Kevin Johnson and Simón Lorenzo
EE 367 Project Proposal
Feb. 8, 2019

3D Scanning using Calibrated Camera Arrays

Abstract:

We propose a camera array mounted on a scanning device to produce colored three-dimensional maps of an object of interest. By calibrating pairs of low-cost internet-protocol (IP) cameras, we will generate depth maps of the object from various angles. The cameras pairs will be mounted on an aluminum frame scanning arc controlled by a motor. The motor will rotate the aluminum arc to perform a scan of any object placed on the stage's central mount. For improved illumination, the frame will also include an LED array. We will combine the depth maps from each camera pair over the duration of the scan to yield centimeter-scale resolution images of the exposed object surfaces. If time allows, we will also perform the same processing with an industry-quality depth camera and compare the resulting scan resolutions.

Motivation:

Our primary motivation is to develop an economical and robust 3D scanner for producing 3D models of sub-meter scale objects with centimeter resolution [Fig.1,2]. Current 3D scanning technologies use high-cost depth cameras, and we aim to provide both the hardware and software techniques for achieving a similar depth resolution with lower-cost resources [1,2]. If successful, the final 3D scan could be converted to the required file format for existing 3D printing technologies, demonstrating a use in rapid prototyping.

Related Work:

In this project, we will use existing software to interface with, collect, and process image data from commercially available hardware. We will be using the MATLAB Computer Vision Toolbox for calibrating images and generating point clouds from stereo pairs of ELP Megapixel Mini™ IP cameras [3,5]. We will collect data from these cameras using the open-source ONVIF device manager and iSpy surveillance software [6,7]. As a possible depth resolution reference, we may also collect data from an Intel® RealSense™ D435 depth camera [4].

Our envisioned final product is similar to stereo scanning technology already on the market. The David Visions HP 3D-scanner at Stanford's Product Realization Laboratory (PRL) is one such example [1,2]. This technology can achieve sub-millimeter resolution, but costs upwards of four thousand US dollars. We can use these cost and performance metrics as reference points for our final device.

Implementation and Milestones:

1) Intel® RealSense™ depth camera [4]
    a. Acquire camera
    b. Acquire PC using OS of Windows 7 and up with USB3.0 ports
    c. Download relevant software and camera drivers
    d. Display and read real-time depth and color data
    e. Save point cloud data to shared Dropbox drive
2) ELP Megapixel Mini™ IP cameras [5]
    a. Acquire (x n) pairs
    b. Acquire converter to power-over-ethernet (POE)
    c. Acquire POE power supply
    d. Acquire ethernet cables to interface with PC used in 1)

- e. Configure computer and camera IP addresses
- f. Install ONVIF open source device manager [6]
- g. Install iSpy open source camera software [7]
- h. Calibrate using checkerboards and the MATLAB Computer Vision toolbox [3]
- i. Generate point clouds using MATLAB Computer Vision toolbox [3]
- j. Save point cloud data to shared Dropbox drive

3) Scanning Stage [Fig. 1,2]
   - a. Acquire aluminum t-slot framing and connections
     - i. Cut, press, and bend frame to shape
     - ii. Use compatible brackets
   - b. Acquire acrylic slats
     - i. Laser-cut to shape
   - c. Acquire server rack
     - i. Mount compatible motor controller, POE switch, and power supply
   - d. Acquire motor and power supply
     - i. Interface with central PC
   - e. Acquire ball bearings for motor drive shaft
   - f. 3D print frame-to-motor connection
   - g. Acquire object mount
     - i. Choose appropriate background for object imaging
   - h. Mount cameras to frame
     - i. Using t-slot compatible screw mounts
   - i. Assemble scanner
4) Post-processing [Fig. 3,4]
   - a. Combine the point cloud data from the scan into a full 3D scene reconstruction
     - i. Use MATLAB Computer Vision toolbox [3]
       1. Intel® RealSense™ camera
       2. Stereo camera pair array
   - b. Compare resolution
   - c. (Optional) Convert to 3D printer compatible format

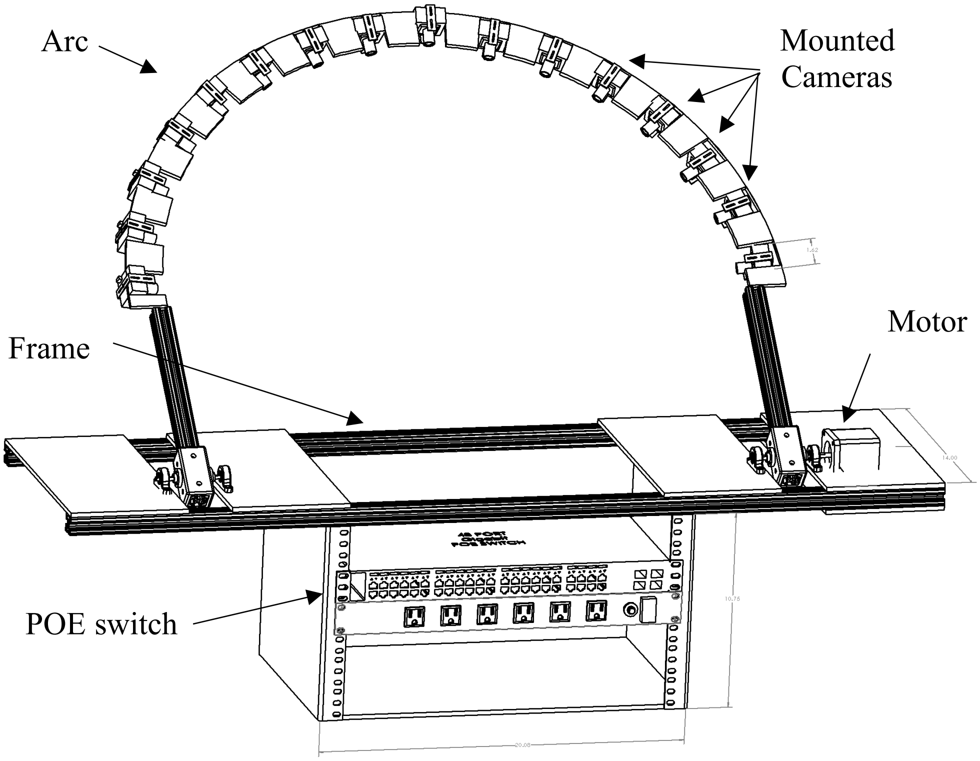

Figure 1: Mechanical design of the 3D scanner with mounted camera array and supporting server rack.

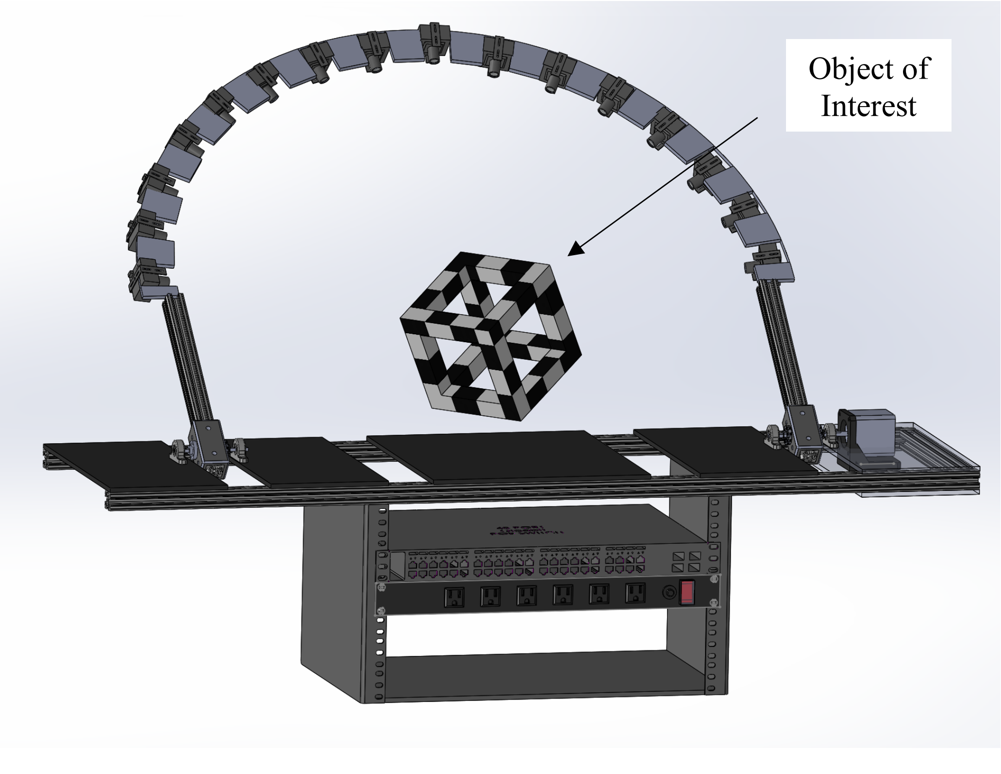

Figure 2: Colored illustration of the 3D scanning device. Cameras are mounted to the arc controlled by the motor which rotates the structure around the object of interest.

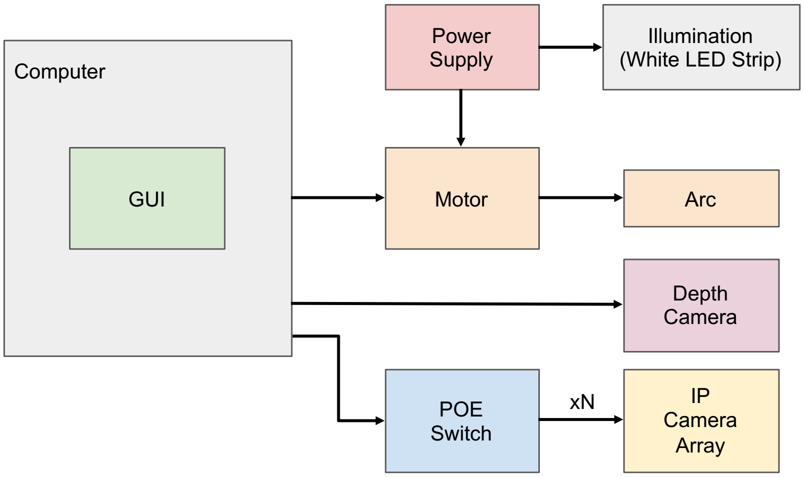

Figure 3: High-Level hardware system design. A single central computer with graphical user interface controls the IP camera array, reference depth camera, scanner arc motion, and scene illumination.

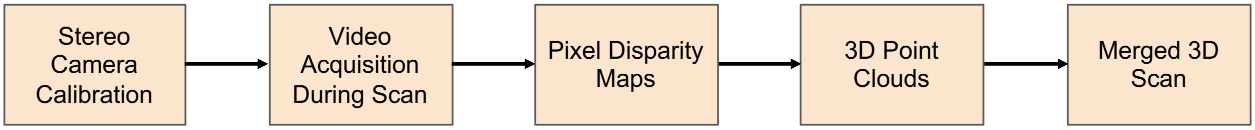

Figure 4. High-level software design. We begin by calibrating the pairs of stereo cameras and acquiring scan data. We use the calibration data to correct the scan video and generate pixel disparity maps between the left and right cameras in the pair. We then generate and merge the 3D point clouds. Software processing done in MATLAB ™ with the aid of the Computer Vision toolbox [3].

References:

1) Stanford University. (2018). 3D Scanning. *Stanford Product Realization Lab.* Retrieved from https://productrealization.stanford.edu/resources/processes/3d-scanning.

2) David Group. (2019). 3D Scanner Introduction. *David and HP Company.* Retrieved from https://www.david-3d.com/en/support/david4/introduction.

3) The MathWorks Inc. (2019). Computer Vision Systems Toolbox. *MathWorks.* Retrieved from https://www.mathworks.com/products/computer-vision.html.

4) Intel. (2019). Intel RealSense Technology. *Intel Software Developer Zone.* Retrieved from https://software.intel.com/en-us/realsense.

5) Amazon. (2019). ELP 1280720p 1.0 Megapixel Mini IP Camera, Mini Hidden Network Camera. *Amazon.* Retrieved from https://www.amazon.com/ELP-1280720p-Megapixel-Camera-Network/dp/B00KA4M4WS.

6) Slashdot Media. (2019). ONVIF Device Manager. *SOURCEFORGE.* Retrieved from https://sourceforge.net/projects/onvifdm/.

7) DeveloperInABox. (2019). Open Source Video Surveillance Software. *ISPYCONNECT.* Retrieved from https://www.ispyconnect.com/