www.cadanda.com
Computer-Aided Design AND APPLICATIONS

Development of a Low-Cost Vision System for CAD-Based Guidance of Industrial Robots

Jacopo Lettori[1], Roberto Raffaeli[2], Pietro Bilancia[2] and Marcello Pellicciari[2]

[1]InterMech-DIEF, University of Modena and Reggio Emilia, jacopo.lettori@unimore.it
[2]InterMech-DISMI, University of Modena and Reggio Emilia, roberto.raffaeli@unimore.it, pietro.bilancia@unimore.it, marcello.pellicciari@unimore.it

Corresponding author: Jacopo Lettori, jacopo.lettori@unimore.it

Abstract. Visual Servoing (VS) is an approach that integrates vision systems, i.e., cameras, with robotic arms to provide flexibility to operations such as pick and place or machining that require good positioning accuracy. Also, VS can be used in flexible robotic manufacturing to automatically identify and change the required tool for a specific operation. This paper proposes a vision system based on low-cost depth cameras with an eye-in-hand configuration mounted on a KUKA KR210 R2700 prime with 6 degrees of freedom. The alignment of the real target to the nominal one is based on the iterative closest point algorithm. Two case studies have been developed to test the reliability of the system. First, tasks were picked and placed for a small block in different operating conditions. After, the vision system was used to guide the robot in grasping a tool changer. In both cases, the alignment tasks were executed, ensuring sufficient precision.

Keywords: Vision System, Tool Changer, Robot, Pick-and-Place, Computer Vision Algorithm, Robot Guidance.
DOI: https://doi.org/10.14733/cadaps.2025.976-994

1 INTRODUCTION

Nowadays, the ability of vision systems to automate robot operations such as pick and place or machining that require good positioning accuracy is a topic of strong interest both for scientific research and industry. Vision systems integrated with robotic arms [20] can be adopted to check the actual position of components and act accordingly [5]. For example, aircraft parts are analyzed for inspection purposes in [16]. Similarly, it is possible to control the movement of robots by calculating the distance of a given target in real time, thanks to the information received from cameras or 3D sensors.

This approach is called Visual Servoing (VS), and it splits into two types, i.e., Image-Based Visual Servoing (IBVS) and Position-Based Visual Servoing (PBVS) [12]. Moreover, two types of configurations can be distinguished according to the camera location. A first configuration is called eye-in-hand, when the camera is attached to the robot. On the contrary, in the eye-to-hand configuration, the camera is fixed and externally observes the robot working space [3]. The VS is

crucial in automated drilling [31], welding [27], and assembly [22] to increase the positioning accuracy of the manufacturing processes. In this context, in [19], a light plane is used to detect deviations and calculate the tool position according to a given part. Also, the VS technique is often implemented in pick-and-place operations. Indeed, it is possible to identify strategies that use an initial transformation of approximation and then the Iterative Closest Point (ICP) algorithm [18] with a PBVS system. Note that a depth camera is required for this type of application.

In flexible robotic manufacturing cells [26], where several operations are performed, the tool attached to the flange of the robot needs to be changed from time to time. For example, clamps, drillers, and other manufacturing tools are required in production lines [21] and should be interchanged as robot end effectors when needed. However, calibrating the robot with the exact position to find the tool is necessary. In the current industrial practice, the definition and the recording of the robot tool positions involve the manual coupling of the robot flange with the required tool recovered from a storage stand. This procedure is called teach by showing, and it is time-consuming, especially for a 6 degree of freedom manipulator, leading to production downtime.

Furthermore, this process must be repeated when cell reorganization occurs, resulting in additional downtime that corresponds to a reduction of productivity for the company. A vision system to get the correct tool position can be used along with a VS technique in the change procedure, thus reducing operator intervention and time inefficiency. For instance, the tool change procedure based on vision systems has been experimented with in some works in the literature [30]. However, these works only focus on the robot control architecture.

In this context, this paper proposes developing a vision system based on low-cost depth cameras which are spreading in the market. In particular, an eye-in-hand configuration was selected to be applied to a 6-degree-of-freedom anthropomorphic robot, a KUKA KR210 R2700 prime. The alignment of the real target to the nominal one is based on the ICP algorithm [29]. First, pick and place tests were conducted to verify the behavior of a depth camera under different operation conditions, varying the brightness, the distance from the focal point, and the object orientation. The tests were conducted using a small block to evaluate the performance reachable with the developed application. Subsequently, this setup was tested on a more complex case study of industrial interest, that is grasping a tool changer.

2 STATE OF THE ART

This section discusses the topics of vision systems and pose estimation algorithms in depth. In particular, Section 2.1 describes the differences between the two types of VS, while Section 2.2 introduces the algorithms that can be implemented for related applications.

2.1 Vision System for Industrial Applications

Using a vision system to control the positioning of a manipulator, i.e., VS, presents two challenges. Firstly, it involves the identification of the target point for the robot, followed by implementing a control law to guide the robot from its current position to the desired position using the feedback information provided by the vision system. Performing VS control means including information from the vision system integrated with the robot into the control loop in low-level feedback.

As introduced, it is possible to classify the VS control system into two groups. The first group is called IBVS, and the image is acquired by a camera that is compared with the image of the target. A positioning error is calculated to be provided as input to the control law to correct the imposed trajectory [2]. The structure of a generic IBVS-type controller is depicted in Figure 1.

The benefit of IBVS-type controls is their ability to withstand potential calibration errors. Implementing these controls necessitates the computation of the image Jacobian. This matrix captures crucial information regarding the connection between the camera reference system, the object position, and the device velocity in space. Evaluating this matrix is not always straightforward. Moreover, image-based controls in the plane exhibit strong nonlinearity and coupling, which can lead to complications when the manipulator is close to singularity configurations [23].

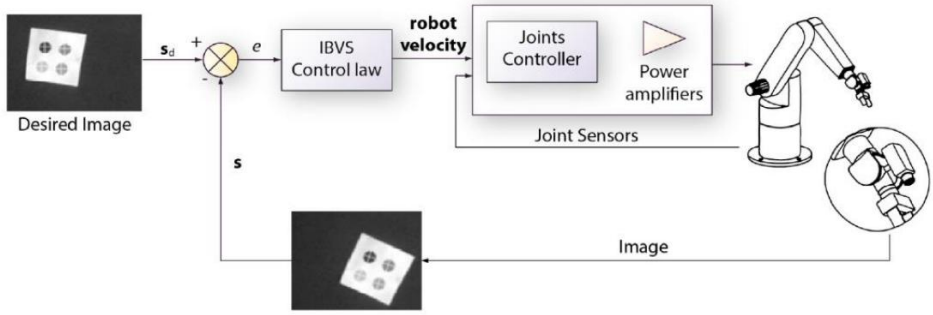

Figure 1: Structure of an IBVS-type controller, taken from [7]. s_d: desired state; s: current state; e: error.

On the contrary, PBVS control implements depth cameras to obtain the pose in 6 dimensions in the space of the target point. In this context, it is possible to calculate the transformation matrix that superposes the target reference system to the one obtained from the camera [6]. The structure of a generic PBVS-type controller is depicted in Figure 2.

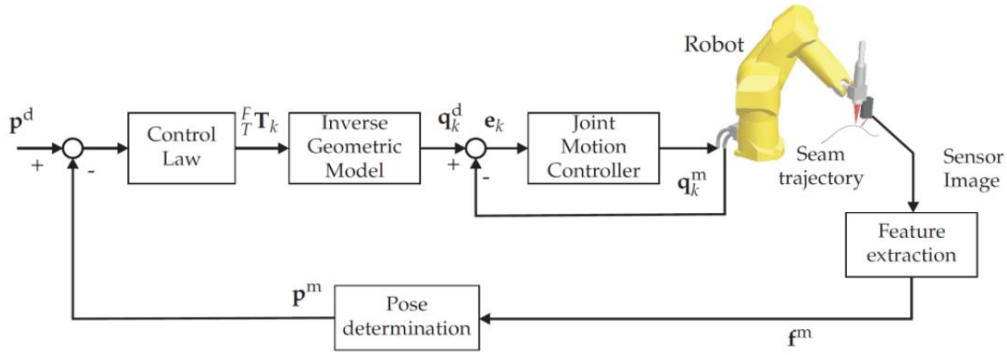

Figure 2: Structure of a PBVS-type controller, taken from [9]. p^d: desired pose; p^m: pose of the object; $^F_T T_k$: image function; q_k^d: reference input for a joint motion controller; q_k^m: actual joint measurements; e_k: tracking error; f^m: feature vector of the image.

Although IBVS systems are less prone to calibration errors, it is possible to define robot tasks with less effort with PBVS systems because there is no need to map position and relative velocity between the camera reference system and object position.

As mentioned in Section 1, it is possible to identify two types of setups for the robot/camera configuration in the field of vision system controls. In the first type, the eye-in-hand configuration, the camera is mounted on the robot as an integral part, as depicted in Figure 3.

In this configuration, it is mandatory to determine the transformation matrix T_e^c that defines the position of the camera reference system O_c concerning the robot end-effector frame O_e. Furthermore, it is important to consider the displacement of the target in the image as the manipulator moves.

In the second type, i.e., the eye-to-hand configuration, the camera is stationary and observes the target externally, as shown in Figure 4.

In this setup, the transformation matrix T_b^c which represents the relationship between the camera frame O_c and the robot base frame O_b remains constant and is calculated only once. On the other hand, the transformation X_b^t from O_c to the robot end-effector frame O_e requires computation at each

iteration by utilizing transformations between various frames. Also, the positioning of the camera needs to be carefully defined to prevent collisions with the robot. Then, some camera types, i.e., stereo cameras, have a limited field of view, requiring a specific distance from the object. These factors contribute to a significant decrease in the accuracy of target position estimation [17].

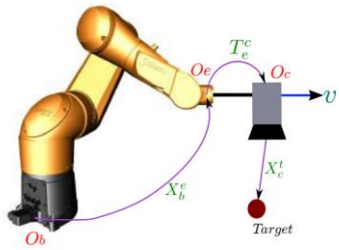

Figure 3: Eye-in-hand system, adapted from [3]. O_b: robot base frame; O_e: end-effector frame; O_c: camera frame; X_b^e: pose of O_e with respect to O_b; X_c^t: pose of the target relative to O_c; T_e^c: transformation matrix between O_e and O_c; v: motion.

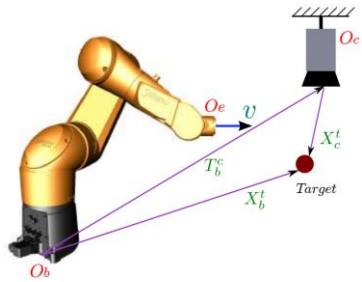

Figure 4: Eye-to-hand system, adapted from [3]. O_b: robot base frame; O_e: end-effector frame; O_c: camera frame; X_b^t: pose of the target with respect to O_b; X_c^t: pose of the target relative to O_c; T_b^c: transformation matrix between O_b and O_c; v: motion.

For these reasons, an eye-in-hand configuration with PBVS was chosen, ensuring system flexibility and accuracy. The main pose estimation algorithms used in the VS are discussed in the next section.

2.2 Algorithms for the Pose Estimation

Different types of algorithms are employed to enable the recognition of an object's position within an environment and facilitate the movement of the manipulator toward that position. A major group is based on Artificial Intelligence [14,13], as depicted in Figure 5.

Implementing neural network-based algorithms for estimating target positions encounters difficulties when dealing with partial point clouds. When considering the acquisition of data from a single camera, the mapped environment in the point cloud is susceptible to occlusion-related issues, leading to incomplete environmental mapping. Furthermore, employing artificial intelligence methods entails the possibility of not successfully detecting the target object or, in the worst-case scenario, making an incorrect target identification, resulting in unpredictable system behavior.

A second type of algorithm utilizes the ICP procedure [29,15]. These algorithms demonstrate greater reliability than those reliant on artificial intelligence, enhancing the precision and accuracy of the estimated pose. Nonetheless, ICP tends to find a local minimum, hindering algorithm convergence, particularly when an initial transformation that adequately approximates the position of the target cloud in its actual environment is not provided [11].

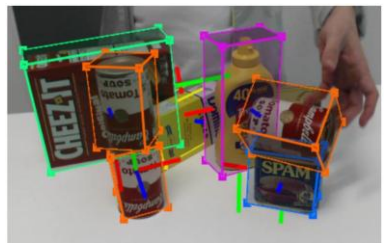

Figure 5: Example of the use of neural networks for recognition and distance estimation of objects, taken from [28].

The conventional ICP algorithm involves an initial transformation typically specified by the user when calling the function within the code [1]. Subsequently, it employs the closest point algorithm, also known as the nearest neighbor, to define the nearest point of the target cloud for each point in the source cloud [8]. The transformation associated with the previously defined closest points is then executed. Finally, the executed transformation Mean Squared Error (MSE) is evaluated and compared with that of the previous transformation to determine the algorithm correct evolutionary direction. This process is repeated until the MSE converges or for a user-defined number of iterations. The working principle of an ICP algorithm is illustrated in Figure 6.

In general, the alignment of the point cloud with the actual target can be performed with three different strategies, i.e., point-to-point, point-to-plane, and plane-to-plane. In particular, it is observed that strategies of the point-to-plane type perform better than the classical point-to-point approach on different datasets [24]. However, the first approximation to allow the point-to-plane algorithm to run properly must be defined accurately concerning the point-to-point to ensure convergence based on preliminary experiments. As a result, more setup time is required for the initial system. Additionally, hybrid applications combine neural networks for object recognition and the filtering of point clouds, followed by the application of the ICP algorithm to enhance position estimation accuracy [25]. In both scenarios, the objective is to align two-point clouds: one representing the target object and the other depicting the working environment. A traditional point-to-point approach was chosen and employed in this work.

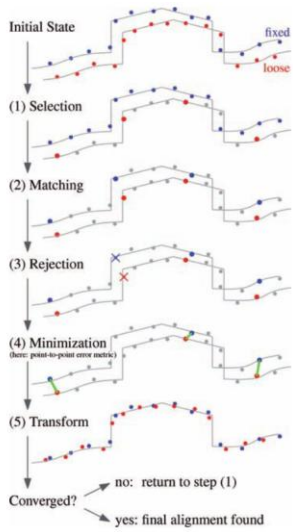

Figure 6: Main steps of the ICP algorithms, taken from [8].

3 APPROACH FOR A PICK AND PLACE OPERATION

The following Figure 7 depicts the main phases of the proposed work.

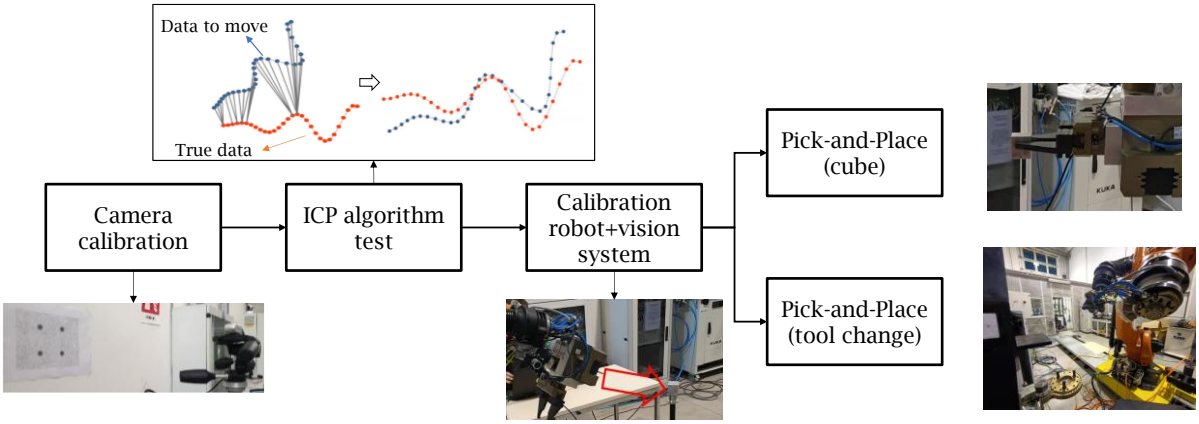

Figure 7: Main phases of the proposed approach.

The first case study is related to the pick and place operation of a solid block (50x50x50 mm). The selected camera is the IntelRealSense D435i. First, the camera is calibrated thanks to specific experiments. Then, tests are carried out for tuning the implemented ICP algorithm in the complete system. Finally, the approach was implemented in a tool change operation in an industrial environment. In this case a different camera has been used, i.e., a ZED 2i depth camera, which is characterized by improved accuracy.

3.1 Camera Calibration

The first calibration involves adjusting the extrinsic and intrinsic parameters of the depth camera to minimize the difference between the measured distance value and the real distance. The considered extrinsic parameters are:

1. *RotationLeftRight:* the rotation matrix between the reference system of the right camera and that of the left camera of the device;
2. *TranslationLeftRight*: the translation between the reference system of the right camera to that of the left camera;
3. *RotationLeftRGB*: the rotation matrix of the RGB module to the left camera;
4. *TranslationLeftRGB*: the translation between the RGB module and the camera on the left.

Then, the following intrinsic camera parameters have been identified:

1. Focal point;
2. Principal point;
3. Distortion, described through Brown's model.

The Depth Quality Tool provided by Intel for the IntelRealSense D435i camera was used to calibrate the camera. The selected parameters to evaluate the image acquisition are:

1. *Plane Fit RMS Error*: the RMS error expressed as a percentage concerning the mean plane (plane fit) passing through the acquired points;
2. *Subpixel RMS Error*: it represents the RMS error expressed in pixel units to the mean plane passing through the acquired points;
3. *Fill-Rate*: the percentage of pixels with a depth value considered correct;

4. *Z Accuracy*: the percentage disparity between the distance of the plane fit and the actual distance (ground truth measured with a laser tracker).

So, tests were conducted at different target distances to verify the behavior of the camera (Figure 8). As a result, it is noted that the Z accuracy error tends to increase with the distance from the target. Then, a recalibration process (Tare Calibration [10]) was performed using a FARO Vantage E laser tracker using a known target, as shown in Figure 9a. The target is placed at a distance equal to 700mm, which is considered plausible for pick-and-place operations. The relative plans of the camera and target were reconstructed by acquiring significant points by the FARO laser tracker (Figure 9b). The Z accuracy error passes from more than 3% to below 0.5% thanks to the recalibration process.

It is important to note that the *High Accuracy* preset for *Object Scanning* provided by Intel was used. The following parameters were then modified to improve the performance of the camera for the specific application:

- *Resolution*: The resolution has been increased to 848x480, as recommended by the manufacturer. This resolution strikes the best balance between image quality and the minimum observable distance required for the device's correct operation.
- *Frame rate*: The frame rate has been reduced from 30 to 6 to allow the device more time for computational processing per individual image acquisition. This change directly impacts the quality and accuracy of the generated data.
- *Infrared*: The infrared functionality has been kept active. While this helps the sensor to visualize the working environment as mentioned earlier, activating the projector introduces additional noise into the generated point cloud.
- *Sensor Sampling Steps*: The sampling steps of the sensor have been reduced from 0.001mm to 0.0001mm. Since the system can sample at 16 bits, this reduction results in a maximum measurable distance decrease from 16m to 6m.
- *Point Cloud Filtering*: Filtering the point cloud generated at 4m reduces the software's computational workload, as further explained in the following.

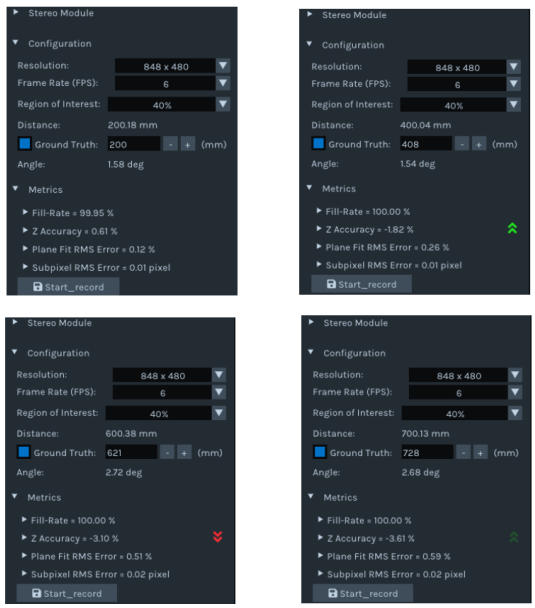

Figure 8: Images of the settings of the tool adopted to evaluate Camera Z accuracies at different distances.

3.2 Preliminary Accuracy Tests with the Camera

An experimental campaign was developed to evaluate the camera performance, considering a cube as reference geometry. The camera was intentionally tilted approximately 20° downwards to enhance the performance of the matching algorithm. Specifically, the ICP algorithm requires overlapping the point cloud and the reference CAD model in different spatial directions. This ensures that there are no remaining degrees of freedom between the two point clouds, and the relative position of one for the other is uniquely determined.

Thus, the cube was placed on graph paper at known positions, ensuring that at least three faces were visible from the camera, as depicted in Figure 10. The sampling points were strategically selected to cover as much of the camera's field of view within the nominal 700mm area of interest. Each sampled point was positioned 50mm from its adjacent point along the X and Y axes, as marked on the plan.

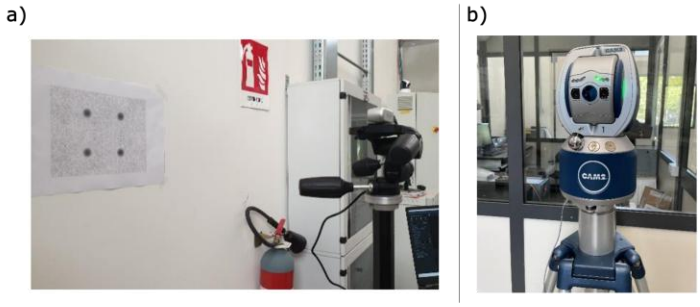

Figure 9: a) Calibration setup; b) FARO Vantage E laser tracker.

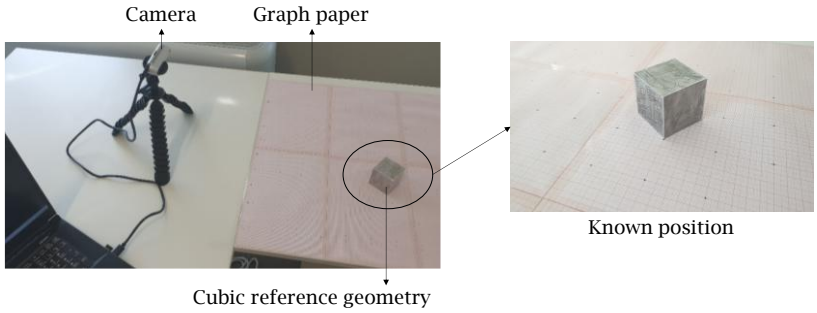

Figure 10: Experimental setup.

In this context, a point-to-point ICP algorithm provided by the Open3D Python library was used and tested to superimpose the obtained point cloud with the actual 3D geometry. Also, the complete system that combines the ICP algorithm and the camera was tested, i.e. the repeatability and the approximation error. Considering the repeatability test, 100 acquisitions were conducted for each position, from which one initial transformation was selected for the subsequent 100 executions of the algorithm between the unique source cloud and the respective target clouds.

Figure 11 illustrates that increasing the distance between the target and the camera leads to a deterioration in the system's repeatability. Specifically, it is evident that the maximum deviation from the average approximation value appears to increase in absolute terms in the initial two graphs.

Additionally, the quality of the approximation reduces as the distance increases. The fitness decreases from 98% for the closest positions to 75% for the most distant ones, while the initial RMSE error of 1.5 mm grows to 2.5 mm in the latest runs. The standard deviation graph in Figure 11 also confirms this trend. The reported Fitness value represents the percentage of points of the source cloud that are correctly coupled with the target cloud. This means that the fitness parameter is directly connected with the level of accuracy that is intended to be achieved and settable with the

parameter named Approximation in the ICP algorithm. The Root Mean Square Error (RMSE) provides a measure of the mean squared distance from each point in the source to its closest point in the target.

Then, a study was conducted to analyze the trend of relative measurement errors between two targets based on their distance. The reference measurement with the highest degree of accuracy, representing the central and most advanced test plan position, served as the basis for comparison. Figure 12 depicts the results of comparing estimated distances and real distances. Absolute (i.e. *displacement* graph) and the relative error (i.e., *percentage_displacement* graph) are also reported. The measurement error increases as the distance between the two targets grows, particularly when one target is positioned farther from the optical sensor than the other. This observation aligns with the findings from previous tests, where less precise results for distant targets had a cascading impact on relative measurements involving other targets.

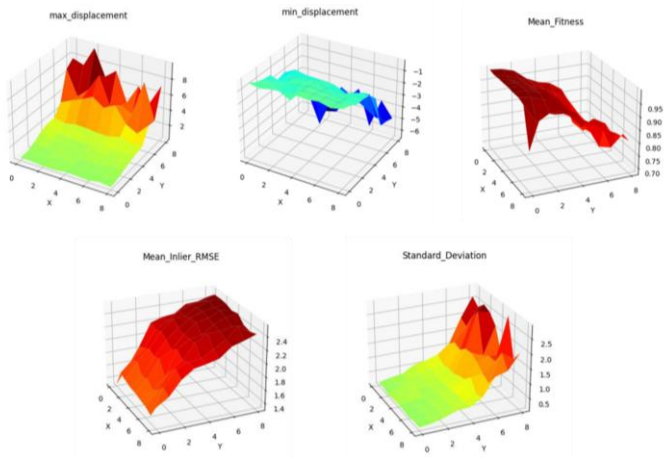

Figure 11: Results obtained for the test on the repeatability of the overall camera-algorithm system. X and Y axes report the indices of the positions. The values in the Z axes are millimeters.

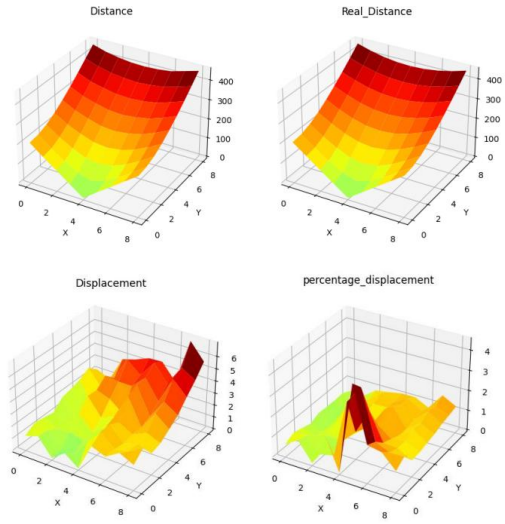

Figure 12: Results obtained for the test on the relative error of the overall camera algorithm system. X and Y axes report the indices of the positions. The values in the Z axes are millimeters for the first three graphs, while in the last one, a percentage is reported.

The detected error in measurement results is relatively predictable and varies consistently, except under singular conditions possibly linked to an initial suboptimal transformation. While incorporating relative measurements enhances overall system performance, it necessitates specific calibration procedures and a comprehensive understanding of a target location reference within the working environment to ensure high reliability. However, replicating these conditions proves challenging in an industrial setting. Consequently, in developing subsequent applications, reliance will be placed on individual measurements captured by the camera rather than estimations of relative positions to known targets.

3.3 Application to a Pick and Place Operation

A support has been realized to mount the camera on the robot and ensure its proper localization. Additionally, the system has been designed to incorporate a support structure for the target utilized by the FARO laser tracker, i.e., Spherically Mounted Retroreflector (SMR), for accurate positioning reference. This allows for establishing a known distance between the reference system constructed during the calibration process on the SMR and that of the camera. The complete system is depicted in Figure 13.

Figure 13: Complete system: robot, camera and SMR.

After the characterization and calibration procedures described in the previous sections have been accomplished, the process starts with identifying the primary directions of the system reference coordinate system. This involves a procedure based on elementary rotations of single joints of the robot, tracking the movement of the SMR with the laser tracker and fitting circumference through the obtained trajectories (Figure 14). This procedure, described in detail in a previous work [4], allows the determination of the transformation linking the reference system of the laser tracker to the robot flange reference system. Leveraging the known dimensions of the camera support from its CAD model, it becomes possible to ascertain the transformation matrix of the eye of the camera both in the robot base reference system and in the laser tracker reference system.

Figure 14: Assembly of the SMR for calibration and identification of the two planes for the axis calculation.

The process is governed by a Python application that interacts with the devices, i.e. robots and the camera, and elaborates the acquired geometries. The process is not intended to be real-time: motion instructions are sent to the robot as soon as the acquisition of the point clouds and their elaboration is completed. It takes a few seconds, and the robot waits until the computation ends.

The main tasks of the experiments were conducted as follows:

- Robot positioning: During this phase, a view of the RGB image and depth data are provided to the operator so that he can move the robot to a specific Cartesian position relative to its base. This ensures that the camera frames the target.
- The aluminum cube was positioned on a support (Figure 15a).
- Selection of the first transformation: the operator chooses an initial approximate transformation necessary for the correct functioning of the ICP algorithm. Such initial transformation is obtained manually by jogging the robot using the teach pendant to a position sufficiently close to the grasping one and saving the orientation matrix.
- Robot approach: the first transformation is utilized to bring the robot to a starting position (Figure 15b top). This has been accomplished thanks to the Python application with streams commands to the robot controller activating its motion. Checks have been implemented to verify whether the point-to-point ICP algorithm can successfully align the point clouds. The algorithm stops if it undergoes five successive approaches without detecting an improvement in the result. This indicates poor precision in the initial transformation provided, prompting the application to request a more precise one with the new data obtained.

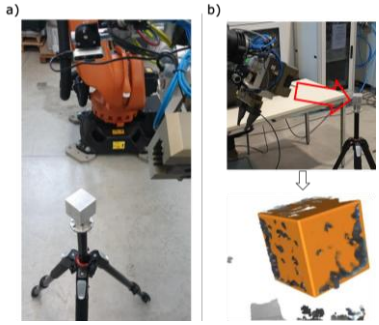

Figure 15: a) Setup for positioning the target in the workspace; b) Top: first movement provided manually by the operator to make the target visible to the camera; Bottom: scanning result obtained from the Refinement operation.

- Subsequently, the application requests ten acquisitions from the camera and executes the ICP algorithm to mediate the results obtained by the algorithm, enabling a better estimation of the target pose (Figure 15b bottom)
- Grasping: the robot is moved to position the gripper fingers appropriately for closing and grasping the cube, thereby completing the requested operation (Figure 16).

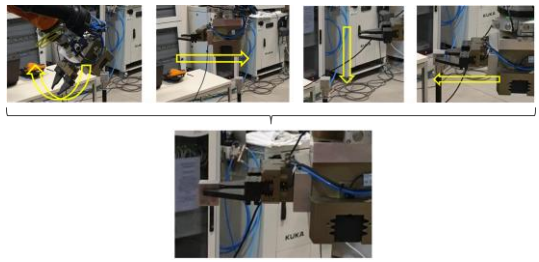

Figure 16: Phases of pick and place operations. From left to right: Gripper reorientation; Translations to reach the cube; Pick of the target (bottom).

4 APPLICATION TO A TOOL CHANGER

This section presents an industrial application. The considered robotic system was equipped with a tool changer composed of a master part that is mounted on the robot flange (Figure 14 left) and a

slave part that is fixed on the tool to be grasped (Figure 17). Such a solution allows more than one slave applied to different tools to be picked by the same master counterpart so that the robot can change the tool according to the required operation. In this context, a different camera was implemented (ZED 2i depth camera). This choice is motivated by better performances requested by the tested industrial application context. The ZED 2i camera, which recently appeared on the market, guarantees better characteristics than the previously adopted one. Thus, a new identification of optimal camera parameters was needed to ensure proper acquisition. A similar procedure to Sections 3.1 and 3.2 has been done to identify the camera reference for the robot flange. Then, experimental tests were performed to verify the system's reliability.

4.1 Preliminary Tests

ZED Depth Viewer is the software provided by the manufacturer to set up and tune the camera. The available parameters were analyzed to determine the optimal values for acquiring quality point clouds. The parameters in the *Camera* section were maintained at their default values, as altering them did not consistently yield improved results. Different resolutions, ranging from HD2K to VGA, were considered. HD2K resolution, allowing for 15 FPS image acquisition, was deemed sufficient for the application, and it was therefore selected.

Furthermore, the parameters in the *Processing* section were assessed. These parameters significantly impact the camera's performance, particularly its depth accuracy. The ULTRA and NEURAL depth modes were identified as the best for generating point clouds. ULTRA exhibited superior depth accuracy, especially at close distances, while NEURAL performed better for objects at greater distances. The NEURAL mode provided a more accurate depth estimate for objects positioned farther away, whereas the ULTRA mode excelled at shorter distances. Adjusting the *Depth Min* and *Depth Max* parameters allowed for filtering unwanted objects near and far from the camera, respectively, based on their depth.

Figure 17: Left: acquisition test of the tool changer slave; Right: acquisition example carried out in NEURAL mode at almost 1 m distance along Z.

Then, an investigation was carried out to determine the influence of the depth stabilization parameter on acquisition quality. The depth stabilization is a filtering feature provided by the camera manufacturer. It is based on time-depended averaging, and it is expressed by a number ranging from 0 to 100. The ICP algorithm was utilized to assess the validity of each acquisition by observing the alignment results. This involved running the ICP algorithm with varying depth stabilization parameters for each point cloud while keeping all other parameters constant.

Depth stabilization	*Fitness*	*RMSE*	*Fitness/RMSE*
100	0.364	2.740	0.133
99	0.355	2.840	0.125
98	0.377	2.780	0.136
97	0.393	2.800	0.140
96	0.360	2.700	0.133

95	0.370	2.700	0.137
94	0.364	2.810	0.130
93	0.365	2.710	0.135
92	0.377	2.730	0.138
91	0.366	2.690	0.136
90	0.365	2.810	0.130
89	0.346	2.730	0.127
88	0.369	2.730	0.135
87	0.363	2.800	0.130
86	0.357	2.790	0.128
85	0.370	2.790	0.133
84	0.368	2.700	0.136
83	0.360	2.790	0.129
82	0.362	2.740	0.132
81	0.369	2.770	0.133

Table 1: Fitness of point clouds matching varying the depth stabilization parameter.

The source cloud used for the experiment was obtained from a high-density mesh model of the tool changer derived from its CAD model. In each execution, the acquired point cloud from the camera served as the target cloud, with the depth stabilization parameter being varied. The initial transformation was perturbed to simulate different approaching conditions imposing random rotations to the source point cloud. The executed tests have shown that the depth stabilization parameter does not yield significant variations if it is set within the range of 80 to 100, as depicted in Table 1.

Similarly, a sensitivity analysis was conducted to evaluate the influence of the texture confidence parameter on the quality of acquisitions. Such parameter varies from 0 to 100 and sets a level to discard points from uniform regions of the acquired image, which lead to poor depth estimation. The ICP algorithm was run with multiple point clouds, keeping all other parameters and acquisition positions constant while varying only the texture confidence. The findings from this experiment indicated that the optimal texture confidence value is 100, as reported in Table 2.

Texture confidence	*Fitness*	*RMSE*	*Fitness/RMSE*
100	0.394	2.570	0.153
97	0.320	2.870	0.111
94	0.318	2.890	0.110
91	0.369	2.660	0.139
88	0.290	2.960	0.098
85	0.285	2.990	0.095
82	0.276	3.010	0.092
79	0.274	3.060	0.090
76	0.311	2.790	0.111
73	0.250	3.120	0.080
70	0.220	3.180	0.069
67	0.270	2.930	0.092
64	0.210	3.200	0.066

Table 2: Fitness of point clouds matching varying the texture confidence parameter.

Further tests restricted to the range from 98 to 100 of the texture confidence parameter confirmed that the optimal value for the specific test is 100, as shown in Figure 18.

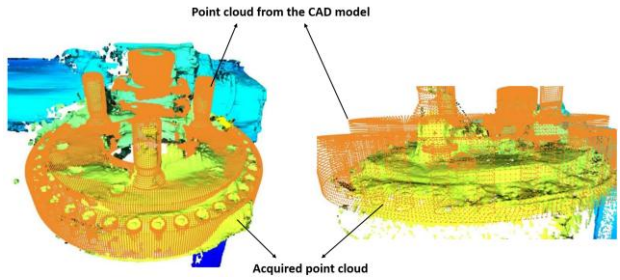

Figure 18: Texture confidence parameter assessment. Left: texture confidence equal to 100; Right: texture confidence equal to 97.

Finally, a sensitivity analysis was also conducted to determine an appropriate confidence value. The confidence parameter sets a level to remove points from the borders of objects in the acquisition, i.e. the regions with significant depth change. This thorough analysis underlines the significance of depth stabilization and texture confidence parameters in influencing the quality of acquisitions, with the study offering valuable insights into their optimal values for high-quality results. As can be noted from Table 3, the best value of confidence turned out to be 100.

Confidence	*Fitness*	*RMSE*	*Fitness/RMSE*
100	0.410	2.570	0.160
97	0.348	2.790	0.125
94	0.366	2.640	0.139
91	0.296	2.870	0.103
88	0.364	2.640	0.138
85	0.290	2.870	0.101
82	0.295	2.870	0.103
79	0.292	2.900	0.101

Table 3: Fitness/RMSE ratio as the confidence parameter varies.

4.2 Process to Grasp the Tool Changer

The final experimental tests are conducted to assess the complete system composed of the camera mounted on the robot in grasping a tool changer. In an industrial setting, the robot cell geometry is commonly predetermined and defined by a virtual model, i.e., its CAD model. The knowledge of this model, in particular the location of the robot and of the tool changer let the initial transform for the acquisition of the scene to be determined. Subsequently, the ICP algorithm is employed to correct the transformation according to the discrepancies between the physical environment and the virtual CAD model.

4.2.1 Experimental setup

A suitable support structure was designed to hold the camera and the SMRs. The support is designed to accommodate three spherical SMRs for the FARO laser tracker, as depicted in Figure 19.

Given the objective of the devised algorithm of aligning the camera with the slave tool changer, it is necessary to combine this acquisition with a fixed transformation that brings the camera reference system to the robot flange to obtain a combined transformation that aligns the master tool changer with the slave one, as shown in Figure 20.

To this aim, the three SMRs enable the accurate measurement of both the orientation of the camera and the robot flange. Specifically, a plane is defined by the three points obtained by housing an SMR in each of the seats located with precision using the laser tracker. Furthermore, it is possible to identify a plane lying on the robot flnger as mentioned in a previous section.

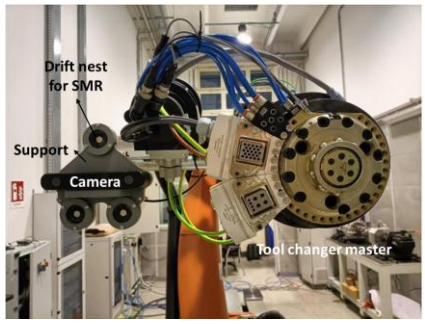

Figure 19: KUKA robot equipped with the camera and the SMR seats.

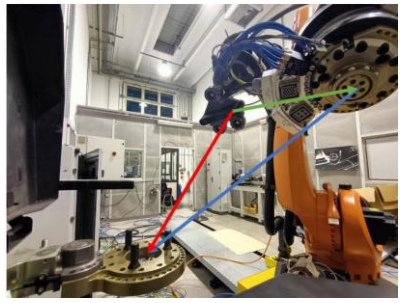

Figure 20: Tool changer application. Red vector: alignment transformation that is achieved by ICP. Blue vector: desired transformation. Green vector: fixed transformation to be combined with the red one.

4.2.2 Development of the software application

A Python application has been developed to control the system which comprises three main phases:

- *Acquisition of Point Cloud*: The first part involves capturing a point cloud of the slave tool changer using the depth camera. The acquisition process utilizes parameters optimized through experiments conducted in Section 4.1.
- *Filtering of Irrelevant Objects*: The second part focuses on filtering out irrelevant objects from the previously acquired point cloud, leaving only the tool changer slave to enhance the results in the subsequent processing.
- *ICP Alignment*: The ICP algorithm is used to align the source point cloud with the point cloud acquired using the ZED 2i stereo camera.

As a first step, the camera acquires a point cloud of the slave tool changer. Regarding the filtering stage, points too far from the tool changer bounding box are discarded, removing details that are not useful for the application. To this aim, the bounding box of the tool changer is computed out of the nominal CAD model relying on limited deviations in the positioning of the tool changer from its nominal position. Therefore, the process is automatic and relies on the knowledge of the approximate position of the tool changer from the CAD model of the cell.

4.2.3 Results of the two-stage ICP alignment

Given the noise in the depth camera acquisition and its limited accuracy, a specific strategy to increase the performance of the ICP algorithm has been developed. In particular, a two-step approach has been devised: the output transformation from the first run is utilized as the initial input transformation in the second run. In the first run, a high approximation value is employed, enabling

the two clouds to come into closer proximity than would not be possible with a lower value. In the second step a lower approximation value is used that enables a more precise result than a single ICP phase.

Tests have been executed varying the number of iterations in the first stage as well in the second. In Table 4, some results are reported for the tool changer application experimented in the laboratory. The combination of 25 iterations for the first step and 80 for the second one yields an optimal result with minimal calculation times, thus representing a recommendable compromise.

N° of Iteration (first ICP)	*N° of Iteration (second ICP)*	*Fitness*	*RMSE*	*Fitness/RMSE*
4000	400	0.494	2.520	0.196
2000	400	0.494	2.520	0.196
2000	0	0.401	2.900	0.138
10000	0	0.401	2.900	0.138
100	10	0.439	2.810	0.156
100	80	0.494	2.520	0.196
100	25	0.470	2.700	0.174
30	30	0.480	2.650	0.181
25	40	0.491	2.640	0.186
25	50	0.500	2.580	0.194
25	80	0.495	2.510	0.197
0	200	0.288	2.840	0.101
0	1000	0.440	2.770	0.159

Table 4: Results of the two-step ICP.

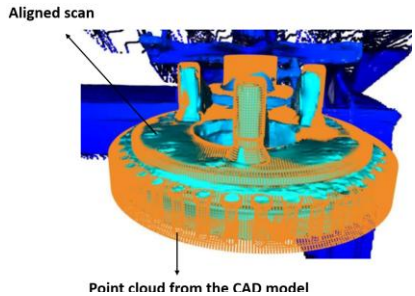

Figure 21: Acquired point cloud aligned with the CAD extracted one.

The first phase approximates, while the second phase refines the transformation. The experiment indicates that performing the ICP in two phases with two appropriately selected approximation values is more effective, rather than conducting it in a single phase. Even though reached results are not always the best, a good approximation is reached with a limited total number of iterations. An example of final point clouds alignment is depicted in Figure 21.

5 CONCLUSIONS AND FUTURE WORKS

The goal of the presented work was the assessment of low-cost depth vision camera to compensate positioning errors in pick-and-place operations with industrial anthropomorphic robots. As a first step, an Intel RealSense D435i depth camera was selected for an initial problem exploration. Afterward, the procedure was applied to an industrial scenario involving coupling the two halves of a tool changer and using a ZED 2i depth camera. In this second case, a two-step ICP process has

been devised. Both the cameras are easily available for less than 500 euros. The activity with the two different cameras allowed a better assessment of the compatibility of the highly smoothed point clouds resulting from the acquisitions with the requirements of the specific application context.

Despite the low accuracy of the scans of the adopted cameras, the results in robot positioning are promising, even if a considerable level of error remains. In both experiments, the alignment tasks were executed, ensuring a sufficient degree of precision. As reported in Table 4, the alignment is achieved with a fitness lower than 0.5 mm and this value is compatible with the typical mechanical allowance of a tool changer coupling.

In future work, substituting the cameras with more advanced 3D scanning systems can considerably improve accuracy. In addition, the ICP algorithm can be further tuned to reach a higher reliability level and avoid falling into local minima solutions, leading to wrong orientations and potential collision. In particular, the initial alignment could be solved by applying methods for the estimation of object poses using affine invariant 3D descriptors. Finally, further applications would ensure a more extended validation of the proposed system.

Jacopo Lettori, https://orcid.org/0000-0002-8523-8469
Roberto Raffaeli, https://orcid.org/0000-0003-0301-454X
Pietro Bilancia, https://orcid.org/0000-0002-4931-1745
Marcello Pellicciari, https://orcid.org/0000-0003-2578-4123

REFERENCES

[1] Besl, P. J.; McKay, N. D.: A Method for Registration of 3-D Shapes. In *Sensor fusion IV:* control paradigms and data structures; Spie; Vol. 1611, pp 586–606. https://doi.org/10.1109/34.121791.
[2] Bourquardez, O.; Mahony, R.; Guenard, N.; Chaumette, F.; Hamel, T.; Eck, L.: Image-Based Visual Servo Control of the Translation Kinematics of a Quadrotor Aerial Vehicle, IEEE Transactions on Robotics, 25(3), 2009, 743–749.
[3] Cai, C.: 6d Visual Servoing for Industrial Manipulators Applied to Human-Robot Interaction Scenarios.
[4] Ferrarini, S.; Bilancia, P.; Raffaeli, R.; Peruzzini, M.; Pellicciari, M.: A Method for the Assessment and Compensation of Positioning Errors in Industrial Robots, Robotics and Computer-Integrated Manufacturing, 85, 2024, 102622. https://doi.org/10.1016/j.rcim.2023.102622.
[5] Franceschini, F.; Galetto, M.; Maisano, D.; Mastrogiacomo, L.: Large-Scale Dimensional Metrology (LSDM): From Tapes and Theodolites to Multi-Sensor Systems, International journal of precision engineering and manufacturing, 15, 2014, 1739–1758. https://doi.org/10.1007/s12541-014-0527-2.
[6] Gans, N. R.; Dani, A. P.; Dixon, W. E.: Visual Servoing to an Arbitrary Pose with Respect to an Object Given a Single Known Length. In 2008 American Control Conference; IEEE; pp 1261–1267. https://doi.org/10.1109/ACC.2008.4586666.
[7] Garcia, G. J.; Pomares, J.; Torres, F.; Gil, P.: Event-Based Visual Servoing with Features' Prediction. In ROBOT2013: First Iberian Robotics Conference: Advances in Robotics, Vol. 1; Springer; pp 679–691. https://doi.org/10.1007/978-3-319-03413-3_50.
[8] Glira, P.; Pfeifer, N.; Briese, C.; Ressl, C.: A Correspondence Framework for ALS Strip Adjustments Based on Variants of the ICP Algorithm, Photogrammetrie, Fernerkundung, Geoinformation, No. 4, 2015, 275–289. https://doi.org/10.1127/pfg/2015/0270.
[9] de Graaf, M. W.: Sensor-Guided Robotic Laser Welding.
[10] Grunnet-Jepsen, A.; Sweetser, J.; Khuong, T.; Dorodnicov, S.; Tong, D.; Mulla, O.: Intel® RealSense™ Self-Calibration for D400 Series Depth Cameras, California, USA, Intel, White Paper, 2021.
[11] Hu, H.; Gu, W.; Yang, X.; Zhang, N.; Lou, Y.: Fast 6D Object Pose Estimation of Shell Parts for Robotic Assembly, The International Journal of Advanced Manufacturing Technology, 2022,

1–14. https://doi.org/10.1007/s00170-021-07960-0.
[12] Kermorgant, O.; Chaumette, F.: Combining IBVS and PBVS to Ensure the Visibility Constraint. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems; IEEE; pp 2849–2854. https://doi.org/10.1109/IROS.2011.6094589.
[13] Li, C.-H. G.; Chang, Y.-M.: Automated Visual Positioning and Precision Placement of a Workpiece Using Deep Learning, The International Journal of Advanced Manufacturing Technology, 104(9), 2019, 4527–4538. https://doi.org/10.1007/s00170-019-04293-x.
[14] Lin, C.-M.; Tsai, C.-Y.; Lai, Y.-C.; Li, S.-A.; Wong, C.-C.: Visual Object Recognition and Pose Estimation Based on a Deep Semantic Segmentation Network, IEEE sensors journal, 18(22), 2018, 9370–9381. https://doi.org/10.1109/JSEN.2018.2870957.
[15] Lin, H.-Y.; Liang, S.-C.; Chen, Y.-K.: Robotic Grasping with Multi-View Image Acquisition and Model-Based Pose Estimation, IEEE Sensors Journal, 21(10), 2020, 11870–11878. https://doi.org/10.1109/JSEN.2020.3030791.
[16] Liu, C.; Li, Y.; Hao, X.: An Adaptive Machining Approach Based on In-Process Inspection of Interim Machining States for Large-Scaled and Thin-Walled Complex Parts, The International Journal of Advanced Manufacturing Technology, 90, 2017, 3119–3128. https://doi.org/10.1007/s00170-016-9647-4.
[17] Luo, G.-Y.; Cheng, M.-Y.; Chiang, C.-L.: Vision-Based 3-D Object Pick-and-Place Tasks of Industrial Manipulator. In 2017 International Automatic Control Conference (CACS); IEEE; pp 1–7. https://doi.org/10.1109/CACS.2017.8284250.
[18] Luo, R. C.; Kuo, C.-W.; Chung, Y.-T.: Model-Based 3D Object Recognition and Fetching by a 7-DoF Robot with Online Obstacle Avoidance for Factory Automation. In 2015 IEEE International Conference on Robotics and Automation (ICRA); IEEE; pp 2647–2652. https://doi.org/10.1109/ICRA.2015.7139556.
[19] Mahapatra, P. K.; Thareja, R.; Kaur, M.; Kumar, A.: A Machine Vision System for Tool Positioning and Its Verification, Measurement, and Control, 48(8), 2015, 249–260. https://doi.org/10.1177/0020294015602499.
[20] Mineo, C.; Pierce, S. G.; Nicholson, P. I.; Cooper, I.: Robotic Path Planning for Non-Destructive Testing–A Custom MATLAB Toolbox Approach, Robotics and Computer-Integrated Manufacturing, 37, 2016, 1–12. https://doi.org/10.1016/j.rcim.2015.05.003.
[21] Mourtzis, D.; Angelopoulos, J.; Papadokostakis, M.; Panopoulos, N.: Design for 3D Printing of a Robotic Arm Tool Changer under the Framework of Industry 5.0, Procedia CIRP, 115, 2022, 178–183. https://doi.org/10.1016/j.procir.2022.10.070.
[22] Nelson, B. J.; Papanikolopoulos, N. P.; Khosla, P. K.: Robotic Visual Servoing and Robotic Assembly Tasks, IEEE Robotics & Automation Magazine, 3(2), 1996, 23–31. https://doi.org/10.1109/100.511777.
[23] Palmieri, G.; Palpacelli, M.; Battistelli, M.; Callegari, M.: A Comparison between Position-Based and Image-Based Dynamic Visual Servoings in the Control of a Translating Parallel Manipulator, Journal of Robotics, 2012, 2012. https://doi.org/10.1155/2012/103954.
[24] Rusinkiewicz, S.; Levoy, M.: Efficient Variants of the ICP Algorithm. In Proceedings third international conference on 3-D digital imaging and modeling; IEEE; pp 145–152. https://doi.org/10.1109/IM.2001.924423.
[25] Song, K.-T.; Chang, Y.-H.; Chen, J.-H.: 3D Vision for Object Grasp and Obstacle Avoidance of a Collaborative Robot. In 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM); IEEE; pp 254–258. https://doi.org/10.1109/AIM.2019.8868694.
[26] Thamilarasi, P.; Ragunathan, S.: Study on the Effect of Machine Vision in Real Time Application—Robot Welding. In 2010 IEEE International Conference on Computational Intelligence and Computing Research; IEEE; pp 1–5. https://doi.org/10.1109/ICCIC.2010.5705801.
[27] Tingelstad, L.; Capellan, A.; Thomessen, T.; Lien, T. K.: Multi-Robot Assembly of High-Performance Aerospace Components, IFAC Proceedings Volumes, 45(22), 2012, 670–675. https://doi.org/10.3182/20120905-3-HR-2030.00117.
[28] Tremblay, J.; To, T.; Sundaralingam, B.; Xiang, Y.; Fox, D.; Birchfield, S.: Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects, arXiv preprint

arXiv:1809.10790, 2018.
[29] Wang, Z.; Fan, J.; Jing, F.; Liu, Z.; Tan, M.: A Pose Estimation System Based on Deep Neural Network and ICP Registration for Robotic Spray Painting Application, The International Journal of Advanced Manufacturing Technology, 104, 2019, 285–299. https://doi.org/10.1007/s00170-019-03901-0.
[30] Wei, D.; Trombley, C. M.; Sherehiy, A.; Popa, D. O.: Precise and Effective Robotic Tool Change Strategy Using Visual Servoing With RGB-D Camera. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; American Society of Mechanical Engineers; Vol. 85451, p V08BT08A028. https://doi.org/10.1115/DETC2021-72123.
[31] Zhu, W.; Mei, B.; Yan, G.; Ke, Y.: Measurement Error Analysis and Accuracy Enhancement of 2D Vision System for Robotic Drilling, Robotics and Computer-Integrated Manufacturing, 30(2), 2014, 160–171. https://doi.org/10.1016/j.rcim.2013.09.014.