

12-2023

Gaze Tracking Embedded Collaborative Robots for Automated Metrology and Reverse Engineering

Sachithra H. Karunathilake
The University of Texas Rio Grande Valley

Follow this and additional works at: <https://scholarworks.utrgv.edu/etd>



Part of the [Industrial Engineering Commons](#)

Recommended Citation

Karunathilake, Sachithra H., "Gaze Tracking Embedded Collaborative Robots for Automated Metrology and Reverse Engineering" (2023). *Theses and Dissertations*. 1439.
<https://scholarworks.utrgv.edu/etd/1439>

This Thesis is brought to you for free and open access by ScholarWorks @ UTRGV. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ScholarWorks @ UTRGV. For more information, please contact justin.white@utrgv.edu, william.flores01@utrgv.edu.

GAZE TRACKING EMBEDDED COLLABORATIVE ROBOTS FOR AUTOMATED
METROLOGY AND REVERSE ENGINEERING

A Thesis

By

SACHITHRA H. KARUNATHILAKE

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: Engineering Management

The University of Texas Rio Grande Valley

December 2023

GAZE TRACKING EMBEDDED COLLABORATIVE ROBOTS FOR AUTOMATED
METROLOGY AND REVERSE ENGINEERING

A Thesis
By
SACHITHRA H. KARUNATHILAKE

COMMITTEE MEMBERS

Dr. Douglas Timmer
Chair of Committee

Dr. Zhaohui Geng
Co-chair of Committee

Dr. Jianzhi Li
Committee Member

Dr. Yangyang Long
Committee Member

December 2023

Copyright 2023 Sachithra H. Karunathilake

All Rights Reserved

ABSTRACT

Karunathilake, Sachithra H., Gaze Tracking Embedded Collaborative Robots for Automated Metrology and Reverse Engineering. Master of Science in Engineering (MSE), December, 2023, 54 pp., 22 tables, 11 figures, 29 references, 31 titles.

Conventional geometric metrology, or three-dimensional (3D) scanning, and reverse engineering heavily rely on the experience of the operators. With an increasing need for automation, robot arms have been adopted for this task. However, due to the large variety of parts and designs, automated path planning could provide a scanning solution that may overlook the critical area, which could potentially deteriorate the scan results. This study explores the integration of collaborative robotics (cobots) with eye tracking technology to improve the autonomous 3D scanning process. The primary objective of this study is to enhance the accuracy and efficiency of cobots in 3D scanning, particularly in the capture of functionally critical areas, and to provide a detailed description of regions with complex geometric features. The study develops a framework where the scanning path of the robot-carried scanner is partially guided by the eye-tracking data, i.e., the calibrated gaze tracking, to improve the automated 3D scanning process. This framework provides an innovative integration of human gaze movement with automatic robot path planning, providing a new way of human-autonomy teaming. Case studies are presented to present and validate the proposed framework to automatically improve the 3D point cloud collection process, specifically in the areas that usually require human manual intervention to capture details.

DEDICATION

This work is dedicated to my family and all my loved ones.

ACKNOWLEDGMENTS

I extend my profound gratitude to Zhaohui Geng, Ph.D., for his unwavering support and for providing me with significant advice, mentorship, and substantial knowledge during my research. I also extend my sincere gratitude to Dr. Douglas Timmer, Ph.D., Jianzhi Li, Ph.D., and Yangyang Long Ph.D. for their steadfast input and encouragement.

My parents' unending support and confidence in my abilities have been a constant source of inspiration for me. My steadfast commitment to overcoming obstacles and pursuing achievement has been motivated by their unshakable faith in my potential. The substantial influence of their assistance is honored in this thesis.

This work represents the collaborative inspiration, prodding, and constant support of my professors, and lab mates, particularly Shahriar Forhad. I hope it will serve as a way for me to express my gratitude for all who have done to help me advance both academically and personally. To all mentioned above and those who may have inadvertently been left out, thank you for your contributions and for being a part of this significant milestone in my academic journey.

TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGMENTS.....	v
TABLE OF CONTENTS.....	vi
LIST OF TABLES.....	viii
LIST OF FIGURES.....	x
CHAPTER I INTRODUCTION.....	1
Background to the Research Problem.....	2
Research Questions.....	3
CHAPTER II LITERATURE REVIEW.....	4
Human and Robot Collaboration.....	6
Collaborative Robotics.....	7
Eye Tracking.....	8
Three-Dimensional Scanning Automation.....	9
CHAPTER III METHODOLOGY.....	11
Scanner Coordinates.....	12
Eye Tracker Coordinates.....	18

Projecting 3D Points to 3D Space	20
Marker Registration.....	21
Scanning Using the Cobot.....	24
Eye Tracking Integration to Autonomous Scanning Process.....	25
Critical Area Identification.....	26
Performance Metric for 3D Scanning	26
Experimental Setup	29
CHAPTER IV RESULTS AND DISCUSSION.....	32
Initial Scan.....	40
Second Scan with 180° Object Rotations.....	43
Critical Area Analysis	46
Delta Time Calculation	46
Density and Curvature Metric	48
CHAPTER V CONCLUSION AND FUTURE WORK	49
REFERENCES	51
BIOGRAPHICAL SKETCH	54

LIST OF TABLES

	Page
Table 1: Marker coordinates of dataset 1 obtained from the scanner	15
Table 2: Marker coordinates of dataset 2 obtained from the scanner.	16
Table 3: Marker coordinates of dataset 3 obtained from the scanner.	16
Table 4: Eye tracking LAOI coordinate system	19
Table 5: Coordinates of the corner markers from dataset 1	33
Table 6: Lengths between each pair of coordinates based on dataset 1.....	33
Table 7: Coordinates of the corner markers from dataset 2.....	34
Table 8: Lengths between each pair of coordinates based on dataset 2.....	34
Table 9: Coordinates of the corner markers from dataset 3.....	35
Table 10: Lengths between each pair of coordinates based on dataset 3.....	35
Table 11: Order of the corner markers identified by each dataset.....	36
Table 12: Second set of corner markers identified in dataset 1	37
Table 13: Conversion of 3D coordinates to 2D coordinates.....	38
Table 14: Euclidean distance between the points of 3D coordinates and 2D coordinates	38
Table 15: Gaze movement- initial scan	41

Table 16: Comparison of fully autonomous scanning eye tracking integrated autonomous scanning results in scan 1	42
Table 17: Gaze movement after 180-degree object rotation.....	43
Table 18: Comparison of fully autonomous scanning eye tracking integrated autonomous scanning results after object rotation.....	45
Table 19: Gaze tracking- Critical area	46
Table 20: Eye tracking implementation time.....	47
Table 21: Robotic arm movement time	47
Table 22: Density and curvature metric.....	48

LIST OF FIGURES

	Page
Figure 1: A summary of the metrics development process.....	11
Figure 2: Formula 1 model car used as the scanning object.	12
Figure 3: Shinning 3D Eiscan HX handheld scanner	13
Figure 4: Scanning table with the markers	14
Figure 5: Boundaries of the LAOIs	20
Figure 6: KUKA robotic arm fully autonomous scanning path.....	24
Figure 7: Gaze movement as seen by the scene camera of the eye tracker.	25
Figure 8: Critical area of the model Formula 1 car.....	26
Figure 9: Visualization of 3D to 2D coordinate conversion for LAOI 1.....	39
Figure 10: Visualization of 3D to 2D coordinate conversion for LAOI 2.....	39
Figure 11: Eye tracker coordinates and scanner coordinates registration.....	40

CHAPTER I

INTRODUCTION

Three-dimensional (3D) scanning is a technology that could transform an object or an environment from a physical world to a digital world (Geng & Bidanda, 2017). With the major advancements and needs in Industrial 4.0 and 5.0, including digital twins, virtual reality/mixed reality. 3D scanning technologies have attracted significant interest in different applications. For example, in additive manufacturing, reverse engineering adopts 3D scanning techniques for the digitization step to translate the physical object into a digital point cloud for design reconstruction or metrology purposes(Geng & Bidanda, 2021); in autonomous driving, or robotics in general, 3D scanning could provide a survey of the surrounding environment with depth information, which provides more detailed information for path planning or decision making (Li et al., 2021). However, manual operations are typically required for conventional 3D scanning, especially those with requirements for high accuracy and precision with a relatively fast speed, e.g., in a manufacturing setting. In this case, an experienced operator carries an arm or handheld scanner, equipped with laser scanning capability or structured light scanners, to digitize the target objects. The quality of the collected point cloud is heavily influenced by the geometric complexity of the objects and the experience of the operator. Although multiple industrial vendors propose automated scanning solutions, where robots are adopted to carry the scanning instrument, automated path planning could provide a uniform scanning quality for different designs, which can overlook areas with complex geometric features. These features are generally

related to the requirements in the functional specifications. A scan of these areas with low-quality point clouds or low point density could impact subsequent decision-making or process planning in a manufacturing system. On the other hand, a more detailed scan with all areas of the targets could increase the burden for computing and point cloud processing, which, in turn, could impact the efficiency of the scanning project.

In this study, we propose an innovative human-autonomy teaming framework that integrates gaze tracking into the online programming of a robotic arm carrying a 3D scanner to perform the scanning task. The robotic arm, equipped with automated path planning for scanning, is partially guided by the gaze movement of the operators, which introduces flexibility into the path to adapt to the target and the operator's expert knowledge regarding the specific object. In this way, we can increase the flexibility and efficiency of the 3D scanning task while preserving the safety and effectiveness of the operators.

Section 2 reviews the literature and major advances in recent trends in Industry 5.0 and human-robot collaboration, with a particular focus on embedded collaborative robotics with eye tracking. Section 3 presents our framework that integrates gaze-tracking for scanning path planning. The results and corresponding discussions of the proposed framework are in Section 4. Concluding remarks and directions for future research motivated by our framework are presented in Section 5.

Background to the Research Problem

While robots are highly precise, their ability to autonomously identify and focus on areas of an object that have intricate details or complex geometry is not always perfect. This limitation

can lead to less accurate 3D point cloud generation, especially for objects with complex designs or very fine features. The issue lies in the autonomous scanning system, which might not always differentiate between areas needing detailed attention and those that do not. As a result, some detailed parts might not be scanned with the required resolution, impacting the overall quality of the 3D model.

Furthermore, in fully autonomous 3D scanning, ensuring that every part of the object is scanned with equal precision is challenging. Some areas, particularly those that are hard to reach, may not get scanned properly. This can lead to missing points in some parts of the 3D model. Consequently, human supervision in autonomous scanning often becomes necessary. Operators might need to intervene, adjusting the robot's position or the scanning parameters to ensure that areas with high-detail and complex features are captured accurately. Therefore, there is a need for human intervention in the autonomous scanning process through collaborative robots to optimize and increase the efficiency of the scanning process.

Research Questions

The research aims to develop a method to incorporate eye tracking into 3D scanning using a robotic arm to optimize autonomous scanning. The research questions of this study are presented as follows:

- I. Can gaze tracking be integrated into fully autonomous scanning?
- II. Can the integration of gaze tracking improve the scanning results?

CHAPTER II

LITERATURE REVIEW

The Fourth Industrial Revolution, also known as Industrial 4.0, applies cutting-edge technology to connect equipment and share data to build smart factories. These factories increase output, improve efficiency, and create new business prospects. The words "Industrial 4.0" and "Industrial 5.0" indicate important stages in the ongoing growth of industry in the fields of manufacturing and technology (Fatima et al., 2022). It directed automation (Kolberg & Zühlke, 2015), data-driven decision-making, and the Internet of Things (IoT) into industrial processes, changing manufacturing and supply chains all over the world (Riley et al., 2021).

This evolution of Industry 5.0 builds on the groundwork set by Industry 4.0 with additional needs and enhancements in technology and human interaction to a new level (Leng et al., 2022). These advances require a broader approach to rethinking the role of technology in industry, with an emphasis on adaptation, sustainability, and resilience. As one of the fundamental driving forces in Industry 5.0, human-machine collaboration (Yang et al., 2022), or human-autonomy teaming, emphasizes not only automation but also the participation of operators in performing complex and flexible tasks efficiently and effectively.

Collaborative robotics, or cobot, is one of the most representative advancements in this transition, from Industry 4.0 to Industry 5.0. Cobots, introduced during the Industry 4.0 era,

were designed to work alongside human operators, providing a more secure and adaptable automation solution in many industrial applications, primarily material handling (Vicentini, 2021). These robots are designed to enhance precision and strength in work assignments and provide repetitive motions on the production line to improve the safety of the operators.

Collaborative robotics in Industry 4.0 was integrated by sophisticated sensor technologies, analytic algorithms, and intelligence to adapt when human operators are present in its working environment. These robots could function within close proximity to humans without risking their safety, resulting in higher efficiency and more dynamic manufacturing processes. Industry 5.0 elevates human-robot collaboration to a new level of integration and engagement. While Industry 4.0 emphasized process optimization and task automation, Industry 5.0 emphasized the holistic inclusion of human workers in advanced decision-making and problem-solving alongside robots.

In Industry 5.0, collaborative robots extend beyond mere task execution. These robots are equipped with advanced machine learning and artificial intelligence algorithms that allow them to learn from human operators with more intelligent decision-making processes and adapt to unexpected events in (near) real-time (Soori et al., 2023). Since then, they have evolved to be able to participate in operational tasks actively. Furthermore, emotional interactions and social contact between humans and robots are also prioritized in Industry 5.0 (Chin, 2021). This indicates that robots assist humans in physical work and in recognizing and responding to human emotions and needs (Lu et al., 2022). This level of collaboration improves the entire work experience and results in pleasant and safer human-robot collaboration.

Human and Robot Collaboration

Human-robot collaboration, enabled by the integration of cutting-edge technology such as recording human biological information, computer vision, and eye tracking sensors, marks a significant step forward in industry 5.0. The robot's ability to record and comprehend human biometric data, such as the actions of human workers, is a crucial component of this relationship between robots and humans. Robots can see and comprehend their human counterparts' gestures, posture, and motion patterns thanks to cutting-edge computer vision capabilities (Tapus et al., 2019). Given that robots may adapt their activities in real-time to complement the motions of the human worker, this knowledge is essential for tasks that call for close collaboration. Furthermore, robots can now record and read human biometric information beyond simple physical movement. Robots can recognize subtle information like facial expressions and emotional states based on different algorithms such as Convolutional Neural Networks (CNN) and Visual Geometry Groups (VGG), Xception Networks, and Deep Face Networks, which offer deeper insights into the emotions and engagement of human operators (Chiurco et al., 2022). However, Lampi et al. (2023) mentioned that there are instances where technostress emerges among employees working with physical robots. Computer vision's thorough comprehension of biometric data improves productivity and enables a more sympathetic and flexible approach to human-robot collaboration. Computer vision enables robots to perform tasks precisely while simultaneously improving safety by avoiding obstacles. Furthermore, eye-tracking technology enables robots to detect human intents, increasing engagement and adaptability in dynamic working contexts.

Collaborative Robotics

Collaborative robots have been significantly growing and revolutionizing the reverse engineering and manufacturing industry. They are specifically designed to work with humans in performing tasks such as material handling, assembly, and welding. Combining the unique strengths of humans and robots, allows tasks to be performed effectively and efficiently. The main goal of collaborative robotics is to integrate robotic systems to complete tasks challenging for human operators, both from workplace safety or ergonomic perspectives and from the view of production quality (Pauliková et al., 2021).

Vicentini (2021) stated that robots do not determine the level of collaboration. Applications and the unique ways humans use robot systems for tasks define the level of collaboration. Different levels of collaboration are available depending on how the operator interferes and works alongside the robot. For instance, confined space collaborative robots refer to the use of robots or robotic systems designed to operate in restricted or confined spaces while working alongside human operators (Deshpande et al., 2018). Other types include open space (Follini et al., 2021) and human interactions (Hentout et al., 2019).

Cobots can be effectively used in difficult or dangerous environments for humans to access. In manufacturing, these robots can be used for assembly tasks in tight spaces where human workers may have difficulty reaching. In emergency response scenarios, robots can be deployed to explore and assess hazardous environments, such as collapsed structures or contaminated areas.

Cobots are versatile and can be easily integrated into different stages of the production process. Metrology is one such area that has attracted interest. The addition of cobots in

metrology also supports the trend towards smart manufacturing, as they can be connected to a network, enabling real-time data exchange and analysis. This connectivity allows instant feedback and adjustments in the manufacturing process, leading to reduced process time and higher-quality products. Moreover, the data collected by cobots can be used for process optimization.

Eye Tracking

Eye-tracking technology has developed as an effective tool for studying and analyzing human behavior and cognition. This technology allows for the comprehensive monitoring and analysis of eye movements and gaze points, revealing where people are looking, how long they concentrate on specific things, and their visual attention patterns. This information has great potential to learn the intentions and behavior of operators more efficiently and to improve collaborations between the human operator and automation systems (Ajoudani et al., 2018).

The Gaze Tracking Algorithm is one of the most important algorithms used in eye tracking for cobots. The cobot can match its actions to the worker's focus due to this algorithm, which enables it to estimate a worker's gaze location precisely (Palinko Oskar et al., 2016). By streamlining assembly procedures, they become more straightforward to understand and more effective.

Cobots can easily coordinate their movements with the visual cues of the operator because of this level of object awareness (Krishna Sharma et al., 2020). This technology can be used to make object scanning more effective and streamlined. Gesturing gains extra depth when paired with Object Recognition Algorithms. Cobots can now recognize not just where an

operator is looking but also what they are looking at due to object identification, which is frequently driven by deep learning algorithms (van Dyck et al., 2021). Cobots could recognize specific objects of interest inside their field of view by studying the real-time video feed from cameras or sensors and utilizing image recognition techniques (De et al., 2019).

Eye and head movements are important in human-robot interaction because they allow robots to recognize significant areas of interest and respond effectively to human signals. When humans move their gaze on certain items or regions, it is a potent indicator of their concentration and intent (Klin et al., 2002). Robots with advanced computer vision and eye-tracking technology can detect these minute movements in look and head movement, allowing them to figure out what the human operator considers important or unpleasant in their surroundings. This ability allows robots to adapt their actions, prioritize work, and even offer aid when needed, resulting in a more natural and efficient human-machine collaboration. The ability of robots to understand and respond to human gaze and head movements improves safety, productivity, and overall user experience in industrial settings, healthcare, and everyday life (Grewal et al., 2020).

Three-Dimensional Scanning Automation

Cobots can move quickly and precisely scan a wide range of objects with various designs and large scales. Their agility and precise scanning paths enable them to adapt the context of the object and ensure comprehensive scanning coverage. Cobots are perfect for scanning complex components because of their agility and precision in scanning paths, which allow them to move around complex shapes swiftly. Their ability to work continuously without fatigue reduces downtime and increases overall productivity. In addition to these operational benefits, cobots offer a safer alternative to manual scanning, reducing the risk of workplace injuries and allowing

human workers to focus on more strategic, less repetitive tasks. This combination of speed, precision, and versatility makes cobots indispensable in modern scanning and quality control processes.

However, there are challenges associated with using cobots in 3D scanning. While cobots offer high precision, the level of detail they can capture in scanning can be limited due to their inability to determine the areas of the object with high details and geometric complexity. The current practice of robot-driven 3D scanning ignores the regional complexity of the target while placing equal weight on different design features. A rough scan may overlook areas with complex geometric features or details that have a strong connection with the target's functionality; a detailed scan of all features could solve this issue while significantly increasing the burden in point cloud processing. It seems trivial for an experienced operator to plan the scanning path carefully, but it can be hard to program the robot to adapt to different objects. Therefore, there is a need for a framework to incorporate human knowledge or guidance into automated path planning.

CHAPTER III

METHODOLOGY

In this study, the primary objective was to integrate gaze tracking into the handheld 3D point cloud scanner to increase the accuracy of the robotic arm movement arm in identifying critical areas in an object to scan. This section of the paper describes the methods used to register the marker coordinates acquired from the handheld scanner with the coordinates from the eye tracking system.

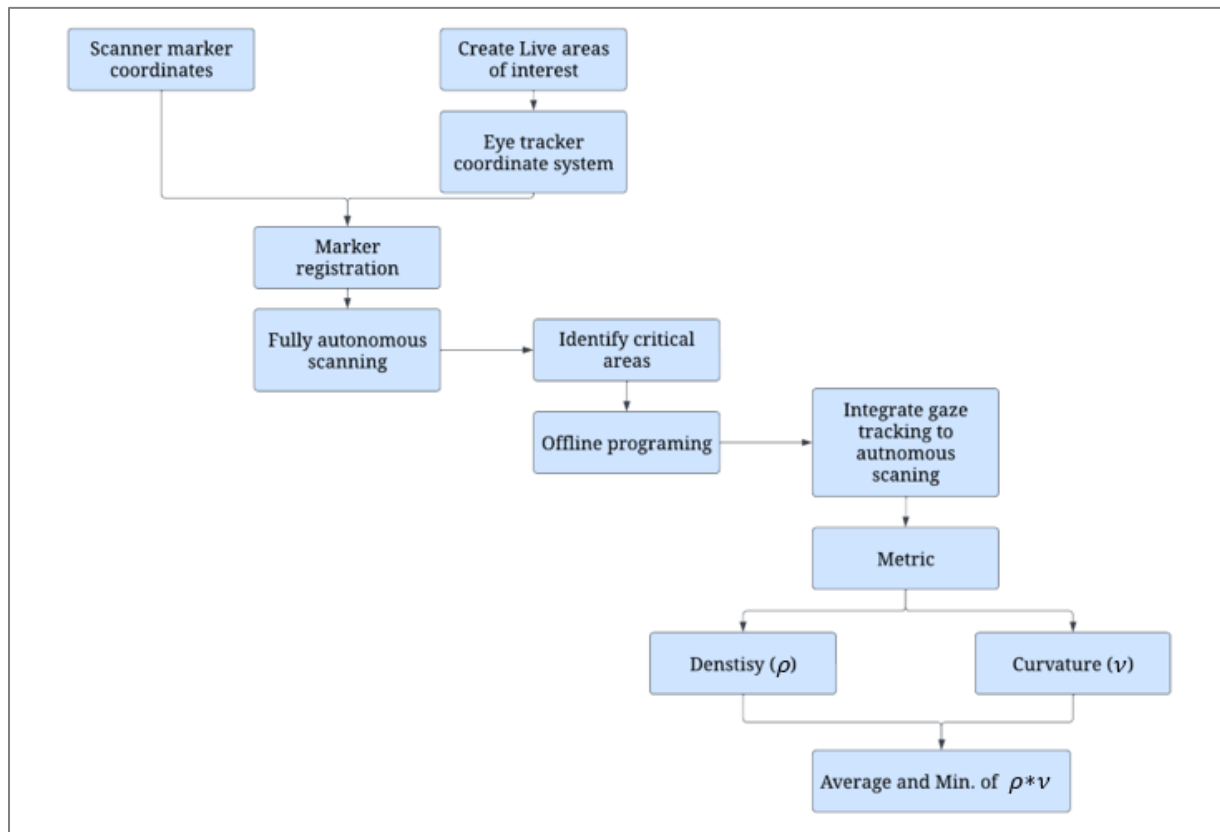


Figure 1: A summary of the metrics development process

The integration of the registered coordinates for planning the robotic arm's path for scanning is detailed, as well as the application of gaze tracking to identify critical areas that require rescanning. Figure 1 summarizes the metrics development process carried out in this study. The Formula 1 model car shown in Figure 2 was used as the scanning object.

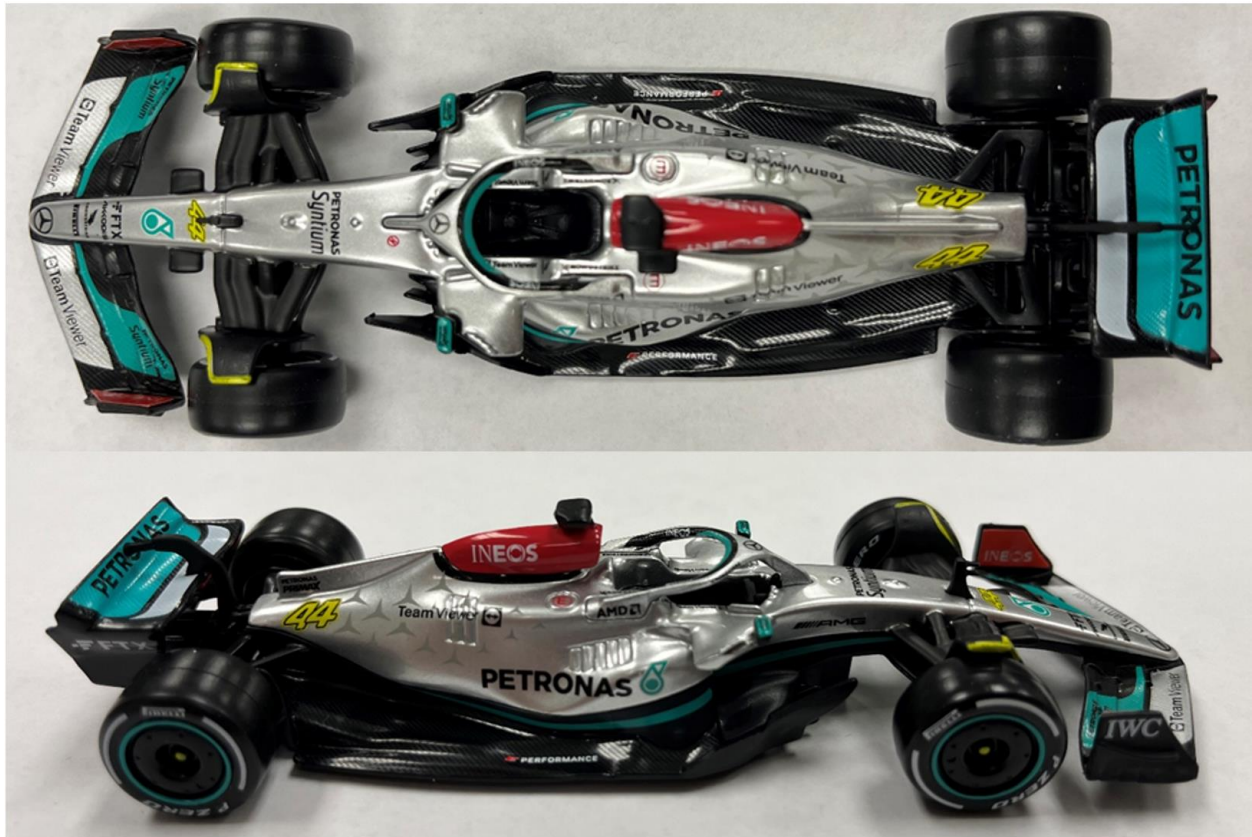


Figure 2: Formula 1 model car used as the scanning object.

Scanner Coordinates

The object was scanned using a Shining 3D EinScan HX scanner, as shown in Figure 3. These scanners usually use markers, small circular stickers pasted on the surface of the scanning object or platform, as reference points. These markers serve well in many aspects of the scanning process. They are mainly used for tracking and aligning physical objects with scanning software.

Handheld scanners such as the EinScanner HX continually scan objects from various perspectives as the operator moves them around. The EX-scanner HX software uses the markers as a frame of reference to track its location in relation to the object. This facilitates the software's ability to align precise 3D models by helping it realize how one scan fits in relation to the others. Moreover, these markers help maintain the accuracy of the scanning process and minimize errors. Certain objects are challenging for the scanner's software to distinguish from one region to another as they have repetitive textures or lack unique features. Markers generate distinct points that are simple for the program to recognize and track.

Unlike stationery or tripod scanners, with a base, handheld scanners typically do not have a fixed “origin point”. An origin point is not a physical spot on the scanner as it is a marker-based coordinate system established throughout the scanning process. The portable scanner's software frequently needs an initialization step before beginning a scan.



Figure 3: Shinning 3D Eiscan HX handheld scanner

A critical step of this study is to identify the coordinates of the markers in the EX-Scanner HX software and to define the origin point of the scan.

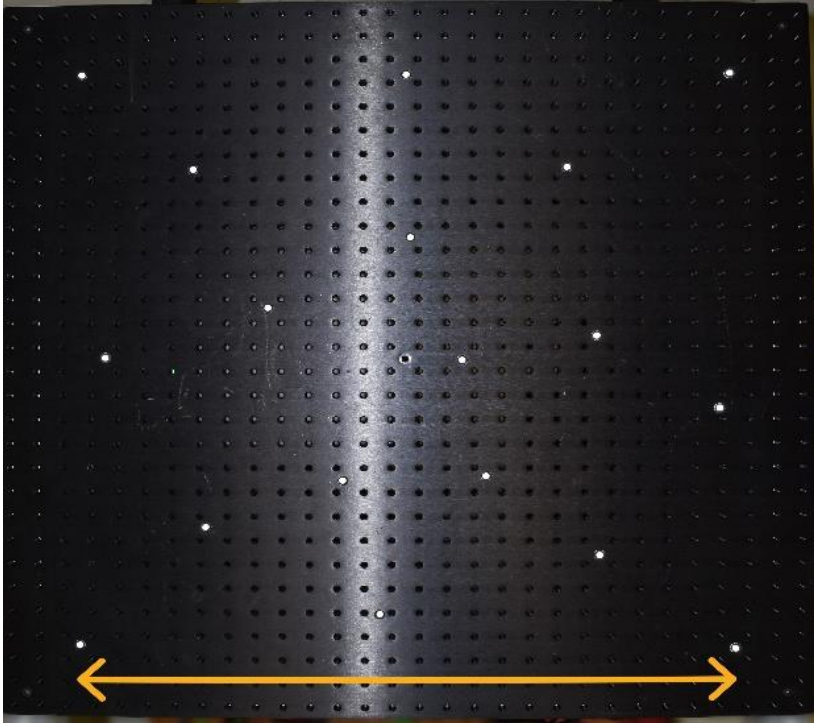


Figure 4: Scanning table with the markers

Initially, eighteen markers were placed on the scanning table, as shown in Figure 4. Four markers were placed on the corners of the table at an equal distance, forming a perfect square shape, while the other fourteen markers were placed randomly within the boundaries set by the four corner markers.

Marker coordinates were scanned and exported from the EX Scanner HX software in .P3 file format, As shown in Table 1. The file presented the XYZ (3D) coordinates sequentially line by line. However, the data does not define whether these coordinates follow a particular order of the scan or if they correspond to the origin point of the scan, which poses difficulties in

identifying the origin point considered by the software and aligning it with an origin point of the physical markers.

The initial step involves identifying which line in the dataset of the software represents which physical markers. Therefore, the same eighteen markers were scanned three times, resulting in three sets of coordinate data to compare and determine whether each dataset from the scanner constantly corresponded to the same physical markers with every scan. The three datasets are given in Table 1, Table 2, and Table 3, respectively.

Table 1: Marker coordinates of dataset 1 obtained from the scanner.

X	Y	Z
-8.748	88.0854	444.229
141.331	44.7749	501.154
-99.9607	-29.55	394.108
66.5732	187.866	485.654
184.314	207.819	536.017
-160.749	239.649	397.871
60.3446	308.54	495.921
-54.6399	309.828	448.872
75.3862	441.102	516.1
-68.6248	485.521	461.768
42.3092	608.279	520.249
-229.345	554.506	402.753
238.737	548.835	594.345
313.305	669.07	637.304
207.284	394.718	565.181
352.564	372.877	622.266
307.02	184.201	583.675
437.552	82.9147	626.005

Table 2: Marker coordinates of dataset 2 obtained from the scanner.

X	Y	Z
-94.3461	48.9136	319.862
70.4723	25.7427	375.433
-34.0357	-99.3699	325.641
-147.034	225.793	319.412
-217.695	-64.8929	264.554
16.8799	-238.464	329.02
-143.775	-163.482	280.36
-230.729	-238.783	241.939
-339.727	-15.2919	226.708
-139.012	-327.373	265.025
-355.006	-232.586	198.849
-76.9208	-445.442	274.628
-289.945	-385.409	205.996
-503.099	-223.126	147.422
153.593	-175.817	383.59
292.827	-169.385	432.896
-68.348	-615.128	259.837
95.0214	-374.68	342.244

Table 3: Marker coordinates of dataset 3 obtained from the scanner.

X	Y	Z
-33.4292	92.8872	519.32
-157.011	71.8072	472.691
121.56	42.4	569.505
-147.227	-99.4466	458.073
-305.697	124.036	424.887
-262.393	10.8022	428.345
-302.833	-111.457	400.801
1-415.078	84.3113	381.371
-332.534	256.187	429.339
-493.883	-20.6148	341.822
-482.804	247.378	374.496
-656.589	256.07	312.896
-415.692	404.321	415.262
-193.16	258.775	479.563
-211.117	417.403	490.004
-352.808	-256.498	367.154
-48.739	289.58	534.61
-177.739	558.932	516.773

The coordinates of the corner markers of the squares were determined by the following algorithms.

Maximum Distance

This algorithm considers the diagonal of the square as the longest line segment that can be drawn between any two points among these 18 markers. Therefore, the opposing corners of the square were determined by finding the two points that are the farthest away.

In the algorithm, the Euclidian distance is utilized to measure the distance between any two points. The Euclidian distance, $d(p_i, p_j)$ between point $p_i = (x_i, y_i, z_i)$ and point $p_j = (x_j, y_j, z_j)$ is as follows,

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

Distances between each pair of points are computed. The pair of points yielding the greatest (farthest) distance among all points are identified and designated as the corner points, which can be utilized for further alignment. In this way, two pairs of corner points are identified sequentially, while any selected points are removed from the candidate list.

Convex Hull

The convex hull algorithm is another way to identify corner points for alignment purposes. The convex hull is a concept in computational geometry that refers to the smallest convex set that contains a set of points. This convex hull is presented as a polygon in a two-dimensional space, whereas in a three-dimensional space, it is presented as a polyhedron. The vertices of this convex polygon or polyhedron are a subset of the original points and are called the “hull points”. The convex hull is unique because it creates a border that defines the external

boundary of a set of points, excluding all internal points. To identify corner points from a set of 3D points, the convex hull algorithm could be used.

The correspondence of the order of these points is explored by considering their angular position relative to a central point, i.e., the centroid. These points were consistently sorted by measuring the angle from the centroid to each point on the hull. Although it was not explicitly constructed as a convex hull, this approach can detect extreme or border points in the dataset.

The coordinates identified from the algorithms mentioned above as corner markers are established by calculating the distance between each point using the Euclidean distance function. This measurement can confirm whether the distances match up in a square pattern.

These two methods were repeated after eliminating the identified coordinates of the corner points in the dataset. The intention is to find the coordinates of the next possible corners out of the remaining markers.

Eye Tracker Coordinates

After identifying the coordinate points of the markers in the software with the physical markers, the coordinates of the eye tracker have to be determined prior to marker registration. Therefore, a coordinate system must be assigned to the regions the eye tracker captures. The eye tracker used in the study provides the option to create a Live Area of Interest (LAOI), which can be designated to image objects or features of the environment captured through the eye tracker's scene camera.

The operator was instructed to look at the scanning table, and the four corner markers were designated to be the boundaries of the LAOI, as shown in the figure. In this study, two LAOIs

were created to accompany the identified eight corners, with four markers assigned to each of the two LAOI1 and LAOI2. The distance between the four corner markers was physically measured to be 597mm. After defining the two LAOI regions the operator can walk around the scanning table while the LOAIs remain defined and unchanged regardless of the movement of the operator. The gaze movement of the operator was tracked within the two defined LOAIs.

As we have clearly defined LAOIs in the eye-tracking coordinate system, the closest point marker to the operator from the operator’s point of view can be recognized as the origin point of the eye-tracking coordinate system. The coordinate system is shown in Table 4. A visual representation of the two LAOI boundaries is shown in Figure 5.

Table 4: Eye tracking LAOI coordinate system.

LAOI	Point	X	Y
LAOI 1	0	0	0
	1	597	0
	2	597	597
	3	0	597
LAOI 2	0.5	257	55
	1.5	567	375
	2.5	312	592
	3.5	49	363

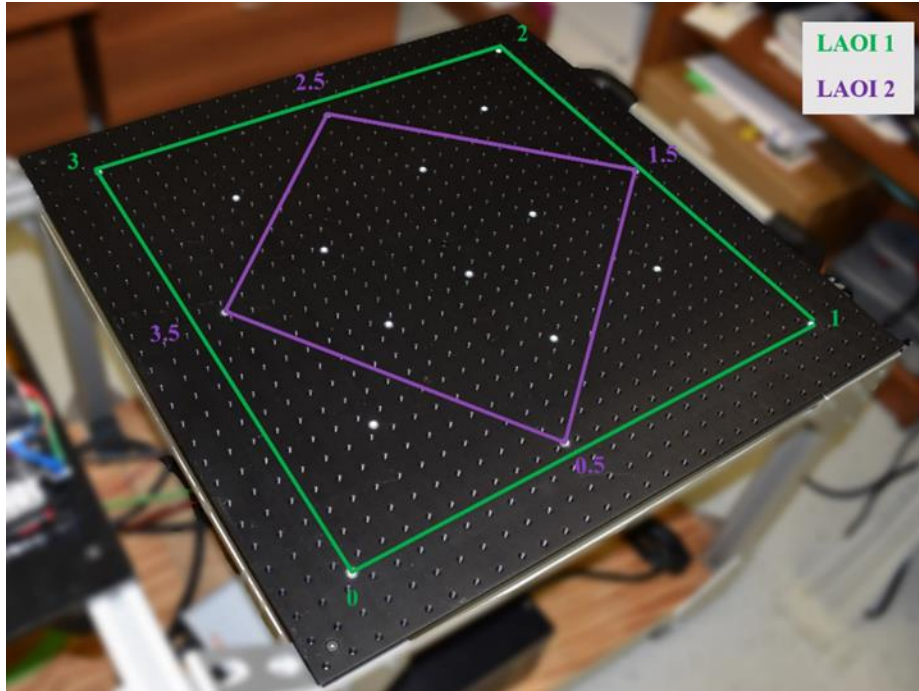


Figure 5: Boundaries of the LAOIs

Projecting 3D Points to 3D Space

Since the coordinates of the markers provided by the 3D scanner and the vision system on the eye tracker have different dimensions, a multidimensional scaling (MDS) algorithm (Carroll & Arabie, 1998) is used to convert the coordinates for registration. The goal of MDS is to preserve pairwise distances between points while locating a lower-dimensional representation of the points. In addition, the stress value is calculated by measuring the difference between the original and decreased distances.

In this study, two variants of MDS are used: one preserves Euclidean distances, while the other preserves Manhattan distances. The most popular and basic distance metric is the Euclidean distance, the straight-line distance between two locations in Euclidean space. In

contrast, the Manhattan distance, which sums the absolute differences of their Cartesian coordinates, is used in grid-like path calculations. The Manhattan distance formula is given in the following formula:

$$d(p_i, p_j) = |x_i - x_j| + |y_i - y_j| + |z_i - z_j|$$

Both metrics influence the location of points in the MDS-transformed space, which affects how the items' relationships are represented. The code uses these distances to conduct MDS on a given dataset, producing a two-dimensional representation of the points and comparing item configurations using various distance metrics. The stated "stress" number represents how well the MDS representation preserves the original distances, with lower values indicating a better match.

Marker Registration

The registration of the marker coordinates obtained from both the scanner and the eye tracker was a crucial part of this study. These coordinates should be precisely aligned so that the eye tracking information can be utilized to guide movement or path planning for the robotic 3D scanning system. Operators could quickly identify critical regions or areas with poor point cloud quality, while the eye tracking device could integrate this information into the movement instructions for the robotic arm.

As one of the most popular registration algorithms, the iterative closest point (ICP) algorithm is implemented to remove the unnecessary rotational and translational factors in the coordinates of the markers from two sources. The ICP algorithm uses best-fit transform to reduce the distance between the two-point sets, which the algorithm repeatedly determines the best-fit via

homogeneous transformations, singular value decomposition (SVD), and closest neighbor search. Convergence is accelerated by using centroids for an initial posture estimate.

Let $\mathbf{A} = \{\mathbf{a}_i\}_{i=1}^n$ and $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^n$ be the sets of markers collected by 3D scanning and eye tracking device. The main concept behind the ICP algorithm is to find the nearest neighbor of one set of points and then compute the rotation matrix and translation vector for these pairs of points. Then the error is minimized and optimized using SVD. The ICP algorithm can be explained using the following steps (Prochzkov & Martiek, 2018)

For the computation of SVD, the centroids need to be calculated, which are presented as follows,

$$C_A = \frac{1}{n} \sum_{i=1}^n A_i$$

$$C_B = \frac{1}{n} \sum_{i=1}^n B_i$$

Then, all the points in sets A and B are moved to the position, where the centroids are located at the origin, while the translational factors are removed,

$$A' = \{\mathbf{a}'_i\}_{i=1}^n = \{\mathbf{a}_i - C_A\}_{i=1}^n$$

$$B' = \{\mathbf{b}'_i\}_{i=1}^n = \{\mathbf{b}_i - C_B\}_{i=1}^n$$

Next, a rotation matrix R can be found using SVD of the matrix $H = A' B'^T$ to remove the rotational factor and align the set of points A' onto B' .

As for the pair of points that lack correspondence, the nearest neighbor algorithm could find the closest point in set B for each point in set A .

The ICP algorithm iteratively updates the transformation (using the best-fit transform) to minimize the distance between the corresponding points in sets A and B . A pre-set tolerance can control the stopping condition of the algorithm.

After aligning the markers in the scanning software and the scanning table with the LAOI coordinate system of the eye tracker, the robot could automatically identify the crucial areas from the gaze hint from the operator.

Scanning Using the Cobot

In the initial scanning process, the path of the robotic arm was designed to move around the scanning table once as shown in Figure 6. This autonomous scanning process used a path that covered and captured 360° and the top of the object. After completing the initial one-round fully autonomous scan, we observed the scan 3D point cloud to identify areas with high details or geometric complexities that have not been appropriately scanned or that have missing points. The object was scanned again using integrated gaze tracking. An important part of the scanning process is the control of the robot's motion. The eight points registered by the ICP algorithm were used as reference points to move the robotic arm.

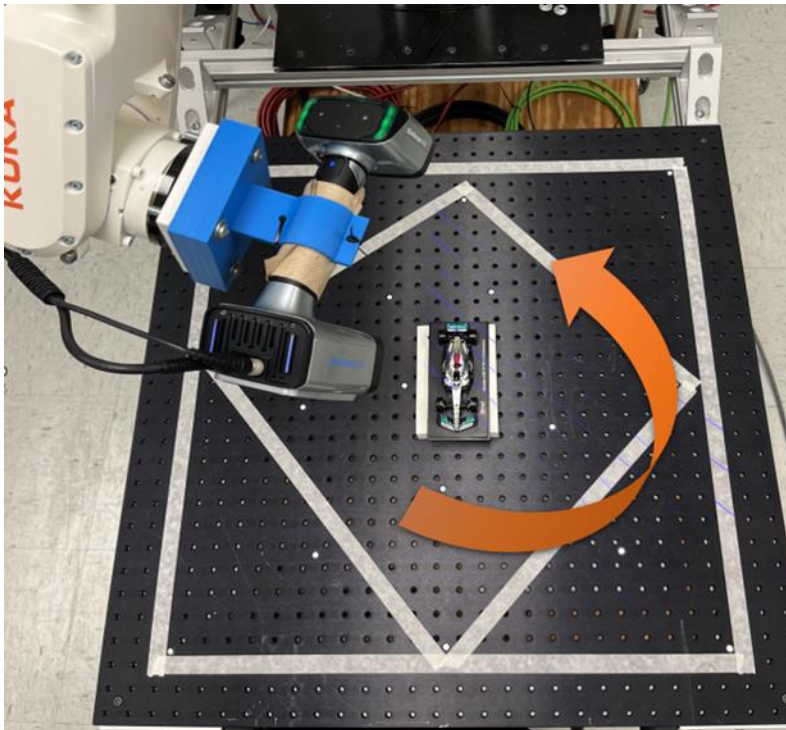


Figure 6: KUKA robotic arm fully autonomous scanning path.

Eye Tracking Integration to Autonomous Scanning Process.

Prior to setting up the scanning process. The robotic arm was moved to eight locations and obtained the coordinates of the arm position for each marker. In the eye tracking embedded scanning process, the operator identified the rescanning areas based on the autonomous scan and determined the areas that require rescanning. The operator moved the gaze to the marker corresponding to the areas requiring rescanning and obtained the coordinates from the eye tracker coordinate system pertaining to the marker with the gaze, as shown in Figure 7. Subsequently, the coordinates were input into the robotic arm program. The robotic arm was then moved to the assigned marker with the gaze and scanned again to capture the point cloud. The location of the scanning table in relation to the robot was crucial to its mobility. The robot's end effector can move within a range of proximity, but several limitations exist on how far it can go from its base. Moreover, because of the torque required at each joint, the movement to new positions could not maintain consistent velocities.

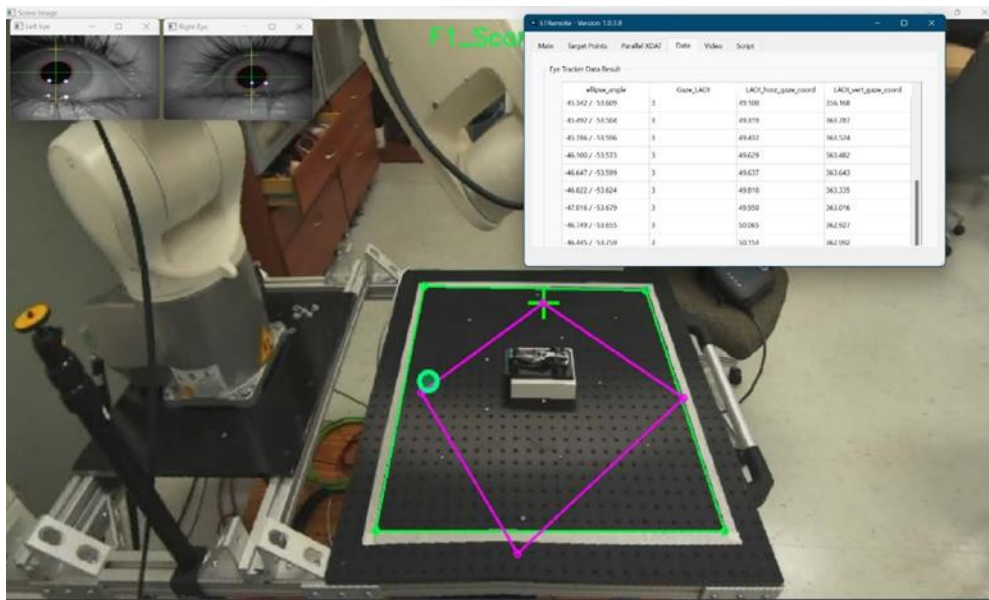


Figure 7: Gaze movement as seen by the scene camera of the eye tracker.

Critical Area Identification

To further analyze whether integrating human input into the autonomous scanning process through gaze tracking could optimize the scanning of complex geometries, we selected a critical area with a high level of detail and curvature. As shown in Figure 8, the cockpit area of the f1 model car was selected as the critical area as it had more details compared to the overall design of the car.



Figure 8: Critical area of the model Formula 1 car.

Performance Metric for 3D Scanning

To compare the performance for the two ways of scanning using cobot, an innovative performance metric considering the local point density with respect to the local curvature is proposed in this study.

In 3D point cloud scanning, the local point density, or the number of points in a unit volume, can be utilized to describe how much information is included in a local area. Typically, the higher the local point density is, the more details can be captured in a 3D scanning project. A high local point density is essential to capture the fine features of an object, especially the areas of complex features, while a relatively low local point density is acceptable for simple features, such as prismatic features.

The local point density, ρ_i , for any point i in the point cloud can be calculated as follows,

$$\rho = \frac{N}{\frac{4}{3}\pi r^3}$$

where N_i is the number of points located in a local neighborhood of point i with a pre-determined radius r .

Conversely, curvature represents a surface's deviation from being flat at a given location in computational geometry (Geng et al., 2022). It is a geometric property that characterizes how quickly the direction of the tangent vector of a point on the surface changes as one moves over the surface. At one extreme, the curvature for any point on a flat surface has an infinitesimal curvature; while, at the other extreme, a sharp edge has curvature that goes to infinity. Therefore, curvature could be utilized as a descriptor of the complexity of a surface. Regions with sharp edges or curves, suggestive of high complexity, have been determined to be correlated with substantial curvature. On the other hand, areas with flat surfaces correspond to minimal curvature values, indicating less complexity. The idea was that areas with more curvature have additional complex designs by nature. The local curvature v at a point is estimated by fitting a plane to its neighboring points using the least-squares method described below.

Consider the

$$G_i = U_i S_i V_i^T$$

where G_i is a matrix whose rows are the neighboring points of point i and, then, SVD is used to decompose the matrix G . The normal vector to the best-fit plane is obtained from the last row of V_i^T in SVD. Finally, the curvature is approximated by the standard deviation of the distance between the neighboring points and the fitted plane. The deviation between the j^{th} point in the neighborhood and the fitted plane is given by

$$d_i = \frac{(G_{ij} - p_i) \cdot n_i}{\|n_i\|}$$

where G_{ij} represents the j^{th} point in the neighborhood of point i , n_i is the normal vector of point i . Then the curvature can be calculated by,

$$v_i = 2 \times S_{\{d_j\}}$$

where 2 is a scaling factor used to adjust the curvature estimation, and $S_{\{d_j\}}$ is the standard deviation of the set of deviations $\{d_j\}$.

The metric proposed in this study is the product of the local curvature and the point density ($\rho_i \times v_i$). This metric integrates the local curvature, v_i , which represents the complexity or complexity of the surface at that point, and the local point density, ρ_i , which indicates the number of points utilized to represent a specific area. The metric representing regions of the scan with high geometric complexity and detail levels may be obtained by multiplying the local point density by local curvature. A high value of $\rho_i \times v_i$ would refer to a scanned area that has a relatively adequate number of points and is both geometrically complexity and well-described.

On the other hand, low values of $\rho_i \times v_i$ can represent regions that are under-described, which may need additional, more careful scans.

When we compare the different scanning strategies, both the average and the smallest value of $\rho_i \times v_i$ are reported. In this study, we are assessing the details and geometric complexity of the point cloud generated from fully autonomous scanning and eye tracking integrated autonomous scanning. Ideally, we look for high $\rho_i \times v_i$ value for the autonomous scan integrated with eye tracking compared to the fully autonomous scanning process.

Experimental Setup

The operator's gaze movement was captured using the Argus Science ET Vision glasses. The ETVision device uses a technique known as "Pupil to CR" Tracking to monitor eye movements. This method uses the dark pupil's location and the corneal reflections (CRs) to determine an individual's gaze direction. The cornea's anterior surface is what generates these reflections in the eye. With the help of two near-infrared LEDs mounted on the headset, the system illuminates the eyes. The eye camera located in the device captures the infrared light, which is barely visible to the operator's eyes. The corneal reflections appear as bright spots when viewed through the camera. The pupil's center moves in relation to these bright dots as the eye moves. The eye tracking system determines the direction of gaze with respect to a coordinate system centered on the scene camera by measuring the change in location between the pupil and the CRs. The position of the pupil or CRs alone allows the system to track the operator's gaze even while the headset is stationary.

The technical specifications of the eye tracker used in the experiment were as follows: Gaze Measurement Frequency of 180 Hz, enabling precise data collection. The eye tracker's

estimated accuracy is around 0.5 degrees, allowing precise eye movement measurement. The ET vision system also consists of SDK samples for a python programming language to import live eye tracking data through the ET remote software.

The object scanning was done using a Shinning 3D Einscan HX handheld scanner attached to the KUKA KR10 R1100 robotic arm. It provides two scanning modes: rapid scanning and laser scanning. This study only utilized the laser scanning mode. However, the eye tracking procedure used in the study can be applied to optimize rapid scanning as well.

The Laser Scan mode can increase precision up to 0.04mm and volumetric accuracy to 0.04+0.06mm/m; Rapid Scan offers accuracy of up to 0.05mm and a volumetric accuracy of 0.05+0.1mm/m. In contrast to the Laser Scan's 480,000 points per second at 55 frames per second (FPS), the Rapid Scan mode achieved an impressive scan speed of 1,200,000 points per second at 20 FPS. The camera frame rate in both modes was 55 frames per second. In terms of alignment, Laser Scan only used Markers Alignment while Rapid Scan used Texture, Markers, Texture Alignment, and Hybrid alignment. Both had a working distance of 470mm. The depth of field ranges for Rapid Scan and Laser Scan were 200 mm to 700 mm and 350 mm to 610 mm, respectively. For both Rapid Scan and Laser Scan, the maximum field of view (FOV) was 420 x 440 mm and 380 x 400 mm, respectively. Finally, for Rapid Scan, the point distance varied from 0.25mm to 3mm, and for Laser Scan, tightened to 0.05mm to 3mm.

The EinScanner HX was attached to the KUKA robotic arm. The 11.1 kg maximum payload capacity of the KUKA robot made it easy to carry and move the 345g weight EinScanner with ease. With a posture repeatability of ± 0.02 mm in accordance with ISO 9283 standards, it demonstrated its accuracy in movement and placement. The six axes of motion ensured that the robot could move in many directions. In particular, the motion range of Axis 1

(A1) was ± 170 degrees; that of Axis 2 (A2) was ± 190 to 45 degrees; that of Axis 3 (A3) was ± 120 to 156 degrees; that of Axis 4 (A4) was ± 185 degrees; that of Axis 5 (A5) was ± 120 degrees; and that of Axis 6 (A6) was ± 350 degrees. This range of motion made it possible to be quite flexible.

CHAPTER IV

RESULTS AND DISCUSSION

This Chapter includes the results obtained for the proposed methodologies mentioned in the previous Chapter. The computations were carried out using Python programming language.

The Maximum Distance and convex hull algorithms were used to identify the coordinates of the four corner markers from the scanner for each of the three datasets. In the Maximum Distance method, corner markers were identified according to their maximum Distance from one another. Convex Hull algorithm typically uses the outermost points that form the smallest convex shape encompassing all the points. The code was formulated to give the coordinates of the markers counterclockwise for both methods.

Table 5 depicts the coordinates of the corner markers, as identified by each algorithm for the first dataset. It was observed that, out of the eighteen markers, both algorithms recognized the same markers as the outermost markers. However, the order in which these coordinates were arranged differed between the two algorithms. The order identified by the Maximum Distance algorithm is 14, 12, 3, and 18. Conversely, the convex hull algorithm determined the sequence as 3, 18, 14, and 12.

Table 5: Coordinates of the corner markers from dataset 1.

Order	X	Y	Z	Max_distance	Convex hull
1	-8.748	88.0854	444.229		
2	141.331	44.7749	501.154		
3	-99.9607	-29.55	394.108	3	1
4	66.5732	187.866	485.654		
5	184.314	207.819	536.017		
6	-160.749	239.649	397.871		
7	60.3446	308.54	495.921		
8	-54.6399	309.828	448.872		
9	75.3862	441.102	516.1		
10	-68.6248	485.521	461.768		
11	42.3092	608.279	520.249		
12	-229.345	554.506	402.753	2	4
13	238.737	548.835	594.345		
14	313.305	669.07	637.304	1	3
15	207.284	394.718	565.181		
16	352.564	372.877	622.266		
17	307.02	184.201	583.675		
18	437.552	82.9147	626.005	4	2

These coordinates are confirmed by calculating the lengths of the four sides of the square using the Euclidean distance, as shown in Table 6. Upon measuring the physical distance between the corner points, the length was approximately 597 mm. Interestingly, the calculated lengths were approximately equal to the physical length of the corner markers.

Table 6: Lengths between each pair of coordinates based on dataset 1.

Length	Max_dis	Convex hull
(1,2)	602.1695	596.1077
(2,3)	598.2779	599.2854
(3,4)	596.1077	602.1695
(4,1)	599.2854	598.2779

Table 7: Coordinates of the corner markers from dataset 2.

Order	X	Y	Z	Max_distance	Convex hull
1	-94.3461	48.9136	319.862		
2	70.4723	25.7427	375.433		
3	-34.0357	-99.3699	325.641		
4	-147.034	225.793	319.412	2	4
5	-217.695	-64.8929	264.554		
6	16.8799	-238.464	329.02		
7	-143.775	-163.482	280.36		
8	-230.729	-238.783	241.939		
9	-339.727	-15.2919	226.708		
10	-139.012	-327.373	265.025		
11	-355.006	-232.586	198.849		
12	-76.9208	-445.442	274.628		
13	-289.945	-385.409	205.996		
14	-503.099	-223.126	147.422	3	1
15	153.593	-175.817	383.59		
16	292.827	-169.385	432.896	1	3
17	-68.348	-615.128	259.837	4	2
18	95.0214	-374.68	342.244		

Table 8: Lengths between each pair of coordinates based on dataset 2.

Length	Max_dis	Convex hull
(1,2)	602.098	596.0798
(2,3)	598.24	599.2359
(3,4)	596.0798	602.098
(4,1)	599.2359	598.24

Table 9: Coordinates of the corner markers from dataset 3.

Order	X	Y	Z	Max distance	Convex hull
1	-33.4292	92.8872	519.32		
2	-157.011	71.8072	472.691		
3	121.56	42.4	569.505	4	2
4	-147.227	-99.4466	458.073		
5	-305.697	124.036	424.887		
6	-262.393	10.8022	428.345		
7	-302.833	-111.457	400.801		
8	-415.078	84.3113	381.371		
9	-332.534	256.187	429.339		
10	-493.883	-20.6148	341.822		
11	-482.804	247.378	374.496		
12	-656.589	256.07	312.896	2	4
13	-415.692	404.321	415.262		
14	-193.16	258.775	479.563		
15	-211.117	417.403	490.004		
16	-352.808	-256.498	367.154	3	1
17	-48.739	289.58	534.61		
18	-177.739	558.932	516.773	1	3

Table 10: Lengths between each pair of coordinates based on dataset 3.

Length	Max_dis	Convex hull
(1,2)	602.1533	596.0796
(2,3)	598.2916	599.3045
(3,4)	596.0796	602.1533
(4,1)	599.3045	598.2916

The Max distance and convex hull algorithms were applied to the second and third datasets to determine if the same markers would be identified as corner markers upon multiple scans. As depicted in Table 7 and Table 9, the results showed that both algorithms identified the same markers as corner markers but in a different sequence in their respective datasets. The accuracy of these coordinates was confirmed by calculating the Euclidean distance, as shown in Table 8 and Table 10. However, the order of the corner markers identified in the second and

third datasets was different compared to the first dataset. In all three datasets, the length of the four sides calculated based on the coordinates were approximately equal regardless of the order.

The results show that the order of the markers changes every time a scan is done. The scanner starts the scanning process by identifying the object's orientation and creating a coordinate system based on its starting location in relation to the object, which can be defined as an arbitrary “origin” point. Therefore, the origin point can change every time a scan is done. As the scanning process proceeds, the scan uses the markers to identify its location in relation to the origin point. Therefore, the corner markers cannot be identified based on the order of the XYZ marker coordinates presented in the .P3 file as shown in Table 11.

Table 11: Order of the corner markers identified by each dataset.

Dataset	The row that was identified as the corner markers
1	3, 12, 14, 18
2	4, 14, 16, 17
3	3, 12, 16, 18

The subsequent outermost points were determined by excluding the four identified in the three datasets. The algorithms' accuracy had already been confirmed, and only the Max distance method was used for this purpose because, despite their different orders, both algorithms produced the same coordinates. Only dataset 1 was used in the computation of the remaining coordinates. Based on the findings, the next four markers were found to be rows 6, 11, 16, and 2.

Table 12: Second set of corner markers identified in dataset 1.

Order	X	Y	Z	Max distance
1	-8.748	88.0854	444.229	
2	141.331	44.7749	501.154	4
4	66.5732	187.866	485.654	
5	184.314	207.819	536.017	
6	-160.749	239.649	397.871	1
7	60.3446	308.54	495.921	
8	-54.6399	309.828	448.872	
9	75.3862	441.102	516.1	
10	-68.6248	485.521	461.768	
11	42.3092	608.279	520.249	2
13	238.737	548.835	594.345	
15	207.284	394.718	565.181	
16	352.564	372.877	622.266	3
17	307.02	184.201	583.675	

The original XYZ (3D) coordinates for the eight markers were converted to XY (2D) coordinate system to be aligned with the 2D coordinates of the eye tracker coordinate system. The new original coordinate values and transformed coordinate values are shown in Table 12.

Table 13: Conversion of 3D coordinates to 2D coordinates.

	3D Coordinates			Transformed 2D Coordinates	
	X	Y	Z	X	Y
LAOI 1	-99.9607	29.55	394.108	-392.296788	156.8594507
	-229.345	554.506	402.753	157.08576	394.0920478
	313.305	669.07	637.304	393.9074819	-159.5039278
	437.552	82.9147	626.005	-158.696454	-391.4475707
LAOI 2	-160.749	239.649	397.871	241.3625669	158.33677611
	42.3092	608.279	520.249	131.4855523	-265.85363746
	352.564	372.877	622.266	-251.02048405	-140.08612556
	141.331	44.7749	501.154	-121.82763515	247.60298691

The newly transformed 2D coordinates were then confirmed by calculating the Euclidean distance between the points, as shown in Table 14. The calculated distances of each pair of points in the 2D system are approximately equal to the corresponding distances in the 3D system. The stress value was found out to be 0.12914. Figure 9 and Figure 10 depict the 3D to 2D point coordinate conversion for LAOI 1 and 2.

Table 14: Euclidean distance between the points of 3D coordinates and 2D coordinates.

	Points	Lengths using 3D Coordinates	Lengths using 2D coordinates
LAOI 1	(0,1)	596.1077	598.415
	(1,2)	599.2854	602.1238
	(2,3)	602.1695	599.3071
	(3,0)	598.2779	595.9947
LAOI 2	(0.5,1.5)	438.2888	438.19
	(1.5,2.5)	402.5911	402.6516
	(2.5,3.5)	408.5811	408.6486
	(3.5,0.5)	374.0262	373.9994

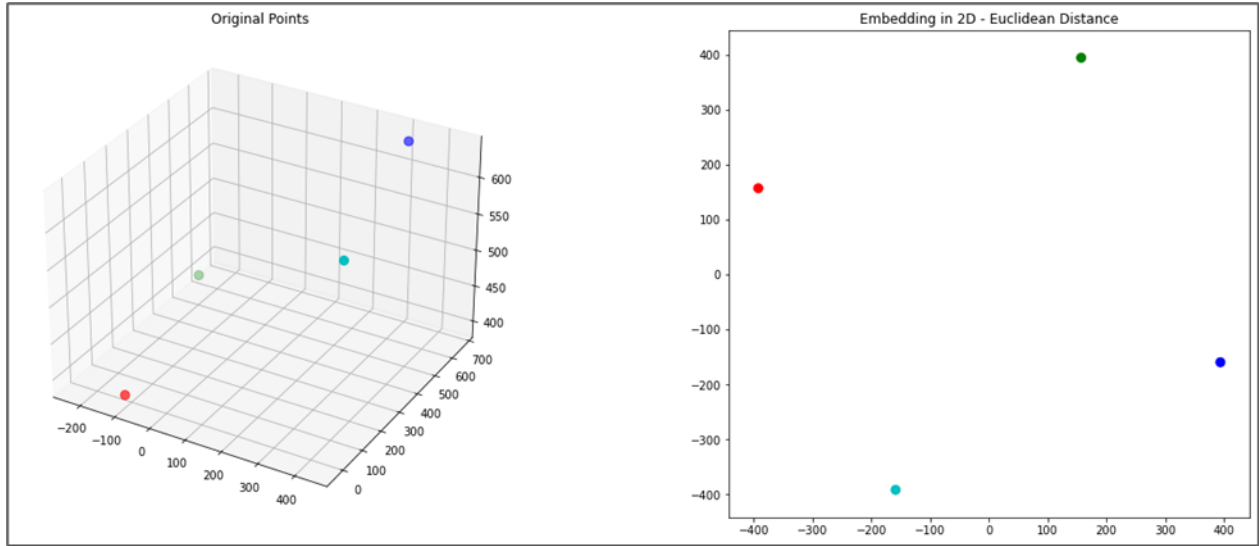


Figure 9: Visualization of 3D to 2D coordinate conversion for LAOI 1.

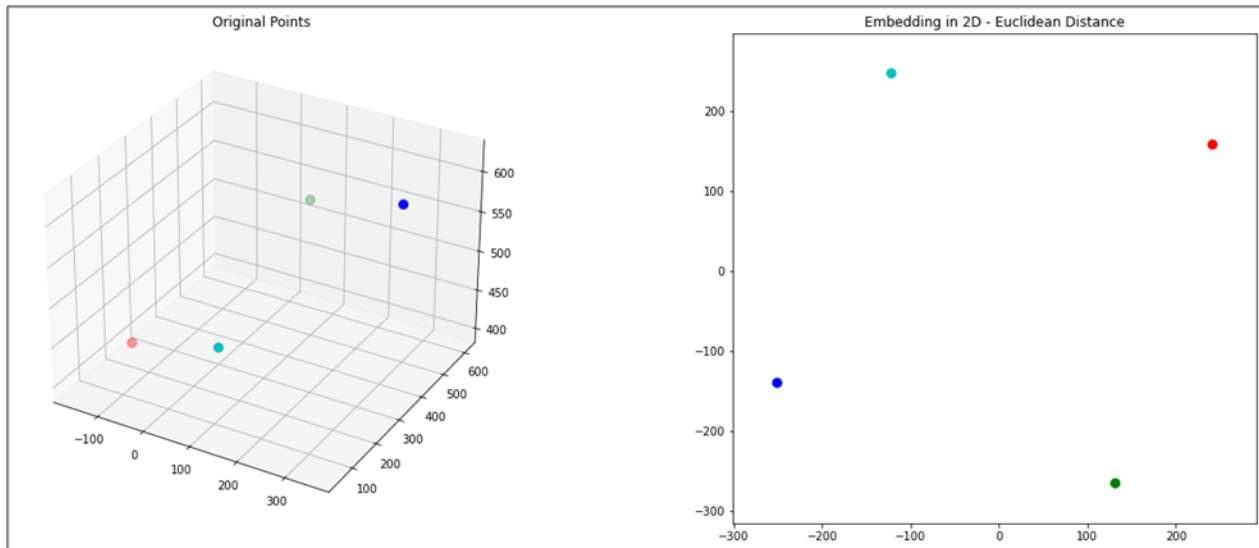


Figure 10: Visualization of 3D to 2D coordinate conversion for LAOI 2.

The ICP point registration algorithm was applied to align the 2D coordinates of the scanner with the coordinates of the eye tracking coordinate system. Figure 11 demonstrates that 4 corner markers 0,1,2,3 were aligned somewhat accurately with only slight deviations.

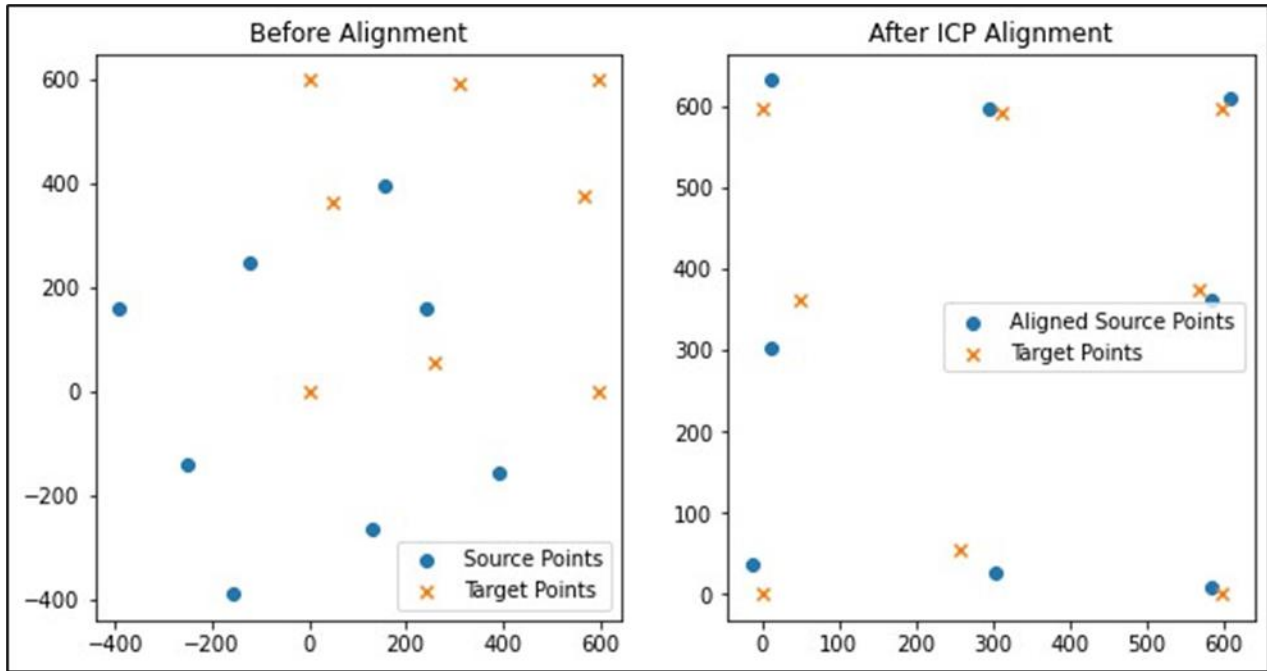


Figure 11: Eye tracker coordinates and scanner coordinates registration.

Initial Scan

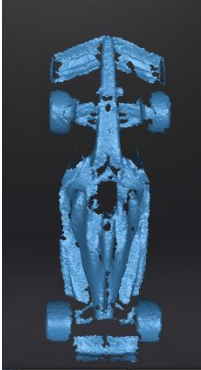
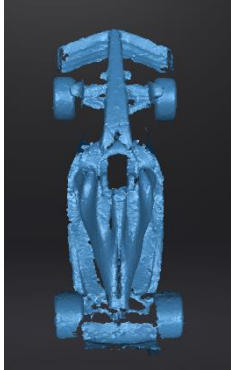
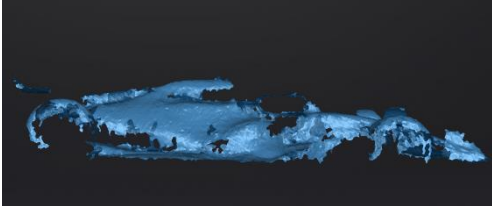
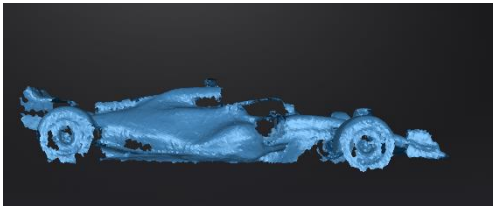
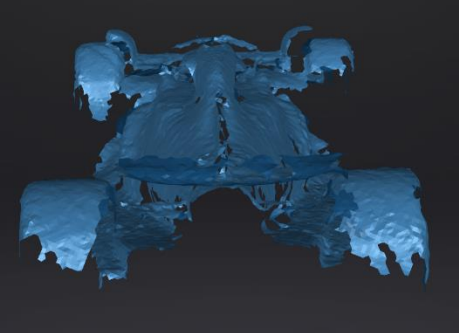
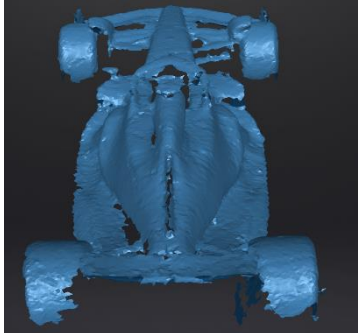
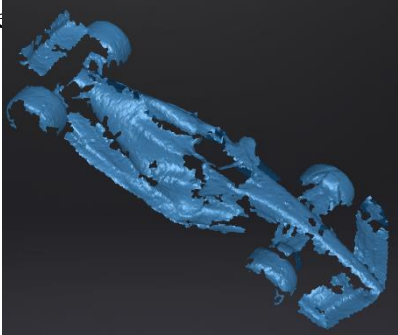
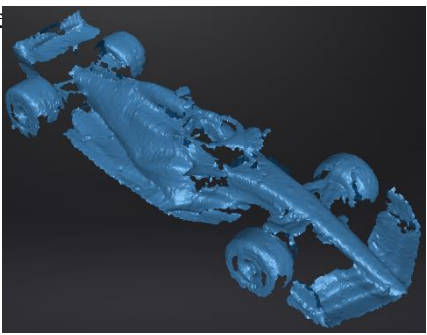
After the fully autonomous scan, the operator observed the areas with high details that required eye tracking integration to optimize the scanning results. Based on the observation, the robotic arm was moved to the markers as shown in Table 15.

Table 15: Gaze movement- initial scan.

Movement	Eye tracker LAOI coordinate
1	0.5
2	3.5
3	2
4	3.5

The images in Table 16 showcase mesh data from scans of the F1 model car. The first column displays the results from the fully autonomous scan, while the second column displays the images of autonomous scanning optimized by eye tracking. In comparison, gaps observed between the points in the first image indicated that the point cloud has a lower point density. As a result, the surface may appear rougher and have less detail. The scanned data of the fully autonomous scan exhibits visible irregularities, as several regions show sparse dots, potentially indicating that the scan did not fully capture all the characteristics of the F1 model car. Closer visual observations indicated possible noise in both scans.

Table 16: Comparison of fully autonomous scanning eye tracking integrated autonomous scanning results in scan 1.

Autonomous scanning	Autonomous scanning optimized by eye tracking
Top 	Top 
Side 	Side 
Rear 	Rear 
Front side 	Front-side 

While conducting the scanning process, we observed that the robotic arm moved faster when performing the scan nearest to its base than it did toward other markers. We believe this prevented the scanner from capturing points below the model car's rear wing, which had intricate details and significant curvature, as shown in the images of the rear of the car in both scans. Therefore, we rotated the model car 180° and performed a second scan to address this issue.

Second Scan with 180° Object Rotations

Following the rotation of the model, fully autonomous scanning was performed. Similar to the previous scan, the results indicated imperfections in the scan and an inability to capture areas with high details, indicating the requirement to integrate gaze tracking to identify the areas with high details to obtain a better scan.

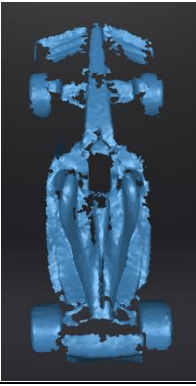
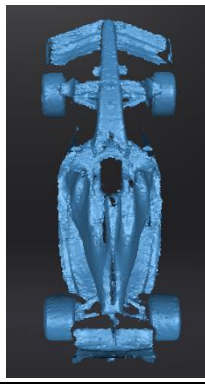
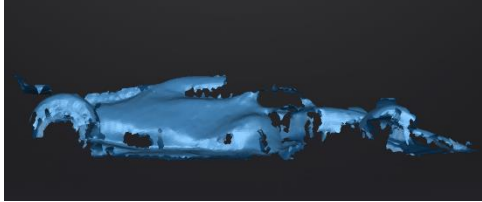
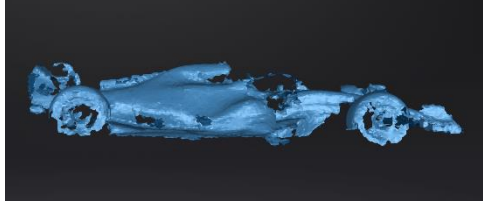
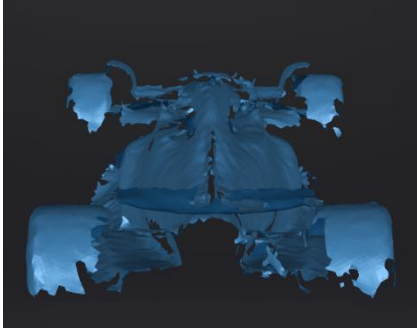
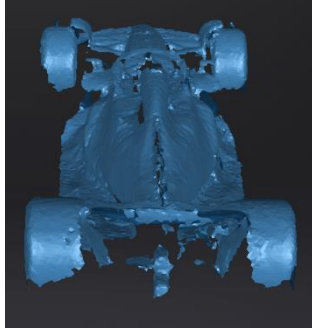

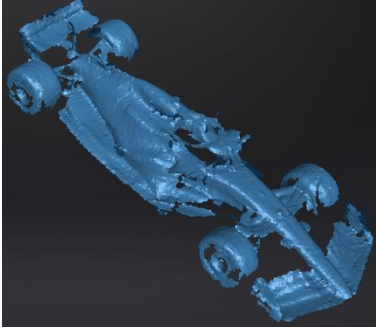
The operator observed the areas with high details that required eye tracking integration to optimize the scanning results. Based on the observation, the robotic arm was moved to the five markers, as shown in Table 17.

Table 17: Gaze movement after 180-degree object rotation.

Movement	Eye tracker LAOI coordinate
1	2.5
2	1.5
3	1
4	1.5
5	3

Upon closer examination, it was evident that the integration of gaze tracking enhanced the scanning procedure, as the table illustrates, as this method gathered information more successfully than the fully autonomous scan. The images shown in Table 18 additionally demonstrate the successful recording of points beneath the rear wing of the model car using the gaze-tracking-enhanced scan. Turning the model car 180° had no effect on the point cloud generation in the front area of the model car, which has less detailed than the rear.

Table 18: Comparison of fully autonomous scanning eye tracking integrated autonomous scanning results after object rotation.

Autonomous scanning	Autonomous scanning optimized by eye tracking
<p>a</p> 	<p>b</p> 
	
	
	

Critical Area Analysis

The same procedure was followed to obtain point cloud data for the critical area of the F1 model car. The gaze tracking was integrated to the following marker to optimize the details captured in the process as shown in Table 19.

Table 19: Gaze tracking- Critical area.

Movement	Eye tracker LAOI coordinate
1	3.5
2	2.5

Delta Time Calculation

Table 20 shows the approximate time spent obtaining the eye tracking coordinates for each process. The approximate time taken to input the coordinates in the KUKA robotic arm software and move the robotic arm is given in Table 21. The robotic arm showed an average movement time of 58 seconds from the origin to the coordinate obtained from gaze tracking and back to the origin point. The entire process of offline programming to integrate eye tracking into the autonomous scanning system takes approximately 5 minutes and 6 seconds.

Table 20: Eye tracking implementation time.

Method	Duration
Autonomous scanning optimized by eye tracking.	94 seconds
Autonomous scanning optimized by eye tracking- 180 object rotation	140.478 seconds
Autonomous scanning of the critical area optimized by eye tracking	46.022 seconds

Table 21: Robotic arm movement time.

LAOI Marker	Origin to Marker	Origin to marker & back to point
0	30 sec	51 seconds
0.5	31 sec	59 seconds
1	25 sec	86 seconds
1.5	28 seconds	70 seconds
2	21 seconds	76 seconds
2.5	23 seconds	47 seconds
3	21 seconds	47 seconds
3.5	15 seconds	38 seconds
Center	20 seconds	48 seconds
Average	23 seconds	58 seconds

Density and Curvature Metric

Table 22 depicts the average $\rho*v$ and minimum $\rho*v$ values from each fully autonomous scans and eye tracking integrated autonomous scans carried out in the study. All three scans carried out by the fully autonomous scanning process show a modest average $\rho*v$ value, which indicates a balanced mixture of less detailed and highly detailed regions throughout the scan. However, the three eye tracking integrated autonomous scans have a higher average $\rho*v$ compared to the fully autonomous scans, suggesting that, overall, they contain more complex and detailed regions than the first scan. The higher average implies better capturing intricate details or inherently more complex surfaces.

A minimum value of 0 for $\rho*v$ in both scanning methods indicates the presence of surface areas with low point density or curvature.

Table 22: Density and curvature metric.

Method	Scan	Average $\rho*v$	Min. $\rho*v$
Fully autonomous	Scan one	0.376412	0
	Scan two with a 180° object rotations	0.354264	0
	Critical area	0.340837	0
Gaze tracking integrated	Scan one	0.46072	0
	Scan two with a 180° object rotations	0.44095	0
	Critical area	0.433029	0

CHAPTER V

CONCLUSION AND FUTURE WORK

This study proposes a method to integrate eye tracking to improve the autonomous 3D point cloud scanning process using collaborative robots. The experiment findings demonstrate a significant improvement in the autonomous scanning process in capturing high details and complex curvature with the integration of the human gaze to guide the robotic arm to critical areas with high details and geometric complexity.

The metrics proposed in the paper provide valuable quantitative evaluation and comparison of point cloud scans' complexity and detail, which will be helpful in applications where accuracy and detail are essential, such as reverse engineering.

A new direction of research can be developed based on the proposed metric. Moreover, the coordinate data obtained from eye-tracking can be extracted and streamed live to control the movement of the robotic arm in real-time. The marker registration can also be extended to other markers to increase the variations of the robotic arm.

Aligning 3D markers in 3D space can also be given higher priority. The technology evaluates the surroundings on a 2D plane while utilizing eye-tracking glasses. However, greater focus can be placed on aligning 3D coordinates produced from the scanner with the 3D coordinate system of a 3D vision system, such as the HoloLens 2, which includes gaze tracking.

The methodology suggested in this paper could be used in autonomous driving, where combining gaze tracking with 3D scanning can significantly improve the vehicle's perception system by focusing on regions of interest detected by the human gaze. Incorporating eye tracking integrated 3D scanning into an autonomous driving system may result in more efficient and context-aware scanning of the driving environment. For instance, if the driver of an autonomous car frequently glances at a specific object or region, the system may prioritize scanning of that area. This method assures that critical objects on the road, such as road signs, pedestrians, and unforeseen barriers, are identified quickly and precisely. Autonomous vehicles may construct detailed and prioritized 3D maps of their surroundings by altering the scanning focus based on gaze direction and gathering depth and spatial information. The LAOI feature of the ET vision eye tracking glasses is meant to account for object relative position, making it especially suitable for autonomous driving applications. The system's eye-tracking capabilities are maintained within a consistent reference frame as the vehicle navigates through varied surroundings, ensuring that head movement or vehicle motion does not interfere with the data acquired. As a result, future research could look into applying the proposed methods in the paper to autonomous driving systems.

REFERENCES

- Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., & Khatib, O. (2018). Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5), 957–975. <https://doi.org/10.1007/s10514-017-9677-2>
- Carroll, J. D., & Arabie, P. (1998). Multidimensional Scaling. *IBM SPSS Statistics 26 Step by Step*, 247–257. <https://doi.org/10.4324/9780429056765-20>
- Chin, S. T. S. (2021). Influence of emotional intelligence on the workforce for industry 5.0. *IBIMA Business Review*, 2021. <https://doi.org/10.5171/2021.882278>
- Chiurco, A., Frangella, J., Longo, F., Nicoletti, L., Padovano, A., Solina, V., Mirabelli, G., & Citraro, C. (2022). Real-time Detection of Worker’s Emotions for Advanced Human-Robot Interaction during Collaborative Tasks in Smart Factories. *Procedia Computer Science*, 200, 1875–1884. <https://doi.org/10.1016/j.procs.2022.01.388>
- De, O., Lázaro, M., Mohammed, W. M., Ferrer, R., Bejarano, R., & Lastra, J. L. M. (2019). *An Approach for adapting a Cobot Workstation to Human Operator within a Deep Learning Camera*.
- Deshpande, N., Ortiz, J., Sarakoglou, I., Semini, C., Tsagarakis, N., Brygo, A., Fernandez, J., Frigerio, M., Saccares, L., Toxiri, S., & Caldwell, D. G. (2018). Next-generation collaborative robotic systems for industrial safety and health. *WIT Transactions on the Built Environment*, 174, 187–200. <https://doi.org/10.2495/SAFE170181>
- Fatima, Z., Tanveer, M. H., Waseemullah, Zardari, S., Naz, L. F., Khadim, H., Ahmed, N., & Tahir, M. (2022). Production Plant and Warehouse Automation with IoT and Industry 5.0. In *Applied Sciences (Switzerland)* (Vol. 12, Issue 4). MDPI. <https://doi.org/10.3390/app12042053>
- Follini, C., Magnago, V., Freitag, K., Terzer, M., Marcher, C., Riedl, M., Giusti, A., & Matt, D. T. (2021). Bim-integrated collaborative robotics for application in building construction and maintenance. *Robotics*, 10(1), 1–19. <https://doi.org/10.3390/robotics10010002>
- Geng, Z., & Bidanda, B. (2017). Review of reverse engineering systems—current state of the art. In *Virtual and Physical Prototyping* (Vol. 12, Issue 2, pp. 161–172). Taylor and Francis Ltd. <https://doi.org/10.1080/17452759.2017.1302787>
- Geng, Z., & Bidanda, B. (2021). Geometric precision analysis for Additive Manufacturing processes: A comparative study. *Precision Engineering*, 69, 68–76. <https://doi.org/10.1016/j.precisioneng.2020.12.022>

- Geng, Z., Sabbaghi, A., & Bidanda, B. (2022). Automated variance modeling for three-dimensional point cloud data via Bayesian neural networks. *IISE Transactions*, 55(9), 912–925.
- Grewal, D., Kroschke, M., Mende, M., Roggeveen, A. L., & Scott, M. L. (2020). Frontline Cyborgs at Your Service: How Human Enhancement Technologies Affect Customer Experiences in Retail, Sales, and Service Settings. *Journal of Interactive Marketing*, 51, 9–25. <https://doi.org/10.1016/j.intmar.2020.03.001>
- Hentout, A., Aouache, M., Maoudj, A., & Akli, I. (2019). Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017. *Advanced Robotics*, 33(15–16), 764–799. <https://doi.org/10.1080/01691864.2019.1636714>
- Klin, A., Jones, W., Schultz, R., Volkmar, F., & Cohen, D. (n.d.). *Visual Fixation Patterns During Viewing of Naturalistic Social Situations as Predictors of Social Competence in Individuals With Autism*.
- Kolberg, D., & Zühlke, D. (2015). Lean Automation enabled by Industry 4.0 Technologies. *IFAC-PapersOnLine*, 28(3), 1870–1875. <https://doi.org/10.1016/j.ifacol.2015.06.359>
- Krishna Sharma, V., Saluja, K., Mollyn, V., & Biswas, P. (2020, February 6). Eye gaze controlled robotic arm for persons with severe speech and motor impairment. *Eye Tracking Research and Applications Symposium (ETRA)*. <https://doi.org/10.1145/3379155.3391324>
- Lampi, A., Salo, M., Venermo, K., & Pirkkalainen, H. (2023). *Emergence of technostress among employees working with physical robots*. <http://rightsstatements.org/page/InC/1.0/?language=en>
- Leng, J., Sha, W., Wang, B., Zheng, P., Zhuang, C., Liu, Q., Wuest, T., Mourtzis, D., & Wang, L. (2022). Industry 5.0: Prospect and retrospect. *Journal of Manufacturing Systems*, 65(August), 279–295. <https://doi.org/10.1016/j.jmsy.2022.09.017>
- Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M. A., Cao, D., & Li, J. (2021). Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8), 3412–3432. <https://doi.org/10.1109/TNNLS.2020.3015992>
- Lu, Y., Zheng, H., Chand, S., Xia, W., Liu, Z., Xu, X., Wang, L., Qin, Z., & Bao, J. (2022). Outlook on human-centric manufacturing towards Industry 5.0. *Journal of Manufacturing Systems*, 62, 612–627. <https://doi.org/10.1016/j.jmsy.2022.02.001>
- Palinko Oskar, Rea Francesco, Sandini Giulio, & Sciutti Alessandra. (2016). *A Robot Reading Human Gaze: Why Eye Tracking Is Better Than Head Tracking for Human-Robot Collaboration*.
- Pauliková, A., Babel'ová, Z. G., & Ubárová, M. (2021). Analysis of the impact of human–cobot collaborative manufacturing implementation on the occupational health and safety and the quality requirements. *International Journal of Environmental Research and Public Health*, 18(4), 1–15. <https://doi.org/10.3390/ijerph18041927>
- Prochzkov, J., & Martiek, D. (2018). Notes on iterative closest point algorithm. 17th Conference on Applied Mathematics, APLIMAT 2018 - Proceedings, 2018-February(April), 876–884.

- Riley, C., Vrbka, J., & Rowland, Z. (2021). Internet of things-enabled sustainability, big data-driven decision-making processes, and digitized mass production in industry 4.0-based manufacturing systems. *Journal of Self-Governance and Management Economics*, 9(1), 42–52. <https://doi.org/10.22381/jsme9120214>
- Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. In *Cognitive Robotics* (Vol. 3, pp. 54–70). KeAi Communications Co. <https://doi.org/10.1016/j.cogr.2023.04.001>
- Tapus, A., Bandera, A., Vazquez-Martin, R., & Calderita, L. V. (2019). Perceiving the person and their interactions with the others for social robotics – A review. *Pattern Recognition Letters*, 118, 3–13. <https://doi.org/10.1016/j.patrec.2018.03.006>
- Van Dyck, L. E., Kwitt, R., Denzler, S. J., & Gruber, W. R. (2021). Comparing Object Recognition in Humans and Deep Convolutional Neural Networks—An Eye Tracking Study. *Frontiers in Neuroscience*, 15. <https://doi.org/10.3389/fnins.2021.750639>
- Vicentini, F. (2021). Collaborative Robotics: A Survey. *Journal of Mechanical Design*, 143(4), 1–20. <https://doi.org/10.1115/1.4046238>
- Yang, J., Liu, T., Liu, Y., & Morgan, P. (2022). Review of Human Machine Interaction Towards Industry 5.0: Human-Centric Smart Manufacturing.

BIOGRAPHICAL SKETCH

Sachithra Karunathilake acquired his Bachelor of Science degree in International Management and Business from University of Plymouth, UK in 2017. He obtained his Master of Science in management from Minot State University, ND, USA. He joined the Manufacturing and Industrial Engineering department at The University of Texas Rio Grande Valley (UTRGV), TX, USA and graduated with a Master of Science in Engineering Management in December 2023. Through his exceptional academic performance and good standing with a GPA of 4.0/4.0, Sachithra earned some other institutional awards, such as the T.E Ann Mercer Scholarship, G.R and L. Ranganath Scholarship, and the Dean's Scholarship Award for multiple semesters. Also, with the support of the US Department of Defense (DoD) I-Dream4D Consortium, Sachithra worked as a Research Assistant under the supervision of Dr. Zhaohui Geng. Moreover, he also worked as student project coordinator at the Texas Manufacturing Assistance Center at UTRGV. Sachithra can be reached via email at sachithra.karunathilake@gmail.com.