1 Linear Approximation

The linear approximation of $f(x)$ at a point a is the linear function:

$$L(x) = f(a) + f'(a)(x-a).$$

The linear approximation of $f(x,y)$ at (a,b) is the linear function:

$$L(x,y) = f(a,b) + f_x(a,b)(x-a) + f_y(a,b)(y-b).$$

the linearization can be written more compactly using the **gradient** as:

$$L(\tilde{\mathbf{x}}) = f(\tilde{\mathbf{a}}) + \nabla f(\tilde{\mathbf{a}}) \cdot (\tilde{\mathbf{x}} - \tilde{\mathbf{a}}).$$

1.1 Chain Rule

The single variable chain rule is given by:

$$\frac{dy}{dx} = \frac{df}{du} \cdot \frac{du}{dx}$$

For a multivariate function, the chain rule is:

$$\frac{\partial y}{\partial x_i} = \frac{\partial f}{\partial u} \cdot \frac{\partial u}{\partial x_i} + \frac{\partial f}{\partial v} \cdot \frac{\partial v}{\partial x_i} + \frac{\partial f}{\partial w} \cdot \frac{\partial w}{\partial x_i} + \cdots$$

The matrix form of the multivariate chain rule is expressed using Jacobian matrices:

$$D(\mathbf{y})(\mathbf{x}) = Df(\mathbf{u})(\mathbf{x}) \cdot Dg(\mathbf{x})$$

where $Df(\mathbf{u})$ and $Dg(\mathbf{x})$ are the Jacobian matrices of f and g respectively.

1.2 Tangent line, tangent plane & tangent hyperplane

The equation of the tangent line to the curve $y = f(x)$ at the point $(a, f(a))$ is:

$$y = f(a) + f'(a)(x-a)$$

The equation of the tangent plane to the surface $z = f(x,y)$ at the point $(a, b, f(a,b))$ is:

$$z = f(a,b) + f_x(a,b)(x-a) + f_y(a,b)(y-b)$$

Here, f_x and f_y denote the partial derivatives of f with respect to x and y, respectively.

For a function $f(x_1, x_2, \ldots, x_n)$ at a point $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, the tangent hyperplane is given by:

$$f(\mathbf{x}) \approx f(\mathbf{a}) + \nabla f(\mathbf{a}) \cdot (\mathbf{x} - \mathbf{a})$$

where $\nabla f(\mathbf{a})$ is the gradient of f at $\mathbf{a}$ and $\mathbf{x}$ is the vector of variables.

1.3 Derivatives of Vector Functions

Let $F : \mathbb{R}^n \to \mathbb{R}^m, F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{bmatrix}$ be a vector function of the variables x = $(x_1, ..., x_n)$.

Recall that the derivative of the vector function F with respect to the vector of variables $\tilde{x}$ is defined as

$$\frac{\partial \tilde{F}}{\partial \tilde{x}} = J_F(\tilde{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\tilde{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\tilde{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\tilde{x}) & \cdots & \frac{\partial f_m}{\partial x_n}(\tilde{x}) \end{bmatrix}$$

The second derivative of the function $f : \mathbb{R}^n \to \mathbb{R}$ (here m = 1) is given as

$$\frac{\partial^2 f}{\partial \tilde{x}^2} = \frac{\partial}{\partial \tilde{x}} \left(\frac{\partial f}{\partial \tilde{x}} \right)^T$$

.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex on D, if

$$f(t\tilde{x} + (1-t)\tilde{y}) \leq t f(\tilde{x}) + (1-t) f(\tilde{y})$$

for all $\tilde{x}, \tilde{y} \in D$ and for all $t \in [0,1]$. The function f is concave on D, if the function $-f$ is convex on D.

1.4 Rules for Differentiating Vector Functions

1. $\frac{\partial \tilde{x}}{\partial \tilde{x}} = I_n$
2. If $A \in \mathbb{R}^{m \times n}$, then $\frac{\partial A\tilde{x}}{\partial \tilde{x}} = A$.
3. If $\tilde{a} \in \mathbb{R}^n$, then $\frac{\partial \tilde{a}^T \tilde{x}}{\partial \tilde{x}} = \tilde{a}^T$.
4. If $A \in \mathbb{R}^{n \times n}$, then $\frac{\partial (\tilde{x}^T A \tilde{x})}{\partial \tilde{x}} = \tilde{x}^T (A + A^T)$.
5. If $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix, then $\frac{\partial (\tilde{x}^T A \tilde{x})}{\partial \tilde{x}} = 2\tilde{x}^T A$.
6. $\frac{\partial \|\tilde{x}\|^2}{\partial \tilde{x}} = 2\tilde{x}^T$.
7. If $\tilde{z} = \tilde{z}(\tilde{x})$ and $\tilde{y} = \tilde{y}(\tilde{x})$, then $\frac{\partial (\tilde{y}^T \tilde{z})}{\partial \tilde{x}} = \tilde{y}^T \frac{\partial \tilde{z}}{\partial \tilde{x}} + \tilde{z}^T \frac{\partial \tilde{y}}{\partial \tilde{x}}$.
8. If $G : D_G \subseteq \mathbb{R}^m \to \mathbb{R}^n$ and $F : D_F \subseteq \mathbb{R}^n \to \mathbb{R}^p$ and $H = F \circ G$, then $\frac{\partial H}{\partial \tilde{x}} = \frac{\partial F}{\partial G}(\tilde{G}(\tilde{x})) \frac{\partial G}{\partial \tilde{x}}$.

1.5 Linear Approximation of Vector Functions

$$f(X(p_1, p_2, \ldots, p_m), S(p_1, p_2, \ldots, p_m)) \approx k_n(p_1, p_2, \ldots, p_m)$$

$$X = \begin{bmatrix} X_1(p_1, \ldots, p_m) \\ \vdots \\ X_n(p_1, \ldots, p_m) \end{bmatrix}, \quad \delta = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix}, \quad p = \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix}$$

$$f(X(p_0 + \delta)) \approx f(X(p_0)) + df$$

$$df = \frac{\partial f}{\partial X} \frac{\partial X}{\partial p} \delta = J_f \delta, \quad \text{take expansion of } \frac{\partial X}{\partial p} \text{ around } p_0$$

$$dX = \begin{bmatrix} \frac{\partial X_1}{\partial p_1} & \frac{\partial X_1}{\partial p_2} & \cdots & \frac{\partial X_1}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial X_n}{\partial p_1} & \frac{\partial X_n}{\partial p_2} & \cdots & \frac{\partial X_n}{\partial p_m} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_m \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{m} \frac{\partial X_1}{\partial p_j} \delta_j \\ \vdots \\ \sum_{j=1}^{m} \frac{\partial X_n}{\partial p_j} \delta_j \end{bmatrix}$$

$$J_f = \frac{\partial f}{\partial X} \frac{\partial X}{\partial p}$$

$$f(X(p_0 + \delta)) \approx f(X(p_0)) + J_f \delta$$

2 Optical Flow

Must watch: All about optical flow.

2.1 Optical Flow Constraint Equation

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

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + I_x\delta x + I_y \delta y + I_t \delta t \quad (2)$$

Subtracting equation (1) from (2):

$$I_x\delta x + I_y \delta y + I_t \delta t = 0$$

Divide by δt and take the limit as $\delta t \to 0$:

$$I_x \frac{\partial x}{\partial t} + I_y \frac{\partial y}{\partial t} + I_t = 0 \quad (3)$$

(3) works because of limit - derivative property:

$$\frac{dy}{dx} = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$$

$$m = \lim_{x\to a} \frac{f(x)-f(a)}{x \to a}$$

The constraint equation for optical flow is then:

$$I_x u + I_y v + I_t = 0 \quad \text{(u, v): Optical Flow}$$

The problem is that in this equation we have 2 unknowns and only 1 equation. We need to solve for u and v.

We solve this by using the **Lucas-Kanade** and **Horn-Schunck** methods. Where we use the local neighbourhood to solve the system of equations.

2.2 Lucas-Kanade Method

$$u = \delta_x = \frac{-\left(\sum I_y^2\right)\left(\sum I_x I_t\right) + \left(\sum I_x I_y\right)\left(\sum I_y I_t\right)}{\left(\sum I_x^2\right)\left(\sum I_y^2\right) - \left(\sum I_x I_y\right)^2}$$

$$v = \delta_y = \frac{-\left(\sum I_x I_y\right)\left(\sum I_x I_t\right) - \left(\sum I_x^2\right)\left(\sum I_y I_t\right)}{\left(\sum I_x^2\right)\left(\sum I_y^2\right) - \left(\sum I_x I_y\right)^2}$$

All the sums go from 1 to n, where n is the number of pixels in the local neighbourhood.

where:

$$I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t},$$

and

$$\left(\sum I_x^2\right)\left(\sum I_y^2\right) - \left(\sum I_x I_y\right)^2$$

is the determinant of covariance matrix.

To further clarify the system, it can be written in matrix form as:

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$$\mathbf{A}^T\mathbf{A}\mathbf{d} = \mathbf{A}^T\mathbf{b}$$

where **A** and **b** are matrices containing the sums of gradients, and **d** is the displacement vector.

And $\mathbf{A}^T\mathbf{A}$ is a covariance matrix of local gradients.

When is this system solvable?

- $\mathbf{A}^T\mathbf{A}$ must not be singular (cannot invert it otherwise)
  - Eigenvalues λ_1 and λ_2 of $\mathbf{A}^T\mathbf{A}$ must not be too small
- $\mathbf{A}^T\mathbf{A}$ has to be well conditioned
  - Ratio $\frac{\lambda_1}{\lambda_2}$ must not be too large
  - (λ_1 is the larger eigenvalue)

Consequences:

- large λ_1, small λ_2 → unreliable, we don't know the direction (e.g. edge).
- small λ_1 & λ_2 → unreliable optical flow (e.g. constant background).
- large λ_1 & λ_2 → reliable optical flow (e.g. rich texture, lots of different brightness values).

Conclusions The Lucas-Kanade method is designed for optical flow estimation, leveraging assumptions of brightness constancy, small motion, and spatial coherence to track motion in visual scenes. It excels in controlled environments with minimal background changes, making it efficient and robust for applications with fixed or slow-moving cameras. However, its limitations include difficulty handling large motions, varying lighting conditions, appearance changes, occlusions, and complex scenes.

2.3 Horn-Schunck Method

Flow smoothness error:

$$Es = \iint_D (u_x^2 + u_y^2 + v_x^2 + v_y^2)\, dx\, dy$$

Color constancy + small motion error:

$$Ec = \iint_D (I_x u + I_y v + I_t)^2\, dx\, dy$$

$$E = Ec + \alpha Es$$

$$E = \iint_D L(u, v, u_x, u_y, v_x, v_y, x, y)\, dx\, dy$$

$$L = I_x(I_x u + I_y v + I_t) - \alpha(u_{xx} + u_{yy}) \leq 0$$

Euler-Lagrange equations:

$$\frac{\partial L}{\partial v} - \frac{d}{dx}\left(\frac{\partial L}{\partial u_x}\right) - \frac{d}{dy}\left(\frac{\partial L}{\partial u_y}\right) = 0$$

$$\frac{\partial L}{\partial v} - \frac{d}{dx}\left(\frac{\partial L}{\partial v_x}\right) - \frac{d}{dy}\left(\frac{\partial L}{\partial v_y}\right) = 0$$

$$\Delta = \begin{bmatrix} \frac{\partial^2}{\partial x^2} \\ \frac{\partial^2}{\partial y^2} \end{bmatrix}$$

$$\begin{bmatrix} I_x(I_x u + I_y v + I_t) - \alpha \Delta u = 0 \\ I_y(I_x u + I_y v + I_t) - \alpha \Delta v = 0 \end{bmatrix}$$

By discretization $\Delta u = \bar{u} - u$, $\Delta v = \bar{v} - v$:

$$I_x(I_x u + I_y v + I_t) - \alpha \Delta u = 0 \quad \Rightarrow \quad (I_x^2 + \alpha)u + I_x I_y v = \alpha \bar{u} - I_x I_t$$

$$I_y(I_x u + I_y v + I_t) - \alpha \Delta v = 0 \quad \Rightarrow \quad I_x I_y u + (I_y^2 + \alpha)v = \alpha \bar{v} - I_y I_t$$

And in matrix form:

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

Solve for $[u, v]$:

$$\begin{bmatrix} (I_x^2 + \alpha) & I_x I_y \\ I_x I_y & (I_y^2 + \alpha) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u - I_x P \\ v - I_y P \end{bmatrix} \quad , \quad P = \frac{(I_t + I_x \bar{u} + I_y \bar{v})}{(I_x^2 + I_y^2 + \alpha)}$$

Solve iteratively by Gauss-Seidel-like approach independently for each pixel:

$$\begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} = \begin{bmatrix} \bar{u}^{(k-1)} - I_x P \\ \bar{v}^{(k-1)} - I_y P \end{bmatrix} \quad , \quad P = \frac{(I_t + I_x u^{(k-1)} + I_y v^{(k-1)})}{(I_x^2 + I_y^2 + \alpha)}$$