<u>Matching</u>

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} = \Sigma$$

population

$$\begin{bmatrix} S_x^2 & \underline{S_{xy}} \\ S_{xy} & S_y^2 \end{bmatrix}$$ sample

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \quad -1 \rightarrow +1$$

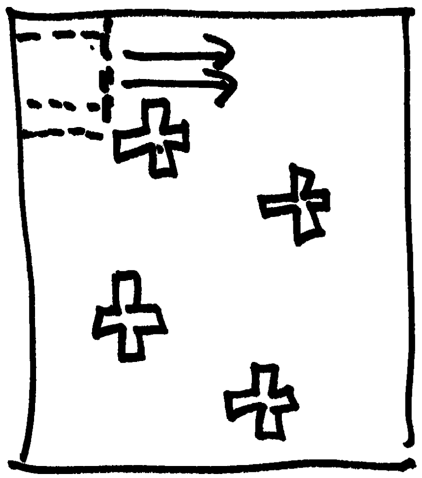
$$r_{xy} = \frac{S_{xy}}{S_x S_y}$$

$$\left.\begin{aligned} S_{xy} &= \frac{1}{N-1} \sum (x_i - \bar{x})(y_i - \bar{y}) \\ S_x &= \sqrt{\frac{\sum (x_i - \bar{x})^2}{N-1}} \\ S_y &= \sqrt{\frac{\sum (y_i - \bar{y})^2}{N-1}} \end{aligned}\right\}$$

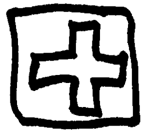
$$\boxed{r_{xy} = \frac{\sum (x_i-\bar{x})(y_i-\bar{y})}{\left[\sum (x_i-\bar{x})^2 \cdot \sum (y_i-\bar{y})^2\right]^{1/2}}}$$

normalized cross correlation statistic (sampled)

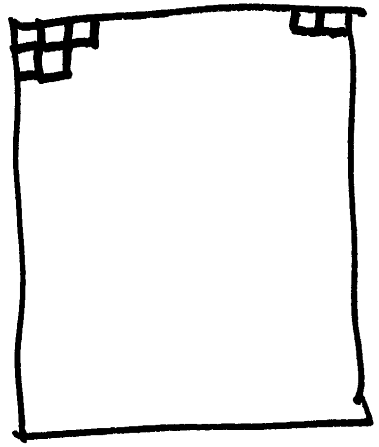
X: data set #1, Y: data set #2



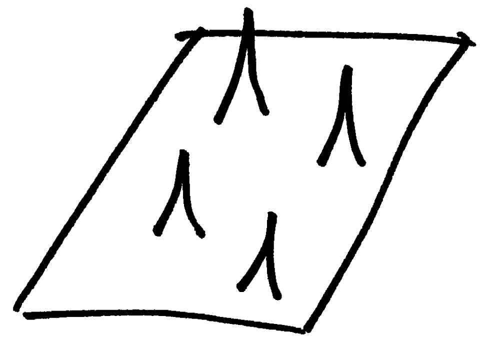
image



template



response map

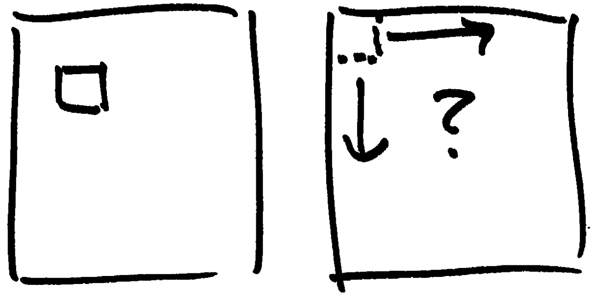


look through response map

look for locations exceed threshold

<u>template matching</u>

image matching



calculated cross correlation in <u>space</u> <u>domain</u>

Fourier Theorem : you can decompose a function into a weighted sum of <u>sines</u> + <u>cosines</u> of increasing frequency

euler's equation : $\cos\theta + i\sin\theta = e^{i\theta}$ (complex exponential)

$\cos\omega t + i\sin\omega t = e^{i\omega t}$

$\cos 2\pi f t + i\sin 2\pi f t = e^{i2\pi f t}$

$\cos 2\pi f x + i\sin 2\pi f x = e^{i2\pi f x}$

euler : same guy referred to in "euler angles"

ω : radian frequency , f : freq, hz, cycles/second

f : freq cycles/meter

some use "i" some use "j" for $\sqrt{-1}$

for discrete $\left.\begin{array}{l}\text{DFT}\\\text{IDFT}\end{array}\right\}$ give relation between function values + weights

DFT $\quad F(\mu) = \dfrac{1}{N} \displaystyle\sum_{x=0}^{N-1} f(x)\, e^{-i 2\pi x \mu / N}$ discrete fourier transform

IDFT $\quad f(x) = \displaystyle\sum_{\mu=0}^{N-1} F(\mu)\, e^{+i 2\pi x \mu / N}$ inverse discrete f.......

$F(\mu)$: weights applied to sines + cosines

$f(x)$: function

$F(\mu)$: describes function in freq. domain

$f(x)$: " " " space domain

1D, also works in 2D

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y)\, e^{-i2\pi(ux/N + vy/M)}$$ DFT, 2D

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u,v)\, e^{+i2\pi(ux/N + vy/M)}$$ IDFT, 2D

in practice DFT's by FFT, # elements of size 2^n

2D FFT's are separable + computed quickly using 1D FFT's

FFT = fast fourier transform
it is possible to state DFT as a matrix-vector product, FFT comes from a clever factorization/decomposition of the matrix

famous property of F.T.

(is equivalent to)

multiplication in Freq. Domain $\Longleftrightarrow$ convolution in space/time domain

(+ vice versa)

convolution $f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$

correlation $f(t) * g(-t) = \int_{-\infty}^{\infty} f(\tau)\, g(t+\tau)\, d\tau$

element by element mult.

conv: $f(t) * g(t) \Longleftrightarrow F(\omega) \cdot G(\omega)$

corr: $f(t) * g(-t) \Longleftrightarrow F(\omega) \cdot G^{*}(\omega)$

$*$ = convolution

$*$ = complex conjugate

$$f(t) * g(t) = \mathcal{F}^{-1}\left(F(\omega)\, G(\omega)\right)$$

$$\boxed{f(t) \text{ corr } g(t) = \mathcal{F}^{-1}\left\{F(\omega)\, G^*(\omega)\right\}}$$

if data has length 2^n, use FFT

1. F.T. signal/image to Freq. Domain (incl. conjugate)
2. element by element mult.
3. inverse F.T. back to space/time domain
4. have CC response map

matlab has DFT, FFT, FFT2, IFFT, IFFT2, .* for element-wise multiplication