CS-565 Computer Vision

Nazar Khan

Department of Computer Science
University of the Punjab

13. Optic Flow – Local

Optic Flow

Optic Flow

Optic Flow

Where does pixel (x, y) in frame z move to in frame $z + 1$?

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix}$$

We want to find the displacement vector $(u, v)^T$ for every pixel.

- Input: image sequence $I(x, y, z)$, where (x, y) specifies the location and z denotes time/frame number
- Goal: displacement vector field of the image structures:
  - optic flow $(u(x, y, z), v(x, y, z))$
- Such correspondence problems are key problems in computer vision.

Grey Value Constancy (GVC) Assumption

- Corresponding pixels should have the same grey value.
- That is, the optic flow between frame z and $z + 1$ should satisfy

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

Taylor's Approximation for 2D Functions

- Recall that 1st-order Taylor's approximation for 1D functions is

$$f(x+u) \approx f(x) + \frac{u}{1!} f'(x)$$

- For 2D functions, a 1st-order Taylor's approximation is

$$f(x+u, y+v) \approx f(x,y) + \frac{u}{1!} f_x(x,y) + \frac{v}{1!} f_y(x,y)$$

- For 3D functions, a 1st-order Taylor's approximation is

$$f(x+u, y+v, z+w) \approx f(x,y,z) + \frac{u}{1!} f_x(x,y,z) + \frac{v}{1!} f_y(x,y,z) + \frac{w}{1!} f_z(x,y,z)$$

- For our case,

$$I(x+u, y+v, z+1) \approx$$
$$I(x,y,z) + uI_x(x,y,z) + vI_y(x,y,z) + 1I_z(x,y,z)$$

Optic Flow Constraint

- The grey value constancy (GVC) assumption

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

can then be approximated as

$$\begin{aligned} & I(x,y,z) + uI_x(x,y,z) + vI_y(x,y,z) + 1I_z(x,y,z) \approx I(x,y,z) \\ \Longrightarrow\ & I_x(x,y,z)u + I_y(x,y,z)v + I_z(x,y,z) \approx 0 \end{aligned}$$

assuming (u, v) is a small displacement.

- The is known as the *linearized optic flow constraint (OFC)*

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

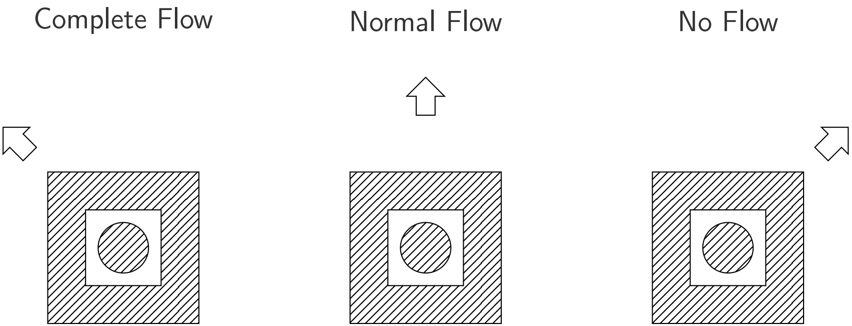
where location (x, y, z) is implied.

How good are the assumptions?

- We have made two assumptions
  1. Gray value constancy
  2. Small displacements (since we use first-order Taylor series approximation)
- Both assumptions are (almost) true in surprisingly many scenarios.
  1. Gray values do not change much between *consecutive*[1] frames.
  2. Objects do not move too much between *consecutive* frames.
     - For large displacements, image pyramid can be used.

[1] For a video recorded at 25 frames per second (fps), consecutive frames are only $\frac{1}{24}$ seconds apart.

Aperture Problem



When seen through an aperture, true movement cannot be determined. Only the component of movement normal to edge direction can be determined.

Normal Flow

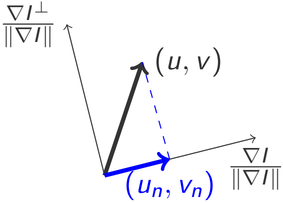
- The OFC is one equation in two unknowns (infinite solutions).
- Can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix}^T \nabla I + I_z = 0$$

- Adding any flow component orthogonal to image gradient does not affect the OFC.

$$\begin{aligned} \left(\begin{bmatrix} u \\ v \end{bmatrix} + k\nabla I^{\perp} \right)^T \nabla I + I_z &= \begin{bmatrix} u \\ v \end{bmatrix}^T \nabla I + k \underbrace{\nabla I^{\perp T} \nabla I}_{0} + I_z \\ &= \begin{bmatrix} u \\ v \end{bmatrix}^T \nabla I + I_z \\ &= 0 \end{aligned}$$

Normal Flow



$$
\begin{aligned}
\begin{bmatrix} u_n \\ v_n \end{bmatrix} &= \left(\begin{bmatrix} u \\ v \end{bmatrix} \bullet \frac{\nabla I}{\|\nabla I\|} \right) \frac{\nabla I}{\|\nabla I\|} \\
&= \frac{-I_z}{\|\nabla I\|} \frac{\nabla I}{\|\nabla I\|} \quad \left(\because \begin{bmatrix} u \\ v \end{bmatrix} \bullet \nabla I + I_z = 0 \right) \\
&= \frac{-1}{I_x^2 + I_y^2} \begin{bmatrix} I_x I_z \\ I_y I_z \end{bmatrix}
\end{aligned}
$$

- Only the component of flow in the direction of the gradient ∇I can be computed.
- Since gradient is normal to the edge direction, this flow vector is called the *normal flow*.
- To compute a better estimate of optic flow, we need to make some assumptions.

Local Optic Flow Method of Lucas & Kanade

- Lucas & Kanade[2] make the following assumption:

> Pixels around (i, j) all have the same displacement (u, v).

- For 3×3 neighbourhoods, this gives 9 OFCs all having the same 2 unknowns (u, v).
- The optimal unknown displacement minimizes the sum-squared-error

$$E(u, v) = \frac{1}{2} \sum_{\mathcal{N}_{ij}} (I_x u + I_y v + I_z)^2$$

[2] Bruce D. Lucas and Takeo Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision". In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'81. Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, 674–679.

Local Optic Flow Method of Lucas & Kanade

- Setting $\frac{\partial E}{\partial u} = 0$ and $\frac{\partial E}{\partial v} = 0$ yields a linear system

$$\begin{bmatrix} \sum_{\mathcal{N}_{ij}} I_x^2 & \sum_{\mathcal{N}_{ij}} I_x I_y \\ \sum_{\mathcal{N}_{ij}} I_x I_y & \sum_{\mathcal{N}_{ij}} I_y^2 \end{bmatrix} \begin{bmatrix} u^* \\ v^* \end{bmatrix} = \begin{bmatrix} -\sum_{\mathcal{N}_{ij}} I_x I_z \\ -\sum_{\mathcal{N}_{ij}} I_y I_z \end{bmatrix}$$

- Replacing the sums by Gaussian averaging yields

$$\underbrace{\begin{bmatrix} G_\rho * I_x^2 & G_\rho * I_x I_y \\ G_\rho * I_x I_y & G_\rho * I_y^2 \end{bmatrix}}_{A} \begin{bmatrix} u^* \\ v^* \end{bmatrix} = \begin{bmatrix} -G_\rho * I_x I_z \\ -G_\rho * I_y I_z \end{bmatrix} \quad (1)$$

- Notice the re-appearance of the structure tensor which now serves as the system matrix. Previously, we used it for corner detection.
- Flow vector can be found if $\text{rank}(A) = 2$.

Local Optic Flow Method of Lucas & Kanade

- If $\text{rank}(A) = 0$, no gradients exist in the neighbourhood. So no optic flow can be computed.
- If $\text{rank}(A) = 1$, gradient vectors over all pixels in the neighbourhood are identical. Only normal flow can be computed.

$$\begin{bmatrix} u_n \\ v_n \end{bmatrix} = \frac{-1}{I_x^2 + I_y^2} \begin{bmatrix} I_x I_z \\ I_y I_z \end{bmatrix}$$

- To save computations, avoid computing rank.

$$\text{trace}(A) = A_{11} + A_{22} \approx 0 \implies \text{rank}(A) = 0$$
$$\text{trace}(A) \not\approx 0 \text{ and } \det(A) = A_{11}A_{22} - A_{12}^2 \approx 0 \implies \text{rank}(A) = 1$$

Lucas & Kanade
Algorithm

Input: Frames I_1 and I_2.
Parameters:
1) Noise smoothing scale σ,
2) Gradient smoothing scale ρ,
3) Thresholds τ_{trace} and τ_{det}.

1. Compute Gaussian derivatives at noise smoothing scale σ
$$I_x = \frac{\partial G_\sigma}{\partial x} * I_1 \quad \text{and} \quad I_y = \frac{\partial G_\sigma}{\partial y} * I_1$$
2. Compute temporal derivative $I_z = I_2 - I_1$.
3. Compute the products
$$I_x^2 \quad I_y^2 \quad I_x I_y \quad I_x I_z \quad \text{and} \quad I_y I_z$$
4. Smooth the products at gradient smoothing scale ρ
$$G_\rho * I_x^2 \quad G_\rho * I_y^2 \quad G_\rho * I_x I_y \quad G_\rho * I_x I_z \quad \text{and} \quad G_\rho * I_y I_z$$
and construct the linear system in (1) at every pixel.

Lucas & Kanade
Algorithm

5. For every pixel, solve the linear system conditioned on the rank.
   if $A_{11} + A_{22} < \tau_{\text{trace}}$
       rank(A)=0 so no flow
   else if $A_{11}A_{22} - A_{12}^2 < \tau_{\text{det}}$
       rank(A)=1 so normal flow
   else
       rank(A)=2 so complete optic flow

Visualising Displacement Vectors

The HSV Color Space

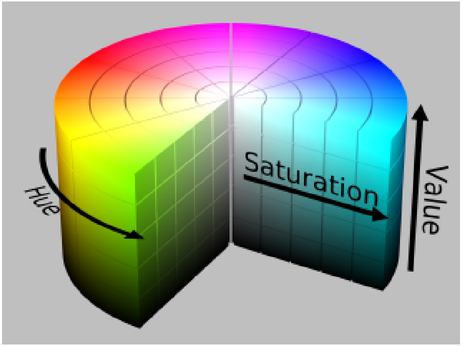


Figure: The HSV color space.

http://reilley4color.blogspot.com/2016/05/munsell-hue-circle.html

Each color is represented by 3 values

1. **H**ue or shade as an angle from 0° to 360°.
2. **S**aturation or strength of the color
3. **V**alue or brightness

Visualising Displacement Vectors

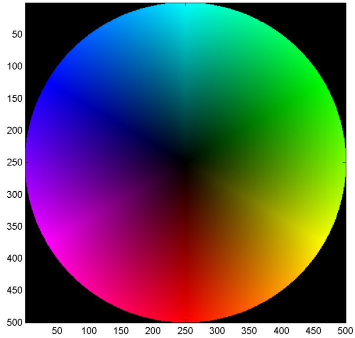


Figure: Vector angle represented by hue/shade of color and vector magnitude represented by the saturation/strength of color. HSV color space is useful for such a mapping. $H(x, y) = \theta(x, y)$, $S(x, y) = \sqrt{u(x, y)^2 + v(x, y)^2}$ and $V(x, y) = \text{constant}$.

Lucas & Kanade

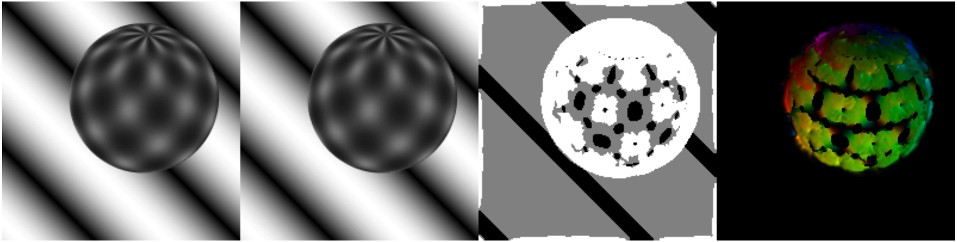


Figure: Left to right: frame 1, frame 2, flow classification and false color visualization of optic flow vectors. For flow classification: white = optic flow, gray = normal flow and black = no flow. Integration scale was $\rho = 1$. Author: N. Khan (2015)

Lucas & Kanade

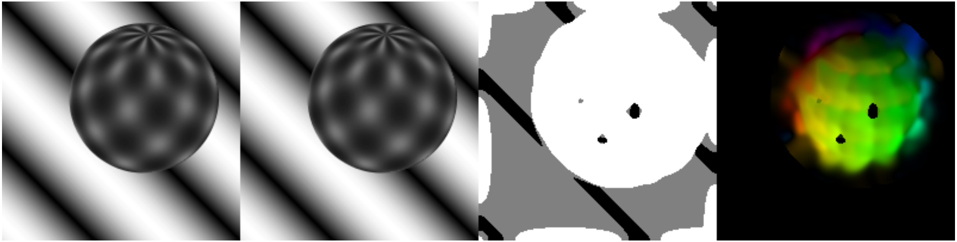


Figure: Left to right: frame 1, frame 2, flow classification and false color visualization of optic flow vectors. For flow classification: white = optic flow, gray = normal flow and black = no flow. Increasing the integration scale ρ to 4 fills up pixels with no flow using values from neighbouring pixels having normal or complete optic flow. Author: N. Khan (2015)

Lucas & Kanade

Lucas & Kanade

Lucas & Kanade

Figure: Left to right: false color visualization of optic flow vectors, flow classification and flow magnitude. For flow classification: white = optic flow, gray = normal flow and black = no flow. Noise smoothing scale was $\sigma = 1$ and integration scale was $\rho = 2$. Author: N. Khan (2018)

Lucas & Kanade

Lucas & Kanade

Lucas & Kanade

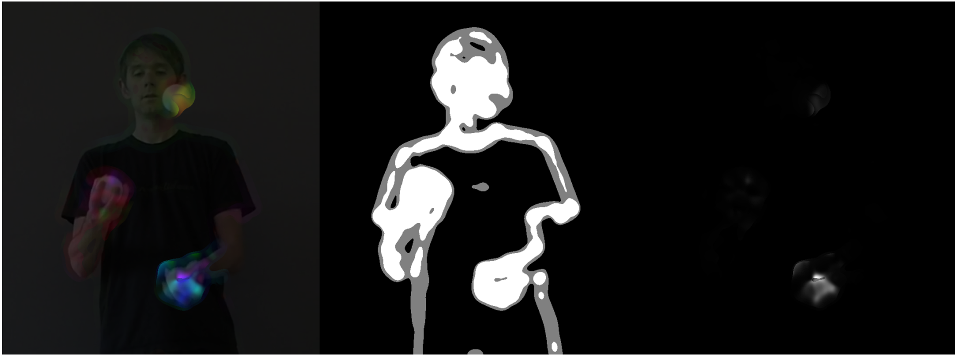


Figure: Left to right: false color visualization of optic flow vectors, flow classification and flow magnitude. For flow classification: white = optic flow, gray = normal flow and black = no flow. Noise smoothing scale was $\sigma = 1$ and integration scale was $\rho = 8$. Author: N. Khan (2018)

Lucas & Kanade

Summary

Advantages

- Simple and fast method.
- Requires only two frames (low memory requirements).
- Good value for money: results often superior to more complicated approaches.

Disadvantages

- Problems at locations where the local constancy assumption is violated: flow discontinuities and non-translatory motion (e.g. rotation).
- Local method that does not compute the flow field at all locations.

Next we study a global method that produces dense flow fields (*i.e.*, at every pixel).