LECTURE 11 : OPTIC FLOW

[1] Introduction

What is optical flow?

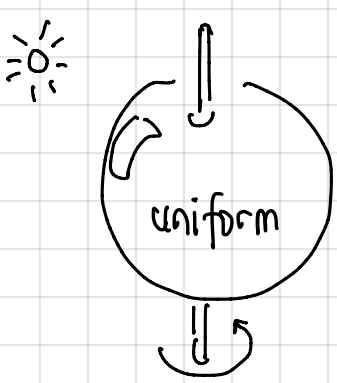
The apparent motion of brightness patterns

↓

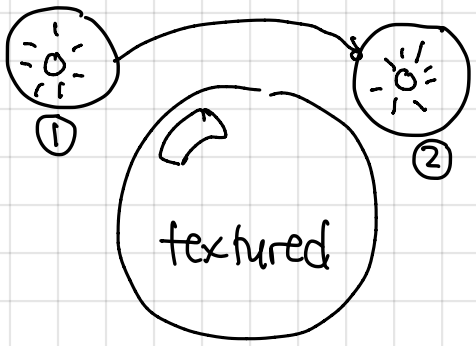
It consists of motion direction & magnitude

It's not the same as motion flow (= the actual motion / velocity vector in the real world)

The differences between optical flow & motion flow:

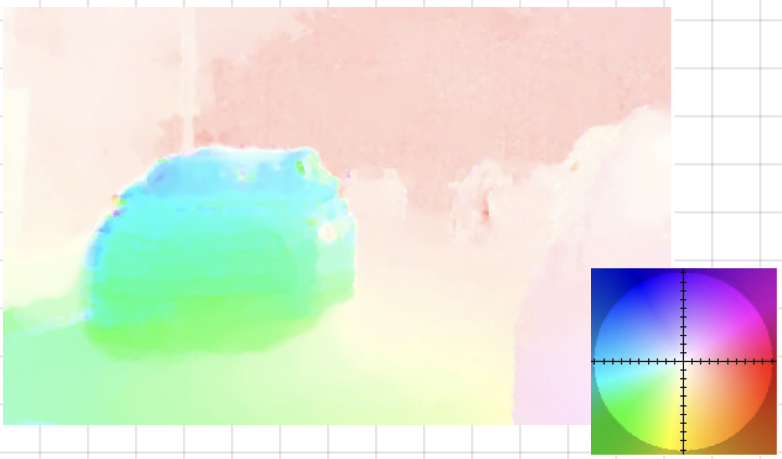


Non-zero motion field
Zero optical flow



Nonzero optical flow
Zero motion field

Many applications of optical flow: object tracking, motion analysis, collision detection, image registration, obstruction detection, motion segmentation, etc.



[2] Lucas-Kanade Optic Flow

Brightness Constancy Constraint (BCC): Common assumption

The brightness of pixels representing an object is constant throughout frames.

Intensity/brightness of a pixel: $I(x, y, t)$

Under the BCC:

$$I(x, y, t) = I(x+u, y+v, t+1)$$

where: (u, v) is the optic flow vector

t inclicates the frame index

Applying the chain rule:

$$\frac{dI(x,y,t)}{dt} = \frac{dI}{dx}\frac{dx}{dt} + \frac{dI}{dy}\frac{dy}{dt} + \frac{dI}{dt} = 0$$

$$u = dx/dt \quad ; \quad v = dy/dt$$

in discrete: $\frac{\Delta x}{\Delta t} = \frac{x_{t+1} - x_t}{(t+1) - t} = \frac{(x_t + u) - x_t}{1} = u$

Hence:

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

the BCC

Lucas-Kanade's objective function:

$$\min E(u, v) = \min \left(I_x u + I_y v + I_t \right)^2$$

Minimization:

① $\frac{\partial E(u,v)}{\partial u} = 2\left(I_x u + I_y v + I_t\right) I_x = 0$

② $\frac{\partial E(u,v)}{\partial v} = 2\left(I_x u + I_y v + I_t\right) I_y = 0$

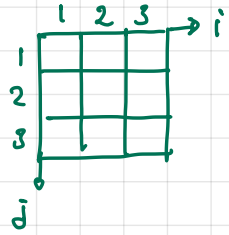
$$\begin{bmatrix} I_x I_x & I_y I_x \\ I_x I_y & I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} I_x I_x & I_y I_x \\ I_x I_y & I_y I_y \end{bmatrix}^{-1}}_{A^{-1}} \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}$$

To be invertible, $\det(A) \neq 0$

$\det(A) = I_x^2 I_y^2 - (I_x I_y)(I_y I_x) = 0 \rightarrow A$ not invertible

Solution: Instead a pixel operation, we employ a patch operation (e.g. 3x3, 5x5, 7x7, etc)



Considering a 3x3 patch:

$$I_{x_{1,1}}^2 u + I_{y_{1,1}} I_{x_{1,1}} v = - I_{t_{1,1}} I_{x_{1,1}}$$
$$I_{x_{1,2}}^2 u + I_{y_{1,2}} I_{x_{1,2}} v = - I_{t_{1,2}} I_{x_{1,2}}$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$I_{x_{3,3}}^2 u + I_{y_{3,3}} I_{x_{3,3}} v = - I_{t_{3,3}} I_{x_{3,3}}$$

$+$

$$\sum_{i,j} I_{x_{ij}}^2 u + \sum_{i,j} I_{y_{ij}} I_{x_{ij}} v = - \sum_{ij} I_{t_{ij}} I_{x_{ij}}$$

Finally:

$$\begin{bmatrix} \sum_{i,j} I_{x_{i,j}}^2 & \sum_{i,j} I_{x_{i,j}} I_{y_{i,j}} \\ \sum_{i,j} I_{y_{i,j}} I_{x_{i,j}} & \sum_{i,j} I_{y_{i,j}}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_{i,j} I_{t_{i,j}} I_{x_{i,j}} \\ -\sum_{i,j} I_{t_{i,j}} I_{y_{i,j}} \end{bmatrix}$$

Based on the equation, (u, v) is sparse & local flow field; since for many regions are texturefless:

$$\sum_{i,j} I_{x_{i,j}}^2 = \sum_{i,j} I_{y_{i,j}}^2 = \sum_{i,j} I_{y_{i,j}} I_{x_{i,j}} = 0$$

In practice: $\begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} G * I_x^2 & G * I_x I_y \\ G * I_x I_y & G * I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} G * I_x I_t \\ G * I_y I_t \end{bmatrix}$

Note: G is a Gaussian kernel with a certain window size (e.g. 5x5), where $(G * I_x^2)(G * I_y^2) \neq (G * I_x I_y)^2$

Example:

Q: How to compute a dense & global optic flow?

↓

(every pixel)

[3] Horn-Schunck Optical Flow

Objective function:

$$E(u,v) = \underbrace{(I_x u + I_y v + I_t)^2}_{\text{data term}} + \alpha \underbrace{\left(\|\nabla u\|^2 + \|\nabla v\|^2 \right)}_{\text{smoothness prior term}}$$

where: $\nabla u = \begin{bmatrix} \partial u/\partial x \\ \partial u/\partial y \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$

$$\|\nabla u\|^2 = \left(\sqrt{u_x^2 + u_y^2} \right)^2 = u_x^2 + u_y^2$$

Global objective function: $J(u,v) = \int E(u,v)\, dx\, dy = 0$

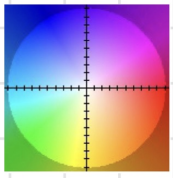
Goal: To find (u,v) for every pixel that minimizes the entire image.

This is exactly the same problem studied in calculus of variations: a field dealing with optimizing a functional (= a function of a function). $E(u,v)$ is a function of (u,v) functions.

One of the methods to solve the global minimization:

Euler-Lagrange Equation

Example of dense flow:



[4] Euler - Lagrange Equation

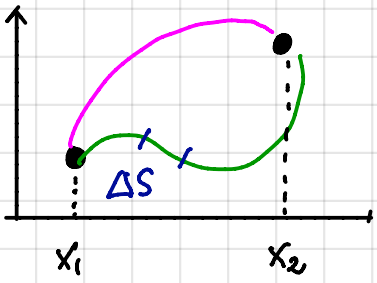
General form: $J = \int F(t, \bar{y}, \bar{y}')\, dt$; $\bar{y}' = d\bar{y}/dx$

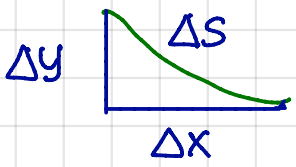
J will have a stationary value if : $\frac{\partial F}{\partial \bar{y}} - \frac{\partial}{\partial t}\left(\frac{\partial F}{\partial y'}\right) = 0$

Euler- Lagrange equation is key in calculus of variations.

Example :

The shortest distance between 2 points :
(We know it will be a straight line,
but how can we prove it ?)





$$\Delta S \cong \sqrt{\Delta x^2 + \Delta y^2}$$

$$dS = \sqrt{dx^2 + dy^2} \rightarrow dS = \frac{dx}{dx}\sqrt{dx^2 + dy^2}$$

$$S = \int \sqrt{1 + (y')^2}\, dx \leftarrow \int ds = \int \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$$

To minimise : $S = \int F(x, y')\, dx$; $F = \sqrt{1 + (y')^2}$

where according to the Euler- Lagrange eq. :

$$\boxed{\frac{\partial F}{\partial y} - \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial y'}\right) = 0}$$

(1) $\frac{\partial F}{\partial y} = 0$, since function S is independent from y .

(2) $\frac{\partial F}{\partial y'} = \frac{d}{\partial y'}\left(\sqrt{1 + (y')^2}\right) = \frac{1}{2}\left(1 + y'\right)^{-1/2} 2y' = \frac{y'}{\sqrt{1 + (y')^2}}$

Hence: $\frac{\partial}{\partial x}\left(\frac{y'}{\sqrt{1+(y')^2}}\right) = 0$

Recall: Ordinary Differential Equation:

$$\frac{dy}{dx} = 0$$

$$\int \frac{dy}{dx}\,dx = \int 0\,dx$$

$$\int dy = C_1$$

$$y + C_2 = C_1$$

$$y = C$$

because the derivative of C w.r.t. x is 0

Therefore:

$$\frac{\partial}{\partial x}\left(\frac{y'}{\sqrt{1+(y')^2}}\right) = 0 \rightarrow \frac{y'}{\sqrt{1+(y')^2}} = C$$

$$y' = C\sqrt{1+(y')^2} = \sqrt{C^2 + C^2(y')^2}$$

$$(y')^2 = C + C(y')^2$$; whether it's C or C^2, it doesn't matter

$$(y')^2(1-C) = C$$

$$y' = \frac{dy}{dx} = \sqrt{\frac{C}{1-C}} = C$$

$$\int \frac{dy}{dx}\,dx = \int C\,dx$$

$y = cx + b$ → It's indeed a straight line.

[5] HS Global Optimization

Global objective function: $J(u,v) = \int E(u,v)\, dx\, dy$

where: $E(u,v) = (I_x u + I_y v + I_t)^2 + \alpha (\|\nabla u\|^2 + \|\nabla v\|^2)$

Using the Euler-Lagrange equation:

① $\frac{\partial E}{\partial u} - \frac{\partial}{\partial x}\frac{\partial E}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial E}{\partial u_y} = 0$

② $\frac{\partial E}{\partial v} - \frac{\partial}{\partial x}\frac{\partial E}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial E}{\partial v_y} = 0$

$\frac{\partial}{\partial x}\left(\frac{\partial E}{\partial u_x}\right) = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial u_x}(\alpha u_x^2)\right) = 2\alpha u_{xx}$

① $\frac{\partial E}{\partial u} - \frac{\partial}{\partial x}\frac{\partial E}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial E}{\partial u_y} = 2(I_x u + I_y v + I_t) I_x - 2\alpha u_{xx} - 2\alpha u_{yy}$

$= 2(I_x u + I_y v + I_t) I_x - 2\alpha \Delta u = 0$

where: $\Delta u = \frac{\partial^2}{\partial x^2} u + \frac{\partial^2}{\partial y^2} u = u_{xx} + u_{yy}$

↓ Laplace operator

Hence:

$$2(I_x u + I_y v + I_t) I_x - 2\alpha \Delta u = 0$$
$$2(I_x u + I_y v + I_t) I_y - 2\alpha \Delta v = 0$$

Q: $\Delta U = U_{xx} + U_{yy}$

How to compute Δu from an image?

A: $U_{xx} = \partial^2 U / \partial x^2$

$$= \frac{\partial}{\partial x}\left(\frac{\partial U}{\partial x}\right) = \frac{\partial}{\partial x}\left(U(x+1,y) - U(x,y)\right)$$

$$= \left(U(x+1,y) - U(x,y)\right) - \left(U(x,y) - U(x-1,y)\right)$$

$$= \left(U(x+1,y) + U(x-1,y)\right) - 2U(x,y)$$

$$U_{xx} = \left(U(x+1,y) + U(x-1,y)\right) - 2U(x,y)$$

$$U_{yy} = \left(U(x,y+1) + U(x,y-1)\right) - 2U(x,y)$$

$$\frac{1}{4}\Delta U = \frac{1}{4}\left(U_{xx} + U_{yy}\right)$$

$$= \underbrace{\frac{1}{4}\left[U(x+1,y) + U(x-1,y) + U(x,y+1) + U(x,y-1)\right]}_{\text{average of neighboring pixels of } (x,y)} - U(x,y)$$

$$\frac{1}{4}\Delta U(x,y) = \underbrace{\bar{U}(x,y)}_{\text{local average}} - U(x,y)$$

Recall:

$$(I_x u + I_y v + I_t) I_x - \alpha \Delta U = 0$$
$$(I_x u + I_y v + I_t) I_y - \alpha \Delta v = 0$$

$$\downarrow\downarrow$$

$$(I_x u + I_y v + I_t) I_x - \alpha' (\bar{U} - U) = 0 \quad ; \quad \alpha' = \alpha/4$$
$$(I_x u + I_y v + I_t) I_y - \alpha' (\bar{v} - v) = 0$$

$$\downarrow\downarrow$$

Since α is decided by us we can also deem $\alpha = \alpha'$

$$\begin{bmatrix} \alpha + I_x^2 & I_x I_y \\ I_y I_x & \alpha + I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha \bar{U} - I_x I_t \\ \alpha \bar{v} - I_y I_t \end{bmatrix}$$

We can't solve it, as we don't know $\bar{U}$ & $\bar{v}$

$$\begin{bmatrix} \alpha + I_x^2 & I_x I_y \\ I_y I_x & \alpha + I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha \bar{u} - I_x I_t \\ \alpha \bar{v} - I_y I_t \end{bmatrix}$$

For 1 pixel: $\underset{2\times2}{\mathbb{A}} \underset{2\times1}{\bar{w}} = \underset{2\times1}{\bar{b}}$

For N pixels: $\underset{2N\times2N}{\mathbb{A}} \underset{2N\times1}{\bar{w}} = \underset{2N\times1}{\bar{b}}$

At this point, we can use any method to solve the equation. Yet, we should consider that $\mathbb{A}$ is a huge matrix and $\bar{u}$, $\bar{v}$ in $\bar{b}$ are unknown.

e.g. $N = 2$

$$\mathbb{A} = \begin{bmatrix} \alpha + I_{x_{1,1}}^2 & I_{x_{1,1}} I_{y_{1,1}} & 0 & 0 \\ I_{y_{1,1}} I_{x_{1,1}} & \alpha + I_{y_{1,1}}^2 & 0 & 0 \\ 0 & 0 & \alpha + I_{x_{1,2}}^2 & I_{x_{1,2}} I_{y_{1,2}} \\ 0 & 0 & I_{y_{1,2}} I_{x_{1,2}} & \alpha + I_{y_{1,2}}^2 \end{bmatrix}; \quad \bar{w} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{bmatrix}$$

[•] Further solution proposed by Horn & Schunck:

Considering each pixel (i, j):

$$\begin{bmatrix} \alpha + I_{x_{ij}}^2 & I_{x_{ij}} I_{y_{ij}} \\ I_{x_{ij}} I_{y_{ij}} & \alpha + I_{y_{ij}}^2 \end{bmatrix} \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \begin{bmatrix} \alpha \bar{u}_{ij} - I_{x_{ij}} I_{t_{ij}} \\ \alpha \bar{v}_{ij} - I_{y_{ij}} I_{t_{ij}} \end{bmatrix}$$

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha + I_{x_{ij}}^2 & I_{x_{ij}} I_{y_{ij}} \\ I_{x_{ij}} I_{y_{ij}} & \alpha + I_{y_{ij}}^2 \end{bmatrix}^{-1}}_{\mathbb{A}} \begin{bmatrix} \alpha \bar{u}_{ij} - I_{x_{ij}} I_{t_{ij}} \\ \alpha \bar{v}_{ij} - I_{y_{ij}} I_{t_{ij}} \end{bmatrix}$$

$$\mathbb{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \mathbb{A}^{-1} = \frac{1}{\det(\mathbb{A})} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\det(\mathbb{A}) = (\alpha + I_{x_{ij}}^2)(\alpha + I_{y_{ij}}^2) - I_{x_{ij}}^2 I_{y_{ij}}^2$$

$$= \alpha^2 + \alpha I_{x_{ij}}^2 + \alpha I_{y_{ij}}^2$$

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \frac{1}{\alpha^2 + \alpha I_{x_{ij}}^2 + \alpha I_{y_{ij}}^2} \begin{bmatrix} \alpha + I_{y_{ij}}^2 & -I_{x_{ij}} I_{y_{ij}} \\ -I_{x_{ij}} I_{y_{ij}} & \alpha + I_{x_{ij}}^2 \end{bmatrix} \begin{bmatrix} \alpha \bar{u}_{ij} - I_{x_{ij}} I_{t_{ij}} \\ \alpha \bar{v}_{ij} - I_{y_{ij}} I_{t_{ij}} \end{bmatrix}$$

$$u_{ij} = \frac{(\alpha + I_{y_{ij}}^2)(\alpha \bar{u}_{ij} - I_{x_{ij}} I_{t_{ij}}) - (I_{x_{ij}} I_{y_{ij}})(\alpha \bar{v}_{ij} - I_{y_{ij}} I_{t_{ij}})}{\alpha^2 + \alpha I_{x_{ij}}^2 + \alpha I_{y_{ij}}^2}$$

$$= \frac{\alpha^2 \bar{u}_{ij} - \alpha I_{x_{ij}} I_{t_{ij}} + \alpha I_{y_{ij}}^2 \bar{u}_{ij} - \alpha I_{x_{ij}} I_{y_{ij}} \bar{v}_{ij}}{\alpha(\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2)}$$

$$(\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) u_{ij} = (\alpha + I_{y_{ij}}^2) \bar{u}_{ij} - I_{x_{ij}} I_{y_{ij}} \bar{v}_{ij} - I_{x_{ij}} I_{t_{ij}}$$

$$(\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) u_{ij} - (\alpha + I_{y_{ij}}^2) \bar{u}_{ij} = -(I_{x_{ij}} I_{y_{ij}} \bar{v}_{ij} + I_{x_{ij}} I_{t_{ij}})$$

$$(\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) u_{ij} - (\alpha + I_{y_{ij}}^2 + I_{x_{ij}}^2) \bar{u}_{ij} = -I_{x_{ij}} (I_{y_{ij}} \bar{v}_{ij} + I_{t_{ij}}) - I_{x_{ij}}^2 \bar{u}_{ij}$$

$$(\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) u_{ij} = (\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) \bar{u}_{ij} - \left[I_{x_{ij}} (I_{y_{ij}} \bar{v}_{ij} + I_{x_{ij}} \bar{u}_{ij} + I_{\epsilon_{ij}}) \right]$$

Applying the same operations to v_{ij} :

$$(\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \begin{bmatrix} (\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) \bar{u}_{ij} - \left[I_{x_{ij}} (I_{y_{ij}} \bar{v}_{ij} + I_{x_{ij}} \bar{u}_{ij} + I_{\epsilon_{ij}}) \right] \\ (\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2) \bar{v}_{ij} - \left[I_{y_{ij}} (I_{y_{ij}} \bar{v}_{ij} + I_{x_{ij}} \bar{u}_{ij} + I_{\epsilon_{ij}}) \right] \end{bmatrix}$$

$= \bar{b}$

Iterative solution using the Gauss - Seidel method :

$$A\bar{x} = \bar{b} \quad ; \quad A = L + D + U$$

then :

$$\bar{x}^{new} = (L + D)^{-1} (\bar{b} - U \bar{x}^{old})$$

$$A = \begin{bmatrix} \alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2 & 0 \\ 0 & \alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2 \end{bmatrix} \quad ; \quad \begin{aligned} L &= 0 \\ D &= A \\ U &= 0 \end{aligned}$$

$$\bar{x} = \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix}$$

Hence :

$$\bar{x}^{new} = (\mathbb{L} + \mathbb{D})^{-1} (\bar{b} - \mathbb{U}\bar{x}^{old}) \quad ; \quad \mathbb{L} = \mathbb{U} = 0$$

$$= \mathbb{D}^{-1} \bar{b}$$

$$\mathbb{D} = \begin{bmatrix} \alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2 & 0 \\ 0 & \alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2 \end{bmatrix} \rightarrow \mathbb{D}^{-1} = \frac{1}{\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2}$$

$$U_{ij}^{new} = \bar{U}_{ij}^{old} - \frac{I_{x_{ij}} (I_{x_{ij}} \bar{U}_{ij}^{old} + I_{y_{ij}} \bar{v}_{ij}^{old} + I_{t_{ij}})}{\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2}$$

$$v_{ij}^{new} = \bar{v}_{ij}^{old} - \frac{I_{y_{ij}} (I_{x_{ij}} \bar{U}_{ij}^{old} + I_{y_{ij}} \bar{v}_{ij}^{old} + I_{t_{ij}})}{\alpha + I_{x_{ij}}^2 + I_{y_{ij}}^2}$$

Final equations proposed by Horn & Schunck

Note:

$$\begin{bmatrix} \alpha^2 + I_x^2 & I_x I_y \\ I_x I_y & \alpha^2 + I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha^2 \bar{u} - I_x I_t \\ \alpha^2 \bar{v} - I_y I_t \end{bmatrix}$$

$$\boxed{\mathbb{A}\,\bar{w} = \bar{b}}$$

For N pixels: $(2N \times 2N)\ (2N \times 1)\ (2N \times 1)$

where:

$\mathbb{A}$ is a matrix whose elements are known.

$\bar{w}$ is a vector containing (u, v) that are unknown.

$\bar{b}$ is a vector containing $(\bar{u}, \bar{v})$ that are also unknown.

Q: Can we assume $\bar{u}^{old}$ & $\bar{v}^{old}$ to get vector $\bar{b}$?

A: One possibility is to initialize u & v maps with zero, and then compute $\bar{b}$.

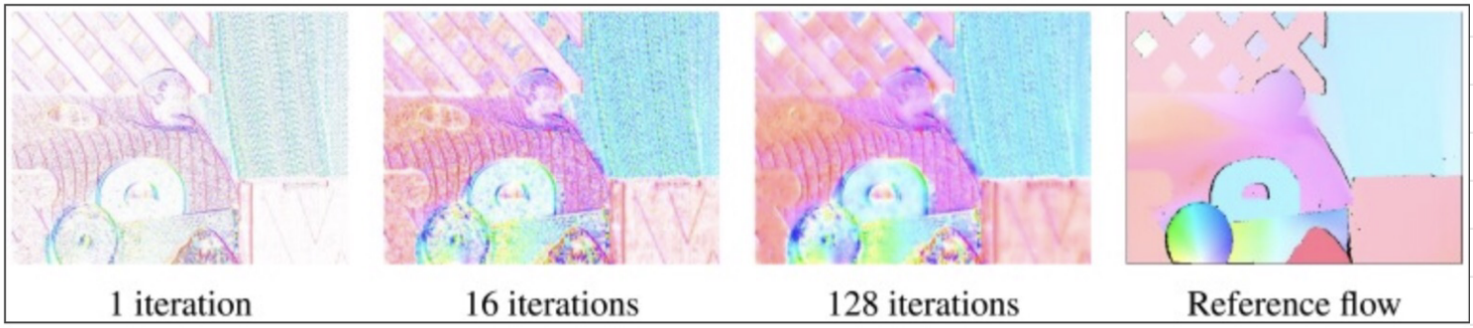
Having computed $\bar{b}$, we obtain $\bar{w} = \mathbb{A}^{-1}\,\bar{b}(w_{init})$

and do this iteratively: $\bar{w}_{new} = \mathbb{A}^{-1}\,\bar{b}(\bar{w}_{old})$.

However, computing $\mathbb{A}^{-1}$ in every update is expensive, since if we have a 100×100 image $\rightarrow N = 10{,}000$, thus the dimension of $\mathbb{A} = 20{,}000 \times 20{,}000$.

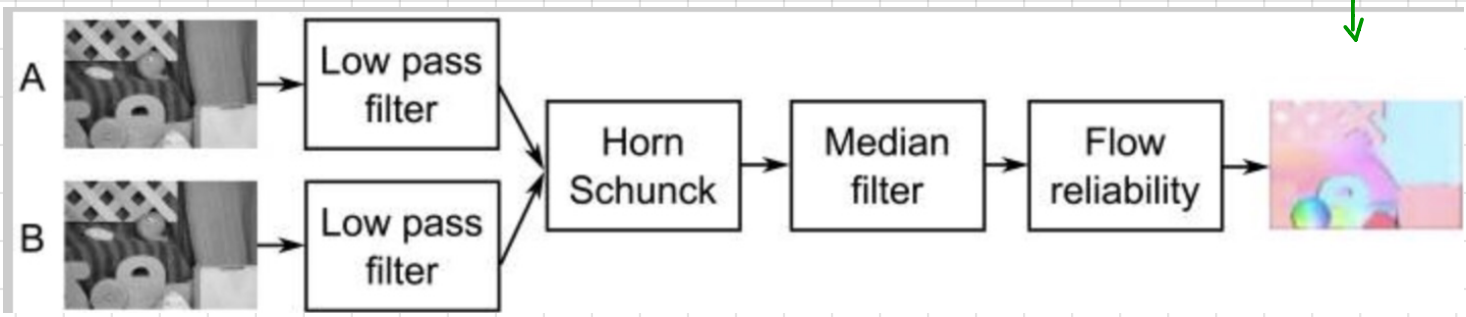
[6] Secrets of Robust Optic Flow

Optical flow results using the HS method:



It works, but the results are noisy & inaccurate.

Below is one of the possible ways to improve:

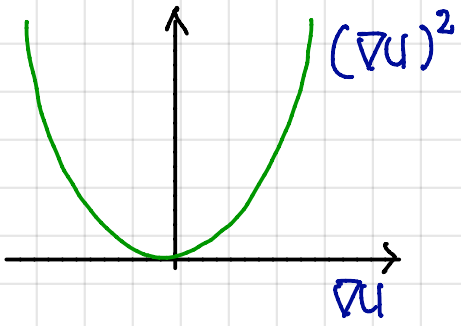


Q: What are the secrets for robust flow estimation?

A: (1) Robust objective function
(2) Coarse to fine mechanism
(3) Interpolation
(4) Median Filtering
(5) Pre-processing
(6) Gradient Constancy Constraint

[7] Robust Objective Functions

Problem: A quadratic function is convenient to optimize, but it is statistically not robust.



This curve means: when ∇u is larger the penalty, $(\nabla u)^2$, will be higher.

↓

It's indeed something we want.

↓

However, when u suffers from noise, inevitably the penalty is also high; which is not good, since the actual penalty should be small.

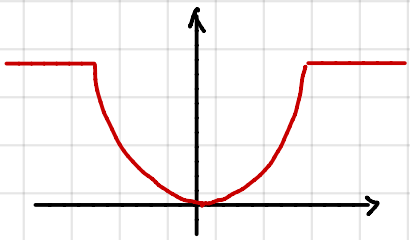
Since, without noise, that value of u is the correct one.

↓

In other words, we penalize a correct value of u with a high cost because it suffers from noise.

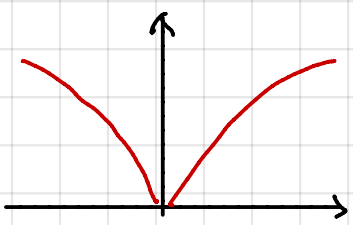
More robust functions:

(1) Truncated Quadratic:



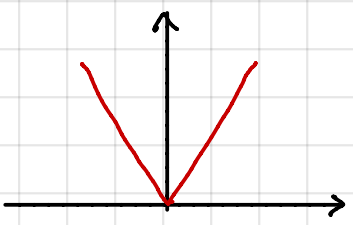
$$f(x) = \begin{cases} \lambda x^2 & \text{if } |x| < \frac{\sqrt{\alpha}}{\sqrt{\lambda}} \\ \alpha & \text{else} \end{cases}$$

(2) Lorentzian



$$f(x) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right)$$

(3) Charbonnier



$$f(x) = (x^2 + \varepsilon^2)^a$$

$a = 0.5$

if $a < 0.5$ the function becomes non-convex.

[8] Coarse to Fine

Problem: Original HS method doesn't work for large displacement or large motion.

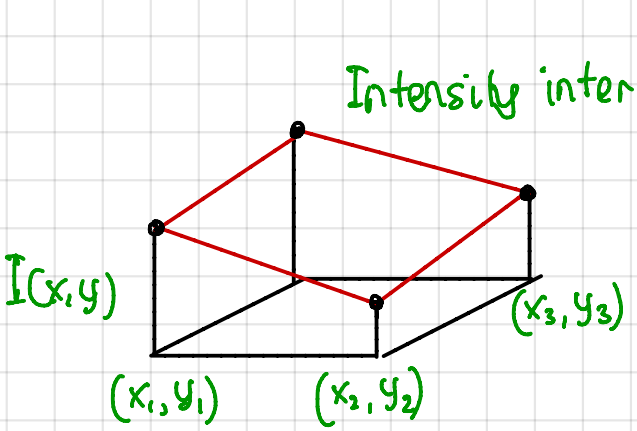
Solution:

(1) Perform a standard multi-resolution technique [to generate pyramid images]
(2) Estimate optic flow at a coarse level
(3) For the next finer level, warp the second image to the first image using the estimated optic flow in step #2.
(4) Estimate optic flow from the warped images 2 & 1.
(5) Do step #2 to #4 till the finest level.

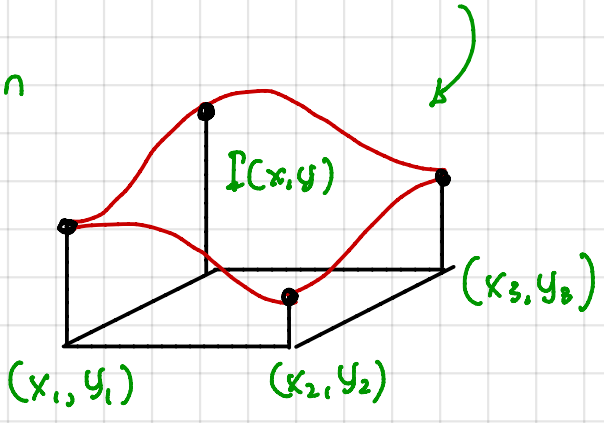
[11] Interpolation

Problem: The optic flow from the coarser level is more sparse than the number of pixels in the finer level.

Solution: Interpolation → Bicubic interpolation



Bilinear Interpolation



Bicubic Interpolation

[9] Median Filtering

Problem: Optic flow estimation can be noisy, and the median filtering can smooth out the noise.

Median filter: To replace each point or pixel with the median of the neighboring points (5×5 pixels)

↓

Solution: To apply the median filtering to the intermediate flow after every warping iteration.

---

[10] Gradient Constancy Constraint

Problem: The BCC is not robust to the light changes

Solution: Gradient constancy assumption (GCC):

$$\nabla I(x,y,t) = \nabla I(x+u, y+v, t+1)$$

$$\begin{bmatrix} \frac{\partial I(x,y,t)}{\partial x} \\ \frac{\partial}{\partial y}(x,y,t) \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x}(x+u, y+v, t+1) \\ \frac{\partial I}{\partial y}(x+u, y+v, t+1) \end{bmatrix}$$

$$\frac{d}{dt}\nabla I(x,y,t) = 0$$

GCC

$$\frac{d}{dt}\begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I_x}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I_x}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I_x}{\partial t} \\ \frac{\partial I_y}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I_y}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I_y}{\partial t} \end{bmatrix} = \begin{bmatrix} I_{xx}u + I_{xy}v + I_{xt} \\ I_{yx}u + I_{yy}v + I_{yt} \end{bmatrix} = 0$$

We can combine the BCC & GCC in one objective function.

Combining the BCC & GCC :

$$E(u,v) = \underbrace{(I_x u + I_y v + I_t)^2}_{\text{BCC}} + \gamma \underbrace{\left\| \begin{matrix} I_{xx} u + I_{xy} v + I_{xt} \\ I_{yx} u + I_{yy} v + I_{yt} \end{matrix} \right\|_2^2}_{\text{GCC}} + \alpha \underbrace{\left(\|\nabla u\|_2 + \|\nabla v\|_2 \right)}_{\text{Regularization}}$$