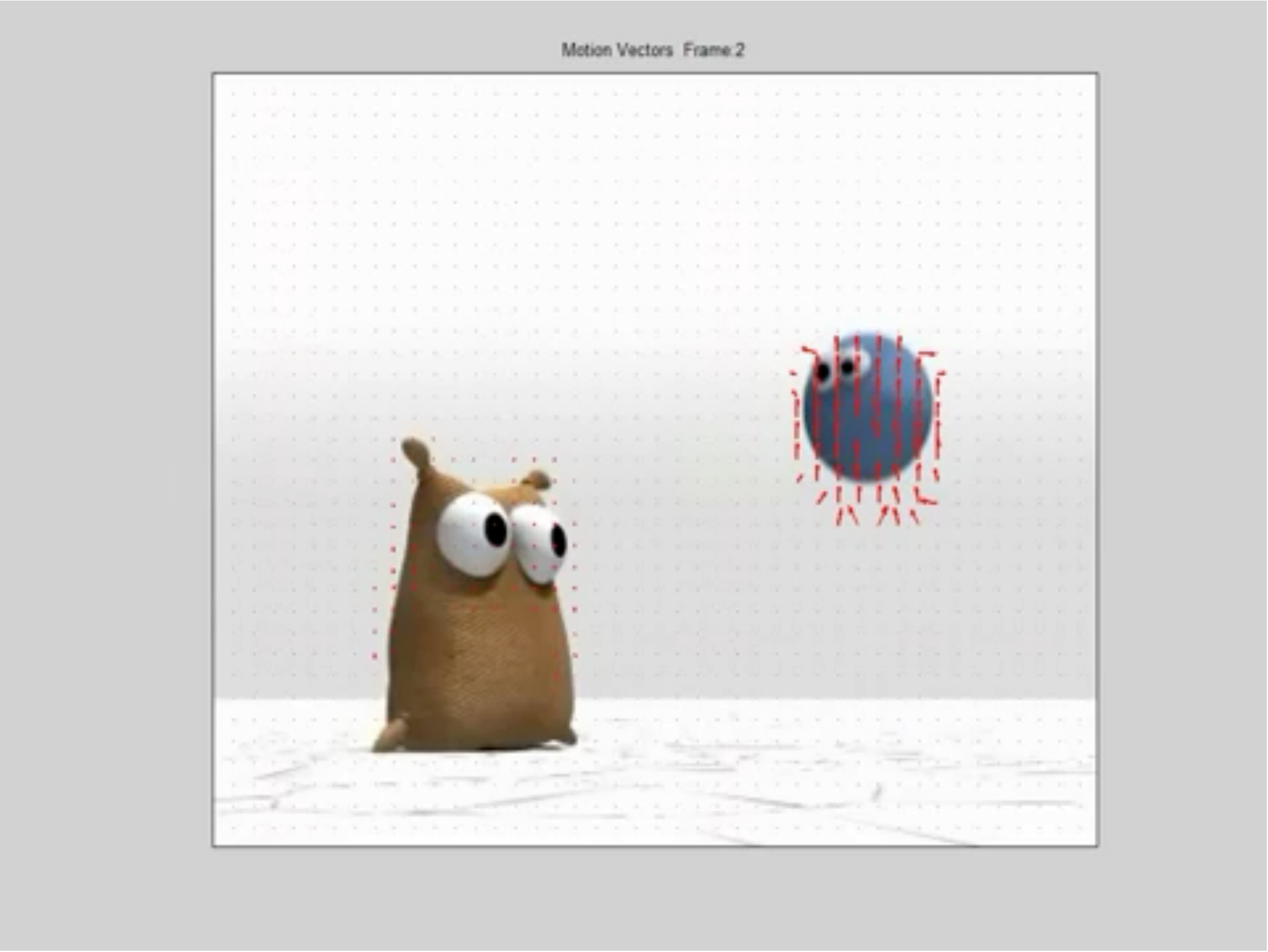
CSE 152: Computer Vision

Hao Su

Lecture 15: Motion and Optical Flow



https://www.youtube.com/watch?v=BDtvjKMl30w

https://www.youtube.com/watch?v=5rR_9YIcg_s

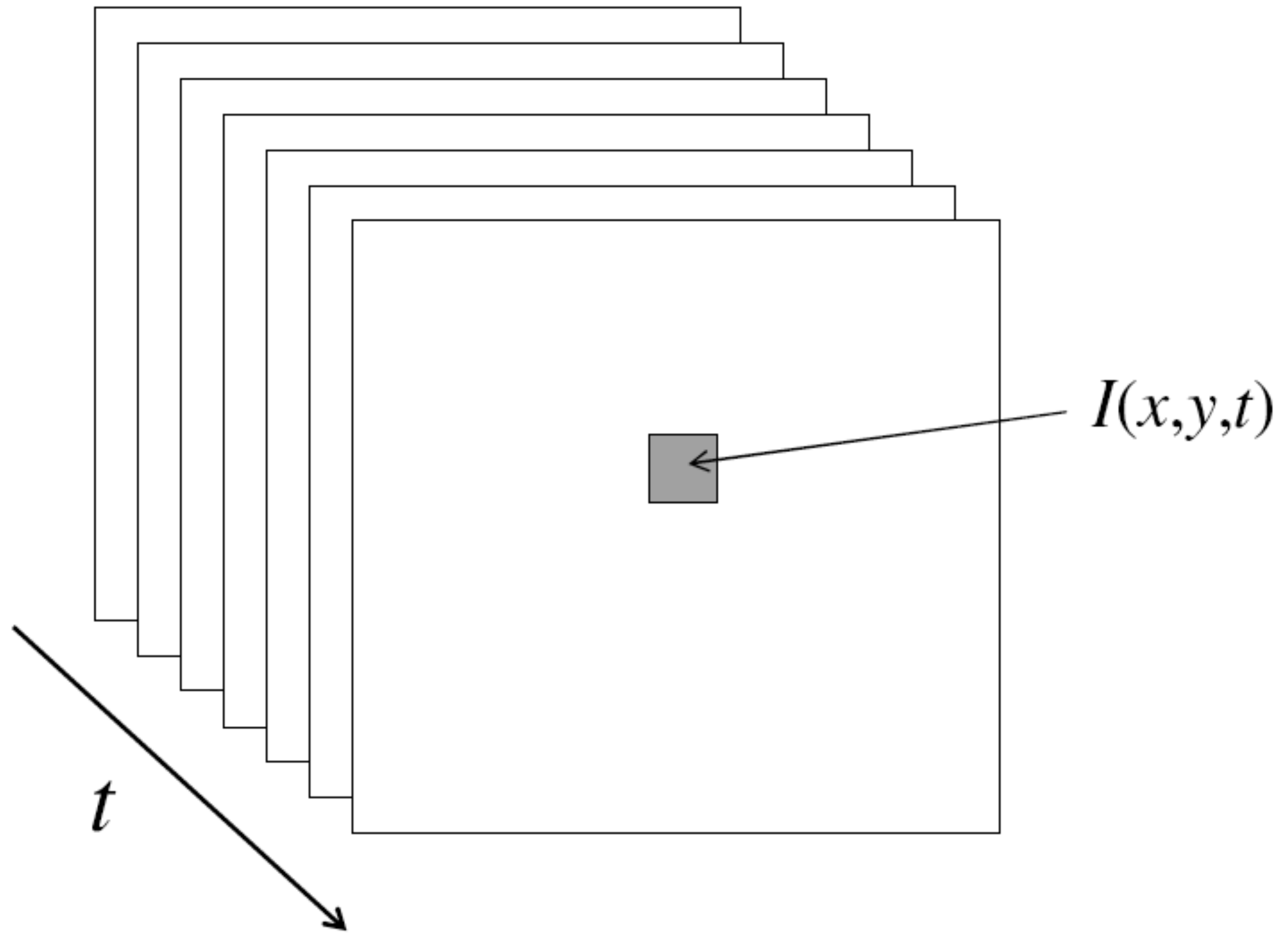
Plan

- Motion Field
- Patch-based / Direct Motion Estimation

- External Resource:
  - Mubarak Shah's lecture on optical flow
    - http://www.youtube.com/watch?v=5VyLAH8BhF8

Video

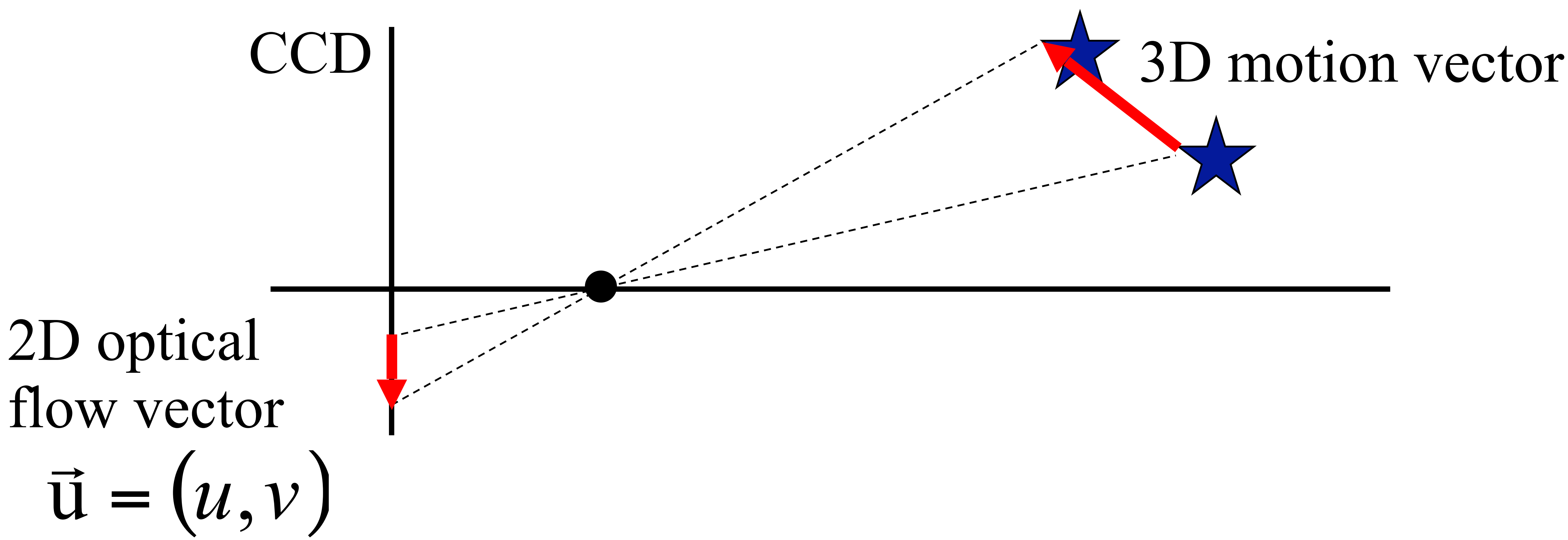
- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



Slide adapted from K. Grauman.

Motion Field & Optical Flow Field

- Motion Field = Real world 3D motion
- Optical Flow Field = Projection of the motion field onto the 2d image



Motion field + camera motion

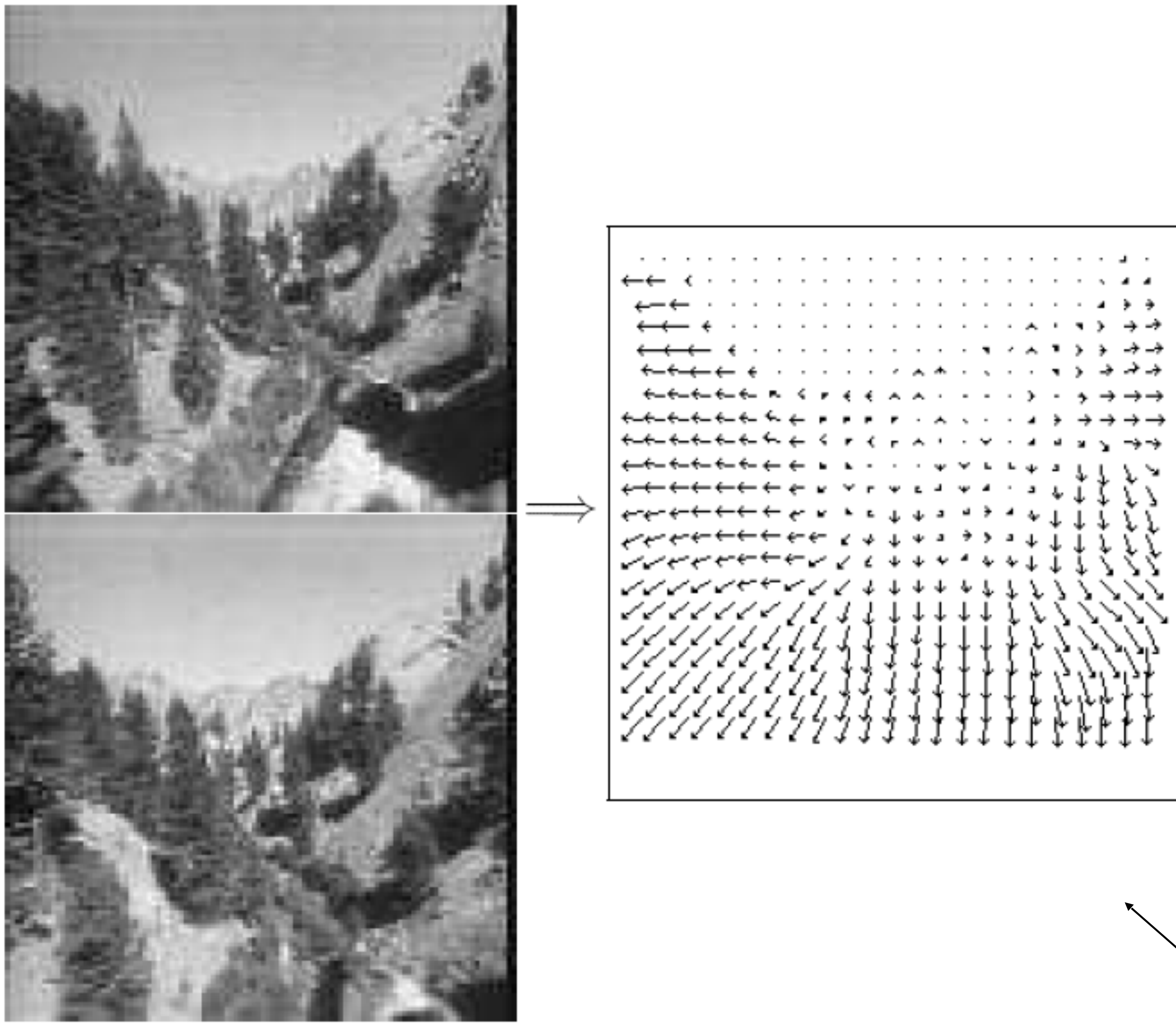


Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

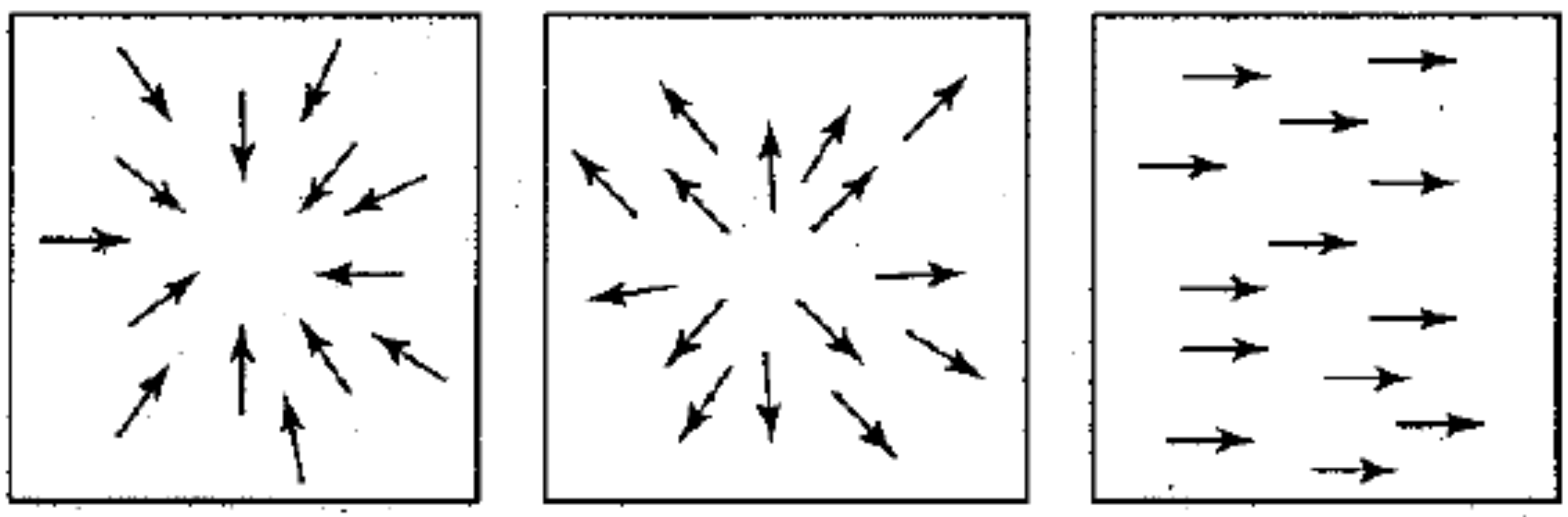
Length of flow vectors inversely proportional to depth Z of 3d point

points closer to the camera move more quickly across the image plane

Figure from Michael Black, Ph.D. Thesis

Slide adapted from K. Grauman.

Motion field + camera motion



Zoom out **Zoom in** **Pan right to left**

Slide adapted from Savarese and Grauman.

Motion estimation techniques

- Feature-based methods
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small

Slide adapted from K. Grauman.

What we will learn today?

- Optical flow
- Lucas-Kanade method
- Applications

Optical Flow

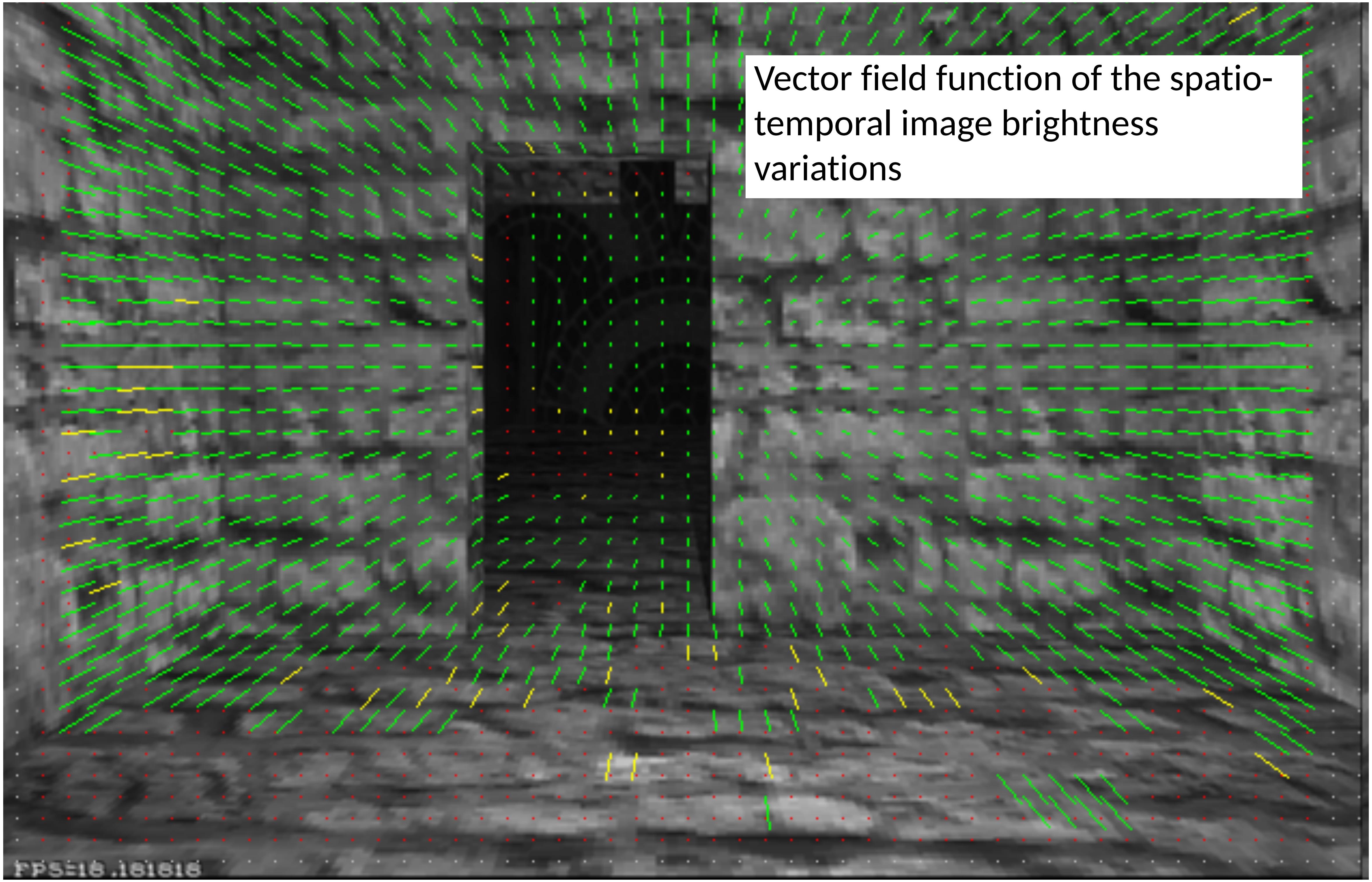
- Definition: optical flow is the apparent motion of brightness patterns in the image

- Note: apparent motion can be caused by lighting changes without any actual motion
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

GOAL: Recover image motion at each pixel from optical flow

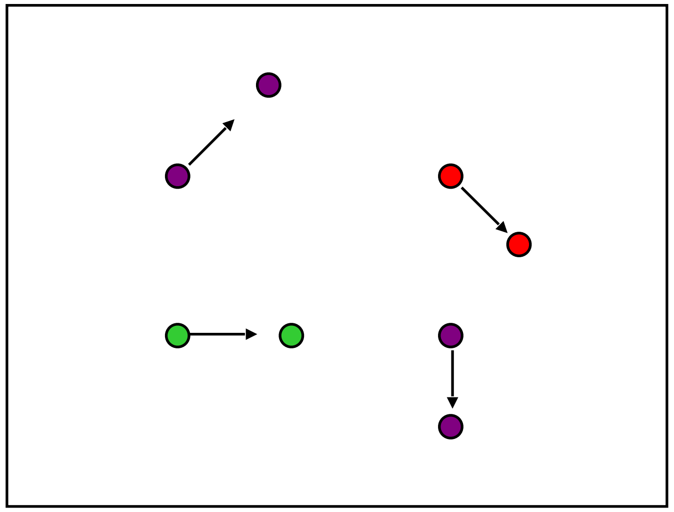
Source: Silvio Savarese

Optical Flow

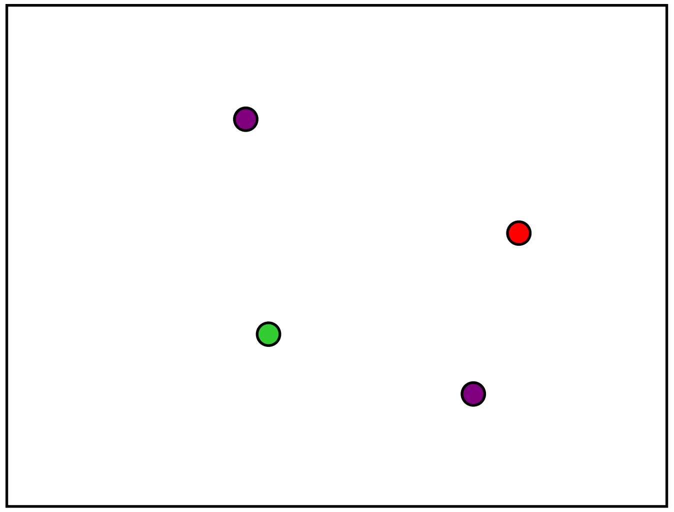


Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

Estimating Optical Flow



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



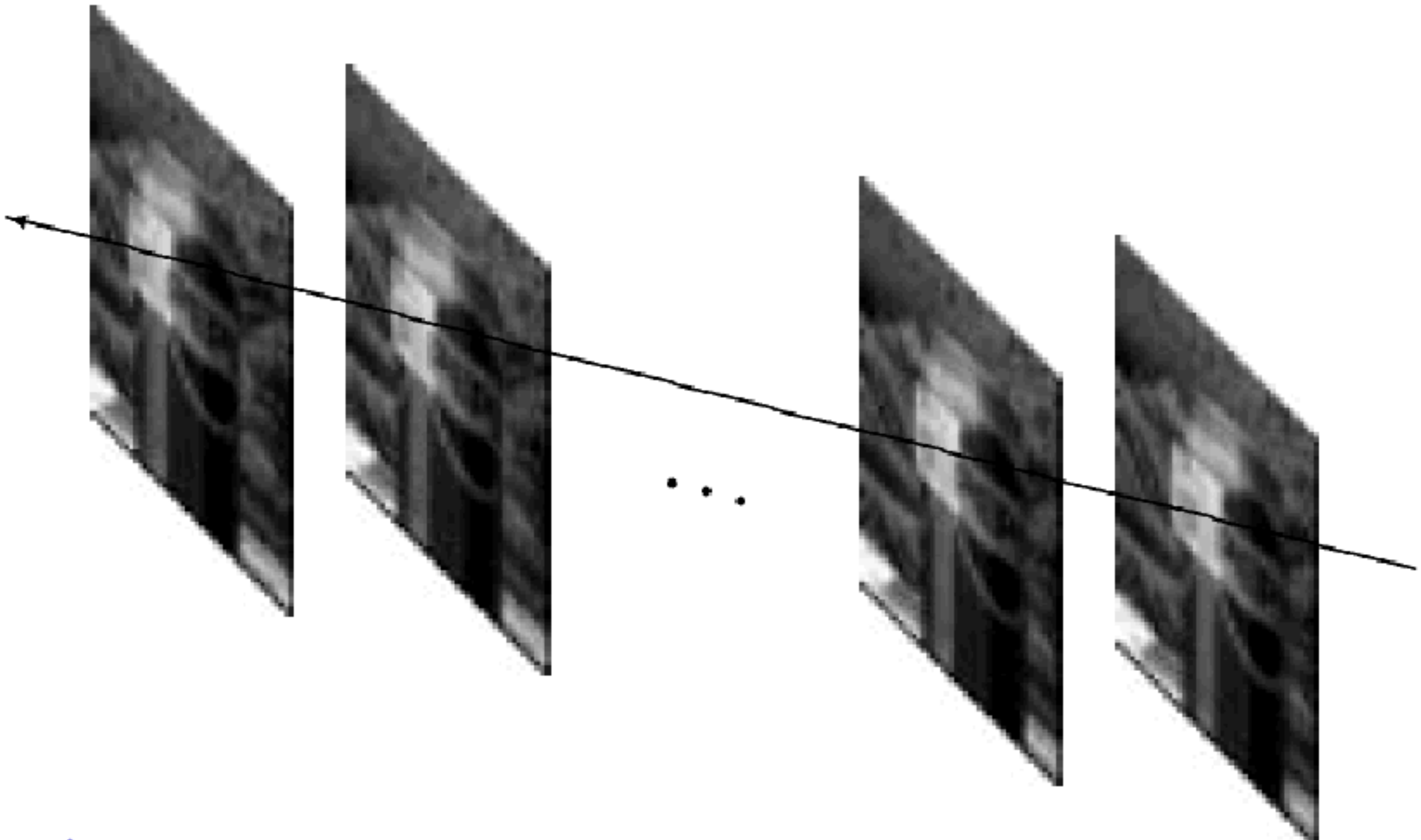
$I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field u(x,y), v(x,y) between them

- 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

Source: Silvio Savarese

Key Assumptions: Small Motions

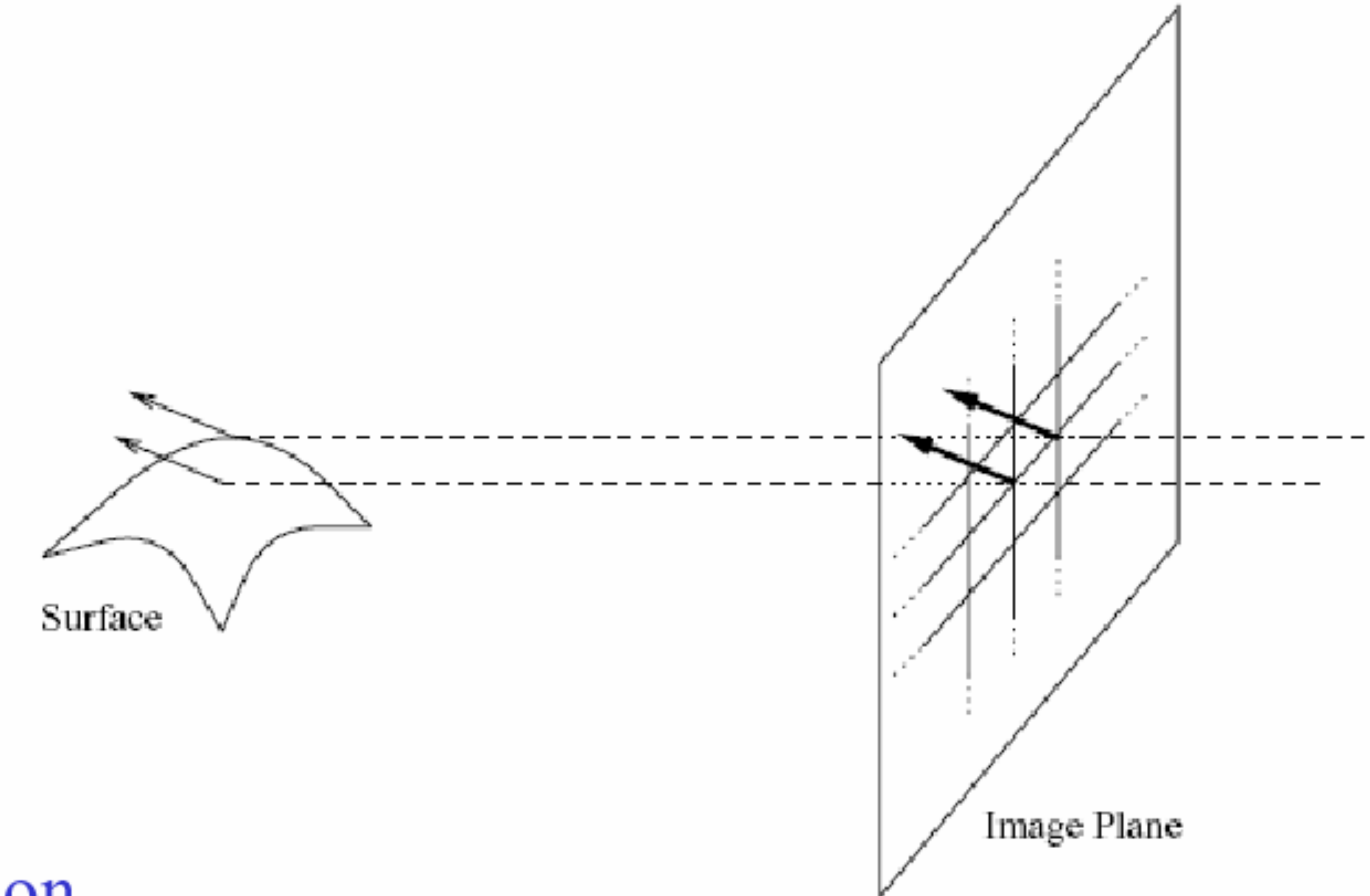


Assumption:

The image motion of a surface patch changes gradually over time.

* Slide from Michael Black, CS143 2003

Key Assumptions: Spatial Coherence

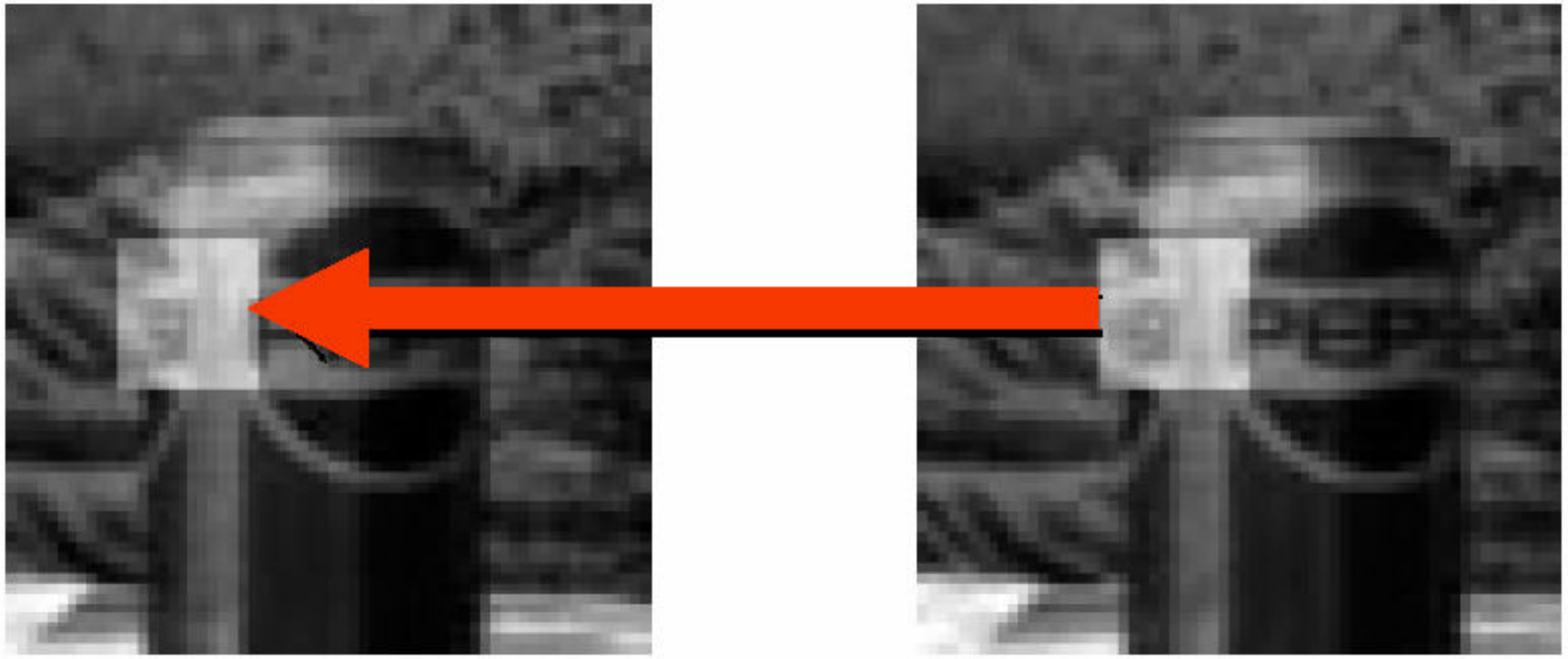


Assumption

* Neighboring points in the scene typically belong to the same surface and hence typically have similar motions.
* Since they also project to nearby points in the image, we expect spatial coherence in image flow.

* Slide from Michael Black, CS143 2003

Key Assumptions: Brightness Constancy



Assumption

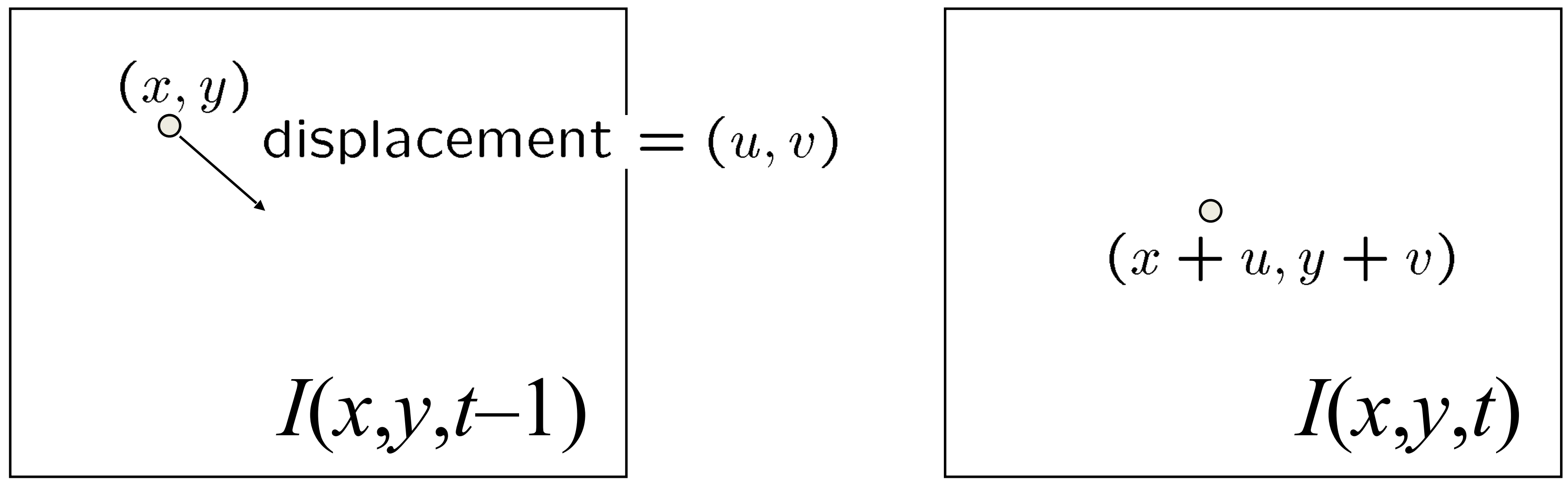
Image measurements (e.g. brightness) in a small region remain the same although their location may change.

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

(assumption)

* Slide from Michael Black, CS143 2003

The Brightness Constancy Constraint



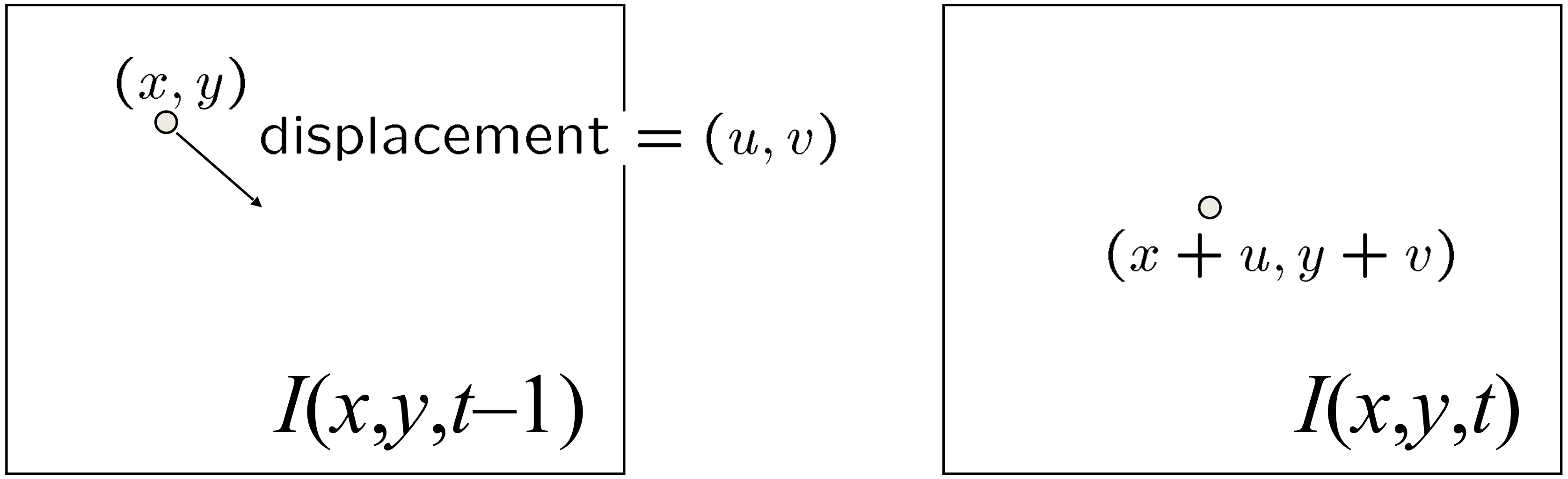
- Brightness Constancy Equation:

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

Linearizing the right side using Taylor expansion:

Source: Silvio Savarese

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:

Image derivative along x

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

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

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

Source: Silvio Savarese

Filters used to find the derivatives

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{first image} \qquad \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{first image} \qquad \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \text{first image}$$

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{second image} \qquad \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \text{second image} \qquad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \text{second image}$$

$$I_x \qquad\qquad I_y \qquad\qquad I_t$$

The brightness constancy constraint

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

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

- How many equations and unknowns per pixel?
  - One equation (this is a scalar equation!), two unknowns (u,v)

Source: Silvio Savarese

The brightness constancy constraint

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

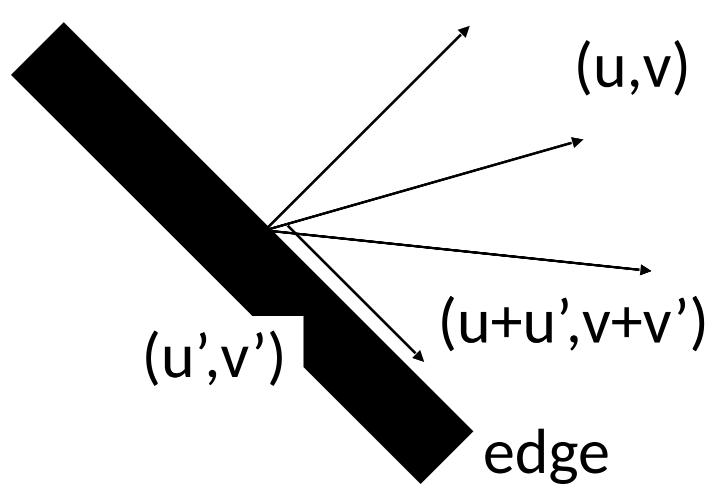
$$\nabla I \cdot \left[u \ v\right]^T + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation (this is a scalar equation!), two unknowns (u,v)

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

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



Source: Silvio Savarese

What we will learn today?

- Optical flow
- Lucas-Kanade method
- Applications

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…

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

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

Source: Silvio Savarese

Lucas-Kanade flow

- Over-constrained linear system:

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

Source: Silvio Savarese

Lucas-Kanade flow

- Over-constrained linear system

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

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

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

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

Source: Silvio Savarese

Conditions for solvability

– Optimal (u, v) satisfies Lucas-Kanade equation

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

When is This Solvable?

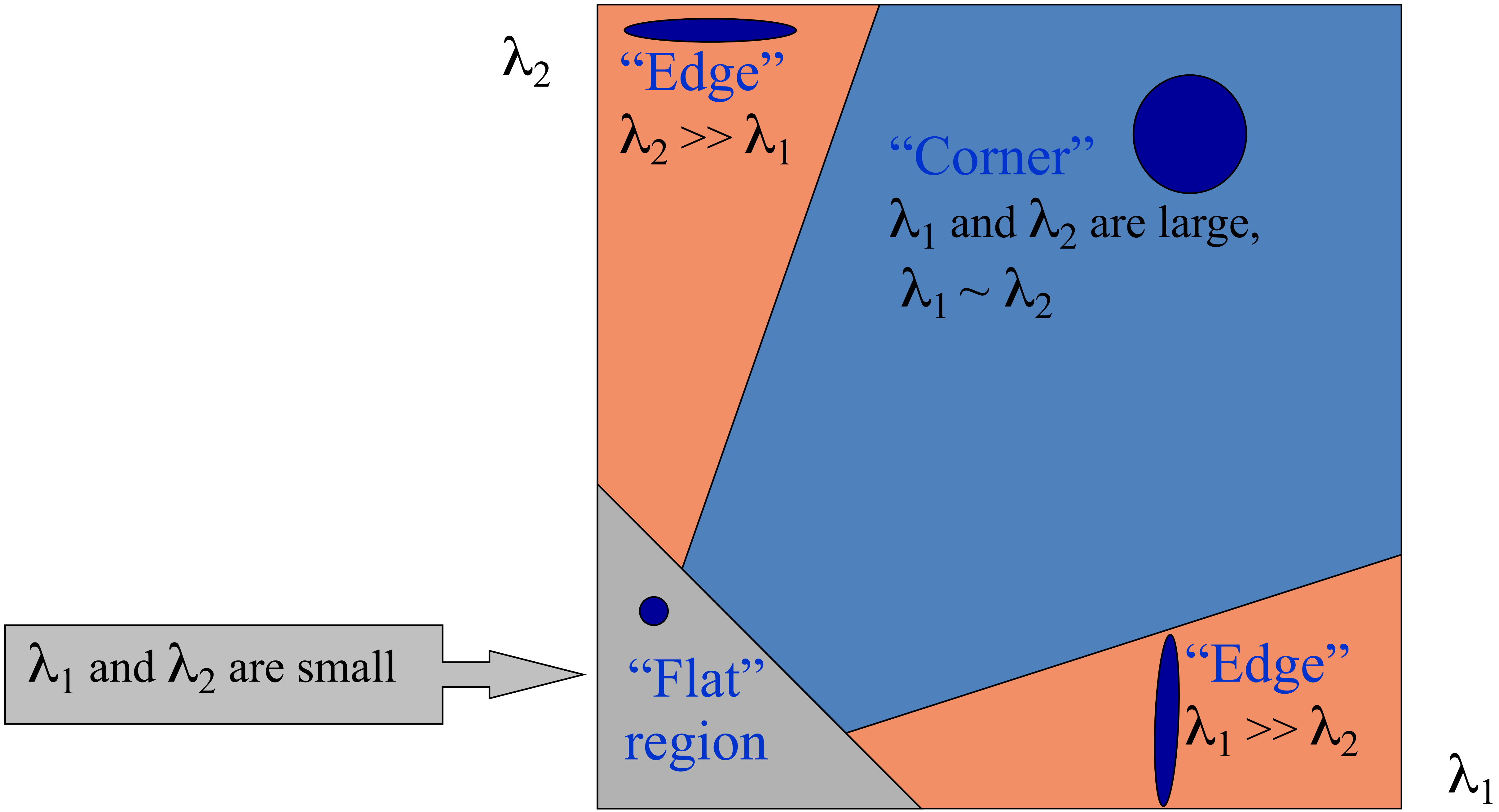
- $A^T A$ should be invertible
- $A^T A$ should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of $A^T A$ should not be too small
- $A^T A$ should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Does this remind anything to you?

Source: Silvio Savarese

Interpreting the eigenvalues

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



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

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?

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Errors in Lukas-Kanade

Examples:

1. A uniform rotating sphere
   – nothing seems to move, yet it is rotating

2. Changing directions or intensities of lighting can make things seem to move
   – for example, if the specular highlight on a rotating sphere moves.

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

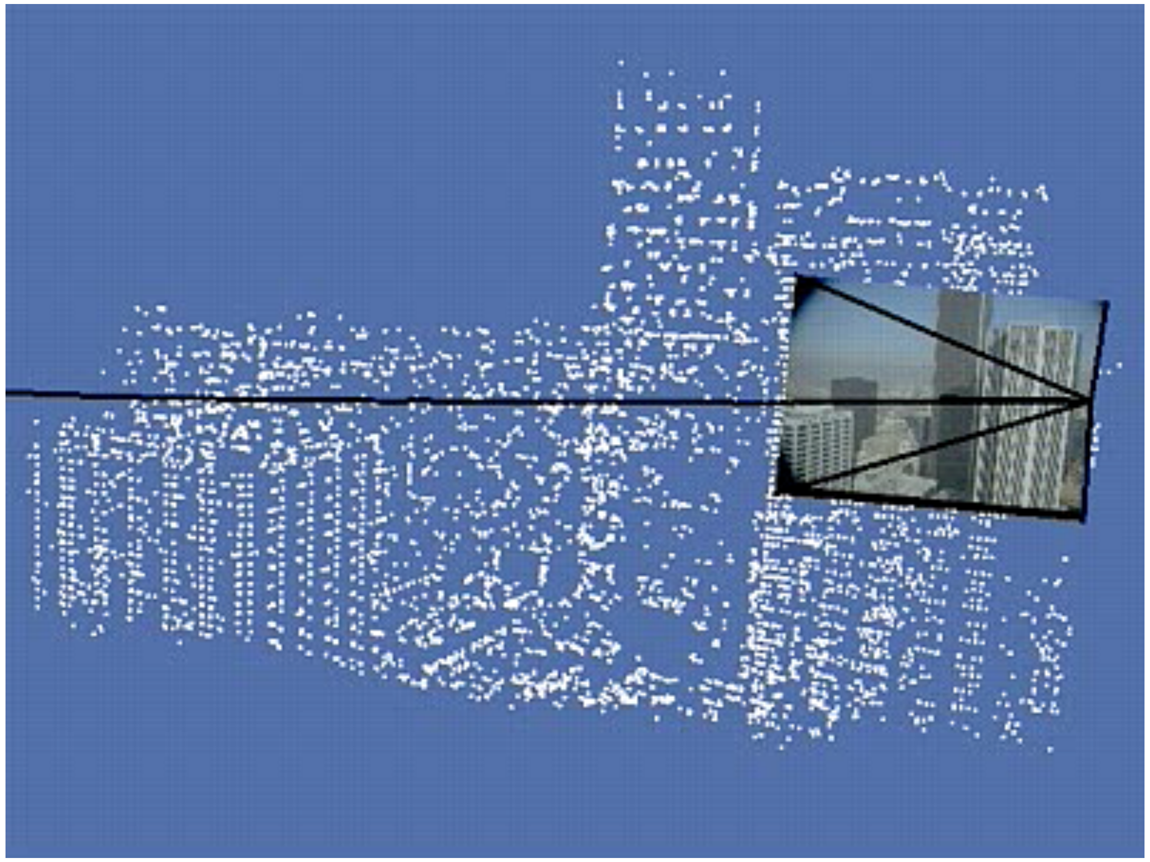
What we will learn today?

- Optical flow
- Lucas-Kanade method
- Applications

Uses of motion

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

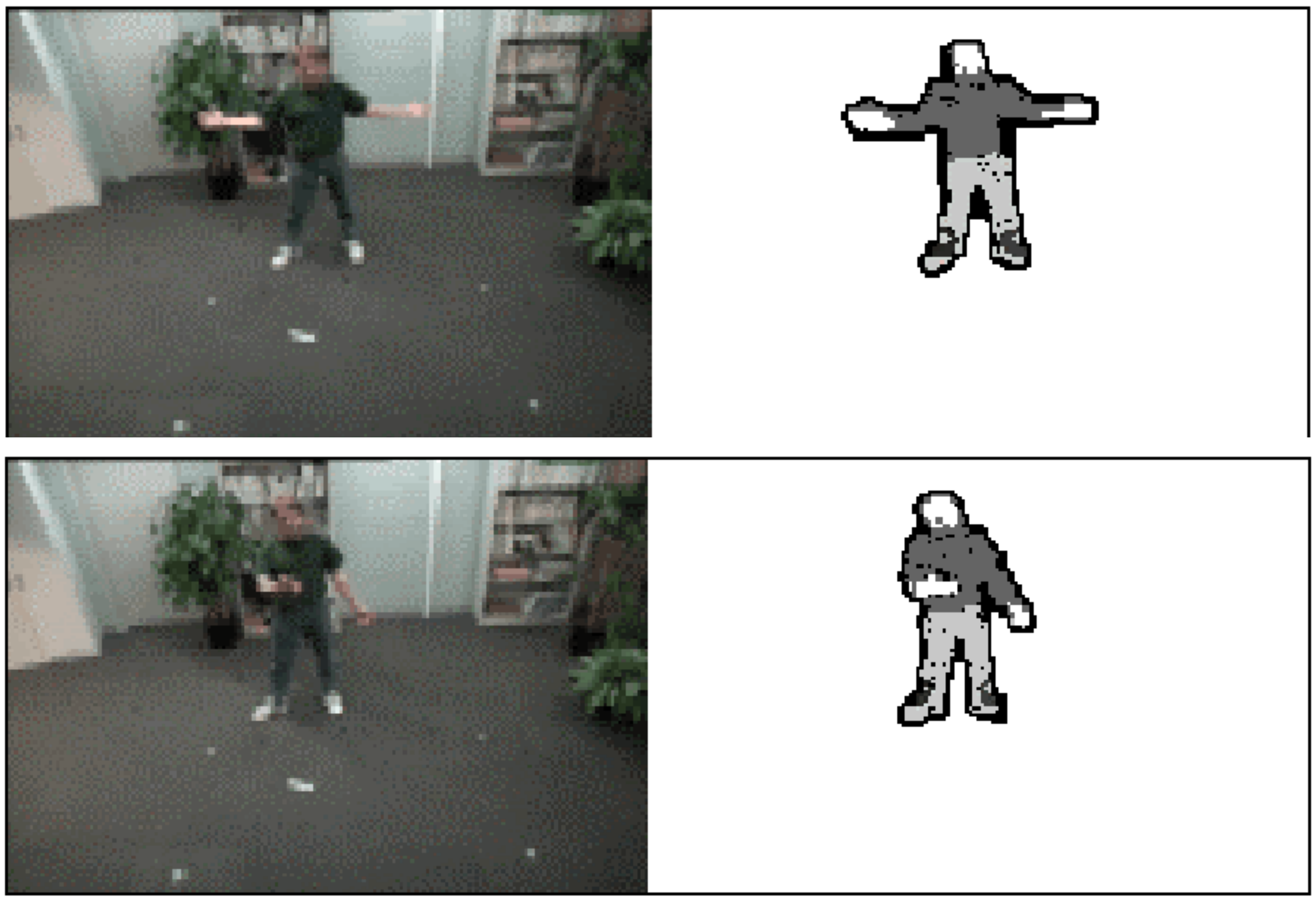
Estimating 3D structure



Source: Silvio Savarese

Segmenting objects based on motion cues

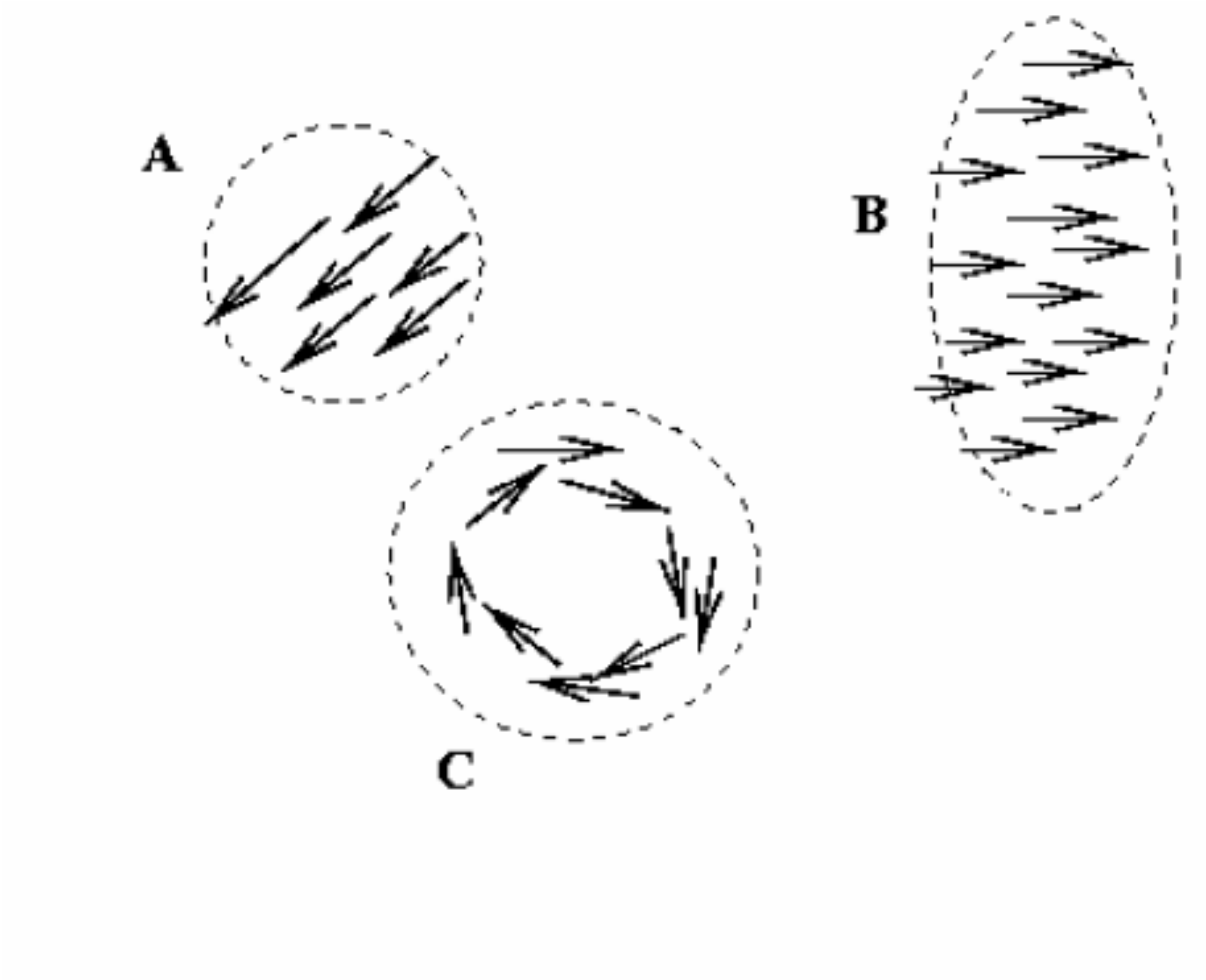
- Background subtraction
  - A static camera is observing a scene
  - Goal: separate the static background from the moving foreground

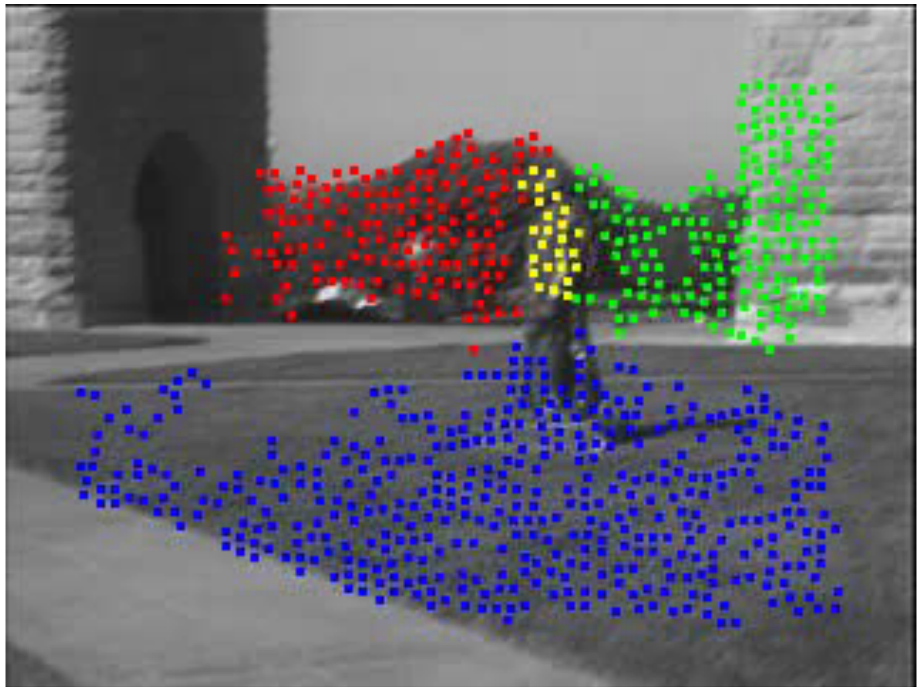


Source: Silvio Savarese

Segmenting objects based on motion cues

- Motion segmentation
  - Segment the video into multiple coherently moving objects

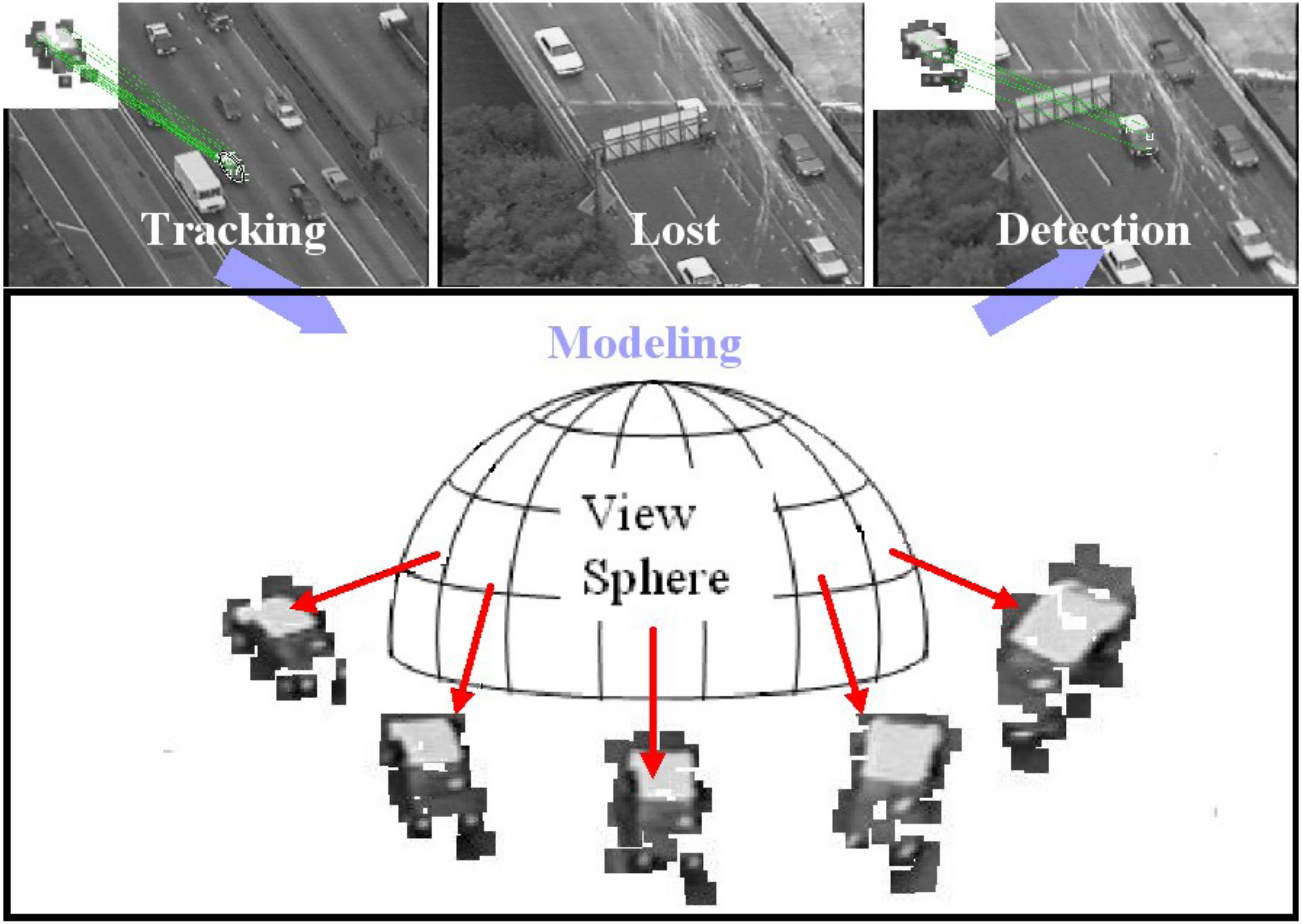




S. J. Pundlik and S. T. Birchfield, Motion Segmentation at Any Speed, Proceedings of the British Machine Vision Conference (BMVC) 2006

Source: Silvio Savarese

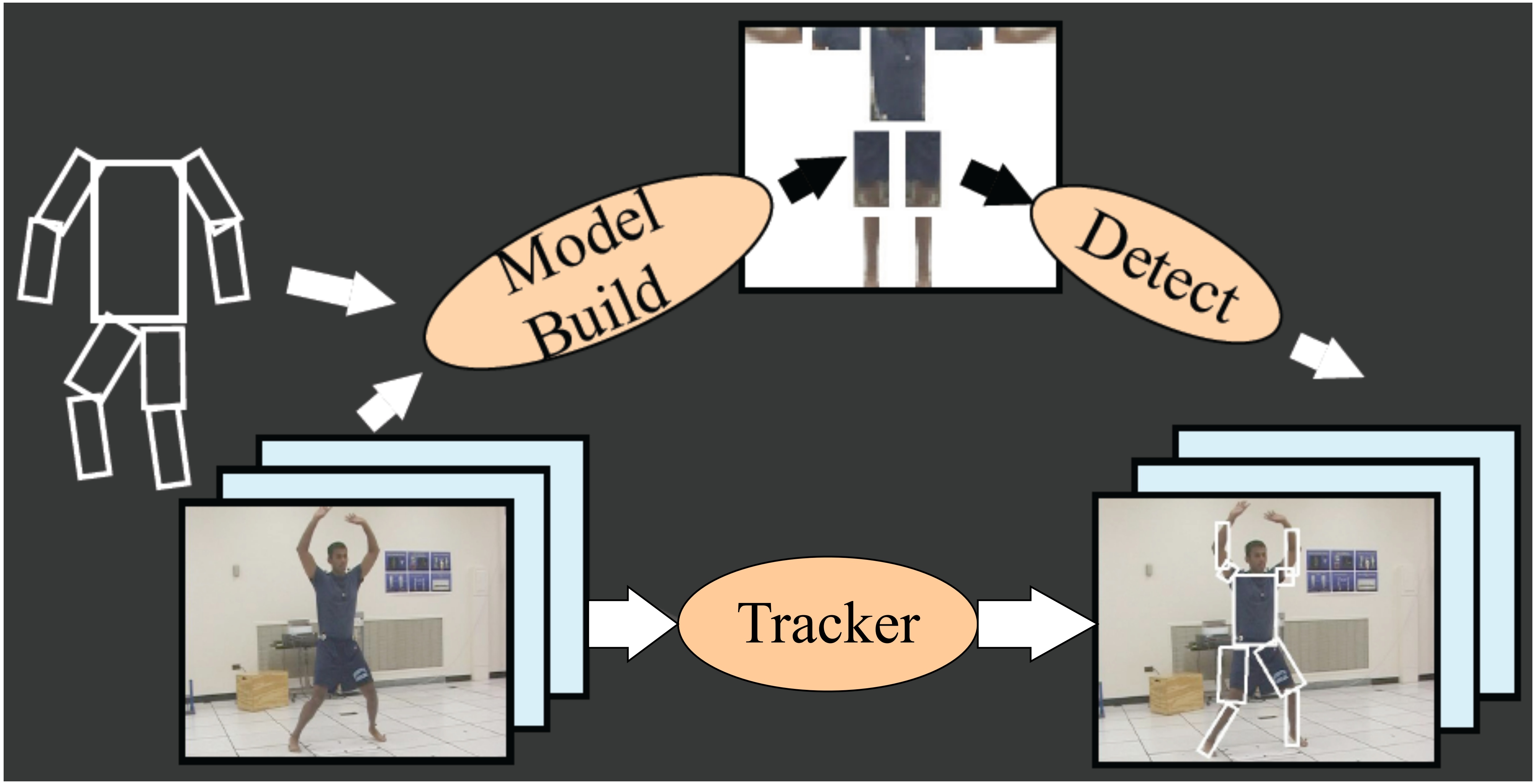
Tracking objects



Z.Yin and R.Collins, "On-the-fly Object Modeling while Tracking," IEEE Computer Vision and Pattern Recognition (CVPR '07), Minneapolis, MN, June 2007.

Source: Silvio Savarese

Recognizing events and activities

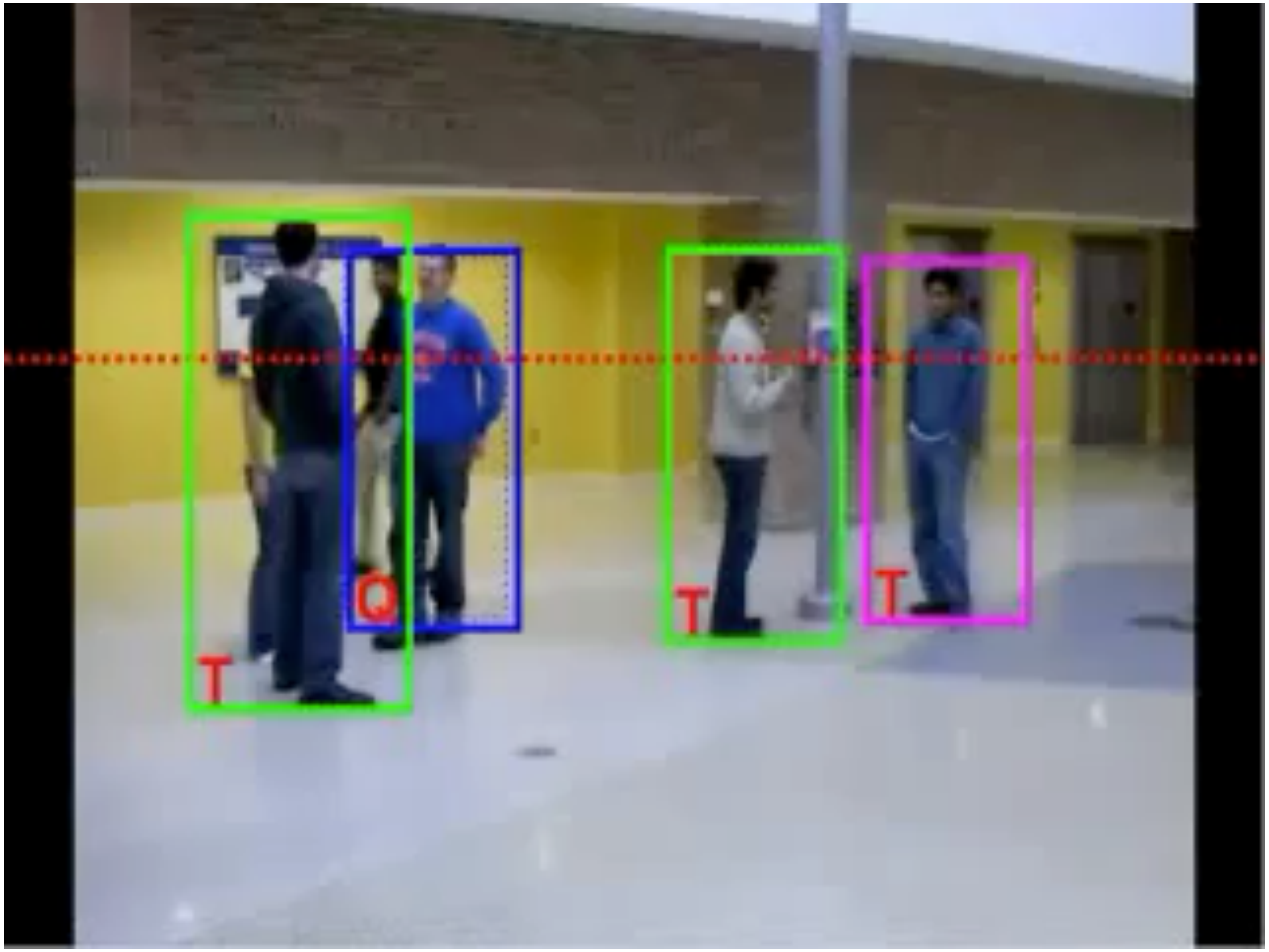


D. Ramanan, D. Forsyth, and A. Zisserman. [Tracking People by Learning their Appearance](). PAMI 2007.

Source: Silvio Savarese

Recognizing events and activities

Crossing – Talking – Queuing – Dancing – jogging



W. Choi & K. Shahid & S. Savarese WMC 2010

Source: Silvio Savarese

Optical flow without motion!