3dscanner

Anything updated or added since the initial project report is red

Progress Report 1

Anything updated or added since progress report 1 is green

Progress Report 2

Members

- Matthew Diamond
- Sean Cascketta
- Nishant Gupta
- Miguel Del Valle

Topic/Objectives

We plan to write a program that will extract 3D models of scenes from videos taken with the Nintendo 3DS.

- **Objective: Understand the AVI file format produced by 3DS systems**
  - Result: two separate video streams for the left and right camera to make processing simple
  - Result: code that allows OpenCV to properly interpret a single video stream that contains a stereo pair of images for each frame as two separate video streams
  - Work required: research
  - ~~Achievable by 11/13/14~~
  - **Completed!**
    - See process_3ds_video.sh and process_3ds_video_info.md
    - Examples of output can be found in test_data/videos
- **Objective: Reconstruct a 3D scene using a single stereo image pair produced by 3DS systems**
  - Result: a proper tuning of the StereoSGBM class in OpenCV that will produce a disparity map that appears to be correct
  - Result: a 3D point cloud of the scene that appears to be an accurate reconstruction
  - Work required: documentation reading/experimentation
  - Achievable by 12/2/14
  - We have not yet been able to get a satisfying disparity map
    - See 3dscanner.py, 3dscalibrator.py
      - It is still very much a work in progress

      - We have a GUI application for tuning StereoSGBM, but are having trouble getting good results
      - We are looking into camera calibration using recognizable patterns (Ex: a chessboard) so that we know the details of the stereo configuration and can use them for proper rectification
    - See 3drig.py, 3drigcalibrator.py
      - We have constructed a constructed stereo rig (two webcams attached to something for a consistent configuration)
      - We are getting decent depth images from this, will require calibration
- **Objective: Track 3DS camera position**
  - Result: Information regarding the extrinsic parameters of the camera
  - Work required: documentation reading/looking at examples or tutorials/experimentation
  - Hopefully achievable by 12/2/14 (perhaps not achievable)
  - We have no progress towards this goal
    - We are still researching how to accomplish this
    - From what we understand, this may be the single hardest aspect of our project, and as such we are going to attempt rotating the object instead of moving the viewpoint as a less ambitious starting point
    - We are prioritizing other goals for the moment
- **Objective: Properly rectify multiple 3D scenes together**
  - Result: a point cloud of a single scene that has been composited together from multiple angles of the same scene with a visually satisfying degree of accuracy
  - Work required: documentation reading/ looking at examples or tutorials/research/experimentation
  - Achievable by 11/20/14
  - We have no progress towards this goal
    - We are researching how to accomplish this, looking into space carving
    - We plan to first use 8 viewpoints that are each offset by 45 degrees more than the last viewpoint (0, 45, 90, etc.) for an easy set of point clouds to work with that all have known viewpoints
    - Eventually, we hope to be able to use extrinsic parameters of the camera to determine how to piece together the point cloud, rather than using a preconfigured set of viewpoints
- **Objective: Final product**
  - Result: A python script that will take as input a single AVI file produced by a Nintendo 3DS system and output a 3D point cloud of the scene in the video with a visually satisfying degree of accuracy
  - Work required: everything else must be complete
  - Achievable by 12/2/14