Lecture 19: Optical Flow

*This slide deck was copied wholesale from David's FA2019 442

Administrative: Homework

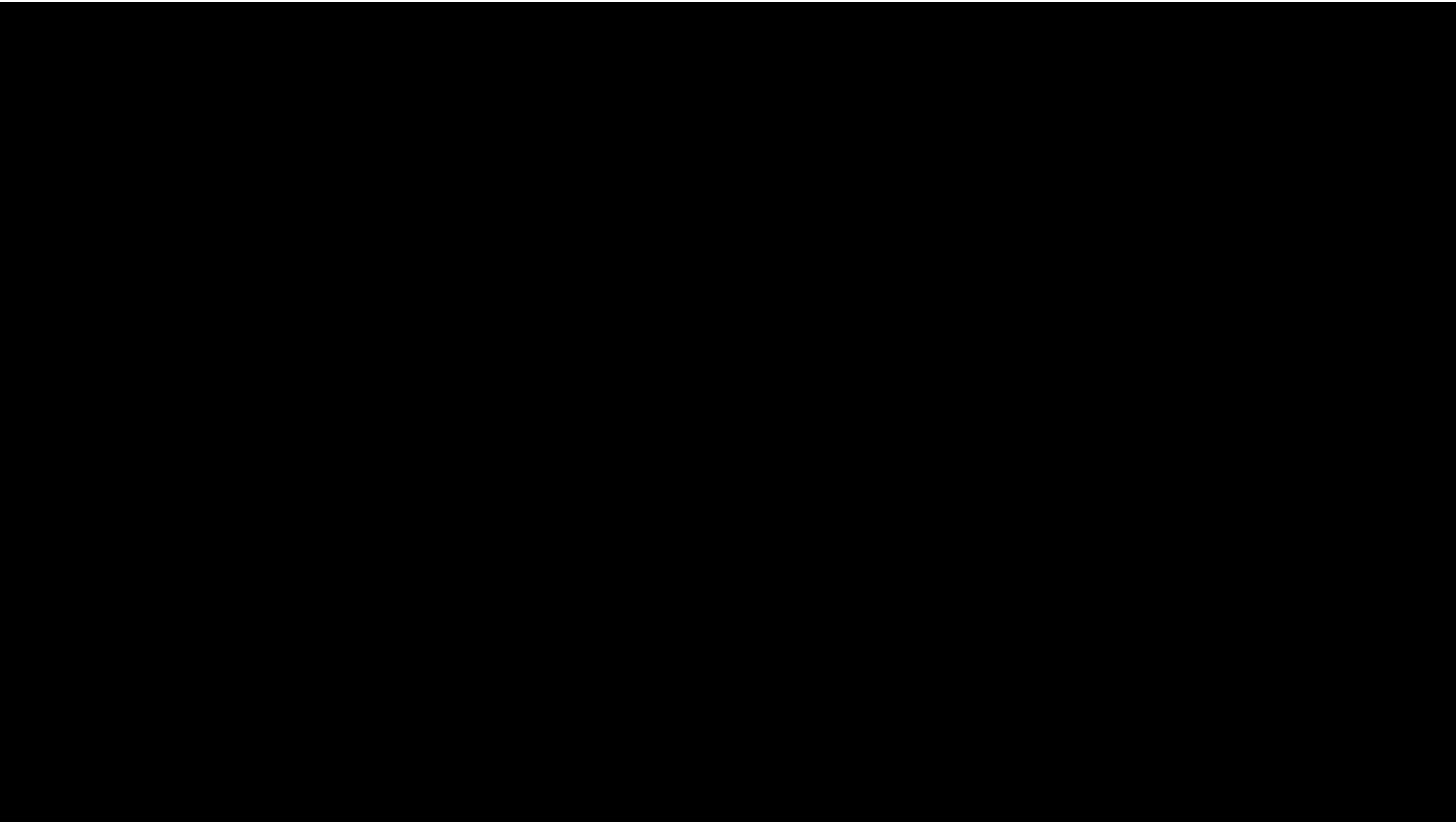
HW4 due yesterday, March 29

HW5 Released, due Friday April 9, 11:59pm ET

Administrative: Project Proposal

- Project Proposal due Monday 4/5/2021
- We've prepared six recommended projects https://docs.google.com/document/d/1a2RY4_7s7DEiyXF_qsIKZCTTAzyLaXKgTBtV8MumTfg/edit
- 3-5 People per group
- Prepare a 1 page PDF; see details on course website
- Submit once per group to Gradescope
- Also fill out Google Form to register your project: https://forms.gle/YhcEWfD9Y5cTFBa4A
- If you still need a group: Project matching form https://docs.google.com/forms/d/e/1FAIpQLSfapvf4y1kx0Yg2cCY4dxTYf-Y_cF2NR_DC74whS34CHXh-Fw/viewform

Today: Optical Flow



https://www.youtube.com/watch?v=G3QrhdfLCO8

Optical Flow

Idea first introduced by psychologist JJ Gibson in ~1940s to describe how to perceive opportunities for motion

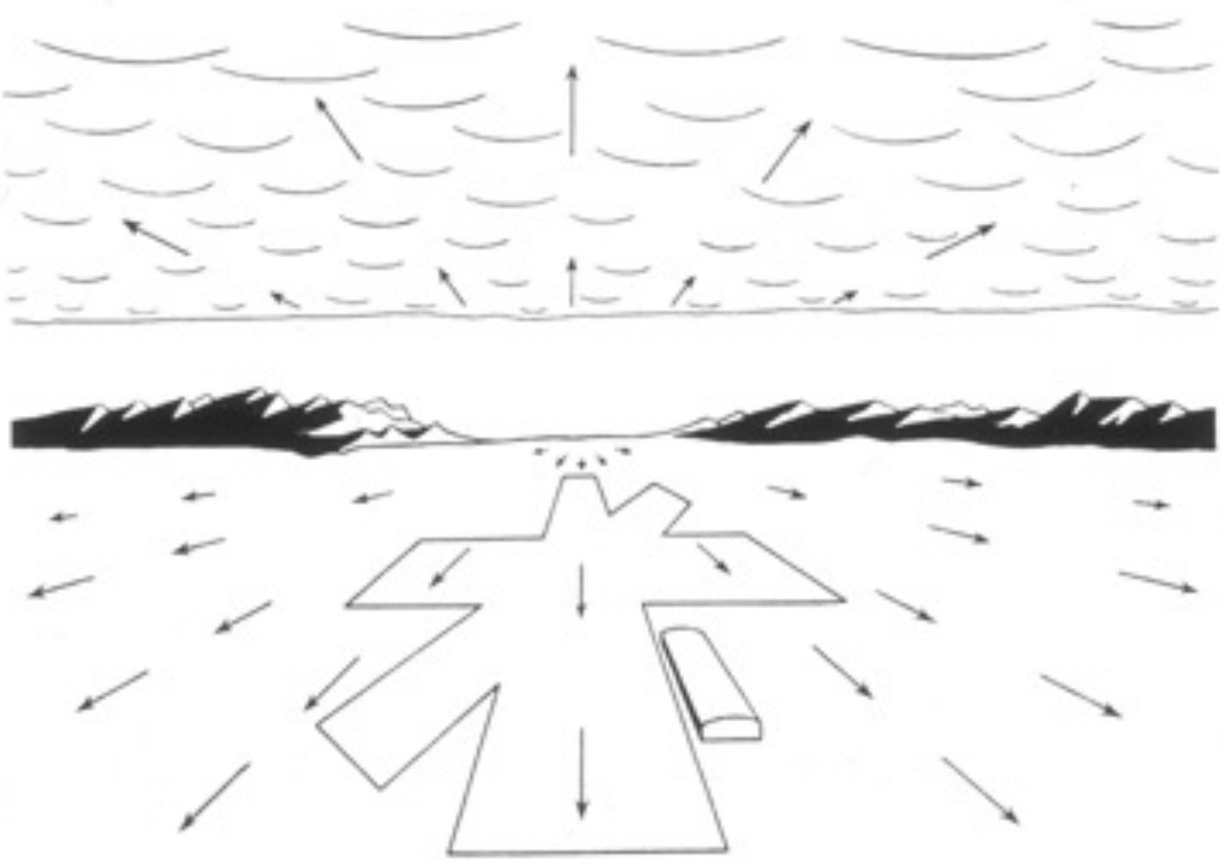
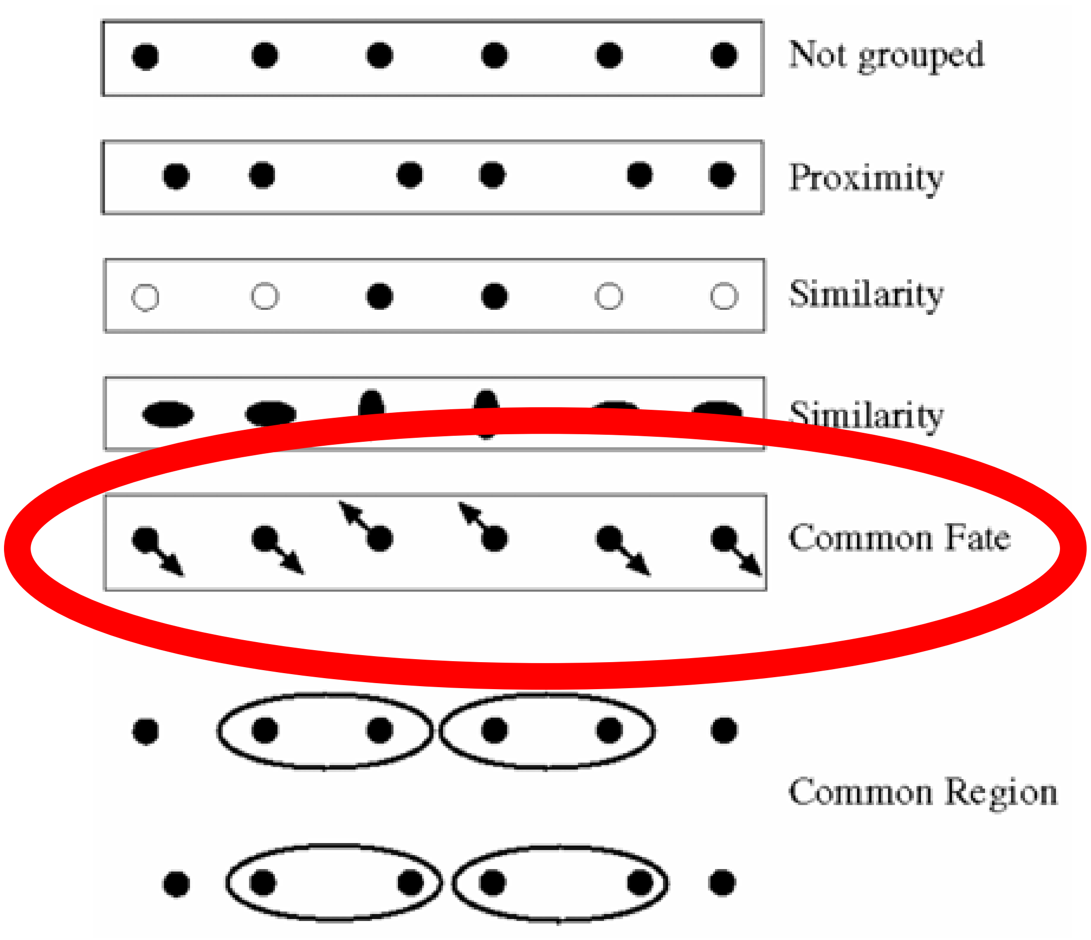


Image Credit: Gibson

Motion Perception

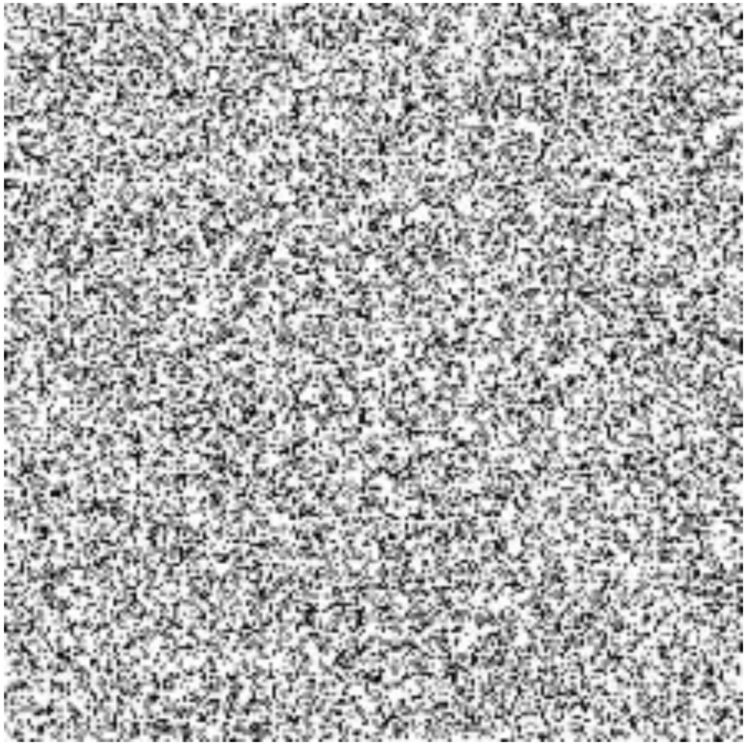


Gestalt psychology
Max Wertheimer
1880-1943

Slide Credit: S. Lazebnik

Motion and perceptual organization

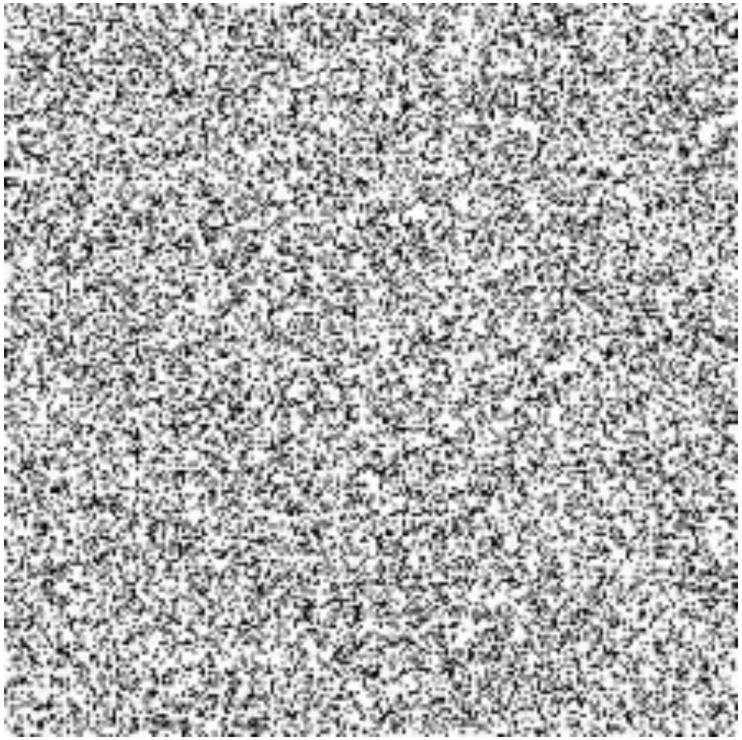
Sometimes motion is the only cue



Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Motion and perceptual organization

Sometimes motion is the only cue



Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Motion and perceptual organization

Even impoverished motion data
can create a strong percept

Slide Credit: S. Lazebnik

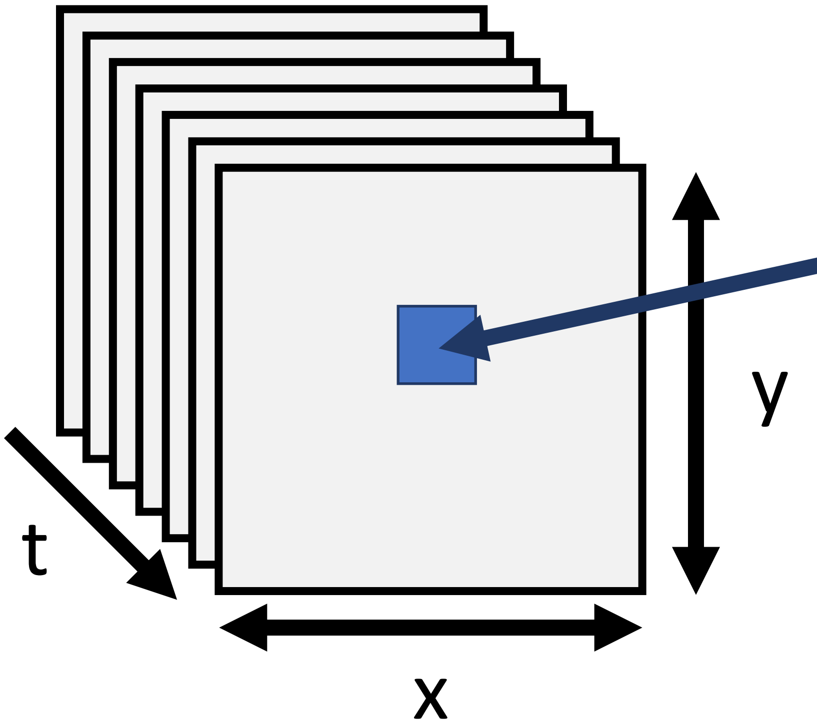
Motion and perceptual organization

Even impoverished motion data
can create a strong percept

Slide Credit: S. Lazebnik

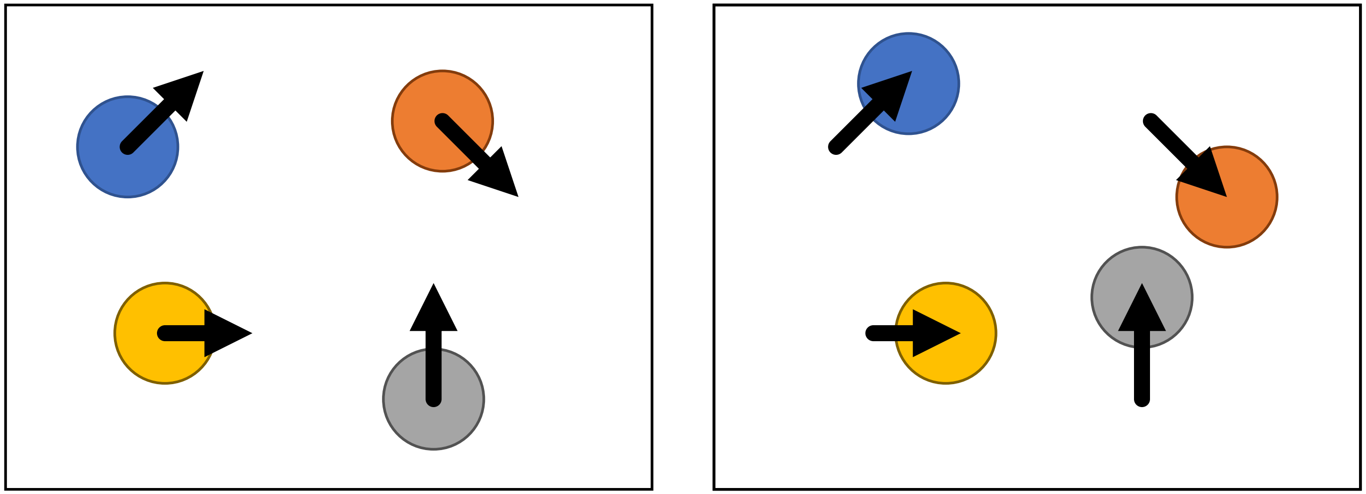
Video

Video: sequence of frames over time
Image is function of space (x,y) and time t
(and channel c)



I(x,y,c,t)
x,y – location
c – channel
t – time

Problem Definition: Optical Flow



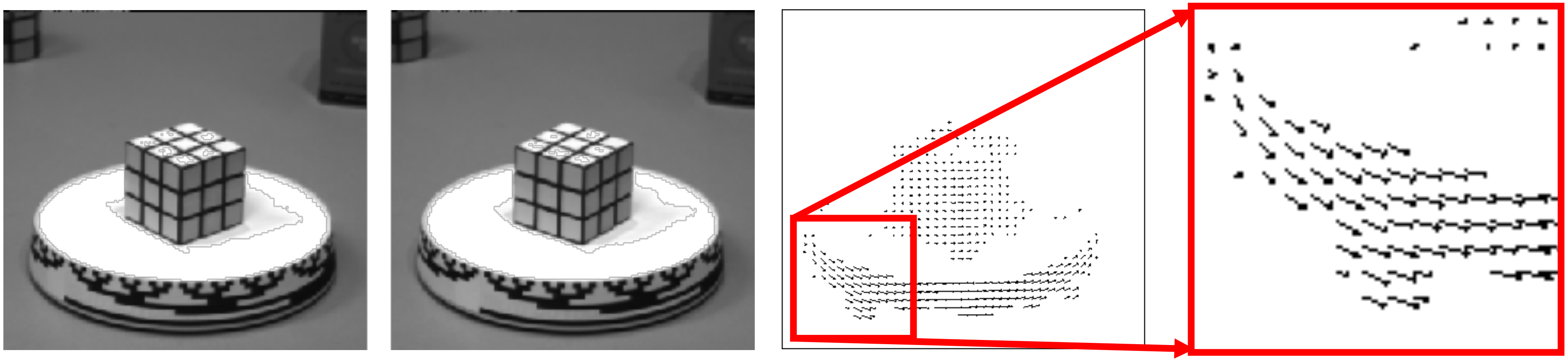
I(x,y,t) I(x,y,t+1)

Want to estimate pixel motion from image I(x,y,t) to image I(x,y,t+1)

Optical flow

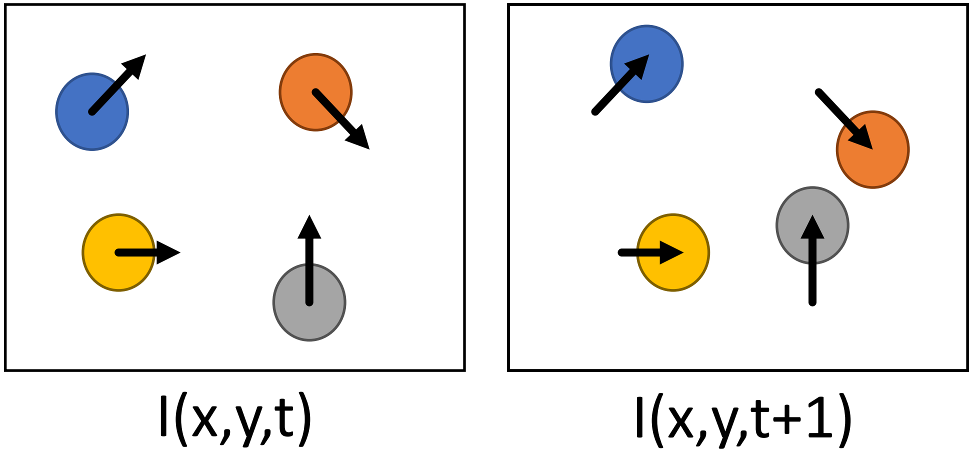
Optical flow is the *apparent* motion of objects

May be different from *actual* motion: Imagine a moving shadow on a stationary object



Will start by estimating motion of each pixel separately
Then will consider motion of entire image

Optical Flow

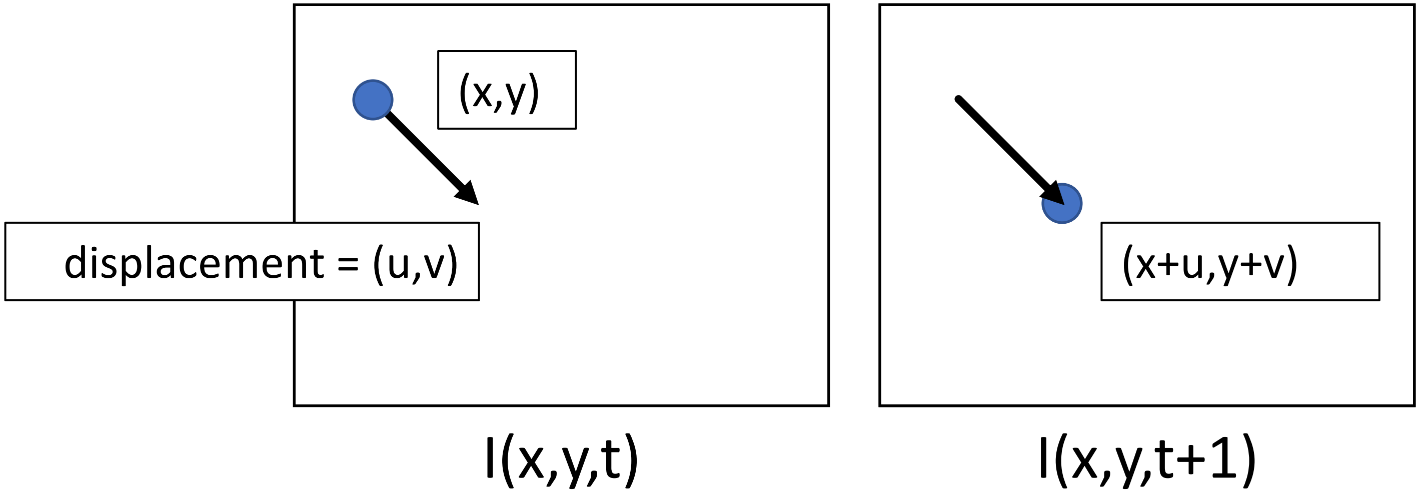


Solve correspondence problem: given pixel at time t, find **nearby** pixels of the **same color** at time t+1

Key assumptions:

- **Color/brightness constancy**: point at time t looks same at time t+1
- **Small motion**: points do not move very far

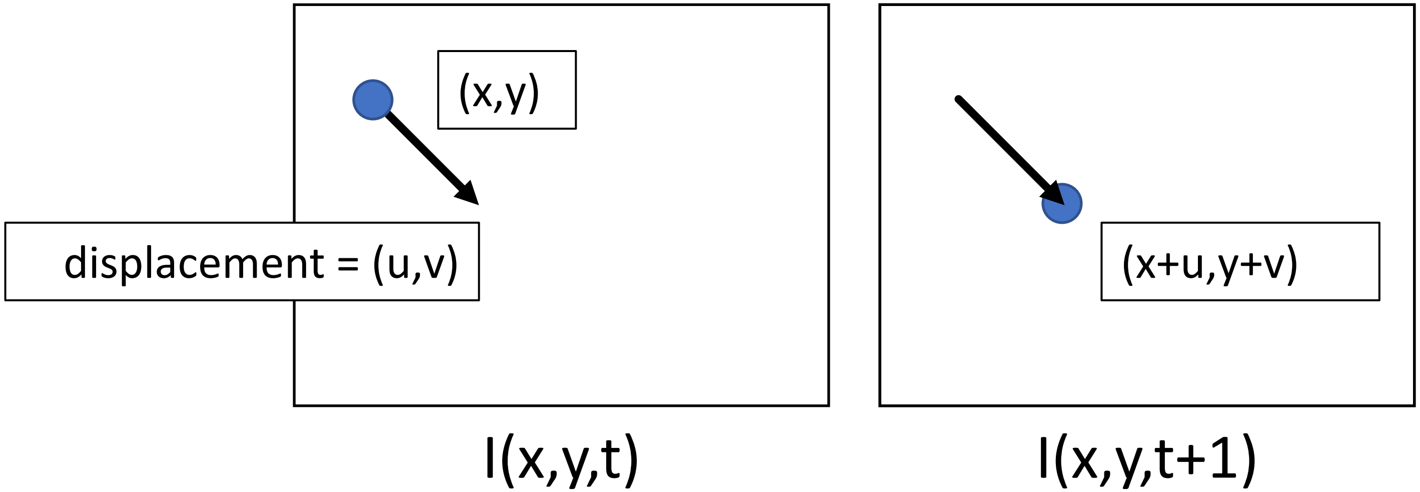
Optical Flow



Brightness constancy: $$I(x, y, t) = I(x + u, y + v, t + 1)$$

Wrong way to do things: brute force match

Optical Flow



Brightness constancy: $$I(x, y, t) = I(x + u, y + v, t + 1)$$

Recall Taylor Expansion: $$I(x + u, y + v, t) = I(x, y, t) + I_x u + I_y v + \cdots$$

Optical Flow Equation

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

$$\begin{aligned} 0 &\approx I(x+u, y+v, t+1) - I(x,y,t) \\ &= I(x,y,t+1) + I_x u + I_y v - I(x,y,t) \\ &= \underbrace{I(x,y,t+1) - I(x,y,t)}_{} + I_x u + I_y v \end{aligned}$$

Taylor Expansion

If you had to guess, what would you call this?

Adapted from S. Lazebnik slides

Optical Flow Equation

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

$$\begin{aligned} 0 &\approx I(x+u, y+v, t+1) - I(x, y, t) \\ &= I(x, y, t+1) + I_x u + I_y v - I(x, y, t) \\ &= I(x, y, t+1) - I(x, y, t) + I_x u + I_y v \\ &= I_t + I_x u + I_y v \\ &= I_t + \nabla I \cdot [u, v] \end{aligned}$$

Taylor Expansion

When is this approximation exact?

[u,v] = [0,0]

When is it bad?

u or v big.

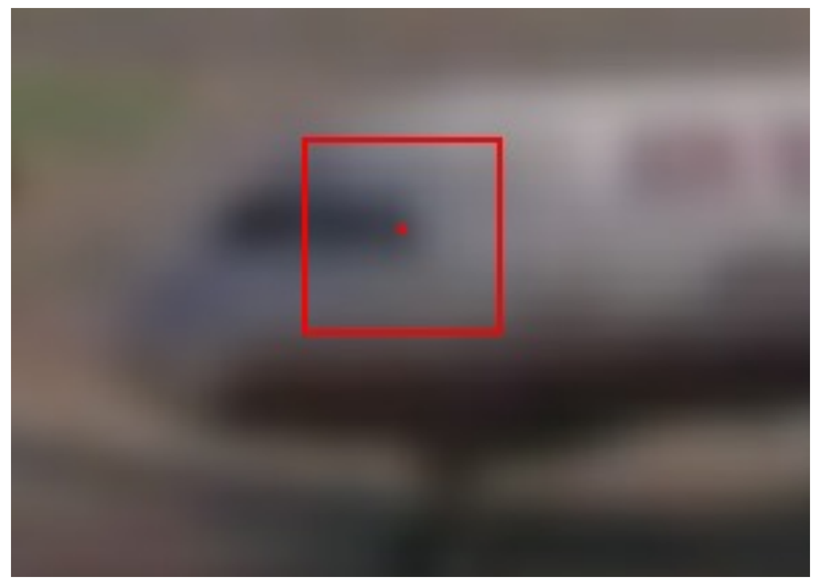
Adapted from S. Lazebnik slides

Optical Flow Equation

Brightness constancy equation

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

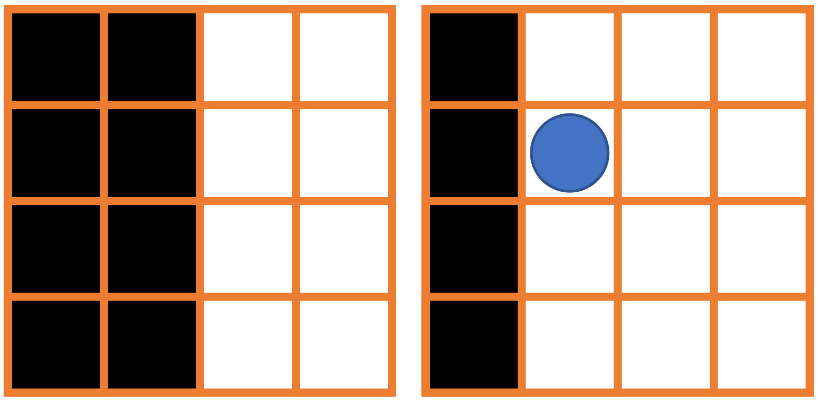
What do static image gradients have to do with motion estimation?



Slide Credit: S. Lazebnik

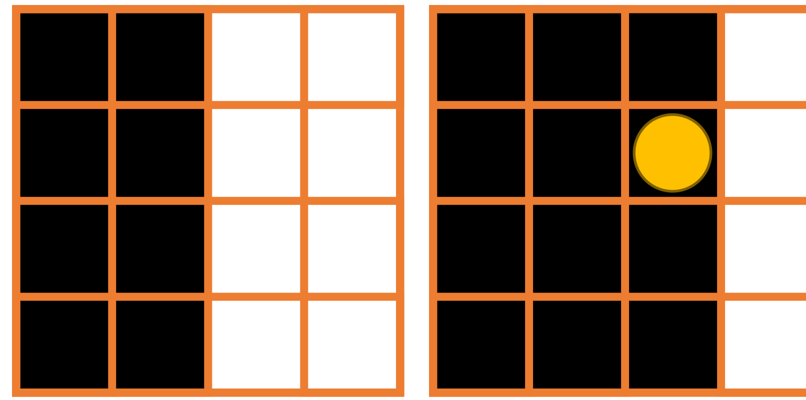
Brightness Constancy Example

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



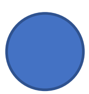
t

t+1



t

t+1

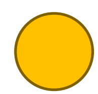
@ 

It = 1-0 = 1

Iy = 0

Ix = 1-0 = 1

What's u?

@ 

It = 0-1 = -1

Iy = 0

Ix = 1-0 = 1

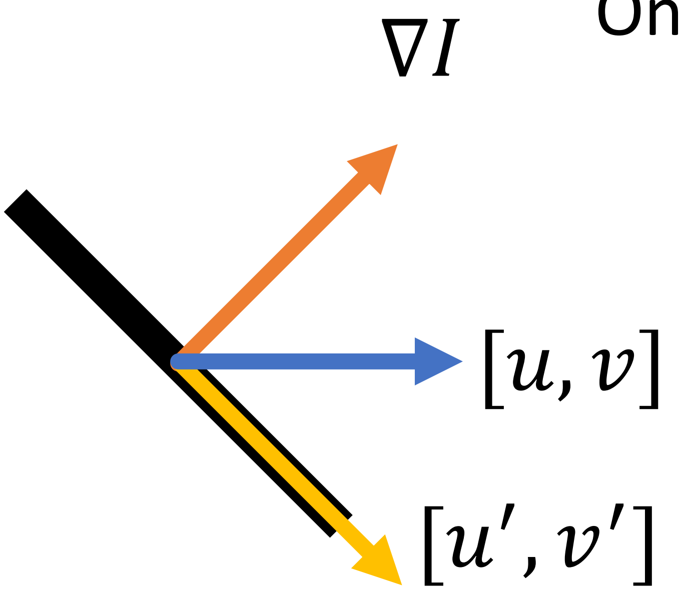
What's u?

Optical Flow Equation

Have: $I_x u + I_y v + I_t = 0 \qquad I_t + \nabla I \cdot [u, v] = 0$

How many equations and unknowns per pixel?

1 (single equation), 2 (u and v)



One nasty problem:

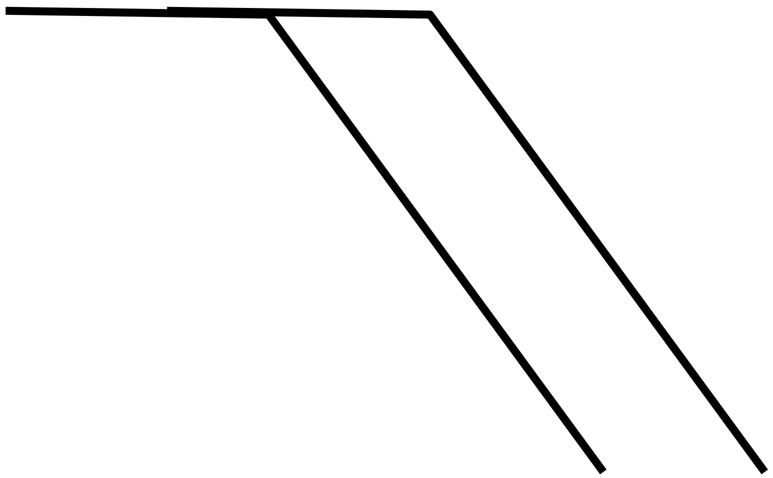
Suppose $\nabla I^T [u', v'] = 0$

$\mathrm{I_t} + \nabla I^T [u + u', v + v'] = 0$

Can only identify the motion along gradient and **not** motion perpendicular to it

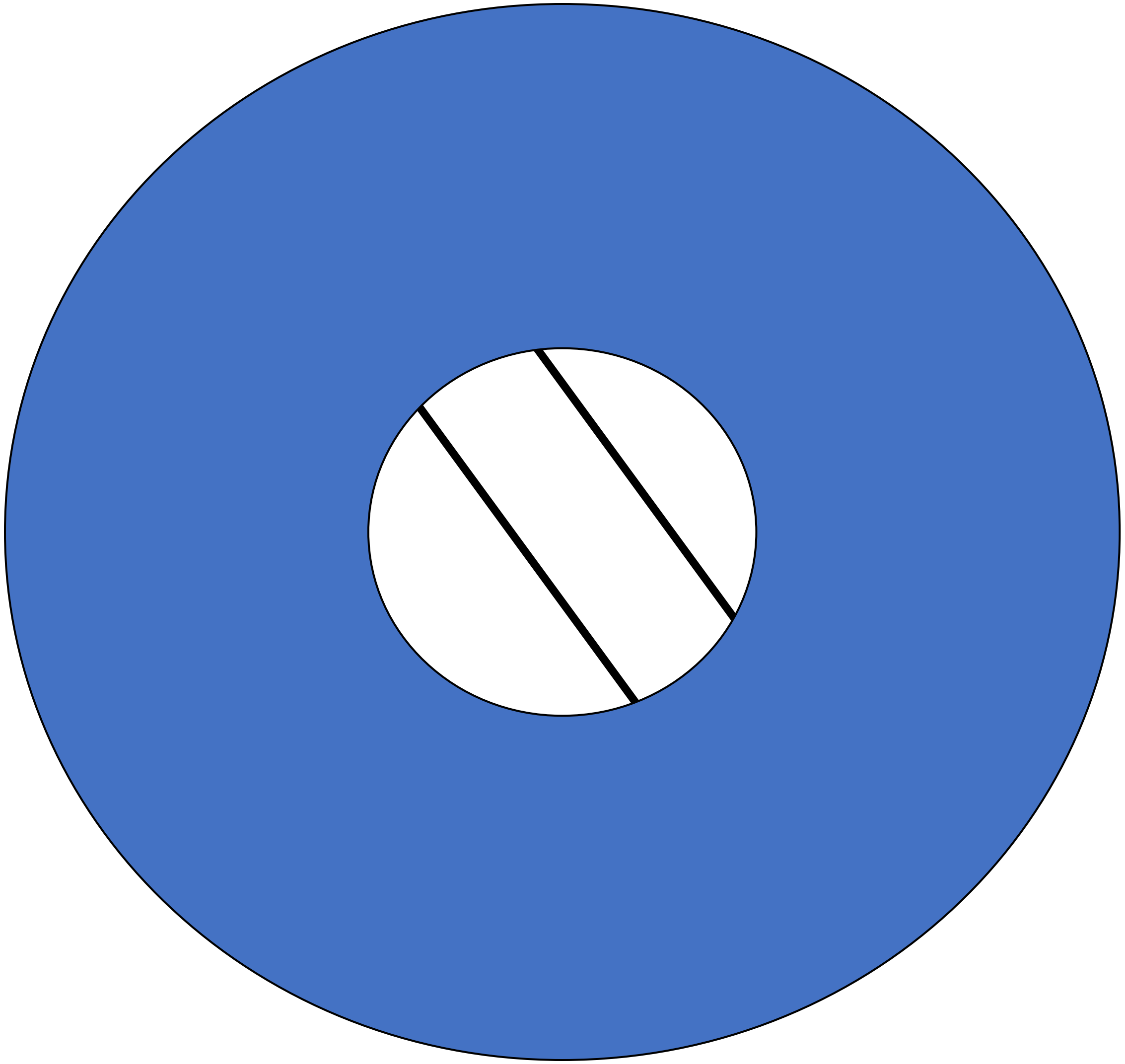
Adapted from S. Lazebnik slides

Aperture problem



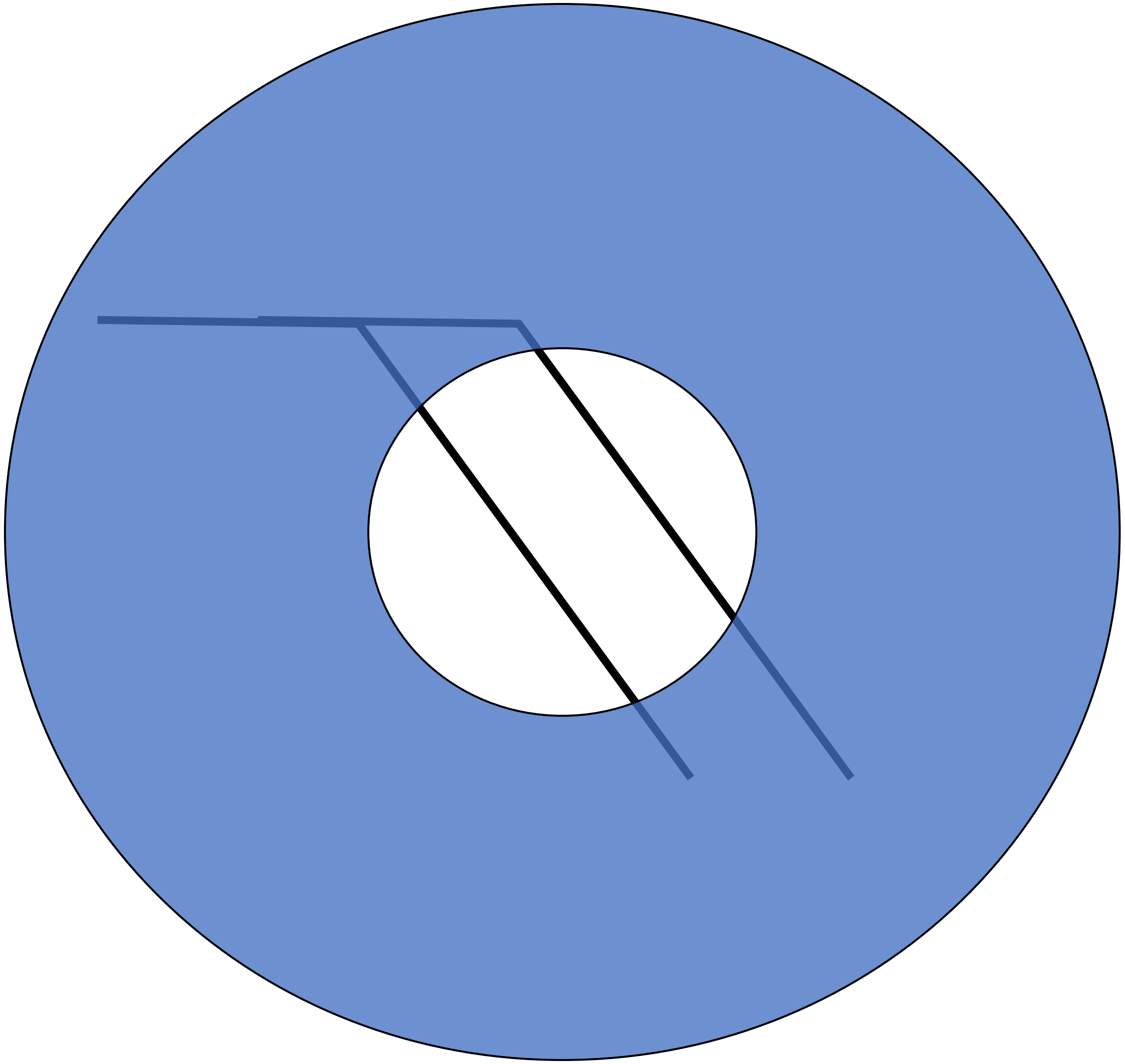
Slide Credit: S. Lazebnik

Aperture problem



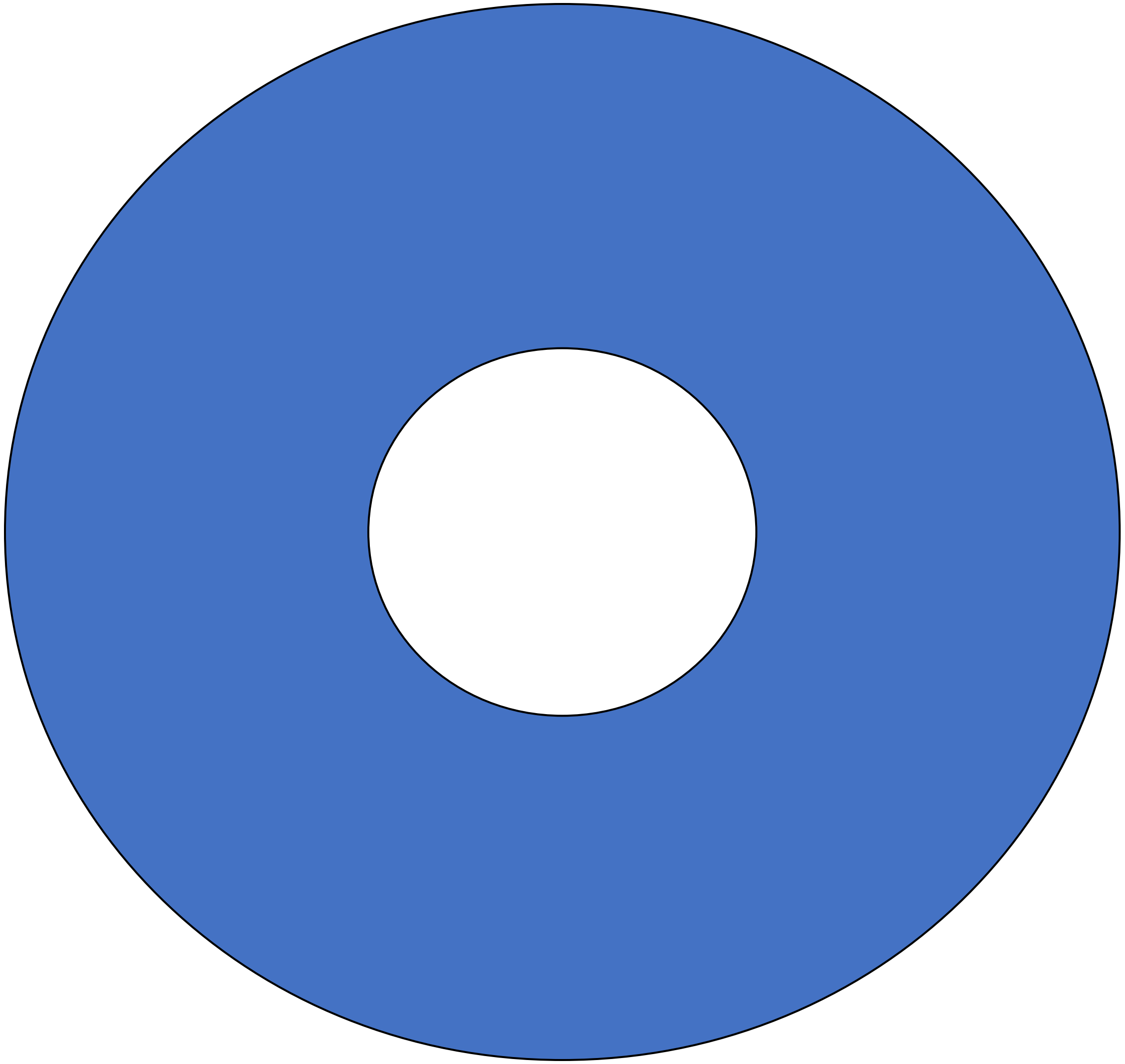
Slide Credit: S. Lazebnik

Aperture problem

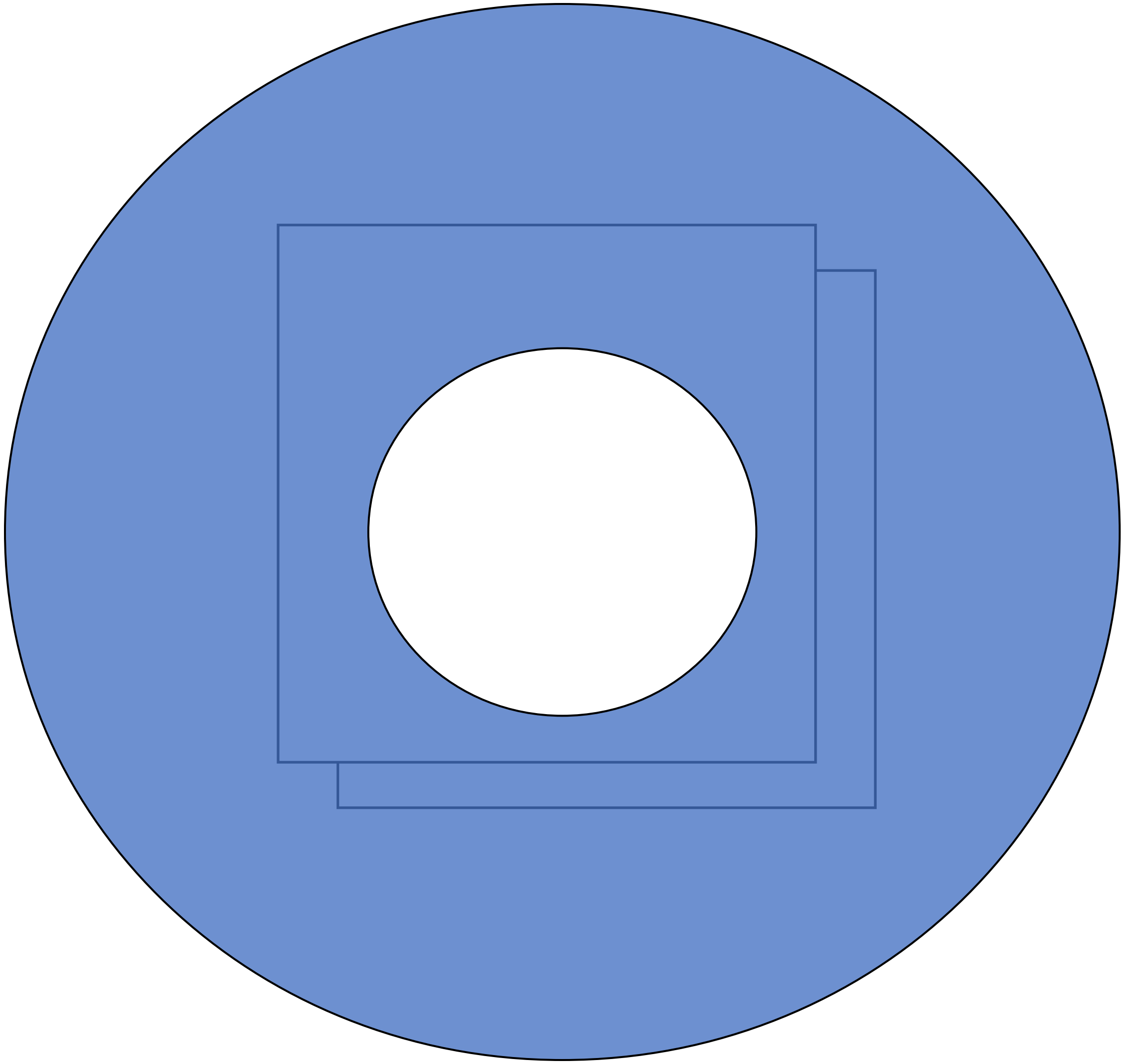


Slide Credit: S. Lazebnik

Other Invisible Flow



Other Invisible Flow



Solving Ambiguity – Lucas Kanade

2 unknowns [u,v], 1 eqn per pixel

How do we get more equations?

Assume *spatial coherence*: pixel's neighbors have *move together* / have same [u,v]

5x5 window gives 25 new equations

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

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

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.

Solving for u, v

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_t(p_1) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \qquad \underset{25x2}{\boldsymbol{A}} \; \underset{2x1}{\boldsymbol{d}} = \underset{25x1}{\boldsymbol{b}}$$

What's the solution?

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{d} = \boldsymbol{A}^T\boldsymbol{b} \quad \rightarrow \quad \boldsymbol{d} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$$

Intuitively, need to solve (sum over pixels in window)

$$\underbrace{\begin{bmatrix} \sum \boldsymbol{I}_x\boldsymbol{I}_x & \sum \boldsymbol{I}_x\boldsymbol{I}_y \\ \sum \boldsymbol{I}_x\boldsymbol{I}_y & \sum \boldsymbol{I}_y\boldsymbol{I}_y \end{bmatrix}}_{\boldsymbol{A}^T\boldsymbol{A}} \begin{bmatrix} u \\ v \end{bmatrix} = -\underbrace{\begin{bmatrix} \sum \boldsymbol{I}_x\boldsymbol{I}_t \\ \sum \boldsymbol{I}_y\boldsymbol{I}_t \end{bmatrix}}_{\boldsymbol{A}^T\boldsymbol{b}}$$

Adapted from S. Lazebnik slides

Solving for [u,v]

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

$$A^T A \qquad\qquad A^T b$$

What does this remind you of?

Harris corner detection!

When can we find [u,v]?

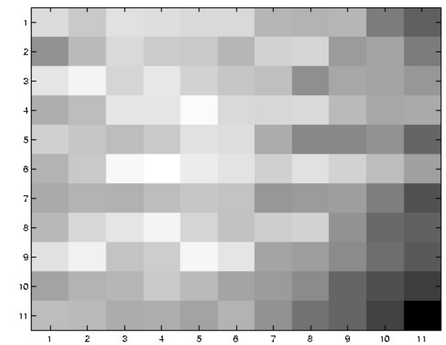
$A^T A$ invertible: precisely equal brightness isn't

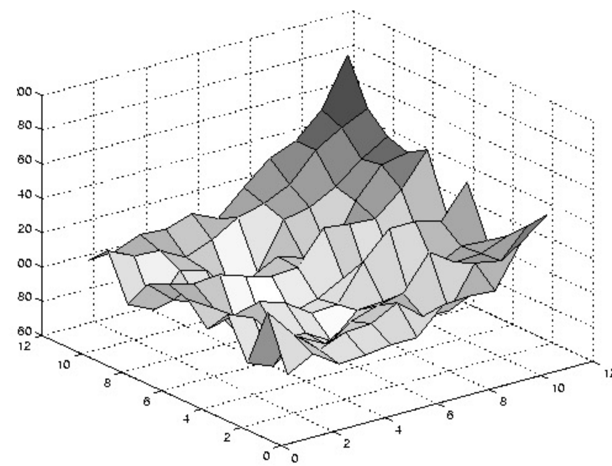
$A^T A$ not too small: noise + equal brightness

$A^T A$ well-conditioned: $|\lambda_1| / |\lambda_2|$ not large (edge)

Adapted from S. Lazebnik slides

Low texture region





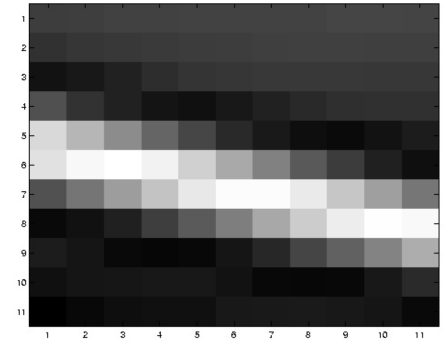
$$\begin{bmatrix}\sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y\end{bmatrix} = \sum \nabla \mathrm{I} (\nabla \mathrm{I})^{\mathrm{T}}$$

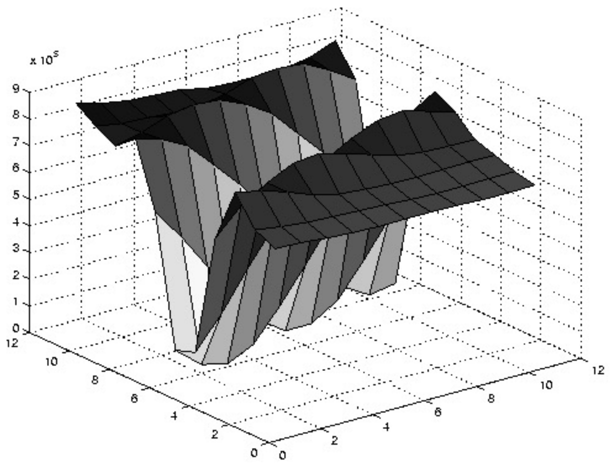
– gradients have small magnitude

– small λ_1, small λ_2

Slide Credit: S. Lazebnik

Edge





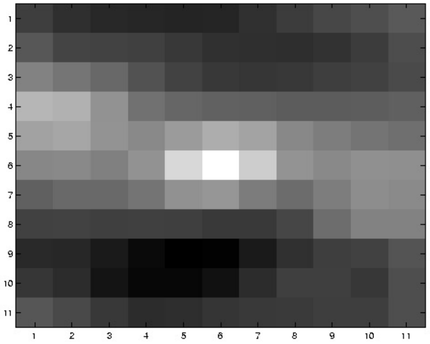
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \nabla \mathrm{I} (\nabla \mathrm{I})^{\mathrm{T}}$$

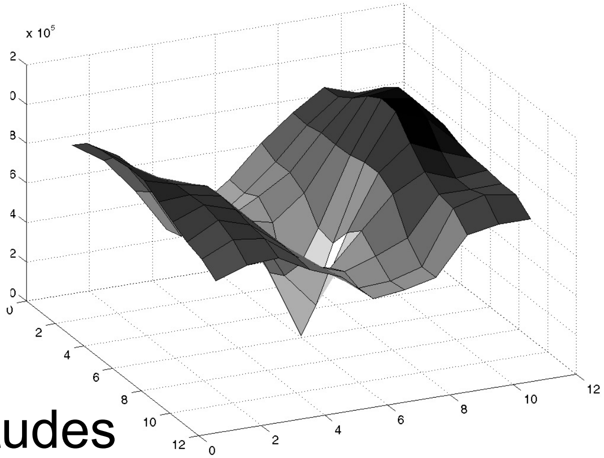
– large gradients, all the same

– large λ_1, small λ_2

Slide Credit: S. Lazebnik

High texture region





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

– gradients are different, large magnitudes

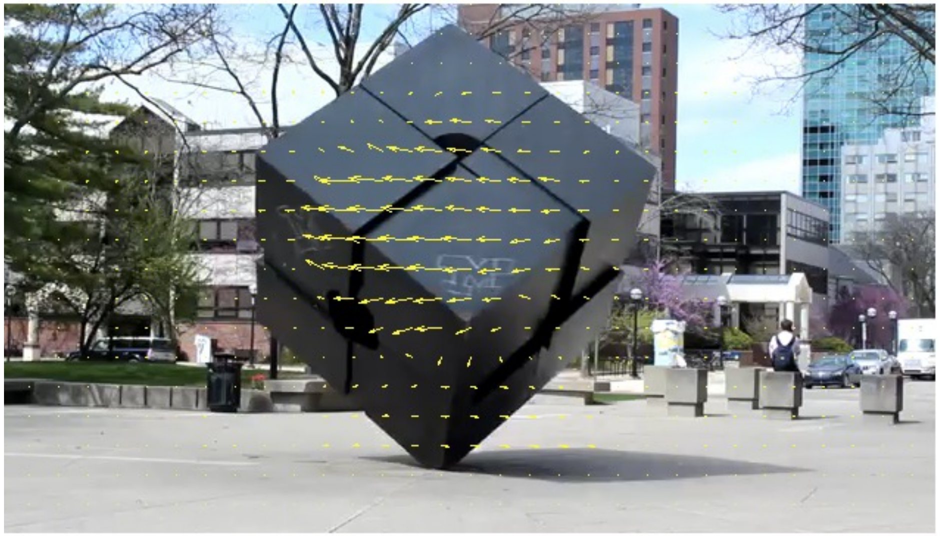
– large λ_1, large λ_2

Slide Credit: S. Lazebnik

Lucas-Kanade flow example

Input frames

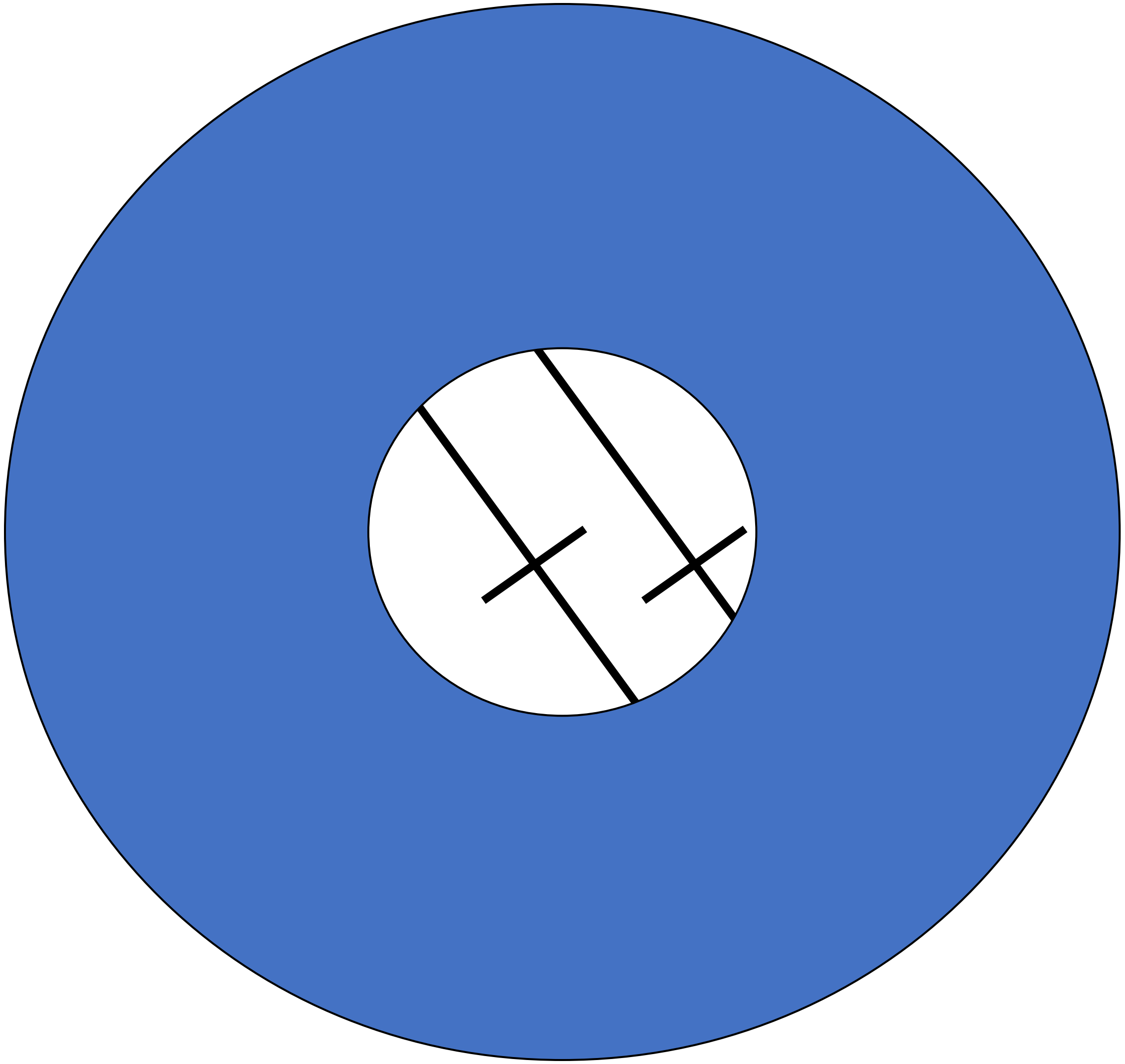
Output



Slide Credit: S. Lazebnik

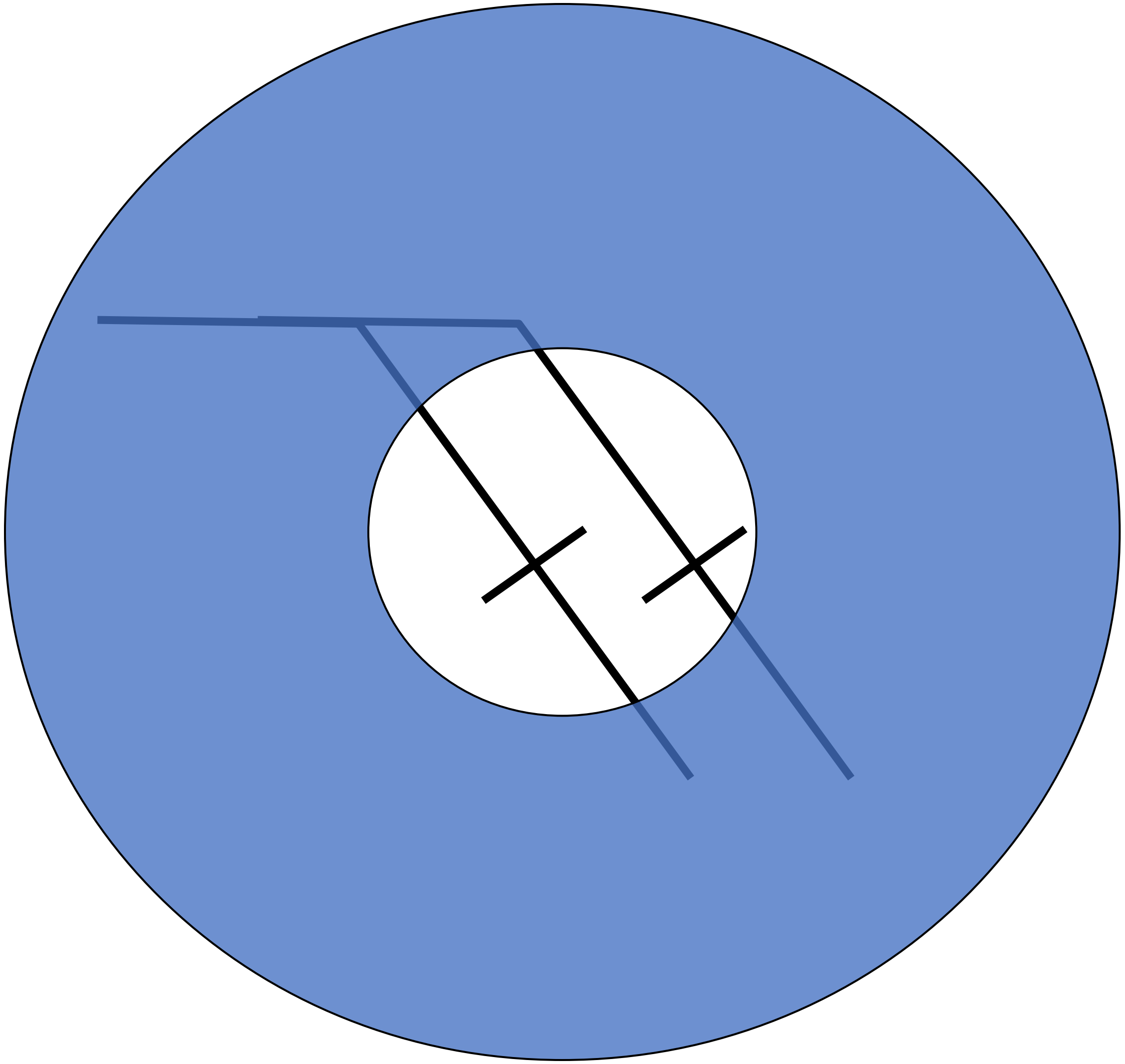
Source: [MATLAB Central File Exchange]

Aperture problem Take 2



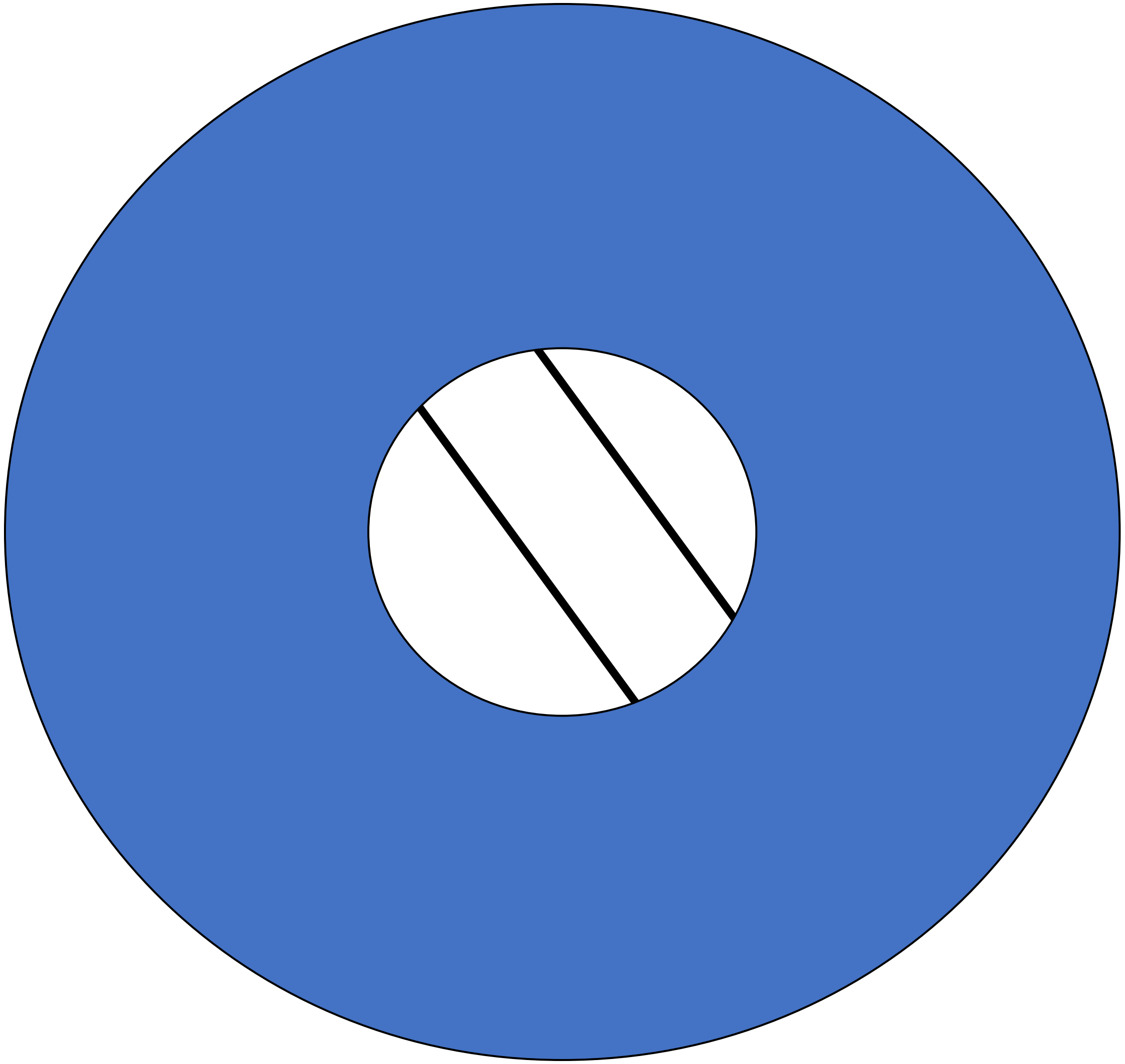
Slide Credit: S. Lazebnik

Aperture problem Take 2



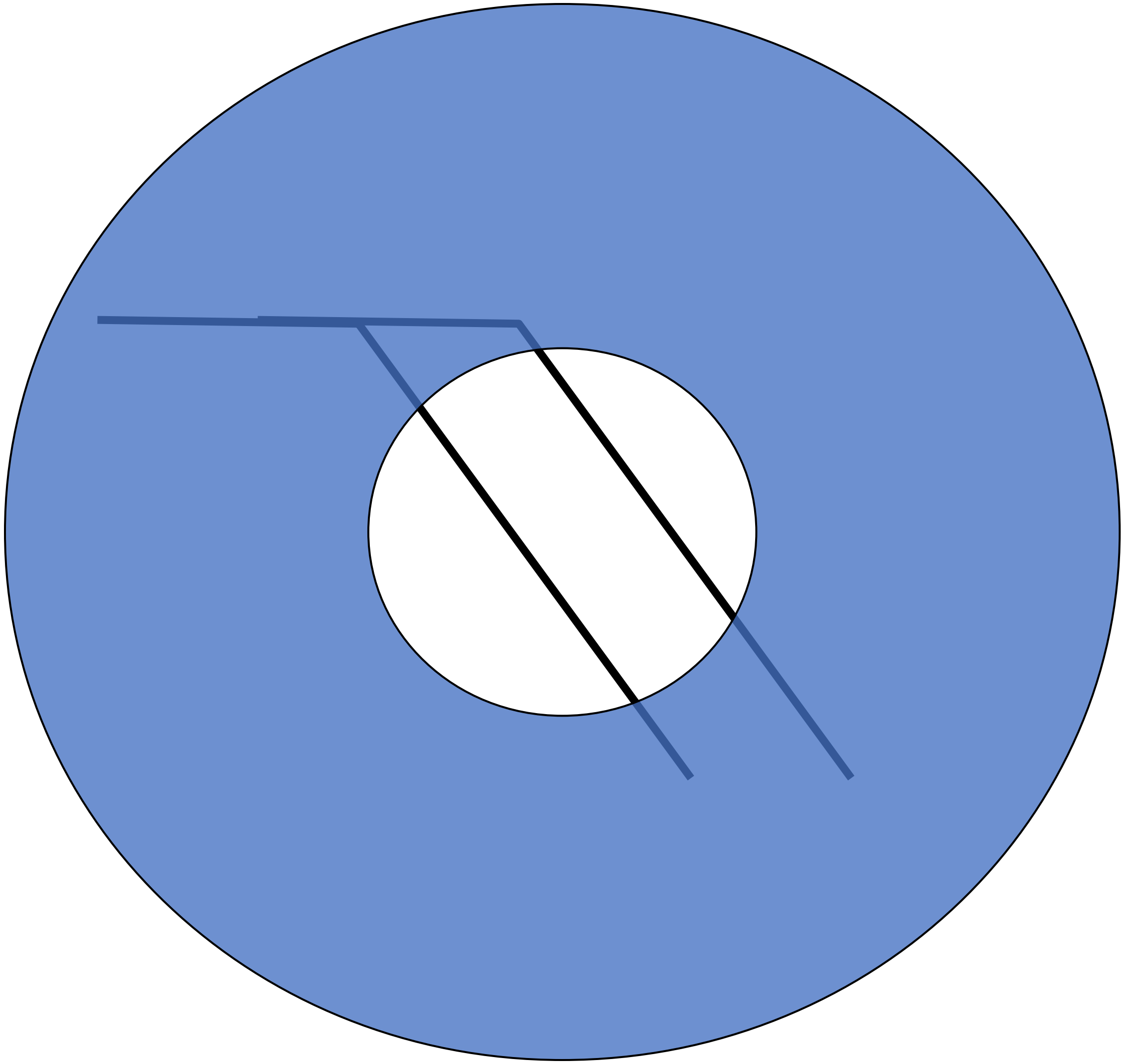
Slide Credit: S. Lazebnik

For Comparison



Slide Credit: S. Lazebnik

For Comparison



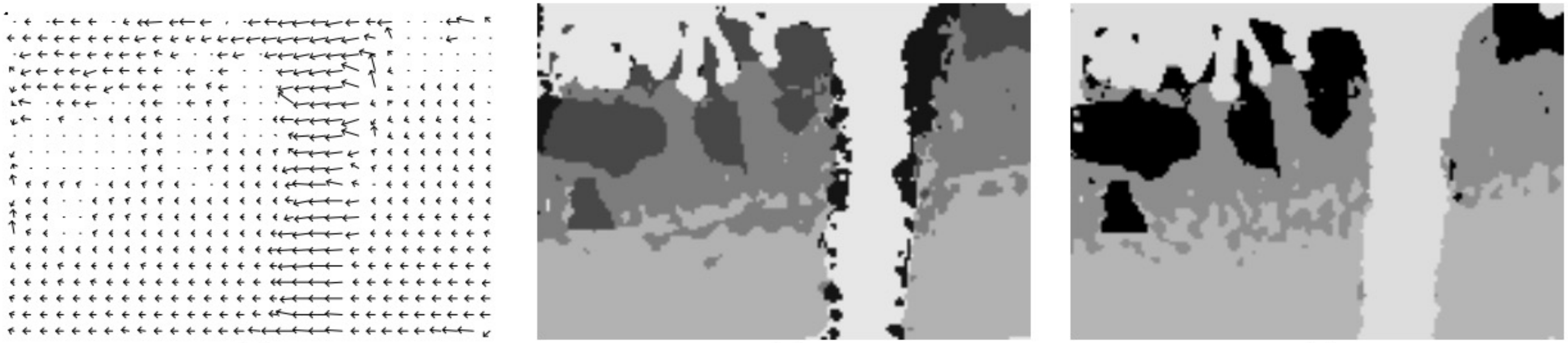
Slide Credit: S. Lazebnik

So How Does This Fail?

- Point doesn't move like neighbors:
  - **Why would this happen?**
  - Figure out which points move together, then come back and fix.

So How Does This Fail?

- Point doesn't move like neighbors:
  - **Why would this happen?**
  - Figure out which points move together, then come back and fix.



(a) (b) (c)

Figure 11: (a) The optic flow from multi-scale gradient method. (b) Segmentation obtained by clustering optic flow into affine motion regions. (c) Segmentation from consistency checking by image warping. Representing moving images with layers.

J. Wang and E. Adelson, [Representing Moving Images with Layers](), IEEE Transactions on Image Processing, 1994

So How Does This Fail?

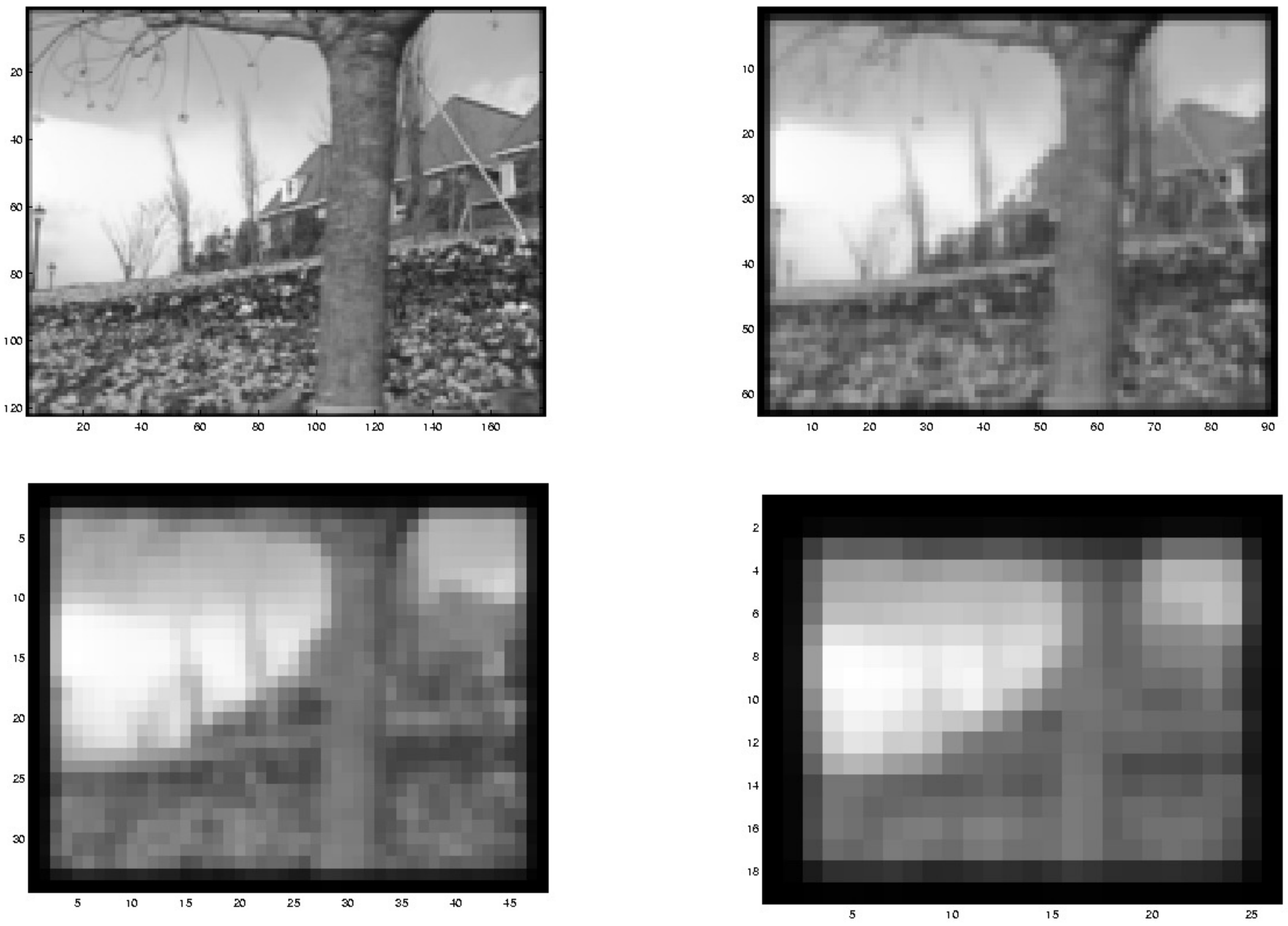
- Point doesn't move like neighbors:
  - **Why would this happen?**
  - Figure out which points move together, then come back and fix.
- Brightness constancy isn't true
  - **Why would this happen?**
  - Solution: other form of matching (e.g. SIFT)
- Taylor series is bad approximation
  - **Why would this happen?**
  - Solution: Make your pixels big

Revisiting small motions

- Is this motion small enough?
  - Probably not—it's much larger than one pixel
  - How might we solve this problem?

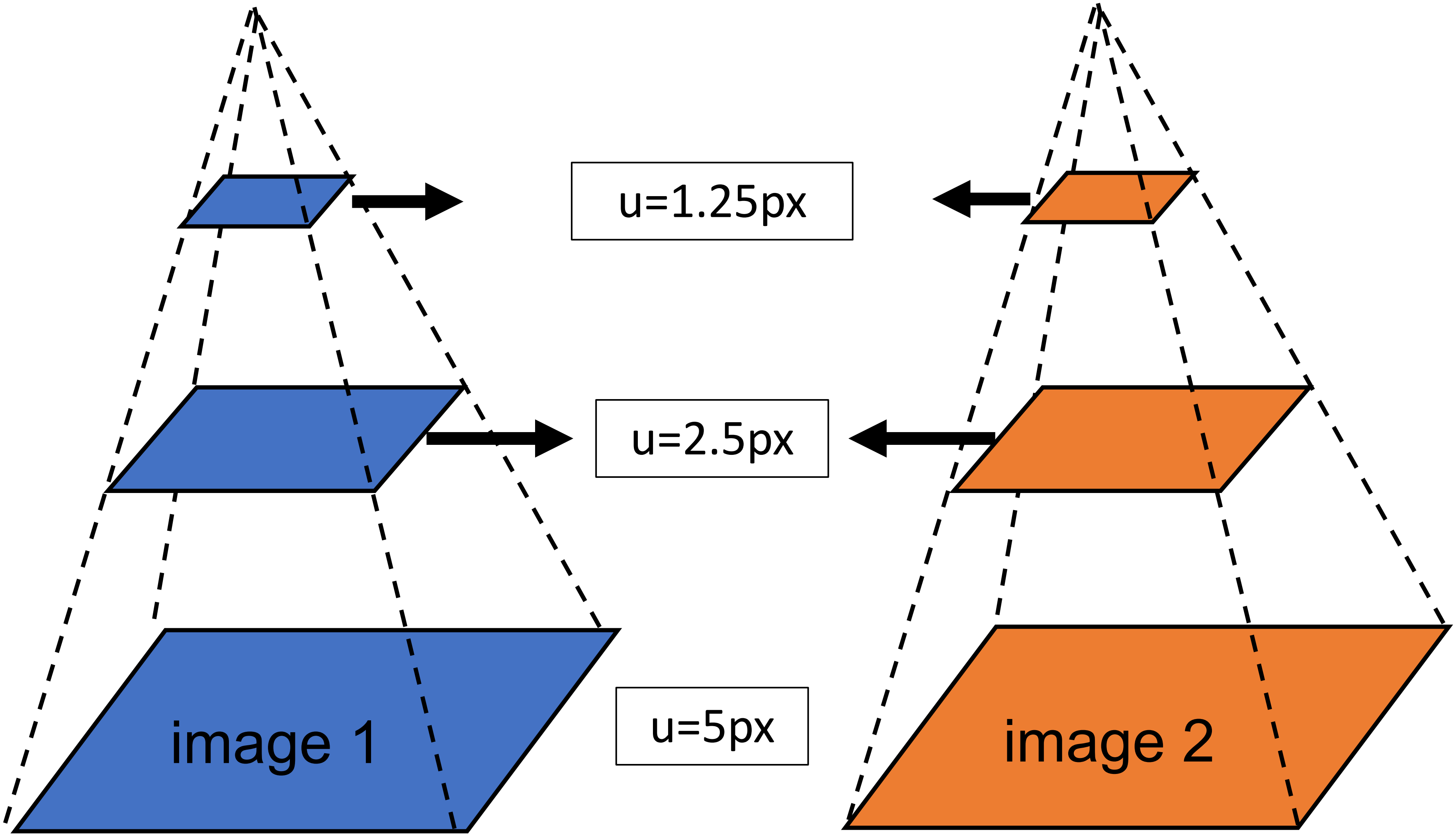
Slide credit: S. Lazebnik

Reduce the resolution!



Slide credit: S. Lazebnik

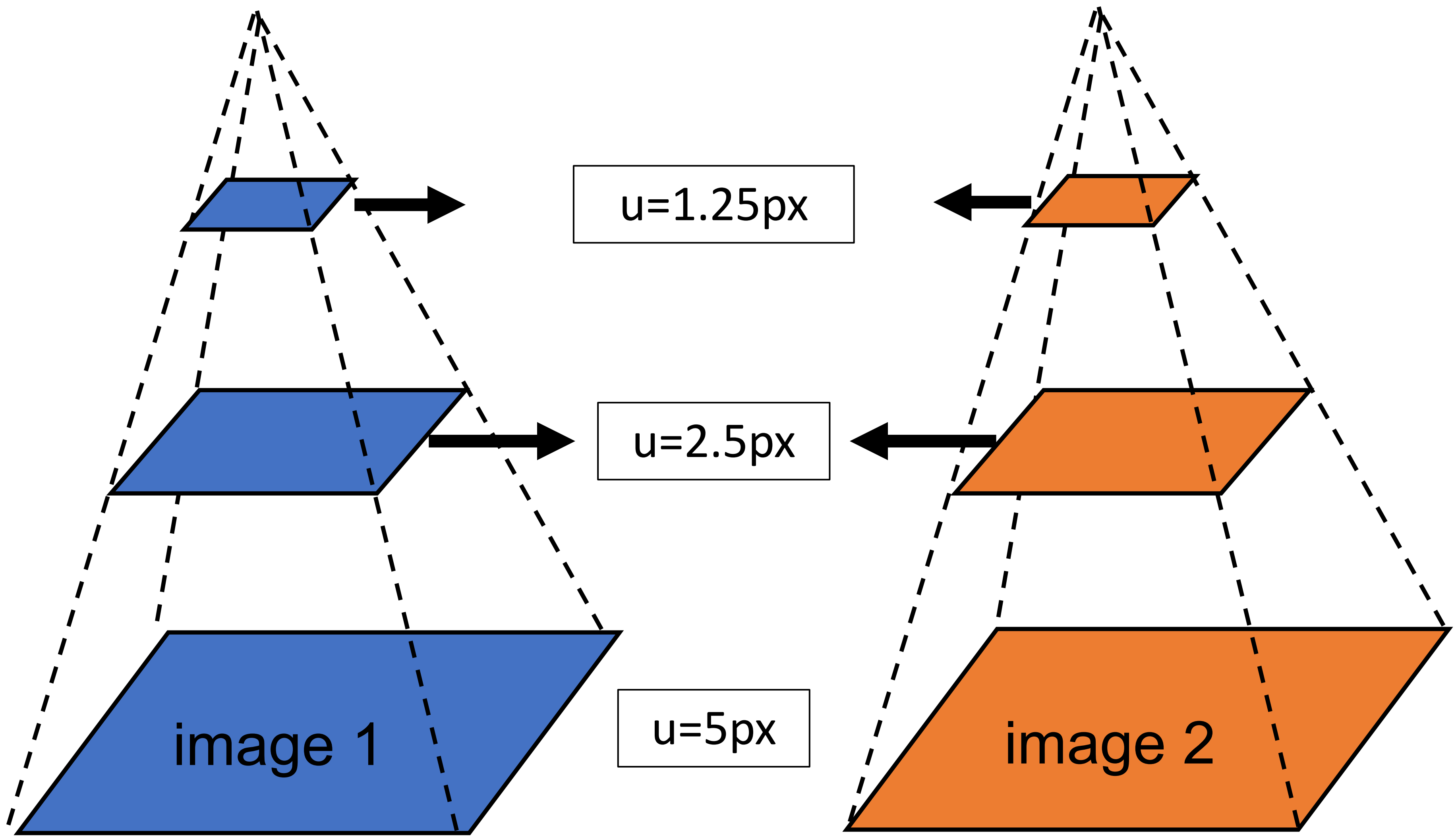
Coarse-to-fine optical flow estimation



Typically called Gaussian Pyramid

Slide credit: S. Lazebnik

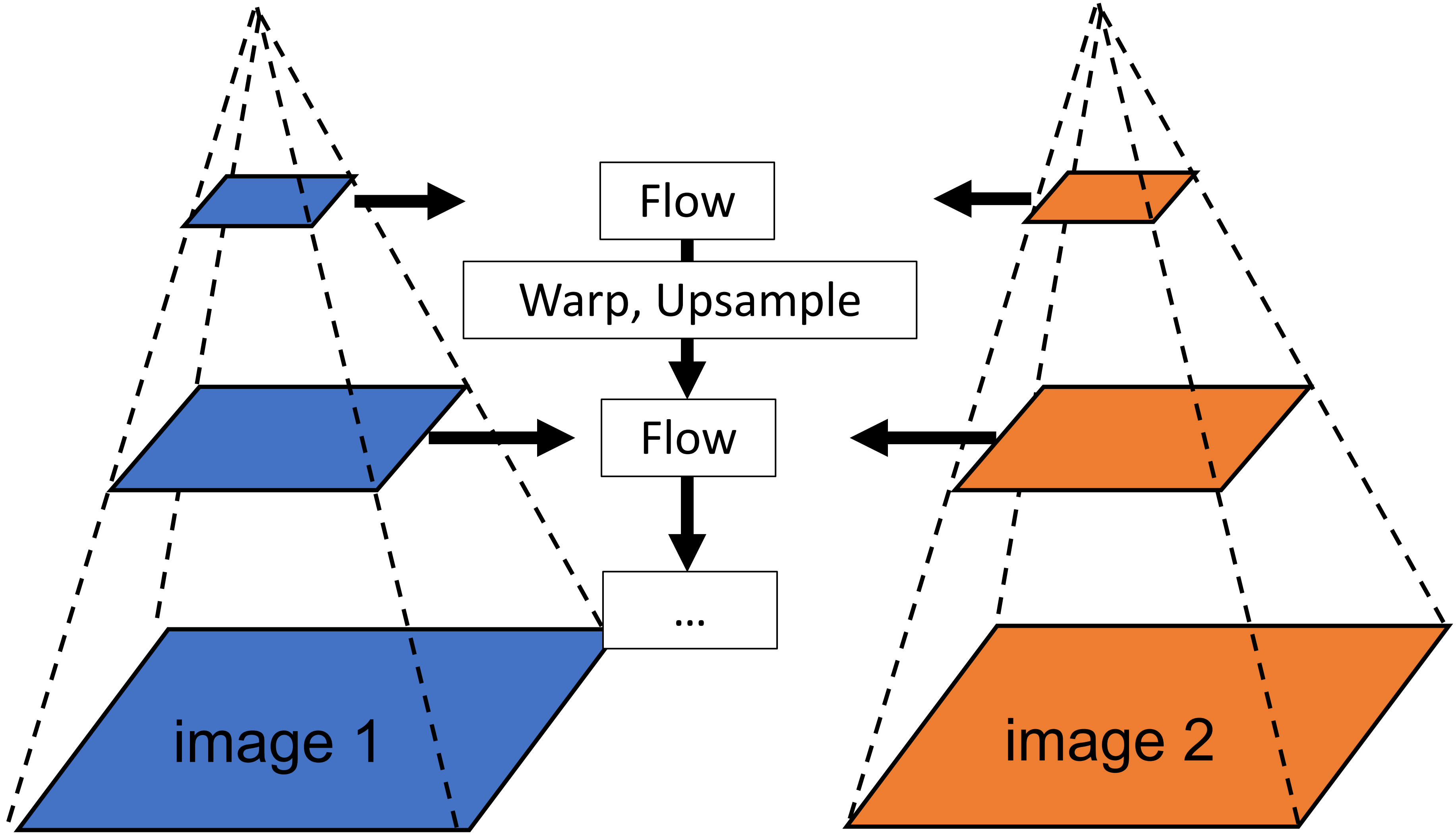
Coarse-to-fine optical flow estimation



Start at bottom or top to align?

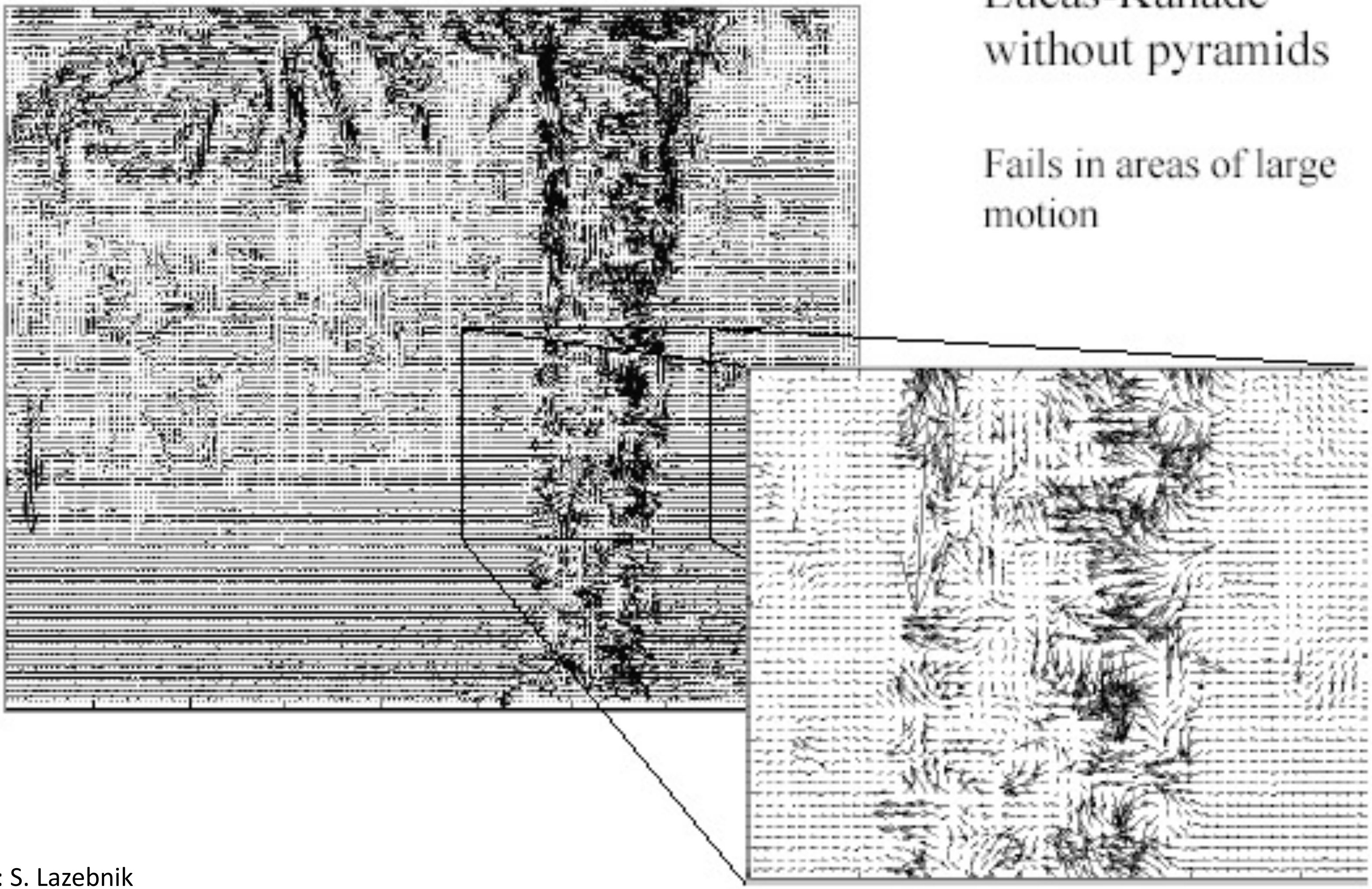
Slide credit: S. Lazebnik

Coarse-to-fine optical flow estimation



Slide credit: S. Lazebnik

Optical Flow Results

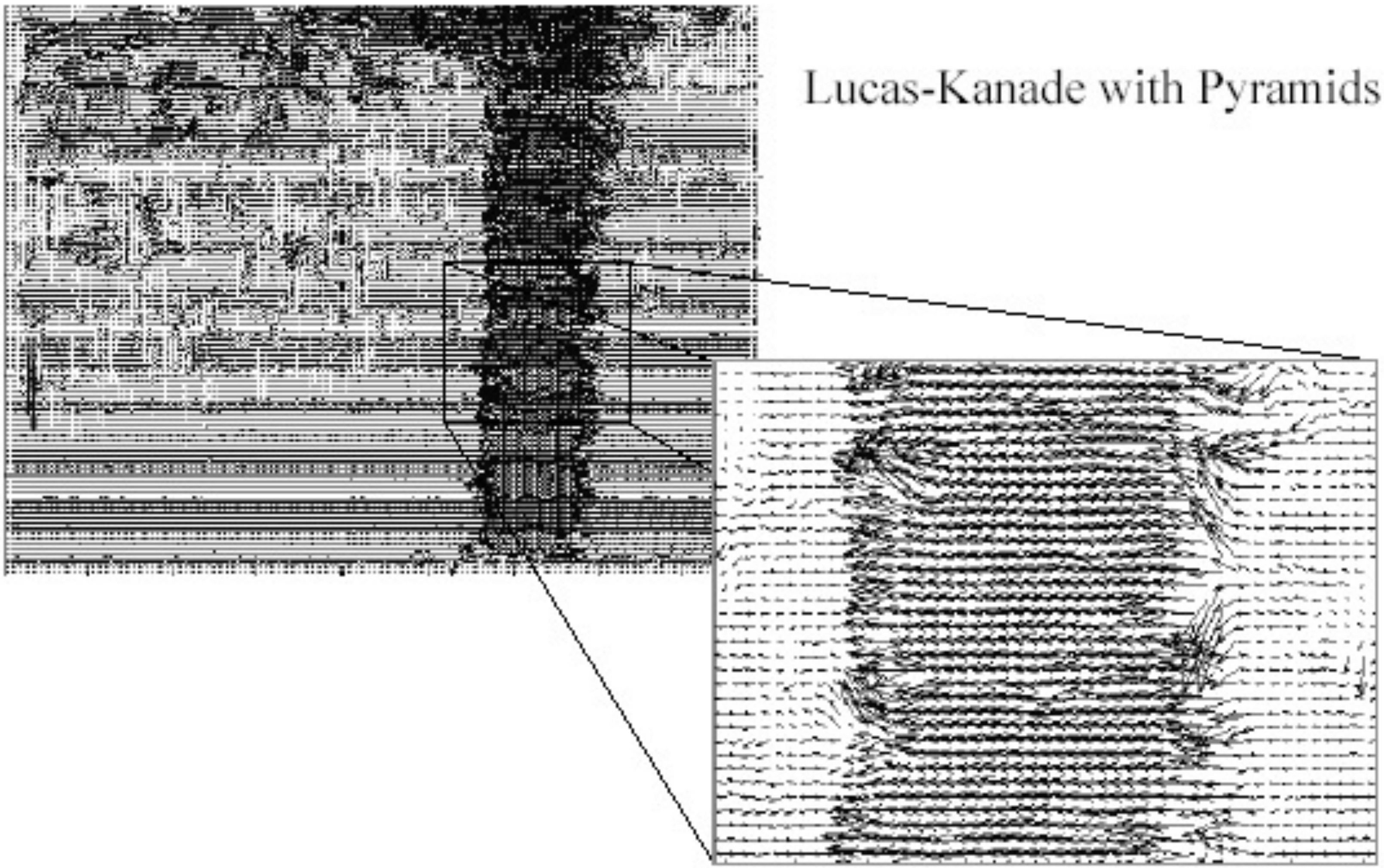


Lucas-Kanade without pyramids

Fails in areas of large motion

Slide credit: S. Lazebnik

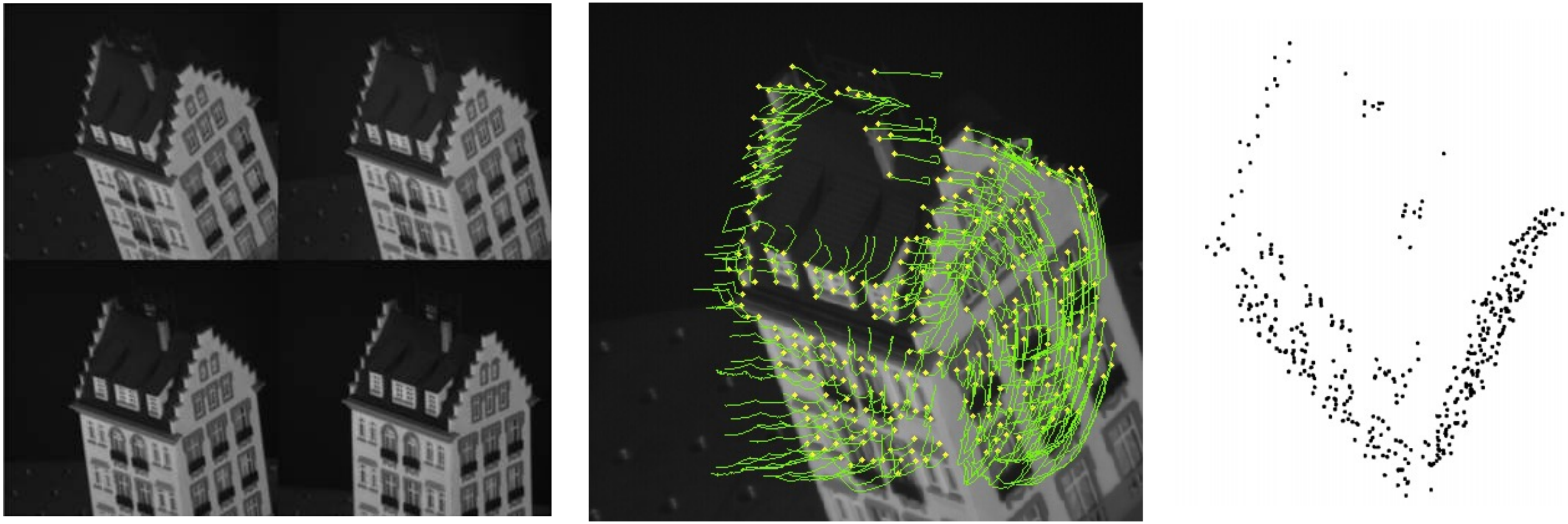
Optical Flow Results



Slide credit: S. Lazebnik

Application: Tracking

- Would like tracks of where things move (e.g., for reconstruction)



C. Tomasi and T. Kanade. [Shape and motion from image streams under orthography: A factorization method.](#) *IJCV*, 9(2):137-154, November 1992.

Application: Tracking

- Which features should we track?
  - Use eigenvalues of $A^T A$ to find corners
- Use flow to figure out [u,v] for each "track"
- Register points to first frame by affine warp

J. Shi and C. Tomasi. Good Features to Track. CVPR 1994.

Tracking example

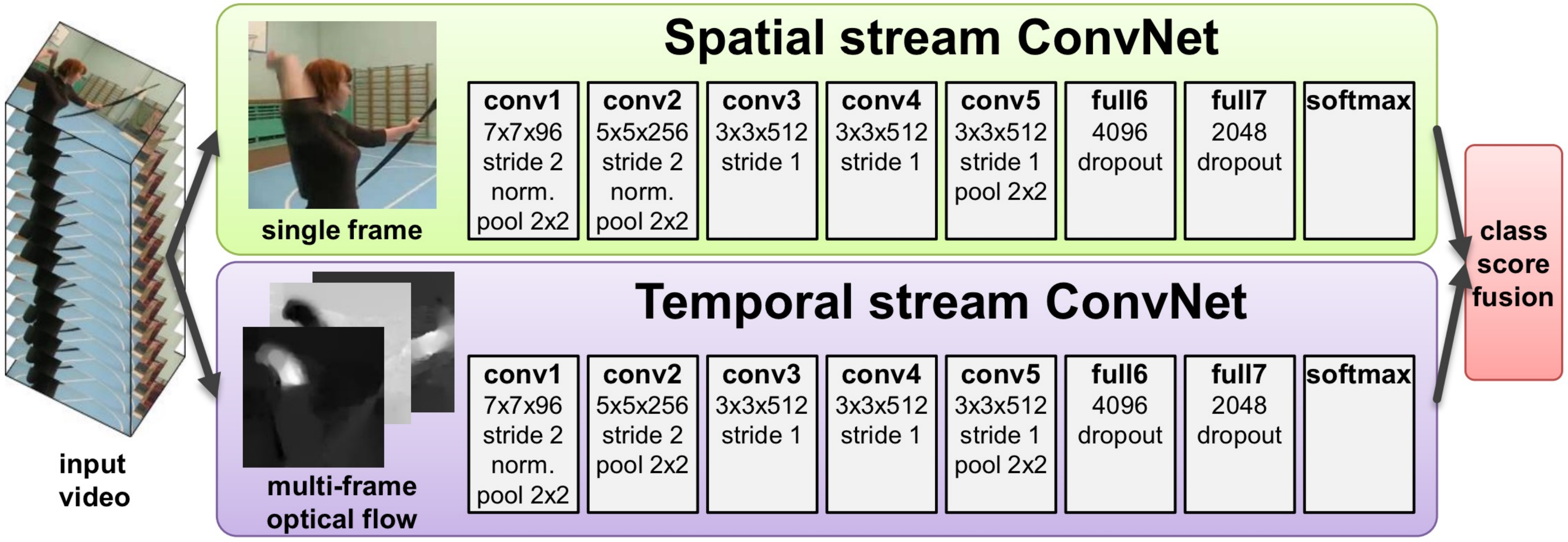
Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

J. Shi and C. Tomasi. [Good Features to Track](). CVPR 1994.

Application: Video Recognition

Optical Flow sometimes used as an **input feature** for video classification with CNNs



Simonyan and Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos", NeurIPS 2014

Application: Motion Magnification

Idea: take flow, magnify it

Liu et al, "Motion Magnification", SIGGRAPH 2005

Application: Motion Magnification

Liu et al, "Motion Magnification", SIGGRAPH 2005

Application: Motion Magnification

Liu et al, "Motion Magnification", SIGGRAPH 2005

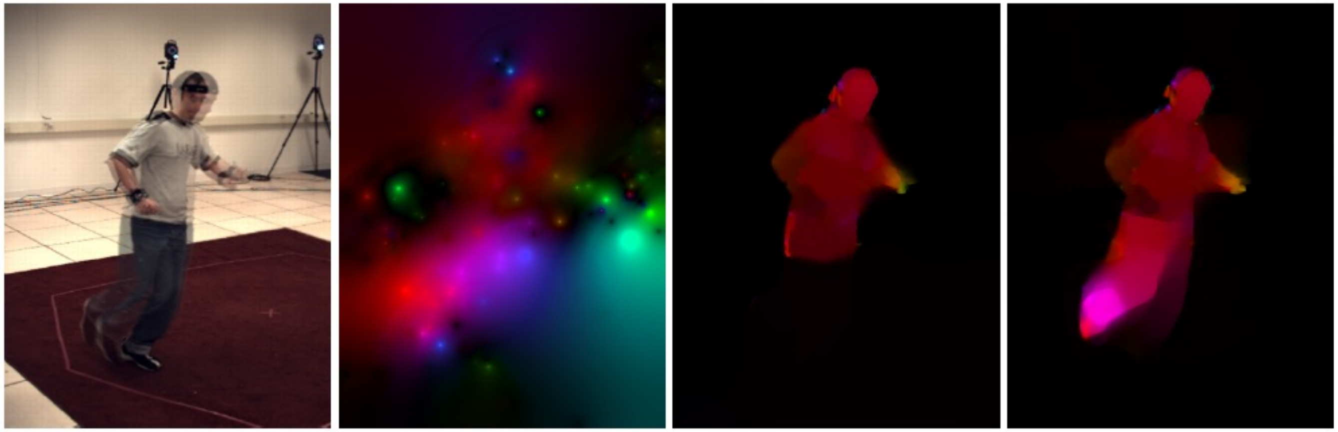
State-of-the-art optical flow, 2009

Start with something similar to Lucas-Kanade

+ gradient constancy

+ energy minimization with smoothing term

+ region matching

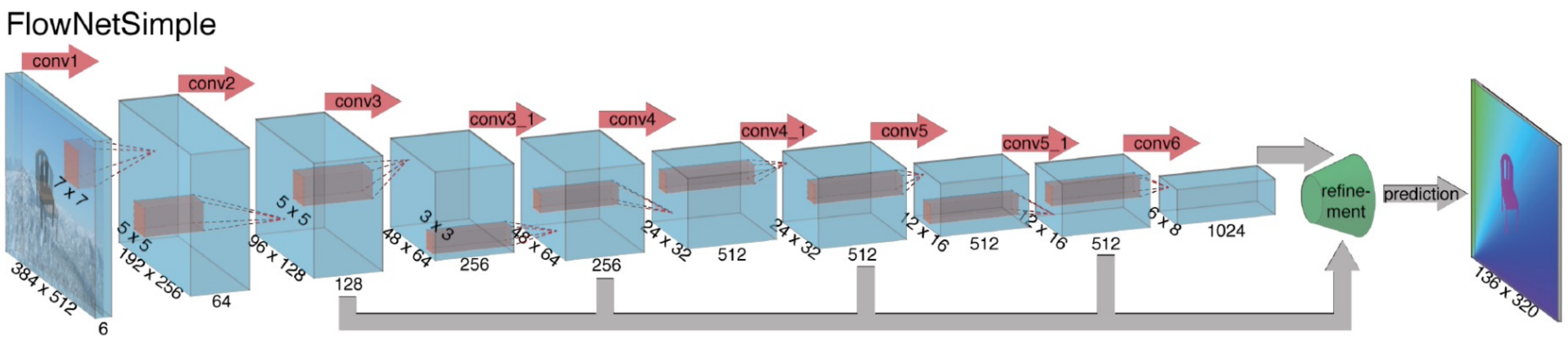


Region-based +Pixel-based +Keypoint-based

[Large displacement optical flow](), Brox et al., CVPR 2009

State-of-the-art optical flow

- Input: 6 channel input (RGB @ t, RGB @ t+1)
- Output: 2 channel input (u,v)
- Current best methods are learned



Dosovitskiy*, Fischer*, et al, “FlowNet: Learning Optical Flow with Convolutional Networks”, ICCV 2015

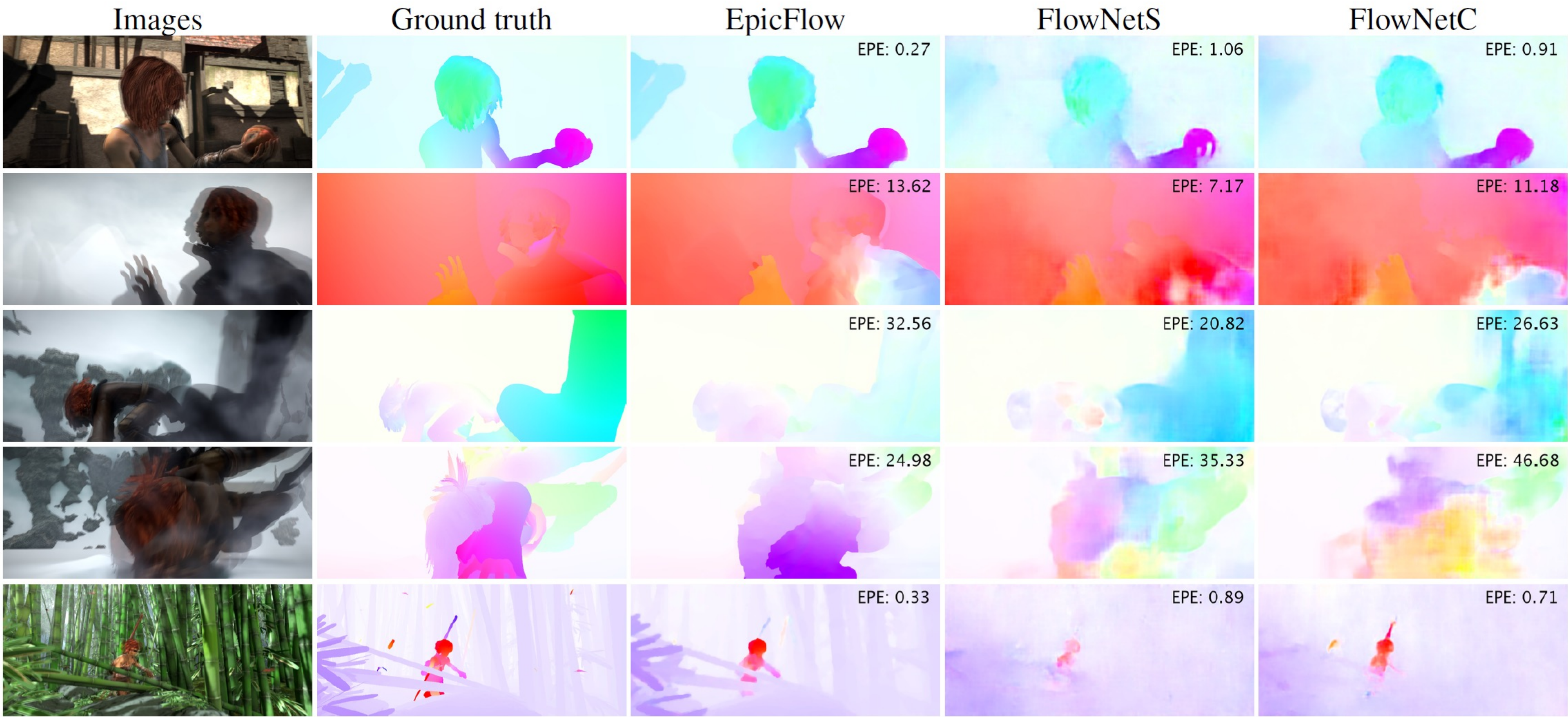
Training Data

Flying Chairs Dataset

Dosovitskiy*, Fischer*, et al, "FlowNet: Learning Optical Flow with Convolutional Networks", ICCV 2015

Deep Optical Flow

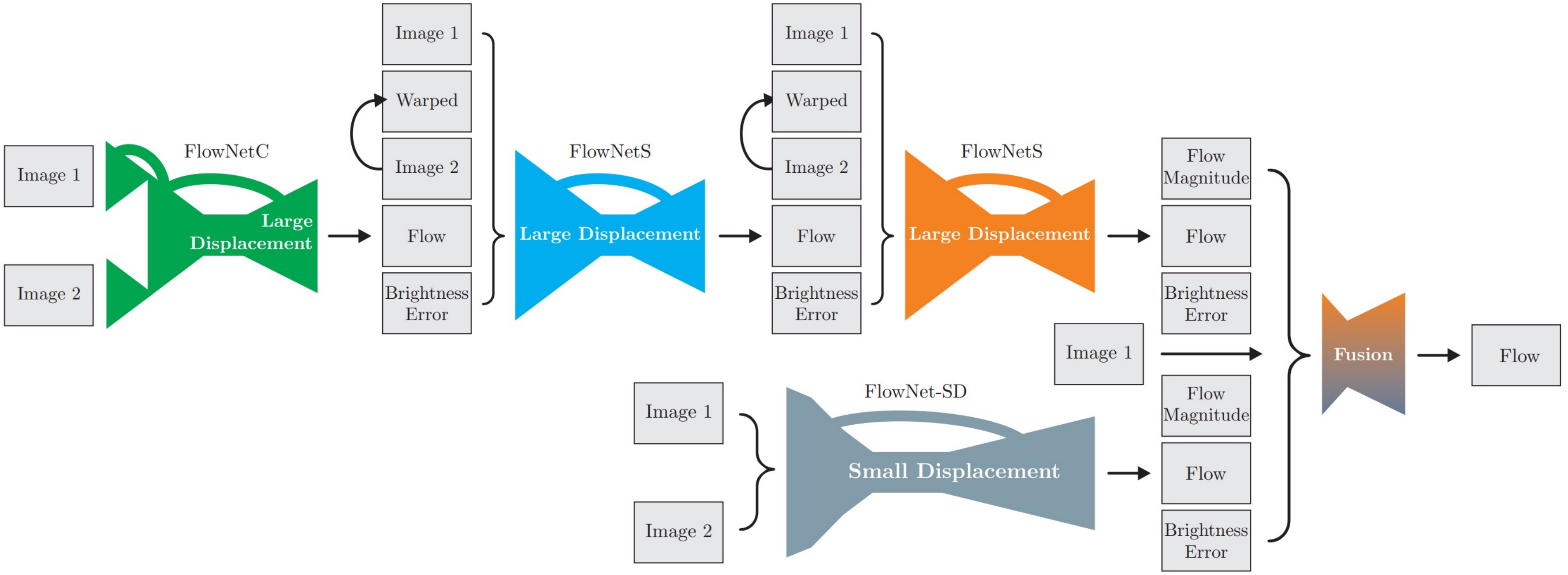
Results on Sintel (standard benchmark)



Dosovitskiy*, Fischer*, et al, "FlowNet: Learning Optical Flow with Convolutional Networks", ICCV 2015

Deep Optical Flow

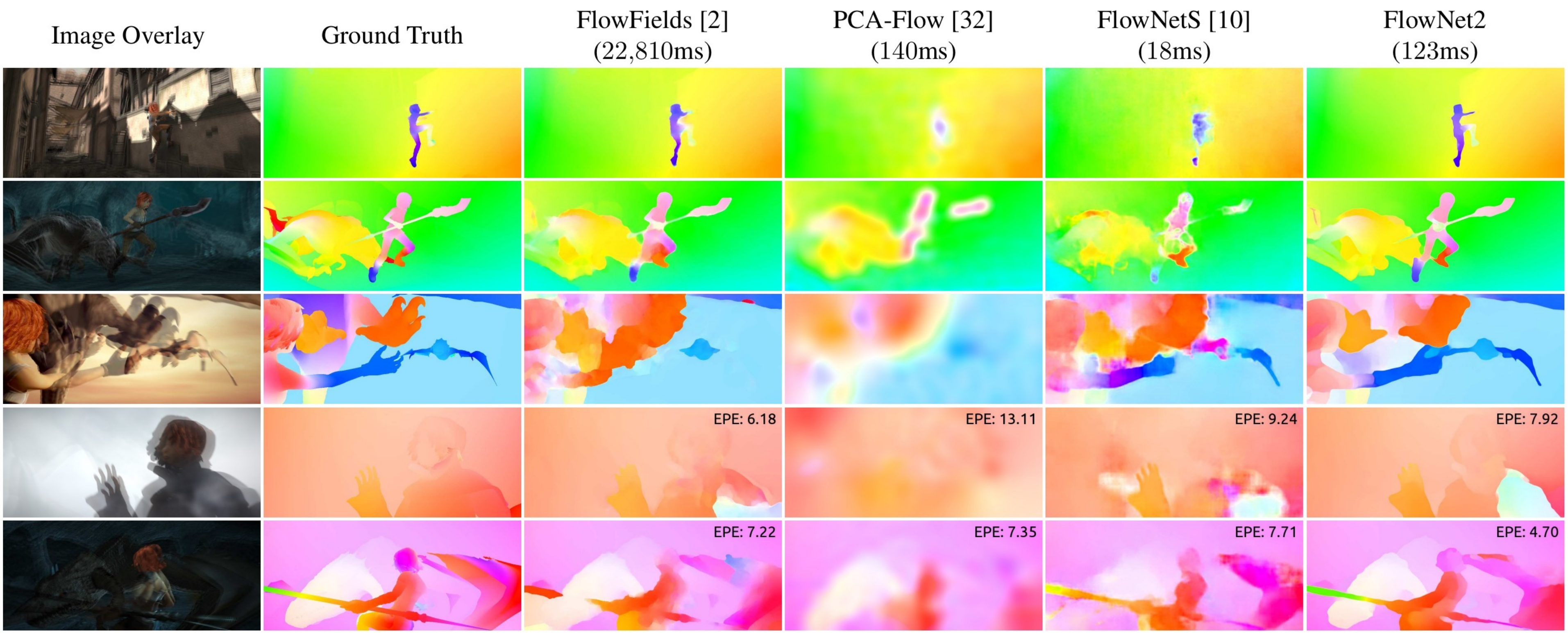
Deeper networks and more sophisticated architectures improve results



Ilg et al, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks", CVPR 2017

Deep Optical Flow

Deeper networks and more sophisticated architectures improve results



Ilg et al, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks", CVPR 2017

Recap: Optical Flow

- Optical flow is the *apparent motion* of pixels in a video
- Can be estimated using *brightness consistency equations*
- *Aperture problem:* Flow might be ambiguous from a small window
- *Lucas-Kanade* solves for 5x5 patches of flow
- *Image pyramids* help with large motion
- Applications: Tracking, video recognition, motion magnification

Next Time:
3D Vision + Calibration