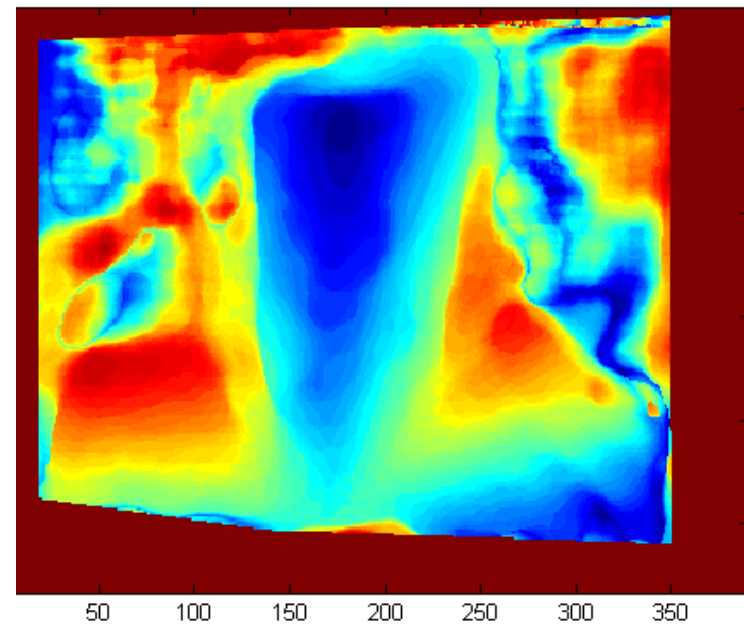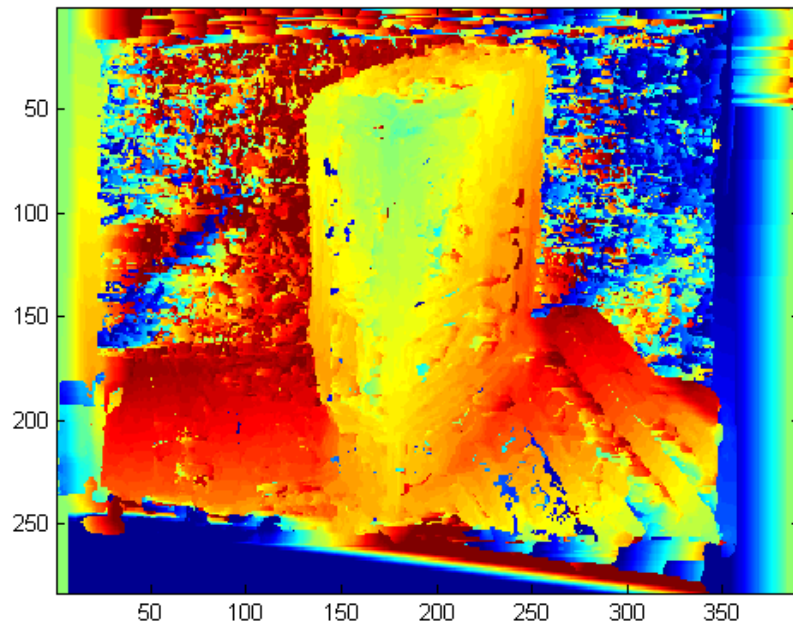# Project Presentation – Week 9

Joshua Michalczak
For UCF REU in Computer Vision, Summer 2010

Last Week

- Fix'd Rectification
- Homebrew Disparity Mapping
- Matlab code

This Week

- Rebuilding Matlab pipeline in C#
  - Using "Emgu CV" to map opencv (C++) to C#
    - Lots of things nice; easy (for example, GUI)
    - A few things not so much; managed memory (C#) sometimes doesn't like working with unmanaged memory (C++)

This Week

- Currently able to...
  - ... single camera ...
    - ... calibrate camera
    - ... find fundamental matrix between two views
    - ... rectify images and find disparity using SGBM
  - ... stereo rig ...
    - ... calibrate cameras individually
    - ... BUG: can't calibrate rig because EmguCV undistort code is broken
      - Fixed in a current SVN revision; downloaded and building
    - ... rectify images and find disparity using SGBM

Results (No pictures ☹)

- **Single camera**
  - Fundamental matrix sometimes fails
    - "Critical Movement"
  - Mixed Quality in disparity maps
    - Mostly "nonsense", sometimes "good"; parameters?
- **Stereo rig**
  - Expect better quality
    - Pre-runtime calibration
    - 3D points at time t rather than a combo of t and $t+1$

Quick Question for Everyone

- What is my topic on?
  - "3D"? "Structure from Motion"? "I have no idea"?
- Structure from Motion is only a step in a pipeline
- I'm working on leveraging 3D scene information to detect objects which are moving relative to the scene
  - Previous work mostly flow based
    - Scene-to-Camera & Object-to-Camera movement, not necessarily representative of Scene-to-Object

Thoughts on how to do so

- Determine the "Empty Volume"
  - When we recover a 3D surface from a camera, we can make assumptions that the space between the surface and the camera center is empty
  - We can recover the volume of each reconstruction and it's location relative to the world; volumes can be merged into a single solid
  - Objects which "puncture" the solid (within some threshold) must have done so through movement

Thoughts on how to do so

- "Scene Flow"
  - Develop correspondences between 3D points at time *t1* and *t2* using mixture of information (photo consistency, camera/rig locations, etc)
  - Points which are matched can be tracked in 3D coordinates; movement within some threshold (or outside some distribution) can be marked vs. noise-induced false movement

Thoughts on how to do so

- "Object Distributions"
  - Segment 3D reconstruction into objects
  - Model an object as a centroid (mean of object points) and a standard deviation in each cardinal direction
  - Objects which move will show increasingly larger deviations over time as well as a moving centroid; objects where these properties cross a threshold can be marked for movement