Optical Flow - First Principles of Computer Vision

Michel Liao

December 2023

Contents

1 Introduction

The following notes were taken based off Columbia University Professor Shree Nayar's CV videos on optical flow.

2 Motion Field and Optical Flow

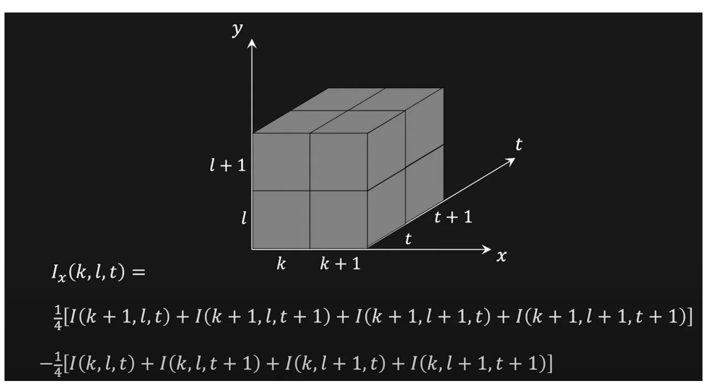
- **Motion field** is the vector field of how things are actually moving.
- **Optical flow** is a change in brightness values that correlate with how the camera perceives things moving.
- Motion field and optical flow aren't necessarily equal.

3 Optical Flow Constraint Equation

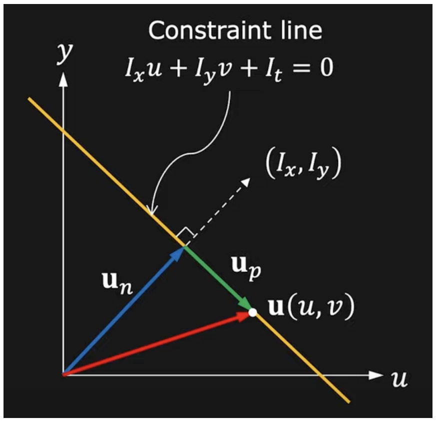
- Given an image taken at time t and time δt, we can track a point that travels from (x, y) to $(x+\delta x, y+\delta y)$ with the displacement vector $\langle \delta x + \delta y \rangle$. This gives us the optical flow vector

$$\langle u, v \rangle = \left\langle \frac{\delta x}{\delta t}, \frac{\delta y}{\delta t} \right\rangle.$$

- To find the optical flow of this point, we need to make some assumptions:
  1. The brightness of an image point remains constant over time.
  $$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t), \tag{1}$$
  where $I(x, y, t)$ is the intensity of a pixel.
     - At least for consecutive images captured in quick succession. Otherwise, this is impossible to solve.
  2. The displacement vector $\langle \delta x, \delta y \rangle$ and the time step δt are very small. Then,
  $$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + I_x \delta x + I_y \delta y + I_t \delta t. \tag{2}$$
     - This allows us to come up with a linear approximation for the intensity $I(x + \delta x, y + \delta y, t + \delta t)$ based off a Taylor series:
     $$f(x + \delta x) \approx f(x) + \frac{\partial f}{\partial x}\delta x + \frac{\partial^2 f}{\partial x^2}\frac{\delta x^2}{2!} + \ldots + \frac{\partial^n f}{\partial x^n}\frac{\delta x^n}{n!}.$$
       * If δx is small, then we approximate $f(x+\delta x) = f(x) + \frac{\partial f}{\partial x}\delta x$.
       * For a function of three variables, we have $f(x+\delta x, y+\delta y, t+\delta t) = f(x, y, t) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y + \frac{\partial f}{\partial t}\delta t$.
- Subtracting (2) from (1) yields
$$I_x \delta x + I_y \delta y + I_t \delta t = 0.$$
- Divide by δt and take the limit as $\delta t \to 0$:
$$I_x \frac{\partial x}{\partial t} + I_y \frac{\partial y}{\partial t} + I_t = 0.$$
- **Optical flow constraint equation:**
$$I_x u + I_y v + I_t = 0,$$
where (u, v) represent optical flow.



–



- Where $\boldsymbol{u}_n$ is the normal flow and $\boldsymbol{u}_p$ is the parallel flow. We know $\boldsymbol{u}_n = \frac{|I_t|}{(I_x^2+I_y^2)}(I_x, I_y)$, but we cannot determine $\boldsymbol{u}_p$.
  - This results in the **aperature problem**, where, locally, we can only determine the normal flow.

4 Lucas-Kanade Method

- **Assumption:** For each pixel, assume the motion field, and therefore optical flow (u, v), is constant within a small neighborhood W.
- From the optical flow constraint equation, we have for all points $(k, l) \in W$,

$$I_x(k, l)u + I_y(k, l)v + I_t(k, l) = 0.$$

- If the size of window W is $n \times n$, then

$$\begin{bmatrix} I_x(1,1) & I_y(1,1) \\ I_x(k,l) & I_y(k,l) \\ \vdots & \vdots \\ I_x(n,n) & I_y(n,n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} I_t(1,1) \\ I_t(2,2) \\ \vdots \\ I_t(n,n) \end{bmatrix},$$

  which is equivalent to $A\boldsymbol{u} = B$, where A is a known matrix of size $n^2 \times 2$, $\boldsymbol{u}$ is an unknown vector of size 2×1, and B is a known vector of size $n^2 \times 1$.
- The least squares solution gives

$$\boldsymbol{u} = (A^T A)^{-1} A^T B.$$

- $A^T A$ must be **invertible**, or $\det(A^T A) \neq 0$.
- Another way to think about it: $A^T A$ must be **well-conditioned.** In other words, a change in the output allows you to estimate a change in the input.
  - Mathematically, if λ_1 and λ_2 are eigenvalues of $A^T A$, then $\lambda_1 > \epsilon$, $\lambda_2 > \epsilon$, $\lambda_1 \geq \lambda_2$ but not $\lambda_1 \gg \lambda_2$.

- Smooth regions (gradients are small) and edges are bad for computing optical flow.
- Textured regions are good.

5 Coarse-to-Fine Flow Estimation

- The optical flow constraint equation assumes δx and δy are small. What if they aren't and you have large motion? (E.g. camera moving)
  - The Taylor series approximation is invalid and thus the constraint equation is.
- We will use a resolution pyramid. We have two images, one at time t, one at time $t + \delta t$. Both have resolution $N \times N$.
- Note that if we lower the resolution of the images to $\frac{N}{2} \times \frac{N}{2}$, the distance a pixel travels will be half the distance it did in the $N \times N$ picture.
- If we lower the resolution enough, eventually the motion will be less than a pixel and our optical flow constraint equation applies.
- **Coarse-to-Fine Estimation Algorithm:**
  - Start with our lowest resolution images and apply optical flow equation to find $(u, v)^{(0)}$.
  - Now, look at the next two images at the next higher resolution. Use $(u, v)^{(0)}$ to warp the image at time t. Use the warped image and the image at time $t + \delta t$ to compute the residual optical flow $\Delta(u, v)^{(1)}$. Then, $(u, v)^{(1)} = (u, v)^{(0)} + \Delta(u, v)^{(1)}$.
  - Repeat the steps above until you find the final optical flow $(u, v)^{(n)}$.
- Another approach uses template matching: Define a small template window T in image I_1 at time t. Using search window S larger than T in image I_2 at time $t + \delta t$, find the match to T using template matching. The difference between the match and the T is the optical flow.
  - Slower than Coarse-to-Fine Estimation when S is big
  - Mismatches are possible