Displacement estimation

- Displacement estimation by block matching
  - Search strategies
  - Subpixel estimation
- Gradient-based displacement estimation (“optical flow”)
  - Lukas-Kanade
  - Multi-scale coarse-to-fine
  - Parametric displacement estimation

Where is the defect?

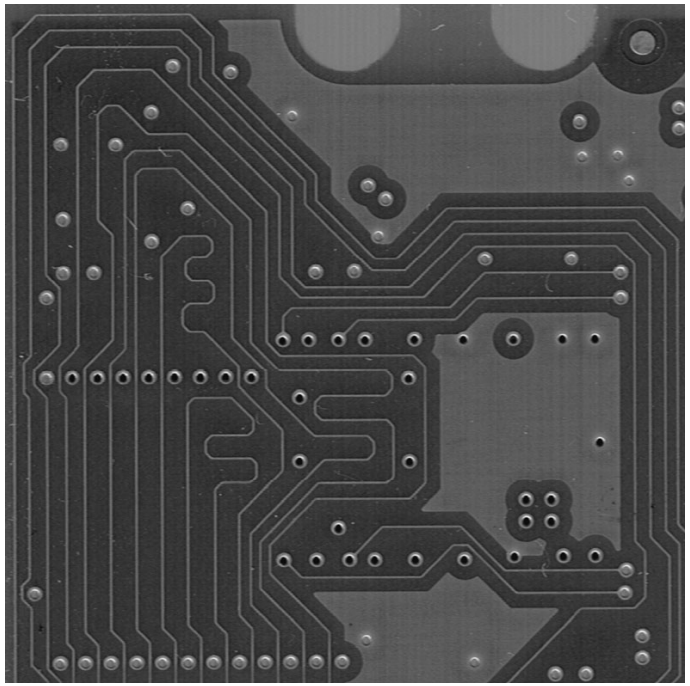


Image $g[x,y]$ (no defect)

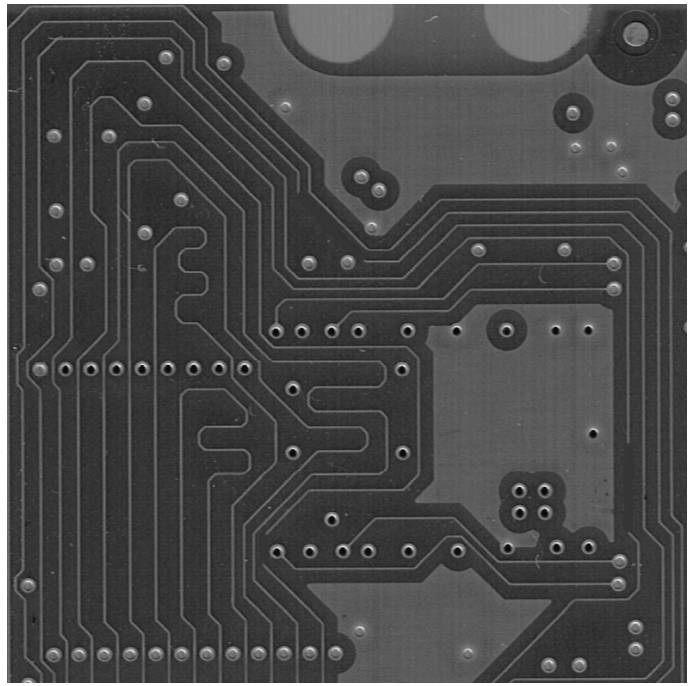
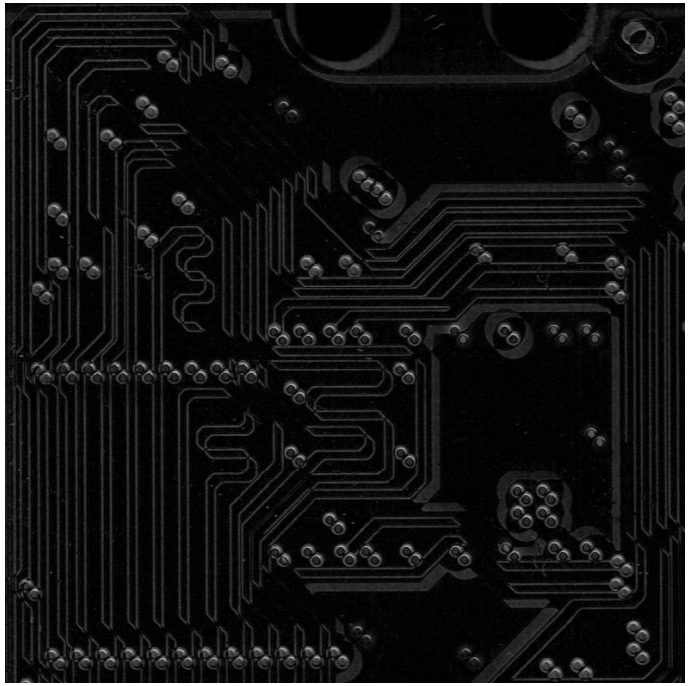
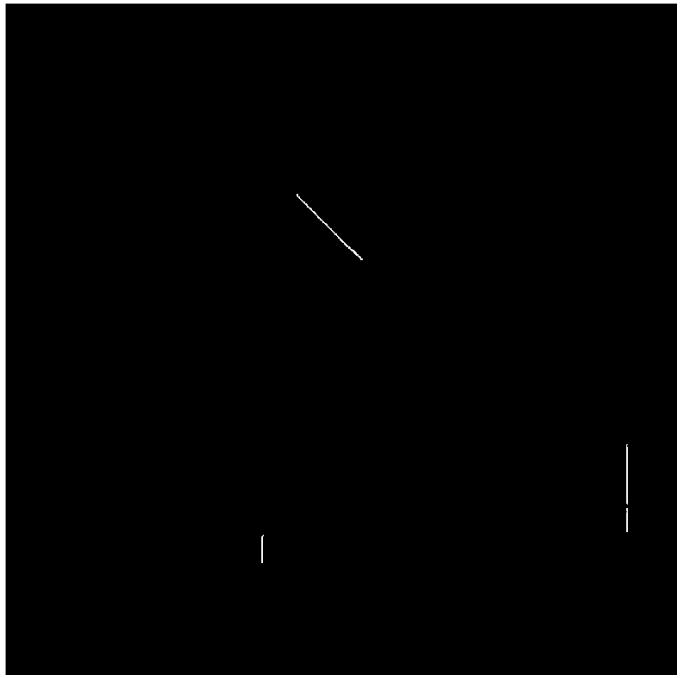


Image $f[x,y]$ (w/ defect)

Absolute difference between two images



$|f\text{-}g|$ w/o alignment



$|f\text{-}g|$ w/ alignment

Displacement estimation by block matching

Measurement window is compared with a shifted array of pixels in the other image, to determine the best match

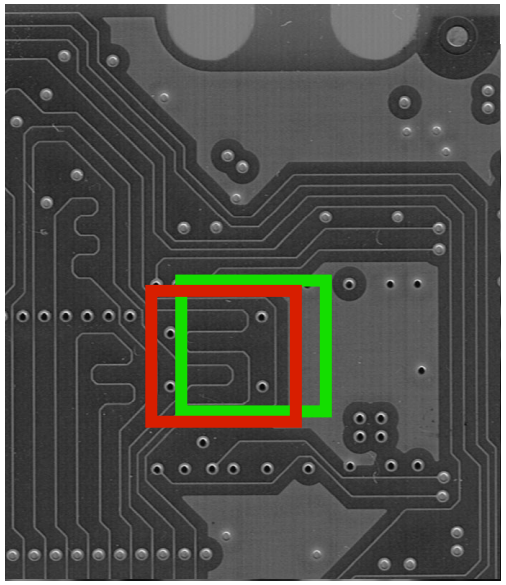


Image $g[x,y]$

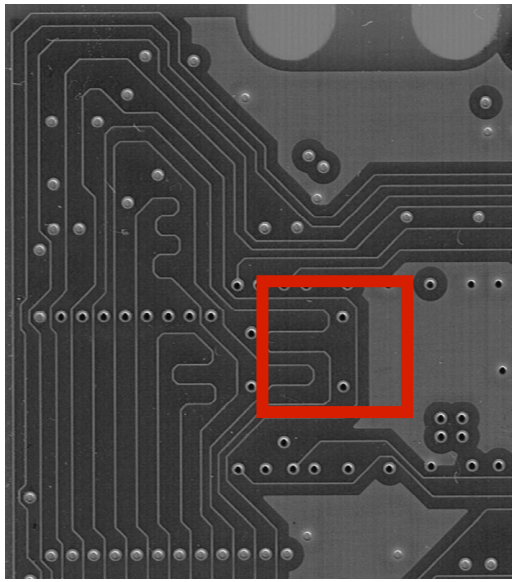


Image $f[x,y]$

Rectangular array of pixels is selected as a measurement window

Displacement estimation by block matching

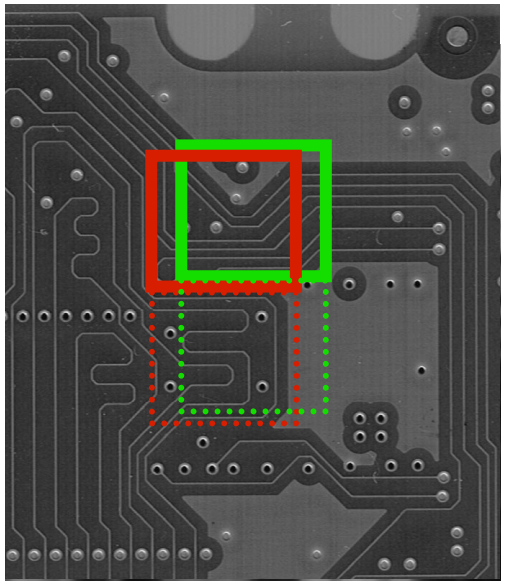


Image $g[x,y]$

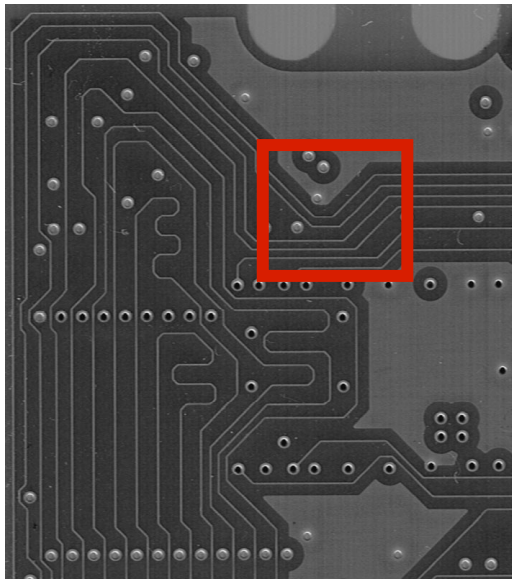


Image $f[x,y]$

. . . process repeated for another measurement window position.

Integer pixel shifts

Measurement window is compared with a shifted array of pixels in the other image, to determine the best match

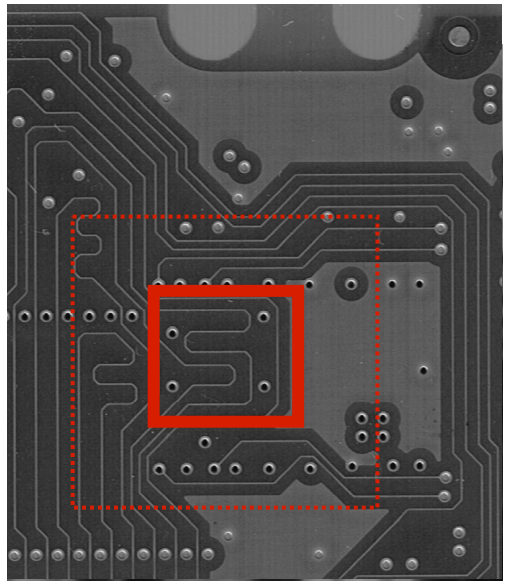


Image $g[x,y]$

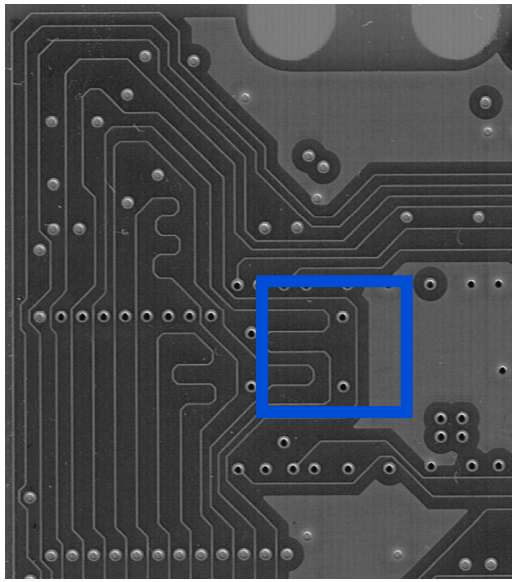
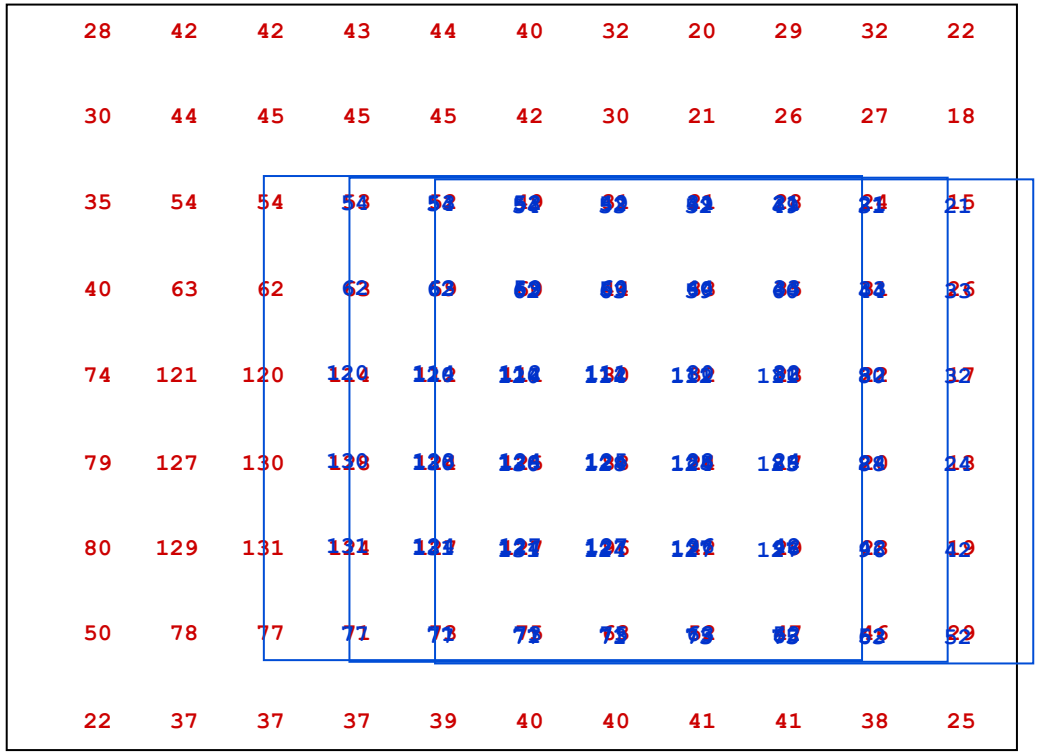


Image $f[x,y]$

Rectangular array of pixels is selected as a measurement window

Integer pixel shifts



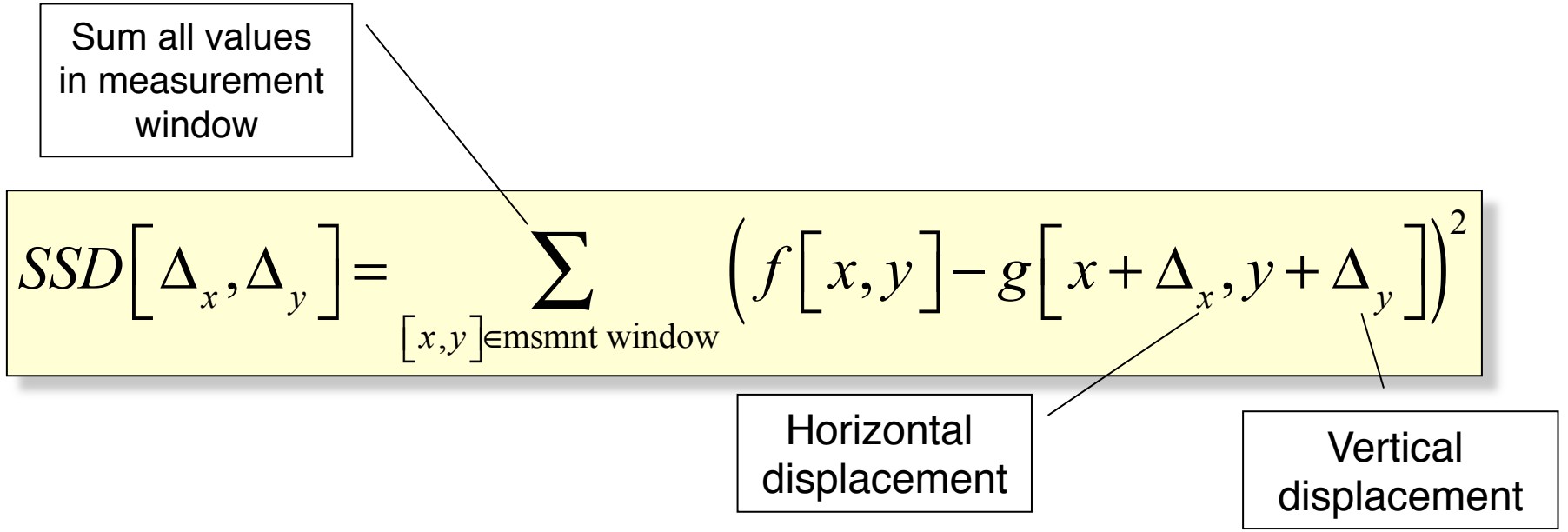
Measurement window is compared with a shifted array of pixels in the other image, to determine the best match

54	53	52	49	31	21
62	63	59	60	44	33
120	114	112	111	80	32
130	128	124	125	88	24
131	124	127	127	96	42
77	71	73	75	63	52

Rectangular array of pixels is selected as a measurement window

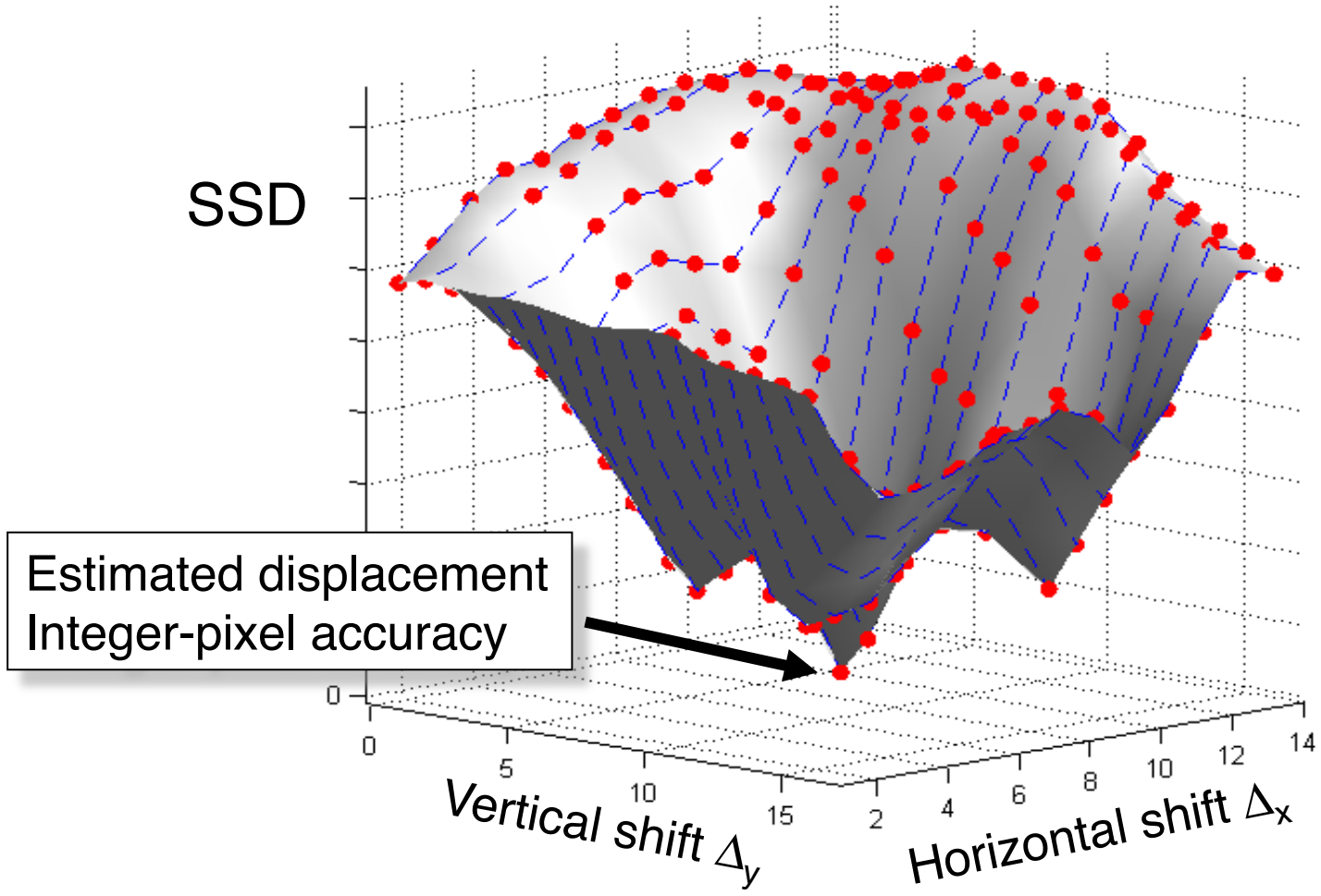
Error metric

- *Sum of Squared Differences*



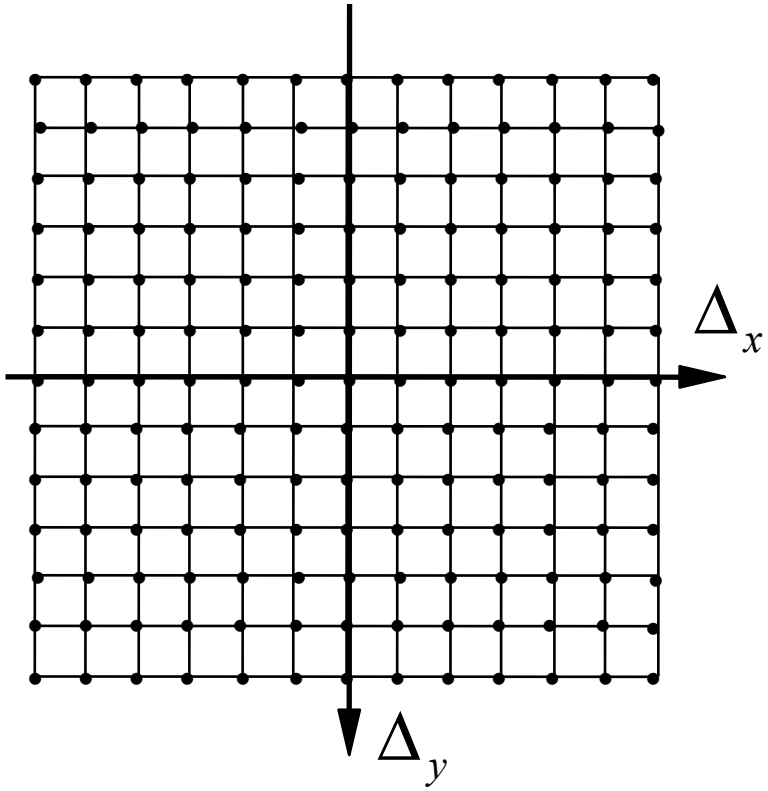
- Alternatives: SAD (*Sum of Absolute Differences*), cross correlation, mutual information . . .
- Robustness against outliers: sum of saturated squared differences, median of squared differences . . .

SSD values resulting from block matching



Block matching: search strategies

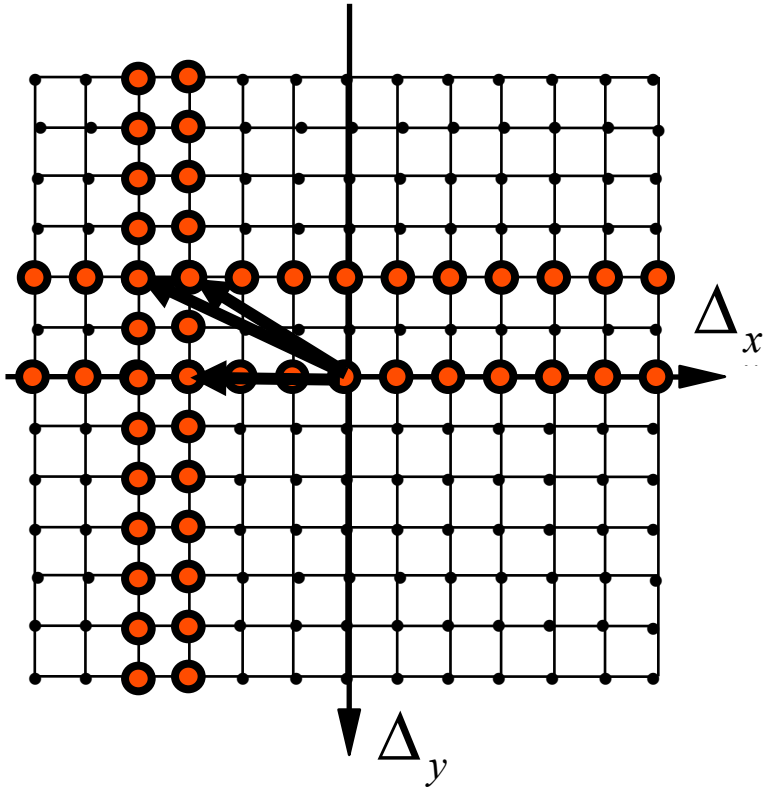
Full search



- All possible displacements within the search range are compared.
- Computationally expensive
- Highly regular, parallelizable

Block matching: search strategies

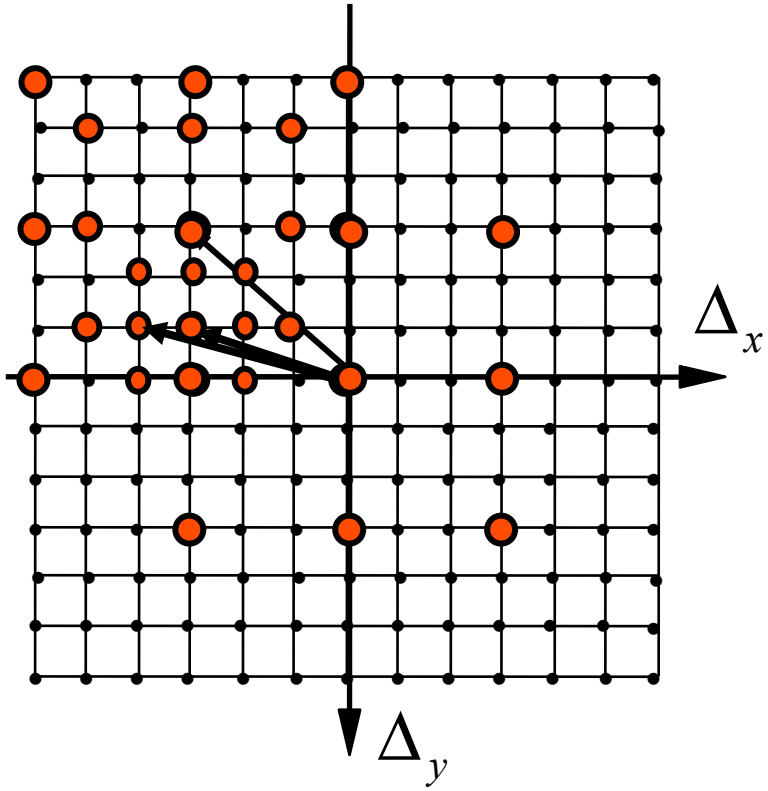
Conjugate direction search



- Alternate search in x and y directions
- Stop when there is no further improvement

Block matching: search strategies

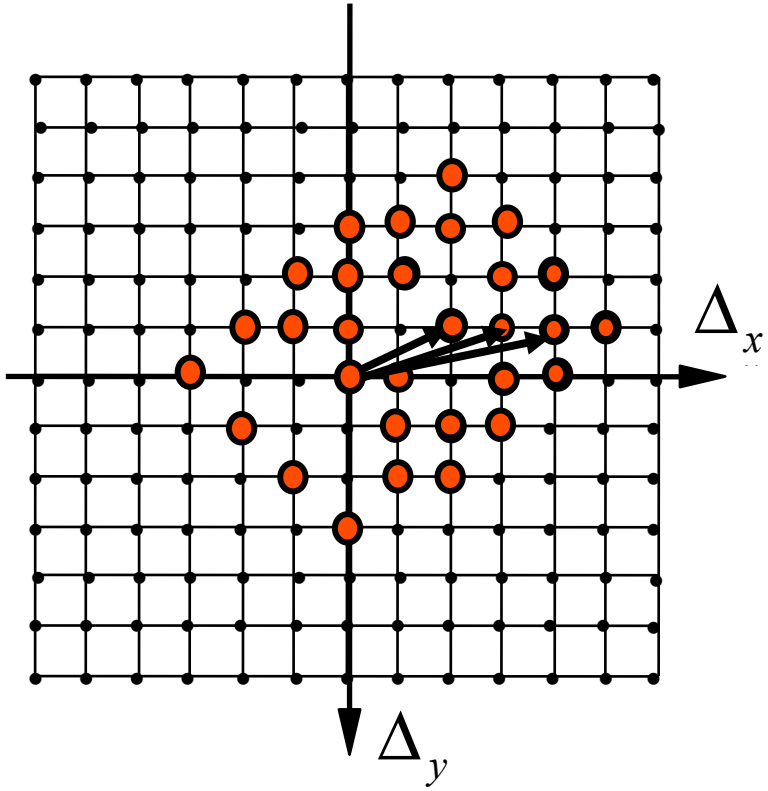
Coarse-to-fine



- Start with coarsely spaced candidate displacements
- Smaller pattern when best match is in the middle
- Stop when desired displacement accuracy is reached

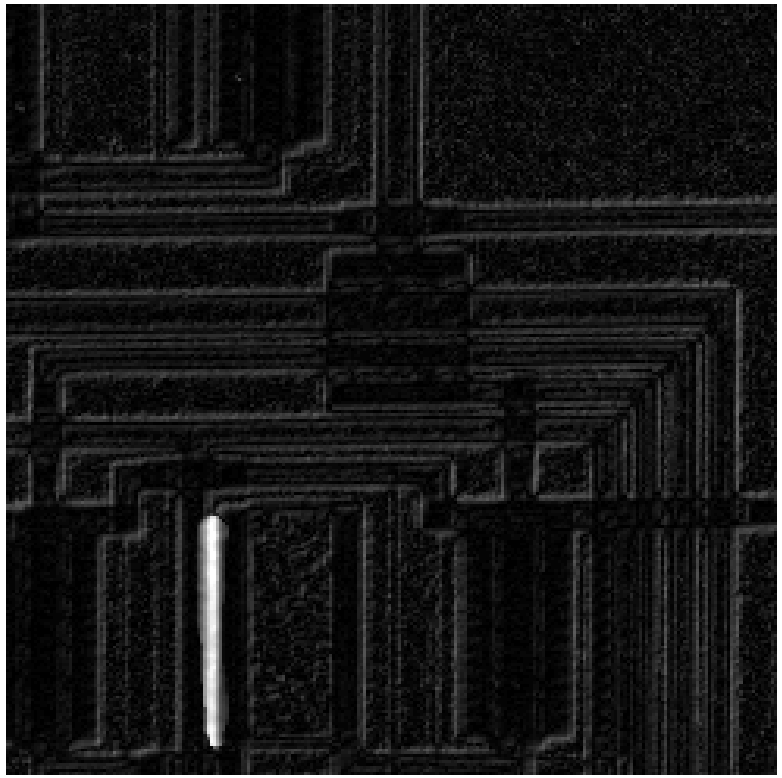
Block matching: search strategies

Diamond search [Li, Zeng, Liou, 1994] [Zhu, Ma, 1997]

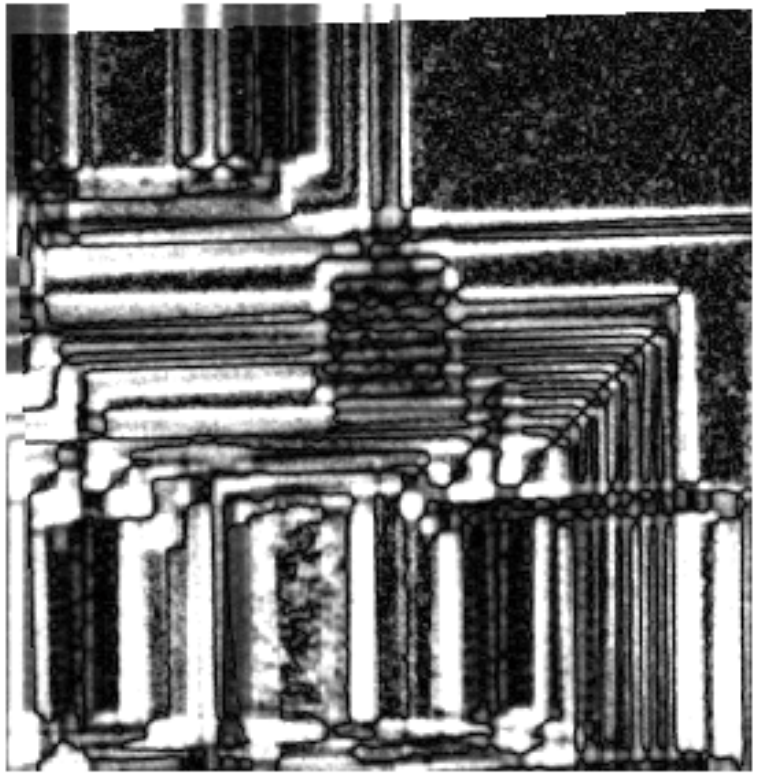


- Start with large diamond pattern at [0,0]
- If best match lies in the center of large diamond, proceed with small diamond
- If best match does not lie in the center of large diamond, center large diamond pattern at new best match

Absolute difference between images

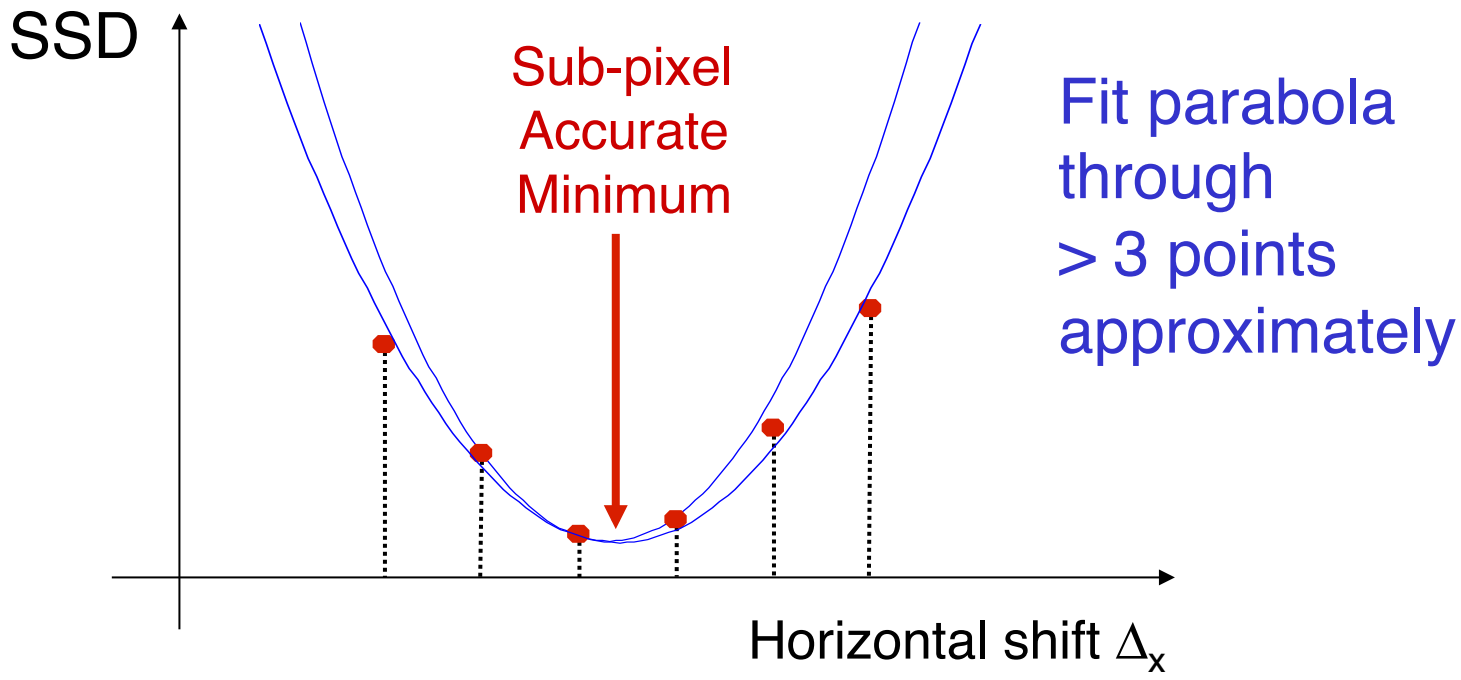


w/ integer-pixel alignment



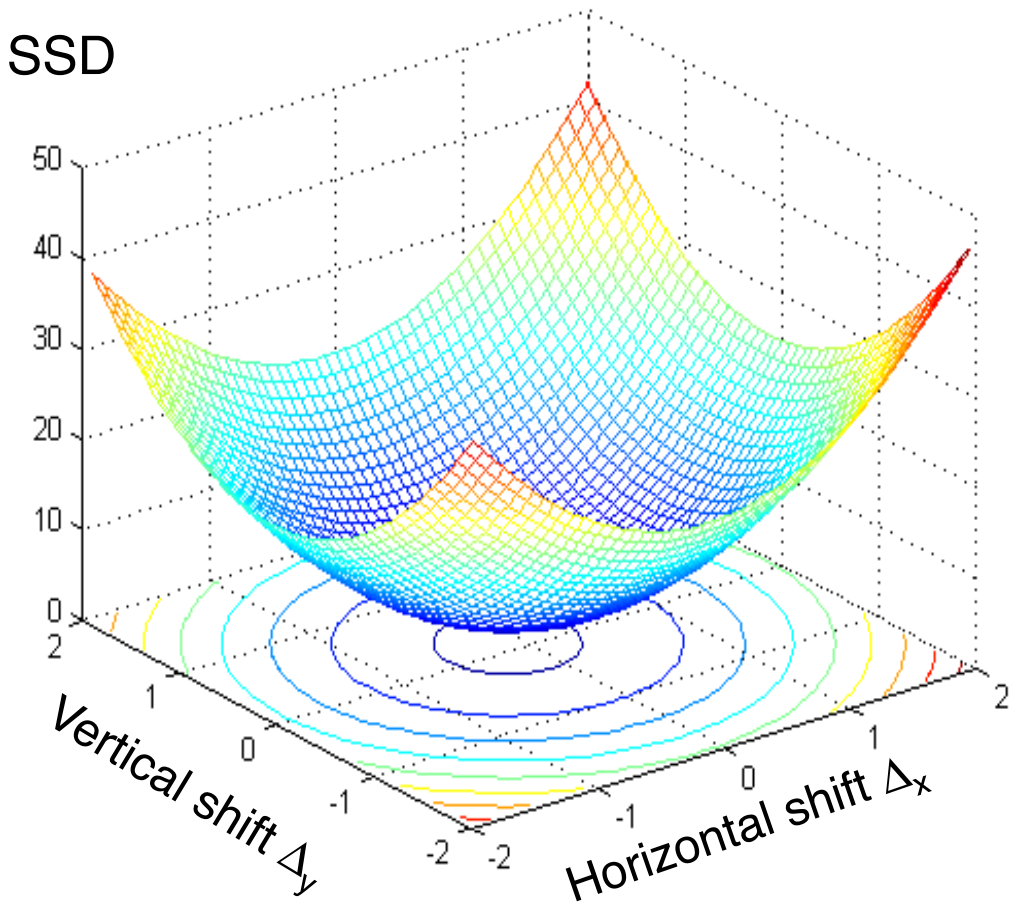
w/o alignment

Interpolation of the SSD Minimum



Fit parabola through > 3 points approximately

2-d Interpolation of SSD Minimum

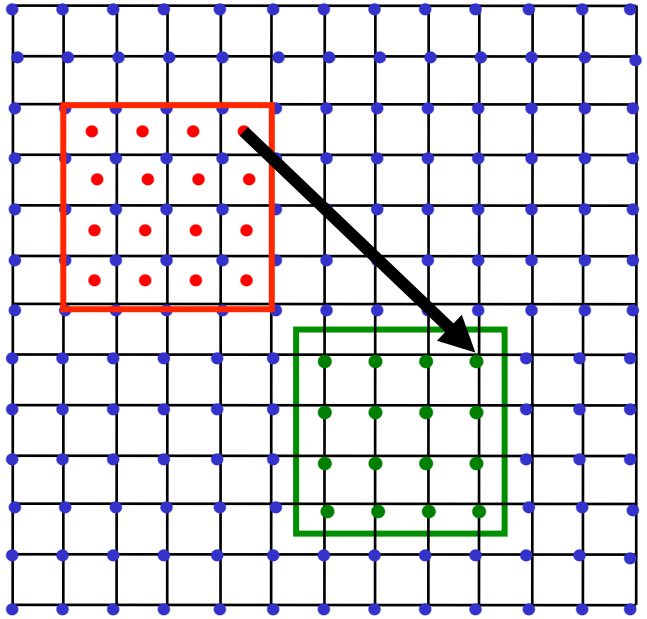


Paraboloid

- Perfect fit through 6 points
- Approximate fit through > 6 points

Sub-pixel accuracy

- Interpolate pixel raster of the reference image to desired sub-pixel accuracy (e.g., by bi-linear or bi-cubic interpolation)
- Straightforward extension of displacement vector search to fractional accuracy
- Example: half-pixel accurate displacements



$$\begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix} = \begin{pmatrix} 4.5 \\ 4.5 \end{pmatrix}$$

Gradient-based displacement estimation

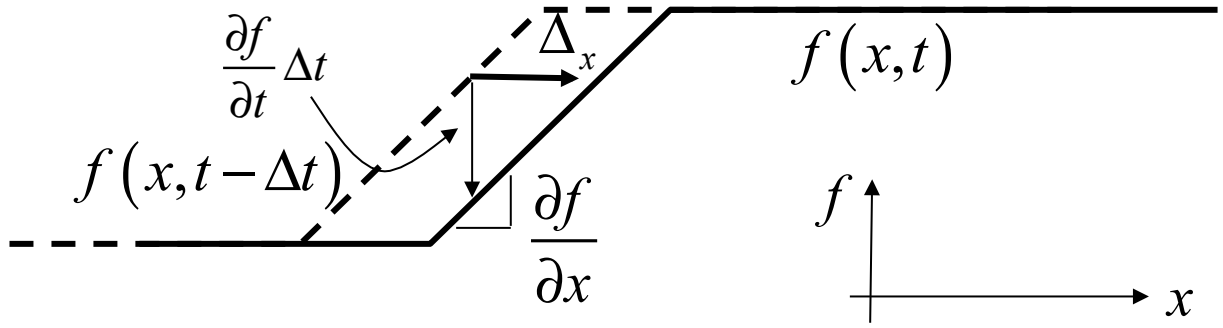
- Consider a space-time continuous signal $f(x,y,t)$
- Optical flow: direction $\begin{pmatrix} \Delta_x(x,y,t) & \Delta_y(x,y,t) & 1 \end{pmatrix}^T$ along which $f(x,y,t)$ is constant

- Derivative in direction of optical flow

$$\begin{pmatrix} \Delta_x(x,y,t) & \Delta_y(x,y,t) & 1 \end{pmatrix} \begin{pmatrix} \frac{\partial f(x,y,t)}{\partial x} & \frac{\partial f(x,y,t)}{\partial y} & \frac{\partial f(x,y,t)}{\partial t} \end{pmatrix}^T = 0$$

$$\boxed{\frac{\partial f}{\partial x}\Delta_x + \frac{\partial f}{\partial y}\Delta_y = -\frac{\partial f}{\partial t}}$$

Illustration of optical flow equation



$$\frac{\partial f}{\partial x}\Delta_x + \frac{\partial f}{\partial y}\Delta_y = -\frac{\partial f}{\partial t}$$

Aperture problem

$$\frac{\partial f}{\partial x}\Delta_x + \frac{\partial f}{\partial y}\Delta_y = -\frac{\partial f}{\partial t}$$

- One equation, two unknowns!
- Can only determine component of Δ_x, Δ_y in direction of brightness gradient (normal flow)
- Must combine at least two observations with brightness gradients of different direction
- Consider Barber Pole Illusion

Lukas-Kanade Algorithm

- Assume single displacement within a measurement window W
- Minimize mean-squared error cost function

$$E(\Delta_x, \Delta_y) = \sum_{x,y \in W} \left(f_x[x,y]\Delta_x + f_y[x,y]\Delta_y + f_t[x,y] \right)^2$$

$f_x[x,y]$ – horizontal image gradient
$f_y[x,y]$ – vertical image gradient
$f_t[x,y]$ – temporal image gradient

$$\frac{\partial}{\partial \Delta_x} E(\Delta_x, \Delta_y) = \sum_{x,y \in W} 2 f_x \left(f_x \Delta_x + f_y \Delta_y + f_t \right) = 0$$

$$\frac{\partial}{\partial \Delta_y} E(\Delta_x, \Delta_y) = \sum_{x,y \in W} 2 f_y \left(f_x \Delta_x + f_y \Delta_y + f_t \right) = 0$$

- Solution

$$\begin{pmatrix} \sum_{x,y \in W} f_x^2 & \sum_{x,y \in W} f_x f_y \\ \sum_{x,y \in W} f_x f_y & \sum_{x,y \in W} f_y^2 \end{pmatrix} \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix} = - \begin{pmatrix} \sum_{x,y \in W} f_x f_t \\ \sum_{x,y \in W} f_y f_t \end{pmatrix}$$

Lucas-Kanade Algorithm (cont.)

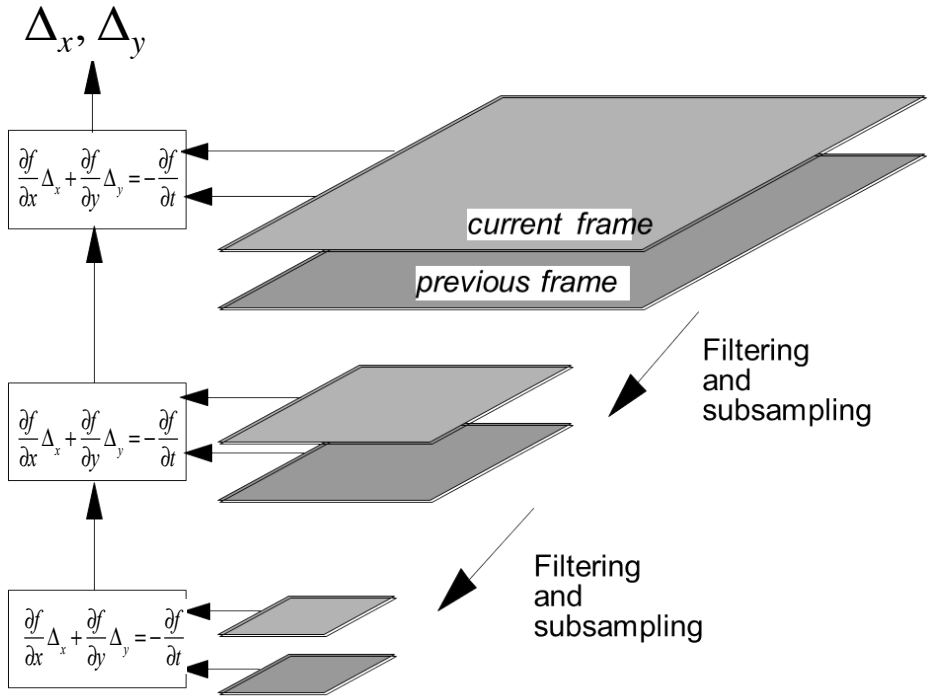
$$\begin{pmatrix} \sum_{x,y\in W} f_x^2 & \sum_{x,y\in W} f_x f_y \\ \sum_{x,y\in W} f_x f_y & \sum_{x,y\in W} f_y^2 \end{pmatrix} = M \qquad \text{Structure matrix!}$$

- Structure matrix M must be inverted to obtain Δ_x, Δ_y
- M can be singular
  - If all gradient vectors are zero (flat area of image)
  - If all gradient vectors point in the same direction (aperture problem)
  - Measurement window W comprises only one pixel
- Not singular for corners, blobs, textured areas …
- Kanade-Lucas-Tomasi (KLT) tracker
  - Compute eigenvalues λ_1, λ_2 of M
  - Only consider windows, where $\min(\lambda_1, \lambda_2) > \theta$

Sampling issues

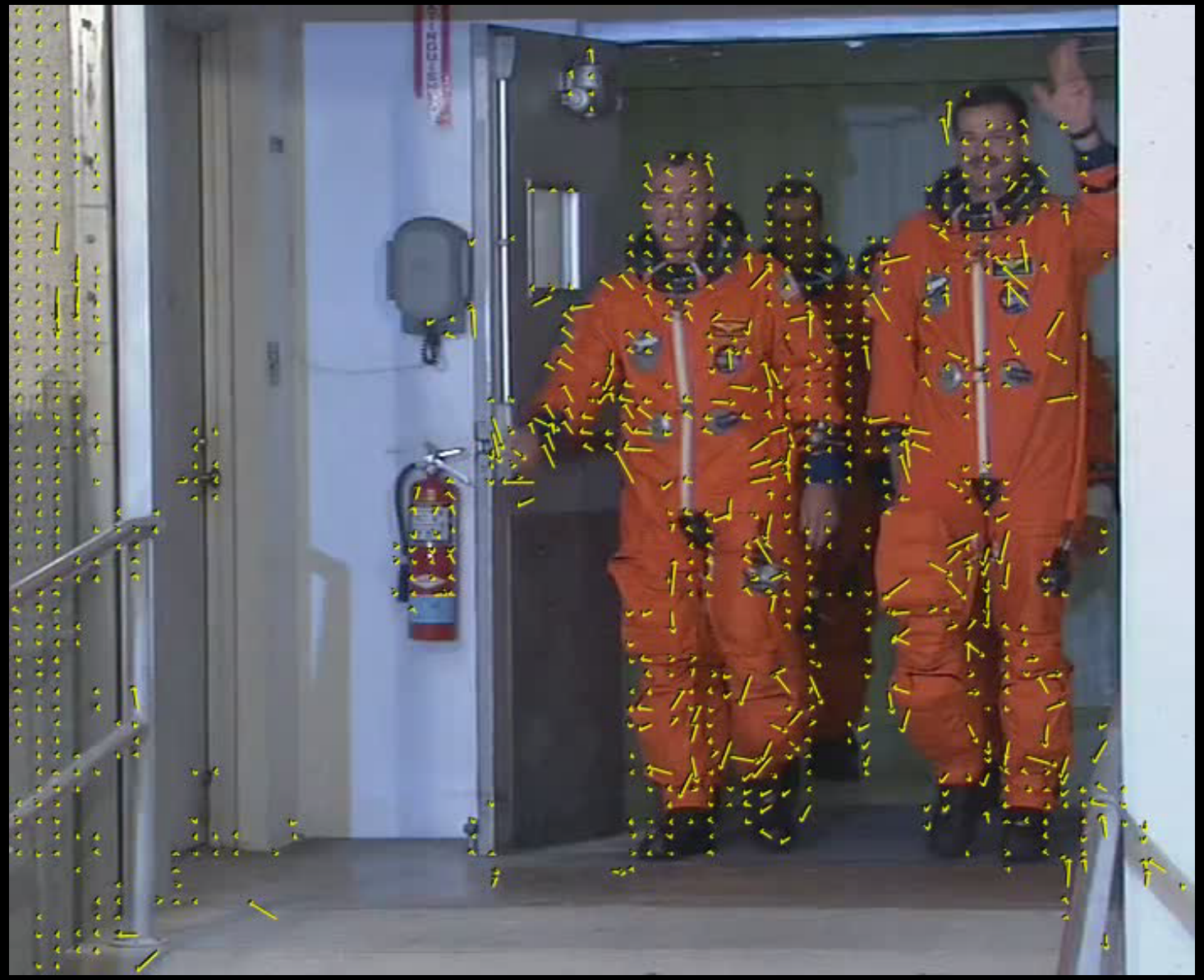
- Image gradients must be approximated by pixel differences, assuming a linear signal in a small spatiotemporal neighborhood
- Linearization inaccurate for displacements > 0.5 pixel

$\Longrightarrow$ Multiple iterations with displacement compensation
Multi-scale implementation

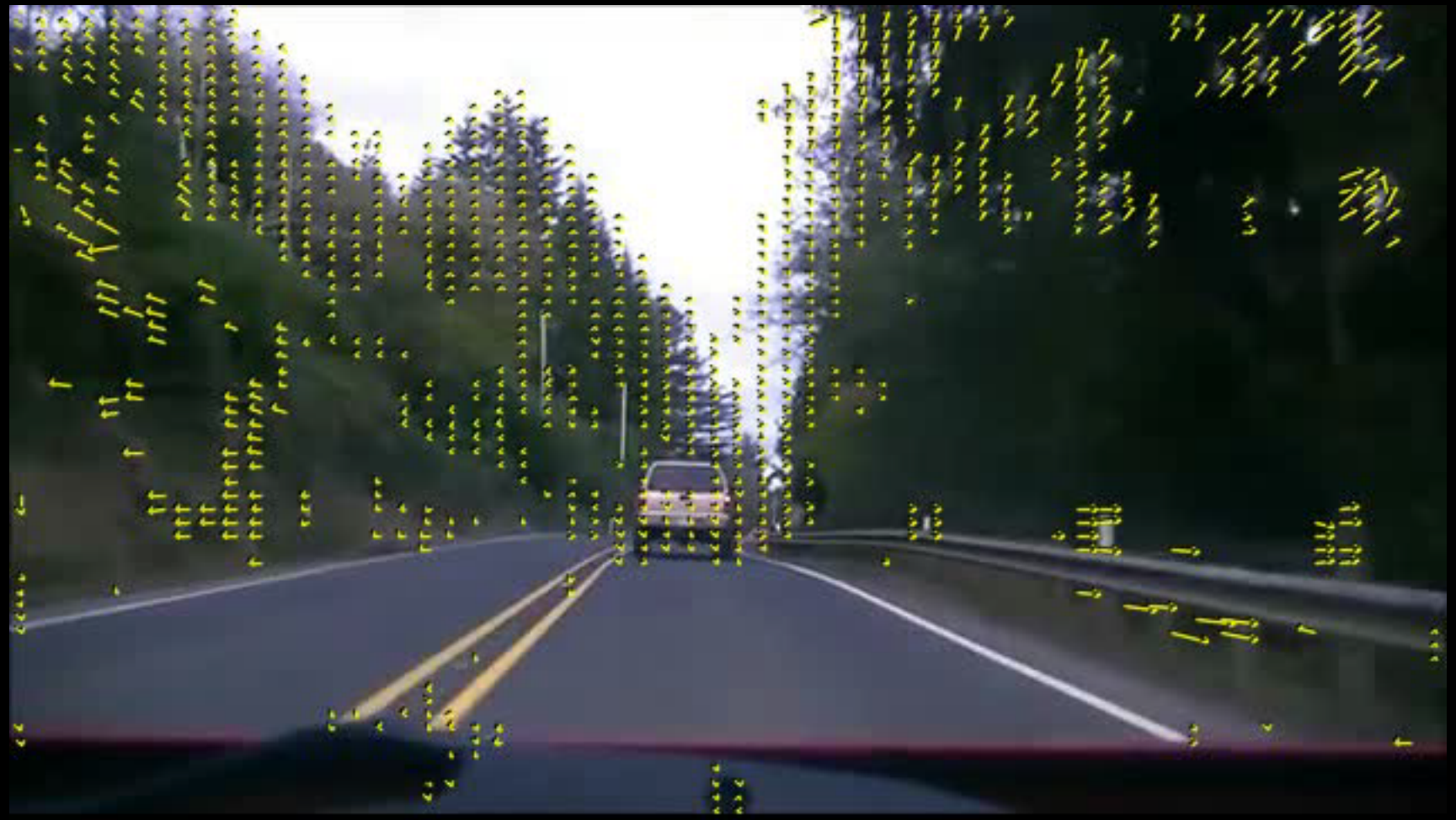


Lucas-Kanade optical flow

Lucas-Kanade optical flow



Lucas-Kanade optical flow



Parametric displacement estimation

- Displacement vector field is represented as

$$\Delta_x(x,y) = a_1\Delta_{x1}(x,y) + a_2\Delta_{x2}(x,y) + a_3\Delta_{x3}(x,y)$$

$$\Delta_y(x,y) = b_1\Delta_{y1}(x,y) + b_2\Delta_{y2}(x,y) + b_3\Delta_{y3}(x,y)$$

- Optical flow constraint

$$\frac{\partial f}{\partial x}\Delta_x + \frac{\partial f}{\partial y}\Delta_y = -\frac{\partial f}{\partial t}$$

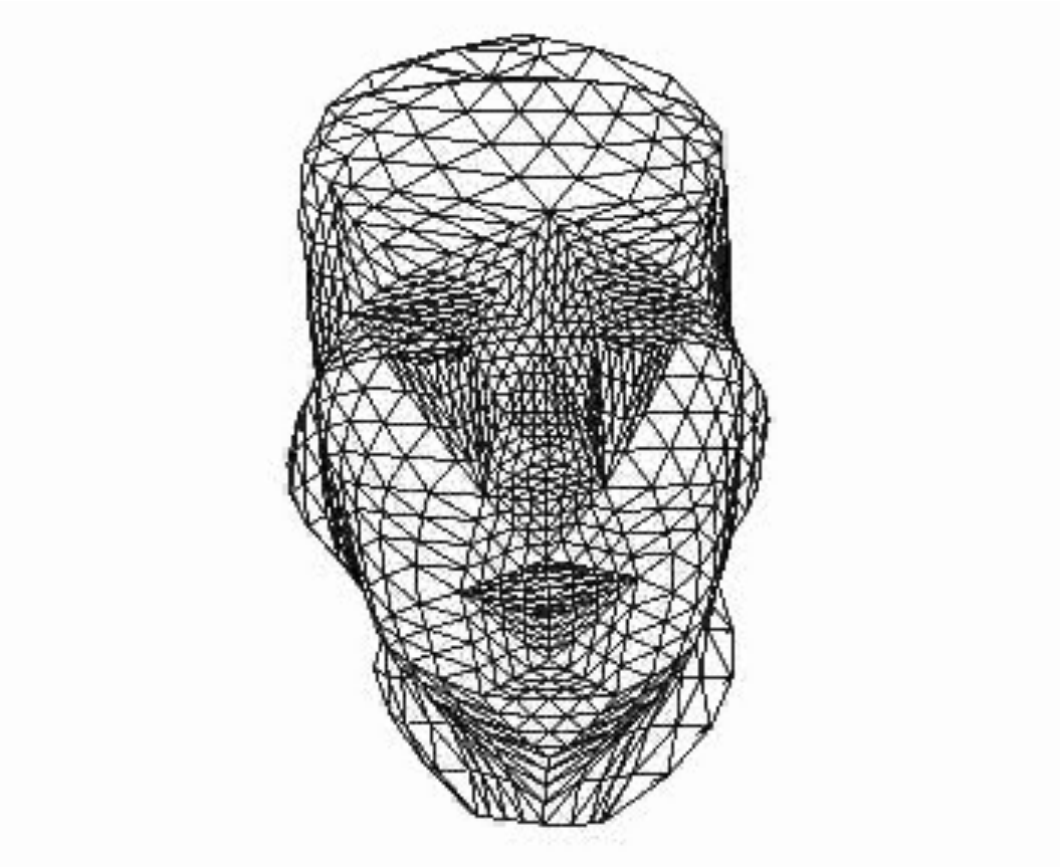
- Combination

$$\frac{\partial f}{\partial x}(a_1\Delta_{x1} + a_2\Delta_{x2} + a_3\Delta_{x3}) + \frac{\partial f}{\partial y}(b_1\Delta_{y1} + b_2\Delta_{y2} + b_3\Delta_{y3}) = -\frac{\partial f}{\partial t}$$

yields a system of linear equations:

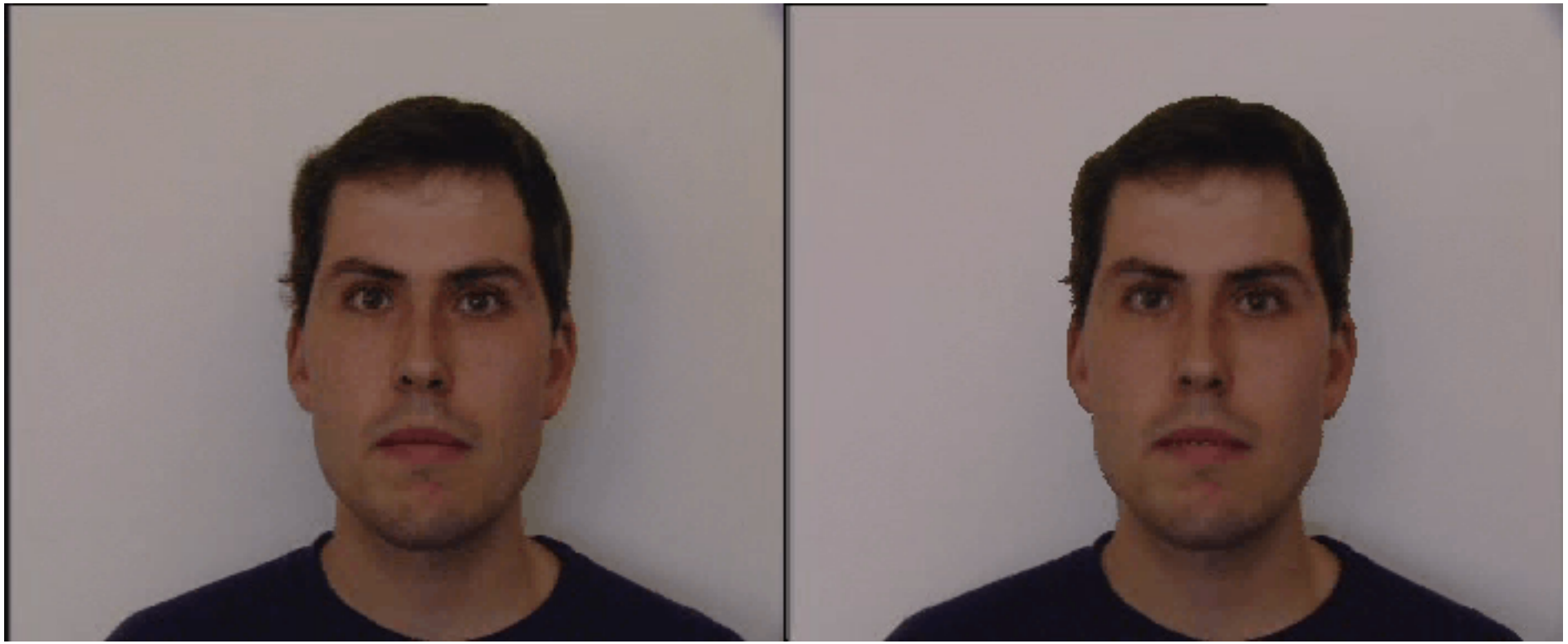
$$\left(\frac{\partial f}{\partial x}\Delta_{x1}, \frac{\partial f}{\partial x}\Delta_{x2}, \frac{\partial f}{\partial x}\Delta_{x3}, \frac{\partial f}{\partial y}\Delta_{y1}, \frac{\partial f}{\partial y}\Delta_{y2}, \frac{\partial f}{\partial y}\Delta_{y2},\right)\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = -\frac{\partial f}{\partial t}$$

Parametric model of human head



Rigid motion: 6 d.o.f.
Facial expression: 18 d.o.f.

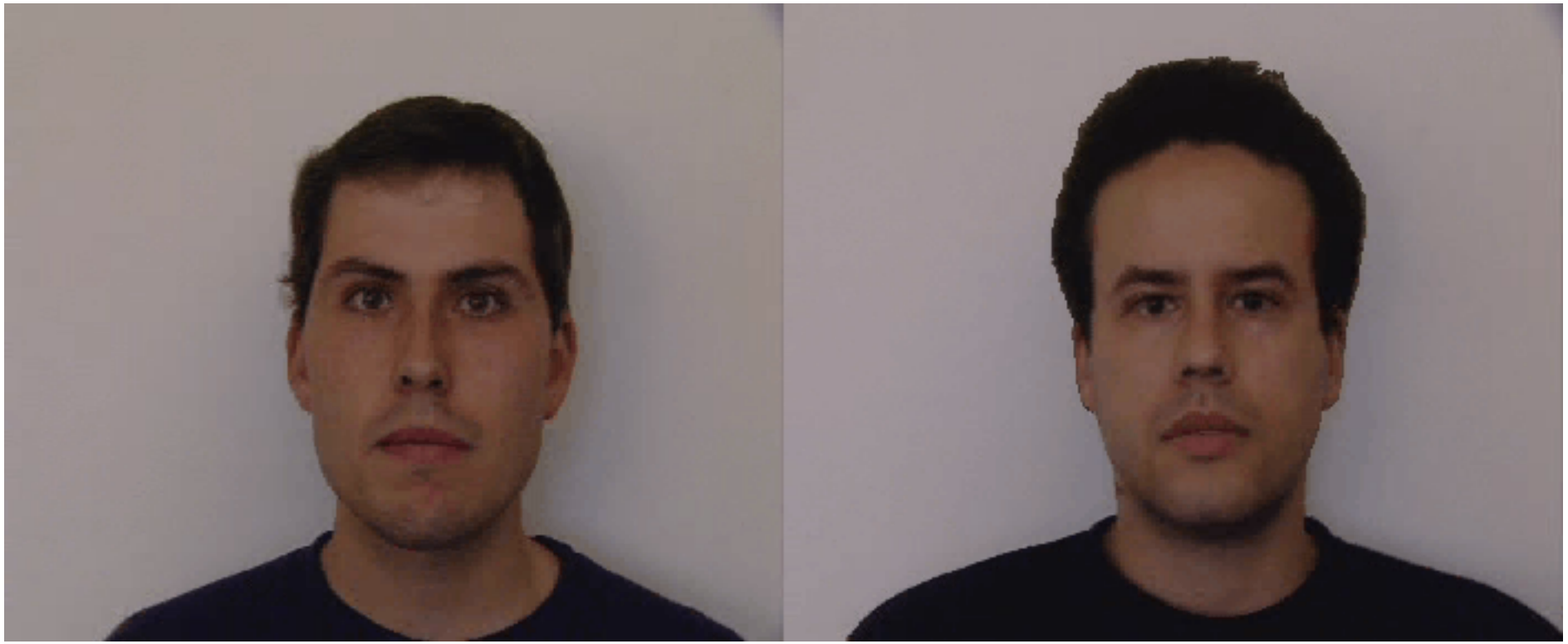
Facial expression tracking



Input video

Avatar

Facial expression tracking



Input video

Avatar

Facial expression tracking

Input video

Avatar