

Three Dimensional Scanning and Imaging of Miniature Objects

Vaishnavi K Bhat
4CB13EC117
Canara Engineering College
Mangalore, Karnataka

Shilpa Prabhu K
4CB13EC093
Canara Engineering College
Mangalore, Karnataka

Saideep D Ullal
4CB13EC089
Canara Engineering College
Mangalore, Karnataka

Ashish Kini
4CB13EC010
Canara Engineering College
Mangalore, Karnataka

Abstract: A 3D Scanner is a hardware and software system for digitizing the shape and color of physical objects under known environmental conditions. The digitized data can then be used to construct digital three-dimensional models. The usage of digitized data or three-dimensional models created by 3D scanners comprises industrial design, medical diagnosis, cultural heritage, multimedia, entertainment, rapid prototyping, and digital fabrication. Some limitations are depending on the kind of objects that should be digitized. For example, line laser based 3D scanners have some difficulties with shiny, mirroring or transparent objects. The most important issue is caused by influence of ambient lighting.

Keywords- Do it Yourself, FabScan.

I. INTRODUCTION

The main motivation behind the selection of the idea is the abundance of the open source information and community present on github (Fabscan).

A. Fabscan

The FabScan is an open-source, do-it-yourself 3D laser scanner that started out as a thesis project by Fancis Engelmann .The idea behind the FabScan was to supply the FabLab Aachen with a low-cost 3D scanner. In 2014, development of the project was taken over by Ren`e Bohne and me. FabScan was featured on Thingiverse1 and it is supported by a steadily growing user group on Google Groups.

B. FabScan Pi a Web-Enabled Stand-Alone 3D Scanner

FabScan Pi is the next generation of FabScan. The hardware setup is based on the FabScan CUBE . The webcam, Arduino , and FabScan shield are replaced by a Raspberry Pi with the Raspberry Pi camera module and a new developed FabScan HAT for Raspberry Pi. A HAT (Hardware Attached on Top) is a PCB that extends the Raspberry Pi just like an Arduino shield extends the Arduino. The old FabScan software is replaced by a new server-based software which consists of two parts. A web-enabled user interface and a back-end FabScan API server.

Since the new FabScan Pi software is web-enabled the user is able to perform a scan without installing any software parts on a laptop, mobile phone or tablet. The whole scan process can be controlled over a webenabled user interface. The user is able to perform a scan by calling the FabScan Pi server URL in a webbrowser.

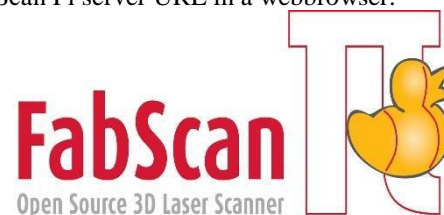


Fig. 1 The fabscan logo

II. CONSTRUCTION AND WORKING

The implementation of Fabscan pi is an enhanced prototype of the Fabscan CUBE.Its construction can be studied in depth as follows

A. Prototype 1

For this prototype a Raspberry Pi 2 was added to the existing FabScan CUBE setup. The Raspberry Pi2 has a quad core processor and should provide enough power for image processing. Therefore the Logitech C270 webcam is replaced by a Raspberry Pi camera module. With the usage of the Raspberry Pi camera module there is no need to disassemble a webcam. The Arduino with the FabScan shield on top is connected to the USB port of the Raspberry Pi 2. An additional 5V power supply is added for powering theRaspberry Pi 2. The first software prototype was written in Python. For this first Python based server prototype a rough web-enabled user interface was designed for testing basic scanning operations on a Raspberry Pi 2.A scan with the first prototype showed that the Raspberry Pi 2 in combination with the FabScan CUBE components, is suitable for fast image processing and 3D scanning. A quality improvement to the FabScan CUBE was given usingtheRaspberryPi2cameramodule.

The camera module has a maximum resolution of 2592x1944 pixels. The Logitech c270 webcam, used by the old FabScan CUBE setup provided a maximal resolution of 1280 x 720 pixels. Therefore the first quality improvement is done by doubling the camera resolution what results in a higher point cloud density. Also the assembly of the hardware setup is easier for the user, because the camera module is ready to use with the Raspberry Pi 2. The camera consists of a small (25x20x9mm) circuit board, which connects to the

B. Prototype 2

For the second prototype the Arduino and the FabScan shield are replaced by a Raspberry Pi FabScan HAT, shown in Figure. The FabScan HAT carries the motor drivers 1 for the turntable and laser stepper motor. The FabScan HAT can carry up to four motor drivers. The motors can be easily connected to screw terminals, which are positioned in front of the motor drivers 3. A second laser and an LED port was added to the FabScan HAT 2. The FabScan HAT is powered by a 12 Volt power supply, which can be connected to a power jack 5. A small step-down circuit placed on the HAT ensures that the Raspberry Pi 2 is powered with 5 Volt. An ATmega328 IC with Arduino Firmware is placed on the HAT 4. The FabScan firmware is flashed to the ATmega328. The Arduino FabScan firmware communicates with the Raspberry Pi 2 over the GPIO serial connection. The Pololu motor drivers are replaced by silent step stick motor drivers for smoothing the motor rotations to prevent object shifting while the table is rotating. Some white LEDs are placed beside the camera to get a better texture scan.

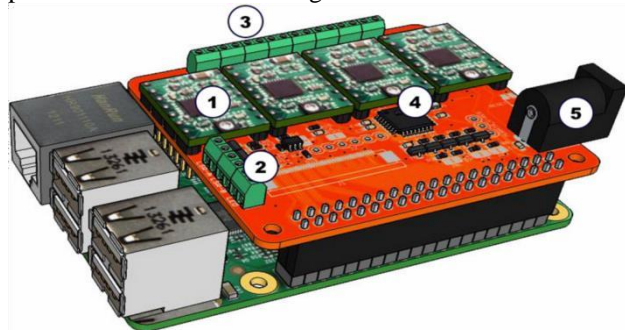


Fig. 2 Implementation of Prototype 2 using the Raspberry pi HAT

C. Software

Figure shows the different layers of the FabScan Pi software architecture. The base of the software is represented by the Arduino compatible firmware of the FabScan Pi HAT “Second Hardware Prototype”. The second layer consists of the back-end server.

Raspberry Pi’s Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. After it is connected to the Raspberry Pi, it has to be screwed to the FabScan case front. The wiring of this prototype is overhead. Also the USB connection to an additional Arduino board with a FabScan shield can be removed. Because the object shifts when the turntable is rotating, the object has to be glued to the table with some tape.

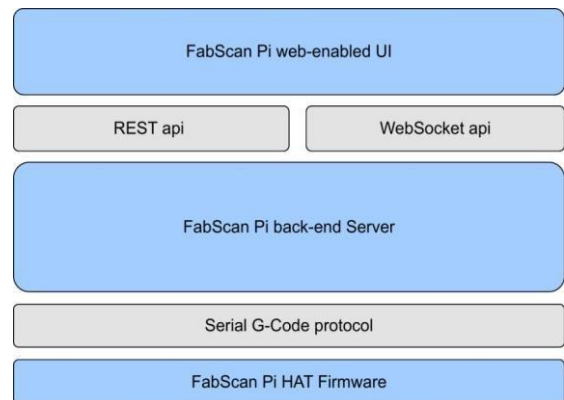


Fig. 3 Block diagram representation of Fabscan Software Implementation.

D. Web Enabled User Interface

FabScan Pi provides a web-enabled user interface. The interface is an application which is written in JavaScript with the usage of AngularJS. It is widely independent of the FabScan back-end server. For the graphical visualization the user interface uses the WebGL standard. The first version of the FabScan Pi web-enabled user interface was created with a mockup software tool called balsamiq. The settings dialog provides a live preview of the laser line recognition.

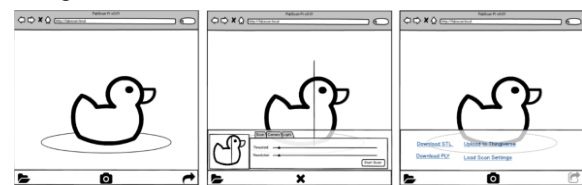


Figure. 4: Main view. The view which is shown when fabscan.local is called in a browser.

The settings dialog provides a live preview of the laser line recognition. Changes the scan, camera or light settings the changes are visible in a real-time video stream. Also the resolution is set in the settings dialog. Those settings are saved in a file beside the other scan data. The share dialog also offers the possibility of loading the scan settings which were used for the related object. This can be helpful when the user wants to scan an object with similar shape, color or surface properties.



Fig. 5 Sharing dialog. Offers the opportunity to download scans in different formats and delete scans.

III. ADVANTAGES BASED ON REQUIREMENTS

1. *Do-it-Yourself.*

Even the FabScan Pi can be build by everyone. The hardware setup changed in a few details. The Arduino and FabScan shield is replaced by the FabScan Pi shield and the Raspberry Pi 2. The webcam is replaced by the Raspberry Pi camera module. Due to the fact that the Raspberry Pi camera module is delivered as a small PCB, it is not necessary anymore to disassemble a webcam module. The software installation process is also easier with the FabScan Pi. The user only needs to download and flash the SD-Card image. The firmware is installed automatically with the first start of the FabScan Pi server.

2. *Standalone Device.*

By using a Raspberry Pi 2 with the FabScan Pi server software and the web-enabled user interface, there is no need to install software on third party hardware like computers, laptops, etc. The FabScan Pi is a full usable stand-alone 3 D scanner.

3. *Low Cost.*

The overall cost of the prototypes is lower than the usual design of 3-D Scanners and can be implemented by hobbyists, makers.

4. *Downloading scans in different formats.*

The FabScan Pi's web-enabled user interface provides the opportunity of downloading the scans in different file

formats. The user is able to download scans in STL, PLY and X3D.

III. FUTURE WORK

1. *Adding more REST API functions*

Currently the FabScan Pi back-end server provides only basic REST API functionality. In a new version it can be useful if the REST API supports also command functions.

2. *Introducing ICP algorithm*

One aspect of laser range scanners is that most objects with re-entrant angles are difficult to digitize. The laser can not reach those angles or the camera is not able to capture parts of the object where the laser is covered by other object parts. The ICP (Iterative Closest Point) algorithm can solve this problem. ICP is used to match point clouds. With the implementation of ICP it would be possible to scan the object with different positions. Several point clouds of different object positions could be matched to one point cloud. With this feature the re-entrant angle problem can be solved.

3. *Scanner Auto Calibration*

An auto calibration mechanism can be realized by placing a checkerboard on the turntable and measuring the distances of camera and turntable. Also the horizontal and vertical alignment of the camera can be determined by the measurement mechanism.

4. *Auto Settings Dialog.*

The usability could be improved by introducing an auto settings dialog. If the FabScan is able to detect the object material and color, it would be possible to detect the best matching scanner settings. With an auto setting feature maybe the scan process speed and quality can be increased.

IV. REFERENCES

- [1] James Adams. Definition of raspberry pi hat. <https://www.raspberrypi.org/introducing-raspberry-pi-hats/2014>.
- [2] W.-K. Chen, *Linear Networks and Systems*. Belmont, CA: Wadsworth, 1993, pp. 123–135.
- [3] F Blais. Review of 20 years range sensor development. *Journal of Electronic Imaging*, September 2004.
- [4] STL File Format. <http://mech.fsv.cvut.cz/dr/papers/Lisbon04/node2.html>, October 2011.
- [5] NI Open. Introducing OpenNI www.openni.org, September 2011.
- [6] Yuan Xin, Zhu Ruishuang, and Su Li. A Calibration Method Based on OpenCV. *Intelligent Systems and Applications (ISA)*, 2011 3rd International Workshop on, pages 1–4, 2011.
- [7] E.L. Hall, C.A. MCPerson, and Tio J B K. Measuring curved surfaces for robot vision. *Computer*, 15(12):42–54, September 1982.