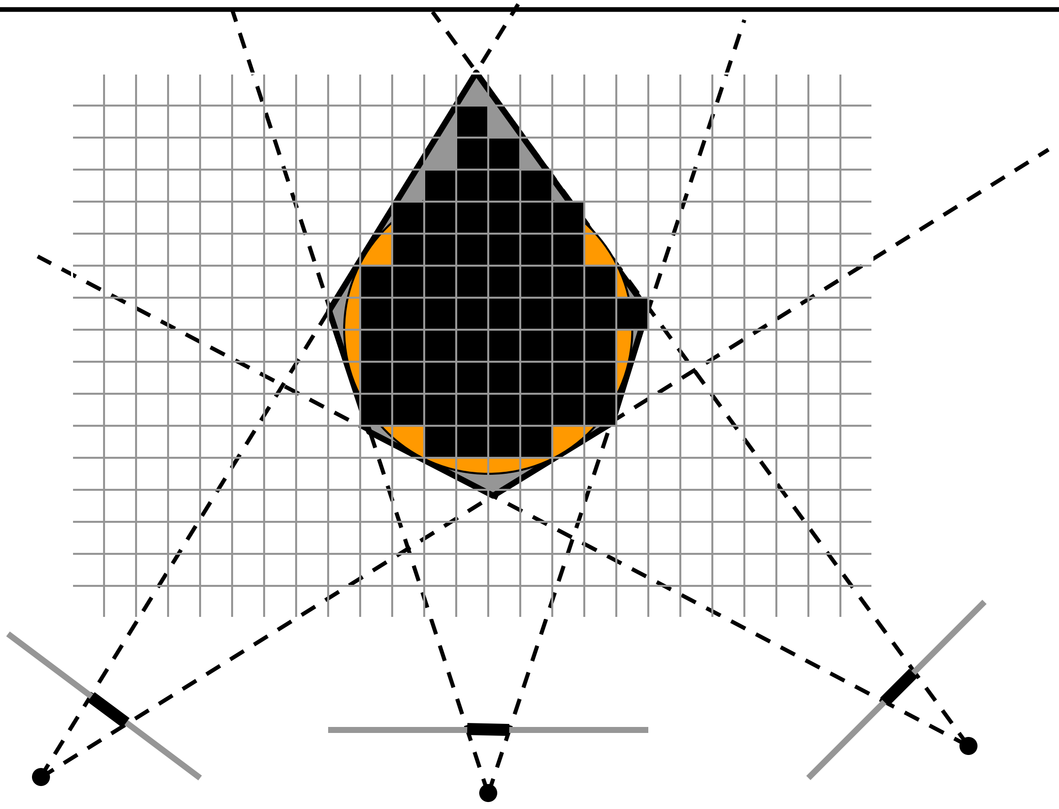
Exact Voxel Occupancy with Graph Cuts

Dan Snow, Paul Viola and Ramin Zabih

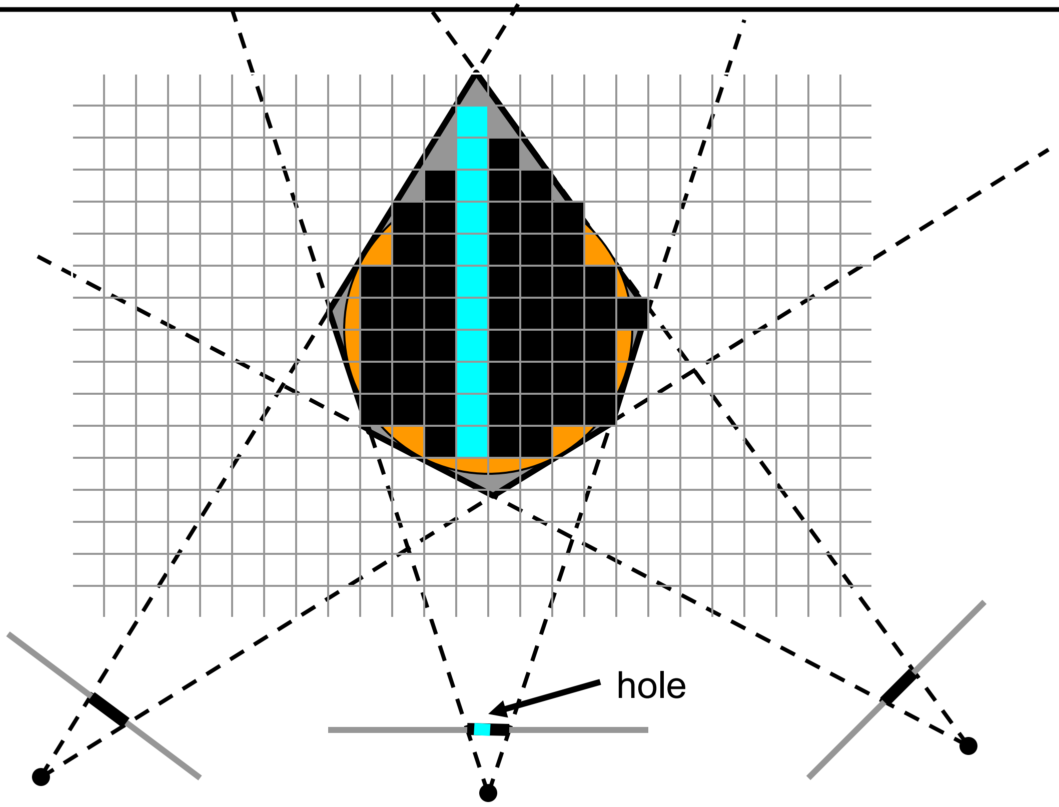
CVPR 2000

Volume (silhouette) intersection



Color voxel black if on silhouette in every image

Volume (silhouette) intersection



Color voxel black if on silhouette in every image

Issues

Early hard decisions are a problem

- Errors in silhouette identification
- Can lead to “holes”

Can we add spatial smoothness with energy minimization?

Can we solve it fast?

Problem statement

Given voxels $p \in P$, assigned binary labels f_p

- filled (1) or empty (0)

Each pixel corresponds to a line of voxels

- If pixel is different from background, these voxels prefer 1, else 0

Energy function

$D_p(f_p)$ = cost for voxel p to have label f_p

D_p(empty) is large if voxel is different from background

D_p(full) = constant

We seek the labeling f that minimizes

$$E(f) = \sum_p D_p(f_p) + E_{smooth}(f)$$

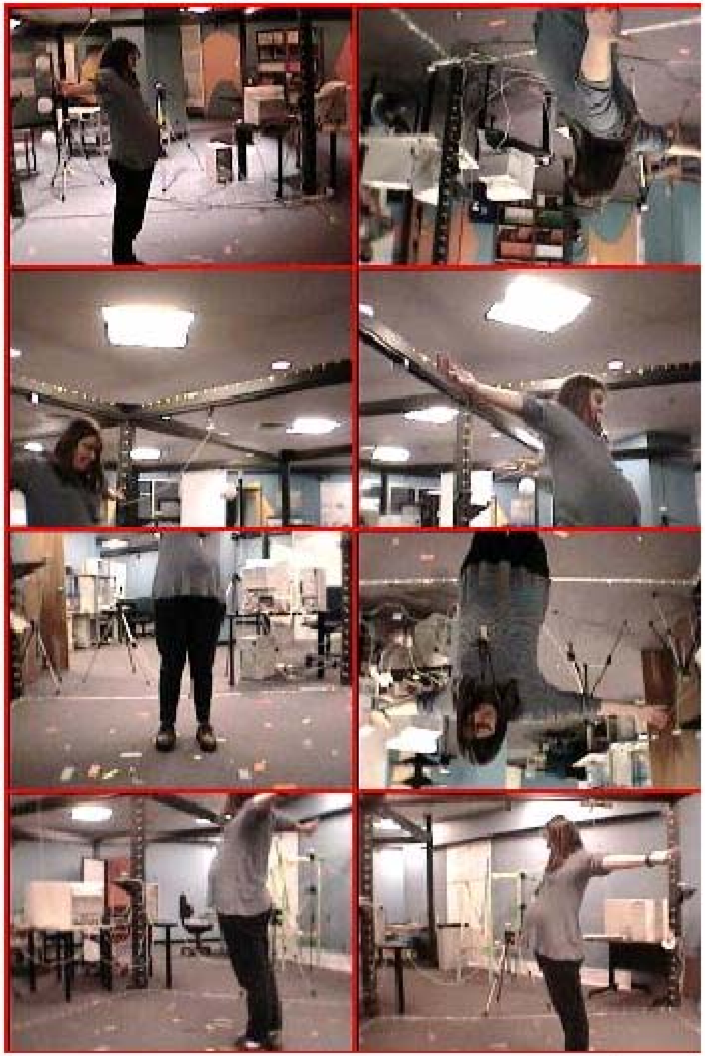
What smoothness term?

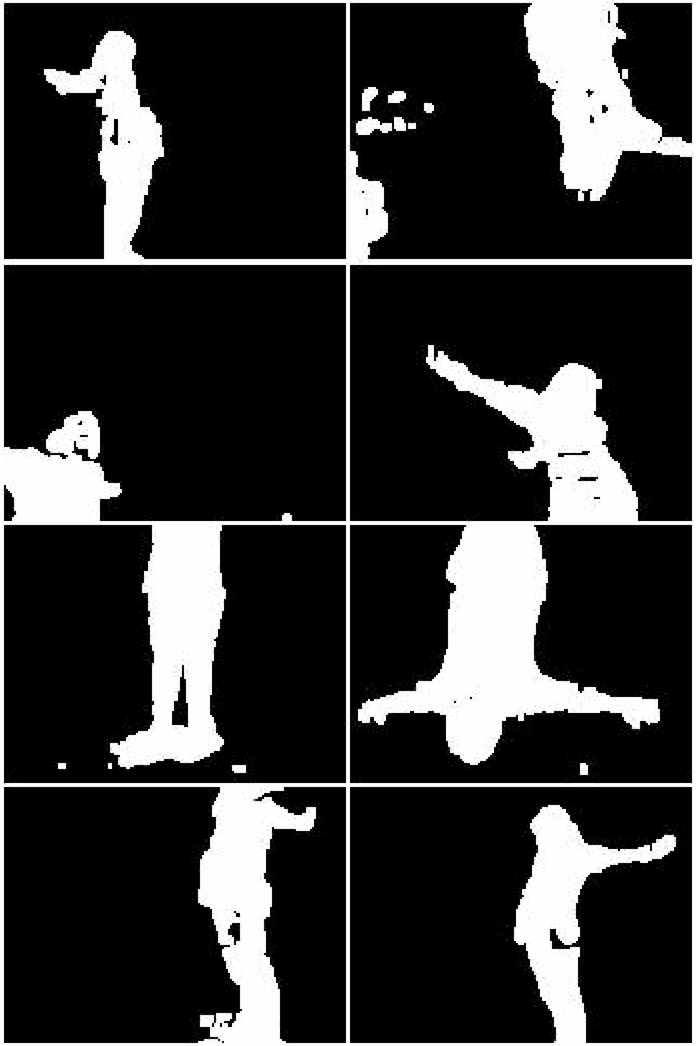
Here is a natural one where we can efficiently find the global minimum!

$$E_{smooth}(f) = \sum_{p,q \in N} V(f_p, f_q)$$

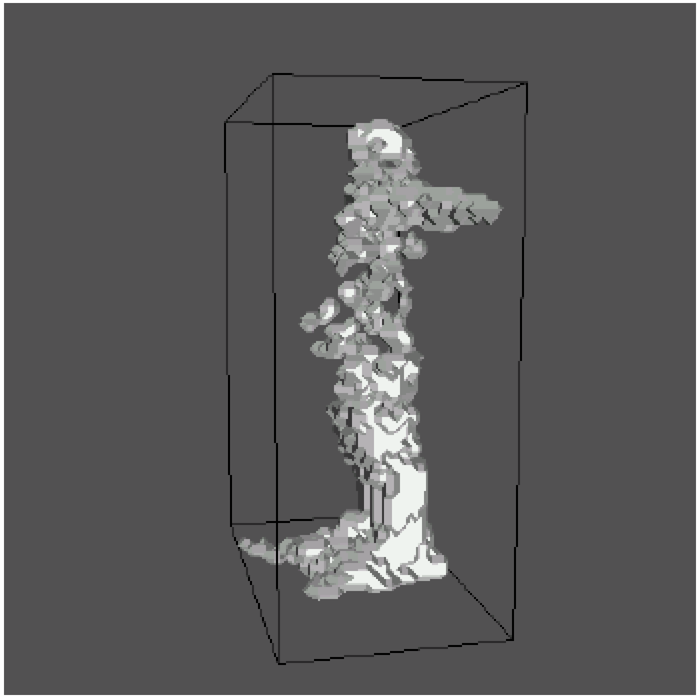
$$V(f_p, f_q) = T[f_p = f_q]$$

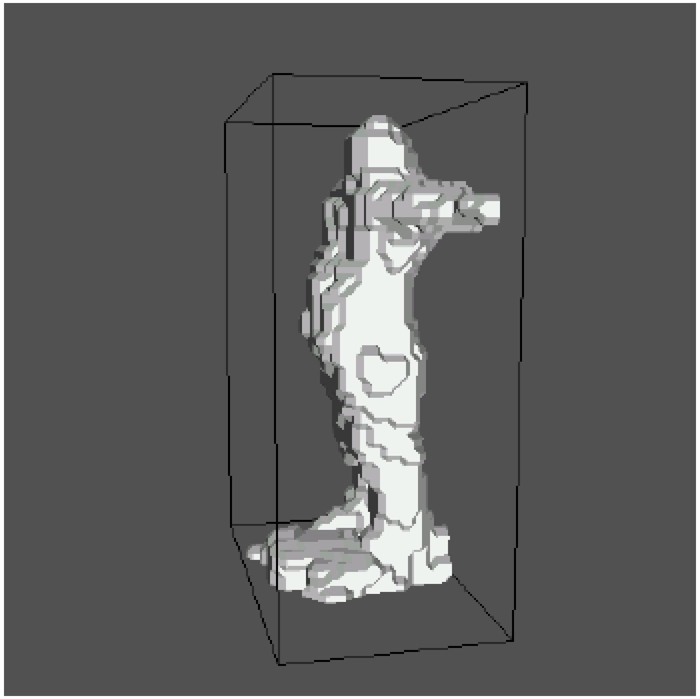
Result





Result





Running time

For 100x100x100 grid, 16 cameras

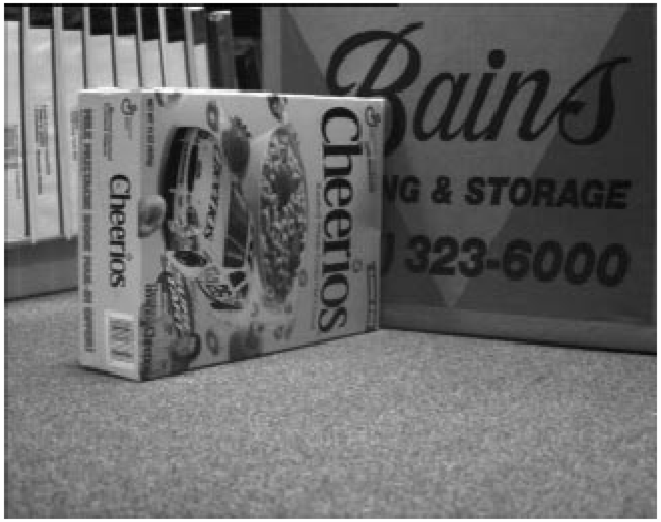
- 9 seconds total, on 500Mhz Intel PIII
  - > 7 sec: graph prep + project
  - > 1 sec: max flow computation
  - > 1 sec: min-cut labeling of voxels from max-flow

Multiway Cut for Stereo and Motion with Slanted Surfaces

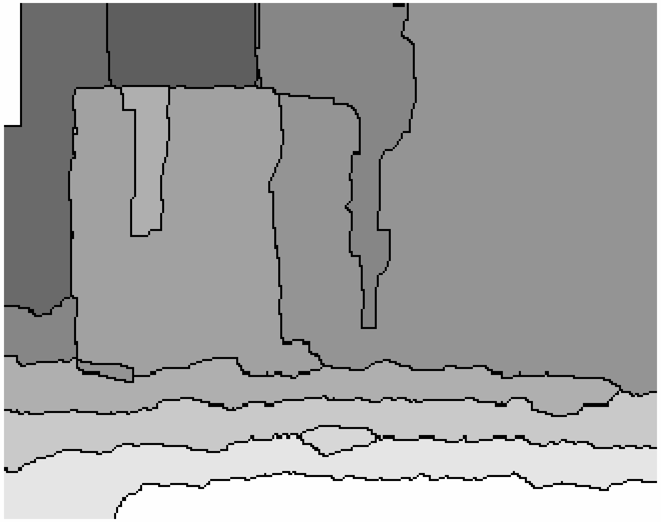
Stan Birchfield and Carlo Tomasi

ICCV 1999

Motivation



an image from a stereo pair

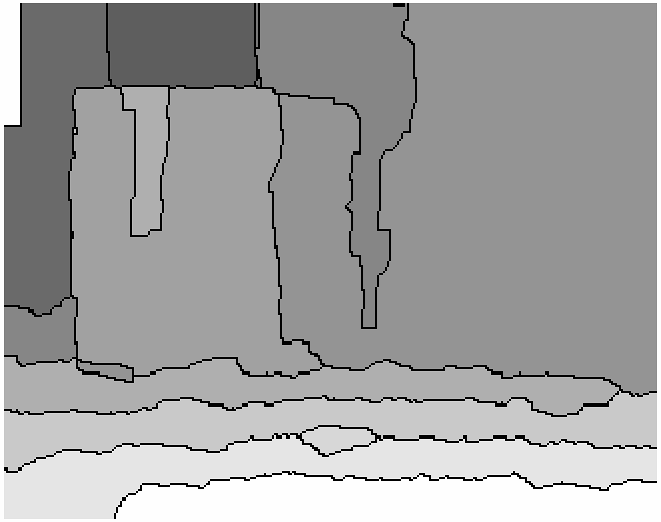


disparity map from graph cuts

Why does it look so bad?

Solution

an image from a stereo pair



disparity map from graph cuts

Think of this as a segmentation

- Fit plane to each region to give more accurate results
- Once you have these planes, reassign pixels to get better fit

Algorithm

1. Initialize a set of pixel labels
2. For each label, fit a plane to all pixels with that label
   - they do this by solving for an affine transformation (Lukas-Kanade) that best aligns region in left image to corresponding region in right image
     - > this is appropriate under orthographic projection
3. Assign labels (planes) to pixels
   - use graph cuts of course!
4. Repeat Steps 2 & 3 until convergence
5. Cleanup phase
   - merge neighboring planes

This style of algorithm should look familiar...

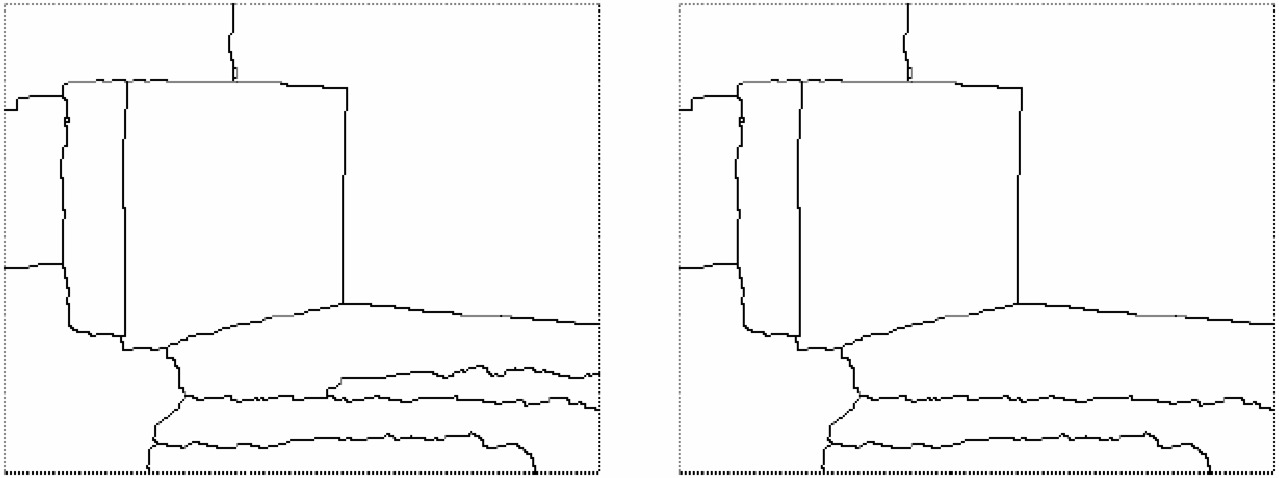
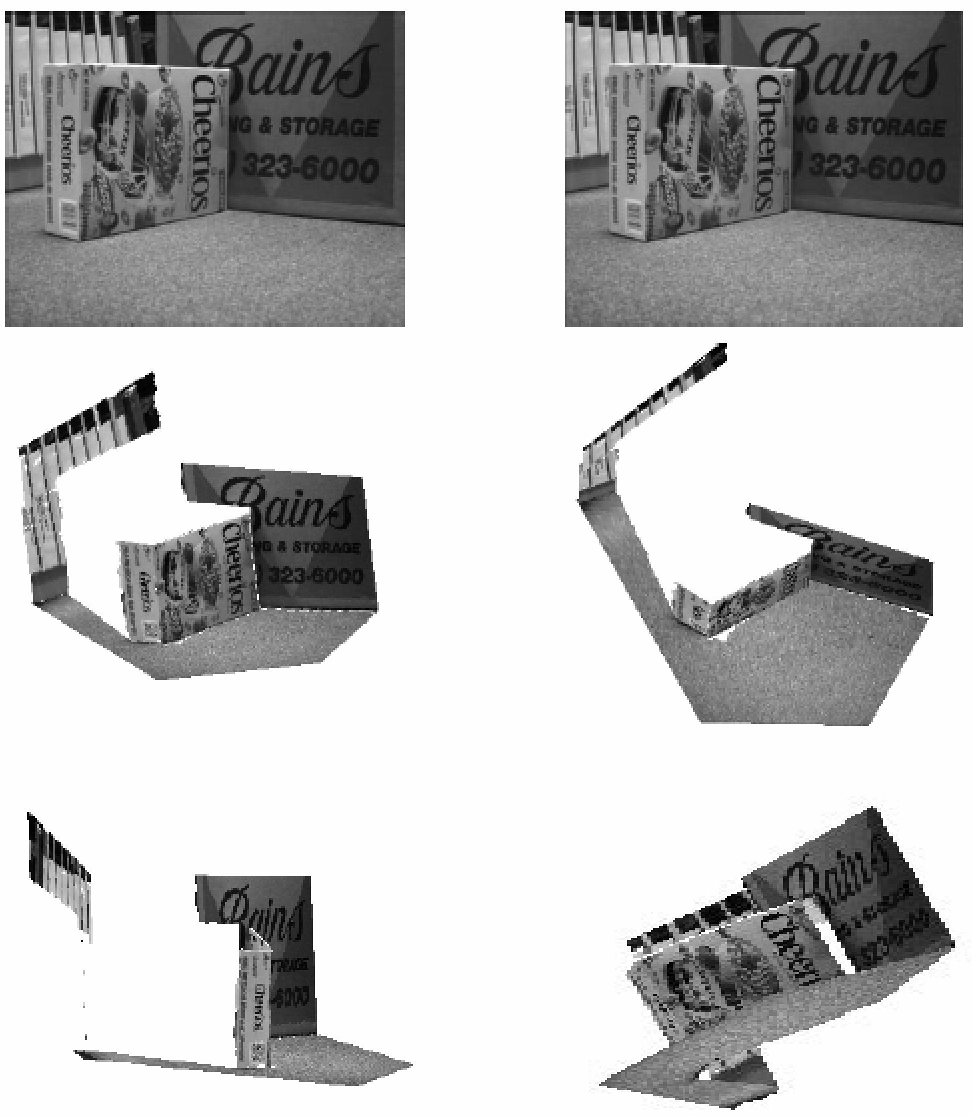
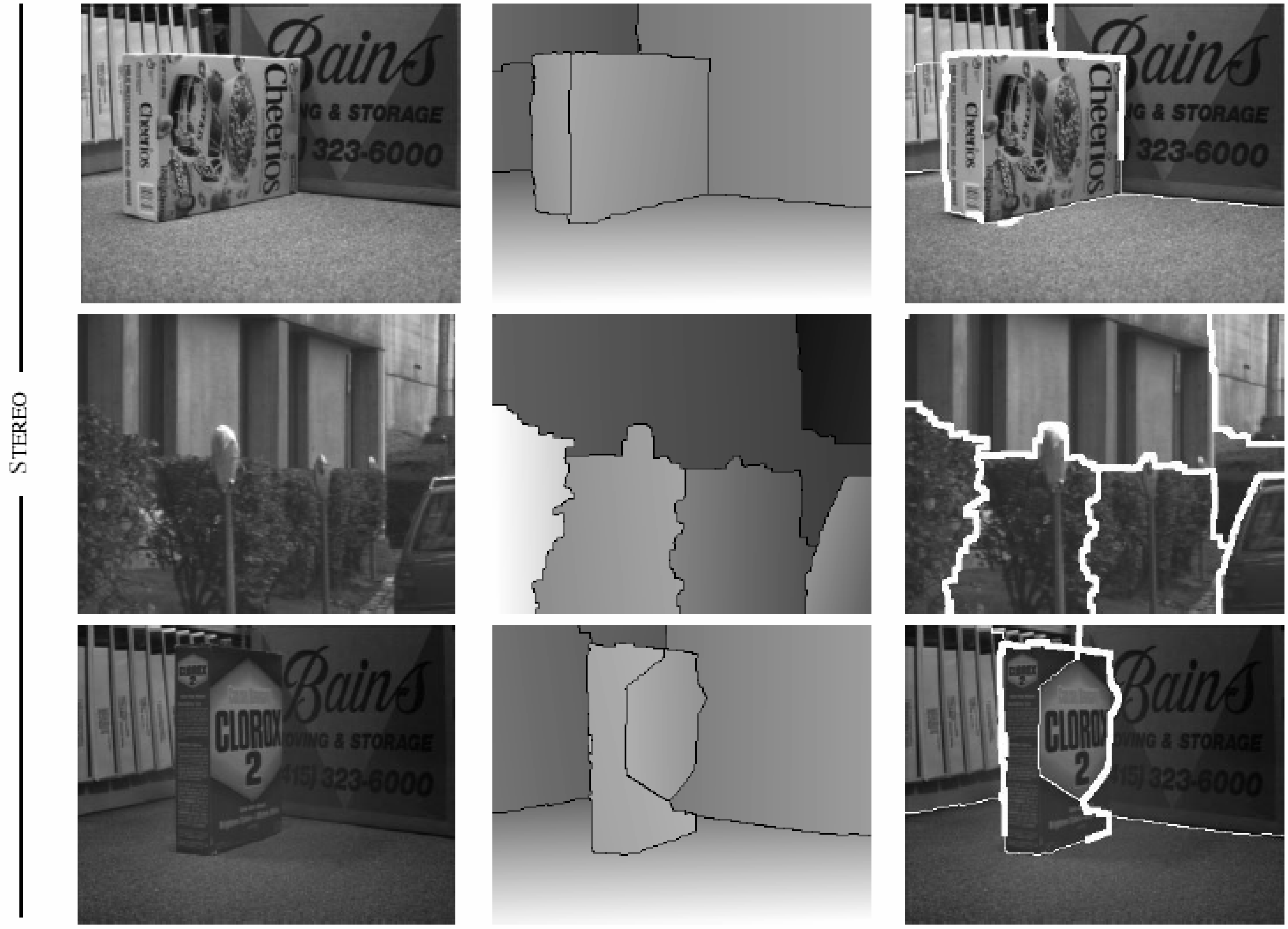


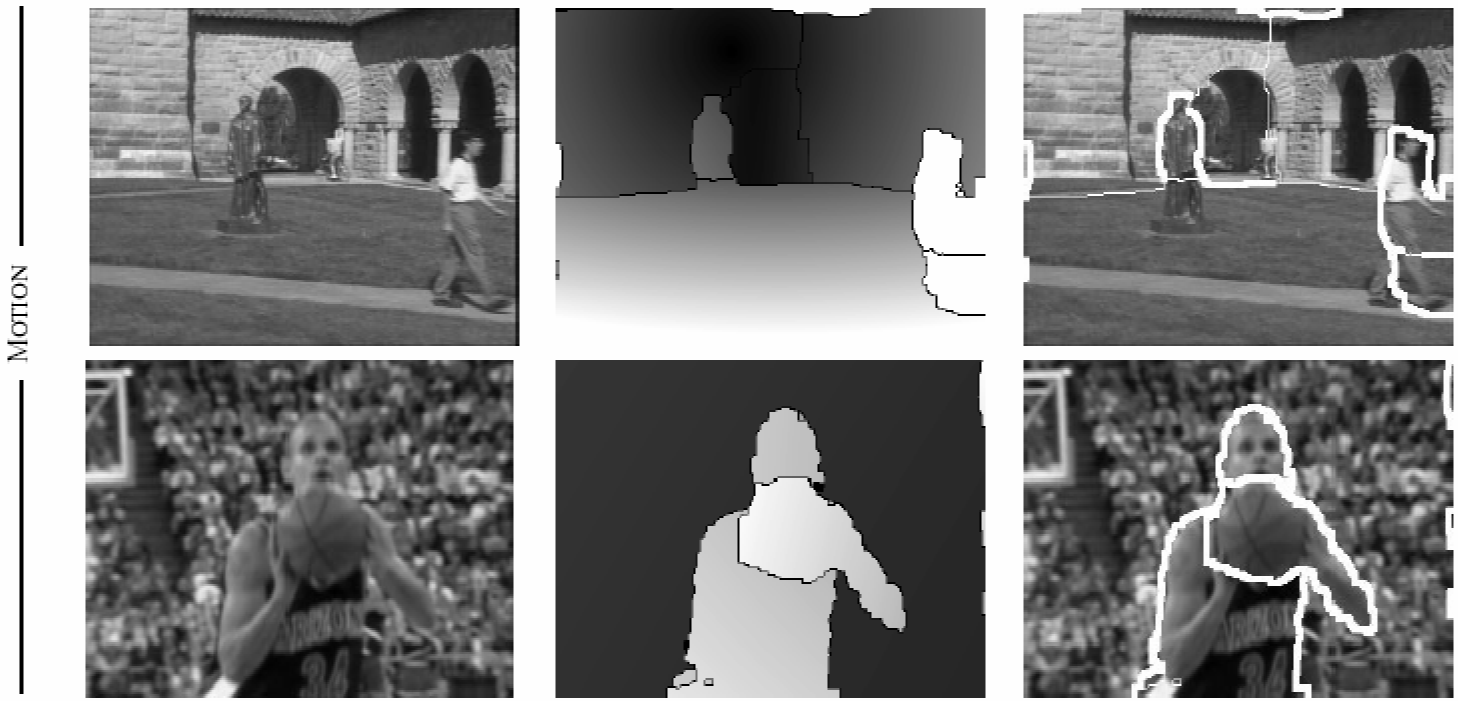
Figure 3: LEFT: Segmentation of the Cheerios image after the convergence of the multiway cut and affine-parameter fitting steps. RIGHT: Two regions on the ground plane have been merged, with more to follow.


Bains
Cheerios
NG & STORAGE
323-6000

Stereo results



Motion segmentation



Discussion questions

1. Graph cuts vs. belief propagation
   - a winner?
2. Applications of GC, BP, EM
   - so far, apps are pretty narrow—too much hype?
3. What improvements to GC, BP, EM do you think are possible?