A comparison of open source libraries ready for 3D reconstruction of wounds

Kumar, Syamantak, Jaglan, Dhruv, Ganapathy, Nagarajan, Deserno, Thomas

Syamantak Kumar, Dhruv Jaglan, Nagarajan Ganapathy, Thomas M. Deserno, "A comparison of open source libraries ready for 3D reconstruction of wounds," Proc. SPIE 10954, Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications, 109540A (15 March 2019); doi: 10.1117/12.2513411

SPIE.

Event: SPIE Medical Imaging, 2019, San Diego, California, United States

A comparison of open source libraries ready for 3D reconstruction of wounds

Syamantak Kumar[a], Dhruv Jaglan[a], Nagarajan Ganapathy[b, c], Thomas M. Deserno[*c]
[a]Indian Institute of Technology Bombay, Powai, Mumbai, Maharashtra, India - 400076; [b]Indian Institute of Technology Madras, IIT P.O., Chennai, India - 600036; [c]Peter L. Reichertz Institute for Medical Informatics, TU Braunschweig and Hannover Medical School, Muehlenpfordtstr. - 23, Braunschweig, Germany - 38106

ABSTRACT

Quantitative assessment is essential to ensure correct diagnosis and effective treatment of chronic wounds. So far, devices with depth cameras and infrared sensors have been used for the computer-aided diagnosis of cutaneous wounds. However, these devices have limited accessibility and usage. On the other hand, smartphones are commonly available, and three-dimensional (3D) reconstruction using smartphones can be an important tool for wound assessment. In this paper, we analyze various open source libraries for smartphone-based 3D reconstruction of wounds. For this, point clouds are obtained from cutaneous wound regions using Google ARCore and Structure from Motion (SfM) libraries. These point clouds are subjected to de-noising filters to remove outliers and to improve the density of the point cloud. Subsequently, surface reconstruction is performed on the point cloud to generate a 3D model. Six different mesh-reconstruction algorithms namely Delaunay triangulation, convex hull, point crust, Poisson surface reconstruction, alpha complex, and marching cubes are considered. The performances are evaluated using the quality metrics such as complexity, the density of point clouds, the accuracy of depth information and the efficacy of the reconstruction algorithm. The result shows that the point clouds are able to perform 3D reconstruction of wounds using open source libraries. It is found that the point clouds obtained from SfM have higher density and accuracy as compared to ARCore. Comparatively, the Poisson surface reconstruction is found to be the best algorithm for effective 3D reconstruction from the point clouds. However, research is still required on the techniques to enhance the quality of point clouds obtained through the smartphones and to reduce the computational cost associated with point cloud based 3D-reconstruction.

Keywords: Wound, segmentation, point clouds, 3D reconstruction, smartphones, assessment, algorithms

1. INTRODUCTION

Accurate wound assessment is a critical component of effective wound management. It ensures correct diagnosis and supports an efficient treatment procedure. Existing techniques for wound assessment performed by clinicians are manual, inefficient, and expensive. Typically, wound area measurement is performed by tracing the boundary of the wound on special transparent sheets[1]. Though these traditional methods are simple to implement, they have major drawbacks in terms of limited accuracy. In most cases, these methods involve direct contact with the patient, exposing the patient to the risk of an infection. Further, these measurements fail to capture the qualitative information such as color and texture of the wounds. Moreover, these methods do not consider the wound-depth which may be useful for an accurate assessment. Therefore, there has been a necessity for effective techniques to overcome these challenges[2, 3].

Specialized hardware has been developed to overcome the problems faced by manual wound assessment[2]. However, these devices lack portability and their high-cost makes them infeasible for use by patients individually. Various techniques have been developed to measure the quantitative characteristics of chronic wounds such as color, size, the rate of healing using two-dimensional (2D) image along with the paper ruler[3] and color card[4]. These imaging techniques provide a fast, convenient segmentation and measurement tool for the assessment of treatment. However, it does not provide the depth

* Corresponding author: Thomas M Deserno
Email: thomas.deserno@plri.de

Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications,
edited by Po-Hao Chen, Peter R. Bak, Proc. of SPIE Vol. 10954, 109540A
 · doi: 10.1117/12.2513411

and volume information of the wounds for healing assessments. Thus, it highlights the need for inexpensive and portable devices for three-dimensional (3D) imaging of skin lesions.

Recently, a new generation of 3D reconstruction methods have evolved. These approaches include stereo imaging and depth, direction sensing devices for 3D surface reconstruction. For example, researchers at the University of Zagreb have used direction sensor measurements from accelerometer to develop a time-efficient and accurate 3D registration method[5]. In another study, Li et al.[6] have designed a multi-angle 3D reconstruction system-using laser imaging and the Structure from Motion (SfM) algorithm. SfM algorithm computes point clouds from the set of two-dimensional images of similar scene. The point clouds based 3D reconstruction model from multiple views of camera has been an active area of research. Lin et al.[7] have developed a prototype system for capturing rotational stereo images of an object. Further, they have also developed algorithms to calculate texture maps. In an interesting study, Newcombe et al.[8] have proposed a real-time dense surface mapping and tracking technique using the Kinect sensor.

Though, these devices are portable and cost efficient, they still require special algorithms for automated reconstruction. Recently, wound assessment using low-cost mobile devices has been an intense topic of research. Poisei et al.[9] have developed a cloud-based collaborative 3D reconstruction method using an incremental SfM pipeline. Similarly, Ondruska et al.[10] presents a system with real-time volumetric surface reconstruction using mobile phones. In an interesting study, Google introduced the project Tango[11] for robust 3D reconstruction using the depth and infrared sensor data. Dastjerdi et al.[3] have used the project Tango for the analysis of skin lesions using point clouds. The author has used point clouds to analyze lesions even at bent surfaces. In order to overcome the hardware dependencies and limited access of project Tango, Google has launched the ARCore[12] toolkit for smartphones to generate point clouds.

In this work, we aim to determine the feasibility of using ARCore and SfM techniques for performing 3D wound reconstruction using point clouds. Our proposed framework is closest in spirit to the one followed by Sirazitdinova et al[13], similarly designed for developing efficient wound assessment techniques. In this study, two different approaches namely ARCore and SfM have been explored for automated point cloud generations. Further, the ARCore provides access to the point cloud for debugging purposes; thus, an attempt has been made to use this point cloud for 3D surface reconstruction.

2. MATERIAL & METHODS

2.1 Proposed framework

The schematic pipeline of the proposed framework is shown in Figure 1. Three different approaches have been followed for extracting point clouds from 3D objects. In the first approach, ARCore is used to obtain the point clouds for each frame by using anchors in space. For this, the mobile device is first moved around the object of interest to record point cloud data. Later, these tracked point clouds are stored separately. In the other two approaches, SfM is used to generate the point cloud from 2D image sequences around the object. In the second approach, images captured from the ARCore application (640 x 480-pixel resolution) are provided as input to the SfM pipeline, whereas in the third approach images captured from the device camera (4032 x 4032-pixel resolution) are used. These obtained point clouds are subjected to six different 3D reconstruction algorithms namely, (i) Delaunay Triangulation, (ii) Convex Hull, (iii) Point Crust, (iv) Poisson surface reconstruction, (v) Alpha Complex, and (vi) Marching Cubes are considered for surface reconstruction. The performance of the algorithms is compared using quality metrics such as complexity, the density of point clouds, the accuracy of depth information, and the efficacy of the reconstruction algorithm.

2.1.1 Google ARCore

ARCore is a Google API for building augmented reality experiences. Using different APIs, ARCore enables the phone to sense its environment, understand the world and interact with information. It can identify key points, detect flat surfaces, like a table or the floor, and also estimate the average lighting in the scenes. Using phone camera, ARCore uses three key capabilities namely, motion tracking, environmental understanding, and light estimation to integrate virtual content with the real world

a) Motion tracking

It allows the phone to understand and track its position relative to the world. As the phone moves through the world, ARCore uses a process called concurrent odometry and mapping (COM), to understand the relative position of the phone in the world. Further, it detects visually distinct features named feature points in the captured camera image and uses these points to compute its change in location. Here, the visual information is combined with inertial measurements from the device inertial measurement unit (IMU) to estimate the pose (position and orientation) of the camera relative to the world over time.

b) Environmental understanding

It allows the phone to detect the size and location of various surfaces namely, horizontal, vertical and angled surfaces. It distinguishes various surface like the ground, a coffee table or walls by detecting clusters of feature points that appear to lie on common horizontal or vertical surfaces. Here, the detection of planes helps to improve the 3D understanding of the environment.

c) Light estimation

It allows the phone to estimate the environment's current lighting conditions. It provides the average intensity and color correction of a given camera image which can be used for the accurate measurement of color and texture of the surface. Accurate measurement of color and texture is important to clearly discriminate the foreground and background objects.

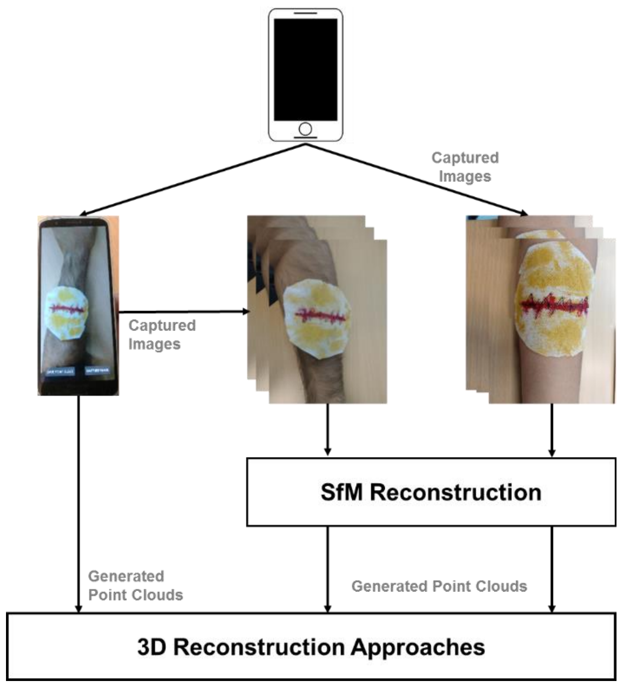

Figure 1: The schematic pipeline of the proposed approach.

2.1.2 Theia SfM

The Theia library[14] is a C++ library that includes a state-of-the-art SfM pipeline. It has a vast collection of multi-view geometry tools and algorithms that utilize image and video inputs to create high quality 3D reconstructions. It contains incremental and global SfM pipelines which follows a standard sequential SfM procedure[15]. It starts by extracting feature descriptors (in parallel) at salient points within images. After features are extracted, images are matched to determine a two-view geometry between the images that observe the same scene. Generally, Theia uses the extremely fast cascade hashing method[16] to compute image matches with multiple threads. These geometrically verified two-view matches are used to estimate camera poses with global motion averaging schemes. Finally, the obtained 3D points are triangulated in parallel and refined with a nonlinear optimization.

2.2 Extraction of point cloud

The point clouds are points in 3D space that represent the features of the object. For wound measurements, dense point-clouds are required[5]. ARCore estimates the position of the device in space and detects point clouds in every frame in the world coordinate system. These point clouds are computed for each specific frame of the video. It uses anchors as a tool to track a position of interest over time and accumulate information over each frame. Here, an anchor is made at each 3D point detected by ARCore. Further, care was taken to avoid duplicate anchors at any point. In order to calculate the RGB values for each point, images at each frame are considered and fed as an input to transform the 3D coordinates to 2D using the view and projection matrices. Finally, the point cloud is saved in the coordinate system of the last frame on the mobile device as a Polygon file format (*.ply). An overview of the detailed algorithm for extracting and storing the point-clouds through ARCore is shown in Figure 2. In the other two approaches, the Theia library was used to extract the point clouds from images. In one approach, images stored for specific frames in ARCore is provided as input to SfM for point clouds generation. In second approach, multiple sets of images of the object under consideration were taken using the mobile phone and passed as input to the SfM algorithm for point cloud generation. Though SfM generates point clouds using multiple set of images, the SfM library does not provide RGB information inherently. In this study, RGB information was extracted from the input images and the feature points on the images, matched for individual 3D points.

Objective:

Extract a 3D point cloud P from a series of images S (as a continuous video or as separate images) and perform surface reconstruction to construct a 3D-model M

Initialization

- Coordinate list (optional): To visualize the points captured in each frame for debugging
- Anchor list: Store the list of anchors being made at each feature point
- Color list: Store the color of each feature point being captured cumulatively over each frame

Loop over each frame of interest

- **Capture point-cloud P**

  ARCore provides access to the point cloud being captured in each frame for debugging purposes

- **Acquire camera image I**

  The image is captured in the YUV_420_888 format. ARCore does not provide detailed access to the image and so the conversion to an RGB bitmap needs to be performed externally. This will later be used for providing color to each 3D point captured.

Loop over all the point captured in the frame

- Get the projection and view matrices for frame and extract (x, y, z) coordinates from each point in the point cloud
- Form the pose matrix using the (x, y, z) coordinates for the translation and assume no rotation
- Make an anchor at the above pose to track the point in 3D as the device moves
- Use the 3D world coordinates, the projection matrix and the view matrix to extract the 2D screen coordinates for the point
- Once the 2D coordinates are known, extract the (R, G, B) values at the 3D point using the image captured I for the frame
- Finally, store the anchor created in the anchor list, the (R, G, B) values in the color list, and (optionally) the 2D screen coordinates in the coordinate list. Care must be taken to not store duplicate anchors for the same point.

Click "Save Point Cloud" button

Using the anchor-list and the color-list, write out the entire data of the point-cloud in your desired format and save the file on the device

Surface reconstruction

Once the point cloud has been extracted, the entire data is passed on to the 3D surface reconstruction pipeline to generate smooth surfaces using the point-cloud data.

Figure 2: Overview of the algorithm for mobile based surface reconstruction using ARCore

2.3 Imaging equipment & setup

For the experiments, Moto G6 Plus mobile phone (Motorola, USA) with a dual camera (12 MP camera) have been used. However, the benefits of the dual camera option are not supported by ARCore. To test the robustness of different point cloud libraries and 3D reconstruction algorithm, sample wound patches are mimicked using white cloth and colors. Further, other samples such as human skeleton models, human hand, and the face are also used for the study.

2.4 3D surface rendering methods

Generation of a 3D model from an unorganized set of points is a well-studied problem in computer graphics. It allows fitting of scanned data, filling of surface holes, and re-meshing of existing models. In literature, several algorithms have been reported for the 3D Surface rendering of the scanned point cloud. In this study, two separate libraries, namely Meshlab[17] and the Point Cloud Library[18] (PCL) are used to implement 3D reconstruction on the extracted point clouds. Further, the performance is evaluated quantitatively and qualitatively using the computational cost and accuracy of the individual algorithms, respectively. The set of methods used for surface rendering are summarized below:

a) Marching cube surface reconstruction
It applies the marching cubes algorithm to reconstruct the surface. Initially, the volume data is partitioned into cubes. The algorithm redefines configurations for every partition to the vertices, which are in the surface or on the surface. The surfaces of each cube are decided according to the 15 surface configurations and finally the surfaces of each given cube are connected to the model[19].

b) Poisson surface reconstruction
It creates watertight surfaces from oriented point sets. In this method, oriented points are first transformed to a continuous vector field in 3D space and then a scalar function is approximated. Finally, an appropriate iso-surface is extracted.

c) Delaunay triangulation based surface reconstruction
In this algorithm, for a given set of discrete points P, a triangulation DT(P) is performed such that no point in P is inside the circumcircle of any triangle in DT(P). Here, extension of the Delaunay triangulation algorithm to 3D[20] has been proposed to perform surface reconstruction.

d) Alpha complex surface reconstruction
This algorithm is an example of surface-oriented cell selection. Initially, the input points are decomposed using Delaunay triangulation. In the second step, tetrahedra, triangles and edges of the Delaunay triangulation are removed using α-balls with radius α. Later, the triangles from α-shape are extracted using a set of rules defined as

- If a sphere with radius α passes through all three points of a triangle, and
- If the sphere doesn't contain any other point of the input set, the triangle belongs to the surface.

e) Convex hull surface reconstruction
This algorithm tries to fit a convex surface to a given set of points P. Initially, the points with minimum and maximum x coordinates are identified. Further using the line formed by the two points, the set is divided in two subsets of points, which will be processed recursively. Then, the point on one side of the line, forms a triangle whereas the points lying inside the triangle cannot be a part of the convex hull and therefore will be ignored in the further steps. These steps are further repeated for the remaining set of points to form 3D surfaces.

f) Point crust surface reconstruction
The radius of granularity is fed as an input parameter to the algorithm. Initially, the reconstruction algorithm iterates over the available set of triangles in the 3D point cloud. In order to find out the "surface-triangles", two spheres with the initially defined radius are calculated such that both spheres pass the three points of the triangle. If one of the spheres has no other points inside, then the triangle is a "surface-triangle" and is cached for the output.

3. RESULTS AND DISCUSSIONS

The time for isolation of point clouds along with the size of the point clouds has been recorded in Table 1. For the performance analysis, computational time and the point cloud density of different combinations is computed. Results show that the processing and detection of point clouds using ARCore is efficient and relatively fast as compared to its SfM counterpart. Since, ARCore provides access to the point cloud only for debugging purposes, the accuracy of the point cloud extracted using ARCore is considerably less as compared to the point cloud extracted using the SfM technique. The point clouds extracted using the SfM approach are found to be denser and clear. The point-clouds obtained using the three techniques

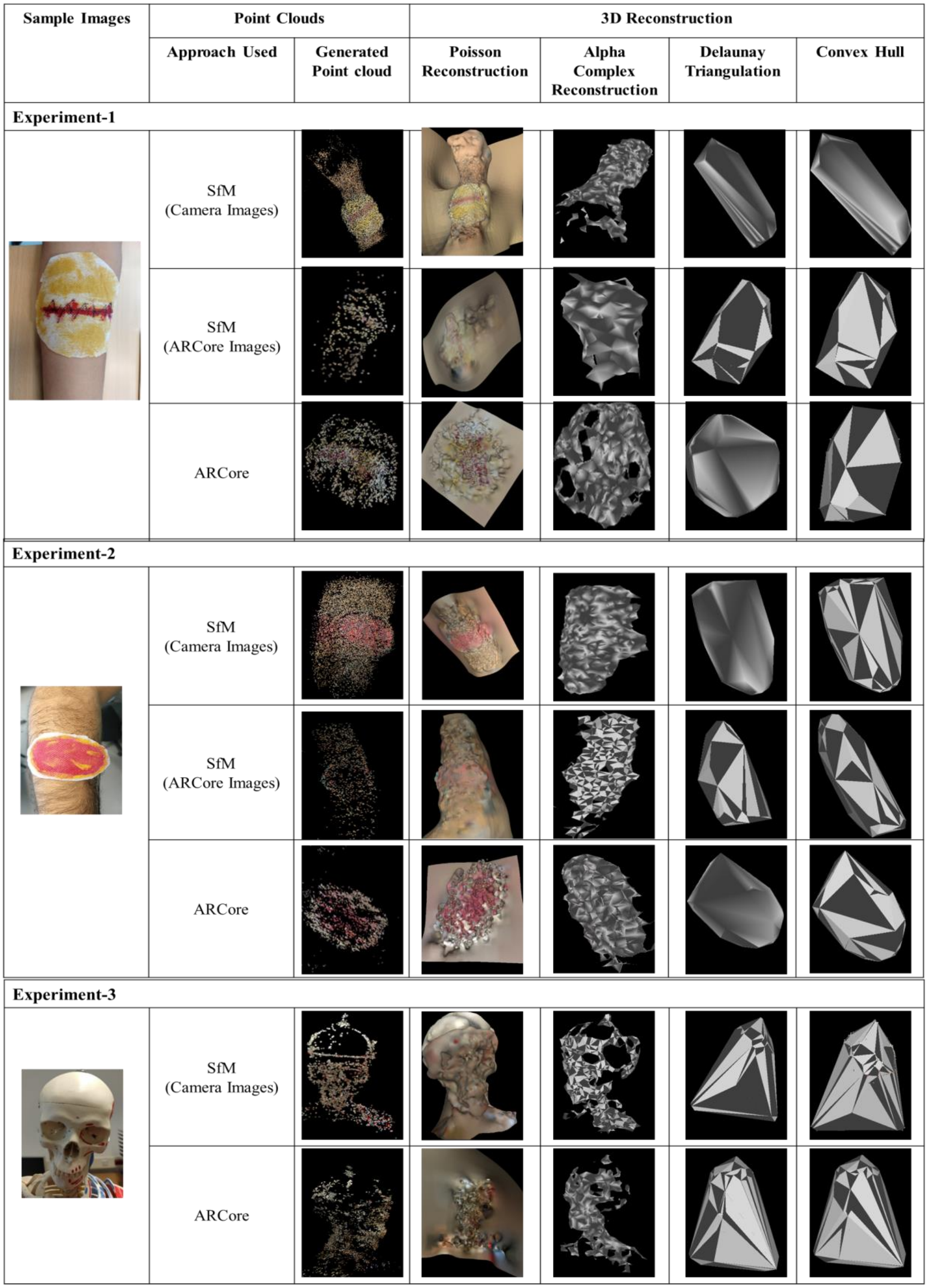

Figure 3: Point cloud distribution and 3D reconstructed images for three different sample inputs.

Method Used	ARCore Point Cloud		SfM Technique using ARCore Images			SfM Technique using Camera Images		
Parameter Recorded	**Time [s]**	**Size [No. of points]**	**Time [s]**	**No. of Images**	**Size [No. of points]**	**Time [s]**	**No. of Images**	**Size [No. of points]**
Experiment-1	~ 2	30,322	16	45	455	701	45	14,636
Experiment-2	~ 2	28,100	22	45	1,506	387	45	15,160
Experiment-3	~ 2	30,345	-	-	-	530	45	3,572

Table 1: Time recorded for processing above models using the above-mentioned techniques

(Section 2.2) and their corresponding 3D reconstructions using the six algorithms (Section 2.3) are displayed in Figure 3. Experiments are carried out using three different samples. For each experiment, observations are made based on the method used for point-cloud extraction and the algorithms used for 3D reconstruction. The first column provides a sample image for the experiment while the second column displays the point clouds captured using different techniques. The subsequent columns display the result of applying 3D reconstruction on the point clouds in different combinations.

The results demonstrate that the SfM technique using camera images results in the best quality of point clouds. SfM achieves outstanding performance on large scale datasets both in terms of efficiency and accuracy. The point cloud extracted by applying SfM technique on images acquired using ARCore is sparse and not suitable for 3D reconstruction. The difference in the quality of point clouds resulting from SfM on camera images and the ARCore images can be attributed to the fact that the quality of images obtained from ARCore is restricted to a (640 x 480) pixels resolution whereas the images taken from the camera have a resolution of (4,032 x 3,024) pixels. It is observed that the ARCore-based 3D points are more efficient as compared to the SfM technique in real-time. However, the point clouds obtained from ARCore are not accurate enough to perform reconstruction due to error in the depth information of the point cloud. This is due to the fact that ARCore, as of now, provides access to the point-cloud in each frame for debugging purpose only.

Based on our experiments, Poisson surface reconstruction algorithm is found to be the best for 3D surface reconstruction. Alpha-Complex surface reconstruction also leads to an accurate 3D reconstruction; however, it does not capture the details of color at each point. Delaunay Triangulation and Convex Hull algorithms fail to give satisfactory results and instead just cover the set of points in a three-dimensional shape. Indeed, a denser and accurate point cloud would be required to obtain accurate results using the other algorithms. Our experiments therefore show that the SfM technique using camera images combined with Poisson surface reconstruction result in the best quality of 3D reconstruction at the cost of time.

4. CONCLUSION

In this work, two different techniques namely ARCore and SfM are analyzed for point clouds generation using smartphones. Further, experiments were carried out to identify the best combination of methods for point cloud and 3D reconstruction. For this, time and density of the point clouds was recorded to analyze the efficiency of the point-cloud acquisition methods. The results demonstrated that there is a trade-off between accuracy and speed of reconstruction on the smartphone devices. Our experiments show that the SfM technique using camera images combined with Poisson surface reconstruction result in the best quality of 3D reconstruction at the cost of time. The quality of the point clouds obtained was insufficient for performing reliable surface reconstruction of wound samples, but further research is required for multiple backgrounds before it could be analyzed for applications in clinical trials. Furthermore, robust 3D reconstruction algorithms aimed towards smartphones could be a possible future task. Integrating the use of dual-camera for providing accurate depth information is seen promising for further research, since one major weakness of our approaches is the limited accuracy of depth estimation.

REFERENCES

[1] Enoch S, Grey J, Harding K. Non-surgical and drug treatments. BMJ. 2006;332(7546):900-903.

[2] Zhou A. A survey of optical imaging techniques for assessing wound healing. Int J Intell Control Syst. 2013;17(3):79-85.

[3] Mirzaalian Dastjerdi H, Töpfer D, Rupitsch S, Maier A. Measuring Surface Area of Skin Lesions with 2D and 3D Algorithms. International Journal of Biomedical Imaging. 2019; 2019:1-9.

[4] Bala S, Sirazitdinova E, Deserno T. Colorimetric calibration of wound photography with off-the-shelf devices. Proc SPIE. 2017. p. 10136 - 10136 - 9.

[5] Pribanić T, Petković T, Đonlić M. 3D registration based on the direction sensor measurements. Pattern Recognition. 2019; 88:532-546.

[6] Li S, He Y, Li Q, Chen M. Using Laser Measuring and SFM Algorithm for Fast 3D Reconstruction of Objects. JRLR: Journal of Russian Laser Research. 2018;39(6):591-599.

[7] Lin H, Subbarao M, Park S. Complete 3D model reconstruction from multiple views. Proc SPIE 4567. 2002. p. 4567 - 4567 – 11.

[8] Newcombe R, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison A et al. Kinect Fusion: Real-time dense surface mapping and tracking. Proc. IEEE ISMAR. 2011. p. 127-136.

[9] Poiesi F, Locher A, Chippendale P, Nocerino E, Remondino F, Van Gool L. Cloud-based collaborative 3D reconstruction using smartphones. Proc. CVMP, ACM. 2017. p. 1-9.

[10] Ondruska P, Kohli P, Izadi S. MobileFusion: Real-Time Volumetric Surface Reconstruction and Dense Tracking on Mobile Phones. IEEE Trans Vis Comput Graph. 2015;21(11):1251-1258.

[11] ATAP Project Tango, Google Developers, 2014[internet].

[12] ARCore, Google Developers, 2018 [internet], https://developers/google.com/ar/discover/ .

[13] Sirazitdinova E, Deserno T. System design for 3D wound imaging using low-cost mobile devices. Proc SPIE 10138. 2017. p. 10138 - 10138 - 7.

[14] Sweeney C. Theia Multiview Geometry Library: Tutorial & Reference [internet], http://theia-sfm.org.

[15] Snavely N, Seitz S, Szeliski R. Photo tourism: exploring photo collections in 3D. ACM Trans Graph. 2006;25(3):835-846.

[16] Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H., Fast and accurate image matching with cascade hashing for 3D reconstruction, Proc. IEEE CVPR, 1–8 (2014).

[17] Cignoni P, Callieri M, Corsini M, Dellepiane M. MeshLab: An Open-Source Mesh Processing Tool. Eurographics Italian Chapter Conference. 2008; 73:129-136.

[18] Rusu, R., Cousins, S., 3D is here: Point cloud library (PCL), Proc. IEEE ICRA, 1–4 (2011).

[19] Jin J, Wang Q, Shen Y, Hao J. An Improved Marching Cubes Method for Surface Reconstruction of Volume Data. Proc. WCICA. IEEE; 2006. p. 10454–10457.

[20] Cazals F, Giesen J. Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms. RR-5393, INRIA. 2004. p. 42.