Announcement

We will have Quiz 3 in class, Monday, March 20th.

Minor Issues in Project 1

Result is reasonable, but not accurate

Part 1:

- Need more points for training
- Ensure the points sampled from two planes

Part 2:

- Insufficient iterations - estimate number of iterations using the probability of the outliers
- Insufficient inliers - the choice of K & threshold

Today

Early vision on multiple images

- Motion estimation

Early Vision on Multiple Images: Motion Estimation

Image content in different frames varies because of the relative motion between the camera and the scene

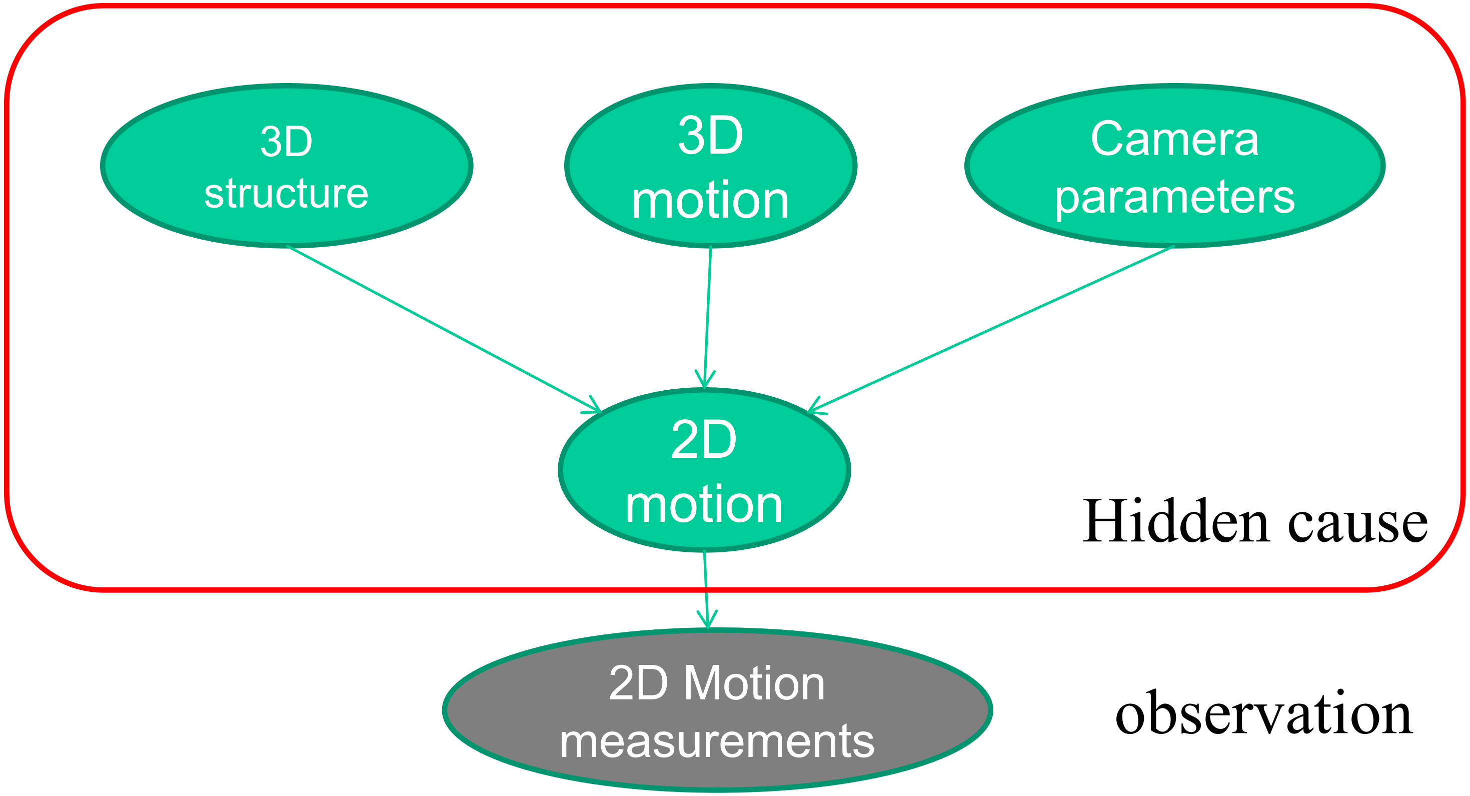
Motion estimation can answer the questions

- How many moving objects in the scene?
- What are the moving directions of the objects?
- What are the moving speeds of the objects?
- What are the structures of the moving objects?

Different conditions:

- Moving objects/scene and static scene/objects
- Moving objects and moving scene

Motion Modeling and Analysis



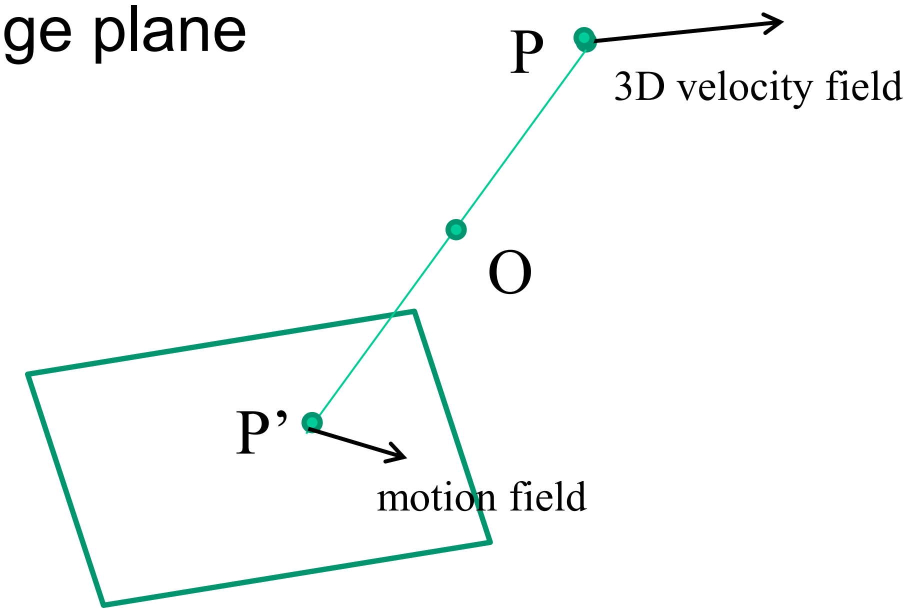
Tasks:

- Estimate 2D motion represented by ***motion field***
- Infer the 3D motion and/or 3D structure

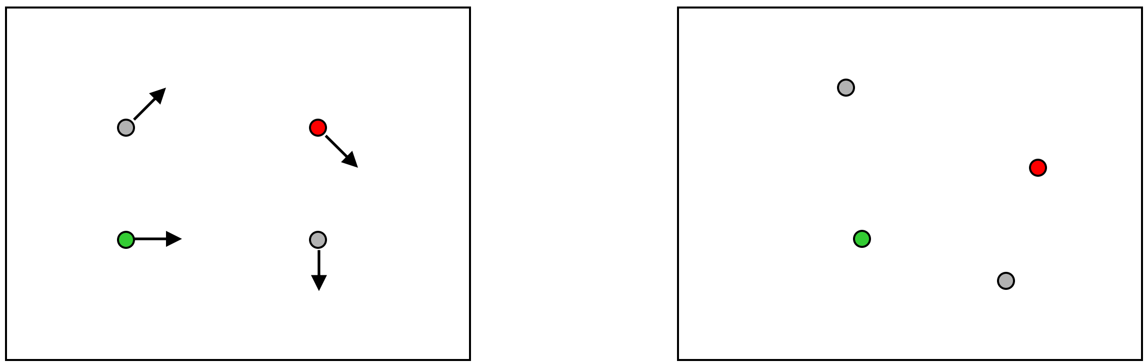
Motion Analysis

Motion field:

- For each image point, a *motion field* is defined as its 2D vector field of velocities caused by the relative 3D motion between the camera and the scene.
- Can be interpreted as the projection of 3D velocity field on the image plane



Optical Flow



$H(x, y)$ $I(x, y)$

How to estimate pixel motion from image *H* to image *I*?

- First, solve pixel correspondence problem
  - given a pixel in ***H***, look for nearby pixels of the same color in ***I***

Key assumptions

- **color constancy:** a point in H looks the same in I
  - For grayscale images, this is **brightness constancy**
- **small motion**: points do not move very far

This is called the optical flow **problem**

Optical Flow v.s. Motion Field

Optical flow: the **apparent motion** of **brightness patterns**

Most of the time, optical flow corresponds to motion field

Optical flow ≠ Motion field

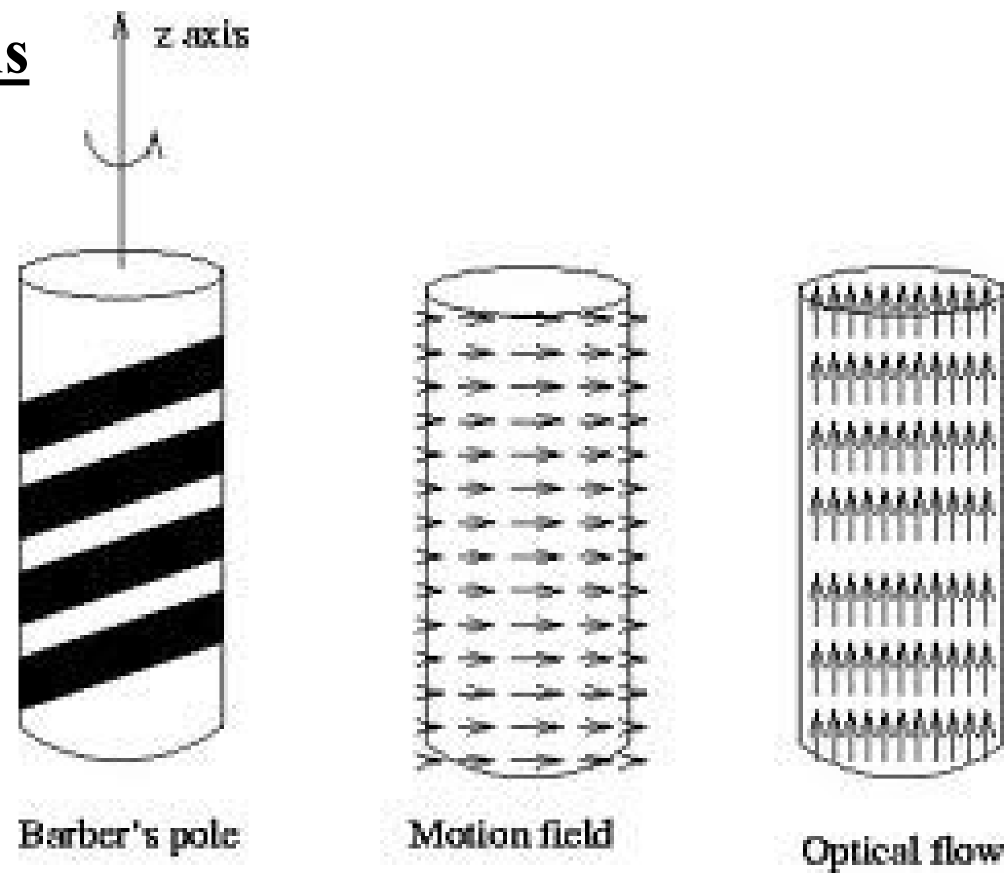
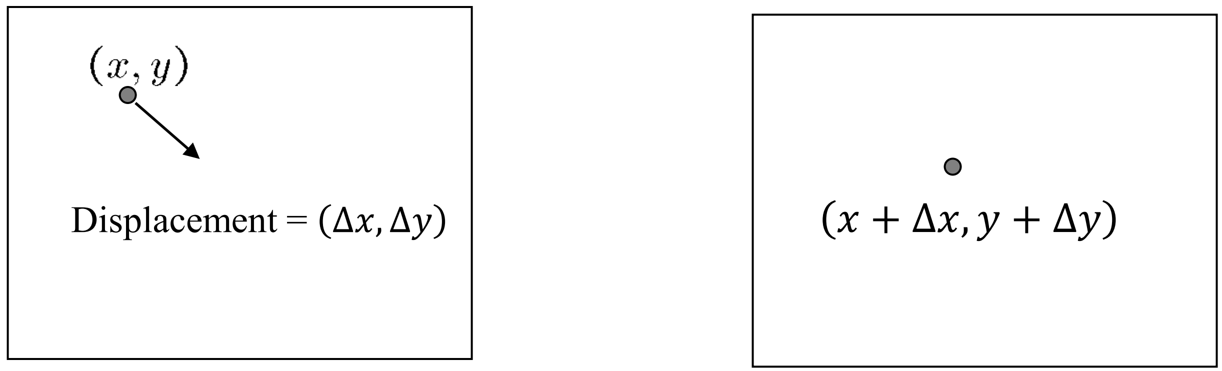


Figure from http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT12/node4.html

Optical Flow Constraints (Grayscale Images)



$I(x, y, t)$ $\qquad$ $I(x, y, t + \Delta t)$

- **Brightness constancy**: $\quad I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x, y, t)$

- **Small motion**:
  - Take the Taylor series expansion of $I(x + \Delta x, y + \Delta y, t + \Delta t)$:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta \mathrm{y} + \frac{\partial I}{\partial t}\Delta t + \text{high order terms}$$

Optical Flow Equation

Combining these two equations

$$I(x+\Delta x, y+\Delta y, t+\Delta t) - I(x,y,t) = 0$$

$$\Rightarrow \quad \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0$$

$$\Rightarrow \quad \frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} = 0$$

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

$$\Rightarrow \quad I_x v_x + I_y v_y + I_t = 0 \qquad \nabla I = [I_x \quad I_y]^T$$ Gradient

$$\Rightarrow \quad \nabla I \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} + I_t = 0$$

Unknown 2D motion field $I_t = I(x,y,t+1) - I(x,y,t)$

Optical-Flow Equation

$$\nabla I \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} + I_t = 0$$

Q: how many unknowns and equations per pixel?

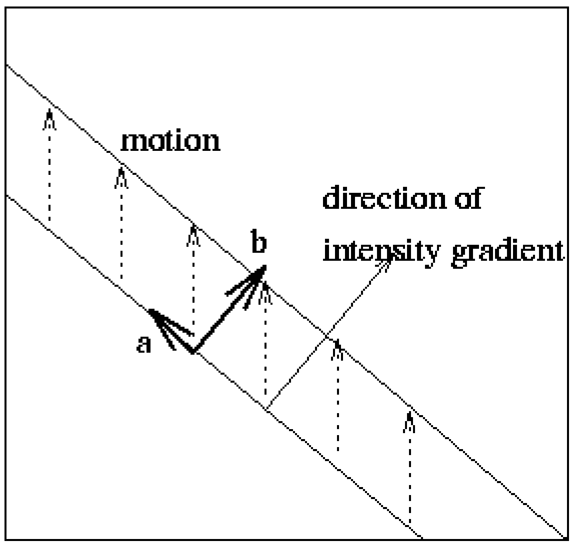
Two unknowns and one equation

How can we solve the equation?

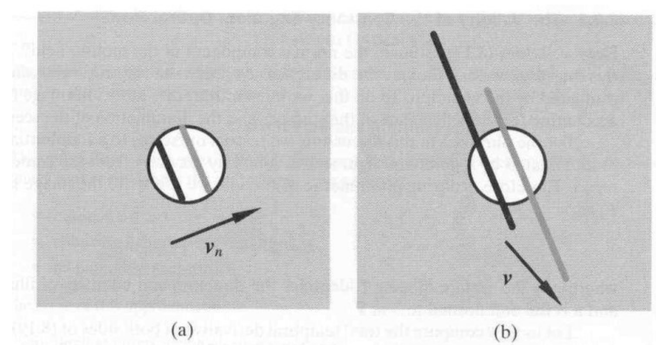
Aperture Problem

The motion field consists of two components:

- a component of the flow in the gradient direction
- a component of the flow parallel to an edge (perpendicular to the gradient)



$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = b \cdot \nabla I + a \cdot \nabla I_{\perp}$$



We can only measure the component *b* due to the aperture problem

Solving the Aperture Problem

How to get more equations for a pixel?

- Basic idea: impose additional constraints
  - most common is to assume that the flow field is smooth locally
  - one method: pretend the pixel's neighbors have the same (v_x, v_y)
    - If we use a 5x5 window, that gives us 25 equations per pixel!

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot \begin{bmatrix} v_x & v_y \end{bmatrix}$$

$$\underset{\substack{A \\ 25\times 2}}{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix}} \underset{\substack{\mathbf{v} \\ 2\times 1}}{\begin{bmatrix} v_x \\ v_y \end{bmatrix}} = - \underset{\substack{b \\ 25\times 1}}{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}}$$

RGB Version

How to get more equations for a pixel?

- If we use a 5x5 window, that gives us 25*3 equations per pixel!

$$0 = I_t(\mathbf{p_i})[0,1,2] + \nabla I(\mathbf{p_i})[0,1,2] \cdot \begin{bmatrix} v_x & v_y \end{bmatrix}$$

$$\begin{bmatrix} I_x(\mathbf{p_1})[0] & I_y(\mathbf{p_1})[0] \\ I_x(\mathbf{p_1})[1] & I_y(\mathbf{p_1})[1] \\ I_x(\mathbf{p_1})[2] & I_y(\mathbf{p_1})[2] \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}})[0] & I_y(\mathbf{p_{25}})[0] \\ I_x(\mathbf{p_{25}})[1] & I_y(\mathbf{p_{25}})[1] \\ I_x(\mathbf{p_{25}})[2] & I_y(\mathbf{p_{25}})[2] \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1})[0] \\ I_t(\mathbf{p_1})[1] \\ I_t(\mathbf{p_1})[2] \\ \vdots \\ I_t(\mathbf{p_{25}})[0] \\ I_t(\mathbf{p_{25}})[1] \\ I_t(\mathbf{p_{25}})[2] \end{bmatrix}$$

$$\underset{75\text{x}2}{A} \qquad \underset{2\times 1}{\mathbf{v}} \qquad \underset{75\text{x}1}{b}$$

Lukas-Kanade Flow

Problem: we have more equations than unknowns

$$\underset{25\times 2}{\mathbf{A}}\ \underset{2\times 1}{\mathbf{v}} = \underset{25\times 1}{\mathbf{b}} \longrightarrow \text{minimize} \quad \|\mathbf{Av}-\mathbf{b}\|^2$$

Solution: solve least squares problem

- minimum least squares solution given by solution of:

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

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

- The summations are over all pixels in the *K* x *K* window
- This technique was first proposed by Lukas & Kanade (1981)
  - described in Trucco & Verri reading

Conditions for Solvability

- Optimal $[v_x \quad v_y]$ satisfies Lucas-Kanade equation

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

When is this Solvable?

- **A^TA** should be invertible
- **A^TA** should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of **A^TA** should not be too small
- **A^TA** should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Errors in Lukas-Kanade

What are the potential causes of errors in this procedure?

- Suppose A^TA is easily invertible
- Suppose there is not much noise in the image

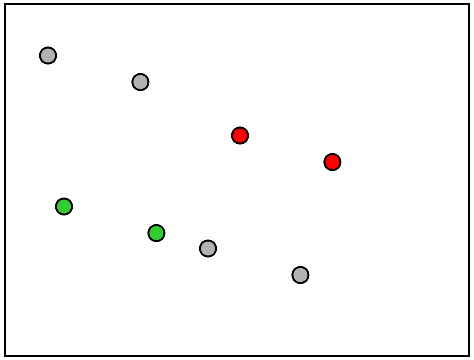
When our assumptions are violated

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

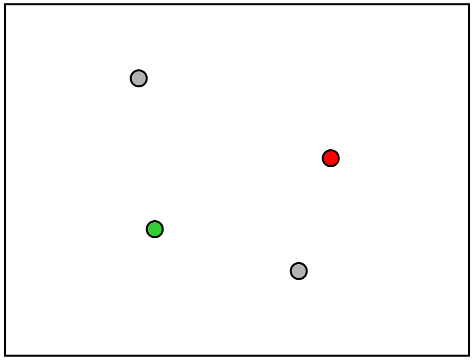
Iterative Refinement

Iterative Lukas-Kanade Algorithm

1. Estimate velocity at each pixel by solving Lucas-Kanade equations
2. Warp $\mathbf{H}$ towards $\mathbf{I}$ using the estimated flow field
   - *use image warping techniques*
3. Repeat until convergence



$H(x, y)$



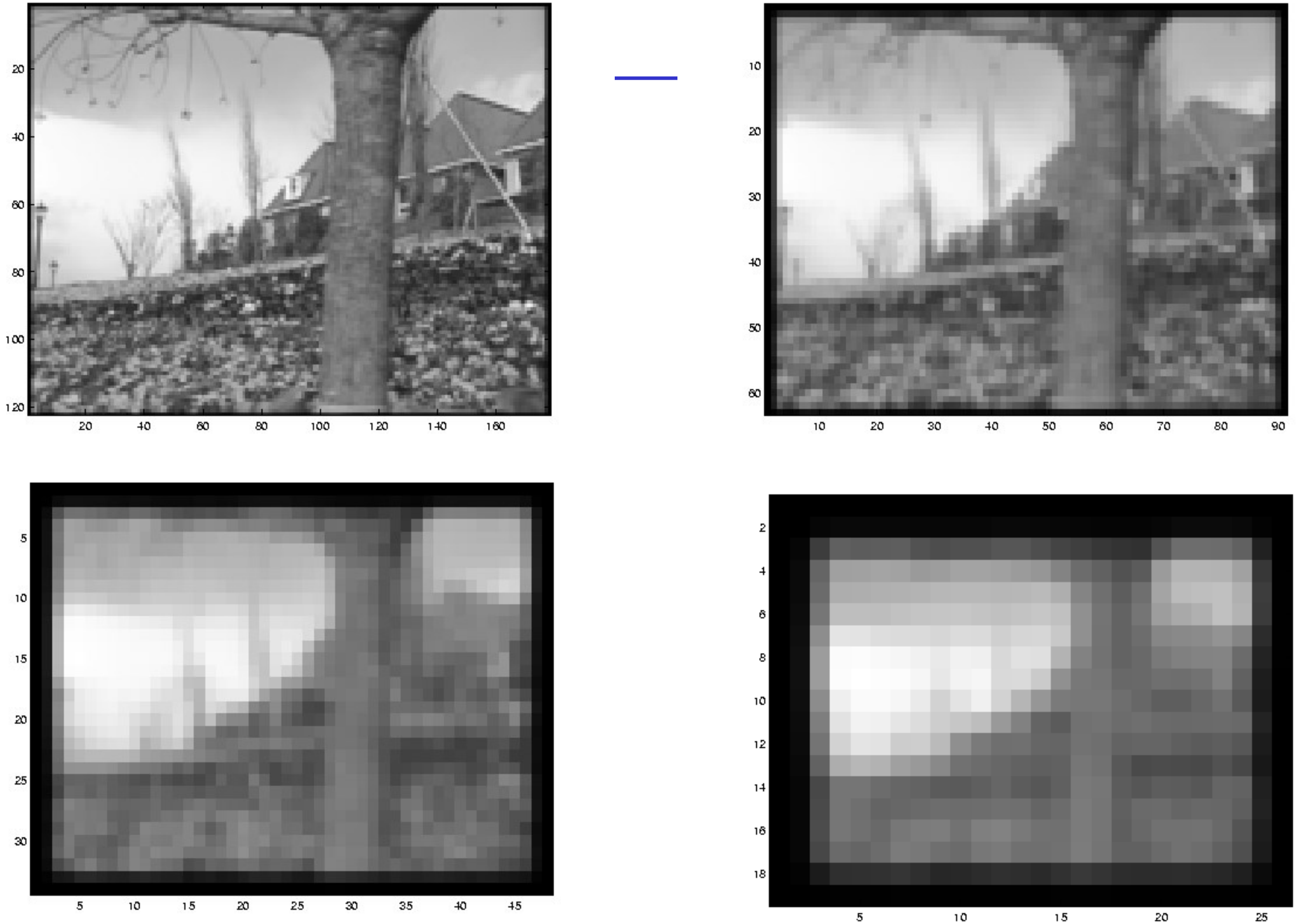
$I(x, y)$

Revisiting the Small Motion Assumption

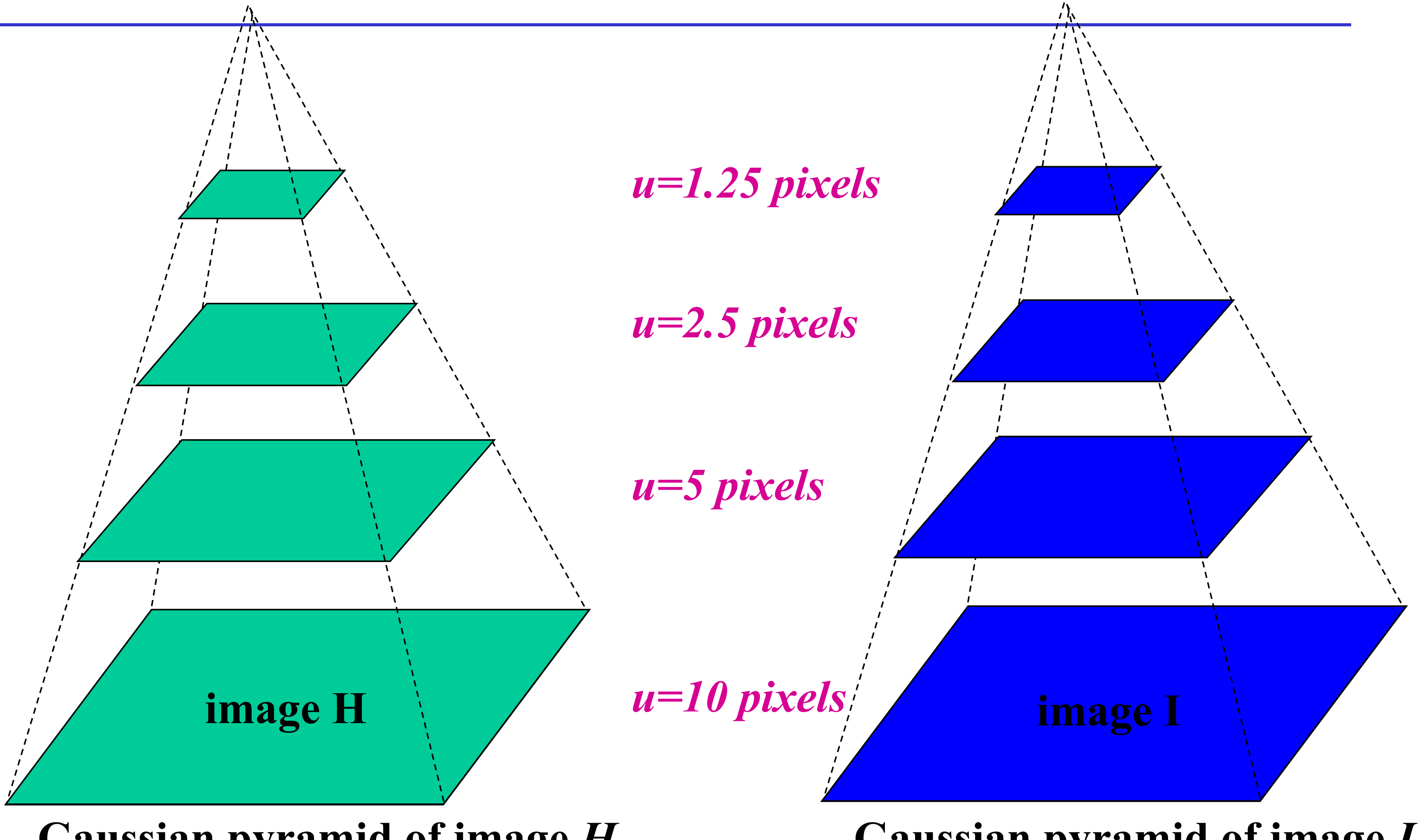
Is this motion small enough?

- Probably not—it's much larger than one pixel (2nd order terms dominate)
- How might we solve this problem?

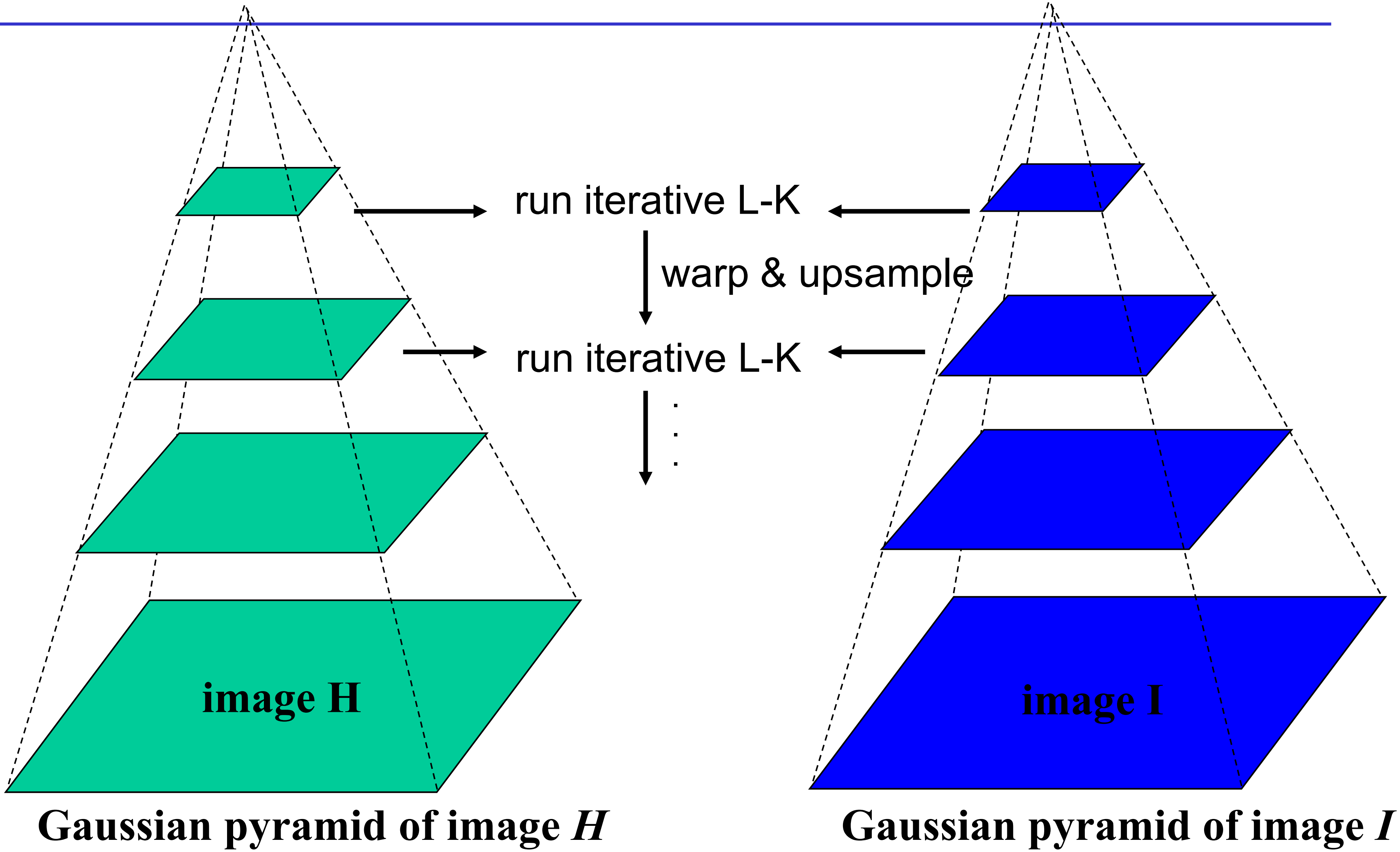
Reduce the Resolution!



Coarse-to-fine Optical Flow Estimation



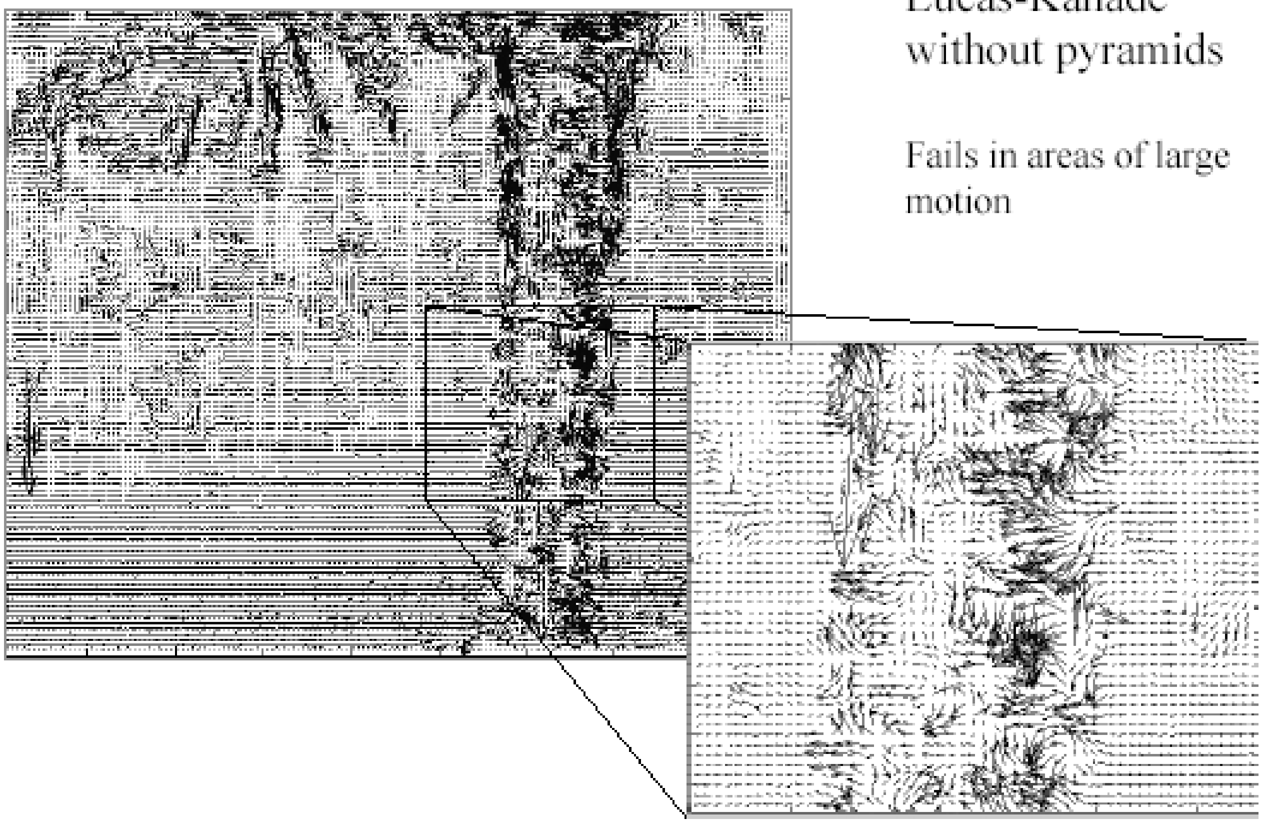
Coarse-to-fine Optical Flow Estimation



Multi-resolution Lucas-Kanade Algorithm

- Compute 'simple' LK at highest level
- At level i
  - Take flow u_{i-1}, v_{i-1} from level i-1
  - bilinear interpolate it to create u_i^*, v_i^* matrices of twice resolution for level i
  - multiply u_i^*, v_i^* by 2
  - compute f_t from a block displaced by $u_i^*(x,y)$, $v_i^*(x,y)$
  - Apply LK to get $u_i'(x, y)$, $v_i'(x, y)$ (the correction in flow)
  - Add corrections u_i' v_i', *i.e.* $u_i = u_i^* + u_i'$, $v_i = v_i^* + v_i'$.

Optical Flow Results



Lucas-Kanade without pyramids

Fails in areas of large motion

Optical Flow Results