# Image Matching: Correlation

## COMPSCI 773 S1C

A/Prof. Patrice Delmas and Prof. Georgy Gimel'farb

1. Correlation
2. 2D correlation
3. Faster matching
4. LS correlation (optional)
5. Concurrent matching (optional)

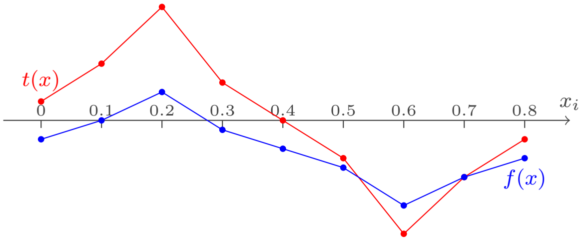
# Correlation Matching: A Least Squares Technique

Given two time or spatial series of 1D signals

$$\left[\{t_i = t(x_i), f_i = f(x_i) :\ i = 1, \ldots, n\}; x_1 < \ldots < x_n\right]$$

find a "constant contrast $b$ – offset $a$" transformation, $f(x) = a + bt(x)$, minimising the sum of squared deviations

$$L(a,b) = \sum_{i=1}^{n} \left(f(x_i) - (a + bt(x_i))\right)^2 \equiv \sum_{i=1}^{n} \left(f_i - (a + bt_i)\right)^2$$



| $i$ | $x_i$ | $t_i$ | $f_i$ |
|---|---|---|---|
| 1 | 0 | 0.5 | −0.50 |
| 2 | 0.1 | 1.5 | 0.00 |
| 3 | 0.2 | 3.0 | 0.75 |
| 4 | 0.3 | 1.0 | −0.25 |
| 5 | 0.4 | 0.0 | −0.75 |
| 6 | 0.5 | −1.0 | −1.25 |
| 7 | 0.6 | −3.0 | −2.25 |
| 8 | 0.7 | −1.5 | −1.50 |
| 9 | 0.8 | −0.5 | −1.00 |

# Minimiser $(a^*, b^*)$ for Matching Score $L(a, b)$

$$
\begin{aligned}
L(a,b) &= \sum_{i=1}^{n} \left(f_i - (a + bt_i)\right)^2 \\
&\equiv S_{ff} - 2aS_f - 2bS_{ft} + a^2 n + 2abS_t + b^2 S_{tt}
\end{aligned}
$$

where

$$
S_{ff} = \sum_{i=1}^{n} f_i^2; \quad S_{ft} = \sum_{i=1}^{n} f_i t_i; \quad S_{tt} = \sum_{i=1}^{n} t_i^2;
$$
$$
S_t = \sum_{i=1}^{n} t_i; \qquad S_f = \sum_{i=1}^{n} f_i
$$

**Normal equations**:

$$
\begin{aligned}
\frac{\partial L}{\partial a} &= -2S_f + 2an + 2bS_t &= 0 \quad \Rightarrow \quad an + bS_t = S_f \\
\frac{\partial L}{\partial b} &= -2S_{ft} + 2aS_t + 2bS_{tt} &= 0 \quad \Rightarrow \quad aS_t + bS_{tt} = S_{ft}
\end{aligned}
$$

$$
\Rightarrow \qquad \left[\begin{array}{cc} n & S_t \\ S_t & S_{tt} \end{array}\right] \left[\begin{array}{c} a \\ b \end{array}\right] = \left[\begin{array}{c} S_f \\ S_{ft} \end{array}\right]
$$

# Minimiser $(a^*, b^*)$ for Matching Score $L(a, b)$

Solving normal equations:

$$\begin{bmatrix} a^* \\ b^* \end{bmatrix} = \frac{1}{nS_{tt} - S_t^2} \begin{bmatrix} S_{tt} & -S_t \\ -S_t & n \end{bmatrix} \begin{bmatrix} S_f \\ S_{ft} \end{bmatrix}$$

$$a^* = \tfrac{1}{nS_{tt} - S_t^2} \left( S_{tt} S_f - S_t S_{ft} \right)$$

$$b^* = \tfrac{1}{nS_{tt} - S_t^2} \left( -S_t S_f + n S_{ft} \right)$$

$$\Rightarrow \; a^* = \frac{S_f}{n} - b^* \cdot \frac{S_t}{n} \; \Rightarrow \; f^*(x) = \frac{S_f}{n} + b^* \cdot \left( t(x) - \frac{S_t}{n} \right)$$

Minimum sum of squared deviations ($\bar{f} = \frac{S_f}{n}$; $\bar{t} = \frac{S_t}{n}$ – mean signals):

$$L(a^*, b^*) = \sum_{i=1}^{n} \left( f(x_i) - \bar{f} \right)^2 - \frac{\left( \sum\limits_{i=1}^{n} \left( f(x_i) - \bar{f} \right) \left( t(x_i) - \bar{t} \right) \right)^2}{\sum\limits_{i=1}^{n} \left( t(x_i) - \bar{t} \right)^2}$$

# Correlation Matching

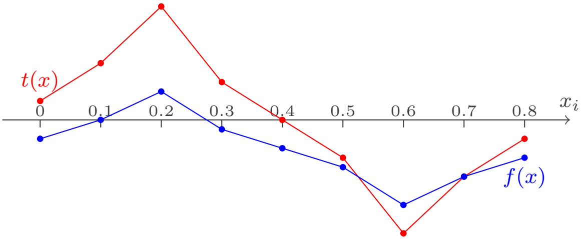
Minimum sum of squared deviations, or matching distance

$$D^*_{ft} \equiv L(a^*, b^*) = n\left(\sigma_f^2 - \frac{\sigma_{ft}^2}{\sigma_t^2}\right) \equiv n\sigma_f^2\left(1 - C_{ft}^2\right)$$

where

- $\sigma_f^2 = \frac{1}{n}\sum\limits_{i=1}^{n}\left(f(x_i) - \bar{f}\right)^2$ – the variance of the signals $f$
- $\sigma_t^2 = \frac{1}{n}\sum\limits_{i=1}^{n}\left(t(x_i) - \bar{t}\right)^2$ – the variance of the signals $t$
- $\bar{f} = \frac{S_f}{n}$ and $\bar{t} = \frac{S_t}{n}$ – the mean signals $f$ and $t$
- $\sigma_{ft} = \frac{1}{n}\sum\limits_{i=1}^{n}\left(f(x_i) - \bar{f}\right)\left(t(x_i) - \bar{t}\right)$ – the signal covariance
- $C_{ft} = \frac{\sigma_{ft}}{\sigma_f\sigma_t}$; $-1 \leq C_{ft} \leq 1$ – the **correlation** (matching score)

# Correlation Matching: An Example



| $i$ | $x_i$ | $t_i$ | $f_i$ |
|---|---|---|---|
| 1 | 0 | 0.5 | −0.50 |
| 2 | 0.1 | 1.5 | 0.00 |
| 3 | 0.2 | 3.0 | 0.75 |
| 4 | 0.3 | 1.0 | −0.25 |
| 5 | 0.4 | 0.0 | −0.75 |
| 6 | 0.5 | −1.0 | −1.25 |
| 7 | 0.6 | −3.0 | −2.25 |
| 8 | 0.7 | −1.5 | −1.50 |
| 9 | 0.8 | −0.5 | −1.00 |

$$S_t = 0;\ S_{tt} = 25;\ S_f = -6.75;\ S_{ff} = 11.3125;\ S_{ft} = 12.5 \Rightarrow$$

$$b^* = \frac{1}{9\cdot 25 - 0^2}(-0\cdot(-6.75) + 9\cdot 12.5) = 0.5;\ a^* = \frac{-6.75}{9} - 0.5\cdot\frac{0}{9} = -0.75$$

$$\Rightarrow f(x) = -0.75 + 0.5\cdot t(x);\ \bar{f} = -0.75;\ \sigma_f^2 = \frac{6.25}{9};\ \sigma_t^2 = \frac{25}{9};\ \sigma_{ft} = \frac{12.5}{9}$$

$$\Rightarrow C_{ft} = \frac{\frac{25}{9}}{\frac{2.5}{3}\frac{5}{3}} = 1;\ D_{ft}^* = 9\frac{6.25}{9}(1 - 1^2) = 0$$

# Correlation Matching: Probability Model of Signals

Signals $f$ as a transformed template $t$ corrupted by a centre-symmetric independent random noise $r$ (e.g. the Gaussian noise)

For $i = 1, \ldots, n$,

$$f_i = a + bt_i + r_i \;\Rightarrow\; p(r_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(f_i - (a+bt_i))^2}{2\sigma^2}\right)$$

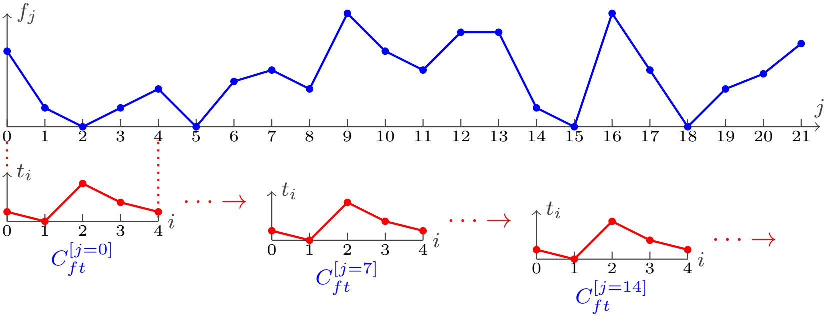
$$\Rightarrow\; P_{a,b}(f|t) = \prod_{i=1}^{n} p(r_i) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left(-\frac{\sum\limits_{i=1}^{n} (f_i - (a+bt_i))^2}{2\sigma^2}\right)$$

Maximum likelihood of $t$ for $f$ in transforming parameters $a$ and $b$ results in the correlation matching:

$$\max_{a,b} P_{a,b}(f|t) \;\Rightarrow\; \min_{a,b} \sum_{i=1}^{n} (f_i - (a + bt_i))^2$$

# Search for the Best Matching Position

- Matching a template $t = [t_i : \ i = 1, \dots, n]$ to a much longer data sequence $f = [f_j : \ j = 1, \dots, N]$; $N > n$
- Goal position $j^*$ maximises the correlation $C_{ft}$ (or minimises the distance $D_{ft}$) between $t$ and the segment $[f_{j+i} : \ i = 1, \dots, n]$ of $f$



# 2D Correlation

2D $m \times n$ template $t$ and $M \times N$ image $f$; $m < M$; $n < N$:

$$\begin{array}{lcllll} t & = & [t_{i'j'}: & i' = 0, \ldots, n-1; & j' = 0, \ldots, m-1] \\ f & = & [f_{ij}: & i = 0, \ldots, N-1; & j = 0, \ldots, M-1] \end{array}$$

An example:

Eye template $t$ $32 \times 18$ pixels:

Facial image $f$ $200 \times 200$ pixels:

$t$

$f$

**Moving window matching**:
Search for a window position $(i^*, j^*)$ in $f$ such that maximises the correlation $C_{ft}$ (minimises the distance $D_{ft}$) between the template $t$ and the underlying region of the image $f$ in the moving window

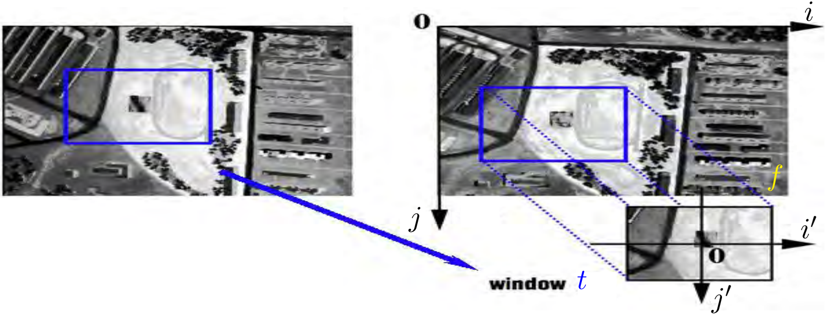
# 2D Correlation

Template $t : \mathbf{W} \rightarrow \mathbf{Q}$

- $\mathbf{W} = ((i', j') : i' = 0, \ldots, n-1;\ j' = 0, \ldots, m-1)$ – a fixed-size rectangular window of size $m \times n$ supporting the template

Target $f : \mathbf{R} \rightarrow \mathbf{Q}$

- $\mathbf{R} = ((i, j) : i = 0, 1, \ldots, N-1;\ j = 0, 1, \ldots, M-1)$ – a fixed-size arithmetic lattice supporting the target



# 2D Correlation

Distance between the template $t$ and the moving window in position $(i, j)$ in the image $f$:

$$D_{ij} = \underbrace{\sum_{i'=0}^{n-1}\sum_{j'=0}^{m-1} \tilde{f}^2_{i+i',j+j'}}_{\sigma^2_{f:[ij]}} - \left. \frac{\left(\displaystyle\sum_{i'=0}^{n-1}\sum_{j'=0}^{m-1} \tilde{f}_{i+i',j+j'}\tilde{t}_{i',j'}\right)^2}{\displaystyle\sum_{i=1}^{n} \tilde{t}^2_{i',j'}} \right\} \frac{\sigma^2_{ft:[ij]}}{\sigma^2_t}$$

- Centred signals: $\tilde{f}_{i+i',j+j'} = f_{i+i',j+j'} - \bar{f}_{[ij]}$ and $\tilde{t}_{i',j'} = t_{i',j'} - \bar{t}$
- Mean for the moving window: $\bar{f}_{[ij]} = \frac{1}{mn}\sum\limits_{i'=0}^{n-1}\sum\limits_{j'=0}^{m-1} f_{i+i',j+j'}$
- Variance for the window: $\sigma^2_{f:[ij]} = \frac{1}{mn}\sum\limits_{i'=0}^{n-1}\sum\limits_{j'=0}^{m-1} \left(f_{i+i',j+j'} - \bar{f}_{[ij]}\right)^2$

# 2D Correlation

- Fixed template mean: $\bar{t} = \frac{1}{mn} \sum\limits_{i'=0}^{n-1} \sum\limits_{j'=0}^{m-1} t_{i',j'}$
- Fixed template variance: $\sigma_t^2 = \frac{1}{mn} \sum\limits_{i'=0}^{n-1} \sum\limits_{j'=0}^{m-1} \left(t_{i',j'} - \bar{t}\right)^2$
- Window–template covariance:

$$\sigma_{ft:[ij]} = \frac{1}{mn} \sum_{i'=0}^{n-1} \sum_{j'=0}^{m-1} \left(f_{i+i',j+j'} - \bar{f}_{[ij]}\right) \left(t_{i',j'} - \bar{t}\right)$$

- **Correlation matching**: $C_{ft:[ij]} = \frac{\sigma_{ft:[ij]}}{\sigma_{f:[ij]}\sigma_t}$; $-1 \leq C_{ft:[ij]} \leq 1$
  - Distance: $D^*_{ft:[ij]} \equiv L(a^*, b^*) = mn\sigma^2_{f:[ij]} \left(1 - C^2_{ft:[ij]}\right)$

# An Aerial Stereo Pair

Note road traffic differences due to acquisition at different time; occluded walls of high buildings; varying contrast / brightness, etc.

# Noise Models to Find the Matching Score

- $f(i+i',j+j') = t(i',j') + r(i',j')$; independent Gaussian noise $r$: Sum of squared distances

$$\mathrm{SSD}(i,j) = \sum_{(i',j')\in\mathbf{W}} (f(i+i',j+j') - t(i',j')^2$$

- $f(i+i',j+j') = t(i',j') + r(i',j')$; independent symmetric noise $r$: Sum of absolute distances

$$\mathrm{SAD}(i,j) = \sum_{(i',j')\in\mathbf{W}} |f(i+i',j+j') - t(i',j')|$$

- Uniform contrast/offset $f(i+i',j+j') = a + bt(i',j') + r(i',j')$; independent Gaussian noise $r$:

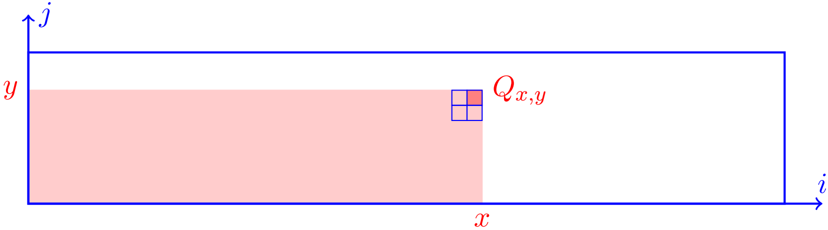
$$\text{Cross-correlation } C_{ft:[ij]}$$

- Varying contrast/offset $b(i,j)$; $a(i,j)$: Numerical distance minimisation (e.g. by quadratic programming)

# Faster Implementation of Window Based Operations

Accumulator of pixel-wise values:

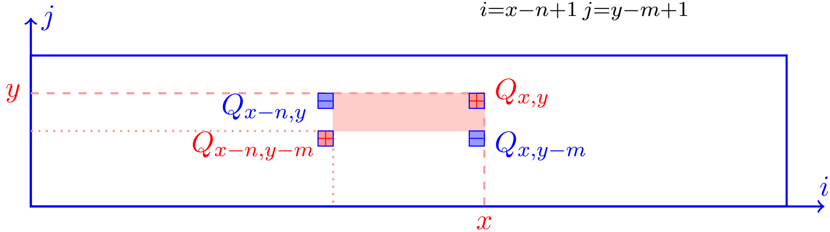
$$Q_{x,y} = \sum_{i=0}^{x} \sum_{j=0}^{y} q_{i,j}; \quad (x,y) \in \mathbf{R}$$



$$Q_{x,y} = \begin{cases} q_{x,y} & x = y = 0 \\ q_{x,y} + Q_{x-1,y} & x > 0; y = 0 \\ q_{x,y} + Q_{x,y-1} & x = 0; y > 0 \\ q_{x,y} + Q_{x-1,y} + Q_{x,y-1} - Q_{x-1,y-1} & x > 0; y > 0 \end{cases}$$

# Faster Implementation of Window Based Operations

Sum of values in a window: $U_{x,y;n,m} = \sum_{i=x-n+1}^{x} \sum_{j=y-m+1}^{y} q_{i,j}$



Using the accumulator:

$$U_{x,y;n,m} = Q_{x,y} - Q_{x-n,y} - Q_{x,y-m} + Q_{x-n,y-m}$$

Complexity of fast computation: $O(MN)$

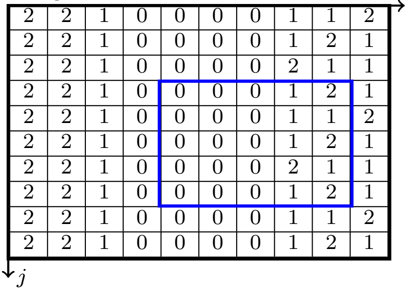
- $MN = |\mathbf{R}|$ – the image size

Complexity of straightforward computation: $O(mnMN)$

- $mn = |\mathbf{W}|$ – the window size
- Even for a small window $11 \times 11$ pixels: $\approx 120$ times faster!

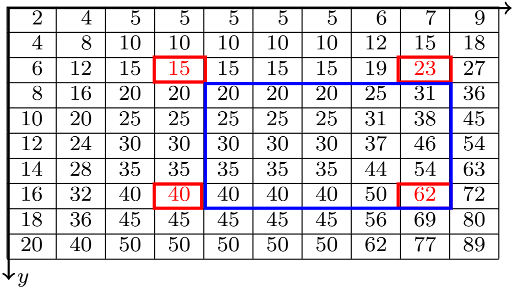
# Accumulating Window Sums

Image $10 \times 10$ and a window $5 \times 5$



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 1 | 1 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 1 | 1 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |

Accumulator $10 \times 10$



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 5 | 5 | 5 | 5 | 5 | 6 | 7 | 9 |
| 4 | 8 | 10 | 10 | 10 | 10 | 10 | 12 | 15 | 18 |
| 6 | 12 | 15 | 15 | 15 | 15 | 15 | 19 | 23 | 27 |
| 8 | 16 | 20 | 20 | 20 | 20 | 20 | 25 | 31 | 36 |
| 10 | 20 | 25 | 25 | 25 | 25 | 25 | 31 | 38 | 45 |
| 12 | 24 | 30 | 30 | 30 | 30 | 30 | 37 | 46 | 54 |
| 14 | 28 | 35 | 35 | 35 | 35 | 35 | 44 | 54 | 63 |
| 16 | 32 | 40 | 40 | 40 | 40 | 40 | 50 | 62 | 72 |
| 18 | 36 | 45 | 45 | 45 | 45 | 45 | 56 | 69 | 80 |
| 20 | 40 | 50 | 50 | 50 | 50 | 50 | 62 | 77 | 89 |

- Straightforward summing (25 operations):

$$\begin{aligned} U_{6,5;2,2} &= 0+0+0+1+2+0+0+0+1+1+0+0+0 \\ &\quad +1+2+0+0+0+2+1+0+0+0+1+2=14 \end{aligned}$$

- Summing with the use of the accumulated values (4 operations):

$$U_{6,5;2,2} = \underbrace{62}_{Q_{8,7}} - \underbrace{40}_{Q_{3,7}} - \underbrace{23}_{Q_{8,2}} + \underbrace{15}_{Q_{3,2}} = 14$$

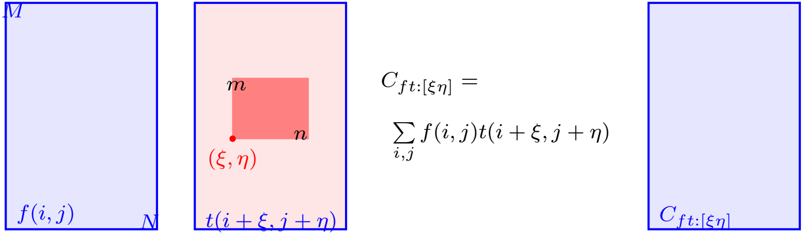
# Faster Implementation of Correlation

- To compute fast the mean signals $\bar{f}_{[ij]}$: $q_{ij} \leftarrow f_{ij}$
- To compute fast the variances $\sigma^2_{f:[ij]}$:
  1. $q_{ij} \leftarrow f^2_{ij}$ to compute fast $S_{ff:[ij]} = \sum\limits_{(i',j')\in\mathbf{W}} f^2_{i+i',j+j'}$
  2. $\sigma^2_{f:[ij]} = \frac{1}{mn} S_{ff:[ij]} - \bar{f}^2_{[ij]}$
- But the sums $S_{ft:[ij]} = \sum_{(i',j')\in\mathbf{W}} f_{i+i',j+j'} t_{i',j'}$ to compute the covariances $\sigma_{ft:[ij]}$ cannot be obtained fast so simply
- An alternative approach – to use the spectral space and FFT:

$$\begin{aligned}\mathbb{F}(u,v) &= \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} f(i,j)\exp\left(-\tfrac{2\pi\iota}{N}u - \tfrac{2\pi\iota}{M}v\right)\\ \mathbb{T}(u,v) &= \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} t(i,j)\exp\left(-\tfrac{2\pi\iota}{N}u - \tfrac{2\pi\iota}{M}v\right)\end{aligned}$$

(here and below, $\iota = \sqrt{-1}$)

# Faster Implementation of Correlation



$$C_{ft:[\xi\eta]} = \sum_{i,j} f(i,j)t(i+\xi,j+\eta)$$

- Correlation in the spectral space:
  1. Compute the target and template spectra $\mathbb{F}(u,v)$ and $\mathbb{T}(u,v)$
  2. Compute the correlation spectrum $\mathbb{C}(u,v) = \mathbb{F}(u,v)\mathbb{T}^{\circ}(u,v)$
  3. Find the correlations $\left(C_{ft:[\xi\eta]} : (\xi,\eta) \in \mathbf{R}\right)$ by the inverse FFT
  4. Find the maximum correlation $(\xi^*,\eta^*) = \arg\max\limits_{\xi,\eta} C_{ft:[\xi\eta]}$
- Complexity of the spectral approach: $O\left(MN(\log MN)\right)$
- The accumulator-based acceleration is used for fast correlation stereo matching (with one accumulator per disparity level)

# Least Squares Correlation

Search for geometric transformations $\mathbf{a}$, which maximise the cross-correlation between the template and the target:

$$C_{ft:\mathbf{a}^*} = \arg\max_{\mathbf{a}}\{C_{ft:\mathbf{a}}\}$$

Simplified case: affine transformations

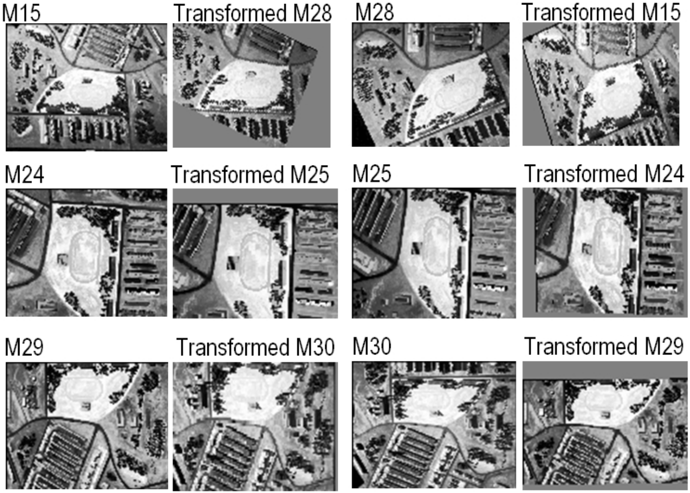
$$\begin{cases} x_{\mathbf{a}} & = & a_1 x + a_2 y + a_3 \\ y_{\mathbf{a}} & = & a_4 x + a_5 y + a_6 \end{cases}$$

Combined exhaustive and directed (e.g. gradient-based) search for affine parameters:

- Exhaustion of a sparse grid of relative translations $(a_3, a_6)$ of a fixed template $t$ with respect to the target image $f$
- Directed optimisation of $C_{\mathbf{a}}$ in all $\mathbf{a} = [a_1, \ldots, a_6]$ affine parameters starting from every grid point $[1, 0, a_3, 0, 1, a_6]$

# Affinely transformed corresponding windows

"Radius" Database (multiple images of the same scene): R. M. Haralick, 1995



Note imperfect affine transformations between M15 and M28

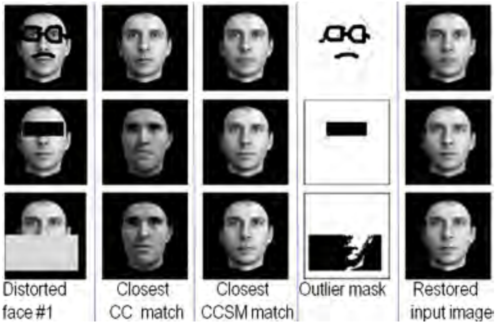
# Soft masking of outliers

A more general noise model for the windows:

- Mixture of independent Gaussian errors and uniform outliers
- Unknown errors prior and variance
- Uniform global contrast and offset

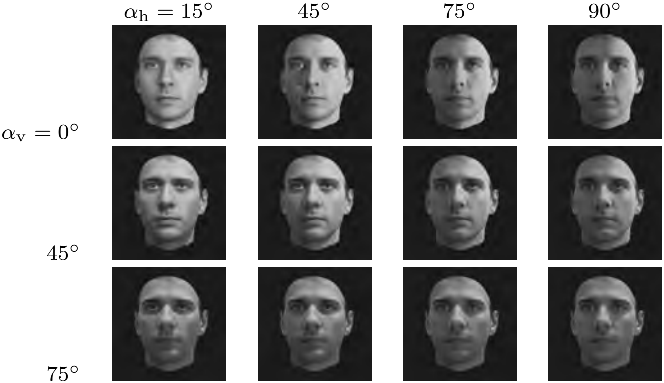
$\Rightarrow$ Iterative cross-orrelation with soft masking of outliers

Matching a distorted image to 24 templates (MIT database)



# Polynomial Contrast / Offset Deviations

Image changes under varying illumination in terms of horizontal, $\alpha_{\mathrm{h}}$, and vertical, $\alpha_{\mathrm{v}}$, angular positions of two light sources



Polynomial model: an image $p_n(x,y)t(x,y) + q_n(x,y)$ – polynomials of order $n$: e.g. $(a_{10}x + a_{01}y + a_{00})t(x,y) + (b_{10}x + b_{01}y + b_{00})$ for $n = 1$ (linear deviation model)

# Least Squares Image Matching

- Polynomial transformations of a template $\mathbf{t} = [t(x,y) : \ (x,y) \in \mathbf{R}]$:

$$\begin{aligned} & (a_{10}x + a_{01}y + a_{00})t(x,y) + (b_{10}x + b_{01}y + b_{00}) \\ \equiv \quad & a_{10}xt(x,y) + a_{01}yt(x,y) + a_{00}t(x,y) + b_{10}x + b_{01}y + b_{00} \\ \equiv \quad & \underbrace{[a_0, a_1, a_2, b_0, b_1, b_2]}_{\mathbf{a}^\mathsf{T}} \underbrace{\begin{bmatrix} xt(x,y) \\ yt(x,y) \\ t(x,y) \\ x \\ y \\ 1 \end{bmatrix}}_{\boldsymbol{\tau}(x,y)} \end{aligned}$$

- Squared distance between an image g and the transformed template:

$$D(\mathbf{f}, \mathbf{t}) = \min_{\mathbf{a}} \sum_{(x,y)\in\mathbf{R}} \left( f(x,y) - \mathbf{a}^\mathsf{T}\boldsymbol{\tau}(x,y) \right)^2$$

# Least Squares Image Matching

$$\boldsymbol{\nabla} D(\mathbf{f},\mathbf{t}) = \mathbf{0} \Rightarrow \underbrace{\sum_{(x,y)\in\mathcal{R}} \boldsymbol{\tau}(x,y)\boldsymbol{\tau}^{\mathsf{T}}(x,y)}_{\mathbf{A}}\,\mathbf{a} = \underbrace{\sum_{(x,y)\in\mathcal{R}} f(x,y)\boldsymbol{\tau}(x,y)}_{\mathbf{b}}$$

The $6\times 6$ matrix $\mathbf{A}$ and $6\times 1$ vector $\mathbf{b}$:

$$\mathbf{A} = \begin{bmatrix} \mathrm{X^2T^2} & \mathrm{XYT^2} & \mathrm{XT^2} & \mathrm{X^2T} & \mathrm{XYT} & \mathrm{XT} \\ \mathrm{XYT^2} & \mathrm{Y^2T^2} & \mathrm{YT^2} & \mathrm{XYT} & \mathrm{Y^2T} & \mathrm{YT} \\ \mathrm{XT^2} & \mathrm{YT^2} & \mathrm{T^2} & \mathrm{XT} & \mathrm{YT} & \mathrm{T} \\ \mathrm{X^2T} & \mathrm{XYT} & \mathrm{XT} & \mathrm{X^2} & \mathrm{XY} & \mathrm{X} \\ \mathrm{XYT} & \mathrm{Y^2T} & \mathrm{YT} & \mathrm{XY} & \mathrm{Y^2} & \mathrm{Y} \\ \mathrm{XT} & \mathrm{YT} & \mathrm{T} & \mathrm{X} & \mathrm{Y} & N \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} \mathrm{FXT} \\ \mathrm{FYT} \\ \mathrm{FT} \\ \mathrm{FX} \\ \mathrm{FY} \\ \mathrm{F} \end{bmatrix}$$

where $\mathrm{X}^i\mathrm{Y}^j\mathrm{T}^k = \sum_{(x,y)\in\mathcal{R}} x^i y^j t^k(x,y);\ i,j,k \in \{0,1,2\}$,
$\mathrm{FX}^i\mathrm{Y}^j\mathrm{T}^k = \sum_{(x,y)\in\mathcal{R}} x^i y^j t^k(x,y) f(x,y)$, and
$N = |\mathbf{R}|$ is the lattice cardinality (the number of pixels)

# Concurrent Template Matching

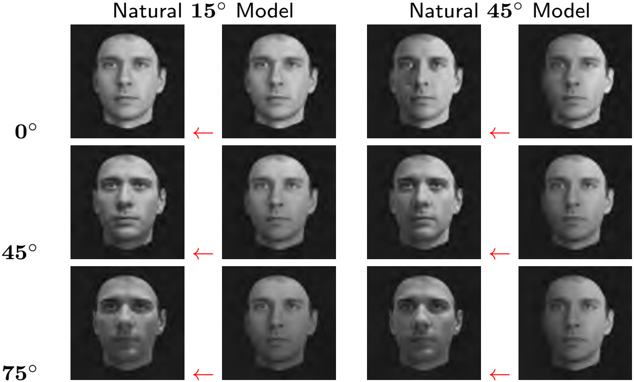
- Maximiser $\mathbf{a}^*$ – the solution of the linear system $\mathbf{A}\mathbf{a} = \mathbf{b}$ with the $6 \times 6$ square symmetric matrix $\mathbf{A}$: $\mathbf{a}^* = \mathbf{A}^{-1}\mathbf{b}$
- Transformed template $\mathbf{T}_{[6]} = \{\mathbf{t}_k : k = 1, \ldots, 6\}$: for $(x, y) \in \mathbf{R}$,

$$\begin{array}{lll} t_1(x,y) = xt(x,y); & t_2(x,y) = yt(x,y); & t_3(x,y) = t(x,y); \\ t_4(x,y) = x; & t_5(x,y) = y; & t_6(x,y) = 1 \end{array}$$

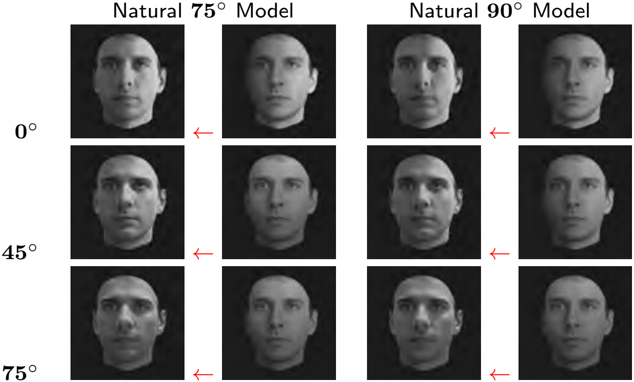
- Orthogonalisation of $\mathbf{T}_{[6]}$ $\Rightarrow$ Concurrent matching with the bank of orthonormal templates $\mathbf{E}_{[6]} = \{\mathbf{e}_k : k = 1, \ldots, 6\}$
  - Squared distance between an image $\mathbf{f}$ and the transformed $\mathbf{t}$:

$$D(\mathbf{f}, \mathbf{t}) = \min_{\mathbf{a}} \sum_{(x,y)\in\mathcal{R}} f^2(x,y) - \sum_{i=1}^{6} \left( \sum_{(x,y)\in\mathbf{R}} f(x,y) e_i(x,y) \right)^2$$

# Orthonormal Templates: Modelling Illumination Changes



# Orthonormal Templates: Modelling Illumination Changes

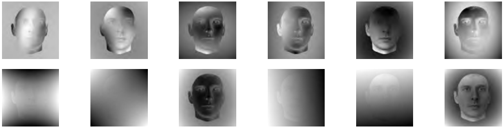


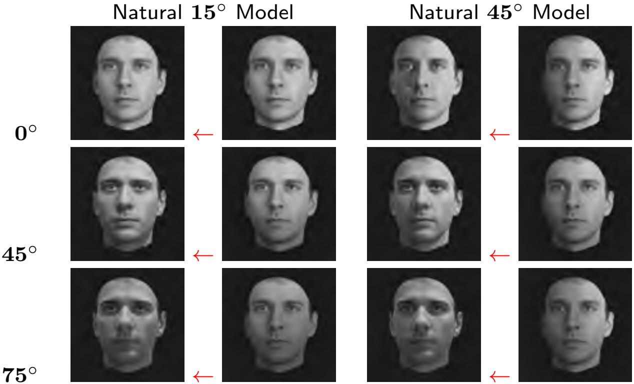
# $2^{nd}$-order Polynomial: Concurrent Template Matching

Bank of the $2^{nd}$-order orthonormal templates:

- Real illumination-dependent image changes are modelled only roughly with the first-order polynomial contrast / offset
- More accurate modelling - with the second- or higher-order polynomial contrast / offset
  - But the number of templates grows as $(n+1)(n+2) = O(n^2)$ in the polynomial order $n$

# $2^{nd}$-order Templates: Modelling Illumination Changes





# Orthonormal Templates: Modelling Illumination Changes