CS231M · Mobile Computer Vision

Lecture 7

Optical flow and tracking

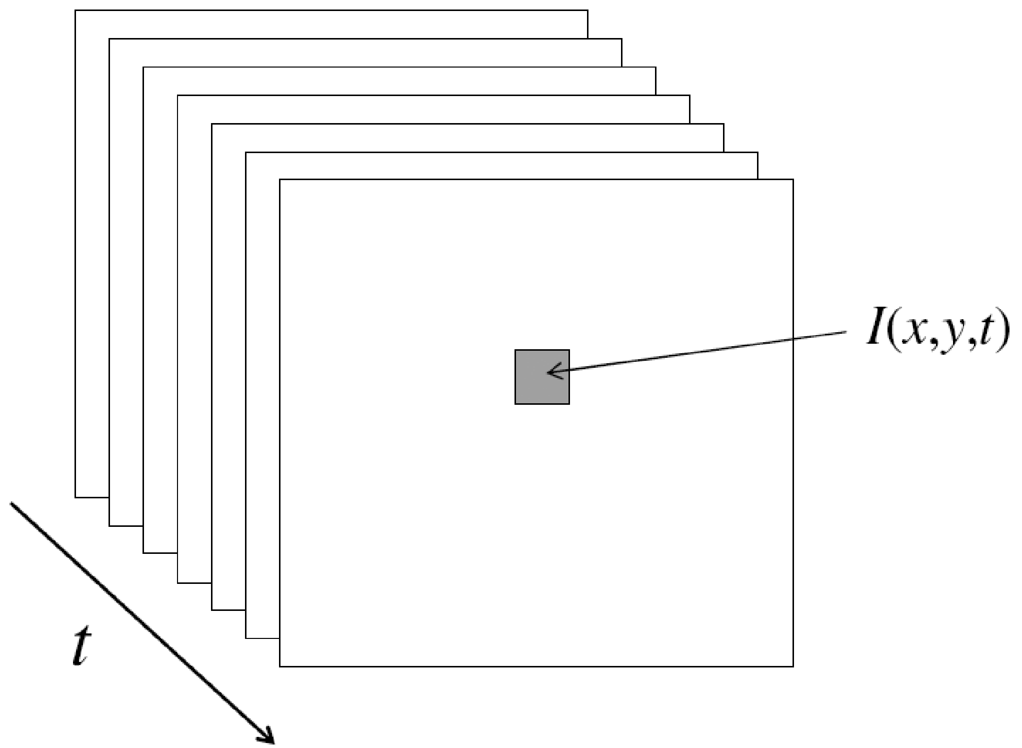
- Introduction
- Optical flow & KLT tracker
- Motion segmentation

Forsyth, Ponce “Computer vision: a modern approach”:
- Chapter 10, Sec 10.6
- Chapter 11, Sec 11.1

Szeliski, “Computer Vision: algorithms and applications"
- Chapter 8, Sec. 8.5

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)



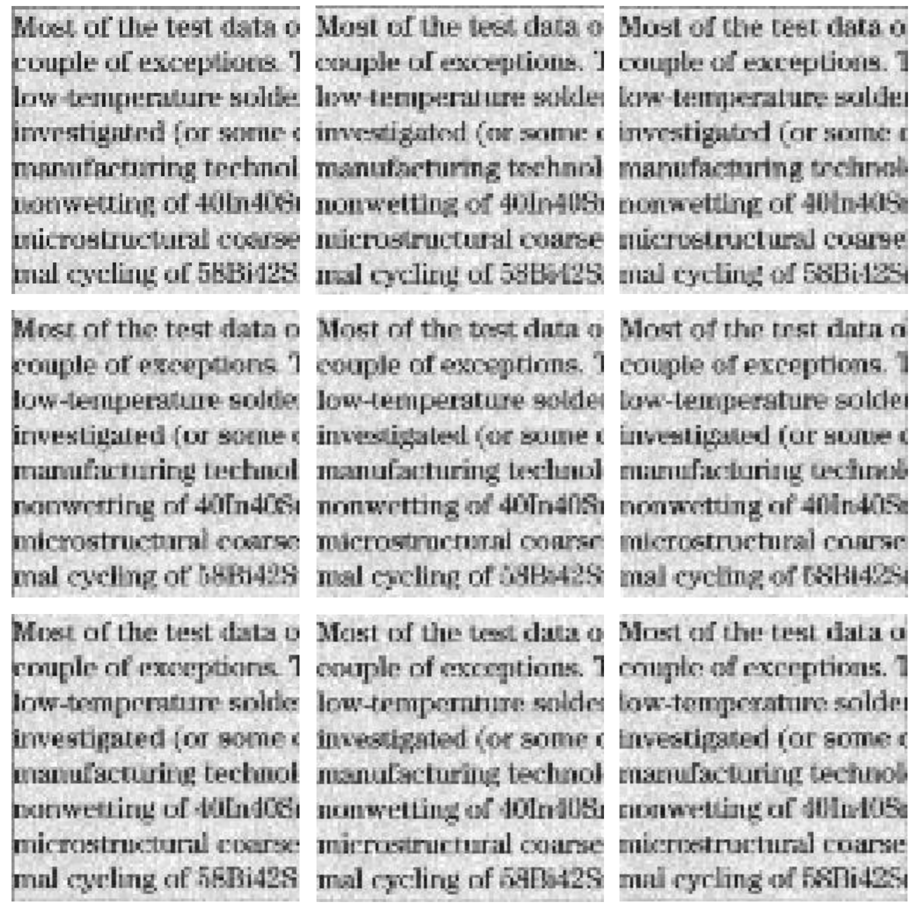
Uses of motion

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

Super-resolution

- Irani, M.; Peleg, S. (June 1990). "Super Resolution From Image Sequences". International Conference on Pattern Recognition
- Fast and Robust Multiframe Super Resolution, Sina Farsiu, M. Dirk Robinson, Michael Elad, and Peyman Milanfar, EEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 10, OCTOBER 2004

Example: A set of low quality images



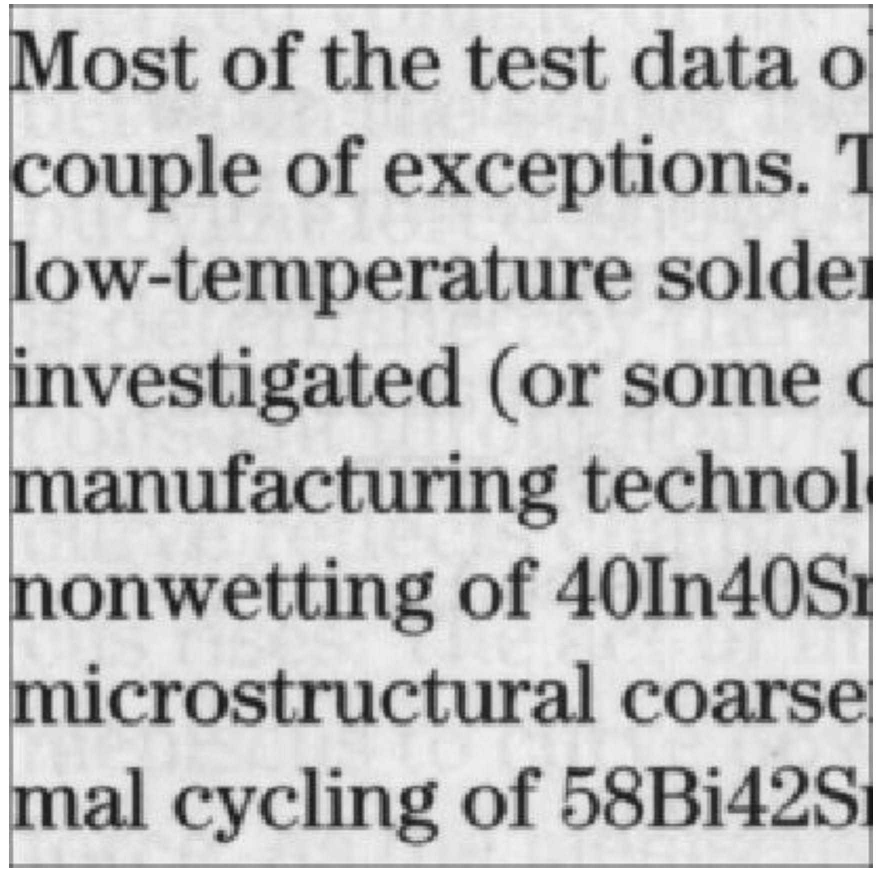
Super-resolution

Each of these images looks like this:

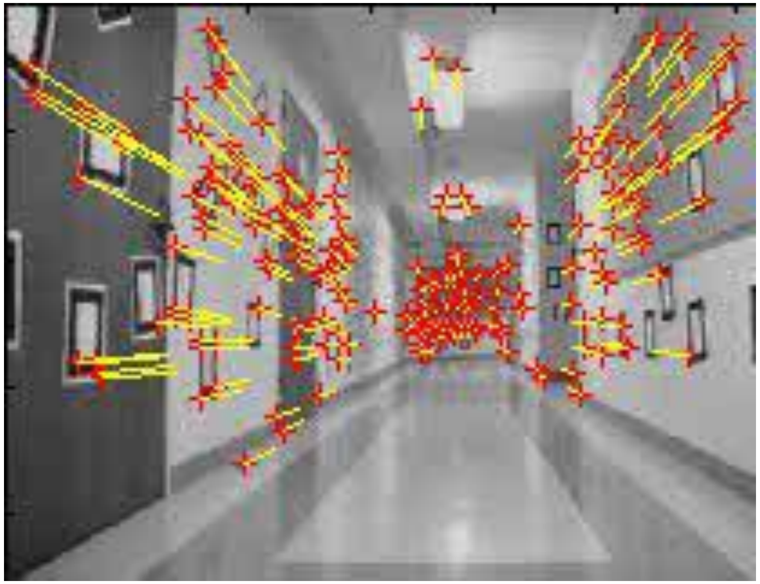

Most of the test data o
couple of exceptions. T
low-temperature solder
investigated (or some c
manufacturing technol
microstructural coarse

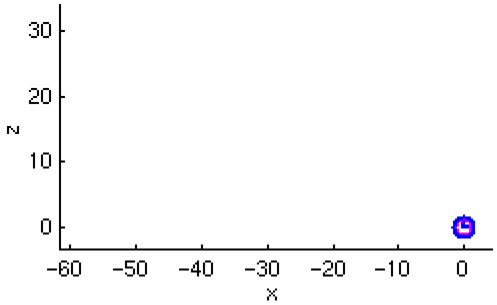
Super-resolution

The recovery result:


Most of the test data o
couple of exceptions. T
low-temperature solder
investigated (or some o
manufacturing technol
nonwetting of 40In40Sn
microstructural coarse
mal cycling of 58Bi42Sn

Visual SLAM

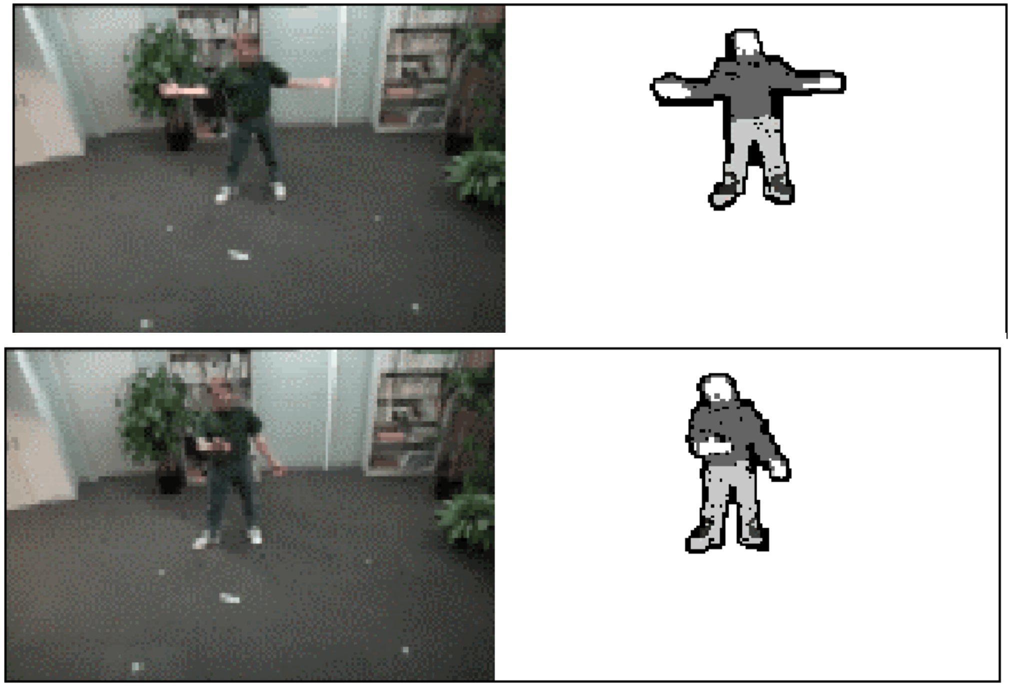




Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology

Segmenting objects based on motion cues

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

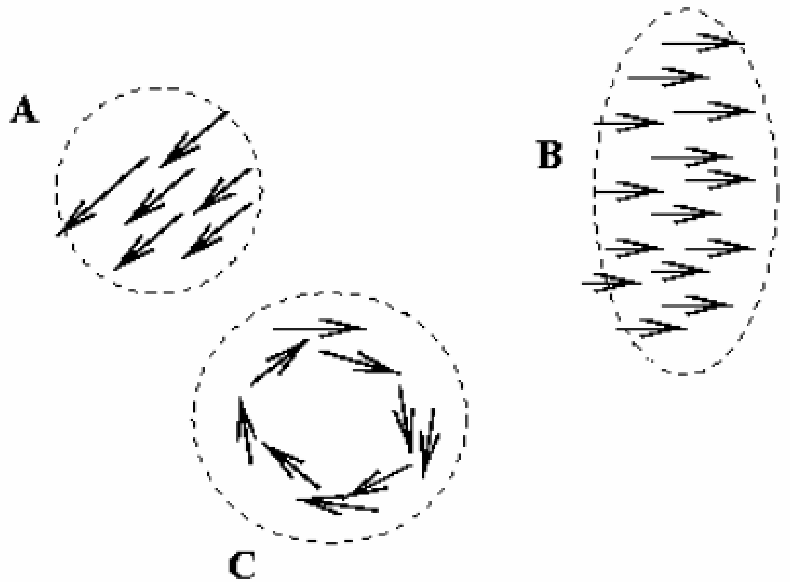


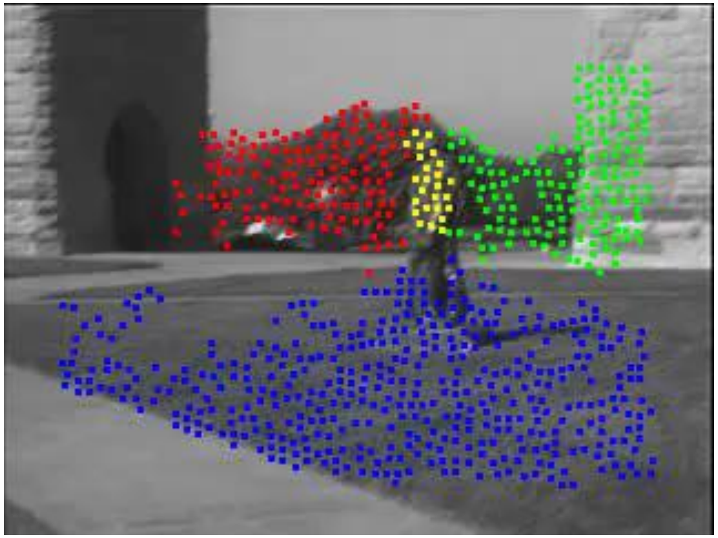
https://www.youtube.com/watch?v=YAszeOaInUM

Suha Kwak, Taegyu Lim, Woonhyun Nam, Bohyung Han, Joon Hee Han: Generalized background subtraction based on hybrid inference by belief propagation and Bayesian filtering. ICCV 2011

Segmenting objects based on motion cues

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

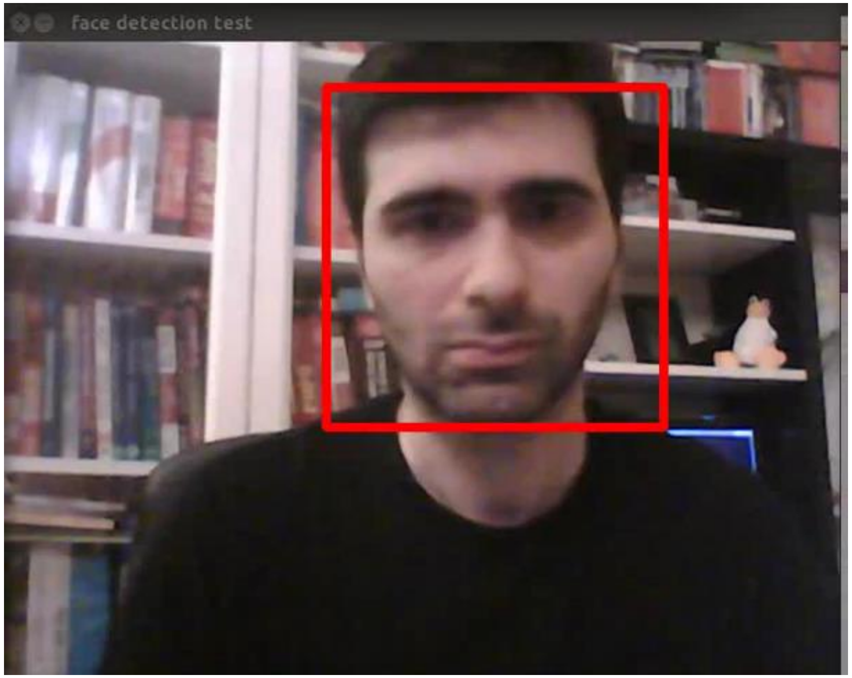




S. J. Pundlik and S. T. Birchfield, Motion Segmentation at Any Speed, Proceedings of the British Machine Vision Conference 06

Tracking objects

- Facing tracking on openCV

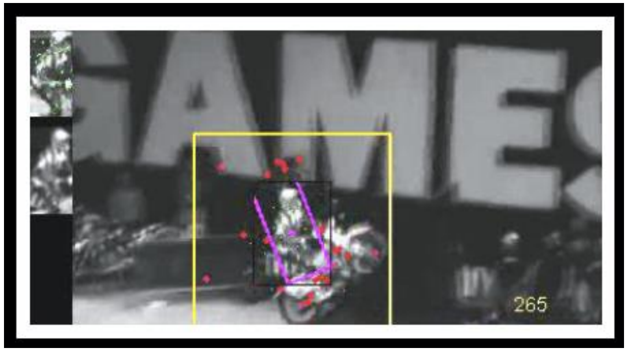


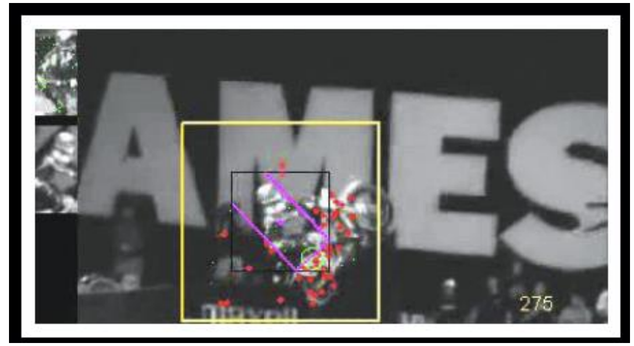
OpenCV's face tracker uses an algorithm called Camshift (based on the meanshift algorithm)

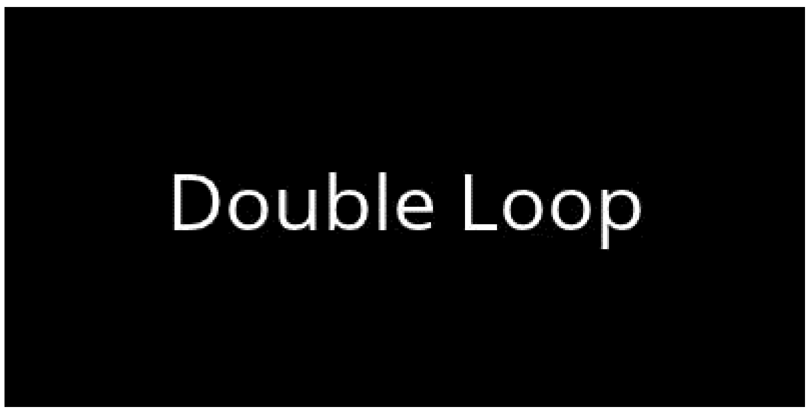
http://www.youtube.com/watch?v=HTk_UwAYzVk

Tracking objects

Object Tracking by Oversampling Local Features. Del Bimbo, and F. Pernici, IEEE Transaction On Pattern Analisys And Machine Intelligence, 2014







- Use Scale Invariant Feature Transform (SIFT) when applied to (flat) objects

http://www.micc.unifi.it/pernici/#alien

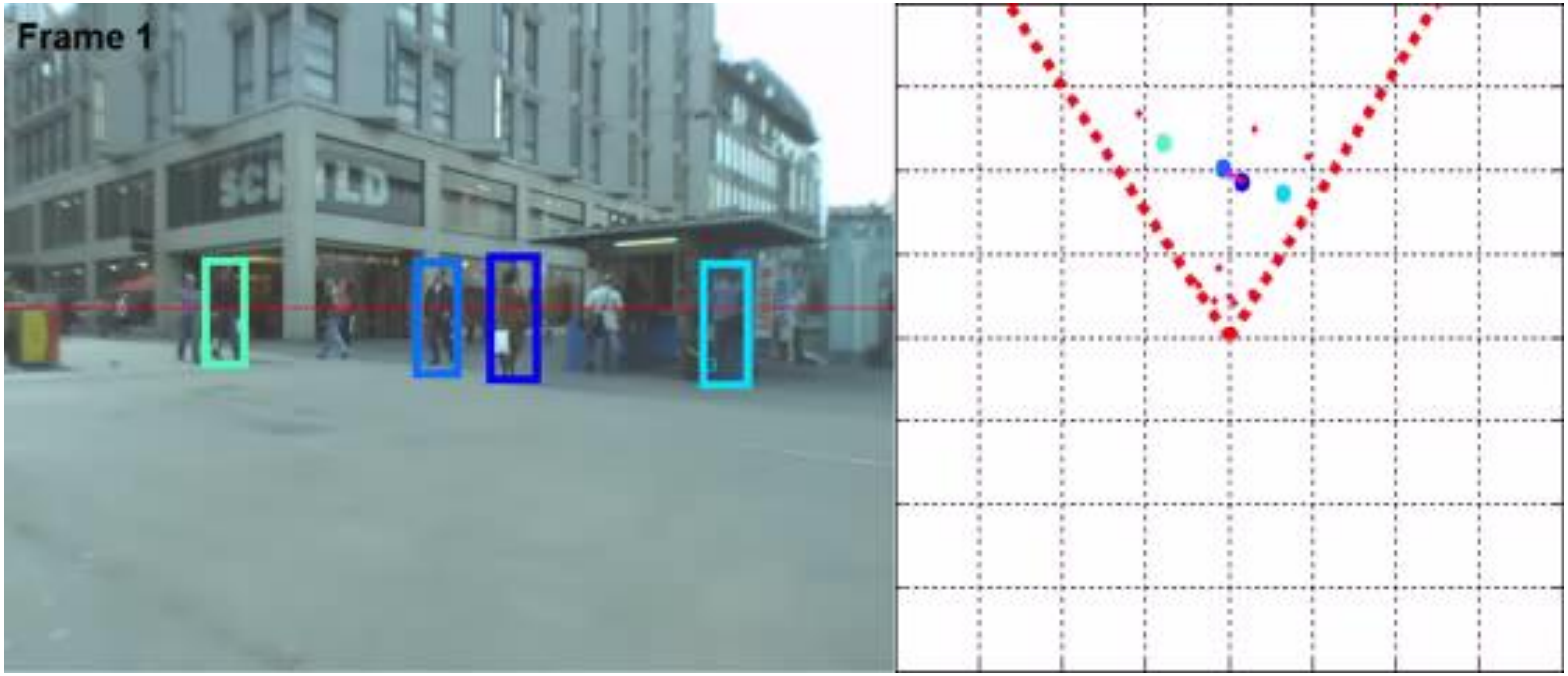
DOWNLOAD http://www.micc.unifi.it/pernici/

Tracking objects

Tracking objectsReal-Time Facial Feature Tracking on a Mobile Device
P. A. Tresadern, M. C. Ionita, T. F. Cootes in IJCV (2012)

Fig. 1 Facial feature tracking running in real-time on the Nokia N900 smartphone. A video is available from http://www.youtube.com/watch?v=Y86rOh1Y_kk

Joint tracking and 3D localization



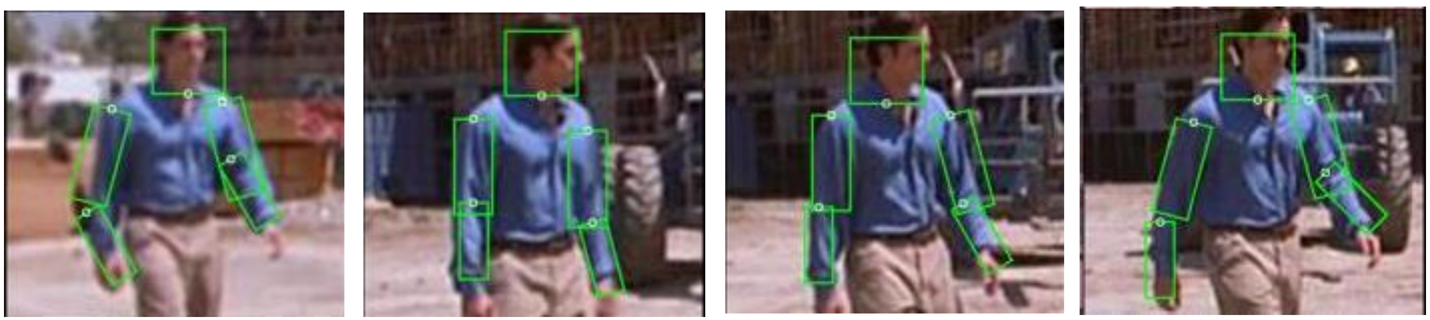
W. Choi & K. Shahid & S. Savarese WMC 2009
W. Choi & S. Savarese , ECCV, 2010

Tracking and Virtual Reality insertions

"Server-side object recognition and client-side object tracking for mobile augmented reality", Stephan Gammeter , Alexander Gassmann, Lukas Bossard, Till Quack, and Luc Van Gool, CVPR-W, 2010

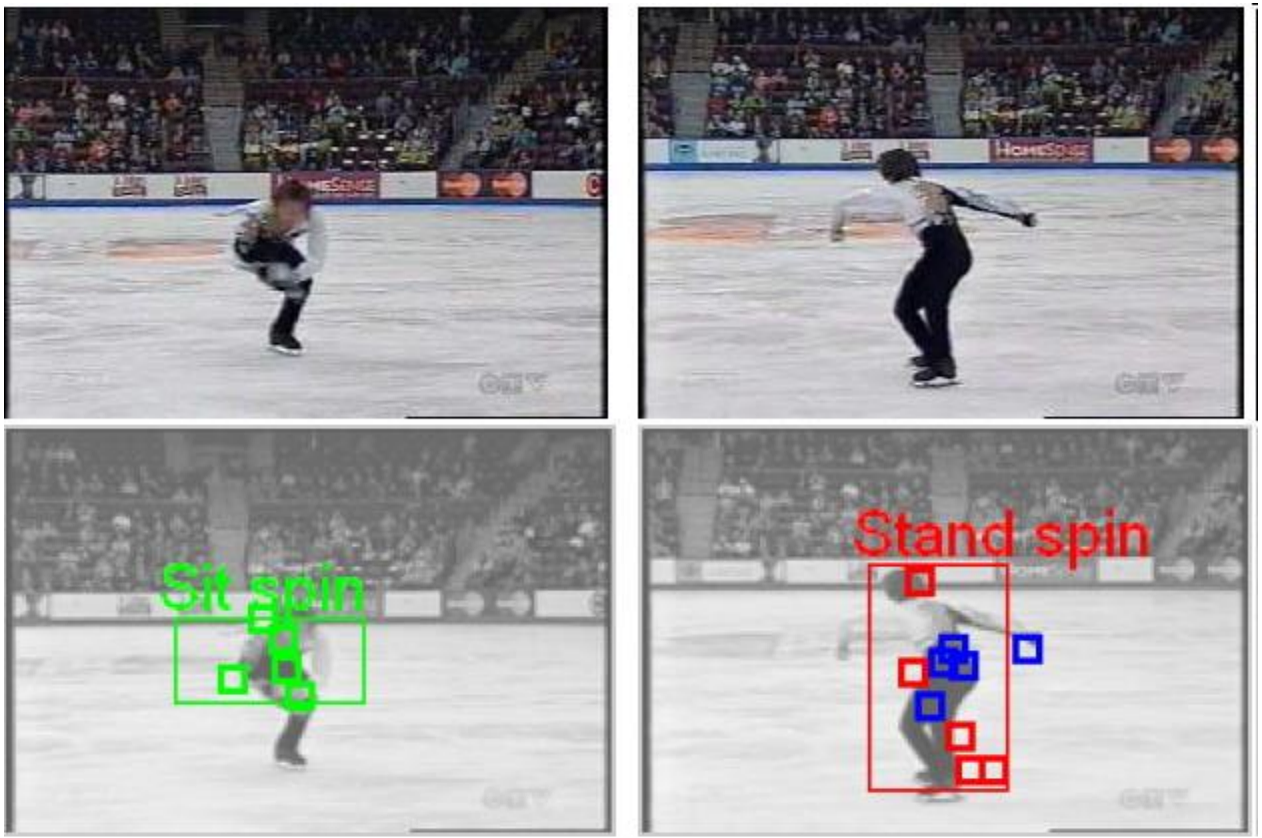
Tracking body parts

Cascaded Models for Articulated Pose Estimation, B Sapp, A Toshev, B Taskar, Computer Vision–ECCV 2010, 406-420



Courtesy of Benjamin Sapp

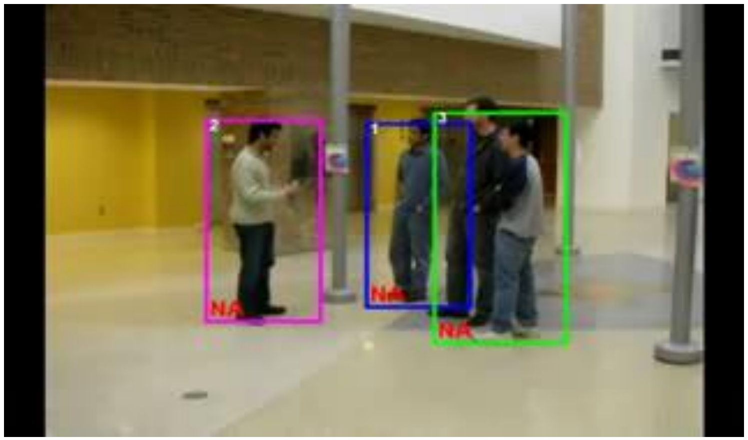
Recognizing events and activities

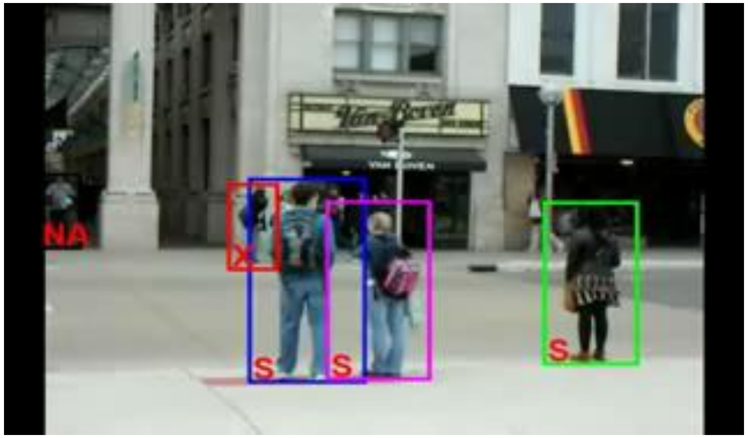


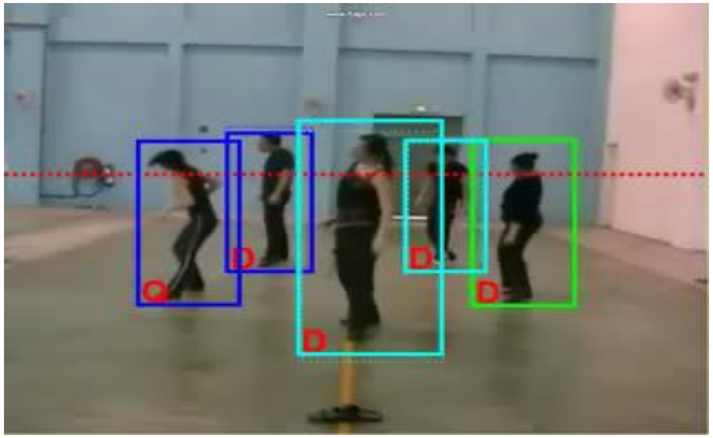
Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, **Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words**, *(BMVC)*, Edinburgh, 2006.

Recognizing group activities

Crossing – Talking – Queuing – Dancing – jogging







Choi & Savarese, CVPR 11
Choi & Savarese, ECCV 2012

X: Crossing, S: Waiting, Q: Queuing,
W: Walking, T: Talking, D: Dancing

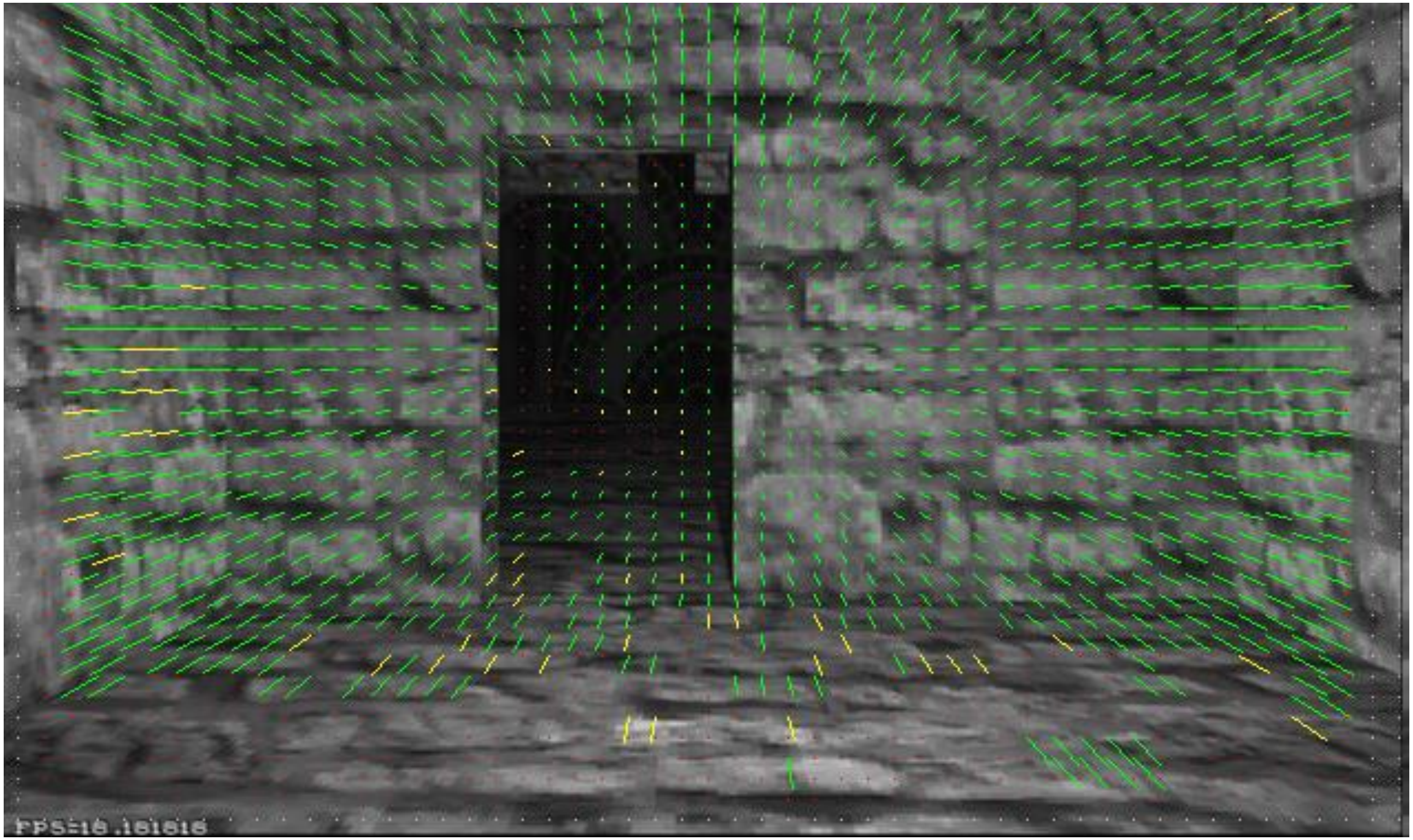
Motion estimation techniques

- 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

Vector field function of the spatio-temporal image brightness variations



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

Optical flow

Vector field function of the spatio-temporal image brightness variations

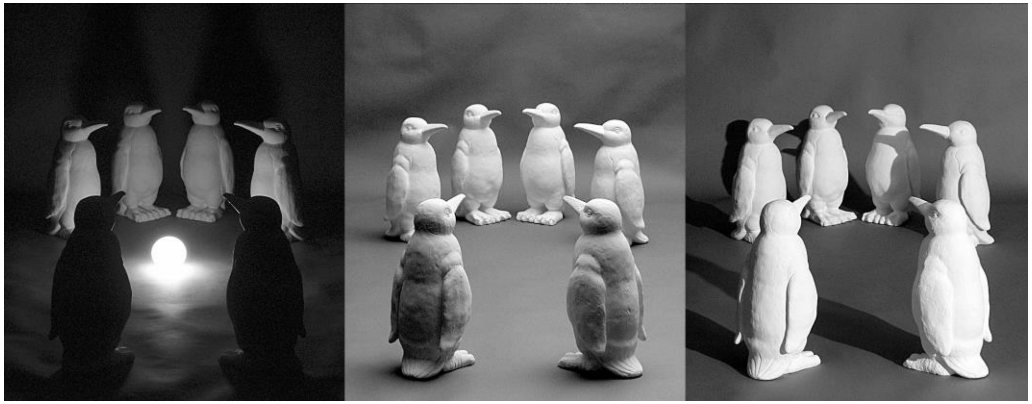
http://www.youtube.com/watch?v=JlLkkom6tWw

Optical flow

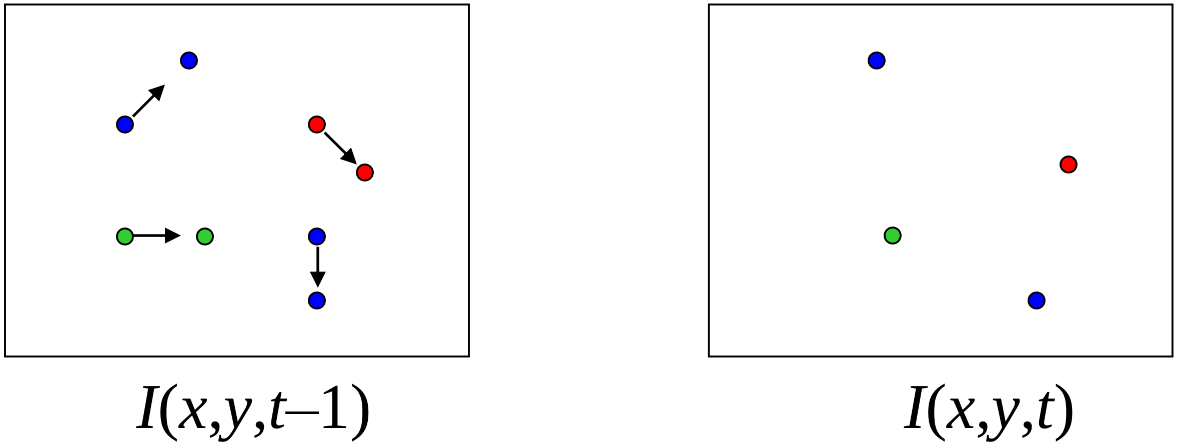
Definition: optical flow is the *apparent* motion of brightness patterns in the image

GOAL: Recover image motion at each pixel by optical flow

Note: apparent motion can be caused by lighting changes without any actual motion



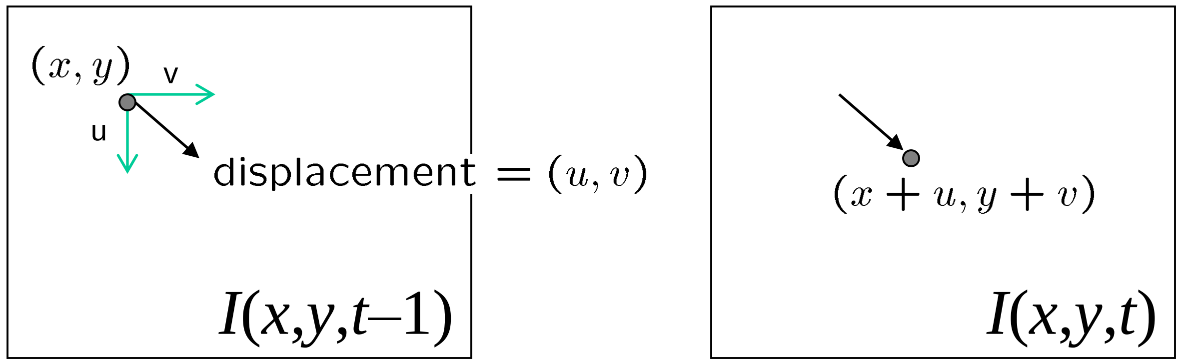
Estimating optical flow



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:

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

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

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

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)

Adding constraints….

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

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

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

Lucas-Kanade flow

Overconstrained linear system:

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

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

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?

- $\mathbf{A^T A}$ should be invertible
- 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?

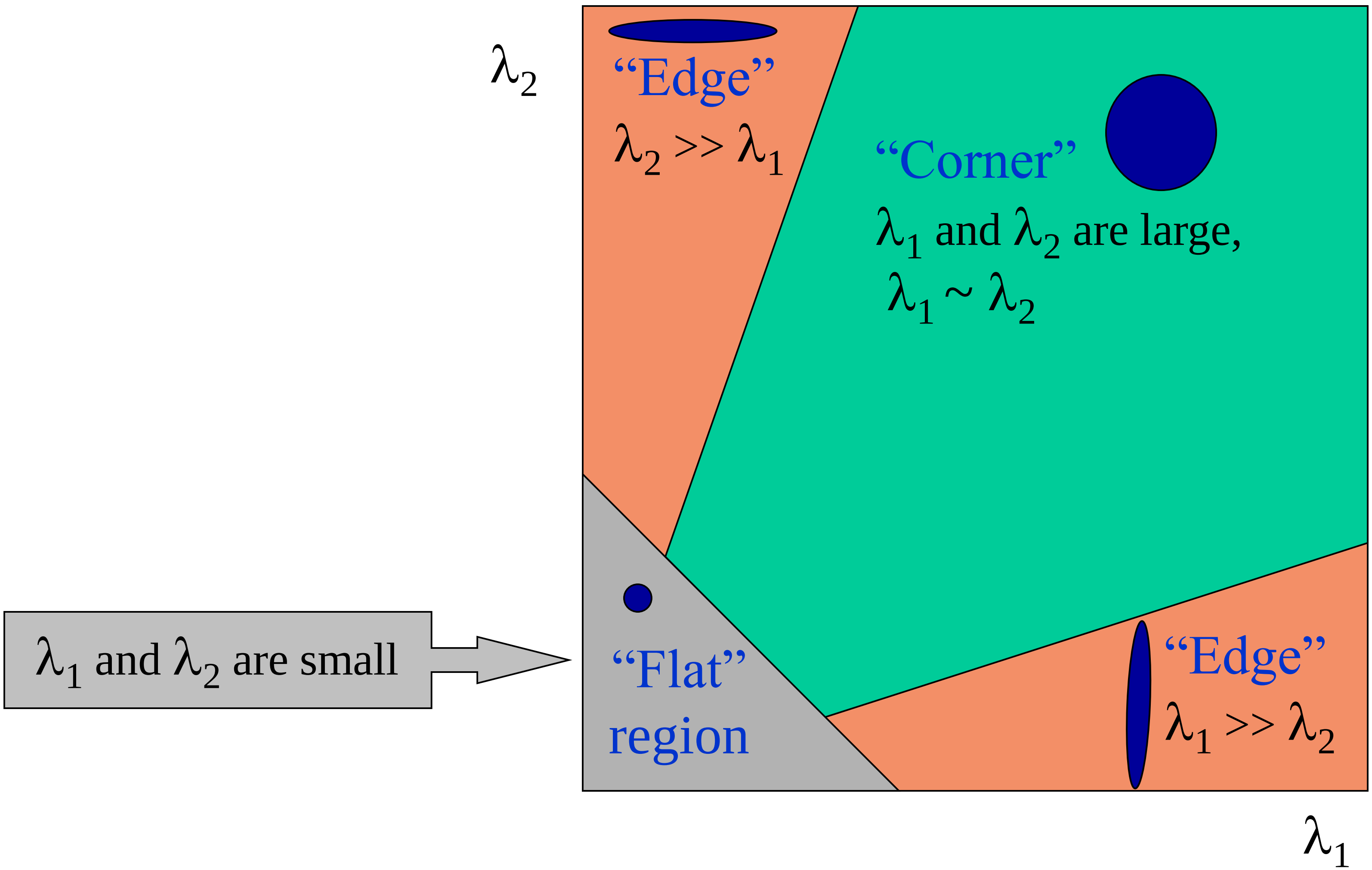
$M = A^T A$ is the *second moment matrix* !
(Harris corner detector…)

$$M = A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

- 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

Interpreting the eigenvalues

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



Low-texture region

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

– gradients have small magnitude

– small λ_1, small λ_2

Edge

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

– gradients very large or very small

– large λ_1, small λ_2

High-texture region

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

– gradients are different, large magnitudes

– large λ_1, large λ_2

What are good features to track?

J. Shi and C. Tomasi (June 1994). Good Features to Track. 9th IEEE Conference on Computer Vision and Pattern Recognition. Springer.

Can we measure "quality" of features from just a single image

Good features to track:

- Harris corners (guarantee small error sensitivity)

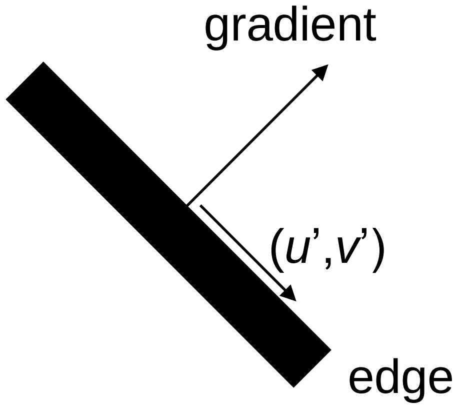
Bad features to track:

- Image points when either λ_1 or λ_2 (or both) is small (i.e., edges or uniform textured regions)

Ambiguities in tracking a point on a line

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

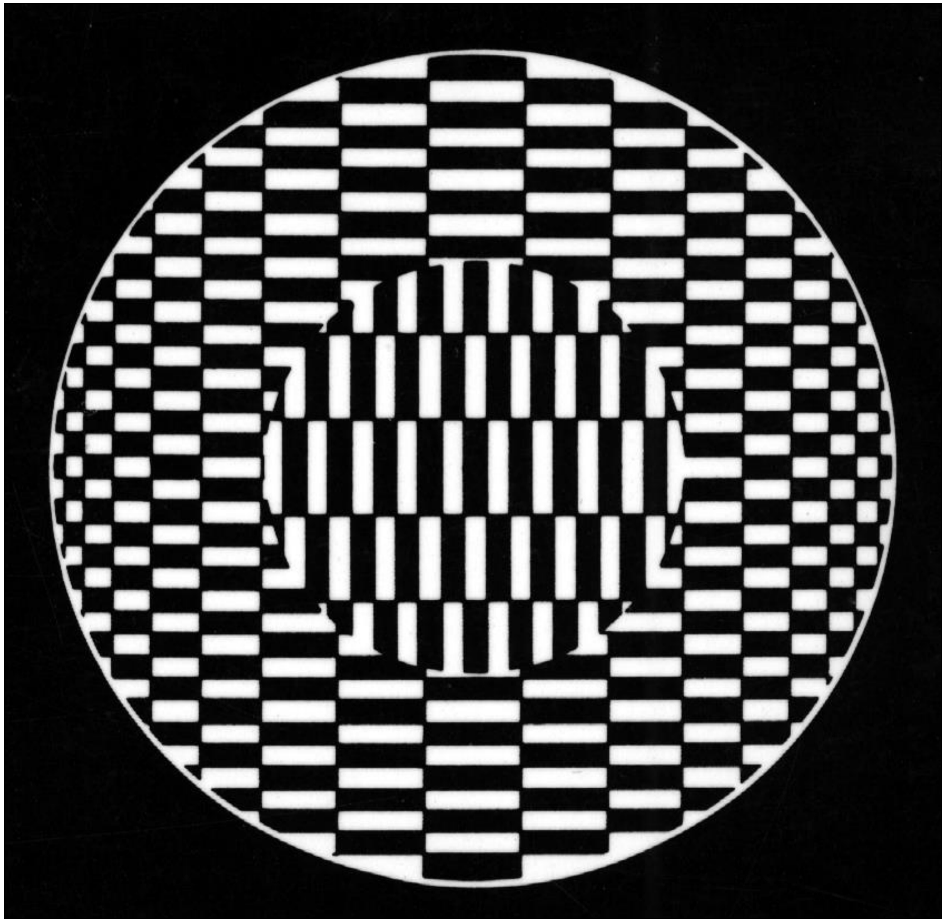
This equation $\nabla \mathrm{I} \cdot \left[\mathrm{u'} \;\; \mathrm{v'}\right]^{\mathrm{T}} = 0$ is always satisfied when (*u'*, *v'*) is perpendicular to the image gradient



The barber pole illusion

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

Aperture problem cont'd



* From Marc Pollefeys COMP 256 2003

Motion estimation techniques

Optical flow

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

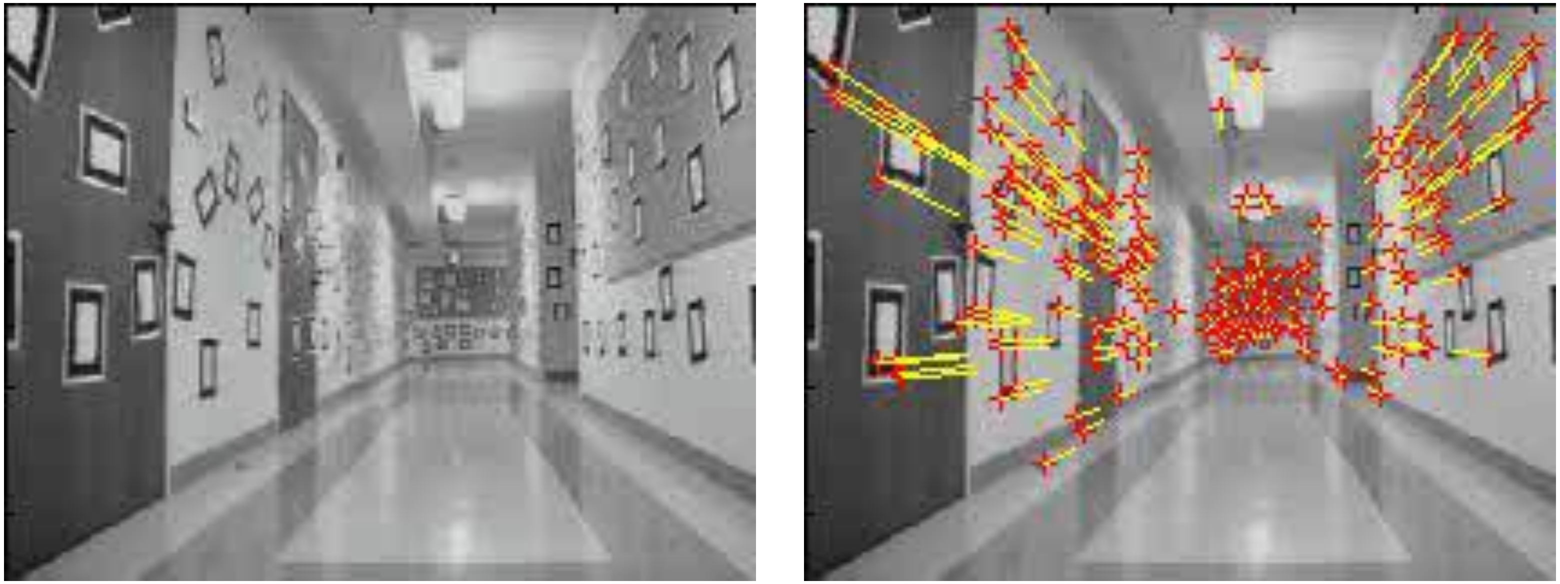
Feature-tracking

- Extract visual features (corners, textured areas) and “track” them over multiple frames

  - Shi-Tomasi feature tracker
  - Tracking with dynamics

- Implemented in Open CV

Tracking features



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

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

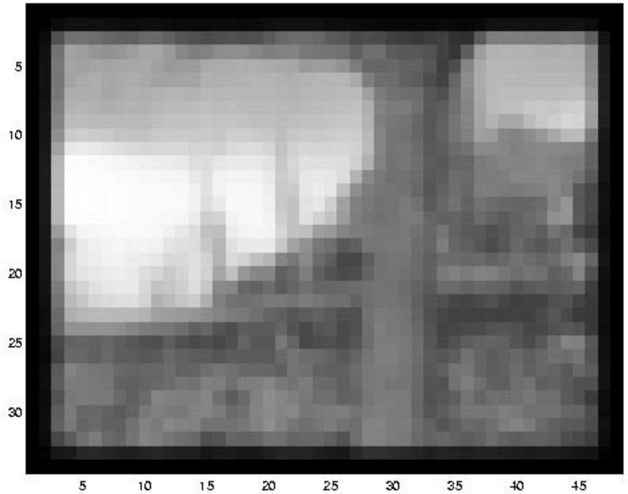
Revisiting the small motion assumption

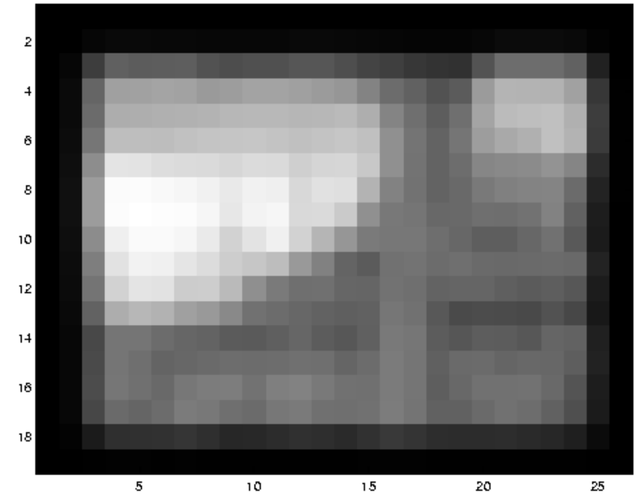
Is this motion small enough?

- 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!

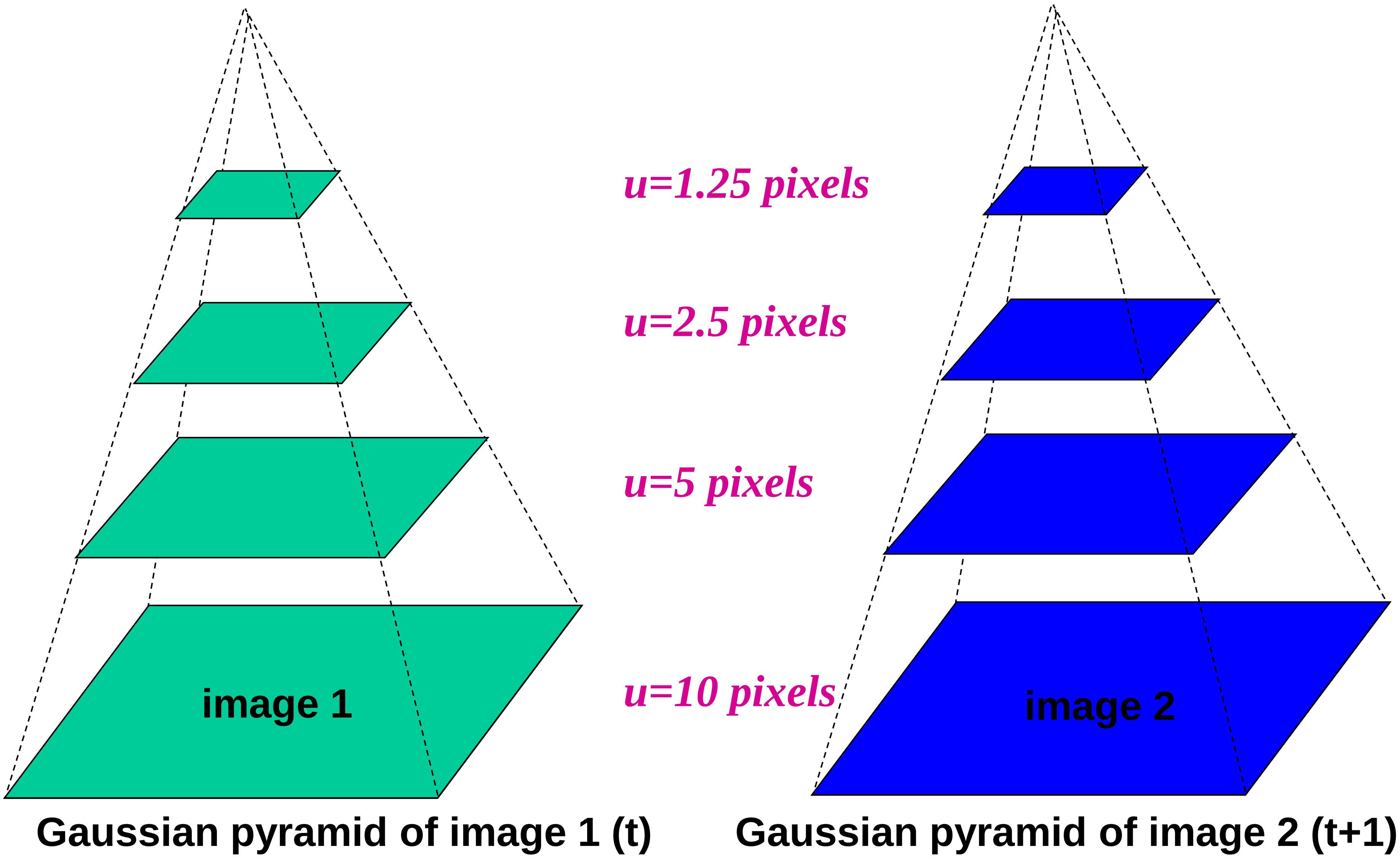




* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

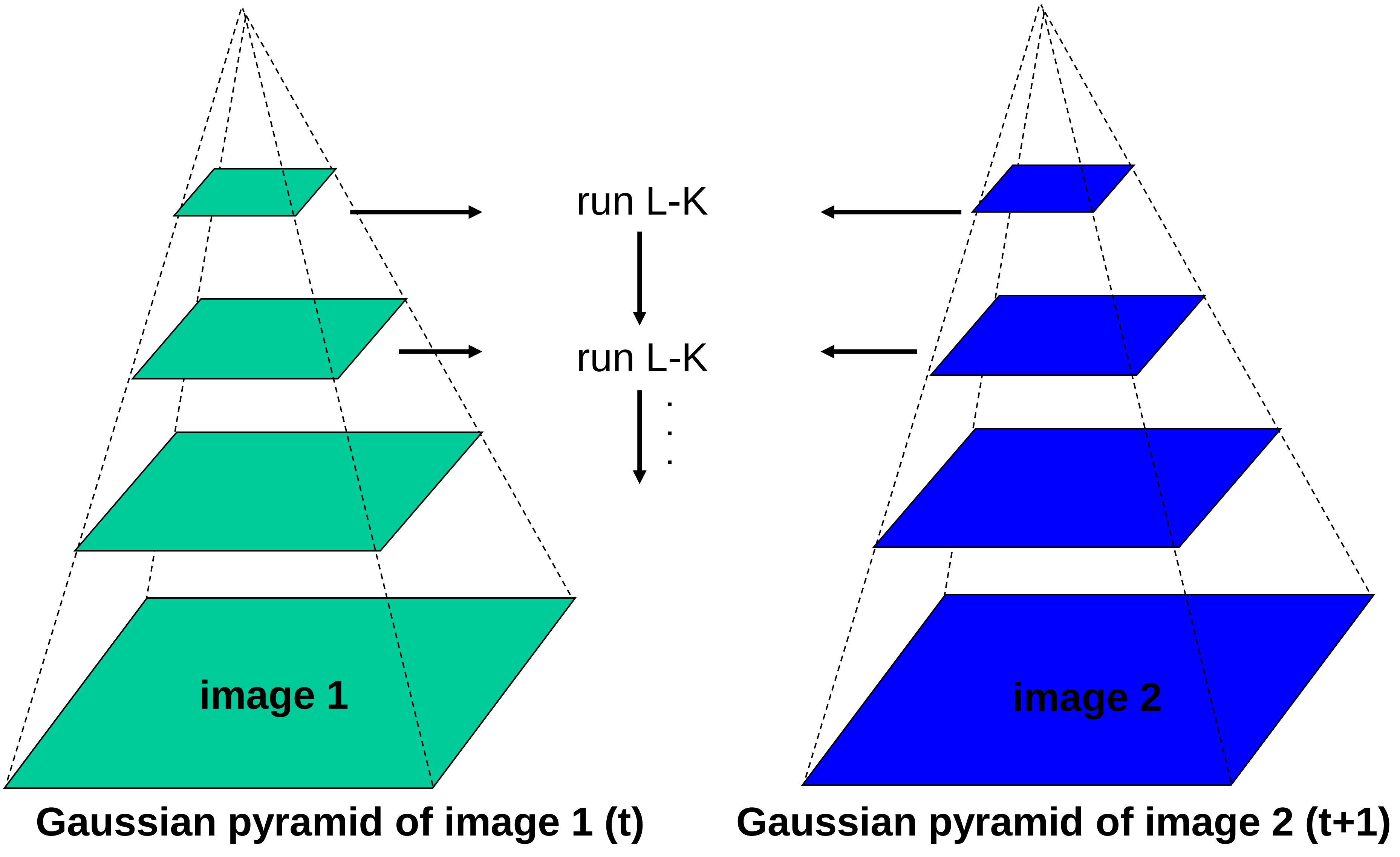
Coarse-to-fine optical flow estimation

JY. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature TrackerDescription of the algorithm, Tech. Report: http://robots.stanford.edu/cs223b04/algo_tracking.pdf

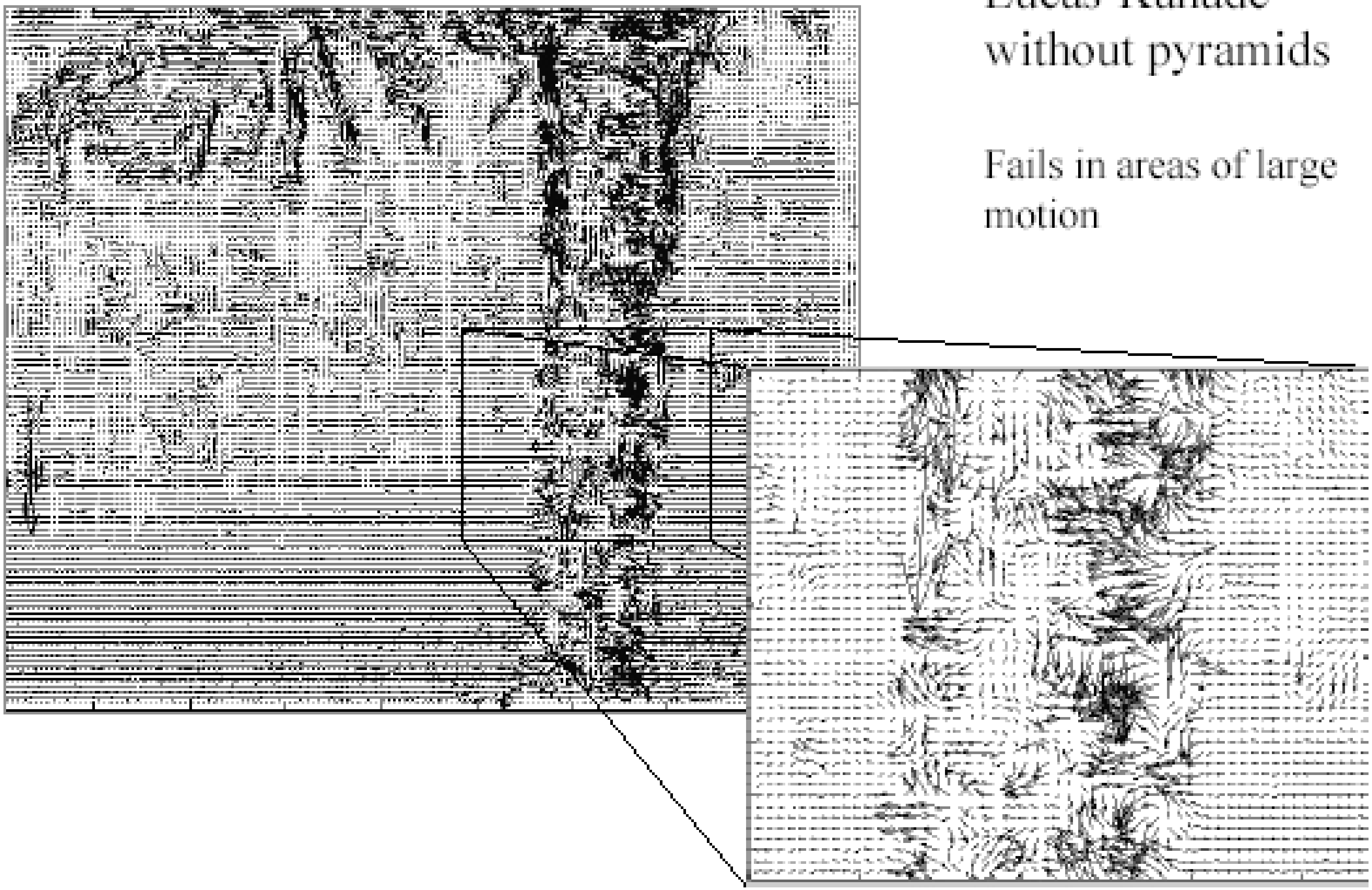


Coarse-to-fine optical flow estimation

JY. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature TrackerDescription of the algorithm,
Tech. Report: http://robots.stanford.edu/cs223b04/algo_tracking.pdf



Optical Flow Results

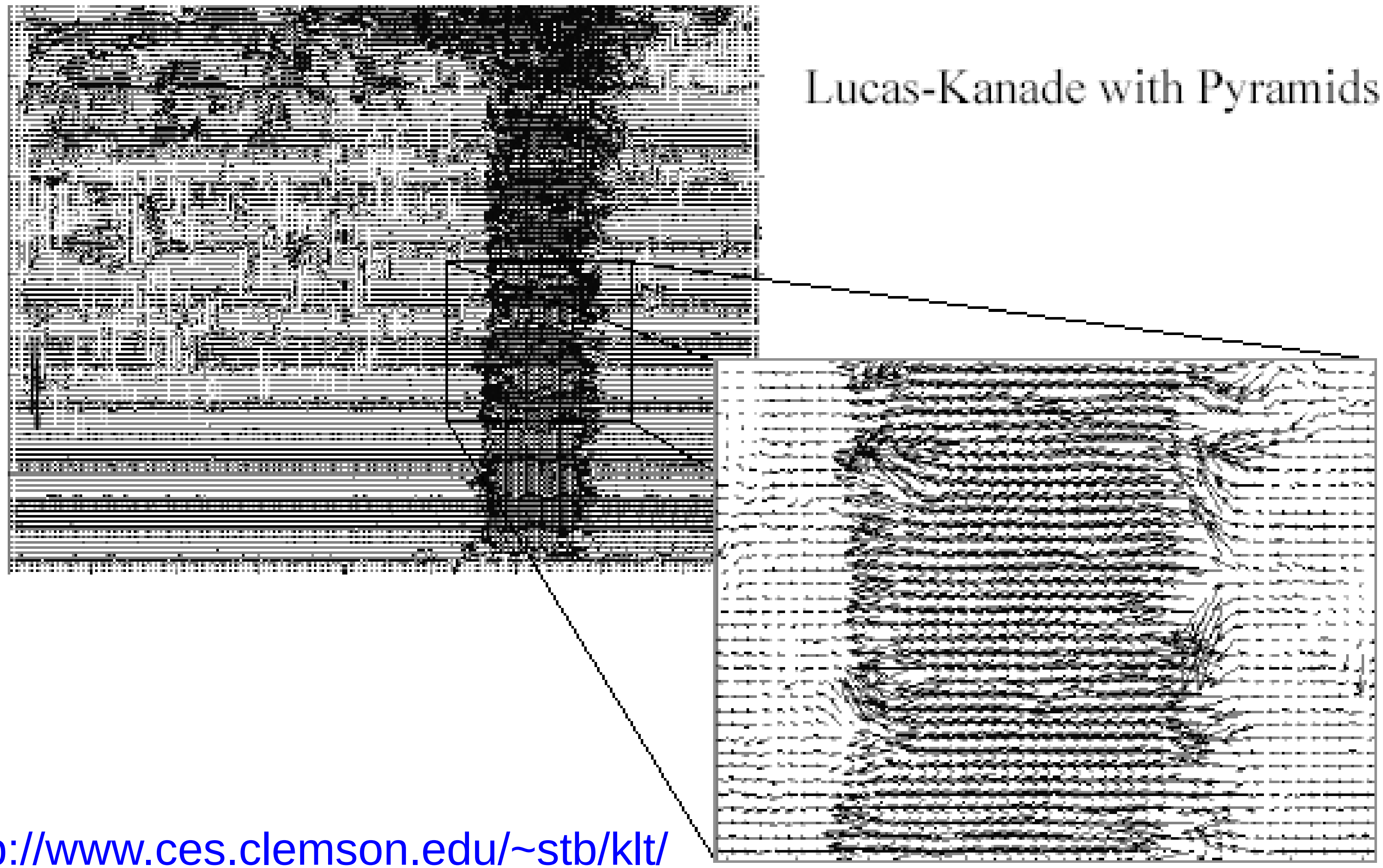


Lucas-Kanade
without pyramids

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

Motion segmentation

How do we represent the motion in this scene?

Motion segmentation

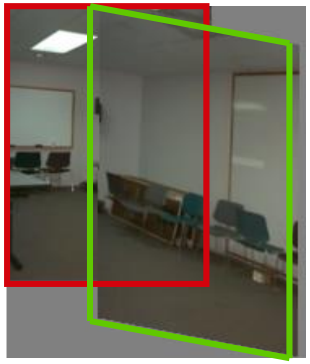
J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

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

Affine motion

$$u(x, y) = a_1 + a_2 x + a_3 y$$

$$v(x, y) = a_4 + a_5 x + a_6 y$$



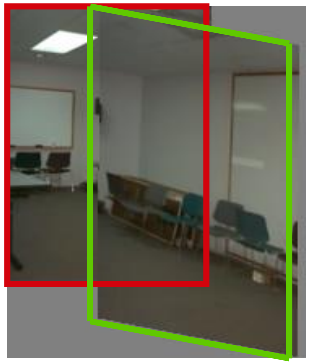
Substituting into the brightness constancy equation:

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

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:

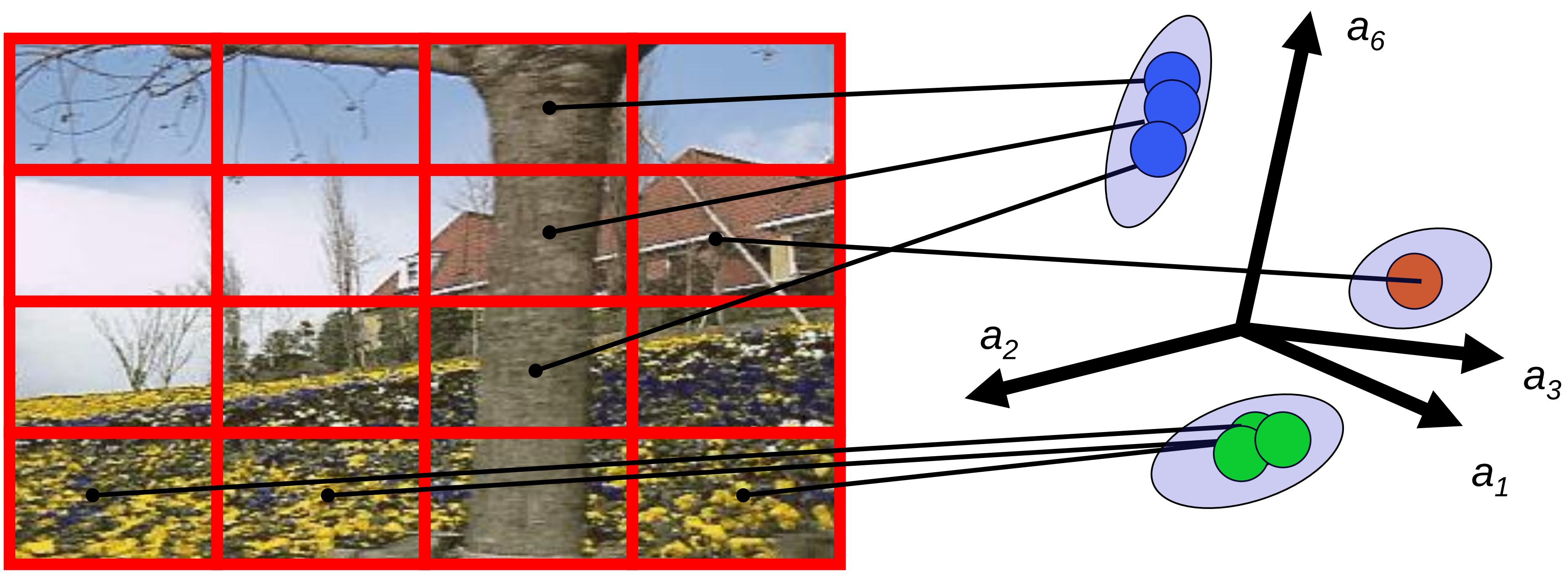
$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

- Each pixel provides 1 linear constraint in 6 unknowns
- If we have at least 6 pixels in a neighborhood, $a_1 \ldots a_6$ can be found by least squares minimization:

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

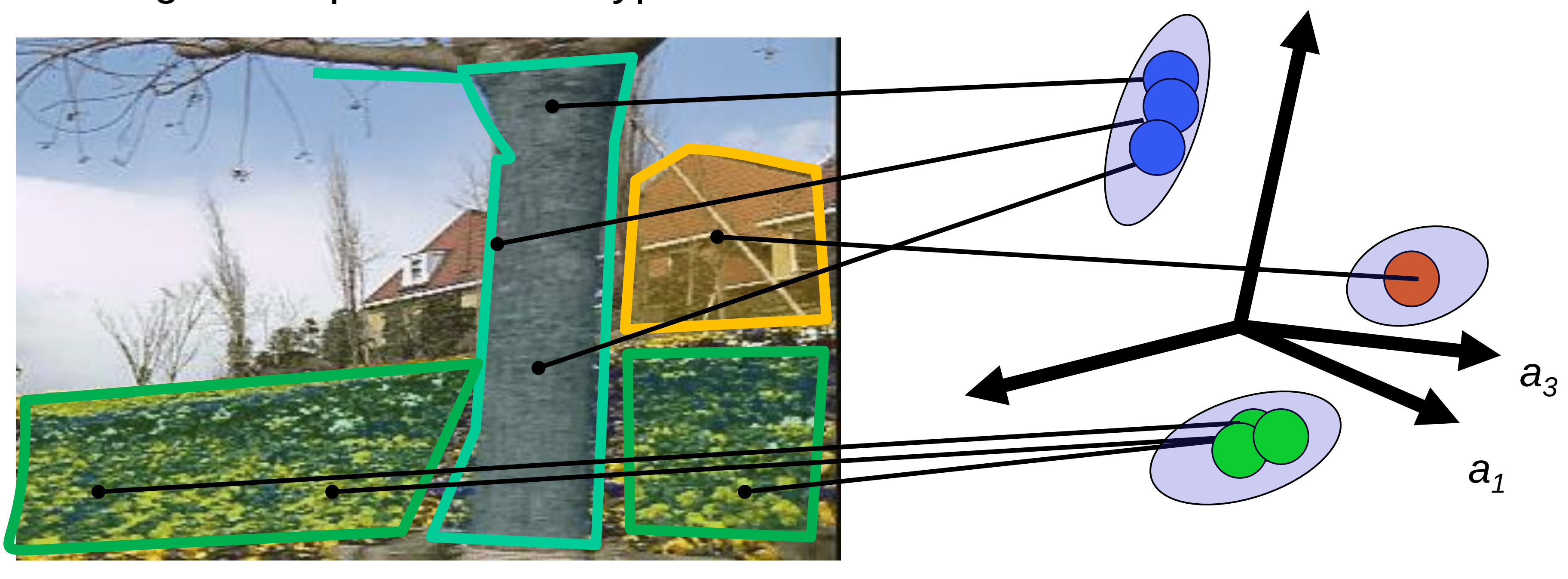
How do we estimate the layers?

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
2. Map into motion parameter space
3. 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

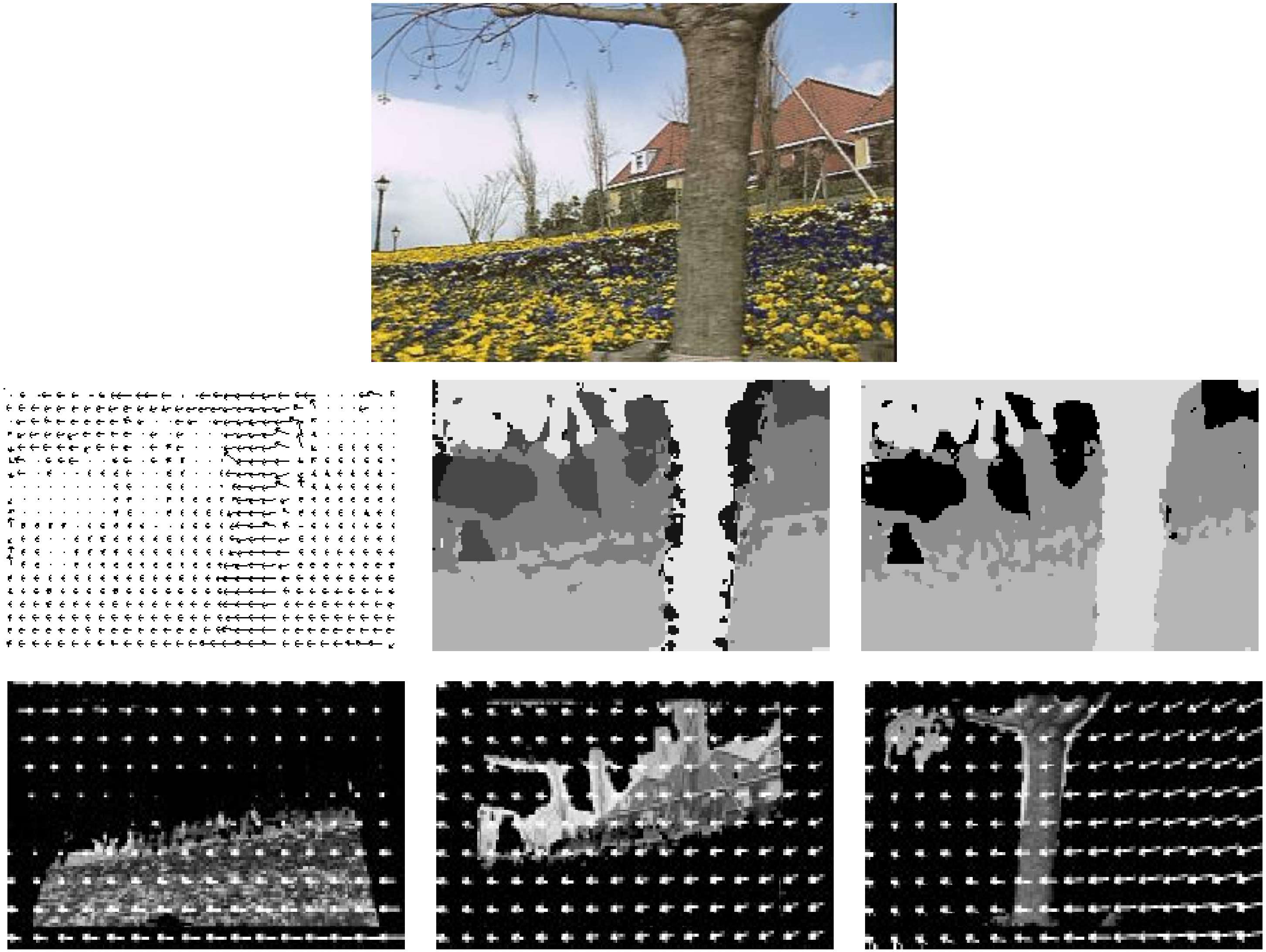


How do we estimate the layers?

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
2. Map into motion parameter space
3. 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
4. Assign each pixel to best hypothesis --- iterate



Example result



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

CS231M · Mobile Computer Vision

Next lecture:

Neural networks and decision trees for machine vision