Lecture: Motion

Juan Carlos Niebles and Ranjay Krishna
Stanford Vision and Learning Lab

What we will learn today?

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Common fate
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

What we will learn today?

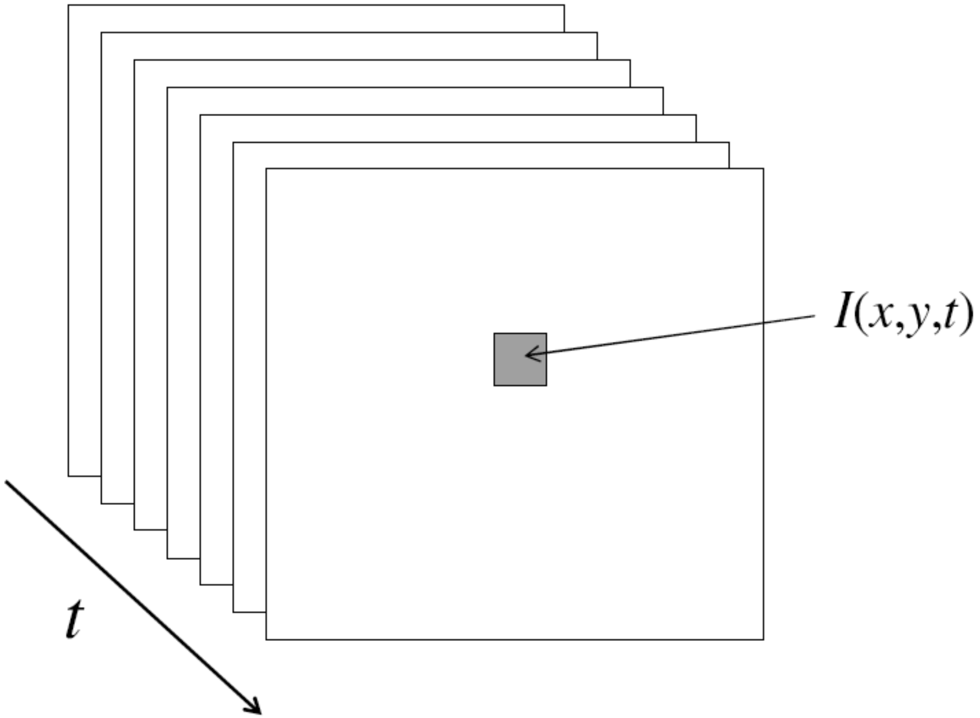
- Optical flow
- Horn-Schunk method
- Lucas-Kanade method
- Pyramids for large motion
- Common fate
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

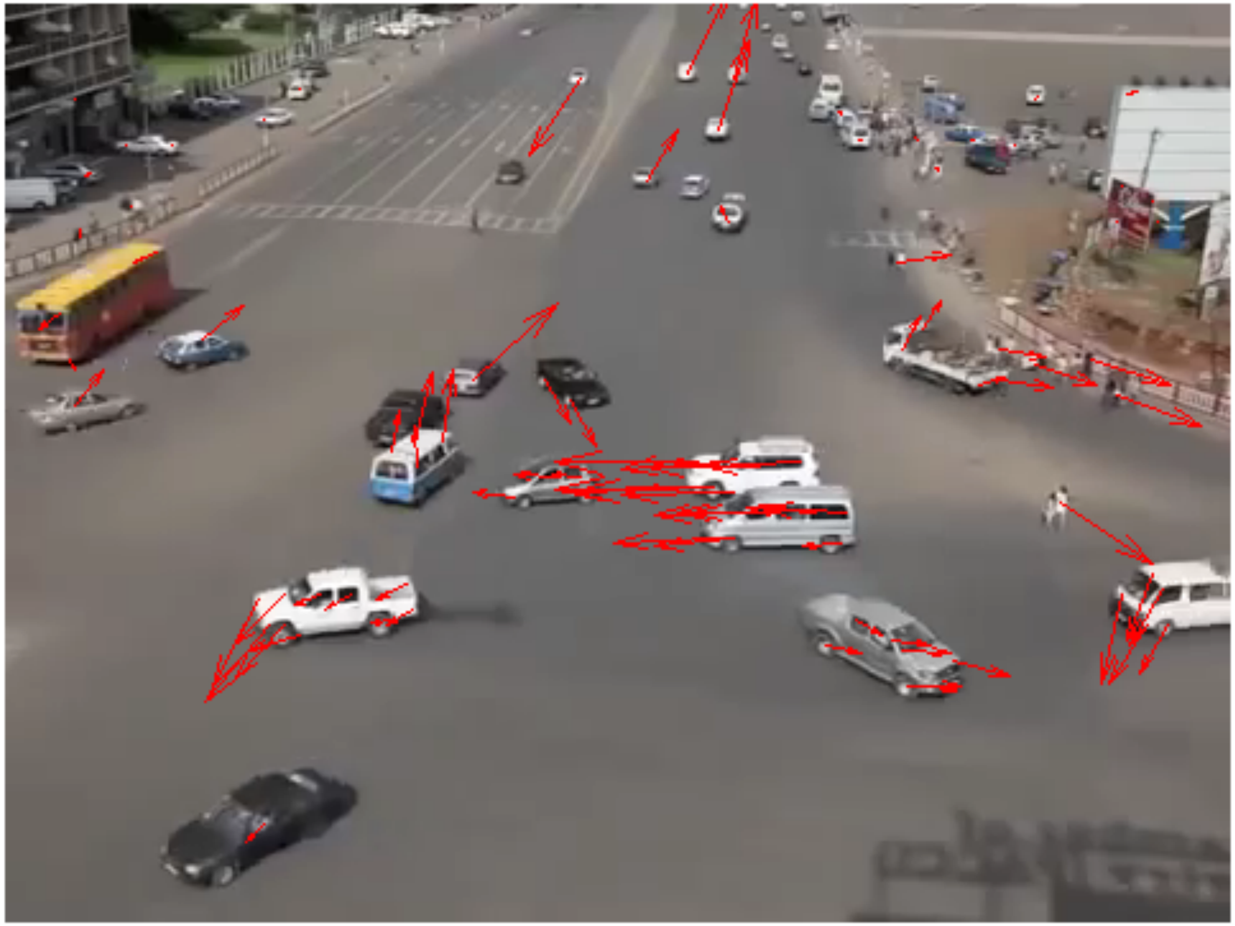
From images to videos

- 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?



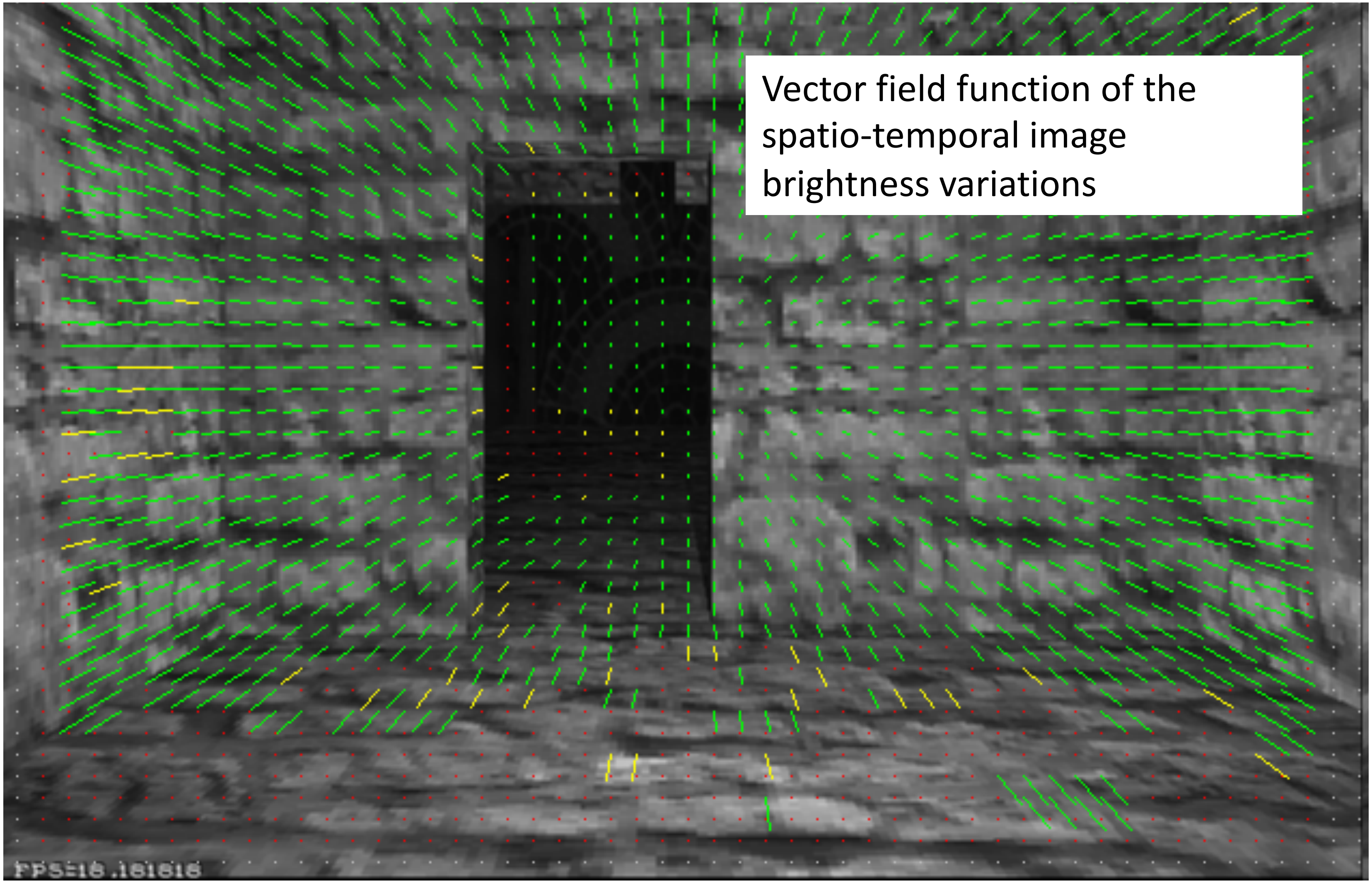
Optical flow

- 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 vs. a stationary sphere under moving illumination

GOAL: Recover image motion at each pixel from optical flow

Source: Silvio Savarese

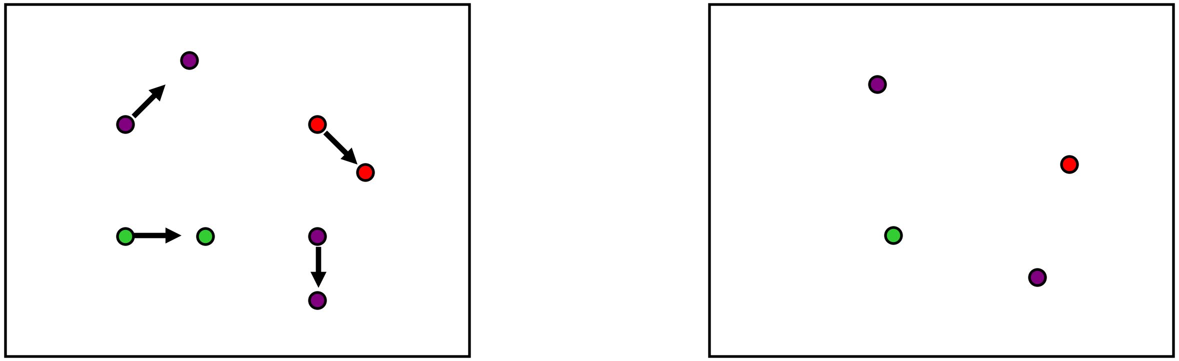
Optical flow



Vector field function of the spatio-temporal image brightness variations

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

Estimating optical flow



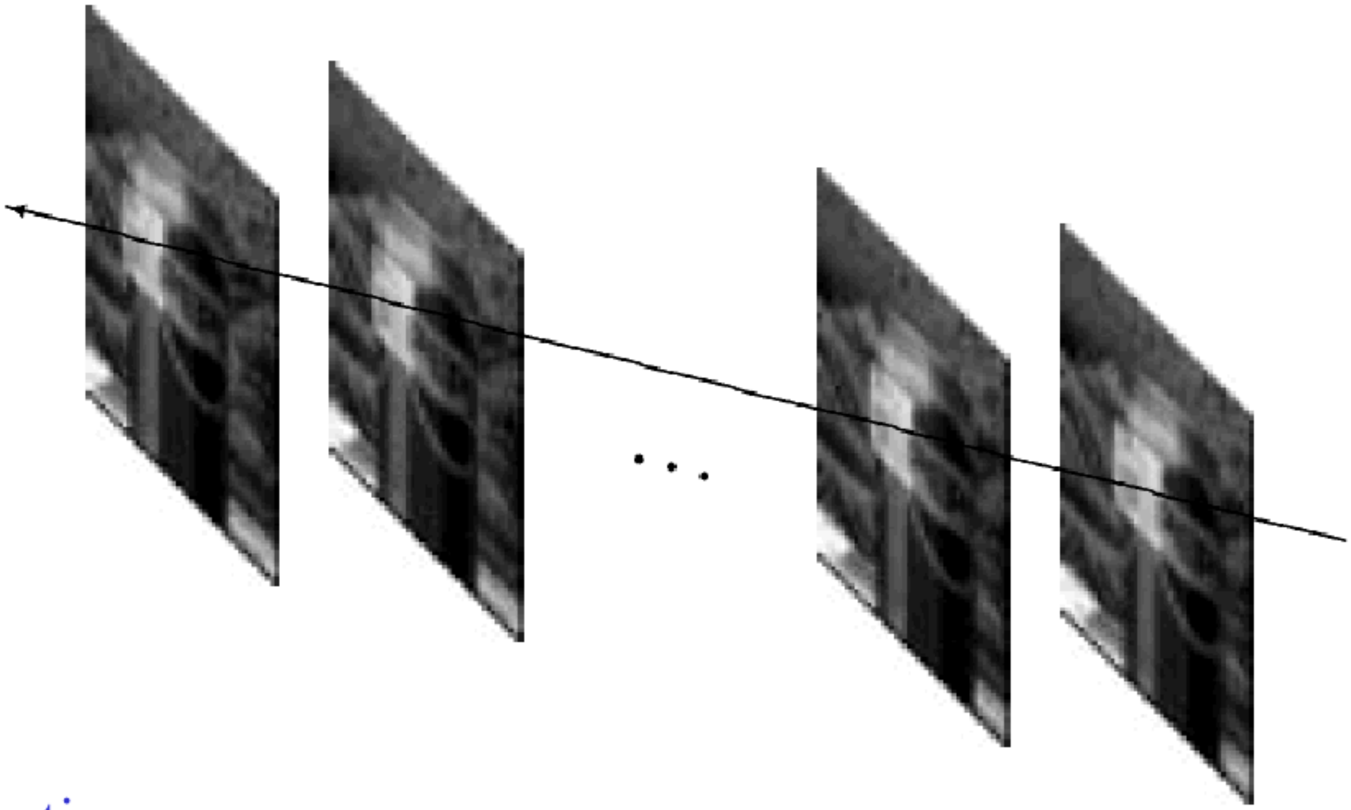
$I(x,y,t-1)$ $I(x,y,t)$

- 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

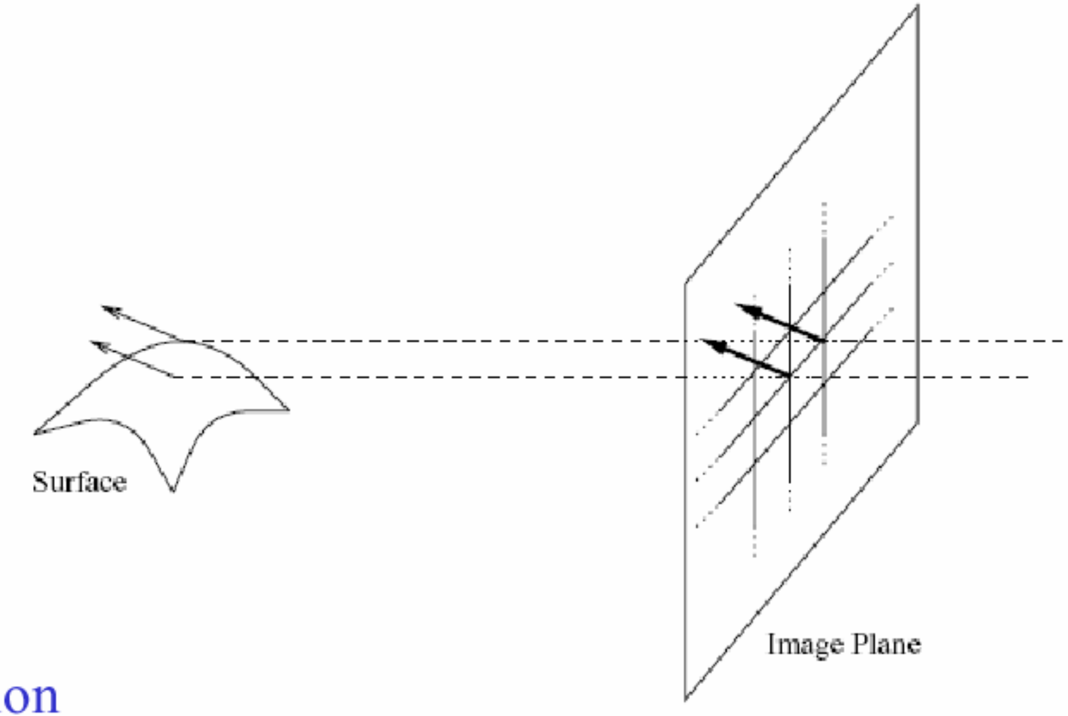


Assumption:

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

* Slide from Michael Black, CS143 2003

Key Assumptions: spatial coherence

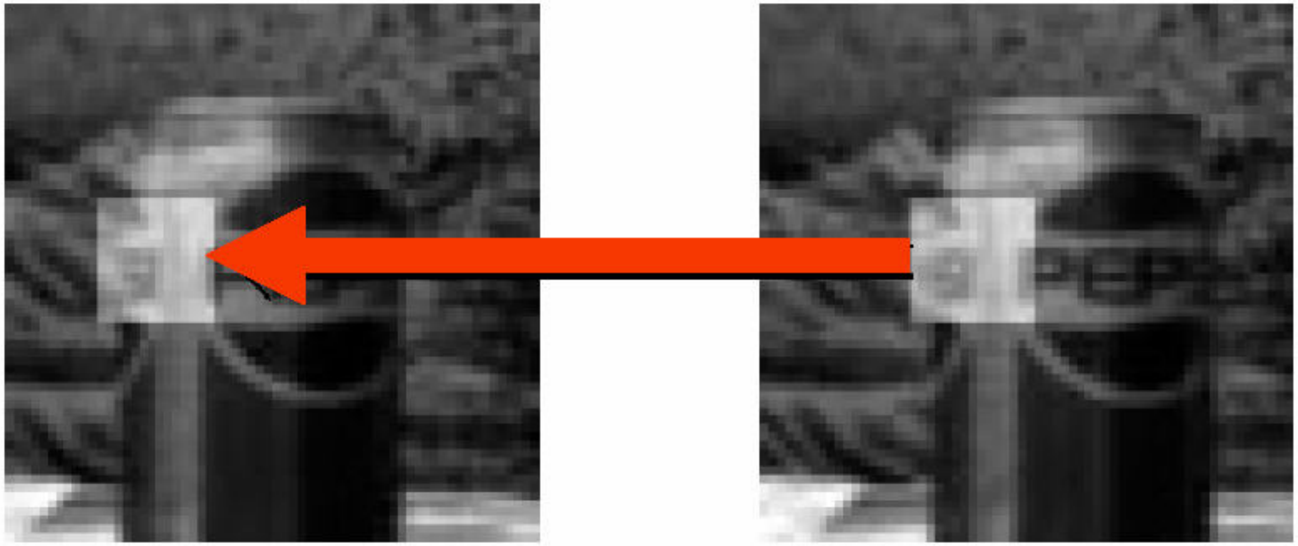


Assumption

* 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

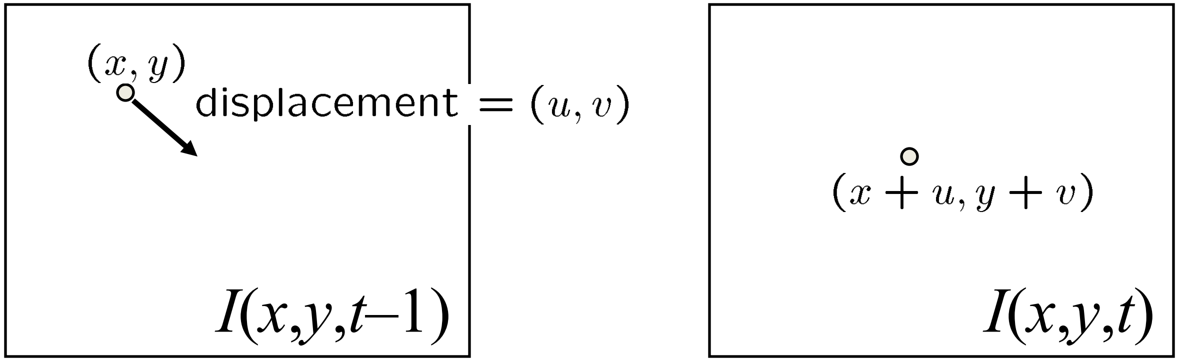
Image measurements (e.g. brightness) in a small region remain the same although their location may change.

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

(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:

Image derivative along x

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

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

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

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

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

I_x

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

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

I_y

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

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ second image

I_t

The brightness constancy constraint

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

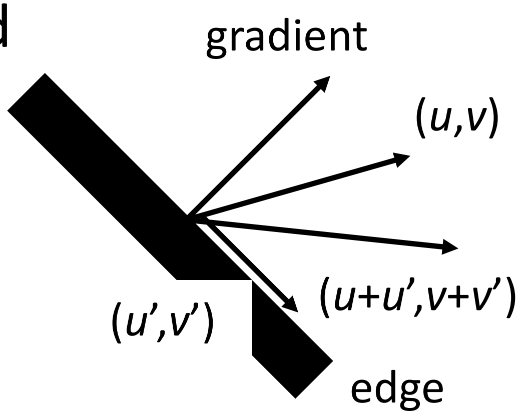
$$\nabla I \cdot [u \;\; v]^T + I_t = 0$$

- 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

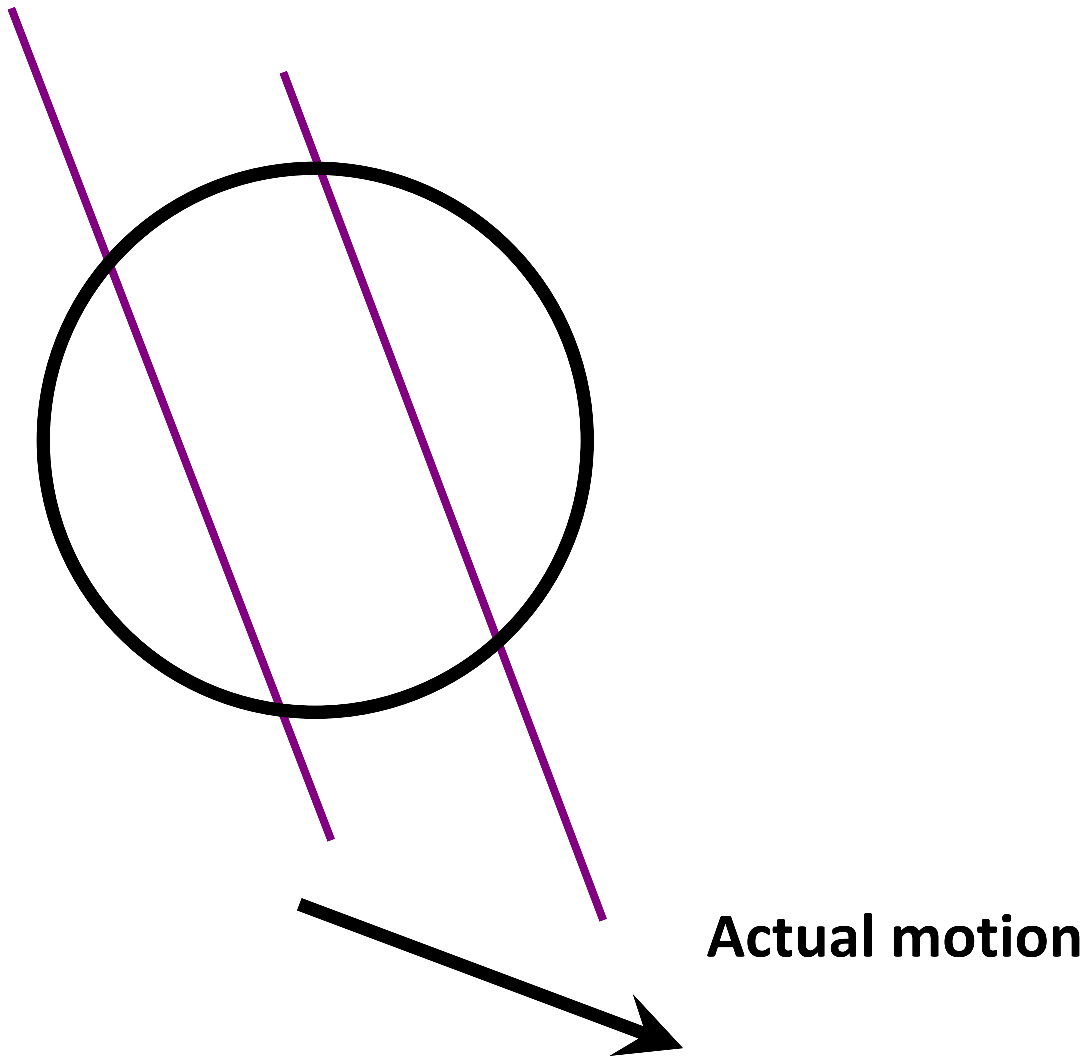
If (u, v) satisfies the equation, so does ($u+u'$, $v+v'$) if

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



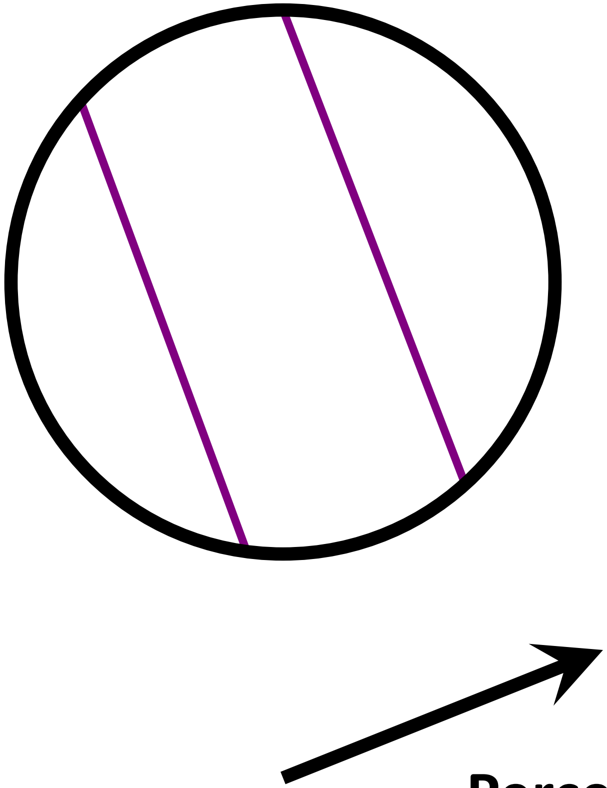
Source: Silvio Savarese

The aperture problem



Source: Silvio Savarese

The aperture problem



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

What we will learn today?

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Common fate
- 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

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

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

Source: Silvio Savarese

Lucas-Kanade flow

- Overconstrained linear system:

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

Source: Silvio Savarese

Lucas-Kanade flow

- Overconstrained linear system

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

$$\underset{25\text{x}2}{A} \quad \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

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

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

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

When is This Solvable?

- $\mathbf{A^T A}$ should be invertible
- $\mathbf{A^T A}$ should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of $\mathbf{A^T A}$ should not be too small
- $\mathbf{A^T A}$ 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

M = A^{T}A is the *second moment matrix* !

(Harris corner detector…)

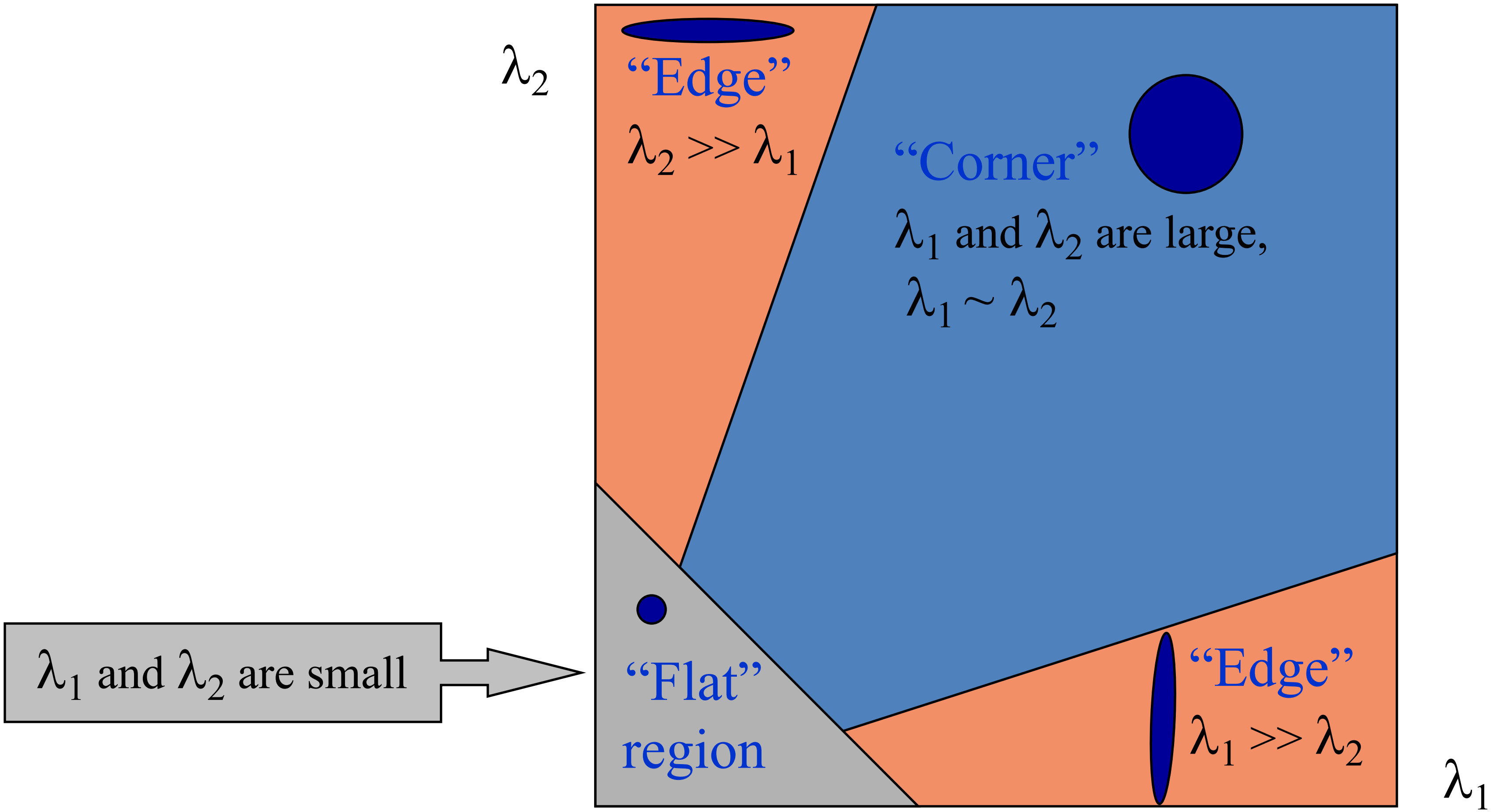
$$A^TA = \begin{bmatrix} \sum I_xI_x & \sum I_xI_y \\ \sum I_xI_y & \sum I_yI_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

- Eigenvectors and eigenvalues of A^{T}A 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

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

Improving accuracy

- Recall our small motion assumption

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

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

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

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

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

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

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

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, TV is based on illusory motion
   – the set is stationary yet things seem to move

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

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

4. Muscle movement can make some spots on a cheetah move opposite direction of motion.
   – And infinitely more break downs of optical flow.

What we will learn today?

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Common fate
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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

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:

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

- The first part of the function is the brightness consistency.

Horn-Schunk method for optical flow

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

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

- The second part is the smoothness constraint. It's trying to make sure that the changes between frames are small.

Horn-Schunk method for optical flow

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

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

- α is a regularization constant. Larger values of α lead to smoother flow.

Horn-Schunk method for optical flow

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

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

- 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. In practice, it is measured using:

$$\Delta u(x,y) = \overline{u}(x,y) - u(x,y)$$

- where $\overline{u}(x,y)$ is the weighted average of u measured at (x,y).

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 for each pixel individually.

Iterative Horn-Schunk

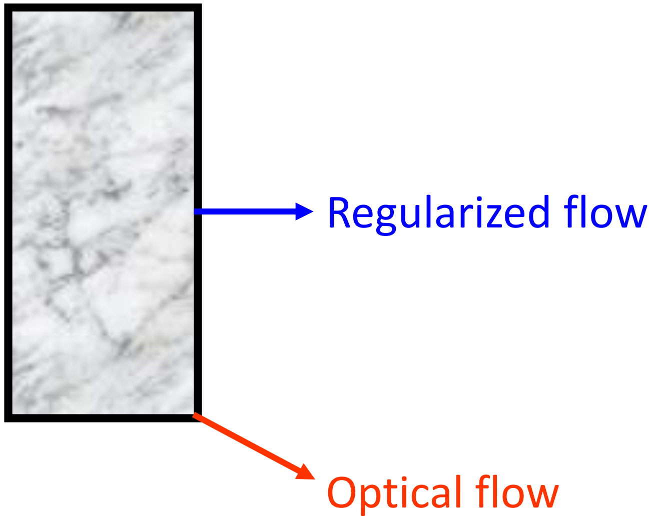
- But since the solution depends on the neighboring values of the flow field, it must be repeated once the neighbors have been updated.
- So instead, we can iteratively solve for u and v using:

$$u^{k+1} = \overline{u}^k - \frac{I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

$$v^{k+1} = \overline{v}^k - \frac{I_y(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

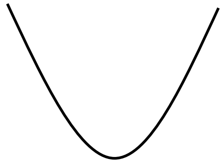
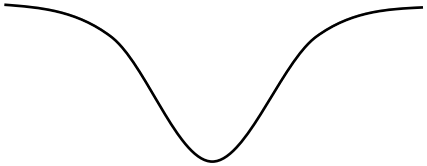
What does the smoothness regularization do anyway?

- It's a sum of squared terms (a Euclidian distance measure).
- We're putting it in the expression to be minimized.
- => In texture free regions, *there is no optical flow*
- => On edges, points will flow to nearest points, solving the aperture problem.



Slide credit: Sebastian Thurn

Dense Optical Flow with Michael Black's method

- Michael Black took Horn-Schunk's method one step further, starting from the regularization constant: $\|\nabla u\|^2 + \|\nabla v\|^2$
- Which looks like a quadratic: 
- And replaced it with this: 
- Why does this regularization work better?

What we will learn today?

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Common fate
- Applications

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

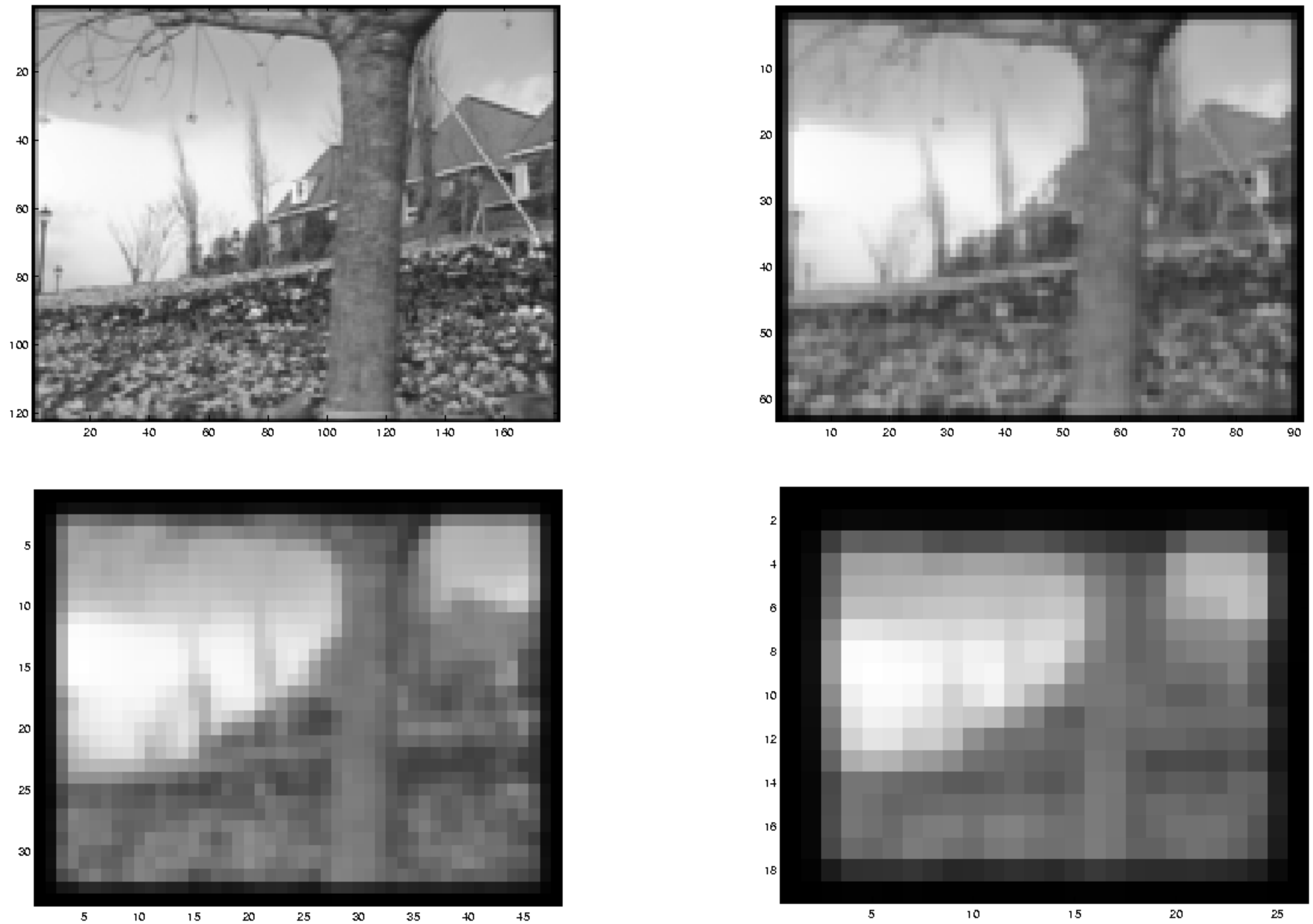
Source: Silvio Savarese

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?

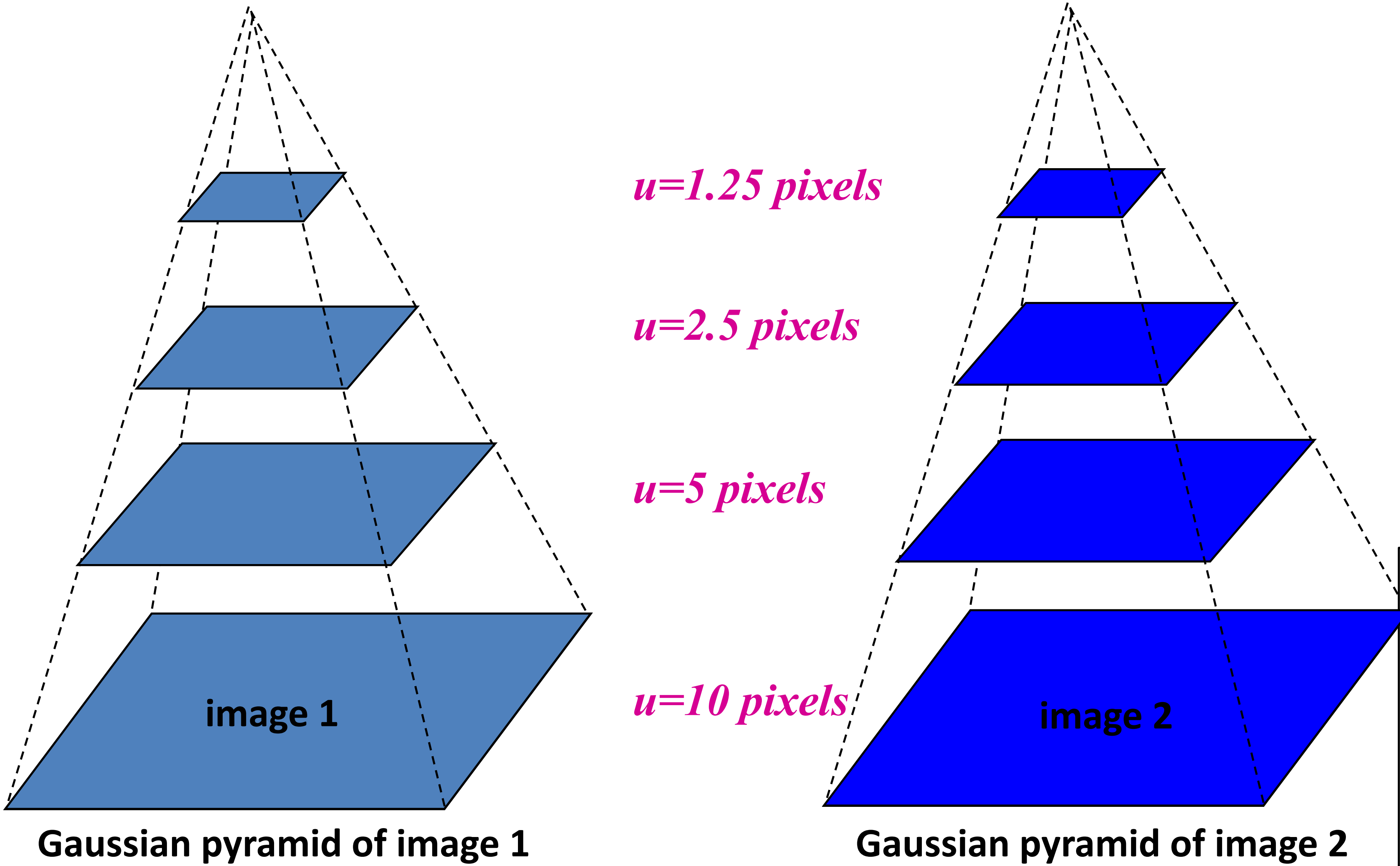
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Reduce the resolution!



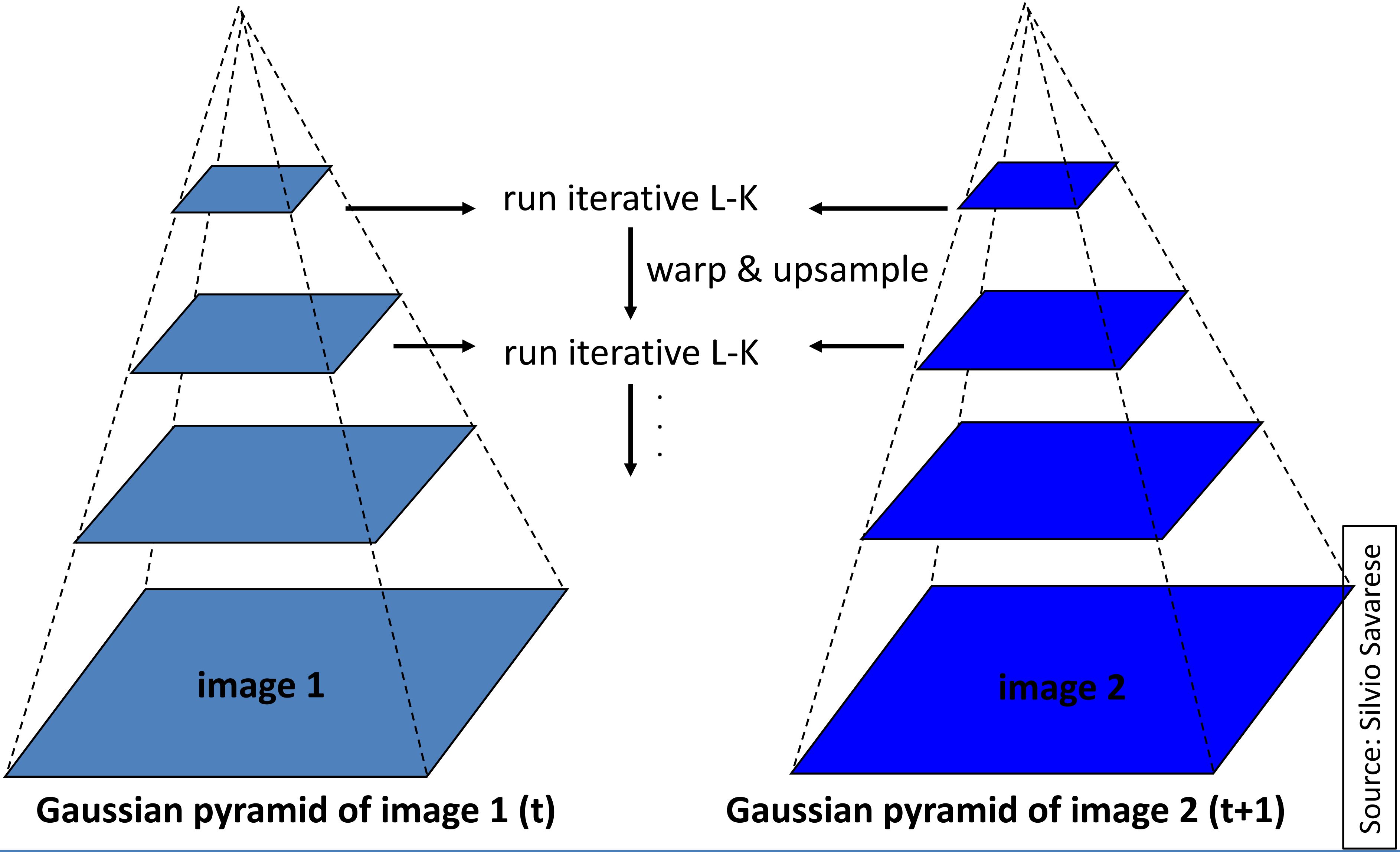
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Coarse-to-fine optical flow estimation

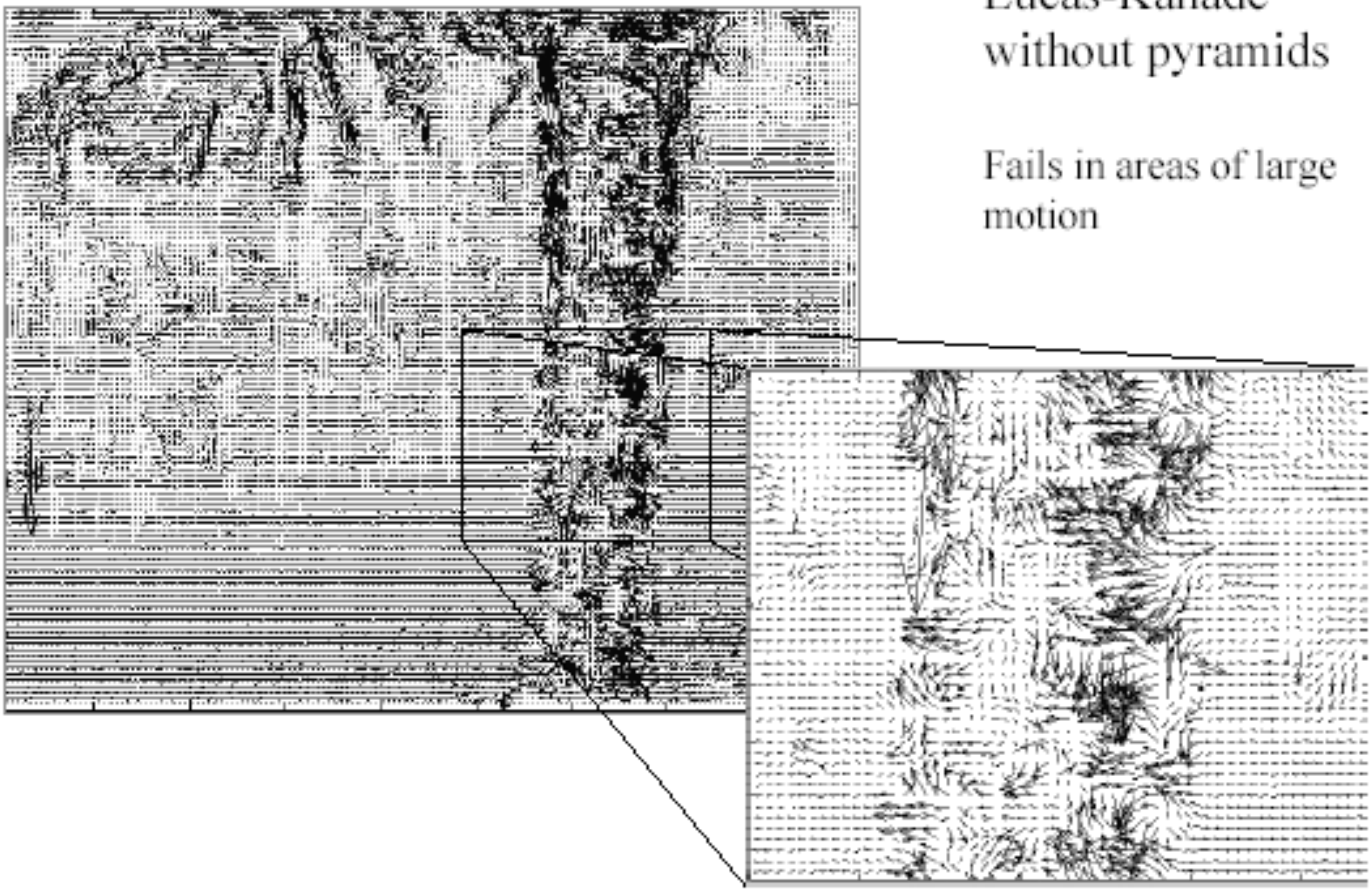


Source: Silvio Savarese

Coarse-to-fine optical flow estimation



Optical Flow Results

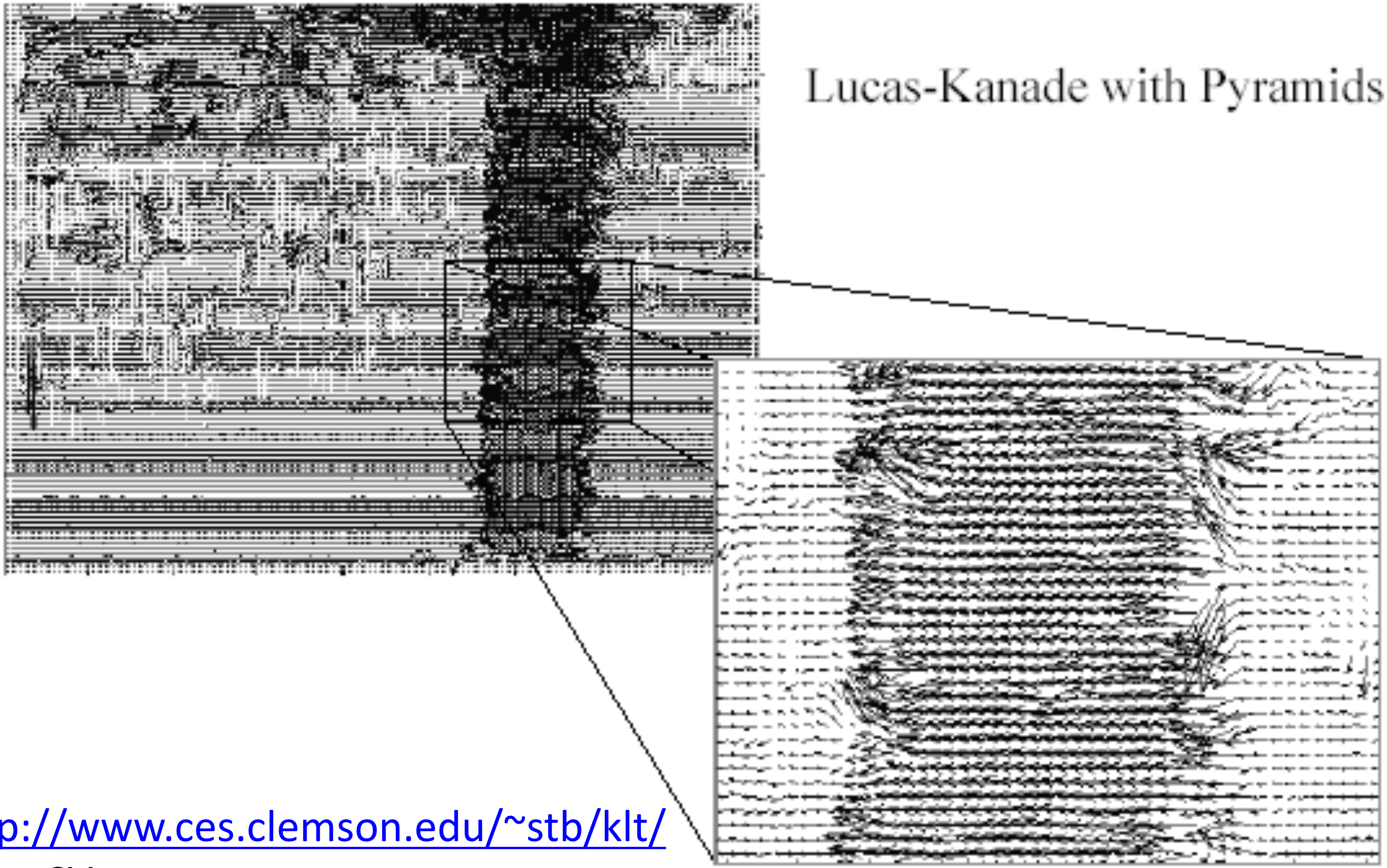


Lucas-Kanade
without pyramids

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

What we will learn today?

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Common fate
- Applications

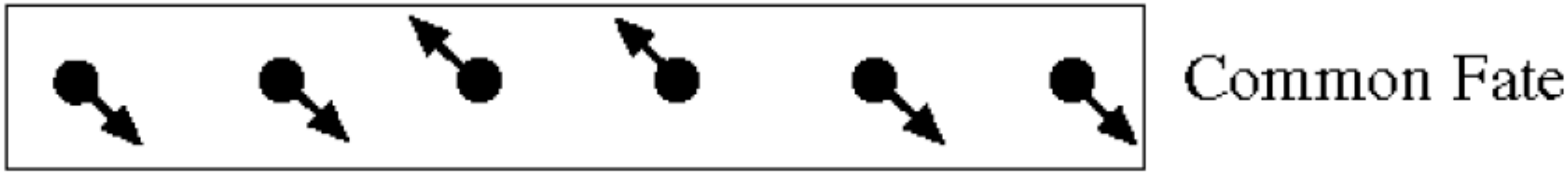
Recap

- 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

Reminder: Gestalt – common fate



Motion segmentation

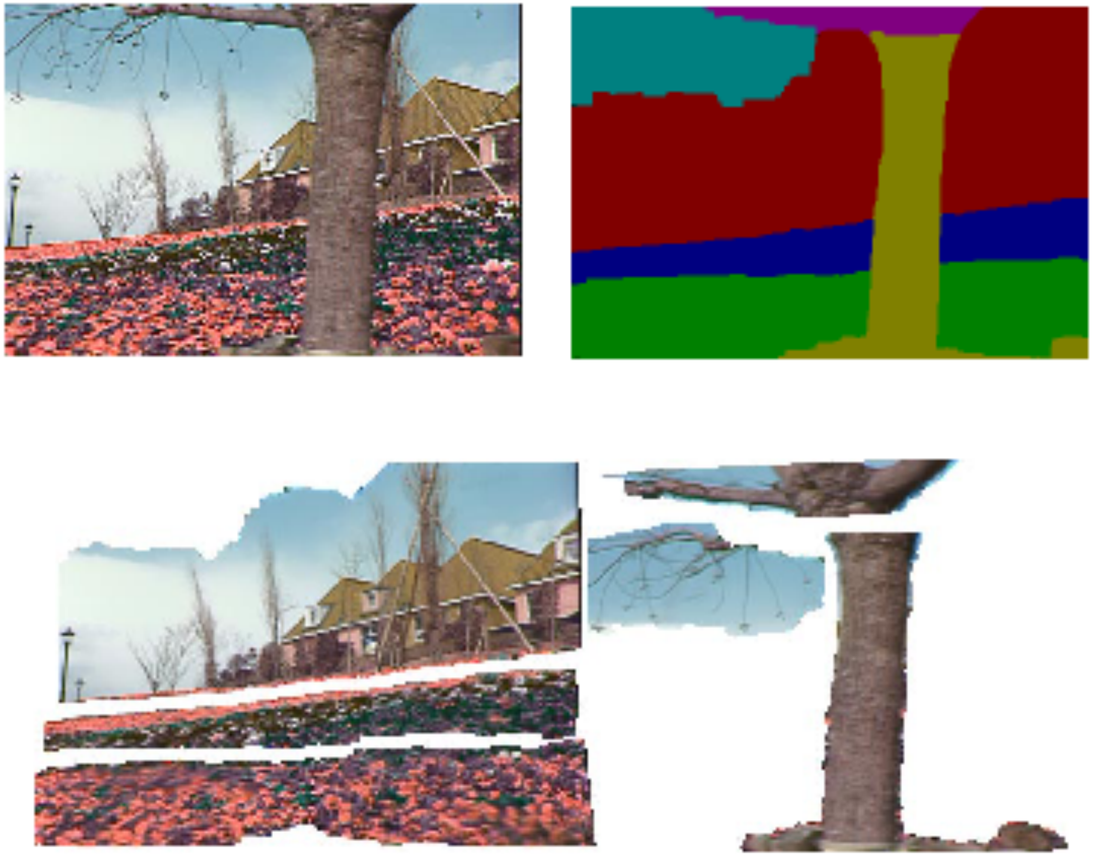
- How do we represent the motion in this scene?

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

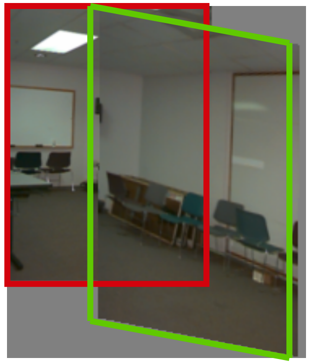


Source: Silvio Savarese

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:

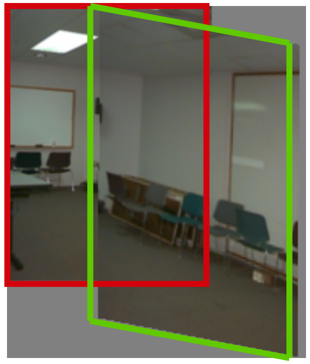
$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

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:

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

- Each pixel provides 1 linear constraint in 6 unknowns

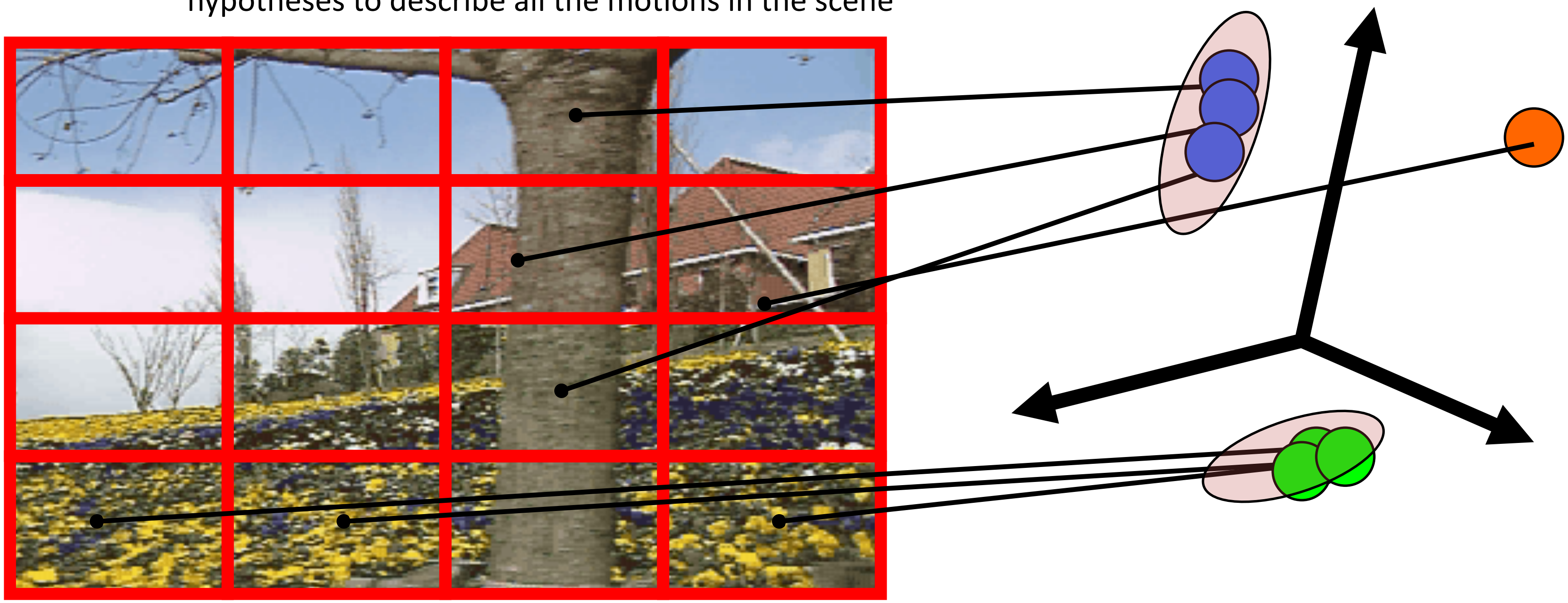
- Least squares minimization:

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

Source: Silvio Savarese

How do we estimate the layers?

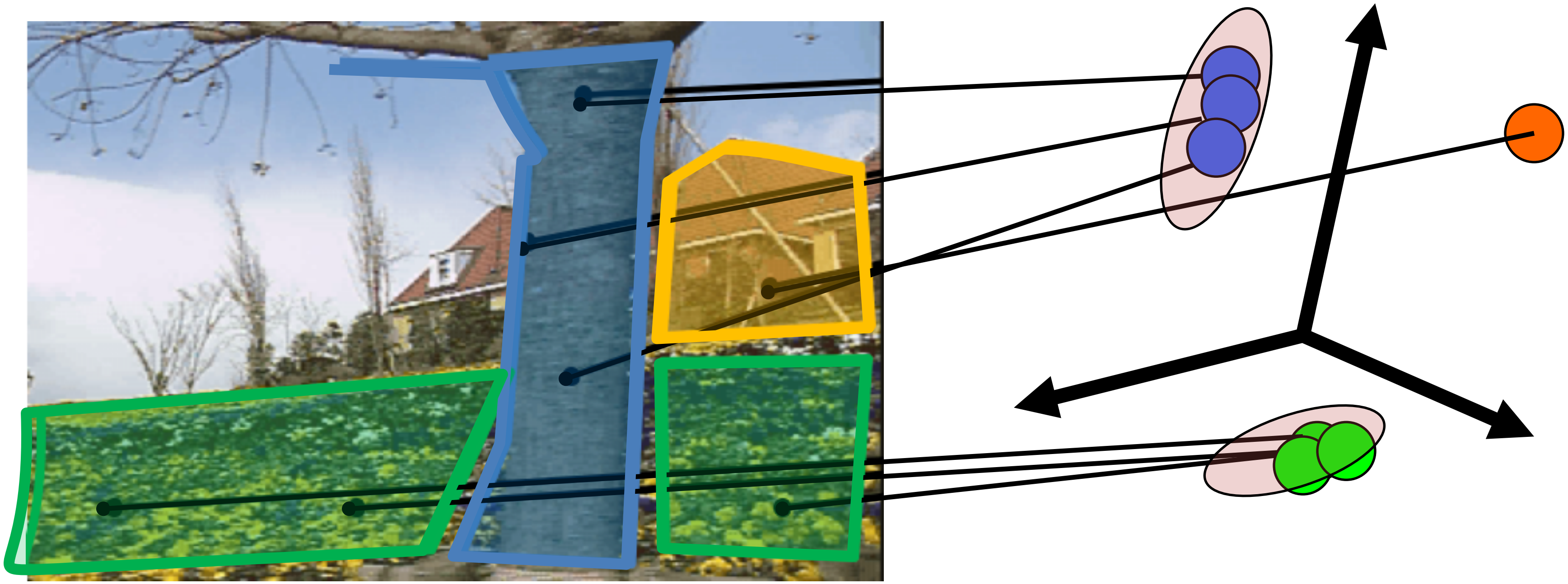
- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error
- Map into motion parameter space
- Perform k-means clustering on affine motion parameters
  - 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



Source: Silvio Savarese

How do we estimate the layers?

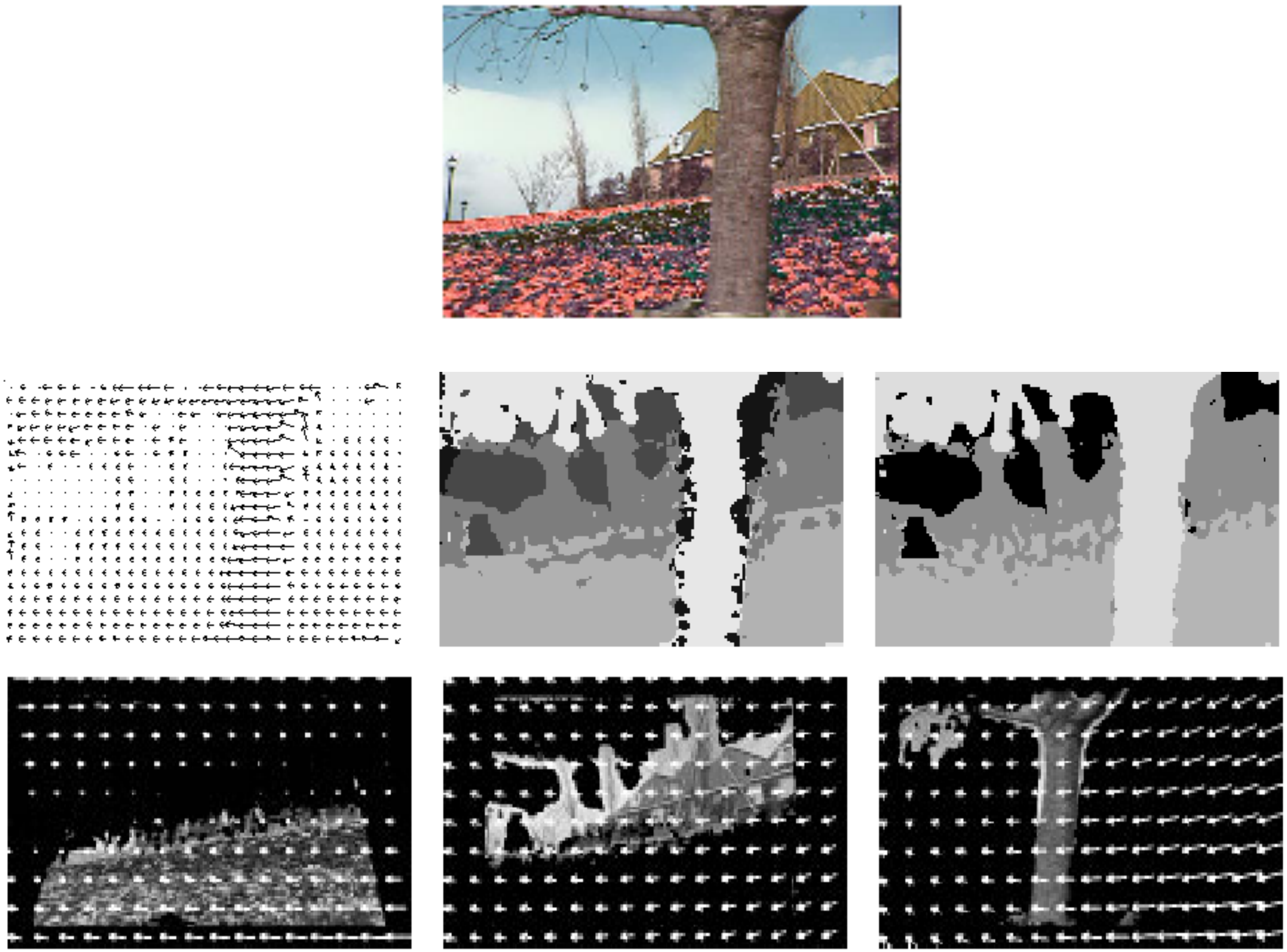
- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error
  - Map into motion parameter space
  - Perform k-means clustering on affine motion parameters
    - –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

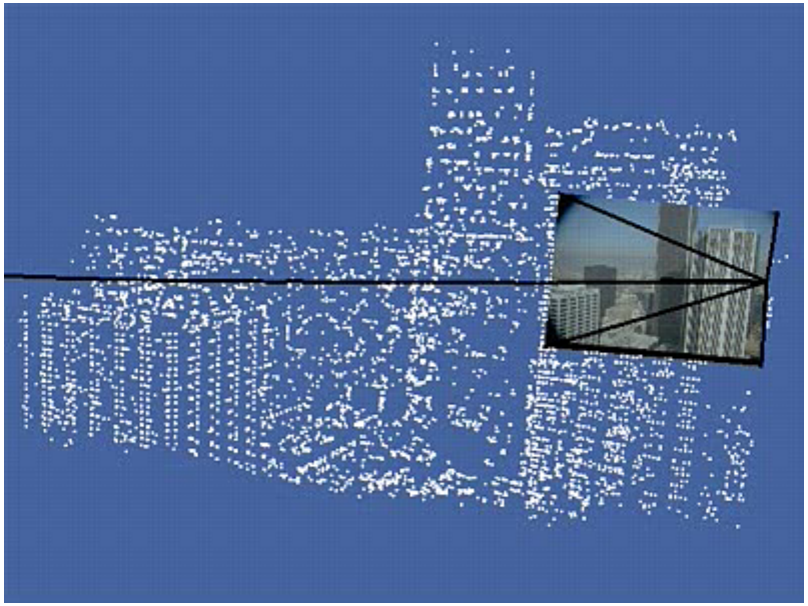
What we will learn today?

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Common fate
- 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

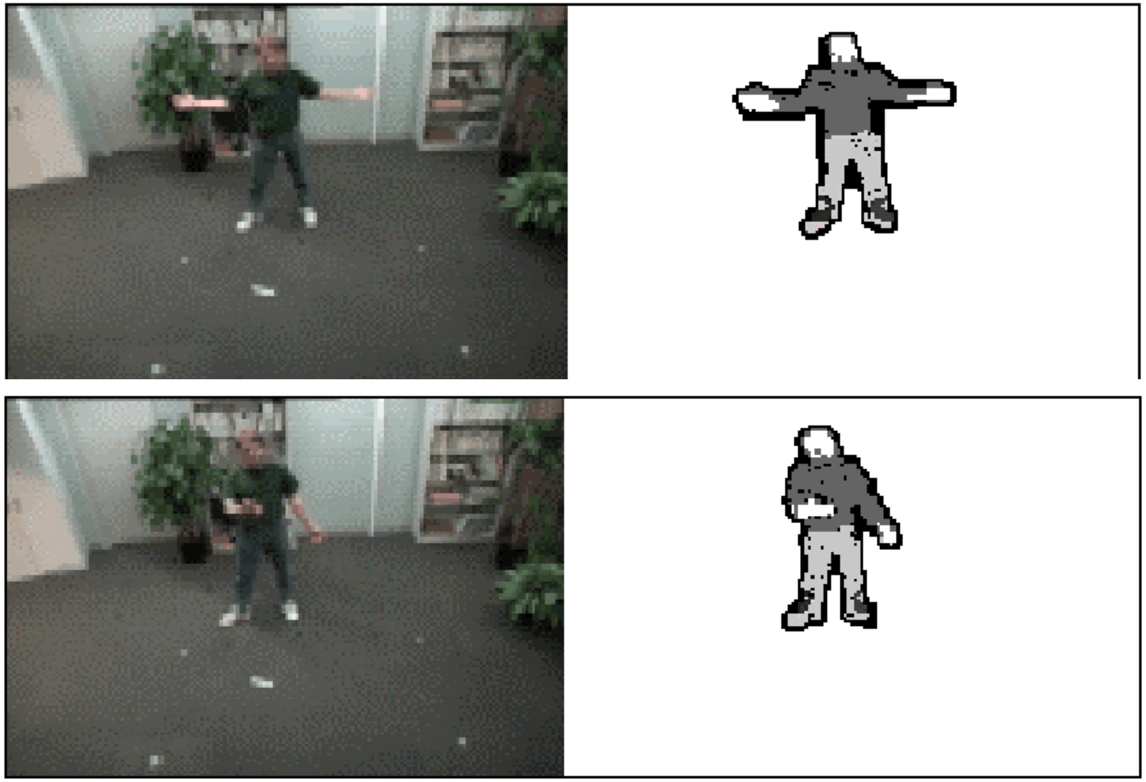
Estimating 3D structure



Source: Silvio Savarese

Segmenting objects based on motion cues

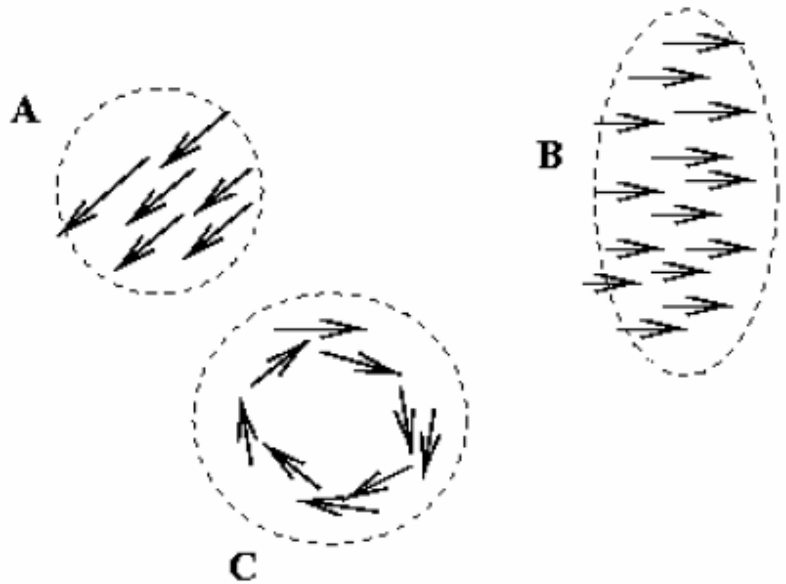
- Background subtraction
  - A static camera is observing a scene
  - Goal: separate the static *background* from the moving *foreground*

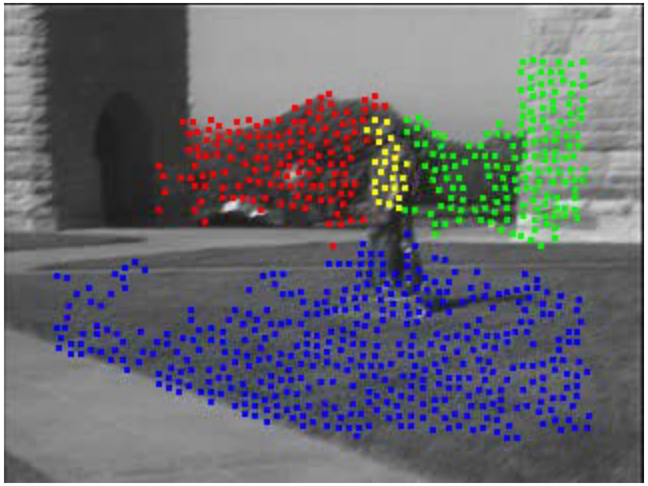


Source: Silvio Savarese

Segmenting objects based on motion cues

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

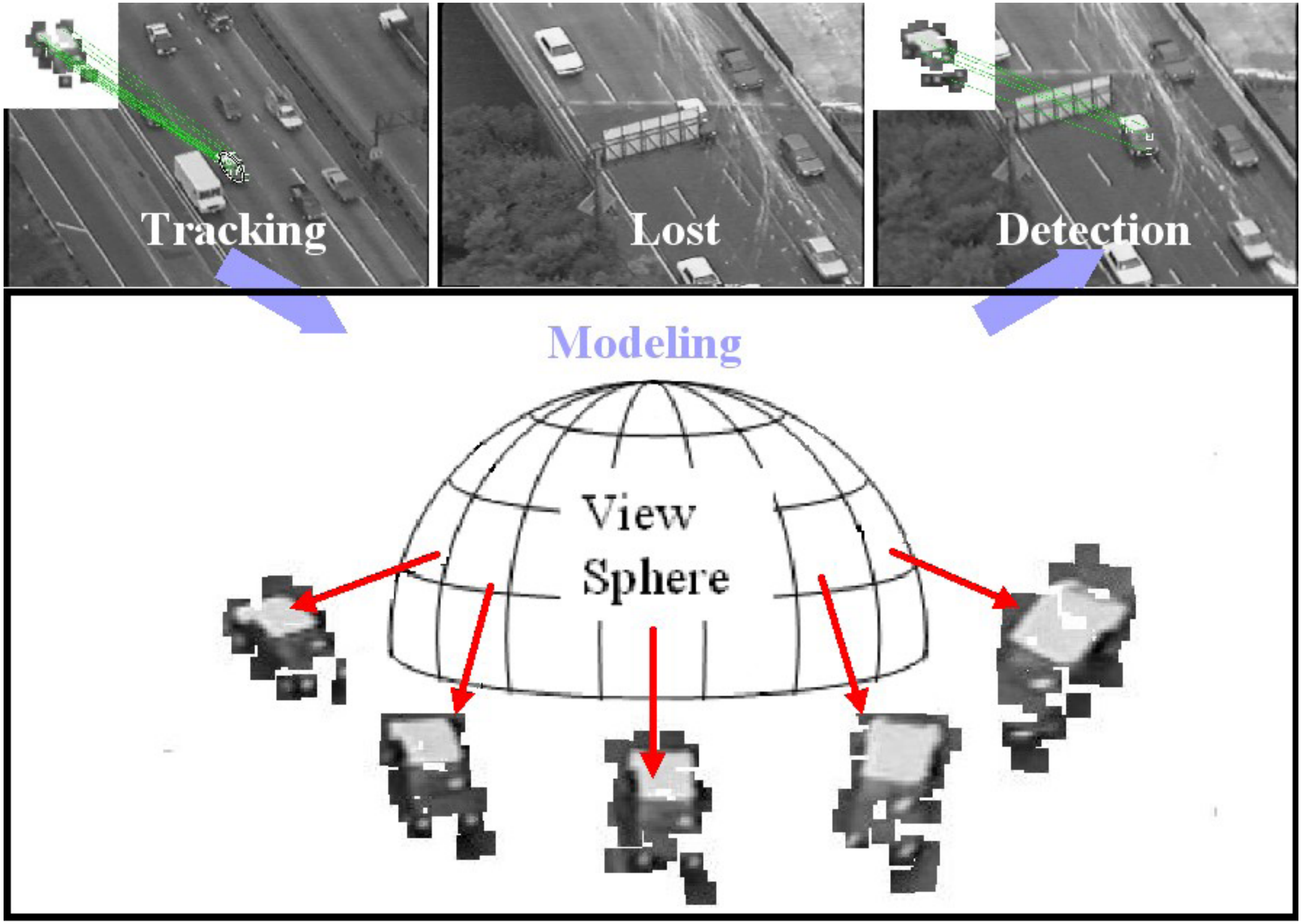




S. J. Pundlik and S. T. Birchfield, Motion Segmentation at Any Speed, Proceedings of the British Machine Vision Conference (BMVC) 2006

Source: Silvio Savarese

Tracking objects



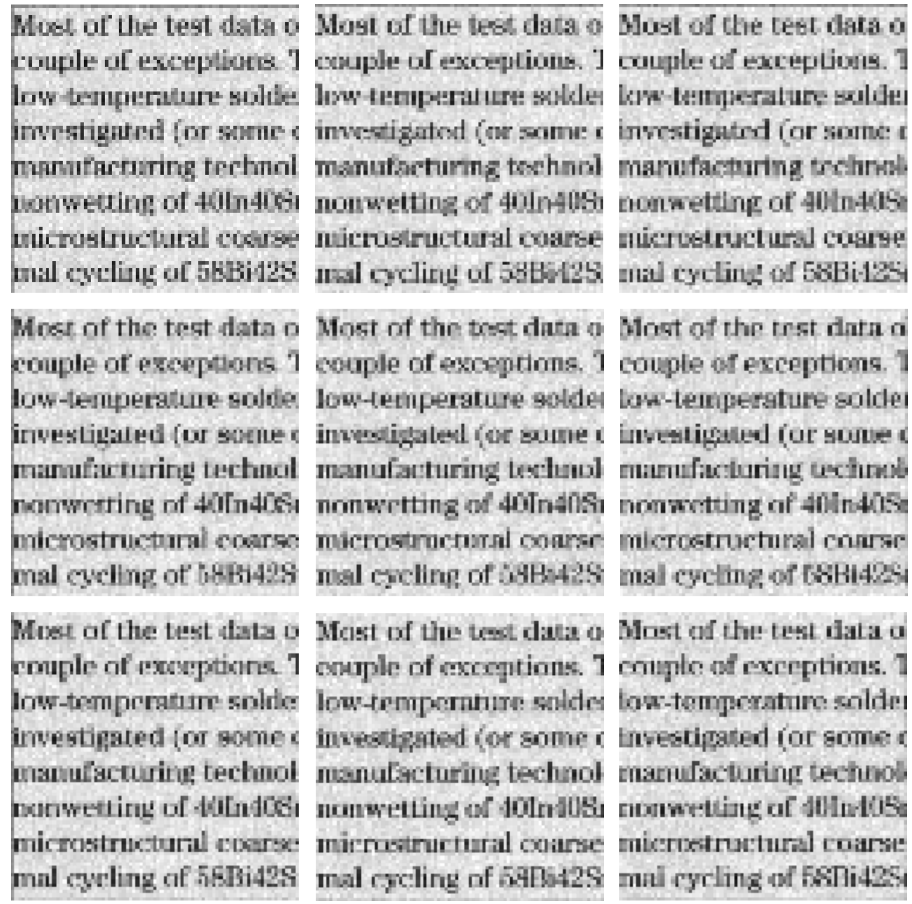
Z.Yin and R.Collins, "On-the-fly Object Modeling while Tracking," *IEEE Computer Vision and Pattern Recognition (CVPR '07),* Minneapolis, MN, June 2007.

Source: Silvio Savarese

Synthesizing dynamic textures

Super-resolution

Example: A set of low quality images



Source: Silvio Savarese

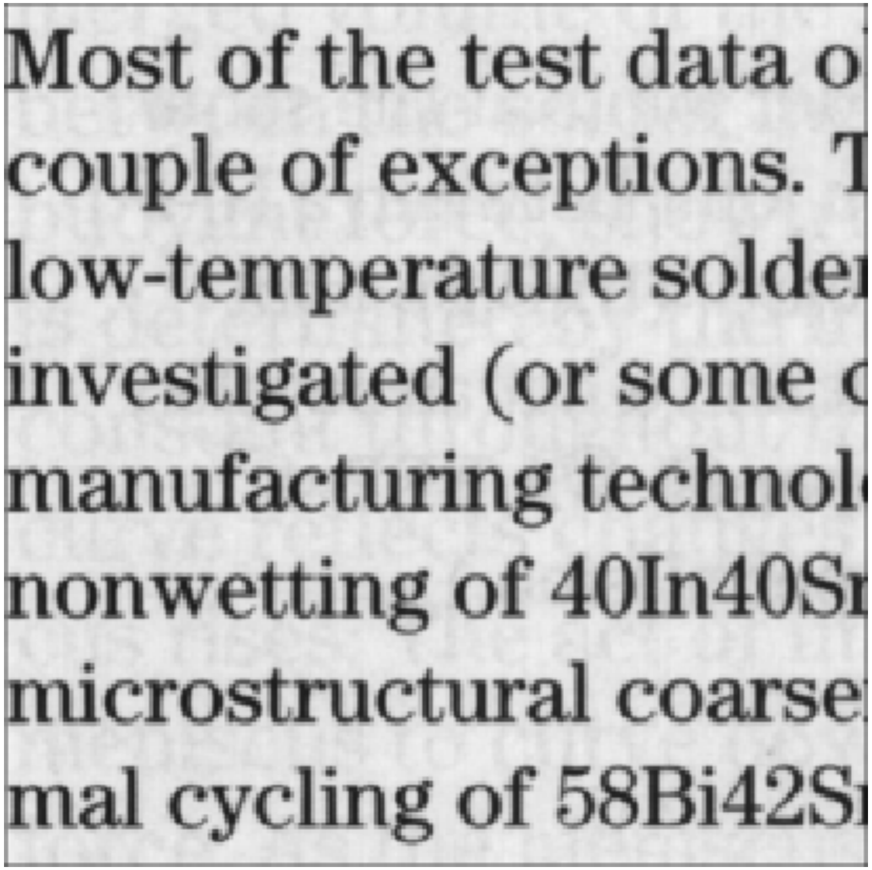
Super-resolution

Each of these images looks like this:

Source: Silvio Savarese

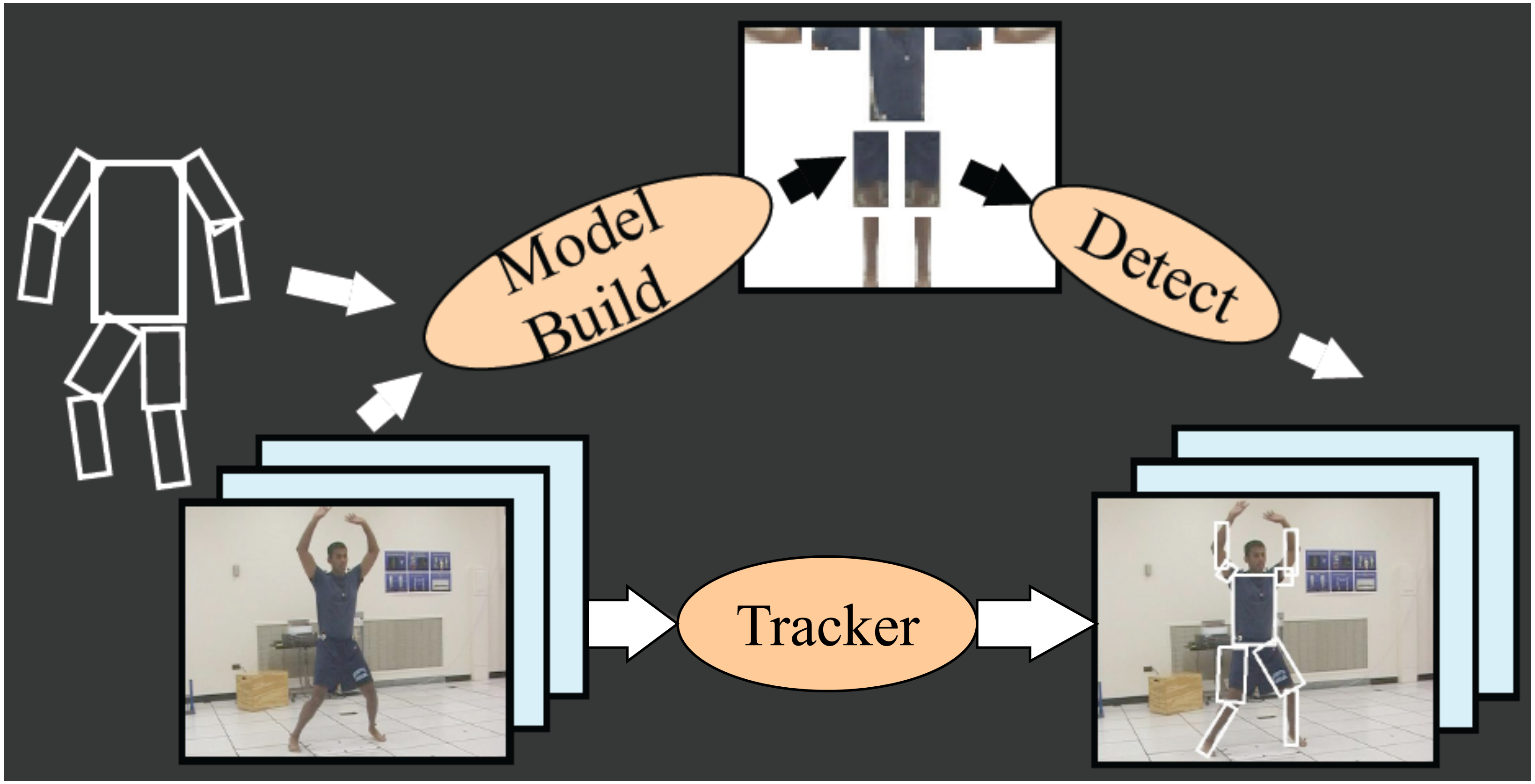
Super-resolution

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

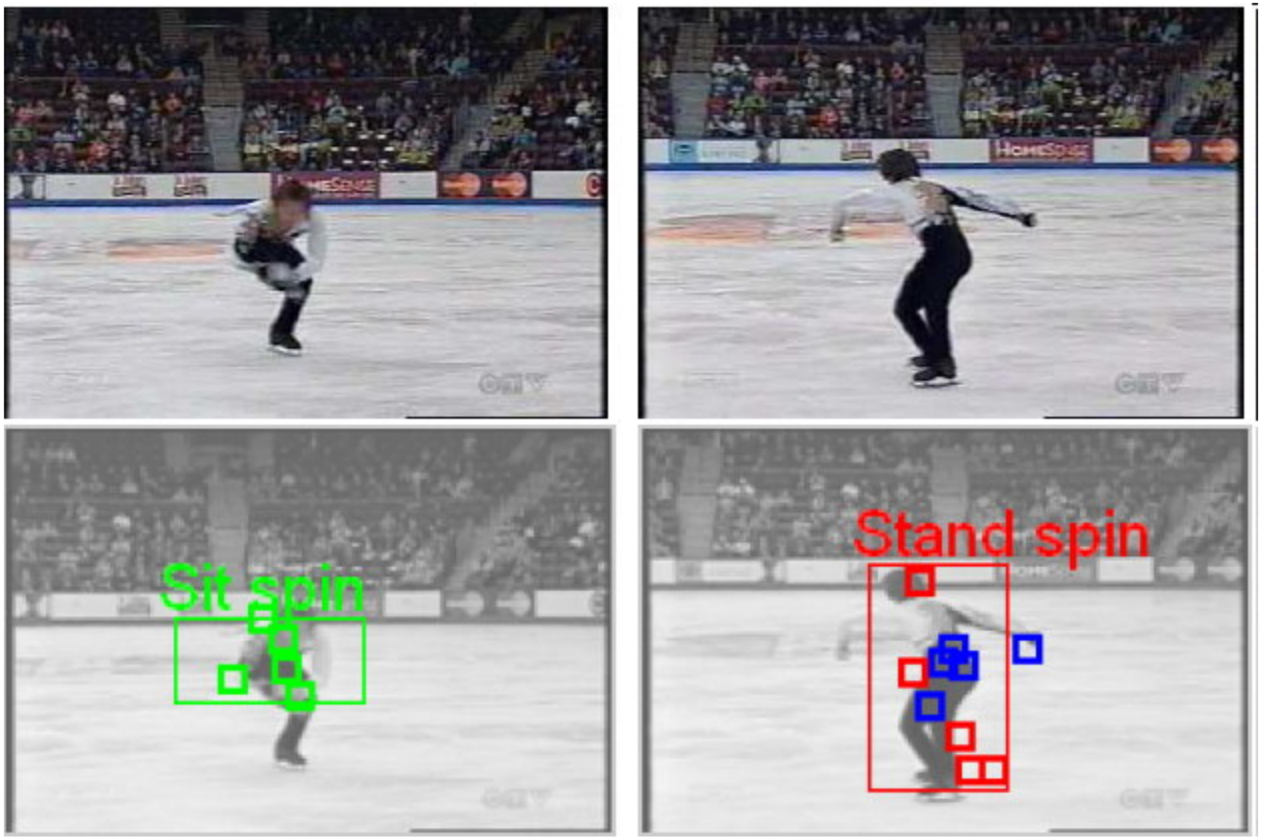
Recognizing events and activities



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

Source: Silvio Savarese

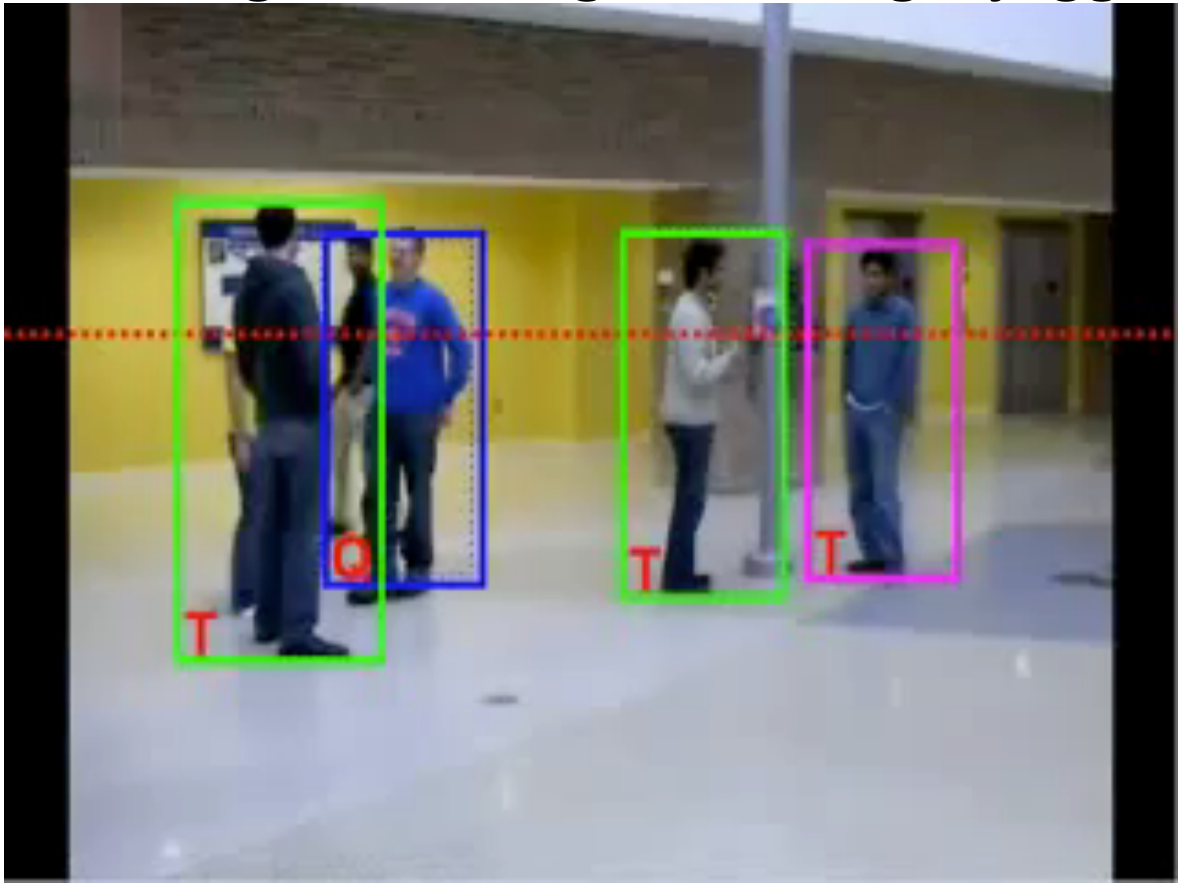
Recognizing events and activities



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

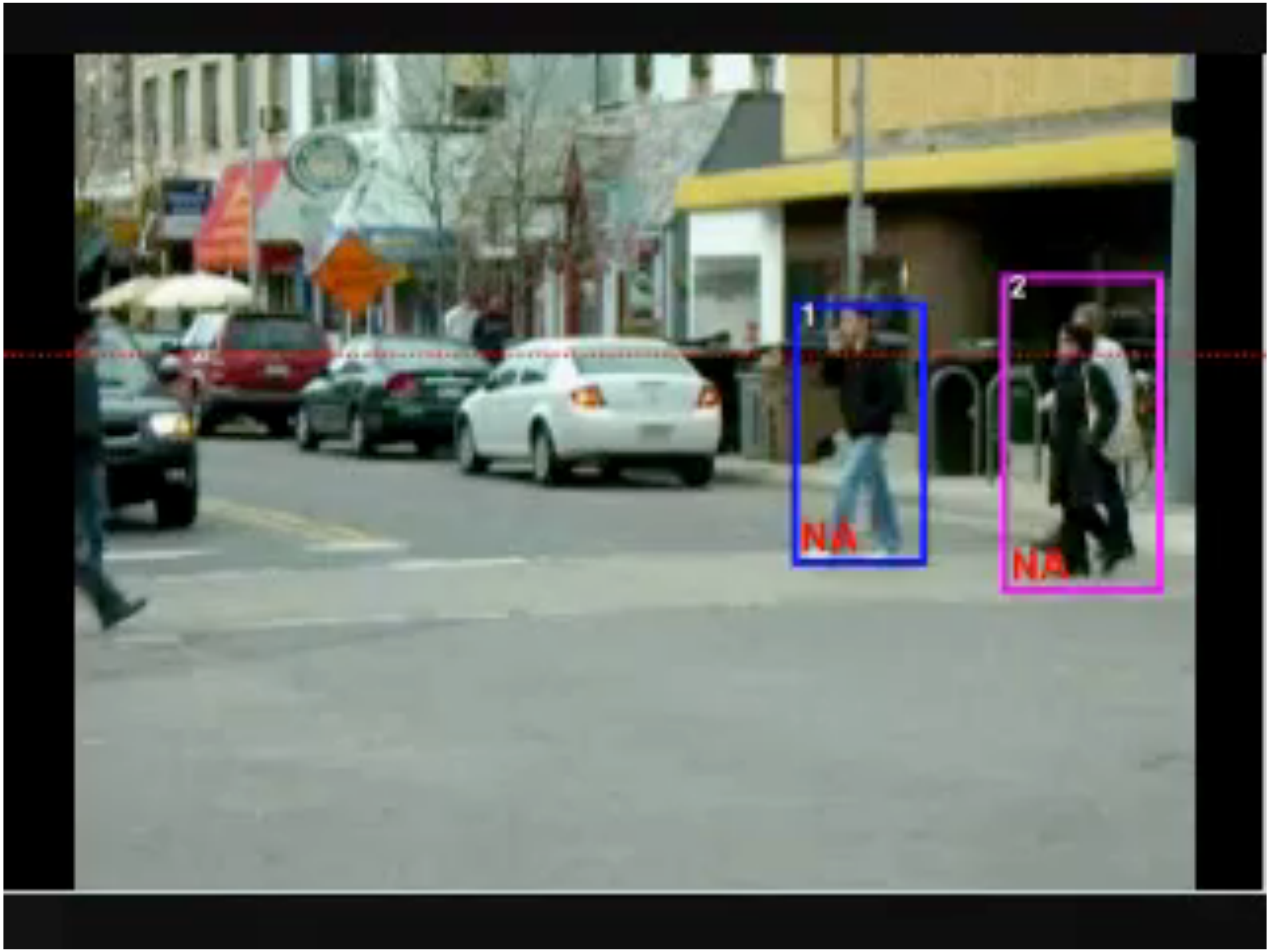
Recognizing events and activities

Crossing – Talking – Queuing – Dancing – jogging



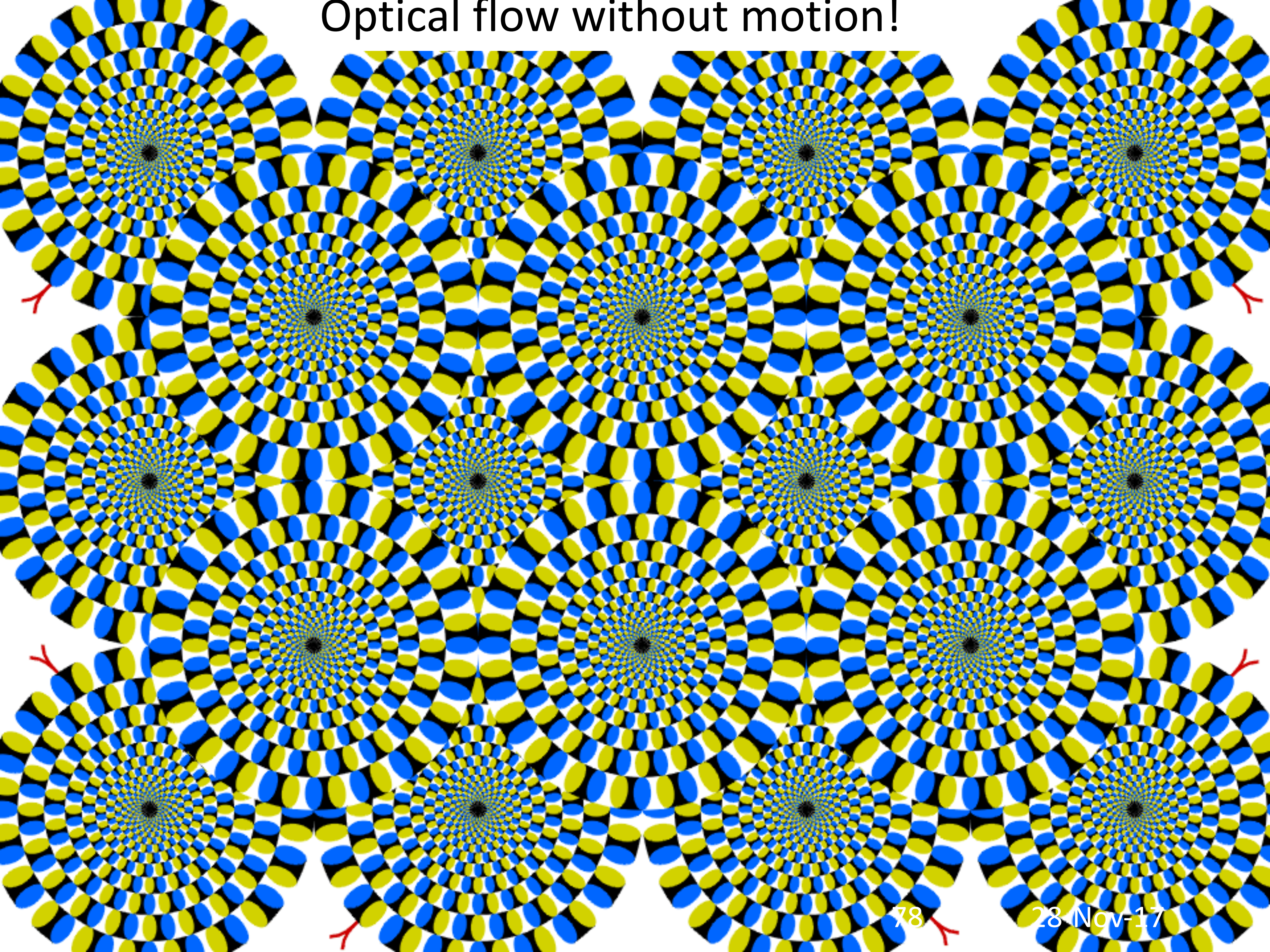
W. Choi & K. Shahid & S. Savarese WMC 2010

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?

- Optical flow
- Lucas-Kanade method
- Horn-Schunk method
- Pyramids for large motion
- Common fate
- Applications

Reading: [Szeliski] Chapters: 8.4, 8.5

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