THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision

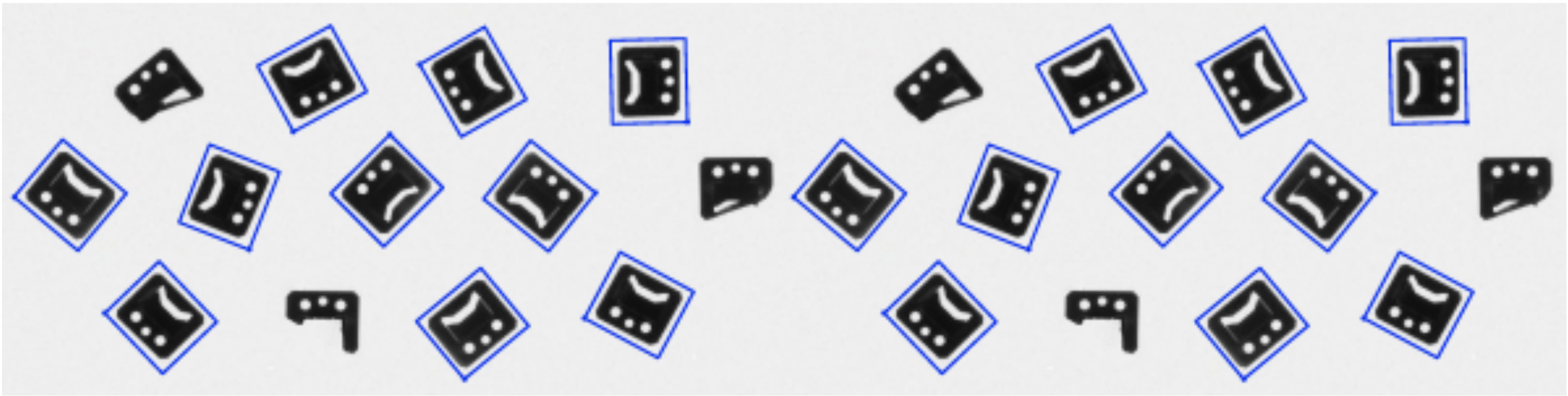


Image Credit: https://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html

Lecture 7: Digital Imaging Pipeline, Template Matching

(unless otherwise stated slides are taken or adopted from **Bob Woodham, Jim Little** and **Fred Tung**)

Menu for Today

Topics:

— **Digital Imaging** Pipeline
— **Scaled** Representations
— Template **Matching**
— Normalised **Correlation**

Readings:

— **Today's** Lecture: Szeliski 2.3, 3.5, Forsyth & Ponce (2nd ed.) 4.5 - 4.7

Reminders:

— **Assignment 1**: due **Thursday** September 28th

— **Assignment 2**: Scaled Representations, Face Detection and Image Blending available now

Today’s “**fun**” Example: Optical Illusions for Traffic Safety

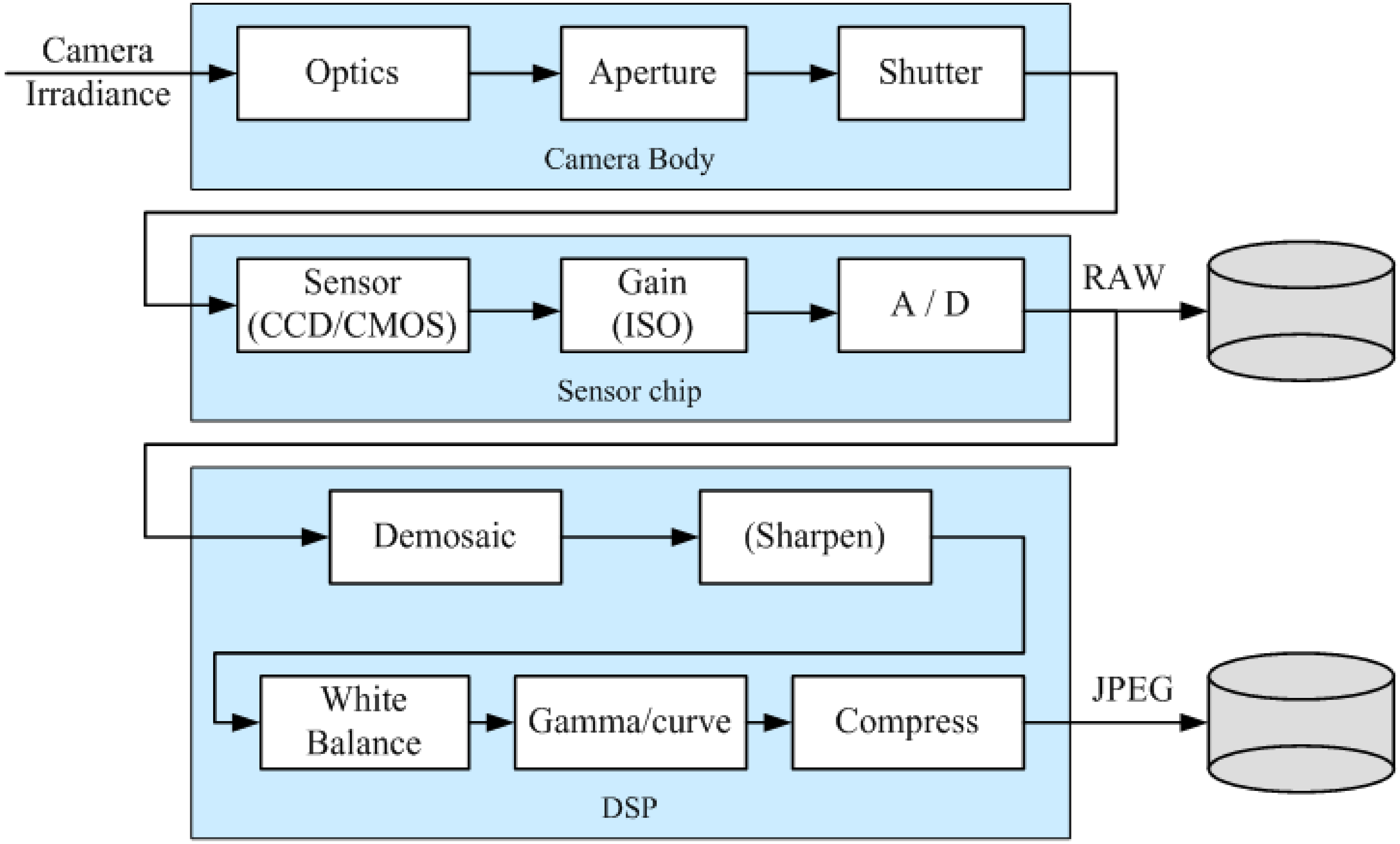
Image From: https://inudgeyou.com/en/nudging-traffic-safety-by-visual-illusions/

Today’s “**fun**” Example: Optical Illusions for Traffic Safety

Image From: https://inudgeyou.com/en/nudging-traffic-safety-by-visual-illusions/

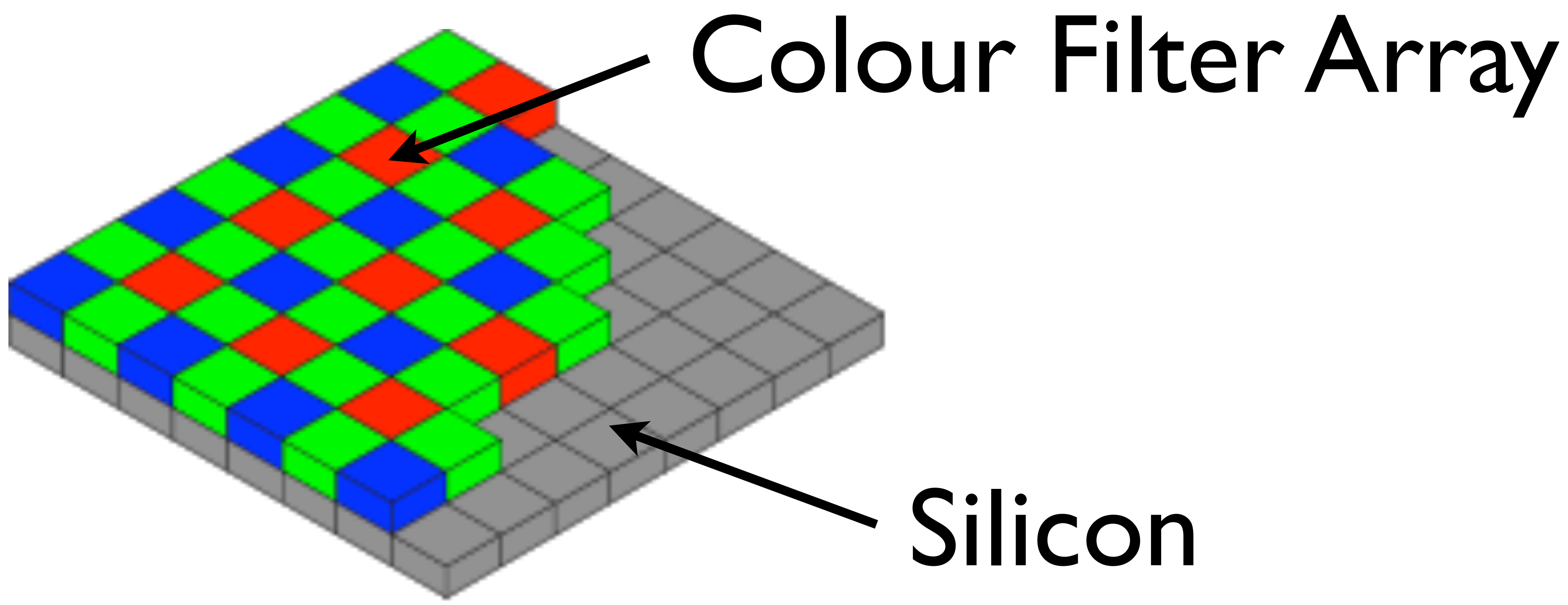
Digital Camera Processing

- Main stages in a digital camera

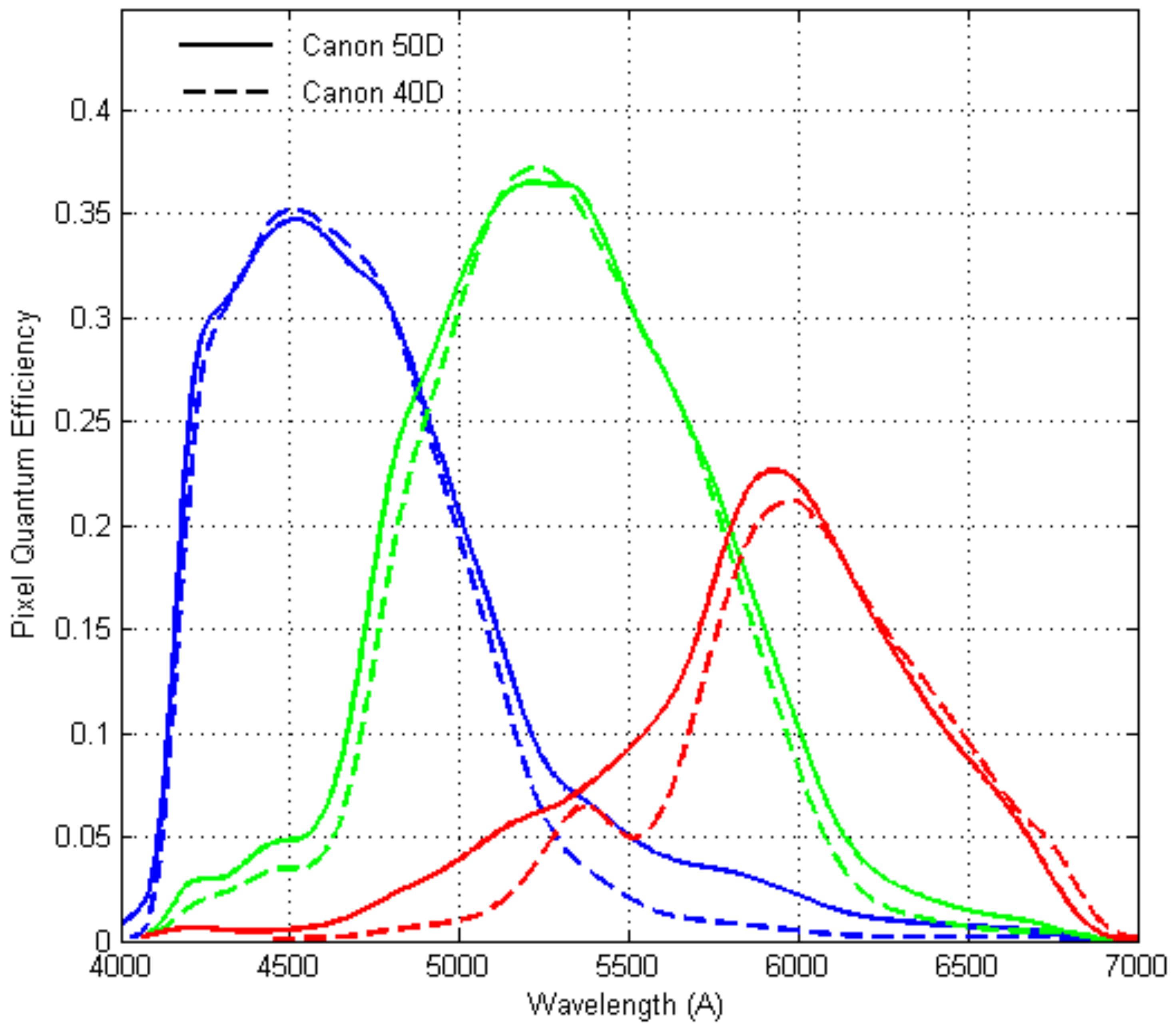


[Szeliski 2.3]

Digital Sensor



Canon 50D

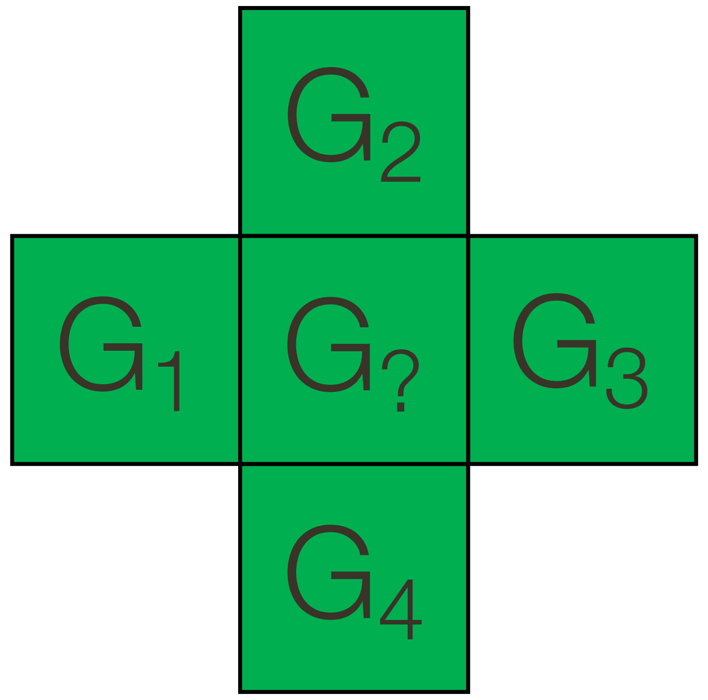


$f(\lambda)$

- Analogue image is sampled by a CMOS (or CCD) sensor
- RGB colour filters arranged in a “Bayer” pattern
- Spectral response of R,G,B filters = Quantum Efficiency
- Counts from this sensor are camera RAW

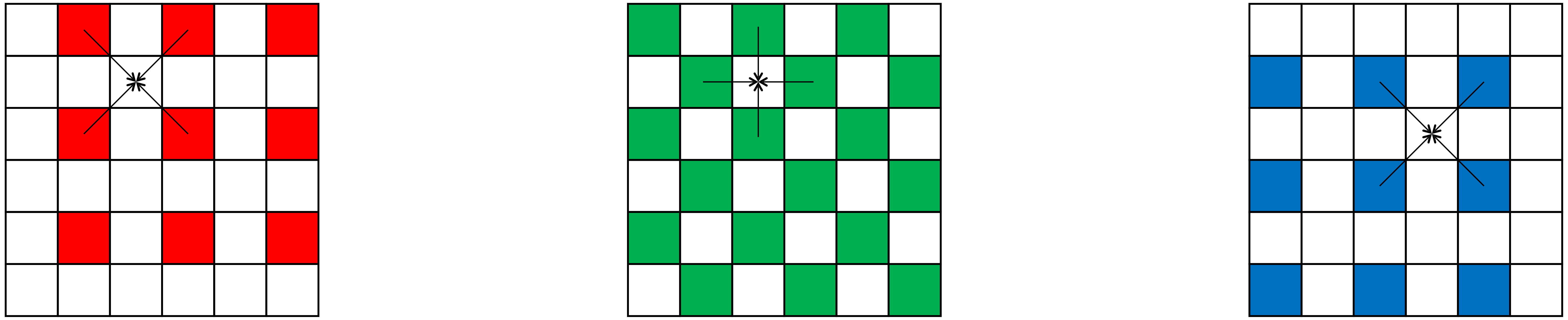
Demosaicing by Bilinear Interpolation

Bilinear interpolation: Simply average your 4 neighbors.



$$G_? = \frac{G_1 + G_2 + G_3 + G_4}{4}$$

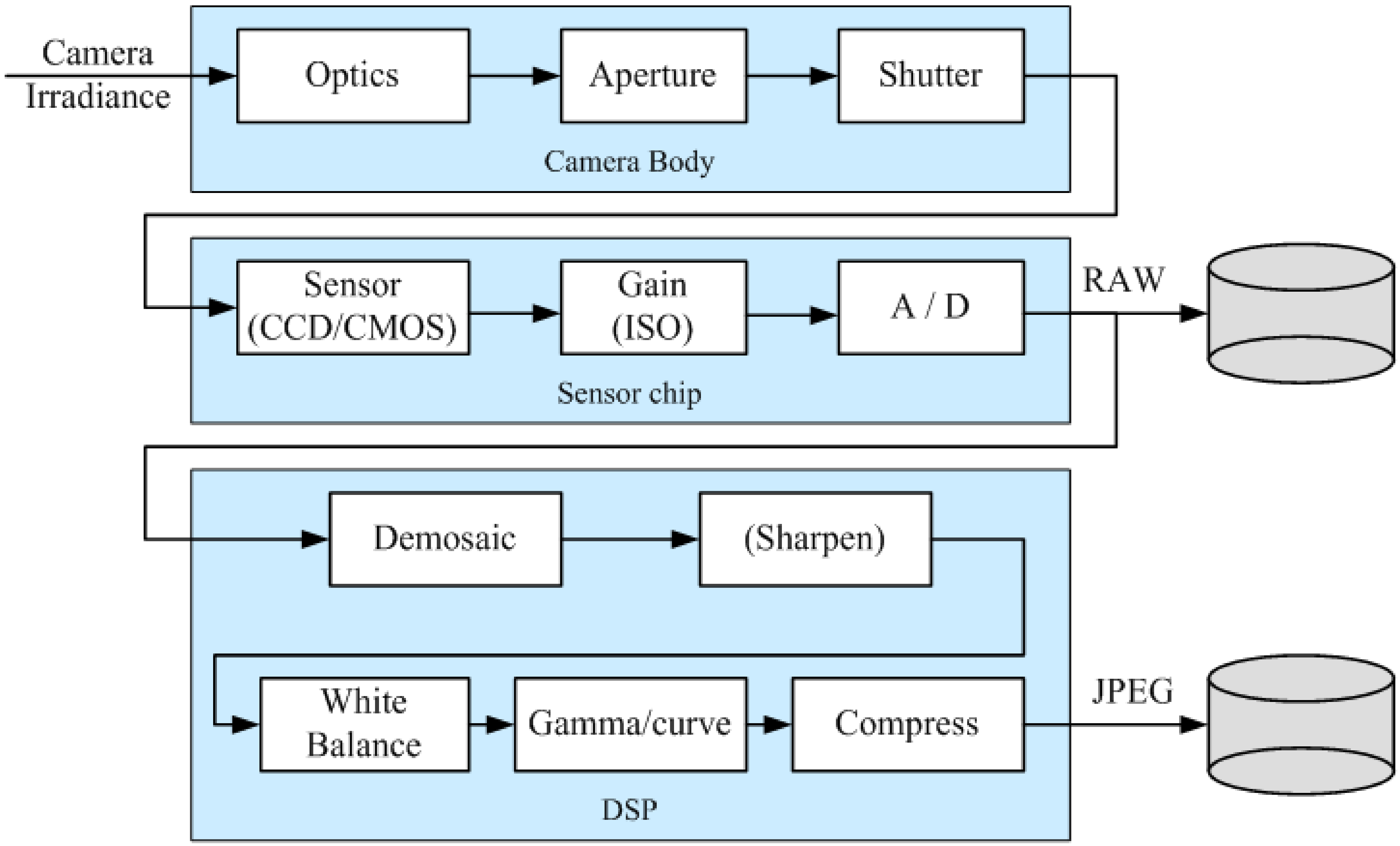
Neighborhood changes for different channels:



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Digital Camera Processing

- Main stages in a digital camera



[Szeliski 2.3]

White Balance

- Humans are good at adapting to global illumination conditions: you would still describe a white object as white whether under blue sky or candle light.
- However, when the picture is viewed later, the viewer is no longer correcting for the environment and the illuminant colour typically appears too strong.
- **White balancing** is the process of correcting for the illuminant

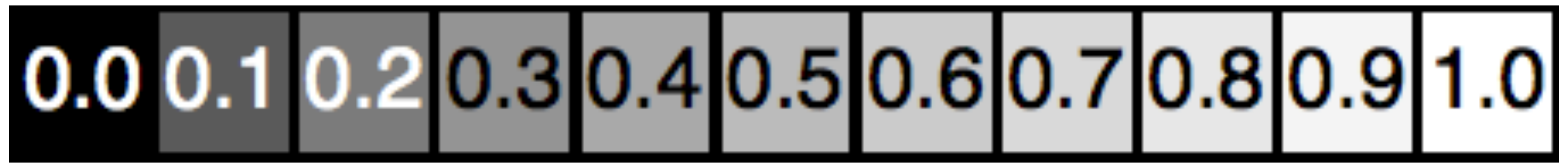
7.1

- A simple white balance algorithm is to assume the scene is grey on average "greyworld", state of the art methods use learning, e.g., Barron ICCV 2015

Gamma Correction

- Equal steps in luminance ≠ equal in perceived brightness

linear luminance (raw)



equal brightness steps

- Equal steps in human perceived brightness are achieved by increasingly large steps in luminance (sensor counts)
- So we encode pixel values V using a power law:

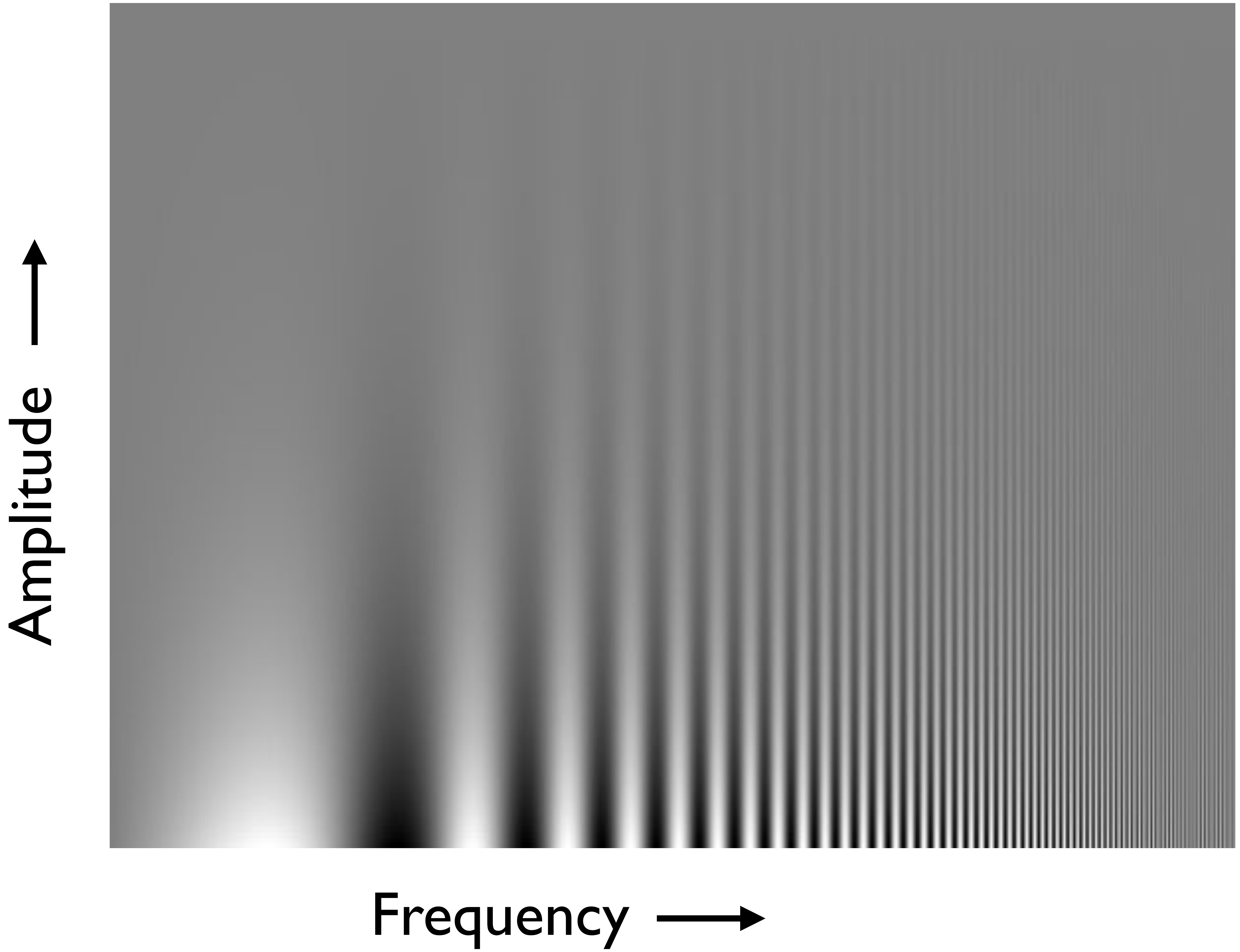
7.2

$$L = V^{\gamma}$$

- Using raw sensor counts wastes bits as we can't differentiate the large values → use **gamma corrected encoding** (V) that allocates more bits to smaller values

Contrast Sensitivity

- Human visual system is most sensitive to mid-frequencies

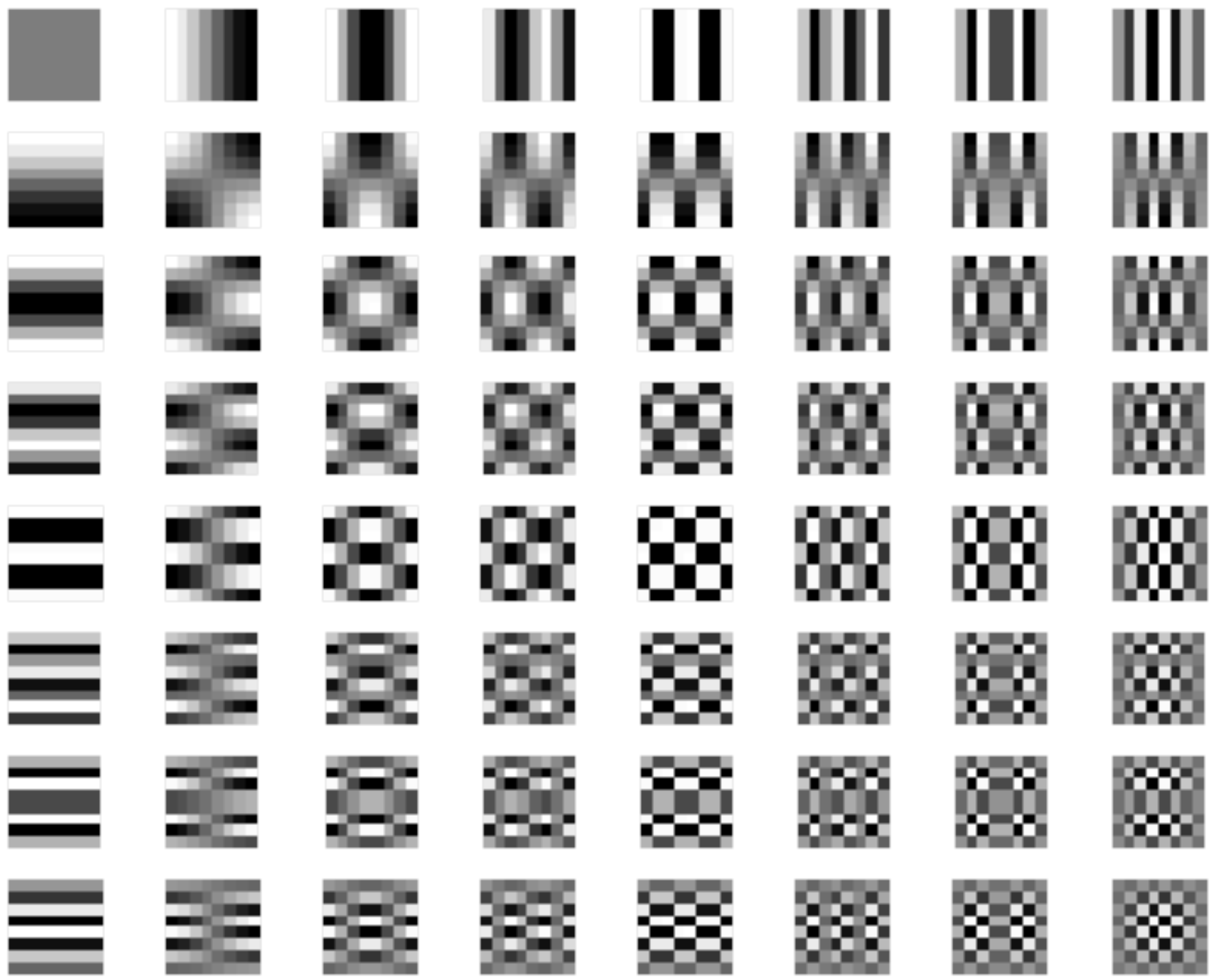


Discrete Cosine Transform

- Basis functions used in JPEG

$$X(m,n) = \alpha_m \alpha_n \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} x(k,l) \cos\left[\frac{(2k+1)m\pi}{2K}\right] \cos\left[\frac{(2l+1)n\pi}{2L}\right]$$

- Energy is concentrated in the low frequency components
- Efficient algorithm to compute (similar to FFT)



8x8 basis functions

Coefficient Quantisation

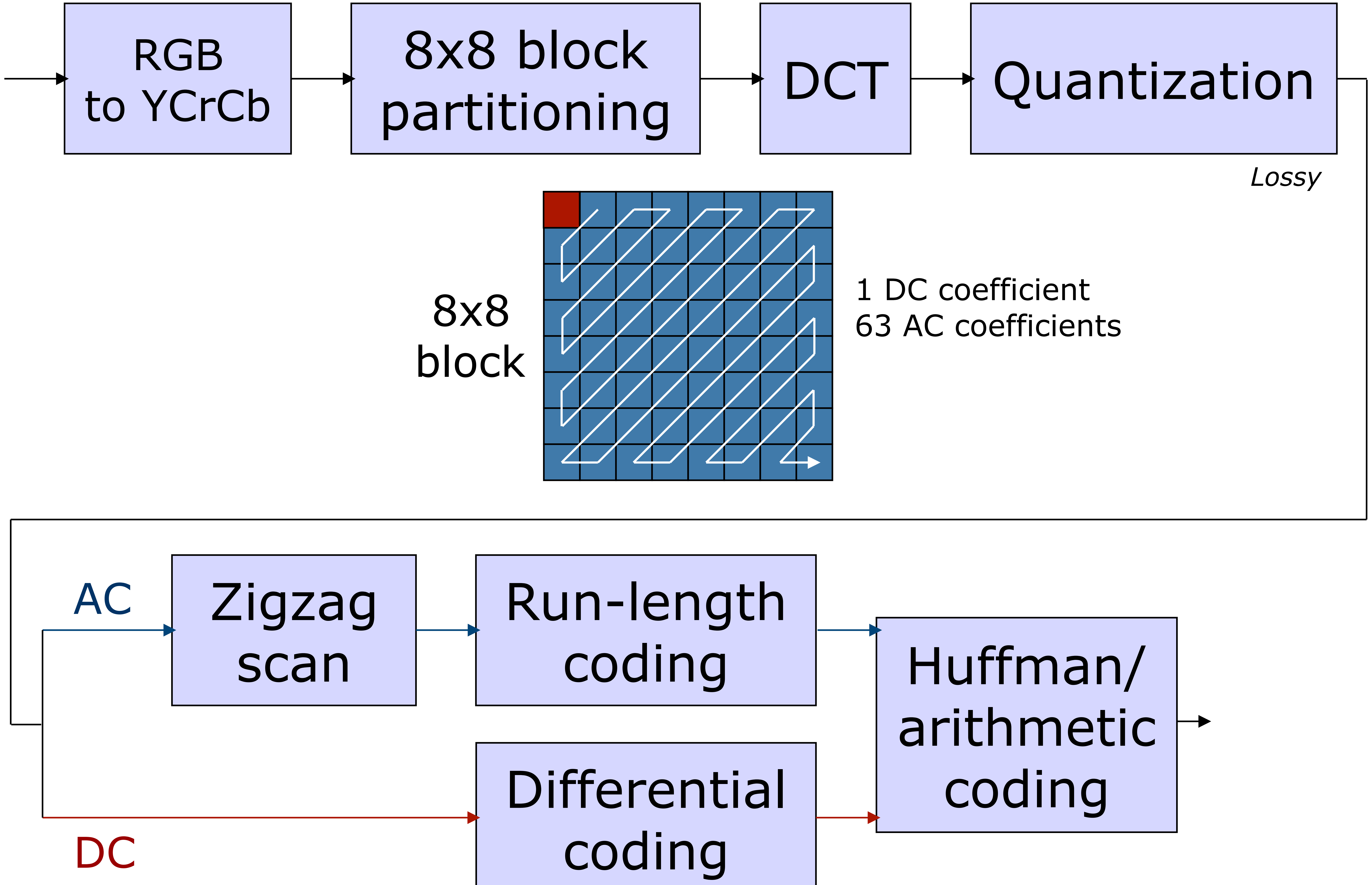
$$F^Q[u,v] = Round\left(\frac{F[u,v]}{Q[u,v]}\right)$$

- DCT coefficients F(u, v) are quantised according to a quantisation table
- High frequencies are less important (high factor)
- Quantisation table entries determine the "lossiness" of the compression

$Q[u,v]$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG Compression



The quantized DC coefficient is encoded as the difference from the DC term of the previous block to account for the strong correlation between adjacent CD coefficient

Assignment 2 available now

Computer Vision (CPSC 425): Winte

https://mattabrown.github.io/425/assignments/Assignment2.html

Assignment 2: Scaled Representations, Face Detection and Image Blending

The purpose of this assignment is to implement and understand the use of normalized cross correlation for object detection in a scaled representations. As well as to experiment with other applications of scaled representations, such as image blending.

Again, there are different ways to import libraries/modules into Python. Styles and practices vary. For consistency (and to make life easier for the markers) you are required to import modules for this assignment **exactly** as follows:

```
from PIL import Image, ImageDraw
import numpy as np
import math
from scipy import signal
import ncc
```

The assignment

Part 1: Face Detection

1. The zip file containing `ncc.py`, sample images and a face template is hw2part1.zip. Download this file to your directory and unzip with the command

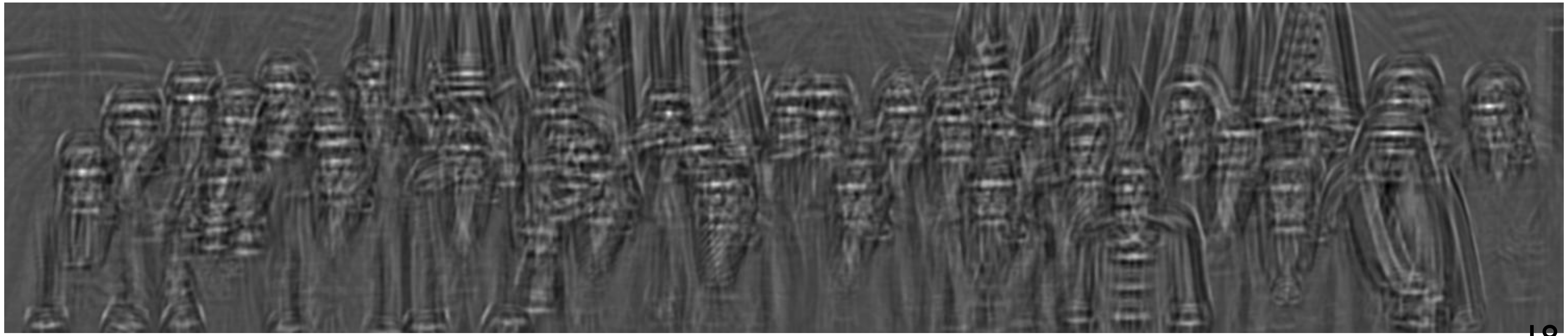
```
unzip hw2part1.zip
```

The face template we will use is in the file `face_detection_template.jpg` and is shown below.

Template Matching

- Convolve image with template, find local maxima

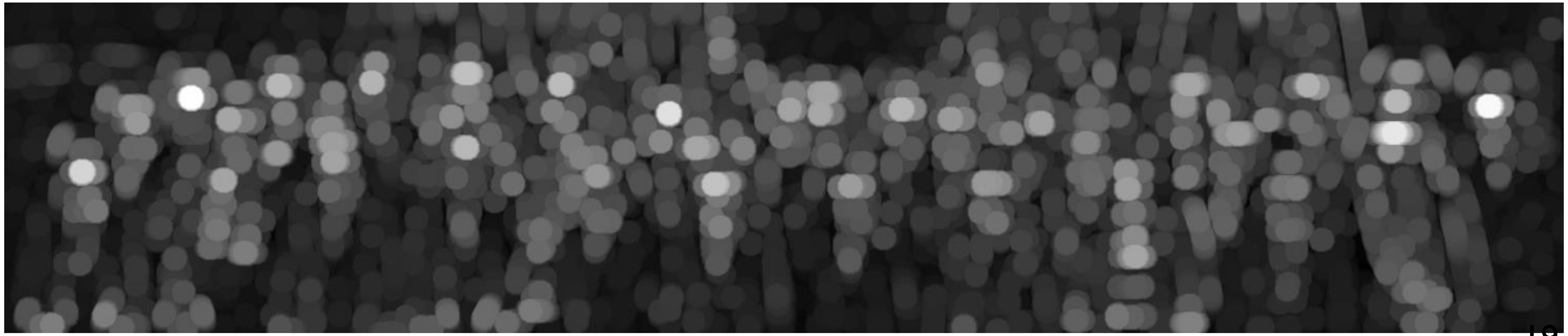
$*$  $\longrightarrow$ Non-max suppress $\longrightarrow$



Template Matching

- Convolve image with template, find local maxima

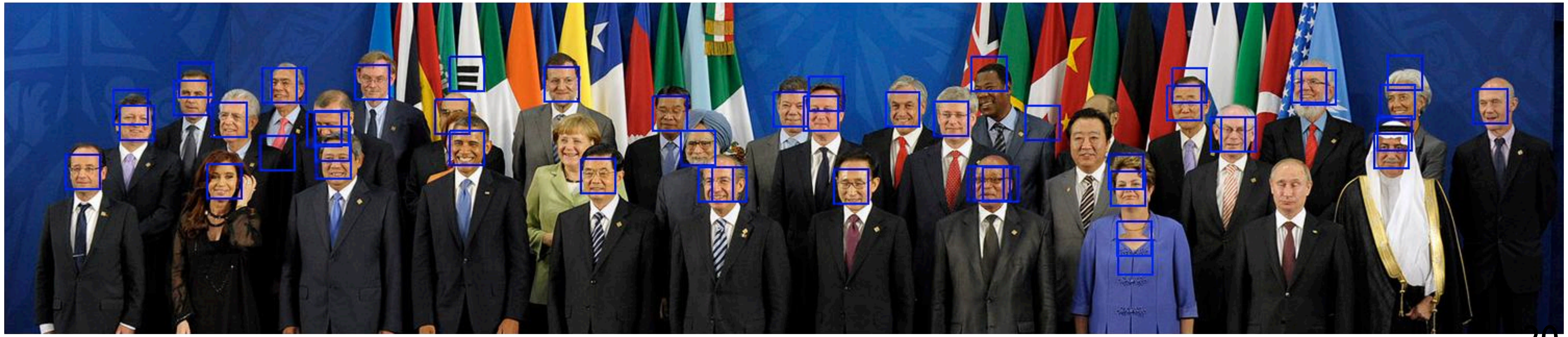
*  → Non-max suppress →



Template Matching

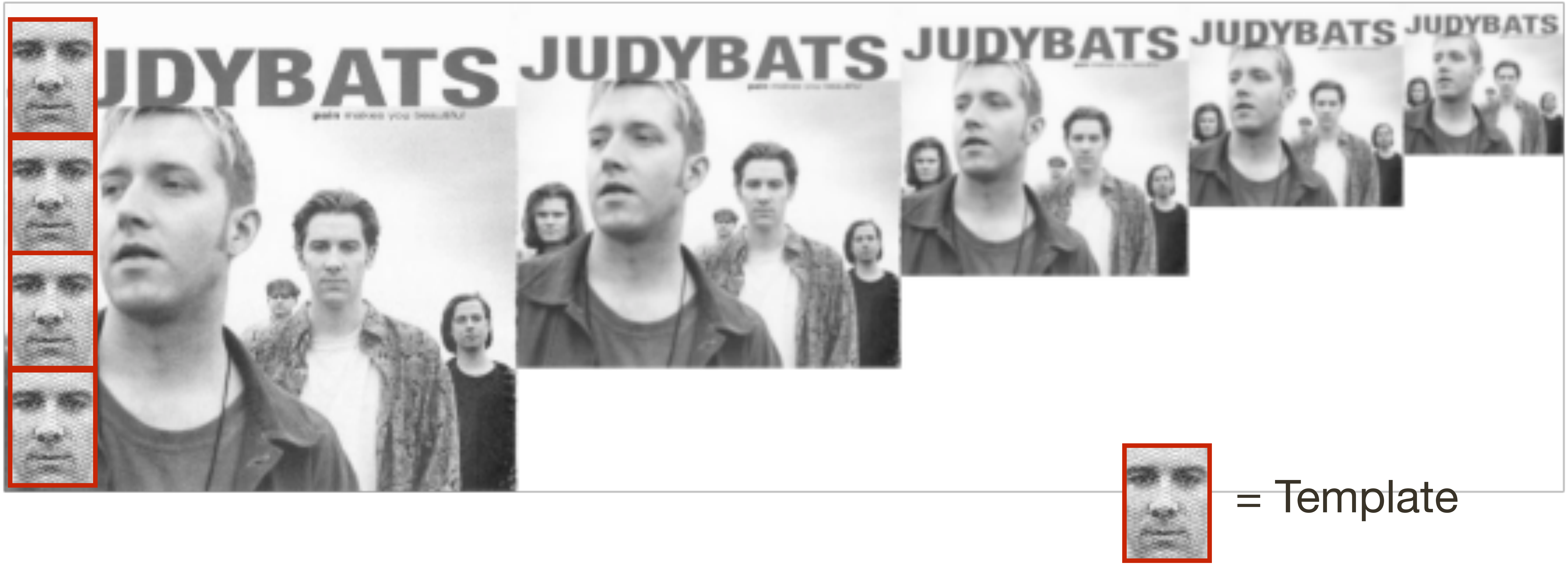
- Convolve image with template, find local maxima

$*$  $\longrightarrow$ Non-max suppress + threshold $\longrightarrow$



Multi-Scale Template Matching

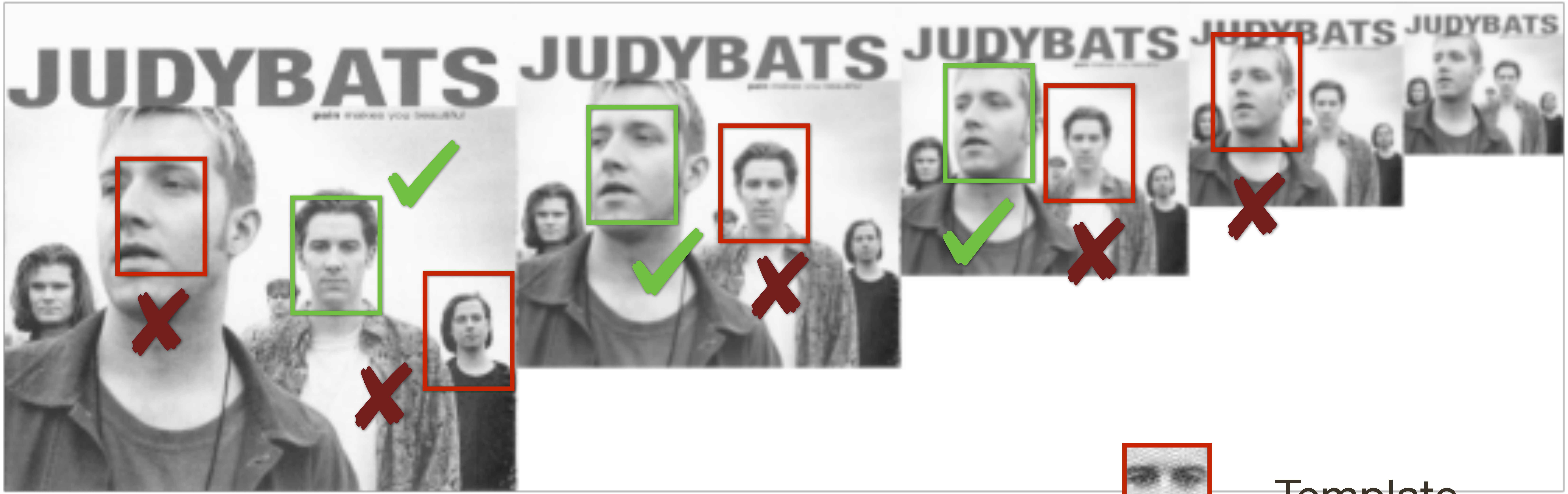
Correlation with a **fixed-sized template** only detects faces at **specific scales**



[Assignment 2]

Multi-Scale Template Matching

Correlation with a **fixed-sized template** only detects faces at **specific scales**



[Assignment 2]

Scaled Representations: Goals

to find **template matches** at all scales

— template size constant, image scale varies

— finding hands or faces when we don't know what size they are in the image

efficient search for image–to–image correspondences

— look first at coarse scales, refine at finer scales

— much less cost (but may miss best match)

to examine all **levels of detail**

— find edges with different amounts of blur

— find textures with different spatial frequencies (i.e., different levels of detail)

Template Matching

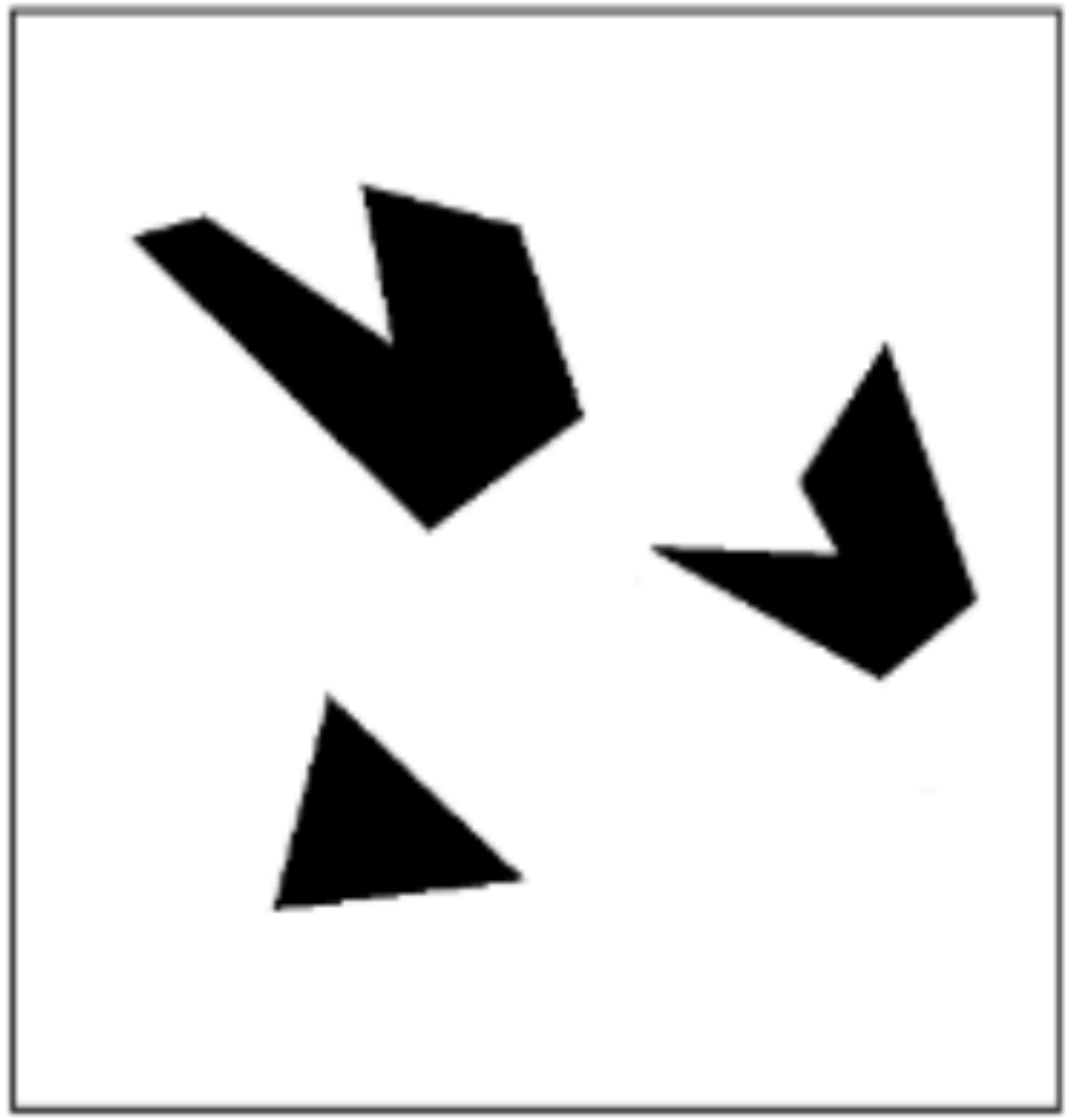
How can we find a part of one image that matches another?

or,

How can we find instances of a pattern in an image?

Key Idea: Use the pattern as a **template**

Template Matching



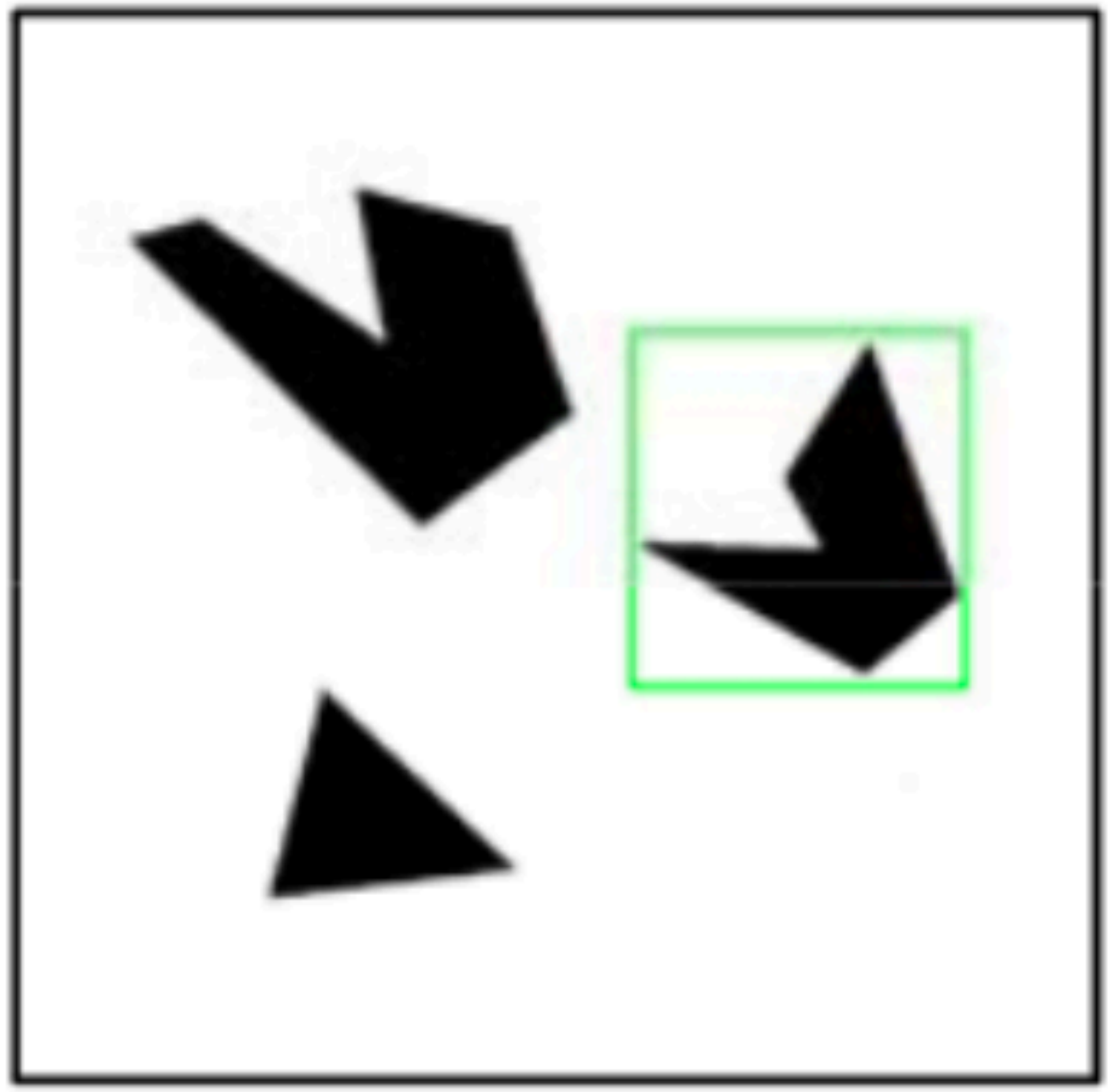
Scene

Template (mask)

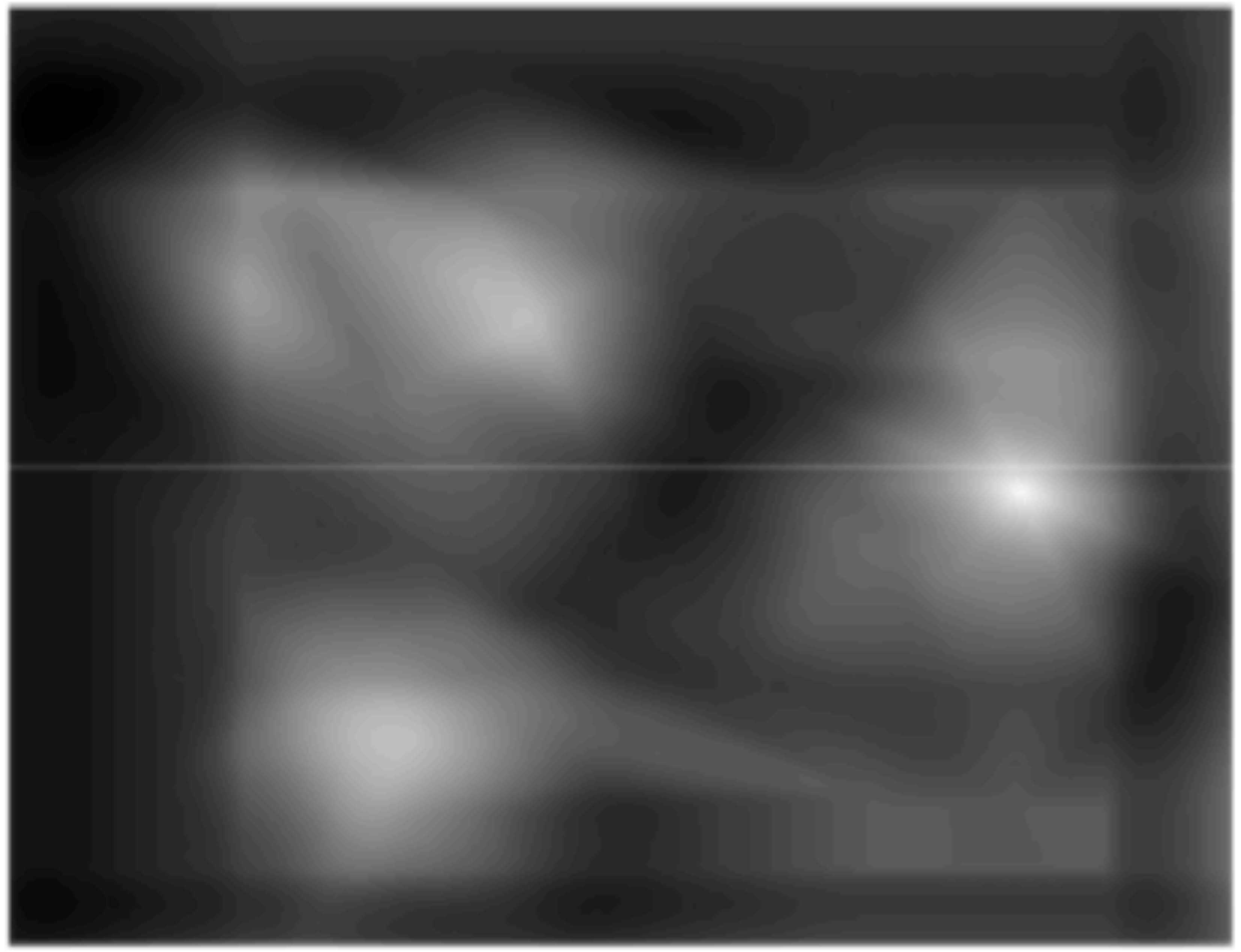
A toy example

Slide Credit: Kristen Grauman

Template Matching



Detected template

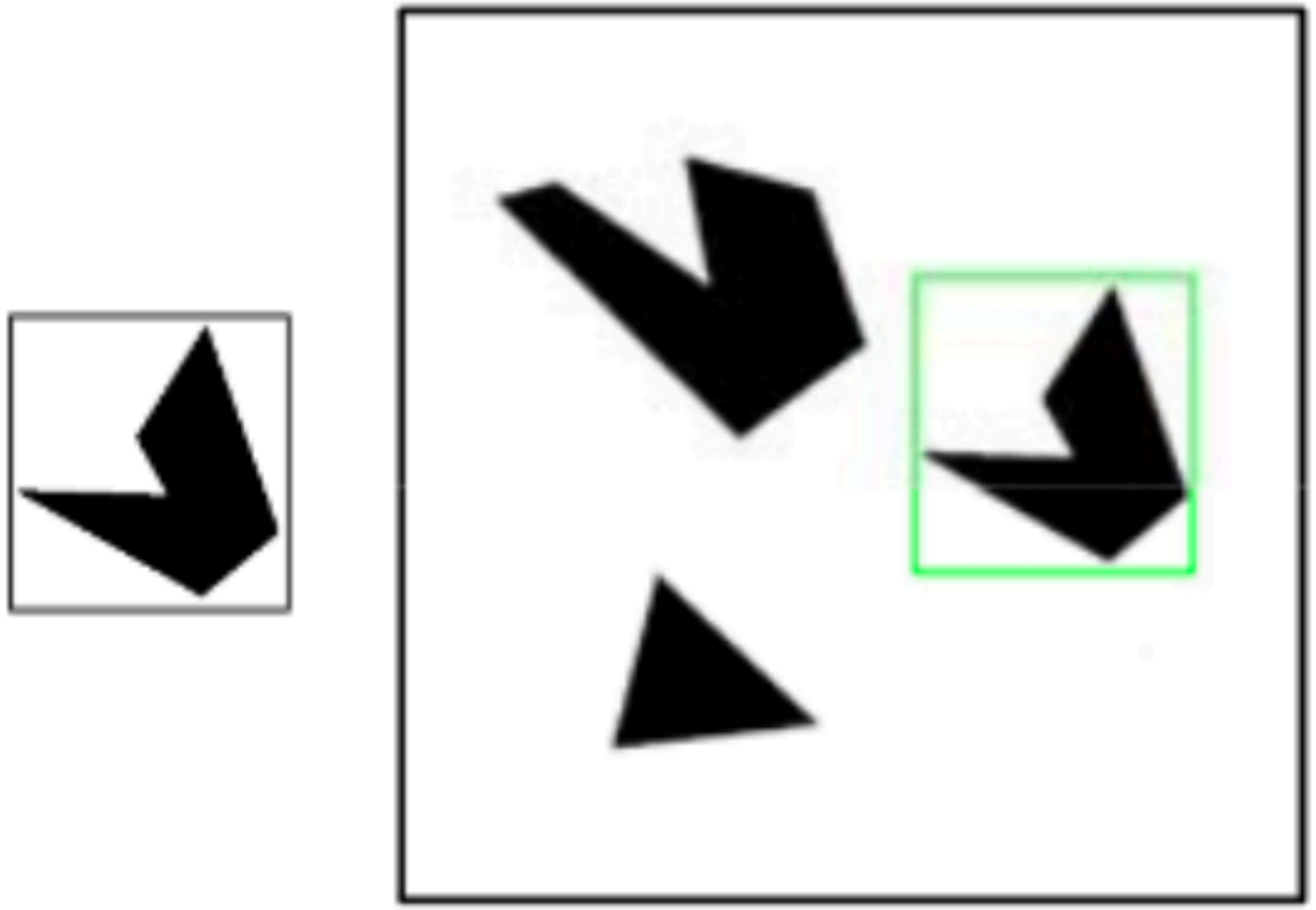


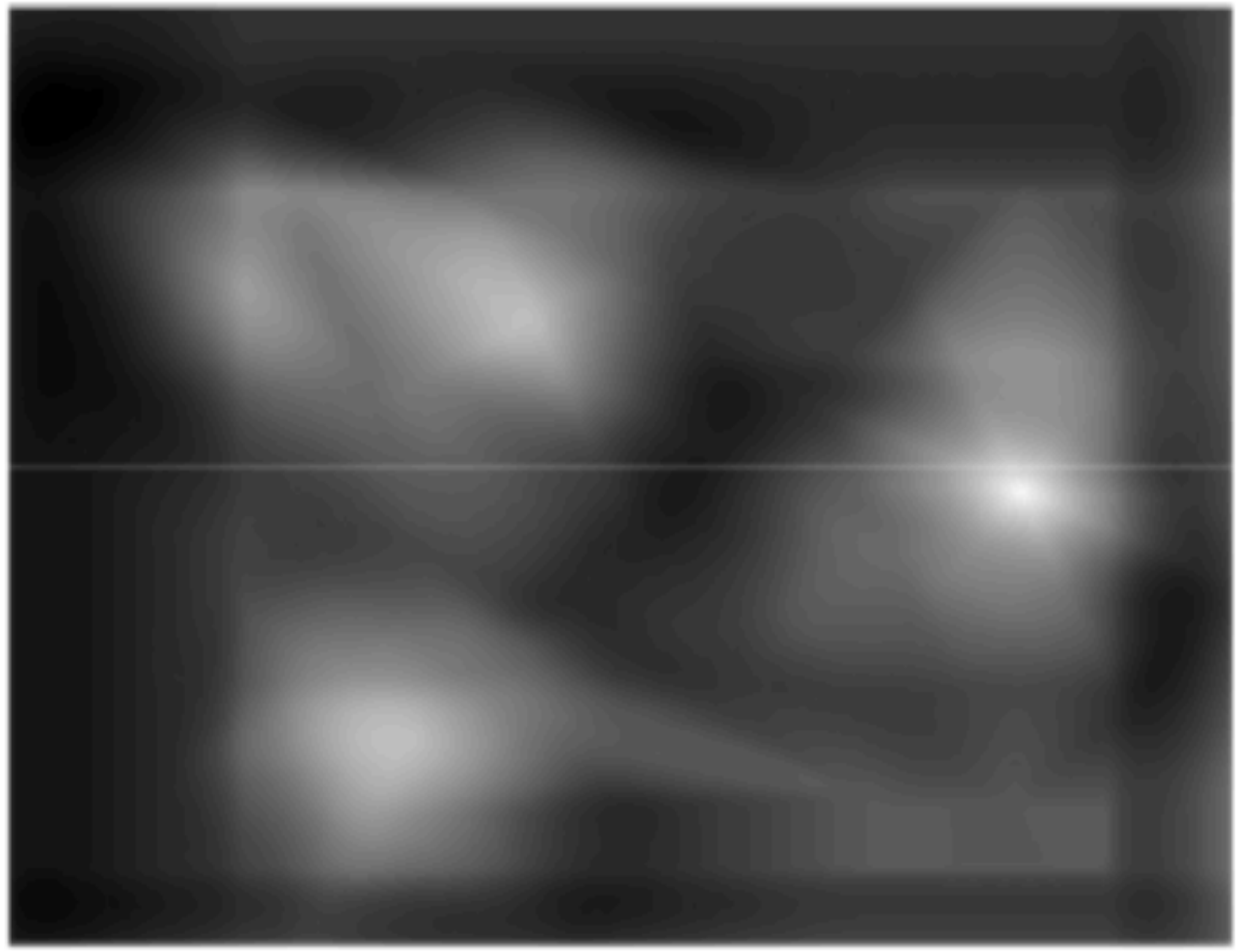
Correlation map

Slide Credit: Kristen Grauman

Template Matching

Assuming template is all positive, what does this tell us about correlation map?



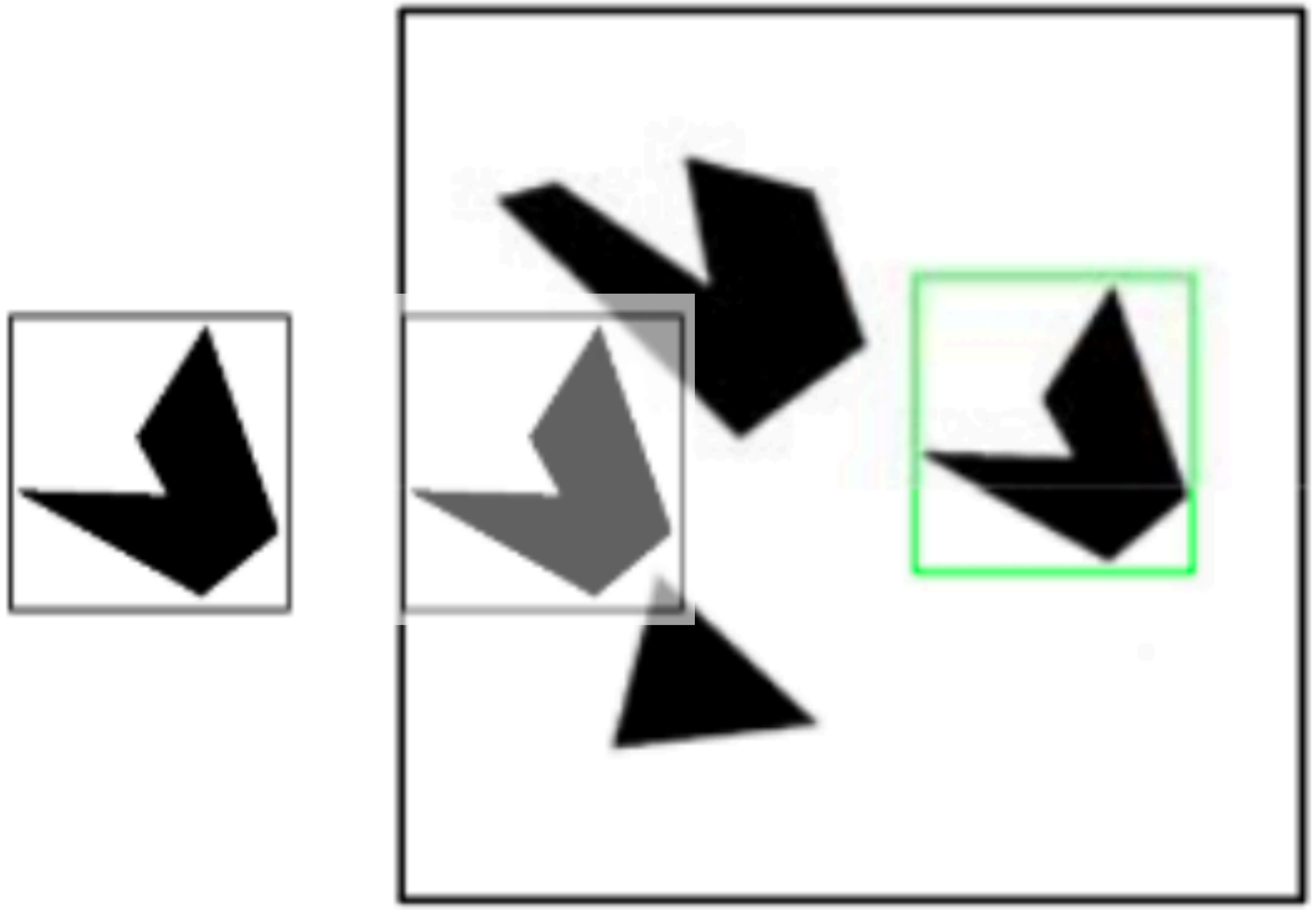


Detected template **Correlation map**

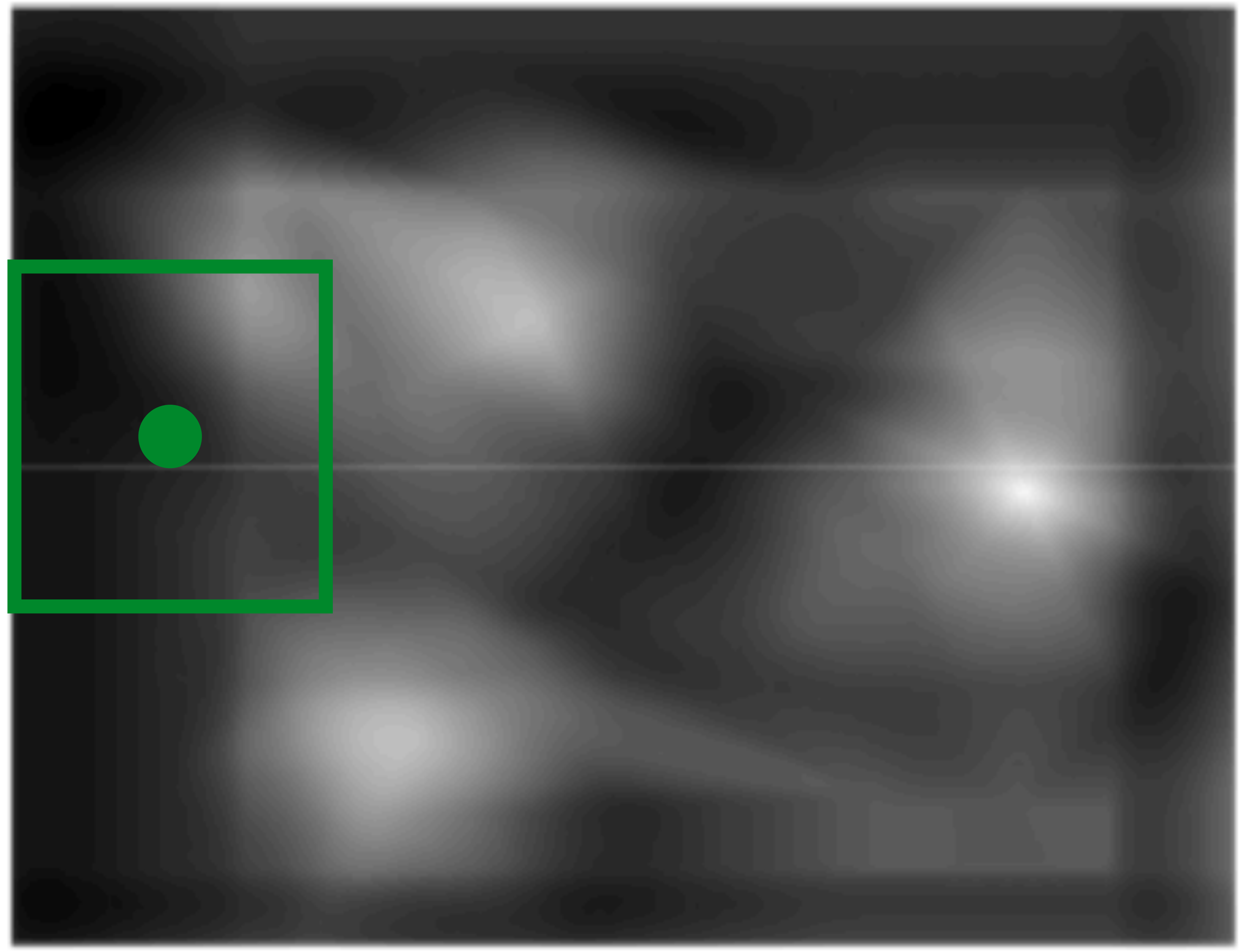
Slide Credit: Kristen Grauman

Template Matching

Assuming template is all positive, what does this tell us about correlation map?



Detected template



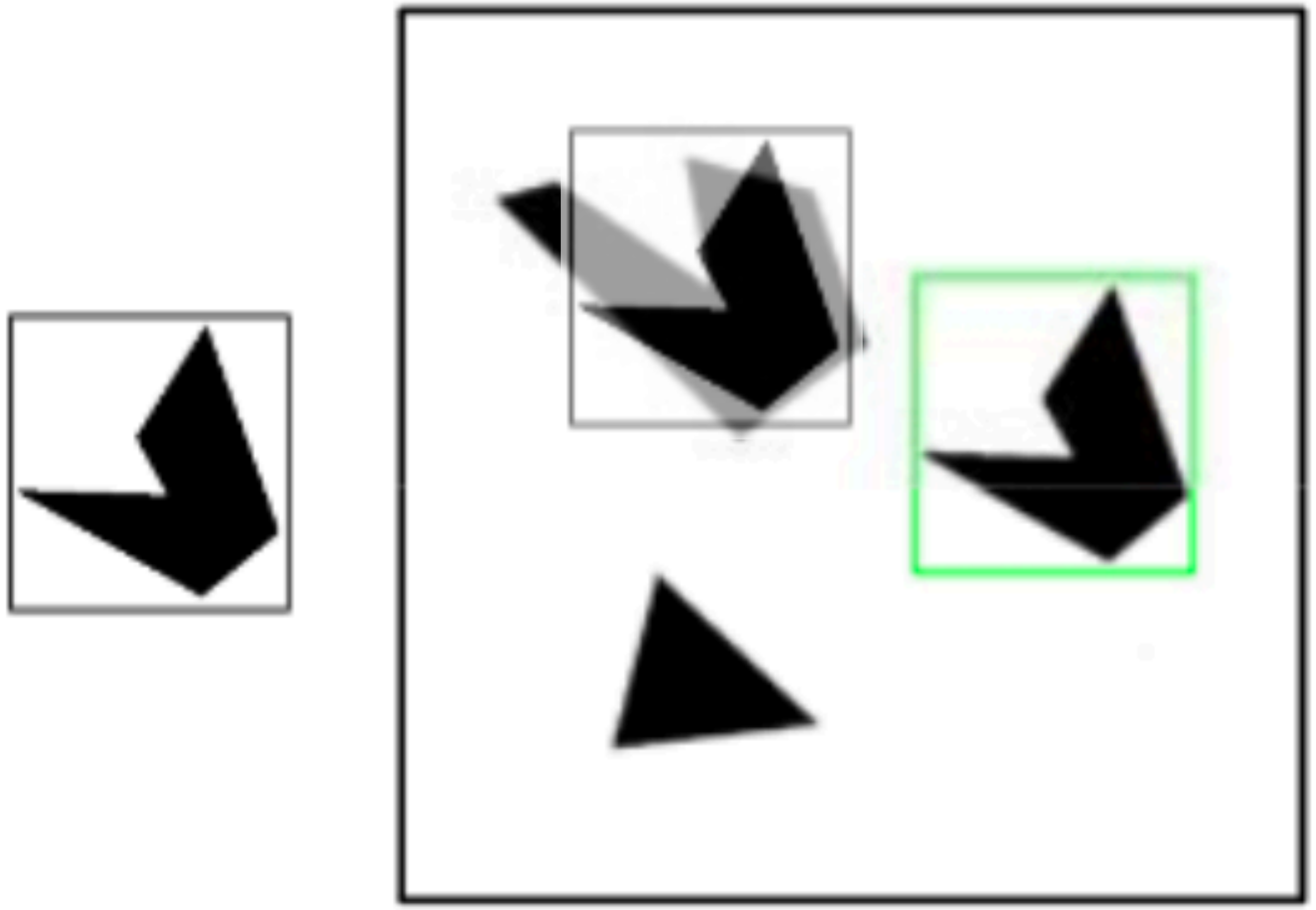
Correlation map

$$\frac{a}{|a|}\frac{b}{|b|} = ?$$

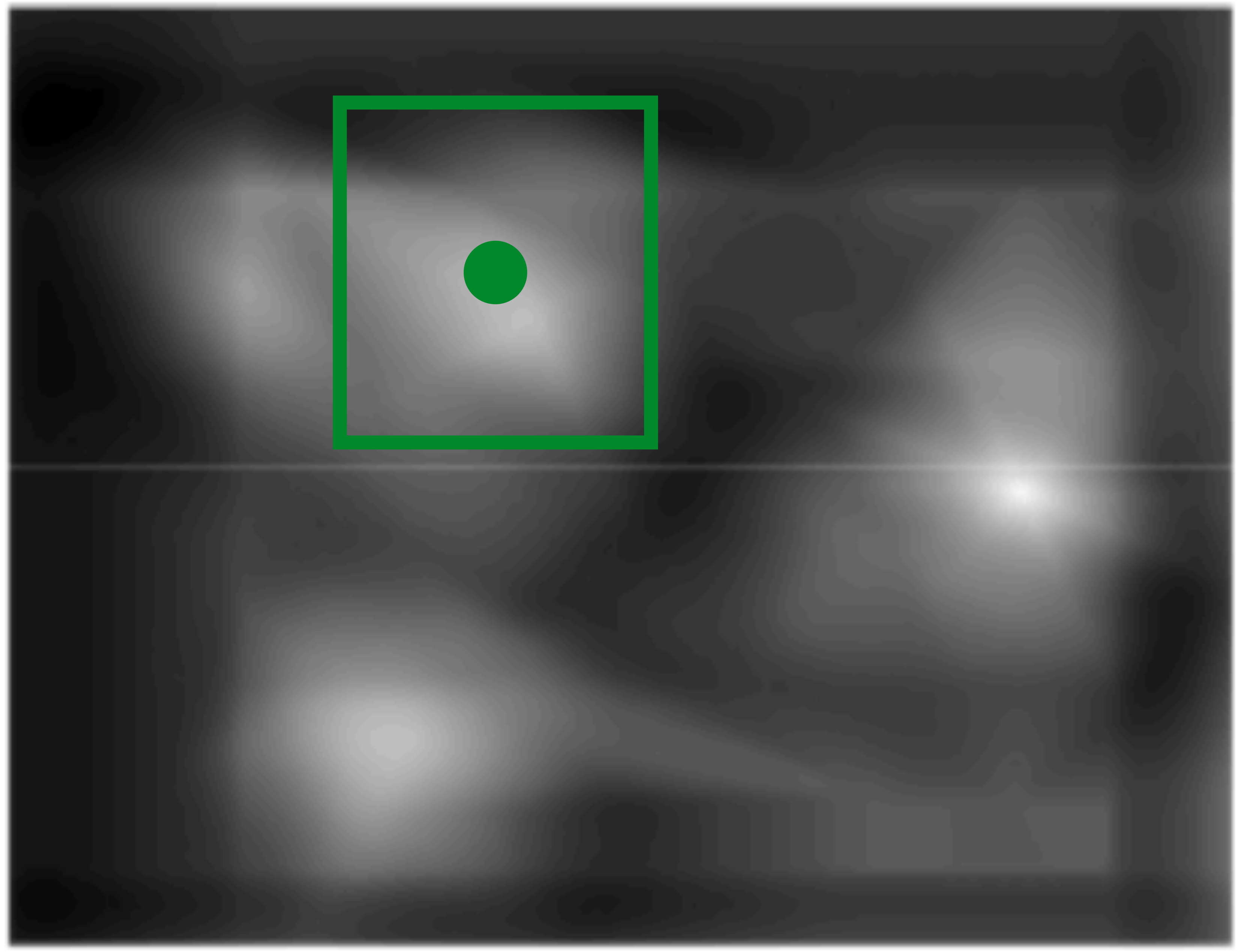
Slide Credit: Kristen Grauman

Template Matching

Assuming template is all positive, what does this tell us about correlation map?



Detected template



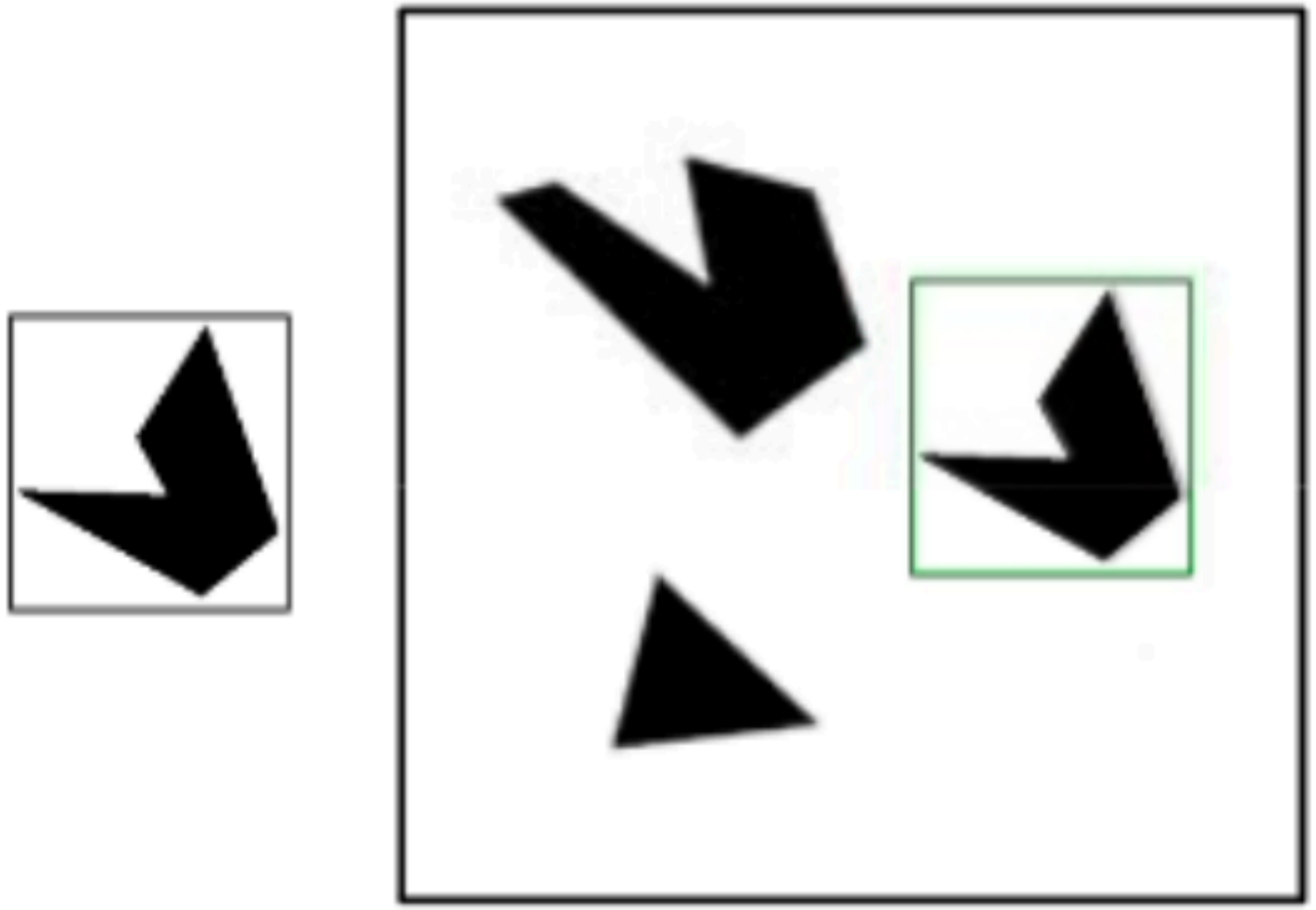
Correlation map

$$\frac{a}{|a|}\frac{b}{|b|} = ?$$

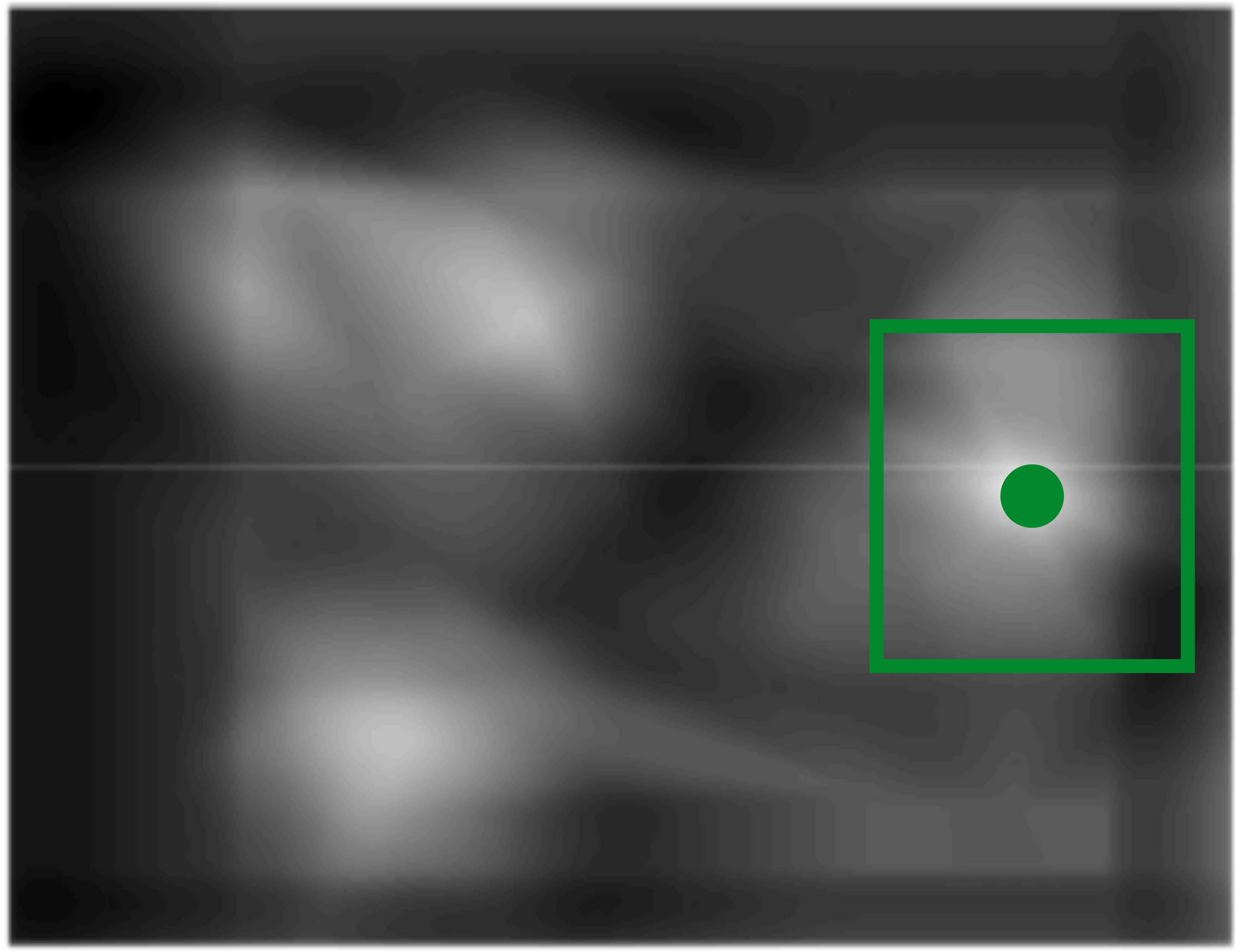
Slide Credit: Kristen Grauman

Template Matching

Assuming template is all positive, what does this tell us about correlation map?



Detected template



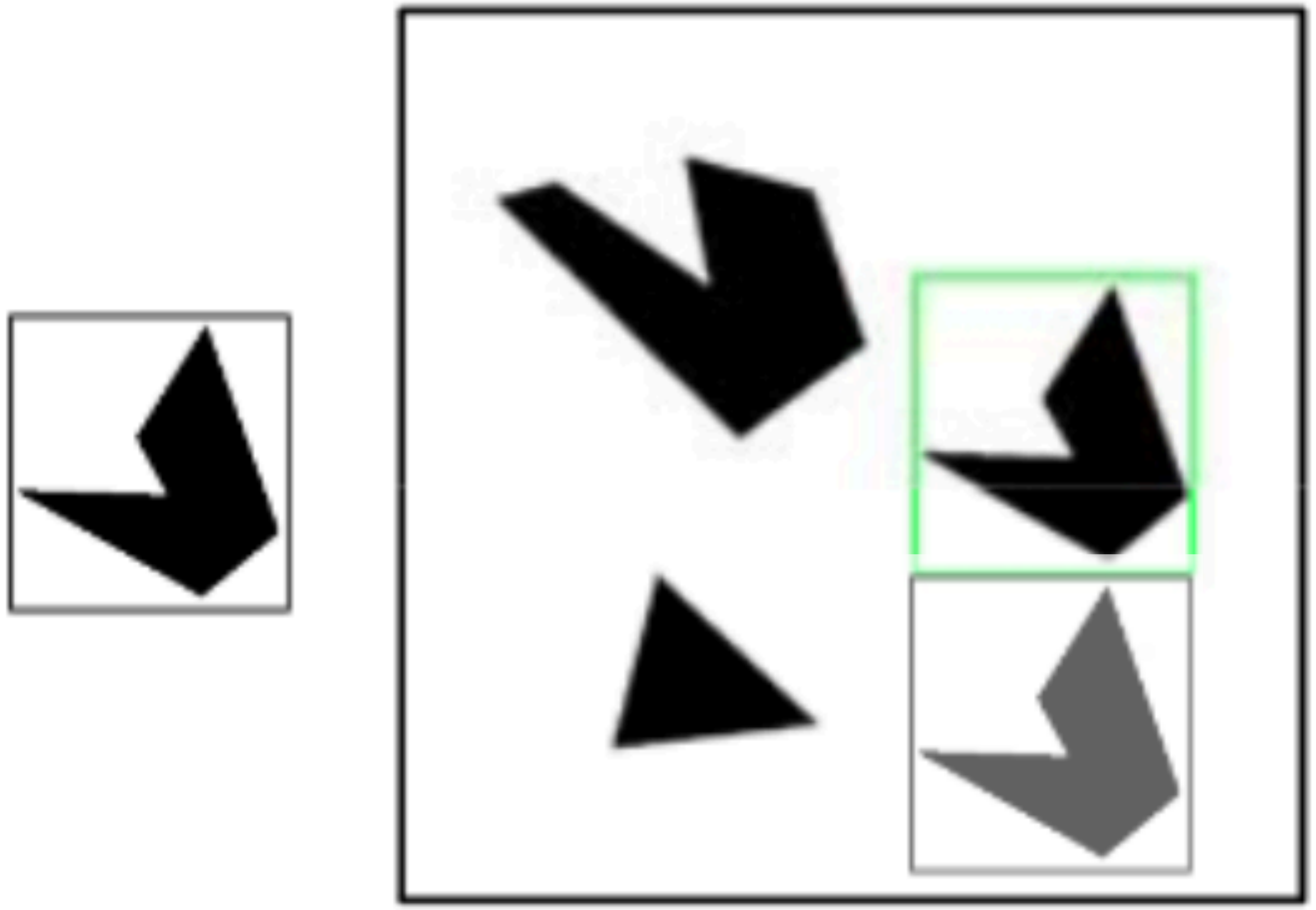
Correlation map

$$\frac{a}{|a|} \frac{b}{|b|} = ?$$

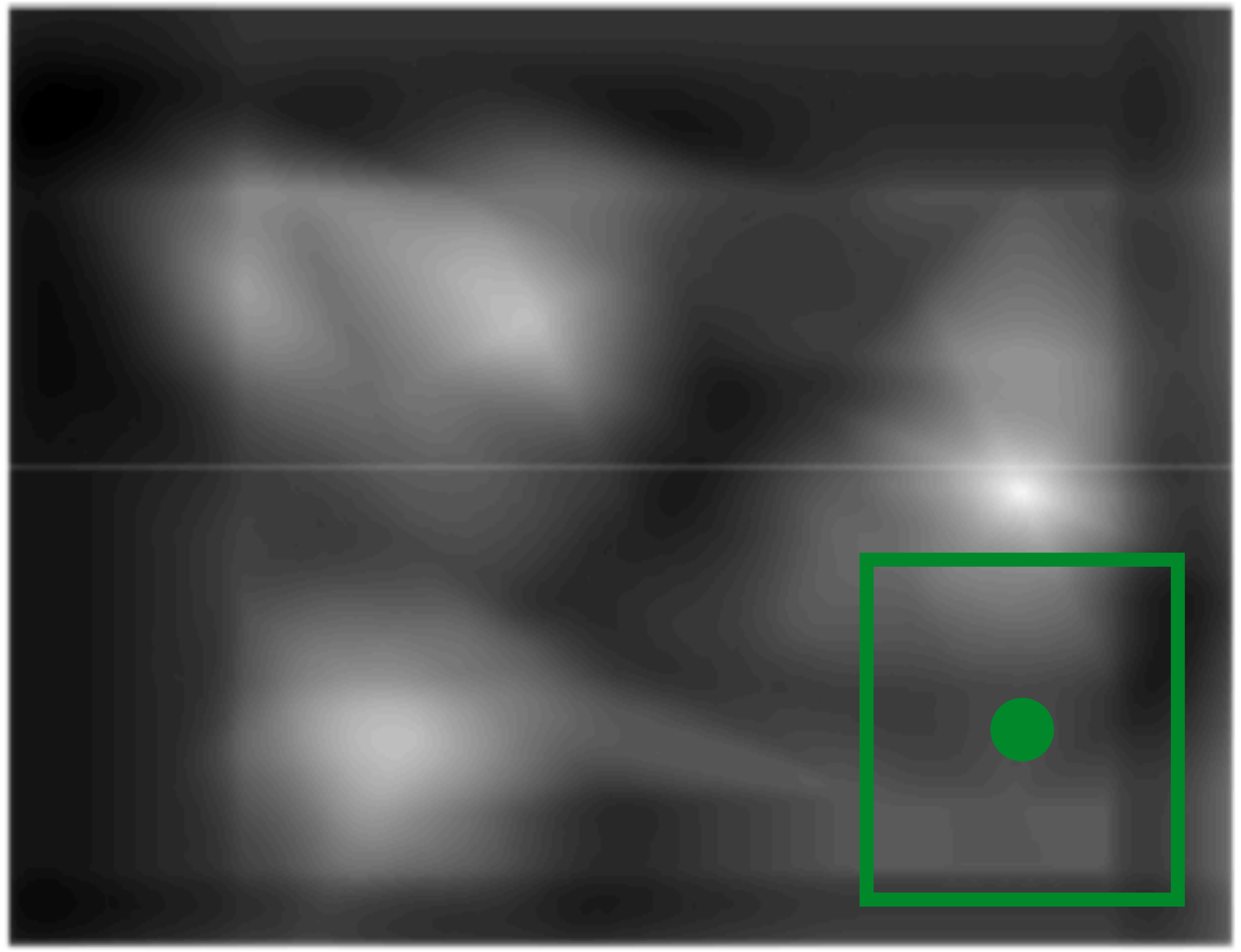
Slide Credit: Kristen Grauman

Template Matching

Assuming template is all positive, what does this tell us about correlation map?



Detected template



Correlation map

$$\frac{a}{|a|}\frac{b}{|b|} = ?$$

Slide Credit: Kristen Grauman

Template Matching

We can think of convolution/**correlation** as comparing a template (the filter) with each local image patch.

— Consider the filter and image patch as vectors.

— Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.

Template Matching

We can think of convolution/**correlation** as comparing a template (the filter) with each local image patch.

— Consider the filter and image patch as vectors.

— Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.

Template

0	0	0
0	1	0
0	1	1

Template Matching

We can think of convolution/**correlation** as comparing a template (the filter) with each local image patch.

— Consider the filter and image patch as vectors.

— Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.

Image Patch 1

0	0	0
0	1	0
0	1	1

Image Patch 2

1	0	1
0	1	0
0	0	0

Template

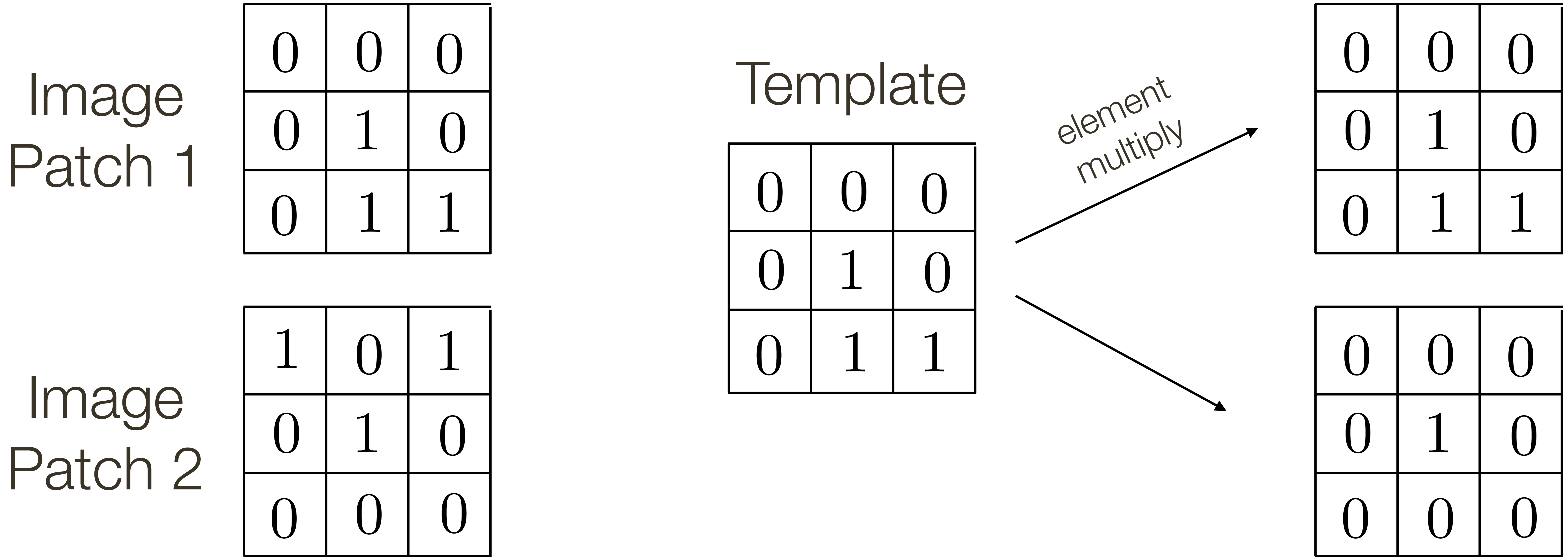
0	0	0
0	1	0
0	1	1

Template Matching

We can think of convolution/**correlation** as comparing a template (the filter) with each local image patch.

— Consider the filter and image patch as vectors.

— Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.



Template Matching

We can think of convolution/**correlation** as comparing a template (the filter) with each local image patch.

— Consider the filter and image patch as vectors.

— Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.

Image Patch 1

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

Image Patch 2

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

Template

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

element multiply

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = 3$$

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

Template Matching

We can think of convolution/**correlation** as comparing a template (the filter) with each local image patch.

— Consider the filter and image patch as vectors.

— Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.

Image Patch 1 $\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 1 & 1 \\ \hline\end{array}$

Image Patch 2 $\begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array}$ $\times 255$

Template $\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 1 & 1 \\ \hline\end{array}$

element multiply

$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 1 & 1 \\ \hline\end{array} = 3$

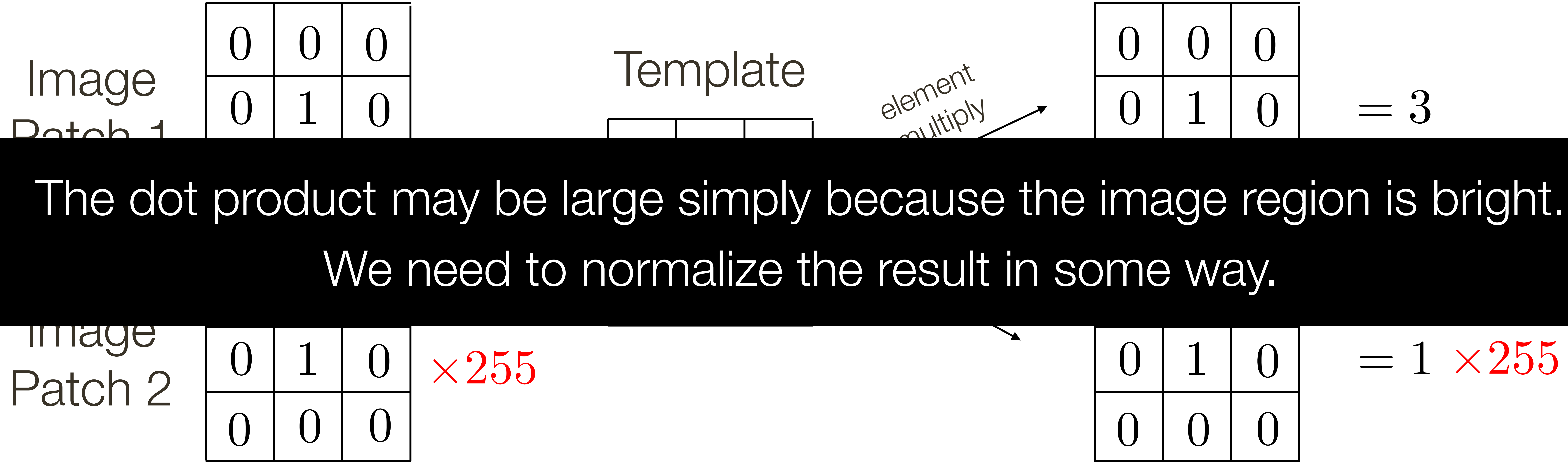
$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array} = 1 \times 255$

Template Matching

We can think of convolution/**correlation** as comparing a template (the filter) with each local image patch.

— Consider the filter and image patch as vectors.

— Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.



The dot product may be large simply because the image region is bright.
We need to normalize the result in some way.

Template Matching

7.3 Correlation, Normalised Correlation, SSD

Template Matching

Similarity measures between a filter $\mathbf{J}$ local image region $\mathbf{I}$

Correlation, CORR = $\mathbf{I} \cdot \mathbf{J} = \mathbf{I}^T \mathbf{J}$

Normalised Correlation, NCORR = $\dfrac{\mathbf{I}^T \mathbf{J}}{|\mathbf{I}||\mathbf{J}|} = \cos\theta$

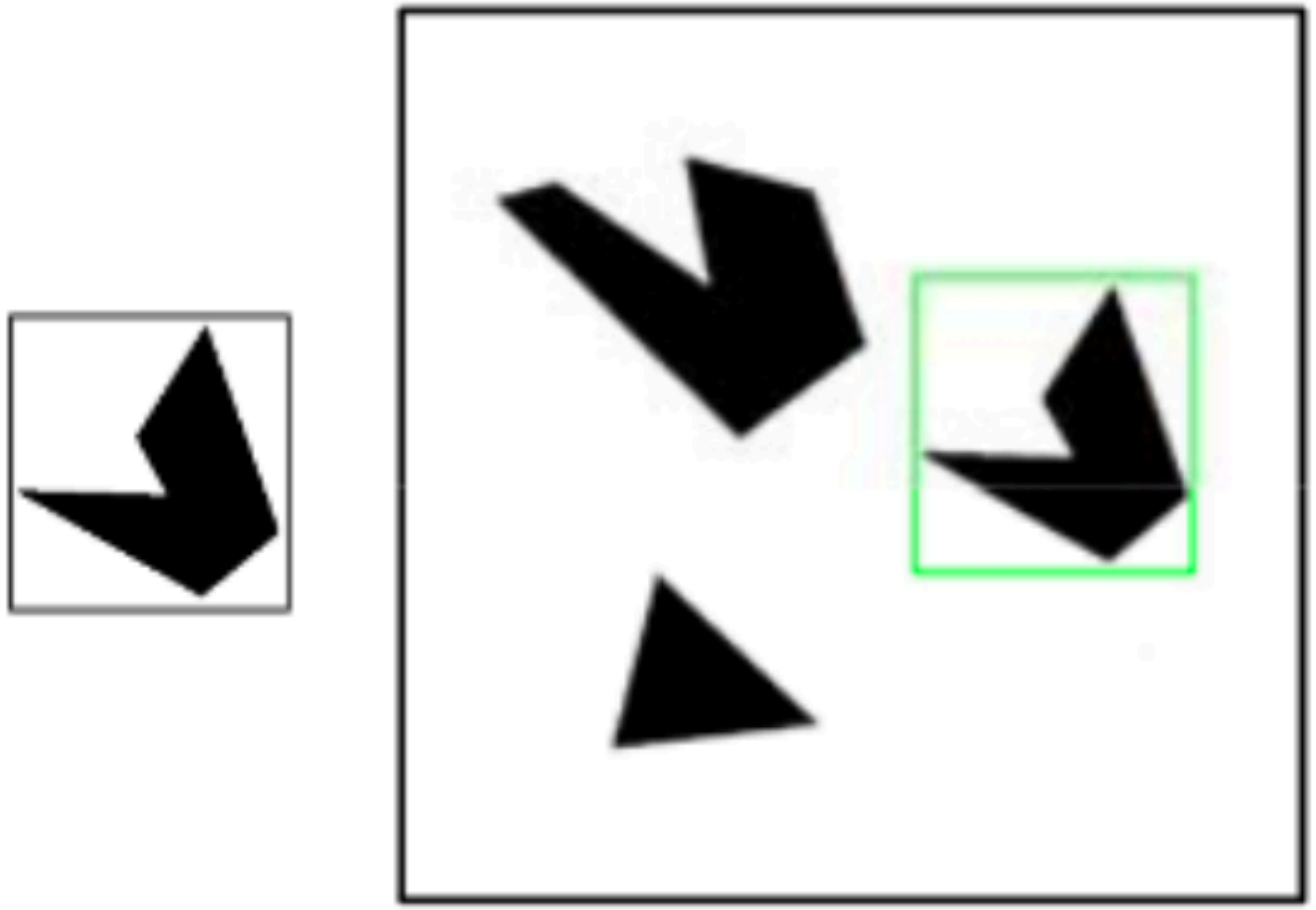
Sum Squared Difference, SSD = $|\mathbf{I} - \mathbf{J}|^2$

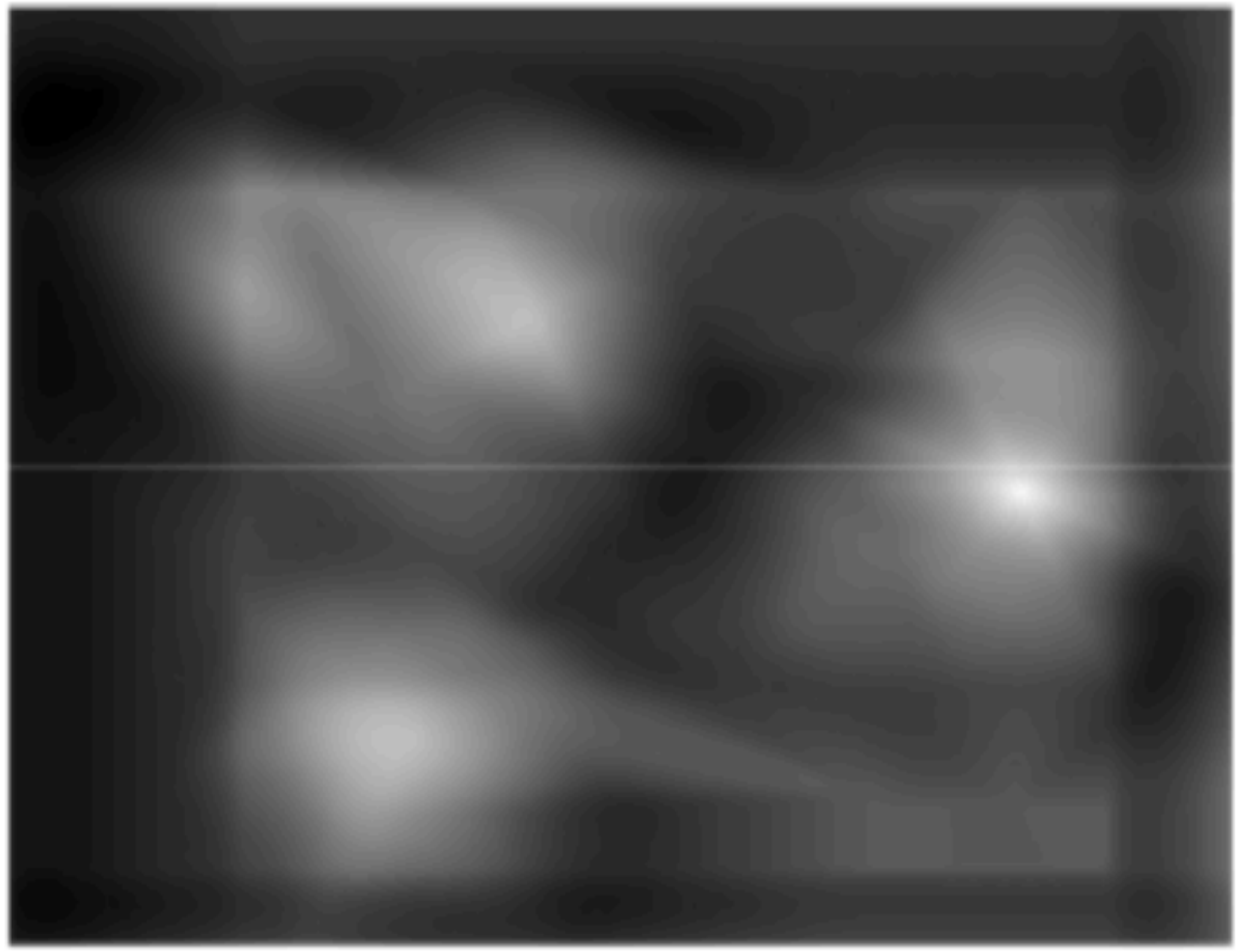
Normalized correlation varies between –1 and 1, attains the value 1 when the filter and image region are identical (up to a scale factor)

Minimising SSD and maximizing Normalized Correlation are equivalent if $|\mathbf{I}| = |\mathbf{J}| = 1$

Template Matching

Detection can be done by comparing correlation map score to a threshold





Detected template

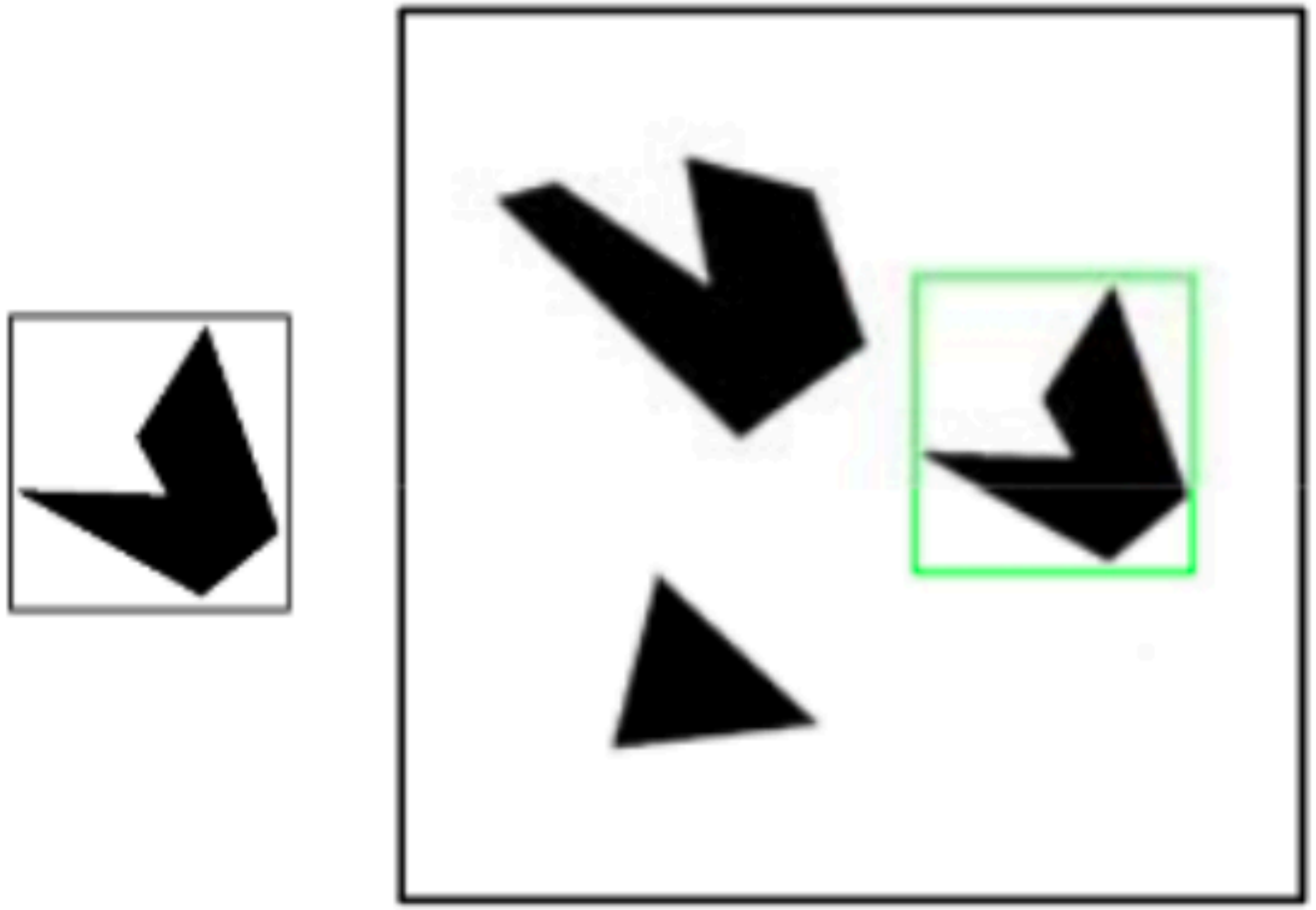
Correlation map

What happens if the threshold is relatively low?

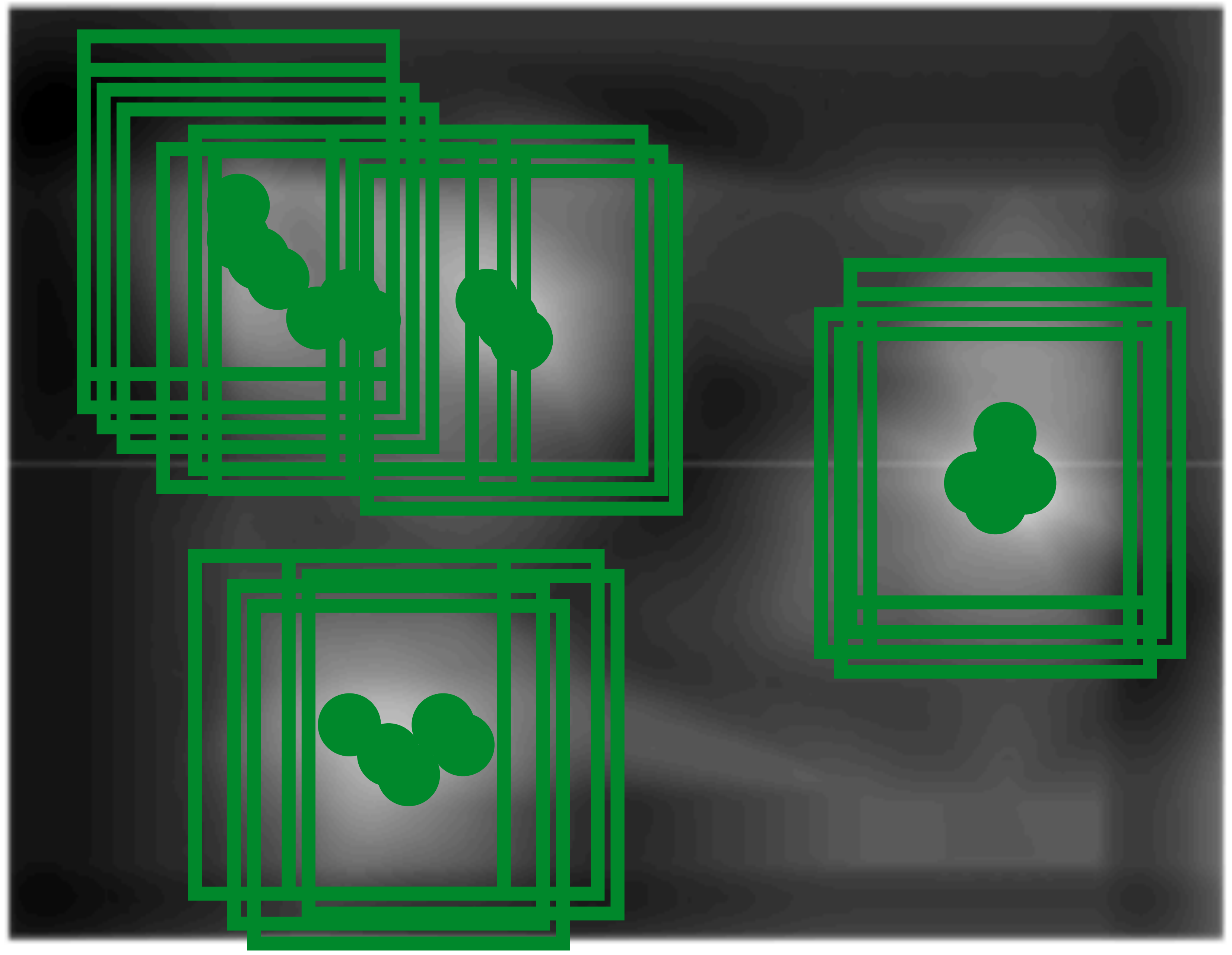
Slide Credit: Kristen Grauman

Template Matching

Detection can be done by comparing correlation map score to a threshold



Detected template



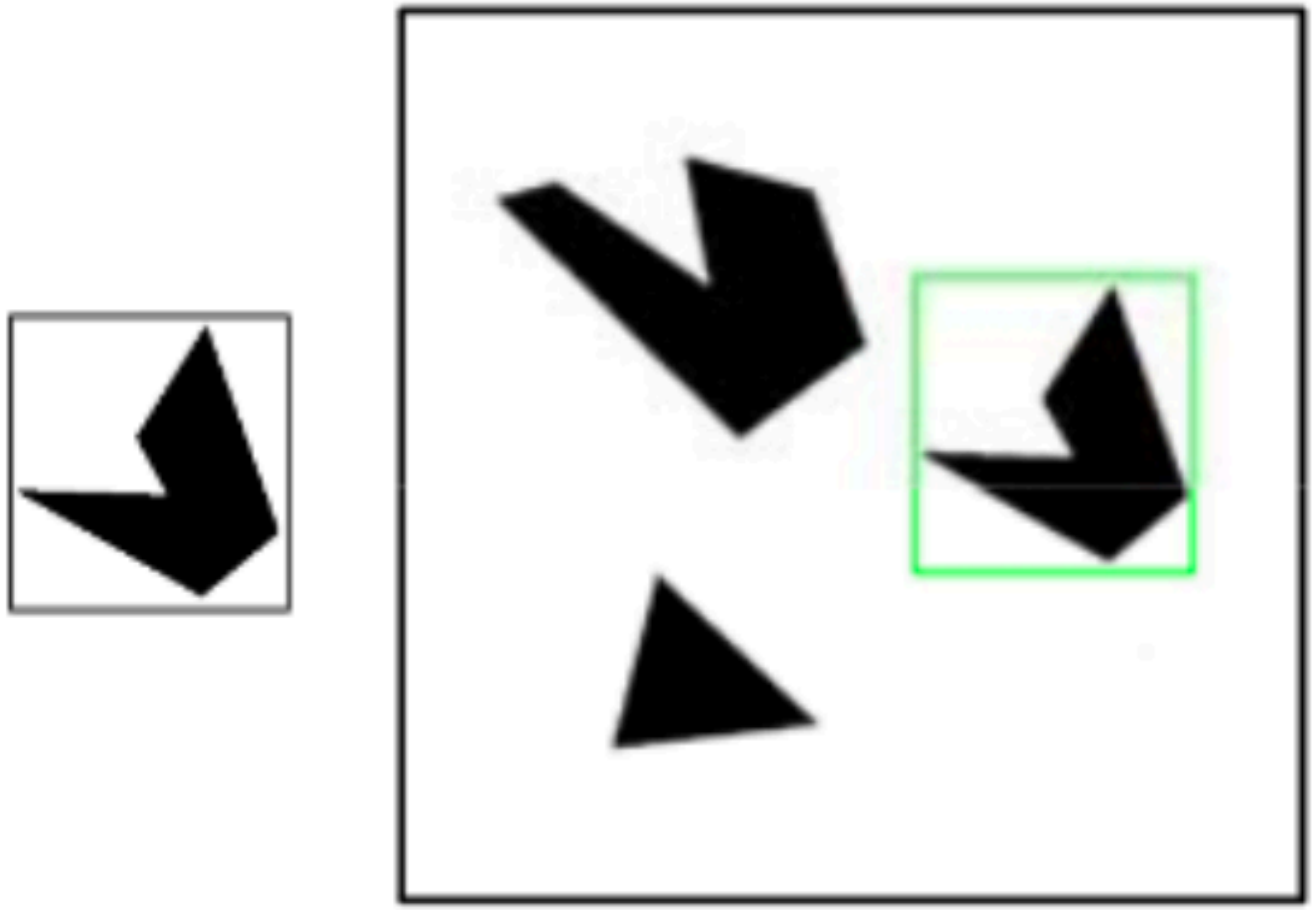
Correlation map

What happens if the threshold is relatively low?

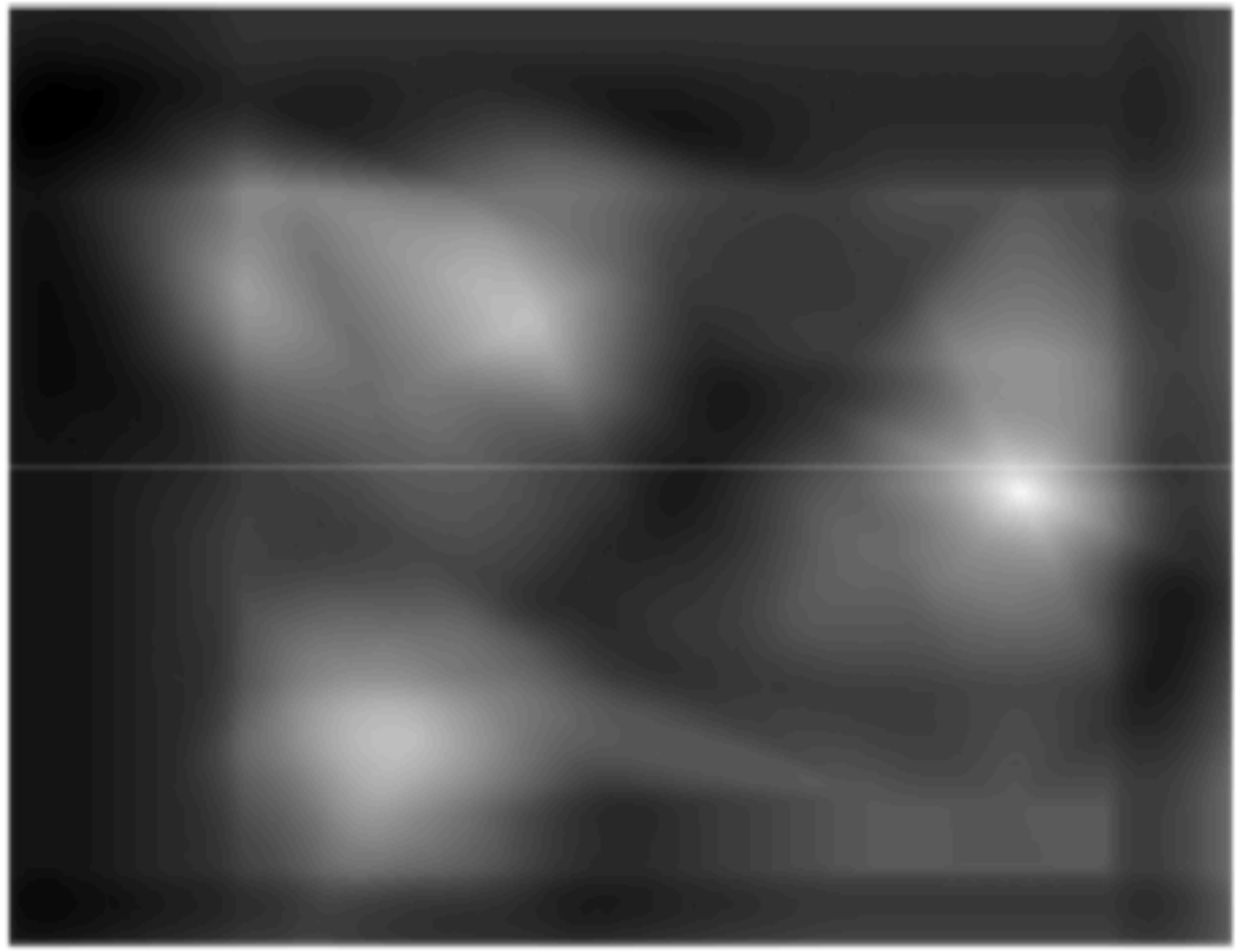
Slide Credit: Kristen Grauman

Template Matching

Detection can be done by comparing correlation map score to a threshold



Detected template



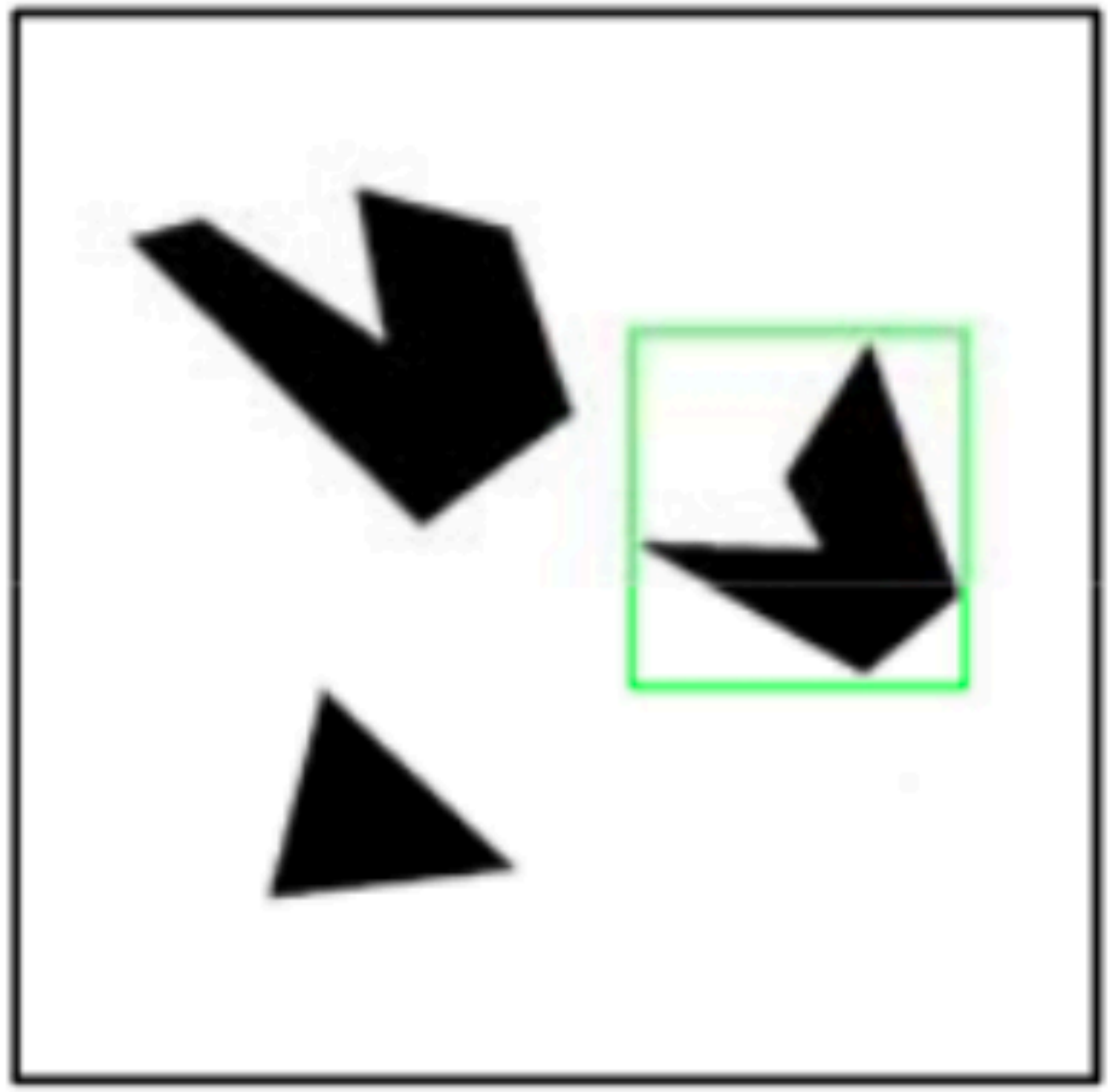
Correlation map

What happens if the threshold is very high (e.g., 0.99)?

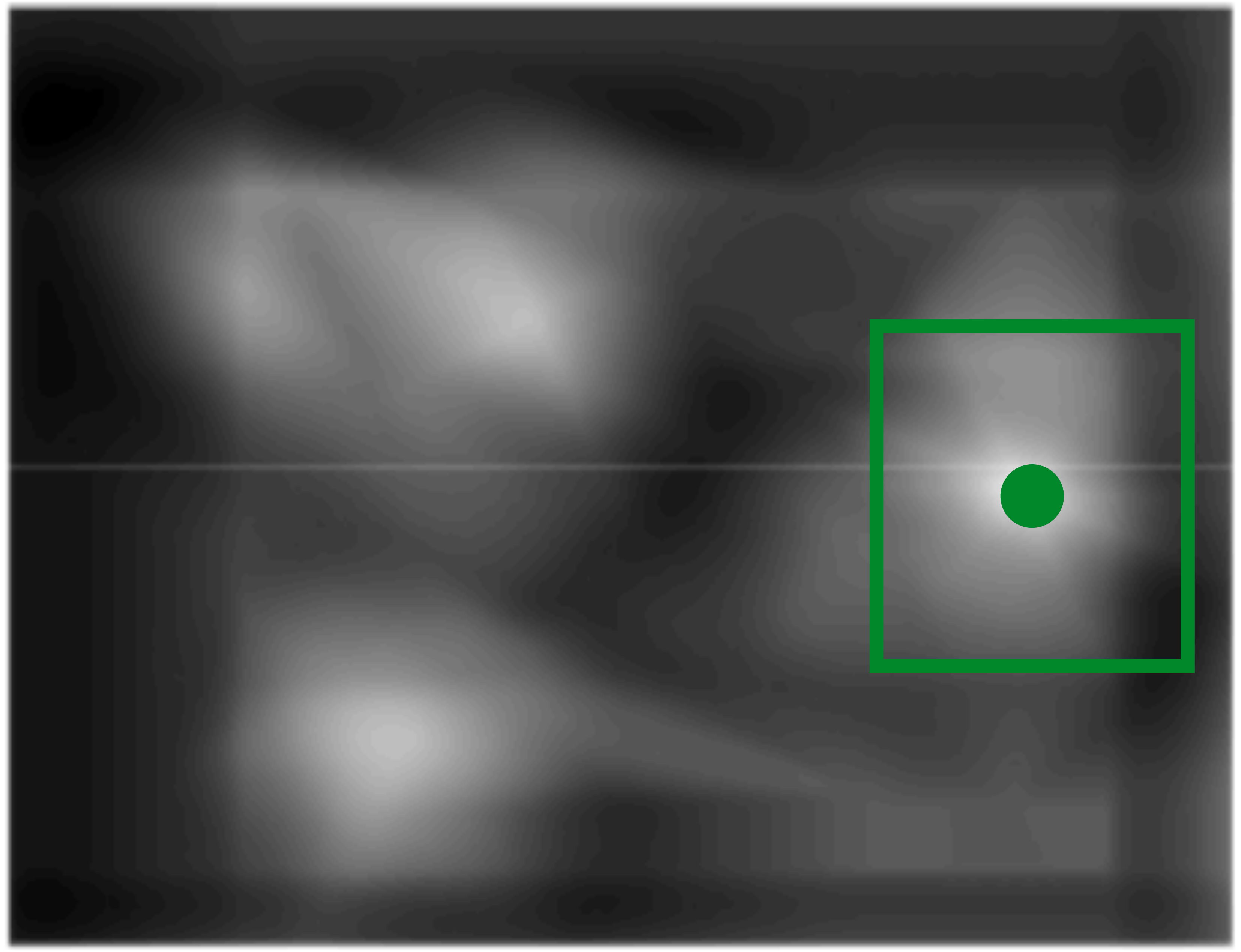
Slide Credit: Kristen Grauman

Template Matching

Detection can be done by comparing correlation map score to a threshold



Detected template



Correlation map

What happens if the threshold is very high (e.g., 0.99)?

Slide Credit: Kristen Grauman

Normalised Correlation — Efficient Implementation

Let a and b be vectors. Let θ be the angle between them. We know

$$\cos\theta = \frac{a \cdot b}{|a||b|} = \frac{a \cdot b}{\sqrt{(a \cdot a)(b \cdot b)}} = \frac{a}{|a|} \bullet \frac{b}{|b|}$$

where · is dot product and | | is vector magnitude

1. Normalize the template / filter (b) in the beginning
2. Compute norm of $|a|$ by convolving squared image with a filter of all 1's of equal size to the the template and square rooting the response
3. We can compute the dot product by convolution of image (a) with normalized filter (b)
4. We can finally compute the normalized correlation by dividing element-wise result in Step 3 by result ins Step 2

Template Matching

Linear filtering the entire image computes the entire set of dot products, one for each possible alignment of filter and image

Important **Insight**:

— filters look like the pattern they are intended to find

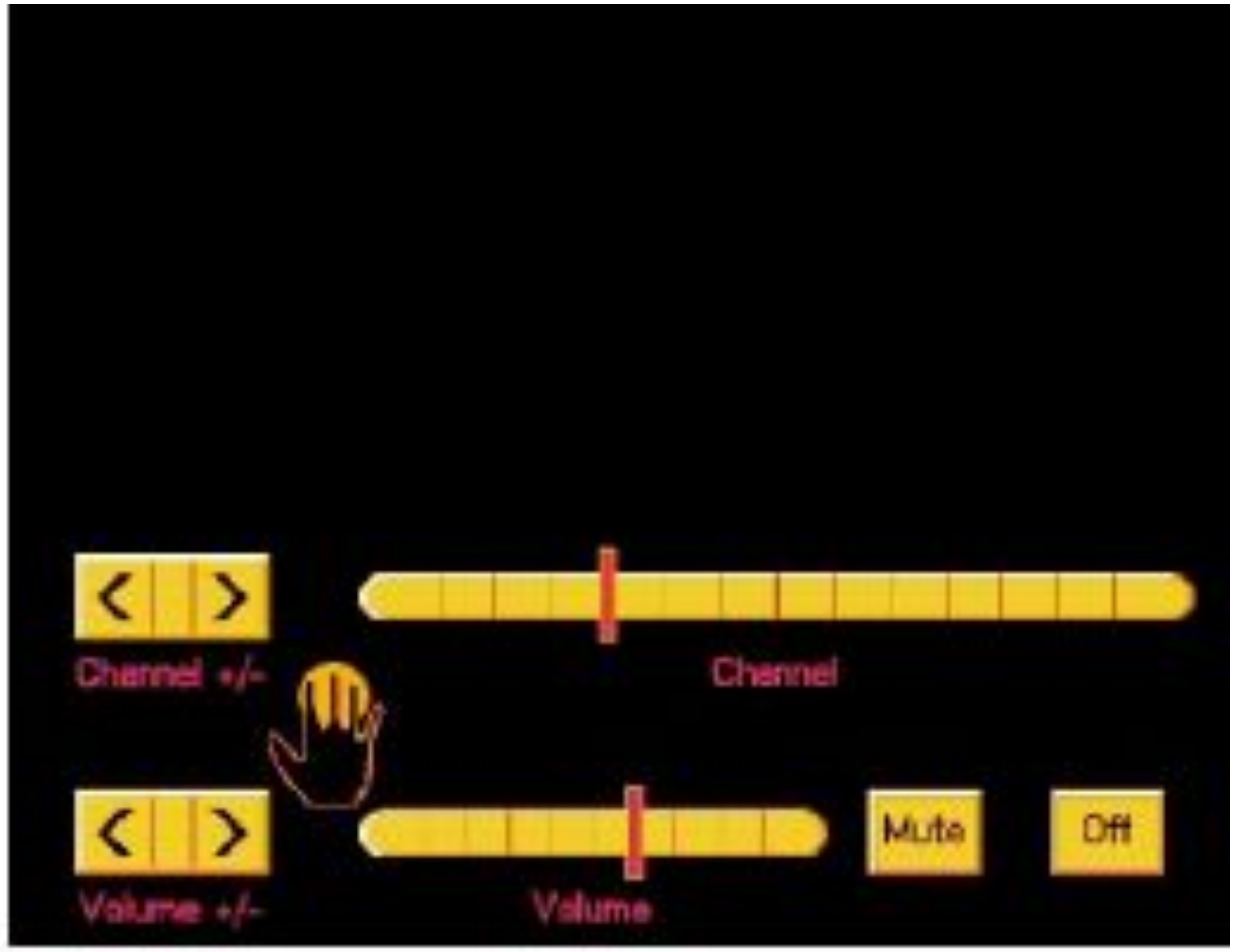
— filters find patterns they look like

Linear filtering is sometimes referred to as **template matching**

Example 1:

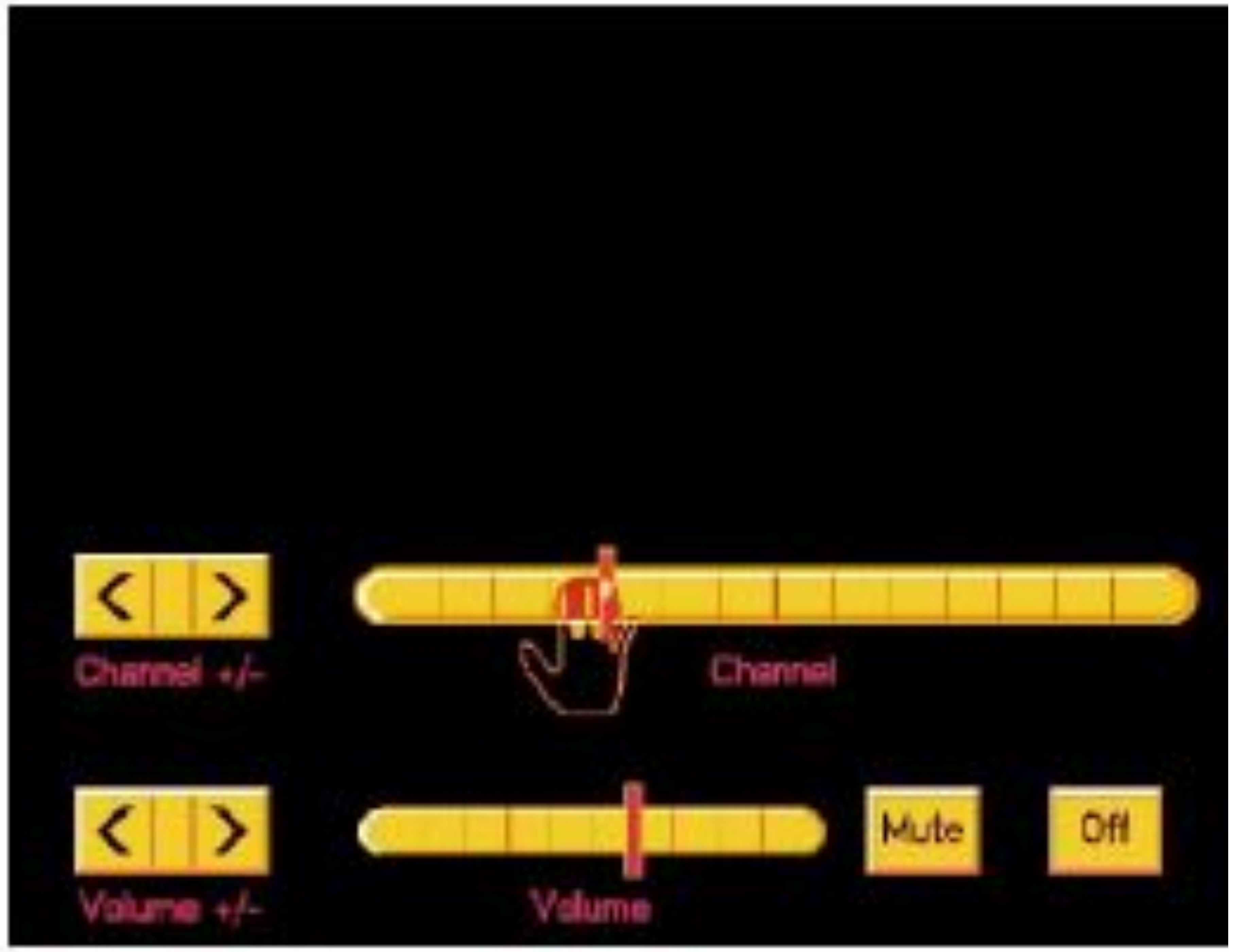
Credit: W. Freeman et al., “Computer Vision for Interactive Computer Graphics,” IEEE Computer Graphics and Applications, 1998

Example 1:



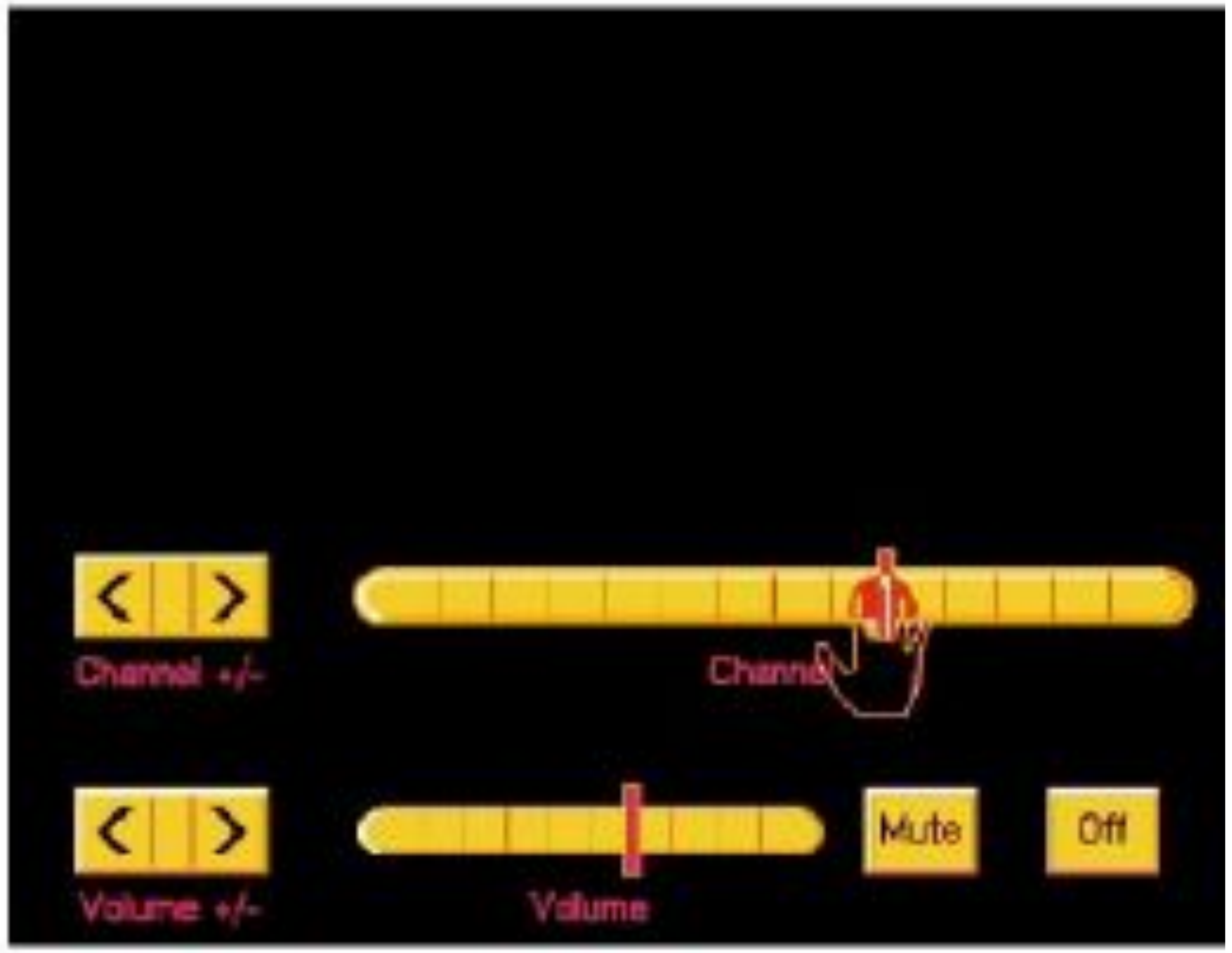
Credit: W. Freeman et al., "Computer Vision for Interactive Computer Graphics," IEEE Computer Graphics and Applications, 1998

Example 1:



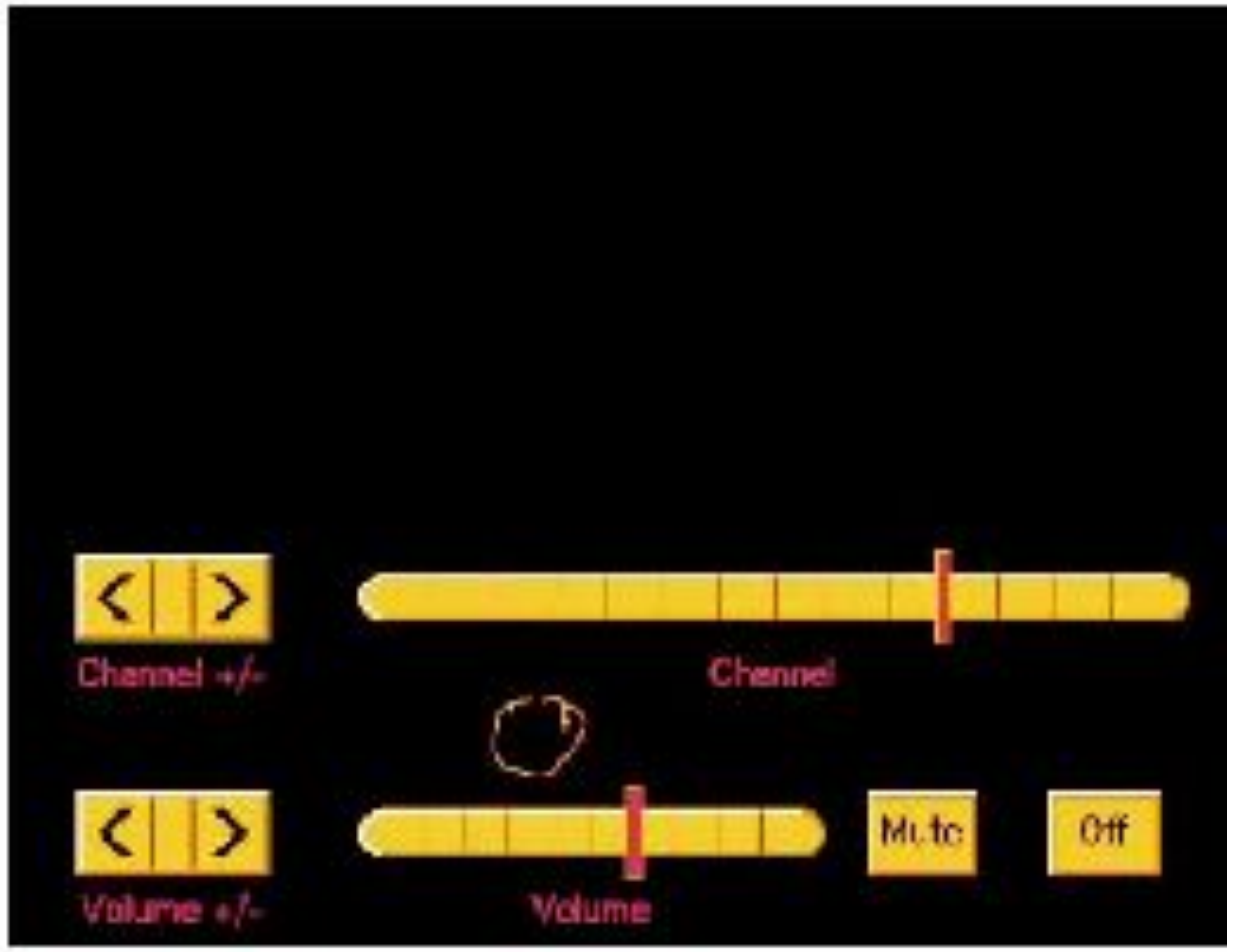
Credit: W. Freeman et al., “Computer Vision for Interactive Computer Graphics,” IEEE Computer Graphics and Applications, 1998

Example 1:



Credit: W. Freeman et al., “Computer Vision for Interactive Computer Graphics,” IEEE Computer Graphics and Applications, 1998

Example 1:



Credit: W. Freeman et al., “Computer Vision for Interactive Computer Graphics,” IEEE Computer Graphics and Applications, 1998

Example 1:

Template (left), image (middle), normalized correlation (right)

Note peak value at the true position of the hand

Credit: W. Freeman et al., "Computer Vision for Interactive Computer Graphics," IEEE Computer Graphics and Applications, 1998

Template Matching

When might **template matching fail** to recognise objects?

Template Matching

When might **template matching fail**?

— Different scales 

Template Matching

When might **template matching fail**?

— Different scales 

— Different orientation 

Template Matching

When might **template matching fail**?

— Different scales  

— Different orientation 

— Lighting conditions 

Template Matching

When might **template matching fail**?

— Different scales  

— Different orientation 

— Lighting conditions 

— Left vs. Right hand  

Template Matching

When might **template matching fail**?

— Different scales

— Different orientation

— Lighting conditions

— Left vs. Right hand

— Partial Occlusions

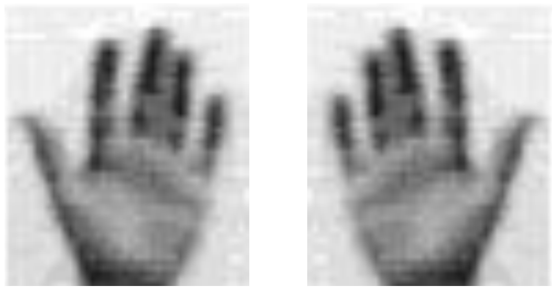
Template Matching

When might **template matching fail**?

— Different scales 

— Different orientation 

— Lighting conditions 

— Left vs. Right hand 

— Partial Occlusions 

— Different Perspective

— Motion / blur

Template Matching Summary

Good News:

— works well in presence of noise

— relatively easy to compute

Bad News:

— sensitive to (spatial) scale change

— sensitive to 2D rotation

More Bad News:

When imaging 3D worlds:

— sensitive to viewing direction and pose

— sensitive to conditions of illumination

Menu for Today

Topics:

— **Digital Imaging** Pipeline
— **Scaled** Representations
— Template **Matching**
— Normalised **Correlation**

Readings:

— **Today's** Lecture: Szeliski 2.3, 3.5, Forsyth & Ponce (2nd ed.) 4.5 - 4.7

Reminders:

— **Assignment 1**: due **Thursday** September 28th

— **Assignment 2**: Scaled Representations, Face Detection and Image Blending available now