Lecture 10

Motion and Tracking

CSE 455 Roadmap

Pixels	Video	Camera	Segment	ML
Convolutions Edges Descriptors	Motion Tracking	Camera 3D Geometry	Segmentation Clustering Detection	Linear Models (Conv) Neural networks

Today's agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

Today's agenda

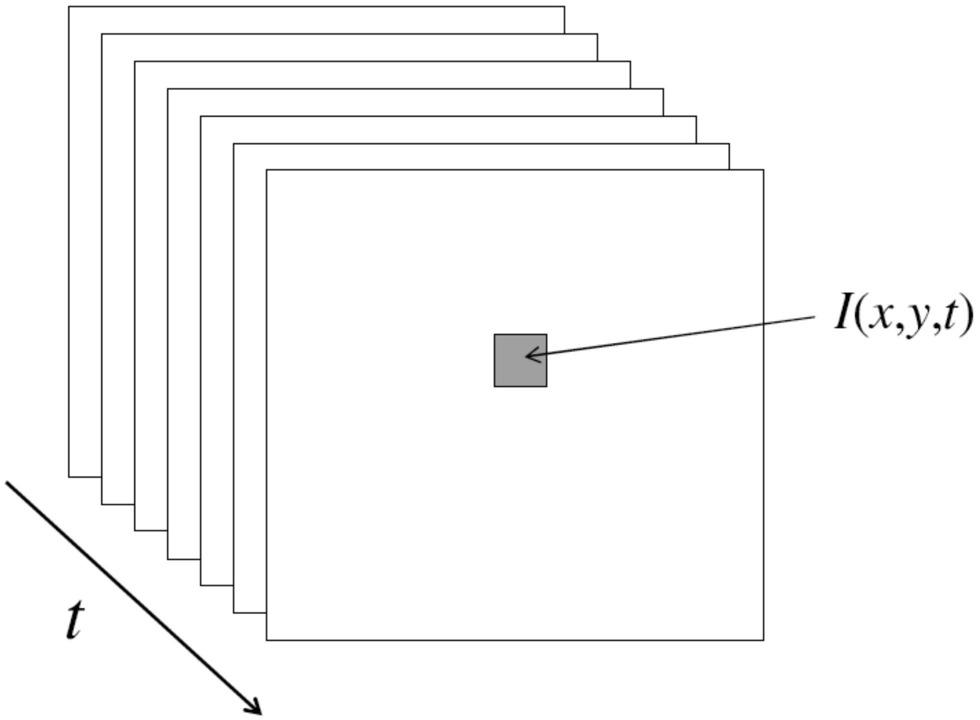
- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

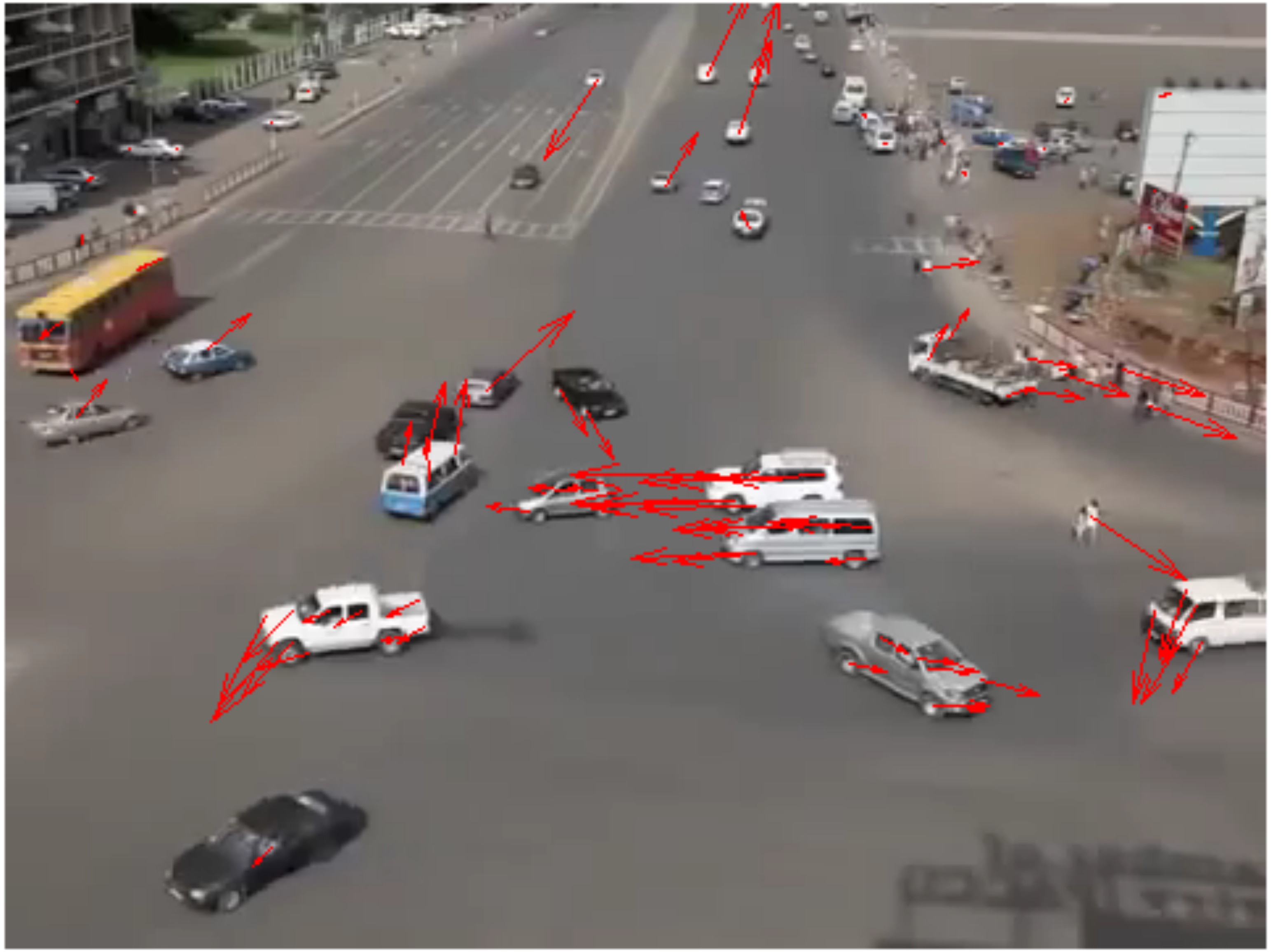
From images to videos

- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



Why is motion useful?

Why is motion useful?



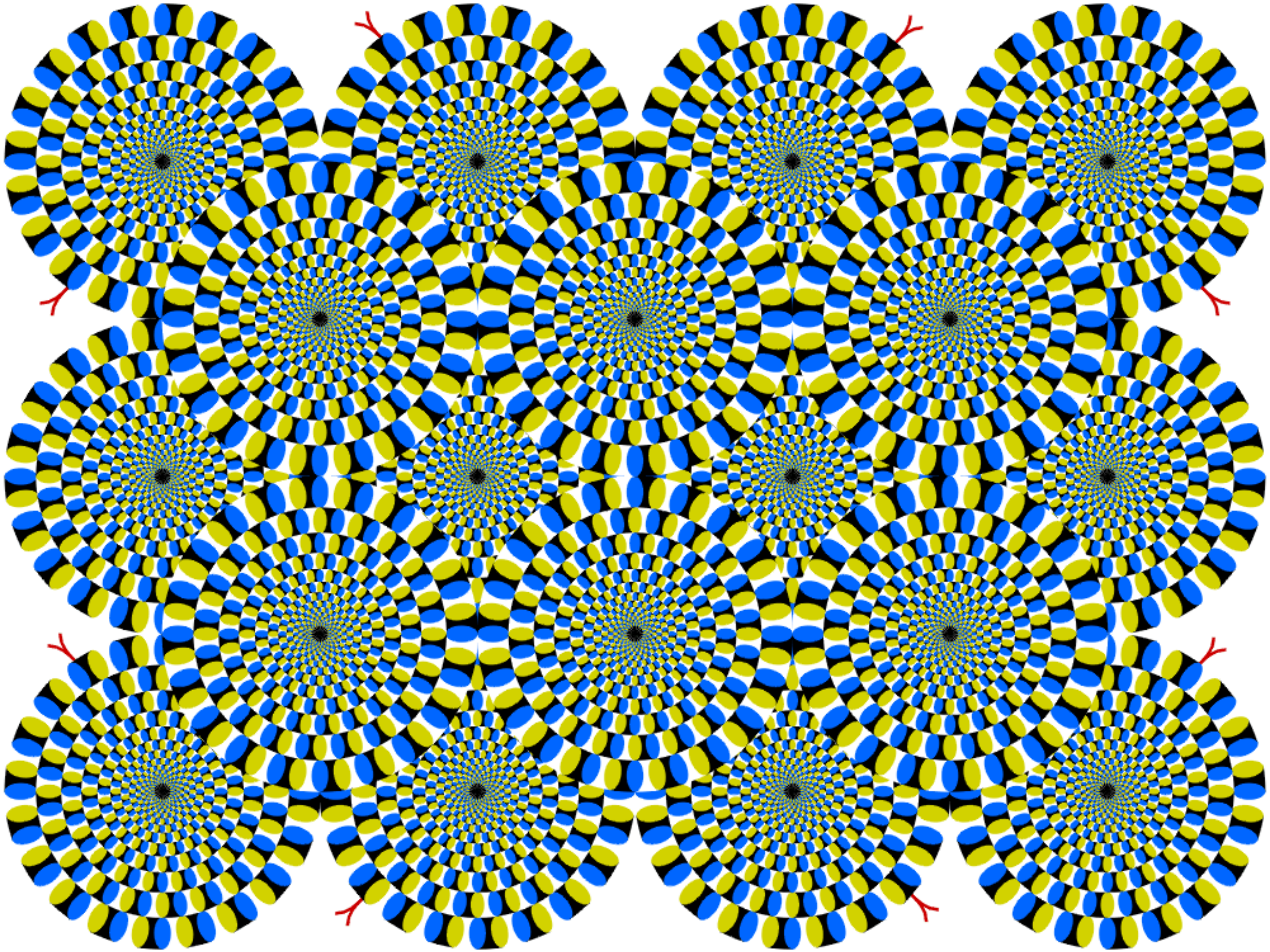
Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image

- Note: apparent motion can be caused by lighting changes without any actual motion
  - Think of a uniform rotating sphere under fixed lighting (has motion but no optical flow)
  - vesus a stationary sphere under moving illumination (no motion but has optical flow)

GOAL: Recover image motion at each pixel from optical flow

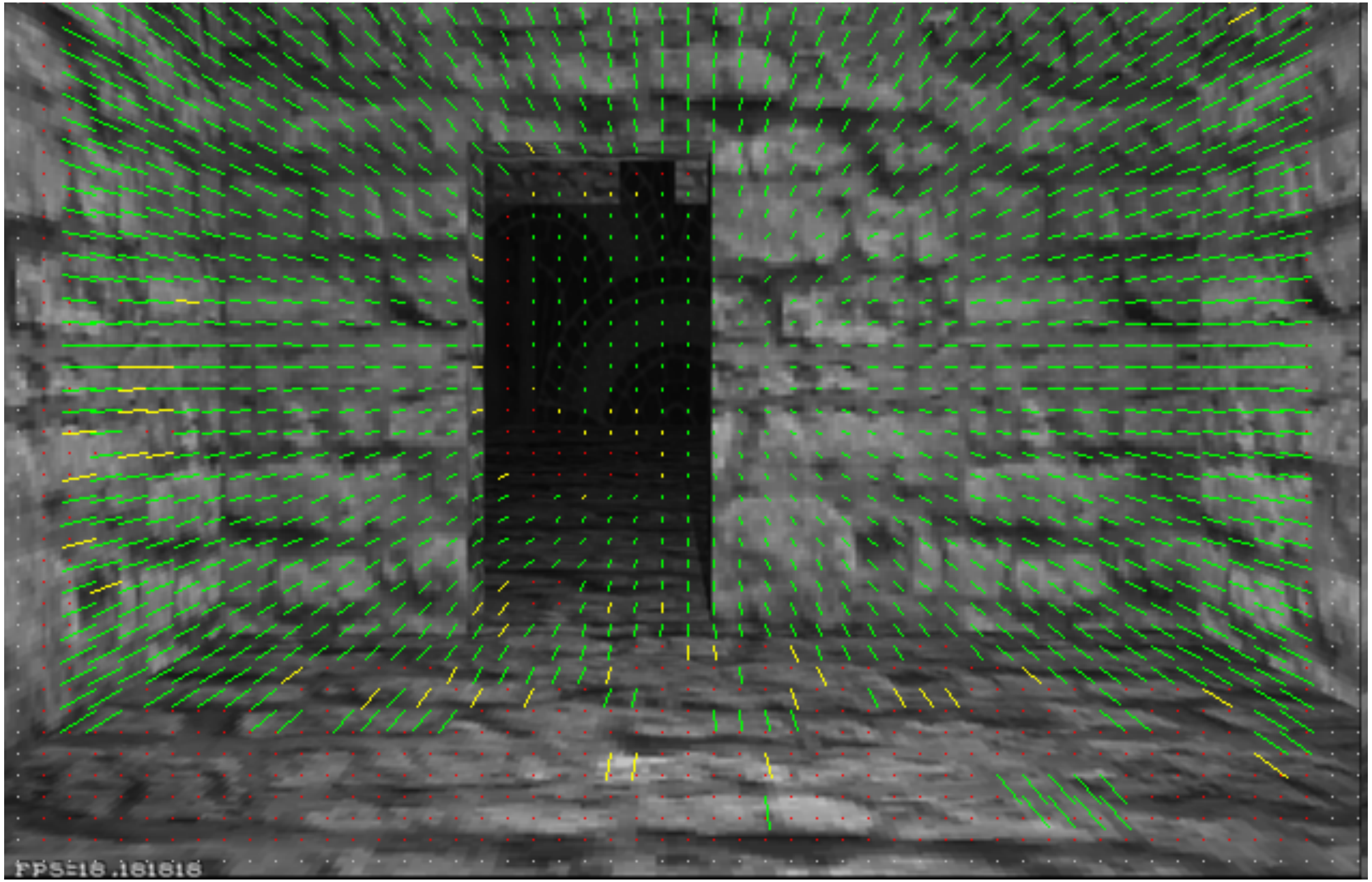
Optical flow without motion!



Optical flow

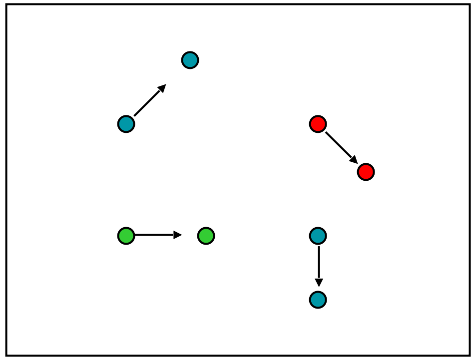
of an image gives us the apparent motion of every pixel

It is a function of the spatio-temporal image brightness variations

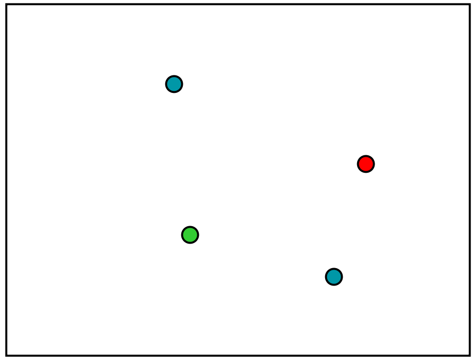


Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

Formalizing optical flow



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



$I(x,y,t)$

- Given two subsequent frames,
- estimate the apparent motion field u(x,y), v(x,y) between them
- u(x, y) measuring the horizontal movement of the pixel at location (x, y).
- v(x, y) measures the vertical movement.

- Together, the pixel at (x, y, t-1) goes to (x+u, y+v, t)

3 assumptions when estimating optical flow

1. **small motions**: points do not move very far

2. **spatial coherence**: points move like their neighbors

3. **brightness constancy**: the brightness of a pixel remains constant between consecutive frames

Key Assumptions: small motions

The **small motions assumption**:
Between consecutive frames the change in pixel locations is small

* Slide from Michael Black, CS143 2003

Key Assumptions: spatial coherence

The **spatial coherence assumption**: Neighboring pixels typically move together because they belong to the same rigid object.

* Slide from Michael Black, CS143 2003

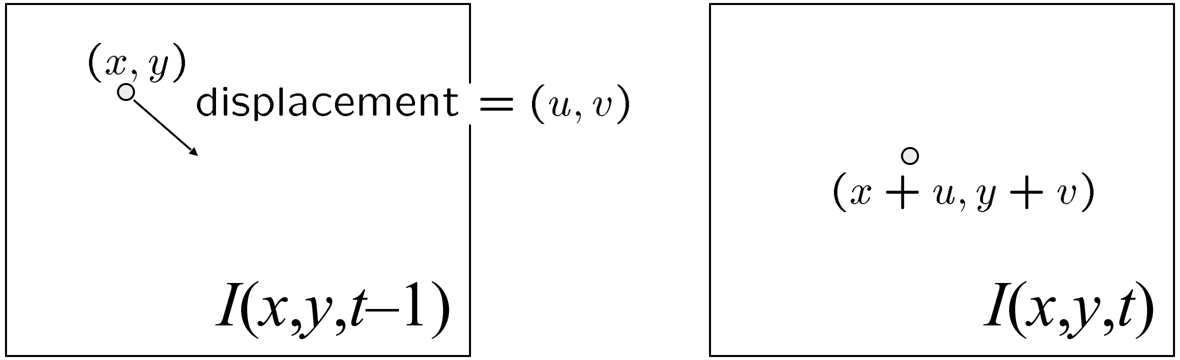
Key Assumptions: brightness constancy

The **brightness constancy assumption**: Average brightness of pixels in a patch stays the same across consecutive frames, although their location might change

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

* Slide from Michael Black, CS143 2003

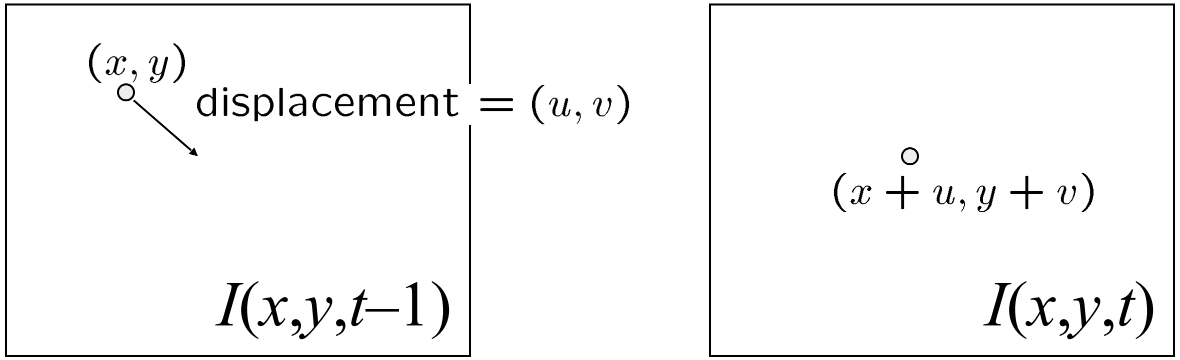
The brightness constancy constraint



- Brightness Constancy Equation:

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

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:

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

Image derivative along x

Image derivative along t

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

Hence, $I_x \cdot u + I_y \cdot v + I_t \approx 0 \quad \rightarrow \nabla I \cdot [u \ v]^T + I_t = 0$

Derivative filters are now 3 dimensional

Derivative in **x direction** now has a new dimension looking at past and future frames

new Time dimension

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Derivative in x doesn't look at frame t-1

$$I_x = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix}$$

Backward derivative at frame t

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Derivative in the x direction doesn't look at frame t+1

Similar for y direction

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Derivative in x doesn't look at frame t-1

$$I_x = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Backward derivative at frame t

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Derivative in the x direction doesn't look at frame t+1

new Time dimension

New backward derivative in the time t dimension

new Time dimension

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Derivative in x doesn’t look at frame t-1

$$I_t = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Backward derivative at frame t

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Derivative in the x direction doesn’t look at frame t+1

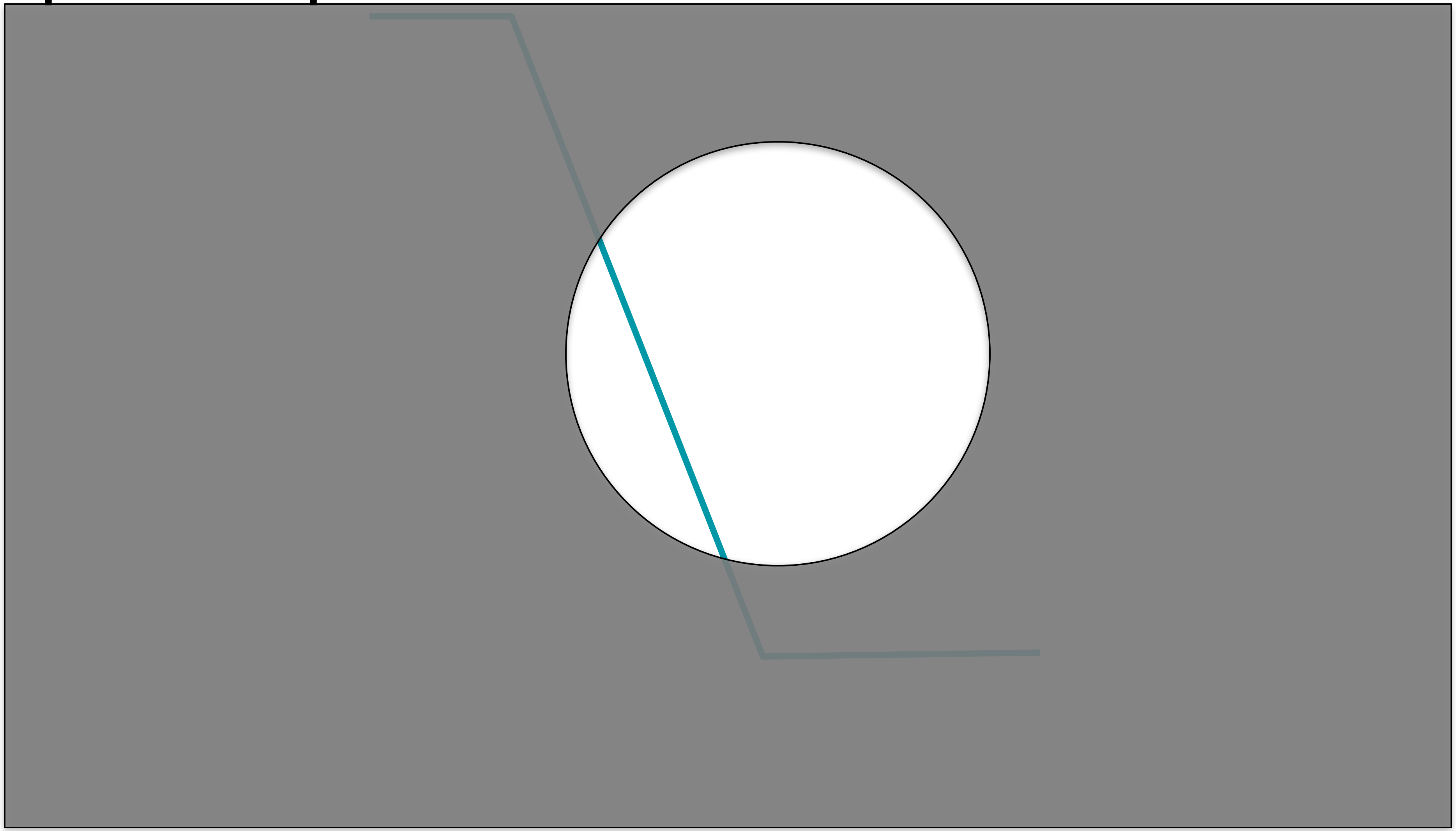
The brightness constancy constraint

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

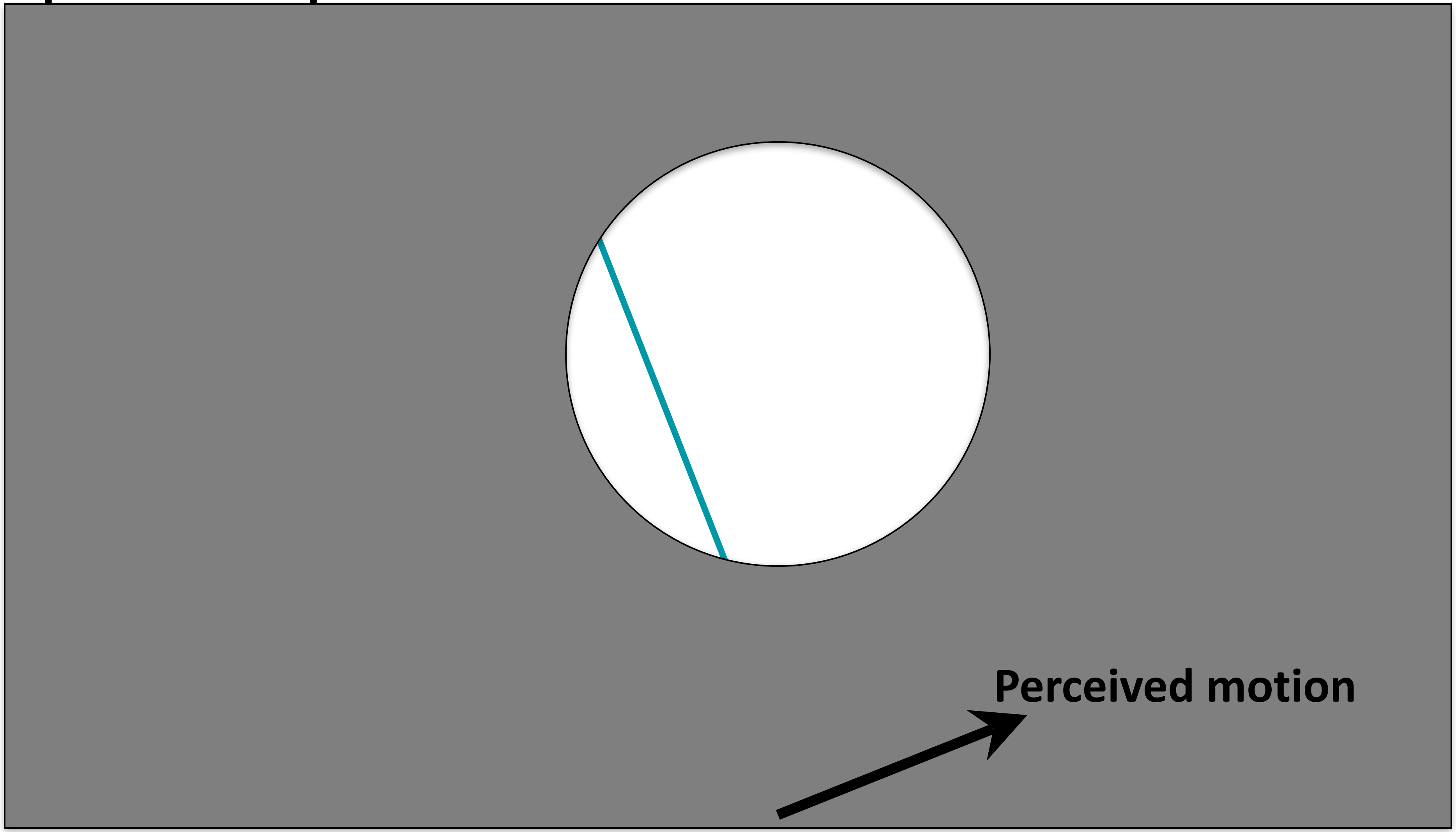
$$\nabla I \cdot [u \ v]^T + I_t = 0$$

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

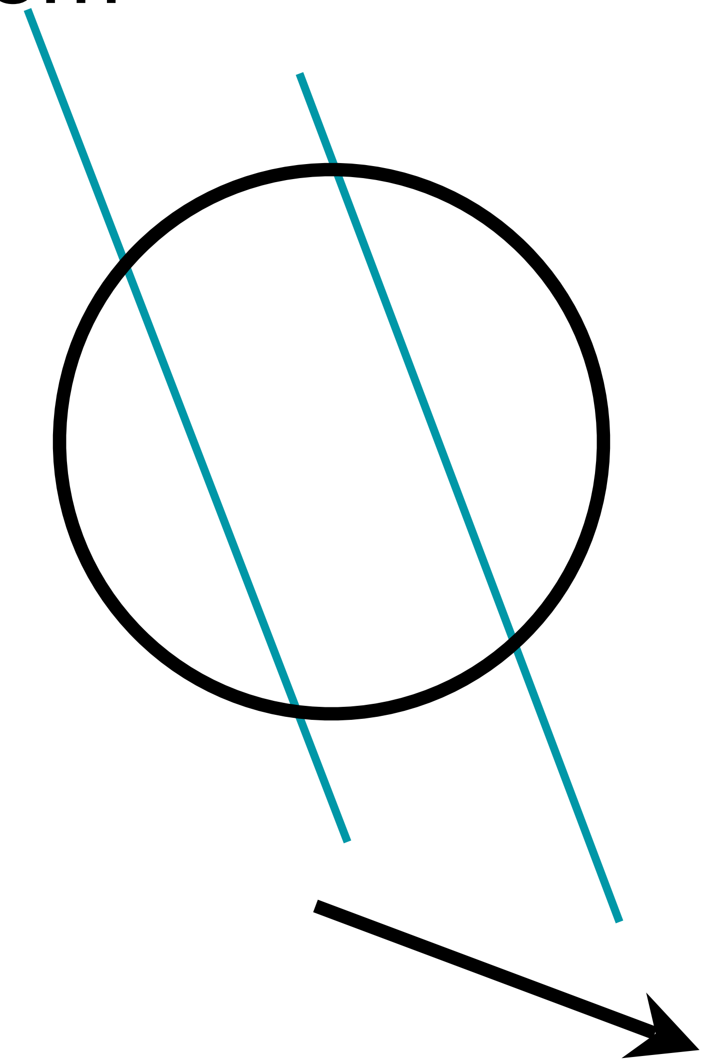
The aperture problem



The aperture problem

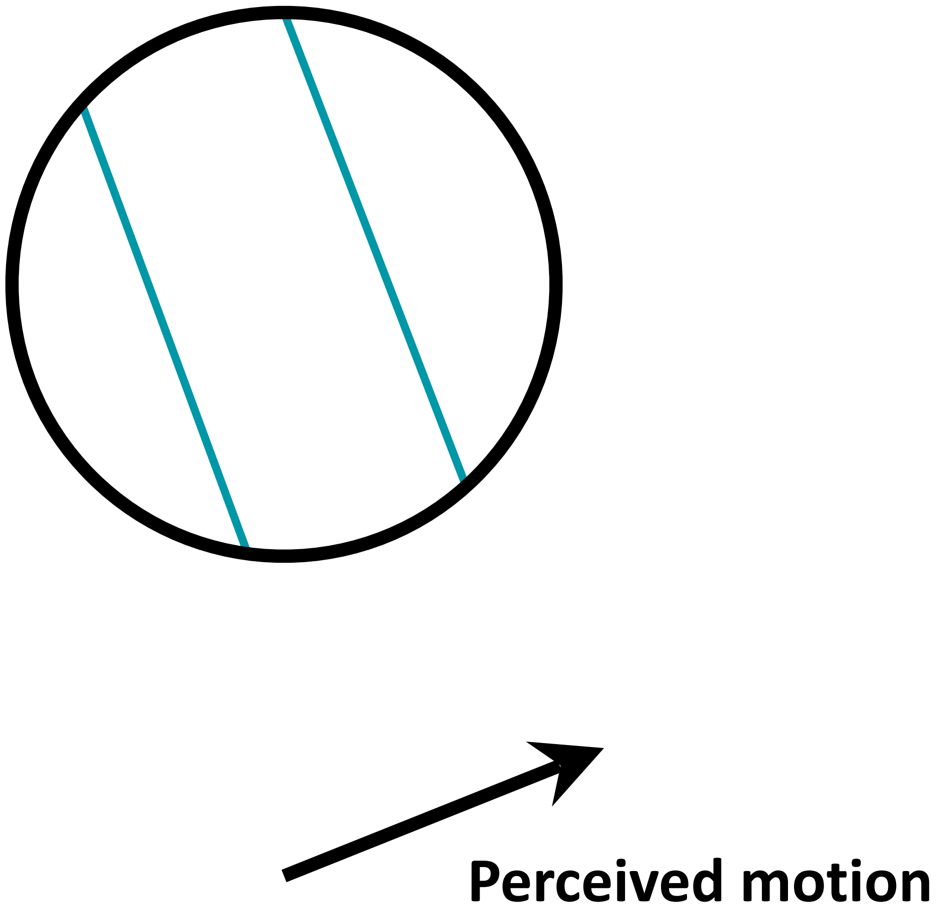


The aperture problem



Actual motion

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

The brightness constancy constraint

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

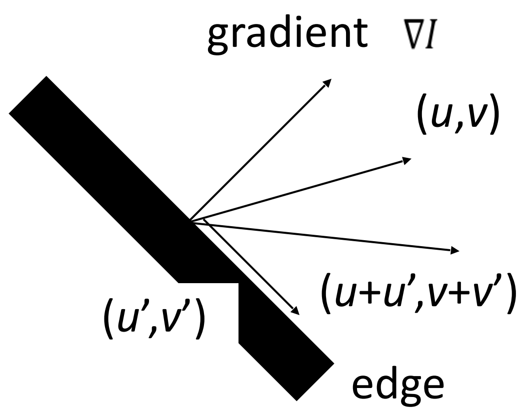
$$\nabla I \cdot [u \ v]^T + I_t = 0$$

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

Problem: The component of the flow perpendicular to the gradient (i.e., parallel to the edge) cannot be measured

If (u, v) satisfies the equation,
so does ($u+u'$, $v+v'$) if

$$\nabla I \cdot [u' \ v']^T = 0$$



Today’s agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

How to get more equations for a pixel?

- **Add in the 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

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

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

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

- Overconstrained linear system:

$$\begin{bmatrix} I_x(\mathrm{p}_1) & I_y(\mathrm{p}_1) \\ I_x(\mathrm{p}_2) & I_y(\mathrm{p}_2) \\ \vdots & \vdots \\ I_x(\mathrm{p}_{25}) & I_y(\mathrm{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathrm{p}_1) \\ I_t(\mathrm{p}_2) \\ \vdots \\ I_t(\mathrm{p}_{25}) \end{bmatrix} \qquad \underset{25\mathrm{x}2}{A} \ \underset{2\mathrm{x}1}{d} = \underset{25\mathrm{x}1}{b}$$

Lucas-Kanade flow

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix} \qquad \underset{25\text{x}2}{A} \; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Multiplying by A^T to solve for *d* gives us: $(A^T A)\ d = A^T b$

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

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

Conditions for solving this Lucas-Kanade equation

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

$$A^T A \qquad\qquad\qquad\qquad A^T b$$

When is This Solvable?

- **A^{T}A** should be invertible
- **A^{T}A** should not be too small, otherwise it is close to being non-invertible
  - 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)

Q. Does this remind anything to you?

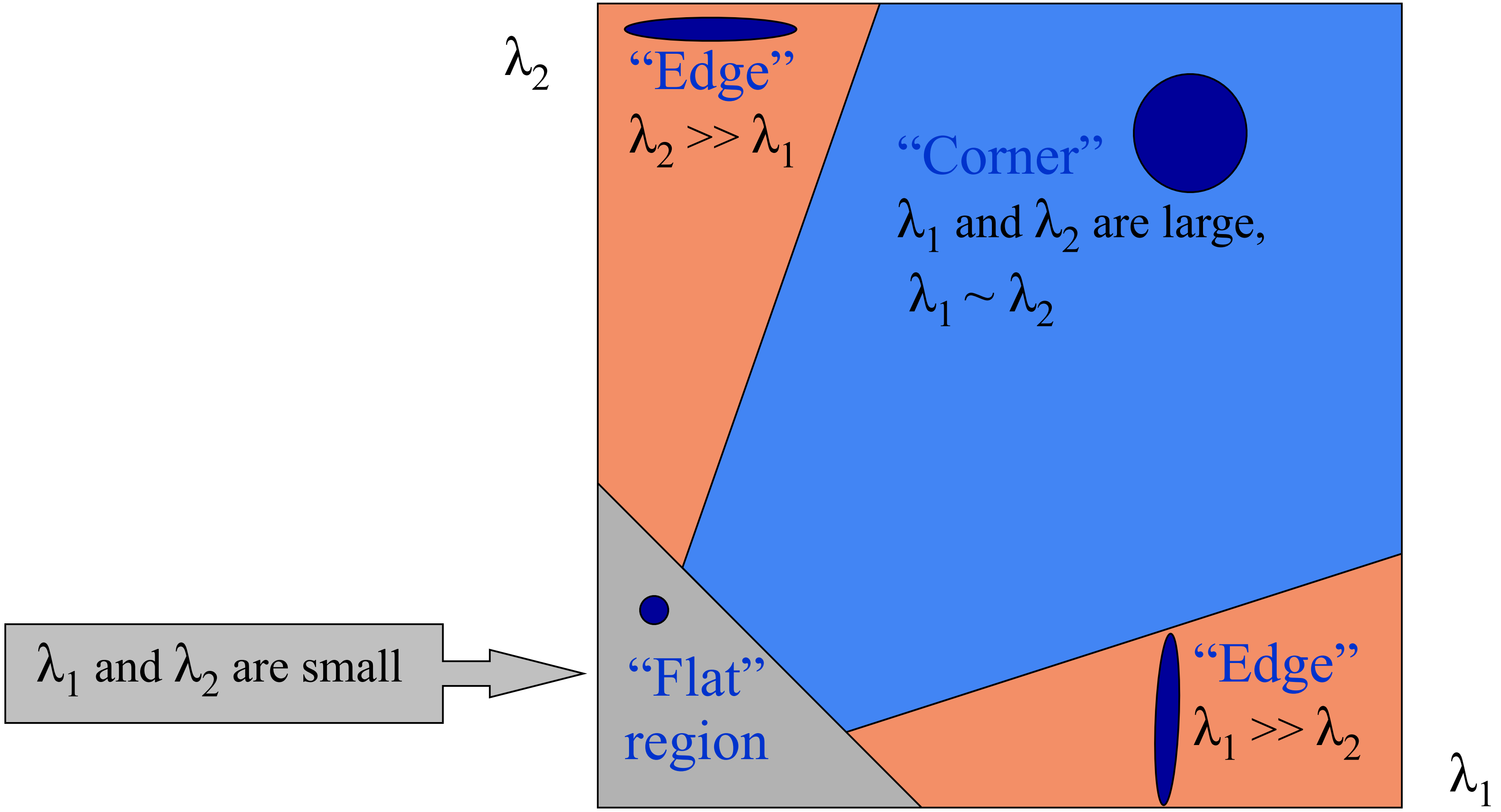
$M = A^TA$ is the Harris corner detector!

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

- Eigenvectors and eigenvalues of A^TA 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

Interpreting the eigenvalues

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



Edges are harder to track

All the points on an edge look the same. It is hard to estimate where each point will move to.

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

– gradients very large or very small

– large λ_1, small λ_2

Low-texture region

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

– gradients have small magnitude

– small λ_1, small λ_2

Low-texture regions have small eigenvalues. The matrix is harder to invert and get accurate estimates of optical flow

High-texture region

These points are easier to estimate optical flow for.

This makes sense intuitively: You could say that corners and blobs (things that are easier to detect) are easier to track over time.

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

– gradients are different, large magnitudes

– large λ_1, large λ_2

Quizz

https://tinyurl.com/cse455-10

Errors in Lucas-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?

Improving accuracy

- Recall our small motion assumption

$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

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

$$I_x \cdot u + I_y \cdot v + \text{higher order terms} + I_t \approx 0$$

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

Iterative Lucas-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

   Calculate I(t) using the calculated optical flow
3. Repeat until convergence

When do the optical flow assumptions fail?

In other words, in what situations does the displacement of pixel patches not represent physical movement of points in space?

1. Well, television (movies) screens appear to contain objects in motion
   – Yet our TVs and monitors are actually stationary
2. Motion that doesn't cause changes in pixels
   – e.g. A uniform rotating sphere. Nothing seems to move, yet it is rotating
3. Lighting changes can make things seem to move
   – for example, if a singular light source moves around a stationary sphere
4. Smaller motions might move in a direction opposite to motion
   – E.g. a cheetah's muscles move opposite direction of motion.

Today’s agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Key assumptions (Errors in Lucas-Kanade)

- **Small motion:** points do not move very far

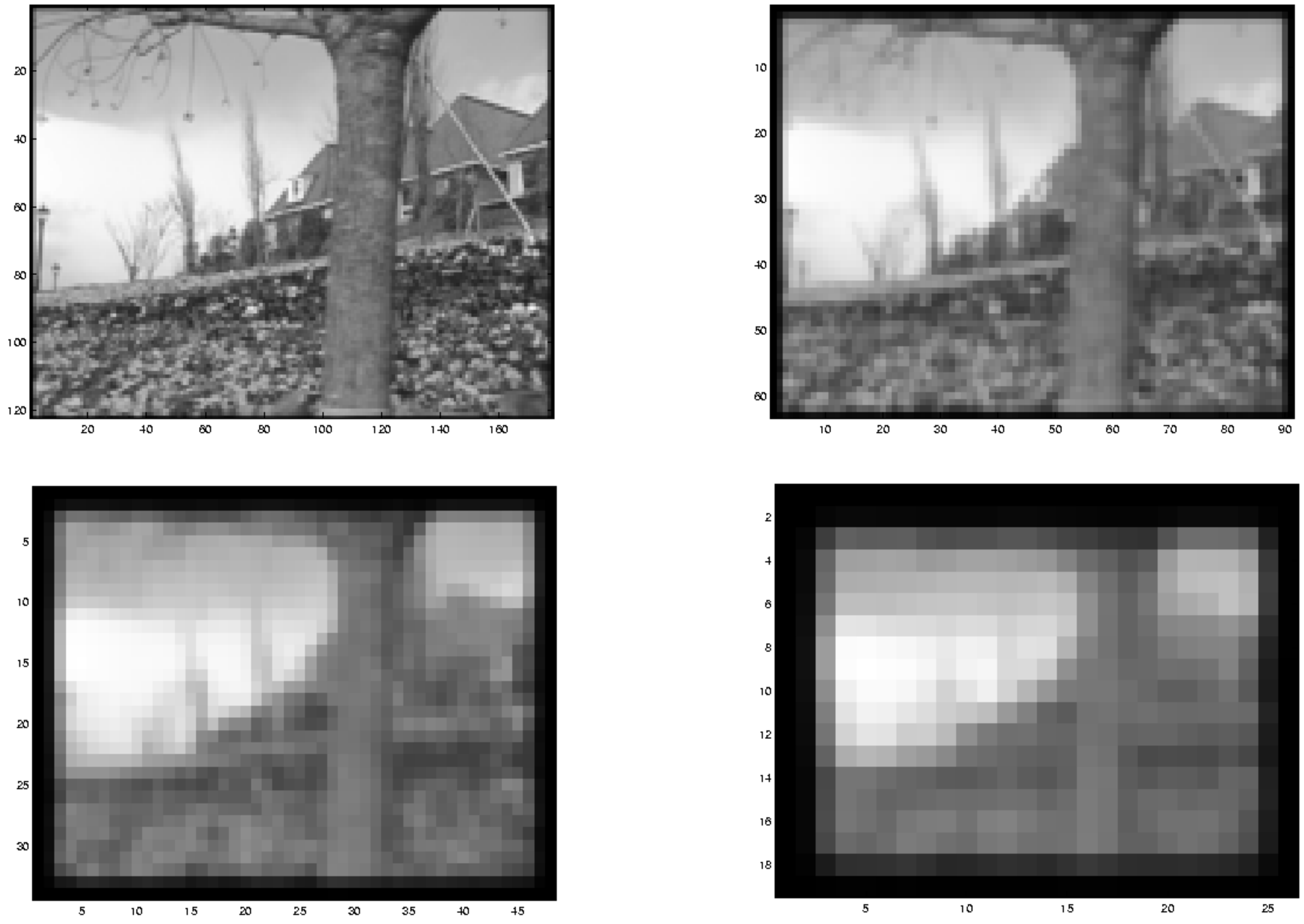
- **Brightness constancy:** the brightness of a pixel remains constant between consecutive frames

- **Spatial coherence:** points move like their neighbors

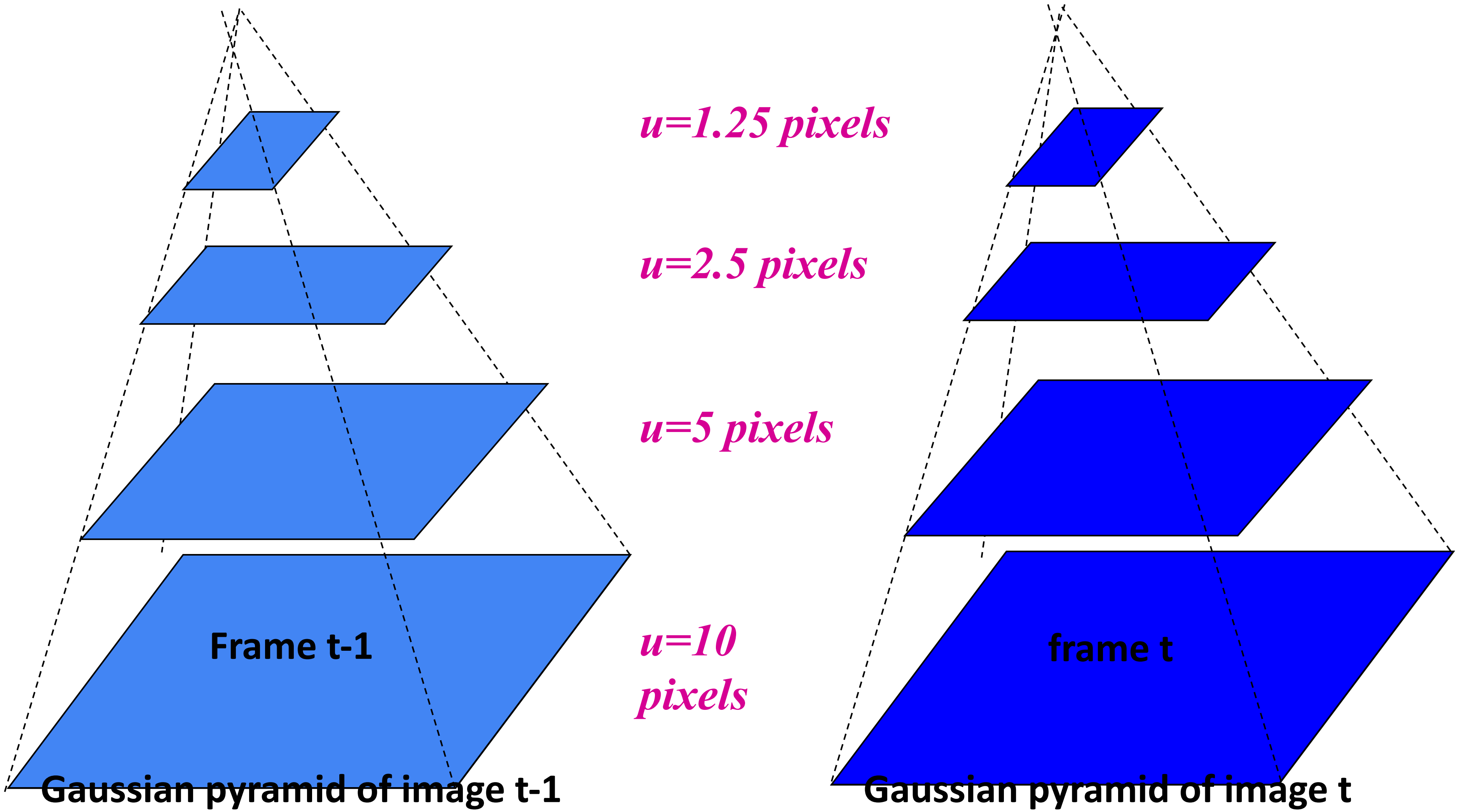
Revisiting the small motion assumption

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

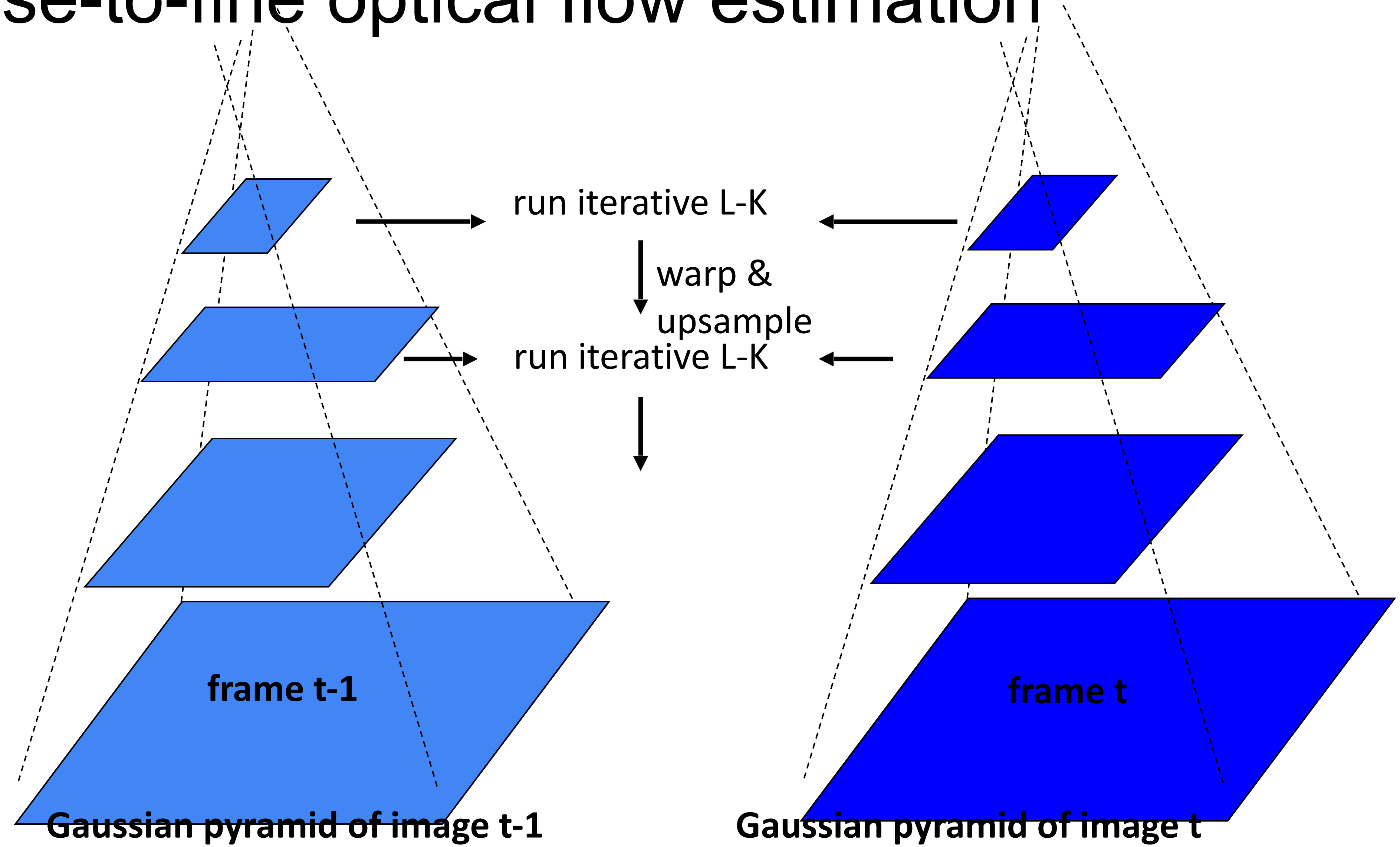
Reduce the resolution so that assumption holds



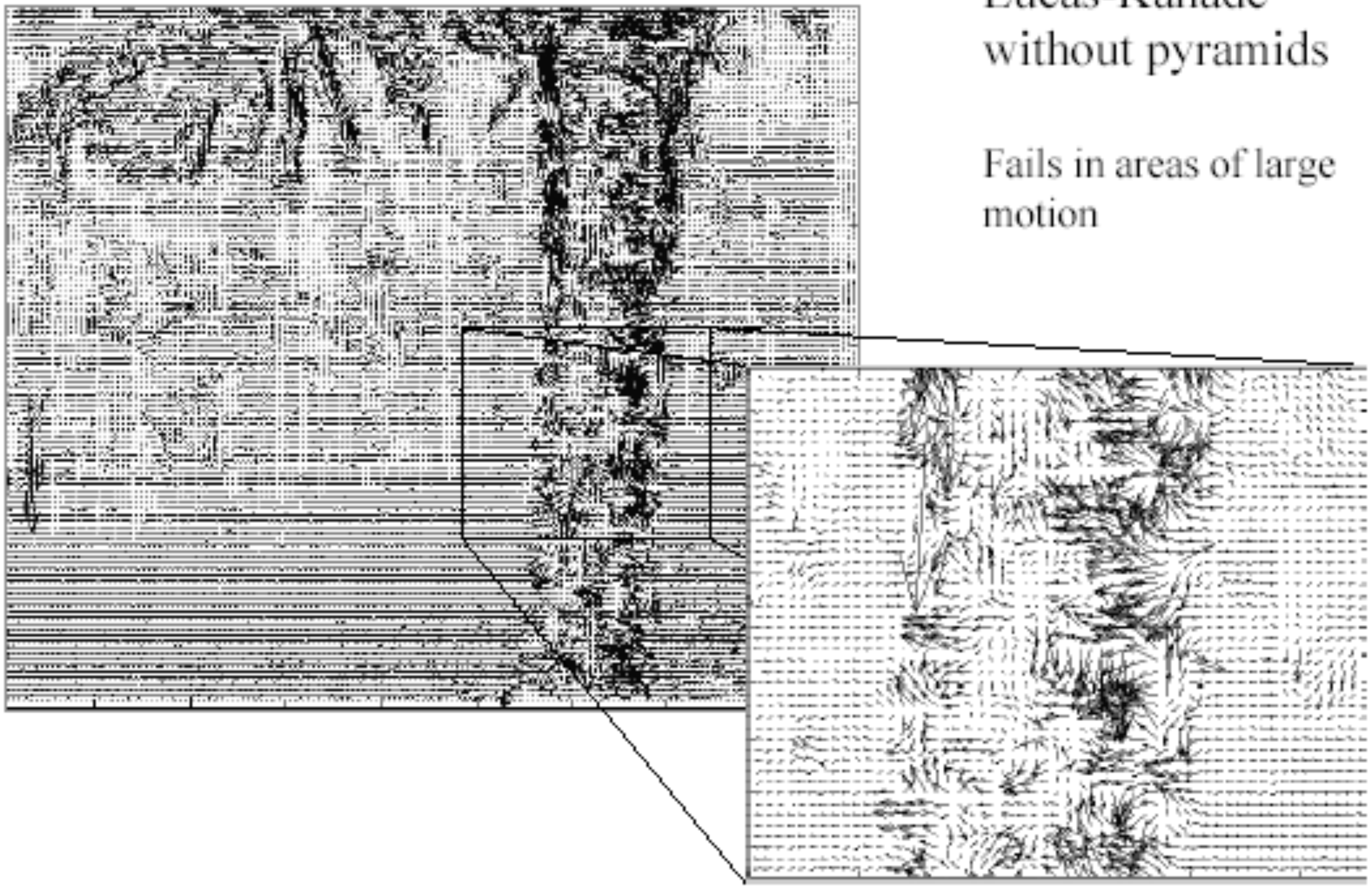
Coarse-to-fine optical flow estimation



Coarse-to-fine optical flow estimation



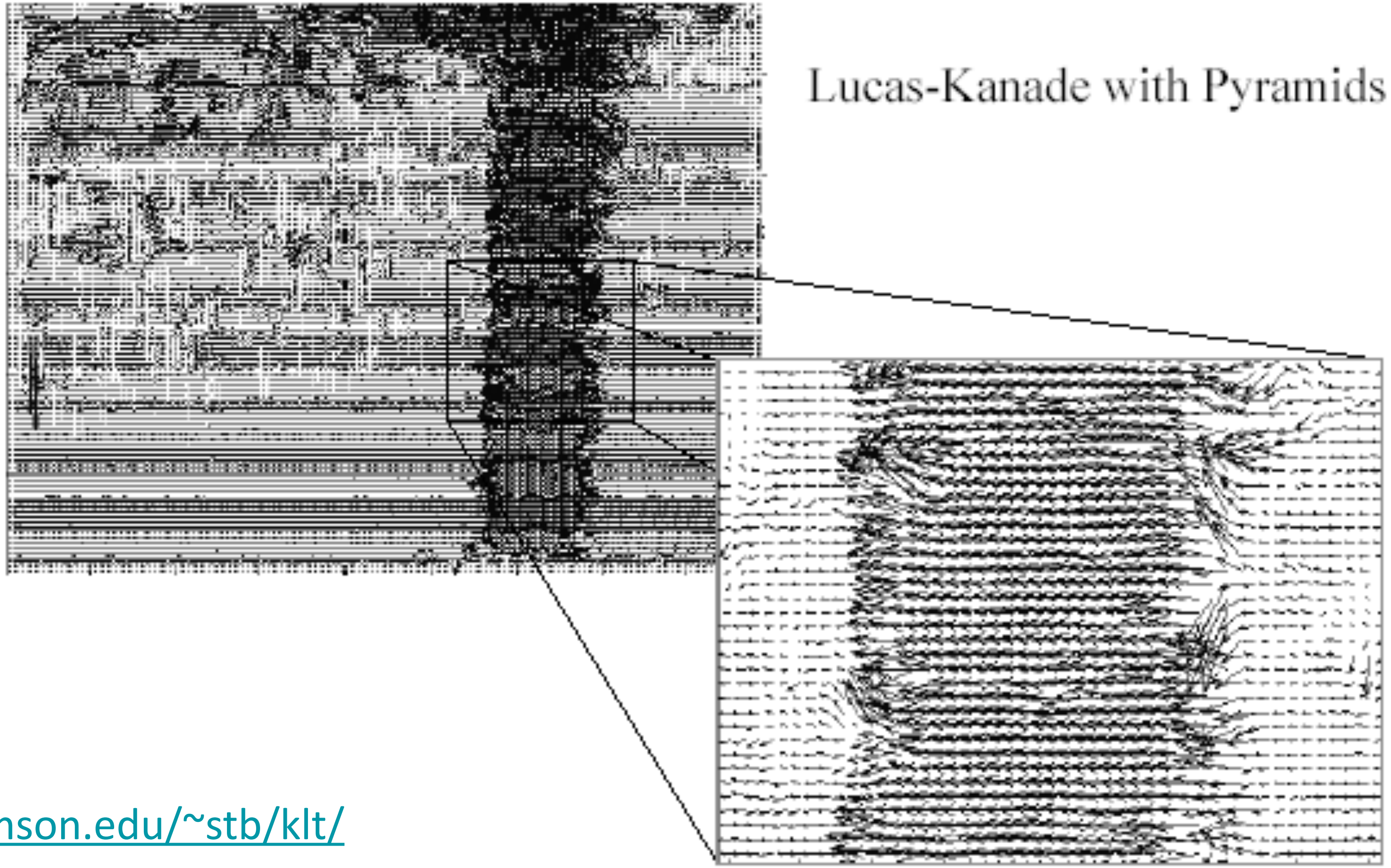
Optical Flow Results



Lucas-Kanade without pyramids

Fails in areas of large motion

Optical Flow Results



- http://www.ces.clemson.edu/~stb/klt/
- OpenCV

Today’s agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

Key assumptions (Errors in Lucas-Kanade)

- **Small motion:** points do not move very far
- **Brightness constancy:** the brightness of a pixel remains constant between consecutive frames
- **Spatial coherence:** points move like their neighbors

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:
- The first part of the function is the brightness constancy.

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:
- The second part is the smoothness constraint. It’s trying to make sure that the changes between pixels are small.

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right) \right] \mathrm{d}x \mathrm{d}y$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:
- ***α*** is a regularization constant. Larger values of ***α*** lead to smoother flows across time.

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

Horn-Schunk method for optical flow

- The flow is formulated as a global energy function which is should be minimized:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x \mathrm{d}y$$

- This minimization can be solved by taking the derivative with respect to u and v, we get the following 2 equations:

$$I_x (I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0$$
$$I_y (I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0$$

Horn-Schunk method for optical flow

- By taking the derivative with respect to u and v, we get the following 2 equations:

$$I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0$$
$$I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0$$

- Where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is called the Lagrange operator. It is hard to calculate. So we estimate it using $\Delta u(x,y) = \overline{u}(x,y) - u(x,y)$
  Intuition: Lagrange is the second derivative. The estimation measures the deviation from the average change.

- where $\overline{u}(x,y)$ is the weighted average of u measured at (x,y) over its neighborhood of 5 x 5 pixels

Horn-Schunk method for optical flow

- Now we substitute $\Delta u(x,y) = \overline{u}(x,y) - u(x,y)$ in:

$$I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0$$
$$I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0$$

- To get:

$$(I_x^2 + \alpha^2)u + I_x I_y v = \alpha^2 \overline{u} - I_x I_t$$
$$I_x I_y u + (I_y^2 + \alpha^2)v = \alpha^2 \overline{v} - I_y I_t$$

- Which is linear in u and v and can be solved analytically for each pixel individually.

Horn-Schunk method for optical flow

- Analytical solution for:

$$(I_x^2 + \alpha^2)u + I_x I_y v = \alpha^2 \overline{u} - I_x I_t$$
$$I_x I_y u + (I_y^2 + \alpha^2)v = \alpha^2 \overline{v} - I_y I_t$$

- is:

$$u = \bar{u} - \frac{I_x(I_x\bar{u} + I_y\bar{v} + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

$$v = \bar{v} - \frac{I_y(I_x\bar{u} + I_y\bar{v} + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

Iterative Horn-Schunk

- Similar to iterative Lucas-Kanade, there is an iterative version of Horn-Schunk algorithm.
- Since the solution depends on $\overline{u}$ and $\overline{v}$ this calculation becomes more accurate as we iteratively update the average flow.

- After each calculate, re-calculate $\overline{u}$ and $\overline{v}$

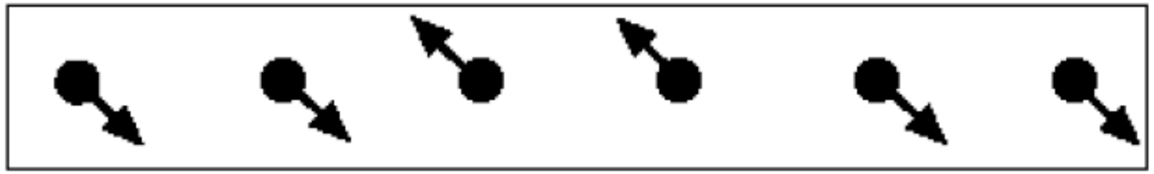
What we will learn today?

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Key assumptions

- **Small motion:** points do not move very far
- **Brightness constancy:** the brightness of a pixel remains constant between consecutive frames
- **Spatial coherence:** points move like their neighbors

Gestalt – common fate



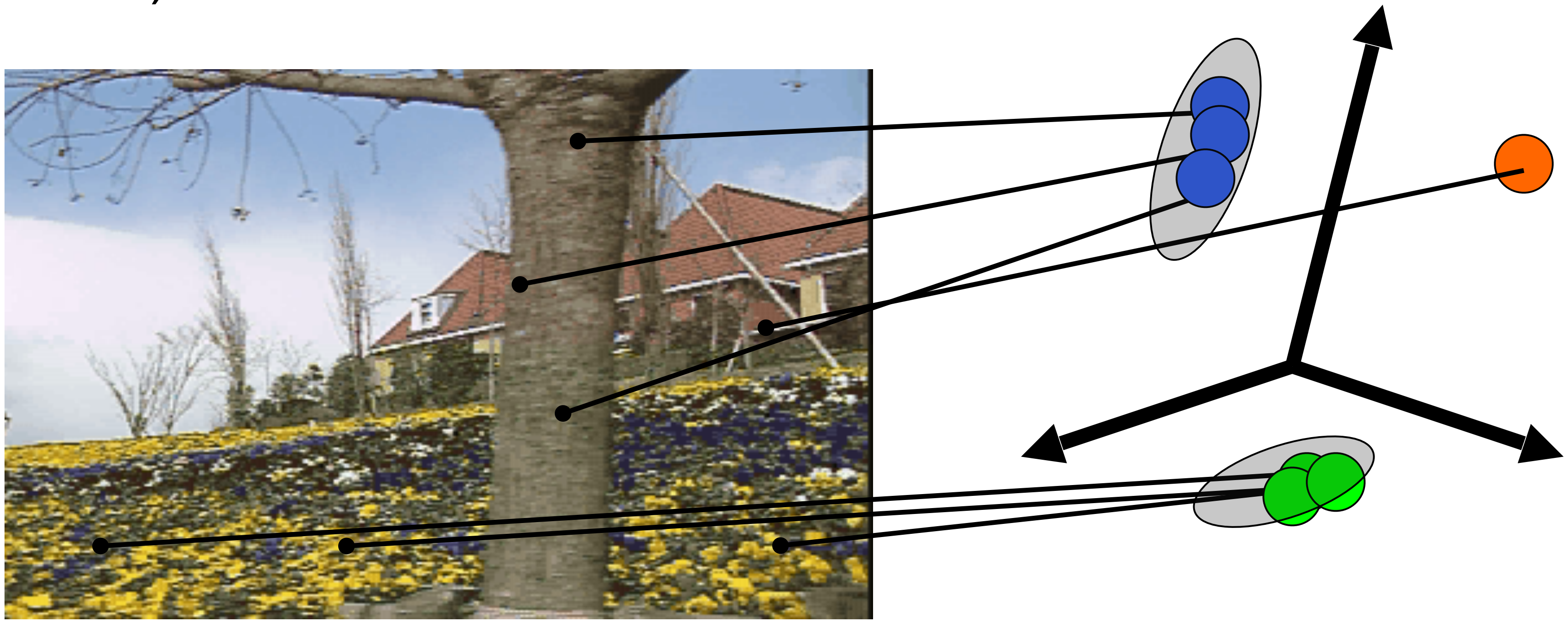
Common Fate

Segmentation using motion

- Use optical flow as the feature representation of each pixel

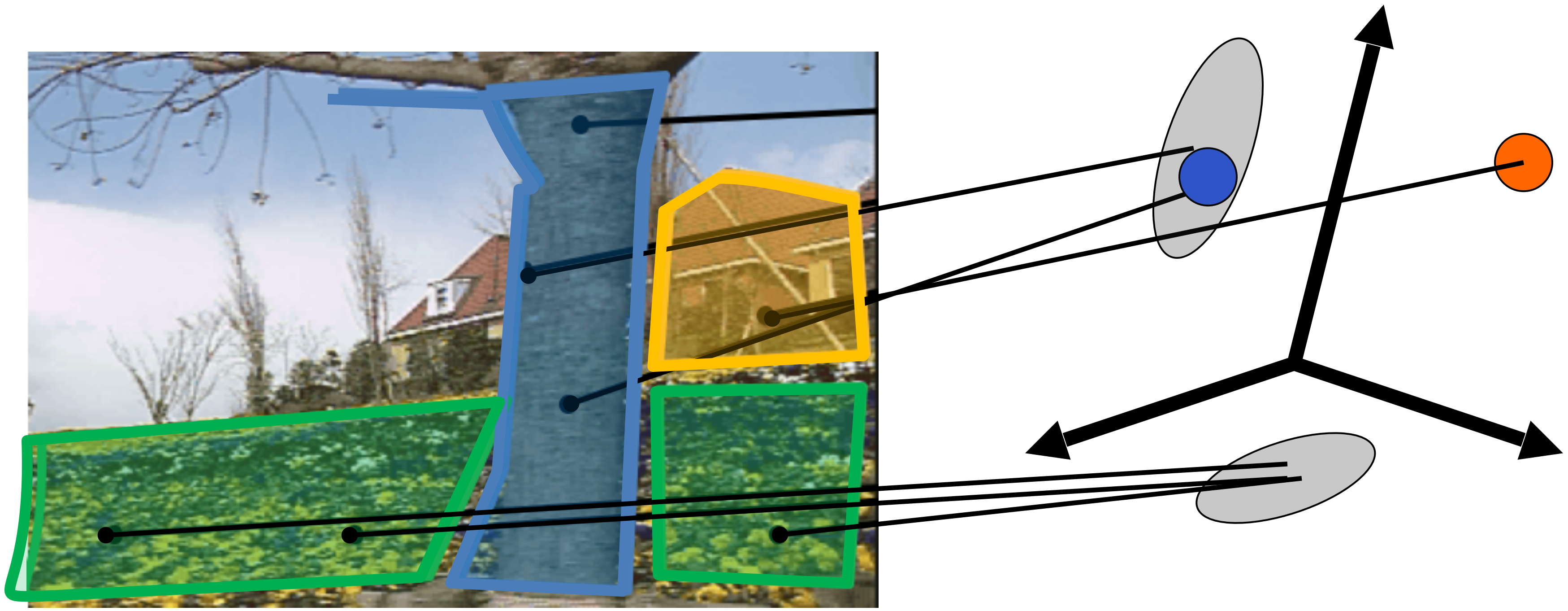
Segmentation using motion

- Use any of the segmentation algorithms: (k-means, Agglomerative clustering, mean shift)



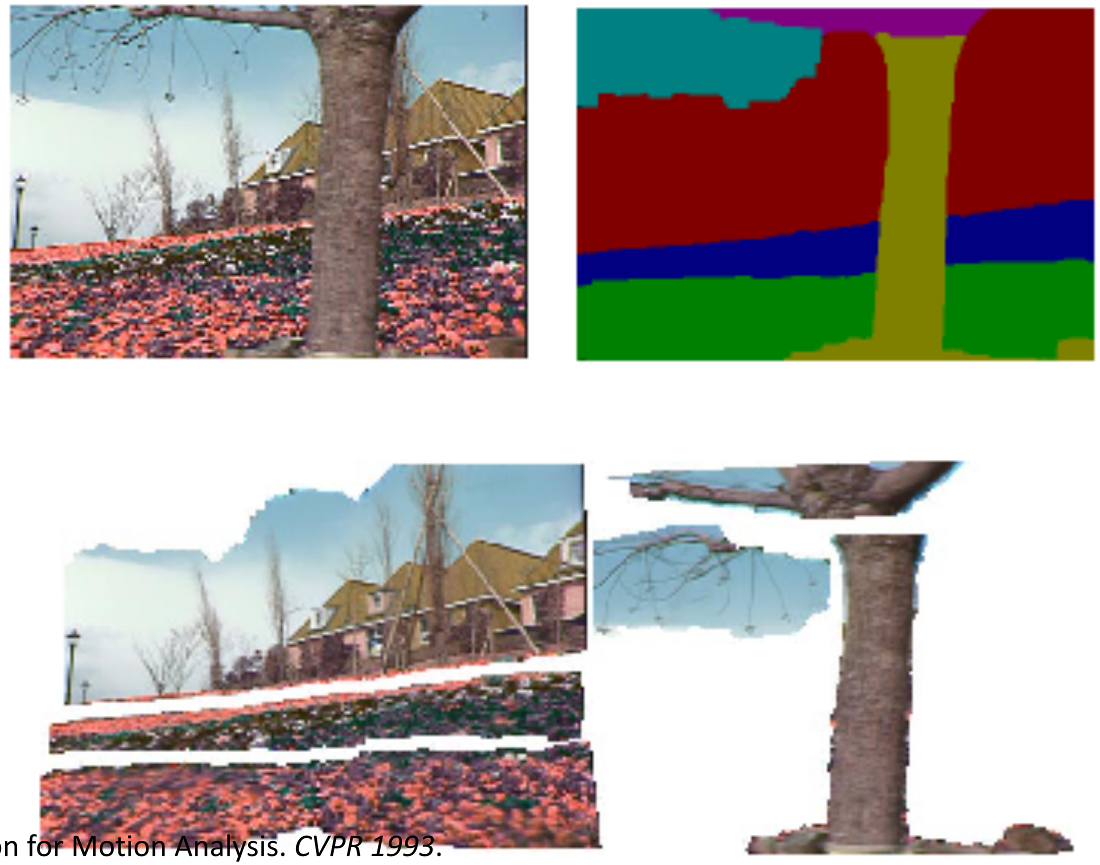
Segmentation using motion

- Use any of the segmentation algorithms: (k-means, Agglomerative clustering, mean shift)



Segmentation using motion

- Use any of the segmentation algorithms: (k-means, Agglomerative clustering, mean shift)

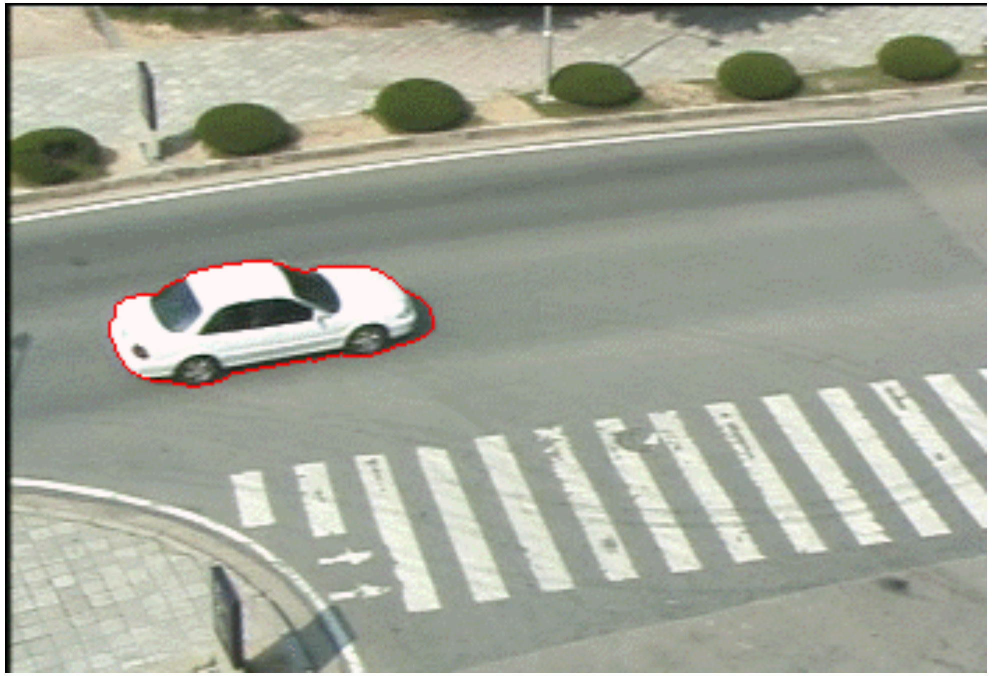


J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

Today’s agenda

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

Single object tracking



Multiple object tracking

Tracking with a fixed camera

Tracking with a fixed camera

Tracking with a moving camera

Challenges in Feature tracking

- 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 over time
- Points may appear or disappear.
  - need to be able to add/delete tracked points.

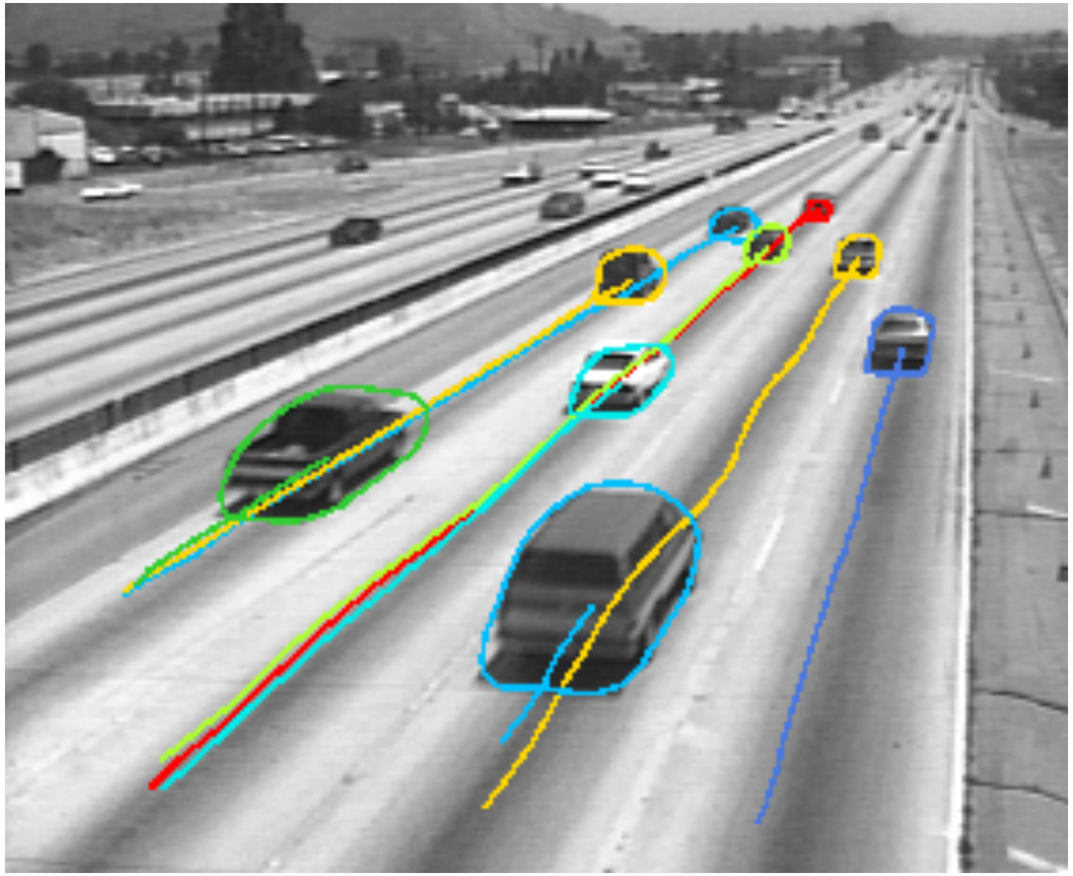
What are good features to track?

- Intuitively, we want to avoid smooth regions and edges. But is there a more is principled way to define good features?

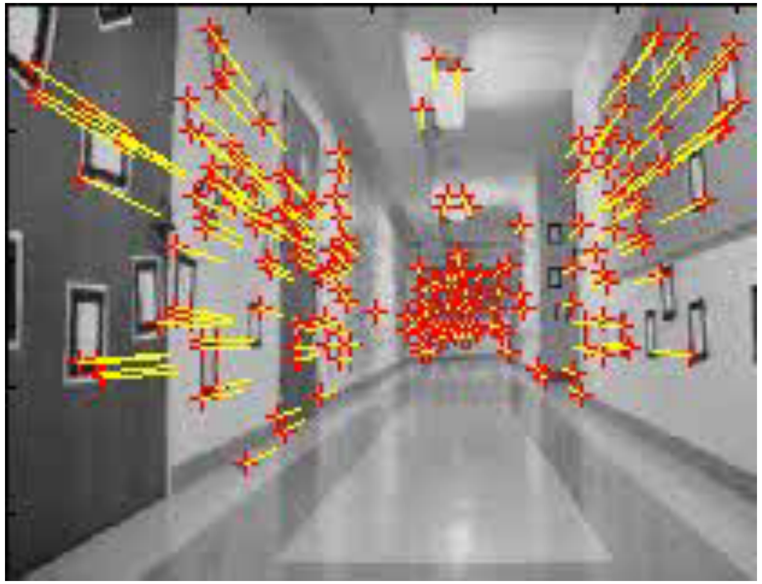
- What kinds of image regions can we detect easily and consistently?
  - SIFT blobs!
  - Harris corners!

Optical flow can help track features

Once we have the features we want to track, lucas-kanade or other optical flow algorithm can help track those features

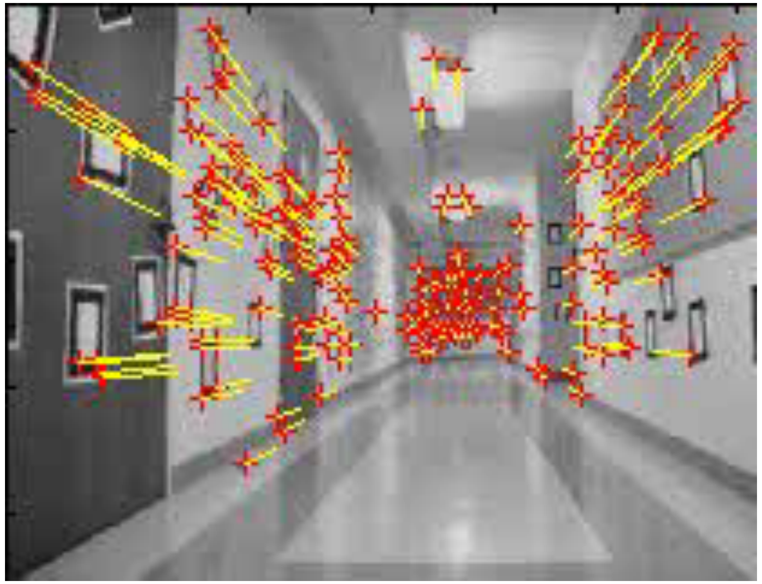


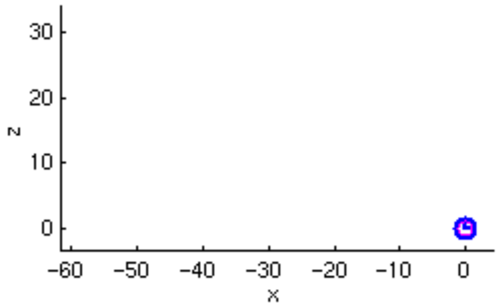
Feature-tracking/Mapping



Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology

Feature-tracking/Mapping





Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology

Simple KLT tracker

1. Find a good point to track (harris corner)
2. For each Harris corner compute optical flow (translation or affine) between consecutive frames.
3. Link motion vectors in successive frames to get a track for each Harris point
4. Introduce new Harris points by applying Harris detector at every m (10 or 15) frames
5. Track new and old Harris points using steps 1-3

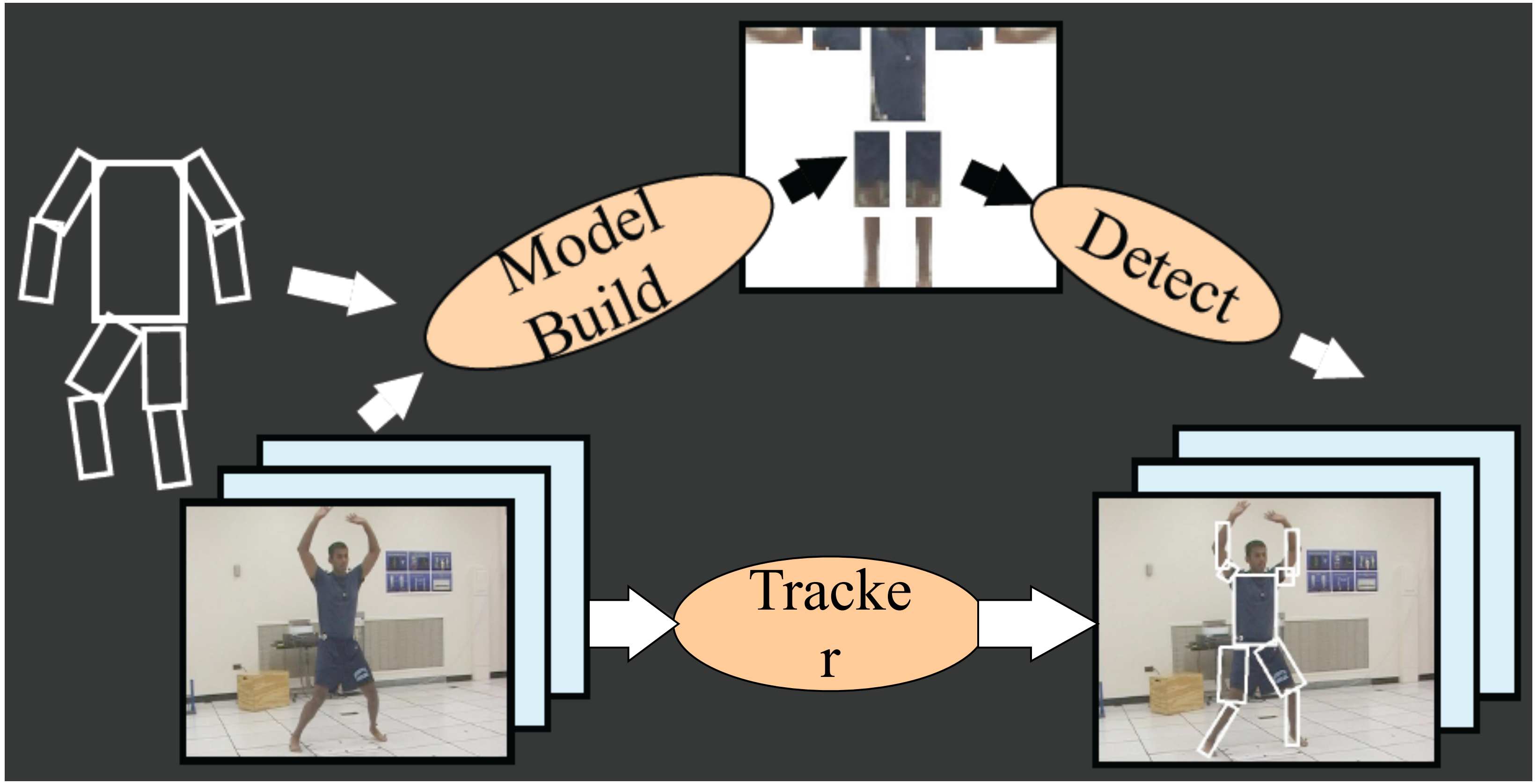
What we will learn today?

- Optical flow
- Lucas-Kanade method
- Pyramids for large motion
- Horn-Schunk method
- Segmentation from motion
- Tracking
- Applications

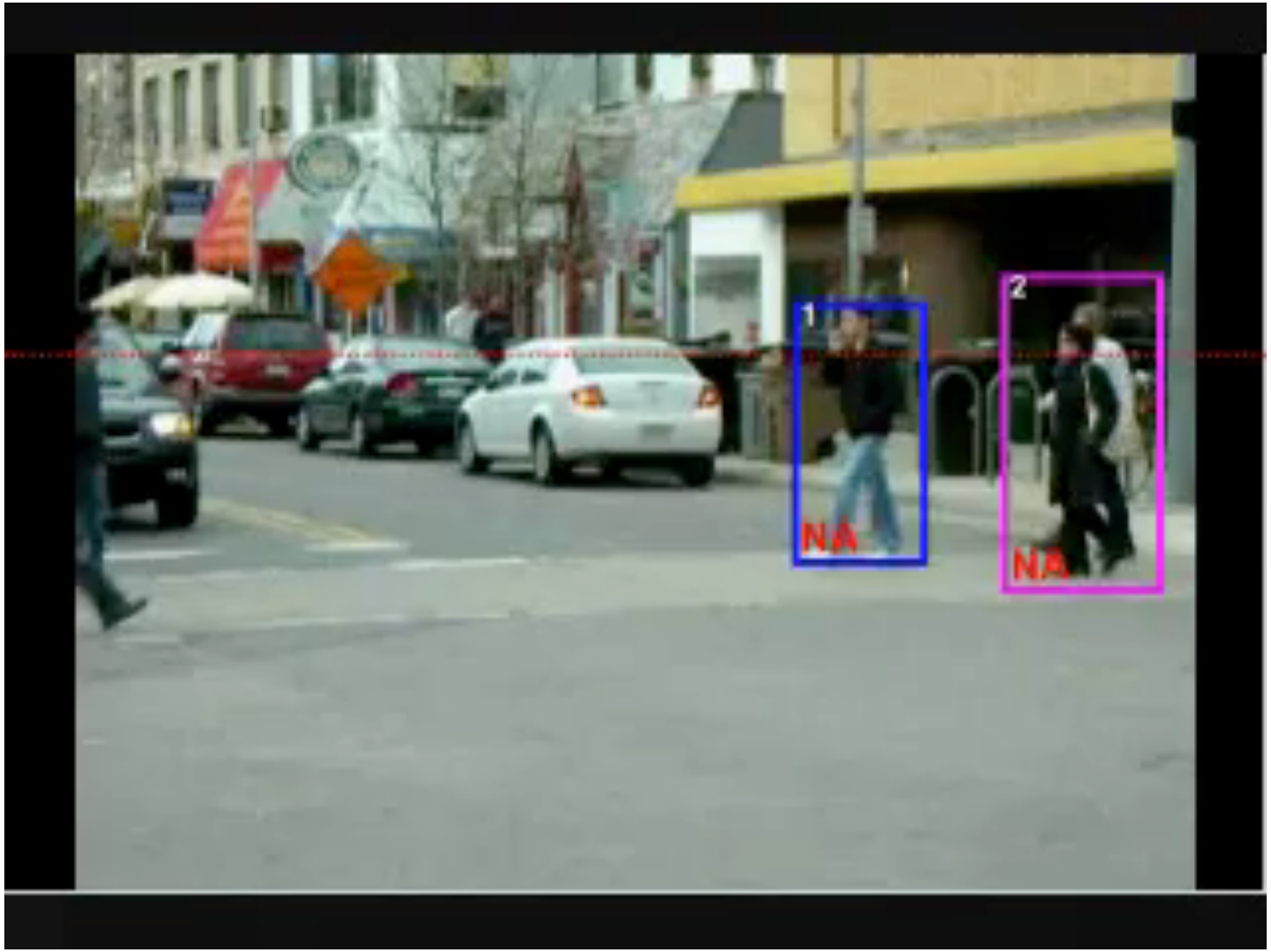
Uses of motion

- Segmenting objects based on motion cues
- Tracking objects
- Learning dynamical models
- Improving video quality
    - Motion stabilization
    - Super resolution
- Recognizing events and activities

Recognizing events and activities



D. Ramanan, D. Forsyth, and A. Zisserman. Tracking People by Learning their Appearance. PAMI 2007.



W. Choi, K. Shahid, S. Savarese, "What are they doing? : Collective Activity Classification Using Spatio-Temporal Relationship Among People", 9th International Workshop on Visual Surveillance (VSWS09) in conjuction with ICCV 09

Today's agenda

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Segmentation from motion
- Tracking
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

Next time

Geometry and cameras