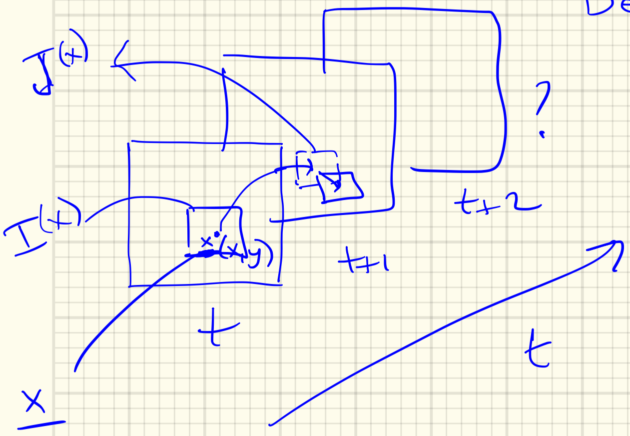
3D Vision

BLG 634E

09.05.2022

Gözde ÜNAL / Motion → Slides uploaded.

Kanade – Lucas – Tomasi Tracker (KLT) [Klette 9.3.2]



Define a warp fn: (geometric transform)

$\underline{w} = \underline{x} + \underline{t}$: translation warp

$\underline{w}(\underline{x}; \underline{p}) \longrightarrow \underline{p} = \underline{t}$

Affine warp: $\underline{w} = \underline{\underline{A}}\,\underline{x} + \underline{t}$

$\left.\begin{array}{l} \underline{p} = \underline{\underline{A}} \rightarrow \\ \underline{p} = \underline{t} \rightarrow \end{array}\right\}$ parameters of the warp are to be estimated.

KLT: a current estimate for $\underline{p}$ exists, iteratively search for the new $\underline{p}$:

Use SSD: $$E(\underline{p}) = \sum_{x} \Big[I\big(\underbrace{w_p(\underline{x})}_{w(\underline{x};\,\underline{p})}\big) - J(\underline{x}) \Big]^2$$

Consider $\underline{p} + \Delta\underline{p}$ (perturb the warp):

$$E(\underline{p} + \Delta\underline{p}) = \sum_{x \in \text{window around } x} \Big[\underbrace{I\big(w_{p+\Delta p}(x)\big)}_{\Uparrow} - J(x) \Big]^2 \Longrightarrow$$

T.S expand $I(w_p(x))$ w.r.t. $p+\Delta p$ Jacobian:

$$\rightarrow I(w_{p+\Delta p}(x)) = I(w_p(x)) + \underline{\Delta p}^T \cdot \underline{\nabla I} \cdot \frac{\partial w_p}{\partial p} + \cancel{O(\Delta p)}$$

HOT

partial derivative of the warp w.r.t. each parameter p_j.

e.g. translation warp: $w_p = \underline{x} + \underline{p}$ $\Rightarrow = \begin{pmatrix} x+p^x \\ y+p^y \end{pmatrix}$: translation

Jacobian: $\frac{\partial w_p}{\partial p} = \begin{bmatrix} \frac{\partial w^x}{\partial p^x} & \frac{\partial w^x}{\partial p^y} \\ \frac{\partial w^y}{\partial p^x} & \frac{\partial w^y}{\partial p^y} \end{bmatrix} \rightarrow$

e.g. for translation, what is Jacobi?

$$J = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I.$$

Insert T.S.E into $E(p+\Delta p)$:

$$\sum_{x \in Window} \left[I(w_p(x)) + \underline{\Delta p}^T \underbrace{\left(\frac{\partial w}{\partial p}\right)}\nabla I - J(x) \right]^2 \rightarrow$$

To calculate the optimum $\underline{\Delta p}$ ↗

$\rightarrow$

→ We calculate derivative w.r.t. $\underline{\Delta p}$ & set it to zero:

$$\underset{2\times 1}{\underline{0}} = \cancel{2} \sum_{x} \left(\frac{\partial w}{\partial \underline{p}} \, \underline{\nabla I} \right)^{T} \left[I(w_p(x)) + \Delta p^{T} \left(\frac{\partial w}{\partial p} \right) \nabla I - J(x) \right]$$

We define [approximate Hessian]

$$\underline{\underline{H}} \stackrel{\sim}{\triangleq} \sum_{x} \left(\frac{\partial w}{\partial p} \nabla I \right)^{T} \left(\frac{\partial w}{\partial p} \nabla I \right)$$

$$\Rightarrow \sum_{x} \left(\frac{\partial w}{\partial p} \nabla I \right) I + \underline{\underline{H}} \, \underline{\Delta p} - \frac{\partial w}{\partial p} \nabla I \, J = 0$$

(★)

$$\boxed{\underline{\underline{\Delta p}}^{(x)} = \underline{\underline{H}}^{-1} \cdot \sum_{x} \frac{\partial w}{\partial p} \nabla I \left(I(w_p(x)) - J(x) \right)}$$

KLT Tracker Algorithm:

- Let $\underline{p}$ be an initial warp :

— While STOP criterion (eg. $\|\underline{\Delta p}\| < \varepsilon$ or $< maxNbIter$ $== FALSE$)

DO :

For the given vector $\underline{p}$;

compute the "optimum shift" $\underline{\Delta p}$ by Eq. (★ prev. page)

Let $\underline{p} = \underline{p} + \underline{\Delta p}$

update the window by the warp

— End While

Note : $\exists$ Fast implementations in OpenCV .