3D-Quantify: A workflow for generating Volume of an object using 2D images and Structure from Motion

Ahsan Ali Khoja
Department of Computer Science,
Earlham College,
Richmond, IN, 47374.
aakhoja15@earlham.edu

ABSTRACT

Volume calculations and survey methods are needed to detect changes in the flow of glaciers over time and to calculate the capacity of gullies. The current solutions are either labor-intensive, expensive, or both. Such solutions make the research process slower and costly. This paper provides an open solution called 3D-Quantify, a framework for volumetric calculations using Structure from Motion (SfM). SfM is a cost-efficient way for reconstructing a 3D model of an area by stitching 2D images. These 3D models can then be scaled to extract features, including the volumetric measure and capacity of the area. This project explores existing solutions, including Cloud Compare and WebODM, for the individual components, focusing on the volume calculation algorithm for refinement. The data from our approach is compared with volumetric data (i) measured using manual calculations and estimations, and (ii) calculated using existing volumetric measurement software. We also compare the cost of equipment, prior experience required to use the workflow, and human time and input needed for the workflow to function.

KEYWORDS

Structure from Motion, Multi-view Stereo Reconstruction, UAV, Volume Calculation.

1 INTRODUCTION

Surveying areas is a crucial part of Field Science research. Surveying is an umbrella term that encompasses all geometrical and positional measurement of objects/point. Doing so efficiently and accurately matters, as it makes the analysis faster, cheaper and stronger. One such method of surveying includes collecting images of the survey site and identifying areas of interest within the site.

Surveys also serve as a way to calculate the capacity of a river channel, map changes in a glacier, or soil erosion [9]. Such a survey, in the past, has required Laser scanning techniques that map every object in the field as a set of points with distance from a certain elevation, creating a Point Cloud. Even though this method is efficient and accurate, the cost of equipment acts as a barrier. Additionally, such Laser-based systems require specific training and skills [9] However, the Point Cloud of an area is a valuable resource, as it can be used to generate accurate measurements and distances between two points, or even find the coverage volume/capacity of the city.

Structure from Motion (SfM) provides a way to generate these Point Clouds with the use of a basic camera. SfM provides an approximate solution to the problem of surveying areas using images[16]. SfM is a technique that generates three-dimensional models from a set of images. It does so by recognizing features across images captured from different angles and perspectives. This leads to the generation of a Point Cloud, which as defined above, treats every object as a set of points with distance from the camera. By transferring the burden from data collection to data processing, we can make the research more effective and efficient.

The existing method of generating volume from the Point Cloud is by converting it into Triangulated Irregular Network (TIN) and creating a triangular mesh. However, that approach does not work on open objects and areas, including river channels and gullies. This paper focuses on the problem of calculating the volume of open objects.

We propose a solution called 3D-Quantify. 3D-Quantify is a framework that takes images as inputs, converts them to a scaled Point Cloud via SfM and the Scalar algorithm, and calculates the volume from that Point Cloud by using the slicing method [17]. The slicing method divides the Point Cloud into slices of uniform height, projects those slices onto an x-y plane, and estimates the partial volume of each slice by calculating its area in a projection plane. It must, however, be noted that for the scope of this project, we have implemented the slicing method of volume calculation, and proposed an algorithm for scaling the Point Clouds. The implementation of scaling algorithm and consequently the development of software that makes these moving parts work as a single framework have been left for future work.

Our implementation of the volume calculation algorithm differs from the slicing method proposed by Zhi et al. [17] in the way the area covered by the polygon is estimated. Both the approaches rely on the Shoelace algorithm for the area calculation, which requires the points to be in counter-clockwise order. Our sorting strategy (see section3) is similar to that suggested by Zhi et al. [17] but differs in how we select our starting point, and our approach towards dealing with multiple points with the same angle (see section 2).

So, the major contributions of this project are:

(1) Proposal for the 3D-Quantify as a framework.
(2) Design for the Scalar algorithm, and
(3) Design and Development of an algorithm for sorting 2D points in a CCW order,
(4) Design and Implementation of the VolCalc algorithm,

It shall be noted that the implementation of the Scalar algorithm and software that would connect the moving parts is out of the scope of this project.

The rest of the paper is structured as follows. Section 2 covers the background for the research, highlighting the areas and concepts most relevant to this project. Section 3 talks about the workflow and methodology, explaining the algorithms used and the reasoning behind choices made. Section 4 outlines the testing and evaluation scheme, while also displaying the achieved results. Section 5 lays the future direction this research will move in, and section 6 concludes this paper.

2 BACKGROUND

This section elaborates on the existing research in SfM, focusing on its origins, existing methods used in the field to enhance data collection, and current areas of application. It also highlights the areas and concepts relevant to this study.

2.1 Structure from Motion

As highlighted in section 1, 3D-Quantify employs SfM as a first step and solution for the generation of the Point Cloud. SfM was developed by Westobi et al. as a low-cost, effective tool and method for surveying [16]. They introduced it as an alternative to existing expensive photogrammetric methods. Traditional methods used the 3D location of the camera and the 3D location of known ground control points. SfM, on the other hand, does not require the use of camera locations and solves the 3D construction problem by using feature matching algorithms. SfM workflow can be better understood by dividing it into a 3 step process, namely (i) feature extraction and matching, (ii) camera motion estimation, and (iii) 3D model recovery from estimated motion and features. All these processes are applied to the set of 2D images [12].

Westobi et al. also introduced methods used to create high-resolution digital elevation models (DEMs). They were able to show that vertical accuracy can be achieved using SfM even for areas with complex topography. As part of 3D-Quantify, SfM will take a set of images as its input and will return a 3D Point Cloud of the area. Carrivick et al. demonstrated that SfM could be applied in GeoSciences [4]. SfM, with multi-view stereo, can be used to detect elevation, position, and volumetric changes, by looking at multi-temporal data. They also claimed that SfM could produce Point Clouds that are comparable in density and accuracy to those produced by laser scans. Hence, by getting a similar density Point Cloud, with a reduction in cost and manual labor through the use of SfM, we can make the process more efficient without significant loss in accuracy.

2.2 The Shoelace Algorithm

After scaling the Point Cloud, we use the slicing method to calculate the area and volume of each slice. A well-known algorithm for calculating the area covered by a polygon is the Shoelace algorithm, also known as The Surveyor's Area Formula [3]. This formula takes as input the list of coordinates of points in a 2D plane and returns the area enclosed as its output. It does so by dividing the polygon into triangles and calculating their areas. The image below shows how the algorithm works in creating the area from the coordinates. The shoelace algorithm does come with a limitation. It expects the list of coordinates to be sorted in counter-clockwise (CCW) order. Since the points we have are in random order, we have created an algorithm to sort the points in the CCW order (see section 3.2).

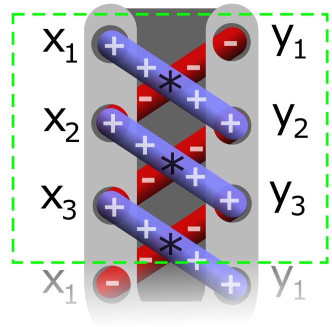

Figure 1: Shoelace algorithm[2]

2.3 Related Applications

Some major examples of SfM can be seen in the following areas:

- A cost-effective UAV based method to investigate calving dynamics of a large outlet glacier draining the ice sheet of Greenland [13].
- Creating a Digital Elevation Model (DEM) of a river environment using digital photographs [6].
- Collecting accurate 3D volumetric data in difficult to access gully systems using SfM in place of LiDAR [9].
- Automate the accurate generation of georeferenced mosaics from Unarmed Aerial Vehicle (UAV) imagery [15].

3 METHODOLOGY & WORKFLOW

This section focuses and elaborates on the design and implementation details of the workflow. Figure 2 shows the structure of the workflow for 3D-Quantify. The following are the major components that form the workflow:

a. **SfM** component will use images from the camera as its input and convert them to a 3D structure. The output from this component is a Point Cloud (PC). SfM component is primarily based on pre-existing software, including WebODM [5] (installed on Cluster servers) and Cloud Compare.
b. **Scaler** component uses the output PC from SfM as its input, along with the Ground Control Point (GCP) locations to produce a scaled Point Cloud.
c. **VolCalc** uses the scaled PC as its input and computes the volume contained in the closed and scaled object. This algorithm produces the final output regarding volume.

3.1 VolCalc algorithm

VolCalc algorithm is a key contribution of this project. This algorithm can be divided into subparts, as can be seen in Figure 3. We calculate the volume of the object by calculating the volume of each horizontal slice of height h and summing the respective results. Refer to Figure 4 and 5 for the Point Cloud slicing. The 3D slice is projected onto the cartesian plane, such that, a point (x_0, y_0, z_0) gets projected to (x_0, y_0). The set of projected points in the 2D plane can be treated as a polygon, which allows us to calculate the bounded area by using the shoelace formula (see section 2.2). The calculated area can then be multiplied by the height h to approximate the volume covered by each slice. Hence, by decreasing the value of h, we can get thinner slices, leading to closer approximations.

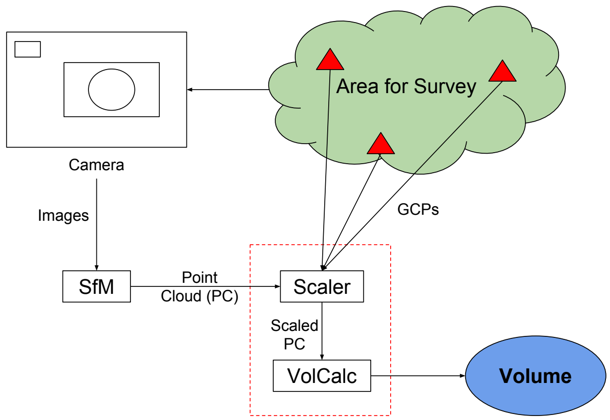

Figure 2: Framework of volume calculation using SfM

As described earlier, the Shoelace formula comes with two major constraints:

- It requires the points to be arranged in Clockwise or Counter-Clockwise order. To solve this problem, we have developed another algorithm that utilizes the polar coordinate system as an intermediary. The sorting algorithm is explained in section 3.2.
- Shoelace formula cannot be used in instances where the object intersects itself, as can be seen in Figure 6. As of now, our implementation of the algorithm does not solve this problem. In cases where we know of such intersecting objects, we either divide the Point Cloud into two separate clouds or increase the number of slices. Increasing the number of slices leads to fewer points being projected to the Cartesian plane, leading to lesser likelihood of a self-intersecting polynomial.

3.2 Sorting algorithm for Shoelace formula

As explained above, the Shoelace formula relies heavily on the order of points. To sort these random points, we have developed the following algorithm:

(1) The points are scanned to find the minimum and maximum values on the x and y axes. Call these values x_0, x_1, y_0, y_1. By averaging these values, we find a midpoint, call it $m_1 = (\frac{x_0+x_1}{2}, \frac{y_0+y_1}{2})$.
(2) All points are translated by m_1 such that m_1 gets transformed to the origin.
(3) The translated points are then converted to their polar forms (r, θ).
(4) The points are then sorted based on the θ, forming a counter-clockwise order.
(5) These sorted points are then converted back to their cartesian equivalents and translated back by m_1, with origin going to m_1 and keeping the sorted order. This step allows us to get the original points back in the sorted order.

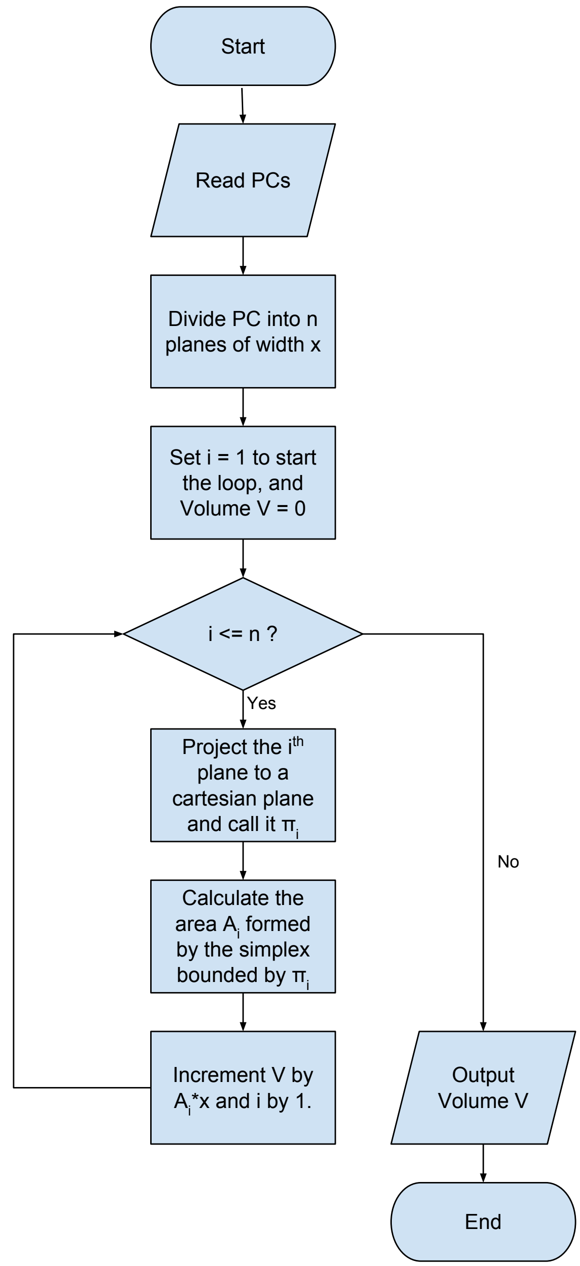

Figure 3: Flow Chart for VolCalc algorithm

It must be noted that, for the sake of simplicity, we are assuming the polygon after translation in step (2) does not have multiple points with the same value of θ.

3.3 Scalar algorithm

SfM transfers the load from hardware to the software side of automated data processing, reducing the reliance on the equipment

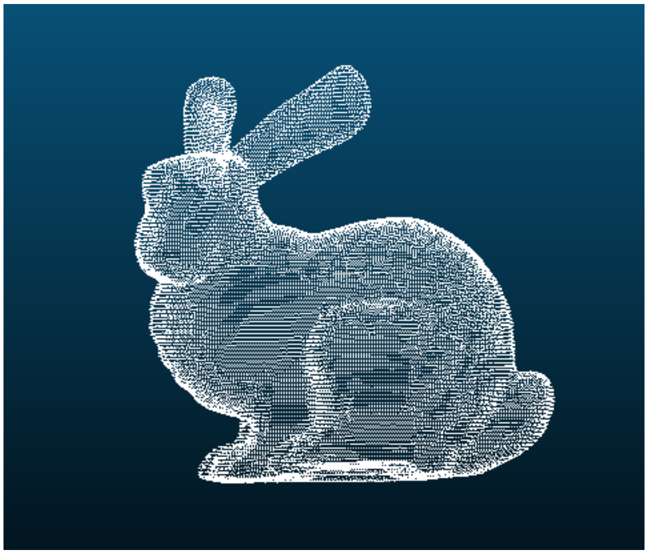

Figure 4: Stanford Bunny Point Cloud

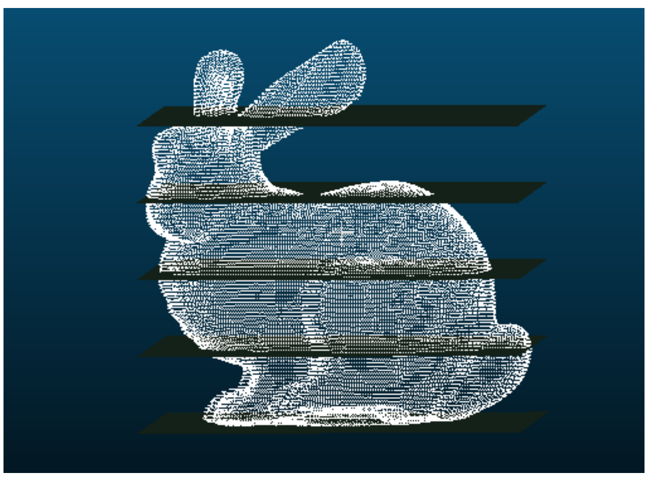

Figure 5: Stanford Bunny Point Cloud with slices

with which the data is collected. The Point Clouds produced by SfM need to be scaled to for accurate measurement of distances between two points in the survey area. This scaling highly relies on the availability of Absolute Ground Control Points, i.e., points with known latitude, longitude, and elevation in the survey area. We have proposed an algorithm for solving this problem with relative Ground Control Points, i.e., points with known latitude and longitude (which can be found via GPS), but with unknown elevation. Our algorithm, which will be implemented in the future, provides a way to use relative GCPs with accurate relative elevations for each GCP to scale the Point Cloud. The process can be seen in Figure 7. This algorithm requires the use of Pythagorean theorem in 3D to calculate the distance between two points, and use that alongside measured distance to find the elevation.

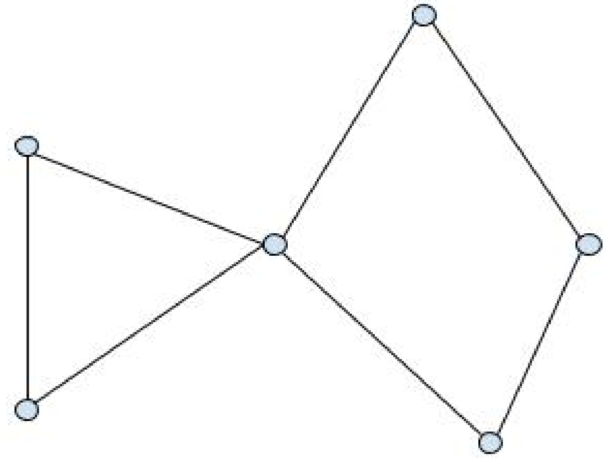

Figure 6: An example of polygon intersecting itself

4 RESULTS & EVALUATION

This section will highlight the test methodology, focusing on the results achieved, and the effectiveness of our program. For the experiments, we passed our Point Clouds through the system and varied the parameters to see changes in output. The subsections below will elaborate on the datasets used, explain the use of existing software, and finally present the result gathered.

4.1 Datasets

It can be seen from the VolCalc algorithm design that the computation does not produce accurate results for objects that do not have a single base, e.g., our algorithm would overestimate the volume occupied by a horse by treating the projection of 4 legs as the sides for a polygon. This means that only certain kinds of datasets could be used for testing purposes to measure the accuracy of the system. For this reason, we have used the following Point Clouds:

(1) Stanford Bunny, shown in figure 4.
(2) Sphere with a radius of 0.5 units. This will allow us to test how our algorithm works with curved surfaces.
(3) Cube with a side of 1 unit.
(4) Vase.

These Point Clouds were available in Polygon file format (.ply files) from the datasets compiled by Georgia Institute of Technology [11] and Alexiou et al. [1].

4.2 Benchmarks

To evaluate the performance of our algorithm, we have used Cloud Compare to find the volume of all the Point Clouds. This software requires the Point Cloud to be converted to a mesh, which can then be used to compute the volume. However, the values used here must be treated as approximations. For the Point Cloud of a Sphere, we can calculate the volume using the formula

$$V = \frac{4}{3}\pi r^3. \tag{1}$$

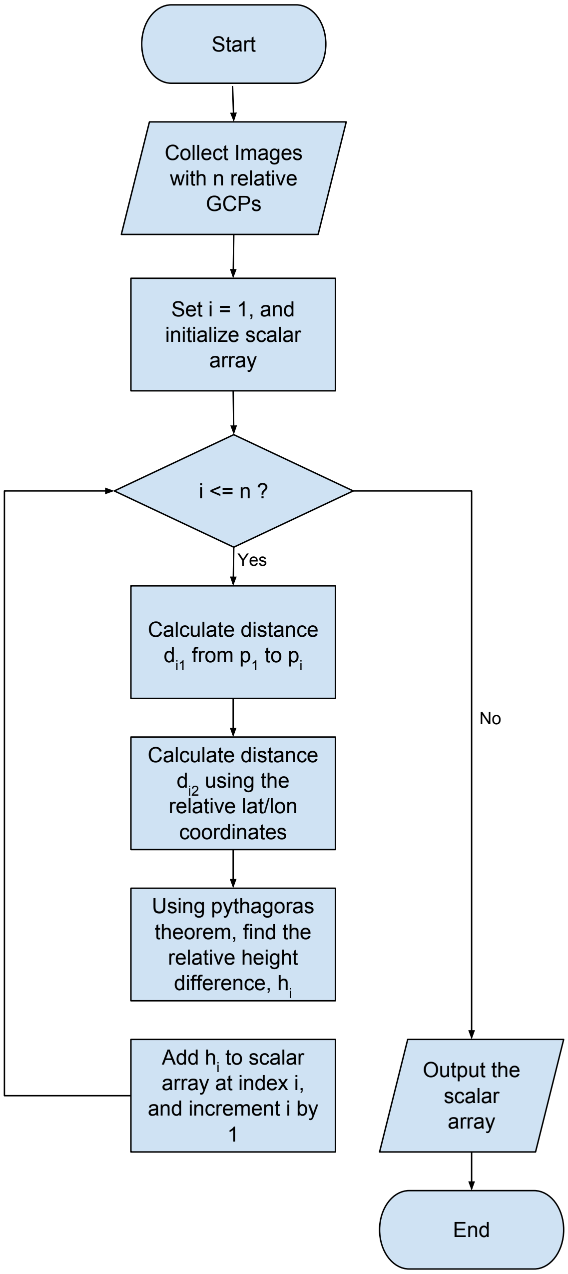

Figure 7: Flow Chart for Scalar algorithm

We got the following values for comparison:

(1) Volume for unit Cube: 1 cubic unit,
(2) Volume for the Sphere with a radius of 0.5 units: 0.5235987756 cubic unit.

4.3 Tests and Results

We evaluated the performance of VolCalc for above-mentioned point clouds in terms of accuracy as well as execution time. We also wanted to evaluate the impact of different tuning parameters on the performance of our algorithm. It should be noted that one of the natural conjectures that can be made regarding VolCalc is that thinner slices lead to a more accurate volumetric measure. This is because thinner slices will result in less overlap of the points, which in turn will result in more accurate volume measurements. We can control the thickness of a slice based on the number of slices being used. Therefore, the height of each slice can be calculated as:

$$h = \frac{maximumZ - minimumZ}{number\ of\ slices}. \tag{2}$$

Since the point cloud is invariable in terms of z coordinates, we varied the number of slices to manipulate the height of each slice. First, we compare the performance of VolCalc with known volume (i.e., the volume of a sphere). Fig. 8 and 9 show the accuracy of VolCalc estimates with respect to the number of slices for cube and sphere, while table 1 and 2 show the actual values calculated by our algorithm. It can be observed from Fig. 9 that the accuracy for sphere improves when the number of slices is increased from 10 to 50, but decreases when the number of slices is increased from 50 to 100. So, we can make a possible conjecture that there lies an optimal number of slices here for every object.

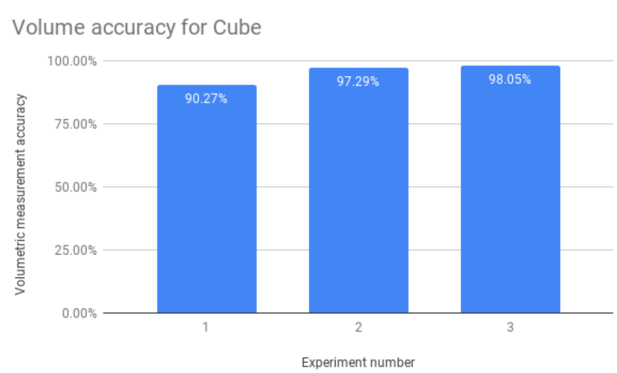

Figure 8: Bar graph displaying the Volume calculation accuracy for Cube

Table 1: Volumetric measure of cube

Experiment	1	2	3
Number of Slices	10	36	50
Volume	0.902733	0.972945	0.980506

The choice for the number of slices was based on the total number of unique z coordinates of the points. In the case of the Sphere, for example, we had 160 unique z coordinates for the points, making 160 our upper limit for the number of slices, if we do not want empty slices. In the case where the slice has no points, our algorithm returns zero volume, as would be expected.

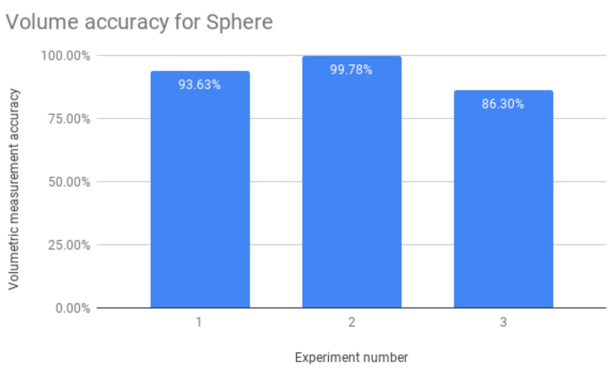

Figure 9: Bar graph displaying the Volume calculation accuracy for Sphere

Table 2: Volumetric measure of sphere

Experiment	1	2	3
Number of Slices	10	50	100
Volume	0.490252	0.522469	0.451873

By testing against the known volumes of cube and sphere against the values produced by Cloud Compare (see Table 5), we were able to detect that Cloud Compare's method of finding volume using meshes was inaccurate in clouds like that of ours, which have "holes" in them.
Tables 3 and 4 show the volumetric measurement derived by VolCalc for the Stanford Bunny and Vase, respectively. As of now, we cannot test the accuracy of our volumetric measure for the Stanford Bunny and the Vase, since the values produced by Cloud Compare for the sphere and the cube were inaccurate.

Table 3: Volumetric measure of Stanford Bunny

Experiment	1	2	3	4
Number of Slices	10	100	1000	2500
Volume	0.232144	0.220070	0.203281	0.169103

Table 4: Volumetric measure of Vase

Experiment	1	2	3	4
Number of Slices	10	100	1000	2500
Volume	0.228226	0.232800	0.214157	0.166578

To measure the cost-effectiveness of the module, we also measured the time it took the program to compute the estimates. Table 6 displays the values gathered. All of these time values are provided in seconds. It shall be noted that this was the real time it took for the program to run. This time may vary on different machines with different capabilities. Since Stanford Bunny is similar to Vase, and Sphere is similar to Cube in the number of unique z coordinates

Table 5: Volumetric measure provided by Cloud Compare

Mesh created with	Cube	Sphere	Bunny	Vase
XY-plane	0.195088	0.132541	0.165937	0.0734697
best-fit plane	0.501907	0.132928	0.175176	0.0498891

Table 6: Time taken by VolCalc algorithm to generate Volume

Point Cloud	Number of Slices		
	10	100	1000
Stanford Bunny	0.552s	0.548s	0.524s
Circle with Radius 0.5 unit	1.499s	1.362s	-

and number of slices used, we only calculated the time for Stanford Bunny and the Sphere.
An interesting and counter-intuitive trend was the decrease in time taken to compute volume with more number of slices. We will only be able to say that with higher confidence in the future when we have more experiments performed.

5 FUTURE WORK

One of the major future work includes implementing the Scalar algorithm for the Point Clouds, to account for the inaccuracy in the current height estimations. Further, I would like to develop the software that connects all the pieces and makes 3D-Quantify operate as one unit.
One major step in 3D-Quantify is 3D reconstruction using SfM, which works poorly with highly reflective surfaces like snow. This makes it hard to capture good quality images, and hence, might reduce the effectiveness of the workflow in winter times.

6 CONCLUSIONS

In this paper, we proposed 3D-Quantify as an open source workflow for generating the volume of a survey area. As part of the workflow, we have proposed Scalar and VolCalc algorithm. The VolCalc algorithm has been implemented and tested against some geometric objects of known volume. Since the underlying Shoelace algorithm for area calculation produces the exact area covered by a polygon, our VolCalc algorithm has generated results between 86% and 99% accuracy. The high accuracy of the VolCalc algorithm, combined with the minimal time and training needed to use and run the software helps us in making the process more efficient and effective.

ACKNOWLEDGMENTS

The author would like to thank Dr. Charles Peck and Daniel Jacob Stoffregen for help with developing the initial idea, Dr. Ajit Chavan for help with direction and design, Dr. Xunfei Jiang and Dr. David Barbella for constant helpful feedback.

REFERENCES

[1] Evangelos Alexiou, Evgeniy Upenik, and Touradj Ebrahimi. 2017. Towards subjective quality assessment of point cloud imaging in augmented reality. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 1–6.

[2] Job Bouwman. 2016. Schematic drawing of the shoelace algorithm. https://en.wikipedia.org/wiki/Shoelace_formula#/media/File:Shoelace3.png [Online; accessed April 10, 2019].
[3] Bart Braden. 1986. The Surveyor's Area Formula. *The College Mathematics Journal* 17, 4 (1986), 326–337.
[4] Jonathan L Carrivick, Mark W Smith, and Duncan J Quincey. 2016. *Structure from Motion in the Geosciences.* John Wiley & Sons.
[5] OpenDroneMap Developers. 2017. WebODM. https://www.opendronemap.org/.
[6] Mark A Fonstad, James T Dietrich, Brittany C Courville, Jennifer L Jensen, and Patrice E Carbonneau. 2013. Topographic structure from motion: a new development in photogrammetric measurement. *Earth Surface Processes and Landforms* 38, 4 (2013), 421–430.
[7] Daniel Girardeau-Montaut. 2015. Cloud compareâĂŤ3d point cloud and mesh processing software. *Open Source Project* (2015).
[8] Steve Harwin and Arko Lucieer. 2012. Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (UAV) imagery. *Remote Sensing* 4, 6 (2012), 1573–1599.
[9] Andreas Kaiser, Fabian Neugirg, Gilles Rock, Christoph Müller, Florian Haas, Johannes Ries, and Jürgen Schmidt. 2014. Small-Scale Surface Reconstruction and Volume Calculation of Soil Erosion in Complex Moroccan Gully Morphology Using Structure from Motion. *Remote Sensing* 6, 8 (2014), 7050–7080.
[10] Tapas R Martha, Norman Kerle, Victor Jetten, Cees J van Westen, and K Vinod Kumar. 2010. Landslide volumetric analysis using Cartosat-1-derived DEMs. *IEEE Geoscience and remote sensing letters* 7, 3 (2010), 582–586.
[11] Georgia Institute of Technology. [n. d.]. Large Geometric Models Archive. https://www.cc.gatech.edu/projects/large_models/ [Online; accessed April 10, 2019].
[12] Onur Ozyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. 2017. A Survey on Structure from Motion. *arXiv preprint arXiv:1701.08493* (2017).
[13] Johnny C Ryan, Alun Lloyd Hubbard, Jason E Box, Joe Todd, Poul Christoffersen, J Rachel Carr, Tom O Holt, and Neal A Snooke. 2015. UAV photogrammetry and structure from motion to assess calving dynamics at Store Glacier, a large outlet draining the Greenland ice sheet. (2015).
[14] Darren Turner, Arko Lucieer, and Steven M De Jong. 2015. Time series analysis of landslide dynamics using an unmanned aerial vehicle (UAV). *Remote Sensing* 7, 2 (2015), 1736–1757.
[15] Darren Turner, Arko Lucieer, and Christopher Watson. 2012. An Automated Technique for Generating Georectified Mosaics from Ultra-High Resolution Unmanned Aerial Vehicle (UAV) Imagery, Based on Structure from Motion (SfM) Point Clouds. *Remote sensing* 4, 5 (2012), 1392–1410.
[16] Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and JM Reynolds. 2012. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology* 179 (2012), 300–314.
[17] Yu Zhi, Yikun Zhang, Hao Chen, Kaifeng Yang, and Hui Xia. 2016/01. A Method of 3D Point Cloud Volume Calculation Based on Slice Method. In *2016 International Conference on Intelligent Control and Computer Application (ICCA 2016)*. Atlantis Press. https://doi.org/10.2991/icca-16.2016.35