Computer Vision

CS-E4850, 5 study credits

Lecturer: Juho Kannala

Lecture 7: Optical flow and keypoint tracking

- Given two subsequent frames of a video, the optical flow field indicates the apparent motion of each pixel

- If we have more than two frames, we can track features from one frame to the next by following the optical flow

Acknowledgement: many slides from Svetlana Lazebnik, Derek Hoiem, Steve Seitz, Rick Szeliski, M. Pollefeys, and others (detailed credits on individual slides)

Reading & software

- Szeliski's book, 1st edition: Chapter 8 or 2nd edition: Chapter 9

- Baker & Matthews: Lucas-Kanade 20 years on, a unifying framework, 2004
  - https://www.ri.cmu.edu/publications/lucas-kanade-20-years-on-a-unifying-framework/

- Shi & Tomasi: Good features to track, 1994
  - http://www.ai.mit.edu/courses/6.891/handouts/shi94good.pdf

- OpenCV software:
  - http://docs.opencv.org/3.1.0/d7/d8b/tutorial_py_lucas_kanade.html

Motivation: glimpse to the state of the art

FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks

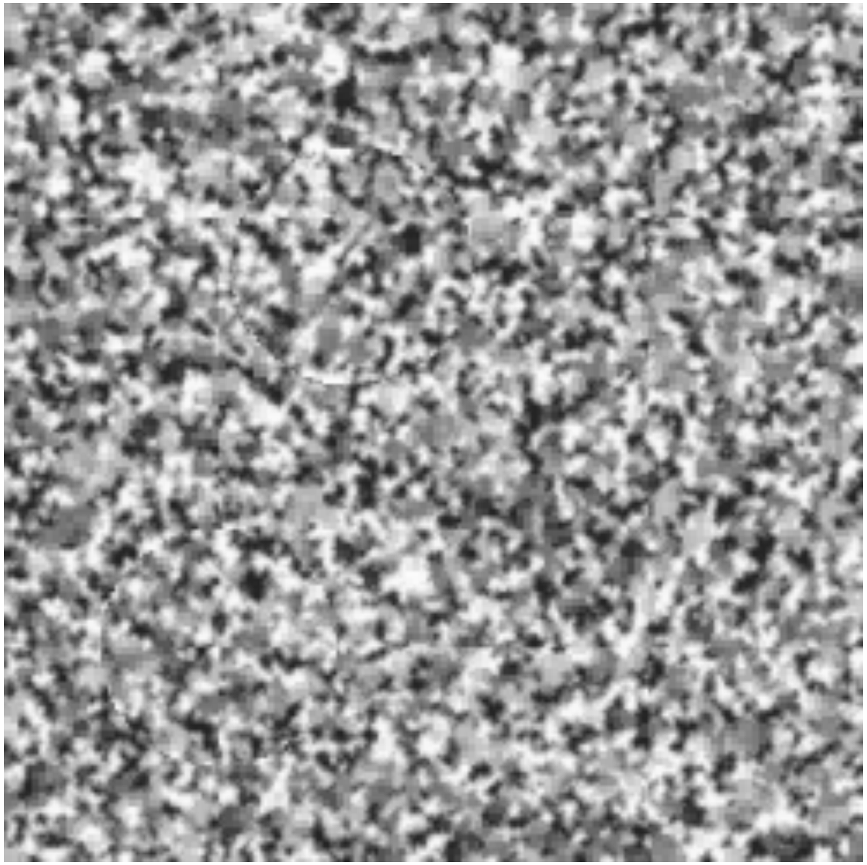
Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, Thomas Brox

University of Freiburg, Germany

—— Supplementary Material ——

Motion is a powerful perceptual cue

- Sometimes, it is the only cue



Motion is a powerful perceptual cue

- 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 is a powerful perceptual cue

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

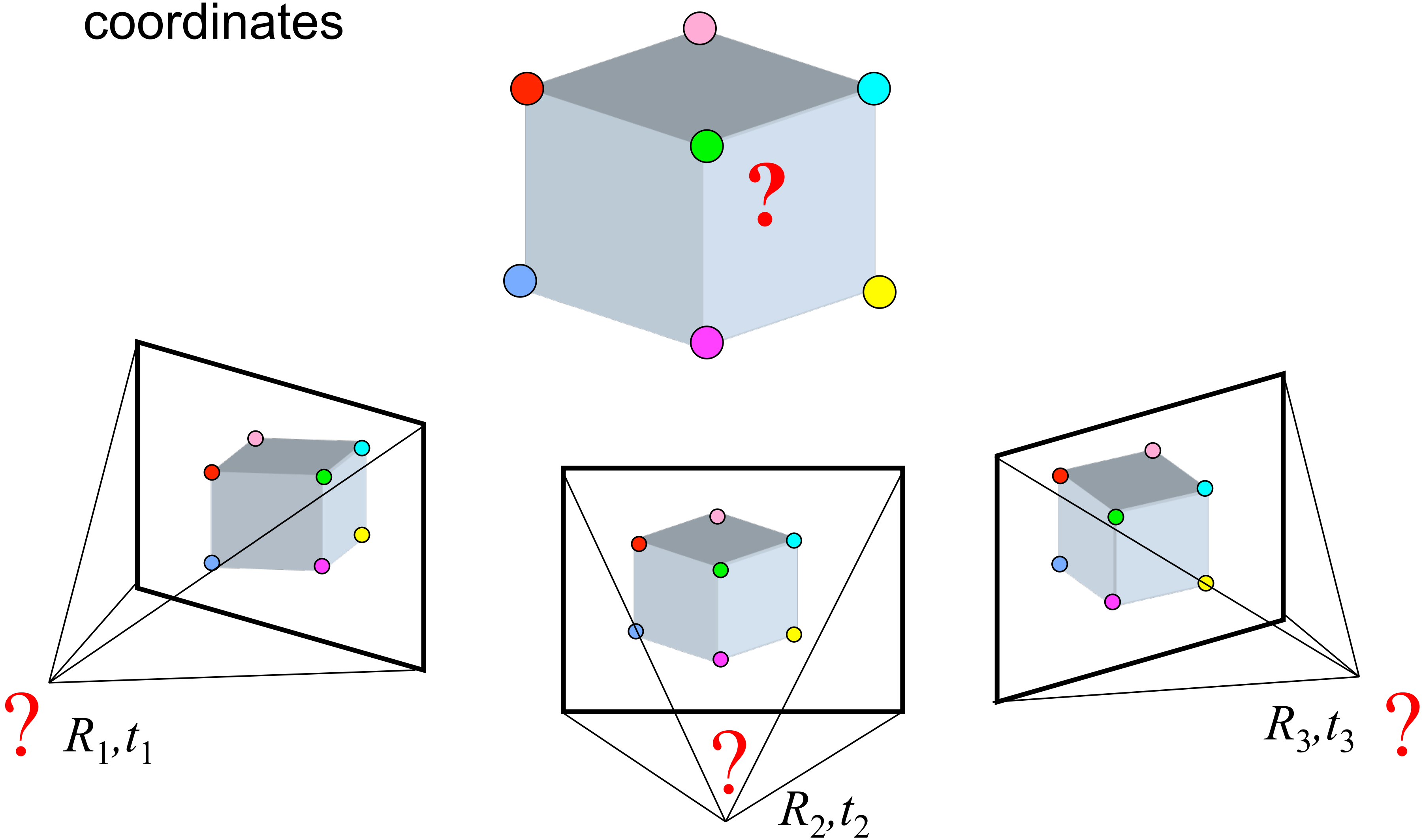
G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis",
Perception and Psychophysics 14, 201-211, 1973.

Uses of motion in computer vision

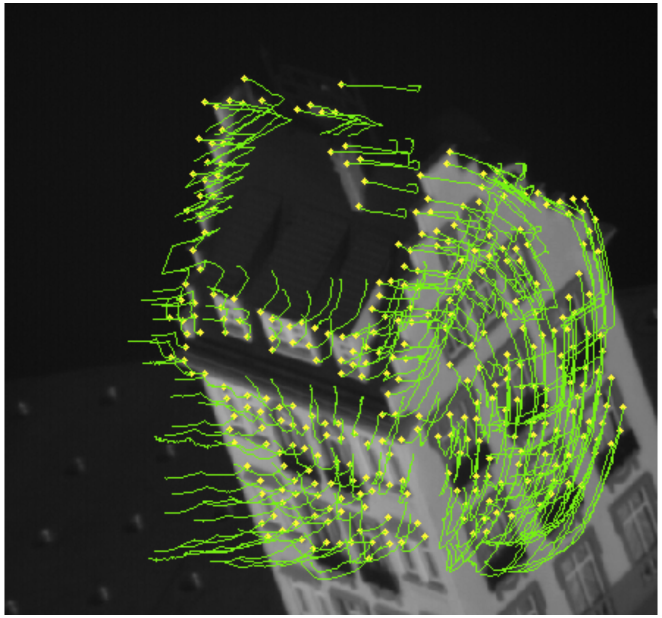
- 3D shape reconstruction
- Object segmentation
- Learning and tracking of dynamical models
- Event and activity recognition

Preview: Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



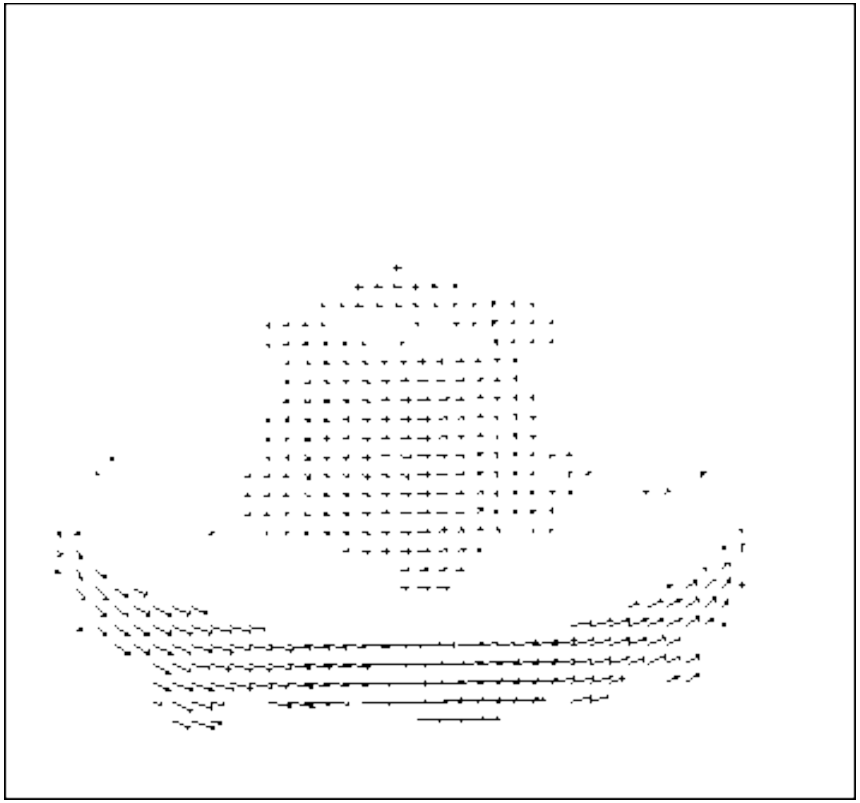
Keypoint tracking



C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Motion field

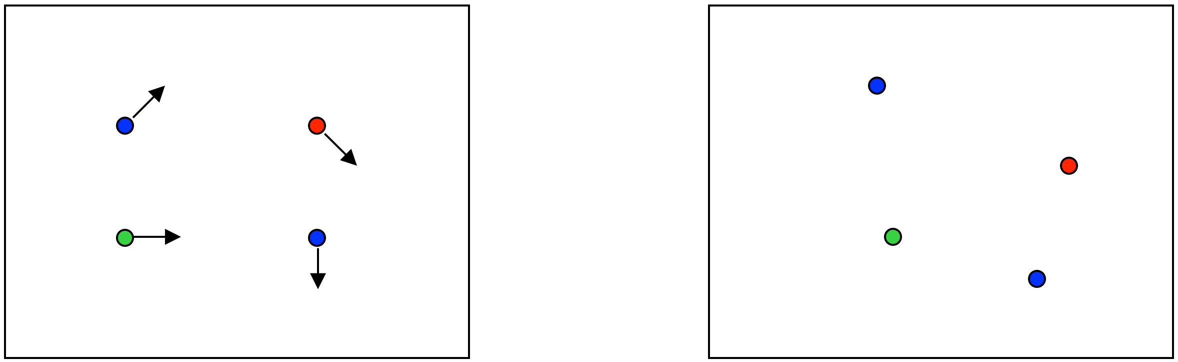
- The motion field is the projection of the 3D scene motion into the image



Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image
- Ideally, optical flow would be the same as the motion field
- Have to be careful: apparent motion can be caused by lighting changes without any actual motion
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

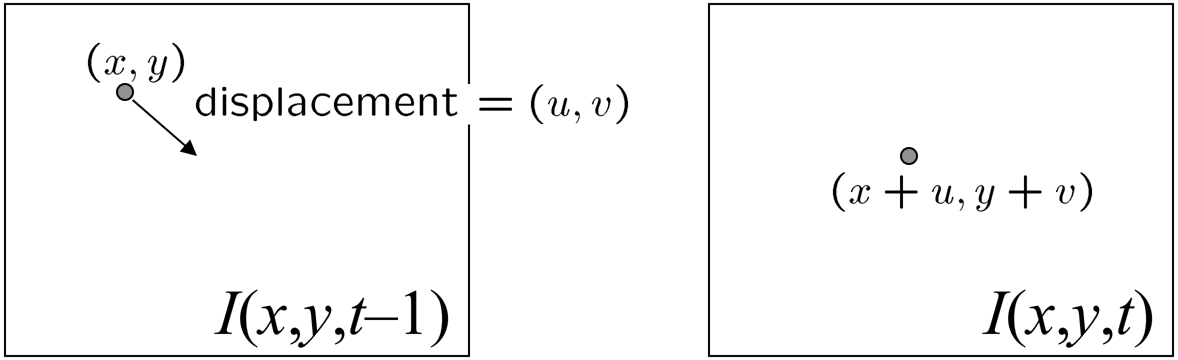
Estimating optical flow



$I(x,y,t-1)$ $\quad$ $I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field $u(x,y)$ and $v(x,y)$ between them
- Key assumptions
  - **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:

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

Linearizing the right side using Taylor expansion:

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

Hence, $$I_x\, u + I_y\, v + I_t \approx 0$$

The brightness constancy constraint

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

- How many equations and unknowns per pixel?
  - One equation, two unknowns

- What does this constraint mean?

$$\nabla I \cdot (u, v) + I_t = 0$$

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

The brightness constancy constraint

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

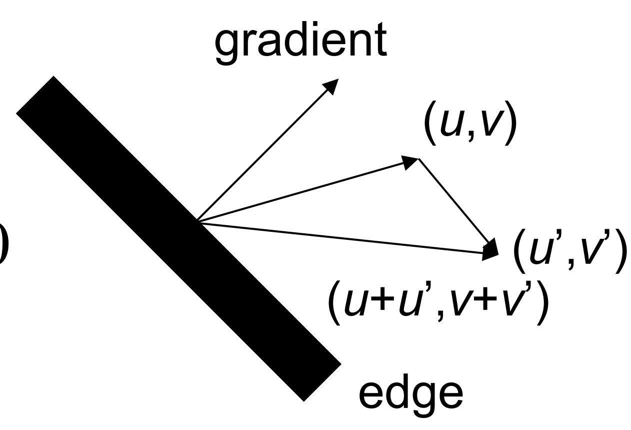
- How many equations and unknowns per pixel?
  - One equation, two unknowns

- What does this constraint mean?

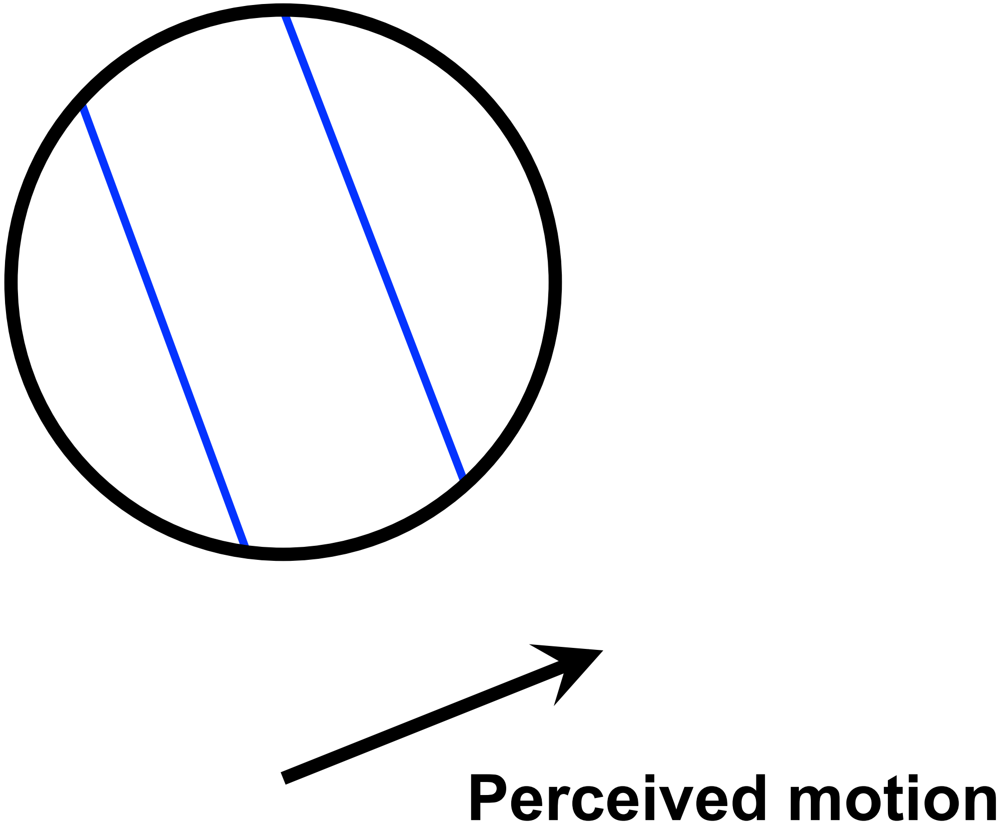
$$\nabla I \cdot (u, v) + I_t = 0$$

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

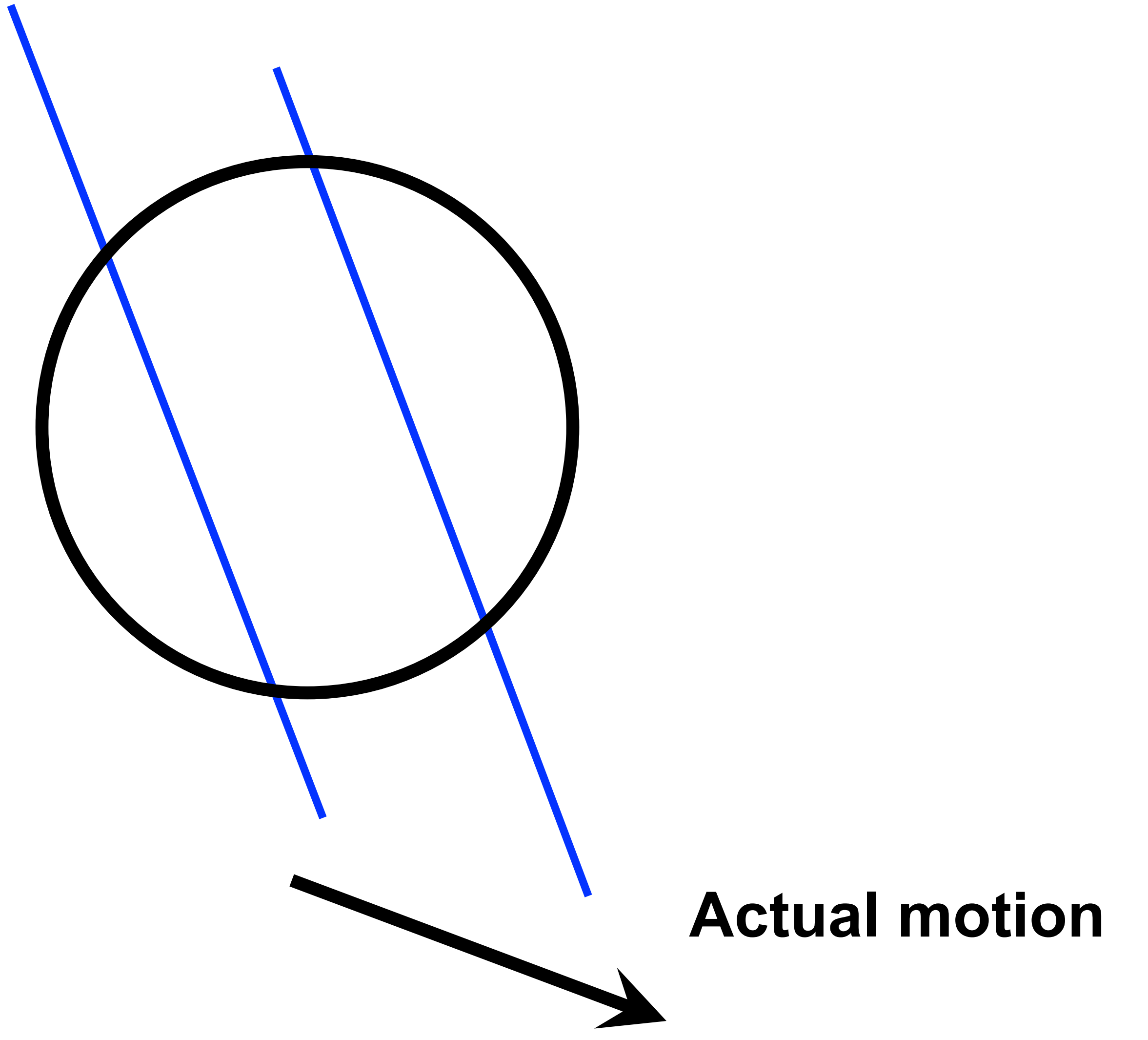
If (u, v) satisfies the equation,
so does ($u+u'$, $v+v'$) if $\nabla I \cdot (u', v') = 0$



The aperture problem



The aperture problem



The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion

The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion

Solving the aperture problem

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

$$\nabla I(\mathbf{x}_i) \cdot [u, v] + I_t(\mathbf{x}_i) = 0$$

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

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.

Solving the aperture problem

- Least squares problem:

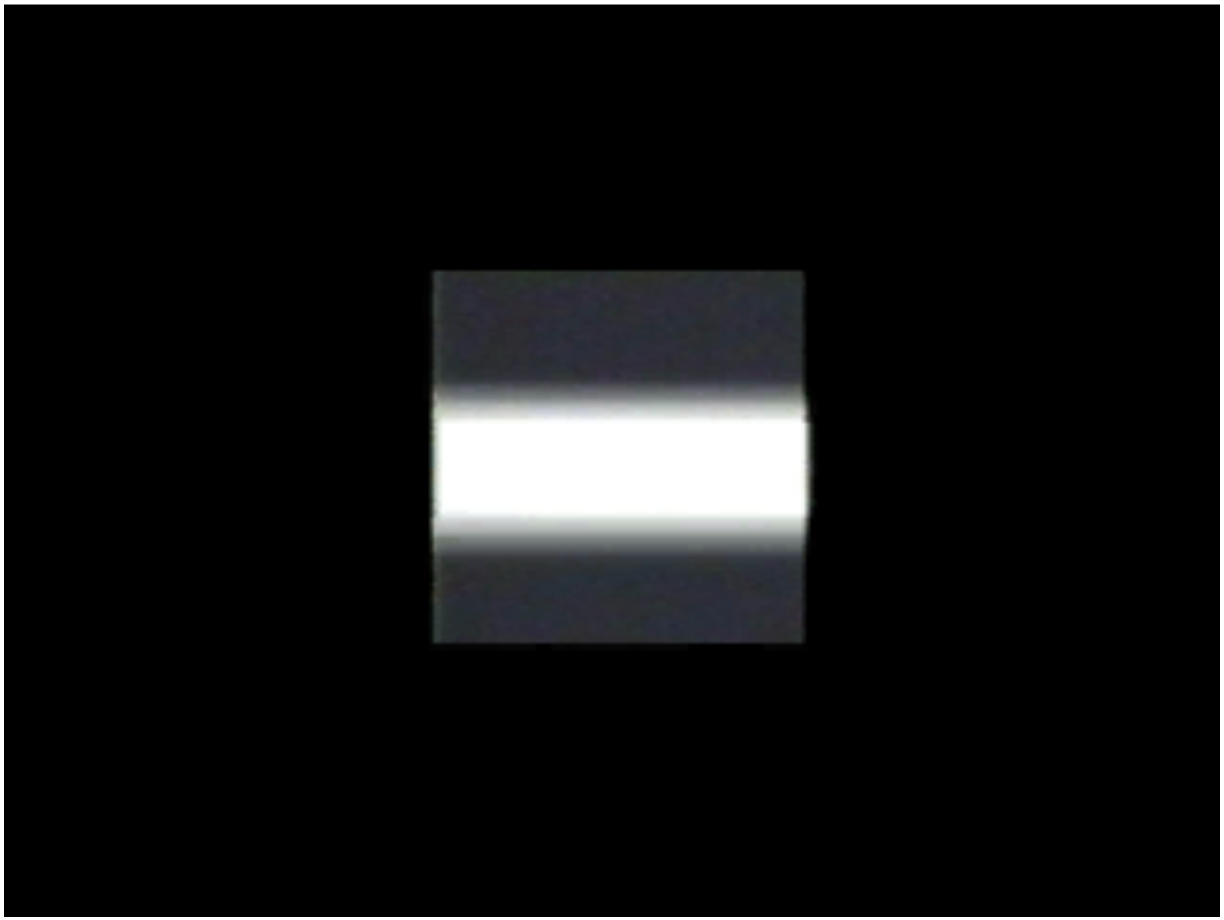
$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

- When is this system solvable?
  - What if the window contains just a single straight edge?

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.

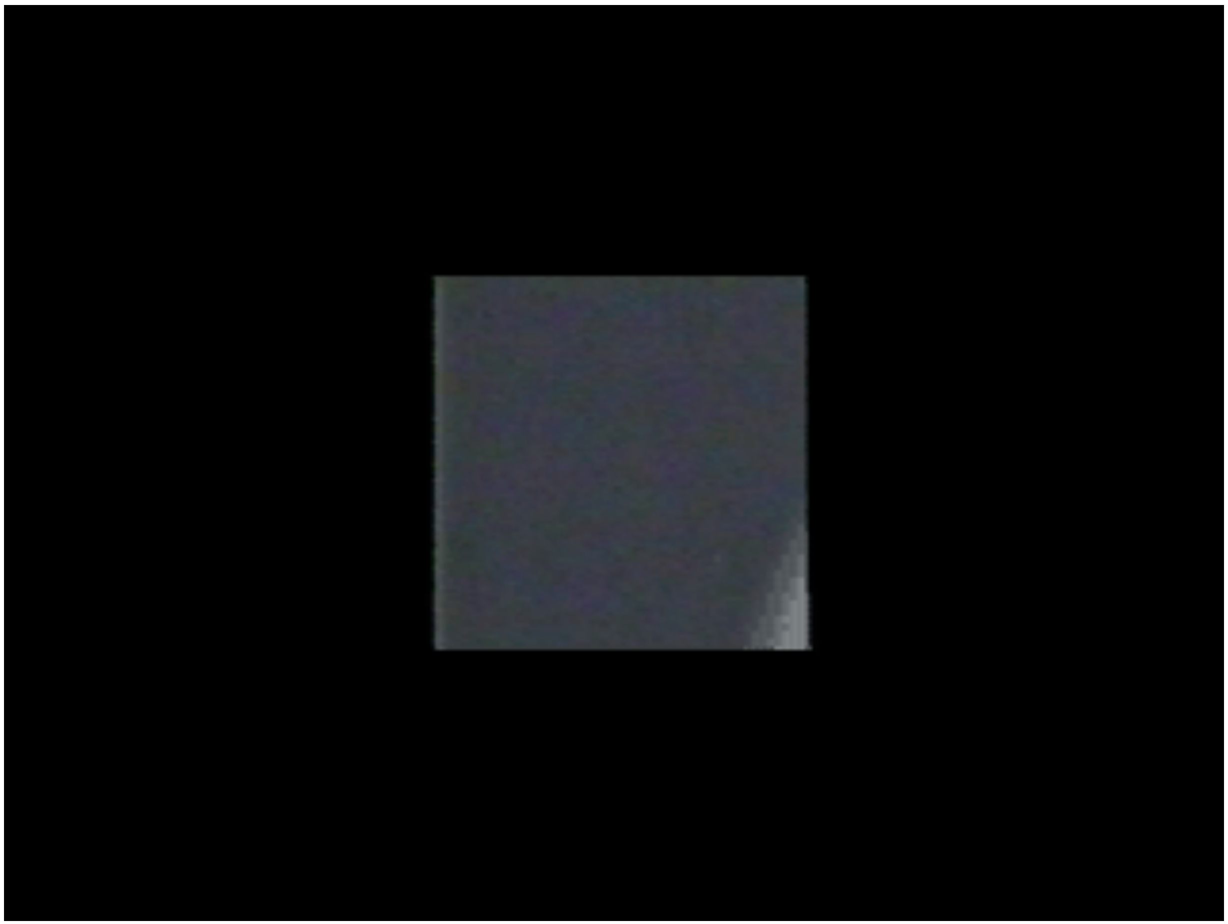
Conditions for solvability

- “Bad” case: single straight edge



Conditions for solvability

- “Good” case



Lucas-Kanade flow

Linear least squares problem

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

$$\underset{n\times 2}{\mathbf{A}} \ \underset{2\times 1}{\mathbf{d}} = \underset{n\times 1}{\mathbf{b}}$$

Solution given by $(\mathbf{A}^T\mathbf{A})\mathbf{d} = \mathbf{A}^T\mathbf{b}$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

The summations are over all pixels in the window

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.

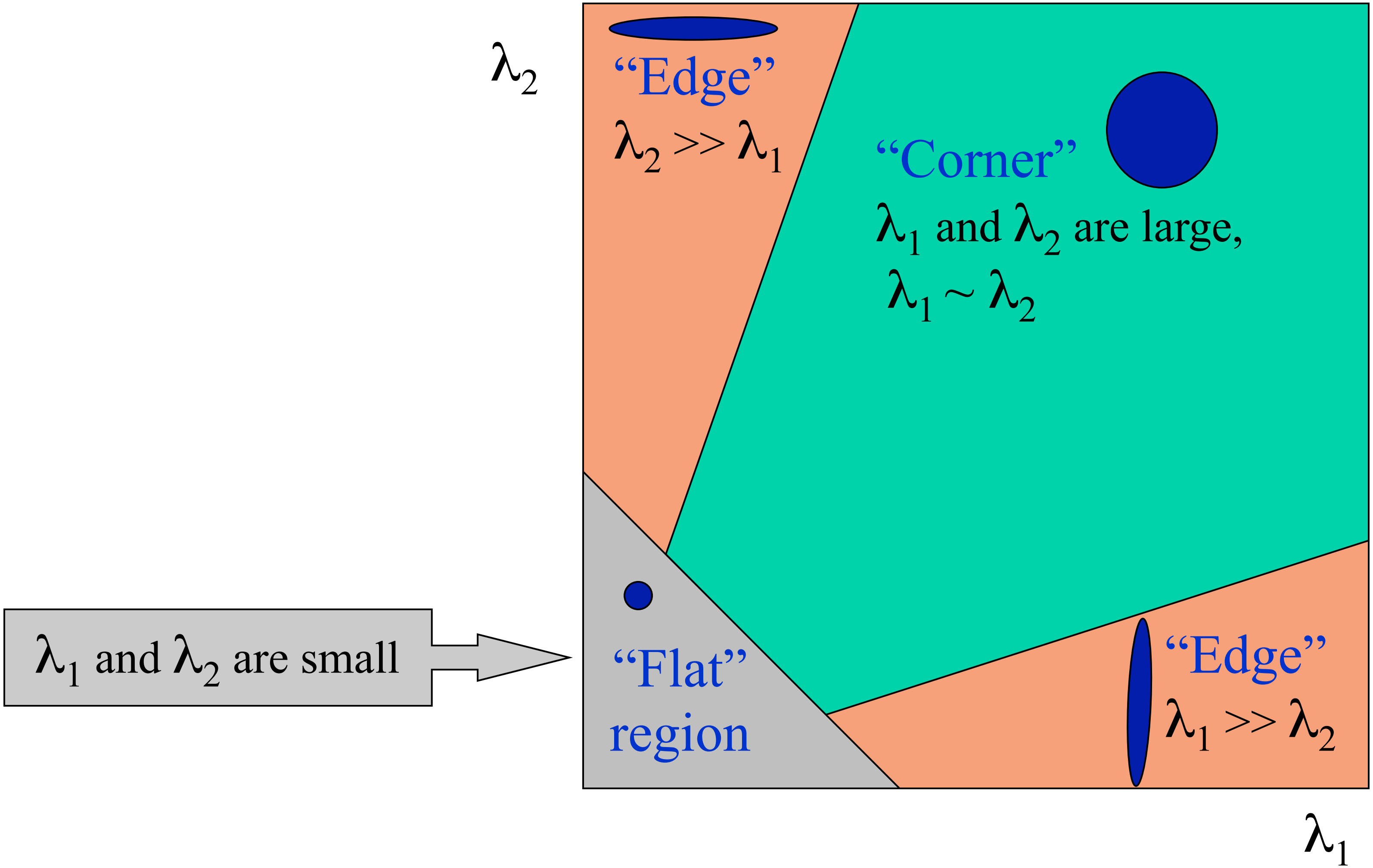
Lucas-Kanade flow

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

- Recall the Harris corner detector: $\mathbf{M} = \mathbf{A}^T\mathbf{A}$ is the *second moment matrix*
- We can figure out whether the system is solvable by looking at the eigenvalues of the second moment matrix
  - The eigenvectors and eigenvalues of **M** relate to edge direction and magnitude
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change, and the other eigenvector is orthogonal to it

Recall: second moment matrix

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



Uniform region

– gradients have small magnitude
– small λ_1, small λ_2
– system is ill-conditioned

Edge

– gradients have one dominant direction
– large λ_1, small λ_2
– system is ill-conditioned

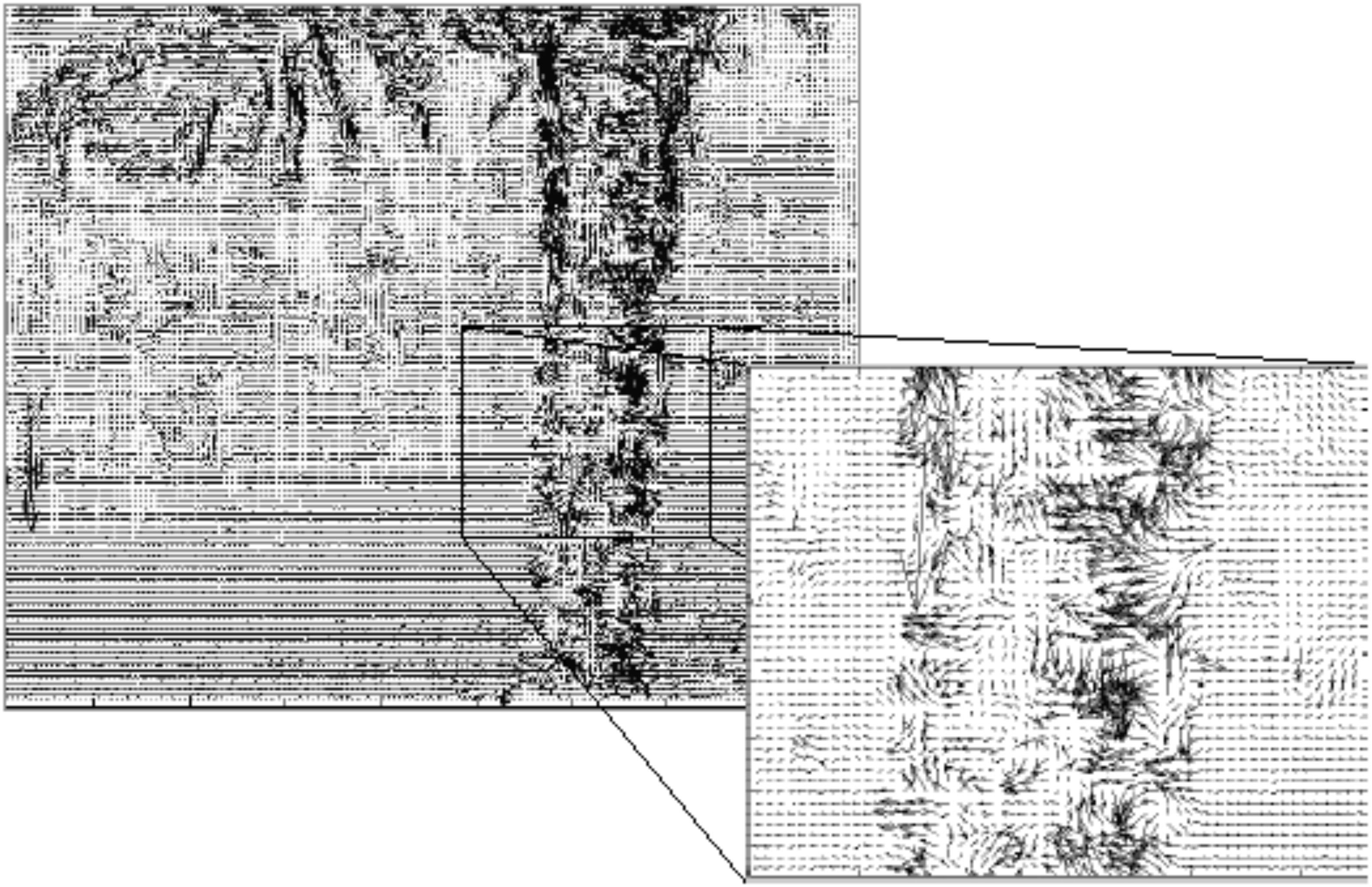
High-texture or corner region

– gradients have different directions, large magnitudes
– large λ_1, large λ_2
– system is well-conditioned

Example of optical flow estimation

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Example of optical flow estimation



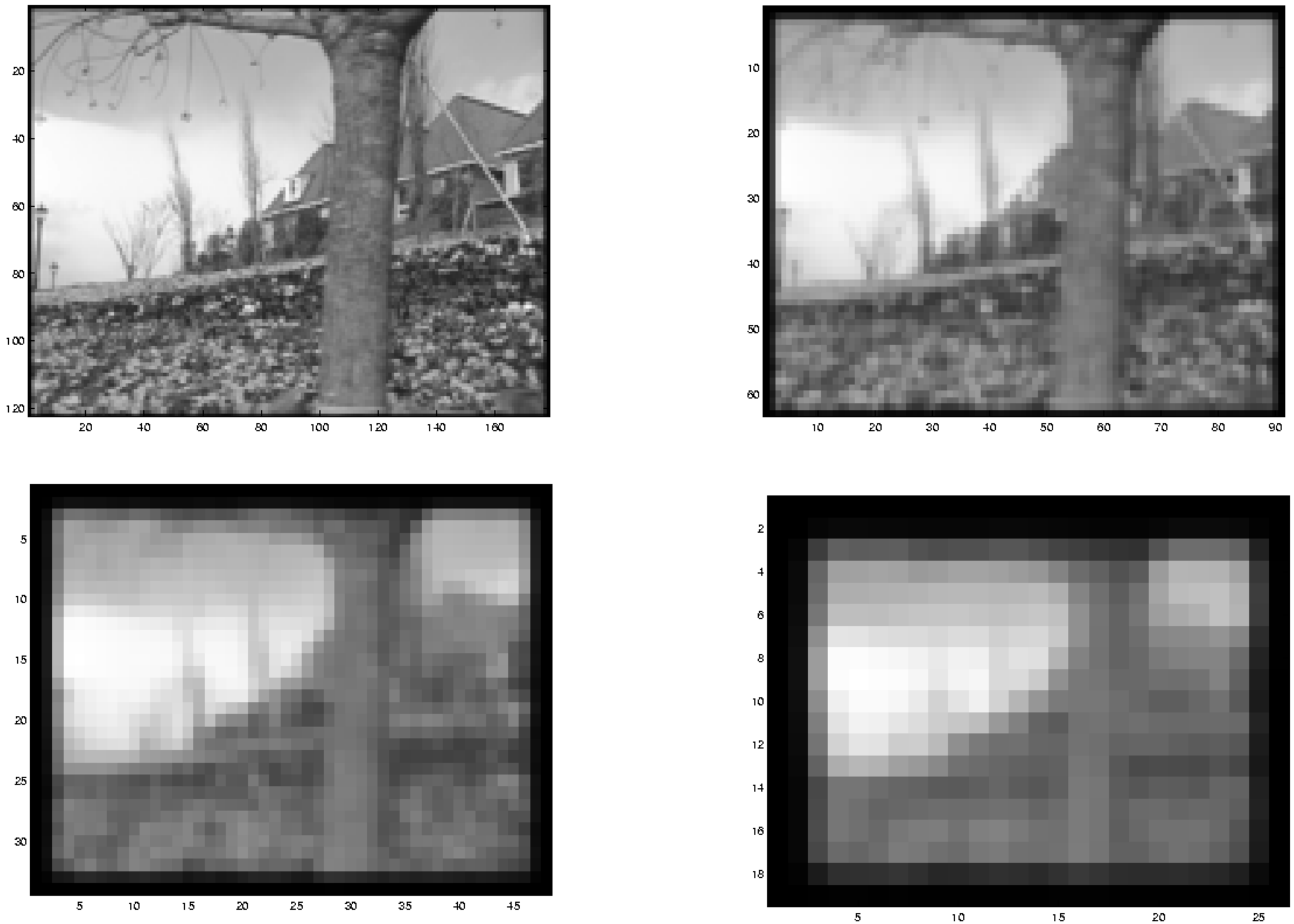
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Iterative Refinement

- Iterative Lukas-Kanade Algorithm
  1. Estimate displacement at each pixel by solving Lucas-Kanade equations
  2. Warp I(t) towards I(t+1) using the estimated flow field
     - Basically, just interpolation
  3. Repeat until convergence

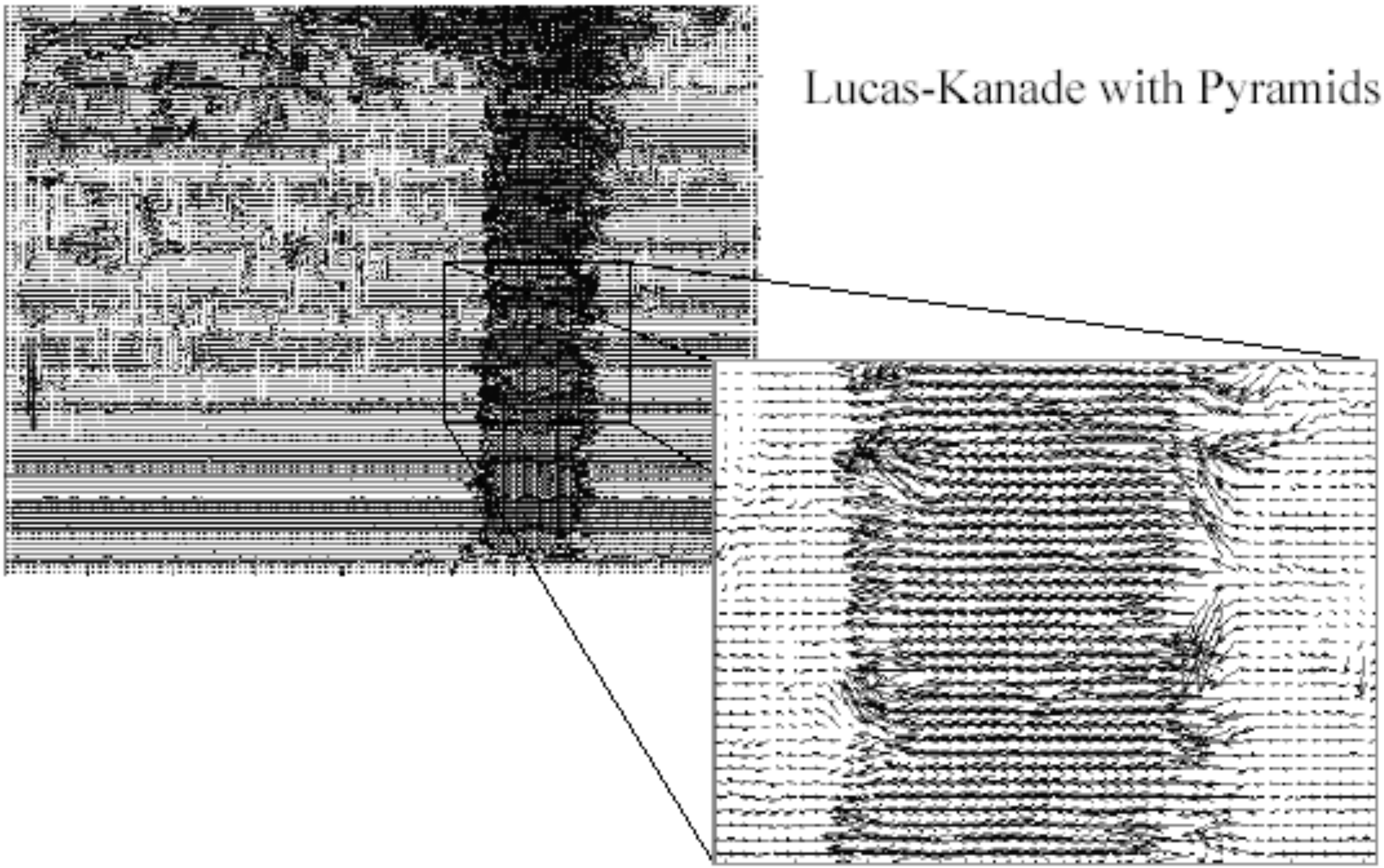
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Multi-resolution estimation



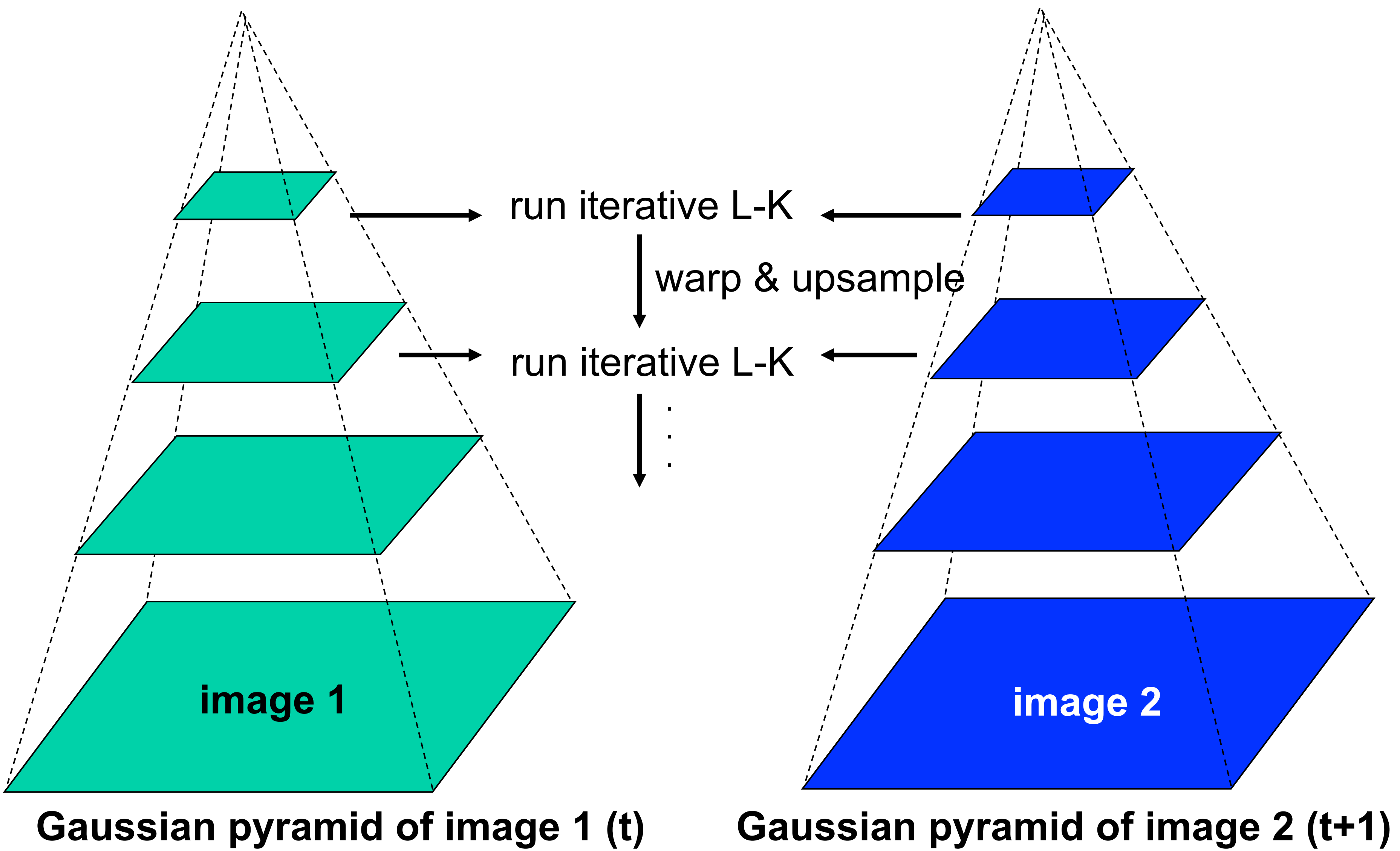
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Multi-resolution estimation



* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

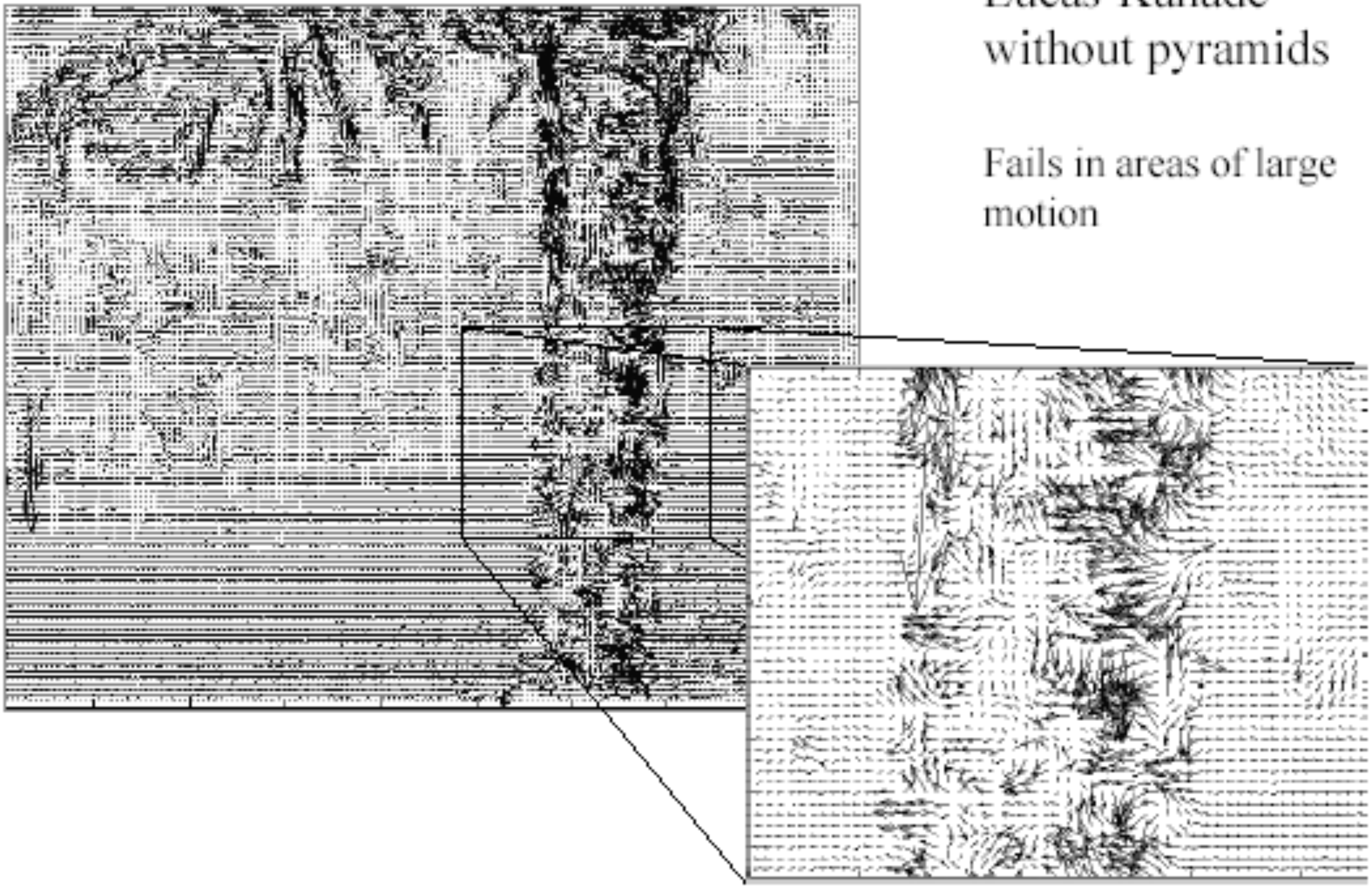
Coarse-to-fine optical flow estimation



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

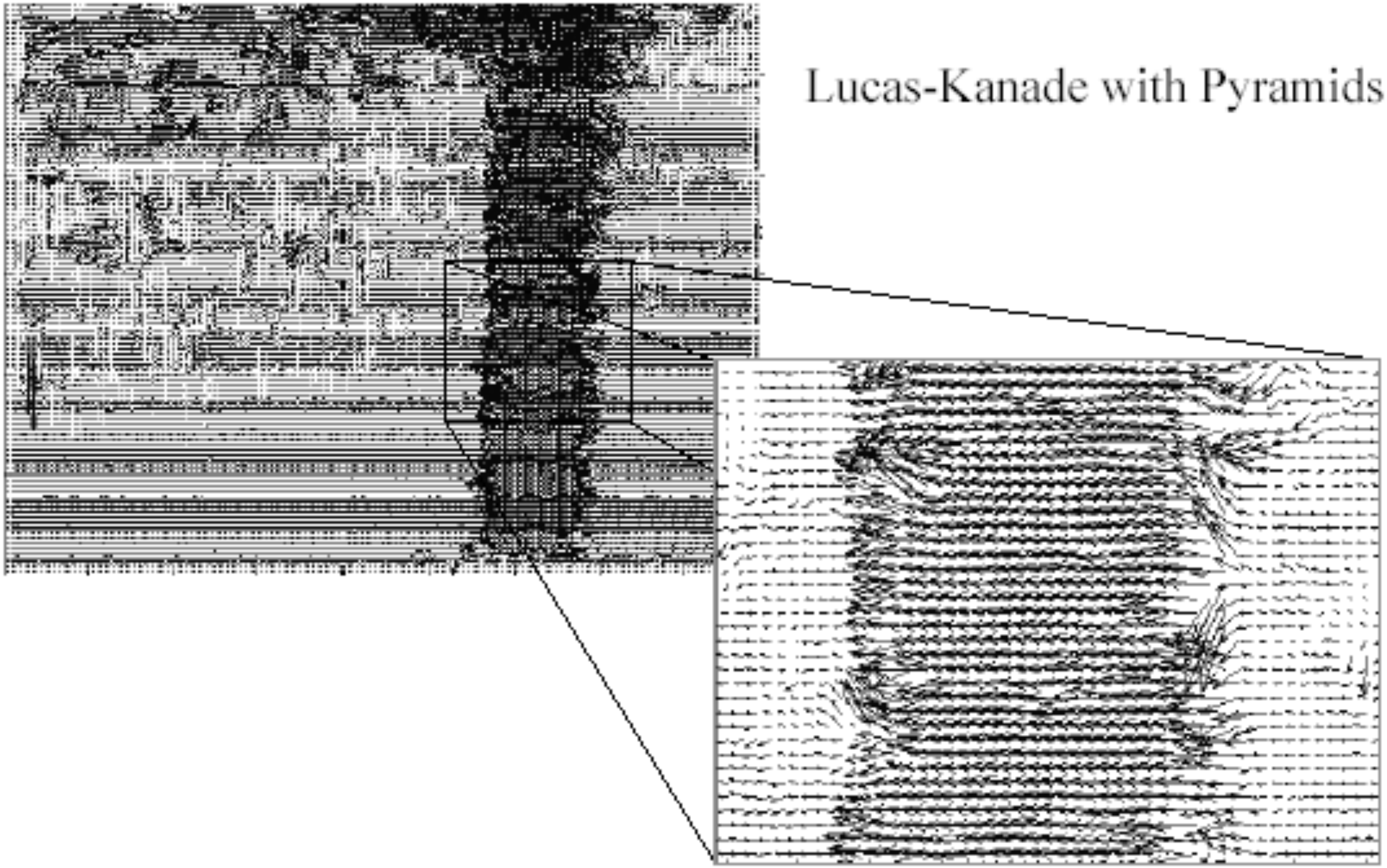


Lucas-Kanade without pyramids

Fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Optical Flow Results



* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Errors in Lucas-Kanade

- The motion is large (larger than a pixel)
  - Coarse-to-fine estimation
  - Iterative refinement
  - Exhaustive neighborhood search (feature matching)
- A point does not move like its neighbors
  - Motion segmentation
- Brightness constancy does not hold
  - Exhaustive neighborhood search with normalized correlation

Large displacement optical flow

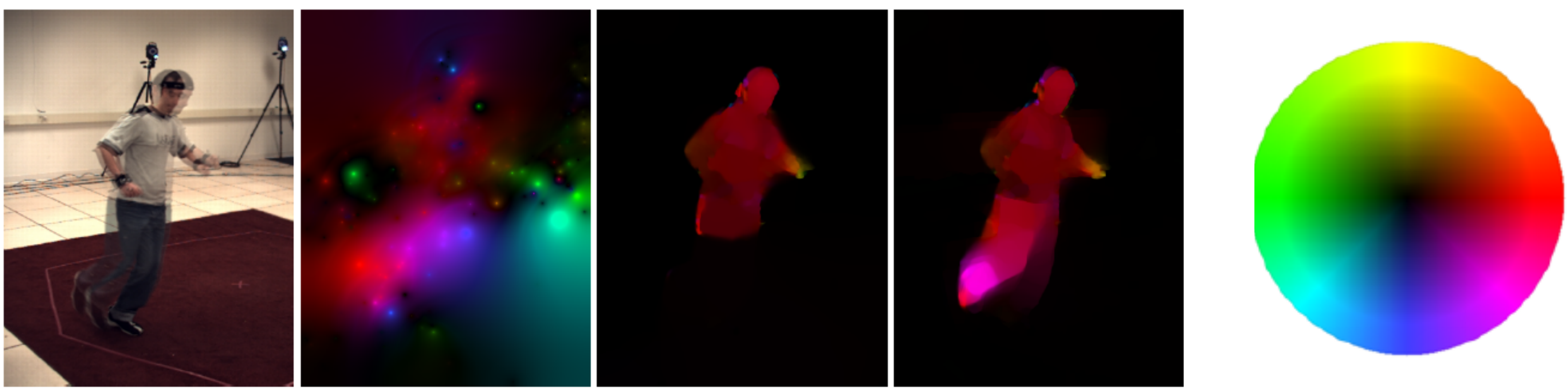
Start with something similar to Lucas-Kanade

+ gradient constancy

+ energy minimization with smoothing term

+ region matching

+ keypoint matching (long-range)

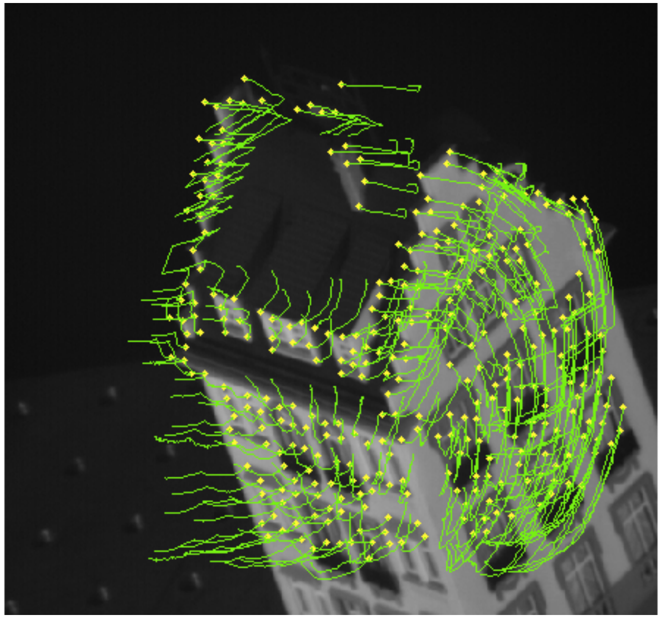


Region-based +Pixel-based +Keypoint-based

Large displacement optical flow, Brox et al., CVPR 2009

Feature tracking

- If we have more than two images, we can track a feature from one frame to the next by following the optical flow
- Challenges
  - Finding good features to track
  - Adding and deleting tracks



Shi-Tomasi feature tracker

- Find good features using eigenvalues of second-moment matrix
  - Key idea: “good” features to track are the ones whose motion can be estimated reliably
- From frame to frame, track with Lucas-Kanade
  - This amounts to assuming a translation model for frame-to-frame feature movement
- Check consistency of tracks by *affine* registration to the first observed instance of the feature
  - Affine model is more accurate for larger displacements
  - Comparing to the first frame helps to minimize drift

J. Shi and C. Tomasi. Good Features to Track. CVPR 1994.

Tracking example

Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

J. Shi and C. Tomasi. Good Features to Track. CVPR 1994.

Summary of KLT tracking

- Find a good point to track (harris corner)
- Use intensity second moment matrix and difference across frames to find displacement
- Iterate and use coarse-to-fine search to deal with larger movements
- When creating long tracks, check appearance of registered patch against appearance of initial patch to find points that have drifted

Implementation issues

- Window size
  - Small window more sensitive to noise and may miss larger motions (without pyramid)
  - Large window more likely to cross an occlusion boundary (and it's slower)
  - 15x15 to 31x31 seems typical

- Weighting the window
  - Common to apply weights so that center matters more (e.g., with Gaussian)

Why not just do local template matching?

- Slow (need to check more locations)
- Does not give subpixel alignment (or becomes much slower)
  - Even pixel alignment may not be good enough to prevent drift
- May be useful as a step in tracking if there are large movements

Summary

- Major contributions from Lucas, Kanade, Shi, Tomasi
  - 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