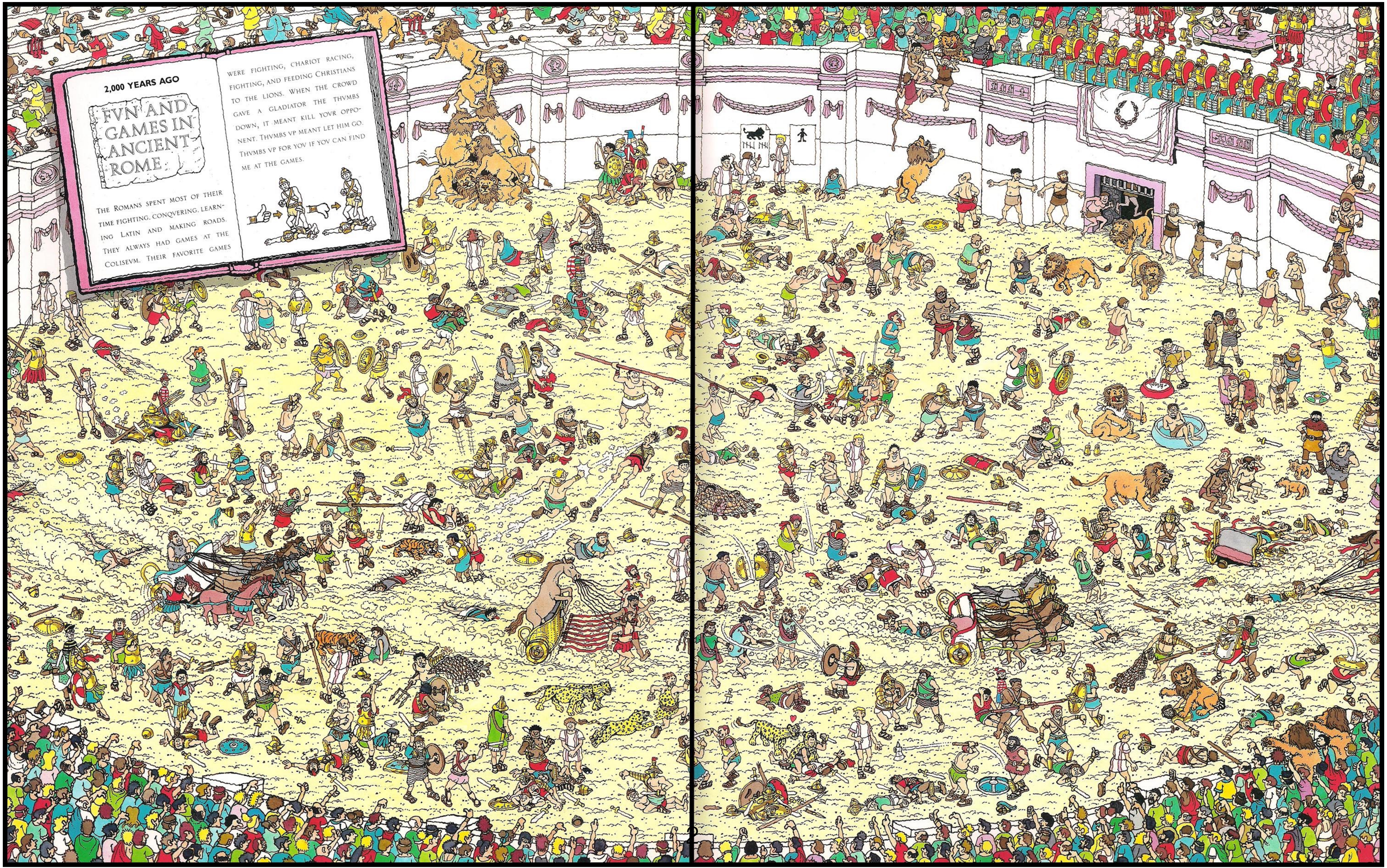
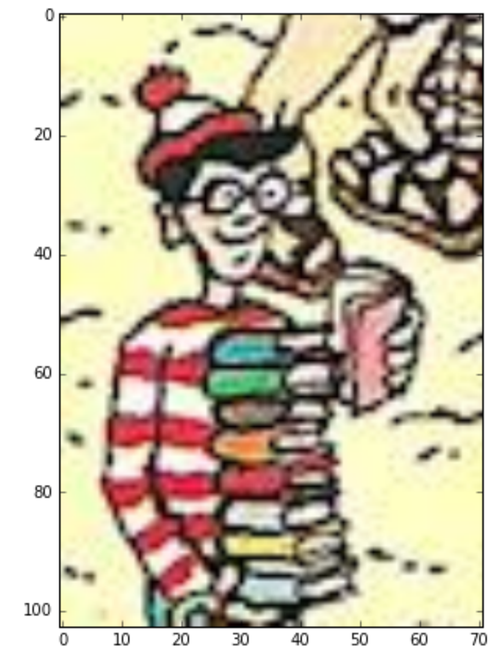
Template Matching

Faisal Qureshi
faisal.qureshi@uoit.ca

Where is Waldo?

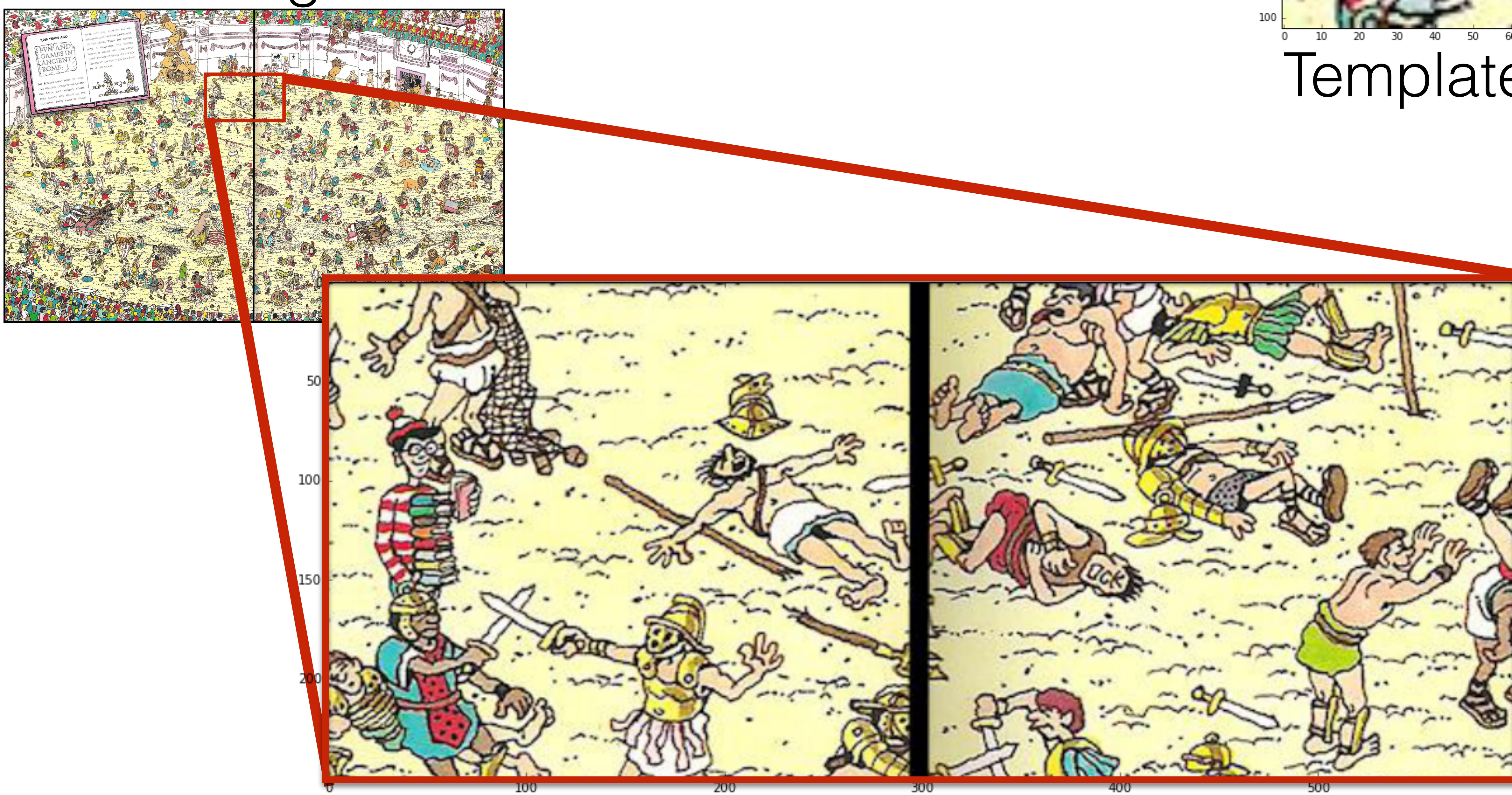


Where is Waldo?



Template

Image



Template Matching

- What is a good similarity or distance measure between two patches?

Patch 1

Patch 2

Patch 3

How to determine if Patch 1 is closer to Patch 2 or 3?

Template Matching

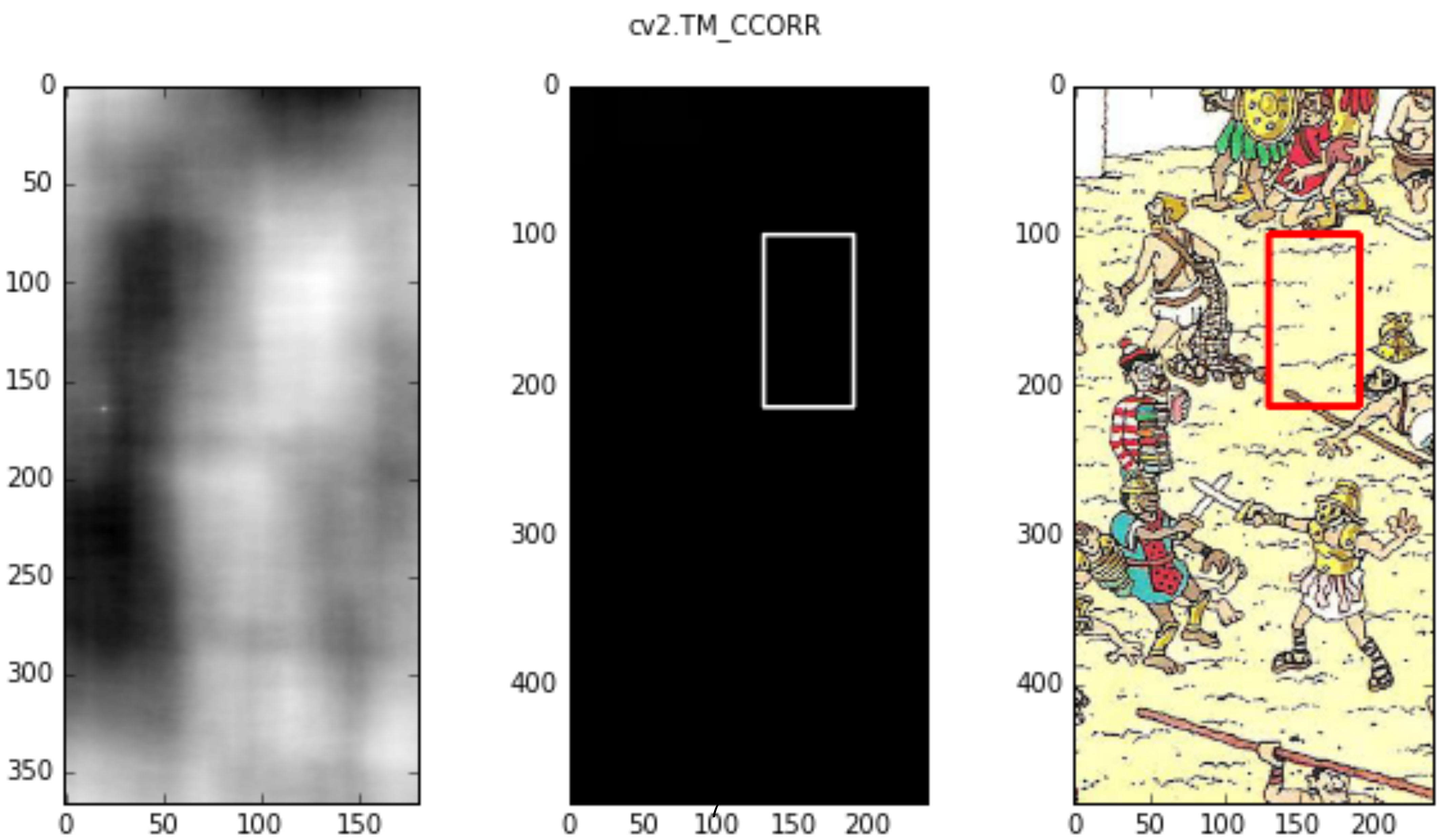
- Find Waldo in the image

Template Matching

- Cross-correlation
- Normalized cross-correlation
- Sum-of-squared differences
- Normalized sum-of-squared differences
- Correlation coefficient
- Normalized correlation coefficient

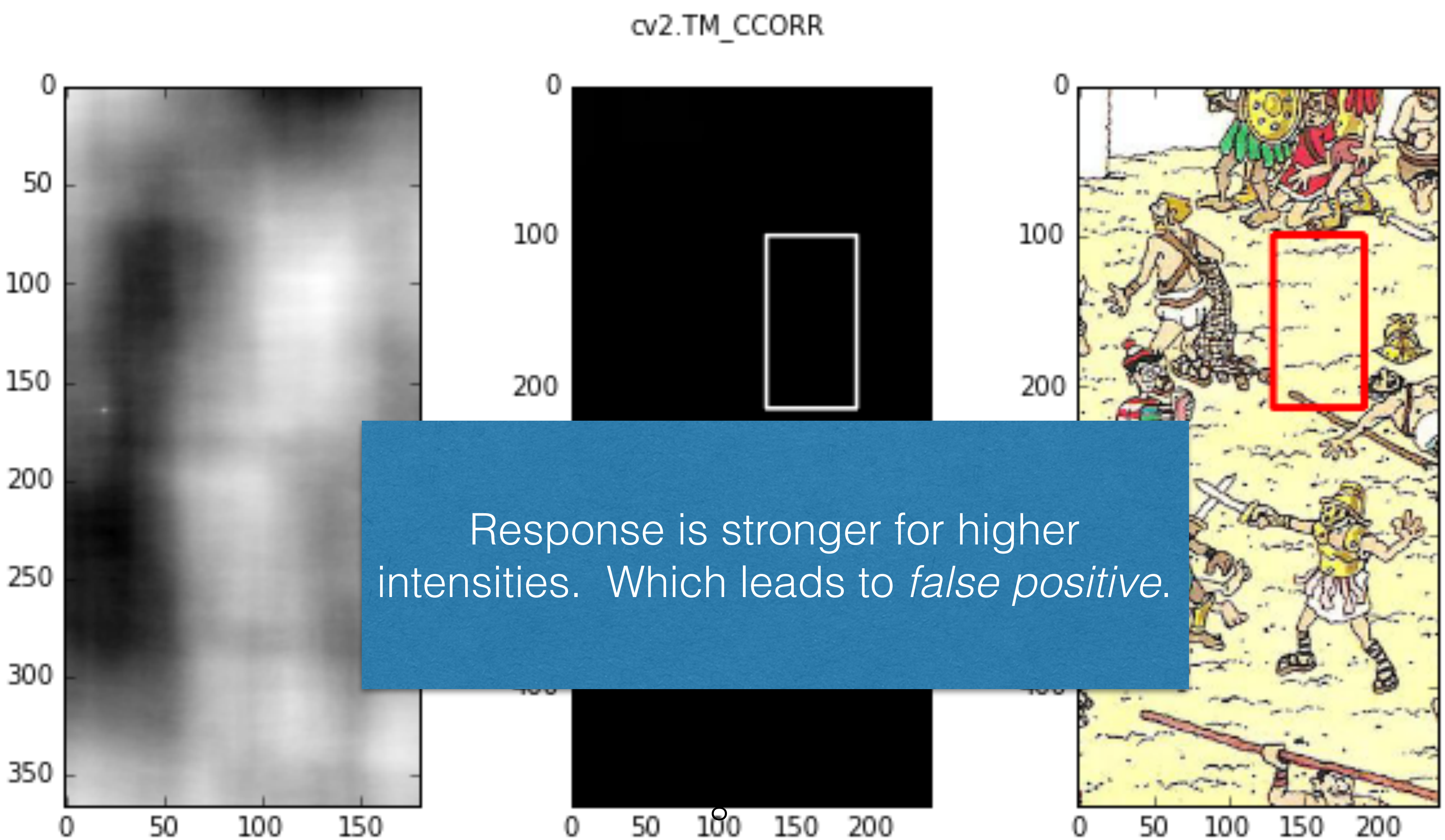
Cross-Correlation

$$\mathbf{g}(i,j) = \sum_{k,l} \mathbf{f}(i+k, j+l)\mathbf{h}(k,l)$$



Cross-Correlation

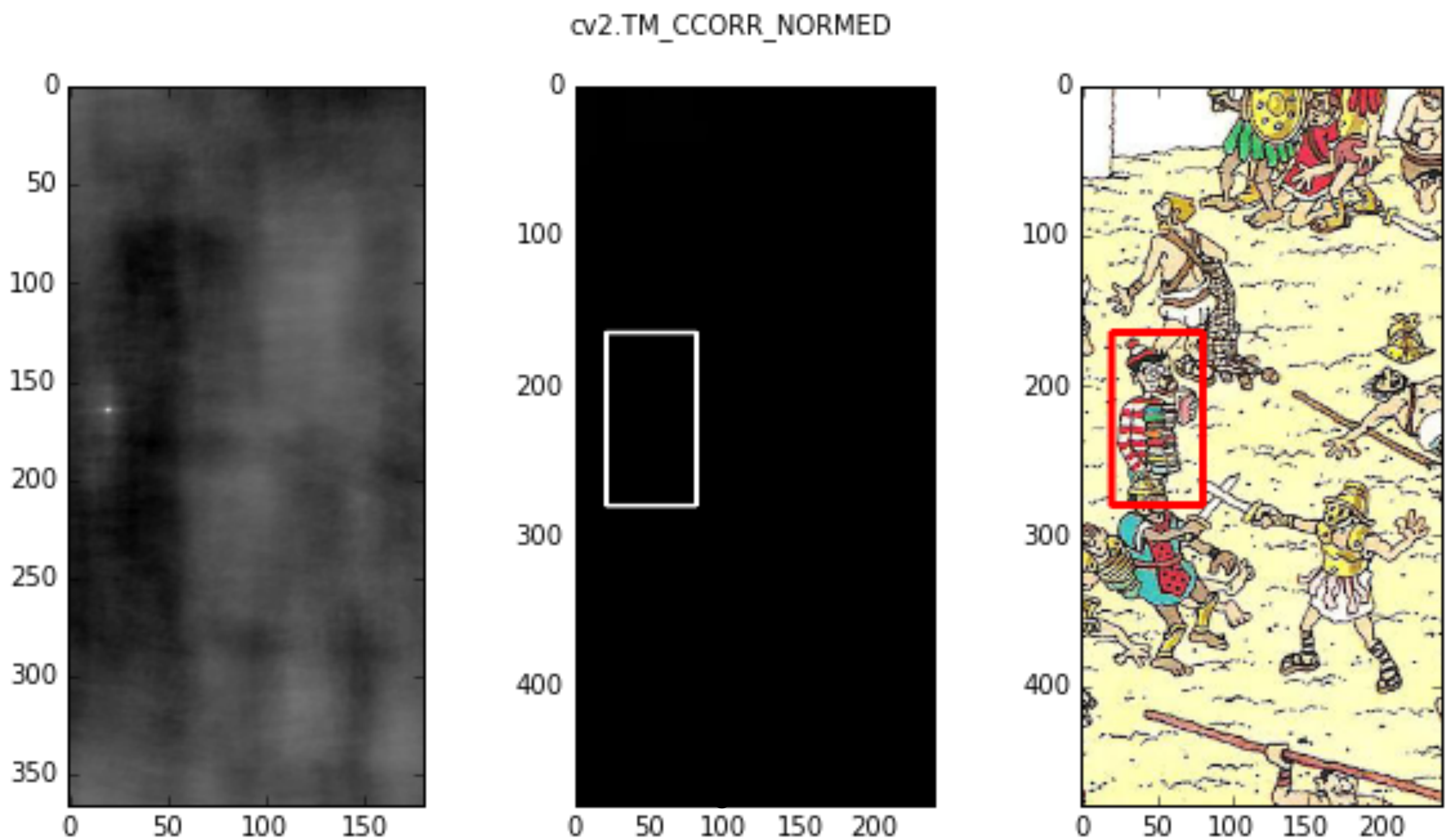
$$\mathbf{g}(i,j) = \sum_{k,l} \mathbf{f}(i+k, j+l)\mathbf{h}(k,l)$$



Response is stronger for higher intensities. Which leads to *false positive*.

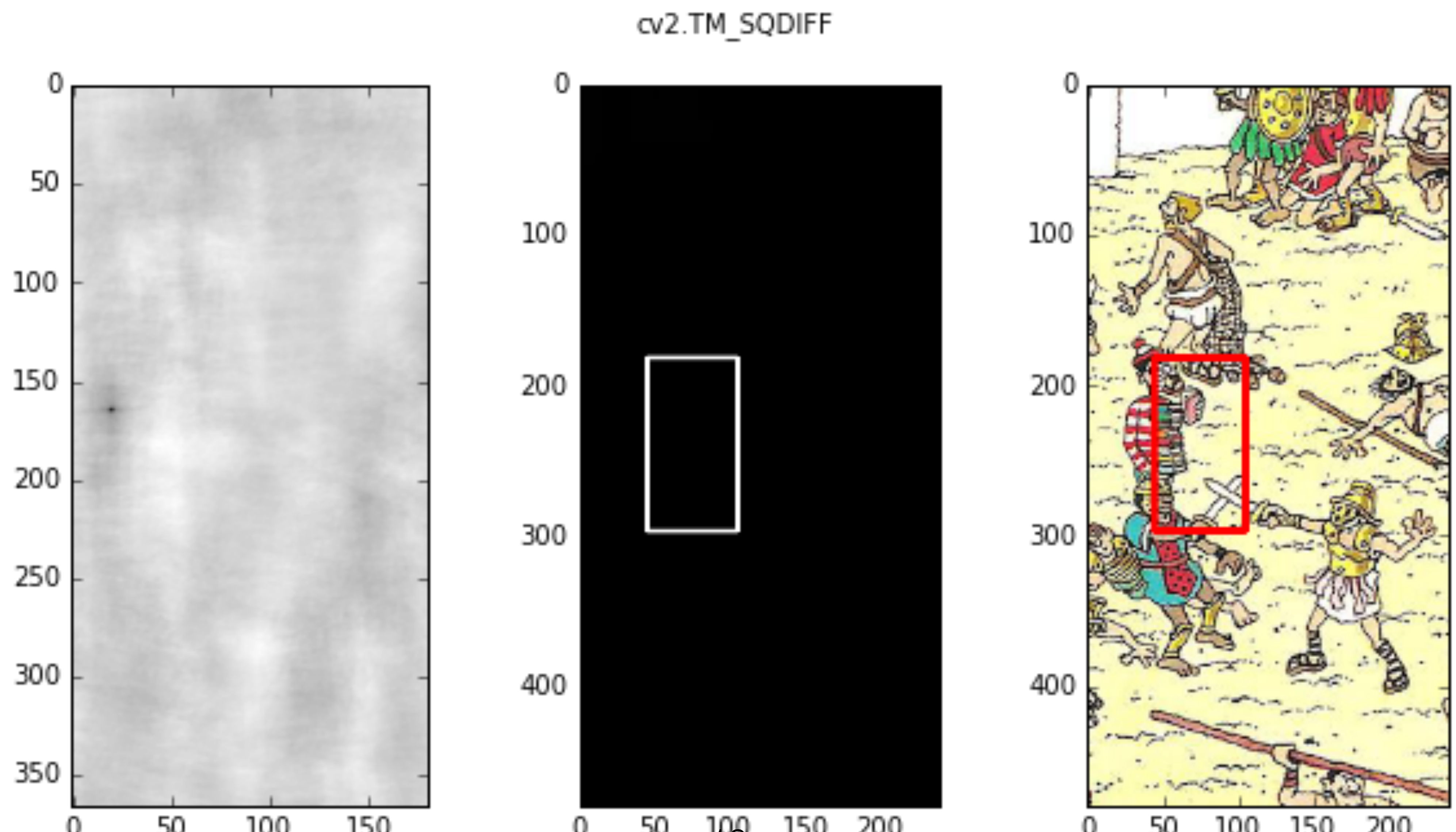
Normalized Cross-Correlation

$$\mathbf{g}(i,j) = \frac{\sum_{k,l}\mathbf{f}(i+k,j+l)\mathbf{h}(k,l)}{\sqrt{\sum_{k,l}\mathbf{f}(i+k,j+l)^2\sum_{k,l}\mathbf{h}(k,l)^2}}$$



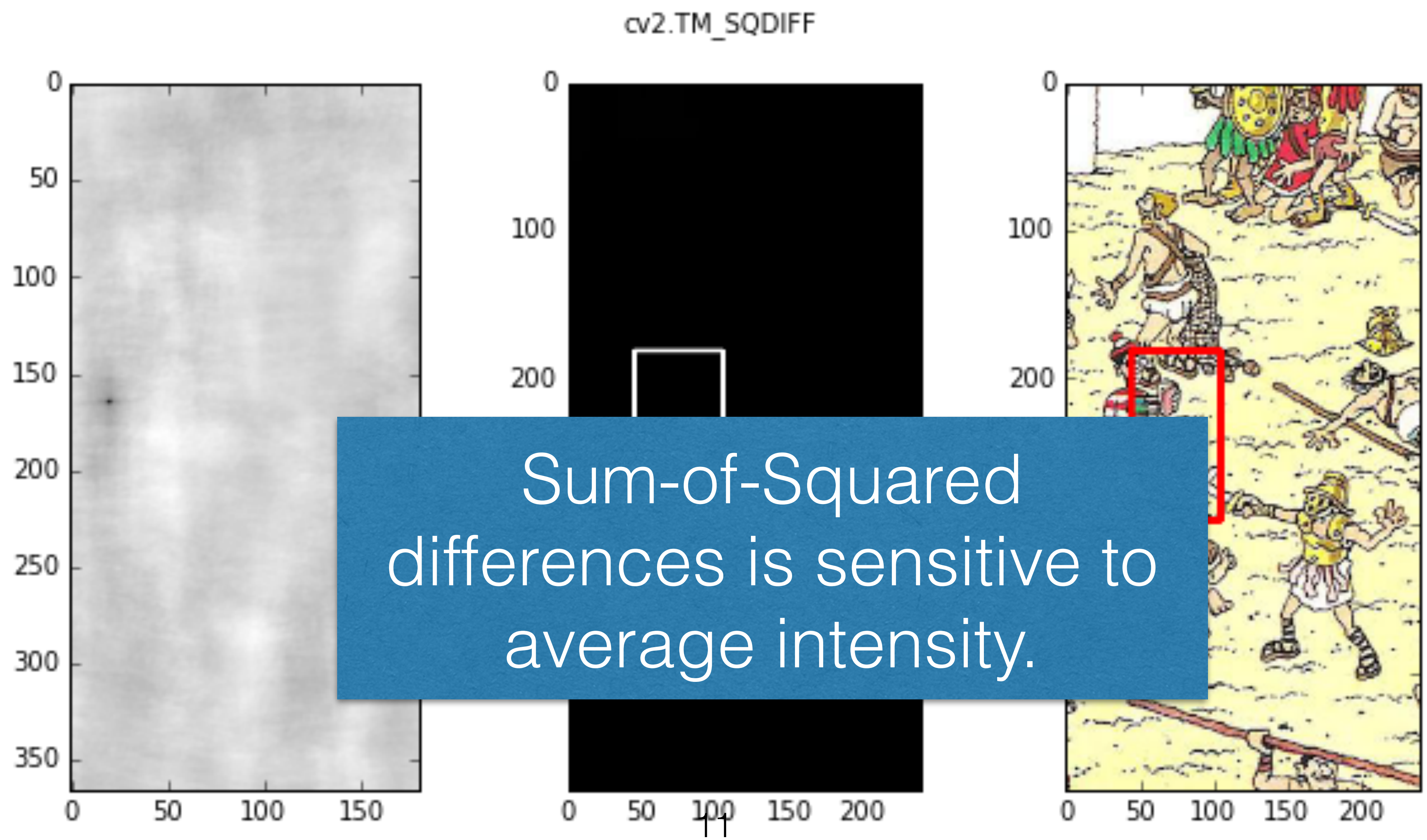
Sum of Squared Differences

$$\mathbf{g}(i,j) = \sum_{k,l} \left(\mathbf{f}(i+k, j+l) - \mathbf{h}(k,l)\right)^2$$



Sum of Squared Differences

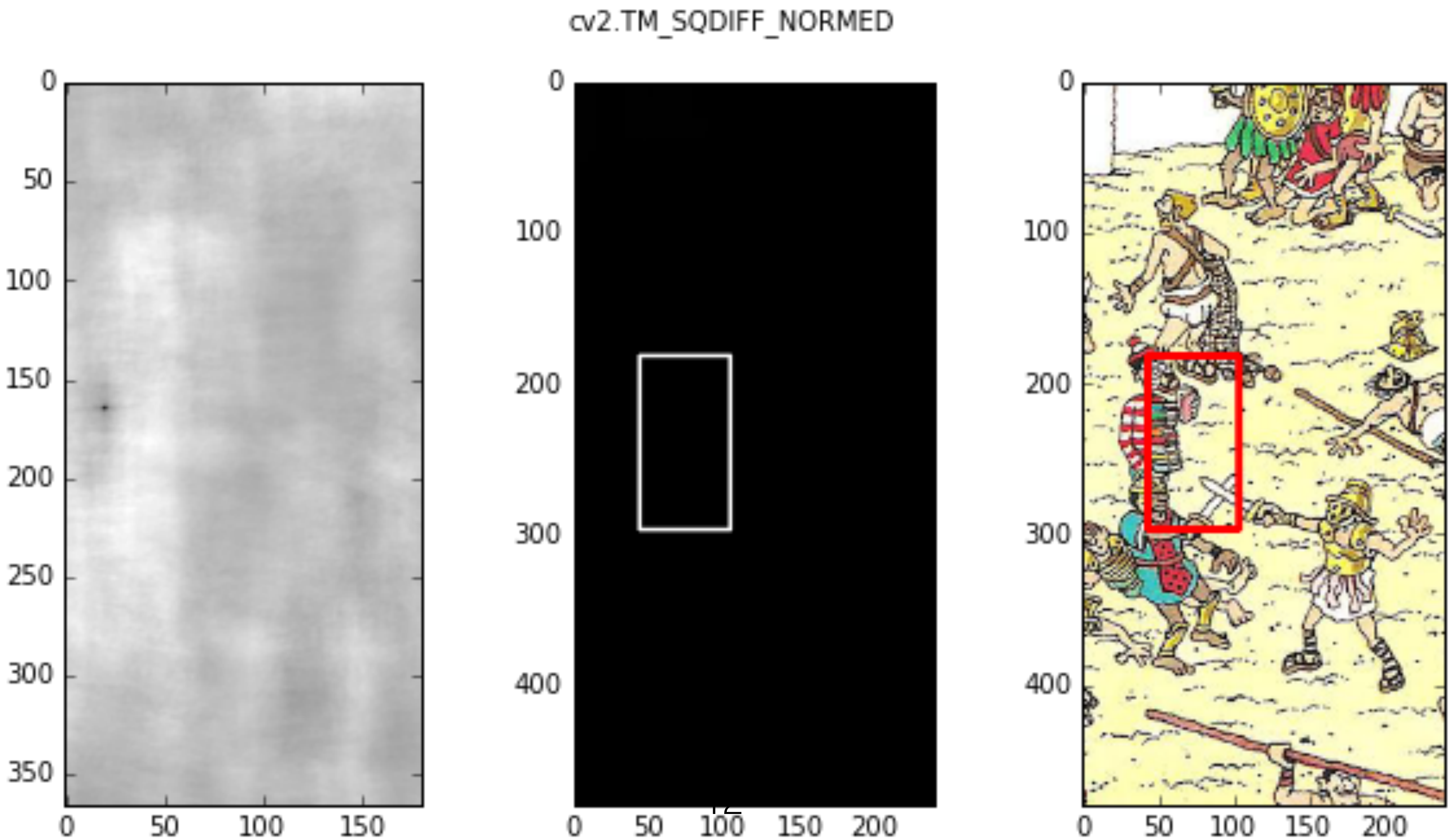
$$\mathbf{g}(i,j) = \sum_{k,l} \left(\mathbf{f}(i+k, j+l) - \mathbf{h}(k,l)\right)^2$$



Sum-of-Squared differences is sensitive to average intensity.

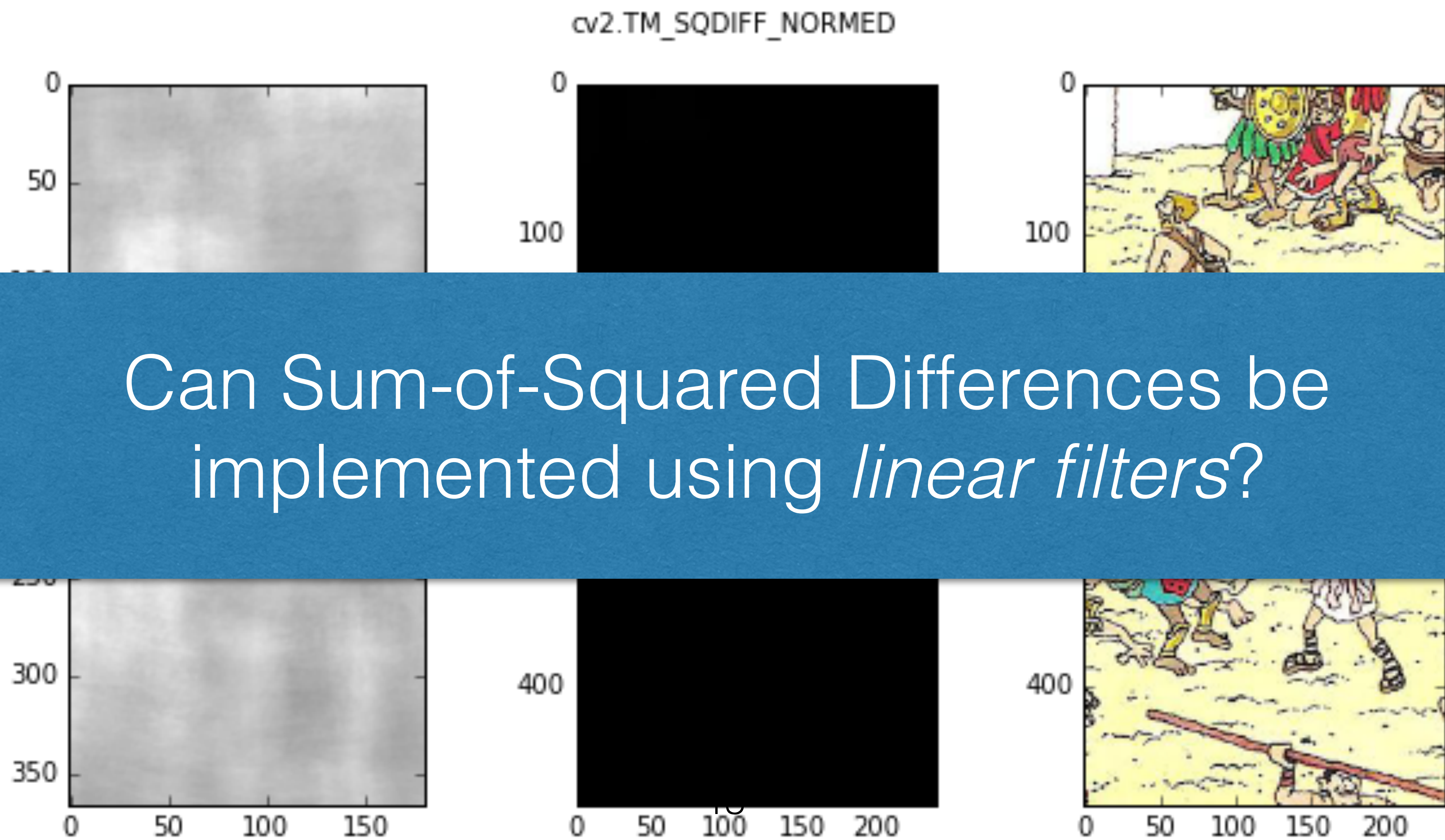
Sum of Squared Differences Normalized

$$\mathbf{g}(i,j) = \frac{\sum_{k,l}\left(\mathbf{f}(i+k,j+l) - \mathbf{h}(k,l)\right)^2}{\sqrt{\sum_{k,l}\mathbf{f}(i+k,j+l)^2 \sum_{k,l}\mathbf{h}(k,l)^2}}$$



Sum of Squared Differences Normalized

$$\mathbf{g}(i,j) = \frac{\sum_{k,l}\left(\mathbf{f}(i+k,j+l) - \mathbf{h}(k,l)\right)^2}{\sqrt{\sum_{k,l}\mathbf{f}(i+k,j+l)^2 \sum_{k,l}\mathbf{h}(k,l)^2}}$$



Can Sum-of-Squared Differences be implemented using *linear filters*?

Correlation Coefficient

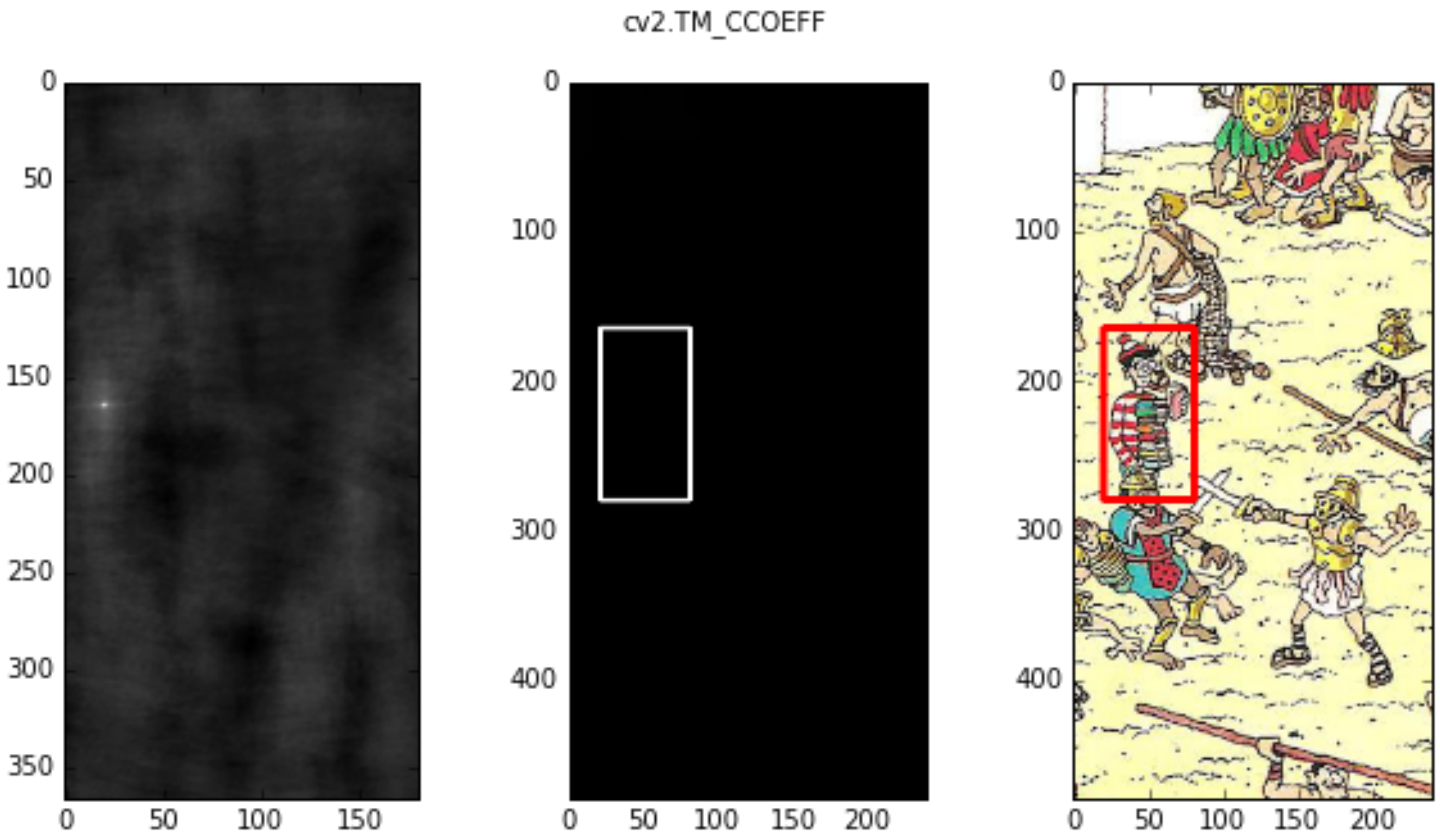
$$\mathbf{g}(i,j) = \sum_{k,l} \mathbf{f}'(i+k, j+l)\mathbf{h}'(k,l)$$

$$\mathbf{h}' = \mathbf{h} - \frac{1}{wh}\sum_{k',l'} \mathbf{h}(k',l')$$

$$\mathbf{f}' = \mathbf{f} - \frac{1}{wh}\sum_{k',l'} \mathbf{f}(i+k', k+l')$$

w and h refer to template $\mathbf{g}$ width and height, respectively

Correlation Coefficient



Correlation Coefficient Normalized

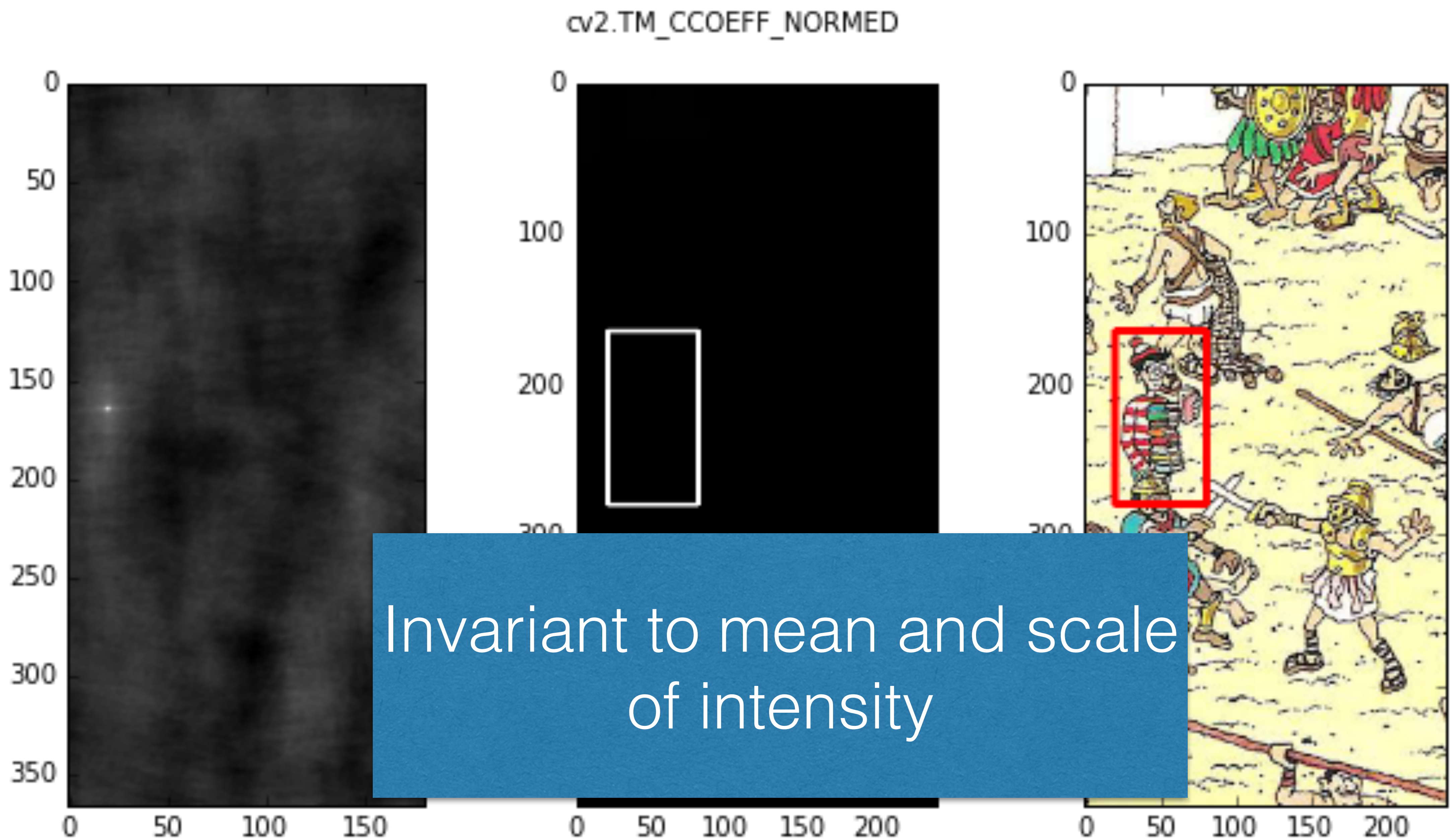
$$\mathbf{g}(i,j) = \frac{\sum_{k,l} \mathbf{f}'(i+k, j+l)\mathbf{h}'(k,l)}{\sqrt{\sum_{k,l} \mathbf{f}'(i+k, j+l)^2 \sum_{k,l} \mathbf{h}'(k,l)^2}}$$

$$\mathbf{h}' = \mathbf{h} - \frac{1}{wh} \sum_{k',l'} \mathbf{h}(k', l')$$

$$\mathbf{f}' = \mathbf{f} - \frac{1}{wh} \sum_{k',l'} \mathbf{f}(i+k', k+l')$$

w and h refer to template $\mathbf{g}$ width and height, respectively

Correlation Coefficient Normalized



Invariant to mean and scale of intensity

Correlation Coefficient

- A measure of the degree to which the movement of two random variables are associated

$$\rho_{xy} = \frac{Cov(x, y)}{\sigma_x \sigma_y}$$

$Cov(x, y)$ is the covariance of variables x and y

σ_x is variance of variable x

σ_y is variance of variable y

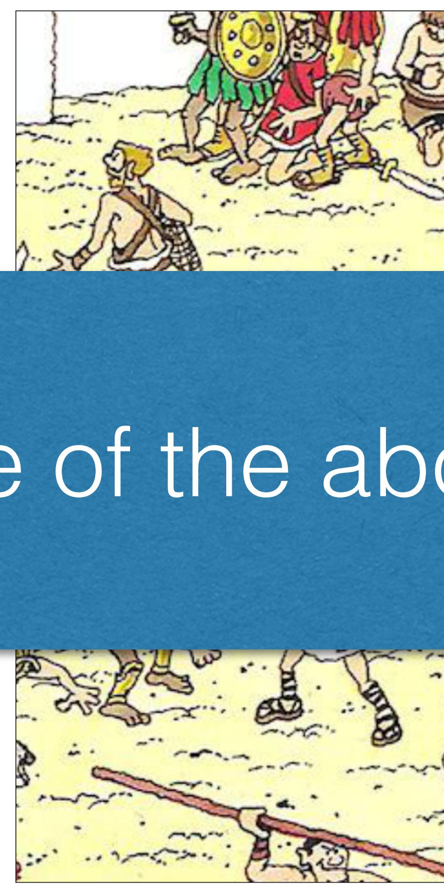
Template Matching

- Which method to use?
    - It depends
- SSD, cross-correlation is faster but sensitive to overall intensity
- Normalized cross-correlation, correlation coefficient is slower, invariant to local average intensity and contrast

Template matching

- What if the size of the template is not the same as it appears in the image?

Template



Waldo appears much smaller than the template in the image

Template

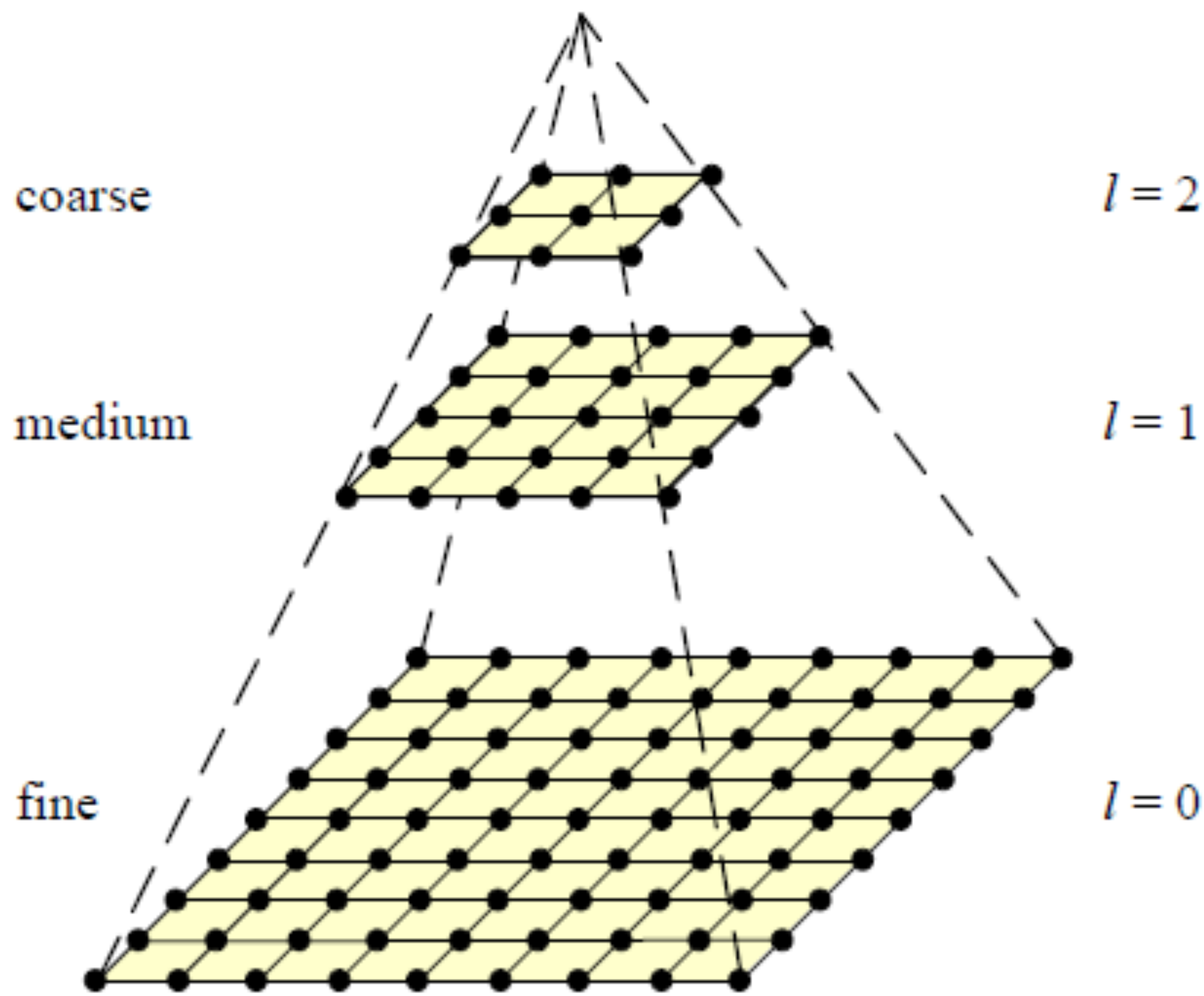
Waldo appears much larger than the template in the image

None of the above methods work

Image Pyramids

Recipe

Blur (Gaussian) and sub-sample by half till you get a single pixel image



[Source: Hoiem]

Image Pyramids

512 256 128 64 32 16 8

Often referred to as Gaussian Image Pyramid

[Source: Forsyth]

Template Matching using Image Pyramids

Recipe

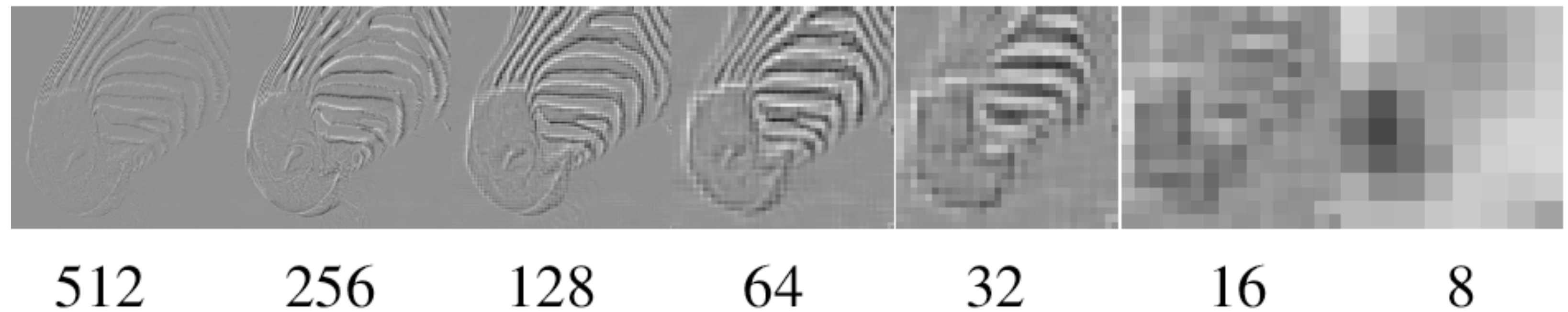
1) Match template at current scale
2) Blur and down sample the image by half
3) Until image is not too small, go to 1
4) Combine filter responses across multiple scales

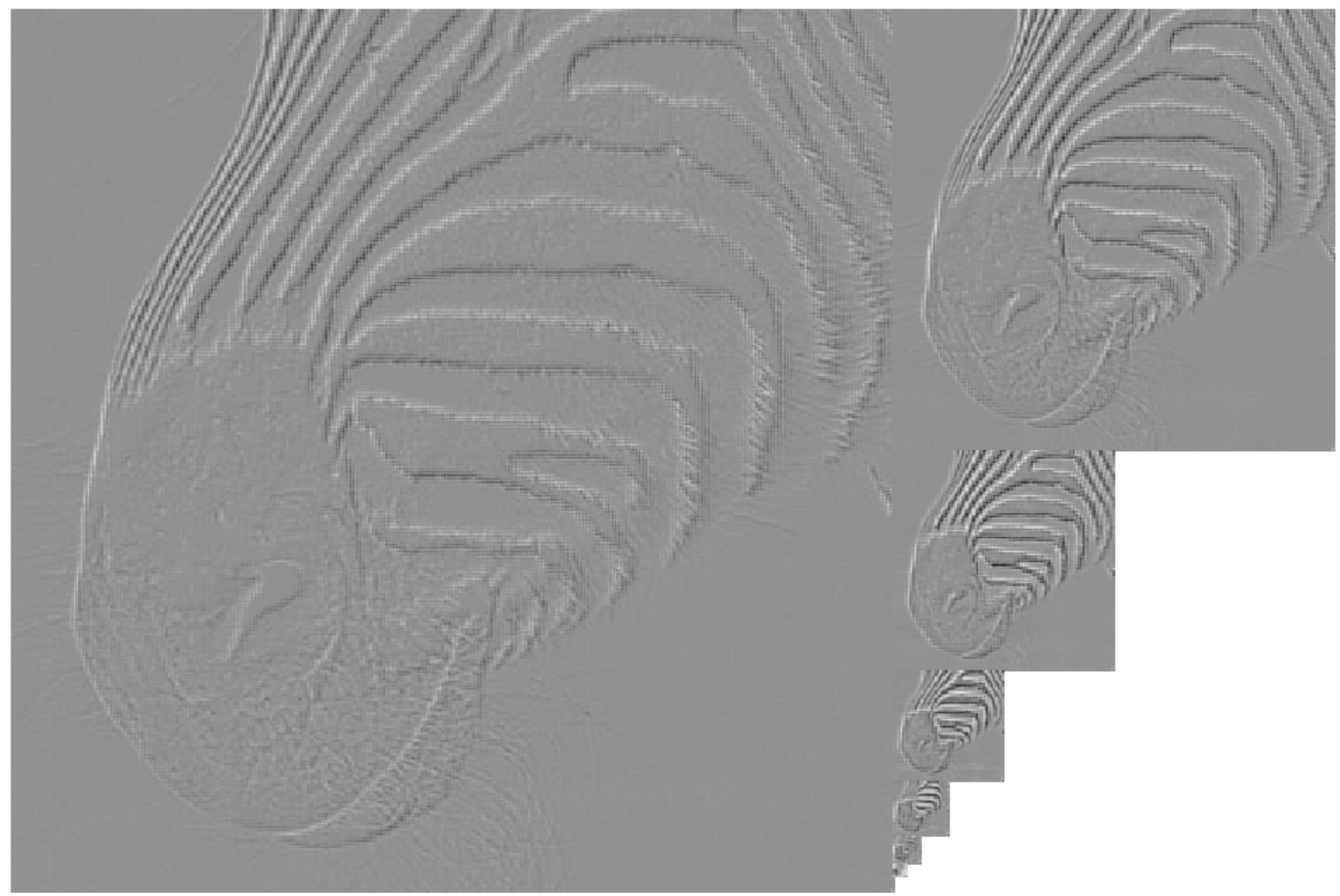
Scale Invariant

Image Pyramids

- Image pyramids are very useful for a number of applications in image processing and computer vision, including:
  - template matching,
  - image registration,
  - image enhancement,
  - interest point detection,
  - object detection, etc.

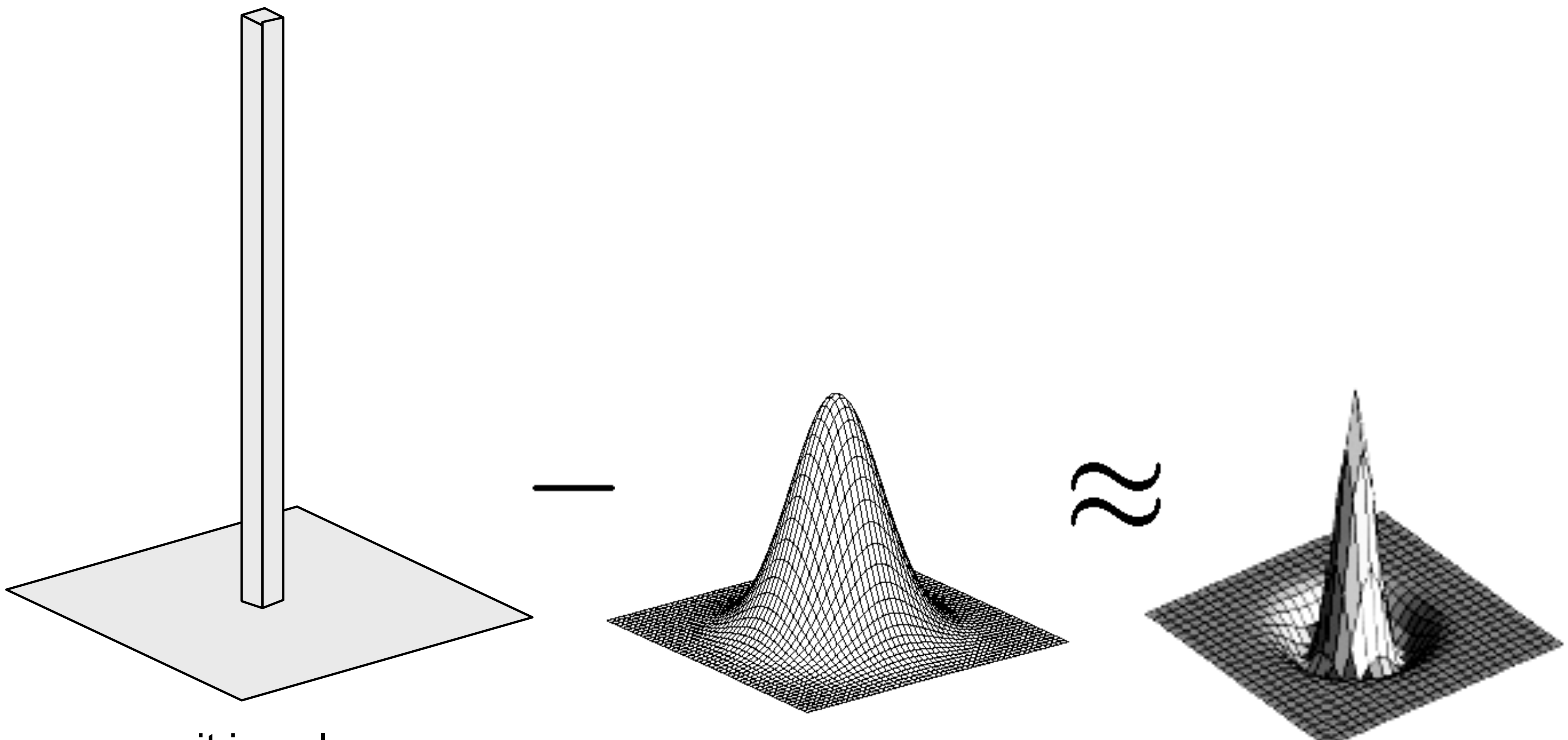
Laplacian Image Pyramid





[Source: Forsyth]

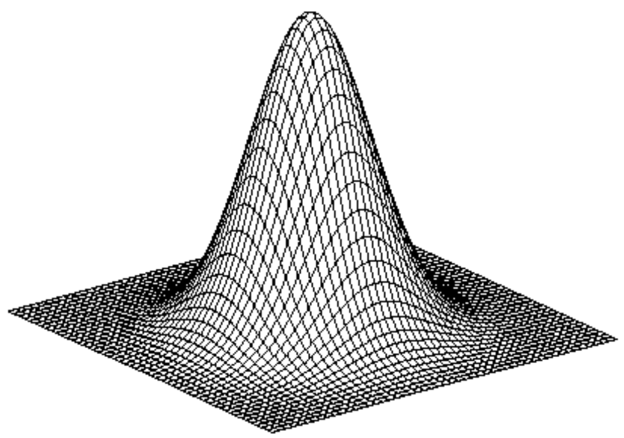
Laplacian Filter



unit impulse Gaussian Laplacian of Gaussian

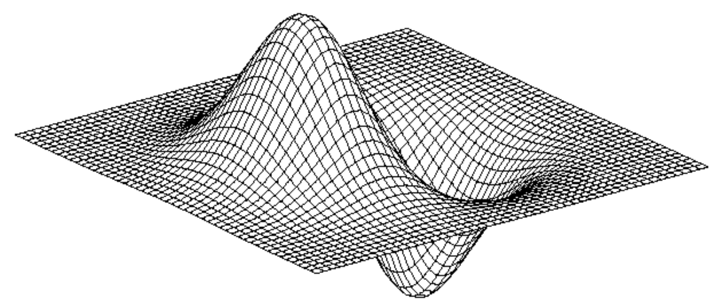
[Source: Lazebnik]

Laplacian Operator



Gaussian

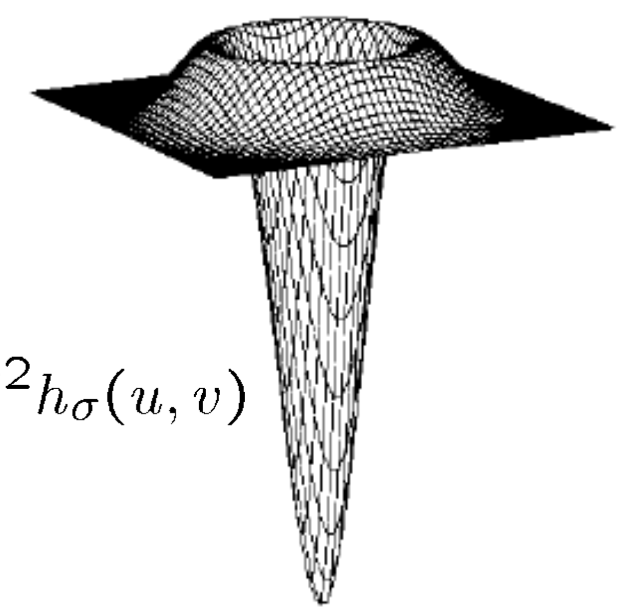
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian



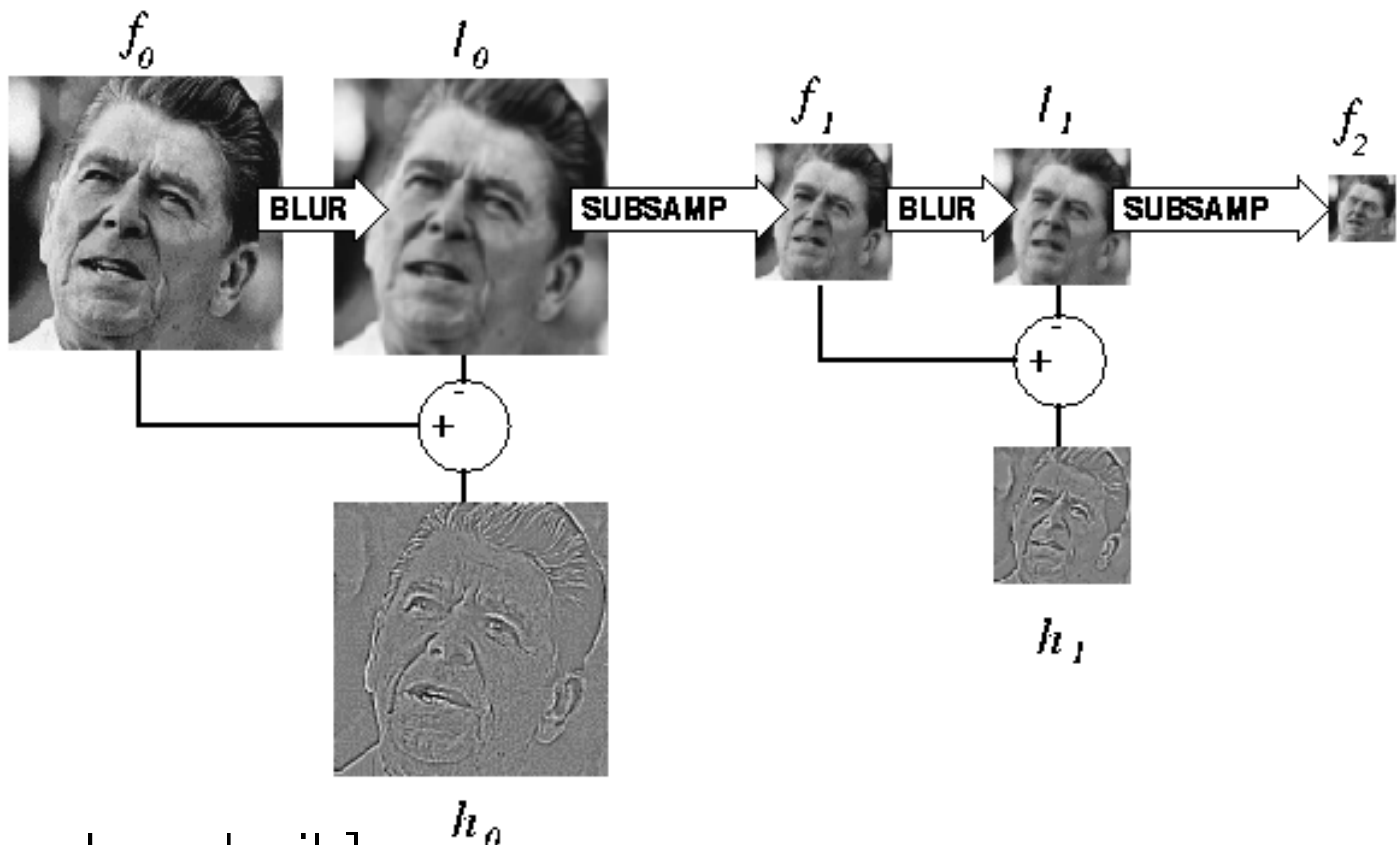
$$\nabla^2 h_\sigma(u,v)$$

∇^2 is the **Laplacian** operator:

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

[Source: Lazebnik]

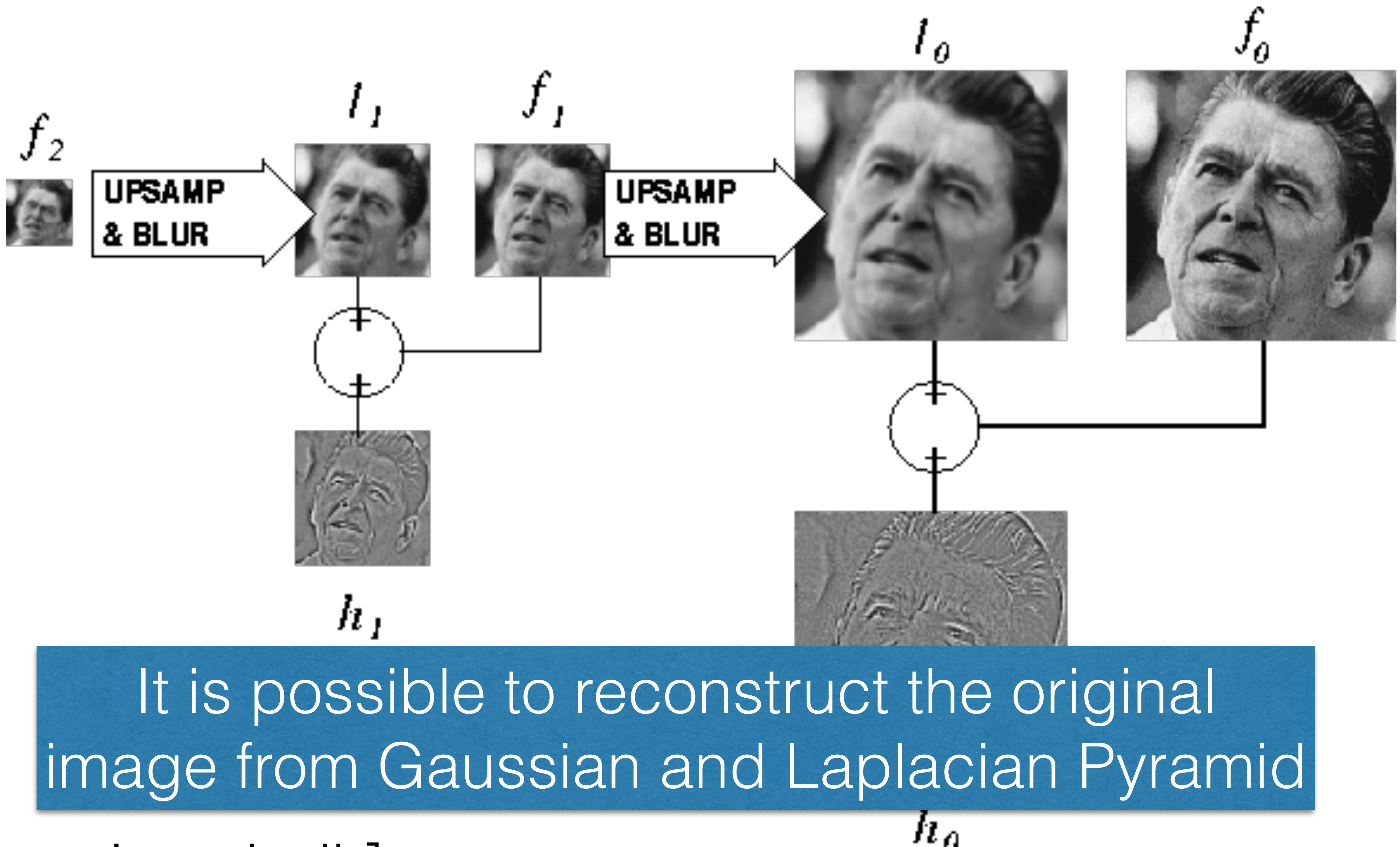
Constructing Gaussian and Laplacian Pyramid



[Source: Lazebnik]

http://sepwww.stanford.edu/~morgan/texturematch/paper_html/node3.html

Reconstructing from Gaussian and Laplacian Pyramid



It is possible to reconstruct the original image from Gaussian and Laplacian Pyramid

[Source: Lazebnik]

http://sepwww.stanford.edu/~morgan/texturematch/paper_html/node3.html

Image Representation

- Pixels: great for spatial processing, poor access to frequency
- Fourier transform: great for frequency analysis, poor spatial info
- Pyramids: trade-off between spatial and frequency information

Summary

- Template matching (SSD or Normxcorr2)
  - SSD cannot be done with linear filters, is sensitive to overall intensity
- Gaussian pyramid
  - Coarse-to-fine search, multi-scale detection
- Laplacian pyramid
  - More compact image representation
  - Can be used for compositing in graphics
- Downsampling
  - Need to sufficiently low-pass before downsampling