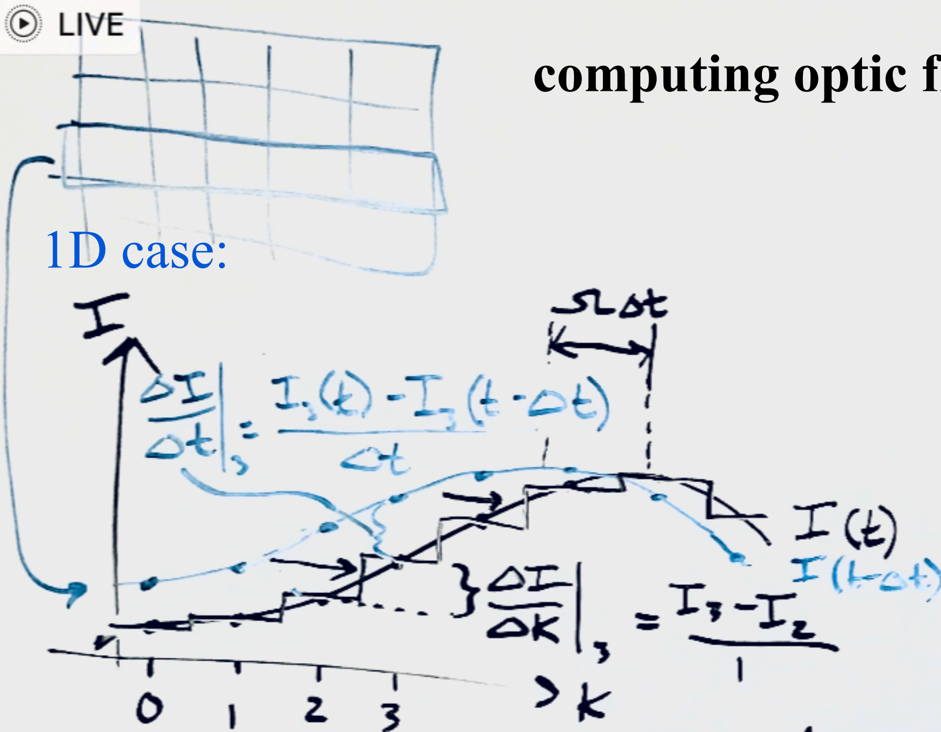
computing optic flow from pixels

1D case:



$$\Omega_m = -\frac{\Delta I/\Delta t}{\Delta I/\Delta k} = \frac{I_t}{I_k} \left[\frac{\text{pixels}}{s} \quad \frac{(k)}{(t)}\right]$$

time derivative (I_t); spatial deriv. (I_k)

- requires $I_k \neq 0$

2D case:

Lucas-Kanade: get x- and y-optic flow using I_x, I_y spatial derivatives along x, y.

$$I_{x_3}\Omega_x + I_{y_3}\Omega_y = I_{t_3} \quad \text{at pixel 3}$$

$$I_{x_4}\Omega_x + I_{y_4}\Omega_y = I_{t_4} \quad \text{at pixel 4}$$

need at least two pixels to solve in least squares sense $Ax = b$, where

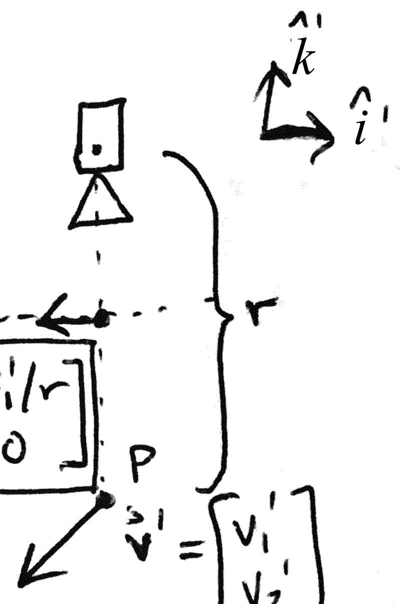
$$A = \begin{bmatrix} I_{x_3} & I_{y_3} \\ I_{x_4} & I_{y_4} \\ I_{x_5} & I_{y_5} \\ \vdots & \vdots \end{bmatrix}, \quad x = \begin{bmatrix} \Omega_x \\ \Omega_y \end{bmatrix}, \quad b = \begin{bmatrix} I_{t_3} \\ I_{t_4} \\ I_{t_5} \\ \vdots \end{bmatrix}$$

python: `x = np.linalg.solve(A, b)`

Def. Optic flow: $\vec{\Omega}$ is velocity of motion projected onto surface 1 unit of distance from camera:

Case 1

camera fixed

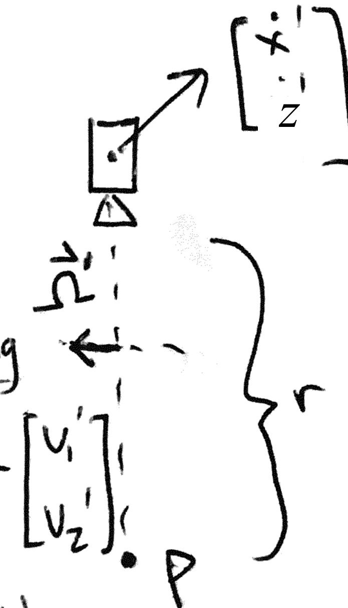


$$\boxed{\vec{\Omega}' = \begin{bmatrix} v_1'/r \\ 0 \end{bmatrix}}$$

$\vec{v}' = \begin{bmatrix} v_1' \\ v_2' \end{bmatrix}$ velocity of point P, in body coords.

Case 2

Consider: P fixed, camera moving at $v' = \begin{bmatrix} \dot{x}' \\ \dot{z}' \end{bmatrix} = -\begin{bmatrix} v_1' \\ v_2' \end{bmatrix}$

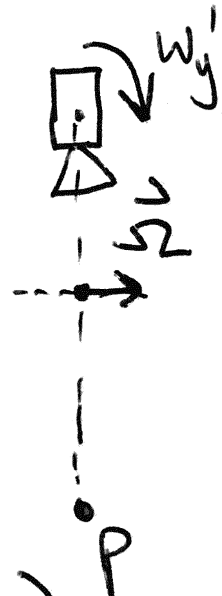


⇒ identical optic flow to case 1:

⇒ $$\boxed{\vec{\Omega}' = \begin{bmatrix} -\dot{x}'/r \\ 0 \end{bmatrix}}$$

Case 3

P fixed, $v' = 0$, $w_y' \neq 0$



$$\vec{\Omega} = \begin{bmatrix} w_y' \\ 0 \end{bmatrix}$$

(idependent of r!)

Case 4 (general case)

P fixed, (assumes world is fixed or slow-moving)

$v' \neq 0$, $w_y' \neq 0$

vector add effects:

$$\boxed{\vec{\Omega} = \begin{bmatrix} w_y' - \frac{\dot{x}'}{r} \\ 0 \end{bmatrix}}$$