Programmatic 3D Printing of a Revolving Camera Track to Automatically Capture Dense Images for 3D Scanning of Objects

Nikolas Lamb, Natasha Kholgade Banerjee, and Sean Banerjee(✉)

Clarkson University, Potsdam, NY 13699, USA
{lambne,nkholgad,sbanerje}@clarkson.edu

Abstract. Low-cost 3D scanners and automatic photogrammetry software have brought digitization of objects into 3D models to the level of the consumer. However, the digitization techniques are either tedious, disruptive to the scanned object, or expensive. We create a novel 3D scanning system using consumer grade hardware that revolves a camera around the object of interest. Our approach does not disturb the object during capture and allows us to scan delicate objects that can deform under motion, such as potted plants. Our system consists of a Raspberry Pi camera and computer, stepper motor, 3D printed camera track, and control software. Our 3D scanner allows the user to gather image sets for 3D model reconstruction using photogrammetry software with minimal effort. We scale 3D scanning to objects of varying sizes by designing our scanner using programmatic modeling, and allowing the user to change the physical dimensions of the scanner without redrawing each part.

Keywords: 3D scanning · 3D printing · Multi-view imaging
Photogrammetry

1 Introduction

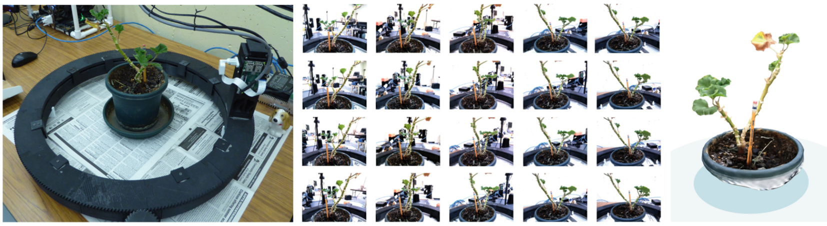
3D models today are instrumental in steamlining tasks such as physics simulations [3], animation of articulated characters [4], and preservation of historical monuments [9]. While traditional 3D models have been manually designed, consumer 3D scanners and automatic photogammetry software provide lower-cost and more dimensionally accurate digitizations of complex or irregular objects.

Commonly available 3D scanners, such as the MakerBot Digitizer and Matter and Form 3D scanner, use one or more stationary cameras to capture an object rotating on a turntable [2]. Rigid or non-rigid motions of the object due to vibrations of the turntable may yield an inaccurate 3D model. Instead of rotating

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-319-73600-6_39) contains supplementary material, which is available to authorized users.

K. Schoeffmann et al. (Eds.): MMM 2018, Part II, LNCS 10705, pp. 390–394, 2018.
https://doi.org/10.1007/978-3-319-73600-6_39

the object, we create a 3D scanner that revolves a camera around the stationary object. Our approach prevents the scanner from disturbing the object during the capture process, allowing our system to capture fragile or delicate objects, such as the potted plant shown in Fig. 1.



(a) (b) (c)

Fig. 1. (a) Our 3D scanner showing Raspberry Pi 3B computer, Raspberry Pi v2 camera, and Nema 23 stepper motor, (b) Images of a plant captured by the scanner using the camera, and (c) Reconstruction of the potted plant using Autodesk ReMake.

2 Related Work

While the emergence of consumer grade 3D scanners such as the Matter and Form 3D scanner are recent phenomena, the technology that enables them has been present and evolving for many decades [5]. Most contactless 3D scanners rely on using either depth or color cameras to scan objects placed on a turntable [8,10]. Commercial photogrammetry software, such as Autodesk 123D Catch, Trnio, and Autodesk ReMake [1], allow users to capture and reconstruct everyday objects with consumer grade cameras or mobile phones. While the approaches provide low-cost digitizations, they require the user to manually capture a dense set of images from multiple angles. This task may prove tedious for an average user as it often requires careful monitoring of image uniformity and placement. Multi-camera approaches require dense arrays of 25 or more RGB cameras for high fidelity stereo matching, or when sparse require RGB-D sensors that either need shaker rigs to reduce interference [7] or fail to capture over short distances [6]. Our approach addresses the issues inherent in turntable 3D scanners by keeping the object of interest static while revolving the camera around the object. It also circumvents the tediousness and imprecision of manual capture by capturing and importing all images automatically.

3 Programatically Modeled 3D Printed Scanner

Our 3D scanner consists of a Raspberry Pi v2 camera and Raspberry Pi 3B computer installed via an adjustable height tower on a circular camera track, shown in Fig. 1. The track is made by connecting twelve segmented arc pieces which are geared to interface with a Nema 23 stepper motor. The geared track moves over ball bearings placed in bearing holders on a lower grooved support track. We prevent the camera track from wandering during motion by using guide clips and a double helical gear design for the geared arc and motor gear. The motor is controlled by the general purpose input/output (GPIO) pins of the Raspberry Pi computer via a bipolar stepper driver. The Raspberry Pi is powered using a 2.5 A 5 V DC power supply, while the stepper motor is powered using a 36 V DC power supply. The Raspberry Pi and the host desktop computer which activates the Raspberry Pi are networked together using a Gigabit switch.

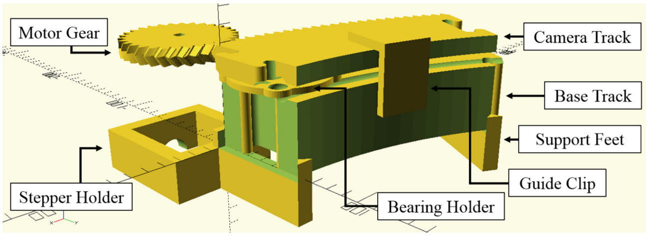


Fig. 2. Mechanical hardware of our 3D scanner in OpenSCAD. The mechanical hardware consists of a double helical motor gear, geared revolving camera track, guide clip, bearing holder, static lower track, stepper holder, and support feet.

We use a 3D printer to generate the mechanical parts of the 3D scanner, i.e., the double helical motor gear, the geared revolving camera track arcs, the guide clip, the bearing holders, the support track, the stepper holder, and support feet. As shown in Fig. 2, we design the parts using OpenSCAD, an open-source application for programmatic 3D model creation. By programmatically modeling the parts, we provide users with the ability to render scanners of varying dimensions for scanning objects of varying size. Our approach allows the user to change the radius, width, and heights of the upper and lower tracks, the diameter of the friction bearings, the elevation of the platform riders, and the width of the stepper motor by changing the variables in our OpenSCAD program. We modularize the camera tower to allow the user to re-position the camera to image objects of varying heights by adding or removing tower pieces.

4 Software Interface

The software interface for our 3D scanner consists of a MATLAB script on the host computer which is used to set the capture parameters, and a Python script on the Raspberry Pi which is used to control the motion of the scanner and to capture images of the object. Our MATLAB control script allows the user to specify the IP address of the Pi, the capture name, and the number of capture images. We transmit the capture parameters via Secure Shell (SSH) to the Pi.

We then remotely call the Python script to control the motion of the camera. The Python script sends motor control commands to the stepper to execute a

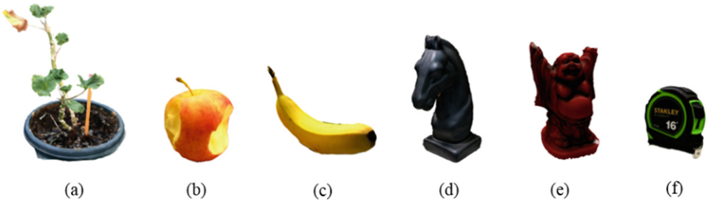


Fig. 3. Reconstructed 3D meshes of objects for natural objects such as (a) a potted plant, (b) a bitten apple, and (c) a banana, and man-made objects such as (d) a horsehead, (e) a Buddha statue, and (f) a tape measure scanned using our 3D scanner.

sequence of small rotations. After each rotation we capture and store an image to the Raspberry Pi before making the next rotation. We continue this process until the camera makes a complete revolution around the object, after which the motor returns to its starting point. We send a signal from the Raspberry Pi to the host computer indicating termination of the Python script. We then automatically transfer each image from the Raspberry Pi to the scanner control computer. We manually import the image set into Autodesk ReMake and generate a textured 3D mesh of the object [1]. In Fig. 3, we show the results of automatically capturing dense images of a variety of natural and man-made objects using our 3D scanner and reconstructing them using Autodesk ReMake.

5 Discussion and Future Work

We provide a 3D scanning system that captures delicate objects, such as potted plants and fruit, without impacting the integrity of the object of interest. Our 3D scanner is designed programmatically using OpenSCAD to allow users to quickly change the dimensional properties without needing to remodel the parts manually. To capture larger objects our system requires the user to add additional camera towers, or reprint the mechanical components. In future we will remove these constraints by designing a system that drives the camera independently of the camera track, and making the camera tracks interchangeable by using interlocking straight and curved pieces.

Acknowledgements. This work was partially supported by the National Science Foundation (NSF) grant #1730183.

References

1. Autodesk ReMake. http://remake.autodesk.com/about
2. Matter and form. https://matterandform.net/scanner
3. Baraff, D., Witkin, A.: Dynamic simulation of non-penetrating flexible bodies. In: SIGGRAPH (1992)
4. Baran, I., Popović, J.: Automatic rigging and animation of 3D characters. ACM Trans. Graph. **26**(3) (2007)

5. Borghese, N.A., Ferrigno, G., Baroni, G., Pedotti, A., Ferrari, S., Savarè, R.: Autoscan: a flexible and portable 3D scanner. CGA **18**(3), 38–41 (1998)
6. Butkiewicz, T.: Low-cost coastal mapping using Kinect v2 time-of-flight cameras. In: Oceans (2014)
7. Butler, D.A., Izadi, S., Hilliges, O., Molyneaux, D., Hodges, S., Kim, D.: Shake'n'sense: reducing interference for overlapping structured light depth cameras. In: SIGCHI (2012)
8. Callieri, M., Fasano, A., Impoco, G., Cignoni, P., Scopigno, R., Parrini, G., Biagini, G.: Roboscan: an automatic system for accurate and unattended 3D scanning. In: 3DPVT (2004)
9. Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., et al.: The digital michelangelo project: 3D scanning of large statues. In: CGIT (2000)
10. Tong, J., Zhou, J., Liu, L., Pan, Z., Yan, H.: Scanning 3D full human bodies using kinects. IEEE TVCG **18**(4), 643–650 (2012)