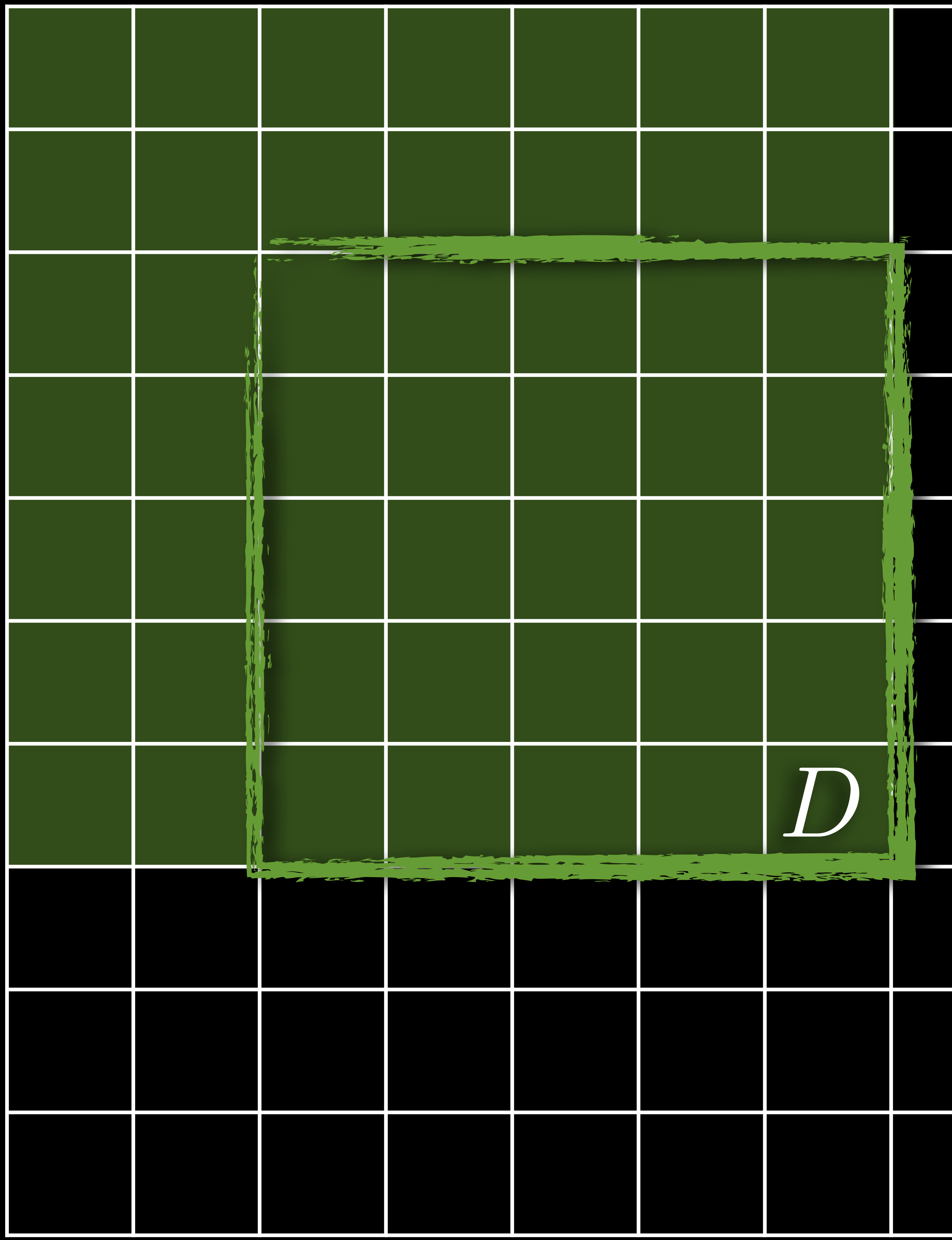
$$D - B - C + A$$



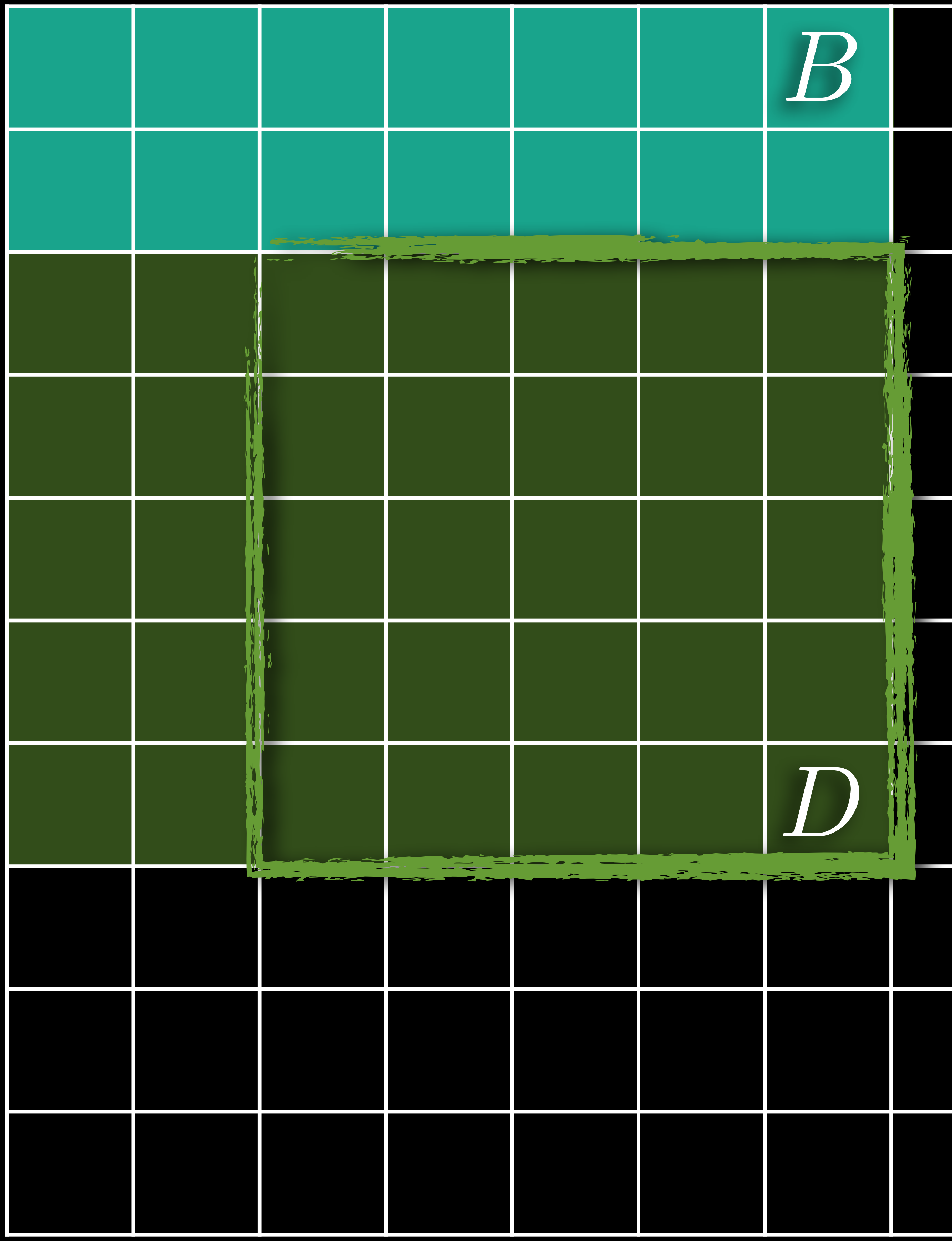
$$D - B - C + A$$



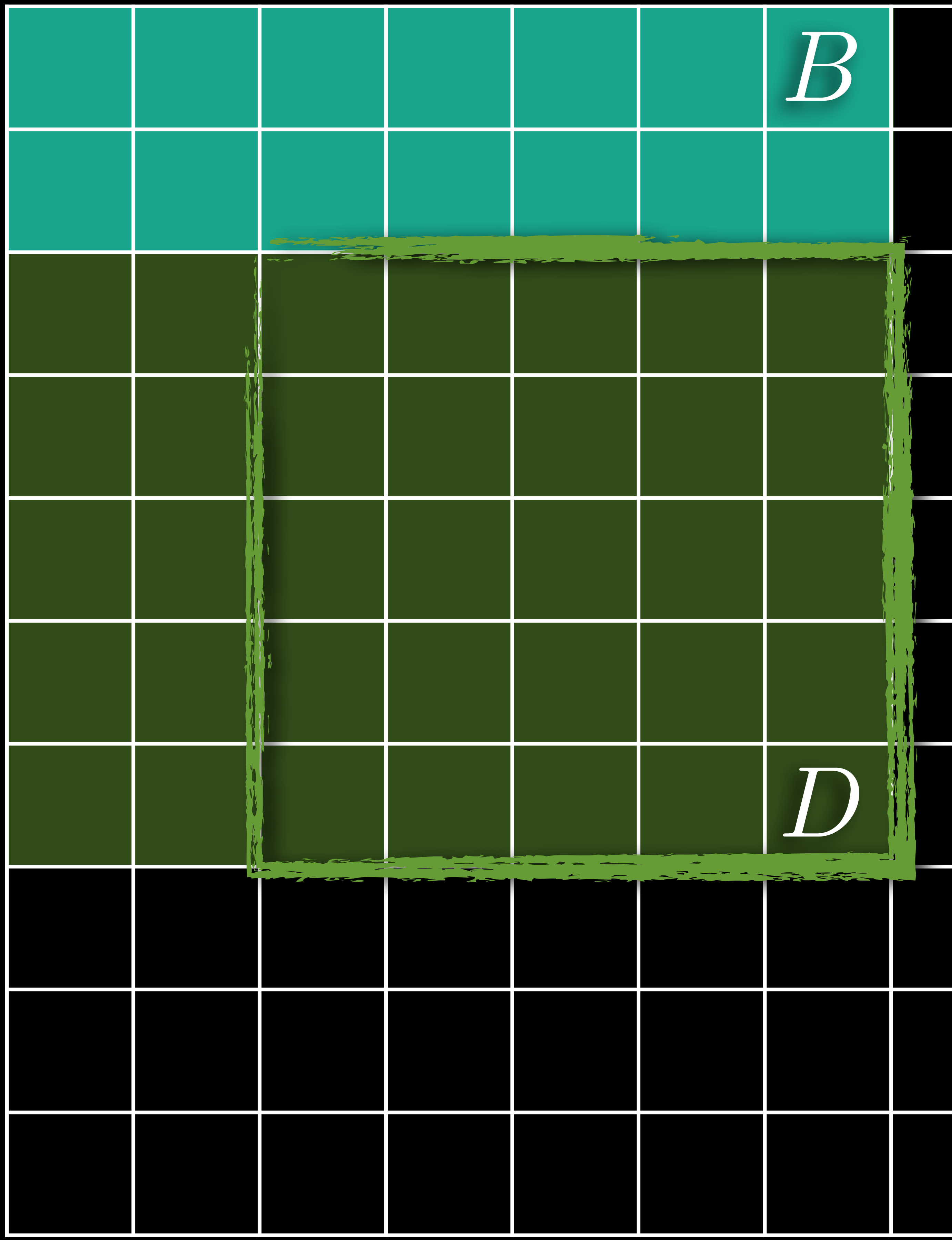
$$D - B - C + A$$



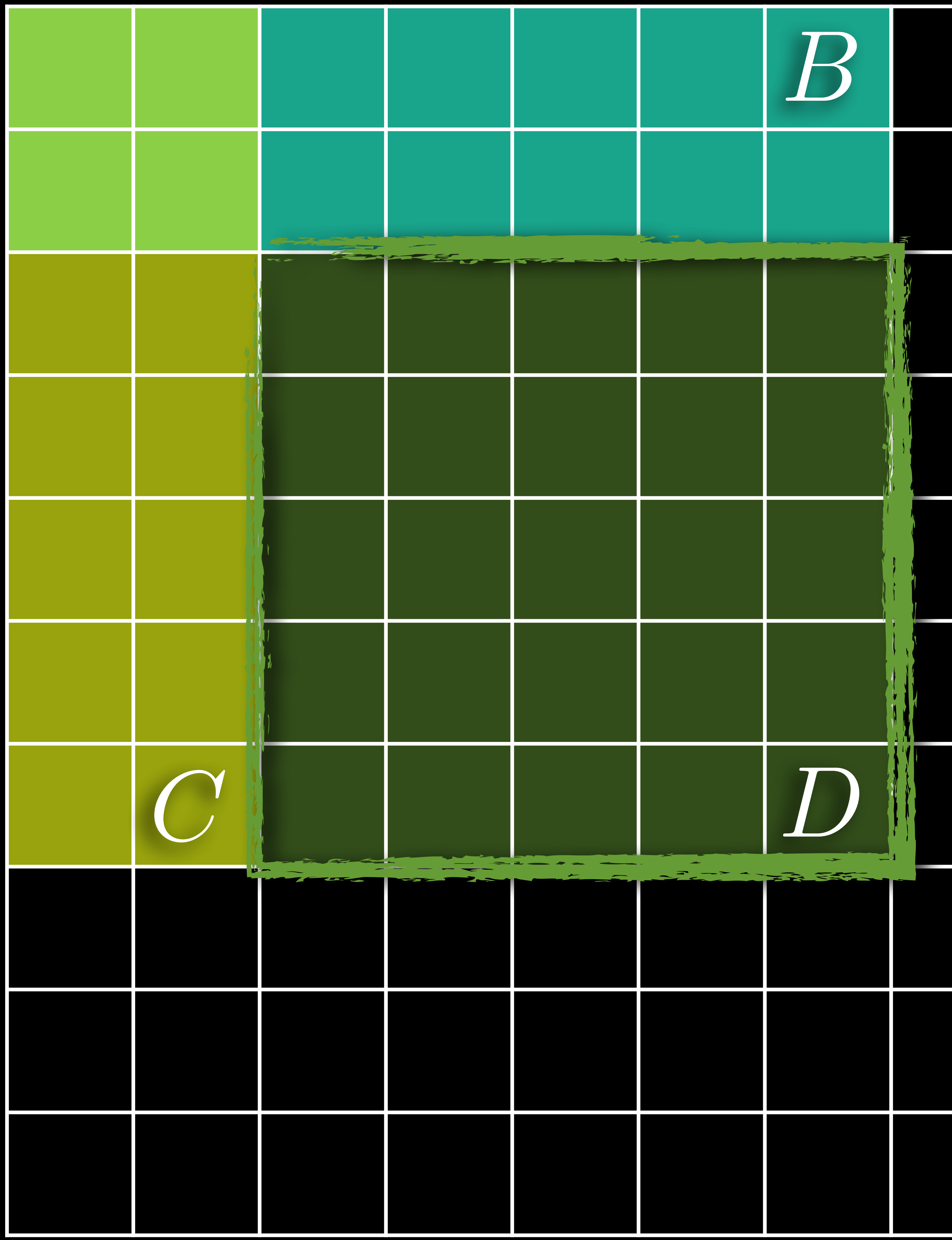
$$D - B - C + A$$



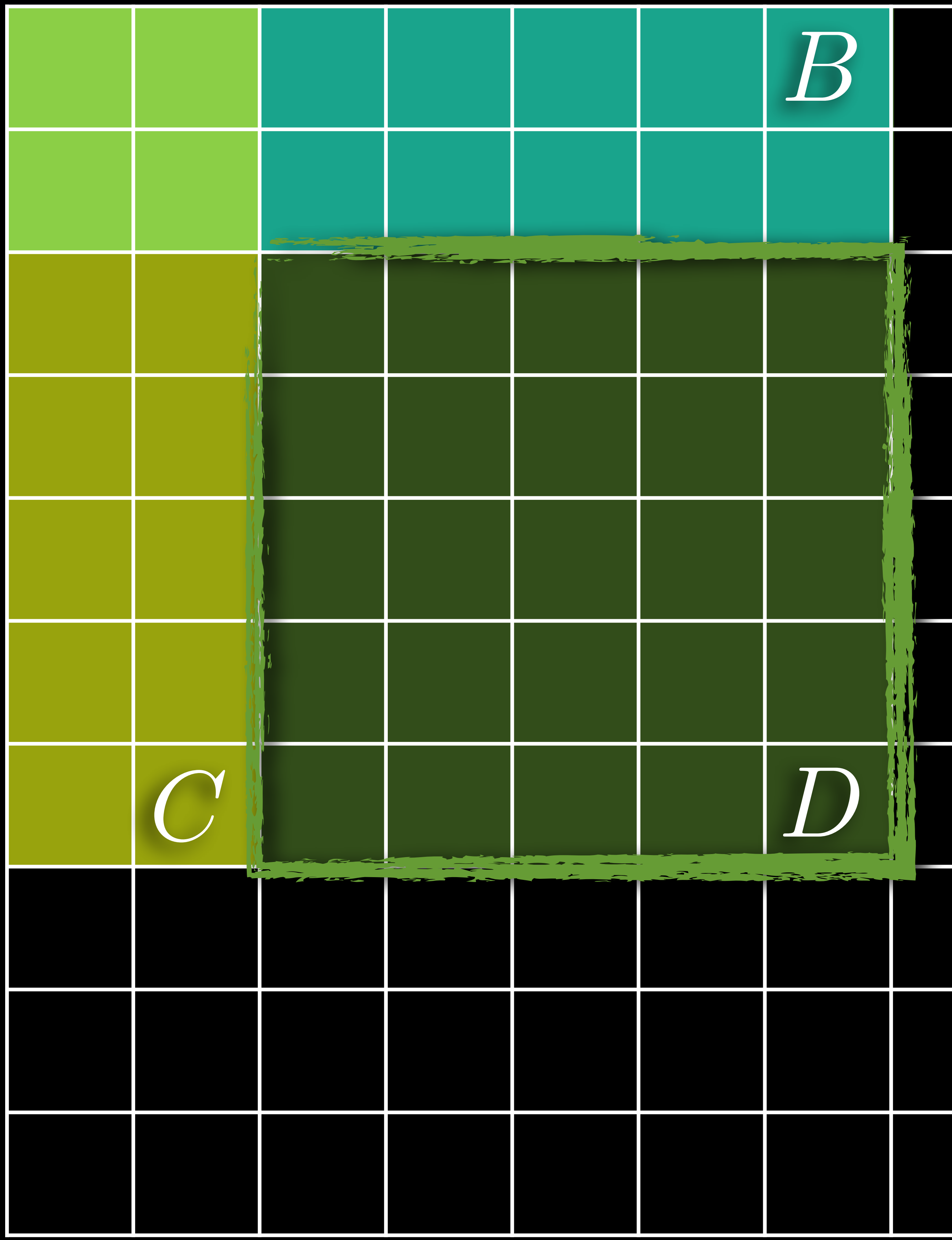
$$D - B - C + A$$



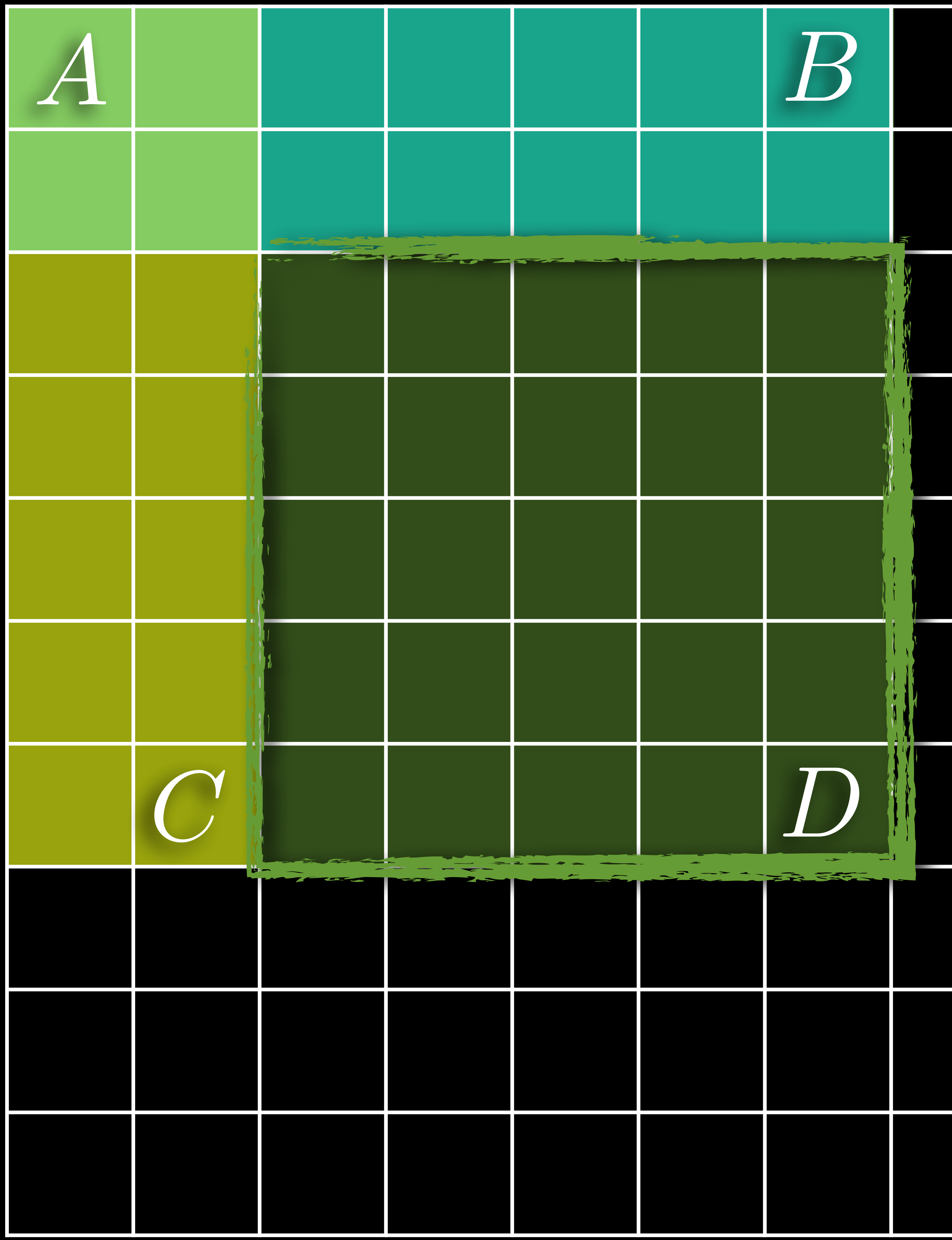
$$D - B - C + A$$



$$D - B - C + A$$



$$D - B - C + A$$



Intro to

Computer Vision

Edge detection

LECTURE TOPICS

Template matching

Review

Correlation $H = G \otimes F$

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

Review

Correlation $H = G \otimes F$

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

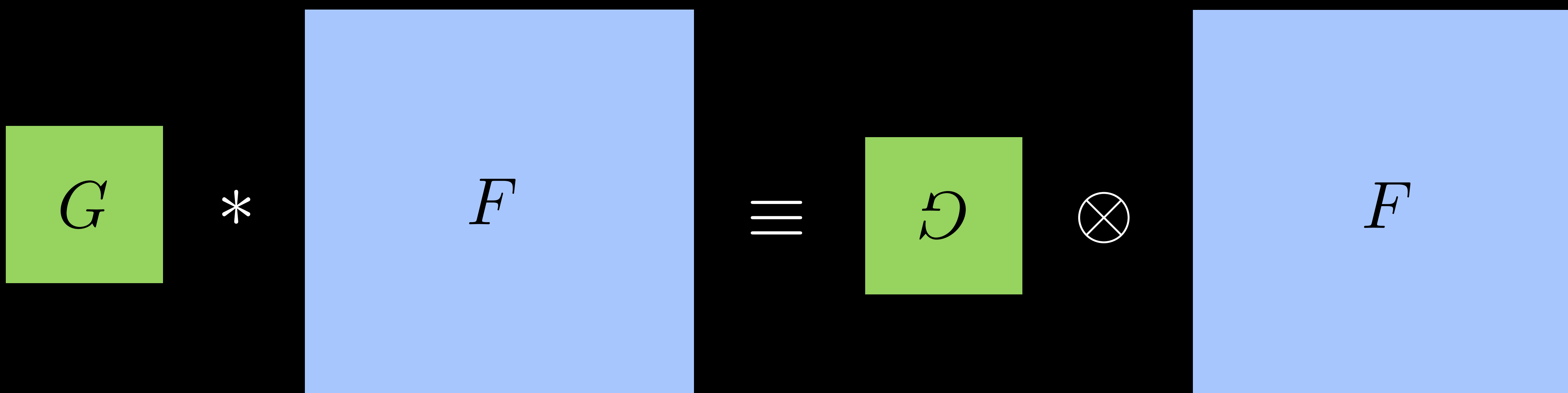
Convolution $H = G * F$

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

Review

Correlation $H = G \otimes F$

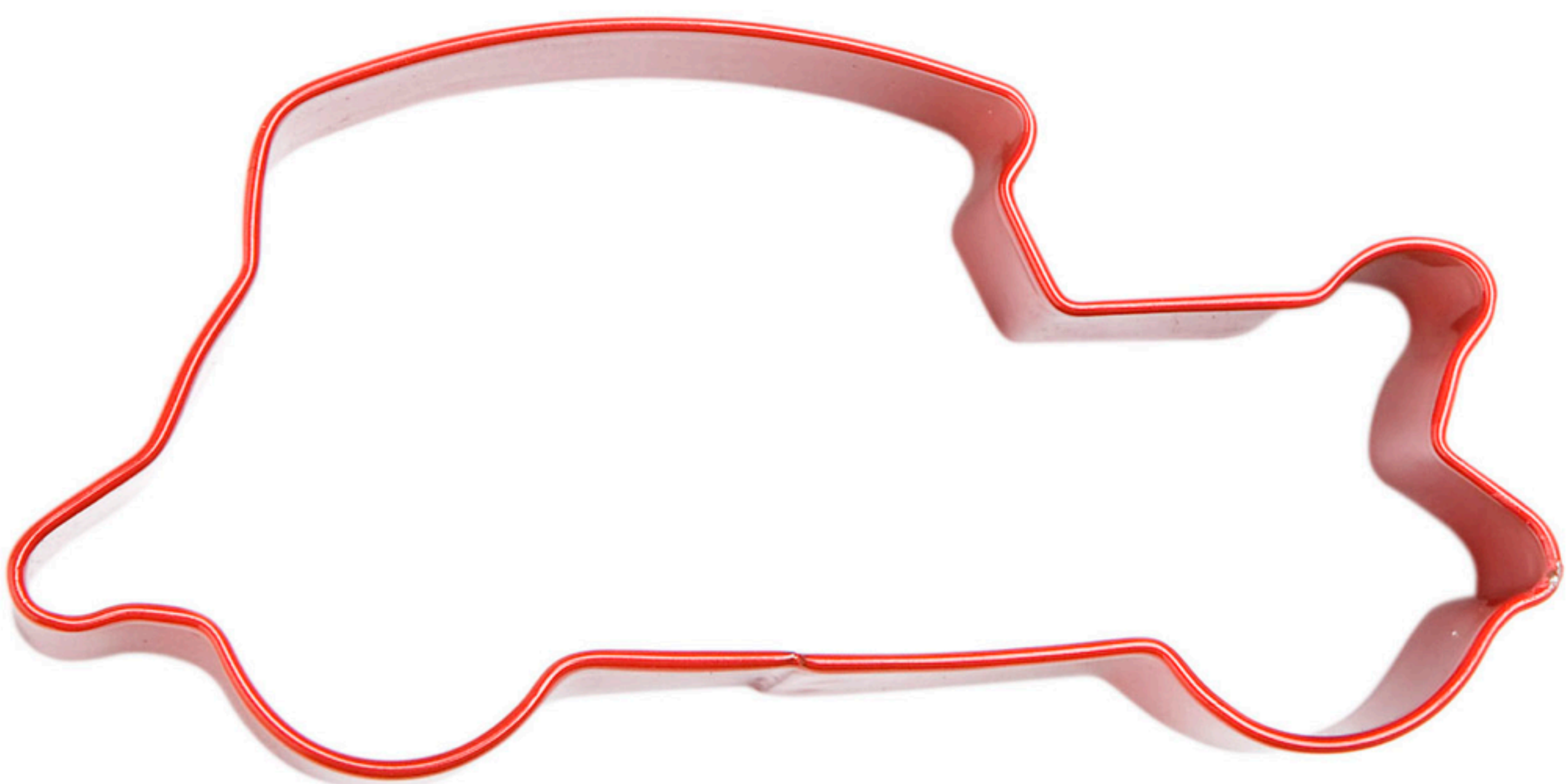
$$H[x,y] = \sum_{u=-K}^{K} \sum_{v=-K}^{K} G[u,v]F[x+u, y+v]$$



Convolution $H = G * F$

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

Template
matching

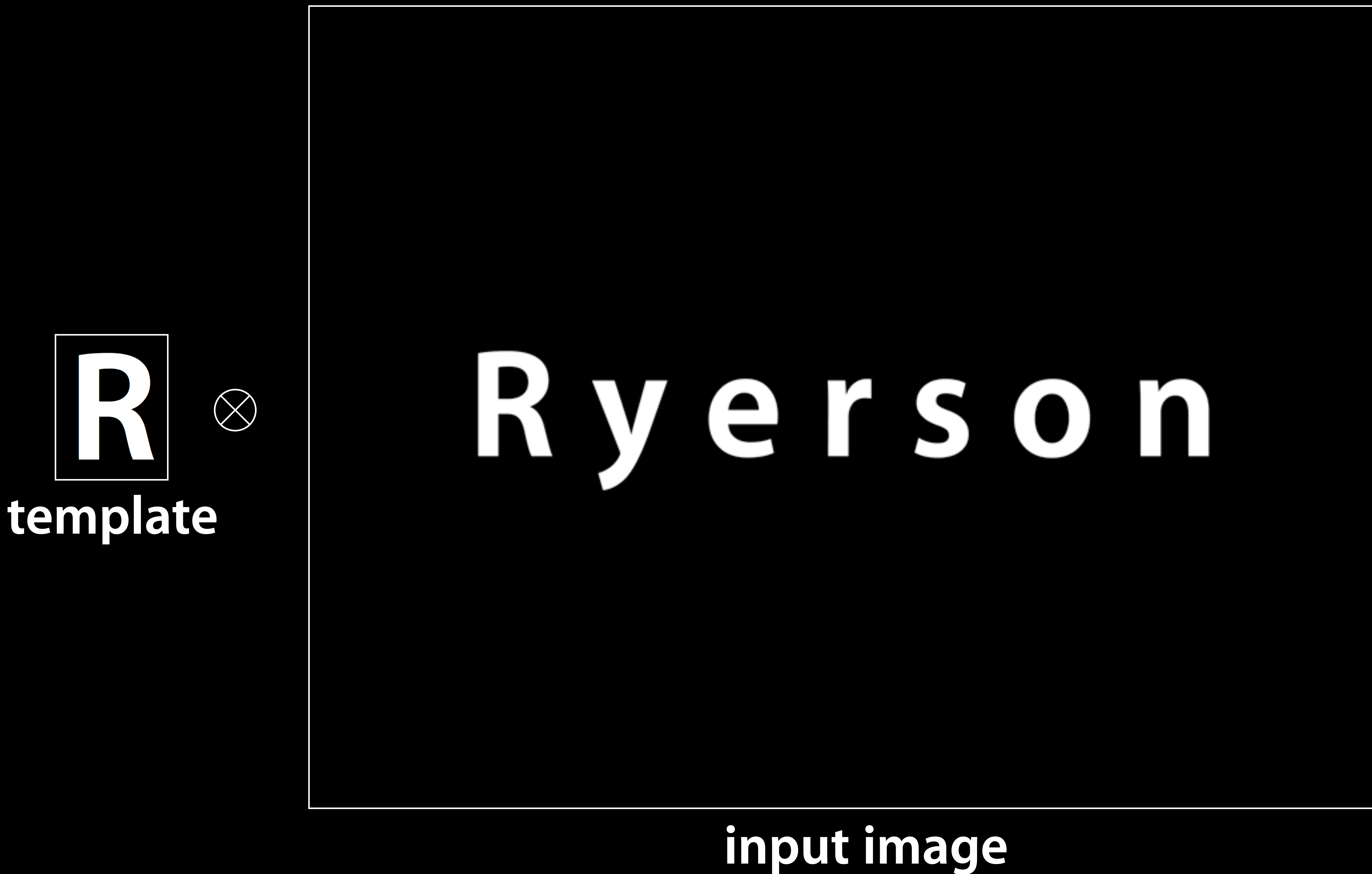


R

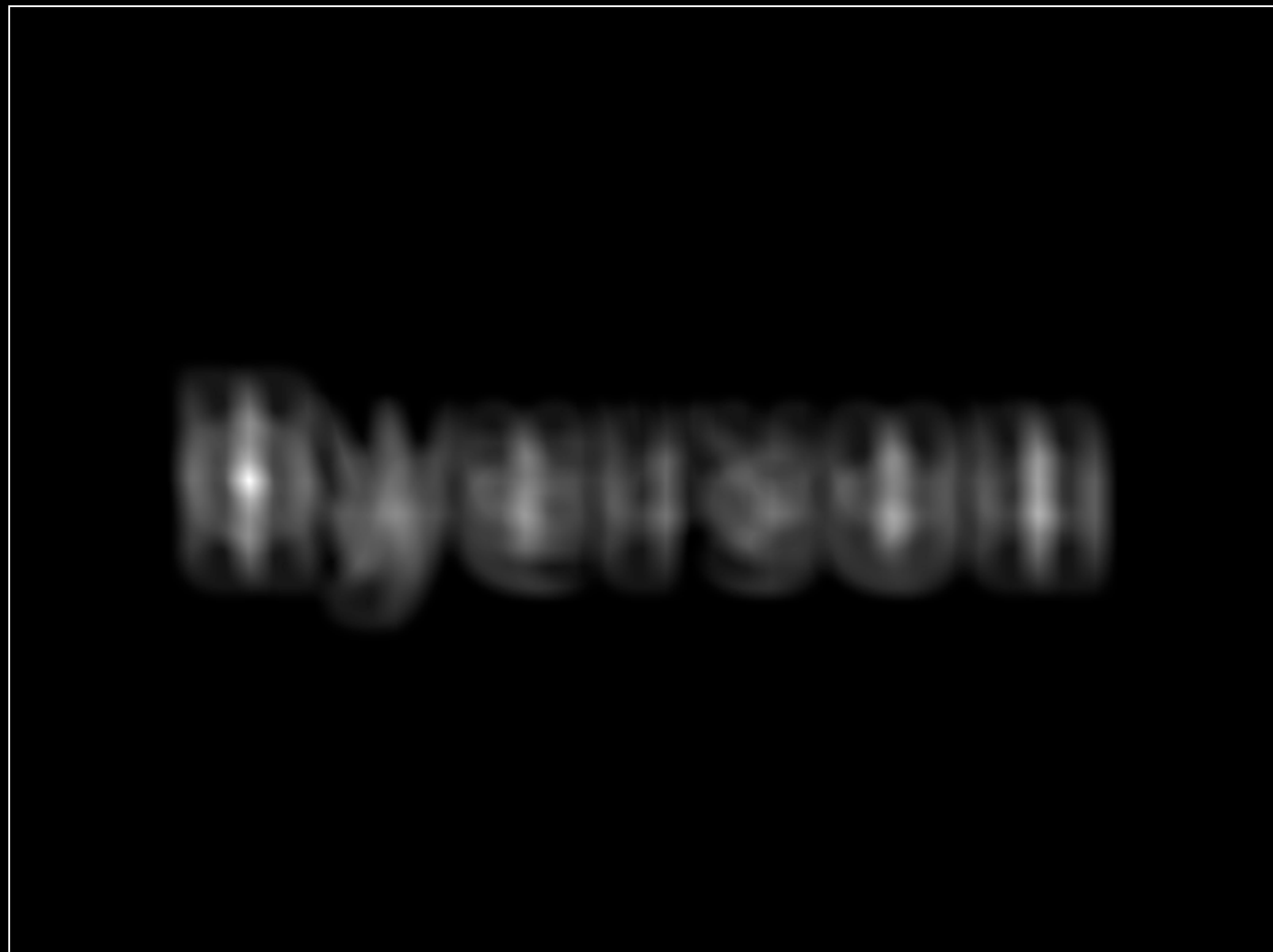
template

template

input image

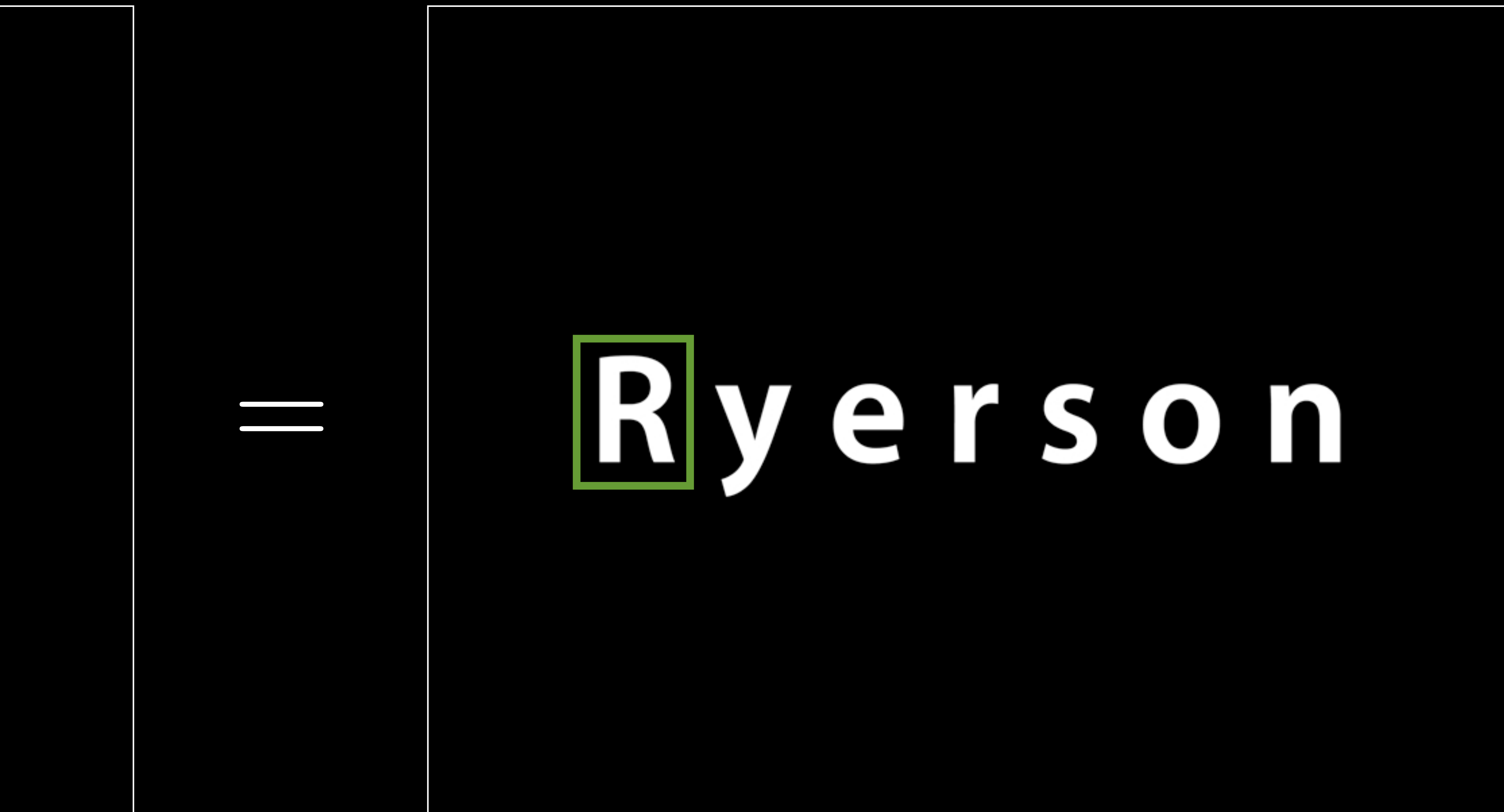

R
template
Ryerson
input image

=



correlation map

correlation map



detected template

Template matching

intuition

Definition:

The dot product, or scalar product, between two vectors is defined as:

Definition:

The dot product, or scalar product, between two vectors is defined as:

$$\mathbf{x} \cdot \mathbf{y}$$

Definition:

The dot product, or scalar product, between two vectors is defined as:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i$$

Definition:

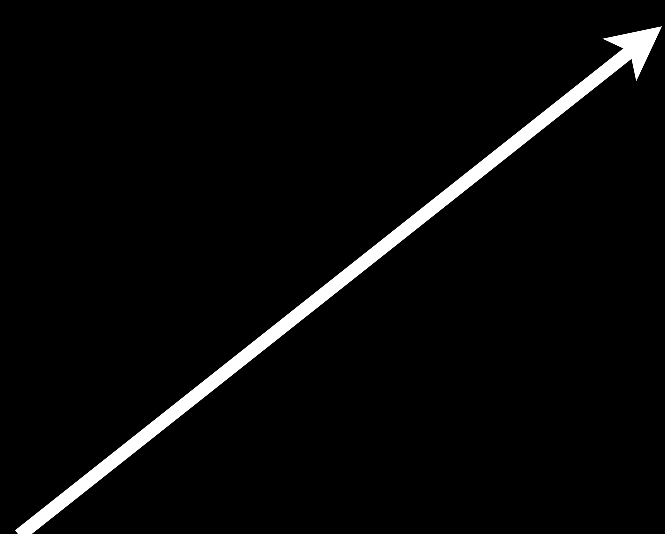
The dot product, or scalar product, between two vectors is defined as:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta$$

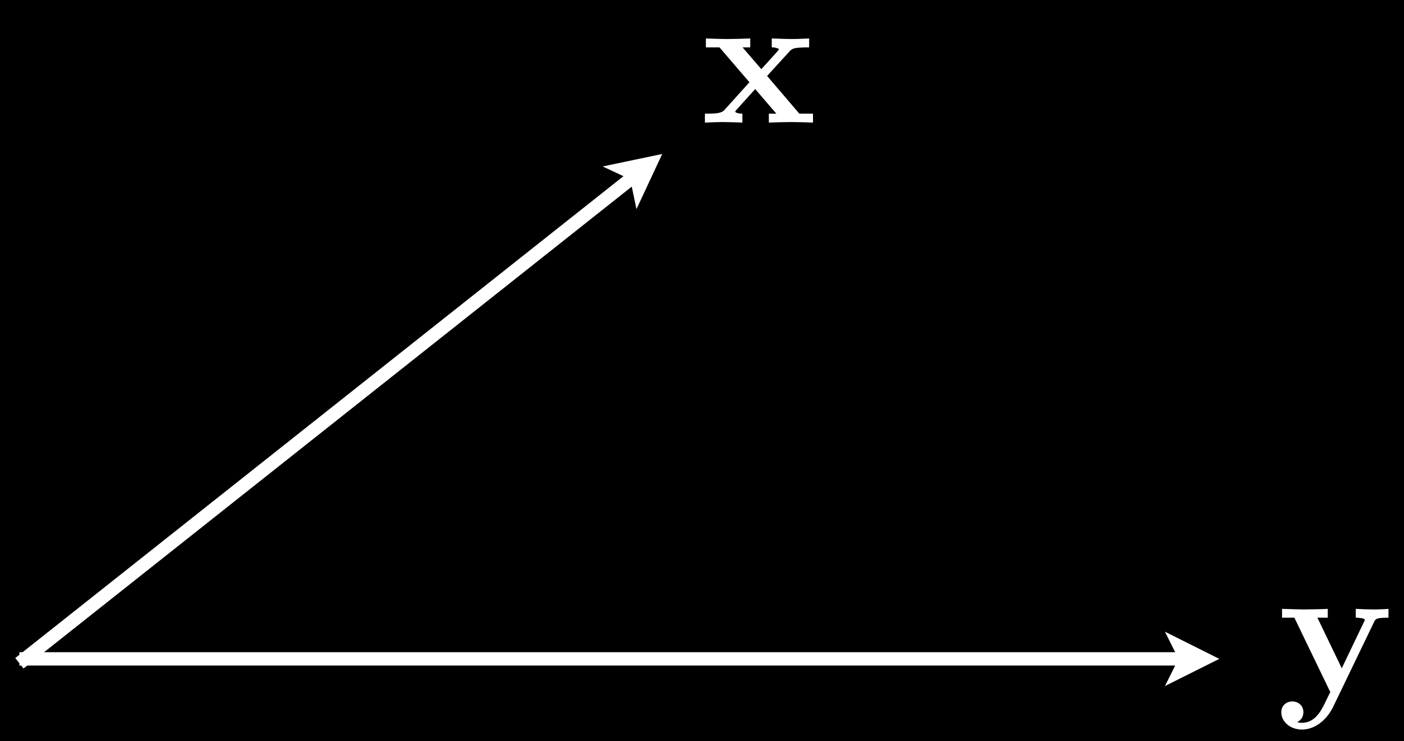
$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta$$

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta$$

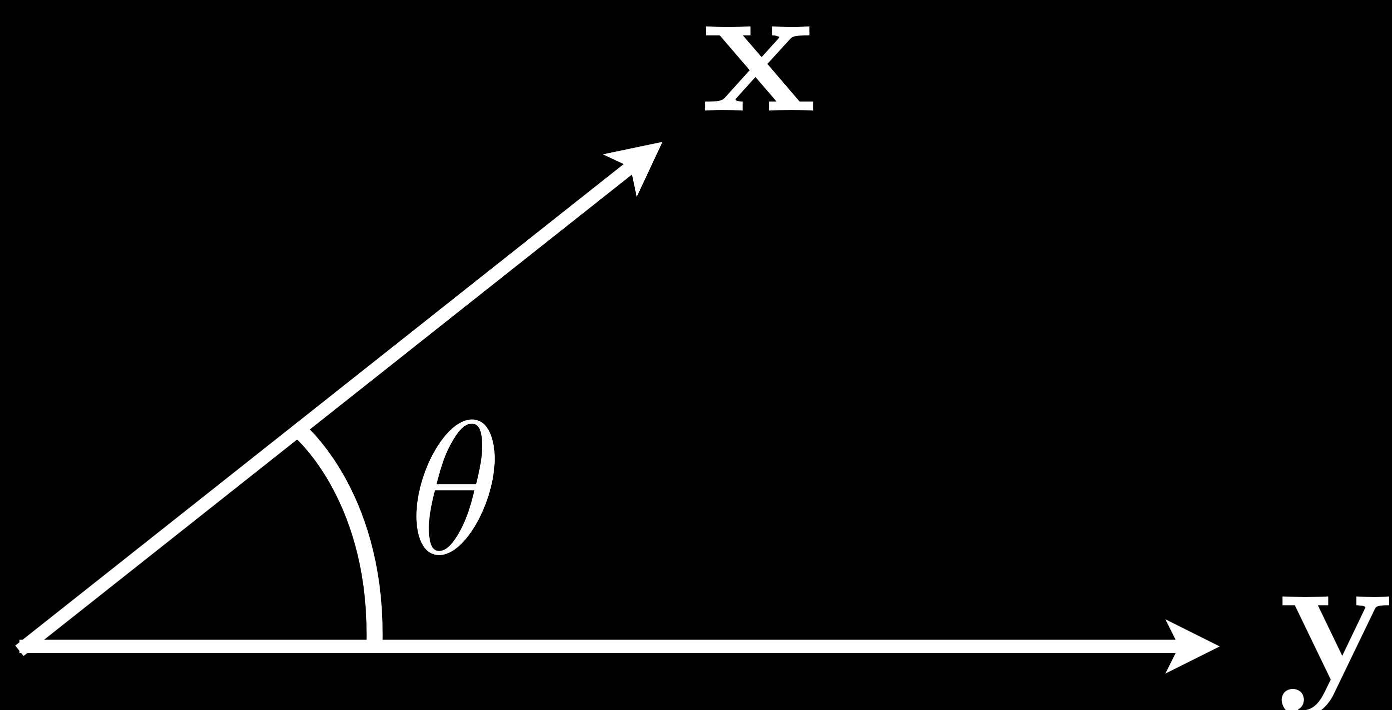
$\mathbf{x}$



$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta$$

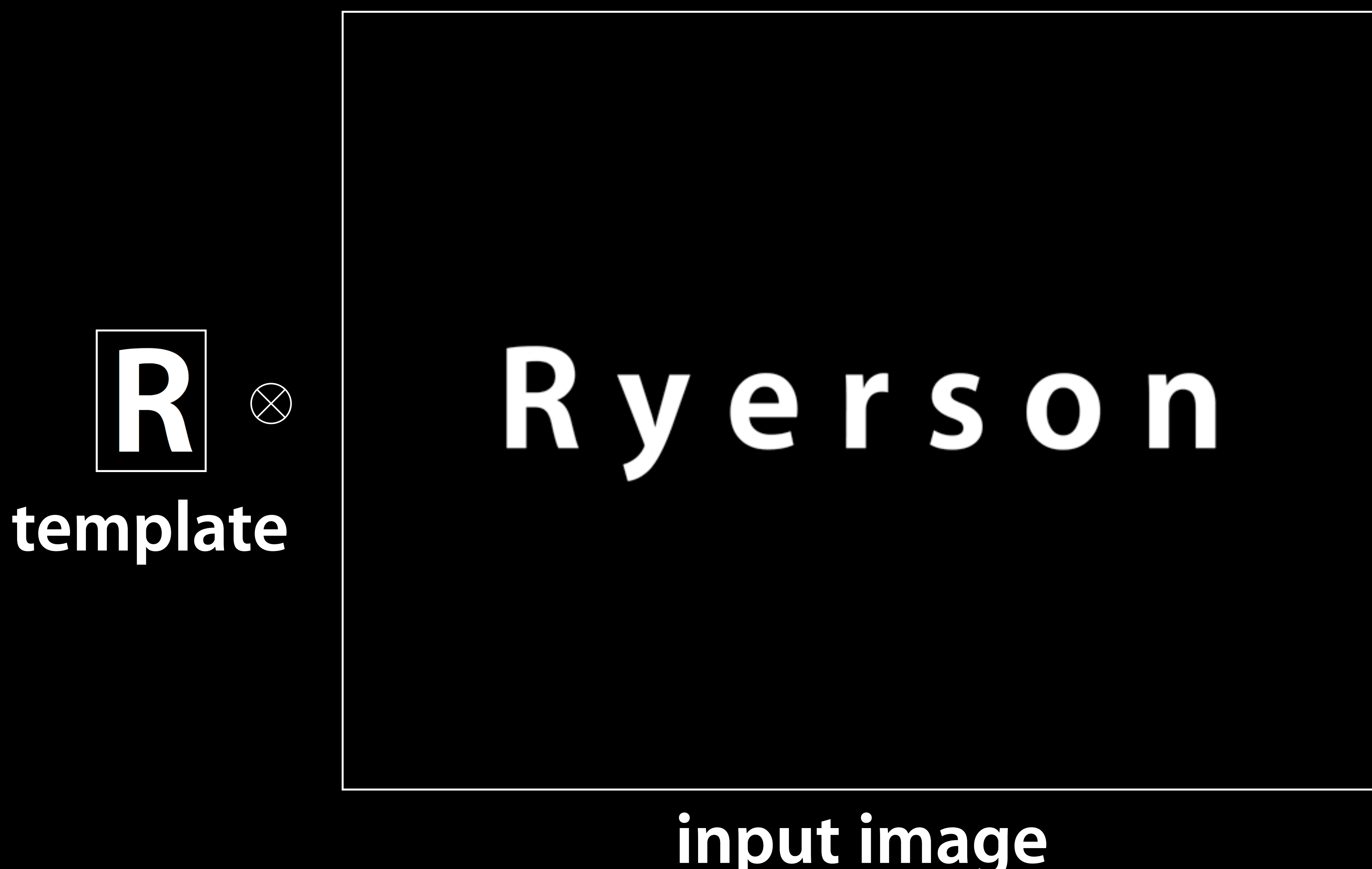


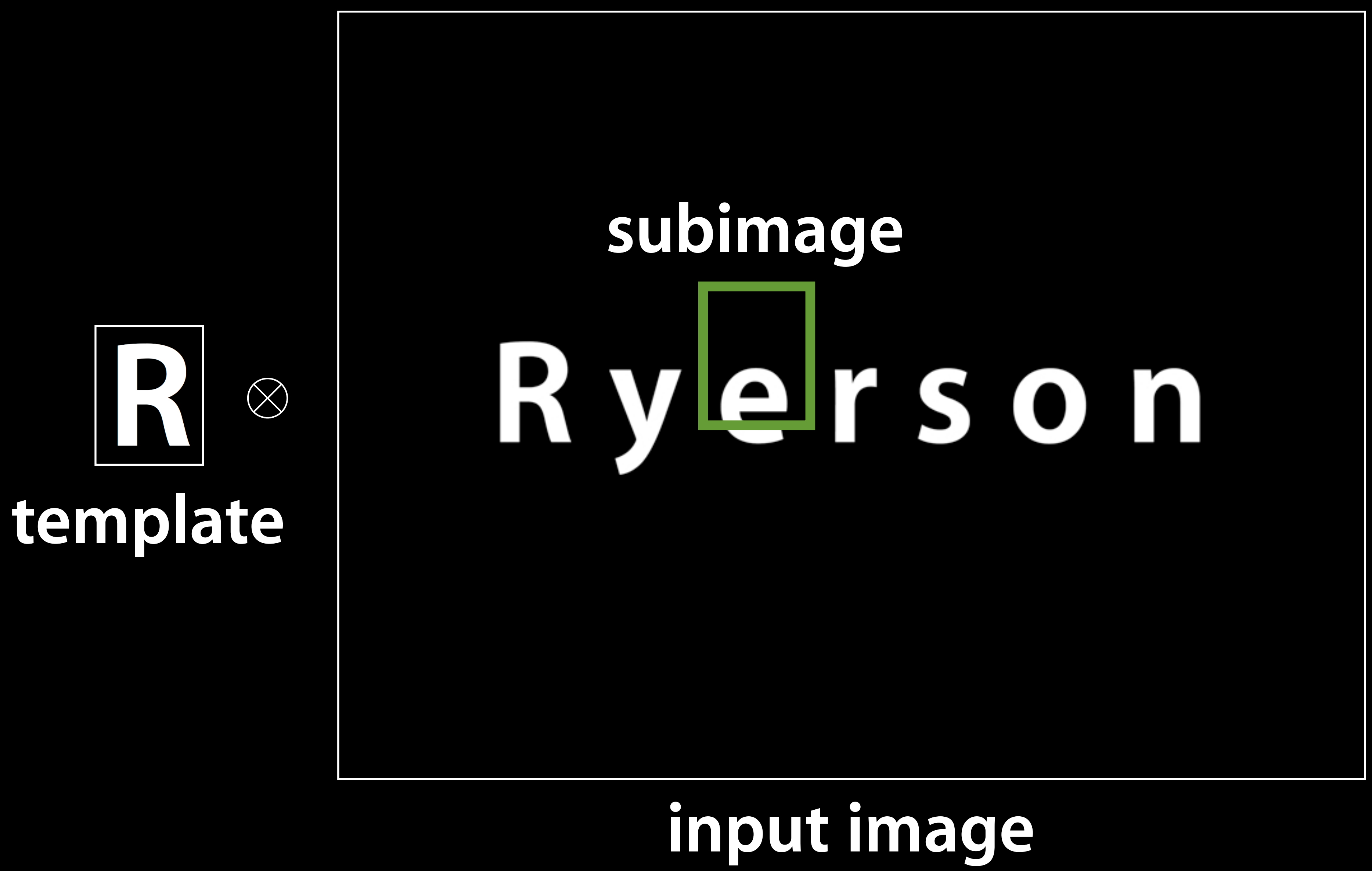
$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta$$

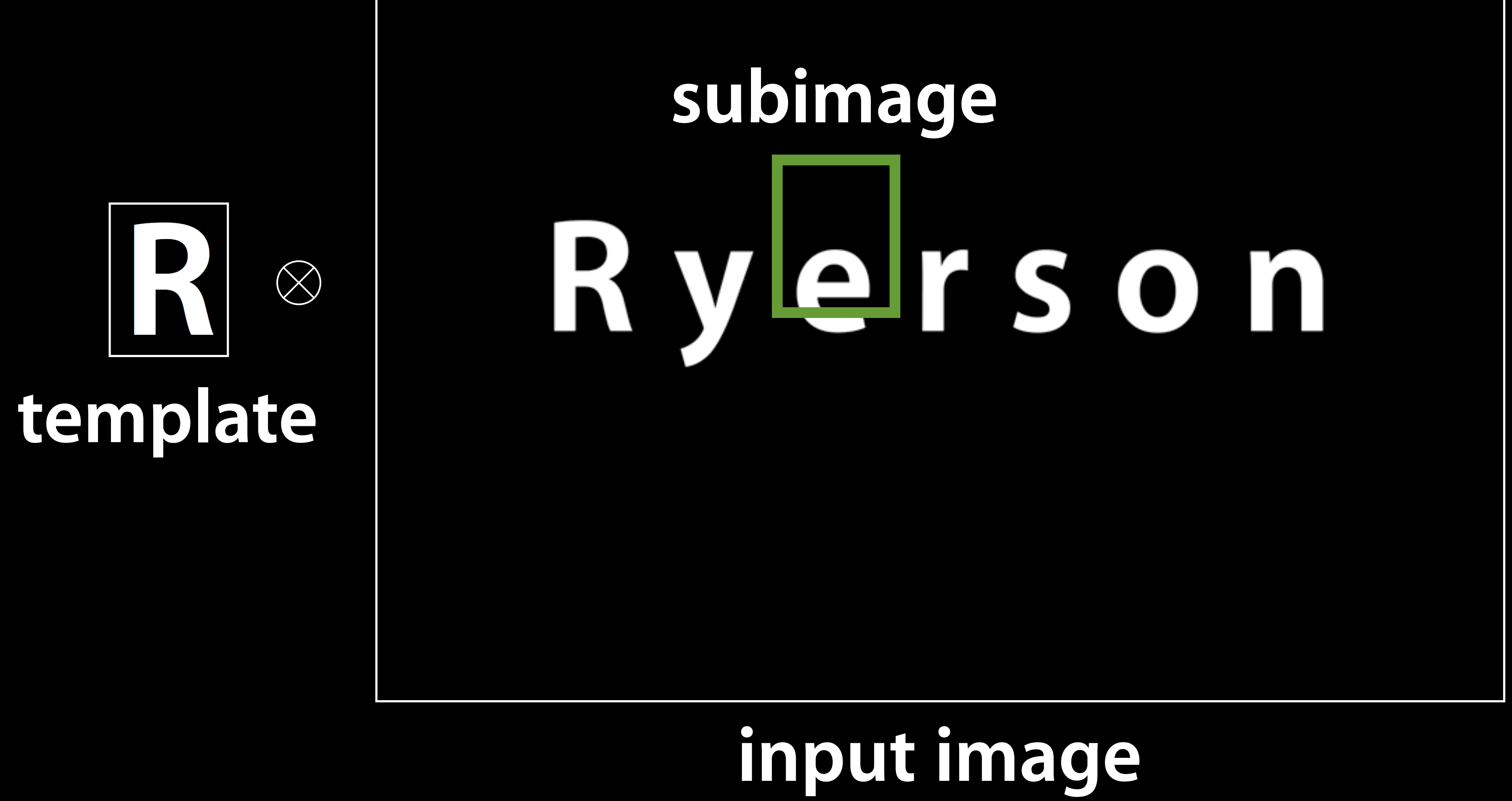


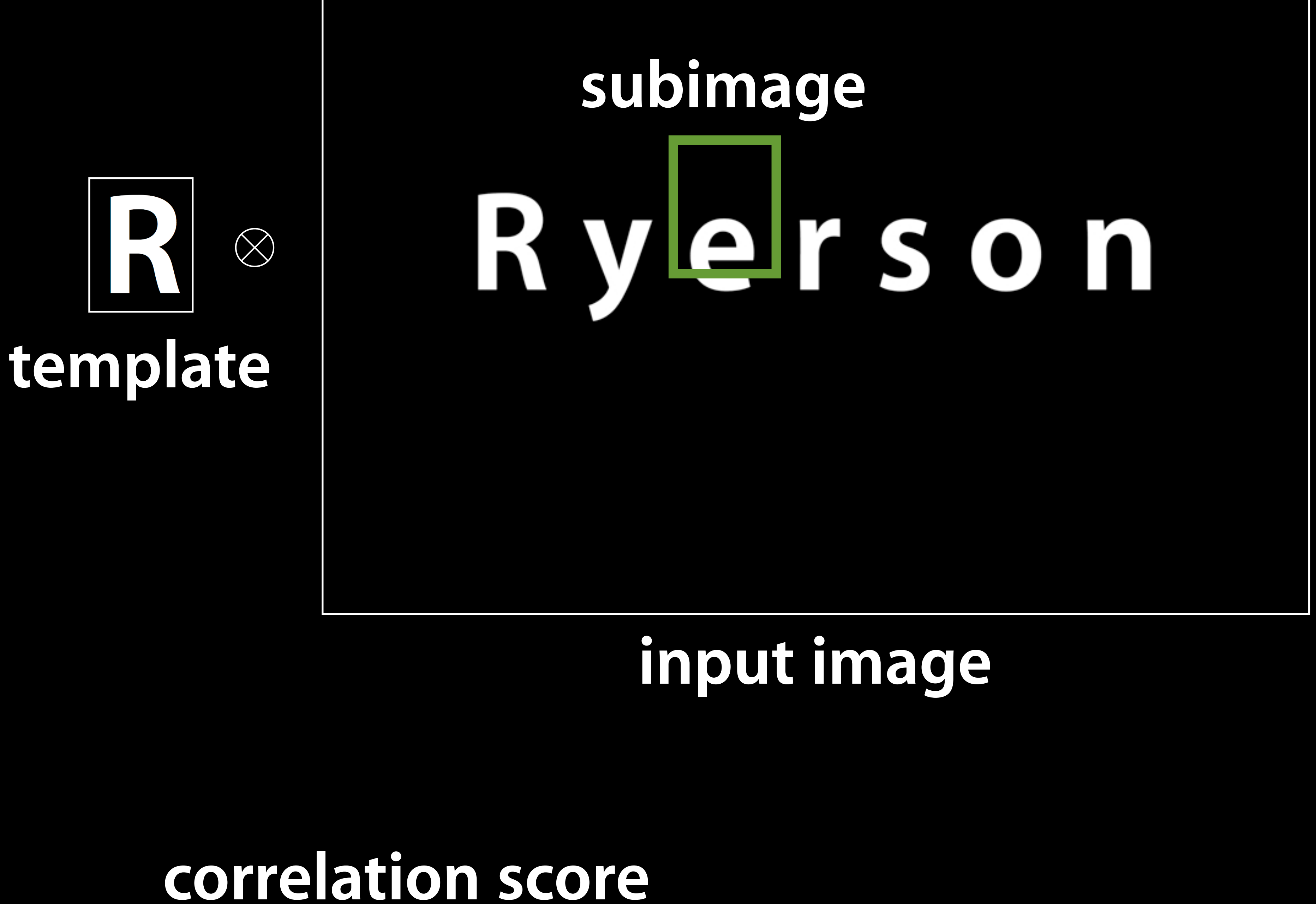
Template matching

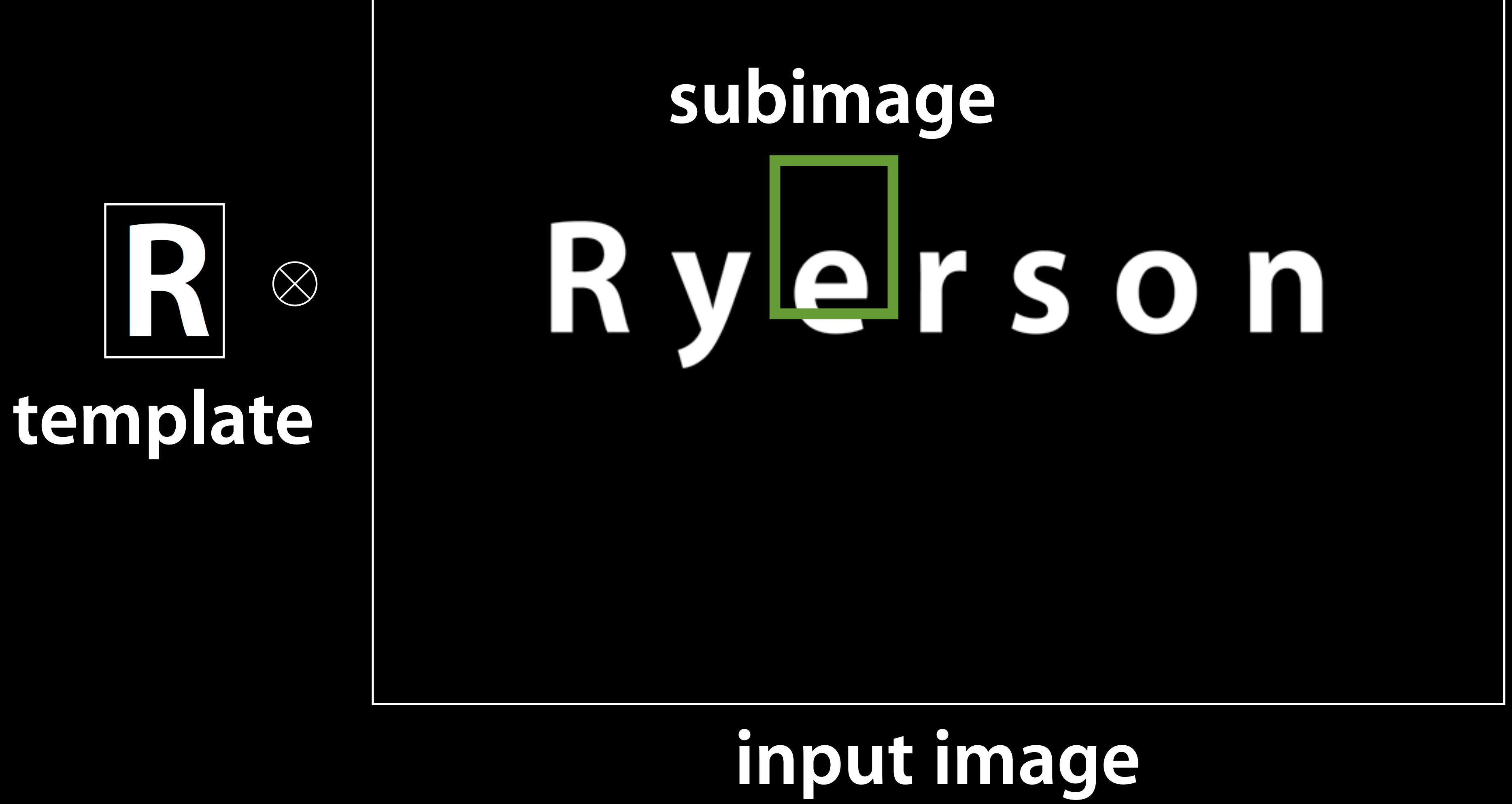
intuition


R
template
Ryerson
input image

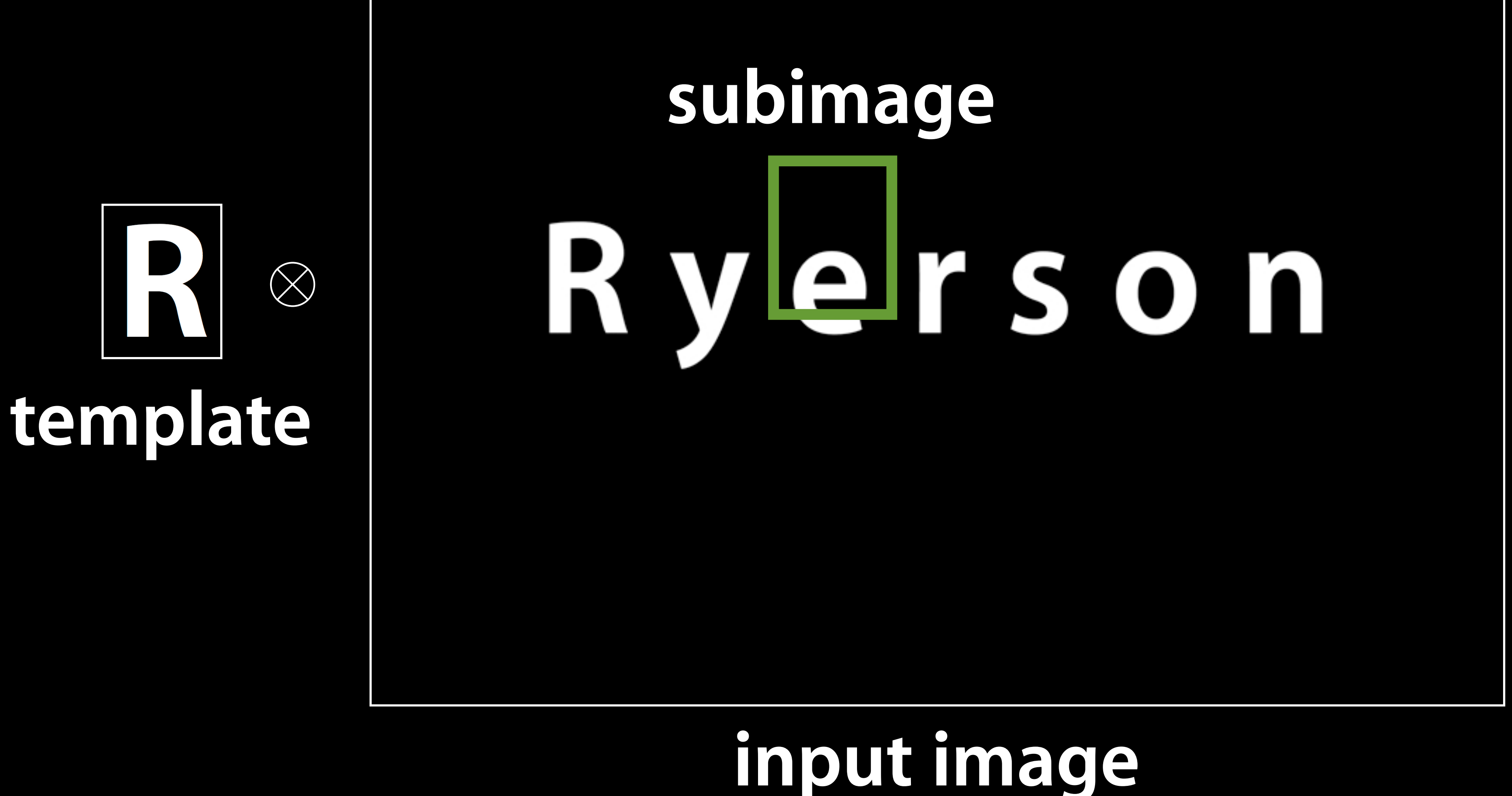

subimage
R
template
R y e r s o n
input image


subimage
R
Ryerson
template
input image


subimage
R
Ryerson
template
input image
correlation score

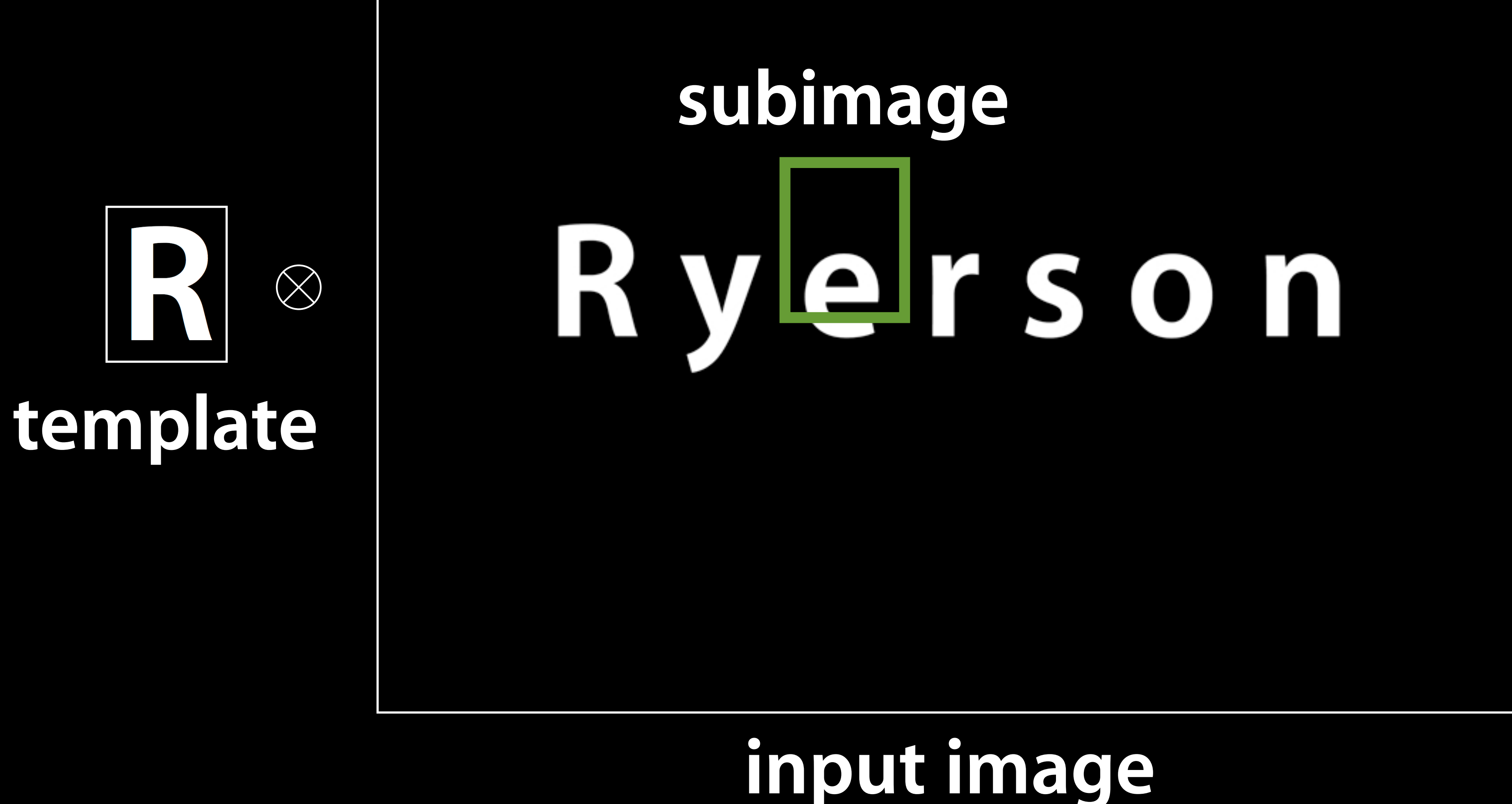


correlation score $= \mathbf{t} \cdot \mathbf{s}$



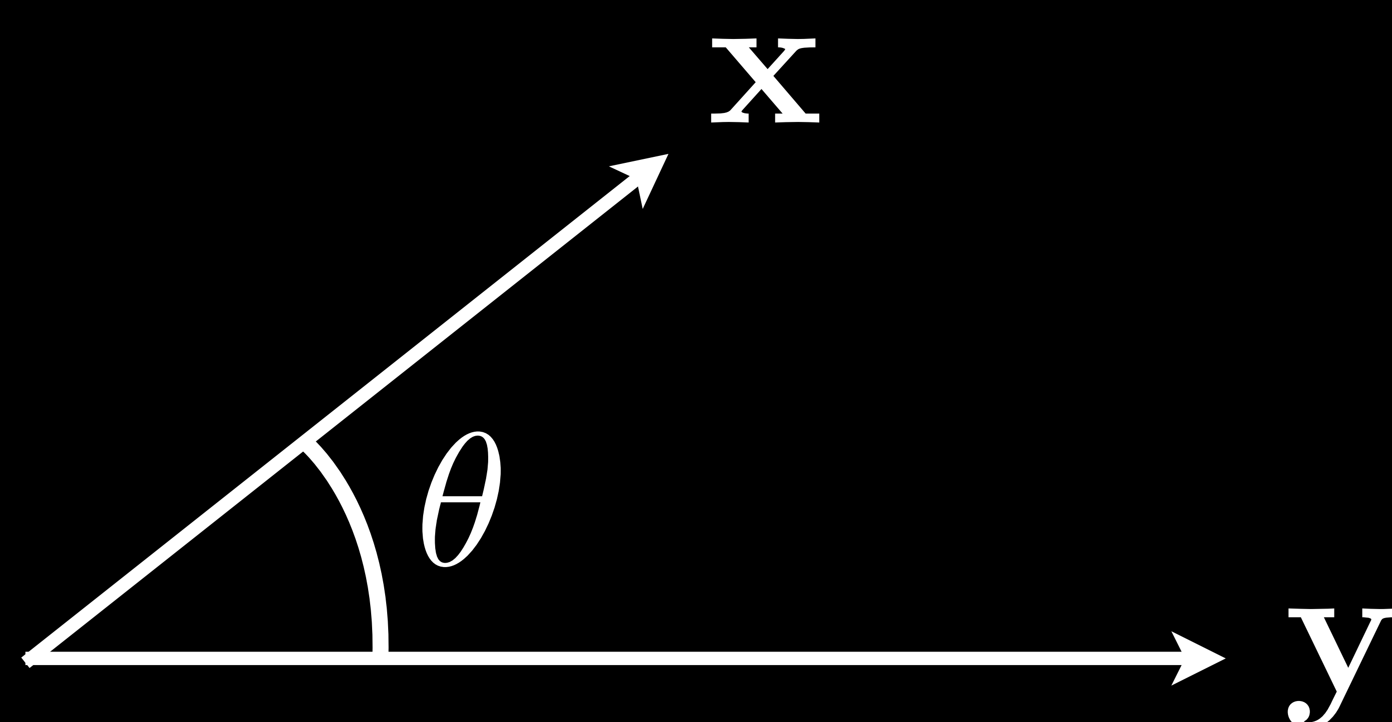
correlation score $= \mathbf{t} \cdot \mathbf{s}$

scalar product between vectors

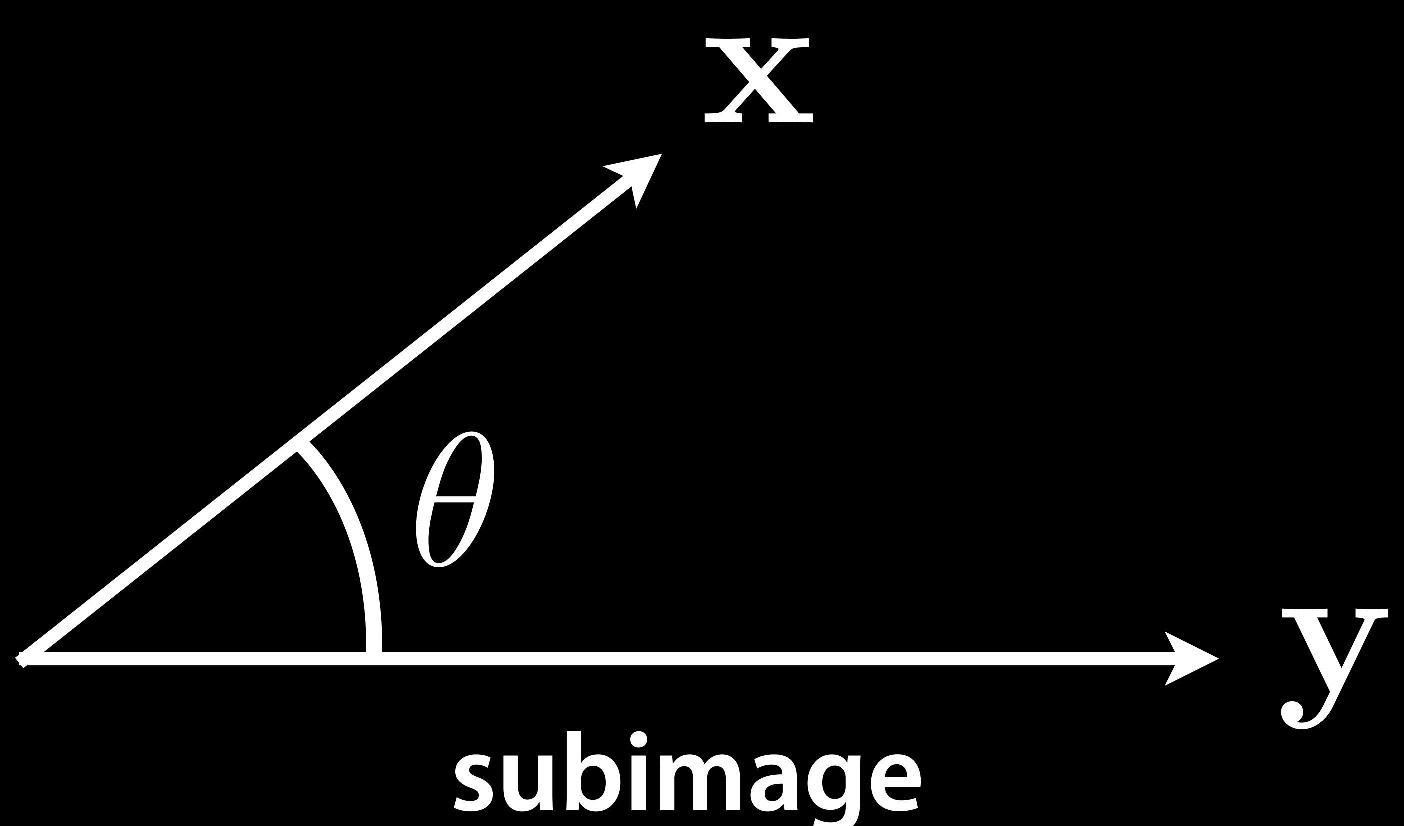


correlation score $= \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\|\|\mathbf{s}\| \cos\theta$

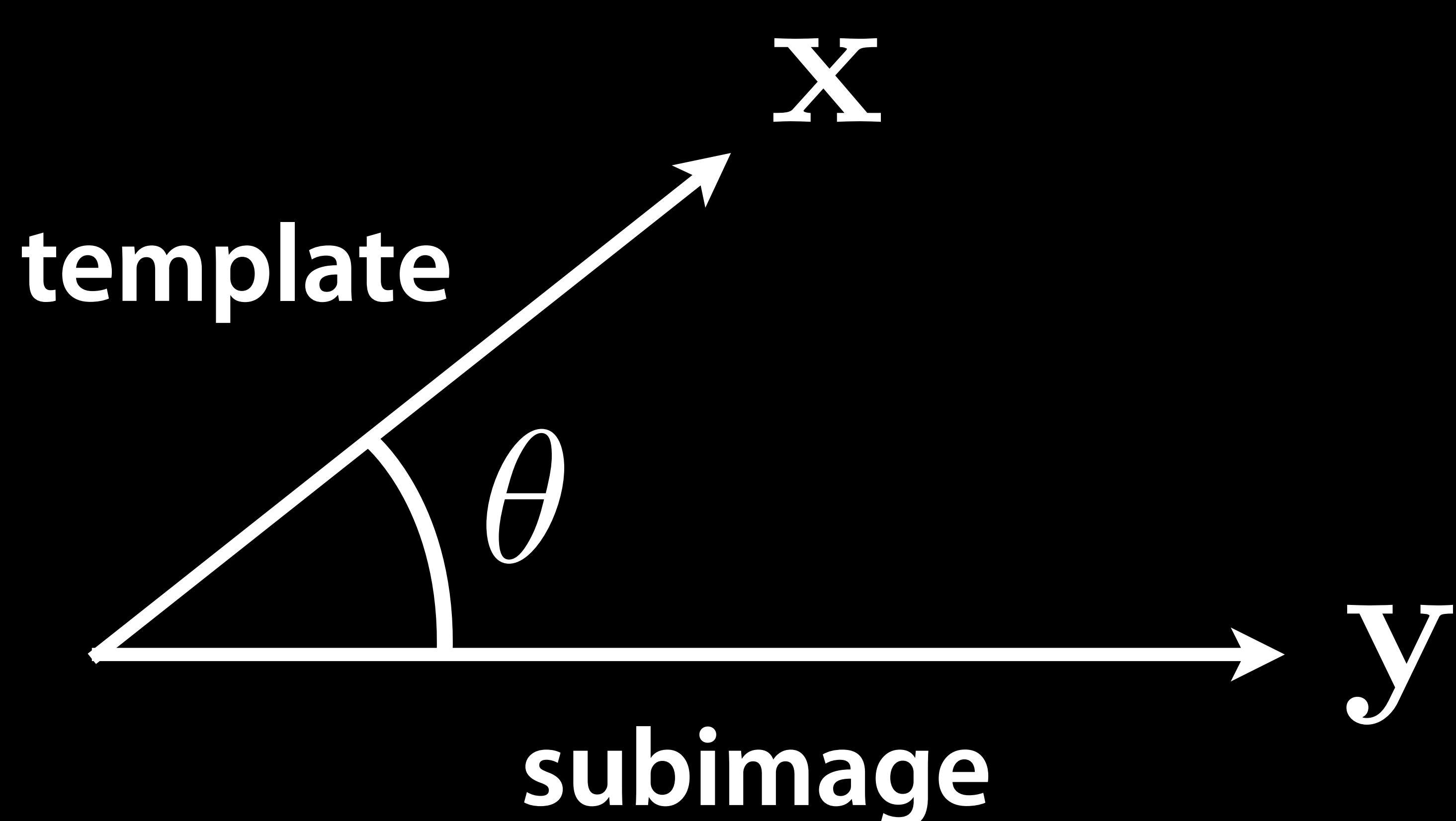
$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta$$

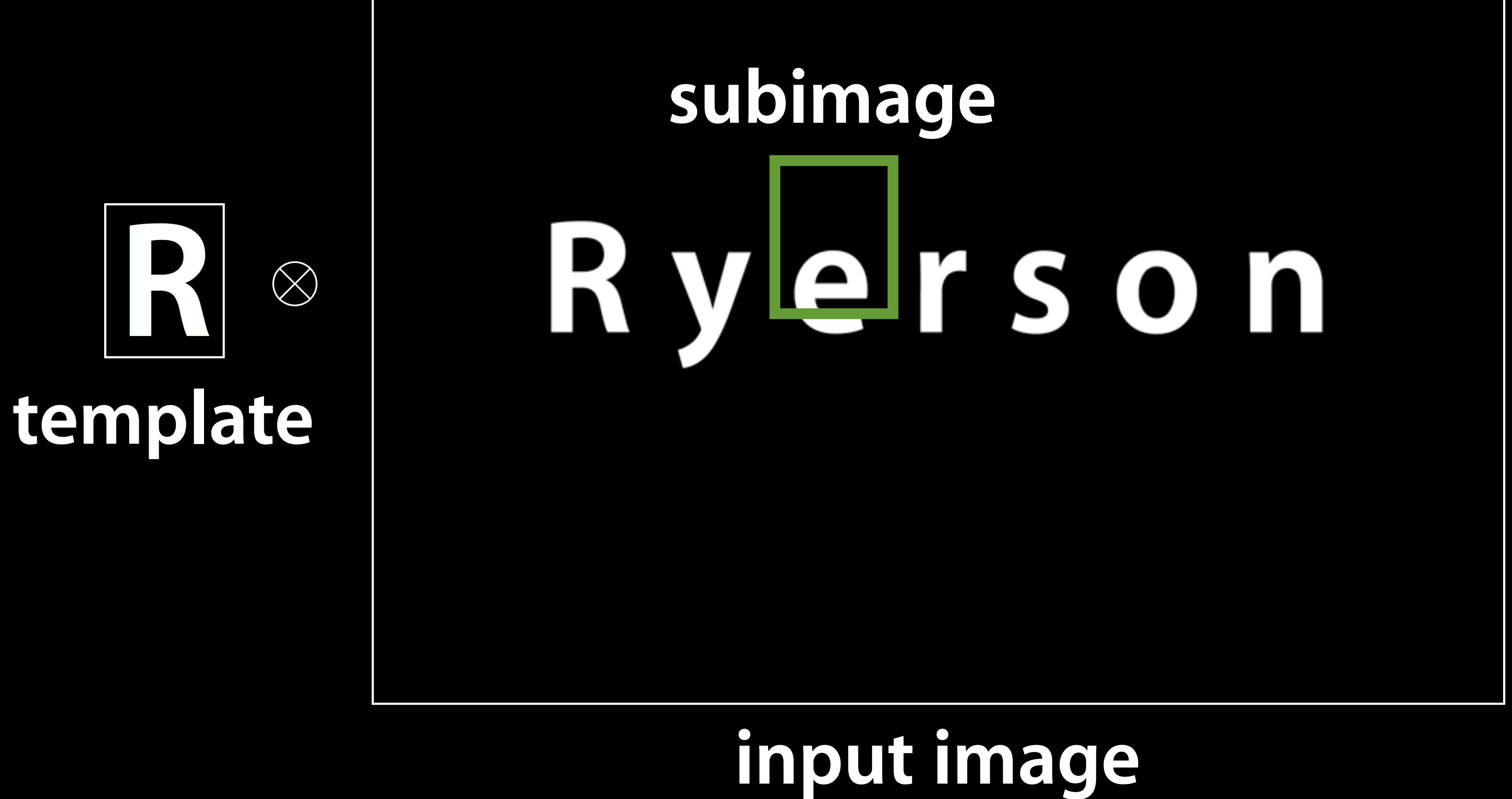


$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta$$

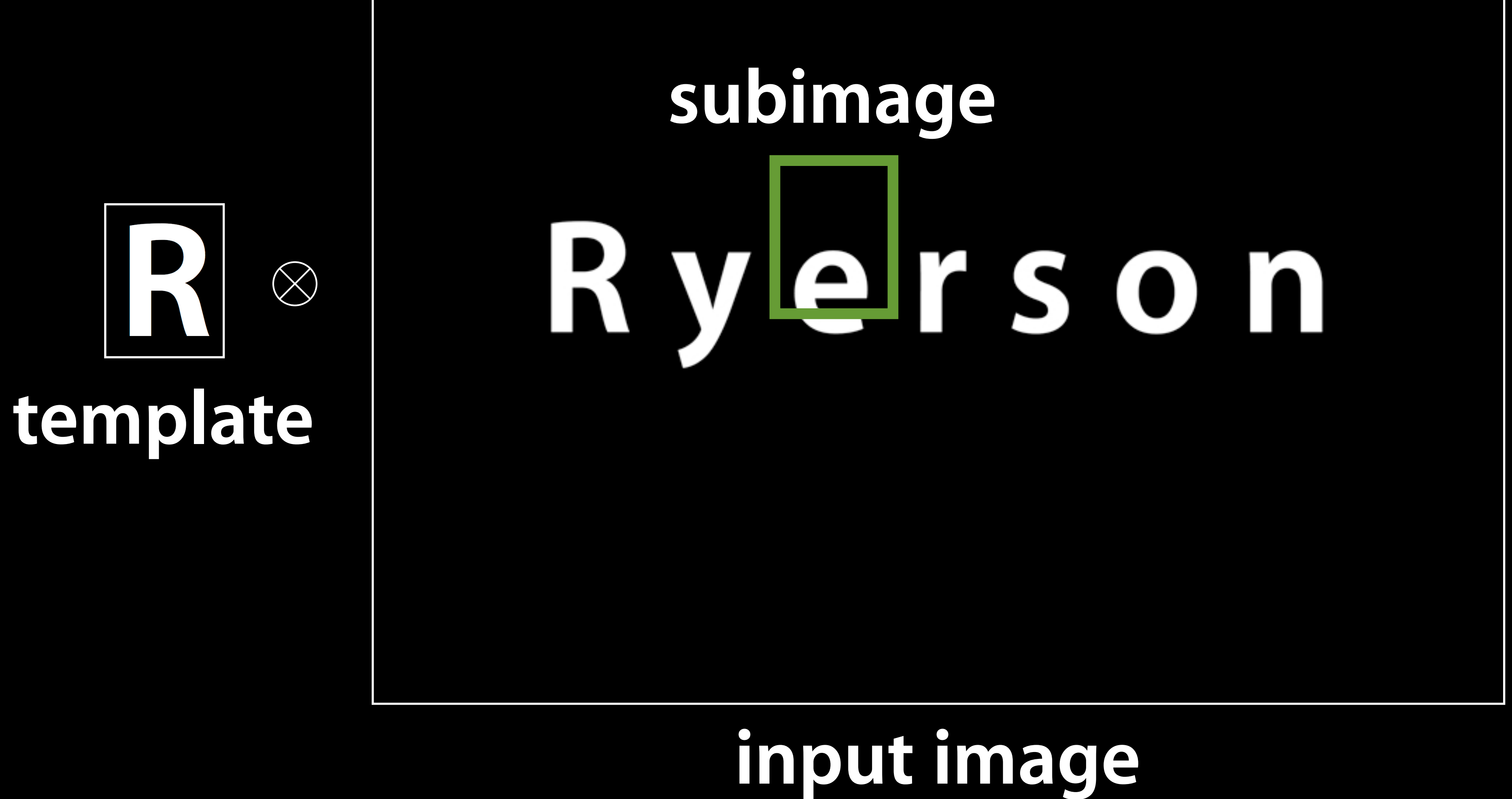


$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta$$



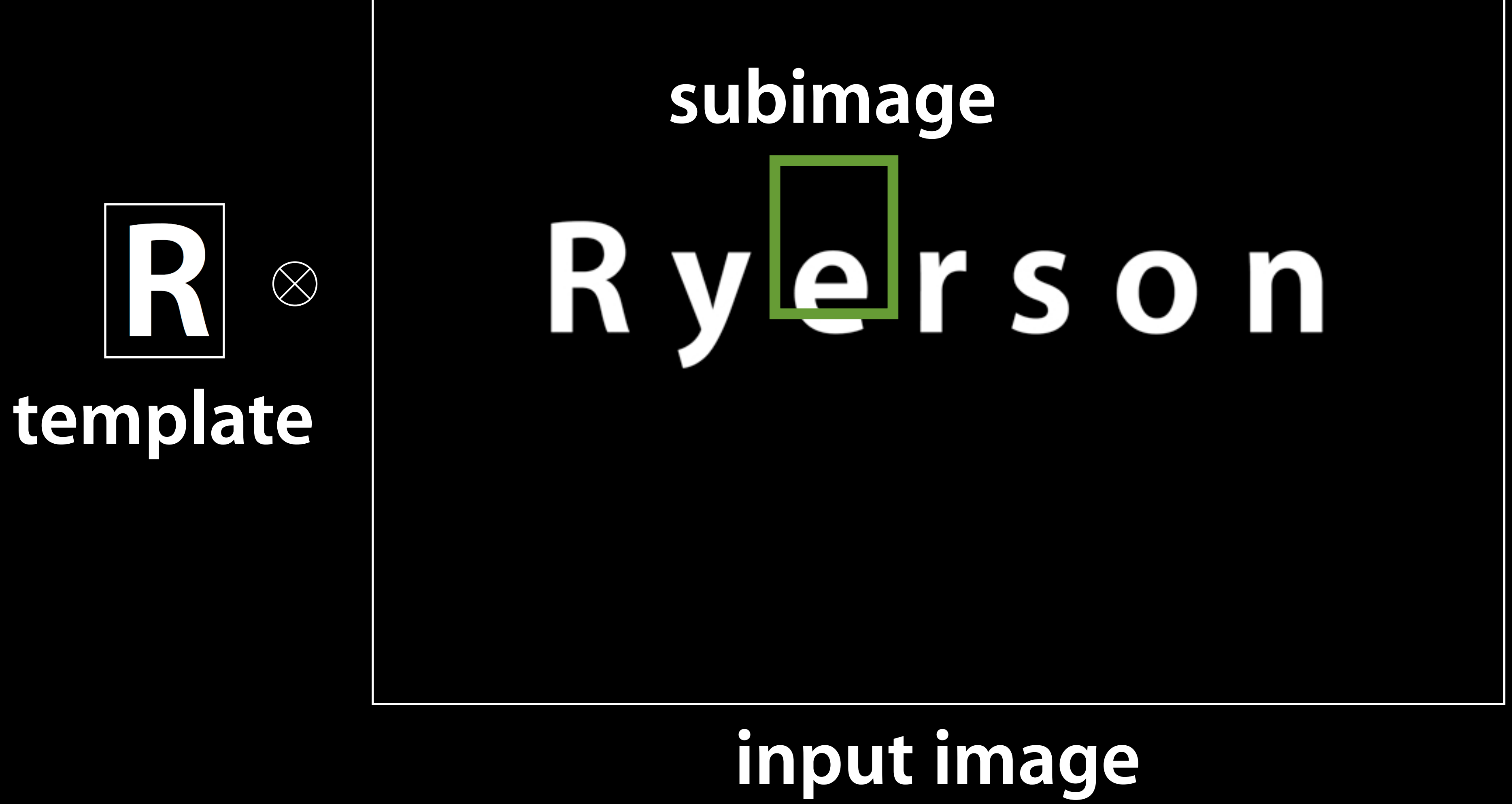


correlation score $= \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$



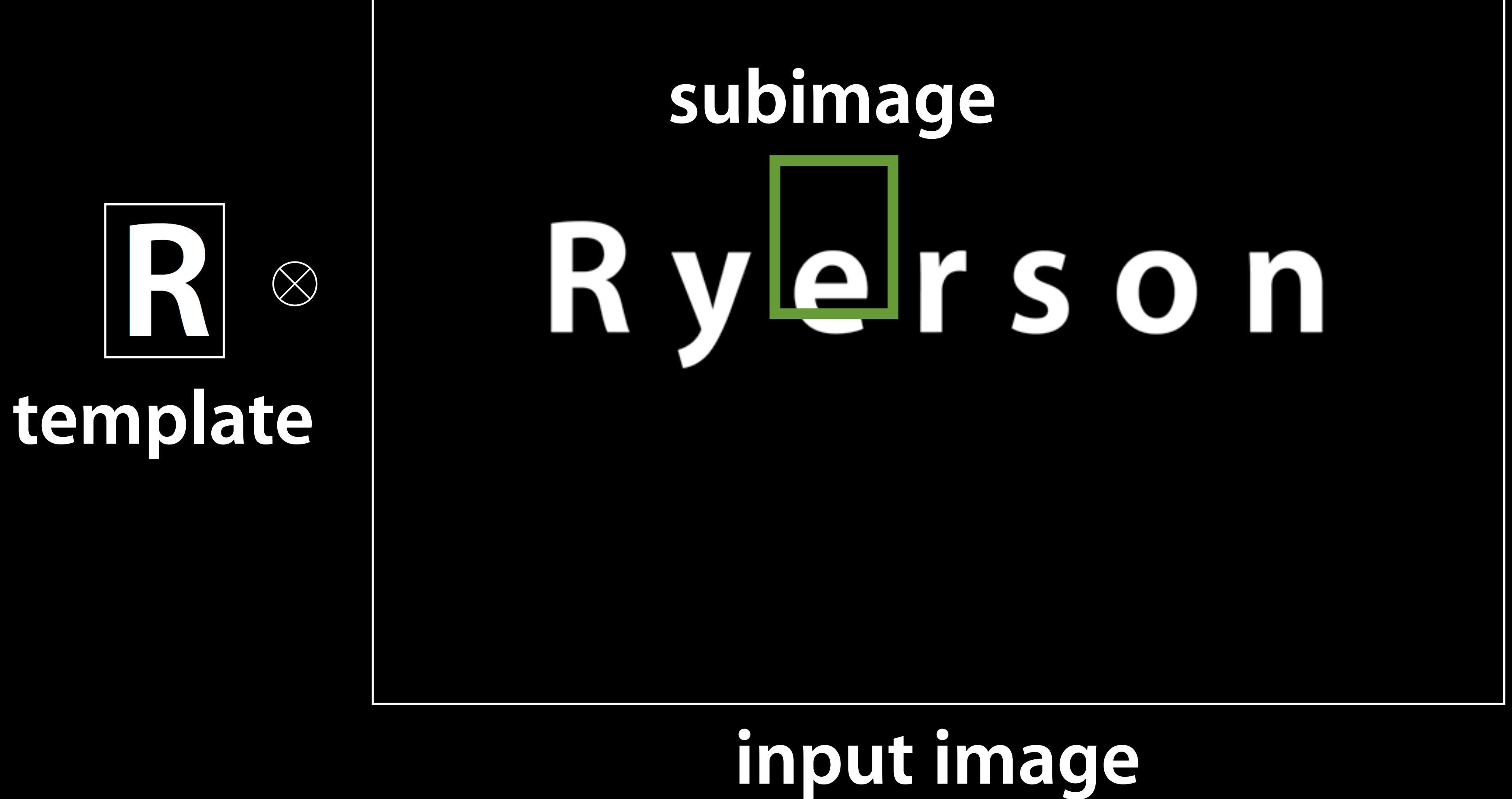
correlation score $= \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$

score is sensitive to multiplicative brightness scaling



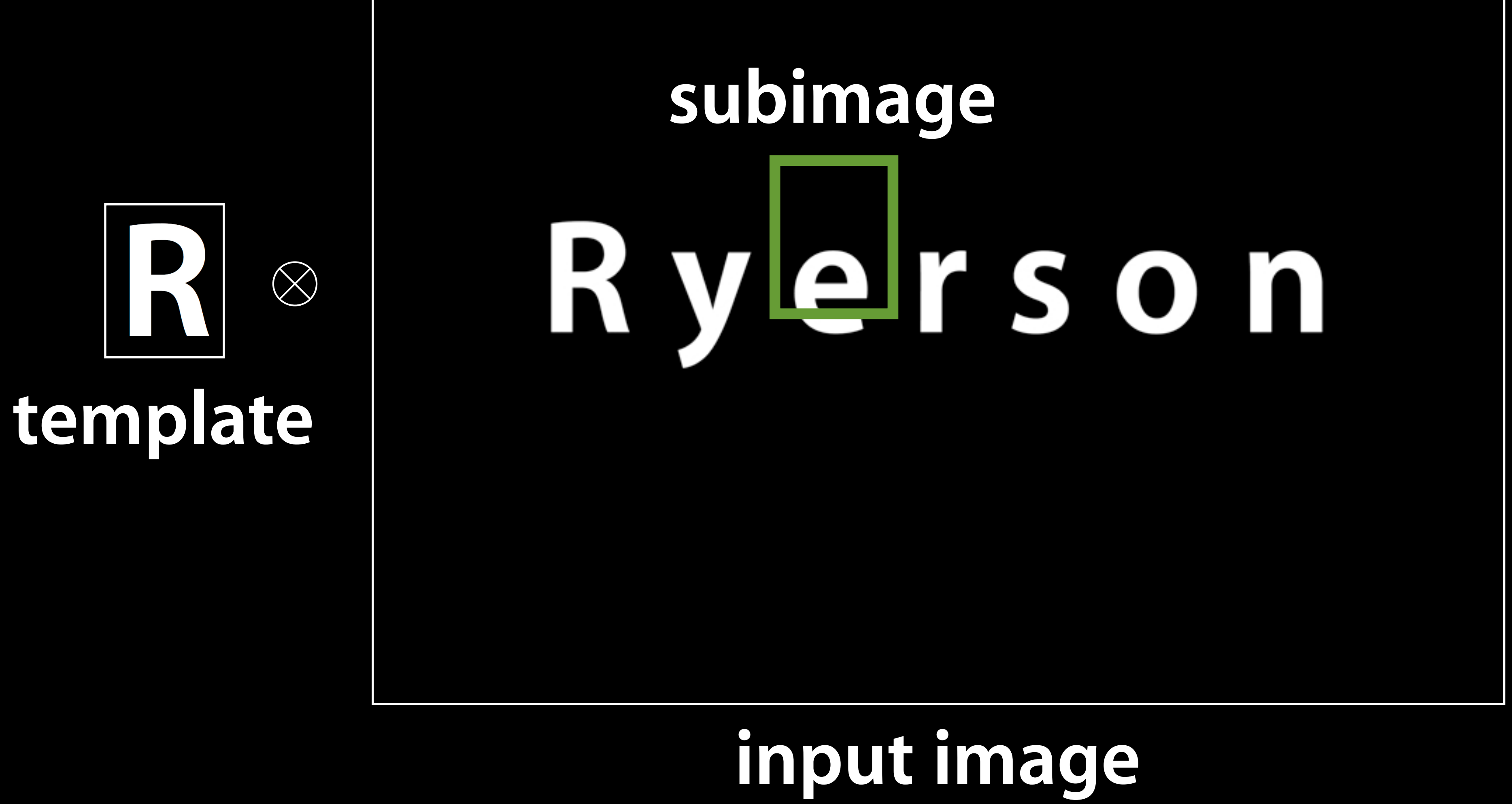
correlation score $= \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\|\|\mathbf{s}\| \cos\theta$

score is sensitive to multiplicative brightness scaling



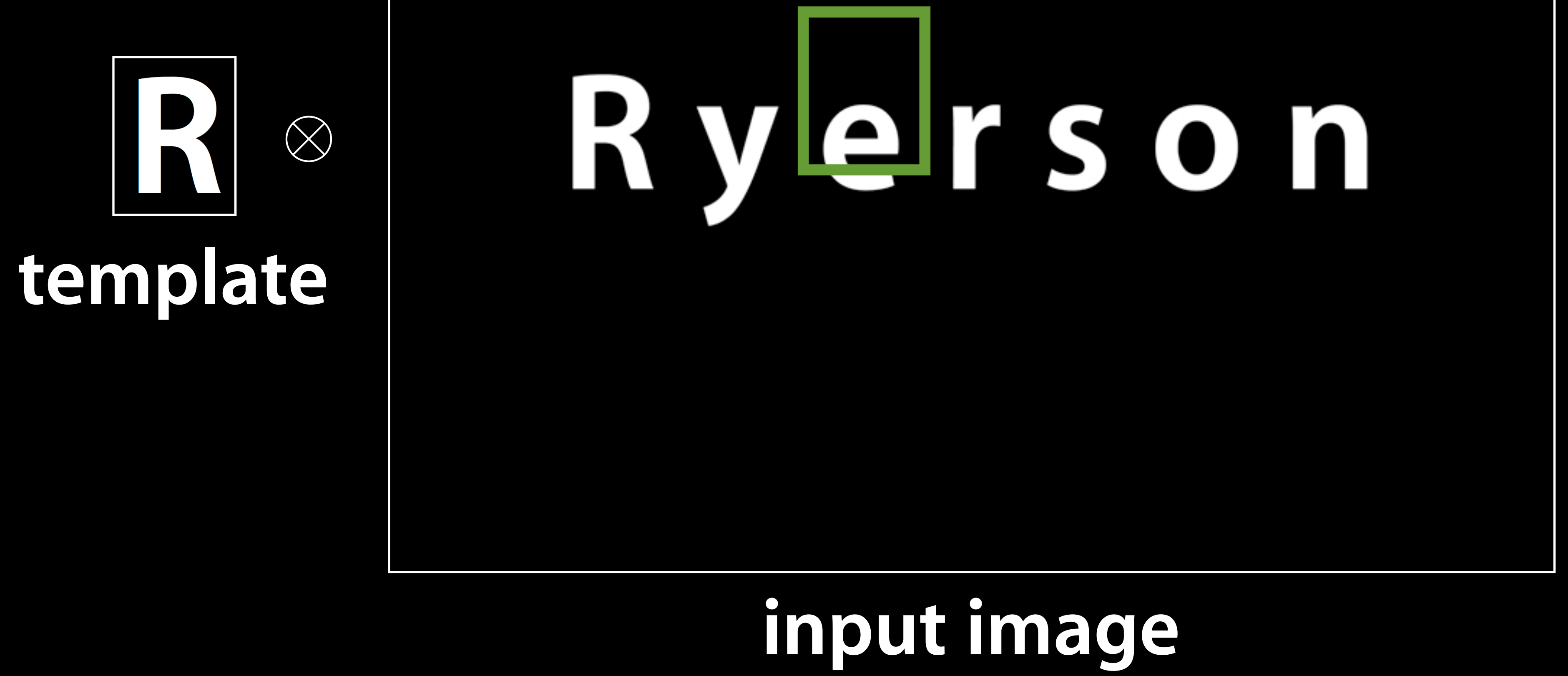
correlation score $= \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\|\|\mathbf{s}\| \cos\theta$

How can we make it invariant to brightness scaling?



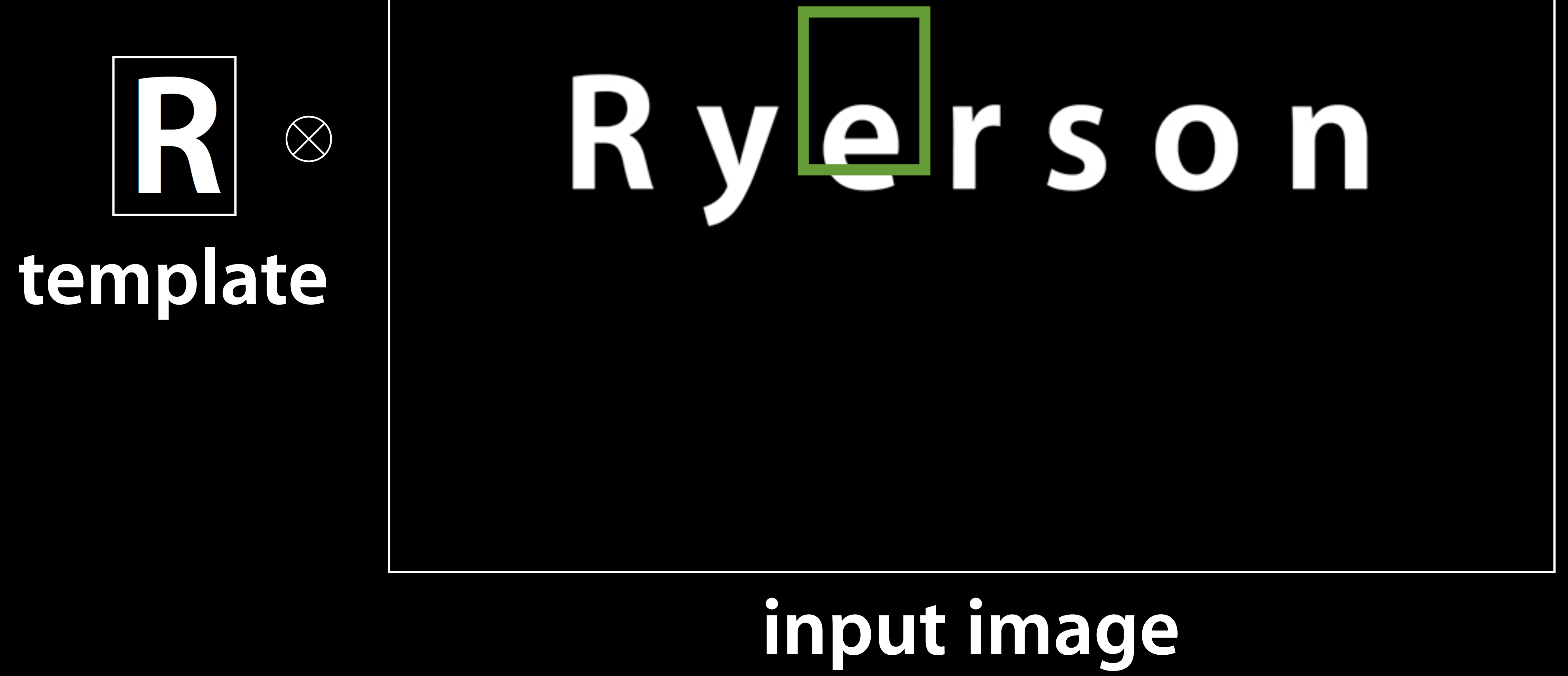
$$\textbf{correlation score} = \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$$

$$\textbf{normalized score} = \frac{\mathbf{t}}{\|\mathbf{t}\|} \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} = \cos\theta$$



$$\textbf{correlation score} = \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$$

$$\textbf{normalized score} = \frac{\mathbf{t}}{\|\mathbf{t}\|} \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} = \cos\theta$$



$$\textbf{correlation score} = \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$$

$$\textbf{normalized score} = \frac{\mathbf{t}}{\|\mathbf{t}\|} \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} = \cos\theta$$

How can we make it invariant to multiplicative and additive brightness change?

template

input image

correlation score $= \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$

normalized score $= \dfrac{\mathbf{t}}{\|\mathbf{t}\|} \cdot \dfrac{\mathbf{s}}{\|\mathbf{s}\|} = \cos\theta$

normalized cross-correlation $= \dfrac{(\mathbf{t} - \bar{\mathbf{t}})}{\|(\mathbf{t} - \bar{\mathbf{t}})\|} \cdot \dfrac{(\mathbf{s} - \bar{\mathbf{s}})}{\|(\mathbf{s} - \bar{\mathbf{s}})\|}$

template

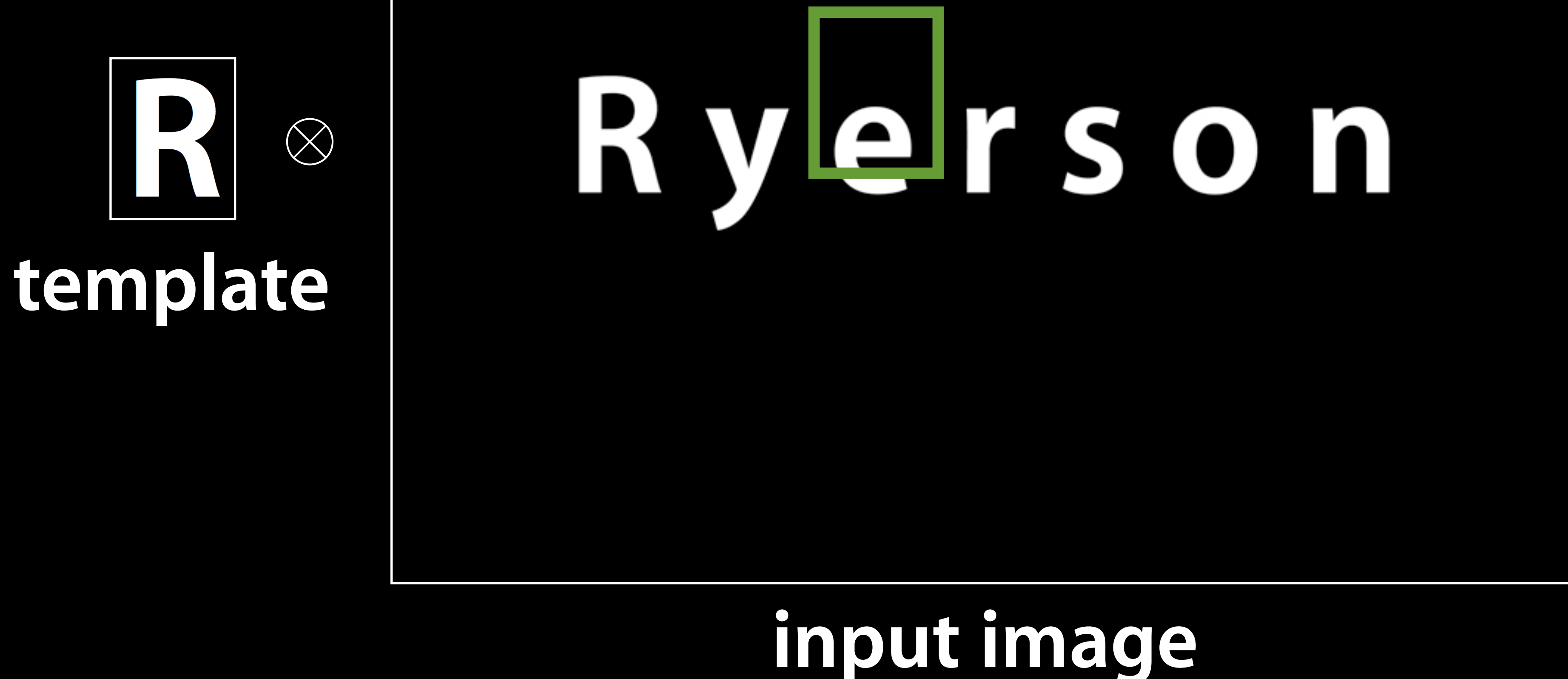
input image

correlation score $= \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$

normalized score $= \dfrac{\mathbf{t}}{\|\mathbf{t}\|} \cdot \dfrac{\mathbf{s}}{\|\mathbf{s}\|} = \cos\theta$

normalized cross-correlation $= \dfrac{(\mathbf{t} - \bar{\mathbf{t}})}{\|(\mathbf{t}} \quad \dfrac{(\mathbf{s} - \bar{\mathbf{s}})}{}$

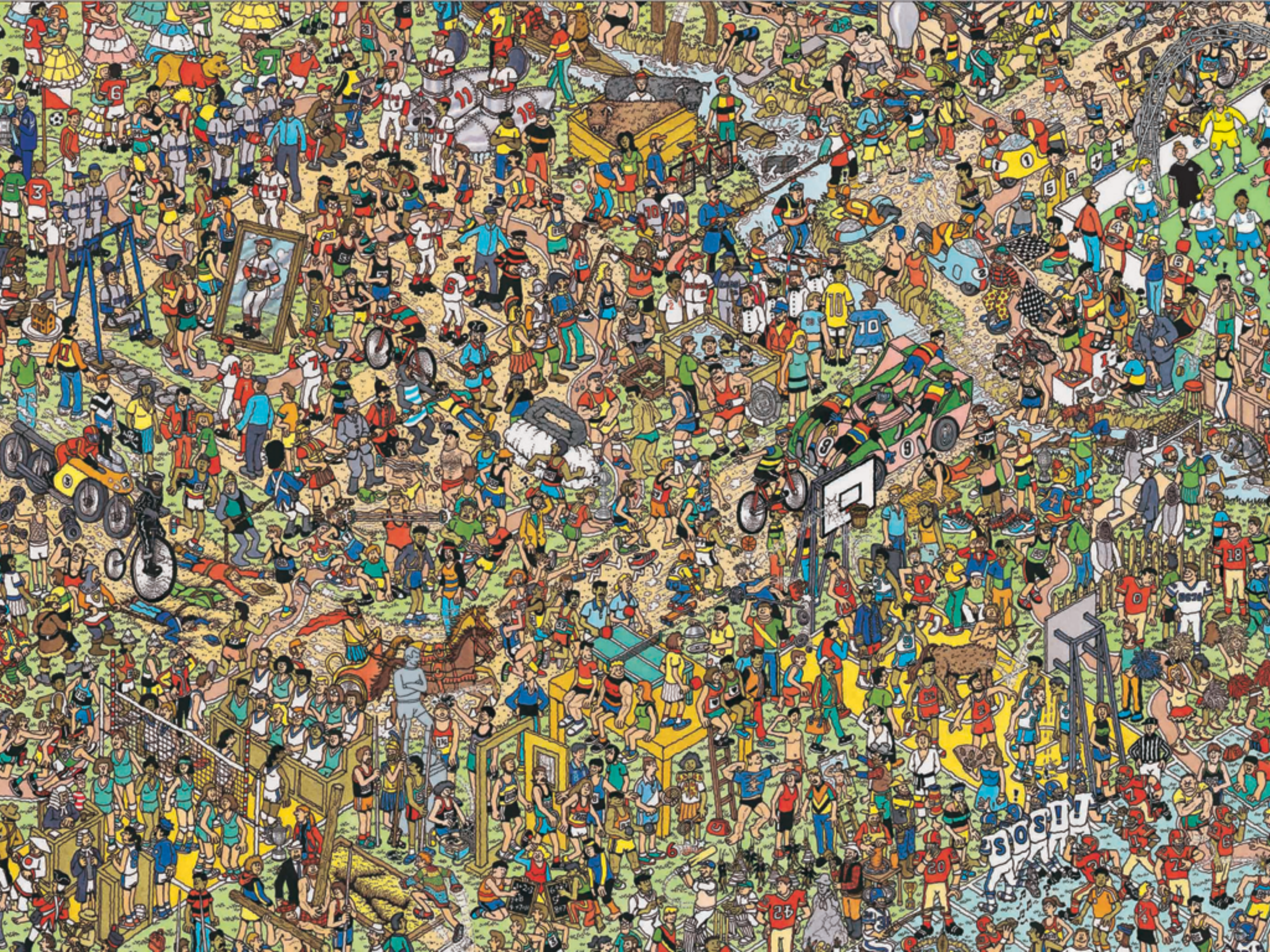
average brightness



$$\textbf{correlation score} = \mathbf{t} \cdot \mathbf{s} = \|\mathbf{t}\| \|\mathbf{s}\| \cos\theta$$

$$\textbf{normalized score} = \frac{\mathbf{t}}{\|\mathbf{t}\|} \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|} = \cos\theta$$

$$\textbf{normalized cross-correlation} = \frac{(\mathbf{t} - \bar{\mathbf{t}})}{\|(\mathbf{t} - \bar{\mathbf{t}})\|} \cdot \frac{(\mathbf{s} - \bar{\mathbf{s}})}{\|(\mathbf{s} - \bar{\mathbf{s}})\|}$$



Where’s Waldo?


3
6
7
2
+
1
=
3


Normalized Cross-Correlation sliding window matching


Normalized Cross-Correlation sliding window matching

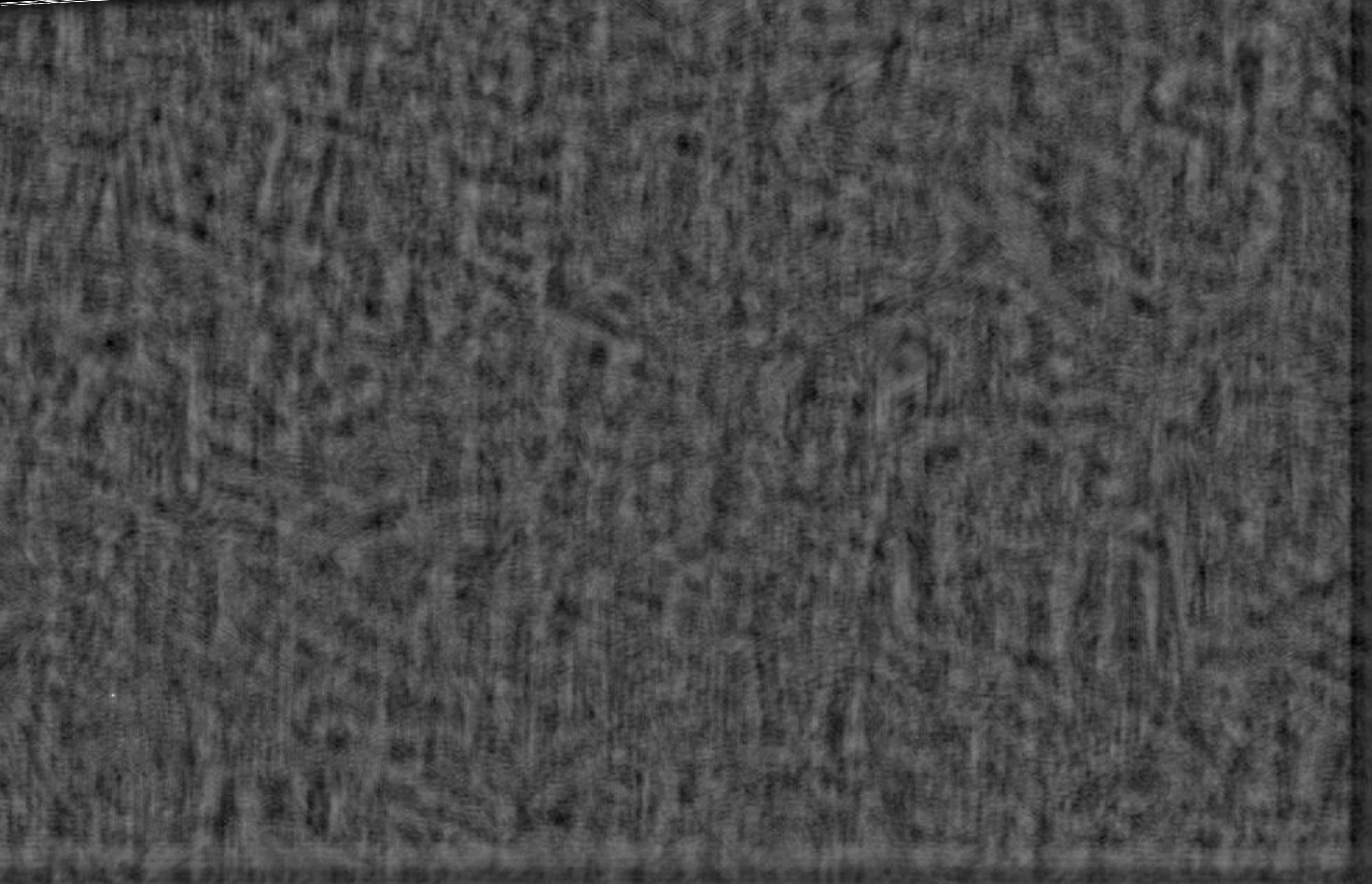

Normalized Cross-Correlation sliding window matching


Normalized Cross-Correlation sliding window matching

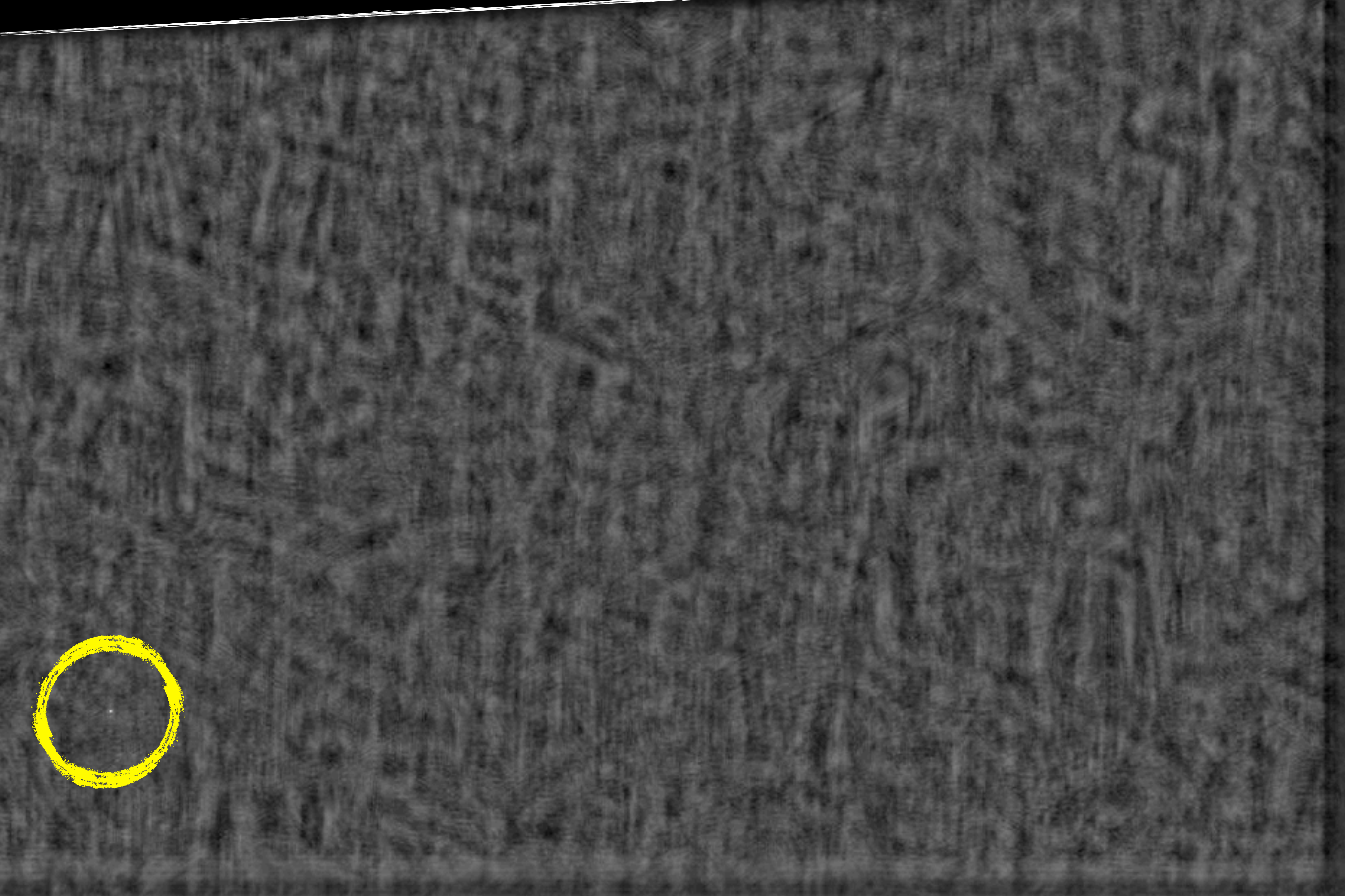

Normalized Cross-Correlation sliding window matching


Normalized Cross-Correlation sliding window matching

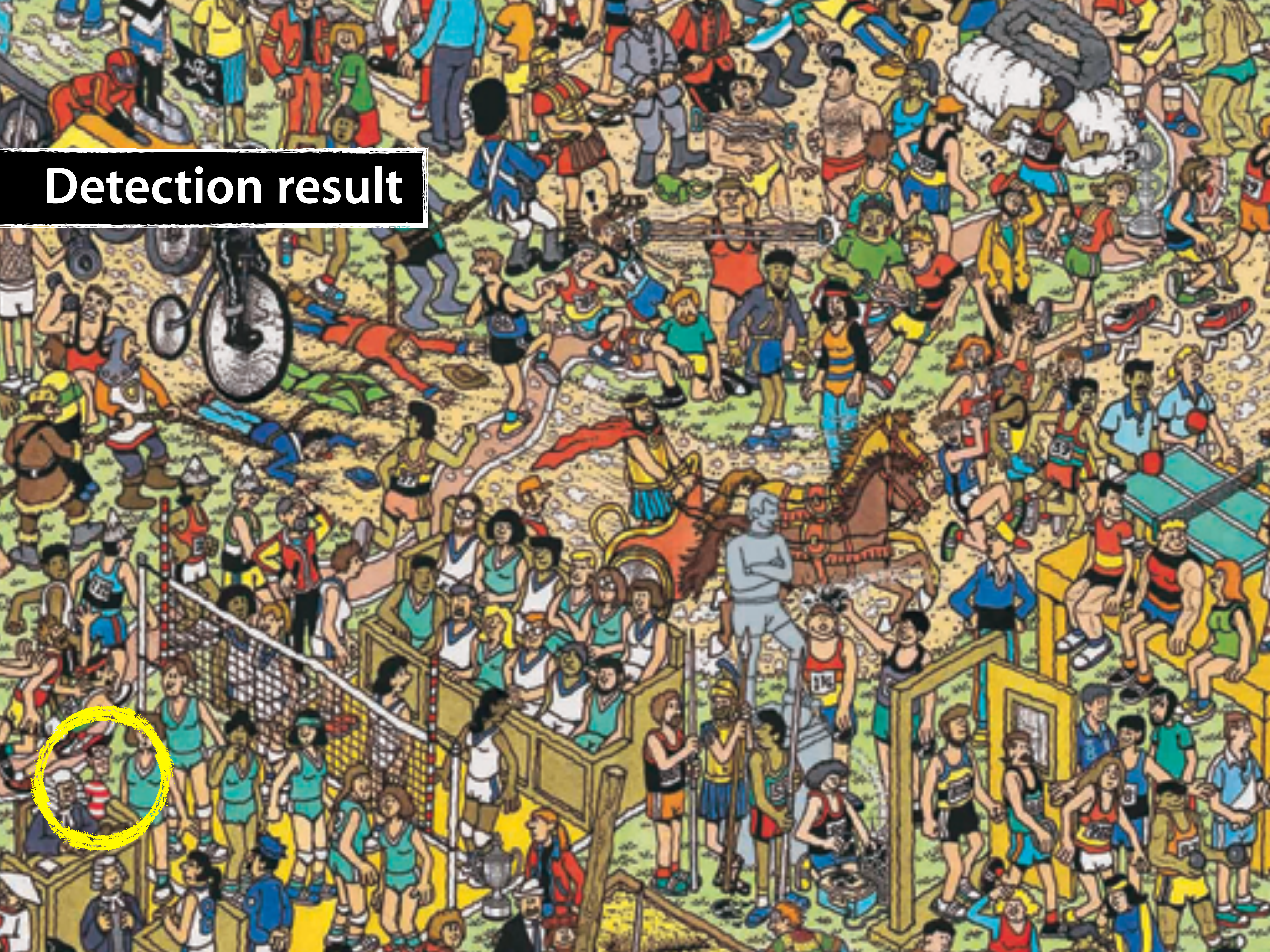
normalized cross-correlation result

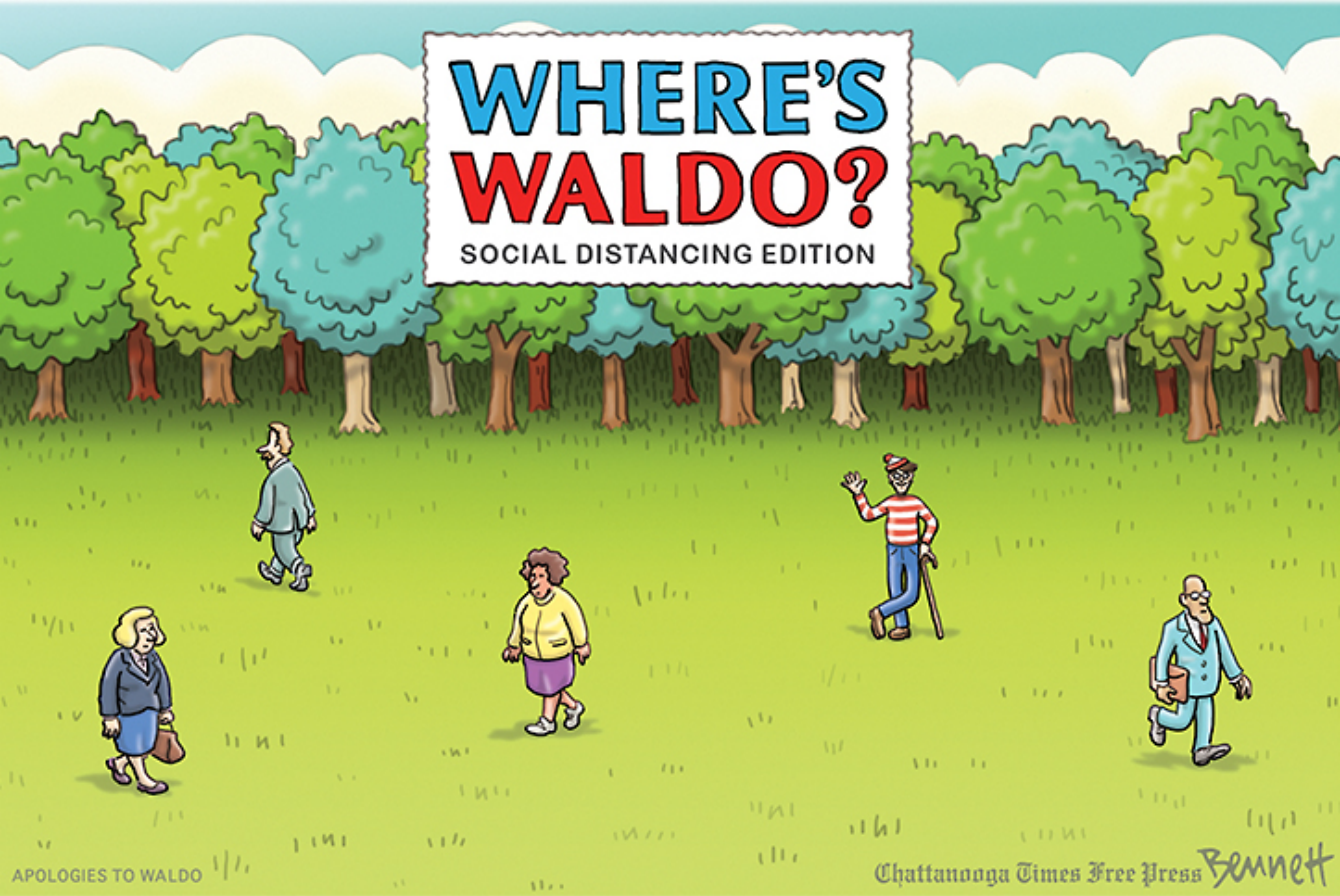


normalized cross-correlation result




Detection result


Detection result


WHERE'S WALDO?
SOCIAL DISTANCING EDITION
APOLOGIES TO WALDO
Chattanooga Times Free Press
Bennett


What if the template and image are not identical?

Varying instances

Varying instances


Varying SCALES

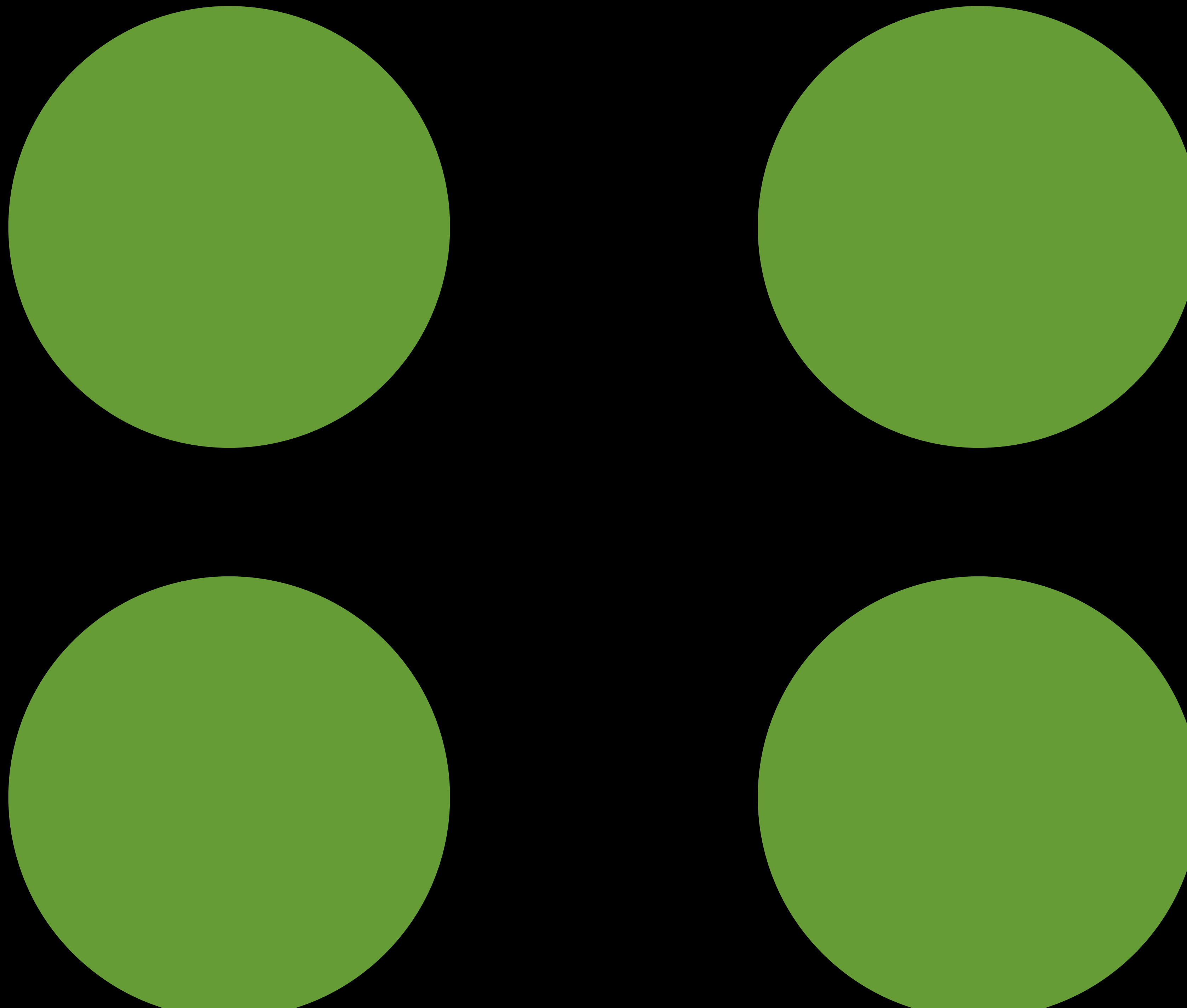

Highly articulated


Highly articulated

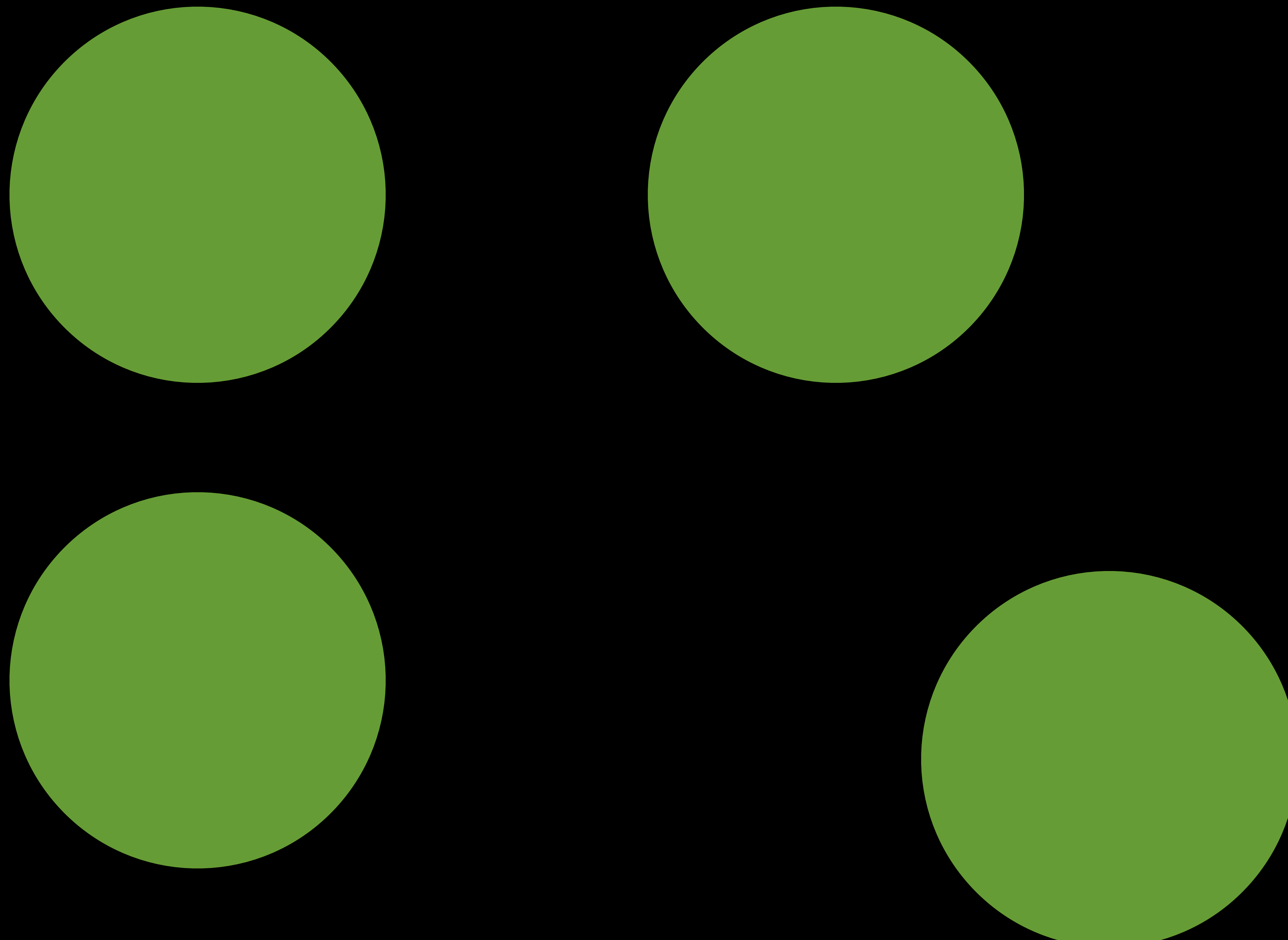
contrast

“Without **contrast** you’re dead.”

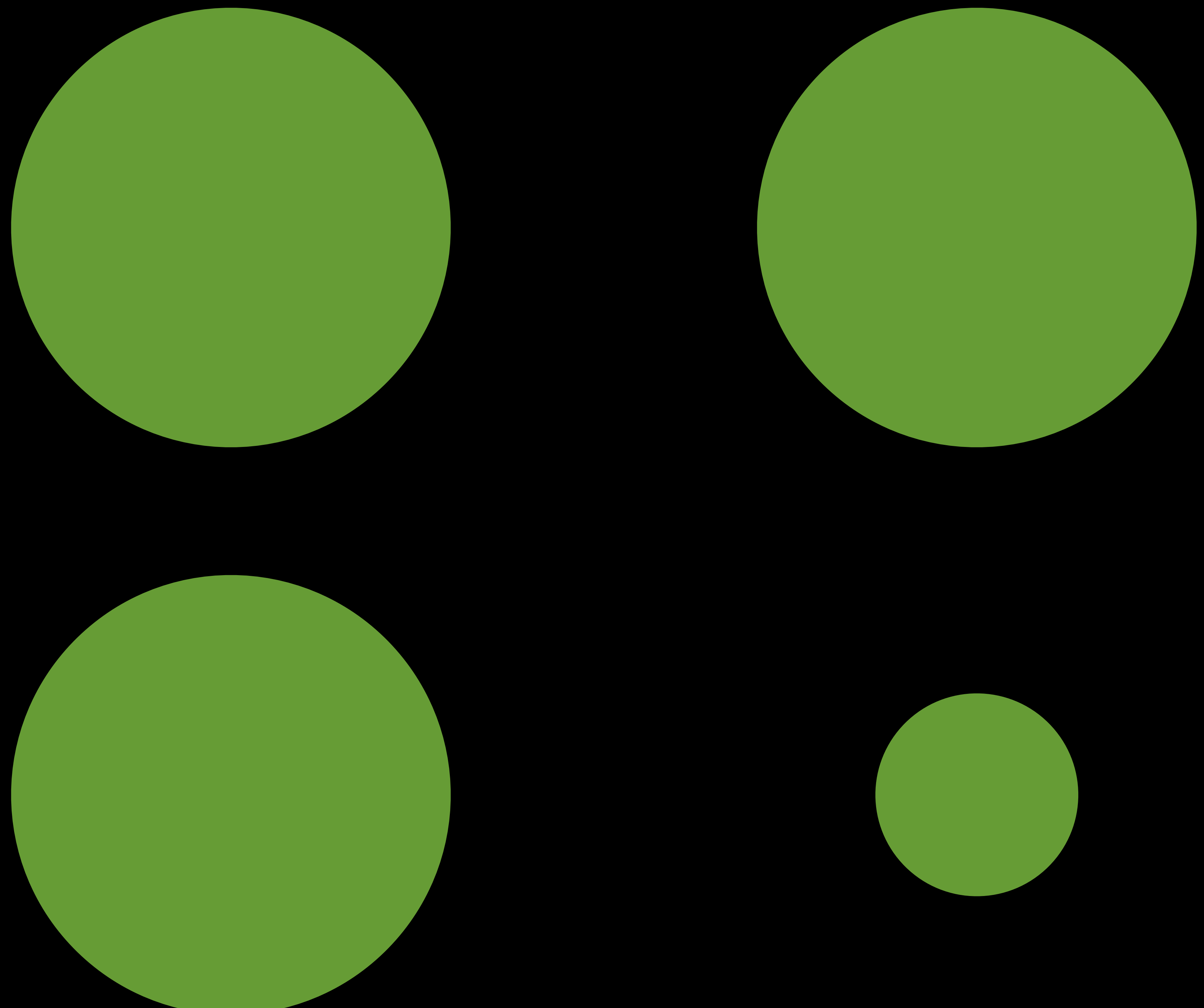
— Paul Rand



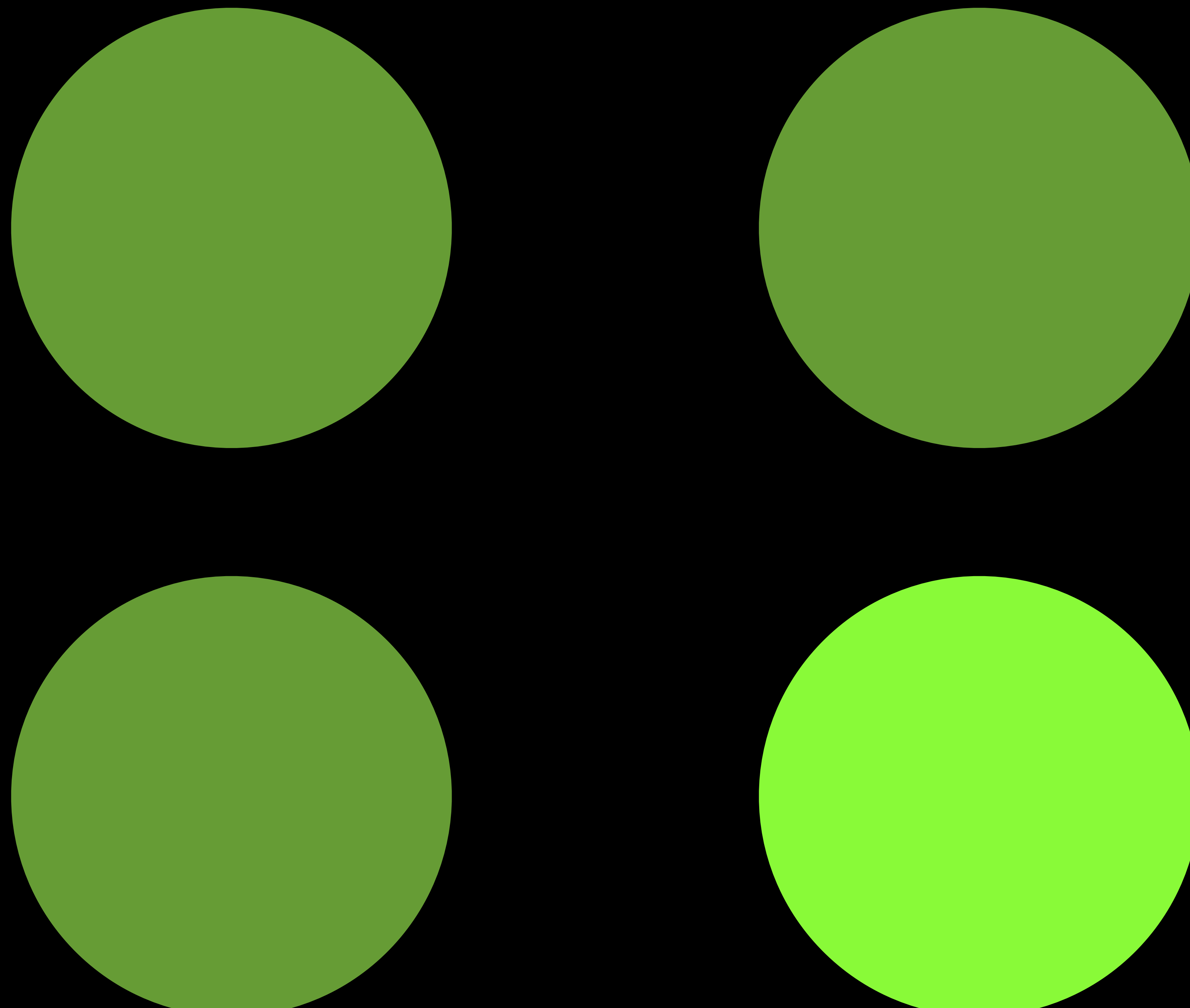
contrast in position



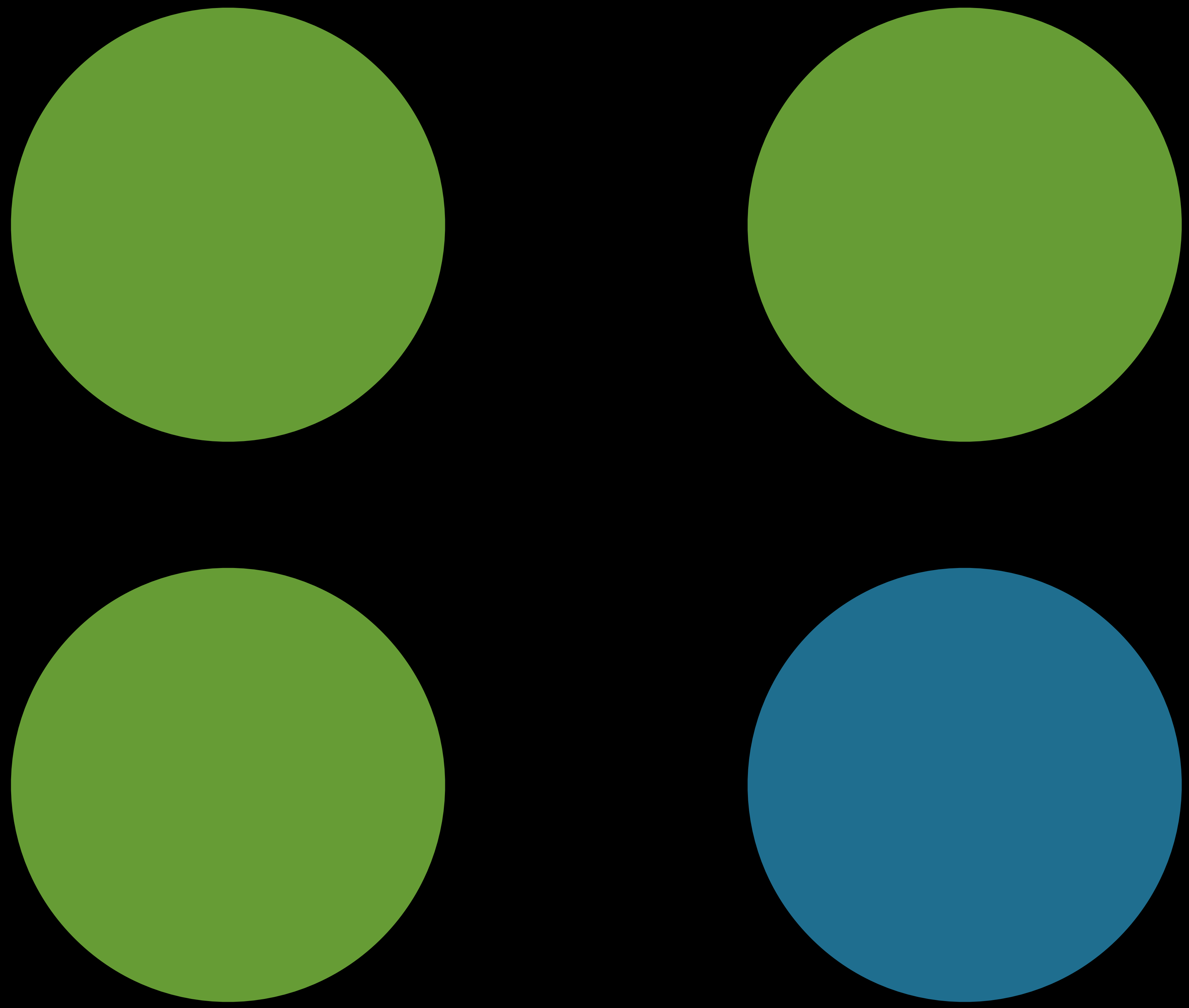
contrast in size



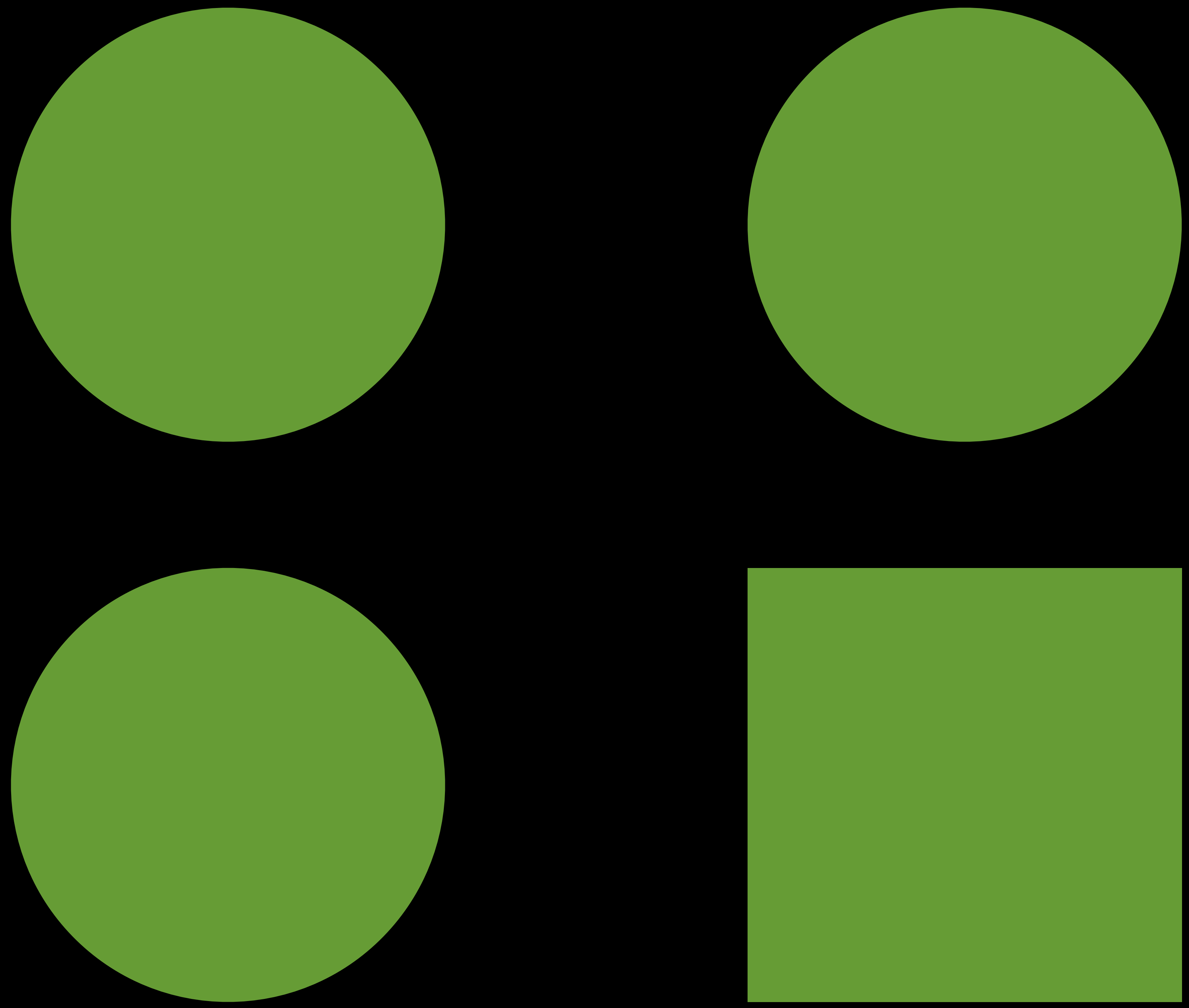
contrast in brightness



contrast in color

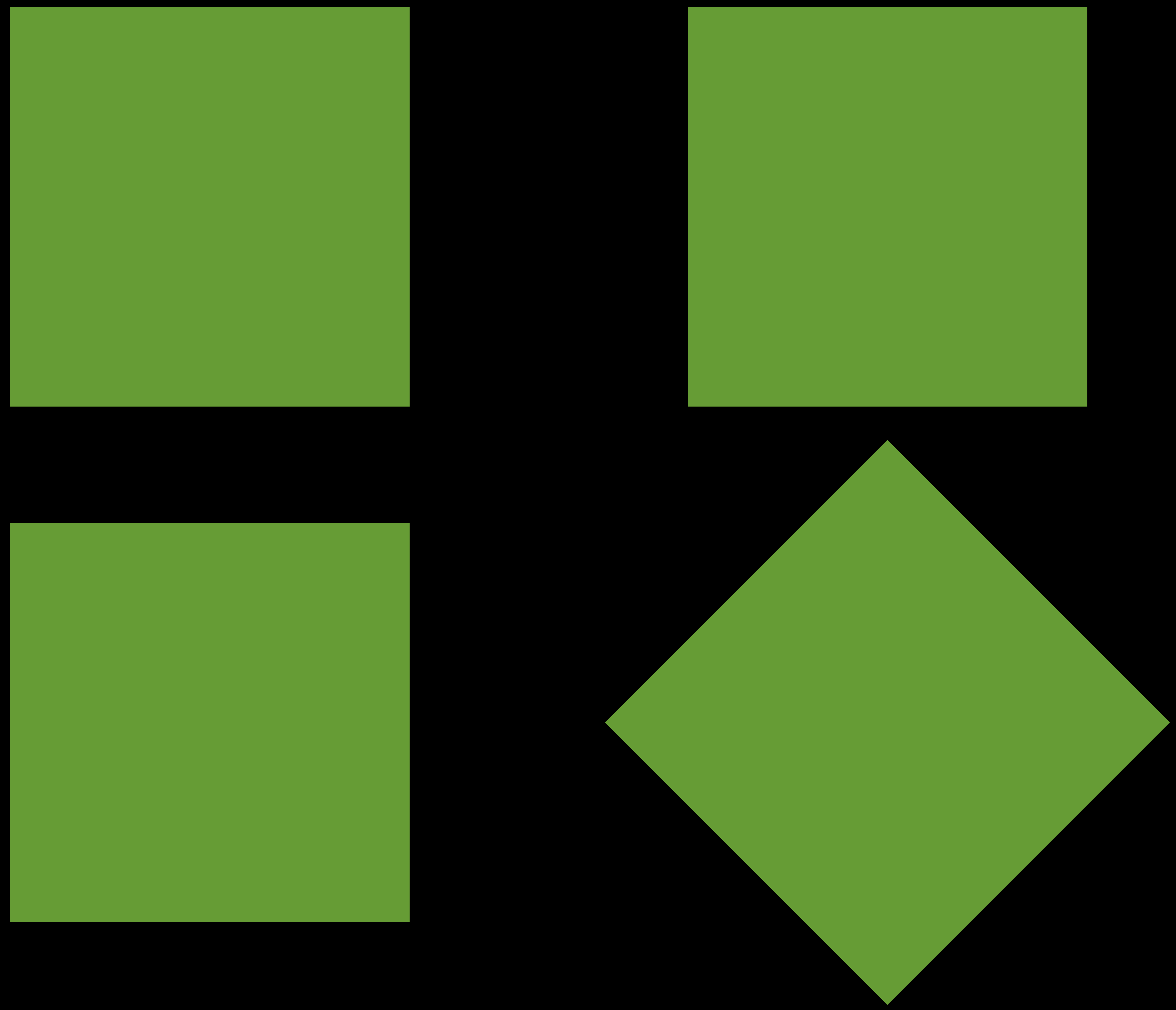


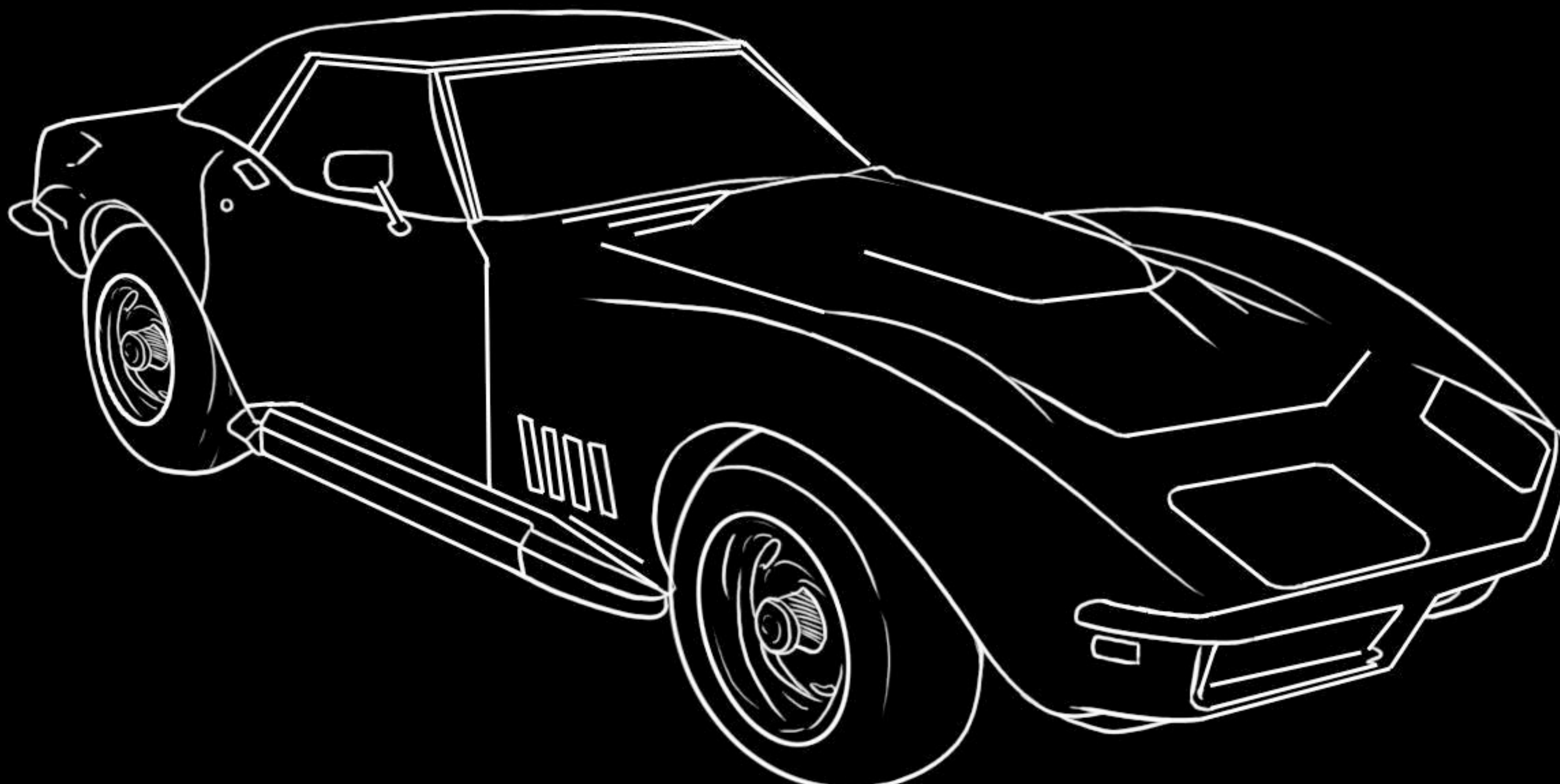
contrast in shape

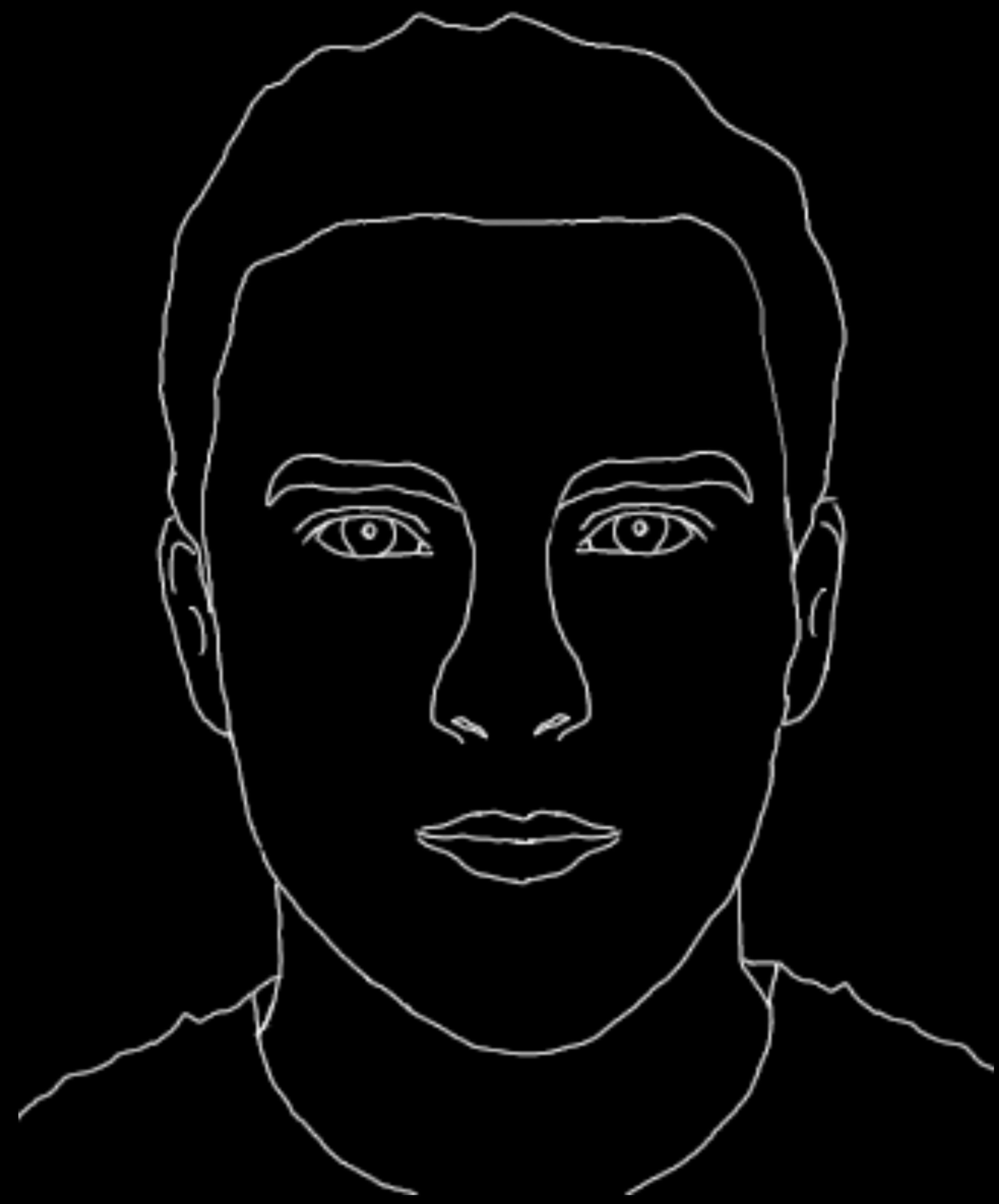


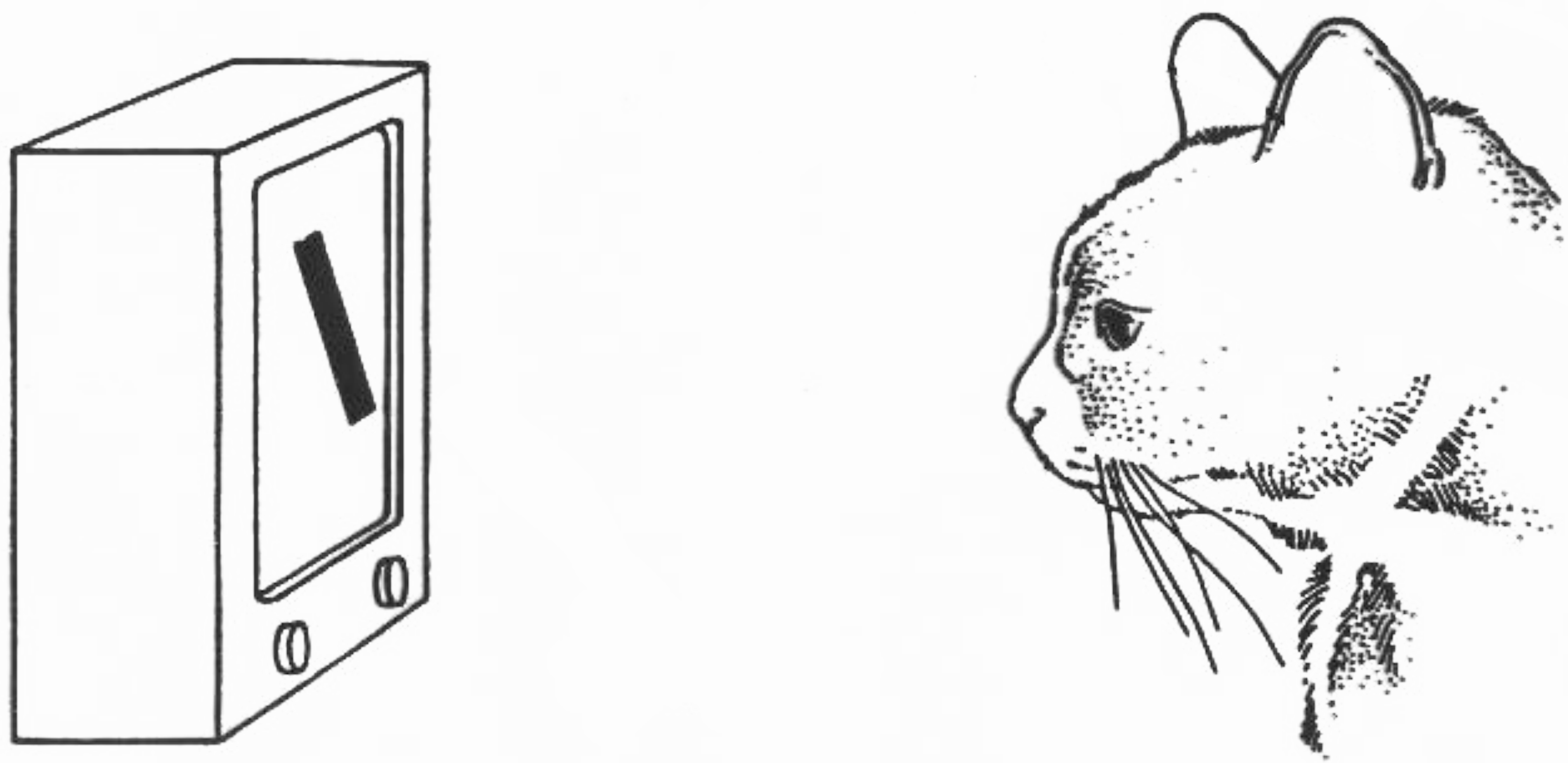
contrast in orientation

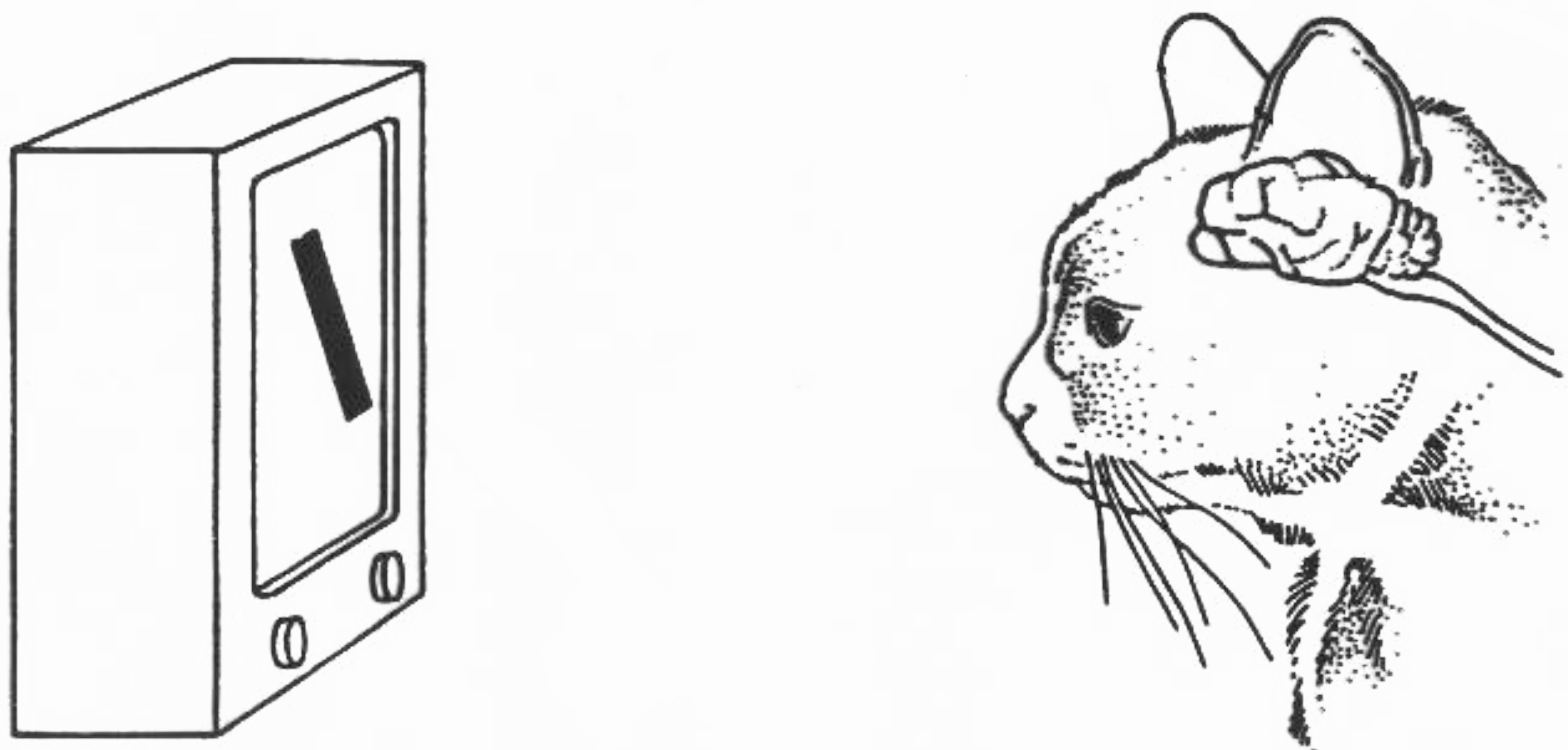
contrast in orientation

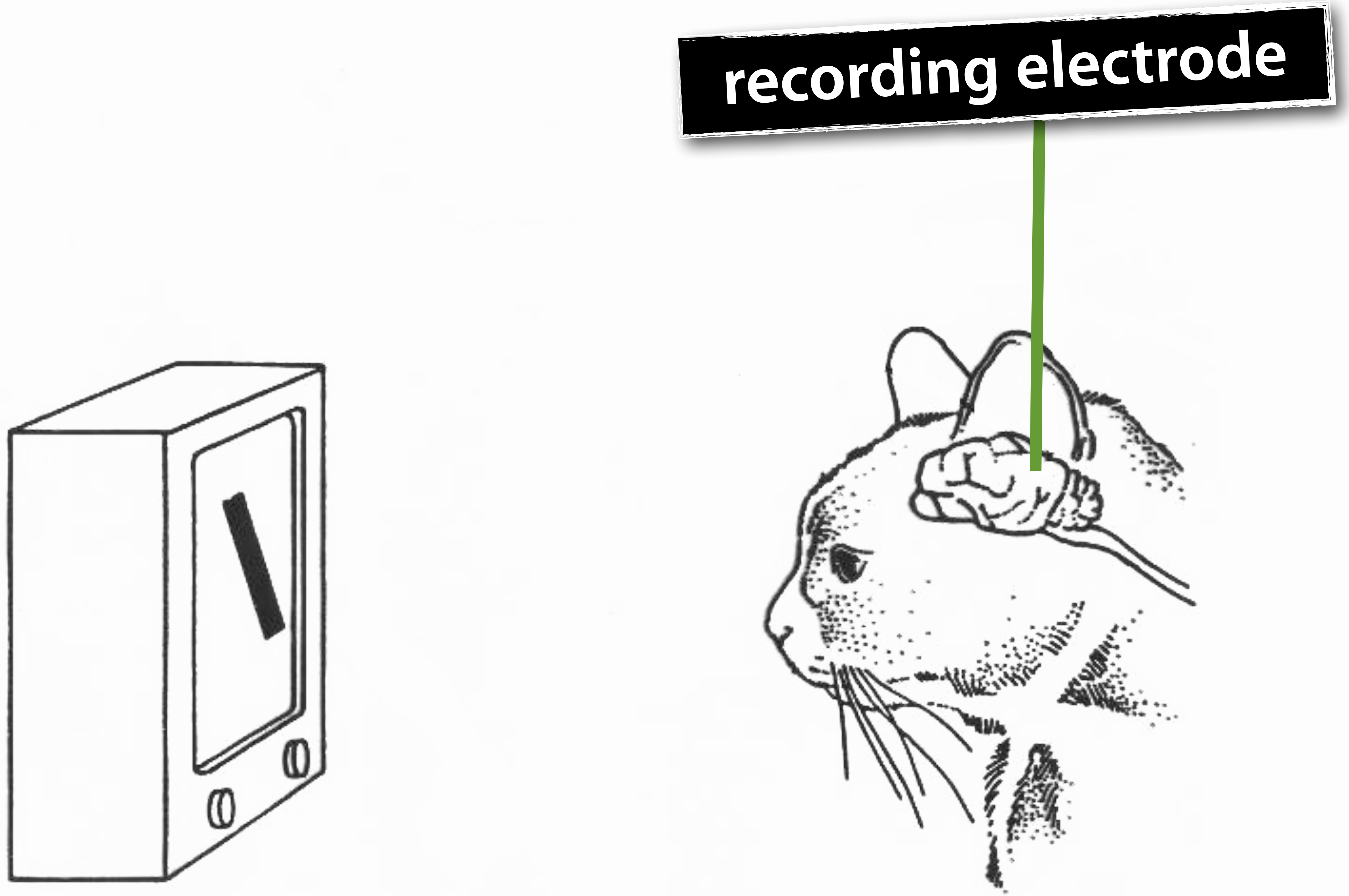


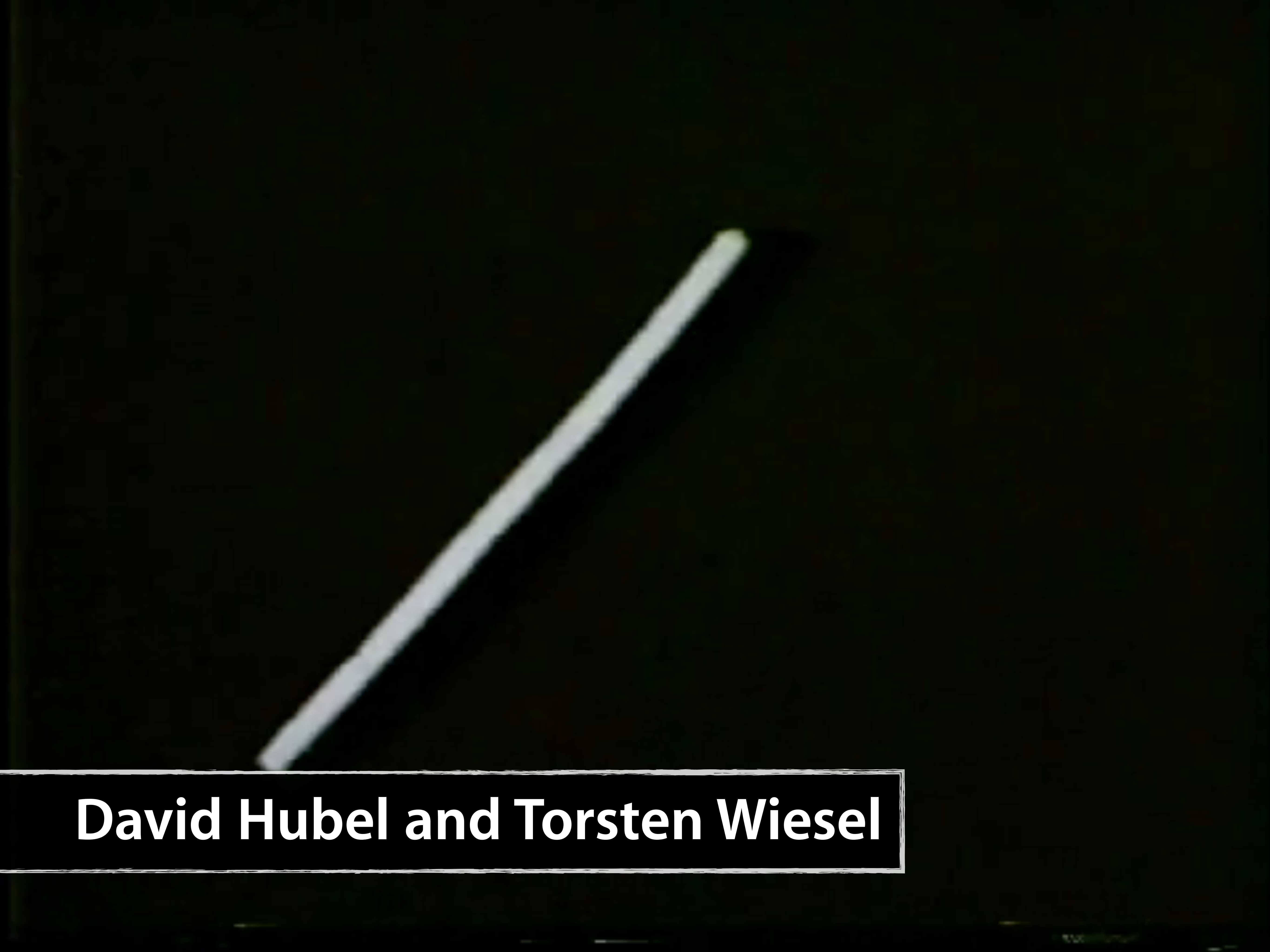


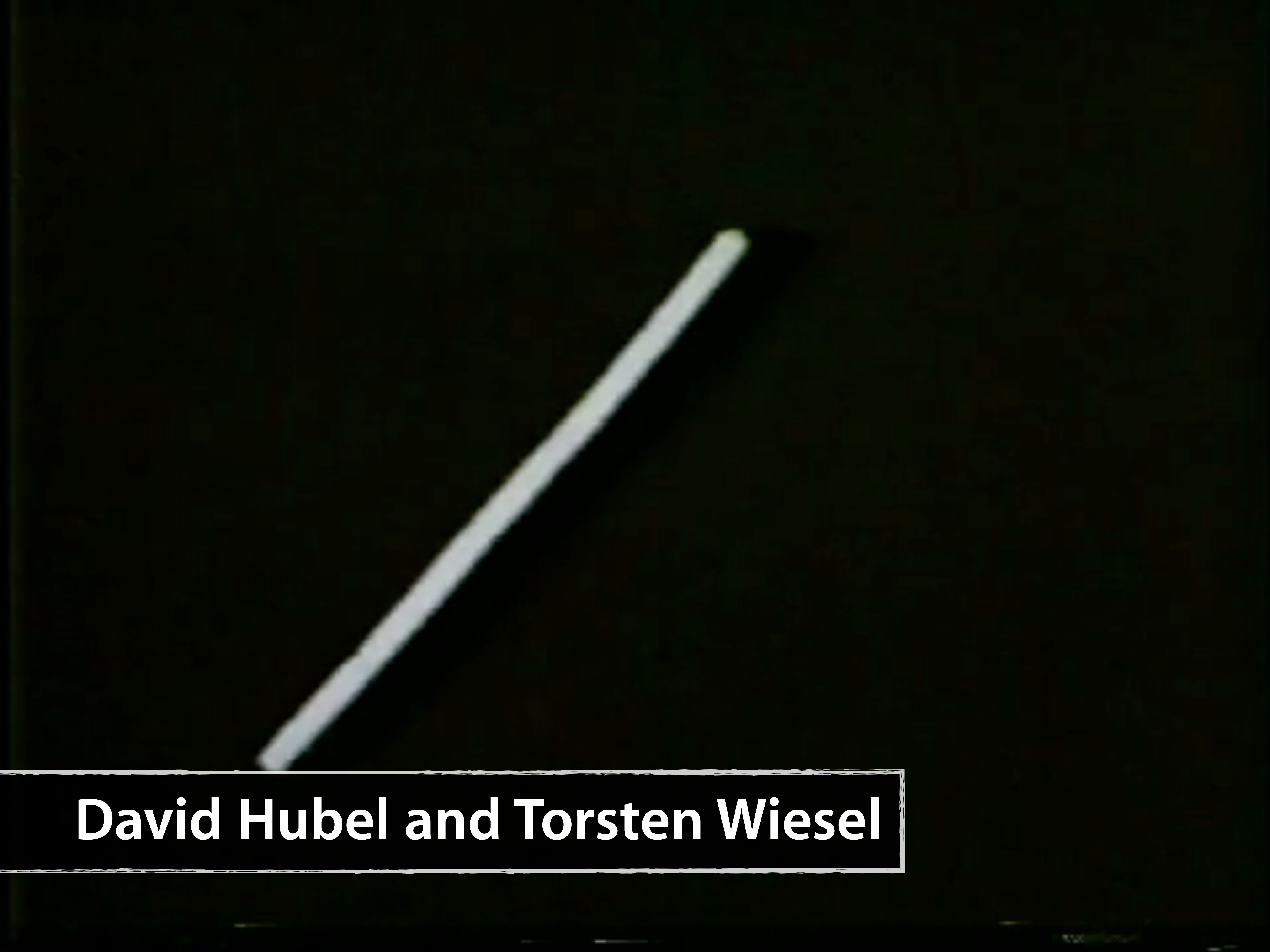







recording electrode


David Hubel and Torsten Wiesel



David Hubel and Torsten Wiesel


SHERIFF'S OFFICE
STATE LIBRARY ARCHIVES


SHERIFF'S
OFFICE
STATE
LIBRARY
ARCHIVES


SHERIFF'S
OFFICE
STATE
LIBRARY
ARCHIVES


SHERIFF'S
OFFICE
STATE
LIBRARY
ARCHIVES
reflectance change


depth discontinuity

cast shadows


surface orientation change


SHERIFF'S
OFFICE
STATE
LIBRARY
ARCHIVES

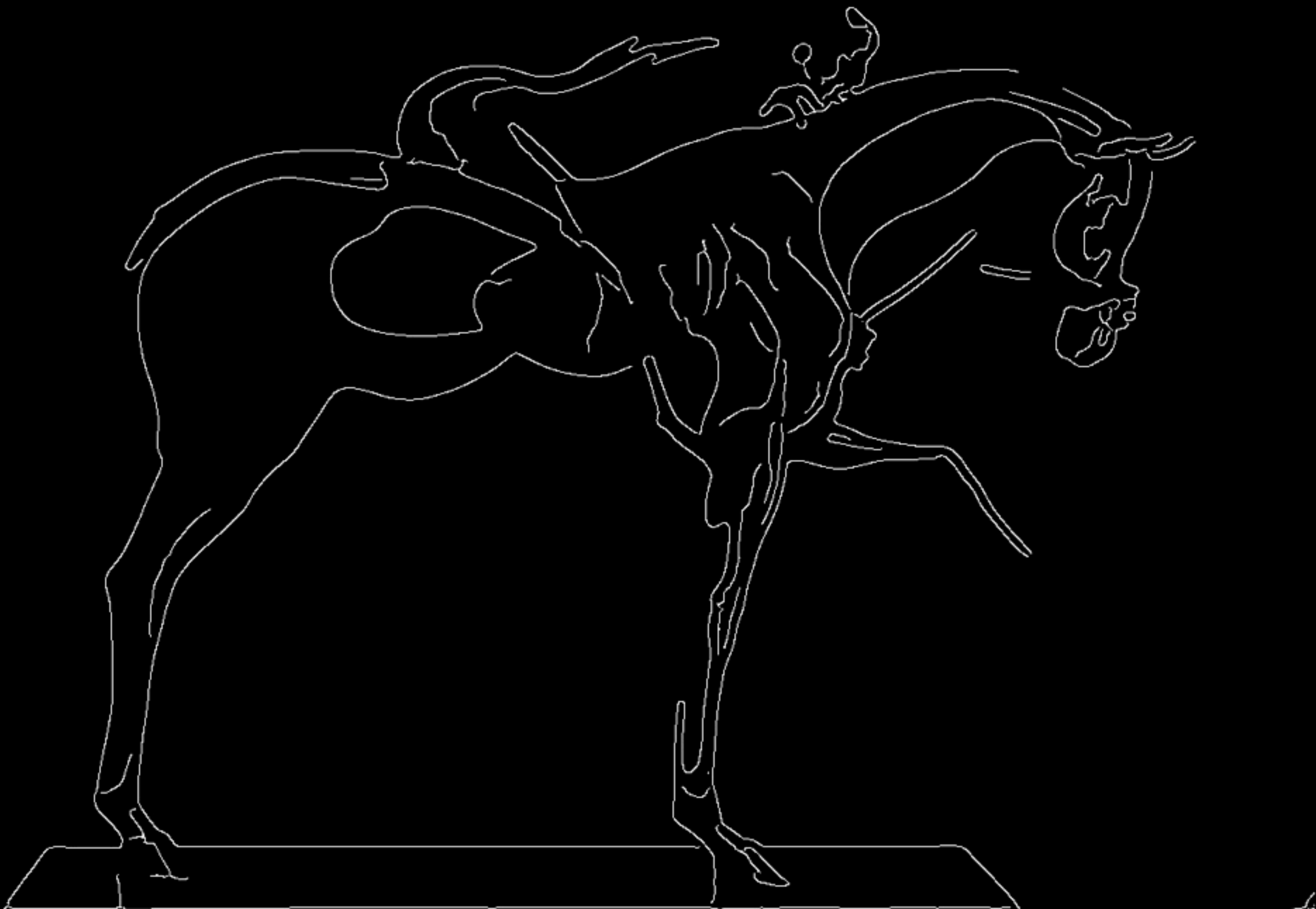

SHERIFF'S
OFFICE
STATE
LIBRARY
ARCHIVES

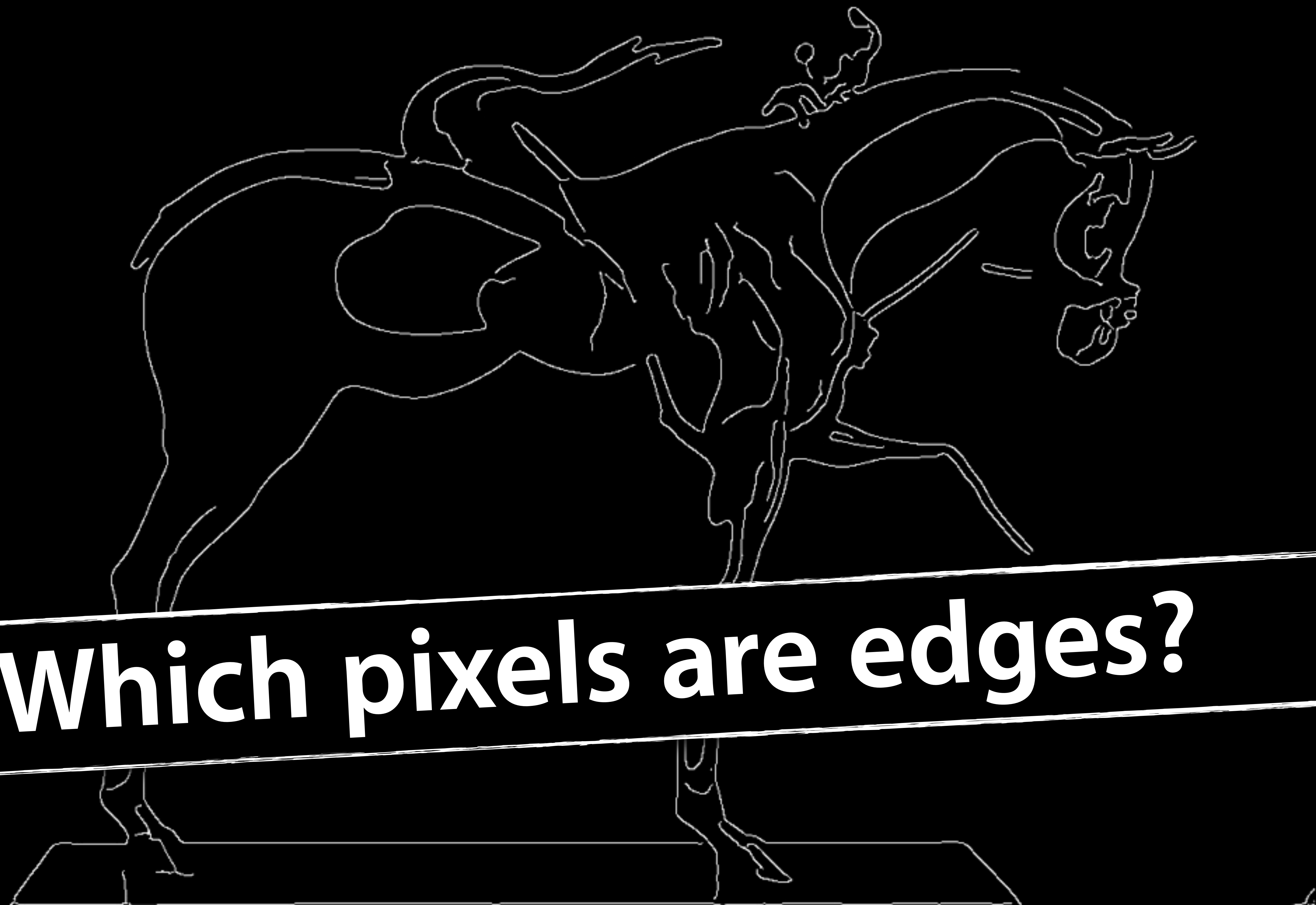

OFFICE
STATE
LIBRARY


uniform
SHERIFF'S OFFICE
STATE LIBRARY ARCHIVES

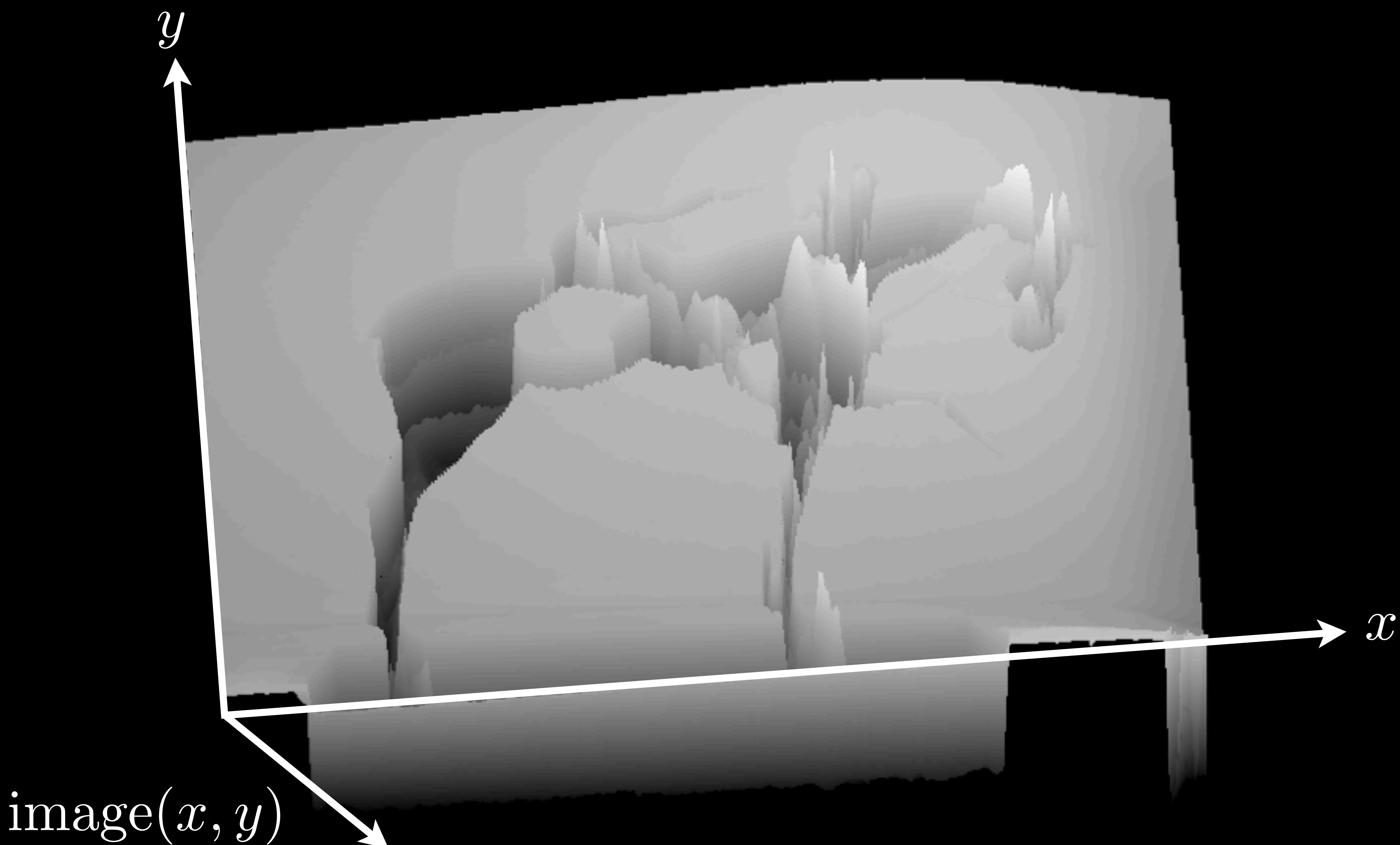

uniform


uniform
contrast

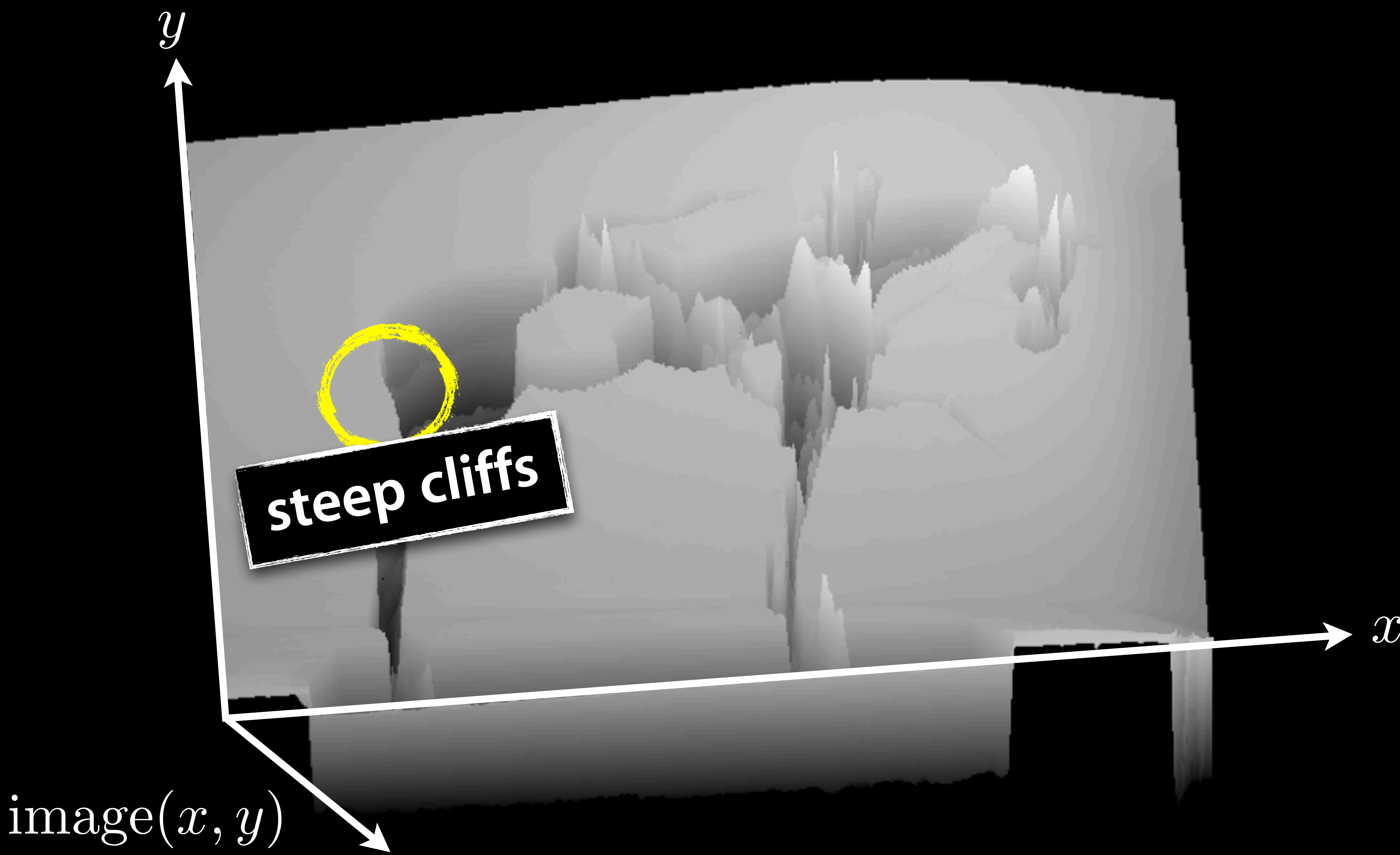




Which pixels are edges?

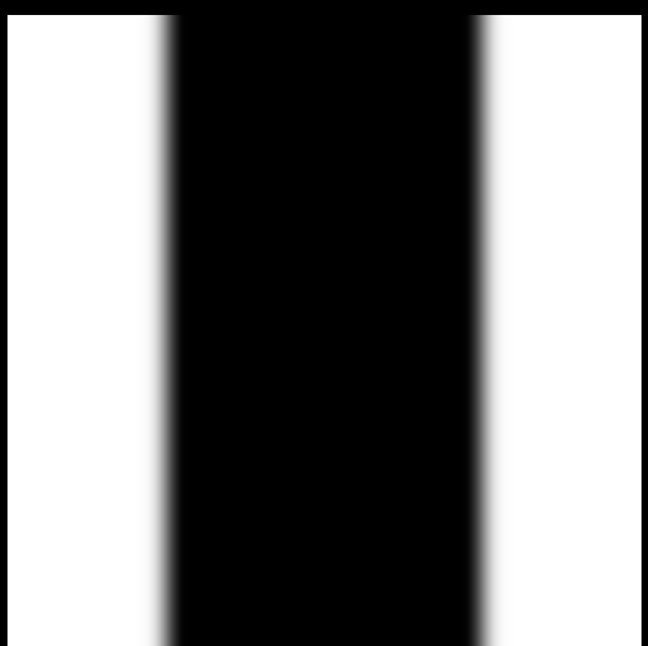


y
x
image(x, y)



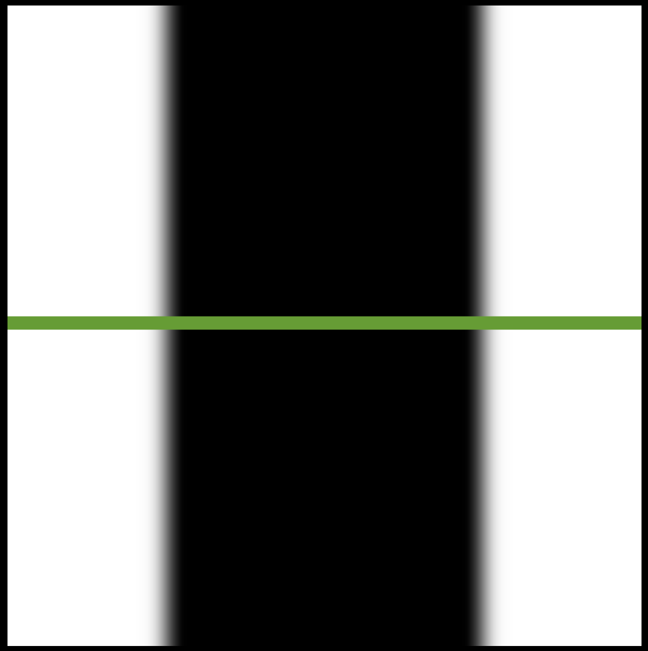
y
steep cliffs
x
$\mathrm{image}(x, y)$

Goal: Identify large local changes in the image



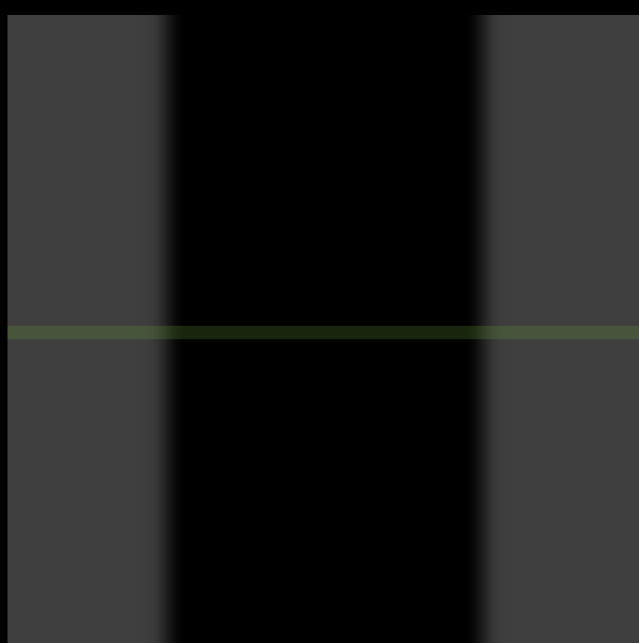
image

Edge Detection

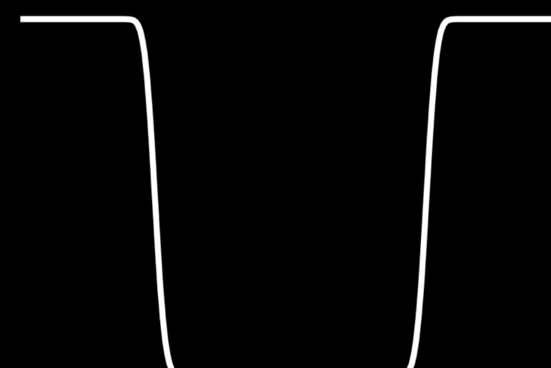


image

Edge Detection

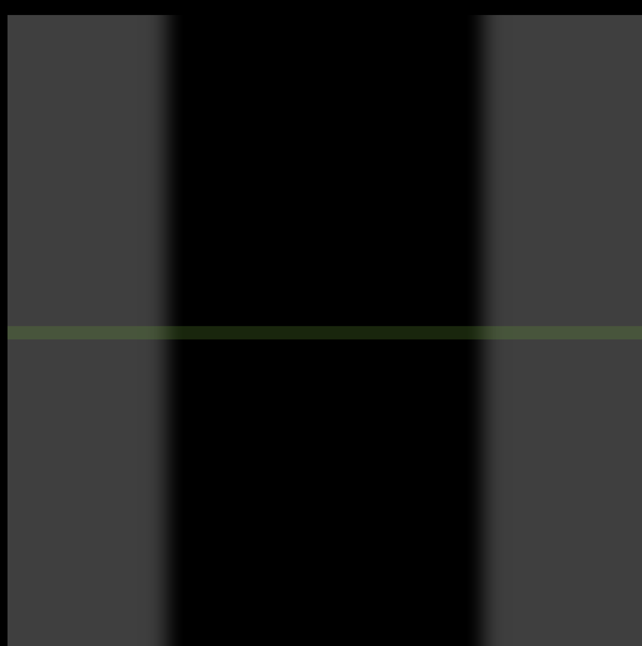


image

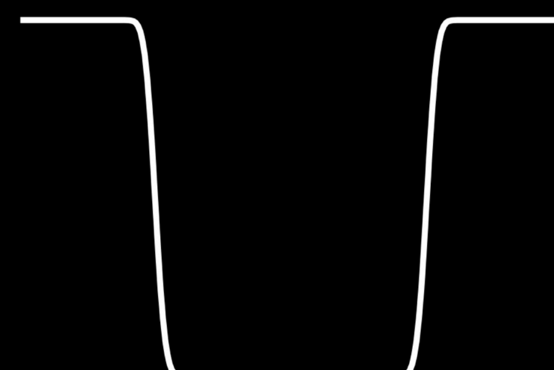


intensity function (slice)

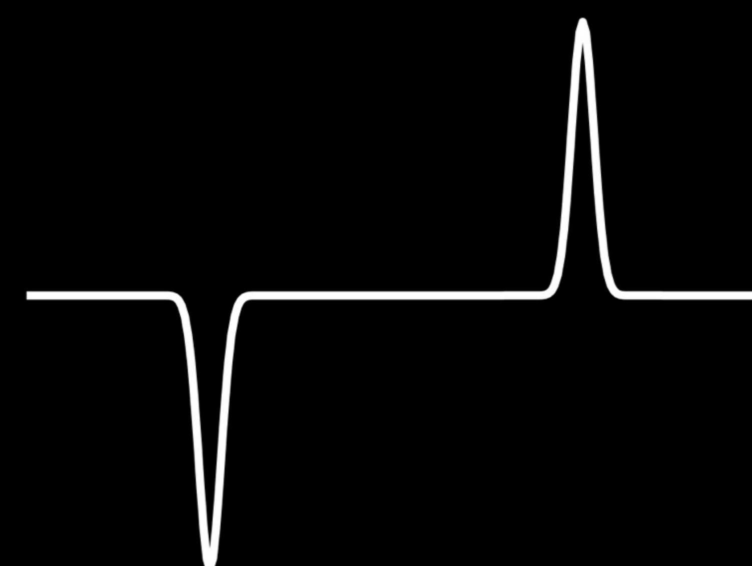
Edge Detection



image

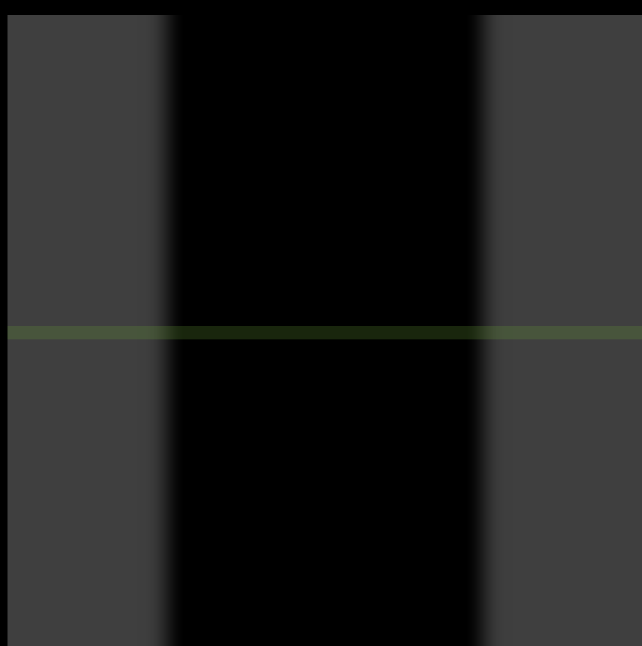


intensity function (slice)

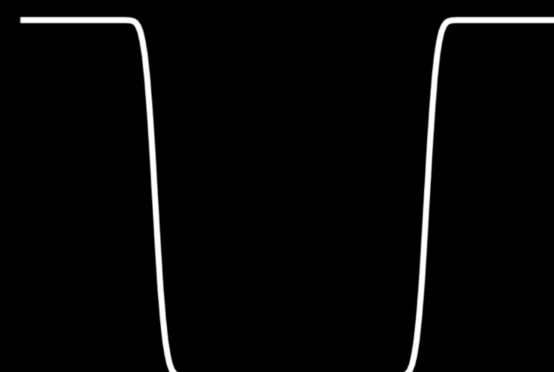


first derivative

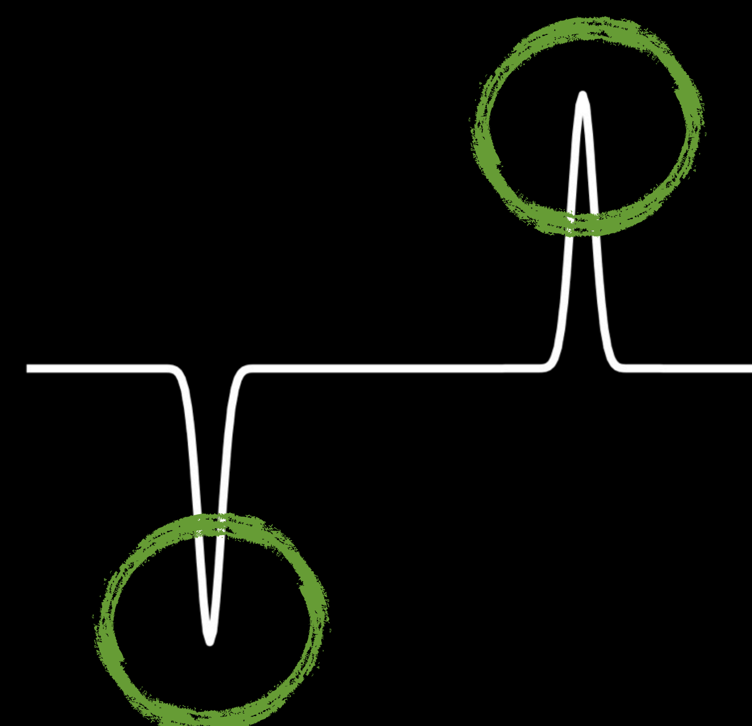
Edge Detection



image

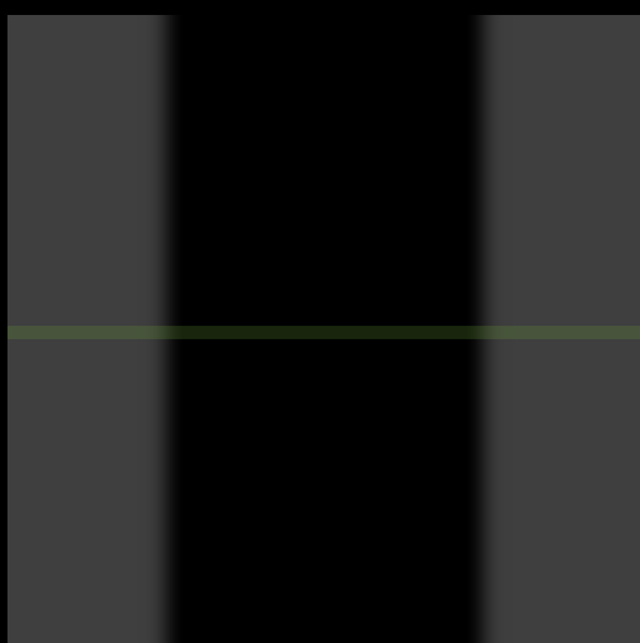


intensity function (slice)

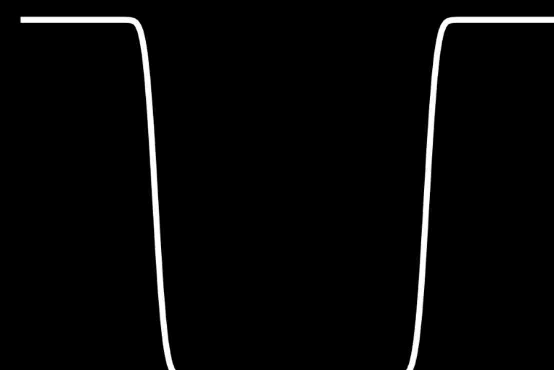


first derivative

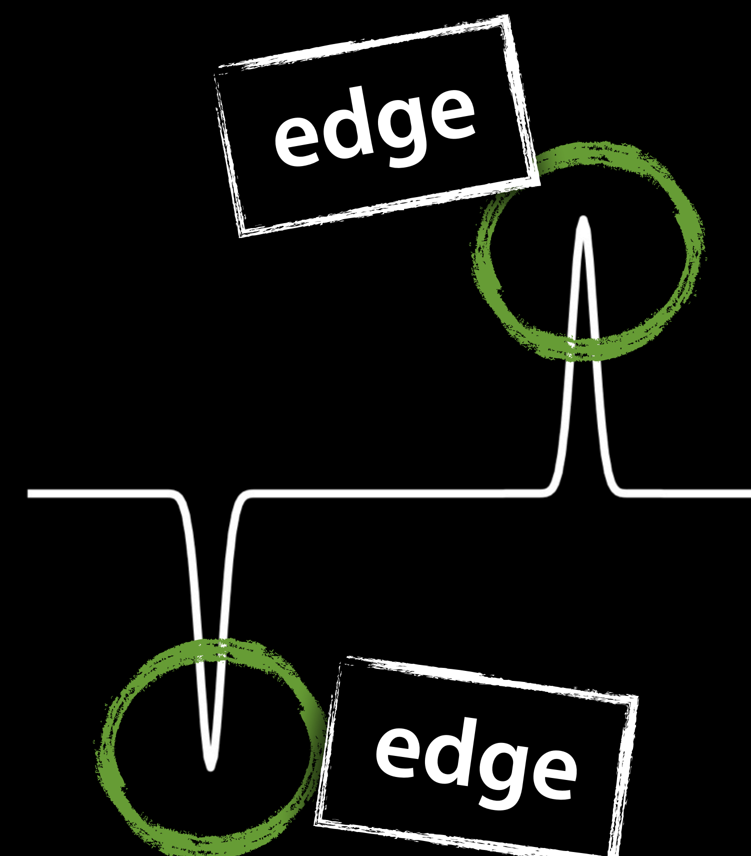
Edge Detection



image



intensity function (slice)

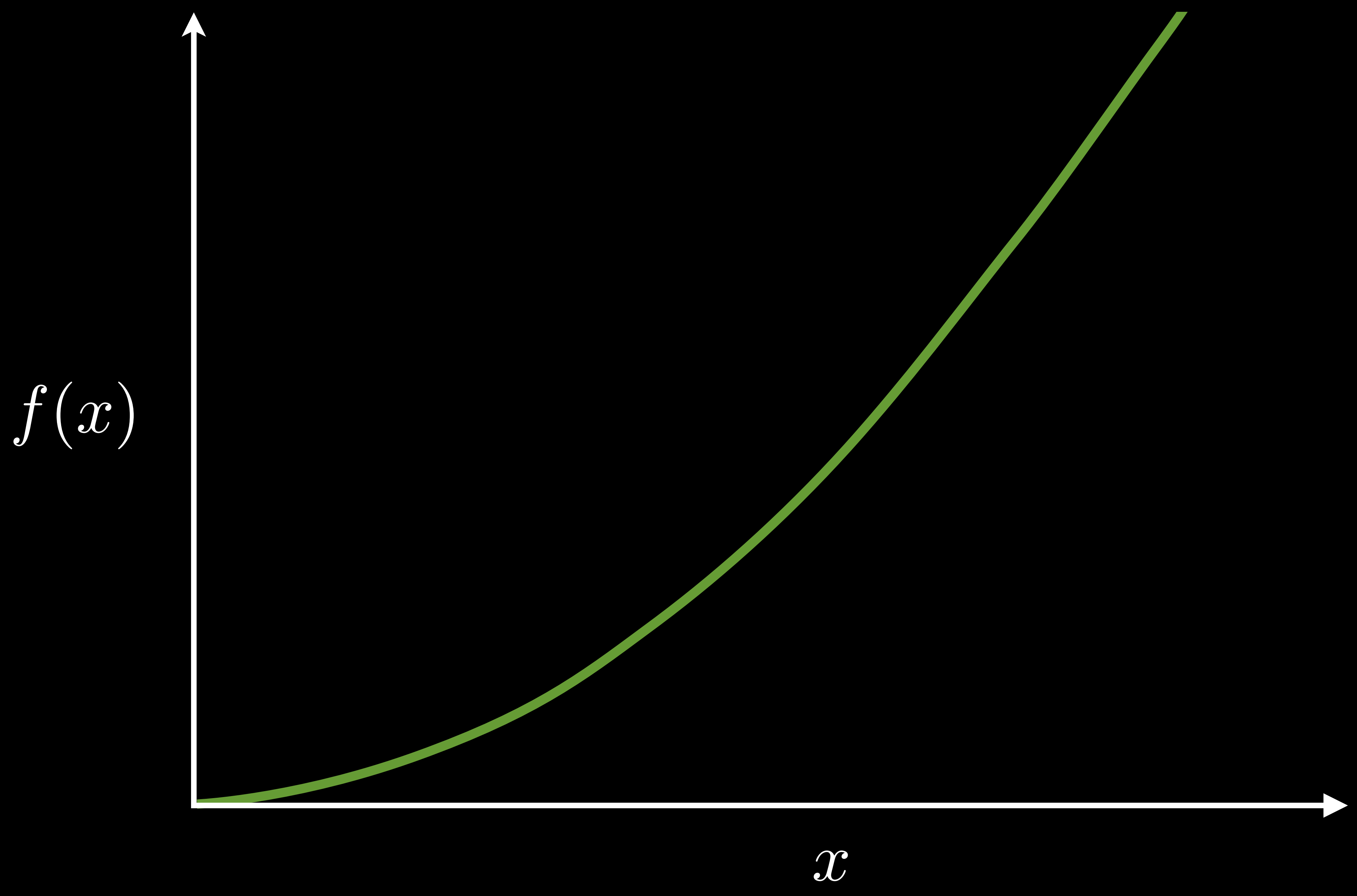


first derivative

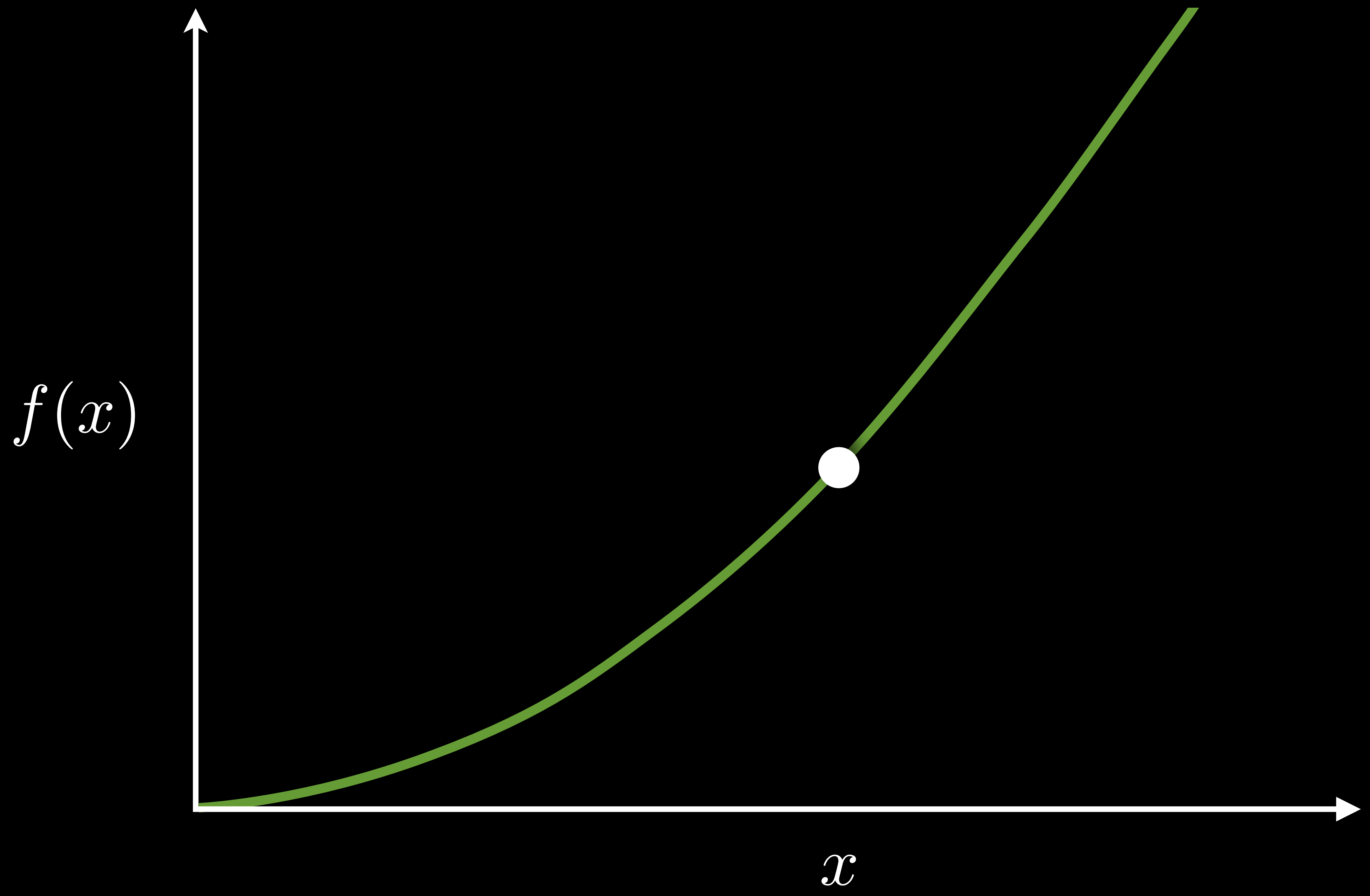
Derivative

$$\frac{df(x)}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

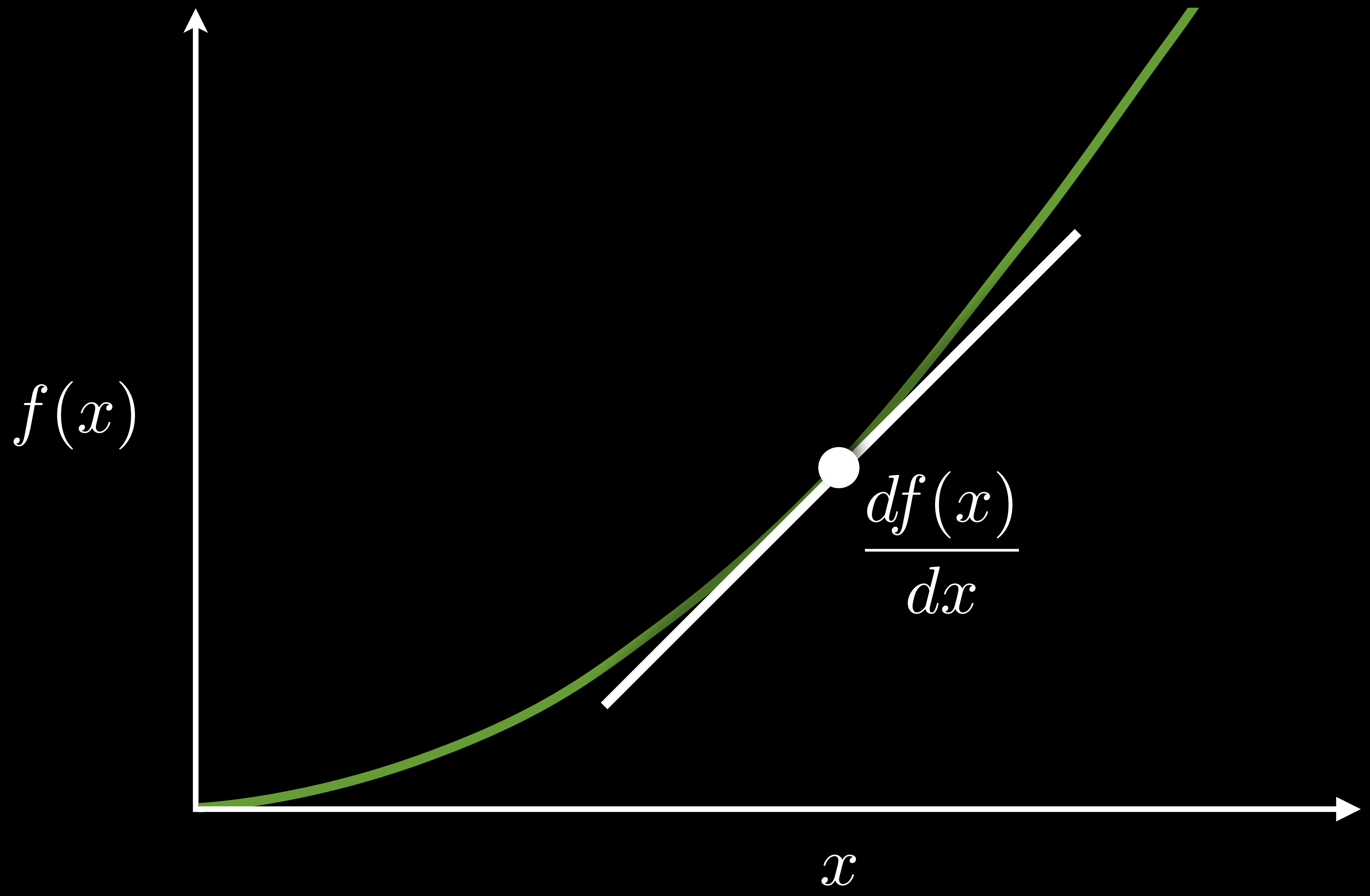
$$\frac{df(x)}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$



$$\frac{df(x)}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$



$$\frac{df(x)}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$



What about multivariate functions?

Partial Derivatives

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon, y) - f(x,y)}{\epsilon}$$

Partial Derivatives

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon, y) - f(x,y)}{\epsilon}$$

$$\frac{\partial f(x,y)}{\partial y} = \lim_{\epsilon \to 0} \frac{f(x, y+\epsilon) - f(x,y)}{\epsilon}$$

Discrete Derivatives

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y) - f(x,y)}{1}$$

Discrete Derivatives

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y)-f(x,y)}{1}$$
$$= f(x+1,y)-f(x,y)$$

Discrete Derivatives

$$\frac{\partial f(x,y)}{\partial x} \approx f(x+1,y) - f(x,y)$$

Discrete Derivatives

$$\frac{\partial f(x,y)}{\partial x} \approx f(x+1,y) - f(x,y)$$

$$\frac{\partial f(x,y)}{\partial y} \approx f(x,y+1) - f(x,y)$$

Discrete Derivatives

$$\frac{\partial f(x,y)}{\partial x} \approx f(x+1,y) - f(x,y)$$

$$\frac{\partial f(x,y)}{\partial y} \approx f(x,y+1) - f(x,y)$$

What are the associated convolution filters?

$$\frac{\partial f(x,y)}{\partial x} \approx f(x+1,y) - f(x,y)$$

$$\frac{\partial f(x,y)}{\partial y} \approx f(x,y+1) - f(x,y)$$

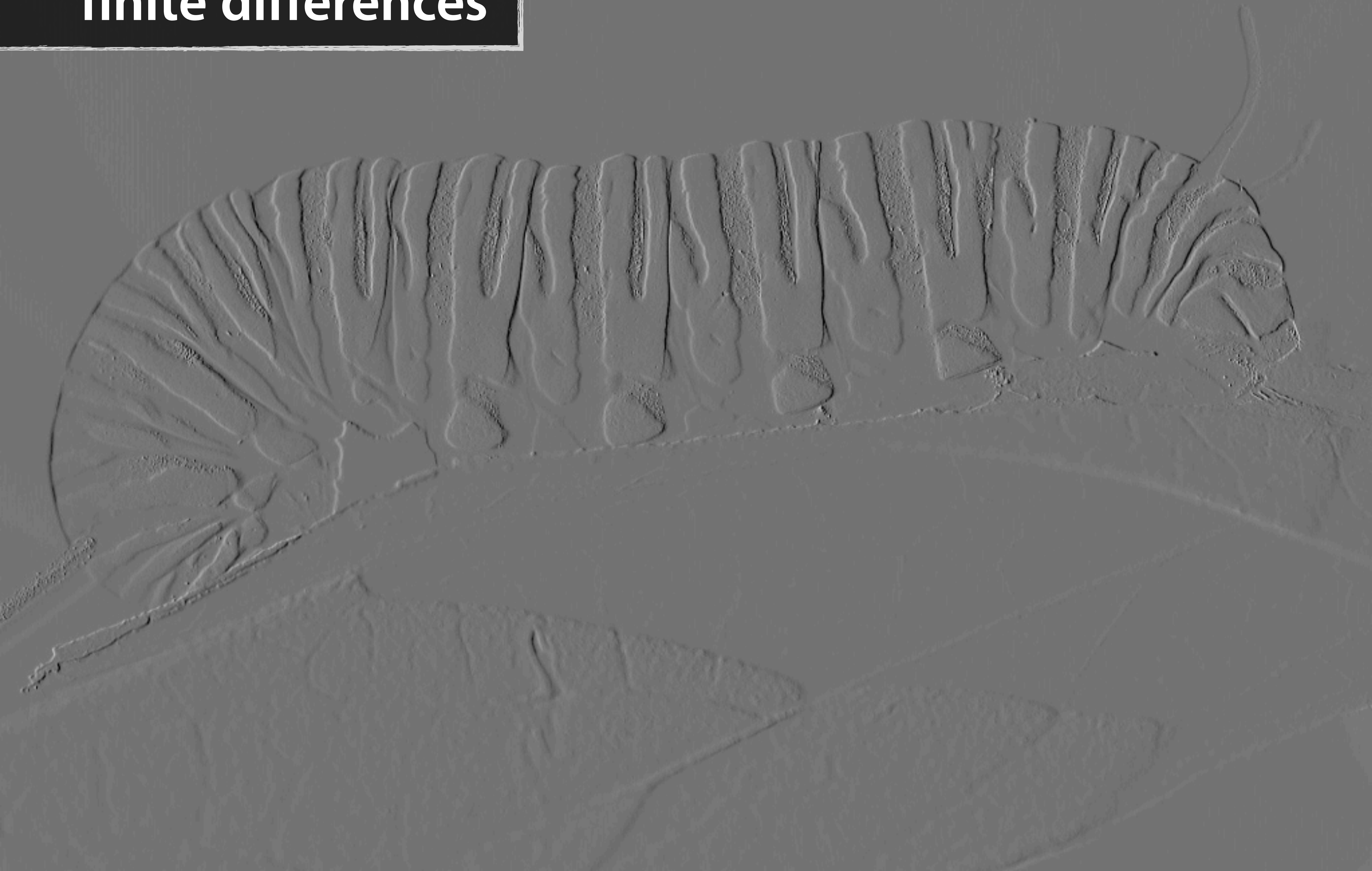
What are the associated convolution filters?

$$\begin{array}{|c|c|}\hline 1 & -1 \\ \hline \end{array} \qquad \begin{array}{|c|}\hline 1 \\ \hline -1 \\ \hline \end{array}$$

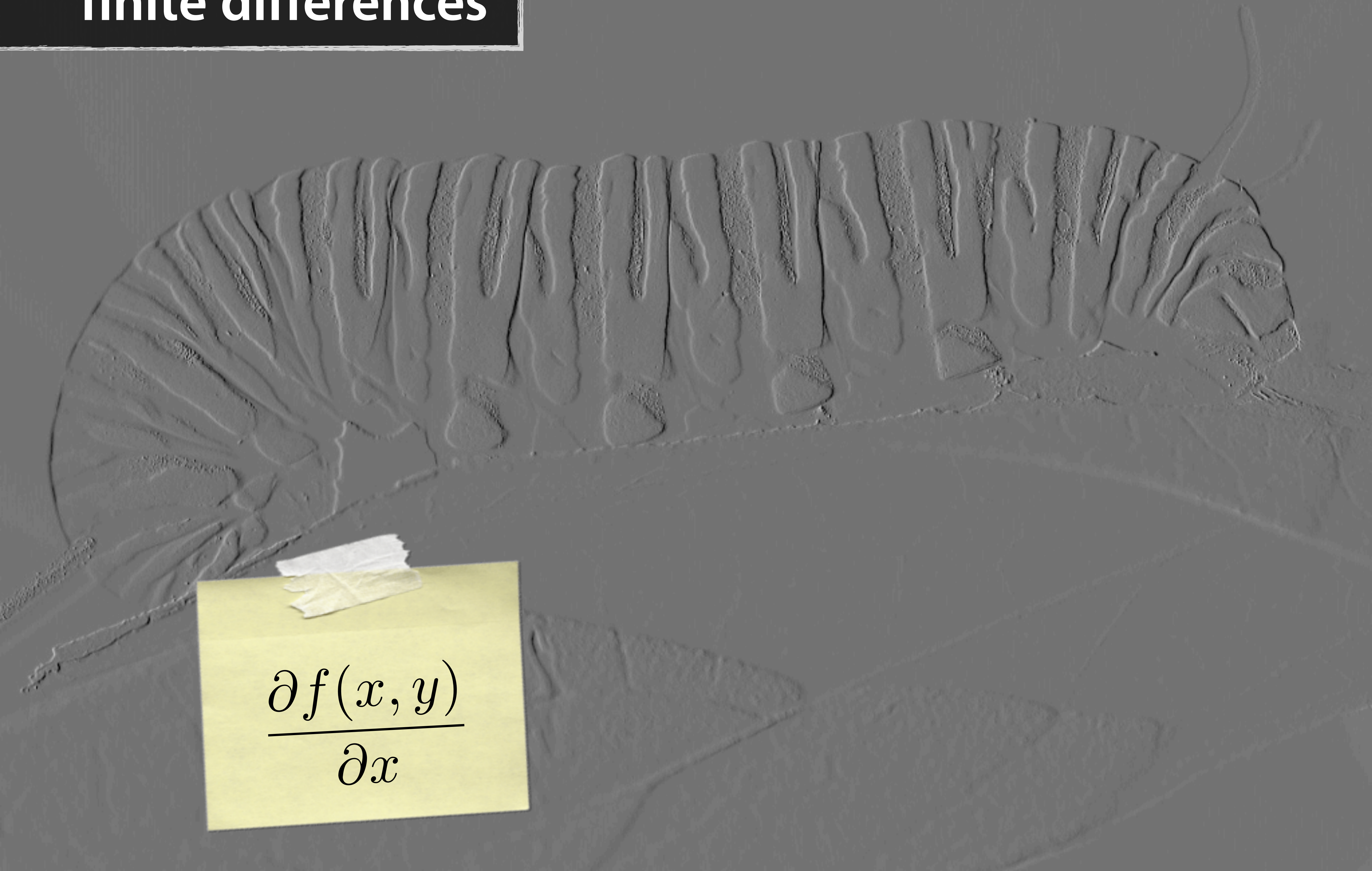
$$\frac{\partial f(x,y)}{\partial x} \qquad \frac{\partial f(x,y)}{\partial y}$$


input image

finite differences

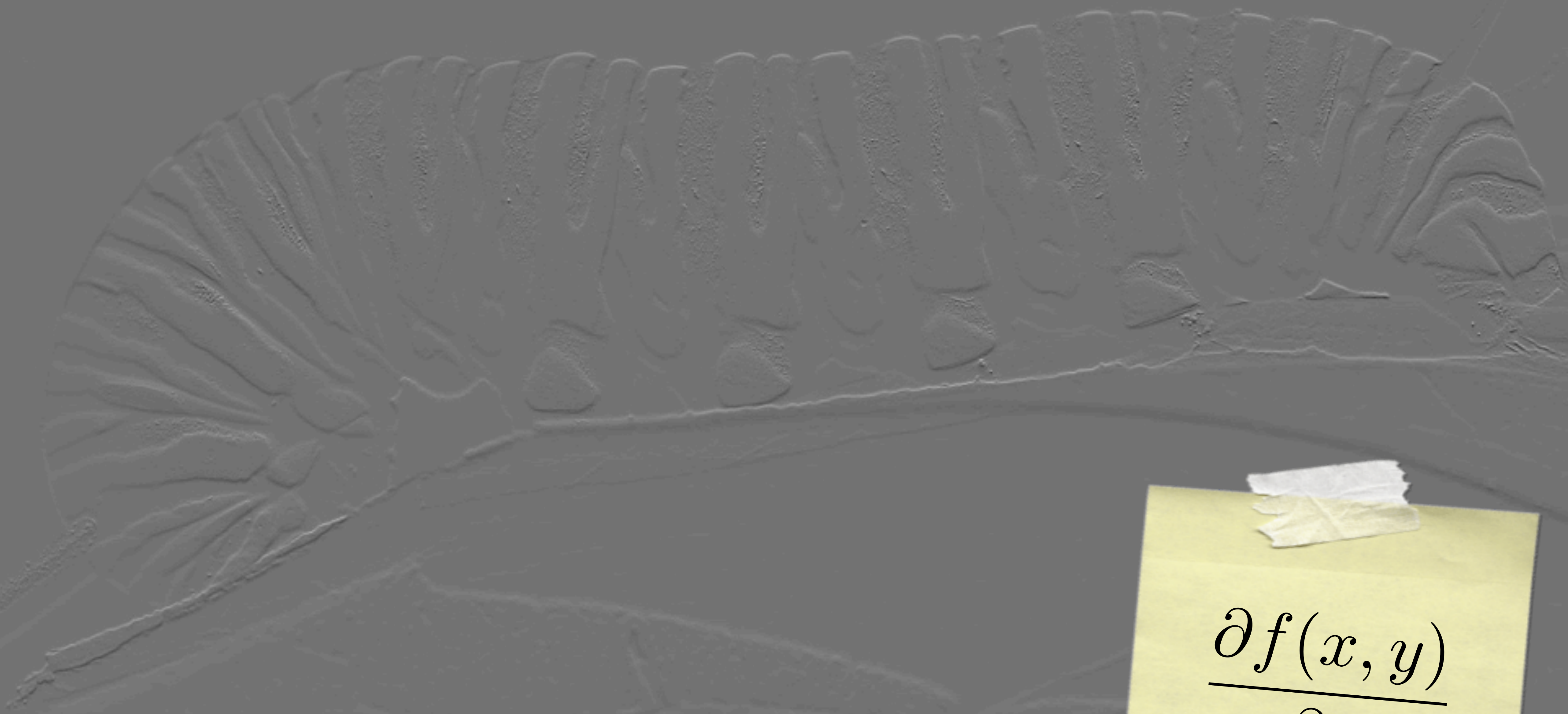


finite differences



$$\frac{\partial f(x,y)}{\partial x}$$

finite differences



$$\frac{\partial f(x,y)}{\partial y}$$

1	2	1
0	0	0
−1	−2	−1

Sobel Filter

separable filters

$$F[x, y] = U[x]V[y]$$

1	2	1
0	0	0
−1	−2	−1

Sobel Filter

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

Sobel Filter

1	0	−1
2	0	−2
1	0	−1

1	2	1
0	0	0
−1	−2	−1

Sobel Filter

Other Derivative Filters

1	0	−1
2	0	−2
1	0	−1

1	2	1
0	0	0
−1	−2	−1

Sobel Filter

1	0	−1
1	0	−1
1	0	−1

1	1	1
0	0	0
−1	−1	−1

Prewitt

Other Derivative Filters

1	0	−1
2	0	−2
1	0	−1

1	2	1
0	0	0
−1	−2	−1

Sobel Filter

1	0	−1
1	0	−1
1	0	−1

1	1	1
0	0	0
−1	−1	−1

Prewitt

0	−1
1	0

−1	0
0	1

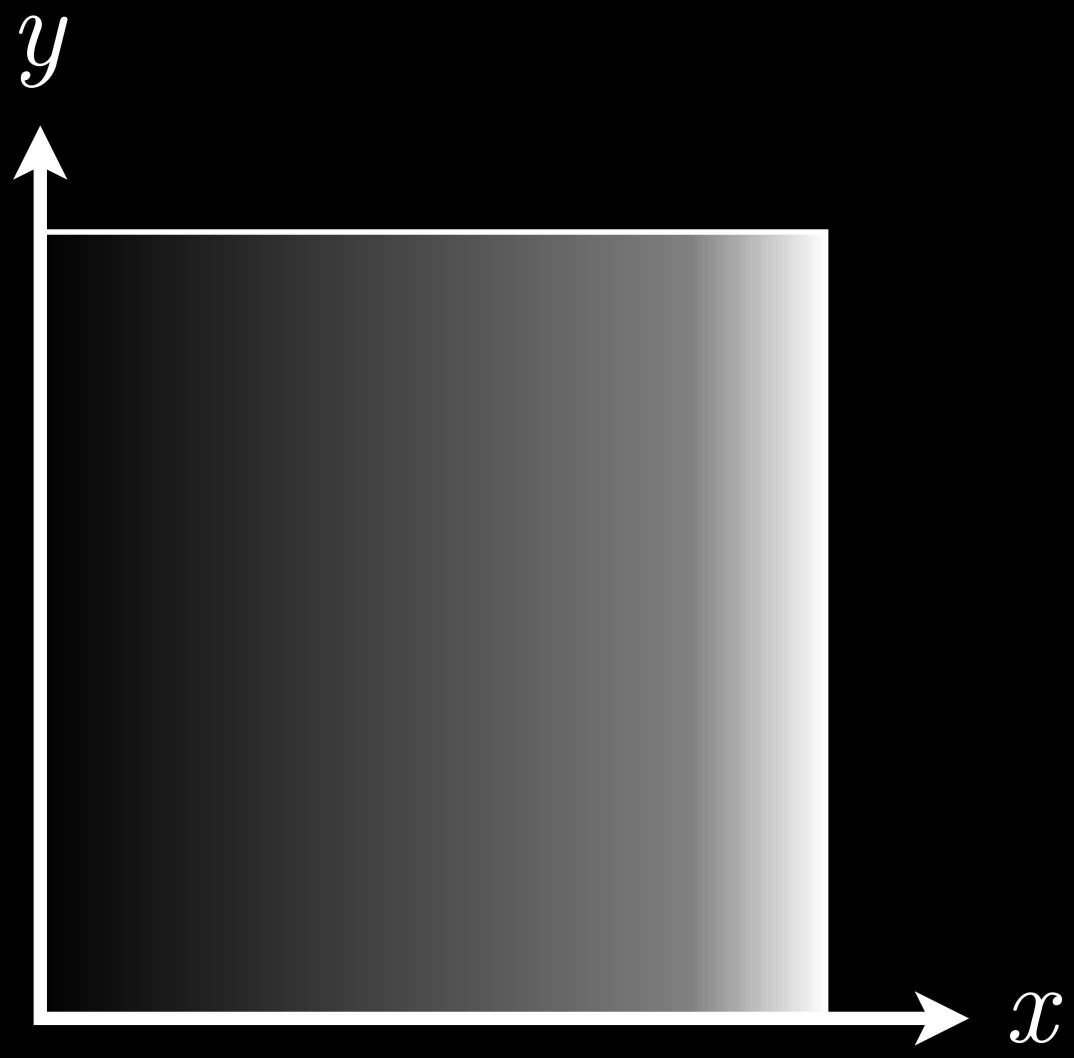
Roberts Cross

gradient

a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase

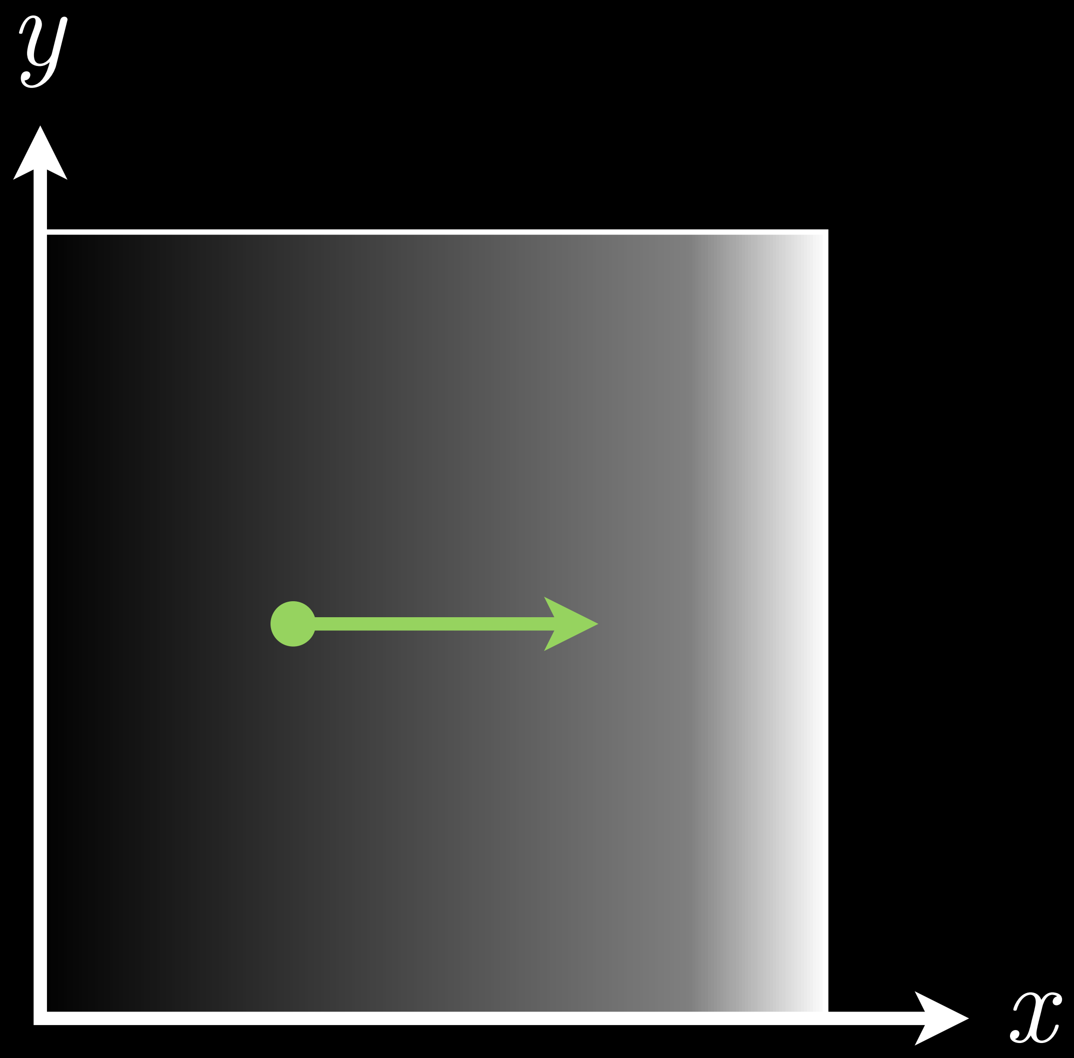
gradient

a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase



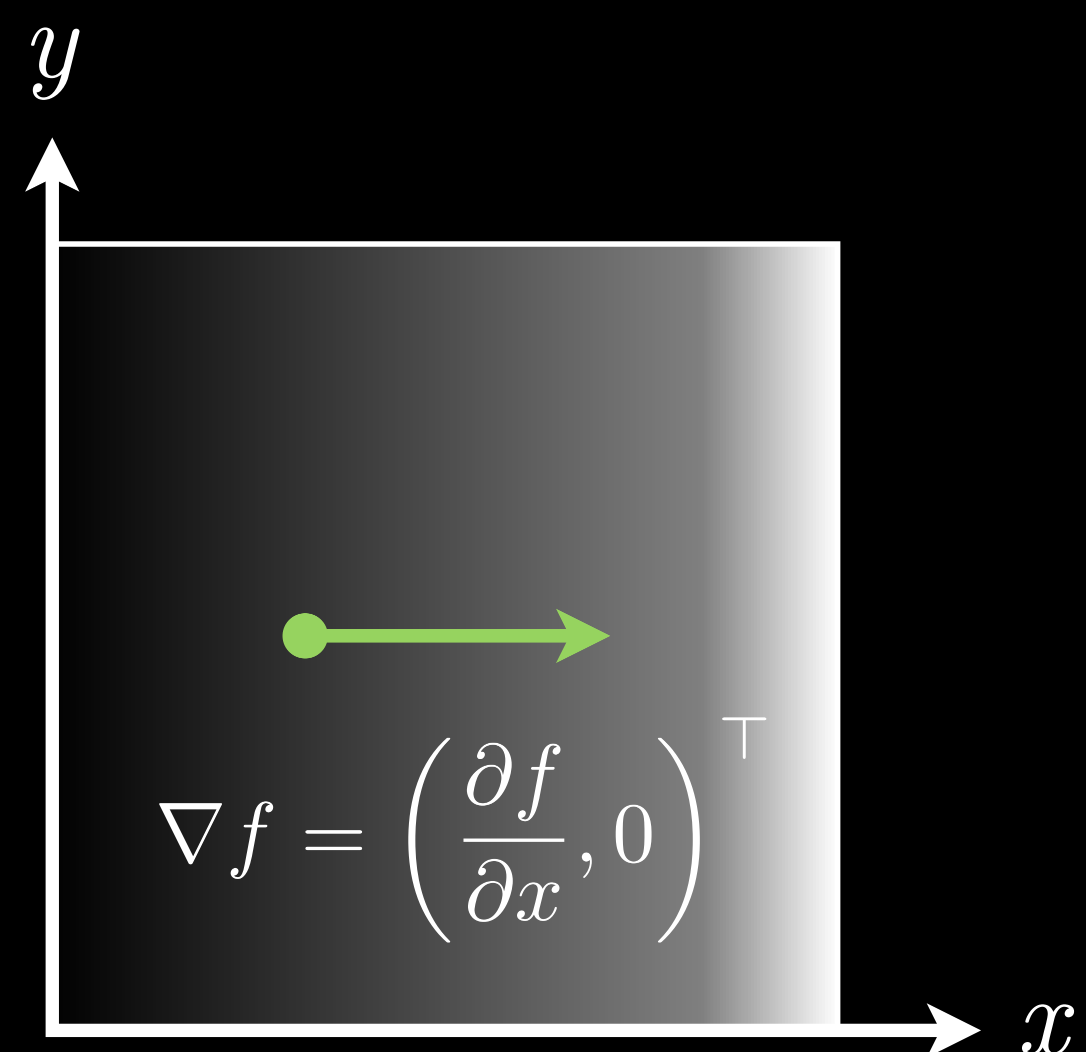
gradient

a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase



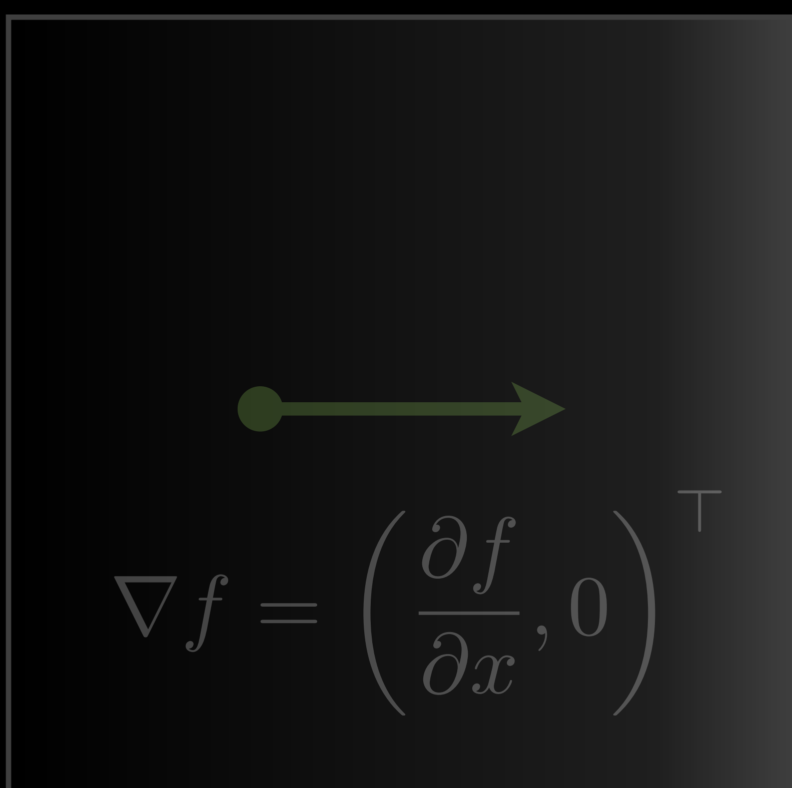
gradient

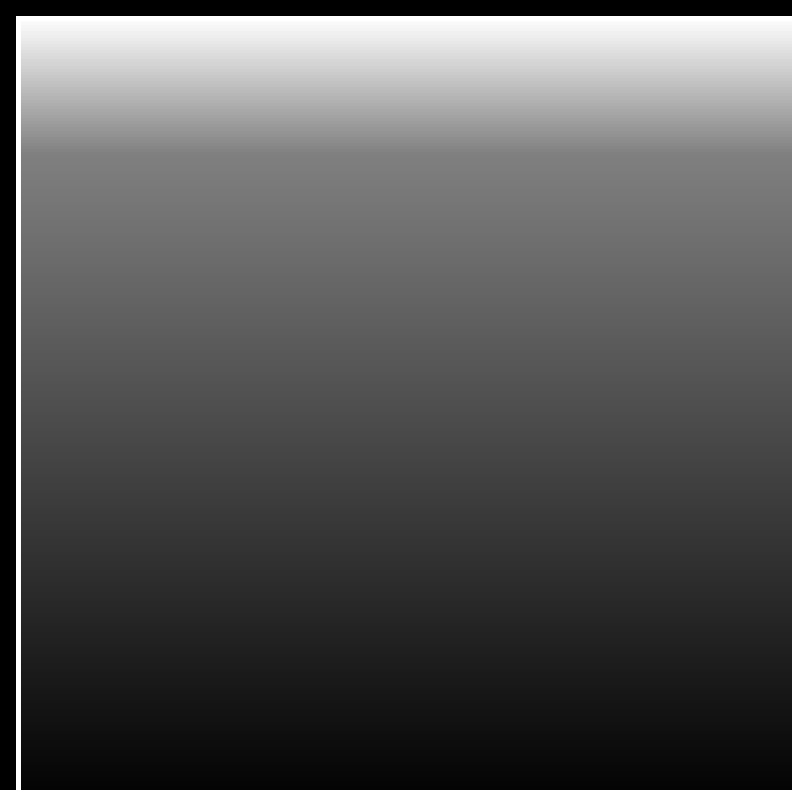
a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase



gradient

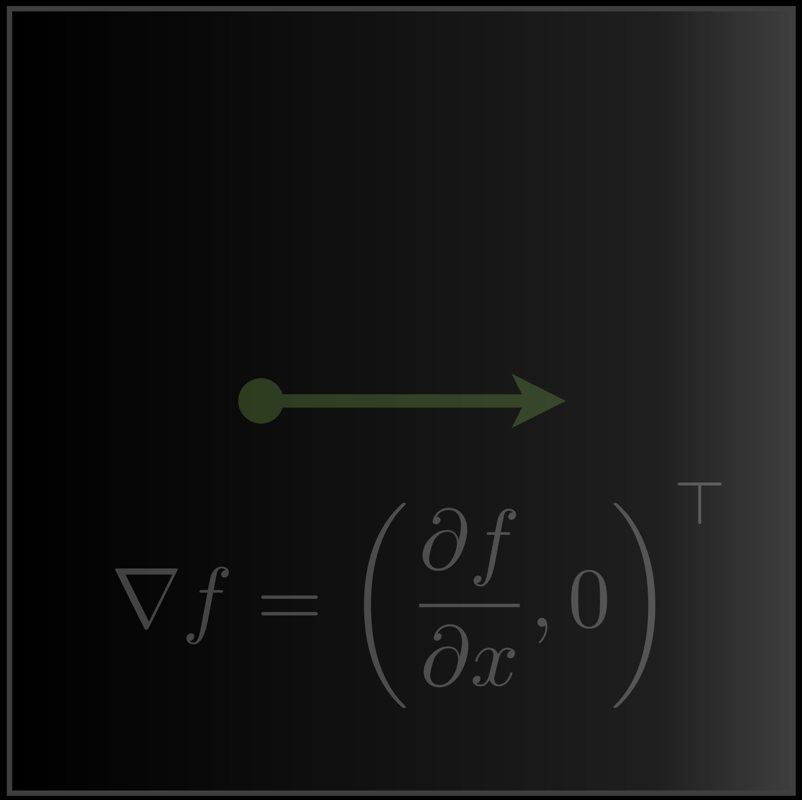
a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase

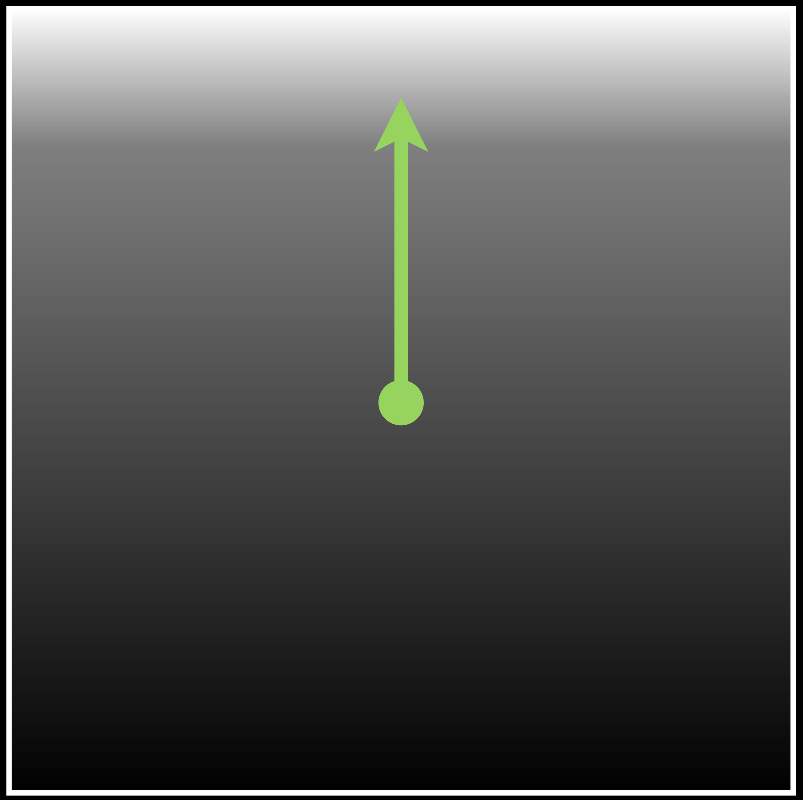




gradient

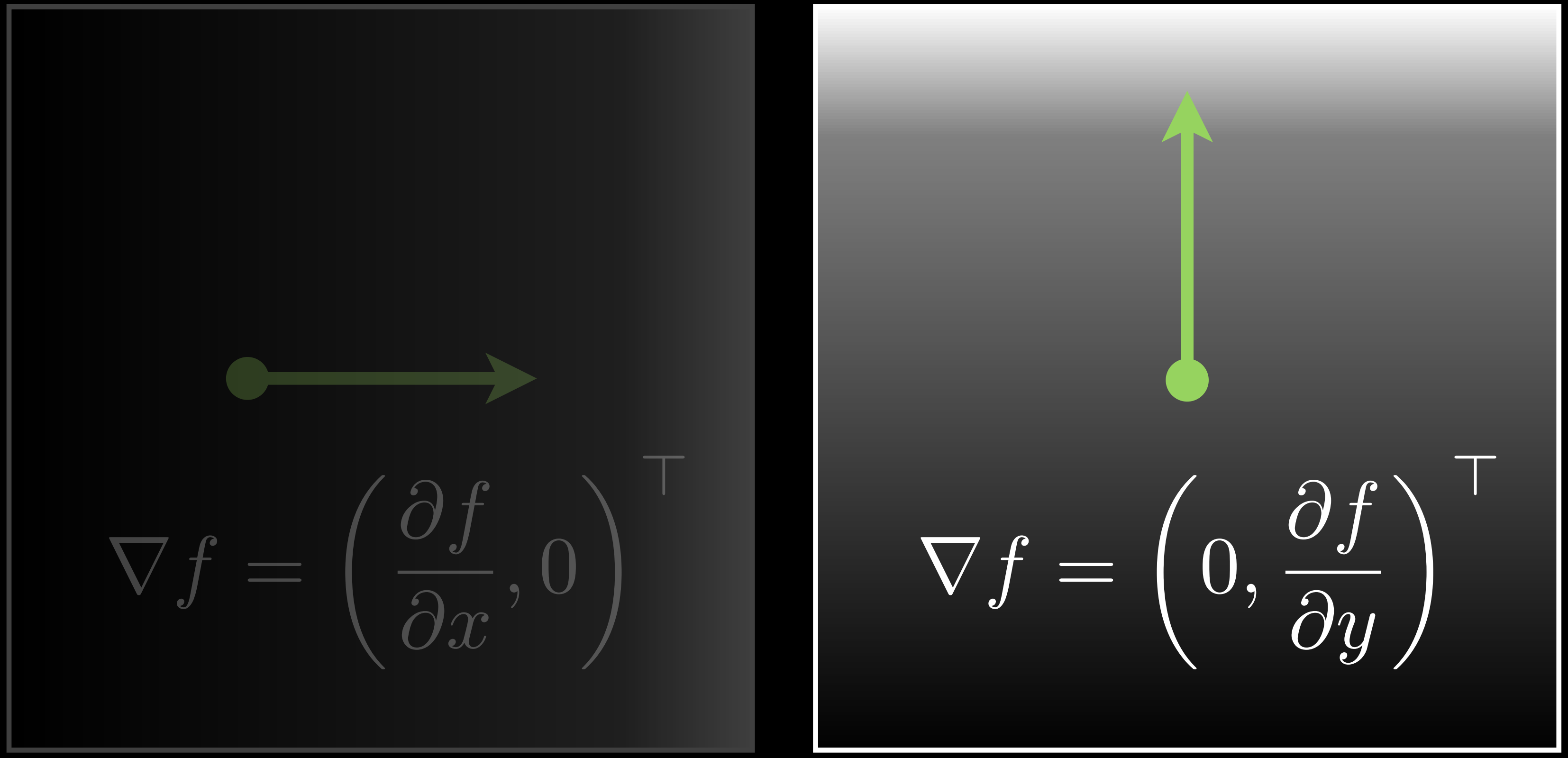
a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase





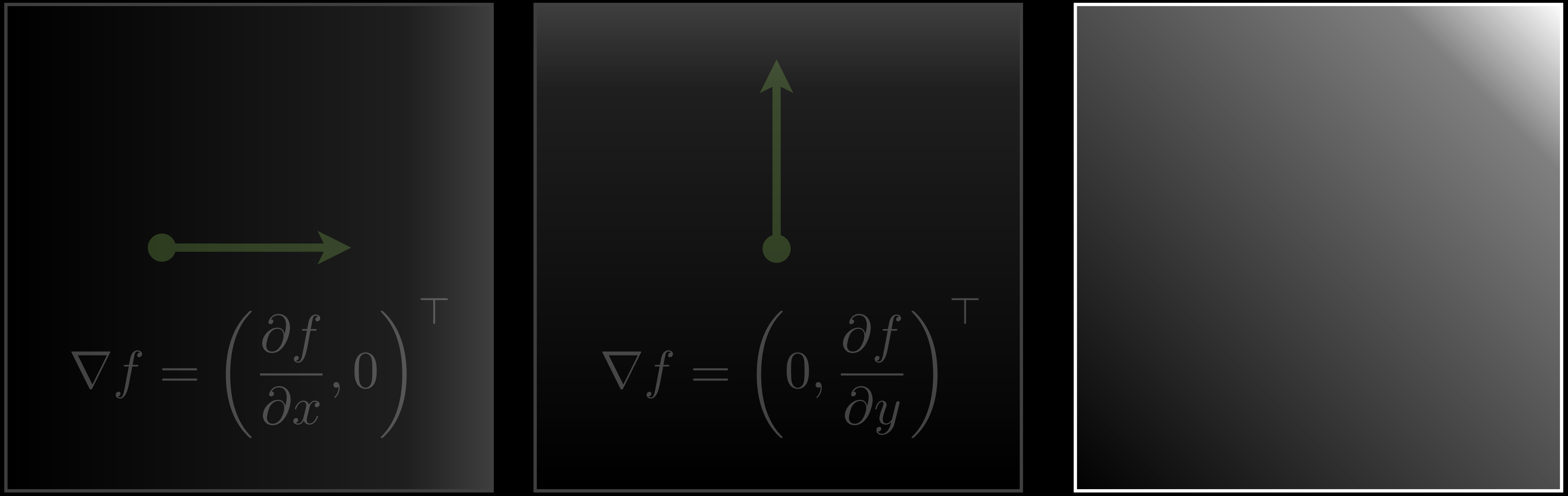
gradient

a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase



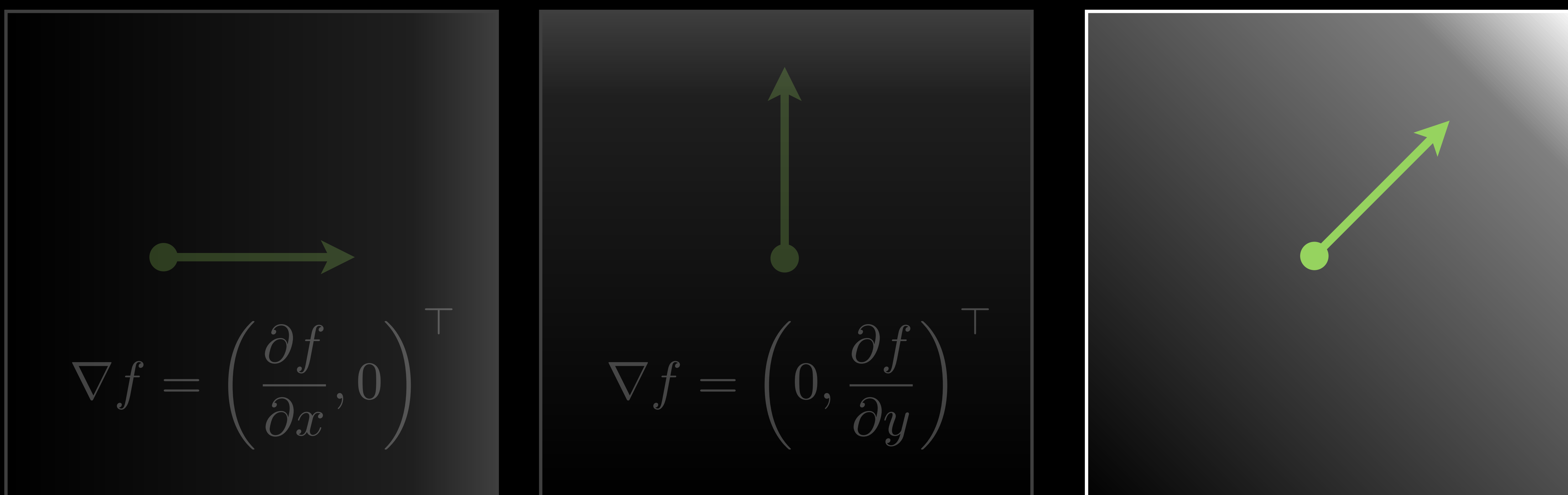
gradient

a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase



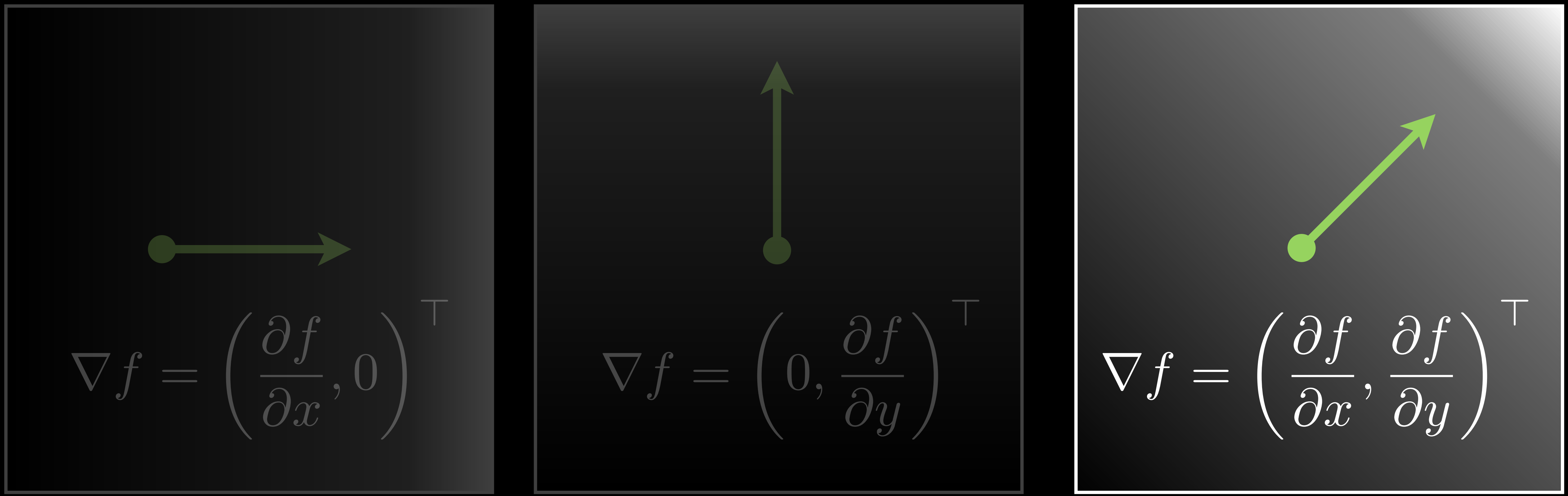
gradient

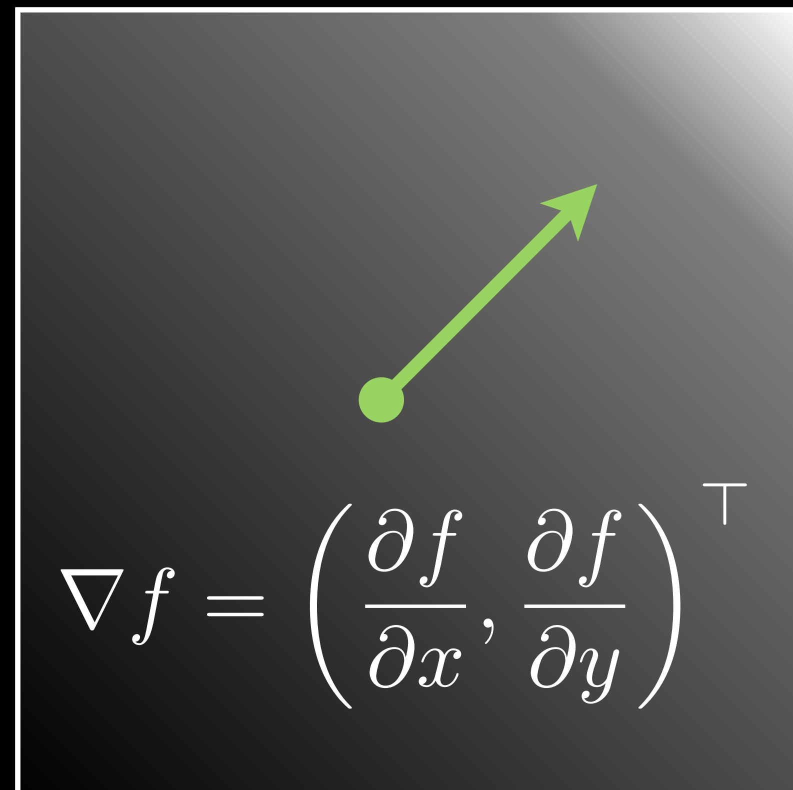
a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase

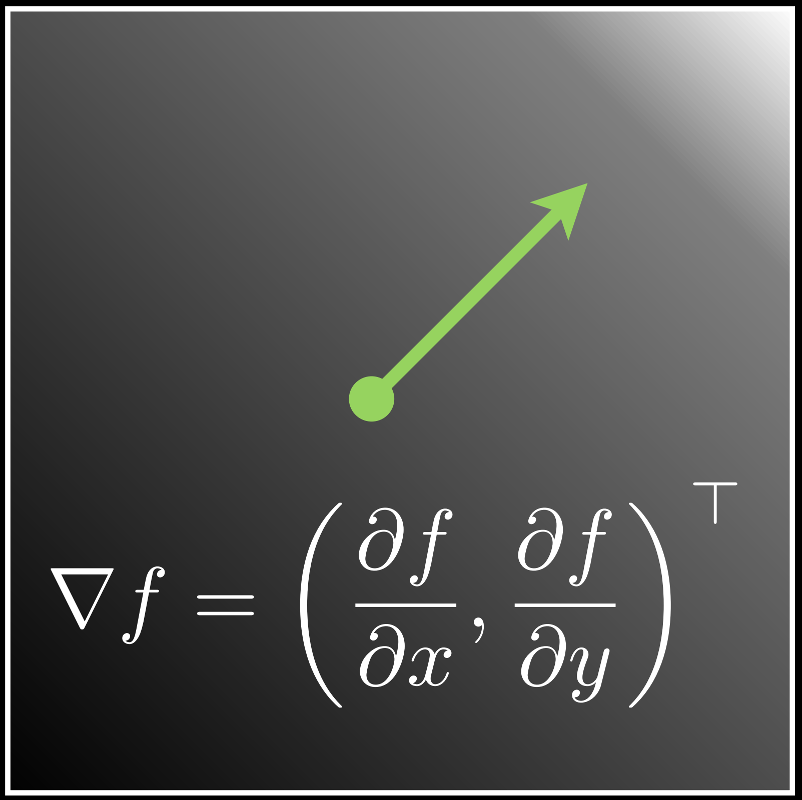


gradient

a vector that points in the direction of the greatest rate of increase of a function and whose magnitude is that rate of increase

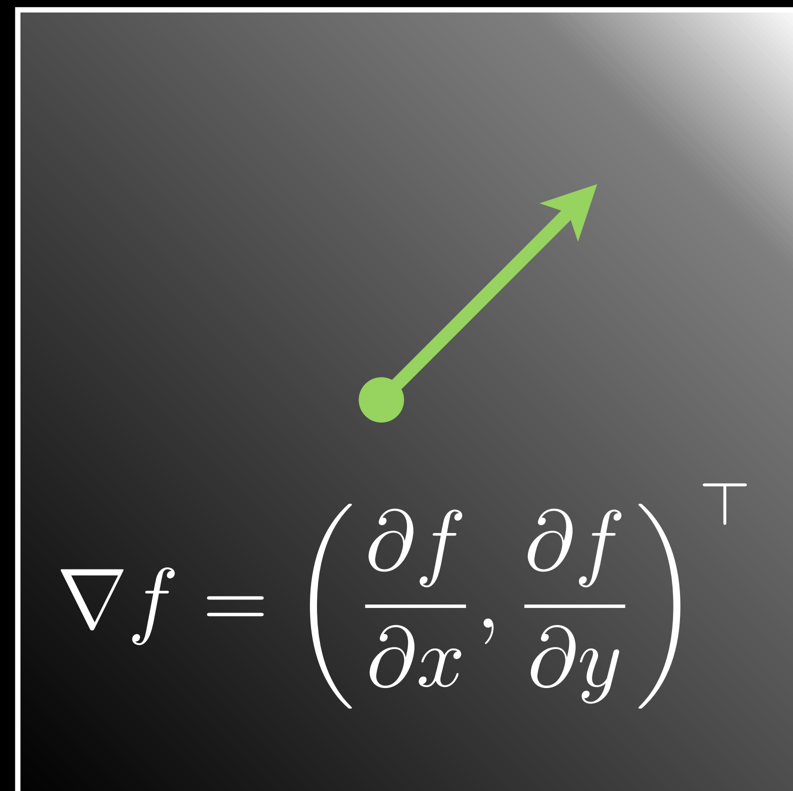



$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^{\top}$$

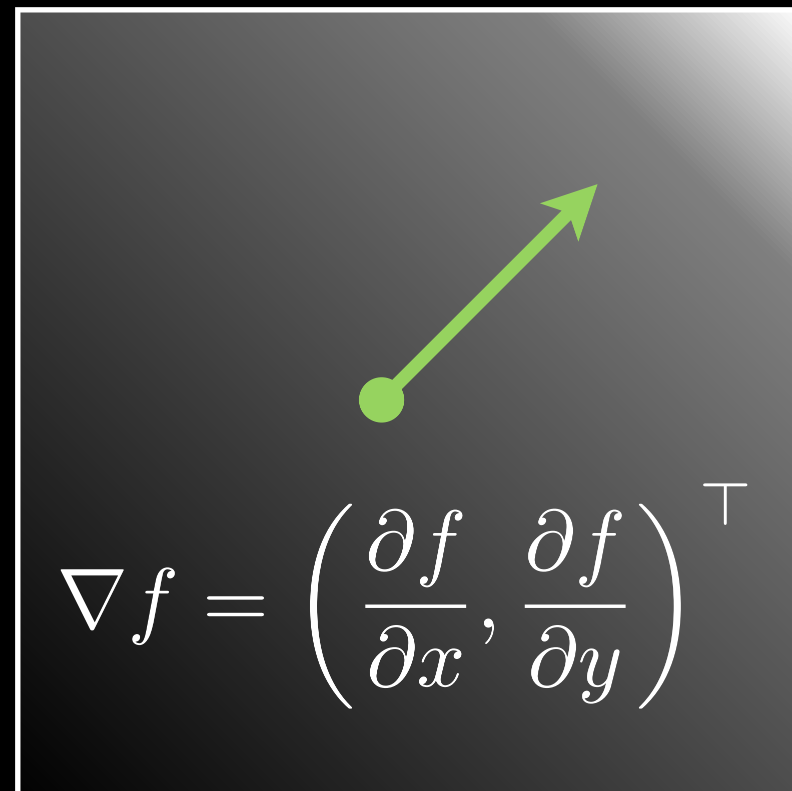


$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^{\top}$$

Gradient magnitude

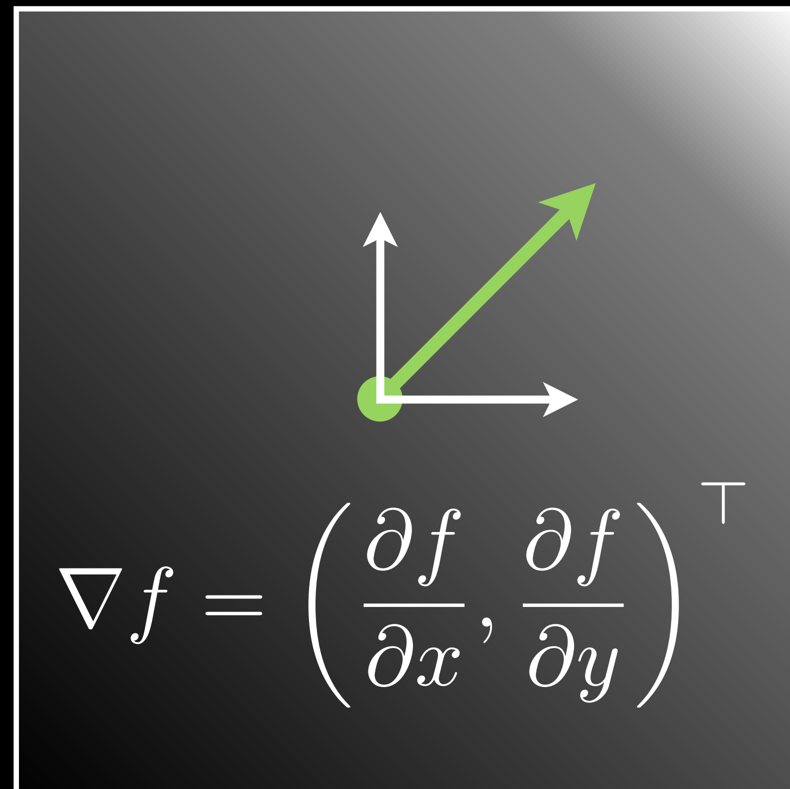


Gradient magnitude $\|\nabla f\| = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$



Gradient magnitude $$\|\nabla f\| = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$$

Gradient direction



Gradient magnitude

$$\|\nabla f\| = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$$

Gradient direction

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)^{\top}$$

Gradient magnitude $$\|\nabla f\| = \sqrt{\frac{\partial f}{\partial x}^{2} + \frac{\partial f}{\partial y}^{2}}$$

Gradient direction

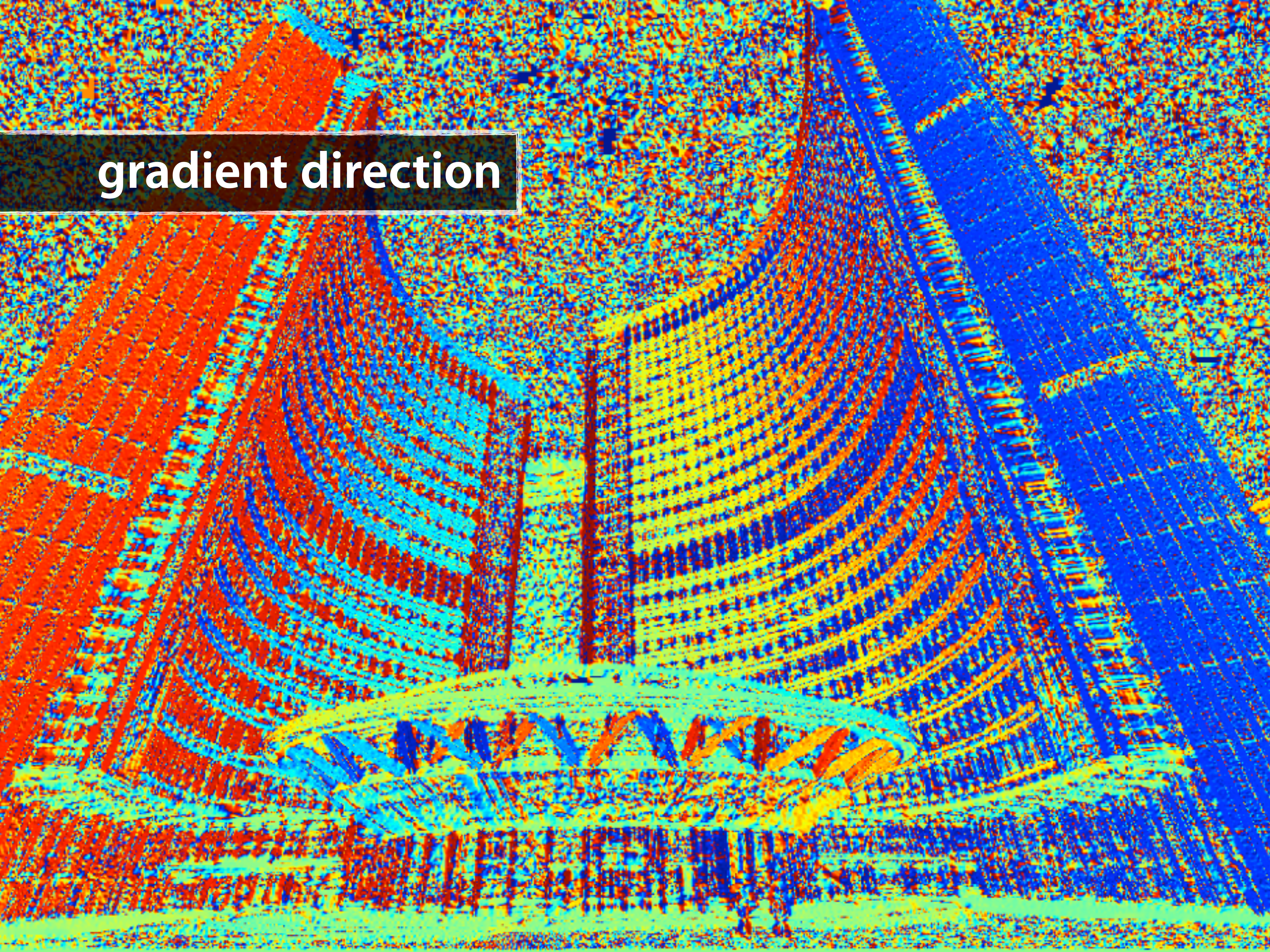
$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^{\top}$$

Gradient magnitude $$\|\nabla f\| = \sqrt{\frac{\partial f}{\partial x}^{2} + \frac{\partial f}{\partial y}^{2}}$$

Gradient direction $$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \bigg/ \frac{\partial f}{\partial x} \right)$$


original


gradient magnitude


gradient direction