correspondence and stereopsis
introduction

- disparity:
  - informally: difference between two pictures
  - allows us to gain a sense of depth

- stereopsis:
  - ability to perceive depth from disparity

- goal:
  - design algorithms that mimic stereopsis
stereo vision

- binocular fusion of features observed by the eyes
- reconstruction of their 3D preimage

left  right  perceived depth
stereo vision: easy case

- single point imaged by two calibrated cameras:
  - preimage found at intersection of the rays from the focal points to the image points

[Forsyth & Ponce]
stereo vision: hard/typical case

- many points being observed
- need way to find correspondence

[Forsyth & Ponce]
components of stereo vision systems

- **camera calibration**: previous lecture
- **image rectification**: simplifies search for correspondences
- **correspondence**: which item in left image corresponds to which item in right image?
- **reconstruction**: recovers 3D information from set of 2D correspondences
epipolar geometry

- **epipolar constraint**: corresponding points must lie along conjugate epipolar lines
  - simplifies search to 1D problem

[Trucco & Verri]
image rectification

warp image such that conjugate epipolar lines become collinear and parallel to u axis

[Trucco & Verri]
image rectification (cont.)

- achieved by “rotating the cameras”
- note: this does not = rotating the images
- the lines through the centers become parallel to each other and the epipoles move to infinity
image rectification (cont.)

- given extrinsic parameters $T$ and $R$ (relative position and orientation of two cameras)

  - rotate the left camera about the projection center so that the epipolar lines become parallel to horizontal axis
  - apply same rotation to right camera
  - rotate the right camera by $R$
  - adjust scale in both reference frames
image rectification (cont.)
disparity

- with rectified images, disparity is just (horizontal) displacement of corresponding features in the two images
  - disparity = 0 for distant points
  - larger disparity for closer points
  - depth of point proportional to 1/disparity
“disparity” map
correspondence

- for an element in the left image, find the corresponding element in the right image
- two classes of methods:
  - correlation-based
  - feature-based
correlation-based correspondence

- input: rectified stereo pair and a point \((u,v)\) in the first image

- method:
  - form window of size \((2m+1) \times (2n+1)\) centered at \((u,v)\) and assemble points into the vector \(w\)
  - for each potential match \((u+d,v)\) in the second image compute \(w'\) and the NCC between \(w\) and \(w'\)
sum of squared differences

- recall: SSD for image similarity

$$\psi(u, v) = -(u - v)^2$$

- negative sign: higher values = greater similarity
normalized cross correlation

- normalize to eliminate sensitivity to overall brightness:

$$\psi(u, v) = \frac{(u - \bar{u})(v - \bar{v})}{\sigma_u \sigma_v}$$

where

$$\bar{u} = \text{mean of}(u)$$

$$\sigma_u = \text{standard deviation}(u)$$

- helps for non-diffuse scenes, can hurt for perfectly diffuse scenes
correlation-based correspondence

- main problems:
  - assumes observed surface is locally parallel to the two image planes
  - if not, unequal amount of foreshortening
  - iterate: compute disparities, warp images, repeat

- other problems:
  - not robust against noise
  - similar pixels may not correspond to physical features
feature-based correspondence

- main idea: significant features should be preferred to matches between raw pixel intensities
- instead of correlation-like measures, use similarity between feature descriptors
- typical features: points, lines, and corners
- example: Marr-Poggio-Grimson algorithm
Marr-Poggio-Grimson Algorithm

- convolve image with Laplacian of Gaussian filters with decreasing widths
- find zero crossings of the Laplacian along horizontal scanlines of the filtered images
- for each $\sigma$, match zero crossings with same parity and similar orientations in a $[-w_\sigma \cdots w_\sigma]$ range with $w_\sigma = 2\sqrt{2\sigma}$
Marr-Poggio-Grimson Algorithm

- use disparities found at larger scales to refine search and cause unmatched regions at smaller scales to come into correspondence
Marr-Poggio-Grimson Algorithm

Matching zero-crossings at a single scale

Matching zero-crossings at multiple scales

Scale Width Match Offset

Rematch
Marr-Poggio-Grimson Algorithm
ordering constraint

- order of matching features usually the same in both images
- but not always: occlusion

[Forsyth & Ponce]
dynamic programming

- treat feature correspondence as a graph problem

Left image features

Right image features

Cost of edges = similarity of regions between image features
dynamic programming

- find min-cost path through the graph

Cost of edges = similarity of regions between image features
dynamic programming
dynamic programming

[Ohta & Kanade]
reconstruction

- given pair of image points \( p \) and \( p' \), and focal points \( O \) and \( O' \), find preimage \( P \)
- in theory: find \( P \) by intersecting the rays \( R=Op \) and \( R'=O'p' \)
- in practice: \( R \) and \( R' \) won’t actually intersect due to calibration and feature localization errors
reconstruction techniques

- geometric: construct the line segment perpendicular to R and R’ that intersects both rays and take its mid-point

[Forysth & Ponce]
reconstruction techniques

- image-space: find the point \( P \) whose projection onto the images minimizes distance to desired correspondences
- nonlinear optimization