

## COMPUTER VISION LECTURE 21 – MOTION

Prof. Dr. Francesco Maurelli  
2019-11-12

1. Optical flow
2. Lucas-Kanade method
3. Horn-Schunck method
4. Pyramids for large motion
5. Common fate
6. Applications

**Reading:** [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005]

<http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf>

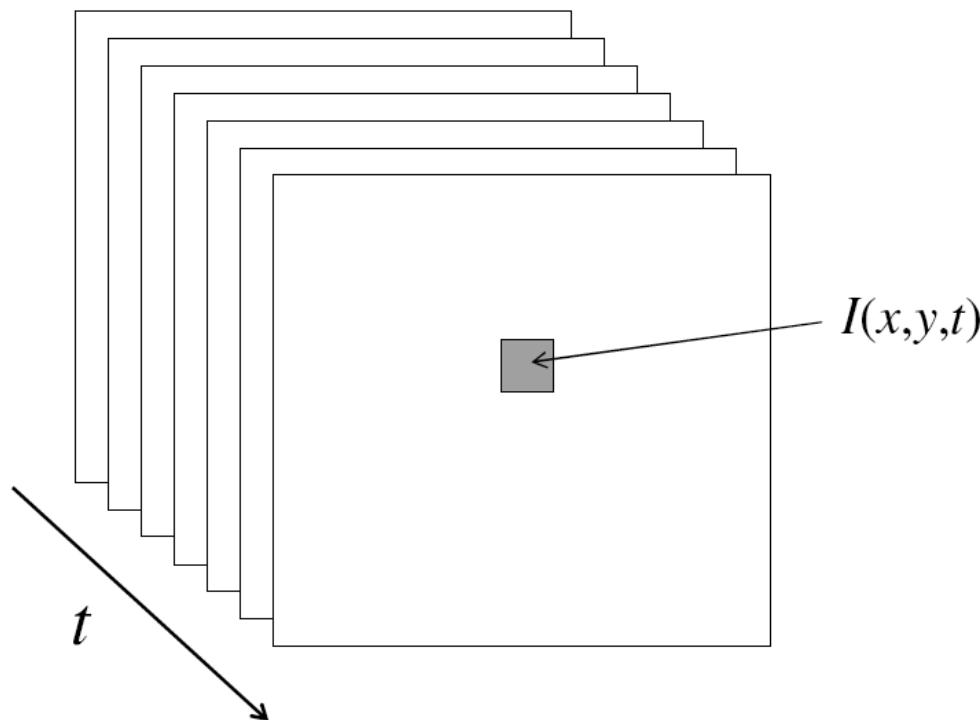
1. Optical flow
2. Horn-Schunk method
3. Lucas-Kanade method
4. Pyramids for large motion
5. Common fate
6. Applications

**Reading:** [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005]

<http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf>

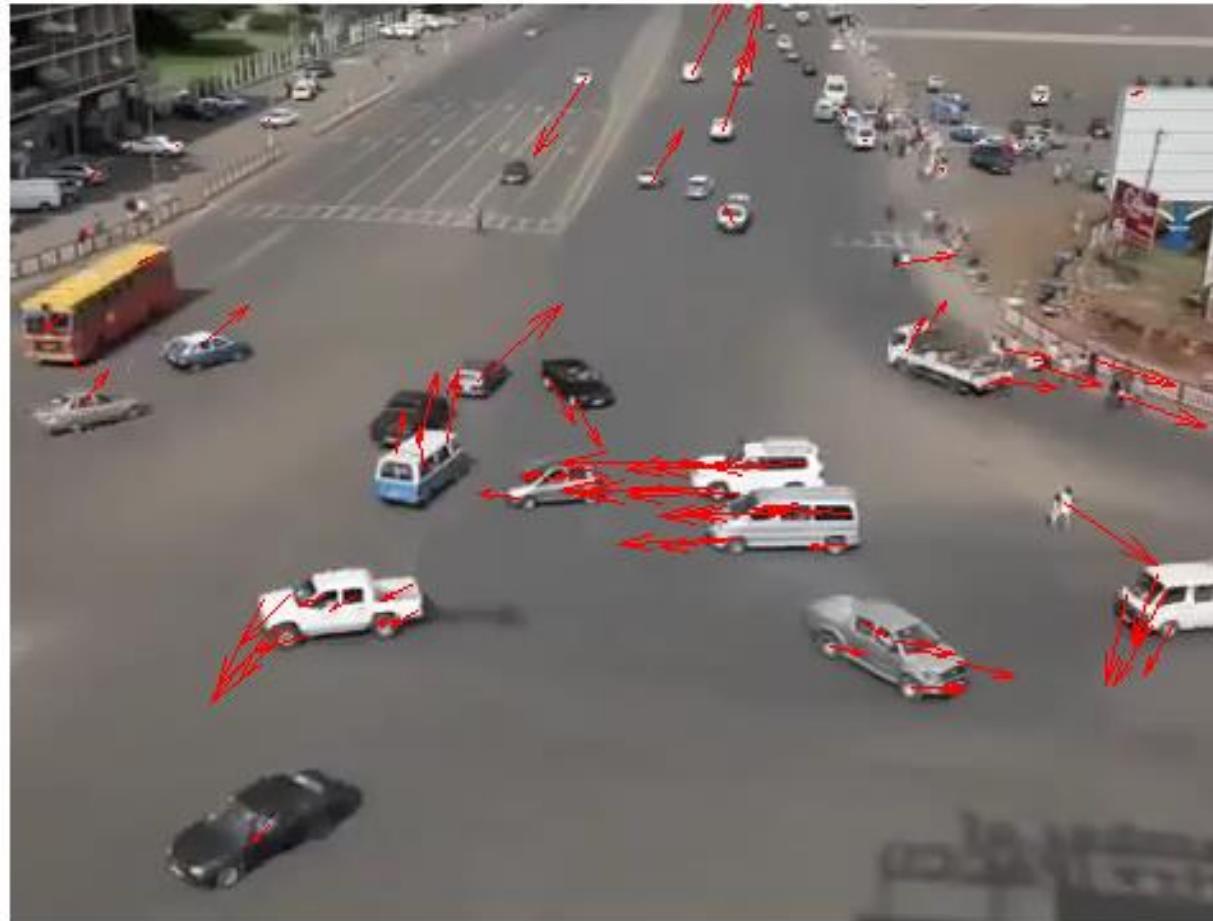
1. A video is a sequence of frames captured over time
2. Now our image data is a function of space ( $x, y$ ) and time ( $t$ )



## WHY IS MOTION USEFUL?



## WHY IS MOTION USEFUL?



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

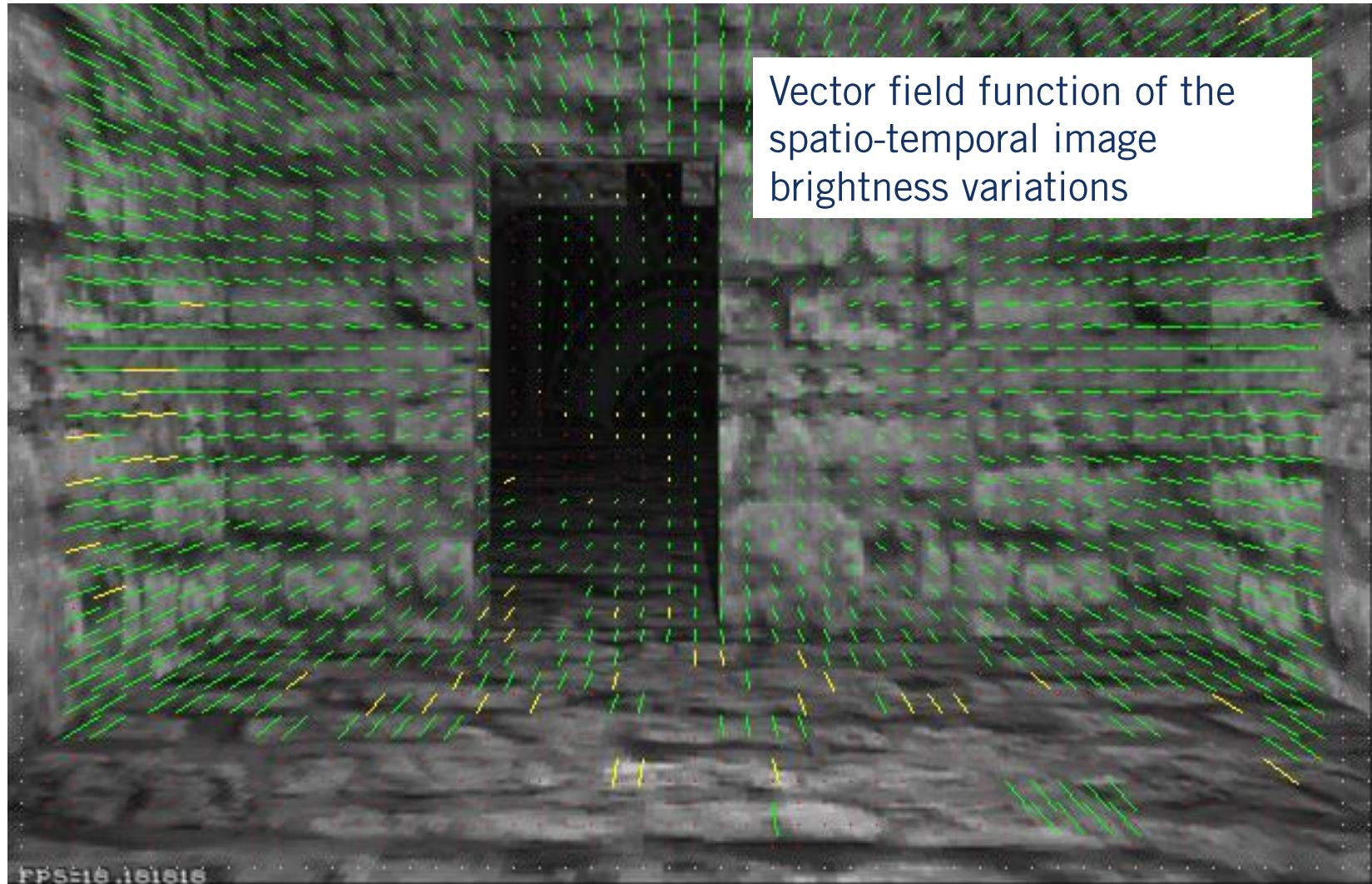
Note: apparent motion can be caused by lighting changes without any actual motion

- Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

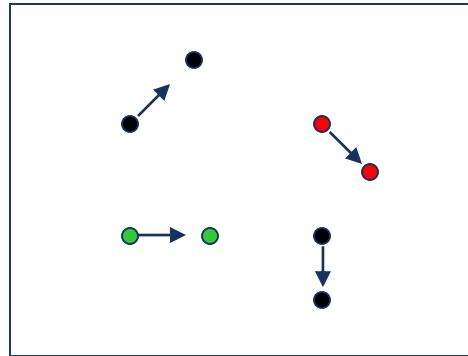
## GOAL

Recover image motion at each pixel from optical flow

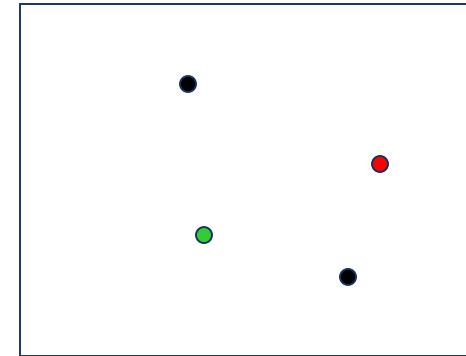
## OPTICAL FLOW



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



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

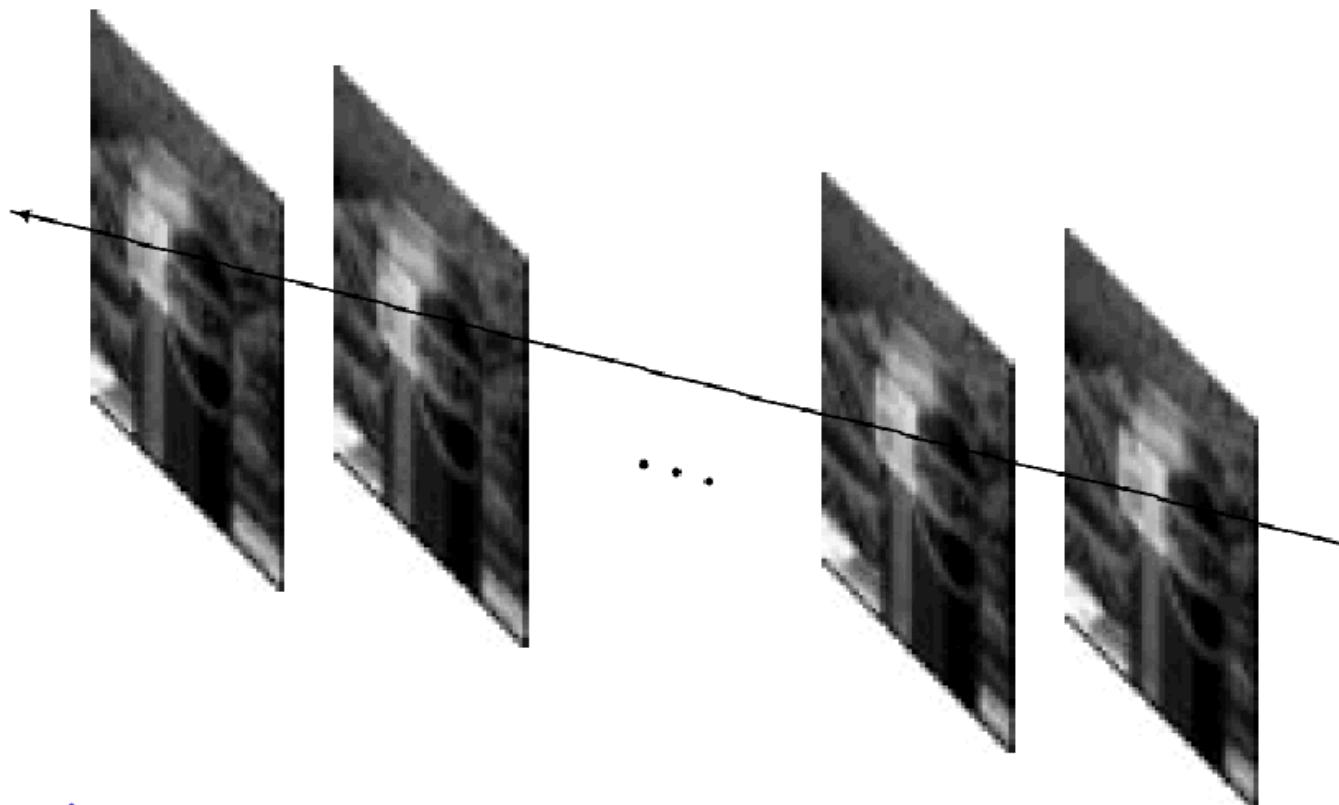


$I(x,y,t)$

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

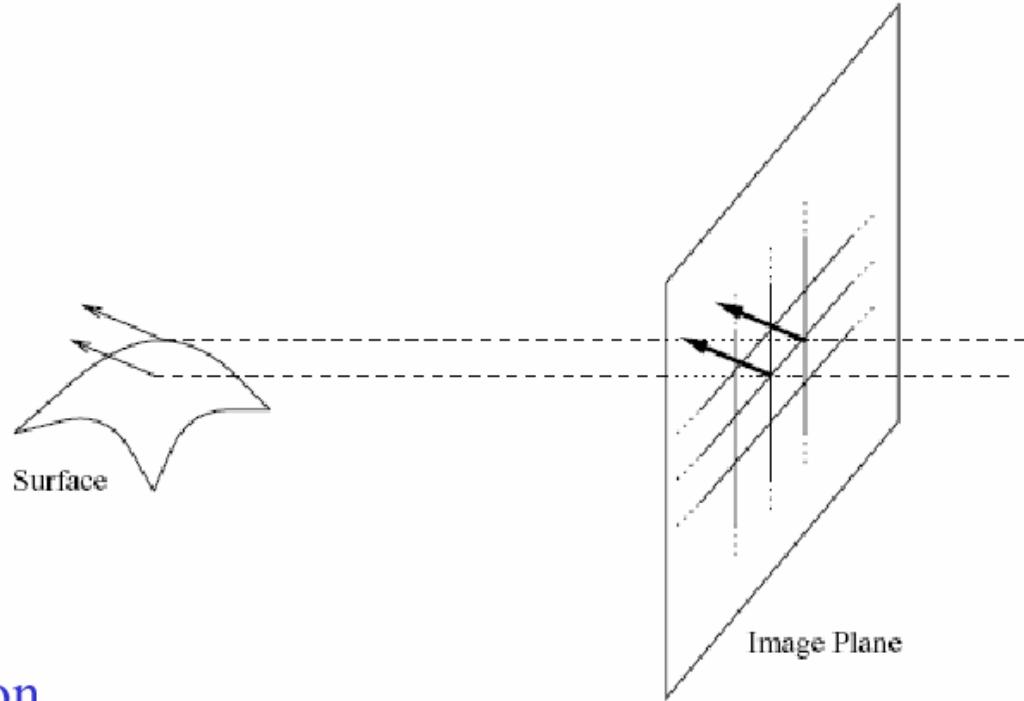
### Key assumptions

- **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



Assumption:

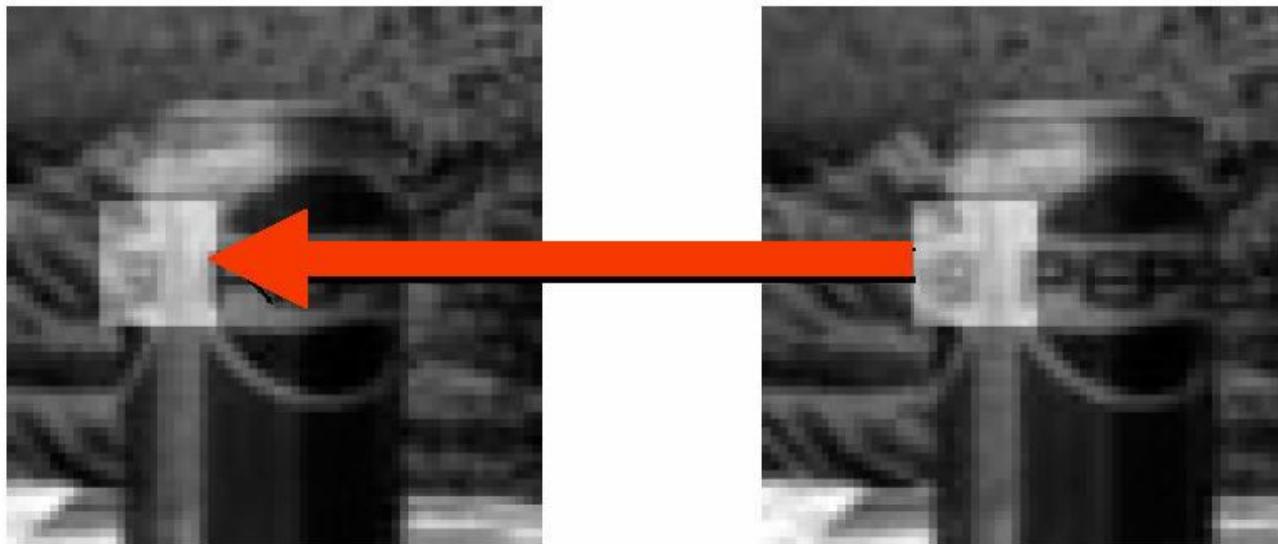
The image motion of a surface patch changes gradually over time.



### 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.

## KEY ASSUMPTIONS: BRIGHTNESS CONSTANCY



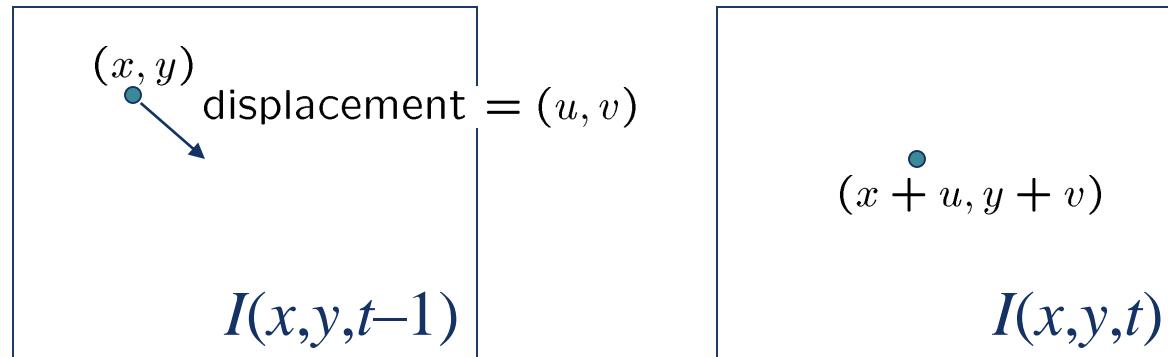
### Assumption

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)

## 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:

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

Image derivative along x

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

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

# THE BRIGHTNESS CONSTANCY CONSTRAINT

C   https://users.cs.cf.ac.uk/Dave.Marshall/Vision\_lecture/node47.html 

[Next](#) [Up](#) [Previous](#)

Next: [Further Constraints](#) Up: [Optical Flow](#) Previous: [Introduction](#)

## Optical Flow Constraint Equation

Let us suppose that the image intensity is given by  $I(x,y,t)$ , where the intensity is now a function of time,  $t$ , as well as of  $x$  and  $y$ .

At a point a small distance away, and a small time later, the intensity is

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \dots, \quad (61)$$

where the dots stand for higher order terms.

Now, suppose that part of an object is at a position  $(x,y)$  in the image at a time  $t$ , and that by a time  $dt$  later it has moved through a distance  $(dx,dy)$  in the image.

Furthermore, let us suppose that the intensity of that part of the object is just the same in our image before and afterwards.

Provided that we are justified in making this assumption, we then have that

$$I(x + dx, y + dy, t + dt) = I(x, y, t), \quad (62)$$

and so

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \dots = 0. \quad (63)$$

However, dividing through by  $dt$ , we have that

$$\frac{dx}{dt} = u, \quad \frac{dy}{dt} = v, \quad (64)$$

as these are the speeds the object is moving in the  $x$  and  $y$  directions respectively. Thus, in the limit that  $dt$  tends to zero, we have

$$-\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v, \quad (65)$$

which is called the *optical flow constraint equation*.

## THE BARBER POLE ILLUSION



1. Optical flow
2. Lucas-Kanade method
3. Horn-Schunck method
4. Pyramids for large motion
5. Common fate
6. 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

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

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

Overconstrained linear system:

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

$A_{25 \times 2} \quad d_{2 \times 1} \quad b_{25 \times 1}$

## 1. Overconstrained linear system

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

Least squares solution for  $d$  given by  $(A^T A)^{-1} d = A^T b$

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

The summations are over all pixels in the  $K \times K$  window

## CONDITIONS FOR SOLVABILITY

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

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

$A^T A$                                      $A^T b$

### When is This Solvable?

- $\mathbf{A}^T \mathbf{A}$  should be invertible
- $\mathbf{A}^T \mathbf{A}$  should not be too small due to noise
  - eigenvalues  $\lambda_1$  and  $\lambda_2$  of  $\mathbf{A}^T \mathbf{A}$  should not be too small
- $\mathbf{A}^T \mathbf{A}$  should be well-conditioned
  - $\lambda_1 / \lambda_2$  should not be too large ( $\lambda_1$  = larger eigenvalue)

Does this remind anything to you?

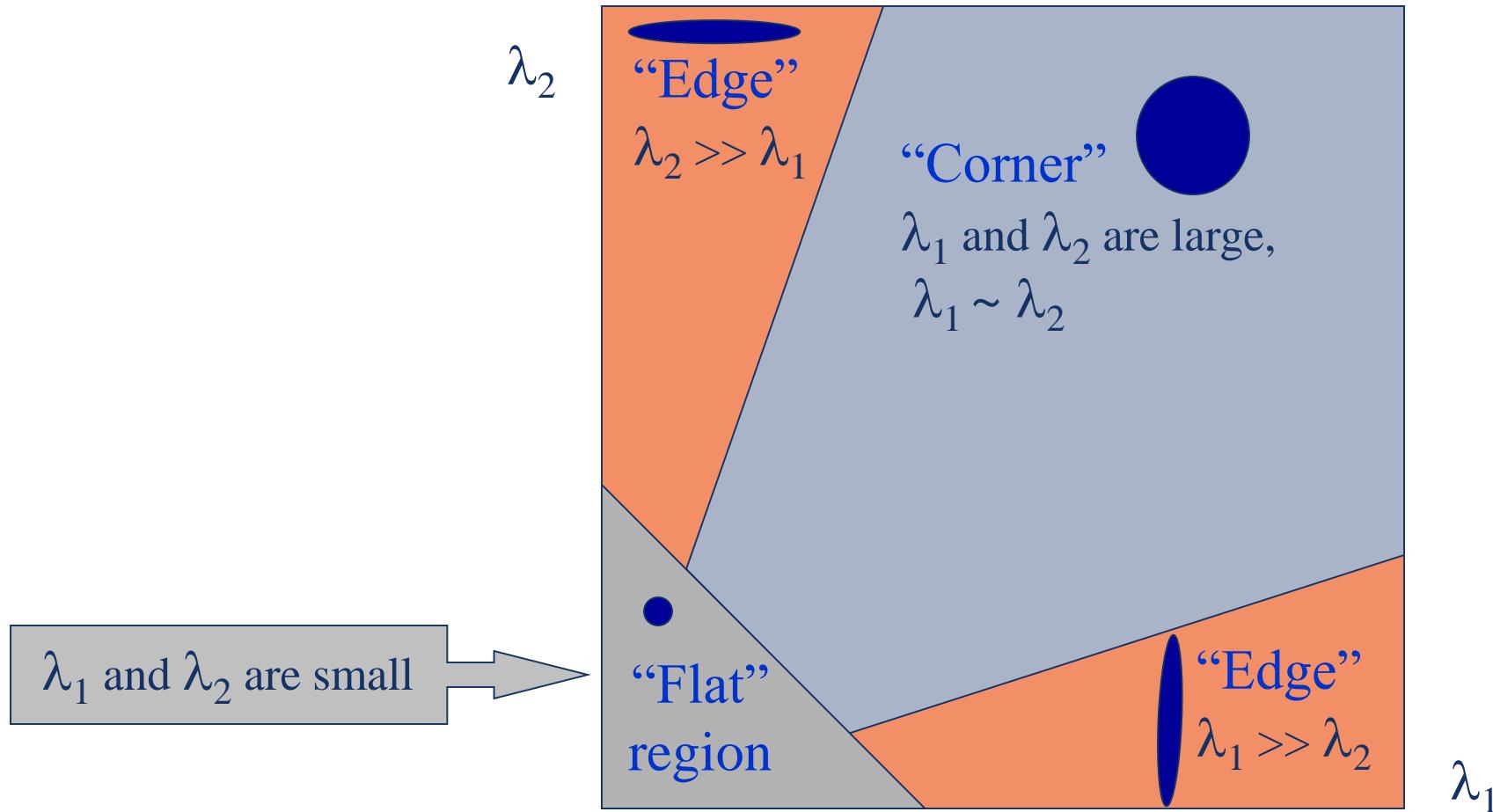
$M = A^T A$  is the *second moment matrix* !  
(Harris corner detector...)

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

Eigenvectors and eigenvalues of  $A^T A$  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

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





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

- gradients very large or very small
- large  $\lambda_1$ , small  $\lambda_2$



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

- gradients have small magnitude
- small  $\lambda_1$ , small  $\lambda_2$

## HIGH-TEXTURE REGION



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

- gradients are different, large magnitudes
- large  $\lambda_1$ , large  $\lambda_2$

What are the potential causes of errors in this procedure?

- Suppose  $A^T A$  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?

Recall our small motion assumption

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

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

- This is not exact

- To do better, we need to add higher order terms back in:

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

- 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

## 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

## 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?

Well, TV is based on illusory motion

- the set is stationary yet things seem to move

A uniform rotating sphere

- nothing seems to move, yet it is rotating

Changing directions or intensities of lighting can make things seem to move

Muscle movement can make some spots on a cheetah move opposite direction of motion.

- And infinitely more break downs of optical flow.

1. Optical flow
2. Lucas-Kanade method
3. Horn-Schunk method
4. Pyramids for large motion
5. Common fate
6. Applications

**Reading:** [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005]

<http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf>

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

$$E = \iint [(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2)] \, dx \, dy$$

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

$$E = \iint [(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2)] \, dx \, dy$$

The first part of the function is the brightness consistency.

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

$$E = \iint [(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2)] \, dx \, dy$$

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

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

$$E = \iint [(I_x u + I_y v + I_t)^2 - \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2)] \, dx \, dy$$

$\alpha$  is a regularization constant. Larger values of  $\alpha$  lead to smoother flow.

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

$$E = \iint [(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2)] \, dx \, dy$$

By taking the derivative with respect to  $u$  and  $v$ , we get the following 2 equations:

$$\begin{aligned} I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u &= 0 \\ I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v &= 0 \end{aligned}$$

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

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

Now we substitute  
in:

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

$$I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0$$

$$I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0$$

to get:

$$(I_x^2 + \alpha^2)u + I_x I_y v = \alpha^2 \bar{u} - I_x I_t$$

$$I_x I_y u + (I_y^2 + \alpha^2)v = \alpha^2 \bar{v} - I_y I_t$$

which is linear in  $u$  and  $v$  and can be solved for each pixel individually.

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:

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

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

---

## WHAT WE WILL LEARN TODAY?

1. Optical flow
2. Lucas-Kanade method
3. Horn-Schunk method
4. Pyramids for large motion
5. Common fate
6. Applications

## 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

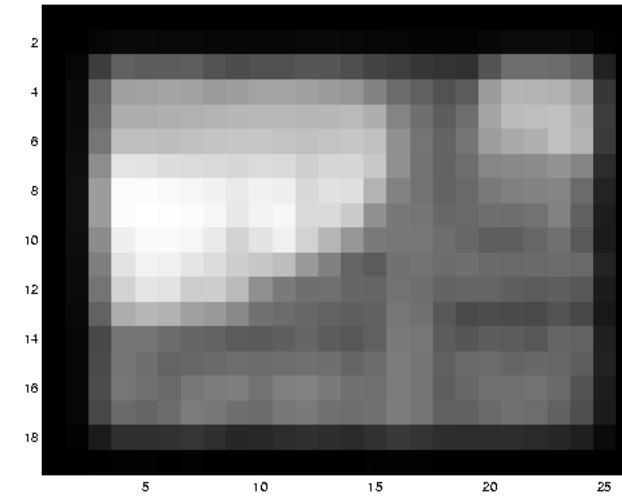
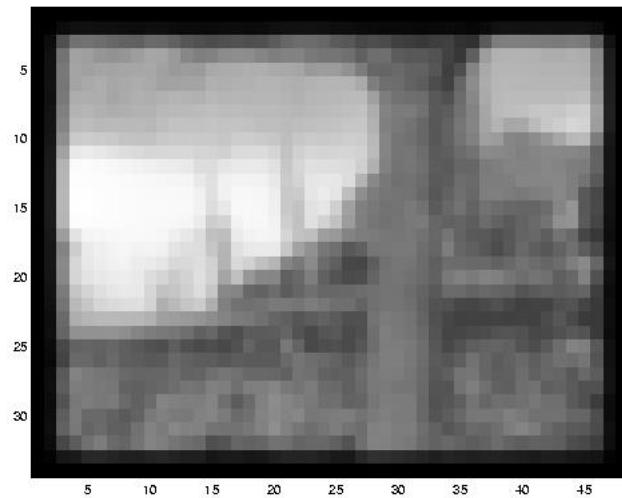
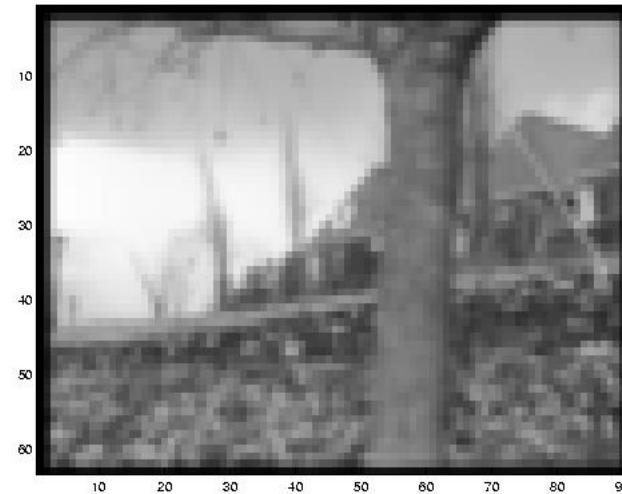
## REVISITING THE SMALL MOTION ASSUMPTION

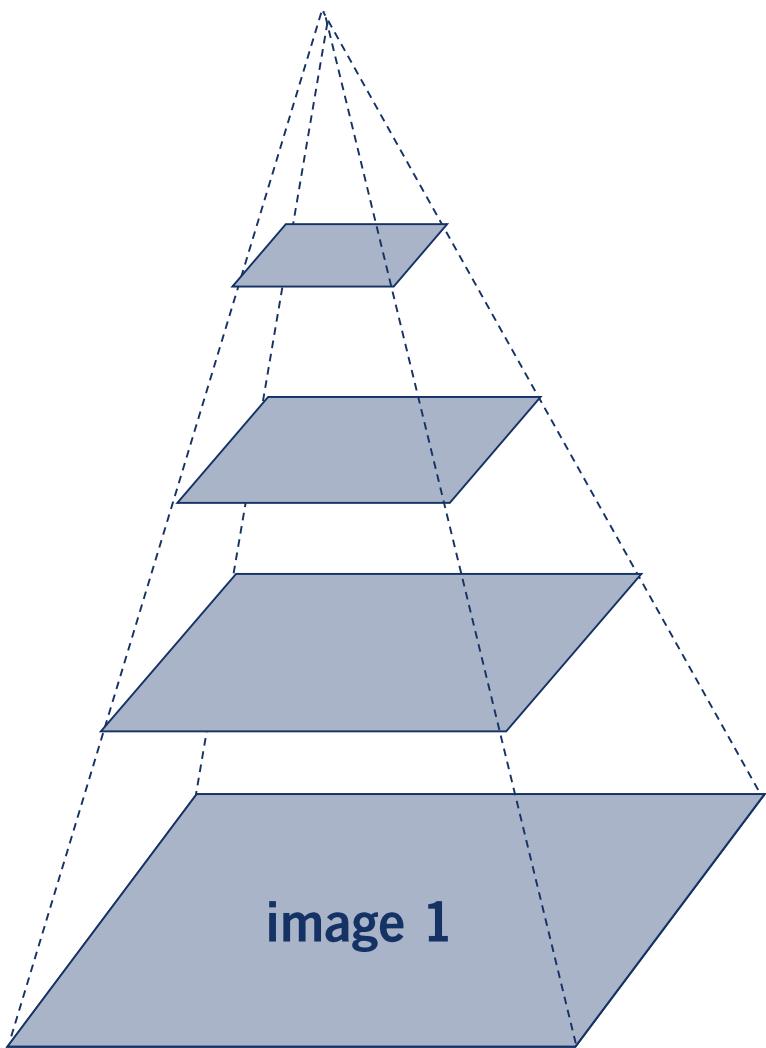


Is this motion small enough?

- Probably not—it's much larger than one pixel ( $2^{\text{nd}}$  order terms dominate)
- How might we solve this problem?

## REDUCE THE RESOLUTION!





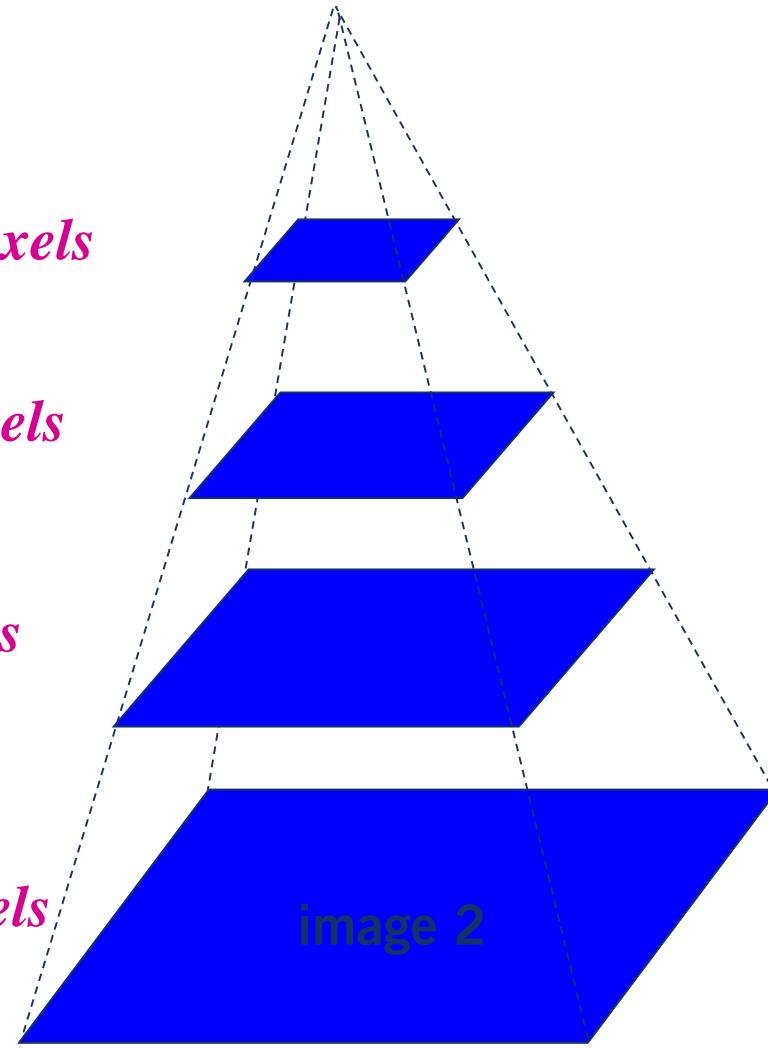
Gaussian pyramid of image 1

$u=1.25 \text{ pixels}$

$u=2.5 \text{ pixels}$

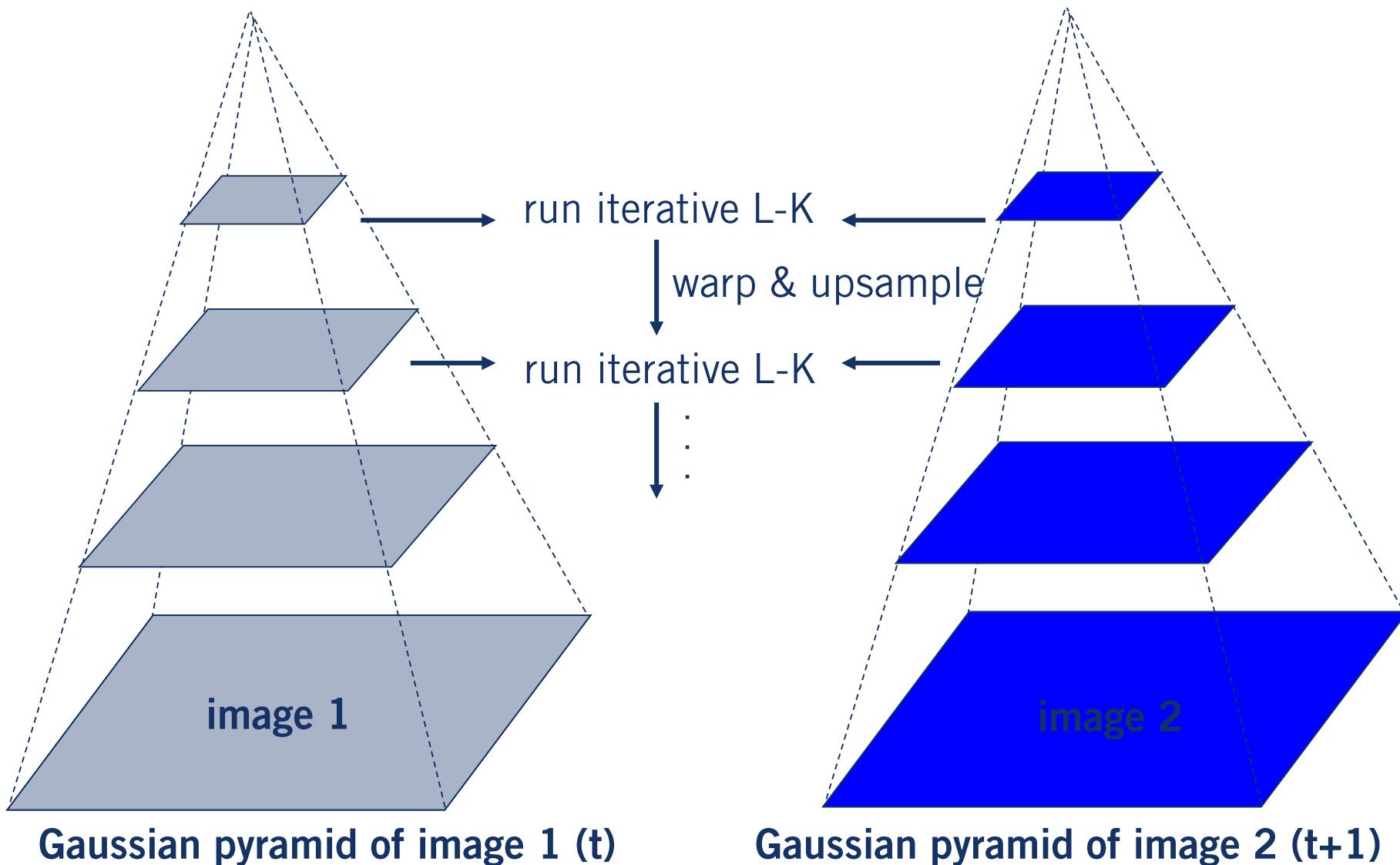
$u=5 \text{ pixels}$

$u=10 \text{ pixels}$

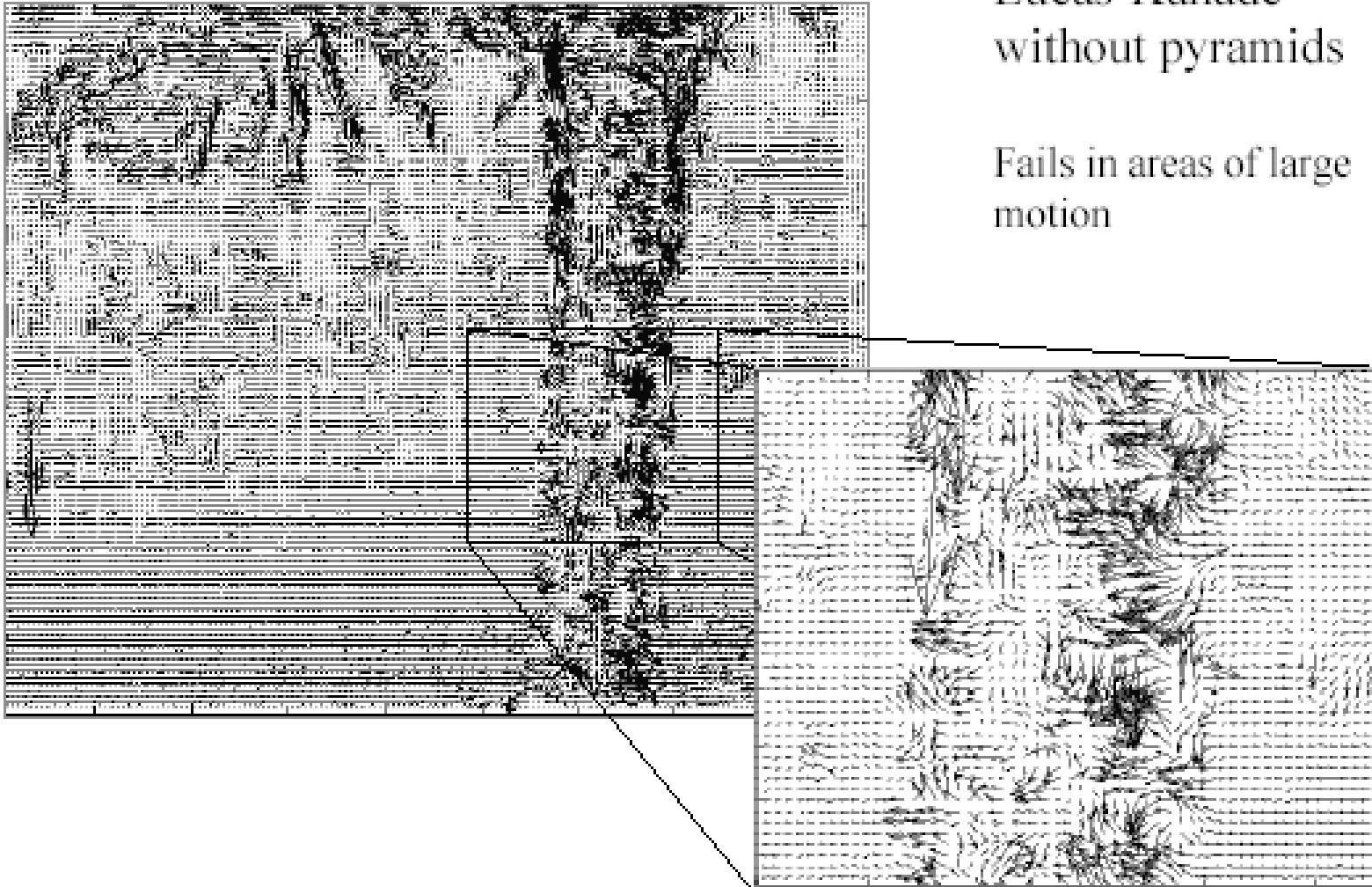


Gaussian pyramid of image 2

## COARSE-TO-FINE OPTICAL FLOW ESTIMATION



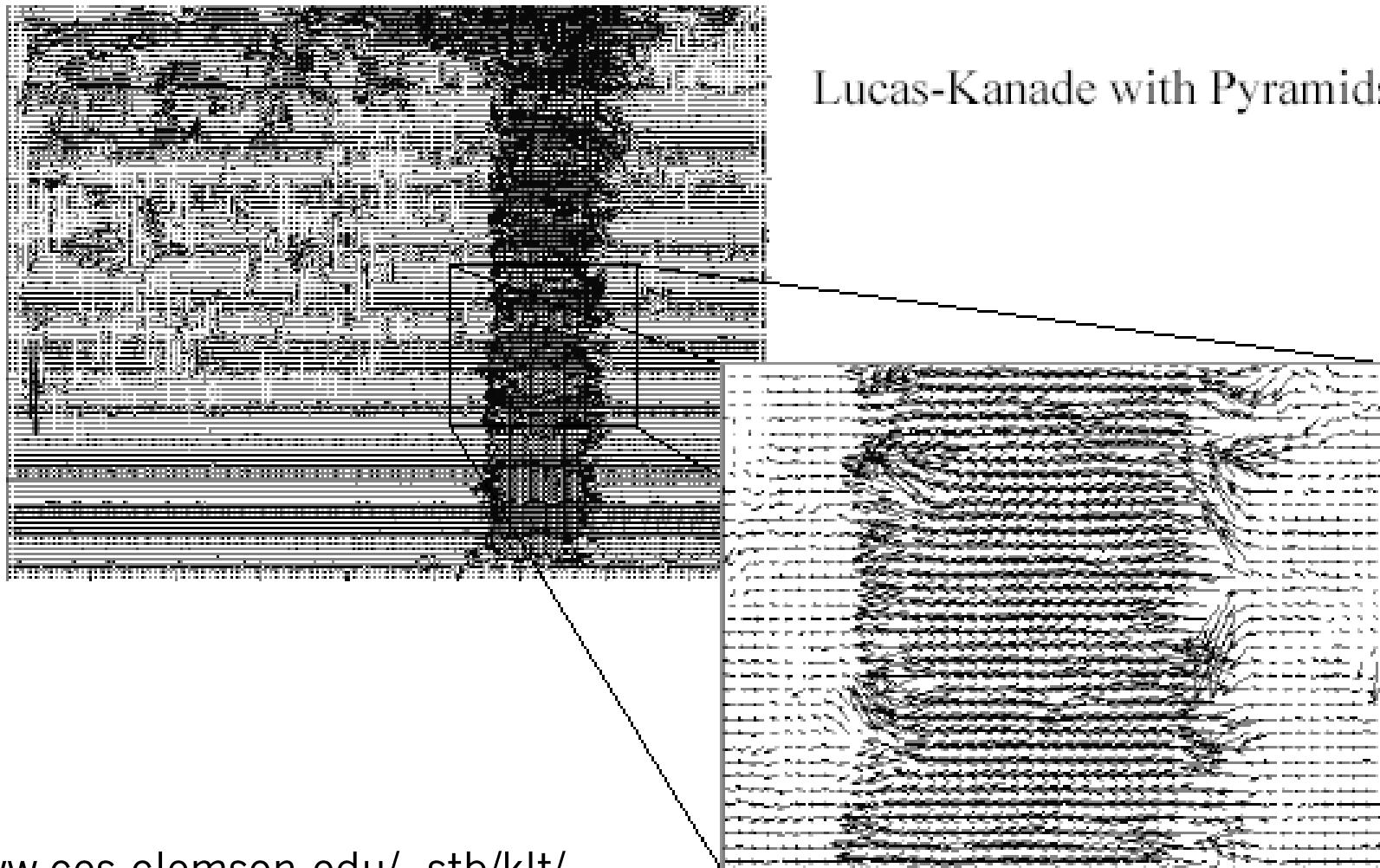
## OPTICAL FLOW RESULTS



Lucas-Kanade  
without pyramids

Fails in areas of large  
motion

## OPTICAL FLOW RESULTS



Lucas-Kanade with Pyramids

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

---

## WHAT WE WILL LEARN TODAY?

1. Optical flow
2. Lucas-Kanade method
3. Horn-Schunk method
4. Pyramids for large motion
- 5. Common fate**
6. Applications

## 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

## REMINDER: GESTALT – COMMON FATE



Common Fate

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

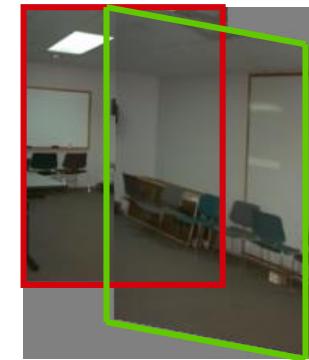


$$u(x, y) = a_1 + a_2x + a_3y$$

$$v(x, y) = a_4 + a_5x + a_6y$$

Substituting into the brightness constancy equation:

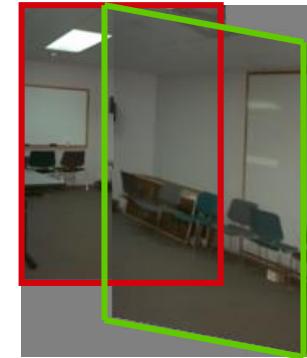
$$I_x \cdot u + I_y \cdot v + I_t \approx 0$$



$$u(x, y) = a_1 + a_2x + a_3y$$

$$v(x, y) = a_4 + a_5x + a_6y$$

Substituting into the brightness constancy equation:



$$I_x(a_1 + a_2x + a_3y) + I_y(a_4 + a_5x + a_6y) + I_t \approx 0$$

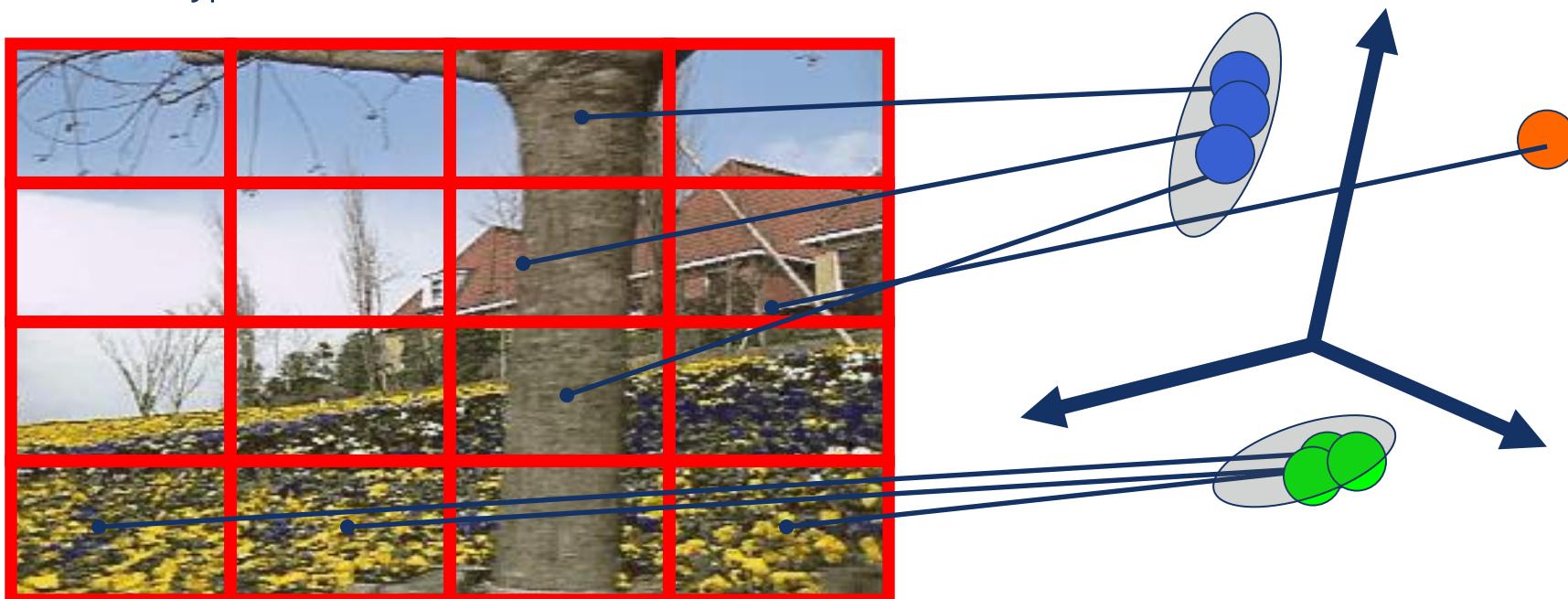
- Each pixel provides 1 linear constraint in 6 unknowns
- Least squares minimization:

$$Err(\vec{a}) = \sum [I_x(a_1 + a_2x + a_3y) + I_y(a_4 + a_5x + a_6y) + I_t]^2$$

## HOW DO WE ESTIMATE THE LAYERS?

### 1. Obtain a set of initial affine motion hypotheses

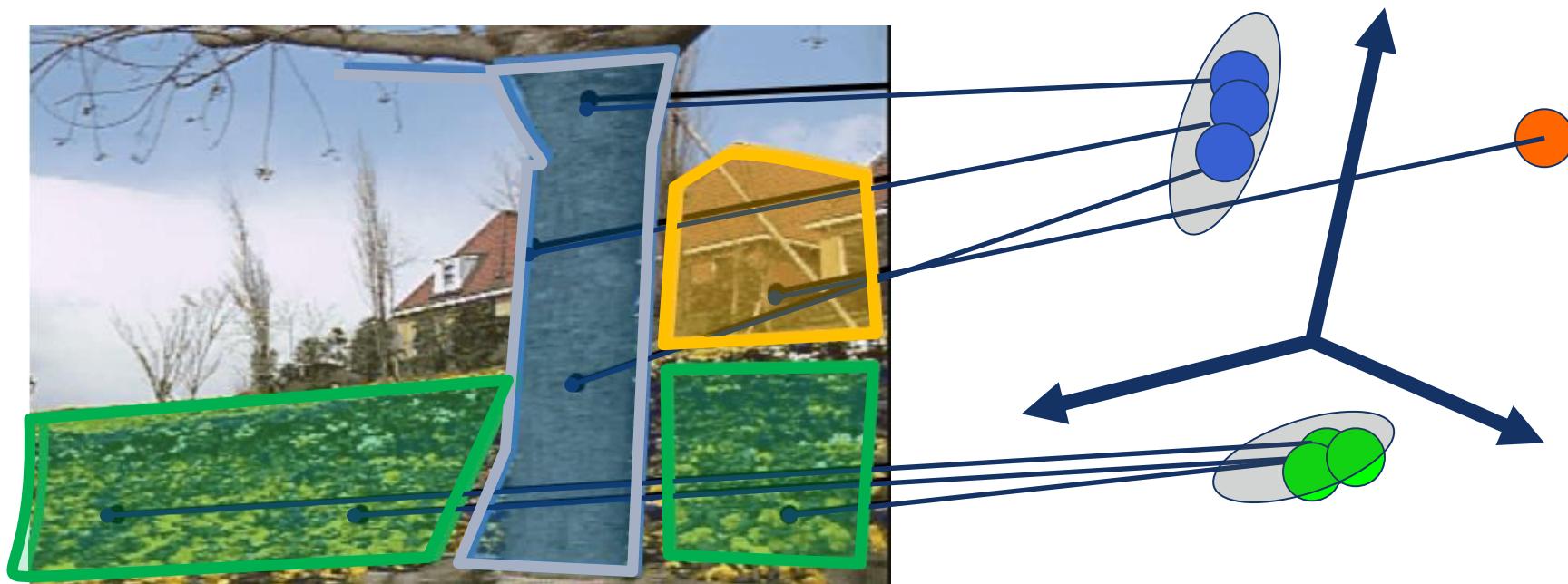
- Divide the image into blocks and estimate affine motion parameters in each block by least squares  
Eliminate hypotheses with high residual error
- Map into motion parameter space
- Perform k-means clustering on affine motion parameters  
Merge clusters that are close and retain the largest clusters 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



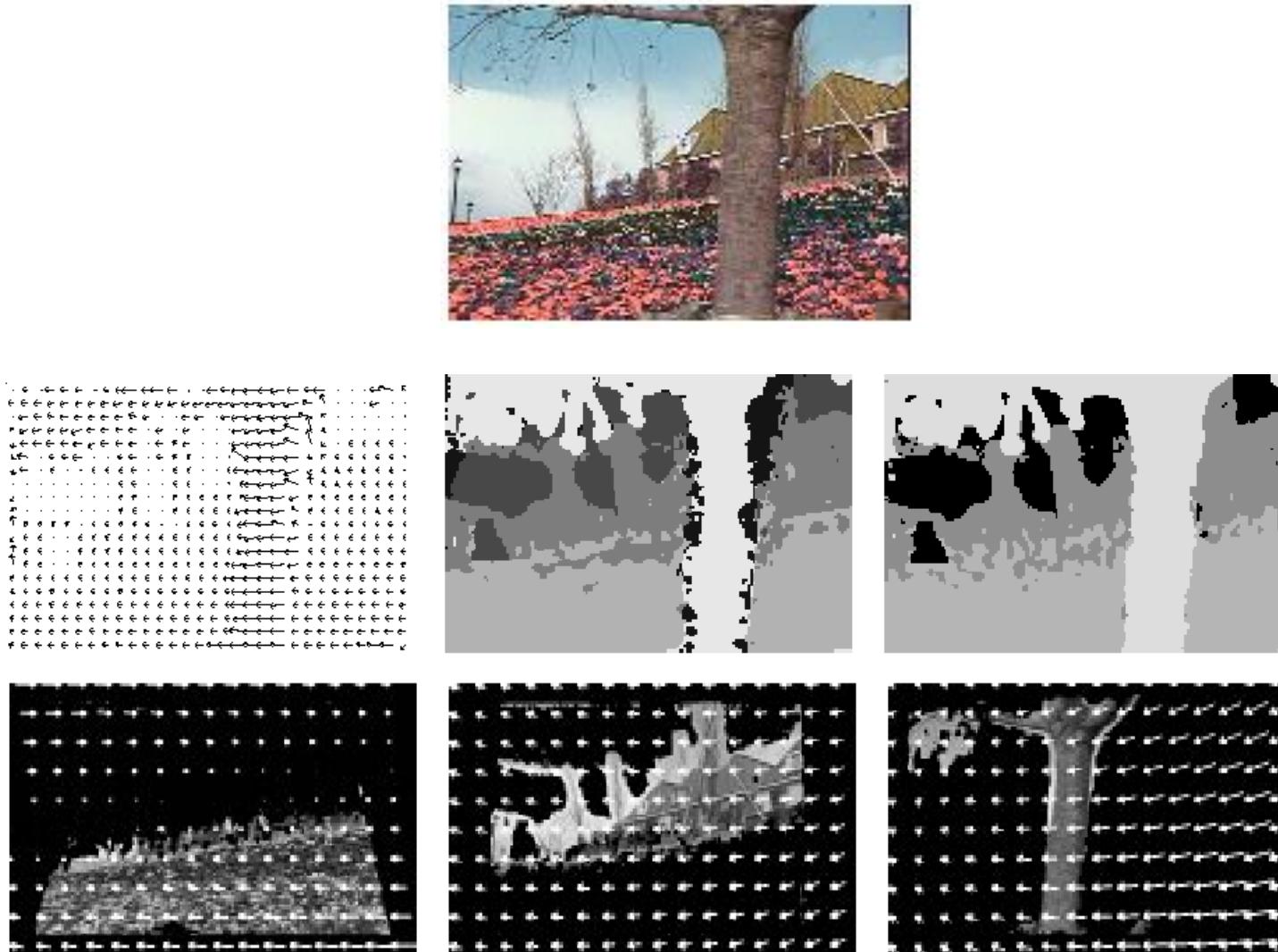
### 1. Obtain a set of initial affine motion hypotheses

- 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

## EXAMPLE RESULT



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

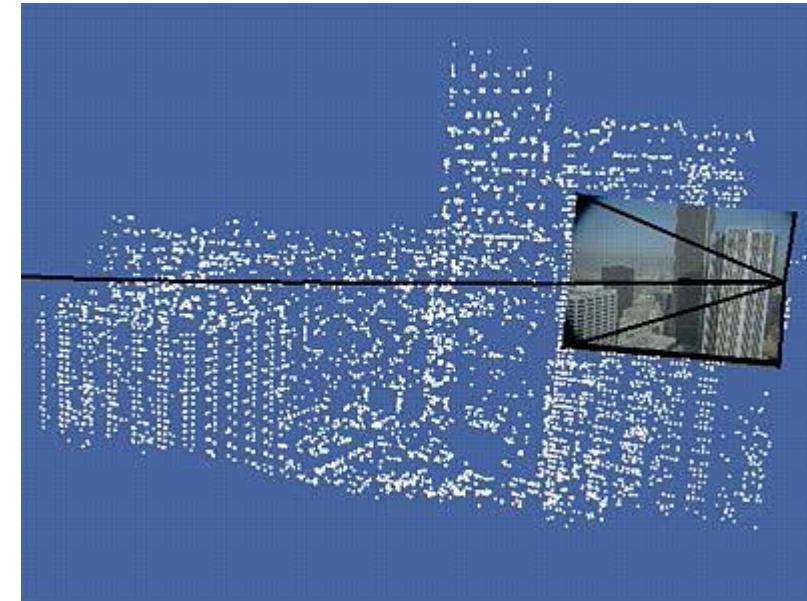
---

## WHAT WE WILL LEARN TODAY?

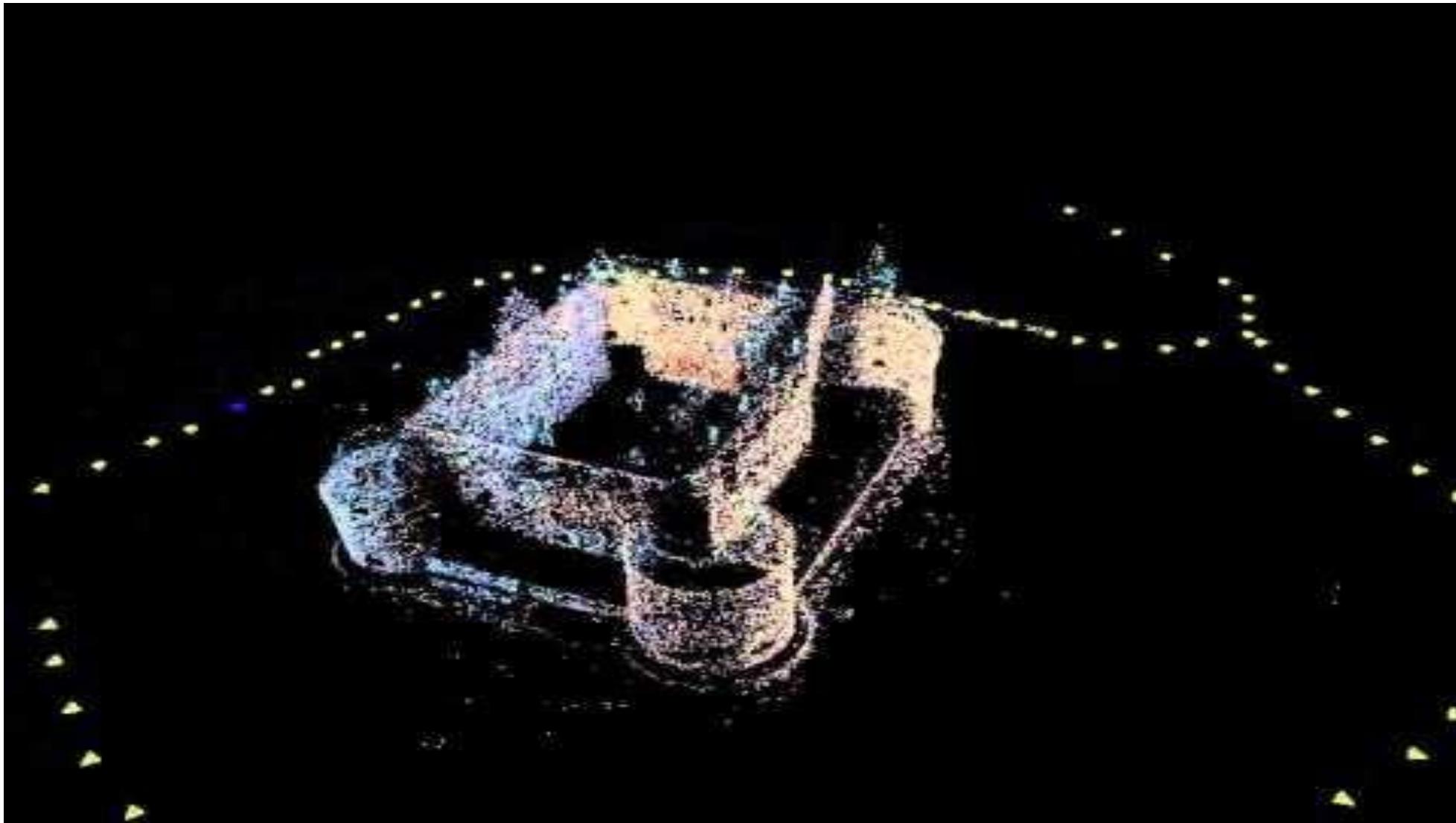
1. Optical flow
2. Lucas-Kanade method
3. Horn-Schunk method
4. Pyramids for large motion
5. Common fate
6. Applications

- 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



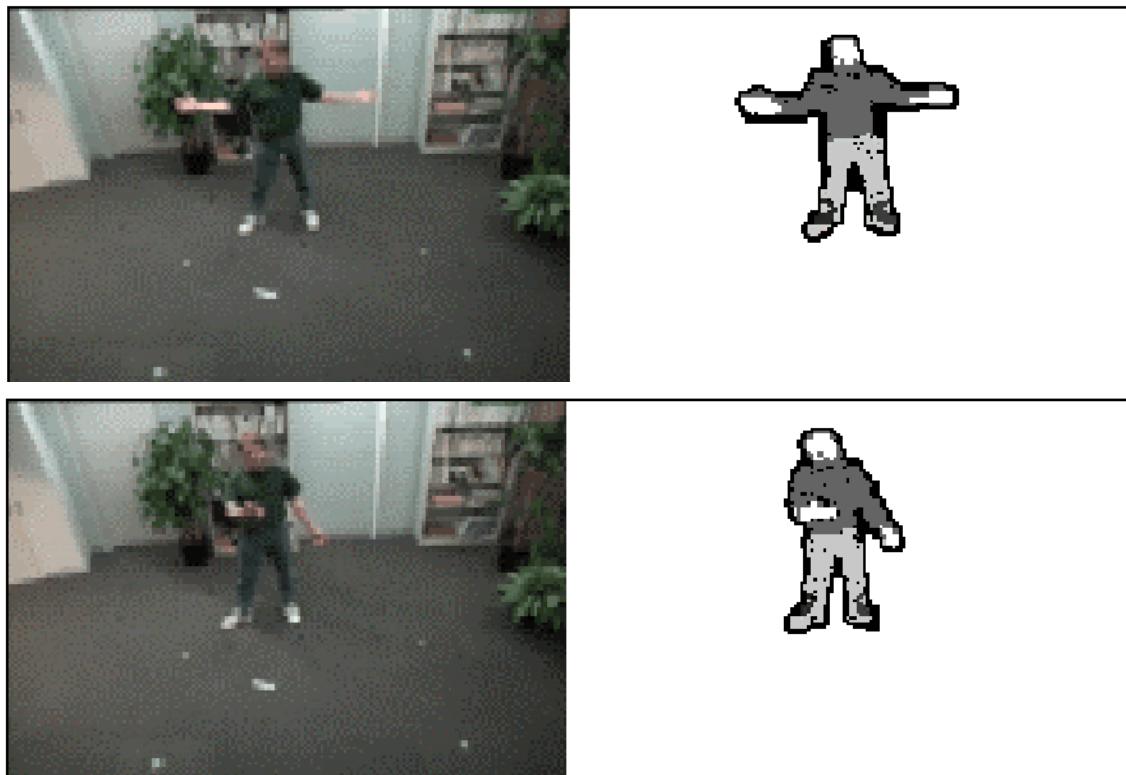
## ESTIMATING 3D STRUCTURE



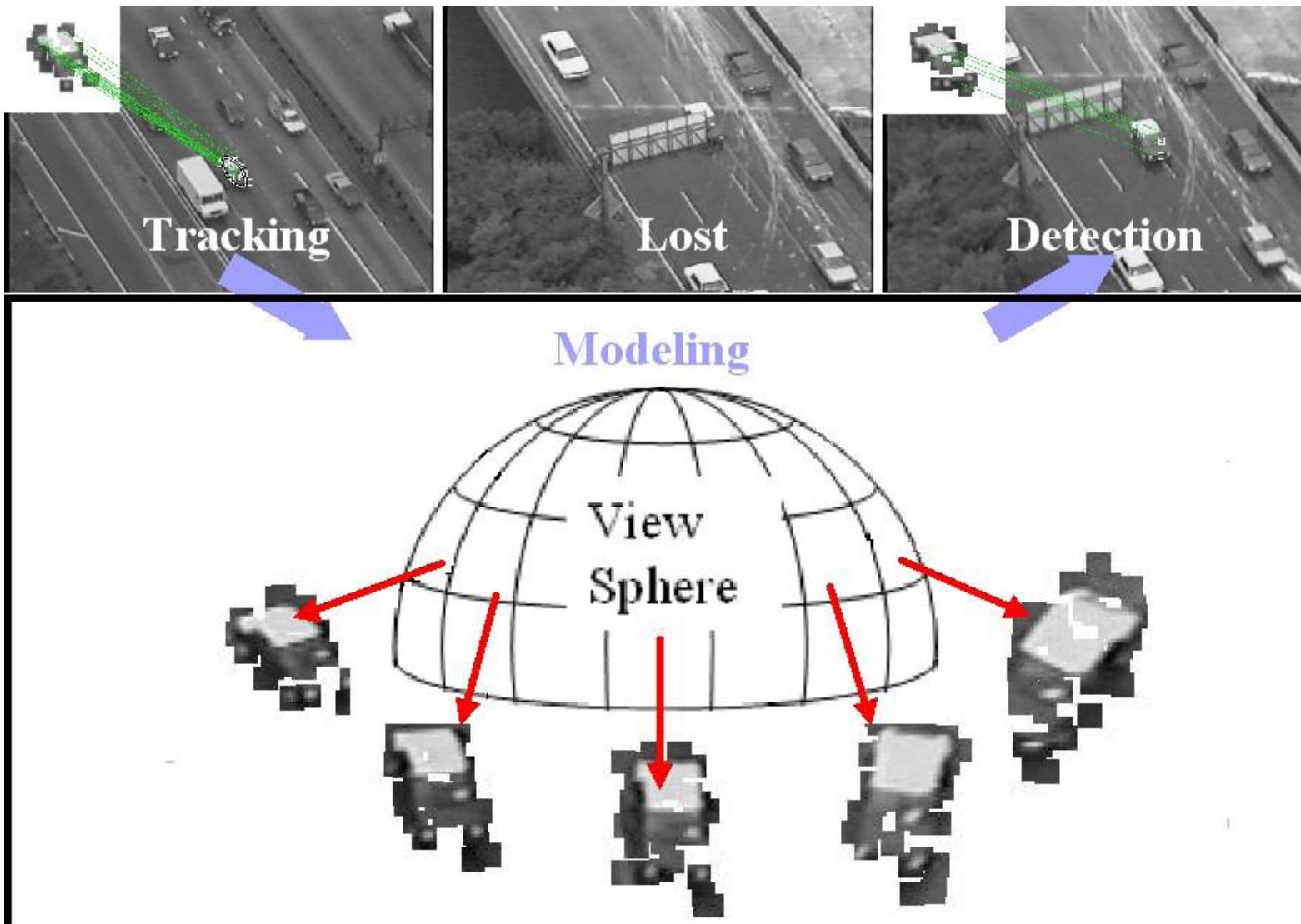
## SEGMENTING OBJECTS BASED ON MOTION CUES

### Background subtraction

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



## 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.

## SYNTHESIZING DYNAMIC TEXTURES



Copyright (c) UCLA, G. Doretto and S. Soatto, 2002

Original      Synthesized

## Example: A set of low quality images

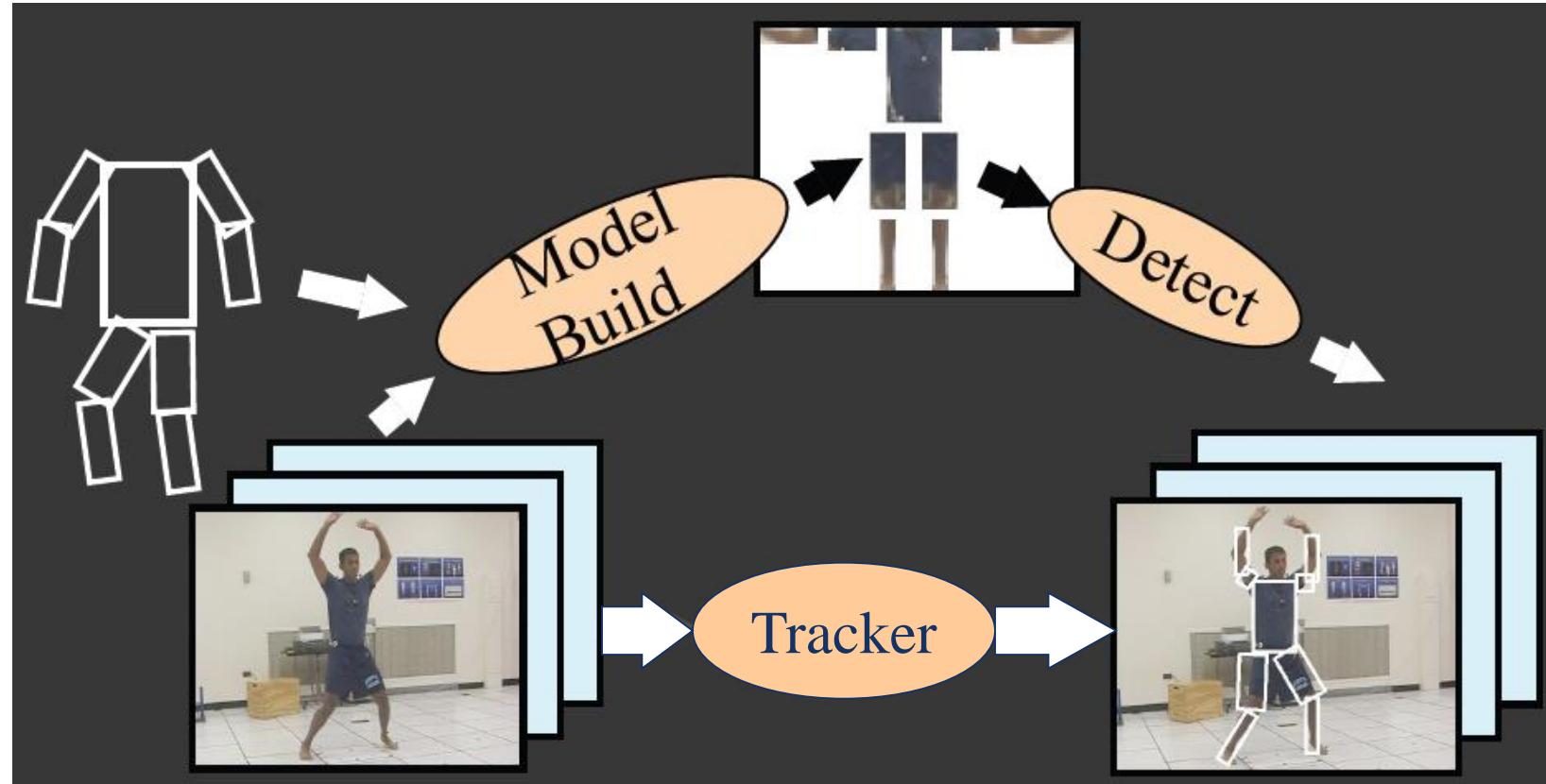
Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S	Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S	Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S
Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S	Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S	Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S
Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S	Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S	Most of the test data o couple of exceptions. 7 low-temperature solders investigated (or some manufacturing technolo nonwetting of 40In40S) microstructural coarse mal cycling of 58Bi42S

Each of these images looks like this:

Most of the test data of a couple of exceptions. The low-temperature solder investigated (or some other manufacturing technology) was wetting of 400°C-105°C microstructural coarse and cycling of 580°C-425°C.

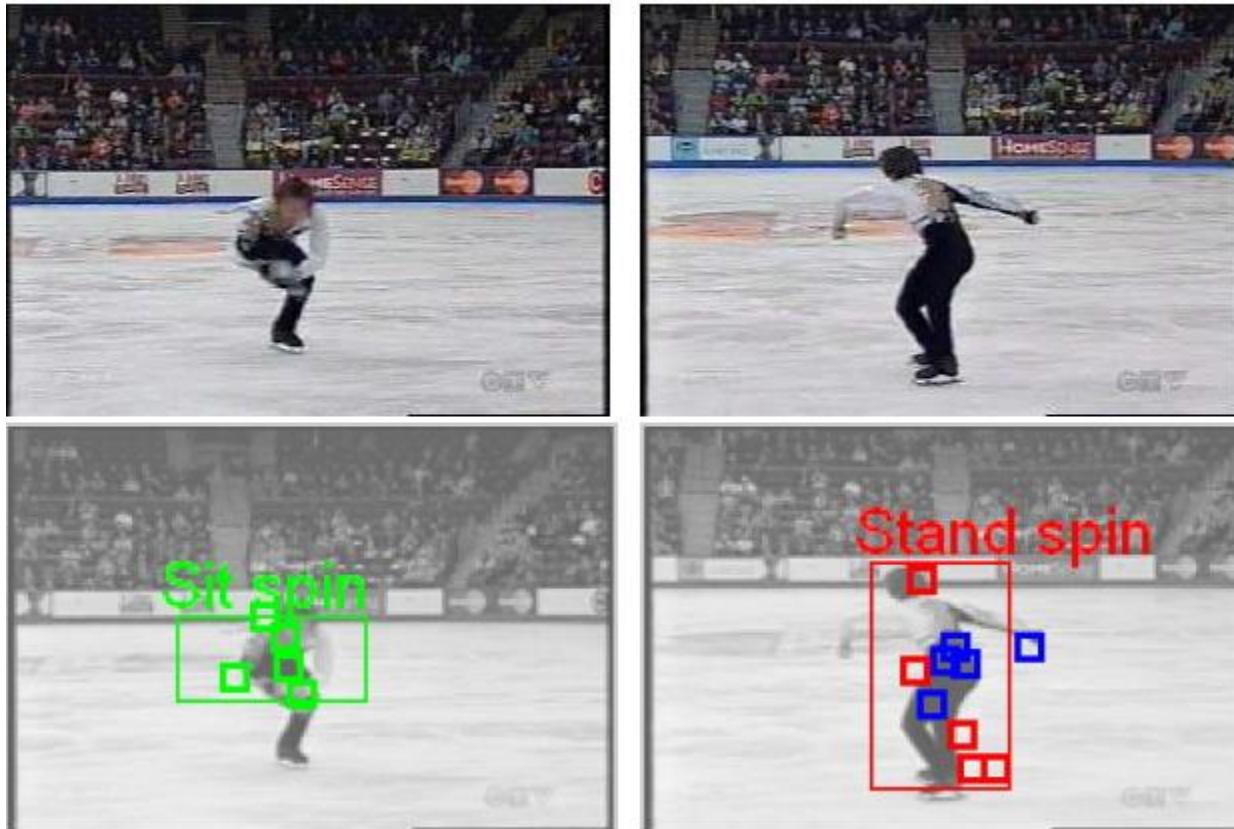
The recovery result:

Most of the test data on the other hand show a couple of exceptions. These are mainly related to low-temperature solder joints which were not investigated (or some data was lost) due to manufacturing technology. The first example is the nonwetting of 40In40Sn at 150°C. This leads to a microstructural coarsening of the solder joint after thermal cycling of 58Bi42Sb at 150°C.



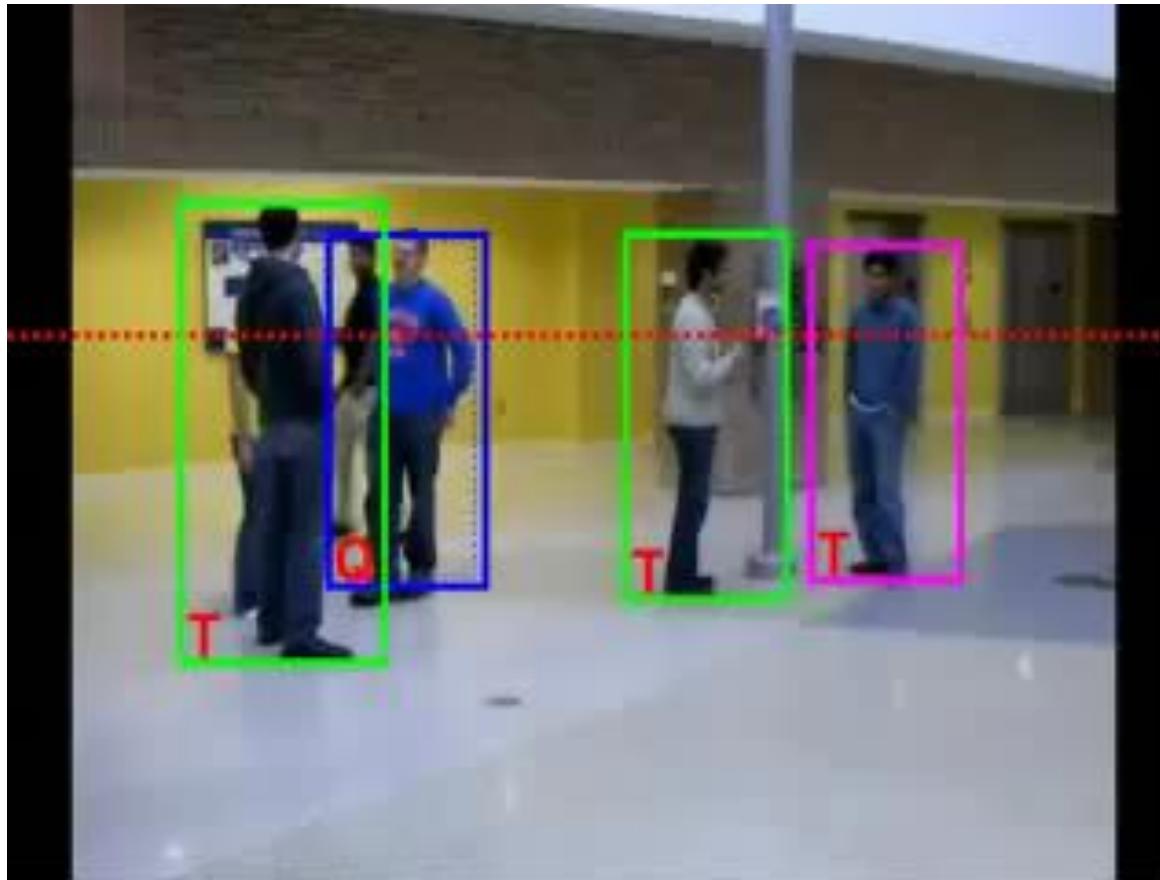
D. Ramanan, D. Forsyth, and A. Zisserman. Tracking People by Learning their Appearance. PAMI 2007.

## 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.

# Crossing – Talking – Queuing – Dancing – jogging



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

## WHAT WE HAVE LEARNED TODAY?

1. Optical flow
2. Lucas-Kanade method
3. Horn-Schunck method
4. Pyramids for large motion
5. Common fate
6. Applications

**Reading:** [Szeliski] Chapters: 8.4, 8.5

[Fleet & Weiss, 2005]

<http://www.cs.toronto.edu/pub/jepson/teaching/vision/2503/opticalFlow.pdf>