Lecture 13: Tracking motion features – optical flow

Professor Fei-Fei Li

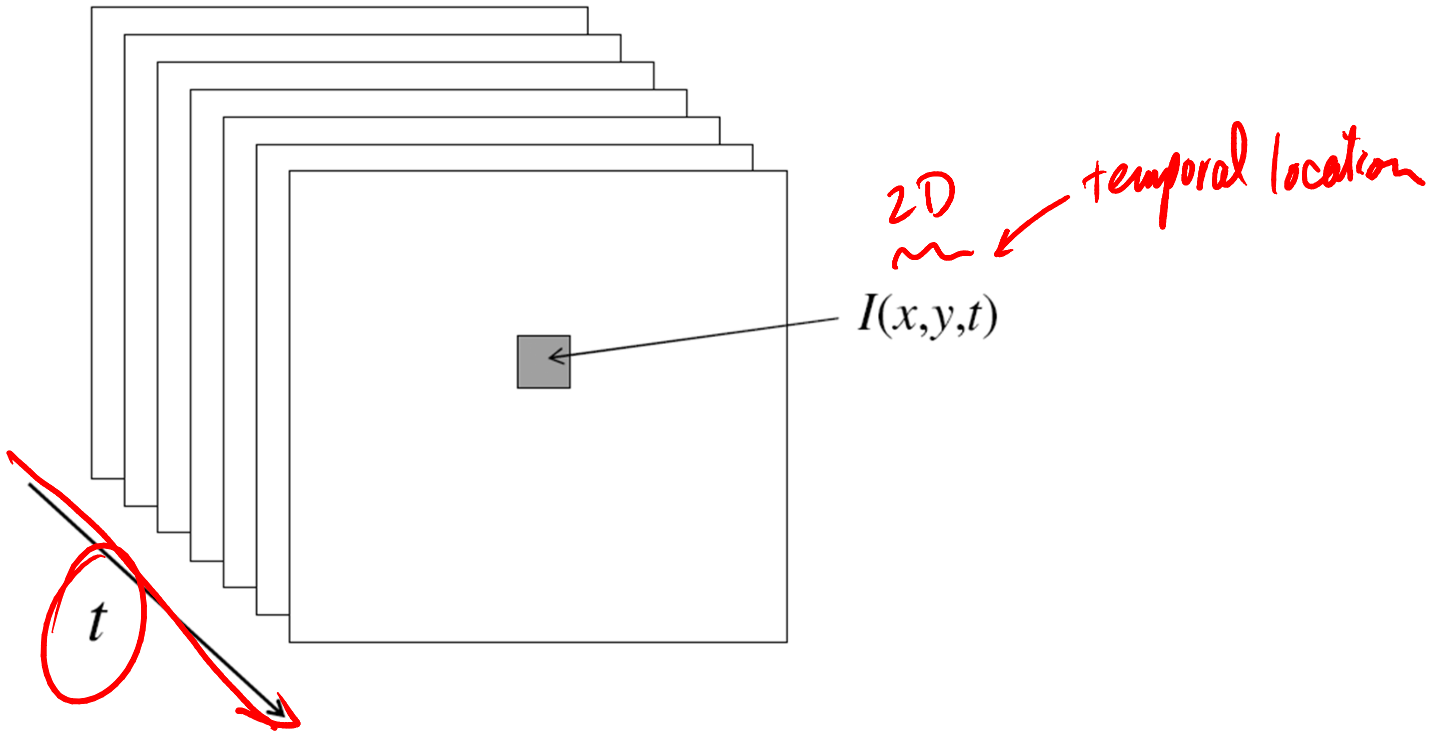
Stanford Vision Lab

What we will learn today?

- Introduction
- Optical flow
- Feature tracking
- Applications
- (Problem Set 3 (Q1))

From images to videos

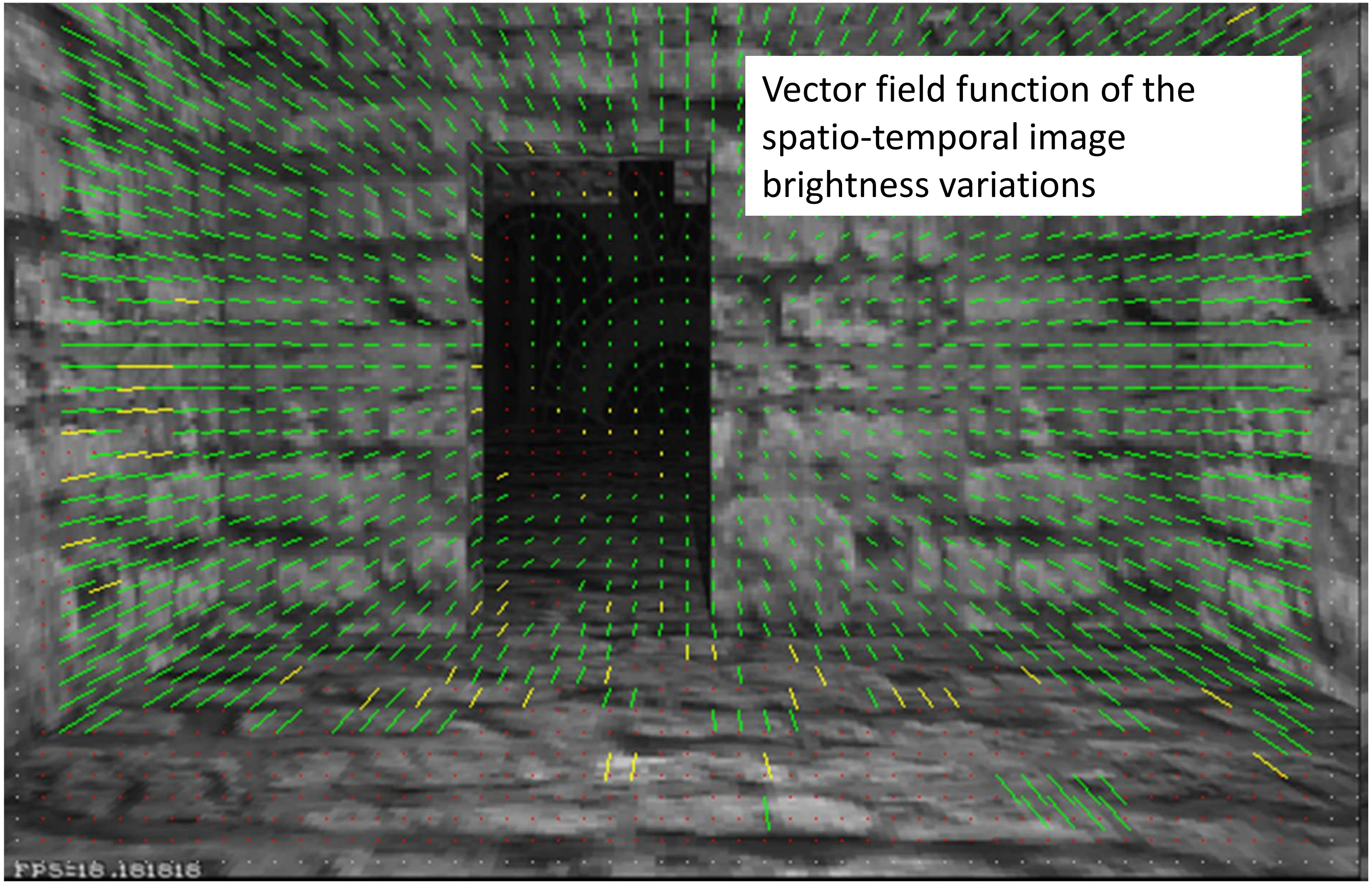
- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



Motion estimation techniques

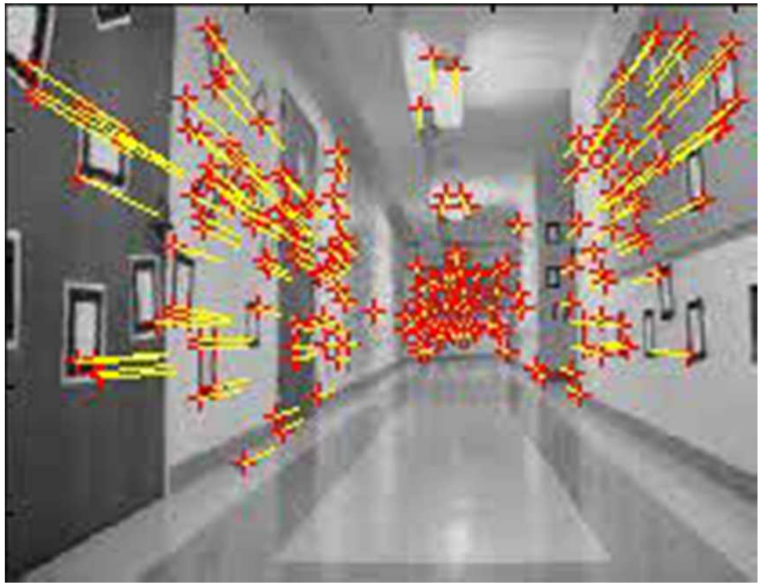
- Optical flow
  - Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)
- Feature-tracking
  - Extract visual features (corners, textured areas) and “track” them over multiple frames

Optical flow



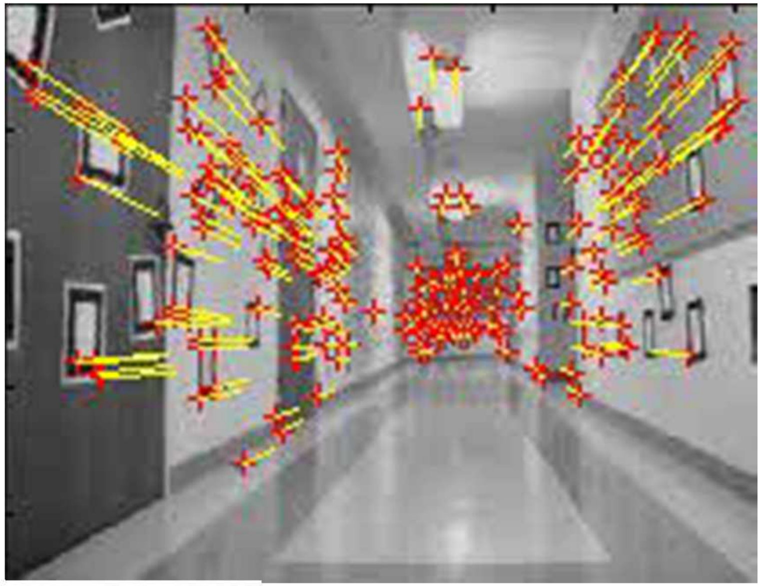
Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

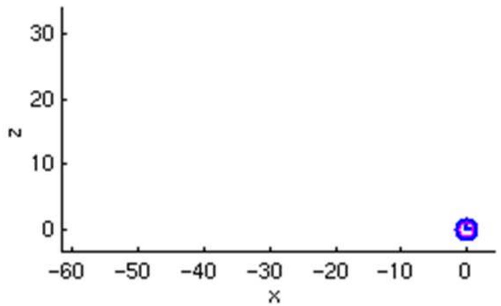
Feature-tracking



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

Feature-tracking





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

Optical flow

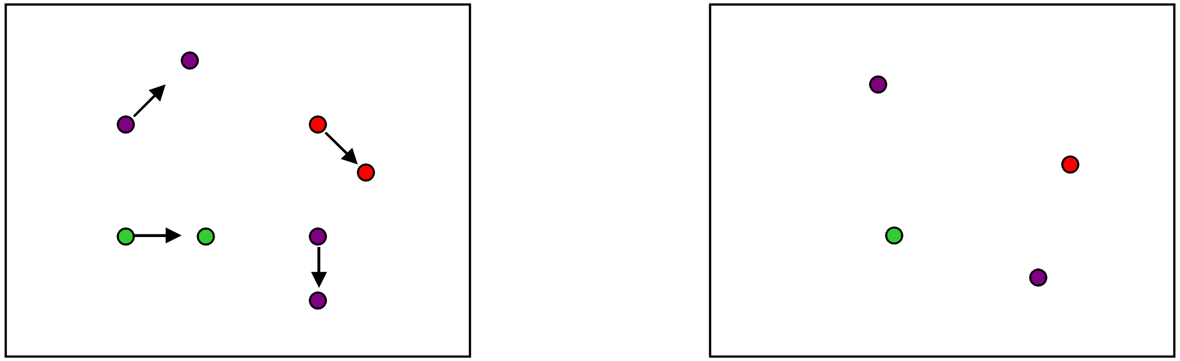
- Definition: optical flow is the *apparent* motion of brightness patterns in the image
- Note: apparent motion can be caused by lighting changes without any actual motion
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

GOAL: Recover image motion at each pixel from optical flow

Source: Silvio Savarese

Estimating optical flow

time (t-1)



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

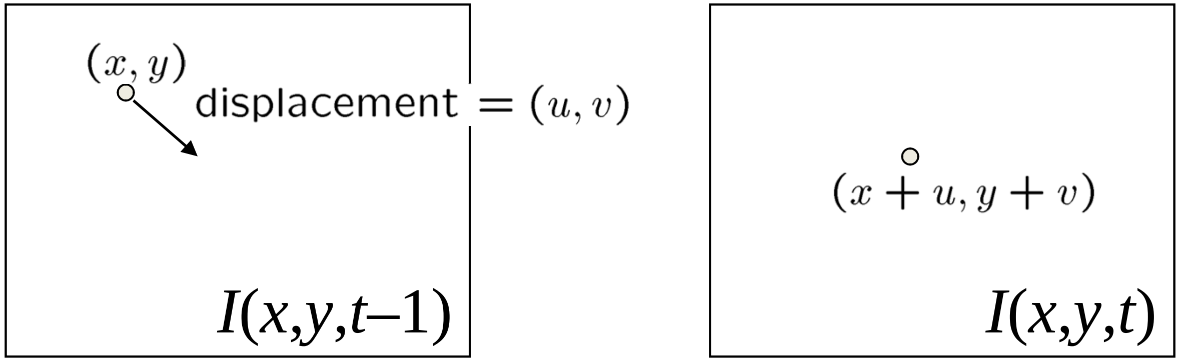
- Given two subsequent frames, estimate the apparent motion field u(x,y), v(x,y) between them

motion

- 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

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

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

$$0 = I(x+u, y+u, t) - I(x, y, t-1) = I_x \cdot u(x, y) + I_y \cdot v(x, y) + I_t$$

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

Source: Silvio Savarese

The brightness constancy constraint

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

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

- 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

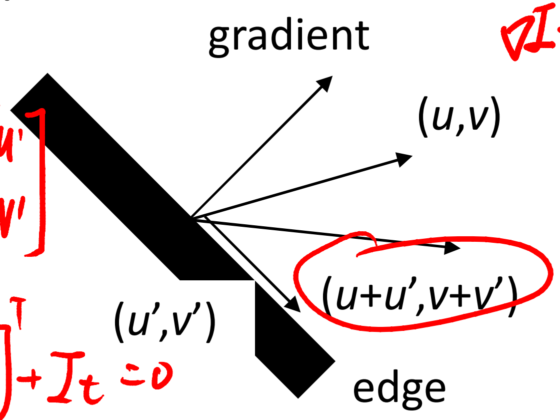
$\nabla I \cdot [u, v]^T + I_t = 0$

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

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

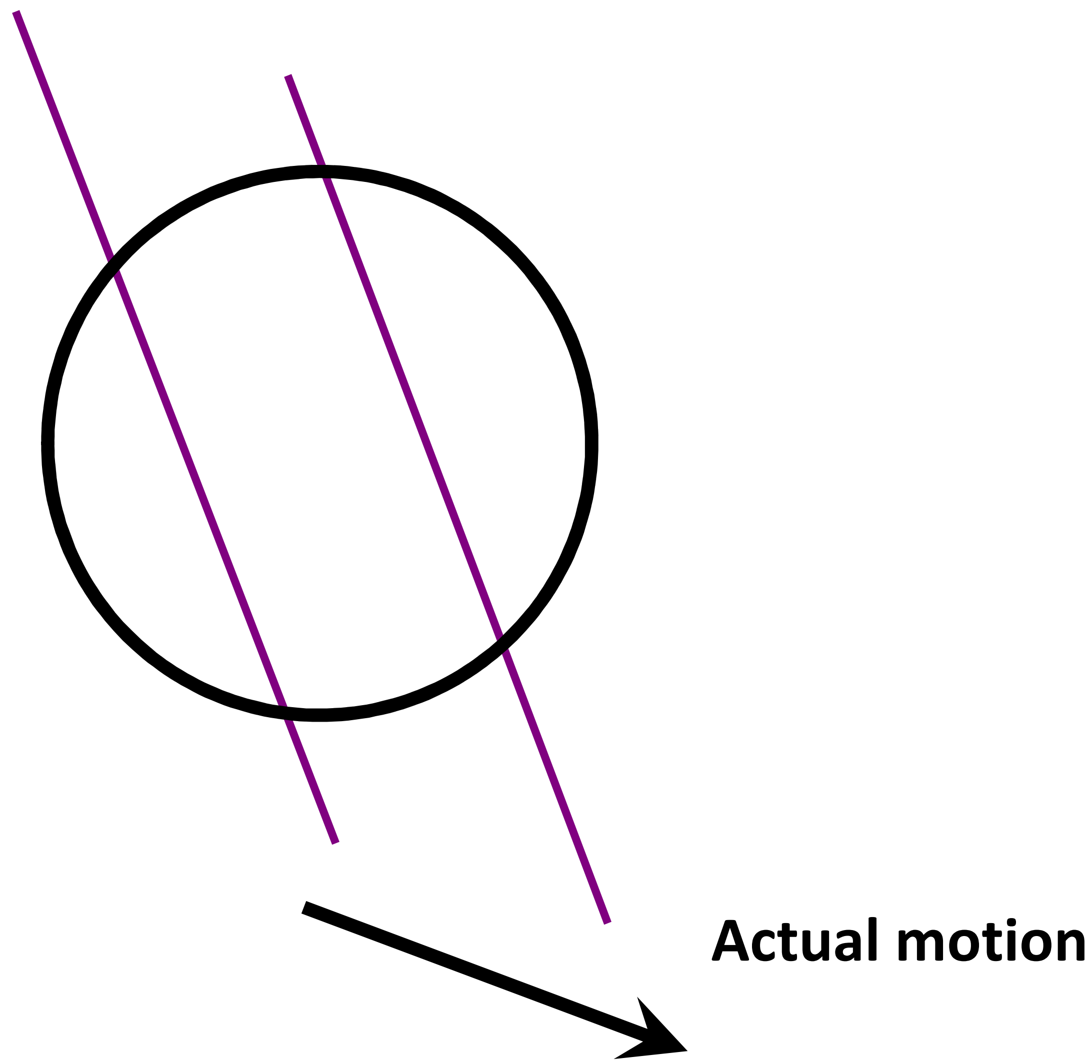
$\underbrace{\nabla I \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \nabla I \cdot \begin{bmatrix} u' \\ v' \end{bmatrix}}$

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



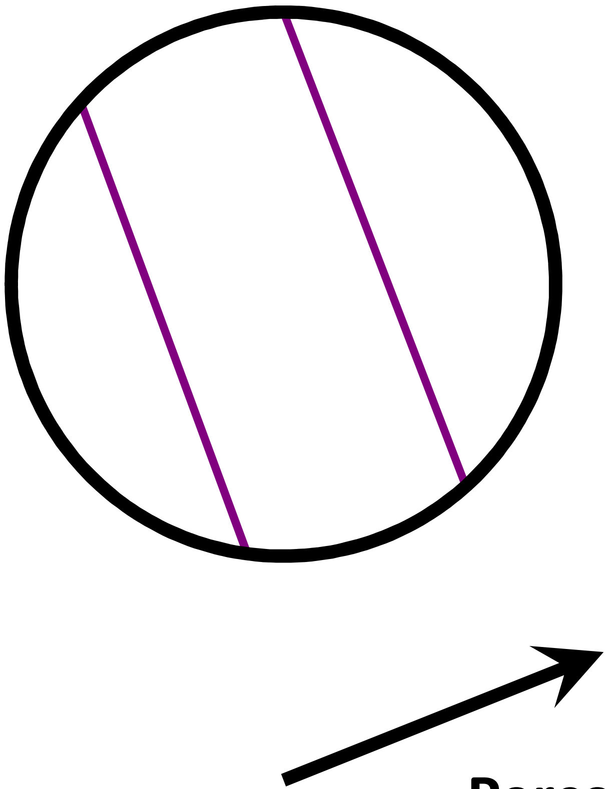
Source: Silvio Savarese

The aperture problem



Source: Silvio Savarese

The aperture problem



Perceived motion

Source: Silvio Savarese

The barber pole illusion

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

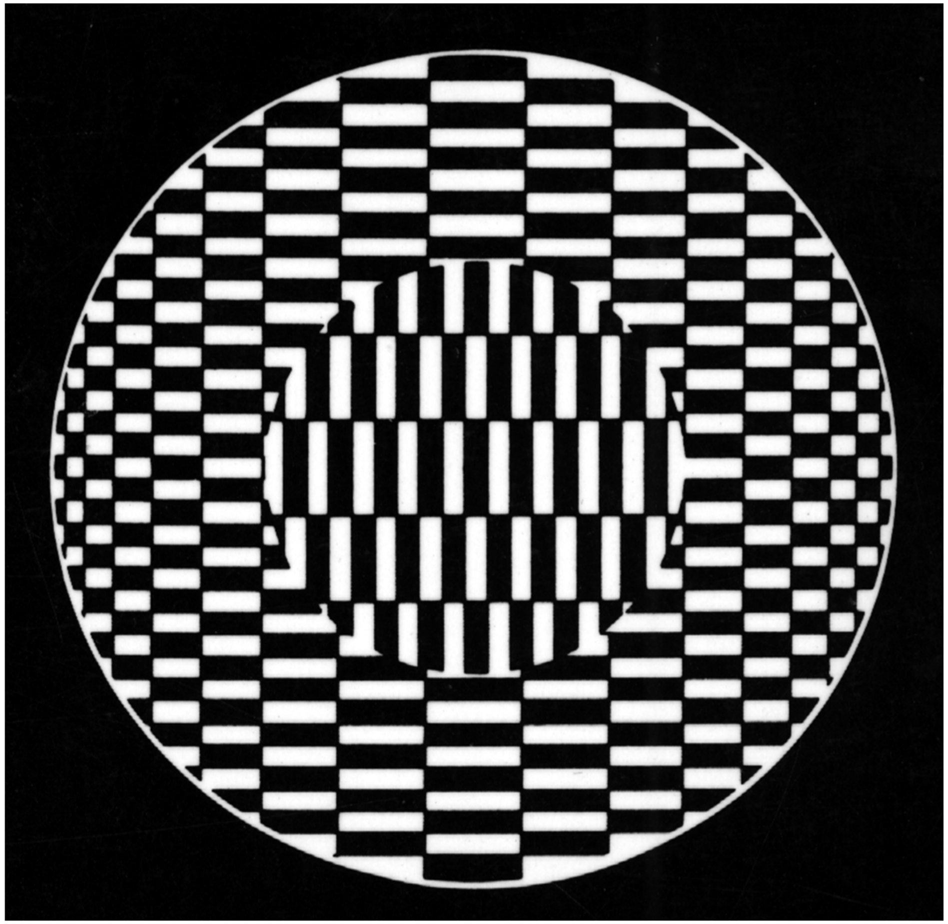
Source: Silvio Savarese

The barber pole illusion

Source: Silvio Savarese

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

Aperture problem cont'd



* From Marc Pollefeys COMP 256 2003

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

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

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

Source: Silvio Savarese

Lucas-Kanade flow

- Overconstrained linear system:

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

Source: Silvio Savarese

Conditions for solvability

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

Source: Silvio Savarese

Lucas-Kanade flow

- Overconstrained linear system

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

25x2

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

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

$$\underbrace{\underset{25\times 2}{A^T}\ \underset{2\times 25}{A}}_{2\times 2} \qquad\qquad A^T b$$

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^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)

Does this remind anything to you?

Source: Silvio Savarese

M = A^TA is the *second moment matrix* !

(Harris corner detector…)

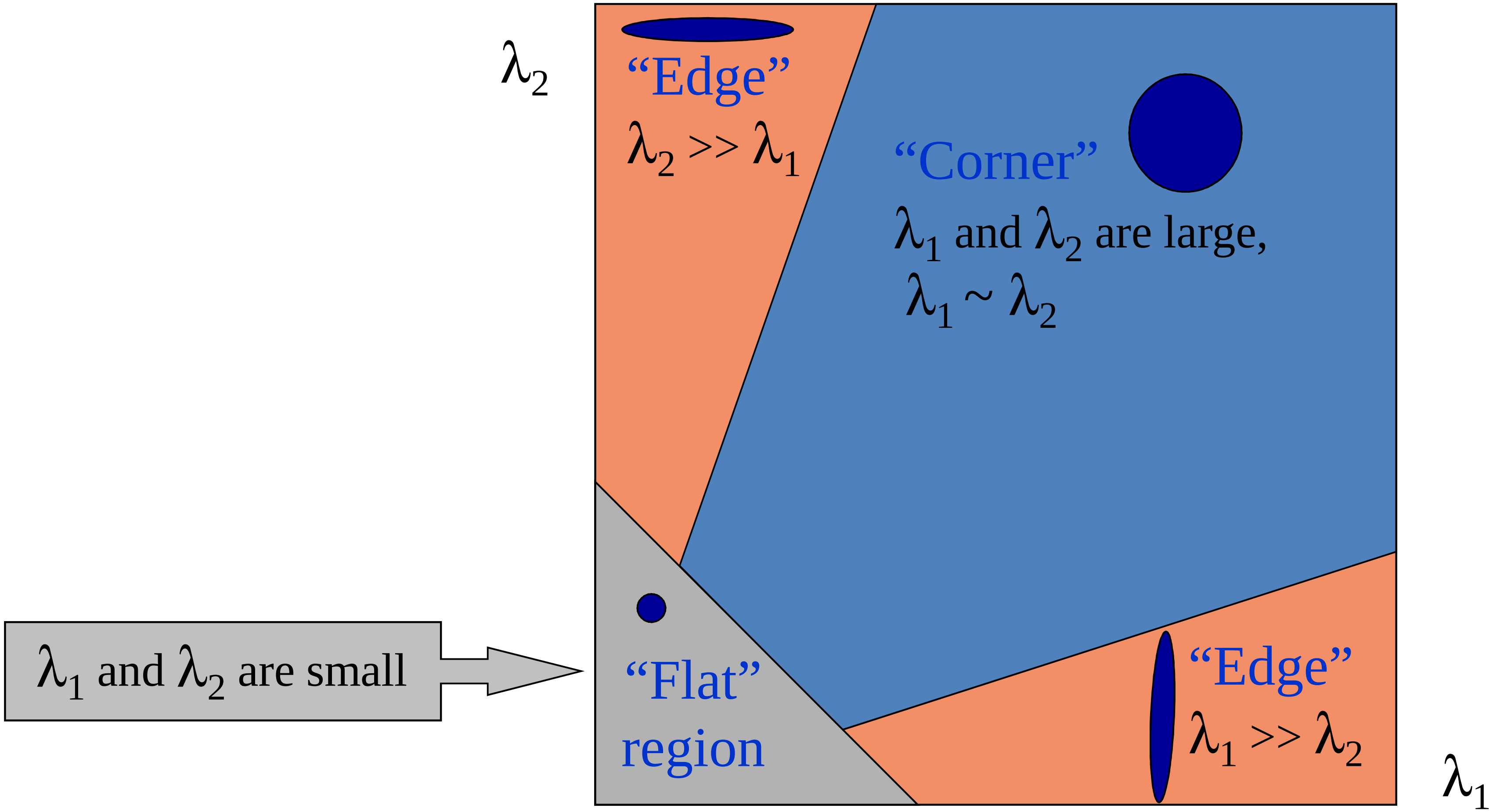
$$A^TA = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

- 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

Source: Silvio Savarese

Interpreting the eigenvalues

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



Source: Silvio Savarese

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

What are good features to track?

- Can measure “quality” of features from just a single image
- Hence: tracking Harris corners (or equivalent) guarantees small error sensitivity!

→ Implemented in Open CV

Source: Silvio Savarese

Recap

- Key assumptions (Errors in Lucas-Kanade)

  - **Small motion:** points do not move very far
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Spatial coherence:** points move like their neighbors

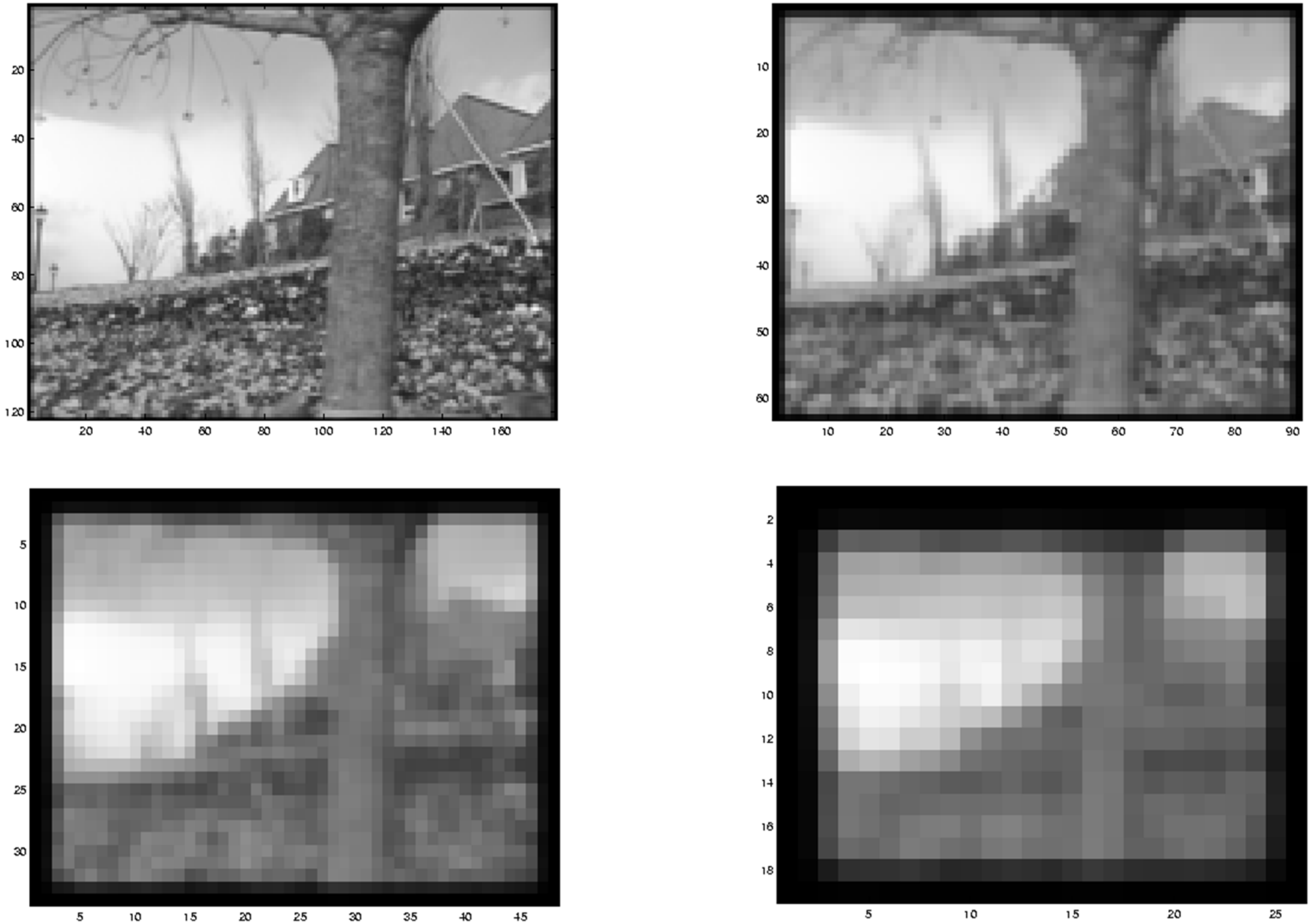
Source: Silvio Savarese

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?

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Reduce the resolution!



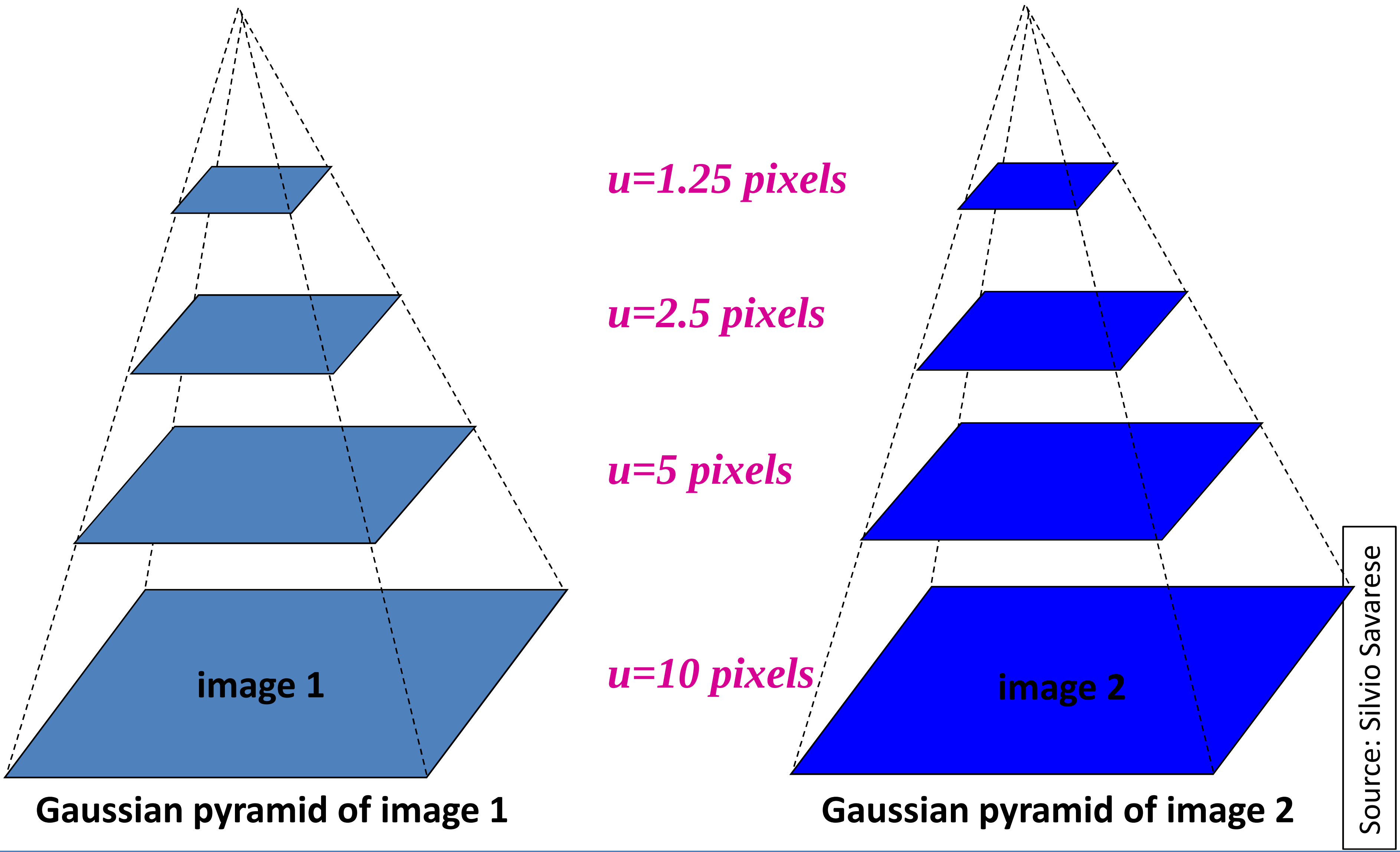
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

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.* $u_i = u_i^* + u_i'$, $v_i = v_i^* + v_i'$.

Source: Silvio Savarese

Coarse-to-fine optical flow estimation

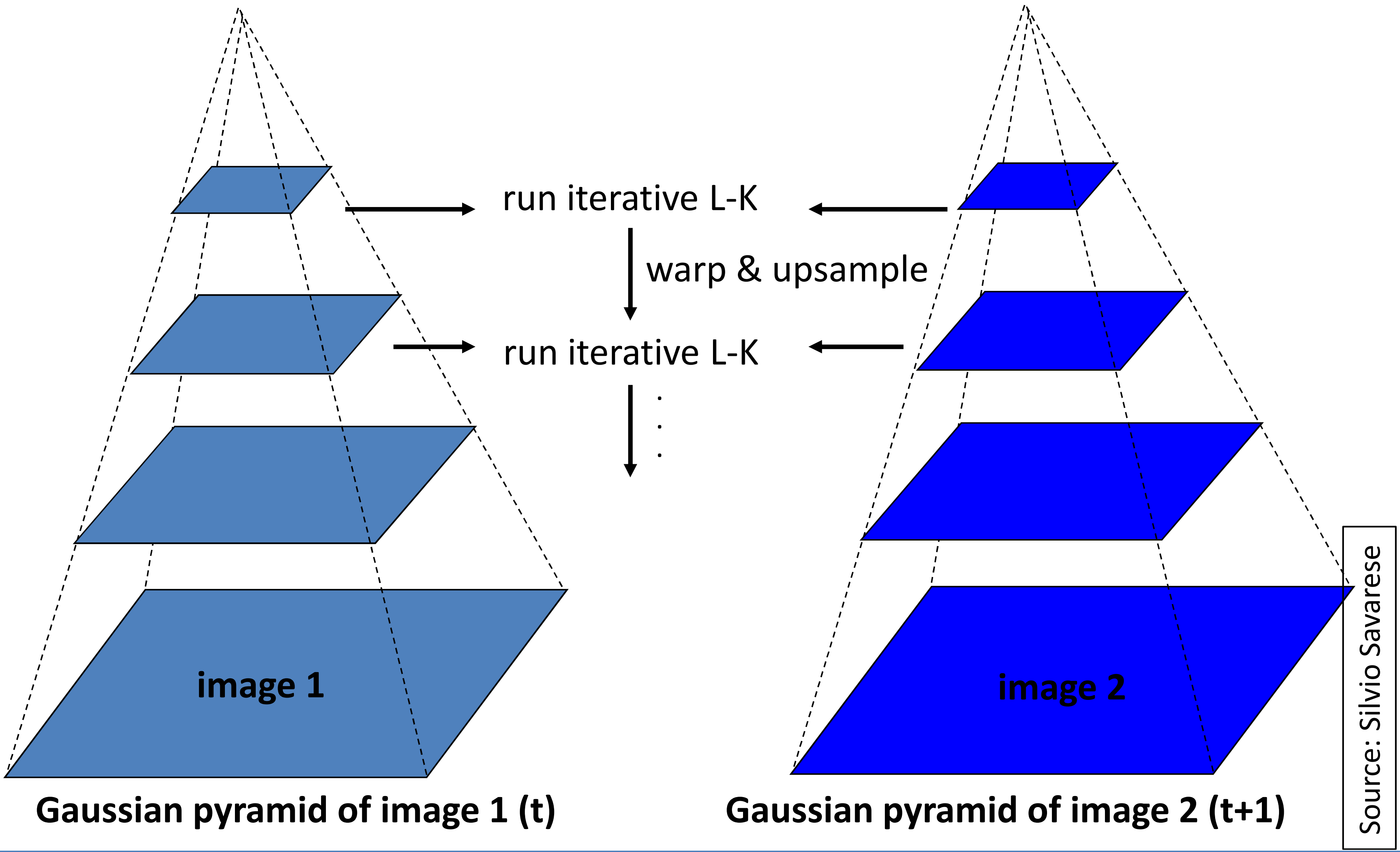


Iterative Refinement

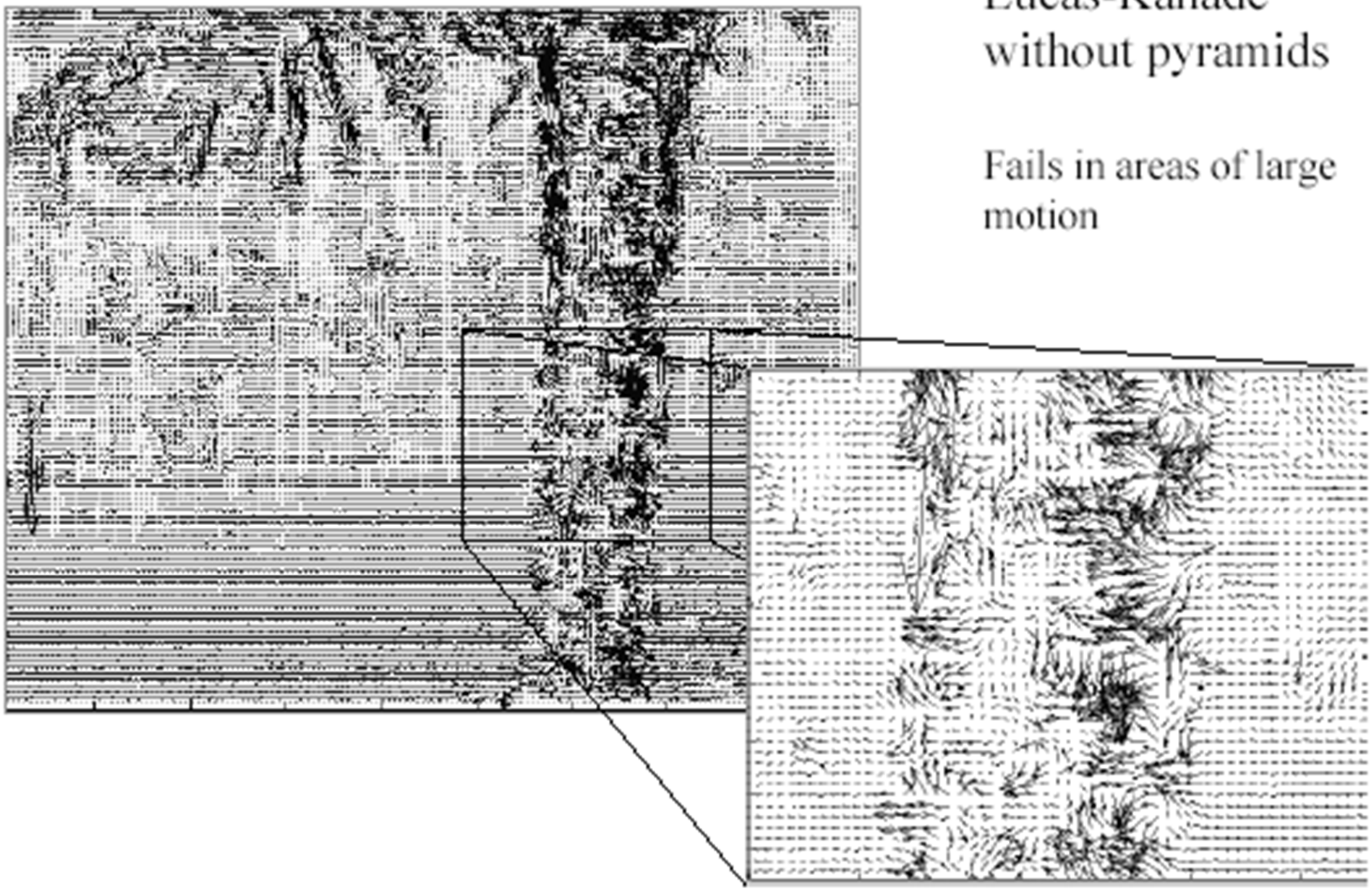
- Iterative Lukas-Kanade Algorithm
  1. Estimate velocity at each pixel by solving Lucas-Kanade equations
  2. Warp I(t-1) towards I(t) using the estimated flow field
     - use image warping techniques
  3. Repeat until convergence

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Coarse-to-fine optical flow estimation



Optical Flow Results

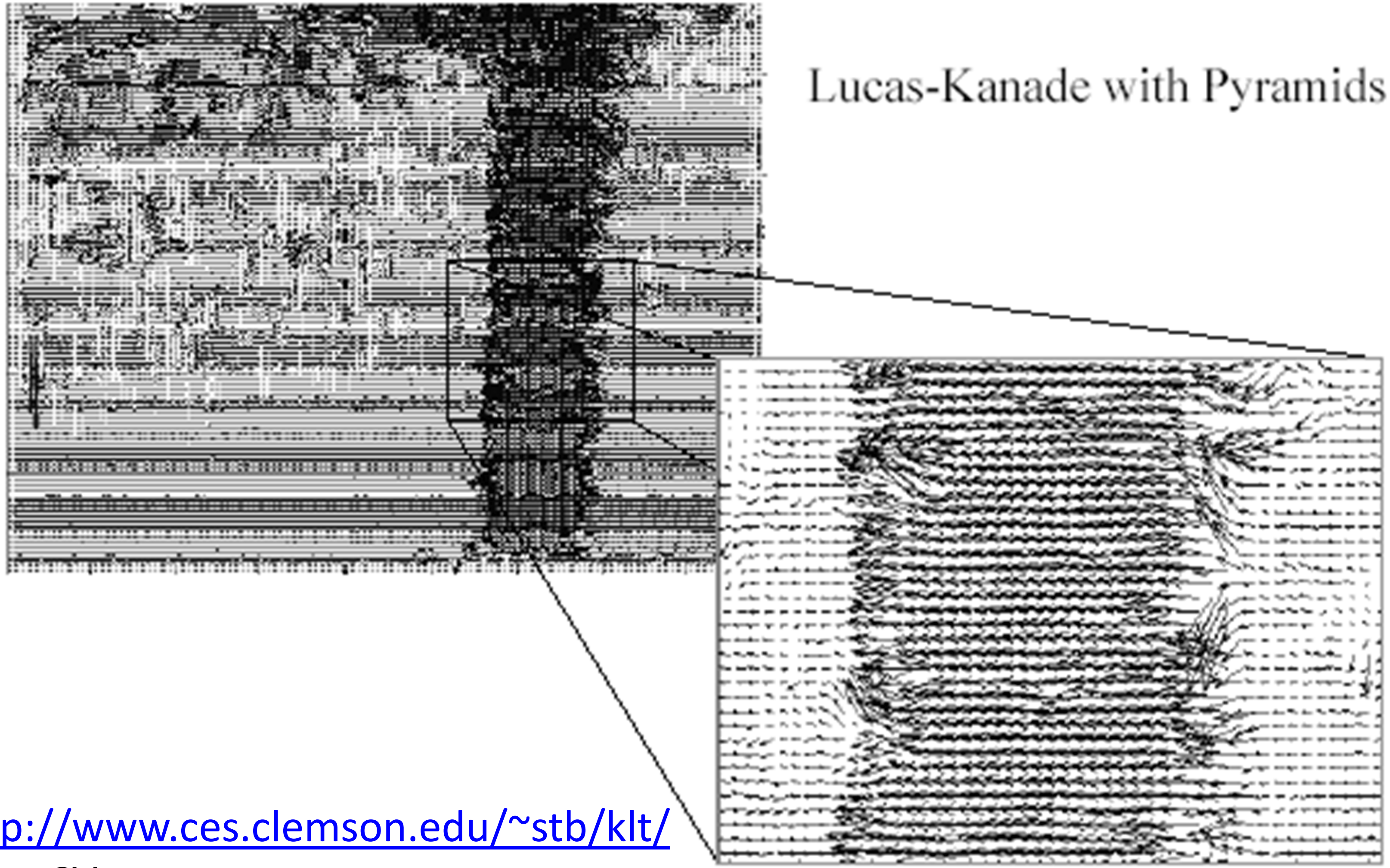


Lucas-Kanade
without pyramids

Fails in areas of large
motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Optical Flow Results



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

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Recap

- Key assumptions (Errors in Lucas-Kanade)

  - **Small motion:** points do not move very far
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Spatial coherence:** points move like their neighbors

Source: Silvio Savarese

Motion segmentation

- How do we represent the motion in this scene?

Source: Silvio Savarese

Motion segmentation

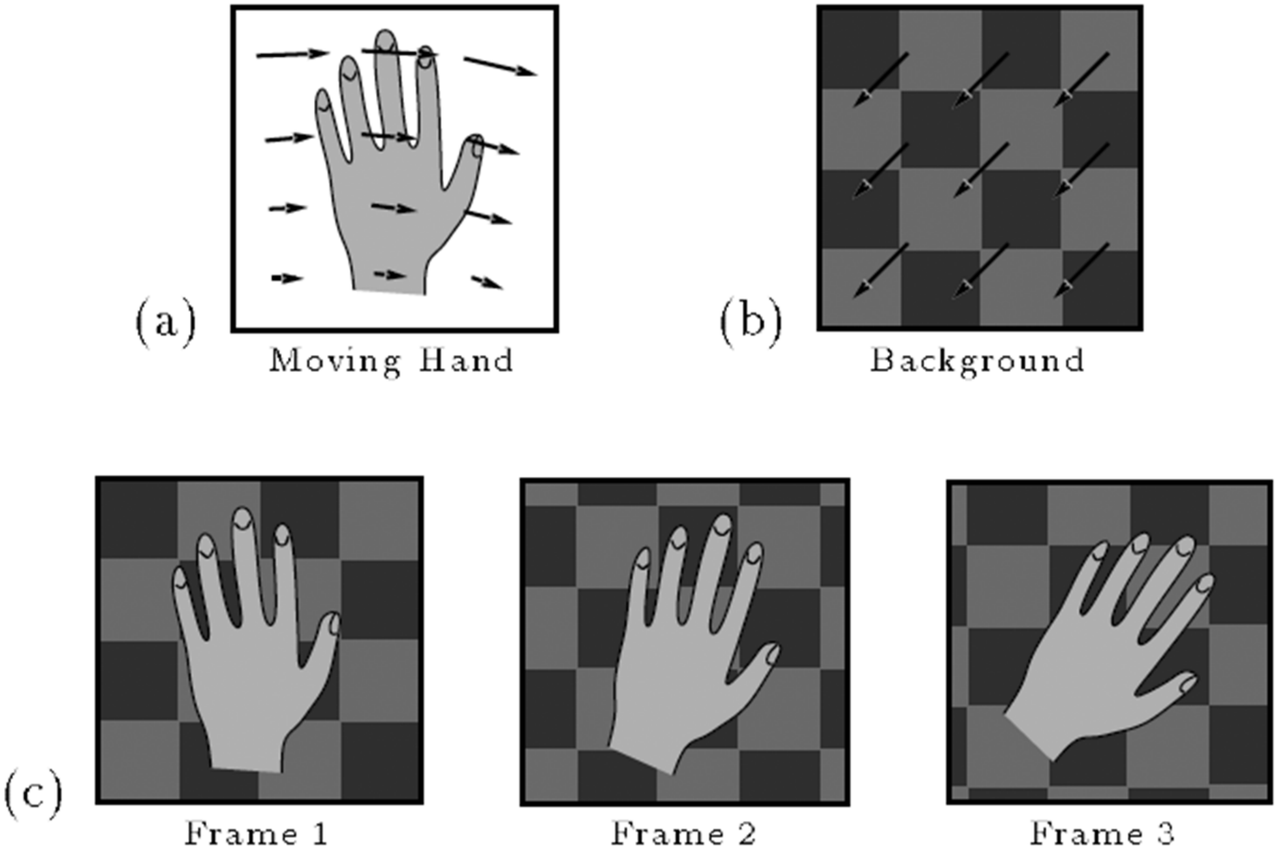
J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

- Break image sequence into “layers” each of which has a coherent (affine) motion

Source: Silvio Savarese

What are layers?

- Each layer is defined by an alpha mask and an affine motion model

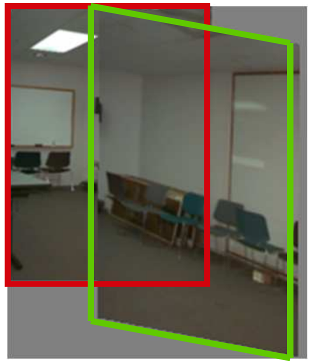


J. Wang and E. Adelson. [Layered Representation for Motion Analysis.](#) *CVPR 1993*.

Source: Silvio Savarese

Affine motion

$$u(x, y) = a_1 + a_2 x + a_3 y$$
$$v(x, y) = a_4 + a_5 x + a_6 y$$



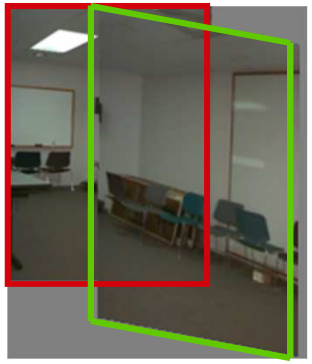
- Substituting into the brightness constancy equation:

$$\boxed{I_x \cdot u + I_y \cdot v + I_t \approx 0}$$

Source: Silvio Savarese

Affine motion

$$u(x,y) = a_1 + a_2 x + a_3 y$$
$$v(x,y) = a_4 + a_5 x + a_6 y$$



- Substituting into the brightness constancy equation:

$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

- Each pixel provides 1 linear constraint in 6 unknowns

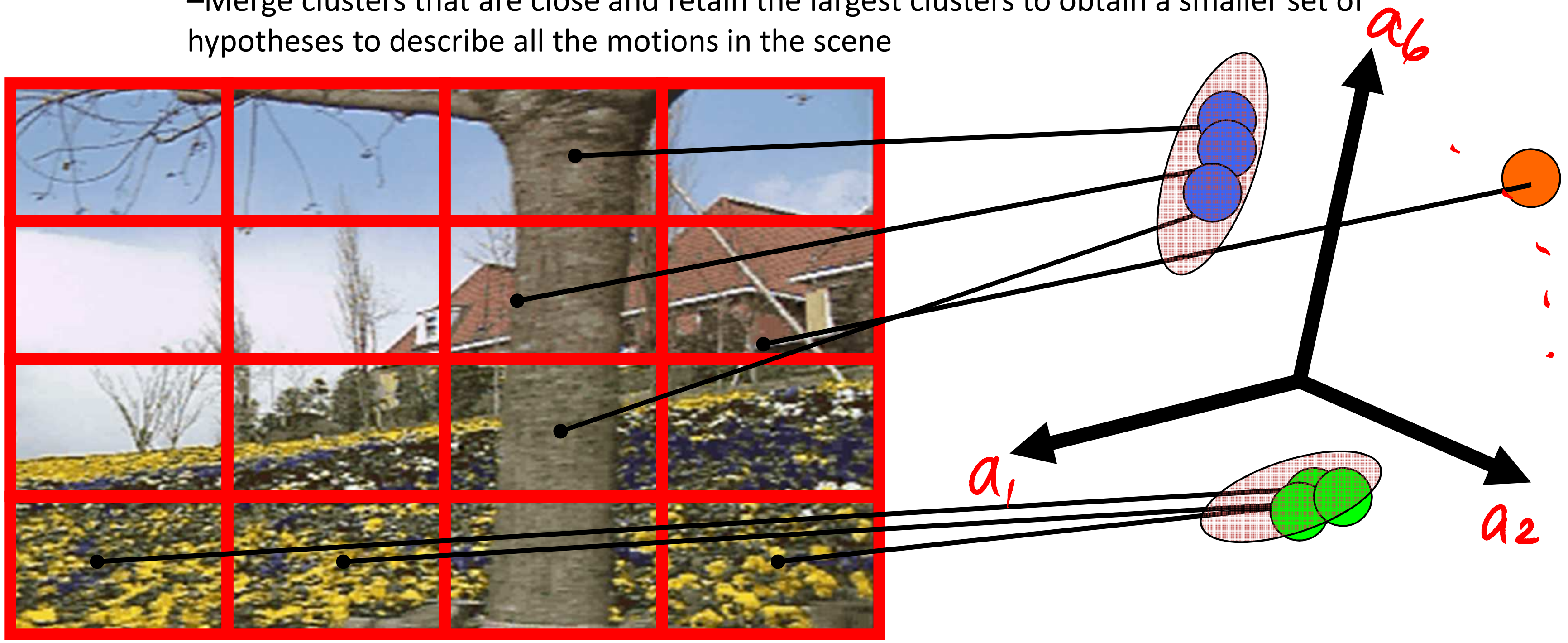
- Least squares minimization:

$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

Source: Silvio Savarese

How do we estimate the layers?

- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error
- Map into motion parameter space
- Perform k-means clustering on affine motion parameters
  - –Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene



Source: Silvio Savarese

How do we estimate the layers?

- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error
  - Map into motion parameter space
  - Perform k-means clustering on affine motion parameters
    - Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene

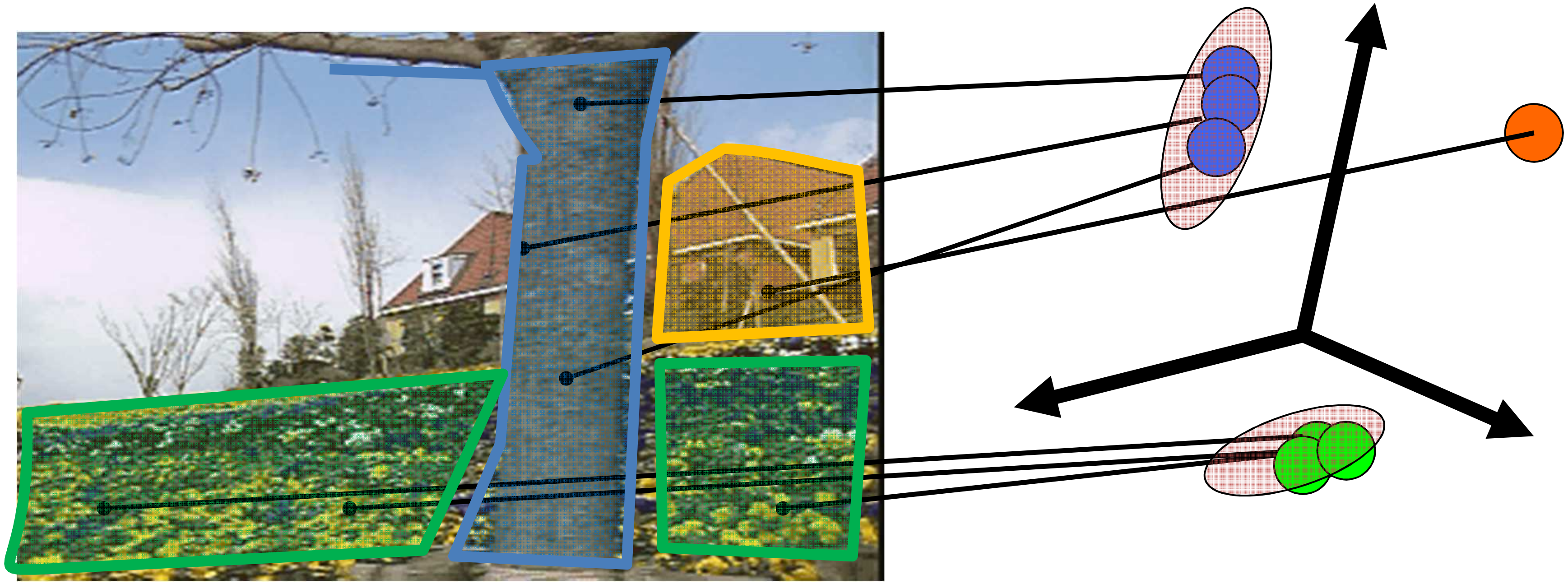
2. Iterate until convergence:

- Assign each pixel to best hypothesis
  - Pixels with high residual error remain unassigned

Source: Silvio Savarese

How do we estimate the layers?

- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error
- Map into motion parameter space
- Perform k-means clustering on affine motion parameters
  - –Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene



Source: Silvio Savarese

How do we estimate the layers?

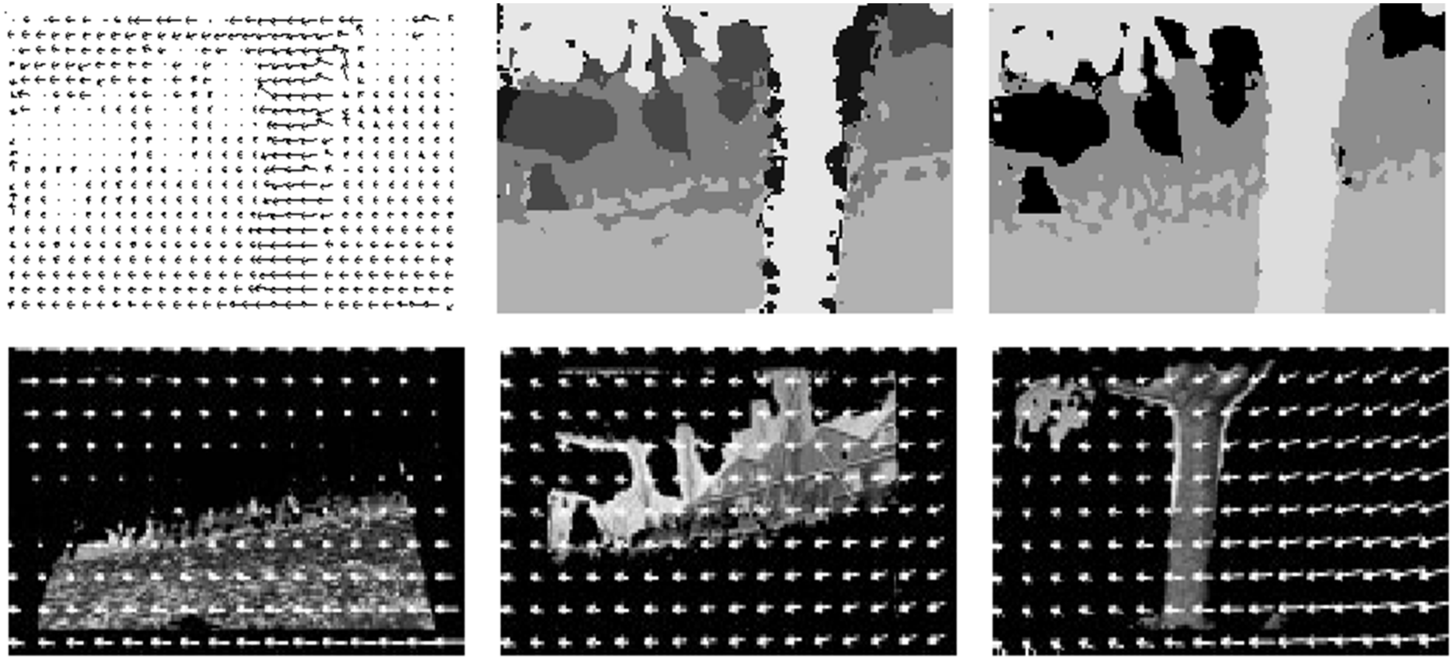
- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error
  - Map into motion parameter space
  - Perform k-means clustering on affine motion parameters
    - Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene

2. Iterate until convergence:

- Assign each pixel to best hypothesis
  - Pixels with high residual error remain unassigned
- Perform region filtering to enforce spatial constraints
- Re-estimate affine motions in each region

Source: Silvio Savarese

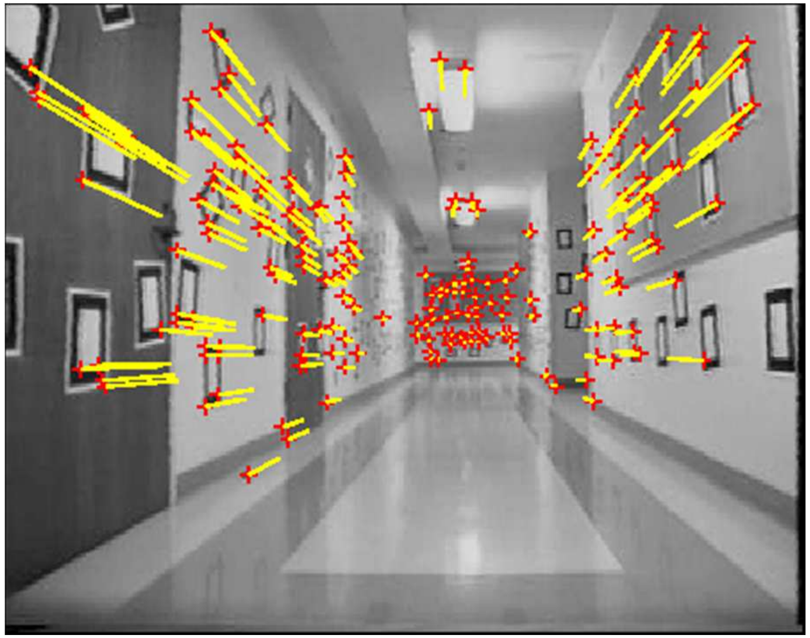
Example result

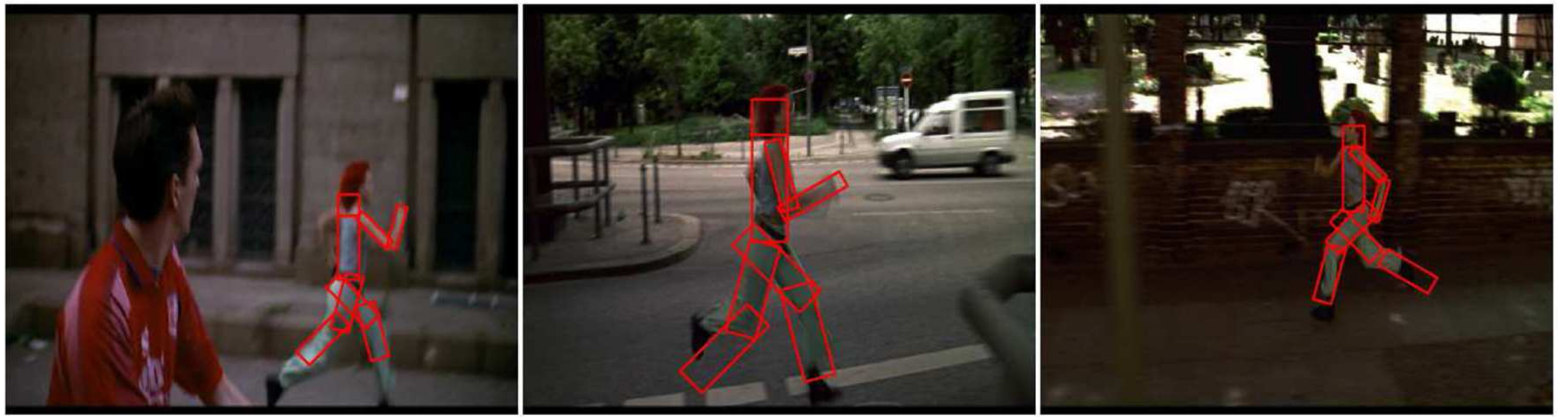


J. Wang and E. Adelson. [Layered Representation for Motion Analysis.](#) *CVPR 1993*.

Source: Silvio Savarese

Tracking





Sources: Kristen Grauman, Deva Ramanan

What we will learn today?

- Introduction
- Optical flow
- Feature tracking
- Applications

Motion estimation techniques

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

- Feature-tracking
  - Extract visual features (corners, textured areas) and "track" them over multiple frames
    - Shi-Tomasi feature tracker
    - Tracking with dynamics

Source: Silvio Savarese

Feature tracking

- So far, we have only considered optical flow estimation in a pair of images
- If we have more than two images, we can compute the optical flow from each frame to the next
- Given a point in the first image, we can in principle reconstruct its path by simply "following the arrows"

Source: Silvio Savarese

Tracking challenges

- Ambiguity of optical flow
  – Find good features to track
- Large motions
  – Discrete search instead of Lucas-Kanade
- Changes in shape, orientation, color
  – Allow some matching flexibility
- Occlusions, dis-occlusions
  – Need mechanism for deleting, adding new features
- Drift – errors may accumulate over time
  – Need to know when to terminate a track

Source: Silvio Savarese

Shi-Tomasi feature tracker

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

- Find good features using eigenvalues of second-moment matrix
  - Key idea: "good" features to track are the ones that can be tracked reliably
- From frame to frame, track with Lucas-Kanade and a pure *translation* model
  - More robust for small displacements, can be estimated from smaller neighborhoods
- 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

Source: Silvio Savarese

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).

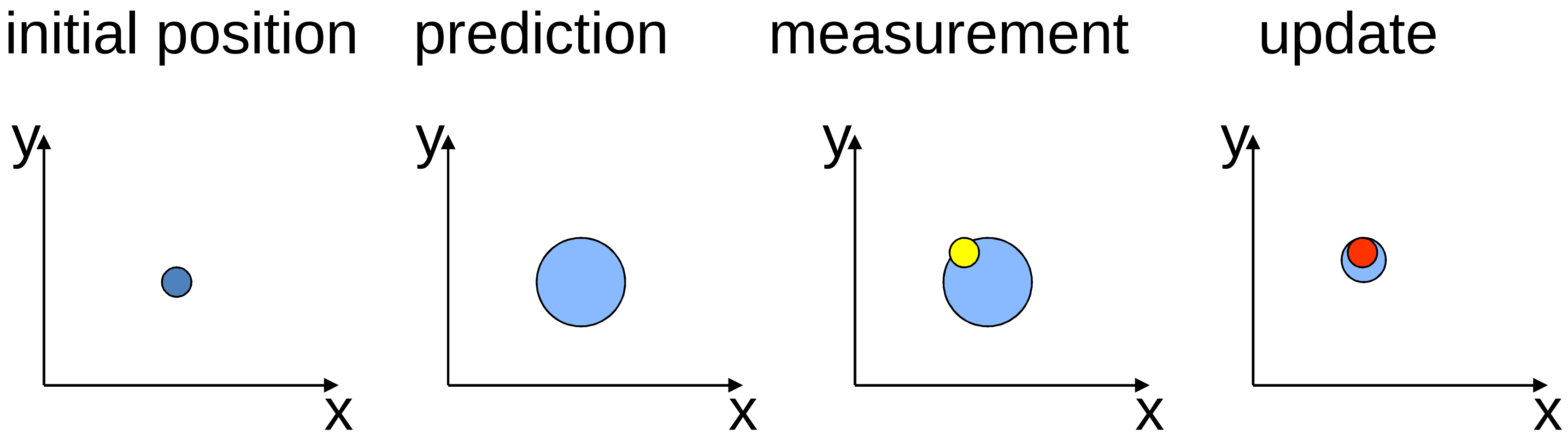
Source: Silvio Savarese

Tracking with dynamics

- Key idea: Given a model of expected motion, predict where objects will occur in next frame, even before seeing the image
  - Restrict search for the object
  - Improved estimates since measurement noise is reduced by trajectory smoothness

Source: Silvio Savarese

Tracking with dynamics



The Kalman filter:

- Method for tracking linear dynamical models in Gaussian noise
- The predicted/corrected state distributions are Gaussian
  - Need to maintain the mean and covariance
  - Calculations are easy (all the integrals can be done in closed form)

Source: Silvio Savarese

2D Target tracking using Kalman filter in MATLAB

by AliReza KashaniPour

http://www.mathworks.com/matlabcentral/fileexchange/14243

Source: Silvio Savarese

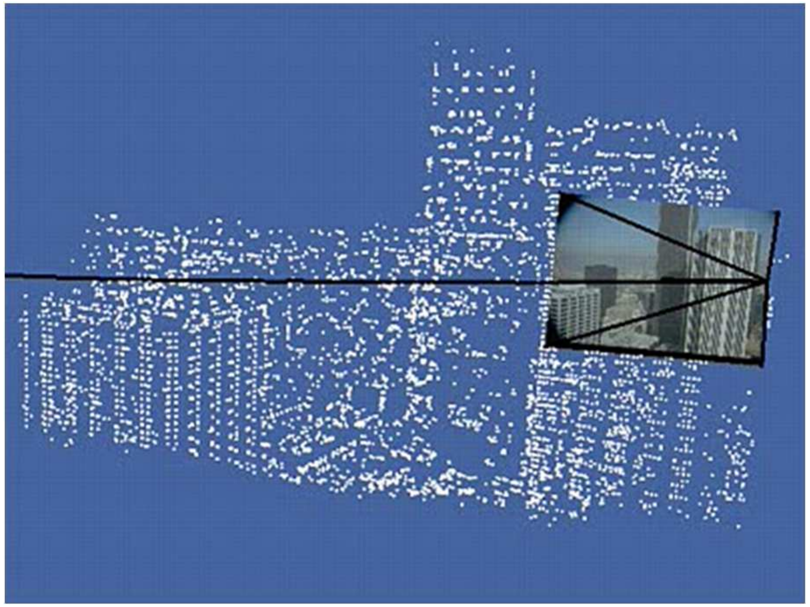
What we will learn today?

- Introduction
- Optical flow
- Feature tracking
- 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

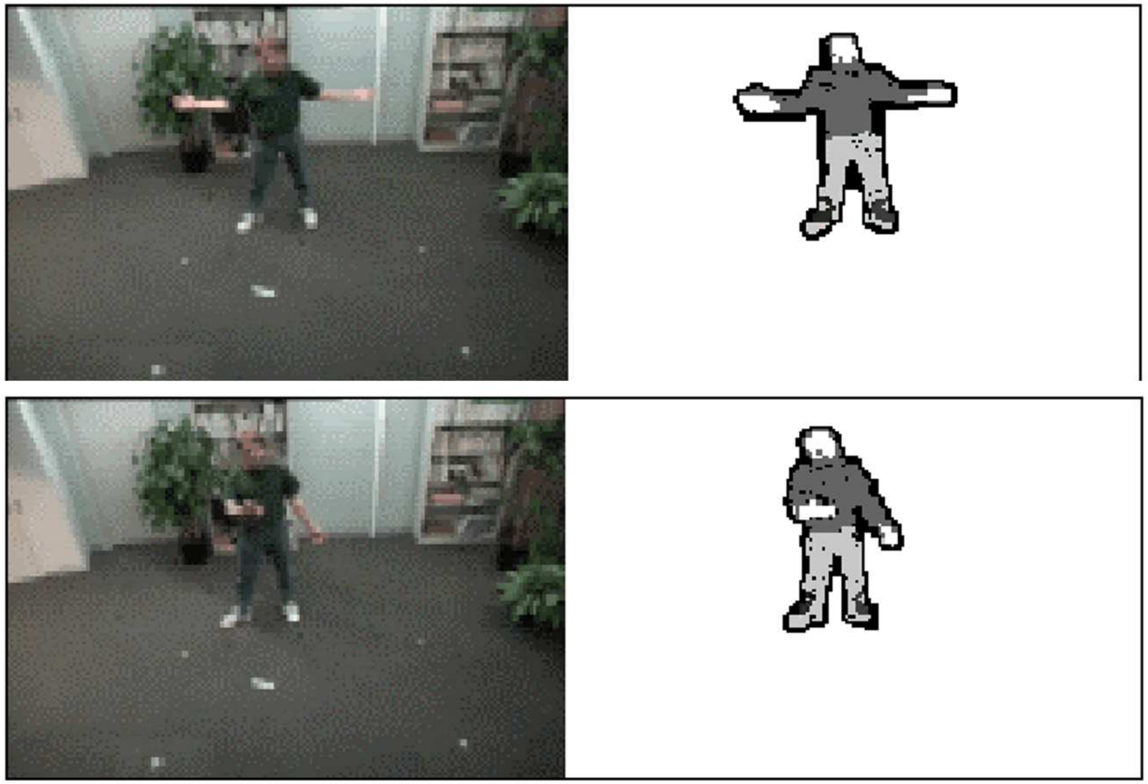
Estimating 3D structure



Source: Silvio Savarese

Segmenting objects based on motion cues

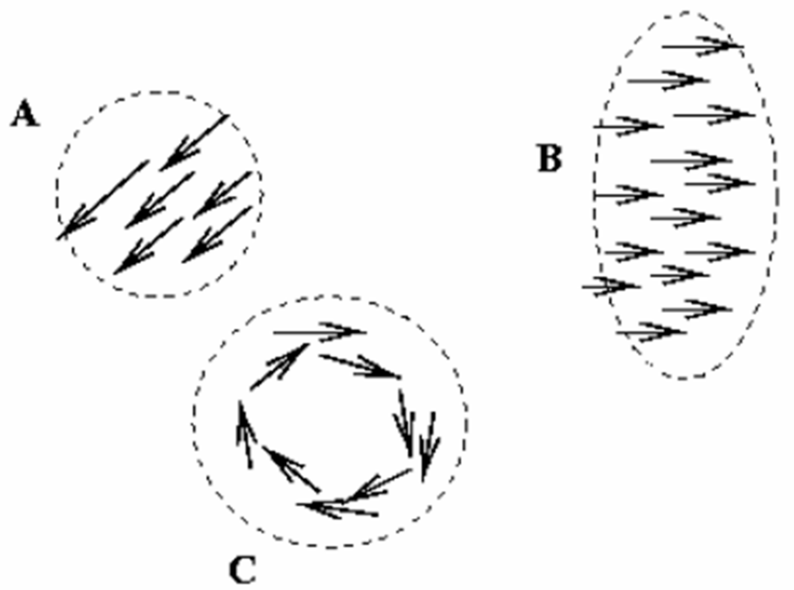
- Background subtraction
  - A static camera is observing a scene
  - Goal: separate the static *background* from the moving *foreground*

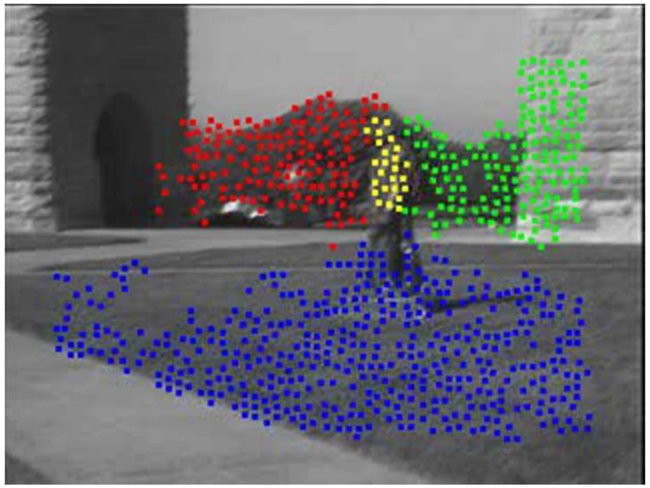


Source: Silvio Savarese

Segmenting objects based on motion cues

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

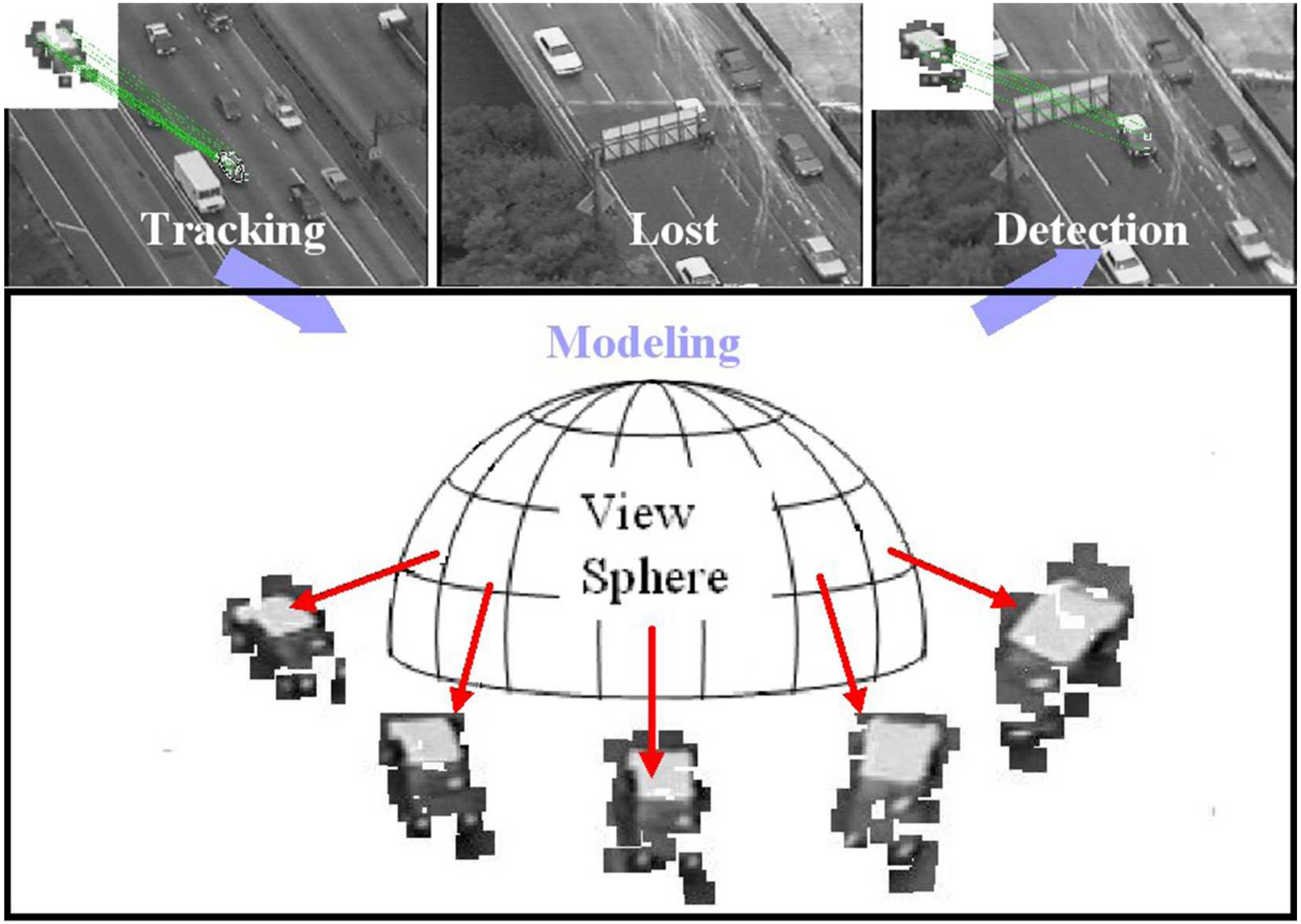




S. J. Pundlik and S. T. Birchfield, Motion Segmentation at Any Speed, Proceedings of the British Machine Vision Conference (BMVC) 2006

Source: Silvio Savarese

Tracking objects



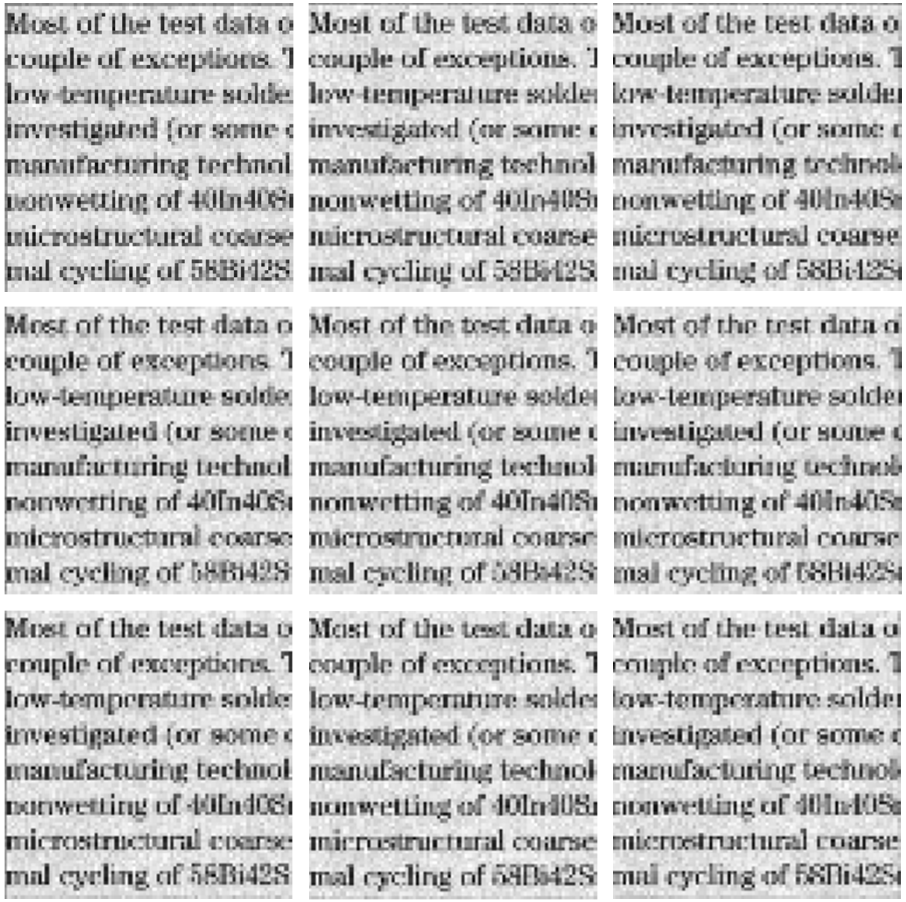
Z.Yin and R.Collins, "On-the-fly Object Modeling while Tracking," *IEEE Computer Vision and Pattern Recognition (CVPR '07),* Minneapolis, MN, June 2007.

Source: Silvio Savarese

Synthesizing dynamic textures

Super-resolution

Example: A set of low quality images



Source: Silvio Savarese

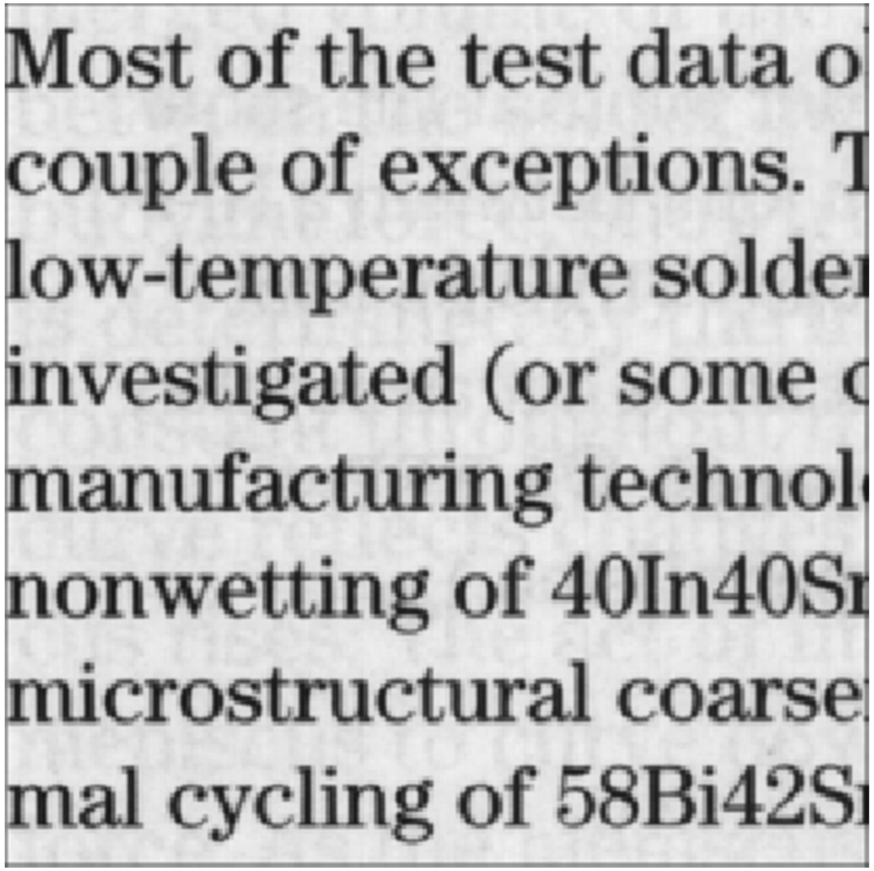
Super-resolution

Each of these images looks like this:

Source: Silvio Savarese

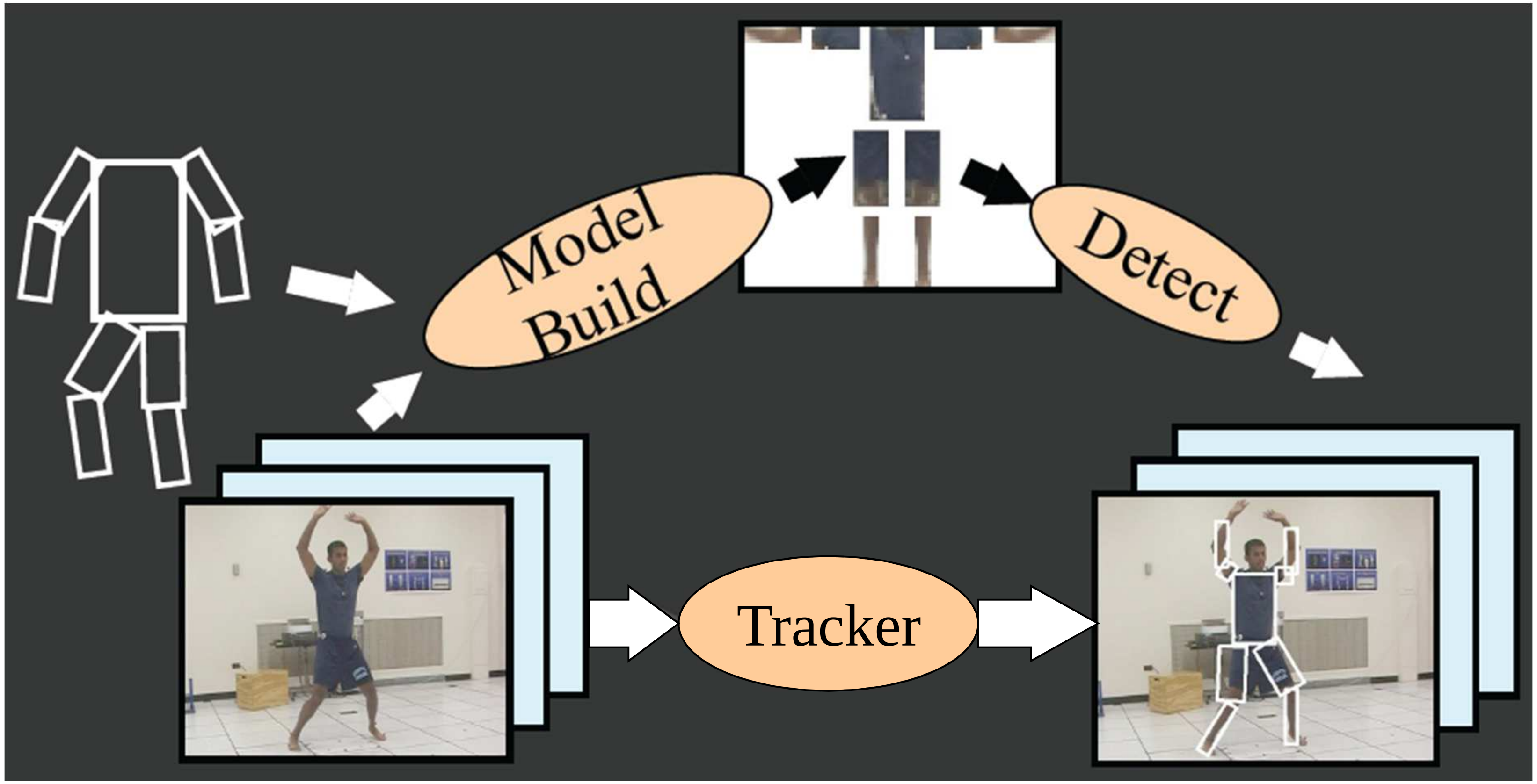
Super-resolution

The recovery result:


Most of the test data o
couple of exceptions. T
low-temperature solde
investigated (or some
manufacturing technol
nonwetting of 40In40S
microstructural coarse
mal cycling of 58Bi42S

Source: Silvio Savarese

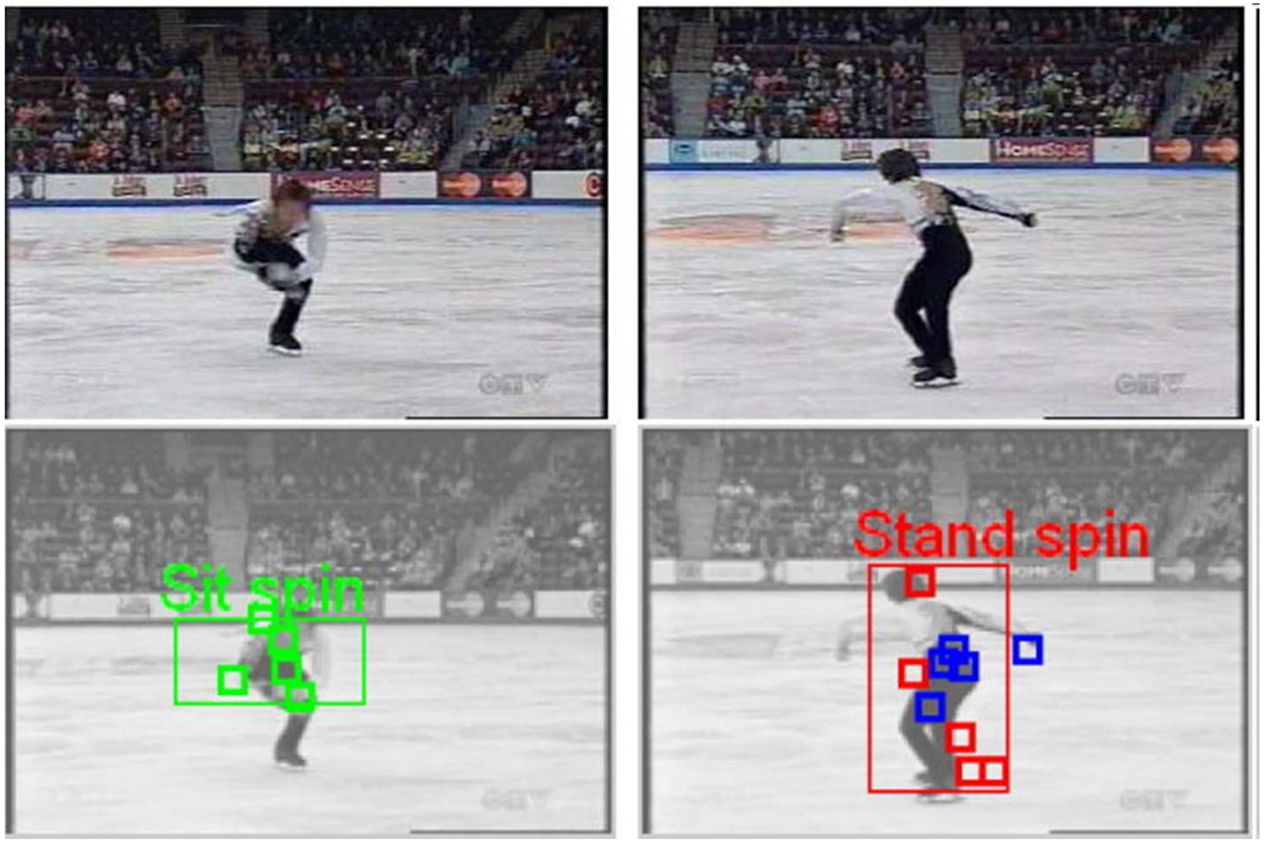
Recognizing events and activities



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

Source: Silvio Savarese

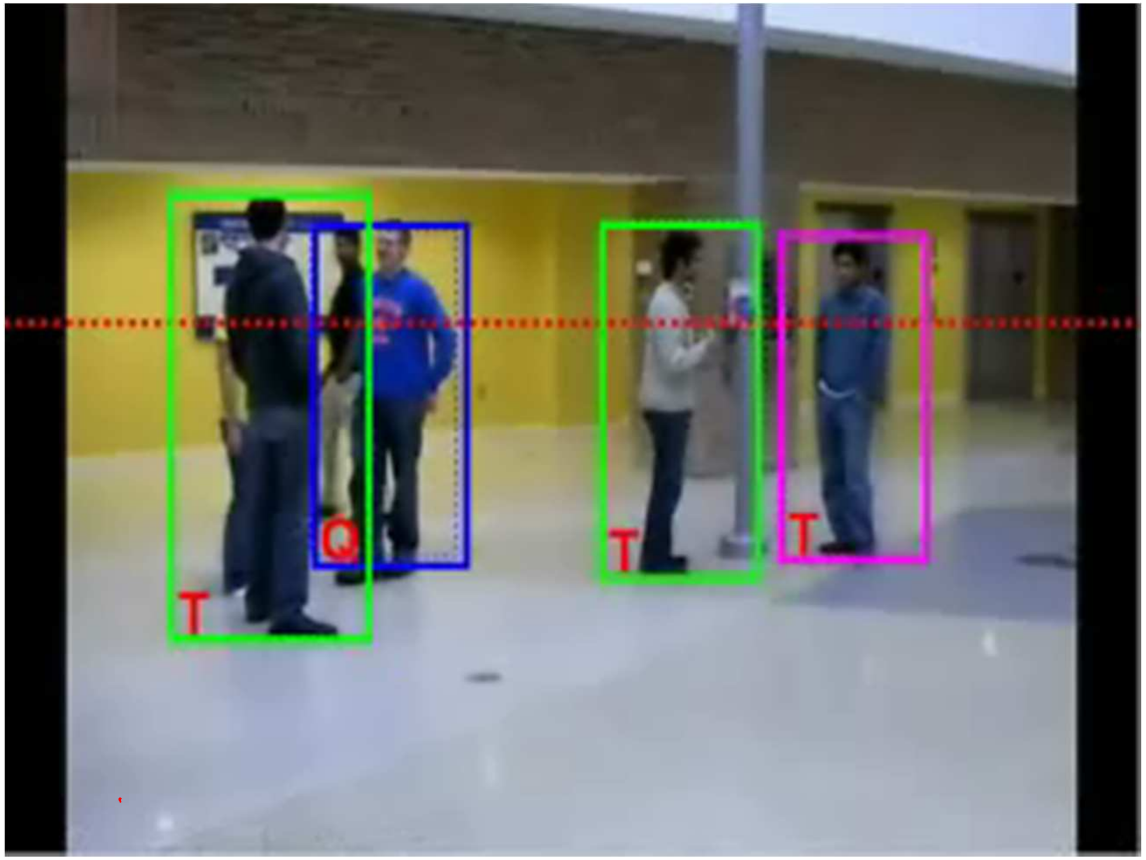
Recognizing events and activities



Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, **Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words**, *(BMVC)*, Edinburgh, 2006.

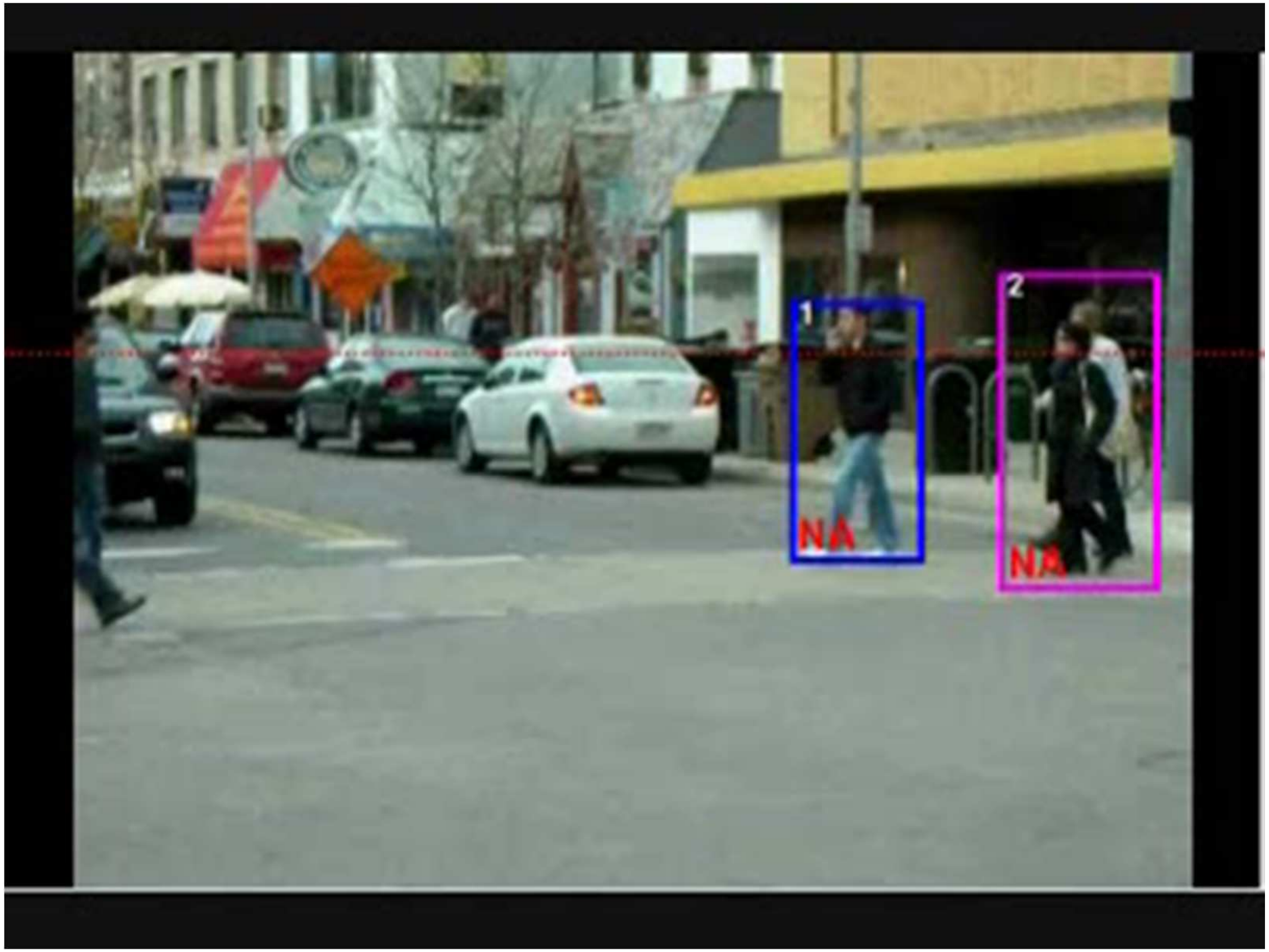
Recognizing events and activities

Crossing – Talking – Queuing – Dancing – jogging



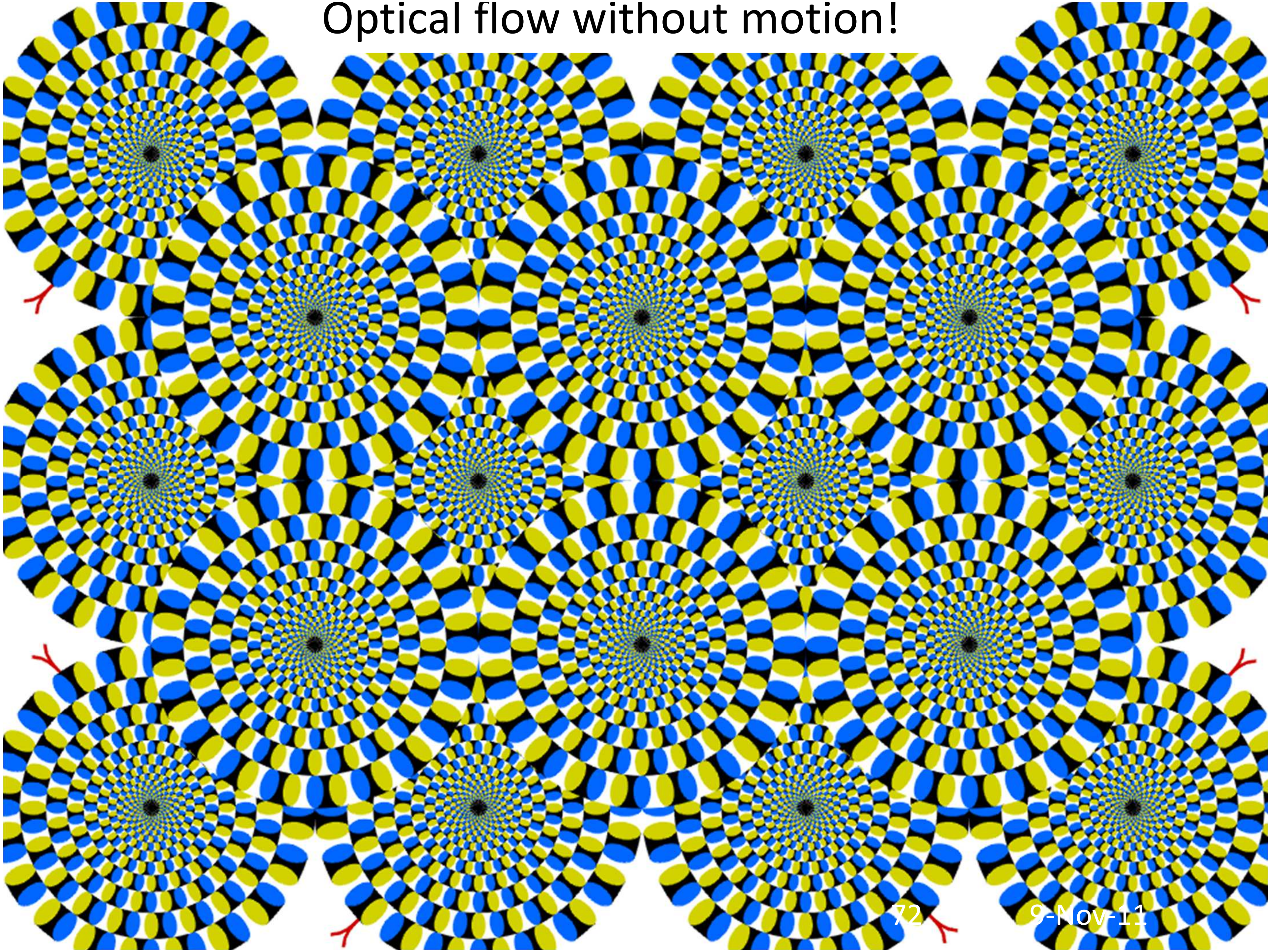
W. Choi & K. Shahid & S. Savarese WMC 2010

Source: Silvio Savarese



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

Optical flow without motion!



What we have learned today?

- Introduction
- Optical flow
- Feature tracking
- Applications
- (Problem Set 3 (Q1))