CEE598 - Visual Sensing for Civil Infrastructure Eng. & Mgmt.

Session 17 – Optical Flow and Tracking

Mani Golparvar-Fard
Department of Civil and Environmental Engineering
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu

Tracking

- Pattie Maes- MIT Media Lab – Six Sense

http://www.youtube.com/watch?v=nZ-VjUKAsao&feature=player_embedded#at=290

- Vehicle/Pedestrian Tracking:

http://www.youtube.com/watch?v=vQ-3dApJi2s&feature=related

- Hokey players:

http://www.youtube.com/watch?v=jGMUQwWTjdM&feature=related

- Soccer players:

http://www.youtube.com/watch?v=fRowYlxKt7s&feature=related

Outline

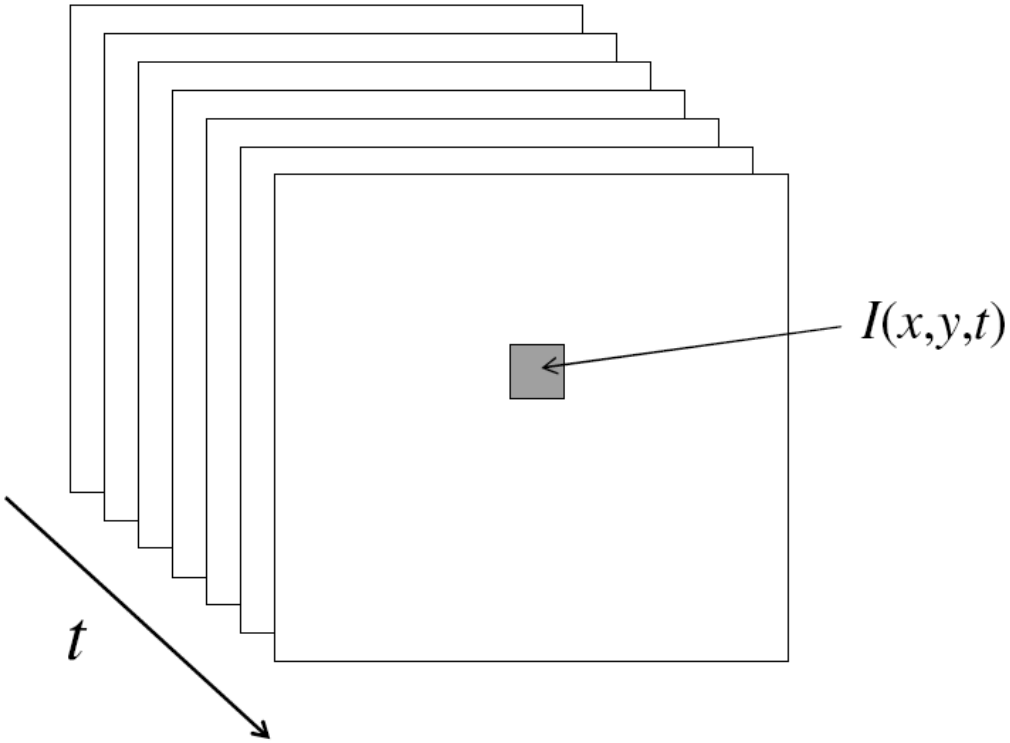
- Optical Flow and Tracking

- Introduction
- Lucas-Kanade algorithm
- Motion segmentation
- Tracking

Some slides of this lectures are courtesy of prof S. Savarese, prof S. Lazebnik & S. Seitz, R. Szeliski, M. Pollefeys, K. Hassan-Shafique. S. Thrun

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)

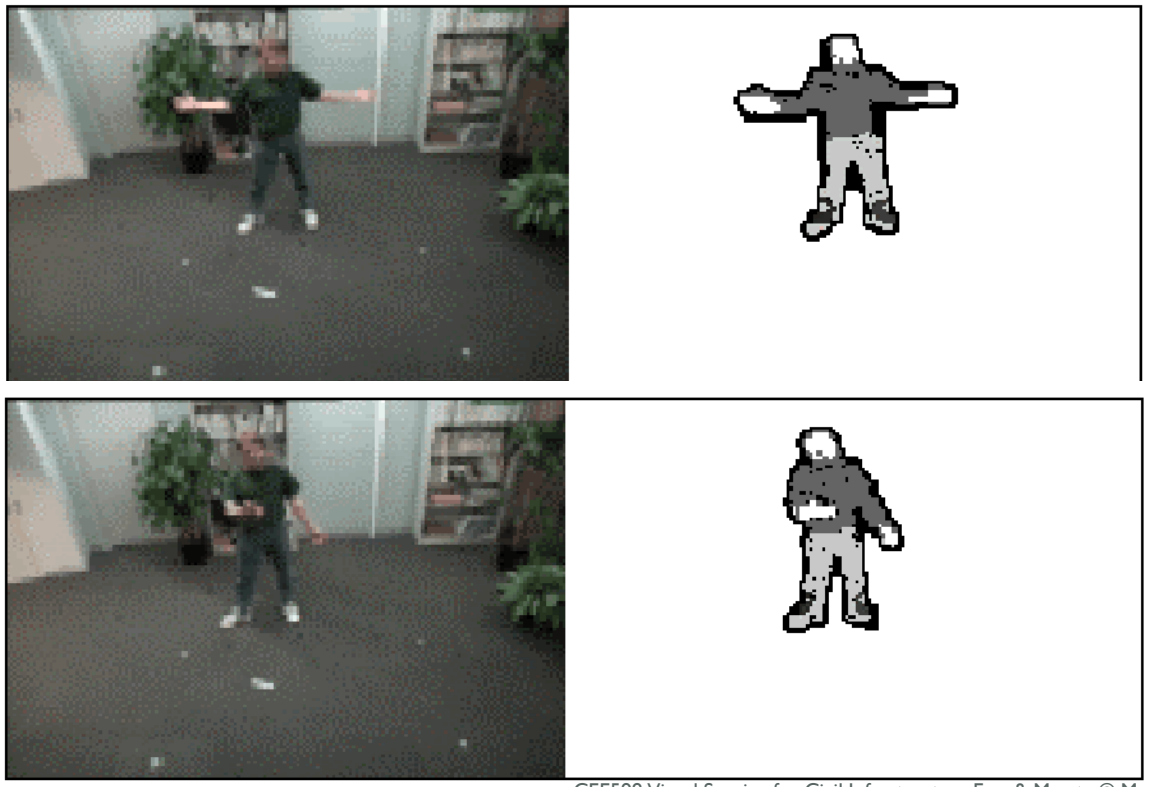


Uses of motion

- Estimating 3D structure
- Tracking objects
- Segmenting objects based on motion cues
- Learning dynamical models
- Recognizing events and activities
- Improving video quality (motion stabilization)

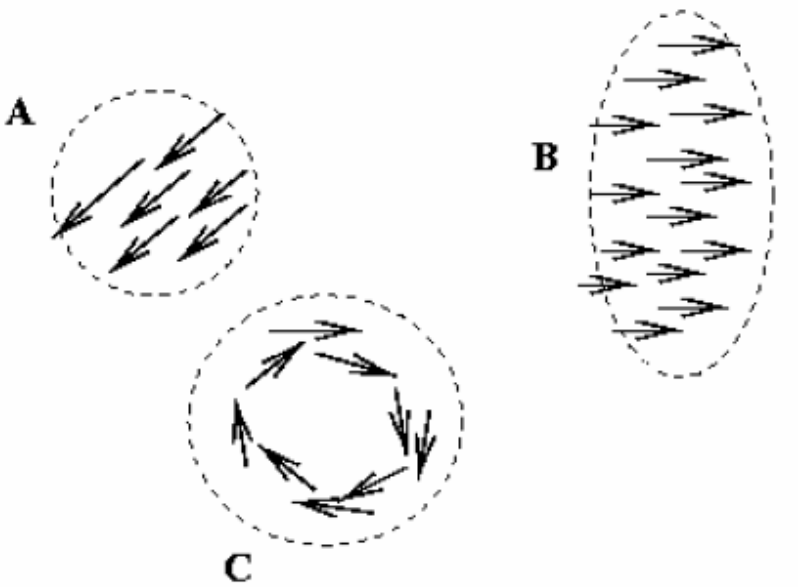
Segmenting objects based on motion cues

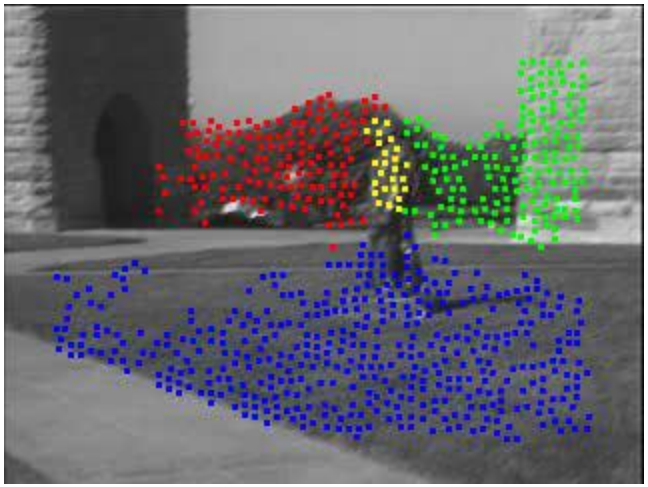
- Background subtraction
  - A static camera is observing a scene
  - Goal: separate the static *background* from the moving *foreground*



Segmenting objects based on motion cues

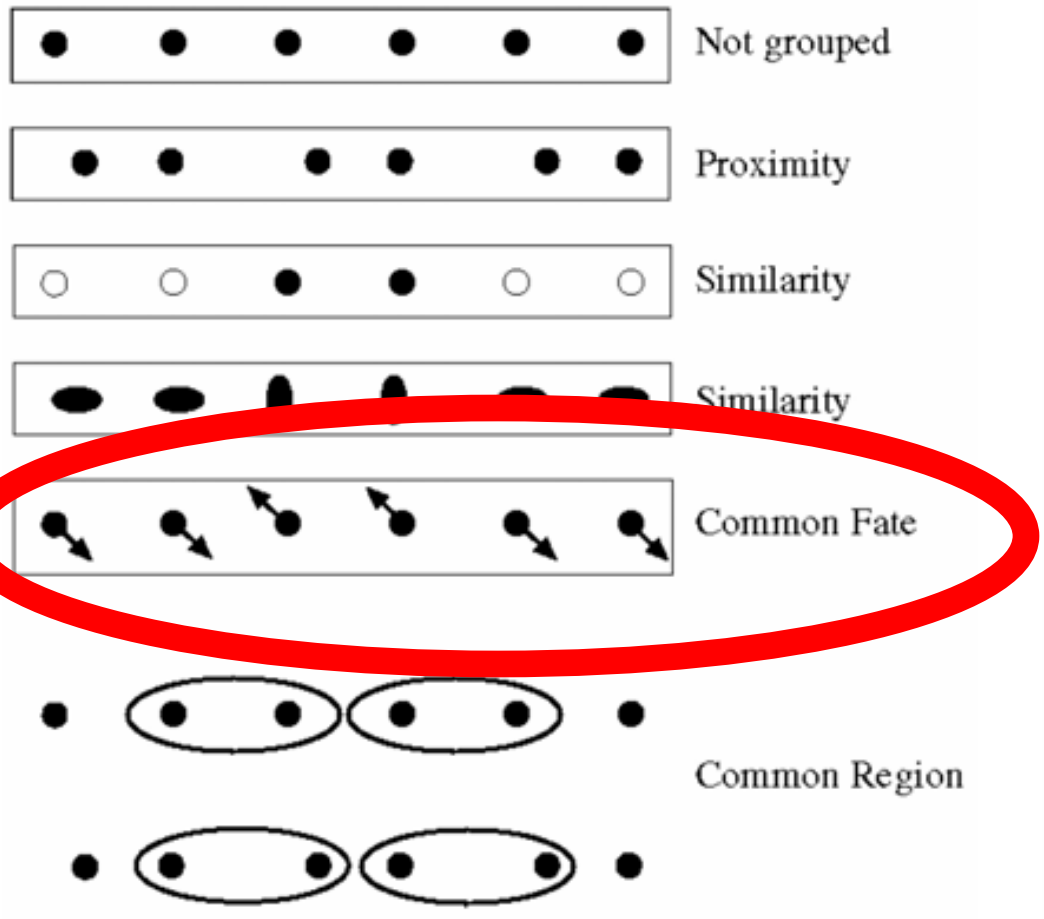
- Motion segmentation
  - Segment the video into multiple *coherently* moving objects

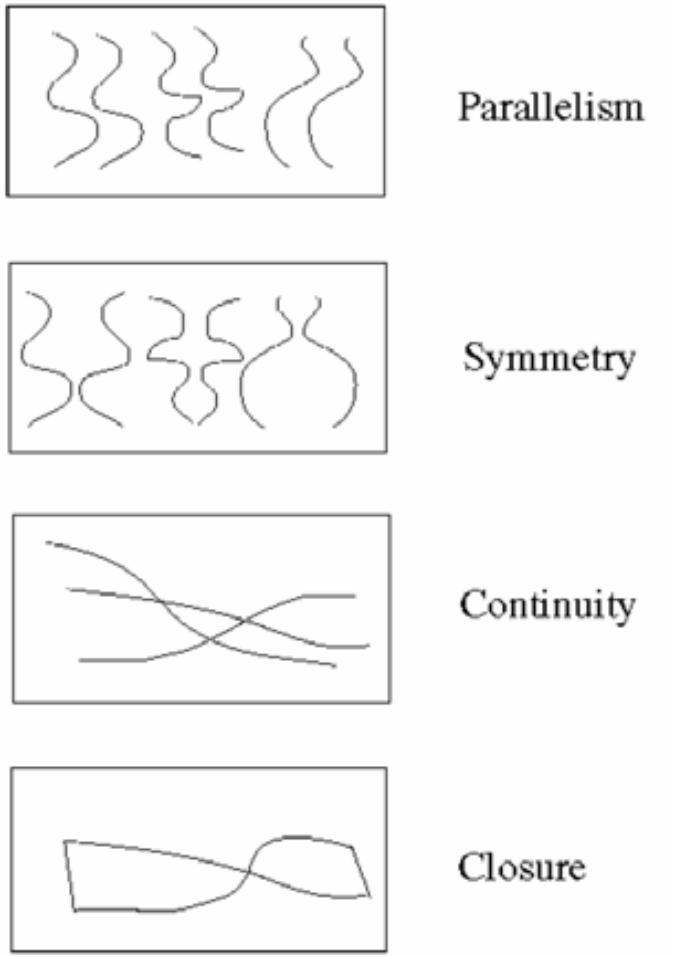




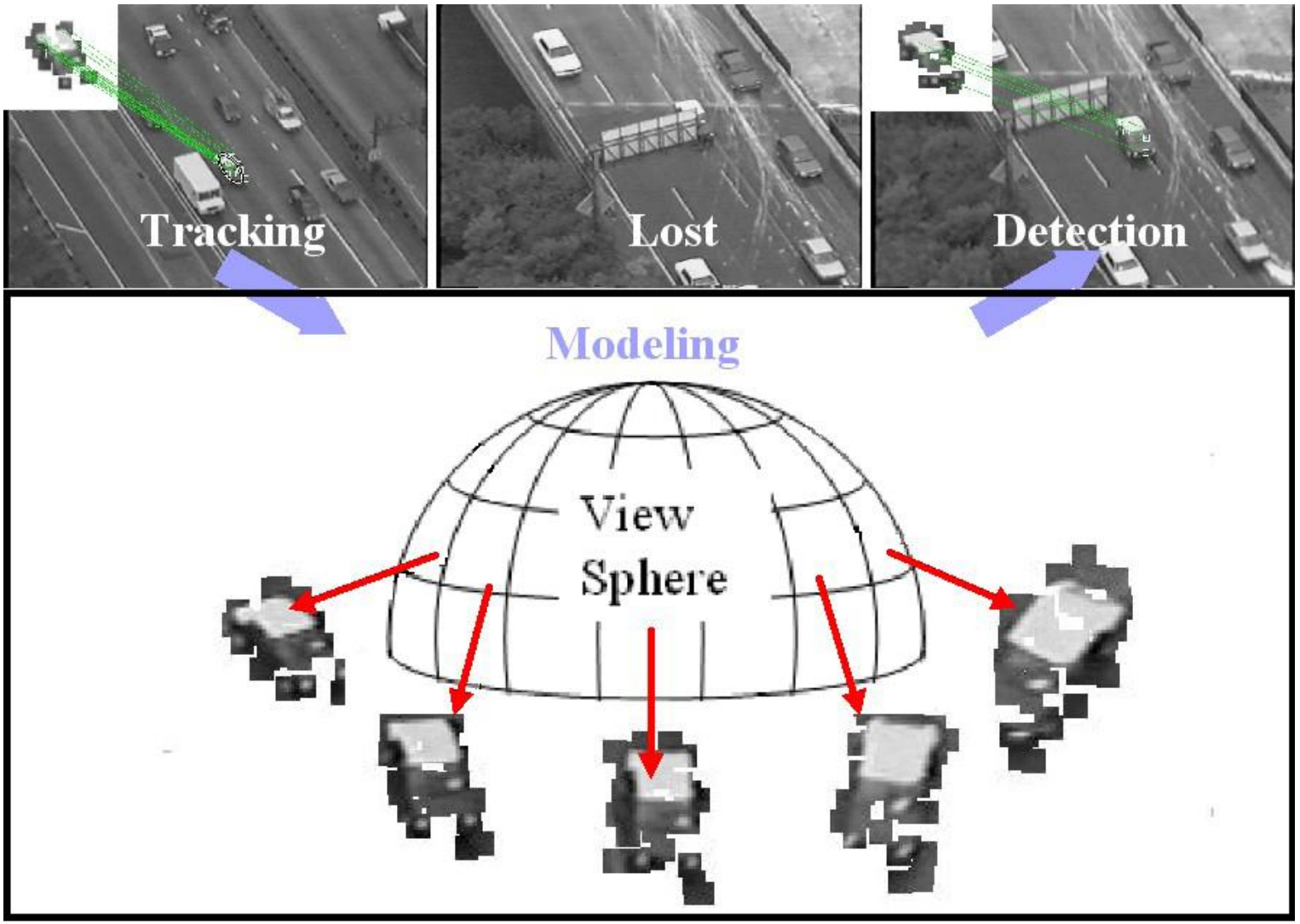
S. J. Pundlik and S. T. Birchfield, Motion Segmentation at Any Speed, Proceedings of the British Machine Vision Conference 06

Motion and perceptual organization



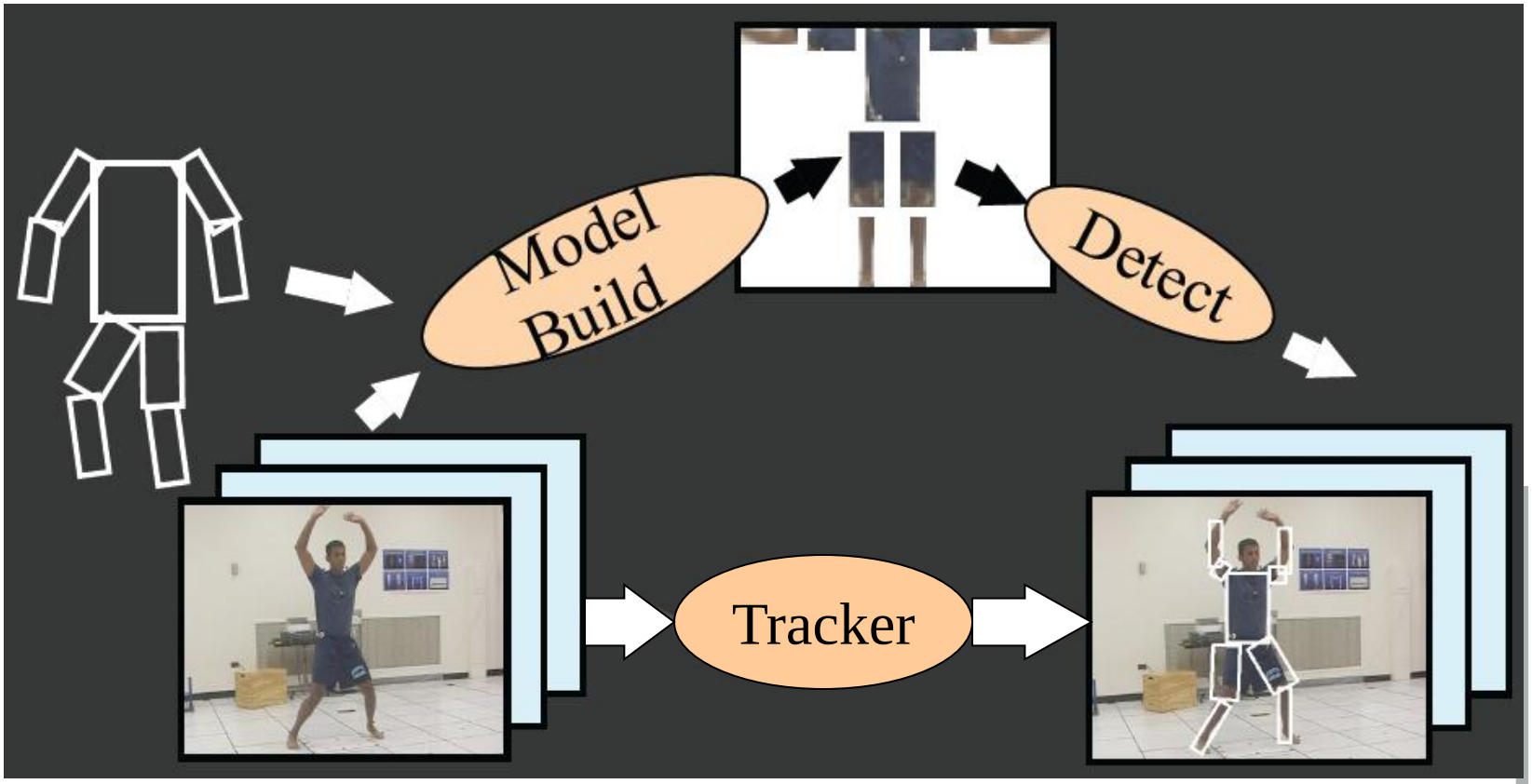


Tracking objects



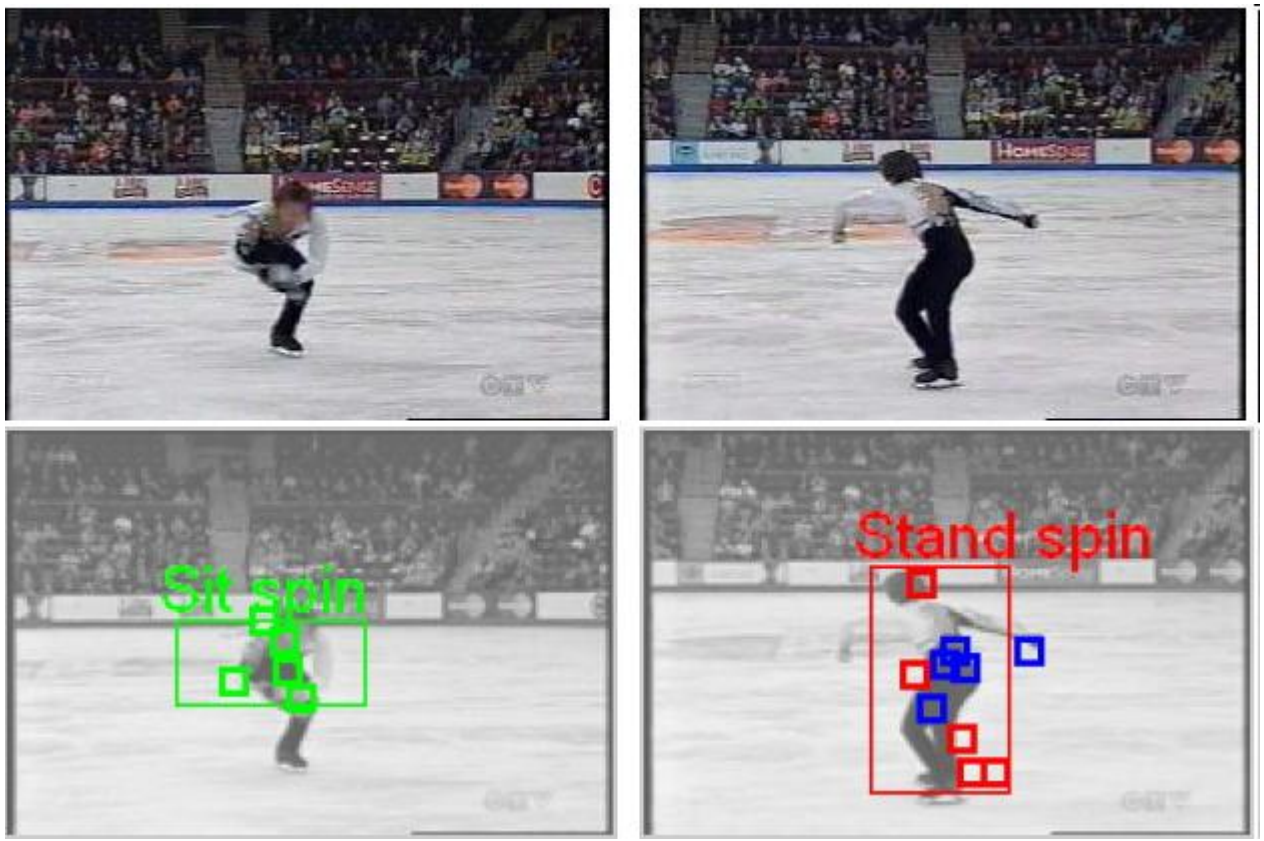
Z.Yin and R.Collins, "On-the-fly Object Modeling while Tracking," *IEEE Computer Vision and Pattern Recognition (CVPR '07),* Minneapolis, MN, June 2007, 8 pages.

Recognizing events and activities



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

Recognizing events and activities



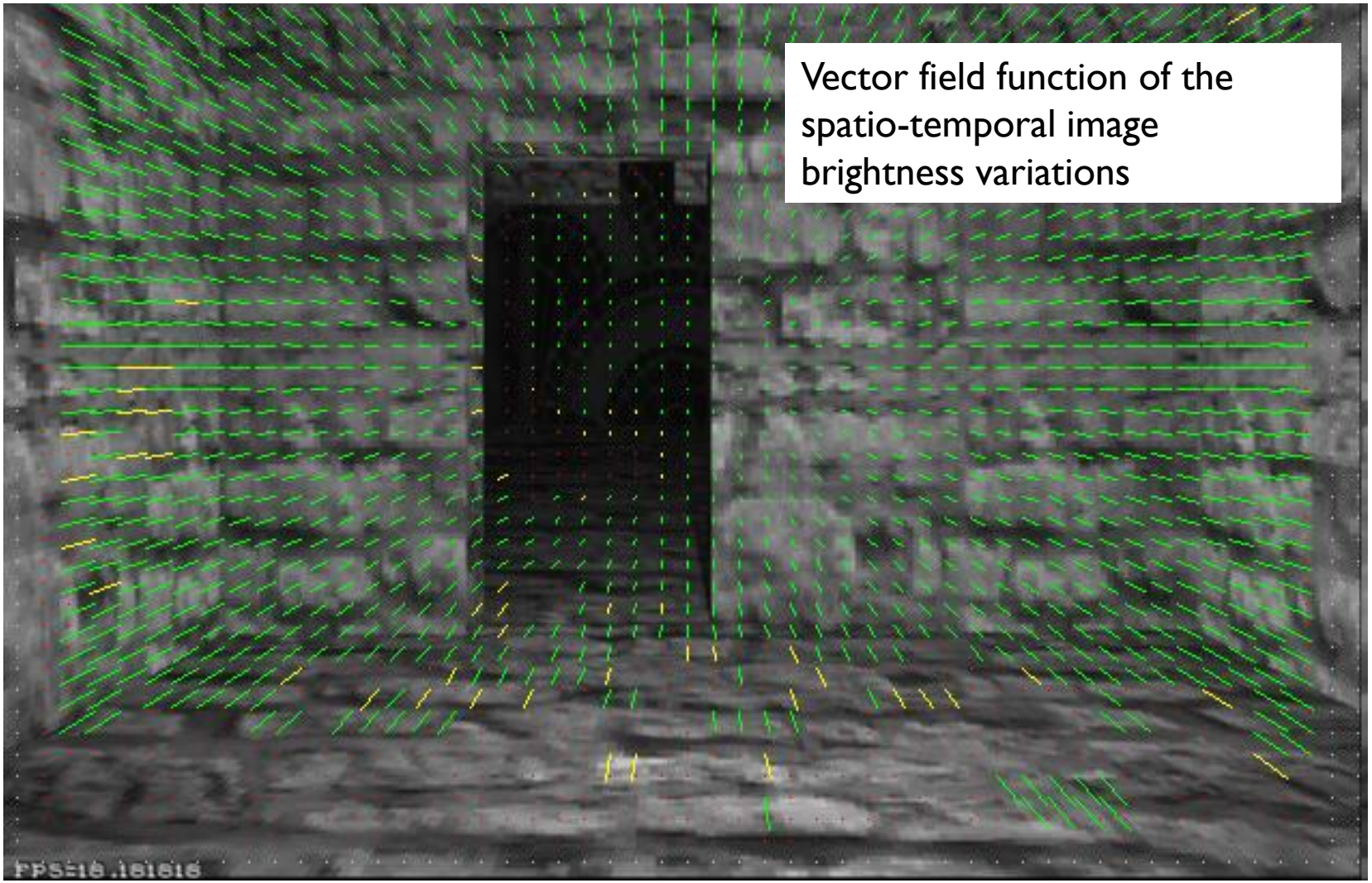
Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, **Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words**, *(BMVC)*, Edinburgh, 2006.

Learning dynamical models

Motion estimation techniques

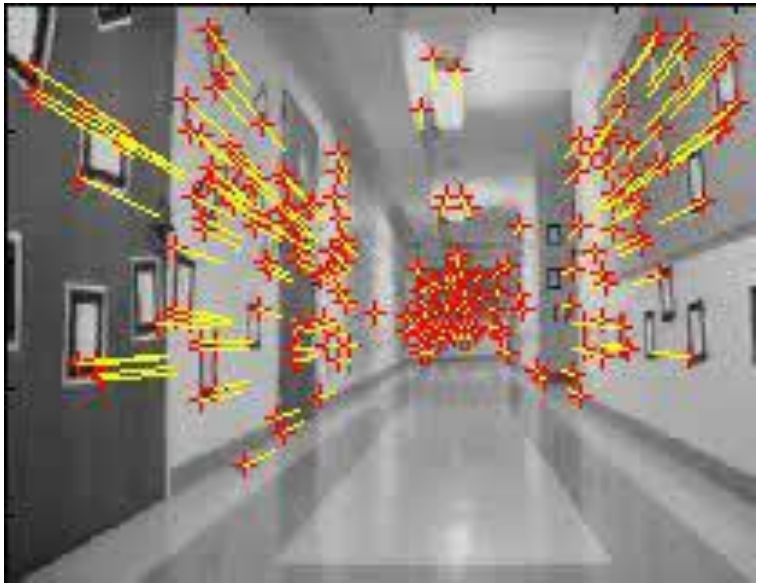
- Optical flow
  - Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)
- Feature-tracking
  - Extract visual features (corners, textured areas) and "track" them over multiple frames

Optical flow



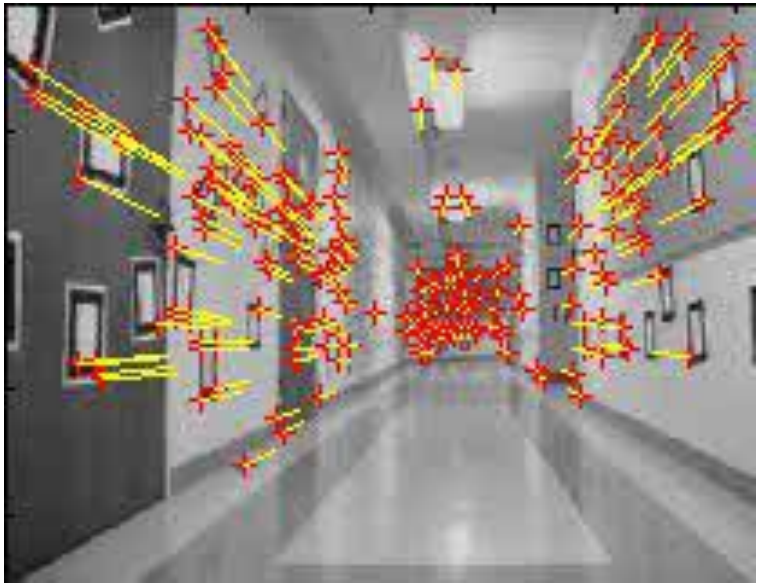
Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

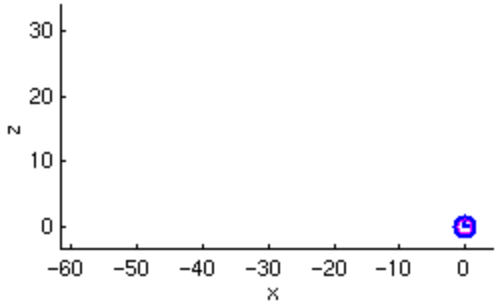
Feature-tracking



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

Feature-tracking





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

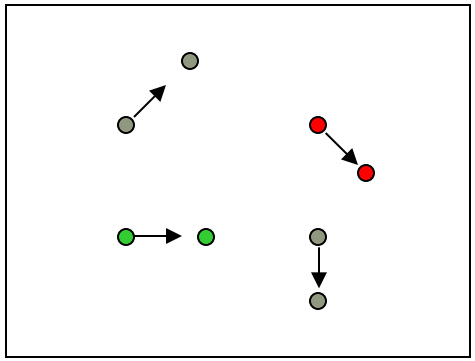
Optical flow

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

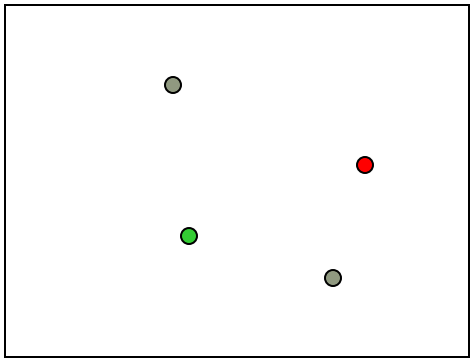
- Note: apparent motion can be caused by lighting changes without any actual motion
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

GOAL: Recover image motion at each pixel from optical flow

Estimating optical flow



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

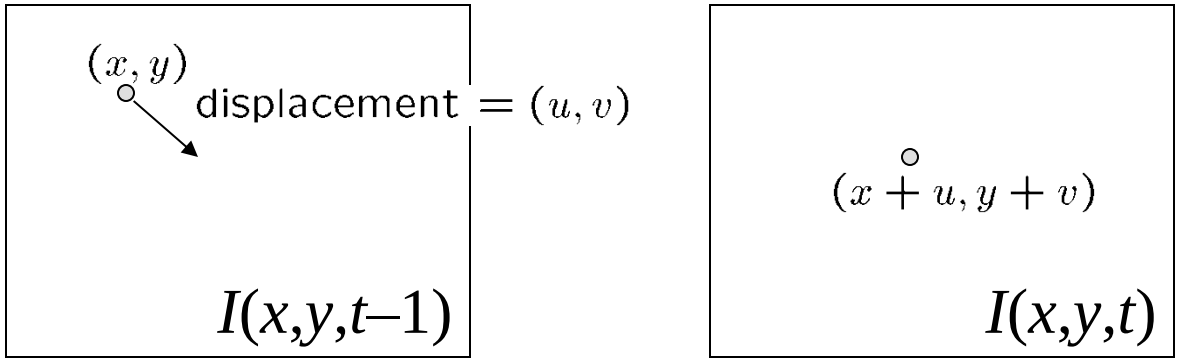


$I(x,y,t)$

- 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

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

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

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

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

The brightness constancy constraint

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

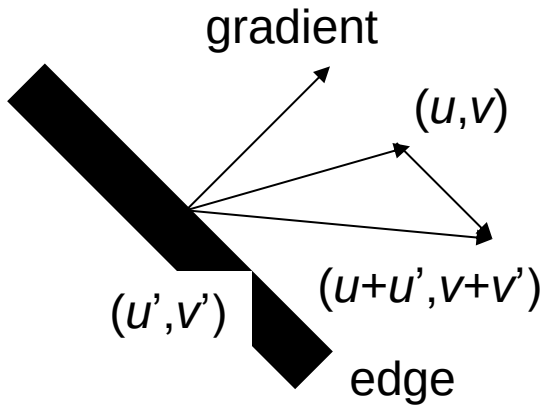
$$\nabla I \cdot \begin{bmatrix} u & v \end{bmatrix}^T + I_t = 0$$

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

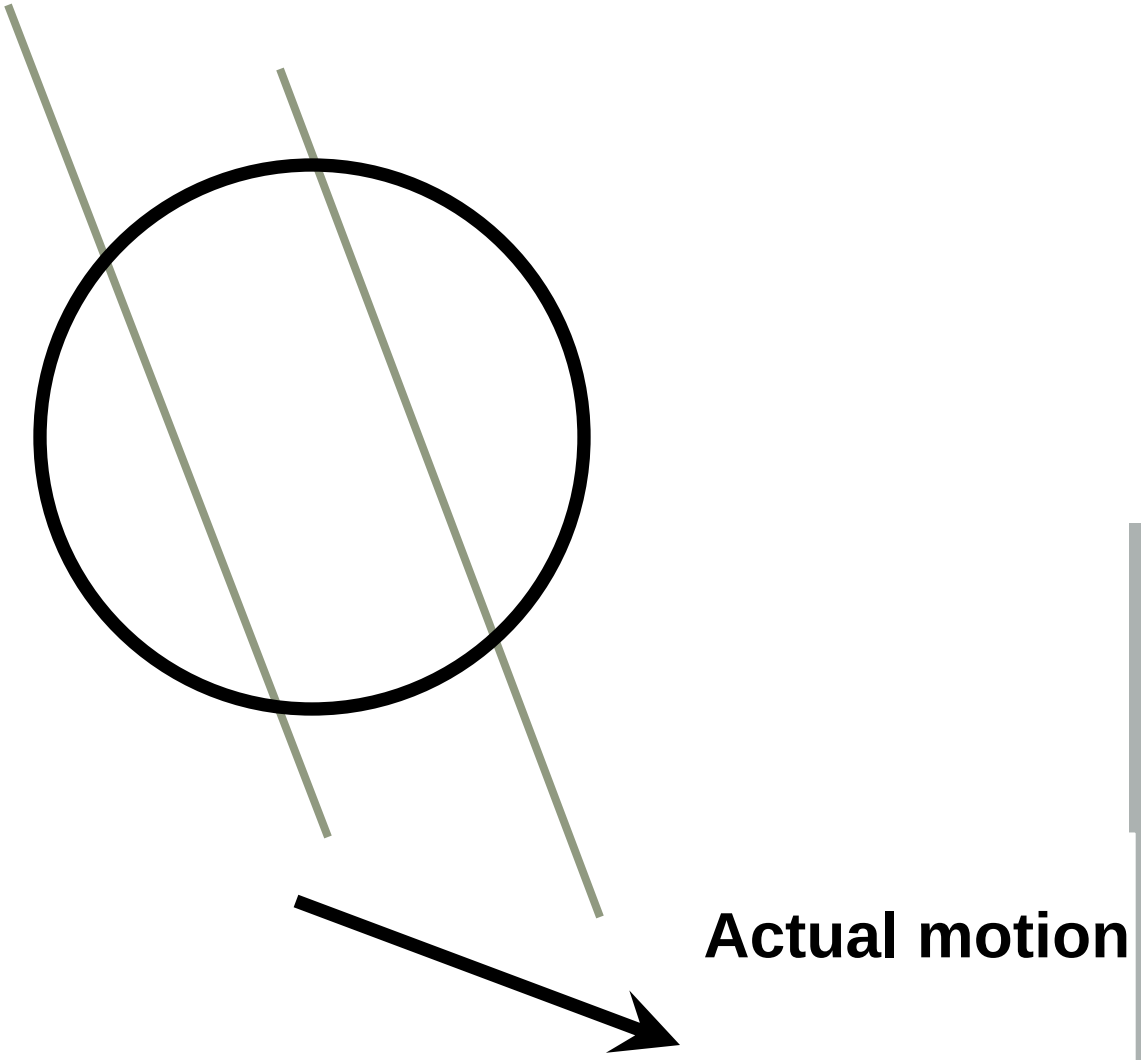
The component of the flow perpendicular to the gradient (i.e., parallel to the edge) cannot be measured

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

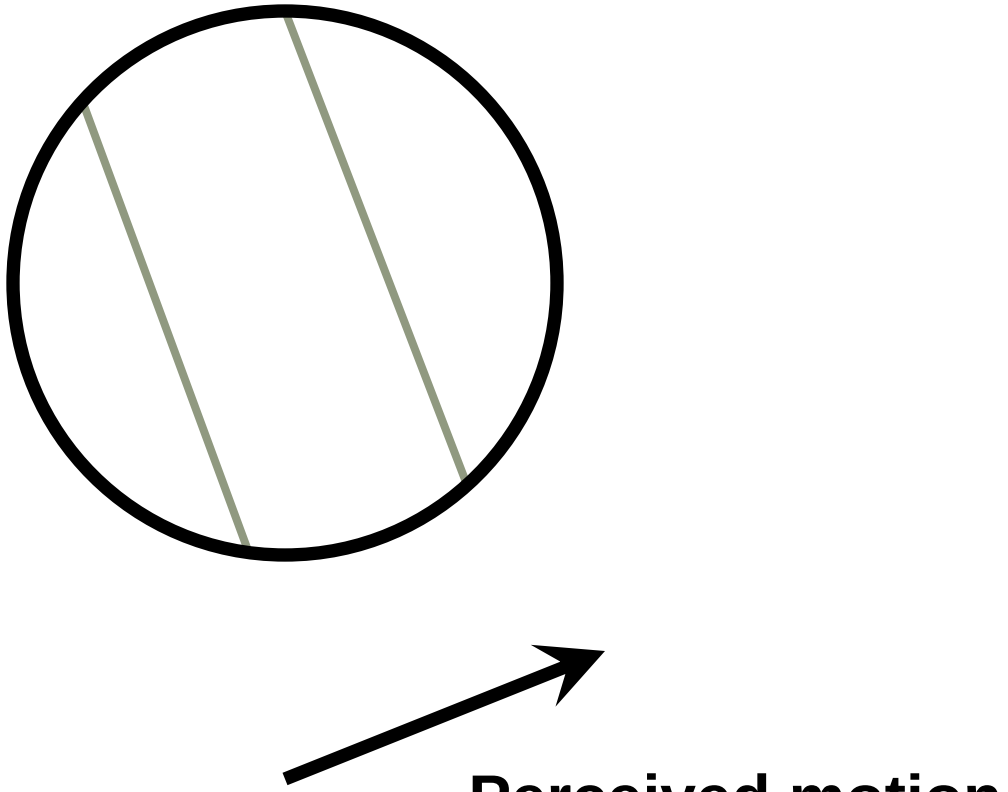
$$\nabla I \cdot \begin{bmatrix} u' & v' \end{bmatrix}^T = 0$$



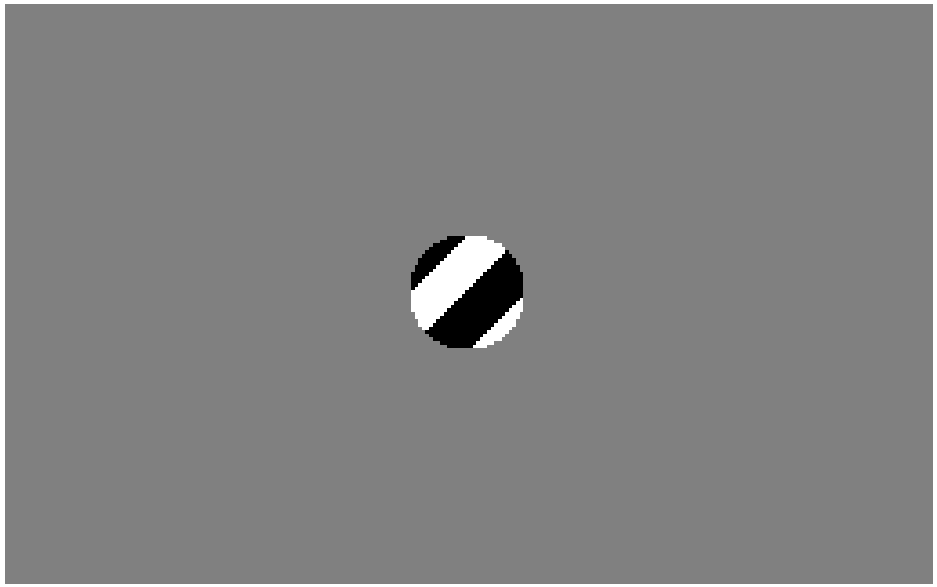
The aperture problem



The aperture problem

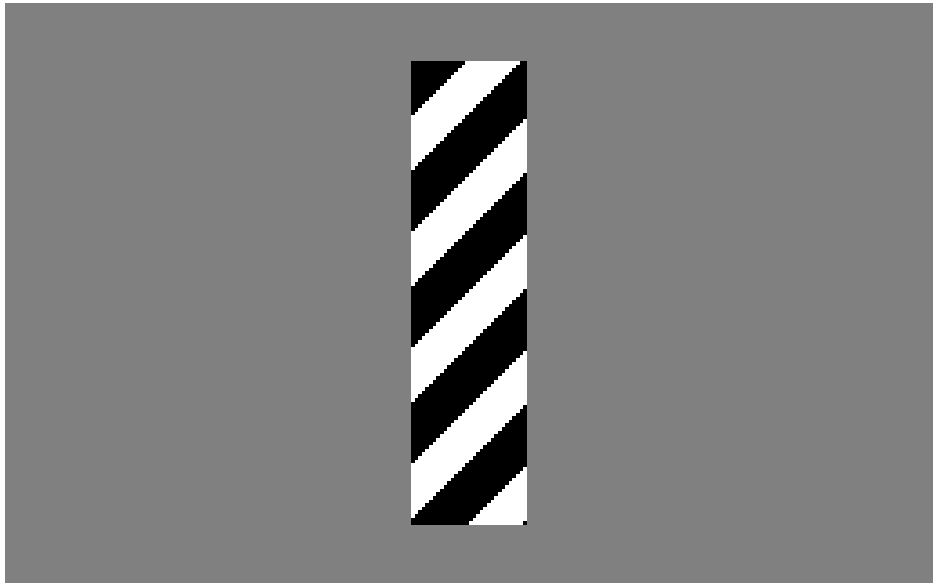


The barber pole illusion



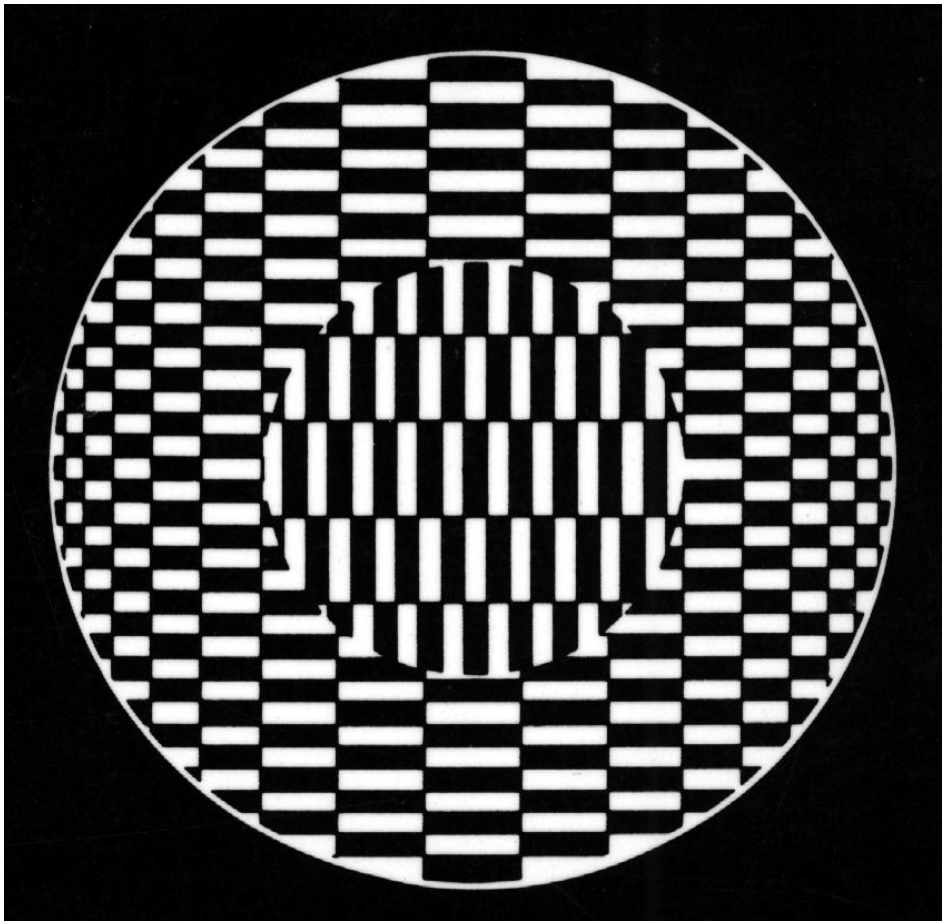
http://en.wikipedia.org/wiki/Barberpole_illusion

The barber pole illusion



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

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

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

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

Solving the ambiguity…

- Least squares problem:

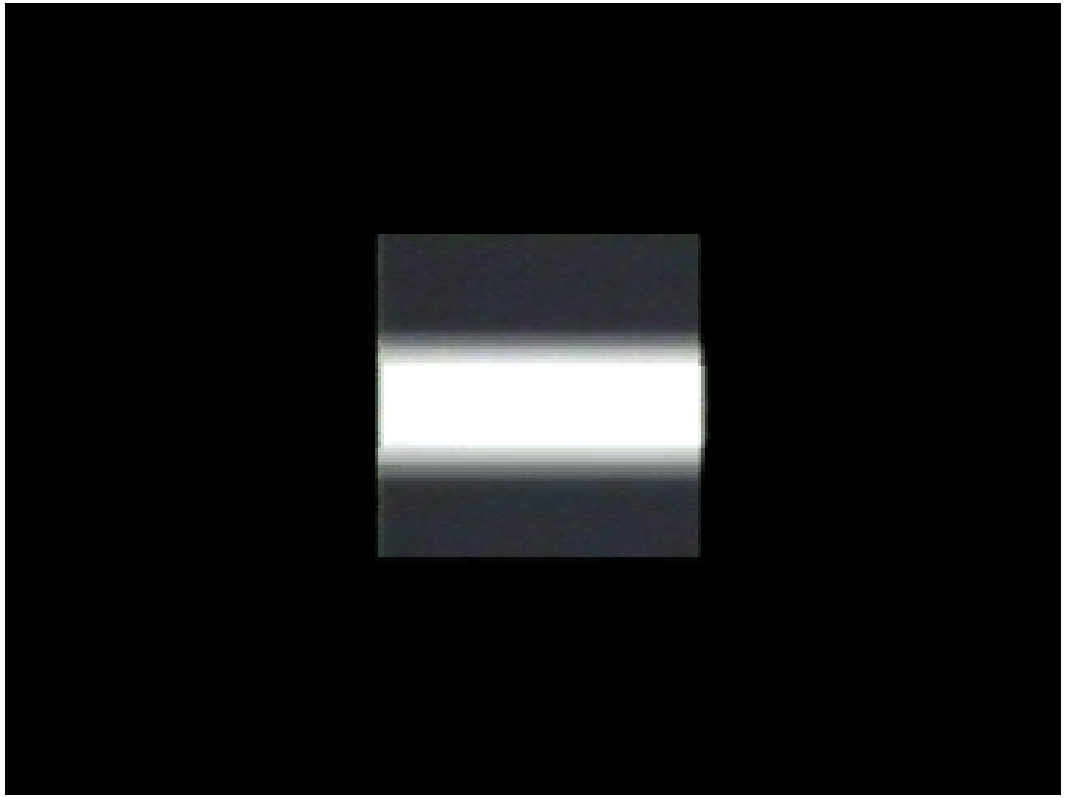
$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25})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

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

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

Conditions for solvability

- "Bad" case: single straight edge



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 $(A^T A)\ d = A^T b$

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

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

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

Conditions for solvability

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

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

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

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?

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

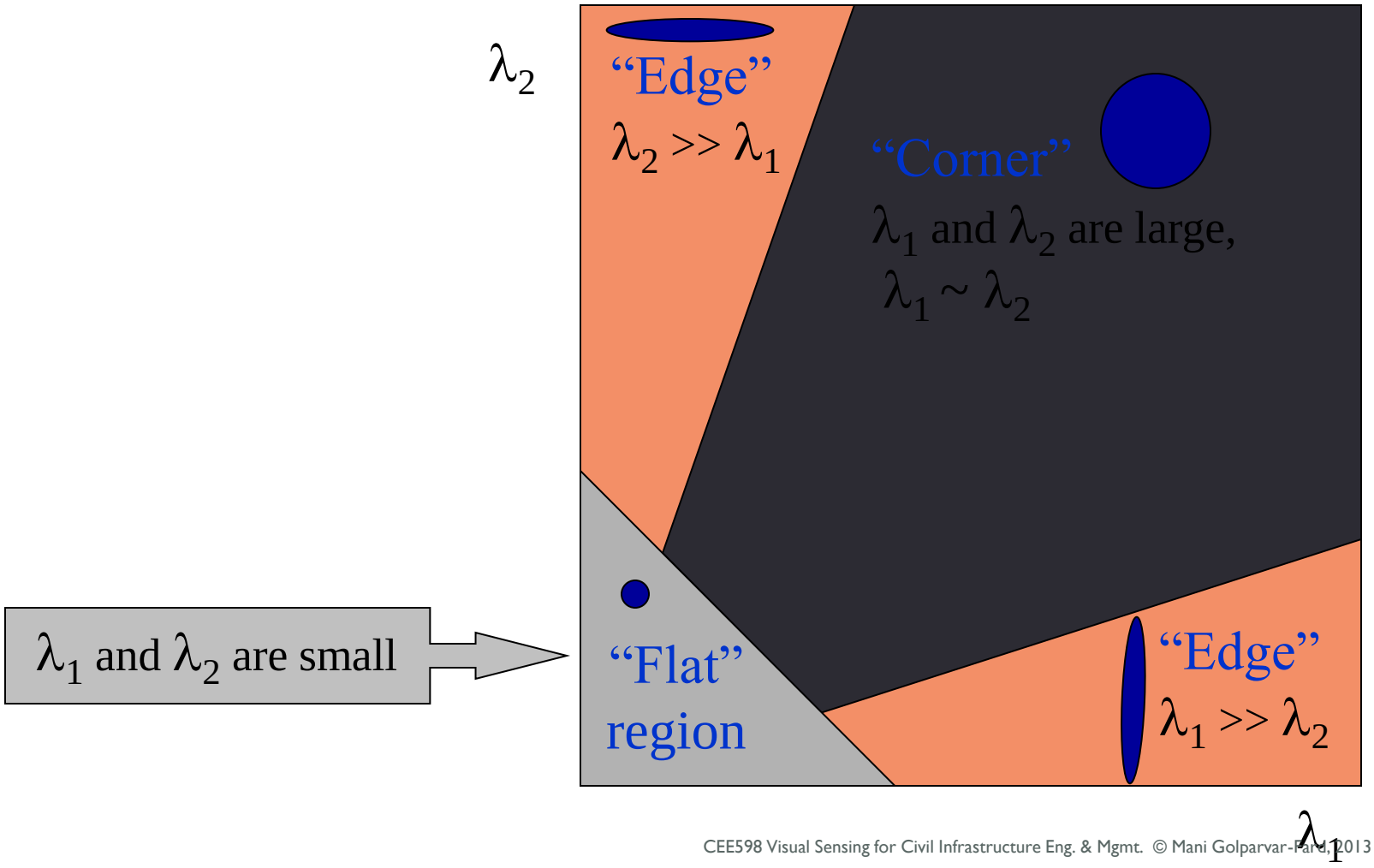
(Harris corner detector…)

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

- 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:



Edge

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

– 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

High-texture region

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

– gradients are different, large magnitudes

– large λ_1, large λ_2

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

Recap

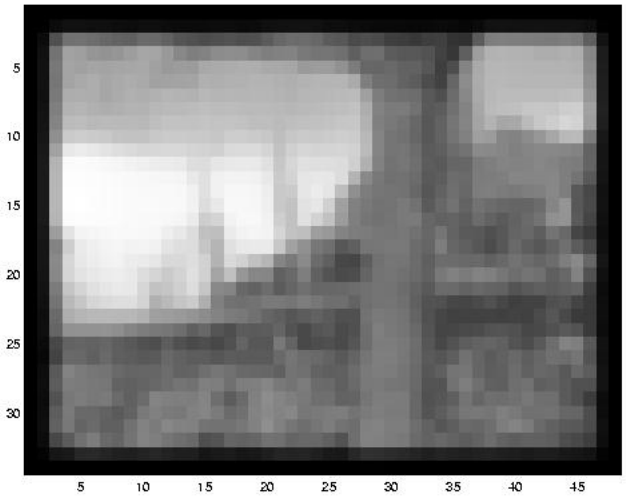
- Key assumptions (Errors in Lucas-Kanade)
  - **Small motion:** points do not move very far
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Spatial coherence:** points move like their neighbors

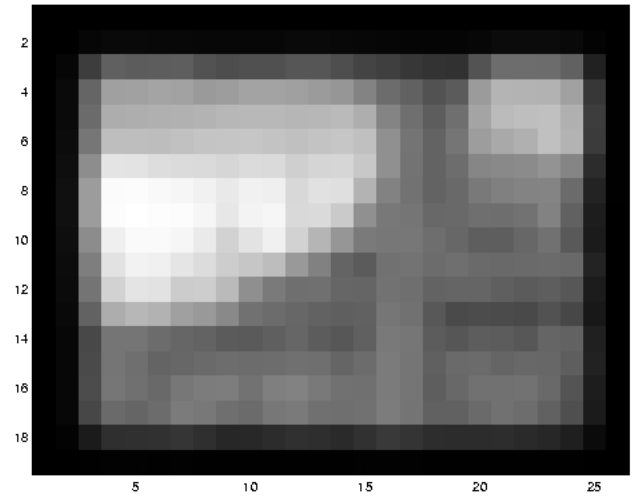
Revisiting the small motion assumption

- 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!



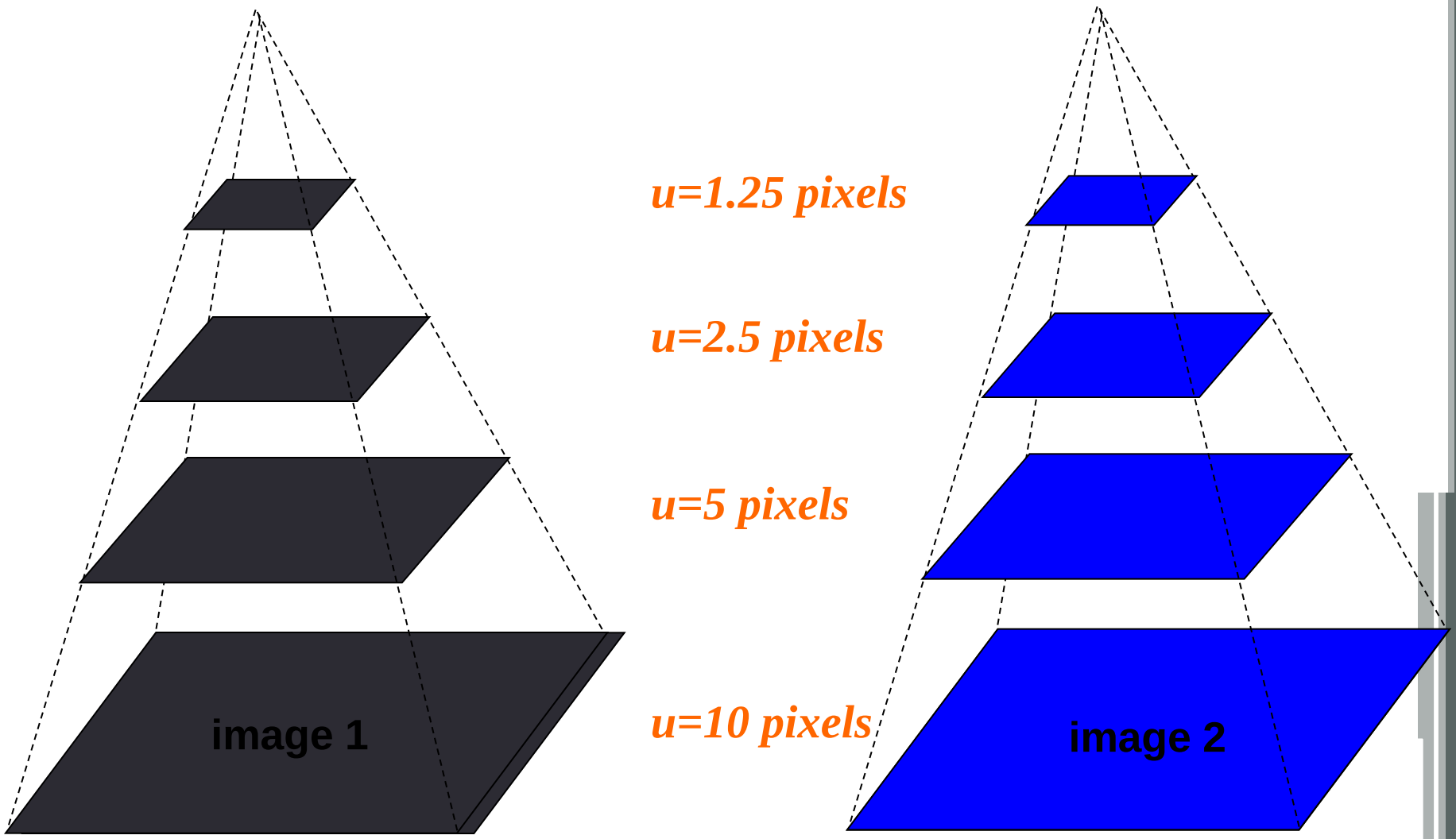


* From Khurram Hassan-Shafique

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

Coarse-to-fine optical flow estimation



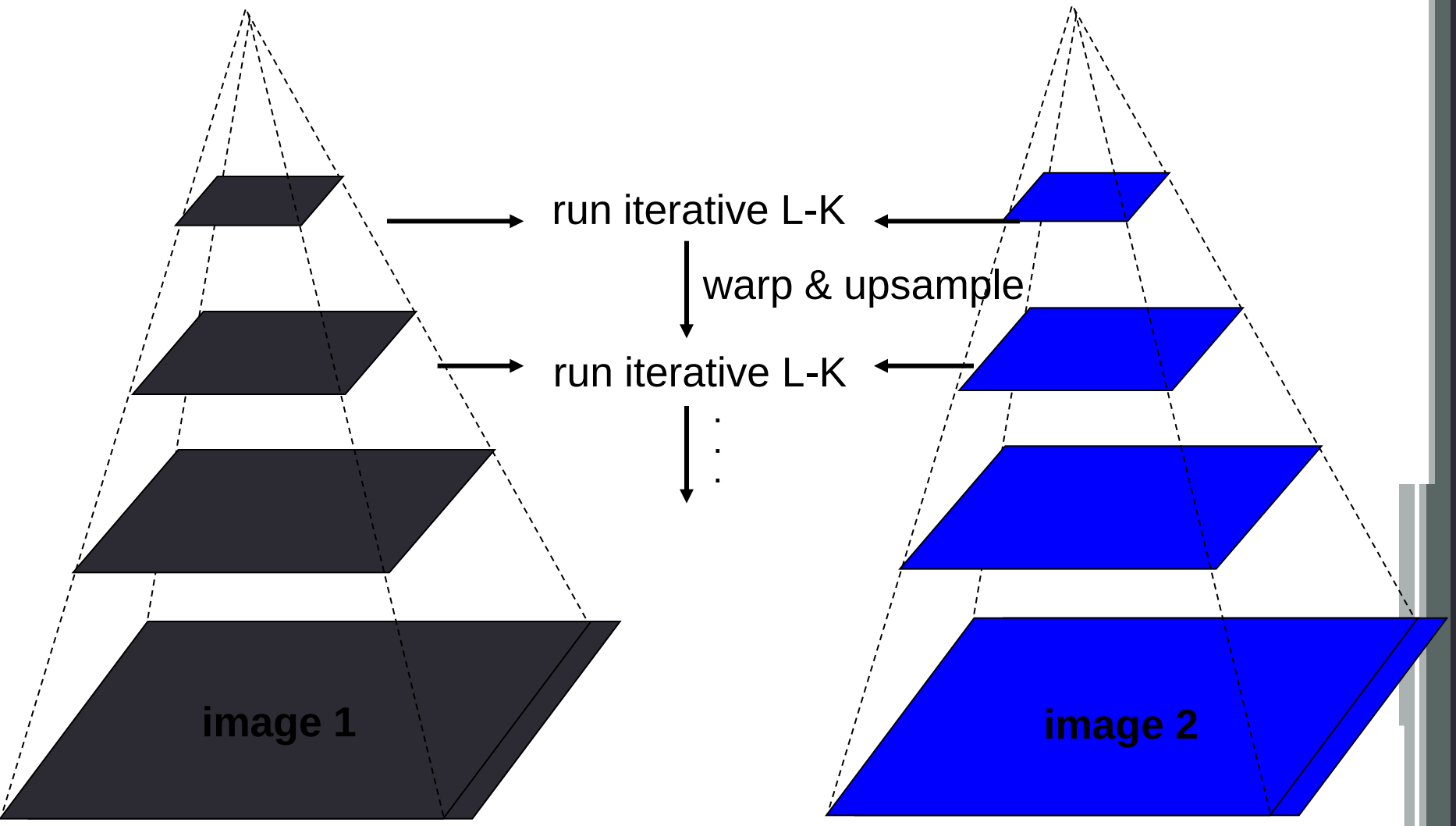
Gaussian pyramid of image 1

Gaussian pyramid of image 2

Iterative Refinement

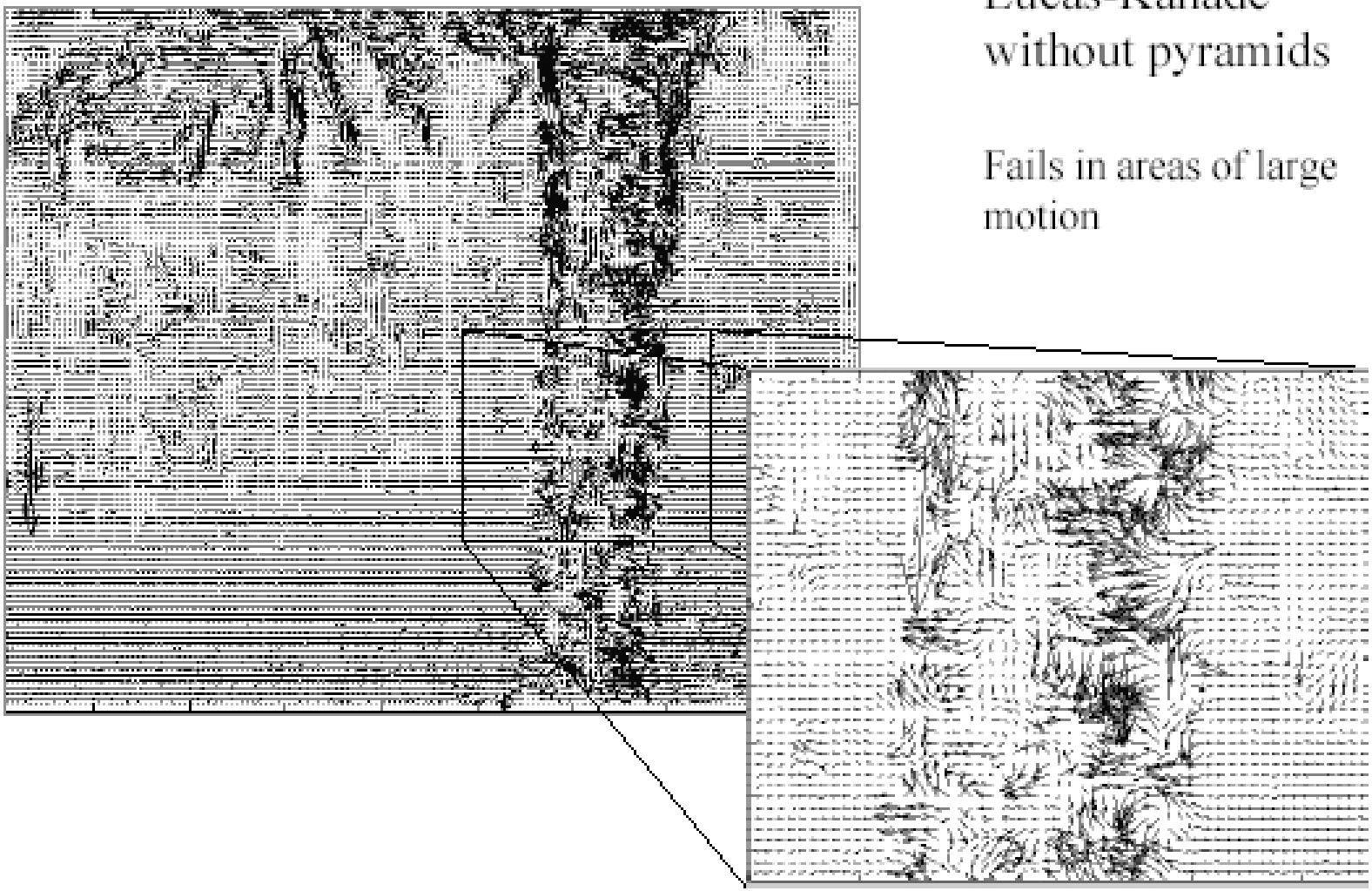
- Iterative Lukas-Kanade Algorithm
  1. Estimate velocity at each pixel by solving Lucas-Kanade equations
  2. Warp I(t-1) towards I(t) using the estimated flow field
     - *use image warping techniques*
  3. Repeat until convergence

Coarse-to-fine optical flow estimation



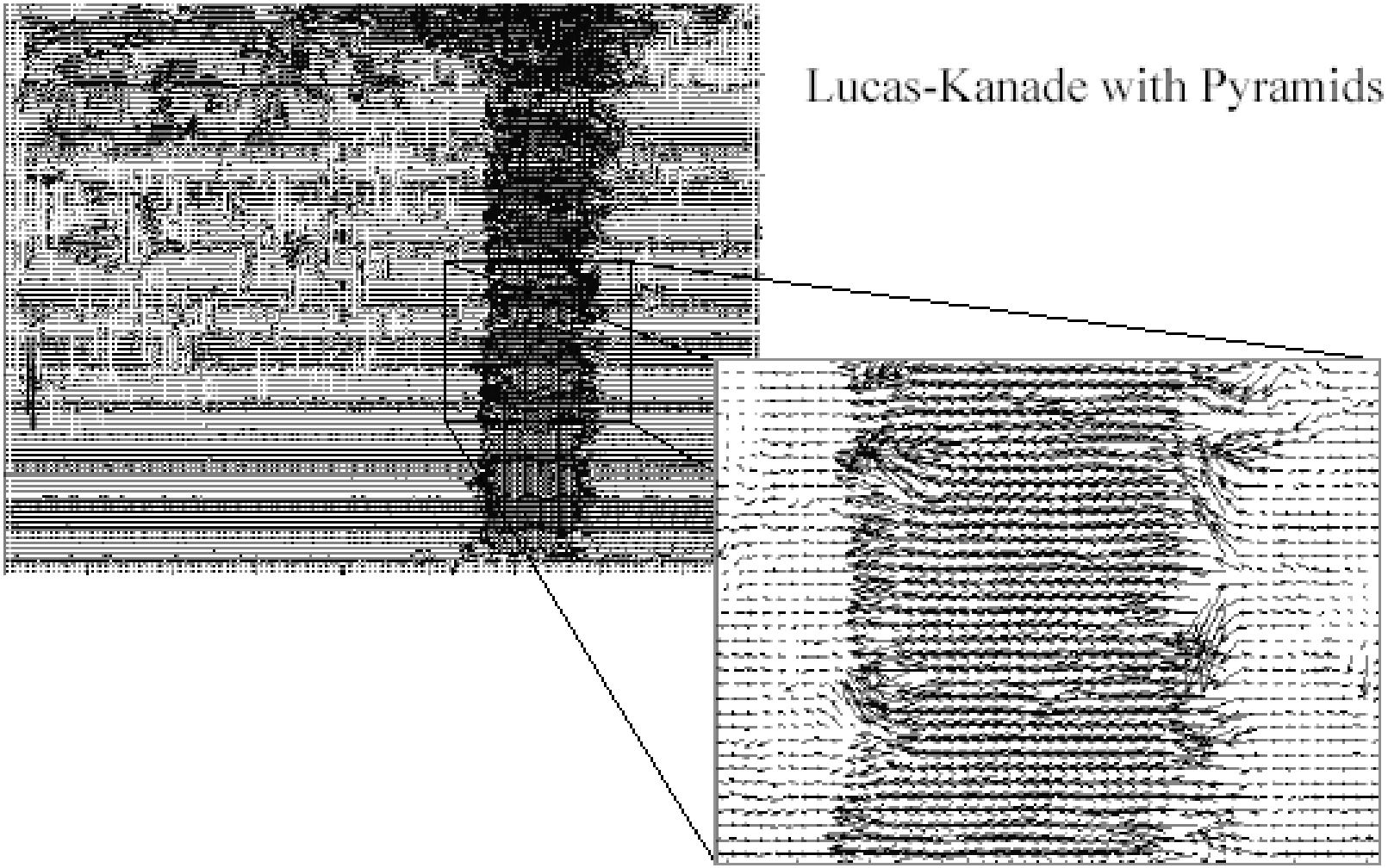
Gaussian pyramid of image 1 (t) **Gaussian pyramid of image 2 (t+1)**

Optical Flow Results



*From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Optical Flow Results



*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

Motion segmentation

- How do we represent the motion in this scene?

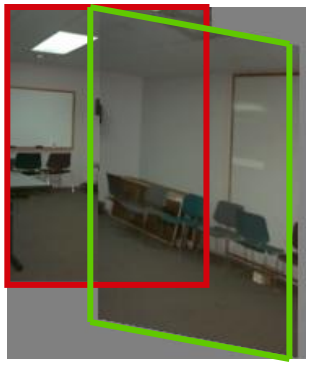
Motion segmentation

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

Affine motion

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

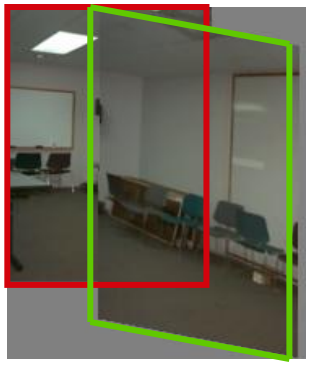


- Substituting into the brightness constancy equation:

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

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:

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

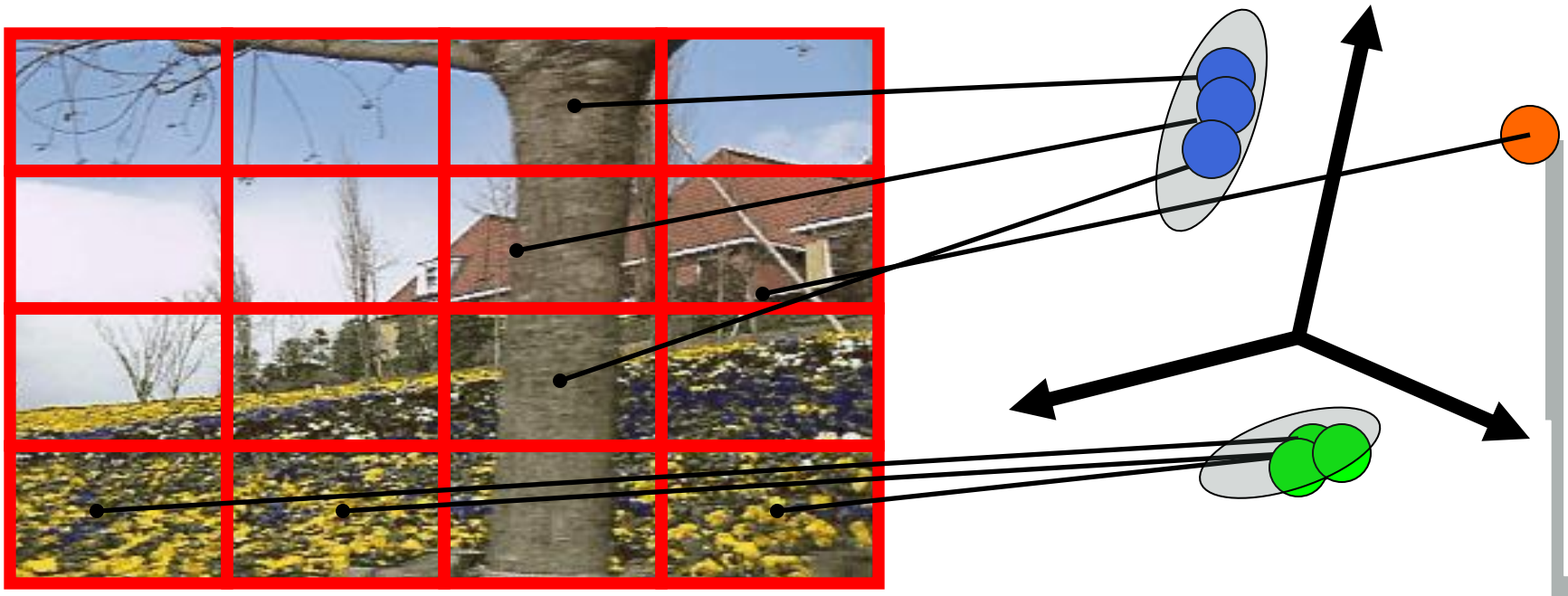
- Each pixel provides 1 linear constraint in 6 unknowns

- Least squares minimization:

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

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



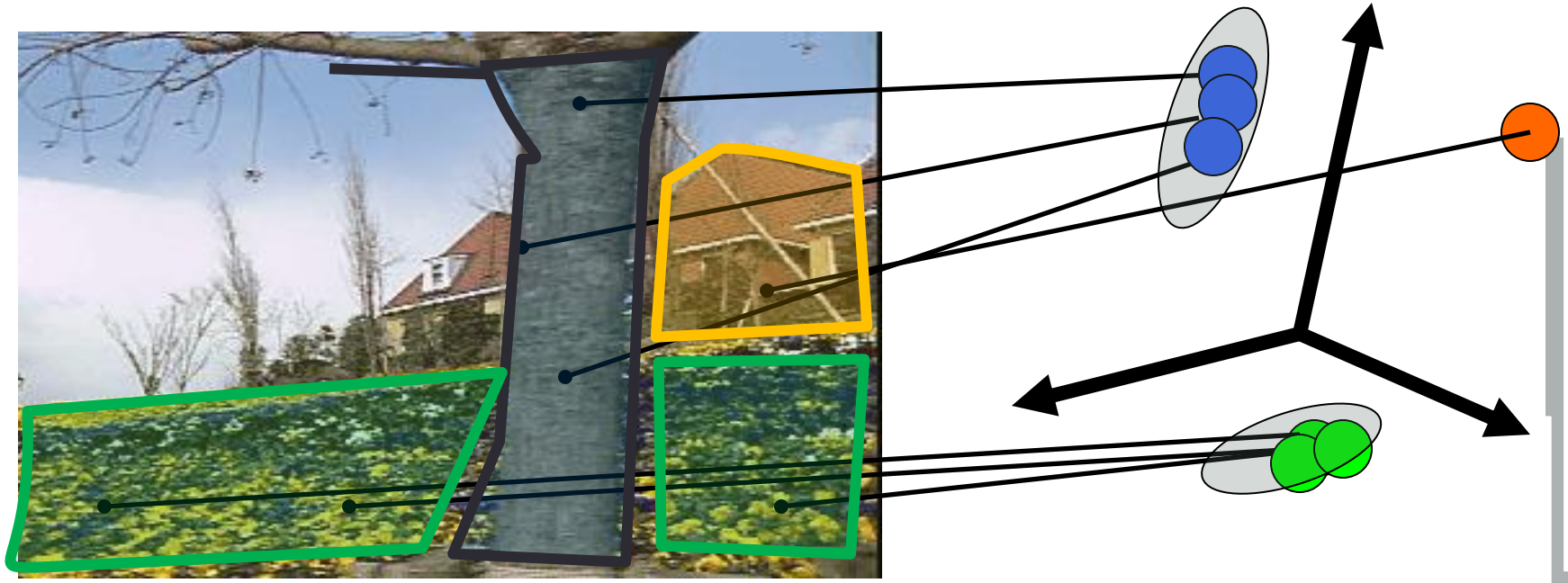
How do we estimate the layers?

- 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

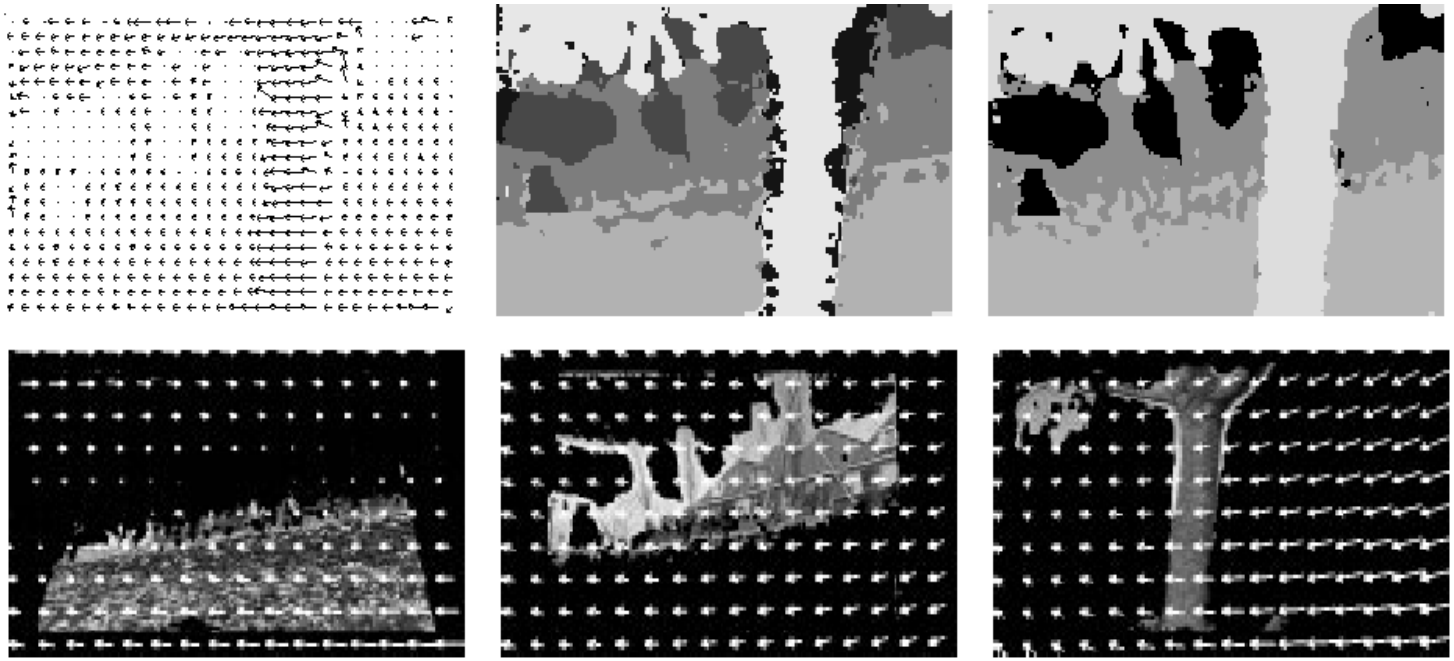
2. Iterate until convergence:
  - Assign each pixel to best hypothesis
    - Pixels with high residual error remain unassigned
  - Perform region filtering to enforce spatial constraints
  - Re-estimate affine motions in each region

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



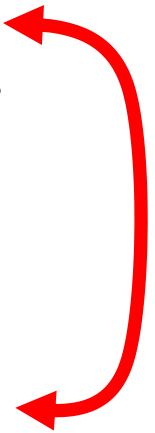
Example result



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

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

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"

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

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

Tracking example

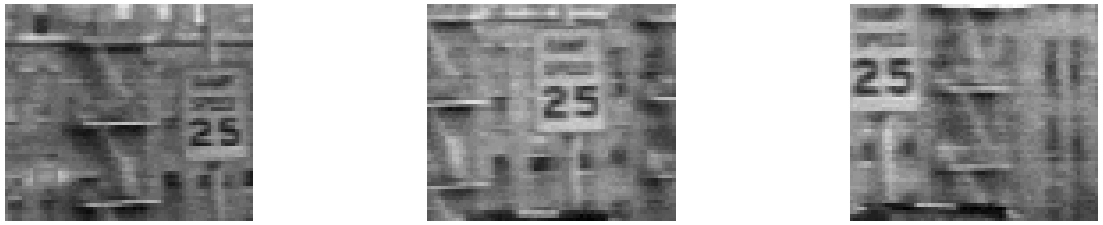


Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

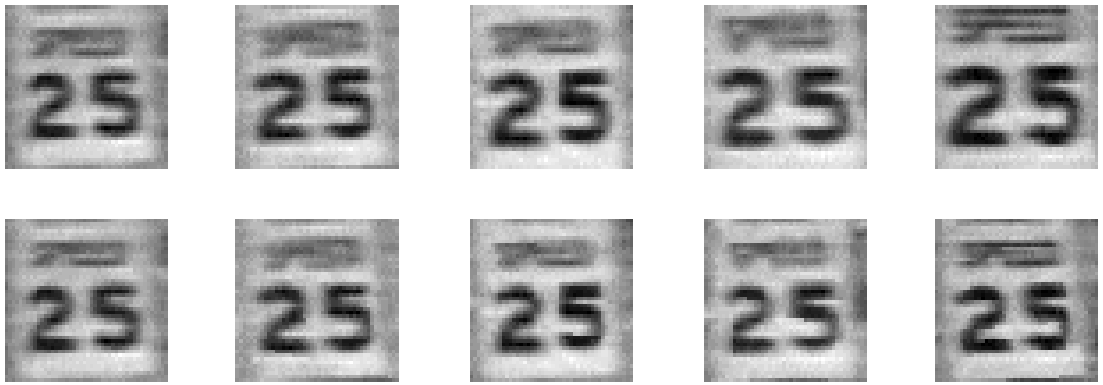
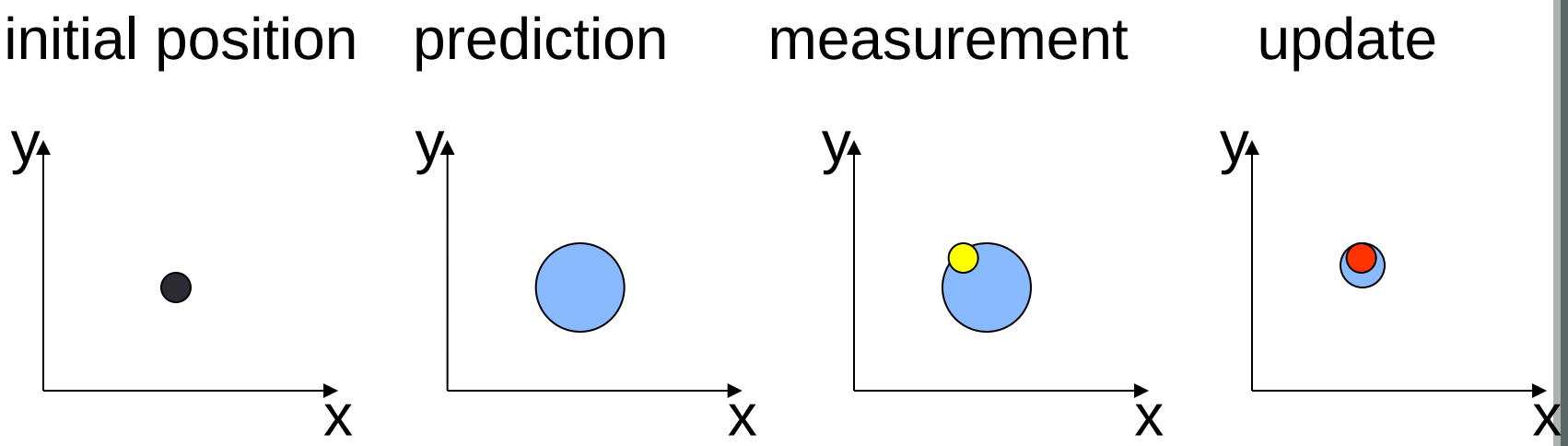


Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

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

Tracking with dynamics



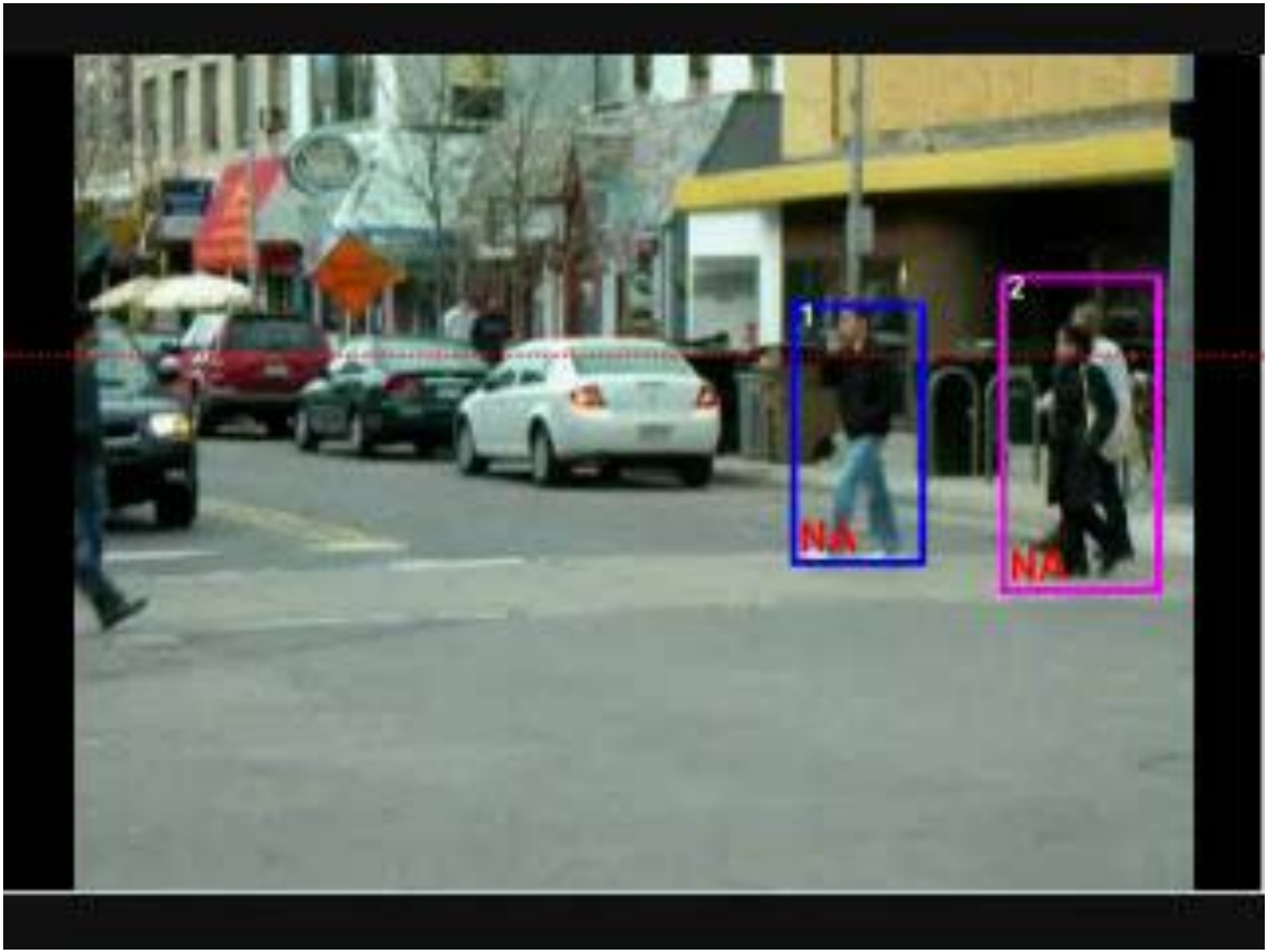
The Kalman filter:

- 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)

2D Target tracking using Kalman filter in MATLAB

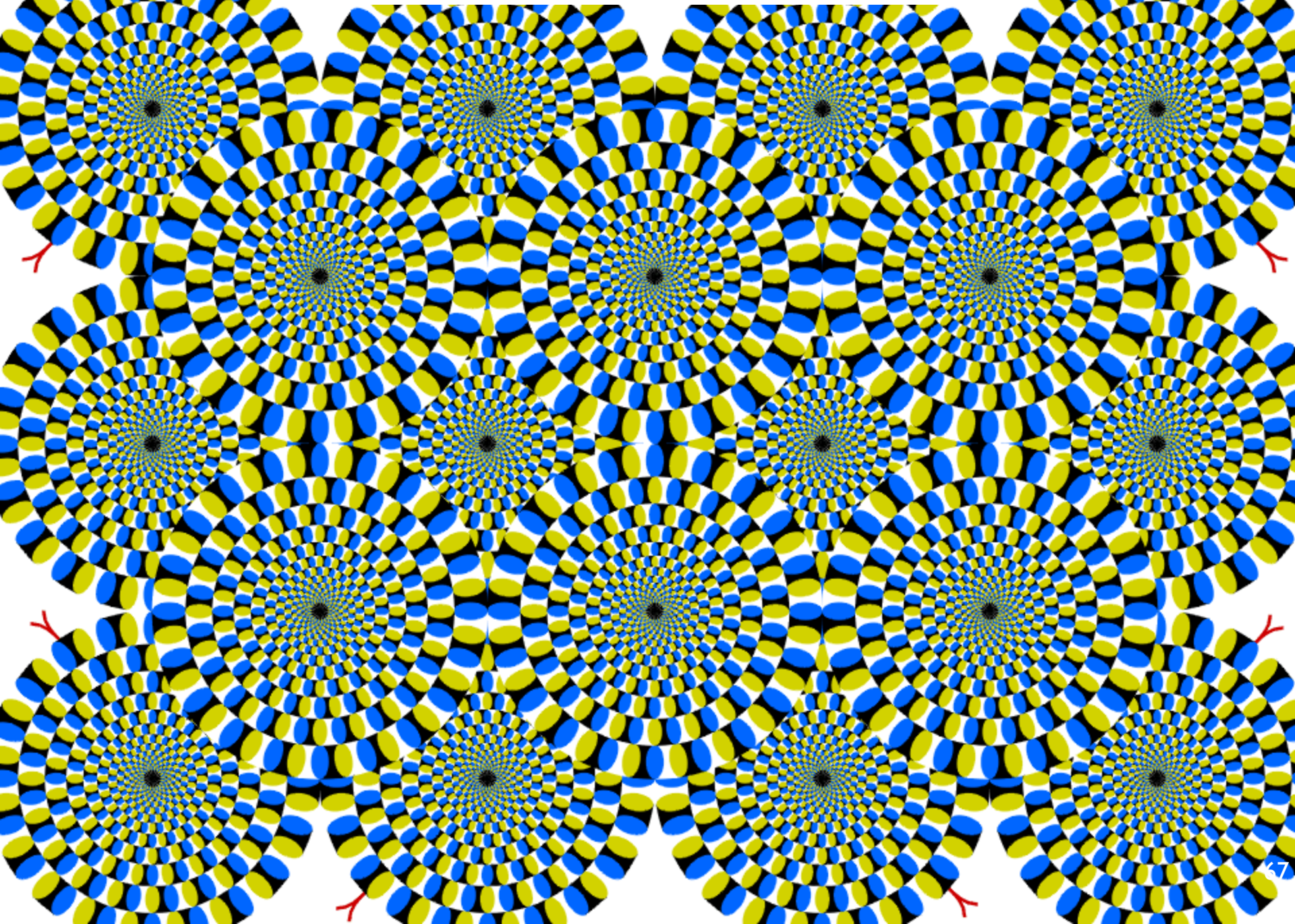
by AliReza KashaniPour

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



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!



http://www.vision.ee.ethz.ch/showroom/media/soccer.mpg

Next Class

- Object Recognition - intro