Target Detection: Find all "T"s

Image

Model of "Target"

Target Detection in spatial domain: Correlation Filter

Template matching using squared Euclidean Distance

$$d^2_{f,t}(u,v) = \sum_{x,y}[f(x,y) - t(x-u,y-v)]^2$$

f(x,y): Image
t(x,y): template
Summation over template window

Expanding...

$$d^2_{f,t}(u,v) = \sum_{x,y}[f^2(x,y) - 2f(x,y)t(x-u,y-v) + t^2(x-u,y-v)]$$

constant — constant

Cross-correlation formula:

$$c(u,v) = \sum_{x,y} f(x,y)t(x-u,y-v)$$

Lewis, J. P. "Fast normalized cross-correlation." *Vision interface*, 1995

Target Detection by Convolution

Object/Target" Detection by Convolution

Convolution in spatial domain is the same as element-wise multiplication in frequency domain

(Gonzalez 3rd Edition, 4.2.5 Convolution, Page 232)

"Object" Detection: Finding Faces

Image

CVRR

Thanks, Kevan Yuen

Target Detection by convolution in frequency domain

Image

Template 

(enlarged for viewing purposes only)

CVRR

Thanks, Kevan Yuen

Target Detection by Convolution

Convolution in spatial domain is the same as element-wise multiplication in frequency domain
(Gonzalez 3rd Edition, 4.2.5 Convolution, Page 232)

$$f(t) \star h(t) \Leftrightarrow H(\mu) F(\mu)$$

Pad image with zeros to size [M_I + M_T – 1, N_I + N_T – 1]

Thanks, Kevan Yuen

Target Detection by Convolution

Pad image with zeros to size [M_I + M_T – 1, N_I + N_T – 1]

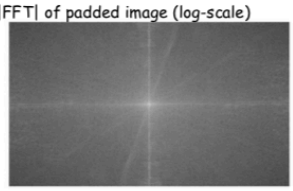
(Note: Not recommended to print this slide)

CVRR

Thanks, Kevan Yuen

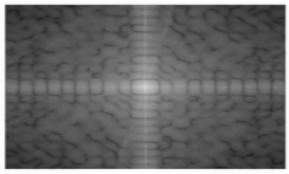
Target Detection by Convolution

|FFT| of padded image (log-scale)

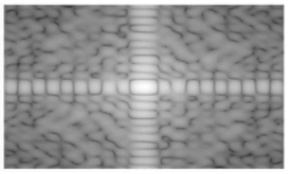


Take Fourier transforms of images then multiply

The result of element-wise multiplying the two FFT images together



|FFT| of padded template (log-scale)



Thanks, Kevan Yuen

Note: Absolute value and log-scale is done only for viewing purposes

Target Detection by Convolution

Taking the inverse Fourier transform back into spatial domain is the resulting cross-correlated image using convolution in frequency domain:

But this doesn't highlight the face that well!

Thanks, Kevan Yuen

Target Detection by Convolution

Use normalized cross correlation:

$$\gamma(u,v) = \frac{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}][t(x-u,y-v) - \bar{t}]}{\left\{\sum_{x,y}[f(x,y) - \bar{f}_{u,v}]^2 \sum_{x,y}[t(x-u,y-v) - \bar{t}]^2\right\}^{0.5}} \quad (2)$$

where $\bar{t}$ is the mean of the feature and $\bar{f}_{u,v}$ is the mean of $f(x,y)$ in the region under the feature. We refer to (2) as *normalized cross-correlation*.

The idea is to normalize the cross correlation within each window.

An implementation of this is done in MATLAB's xnormcorr2 which still uses the cross-correlated image for the formula above. See paper or 'open normxcorr2' in MATLAB for details on how this is done.

Thanks, Kevan Yuen

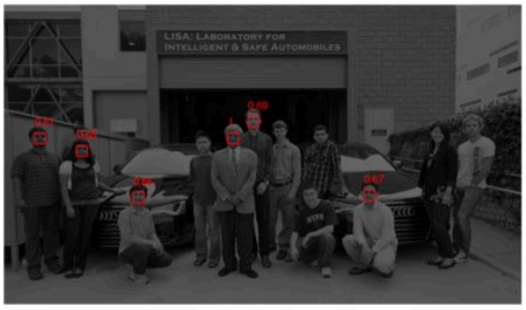
Lewis, J. P. "Fast normalized cross-correlation." *Vision interface*, 1995

Target Detection by Convolution

The resulting image after normalized cross correlation:

Thanks, Kevan Yuen

Target Detection by Convolution



Detected Targets by matching templates with NCC scores > 0.5

CVRR

Thanks, Kevan Yuen

Target Detection by Convolution

Detected Targets by matching templates with NCC scores > 0.3

CVRR

Thanks, Kevan Yuen