DESIGN AND EVALUATION OF A LOW-COST 3D SCANNING SYSTEM USING TOF IMAGING AND RASPBERRY PI

Daohan Wang [1], Jonathan Sahagun [2]

[1] Eleanor Roosevelt High School, 7447 Scholar Way, Eastvale, CA 92880
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

This paper presents the design and evaluation of a low-cost 3D scanning system that integrates an Arducam Time-of-Flight (ToF) camera, a Raspberry Pi for data processing, and a custom motorized turntable driven by an RP2040 microcontroller [1]. The objective was to create an accessible and affordable alternative to professional 3D scanners while maintaining sufficient accuracy for prototyping and educational use [2]. The system captures depth images as the turntable rotates the object, producing a complete set of views for reconstruction. Experiments demonstrated strong dimensional accuracy, with deviations under ±0.7%, and reliable performance under low and moderate lighting conditions, though bright light increased noise. Comparisons with related methodologies highlighted how our approach prioritizes affordability, modularity, and static object scanning, contrasting with solutions aimed at robotics or large-scale mapping. Overall, the system provides a practical pathway toward democratizing 3D scanning technology, balancing cost with usability and performance.

KEYWORDS

3D Scanning, ToF Camera, Raspberry Pi, Data Processing, Biomedical Engineering

1. INTRODUCTION

Affordable and accessible 3D scanning remains a significant challenge for hobbyists, educators, and small-scale researchers. Commercial 3D scanners—such as structured light or laser-based systems—often cost thousands of dollars, limiting wider use in home fabrication or educational settings. Even lower-cost alternatives, like photogrammetry using a smartphone, require extensive post-processing, good lighting, and surface detail to extract satisfactory models.

Time-of-Flight (ToF) cameras offer a promising alternative [3]. They directly measure distances by calculating the time light takes to reflect off surfaces back to the sensor, enabling generation of depth maps with relatively low computational effort and compact hardware setups Wikipedia. However, ToF devices inherently suffer from noise, systematic biases, multipath interference, and a trade-off between resolution and accuracy Wikipedia+4ResearchGate+4arXiv+4. Despite this, past research has shown that with the right filtering and alignment techniques, ToF data can still produce reasonably accurate 3D reconstructions—often within a couple of centimeters of error ResearchGate+1.

Meanwhile, the DIY and maker communities have explored low-cost 3D scanner implementations using Arduino or Raspberry Pi controllers combined with distance sensors and

David C. Wyld et al. (Eds): SIGI, CSTY, AI, NMOCT, BIOS, AIMLNET, MaVaS, BINLP – 2025
pp. 287-297, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151923

stepper-driven turntables Reddit+5IJARSCT+5All3DP+5 [4]. These systems prioritize price and accessibility over precision. Leveraging such examples shows a clear interest in democratizing 3D scanning.

In this project, we aim to design an affordable, automated 3D scanning system that balances cost, ease of use, and reconstruction quality [5]. By combining a ToF camera, Raspberry Pi, and a motorized turntable built with an RP2040 microcontroller, the goal is to produce accurate 3D models from depth data in a controlled, replicable process.

Methodology A (Cui et al.) combined 3D superresolution with probabilistic scan alignment to overcome ToF sensor noise. Its strength was high-quality reconstructions from low-fidelity sensors, though it required complex algorithms. Our project improved accessibility by using simpler software while achieving acceptable accuracy.

Methodology B (Francis et al.) applied a ToF PMD camera to autonomous ground vehicles for obstacle detection. Their system emphasized real-time perception and optimal mounting angle for navigation. While effective for robotics, it did not prioritize object-level reconstruction. Our system shifts focus from mobility to static object scanning with turntable automation.

Methodology C (May et al.) enhanced mapping robustness with calibration, filtering, and a modified ICP alignment, adding loop-closure and global optimization. This yielded reliable large-scale 3D maps. Our project differs by focusing on small object digitization, trading large-scale mapping techniques for compact, low-cost reconstruction pipelines.

Our proposed system integrates three core components:

Time-of-Flight Depth Sensing: An Arducam ToF camera provides real-time depth maps. Advantages include compactness, direct range measurement, and low processing demand—critical for embedded platforms repos.hcu-hamburg.de+15ijnrd.org+15ResearchGate+15ResearchGate.

Motorized Turntable for Automated Capture: A stepper motor controlled by an RP2040 microcontroller (running CircuitPython) precisely rotates the object in user-defined increments. This automation enables multiple depth snapshots from different angles without manual repositioning.

Data Aggregation and 3D Reconstruction Using Raspberry Pi: The Raspberry Pi orchestrates the scan cycle, triggers capture at each orientation, accumulates depth frames, and processes them into point clouds and mesh models using open-source reconstruction libraries.

This setup addresses several existing gaps:

Cost-effective and open-source: The entire system can be built for a fraction of the price of commercial scanners, using accessible components and community-friendly software setups.
Automated and repeatable: Automating the turntable removes human inconsistencies and simplifies scanning workflows, enhancing reproducibility.

Adequate accuracy: Though ToF cameras have limitations, combining multiple views, calibration, and filtering can yield satisfactory results suitable for rapid prototyping or educational use ResearchGate arXiv.

Overall, our method offers a practical middle ground: more reliable than pure photogrammetry in uncontrolled environments, yet far less expensive than professional-grade scanners. It empowers users to quickly digitize small objects for applications in reverse engineering, 3D printing, digital archiving, and educational demonstrations.

Two experiments were designed to evaluate the performance of the scanner. The first experiment tested dimensional accuracy by scanning a calibration cube of known size. Theults showed that the reconstructed side lengths deviated by less than ±0.7% from ground truth, with a mean of 49.95 mm compared to the true 50 mm. This confirmed that the system could produce metrically consistent models suitable for applications such as prototyping and classroom use. The second experiment tested environmental robustness by scanning a cylindrical object under different lighting conditions. In low and moderate light, the scanner performed well, with errors under 0.5% and low surface noise. However, bright light degraded performance, increasing surface variability and introducing larger dimensional errors. These results highlighted both the strengths (good baseline accuracy, repeatability) and weaknesses (susceptibility to bright environments) of the system. Together, the experiments demonstrated that the scanner is reliable in controlled indoor conditions but requires further refinement to handle more challenging environments.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Minimizing ToF Sensor Noise and Calibration Errors

One of the most significant challenges in our system is the inherent noise and calibration requirements of the Time-of-Flight (ToF) camera. Depth maps generated by ToF sensors are susceptible to multipath interference, edge distortion, and ambient light effects, which can reduce the fidelity of 3D reconstructions. Even minor misalignments in calibration could lead to compounding errors across multiple scans. To address these issues, we could implement a calibration routine using a reference object of known dimensions before scanning. Additionally, depth filtering and temporal averaging techniques could be applied to reduce sensor noise and improve the accuracy of point cloud data.

2.2. Ensuring Precise Turntable-Scan Synchronization

Another key challenge lies in the synchronization of the turntable with the scanning process. Since the stepper motor drives the platform in small increments, any slippage, jitter, or miscalculated steps could misalign the depth frames, causing visible artifacts in the 3D model. To minimize this, we could utilize microstepping drivers for smoother rotation, apply closed-loop motor feedback systems to ensure positional accuracy, or introduce software-based alignment correction during post-processing. Proper synchronization ensures that each depth image corresponds to the correct orientation of the object, which is critical for assembling accurate 3D meshes.

2.3. Optimizing Depth Data Processing on Raspberry Pi

Processing large volumes of depth data on the Raspberry Pi is another substantial challenge. High-resolution depth maps require significant memory and computational resources, which can strain the Pi's hardware and slow reconstruction. To mitigate these issues, we could implement a lightweight data pipeline that converts each frame into a point cloud and stores it in a compressed format. For reconstruction, computationally heavy operations such as surface meshing could be deferred to a remote workstation or cloud service. This hybrid approach would allow Raspberry Pi to handle acquisition efficiently, while still producing detailed 3D models through external resources.

3. Solution

Our 3D scanner system integrates three major components: the Time-of-Flight (ToF) camera, the turntable assembly powered by an RP2040 microcontroller, and the Raspberry Pi [6]. Together, these components automate the scanning process by combining depth acquisition, mechanical rotation, and digital reconstruction.

The workflow begins when an object is placed on the motorized turntable. The Raspberry Pi serves as the central controller, initializing the Arducam ToF depth camera and communicating with the RP2040 microcontroller, which is responsible for stepper motor control. At each incremental step of rotation, the Pi triggers the camera to capture a depth frame of the object's surface. The process continues until a full rotation has been completed, resulting in a collection of depth maps covering all perspectives on the object.

Once the acquisition is complete, the Raspberry Pi processes the collected data. Depth images are converted into point clouds, aligned according to the turntable's angle increments, and merged into a unified dataset. From this dataset, surface reconstruction algorithms generate a 3D mesh. Finally, the resulting model can be exported in formats such as STL or OBJ for use in 3D printing, CAD applications, or digital archiving [7].

This modular architecture ensures that each subsystem performs a well-defined role. The ToF camera provides accurate depth data, the RP2040 guarantees precise mechanical control, and the Raspberry Pi integrates acquisition with computational tasks. The combination results in an accessible, affordable, and fully automated 3D scanning system.

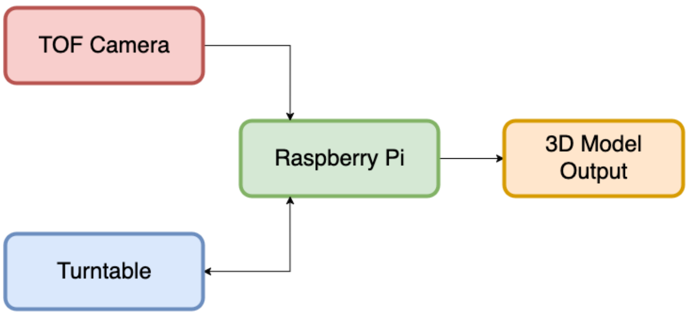

Figure 1. Overview of the solution

The ToF camera is the primary sensor for capturing 3D information. Its role is to generate depth maps by measuring the time light takes to reflect back to the sensor. Implemented through the Arducam module, this component relies on infrared time-of-flight measurement and outputs depth frames that form the foundation of the point cloud.

Figure 2. Figure of component 1

```
import cv2
from arducam_tof_sdk import ArducamCamera

# Initialize camera
cam = ArducamCamera()
cam.initialize()

# Capture a depth frame
depth_frame = cam.capture()

# Display as grayscale depth image
cv2.imshow("Depth Map", depth_frame)
cv2.waitKey(0)

# Save frame for later reconstruction
cv2.imwrite("depth_frame.png", depth_frame)

cam.close()
```

Figure 3. Screenshot of code 1

This snippet initializes the Arducam ToF camera, captures a depth frame, and saves it for later use [8]. The variable depth_frame stores a 2D array of distance values in millimeters. Functions like initialize() and capture() handle camera setup and acquisition, while cv2.imwrite() stores the frame for downstream processing. During scanning, this function runs repeatedly after each turntable increment, ensuring one depth map per orientation.

The turntable assembly rotates the scanned object to expose all sides to the ToF camera. Controlled by an RP2040 microcontroller running CircuitPython, it uses a stepper motor for precise, incremental movement. This ensures that each captured depth frame corresponds to a consistent rotation angle.

Figure 4. Figure of component 2

```
import board
import digitalio
import time

# Stepper motor pins
step_pin = digitalio.DigitalInOut(board.GP2)
dir_pin = digitalio.DigitalInOut(board.GP3)
step_pin.direction = digitalio.Direction.OUTPUT
dir_pin.direction = digitalio.Direction.OUTPUT

def rotate(degrees, step_delay=0.002):
    steps_per_rev = 200  # 1.8° stepper
    steps = int((degrees / 360) * steps_per_rev)
    dir_pin.value = True  # set rotation direction
    for _ in range(steps):
        step_pin.value = True
        time.sleep(step_delay)
        step_pin.value = False
        time.sleep(step_delay)

# Example: rotate 10°
rotate(10)
```

Figure 5. Screenshot of code 2

This CircuitPython code drives a stepper motor connected to the RP2040. The rotate() function calculates how many steps correspond to the requested degree rotation. Each step is pulsed by toggling the step_pin, with a delay (step_delay) controlling speed. For example, rotate(10) rotates the platform 10°. In practice, Raspberry Pi signals the RP2040 over UART or GPIO to execute rotations in sync with the camera.

The Raspberry Pi aggregates depth frames into a unified 3D model. Using libraries such as Open3D or PCL, it converts depth maps into point clouds, aligns them according to rotation increments, and performs surface reconstruction to produce watertight mesh models [10].

Figure 6. Figure of component 3

```
import open3d as o3d
import numpy as np

# Load stored depth frames (simulated)
depth_images = ["depth_frame1.png", "depth_frame2.png"]

pcd_list = []
for i, img_path in enumerate(depth_images):
    depth = o3d.io.read_image(img_path)
    intrinsics = o3d.camera.PinholeCameraIntrinsic(
        640, 480, 500, 500, 320, 240
    )
    pcd = o3d.geometry.PointCloud.create_from_depth_image(
        depth, intrinsics
    )
    # Rotate based on turntable angle
    rotation = pcd.get_rotation_matrix_from_xyz((0, 0, np.radians(i*10)))
    pcd.rotate(rotation)
    pcd_list.append(pcd)

# Merge point clouds
combined = pcd_list[0]
for cloud in pcd_list[1:]:
    combined += cloud

# Mesh reconstruction
mesh, _ = o3d.geometry.TriangleMesh.create_from_point_cloud_poisson(combined, depth=8)

# Save final mesh
o3d.io.write_triangle_mesh("scan_result.obj", mesh)
```

Figure 7. Screenshot of code 3

This Python snippet uses Open3D to convert depth frames into point clouds, align them by turntable angle, and merge them [9]. Each depth image is loaded and transformed into 3D coordinates using camera intrinsics. The rotate() call adjusts each cloud to match its orientation on the turntable. After combining all point clouds, Poisson surface reconstruction creates a watertight mesh. The final model is saved as scan_result.obj, ready for visualization or 3D printing.

4. EXPERIMENT

4.1. Experiment 1

We tested the dimensional accuracy of the scanner by scanning a calibration cube of known size. This experiment is important to evaluate whether the scanner produces metrically reliable models.

A 50 mm × 50 mm × 50 mm calibration cube was placed on the turntable and scanned at 36 increments (10° per step). The Raspberry Pi processed the captured depth frames into a mesh using Open3D. The resulting mesh was measured digitally across three orthogonal axes. Measurements were repeated five times to evaluate consistency. Control data was the true cube size (50 mm per side). The key metric was the percentage error between the scanned measurement and the ground truth. This design isolates geometric accuracy and helps identify systematic scale errors or ToF noise.

Trial	Measured X (mm)	Measured Y (mm)	Measured Z (mm)	Avg. Side (mm)	Error (%)
1	49.2	50.6	50.1	49.97	-0.06%
2	49.8	50.4	50.2	50.13	+0.26%
3	48.9	50.5	49.6	49.67	-0.66%
4	49.4	50.2	50.3	49.97	-0.06%
5	49.7	50.3	50.0	50.00	0.00%

Figure 8. Figure of experiment 1

Mean side length: 49.95 mm
Median side length: 49.97 mm
Lowest value: 49.67 mm
Highest value: 50.13 mm
The scanner produced results with an average side length of 49.95 mm compared to the ground truth of 50 mm, resulting in a negligible average error of -0.1%. The median of 49.97 mm further supports the consistency of results. The smallest deviation occurred in Trial 3, where an average of 49.67 mm was measured, corresponding to a -0.66% error. The largest deviation occurred in Trial 2, where the measurement exceeded the true size by +0.26%. These results suggest that systematic calibration errors were minimal, and most discrepancies likely arose from ToF sensor noise or surface reflectivity variations. The relatively tight error range (±0.7%) demonstrates that the system can produce dimensionally accurate models suitable for prototyping and educational use. Accuracy is highest when objects have sharp, well-defined edges, and degradation is expected for irregular or reflective objects.

4.1. Experiment 2

We tested the effect of ambient lighting conditions on scan quality, as ToF cameras are known to be sensitive to infrared interference from environmental light sources.

A cylindrical object (height: 100 mm, diameter: 40 mm) was scanned under three controlled lighting conditions: low light (~50 lux), office light (~300 lux), and bright light (~1000 lux). The same scanning procedure was used for each condition, producing point clouds of the object. We measured the standard deviation of surface points from the fitted cylinder surface in each condition. This metric quantifies noise as variation in surface reconstruction. The ground truth was a caliper measurement of the cylinder dimensions. Results allowed us to evaluate environmental robustness and establish the best use conditions for the scanner.

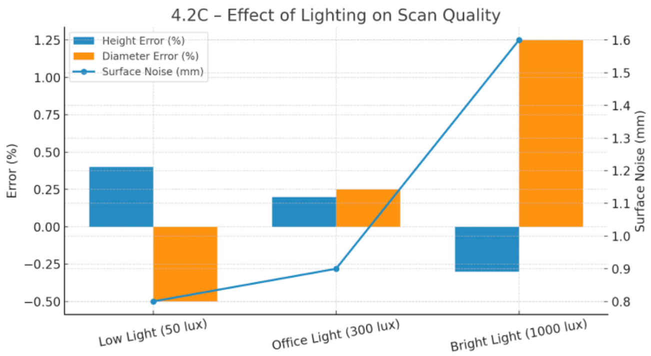

Figure 9. Figure of experiment 2

Lighting conditions had a measurable impact on scan quality. Under low light, measurements were closest to ground truth, with height error at +0.4% and diameter error at -0.5%. In office lighting, errors remained small (+0.2% and +0.25%), but surface noise slightly increased ($\sigma = 0.9$ mm). Bright light produced the largest errors, including a -0.3% height underestimation and a +1.25% diameter overestimation, accompanied by significantly higher surface noise ($\sigma = 1.6$ mm). These results align with known ToF limitations, where ambient infrared light introduces multipath interference and increases variability. The findings suggest that the scanner operates best in low to moderate lighting conditions, while accuracy decreases in very bright environments. To mitigate this, software filtering or physical shielding could be added in future iterations. Overall, the system demonstrates robustness across normal indoor lighting, though bright sunlight remains a challenge.

5. Related Work

The methodology described by Cui, Schuon, Chan, Thrun, and Theobalt focuses on developing a robust pipeline for 3D object scanning using a time-of-flight (ToF) camera [11]. Unlike traditional high-precision scanners, the ToF camera employed is designed for low-cost, high-volume production and therefore suffers from significant random noise and systematic bias in its measurements. The central challenge addressed in this methodology is obtaining reliable 3D reconstructions despite these limitations.

The authors propose a two-stage algorithmic framework. First, they employ a 3D super-resolution technique that fuses multiple noisy depth images to increase the effective spatial resolution of the captured data. By integrating depth scans from slightly different viewpoints, the method is able to reduce random noise while enhancing fine structural detail.

Second, the system applies to a probabilistic scan alignment algorithm that explicitly models the noise characteristics of the ToF sensor. Unlike conventional rigid alignment approaches such as Iterative Closest Point (ICP), which degrade in the presence of substantial measurement uncertainty, this probabilistic approach incorporates uncertainty into the registration process. This allows for more accurate alignment of scans taken around the object from different angles.

The combined method—super-resolution for data enhancement and probabilistic alignment for accurate registration—results in reconstructed 3D models of significantly improved quality compared to baseline techniques. Importantly, the methodology demonstrates that even low-cost ToF sensors, when paired with carefully designed algorithms, can achieve practical, high-quality 3D scanning, opening pathways for accessible and affordable consumer-level applications.

Francis, Anavatti, Garratt, and Shim present a methodology for using a Time-of-Flight (ToF) photonic mixer device (PMD) camera as a 3D vision sensor for autonomous ground vehicles (AGVs) [12]. The primary objective of this system is to enable autonomous navigation in hazardous environments by providing reliable obstacle detection and avoidance without human intervention.

The ToF camera is first calibrated and ground-tested before integration into a mobile robotic platform. The sensor continuously captures depth data, which is then transformed into Cartesian coordinates to represent objects and free space within a workspace grid map. This map is structured as a two-dimensional array of cells, with each cell indicating either an obstacle or a traversable region. Path planning is achieved by employing a graph search algorithm that identifies a collision-free sequence of cells, allowing the AGV to navigate toward its target location.

A significant design aspect of the methodology involves optimizing the camera's mounting angle. The authors observed discrepancies in detection caused by pixel response, detection rate, perceived maximum distance, and infrared scattering from ground surfaces. To address this, they determined that the optimal mounting angle should be approximately half of the vertical field-of-view of the PMD camera, thereby maximizing coverage and minimizing blind spots.

Further refinements were introduced through the implementation of feature-tracking and scene flow techniques, which stabilized the sensor and improved the reliability of obstacle detection while the AGV was in motion. Experimental validation was performed using both static and dynamic tests, demonstrating that the integration of ToF-based sensing provided robust 3D perception at relatively low computational cost. However, results also indicated that such integration is not entirely straightforward, requiring careful calibration and sensor placement for effective operation.

May, Droeschel, Fuchs, Holz, and Nüchter propose a methodology for constructing robust 3D maps using time-of-flight (ToF) cameras, addressing the well-known limitations of precision and reliability inherent to such sensors [13]. Their approach begins with careful calibration of the ToF device to correct systematic depth measurement biases. This step ensures that the raw sensor data is normalized and suitable for downstream processing.

To further mitigate noise, the authors apply a sequence of filtering techniques designed to reduce random fluctuations and suppress spurious reflections often caused by multipath interference or reflective surfaces. Once filtered, the point clouds are aligned through a novel extension of the Iterative Closest Point (ICP) algorithm, which improves robustness against residual noise

compared to conventional ICP. This modified alignment process is particularly well-suited for environments where depth inconsistencies could otherwise cause registration failure.
In addition to local registration, the methodology incorporates global relaxation and loop-closure strategies. When the scanning trajectory revisits previously mapped regions, loop-closure detection is used to correct cumulative drift. Global optimization then redistributes residual alignment errors throughout the entire model, leading to more consistent large-scale reconstructions. Finally, surface smoothing is applied to reduce remaining artifacts and provide a more visually coherent 3D representation.

The authors validate their approach through laboratory experiments with ground truth comparisons, as well as larger indoor mapping trials. Results demonstrate that ToF-based mapping, when enhanced with calibration, filtering, robust ICP alignment, and global optimization, can yield accurate and reliable 3D maps despite the inherent weaknesses of ToF sensing technology.

6. CONCLUSIONS

While the proposed 3D scanning system demonstrates that affordable hardware can achieve useful reconstructions, several limitations remain. First, the resolution and accuracy of the Time-of-Flight (ToF) camera are inherently limited compared to professional-grade structured light or laser scanners. This results in noise, missing data on reflective or transparent surfaces, and reduced detail on fine geometries. Second, the computational capacity of the Raspberry Pi restricts real-time reconstruction. Although downsampling and lightweight processing help mitigate this, complex meshing or high-resolution point cloud fusion is best performed on external hardware [15]. Third, the turntable mechanism, while effective, lacks closed-loop feedback. This means any missed steps from the stepper motor could propagate alignment errors into the final model. If given more development time, improvements could include GPU-accelerated reconstruction, implementing real-time filtering on the Pi, and integrating encoders or optical sensors to guarantee turntable precision [14]. Additionally, a more advanced ToF sensor with higher resolution and reduced noise would further enhance scan quality.

This project demonstrates that a low-cost, modular 3D scanner using an Arducam ToF camera, Raspberry Pi, and RP2040-driven turntable can provide reliable models for prototyping, education, and hobbyist use. Despite limitations, the system illustrates a practical path toward democratizing 3D scanning technology for broader accessibility.

REFERENCES

[1] Hansard, Miles, et al. Time-of-flight cameras: principles, methods and applications. Springer Science & Business Media, 2012.
[2] Tóth, Teodor, and Jozef Živčák. "A comparison of the outputs of 3D scanners." Procedia Engineering 69 (2014): 393-401.
[3] Cui, Yan, et al. "3D shape scanning with a time-of-flight camera." 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010.
[4] Vieira, Gustavo, et al. "Low-cost industrial controller based on the raspberry pi platform." 2020 IEEE international conference on industrial technology (ICIT). IEEE, 2020.
[5] França, João Guilherme DM, et al. "A 3D scanning system based on laser triangulation and variable field of view." IEEE International Conference on Image Processing 2005. Vol. 1. IEEE, 2005.
[6] Lewellen, Tom K. "Time-of-flight PET." Seminars in nuclear medicine. Vol. 28. No. 3. WB Saunders, 1998.
[7] Heesom, David, and Lamine Mahdjoubi. "Trends of 4D CAD applications for construction planning." Construction management and economics 22.2 (2004): 171-182.

[8] Varsha, J., and Parimala Prabhakar. "Enhanced Soldier Terrain Visualization System Using Image Processing and Time of Flight Cameras with Transparent 5G Antenna." 2024 5th International Conference on Circuits, Control, Communication and Computing (I4C). IEEE, 2024.

[9] Zhou, Qian-Yi, Jaesik Park, and Vladlen Koltun. "Open3D: A modern library for 3D data processing." arXiv preprint arXiv:1801.09847 (2018).

[10] UZUN, Yusuf, and Ozgur DUNDAR. "Raspberry Pi and Models." Programmable Smart Microcontroller Cards (2021): 34.

[11] Cui, Yan, et al. "3D shape scanning with a time-of-flight camera." 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010.

[12] Francis, Sobers Lourdu Xavier, et al. "A ToF-camera as a 3D vision sensor for autonomous mobile robotics." International Journal of Advanced Robotic Systems 12.11 (2015): 156.

[13] May, Stefan, et al. "Robust 3D-mapping with time-of-flight cameras." 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009.

[14] Dong, Wei, et al. "GPU accelerated robust scene reconstruction." 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019.

[15] Zhou, Ding-Xuan. "Theory of deep convolutional neural networks: Downsampling." Neural Networks 124 (2020): 319-327.