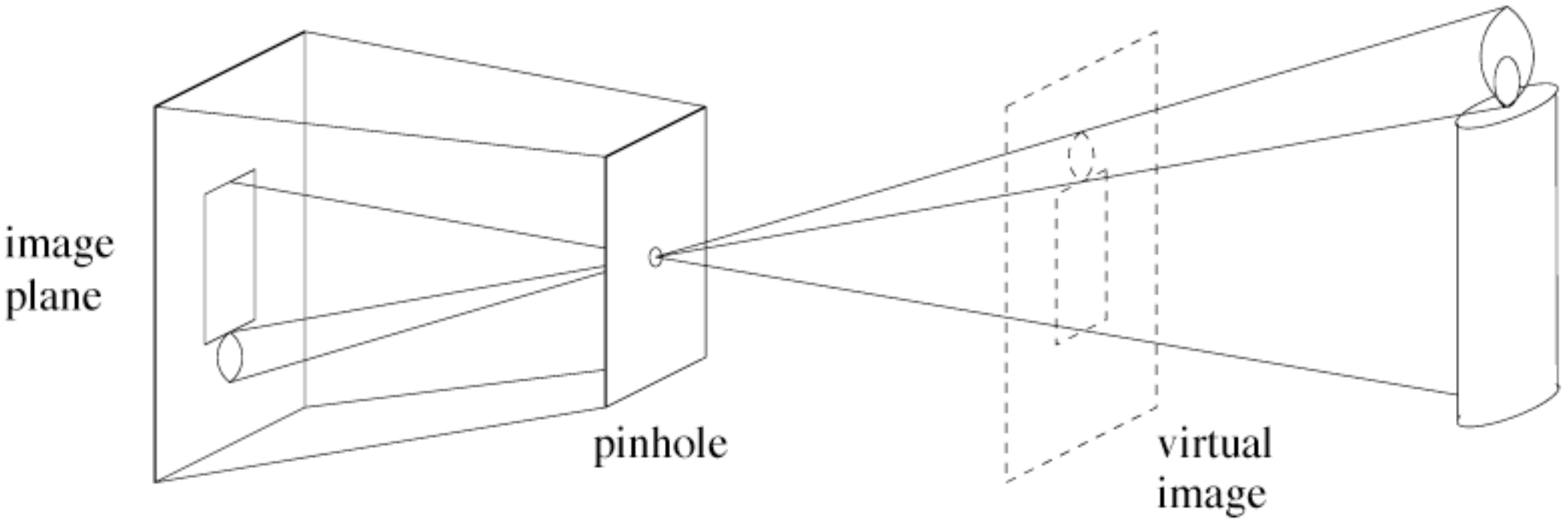
Two cameras

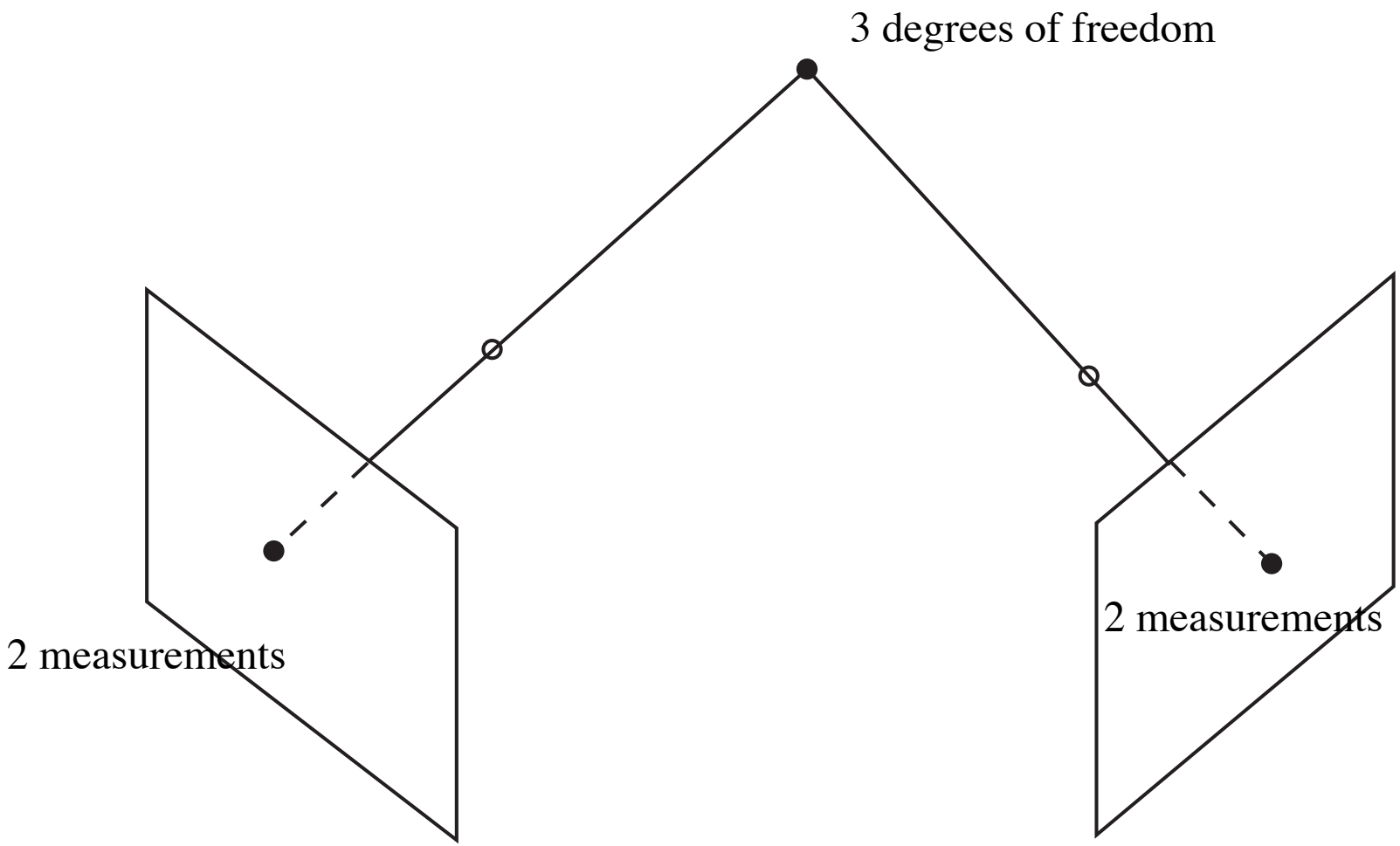
D.A. Forsyth, UIUC

How cameras work

Pinhole camera - an effective abstraction



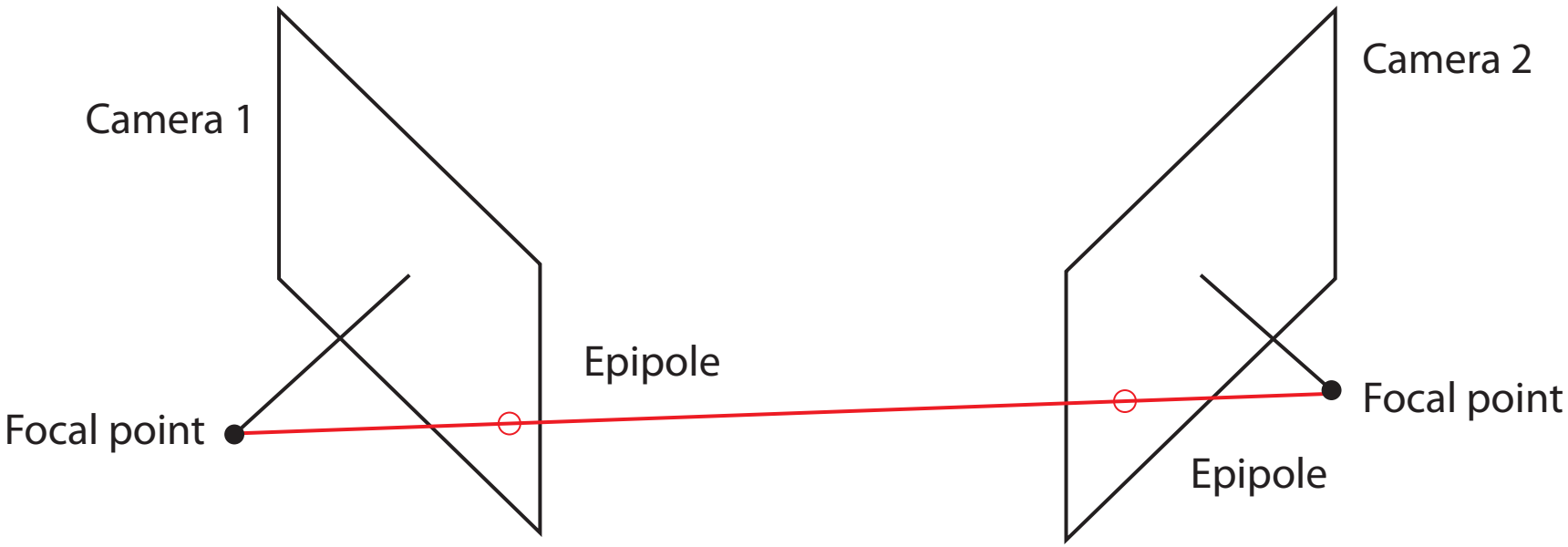
What happens in two views



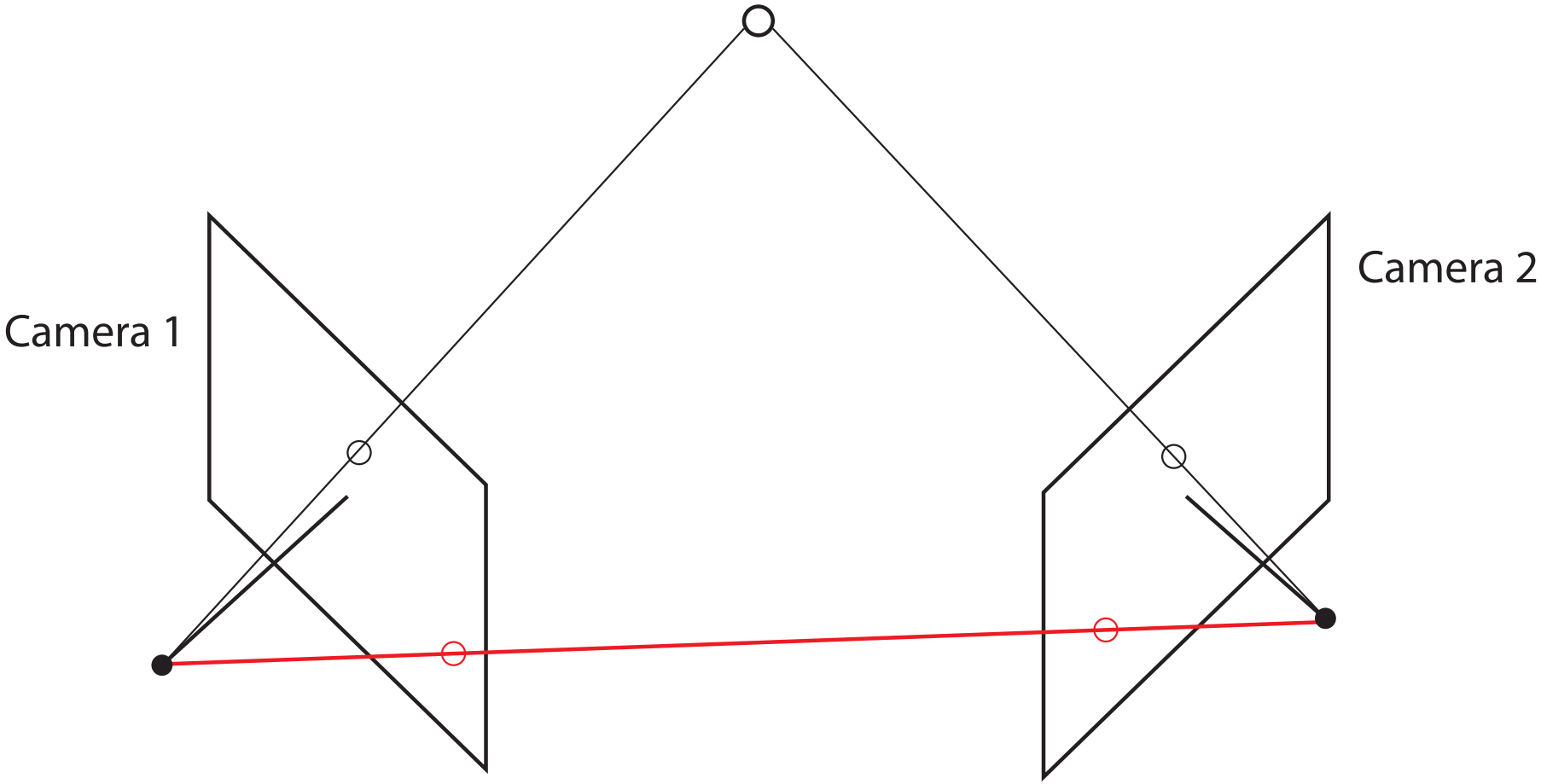
All of Camera Geometry

- From the picture
  - two views of a point give four measurements of three DOF
  - this means
    - correspondence is constrained
    - if we have enough points and enough pix we can recover
      - points
      - cameras

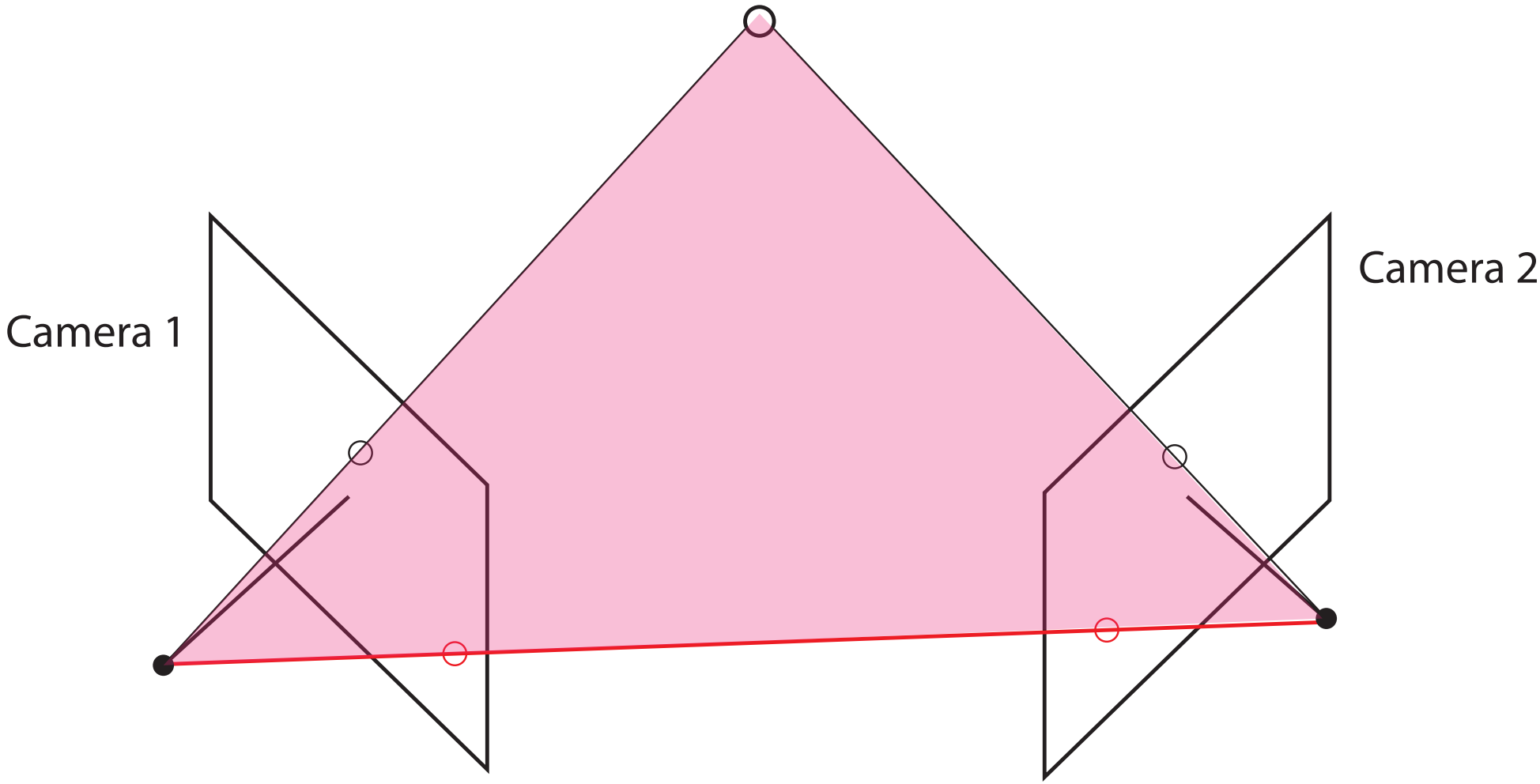
The Epipoles



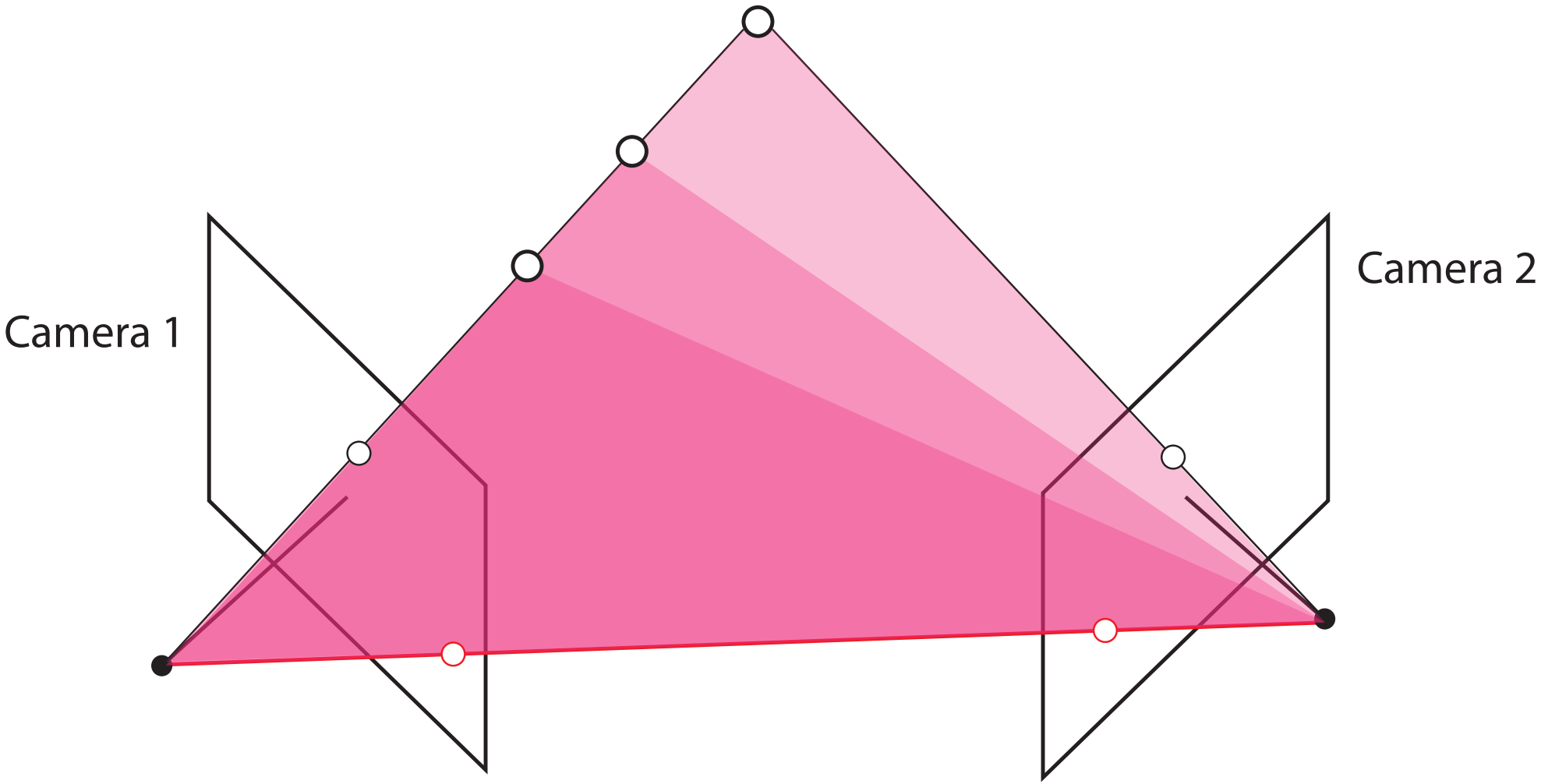
Constraints on correspondence



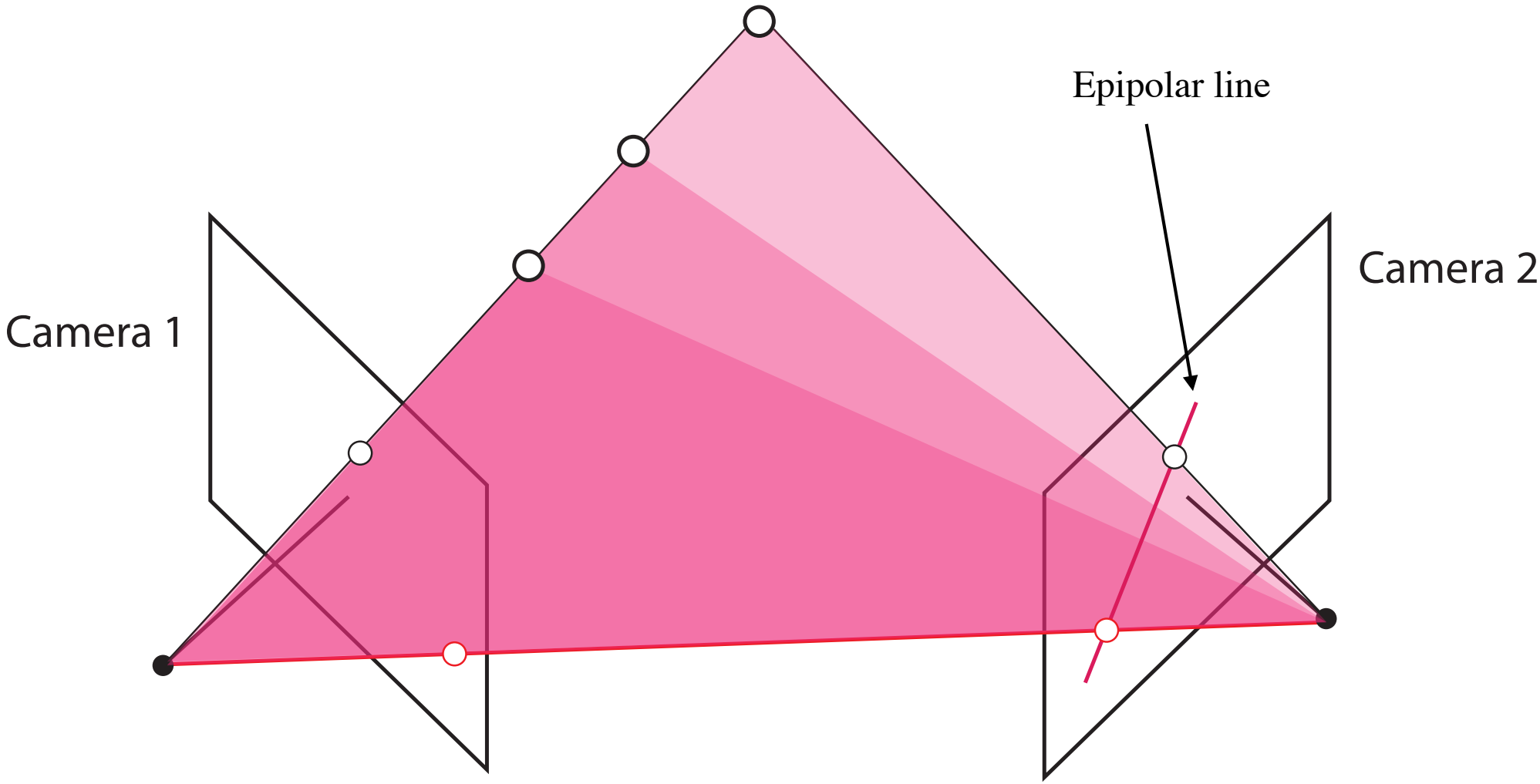
Constraints on CSP -II



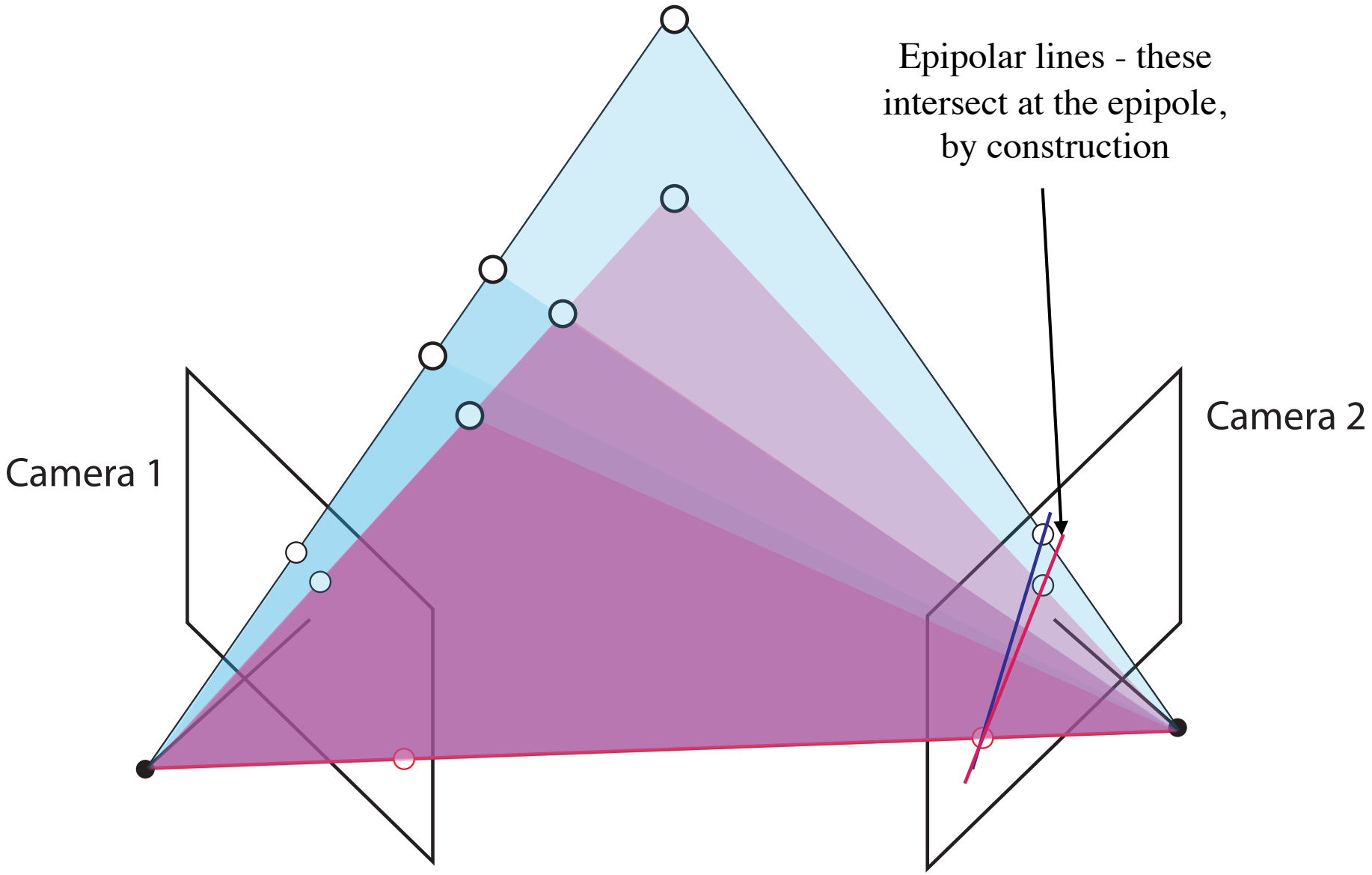
Constraints on CSP - III



Constraints on CSP - III

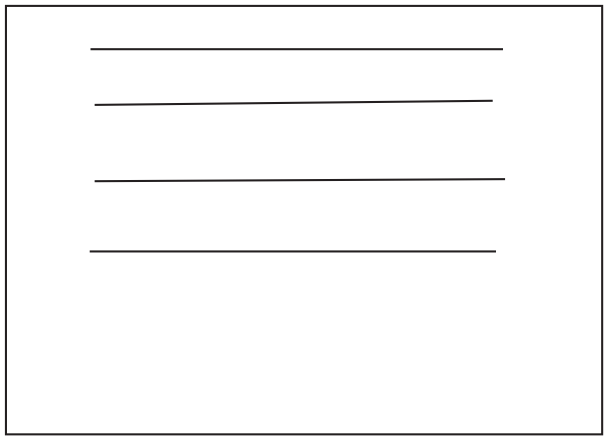


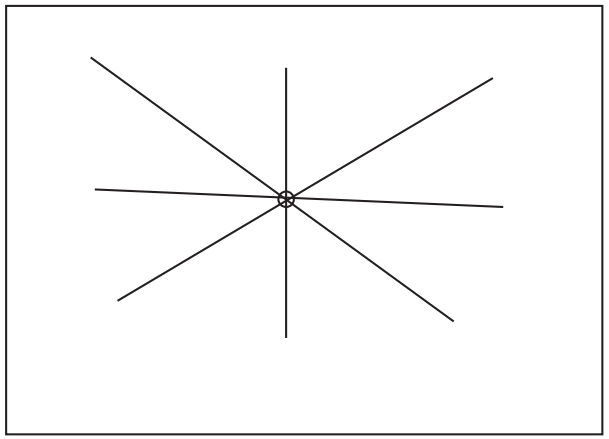
Constraints on CSP - IV



Epipoles (resp. epipolar lines)

- Informative





Epipole and epipolar lines in camera 1 - where is camera 2?

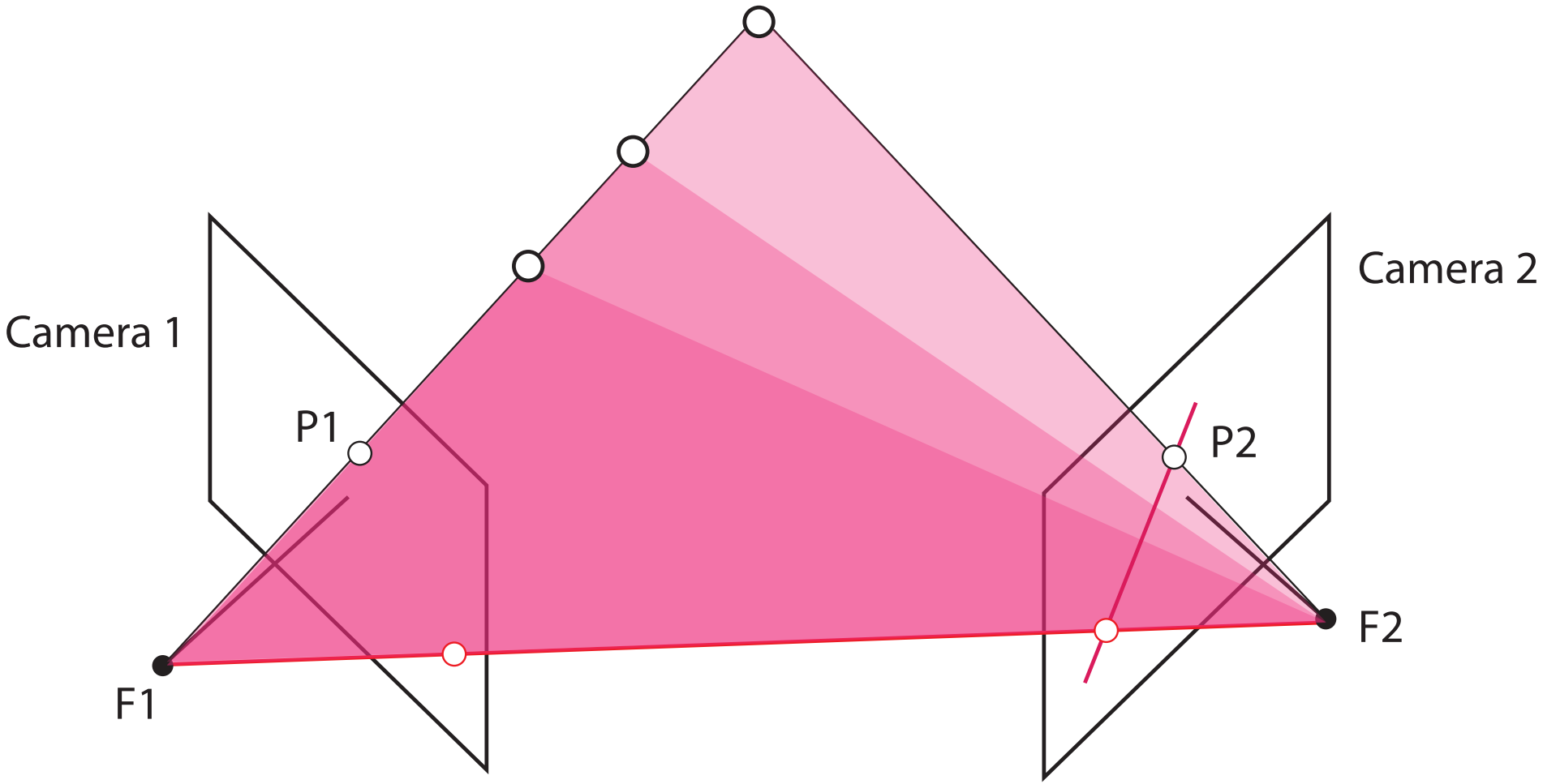
This means:

- A point in camera 1 identifies a line in camera 2
  - of all possible corresponding points in camera 2
- Equivalently, there is a map
  - from points in camera 1 (resp 2)
  - to lines in camera 2 (resp 1)
- Q: what is the form of the map?

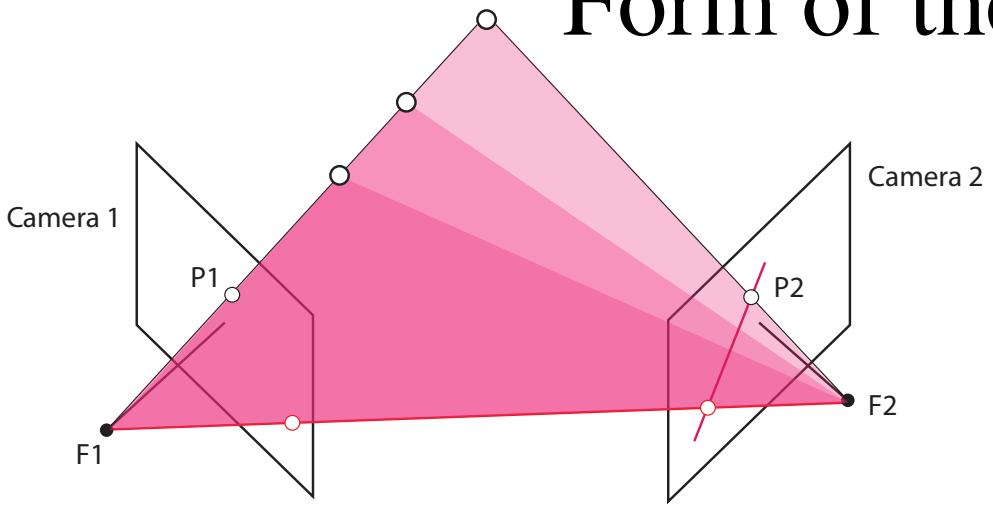
Planes in HCs

- Assume four points P1, P2, P3, P4 are coplanar
- Then
  - determinant([P1, P2, P3, P4])=0
- Trick:
  - equation of plane through three points?
  - determinant([P1, P2, P3, X])=0

Form of the map - notation



Form of the map - II



- 3D coordinates of P1 are linear in image coordinates (p1)
- 3D coordinates of P2 are linear in image coordinates (p2)
- so

$$\det\left([P_1, P_2, F_1, F_2]\right)$$

In HC's

  - linear in p1; linear in p2
- so there is some matrix F (function of cameras) so that

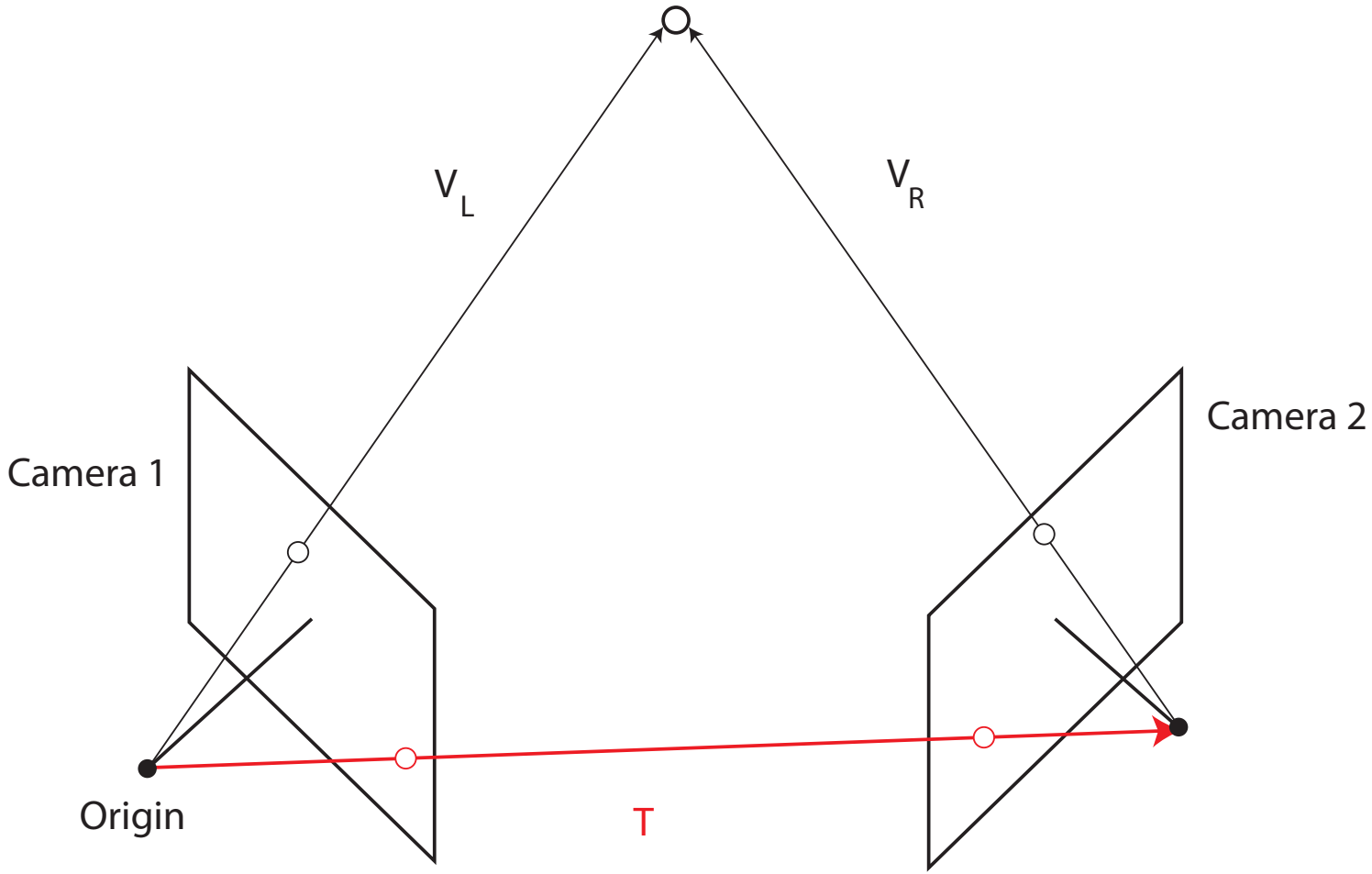
$$\mathbf{p_1}^T \mathcal{F} \mathbf{p_2} = 0$$

The Fundamental Matrix

$$\mathbf{p_1}^T \mathcal{F} \mathbf{p_2} = 0$$

- Easy closed form expression exists
  - in terms of rot, trans between cameras, intrinsics
  - following slides

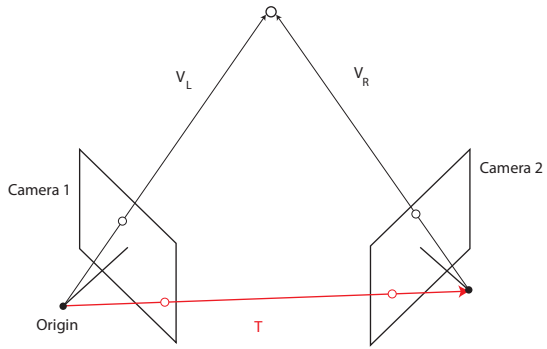
- Can be fit a pair of images using feature correspondences
  - 8 point algorithm
  - robustness is an important issue
  - we'll do this



Camera translation

$$\mathbf{V}_R = \mathcal{R}(\mathbf{V}_L - \mathbf{T})$$

Camera rotation



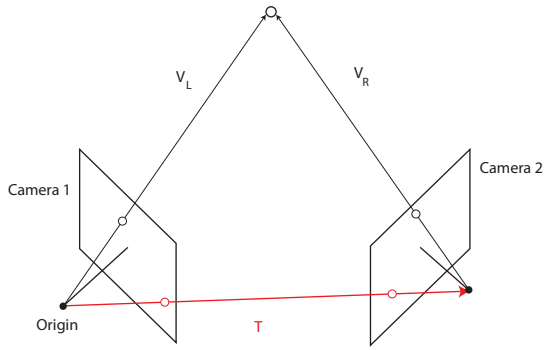
Camera translation

$$\mathbf{V}_R = \mathcal{R}(\mathbf{V}_L - \mathbf{T})$$

Camera rotation

$$\mathcal{S} = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix}$$

$$\mathbf{T}^T \mathcal{S} = \mathbf{0}$$



$$\mathbf{V}_R = \mathcal{R}(\mathbf{V}_L - \mathbf{T})$$

$$\mathbf{V}_R^T \mathcal{R}\mathcal{S}\mathbf{V}_L = (\mathbf{V}_L - \mathbf{T})^T \mathcal{R}^T \mathcal{R}\mathcal{S}\mathbf{V}_L = \mathbf{V}_L^T \mathcal{S}\mathbf{V}_L = 0$$

RECALL: The camera matrix - II

- Turn previous expression into HC's
  - HC's for 3D point are (X,Y,Z,T)
  - HC's for point in image are (U,V,W)

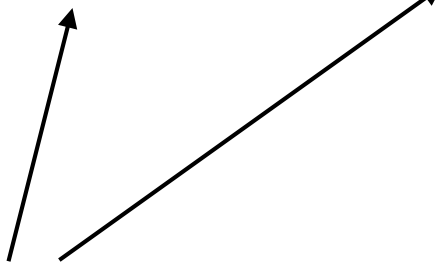
Transforms points from object coordinates into world coordinates most likely a rotation and translation

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \mathcal{C} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathcal{W} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Transforms camera coordinates (f is hidden in here)

So…

$$\mathcal{F} = k\mathcal{C}_L^{-T}\mathcal{R}\mathcal{S}\mathcal{C}_R^{-1}$$



If we know these

we can recover info about R, T from F

Fundamental matrix and epipolar lines

- In homogenous coordinates, line in plane is:

$$aX + bY + cZ = 0$$

- can write:

$$\mathbf{a}^T\mathbf{x} = 0$$

- But look at

$$\mathbf{p_1}^T\mathcal{F}\mathbf{p_2} = 0$$

  - which can be written

Coefficients of a line in image 2 created by F and p_1

$$\mathbf{p}_1^T\mathcal{F}\mathbf{p}_2 = (\mathcal{F}^T\mathbf{p}_1)^T\mathbf{p}_2 = 0$$

The Fundamental Matrix

$$\mathbf{p_1}^T \mathcal{F} \mathbf{p_2} = 0$$

- A map from
  - point in 1 (resp. 2) to line in 2 (resp. 1)
- This is the algebraic version of the picture
  - but the picture tells us more
- Any point in 1 maps to a line through the epipole
  - MOST lines in 2 are NOT in the image of the map
  - only a 1 parameter family of lines IS (the ones through the epipole)
- F has rank 2!
- Left (resp. right) kernel of F is left (resp. right) epipole

The 8 point algorithm

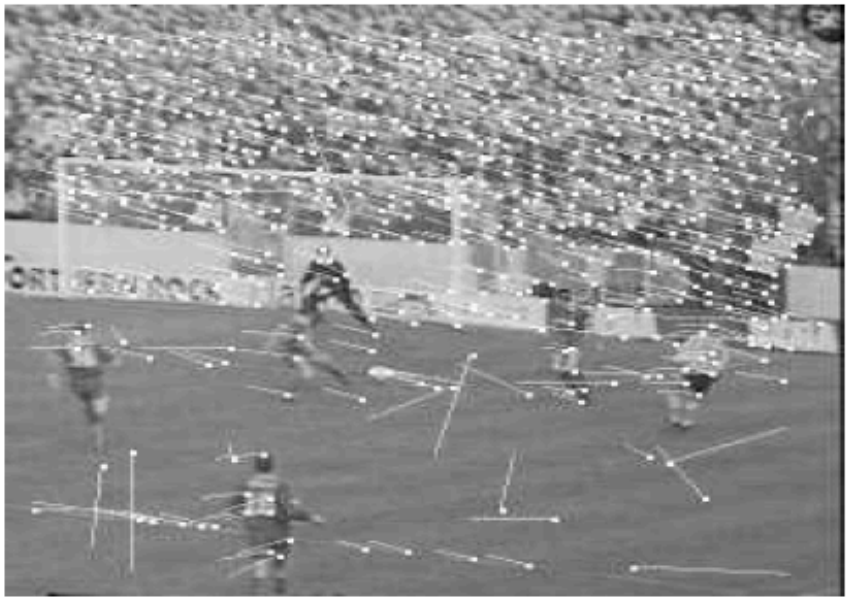
$$\mathbf{p_1}^T \mathcal{F} \mathbf{p_2} = 0$$

- Find 8 p1, p2 pairs
  - this gives 8 homogeneous linear equations in F coefficients
  - solve these
- Improvements
  - you can do it with seven points and solving a cubic (rank deficient)
  - the image coordinate system really matters for the quality of estimate
  - this requires robust estimation to work well
    - RANSAC

RANSAC (outline)

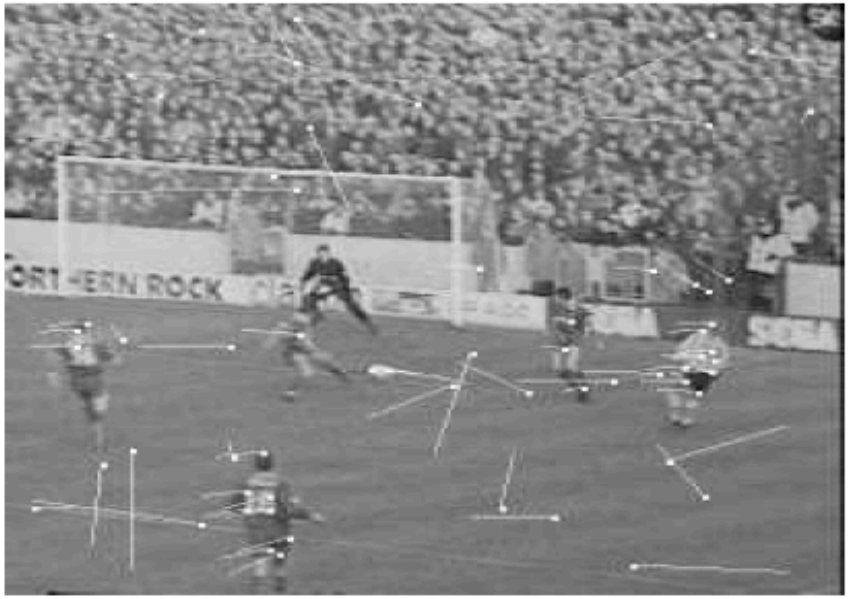
- Repeat many times
  - Find 8 pairs (p1, p2)
  - Fit F using 8 point
  - record the number of inlying pairs
    - pairs p1, p2 where:
      - p2 is “close” to (F p1)^T
      - p1 is “close” to (F p2)^T
      - there’s an appearance match
- Take F with most inlying pairs
  - fit to all inliers
    - using perpendicular distance from point to line
- Q: repeat how many times?
  - A: often enough that you have high prob of seeing 8 inlying pairs

(a)



(b)

(c)



(d)

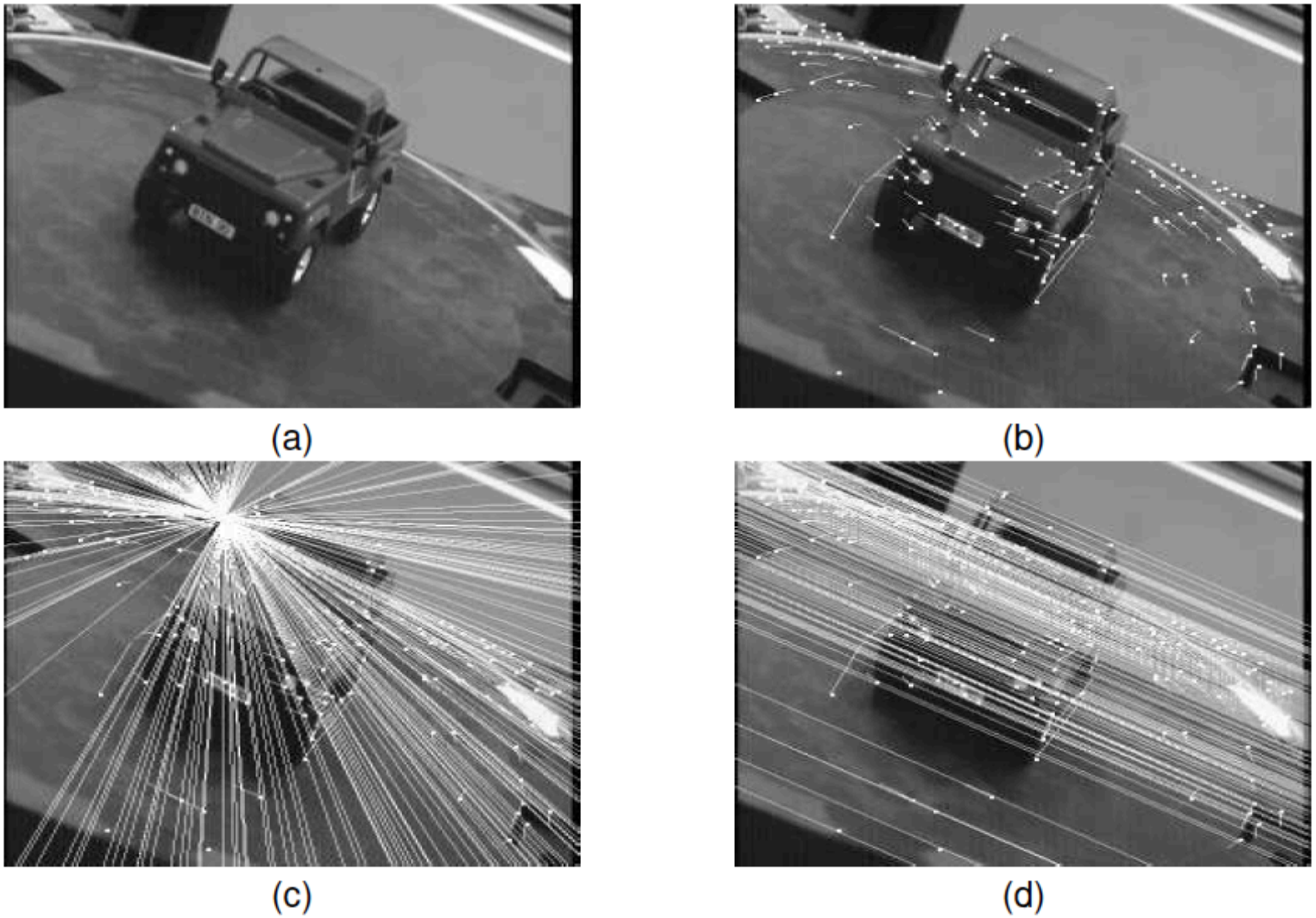


Fig. 18. In (a) (b) two consecutive images of a buggy rotating on a turntable. (b) has 167 matches superimposed on the second image. (c) (d) show two epipolar geometries generated by two distinct fundamental matrices, 139 correspondences are consistent with the fundamental matrix in (a), 131 are consistent with the fundamental matrix in (b) yet the two epipolar geometries obviously differ.

Stereopsis

- Generically:
  - recover depth map from two images of scene
    - cameras may be calibrated/uncalibrated
      - may have large/small baseline
      - if uncalibrated, recover from fundamental matrix, above
  - do so by
    - finding correspondences
    - constructing depth map using correspondences
- Huge literature, with multiple important tricks, etc.
  - I'll mention a small set

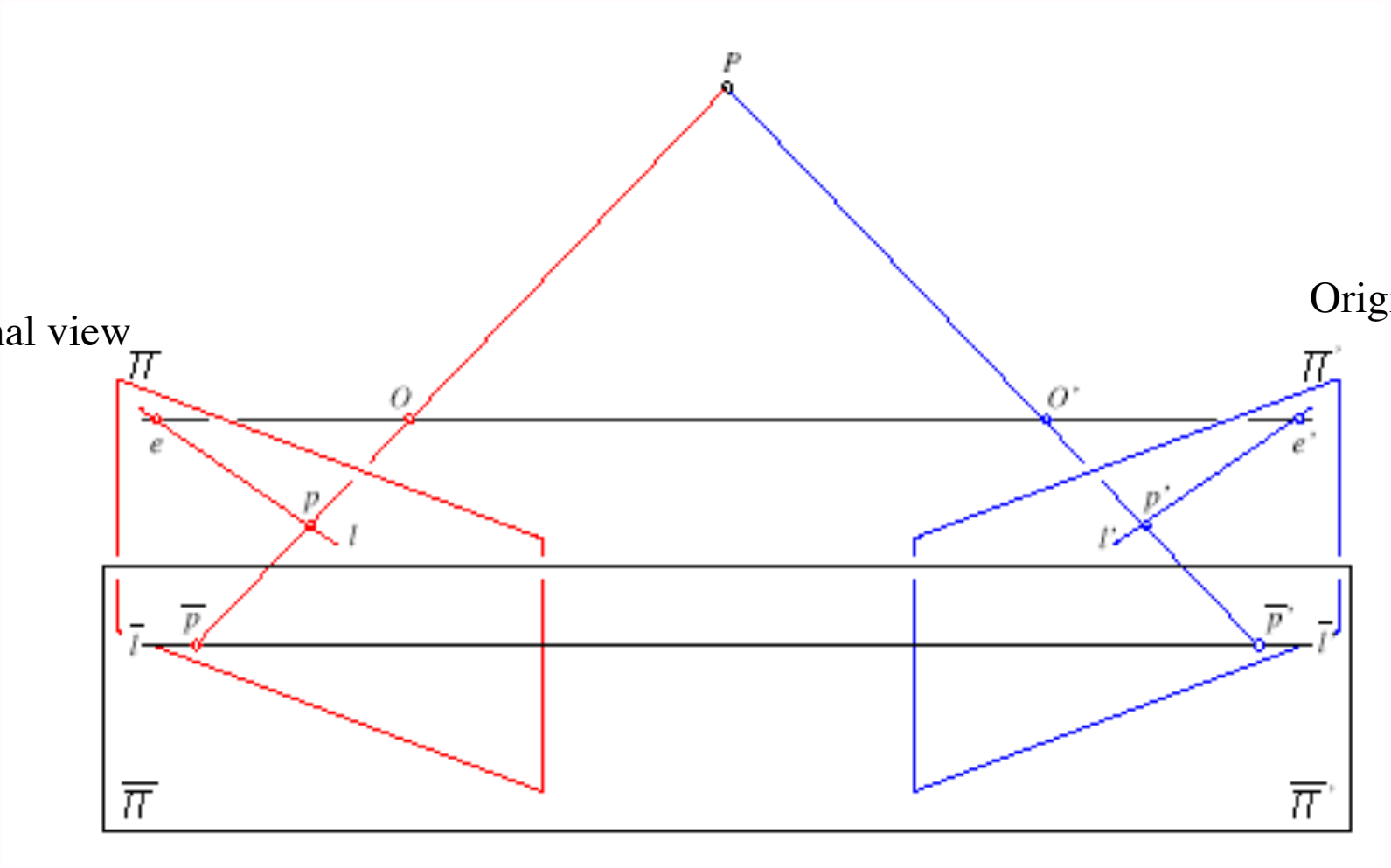
Pragmatics

- Simplify activities by rectifying to ensure
  - That camera image planes are coplanar
  - That focal lengths are the same
  - That the separation is parallel to the scanlines
  - (all this used to be called the epipolar configuration)

Rectification

Original view

Original view



Rectified views

Triangulation

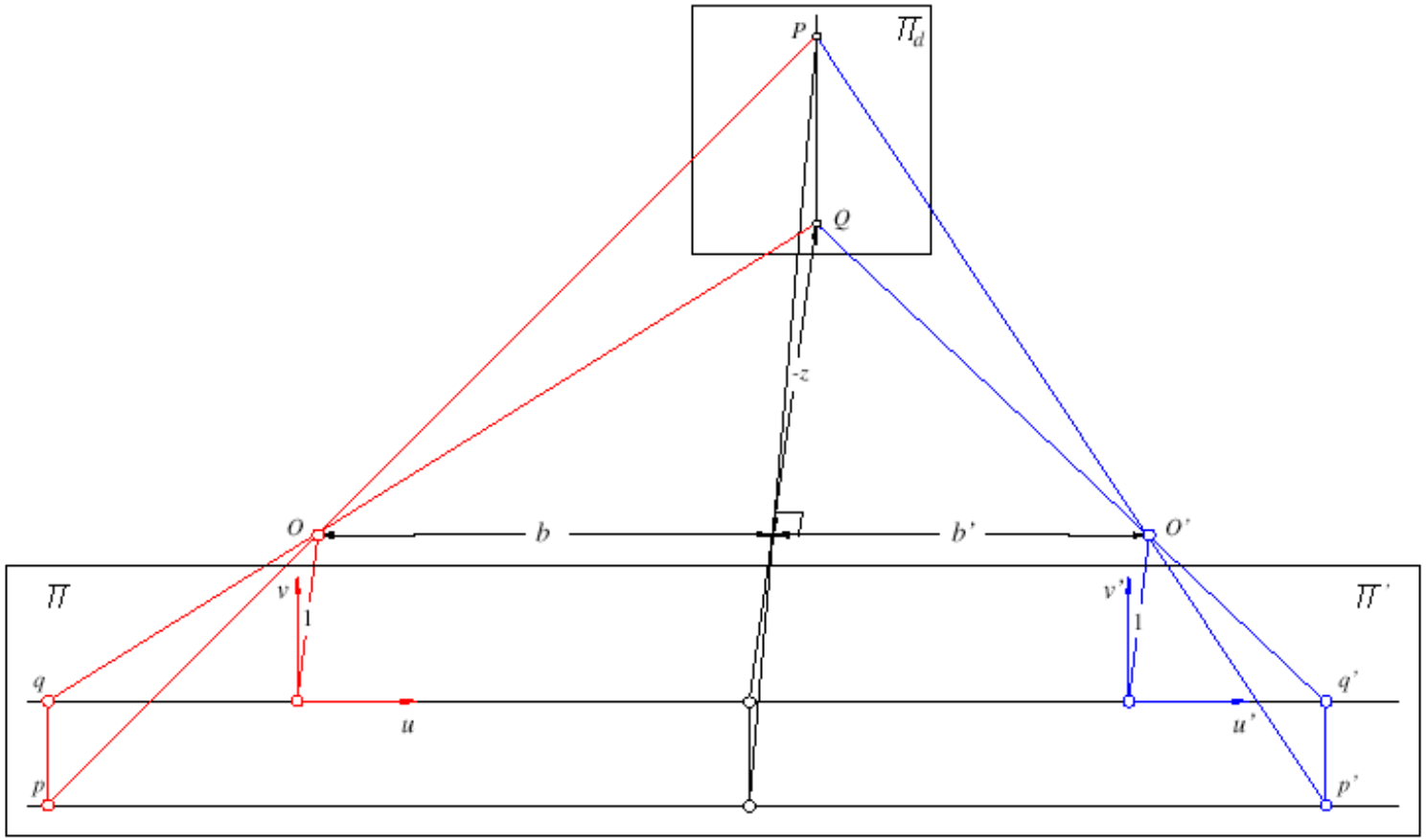
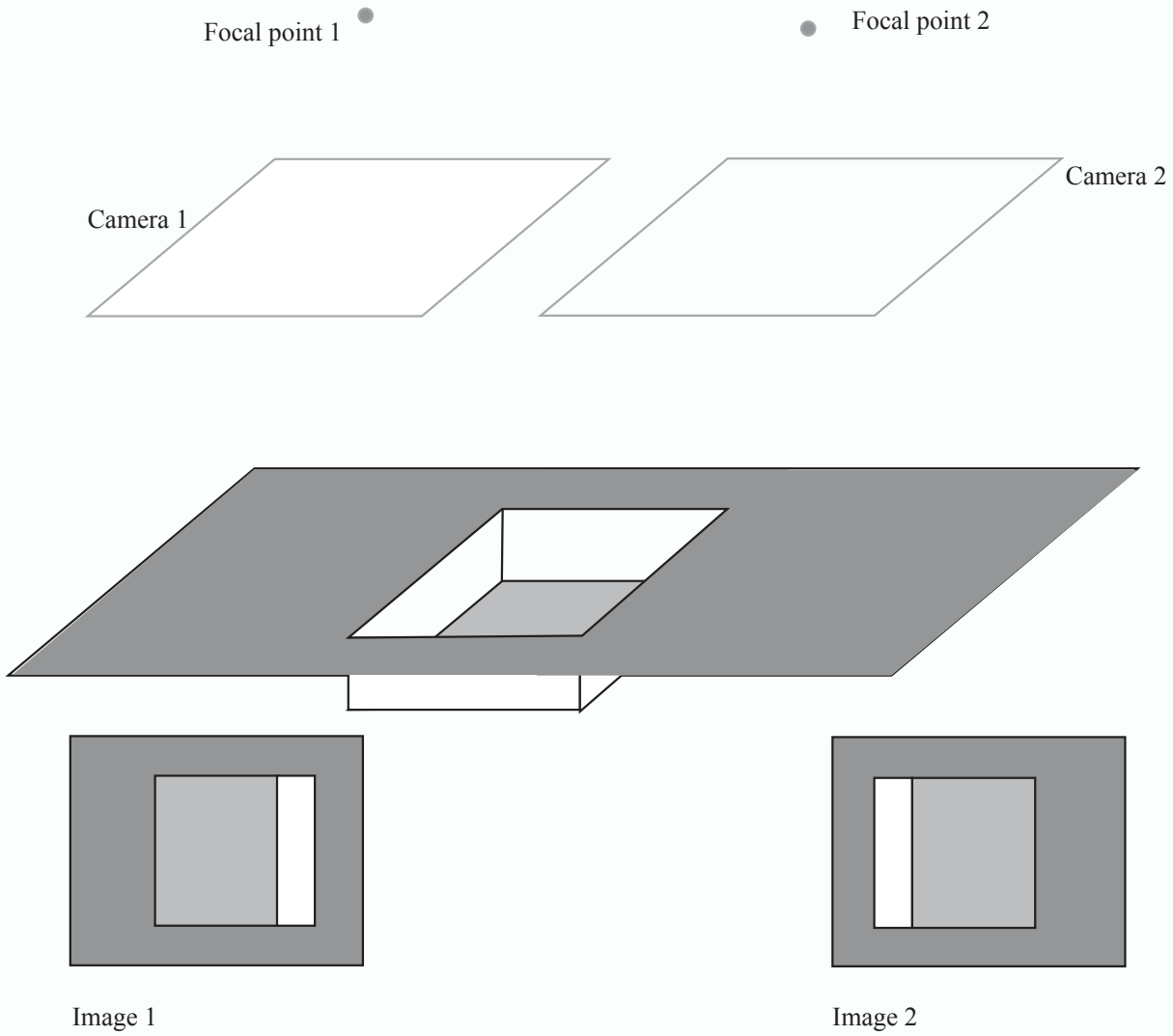


Figure 13.6. Triangulation for rectified images: the rays associated with two points p and p' on the same scanline are by construction guaranteed to intersect in some point P. As shown in the text, the depth of P relative to the coordinate system attached to the left camera is inversely proportional to the disparity $d = u' - u$. In particular, the preimage of all pairs of image points with constant disparity d is a *frontoparallel* plane Π_d (i.e., a plane parallel to the camera retinas).

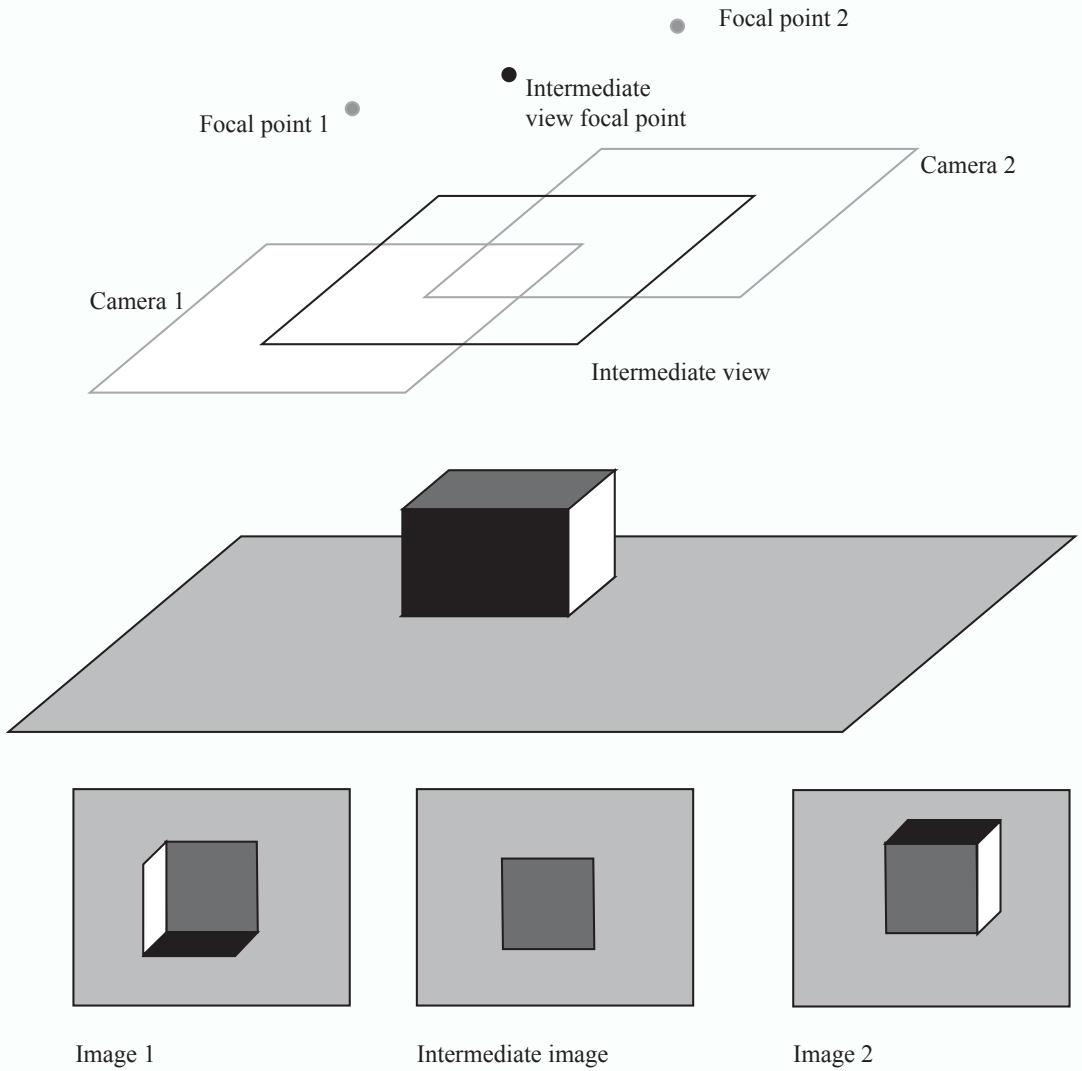
Pragmatics

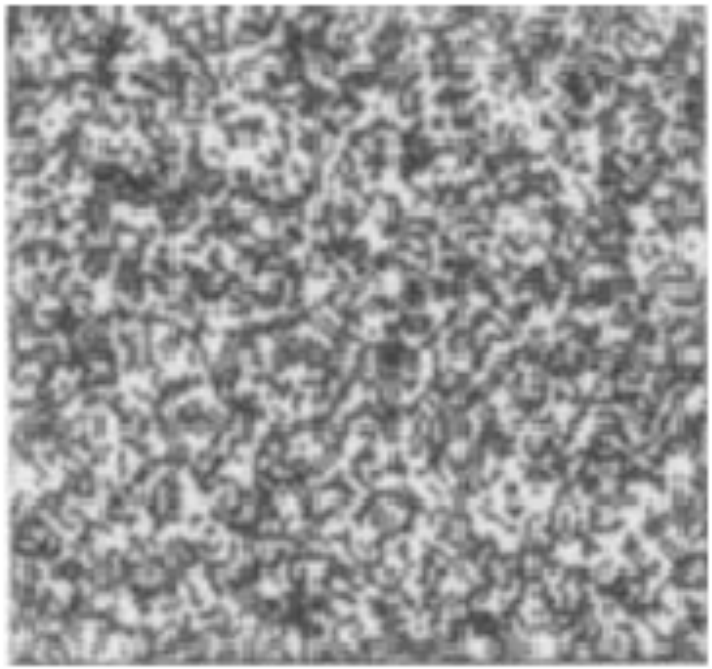
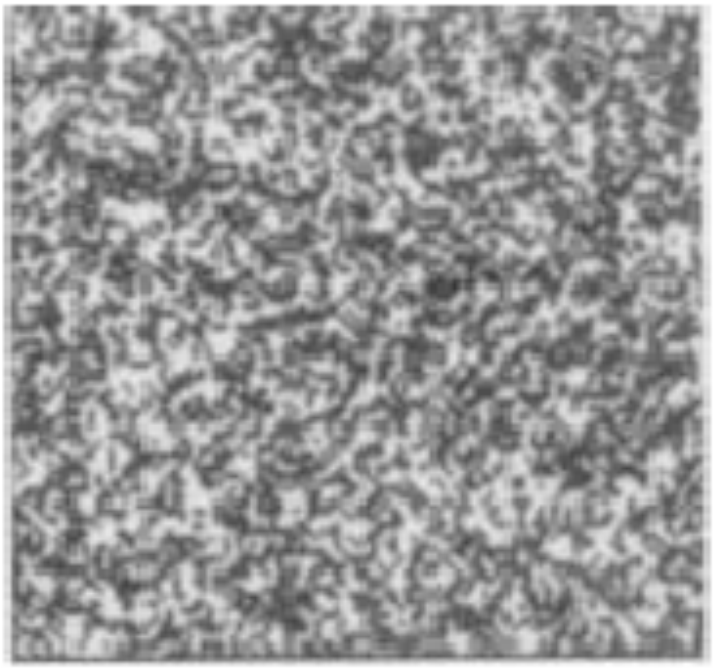
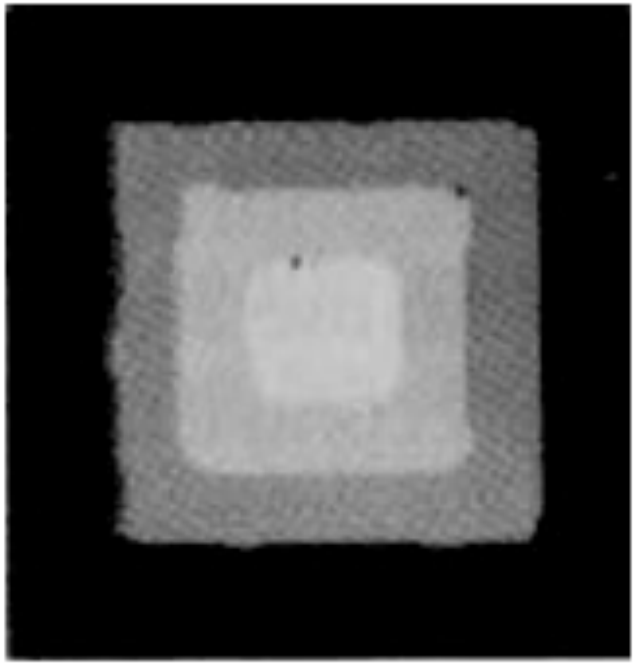
- Issue
  - Match points
- Strategy
  - correspondences occur only along scanlines
  - represent points from coarse to fine
    - scale problems - some scales are misleading
- Issue
  - some points don’t have correspondences (occlusion)
- Match left to right, then right to left
  - if they don’t agree, break match

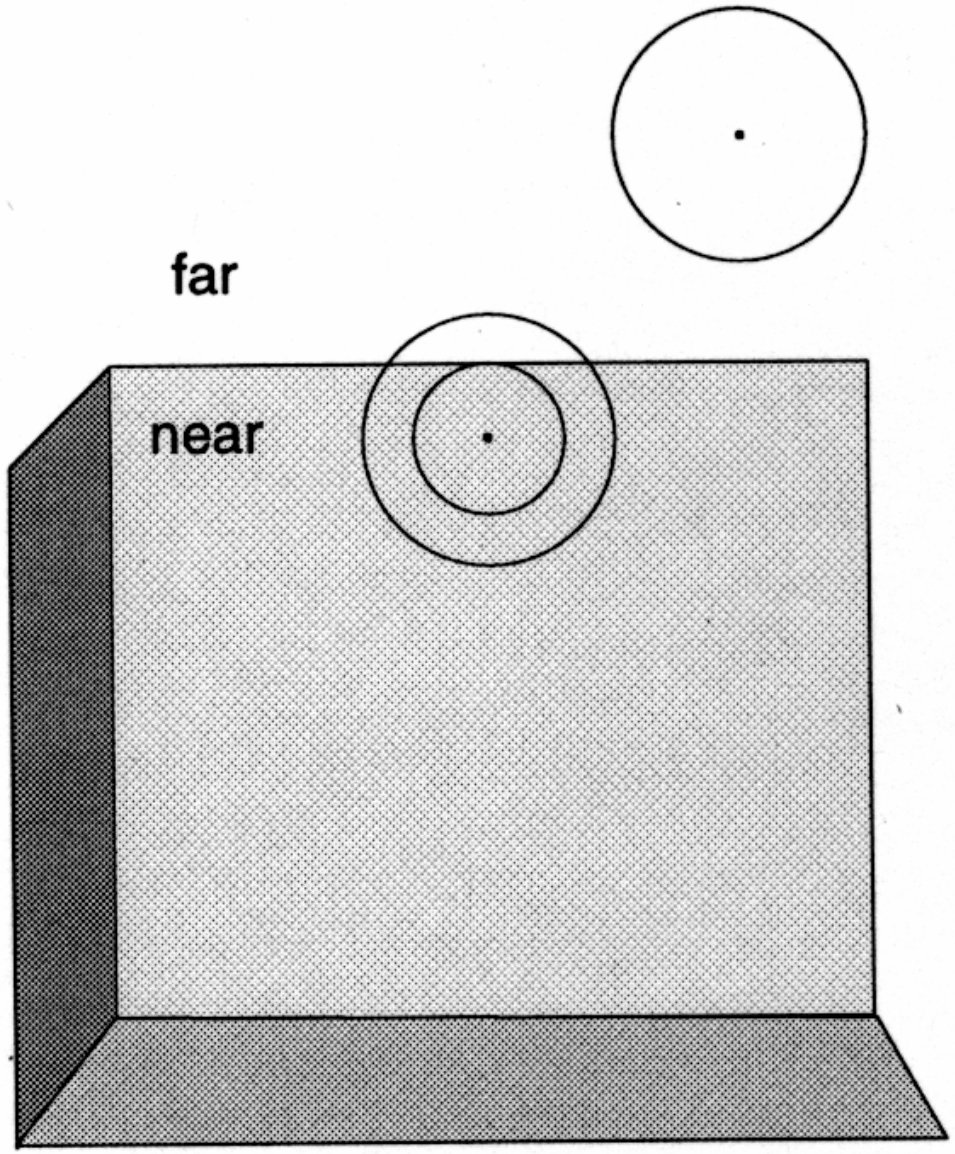
Some points don’t have matches



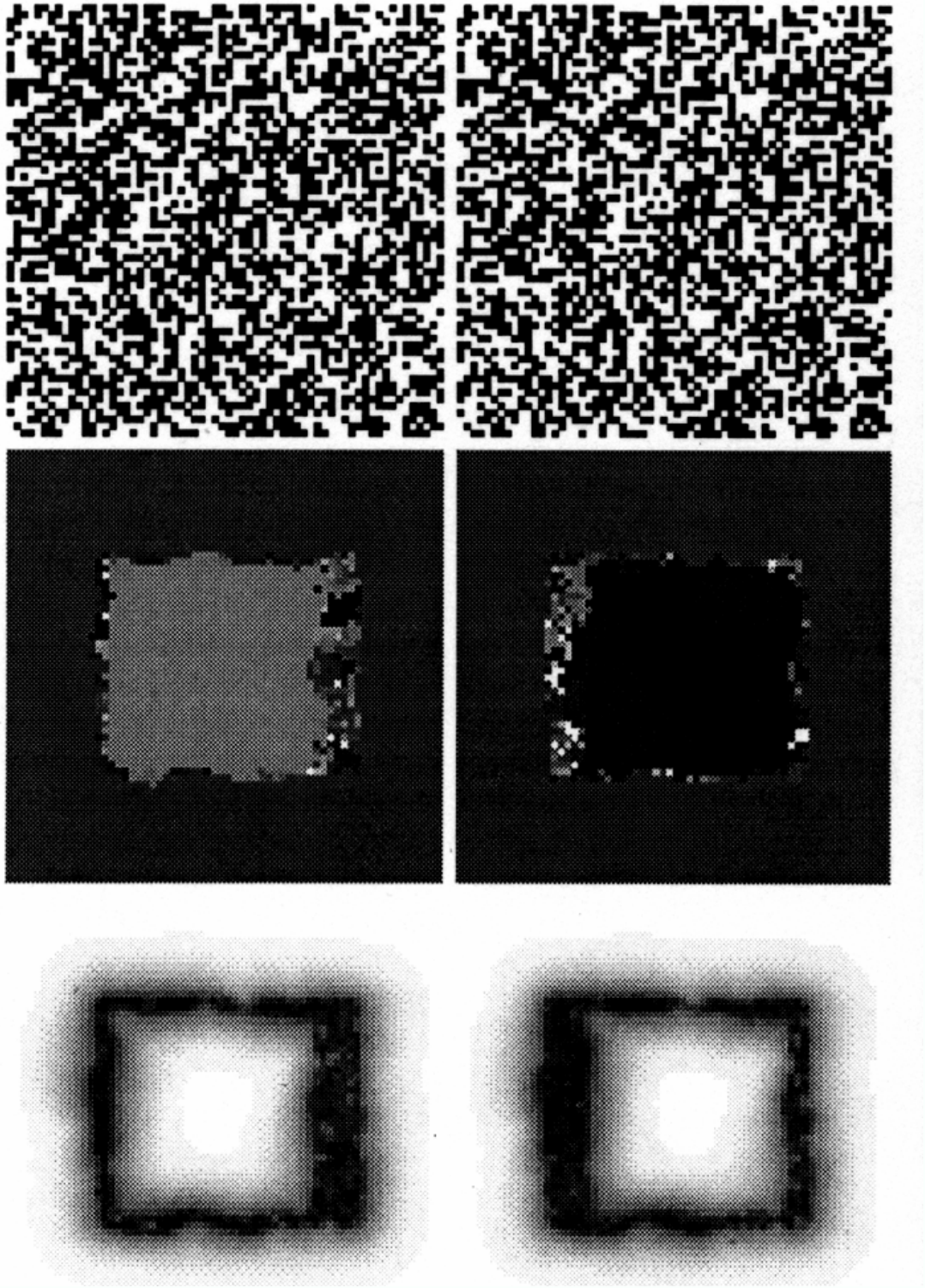
Some points don’t have matches



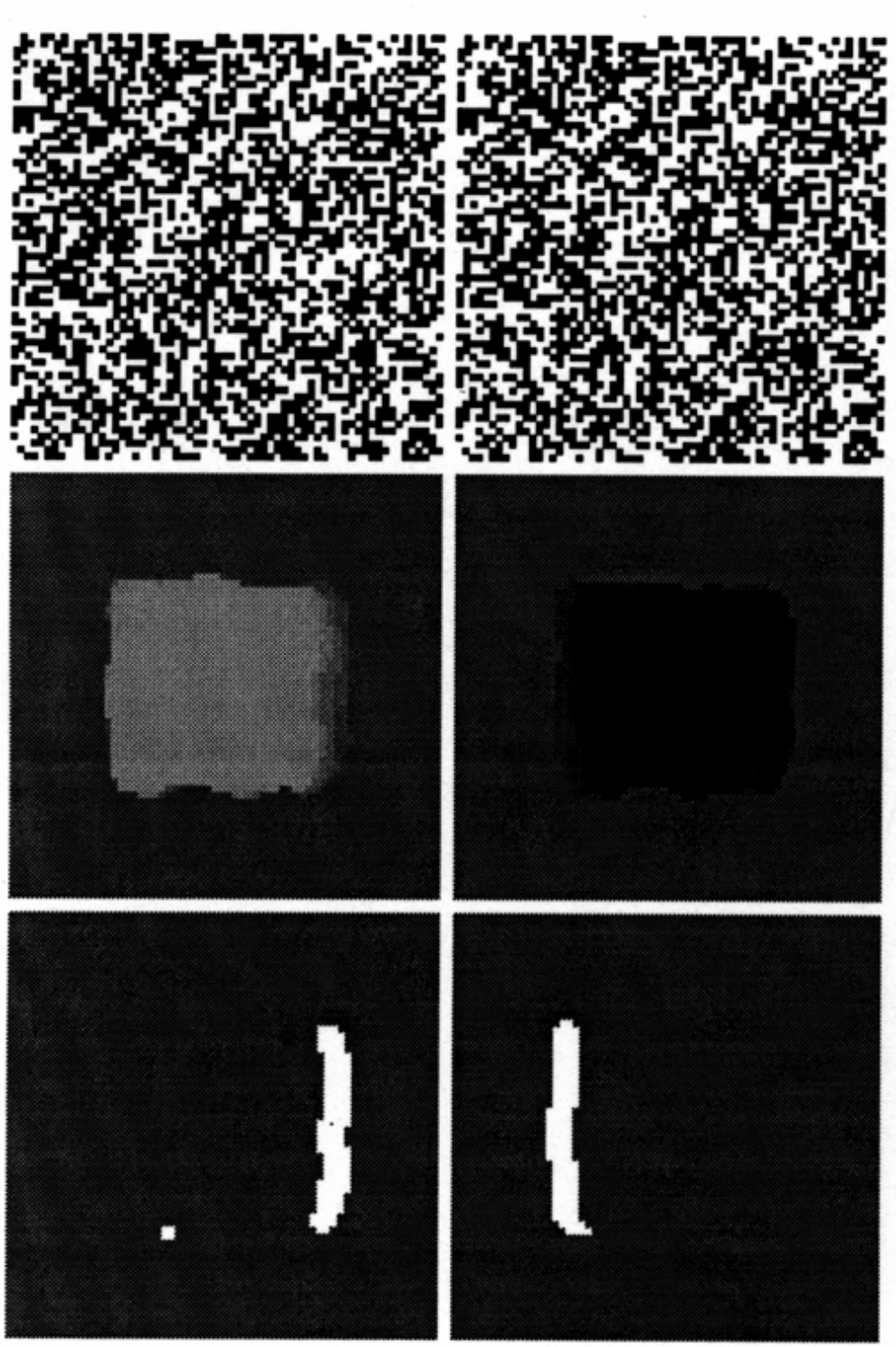






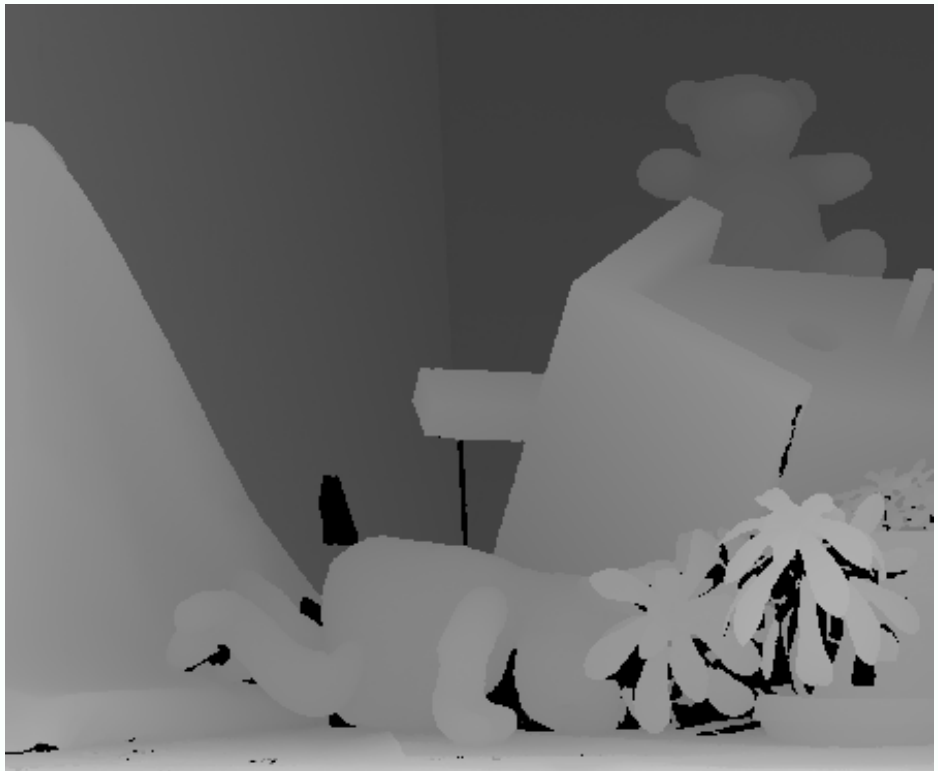
From Jones and Malik, "A computational framework for determining
Stereo correspondences from a set of linear spatial filters

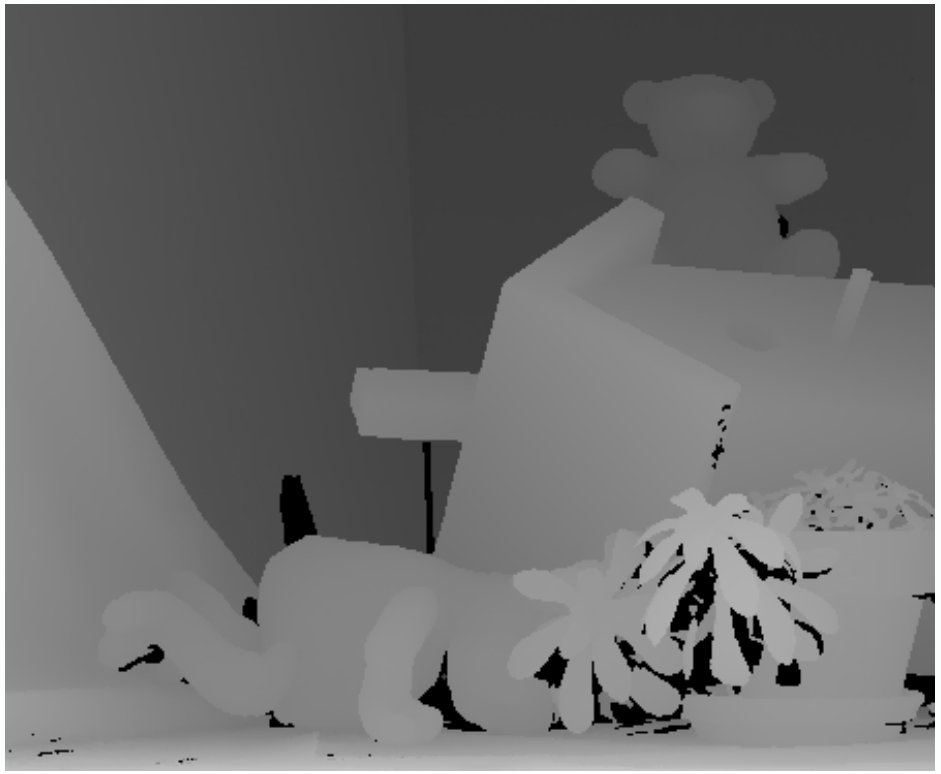


From Jones and Malik, “A computational framework for determining
Stereo correspondences from a set of linear spatial filters

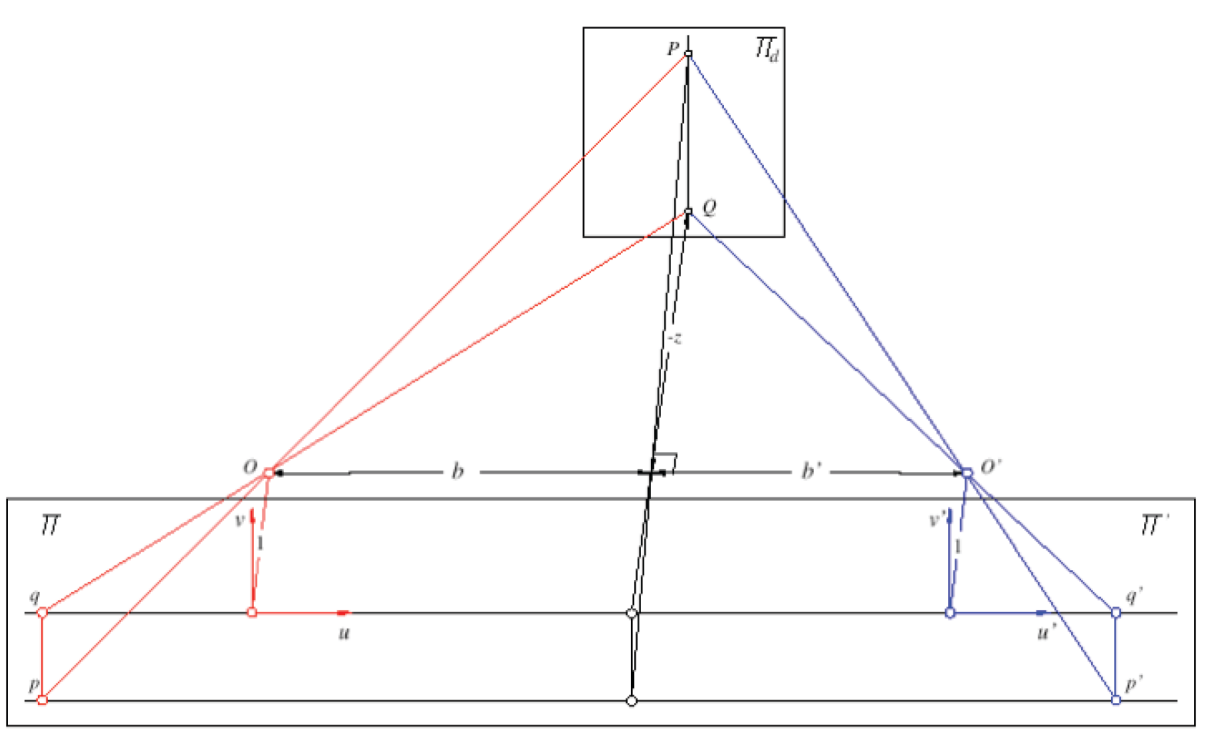


From Jones and Malik, “A computational framework for determining
Stereo correspondences from a set of linear spatial filters





Stereo as an optimization problem



- Original:
  - find q, q' that match, and infer depth
- Now:
  - choose value of depth at q; then quality of match at q' is cost
  - optimize this

Discrete Quadratic Programs

- Minimize:
  - x^T A x + b^t x
  - subject to: x is a vector of discrete values
- Summary:
  - turn up rather often in early vision
    - from Markov random fields; conditional random fields; etc.
  - variety of cases:
    - some instances are polynomial
    - most are NP hard
      - but have extremely efficient, fast approximation algorithms
      - typically based on graph cuts, qv

Stereo as an optimization problem

- Typically:
  - quantize depth to a fixed number of levels
  - unary cost is color match
    - (photometric consistency constraint)
    - it can be helpful to match intensity gradients, too
  - pairwise cost from smoothness constraint on recovered depths
    - eg depth gradient not too big, etc.
  - massive discrete quadratic program

Stereo as an optimization problem (II)

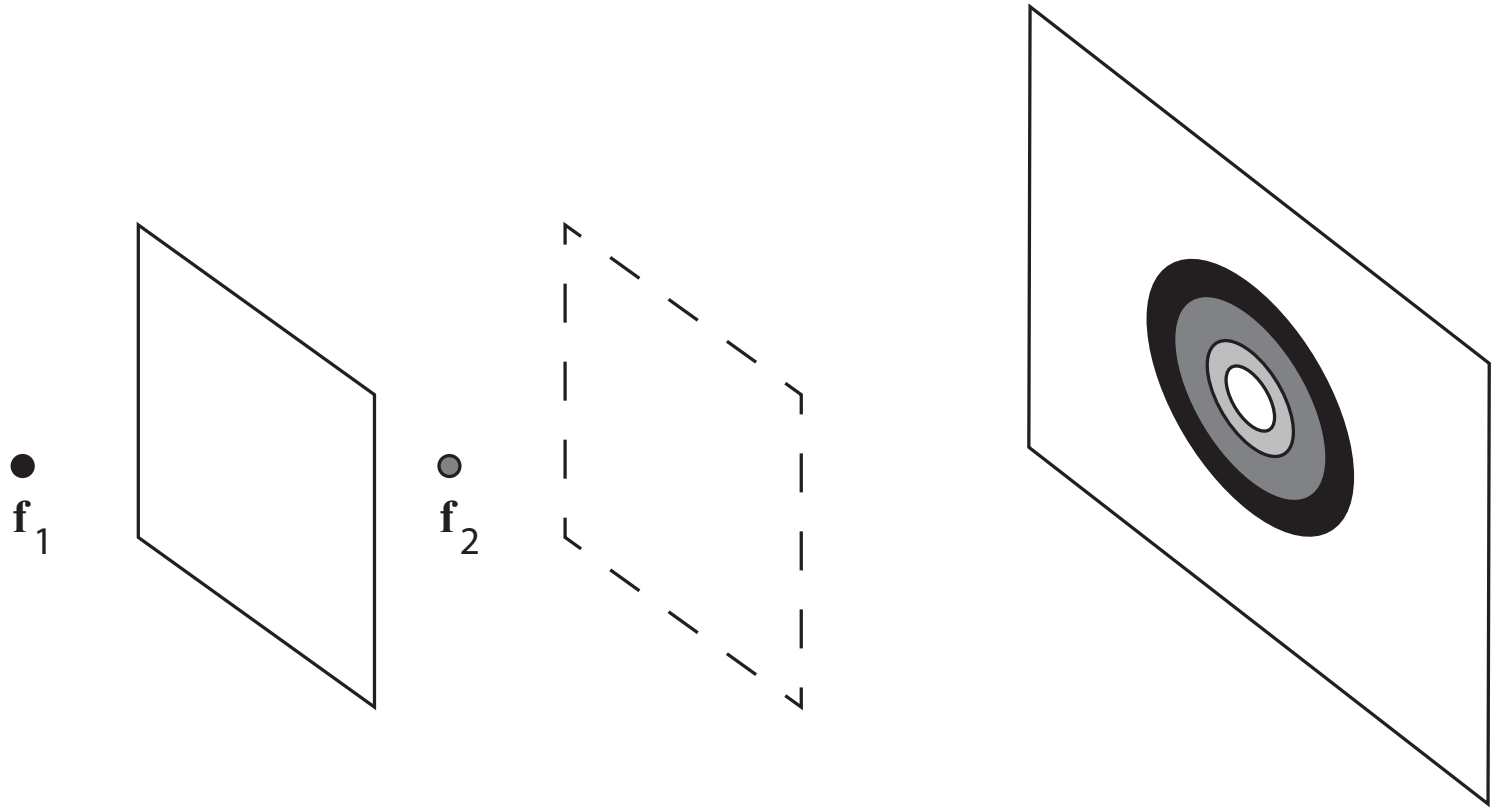
- Segment images into regions
  - NOT semantic; small, constant color+texture
- Each region is assumed to have a linear disparity
  - $d(x, y)=a\ x + b\ y+c$
- Find a quantized "vocabulary" of such disparities
  - eg by initial disparity, incremental fitting
- For each region, choose the "best" in the "vocabulary"
  - This is a discrete optimization problem
  - It's quadratic
    - unary term - does the chosen vocab item "agree" with color data?
    - binary term - are neighboring pairs of models "similar" on boundary?

Stereo resources

- Datasets and evaluations:
  - Middlebury stereo page has longstanding
    - datasets
    - evaluations with leaderboards
    - datasets with groundtruth
    - refs to other such collections
      - (but this is the best known, by a long way)
  - https://vision.middlebury.edu/stereo/

Optic flow

- Generically:
  - a “small” camera movement yields image 2 from image 1
  - determine where points in image 1 move
- Assume we’re moving rigidly in a stationary environment
  - then points will move along their epipolar lines
    - where the epipolar lines follow from fundamental matrix
      - so from camera movement
- Main point of contrast with stereo
  - Images are not usually simultaneous
    - so objects might have moved



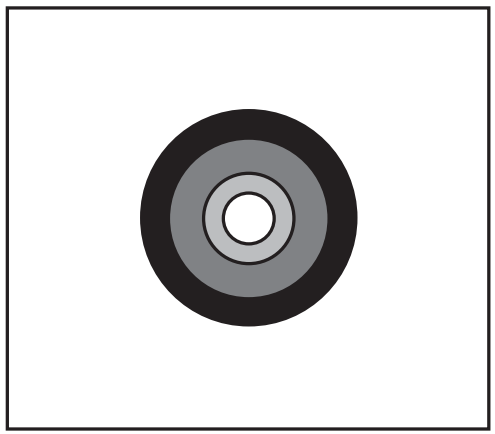


Image 1

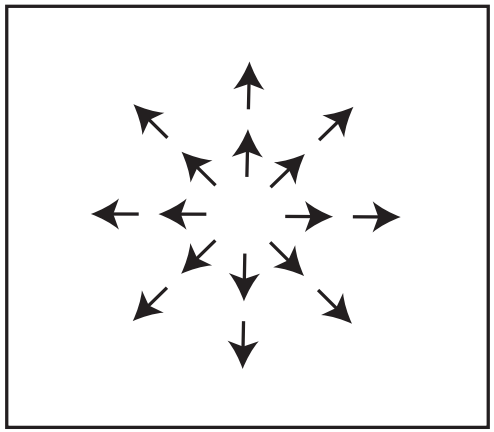


Image 1 optic flow

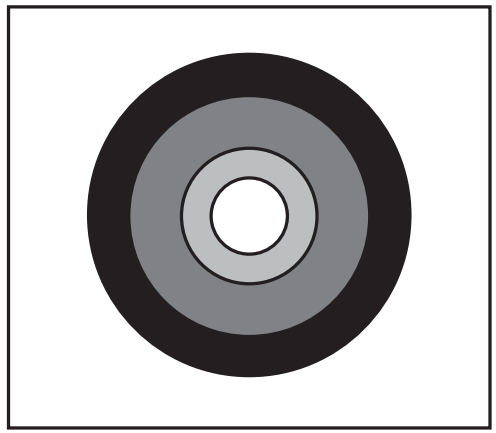
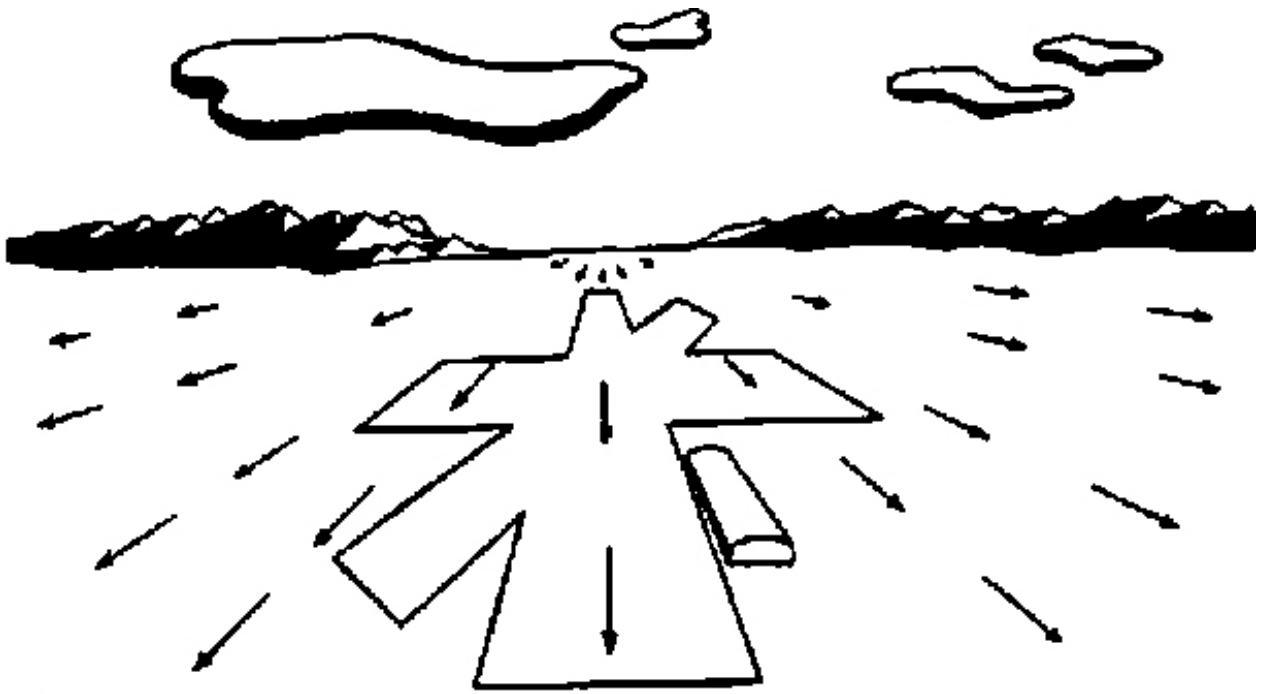
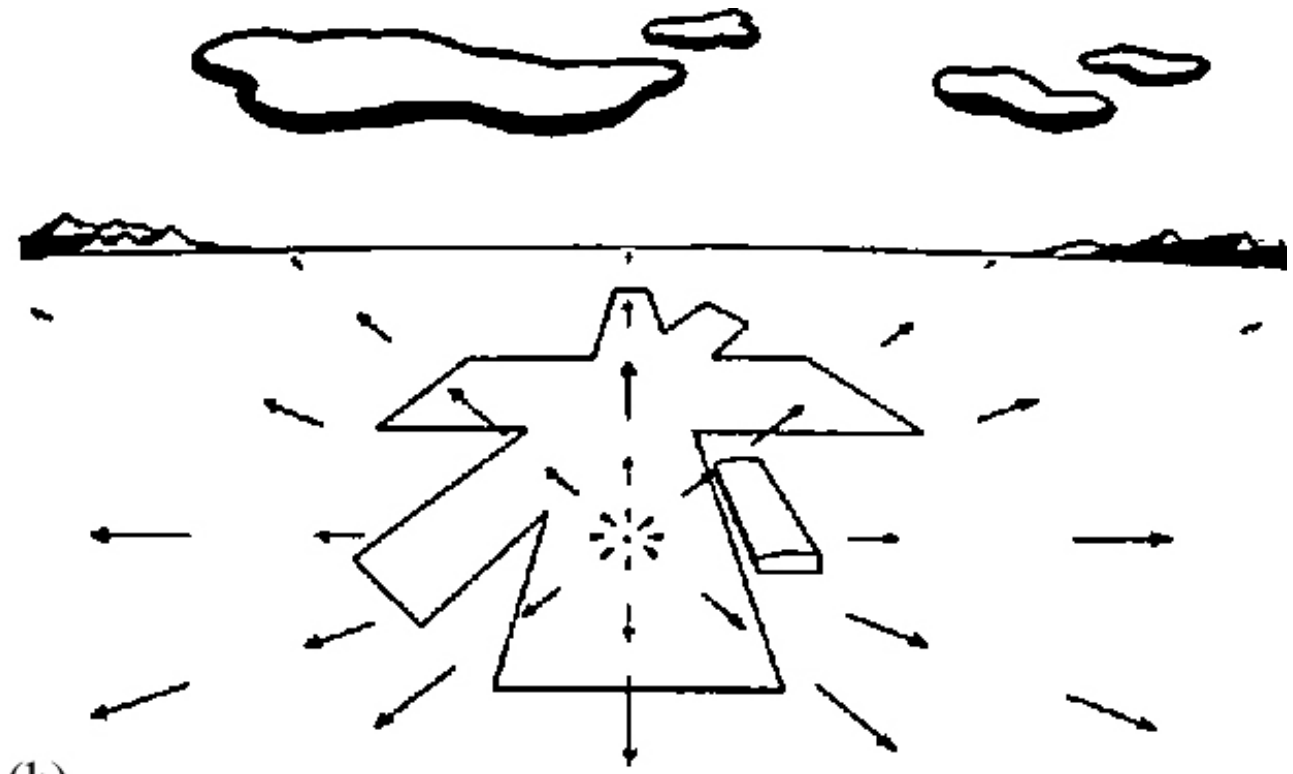


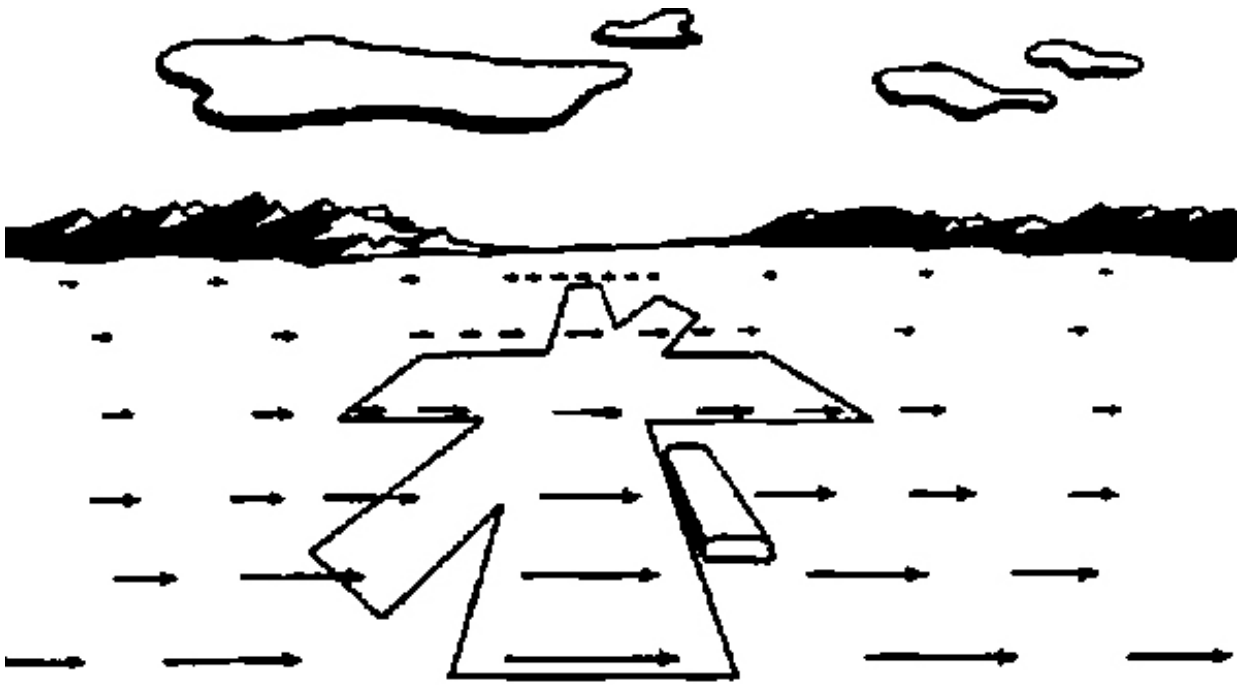
Image 2



(a)

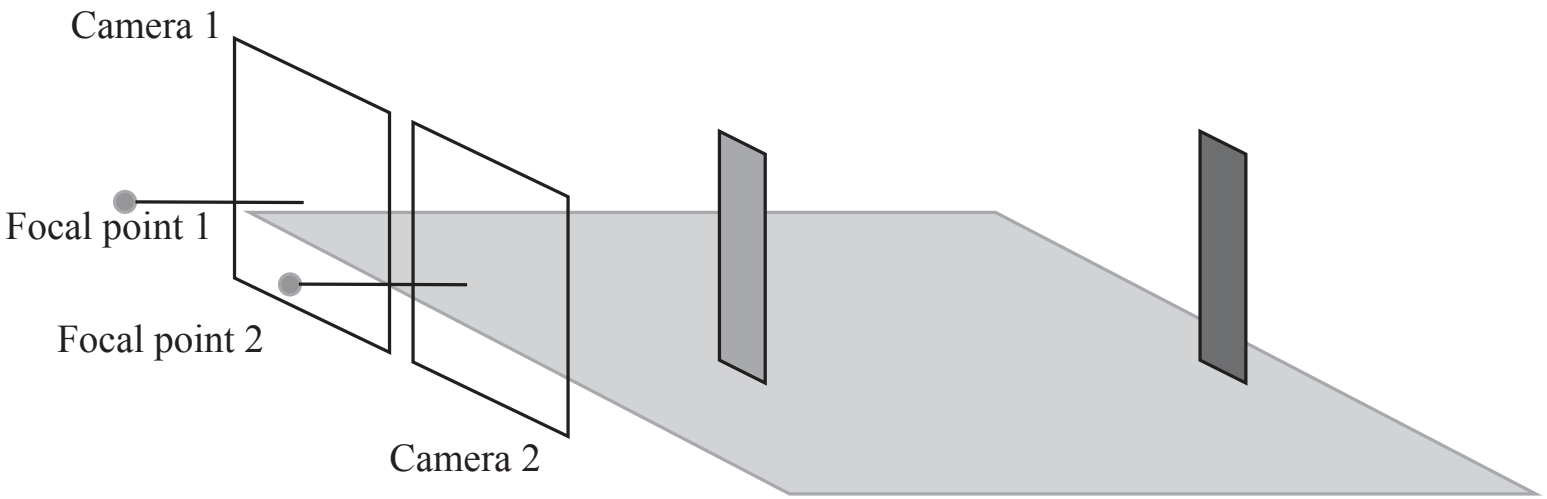


(b)



Optical flow

- Generically:
  - a “small” camera movement yields image 2 from image 1
  - determine where points in image 1 move
- Assume we’re moving rigidly in a stationary environment
  - then points will move along their epipolar lines
    - where the epipolar lines follow from fundamental matrix
      - so from camera movement
- As we saw, HOW FAR they move is determined by depth
  - and by their movement!!!



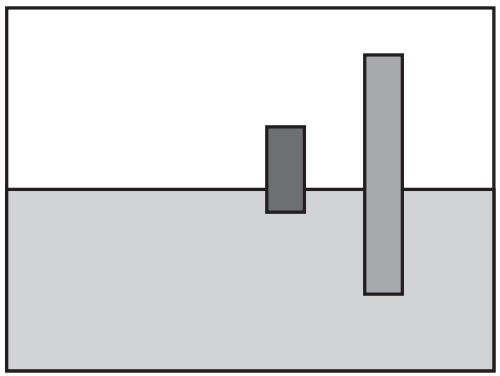


Image 1

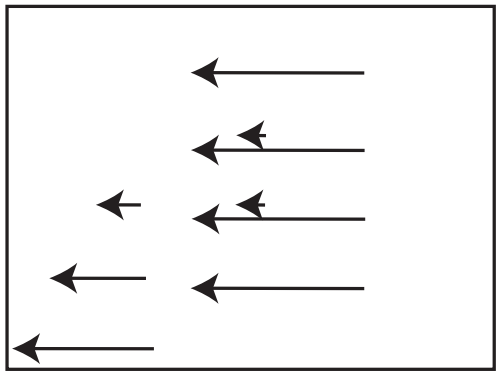


Image 1

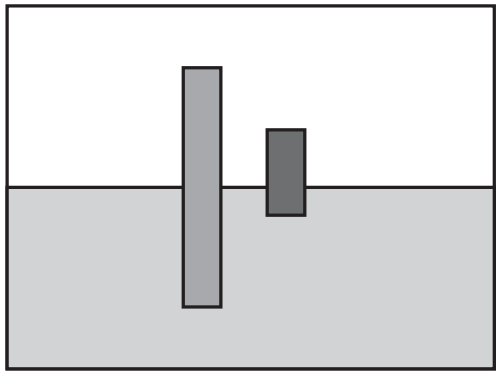


Image 2

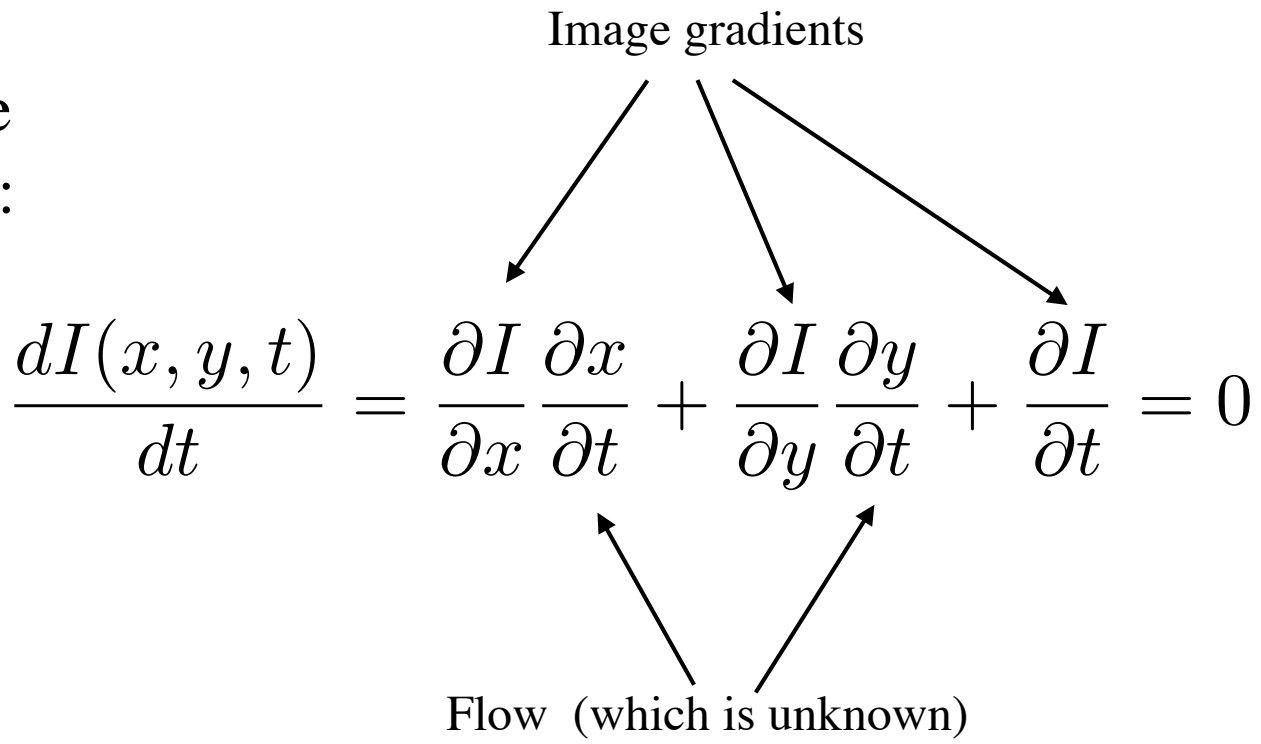

WOMEN IN MA
AND ENG
CORNELL U
dV
V

There is flow here!

For camera motions in a rigid scene, you can determine ground truth.
Evaluation is then by comparison to ground truth.

Recovering optic flow

- Huge literature
- Initial strategy:
  - Assume



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

Recovering optic flow

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

- Strategies:
  - find u(x, y), v(x, y) that minimizes some smoothness cost
    - subject to constraint on flow
    - what smoothness cost?
    - how to impose constraint?
  - assume flow has some parametric form within windows (eg. constant)
    - choose parameters to minimize error in window
    - what parametric model?
    - what windows?
  - If few or no objects move
    - impose a parametric depth model, and use that


Middlebury College
Computer Science
annual high school
computition day
Middlebury College
Computer Science
annual high school
computition day

If objects are moving, much harder to determine ground truth.

IDEA: Interpolate flow to get intermediate frame.

Evaluation is then by comparing interpolate to ground truth frame.

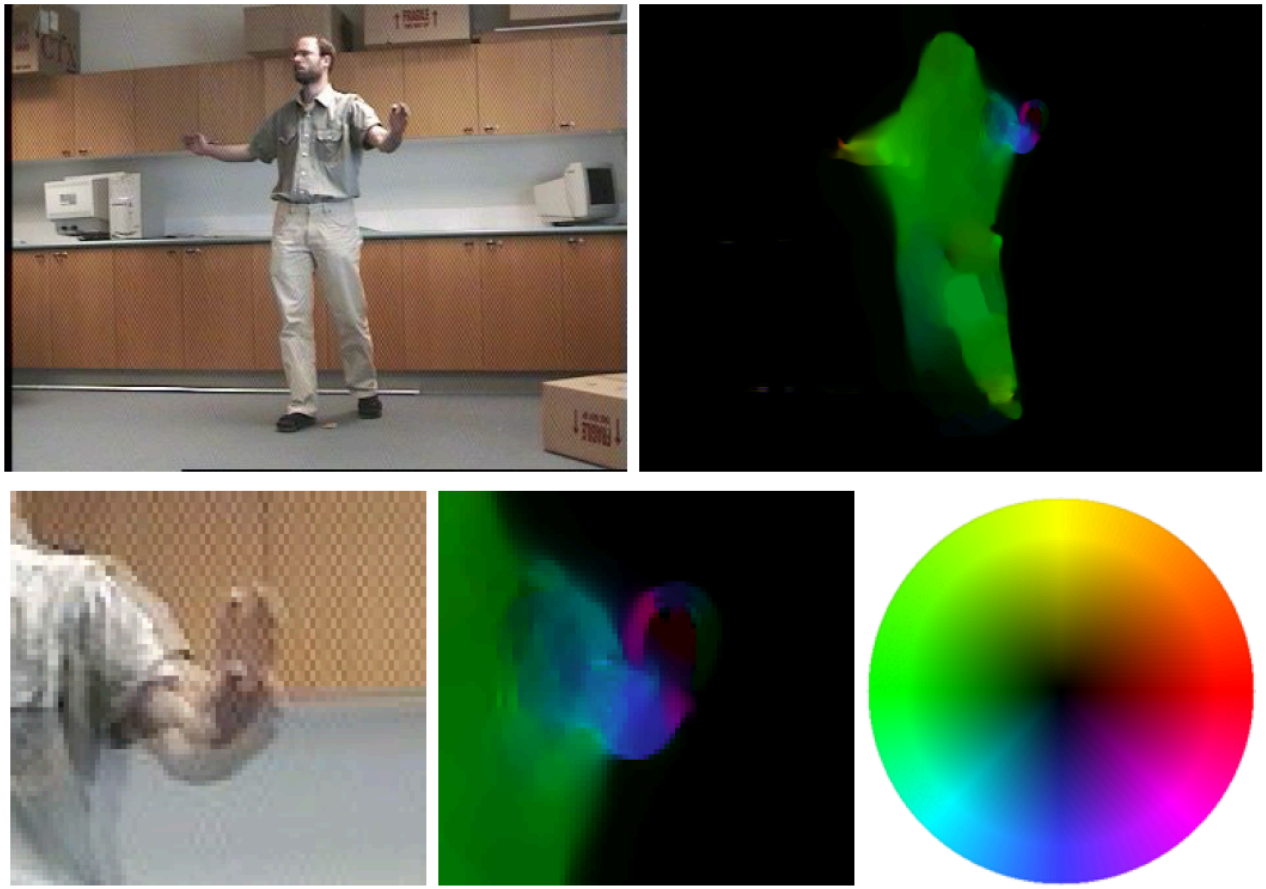


Figure 1. **Top row:** Image of a sequence where the person is stepping forward and moving his hands. The optical flow estimated with the method from [4] is quite accurate for the main body and the legs, but the hands are not accurately captured. **Bottom row, left:** Overlay of two successive frames showing the motion of one of the hands. **Center:** The arm motion is still good but the hand has a smaller scale than its displacement leading to a local minimum. **Right:** Color map used to visualize flow fields in this paper. Smaller vectors are darker and color indicates the direction.

Brox et al 09

Strategy

- Segment into regions, estimate region correspondences
  - use to inform flow estimate

Figure 9. **Left:** Two overlaid images of a tennis player in action. **Center left:** Region correspondences. **Center right:** Result with optical flow from [4]. The motion of the right leg is too fast to be estimated. **Right:** The proposed method captures the motion of the leg.

Brox et al 09

Optical flow resources

- Datasets and evaluations:
  - Middlebury optical flow page has longstanding
    - datasets
    - evaluations with leaderboards
    - datasets with groundtruth
    - refs to other such collections
      - (but this is the best known, by a long way)
  - https://vision.middlebury.edu/flow/

Next up: