B. Julesz: Showed that correspondence is not needed for stereo.

**Correspondence: Epipolar constraint.**

Corresponding point has to lie on the epipolar line
Two classes of algorithms:
- Correlation-based algorithms
  - Produce a DENSE set of correspondences
- Feature-based algorithms
  - Produce a SPARSE set of correspondences

Correspondence: Photometric constraint
- Same world point has same intensity in both images.
  - Lambertian fronto-parallel
- Issues:
  - Noise
  - Specularity
  - Foreshortening
Stereo Matching

For each epipolar line
  For each pixel in the left image
    • compare with every pixel on same epipolar line in right image
    • pick pixel with minimum match cost
    • This will never work, so:

Improvement: match *windows*

Comparing Windows:

\[
SSD = \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2 \\
C_{fg} = \sum_{[i,j] \in R} f(i,j)g(i,j)
\]

Most popular

For each window, match to closest window on epipolar line in other image.
Comparing Windows:

Minimize \[ \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2 \] Sum of Squared Differences

Maximize \[ C_{fg} = \sum_{[i,j] \in R} f(i,j)g(i,j) \] Cross correlation

It is closely related to the SSD:

\[ SSD = \sum_{[i,j] \in R} (f - g)^2 = \]

\[ = \sum_{[i,j] \in R} f^2 + \sum_{[i,j] \in R} g^2 - 2 \sum_{[i,j] \in R} fg \]

Window size

- Effect of window size
  - Better results with adaptive window

<table>
<thead>
<tr>
<th>W = 3</th>
<th>W = 20</th>
</tr>
</thead>
</table>

(S. Seitz)
Stereo results

- Data from University of Tsukuba

Results with window correlation

Window-based matching (best window size)

Ground truth (Seitz)
Results with better method

State of the art

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts,
International Conference on Computer Vision, September 1999.

Ground truth

(Seitz)

Ordering constraint

- Usually, order of points in two images is same.
- Is this always true?
- If we match pixel i in image 1 to pixel j in image 2, no matches that follow will affect which are the best preceding matches.
- *Example with pixels (a la Cox et al.).*
The Ordering Constraint

Points on the epipolar lines appear in the same order

But it is not always the case ... This enables dynamic programming

Dynamic Programming (Baker and Binford, 1981)

Find the minimum-cost path going monotonically down and right from the top-left corner of the graph to its bottom-right corner.

- Nodes = matched feature points (e.g., edge points).
- Arcs = matched intervals along the epipolar lines.
- Arc cost = discrepancy between intervals.
Dynamic Programming (Baker and Binford, 1981)

Find the minimum-cost path going monotonically down and right from the top-left corner of the graph to its bottom-right corner.

• Nodes = matched feature points (e.g., edge points).
• Arcs = matched intervals along the epipolar lines.
• Arc cost = discrepancy between intervals.


Dynamic Programming (Ohta and Kanade, 1985)
Other constraints

- Smoothness: disparity usually doesn’t change too quickly.
  - Unfortunately, this makes the problem 2D again.
  - Solved with a host of graph algorithms, Markov Random Fields, Belief Propagation, ....
- Uniqueness constraint (each feature can at most have one match)
- Occlusion and disparity are connected.

Feature-based Methods

- Conceptually very similar to Correlation-based methods, but:
  - They only search for correspondences of a sparse set of image features.
  - Correspondences are given by the most similar feature pairs.
  - Similarity measure must be adapted to the type of feature used.
Feature-based Methods:

- Features most commonly used:
  - Corners
    - Similarity measured in terms of:
      - surrounding gray values (SSD, Cross-correlation)
      - location
  - Edges, Lines
    - Similarity measured in terms of:
      - orientation
      - contrast
      - coordinates of edge or line’s midpoint
      - length of line

Example: Comparing lines

- $l_l$ and $l_r$: line lengths
- $\theta_l$ and $\theta_r$: line orientations
- $(x_{l_l}, y_{l_l})$ and $(x_{r_r}, y_{r_r})$: midpoints
- $c_l$ and $c_r$: average contrast along lines
- $\omega_l$, $\omega_\theta$, $\omega_m$, $\omega_c$: weights controlling influence

$$S = \frac{1}{\omega_l(l_l - l_r)^2 + \omega_\theta(\theta_l - \theta_r)^2 + \omega_m[(x_l - x_r)^2 + (y_l - y_r)^2] + \omega_c(c_l - c_r)^2}$$

The more similar the lines, the larger $S$ is!