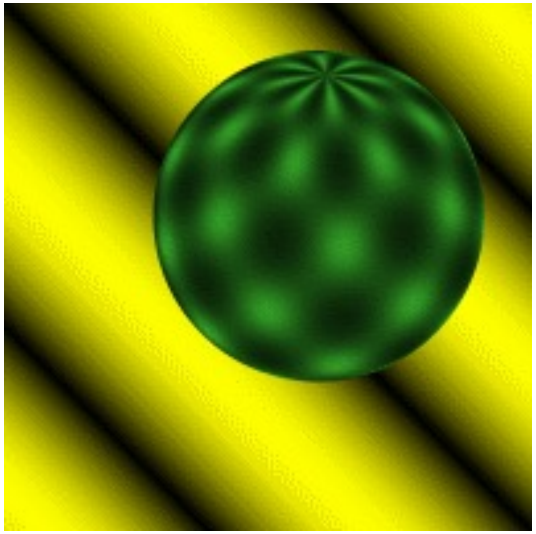
Motion and Optical Flow



We live in a moving world

- Perceiving, understanding and predicting motion is an important part of our daily lives

Motion and perceptual organization

- Even “impoverished” motion data can evoke a strong percept

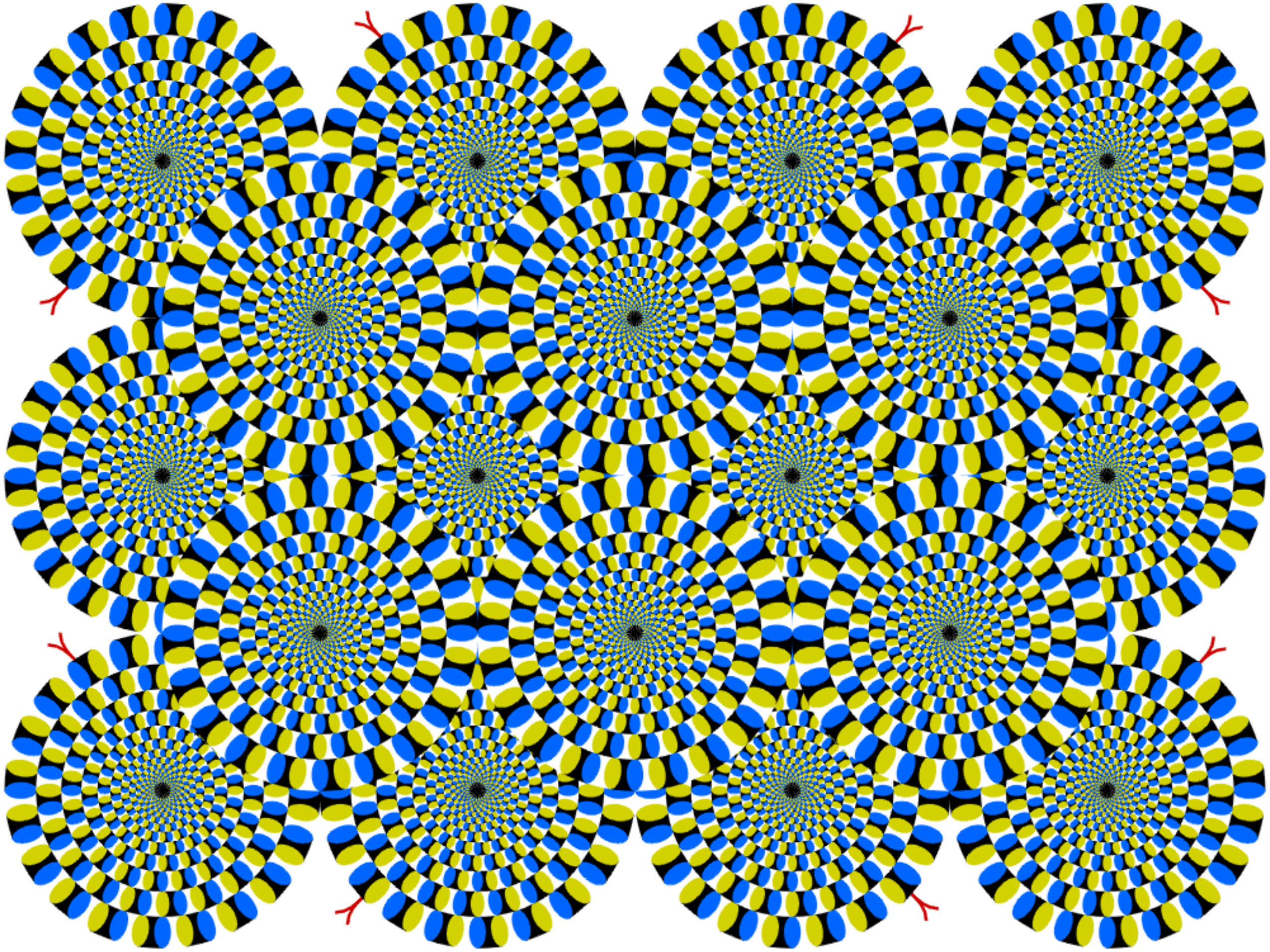
G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

Motion and perceptual organization

- Even “impoverished” motion data can evoke a strong percept

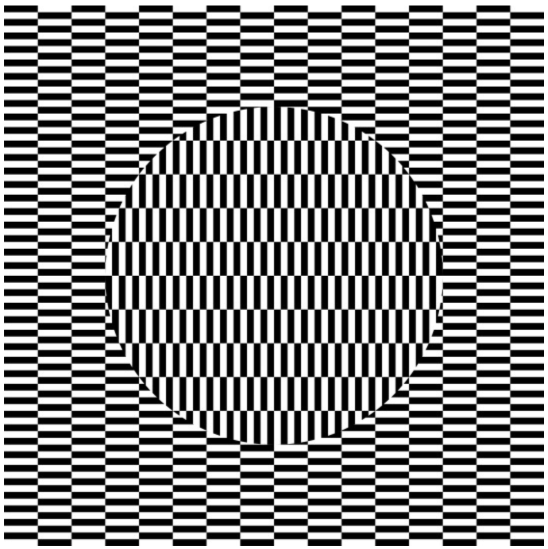
G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

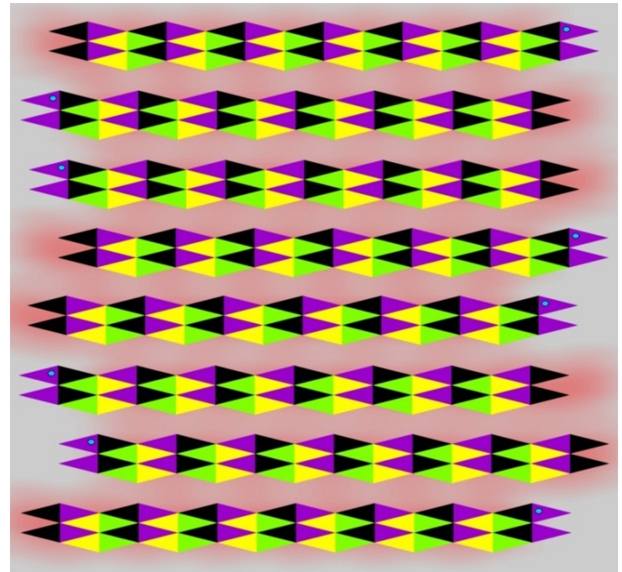
Seeing motion from a static picture?

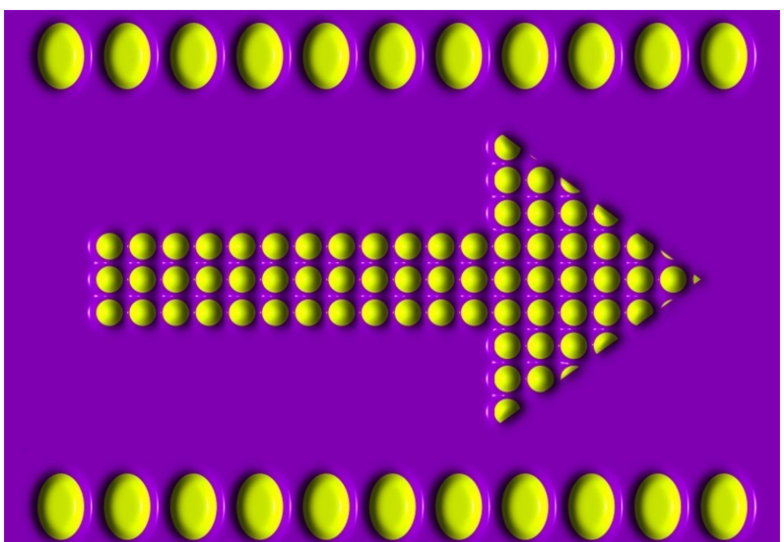


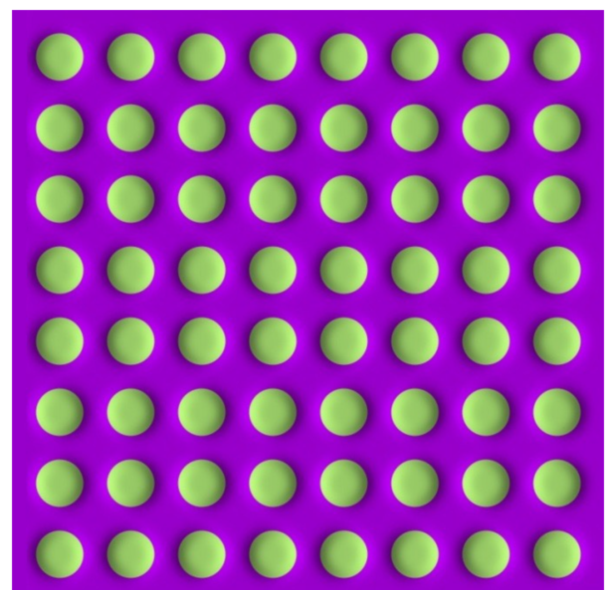
http://www.ritsumei.ac.jp/~akitaoka/index-e.html

More examples



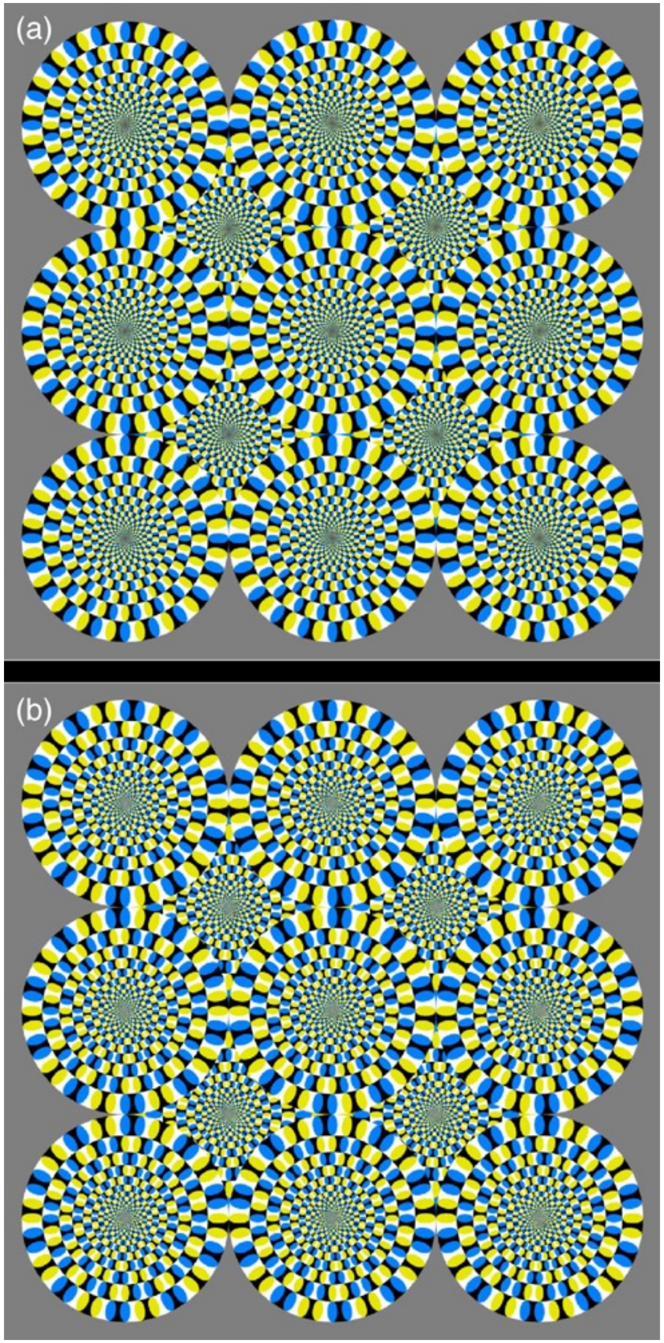






How is this possible?

- The true mechanism is yet to be revealed
- FMRI data suggest that illusion is related to some component of eye movements
- We don't expect computer vision to "see" motion from these patterns



The cause of motion

- Three factors in imaging process
  - Light
  - Object
  - Camera
- Varying either of them causes motion
  - Static camera, moving objects (surveillance)
  - Moving camera, static scene (3D capture)
  - Moving camera, moving scene (sports, movie)
  - Static camera, moving objects, moving light (time lapse)

Recovering motion

- Feature-tracking
  - Extract visual features (corners, textured areas) and "track" them over multiple frames

- Optical flow
  - Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)

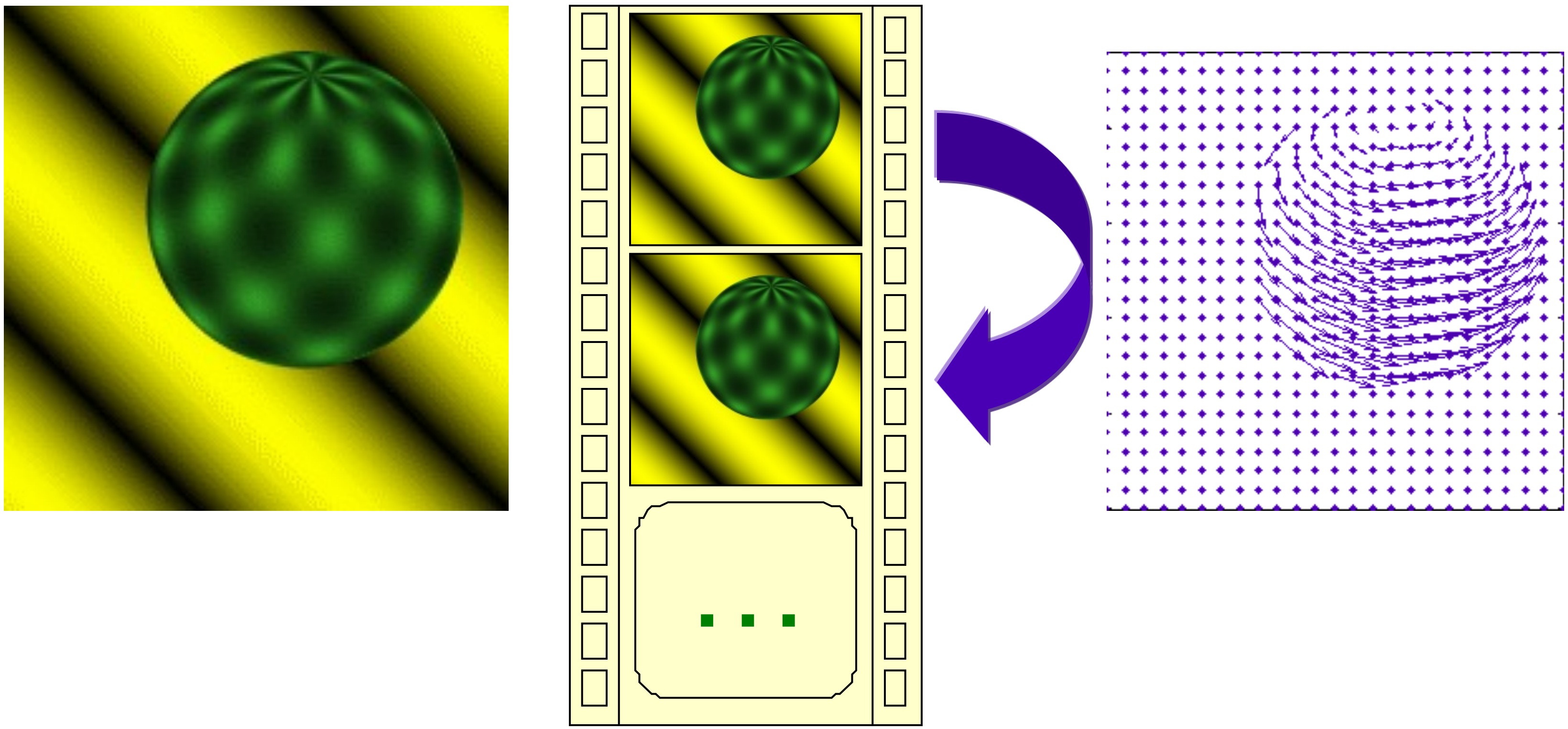
Two problems, one registration method

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.

Feature tracking

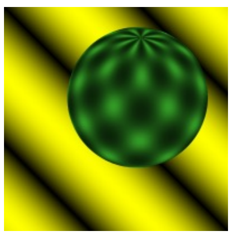
- Challenges
  – Figure out which features can be tracked
  – Efficiently track across frames
  – Some points may change appearance over time (e.g., due to rotation, moving into shadows, etc.)
  – Drift: small errors can accumulate as appearance model is updated
  – Points may appear or disappear: need to be able to add/delete tracked points

What is Optical Flow?



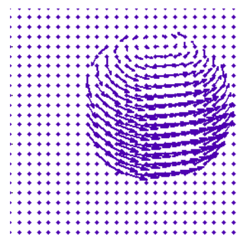
Definitions

The **optical flow** is a velocity field in the image which transforms one image into the next image in a sequence [Horn & Schunck]



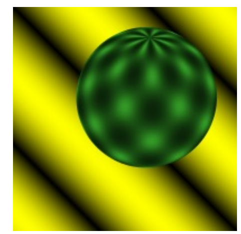
frame #1

+



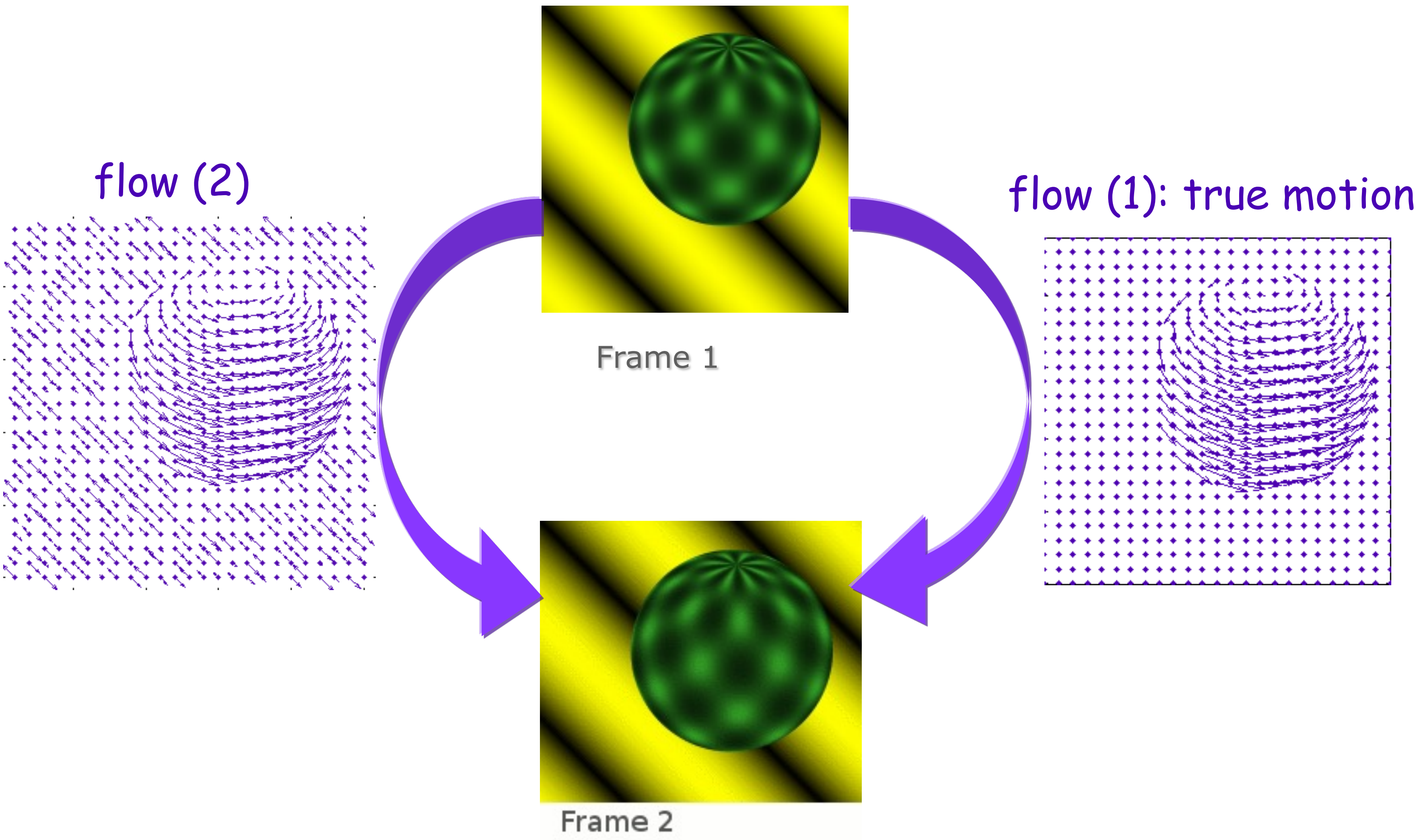
flow field

=

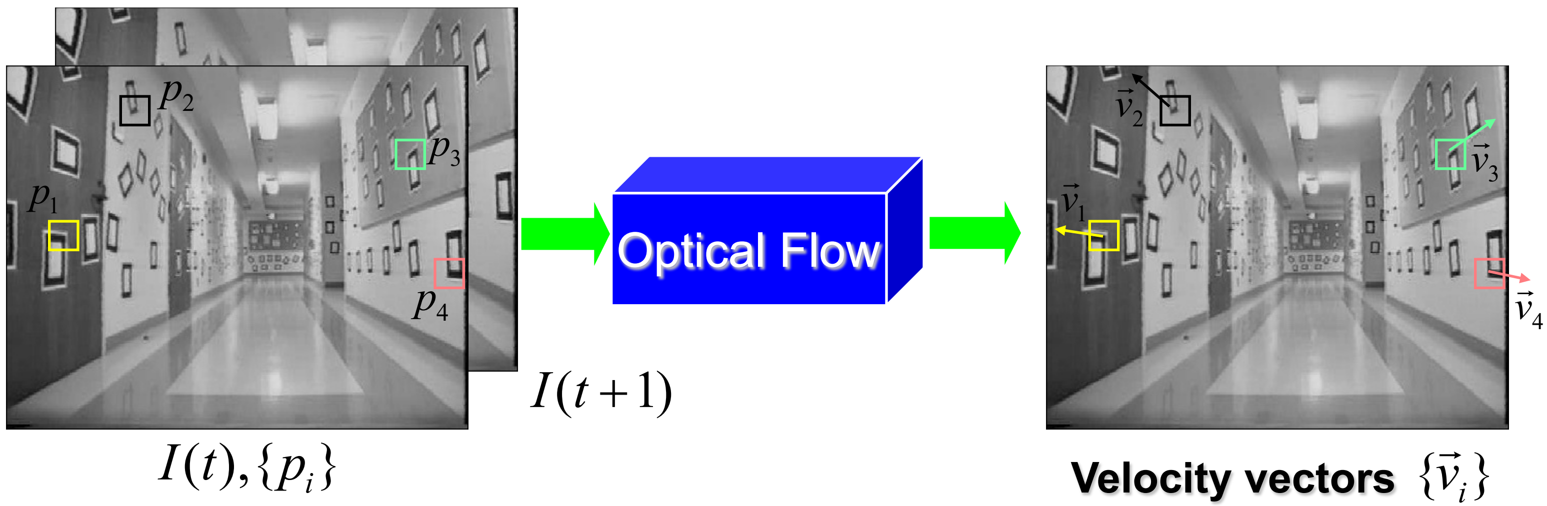


frame #2

Ambiguity of optical flow

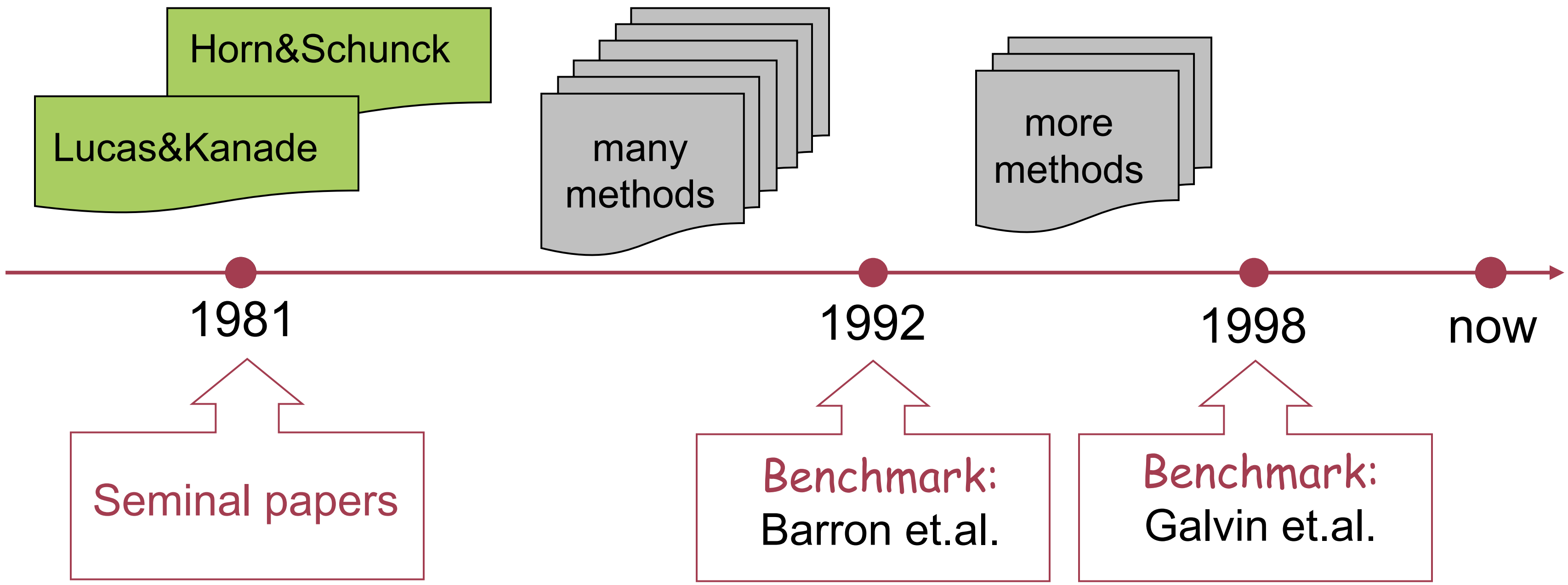


What is Optical Flow?

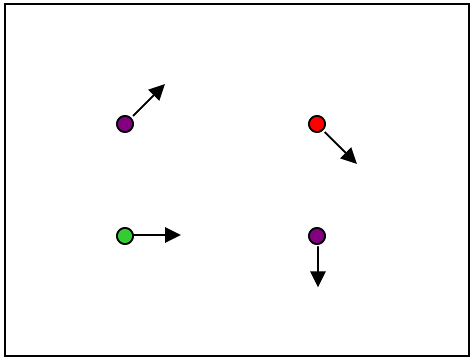


Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene -Wiki

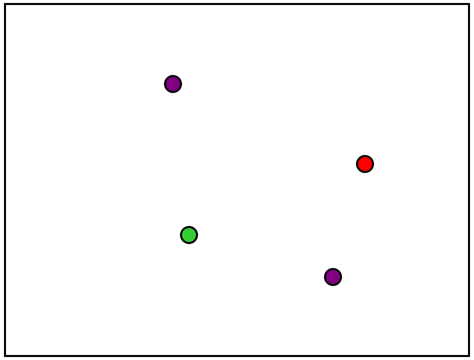
Optical Flow Research: Timeline



Optical Flow



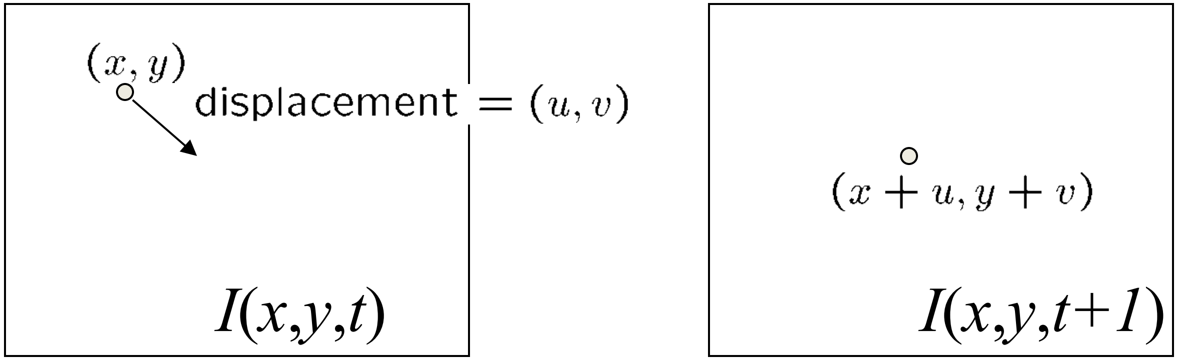
$I(x,y,t)$



$I(x,y,t+1)$

- Given two subsequent frames, estimate the point translation
- Key assumptions of Lucas-Kanade Tracker
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Small motion:** points do not move very far
  - **Spatial coherence:** points move like their neighbors

The brightness constancy constraint



- Brightness Constancy Equation:

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

Take Taylor expansion of $I(x+u,\ y+v,\ t+1)$ at (x,y,t) to linearize the right side:

Image derivative along x   Difference over frames

$$I(x+u, y+v, t+1) \approx I(x,y,t) + \boxed{I_x} \cdot u + I_y \cdot v + \boxed{I_t}$$

$$I(x+u, y+v, t+1) - I(x,y,t) = +I_x \cdot u + I_y \cdot v + I_t$$

So: $$I_x \cdot u + I_y \cdot v + I_t \approx 0 \quad \rightarrow \nabla \mathrm{I} \cdot [\mathrm{u} \ \ \mathrm{v}]^{\mathrm{T}} + \mathrm{I}_{\mathrm{t}} = 0$$

The brightness constancy constraint

Can we use this equation to recover image motion (u,v) at each pixel?

$$\nabla \mathrm{I} \cdot \left[\mathrm{u} \;\; \mathrm{v}\right]^{\mathrm{T}} + \mathrm{I}_{\mathrm{t}} = 0$$

- How many equations and unknowns per pixel?
  - One equation, two unknowns (u,v)

Optical Flow

Notation

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

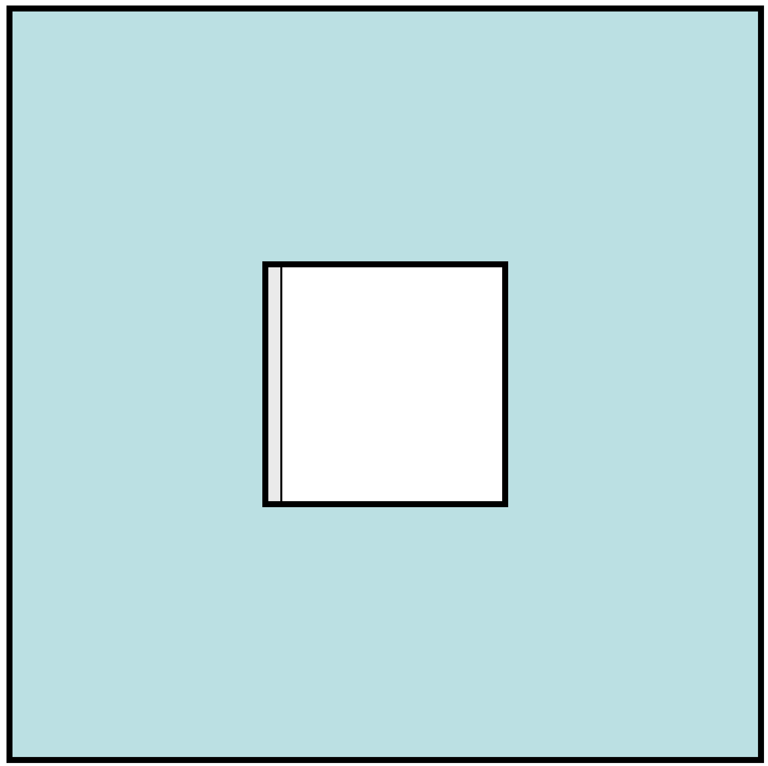
$$\nabla I^T \mathbf{u} = -I_t$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \quad \nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

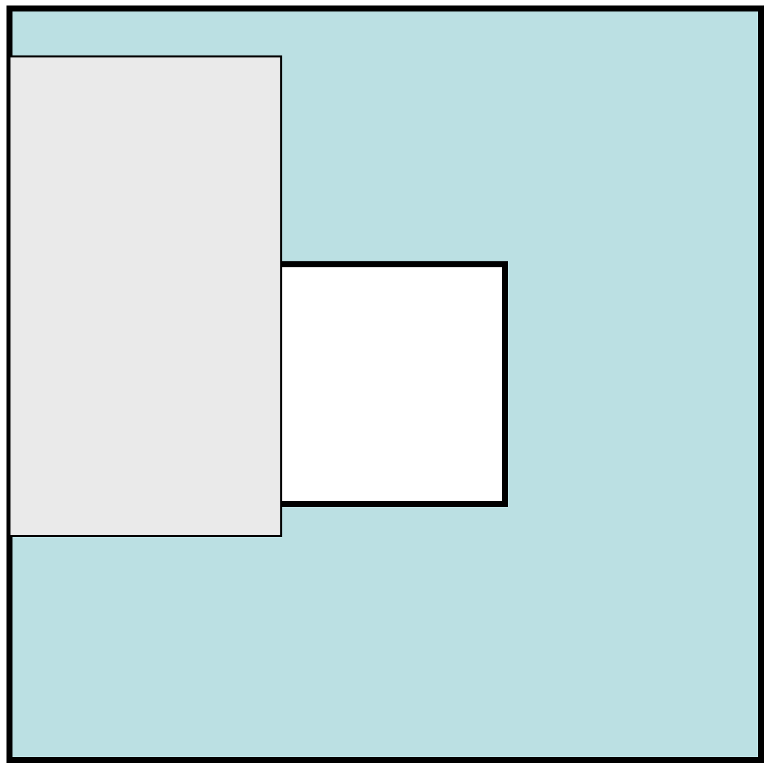
At a single image pixel, we get a line:

$$I_x u + I_y v = -I_t$$

How does this show up visually?
Known as the “Aperture Problem”

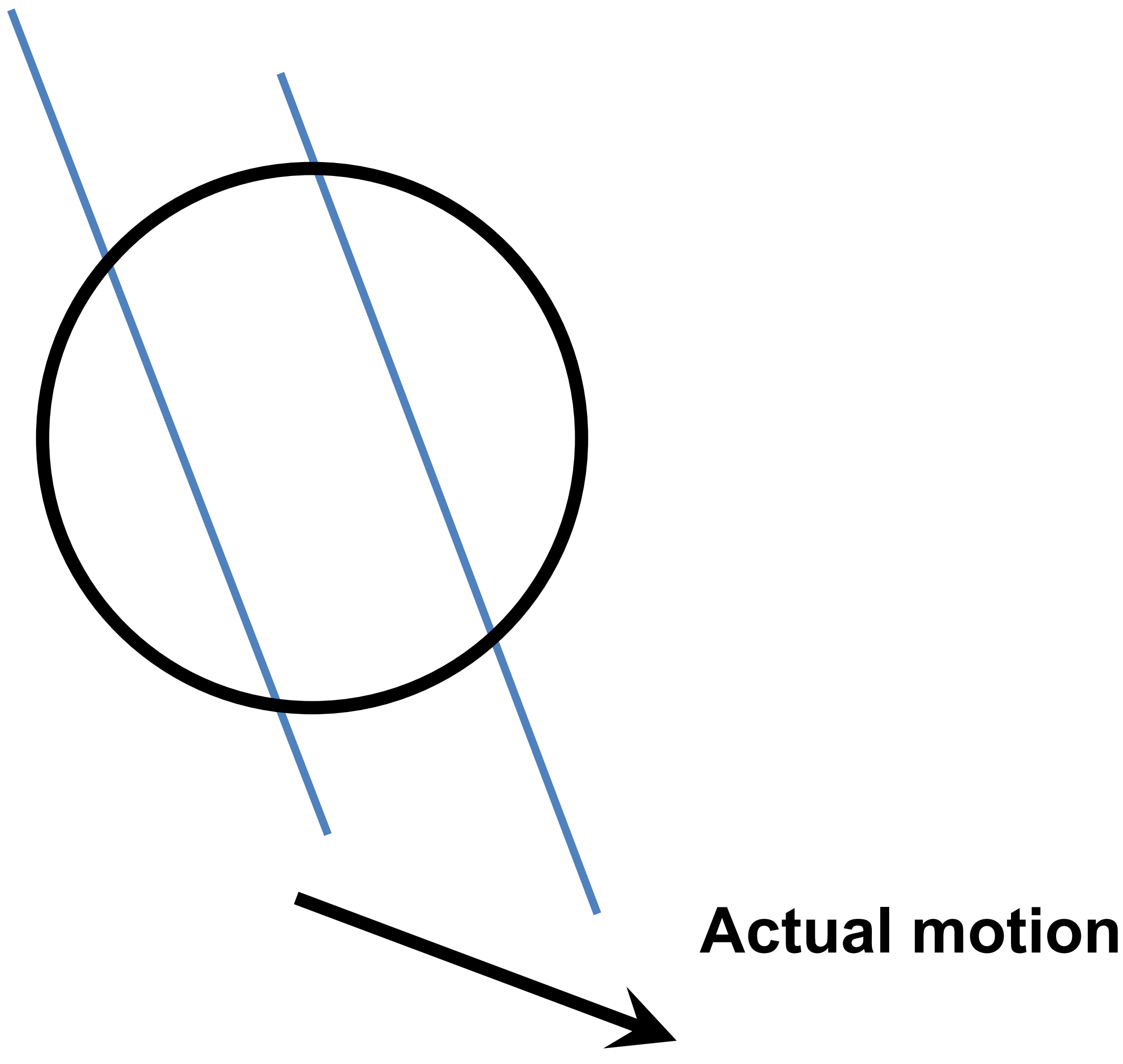


Aperture Problem Exposed

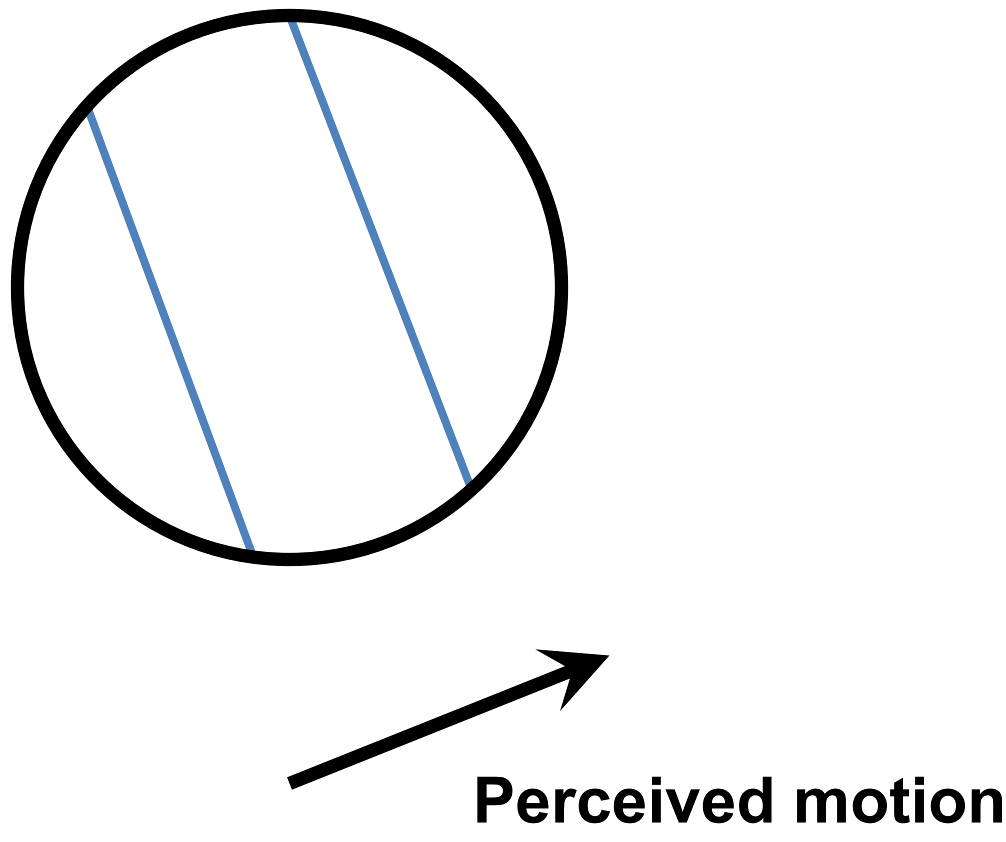


Motion along just an edge is ambiguous

The aperture problem



The aperture problem



Lucas and Kanade

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.

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

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

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

Least squares problem

- Least squares problem:

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

Matching patches across images

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix} \qquad \underset{25\times2}{A} \; \underset{2\times1}{d} = \underset{25\times1}{b}$$

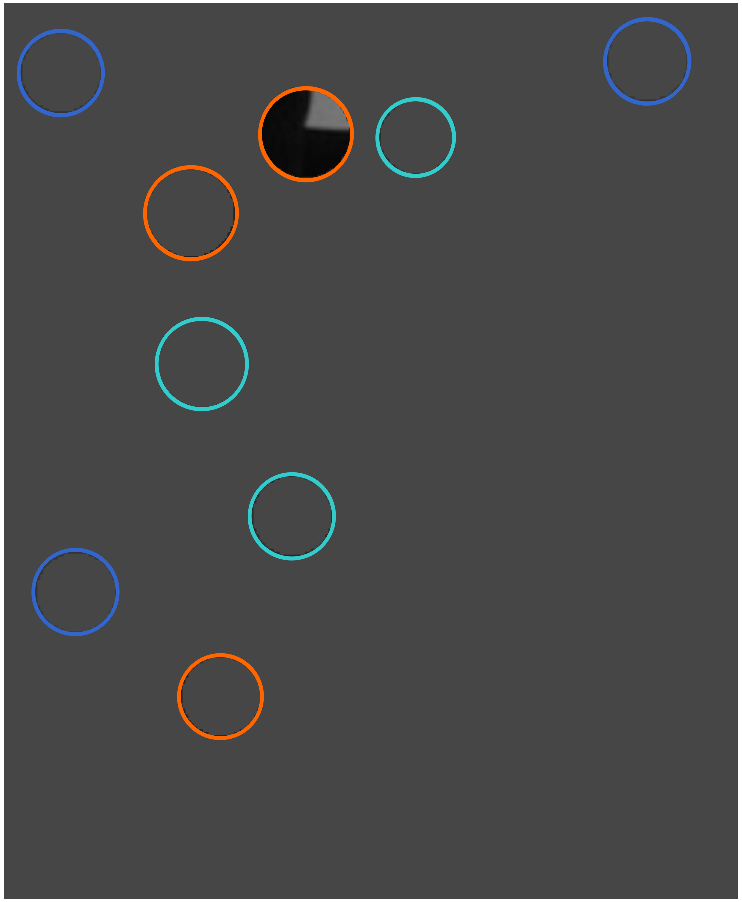
Least squares solution for *d* given by $\quad (A^T A)\ d = A^T b$

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

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

Criteria for Harris corner detector

Aperture problem



Corners

Lines

Flat regions

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

Our assumptions

- 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?

Improving accuracy

- Recall our small motion assumption

$$0 = I(x+u, y+v) - \boldsymbol{I_{t\text{-}1}(x,y)}$$

$$\approx I(x,y) + I_x u + I_y v - \boldsymbol{I_{t\text{-}1}(x,y)}$$

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

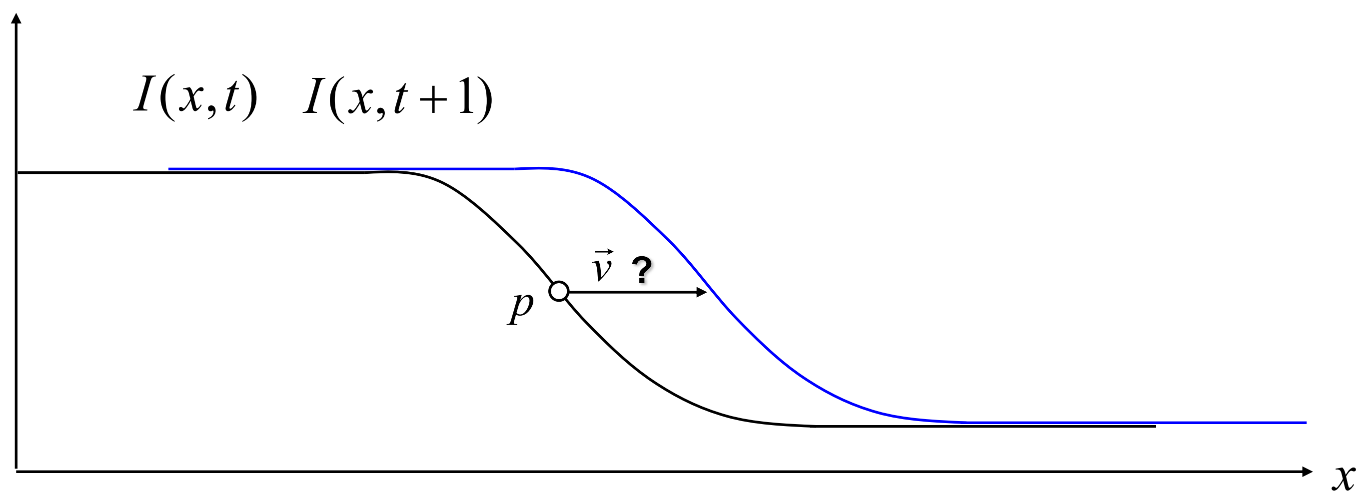
$$= I(x,y) + I_x u + I_y v + \text{higher order terms} - \boldsymbol{I_{t\text{-}1}(x,y)}$$

  - Lukas-Kanade method does one iteration of Newton's method
    - Better results are obtained via more iterations

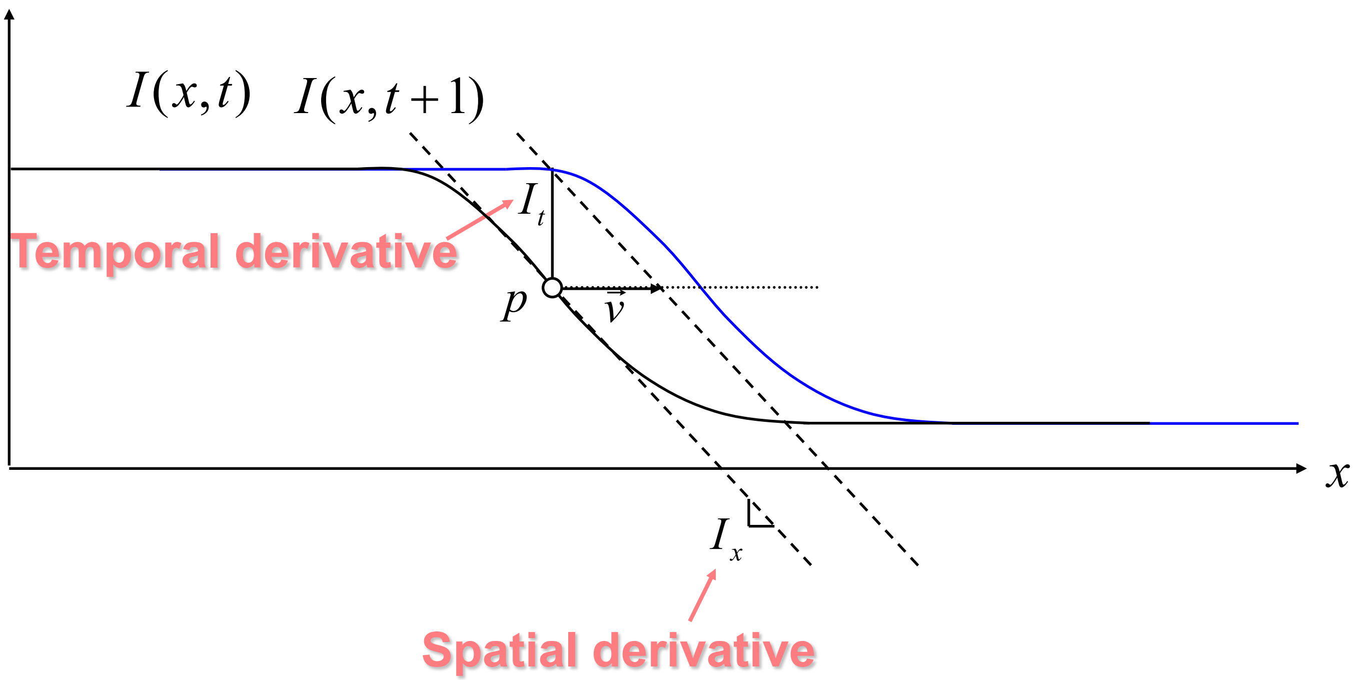
Iterative Refinement

- 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

Tracking in the 1D case:



Tracking in the 1D case:



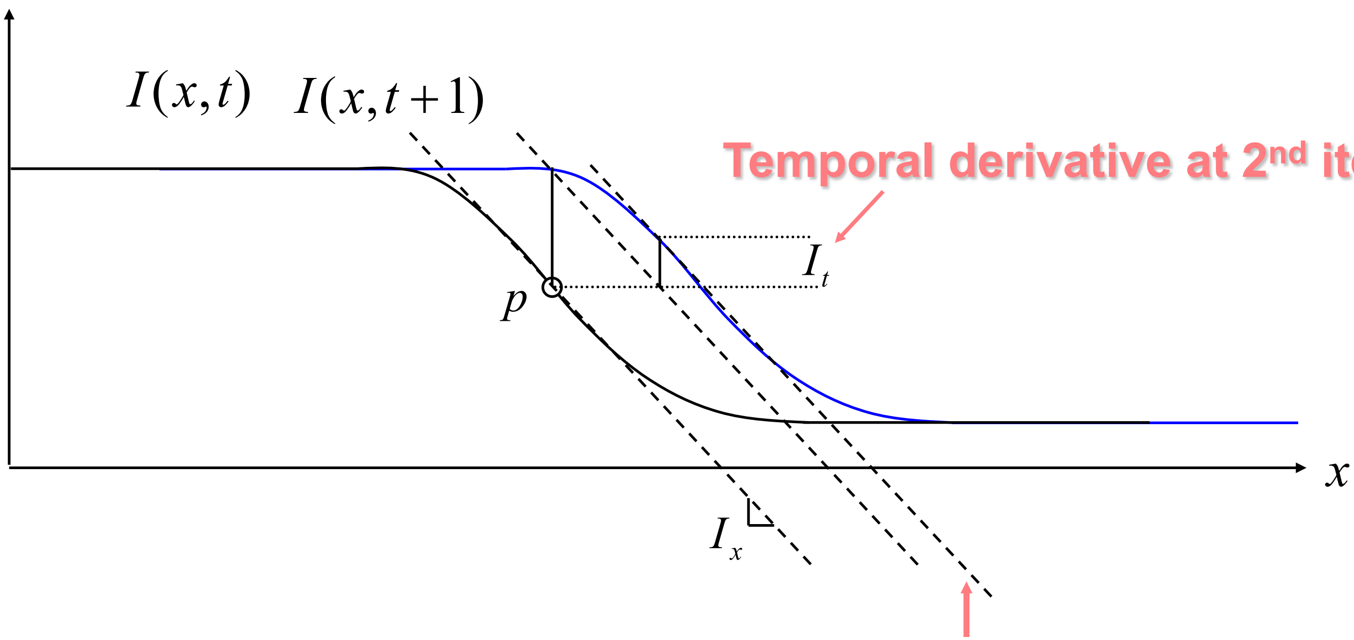
$$I_x = \left.\frac{\partial I}{\partial x}\right|_t \qquad I_t = \left.\frac{\partial I}{\partial t}\right|_{x=p} \qquad \Rightarrow \qquad \vec{v} \approx -\frac{I_t}{I_x}$$

Assumptions:

- **Brightness constancy**
- **Small motion**

Tracking in the 1D case:

Iterating helps refining the velocity vector



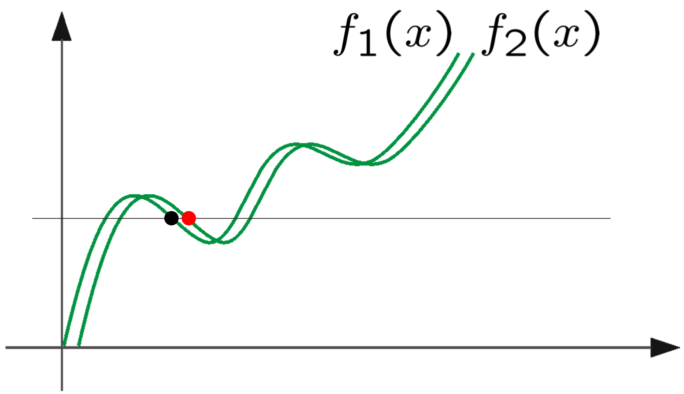
$$\vec{v} \leftarrow \vec{v}_{previous} - \frac{I_t}{I_x}$$

Converges in about 5 iterations

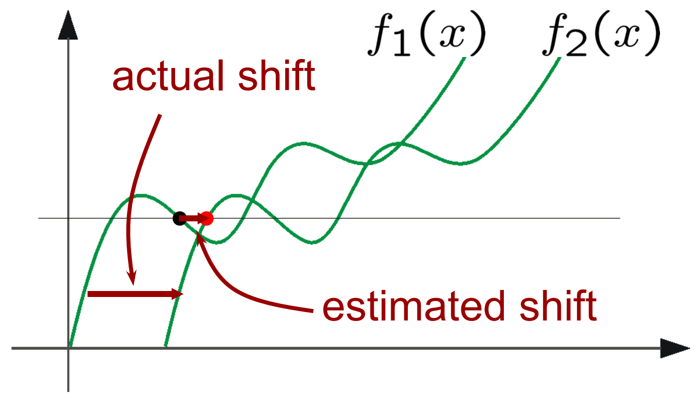
Optical Flow: Aliasing

Temporal aliasing causes ambiguities in optical flow because images can have many pixels with the same intensity.

I.e., how do we know which ‘correspondence’ is correct?



nearest match is correct
(no aliasing)



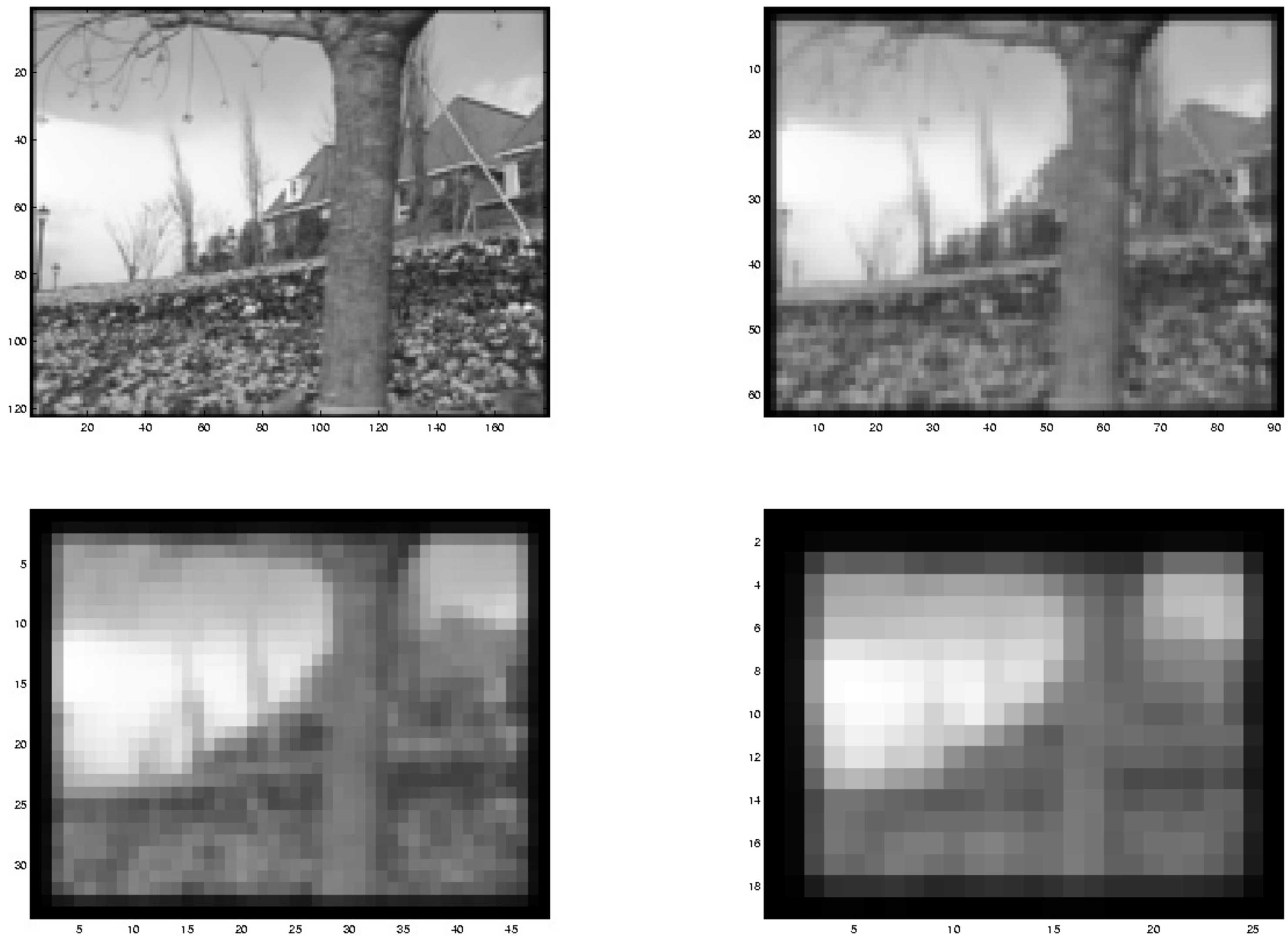
nearest match is incorrect
(aliasing)

To overcome aliasing: coarse-to-fine estimation.

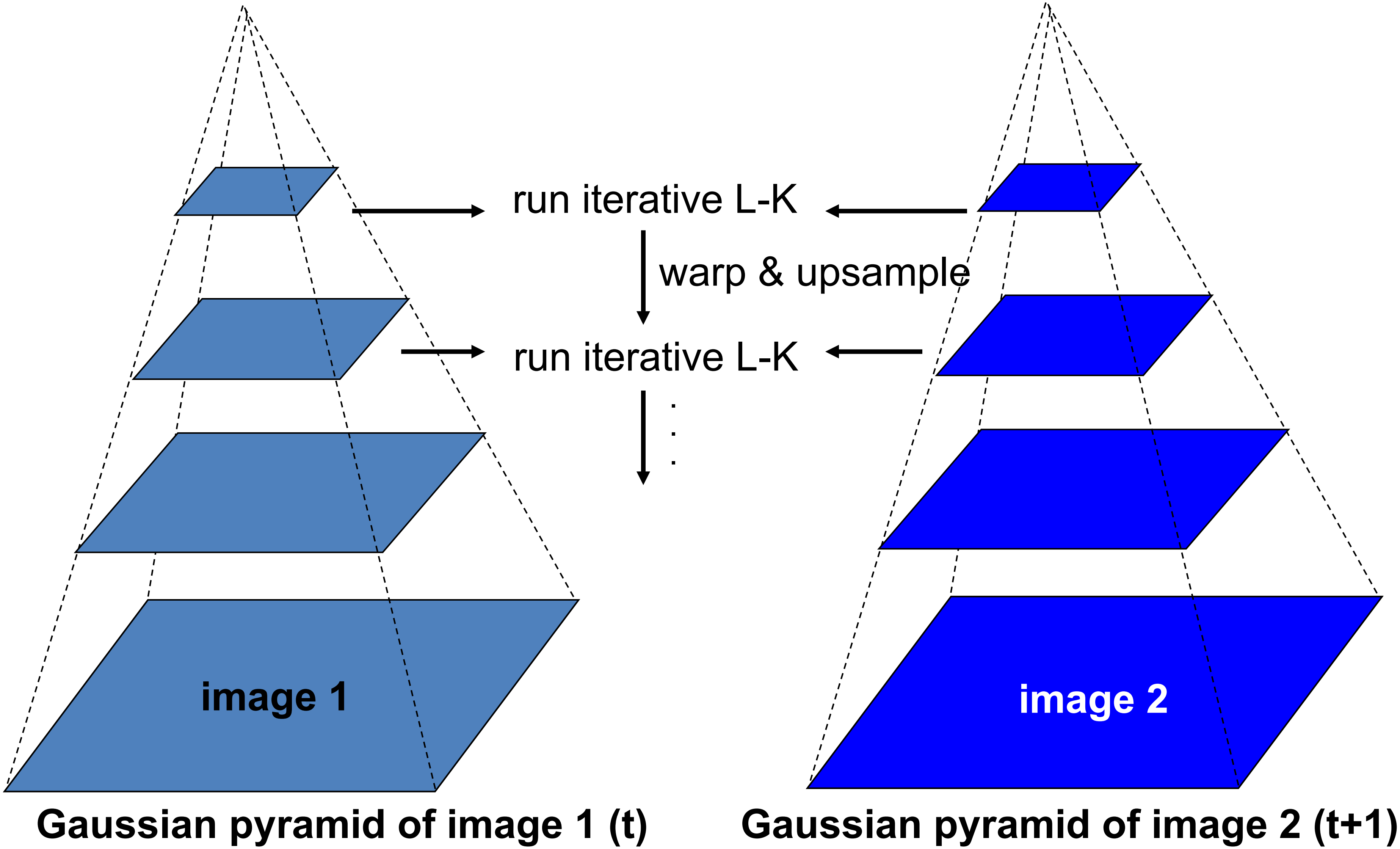
Revisiting the small motion assumption

- Is this motion small enough?
  - Probably not—it's much larger than one pixel
  - How might we solve this problem?

Reduce the resolution!



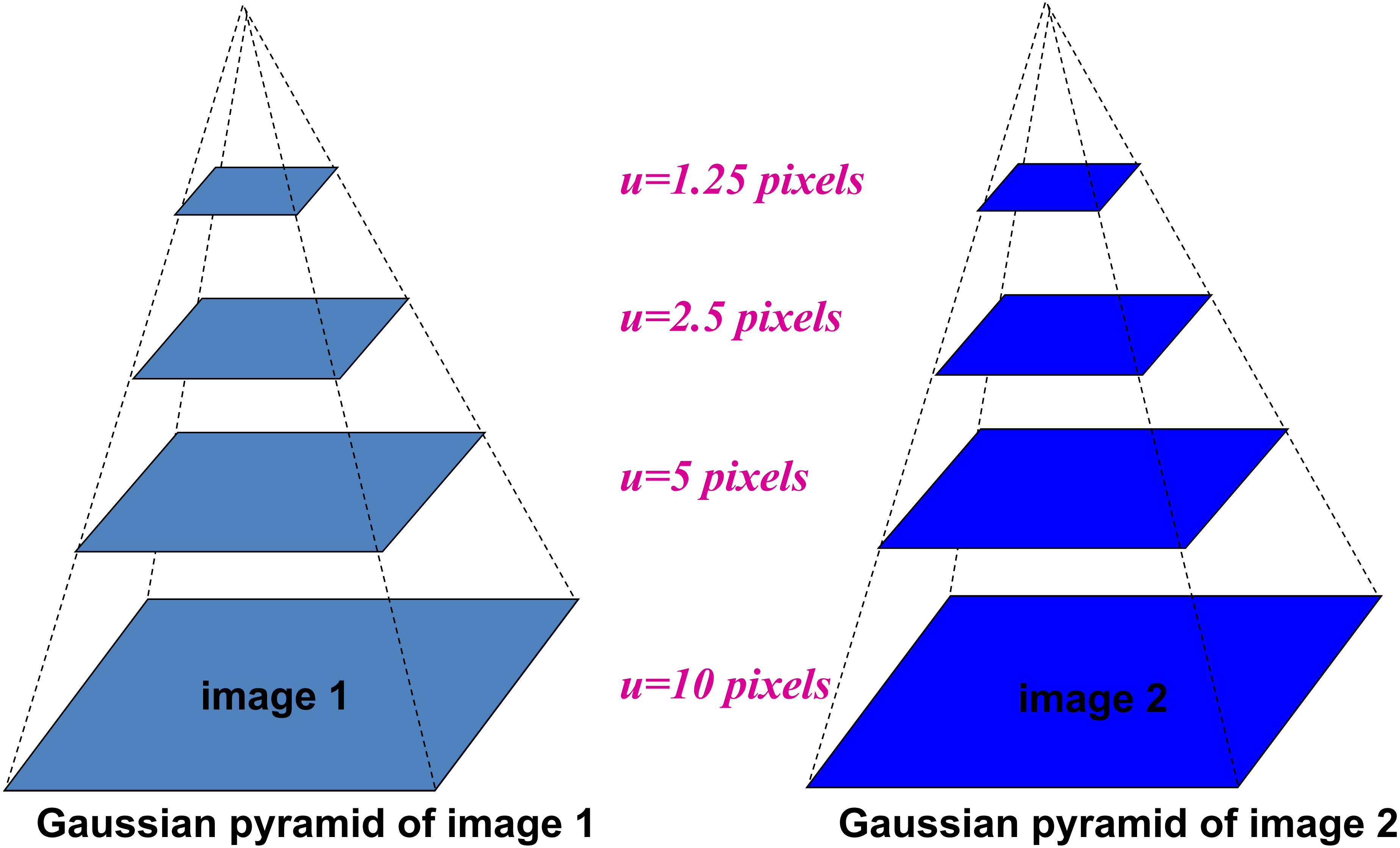
Coarse-to-fine optical flow estimation



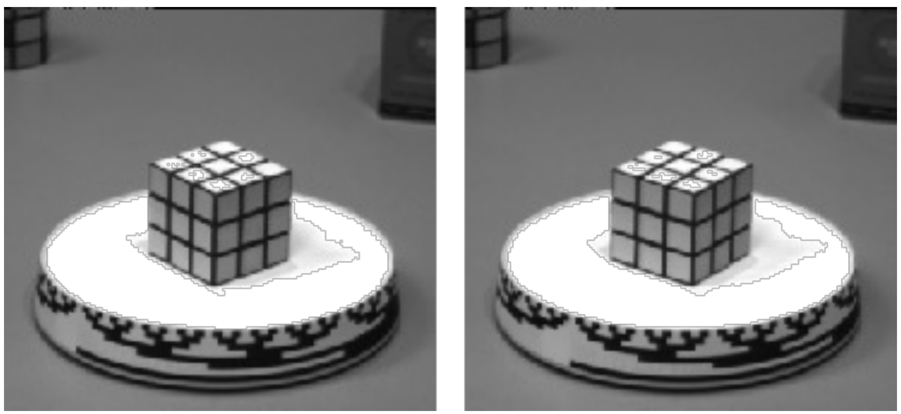
A Few Details

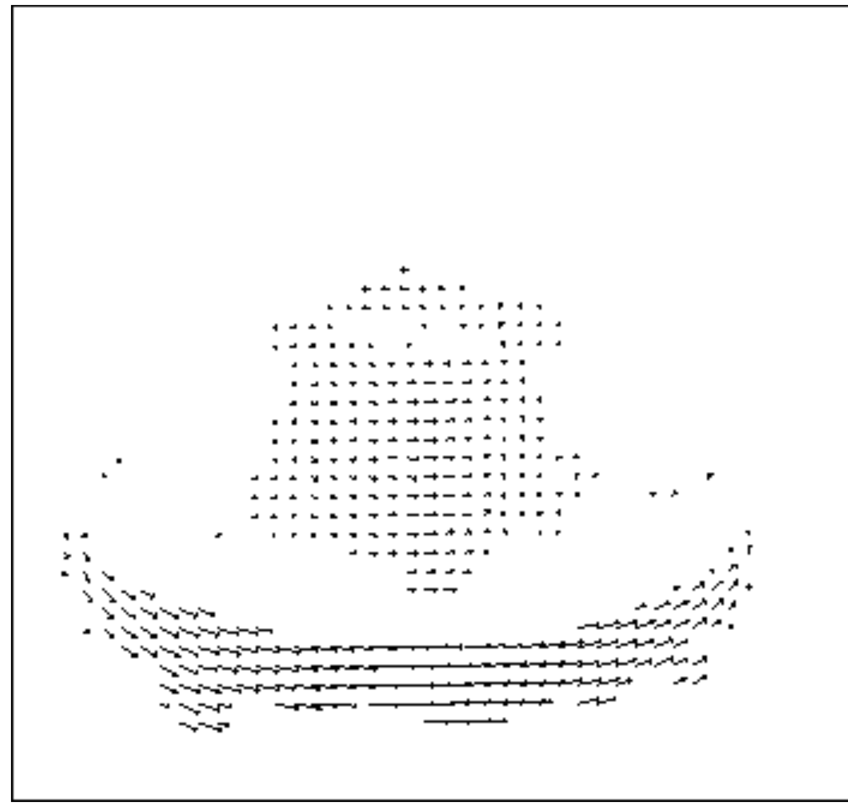
- Top Level
  - Apply L-K to get a flow field representing the flow from the first frame to the second frame.
  - Apply this flow field to warp the first frame toward the second frame.
  - Return L-K on the new warped image to get a flow field from it to the second frame.
  - Repeat till convergence.
- Next Level
  - Upsample the flow field to the next level as the first guess of the flow at that level.
  - Apply this flow field to warp the first frame toward the second frame.
  - Rerun L-K and warping till convergence as above.
- Etc.

Coarse-to-fine optical flow estimation



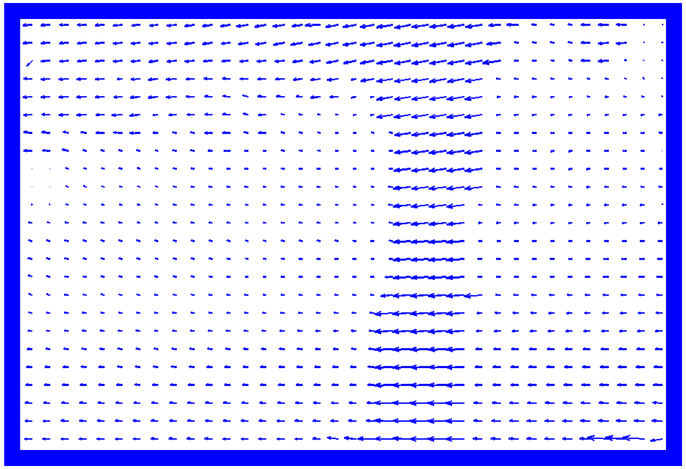
Optical flow result



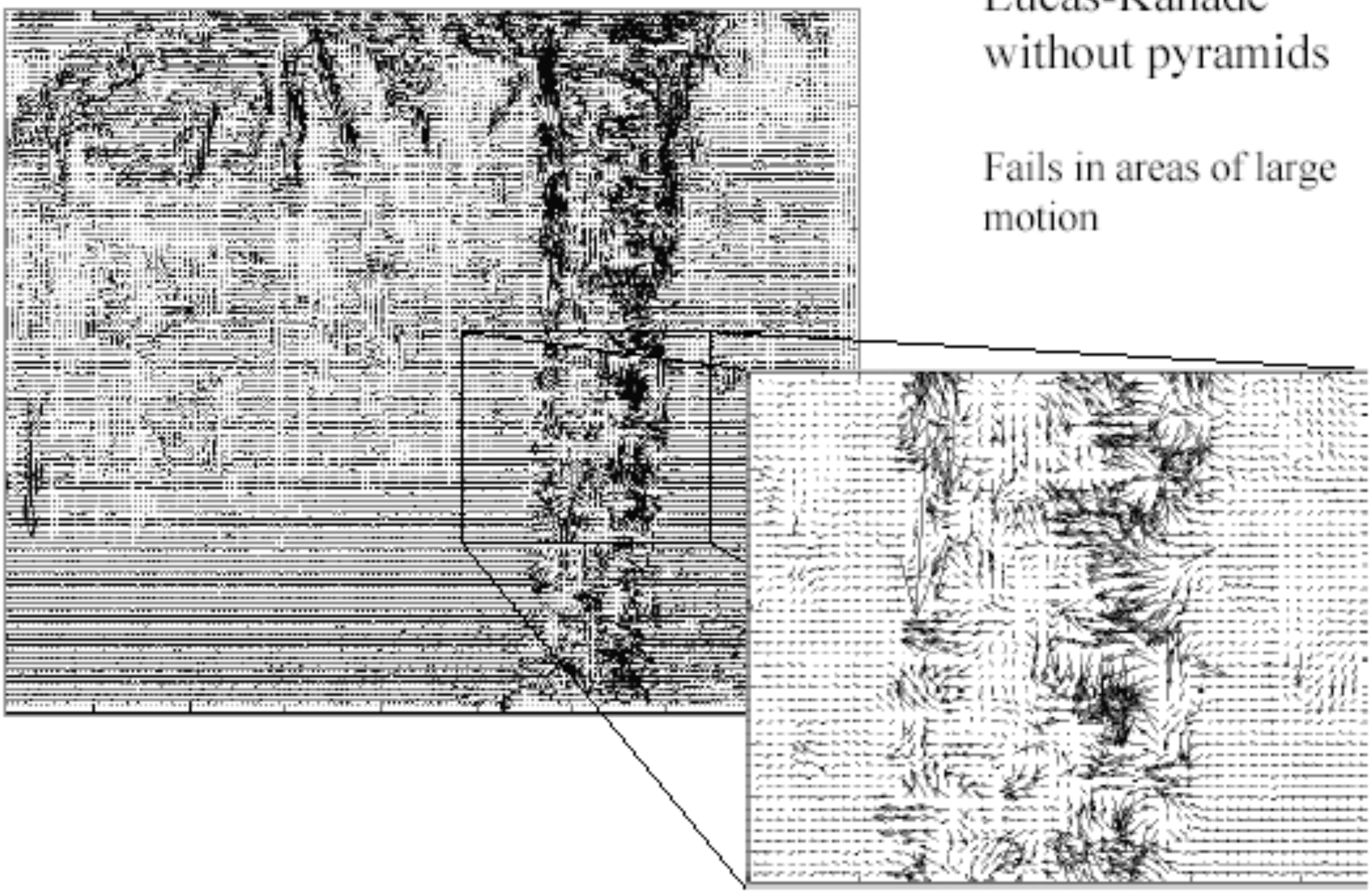


The Flower Garden Video

What should the optical flow be?



Optical Flow Results

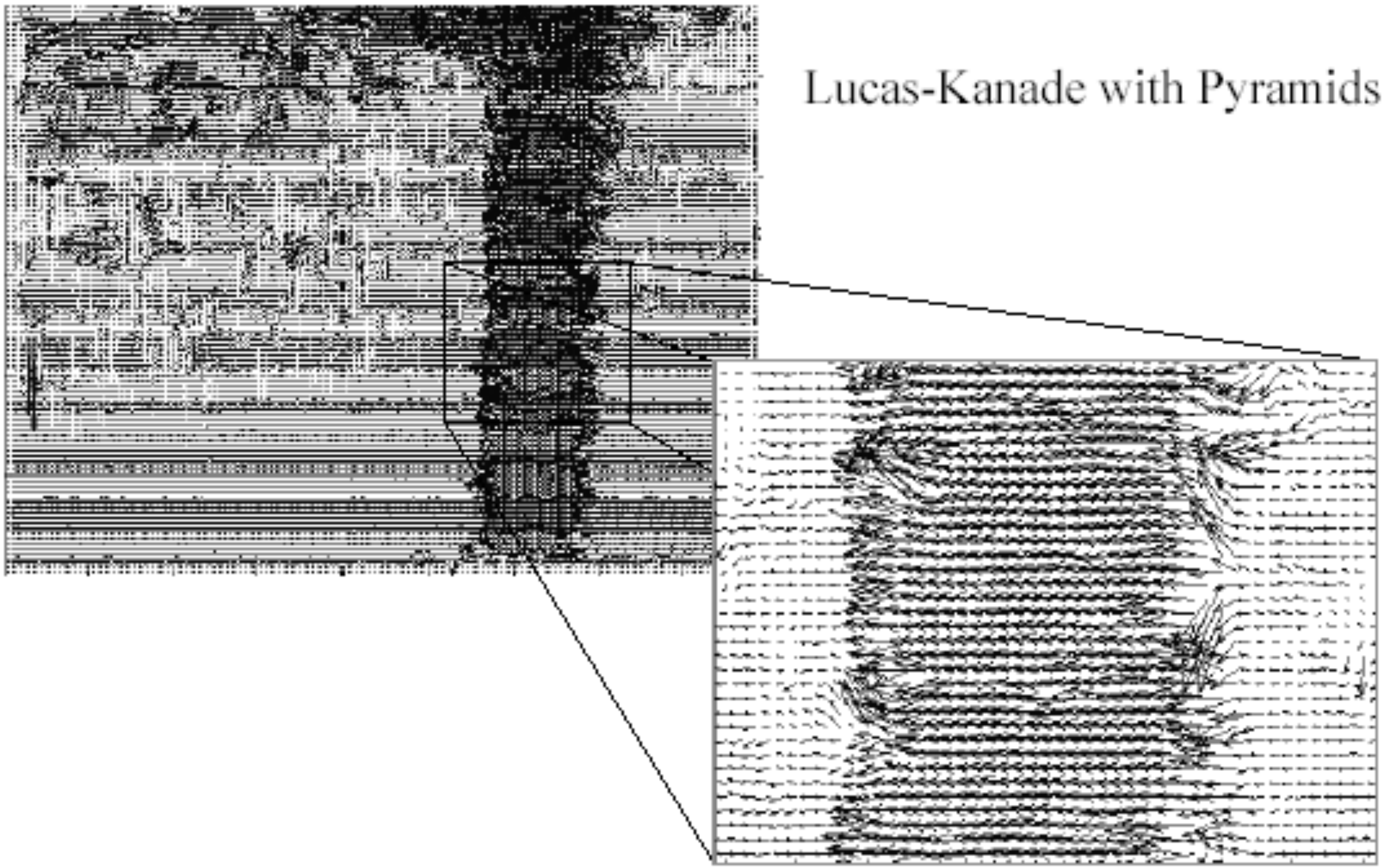


Lucas-Kanade
without pyramids

Fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Optical Flow Results



Lucas-Kanade with Pyramids

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Summary

- Major contributions from Lucas, Tomasi, Kanade
  - Tracking feature points
  - Optical flow

- Key ideas
  - By assuming brightness constancy, truncated Taylor expansion leads to simple and fast patch matching across frames
  - Coarse-to-fine registration