Automated 3D Modeling of Objects using LiDAR

Vijay Singh Sisodiya, Sunil Kumar, Amitangshu Pal
Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur, Uttar Pradesh, India
E-mail:{vijayss22,sunilklc23,amitangshu}@iitk.ac.in

2025 IEEE Virtual Conference on Communications (VCC) | 979-8-3315-8724-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/VCC67261.2025.11351148

***Abstract*—Automated 3D modeling has several potential applications ranging from industrial quality control to archaeological artifact analysis and rapid prototyping. This paper presents the development and implementation of a LiDAR-based system for 3D modeling of various objects. We create a versatile and efficient scanning setup using a 2D LiDAR sensor – by leveraging the advantages of LiDAR technology, we achieve precise distance measurements and create high-resolution 3D models. Through experimentation, we show that our proposed solution produces high-fidelity 3D models for a wide variety of object types, shapes and sizes, which can be used to estimate their precise volume and surface areas. For all the objects tested in our experiments, the volume and area estimation error remains below 10%, whereas for most of the objects, the error remains below 2-3%.**

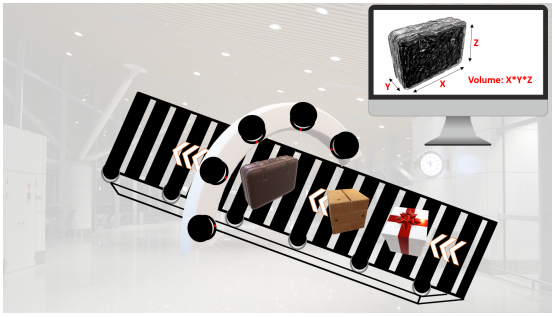
***Index Terms*—3D modeling, volume estimation, LiDAR**

I. Introduction

3D modeling is a process of creating a 3D representation of an object. The field of 3D object modeling has seen significant advancements over the years [1], [2], primarily driven by the the growing demand for rapid, accurate, and cost-effective methods of replicating real-world objects into 3D printed versions. According to a recent article [3], the market for 3D mapping and modeling will worth $12.13 billion by 2028. Traditional methods of creating 3D models for printing often involve time-consuming manual modeling or imprecise scanning processes, which might cause differences between the original object and its replica. By leveraging advanced LiDAR (Light Detection and Ranging) technology for high-precision point cloud acquisition, coupled with sophisticated algorithms for point cloud processing and mesh generation, we aim to streamline the workflow from object scanning to 3D printing. This approach promises to reduce the time and expertise required for creating accurate 3D models.

Motivation and background: The real motivation comes from our day-to-day observation in areas like warehouse management, courier, shipping services etc., where the dimension, area, and volume assessment of objects play important roles and directly proportional to revenue, as these business models incur charges on the basis of object volumes and dimensions. For measuring volumes of simple objects like cubes, cuboids, spheres, it is not difficult. However, if the same is to be done for objects like ovals and other asymmetric shapes, then volume calculation would not be very easy. Therefore, if the same can be done with the help of advance tools in short time and with accurate measurements, it can greatly reduce the processing time. Fig. 1(a) shows one such scenario, where a platform equipped with multiple LiDARs is used to scan some objects placed on a conveyor belt, with the object's dimensions and volumes displayed on a screen.

This research was supported by the IIT Kanpur Initiation Grant IITK/CS/2021164



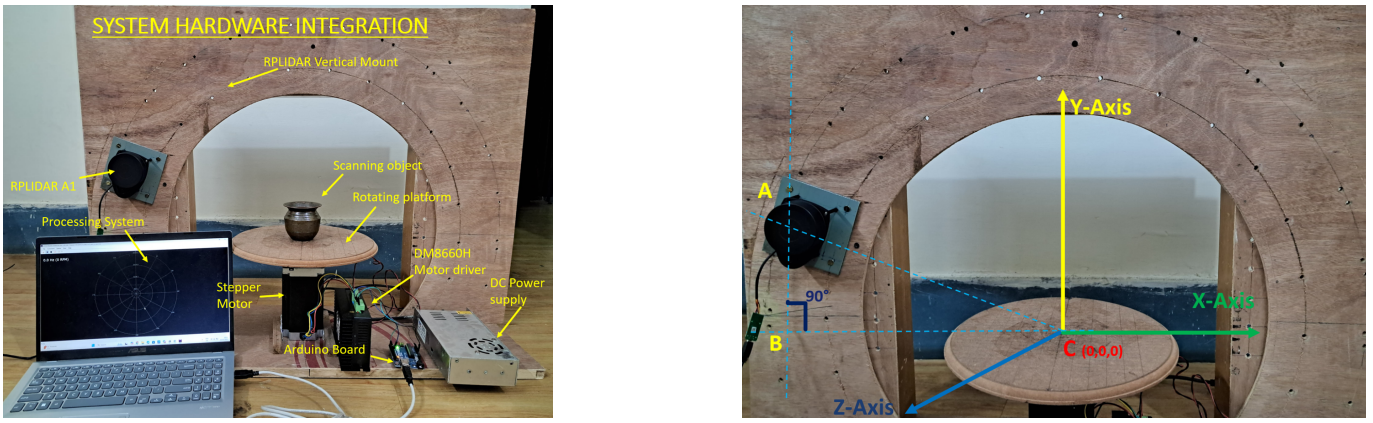
(a)

(b)

Fig. 1. (a) Multiple LiDARs being used to scan and determine the volume or surface area of an object placed on a conveyor belt. (b) 3D models featuring the actual objects on top.

Apart from the surface area or volume estimation, there are numerous scenarios where the proposed solution can be useful. For example, the solution can be used in scanning monuments and artifacts, which can be used for recreating their detailed 3D models. It can be effective tool in creating the base models of animated characters for movies and video games. In addition to that, the solution can also be utilized in 3D digital twins applications [4] where the scanned point clouds can be processed and converted into precise 3D digital twins of physical objects and environments. Also, our model can be utilized in industrial scenarios where a machinery spare part is broken and needs to be reproduced. Through 3D scanning of the damaged component, an accurate digital model can be created, enabling efficient and precise manufacturing of the replacement part.

979-8-3315-8724-6/25/$31.00 ©2025 IEEE



(a) (b)

Fig. 2. (a) Hardware setup of our LiDAR-based 3D modeling system. (b) Axis convention used in our study.

Our contributions: For this purpose, we choose LiDAR as our primary sensor which can be used not only for the object volume or area estimation, but is also highly effective for 3D model replication due to several key advantages. It offers exceptional accuracy, often within millimeters, making it superior to many other methods for capturing fine details and complex geometries. The proposed solution involves the formulation of LiDAR and object placement platform to obtain a 360° scan of the object, including side profiles and top profile. The objective entails the development of an experimental setup where the LiDAR is positioned in the vertical plane such that it captures the vertical profile of the object from one direction. Thereafter, the object or the LiDAR is shifted to different positions for another set of data capture. The solution involves subsequent post-processing steps to use these data points for meaningful interpretations in the form of a 3D shape. With extensive experimentation, we show that the proposed solution reconstruct various object's shapes with high level of accuracy. To be specific, the volume and area estimation error for all the objects tested remain below 10%, whereas for most of the objects, the error remains below 2-3%. A brief demonstration of our proposed solution can be found in https://youtu.be/CMEbtV7EUR0.

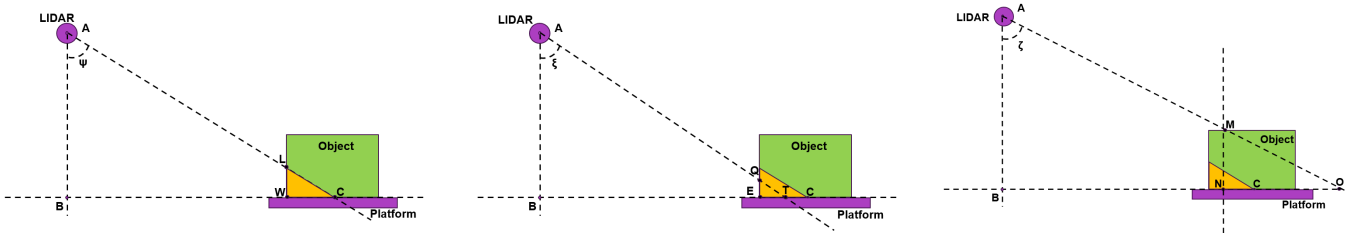
Paper organization: The paper is organized as follows. Section II summaries the related literature. Our system setup and the proposed solution are discussed in section III. Section IV discusses the experimental results. Further discussions related to our work and conclusions are summarized in section V-VI.

II. Related work

Various 3D scanning technologies: Three-dimensional scanning of real-life objects has become increasingly important in various fields, enabling digital interaction and analysis of physical objects [5]. Several techniques have been developed to create detailed 3D representations of objects and environments. Photogrammetry is one of these techniques that involves creating 3D models from multiple 2D images [6]. By analyzing the differences between images taken from different angles, photogrammetric techniques can reconstruct the 3D geometry of an object or scene. Time-of-Flight (ToF) cameras are also used for 3D scanning; these cameras operate by emitting short pulses of light, typically in the infrared spectrum, towards a scene [7]. These pulses reflect off objects and return to the camera sensor. The camera measures the time it takes for each pulse to travel to the object and back, which is then used for distance estimation and depth map generation. Structured light scanning is also used for 3D scanning, which involves projecting a known pattern of light, such as a grid or series of stripes, onto an object [5]. A depth camera then captures images of the deformed pattern, which is then used for constructing the object's surface. The authors in [8] have used three depth cameras for the 3D scanning of full human bodies. The authors in [9] have conducted a case study to investigate the possibilities and challenges of 3D scanning technology for reverse engineering and product quality control.

Methodologies adopted for volume estimation: Volume estimation is a critical application of 3D scanning, particularly in industries where quantifying transported materials are essential. The authors in [10] have proposed a solution for estimating the volume of load in moving trucks using a pair of multi-layer LiDAR sensors. Authors in [6] have utilized multi-view photogrammetry and 3D reconstruction software for automated volumetric measurements of truckloads. Similarly, the authors in [11] have developed an automated volume estimation solution for haul-truck loads in mining operations. The authors in [12] have proposed a method for volumetric estimation of contained soil using 3D sensors, which has applications in construction and earth-moving operations. Several studies have focused on applying 3D scanning techniques to the forestry sector. Authors in [13] have developed an automated log counting system using LiDAR technology, whereas in [14] the authors have utilized laser scanning for roundwood measurement of truck loads. The authors in [15] have used terrestrial laser/LiDAR scanning, to create a detailed point cloud of stems, which can be helpful for monitoring the quality of standing trees.

Similar to our intended approach for volume estimation, the authors in [16] have presented a simulated 3D scanning



(a) (b) (c)

Fig. 3. Illustration of three cases based on the position of the data points w.r.t. the line connecting the LiDAR and the center of the rotating platform.

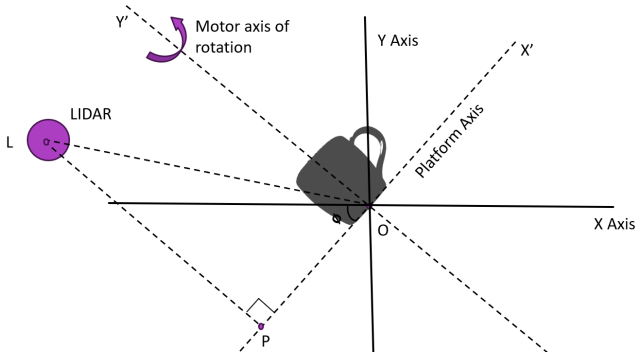


Fig. 4. Illustration of the tilted LiDAR orientation w.r.t. the global axis.

system using a LiDAR device, a rotating platform, and an arc-shaped scanning structure. They have discussed the proof of concept for volume estimation using SimTwo simulator [17] for creating a realistic simulation environment, including non-idealities and dynamic constraints.

III. Framework for our LiDAR based 3D modeling

A. Proposed framework

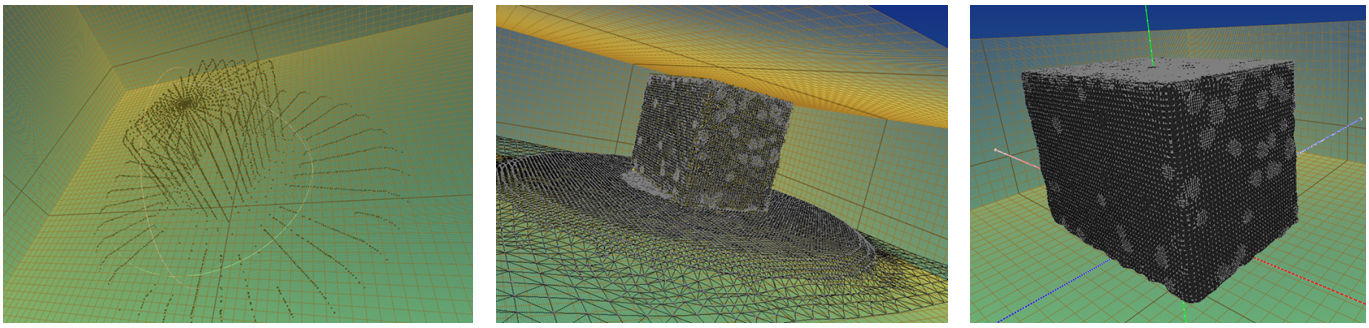
Hardware setup: Our hardware setup consists of a RPLiDAR sensor [18], a stepper motor controlled by an Arduino board [19], and a rotating platform, as shown in Fig. 2(a). The platform is positioned parallel to the ground surface (XZ plane). A high-torque stepper motor, controlled by a DM860H driver and powered by a 24V DC supply converted from 220V AC, rotates a wooden platform on which objects are placed for scanning. The stepper motor's shaft is aligned with the Y-axis and connected to the platform using shaft coupling. An Arduino board, connected to a laptop, receives rotation commands and controls the motor's movement.

This platform serves as the base on which the object to be scanned is placed. The fixture of this wooden platform over the motor shaft coupling is done carefully with precise measurement to avoid calibration errors later in the processing stages. Objects to be scanned are placed at the center of the rotating platform (i.e. at C in Fig. 2(b)) having its base in contact with the platform surface (i.e. in XZ plane at Y = 0). As the object rotates, the RPLiDAR sensor captures point the cloud data. This synchronized operation allows for comprehensive data capture; the system laptop initiates the process, the Arduino controls the platform's rotation, and the RPLiDAR continuously scans the object. Using Power automate tool [20], the data of individual sets at each rotation angle is saved into the local storage. These 2D data points are to be processed later on into 3D global coordinate system, thus enabling the construction of a detailed 3D model.

In order to capture the side profile of the object to be scanned, the best position for LiDAR is ideally at around 45° inclination from the center of the platform. From this position, object side profiles and top profiles, both can be captured. For objects where a single position scan is inadequate in delivering all the data points of its surface contours, multi position scan is required to obviate chances of gaps in data mesh. Therefore, in our setup, we ensure that the center point of the platform, where the object is placed, is equidistant from all the LiDAR positions. All these positions are arranged in a circular format, with the center of the platform as the origin. This arrangement helps us to merge the scanned data from different positions. Additionally, the LiDAR's scanning plane is aligned vertically, as it is placed parallel to the XY plane, with its axis of rotation parallel to the Z axis. This alignment allows the LiDAR to capture vertical profiles of the object.

Setup calibration: Fig. 3(a) illustrates the schematic of our proposed framework, where the LiDAR is kept at one of its positions (say at A), and the center of the platform is at C. In order to measure the sides and angles of $\triangle ABC$, (a) proper calibration of physical arrangements of components viz. stepper motor, (b) platform centering with respect to LiDAR's capturing, (c) platform's tilt adjustment to make it parallel to horizontal plane, (d) LiDAR mount structure alignment to avoid slant movement of LiDAR, and (e) accurate calculation of angle of depression of C from LiDAR i.e. $\angle BAC$, are required to be done. To achieve this, we keep a small size object at C on the platform. The LiDAR is adjusted in its horizontal and vertical plane in such a way that the object is visible on the frame grabber application. Now, we get d_{AC} (i.e. the distance between A and C) and $\angle BAC$ from the LiDAR scan of the platform. With the help of these two variables, we can easily find out d_{AB}, d_{BC} and $\angle BCA$. Since C is the center of reference in the global coordinate system, therefore, with respect to C, we can find the coordinates of A and B as distances d_{AB} and d_{BC} are calculated. The same process is repeated for all other LiDAR positions.

One practical aspect observed during the experimentation is the placement of the object on the rotating platform. An object can be placed anywhere on the platform. It is not important



(a) (b) (c)

Fig. 5. LiDAR scans of a cardboard box, where (a) the point cloud generated from the LiDAR scan are passed through the (b) surface reconstruction process. (c) Final mesh cloud after the post-processing steps.

to align an object with center C of the platform until and unless the object is kept too far away from the C. This is because of the fact that, at certain positions of motor rotations, the object may go completely outside the LiDAR's capturing plane, thereby causing the object to go out of sight of the LiDAR i.e. in blind zone. This will result in missing data points at that particular position. Therefore, while it is not important to align the center of an object with the center of the platform, it is important to place the object closer to the center in order to get maximum data points during each scanning motor rotation.

B. Data Capturing

During scanning, an object is placed close to the center of the rotating platform. The LiDAR is activated by the Frame grabber application and begins scanning in its vertical plane. In the Frame Grabber, we configure settings such as scanning mode, frequency, and other relevant parameters. The application captures and stores data points for the full 360° LiDAR scan; each scan represents a single vertical slice of the object at a specific platform rotation angle.

After a complete vertical scan, the stepper motor rotates the platform by set degrees around the Y-axis. Motor rotation with finer granularity leads to denser data point collection of the object. For dense mesh cloud, capturing profiles at 2° increment is observed to be satisfactory. The rotation of the platform and the LiDAR scanning are synchronized through the controlling computer, ensuring each vertical scan corresponds to a specific rotational position.

C. Data Processing

The captured LiDAR data corresponding to a specific rotation angle are initially stored in the form of tuples, i.e. (Angle, Distance) using the frame grabber application. We next apply the following post-processing steps on these data, to generate the 3D point cloud.

Applying rotation to compensate the LiDAR's orientation: In the LiDAR scanning setup, the local scanning plane of the sensor, defined by its own x-y coordinate frame, is oriented at an inclination angle of ϕ relative to the global x-y plane of the rotating platform, as seen from Fig. 2. This misalignment results in a systematic geometric tilt in the captured data, where scanned objects appear tilted with respect to the platform-fixed coordinate system. This configuration is illustrated in Fig. 4, which shows the LiDAR's internal scanning plane inclined relative to the horizontal axis. To correct this tilt and restore the object to its true upright orientation, a 2D rotation transformation is applied to all scanned points. The transformation rotates each point (x', y'), originally expressed in the LiDAR's tilted frame, into the global frame by an angle ϕ, using the standard two-dimensional rotation matrix:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{1}$$

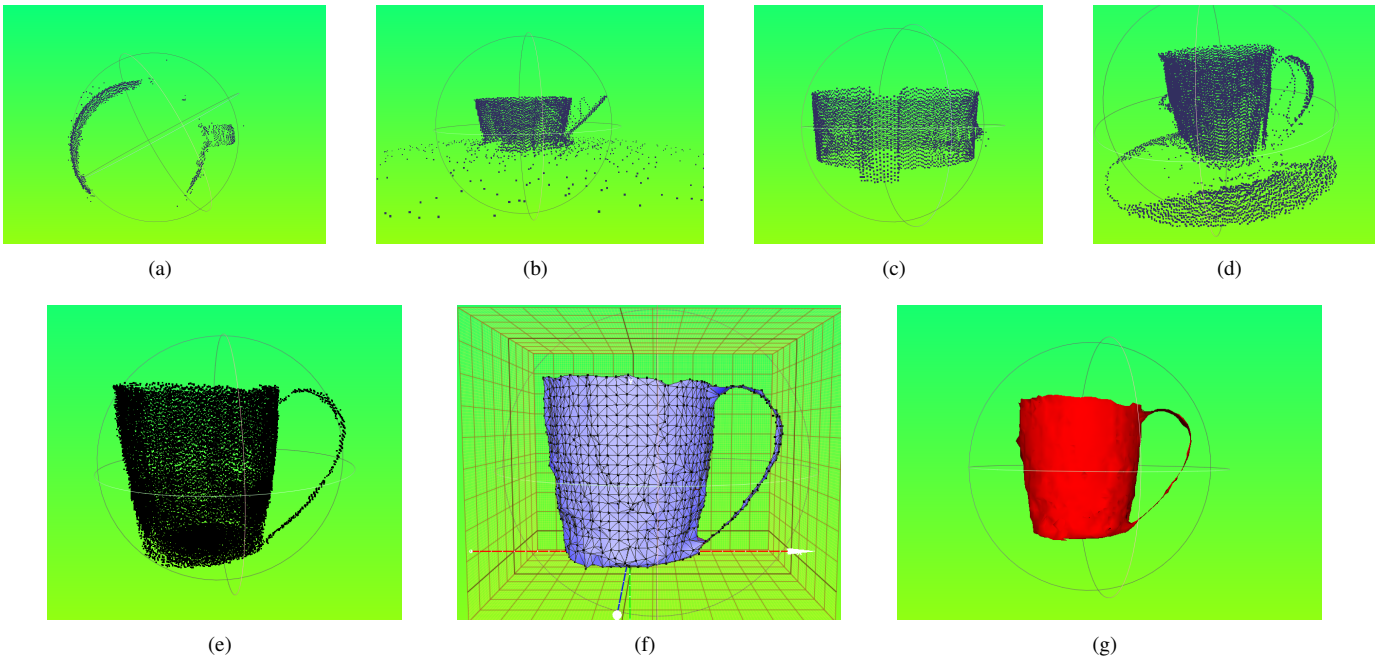
$$x = x'\cos\phi - y'\sin\phi, \quad y = x'\sin\phi + y'\cos\phi \tag{2}$$

After applying this transformation to all points in the scan, the scanned object reorients to an upright position within the global coordinate system. This correction step ensures that the final point cloud reflects the true geometry of the physical object, enabling accurate downstream analysis such as 3D reconstruction, shape modeling, and object detection. By treating the LiDAR's scanning frame as a rotated reference and correcting for its angular displacement, the method achieves reliable spatial alignment without modifying the sensor's physical configuration.

Calculating coordinates of the data points in 2D space: After applying rotation for compensating the LiDAR's orientation in equation(2), we can treat the LiDAR's orientations at all the points in Fig. 2 virtually identical, as depicted in Fig. 3. In Fig. 3(a), we assume that the line connecting A to C intersects the object at L. Now, the data points recorded by the LiDAR can fall in three zones in Fig. 3, i.e. they can lie on yellow zone, green zone or right at the intersection L. In the following, we discuss all three cases separately. For all these discussions, we assume that $\angle CAB$, d_{AC}, d_{AB} and d_{BC} are known during the calibration stage.

Case I, i.e the data point lies at L: This case is shown in Fig. 3(a), where $\angle\psi = \angle CAB$, which is measured by the LiDAR, along with d_{AL}. With these, for getting the coordinates of L and W w.r.t. C, we need to calculate d_{CW} and d_{LW}, which are expressed as:

$$d_{CL} = d_{AC} - d_{AL}, \quad d_{CW} = d_{CL}\sin\psi, \quad d_{LW} = d_{CL}\cos\psi \tag{3}$$



(a) (b) (c) (d) (e) (f) (g)

Fig. 6. LiDAR scans of a mug from different views (a)-(d). These scans are then (e) merged using MeshLab, followed by (f) mesh surface reconstruction, to create (g) a cleaned and closed mesh, ready for surface area estimation.

Case II, i.e. the data points lie in yellow zone: This case is shown in Fig. 3(b), where we assume that $\angle\xi = \angle TAB < \angle CAB$ and d_{AQ} are measured by the LiDAR. With these, the coordinates of Q and E can be measured from d_{QE} and d_{CE}, which are expressed as:

$$d_{AT} = d_{AB}\arccos\xi, \quad d_{QT} = d_{AT} - d_{AQ} \tag{4}$$

$$\therefore d_{QE} = d_{QT}\cos\xi \tag{5}$$

$$d_{ET} = d_{QT}\sin\xi, \quad d_{BT} = d_{AB}\tan\xi \tag{6}$$

$$\therefore d_{CE} = d_{BC} - d_{BT} + d_{ET} \tag{7}$$

Case III, i.e. the data points lie in green zone: This case is shown in Fig. 3(c), where we assume that $\angle\zeta = \angle OAB > \angle CAB$ and d_{AM} are measured by the LiDAR, therefore, the coordinates of M and N can be measured from d_{CN} and d_{MN}, which are expressed as:

$$d_{AO} = d_{AB}\arccos\zeta, \quad d_{MO} = d_{AO} - d_{AM} \tag{8}$$

$$\therefore d_{MN} = d_{MO}\cos\zeta \tag{9}$$

$$d_{NO} = d_{MO}\sin\zeta, \quad d_{BO} = d_{AB}\tan\zeta \tag{10}$$

$$\therefore d_{CN} = d_{BC} - d_{BO} + d_{NO} \tag{11}$$

Calculation of coordinates of data points in 3D space: In this stage, we transform the raw 2D LiDAR scans into a comprehensive 3D point cloud representation of the scanned object. Since, we know the scanning angle resolution of the platform (say δ), therefore, after the k-th rotation instance, the angle of rotation will be $k\delta$. With this, the x and z coordinates of all the points in Fig. 3 are updated as:

$$x = x\cos k\delta, \quad z = x\sin k\delta \tag{12}$$

D. 3D Modeling for Closed Objects

After obtaining the processed point cloud data, we next create a 3D model using MeshLab software [21]. For doing that we perform point cloud simplification using cluster-based decimation [22], where the cell size is set to approximately 1% of the object's bounding box diagonal, effectively reducing the number of points while preserving the overall shape of the object. Following this simplification, normal vectors are computed for the point set – the overall outcome is shown in Fig. 5(a), where a cardboard box is used as an object. Next, the 3D surface is constructed using the Poisson Surface Reconstruction [23], which creates a watertight 3D mesh as shown in Fig. 5(b).

After creating the initial mesh, several post-processing steps are necessary to refine the model. To begin with, irrelevant data points are manually removed to eliminate points that represent the platform base, surrounding objects, or environment, as well as the outliers that distort the overall shape of the object. Next, automated outlier detection and removal are performed using the Statistical Outlier Removal filter [21]. In addition to these, smoothing and hole filling are performed using the Laplacian Smooth filter [24] and Close Holes filter [25] respectively, to ensure a watertight mesh. Fig. 5(c) illustrates the final mesh cloud after all these post-processing steps are applied. This refined mesh provides an accurate representation of the scanned object, ensuring a higher quality input for the volume estimation process.

E. 3D Modeling for Hollow Objects

To generate a hollow object, it is necessary to scan the object from multiple viewpoints. In this discussion, we focus on generating a 3D model of a mug using scans taken from

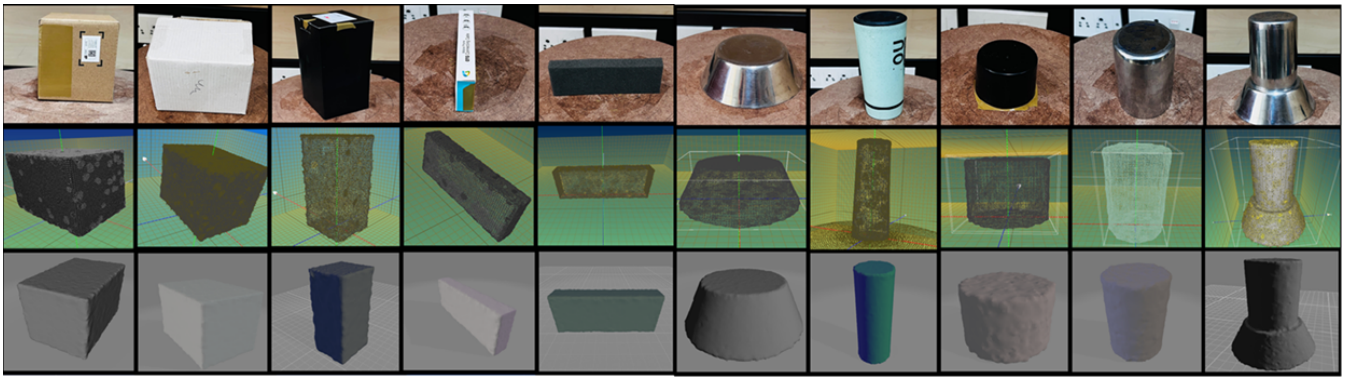


Fig. 7. Outcomes of our 3D modeling experiment on various objects with closed surfaces (top), along with the generated mesh cloud (middle), and the refined, smoothed version (bottom).

TABLE I
SPECIFICATIONS OF THE CLOSED OBJECTS USED

Object	Dimensions (in mm)
Brown cardboard box	$L \times B \times H$: $123 \times 80 \times 92$
White cardboard box	$L \times B \times H$: $123 \times 85 \times 100$
Black cardboard box	$L \times B \times H$: $90 \times 80 \times 178$
Thin cardboard box	$L \times B \times H$: $137 \times 20 \times 55$
Sponge block	$L \times B \times H$: $139 \times 20 \times 45$
Stainless steel (SS) bowl	Dia 1 : 78, Dia 2 : 120, Slant height: 46
Plastic bottle	Dia 1 : 69, Dia 2 : 60, Slant height: 194
Plastic bottle cap	Dia: 61, Height: 46
Stainless steel (SS) tumbler	Dia: 64, Height: 94
Combined bowl + tumbler	Same as for SS Bowl and Tumbler

four different viewpoints, as illustrated in Fig. 6(a)-(d). These four scans are then merged, and then passed through multiple cleaning processes. This involves point cloud simplification using cluster-based decimation [22], where the cell size is set to approximately 1% of the object's bounding box diagonal. After this simplification, normal vectors are computed for the point set, and the resulting object is shown in Fig. 6(e).

For the surface reconstruction of hollow objects, the Ball Pivoting Algorithm (BPA) [26] is used. The algorithm begins by positioning a ball in contact with three sample points to create an initial triangle, then systematically pivots the ball around each edge until it touches another point, forming new triangles as it progresses. This intuitive "walking" process continues along boundary edges, effectively growing a connected triangular mesh across the surface. The radius of the pivoting ball is selected based on the smallest gap between points or the angle of the scan. The reconstructed object using BPA is shown in Fig. 6(f). Post-processing steps are then performed, similar to those used for closed objects. These include outlier removal, surface smoothing, and closing of small holes to enhance the final model quality. The final processed object is shown in Fig. 6(g).

IV. EXPERIMENTAL EVALUATION

We now present the results of our proposed approach. Unless otherwise mentioned, the scanning angle resolution of the rotating surface is kept as 2°. To evaluate the accuracy of our proposed approach for closed objects, we estimate the volume of these objects from the LiDAR scanning and compare them with the original volume, whereas for hollow objects we estimate the surface area of these objects and compare them with their actual surface area.

Shape estimation results for the closed objects: We first scan and analyze a total of 10 closed objects with varying shapes, sizes, and surface properties. The specifications of these objects are shown in Table I. Fig. 7 shows the pictorial depiction of various objects, along with the generated mesh cloud and their refined, smoothed version. From these figures we can observe that, our solution can reconstruct all these different objects shapes with reasonable accuracy.

Volume estimation accuracy: Table II shows the volume estimation accuracy of different objects, which shows that the estimation error varies in between 1.5%-6.5%. From Table II we can also observe that the objects with predominantly flat surfaces (e.g., `cardboard boxes`) demonstrate better accuracy in volume estimation. For example, the brown, white, and black `cardboard boxes` have error percentages of 2.80%, 2.91%, and 3.43% respectively, whereas objects with curved surfaces (e.g., `SS bowl`, `tumbler`) show slightly higher error (typically more than 4%). On the other hand, larger objects generally yield more accuracy. For instance, the `bottle` (having a volume of 634,743 mm^3) has the lowest error rate of 1.54%, whereas smaller objects, such as the `thin cardboard box` and `bottle cap`, show higher error rates (i.e. 4.03% and 4.36% respectively). This suggests that our current setup may have limitations in capturing fine details or accurately representing small objects.

We can also observe that the objects with reflective surfaces, such as the `stainless steel bowl` and `tumbler`, pose challenges in scanning. The LiDAR rays are often reflected away from these objects, resulting in sparse and inaccurate data points, which leads to incomplete mesh clouds and subsequently, less accurate 3D models. These objects have error rates of 4.90% and 4.51% respectively. The `black cardboard box`, which tends to absorb LiDAR rays, result in a lower density mesh cloud and a higher error rate (3.43%), compared to its brown and white counterparts.

The `sponge block`, with its porous surface, shows the highest error rate (7.58%), likely due to the complexity of its surface structure. The porous nature of the sponge likely caused inconsistent LiDAR scans, resulting in the highest error

TABLE II
VOLUME ESTIMATION RESULTS FOR SCANNED CLOSED OBJECTS

Object	Actual Vol (mm^3)	Observed Vol (mm^3)	Error
Brown cardboard box	905,280	930,663.375	2.80%
White cardboard box	1,045,500	1,075,974.75	2.91%
Black cardboard box	1,281,600	1,325,675.125	3.43%
Thin cardboard box	150,700	156,780.671875	4.03%
Sponge block	125,100	134,586.3125	7.58%
SS bowl	320,556.23	336,269.34375	4.90%
Bottle	634,743.47411	644,542.0625	1.54%
Bottle cap	134,460.10	128,590.71875	4.36%
SS tumbler	302,397.14246	316,089.50	4.51%
Combined SS object	622,163.71628	662,393.125	6.46%

TABLE III
EFFECT OF SCANNING ANGLE RESOLUTION FOR CLOSED OBJECTS

Object	2°	4°	6°	8°
Brown box	2.80%	3.14%	3.38%	3.95%
White box	2.91%	3.35%	3.62%	4.32%
Black box	3.43%	3.94%	4.36%	5.22%
Thin box	4.03%	4.86%	5.30%	5.97%
Sponge block	7.58%	8.13%	8.86%	9.44%
SS bowl	4.90%	5.31%	5.89%	6.84%
Bottle	1.54%	1.87%	2.51%	2.97%
Bottle cap	4.36%	4.92%	5.41%	6.65%
SS tumbler	4.51%	4.88%	5.63%	6.19%
Combined object	6.46%	7.24%	8.34%	9.29%

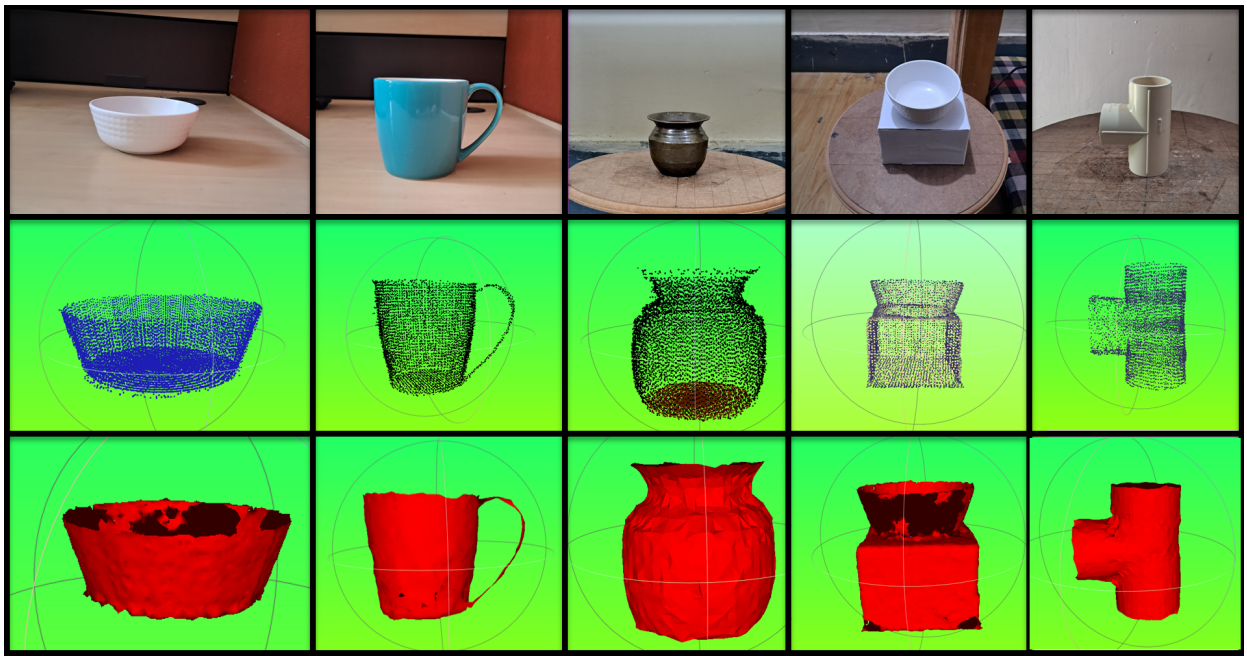


Fig. 8. Outcomes of our experiment on various objects with hollow surfaces (top), along with the generated mesh cloud (middle) and the refined and smoothed version (bottom). The objects from left to right are `hollow bowl`, `mug`, `pot`, `bowl over box`, and a pipe connector with `tee` shape respectively.

rate among all tested objects. The `combined SS object` (i.e. `bowl + tumbler`) on the other hand show a higher error rate (6.46%) compared to the individual objects, suggesting that complex arrangements may reduce accuracy.

Scanning objects with hollow surfaces: We now discuss our results of 3D reconstruction for objects with hollow surfaces. Fig. 8 shows the outcomes of our modified setup with objects having hollow surfaces. From Fig. 8, we can observe that the reconstructed object shapes closely match with the original ones. From these results, we can observe that with minor adjustments of our setup, we can obtain 3D modeling of different types of objects with reasonable level of accuracy.

Surface area estimation accuracy: Table IV shows surface area estimation results for different hollow objects. The observed surface areas closely match the actual values, with all estimation errors remaining below 3%. The `tee` structure shows the highest accuracy, with an error of just 1.24%, followed by the `mug` and `pot` with 2.12% and 1.99% errors, respectively. The `hollow bowl` yields a slightly higher error of 2.67%, likely due to its curved geometry. These results confirm the system's effectiveness in accurately estimating the surface areas of hollow and geometrically diverse objects.

Effect of different scanning angle resolutions: Tables III and V demonstrate the impact of varying scanning angle resolutions on the accuracy of both volume and surface area estimations. As expected, increasing the angle between successive scans leads to a consistent rise in error percentages for all scanned objects. This trend is attributed to the resulting reduction in point cloud density, which impairs the fidelity of surface reconstruction and volumetric estimation. The sensitivity to scanning angle varies depending on object geometry. For instance, complex or composite structures such as the `combined object` and `bowl over box` exhibit more pronounced error escalation compared to simpler shapes like the `bottle` or `brown box`, which maintain relatively lower errors even at coarser resolutions. Notably, objects like the `sponge block` show the highest susceptibility to angular resolution changes, while the `brown box` remains comparatively stable. For surface area estimation, the smallest scanning angle of 2° consistently yields the most accurate results, with errors ranging from 1.24% (for `tee`) to 2.67% (for `bowl`). However, at 8°, errors increase significantly—up

TABLE IV
SURFACE AREA ESTIMATION RESULTS FOR SCANNED HOLLOW OBJECTS

Object	Actual area (mm^2)	Observed area (mm^2)	Error
Hollow bowl	17942.74	18421.76	2.67%
Mug	20769.72	20328.816	2.12%
Pot	18878.68	18502.39	1.99%
Bowl over box	69785.08	71034.32	1.79%
Tee	11247.48	11387.91	1.24%

TABLE V
EFFECT OF SCANNING ANGLE RESOLUTION FOR HOLLOW OBJECTS

Object	2°	4°	6°	8°
Hollow bowl	2.67 %	3.55 %	4.54 %	5.82%
Mug	2.12%	1.81%	6.78%	7.48%
Pot	1.99%	4.11%	5.50%	6.54%
Bowl over box	1.79 %	2.93%	5.53%	6.67%
Tee	1.24%	1.96%	4.74%	9.71%

to 9.71% for the for the `tee`—highlighting the limitations of coarse scanning for capturing intricate geometrical features. These results emphasize the importance of maintaining a fine angular resolution to achieve high-precision 3D scanning, particularly when dealing with complex objects.

V. DISCUSSIONS AND LIMITATIONS

Our experiments and observation reveal several limitations of our proposed solution, which we discuss below.

Scanning transparent objects: Our proposed solution has severe limitations when attempting to scan transparent or glass objects. The refraction of LiDAR rays through these materials lead to highly distorted or completely failed scans.

Objects with intricate shapes: Objects with intricate shapes or sharp edges present challenges in accurate surface reconstruction. The scanning process tends to smooth out sharp edges and therefore misses small indentations.

Scanning the bottom surfaces: Another limitation of our current setup is the inability to capture the bottom part of the objects resting on the platform. This blind spot occurs because the object's base is in direct contact with the scanning platform, preventing LiDAR rays from reaching this area.

Effects of ambient conditions: Although not explicitly tested in our controlled environment, factors such as ambient light, dust, or moisture in the air could potentially affect the LiDAR's performance. These environmental variables could introduce noise or distortions in the scans, particularly for outdoor or industrial applications.

VI. CONCLUSION

In this paper, we demonstrate a LiDAR-based 3D shape modeling solution that yield promising results while also reveal important limitations and areas for future improvement. The solution is a significant step forward in non-contact and low-cost measurement technology. The system's ability to quickly and accurately model and measure a wide range of objects offers exciting possibilities across numerous fields, from manufacturing and logistics to archaeology and conservation. In future, we want to integrate our LiDAR based solution with other sensing technologies such as structured light or photogrammetry, which could help overcome the limitations related to surface reflectivity and transparency. This hybrid approach may provide more accurate 3D models, especially for objects with challenging surface properties.

REFERENCES

[1] "What is 3d modeling what's it used for" https://conceptartempire.com/what-is-3d-modeling/.

[2] "3d modeling," https://www.sw.siemens.com/en-US/technology/3d-modeling/.

[3] "3d mapping and modelling market is expected to reach us$ 12,134.08 million by 2028," https://www.theinsightpartners.com/pr/3d-mapping-modelling-market, 2022.

[4] M. Segovia *et al.*, "Design, modeling and implementation of digital twins," *Sensors*, vol. 22, no. 14, 2022.

[5] J. Straub *et al.*, "Development of a large, low-cost, instant 3d scanner," *Technologies*, vol. 2, no. 2, pp. 76–95, 2014.

[6] M. Acuna *et al.*, "Automated volumetric measurements of truckloads through multi-view photogrammetry and 3d reconstruction software," *Croatian Journal of Forest Engineering*, vol. 40, no. 1, pp. 151–162, 2019.

[7] K. Kimoto *et al.*, "Development of small size 3d lidar," in *IEEE ICRA*, 2014, pp. 4620–4626.

[8] J. Tong *et al.*, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 643–650, 2012.

[9] A. Gallo *et al.*, "3d reconstruction of small sized objects from a sequence of multi-focused images," *Journal of Cultural Heritage*, vol. 15, no. 2, pp. 173–182, 2014.

[10] L. L. Amorim *et al.*, "Simple and effective load volume estimation in moving trucks using lidars," in *SIBGRAPI*, 2019, pp. 210–217.

[11] E. Duff, "Automated volume estimation of haul-truck loads," in *Australian Conference on Robotics and Automation*, 2000, pp. 179–184.

[12] H. Anwar *et al.*, "Volumetric estimation of contained soil using 3d sensors," in *International Commercial Vehicle Technology Symposium*, 2014, pp. 11–13.

[13] H. Marshall *et al.*, "Automated log counting: proof of concept algorithm," in *SilviLaser*, 2011, pp. 1–13.

[14] M. Nylinder *et al.*, "Roundwood measurement of truck loads by laser scanning," *Field study at Arauco pulp mill Nueva Aldea*, pp. 1–9, 2008.

[15] Z. An *et al.*, "Tree stem volume estimation from terrestrial lidar point cloud by unwrapping," *Canadian Journal of Forest Research*, vol. 53, no. 2, pp. 60–70, 2023.

[16] J. A. Braun Neto *et al.*, "Low-cost 3d lidar-based scanning system for small objects," in *IEEE ICIT*, 2021, pp. 1–6.

[17] P. Costa *et al.*, "Simtwo realistic simulator: A tool for the development and validation of robot software," *Theory and Applications of Mathematics & Computer Science*, vol. 1, no. 1, pp. 17–33, 2011.

[18] C. Shikai, "Slamtec rplidar," https://www.slamtec.ai/product/slamtec-rplidar-a1/.

[19] E. Darie *et al.*, "Speed control of the direct current servomotor and the stepper motor with arduino uno platform," in *EENVIRO*, 2021.

[20] A. Guilmette, *Workflow Automation with Microsoft Power Automate: Achieve digital transformation through business automation with minimal coding*. Packt Publishing Ltd, 2020.

[21] P. Cignoni *et al.*, "Meshlab: an open-source mesh processing tool." in *Eurographics Italian chapter conference*, vol. 2008, 2008, pp. 129–136.

[22] M. Pauly *et al.*, "Efficient simplification of point-sampled surfaces," in *IEEE Visualization*, 2002, pp. 163–170.

[23] M. Kazhdan *et al.*, "Poisson surface reconstruction," in *Eurographics symposium on Geometry processing*, vol. 7, no. 4, 2006.

[24] P. Bourke, "Mesh smoothing in meshlab," https://paulbourke.net/miscellaneous/ffmpeg/meshlabsmoothing/.

[25] P. Liepa, "Filling holes in meshes," in *Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 2003, pp. 200—-205.

[26] F. Bernardini *et al.*, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.