COLLEGE OF ENGINEERING
ELECTRICAL ENGINEERING & COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

Motion and Optical Flow

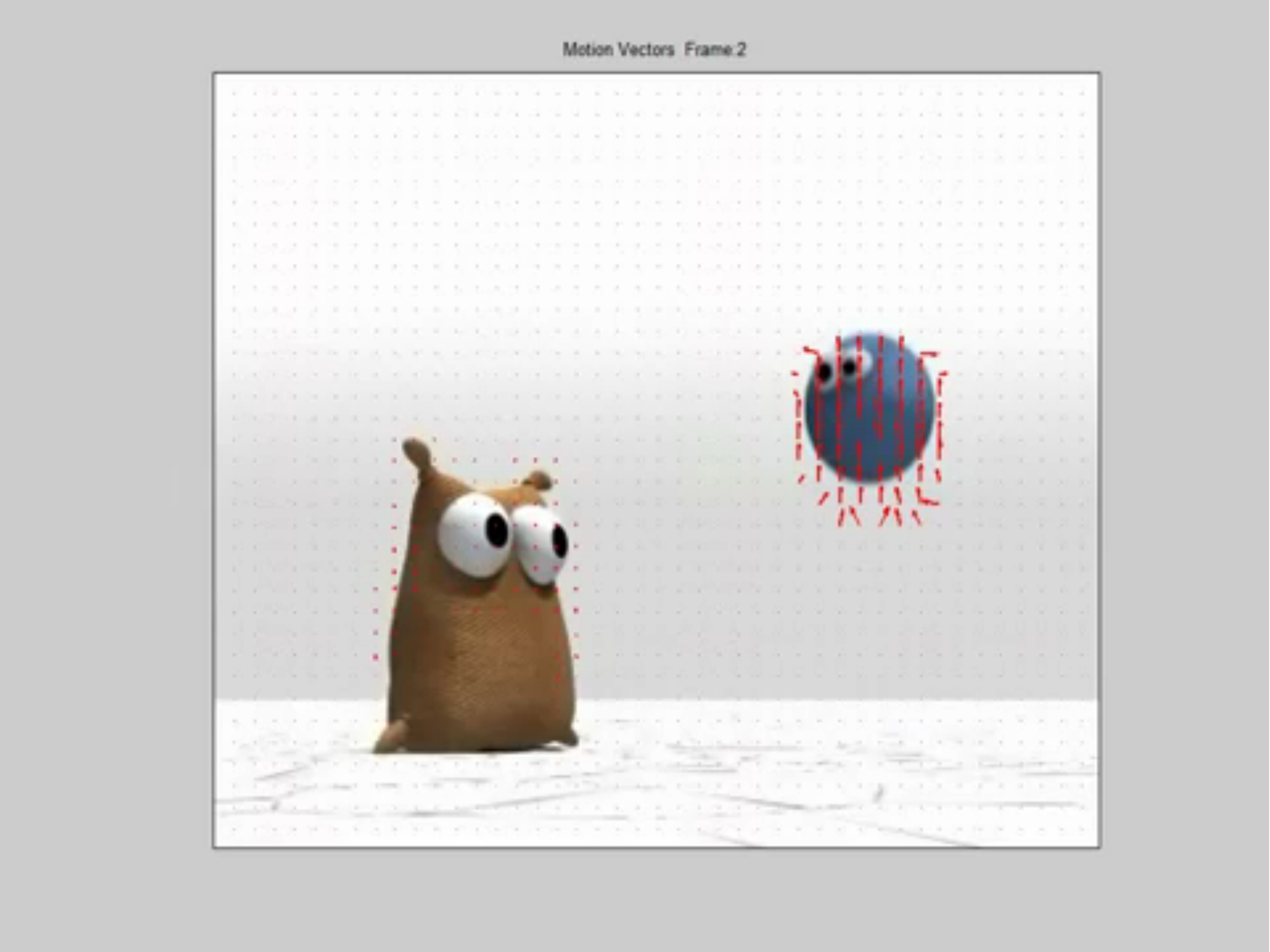
EECS 598-08 Fall 2014
Foundations of Computer Vision

Instructor: Jason Corso (jjcorso)
web.eecs.umich.edu/~jjcorso/t/598F14

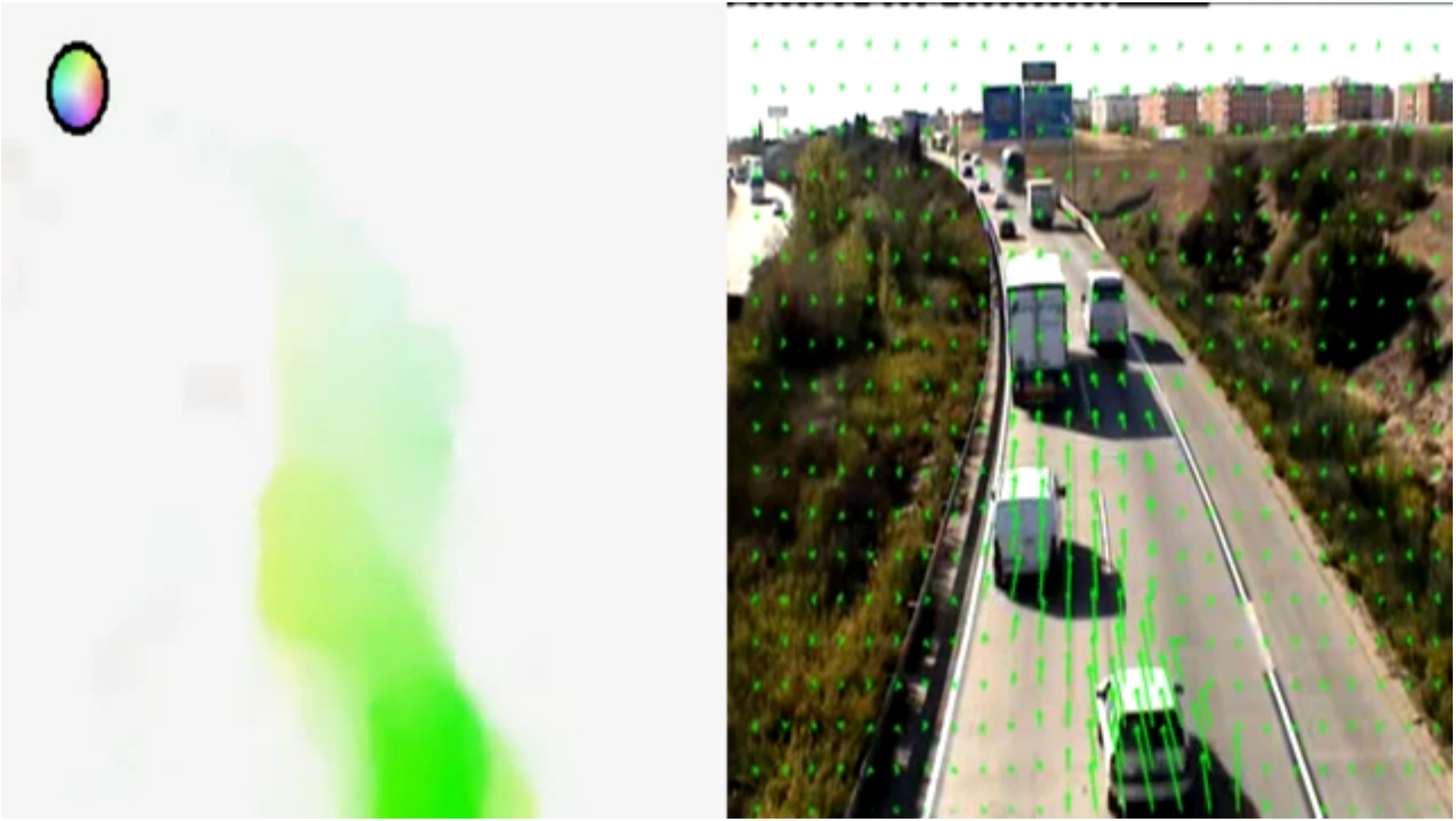
Readings: FP 10.6; SZ 8; TV 8

Date: 10/15/14

Materials on these slides have come from many sources in addition to myself.
Many are adaptations from Savarese, Lazebnik, Darrell, Hager, Pollefeys, Seitz, Szeliski, Saenko and Grauman.
Individual slides reference specific sources when possible.



https://www.youtube.com/watch?v=BDtvjKMl30w



https://www.youtube.com/watch?v=5rR_9YIcg_s

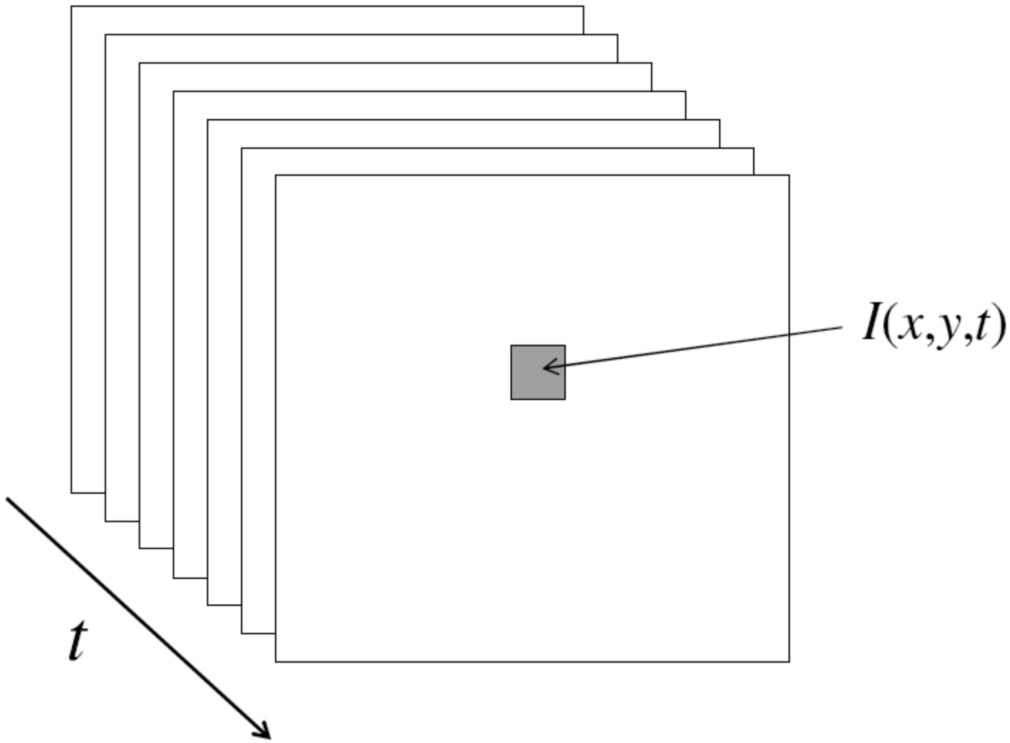
Plan

- Motion Field
- Patch-based / Direct
  Motion Estimation
- (Next: Feature Tracking)
- (Next: Layered Motion
  Models)

- External Resource:
  – Mubarak Shah’s lecture on optical flow
    - http://www.youtube.com/watch?v=5VyLAH8BhF8

Video

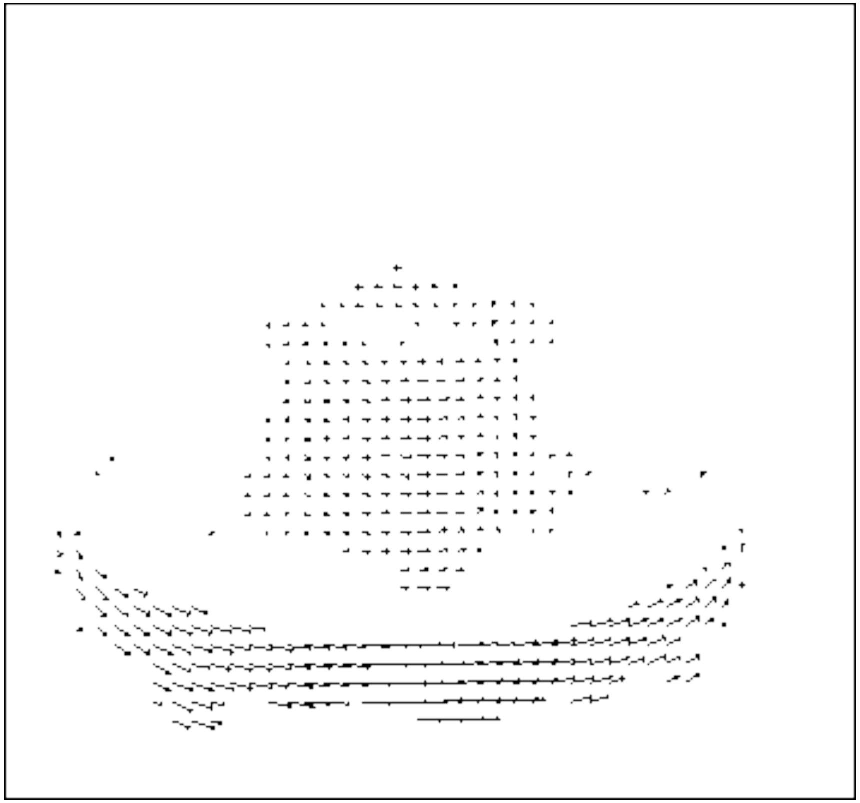
- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



Slide adapted from K. Grauman.

Motion field

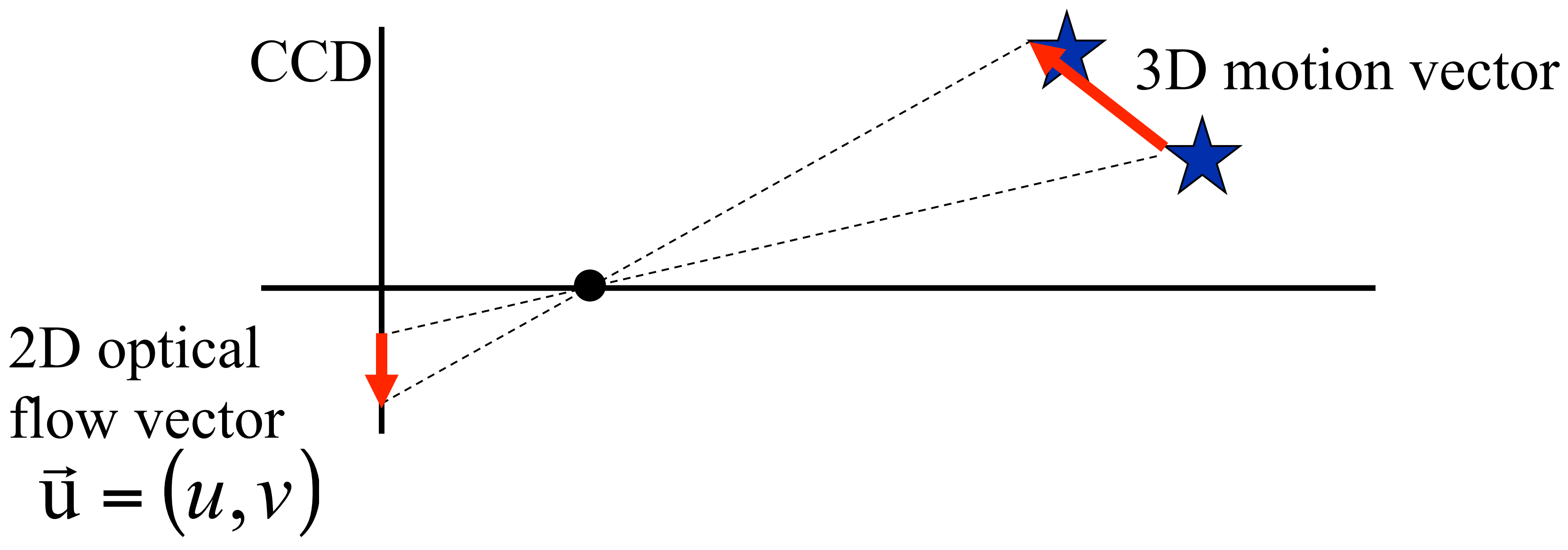
- The motion field is the projection of the 3D scene motion into the image



Slide adapted from K. Grauman; images from Russel and Norvig.

Motion Field & Optical Flow Field

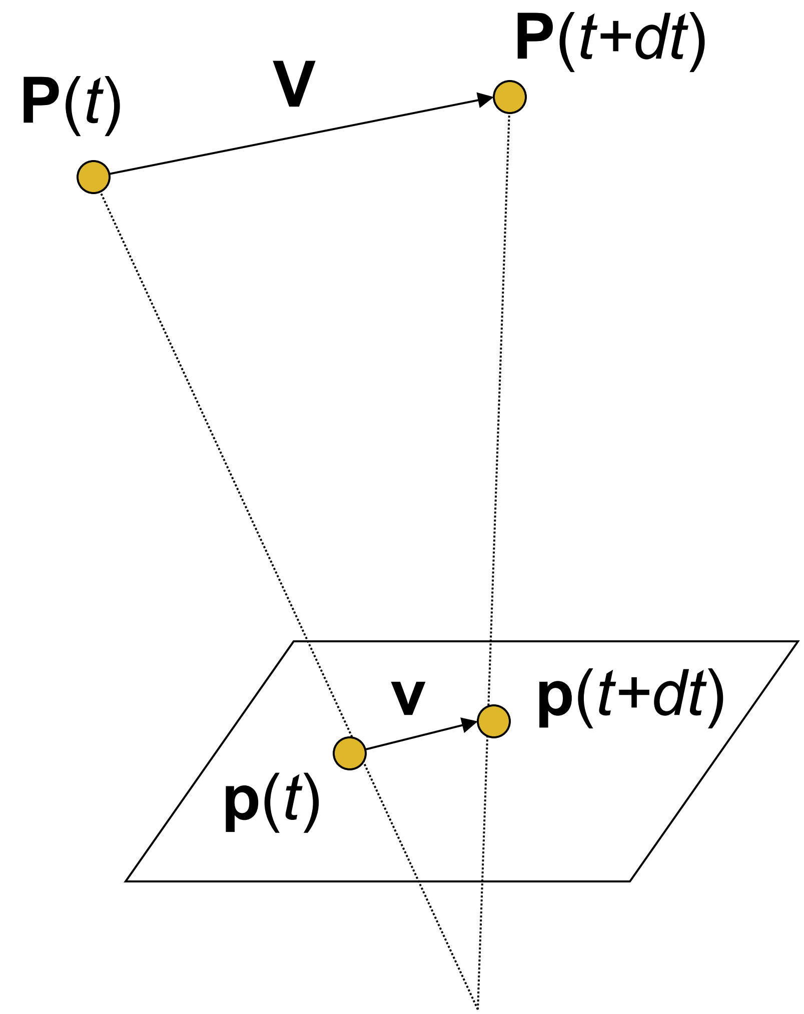
- Motion Field = Real world 3D motion
- Optical Flow Field = Projection of the motion field onto the 2d image



Slide adapted from Savarese.

Motion field and parallax

- $\mathbf{P}(t)$ is a moving 3D point
- Velocity of scene point: $\mathbf{V} = d\mathbf{P}/dt$
- $\mathbf{p}(t) = (x(t),y(t))$ is the projection of $\mathbf{P}$ in the image
- Apparent velocity $\mathbf{v}$ in the image: given by components $v_x = dx/dt$ and $v_y = dy/dt$
- These components are known as the *motion field* of the image

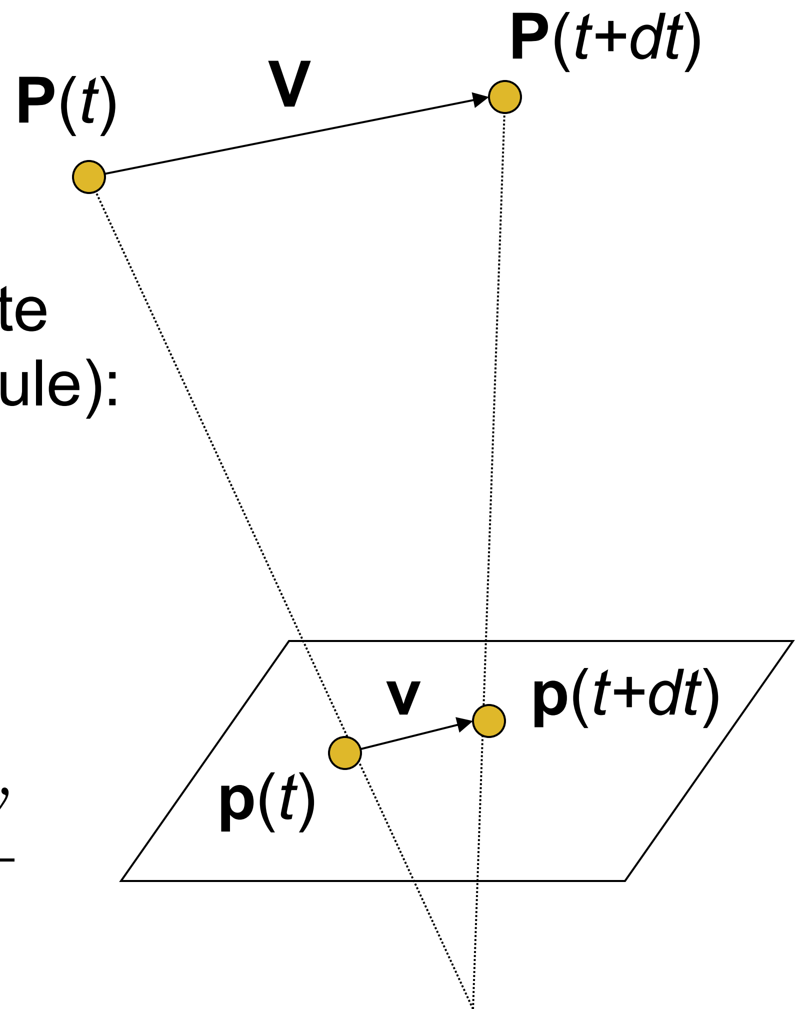


Slide adapted from K. Grauman.

Motion field and parallax

Quotient rule:
D(f/g) = (g f' – g'f)/g^2

$$\mathbf{V} = (V_x, V_y, V_Z) \quad \mathbf{p} = f\frac{\mathbf{P}}{Z}$$



To find image velocity **v**, differentiate **p** with respect to *t* (using quotient rule):

$$\mathbf{v} = f\frac{Z\mathbf{V} - V_z\mathbf{P}}{Z^2}$$

$$v_x = \frac{fV_x - V_z x}{Z} \qquad v_y = \frac{fV_y - V_z y}{Z}$$

Image motion is a function of both the 3D motion (**V**) and the depth of the 3D point (*Z*)

Motion field and parallax

- Pure translation: **V** is constant everywhere

$$v_x = \frac{fV_x - V_z x}{Z}$$

$$v_y = \frac{fV_y - V_z y}{Z}$$

$$\mathbf{v} = \frac{1}{Z}(\mathbf{v}_0 - V_z \mathbf{p}),$$

$$\mathbf{v}_0 = \left(fV_x, fV_y\right)$$

Motion field and parallax

- Pure translation: **V** is constant everywhere

$$\mathbf{v} = \frac{1}{Z}(\mathbf{v}_0 - V_z\mathbf{p}),$$

$$\mathbf{v}_0 = \left(fV_x, fV_y\right)$$

- V_z is nonzero:
  - Every motion vector points toward (or away from) $\mathbf{v}_0$, the vanishing point of the translation direction

Slide adapted from K. Grauman.

Motion field and parallax

- Pure translation: **V** is constant everywhere

$$\mathbf{v} = \frac{1}{Z}(\mathbf{v}_0 - V_z \mathbf{p}),$$

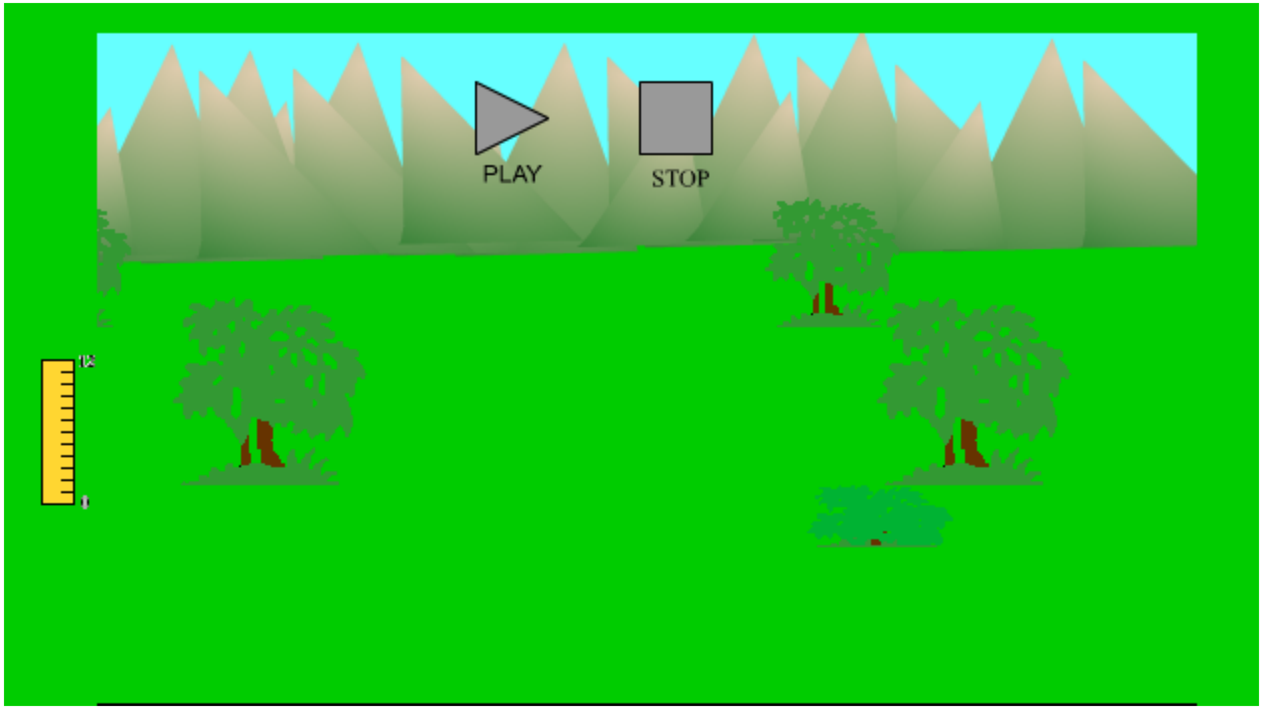
$$\mathbf{v}_0 = (f V_x, f V_y)$$

- V_z is nonzero:
  - Every motion vector points toward (or away from) $\mathbf{v}_0$, the vanishing point of the translation direction
- V_z is zero:
  - Motion is parallel to the image plane, all the motion vectors are parallel
- The length of the motion vectors is inversely proportional to the depth Z

Slide adapted from K. Grauman.

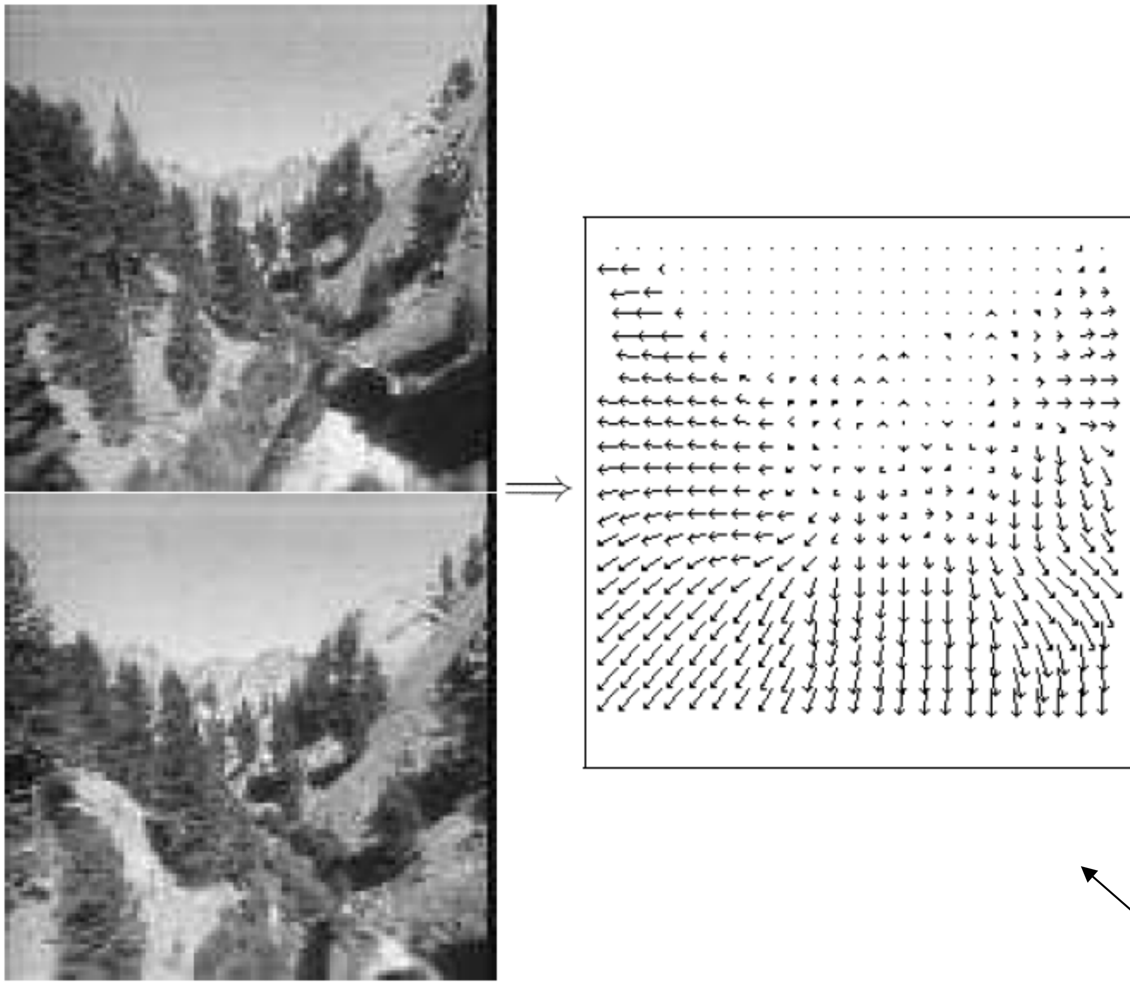
Motion parallax

- http://psych.hanover.edu/KRANTZ/MotionParallax/MotionParallax.html



Slide adapted from K. Grauman.

Motion field + camera motion



Length of flow vectors inversely proportional to depth Z of 3d point

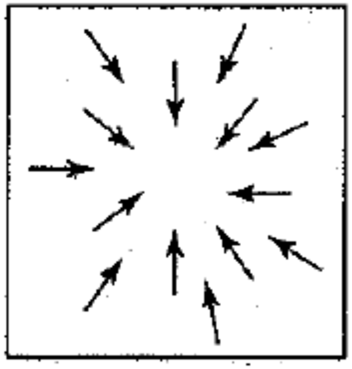
Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

points closer to the camera move more quickly across the image plane

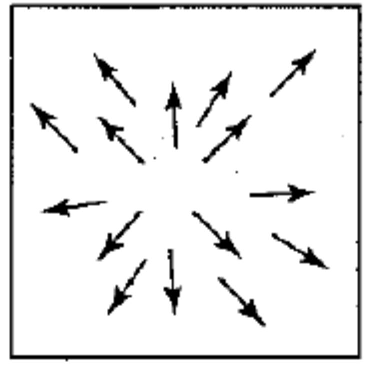
Figure from Michael Black, Ph.D. Thesis

Slide adapted from K. Grauman.

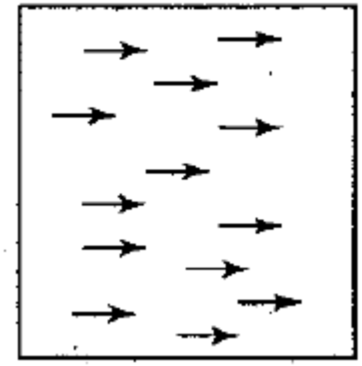
Motion field + camera motion



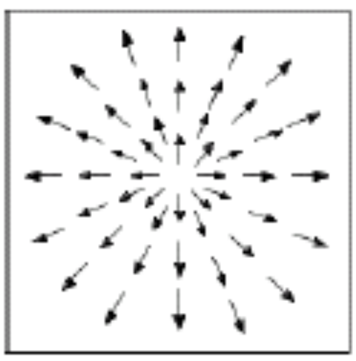
Zoom out



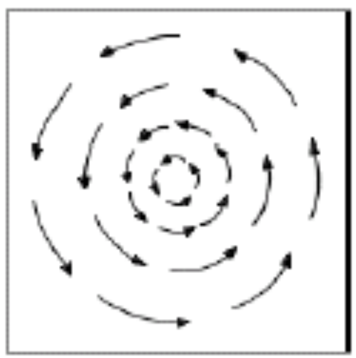
Zoom in



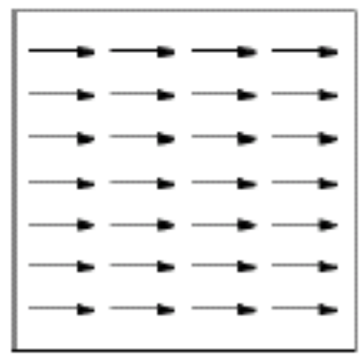
Pan right to left



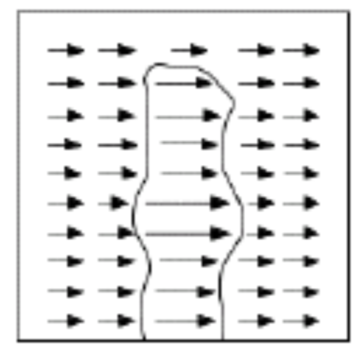
Forward motion



Rotation



Horizontal translation



Closer objects appear to move faster!!

Slide adapted from Savarese and Grauman.

Motion estimation techniques

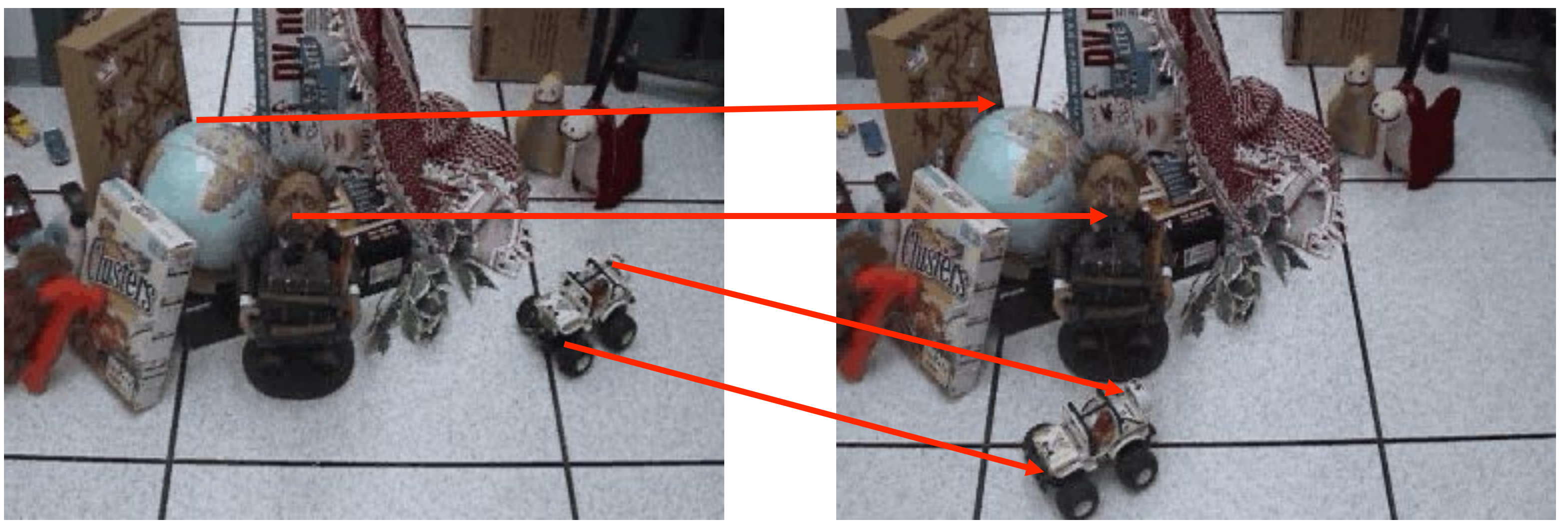
- Feature-based methods
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small

Slide adapted from K. Grauman.

Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image
- Ideally, optical flow would be the same as the motion field
- Have to be careful: apparent motion can be caused by lighting changes without any actual motion

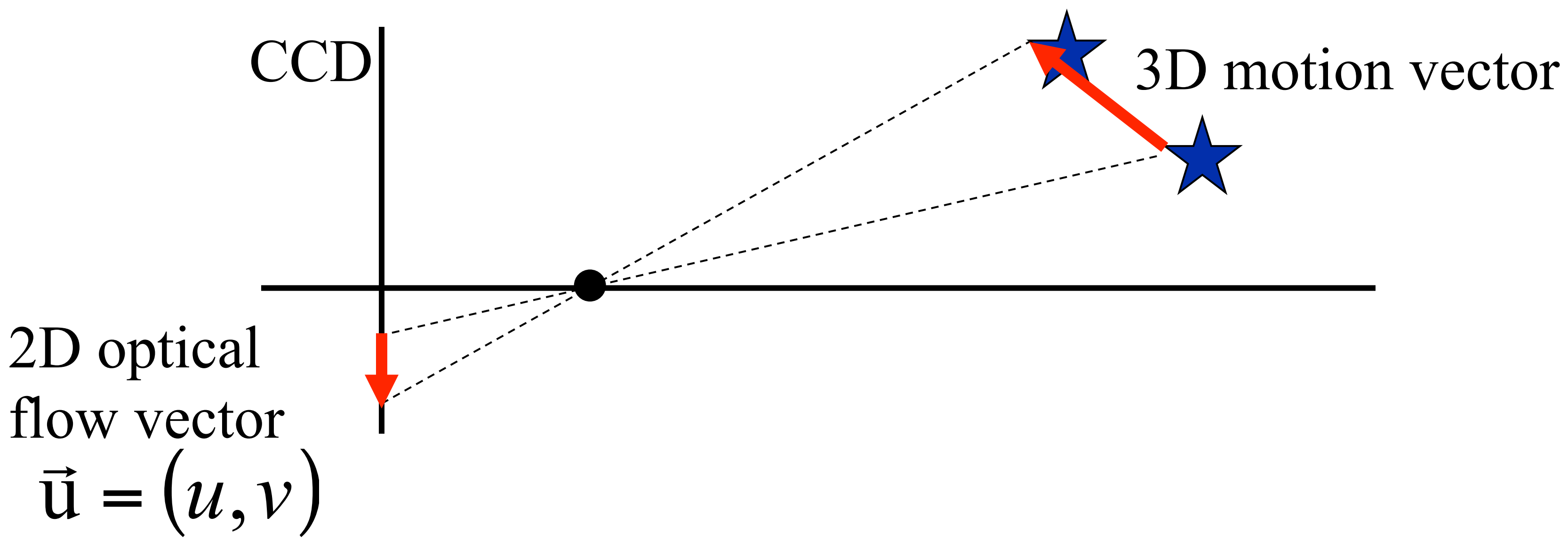


Where did each pixel in image 1 go to in image 2

Slide adapted from K. Grauman, Zelnik-Manor, Savarese.

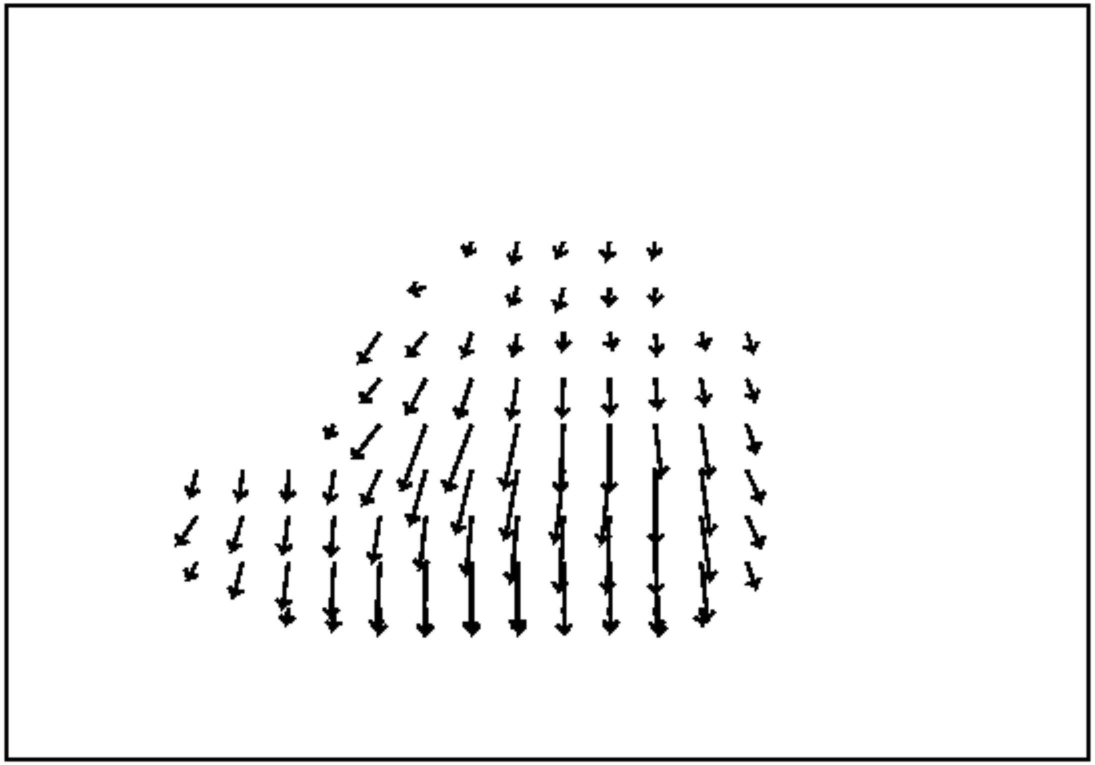
Motion Field & Optical Flow Field

- Motion Field = Real world 3D motion
- Optical Flow Field = Projection of the motion field onto the 2d image



Slide adapted from Savarese.

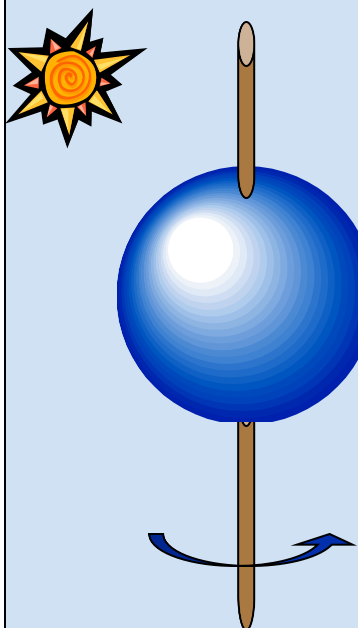
Optical Flow



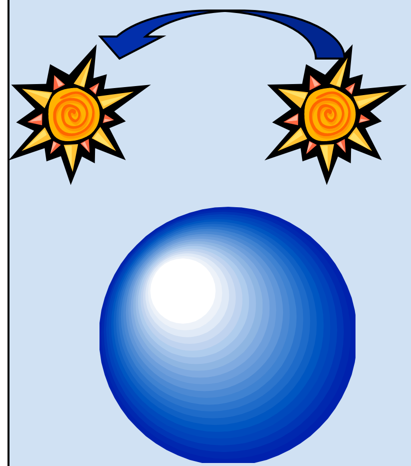
Pierre Kornprobst's Demo

When does it break?

The screen is stationary yet displays motion



Homogeneous objects generate zero optical flow.



Fixed sphere. Changing light source.

Non-rigid texture motion

Slide adapted from Savarese.

Apparent motion ~= motion field

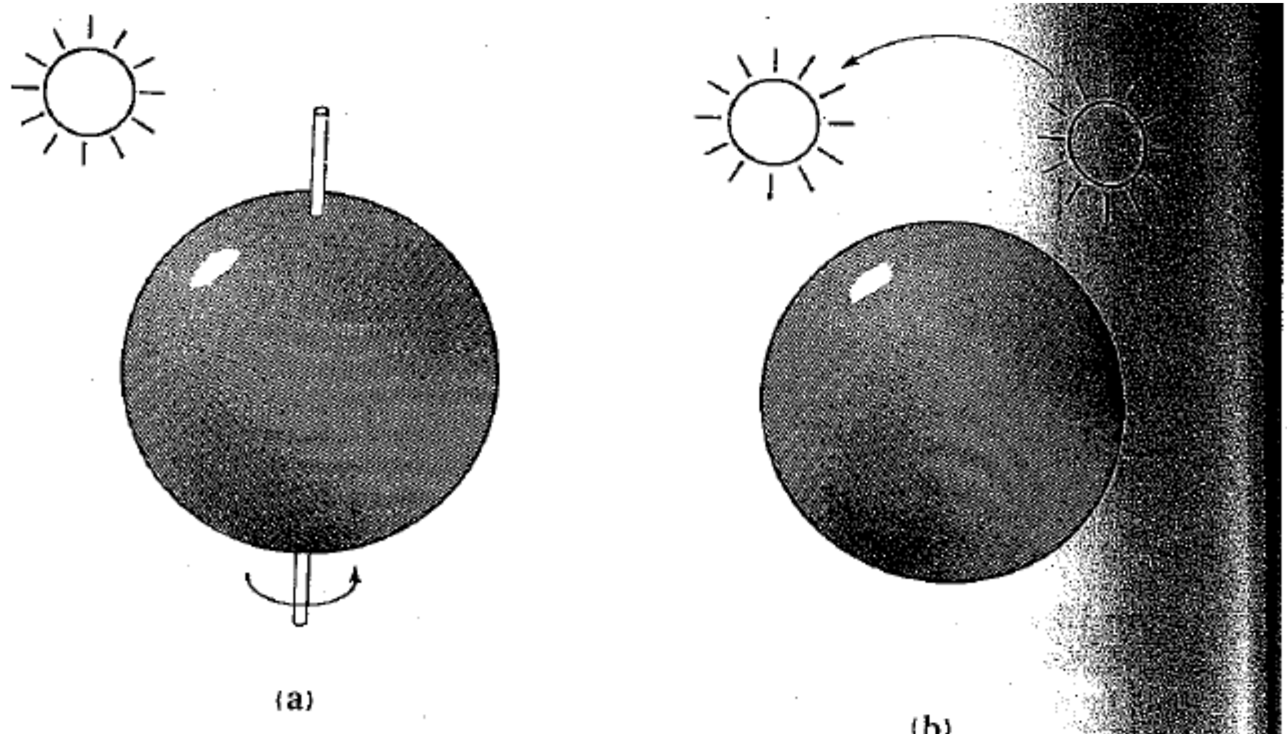


Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field is zero.

Figure from Horn book

Slide adapted from K. Grauman.

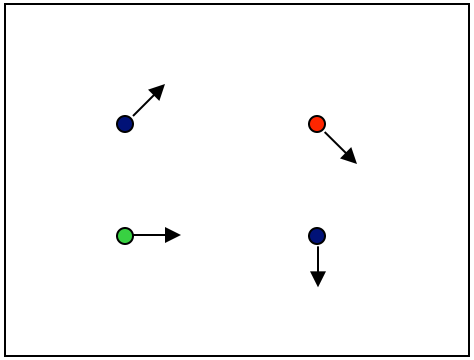
The Optical Flow Field

Still, in many cases it does work….

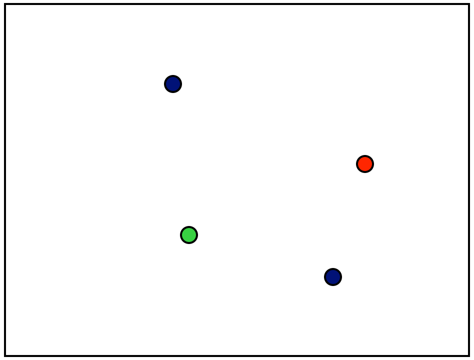
- Goal:
  Find for each pixel a velocity vector $\vec{\mathrm{u}} = (u, v)$
  which says:
  - How quickly is the pixel moving across the image
  - In which direction it is moving

Slide adapted from Savarese.

Estimating optical flow



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



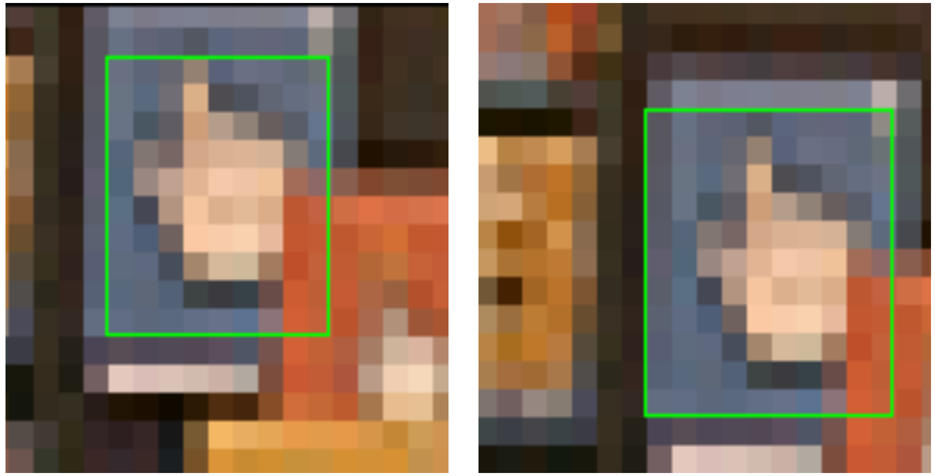
$I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field 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

Slide adapted from K. Grauman.

Brightness constancy



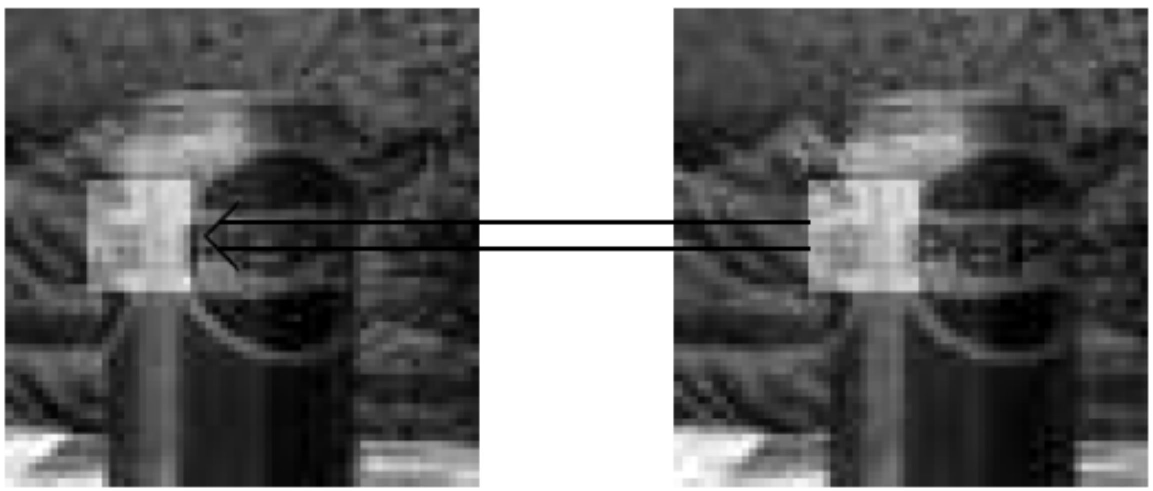
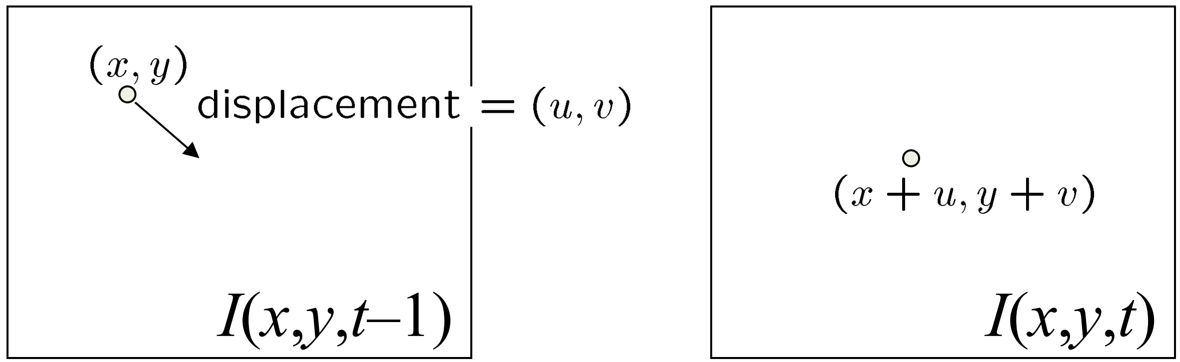


Figure 1.5: Data conservation assumption. The highlighted region in the right image looks roughly the same as the region in the left image, despite the fact that it has moved.

Figure by Michael Black

Slide adapted from K. Grauman, T. Darrell, R. Szeliski.

The brightness constancy constraint



- Brightness Constancy Equation:

$$I(x, y, t-1) = I(x + u(x, y), y + v(x, y), t)$$

Can be written as:

$$\boxed{\text{shorthand: } I_x = \frac{\partial I}{\partial x}}$$

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

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

Slide adapted from K. Grauman.

The brightness constancy constraint

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

- How many equations and unknowns per pixel?
  - One equation, two unknowns

- Intuitively, what does this constraint mean?

$$\nabla I \cdot (u, v) + I_t = 0$$

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

The brightness constancy constraint

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

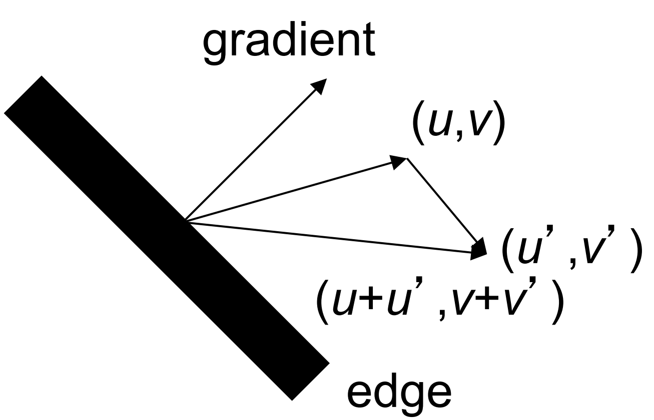
- How many equations and unknowns per pixel?
  - One equation, two unknowns

- Intuitively, what does this constraint mean?

$$\nabla I \cdot (u, v) + I_t = 0$$

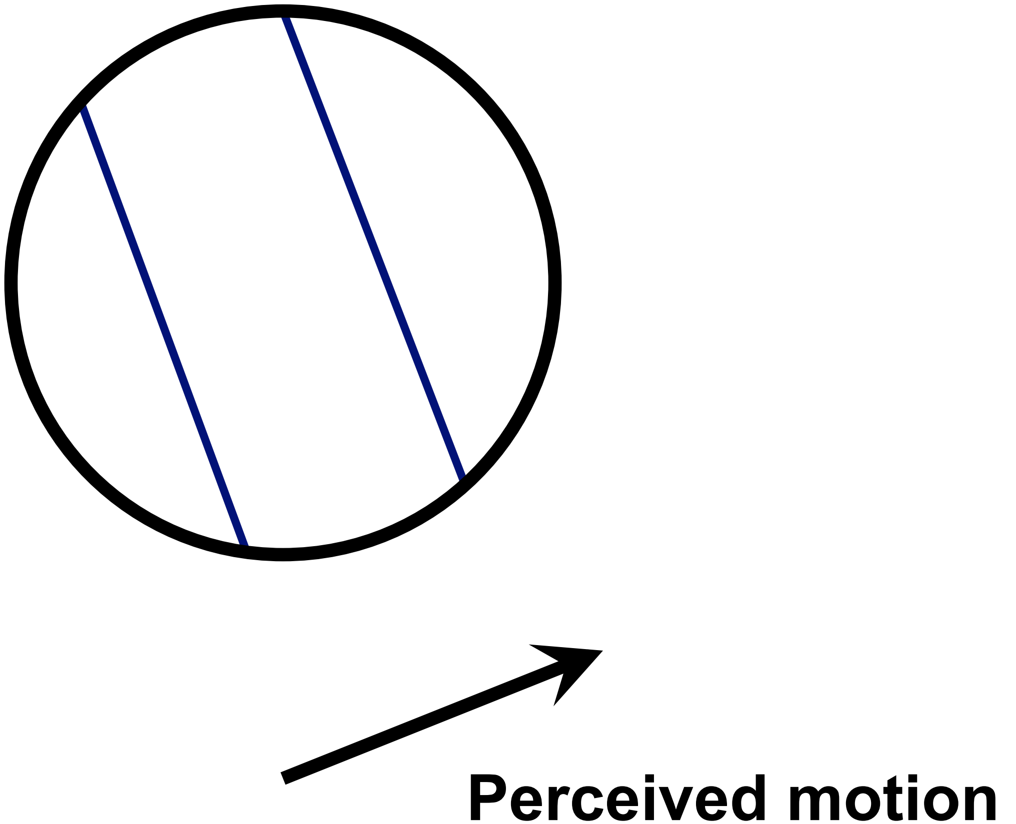
- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

If (u, v) satisfies the equation,
so does ($u+u'$, $v+v'$) if $\nabla I \cdot (u', v') = 0$

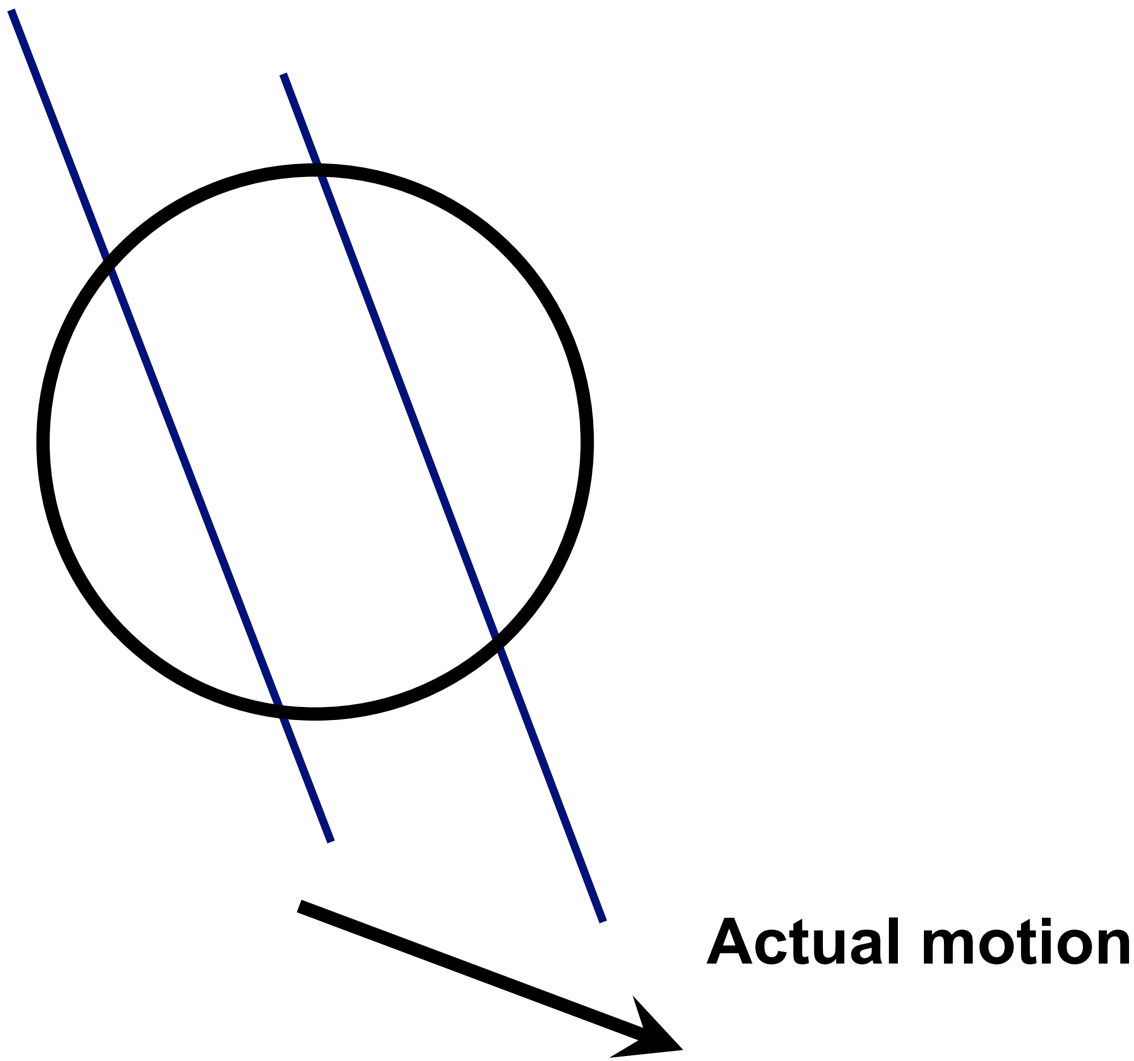


Slide adapted from K. Grauman.

The aperture problem



The aperture problem



Slide adapted from K. Grauman.

The barber pole illusion

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

Slide adapted from K. Grauman.

The barber pole illusion

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

Slide adapted from K. Grauman.

The barber pole illusion

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

Slide adapted from K. Grauman.

Lucas-Kanade: Solving the aperture problem (grayscale image)

- How to get more equations for a pixel?
- **Spatial coherence constraint:** pretend the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

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

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

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

Lucas-Kanade: Solving the aperture problem

Prob: we have more equations than unknowns

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

Solution: solve least squares problem

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

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

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

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

Slide adapted from Seitz, Grauman.

Conditions for solvability

Look Familiar?

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

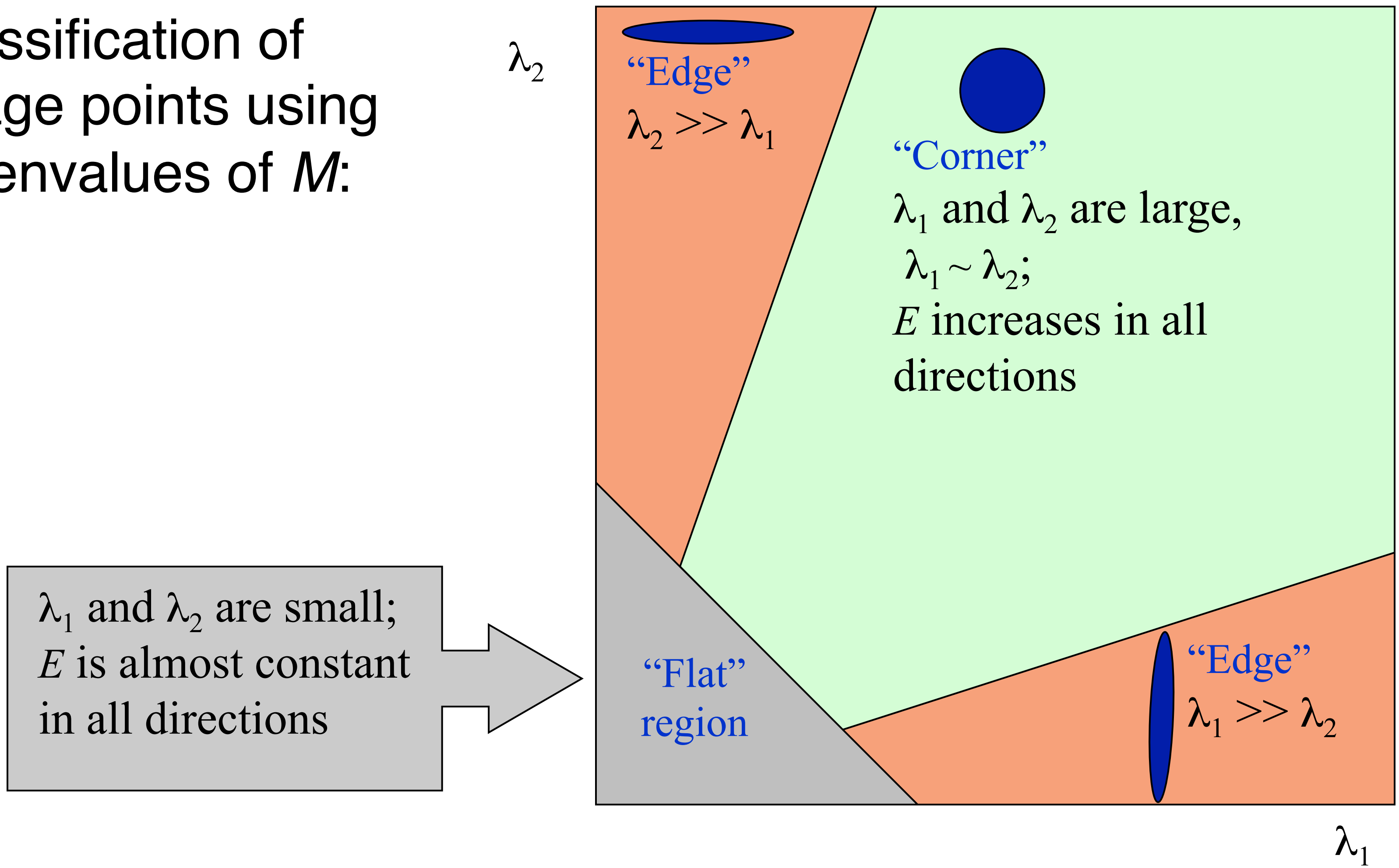
When is this solvable?

- **A^TA** should be invertible
- **A^TA** should not be too small
  - eigenvalues λ_1 and λ_2 of **A^TA** should not be too small
- **A^TA** should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Slide adapted from Seitz, Grauman.

Conditions for solvability

Classification of image points using eigenvalues of *M*:



Source: Kokkinos, Saverese.

Edge

– gradients very large or very small

– large λ_1, small λ_2

Low-texture region

– gradients have small magnitude

– small λ_1, small λ_2

High-texture region

– gradients are different, large magnitudes

– large λ_1, large λ_2

Slide adapted from Seitz, Grauman.

Can we measure optical flow reliability?

- Can we measure “quality” of optical flow in regions from just a single image?

- High Quality / Good features to track:
- - Harris corners (guarantee small error sensitivity)

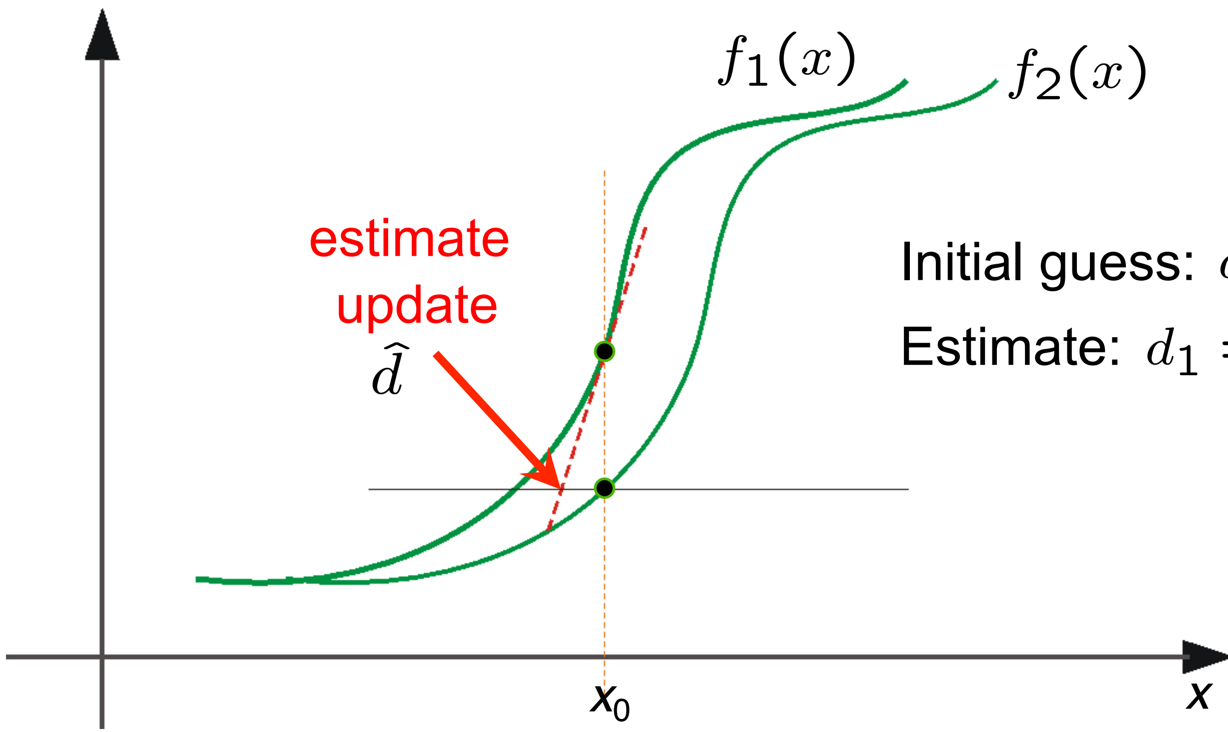
- Poor Quality / Bad features to track:
- - Image points when either λ_1 or λ_2 (or both) is small (i.e., edges or uniform textured regions)

Iterative Refinement

- Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation
- Warp one image toward the other using the estimated flow field

  (easier said than done)
- Refine estimate by repeating the process

Optical Flow: Iterative Estimation



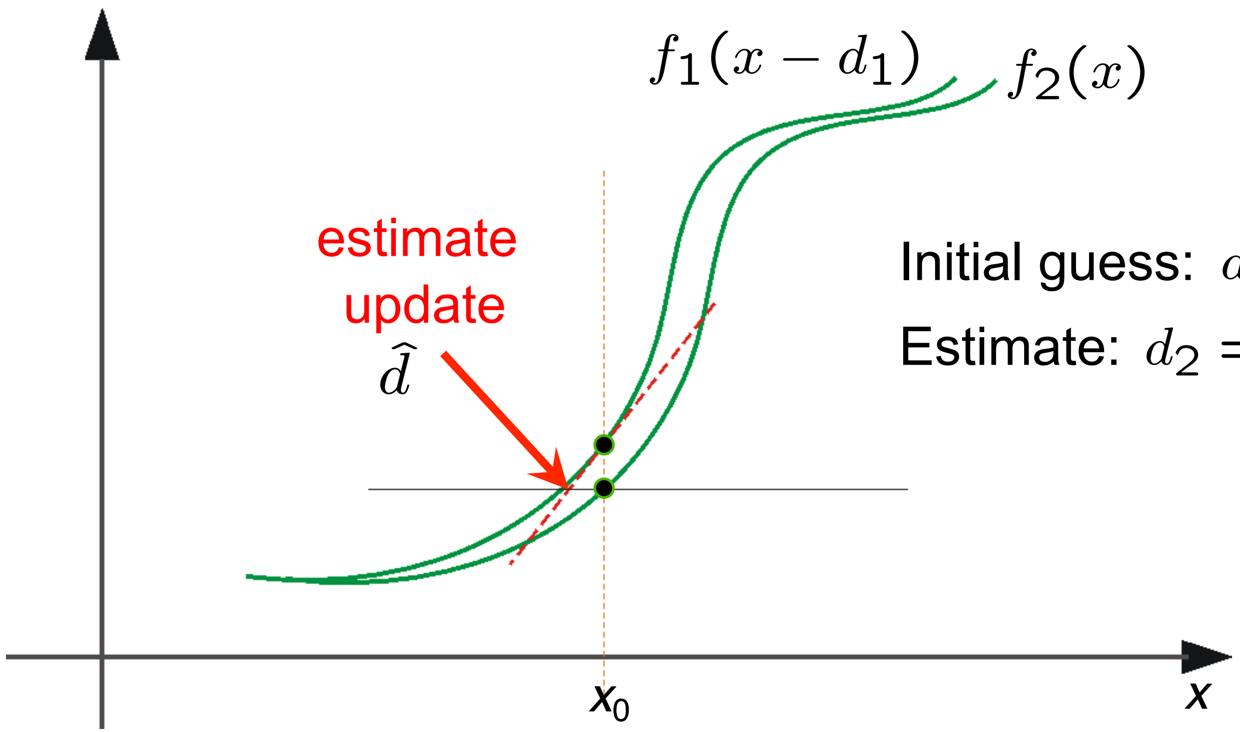
Initial guess: $d_0 = 0$

Estimate: $d_1 = d_0 + \hat{d}$

(using *d* for *displacement* here instead of *u*)

Slide adapted from Szeliski.

Optical Flow: Iterative Estimation

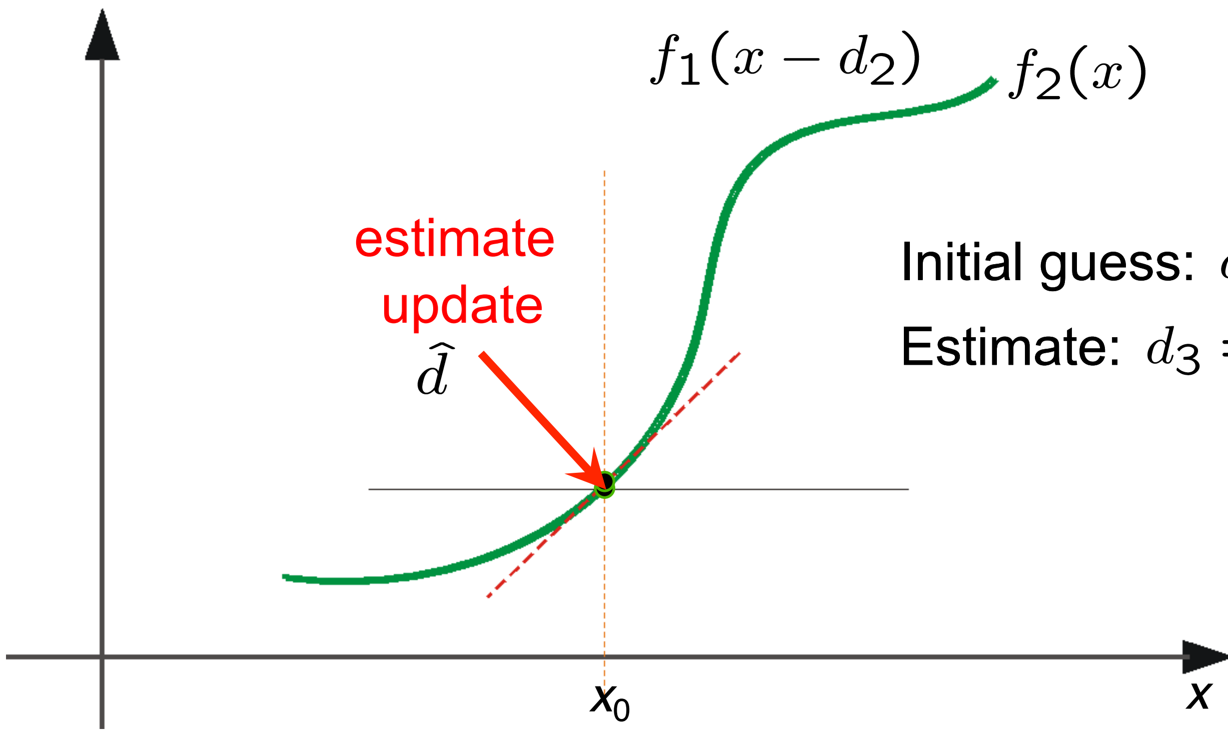


Initial guess: d_1

Estimate: $d_2 = d_1 + \hat{d}$

Slide adapted from Szeliski.

Optical Flow: Iterative Estimation

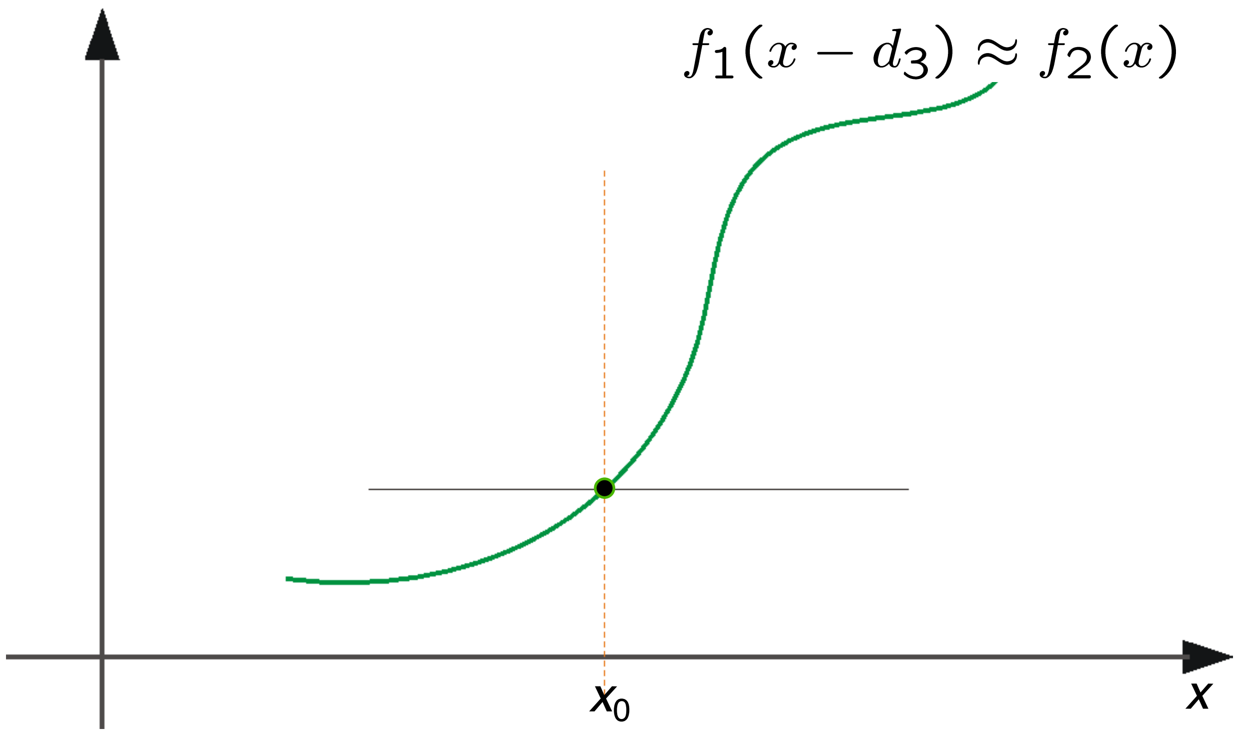


Initial guess: d_2

Estimate: $d_3 = d_2 + \hat{d}$

Slide adapted from Szeliski.

Optical Flow: Iterative Estimation



Slide adapted from Szeliski.

Optical Flow: Iterative Estimation

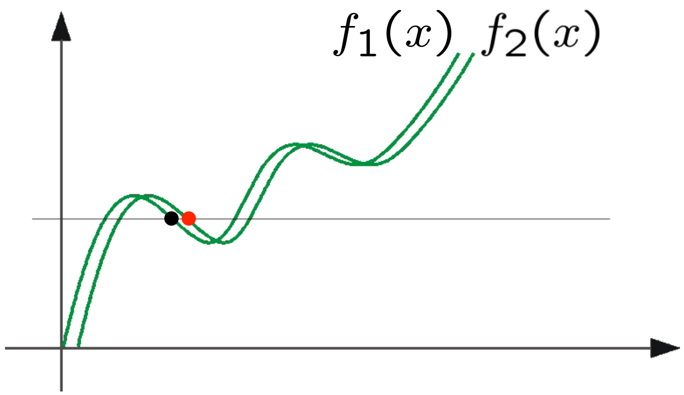
- Some Implementation Issues:
  - Warping is not easy (ensure that errors in warping are smaller than the estimate refinement)
  - Warp one image, take derivatives of the other so you don't need to re-compute the gradient after each iteration.
  - Often useful to low-pass filter the images before motion estimation (for better derivative estimation, and linear approximations to image intensity)

Slide adapted from Szeliski.

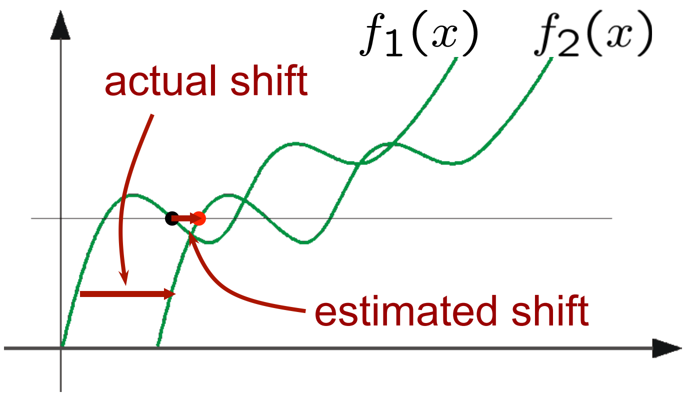
Optical Flow: Aliasing

Temporal aliasing causes ambiguities in optical flow because images can have many pixels with the same intensity.

I.e., how do we know which ‘correspondence’ is correct?



nearest match is correct (no aliasing)



nearest match is incorrect (aliasing)

To overcome aliasing: coarse-to-fine estimation.

Limits of the gradient method

Fails when intensity structure in window is poor

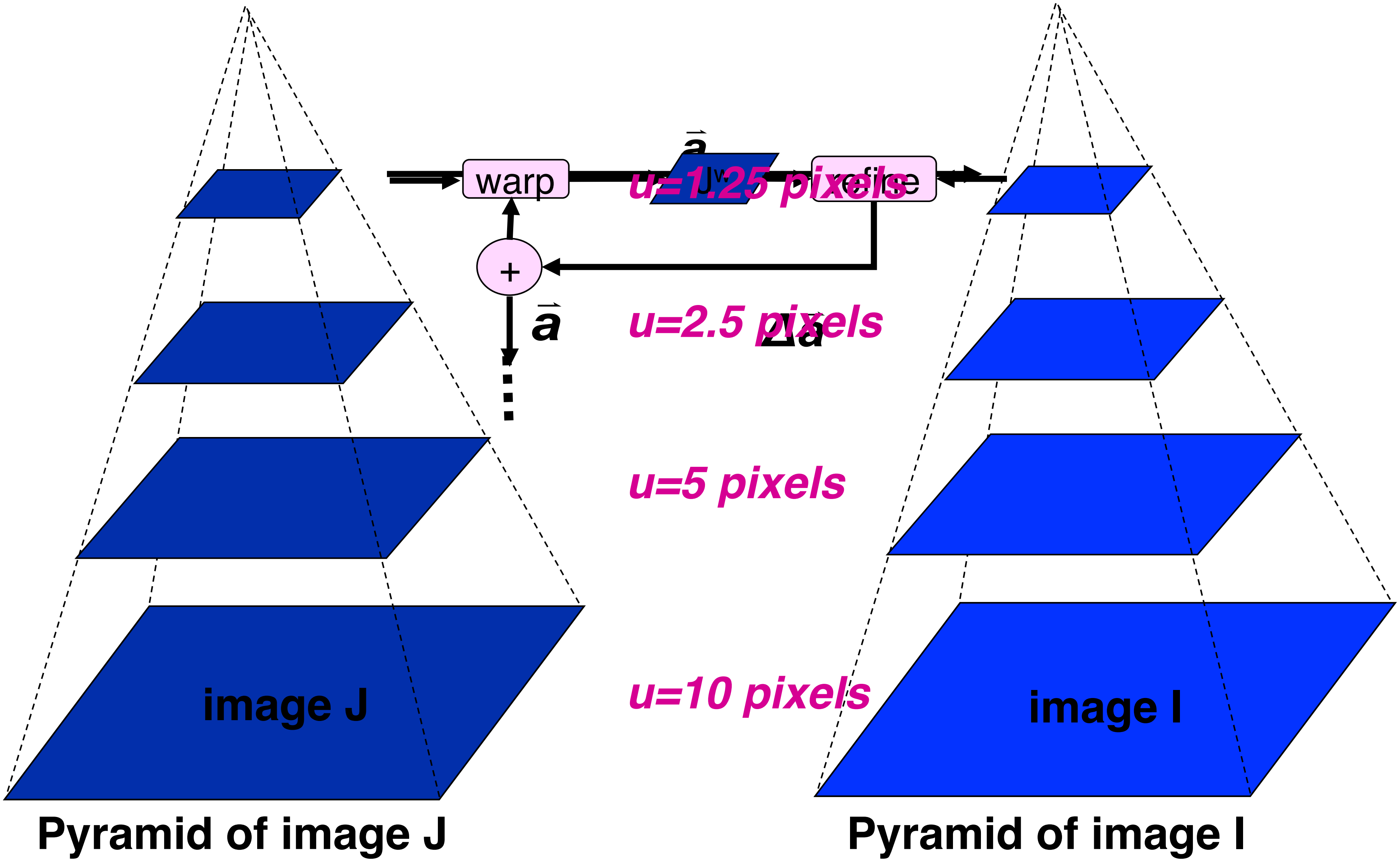
Fails when the displacement is large (typical operating range is motion of 1 pixel)

Linearization of brightness is suitable only for small displacements

- Also, brightness is not strictly constant in images

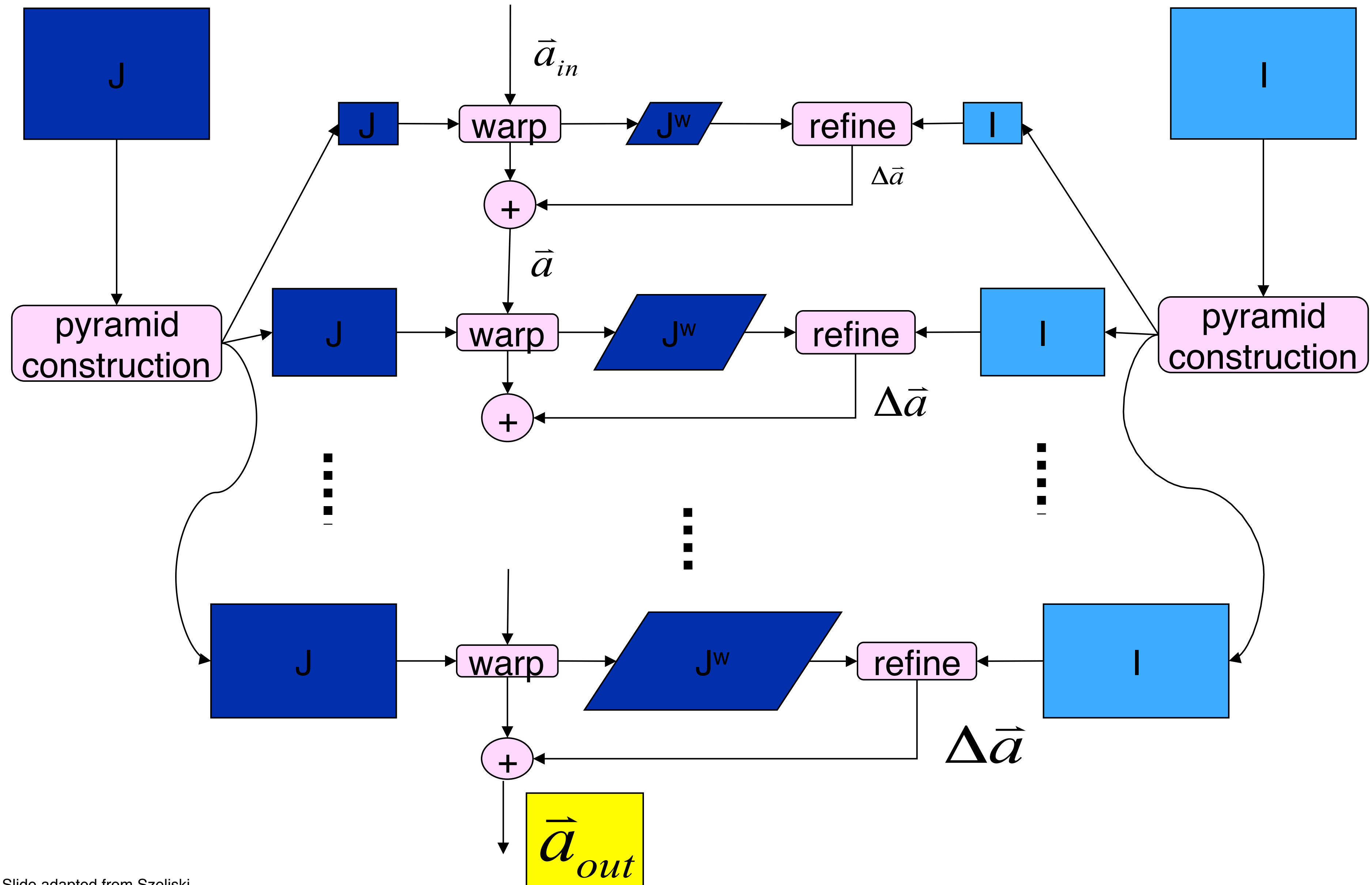
actually less problematic than it appears, since we can pre-filter images to make them look similar

Coarse-to-Fine Estimation



Slide adapted from Szeliski.

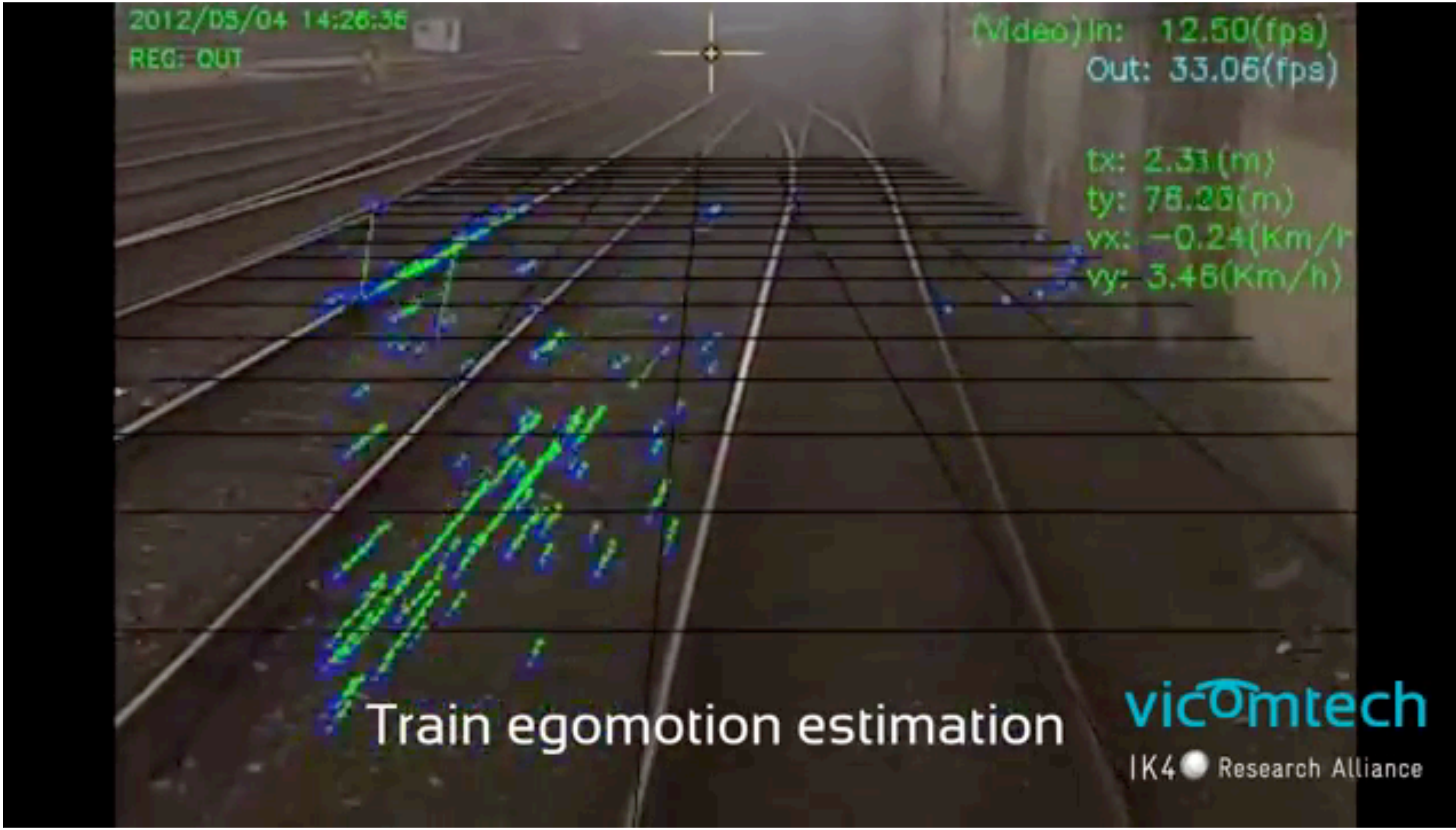
Coarse-to-Fine Estimation



Slide adapted from Szeliski.

Applications of Optical Flow

Egomotion Estimation on the Railway



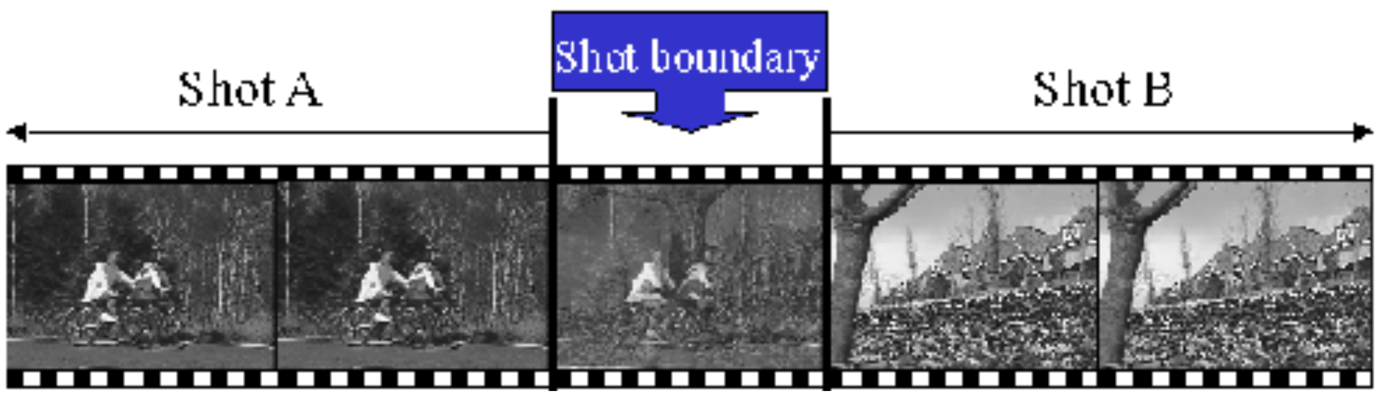
Applications to Segmentation

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

How to come up with background frame estimate without access to "empty" scene?

Applications to Segmentation

- Background subtraction
- Shot boundary detection
  - Commercial video is usually composed of *shots* or sequences showing the same objects or scene
  - Goal: segment video into shots for summarization and browsing (each shot can be represented by a single keyframe in a user interface)
  - Difference from background subtraction: the camera is not necessarily stationary



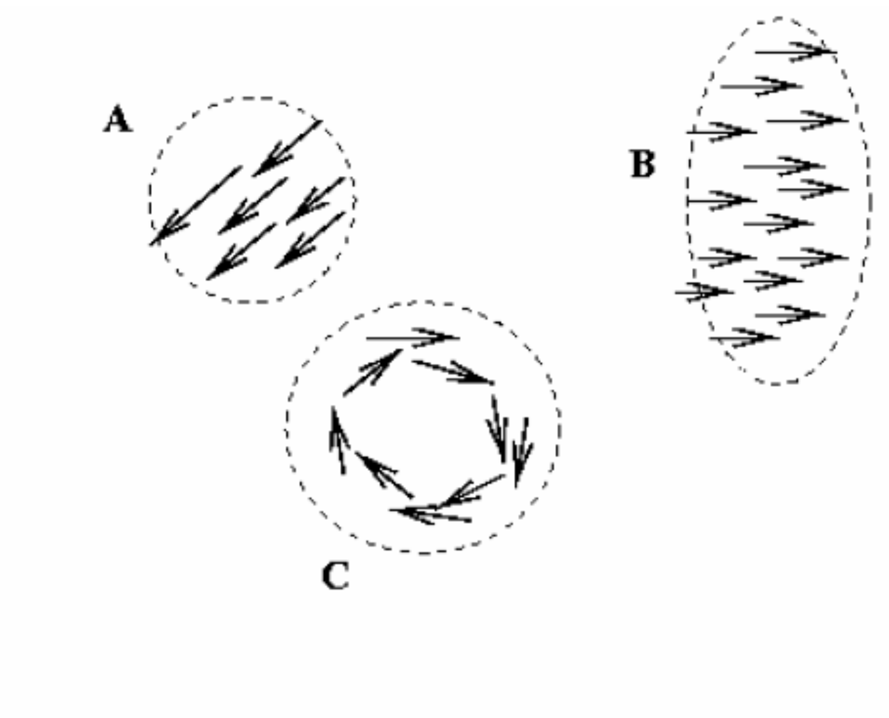
Slide adapted from K. Grauman.

Applications to Segmentation

- Background subtraction
- Shot boundary detection
  - For each frame
    - Compute the distance between the current frame and the previous one
      - Pixel-by-pixel differences
      - Differences of color histograms
      - Block comparison
    - If the distance is greater than some threshold, classify the frame as a shot boundary

Applications To Segmentation

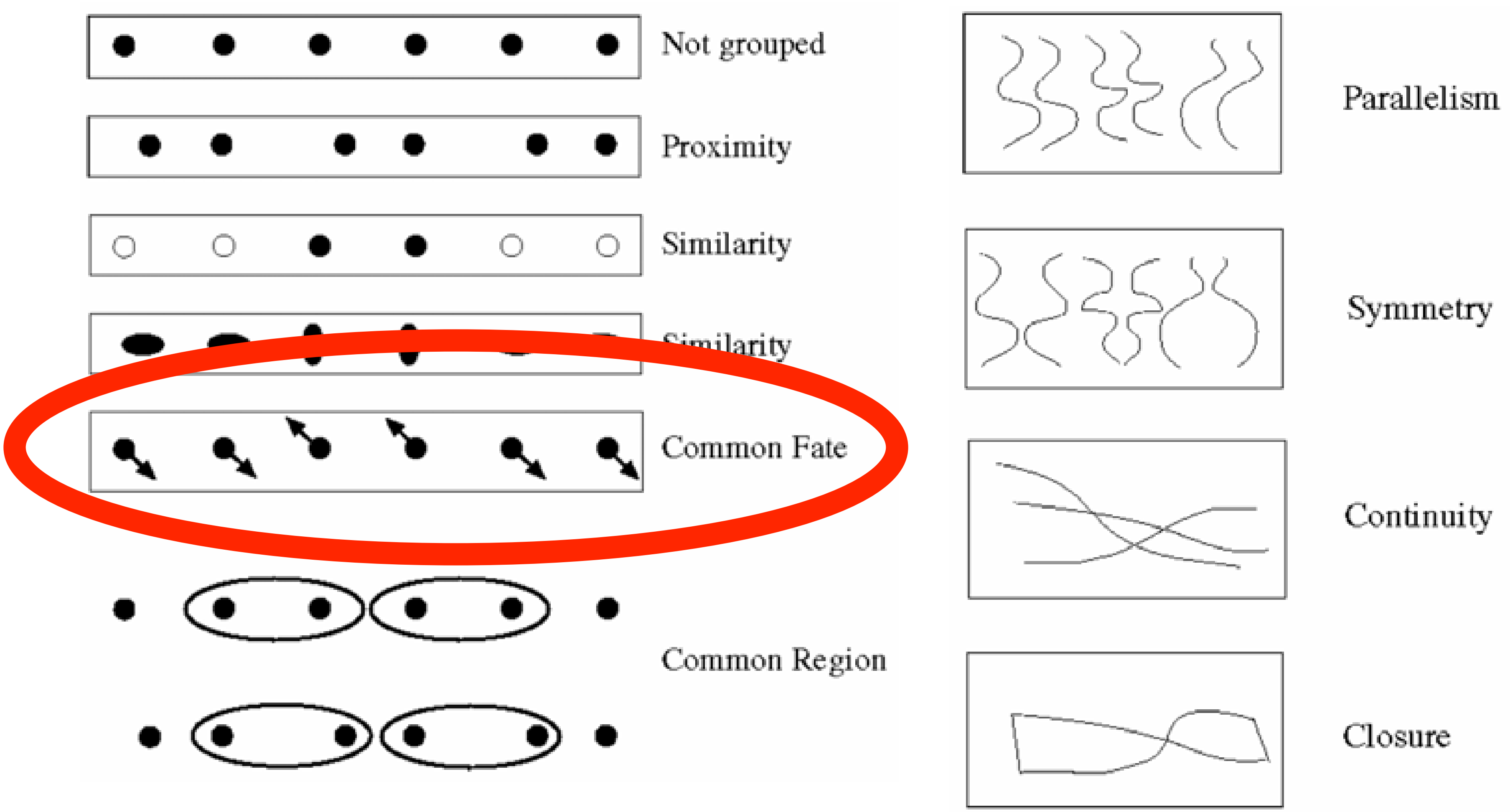
- Background subtraction
- Shot boundary detection
- Motion segmentation
  - Segment the video into multiple *coherently* moving objects



Slide adapted from K. Grauman.

Motion and perceptual organization

- Sometimes, motion is the only cue



Slide adapted from K. Grauman.

Motion and perceptual organization

- Sometimes, motion is foremost cue

Slide adapted from K. Grauman.

Motion and perceptual organization

- Even "impoverished" motion data can evoke a strong percept

Sources: Maas 1971 with Johansson; downloaded from Youtube.

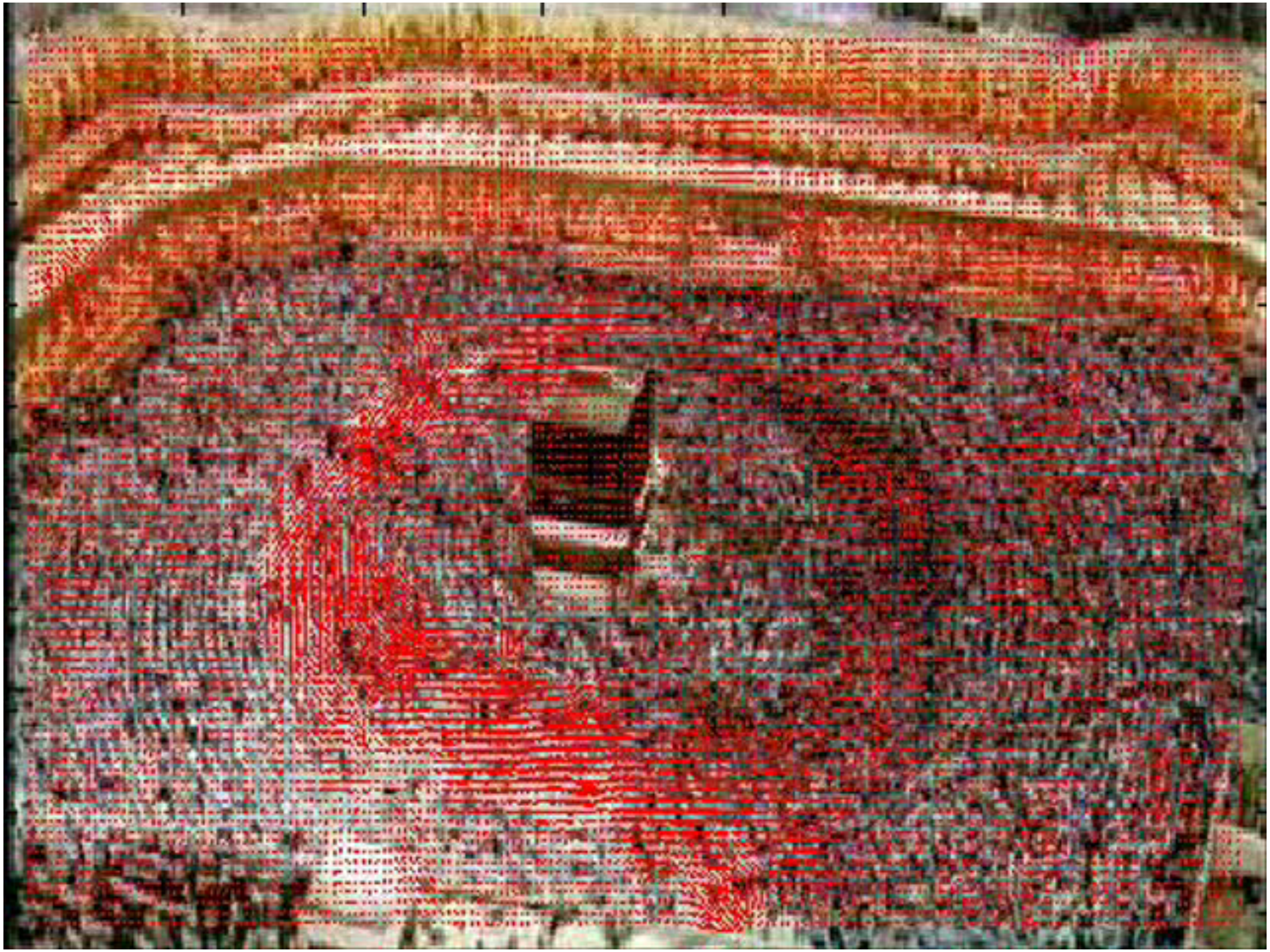
Slide adapted from K. Grauman.

Uses of motion

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

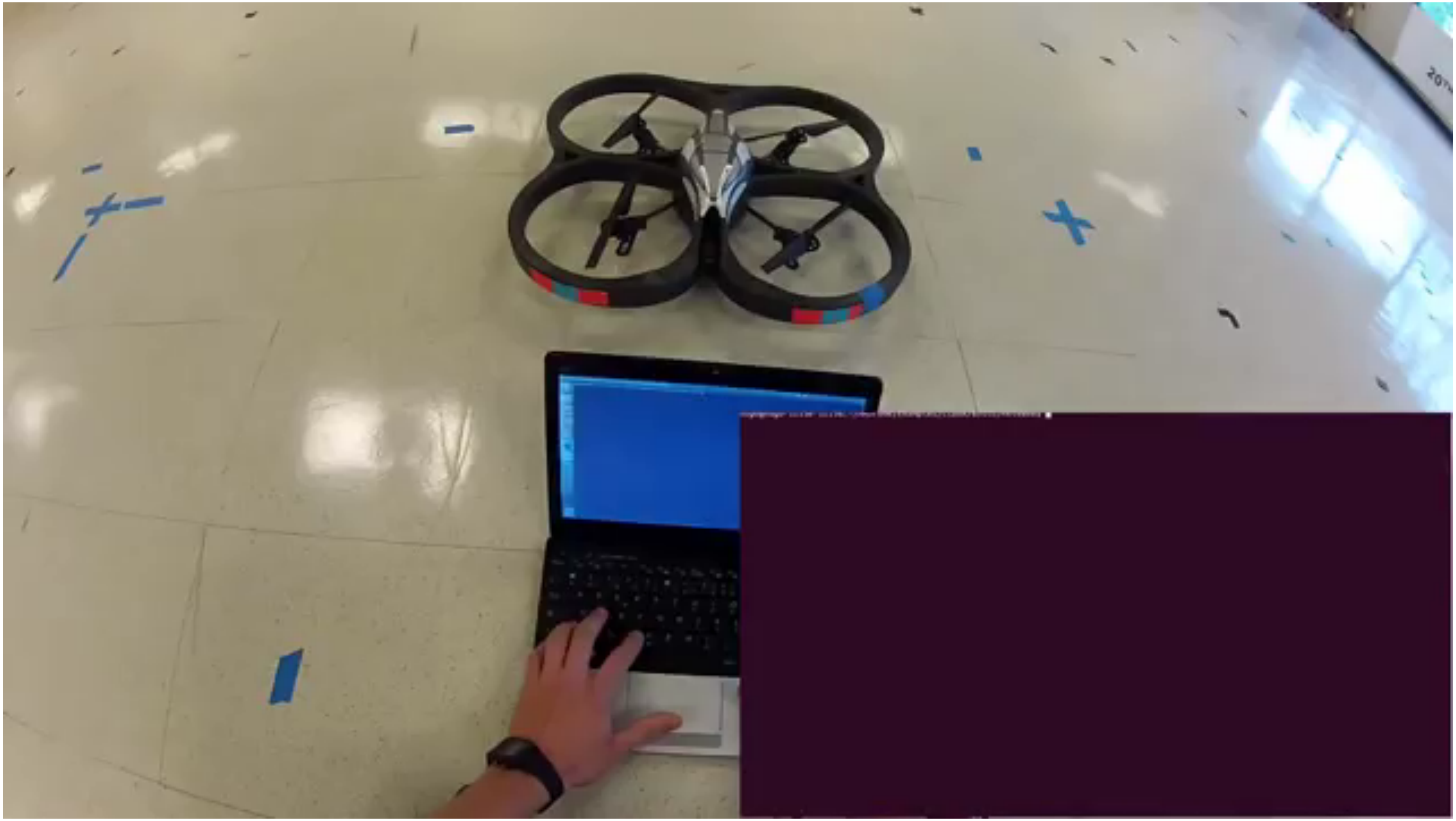
Slide adapted from K. Grauman.

Crowd Analysis



http://www.vision.eecs.ucf.edu/projects/sali/CrowdSegmentation/Mecca_flowfield.wmv

Aerial Vehicle Target Tracking



https://www.youtube.com/watch?v=C95bngCOv9Q

A Camera Mouse

- Video interface: use feature tracking as mouse replacement

- User clicks on the feature to be tracked
- Take the 15x15 pixel square of the feature
- In the next image do a search to find the 15x15 region with the highest correlation
- Move the mouse pointer accordingly
- Repeat in the background every 1/30th of a second

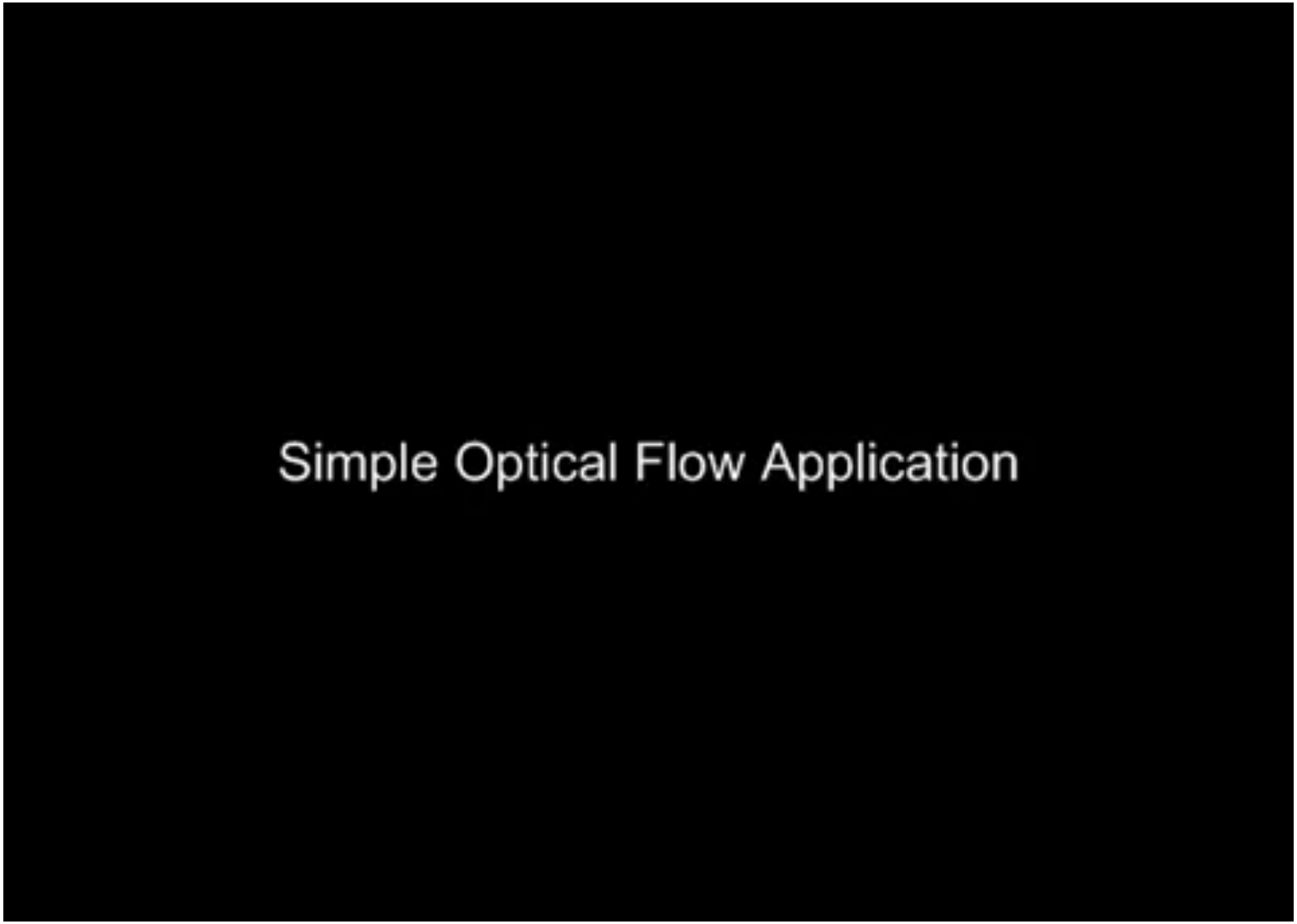
James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

A Camera Mouse

- Specialized software for communication, games

James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

Optical Flow for Games!



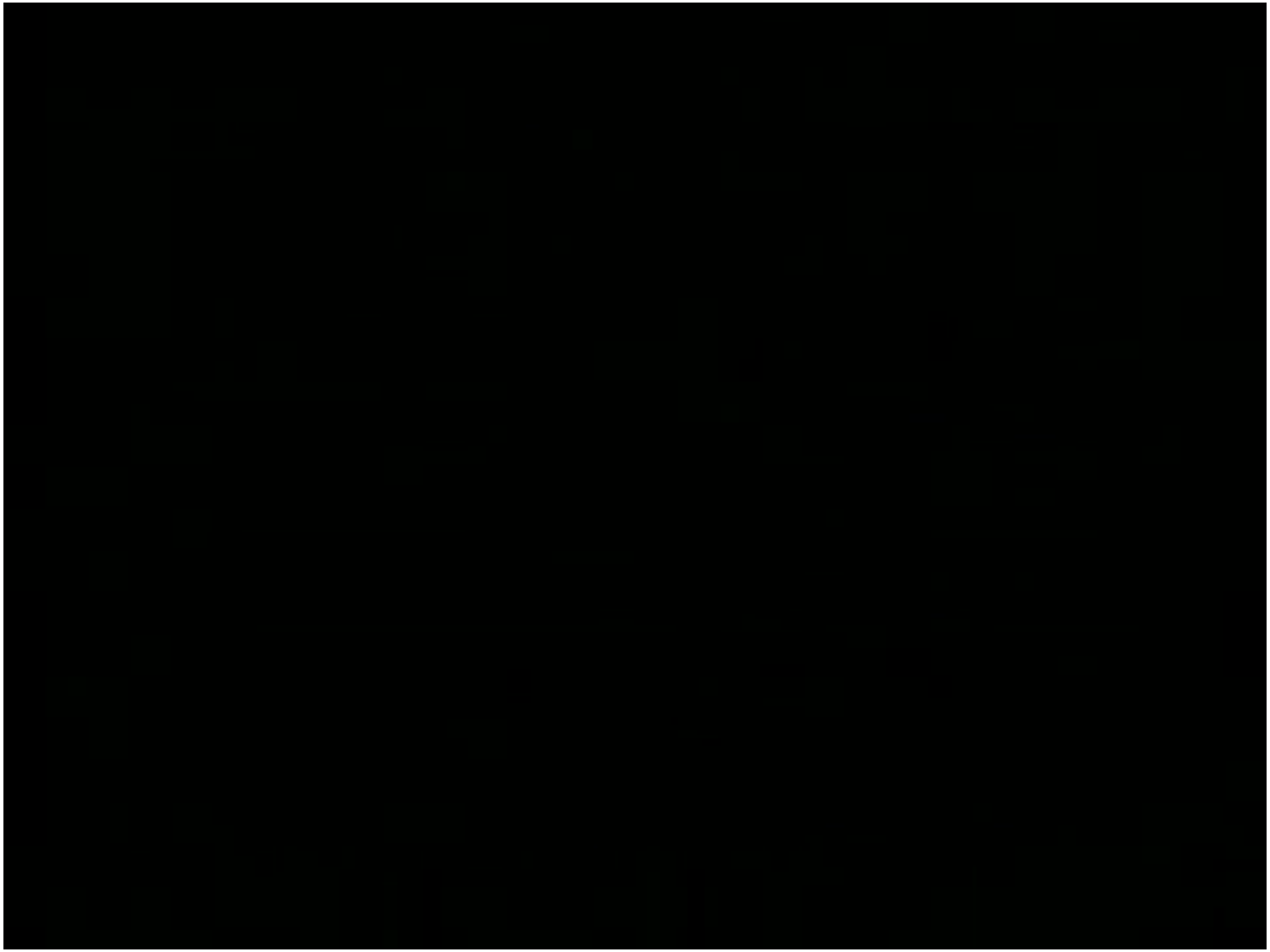
https://www.youtube.com/watch?v=E7h4OaTtCzY

Motion Paint: an example use of optical flow

Use optical flow to track brush strokes, in order to animate them to follow underlying scene motion.

http://www.fxguide.com/article333.html

Motion Paint: an example use of optical flow



https://www.youtube.com/watch?v=S5S_ABFcF_4

Next Lecture: Tracking

- Readings: FP 10.6; SZ 8; TV 8
  - Global, Parametric Motion Models.