Computer Vision

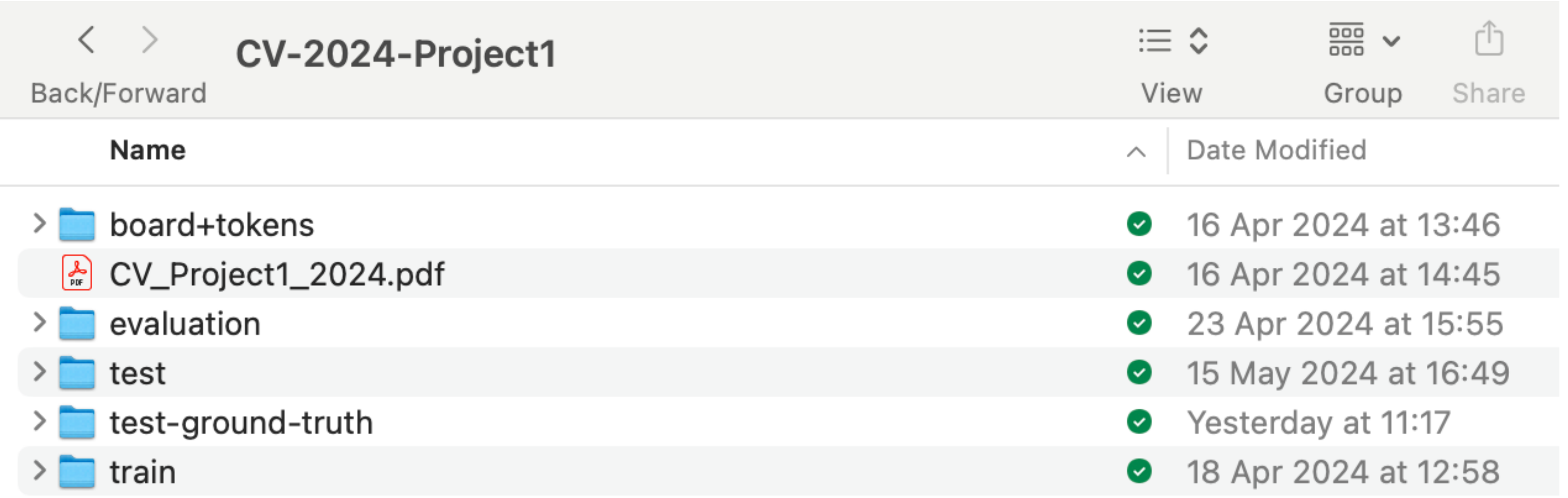
Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2023-2024

Ground-truth annotations added for test data in Project 1

- ground-truth annotations added for test data in my Dropbox where you have all the files for Project 1



Intermediate results for Project 1

-below are results after runing our evaluation routine on your provided predictions vs ground-truth test annotations (maximum 4.5 points)

-still to add: 0.5 points from presentation+pdf, 0.5 points from ex-officio (right format)

Nr. crt.	Nume + prenume student	Grupa	Points (all 3 tasks)
1	Albu Angelica	411	1.935
2	Anghelina Ion Marian	407	4.5
3	Anton Andrei Alexandru	407	4.085
4	Barbu Iulia Andreea	407	3.6525
5	Bejenariu David-Cosmin	407	2.9975
6	Bocu Stefania Denisa	407	2.6
7	Bors Andrei	407	1.8025
8	Breustedt Emil	Erasmus	1.2325
9	Broscoteanu Daria Mihaela	407	4.4725
10	Buta Gabriel-Sebastian	407	4.325
11	Chirobocea Mihail	407	4.42
12	Coconu Teodor Ioan	407	4.0675
13	Costea Razvan George	407	3.8875
14	Dirvareanu Ionut	407	3.14
15	Dutica Maria-Diana	411	1.1225
16	Enache Alexandru	407	4.5
17	Georgescu Mihnea Stefan	407	4.40
18	Gheorghe Silviu Florin	407	4.5
19	Girbea Monica Andreea	407	2.91
20	Hadirca Dionisie	407	4.365
21	Handac Alexandru Radu	407	3.485
22	Hirica Ioan Alexandru	407	4.1725
23	Ionescu Alexandru Theodor	407	4.5
24	Jarca Andrei	407	4.50
25	Marin Mircea-Mihai	407	4.43
26	Mihalcea Dragos Stefan	407	0
27	Musledin Selma	407	4.1525
28	Mutu Razvan	407	1.6
29	Nastase Andrei	407	3.61
30	Nicolae Alexandra Cristina	407	4.215
31	Paduraru Cristian	407	4.4725
32	Popescu Ioana Livia	407	4.5
33	Preda Florin Alexandru	407	2.1325
34	Raducanu Constantin	407	4.0875
35	Rusu Ilie Alexandru	407	4.155
36	Sasu Alexandru Cristian	407	4.50
37	Serafim Alex	407	3.57
38	Sirbu Oana	407	4.5
39	Stafie Calin	407	3.485
40	Stan Eduard George	407	4.5
41	Stoian Ilinca Maria	407	**wrong format**
42	Stoica Remus	411	1.875
43	Tabacaru Andrei	407	3.555
44	Vasai Ana Maria	407	
45	Vasilescu Costin-Tiberiu	407	4.3825
46	Vrajmas Sanda	407	2.9625

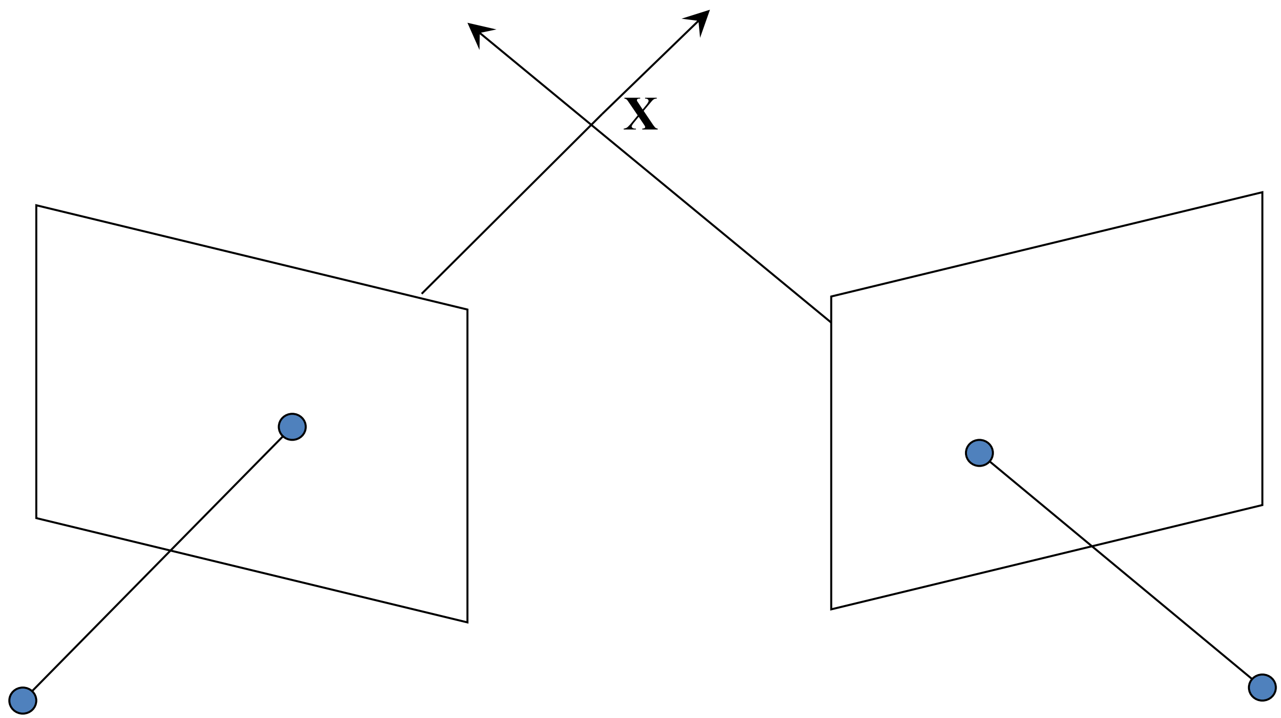
Administrative

- exam date = deadline for project 2 = Tuesday, 25th of June

- Project 2 coming soon (at latest 4th of June) – video analysis

- the plan is to release the Project 2 until week 14, deadline on 25th of June to submit your code/results, presentation from 26th of June in the f2f or online setting.

Last Lecture:
Two–View Geometry

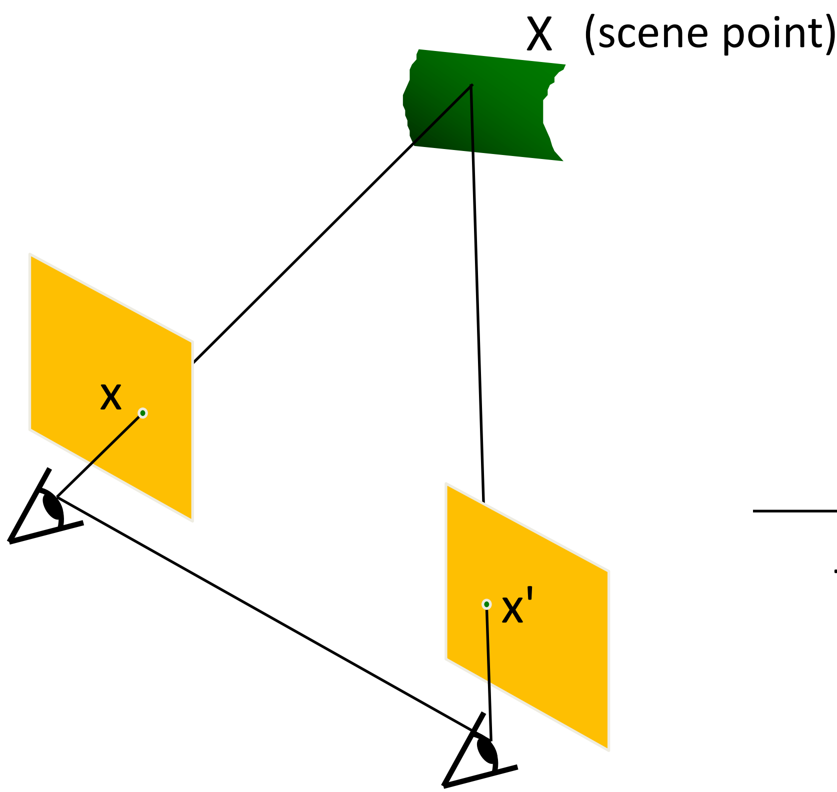


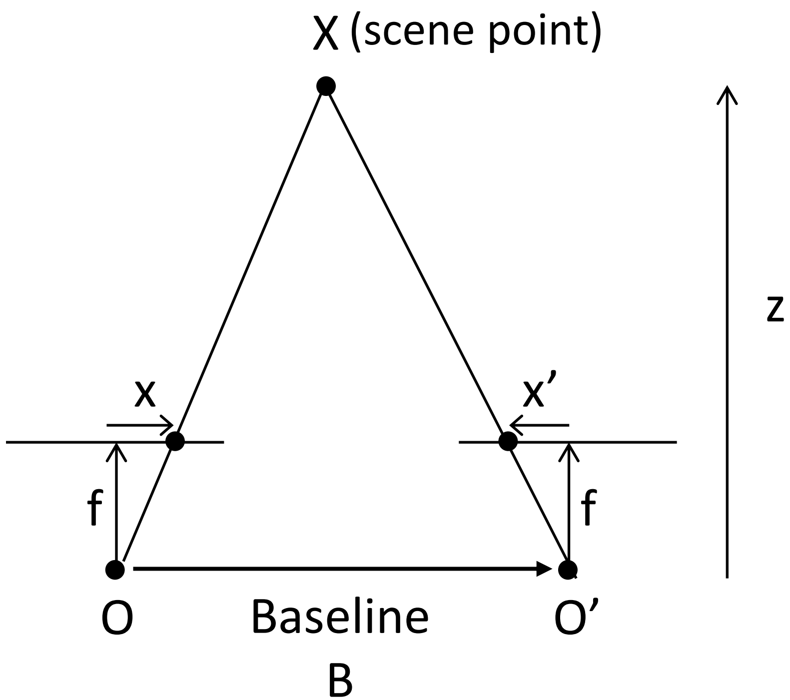
Last Lecture: Two-View Geometry

- Epipolar geometry
  - Relates cameras from two positions
- Stereo depth estimation
  - Recover depth from two images

Last Lecture: Depth from Stereo

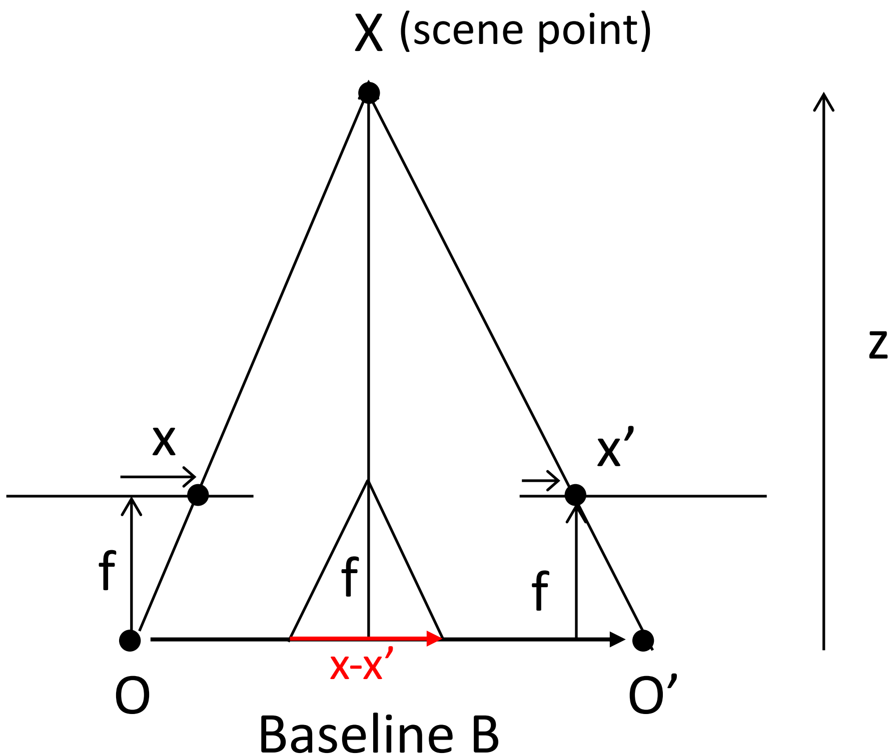
- Goal: recover depth by finding image coordinate x' that corresponds to x





Last Lecture: Depth from Stereo

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

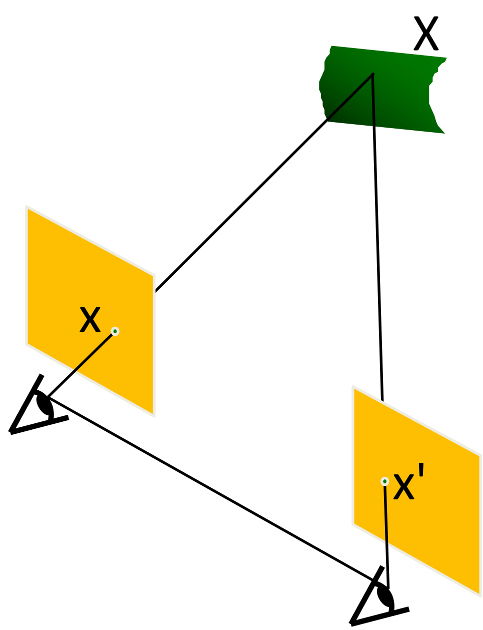


$$disparity = x - x' = \frac{B \cdot f}{z}$$

Disparity is inversely proportional to depth.

Last Lecture: Depth from Stereo

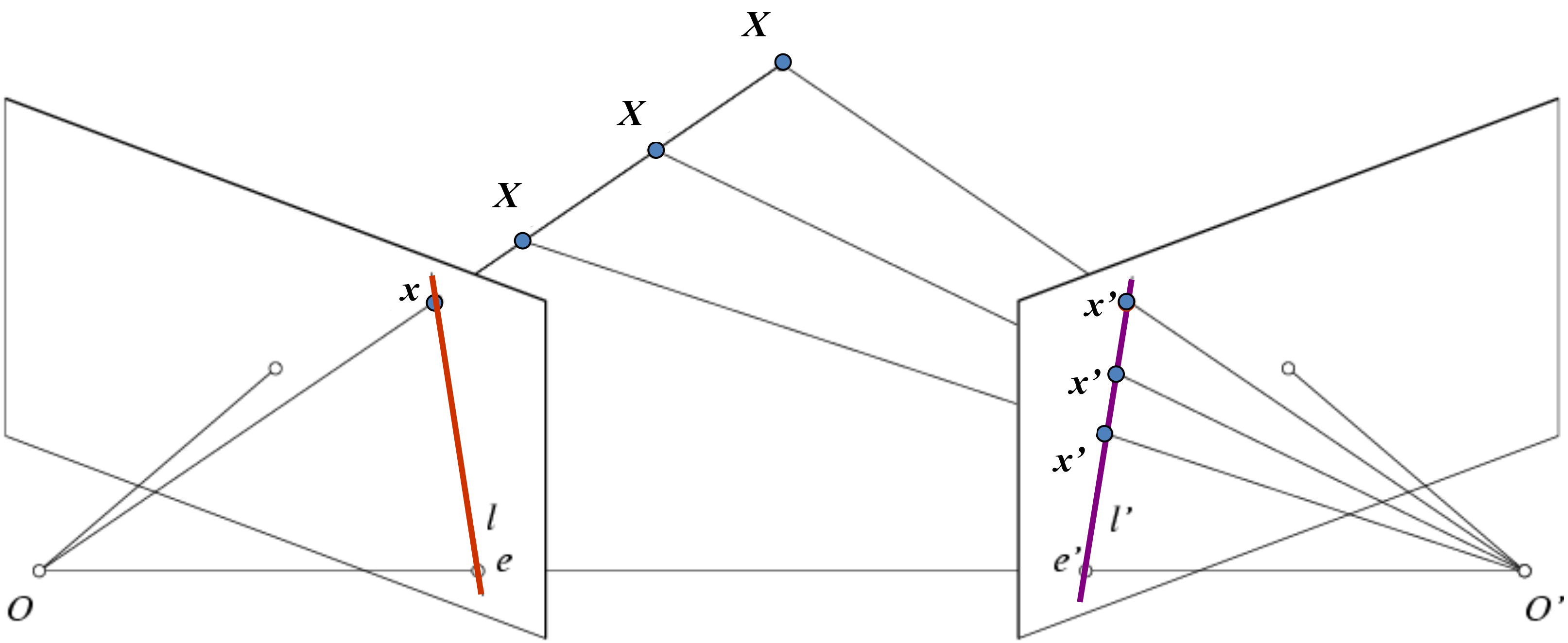
- Goal: recover depth by finding image coordinate x' that corresponds to x
- Sub-Problems
  1. Calibration: How do we recover the relation of the cameras (if not already known)?
  2. Correspondence: How do we search for the matching point x'?



Last Lecture: Correspondence Problem

- We have two images taken from cameras with different intrinsic and extrinsic parameters
- How do we match a point in the first image to a point in the second? How can we constrain our search?

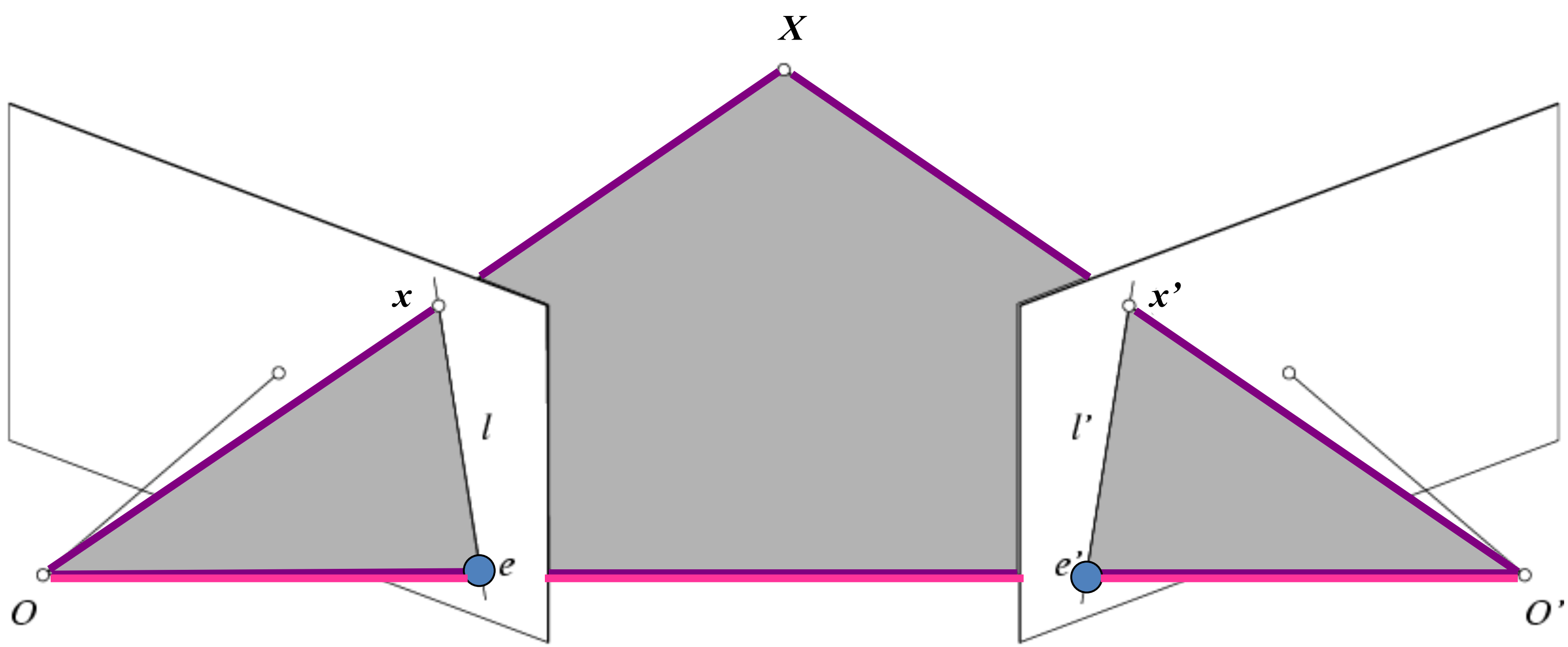
Key idea: Epipolar constraint



Potential matches for *x* have to lie on the corresponding line *l'*.

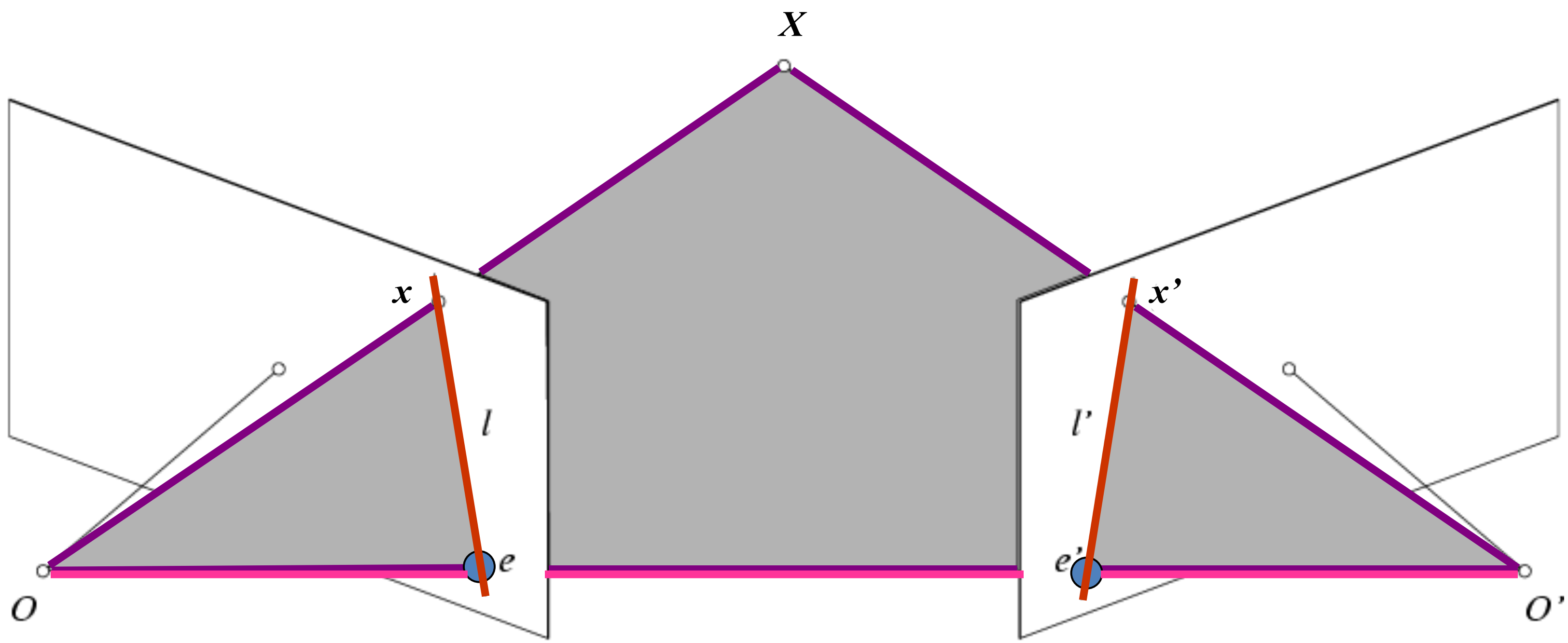
Potential matches for *x'* have to lie on the corresponding line *l*.

Epipolar geometry: notation



- **Baseline** – line connecting the two camera centers
- **Epipoles**
= intersections of baseline with image planes
= projections of the other camera center
- **Epipolar Plane** – plane containing baseline (1D family)

Epipolar geometry: notation



- **Baseline** – line connecting the two camera centers
- **Epipoles**

= intersections of baseline with image planes
= projections of the other camera center

- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$\mathbf{K} = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix}\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$

- Extrinsic parameters
  - Rotation and translation relative to world coordinate system
  - What is the projection of the camera center?

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\widetilde{\mathbf{C}}\end{bmatrix}$$

coords. of camera center in world frame

$$\mathbf{PC} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\widetilde{\mathbf{C}}\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{C}} \\ 1\end{bmatrix} = 0$$

The camera center is the *null space* of the projection matrix!

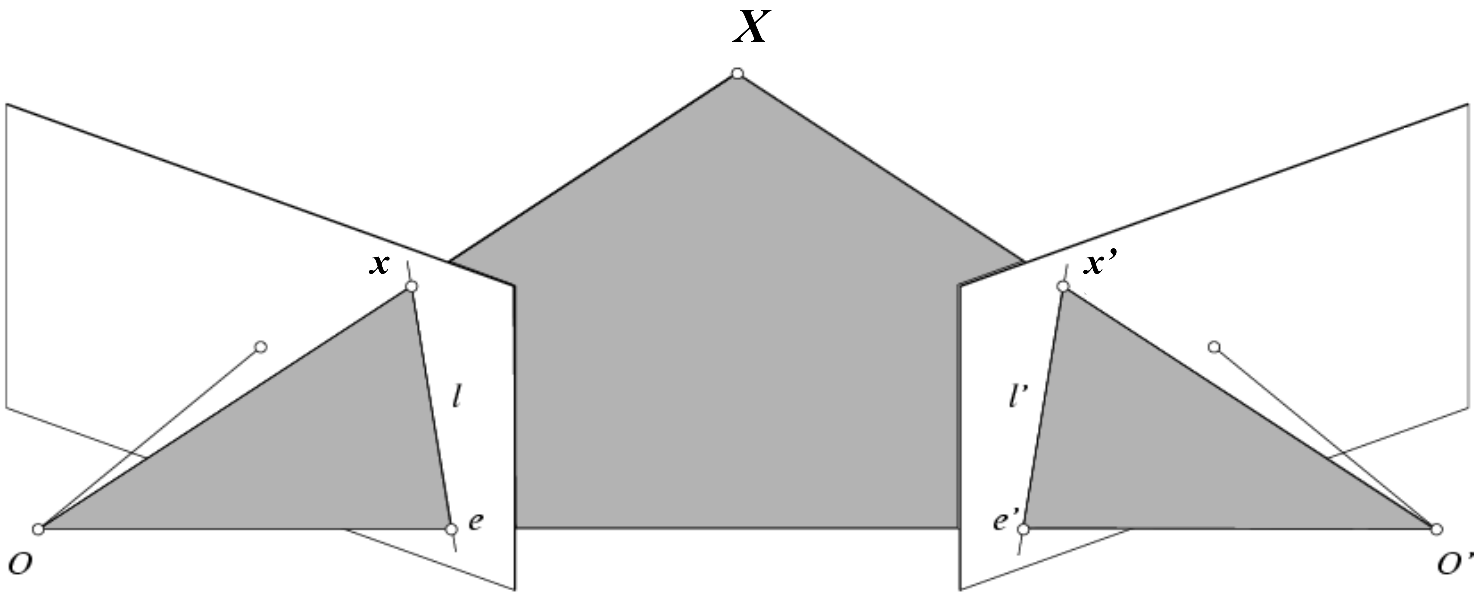
Camera parameters $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]\mathbf{X}$$

$$\Downarrow$$

$$\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

Epipolar constraint: Calibrated case



Given the intrinsic parameters of the cameras:

1. Convert to normalized coordinates by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates

for some scale factor

$$\hat{x} = K^{-1} x = X$$

$$\hat{x}' = K'^{-1} x' = X'$$

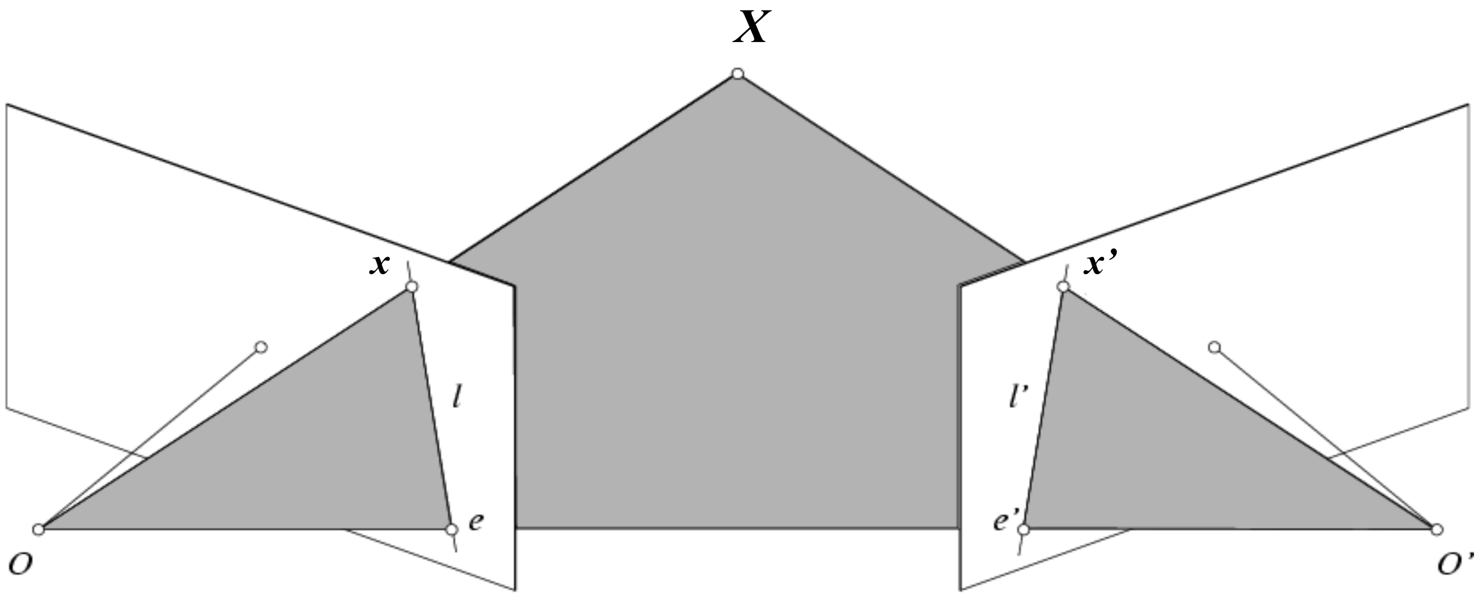
Homogeneous 2d point (3D ray towards X)

2D pixel coordinate (homogeneous)

3D scene point

3D scene point in 2nd camera's 3D coordinates

Epipolar constraint: Calibrated case



Given the intrinsic parameters of the cameras:

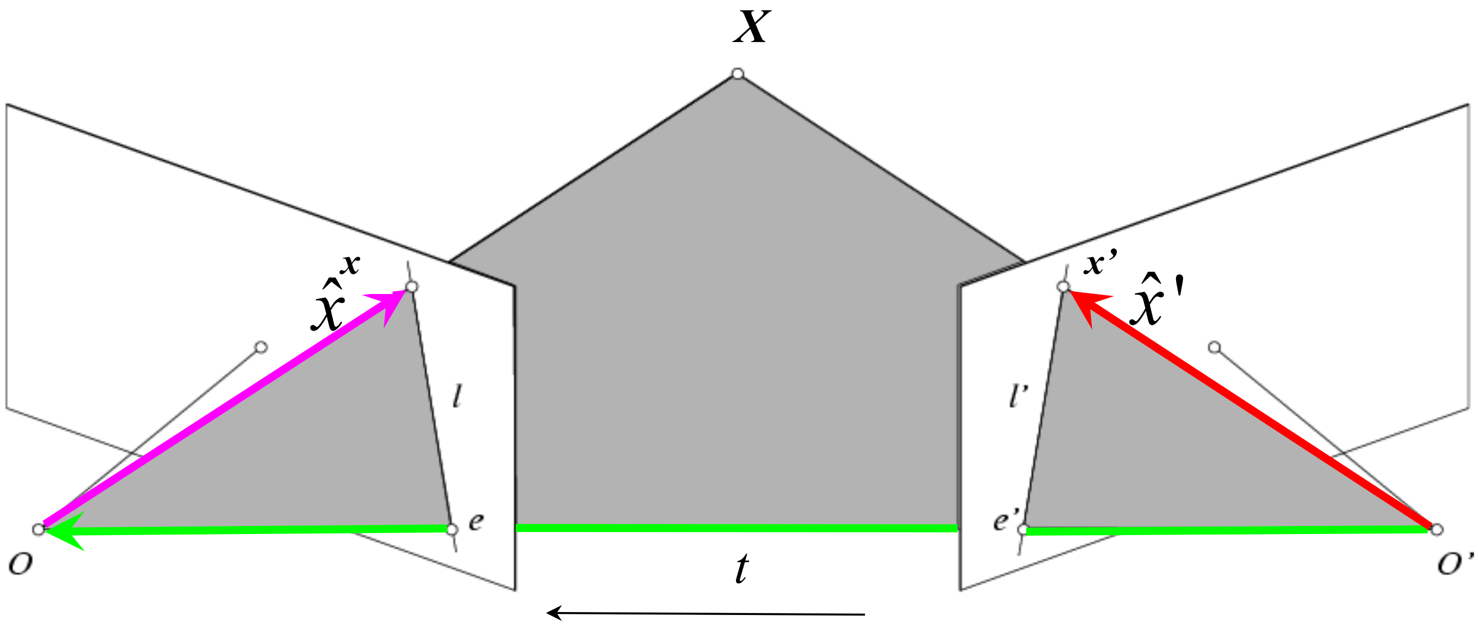
1. Convert to normalized coordinates by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates
2. Define some *R* and *t* that relate X to X' as below

for some scale factor

$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t$$

Epipolar constraint: Calibrated case

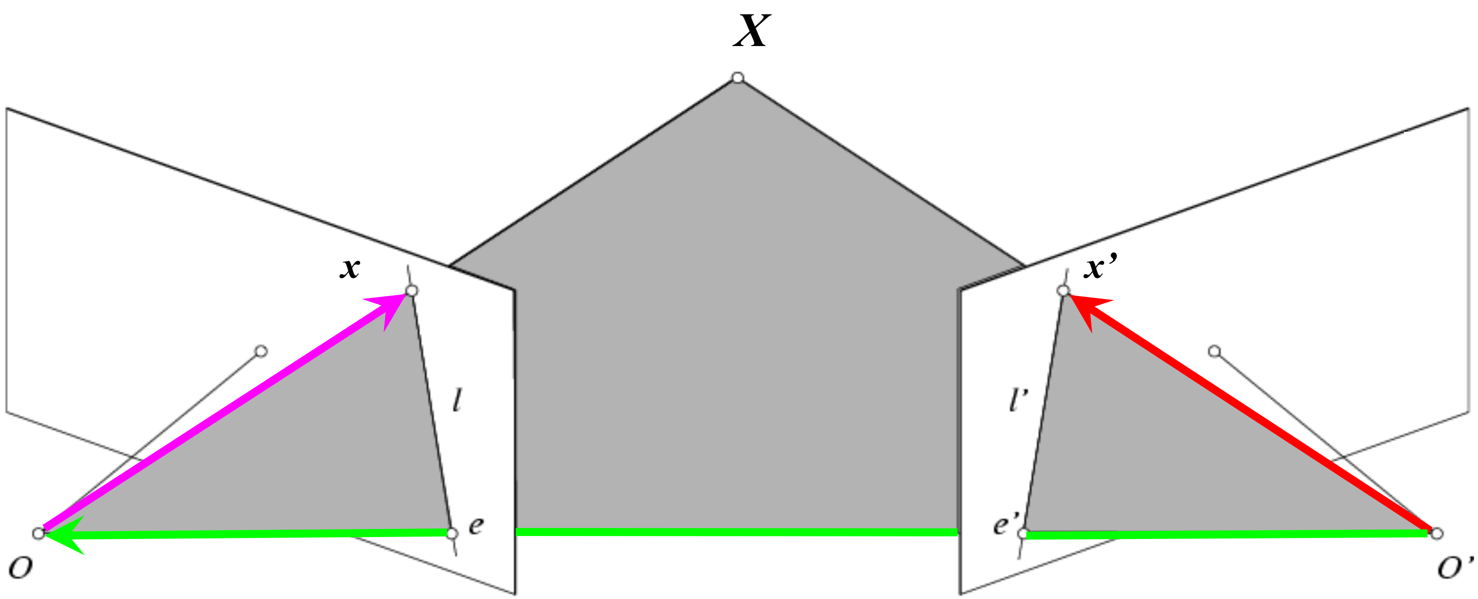


$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t \quad \Rightarrow \quad \hat{x} \cdot [t \times (R\hat{x}')] = 0$$

(because x, $R\hat{x}'$ and t are co-planar)

Last Lecture: Essential matrix



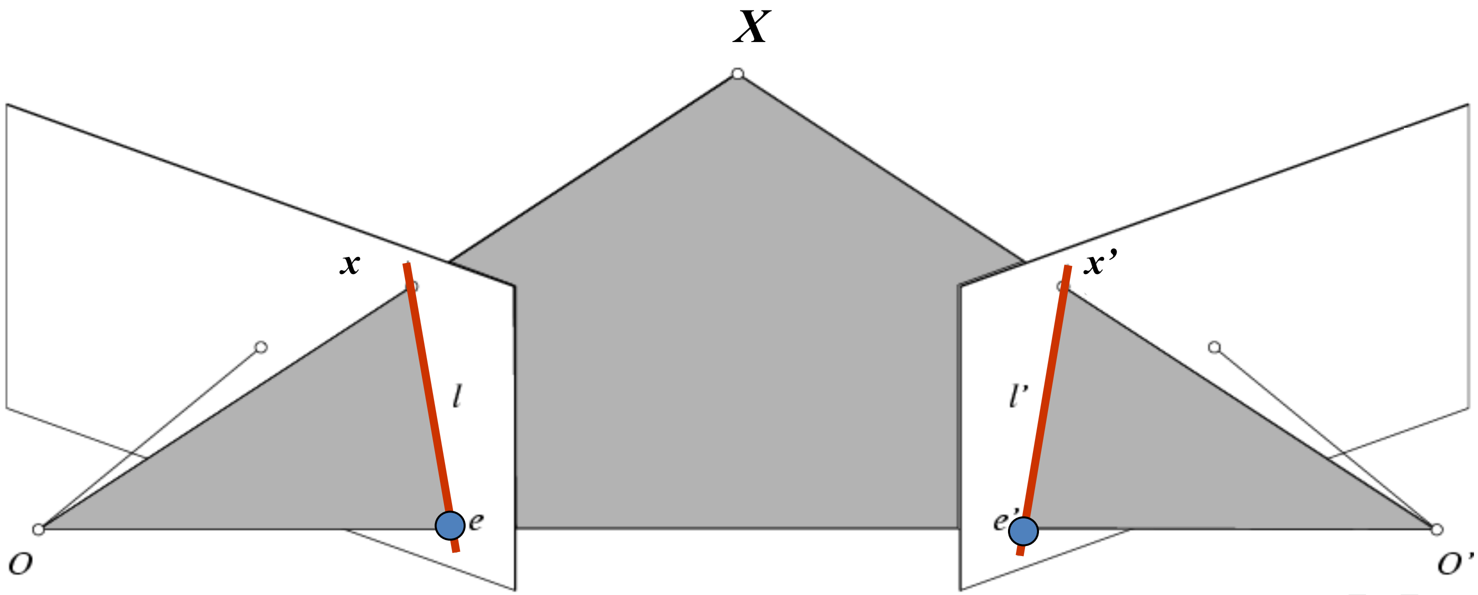
$$\hat{x} \cdot [t \times (R\hat{x}')] = 0 \quad \Rightarrow \quad \hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t]_\times R$$

$$[t]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

$[t]_\times$ is the skew symmetric matrix of $t=(t_1,t_2,t_3)$

Essential Matrix
(Longuet-Higgins, 1981)

Properties of the Essential matrix



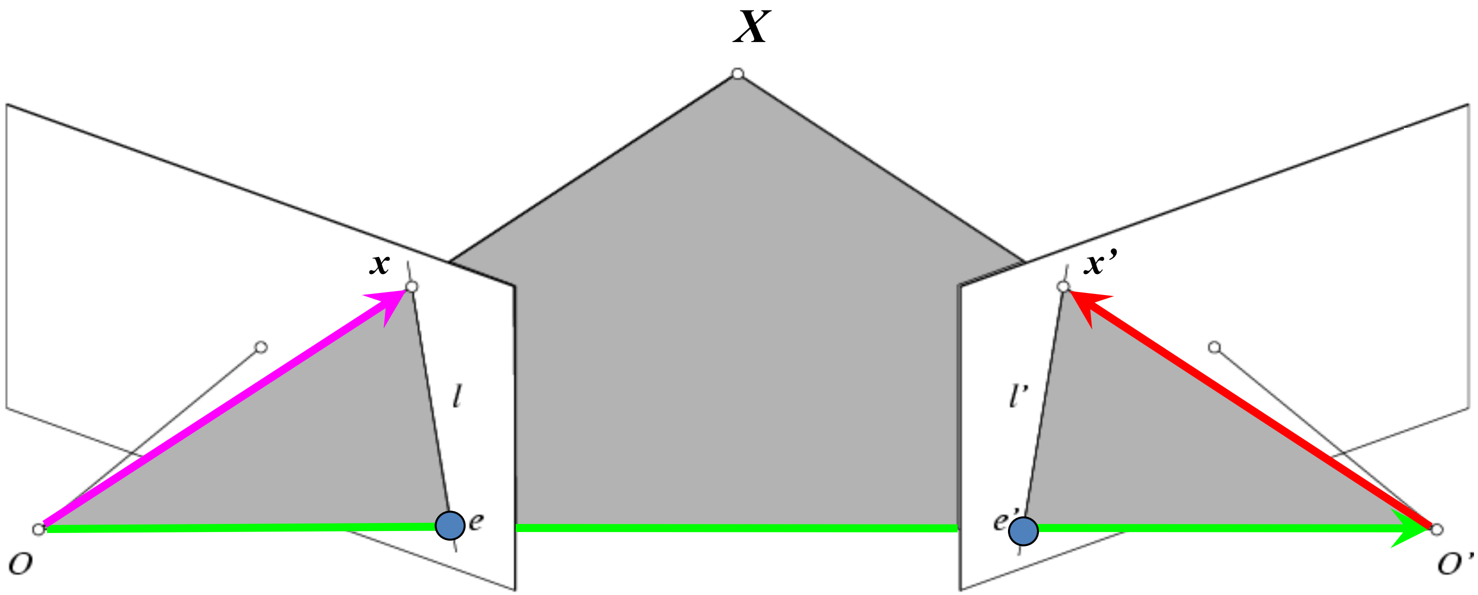
$$\hat{x} \cdot [t \times (R\hat{x}')] = 0 \quad \Rightarrow \quad \hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = \left[t\right]_{\times} R$$

Drop ^ below to simplify notation

Skew-symmetric matrix

- $E\,x'$ is the epipolar line associated with x' ($l = E\,x'$)
- $E^T x$ is the epipolar line associated with x ($l' = E^T x$)
- $E\,e' = 0$ and $E^T e = 0$ (*e and e' are on each l and l'*)
- E is singular (rank two – is determinant = 0)
- E has five degrees of freedom
  - (3 for R, 2 for t because it's up to a scale)

Epipolar constraint: Uncalibrated case



- If we don't know *K* and *K'*, then we can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$\hat{x}^T E \hat{x}' = 0 \qquad\qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

Last Lecture: The Fundamental Matrix

Without knowing K and K', we can define a similar relation using *unknown* normalized coordinates

$$\hat{x}^T E \hat{x}' = 0$$

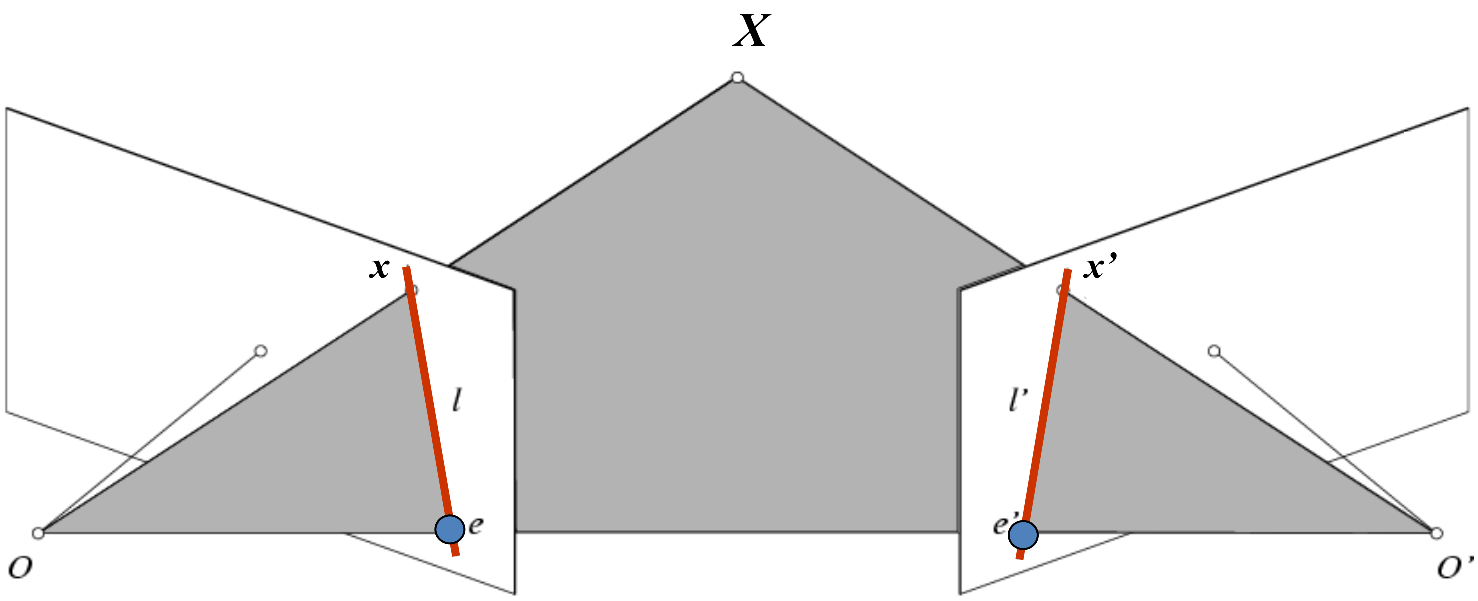
$$\hat{x} = K^{-1} x$$

$$\hat{x}' = K'^{-1} x'$$

$$\Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Properties of the Fundamental matrix

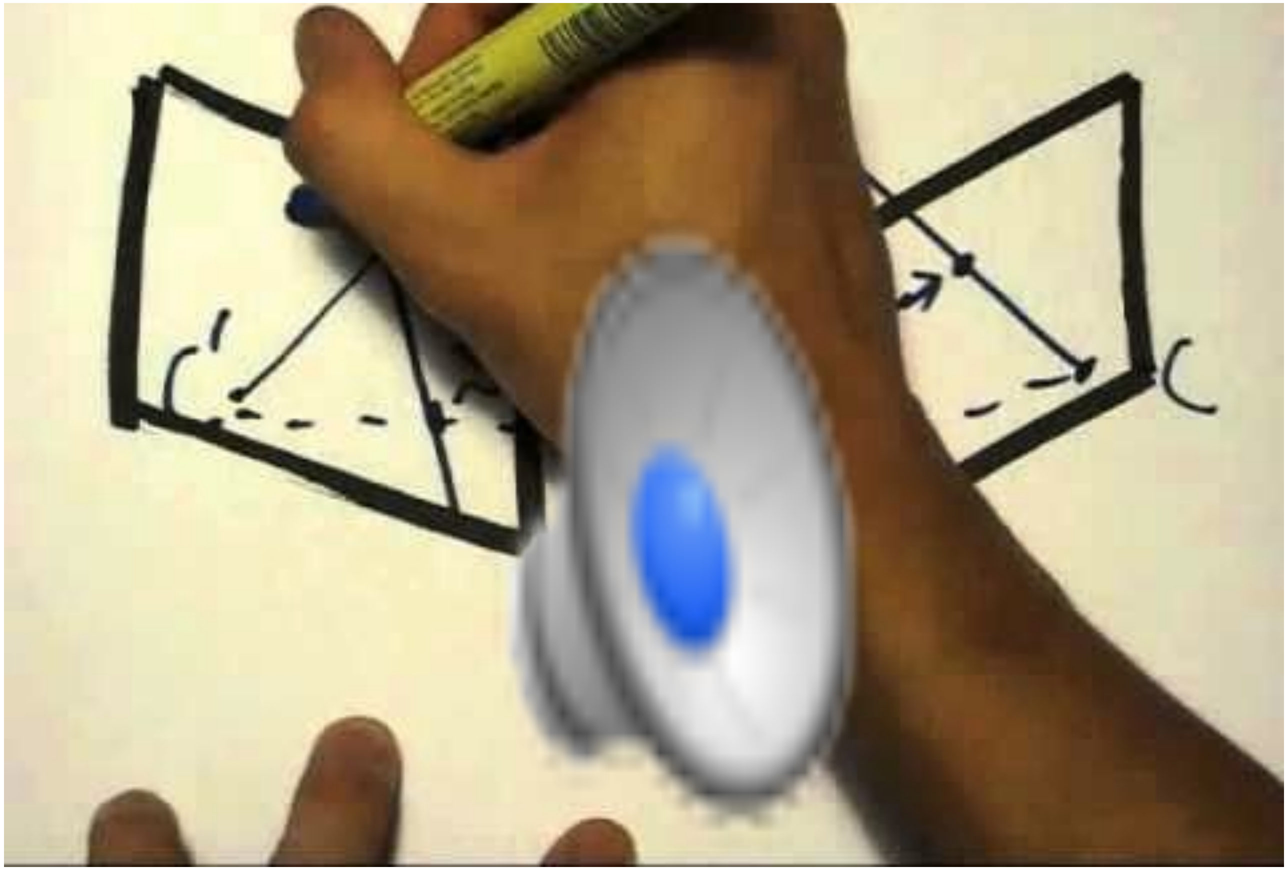


$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- $F x'$ is the epipolar line associated with x' $(l = F x')$
- $F^T x$ is the epipolar line associated with x $(l' = F^T x)$
- $F e' = 0$ and $F^T e = 0$
- F is singular (rank two): det(F)=0
- F has seven degrees of freedom: 9 entries but defined up to scale, det(F)=0

Estimating the fundamental matrix

The Fundamental Matrix Song



http://danielwedge.com/fmatrix/

Laboratory class 6

The goal of this lab is to introduce you to camera and scene geometry. Specifically we will estimate the camera projection matrix, which maps 3D world coordinates to image coordinates, as well as the fundamental matrix, which relates points in one scene to epipolar lines in another. The camera projection matrix and the fundamental matrix can each be estimated using point correspondences. To estimate the projection matrix (camera calibration), the input is corresponding 3D and 2D points. To estimate the fundamental matrix the input is corresponding 2D points across two images. We start by estimating the projection matrix and the fundamental matrix for a scene with ground truth correspondences. Then we'll move on to estimating the fundamental matrix using point correspondences from SIFT.

Tutorial on epipolar geometry is here: https://docs.opencv.org/master/da/de9/tutorial_py_epipolar_geometry.html or here: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_epipolar_geometry/py_epipolar_geometry.html

Data

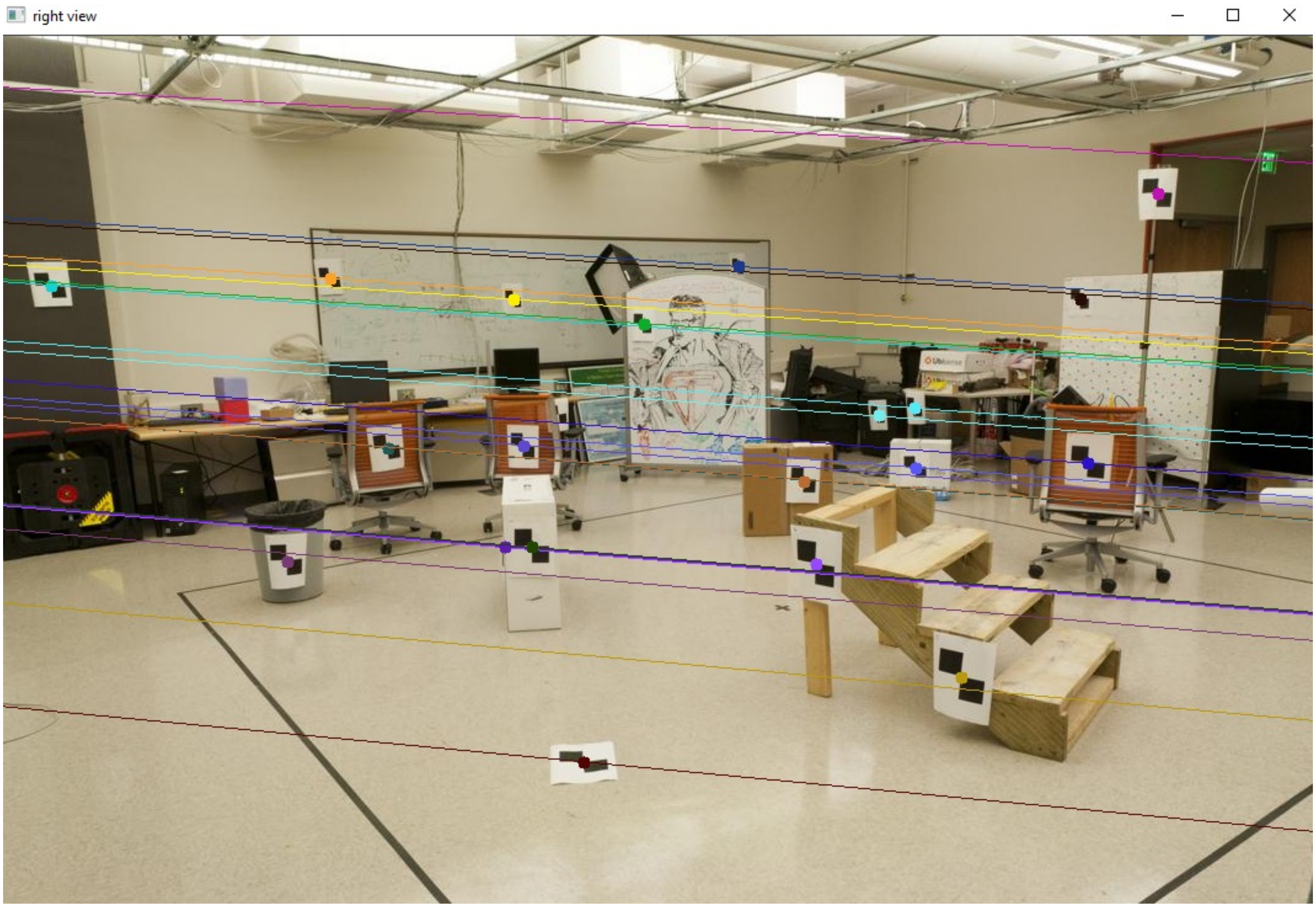
We provide 2D and 3D ground truth point correspondences for the base image pair (pic_a.jpg and pic_b.jpg), as well as other images which will not have any ground truth dataset.

Image 1 of a scene	Image 2 of a scene
	

This laboratory consists of three parts: (1) estimating the projection matrix, (2) estimating the fundamental matrix, (3) estimating the fundamental matrix with unreliable SIFT matches using RANSAC.

Laboratory class 6

Laboratory class 6



Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection, template matching

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

4. **Object Recognition: high – level vision**

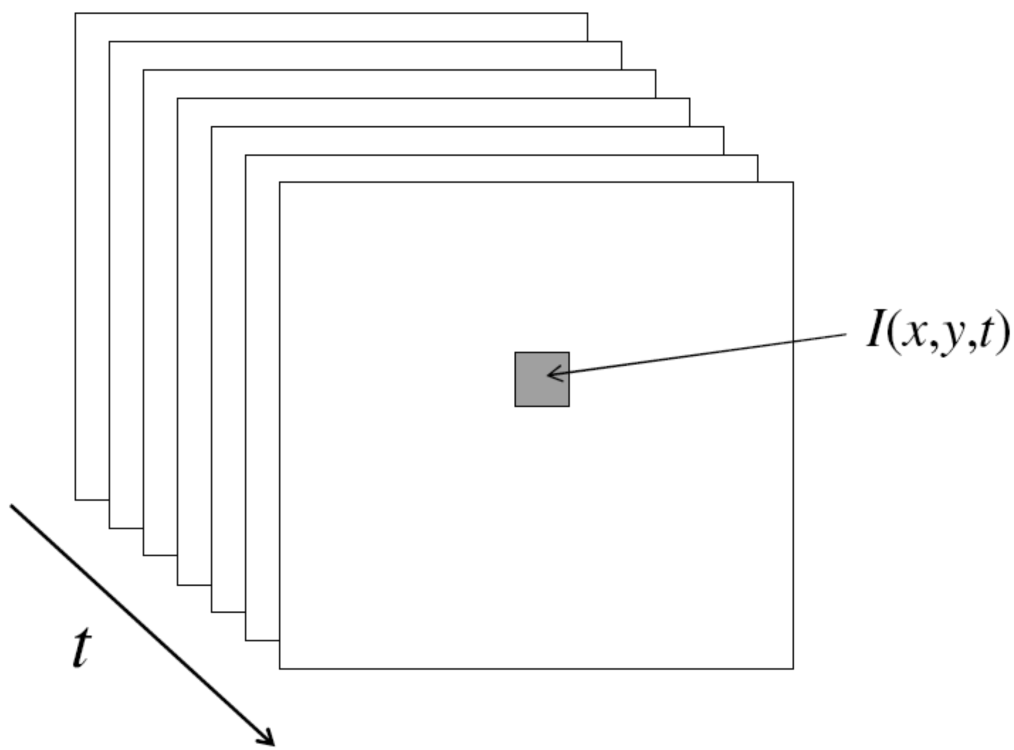
   Object classification, object detection, part based models, bovw models

5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

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)



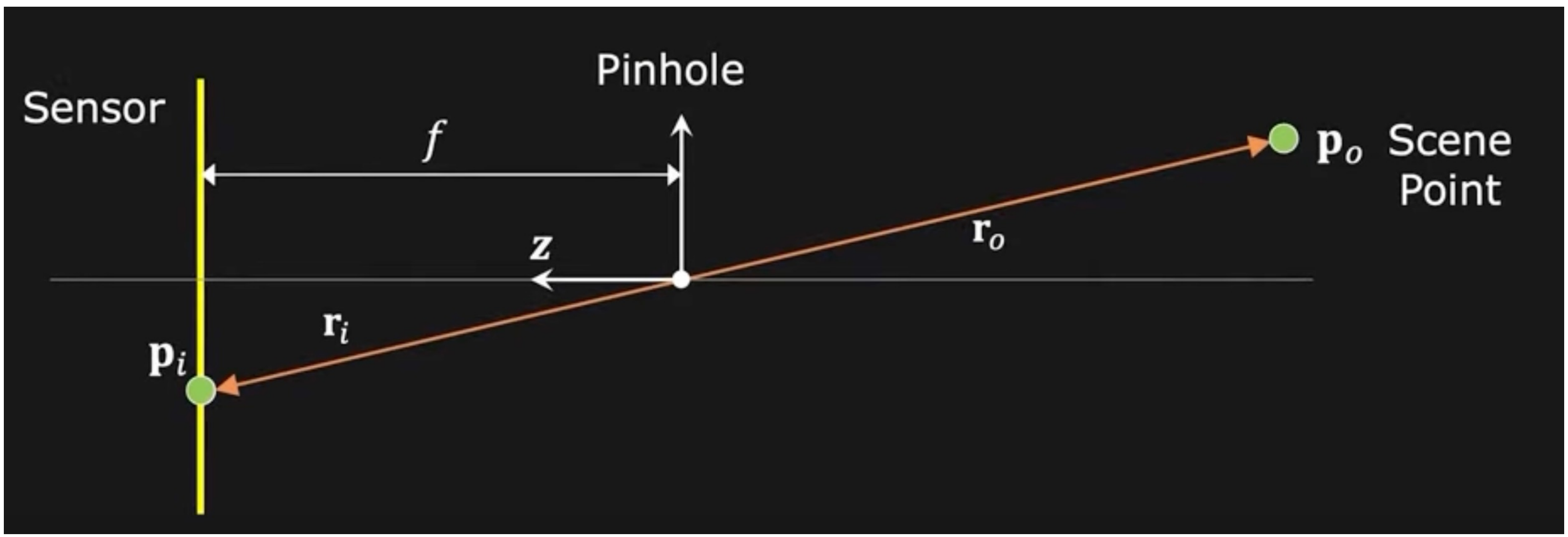
Motion Field and Optical Flow

- Single views vs multiple views of the same stationary scene (objects not moving)

- Usually objects move in the scene

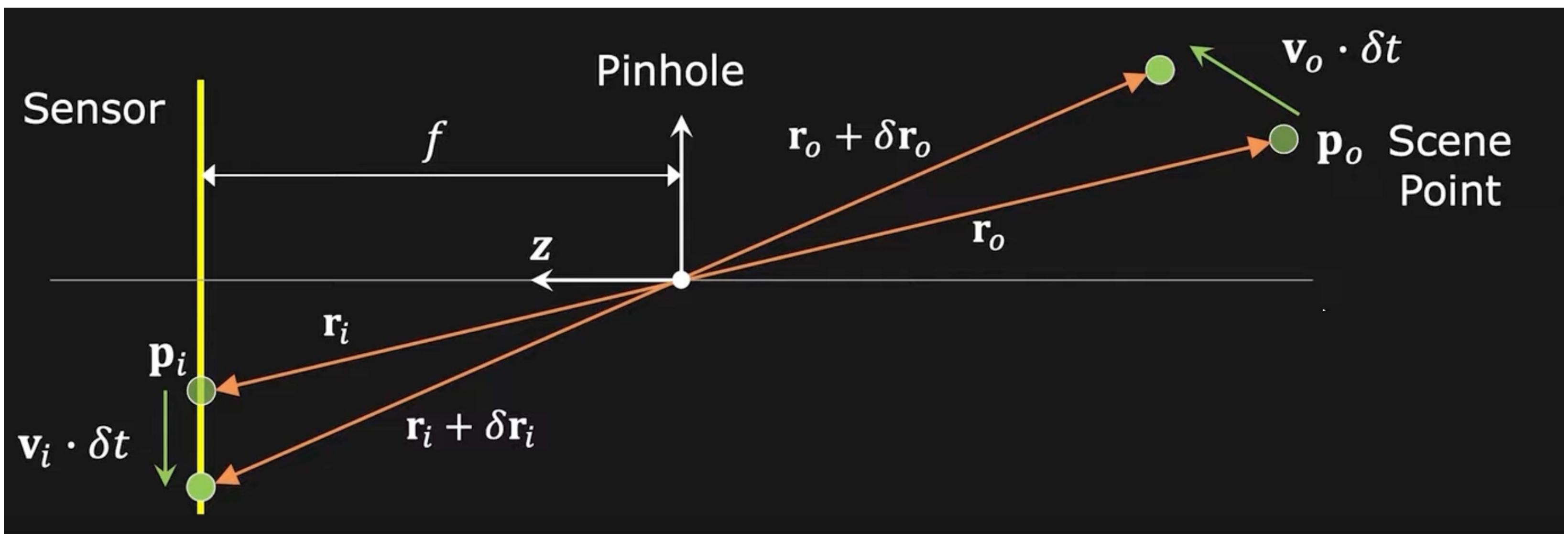
Motion Field and Optical Flow

- Single views vs multiple views of the same stationary scene (objects not moving)
- Usually objects move in the scene



Motion Field and Optical Flow

- Single views vs multiple views of the same stationary scene (objects not moving)
- Usually objects move in the scene



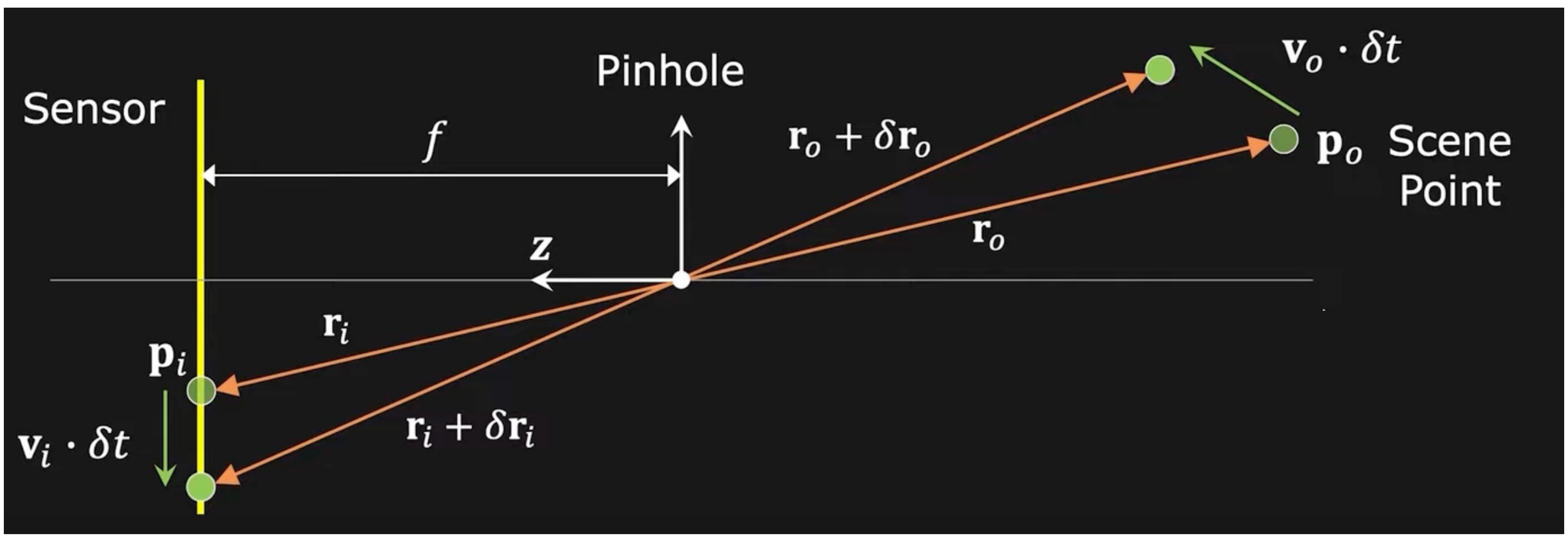
Motion Field and Optical Flow

$$\mathbf{v}_0 = \frac{d\mathbf{r}_0}{dt} \qquad \mathbf{v}_\mathbf{i} = \frac{d\mathbf{r}_i}{dt}$$

$\mathbf{v_0}$ – scene point velocity

$\mathbf{v_i}$ – image point velocity (motion field)

$\mathbf{r}_0$ related to $\mathbf{r}_i$ by $\dfrac{\mathbf{r}_i}{f} = \dfrac{\mathbf{r}_0}{\mathbf{r}_0 \cdot \hat{\mathbf{z}}_0}$



Motion Field and Optical Flow

- Single views vs multiple views of the same stationary scene (objects not moving)
- Usually objects move in the scene
- Motion Field: Projection of 3D relative velocity vectors onto 2D image plane
- Optic flow: Observed 2D displacements of intensity patterns in the image.
- We want to know motion field, by estimating optical flow.

The problem of motion estimation

Assuming that illumination does not change:

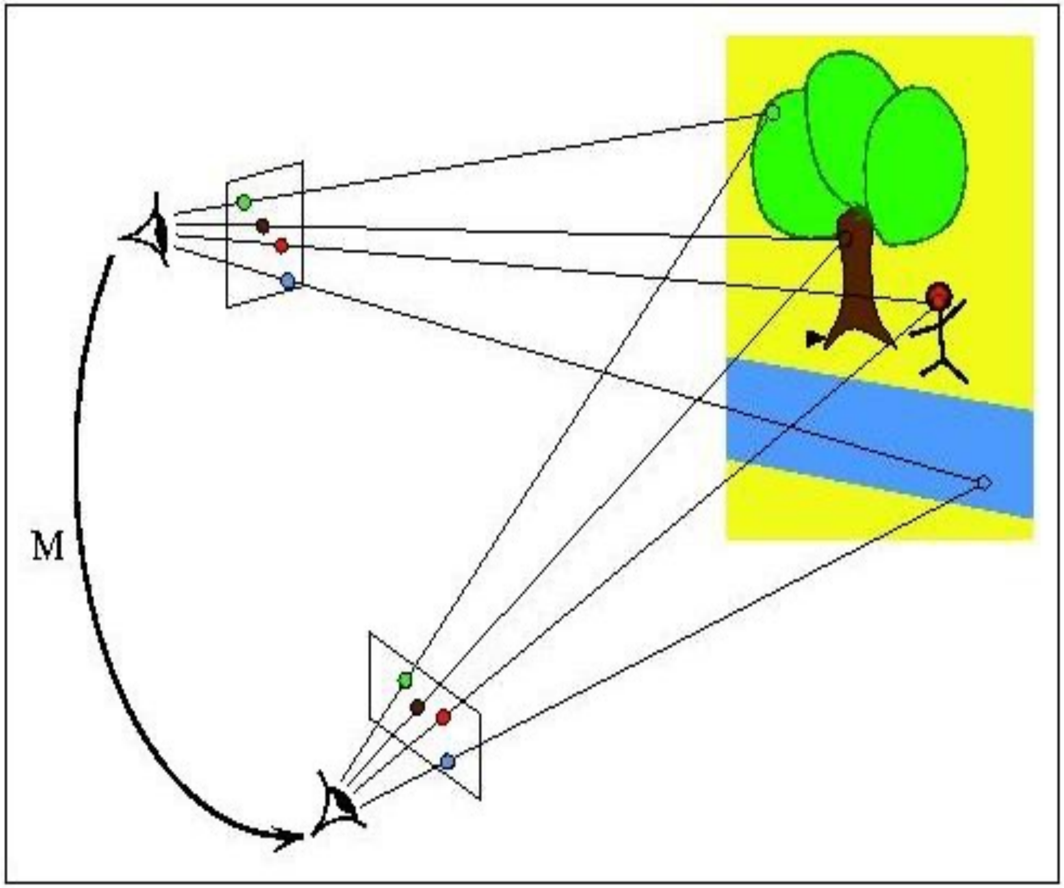
• Image changes are due to the RELATIVE MOTION between the scene and the camera.

• There are 3 possibilities:

– Camera still, moving scene

– Moving camera, still scene

– Moving camera, moving scene



Optical flow

- Optical flow = method to estimate the apparent motion of scene points from a sequence of images

(1) Motion field vs Optical flow

(2) Optical flow constraint equation

(3) Lucas – Kanade Method

(4) Large motion: coarse-to-fine flow estimation

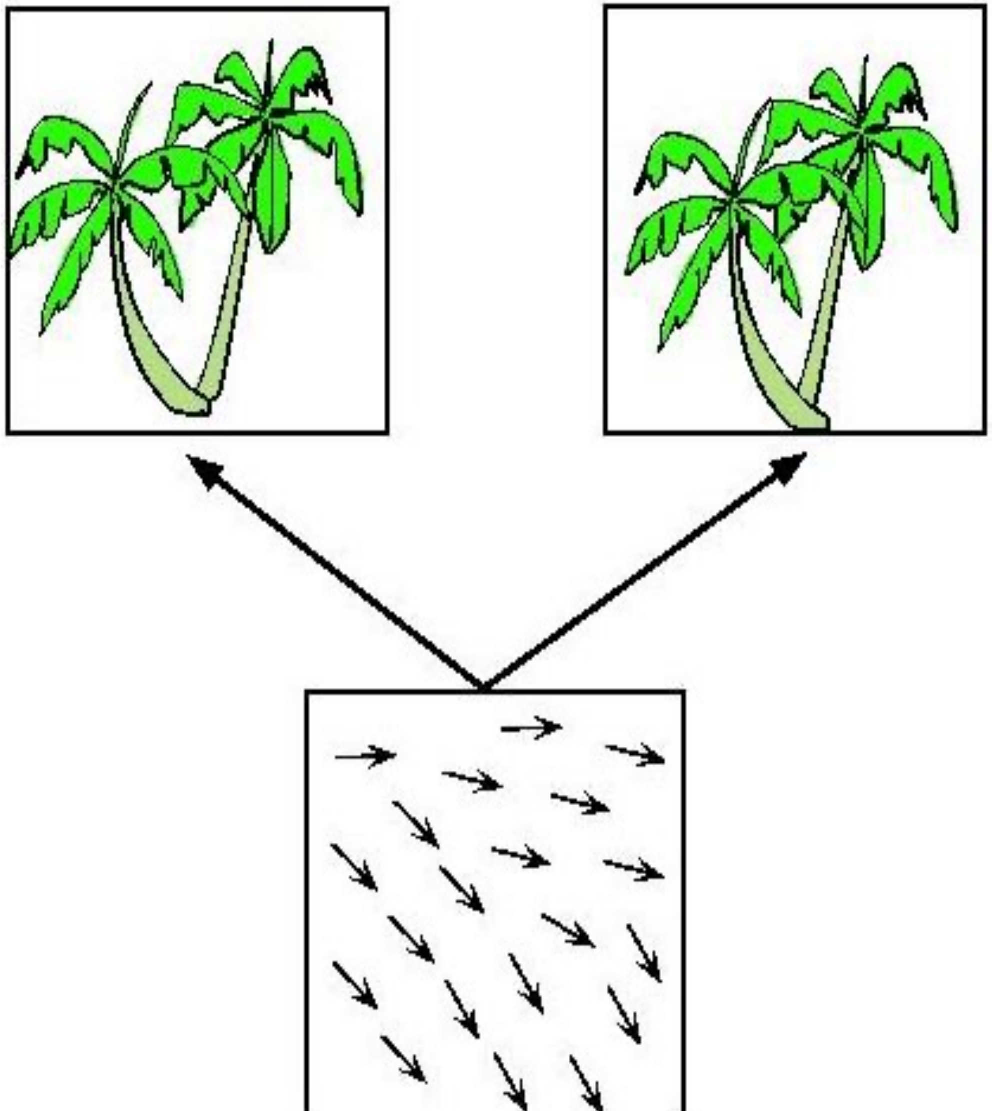
Optical flow

- Where do pixels move to?

Optical flow

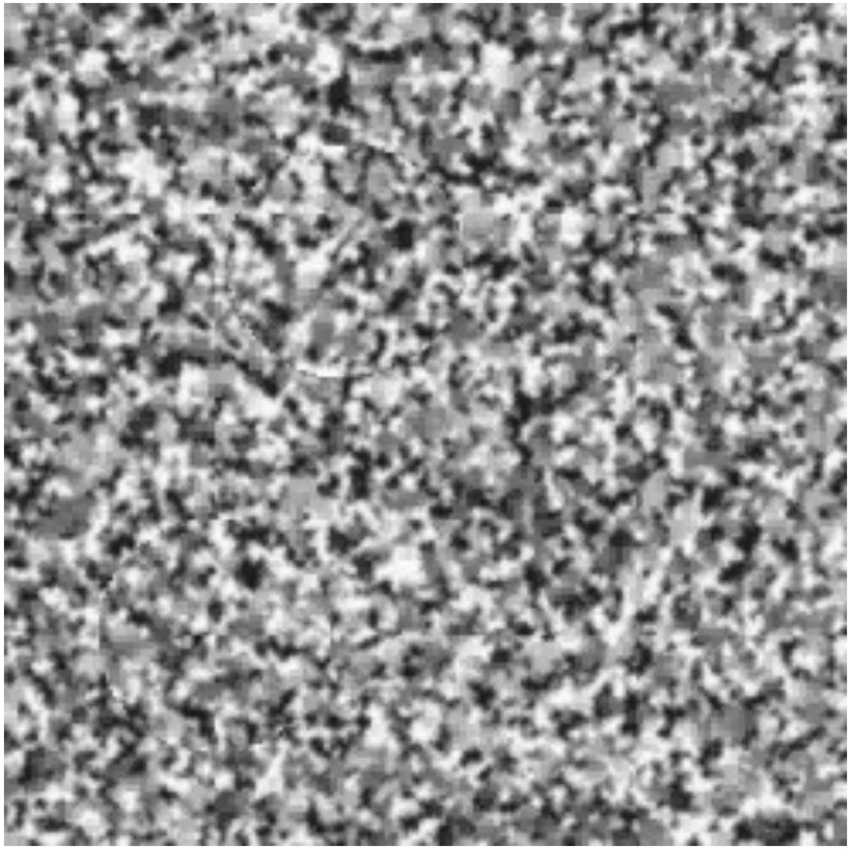
- Where do pixels move to?

Optical flow



Motion is a powerful perceptual cue

- Sometimes, it is the only cue



Motion is a powerful perceptual cue

- 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 is a powerful perceptual cue

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

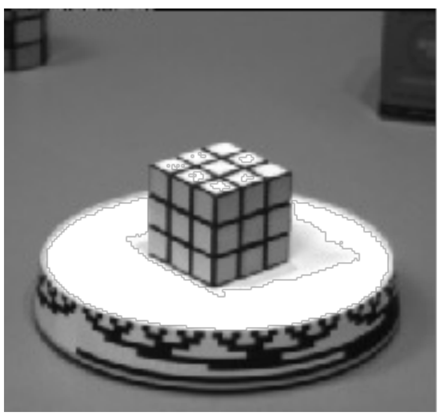
G. Johansson, "Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

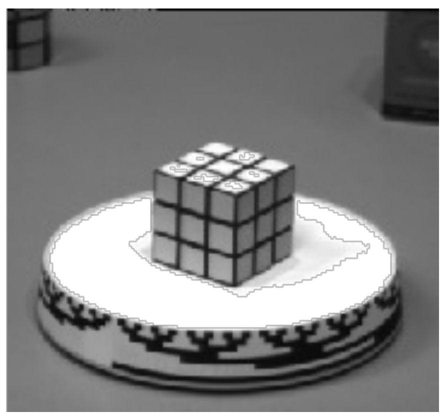
Uses of motion in computer vision

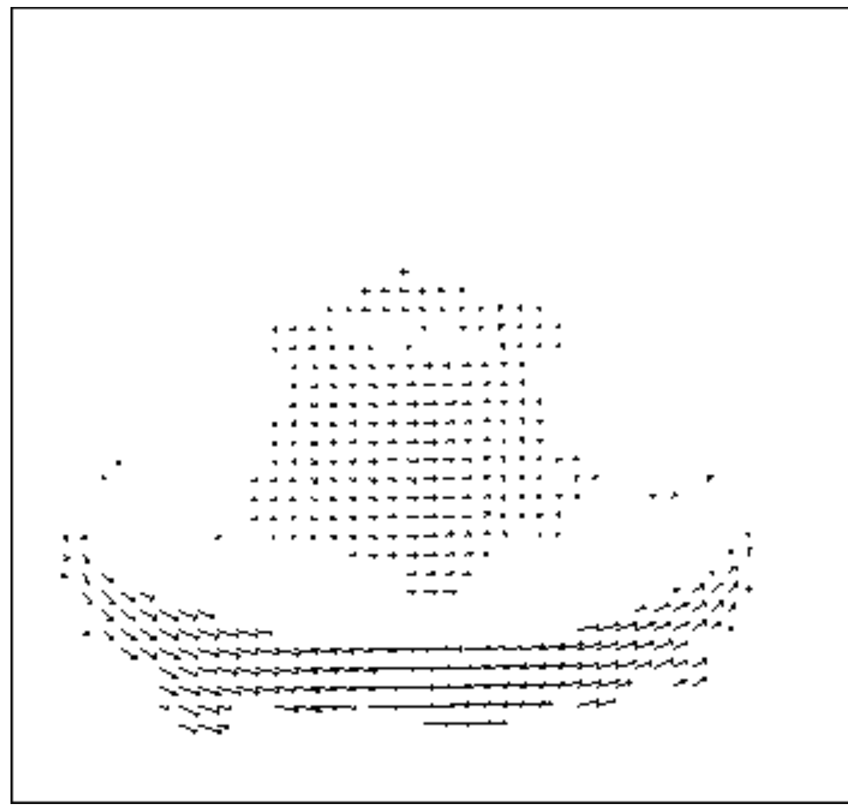
- 3D shape reconstruction
- Object segmentation
- Learning and tracking of dynamical models
- Event and activity recognition
- Self-supervised and predictive learning
- Solving project 2 ☺ (most probable)

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

Apparent motion != motion field

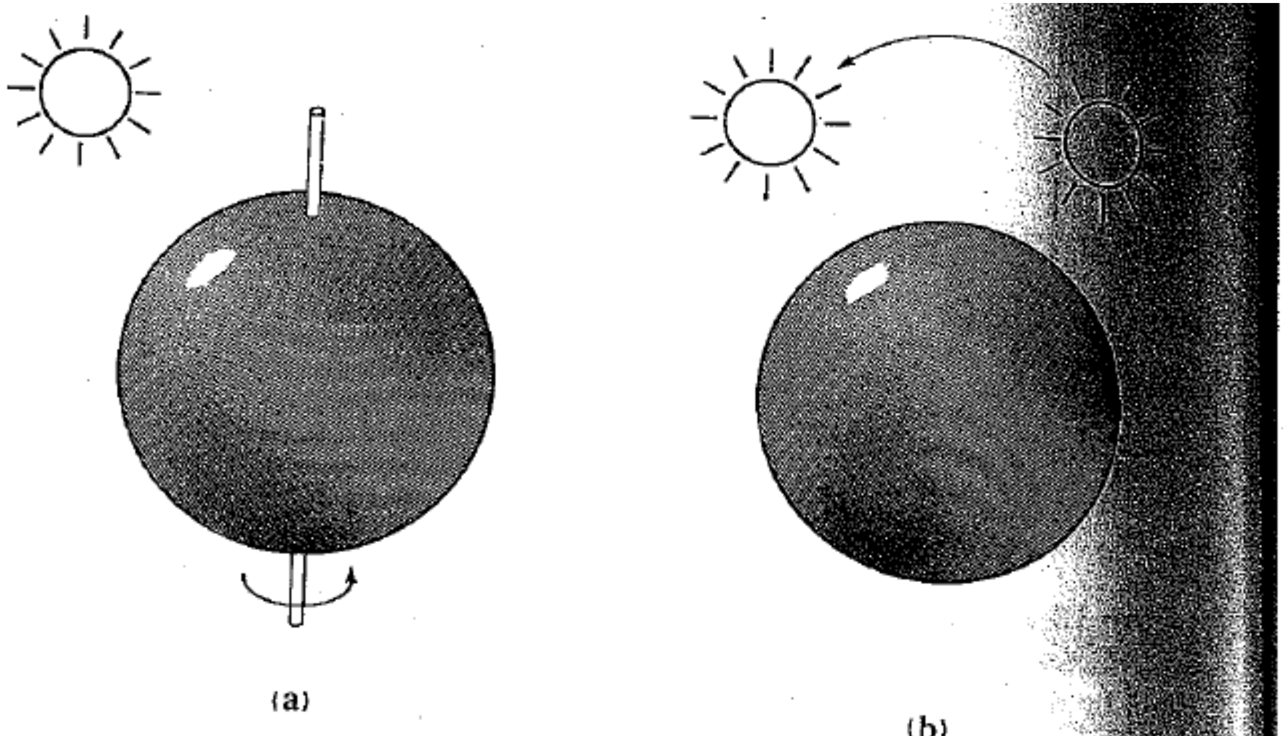
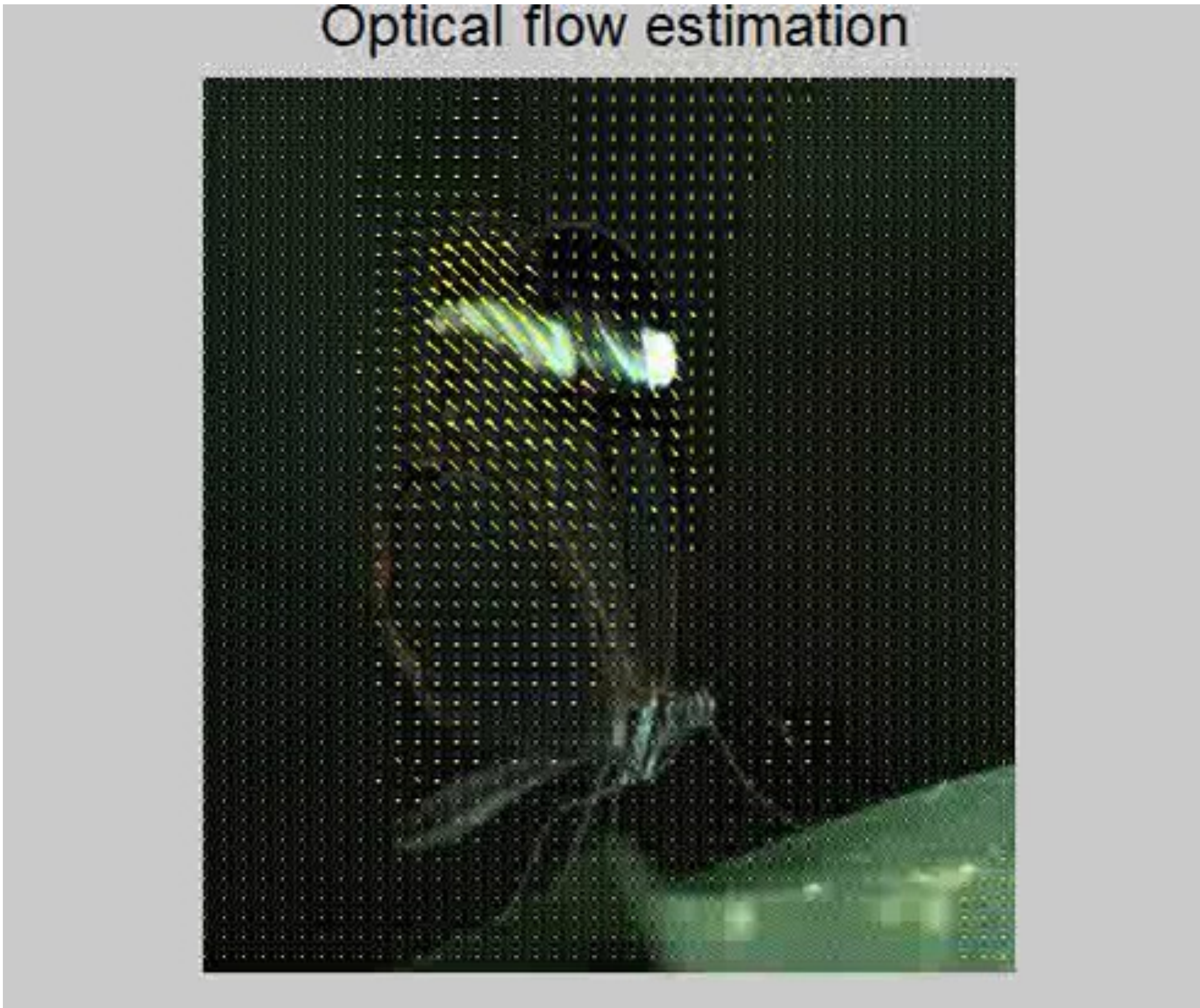


Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field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

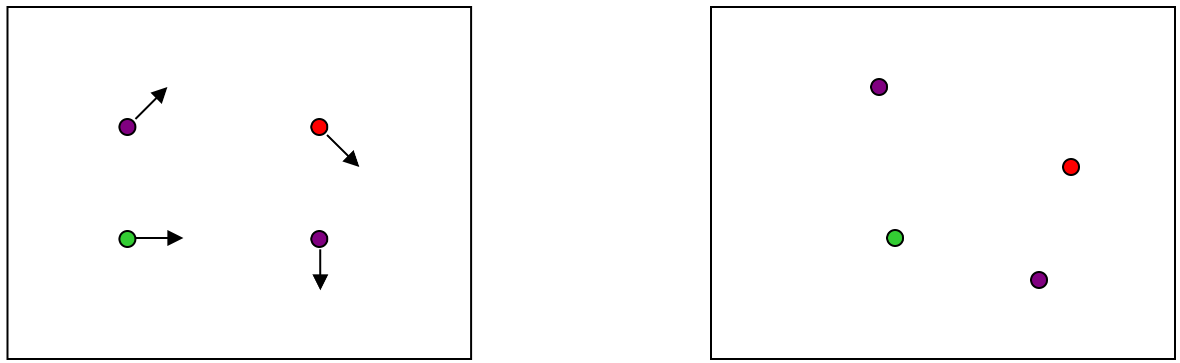
Apparent motion can be caused by lighting changes without any actual motion. Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

Optical flow

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



Estimating optical flow

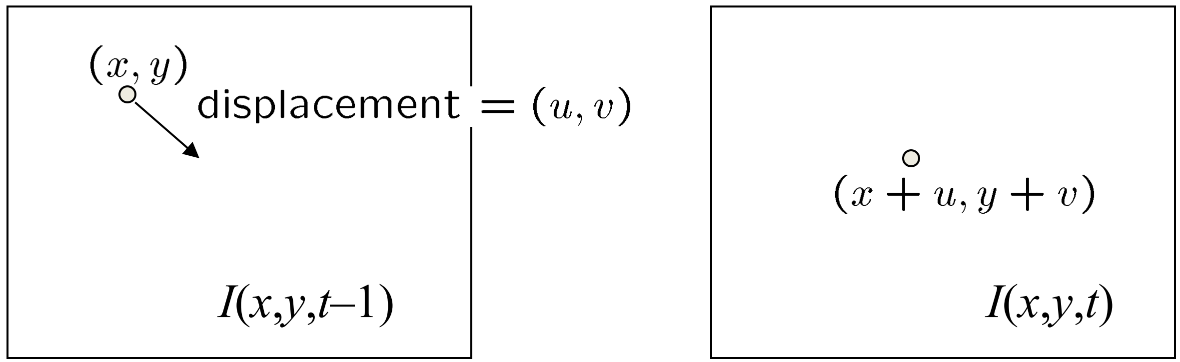


$I(x,y,t\text{-}1)$ $\quad$ $I(x,y,t)$

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

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

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

- What does this constraint mean?

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

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

The brightness constancy constraint

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

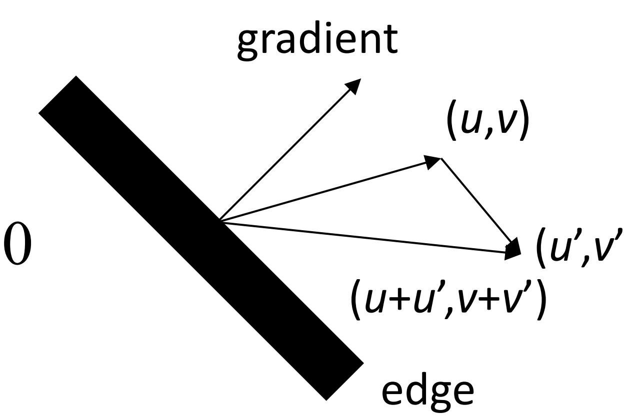
- How many equations and unknowns per pixel?
  - One equation, two unknowns

- What does this constraint mean?

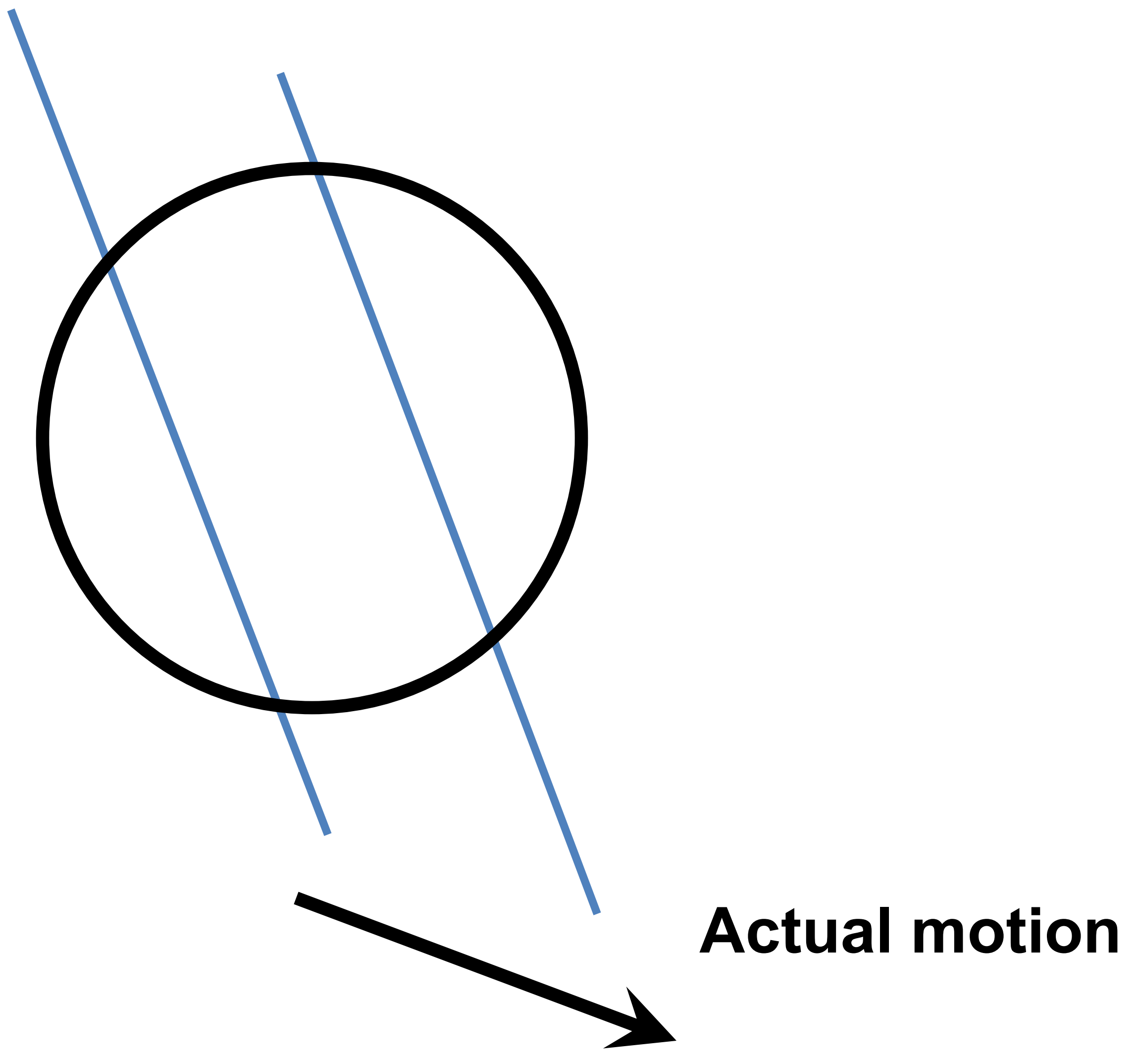
$$\nabla I \cdot (u, v) + I_t = 0$$

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

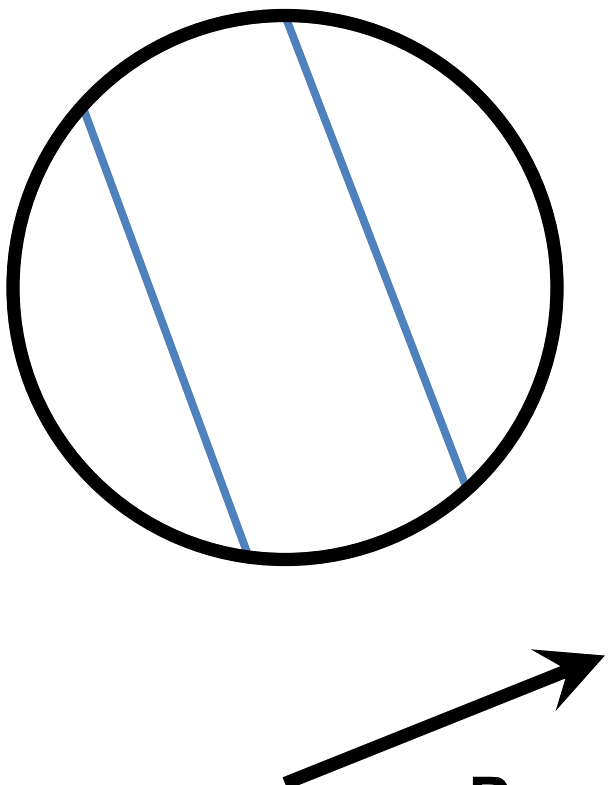
If (u, v) satisfies the equation,
so does $(u+u', v+v')$ if $\nabla I \cdot (u', v') = 0$



The aperture problem



The aperture problem



The barber pole illusion

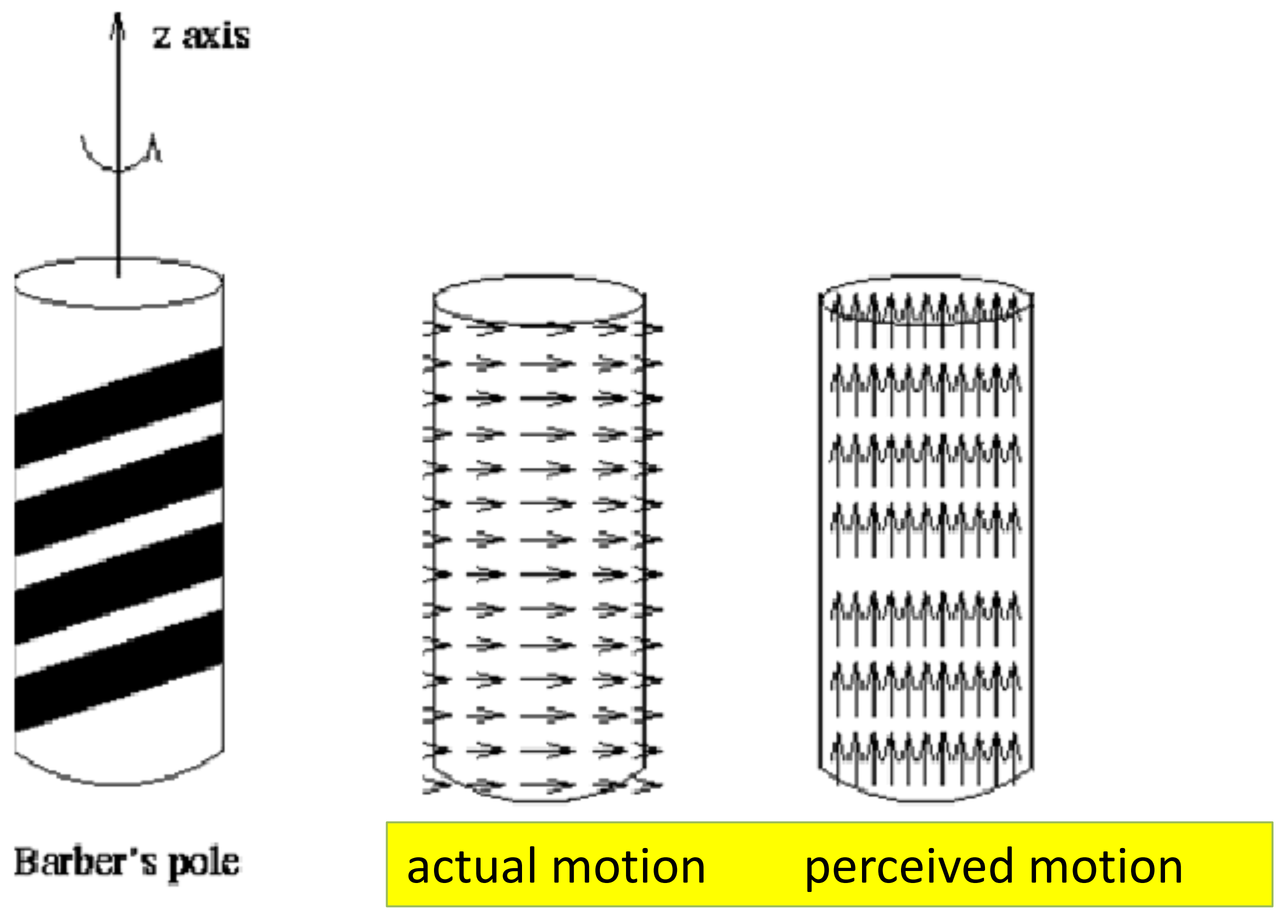
The barberpole illusion is a visual illusion that reveals biases in the processing of visual motion in the human brain. This visual illusion occurs when a diagonally striped pole is rotated around its vertical axis (horizontally), it appears as though the stripes are moving in the direction of its vertical axis (downwards in the case of the animation)

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

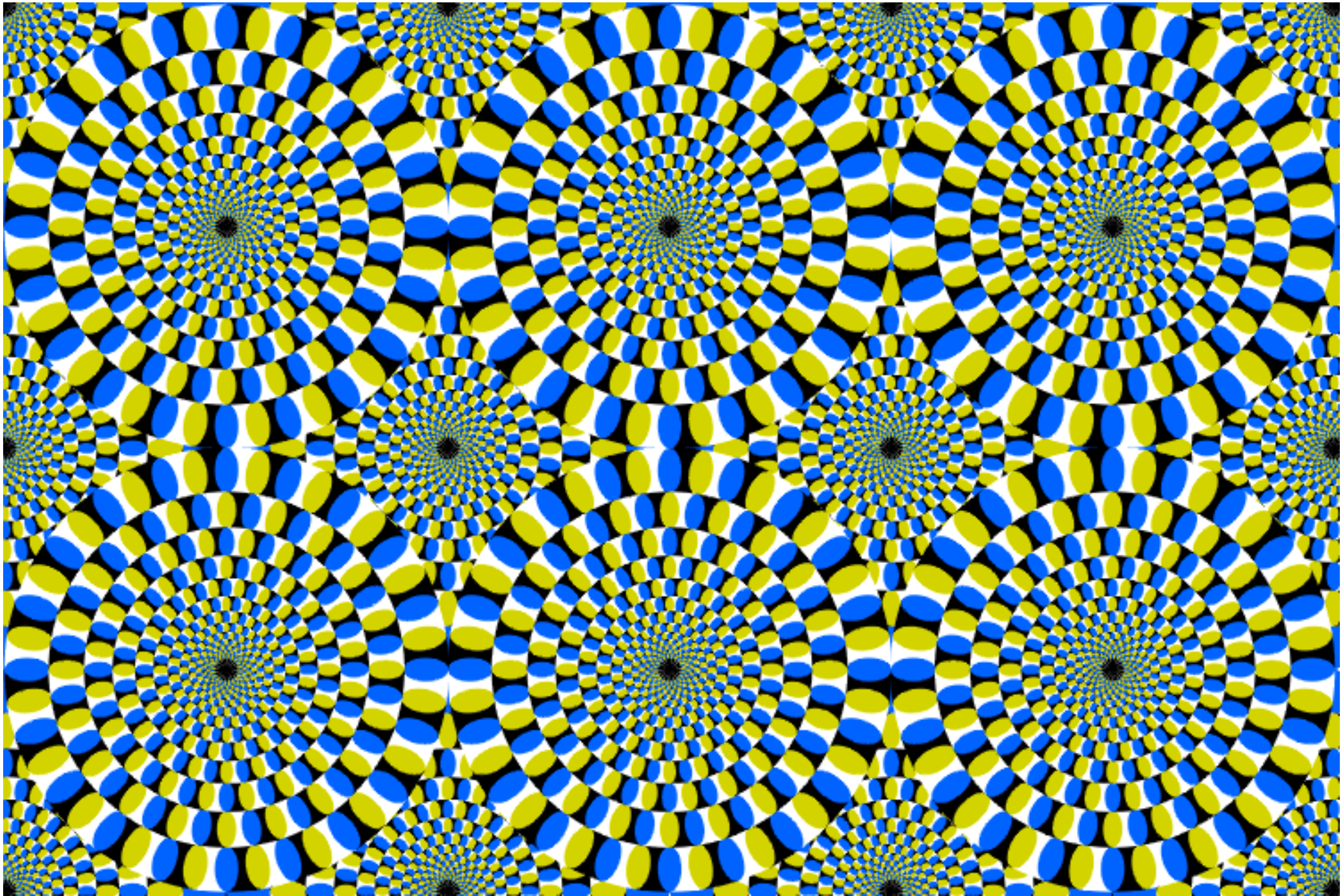
The barber pole illusion

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

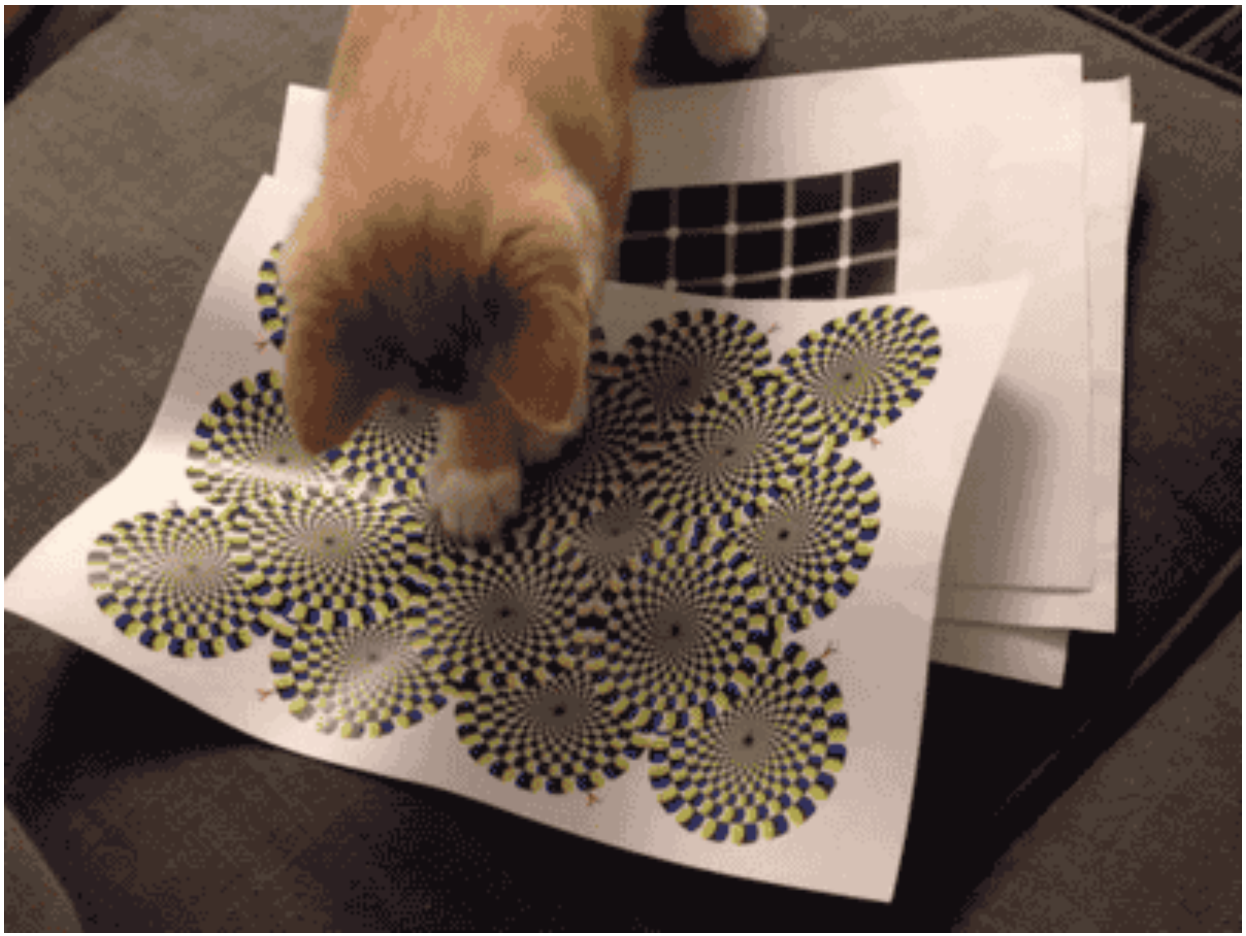
The barber pole illusion



Motion illusions



Does it work on other animals?



Solving the aperture problem

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

Lucas-Kanade flow

- Linear least squares problem:

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

- 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 $(\mathbf{A}^T\mathbf{A})\mathbf{d} = \mathbf{A}^T\mathbf{b}$

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

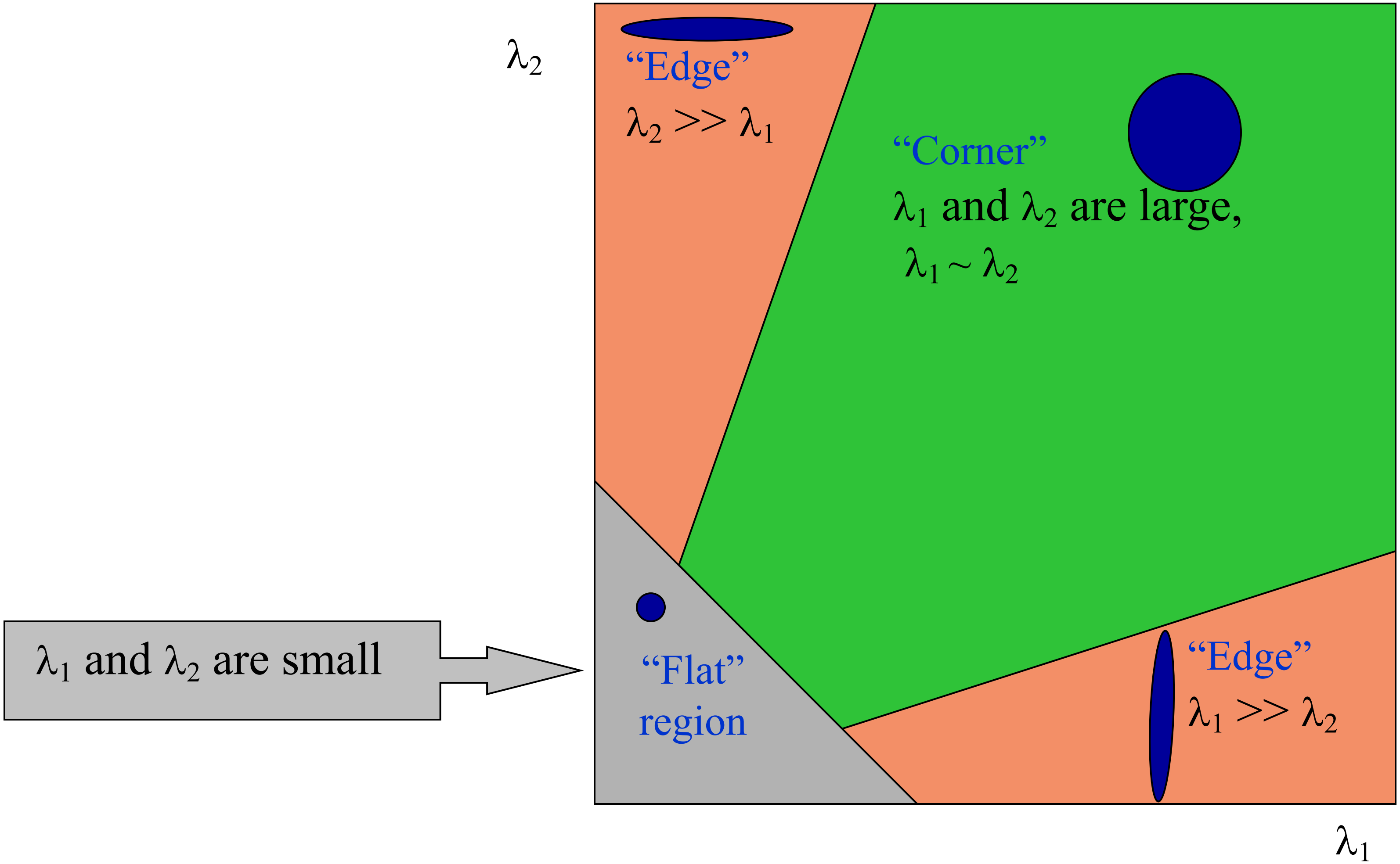
M = $A^T A$ is the second moment matrix!

(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.

Recall: second moment matrix

- Estimation of optical flow is well-conditioned precisely for regions with high "cornerness":



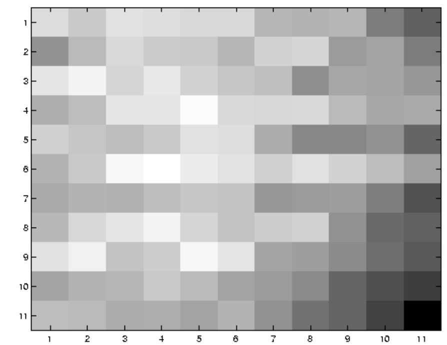
$M = A^T A$ is the *second moment matrix* !

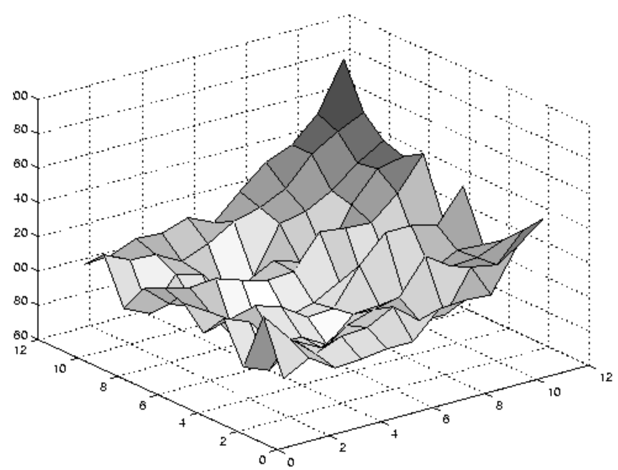
(Harris corner detector…)

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

- Eigenvectors and eigenvalues of $A^T A$ 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

Low-texture region



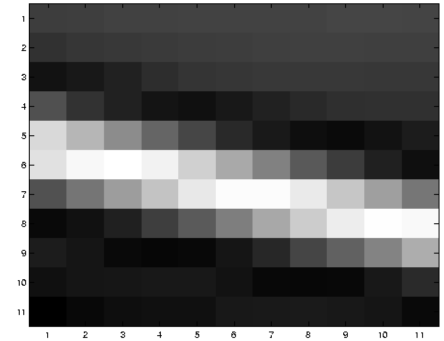


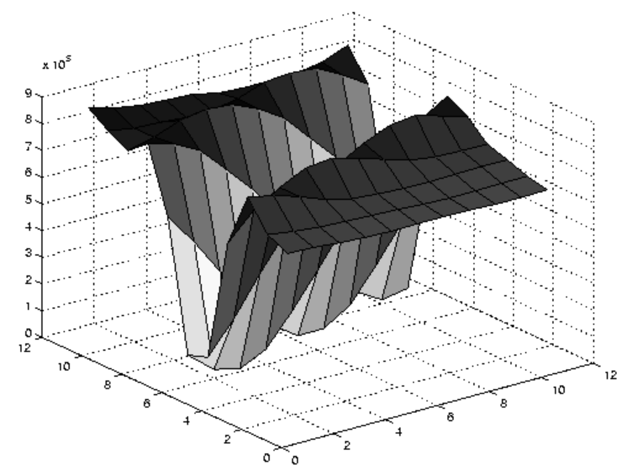
$$\sum \nabla I (\nabla I)^T$$

– gradients have small magnitude

– small λ_1, small λ_2

Edge



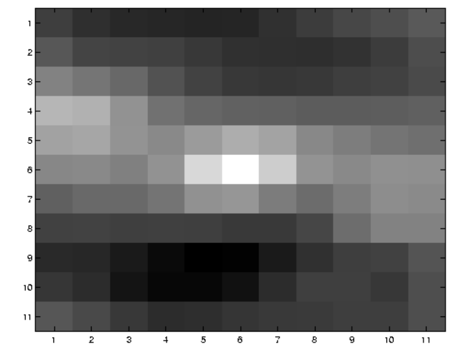


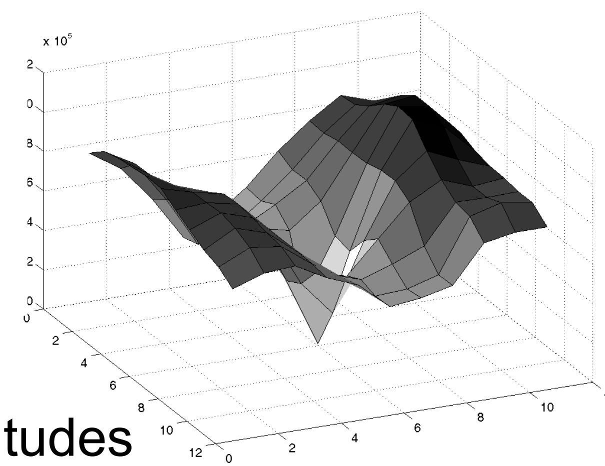
$$\sum \nabla I (\nabla I)^T$$

– large gradients, all the same

– large λ_1, small λ_2

High-textured region



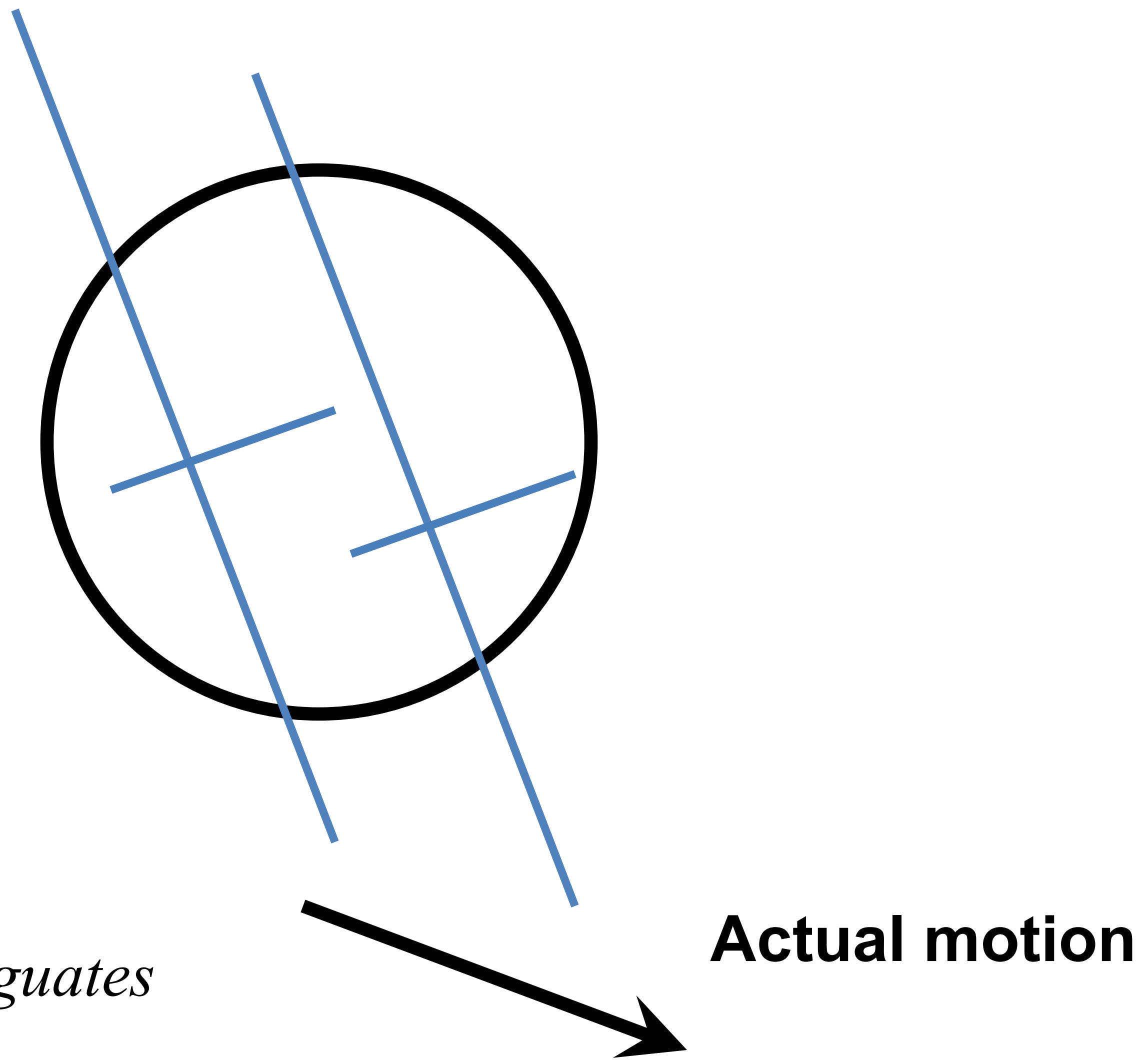


$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

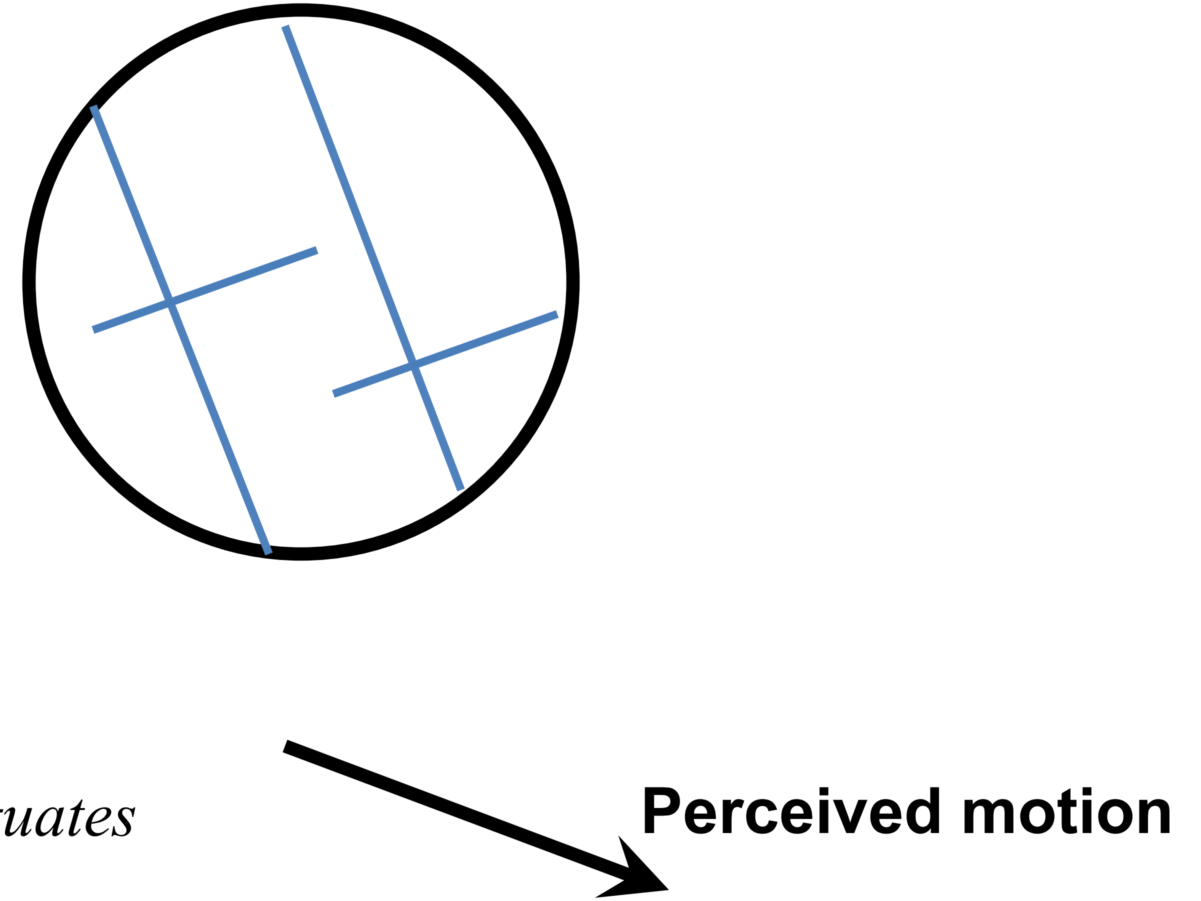
– large λ_1, large λ_2

The aperture problem resolved



Using corners disambiguates the aperture problem

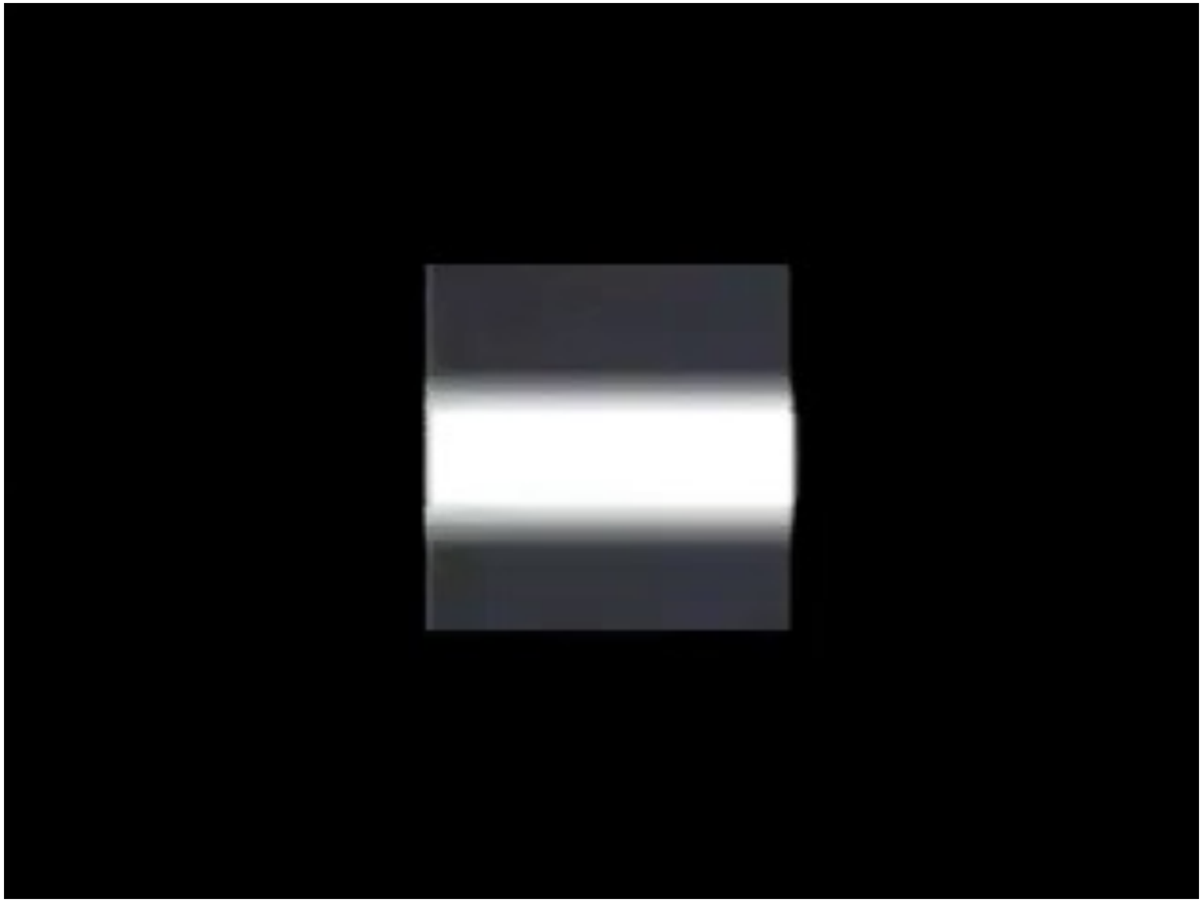
The aperture problem resolved



Using corners disambiguates the aperture problem

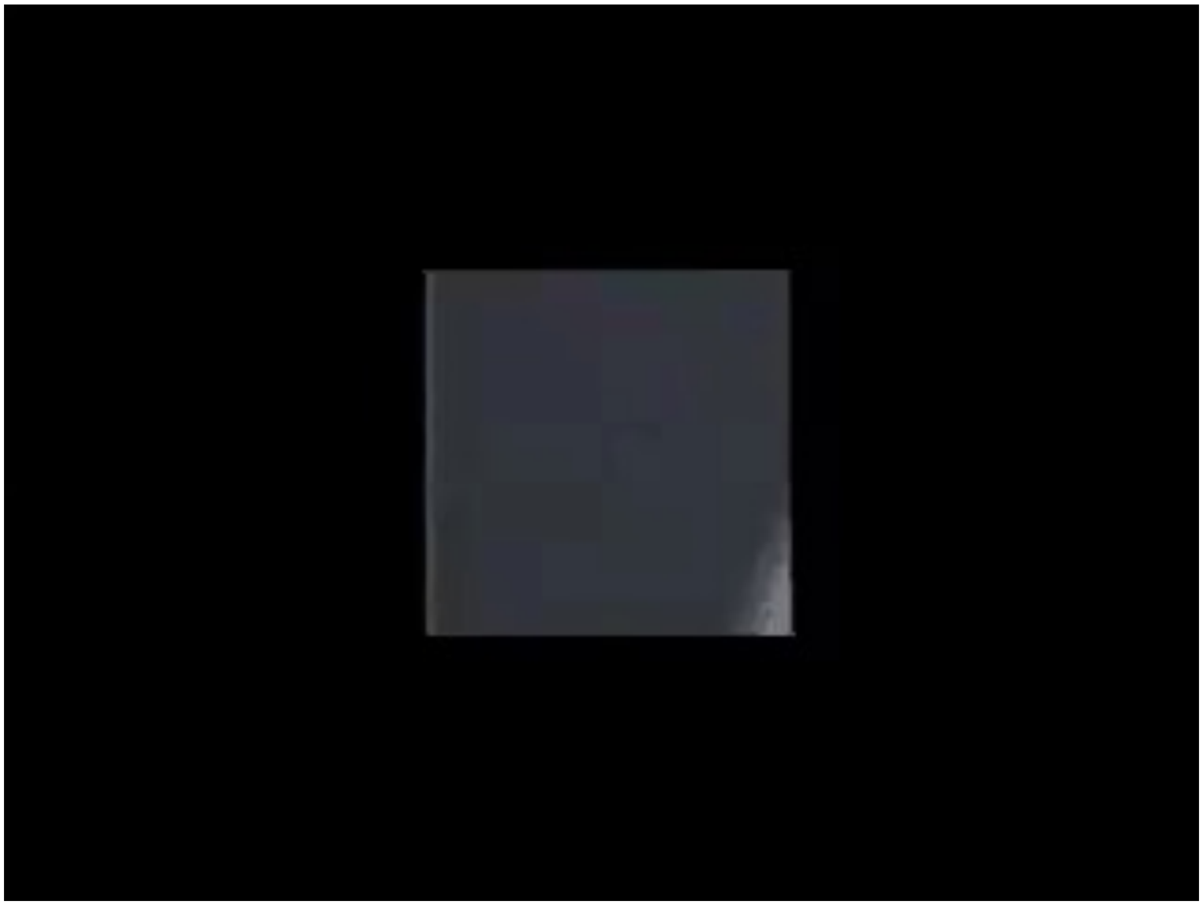
Conditions for solvability

- “Bad” case: single straight edge



Conditions for solvability

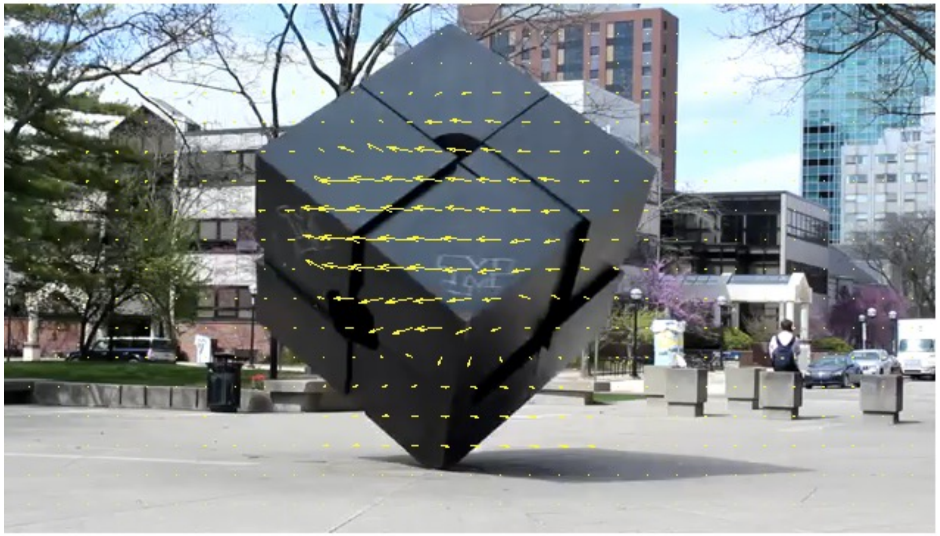
- "Good" case



Lucas-Kanade flow example

Input frames

Output



Source: [MATLAB Central File Exchange]

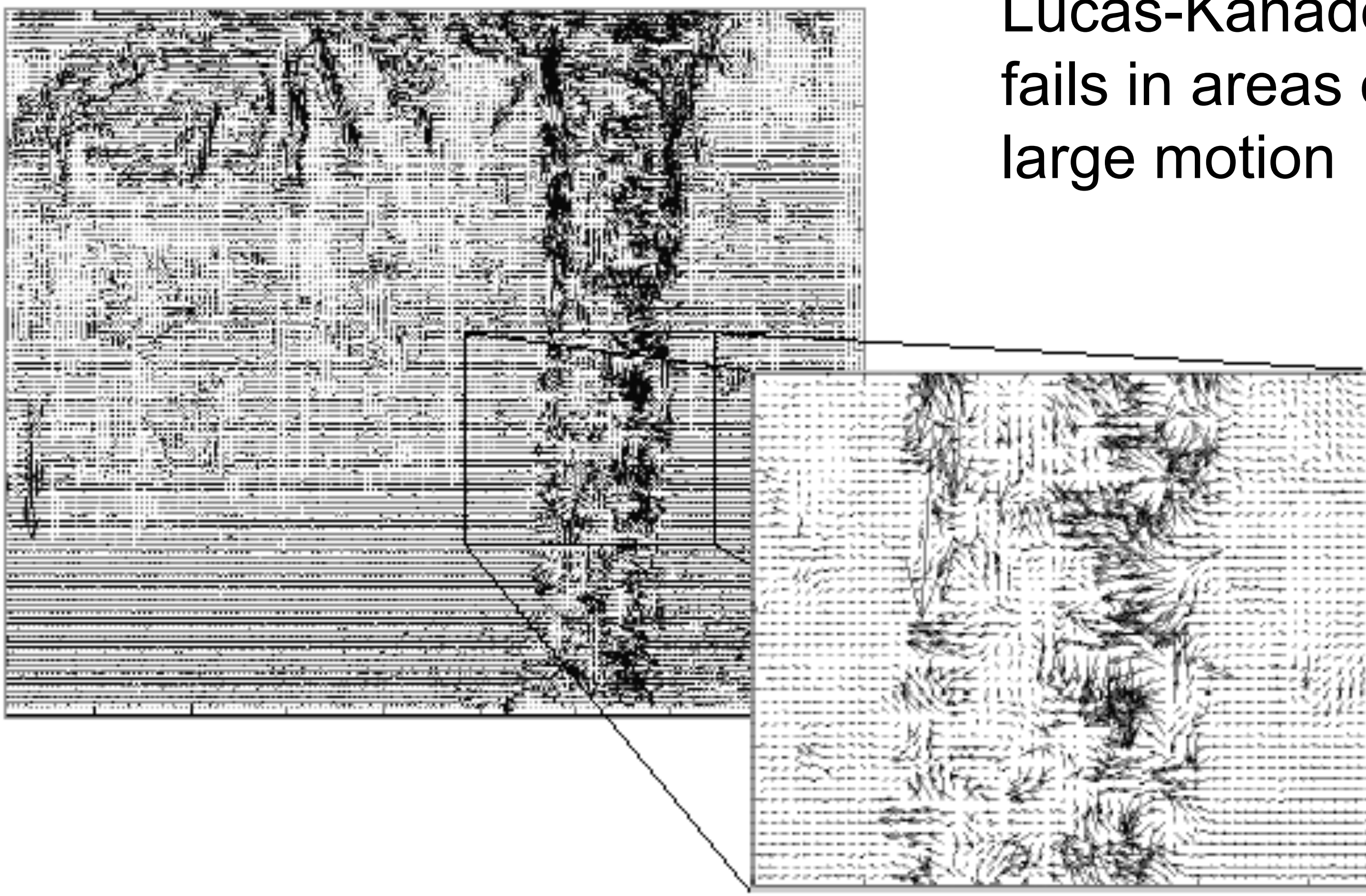
Errors in Lucas-Kanade

- The motion is large (larger than a pixel)
- A point does not move like its neighbors
- Brightness constancy does not hold

“Flower garden” example

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

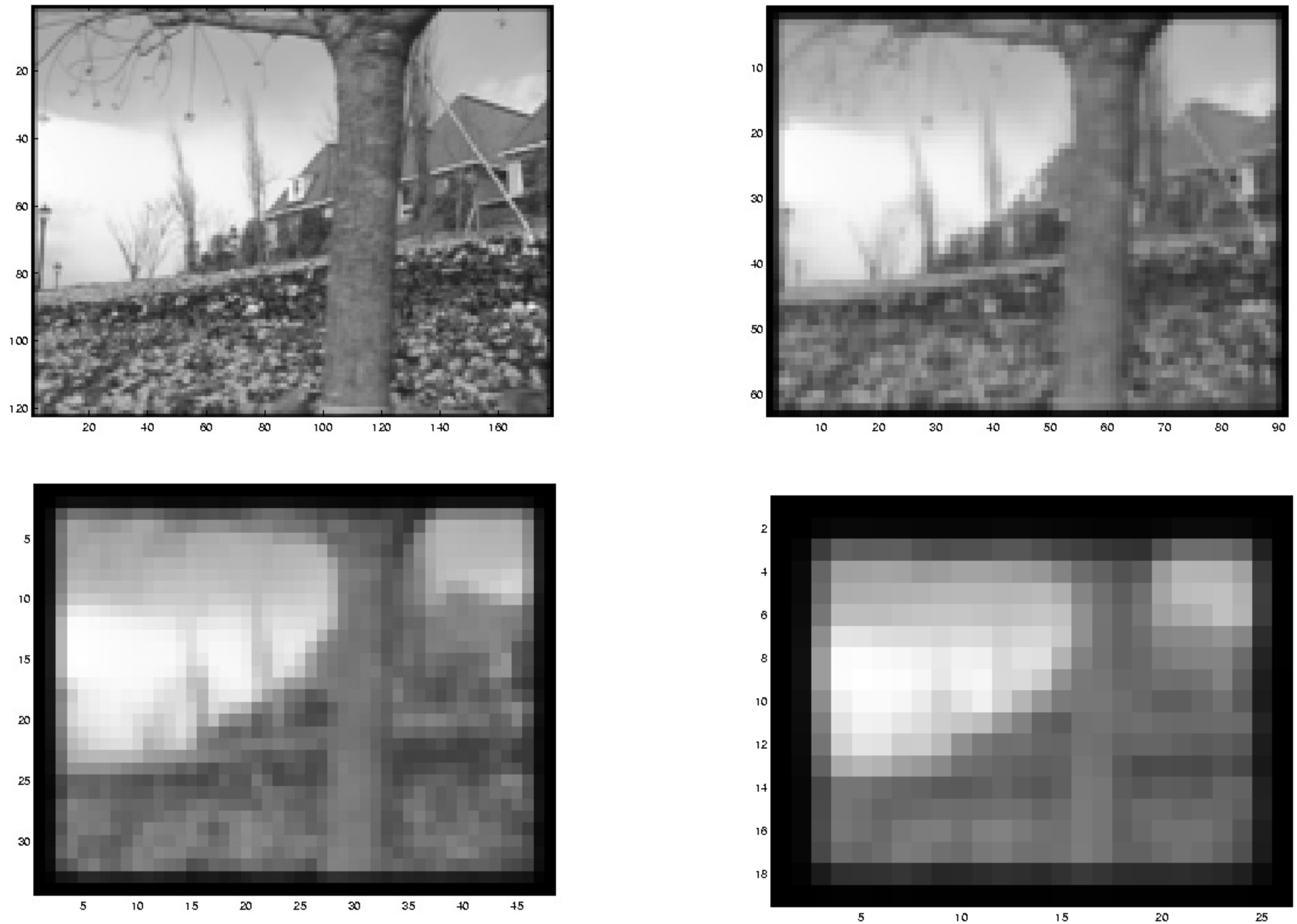
“Flower garden” example



Lucas-Kanade fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

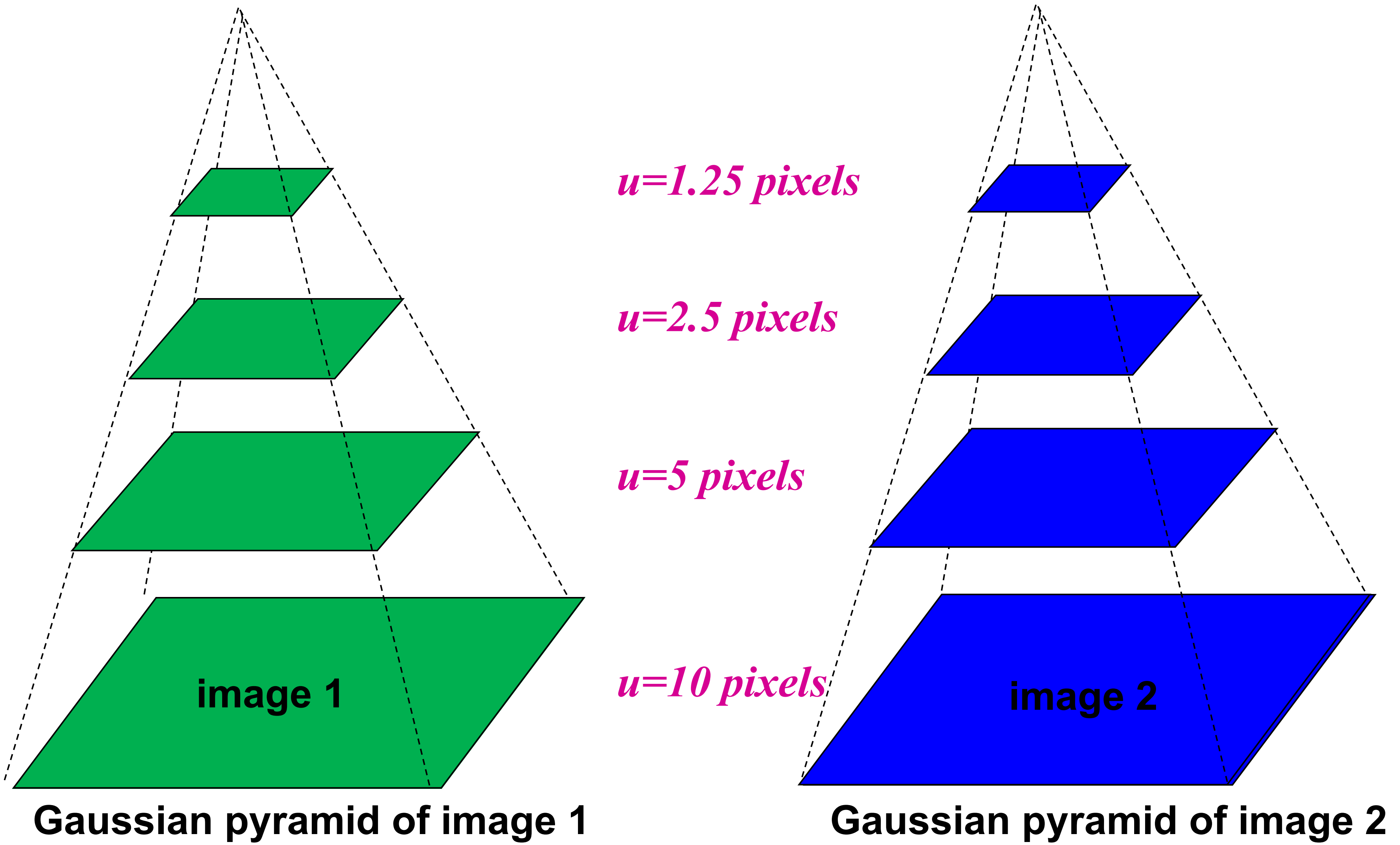
Reduce the resolution!



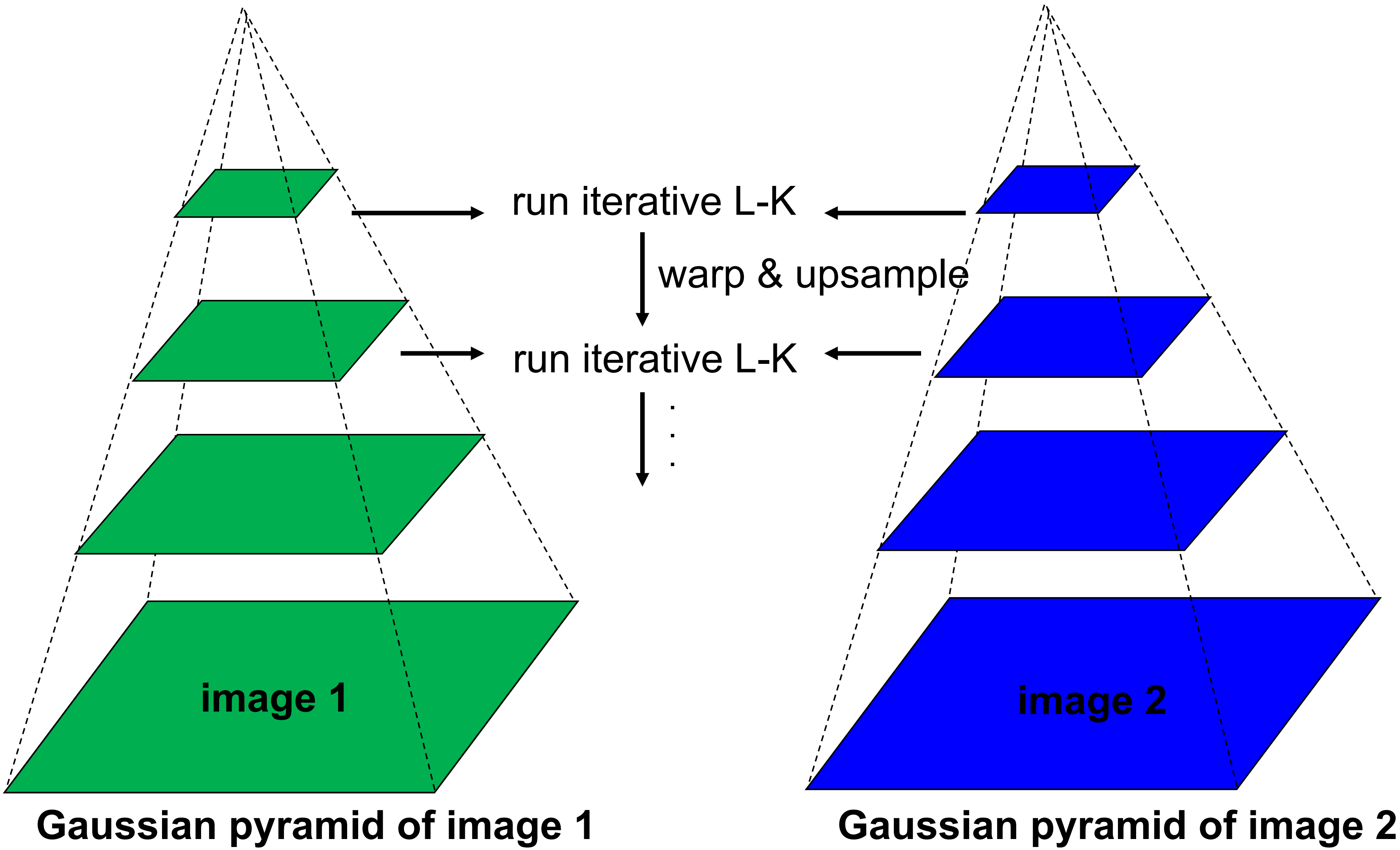
Multi-resolution Lucas Kanade Algorithm

- Compute ‘simple’ LK at highest level
- At level i
  - Take flow u_{i-1}, v_{i-1} from level i-1
  - bilinear interpolate it to create u_i^*, v_i^* matrices of twice resolution for level i
  - multiply u_i^*, v_i^* by 2
  - compute f_t from a block displaced by $u_i^*(x,y)$, $v_i^*(x,y)$
  - Apply LK to get $u_i'(x, y)$, $v_i'(x, y)$ (the correction in flow)
  - Add corrections u_i' v_i', *i.e.* $u_i = u_i^* + u_i'$, $v_i = v_i^* + v_i'$.

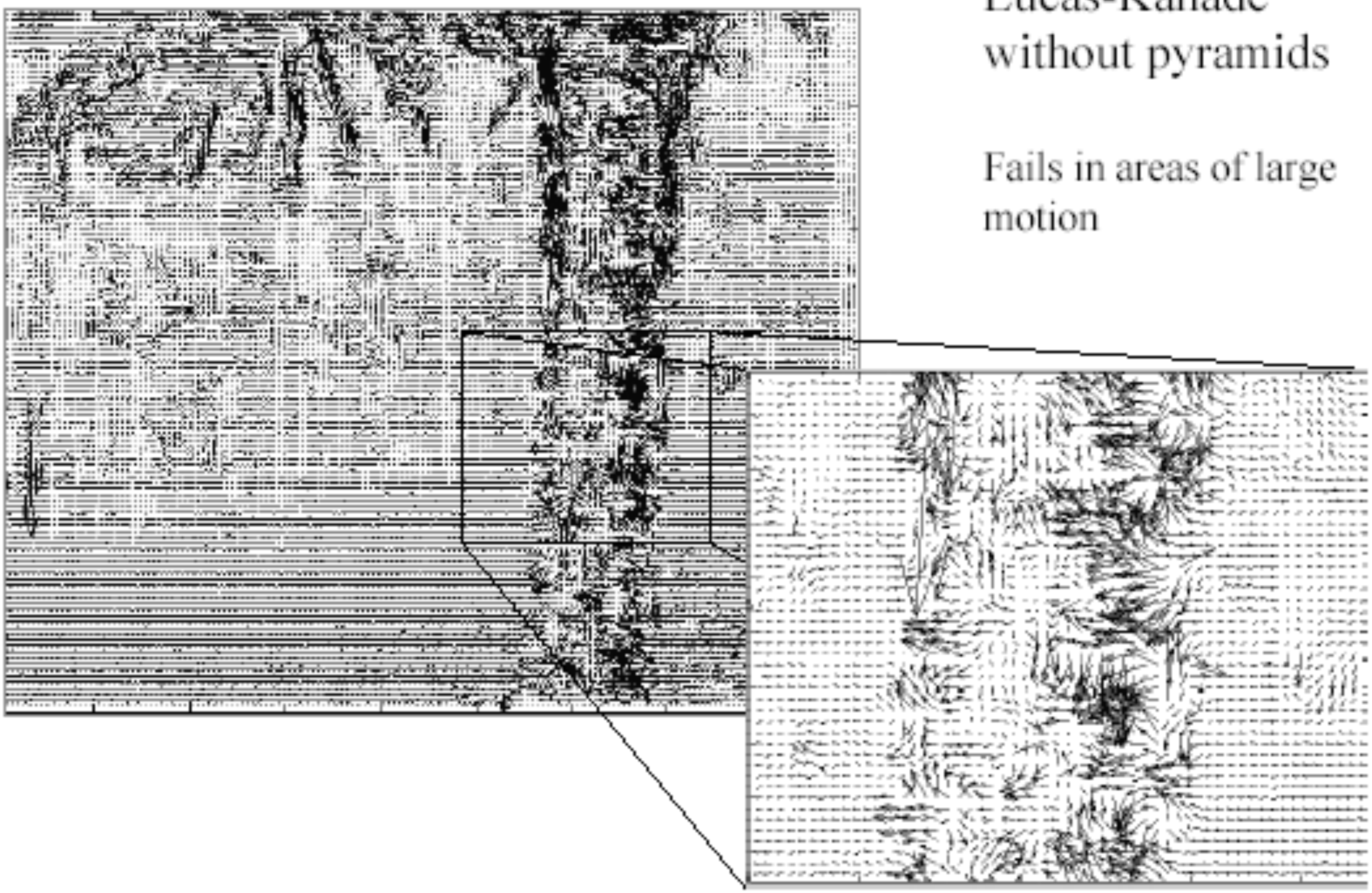
Coarse-to-fine optical flow estimation



Coarse-to-fine optical flow estimation



Optical Flow Results

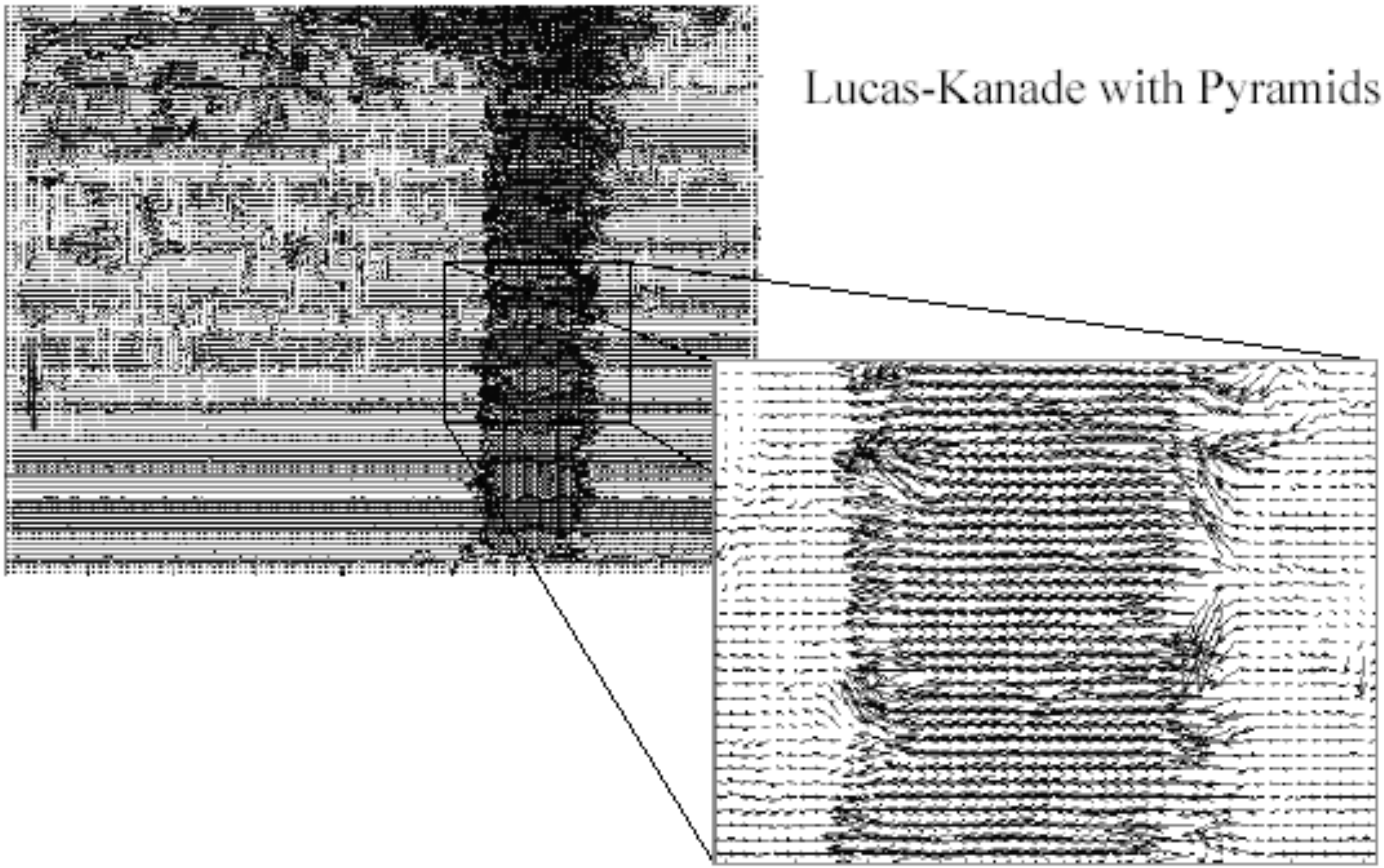


Lucas-Kanade without pyramids

Fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Optical Flow Results



* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Visualize the optical flow

Flow magnitude as saturation, orientation as hue

Visualize the optical flow

Flow magnitude as saturation, orientation as hue

Applications of optical flow

1. Traffic monitoring – finding velocities of vehicles

2. Slow motion effect

3. Image stabilization

4. Face tracking – lip reading, emotion detection