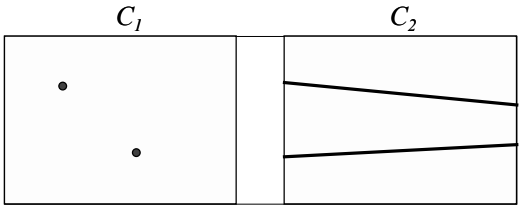
## Stereo matching

- "Stereo matching" is the correspondence problem
  - For a point in Image #1, where is the corresponding point in Image #2?

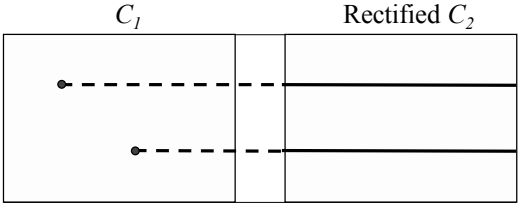



Epipolar lines

- Image rectification makes the correspondence problem easier
  - And reduces computation time

## Stereo matching

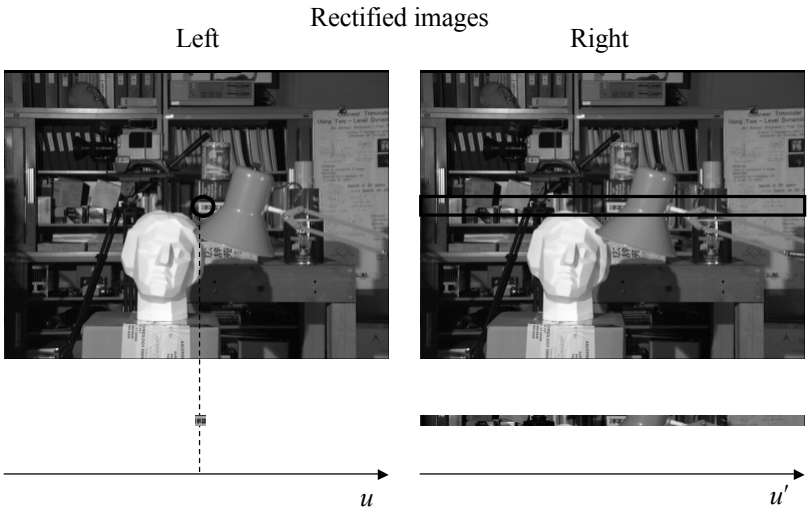
- "Stereo matching" is the correspondence problem
  - For a point in Image #1, where is the corresponding point in| Image #2?



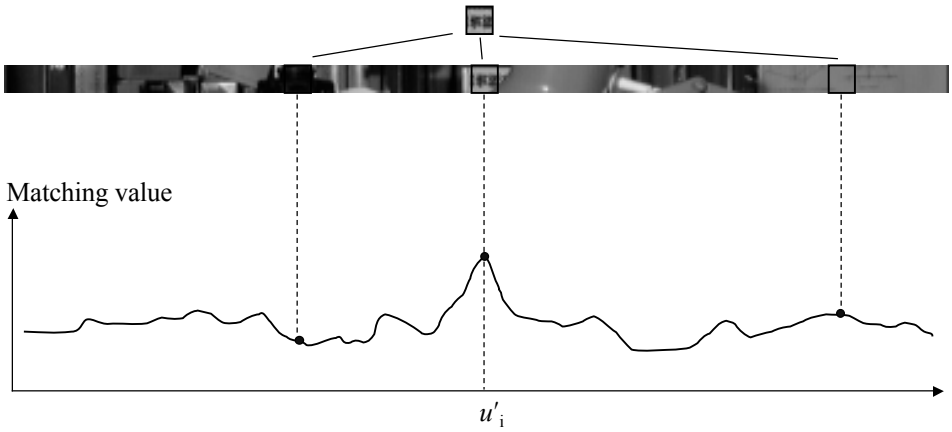

Epipolar lines

- Image rectification makes the correspondence problem easier
  - And reduces computation time

# Stereo matching




# Matching along epipolar line




The best match estimates the “disparity”

- In this case, horizontal disparity only (since images were rectified)

## Area matching

- Correlation
  - Correlate left image patch along the epipolar line in the right image
  - Best match = highest value
    - Normalized correlation would be better!

- Sum of Squared Differences (SSD)
  - Better than correlation, faster than normalized correlation

$$SSD(u,v) = \sum_{\substack{\text{area} \\ \text{around} \\ (u,\,v)}} (I_{left}(i,j) - I_{right}(i,j))^2$$

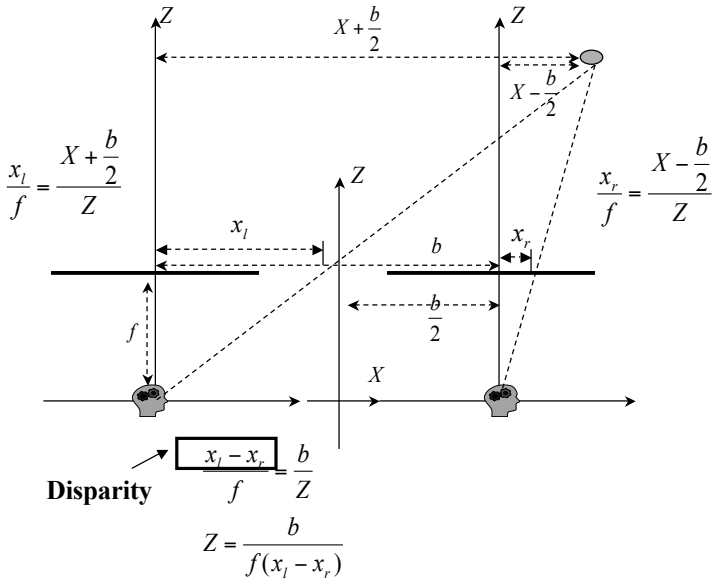
  - Best match = lowest value

## Stereo matching algorithms

- There are many!
  - Edge based
  - Coarse-to-fine
  - Adaptive windows
  - Dynamic programming
  - Markov random fields, graph cuts
  - Multi-baseline
  - Etc.

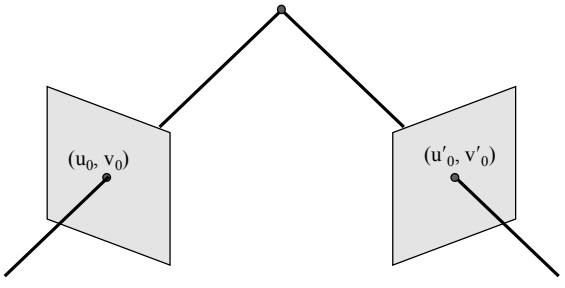
- Pitfalls
  - Specularities
  - Occlusions (missing data)
  - Sensor noise
  - Calibration error
  - Matching ambiguity (constant or low-constrast regions)
  - Etc.

# Basic Stereo Configuration: rectified images




# Stereo disparity

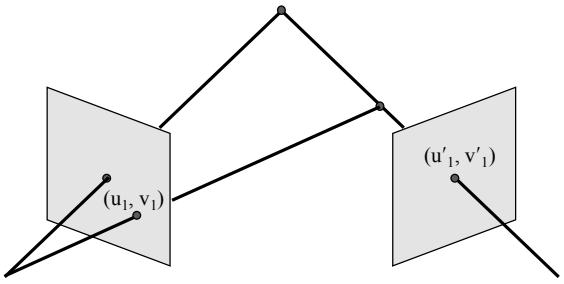
- "Stereo disparity" is the difference in position between correspondence points in two images
  - Disparity is inversely proportional to scene depth




Disparity: $(du_0, dv_0) = (u_0 - u'_0, v_0 - v'_0) = (0, 0)$

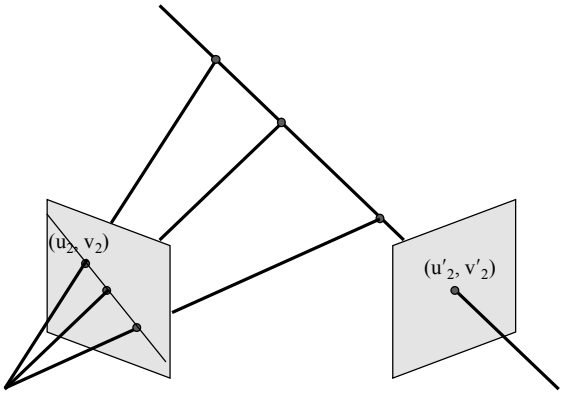
***Disparity is a vector!***

# Stereo disparity




Disparity: $(du_1, dv_1) = (u_1 - u'_1, v_1 - v'_1)$

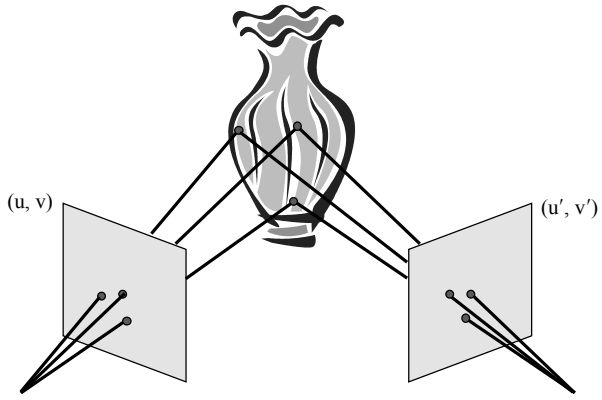
# Stereo disparity




Disparity: $(du_2, dv_2) = (u_2 - u'_2, v_2 - v'_2)$

# Stereo disparity

**Depth = *f* (disparity, geometry)**

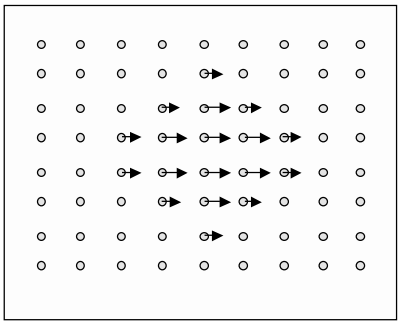


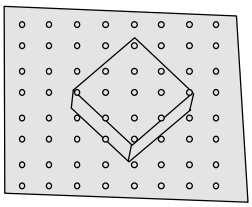

Disparities: $(du_i, dv_i) = (u_i - u'_i, v_i - v'_i)$

# Output of stereo matching

- Dense stereo
  - Disparity at each point
- Sparse stereo
  - Disparity at each feature point

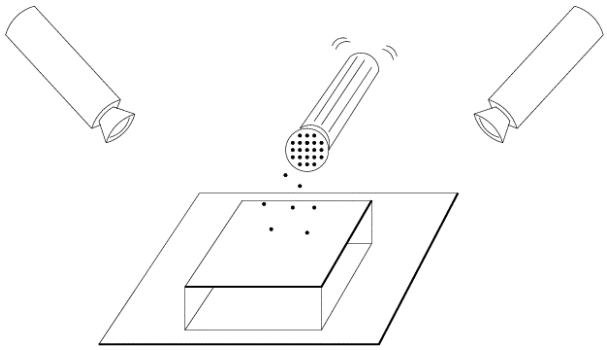
**Depth = *f* (disparity, geometry)**



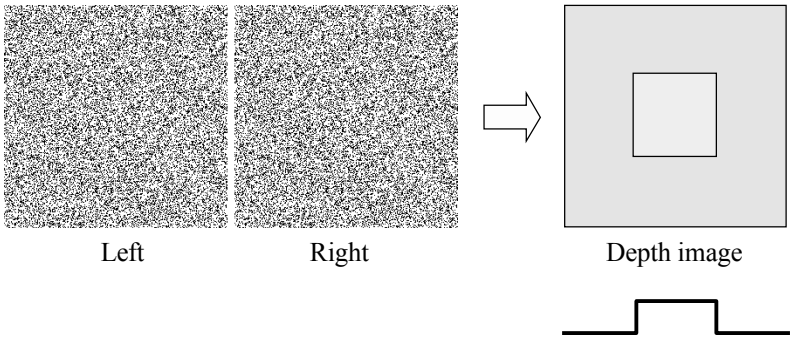


# Random dot stereograms

- Correspondence is not always required in order to see depth
- Existence proof:  random-dot stereograms



# RDS example



Left Right Depth image

How is this possible with completely random correspondence?

Marr -Poggio cooperative stereo algorithm

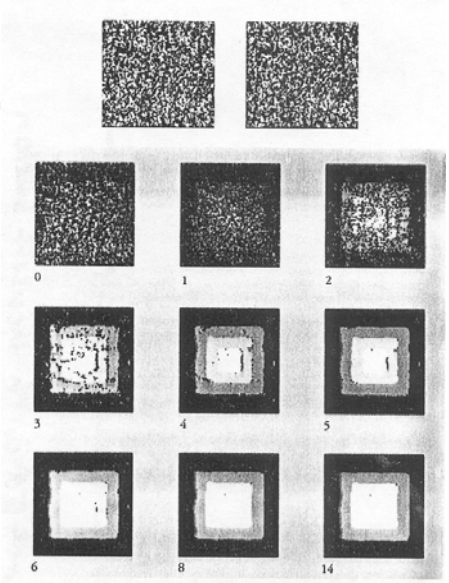
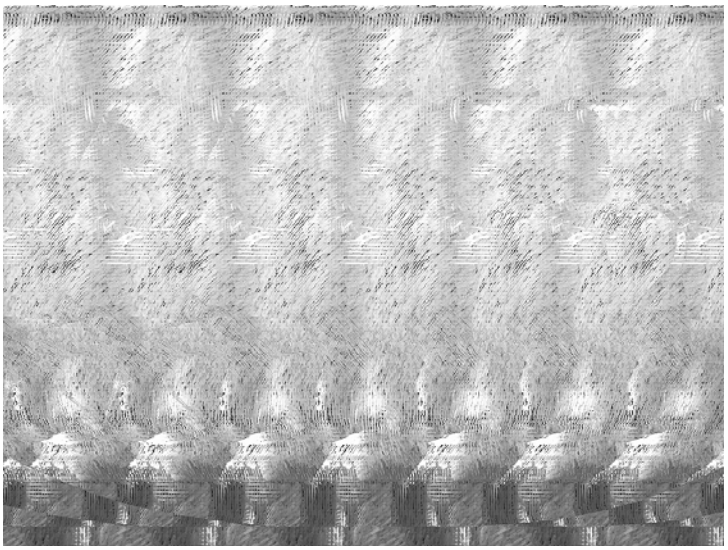
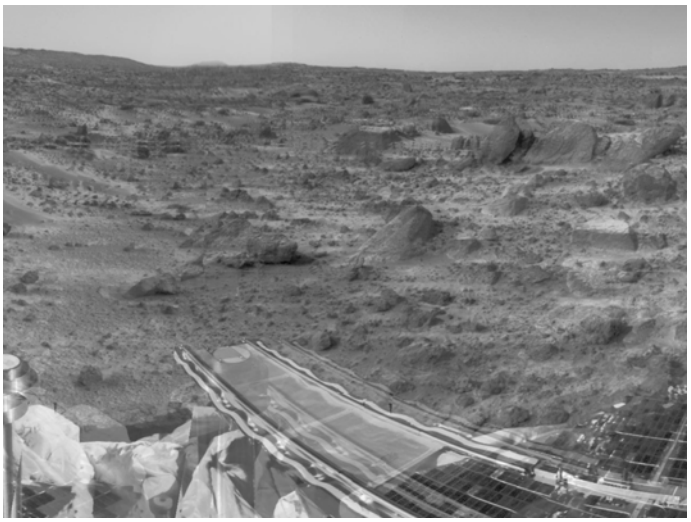



*Figure 3–7.* The decoding of a random-dot stereogram pair by the cooperative algorithm described in the text. The stereogram appears at the top, and the initial state of the network, which includes all possible matches within the prescribed disparity range, is labeled 0. The algorithm runs through a number of iterations, as shown, and gradually the structure is revealed. The different shades of gray represent different disparity values.

## Single image stereograms: should try this!



## Red/Green stereo display



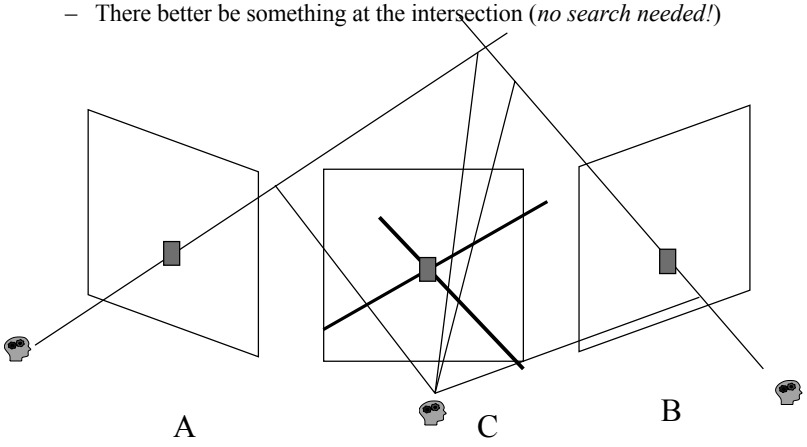
From Mars Pathfinder

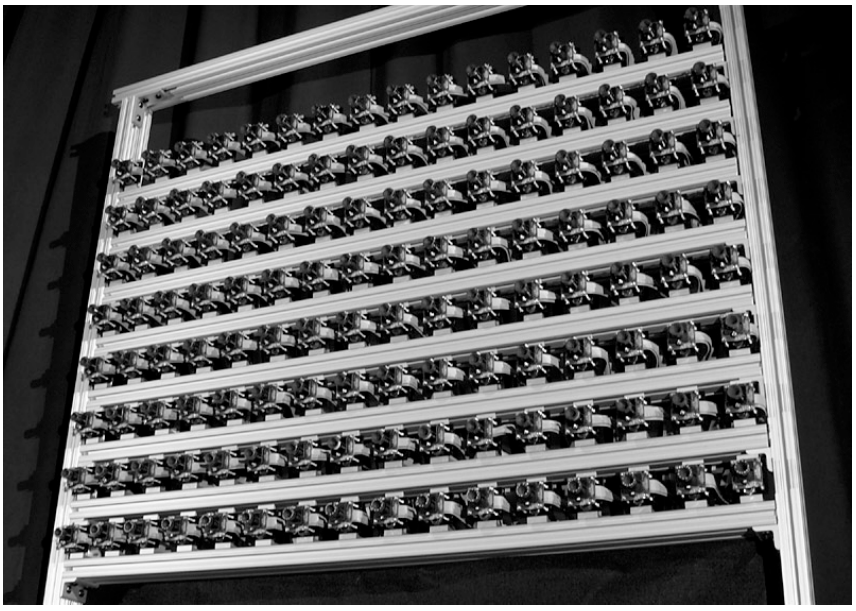
## Multiple camera stereo

- Using multiple camera in stereo has advantages and disadvantages
- Some disadvantages
  - Computationally more expensive
  - More correspondence matching issues
  - More hardware ($)
- Some advantages
  - Extra view(s) reduces ambiguity in matching
  - Wider range of view, fewer “holes”
  - Better noise properties
  - Increased depth precision

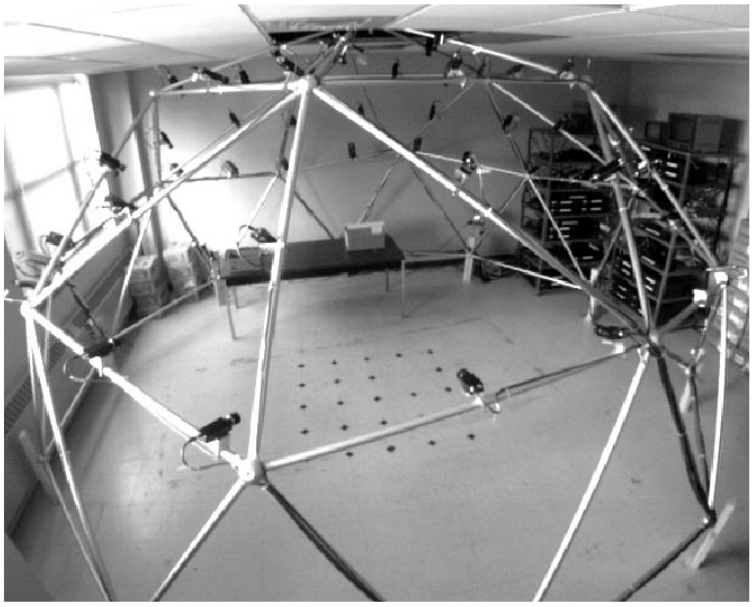
# Three Camera Stereo

- A powerful way of eliminate spurious matches
  - Hypothesize matches between A & B
  - Matches between A & C on green epipolar line
  - Matches between B & C on red epipolar line
  - There better be something at the intersection (*no search needed!*)



**The Stanford Multi-Camera Array**
128 CMOS cameras, 2” baseline



**CMU multi-camera stereo**
51 video cameras mounted on a 5-meter diameter geodesic dome

## Stereo: Summary

- Multiview geometry
  - Epipolar geometry
- Correspondence problem
- Essential Matrix and Fundamental Matrix
- Random dot stereograms