10-2024

Gaze tracking embedded collaborative robots for automated metrology and reverse engineering

Sachithra H. Karunathilake
The University of Texas Rio Grande Valley

Md Shahriar Forhad
The University of Texas Rio Grande Valley

Zhaohui Geng

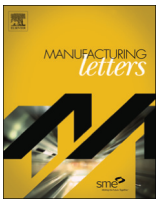
Recommended Citation

Karunathilake, Sachithra, Md Shahriar Forhad, and Zhaohui Geng. "Gaze tracking embedded collaborative robots for automated metrology and reverse engineering." Manufacturing Letters 41 (2024): 1488-1498. https://doi.org/10.1016/j.mfglet.2024.09.175

Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 41 (2024) 1488–1498



52nd SME North American Manufacturing Research Conference (NAMRC 52, 2024)

Gaze Tracking Embedded Collaborative Robots for Automated Metrology and Reverse Engineering

Sachithra Karunathilake[a], Md Shahriar Forhad[b], Zhaohui Geng[c,*]

[a]*Department of Manufacturing and Industrial Engineering, The University of Texas Rio Grande Valley, Edinburg, Texas, United States*
[b]*Center for Advanced Manufacturing Innovation and Cyber Systems (CAMICS), The University of Texas Rio Grande Valley Edinburg, Texas, United States*
[c]*Department of Industrial and Systems Engineering, Ohio University, Athens, Ohio, United States*

Abstract

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 article 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-driven scanner is partially guided by eye tracking data, that is, 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 areas that usually require human manual intervention to capture details.

Keywords: Collaborative Robotics; Human-Autonomy Teaming; Eye Tracking; 3D Scanning; Reverse Engineering

1. Introduction

Three-dimensional (3D) scanning is a technology that could transform an object or an environment from a physical world to a digital world [1]. With the major advancements and needs in Industrial 4.0 and 5.0, including digital twins, virtual reality/mixed reality, etc., 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 [2]; 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 [3]. However, manual operations are typically required for conventional 3D scanning, especially those with requirements in high accuracy and precision with a relatively fast speed, e.g., in a manufacturing setting. In this case, an experienced operator carries an arm scanner or a 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

* Corresponding author. Tel.: +1-740-593-1520
E-mail address: zgeng@ohio.edu (Zhaohui Geng).

areas with complex geometric features. These features are generally related to the requirements in the functional specifications of the corresponding products. 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.

Current fully autonomous 3D scanners follow a preprogrammed path for scanning, and this path doesn't adapt to the complexity of the object being scanned. Cobots do not have the ability to assess or judge the complexity of the object on their own. Therefore, human intervention is still necessary, even in fully autonomous scanning processes.

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 operator's expert knowledge regarding the specific object. In this way, we can both increase the flexibility and efficiency of the 3D scanning task while preserving the safety and effectiveness of the operators.

In Section 2, we review the literature and major advances in recent trends in Industry 5.0 and human-robot collaboration, with a special 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.

2. Literature Review

The evolution of Industry 5.0 is based on the foundation established by Industry 4.0 with additional needs and enhancements in the integration of technology and human interaction to a new level [4]. These advances require an urgent need to rethink the role of humans and technology in practice, with an additional emphasis on adaptation, sustainability, and resilience. As one of the key driving forces in Industry 5.0, human-machine collaboration [5], or human-autonomy teaming, emphasizes not only automation, but also the participation of operators in performing complex and flexible tasks in an efficient and effective way.

One of the most representative advancements in this transition from Industry 4.0 to Industry 5.0 is collaborative robotics, or cobot. Cobots, introduced during the Industry 4.0 era, were designed to be able to work alongside human operators, providing a more secure and adaptable automation solution in many industrial applications, especially material handling [6]. This type of robots is designed to enhance precision and strength in work assignments and provide exceptional repetitive motions on the production line to improve the safety of the operators. Cobots are typically integrated with sensor technologies and analytic algorithms for adaptivity and intelligence when human operators are present in their work envelope. 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 sophisticated decision-making and problem-solving alongside robots. In Industry 5.0, cobots extend beyond task execution. These cobots 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 [7]. Since then, they have evolved to be able to actively participate in operational tasks. Furthermore, emotional interactions and social contact between humans and robots are also prioritized in Industry 5.0 [8]. This indicates that robots have the ability to assist human operators both in physical work and in recognizing and responding to human emotions and needs [9]. This level of collaboration could further improve the work experience, resulting in a pleasant and safe environment for human-robot collaboration.

2.1. Collaborative Robotics

Collaborative robots have been significantly growing and revolutionizing the manufacturing industry. They are specifically designed to work together with humans in tasks such as material handling, assembly, welding, etc. By combining the unique strengths of humans and robots, this collaborative effort provides an efficient and effective way of performing jobs or tasks. The main goal of collaborative robotics is to integrate robotic systems to complete tasks that are challenging for human operators, both from workplace safety or ergonomic perspectives and from the view of production quality [10], while preserving the flexibility and ability of experienced workers.

Depending on how the operator works alongside the cobot and their interference, different levels of collaboration are available. For example, confined-space collaborative robotics refers to the use of robots or robotic systems that are designed to operate in restricted or confined spaces while working alongside human operators [11]. These robots are specifically designed to perform tasks in environments where human access is limited or hazardous. Other types include open space [12] and human interaction [13].

Cobots can be utilized effectively in environments that are difficult or dangerous for humans to access. In production lines, cobots are adopted for assembly tasks in tight spaces where human operators may have difficulty reaching. In emergency response scenarios, robots are deployed to explore and assess hazardous environments, such as collapsed structures or contaminated areas.

Cobots have also attracted interest in the metrology industry. These cobots can move quickly while precisely scanning

objects with various designs or large scales. Cobots are suitable for scanning complex components due to their agility and precision in scanning paths, which could significantly benefit throughput. However, there are challenges associated with using cobots in 3D scanning. 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, which have a strong connection with the target's functionality; a detailed scan of all features could potentially solve this issue while significantly increasing the burden in point cloud processing. It seems to be trivial for an experienced operator to carefully plan the scanning path, 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.

2.2. Human and Robot Collaboration

Human-robot collaboration, enabled by the integration of cutting-edge technologies, such as computer vision, advanced sensors, such as eye tracking sensors, and advances in human factors, 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 the gestures, posture, and motion patterns of their human counterparts thanks to cutting-edge computer vision capabilities [14]. Given that robots could adapt their activities in real-time to the motions of the human worker, this information is important for tasks that call for close collaboration. Furthermore, robots are currently capable of interacting with human biometric information that goes beyond simple physical movement. Robots can recognize subtle information, such as 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 [15]. A thorough understanding of biometric data based on computer vision algorithms improves productivity while also enabling a more sympathetic and flexible approach to human-robot collaboration. On the other hand, eye-tracking technology enables robots to detect human intents, increasing engagement and adaptability in dynamic working contexts.

2.3. Eye Tracking in Collaborative Robotics

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 information on where people are looking, how long they concentrate on specific things, and their visual attention patterns. This information has great potential to learn the intention and behavior of operators more efficiently and to improve collaborations between the human operator and automation systems [16].

The gaze tracking algorithm is one of the most important algorithms used in eye tracking for cobot applications. The cobot can match its actions to the operator's focus based on this algorithm, which enables the cobot to precisely estimate the location of the operator's gaze [17]. By streamlining assembly procedures, they become simpler to understand and more effective.

Cobots can easily coordinate their movements with the visual cues of the operator based on this level of object awareness [18]. This technology can be used to make object scanning more effective and streamline 3D scanning tasks. Gesturing gains extra depth when paired with object recognition algorithms. Cobots are now able to recognize not just where a worker is looking, but also what they are looking at, based on the object identified by the deep learning algorithms [19]. 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 [20].

Eye and head movements are also important in human-robot interaction because they allow robots to recognize significant areas of interest and respond accordingly to human signals. When humans focus their gaze on certain items or regions, it is a potential indicator of their concentration and intent [21]. Robots with advanced computer vision and eye tracking technology can detect these minor movements in look and head movement, allowing them to figure out the intent and areas of interest triggered by the operators' experiences in their surroundings. This ability allows robots to adapt their actions, prioritize work, and even offer assistance 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 [22].

The current state of research in fully automated laser-based handheld scanning involves advancements in the miniaturization of scanning equipment, improved sensor integration to improve the accuracy of scanning, and real-time processing to enable accurate 3D scanning with handheld devices.

The subsequent methodology section outlines the approach and apparatus used in our study. Section 3.2 includes the process of obtaining scanner marker coordinates using the Convex Hull and Maximum Distance algorithms. Section 3.3 describes the method we used to create the coordinate system of the eye tracker. Section 3.4 describes the algorithm used to convert the 3D point coordinators from the scanner to 2D coordinates to register the markers with the 2D coordinates. The marker registration process we used is described in section 3.5. Followed by the scanning process using Cobot described in section 3.6. We selected a critical area to further understand the effectiveness of the gaze tracking integrated 3D scanning process as described in section 3.7. Following that, section 3.8 explains the matrix we created to compare the performance of the fully autonomous 3D scanning process and the gaze tracking integrated 3D scanning process.

3. Methodology

In this study, the major objective is to enhance the efficiency and flexibility of a robotic scanning system by integrating gaze tracking to identify critical areas of a target object so that the scanning path of the robotic scanning system could be updated accordingly to improve the quality of the scanning. This section presents the methods used to register the marker coordinates acquired from the handheld scanner with the coordinates from the eye tracking system. The integration of the registered coordinates for planning the path of the robotic arm for scanning is discussed in detail, as well as the application of gaze tracking to identify critical areas that require re-scanning. The Formula 1 (F1) model car shown in Figure 1 is utilized as the scanning object for the case study as it is of a free-form nature with a combination of complex and simple geometric features.

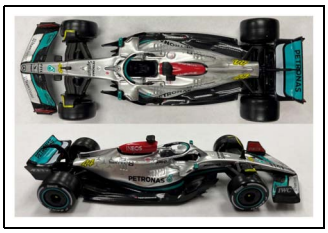


Figure 1: Formula 1 model car used as a scanning object

3.1. Experimental Setup

The gaze movement of the operator is captured using the Argus Science ET Vision glasses. The ET Vision 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 anterior surface of the cornea 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. Corneal reflections appear as bright spots when viewed through the camera. The center of the pupil 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 user’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 estimated accuracy of eye tracker is around 0.5 degrees, allowing a precise measurement of eye movement. The ET vision system also consists of SDK samples for Python programming language to import live eye tracking data through ET remote software.

The scanning of the object was done using a Shinning 3D Einscan HX handheld scanner that was attached to the KUKA KR10 R1100 robotic arm. Rapid Scan and Laser Scan were the two unique scanning modes available with the Einscanner HX. EinScan HX 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 also be applied to optimize rapid scanning.

The laser scan mode can increase precision up to 0.04mm and volumetric accuracy to 0.04+0.06mm/m, with a scanning sped of 480,000 points per second at 55 frames per second (FPS). The camera frame rate in this mode was 55 frames per second. In terms of alignment, laser scan only uses markers to align the surface. The depth of field ranges for Laser Scan was 200 mm to 700 mm and 350 mm to 610 mm.

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 the ISO 9283 standards, it demonstrated its accuracy in movement and placement. The six axes of motion ensure that the robot could move in many different 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.

3.2. Scanner Coordinate System

The scanning of the object was performed using a Shining 3D EinScan HX handheld scanner, as shown in Figure 2. This type of handheld scanners utilizes markers placed onto the surface of the scanning object or platform as reference points to set up the coordinate system for point cloud collection. Unlike coordinate measuring machines or arm scanners, handheld scanners do not have a fixed coordinate system. The relative positions of these markers establish the origin point and the axis directions for point recording. This coordinate system is also utilized to register scans from multiple views for a full-body scan of the object. In our study, these markers are also utilized to align the coordinate systems of the scanner and eye-tracking. In addition, these markers help to maintain the accuracy of the scanning process and minimize errors. Certain objects are challenging for the scanner to distinguish from one region to another as they have repetitive textures or lack unique features. Markers generate distinct points that are easy for the program to recognize and track.

Initially, 18 markers were placed on the scanning table, as shown in Figure 3. Four markers were placed at the corners of the table at an equal distance, forming a perfect square shape, while the other 14 markers were placed randomly within the boundaries set by the four corner markers.

The coordinates of these markers are scanned and exported from the EX Scanner HX software in *.P3* file format. However, the order of these points presented in the raw data file is randomized, which poses difficulties in identifying the corresponding points and aligning them with the physical markers.

Figure 2: Shinning 3D Eiscan HX handheld scanner

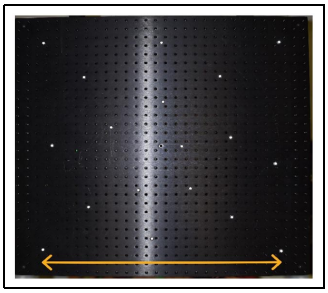


Figure 3: Scanning table with the white color circular markers

Therefore, the first and foremost step involves finding the correspondence between the coordinates and the physical points. Multiple criteria are tested to explore this correspondence.

3.2.1. 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 Euclidean distance is utilized to measure the distance between any two points. The Euclidean distance, $d(p_i, p_j)$ between point $p_i = (x_i, y_i, z_i)$ and point $p_j = \left(x_j, y_j, z_j\right)$ is as follows,

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \qquad (1)$$

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.

3.2.2. 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. In a two-dimensional space, this convex hull is presented as a polygon, 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 to determine the outermost points, which could be considered corners of the overall shape.

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 is capable of detecting extreme or border points in the dataset.

The coordinates identified from the algorithms mentioned above as corner markers are used to determine the distance between each point using the Euclidean distance function. This measurement can be used to 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.

3.3. Eye Tracker coordinates

In order to align the eye-tracking information with the robotic 3D scanning system, the coordinates of the markers from the head-on display with the eye tracking system need to be extracted for registration. A coordinate system can be set up by image processing with the eye tracker and a live area of interest (LAOI) can be set up in the eye tracking system, which can be designated to image objects or features of the environment captured through the scene camera of the eye tracker.

The operator is instructed to look at the scanning table and the four corner markers are designated as the LAOI boundaries as shown in the figure 4. In this study, two LAOIs were created to accompany the eight identified 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 597 mm. After defining the two LAOI regions, the user can walk around the scanning table while the LOAIs remain defined and unchanged regardless of the movement of the user. The gaze movement of the user 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 user from the user’s point of view can be recognized as the origin point of the eye-tracking coordinate system. The coordinate system is shown in Table 1. A visual representation of the two LAOI boundaries is shown in Figure 4.

3.4. Converting 3D Point Coordinates of the EX-Scanner to 2D Coordinates

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

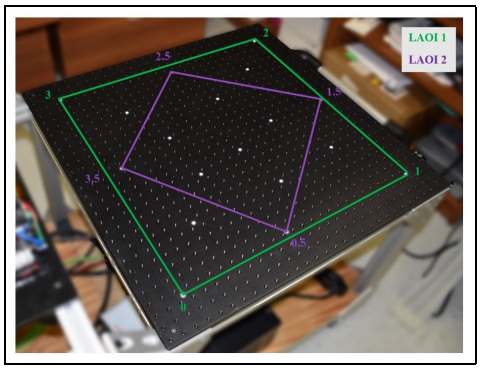


Figure 4: Boundaries of the LAOIs

Table 1: 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

[23] 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, which is 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:

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

Both metrics have an effect on the location of points in the MDS-transformed space, which affects the way the item relationships are represented. This study uses these distances to construct 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.

3.5. 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. Critical regions or areas with poor point cloud quality could be quickly identified by operators, 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. ICP algorithm uses a 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 $A = a_i \,{}_{i=1}^{n}$ and $B = b_i \,{}_{i=1}^{n}$ be the sets of markers, collected by 3D scanning and eye tracking goggle. 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 [24].

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

$$\begin{aligned} c_A &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} a_i, \\ c_B &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} b_i. \end{aligned} \tag{3}$$

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,

$$\begin{aligned} A &= a_i \,{}_{i=1}^{n} = a_i - c_A \,{}_{i=1}^{n}, \\ B &= b_i \,{}_{i=1}^{n} = b_i - c_B \,{}_{i=1}^{n}. \end{aligned} \tag{4}$$

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

As for the pair of points that lack of 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. The stopping condition of the algorithm can be controlled by a pre-set tolerance.

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.

3.6. Scanning with Cobot

In the initial scanning process, the path of the robotic arm was designed to move around the scanning table one time, as shown in Figure 5. This autonomous scanning process is planned based on a path that could uniformly cover the entire object. After finishing this initial automatic scan, the collected 3D point cloud is roughly screened to identify if the areas with complex features or relevant to major functional specifications are scanned properly.

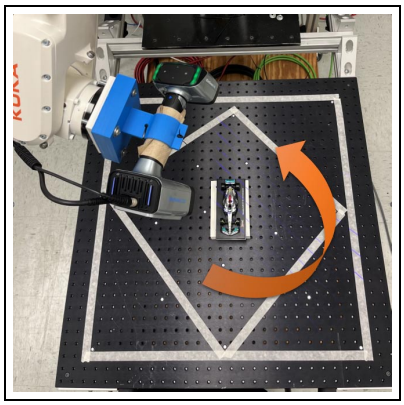


Figure 5: KUKA robotic arm fully autonomous scanning path

In a parallel session, another round of scanning is guided by integrated gaze tracking. The robot moves towards the critical region, guided by the registered gaze position, and a more careful scan of the critical region is performed to collect an additional set of point clouds for detailed description, as presented in Figure 6.

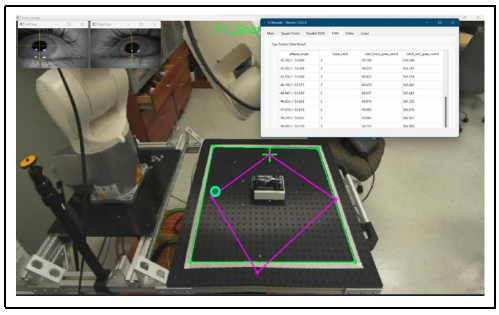


Figure 6: Gaze movement as seen by the scene camera of the eye tracker

In the eye-tracking embedded scanning system, the operator identifies areas of re-scanning based on previous scanning experience. The operator plays a vital role in performing the measurements and critical area identification, as the automated systems lack such judgment skills. Subsequently, the corresponding coordinates are transformed into the robotic arm, which then moves towards the assigned marker with the gaze and scans again to collect additional points. The location of the scanning table in relation to the robot is crucial for its mobility. The robot's end effector can move within a range of proximity, but there are several limitations on how far it can go from its base. Furthermore, due to the torque required at each joint, movement to new positions could not maintain consistent velocities.

3.7. Critical Scanning Area

To further analyze whether integrating human input into the autonomous scanning process through gaze tracking could improve the scanning of complex geometries, we selected a critical area with a high level of detail and curvature. As shown in Figure 7, the cockpit area of the F1 model car is selected as the critical area, as it had more details compared to the overall design of the car.

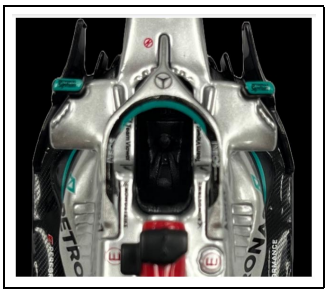


Figure 7: Critical area of the model f1 car

3.8. Performance Metric for 3D Scanning

In order to compare the performance of 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,

$$\rho_i = \frac{N_i}{\frac{4}{3}\pi r^3} \tag{5}$$

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

The curvature, on the other hand, represents the deviation of a surface from being flat at a given location in computational geometry [25]. 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 ν at a point is estimated by fitting a plane to its neighboring points using the least-squares method as described below.

Consider the

$$G_i = U_i s_i V_i^{\mathrm{T}} \tag{6}$$

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

$$d_j = \frac{(G_{ij} - p_i) \cdot n_i}{n_i}, \tag{7}$$

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,

$$\nu_i = 2 \times S_{d_j} \tag{8}$$

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 ($\rho_i \times \nu_i$).This metric integrates the local curvature, ν_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 $\rho * \nu$ 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 $\rho * \nu$ 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 $\rho_i \times \nu_i$ 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 a high $\rho_i * \nu_i$ value for the autonomous scan integrated with eye tracking compared to the fully autonomous scanning process.

4. Results and Discussion

This section presents the results obtained for the methodologies proposed in Section 3.

4.1. Marker registration

The maximum distance and convex hull algorithms were used to identify the coordinates of the four corner markers of the scanner for each of the three data sets. In the maximum distance method, the corner markers were identified according to their maximum distances from each other. The convex hull algorithm typically uses the outermost points that form the smallest convex shape that covers all the points. The code was formulated to give the coordinates of the markers in a counterclockwise manner for both methods.

Table 2: Coordinates of the corner markers from dataset 1

Order	X	Y	Z	MaxDis	CHull
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

Table 2 presents the coordinates of the corner markers, as identified by each algorithm for the data set. It was observed that of the 18 markers, the same coordinates were recognized as the outermost markers by both algorithms. 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. On the contrary, the convex-hull algorithm determined the sequence as 3, 18, 14, and 12. The differences in the order of the marker coordinates obtained from the two methods could be due to the contrasting criteria used in calculating the corners of the polygon. The study only used the coordinates of the identified corner markers, so the order of the arrangement of the markers is not a factor to consider.

These coordinates are confirmed by calculating the lengths of the 4 sides of the square using the Euclidean distance, as shown in Table 3. Upon measuring the physical distance between the corner points, it was found that the length was around 597 mm. It is interesting that the calculated lengths were approximately equal to the physical length of the corner markers.

Table 3: 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

The results show that the order of the markers changes each time a scan is performed. The scanner starts the scanning process by identifying the orientation of the object 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 are identified as 3,12, 14, 18.

The original XYZ (3D) coordinates for the eight markers are converted to XY (2D) coordinate system to be aligned with the 2D coordinates of the eye tracker coordinate system, as presented in Figure 8 and Figure 9. The new original coordinate values and transformed coordinate values are shown in Table 4.

Table 4: Conversion of 3D coordinates to 2D coordinates 3D Coordinates Transformed 2D Coordinates

	3D Coordinates			2D Coordinates	
	X	Y	Z	X	Y
LAOI-1	-99.96	29.55	394.11	-392.29	156.86
	-229.35	554.51	402.75	157.09	394.09
	313.31	669.07	637.30	393.91	-159.50
	437.55	82.91	626.01	-158.69	-391.45
LAOI-2	-160.75	239.65	397.87	241.36	158.34
	42.31	608.28	520.25	131.49	-265.85
	352.56	372.88	622.27	-251.02	-140.09
	141.33	44.77	501.15	-121.83	247.60

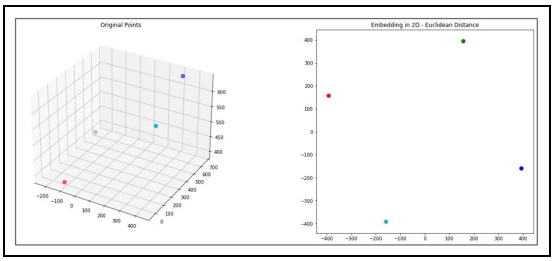


Figure 8: Visualization of 3D to 2D coordinate conversion for LAOI 1

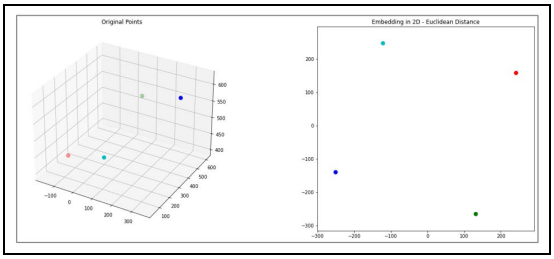


Figure 9: Visualization of 3D to 2D coordinate conversion for LAOI 2

The new transformed 2D coordinates were then confirmed by calculating the Euclidean distance between the points as shown in Table 5. The calculated distances of each pair of points in the 2D system are approximately equal to the corresponding distances in the 3D system.

Table 5: Euclidean distance between the points of 3D coordinates and 2D coordinates

	Points	Lengths in 3D	Lengths in 2D
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

The ICP point registration algorithm is applied to align the 2D coordinates of the scanner with the coordinates of the eye tracking coordinate system. Figure 10 shows that of the 8 corner markers 2.5, 2, and 1 are aligned slightly accurately with slight deviation. However, the rest of the markers have a significant deviation. There could be many reasons for the deviations to occur, particularly with regard to sensor noise and calibration errors of both the scanner and the eye tracker. Even though there is no exact explanation for the deviations of these markers, it is essential to acknowledge that obtaining perfect alignment is challenging in practice.

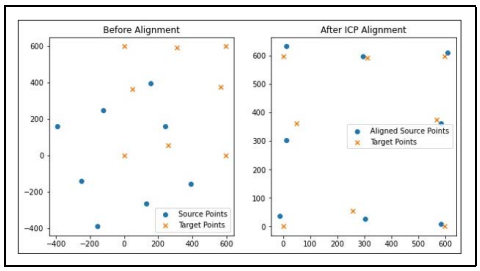
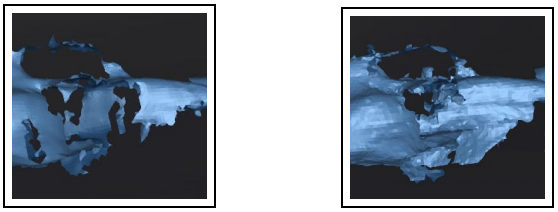


Figure 10: Eye tracker coordinates and scanner coordinates registration



(a) Fully autonomous 3D scanning (b) Gaze tracking integrated 3D scan

Figure 11: Comparison of mesh data of a selected region.

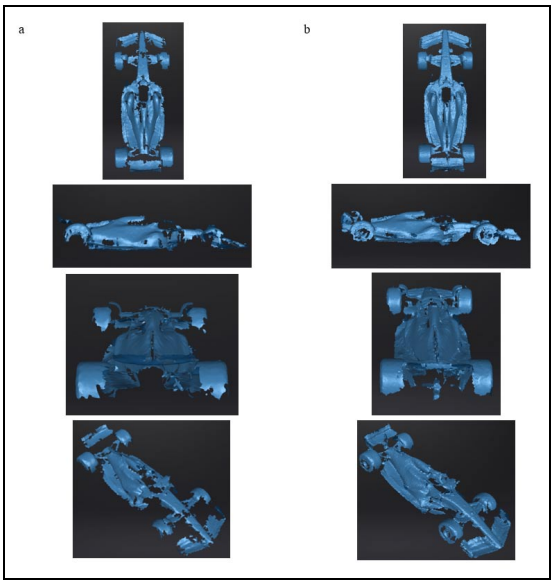


Figure 12: (a) Fully autonomous scan; (b) Gaze tracking integrated scan.

4.2. Object Scan

After a fully autonomous scan, we observe that areas with more details and complex features require eye-tracking integration to improve the scanning results. Based on the observation, the robotic arm was moved 0.5, 3.5, 2, 3.5, the markers with the intention of capturing areas with high density and curvature.

The Figure 11 shows images of a selected region from the F1 model car to interpret the meshdata of the two scans. The images presented in Figure 12 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 to have less detail. The scanned data of the fully autonomous scan exhibits visible irregularities, as several regions show sparse dots, which potentially indicates that the scan did not fully capture all the characteristics of the F1 model car. Closer visual observations indicated possible noise in both scans. To further confirm the metric provided in the study.

4.3. Critical Area Scan

The same procedure was followed to obtain point cloud data for the critical area of the F1 model car. Gaze tracking was integrated into the 3.5, 2.5 marker to optimize the details captured in the process.

4.4. Density and Curvature Metric

The Table 6 depicts the average $\rho_i \times \nu_i$ and minimum $\rho_i \times \nu_i$ values from each fully autonomous scan and eye tracking integrated autonomous scan carried out in the study. All three scans carried out by the fully autonomous scanning process show a modest average $\rho_i * \nu_i$ 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 $\rho_i * \nu_i$, when compared to fully autonomous scans, suggesting that, in general, they contain more complex and detailed regions compared to the first scan. The higher average implies better capturing of intricate details or inherently more complex surfaces.

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

Table 6: Density curvature ($\rho_i * \nu_i$) metric

Method	Scan	Avg($\rho_i * \nu_i$)	Min($\rho_i * \nu_i$)
Fully autonomous	Full scan	0.3543	0
	Critical area	0.3408	0
Gaze tracking integrated	Full scan	0.4410	0
	Critical area	0.4330	0

5. Conclusion and Future Work

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

The metric presented in the study indicated that the gaze integrated autonomous scanning process is compared with the

fully autonomous scanning process. This metric proposed in the paper provides valuable quantitative evaluation and comparison of point-cloud scan complexity and detail, which will be helpful in applications where accuracy and detail are essential, such as reverse engineering and geometric metrology.

A new research direction can be developed based on the proposed metric. Moreover, the coordinate data obtained from eye-tracking can be extracted and streamed in real time to control the movement of the robotic arm. This process involves using the live coordinates of the eye tracking system to guide the movements of the robot arm instantly without delay. Marker registration can also be extended to other markers to increase the variations of the robotic arm.

References

[1] Z. Geng, B. Bidanda, Review of reverse engineering systems – current state of the art, Virtual and Physical Prototyping 12 (2) (2017) 161–172.

[2] Z. Geng, B. Bidanda, Geometric precision analysis for additive manufacturing processes: A comparative study 69 (2021) 68–76.

[3] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, J. Li, Deep learning for lidar point clouds in autonomous driving: A review, IEEE Transactions on Neural Networks and Learning Systems 32 (8) (2021) 3412–3432.

[4] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis, L. Wang, Industry 5.0: Prospect and retrospect, Journal of Manufacturing Systems 65 (2022) 279–295.

[5] J. Yang, T. Liu, Y. Liu, P. Morgan, Review of human-machine interaction towards industry 5.0: Human-centric smart manufacturing, Volume 2: 42nd Computers and Information in Engineering Conference (CIE) (Aug 2022).

[6] F. Vicentini, Collaborative robotics: A survey, Journal of Mechanical Design 143 (4) (Oct 2020).

[7] M. Soori, B. Arezoo, R. Dastres, Artificial intelligence, machine learning and deep learning in advanced robotics, a review, Cognitive Robotics 3 (2023) 54–70.

[8] S. T. Suan CHIN, Influence of emotional intelligence on the workforce for industry 5.0, Journal of Human Resources Management Research (2021) 1–7.

[9] Y. Lu, H. Zheng, S. Chand, W. Xia, Z. Liu, X. Xu, L. Wang, Z. Qin, J. Bao, Outlook on human-centric manufacturing towards industry 5.0, Journal of Manufacturing Systems 62 (2022) 612–627.

[10] A. Pauliková, Z. Gyurák Babelová, M. Ubárová, 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) (2021) 1927.

[11] N. Deshpande, J. Ortiz, I. Sarakoglou, C. Semini, N. Tsagarakis, A. Brygo, J. Fernandez, M. Frigerio, L. Saccares, S. Toxiri, et al., Next-generation collaborative robotic systems for industrial safety and health, WIT Transactions on The Built Environment (Jun 2017).

[12] C. Follini, V. Magnago, K. Freitag, M. Terzer, C. Marcher, M. Riedl, A. Giusti, D. T. Matt, Bim-integrated collaborative robotics for application in building construction and maintenance, Robotics 10 (1) (2020) 2.

[13] A. Hentout, M. Aouache, A. Maoudj, I. Akli, Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017, Advanced Robotics 33 (15–16) (2019) 764–799.

[14] A. Tapus, A. Bandera, R. Vazquez-Martin, L. V. Calderita, Perceiving the person and their interactions with the others for social robotics - a review, Pattern Recognition Letters 118 (2019) 3–13.

[15] A. Chiurco, J. Frangella, F. Longo, L. Nicoletti, A. Padovano, V. Solina, G. Mirabelli, C. Citraro, Real-time detection of worker's emotions for advanced human-robot interaction during collaborative tasks in smart factories, Procedia Computer Science 200 (2022) 1875–1884.

[16] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, O. Khatib, Progress and prospects of the human-robot collaboration, Autonomous Robots 42 (5) (2017) 957–975.

[17] O. Palinko, F. Rea, G. Sandini, A. Sciutti, Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Oct 2016).

[18] V. Krishna Sharma, K. Saluja, V. Mollyn, P. Biswas, Eye gaze controlled robotic arm for persons with severe speech and motor impairment, ACM Symposium on Eye Tracking Research and Applications (Jun 2020).

[19] van Dyck, Leonard Elia, Kwitt, Roland, Denzler, Sebastian Jochen, Gruber, Walter Roland, Comparing object recognition in humans and deep convolutional neural networks-an eye tracking study, Frontiers in Neuroscience 15 (Oct 2021).

[20] O. De Miguel Lazaro, W. M. Mohammed, B. R. Ferrer, R. Bejarano, J. L. Martinez Lastra, An approach for adapting a cobot workstation to human operator within a deep learning camera, 2019 IEEE 17th International Conference on Industrial Informatics (INDIN) (Jul 2019).

[21] A. Klin, W. Jones, R. Schultz, F. Volkmar, D. Cohen, Visual fixation patterns during viewing of naturalistic social situations as predictors of social competence in individuals with autism, Archives of General Psychiatry 59 (9) (2002) 809.

[22] D. Grewal, M. Kroschke, M. Mende, A. L. Roggeveen, M. L. Scott, Frontline cyborgs at your service: How human enhancement technologies affect customer experiences in retail, sales, and service settings, Journal of Interactive Marketing 51 (2020) 9–25.

[23] J. Douglas Carroll, P. Arabie, Multidimensional scaling, Measurement, Judgment and Decision Making (1998) 179–250.

[24] J. Prochazkova, D. Martisek. [link].
URL https://www.researchgate.net/publication/324500004_Notes_on_Iterative_Closest_Point_Algorithm

[25] Z. Geng, A. Sabbaghi, B. Bidanda, Automated variance modeling for three-dimensional point cloud data via bayesian neural networks, IISE Transactions 55 (9) (2022) 912–925.