

Energy minimization

- Another global approach to improve quality of correspondences
- Assumption: disparities vary (mostly) smoothly
- Minimize energy function:

$$E_{\text{data}} + \lambda E_{\text{smoothness}}$$

- E_{data} : how well does disparity match data
- $E_{\text{smoothness}}$: how well does disparity match that of neighbors – regularization

Stereo as energy minimization

- Matching Cost Formulated as Energy

- “data” term penalizing bad matches

$$D(x, y, d) = |\mathbf{I}(x, y) - \mathbf{J}(x + d, y)|$$

- “neighborhood term” encouraging spatial smoothness

$V(d_1, d_2)$ = cost of adjacent pixels with labels d_1 and d_2

$$= |d_1 - d_2| \quad (\text{or something similar})$$

$$E(\{d\}) = \sum_{(x,y)} D(x, y, d_{x,y}) + \sum_{\text{neighbors } (x_1,y_1),(x_2,y_2)} V(d_{x_1,y_1}, d_{x_2,y_2})$$

Energy minimization

$$E(\{d\}) = \sum_{(x,y)} D(x, y, d_{x,y}) + \sum_{\text{neighbors } (x_1,y_1),(x_2,y_2)} V(d_{x_1,y_1}, d_{x_2,y_2})$$

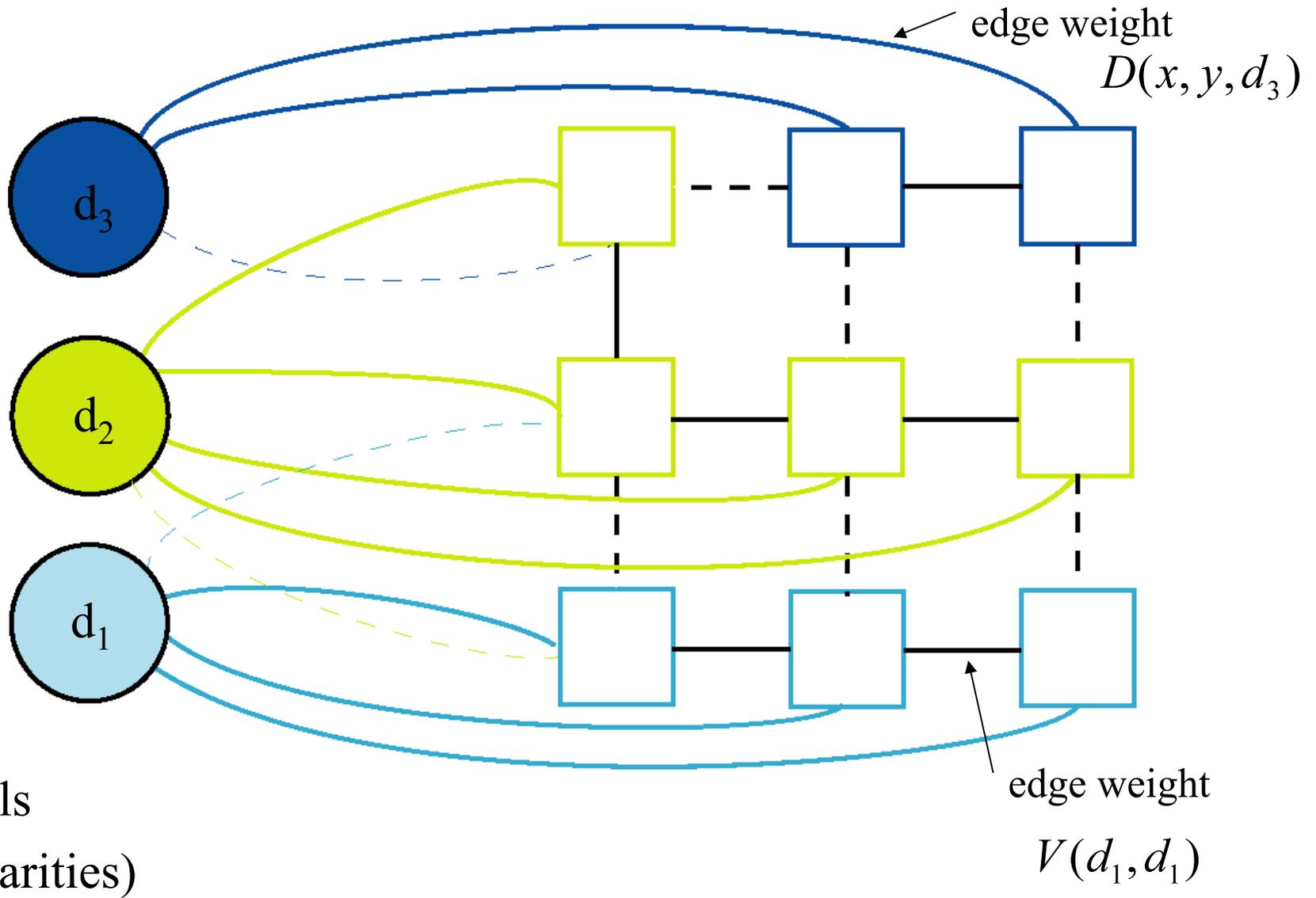
- Many local minimum
 - Why?
 - Gradient descent doesn't work well
- In practice, disparities only piecewise smooth
- Design smoothness function that doesn't penalize large jumps too much
 - Example: $V(\alpha, \beta) = \min(|\alpha - \beta|, K)$
 - Non-convex



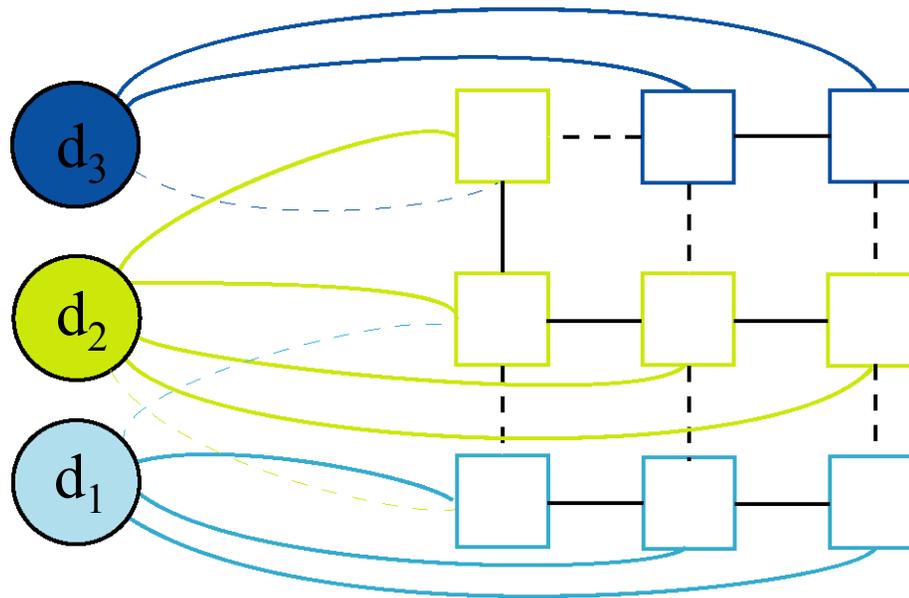
Energy minimization

- Hard to find global minima of non-smooth functions
 - Many local minima
 - Provably NP-hard
- Practical algorithms look for approximate minima (e.g., simulated annealing)

Energy minimization via graph cuts

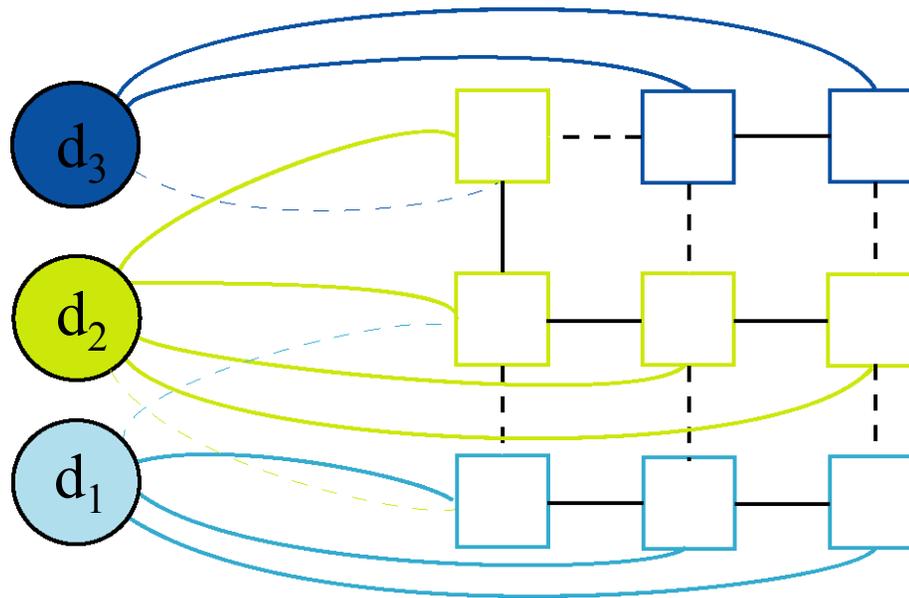


Energy minimization via graph cuts



- Graph Cost
 - Matching cost between images
 - Neighborhood matching term
 - Goal: figure out which labels are connected to which pixels

Energy minimization via graph cuts



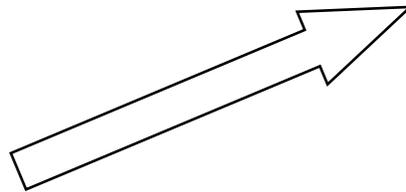
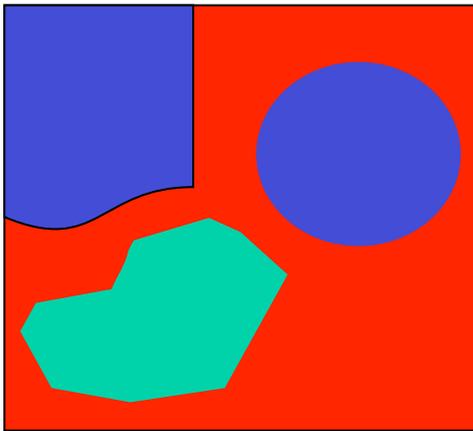
- Graph Cut
 - Delete enough edges so that
 - each pixel is connected to exactly one label node
 - Cost of a cut: sum of deleted edge weights
 - Finding min cost cut equivalent to finding global minimum of energy function

Computing a multiway cut

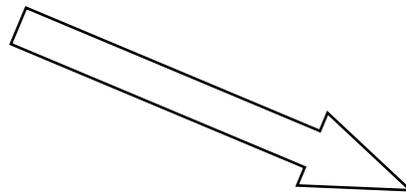
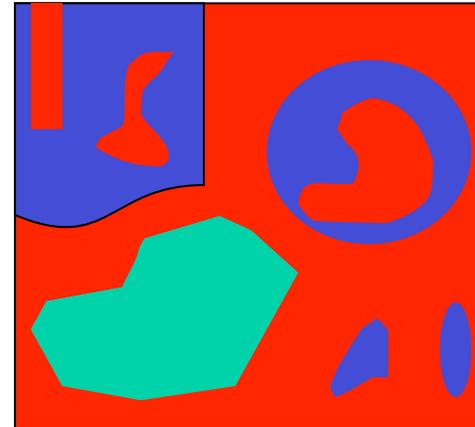
- With 2 labels: classical min-cut problem
 - Solvable by standard flow algorithms
 - polynomial time in theory, nearly linear in practice
 - More than 2 terminals: NP-hard
 - [Dahlhaus *et al.*, STOC '92]
- Efficient approximation algorithms exist
 - Yuri Boykov, Olga Veksler and Ramin Zabih,
[Fast Approximate Energy Minimization via Graph Cuts](#),
International Conference on Computer Vision, September 1999.
 - Within a factor of 2 of optimal
 - Computes local minimum in a strong sense
 - even very large moves will not improve the energy

Move examples

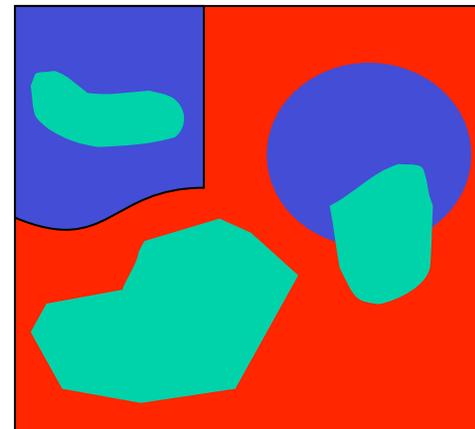
Starting point



Red-blue swap move

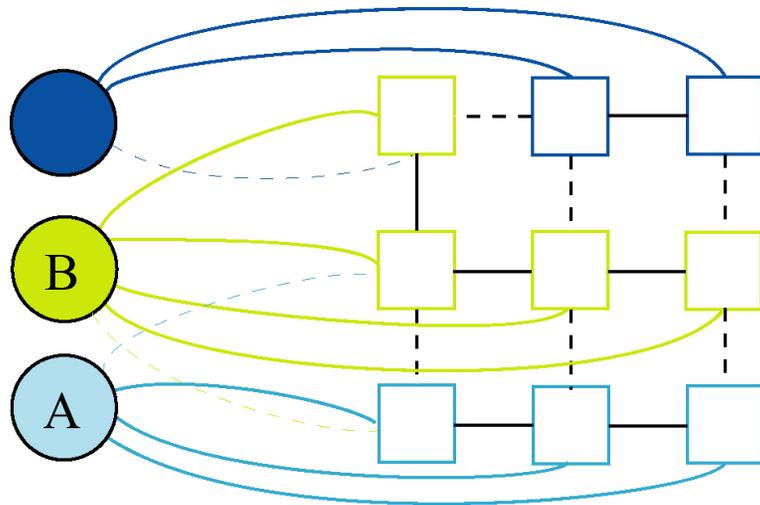


Green expansion move

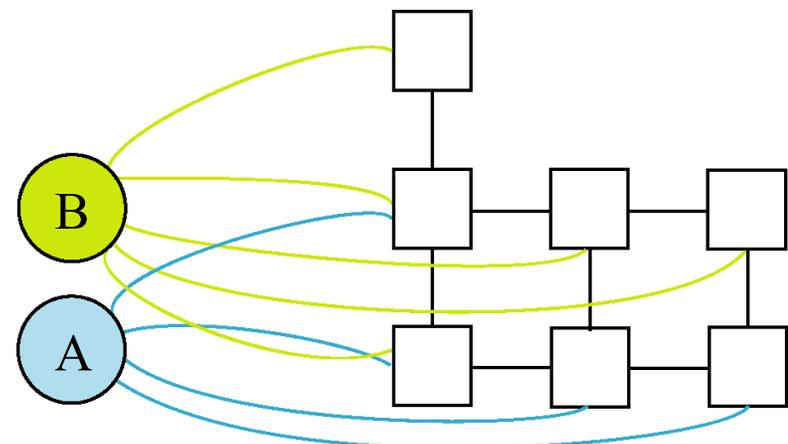


The swap move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label pair (A,B) in some order
 - 2.1 Find the lowest E labeling within a single AB -swap
 - 2.2 Go there if it's lower E than the current labeling
3. If E did not decrease in the cycle, we're done
Otherwise, go to step 2



Original graph



AB subgraph
(run min-cut on this graph)

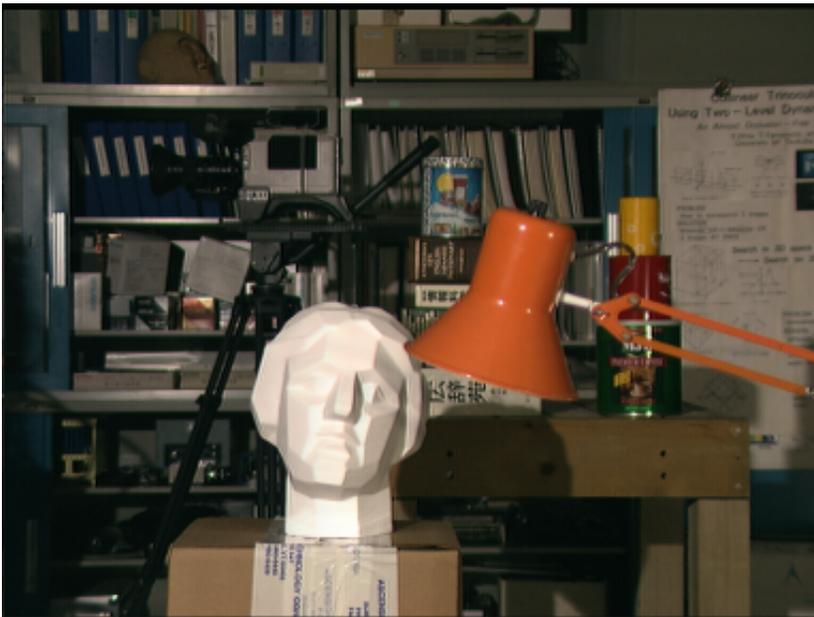
The expansion move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label A in some order
 - 2.1 Find the lowest E labeling within a single A -expansion
 - 2.2 Go there if it's lower E than the current labeling
3. If E did not decrease in the cycle, we're done Otherwise, go to step 2

Multi-way cut \rightarrow A sequence of binary optimization problems

Stereo results

- Data from University of Tsukuba



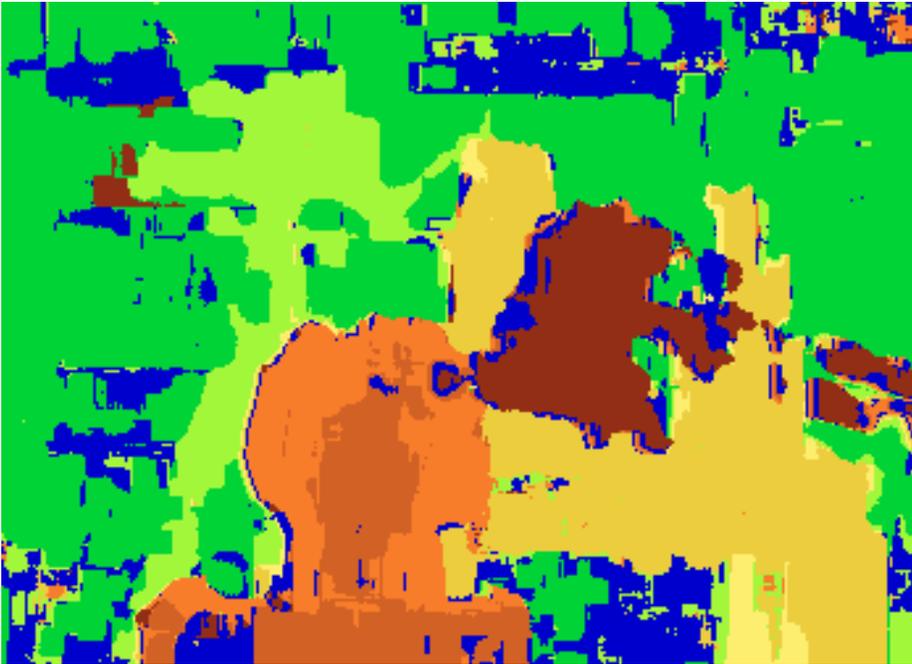
scene



ground truth

<http://cat.middlebury.edu/stereo/>

Results with window correlation



normalized correlation
(best window size)



ground truth

Results with graph cuts



graph cuts
(Potts model E ,
expansion move algorithm)



ground truth

Results with graph cuts

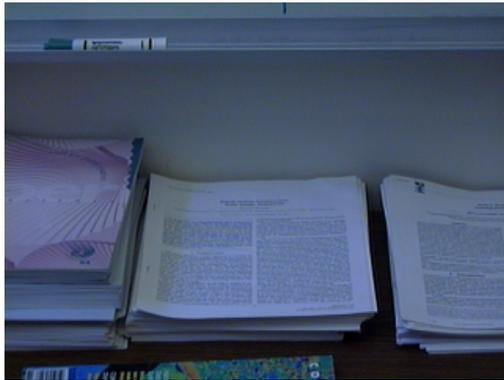


graph cuts
(Potts model E ,
expansion move algorithm)



ground truth

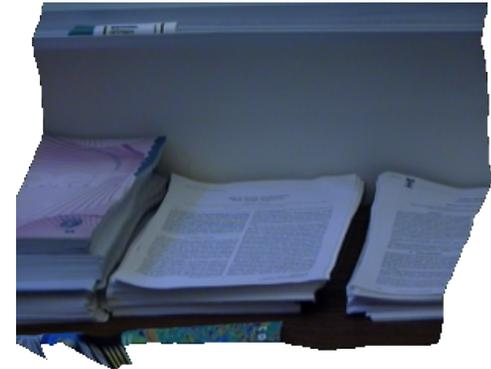
Depth from disparity



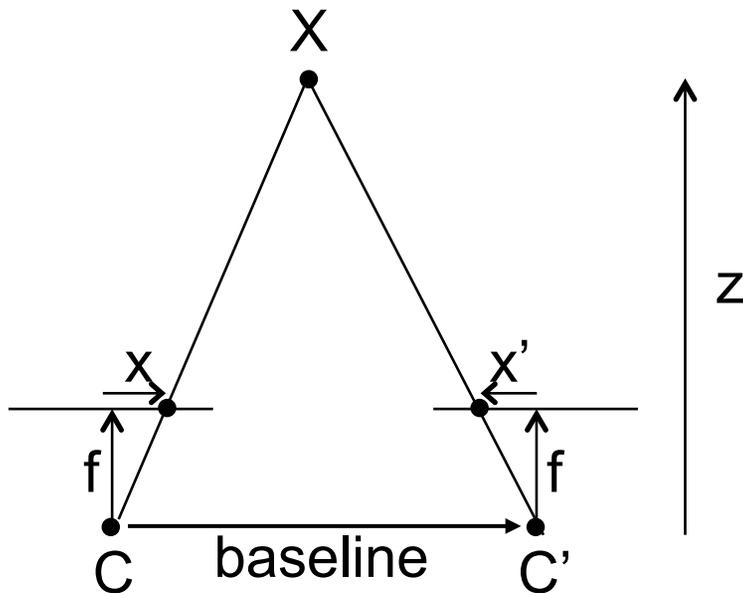
input image (1 of 2)



depth map
[Szeliski & Kang '95]



3D rendering



$$disparity = x - x' = \frac{baseline * f}{z}$$

Real-time stereo



[Nomad robot](http://www.frc.ri.cmu.edu/projects/meteorobot/index.html) searches for meteorites in Antarctica
<http://www.frc.ri.cmu.edu/projects/meteorobot/index.html>

- Used for robot navigation (and other tasks)
 - Several software-based real-time stereo techniques have been developed (most based on simple discrete search)

Stereo reconstruction pipeline

- Steps
 - Calibrate cameras
 - Rectify images
 - Compute disparity
 - Estimate depth

What will cause errors?

- Camera calibration errors
- Poor image resolution
- Occlusions
- Violations of brightness constancy (specular reflections)
- Large motions
- Low-contrast image regions