Visual Computing - Notes Week 6

Ruben Schenk, ruben.schenk@inf.ethz.ch

November 11, 2021

8. Optical Flow

8.1 Brightness Constancy

We define **optical flow** as the apparent motion of brightness patterns. Ideally, the optical flow is the *projection* of the three-dimensional velocity vectors of the image.

$I(x, y, t)$ is the brightness at (x, y) at time t. This way we can define:

Brightness constancy assumption:

$$I(x + \frac{dx}{dt}\delta t, y + \frac{dy}{dt}\delta t, t + \delta t) = I(x, y, t)$$

Optical flow constraint equation:

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

8.2 The Aperture Problem

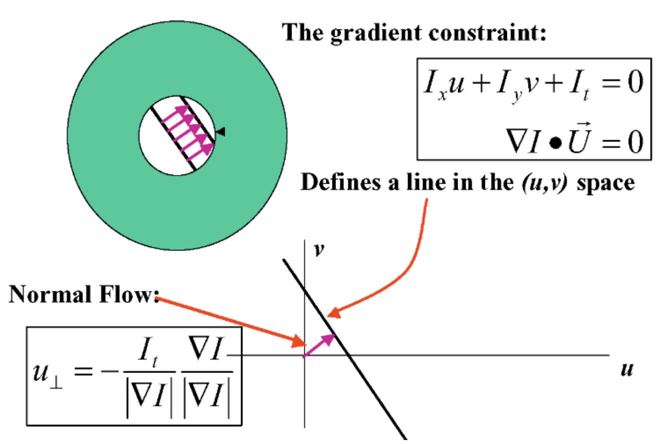
If we do the following substitution:

$$u = \frac{dx}{dt}, v = \frac{dy}{dt}, I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}$$

we arrive at the following equation:

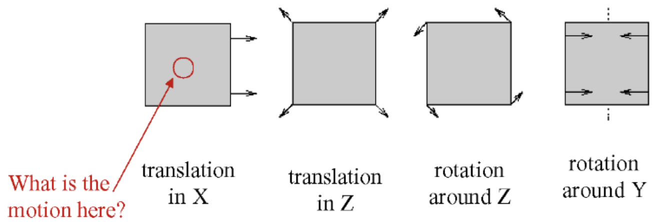
$$I_x u + I_y v + I_t = 0$$

However, this is one equation in two unknowns, which is known as the **Aperture Problem.**



Optical flow is not always well-defined! We can compare the different kinds of flows:

- *Motion Field:* Projection of 3D motion field
- *Normal Flow:* Observed tangent motion
- *Optical Flow:* Apparent motion of the brightness pattern, hopefully equal to motion field



Scene Flow: $\longrightarrow$
Normal Flow: undef
Optic Flow: ?, probably 0

8.3 Regularization

We might use the **Horn & Schuck algorithm,** described as follows:

- We have an additional smoothness constraint

$$e_s = \int\int((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy$$

- besides our Optical Flow constraint equation term:

$$e_c = \int\int(I_x u + I_y v + I_t)^2 dxdy$$

The goal is to *minimize* $e_s + \lambda e_c$!s

8.4 Lucas-Kanade

Assume a single velocity for all pixels within an image patch

$$E(u,v) = \sum_{x,y\in\Omega}\left(I_x(x,y)u + I_y(x,y)v + I_t\right)^2$$

$$\frac{dE(u,v)}{du} = \sum 2I_x\left(I_x u + I_y v + I_t\right) = 0$$

Solve with:

$$\frac{dE(u,v)}{dv} = \sum 2I_y\left(I_x u + I_y v + I_t\right) = 0$$

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

On the LHS: sum of the 2x2 outer product tensor of the gradient vector

$$\left(\sum \nabla I \nabla I^T\right)\vec{U} = -\sum \nabla I I_t$$

With respect to singularities and the aperture problem, we proceed as follows. Let:

$$M = \sum(\nabla I)(\nabla I)^T \quad \text{and} \quad b = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

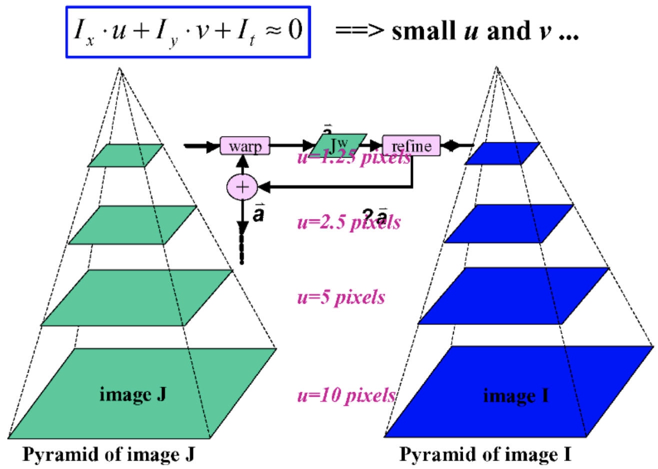
Then, the algorithm is as simple as solving at each pixel for U in $MU = b$. M is *singular* if all gradient vectors point in the same direction.

8.5 Coarse To Fine

The local gradient method has some limitation:

- Fails when the intensity structure within the window is poor
- Fails when the displacement is large (typical operating range is motion of 1 pixel per iteration!)

We can combat this with a Pyramid or "*Coarse-to-fine*" estimation:



8.6 Parametric Motion Models

Global motion models offer:

- more constrained solutions than smoothness models (Horn-Schunck)
- integration over a larger area than a translation-only model can accommodate (Lucas-Kanade)