3DU

A Cost Efficient, User Friendly Raspberry Pi Zero Photogrammetry Solution

Jacob Stauch, Benjamin High - Dr. Scott Sigman & Dr. Chris Branton

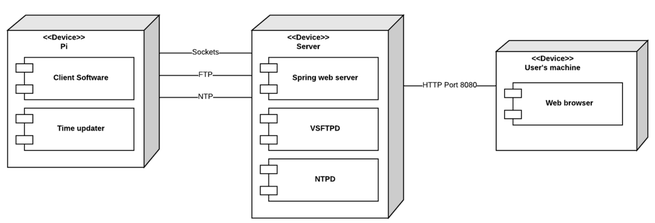
Abstract

Photogrammetry is the process by which a 3D mesh is extracted from a set of images containing a subject at various angles. While an implementation of this idea with Raspberry Pis is not new, we propose many improvements to scalability and ease of use. To achieve this, we have created a performant, modular, and portable system that iterates on previous 3D scanner designs. We have developed a LAN solution utilizing C++ for any number of Raspberry Pi Zeros, all connected to and managed from a central server written in Java, maximizing portability, wrapped in a simple web interface suitable for use by any skill of user. Using sockets as a communication layer, NTP for time synchronization, and FTP for file transfer, server setup is simple, and Pi software can be easily deployed through a pre-built image, as well as updated through the server.

Purpose

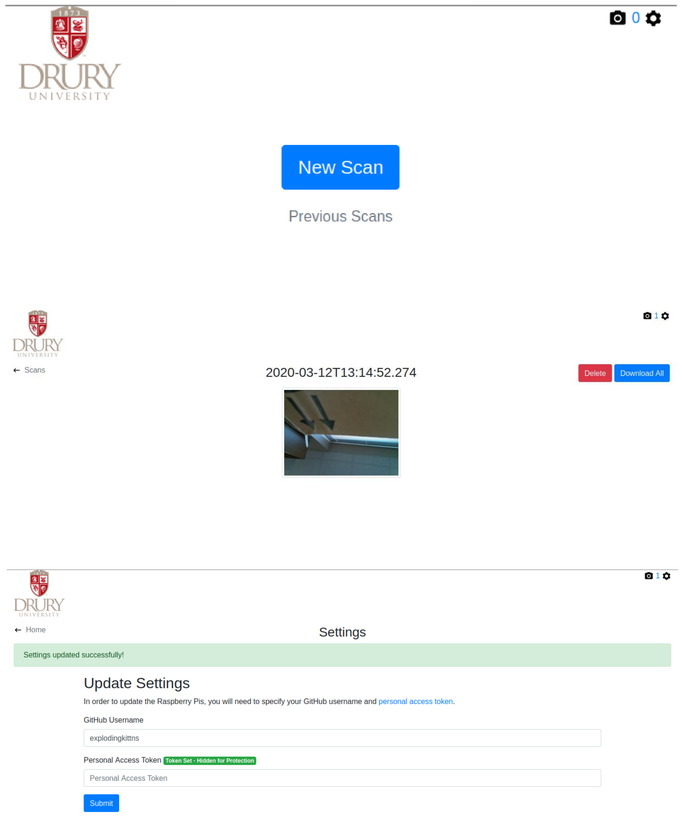
- Utilize Raspberry Pis to record images in sync from multiple angles of objects that may be in motion.
- Organize images on a central server storage device, and allow the user to download images as an archive.
- Provide a web interface for easy access and control.
- Design an update system for easy updating of Raspberry Pi client code.
- Create a modular design to allow for quick and easy installation and removal of cameras.

Architecture



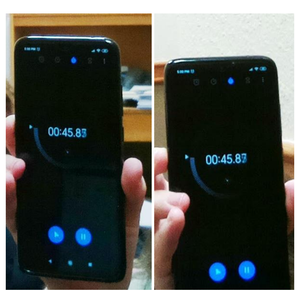
- Pi socket client establishes connection to the Server's listener, synchronizing clocks with NTP.
- Server generates unique IDs for each Pi that connects.
- The Server's web interface allows commands to be sent to the back-end, facilitating simple communication with all Pis simultaneously.
- Resulting images are uploaded through FTP to the Server.

Web Interface



Synchronization

- Pis synchronize using the NTP server.
- Upon capture request, the server generates and sends a UNIX timestamp to all clients, who all wait until the time arrives. When the time arrives, the image is captured.
- Decentralizing the waiting process allows for ~5 ms accuracy, regardless of the number of Pis.



Client Binary Update

- Client binaries are uploaded to GitHub repository as a release.
- Server utilizes GitHub's releases API endpoints to detect when a new update is available, and downloads the binaries.
- Server unpacks and saves the updated binaries to a location on the filesystem available for access through FTP.
- Each client connects via FTP to the server, downloading the updated binaries.
- Once the clients have downloaded the update, they replace their binaries, and reboot.
- A systemd service ensures the program runs at boot.

Future Work

- More advanced synchronization methods
  - Capture image directly using OpenCV
  - Synchronize a video stream instead of initiating a picture
  - NTP stratum 1 server with GPS timer
  - Minimize NTP drift on the Raspberry Pis through a "heartbeat"
- Use Packer to automate Server and Client ISO creation
- More responsive web interface

References

- https://www.instructables.com/id/Multiple-Raspberry-PI-3D-Scanner/
- https://www.raspberrypi.org/blog/affordable-raspberry-pi-3d-body-scanner/

Acknowledgements

Dr. Scott Sigman and Dr. Chris Branton of Drury University's Computer Science department worked as advisors on this project. The project was created for Matt Noblett, M.F.A. of Drury University's Graphic Design department, who provided valuable feedback and information.