

Optical Flow

Namrata Vaswani, namrata@iastate.edu

These notes are still under preparation. Please email me if you find any mistakes and typos.

Most of the material here is based on [1], Chapter 5 of [2] and Chapter 15 of [3].

Optical Flow (OF) is the 2D motion field, $u(x, y)$, $v(x, y)$, for each point (x, y) in an image. First note that the OF does not directly tell us about the real 3D motion of the object, it only gives us the projection of the motion in 2D. OF vectors along with a camera model assumption can be used to estimate the 3D motion (and depth) of objects and this is called “Structure from Motion”.

There are two types of algorithms that can be used to estimate optical flow. Given a set of corresponding points in 2 images, the OF at these points is given by the displacement vector between these points.

To estimate OF at all points in an image (dense OF), we use the “intensity constancy” assumption which says that the intensity of any point on any object remains constant with time (even though the object itself moves), i.e.

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) \quad (1)$$

Assuming small motion between consecutive frames (small $\Delta x, \Delta y$), the above can be expanded using first order Taylor series to get

$$I_x u + I_y v + I_t = 0 \quad (2)$$

where $I_x \triangleq \frac{\partial I}{\partial x}$ and so on. Also $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$. This is called the optical flow equation and forms the basis of a large number of first order algorithms (where the small motion assumption is valid). We discuss 2 algorithms here.

A. Horn and Schunk

First notice, that (2) gives only one equation for 2 variables (or if there are N pixels in the image, it gives N equations for $2N$ variables) and hence requires additional constraints to make it well-posed. Horn and Schunk [?] proposed to use the *smoothness constraint*, i.e. *nearby points on an object move with similar x and y velocities, or that $\|\nabla u\|^2, \|\nabla v\|^2$ is small*. Thus they proposed to find u, v to minimize the following energy functional

$$E(u, v) = \int_{\Omega} [(I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)] dx dy \quad (3)$$

where λ controls the weight given to the smoothness constraint and Ω denotes the image domain. Also, we assume that u, v are known (zero) at the boundaries of the image domain. Now E is a functional, i.e. it is a function of functions, u, v both of which are functions of x, y . Since E is a function of u, v and their derivatives, we use Calculus of Variations to perform the minimization. Defining the Lagrangian as the integrand in (3)

$$L(u(x, y), v(x, y)) \triangleq [(I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)] \quad (4)$$

and using Calculus of Variations, we get

$$\begin{aligned} \nabla_u E &= \frac{\partial L}{\partial u} - \frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} \\ &= I_x^2 u + I_x I_y v - I_x I_t - \lambda \nabla^2 u \end{aligned} \quad (5)$$

$$\nabla_v E = I_x I_y u + I_y^2 v - I_y I_t - \lambda \nabla^2 v \quad (6)$$

where $\nabla^2 u = u_{xx} + u_{yy}$. A necessary condition for E to be minimized is given by $\nabla_u E = 0, \nabla_v E = 0$. This is called the Euler-Lagrange equation and can be re-arranged to give

$$\begin{aligned} I_x^2 u + I_x I_y v &= \lambda \nabla^2 u - I_x I_t \\ I_x I_y u + I_y^2 v &= \lambda \nabla^2 v - I_y I_t \end{aligned} \quad (7)$$

To solve the above for discrete pixels, the Laplacians can be approximated by their discrete central approximation, $\nabla^2 u(x, y) \approx 4(\bar{u}(x, y) - u(x, y))$ where $\bar{u}(x, y) \triangleq \frac{1}{4}(u(x-1, y) + u(x+1, y) + u(x, y-1) + u(x, y+1))$. This gives $2N$ equations in $2N$ variables: defining $\alpha = 4\lambda$, we get

$$\begin{aligned} (I_x^2 + \alpha)u + I_x I_y v &= \alpha \bar{u} - I_x I_t \\ I_x I_y u + (I_y^2 + \alpha)v &= \alpha \bar{v} - I_y I_t \end{aligned} \quad (8)$$

This can in principle be solved directly, but will require inverting a very large, $2N \times 2N$, matrix. A faster method is to obtain an iterative solution using the Gauss-Seidel method which takes advantage of the sparseness of the matrix. Thus to summarize the algorithm:

- 1) At iteration $n = 0$, start with an initial guess of u, v
- 2) Update using:

$$\begin{aligned} u^{n+1} &= \bar{u}^n - I_x \frac{(I_x \bar{u}^n + I_y \bar{v}^n + I_t)}{\alpha + I_x^2 + I_y^2} \\ v^{n+1} &= \bar{v}^n - I_y \frac{(I_x \bar{u}^n + I_y \bar{v}^n + I_t)}{\alpha + I_x^2 + I_y^2} \end{aligned} \quad (9)$$

3) Stop when E does not decrease much or equivalently $I_x \bar{u}^n + I_y \bar{v}^n + I_t$ is small.

Some things to notice:

- 1) Notice that the update for the $(n + 1)^{th}$ iteration of u depends only on values of u computed in the n^{th} iteration and hence *the above algorithm can be parallelized*.
- 2) Uniform intensity regions (where I_x, I_y, I_t are small) are filled using averages from the edges where there is significant intensity variation. In other words OF for uniform regions gets estimated only using the smoothness constraint.
- 3) In regions where image gradient $\|\nabla I\|^2 = I_x^2 + I_y^2$ is large, the effect of α is negligible, i.e. the OF equation is mainly used in these regions.
- 4) Notice from (2) that the OF equation gives no estimate of the OF in directions perpendicular to the image gradient (along edges). Thus in general one gets accurate OF estimates using the above algorithm when nearby image pixels have different directions of the image gradient.
- 5) If a uniform intensity disc is rotating about its center, then the estimate of optical flow using any algorithm will be ZERO because I_t will be zero. Same for a rectangular object aligned with the x-y plane and with an infinitely long y edge, moving along the y axis.
- 6) If there is intensity variation with time introduced due to illumination change, that will wrongly appear as non-zero optical flow even if the scene is not moving.

Two problems with the above method are [2]:

- 1) It smooths across motion edges: When an object is moving against a background, there is no reason to impose smoothness of optical flow across the boundary between the object and background. But the above algorithm does. One solution to this is to impose a weighted smoothness criterion, i.e. replace $\lambda \|\nabla u\|^2$ by $(\nabla u)^T W (\nabla u)$ where $W = [(\nabla I)(\nabla I)^T]^{-1}$ and similarly for $\lambda \|\nabla v\|^2$
- 2) It uses OF equation even for occluded regions. This happens only when motion is more than one pixel per unit time. For a region that was occluded in the first frame but is visible in the second frame, one wrongly gets a large estimate of I_t even though the region may not have undergone any motion. One solution to this problem is to turn off the OF constraint and only use the smoothness constraint for occluded regions (set λ to a very large value in these regions). Occluded region are the interior of regions of high I_t . A second solution is to use a hierarchical method (discussed below) to estimate optical flow, so that at each level, motion is not more than one pixel per unit time.

B. Least Mean Squares: Lucas-Kanade

This assumes that OF at neighboring pixels is equal and obtains a least mean squares estimate (LMSE) of OF. For example let the optical flow be constant in a 3×3 neighborhood of x, y , then we have 9 equations in 2 variables and one can get an LMSE estimate of OF. Read Chapter 5, section 5.3.3 of [2]. More averaging can be performed by also averaging over a sequence of frames. If frames are coming in sequentially, one can use other least squares methods such as Recursive Least Squares (see Least Squares handout).

Now the assumption of constant OF for neighboring pixels is not true for rotatory motion. Thus this method approximates rotatory motion by 3×3 blocks of translation.

C. Hierarchical Approaches

In many cases the small motion assumption is not satisfied for consecutive frames at the original resolution. But if one were to reduce the resolution r times (low pass filter followed by downsampling), the motion between consecutive frames also reduces r times. In these cases, using a hierarchical first order method (either Horn-Schunk or LMSE) gives more accurate results. The hierarchical OF algorithm for 2 levels is:

- 1) Reduce resolution of frames I_0 and I_1 , r times and estimate OF.
- 2) Increase resolution (upsample and interpolate) of the estimated OF r times to get \hat{u}, \hat{v} .
- 3) Estimate $\hat{I}_1(x, y) = I_0(x + \hat{u}(x, y), y + \hat{v}(x, y))$
- 4) Estimate OF between \hat{I}_1 and I_1 , call it $\hat{\hat{u}}, \hat{\hat{v}}$.
- 5) Then total OF between I_0 and I_1 is $\hat{u} + \hat{\hat{u}}, \hat{v} + \hat{\hat{v}}$.

D. Second order and other approaches

To get a more accurate estimate of 2D motion, one uses a second order Taylor series approximation of (1) and then estimates both optical flow velocity and acceleration. Some other approaches assume a parametric model for OF and estimate its parameters assuming intensity constancy. Rigid 3D motion of objects projected using an orthographic camera model, can be described by a 6 dimensional affine transformation on each pixel. Its parameters can be learnt again using a least squares approach.

REFERENCES

- [1] B.K.P. Horn and B.G.Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [2] A.M.Tekalp, *Digital Video Processing*, Prentice Hall, 1995.

- [3] M. Sonka, V. Hlevac, and R. Boyle, *Image Processing, Analysis and Computer Vision*, Brooks/Cole, Pacific Grove, CA, 1999.