Lecture 15: Motion

Lucas-Kanade method

Juan Carlos Niebles and Jiajun Wu

CS131 Computer Vision: Foundations and Applications

What will we learn today?

- Lucas-Kanade method
  - Approach
  - Analysis

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005]
http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf

Solving the ambiguity…

- How to get more equations for a pixel?
- **Spatial coherence constraint:**
- Assume the pixel's neighbors have the same optical flow value (u, v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

$$\nabla I(p_i) \cdot \begin{bmatrix} u \\ v \end{bmatrix} + I_t(p_i) = 0$$

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \cdots & \cdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \cdots \\ I_t(p_{25}) \end{bmatrix}$$

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

Source: Silvio Savarese

Lucas-Kanade flow

- Over-constrained linear system:

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \cdots & \cdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \cdots \\ I_t(p_{25}) \end{bmatrix} \qquad \underset{25\times 2}{A} \; \underset{2\times 1}{d} = \underset{25\times 1}{b}$$

- The least squares solution for d is given by $(A^T A)\, d = A^T b$

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

- The summations are over all pixels in the 5 x 5 window

Source: Silvio Savarese

Lucas-Kanade flow

- The optimal (u, v) satisfies the Lucas-Kanade equation

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

- When is This Solvable?
  - $A^T A$ should be invertible
  - $A^T A$ should not be too small due to noise
    - eigenvalues λ_1 and λ_2 of $A^T A$ should not be too small
  - $A^T A$ should be well-conditioned
    - λ_1/λ_2 should not be too large (λ_1 = larger eigenvalue)
- Do the contents of $A^T A$ and these conditions resemble anything familiar?

Source: Silvio Savarese

M = $A^T A$ is the second moment matrix ! (Harris corner detector…)

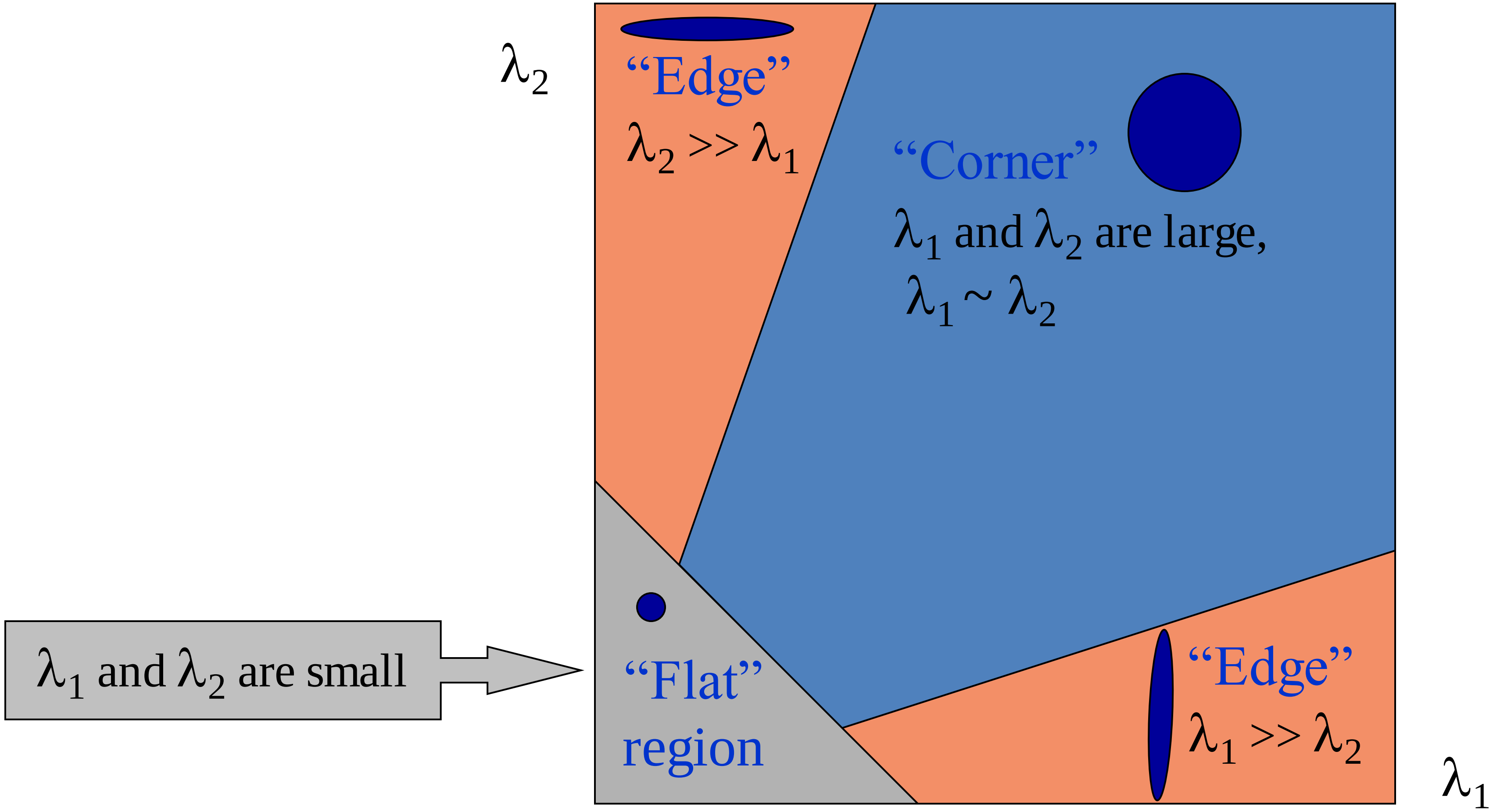
$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \quad I_y] = \sum \nabla I (\nabla I)^T$$

- Eigenvectors and eigenvalues of $A^T A$ relate to edge direction and magnitude
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change
  - The other eigenvector is orthogonal to it

Source: Silvio Savarese

Interpreting the eigenvalues

- Classification of image points using eigenvalues of the second moment matrix:



Source: Silvio Savarese

Edge

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

– gradients very large or very small
– large λ_1, small λ_2

Source: Silvio Savarese

Low-texture region

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

– gradients have small magnitude
– small λ_1, small λ_2

Source: Silvio Savarese

High-texture region

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

– gradients are different, large magnitudes

– large λ_1, large λ_2

Source: Silvio Savarese

Improving accuracy

- Recall our small motion assumption

$$0 = I(x+u, y+v) - I_t(x,y)$$

$$\approx I(x,y) + I_x u + I_y v - I_t(x,y)$$

- This is not exact
  - To do better, we need to add higher order terms back in:

$$= I(x,y) + I_x u + I_y v + \text{higher order terms} - I_t(x,y)$$

- This is a polynomial root finding problem
  - Can solve using **Newton's method (out of scope for this class)**
  - Lukas-Kanade method does one iteration of Newton's method
    - Better results are obtained via more iterations

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Iterative Refinement

- Iterative Lukas-Kanade Algorithm
  1. Estimate velocity at each pixel by solving Lucas-Kanade equations
  2. Warp $I(t-1)$ towards $I(t)$ using the estimated flow field
     - *use image warping techniques*
  3. Repeat until convergence

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Sources of Error in Lukas-Kanade

- What are the potential causes of errors in this procedure?
  – Assumed $A^T A$ is easily invertible
  – Assumed there is not much noise in the image

- When our assumptions are violated
  – The motion is **not** small
  – Brightness constancy is **not** satisfied
  – A point does **not** move like its neighbors
    - window size is too large
    - what is the ideal window size?

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Summary

- Lucas-Kanade method
  - Approach
  - Analysis

Reading: [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005]
http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf