Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 46 No. 04 (2025)

Efficient 3D Artifact Modeling in the Field: A Simplified Photogrammetry Approach

Nagesh B V[1], Balasubramanya H S[2], Durgesh Datar [3], Prasanna Kumara S G [4], Yogesha Lakkegowda[5*]

[1]Department of Physics, Ramaiah Institute of Technology, Bangalore-560054

[2]Department of Mechanical Engineering, Ramaiah Institute of Technology, Bangalore-560054

[3]Department of Physics, Navkis college of Engineering, Hassan-573217

[4]Department of Physics, Government Science College (Autonomous), Hassan-573201

[5]Department of Physics, Government First Grade College, Tiptur-572201

[*]Corresponding author: yogesh.bub@gmail.com

Abstract:

Photogrammetry is a 3D scanning technique that uses a series of 2D photos to create a virtual 3D model of an object. This technique involves measuring and analyzing features from multiple photographs, with photogrammetry software combining these to generate a detailed 3D model. The photogrammetry process is intricate and demands a certain level of expertise to accurately scan an object in 3D. It typically requires manual photography from various angles, with the object needing proper lighting to ensure quality results. While the process can be executed with multiple cameras, this approach can be costly. To simplify and automate photogrammetry, a specialized device called a photogrammetry rig has been developed. This rig automates the photogrammetry process by placing the object on a turntable. The device rotates the object at predefined angles while a camera captures a series of images. After completing one full rotation, the camera shifts to a new angle and the process continues. The captured images are then uploaded to software, which processes them into a high-quality 3D model. By automating the photogrammetry process, this rig makes 3D scanning more accessible and affordable, allowing users to obtain high-quality scans of various objects without requiring specialized knowledge. This technology has broad applications across numerous fields globally.

Keywords: 3D scanner, CATIA, Arduino UNO, NEMA 17, Autodesk ReCap

INTRODUCTION

1.1 3D Scanning

3D scanning is a process of determining the shape of an object's surface or its volume in three-dimensional space. By collecting information about the real-world object using a 3D scanning device, this makes 3D measurement and 3D visualization possible. Accurate 3D measurements derived from a scanned object are useful for material inspection and quality control. If a 3D scanning technology is capable of collecting a lot of 3D data from the scanned object, it has the ability to recreate a high resolution, accurate 3D digital model of the real-world object. The device used to capture 3D data is called as a 3D scanner. [1]

Collected 3D data is useful for a wide variety of applications. These devices are used extensively by the entertainment industry in the production of movies and video games, including virtual reality. Other common applications of this technology include augmented reality, motion capture, gesture recognition, industrial design, orthotics and prosthetics, reverse engineering and prototyping, quality control/inspection and the digitization of cultural artefacts. [2]

1.2 TYPES OF 3D SCANNING TECHNOLOGIES

3D scanning is a technique used to capture the shape and volume of an object's surface in three-dimensional space. It involves a 3D scanning device that gathers detailed data from a physical object, enabling precise measurements and visualization [3]. The high-accuracy data obtained from 3D scanning is essential for tasks such as material inspection and quality control. When the technology captures extensive data, it can generate a high-resolution, accurate digital 3D model of the object. The device responsible for this is called a 3D scanner. The collected 3D data has a wide range of applications across different fields. In the entertainment industry, 3D scanning is widely used for movie and video game production, including virtual reality. Additional applications include augmented reality, motion capture, gesture recognition, industrial design, orthotics and prosthetics, reverse engineering, prototyping, quality control and inspection, as well as the preservation and digitization of cultural artifacts [4].

1.3 Types of 3D Scanning Technologies

Different 3D scanning technologies are available, each with its own advantages, limitations, and costs:

Laser Triangulation: This technology involves projecting a laser beam onto the object's surface and measuring the deformation of the laser ray to gather 3D data. Laser 3D scanners are popular due to their ease of use and provide reasonable accuracy for general applications.

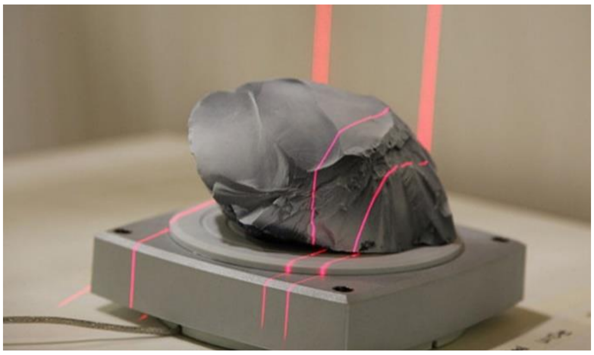

Figure 1: Laser Triangulation

Structured Light: This technology uses a light pattern projected onto the object to measure its shape. The setup includes an LCD projector and an array of cameras [5]. The light pattern, such as alternating dark and light horizontal stripes, is projected onto the object. The scanners capture the object's geometry by observing how the pattern deforms across different points within the camera's field of view.

Figure 2: Structured Light

Photogrammetry is a technique that uses a camera to capture numerous images of an object from different angles [6]. In certain setups, multiple cameras can be used to take all the images simultaneously. These 2D photographs are then processed with specialized software and computational geometry algorithms to reconstruct the object's geometry, creating a virtual 3D model.

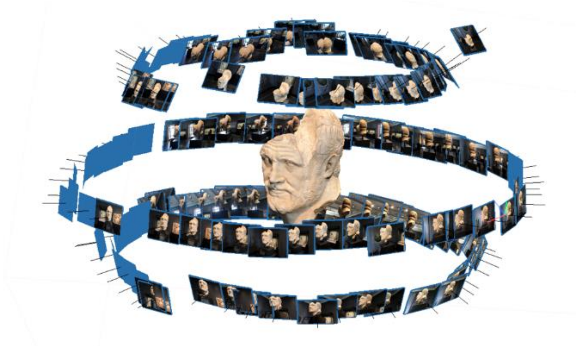

Figure 3: Photogrammetry

1.4 THE PROCESS OF PHOTOGRAMMETRY

Photogrammetry is typically conducted using a single camera that moves around the object to capture images from different angles. Alternatively, the object can be rotated at regular intervals while the camera remains stationary.

In a photogrammetry studio, a large number of cameras are arranged at specific angles, with proper lighting directed at the object to ensure optimal photo quality [7].

The general photogrammetry process involves capturing photos from all angles of the object (360°), uploading these images to photogrammetry software, and generating a 3D model of the object.

The photogrammetry process can be broken down into the following steps:

1. **Setting Up the Object:** Position the object in the designated area, ensuring it is properly prepared for scanning.

2. **Setting Up the Lighting:** Arrange the lighting to illuminate the object evenly and minimize shadows.

3. **Taking Photographs:** Capture a series of photographs from multiple angles, either by moving the camera around the object or rotating the object while keeping the camera stationary.

4. **Uploading Photos to the Software:** Transfer the collected images to photogrammetry software.

5. **Generating a 3D Model:** Use the software to process the images and create a detailed 3D model of the object.

2. DESIGN, FABRICATION AND ASSEMBLY

The automated photogrammetry rig is designed using CATIA software. CATIA is a leading 3D CAD design software renowned for its engineering and design capabilities. It is employed across various industries, including aerospace, automotive, consumer goods, and industrial machinery, for tasks such as design, simulation, analysis, and manufacturing. CATIA caters to a broad range of manufacturing organizations, from original equipment manufacturers (OEMs) and their supply chains to small independent producers. [8]

2.1 PART DESIGN

The parts of the automated photogrammetry rig are designed using CATIA software and is as follows:

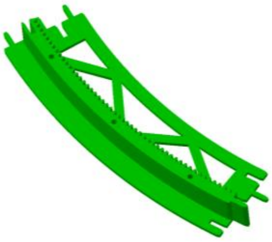

Figure 4: Part 1

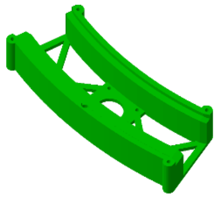

Figure 5: Part 2

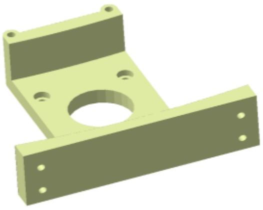

Figure 6: Part 3

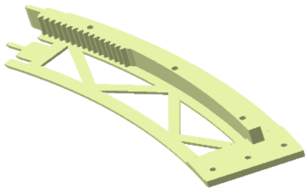

Figure 7: Part 4

2.2 FABRICATION

The parts are fabricated using Fused Deposition Modeling (FDM) 3D printing technique.

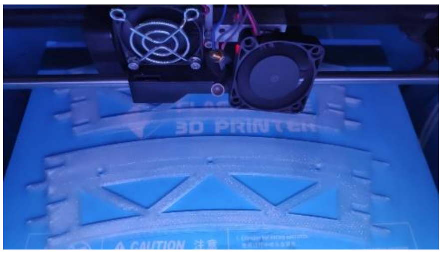

Figure: 3D printing of the parts by FDM method

2.3 ASSEMBLY

The 3D printed parts are assembled and glued together.

Figure: Final assembly of automated photogrammetry rig

2.4 PARTS, COMPONENTS AND PROGRAM

Arduino UNO

The Arduino UNO is an open-source microcontroller board built around the Microchip ATmega328P, created by Arduino.cc. It offers a range of digital and analog input/output (I/O) pins, allowing it to work with various shields and circuits. The Arduino UNO has 14 digital pins and 6 analog pins [9]. It can be programmed through the Arduino IDE (Integrated Development Environment) using a type B USB cable. The board can be powered either via the USB connection or an external 9-volt battery, with a voltage input range of 7 to 20 volts.

Fig.1 Arduino UNO

L298N Dual H Bridge Motor Driver

The L298N Dual H-Bridge Motor Driver is a motor controller breakout board commonly used for regulating motor speed and direction. It can also be applied to control brightness in lighting projects, such as high-powered LED arrays. An H-Bridge is a circuit that allows current to flow in both directions and can be controlled through pulse width modulation (PWM).

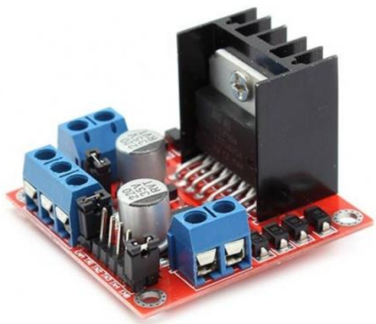

Figure 2: L298N Dual H Bridge Motor Driver

NEMA 17 STEPPER MOTOR

This NEMA 17-size hybrid stepper motor can operate as either a unipolar or bipolar motor and has a 1.8° step angle, translating to 200 steps per revolution. It draws 1.2 A per phase at 4 V, producing a holding torque of 3.2 kg-cm. The motor features six color-coded wires with bare ends, enabling control by both unipolar and bipolar drivers. For unipolar control, all six leads are connected, while for bipolar operation, the yellow and white center-tap wires are left disconnected, with the red-blue pair connected to one coil and the black-green pair to the other.

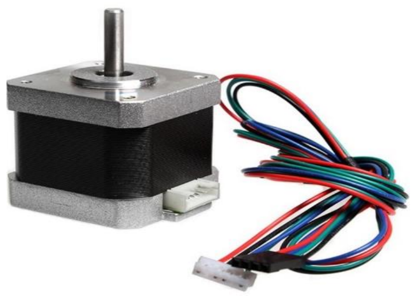

Figure : Stepper motor

CIRCUIT ASSEMBLY

The circuit is designed to control stepper motors using an Arduino board. It includes two L298N drivers, enabling control of two stepper motor axes (X and Y). The Arduino board is responsible for uploading the program from the Arduino IDE software, which operates the system.

2.5 PROGRAM

The program to operate the device is as follows:

```
#include <Stepper.h>
//const int stepsPerRevolution = 200;
// change this to fit the number of steps per revolution
// for your motor
const int stepsPerRevolution1 = 25;
const int stepsPerRevolution2 = 200;
//const int stepsPerRevolution3 = 200;
//const int stepsPerRevolution = 200;
// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution1, 8, 9, 10, 11);
Stepper myStepper2(stepsPerRevolution2, 4, 5, 6, 7);
void setup()
{
 // set the speed at 60 rpm:
 myStepper.setSpeed(100);
   myStepper2.setSpeed(100);
 // initialize the serial port:
 Serial.begin(9600);
}
void loop()
{
       take360pic();
       //myStepper2.step(-stepsPerRevolution2);
        ringmotor();
        ringmotor();
        ringmotor();
        ringmotor();
       delay(500);
       take360pic();
              //myStepper2.step(-stepsPerRevolution2);
```

```
 ringmotor();
 ringmotor();
 ringmotor();
 ringmotor();
 delay(500);
 take360pic();
// myStepper2.step(-stepsPerRevolution2);
 ringmotor();
 ringmotor();
 ringmotor();
 ringmotor();
 delay(500);
 take360pic();
// myStepper2.step(-stepsPerRevolution2);
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
 ringmotor1();
// delay(8000);
 // ringmotor1();
 //delay(2000);
// take360pic();
// myStepper2.step(-stepsPerRevolution2);
//  ringmotor();
/// delay(2000);
// take360pic();
// myStepper2.step(-stepsPerRevolution2);
```

```
      //  ringmotor();
      // delay(2000);
      // take360pic();
      // myStepper2.step(stepsPerRevolution3);
       //delay(6000);
  // step one revolution  in one direction:
  //Serial.println("clockwise");
  //myStepper.step(stepsPerRevolution);
  //delay(500);
  // step one revolution in the other direction:
 // Serial.println("counterclockwise");
  //myStepper.step(-stepsPerRevolution);
  //delay(500);
}
void take360pic() // the process I made to take pictures
{
  myStepper.step(stepsPerRevolution1);
  delay(1000);
  myStepper.step(stepsPerRevolution1);
  delay(1000);
  myStepper.step(stepsPerRevolution1);
  delay(1000);
  myStepper.step(stepsPerRevolution1);
  delay(1000);
  myStepper.step(stepsPerRevolution1);
  delay(1000);
  myStepper.step(stepsPerRevolution1);
  delay(1000);
  myStepper.step(stepsPerRevolution1);
  delay(1000);
  myStepper.step(stepsPerRevolution1);
  delay(1000);
}
void ringmotor()
{
```

```
  myStepper2.step(-stepsPerRevolution2);
// delay(2000);
 // myStepper2.step(-stepsPerRevolution2);
  //delay(5000);
 // myStepper2.step(-stepsPerRevolution2);
  //delay(5000);
  //myStepper2.step(-stepsPerRevolution2);
 // delay(5000);
}
void ringmotor1()
{
    myStepper2.step(stepsPerRevolution2);
  //delay(8000);
//myStepper2.step(stepsPerRevolution2);
  //delay(2000);
 // myStepper2.step(stepsPerRevolution2);
 //delay(2000);
  //myStepper2.step(stepsPerRevolution2);
  //delay(2000);
}
```

2.5 Working operation of automated photogrammetry rig

The automated photogrammetry rig captures images of an object from multiple angles. The object is placed on a rotating platform, and the camera begins by moving to a top position to start taking photographs. After each shot, the platform rotates to showcase different views of the object. Once photos from one position are captured, the camera shifts to a new position, repeating the process. After the full set of images is taken, they are uploaded into photogrammetry software for processing. Autodesk ReCap, short for "Reality Capture," is software designed for processing native point clouds generated from photogrammetry scans. It offers an easy-to-use interface, enabling users to open point cloud files and apply customizable import settings to filter data for more efficient file management. Since ReCap [10] is part of the Autodesk ecosystem, point cloud data can be seamlessly extracted or imported into other Autodesk applications. This technology is particularly useful in mechanical and manufacturing industries, allowing users to create virtual 3D models of existing parts through reality capture. Additionally, new parts can be overlaid on the scanned models for precise size, placement, and tolerance matching, all with minimal effort.

Figure: Virtual 3D scanned model of the object

CONCLUSION

The automated photogrammetry rig streamlines the 3D scanning process, allowing users to achieve accurate results using photogrammetry without needing extensive expertise. This setup is more cost-effective compared to other 3D scanning technologies that often require expensive hardware and software.

By automating the process of capturing images from all angles, the rig addresses common challenges of photogrammetry, such as complexity, accuracy, and the need for human involvement. With minimal human intervention, the system reduces errors and enhances the precision of the captured data. The automation also shortens the time required for 3D scanning, producing more accurate results than manual photogrammetry.

This device makes 3D scanning through photogrammetry more accessible and affordable, while eliminating much of the complexity. It can be used across a wide range of applications, including reverse engineering, structural analysis, archaeology, biomedical fields, virtual and augmented reality, art, and 3D printing.

REFERENCES

[1] T. Schenk - Ohio State University, Object recognition in Digital Photogrammetry, Photogrammetric Record, Vol. 16(95), Pg. 743 – 762, 2000.

[2] L.M. Galantucci, R. Ferrandes, G. Percoco, T Digital Photogrammetry for Facial Recognition, Transactions of the ASME, Vol. 6, Pg. 390 – 396, 2006.

[3] Frantisek Pivnicka, Vladimir Cervenka, Karel Charvat, Ales Limpouch, Cost-Effective Digital Photogrammetry, International Archives of photogrammetry and remote sensing, Vol. XXXI, Pg. 300 – 306, 1996.

[4] Armin Gruen, Devrim Akca, Mobile Photogrammetry, DGPF Tagungsband, Vol.16, Pg. 441 – 451, 2007.

[5] Andre Streilein, Urs Hirschberg, Integration of digital photogrammetry and CAAD, Procedia CIRP, Presented paper on the CAAD Futures '95 International Conference, Pg. 1-14, 1995.

[6] Bashar Sa. Al-Jaboori, Hussein Sabah Jaber, The feasibility of close range photogrammetry in three dimensional modeling, The Iraqi Journal for Mechanical and Material Engineering, Pg. 30 – 40, 2018.

[7] A. Sh. Amini, M. Vrashosaz, M. Saadatseresht, Deformation determination of aircraft parts by photogrammetry, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVII, Part B5, 2008.

[8] D. Gajski, A. Solter, M. Gasparovic, Applications of macro photogrammetry in archaeology, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B5, 2016.

[9] J. Valencia, E.N.B.S. Julio, H.J. Araujo, Applications of photogrammetry to structural assessment Abstract, Society for Experimental Mechanics, Vol.36, 2011.

[10] Amaia Hernandez, Edward Lemaire, A smartphone photogrammetry method for digitizing prosthetic socket interiors, Prosthetics and Orthotics International, Pg.1 – 5, 2016.