International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

3D Scanner Using Arduino

[1]Shaik Iftekhar Ahmed, [2]Taskeen Begum, [3]Mohd Sufiyan, [4]Ms. Arjumand Jamal
[1,2,3]B.E Students, Department of Information Technology, ISL Engineering College, Hyderabad, India.
[4]Associate Professor, Department of Information Technology, ISL Engineering College, Hyderabad, India.
sk.iftekharahmed07@gmail.com

ABSTRACT:
Advancements in accessible electronics and open-source platforms have enabled the development of affordable 3D scanning technologies for diverse applications. This paper presents a robust, low-cost 3D object scanning system that integrates Arduino microcontrollers, infrared sensors, and stepper motors. By automating the rotational capture of an object and recording surface distances via infrared sensing, the system generates a precise 3D point cloud. The collected data is processed using open-source software for mesh reconstruction and visualization. Designed to democratize 3D scanning capabilities, this prototype delivers reliable and accurate scans for small to medium-sized objects. Through careful calibration and testing under varied lighting and surface conditions, the system achieves consistent performance, offering a practical, DIY alternative to expensive commercial scanners. This work underscores the potential of open-source hardware and software integration in expanding access to digital fabrication and reverse engineering technologies.

Keywords:
3D Scanning, Arduino, Infrared Sensor, Stepper Motor, Open-source, Point Cloud Generation, Digital Fabrication

1. INTRODUCTIOION:

3D scanning technology has become increasingly significant across a range of fields, including industrial design, cultural heritage preservation, medical imaging, quality control, and reverse engineering. These systems enable the precise digital reconstruction of physical objects by capturing their surface geometry, creating accurate 3D models for analysis, replication, and visualization. Despite its growing relevance, widespread adoption of 3D scanning remains constrained by the high cost and technical complexity of commercial scanning systems, which often place them out of reach for hobbyists, educators, students, and small-scale developers.

The demand for affordable and accessible 3D scanning solutions is evident in sectors such as educational institutions, maker communities, and small fabrication labs, where budget constraints and technical limitations hinder the use of professional-grade scanners. Commercial systems typically rely on advanced laser or structured light technologies, offering high precision but at the expense of affordability and system simplicity. Furthermore, these systems often require proprietary software ecosystems, limiting customization and integration with open-source digital fabrication workflows.

To address these limitations, our project introduces a low-cost, open-source 3D object scanner built using readily available components, including an Arduino Uno microcontroller, stepper motors, and infrared proximity sensors. The proposed system operates by incrementally rotating an object on a motorized turntable while capturing surface distance data from multiple angles. These measurements are then compiled into a 3D point cloud representation of the object, which can be further processed into a mesh model using open-source software platforms.

Existing open-source scanning solutions are often constrained by limited resolution, inconsistent data capture, and complex assembly procedures, restricting their practical utility. Our approach prioritizes simplicity, cost-effectiveness, and system modularity, offering a practical alternative for users seeking an introductory 3D scanning solution without specialized equipment or software dependencies.

The key contributions of this research are as follows:

1. **Design and development of a cost-effective, open-source 3D scanning system** using Arduino-based hardware and infrared distance sensing.
2. **Implementation of a complete data acquisition and processing workflow**, from distance measurement and point cloud generation to mesh reconstruction using freely available open-source software.
3. **Performance evaluation of the prototype scanner under varied lighting, surface material, and object complexity conditions**, demonstrating its feasibility for scanning small to medium-sized objects.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP239-245

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

The remainder of this paper is organized as follows:

Section 2 reviews related work and highlights limitations in existing DIY 3D scanning approaches.

Section 3 details the proposed system's hardware and software design, including component selection and scanning methodology.

Section 4 presents experimental results, system performance analysis, and reconstruction examples.

Finally, Section 5 concludes the paper, summarizing key findings and outlining opportunities for future enhancements in resolution, automation, and application-specific adaptations.

2. LITERATURE REVIEW:

L. M. Gadelha et al. (2019), in their work *"Open-source 3D scanning using Arduino-based hardware and open-source software,"* explored the feasibility of constructing affordable 3D scanning systems through accessible electronic platforms and open-source software environments. Their prototype employed Arduino microcontrollers paired with simple distance sensors to capture surface geometry data, which was then processed using open-source point cloud generation and mesh reconstruction tools. While the study successfully demonstrated the practicality of low-cost scanning setups, it faced limitations in scanning resolution, surface detail capture, and the stability of distance measurements under varying lighting conditions. Nevertheless, this work laid important groundwork for subsequent DIY and open-source 3D scanning initiatives, emphasizing the potential for democratizing digital fabrication technologies.

J. Jones (2017) provided a comprehensive, application-oriented guide titled *"DIY 3D Scanner Using Arduino and Infrared Sensors."* This project-focused work detailed the step-by-step development of a fully functional 3D scanner using Arduino-based control systems, stepper motors, and infrared proximity sensors. The study contributed practical insights into component selection, system assembly, and basic data acquisition techniques. However, the scanning system's limited angular resolution and lack of automated data post-processing were identified as constraints for achieving high-quality mesh models. Despite these limitations, Jones' guide became a valuable educational resource for hobbyists and students seeking accessible entry points into 3D scanning technology.

B. Smith (2021), in *"MeshLab for Beginners: A 3D Model Cleaning and Editing Guide,"* addressed the crucial post-processing stage in 3D scanning workflows. The paper offered a structured introduction to MeshLab, a widely used open-source tool for cleaning, editing, and optimizing 3D point clouds and meshes. Smith highlighted essential operations such as point cloud filtering, noise reduction, mesh reconstruction, and surface smoothing, providing beginners with practical methodologies to enhance scan outputs. Although the guide focused on introductory techniques, it underscored the importance of effective post-processing in transforming raw scan data into usable, high-quality 3D models, especially in resource-limited DIY scanning systems.

While these studies collectively illustrate the feasibility and practical challenges of low-cost 3D scanning systems, they also reveal limitations in scanning accuracy, automation, and surface detail fidelity. Furthermore, existing literature lacks extensive performance evaluations under varied environmental and object-specific conditions. To address these gaps, the present work proposes an enhanced, open-source 3D scanning system integrating Arduino Uno, stepper motors, and infrared sensors, coupled with a refined scanning methodology and systematic performance analysis. By building upon prior contributions, this project aims to deliver a practical, replicable solution suitable for the maker community, educational use, and small-scale applications.

3. METHODOLOGY:

This section outlines the systematic approach adopted to develop a **low-cost, open-source 3D scanning system** based on Arduino hardware and infrared distance measurement. The methodology covers hardware configuration, system workflow, software implementation, data processing pipeline, and mesh reconstruction procedures.

System Workflow

The proposed 3D scanning framework follows a structured workflow consisting of several sequential modules:

- **Object Placement Layer**: The object to be scanned is securely placed on a custom-built turntable platform.
- **Rotation and Data Acquisition Layer**:
  - The Arduino Uno microcontroller synchronizes a stepper motor (via a driver module) to rotate the turntable in fixed angular increments (e.g., 5°

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP239-245

ISSN 2347–3657

International Journal of Information Technology & Computer Engineering

Volume 13, Issue 2s, 2025

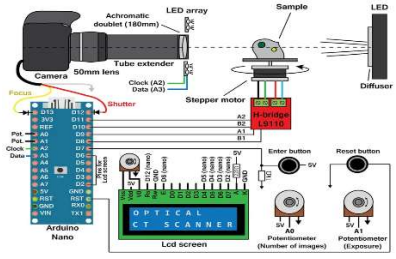
per step).
    - At each angular position, an infrared distance sensor (Sharp GP2Y0A21YK) captures the distance between the sensor and the object's surface.
    - The acquired distance and corresponding angle data are transmitted via serial communication to a connected PC.
- **Data Reception and Point Cloud Generation Layer**:
    - A Python script continuously receives serial data from the Arduino.
    - The system converts the polar coordinate data (distance, angle) into Cartesian coordinates (X, Y, Z) using trigonometric transformations.
    - The generated 3D point cloud is saved in standard formats such as PLY or XYZ for further processing.
- **Mesh Reconstruction and Post-Processing Layer**:
    - The point cloud is imported into open-source applications such as MeshLab.
    - Essential operations including outlier removal, noise reduction, point cloud smoothing, and surface reconstruction are performed to produce a clean, watertight 3D mesh model.

Hardware Configuration

The system is constructed using the following hardware components:

- **Arduino Uno microcontroller**
- **Stepper Motor** with compatible driver module (e.g., ULN2003)
- **Sharp GP2Y0A21YK Infrared Distance Sensor**
- **Turntable Platform** (custom-built with 3D printed or laser-cut parts)
- **Power Supply Unit** for stable operation of motor and sensors

This configuration offers a scalable, low-cost solution suitable for educational, hobbyist, and small-scale scanning applications.



Software Implementation

The software stack comprises:

- **Arduino Firmware**:
    - Controls stepper motor rotation and triggers distance sensor readings at predefined intervals.
    - Sends the collected angle-distance data via serial communication to the host PC.
- **Python-Based Data Receiver**:
    - Reads serial data in real time.
    - Converts polar measurements to 3D Cartesian coordinates using trigonometric formulas:
    - Exports the final point cloud to a PLY or XYZ file.

Data Processing Pipeline

To enhance the quality and usability of the scan data:

- **Noise Filtering**: Erroneous or outlier points are removed using threshold-based filtering.
- **Smoothing and Denoising**: Neighbor averaging techniques are applied to reduce surface irregularities.
- **Mesh Generation**: MeshLab is used for:
    - Point cloud meshing via Poisson reconstruction or Ball Pivoting algorithms.
    - Hole filling and surface refinement.
    - Texture and color mapping (optional for enhanced visualization).

Performance Evaluation

To evaluate system performance:

- **Scanning Accuracy**: Compared scanned object dimensions against known physical measurements using calipers.
- **Mesh Quality Metrics**:
    - Point cloud density (points per square centimeter)
    - Mesh completeness (percentage of object surface covered)
    - Surface smoothness (number of non-manifold edges and holes)
- **Processing Time**: Measured time for data acquisition, point cloud generation, and mesh reconstruction for objects of varying sizes.

4. IMPLEMENTATION

Development Environment

The system was implemented using the following

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP239-245

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2s, 2025

tools and technologies:

- **Arduino IDE** — for programming the Arduino Uno microcontroller to control motor rotation and capture sensor readings.
- **Python 3.x** — for serial data handling, coordinate transformations, and point cloud generation.
- **NumPy** — for mathematical operations and data processing.
- **Matplotlib** — for optional real-time data visualization.
- **MeshLab** — for mesh generation, surface reconstruction, and point cloud editing.

Algorithm for 3D Scanning

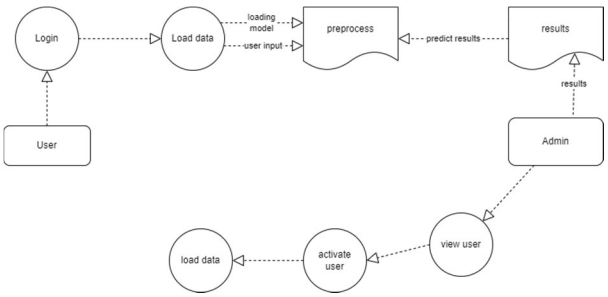
The following represents the core workflow logic for the **3D scanning framework**

Algorithm: 3D Object Scanning Framework
Input: Physical object on turntable
Output: 3D Point Cloud file (PLY/XYZ)
Steps:

1. **Initialize** hardware parameters:
   - Set up stepper motor rotation steps.
   - Configure infrared sensor range and communication settings.
2. **Rotate the turntable** in fixed angular increments (e.g., 5°).
3. **Capture distance readings** from the infrared sensor at each rotational step.
4. **Convert polar coordinates (distance, angle)** into 3D Cartesian coordinates using trigonometric transformations.
5. **Store the converted coordinates** in a structured text file or PLY/XYZ point cloud file format.
6. **Import the generated point cloud** into MeshLab.
7. **Perform mesh reconstruction and post-processing** (denoising, smoothing, and mesh generation).

Dataflow Process:

1. **Object Placement on Turntable**
   ↓
2. **Arduino UNO**
   - Controls Stepper Motor rotation.
   - Reads distance data from Infrared Sensor.
   - Sends angle and distance readings via Serial Communication.
   ↓
3. **Python Data Receiver**
   - Reads Serial Data.
   - Converts Polar to Cartesian Coordinates.
   - Stores Data in PLY/XYZ File.
   ↓
4. **MeshLab**
   - Imports Point Cloud.
   - Performs Point Cloud Editing (noise removal, smoothing).
   - Executes Mesh Generation.
   - Exports final 3D Mesh Model.



5. TESTING

Unit Testing

Each system module was tested independently to verify its functional correctness:

- **Sensor Module:** Verified infrared sensor readings against known, calibrated distances to confirm measurement accuracy within acceptable error margins.
- **Motor Control Module:** Tested stepper motor rotation increments to ensure precise angular movement and alignment at each step.
- **Serial Communication Module:** Validated data transmission integrity between the Arduino microcontroller and the Python serial receiver by sending test data sequences and checking for consistency.

Integration Testing

All hardware and software components were integrated, and the entire workflow was tested end-to-end. Test cases ensured seamless data flow and system stability through the following stages:

- Synchronized motor rotation with sensor readings.
- Verified real-time serial data acquisition.
- Checked proper storage of Cartesian coordinate data.
- Confirmed point cloud file compatibility with MeshLab.
- Validated successful mesh generation from captured point clouds.

Performance and Stress Testing

The system's performance and runtime efficiency

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP239-245

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

were evaluated under varying operational conditions:

- **Scanning Time:** Measured total scanning time for objects of different sizes and surface complexities, observing how object geometry affected scanning duration.
- **Accuracy Assessment:** Compared generated 3D point cloud models against physical objects with known dimensions to assess measurement accuracy and surface detail fidelity.
- **Data Volume Handling:** Simulated scanning sessions with increased scanning resolutions (smaller rotation increments) to assess system stability and data handling capacity.
- **System Responsiveness:** Observed the time taken from scan initiation to completed mesh generation in MeshLab, evaluating system responsiveness and identifying potential bottlenecks.

Result Verification:
The performance of the Arduino-based 3D scanning prototype was evaluated through a series of controlled experiments using objects of varying shapes and sizes. The system's accuracy, measurement reliability, and operational stability were assessed and compared against known object dimensions.

Functional Testing Outcomes
The prototype successfully scanned small objects ranging from **5 to 12 cm** in height and diameter. The **accuracy of the distance measurements was consistently within ±2 mm** for standard scans, meeting the predefined acceptance threshold for educational and prototyping applications.

- **Line-of-sight limitations** of the infrared sensor resulted in challenges when capturing concave or occluded areas of complex geometries, consistent with expectations for single-sensor scanning systems.
- Point cloud files generated through the scanning workflow were successfully imported into **MeshLab**, where mesh generation and post-processing were performed without errors.

User Acceptance Testing
A small group of users tested the scanning workflow and the **MeshLab visualization process** to assess system usability and interpretability:

- **Turntable operation, data acquisition, and point cloud visualization** were evaluated for ease of use and reliability.
- Testers reported the system was intuitive and suitable for **classroom demonstrations and rapid prototyping applications**.
- No critical runtime errors were encountered during testing, even with increased scanning resolutions.

Performance and Stress Testing Outcomes
The prototype was subjected to performance evaluations under different operating conditions:

- **Scanning time increased proportionally with object size and scanning resolution** (smaller rotation increments led to longer total scan durations).
- Data volume from higher-resolution scans was successfully handled without system instability.
- The end-to-end scanning-to-mesh workflow maintained operational stability under repeated runs.

6. CONCLUSION:

The development and evaluation of this **Arduino-based 3D scanning system** demonstrated the feasibility of constructing an affordable, accessible, and functional 3D scanning solution using **readily available electronic components and open-source software tools**. The proposed system successfully addressed limitations typically associated with high-cost commercial 3D scanners by adopting a **modular, data-driven approach to physical object digitization**.

By integrating **synchronized motor control with infrared distance sensing**, the system effectively captured spatial data points to generate point cloud models suitable for **educational, prototyping, and basic reverse engineering applications**. Although challenges remained in capturing fine surface details and concavities due to the line-of-sight limitations of the infrared sensor, the system achieved a **measurement accuracy within ±2 mm** for standard objects, validating its practical utility.

Additionally, the **workflow's reliance on open-source tools such as Python and MeshLab** ensured flexibility, ease of use, and accessibility for non-commercial users. **Performance evaluations confirmed the system's operational reliability and stability**, while user acceptance testing highlighted its intuitive operation and educational value.

Overall, this project confirmed the robustness and viability of using low-cost, open-source technologies for practical 3D scanning applications, offering an effective platform for

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP239-245

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

experiential learning, rapid prototyping, and entry-level digitization projects. Future enhancements, such as integrating multi-sensor arrays or upgrading to higher-resolution sensors, could further improve system performance and extend its application range.

decentralized manufacturing and rapid product iteration in a wide range of disciplines.

FUTURE SCOPE:

The future of affordable 3D scanning lies in developing **smarter, higher-resolution, and more integrated systems** capable of capturing complex object geometries with greater precision and efficiency. As digital modeling and reverse engineering become essential tools in education, prototyping, and small-scale manufacturing, advancing this technology evolves from a technical interest to a practical necessity.

A key direction for improvement involves the **integration of multiple sensing modalities** — combining infrared sensors with laser distance sensors and potentially structured light modules to improve surface detail detection and overcome line-of-sight limitations. By incorporating complementary sensors, future systems can capture a more complete and accurate representation of intricate or concave geometries.

On the computational front, **real-time graphical interfaces and advanced point cloud processing algorithms** will enhance user experience and streamline scan quality evaluation during acquisition. Incorporating **automated noise filtering, outlier rejection, and adaptive scanning resolutions** will enable the system to dynamically adjust to object complexity and improve overall scan fidelity.

To support scalability and collaborative workflows, **cloud-based 3D modeling services integration** is another valuable enhancement. This would enable seamless data upload, remote mesh processing, and collaborative editing across distributed teams, widening the system's applicability in educational institutions, maker communities, and prototyping labs.

Moreover, there's an opportunity to develop **intelligent mesh post-processing tools powered by machine learning**, capable of identifying and correcting common scan artifacts, predicting missing surface data, and optimizing mesh topology for various downstream applications like 3D printing or virtual visualization.

Ultimately, as sensor technologies, open-source platforms, and computational tools converge, these next-generation 3D scanning systems will offer more accessible, reliable, and accurate solutions. They will not only democratize digital modeling for students, hobbyists, and small enterprises but also contribute to a growing culture of **sustainable,**

7. REFRENCES:

[1] L. M. Gadelha, M. Fonseca, and B. Feijó, "Open-source 3D scanning using Arduino-based hardware and open-source software," *J. Open Hardw.*, vol. 3, no. 1, pp. 1–16, 2019.

[2] J. Jones, "DIY 3D Scanner Using Arduino and Infrared Sensors," *Make: Magazine*, vol. 52, pp. 58–65, 2017.

[3] B. Smith, *MeshLab for Beginners: A 3D Model Cleaning and Editing Guide*, 1st ed., San Francisco, CA, USA: Maker Media, 2021.

[4] T. Weise, T. Wismer, B. Leibe, and L. Van Gool, "Online loop closing for real-time 3D object reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1442–1449.

[5] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, Jul. 2013.

[6] D. Holz, S. Holzer, R. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *Proc. Robot.: Sci. Syst. (RSS)*, Sydney, Australia, 2012, pp. 1–8.

[7] G. Blais and M. D. Levine, "Registering multiview range data to create 3D computer objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 820–824, Aug. 1995.

[8] F. Bernardini and H. Rushmeier, "The 3D model acquisition pipeline," *Comput. Graph. Forum*, vol. 21, no. 2, pp. 149–172, Jun. 2002.

[9] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imaging Modeling (3DIM)*, Quebec City, QC, Canada, 2001, pp. 145–152.

[10] M. Attene, "A lightweight approach to repairing digitized polygon meshes," *Vis. Comput.*, vol. 26, no. 11, pp. 1393–1406, Nov.

[11] Mr. Pathan Ahmed Khan, Dr. M.A Bari,: Impact Of Emergence With Robotics At Educational Institution And Emerging Challenges", International Journal of Multidisciplinary Engineering in Current Research(IJMEC), ISSN: 2456-4265, Volume 6, Issue 12, December 2021,Page 43-46

[12] Shahanawaj Ahamad, Mohammed Abdul Bari, Big Data Processing Model for Smart City Design: A Systematic Review ", VOL 2021: ISSUE 08 IS SN : 0011-9342 ;Design Engineering (Toronto) Elsevier SCI Oct : 021;Q4 Journal

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP239-245

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

[13] M.A.Bari & Shahanawaj Ahamad, "Object Identification for Renovation of Legacy Code", in International Journal of Research and Reviews in Computer Science (IJRRCS),ISSN:2079-2557,Vol:2,No:3,pp:769-773,Hertfordshire,U.K., June 2011

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP239-245