Computer Vision – TP12
Optical Flow

Miguel Tavares Coimbra

U.PORTO FC

Outline

- Optical Flow Constraint Equation
- Lucas & Kanade Algorithm

Topic: Optical Flow Constraint Equation

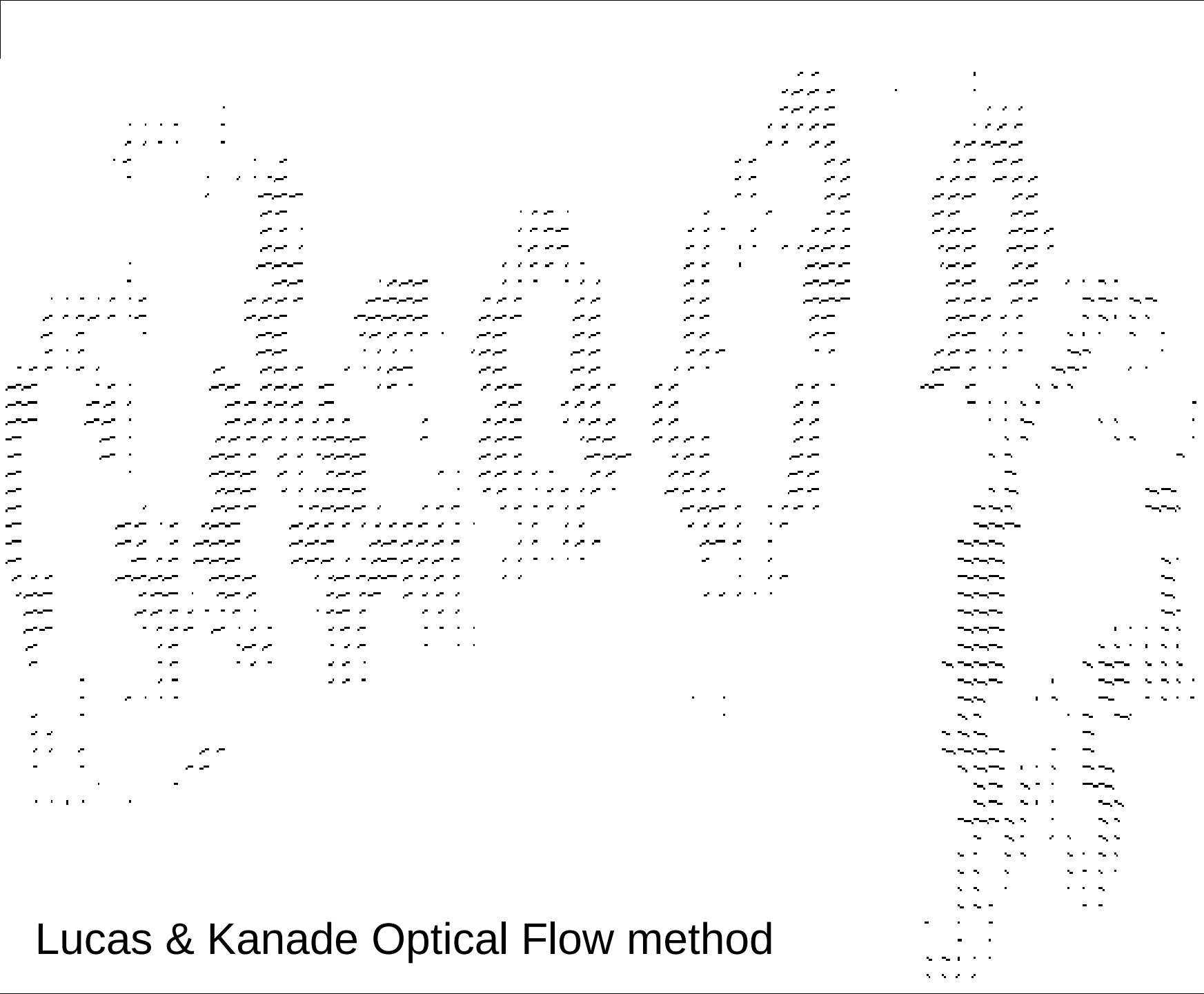
- Optical Flow Constraint Equation
- Lucas & Kanade Algorithm

Optical Flow and Motion

- We are interested in finding the movement of scene objects from time-varying images (videos)

- Lots of uses
  - Track object behavior
  - Correct for camera jitter (stabilization)
  - Align images (mosaics)
  - 3D shape reconstruction
  - Special effects

Where can i find motion?

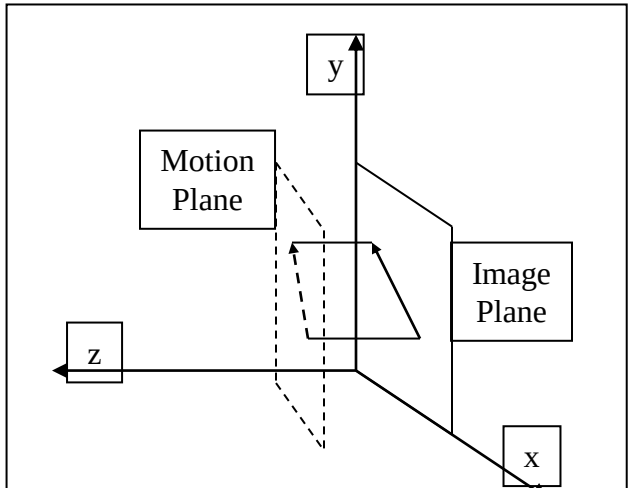


Lucas & Kanade Optical Flow method

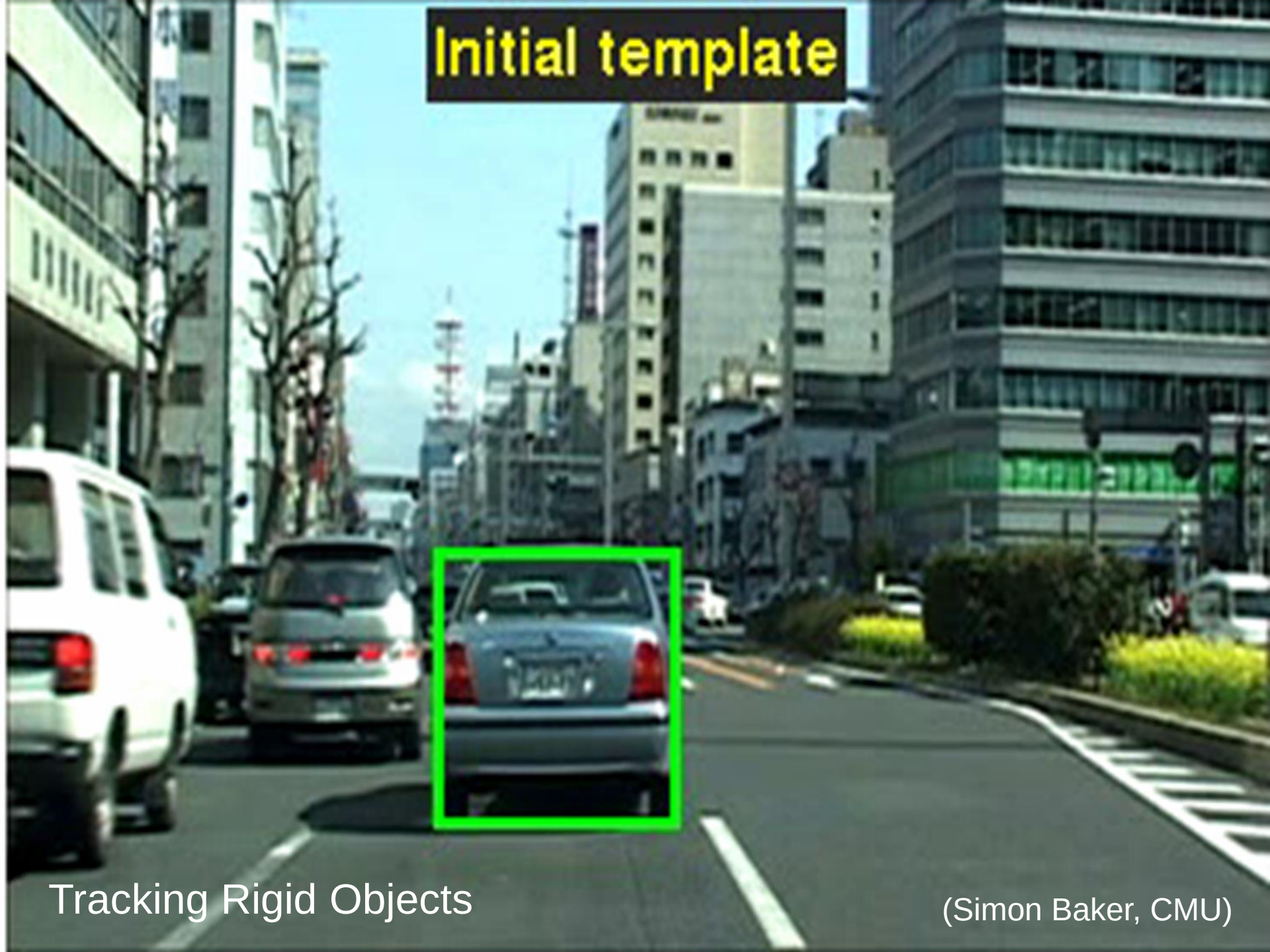
Optical Flow – What is that?

Optical flow is “the distribution of apparent velocities of movement of brightness patterns in an image” – Horn and Schunck 1980

The optical flow field approximates the true motion field which is a “purely geometrical concept..., it is the [2D] projection into the image [plane] of [the sequence’s] 3D motion vectors” – Horn and Schunk 1993



What can i use it for?

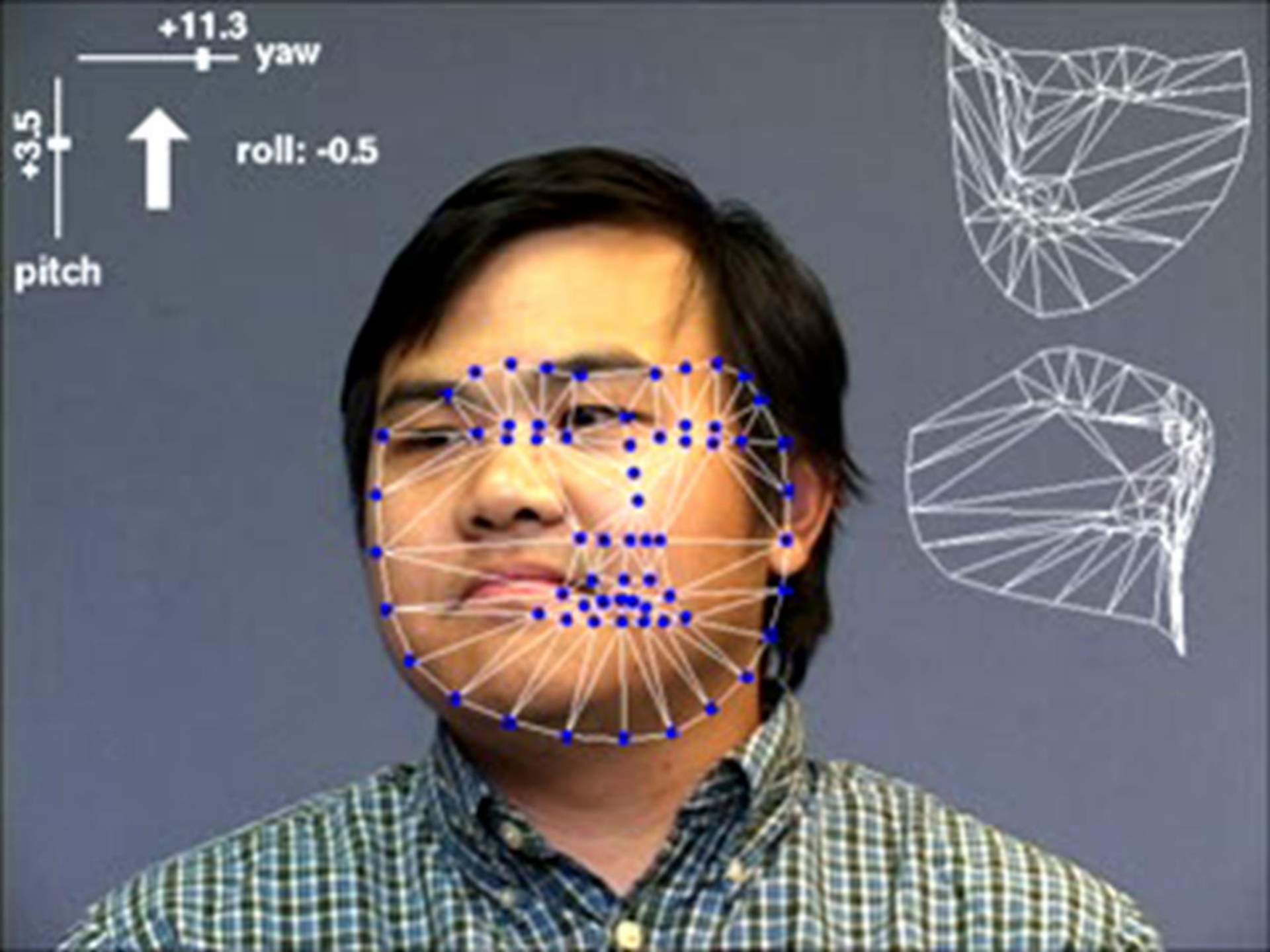


Tracking Rigid Objects

(Simon Baker, CMU)

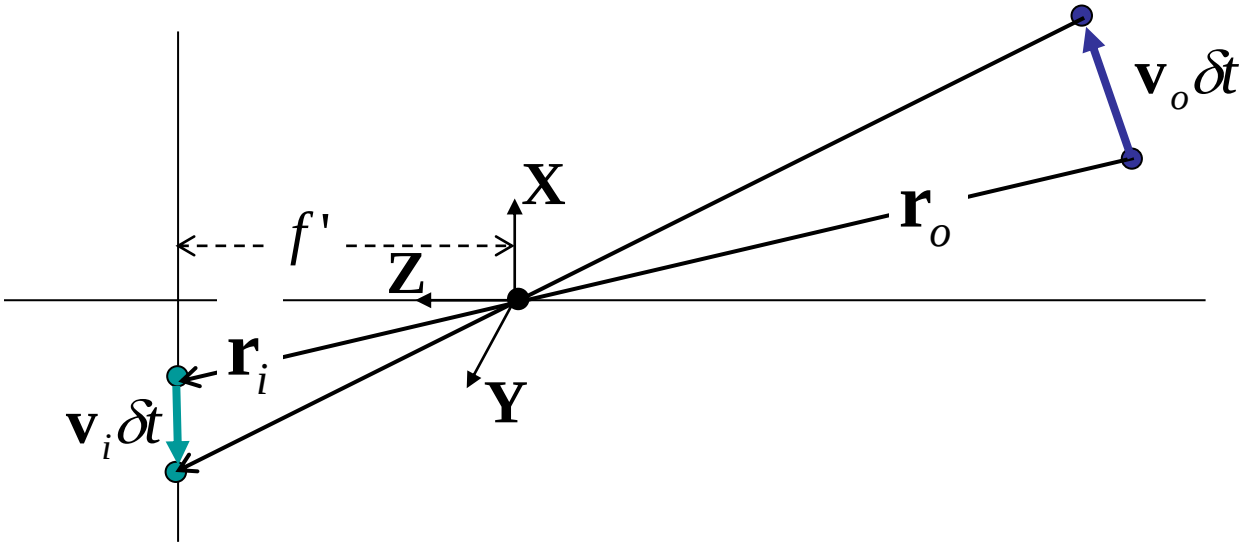
Tracking – Non-rigid Objects

(Comaniciu et al, Siemens)


+11.3
yaw
+3.5
pitch
roll: -0.5

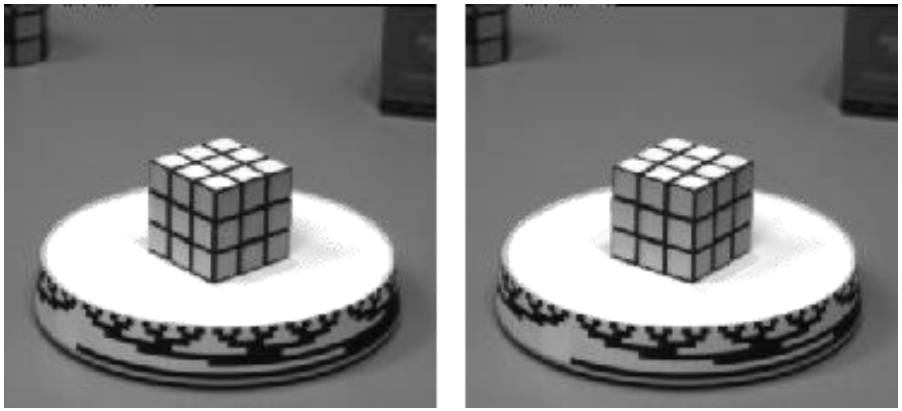
Motion Field

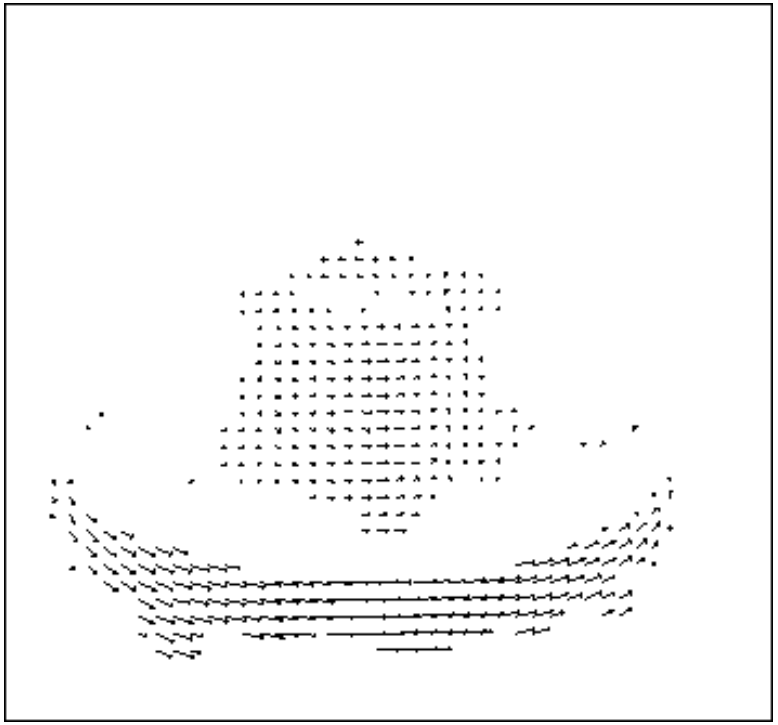
- Image velocity of a point moving in the scene



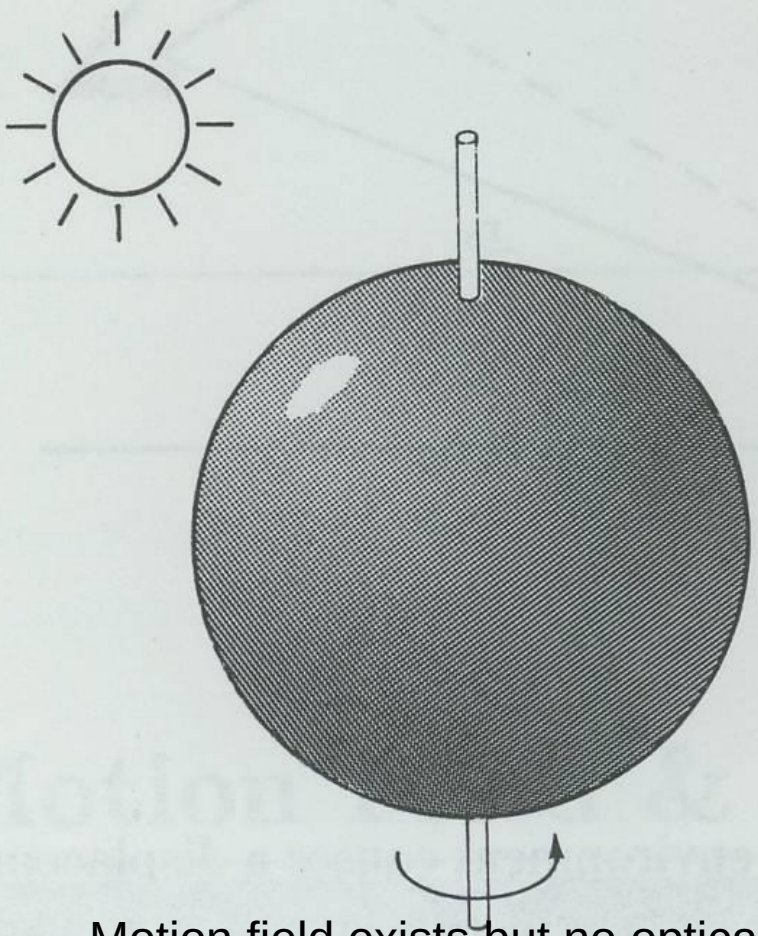
Optical Flow

- Motion of brightness patterns in the image
- **Ideally** Optical flow = Motion field



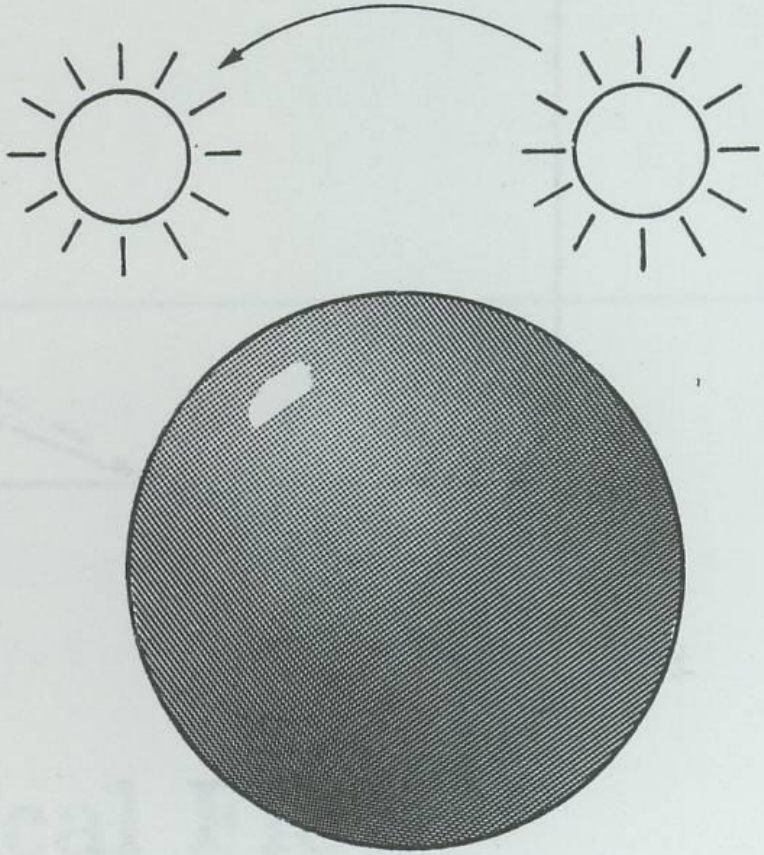


Optical Flow ≠ Motion Field



Motion field exists but no optical flow

(a)

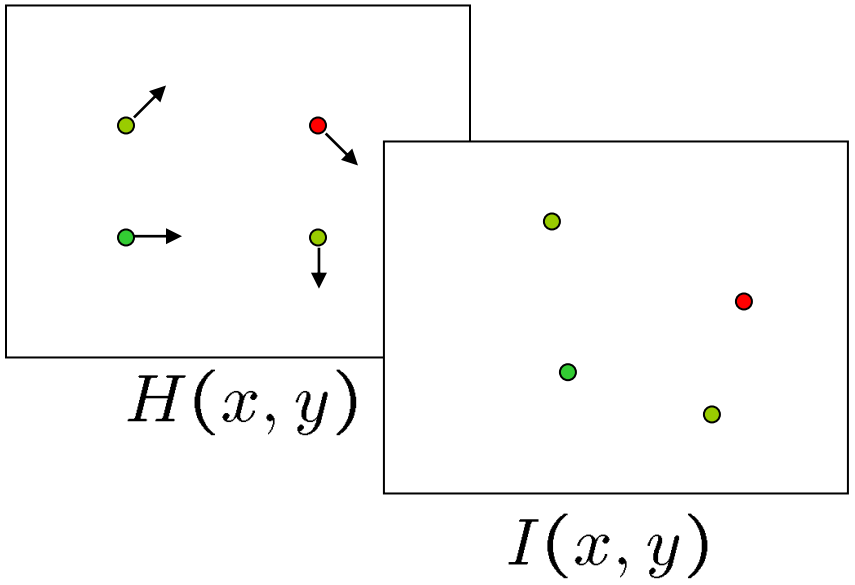


No motion field but shading changes

(b)

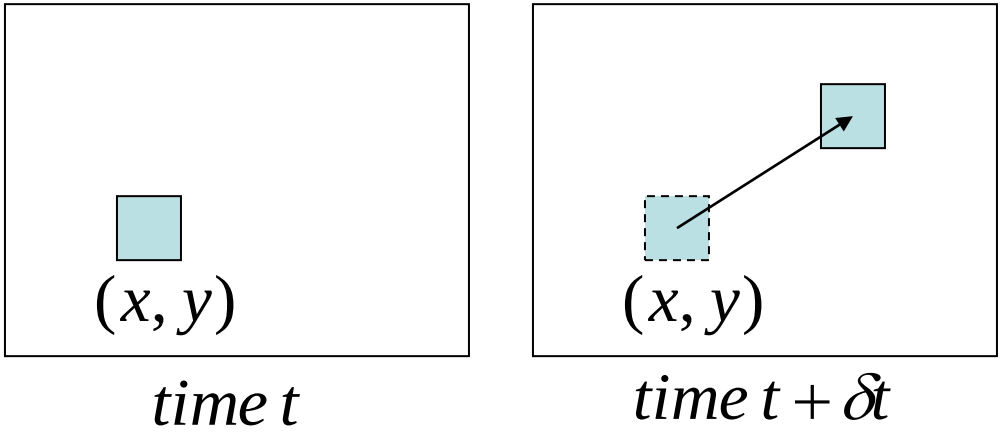
Problem Definition: Optical Flow

- How to estimate pixel motion from image H to image I?
  - Find pixel correspondences
    - Given a pixel in H, look for nearby pixels of the same color in I



- Key assumptions
  - **color constancy:** a point in H looks "the same" in image I
    - For grayscale images, this is **brightness constancy**
  - **small motion**: points do not move very far

Optical Flow Constraint Equation



$time\ t$ $time\ t+\delta t$

Optical Flow: Velocities (u, v)

Displacement:

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

– Assume brightness of patch remains same in both images:

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

– Assume small motion: (Taylor expansion of LHS up to first order)

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

Optical Flow Constraint Equation

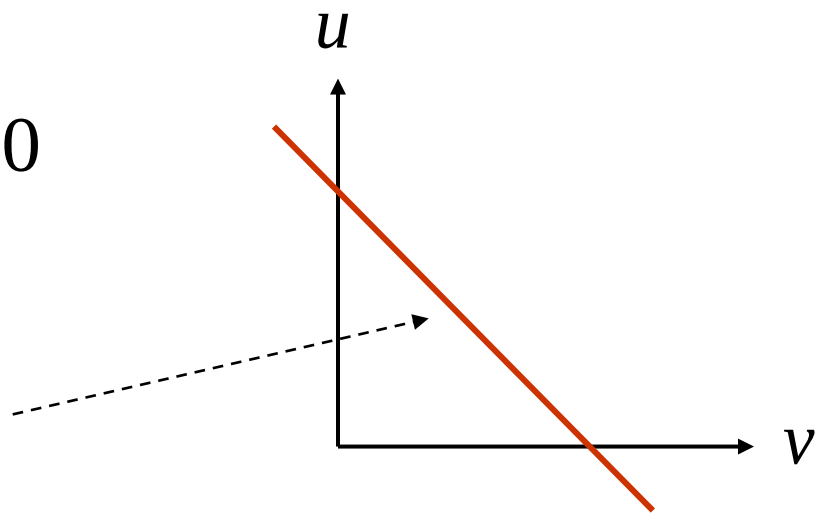
$$\delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} = 0$$

Divide by δt and take the limit $\delta t \to 0$

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

Constraint Equation

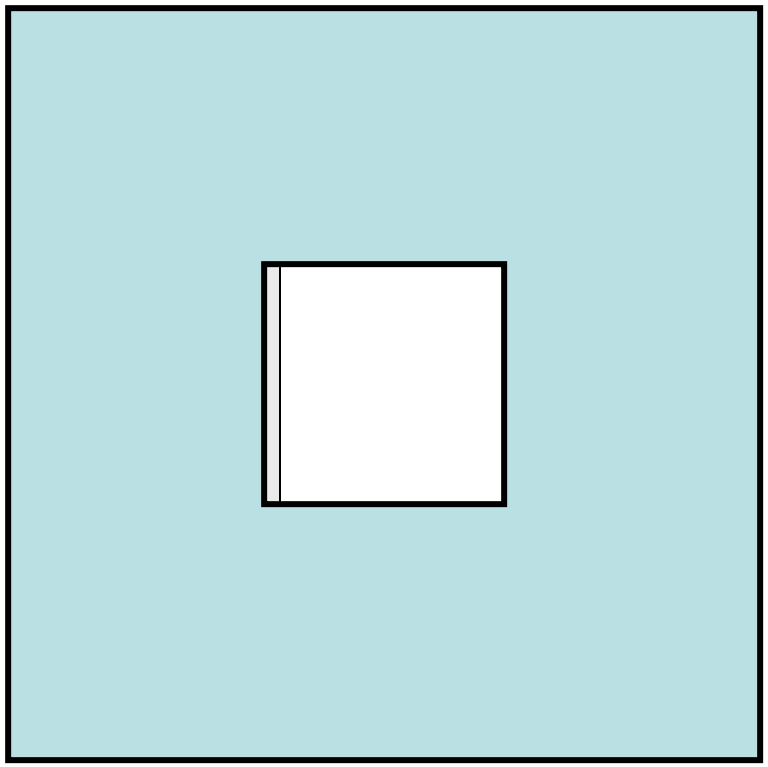
$$E_x\, u + E_y\, v + E_t = 0$$



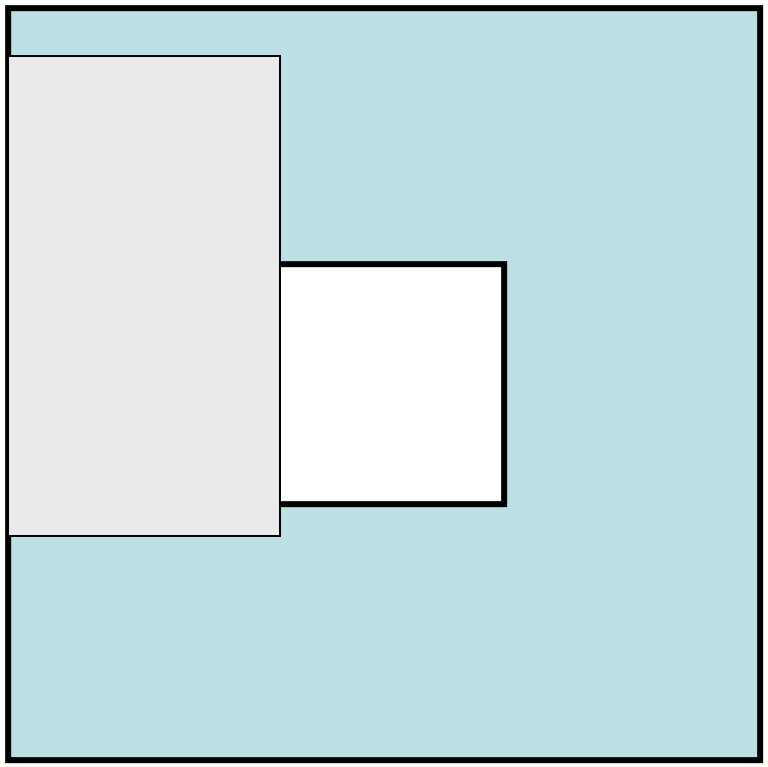
Optical Flow Constraint

- Intuitively, what does this constraint mean?
  - The component of the flow in the gradient direction is determined
  - The component of the flow parallel to an edge is unknown

How does this show up visually? Known as the “Aperture Problem”



Aperture Problem Exposed



Motion along just
an edge is ambiguous

Computing Optical Flow

- Formulate Error in Optical Flow Constraint:

$$e_c = \iint_{image} (E_x u + E_y v + E_t)^2 \, dx \, dy$$

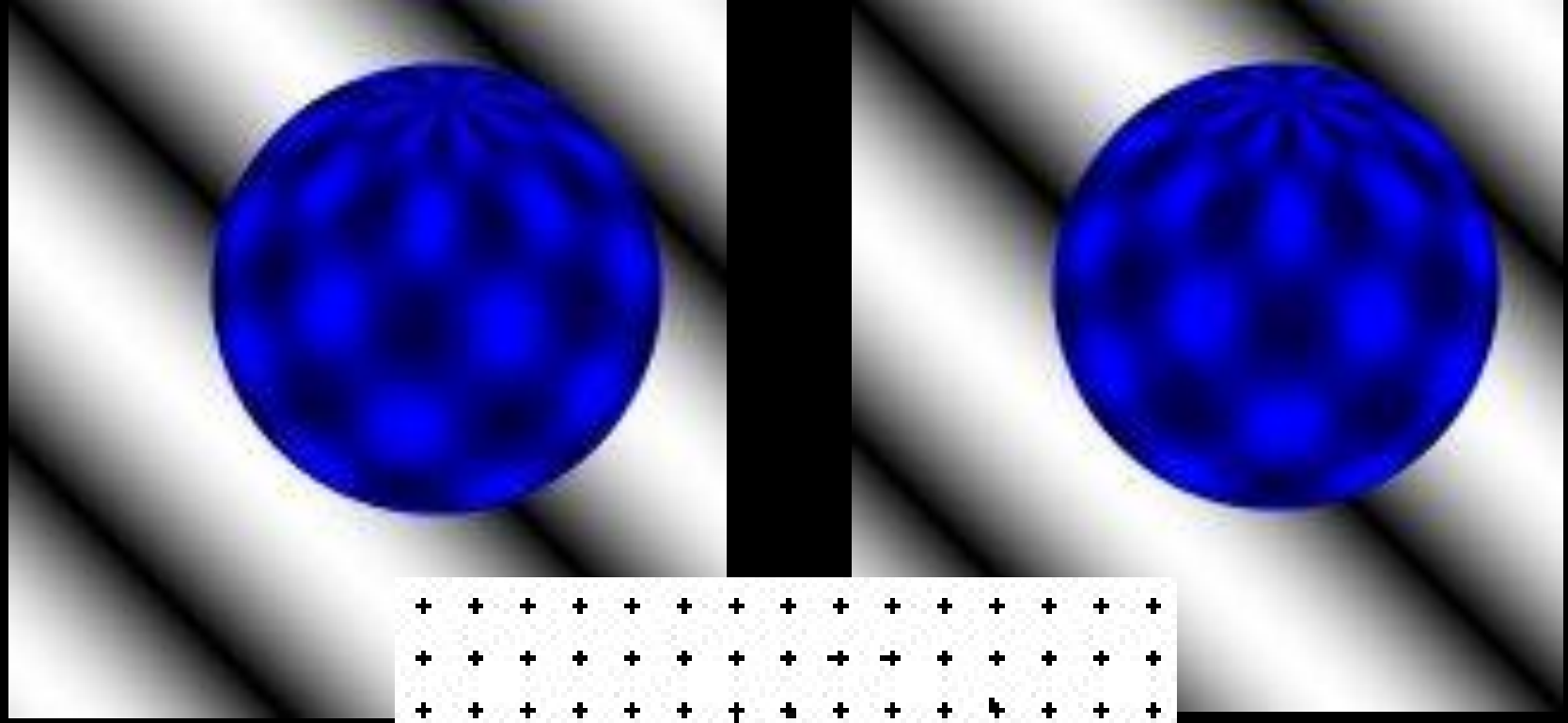
- We need additional constraints!

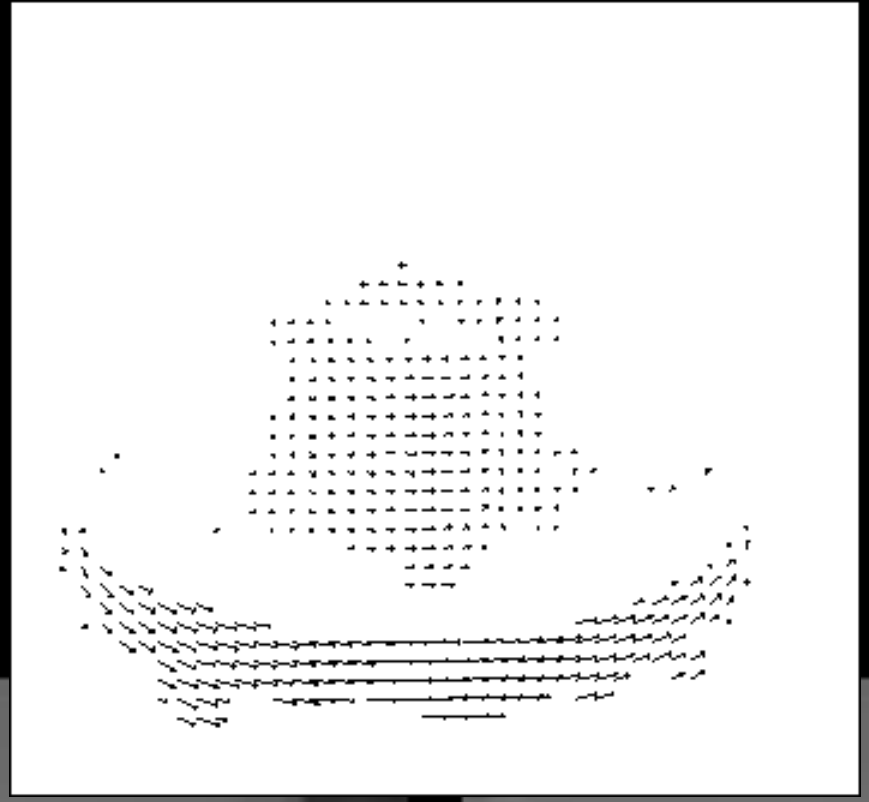
- Smoothness Constraint (as in shape from shading and stereo):
  Usually motion field varies smoothly in the image.
  So, penalize departure from smoothness:

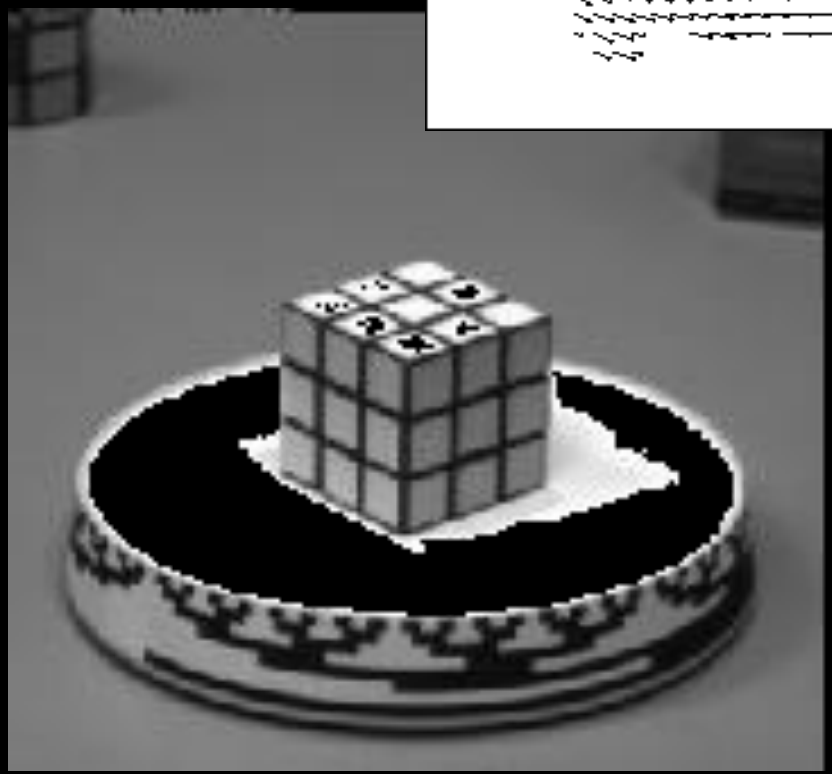
$$e_s = \iint_{image} (u_x^2 + u_y^2) + (v_x^2 + v_y^2) \; dx \, dy$$

weighting factor

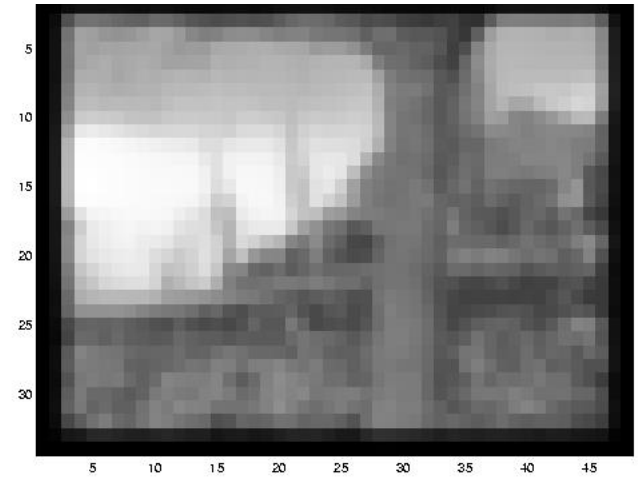
- Find (u,v) at each image point that MINIMIZES: $e = e_s + \lambda e_c$

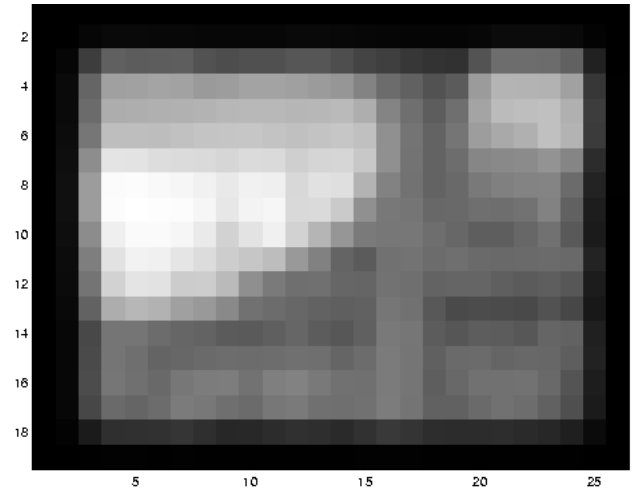




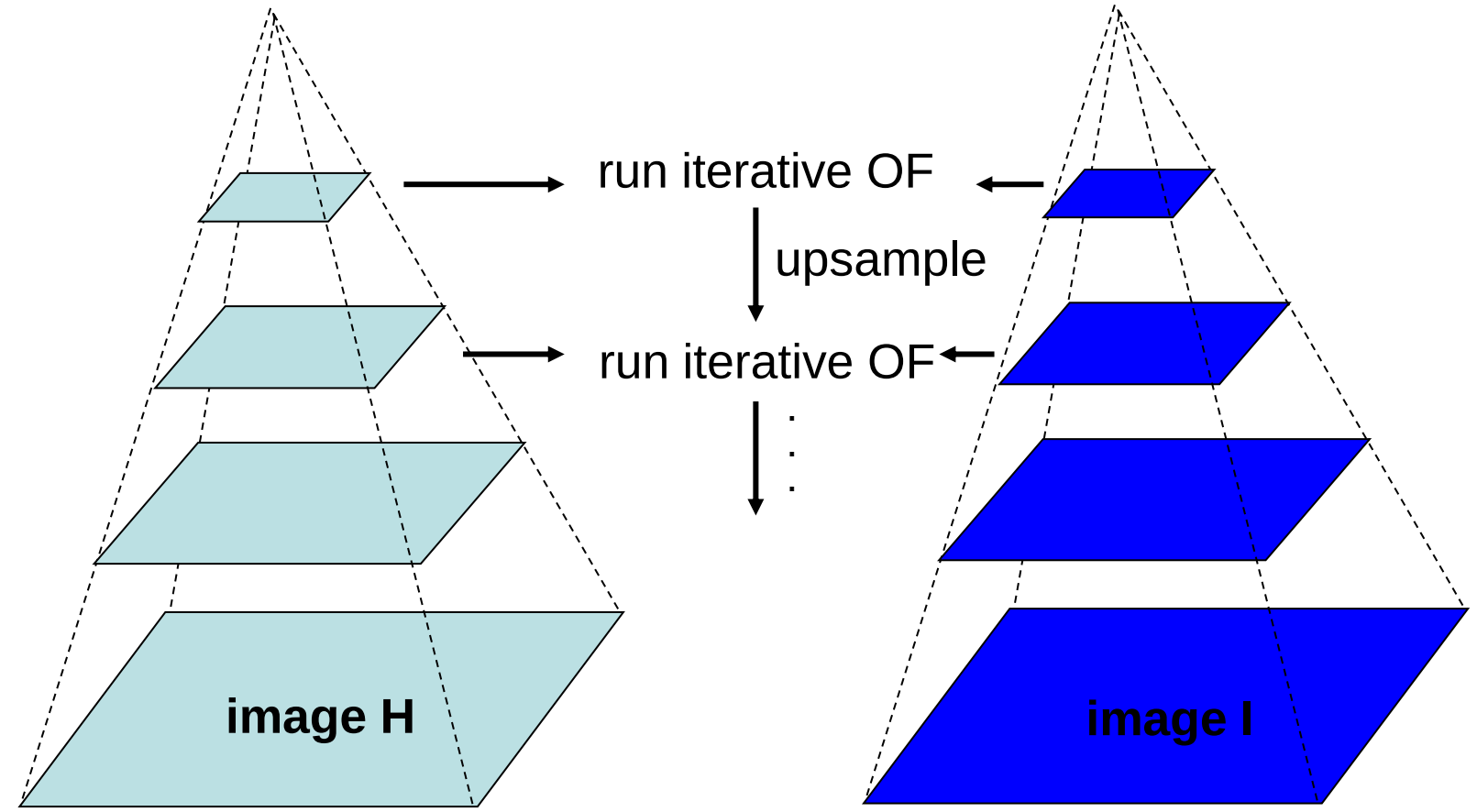


Small Motion Assumption





Coarse-to-fine Optical Flow Estimation



Gaussian pyramid of image *H*

Gaussian pyramid of image *I*

Types of OF methods

- Differential
  - Horn and Schunck [HS80], Lucas Kanade [LK81], Nagel [83].
- Region-based matching
  - Anandan [Anan87], Singh [Singh90], Digital video encoding standards.
- Energy-based
  - Heeger [Heeg87]
- Phase-based
  - Fleet and Jepson [FJ90]

Topic: Lucas & Kanade Algorithm

- Optical Flow Constraint Equation
- Lucas & Kanade Algorithm

Lucas & Kanade Method

- How to get more equations for a pixel?
  - Basic idea: impose additional constraints
    - most common is to assume that the flow field is smooth locally
    - one method: pretend the pixel's neighbors have the same (u,v)
      - If we use a 5x5 window, that gives us 25 equations per pixel!

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

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix}}_{A\ 25\times2} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{d\ 2\times1} = -\underbrace{\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}}_{b\ 25\times1}$$

Lukas-Kanade flow

Problem: we have more equations than unknowns

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } ||Ad - b||^2$$

minimum least squares solution given by solution (in d) of:

$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

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

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

- The summations are over all pixels in the K x K window
- This technique was first proposed by Lukas & Kanade (1981)

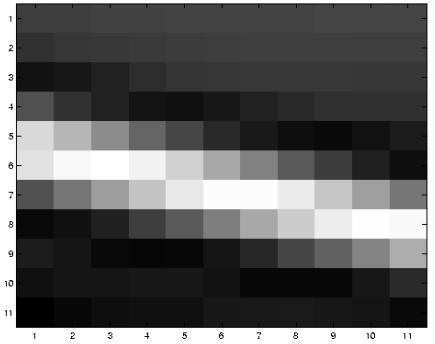
Eigenvectors of A^TA

- Eigenvectors give us the axes of higher intensity variability (gradient)

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

- Eigenvalues give us a measure of gradient strength associated with each eigenvector direction

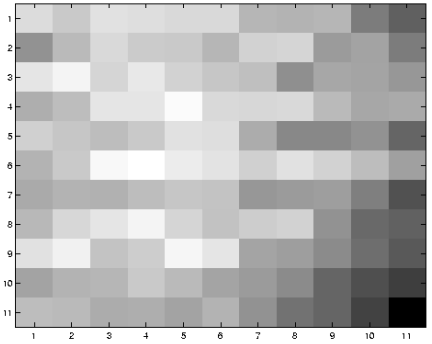
Edges



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

–large λ_1, small λ_2

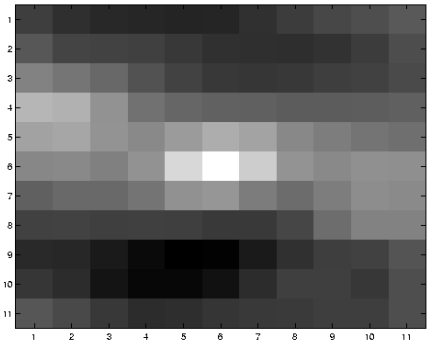
Low texture region



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

–small λ_1, small λ_2

High textured region



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

–large λ_1, large λ_2

Sparse Motion Field

- We are only confident in motion vectors of areas with two strong eigenvectors.
  - Optical flow
- Not so confident when we have one or zero strong eigenvectors.
  - Normal flow (apperture problem)
  - Unknown flow (blank-wall problem)

Summing all up

- Optical flow:
  - Algorithms try to approximate the true motion field of the image plane
  - The Optical Flow Constraint Equation needs an additional constraint (e.g. smoothness, constant local flow)
  - The Lucas Kanade method is the most popular Optical Flow Algorithm
- What applications is this useful for?
- What about block matching?

Resources

- Szeliski, “Computer Vision: Algorithms and Applications”, Springer, 2011
  - Chapter 8 – “Dense Motion Estimation”