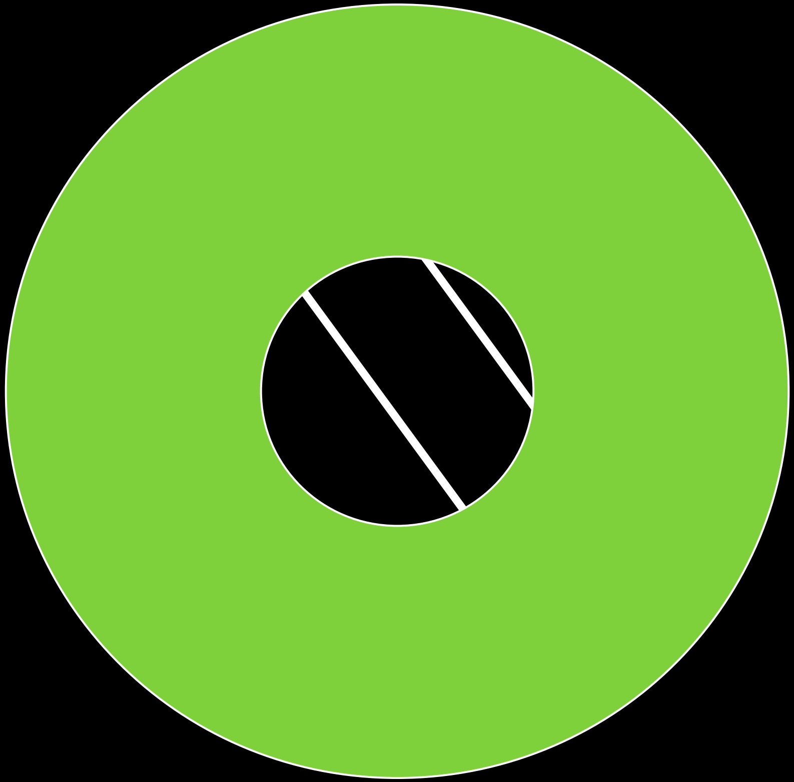
CS4495/6495
Introduction to Computer Vision

6B-L2 *Dense flow: Lucas and Kanade*

Recall: Aperture problem

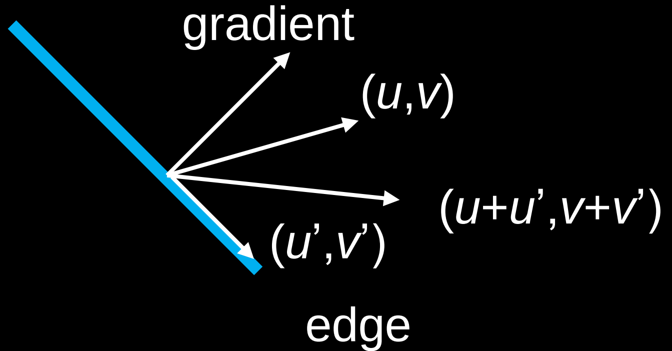


Gradient component of flow

$$0 = I_t + \nabla I \cdot < u, v > \quad \text{or} \quad I_x u + I_y v + I_t = 0$$

Intuitively, what does this constraint mean?

- The component of the flow in the gradient direction is determined
- The component of the flow parallel to an edge is unknown



Solving the aperture problem

- Basic idea: Impose *local* constraints to get more equations for a pixel
  - E.g., assume that the flow field is smooth locally

Solving the aperture problem

- One method: Pretend the pixel's neighbors have the ***same*** (u, v)
  - If we use a 5x5 window, that gives us 25 equations per pixel!

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{A \\ 25\times2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{d \\ 2\times1}} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{b \\ 25\times1}}$$

Lukas-Kanade flow

Problem: We have more equations than unknowns

$$(d = [u\ v])$$

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

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

Lukas-Kanade flow

Solution: Least squares problem

(The summations are over all pixels in the K x K window)

$$\underset{25\times 2}{A}\ \underset{2\times 1}{d} = \underset{25\times 1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

$$\underset{2\times 2}{(A^T A)}\ \underset{2\times 1}{d} = \underset{2\times 1}{A^T b}$$

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

Lukas-Kanade flow

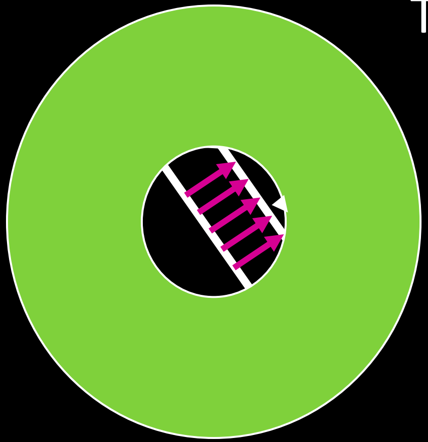
This technique was first proposed by Lukas & Kanade, 1981

$$\underset{25\text{x}2}{A}\;\underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

$$\underset{2\text{x}2}{(A^T A)}\;\underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

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

Aperture Problem and Normal Flow

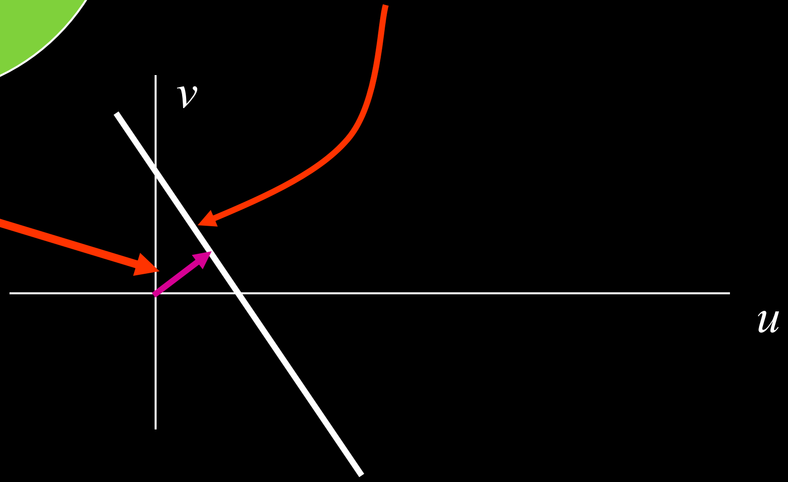


The gradient constraint:

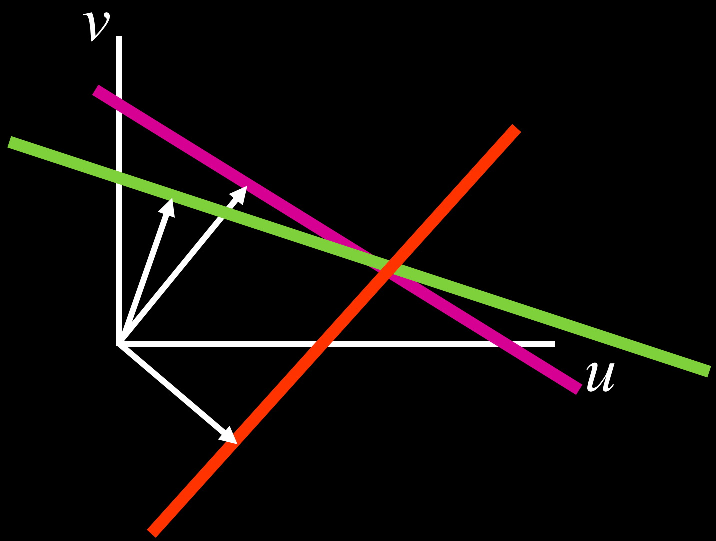
$$\boxed{\begin{aligned} I_x u + I_y v + I_t &= 0 \\ \nabla I \cdot \vec{U} + I_t &= 0 \end{aligned}}$$

Defines a line in the *(u,v)* space

Normal Flow:



Combining Local Constraints



$$\nabla I^1 \bullet U = -I_t^1$$

$$\nabla I^2 \bullet U = -I_t^2$$

$$\nabla I^3 \bullet U = -I_t^3$$

etc.

Conditions for solvability

When is This Solvable?

- $A^T A$ should be invertible
- => So $A^T A$ should be well-conditioned - λ_1/λ_2 should not be too large (λ_1= larger eigenvalue)

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

Conditions for solvability

When is This Solvable?

- Also $A^T A$ should be solvable when there is no aperture problem
  - Does this remind you of something???

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

Eigenvectors of ATA

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

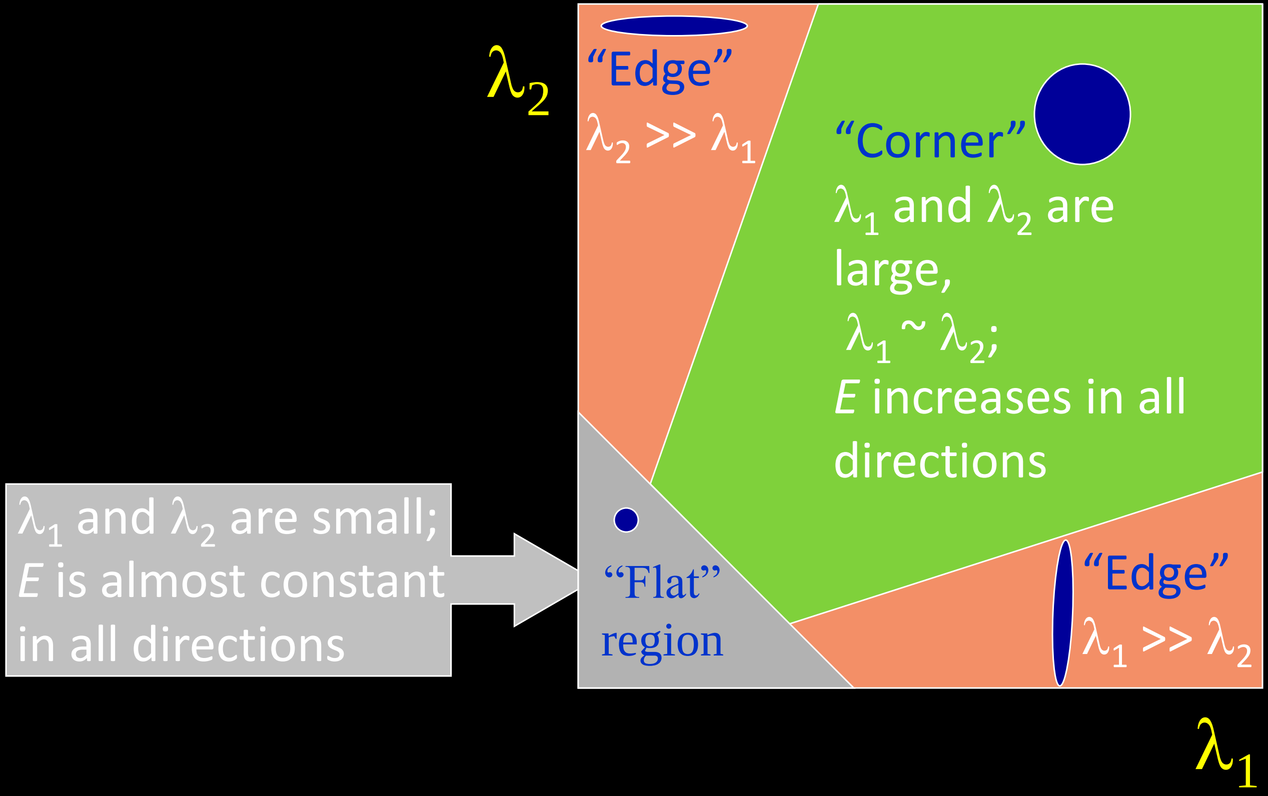
- Recall the Harris corner detector:

  $M = A^T A$ is the second moment matrix

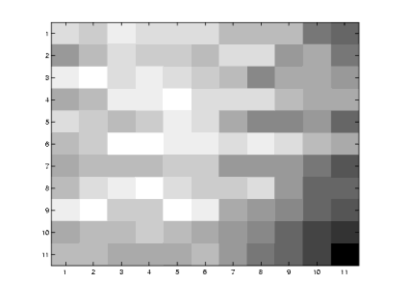
- The eigenvectors and eigenvalues of M relate to edge direction and magnitude

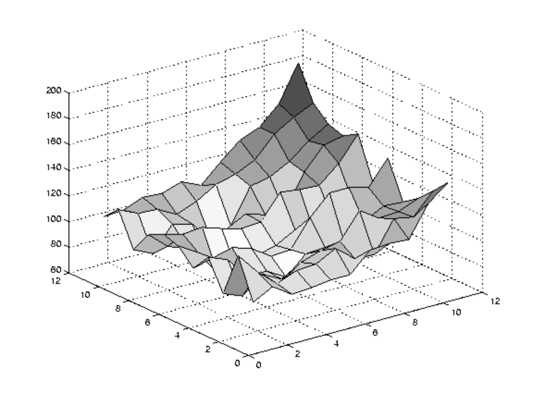
Interpreting the eigenvalues

Classification of image points using eigenvalues of *M*:



Low texture region



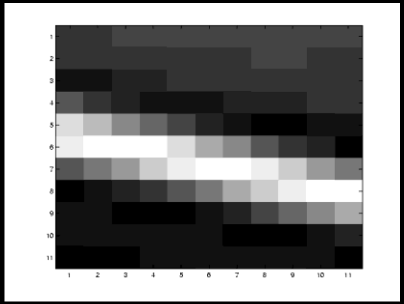


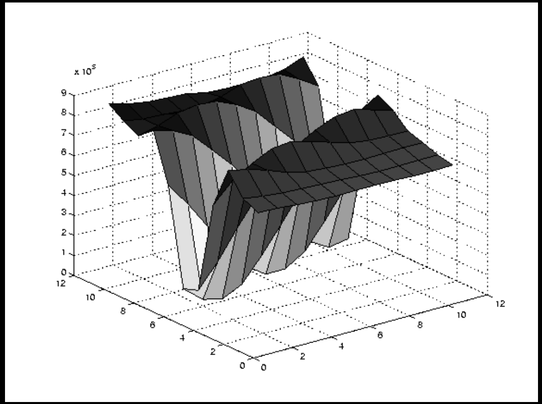
$$M = \sum \nabla I (\nabla I)^T$$

Gradients have small magnitude

=> small λ_1, small λ_2

Edge



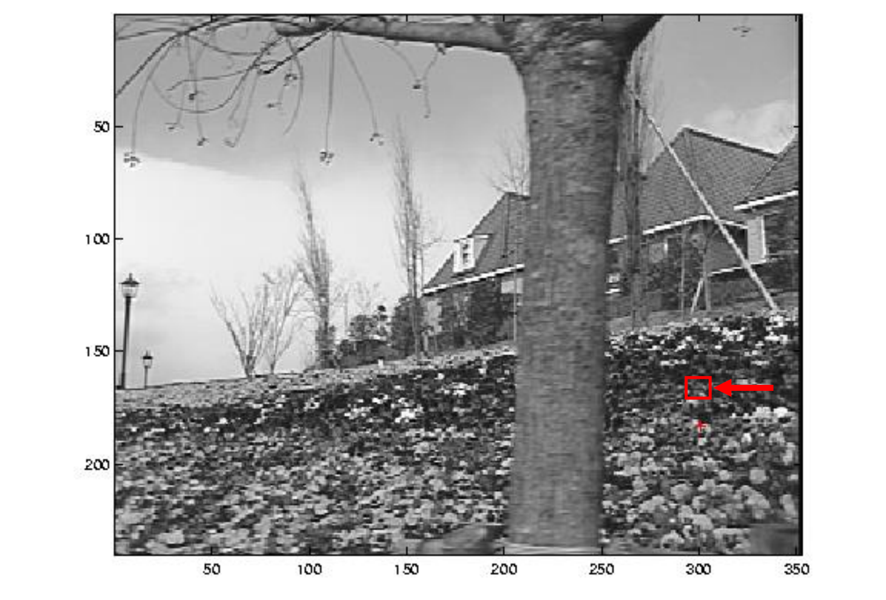


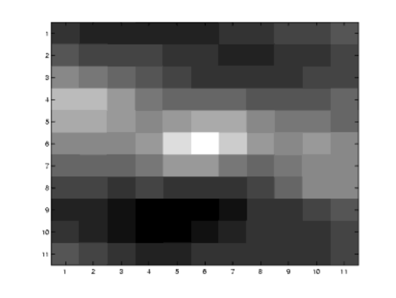
$$M = \sum \nabla I (\nabla I)^T$$

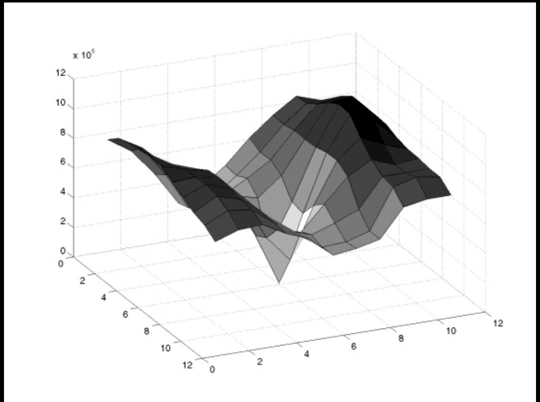
Large gradients, all the same

=> large λ_1, small λ_2

High textured region







$$M = \sum \nabla I (\nabla I)^T$$

Gradients different, large magnitudes

=> large λ_1, large λ_2

RGB version

- One method: pretend the pixel's neighbors have the ***same*** (u,v)
  - If we use a **5x5x3** window, that gives us 75 equations per pixel!

$$0 = I_t(\mathbf{p_i})[0,1,2] + \nabla I(\mathbf{p_i})[0,1,2] \cdot [u \;\; v]$$

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p}_1)[0] & I_y(\mathbf{p}_1)[0] \\ I_x(\mathbf{p}_1)[1] & I_y(\mathbf{p}_1)[1] \\ I_x(\mathbf{p}_1)[2] & I_y(\mathbf{p}_1)[2] \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25})[0] & I_y(\mathbf{p}_{25})[0] \\ I_x(\mathbf{p}_{25})[1] & I_y(\mathbf{p}_{25})[1] \\ I_x(\mathbf{p}_{25})[2] & I_y(\mathbf{p}_{25})[2] \end{bmatrix}}_{\substack{A \\ 75\text{x}2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{d \\ 2\text{x}1}} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p}_1)[0] \\ I_t(\mathbf{p}_1)[1] \\ I_t(\mathbf{p}_1)[2] \\ \vdots \\ I_t(\mathbf{p}_{25})[0] \\ I_t(\mathbf{p}_{25})[1] \\ I_t(\mathbf{p}_{25})[2] \end{bmatrix}}_{\substack{b \\ 75\text{x}1}}$$

RGB version

*Note that **RGB** alone at a pixel is not enough to disambiguate because R, G & B are correlated. Just provides better gradient.*

$$0 = I_t(\mathbf{p_i})[0,1,2] + \nabla I(\mathbf{p_i})[0,1,2] \cdot [u \;\; v]$$

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p}_1)[0] & I_y(\mathbf{p}_1)[0] \\ I_x(\mathbf{p}_1)[1] & I_y(\mathbf{p}_1)[1] \\ I_x(\mathbf{p}_1)[2] & I_y(\mathbf{p}_1)[2] \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25})[0] & I_y(\mathbf{p}_{25})[0] \\ I_x(\mathbf{p}_{25})[1] & I_y(\mathbf{p}_{25})[1] \\ I_x(\mathbf{p}_{25})[2] & I_y(\mathbf{p}_{25})[2] \end{bmatrix}}_{\substack{A \\ \text{75x2}}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{d \\ \text{2x1}}} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p}_1)[0] \\ I_t(\mathbf{p}_1)[1] \\ I_t(\mathbf{p}_1)[2] \\ \vdots \\ I_t(\mathbf{p}_{25})[0] \\ I_t(\mathbf{p}_{25})[1] \\ I_t(\mathbf{p}_{25})[2] \end{bmatrix}}_{\substack{b \\ \text{75x1}}}$$

Errors in Lucas-Kanade

- Brightness constancy does not hold
  - Do exhaustive neighborhood search with normalized correlation - tracking features – maybe SIFT – more later….

Errors in Lucas-Kanade

- A point does not move like its neighbors
  - Motion segmentation

Errors in Lucas-Kanade

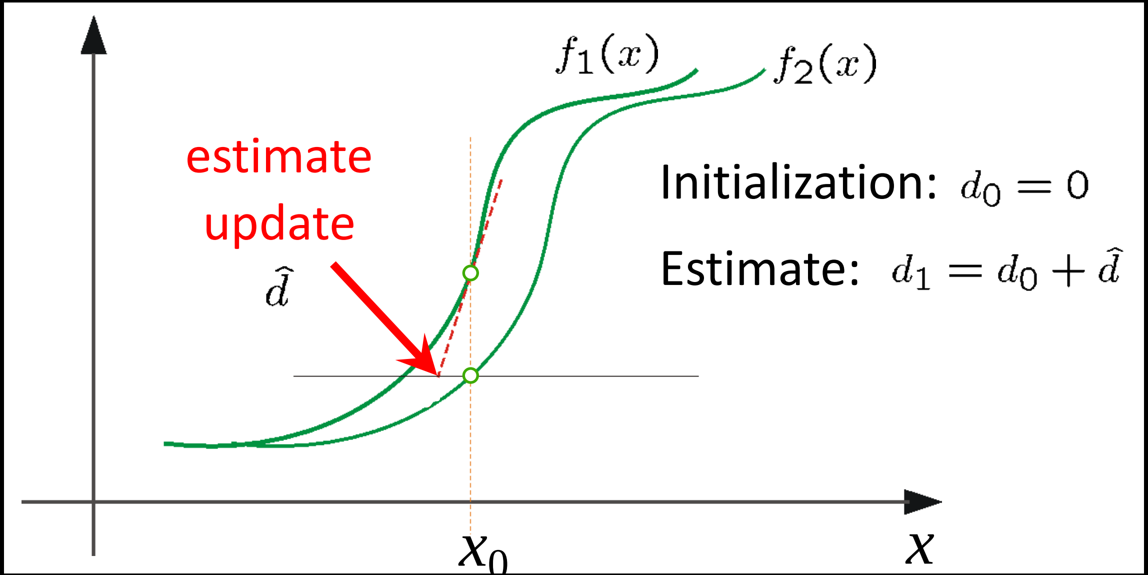
- The motion is large (larger than a pixel) – Taylor doesn’t hold
  - Not-linear: Iterative refinement
  - Local minima: coarse-to-fine estimation

Not tangent: Iterative Refinement

Iterative Lukas-Kanade Algorithm

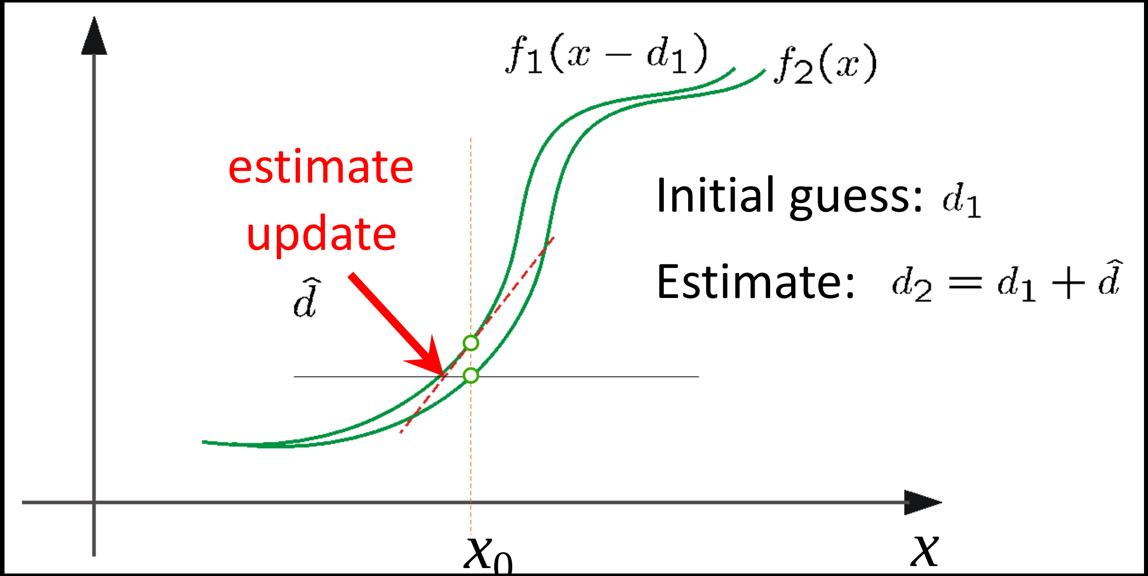
1. Estimate velocity at each pixel by solving Lucas-Kanade equations
2. Warp I_t towards I_{t+1} using the estimated flow field
   - *Use image warping techniques*
3. Repeat until convergence

Optical Flow: Iterative Estimation

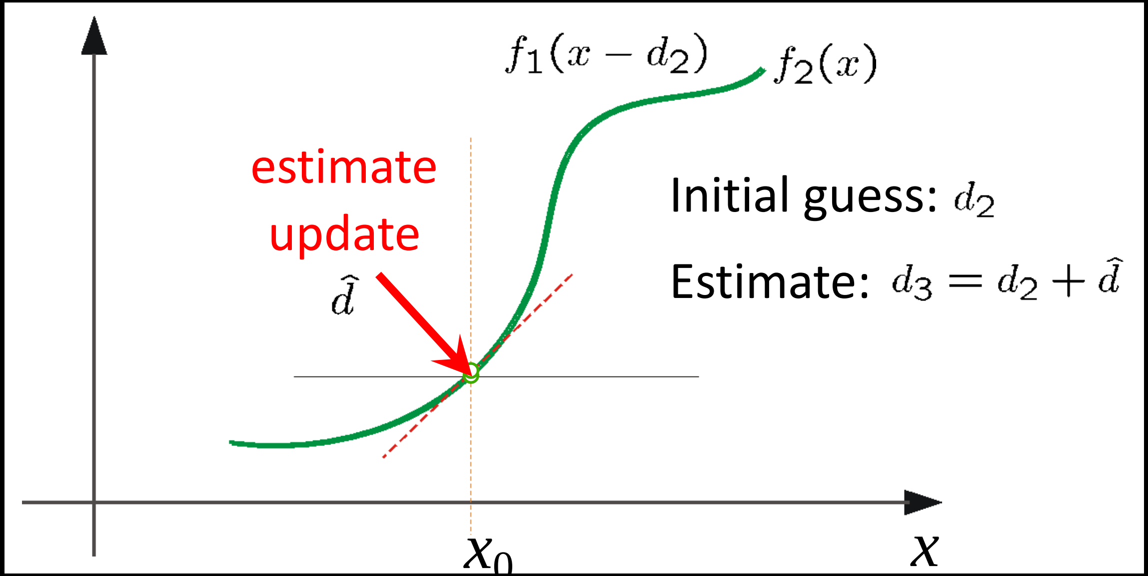


(using *d* for *displacement* here instead of *u*)

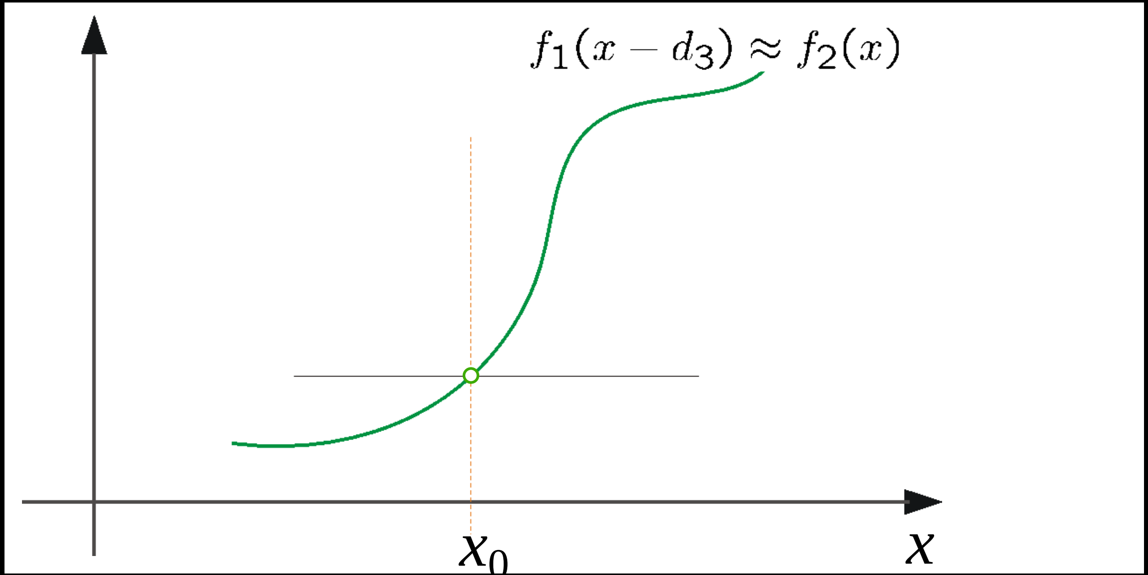
Optical Flow: Iterative Estimation



Optical Flow: Iterative Estimation



Optical Flow: Iterative Estimation



Optical Flow: Iterative Estimation

- Some Implementation Issues:
  - Warping is not easy (ensure that errors in warping are smaller than the estimate refinement) – but it is in MATLAB!
  - Often useful to low-pass filter the images before motion estimation (for better derivative estimation, and linear approximations to image intensity)