CAS CS 585

Image and Video Computing

Lecture by Margrit Betke

Topics: Face Detection, Multi-level Approach, Template Matching (SSD, NCC), Motion Detection, Binary Image Analysis

Finding the Head and its Movement by Detecting Pixels with Skin Color

Finding the Head and its Movement by Detecting Pixels with Skin Color

(a) Input (b) Color

Common Trick in Computer Vision:

Use

"image pyramid" = input image at difference scale.

Here: 6 levels, reduction in x, y by ¼ (other schemes possible)

Then process result pyramid. Why?

Finding Movement by Detecting Pixels with Brightness Changes

(a) Input

(c) Motion

- *DifferenceImage(x,y,t) = Image(x,y,t)-Image(x,y,t-1)*
- *Or DifferenceImage(x,y,t) = Image(x,y,t)-Image(x,y,t-k)*
- *Or DifferenceImage(x,y,t) = |Image(x,y,t)-Image(x,y,t-k)|*

Finding the Face and its Movement by Locating the Best Match of a Face Template

Computer Science

(a) Input

(d) Correlation

Finding the Face and its Movement by Locating the Best Match of a Face Template

Use Jingbin's face as a template

Visualization
of Match Values

Template Matching with Sum-Squared Difference (SSD)

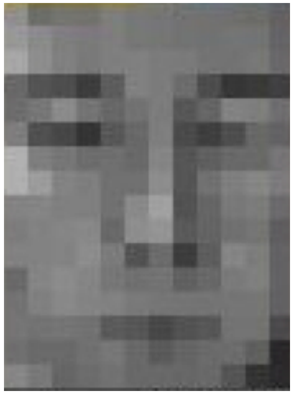
- *Scene subimage s, template image m*
- *same size images = n pixels*
- s_i *= ith pixel in subimage of scene*
- m_i *= ith pixel in template image m*

- $SSD = \Sigma\_(i=1 \text{ to } n) \quad (s_i - m_i)^2$

- *Template matching = exhaustive search algorithm for position of scene subimage that best matches the template (where SSD is smallest)*

Finding the Face and its Movement by Locating the Best Match of a Face Template

Used average face as a template



Visualization
of Match Values

Template Matching via with Normalized Correlation

Computer Science

- Scene subimage *s*, template *image m*
- *Normalized correlation coefficient*

  $r = 1/n\ \Sigma_i\ ((s_i - mean(s)) * (m_i - mean(m)) / (\sigma_s\ \sigma_m))$ where

  s_i and m_i are respective brightness values of the *i*th pixel

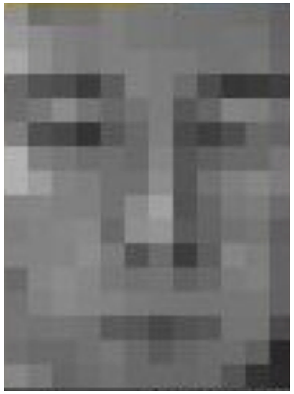
  mean(m) and σ_m are mean and standard deviation of all pixels in the template

  mean(s) and σ_s are mean and standard deviation of all pixels in the subimage of the scene
- Template matching = exhaustive search for position of subimage that produces highest *r*
- *r* can be between -1 and 1

Finding the Face and its Movement by Locating the Best Match of a Face Template

Computer Science

Used average face as a template



Visualization of Normalized Correlation Coefficient Match Values

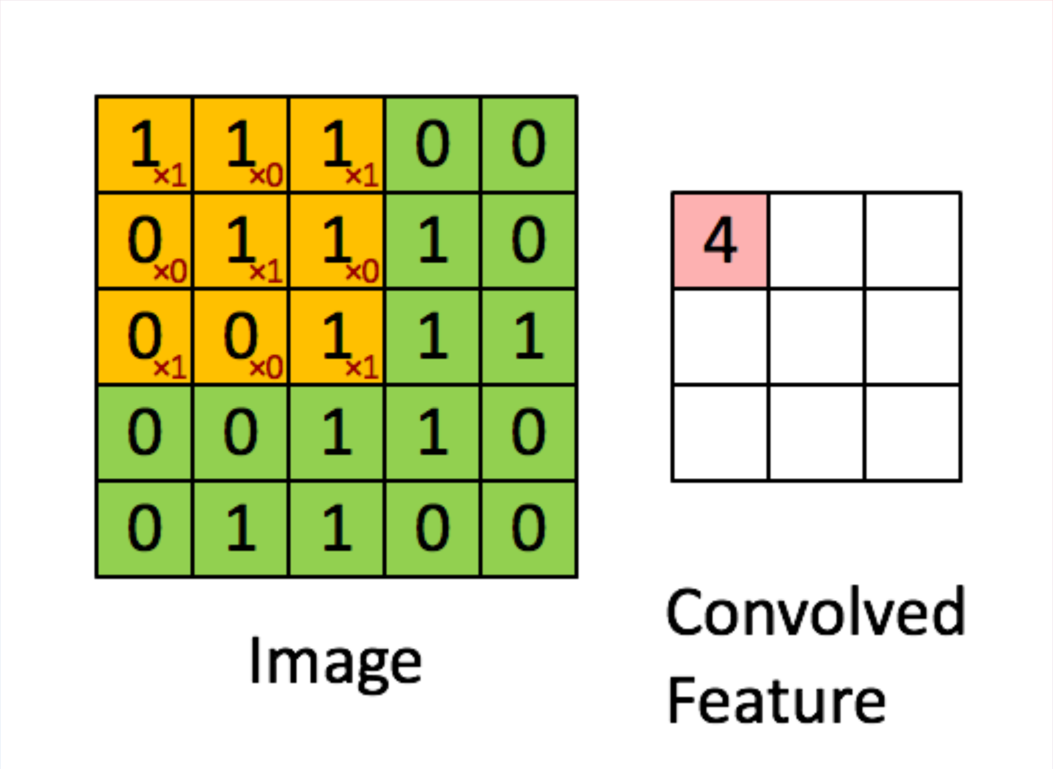
1D Discrete Convolution

Computer Science

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]\, g[n-m]$$
$$= \sum_{m=-\infty}^{\infty} f[n-m]\, g[m].$$

2D generalization:
f = input image, g = template image
(or CNN function)

2D Convolution Example



2D Convolution Example

0	0	0	0	0	0	0
0	1	1	1	1	0	0
0	1	0	0	1	0	0
0	1	1	1	1	0	0
0	1	0	0	1	0	0
0	1	1	1	1	0	0
0	0	0	0	0	0	0

Input: black and white image

1	3	1
3	2	3
1	3	1

Weight matrix

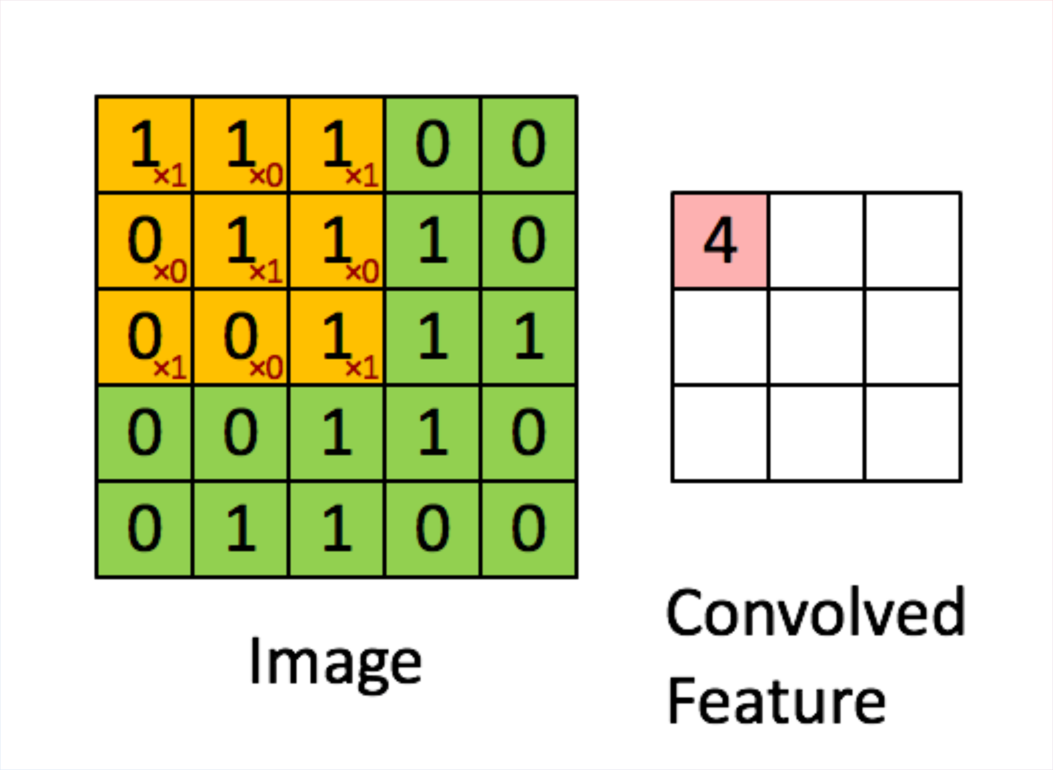
0	0x1	0x3	0x1	0	0	0
0	1x3	1x2	1x3	1	0	0
0	1x1	0x3	0x1	1	0	0
0	1	1	1	1	0	0
0	1	0	0	1	0	0
0	1	1	1	1	0	0
0	0	0	0	0	0	0

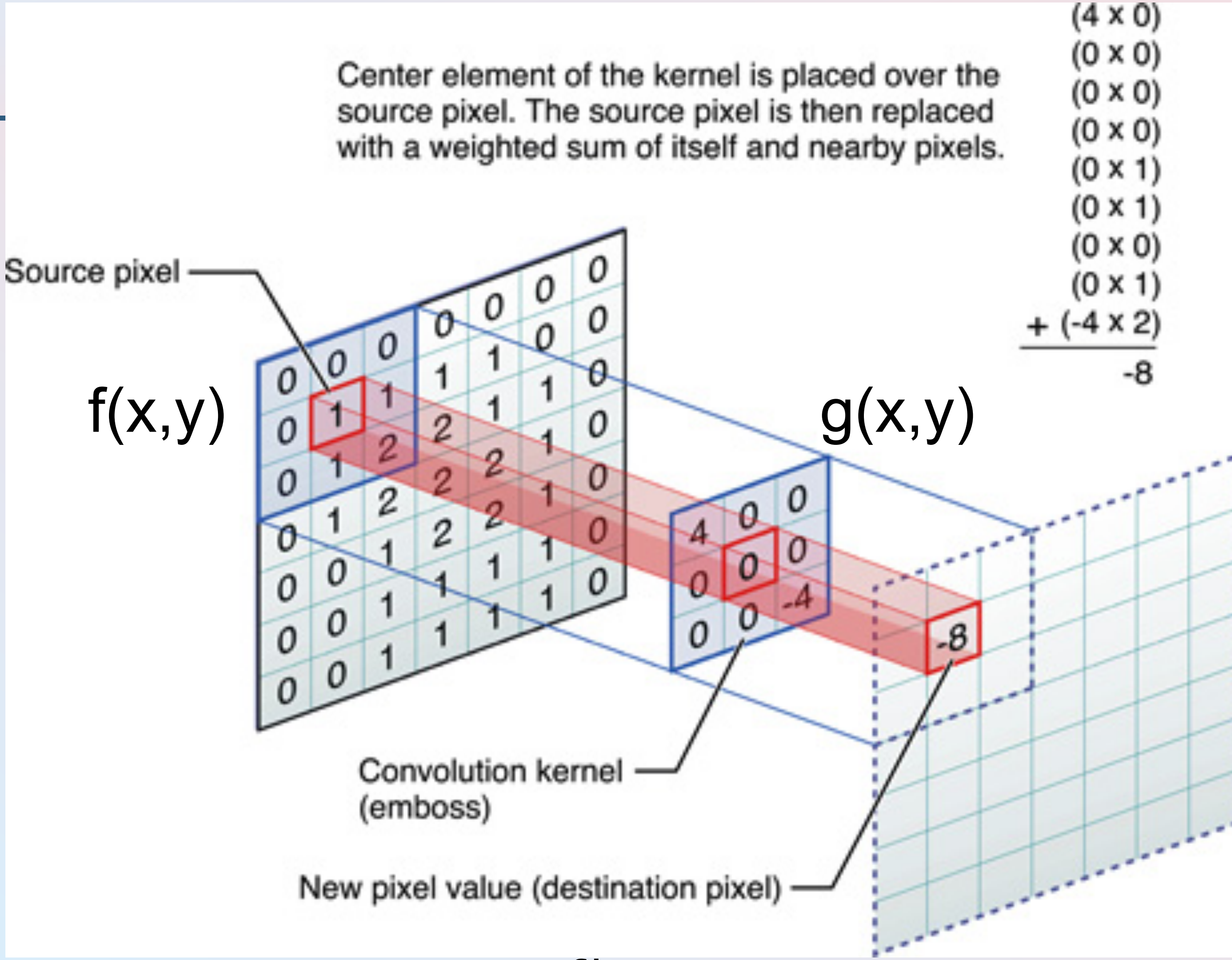
Applying weight matrix

8	9			

Output: convoluted feature

Convolution example




Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
f(x,y)
g(x,y)
Convolution kernel (emboss)
New pixel value (destination pixel)

f*g

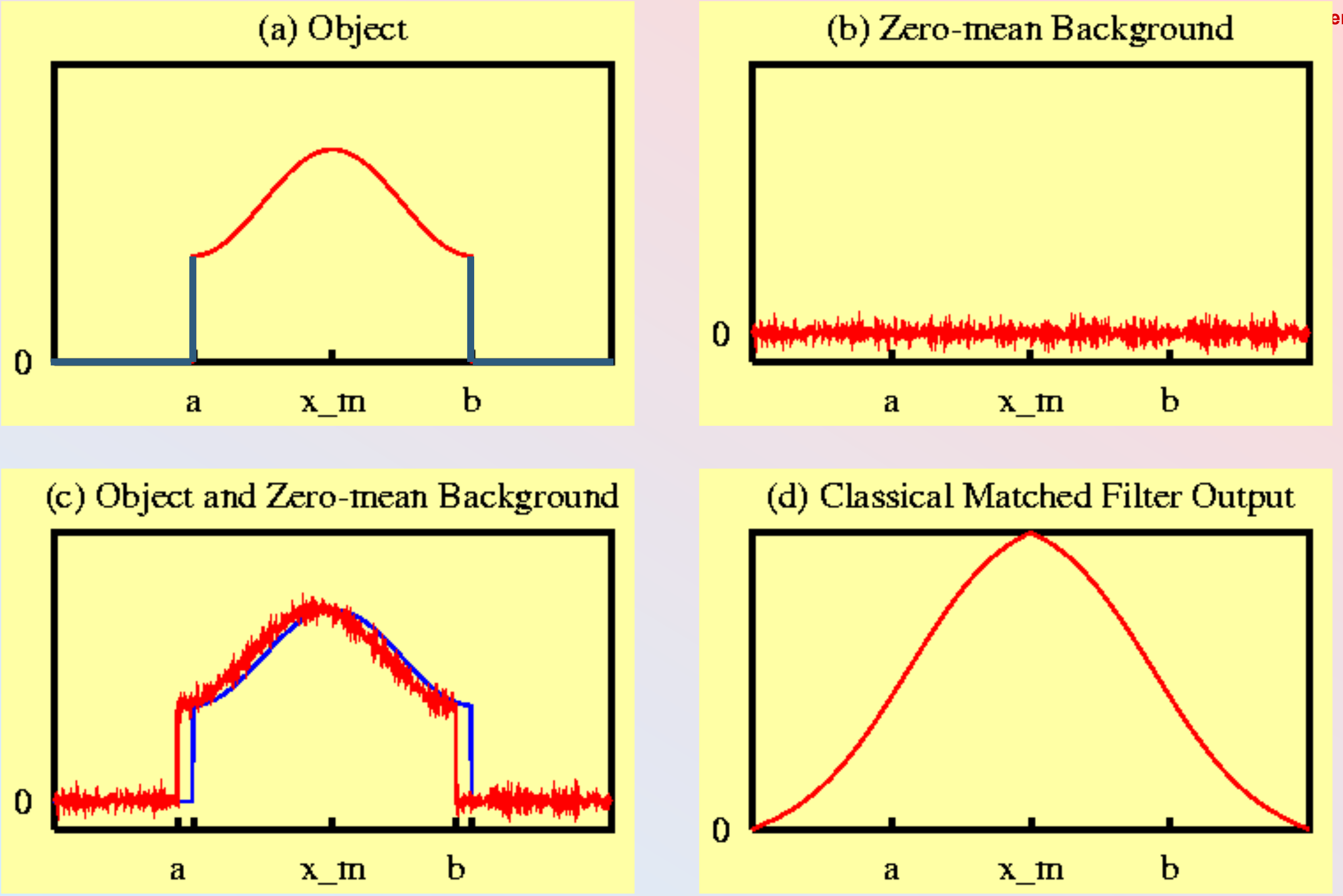
Why is Convolution Powerful?

Signal Processing:

Convolution is used to define a "matched filter" for locating "targets" in time signals

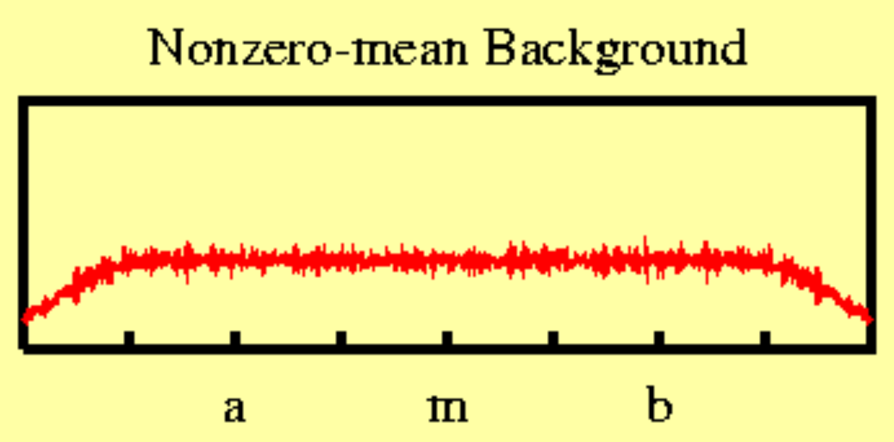
(optimal if Gaussian noise)

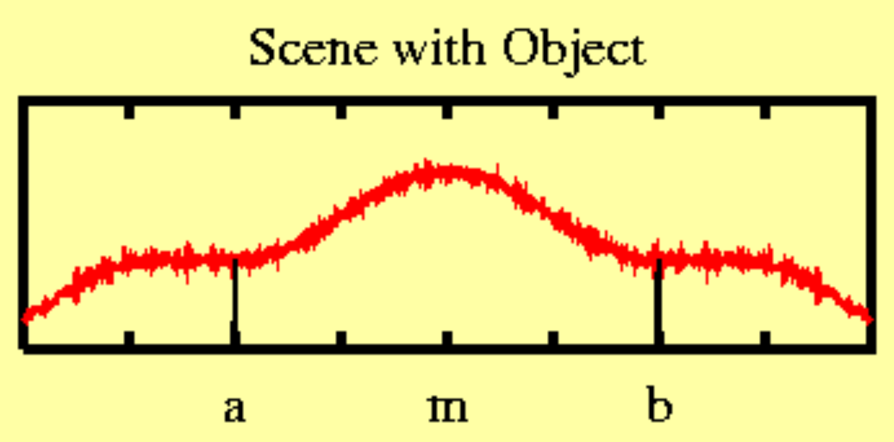
1D Position Estimation: Σ object*background

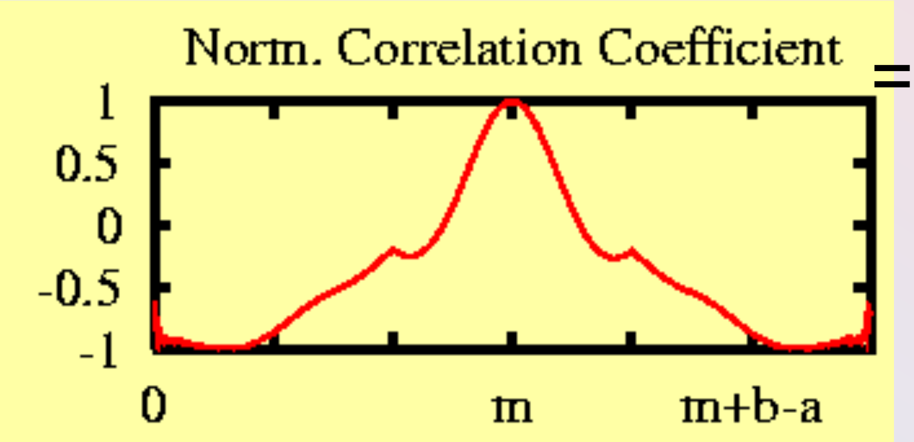


Another 1D convolution example:

Computer Science







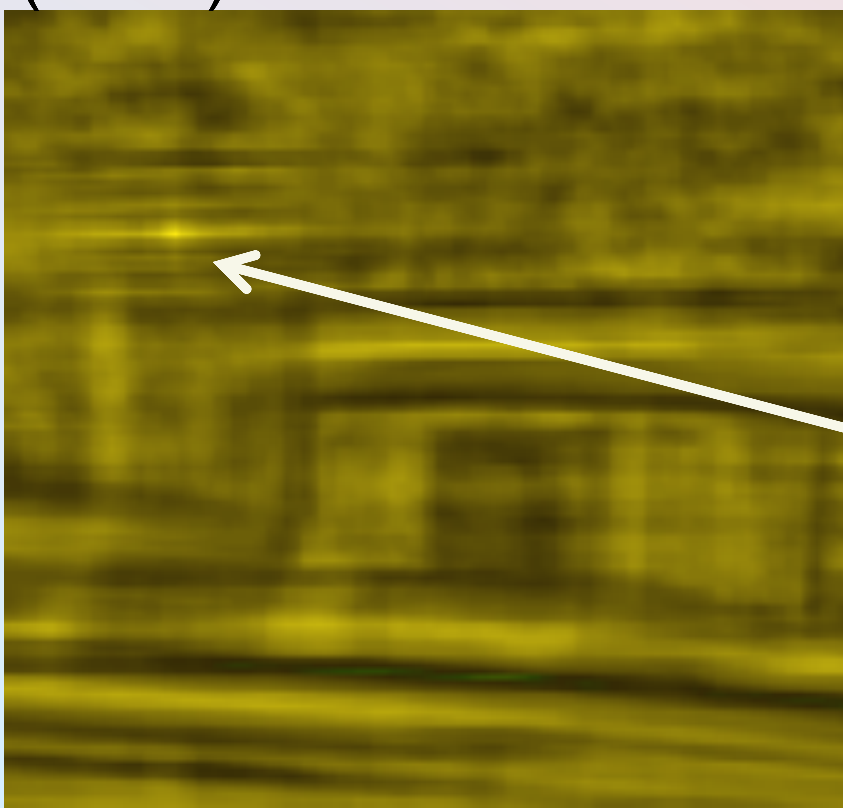
= convolution/st

2D Position Estimation

Convolution of one-way sign with itself 

2 D Position Estimation

Convolution of one-way sign with scene (NCC)



Peak in performance surface (= negative loss fct) at correct location

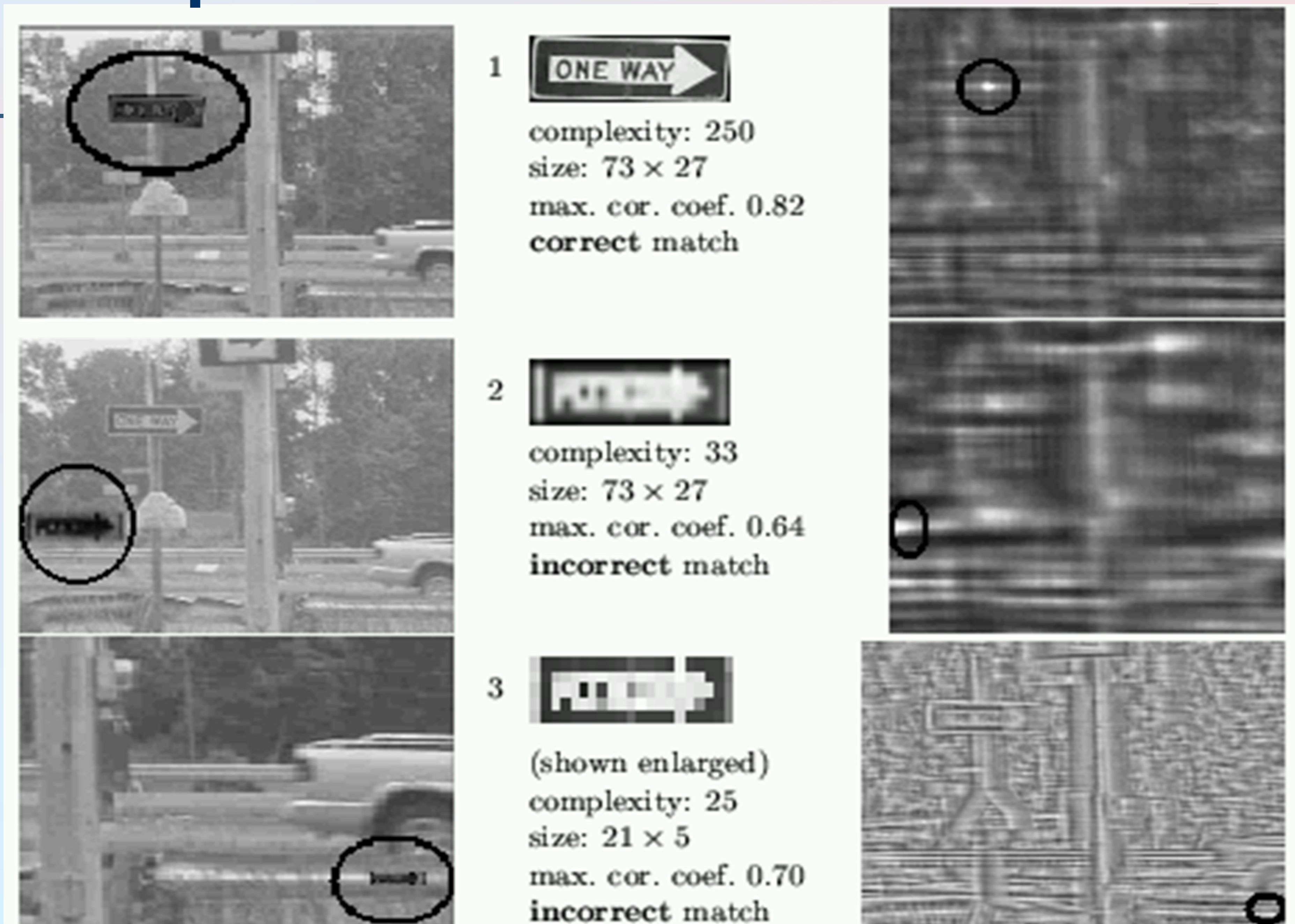
2 D Position Estimation

Convolution of one-way sign with scene (NCC)

This performance surface is computed for correct size of one-way sign

Different surfaces for different sizes of object

Sample Performance Surfaces



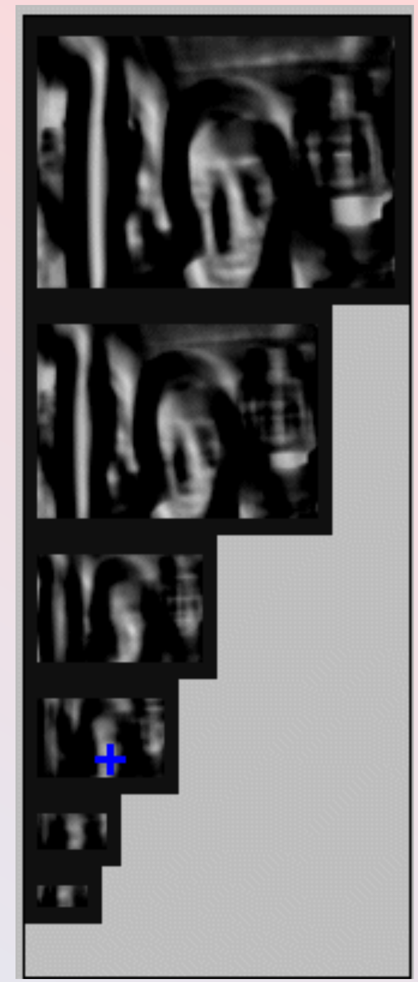
Multi-Resolution Matching

Computer Science

Normalized correlation coefficient over multi-resolution search space:

$$r = \frac{1}{n} \frac{\sum_i (s_i - \mathrm{mean}(s))\,(m_i - \mathrm{mean}(m))}{(\sigma_s\, \sigma_m)}$$

← Template matched over all resolutions →



Finding the Face and its Movement by Locating the Best Match of a Face Template

Computer Science

(a) Input

You can apply template matching to a small version of your input image and use that search result to start searching for a match in the 2nd smallest images. Repeat until the original size is processed.

(d) Correlation

Multi-scale Pyramids

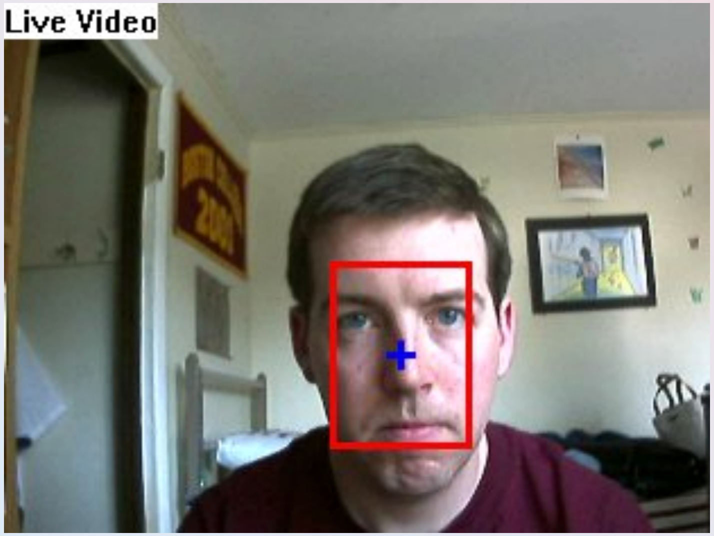
Computer Science

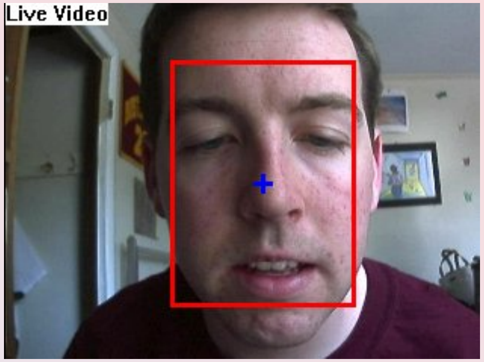
Combine the results of color and motion detection to mask the regions of interest for correlation-based template matching

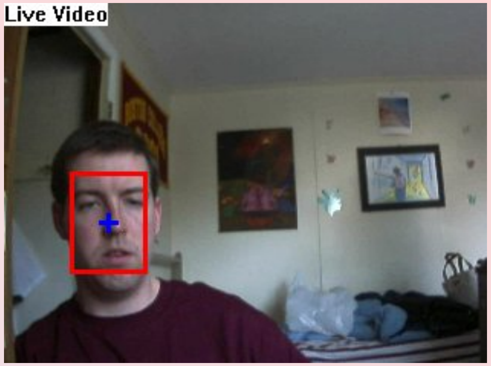
(a) Input (b) Color (c) Motion (low-pass filter (d) Correlation (e) Masked Correlation

Face Detection

Data Variability



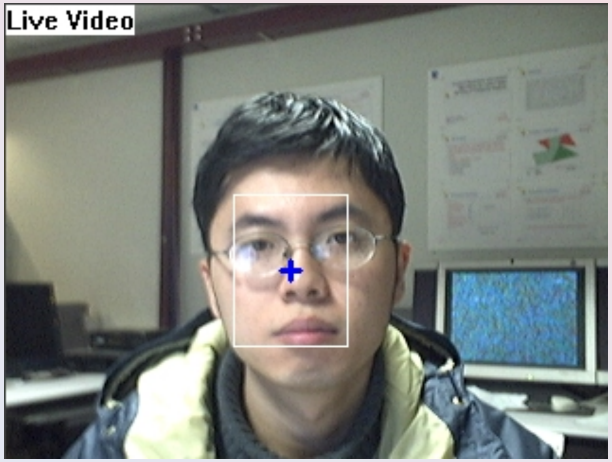




Large Face

Small Face

Shadows
Cluttered background



Algorithm: Multi-scale Face Detection

Video Input

⇓

Color & Motion Analysis

⇓

Mask Input Pyramids

⇓

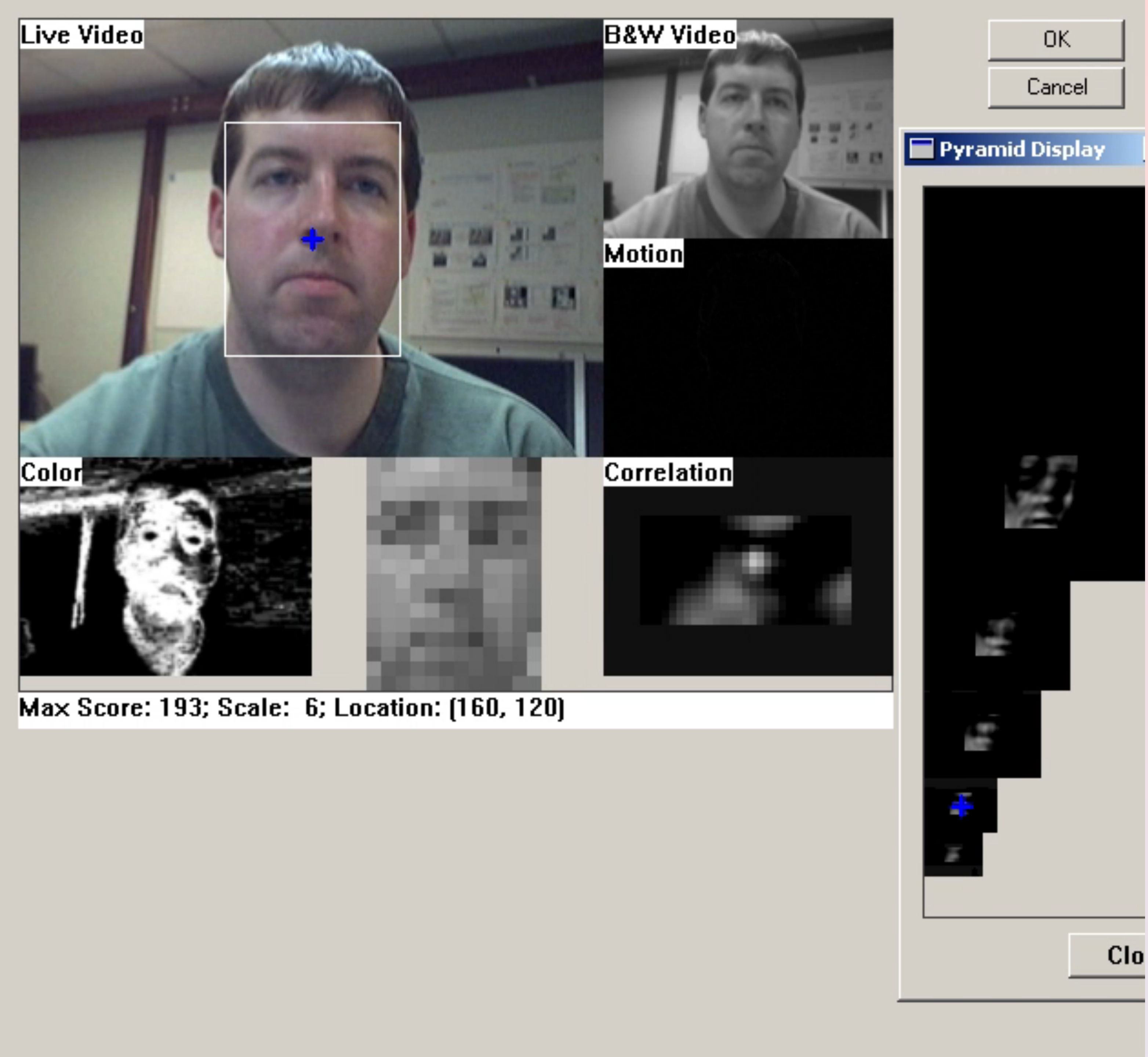
Face Tracking: Template Correlation Over Pyramid

⇓

Output: Face Location and Scale

Face Detection Interface

Computer Science


Live Video
B&W Video
Motion
Color
Correlation
Max Score: 193; Scale: 6; Location: (160, 120)
OK
Cancel
Pyramid Display
Clo

Finding Movement in Video by Comparing to a Somewhat Static Background

A) Establish background model:

Easy -- use first frame:

BackgroundImage(x,y,t) = Image(x,y,0)

Better: Dynamic Gaussian models of brightness changes. Filter

if $k\,\sigma(x,y,t) < |\,Image(x,y,t) - \mu(x,y,t)|$

for k = 5%; mean μ and std. dev. σ updated in some time window

B) Compute difference to background:

DifferenceImage(x,y,t)=Image(x,y,t)-BackgroundImage(x,y,t)

Deal with negative values, e.g., use absolute difference or remap the range of values to 0-255.

C) Interpret difference image

Geometric Properties of Detected Object

Computer Science

- Binary Image $B(x,y) = 1$ if $I(x,y) > \tau$,

  $= 0$ otherwise

- Size, area: $A = \sum_{x=0}^{x=xdim-1} \sum_{y=0}^{y=ydim-1} B(x,y)$

  $A = M_{00}$ "0th moment"

- Position: use 1st moment
- Orientation: use 2nd moments