An Introduction to Stereo Vision and Disparity Computation

6.344 Final Project, Spring 2001

Edwin Olson (eolson@mit.edu)

Melissa Hao (mhao@mit.edu)

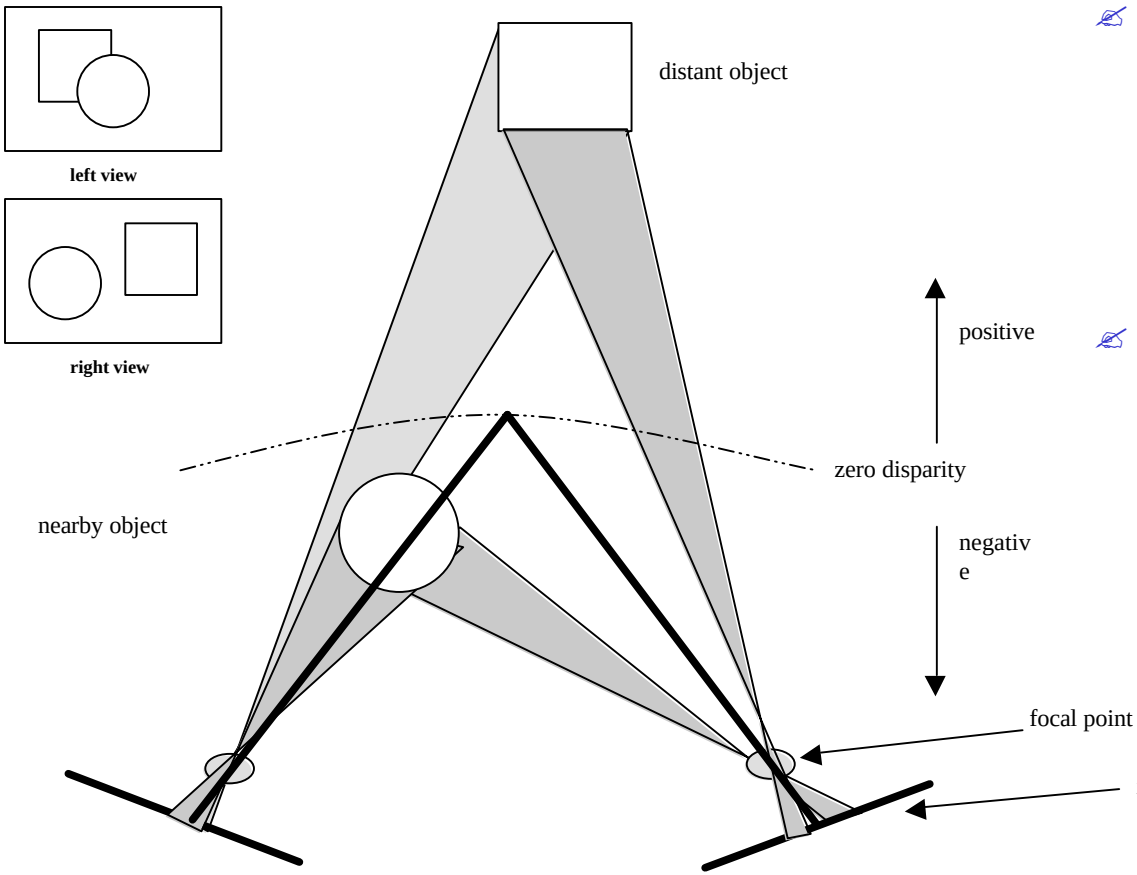
Roadmap

- Introduction
- Background
- Our Program
- Results
- Conclusions

Introduction

- Stereo Vision is perception of depth from two images
  - Humans take it for granted!
  - Very hard problem for computers.
- Applications
  - Industrial robots
  - Object modeling
  - Machine Understanding/AI

Background



- Problem boils down to finding corresponding points
- Similar to motion compensation problem in MPEG encoding.

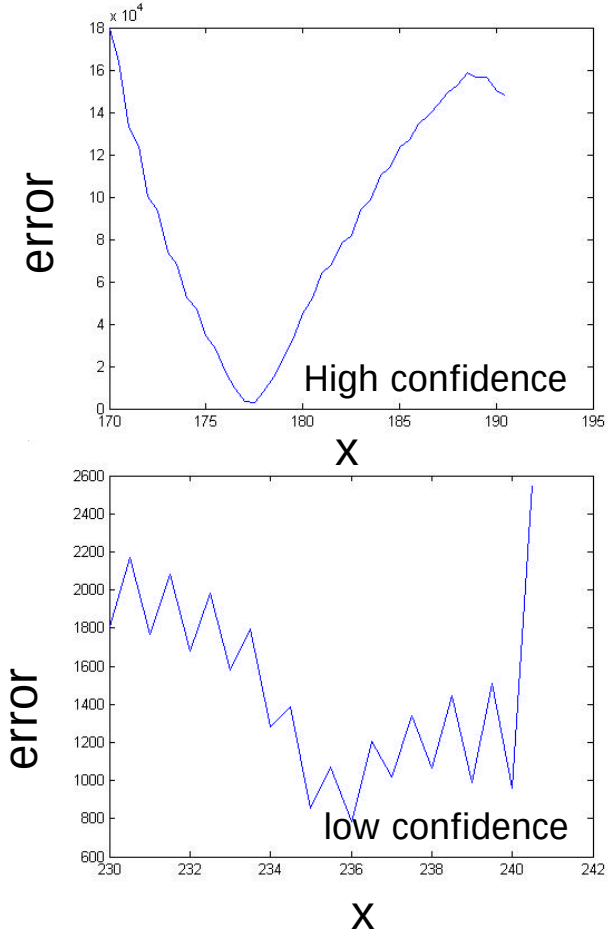
Our Program

- Wrote MatLab program to perform block matching
- Inputs:
  - Two stereo images
    - Color can be exploited
  - Blocksize
- Outputs:
  - Depth Map
  - Confidence Estimate
- Performance
  - Awful! ... but we weren't trying to be fast.

Confidence Estimation

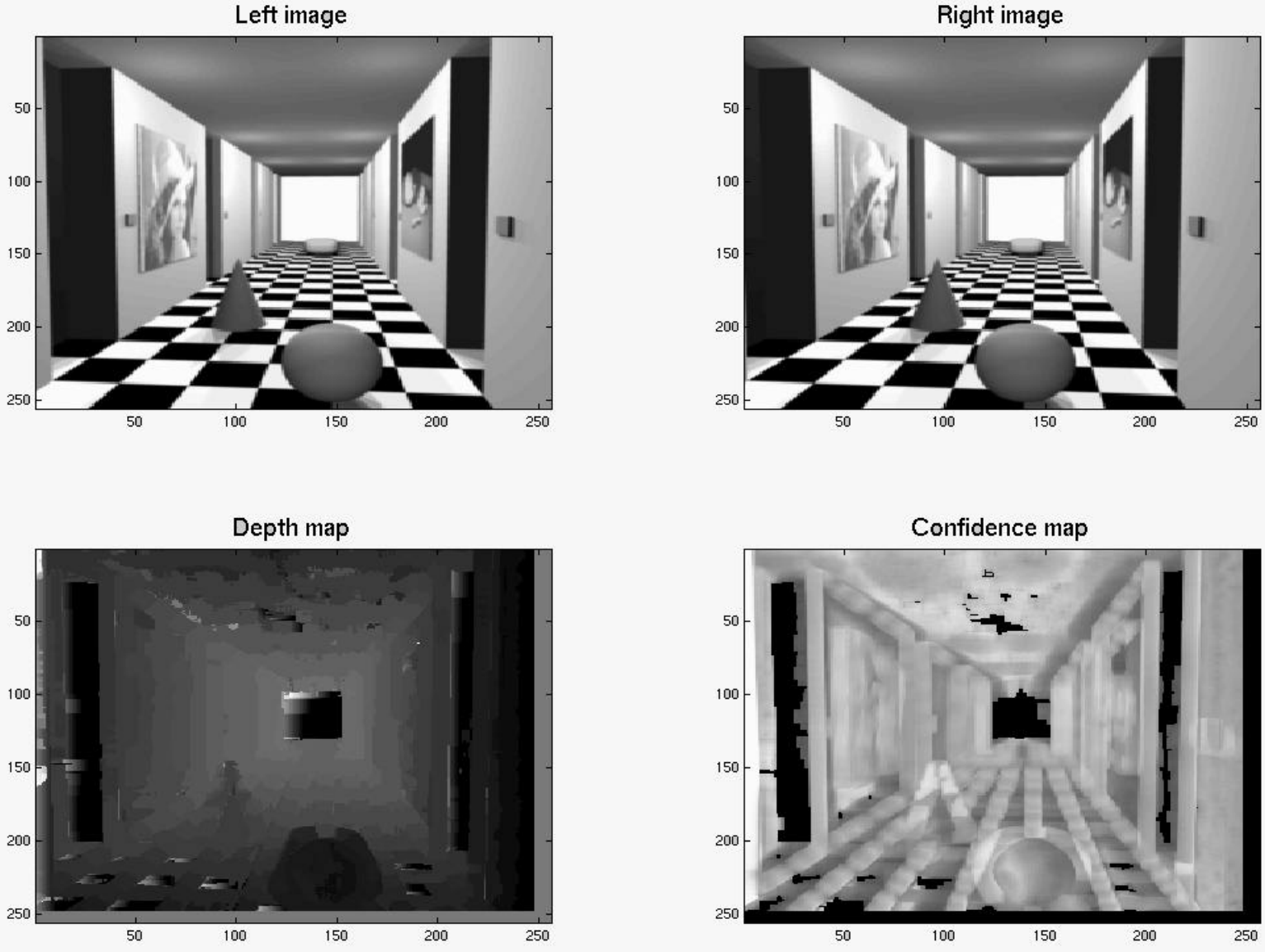
- Block matching sometimes works great, sometimes does something dumb.
  - Noise, Occlusions, Specular highlights
- When it does something dumb, can we detect that?
  - Yes! For the same reason that Log searches work!

Confidence Estimation (2)



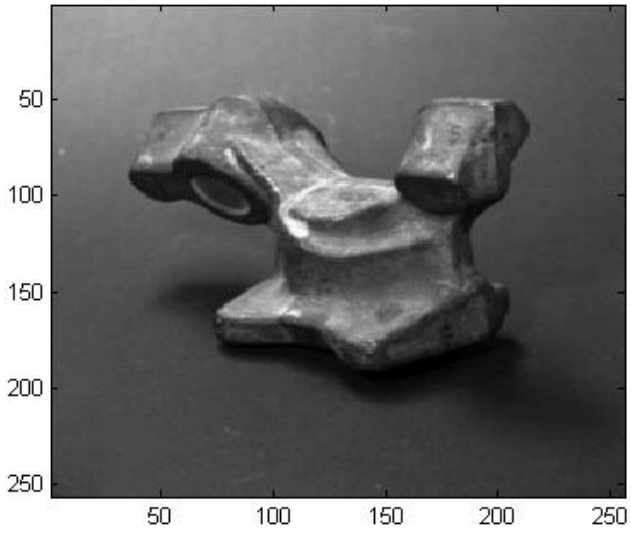
- Block matching works well when there's a single "obvious" minimum.
- Tried Several Metrics:
  - minimumerror
  - averageerror / minimumerror
  - interpret as PDF, compute expected disparity error
- Works, but room for improvement
  - Typically too conservative

Block Matching (16x16) half-pel resolution, synthetic input.

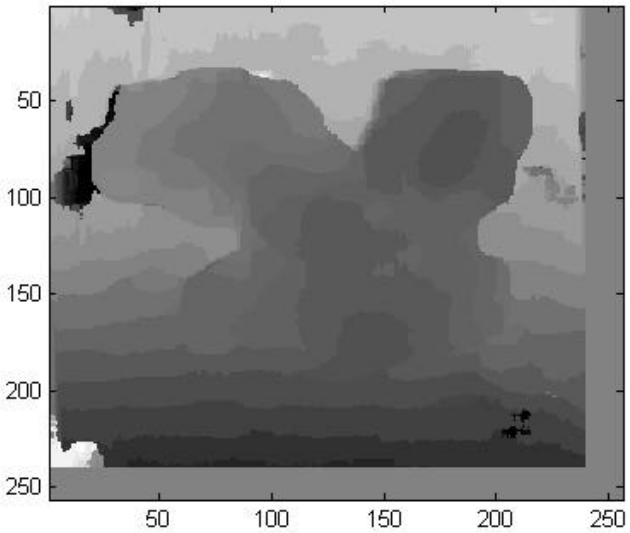


Block Matching (16x16) half-pel resolution

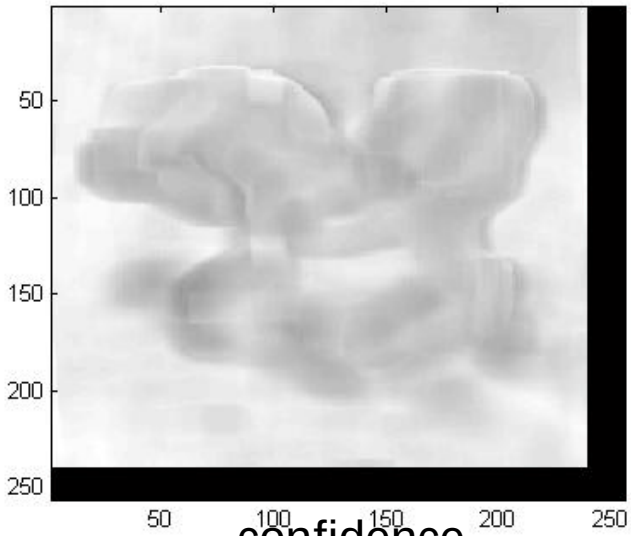
left



right

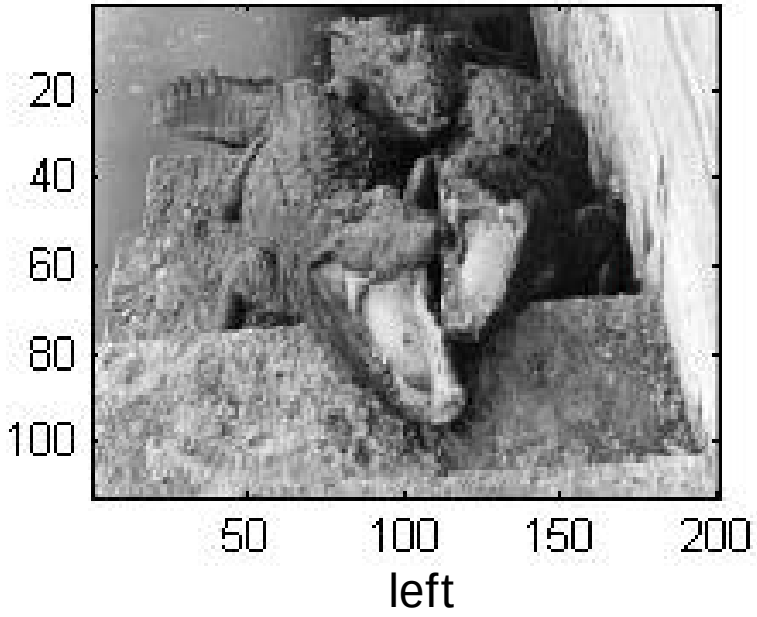


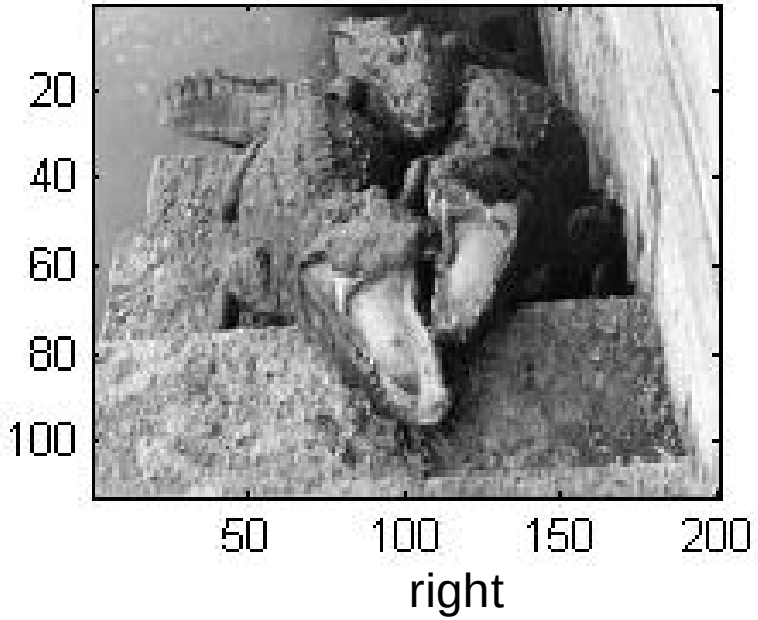
depth

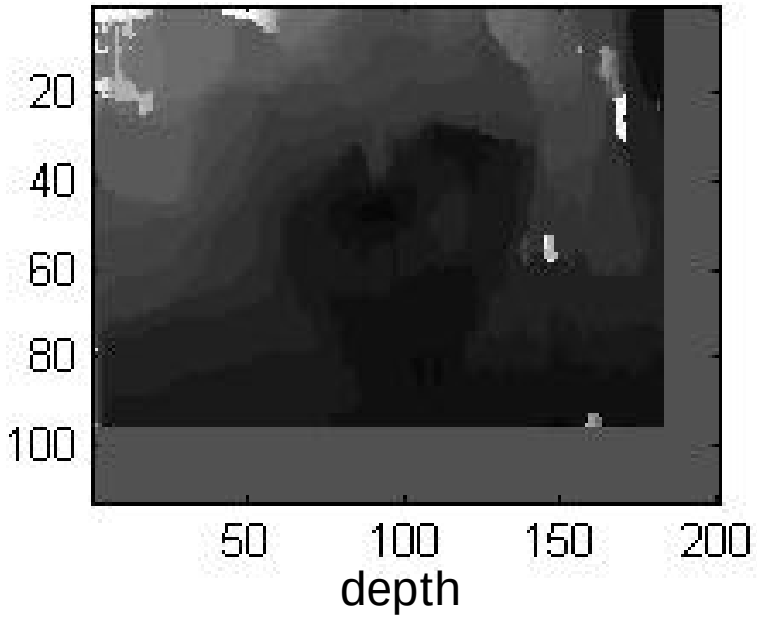


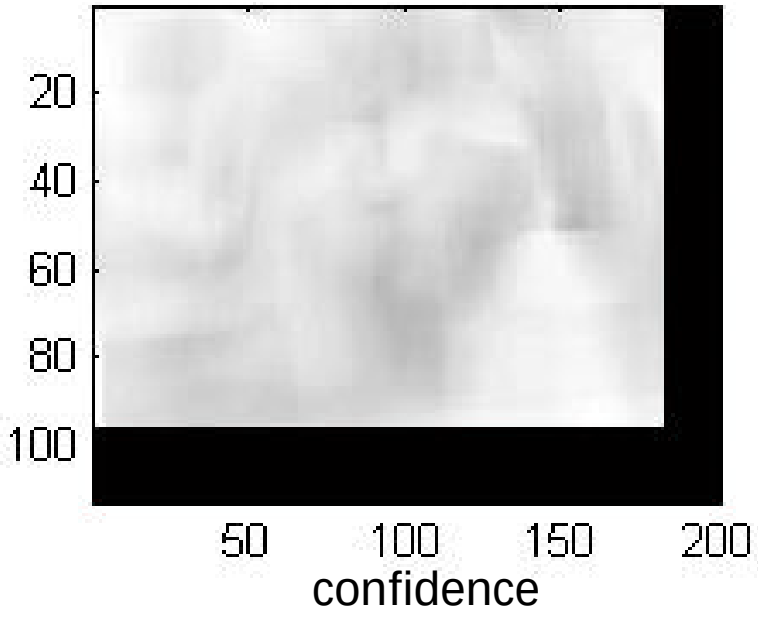
confidence

Block Matching (16x16) half-pel resolution





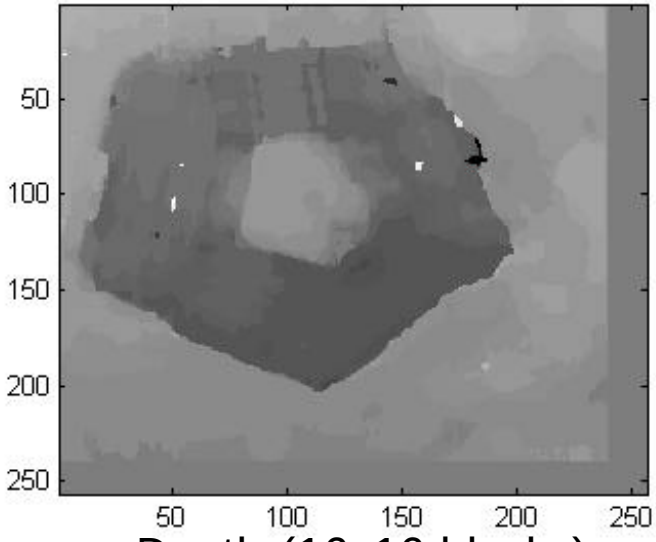




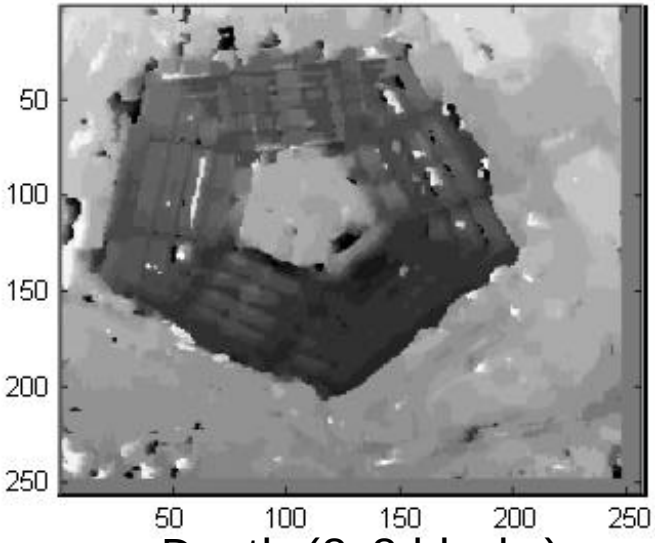
Block Matching half-pel resolution. Block size comparison.

left

right

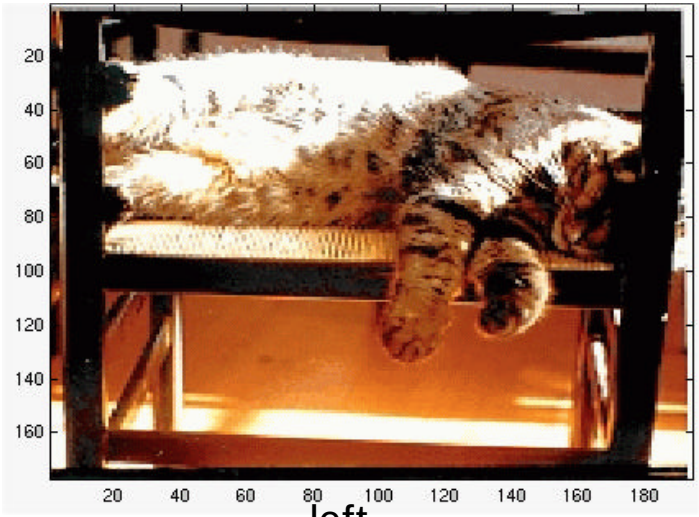


Depth (16x16 blocks)

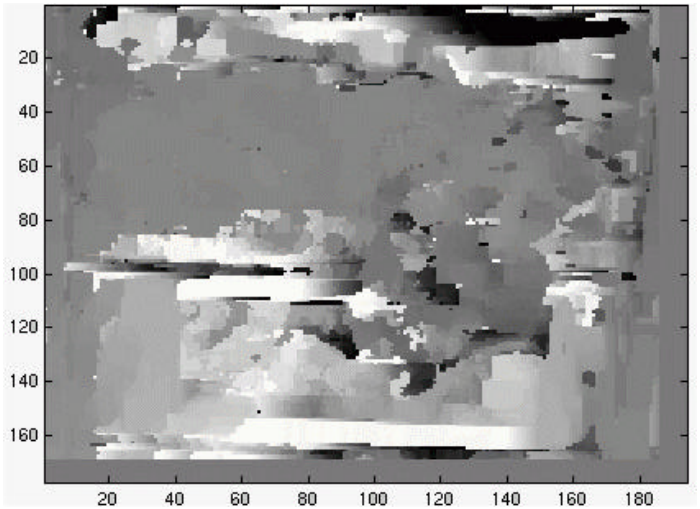


Depth (8x8 blocks)

Block Matching half-pel resolution. Use of Color.

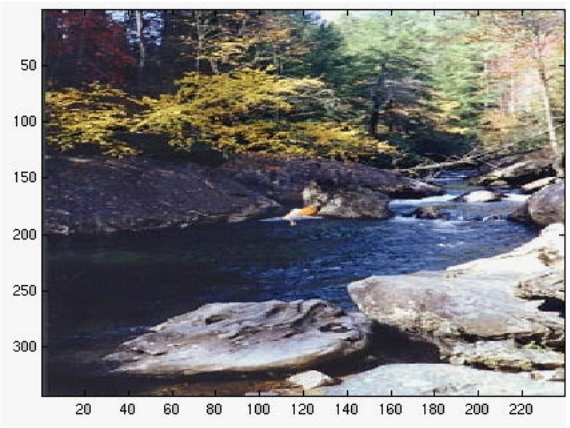


left

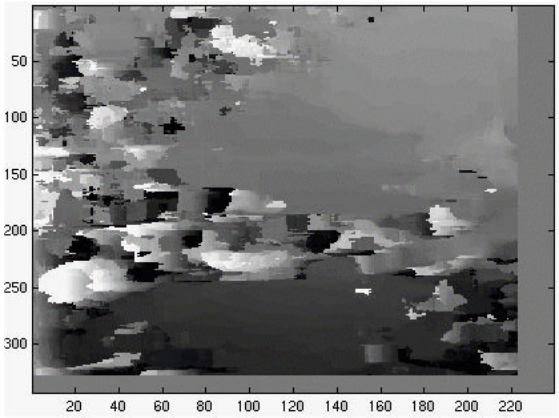


depth

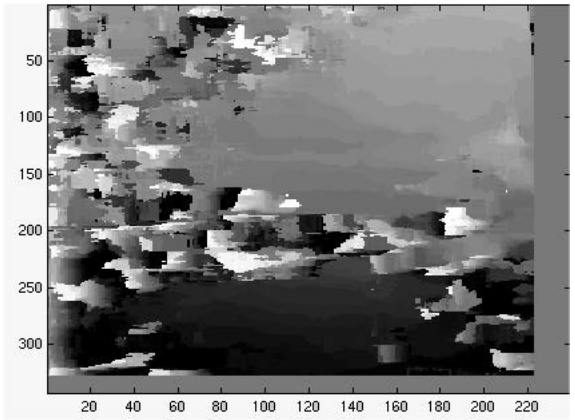
Block Matching half-pel resolution. Use of Color. Poorly aligned images.



left (right was misaligned!)



Depth (using color)



Depth (b&w)

Conclusions

- Block Matching works quite well, but is computationally expensive.
  - Very good performance on regions with significant detail
  - Confidence estimates can give hints on where block matching did poorly.
- Lots of ongoing research
  - Neural nets
  - How do we make it faster?