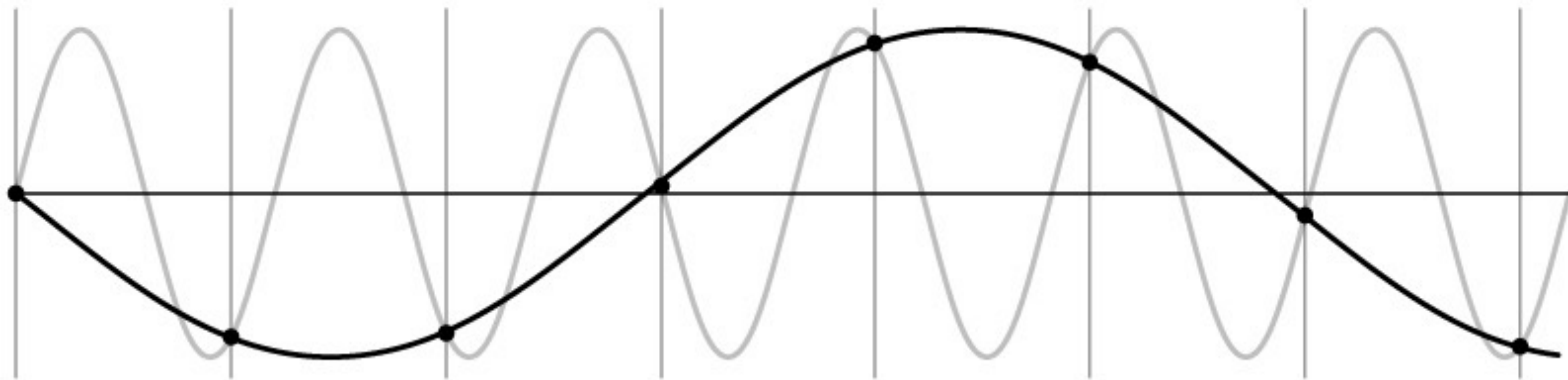


Recap: A Simple Sine Wave

How do we discretize the signal?



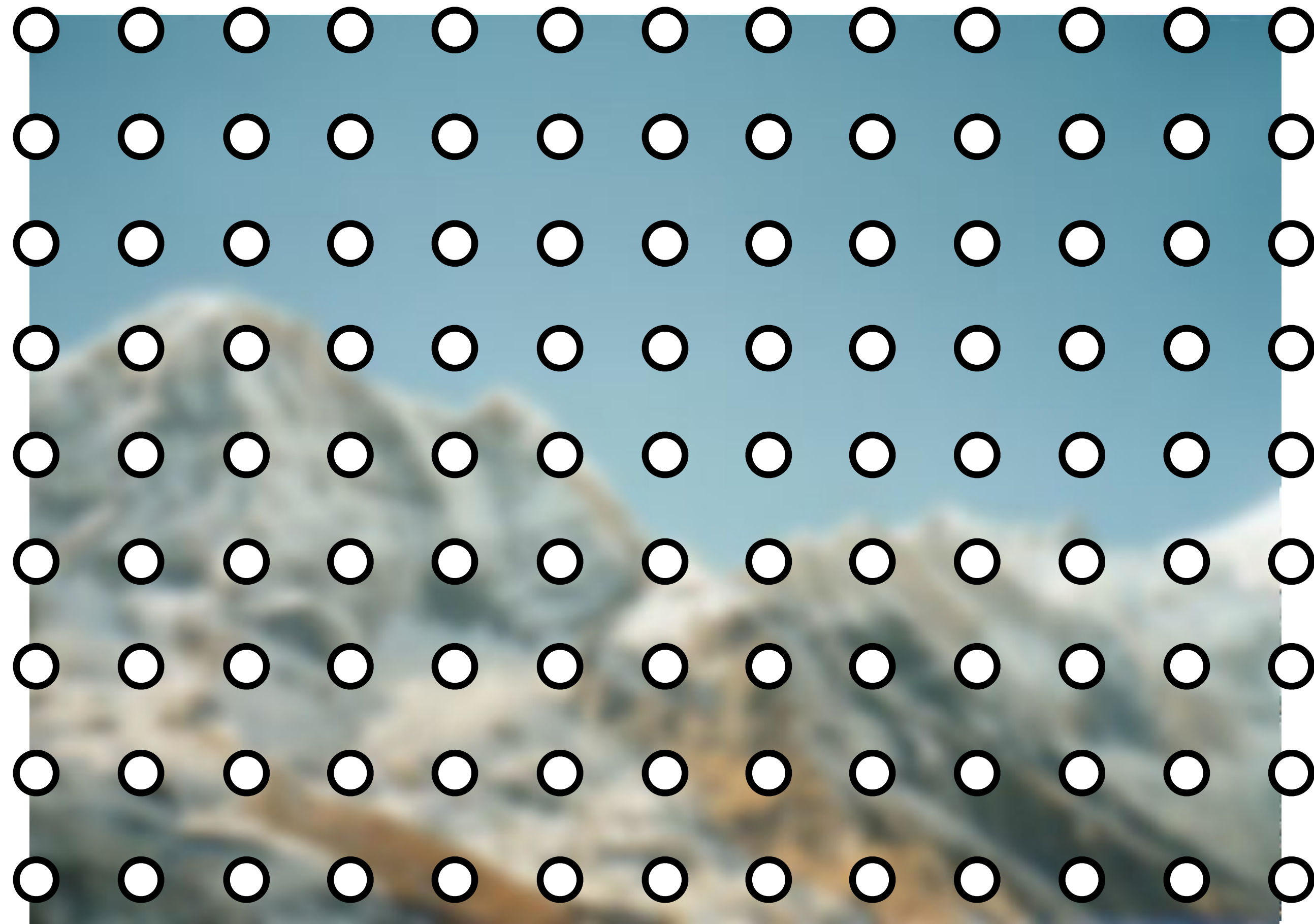
Signal can be confused with one at lower frequency
— This is called “Aliasing”

Temporal Aliasing



Resampling Images

With correct sigma value for a Gaussian, no information is lost



Improved Method: First blur the image (with low-pass) then take n-th pixel

Aliasing Example

Sampling every 5th pixel with and without low-pass blur



No filtering



Gaussian Blur $\sigma = 3.0$

$$\sigma = 1/(2s)$$



CPSC 425: Computer Vision

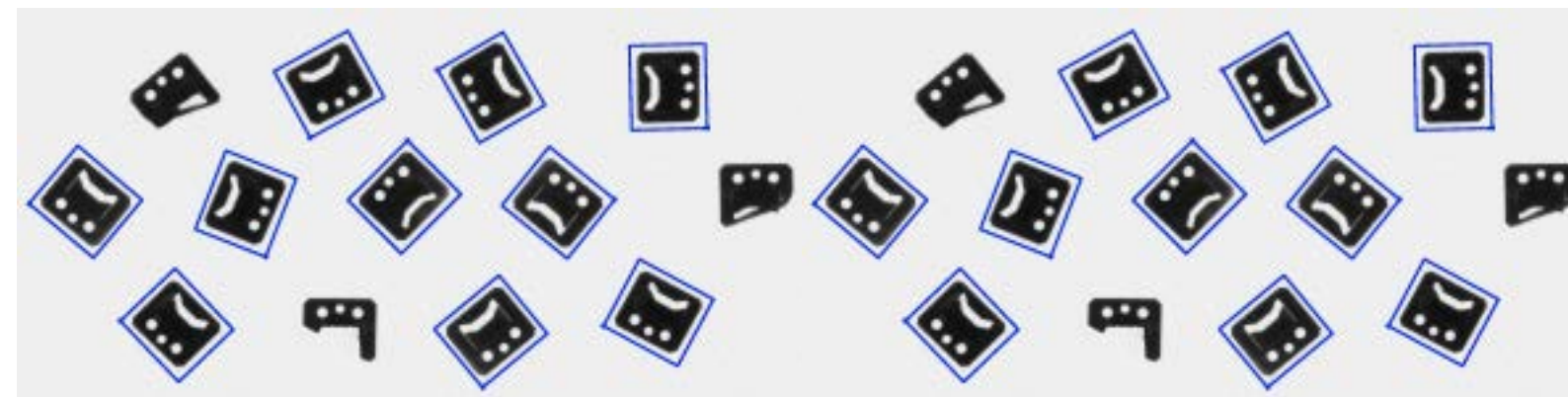


Image Credit: https://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html

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 **Jan 30th**
- **Assignment 2**: Scaled Representations, Face Detection and Image Blending available now

Goal

1. See how image filtering can be used in practice
2. Understand the concept behind template matching

Assignment 2 available now

UBC CSPC 425 Assignment 2 | Kwang Moo Yi @ UBC

https://www.cs.ubc.ca/~kmyi/teaching/cpsc425/assignment2.html

Kwang Moo Yi @ University of British Columbia

Home Team Publications Teaching Openings GitHub

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:

```
import math
import numpy as np

from scipy import signal
from PIL import Image, ImageDraw
```

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

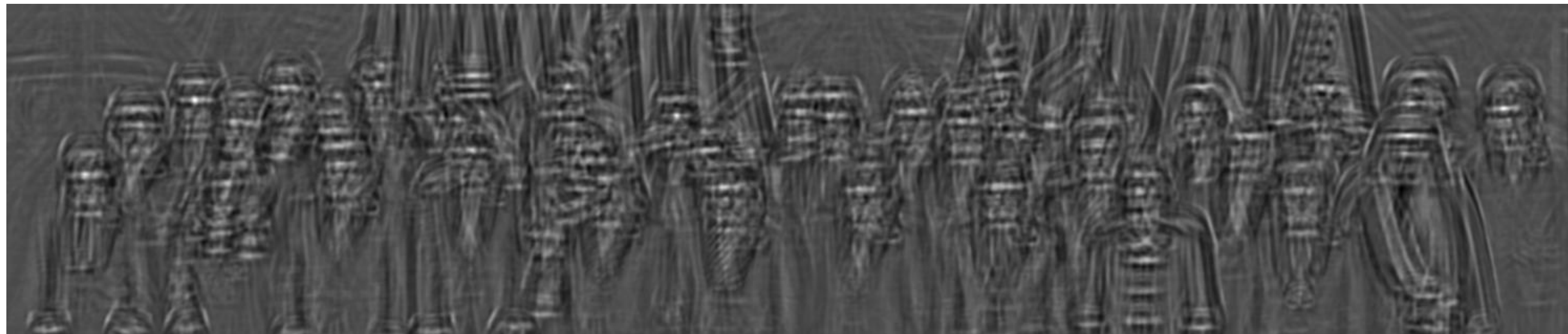
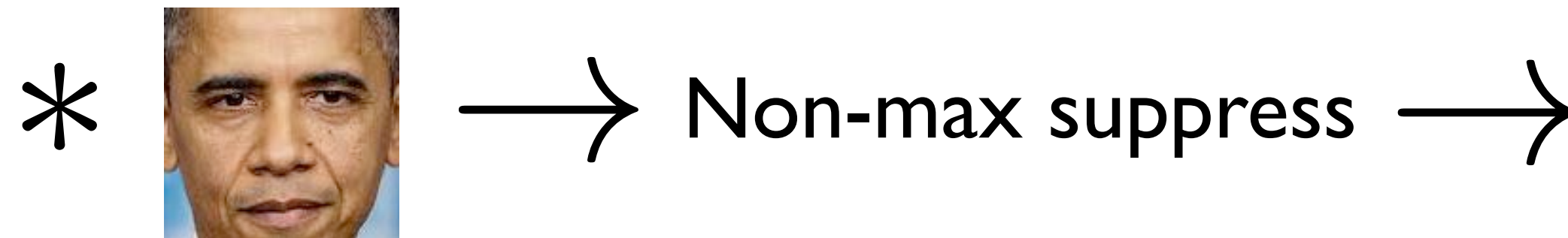
```
unzip hw2part1.zip
```

After you unzip, import the provided `ncc.py` python script with `import ncc`.

The face template we will use is [face_detection_template.jpg](#) and is shown below.

Template Matching

Convolve image with template, find local maxima



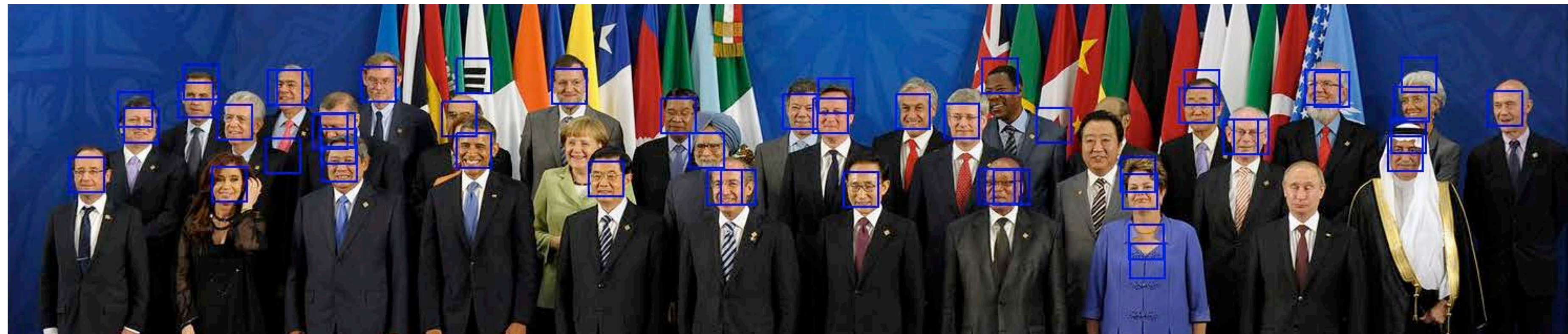
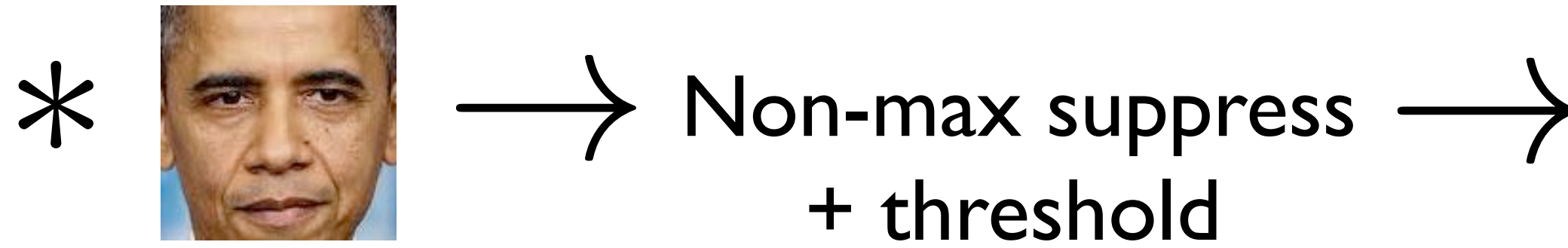
Template Matching

Convolve image with template, find local maxima



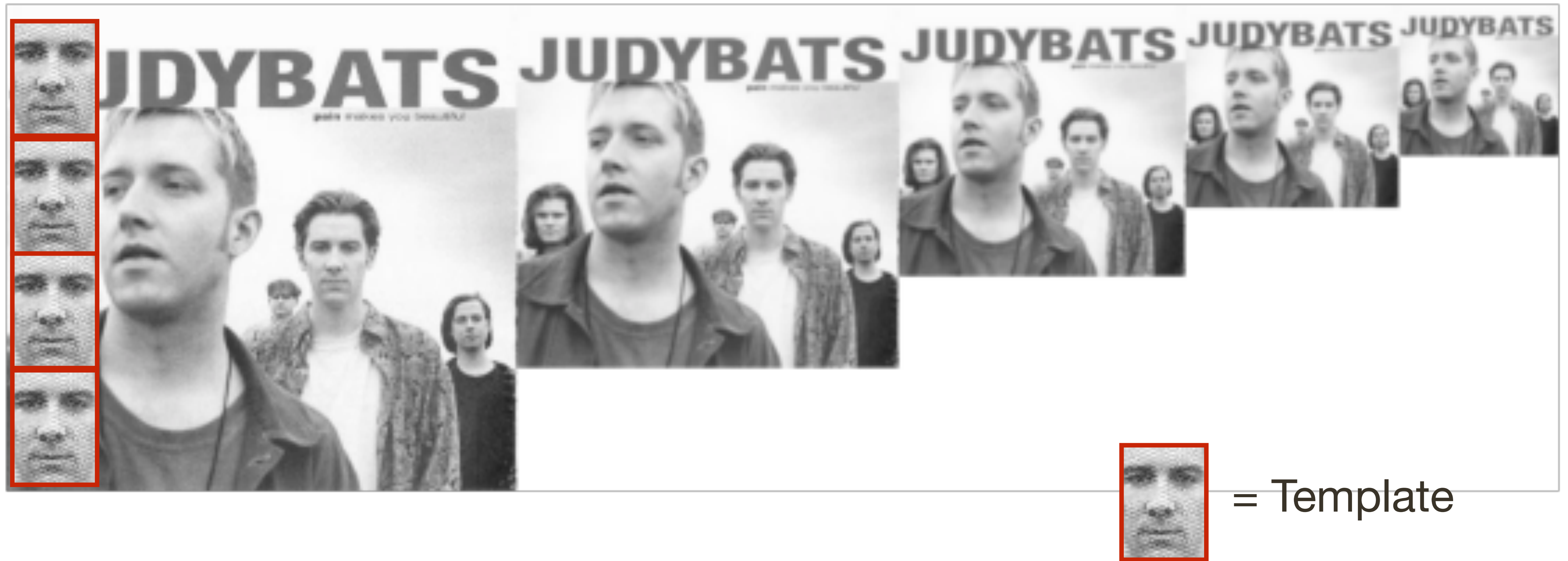
Template Matching

Convolve image with template, find local maxima



Multi-Scale Template Matching

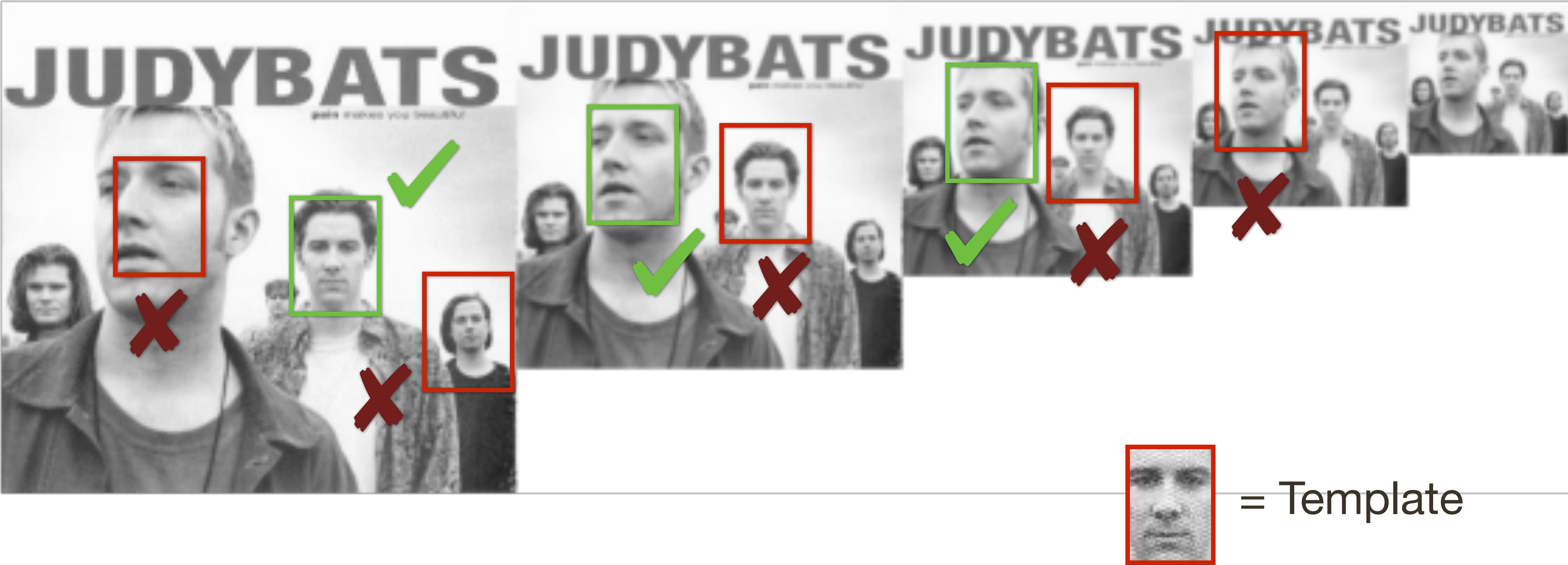
Correlation with a **fixed-sized template** only detects faces at **specific scales**



[Assignment 2]

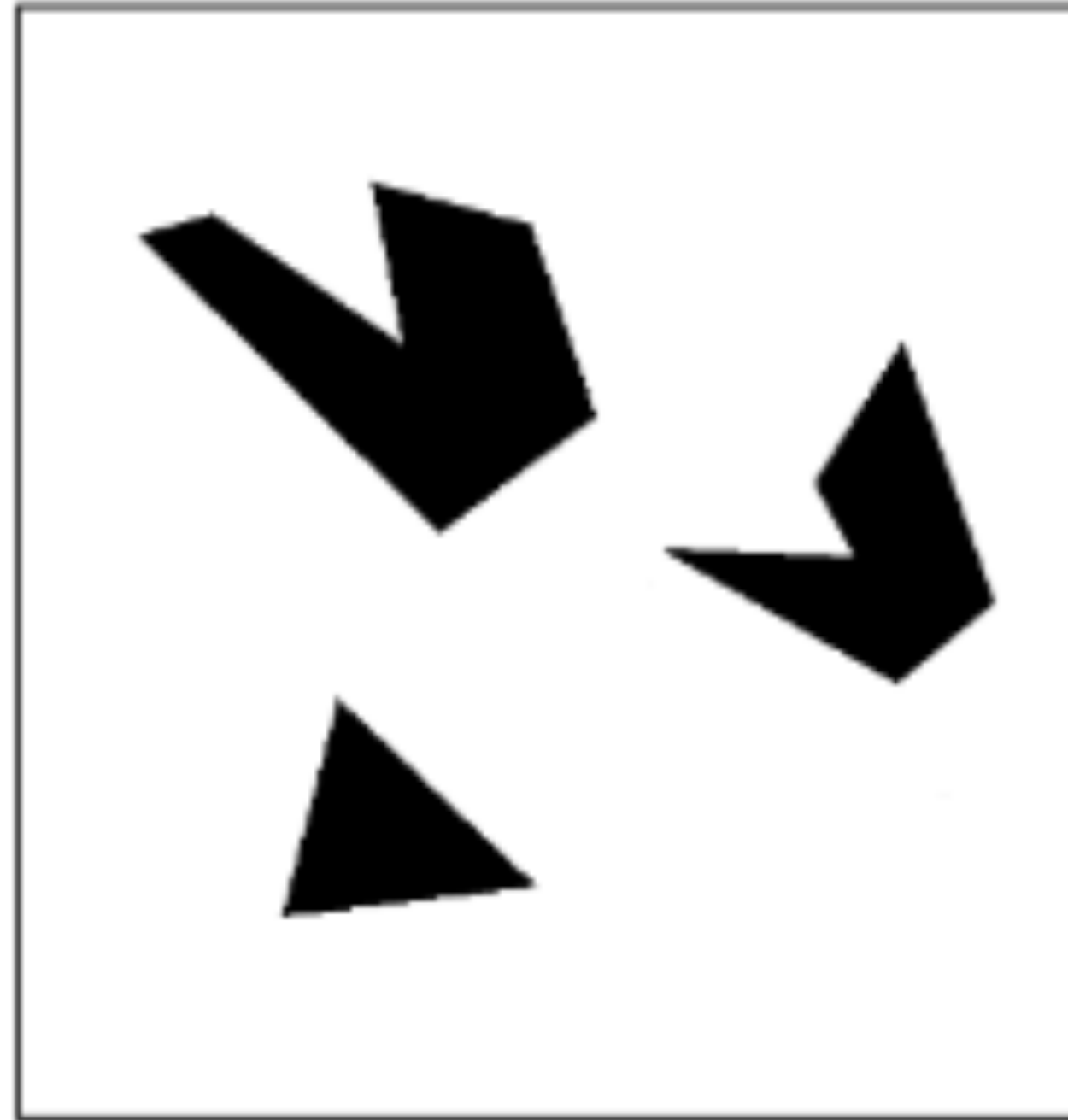
Multi-Scale Template Matching

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



[Assignment 2]

Template Matching



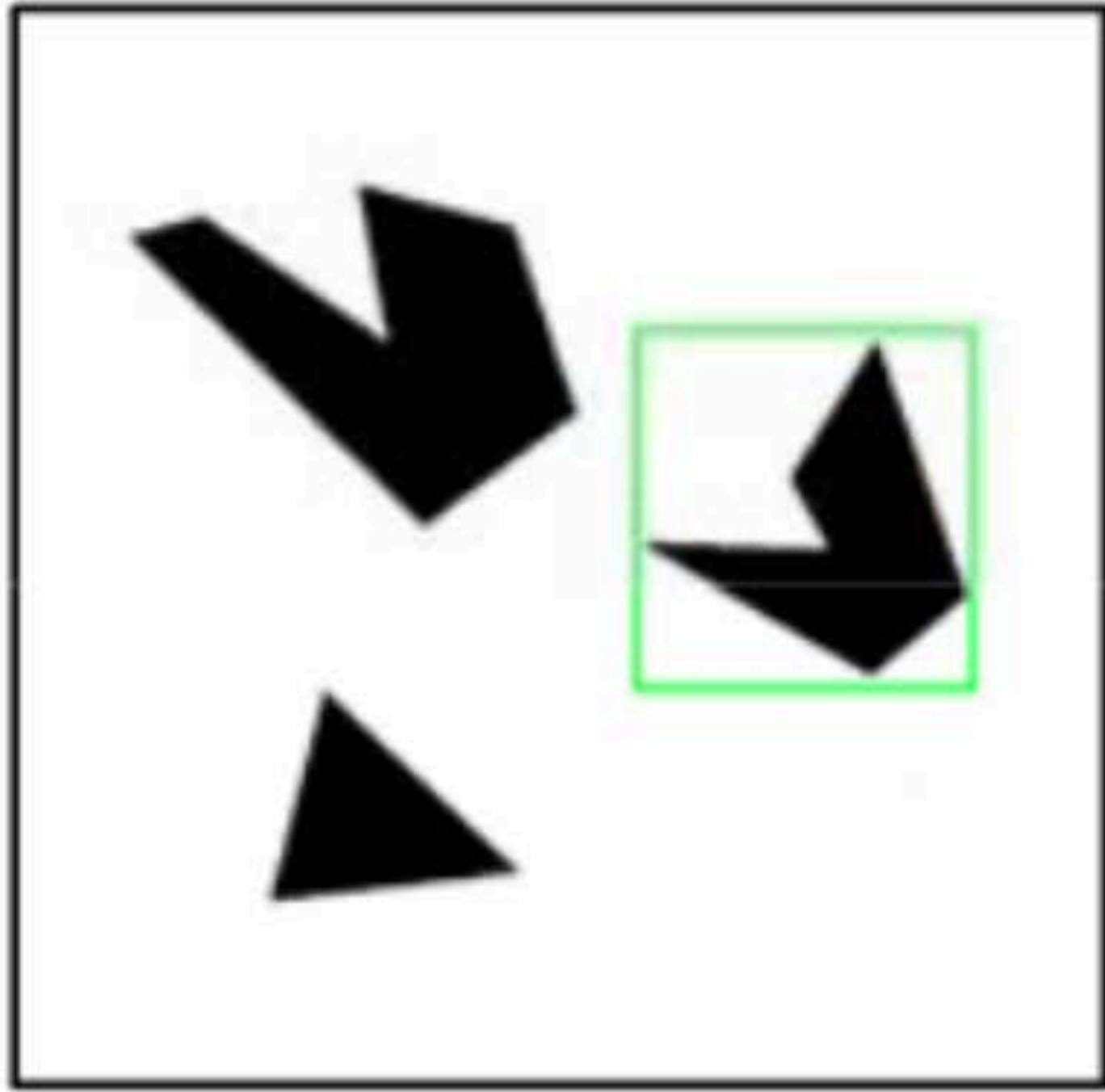
Scene



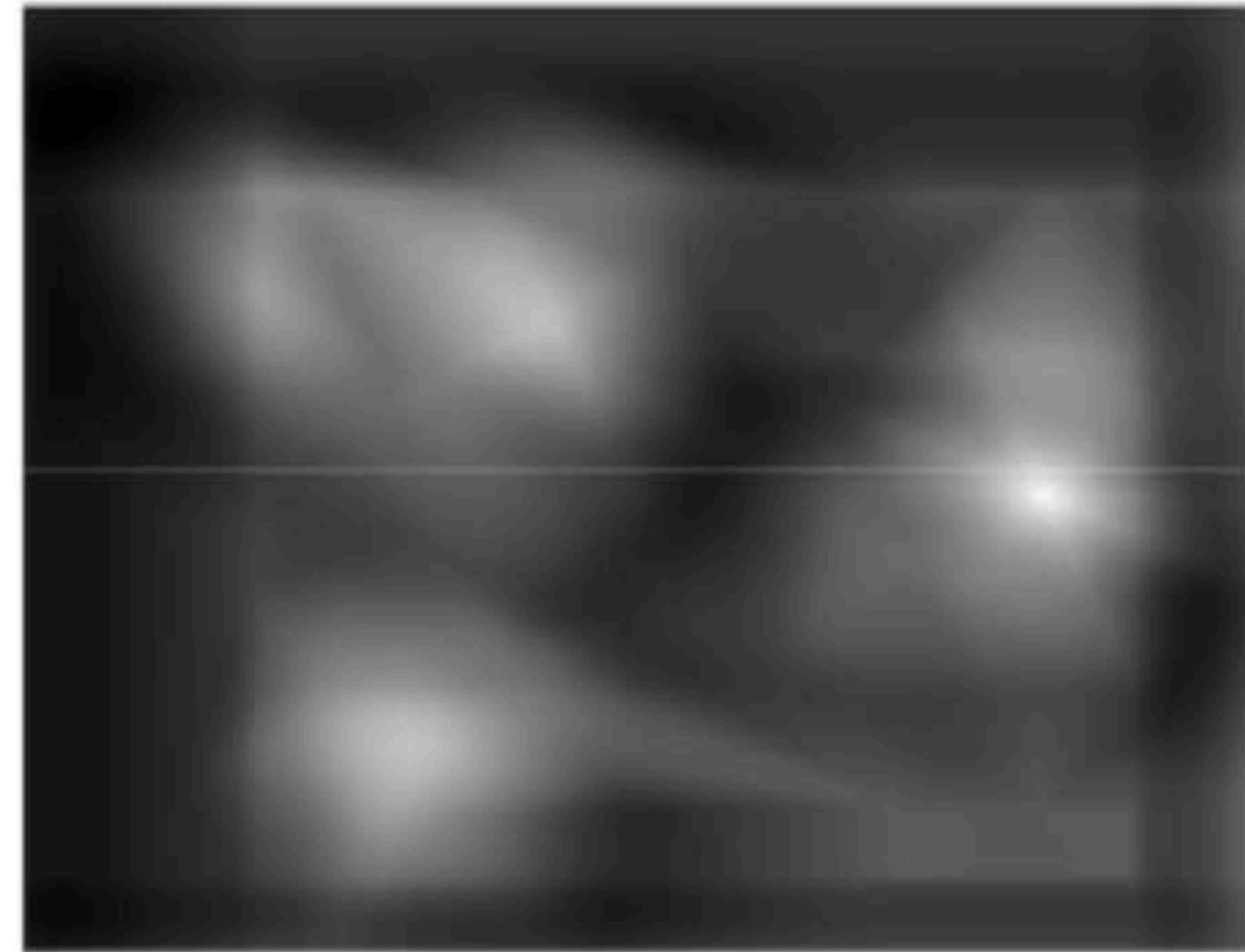
Template (mask)

A toy example

Template Matching



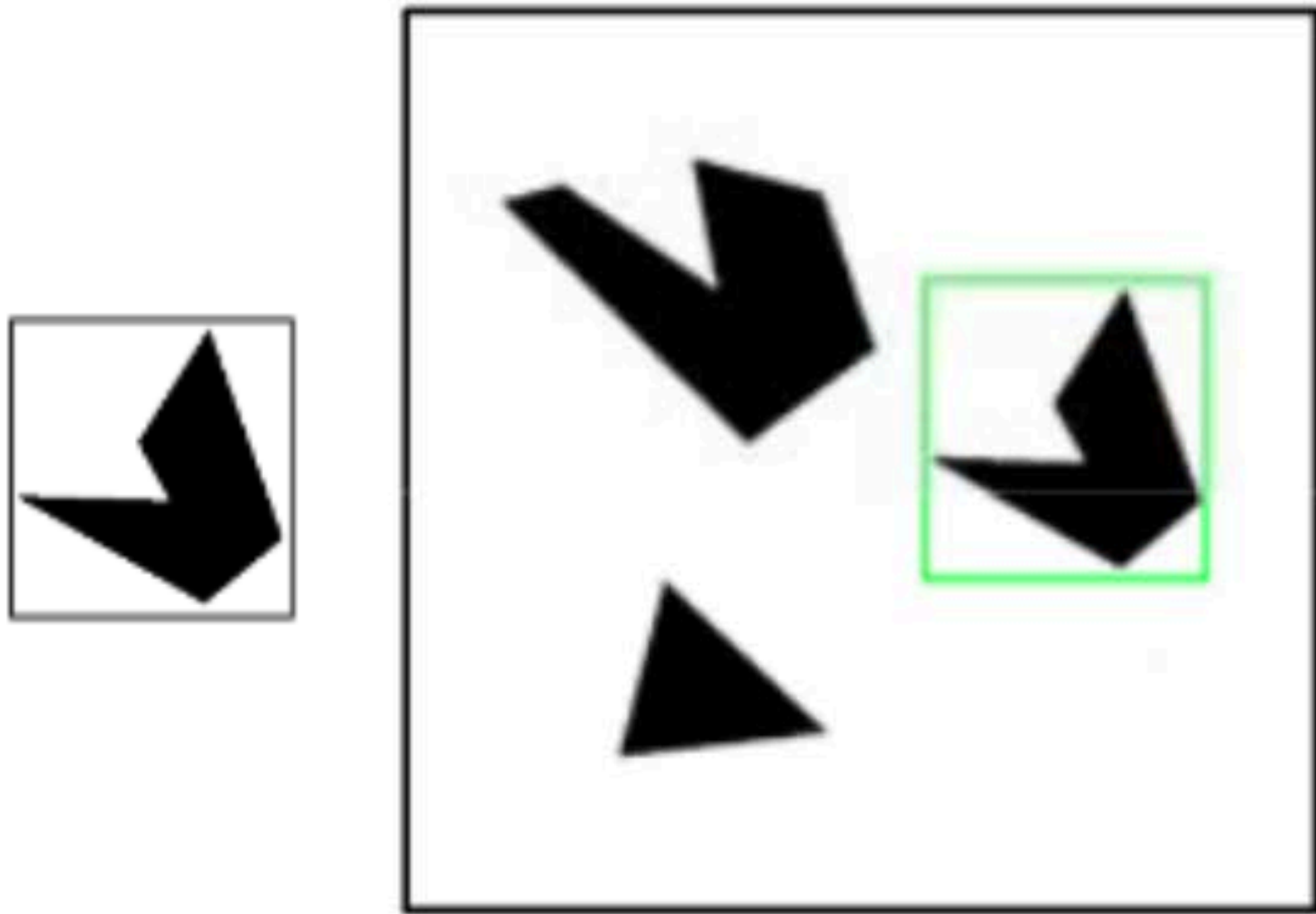
Detected template



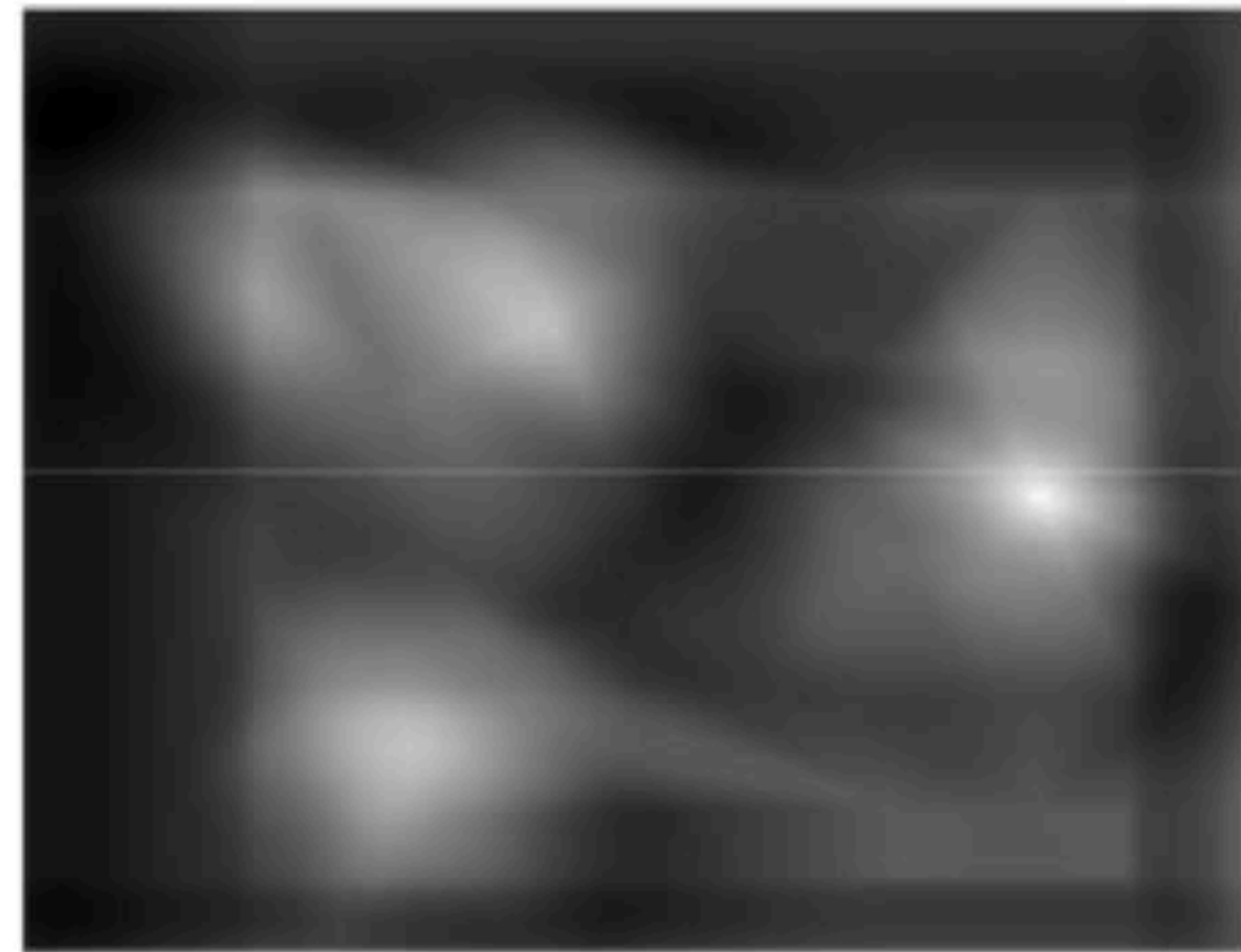
Correlation map

Template Matching

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



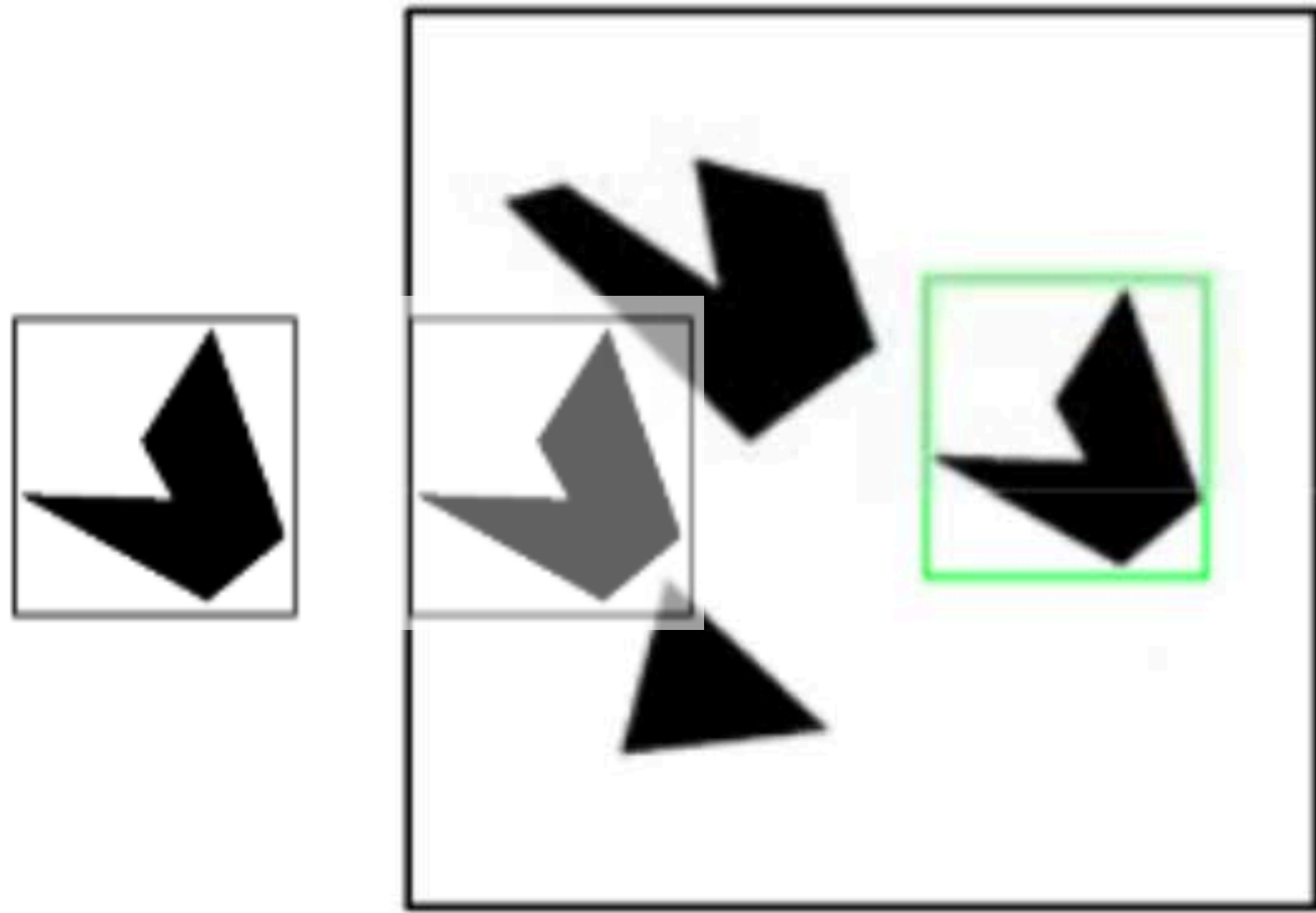
Detected template



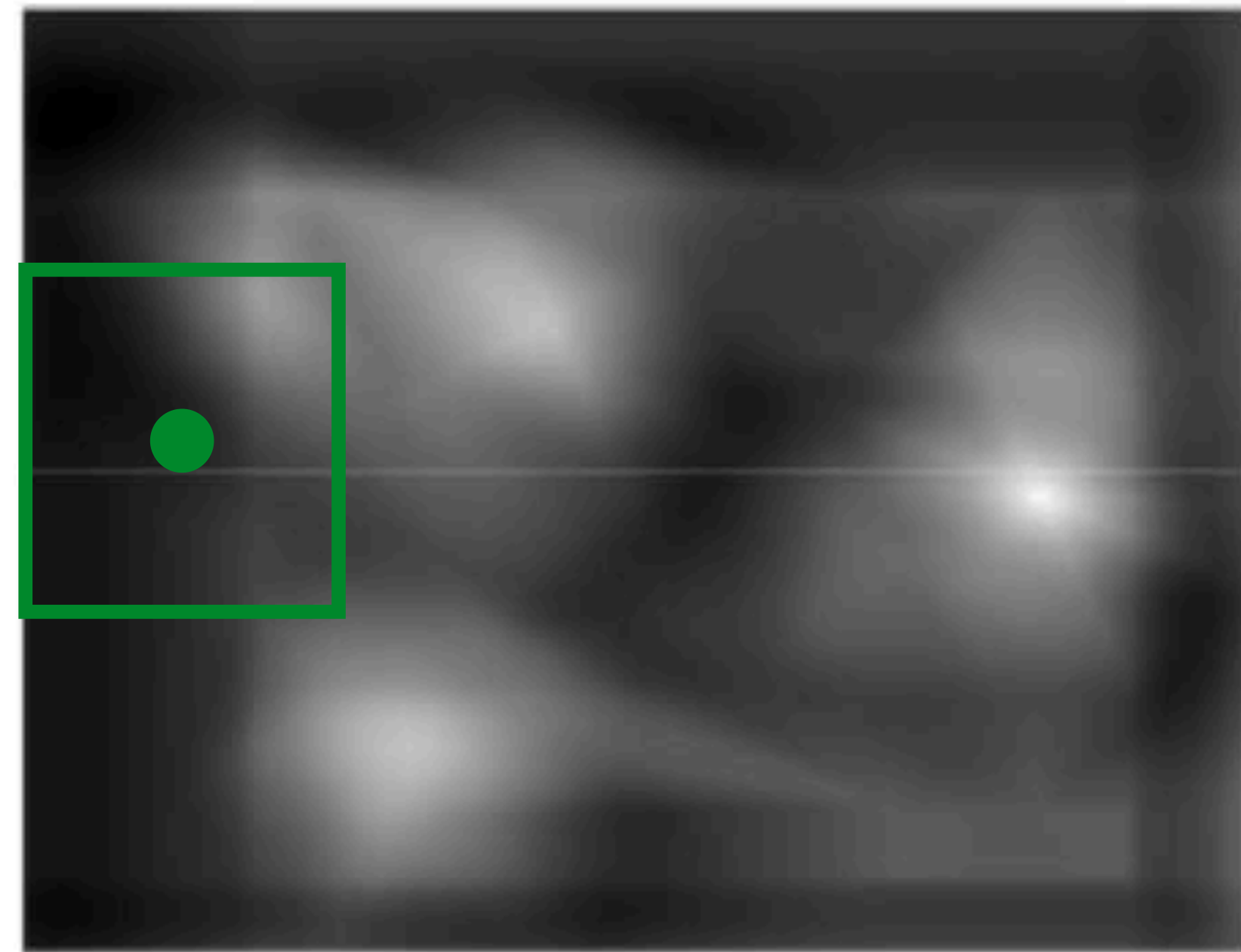
Correlation map

Template Matching

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



Detected template



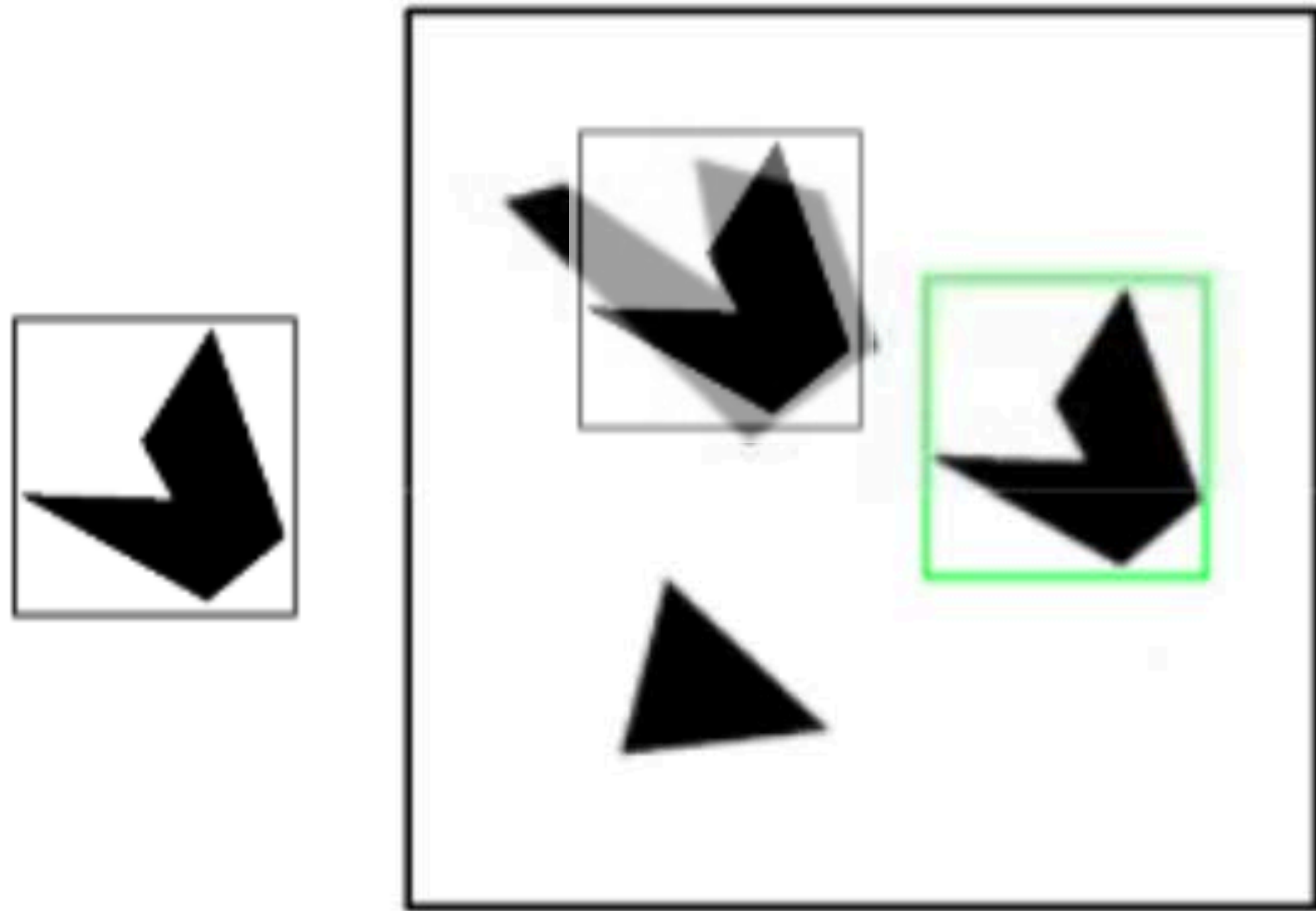
Correlation map

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

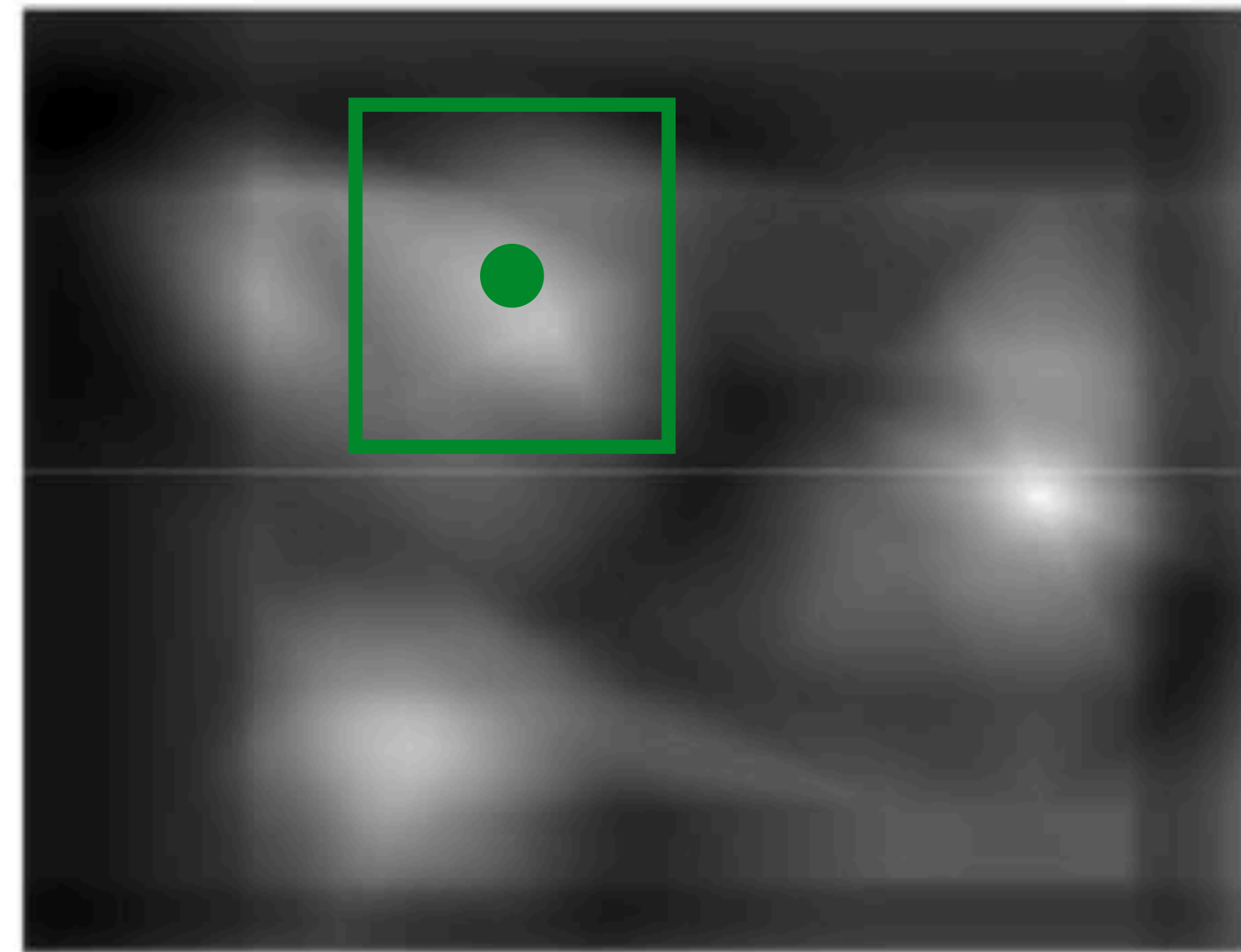
Slide Credit: Kristen Grauman

Template Matching

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



Detected template



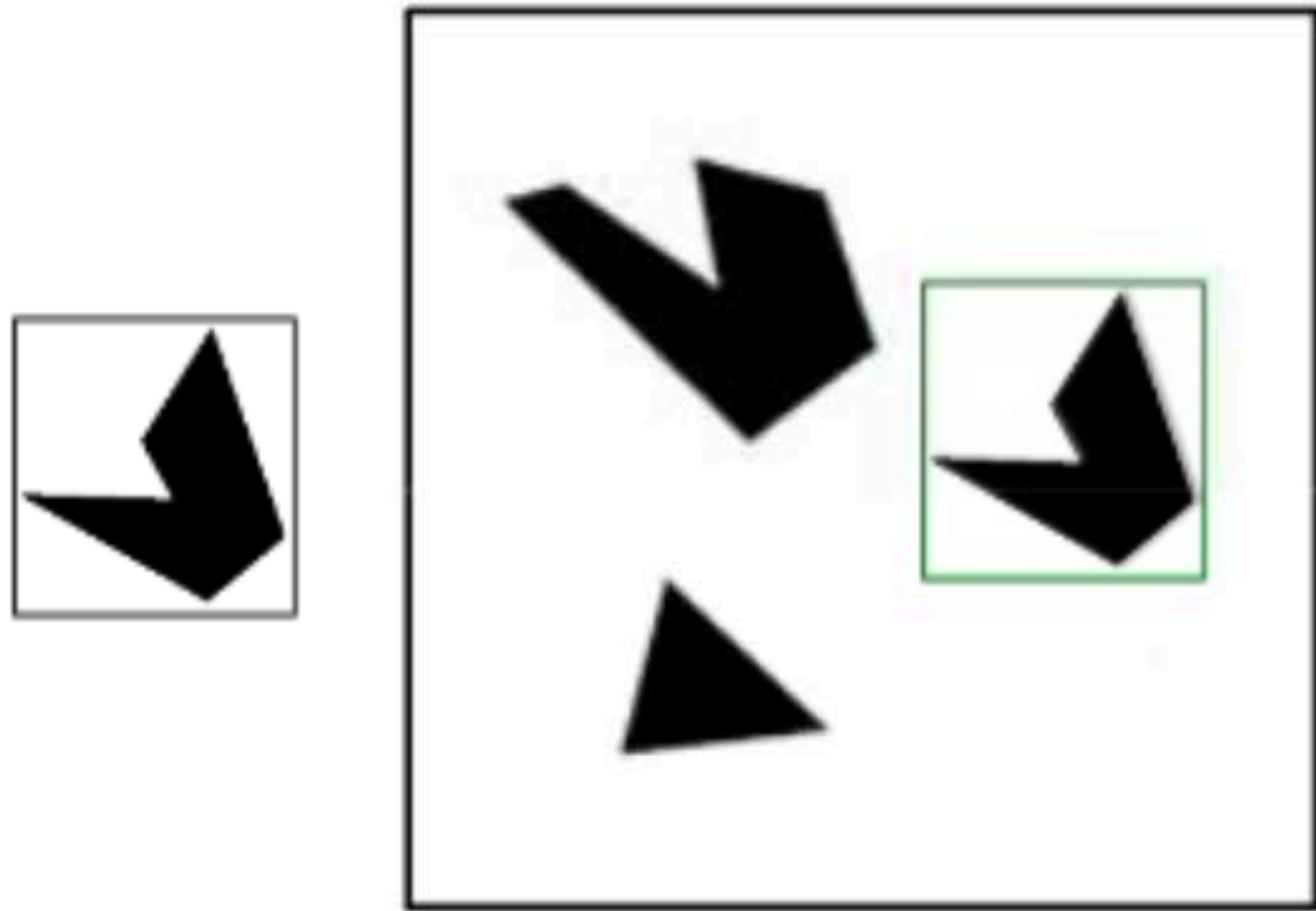
Correlation map

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

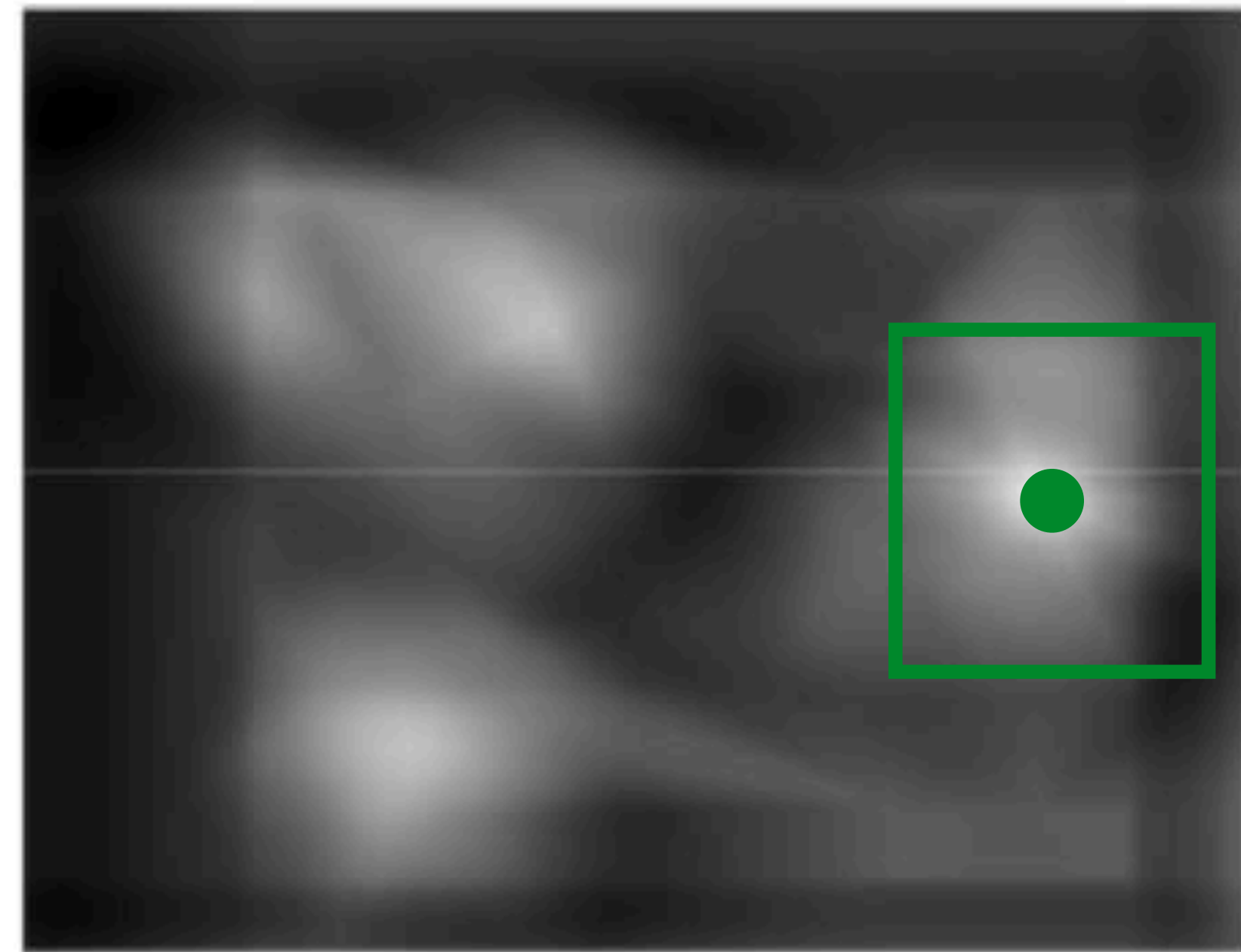
Slide Credit: Kristen Grauman

Template Matching

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



Detected template



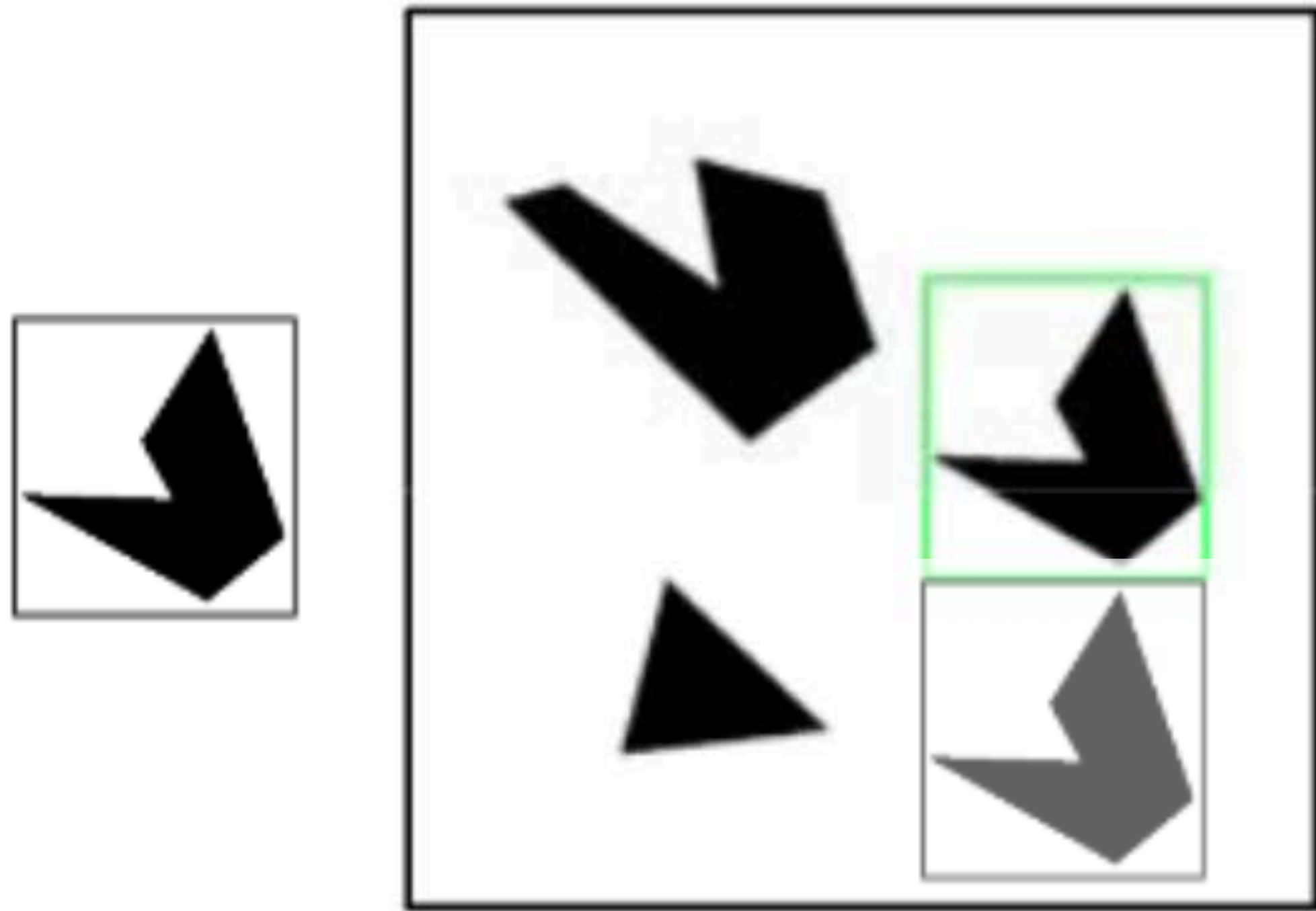
Correlation map

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

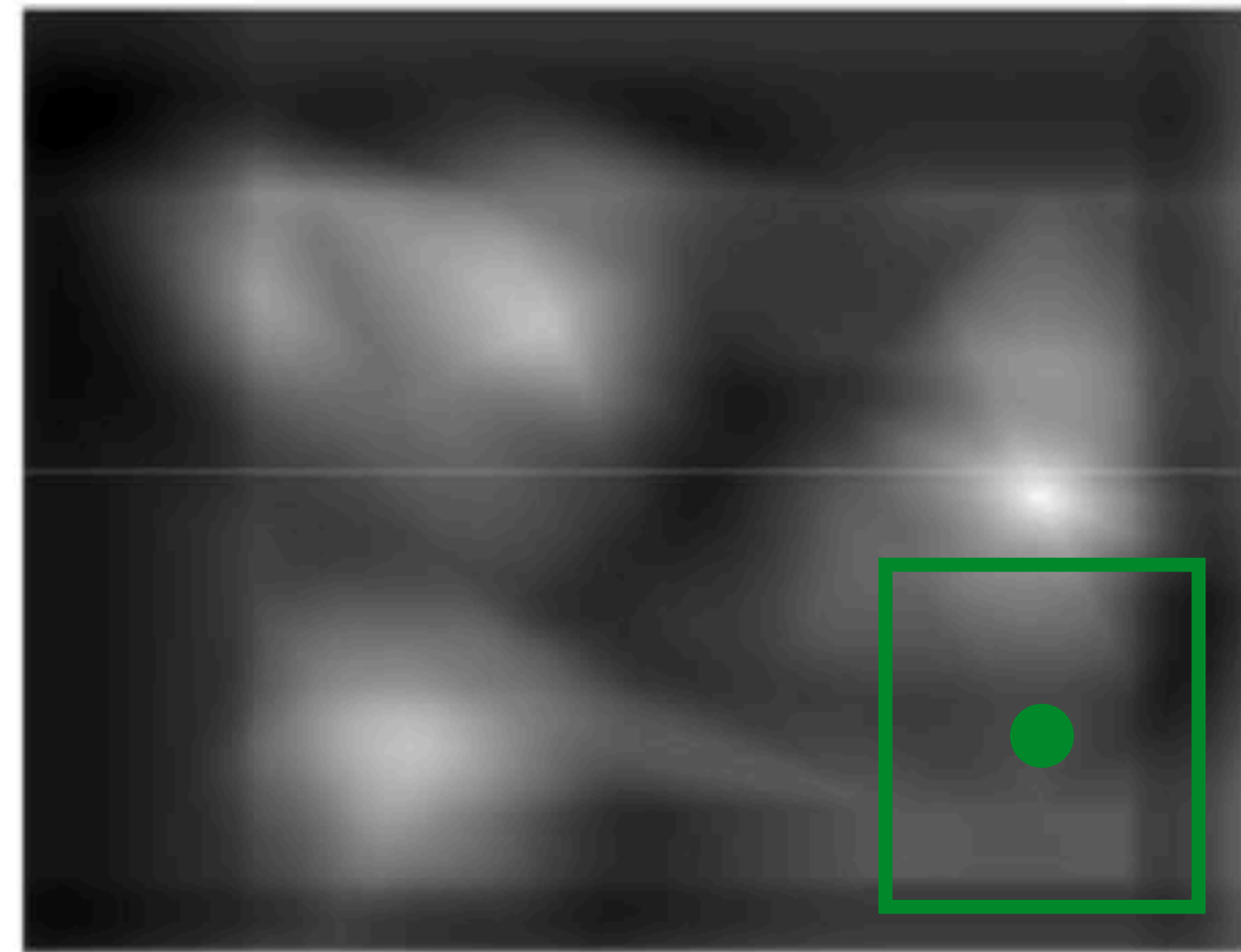
Slide Credit: Kristen Grauman

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

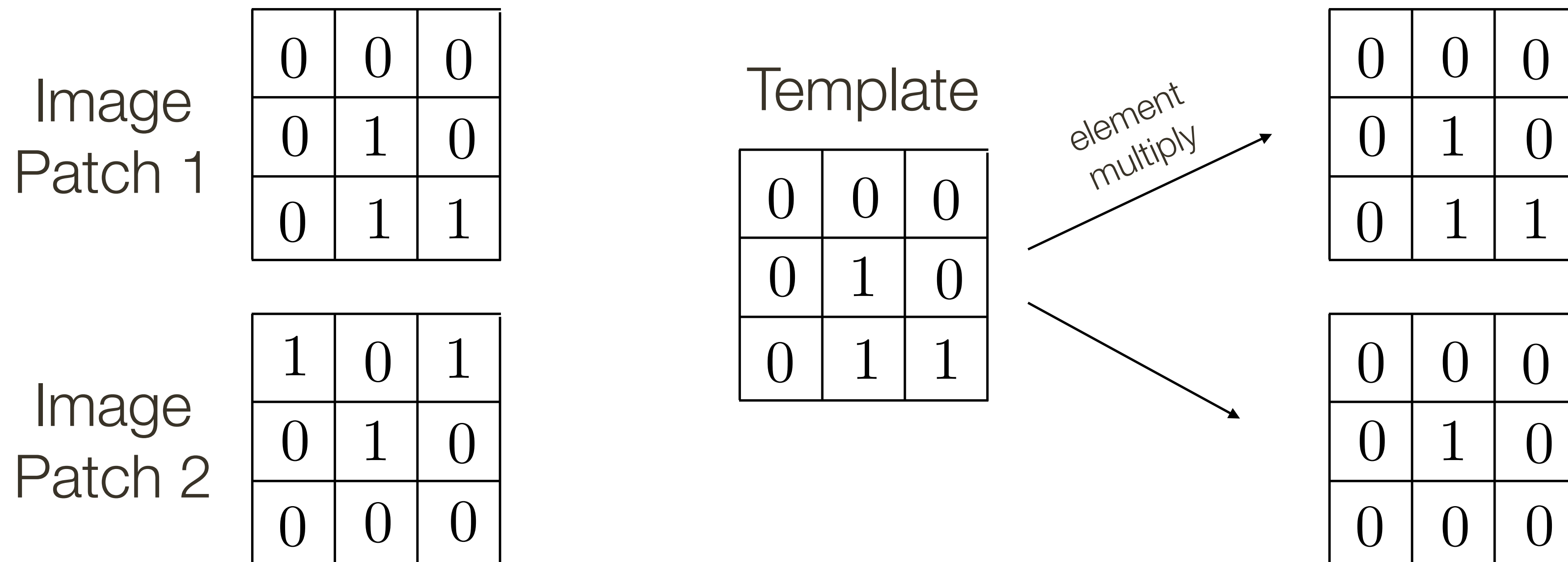
Template

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.

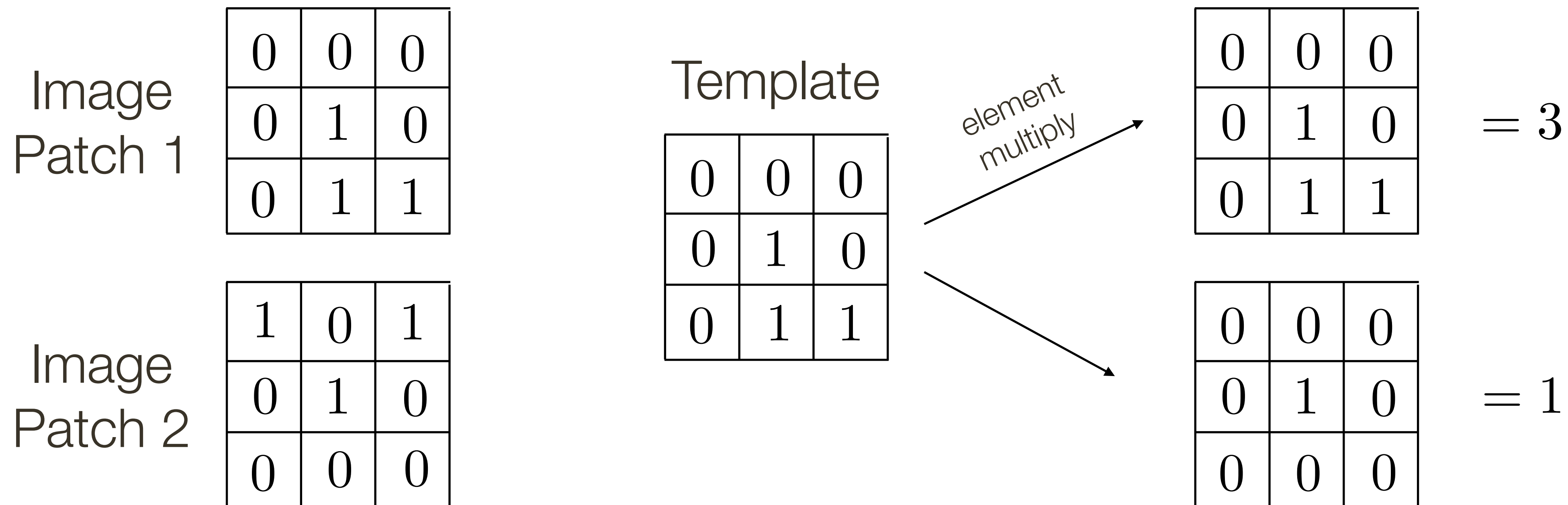
- Consider the filter and image patch as vectors.
- Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.



Template Matching

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

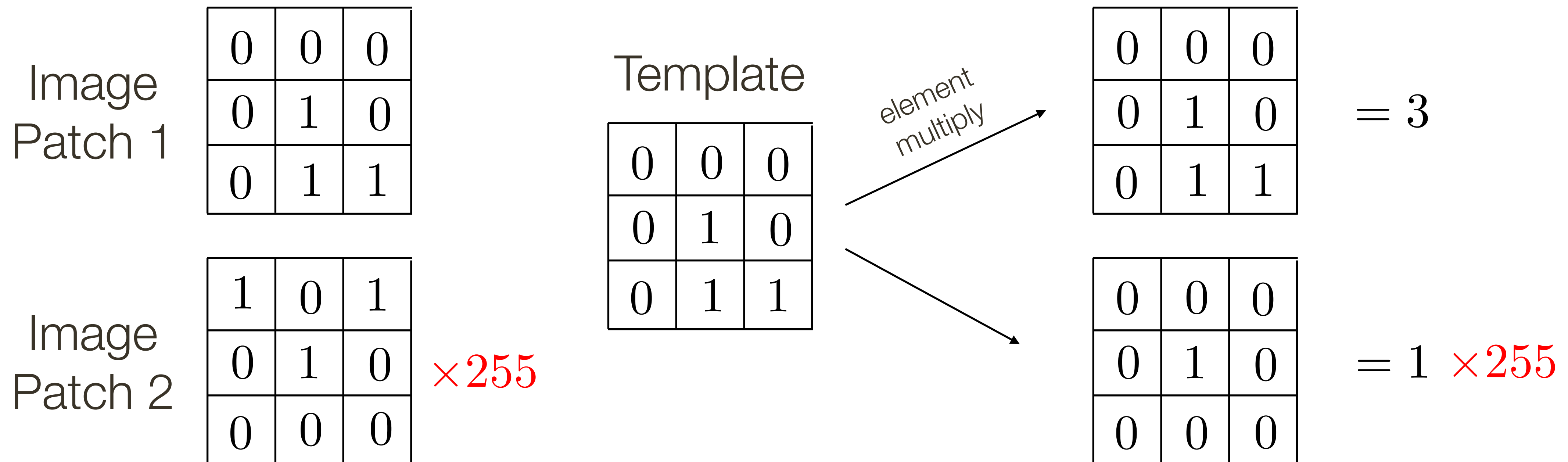
- Consider the filter and image patch as vectors.
- Applying a filter at an image location can be interpreted as computing the dot product between the filter and the local image patch.



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	0	0	0
	0	1	0

Template

--	--	--

element multiply

0	0	0
0	1	0

= 3

The dot product may be large simply because the image region is bright.

We need to normalize the result in some way.

Image Patch 2	0	1	0
	0	0	0

$\times 255$

0	1	0
0	0	0

= 1×255

Template Matching

 **7.3** Correlation, Normalised Correlation, SSD

Template Matching

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

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

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

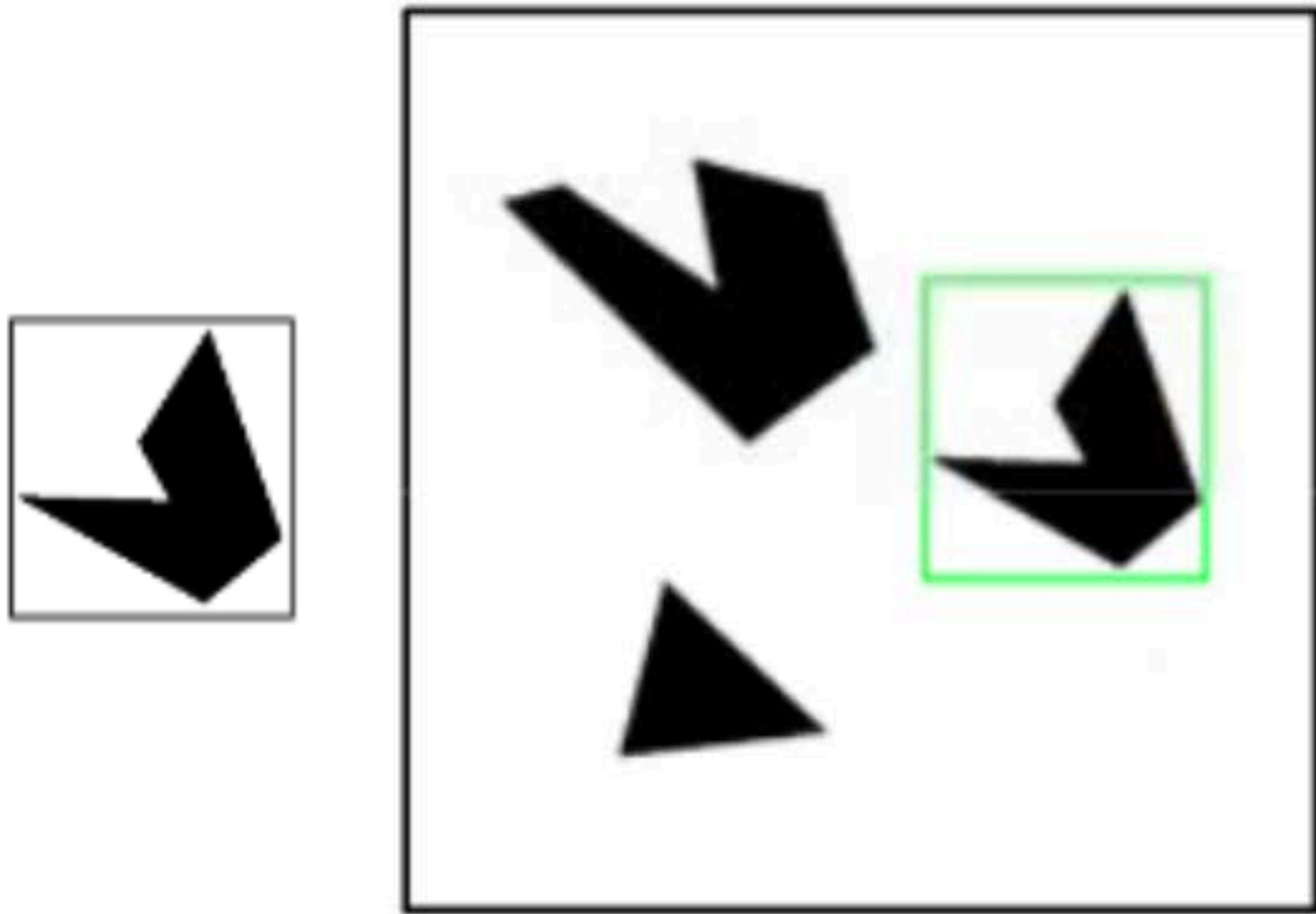
Sum Squared Difference, $\text{SSD} = |\mathbf{I} - \mathbf{J}|^2$

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

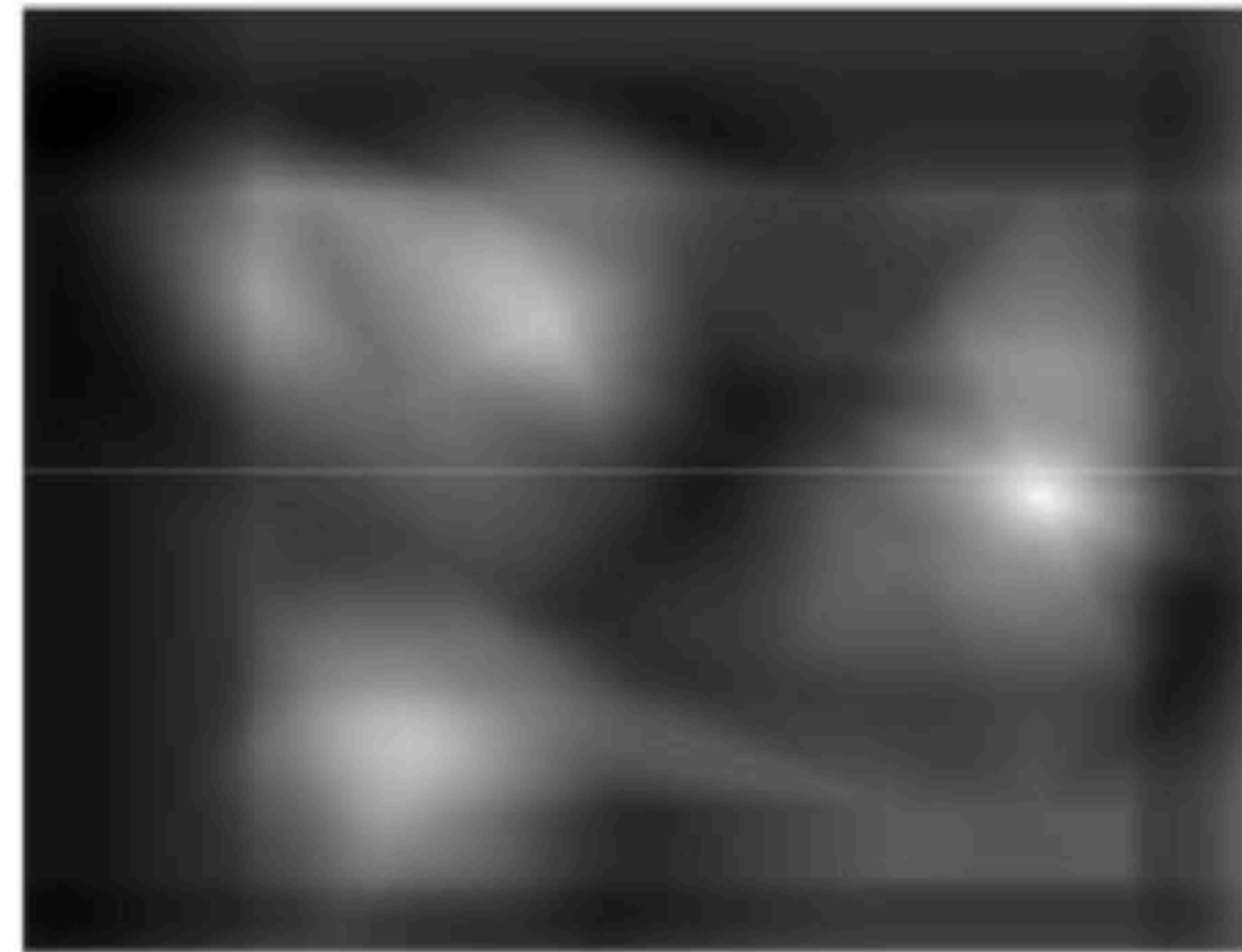
Minimising SSD and maximizing Normalized Correlation are equivalent if $|\mathbf{I}| = |\mathbf{J}| = 1$

Template Matching

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



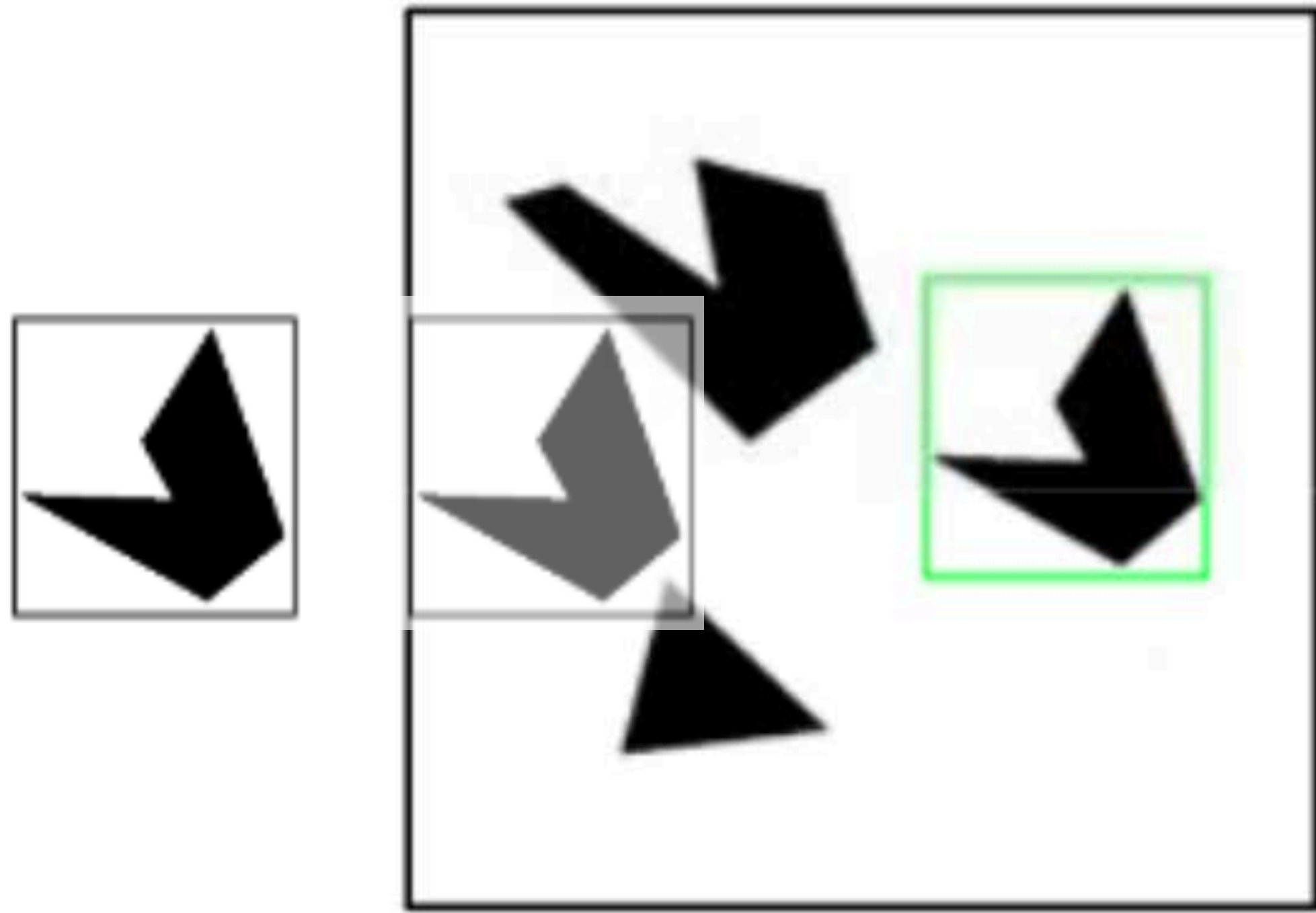
Detected template



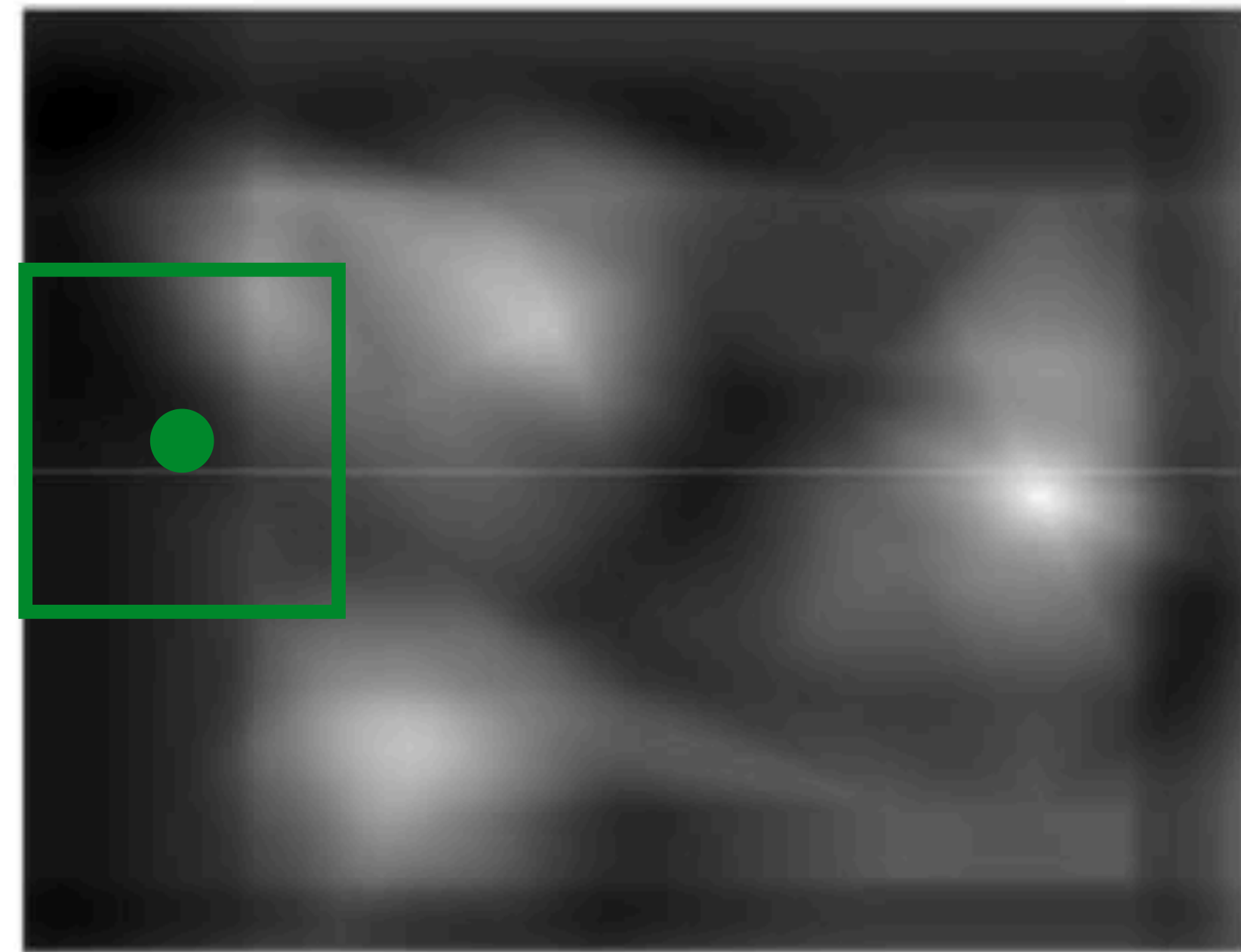
Correlation map

Template Matching

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



Detected template



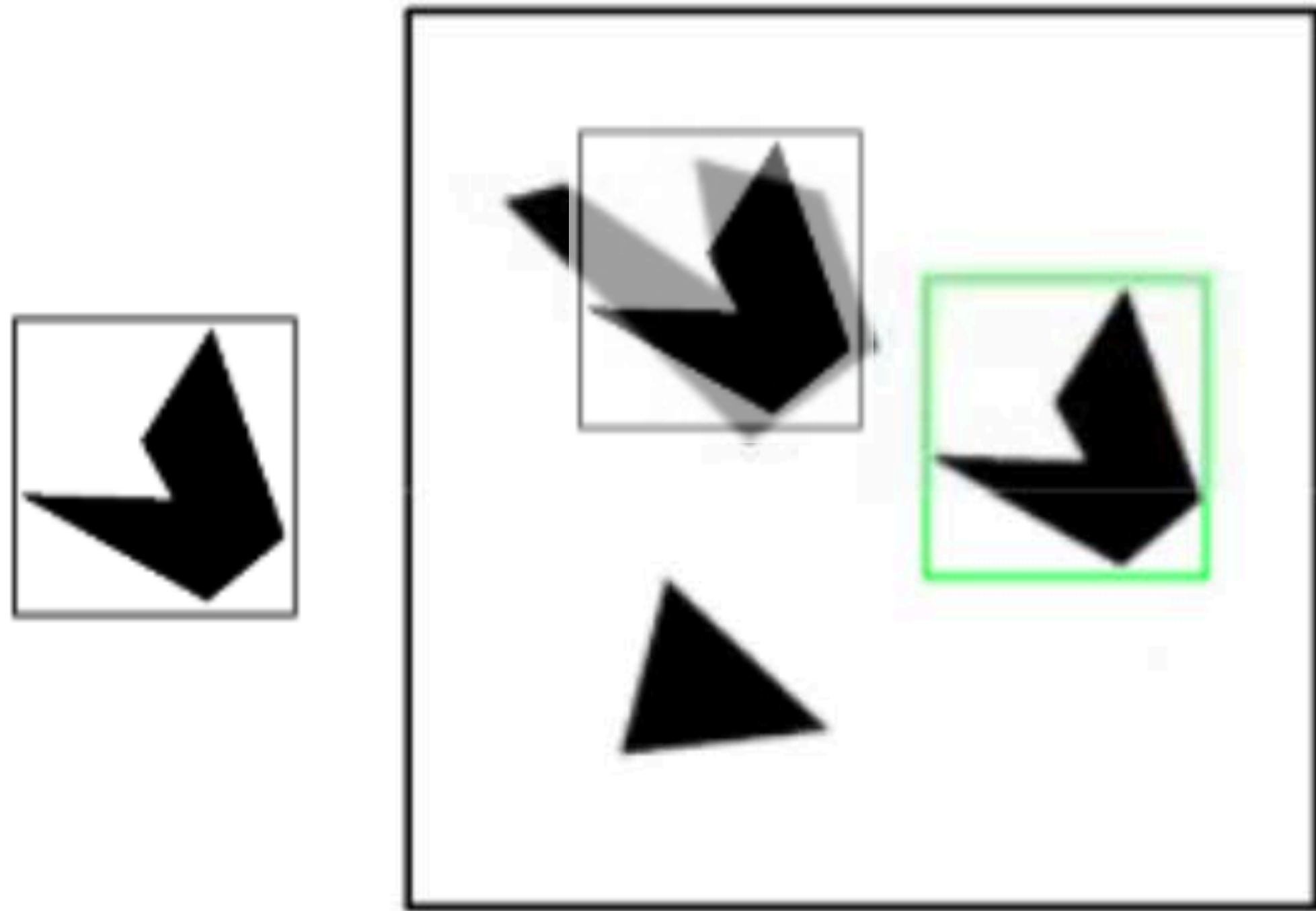
Correlation map

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

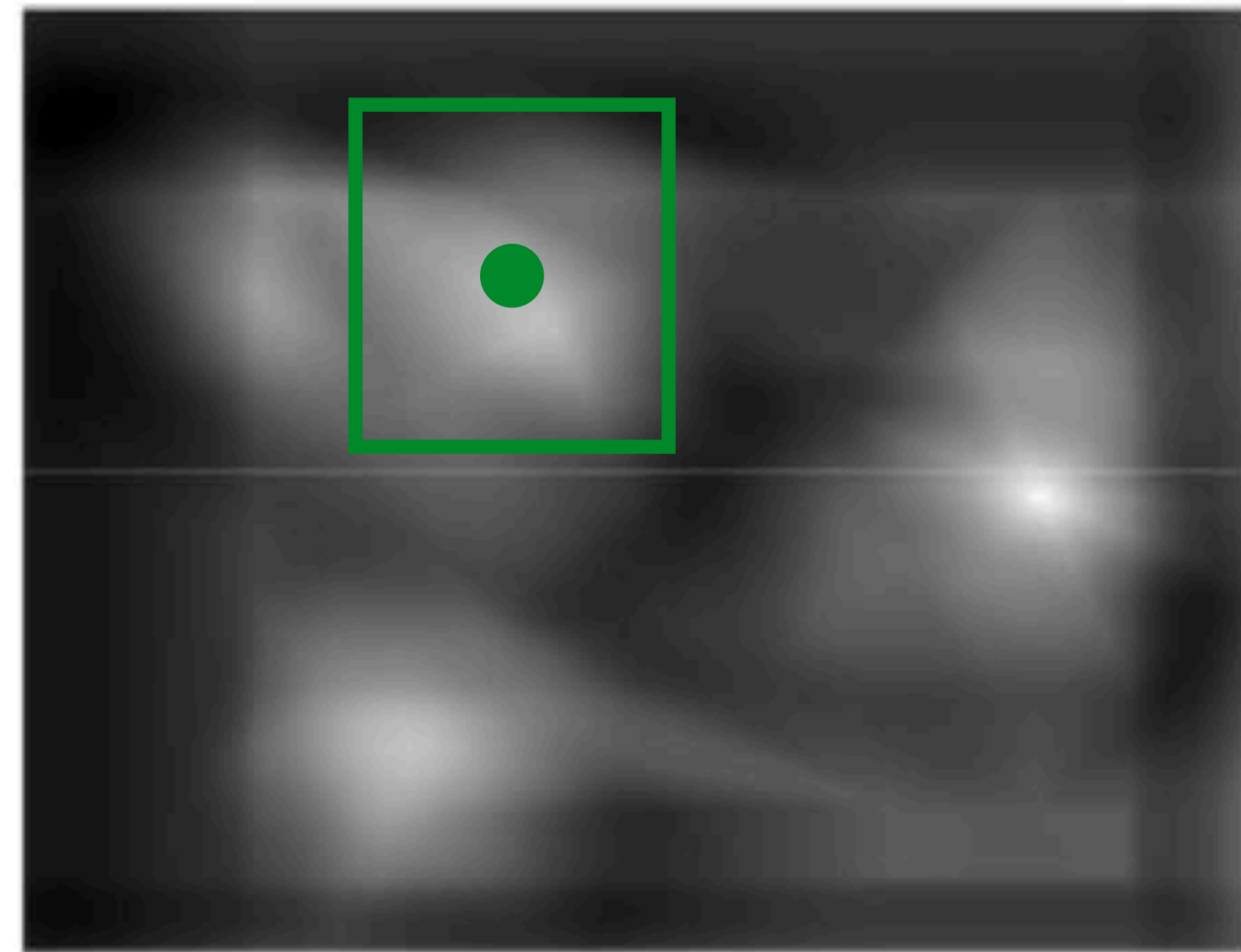
Slide Credit: Kristen Grauman

Template Matching

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



Detected template



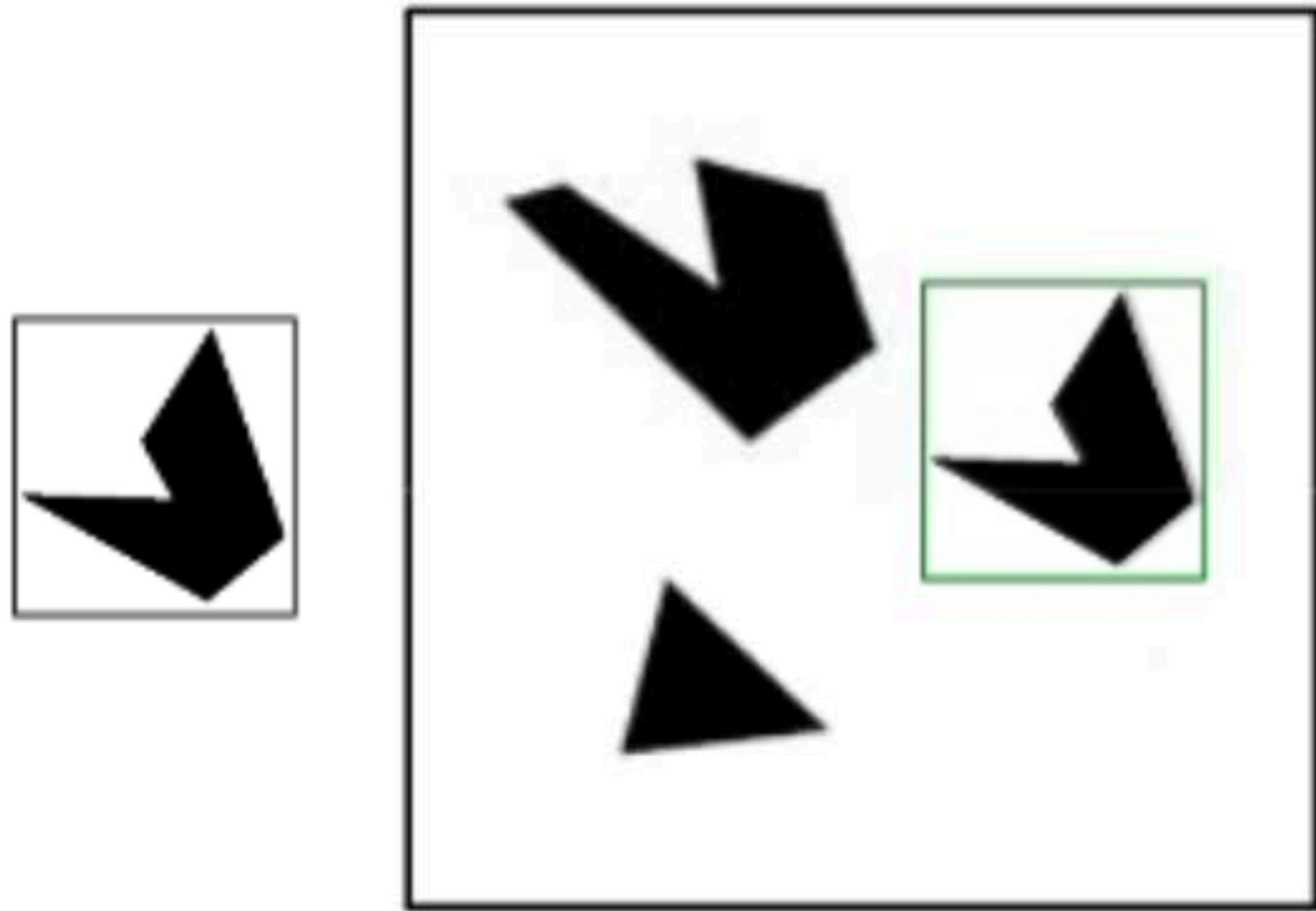
Correlation map

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

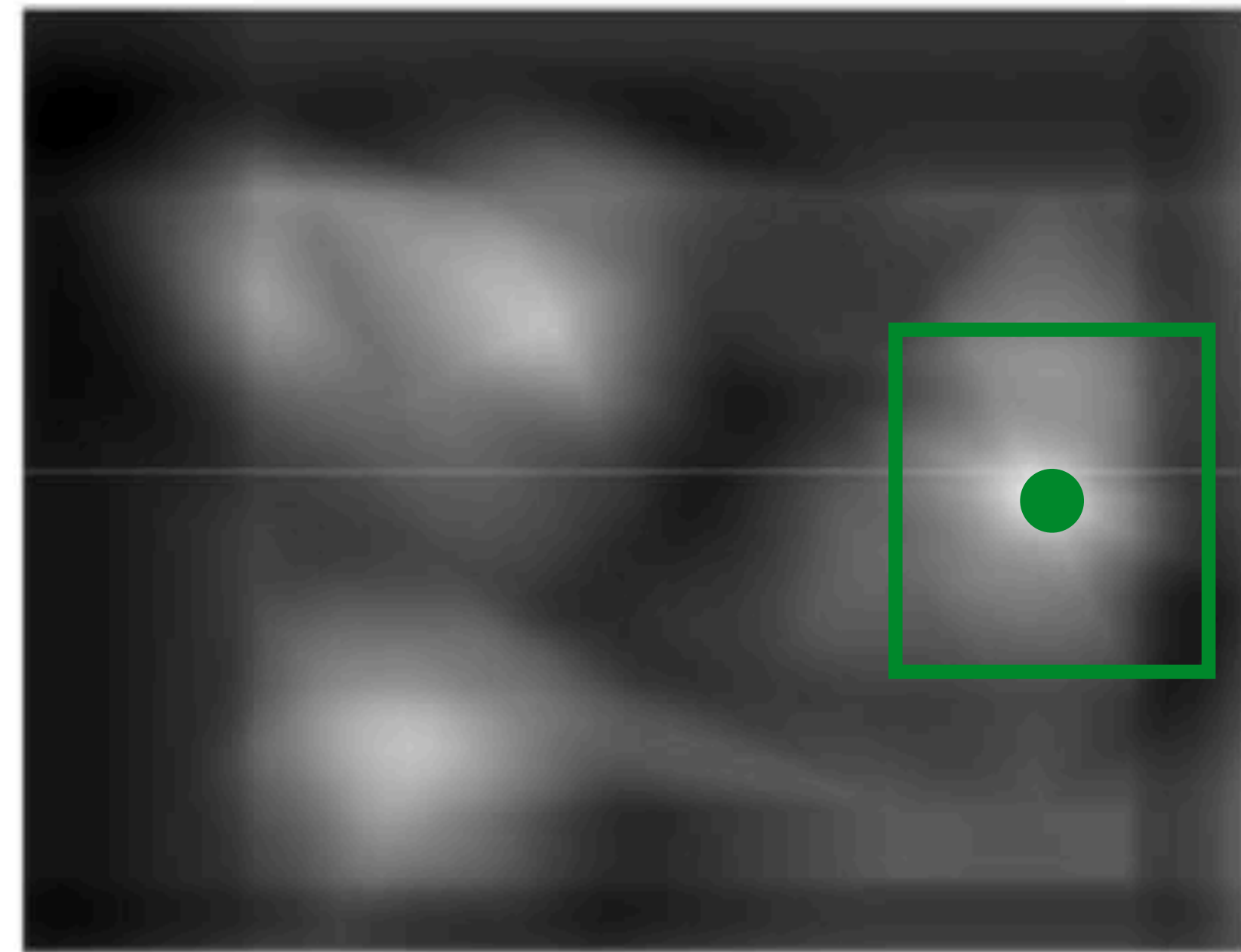
Slide Credit: Kristen Grauman

Template Matching

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



Detected template



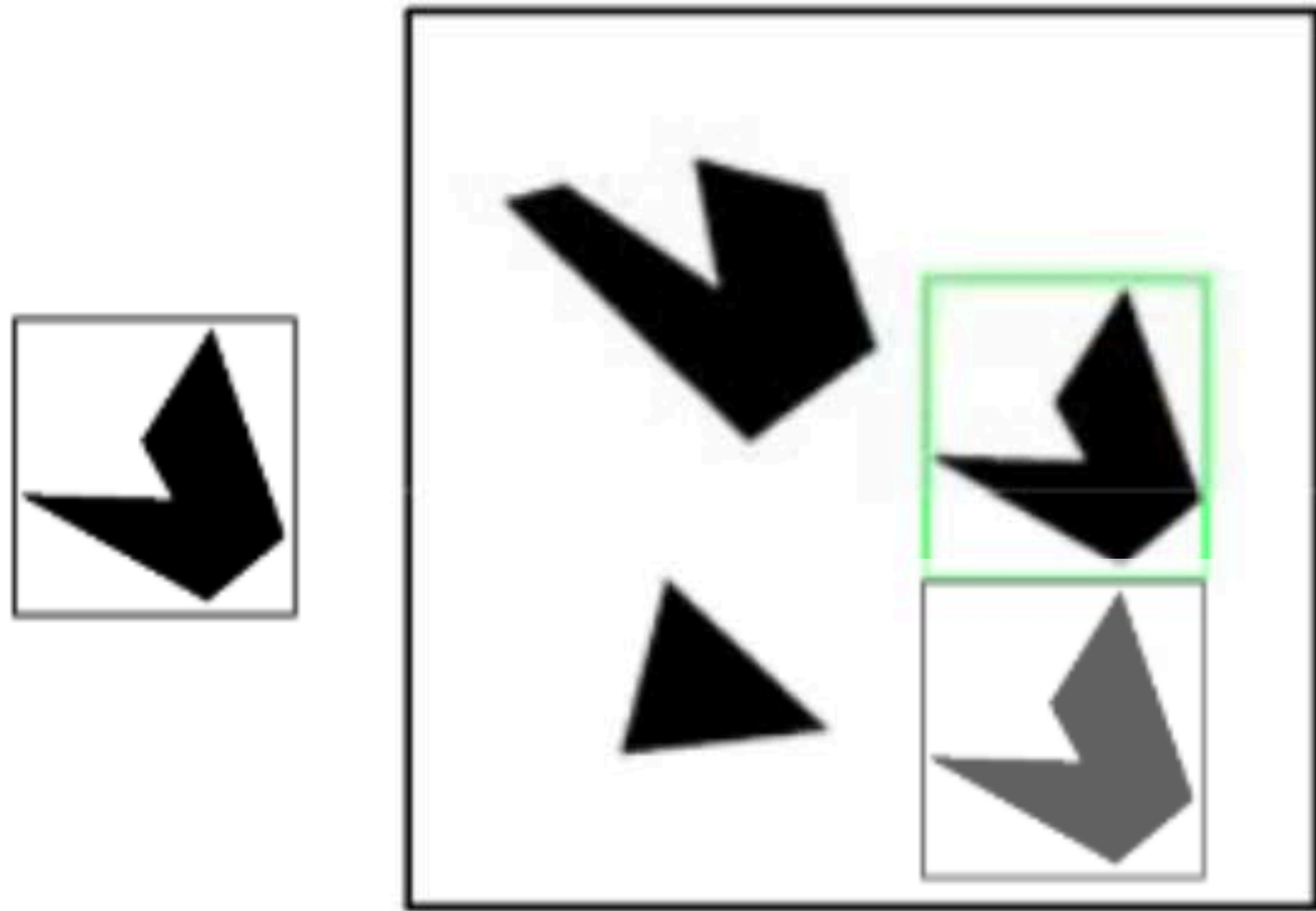
Correlation map

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

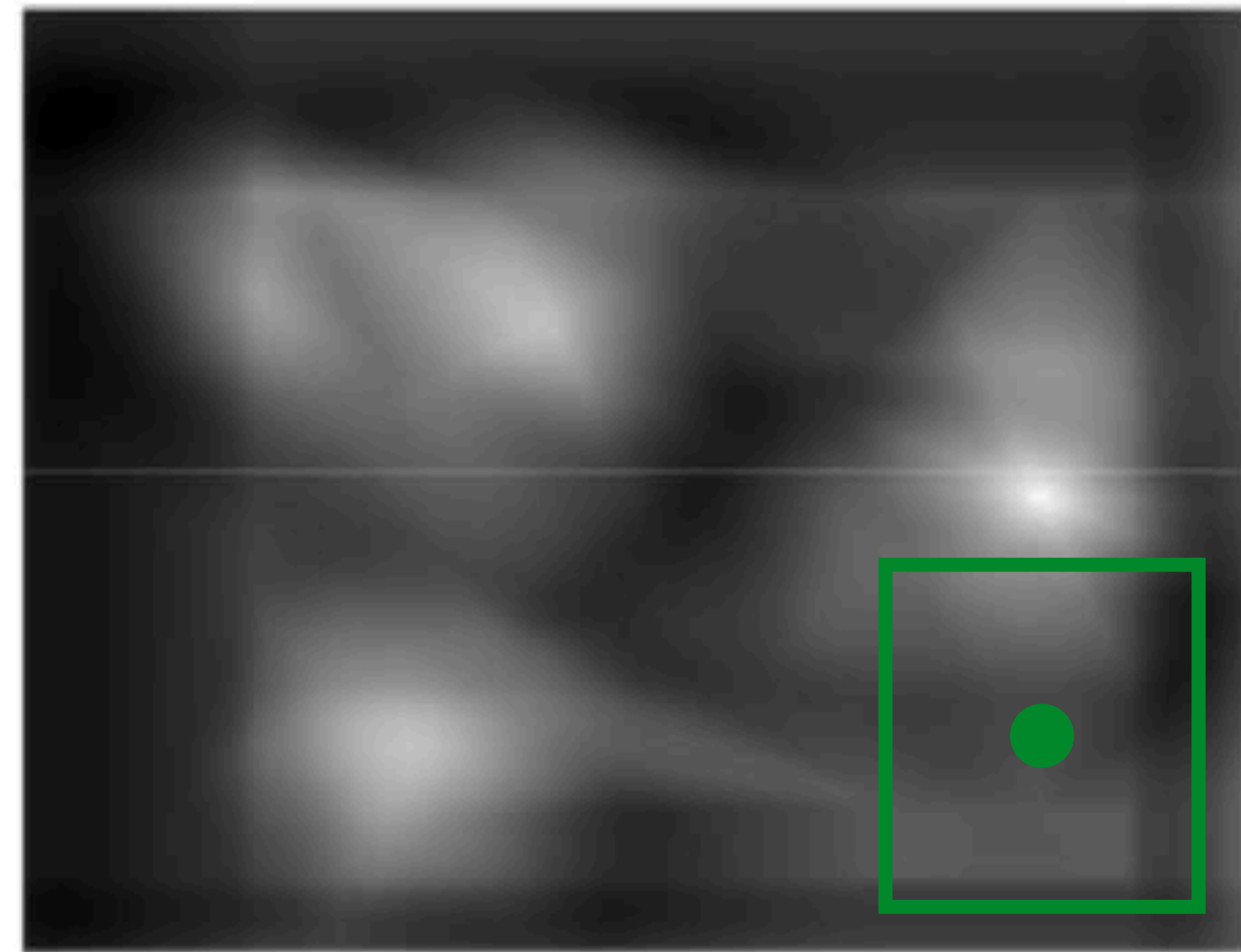
Slide Credit: Kristen Grauman

Template Matching

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



Detected template



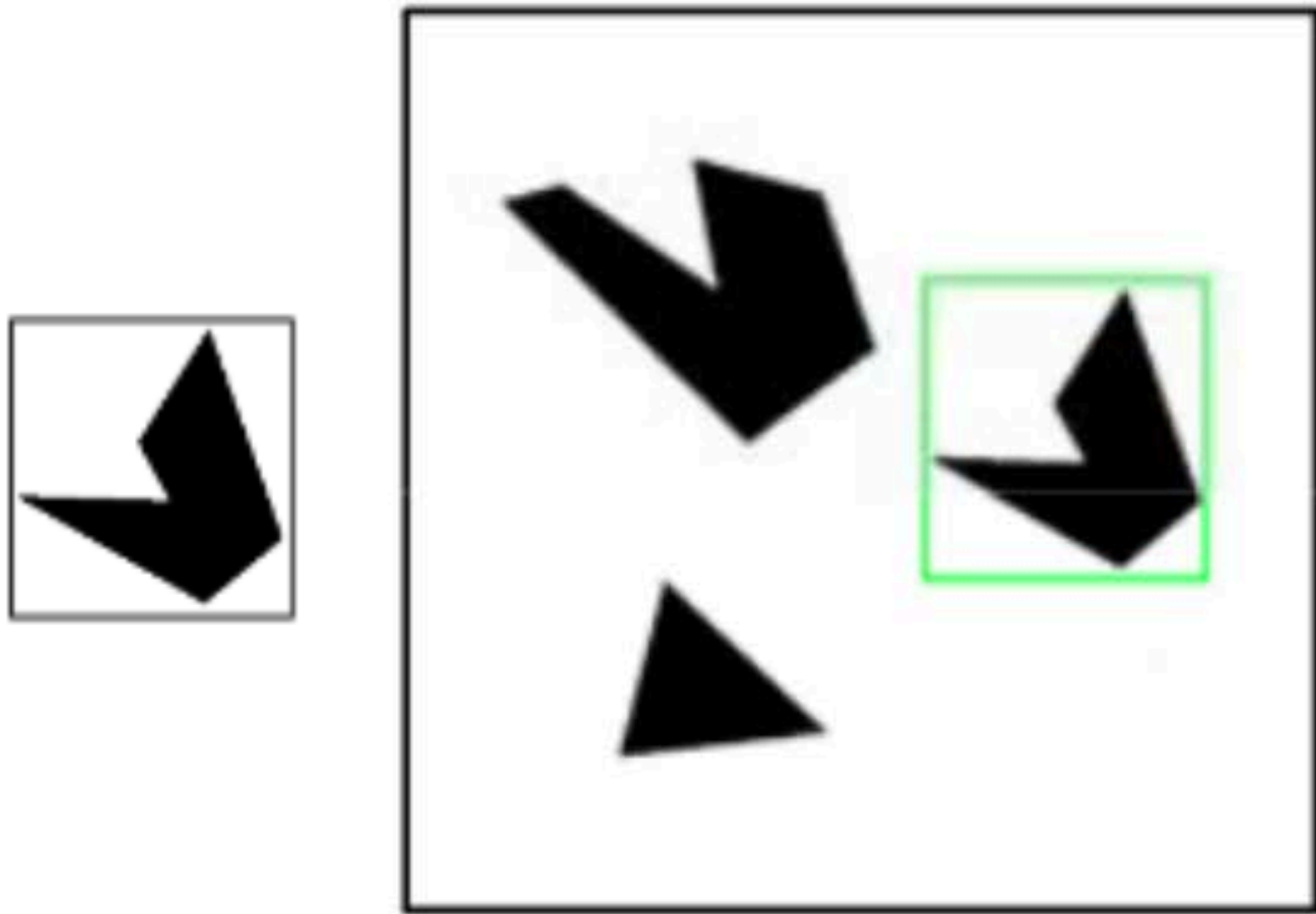
Correlation map

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

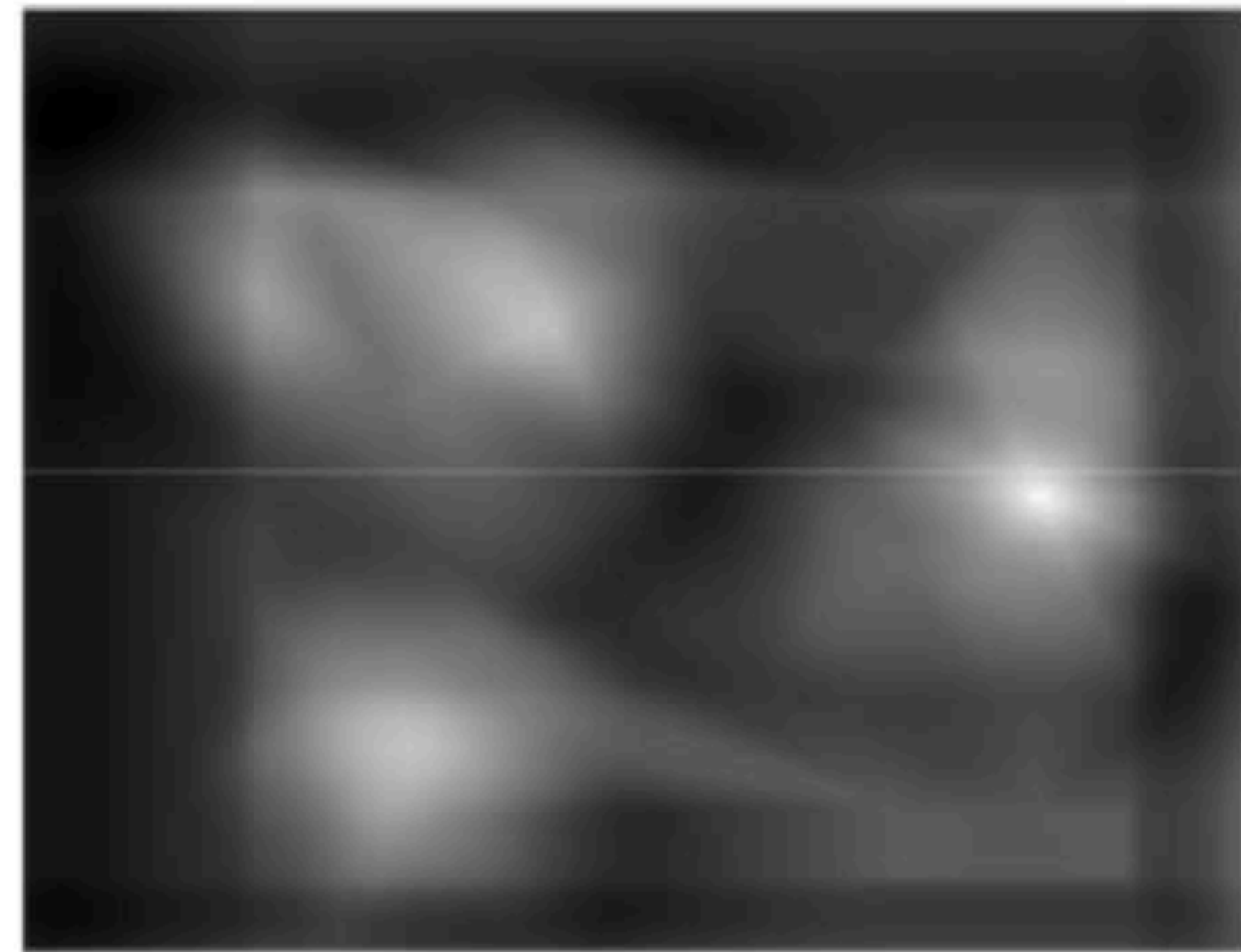
Slide Credit: Kristen Grauman

Template Matching

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



Detected template

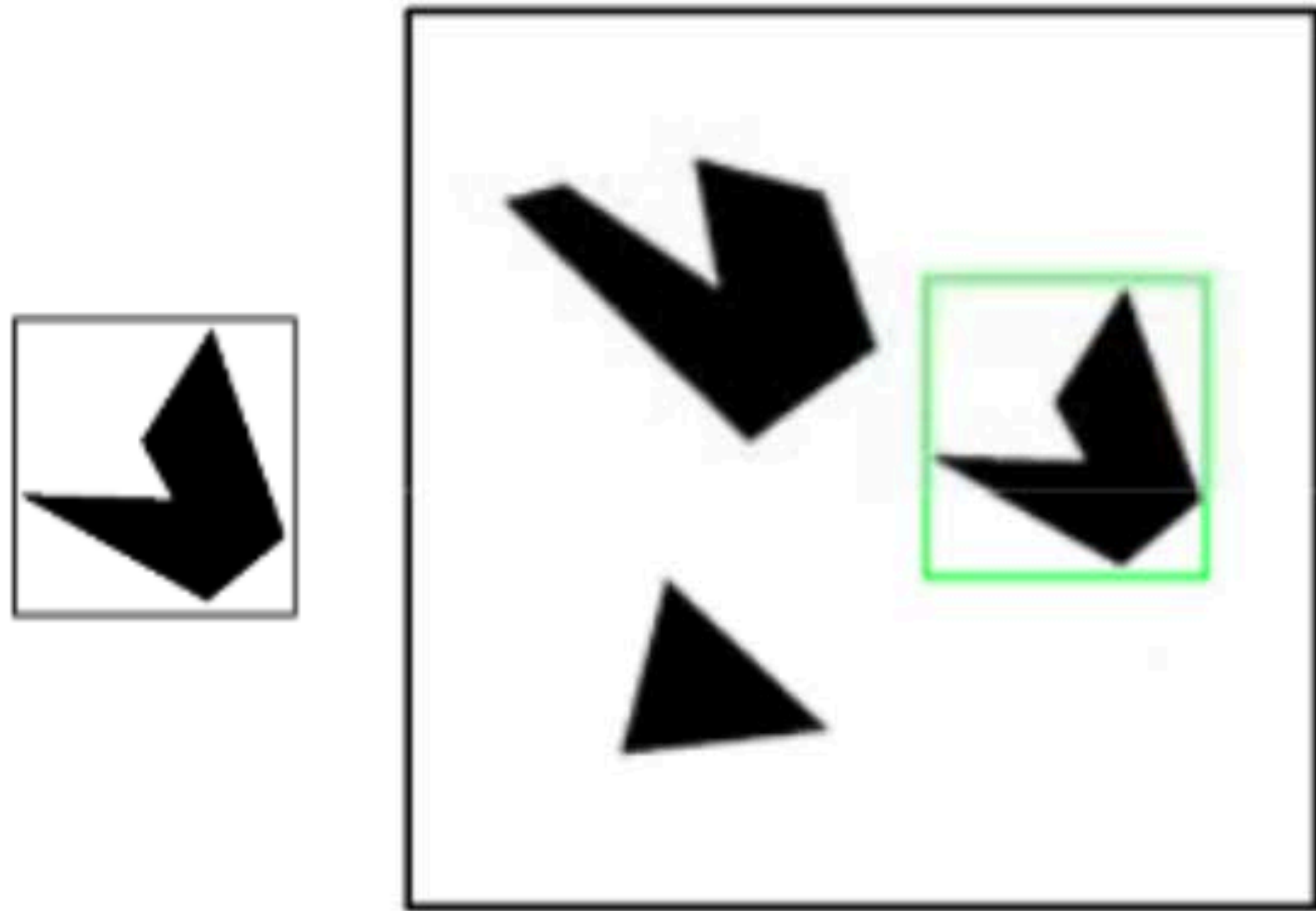


Correlation map

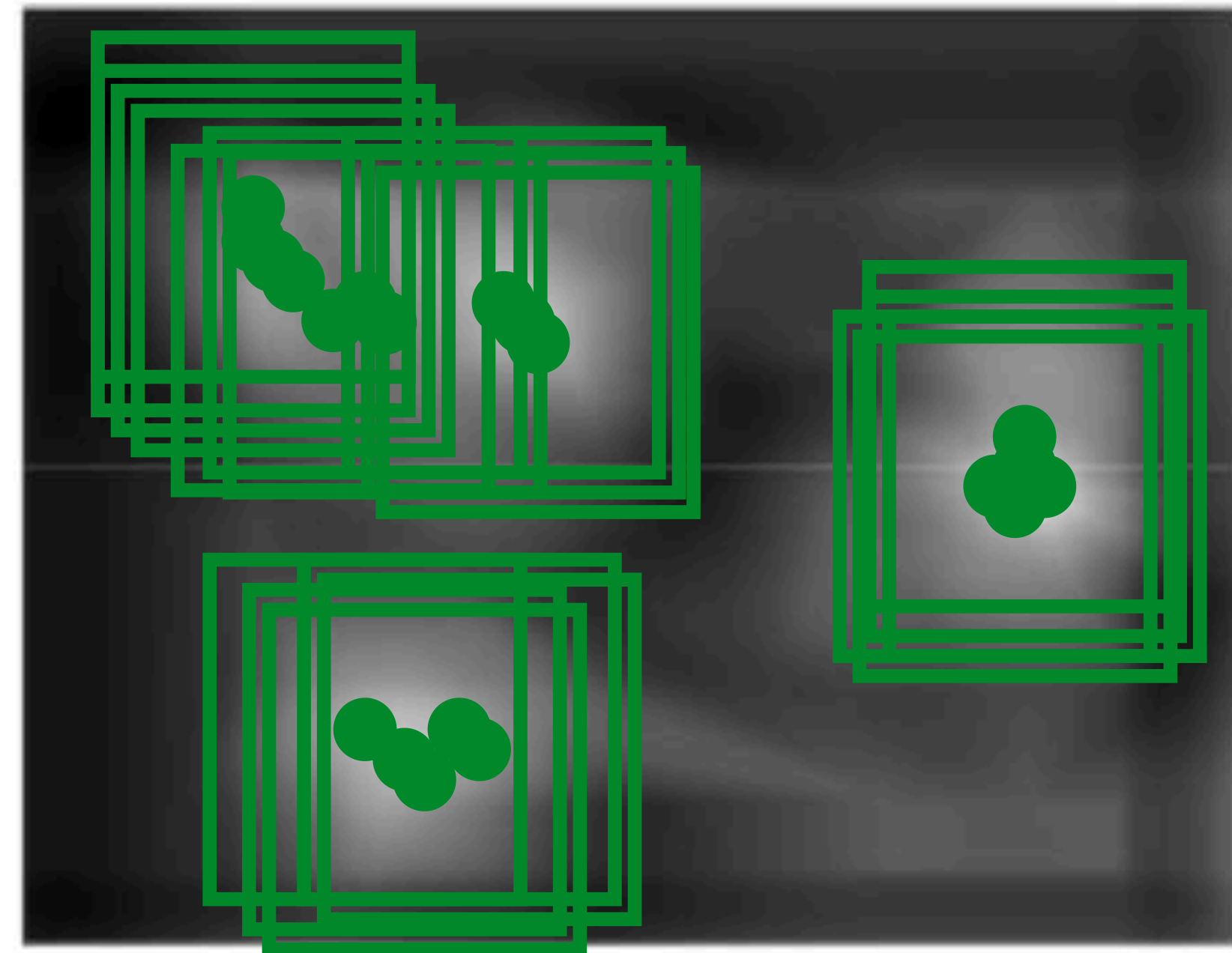
What happens if the threshold is relatively low?

Template Matching

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



Detected template

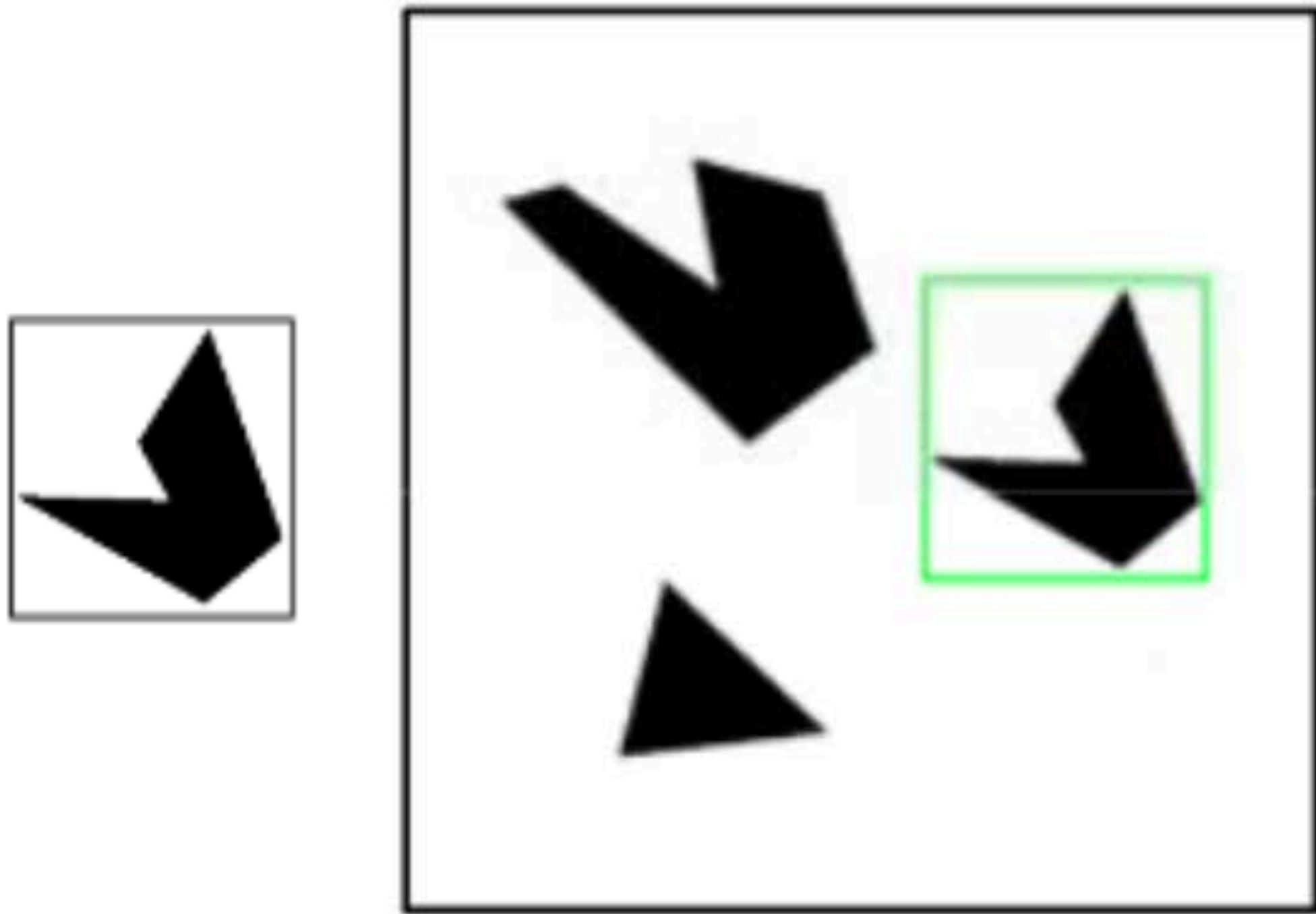


Correlation map

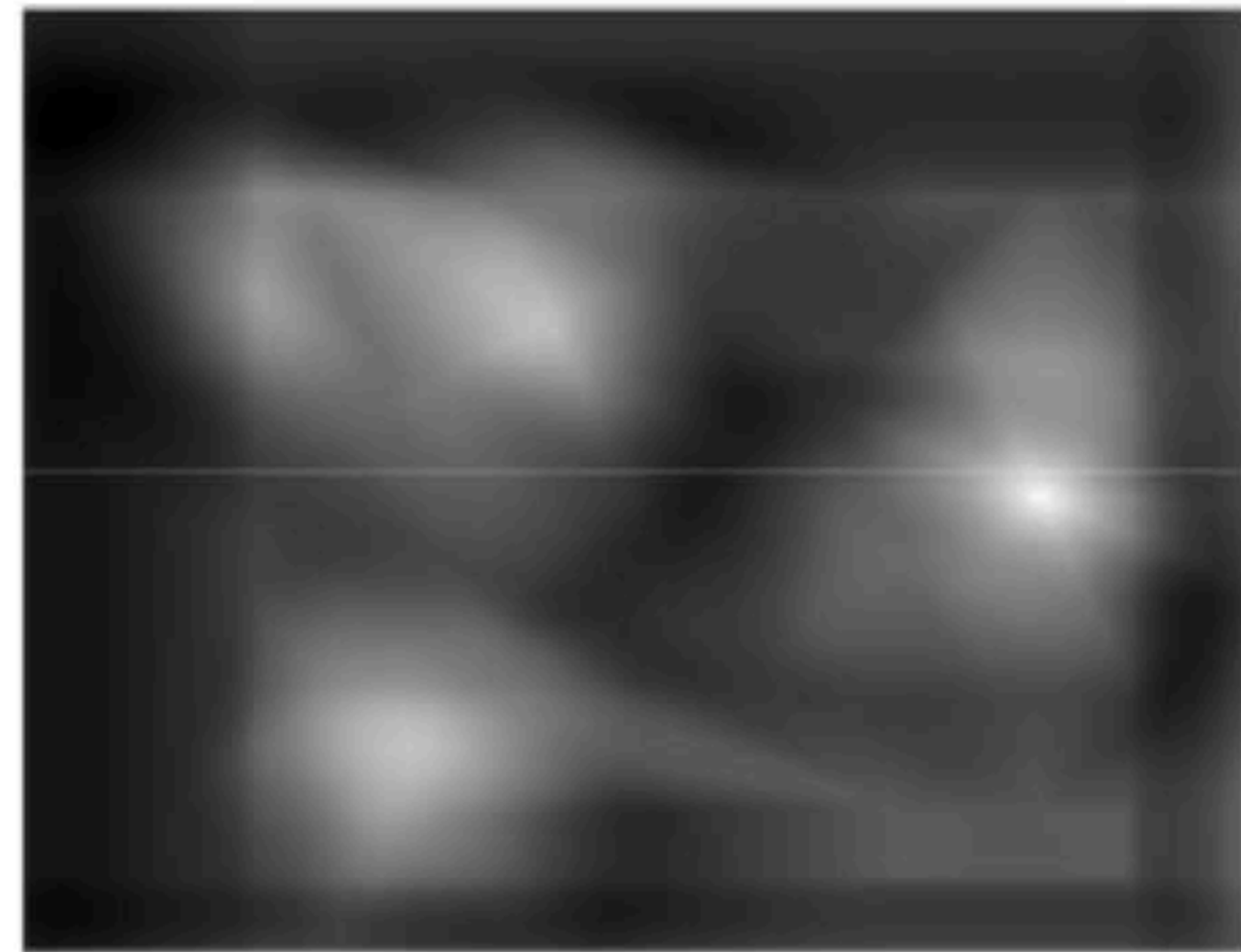
What happens if the threshold is relatively low?

Template Matching

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



Detected template

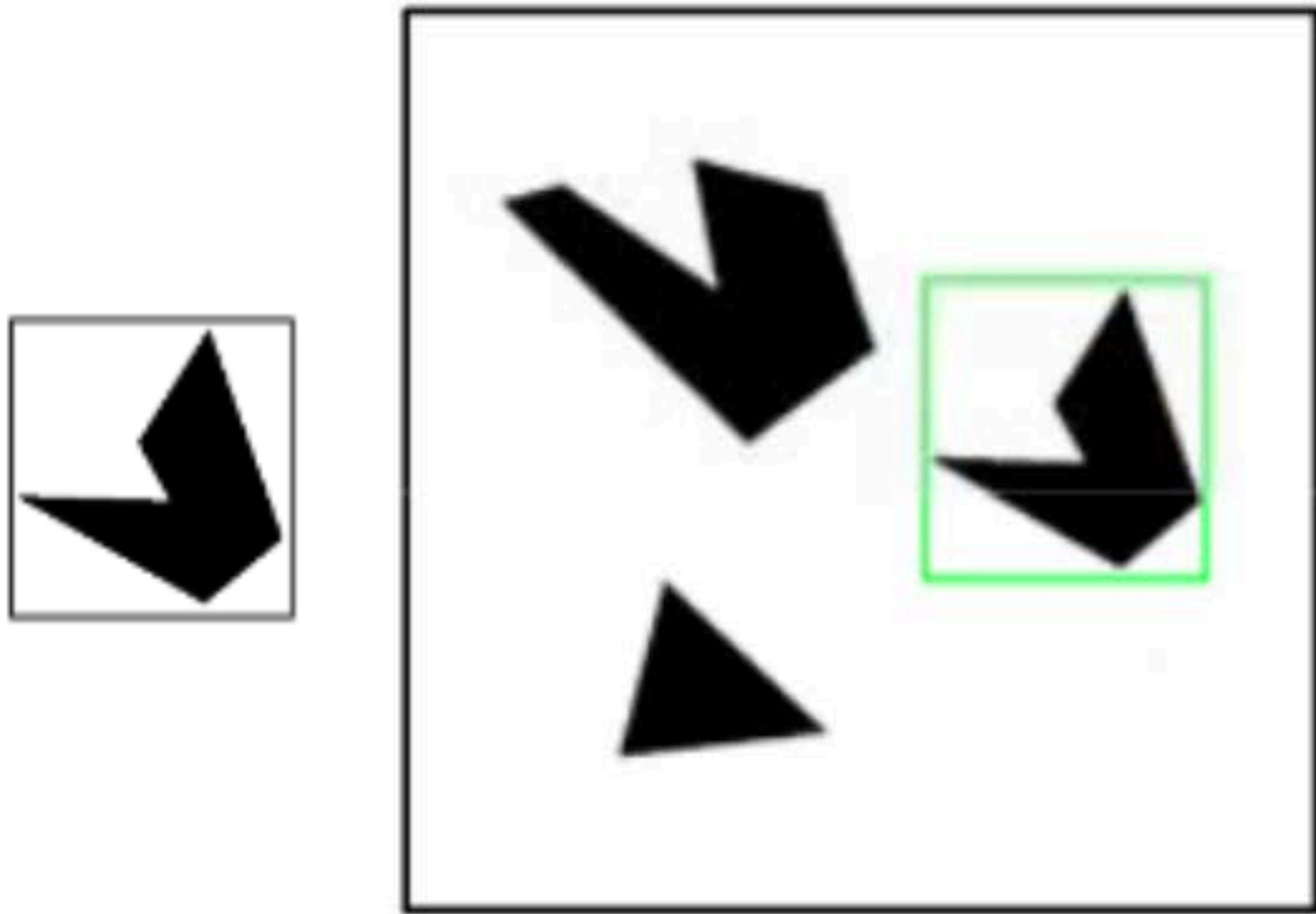


Correlation map

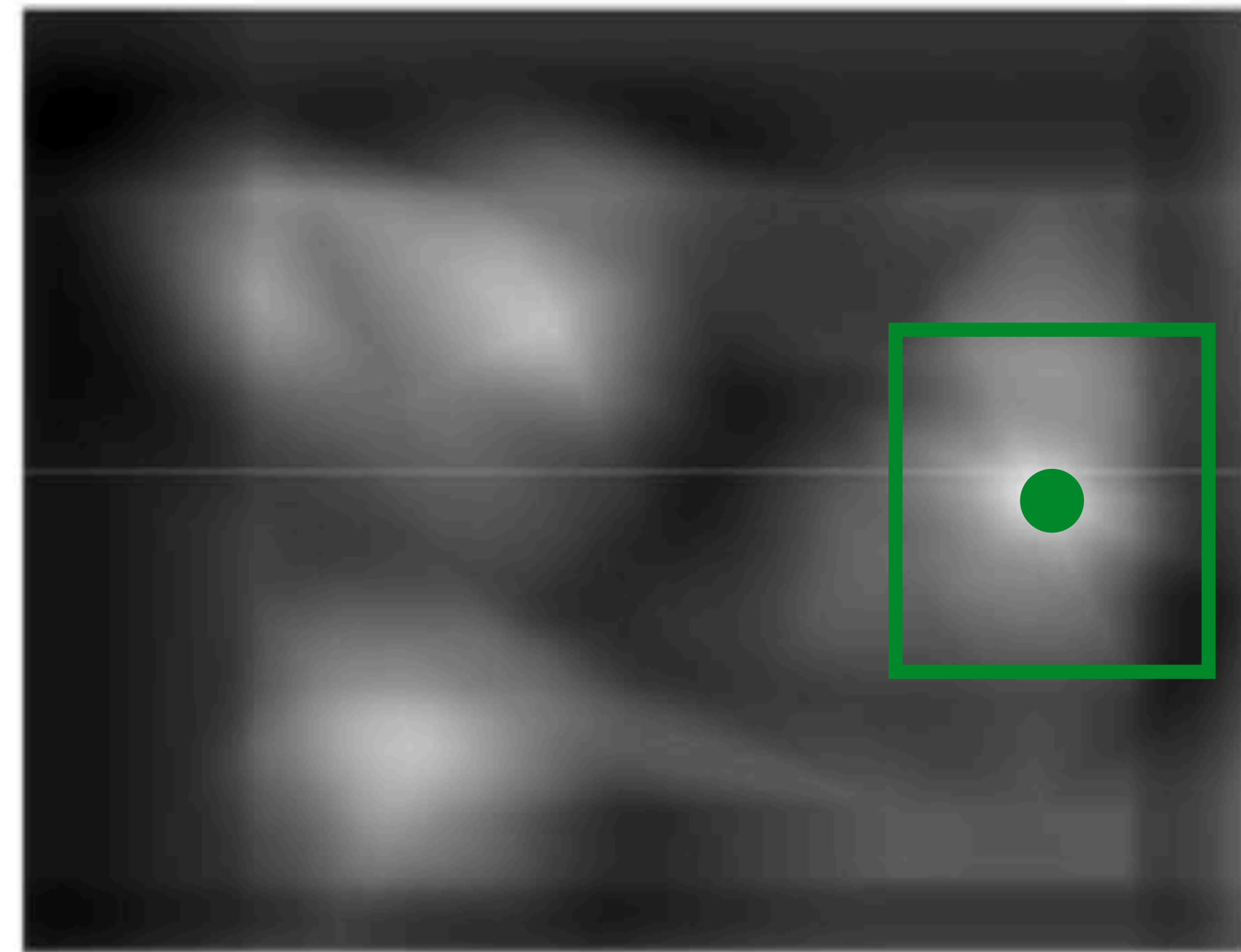
What happens if the threshold is very high (e.g., 0.99)?

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)?

Template Matching — how to implement

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

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

where \cdot is dot product and $| |$ is vector magnitude

Template Matching — how to implement

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

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

where \cdot is dot product and $| |$ is vector magnitude

1. Normalize the template / filter (b) in the beginning

Template Matching — how to implement

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

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

where \cdot 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

Template Matching — how to implement

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

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

where \cdot 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 correlation 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 in Step 2

Template Matching

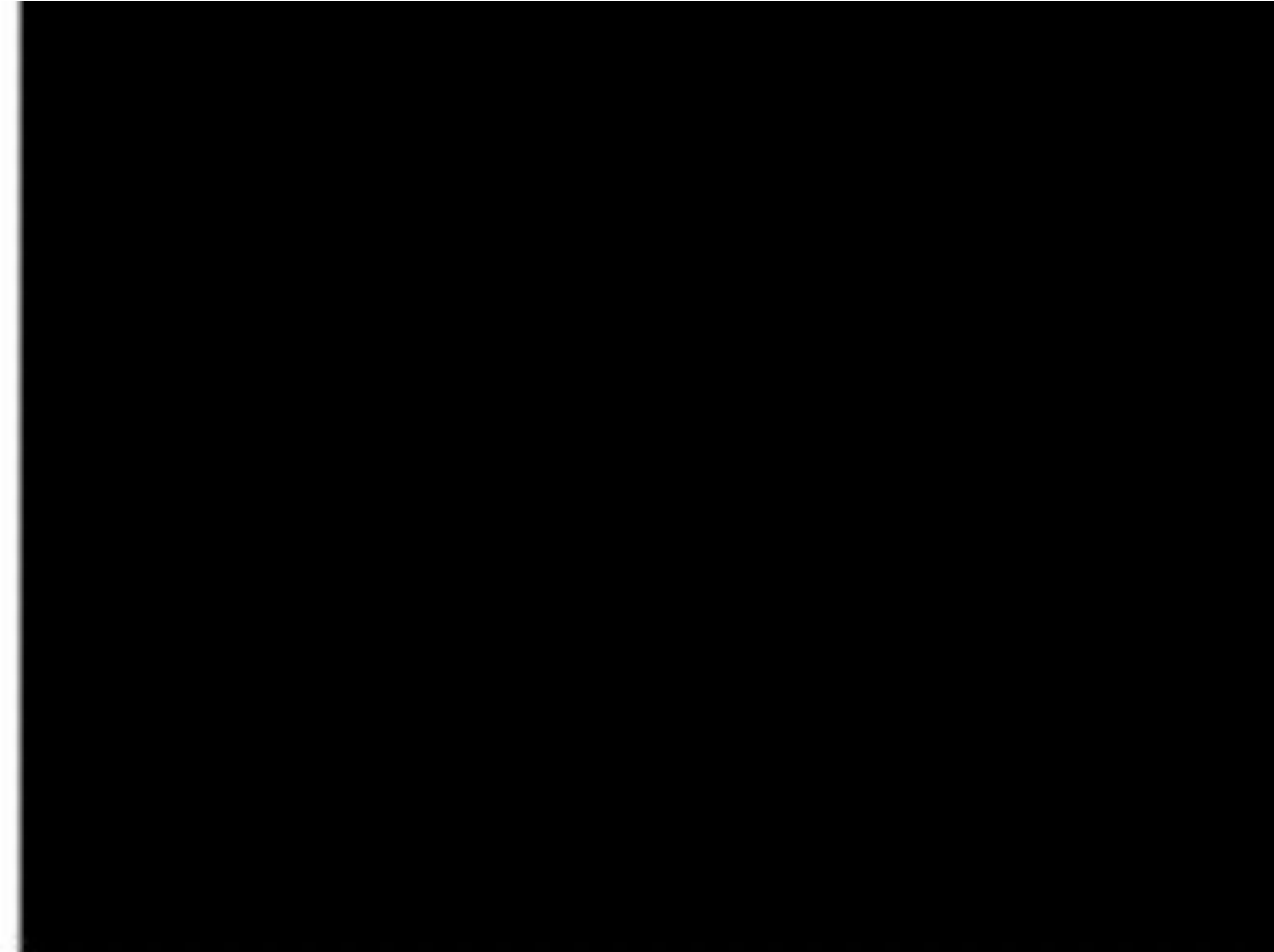
Linear filtering the entire image computes the entire set of dot products, one for each possible alignment of filter and image

Important **Insight**:

- filters look like the pattern they are intended to find
- filters find patterns they look like

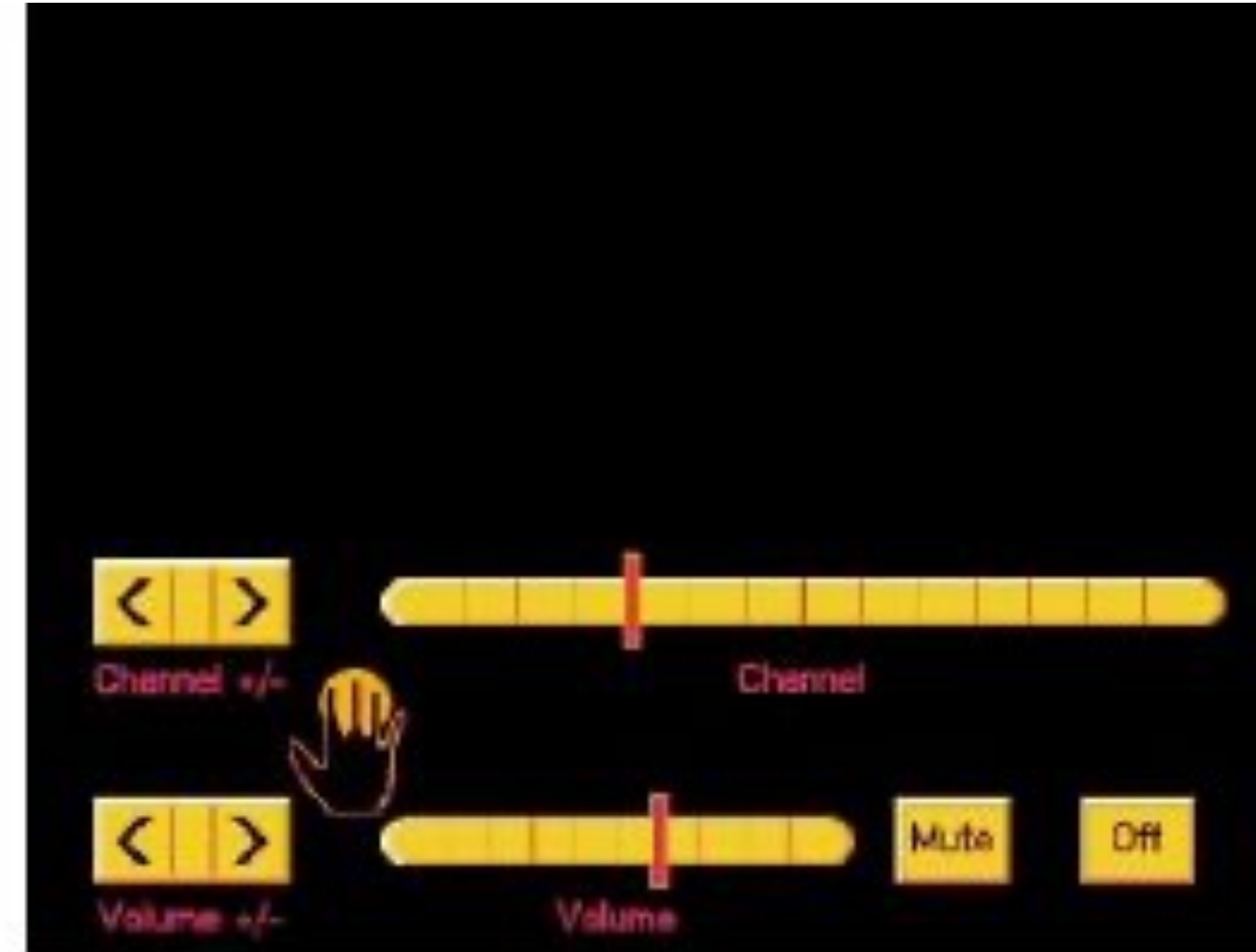
Linear filtering is sometimes referred to as **template matching**

Example 1:



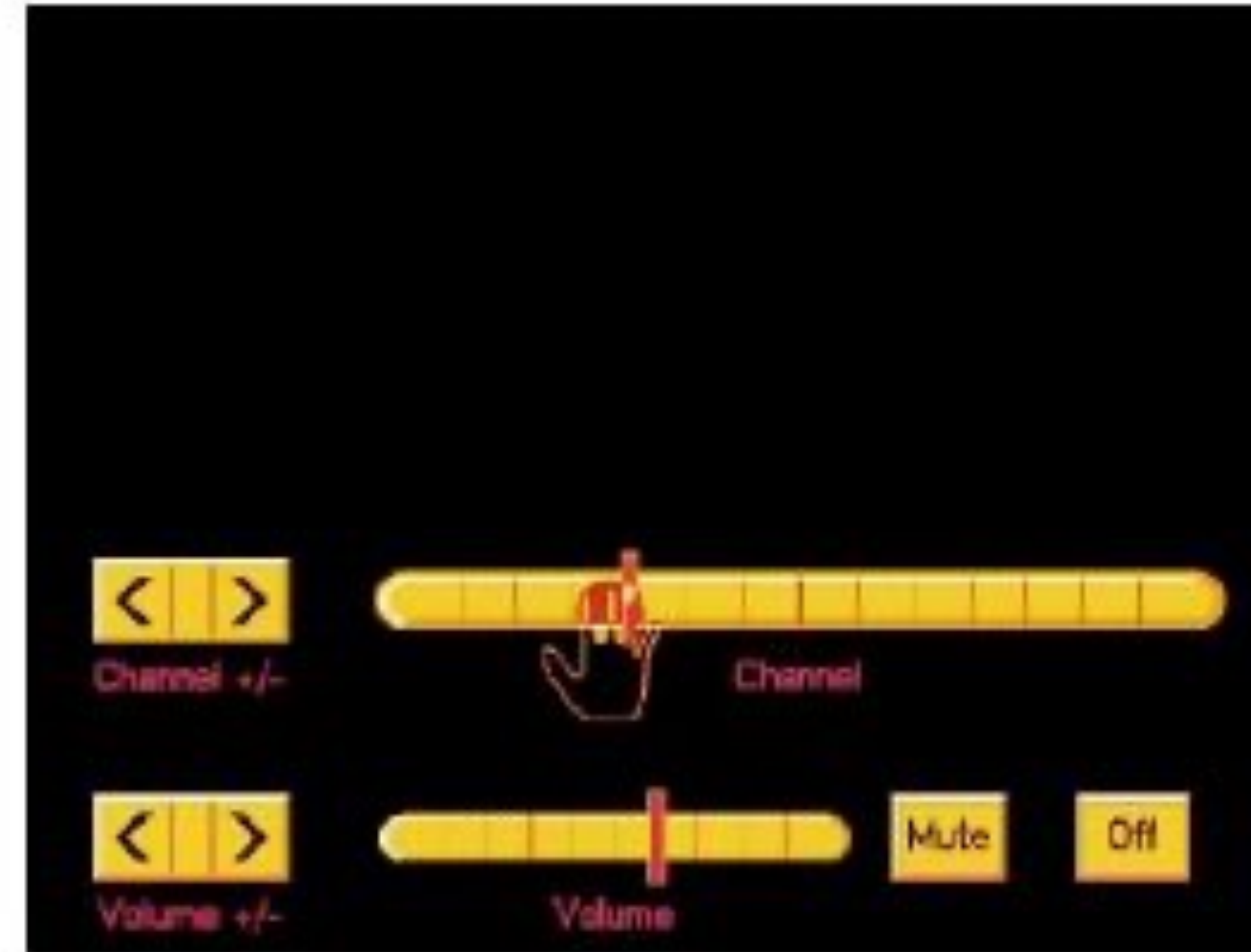
Credit: W. Freeman et al., “Computer Vision for Interactive Computer Graphics,”
IEEE Computer Graphics and Applications, 1998

Example 1:



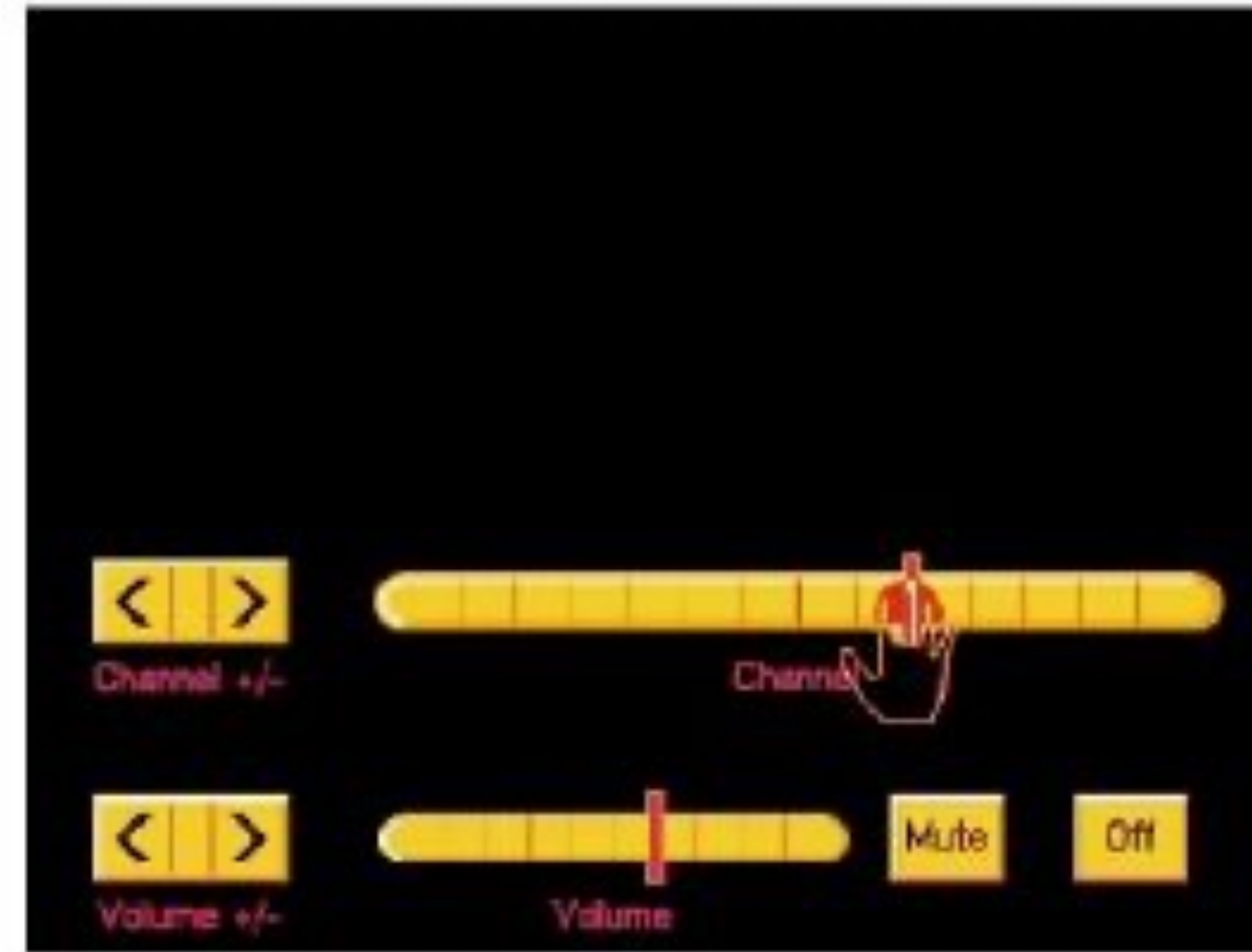
Credit: W. Freeman et al., “Computer Vision for Interactive Computer Graphics,”
IEEE Computer Graphics and Applications, 1998

Example 1:



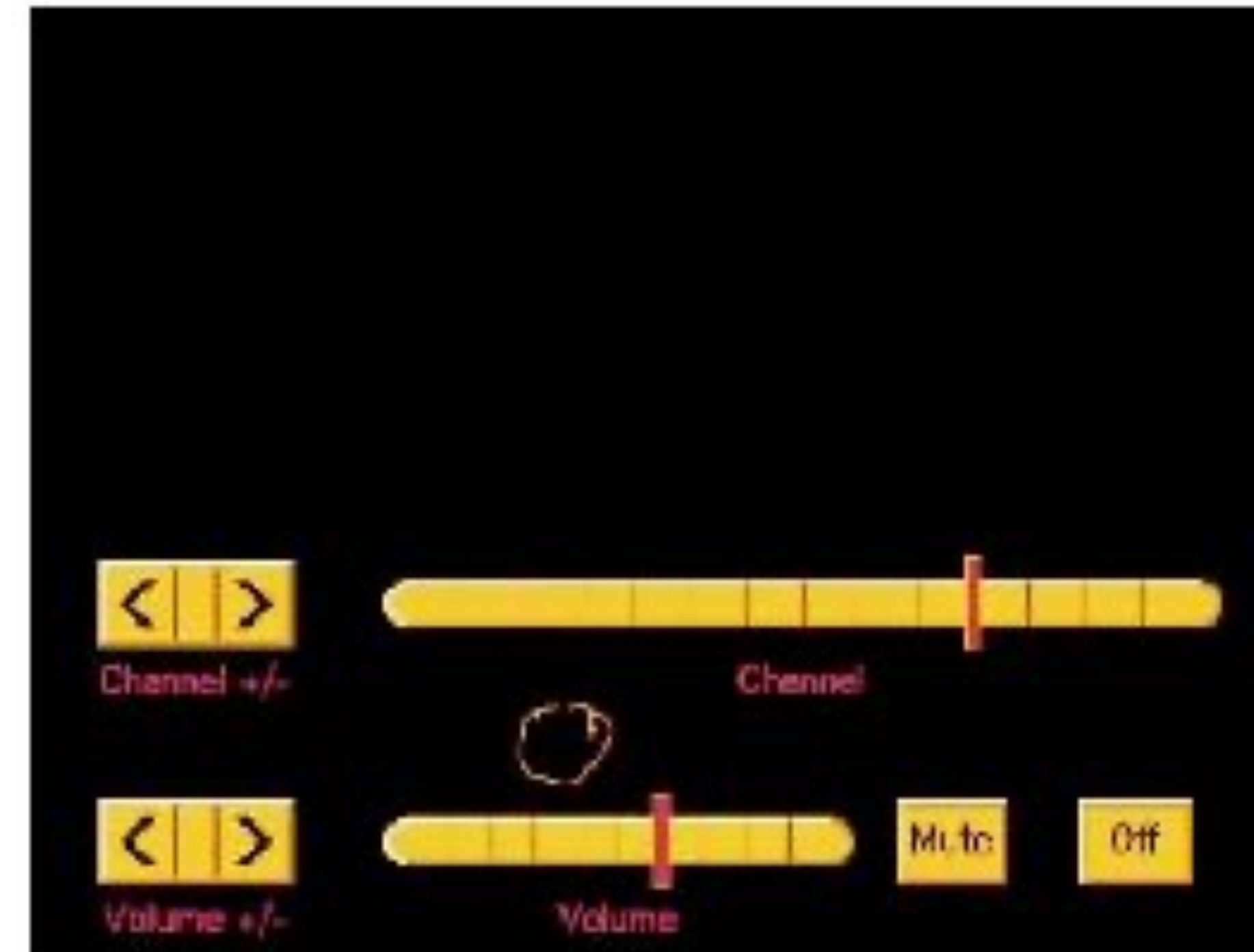
Credit: W. Freeman et al., "Computer Vision for Interactive Computer Graphics," IEEE Computer Graphics and Applications, 1998

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

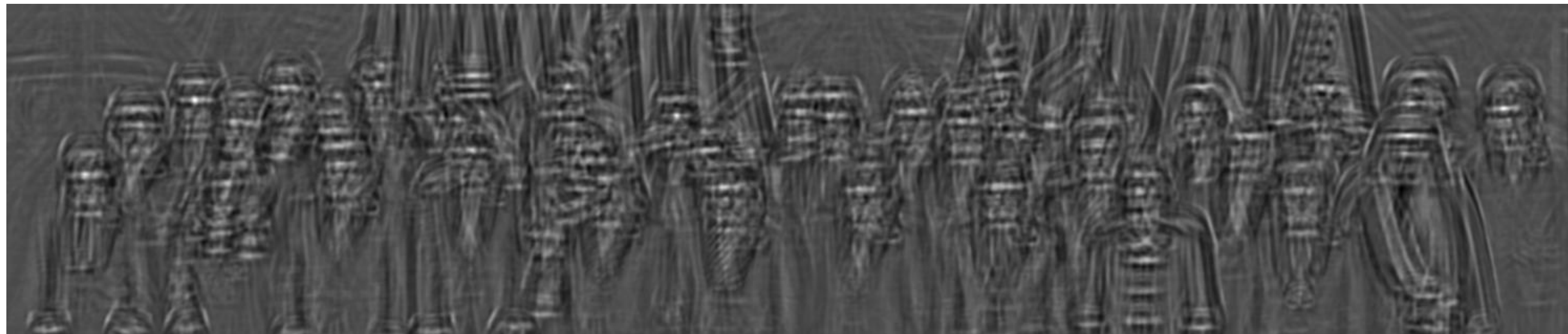
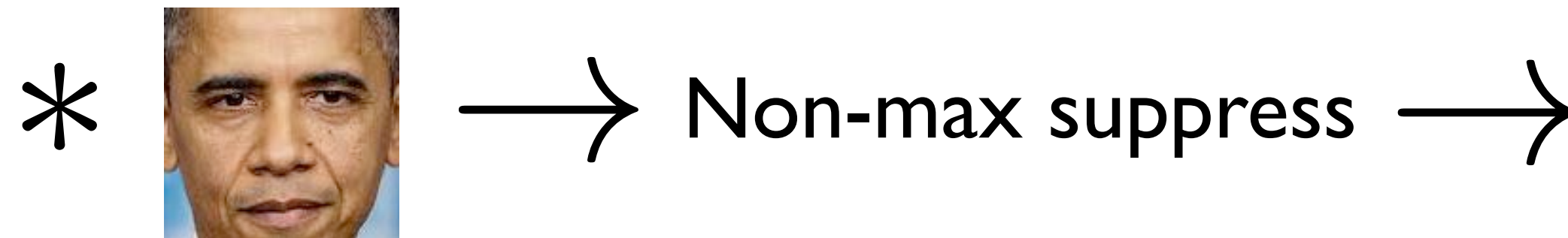


— Different Perspective

— Motion / blur

Template Matching

Convolve image with template, find local maxima



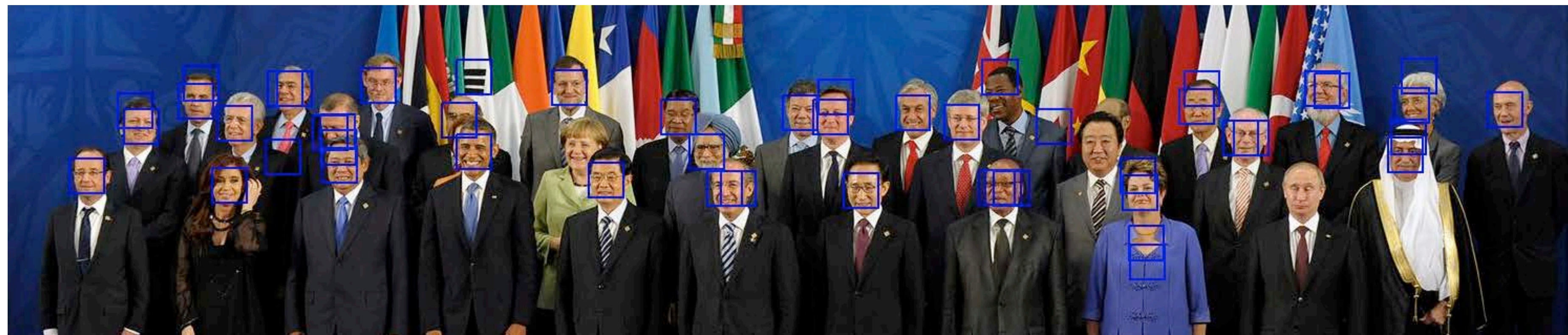
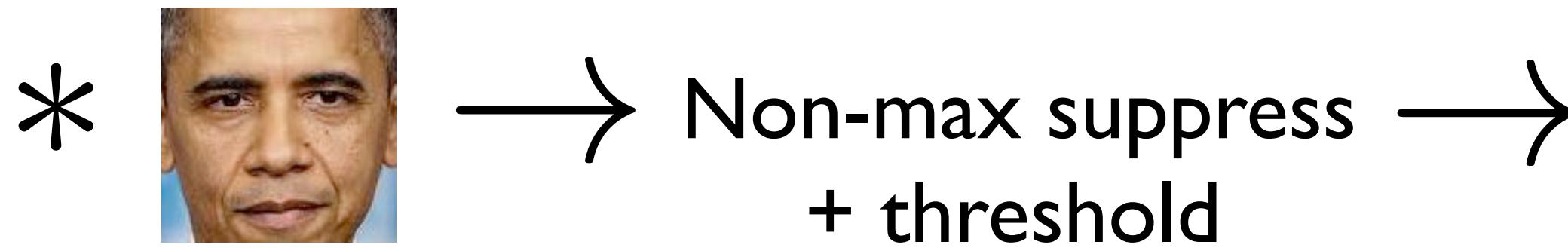
Template Matching

Convolve image with template, find local maxima



Template Matching

Convolve image with template, find local maxima



Detection Performance

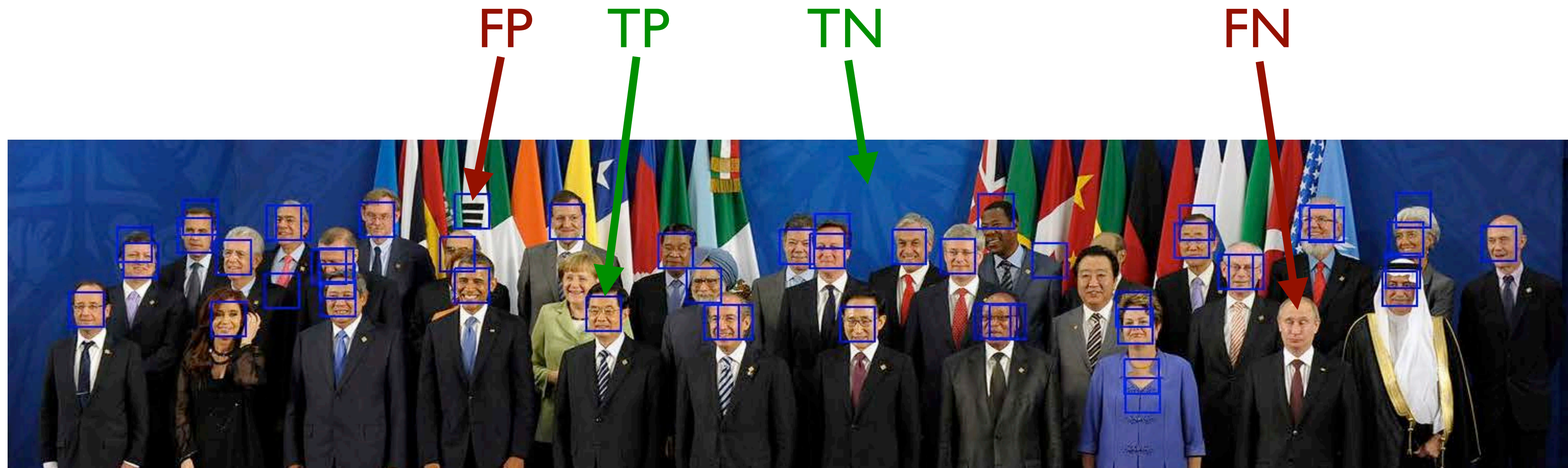
Types of error in detection:

TP = True positive (true face and detected)

FP = False positive (not face and detected)

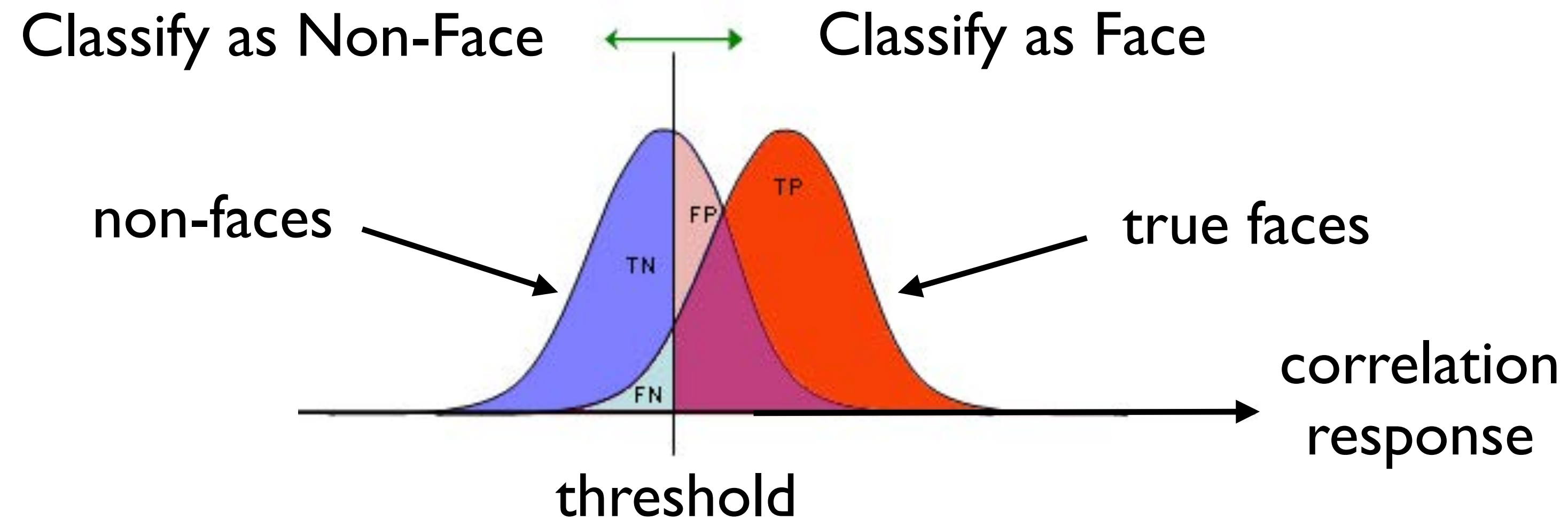
TN = True negative (not face and no detection)

FN = False negative (true face and not detected)



Detection Performance

Depending on where we set the threshold, we can tradeoff between true positives and false positives



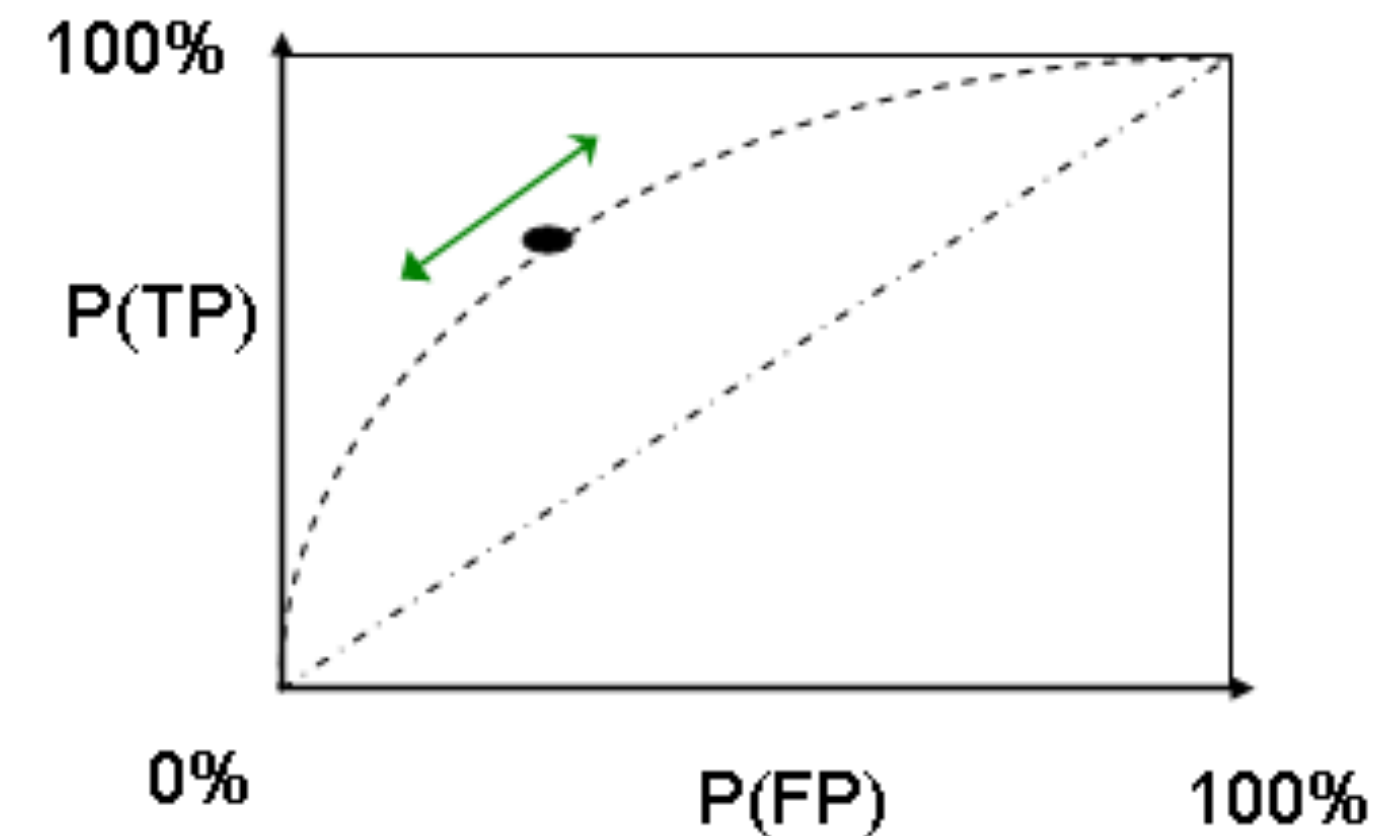
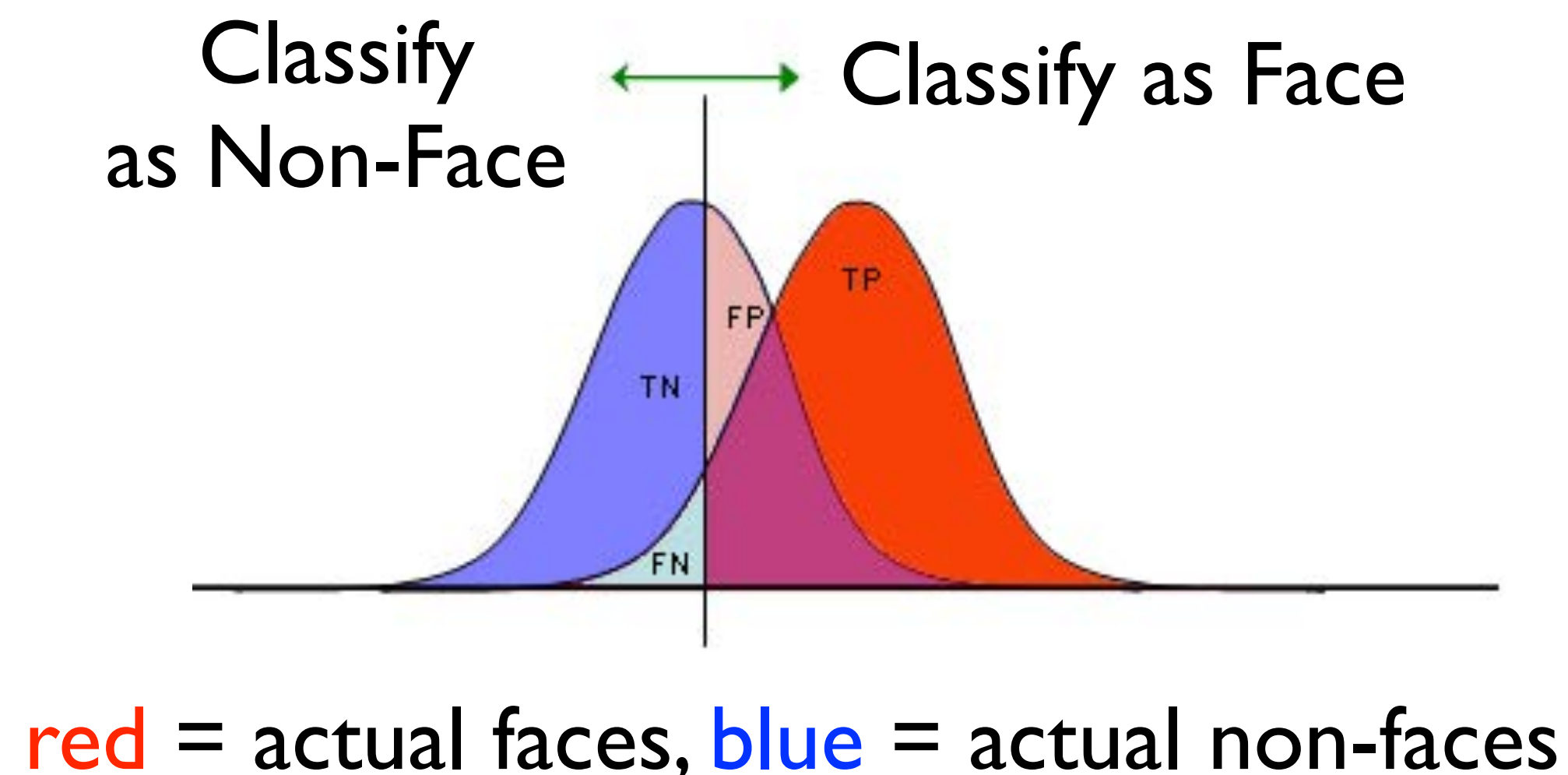
ROC Curves

Note that we can easily get 100% true positives (if we are prepared to get 100% false positives as well!)

It is a tradeoff between **true positive rate (TP)** and **false positive rate (FP)**

We can plot a curve of all TP rates vs FP rates by varying the classifier threshold

This is a **Receiver Operating Characteristic (ROC)** curve



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