Two cameras: Stereo and Optic Flow

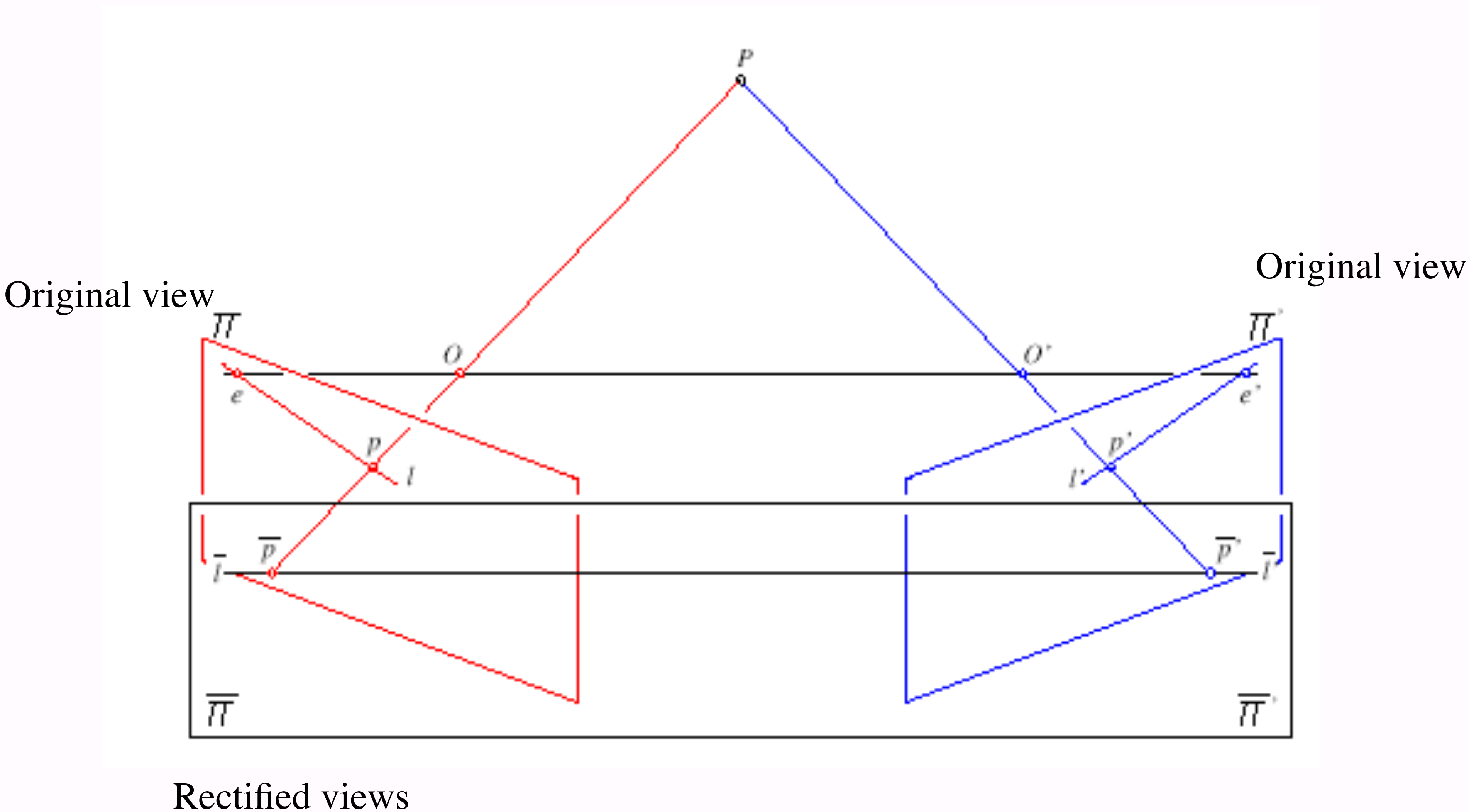
Stereopsis

- 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



Triangulation

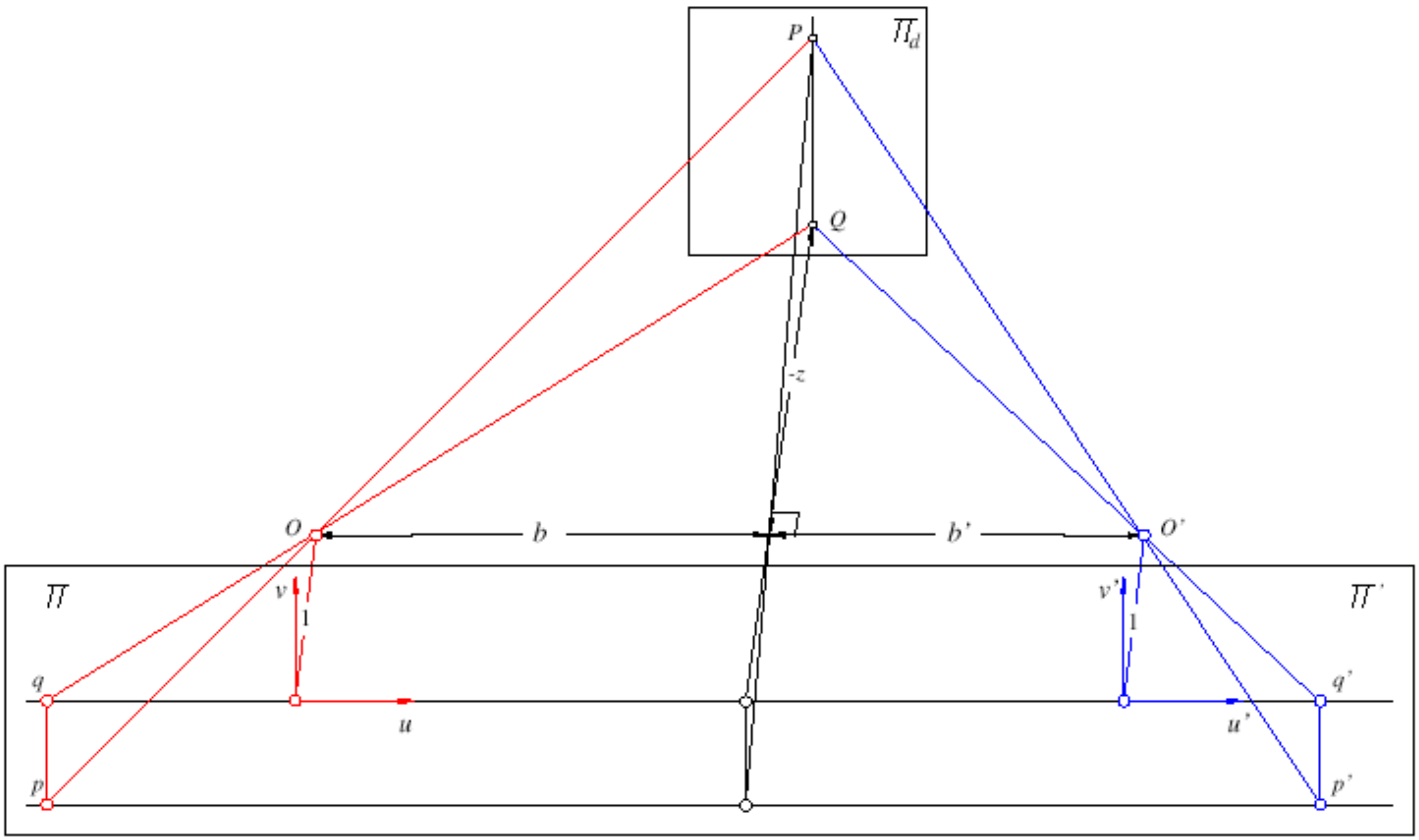
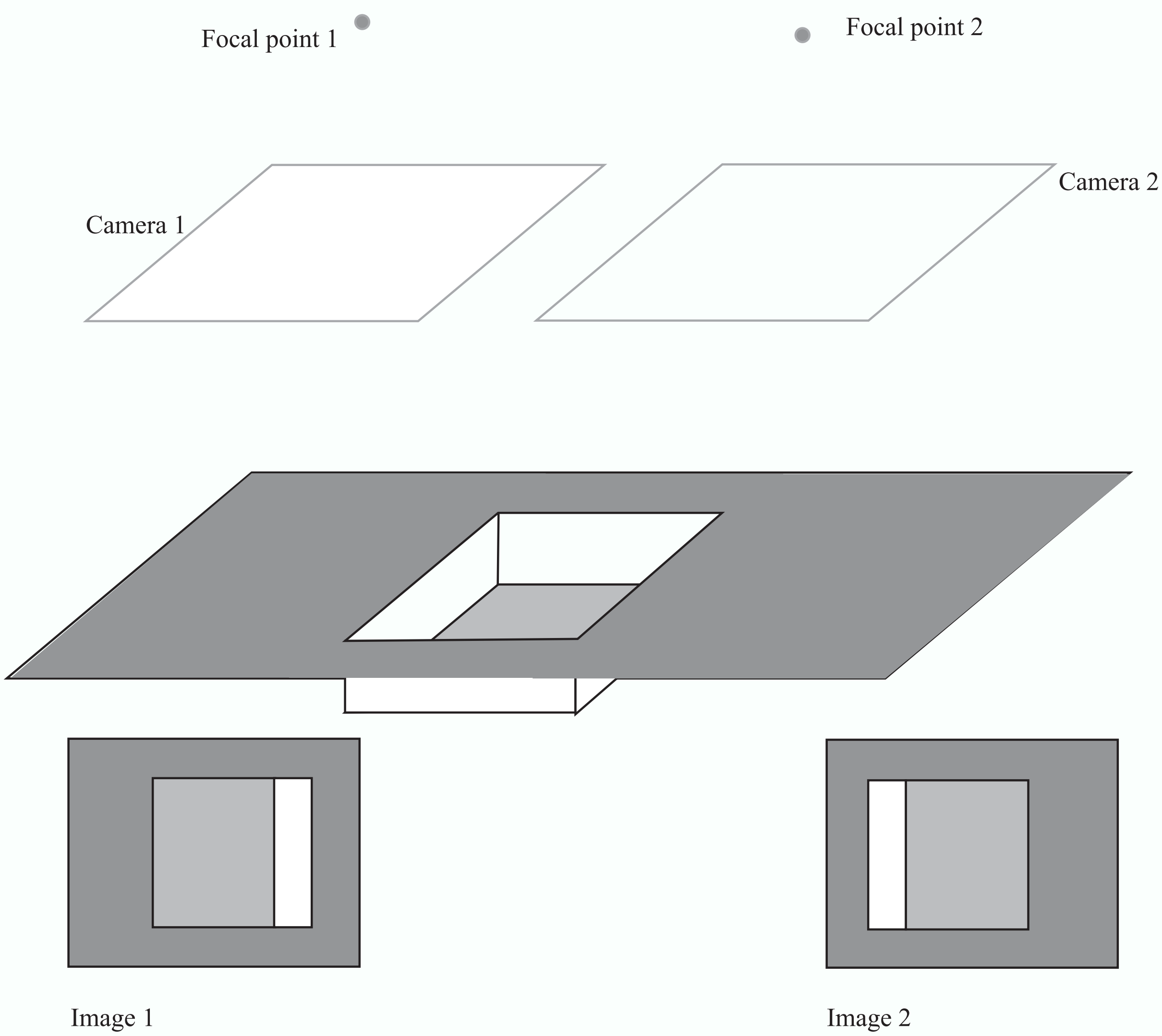


Figure 13.6. Triangulation for rectified images: the rays associated with two points p and p' on the same scanline are by construction guaranteed to intersect in some point P. As shown in the text, the depth of P relative to the coordinate system attached to the left camera is inversely proportional 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).

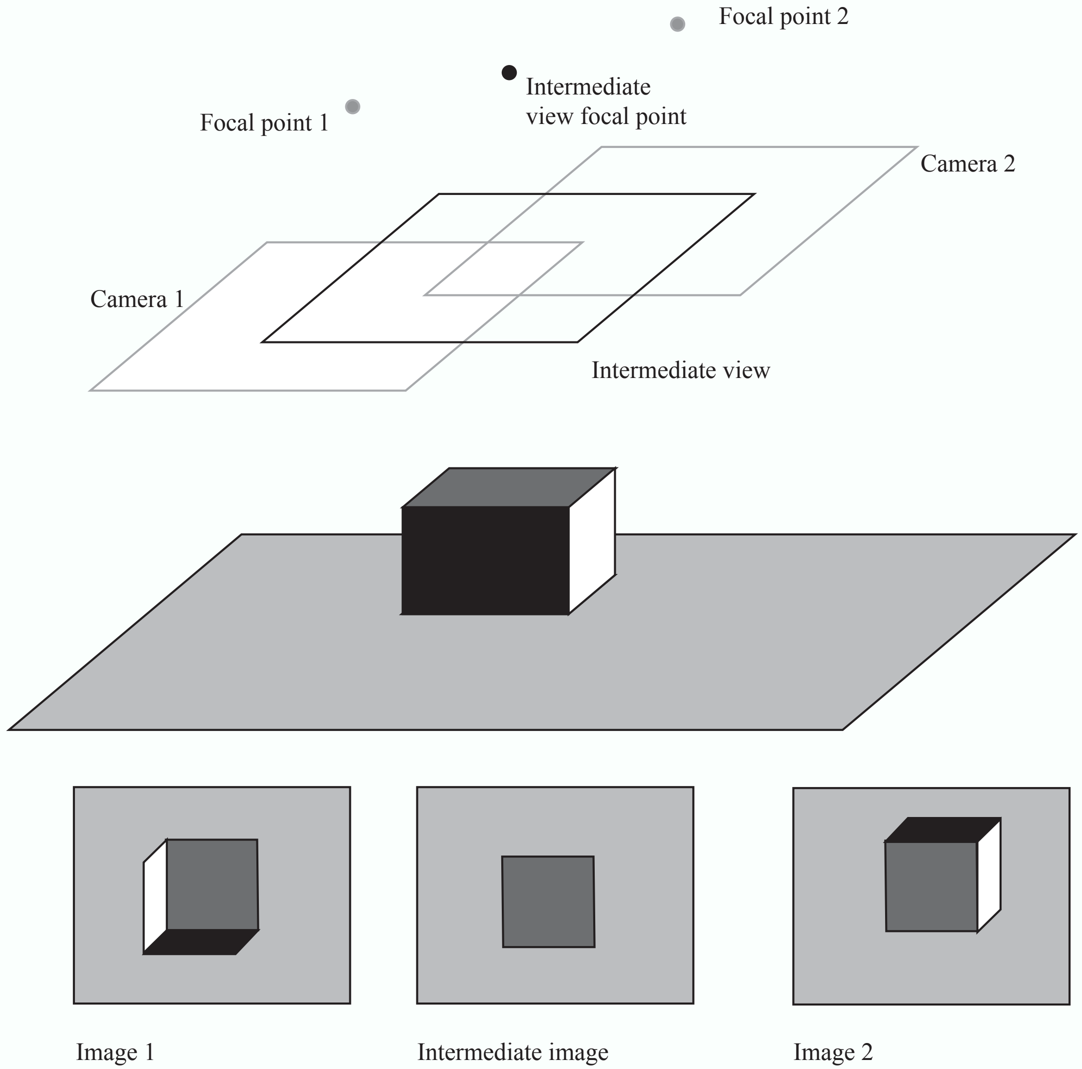
Pragmatics

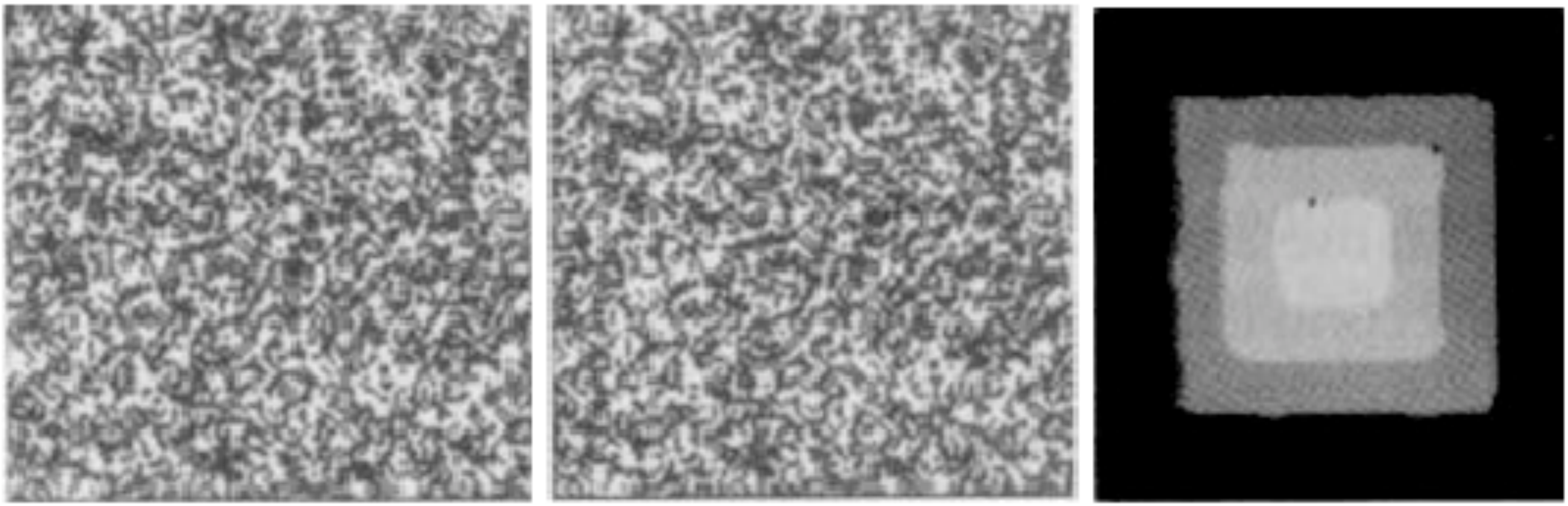
- Issue
  - Match points
- Strategy
  - correspondences occur only along scanlines
  - represent points from coarse to fine
    - scale problems - some scales are misleading
- Issue
  - some points don't have correspondences (occlusion)
- Match left to right, then right to left
  - if they don't agree,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

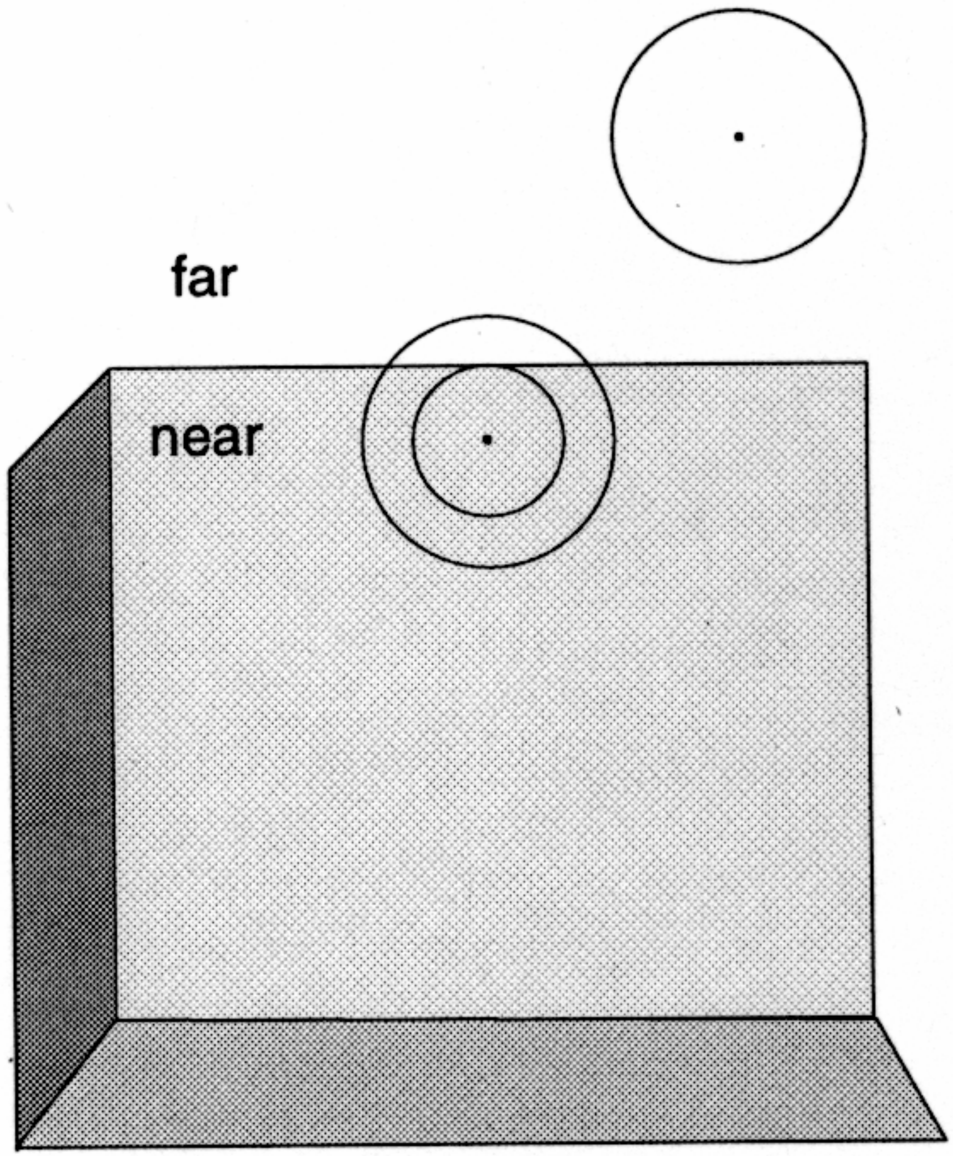
Some points don’t have matches



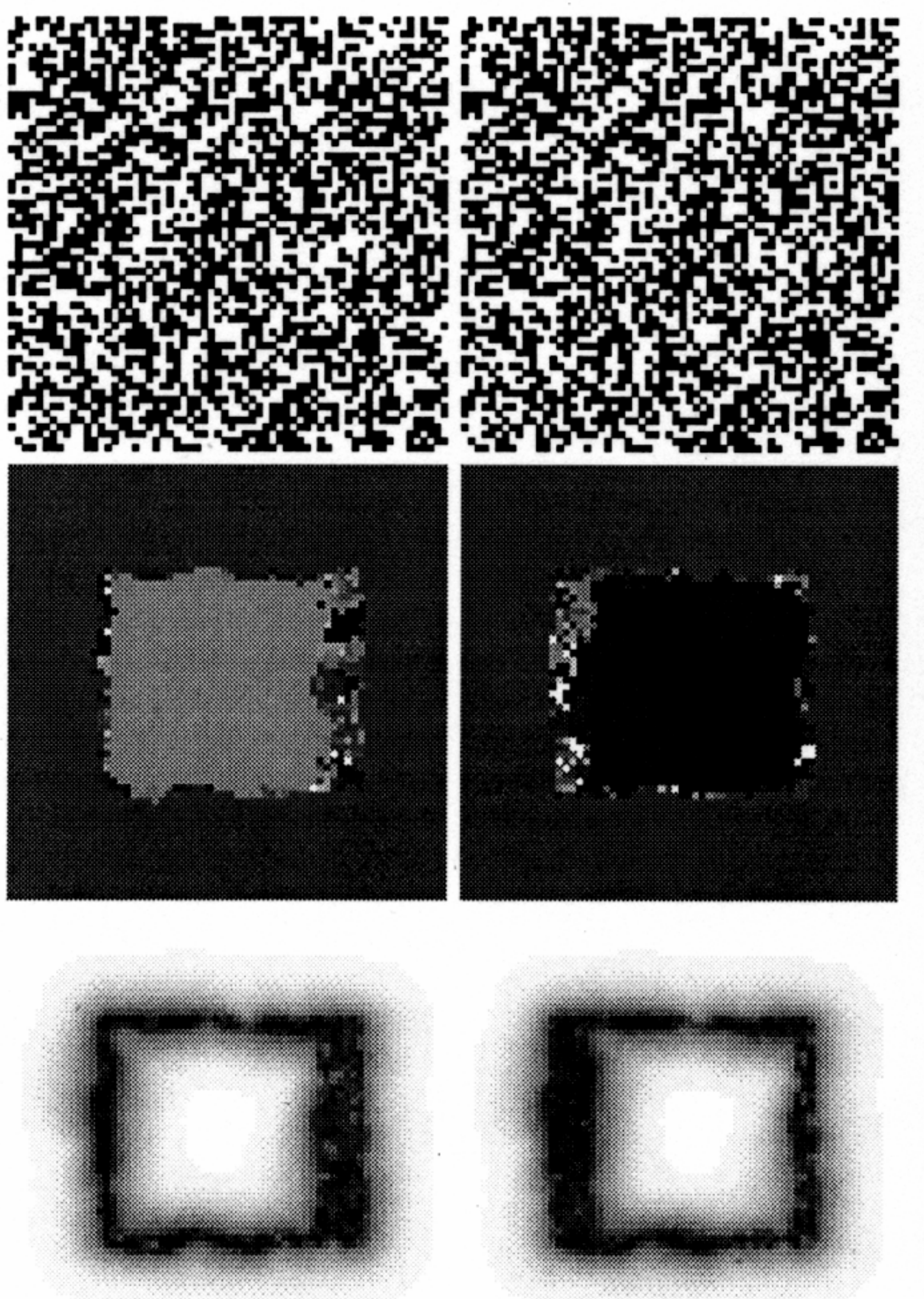
Some points don't have matches



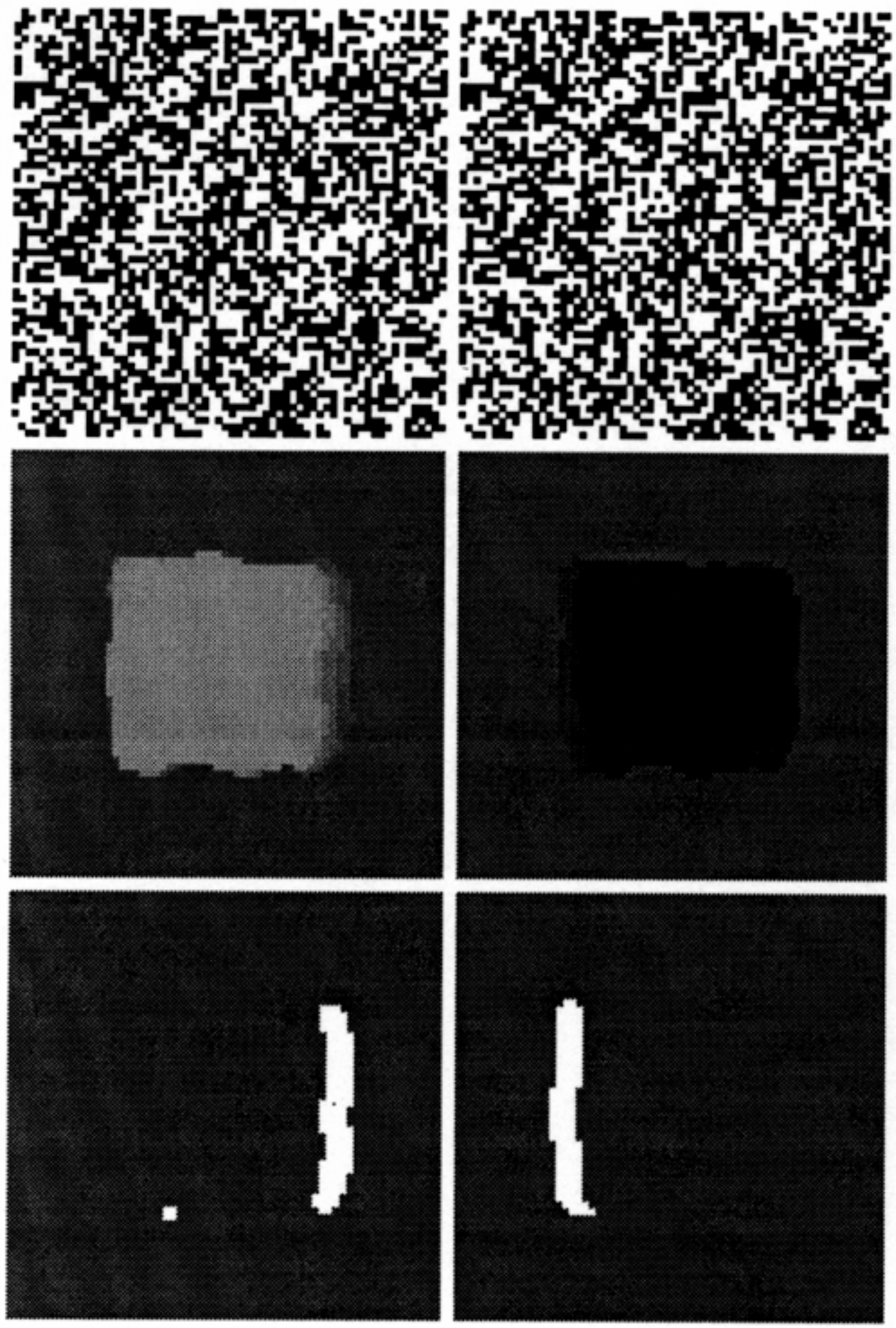




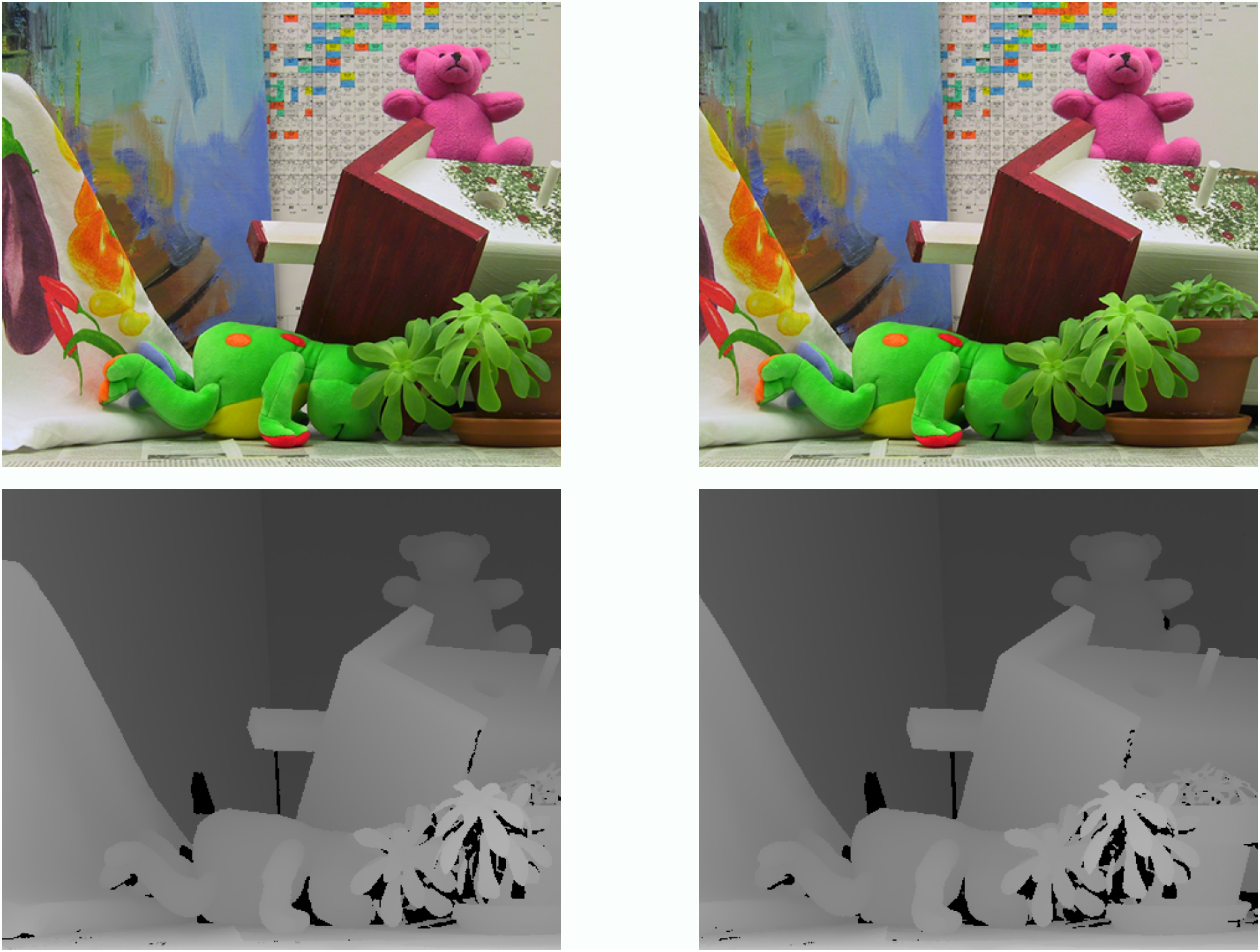
From Jones and Malik, “A computational framework for determining
Stereo correspondences from a set of linear spatial filters



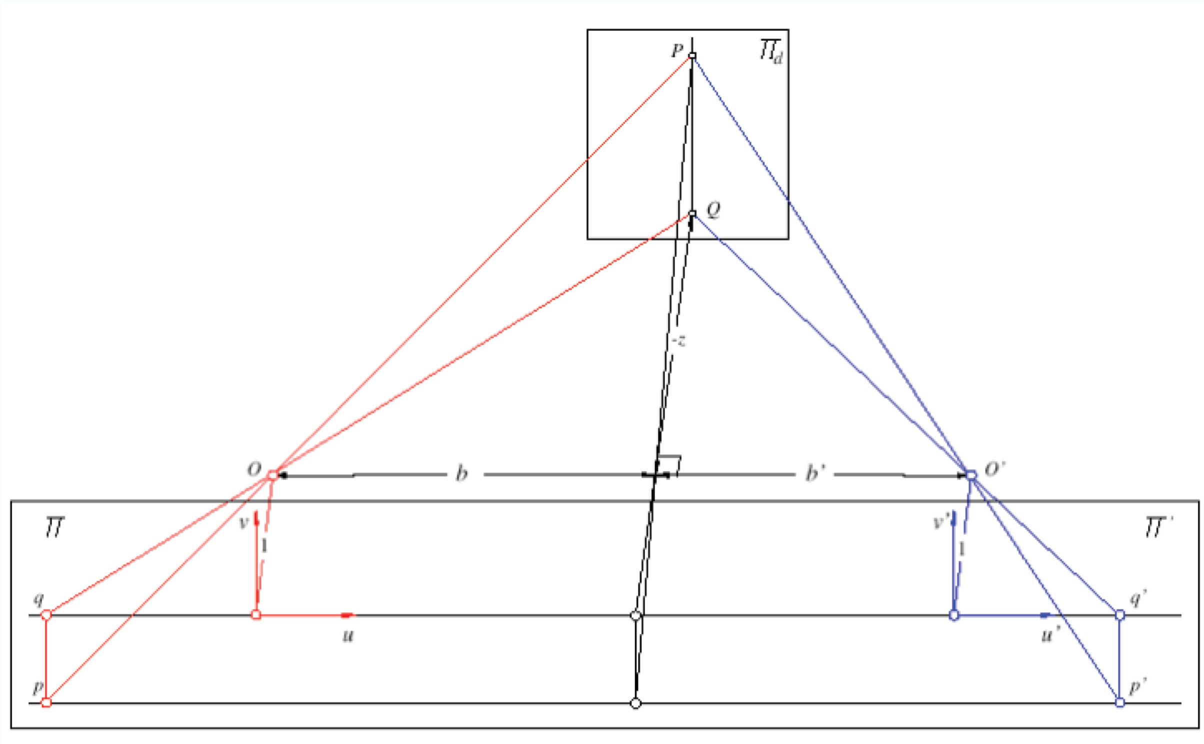
From Jones and Malik, "A computational framework for determining
Stereo correspondences from a set of linear spatial filters



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)

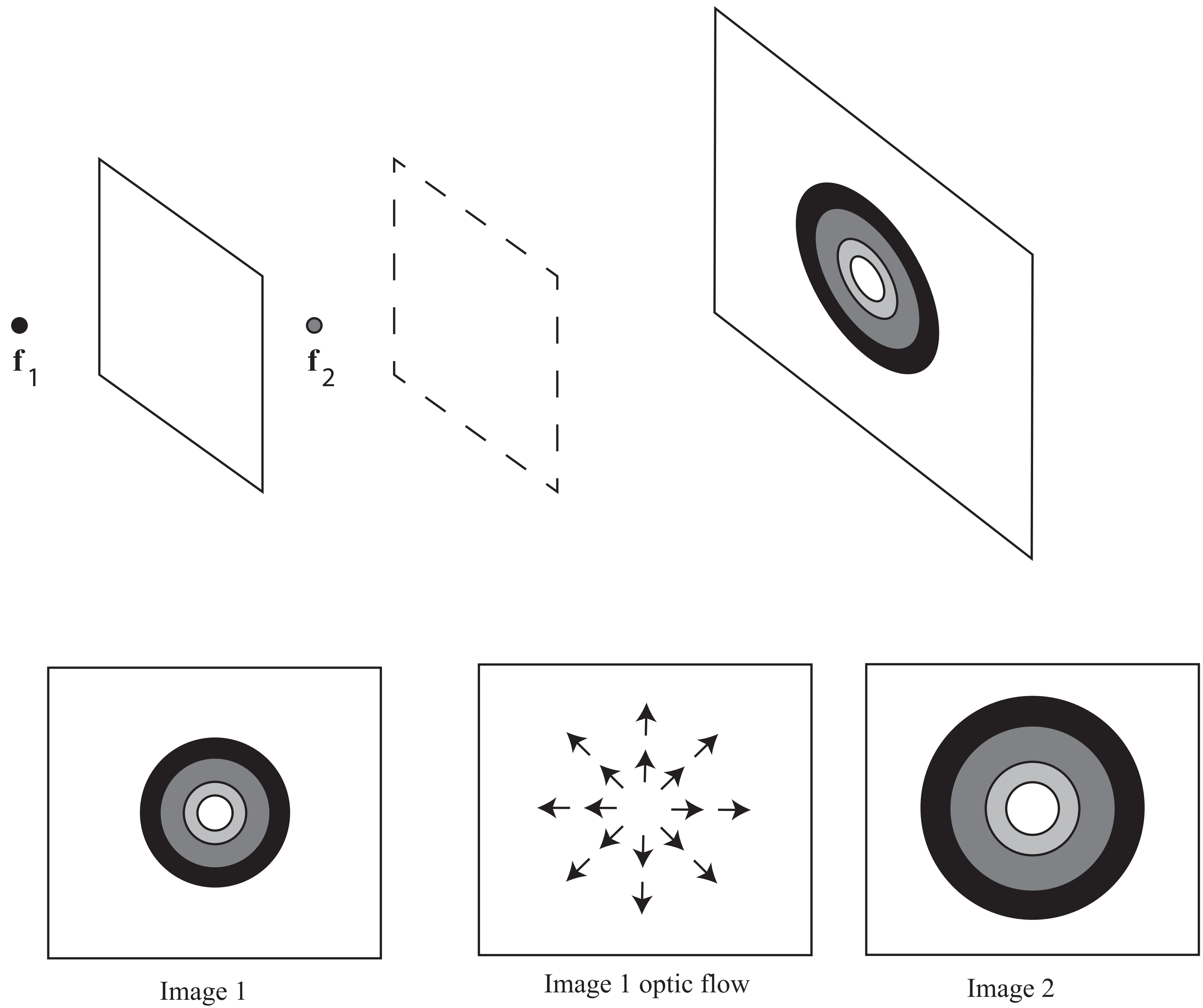
- Segment images into regions
  - NOT semantic; small, constant color+texture
- Each region is assumed to have a linear disparity
  - $d(x, y)=a x + b y+c$
- Find a quantized “vocabulary” of such disparities
  - eg by initial disparity, incremental fitting
- For each region, choose the “best” in the “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

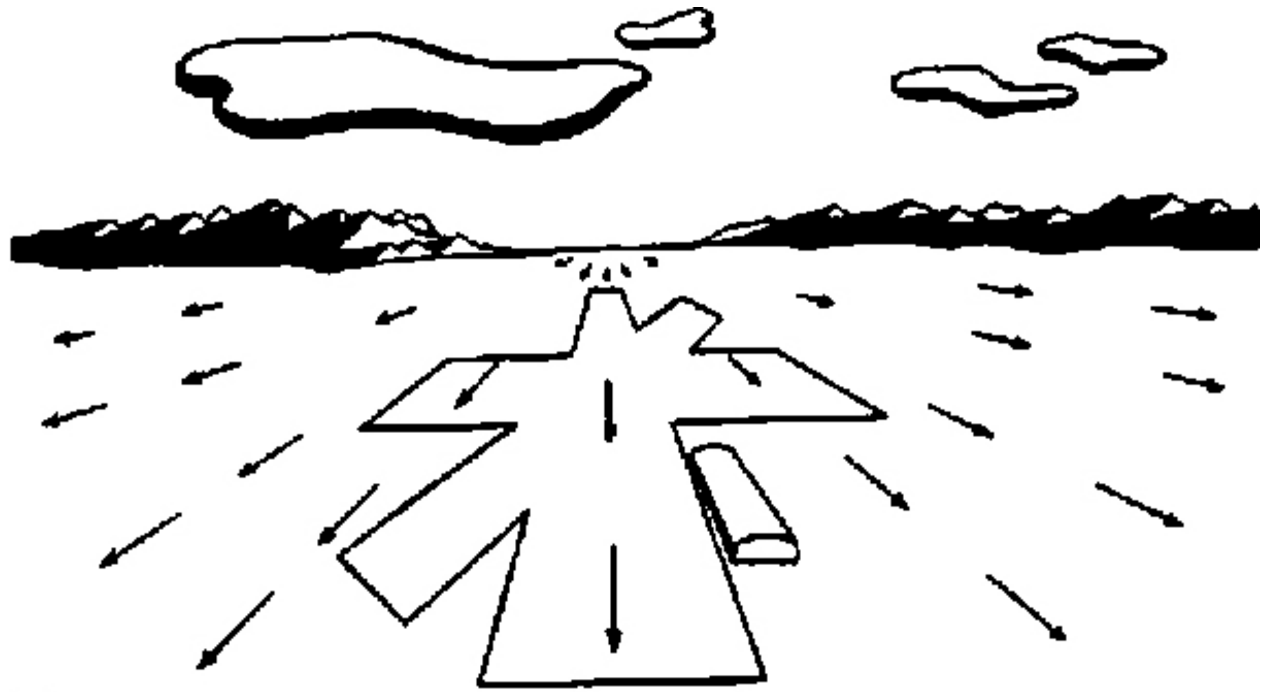
- Datasets and evaluations:
  - Middlebury stereo page has longstanding
    - datasets
    - evaluations with leaderboards
    - datasets with groundtruth
    - refs to other such collections
      - (but this is the best known, by a long way)
  - https://vision.middlebury.edu/stereo/

Optic flow

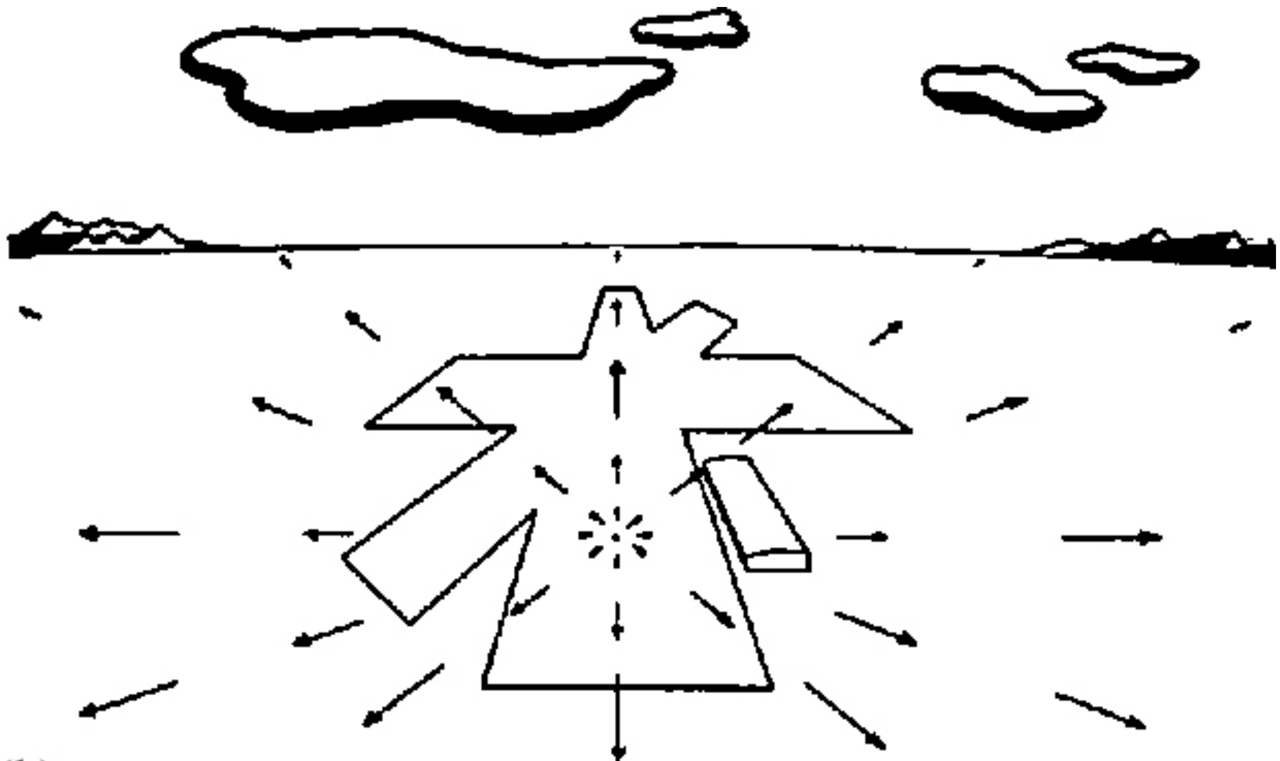
- Generically:
  - a “small” camera movement yields image 2 from image 1
  - determine where points in image 1 move
- Assume we’re moving rigidly in a stationary environment
  - then points will move along their epipolar lines
    - where the epipolar lines follow from fundamental matrix
      - so from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



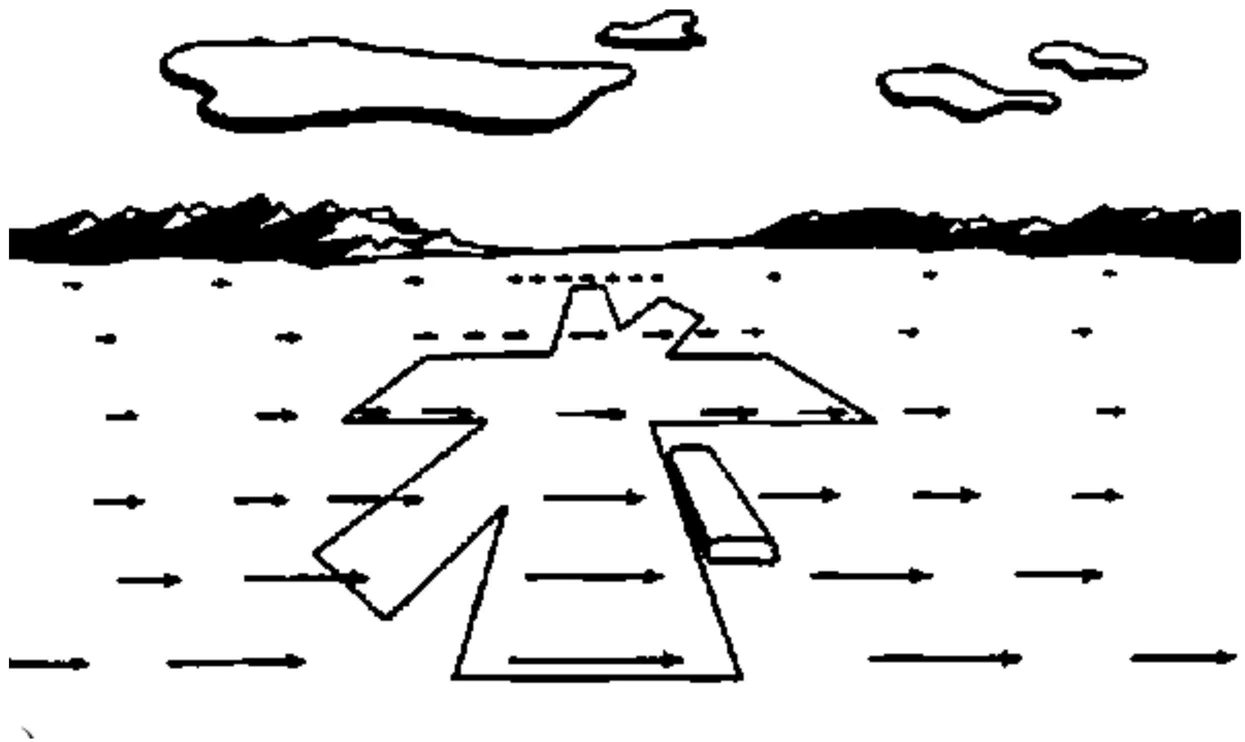
f_1
f_2
Image 1
Image 1 optic flow
Image 2



(a)

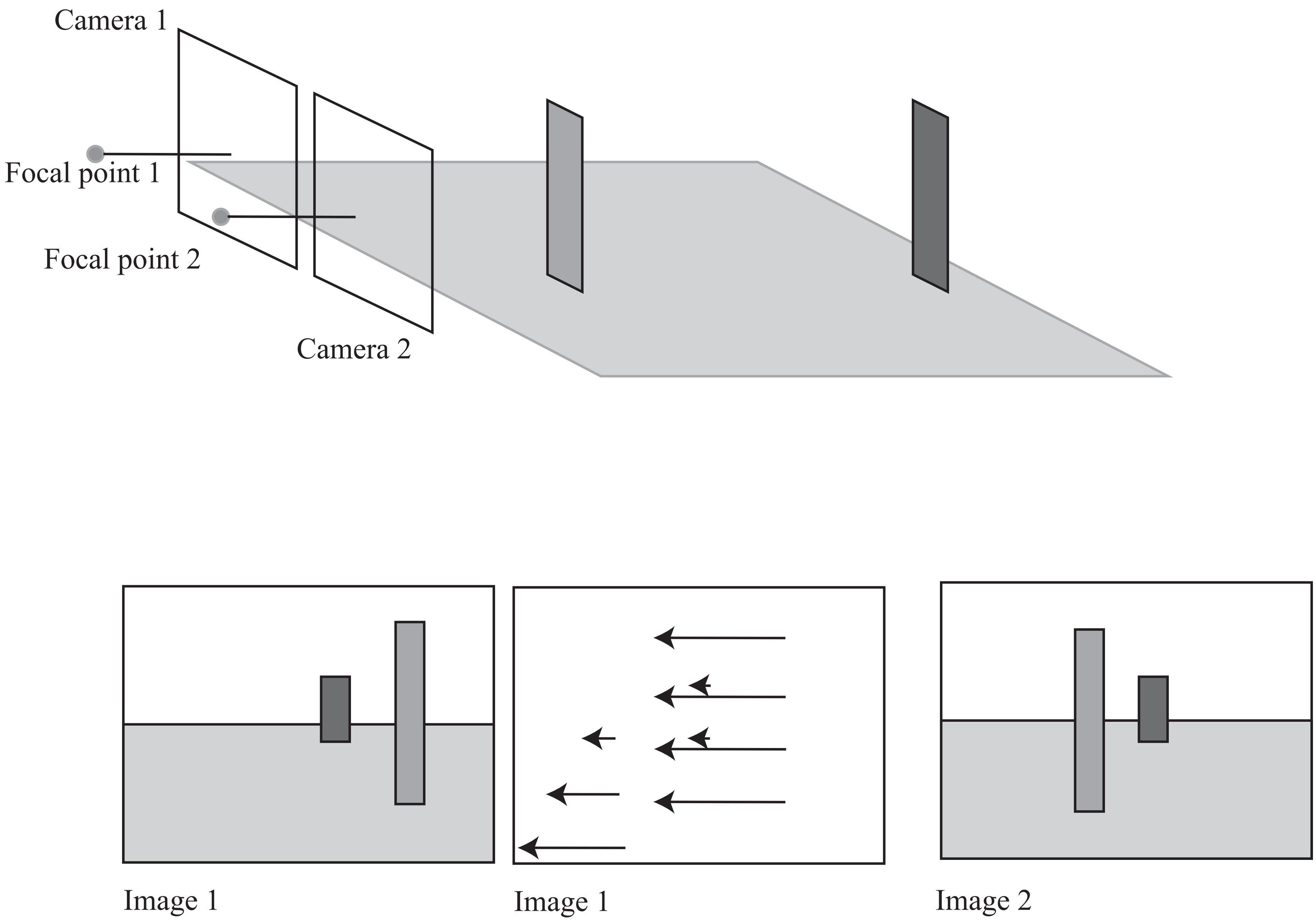


(b)



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!!!


Camera 1
Focal point 1
Focal point 2
Camera 2
Image 1
Image 1
Image 2

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

Image gradients

$$\frac{dI(x,y,t)}{dt} = \frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0$$

Flow (which is unknown)

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

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
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.

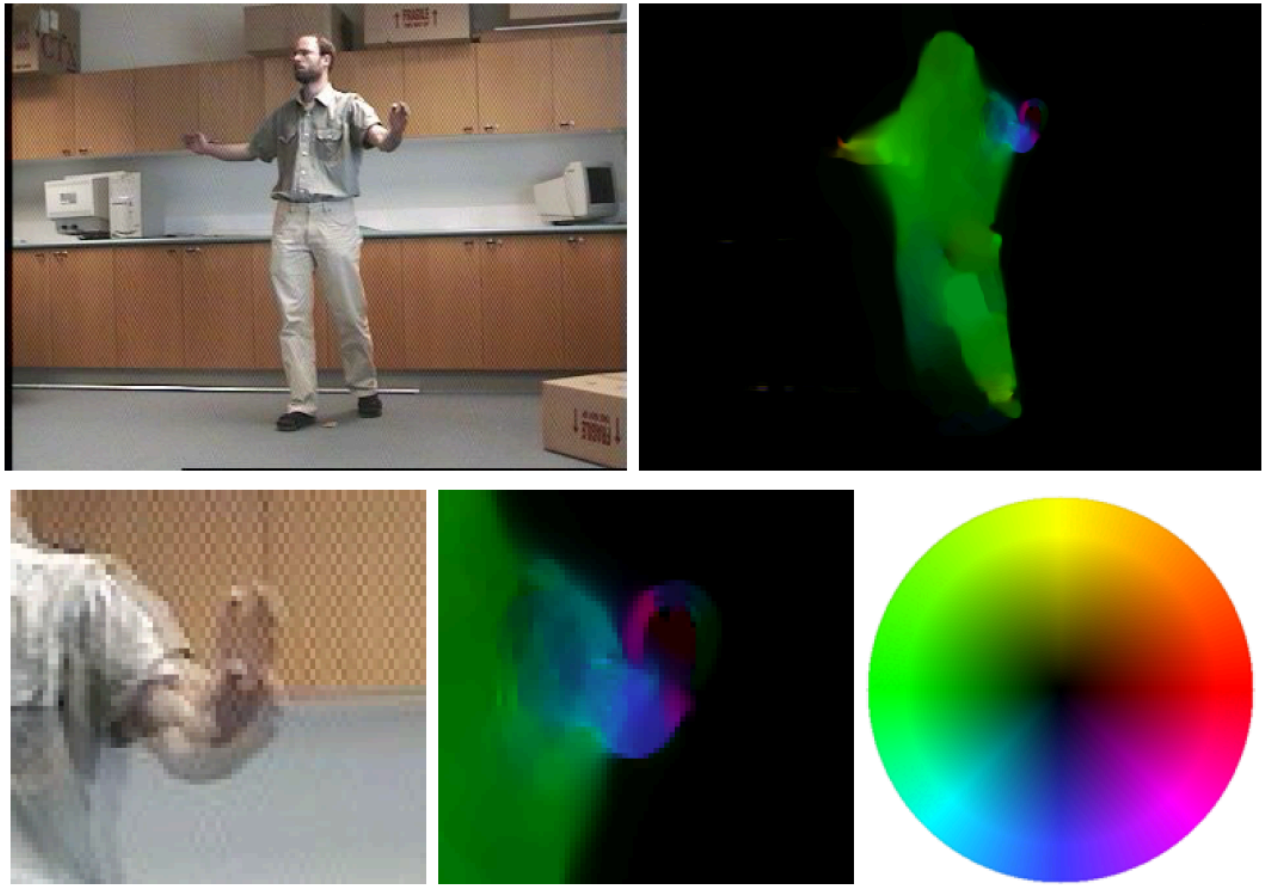


Figure 1. **Top row:** Image of a sequence where the person is stepping forward and moving his hands. The optical flow estimated with the method from [4] is quite accurate for the main body and the legs, but the hands are not accurately captured. **Bottom row, left:** Overlay of two successive frames 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

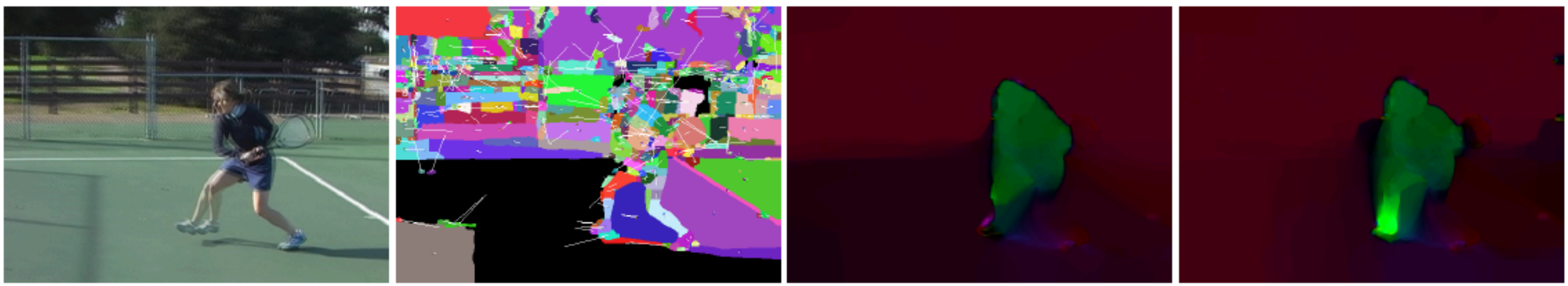


Figure 9. **Left:** Two overlaid images of a tennis player in action. **Center left:** Region correspondences. **Center right:** Result with optical flow from [4]. The motion of the right leg is too fast to be estimated. **Right:** The proposed method captures the motion of the leg.

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/