

Article

Gesture-Controlled Robotic Arm for Small Assembly Lines

Georgios Angelidis *  and Loukas Bampis 

Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; lbampis@ee.duth.gr

* Correspondence: georange12@ee.duth.gr

Abstract: In this study, we present a gesture-controlled robotic arm system for small assembly lines. Robotic arms are extensively used in industrial applications; however, they typically require special treatment and qualified personnel to set up and operate them. Towards this end, hand gestures can provide a natural way for human–robot interaction, providing a straightforward means for control without the need for significant training of the operators. Our goal is to develop a safe, low-cost, and user-friendly system for environments that often involve non-repetitive and custom automation processes, such as in small factory setups. Our system estimates the 3D position of the user’s joints in real time with the help of AI and real-world data provided by an RGB-D camera. Then, joint coordinates are translated into the robotic arm’s desired poses in a simulated environment (ROS), thus achieving gesture control. Through the experiments we conducted, we show that the system provides the performance required to control a robotic arm effectively and efficiently.

Keywords: robotic arm; gesture controlled; RGB-D sensors; machine learning; hand joints detection; ROS



Academic Editors: Bugra Alkan and Malarvizhi Kaniappan Chinnathai

Received: 23 January 2025

Revised: 21 February 2025

Accepted: 23 February 2025

Published: 25 February 2025

Citation: Angelidis, G.; Bampis, L. Gesture-Controlled Robotic Arm for Small Assembly Lines. *Machines* **2025**, *13*, 182. <https://doi.org/10.3390/machines13030182>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robotic science is constantly advancing, addressing increasingly complex challenges to enhance automation in our daily lives. Some of the major sectors that benefit the most from the above advancements include industry, medicine, and unmanned systems. However, modern industry still faces major problems, such as the inability to find specialized personnel, which affects both the production time and the quality of the products. Working accidents are a serious concern, especially in hazardous working environments where workers are exposed to high temperatures, dangerous chemical substances, flammable material, and heavy machinery [1].

In 2011, the German government introduced Industry 4.0 (I4.0). I4.0 follows the third industrial revolution and focuses on the digitalization and connectivity of such applications through advanced technologies, such as artificial intelligence (AI), big data, and cloud computing. In such an ecosystem, machines and systems communicate with their environment in real time in the most efficient “smart” ways [2,3].

Nowadays, we are entering the era of Industry 5.0 (I5.0). I5.0 attempts to reinforce human–robot interaction during industrial processes, ensuring that this type of collaboration occurs seamlessly and intuitively for the user [4]. This approach solves the costly and time-consuming challenge of human adaptation to the technologies introduced by I4.0. This means that users can quickly familiarize themselves with new systems and advanced practices, requiring little to no training for this type of collaboration [5]. This could be crucial in smaller assembly lines where the production is not fixed and there is

a limited workforce, often without specialized knowledge. I5.0 also considers concepts such as livability and economic challenges [6]. It is a broader, more global approach to manufacturing that emphasizes a human-centric industry. This type of industry aims to enhance productivity while also ensuring the ethical integrity of industrial processes [7].

In this context, robots apply to a variety of everyday human activities, often beyond the borders of industry, such as healthcare, agriculture, and entertainment [1,8]. The most common categories of modern robots are mobile agents [8,9], surgical robots [10], household robots [11], and of course, industrial robots [12]. Industrial robots are used in factory environments, performing demanding and hazardous assignments, reducing potential risks for the personnel. They are often utilized to replicate or augment human actions as they perform tasks more accurately. In semi-manual processes, robots collaborate directly with humans, thus enhancing productivity and preventing the risk of human errors. Furthermore, while replacing manual tasks, robots are usually equipped with sophisticated sensors and control systems to handle tools or materials similarly to human workers. This category of robots, especially articulated robotic arms, is designed and constantly developed to complete various operations in an industrial environment.

To successfully complete the desired tasks, the robotic arm's end effector is equipped with a wide variety of tools, such as grippers. Grippers are used for object manipulation, and they are divided into two main categories: flexible and rigid. On the one hand, flexible grippers are made from adaptable materials and are utilized for grasping irregularly shaped objects [13]. On the other hand, rigid grippers usually grip large, non-deformable objects [14]. Each gripper type suits different applications based on the manipulated object characteristics. Industrial robotic arms are mainly used for applications such as assembly [15,16], polishing [17,18], finishing with the help of specialized tools [19], and common manual tasks, such as packaging [20] and material handling [21]. Automation through robotic arms significantly reduces operational costs, establishing more efficient and competitive production practices [1]. Most of the time, robotic systems themselves are not able to keep up with I5.0's principles, so they often integrate other technologies, such as AI [22].

AI has revolutionized modern industry by introducing innovative methods and approaches. AI increases the efficiency of maintenance predictions for the equipment involved, leading to reduced downtime [23]. It contributes to sustainable logistics management by predicting high-demand periods and managing supplies [24]. AI is also a common integration in quality control applications utilizing computer vision technology [25,26]. AI's role in modern automated production lines, where robotic systems need to adapt to various and dynamic tasks, is crucial as it can be used to create a safe and user-centric human-robot interaction environment, paving the way for fully adapting to I5.0's guidelines. One of the most representative applications of AI in robotic systems involves their control, either through data-driven object manipulation [27] or by enabling the intuitive translation of users' gestures into motion commands [28].

Gesture-controlled robotic applications utilize human movement recognition to achieve control. This approach offers significant advantages over traditional methods, such as keyboards and joysticks, as it allows direct control commands in a human-centric manner. Gesture control is possible by various sensors such as accelerometers [29,30] and depth cameras [31,32], which provide direct depth measurements alongside the visible spectrum data. Combined with AI, they are able to estimate human poses [33], providing at the same time visible feedback, which is crucial for the system's operator.

Even though the majority of gesture control systems rely on wearable devices [34,35], there exist some non-wearable-based methods that utilize AI and depth camera sensors. However, they do not consider the orientation of the user's hand, thus failing to provide

precise gripper control [36,37]. To address this gap, we propose a complete system that combines the above. Our system identifies the pose and the hand gestures of an operator, using state-of-the-art machine learning (ML) techniques based on an RGB-D camera, and directly translates them to control commands of a robotic arm, targeting small assembly lines (Figure 1). Our novelties include the following:

- **3-Dimensional Gesture Control:** We present a complete robotic arm control system for object manipulation that mimics the motion, the orientation of the operator's arm, and hand gestures in the 3D space to effectively grip objects in a production line.
- **AI-Driven Hands-Free Control:** The introduction of AI and ML allows a user-friendly control scheme without the need for any additional devices equipped by the operator.
- **Affordability:** The proposed system provides a cost-effective solution that solely requires the addition of a depth camera sensor to control a robotic arm.
- **Versatility and Adaptability:** The proposed work offers a versatile approach that can be adapted to various environments, operations, and robotic arm models as it decouples the robot's control from the gesture recognition pipeline.

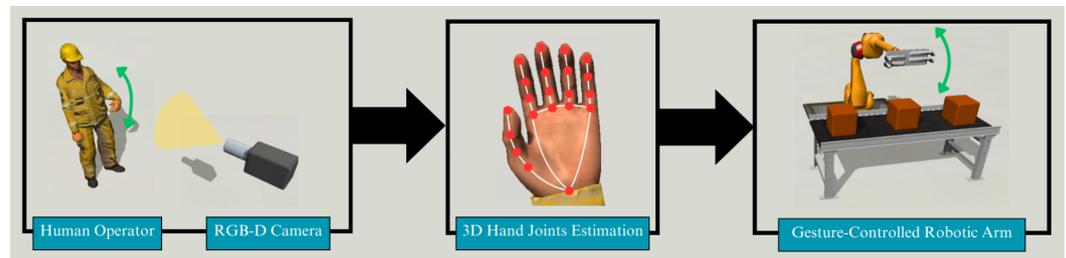


Figure 1. Schematic representation of our proposed gesture-controlled robotic arm system.

The rest of this paper is organized as follows: Section 2 presents the related work on the topic of gesture-controlled robotic arms. Section 3 introduces our system's methodology, while Section 4 describes the experimental setup and evaluation of our system. Finally, Section 5 draws our conclusion and discusses possible extensions of our work.

2. Related Work

Since antiquity, there has been interest in constructing machines that have characteristics similar to modern robots. The intensive development of robots started during the Industrial Revolution. In 1954, George Devol introduced the first industrial programmable robotic arm, a moment that defined the first generation of industrial robots (1950–1967) [38]. During the second generation of industrial robots (1968–1977), extensively utilizing sensors, industrial robots were established in complex tasks in the automotive industry, while during the third one (1978–1999), the industry saw the integration of programming languages and specialized controllers in robotic systems. This means operation with hardly any human supervision. The fourth generation of industrial robots (2000–2017) keeps up with the rapidly increased computational power and the development of sophisticated sensors. Fourth-generation robotic arms showed increased intelligence and adaptability. During this time, the concept of robots who cooperate immediately with humans (cobots), ensuring the safe execution of this type of interaction, was introduced [1,2]. Nowadays, robotics focuses on utilizing robots in order to make everyday activities easier. The fifth generation of industrial robotics (2018–present) are usually versatile articulated robots, such as robotic arms. For this reason, fifth-generation robots are important in I5.0's systems [1]. Following the above evolution, the need for excessively advanced robot control has emerged. In what follows, an analysis of relevant studies is presented regarding gesture-based approaches for controlling a robotic arm.

Aggarwal et al. [34] utilized inertial sensors, which recognize four hand gestures, to control a robotic arm wirelessly for pick-and-place tasks via radio frequency (RF) signals. They also integrated an IP camera into their system to provide real-time video information to the user. Similarly, Pradeep and Paul [35] developed an accelerometer-based system to remotely control a three-degree-of-freedom (DOF) robotic arm. They provided video data and temperature measurements from the robot's surroundings in order to enable a more secure remote control. Khajone et al. [39] used a conventional webcam's output to recognize gestures from a custom database stored in a computer using correlation. Every predefined gesture is translated into a corresponding robotic arm movement. Their algorithm is able to detect gestures regardless of their scale. This system also uses an RF module to transmit data to the arm. Megalingam et al. [36] developed a system for controlling a two-link robotic manipulator using a Kinect sensor. Through Kinect's SDK, they estimated the angles between the joints of the user's arm. These angles were used to control the movements of 3 servo motors, achieving the desired robot's pose in this way. Their algorithm does not include finger estimations. Zidarić [37] enabled gesture control by using an RGB-D camera sensor and calculating the 3D position of the user's hands. His system can also detect an open or closed palm to adjust the end effector's opening accordingly. The above highlights the extensive focus on research aimed at developing user-friendly gesture-controlled robotic arms within the I5.0's protocol. However, there is still the need for less invasive and more intuitive control schemes that do not depend on specific gestures to perform predefined robot movements.

An innovative and efficient way of achieving natural interaction with a robotic arm can be achieved through human pose estimation, which is enabled through AI and ML techniques. In the related literature, several AI-based approaches have been proposed for joint recognition that take advantage of the human body to allow remote control. Stergiopoulou et al. [40] proposed a method for real-time hand detection that combines motion detection, skin tone classification, and morphological features to enhance accuracy in complicated backgrounds. Choi et al. [41] compared four deep-learning methods (MediaPipe [42], Hybrid Inverse Kinematics solution [43], Multi-Hypothesis Transformer [44], and Diffusion-based 3D Pose Estimation [45]) dedicated to 3D human pose and joint estimation, evaluating their performance on real-world video streams under various conditions. They concluded that the algorithms showed reliable performance in uncomplicated poses, unobstructed frames, and conditions of constant lighting. Mitrovic and Milošević [46] introduced a method of monitoring human activities to minimize the risk of injuries. Using MediaPipe, they extracted joint data to assist medical professionals. Kasar et al. [47] employed hand joint estimation to convert hand coordinates into real-time computer cursor movements. They also included a series of gestures, each triggering a distinct cursor action.

The proposed system effectively incorporates AI-based joint detection techniques into the existing literature on gesture-controlled robotic arms. Unlike many examples from the available literature [34,35], which rely on wearable equipment, our approach allows users to control the robotic arm solely through their intuitive motion, eliminating the need for equipping additional devices. It also provides precise gripper control that follows the hand's opening and closing gestures, a feature missing from the majority of vision-based techniques [36,37].

3. Methodology

Figure 2 shows the setup of the proposed system. A depth camera is mounted at the height of the user's chest (x) after calibrating the visible spectrum data through a proper pattern [48]. In addition, depth data aligned to the visible spectrum measurements for an accurate representation of the scene [49]. The user is positioned at a fixed distance d , where

their movements and hands are easily distinguishable in the camera frame. The considered frames of reference are also depicted, with s corresponding to the camera sensor, c to the operator's chest, b to the robot's base, and g to the robot's gripper. The frame of reference for the operator's hand is divided into h_t , which is considered parallel to s and follows the hand's position (translation-only), and h_r , which is considered attached to h_t but follows the hand's orientation (rotation-only). The goal is to calculate the homogenous transformation matrix ${}^c_{h_r}T$, from the user's hand to the user's chest, for every hand's pose and convert its values to the transformation matrix b_gT from the end effector to the robot's base.

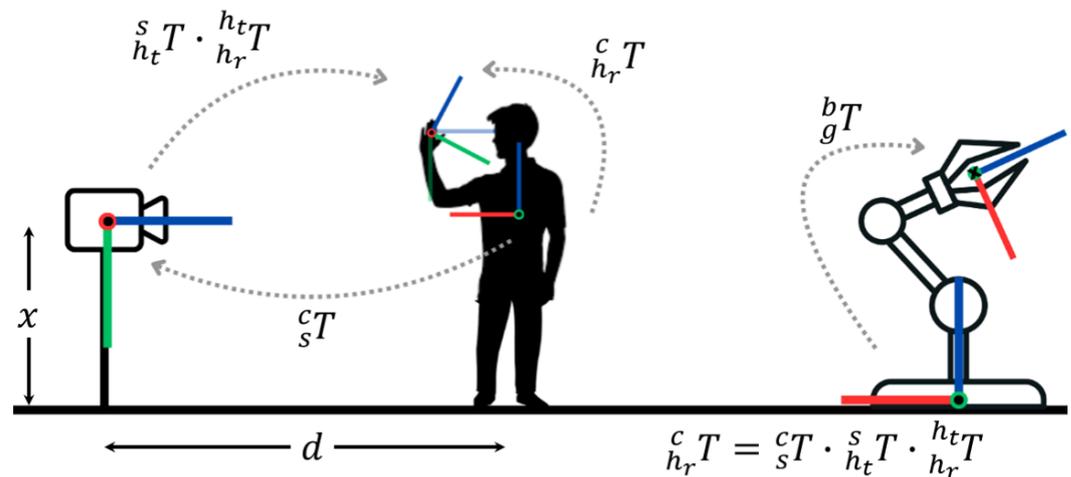


Figure 2. Proposed setup for controlling a robotic arm through human gestures. The frames considered for reference are depicted in red for the x -axis, green for the y -axis, and blue for the z -axis.

The introduced system is compatible with robotic arms of varying DOFs. A 6-DOF manipulator is able to reach any point within its workspace with an arbitrary end-effector orientation, while arms with fewer DOFs cannot. Robotic arms with more than 6 DOFs are often utilized in applications that require enhanced flexibility, such as obstacle avoidance. A higher DOF robotic arm used with the proposed methodology can adapt its movements to the user rather than the other way around. Overall, the proposed system applies to any robotic arm as long as the target end effector's pose is within the robot's workspace.

3.1. Hand Joint Estimation

An ML model based on MediaPipe [42,50] is used for tracking the 21 main hand joints, as seen in Figure 3. The ML model uses a single-shot detector (SSD) to estimate the palm's position. After the palm's detection, the hand joint positions are estimated through regressions and provided in pixel values. The proposed system utilizes four hand joints for its operation: the wrist, the thumb's tip, the index finger's tip, the index finger's proximal interphalangeal (PIP) and metacarpophalangeal (MCP) joints, and the pinky's MCP, or in Figure 3 joints 0, 4, 8, 6, 5 and 17, respectively.

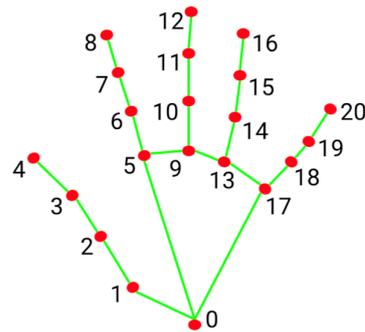


Figure 3. The 21 estimated hand joints from the work presented in [42,50].

3.2. Transforming Hand Pose Coordinates to Camera Coordinates

The full pose (6 DOF) of the human hand is considered for controlling the robotic arm. We choose to divide translation and rotation into two different subsections for better clarity. Here, the computation of the hand's transformation is described, while Section 3.3 is dedicated to the computation of its orientation.

As mentioned above, the frames h_t and h_r are considered to share the same origin, which is assigned to the midpoint $p_m = [x_m, y_m, z_m]^T$ between thumb and index tips. This is computed by combining depth information provided by the RGB-D camera sensor, the pinhole camera model [51], as described by the following projection matrix:

$$P = K[R|t]. \quad (1)$$

In the above, K corresponds to the intrinsic parameters of the camera, R refers to the camera's rotation, and t is the camera's translation. The camera's axes are placed according to common conventions [52]. In this way, we can compute the transformation matrix between frames h_t and s as

$${}^s_{h_t}T = \begin{bmatrix} 1 & 0 & 0 & x_m \\ 0 & 1 & 0 & y_m \\ 0 & 0 & 1 & z_m \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

3.3. Calculating Hand's Orientation

Two direction vectors are calculated to choose the axes in the coordinate system that describe the hand's orientation. For the first one, its starting point is at the index PIP joint and its endpoint is at the user's wrist and, respectively, at the pinky MCP and index MCP joints for the other vector. These vectors are computed via

$$\vec{v} = (x_{joint_2} - x_{joint_1}, \quad y_{joint_2} - y_{joint_1}, \quad z_{joint_2} - z_{joint_1}). \quad (3)$$

These vectors are depicted in Figure 4, and they are normalized to unit vectors, resulting in the z_{h_r} -axis and the x_{h_r} -axis. Then, the y_{h_r} -axis is calculated as the cross-product of the x_{h_r} -axis and the z_{h_r} -axis. Finally, the x_{h_r} -axis is computed once more as the cross-product of the y_{h_r} -axis and z_{h_r} -axis to ensure the orthonormality of the defined reference frame. Based on the above, the hand's orientation, and thus the transformation matrix between frames h_t and h_r , is computed as

$${}_{h_r}^{h_t}T = \begin{bmatrix} R(x_{h_r}, y_{h_r}, z_{h_r}) & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

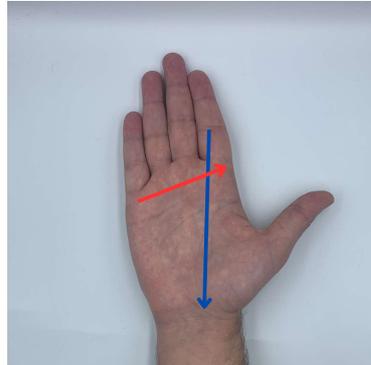


Figure 4. Direct vectors computed between the index PIP joint and the wrist (blue), pinky MCP, and index MCP joints (red). These vectors are used as references for computing the orientation of the operator’s hand (frame h_r).

3.4. Transforming Hand Coordinates to Chest Coordinates

To convert hand coordinates to chest coordinates, the transformation matrix c_sT is required, which is constant. The camera’s coordinate frame is shifted d units away from the reference frame on the human’s chest. Then, the chest’s coordinate system is rotated 90° around the z-axis and then -90° around the x-axis’s new position to become identical to the camera’s coordinate system, as shown in Figure 5. Consequently, the camera’s reference frame is described relative to the chest as

$${}^c_sT = \begin{bmatrix} 0 & 0 & -1 & d \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

Finally, the matrix that transforms the hand’s coordinates to the chest’s is calculated through

$${}^c_{h_r}T = {}^c_sT \cdot {}^s_{h_t}T \cdot {}^{h_t}_{h_r}T. \quad (6)$$

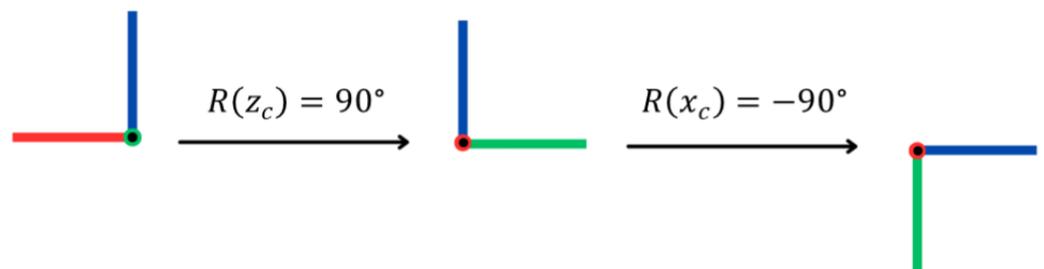


Figure 5. Rotations of the user’s chest frame of reference (c) in order to align its axes with those of the camera’s coordinate frame (s).

3.5. Scaling to Robotic Arm

To determine the gripper's position and orientation b_gT , the computed hand's pose needs to be scaled to the robotic arm's geometric characteristics and reach, as the gripper's orientation matches the human hand's one. If the following is assumed, it is now possible to transform ${}^c_{h_r}T$ into b_gT :

- $x_{b,max}, y_{b,max}, z_{b,max}$ and $x_{b,min}, y_{b,min}, z_{b,min}$ are the maximum reach of the robotic arm along each axis in its base reference frame (b), "max" corresponds to the positive direction while "min" corresponds to the negative direction of the respective axis.
- $x_{c,max}, y_{c,max}, z_{c,max}$ and $x_{c,min}, y_{c,min}, z_{c,min}$ is the furthest extent of the human arm with respect to the coordinate system c , where "max" is dedicated to the positive direction while "min" refers to the negative direction of the respective axis.

The relationship that translates hand coordinates to robot coordinates is given by the following normalization formulas:

$$\begin{aligned}x_b &= \frac{x_c - x_{c,min}}{x_{c,max} - x_{c,min}} \cdot (x_{b,max} - x_{b,min}) + x_{b,min} \\y_b &= \frac{y_c - y_{c,min}}{y_{c,max} - y_{c,min}} \cdot (y_{b,max} - y_{b,min}) + y_{b,min} \\z_b &= \frac{z_c - z_{c,min}}{z_{c,max} - z_{c,min}} \cdot (z_{b,max} - z_{b,min}) + z_{b,min}\end{aligned}\quad (7)$$

In order to deal with the possibility of the operator appearing in the frame but not in the correct position, leading to out-of-bound hand positions, out-of-range values are handled by setting the robotic arm to its maximum reach, until the correct control position is assumed.

3.6. Controlling the Gripper's Opening Width

The maximum gripper opening width is denoted as w_{max} . The Euclidean distance between the thumb tip and index tip is measured on the detected hand joints after their projection to the 3D space to specify the opening width of the robotic gripper (w). Since the motion of the gripper is symmetrical, w is halved and assigned to each of the robotic arm's fingers. To prevent the gripper, from attempting actions beyond its physical limitations, the gripper's opening span was set as $\min(w, w_{max})$.

4. Experiments

4.1. Experimental Setup

For the experimental evaluation of our system, we used a ROS-based simulated version of Franka Emika's Panda robotic arm, a 7 DOF robotic manipulator. The end-effector tool we utilized was a rigid gripper with a maximum opening span of 80 mm. The gripper selection does not affect the outcome of the experiments as the pressure applied by the gripper falls outside of this study's scope. A high DOF robotic arm offers significant flexibility, allowing the end effector to approach target points with multiple orientations while being able to avoid obstacles. The arm was visualized in RViz, and the MoveIt! package handled the inverse kinematics, motion planning, and control. An open-source, high-level programming language was used for implementing the described methods and managing the processes, as in [53]. The RGB-D sensor we utilized was the Intel RealSense D435i RGB-D camera, which provides depth measurements through stereoscopic vision technology. The depth sensor arrived factory pre-calibrated, and the Depth Quality Tool provided by the camera's SDK did not indicate problems in the calibration of the sensor. Finally, the visible spectrum module was calibrated using the Camera Calibrator tool provided by the MATLAB R2018b suite, using a printed copy of a chessboard pattern (each

side of each square was measured at 96 mm). 25 images (848×480 pixels) were provided to the calibration app, in each of which the pattern was easily visible in the frame. Table 1 presents the estimated intrinsic parameters.

Table 1. Camera parameters as calculated from MATLAB’s calibration tool.

Category	Parameter	Value
Camera Intrinsics	FocalLength	[590.6891, 588.9445]
	PrincipalPoint	[418.0332, 254.8811]
	Skew	0.1962
	RadialDistortion	[0.0672, −0.1554]
	TangentialDistortion	[−0.0015, −0.0026]
	ImageSize	[480, 848]
Accuracy of Estimation	MeanReprojectionError	0.3230
Calibration Settings	NumPatterns	25
	WorldPoints	[54×2 double]
	WorldUnits	“millimeters”
	EstimateSkew	1
	NumRadialDistortionCoefficients	2
	EstimateTangentialDistortion	1

The depth camera was set at a height of $x = 141$ cm, while the operator was placed in a fixed position $d = 130$ cm. The reach of our user’s hand was measured in millimeters: $x_c \in [-600, 600]$, $y_c \in [-700, 700]$, and $z_c \in [-350, 700]$. Similarly, the reach of the robotic arm was measured in millimeters: $x_c \in [-770, 770]$, $y_c \in [-700, 700]$, and $z_c \in [-100, 900]$.

4.2. Results

Since the experimental setup of the proposed system uses a simulated robotic arm, only the hand joint detection and the measurements of the depth camera sensor introduce noise. The entirety of the computations is based on the above, and thus, they are further studied and discussed in the section below. Any additional experiments would not accurately reflect the performance of this setup due to the simulation characteristics. Therefore, we are reluctant to provide further evaluations.

4.2.1. Hand Reference Pose Selection

Two types of palm reference poses were studied for controlling the robotic arm: (i) the palm surface perpendicular to the camera and (ii) the palm surface parallel to the camera. Figure 6 illustrates the hand approaching the same point by the two considered categories. For each studied case, 10 depth measurements were taken after fixing the user’s hand at 1220 mm from the camera sensor. These measurements are presented in Table 2. As can be seen, the mean absolute error of reference Pose 2 was 18.8 mm, which is why it was selected for the rest of our study. It is worth mentioning that when the reference pose of the hand was changed, and the depth estimation was not affected, the pose of the robotic arm remained the same.

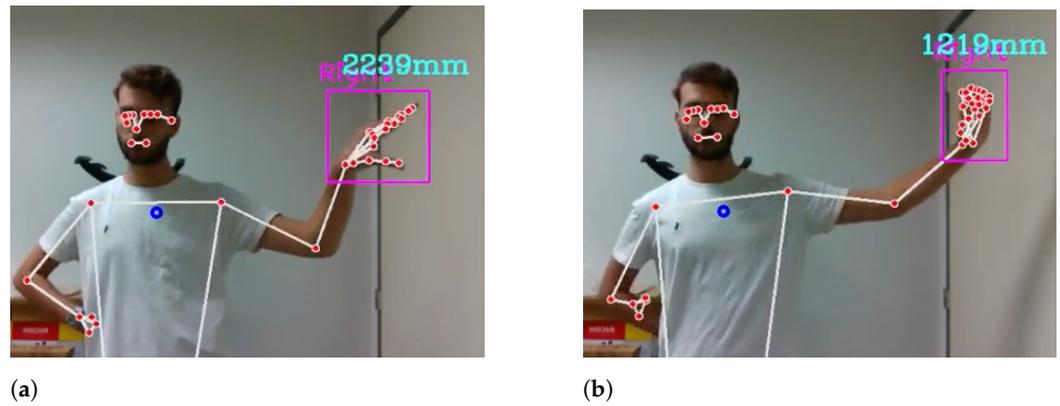


Figure 6. The two categories of studied hand poses for controlling the robotic arm. (a) Pose 1, the palm's surface is perpendicular to the camera; (b) Pose 2, where palm appears parallel relative to the camera.

Table 2. Depth estimations provided by the RGB-D camera for the two classes of examined hand's orientation. The hand's real depth was 1220 mm.

Measurement No.	Pose 1 (mm)	Pose 2 (mm)
1	2239	1219
2	2238	1219
3	2239	1219
4	2236	1216
5	2240	1220
6	2239	1219
7	2239	1220
8	2237	1217
9	2237	1219
10	2238	1220
Mean	2238.2	1218.8
Variance	1.511	1.733
Std. Deviation	1.229	1.316

4.2.2. Fine-Tuning the Gripper Opening Width

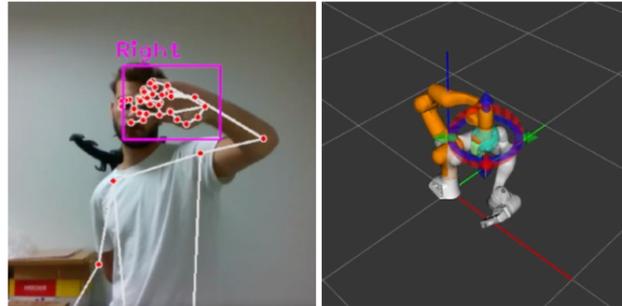
It was observed that when the palm was widely open, or the actual distance between the human fingers tended towards zero, depth estimation became inaccurate due to the stereo depth module's inability to estimate the depth of small surfaces such as fingertips correctly. Because of this, the distance value w calculated by our algorithm became significantly high, reaching approximately 60 m. It was considered that an open palm can not extend 0.5 m. In this way, the gripper opening width was calculated as described in Section 3.6. However, if w values were greater than 0.5 m, the gripper's opening would be set to zero.

4.3. Overall System Performance

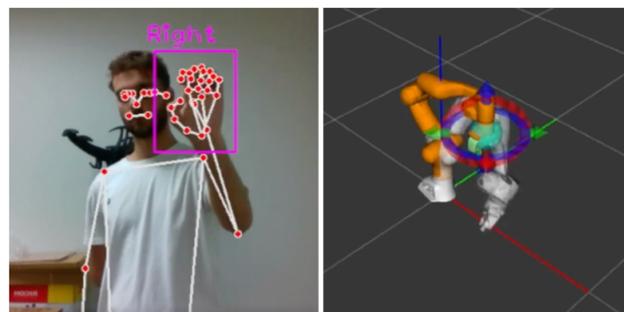
Combining the above, a gesture-controlled robotic arm was developed that mimics the movements of its operator's hand, using a depth camera. Two different wrist orientations for the same hand position are illustrated in the left column of Figure 7, while the right column displays the corresponding poses of the panda robotic arm for each hand pose. Figure 7 indicates that the robotic arm can be intuitively utilized in pick-and-place applications due to how the orientation of the end effector was defined.

Similarly, Figure 8 illustrates the control of the gripper. Specifically, images on the left show the gripper at its maximum opening, while images on the right demonstrate the

gripper adapting to an appropriate opening as the palm closes. Finally, Figure 9 depicts 33 hand positions and the corresponding positions of the gripper during the system's operation. The graphs demonstrate that the robotic arm followed the detected hand's positions with sufficient accuracy.

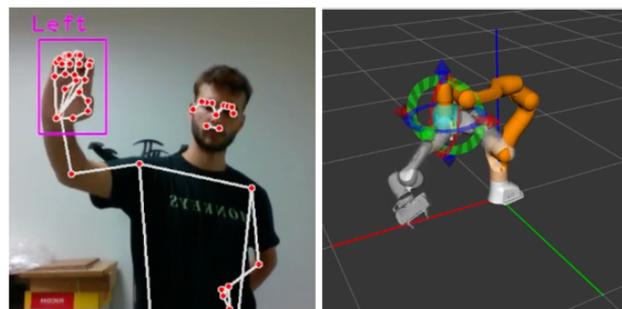


(a) Horizontal orientation of the user's hand which results in the equivalent pose of the robotic arm.

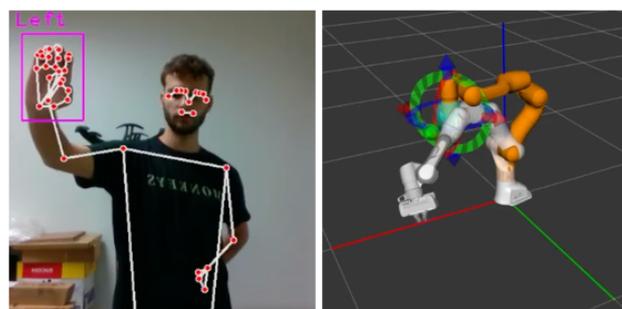


(b) Vertical orientation of the palm which sets the panda arm in a vertical, face-down position.

Figure 7. Snapshots of the developed system's operation.



(a) The user's palm fully open (left), resulting in maximum opening span w_{max} , for the gripper (right).



(b) Almost closed user's palm (left) as the gripper opening width tends to become zero (right).

Figure 8. The two boundary conditions for the gripper's opening w values.

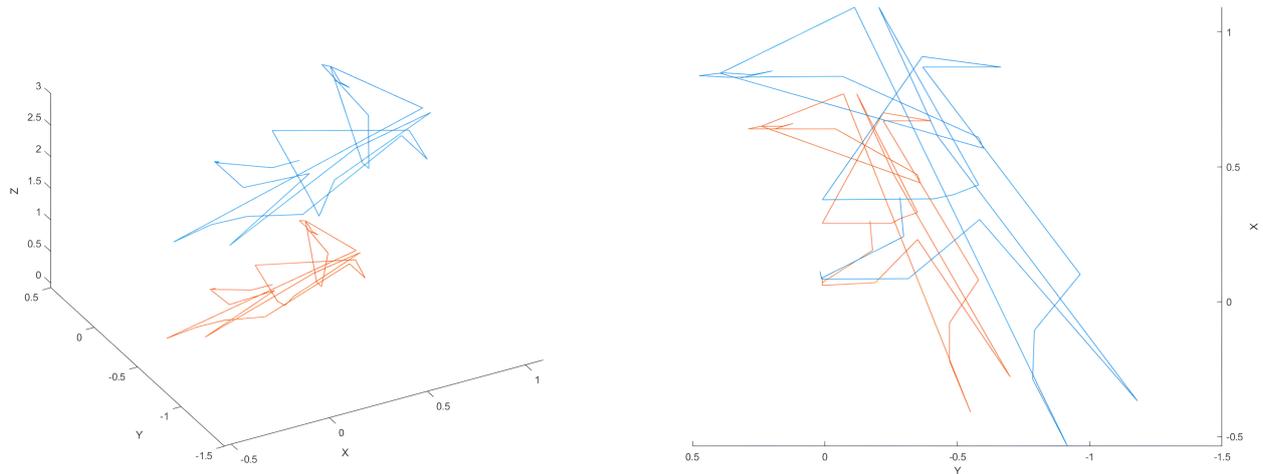


Figure 9. The trajectory of the user’s hand, with respect to the frame of reference c , and the end effector, with respect to b . 33 points are depicted with blue for the human hand and orange for the Panda arm. The left graph is a side-view comparison of the movements, while the right one illustrates the same sequence from a top view.

5. Conclusions

In this paper, a versatile gesture-controlled robotic arm system for small assembly lines was presented, which integrates ISO guidelines. Our setup utilizes AI to enable intuitive control without requiring any additional equipment for the user to achieve human–robot interaction. Furthermore, it provides a reasonable solution for non-repetitive, customized operations that typically take place in small-scale manufacturing lines, and it can be applied in any modern collaborative robotic arm platform. Finally, the experimental setup of the proposed design is presented, and the methodology was tailored to the Panda robotic arm. The obtained results show that our proposal can achieve precise and user-friendly control of a robotic arm.

The precision of our system is mainly affected by the quality of the depth camera’s measurement, as poor depth estimation, when combined with the pinhole model, can lead to incorrect hand position estimations. Even though the selected depth sensor does not provide exceptional accuracy, especially in cases of small surfaces, the observed deviation was minimal, and it did not affect our experiment’s outcome. However, in cases where millimeter precision is required, the utilized depth sensor is not adequate.

Greater precision can be achieved using setups that estimate hand point coordinates by multiple high-resolution sensors [54,55] or through the combination of specialized ML models [56] to retain low equipment cost. Furthermore, considering the elbow joint in future iterations of our system will allow users to select between elbow-up and elbow-down configurations for the robotic arm in cases where the latter holds the required DOFs. This addition would improve the adaptability of our system and provide more intuitive control, allowing the robotic arm to replicate the movements of the user’s arm with greater accuracy and avoid obstacles with better ease.

Additionally, integrating this system into a real industrial environment using a physical robot would provide valuable insights. The proposed system is easy to integrate into various platforms and industrial settings. It supports a wide range of manipulators, including multiple kinds of end effector tools, DOF, and conventional or double encoder joints, to enable robust control and enhance torque sensing [57]. The remote control of the robotic arm keeps operators safe from potential hazards. Moreover, real-time replication of human arm movements and gestures allows the use of the system in non-repetitive tasks.

This could have a significant impact on smaller enterprises, helping them transition from traditional manufacturing techniques to automated ones. Finally, integrating the proposed system into an actual assembly line is valuable, as collecting feedback from personnel using the system is crucial for further optimization and enhancing its applicability in real-world scenarios.

Author Contributions: Conceptualization, L.B.; Methodology, G.A. and L.B.; Software, G.A.; Validation, G.A.; Formal analysis, G.A.; Resources, G.A.; Writing—original draft, G.A.; Writing—review & editing, L.B.; Visualization, G.A.; Supervision, L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

I4.0	Industry 4.0
I5.0	Industry 5.0
AI	Artificial Intelligence
ML	Machine Learning
RF	Radio Frequency
DOF	Degree of Freedom
CNN	Convolutional Neural Network
SSD	Single-shot Detector
PIP	Proximal Interphalangeal
MCP	Metacarpophalangeal

References

1. Barua, R.; Datta, S. Modernization of Robotics Application in the 21st Century: A Review. *J. Mech. Robot.* **2020**, *5*, 26–36. [[CrossRef](#)]
2. Alcácer, V.; Cruz-Machado, V. Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 899–919. [[CrossRef](#)]
3. Hermann, M.; Pentek, T.; Otto, B. Design Principles for Industrie 4.0 Scenarios. In Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 5–8 January 2016. [[CrossRef](#)]
4. Humayun, M. Industrial Revolution 5.0 and the Role of Cutting Edge Technologies. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 605–615. [[CrossRef](#)]
5. Lei, Z.; Shi, J.; Luo, Z.; Cheng, M.; Wan, J. Intelligent Manufacturing from the Perspective of Industry 5.0: Application Review and Prospects. *IEEE Access* **2024**, *12*, 167436–167451. [[CrossRef](#)]
6. Masoomi, B.; Sahebi, I.G.; Ghobakhloo, M.; Mosayebi, A. Do Industry 5.0 Advantages Address the Sustainable Development Challenges of the Renewable Energy Supply Chain? *Sustain. Prod. Consum.* **2023**, *43*, 94–112. [[CrossRef](#)]
7. Almurib, H.A.F.; Al-Qrimli, H.F.; Kumar, N. A Review of Application Industrial Robotic Design. In Proceedings of the 2011 Ninth International Conference on ICT and Knowledge Engineering, Bangkok, Thailand, 12–13 January 2012. [[CrossRef](#)]
8. Agelli, M.; Corona, N.; Maggio, F.; Moi, P.V. Unmanned Ground Vehicles for Continuous Crop Monitoring in Agriculture: Assessing the Readiness of Current ICT Technology. *Machines* **2024**, *12*, 750. [[CrossRef](#)]
9. Peck, C.J.; Langedock, K.; Boone, W.; Fourie, F.; Moulart, I.; Semeraro, A.; Sterckx, T.; Geldhof, R.; Groenendaal, B.; Ponsoni, L. The Use of Autonomous Underwater Vehicles for Monitoring Aquaculture Setups in a High-Energy Shallow Water Environment: Case Study Belgian North Sea. *Front. Mar. Sci.* **2024**, *11*, 1386267. [[CrossRef](#)]
10. Pisla, D.; Al Hajar, N.; Gherman, B.; Radu, C.; Antal, T.; Tucan, P.; Literat, R.; Vaida, C. Development of a 6-DOF Parallel Robot for Potential Single-Incision Laparoscopic Surgery Application. *Machines* **2023**, *11*, 978. [[CrossRef](#)]
11. Rathor, T.; Ali, T.; Khan, M.F.; Kumar, S.; Khan, Z.Z. Domestic Household Multi-Purpose Robot. *Int. J. Res. Publ. Rev.* **2024**, *5*, 6021–6028.

12. Wrede, K.; Zarnack, S.; Lange, R.; Donath, O.; Wohlfahrt, T.; Feldmann, U. Curriculum Design and Sim2Real Transfer for Reinforcement Learning in Robotic Dual-Arm Assembly. *Machines* **2024**, *12*, 682. [[CrossRef](#)]
13. Xu, M.; Liu, Y.; Li, J.; Xu, F.; Huang, X.; Yue, X. Review of Flexible Robotic Grippers, with a Focus on Grippers Based on Magnetorheological Materials. *Materials* **2024**, *17*, 4858. [[CrossRef](#)] [[PubMed](#)]
14. Mehta, S.A.; Kim, Y.; Hoegerman, J.; Bartlett, M.D.; Losey, D.P. RISO: Combining Rigid Grippers with Soft Switchable Adhesives. *arXiv* **2022**, arXiv:2210.15791.
15. Krüger, J.; Schreck, G.; Surdilovic, D. Dual Arm Robot for Flexible and Cooperative Assembly. *CIRP Annals* **2011**, *60*, 5–8. [[CrossRef](#)]
16. Sitjongsataporn, S.; Paenoi, N. Development of Single-Arm Robots for Automotive Body Assembly Automation. In Proceedings of the 2023 International Conference on Power, Energy and Innovations (ICPEI), Phrachuap Khirikhan, Thailand, 18–20 October 2023.
17. Kharidege, A.; Ting, D.T.; Yajun, Z. A Practical Approach for Automated Polishing System of Free-Form Surface Path Generation Based on Industrial Arm Robot. *Int. J. Adv. Manuf. Technol.* **2017**, *93*, 3921–3934. [[CrossRef](#)]
18. Xiao, J.; Zhou, X. Research on Force-Controlled Polishing System for Robotic Arm Based on Compliance Control. *J. Phys. Conf. Ser.* **2024**, *2764*, 012055. [[CrossRef](#)]
19. González, M.; Rodríguez, A.; Pereira, O.; Celaya, A.; López de Lacalle, L.N.; Esparta, M. Axial-Compliant Tools for Adaptive Chamfering of Sharp-Edges: Characterisation and Modelling. *Eng. Sci. Technol. Int. J.* **2023**, *41*, 101407. [[CrossRef](#)]
20. Krotov, A.S. Application of Robotic Arms to Optimize the Packaging Line of a Paper Mill: Modeling in AnyLogic. In Proceedings of the 2024 Conference of Young Researchers in Electrical and Electronic Engineering (ElCon), Saint Petersburg, Russia, 29–30 January 2024.
21. Auh, E.; Kim, J.; Joo, Y.; Park, J.; Lee, G.; Oh, I.; Pico, N.; Moon, H. Unloading Sequence Planning for Autonomous Robotic Container-Unloading System Using A-Star Search Algorithm. *Eng. Sci. Technol. Int. J.* **2024**, *50*, 101610. [[CrossRef](#)]
22. Alojaiman, B. Technological Modernizations in the Industry 5.0 Era: A Descriptive Analysis and Future Research Directions. *Processes* **2023**, *11*, 1318. [[CrossRef](#)]
23. Raza, F. AI for Predictive Maintenance in Industrial Systems. *Cosm. Bull. Bus. Manag.* **2023**, *2*, 167–183. [[CrossRef](#)]
24. Chen, W.; Men, Y.; Fuster, N.; Osorio, C.; Juan, A.A. Artificial Intelligence in Logistics Optimization with Sustainable Criteria: A Review. *Sustainability* **2024**, *16*, 9145. [[CrossRef](#)]
25. Ettalibi, A.; Elouadi, A.; Mansour, A. AI and Computer Vision-Based Real-Time Quality Control: A Review of Industrial Applications. *Procedia Comput. Sci.* **2024**, *231*, 212–220. [[CrossRef](#)]
26. Taleghani, M. The Role of Artificial Intelligence (AI) in Quality Control of Industrial Products. *Iris J. Econ. Bus. Manag.* **2024**, *2*, 1–2.
27. Károly, A.; Galambos, P.; Kuti, J.; Rudas, I. Deep Learning in Robotics: Survey on Model Structures and Training Strategies. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 266–279. [[CrossRef](#)]
28. Barata, J.; Kayser, I. Industry 5.0 – Past, Present, and Near Future. *Procedia Comput. Sci.* **2023**, *219*, 778–788. [[CrossRef](#)]
29. Mohan, A.; Priyadarshini, R. Gesture Controlled Robot using Accelerometer. *Int. J. Eng. Adv. Technol.* **2020**, *9*, 1241–1245. [[CrossRef](#)]
30. Gomathy, C.K.; Niteesh, G.; Sai Krishna, K. The Gesture Controlled Robot. *Int. Res. J. Eng. Technol. (IRJET)* **2021**, *8*, 1721.
31. Cekova, K.; Koceska, N.; Koceski, S. Gesture Control of a Mobile Robot using Kinect Sensor. In Proceedings of the International Conference on Applied Internet and Information Technologies, Bitola, Macedonia, 3–4 June 2016. [[CrossRef](#)]
32. Rosca, S.D.; Leba, M.; Sibisanu, R.C.; Muntean, E. Gesture Control of a Robotic Head using Kinect. In Proceedings of the 7th International Conference on Mathematics and Computers in Sciences and Industry (MCSI), Athens, Greece, 22–24 August 2022; pp. 101–108. [[CrossRef](#)]
33. Kim, J.-W.; Choi, J.-Y.; Ha, E.-J.; Choi, J.-H. Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. *Appl. Sci.* **2023**, *13*, 2700. [[CrossRef](#)]
34. Aggarwal, L.; Gaur, V.; Verma, P. Design and Implementation of a Wireless Gesture Controlled Robotic Arm with Vision. *Int. J. Comput. Appl. (0975-8887)* **2013**, *79*, 39–43. [[CrossRef](#)]
35. Pradeep, J.; Paul, P.V. Design and Implementation of Gesture Controlled Robotic Arm for Industrial Applications. *Int. J. Adv. Sci. Res. Dev.* **2016**, *3*, 202–209.
36. Megalingam, R.K.; Saboo, N.; Ajithkumar, N.; Unny, S.; Menon, D. Kinect Based Gesture Controlled Robotic Arm: A Research Work at HuT Labs. In Proceedings of the 2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE), Jaipur, India, 20–22 December 2013. [[CrossRef](#)]
37. Zidarić, M. Vizijijski Sustav za Upravljanje Industrijskim Robotom Pokretima Ruke. Master’s Thesis, University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Zagreb, Croatia, 2024.
38. Gasparetto, A.; Scalera, L. A Brief History of Industrial Robotics in the 20th Century. *Adv. Hist. Stud.* **2019**, *8*, 24–35. [[CrossRef](#)]

39. Khajone, S.A.; Mohod, S.W.; Harne, V.M. Implementation of a Wireless Gesture Controlled Robotic Arm. *Int. J. Innov. Res. Comput. Commun. Eng.* **2015**, *3*, 375–379. [[CrossRef](#)]
40. Stergiopoulou, E.; Sgouropoulos, K.; Nikolaou, N.; Papamarkos, N.; Mitianoudis, N. Real-Time Hand Detection in a Complex Background. *Eng. Appl. Artif. Intell.* **2014**, *35*, 54–70. [[CrossRef](#)]
41. Choi, J.-Y.; Ha, E.; Son, M.; Jeon, J.-H.; Kim, J.-W. Human Joint Angle Estimation Using Deep Learning-Based Three-Dimensional Human Pose Estimation for Application in a Real Environment. *Sensors* **2024**, *24*, 3823. [[CrossRef](#)] [[PubMed](#)]
42. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Ubowaja, E.; Hays, M.; Zhang, F.; Chang, C.-L.; Yong, M.G.; Lee, J.; et al. Mediapipe: A Framework for Building Perception Pipelines. *arXiv* **2019**, arXiv:1906.08172.
43. Li, J.; Xu, C.; Chen, Z.; Bian, S.; Yang, L.; Lu, C. HybrIK: A Hybrid Analytical-Neural Inverse Kinematics Solution for 3D Human Pose and Shape Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 3383–3393.
44. Li, W.; Liu, H.; Tang, H.; Wang, P.; Van Gool, L. MHFormer: Multi-Hypothesis Transformer for 3D Human Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13147–13156.
45. Shan, W.; Liu, Z.; Zhang, X.; Wang, Z.; Han, K. Diffusion-Based 3D Human Pose Estimation with Multi-Hypothesis Aggregation. *arXiv* **2023**, arXiv:2303.11579.
46. Mitrovic, K.; Milošević, D. Pose Estimation and Joint Angle Detection Using Mediapipe Machine Learning Solution. In *Applied Artificial Intelligence: Medicine, Biology, Chemistry, Financial, Games, Engineering*; Springer: Cham, Switzerland, 2023; pp. 109–120. [[CrossRef](#)]
47. Kasar, M.; Kavimandan, P.; Suryawanshi, T.; Abbad, S. AI-Based Real-Time Hand Gesture-Controlled Virtual Mouse. *Aust. J. Electr. Electron. Eng.* **2024**, *21*, 258–267. [[CrossRef](#)]
48. Using the Single Camera Calibrator App. Available online: <https://www.mathworks.com/help/vision/ug/using-the-single-camera-calibrator-app.html> (accessed on 30 June 2023).
49. Dorodnicov, S.; Grunnet-Jepsen, A.; Wen Rev, G. Projection, Texture-Mapping, and Occlusion with Intel® RealSense™ Depth Cameras. Available online: <https://dev.intelrealsense.com/docs/projection-texture-mapping-and-occlusion-with-intel-realsense-depth-cameras> (accessed on 4 July 2024).
50. Kavana, K.M.; Suma, N.R.; Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.-L. Recognition of Hand Gestures Using Mediapipe Hands. *Int. Res. J. Mod. Eng. Technol. Sci.* **2022**, *4*, 4149–4156.
51. Sturm, P.; Ramalingam, S.; Tardif, J.-P.; Gasparini, S.; Barreto, J. Camera Models and Fundamental Concepts Used in Geometric Computer Vision. *Found. Trends Comput. Graph. Vis.* **2011**, *6*, 1–183. [[CrossRef](#)]
52. Ahn, M.S.; Chae, H.; Noh, D.; Nam, H.; Hong, D. Analysis and Noise Modeling of the Intel RealSense D435 for Mobile Robots. In Proceedings of the 2019 16th International Conference on Ubiquitous Robots (UR), Jeju, Republic of Korea, 24–27 June 2019. [[CrossRef](#)]
53. González, M.; Rodríguez, A.; López-Saratxaga, U.; Pereira, O.; López de Lacalle, L.N. Adaptive Edge Finishing Process on Distorted Features through Robot-Assisted Computer Vision. *J. Manuf. Syst.* **2024**, *74*, 41–54. [[CrossRef](#)]
54. Qian, N.; Lo, C.-Y. Optimizing Camera Positions for Multi-View 3D Reconstruction. In Proceedings of the 2015 International Conference on 3D Imaging (IC3D), Liege, Belgium, 14–15 December 2015.
55. Li, Z.; Wang, K.; Jia, W.; Chen, H.-C.; Zuo, W.; Meng, D.; Sun, M. Multiview Stereo and Silhouette Fusion via Minimizing Generalized Reprojection Error. *Image Vis. Comput.* **2015**, *33*, 1–14. [[CrossRef](#)]
56. Fang, H.-S.; Li, J.; Tang, H.; Xu, C.; Zhu, H.; Xiu, Y.; Li, Y.-L.; Lu, C. AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 7157–7173. [[CrossRef](#)]
57. Mikhel, S.; Popov, D.; Mamedov, S.; Klimchik, A. Advancement of Robots With Double Encoders for Industrial and Collaborative Applications. In Proceedings of the 2018 23rd Conference of Open Innovations Association (FRUCT), Bologna, Italy, 13–16 November 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.