Optical flow

COMPSCI 370: Intro to Computer Vision

Subhransu Maji

April 1 & 3, 2025

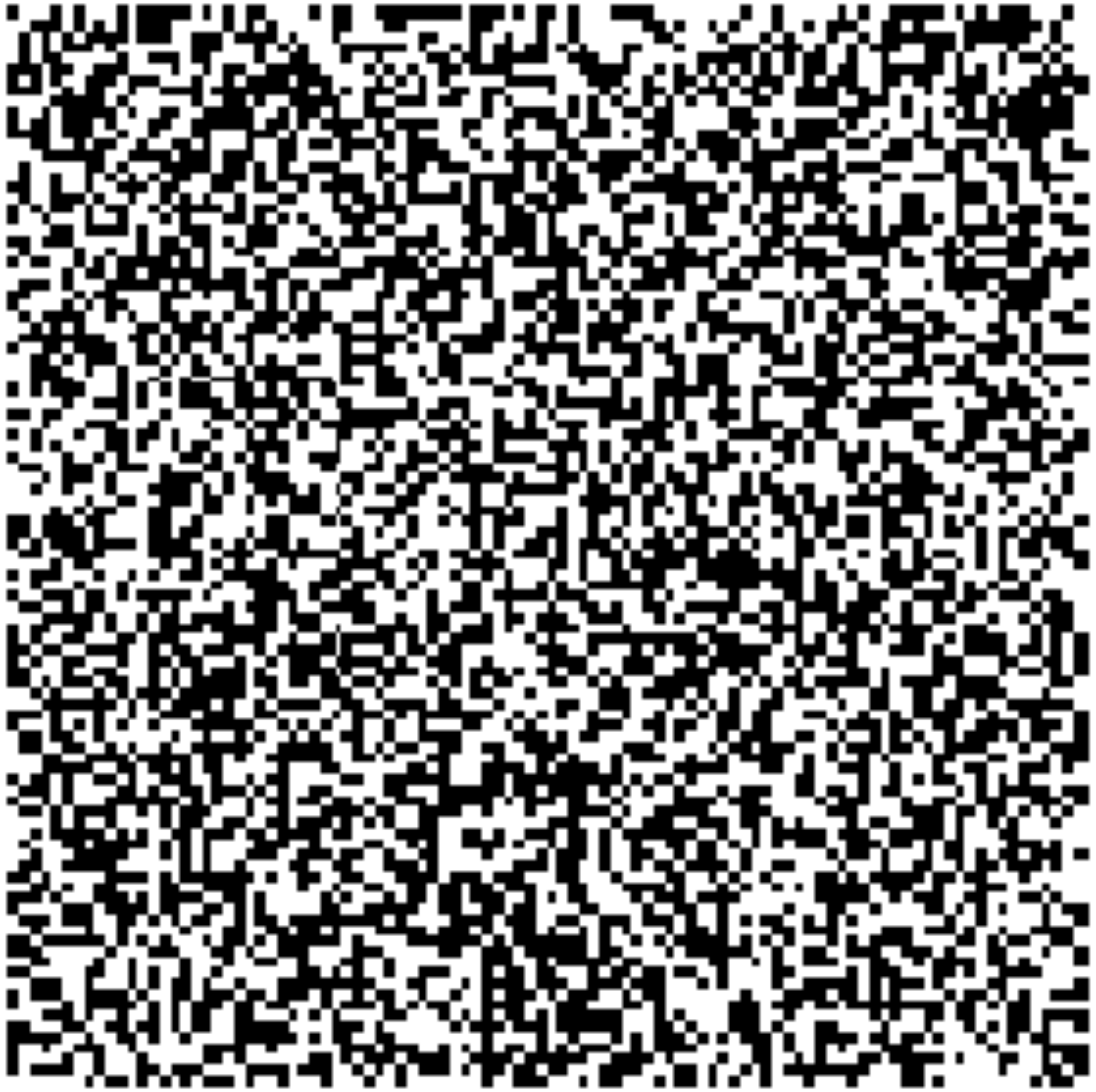
College of
INFORMATION AND
COMPUTER SCIENCES

UMASS
AMHERST

Visual motion

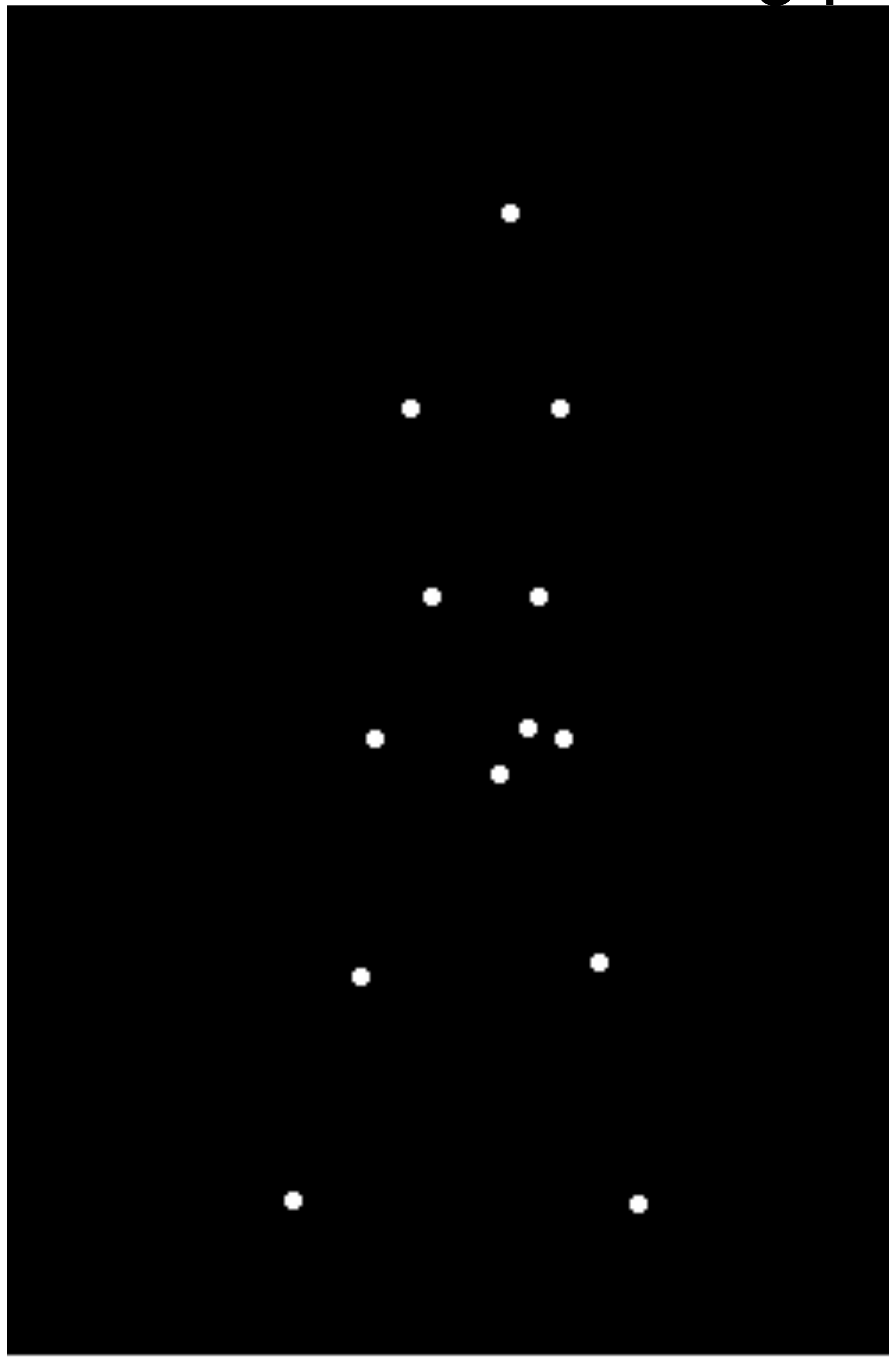
Motion and perceptual organization

Sometimes, motion is the only cue



Motion and perceptual organization

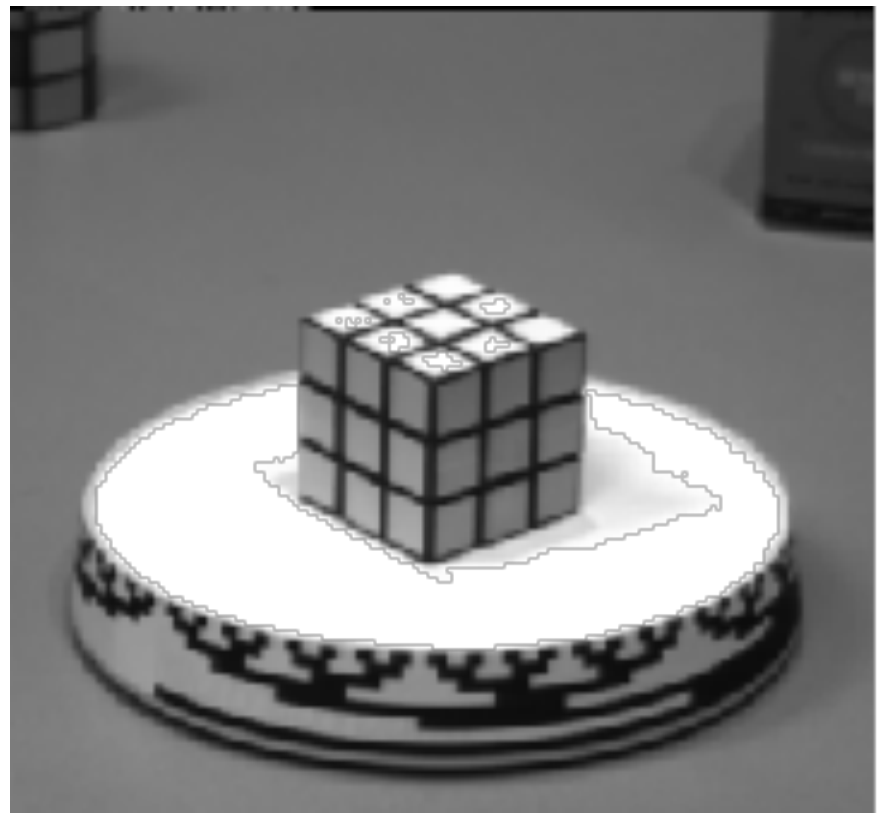
Even “impoverished” motion data can evoke a strong percept

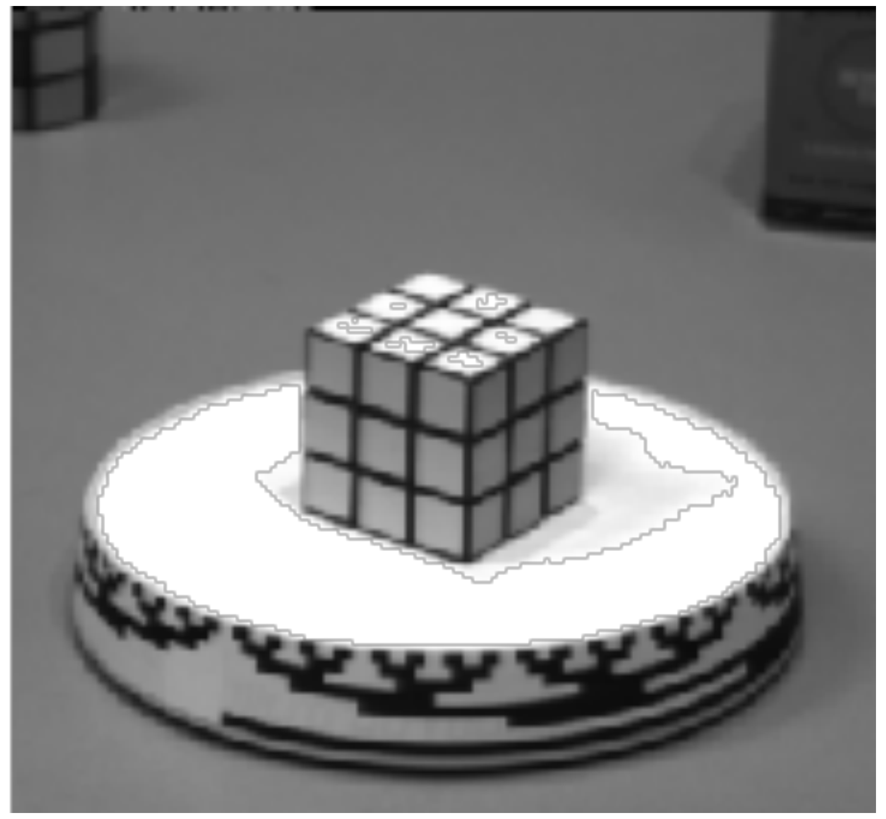


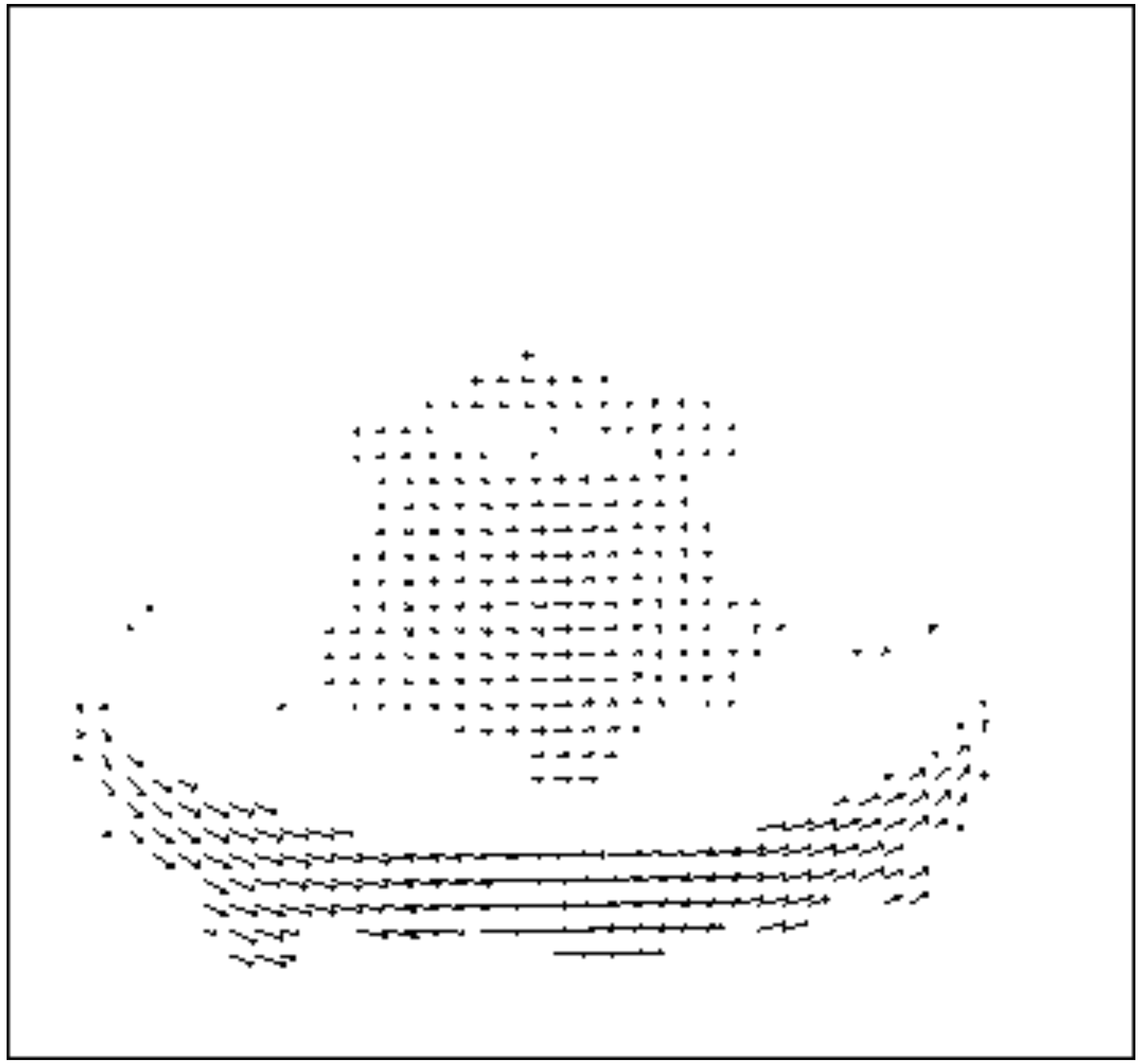
G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", Perception and Psychophysics 14, 201-211, 1973.

Motion field

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







Optical flow

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

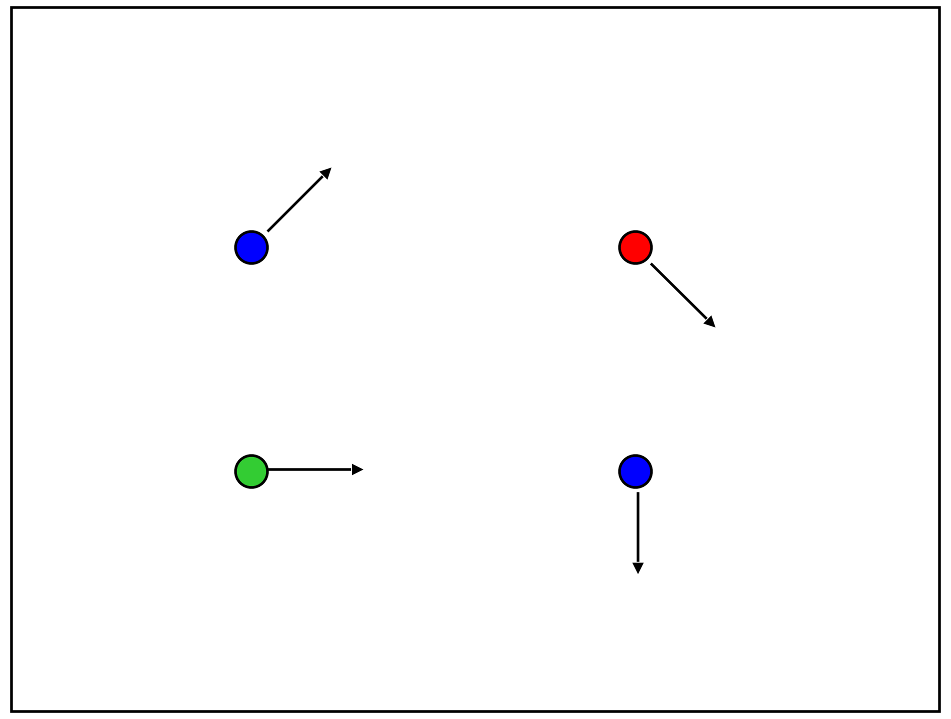
Ideally, optical flow would be the same as the motion field

Have to be careful:

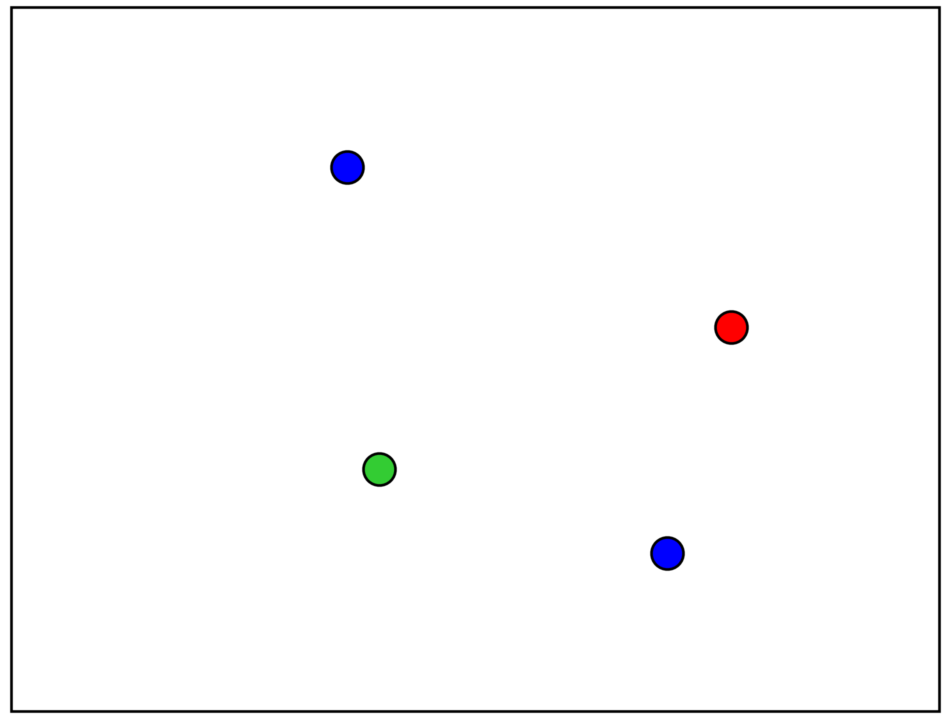
- Apparent motion can be caused by lighting changes without any actual motion
- Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

Estimating optical flow

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



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

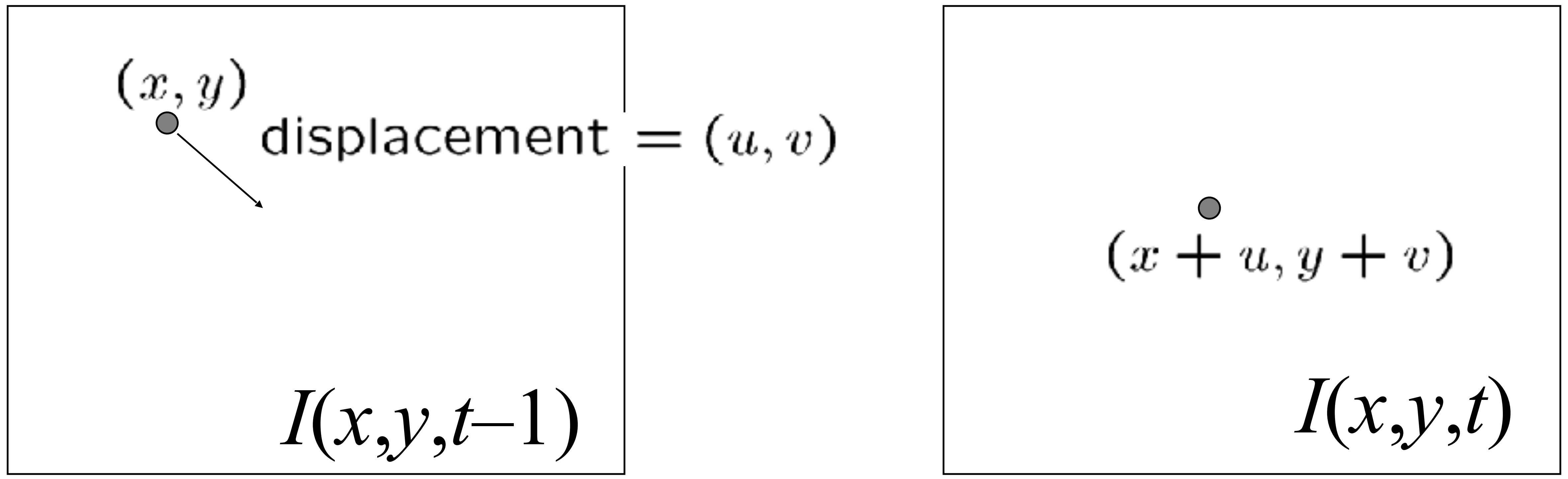


$I(x,y,t)$

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 brightness constancy constraint



Brightness Constancy Equation:

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

Linearizing the right side using Taylor expansion:

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

Hence, $\quad I_x u + I_y v + I_t \approx 0$

The brightness constancy constraint

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

- 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

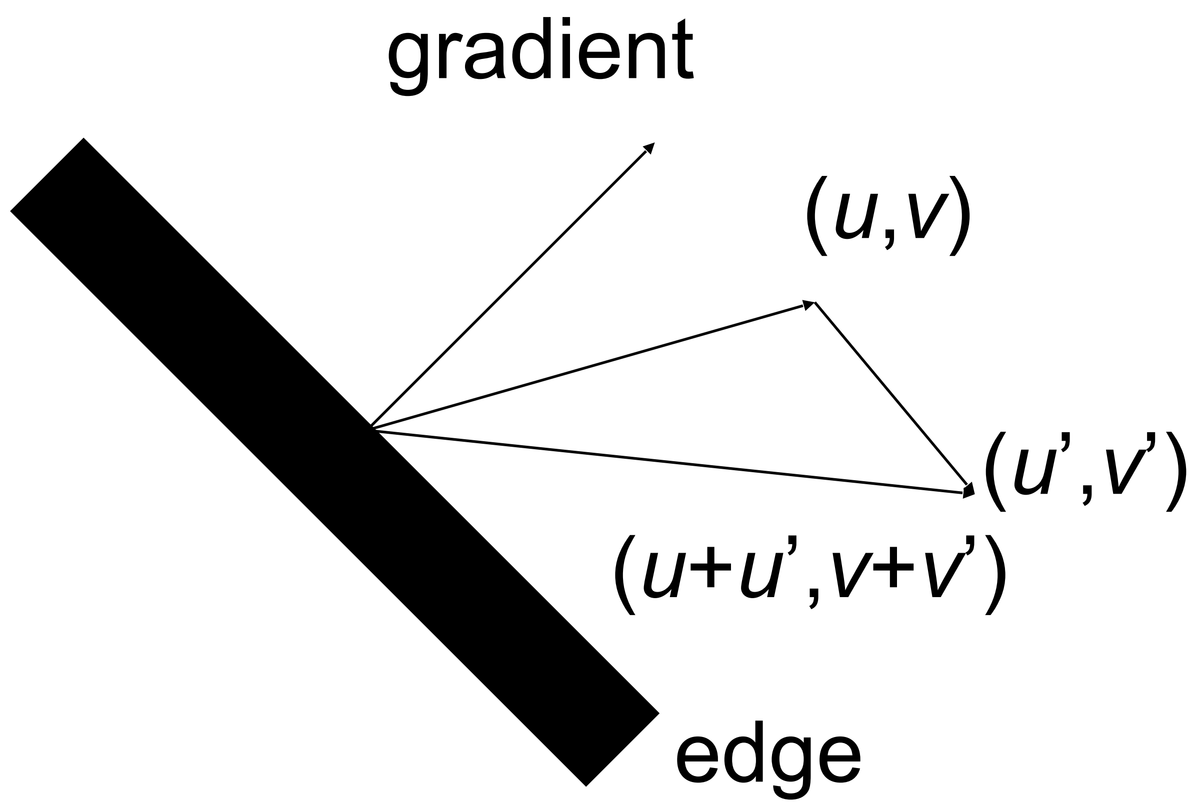
$$I_x u + I_y v + I_t = 0$$

- 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 $\nabla I \cdot (u', v') = 0$



Adding more constraints

How to get more equations for a pixel?

Spatial coherence: pretend the pixel's neighbors have the same flow

- E.g., if we use a 5x5 window, that gives us 25 linear constraints per pixel

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

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

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.

Adding more constraints

Least squares:

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

When is this system solvable?

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.

Lucas-Kanade flow

Linear least squares problem

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

$$\underset{n\times 2}{\mathbf{A}} \; \underset{2\times 1}{\mathbf{d}} = \underset{n\times 1}{\mathbf{b}}$$

Solution given by $(\mathbf{A}^T\mathbf{A})\mathbf{d} = \mathbf{A}^T\mathbf{b}$

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

The summations are over all pixels in the window

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.

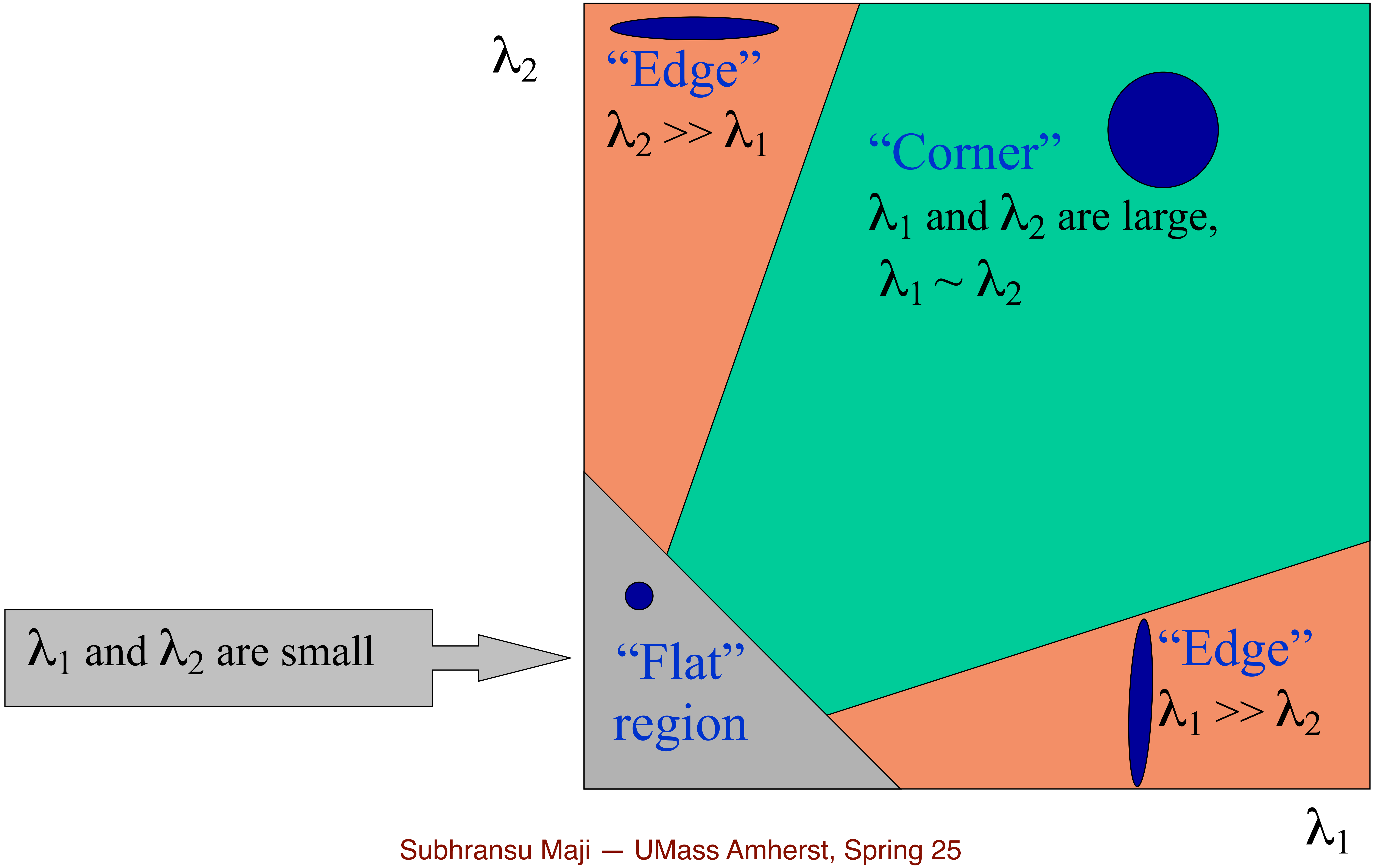
Lucas-Kanade flow — when does it work?

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

- **M** = **A**^{T}**A** is the *second moment matrix*
- We can figure out whether the system is solvable by looking at the eigenvalues of the second moment matrix

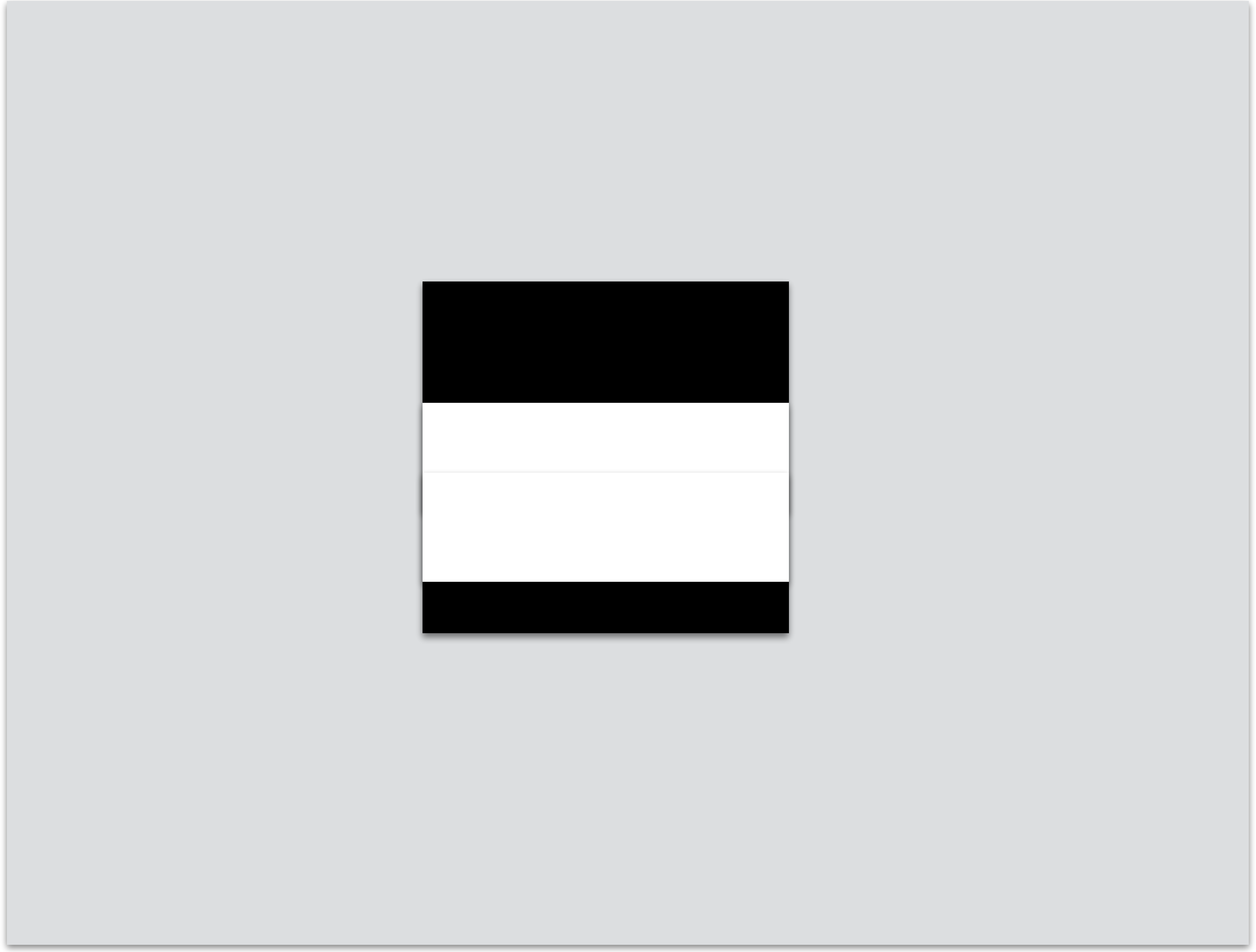
Interpreting the eigenvalues

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



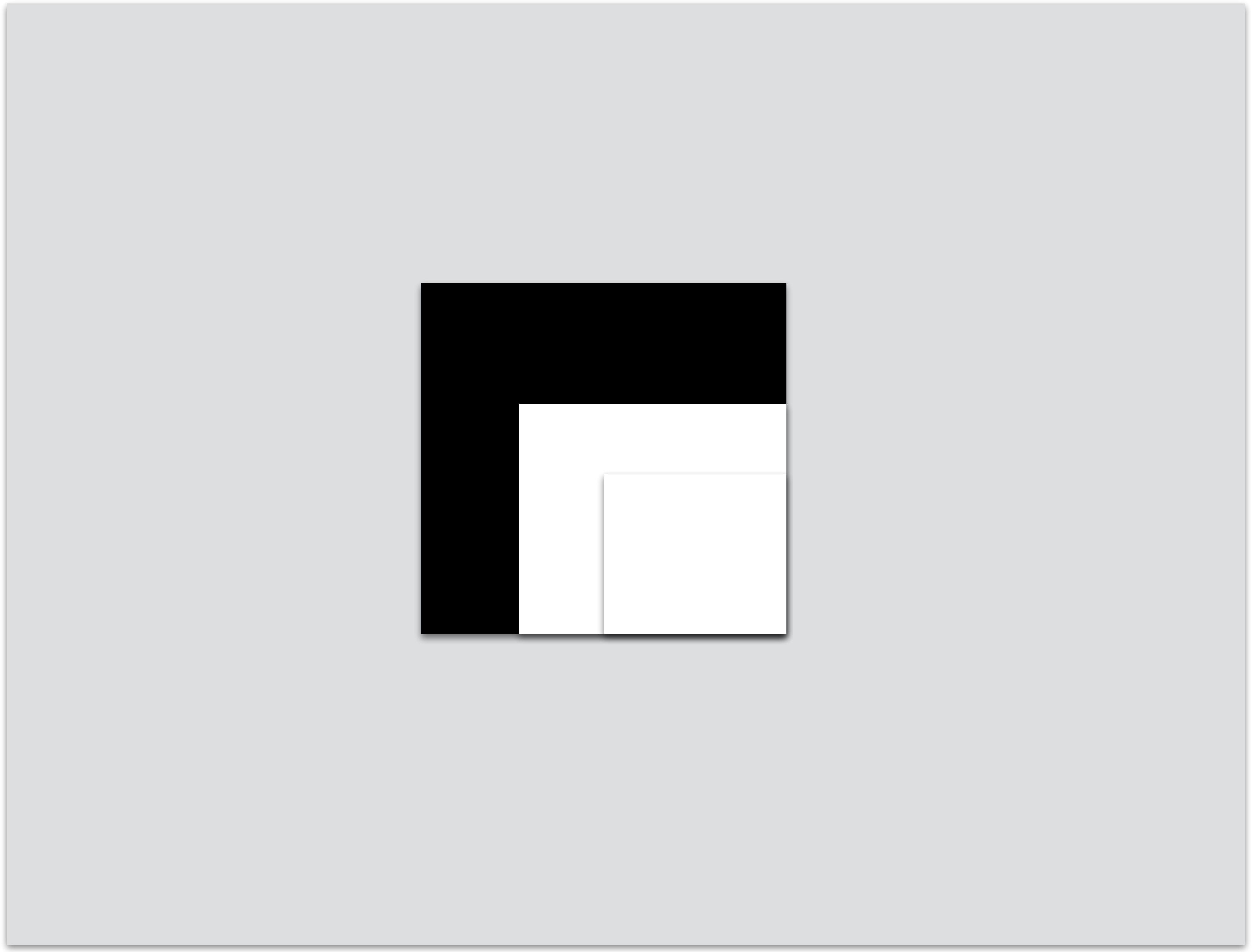
Conditions for solvability

“Bad” case: single straight edge



Conditions for solvability

"Good" case: corner



Example

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Uniform region

– gradients have small magnitude
– small λ_1, small λ_2
– system is ill-conditioned

Edge

– gradients have one dominant direction
– large λ_1, small λ_2
– system is ill-conditioned

High-texture or corner region

– gradients have different directions, large magnitudes
– large λ_1, large λ_2
– system is well-conditioned

Coming up!

Demo

Optical flow demo in OpenCV

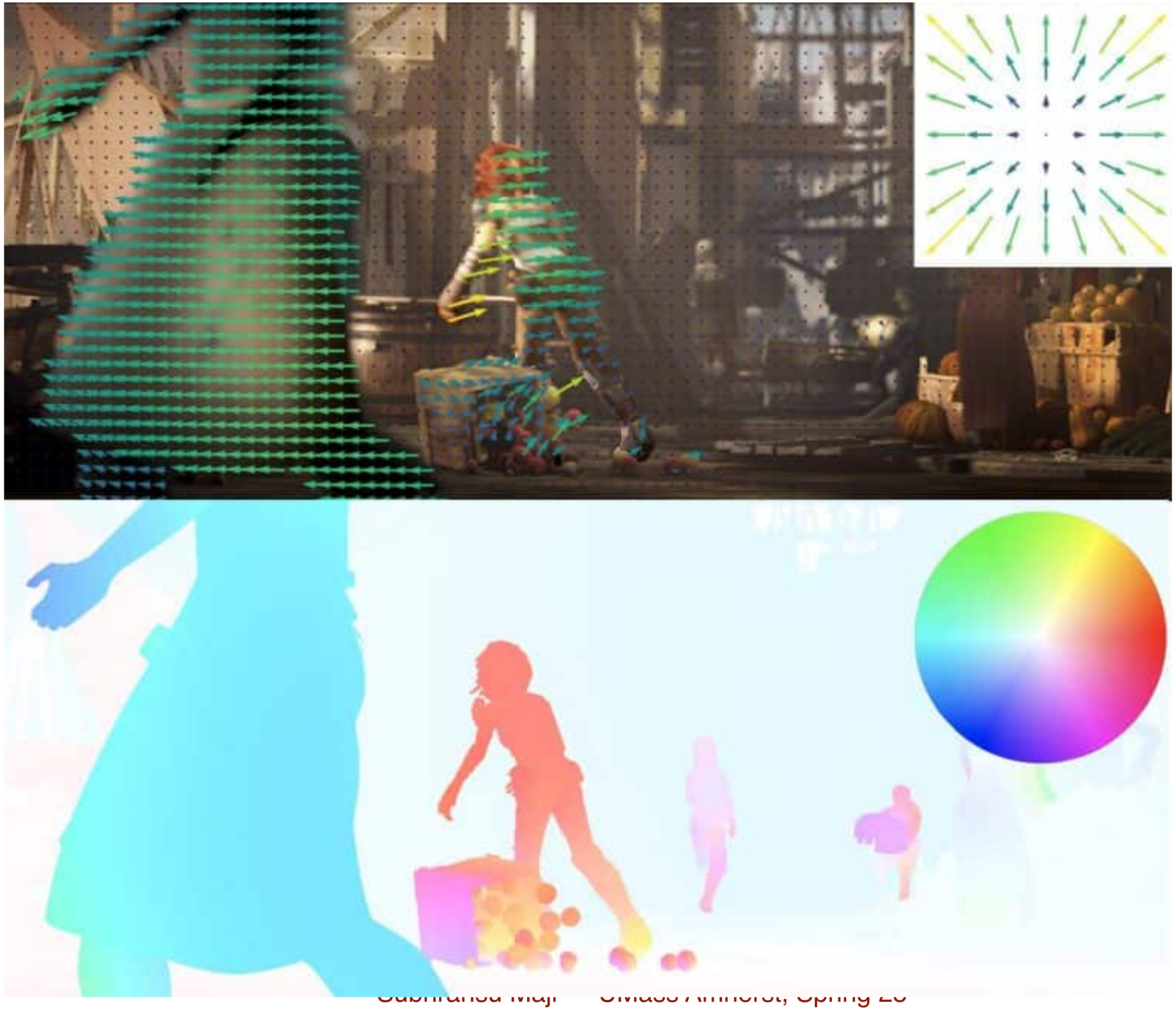
Depth estimation

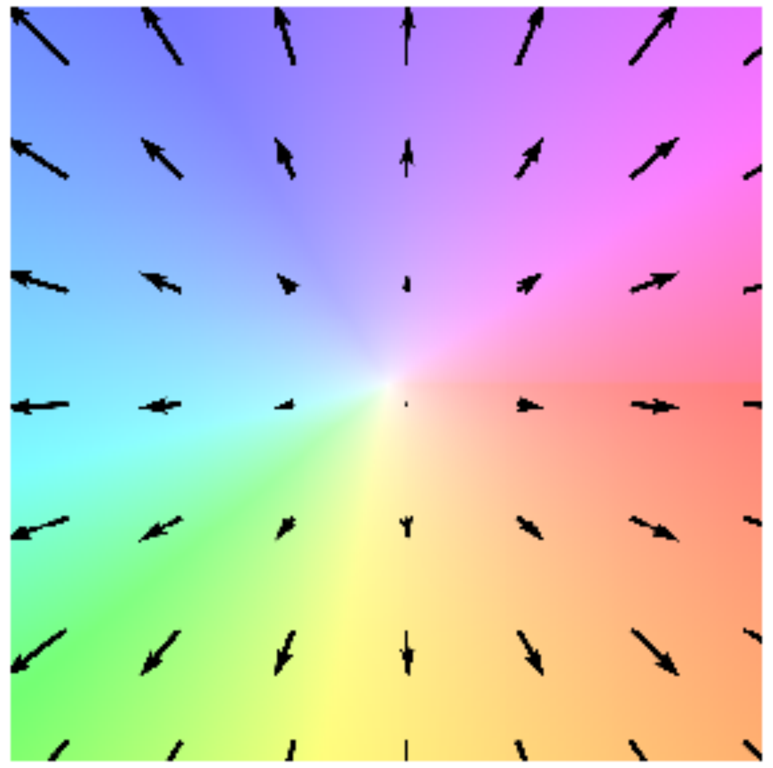
Point tracking

Video interpolation

Challenges in estimating flow and modern approaches

Different ways to visualize flow

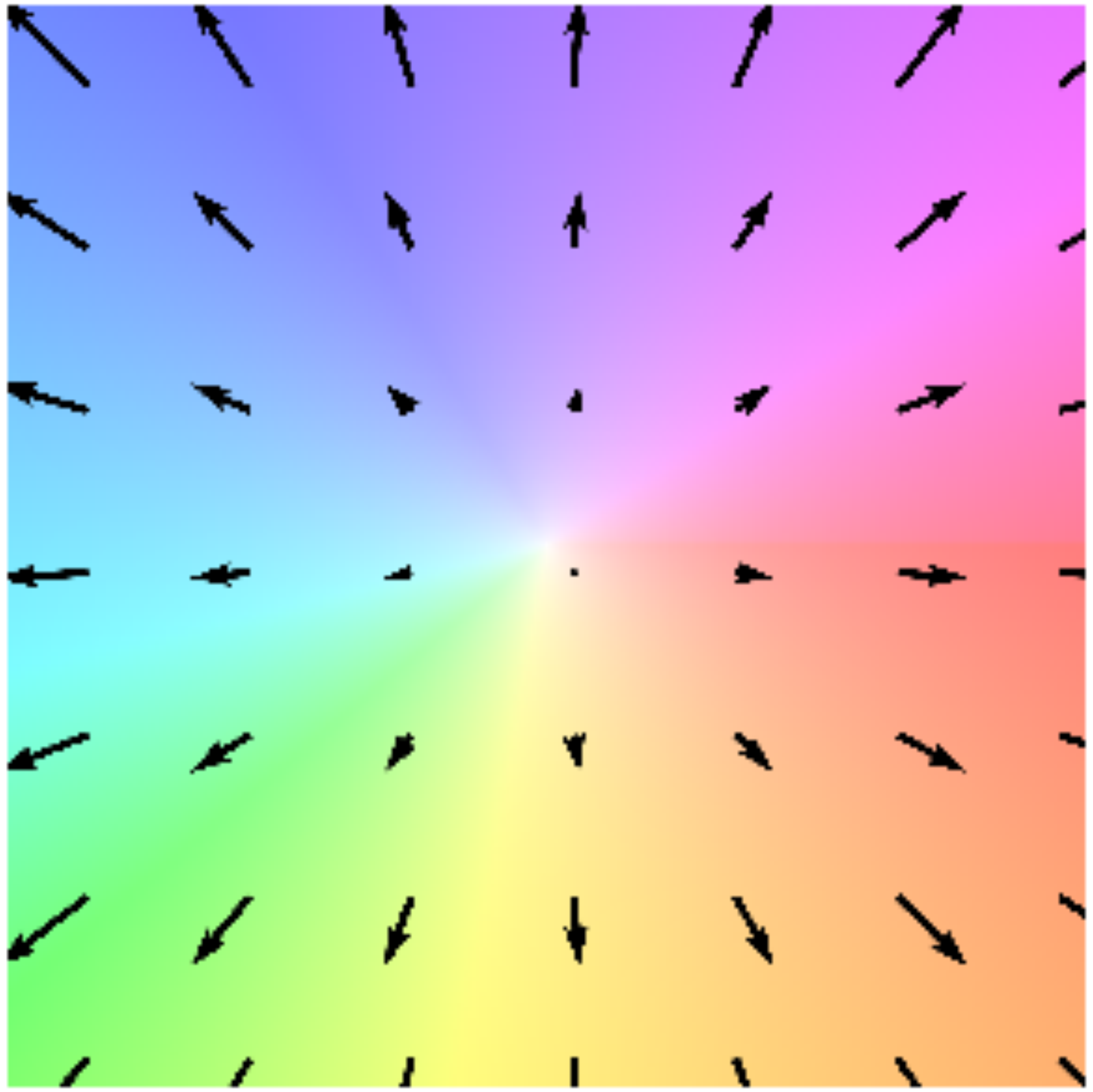




(d) Motion visualization

Visualizing flow using the color wheel

Encode direction as the hue and magnitude as saturation



(d) Motion visualization

Coming up!

Optical flow demo in OpenCV

Depth estimation

Point tracking

Video interpolation

Challenges in estimating flow and modern approaches

Demo

Depth estimation

image 1

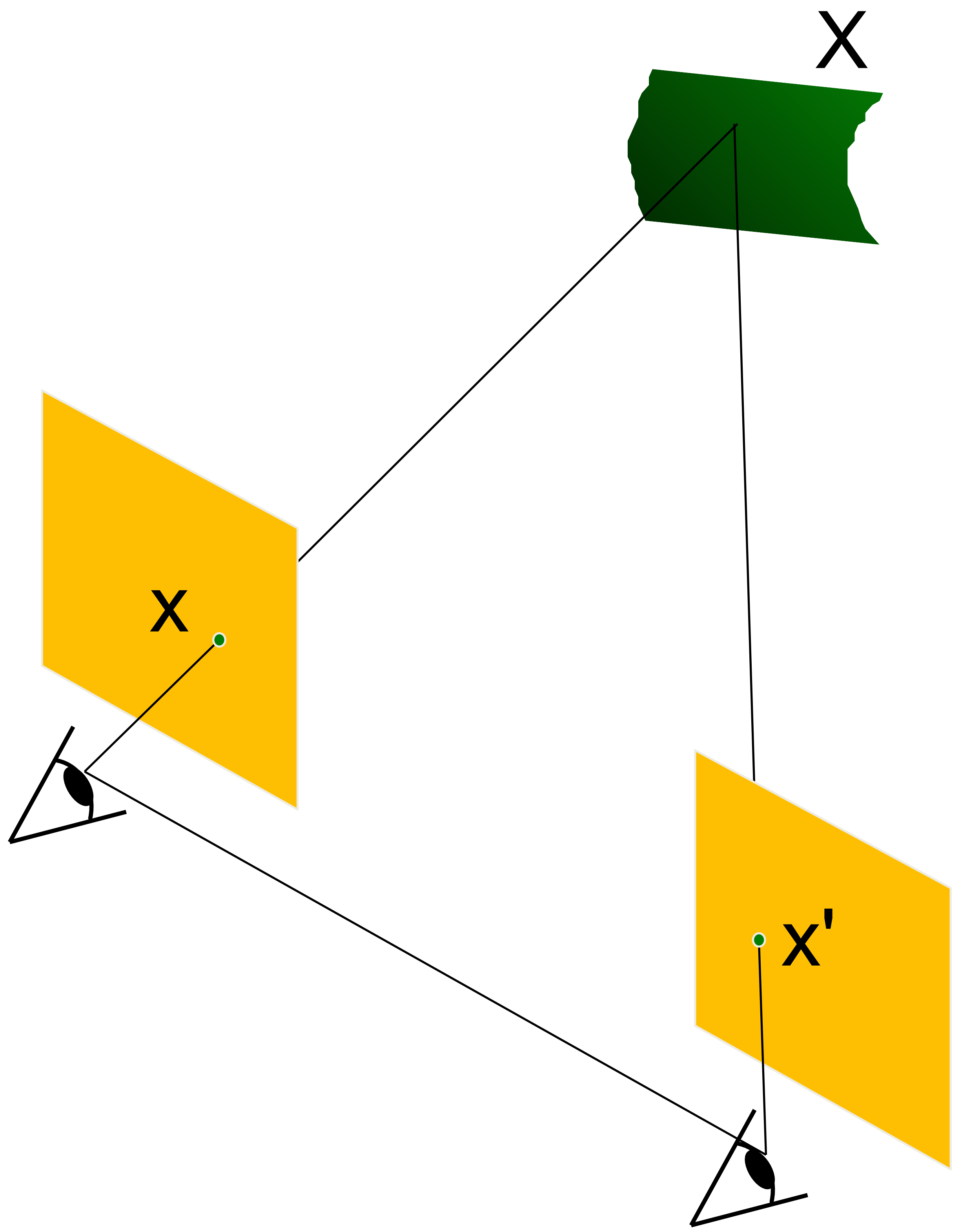
image 2

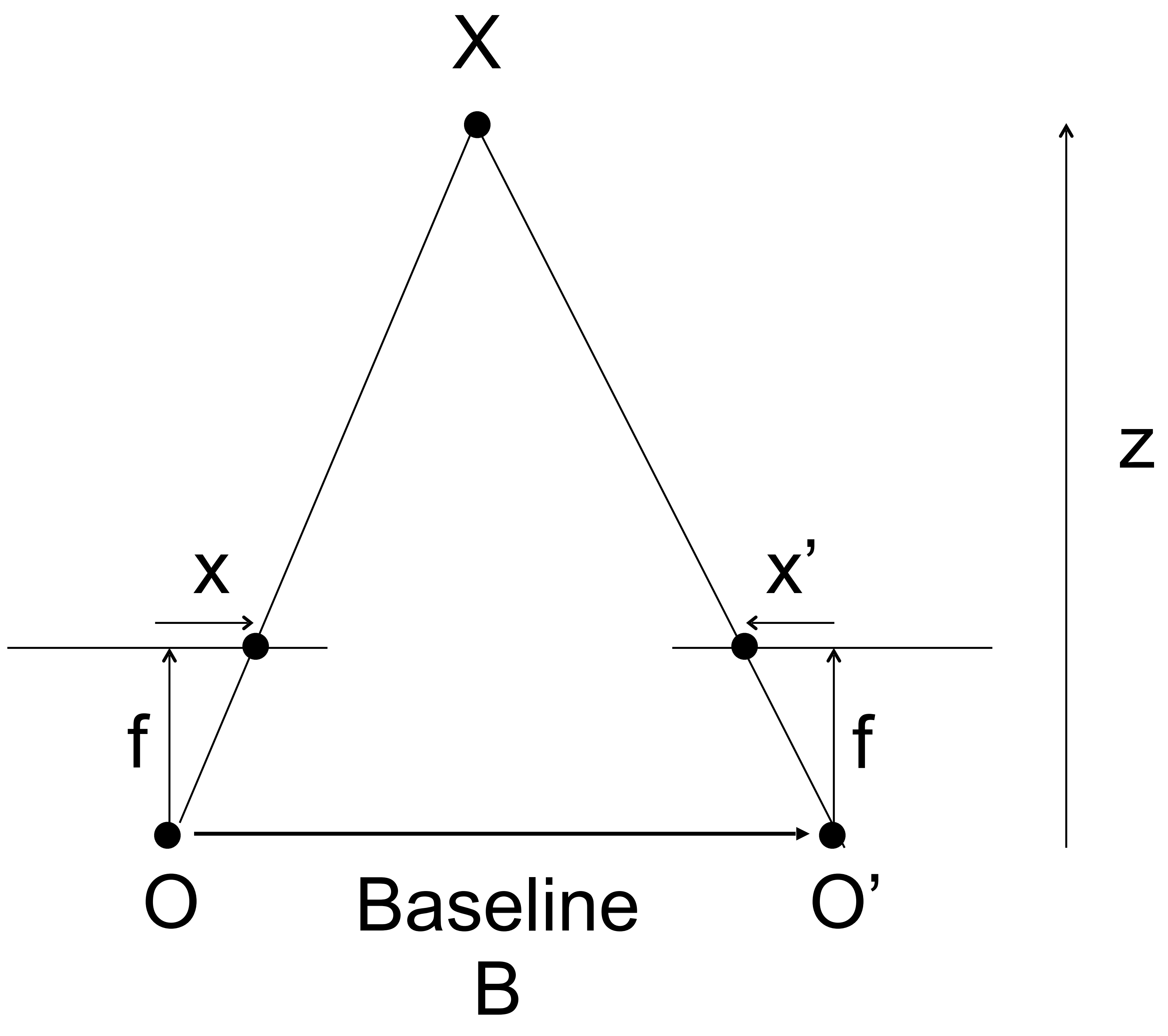
Dense depth map

Some of following slides adapted from Steve Seitz and Lana Lazebnik

Depth from flow

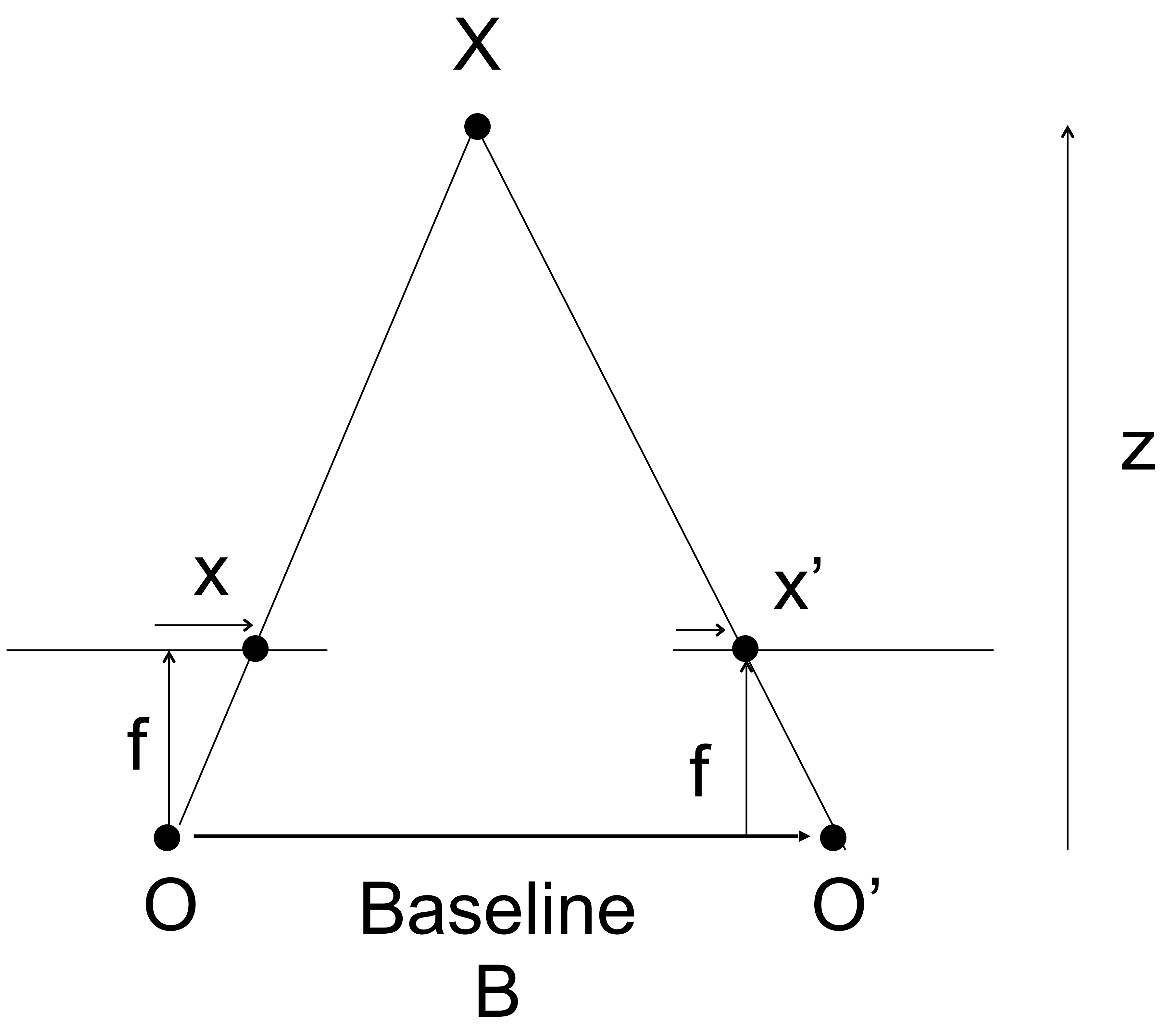
Goal: recover depth by finding image coordinate x' that corresponds to x





Depth from flow

$$\frac{x - x'}{O - O'} = \frac{f}{z}$$



$$disparity = x - x' = \frac{B \cdot f}{z}$$

Disparity is inversely proportional to depth.

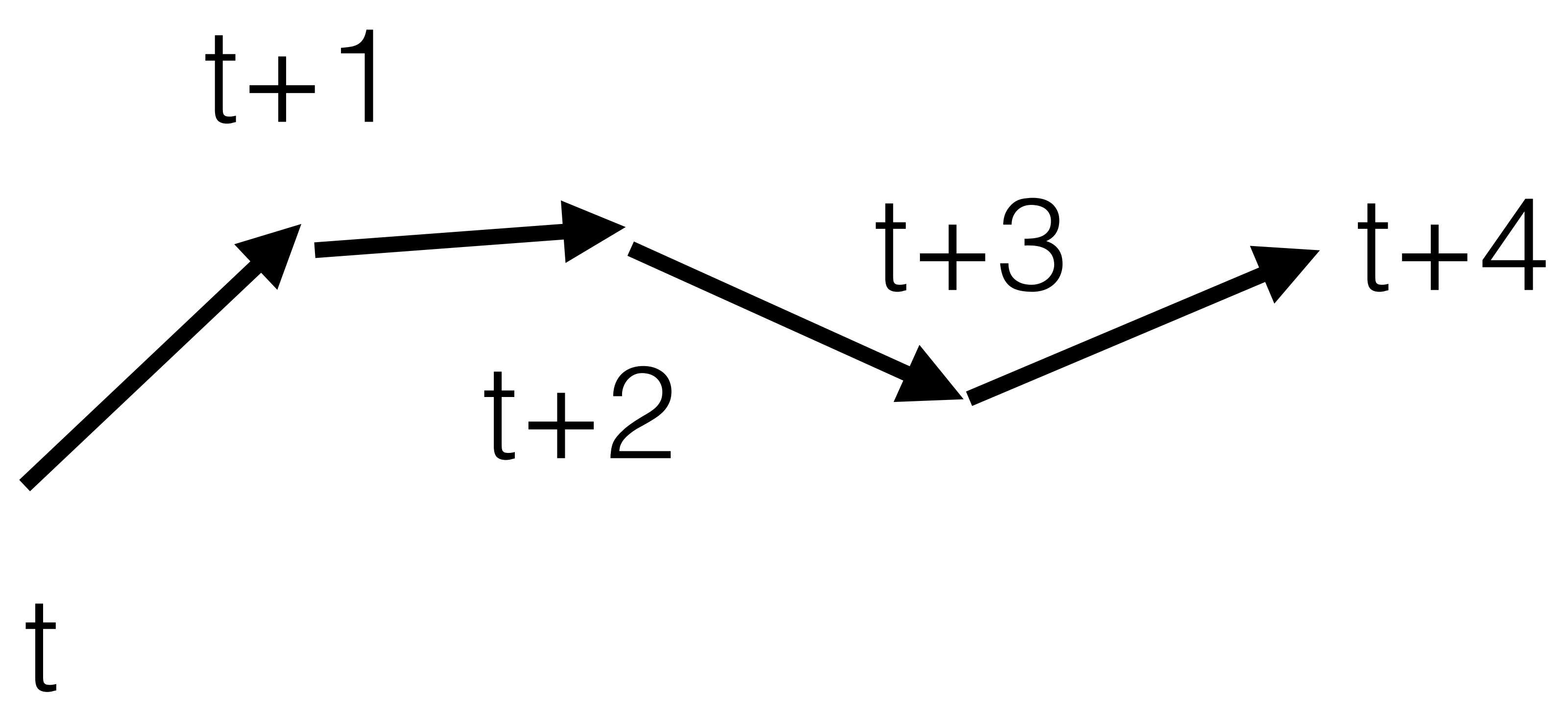
Point tracking

What are good features for tracking? Corners!

Kanade-Lucas-Tomasi (KLT) Tracker

- Estimate optical flow using Lucas-Kanade
- Detect corners in each frame
- Store displacement of each corner using estimate flow
- Chain displacements to form long trajectories

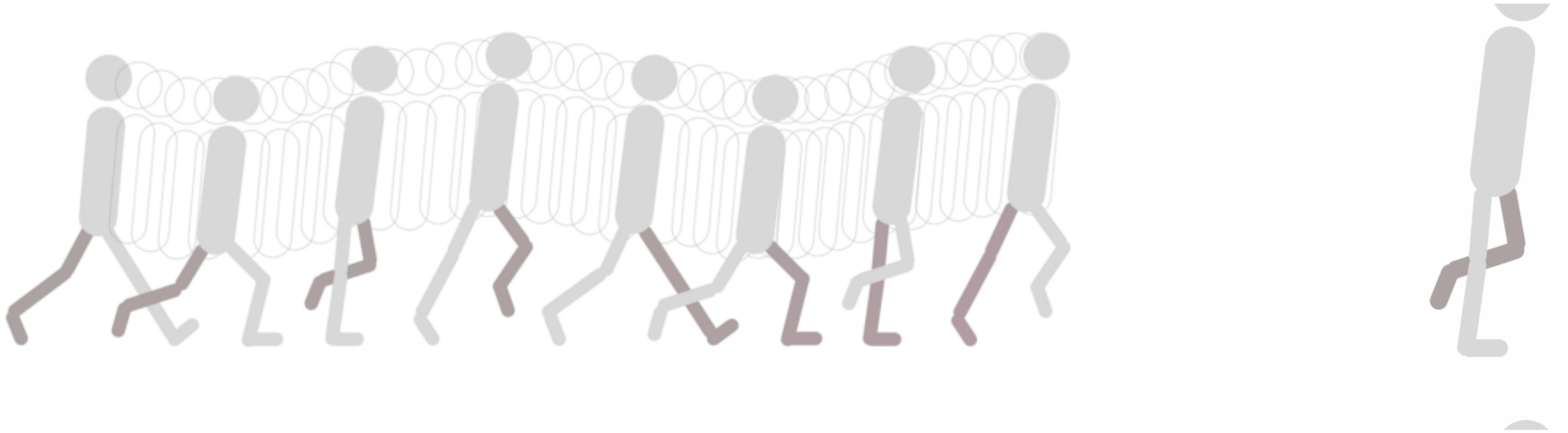
Demo



Video interpolation

Given two frames estimate multiple intermediate frames

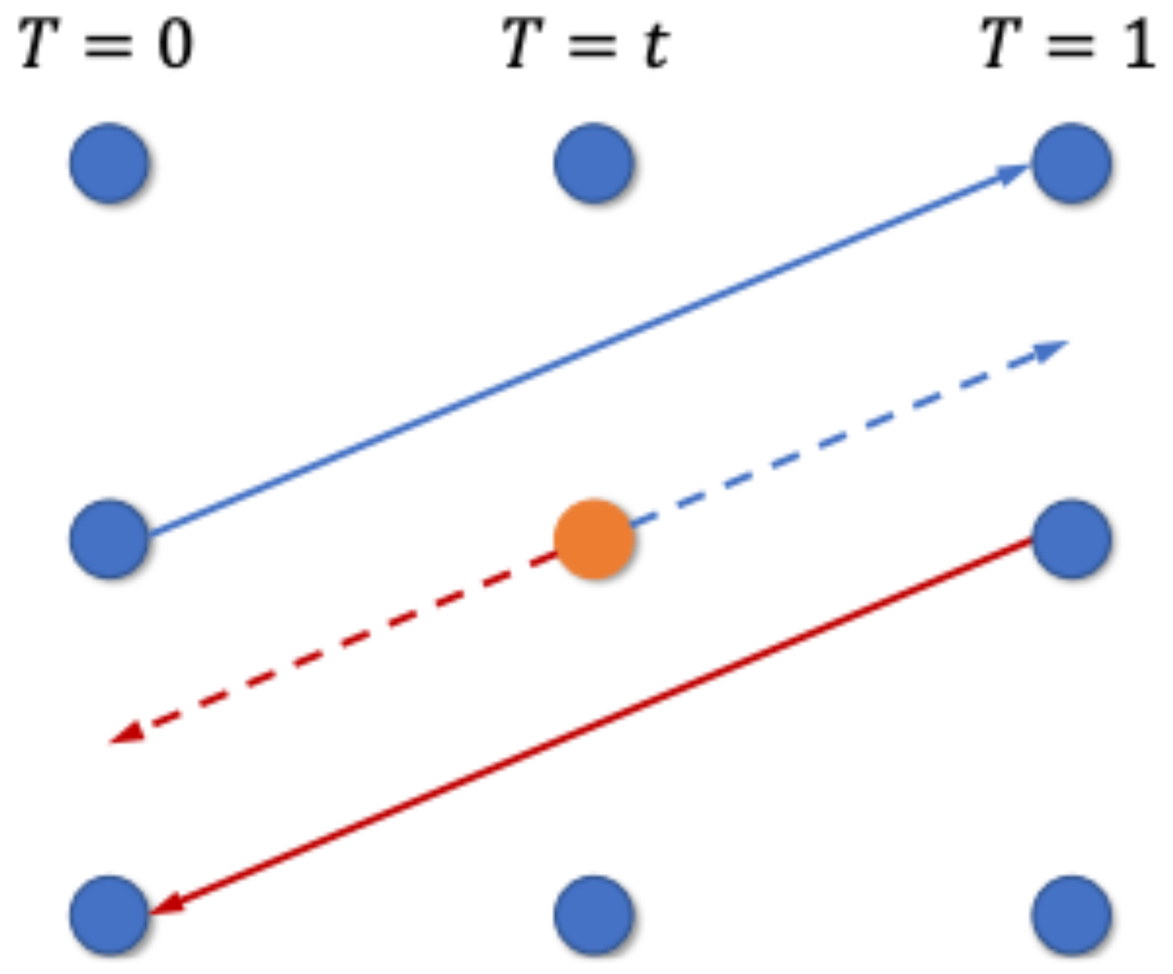
Can be use to play the video at a higher frame rate (or in slow motion)

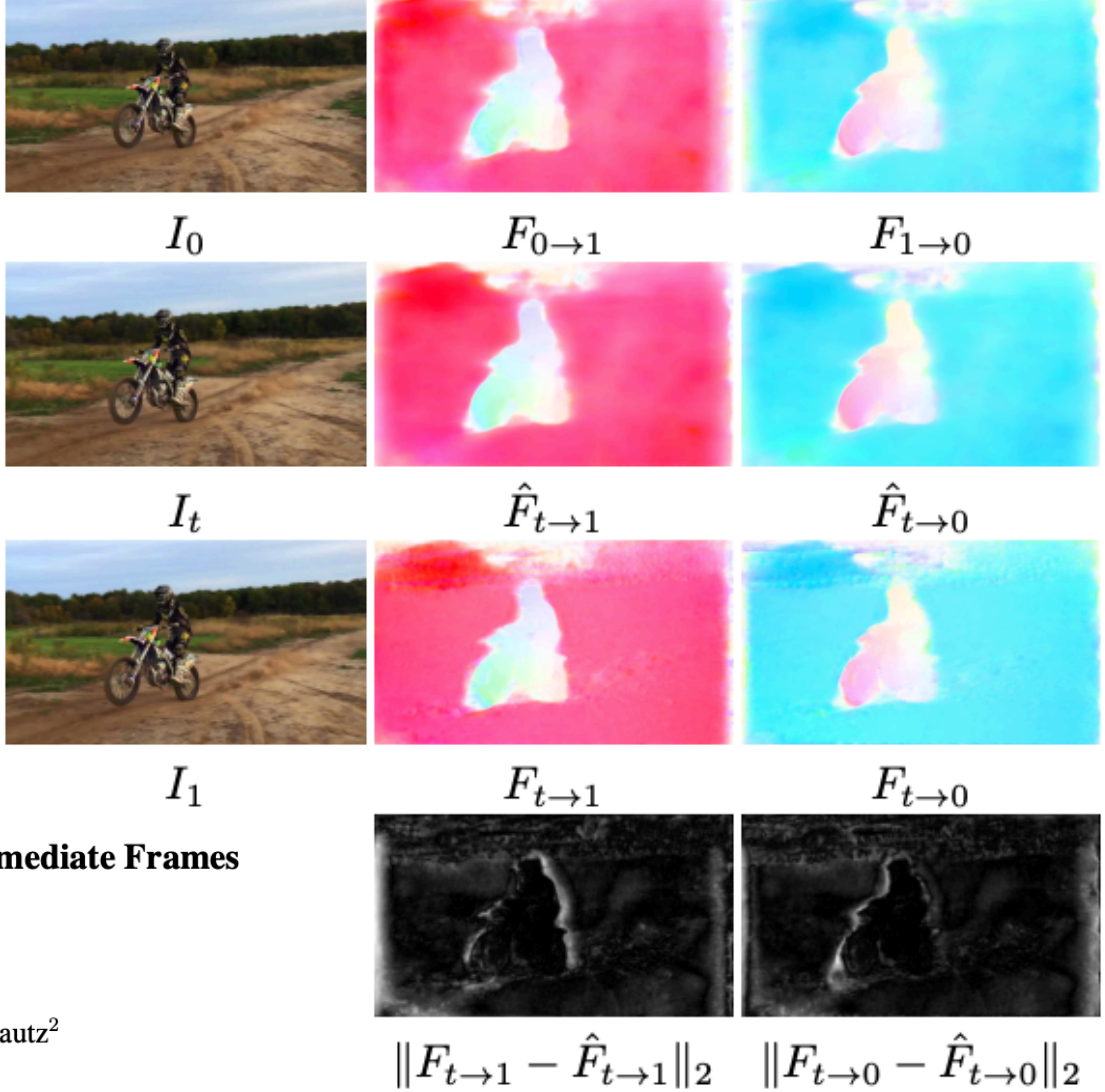


Source https://www.freecodecamp.org/news/understanding-linear-interpolation-in-ui-animations-74701eb9957c/

Video interpolation

https://jianghz.me/projects/superslomo/





Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation

Huaizu Jiang[1] Deqing Sun[2] Varun Jampani[2]
Ming-Hsuan Yang[3,2] Erik Learned-Miller[1] Jan Kautz[2]
[1]UMass Amherst [2]NVIDIA [3]UC Merced
{hzjiang,elm}@cs.umass.edu,{deqings,vjampani,jkautz}@nvidia.com, mhyang@ucmerced.edu

Errors in Lucas-Kanade

The motion is large (larger than a pixel)

- Iterative refinement
- Coarse-to-fine estimation
- Exhaustive neighborhood search (feature matching)

A point does not move like its neighbors

- Motion segmentation

Brightness constancy does not hold

- Exhaustive neighborhood search with normalized correlation

Alternative optical flow methods

Apply a smoothness constraint or regularization on the entire flow field **(Horn-Schunck method)**

Estimate flow by solving an optimization problem across all pixels:

$$\sum \Psi(I_x u + I_y v + I_t) + \alpha L(|\nabla u|) + \alpha L(|\nabla v|)$$

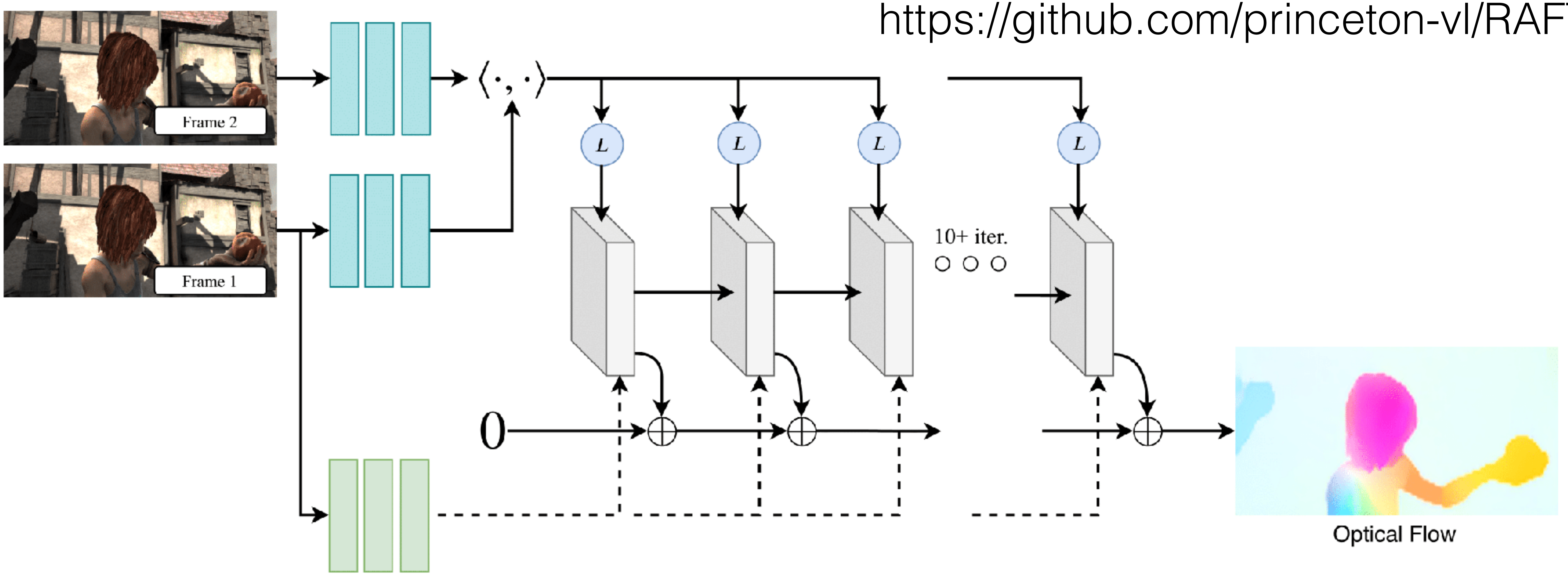
Brightness constancy

Smoothness constraints

Tends to give dense, highly-accurate flow, but requires non-trivial optimization techniques.

Modern approaches (e.g., RAFT)

https://github.com/princeton-vl/RAFT



RAFT: Recurrent All Pairs Field Transforms for Optical Flow
ECCV 2020
Zachary Teed and Jia Deng

Training data

Often trained on synthetic data! Figure from MPI Sintel Dataset, Butler et al., ECCV'12 https://ps.is.mpg.de/code/sintel-optical-flow-dataset