Vision and Perception

Motion estimation and Optical flow

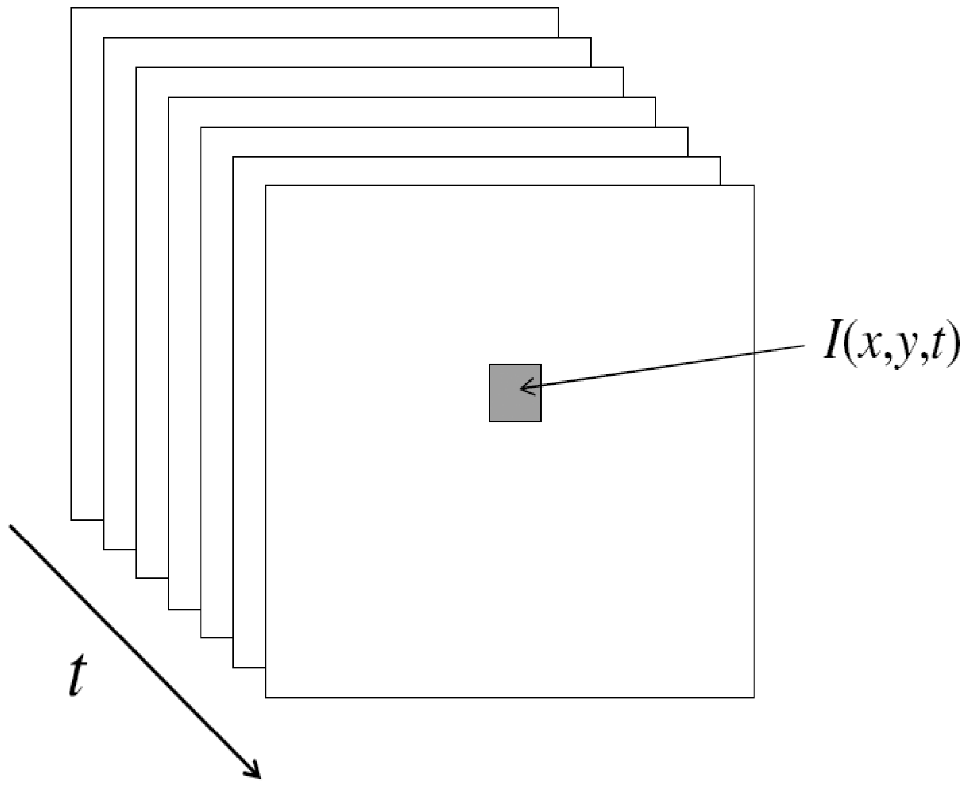
SAPIENZA
UNIVERSITÀ DI ROMA

References

- Basic reading: Szeliski, Chapter 9.3-9.4

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)



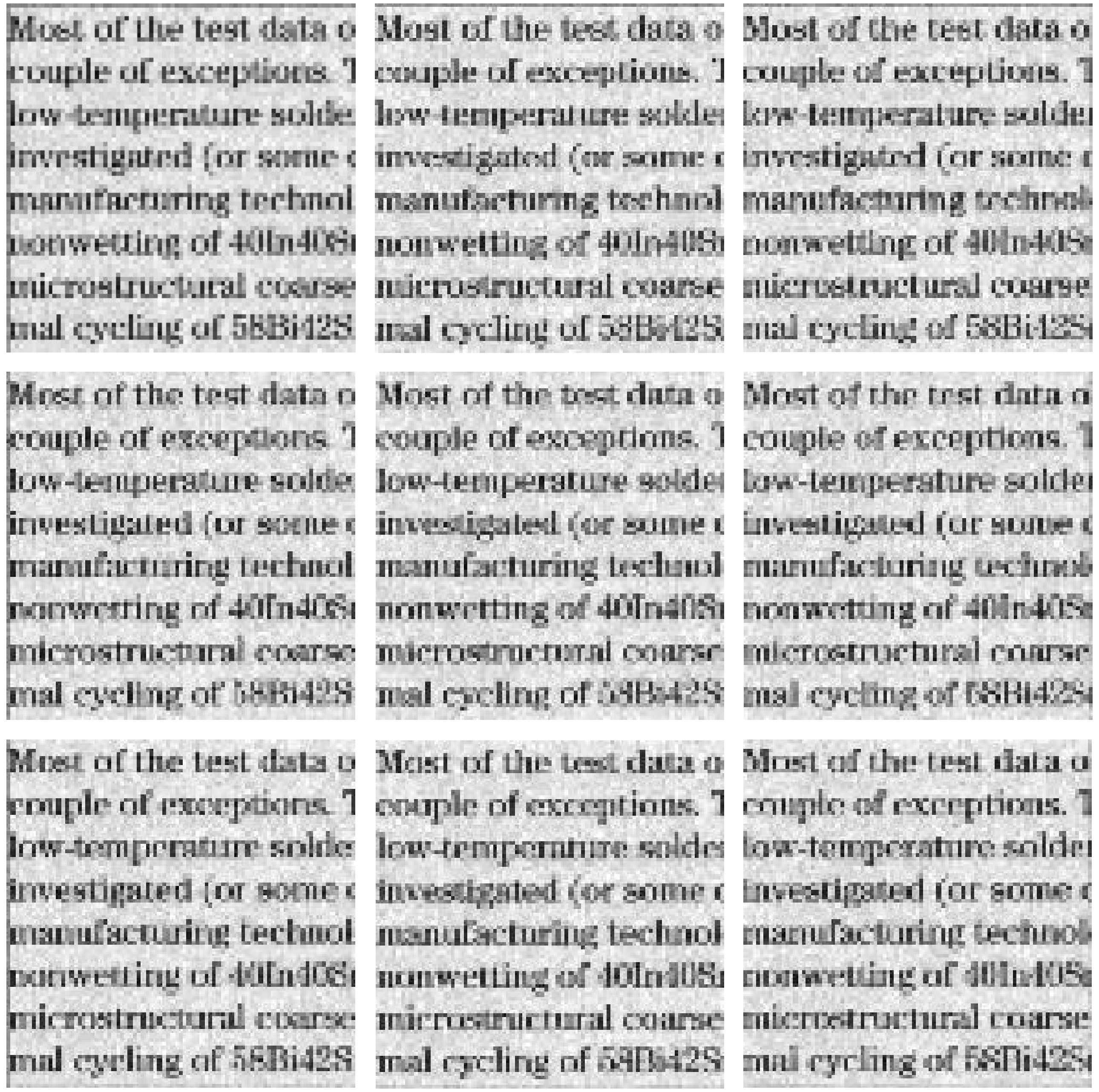
Uses of motion

The estimation of every pixel in a sequence is a problem with many applications in computer vision

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

Super-resolution

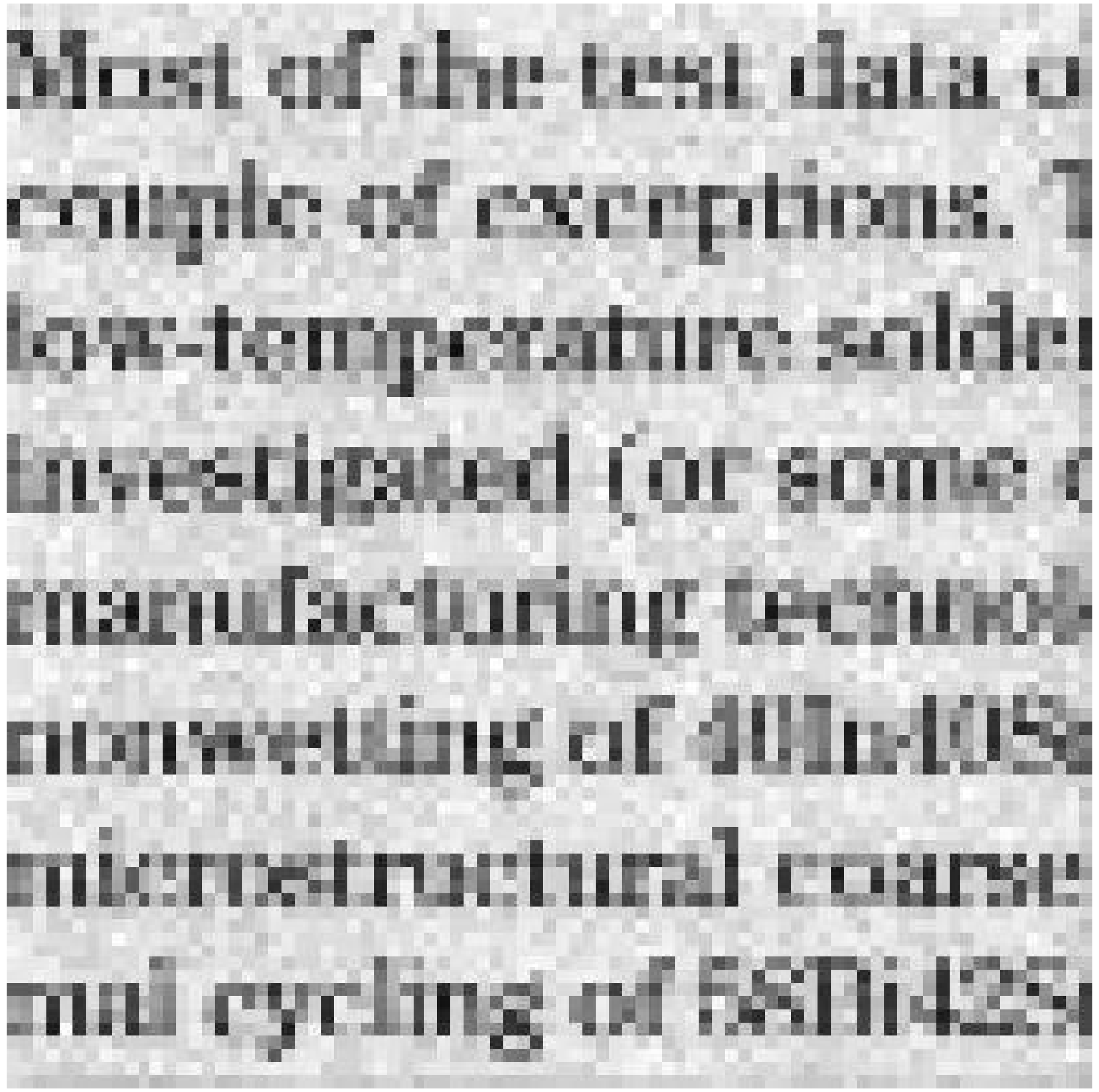
Example: a set of low quality images



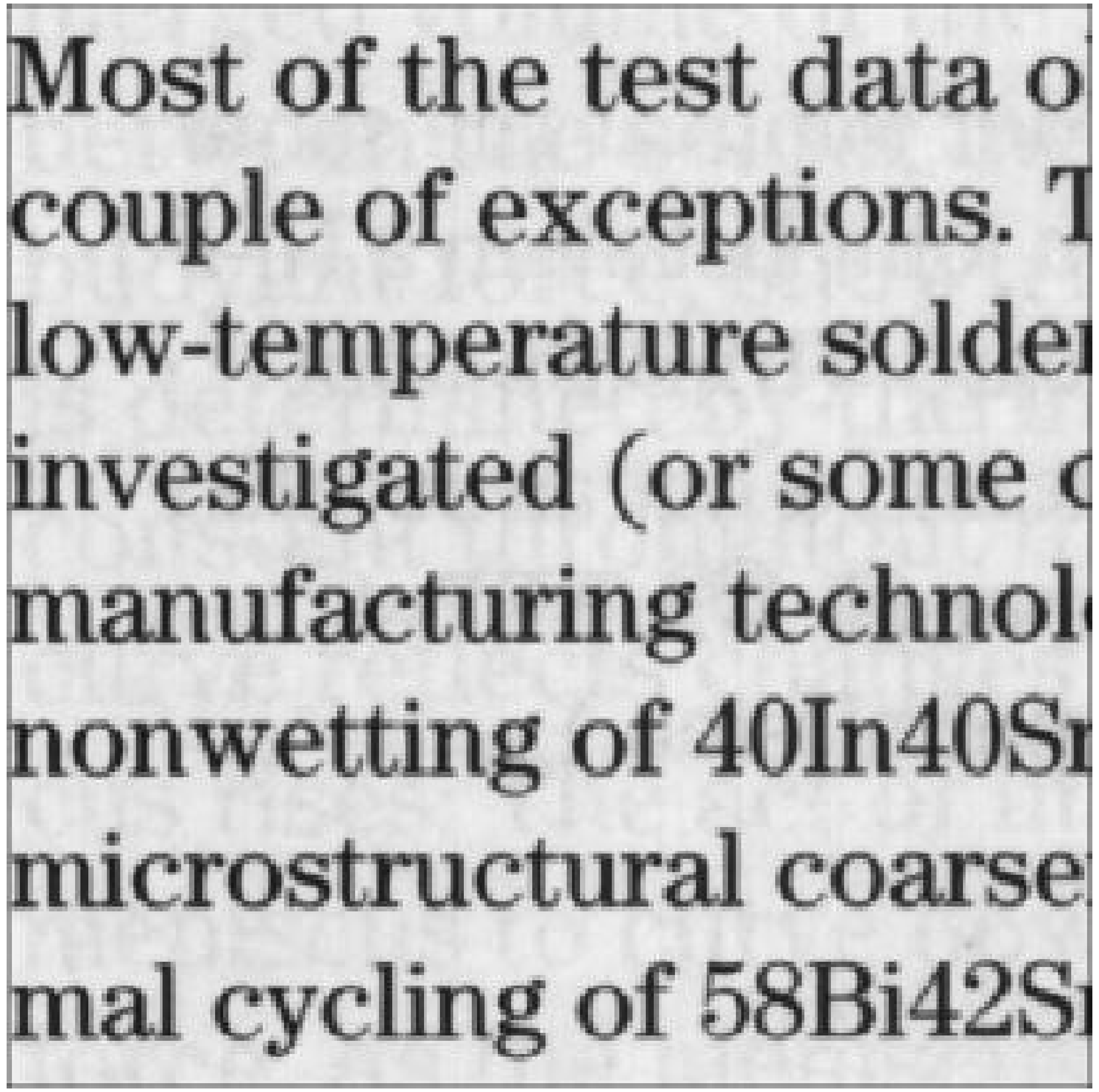
- Irani, M.; Peleg, S. (June 1990). "Super Resolution From Image Sequences". International Conference on Pattern Recognition
- Fast and Robust Multiframe Super Resolution, Sina Farsiu, M. Dirk Robinson, Michael Elad, and Peyman Milanfar, EEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 10, OCTOBER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

Super-resolution

Each of these images looks like this:

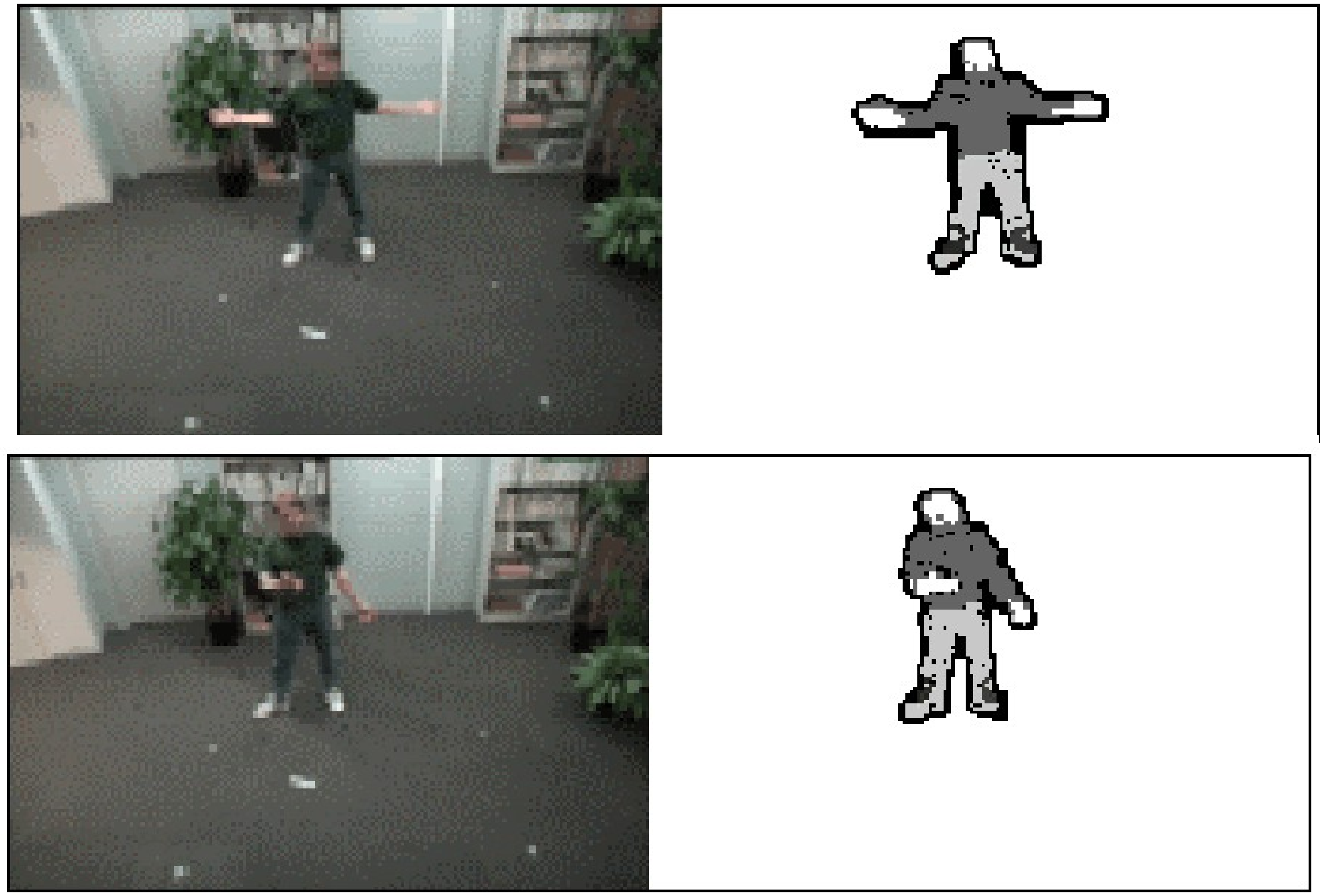


The recovery result:



Segmenting objects based on motion cues

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

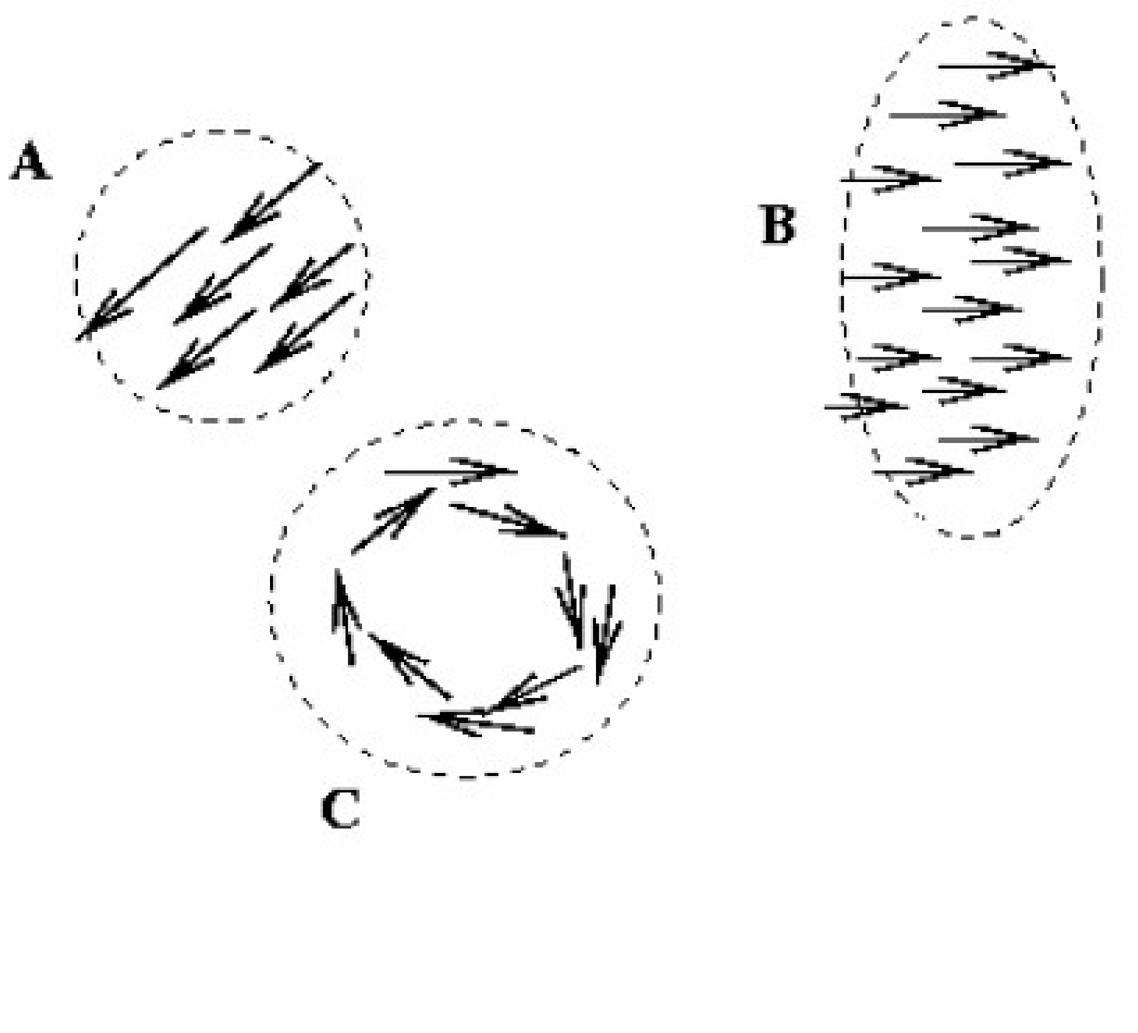


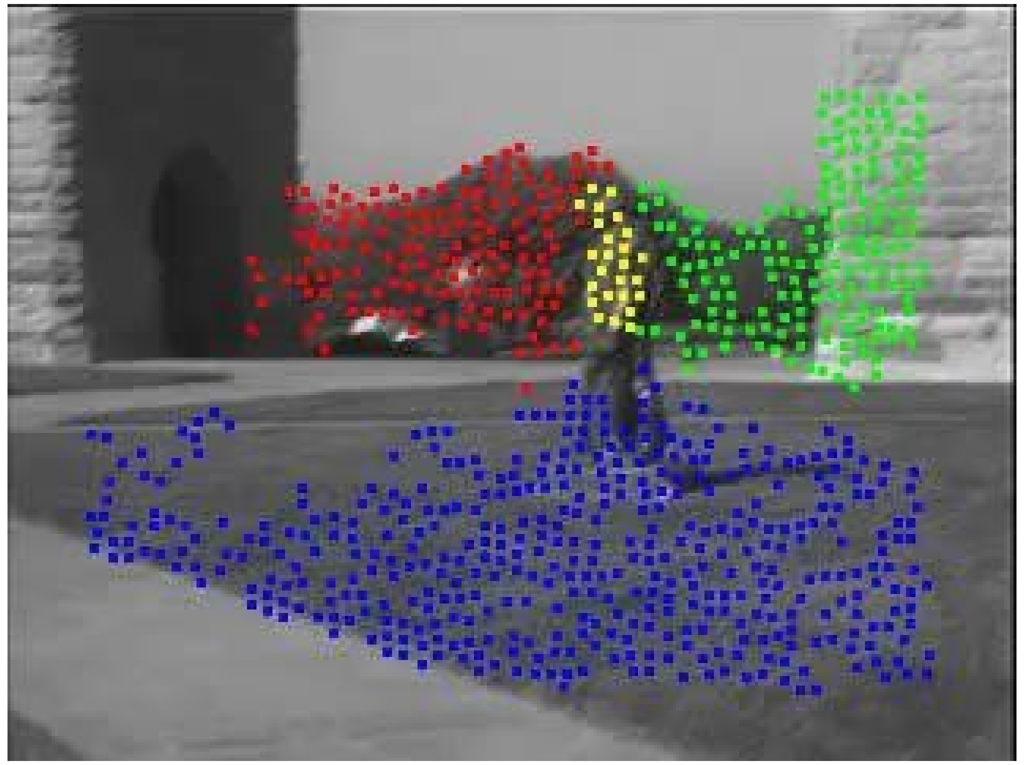
https://www.youtube.com/watch?v=YAszeOaInUM

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

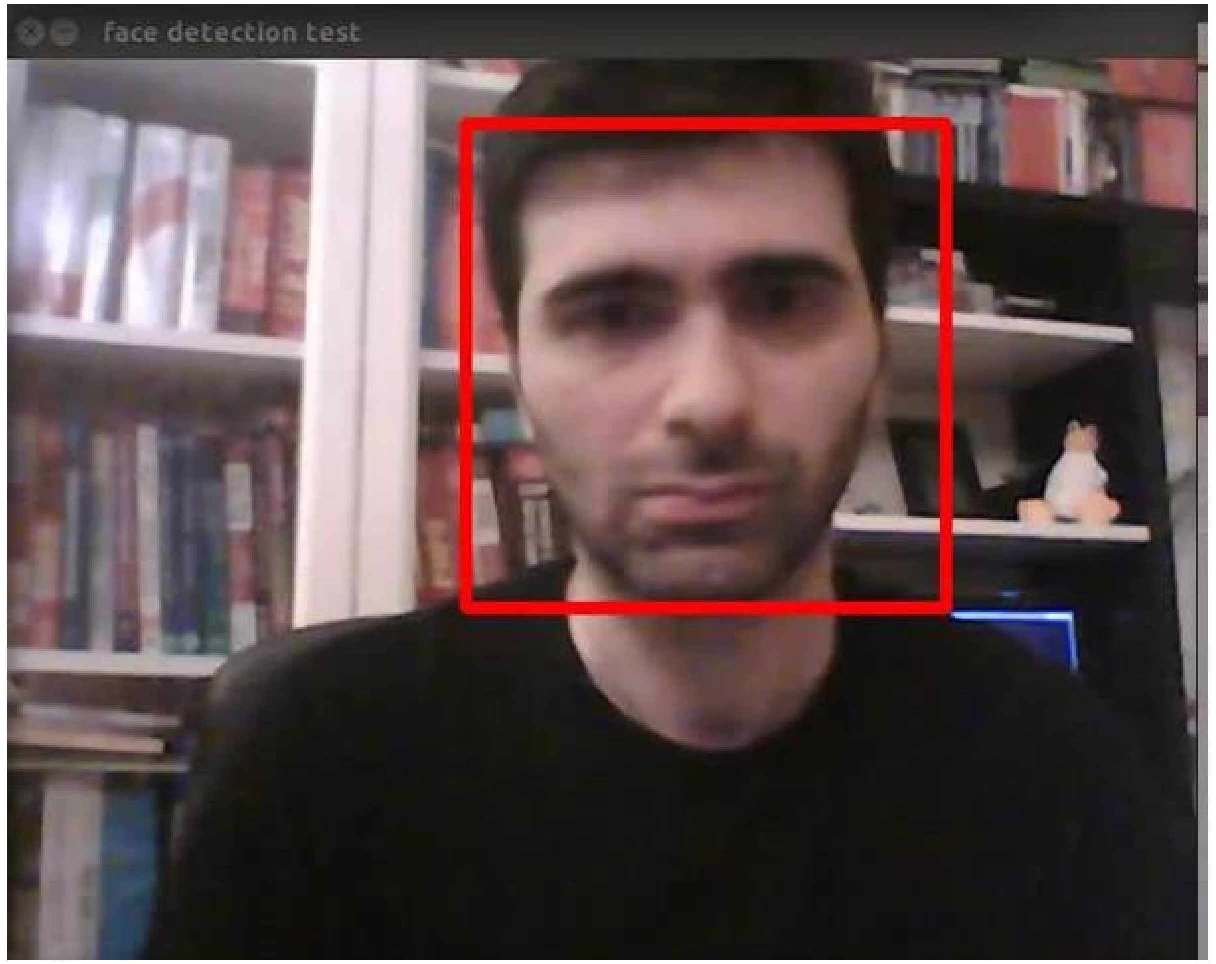




S. J. Pundlik and S. T. Birchfield, Motion Segmentation at Any Speed, Proceedings of the British Machine Vision Conference 2006

Tracking objects

- Facing tracking on openCV

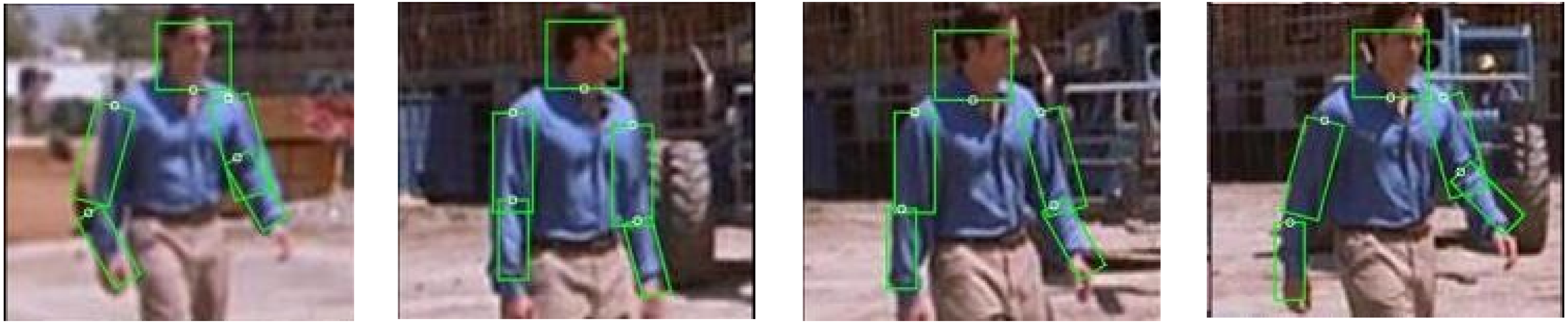


OpenCV's face tracker uses an algorithm called Camshift (based on the meanshift algorithm)

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

http://learnopencv.com/wp-content/uploads/2017/02/real-time-face-tracking.gif

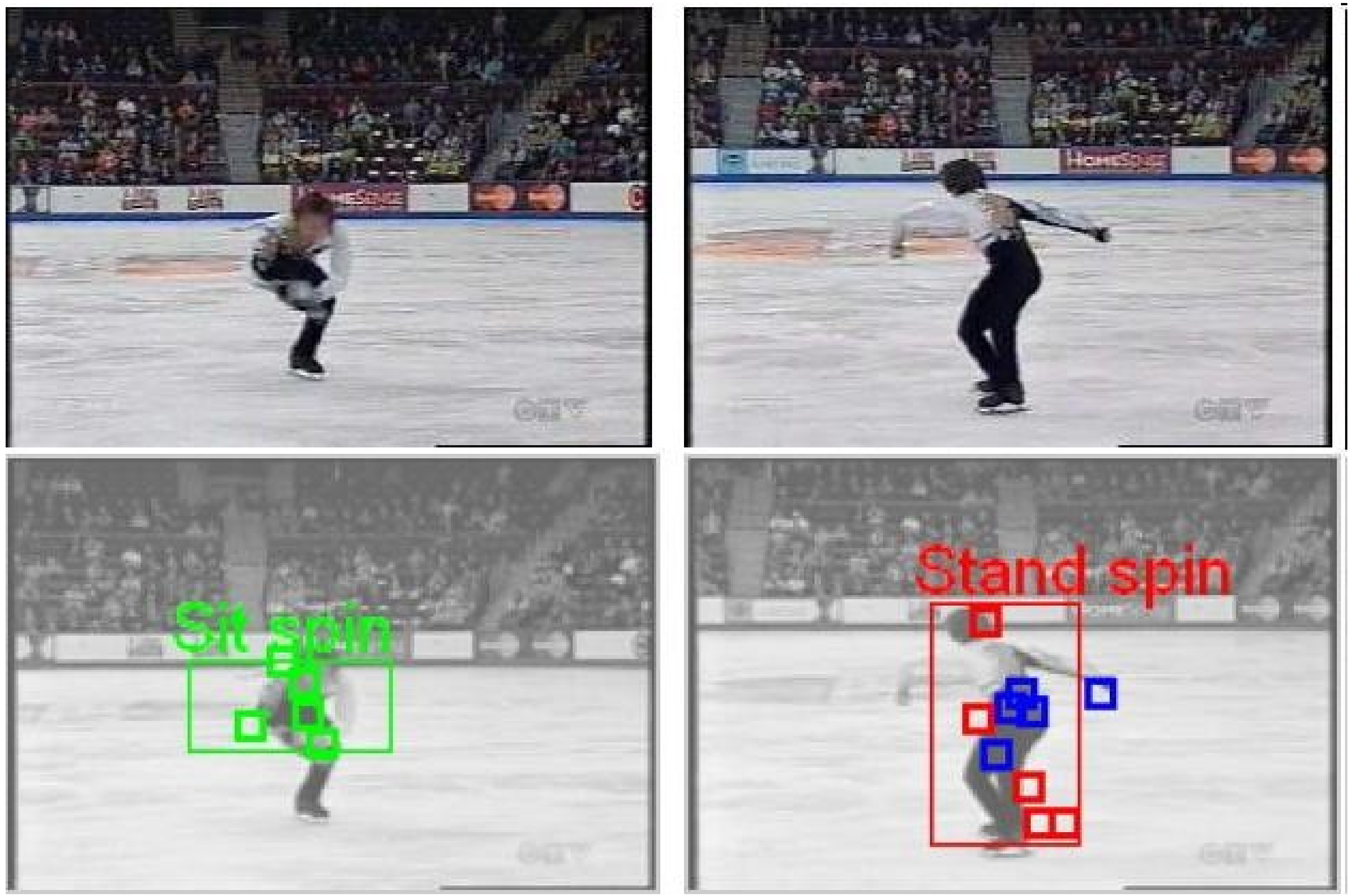
Tracking body parts



Courtesy of Benjamin Sapp

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

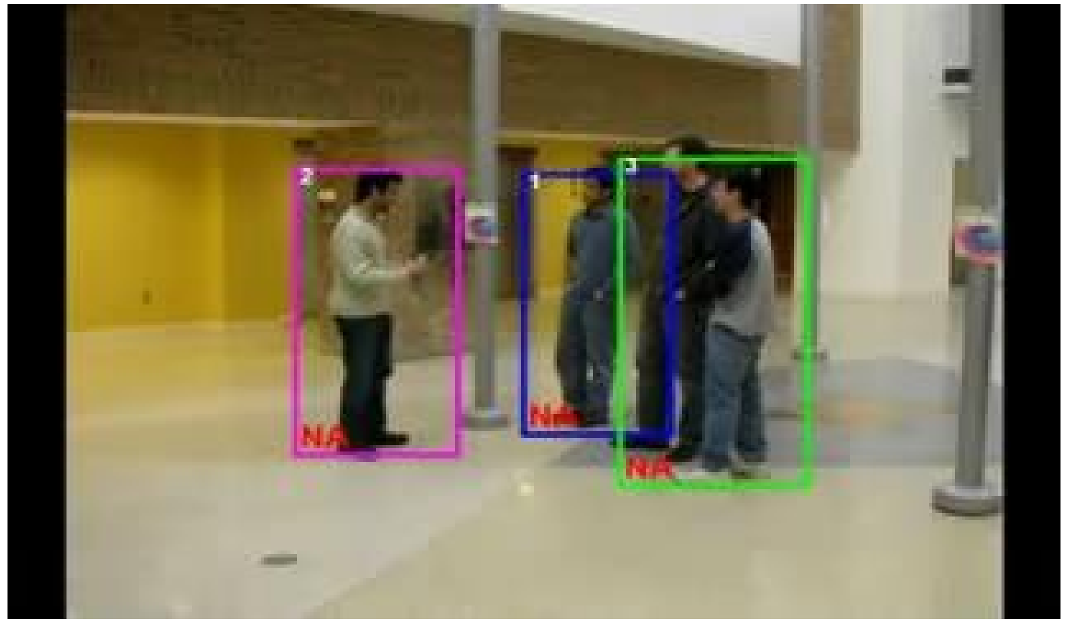
Recognizing events and activities

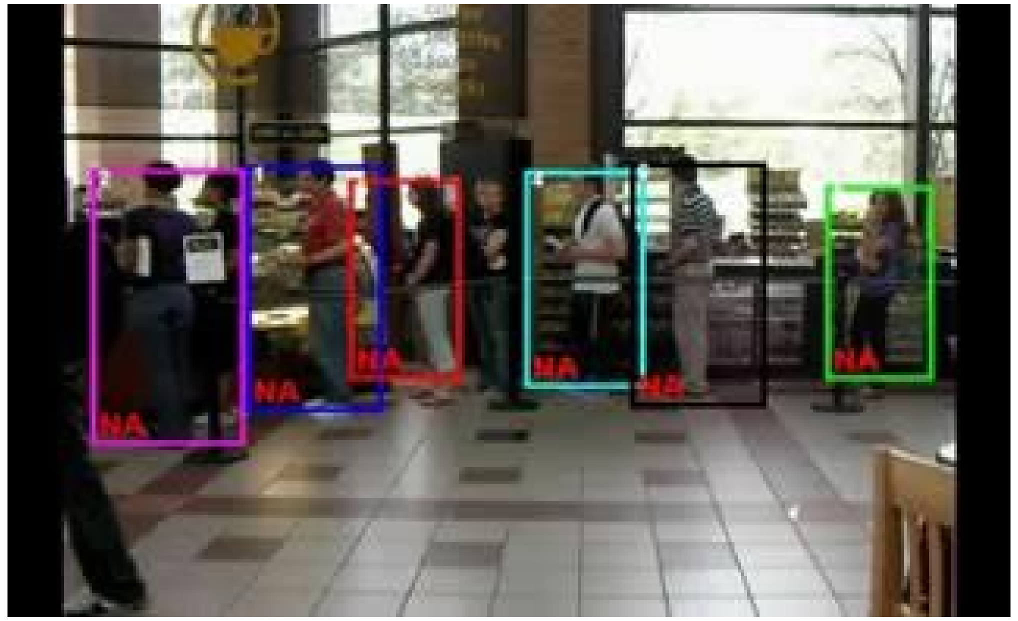


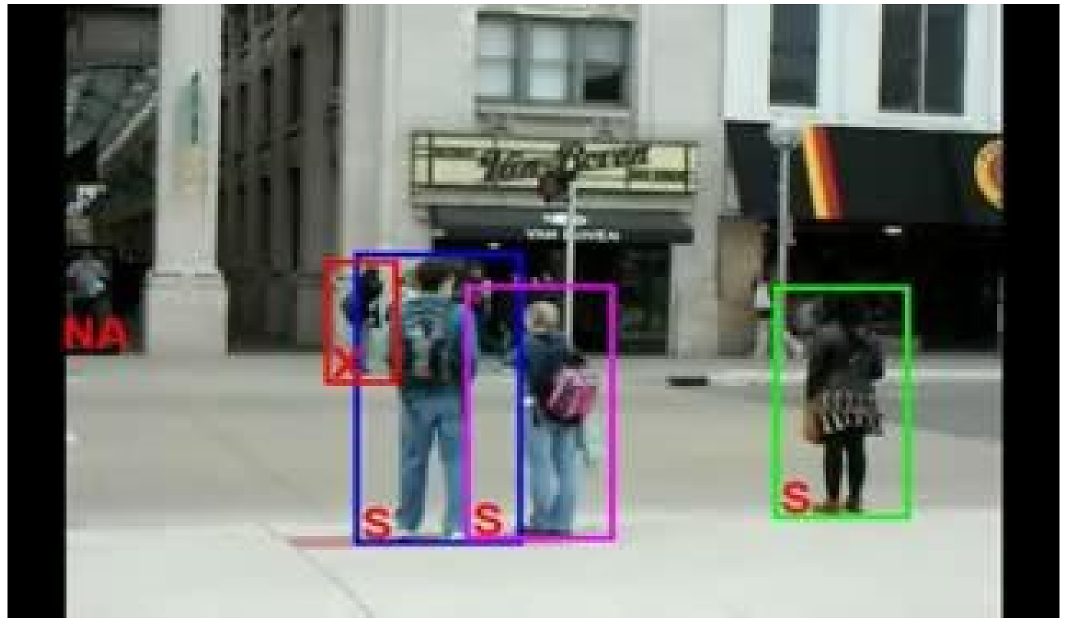
Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei, Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words, *(BMVC)*, Edinburgh, 2006.

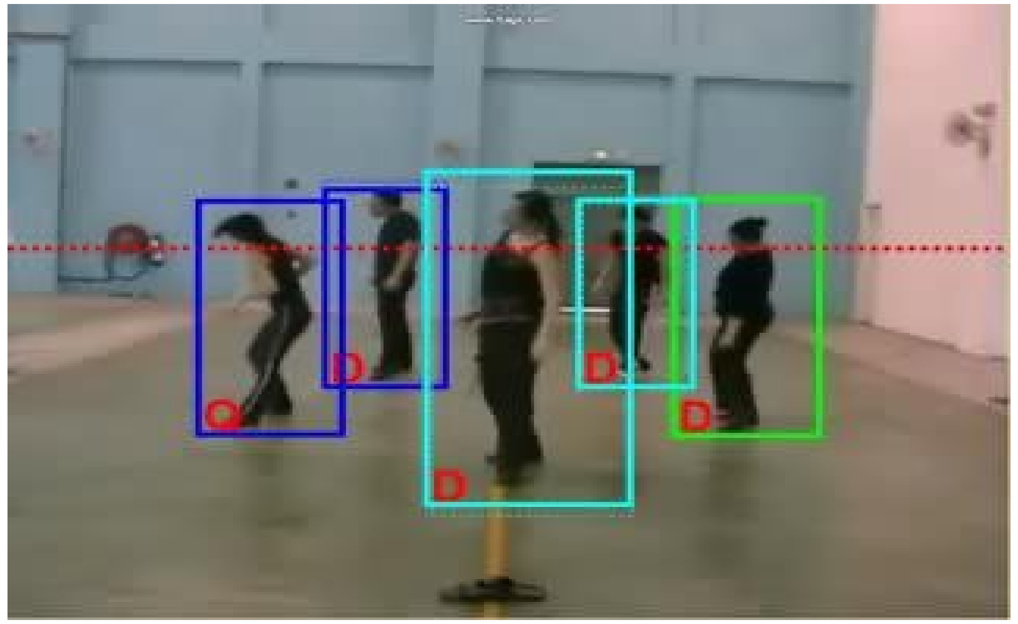
Recognizing group activities

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

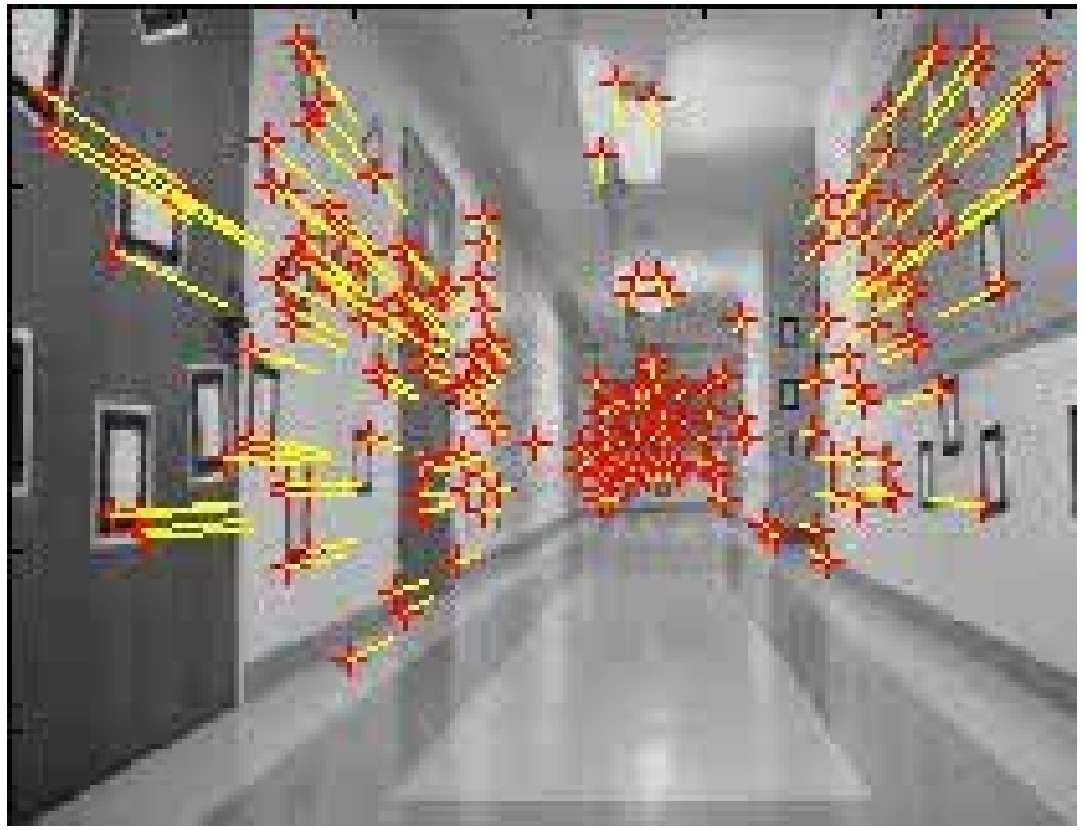
- Recover image motion at each pixel from spatio-temporal image brightness variations

Feature-tracking

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

Tracking features

Tracking object regions frame to frame

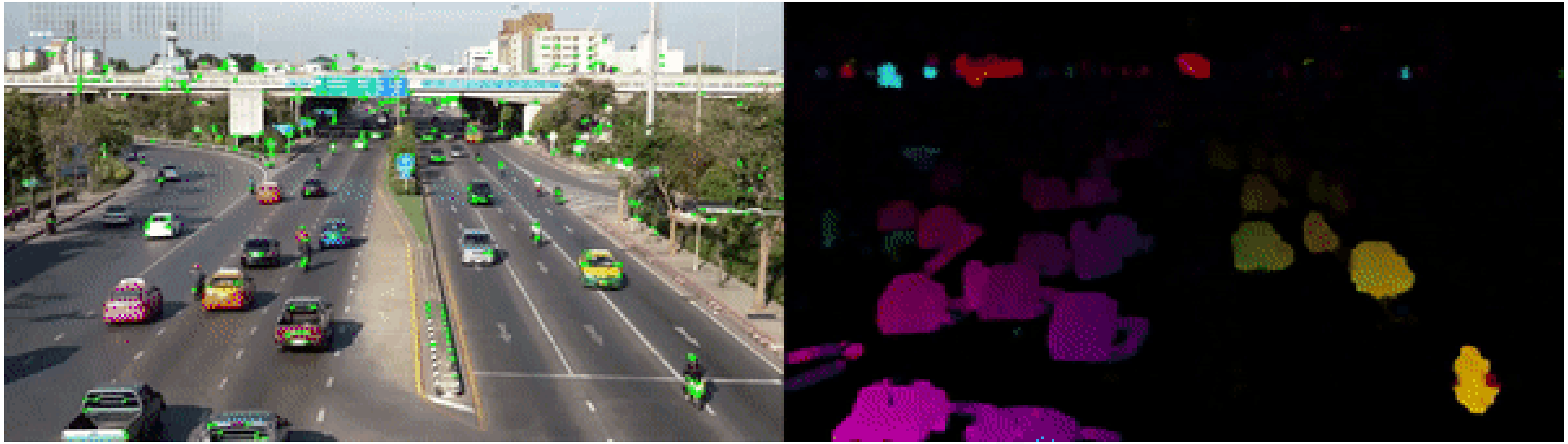



Courtesy of Jean-Yves Bouguet – Vision Lab, California Institute of Technology

Steve Gu, Ying Zheng, and Carlo Tomasi. 2010. Efficient visual object tracking with online nearest neighbor classifier. In Proceedings of the 10th Asian conference on Computer vision - Volume Part I (ACCV'10). Springer-Verlag, Berlin, Heidelberg, 271–282.

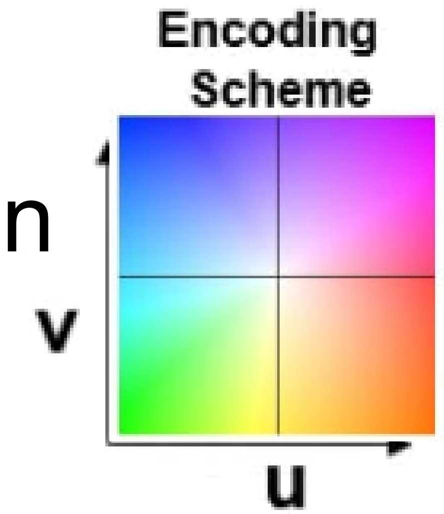
Optical flow

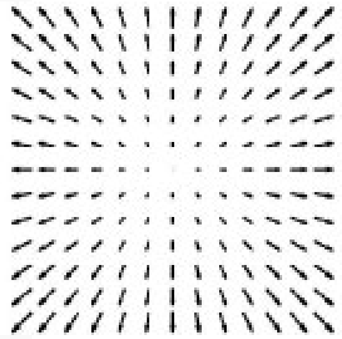
Optical flow is used to see how every point is moving frame to frame in a video sequence



Colour code for visualisation

HSV encoding scheme: hue, saturation, value
Vector field: displacement, velocity





Optical flow

1. **Sparse Optical Flow**: this method processes the flow vectors of only a few of the most interesting pixels from the entire image, within a frame.

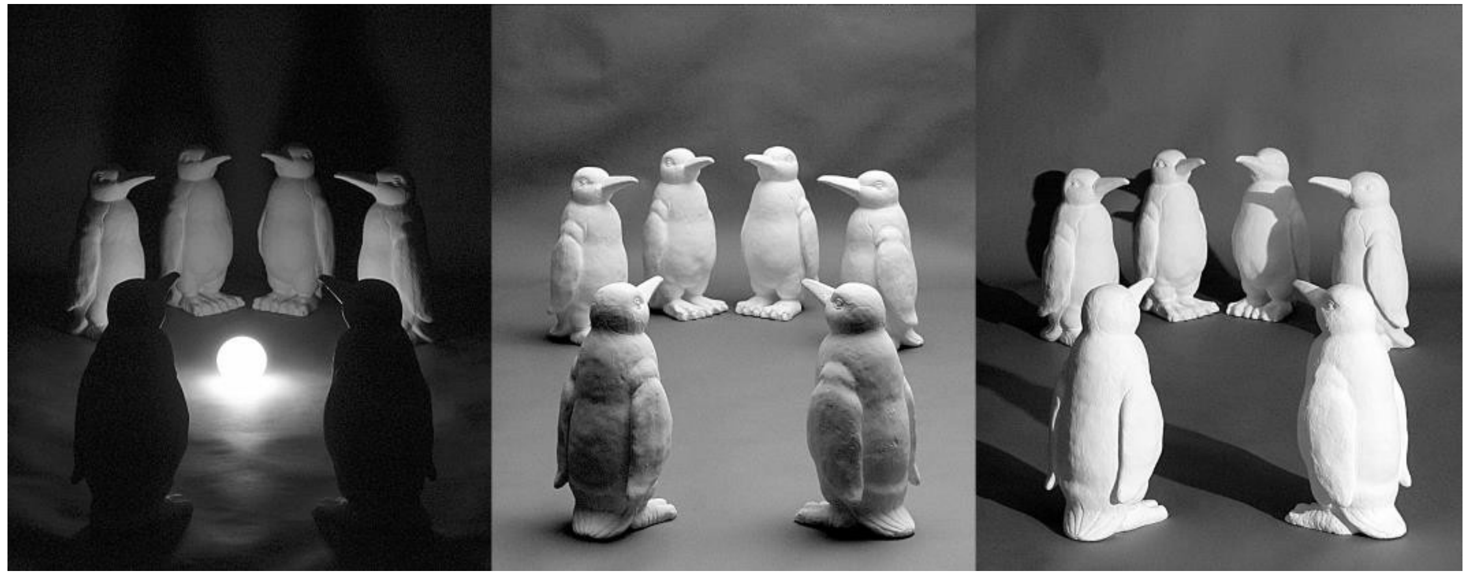
2. **Dense Optical Flow**: the flow vectors of all pixels in the entire frame are processed which, in turn, makes this technique slower but more accurate.

Optical flow

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

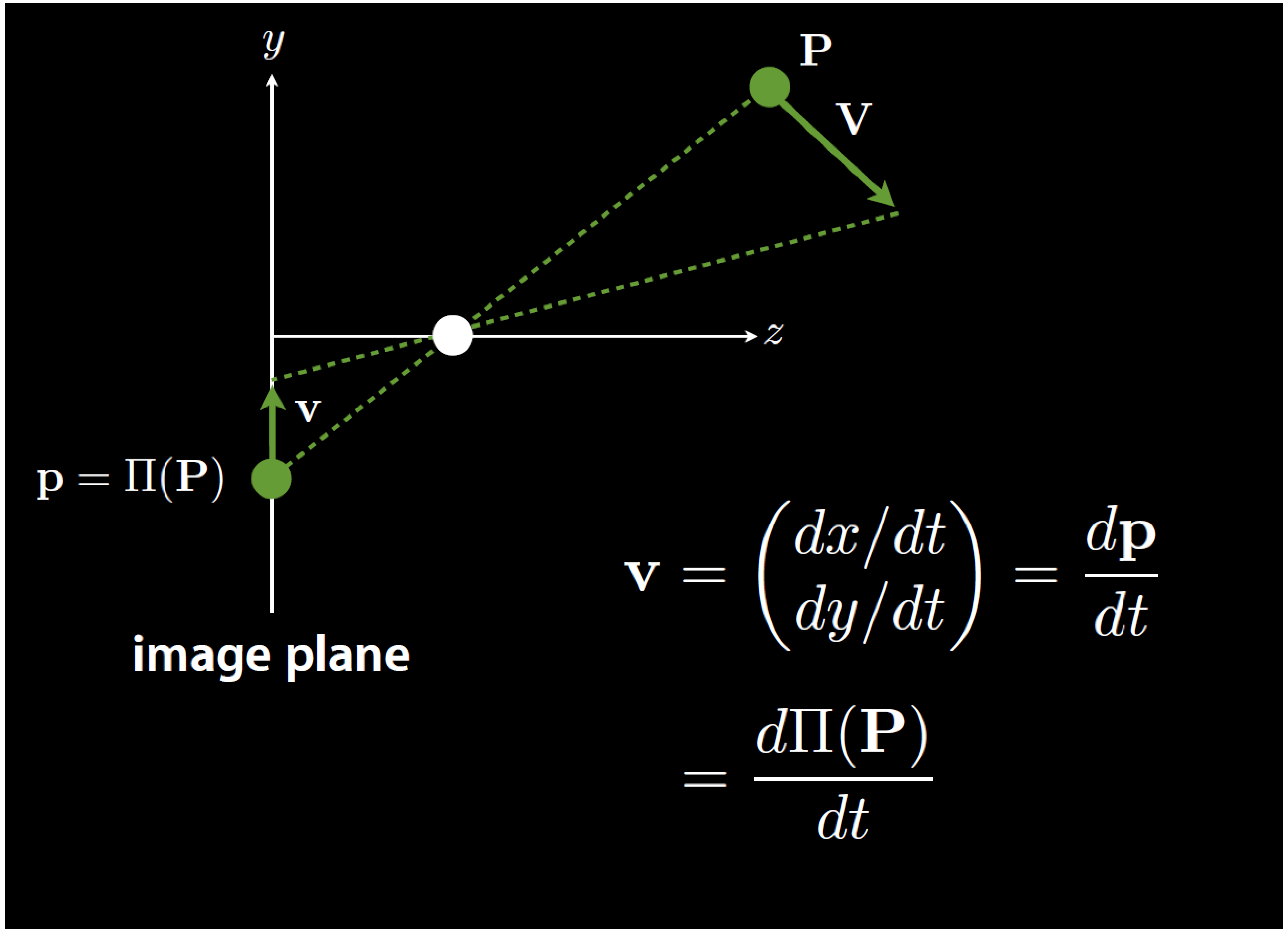
GOAL: Recover image motion at each pixel by optical flow. Pattern of motion of pixels between two consecutive frames. The motion can be caused either by the movement of a scene or by the movement of the camera.

Note: apparent motion can be caused by lighting changes without any actual motion (optical flow is different from the concept of motion field)



Motion field

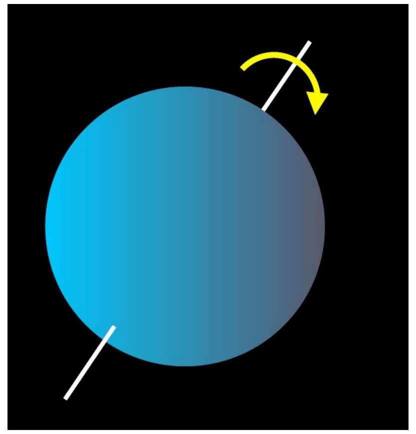
Projection of the 3D scene velocities onto the image plane

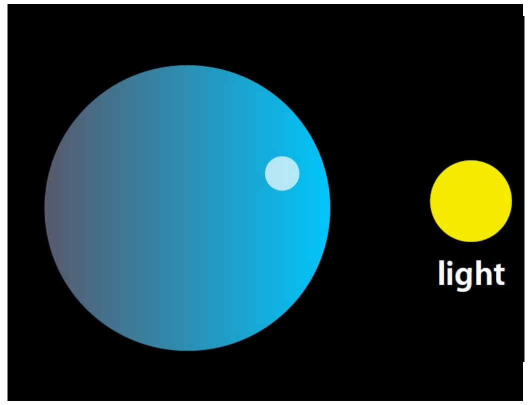


$$\mathbf{v} = \begin{pmatrix} dx/dt \\ dy/dt \end{pmatrix} = \frac{d\mathbf{p}}{dt}$$

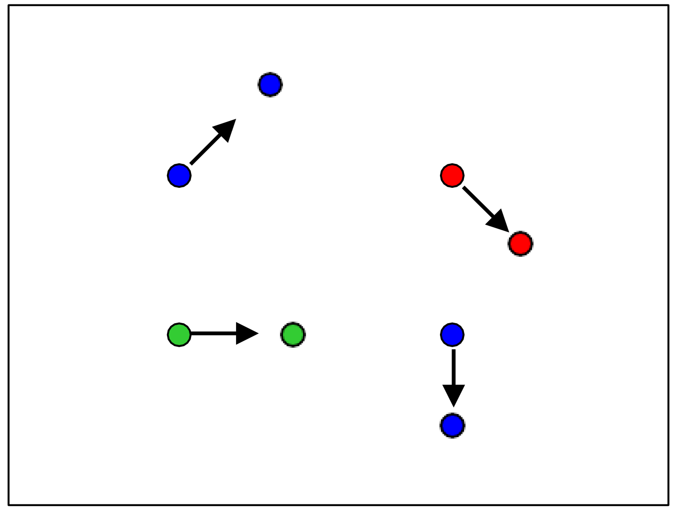
$$= \frac{d\Pi(\mathbf{P})}{dt}$$

It is a geometric concept
The optical flow is a photometric concept
Ideally OF e MF are equivalent

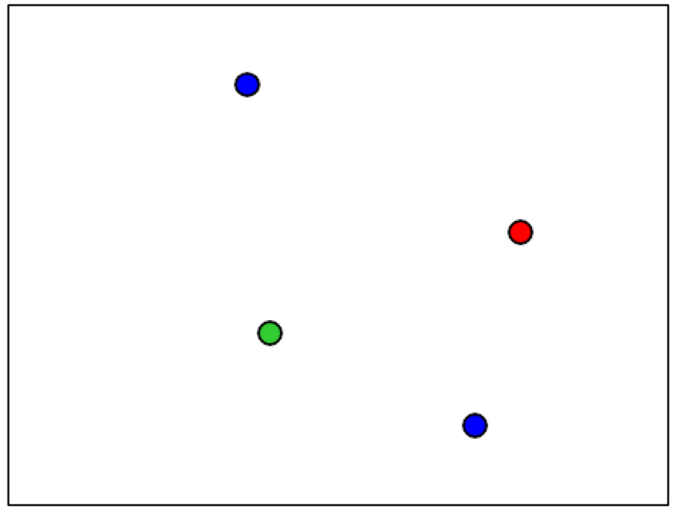




Estimating optical flow



$I(x,y,t)$



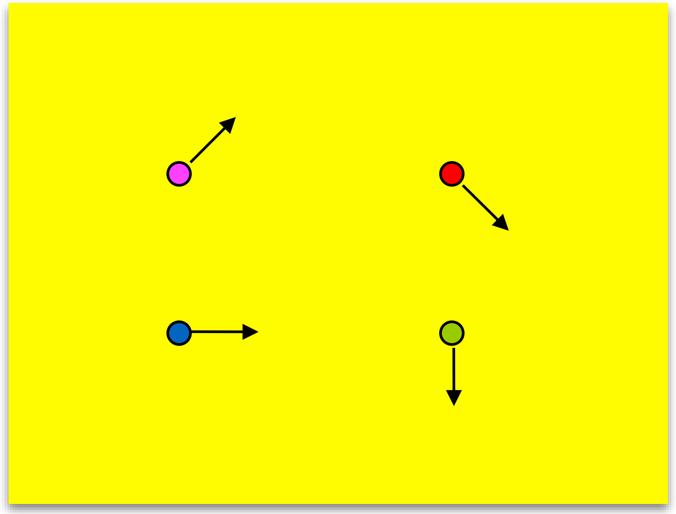
$I(x,y,t')$

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

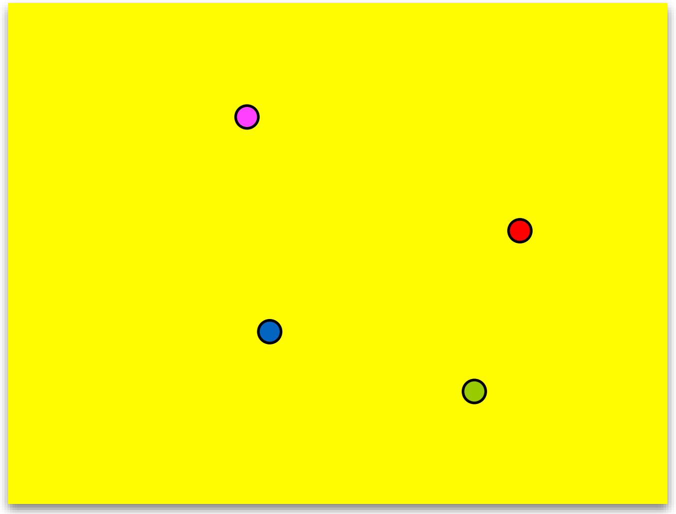
Key assumptions:

- **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 approach



$I(x, y, t)$



$I(x, y, t')$

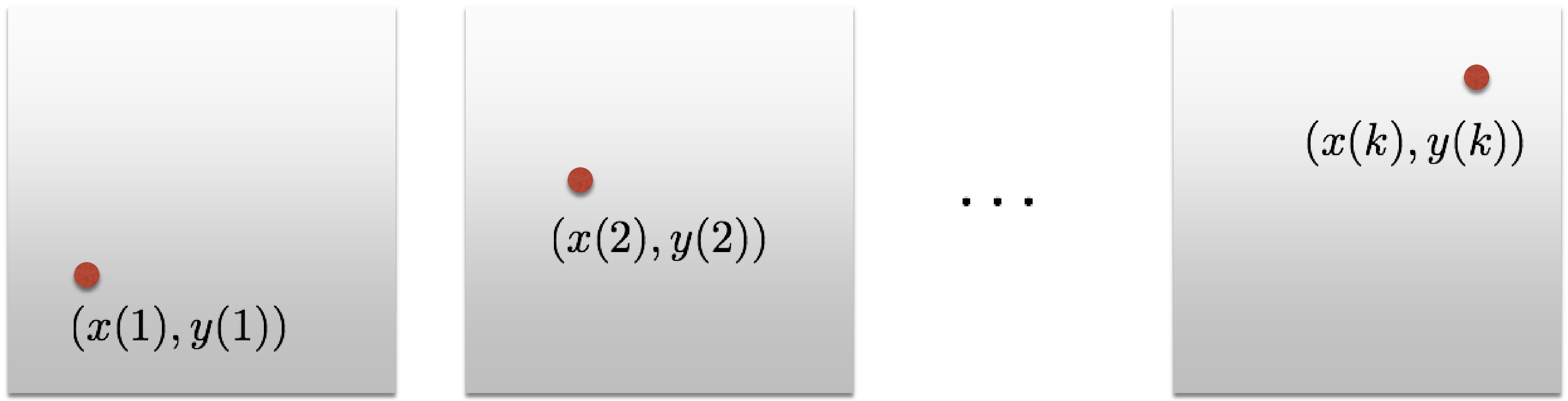
Look for *nearby pixels* with the *same color*

(small motion) (color constancy)

Assumption 1

Brightness constancy

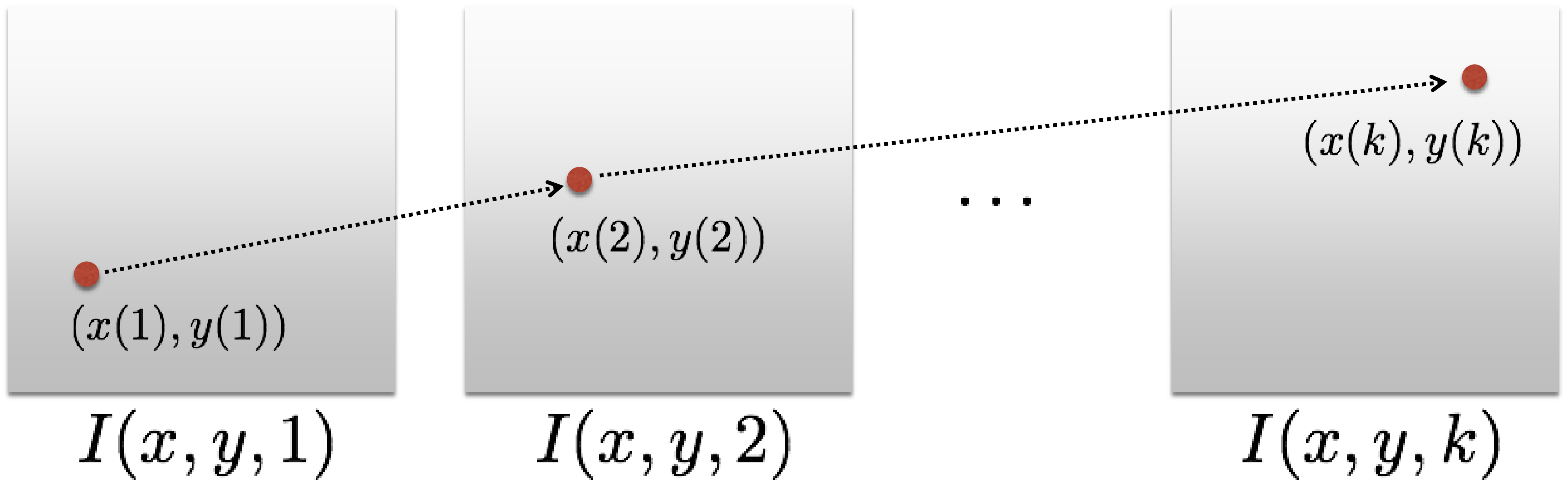
Scene point moving through image sequence



Assumption 1

Brightness constancy

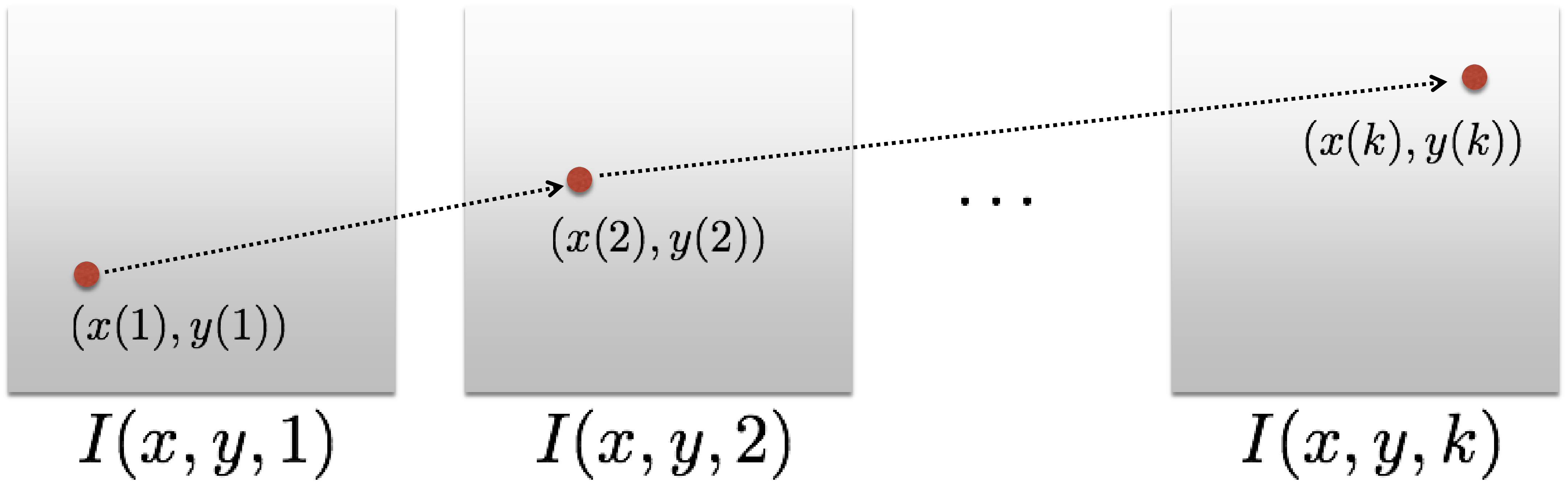
Scene point moving through image sequence



Assumption 1

Brightness constancy

Scene point moving through image sequence

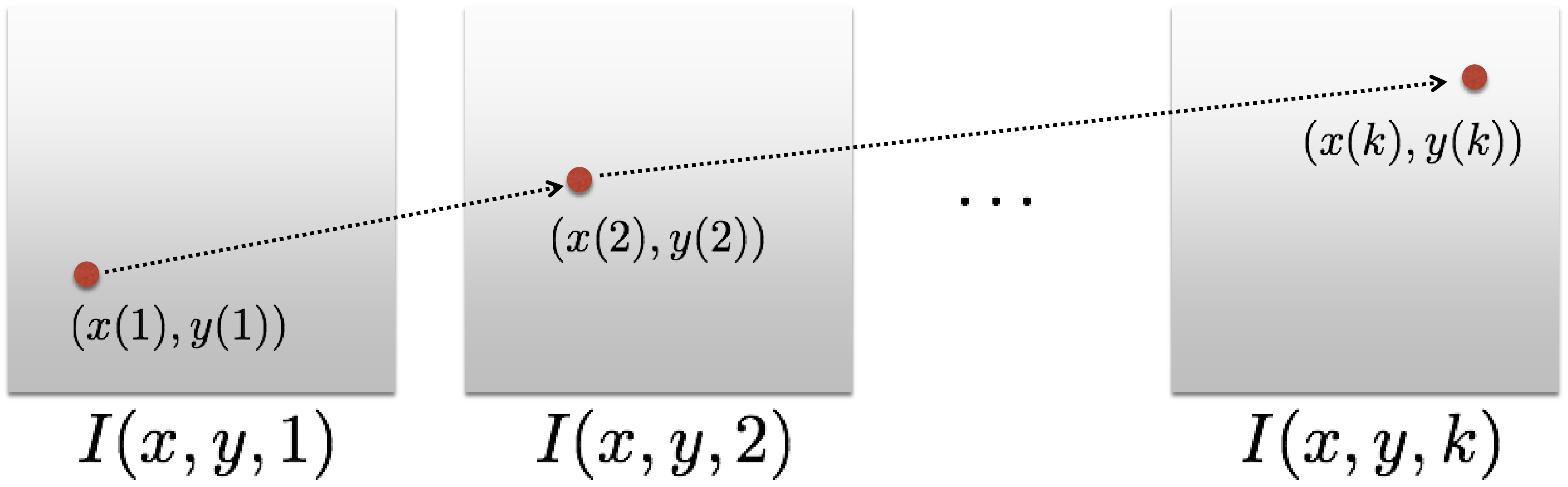


Assumption: brightness of the point will remain the same

Assumption 1

Brightness constancy

Scene point moving through image sequence



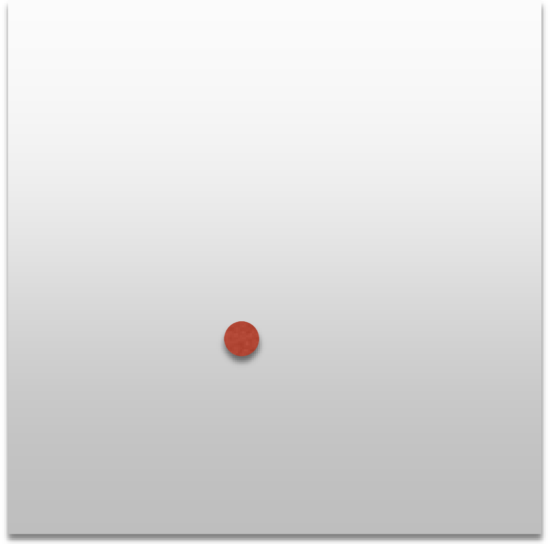
Assumption:Brightness of the point will remain the same

$$I(x(t), y(t), t) = C$$

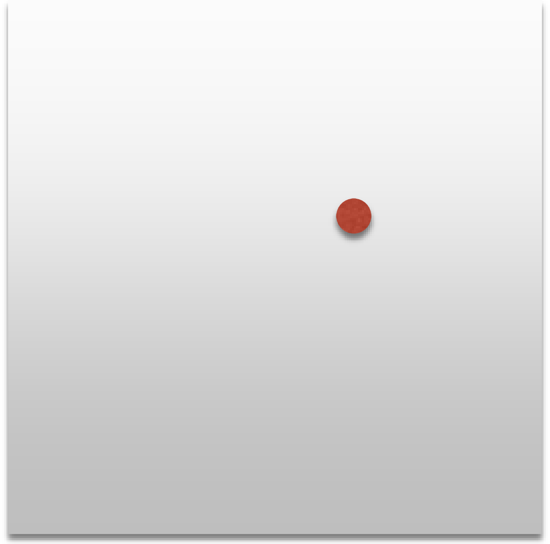
constant

Assumption 2

Small motion



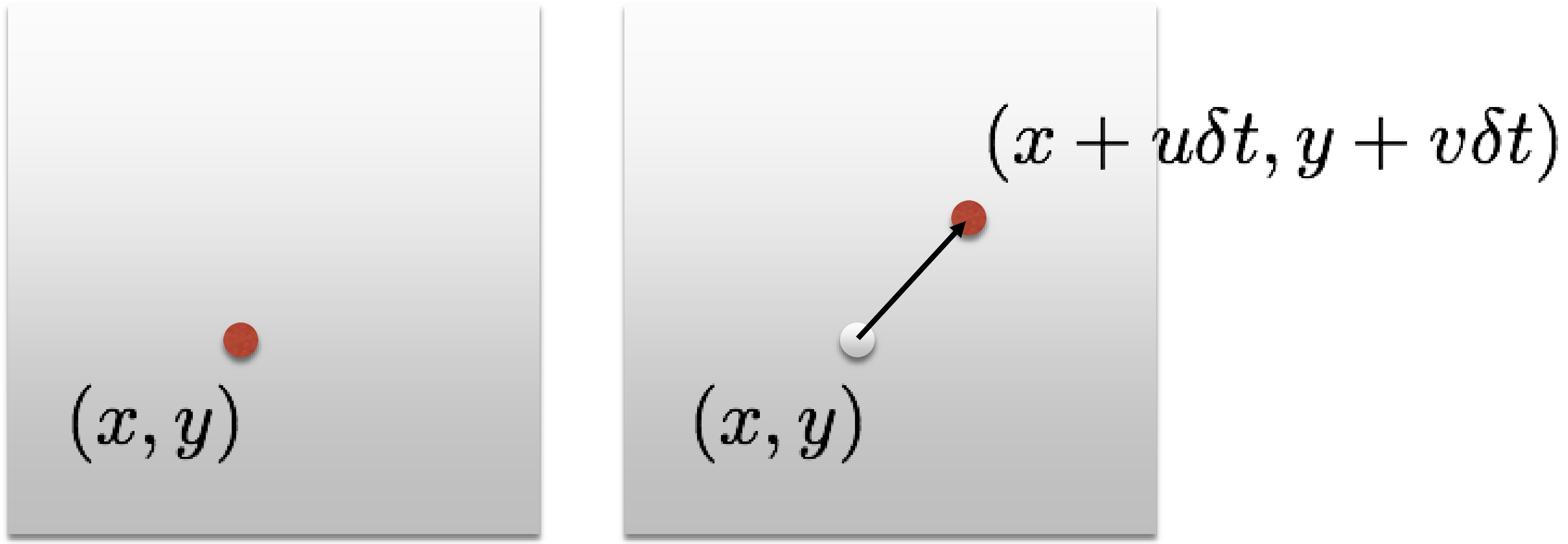
$I(x, y, t)$



$I(x, y, t + \delta t)$

Assumption 2

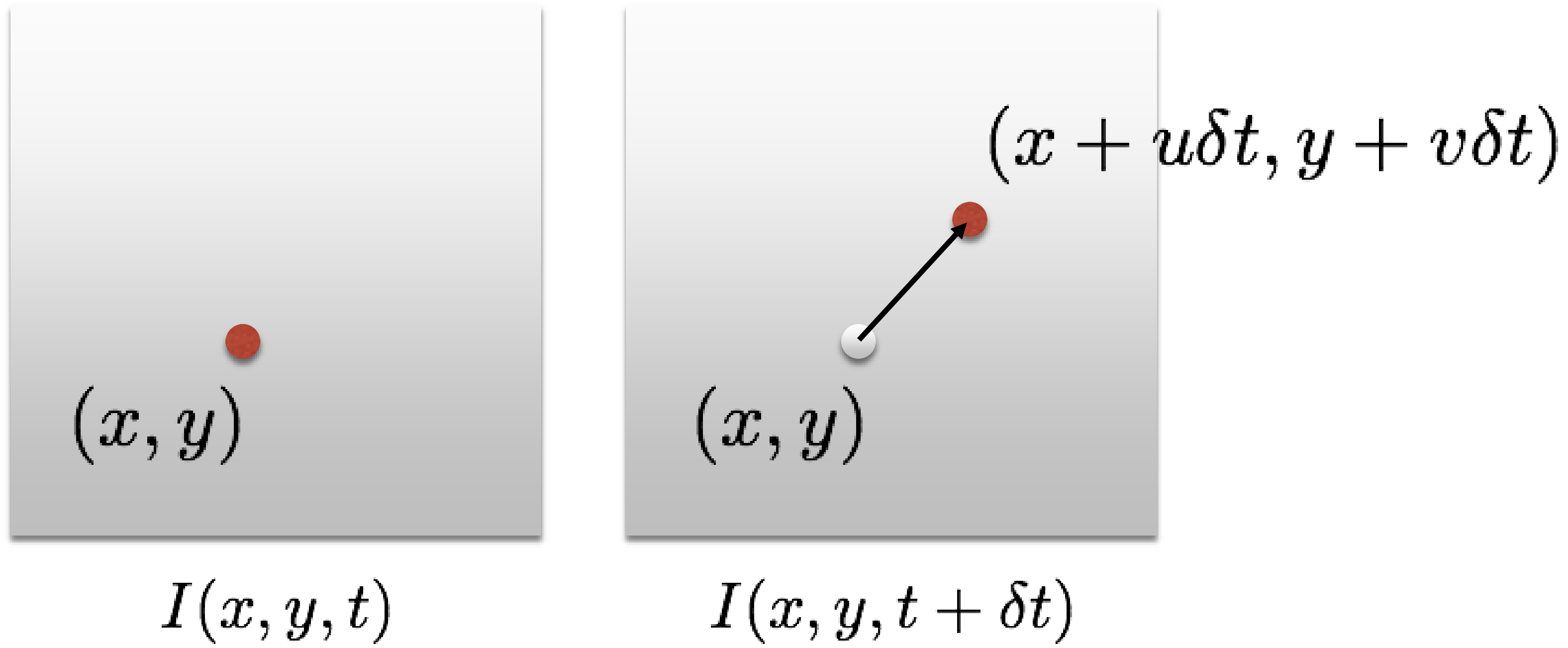
Small motion



$I(x, y, t)$ $I(x, y, t + \delta t)$

Assumption 2

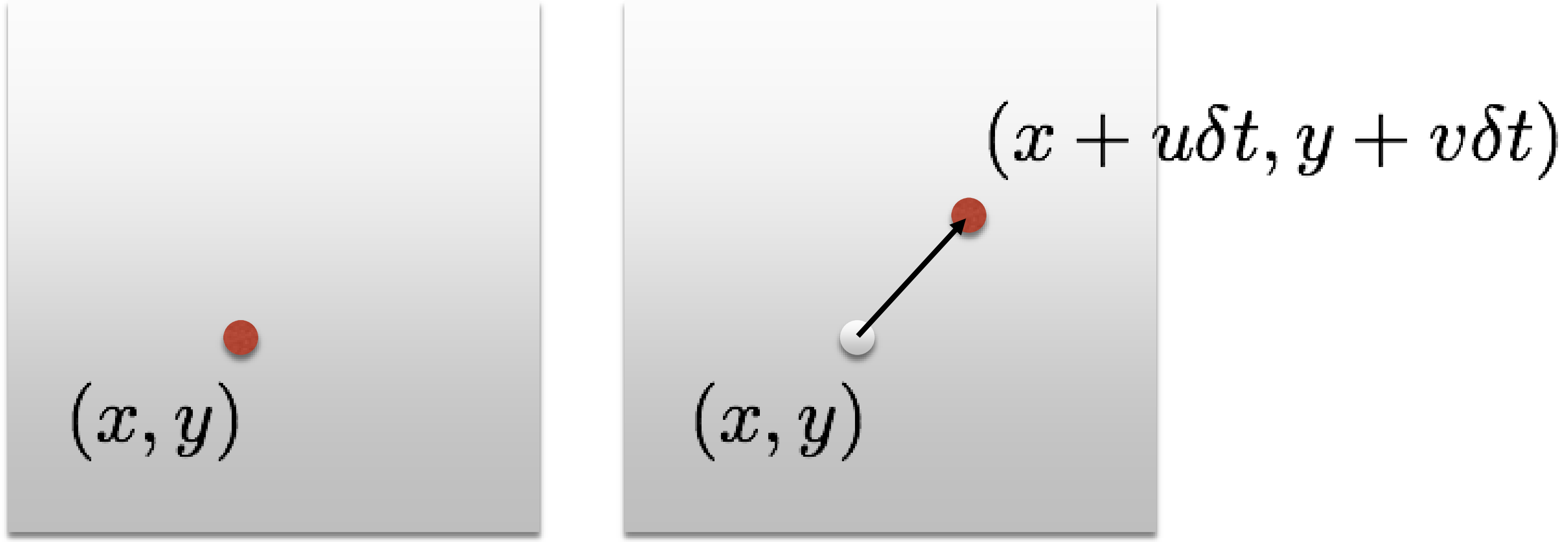
Small motion



Optical flow (velocities): (u, v) Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

Assumption 2

Small motion



$I(x, y, t)$ $\quad$ $I(x, y, t + \delta t)$

Optical flow (velocities): (u, v) $\quad$ Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

For a ***small space-time step***

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

the brightness between two consecutive image frames is the same

Taylor series expansion

For small space-time step, brightness of a point is the same

$$I(x + \underbrace{u\delta t}_{\delta x}, y + \underbrace{v\delta t}_{\delta y}, t + \delta t) = I(x, y, t)$$

Insight:
If the time step is really small, we can *linearize* the intensity function with first order approximation of the Taylor series expansion

Expand a function as an infinite sum of its derivatives
For one variable

$$f(x + \delta x) = f(x) + \frac{\partial f}{\partial x}\delta x + \frac{\partial^2 f}{\partial x^2}\frac{\delta x^2}{2!} + \cdots + \frac{\partial^n f}{\partial x^n}\frac{\delta x^n}{n!}$$

If δx is small:

$$f(x + \delta x) = f(x) + \frac{\partial f}{\partial x}\delta x + \boxed{O(\delta x^2)} \rightarrow \text{Almost Zero}$$

Taylor series expansion

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

$$\Downarrow \sim$$

$$I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = I(x, y, t)$$

assuming small motion

fixed point

Partial derivative

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

cancel terms

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

divide by δt

take limit $\delta t \rightarrow 0$

$$\boxed{\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0}$$

Brightness Constancy Equation

Brightness constancy equation

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

Brightness Constancy Equation

$$I_x u + I_y v + I_t = 0$$

(x-flow) (y-flow)

shorthand notation

$$\nabla I^\top \boldsymbol{v} + I_t = 0$$

(1 x 2) (2 x 1)

vector form

Brightness constancy equation

What do the term of the brightness constancy equation represent?

flow velocities

$$I_x u + I_y v + I_t = 0$$

Image gradients
(at a point p)

temporal gradient

Brightness constancy equation

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Derivative-of-Gaussian filter
…

Frame differencing

Frame differencing

Example of a forward difference

t

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

-

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

=

$$I_t = \frac{\partial I}{\partial t}$$

0	0	0	0	0
0	0	0	0	0
0	9	9	9	9
0	9	0	0	0
0	9	0	0	0
0	9	0	0	0

Example:

3 x 3 patch

t

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

$$I_x = \frac{\partial I}{\partial x}$$

-	0	0	0	-
-	0	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-
-	9	0	0	-

-1 0 1

$$I_y = \frac{\partial I}{\partial y}$$

-	-	-	-	-
0	0	0	0	0
0	9	9	9	9
-	0	0	0	0
0	0	0	0	0
-	-	-	-	-

-1
0
1

$$I_t = \frac{\partial I}{\partial t}$$

0	0	0	0	0
0	0	0	0	0
0	9	9	9	9
0	9	0	0	0
0	9	0	0	0
0	9	0	0	0

Brightness constancy equation

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Derivative-of-Gaussian filter
…

unknown

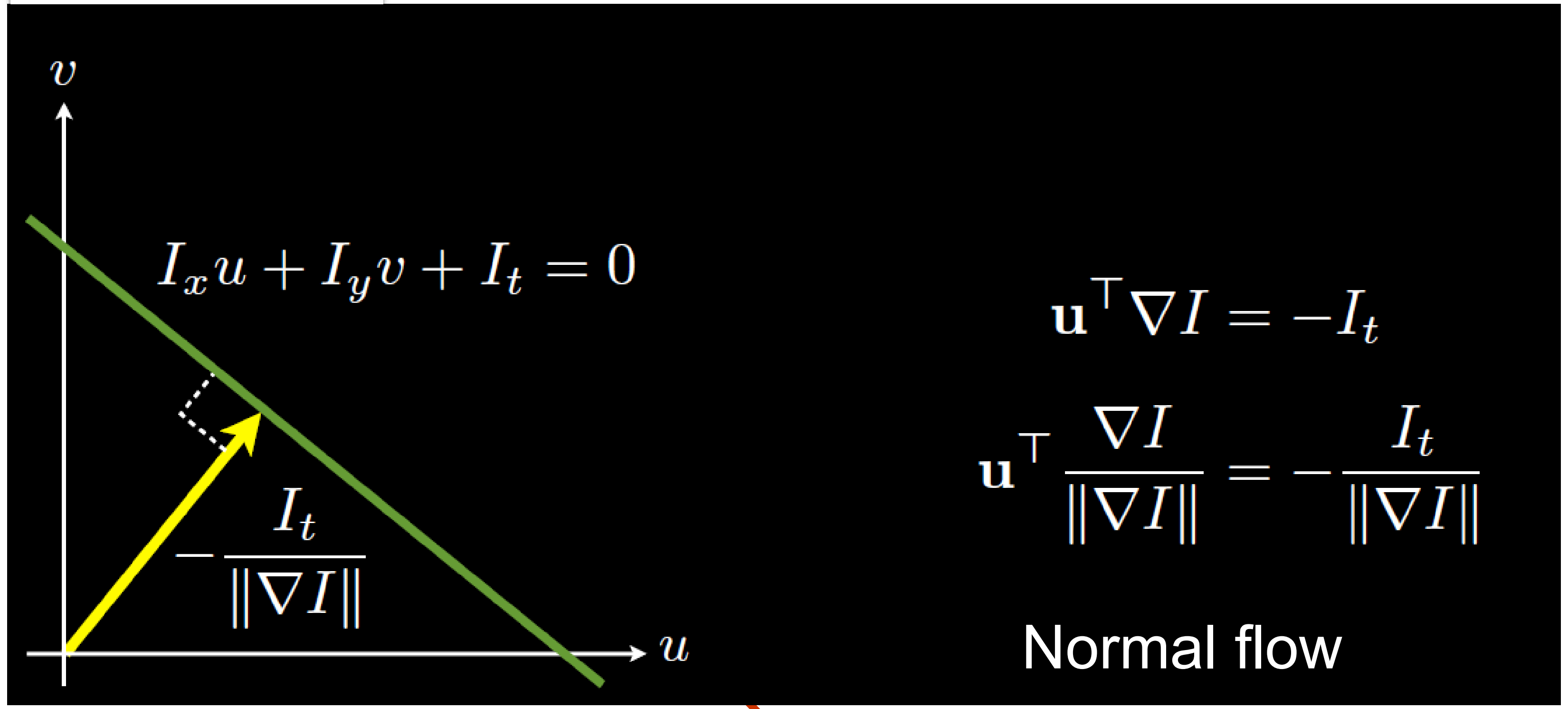
Frame differencing

Brightness constancy equation

Solution lies on a straight line

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

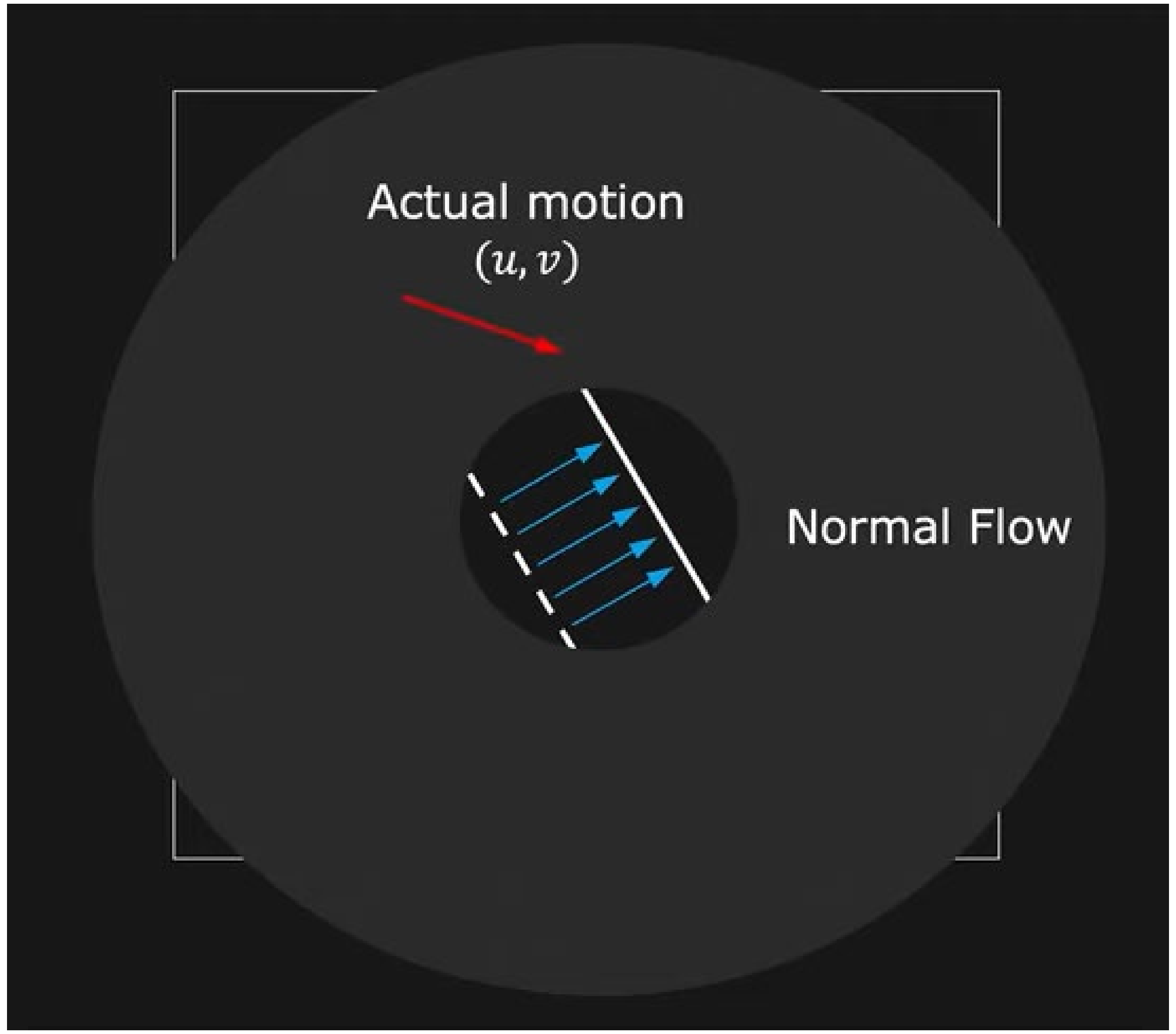
many combinations of u and v will satisfy the equality



Optical flow can be split in two components: normal flow and parallel flow

The solution cannot be determined uniquely with a single constraint (a single pixel). The solution is underconstrained
Where do we get more equations (constraints)?

Aperture problem



Locally we can only determine normal flow

Two main methods

- **Lucas-Kanade Optical Flow (1981)**
  method of differences
  'constant' flow (flow is constant for all pixels)
  local method (sparse)

- **Horn-Schunck Optical Flow (1981)**
  brightness constancy, small motion
  'smooth' flow (flow can vary from pixel to pixel)
  global method (dense)

Lucas-Kanade Optical Flow

Assumptions

Assume that the surrounding patch has '**constant flow**'
Neighboring pixels have same displacement: i.e for each pixel assume motion field, and hence optical flow (u,v), is constant within a small neighborhood.

Using a 5 x 5 image patch, gives us 25 equations

$$I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v = -I_t(\boldsymbol{p}_1)$$
$$I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v = -I_t(\boldsymbol{p}_2)$$
$$\vdots$$
$$I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v = -I_t(\boldsymbol{p}_{25})$$

n^2 (25) equations,
2 unknown: find least square solution

Least square solution

Solve linear system Ax=b
- that it is equivalent to write (least squares using pseudo-inverse)

$$A^\top A \qquad\qquad x \qquad\qquad A^\top b$$

'Lucas-Kanade Optical Flow'

$$\left[\begin{array}{cc} \sum\limits_{p\in P} I_x I_x & \sum\limits_{p\in P} I_x I_y \\ \sum\limits_{p\in P} I_y I_x & \sum\limits_{p\in P} I_y I_y \end{array}\right] \left[\begin{array}{c} u \\ v \end{array}\right] = - \left[\begin{array}{c} \sum\limits_{p\in P} I_x I_t \\ \sum\limits_{p\in P} I_y I_t \end{array}\right]$$

$$2\times2 \qquad\qquad 2\times1 \qquad\qquad 2\times1$$

where the summation is over each pixel ***p*** in patch ***P***

$$x = (A^\top A)^{-1} A^\top b$$

When does optical flow estimation work?

$$x = (A^\top A)^{-1} A^\top b$$

$A^\top A$ should be invertible (det !=0)

$A^\top A$ should be well conditioned

λ_1 and λ_2 should not be too small (both are significant enough)

λ_1 should not be too large respect to λ_2 (λ_1=larger eigenvalue)

When does optical flow estimation work?

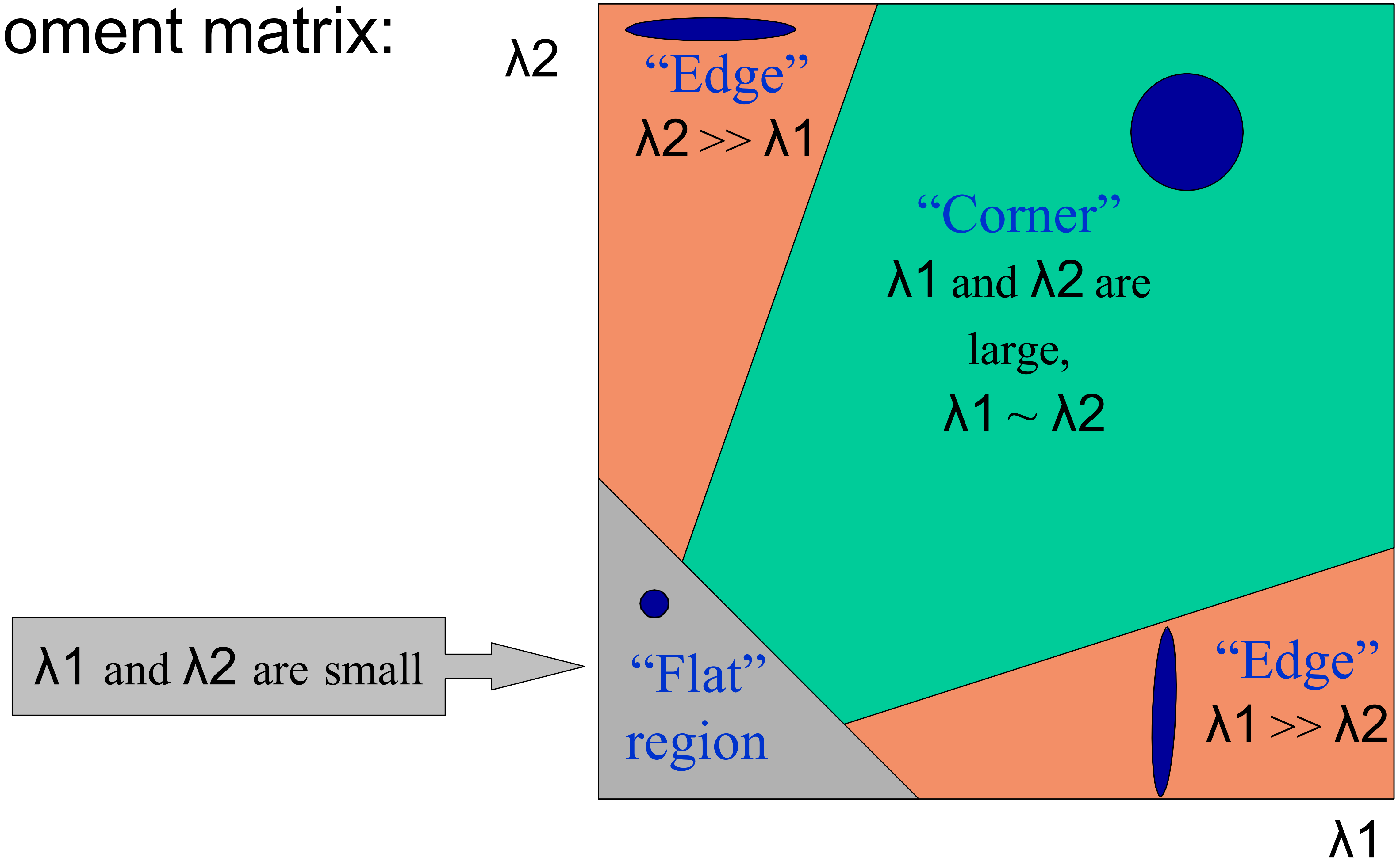
A^TA is the *second moment matrix* (Harris corner detector)

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

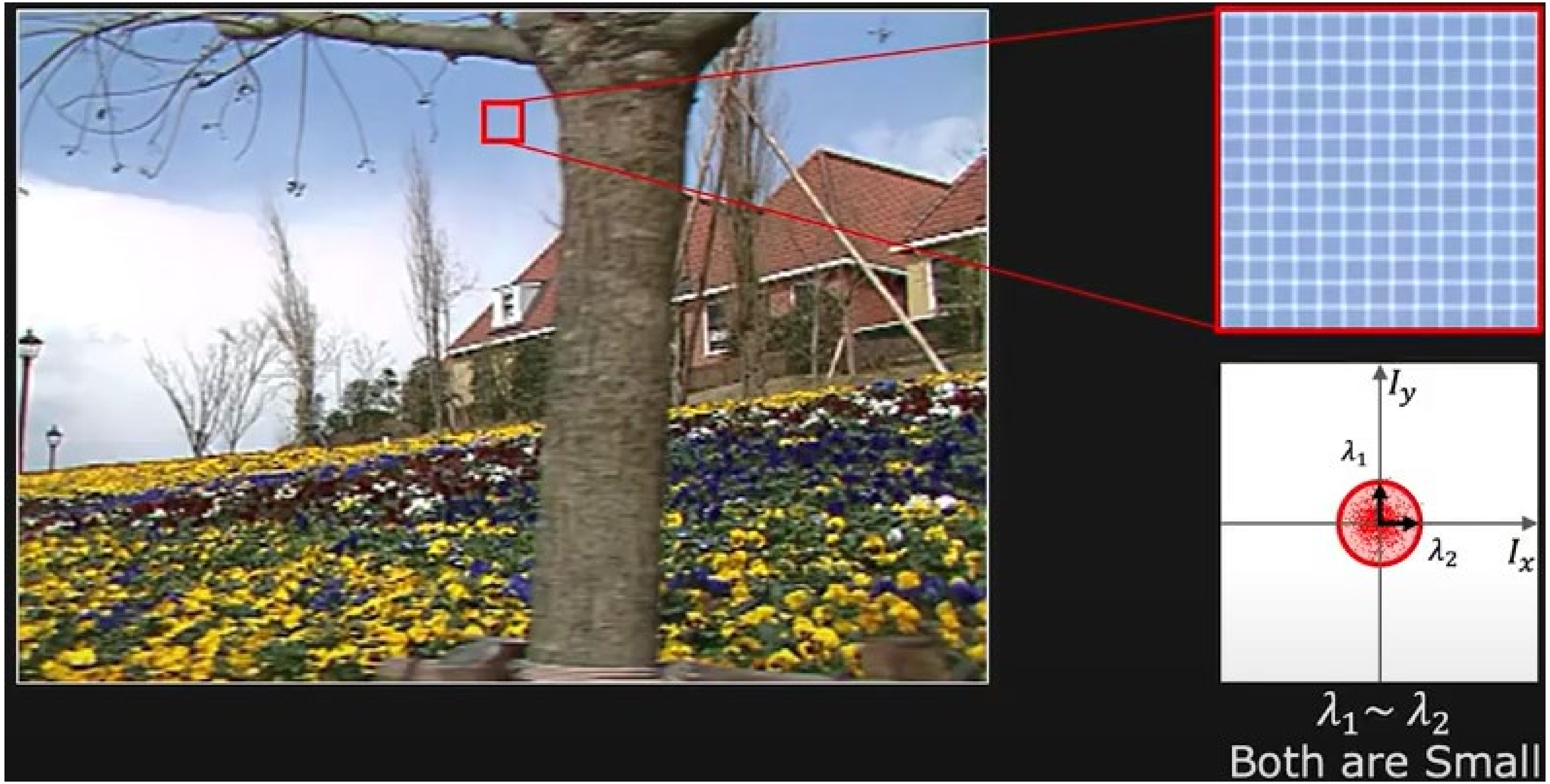
- Eigenvectors and eigenvalues of A^TA relate to edge direction and magnitude
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change
  - The other eigenvector is orthogonal to it

Interpreting the eigenvalues

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

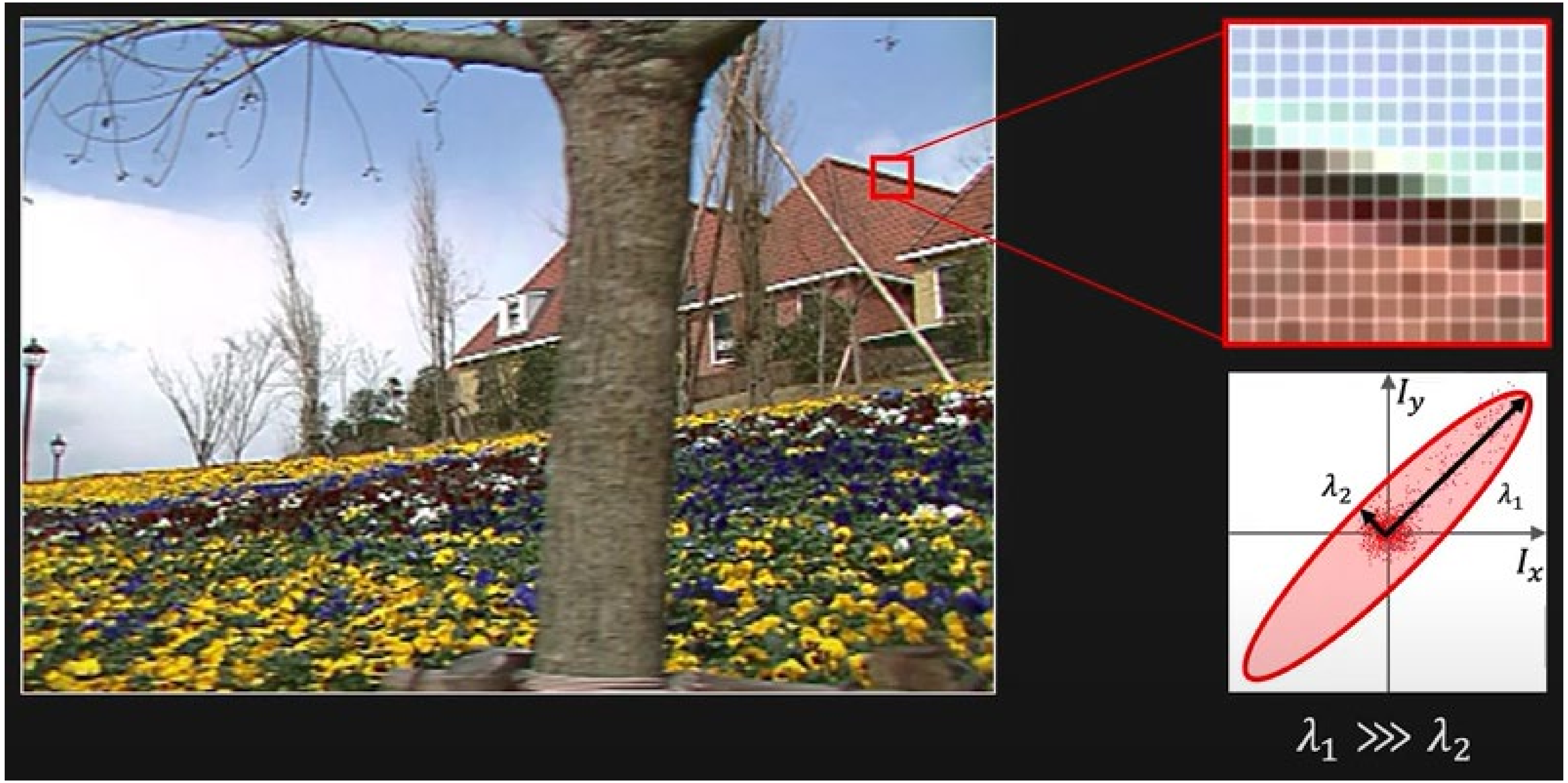


Low-texture region



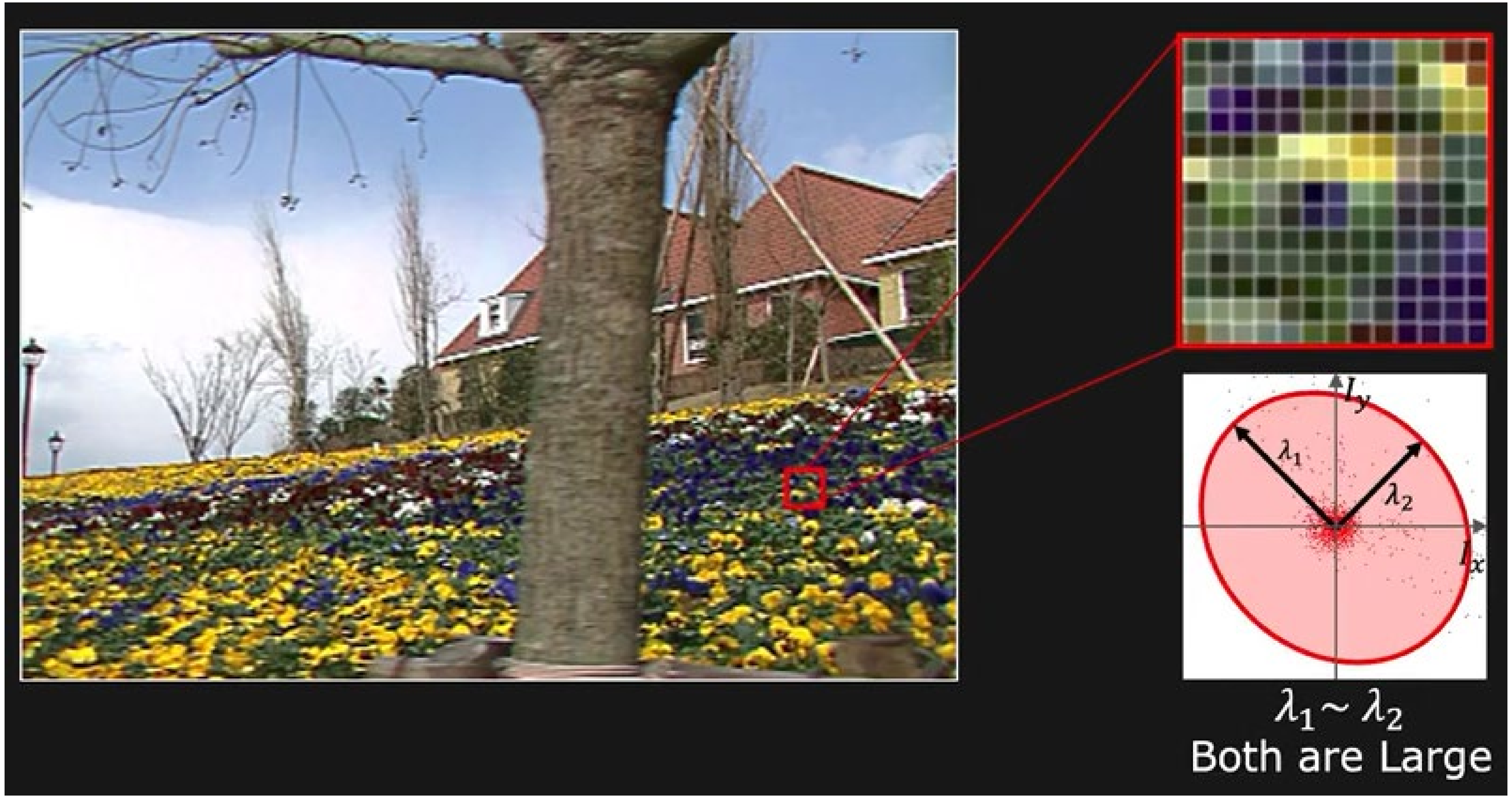
- Equations for all pixels in window are more or less the same
- Gradients have small magnitude
- Can not reliably compute the flow

Edges



– gradients very large or very small
– prominent gradient in one direction-→ badly conditioned
– can not reliably compute the flow

High-texture region

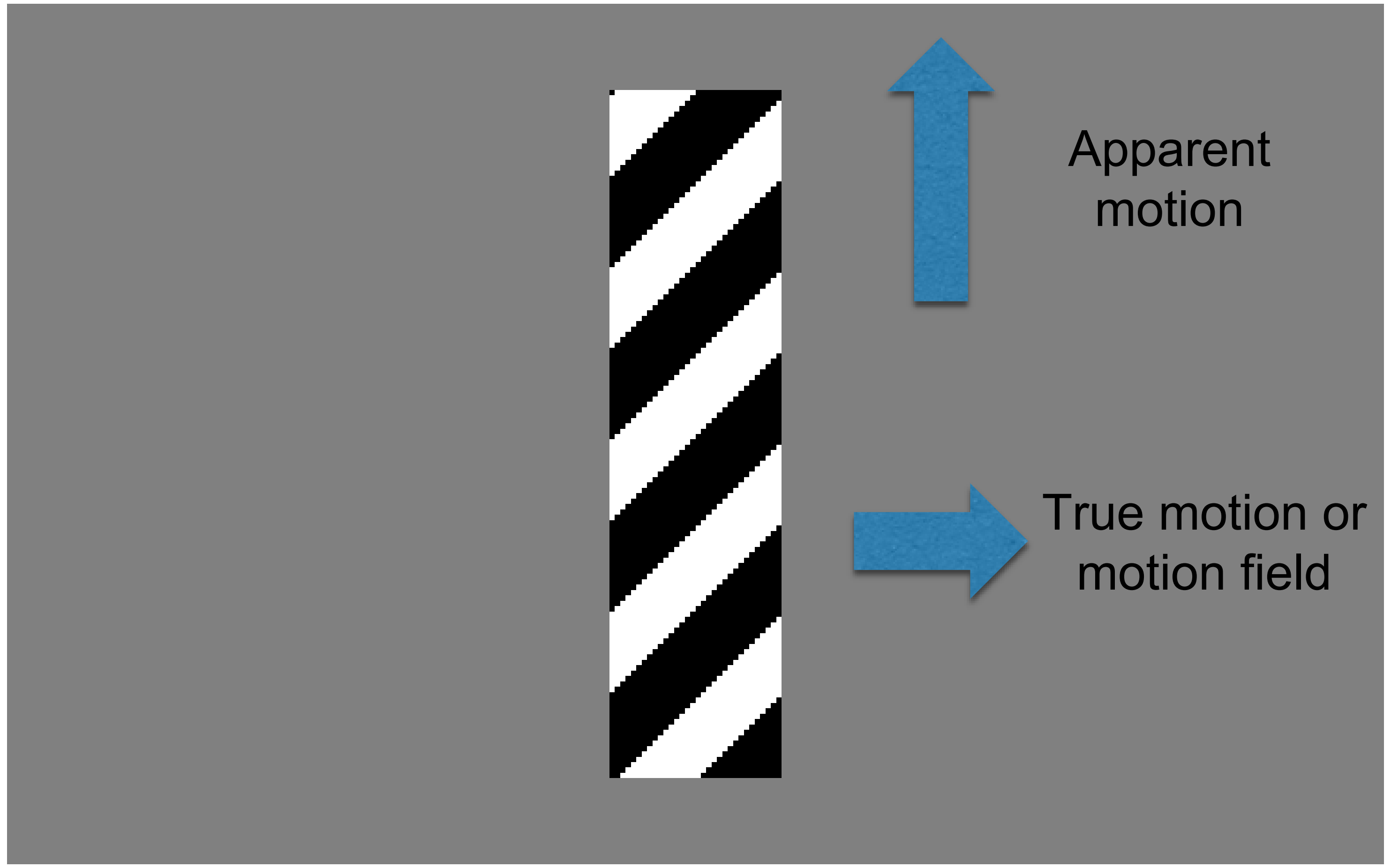


– gradients are different, large magnitudes
– well conditioned
– Can reliably compute optical flow

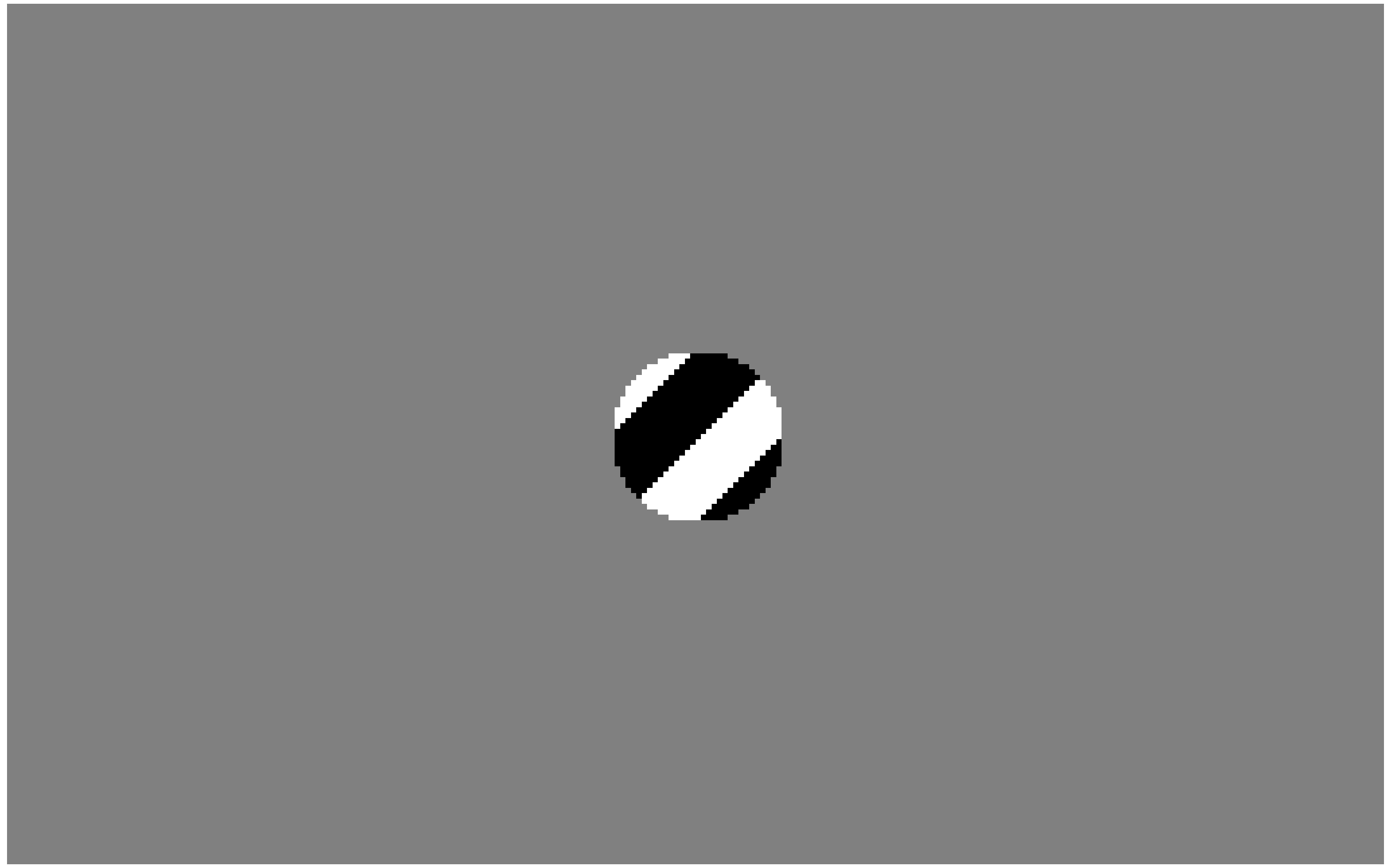
Implications

- Corners and high texture regions are when $\lambda 1$, $\lambda 2$ are big; this is also when Lucas-Kanade optical flow works best
- Corners and high texture regions are regions with two different directions of gradient (at least)
- Corners and high texture regions are good places to compute flow

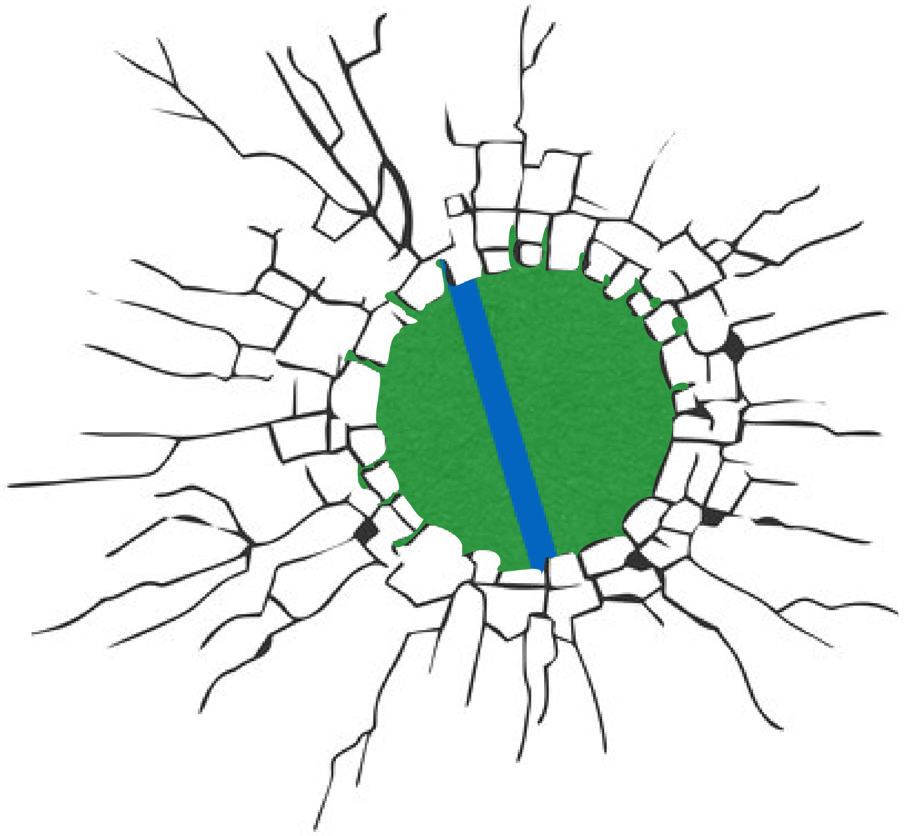
Barber’s pole illusion



Barber's pole illusion



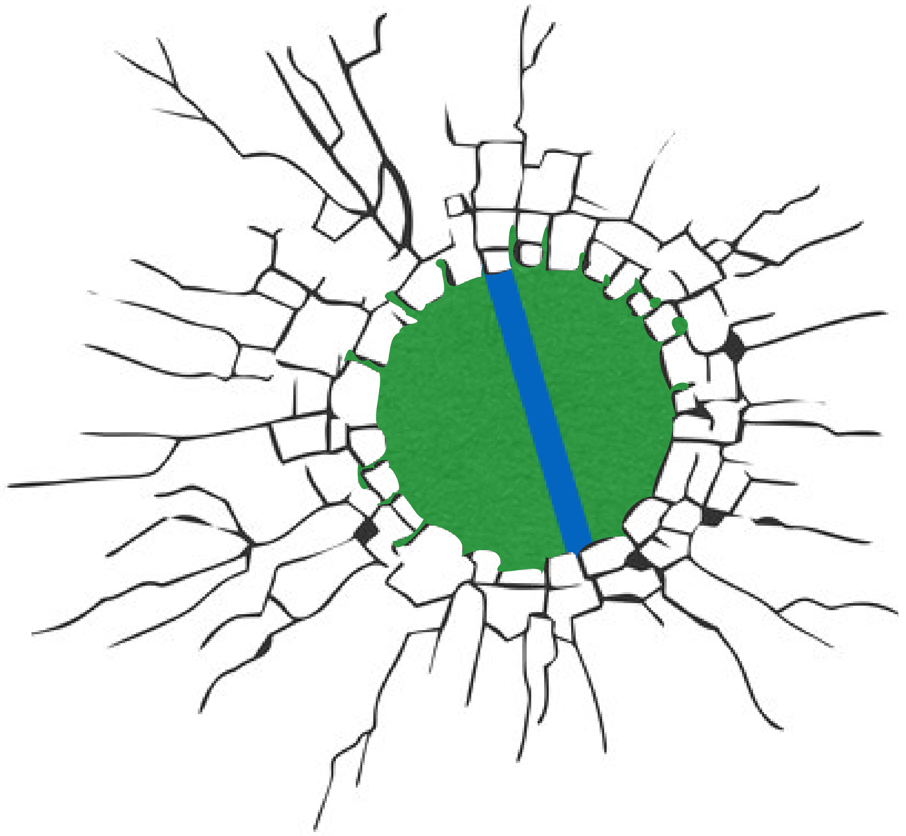
Aperture Problem



small visible image patch

In which direction is the line moving?

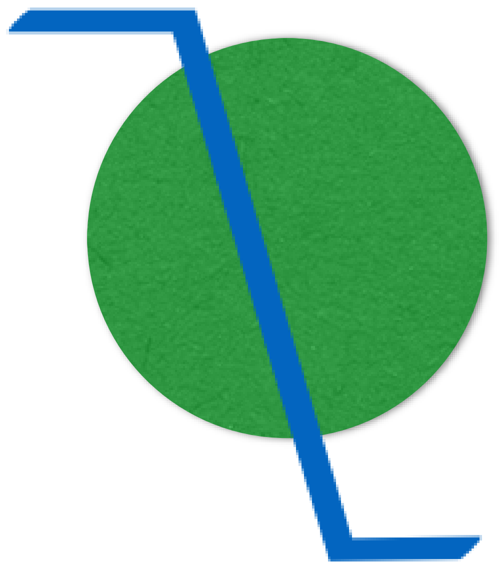
Aperture Problem



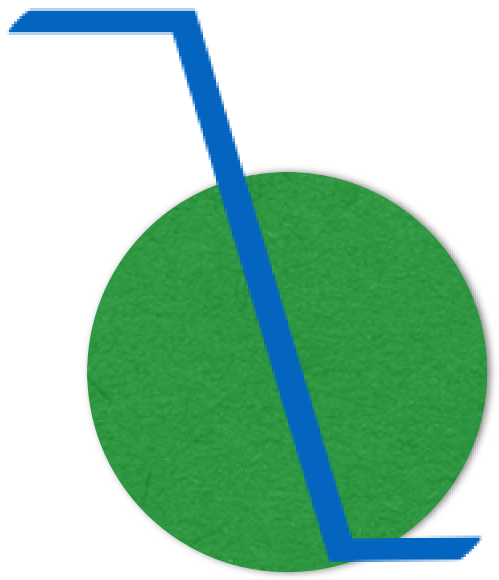
small visible image patch

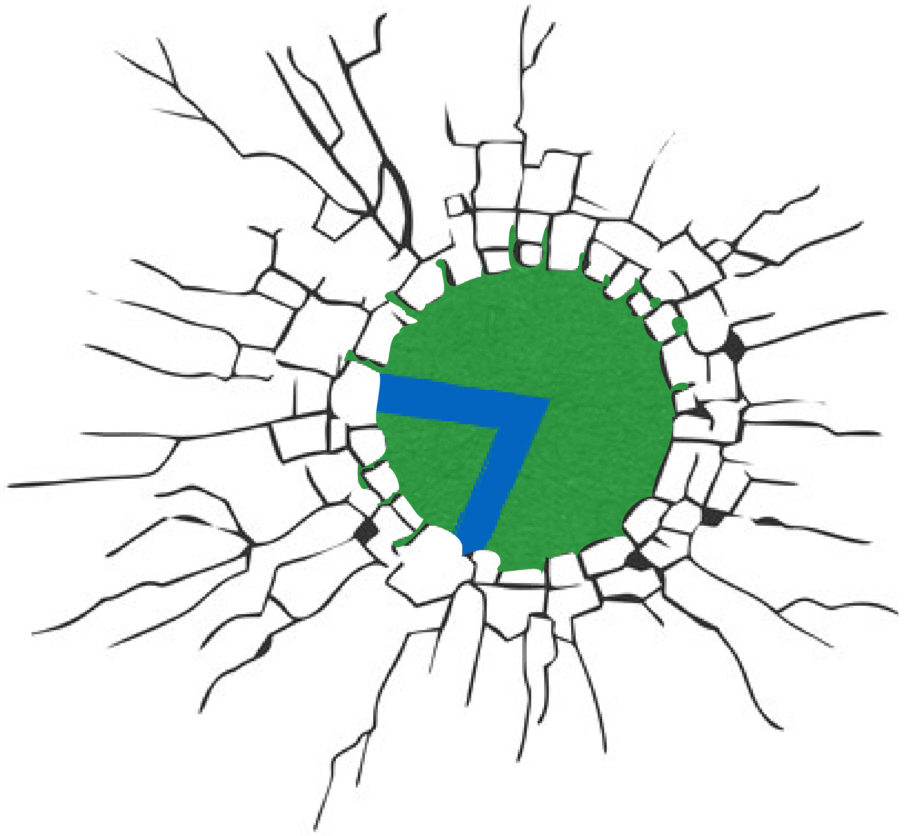
In which direction is the line moving?

Aperture Problem

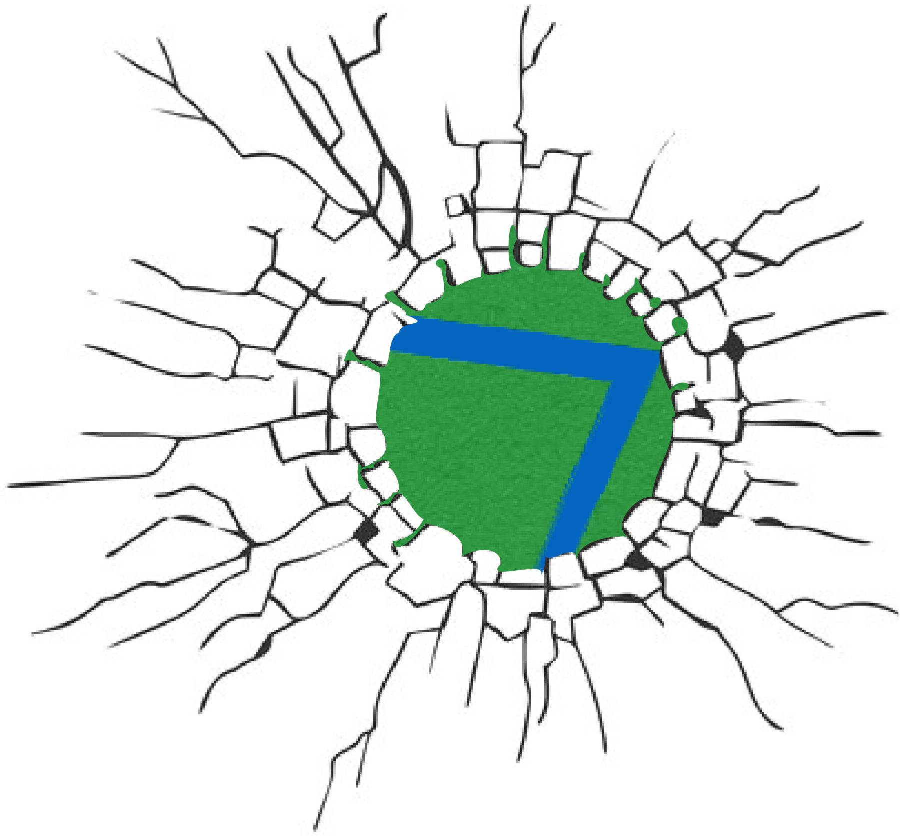


Aperture Problem





Want patches with different gradients to avoid the aperture problem



Want patches with different gradients to avoid the aperture problem

Aperture Problem: example

H(x,y)

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5

x

y

optical flow: (1,1)

I(x,y)

-	-	-	-
-	1	1	1
-	2	2	2
-	3	3	3
-	4	4	4

x

y

$$\cancel{I_x u} + I_y v + I_t = 0$$

Compute gradients

$$I_x(3,3) = 0$$

$$I_y(3,3) = 1$$

$$I_t(3,3) = I(3,3) - H(3,3) = -1$$

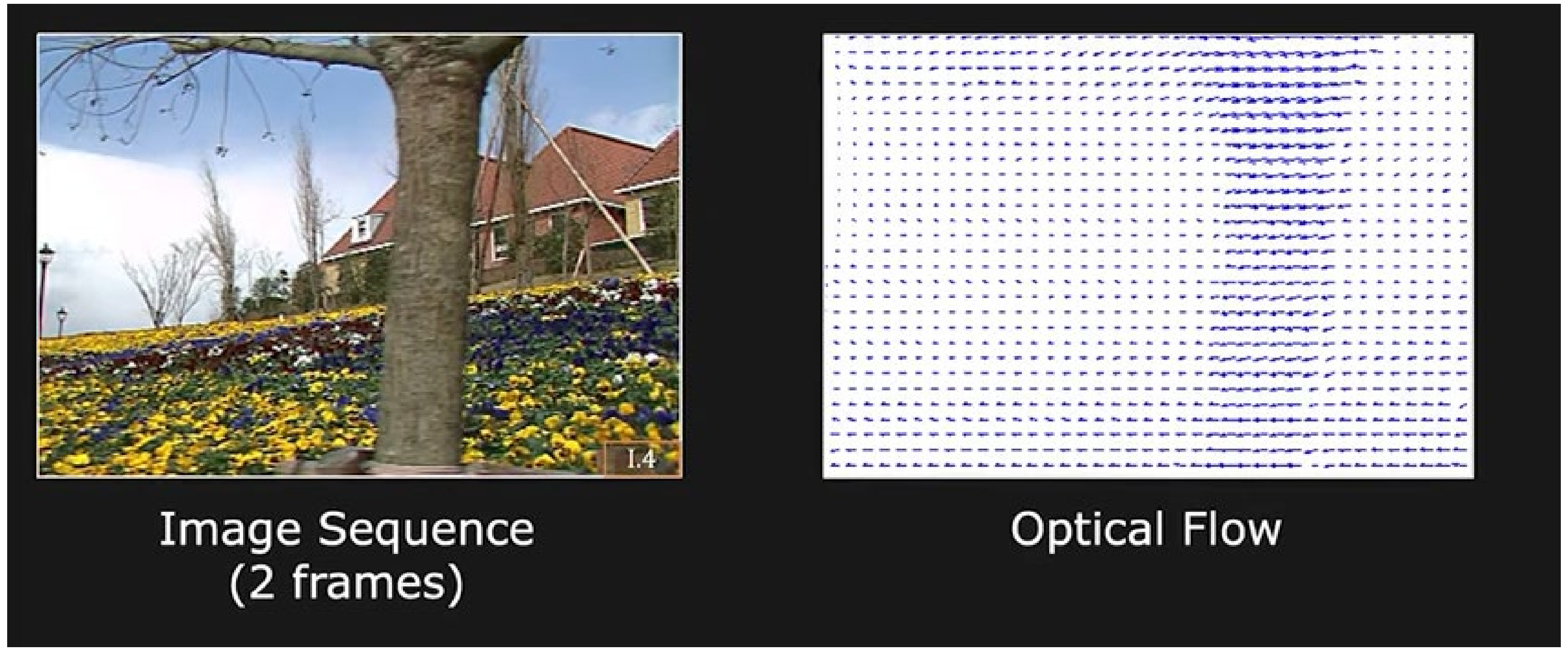
Solution:

$$v = 1$$

We recover the v of the optical flow but not the u.

Lukas-Kanade method

- Decide for local neighborhood of WxW pixels and apply uniformly in frames (smoothing frame first with Gaussian filter with a small standard deviation sigma=1.5)
- At frame t, t+1 calculate the derivatives Ix, Iy, It
- For each couple of frames obtain equational system and solve in the least square sense calcuating the eigenvalues
- Plot the optical flow vectors (directions and magnitude)



Two main methods

- **Lucas-Kanade Optical Flow (1981)**
  method of differences
  'constant' flow (flow is constant for all pixels)
  local method (sparse)

- **Horn-Schunck Optical Flow (1981)**
  brightness constancy, small motion
  'smooth' flow (flow can vary from pixel to pixel)
  global method (dense)

Key idea

(of Horn-Schunck optical flow)

In order to compute optical flow:

Enforce
brightness constancy

Enforce
smooth flow field

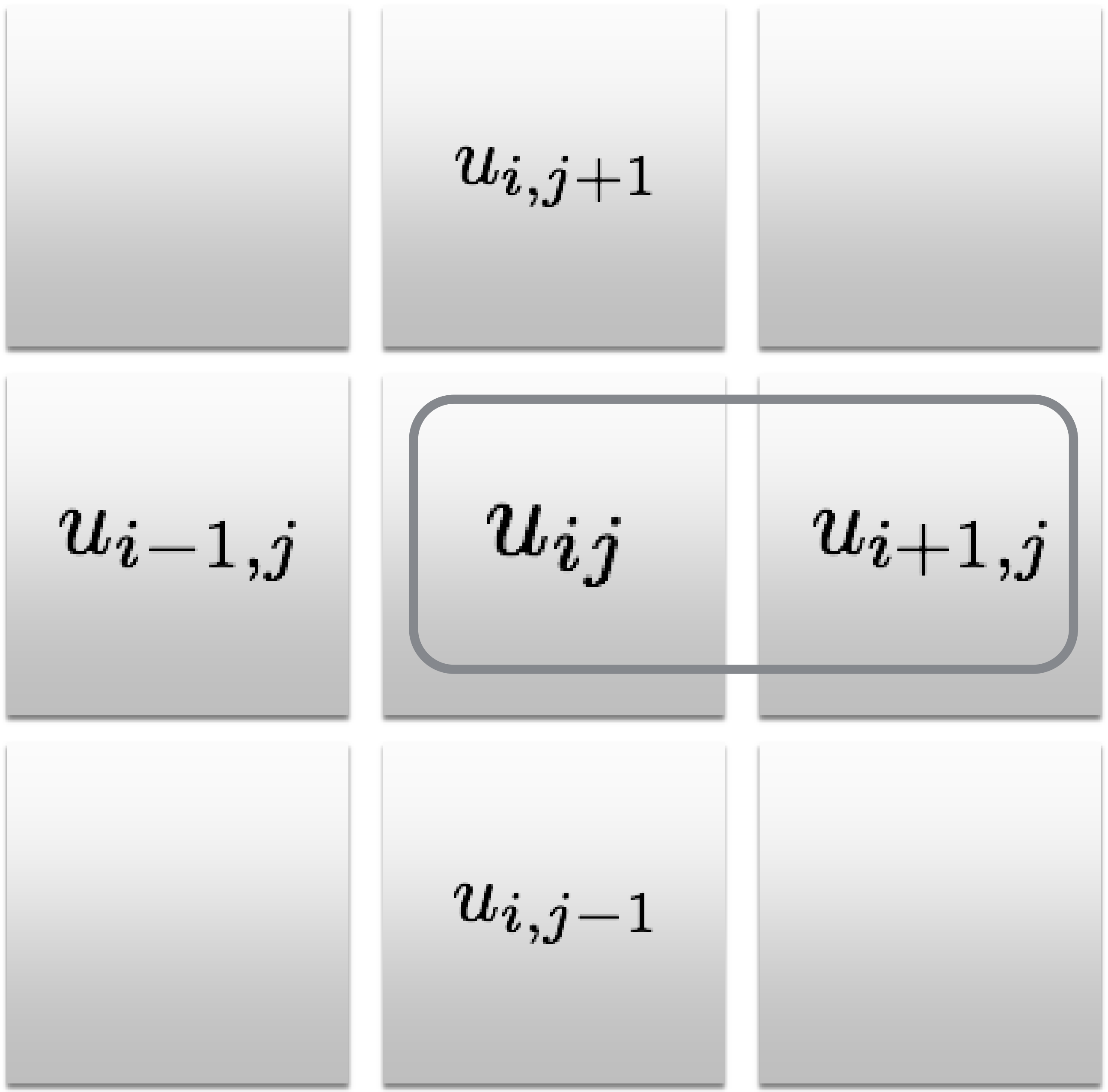
Enforce
brightness constancy

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

For every pixel, we need to solve the second moment matrix that is approximated by a quadratic form as you remember from HCD

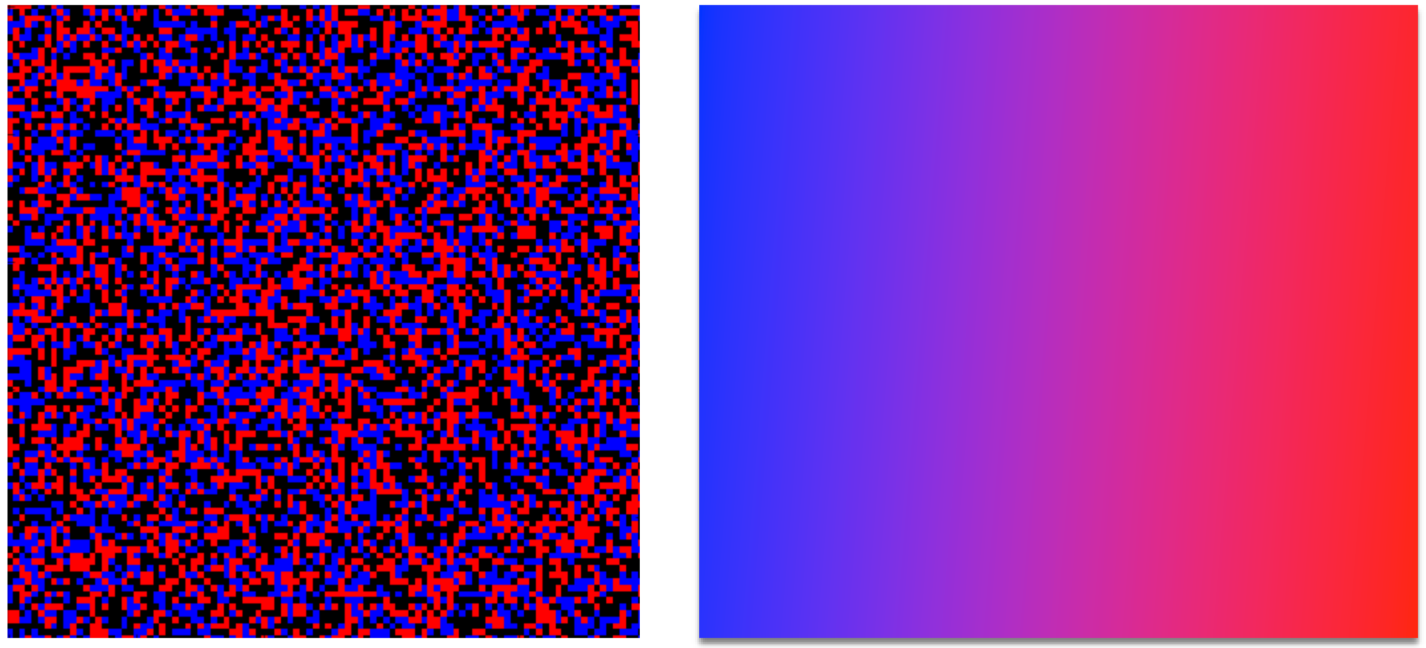
$$\min_{u,v} \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

Enforce **smooth flow field**



u-component of flow

$$\min_{\boldsymbol{u}} (u_{i,j} - u_{i+1,j})^2$$



- we expect optical flow fields to be smooth
- **most objects in the world are rigid or deform elastically moving together coherently**

Horn-Schunck (HS) optical flow

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \Big\{ \overbrace{E_s(i,j)}^{\text{smoothness}} + \overbrace{\underbrace{\lambda}_{\text{weight}} E_d(i,j)}^{\text{brightness constancy}} \Big\}$$

Larger values lead to a smoother flow

Optimization problem

- HS algorithm assume smoothness in the flow over the whole image
- The flow is formulated as a global energy function which is then sought to be minimized
- Compute partial derivative, derive update equations (gradient descent!)

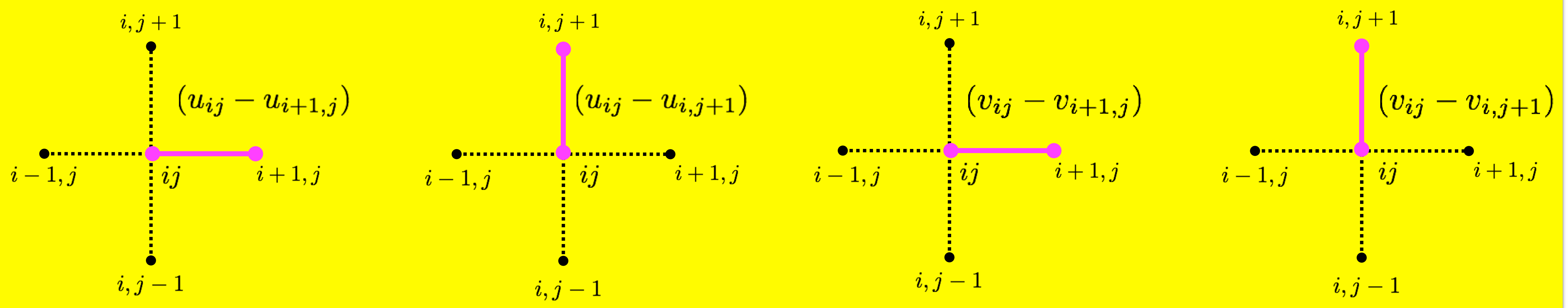
HS optical flow objective function

Brightness constancy

$$E_d(i,j) = \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

Smoothness

$$E_s(i,j) = \frac{1}{4}\left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2 \right]$$



Horn-Schunck
Optical Flow Algorithm

1. Precompute image gradients I_y I_x

2. Precompute temporal gradients I_t

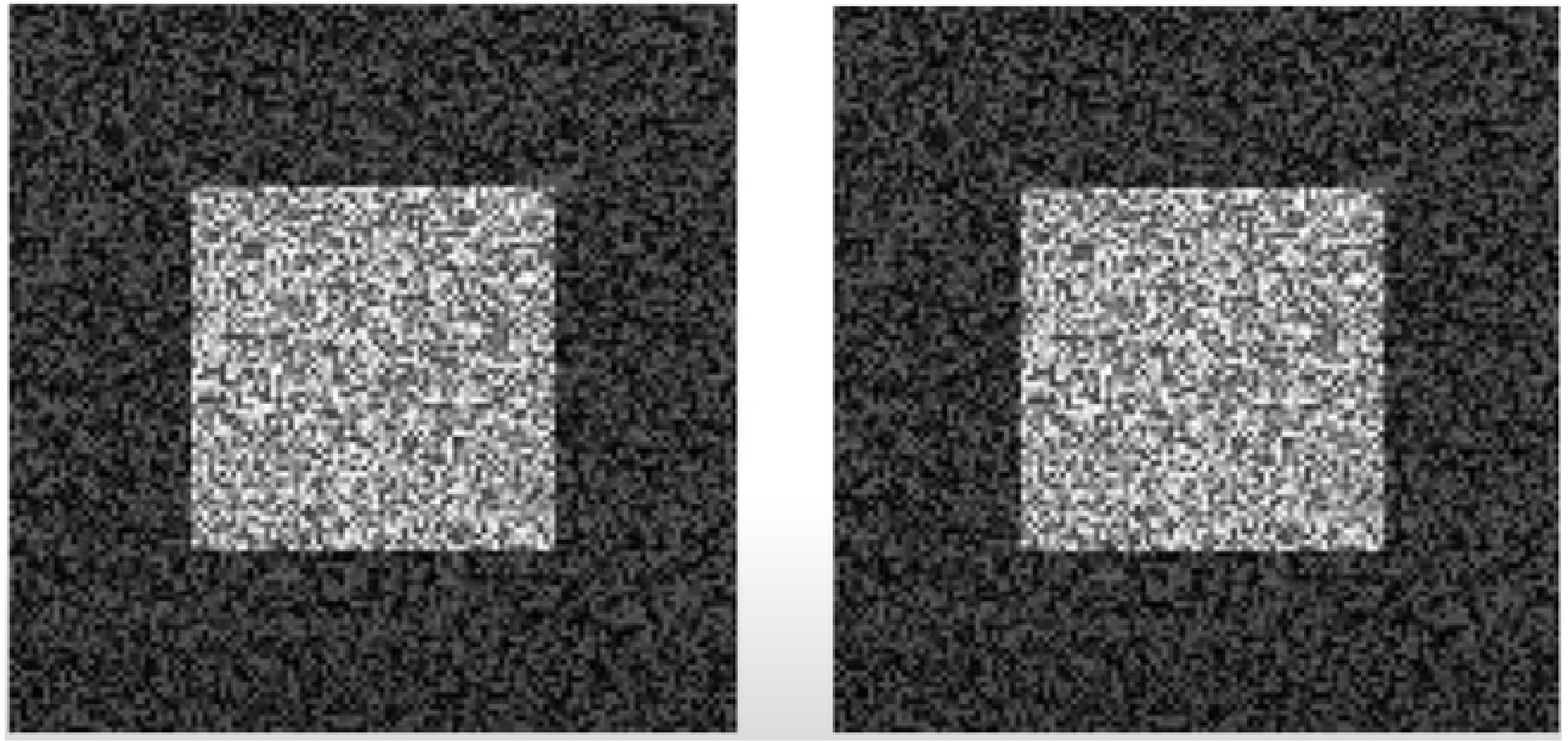
3. Initialize flow field $\boldsymbol{u} = \mathbf{0}$ $\boldsymbol{v} = \mathbf{0}$

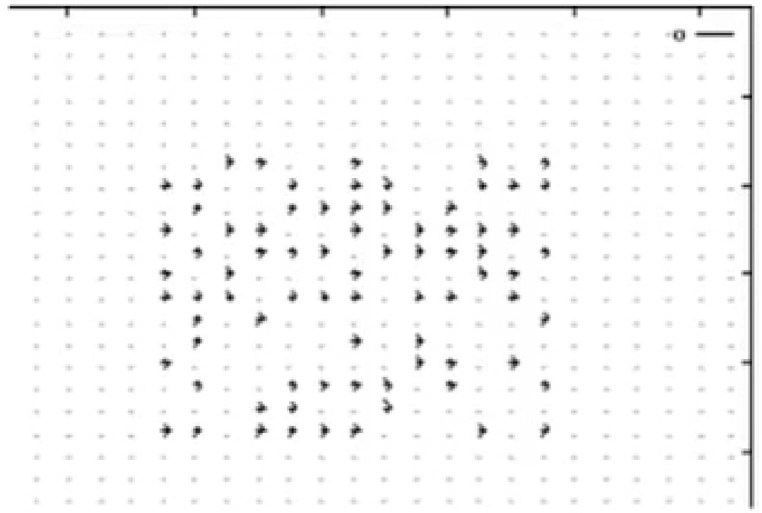
4. While not converged

   Compute flow field updates for each pixel:

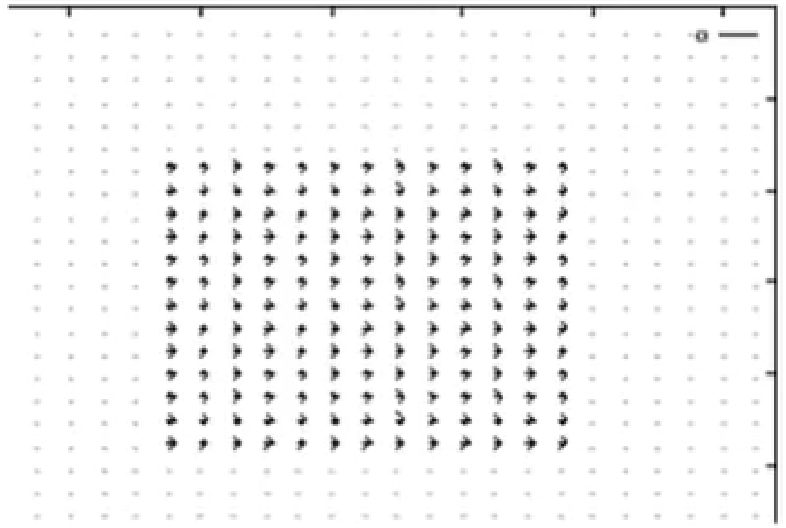
$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_x \qquad \hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_y$$

Horn – Shunck example





One iteration



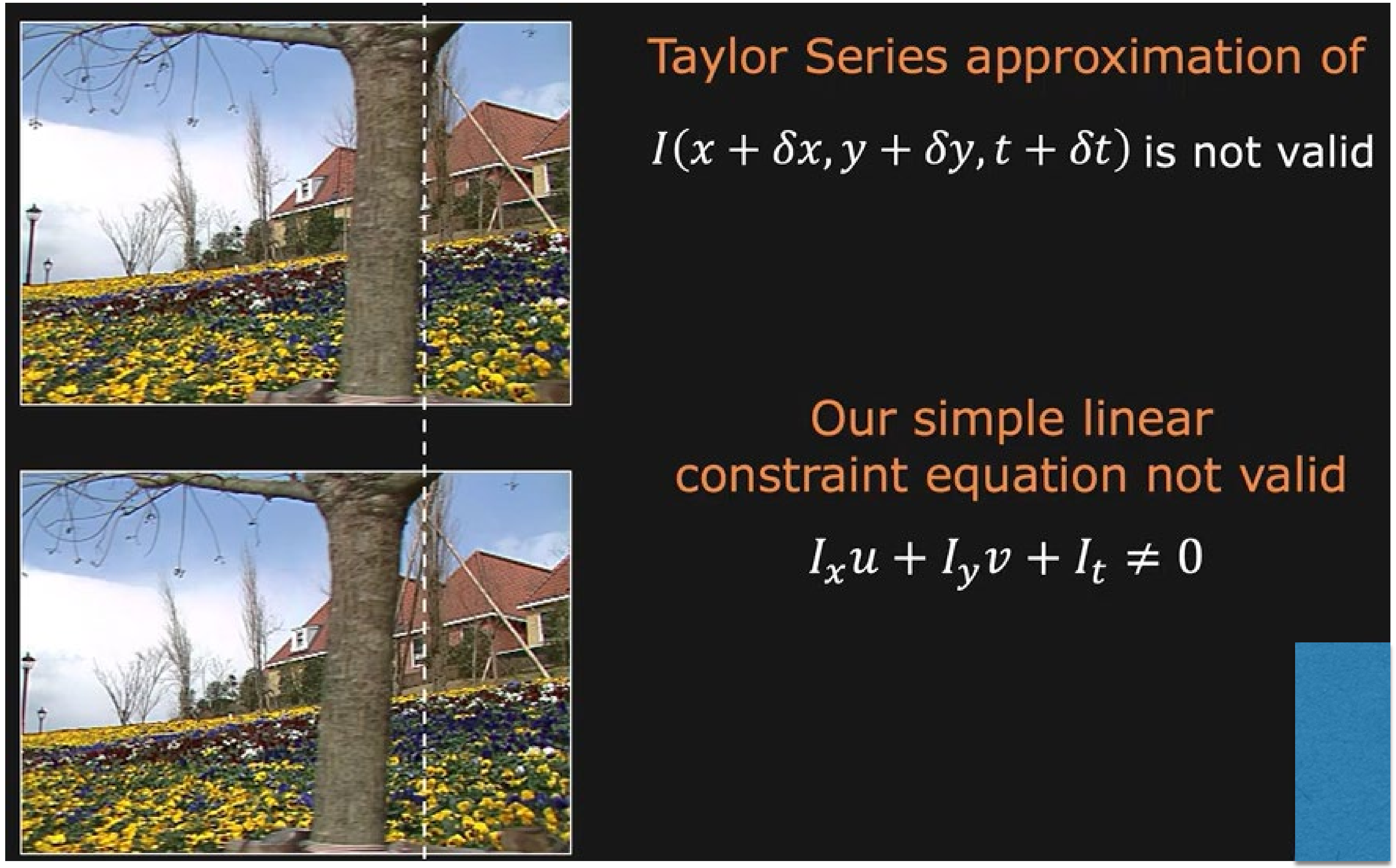
10 iterations

Recap

Key assumptions

- **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

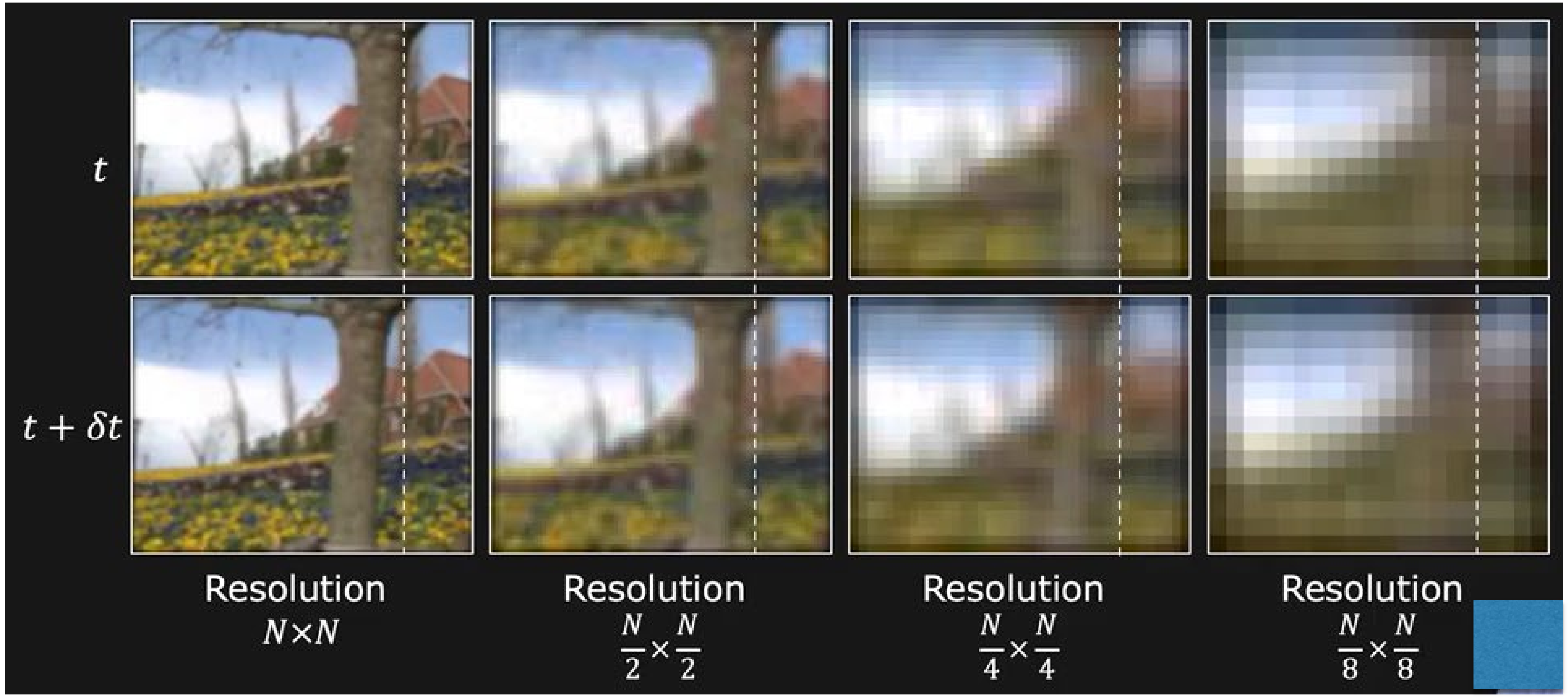


Case of large motion

Is this motion small enough?

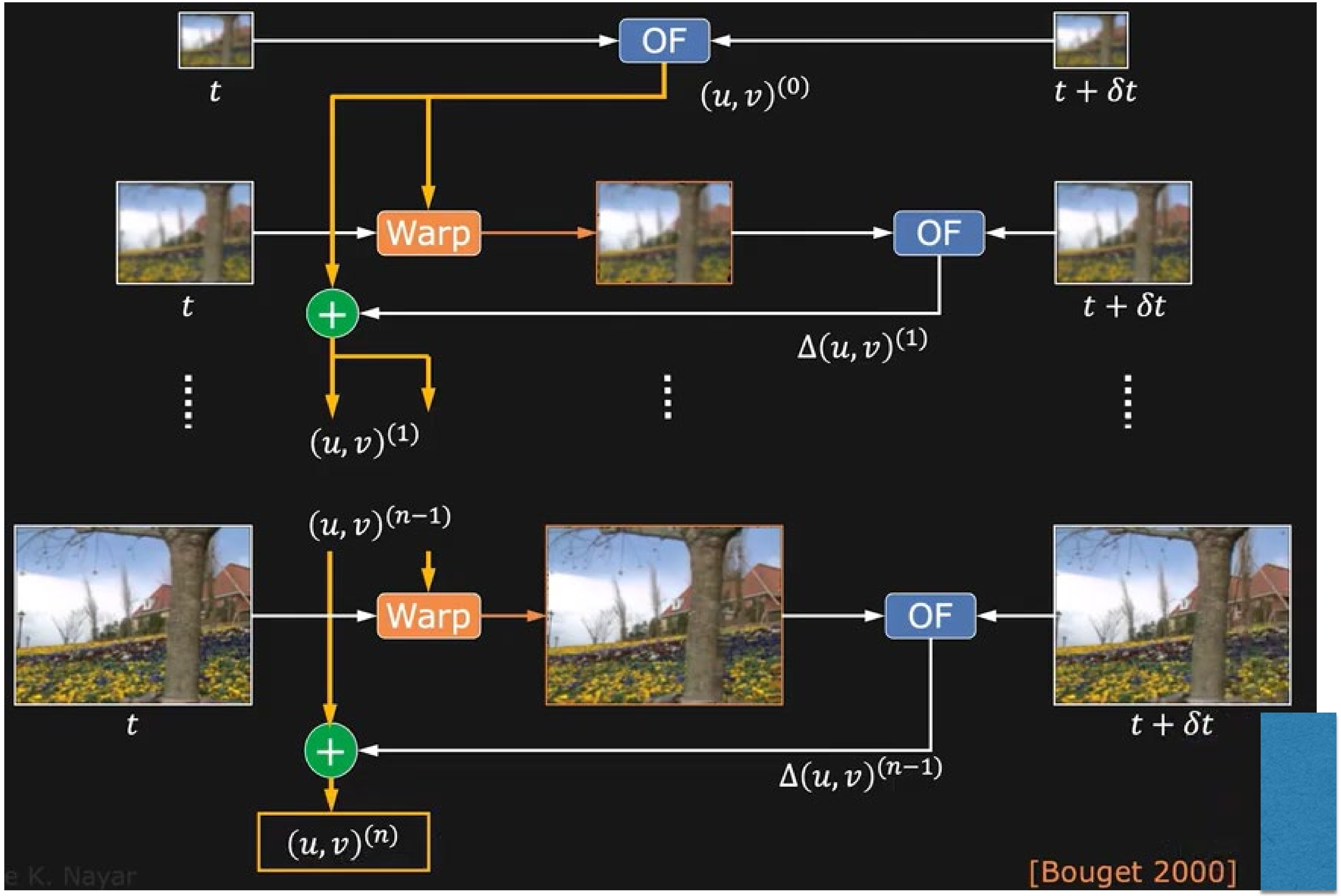
- Probably not: it's much larger than one pixel (2nd order terms dominate)
- How might we solve this problem?

Reduce the resolution

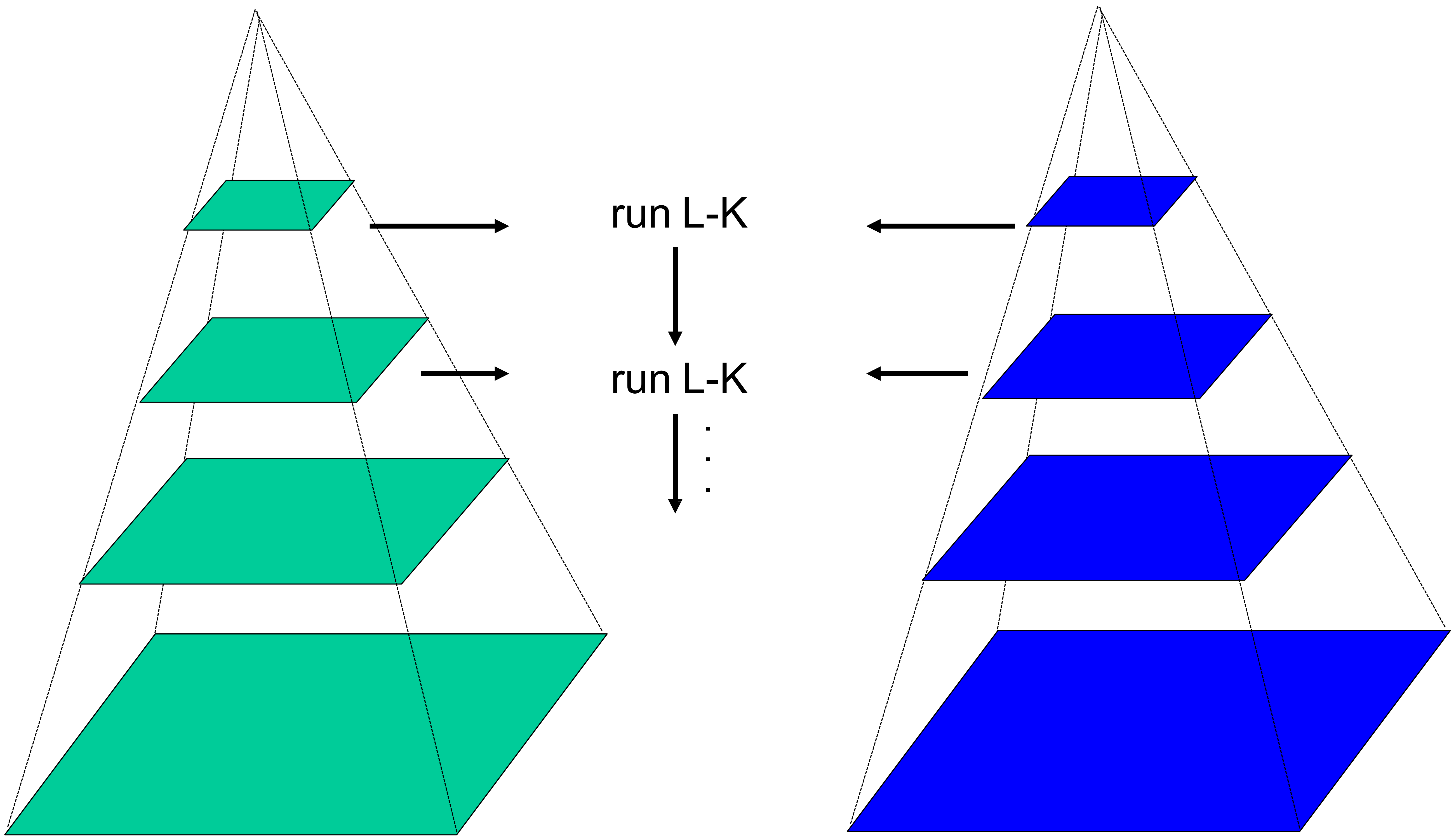


At lowest resolution, motion <=1 pixel

Coarse-to-fine Flow Estimation



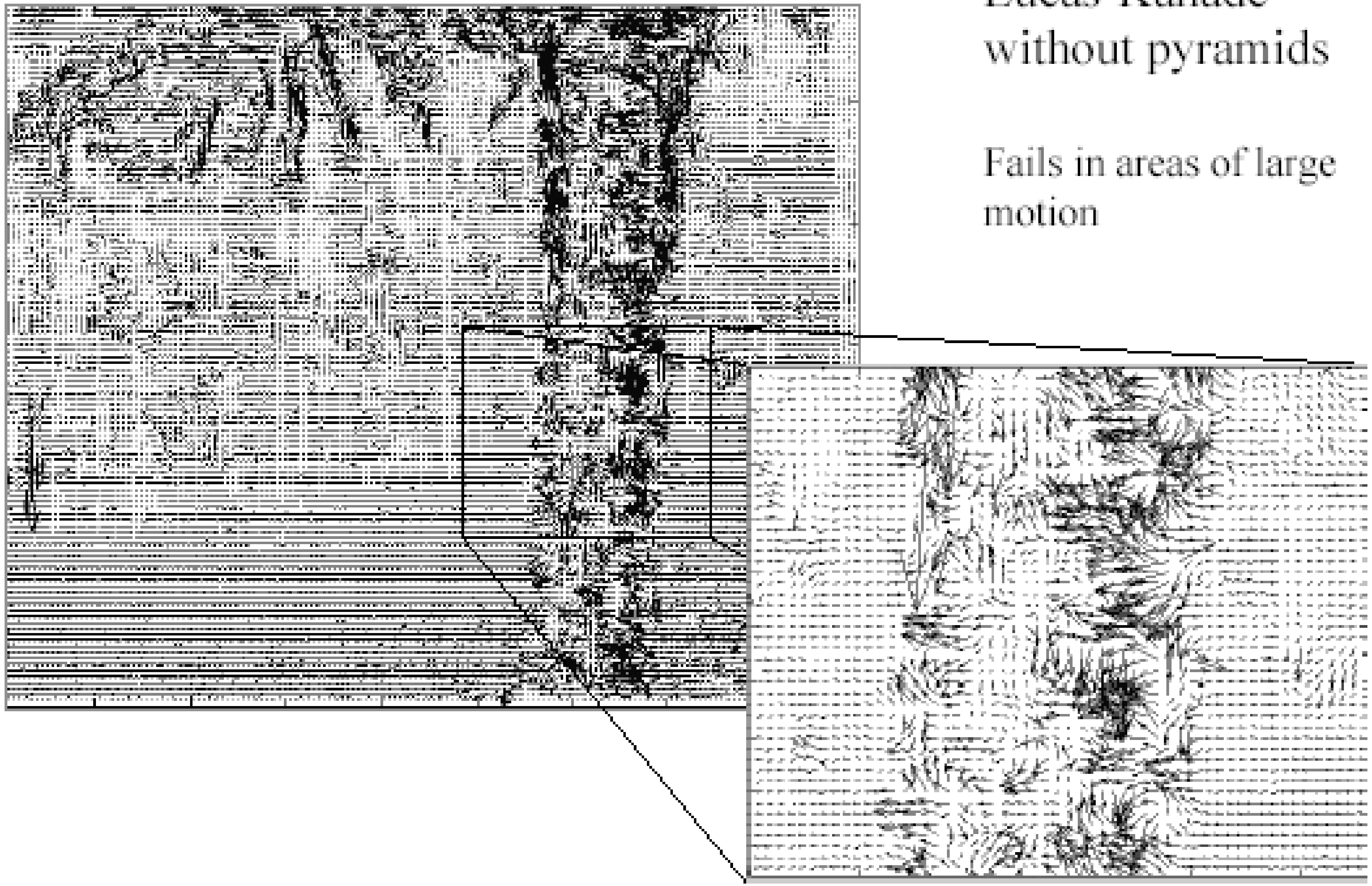
Coarse-to-fine optical flow estimation



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

JY. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature TrackerDescription of the algorithm 2000, Tech. Report: http://robots.stanford.edu/cs223b04/algo_tracking.pdf

Optical Flow Results

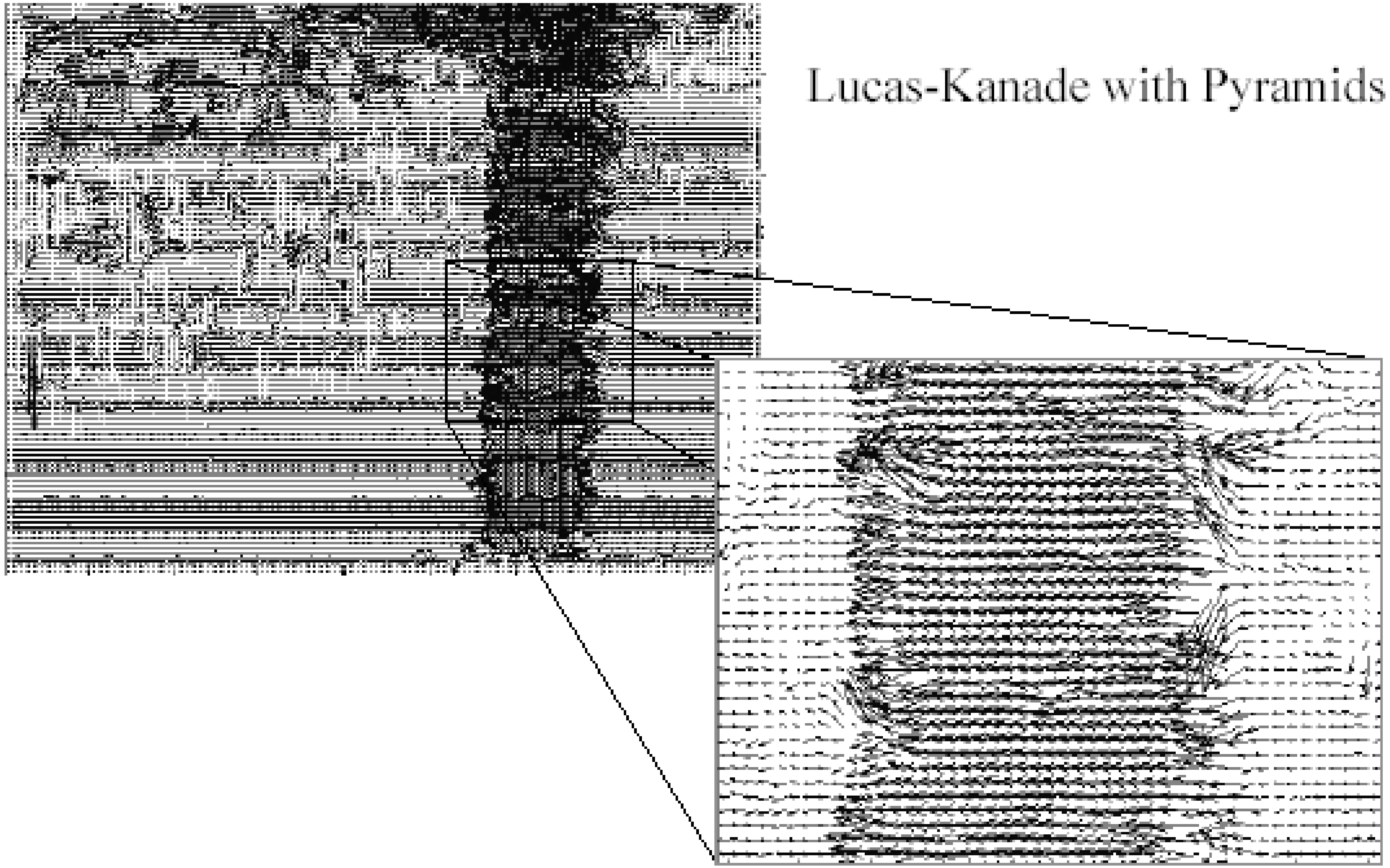


Lucas-Kanade without pyramids

Fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

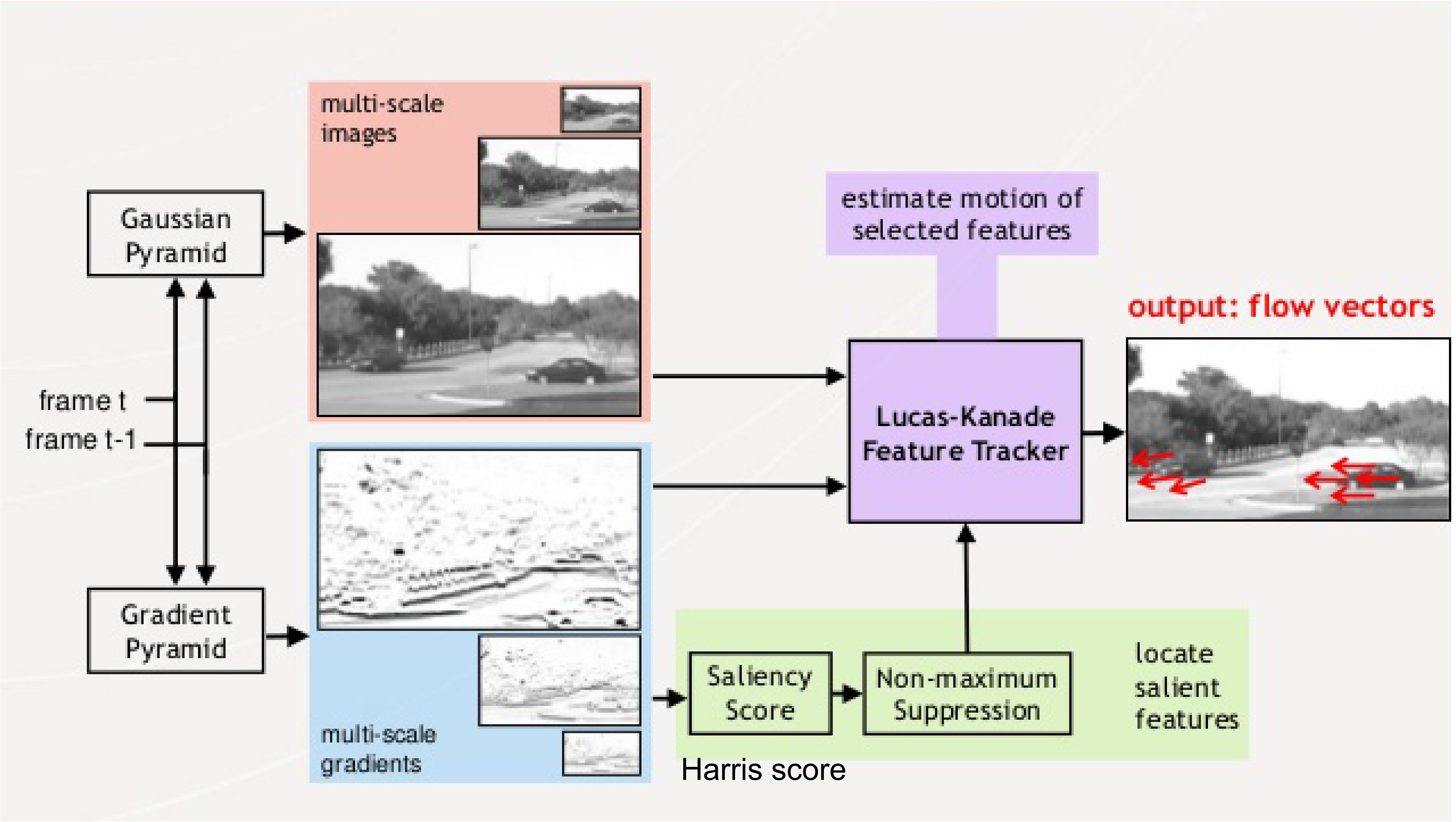
Optical Flow Results



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

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Lucas- Kanade feature tracker



Recap

Key assumptions

- **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

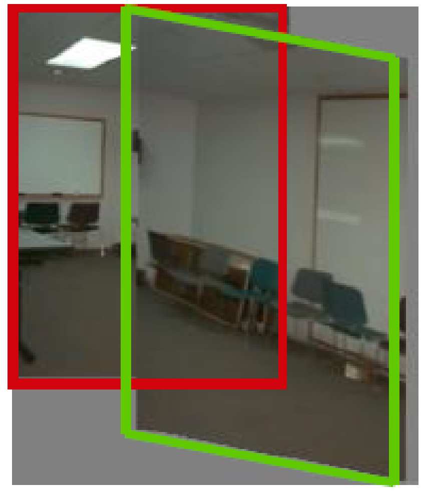
How do we represent the motion in this scene?

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

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

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:

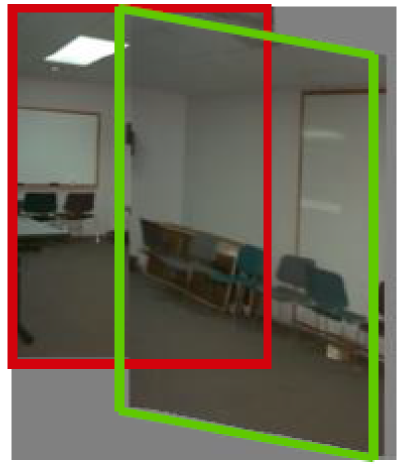
$$\boxed{I_x \cdot u + I_y \cdot v + I_t \approx 0}$$

- An affine model is used to approximate the flow patterns consistent with all types of camera motion
- Affine parameters *a1,..,a6* are calculated by minimizing the least squares error of the motion vectors

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:

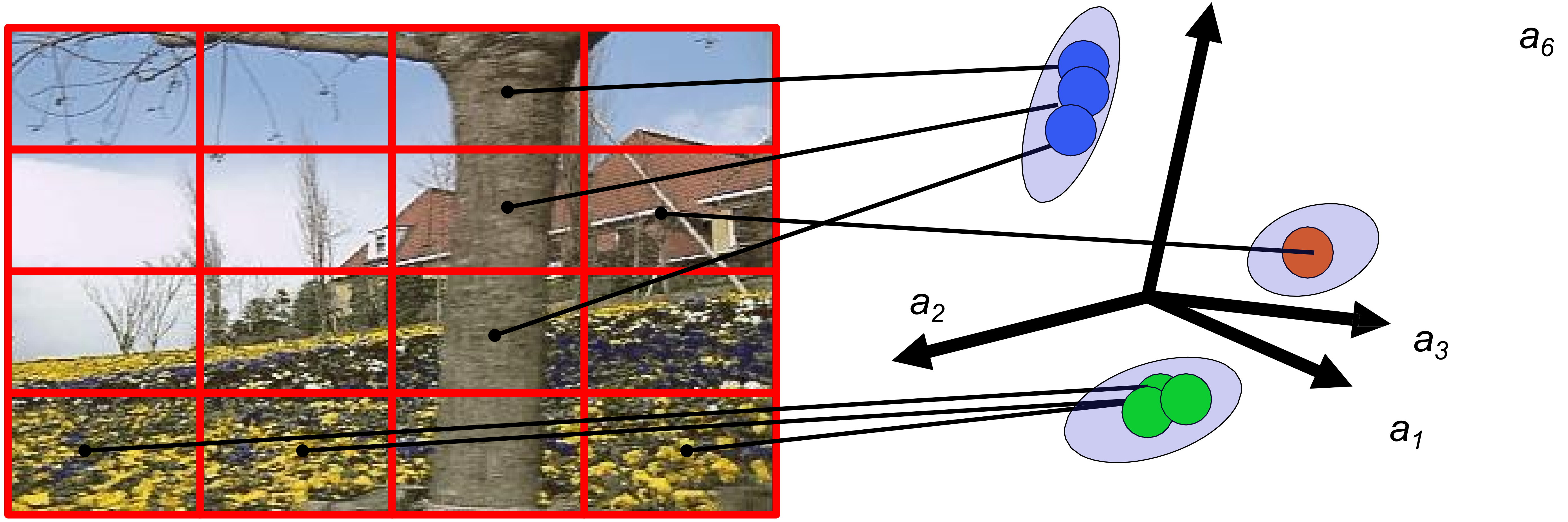
$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

- 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:

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

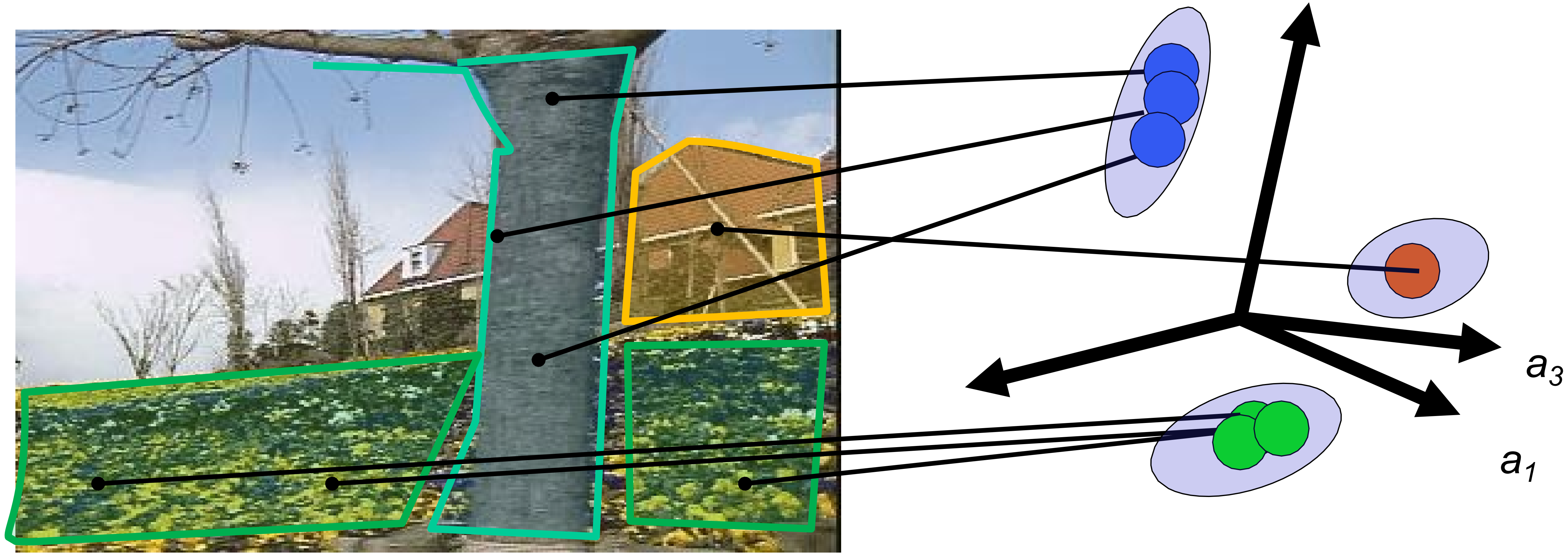
How do we estimate the layers?

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

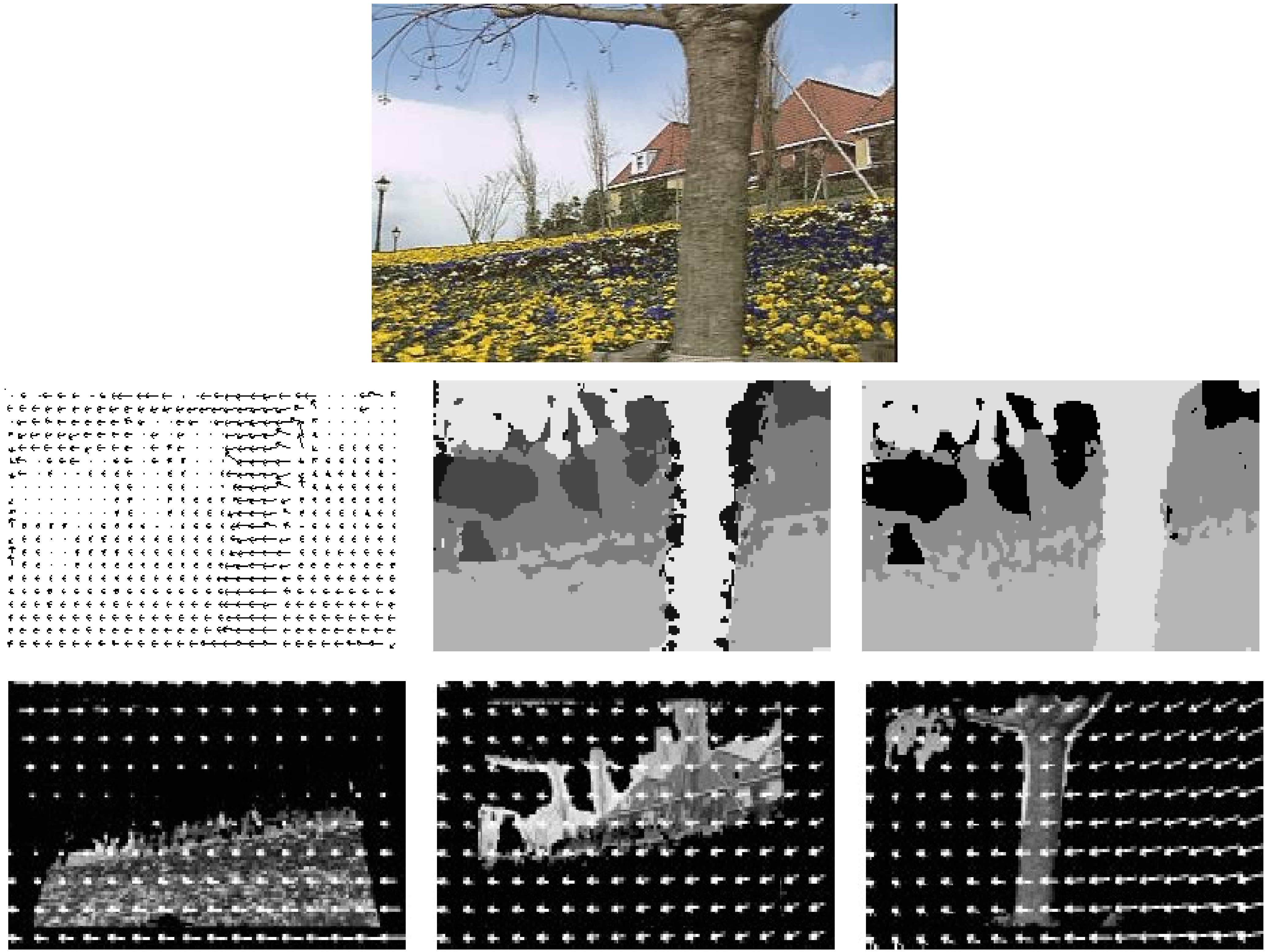


How do we estimate the layers?

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

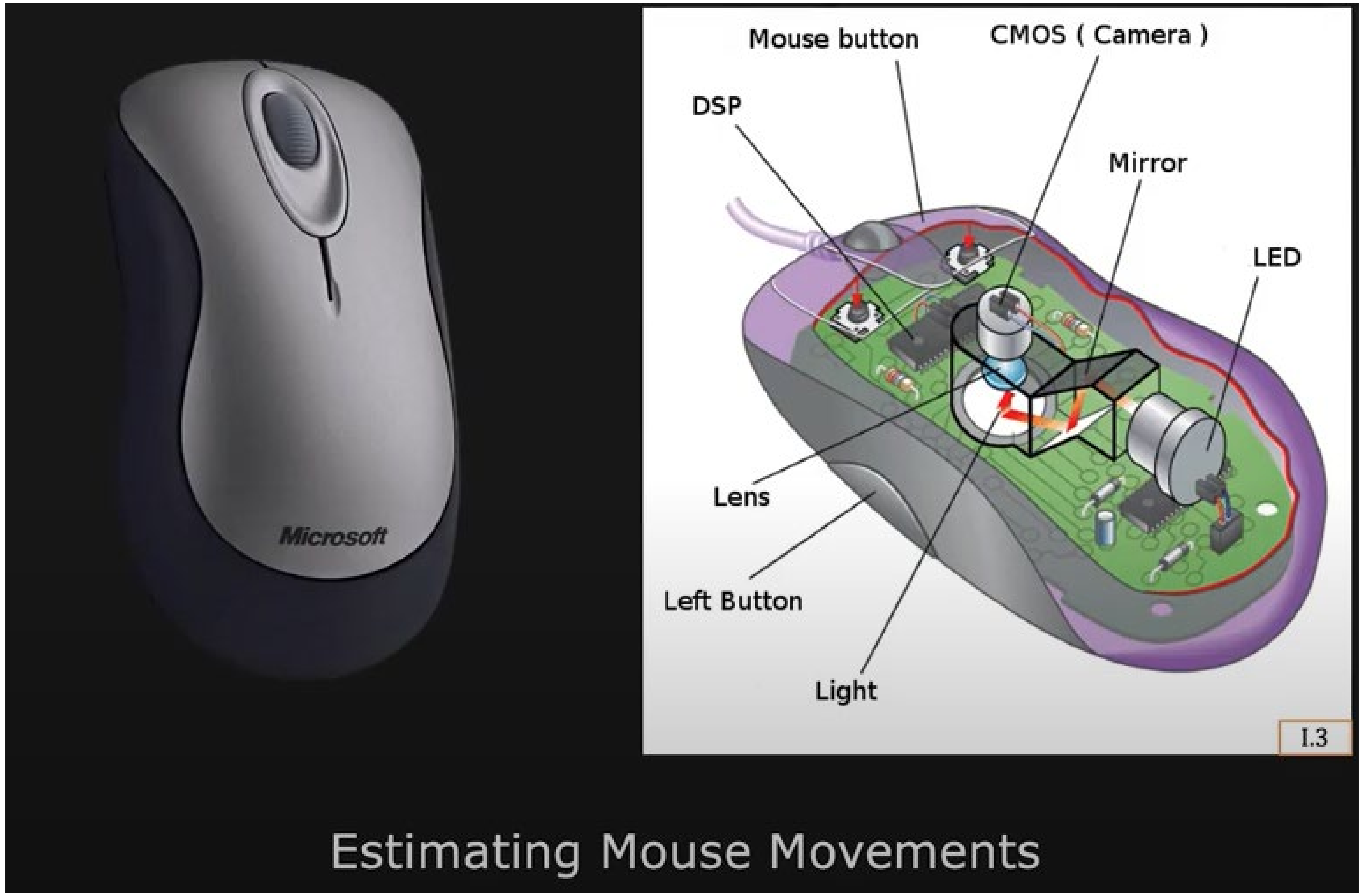


J. Wang and E. Adelson. Layered Representation for Motion Analysis. *CVPR 1993*.

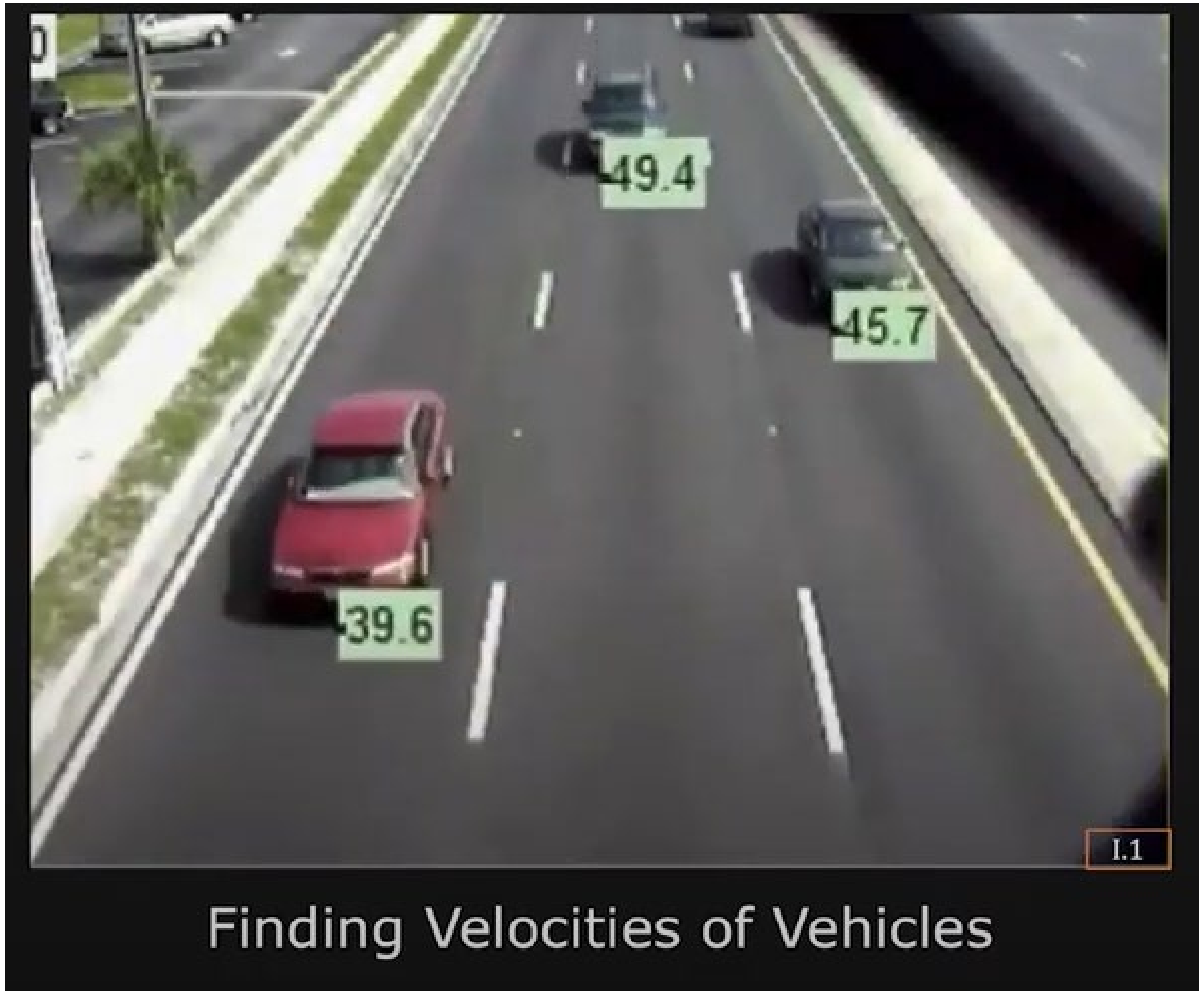
Applications of optical flow

- Video retiming (determine intermediate frames to produce slow motion effects)
- Image stabilization (removing camera shake)
- Face tracking (i.e. eye blinking)
- Games (flow based player interaction)

Optical Mouse



Traffic monitoring



Acknowledgement: some slides and material from Bernt Schiele, Mario Fritz, Michael Black, Bill Freeman, Fei-Fei, Justin Johnson, Serena Yeung, R. Szelisky, Fabio Galasso, Ioannis Gkioulekas, Noah Snavely, Abe Davis, Kris Kitani, Xavier Giró-i-Nieto, Shree Nayar