Article

Grasping of Solid Industrial Objects using 3D Registration

Monica Sileo [1], Domenico D. Bloisi [2,*] and Francesco Pierri [1]

[1] School of Engineering, University of Basilicata (Italy); {monica.sileo,francesco.pierri}@unibas.it
[2] Department of Mathematics, Computer Science, and Economics, University of Basilicata (Italy); domenico.bloisi@unibas.it
* Correspondence: domenico.bloisi@unibas.it

Abstract: Robots allow industrial manufacturers to speed up production and to increase the product quality. This paper deals with the grasping of partially known industrial objects in an unstructured environment. The proposed approach consists of two main steps: 1) the generation of an object model, using multiple point clouds acquired by a depth camera from different points of view; 2) the alignment of the generated model with the current view of the object in order to detect the grasping pose. More in detail, the model is obtained by merging different point clouds with a registration procedure based on the Iterative Closest Point (ICP) algorithm. Then, a grasping pose is placed on the model. Such a procedure only needs to be executed once and it works even in the presence of objects only partially known or when a CAD model is not available. Finally, the current object view is aligned to the model and the final grasping pose is estimated. Quantitative experiments using a robot manipulator and three different real-world industrial objects have been conducted to demonstrate the effectiveness of the proposed approach.

Keywords: Robot grasping; 3D registration; Automotive industry; Industrial robots.

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Machines* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

The term *Industry 4.0* was used for the first time in 2011 in order to denote the fourth industrial revolution, which includes the actions needed to create *Smart Factories* [1]. In these smart factories a novel type of robots, called collaborative robots (or *cobots*) [2] are used in order to overcome the classical division of labour, which requires robots to be confined in safety cages far away from human workers. In the context of Industry 4.0, collaborative robots are designed to work in unstructured environments by leveraging on learning capabilities. A challenging issue in collaborative robotics is the grasping of partially known objects. This problem can be divided into other small tasks, equally important, that include object localization, grasp pose detection and estimation and force monitoring during the grasp phase. Moreover, the choice of the contact point between the robot end-effector and the object and the type and amount of forces to be applied is a nontrivial task. The object localization and grasp pose detection task can be resolved by using vision sensors that allow the robot to get information about the environment without entering in contact with it.

It is important to notice that the visual techniques have some drawbacks. In particular, they are affected by the lighting conditions of the environment and the object texture or reflection. Also calibration errors and partial occlusions can occur, especially in the presence of an *eye-in-hand* configuration, i.e., when the camera is rigidly mounted on the robot end-effector (see Fig. 1). This configuration differs from the so-called *eye-to-hand* setup, where the camera observes the robot within its work space. A camera in eye-in-hand configuration has a limited, but more precise, view of the scene, whilst a camera in eye-to-hand configuration has a global, but less detailed, sight of the scene [3].

In this work, we focus on the problem of grasping partially known objects for which a model is not available, with an industrial robot equipped with an eye-in-hand depth sensor in an unstructured environment.

Figure 1. Robot and camera setup for the data acquisition.

The proposed method consists of two steps:

1. The generation of a model of the object based on a set of point clouds acquired from different points of view. The point clouds are merged by means of a 3D registration procedure based on the ICP algorithm. Once the model is obtained, the grasping pose is selected. It is worth noticing that such a procedure is needed only once.
2. The alignment of the obtained model with the current view of the object in order to detect the grasping pose.

The contributions of the paper is threefold.

1. As a difference with respect to expensive 3D scanning systems usually adopted for high production batches, the proposed strategy only requires an off-the-shelf low-cost depth sensor to generate the model and to acquire the current view of the object. Moreover, the proposed system is highly flexible with respect to the position of the object and it allows to acquire different views of the object, since the camera is mounted on the wrist of a robot manipulator.
2. According to the Industry 4.0 road-map, our system is robust to possible failures. In fact, it can detect a potential misalignment between the acquired point cloud and the model. In such a case, the point of view is modified and the whole procedure is restarted.
3. While deep learning-based approaches to object grasping pose detection usually require a huge amount of data and a high computational burden to train the network, the proposed approach exploits a fast model reconstruction procedure.

The rest of the paper is organized as follows. Related work is discussed in Section 2; our strategy for grasping the objects and the adopted methods are described in Section 3. The hardware setup and the software details are presented in Section 4. Section 5 shows the experimental tests conducted by considering three different automotive components. Finally, conclusions are drawn in Section 6.

2. Related work

In this section, the related approaches to object grasping pose detection and some recent registration methods are analyzed.

2.1. Object Grasping

Approaches to the problem of object grasping can be roughly classified into analytic and data-driven [4].

- Analytic methods require a knowledge (at least partial) of the object features (e.g., shape, mass, material) and a model of the contact [5].
- Data-driven approaches aim at detecting the grasp pose candidates for the object via empirical investigations [6].

Among data-driven methods, deep-learning based approaches are becoming very popular thanks to the availability of powerful GPUs. More in detail, in order to make deep-learning techniques very effective a database with geometric object models and a number of good grasp poses is needed. In [7], Convolutional Neural Network (CNN) are adopted with a mobile manipulator, in order to perform a 2D object detection, which combined with the depth information allow to grasp the object. They propose an improvement of the structure of the Faster R-CNN neural network to achieve a better performance and a significant reduction in computational time.

In [8,9] a Generative Grasping Convolutional Neural Network (GG-CNN) has been proposed. It directly generates a grasp pose and quality measure for every pixel in an input depth image and it is fast enough to perform grasping in dynamic environments. Given a depth image $\boldsymbol{I} \in \mathbb{R}^{h \times w}$, where h and w are the height and width of the image, respectively, a grasp is described by $\tilde{\boldsymbol{g}} = (\boldsymbol{s}, \tilde{\phi}, \tilde{w}, q)$, where $\boldsymbol{s} = (u, v)$ is the center in pixel of the box representing the grasp pose, $\tilde{\phi}$ is the grasp rotation in the camera reference frame, $\tilde{w}$ is the grasp width in image coordinates, i.e., the gripper width required for a successful object grasp, and q is a scalar quality measure, representing the chances of grasp success.

The set of grasp poses in the image space is referred as the *grasp map* of $\boldsymbol{I}$, $\boldsymbol{G}$, from which it is possible to compute the best visible grasp in the image reference frame. Then, through the calibration matrices, this pose is expressed in the inertial reference frame to command the robot and grasp the object.

In CNN-based grasping approaches, when the camera is in eye-in-hand configuration, once the grasp pose is determined, often, the robot executes the motion without visual feedback since occlusion appears under a certain distance. For this reason, a precise calibration between the camera and the robot and a completely structured environment are often required. Recently, in [10], grasping of partially known objects in unstructured environments is proposed based on an extension to industrial context of the well-known technique of Background Subtraction [11]. In [12], the authors propose a CNN-based architecture, named GraspNet, in charge of distinguish on the object surface the candidate grasping region.

In the case of unknown objects, where it is assumed neither object knowledge nor grasp pose candidates are available, some approaches approximate the object with shape primitives, e.g., by determining the quadratic function that best approximates the shape of the object using multi-view measurements [13]. Other approaches require to identify some features in sensory data for generating grasp pose candidates [14]. The concept of familiar objects, i.e., known objects similar to that to be grasped in terms of shape, color, texture or grasp poses is exploited in [6]: to transfer the grasp experience, the objects are classified on the basis of a similarity metric. Similarly, in [15] the grasp pose candidates are determined by identifying parts to which a grasp pose has already been successfully tested, and in [16] the objects are classified in categories characterized by the same grasp pose candidates. In [17] a data-driven object grasp approach using only depth-image perception is proposed. In this case, a Deep Convolutional Neural Network has been trained in a simulated environment. The grasps are generated by analytical grasp planners and the algorithm learns grasping-relevant features. At execution time, a single-grasp solution is generated for each object. In [18], some strategies that exploit shape adaptation are presented. Two types of adaptation are used to implement these strategies: the hand/object and the hand/environment adaptation. The first allows to simplify the scene perception. Indeed, the algorithm can make errors in determining the object shape, because they are canceled by the shape adaptation. Moreover, shape adaptation also occurs between the hand and the environment, i.e., the algorithm optimizes the grasping strategy based on the constraints induced by the environment.

The work proposed in [19] is focused on grasping unknown objects in cluttered scenes. A shape-based method, called Symmetry Height Accumulated Features (SHAF), is introduced. This method reduces the scene description complexity and the use of machine learning techniques becomes feasible. SHAF derive from Height Accumulated Features (HAF) [20]. The HAF approach is based on the idea that to grasp an object from top, parts of the end-effector need to envelop the object and, for this reason, need to go further down than the top of the object. Considering small regions, the differences between the average heights give an abstraction of the objects shape. The HAF approach does not check if there is symmetry between features, hence in [19] this approach has been extended by an additional feature type. These symmetry features are used to train a SVM classifier.

An approach that requires as input only the raw depth data from a single frame, does not use explicit object model and is free from online training, is proposed in [21]. The inputs of the algorithm are a depth map and a registered image acquired from a stereo sensor. The first step consists of finding a candidate grasp pose in a 2D slice of the depth map. After that, based on the idea that a solid grasp requires that the shape of the grasped part should be similar to the shape of the gripper interior, the regions of the depth map which better approximate the 3D shape of the gripper interior is computed. To choose between all the found regions, an objective function that assign a score to each region is defined and needs to be maximized. The method is reliable and robust, but, since only a single view is exploited, uncertainties on grasp pose selection could be experienced due to the presence of occluded regions. To overcome this problem, different views can be added.

2.2. 3D Registration

Thanks to the diffusion of powerful graphical processors and low-cost depth sensors, many 3D registration algorithms have been proposed to solve the object localization and reconstruction problem [22]. For example, in [23], the reconstruction of a non-flat steel 3D surface is performed by means of the 3D-Digital Image Correlation (3D-DIC) [24]. Such a method leads to very accurate results, but it requires a time-consuming elaboration and the presence of a known pattern on the surface. Another technique that overcome this drawback is the Iterative Closest Point (ICP) [25], based on an iterative minimization of a suitable cost function. The ICP algorithm has been adopted to reconstruct an entire object starting from point clouds acquired from different views [26,27]. In [28], the ICP, combined with a Genetic Algorithm in order to improve its robustness to local minima, is adopted in an automotive factory environment in order to estimate the pose of car parts. The problem of local minima is addressed also by [29], where a global optimal ICP, based on a branch-and-bound theory, is presented. A recent algorithm for registration of point clouds in the presence of outlier can be found in [30], where the registration problem is reformulated using a truncated least squares cost function. It allows to decouple scale, rotation, and translation estimation in three subproblems solved in cascade thanks to an adaptive voting scheme.

3. Proposed approach

The proposed strategy is shown in Fig. 2 and includes the following steps:

(a) 3D data of the object are acquired from different points of view, e.g., by using a RGB-D camera, in order to obtain n different point clouds of various portions of the object.
(b) The point clouds are merged to obtain the model of the object surface, through a registration algorithm.

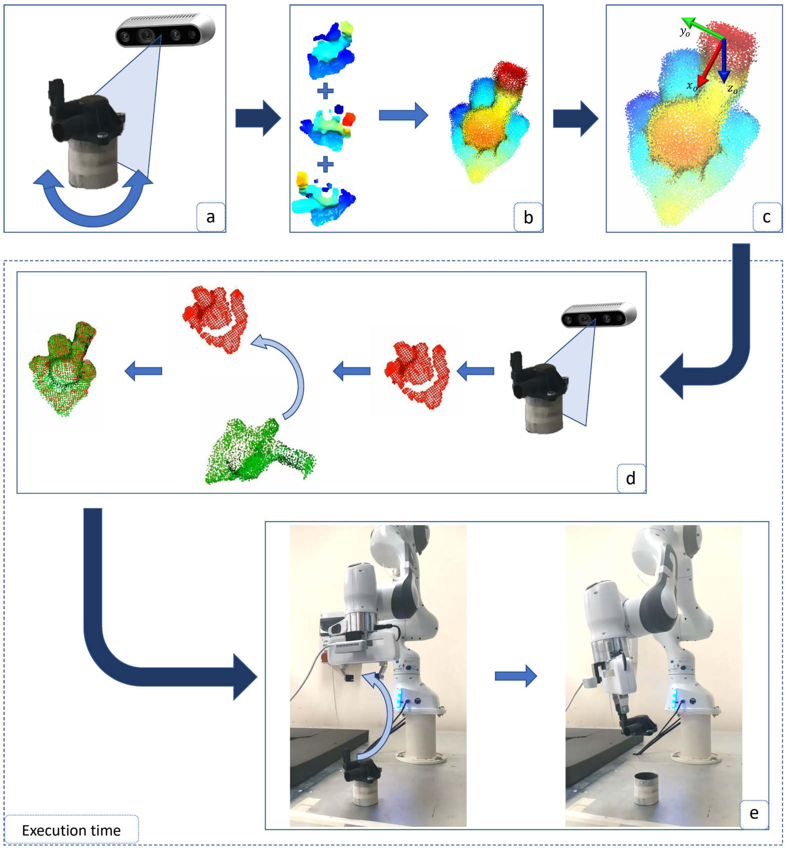


Figure 2. Proposed strategy. (a) Object data acquisition; (b) Model generation; (c) Grasp pose fixing. At execution time: object data acquisition and overlapping with the model (d); coordinate frame transformation and object grasping (e).

(c) A frame that represents the best grasping pose for the object is attached to a point of the model built in the previous step. The grasping point is selected on the basis of the object geometry and the available gripper. Since more than a grasping point can be defined for each object, the one closest to the end-effector frame is selected.
(d) The model is aligned to the current point cloud, in order to be able to transport the grasp pose on the current object. As a measure of the alignment, a *fitness* metric is computed. Thus, in the case of bad alignment, the robot can move the camera in a new position, acquire the object point cloud from a different point of view, and repeat the alignment.
(e) The current grasp pose is transformed into the robot coordinates frame through the camera-end-effector calibration matrix and the robot is commanded to perform the grasp.

The registration algorithm used to merge the initial point clouds to obtain the object model is the Iterative Closest Point algorithm (ICP) [25]. In particular, the point-to-plane version described in [31,32] has been used. The calibration matrix is computed by acquiring a series of images of a calibration target, in arbitrary positions. A calibration target is a panel, with a predefined pattern, and the calibration software knows exactly its size, the color tone and the surface roughness.

3.1. Object model reconstruction

Consider two point clouds obtained by the same surface from two different points of view, $\mathcal{S}$ and $\mathcal{Q}$. They are in registration if, for any pair of corresponding points $s_i \in \mathcal{S}$ and

$q_j \in \mathcal{Q}$, representing the same point on the surface, there exists a unique homogeneous transformation matrix $\boldsymbol{T} \in \mathbb{R}^{4\times 4}$ such that

$$\forall s_i \in \mathcal{S}, \exists q_j \in \mathcal{Q} \;\mid\; \|\boldsymbol{T}\tilde{s}_i - \tilde{q}_j\| = 0. \tag{1}$$

The symbol ˜ in (1) is the homogeneous representation of the coordinate vectors [33], i.e., $\tilde{\boldsymbol{s}}_i = [\boldsymbol{s}_i^{\mathrm{T}} 1]^{\mathrm{T}}$.

Consider n point cloud acquired by means of a RGB-D camera from different views, $\mathcal{P}_i$ $(i = 1, \ldots, n)$, the registration requires to find the homogeneous transformation matrices, $\boldsymbol{T}_i$, that align the point clouds in a common reference frame.

To this aim, the same approach followed in [32], based on the *point-to-plane* Iterative Closest Point (ICP) algorithm, has been adopted. More in detail, the transformation matrix $\boldsymbol{T}_j^i$ $(j = i+1, \ldots, n)$ that aligns $\mathcal{P}_j$ to $\mathcal{P}_i$ is derived by minimizing the following cost function with respect to $\boldsymbol{T}_j^i$

$$\mathcal{C}(\boldsymbol{T}_j^i) = \sum_{\boldsymbol{\pi}_{j,l} \in \mathcal{P}_j,\ \boldsymbol{\pi}_{i,l} \in \mathcal{P}_i} \left((\boldsymbol{T}_j^i \tilde{\boldsymbol{\pi}}_{j,l} - \tilde{\boldsymbol{\pi}}_{i,l})^{\mathrm{T}} \tilde{\boldsymbol{n}}_{j,l}^i \right)^2, \tag{2}$$

where $\tilde{\boldsymbol{n}}_{j,l}^i = \boldsymbol{T}_j^i \tilde{\boldsymbol{n}}_{j,l}$ is the homogeneous representation of the unit vector normal to the surface represented by the point cloud $\mathcal{P}_j$ in the point $\boldsymbol{\pi}_{j,l}$. Each $\boldsymbol{T}_j^i$ is characterized by 12 unknown components: by resorting to a a least-squares estimation, finding the matrix $\boldsymbol{T}_j^i$ that minimizes the function (2) requires at least 4 pair of corresponding points.

This method is exploited in the `multiway registration` algorithm, implemented in the `Open3D` library [34], which has been run to register the acquired point clouds, $\mathcal{P}_i$.

The registered point clouds are, finally, merged in a single point cloud to have the reconstructed object model, i.e.,

$$\mathcal{P}_r = \bigcup_{i=1}^{n} \bigcup_{j=1}^{N_i} \boldsymbol{T}_i \tilde{\boldsymbol{\pi}}_{i,j}. \tag{3}$$

3.2. Grasp point estimation

Once the model of the object has been built, one grasp point candidate, O, is selected and the relative coordinate frame $\mathcal{F}_o = O, x_o y_o z_o$ is defined. The RGB-D camera acquires a point cloud, $\mathcal{P}_a$, of the object to be grasped and such a point cloud is aligned to the known one (3) representing the model. Again, a procedure based on the ICP algorithm is applied:

1. A set of local features, called Fast Point Feature Histograms (FPFH), are extracted from each point of $\mathcal{P}_a$ [35];
2. The corresponding points of the two point clouds are computed by using a RANSAC (RANdom SAmple Consensus) algorithm [36]: at each iteration, given μ points randomly extracted from $\mathcal{P}_r$ the corresponding points of $\mathcal{P}_a$ are the nearest with respect to the extracted features.
3. The transformation matrix computed at previous step is used as an initial guess for the ICP algorithm aimed at refining the alignment.

If the acquired point cloud is not very detailed, the previous algorithm leads to accurate results only in the presence of a small orientation error between the two point clouds, otherwise poor surface alignments can be obtained. To avoid this issue and to have an accurate estimation of the grasping pose, the acquired point cloud is compared with n_R different point clouds, obtained by rotating the reconstructed model of an angle $2\pi/n_R$. The point cloud with the best match is then selected to compute the grasping pose. The best match is measured through a *fitness* metric, which measures the overlapping area between the two point clouds. In particular, the fitness is computed as the ratio between the number

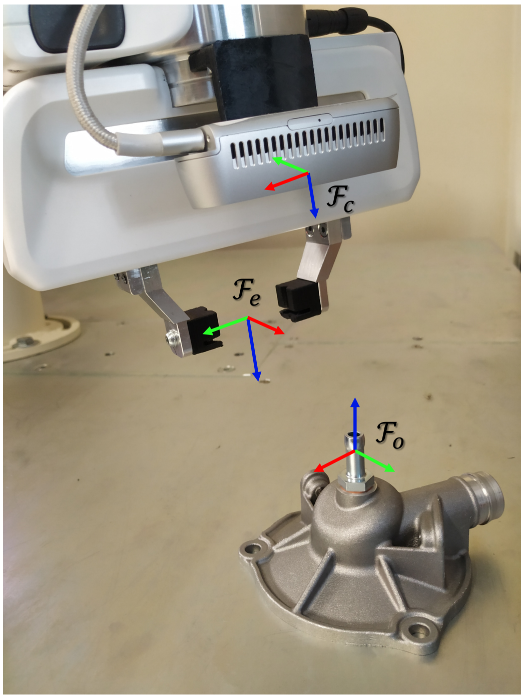


Figure 3. Reference frames for the end-effector, the camera, and the object.

of correspondence points, i.e., points for which has been found the corresponding point in the target point cloud and the number of the points in the target point cloud.

Once the point cloud model $\mathcal{P}_r$ is aligned with the acquired one $\mathcal{P}_a$, it is possible to localize the position of the grasping point O and the orientation of the corresponding reference frame in the camera coordinate frame. Finally, trough a camera calibration process [37], it is possible to compute the camera-end-effector transformation and transform the grasping pose in the robot base coordinate frame.

3.3. Grasping

Define the coordinate frame $\mathcal{F}_e$ attached to the robot end-effector as shown in Fig. 3. The grasp requires the alignment of $\mathcal{F}_e$ to the object's frame $\mathcal{F}_o$. To this aim, a trajectory planner for the end-effector is implemented by assigning three way-points, namely: the current pose, a point along the z axis of the object reference frame at a distance of 10 cm to the origin O, and the origin of the object frame O.

Regarding the orientation of $\mathcal{F}_e$, the planner aligns the axis z_e to $-z_o$ and y_e to y_o before reaching the second way-point and then it is kept constant for the last part of the path.

By denoting with $\boldsymbol{x}_d$ and $\boldsymbol{x}_e$ the desired and the actual end-effector pose, respectively, the velocity reference for the robot joints, $\dot{\boldsymbol{q}}_r$, are computed via a closed-loop inverse kinematics algorithm [33]

$$\dot{\boldsymbol{q}}_r = \boldsymbol{J}^{\dagger}(\boldsymbol{q})(\dot{\boldsymbol{x}}_d + \boldsymbol{K}(\boldsymbol{x}_d - \boldsymbol{x}_e)), \tag{4}$$

where $\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ is the right pseudo-inverse of the Jacobian matrix and $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ is a matrix of positive gains.

4. Implementation

The experimental setup consists of a collaborative robot Franka Emika Panda [38] equipped with an Intel RealSense D435 camera in eye-in-hand configuration as shown in Fig. 1. The `libfranka` C++ open source library is used to control the robot by means of an external workstation through Ethernet connection. The workstation, equipped with an Intel Xeon 3.7 GHz CPU with 32 GB RAM, runs the Ubuntu 18.04 LTS operating system with a real-time kernel. The camera has been previously calibrated with a set of 30 images of a 2D

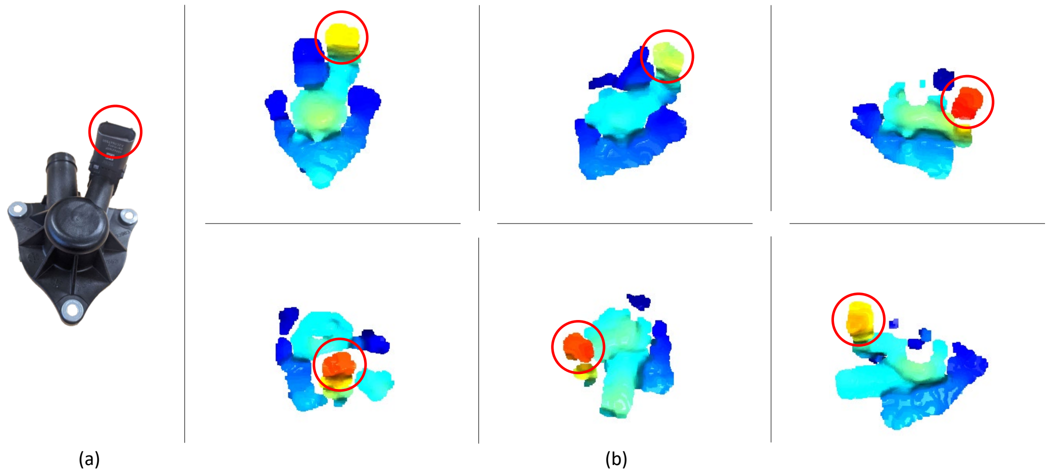


Figure 4. Some examples of point cloud (b) for the plastic oil separator crankcase (a). The red circle indicates the same part in the various views.

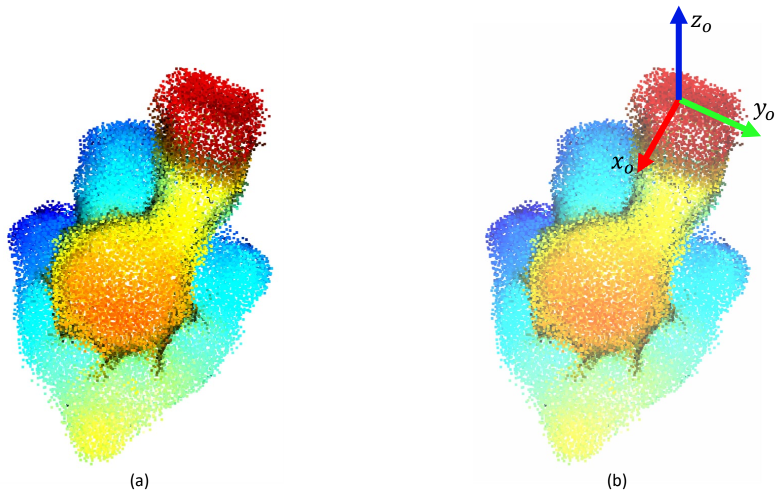


Figure 5. Example of the generated model for one object (a) and the relative grasp pose (b).

checkerboard flat pattern through the method developed in [37]. The vision software runs on the same workstation of the robot control, while the camera data acquisition requires the `librealsense2` library.

5. Experimental results

The proposed approach has been evaluated by considering three different objects used in a real-world automotive factory. Each object is located above the table surface to allow a faster background elimination from the point cloud.

To generate the model, having the camera in a fixed position, the object have been rotated to allow the data acquisition in 30 different configurations. Examples of acquired point cloud are shown in Fig. 4. Then, according to the method described in Section 3.1, these point clouds are merged by using the ICP algorithm and a point cloud of the whole object is obtained (see Fig. 5a). This point cloud represents the object model in which the grasp pose will be selected, by using any modeling software. An example of a grasping pose is shown in Fig. 5b.

The mechanical workpieces used in the experiments and their corresponding generated model are shown in Fig. 6.

When an object needs to be grasped, its point cloud is acquired and it is overlapped with the one that represents the model. As detailed in Section 3.2, the current point cloud is

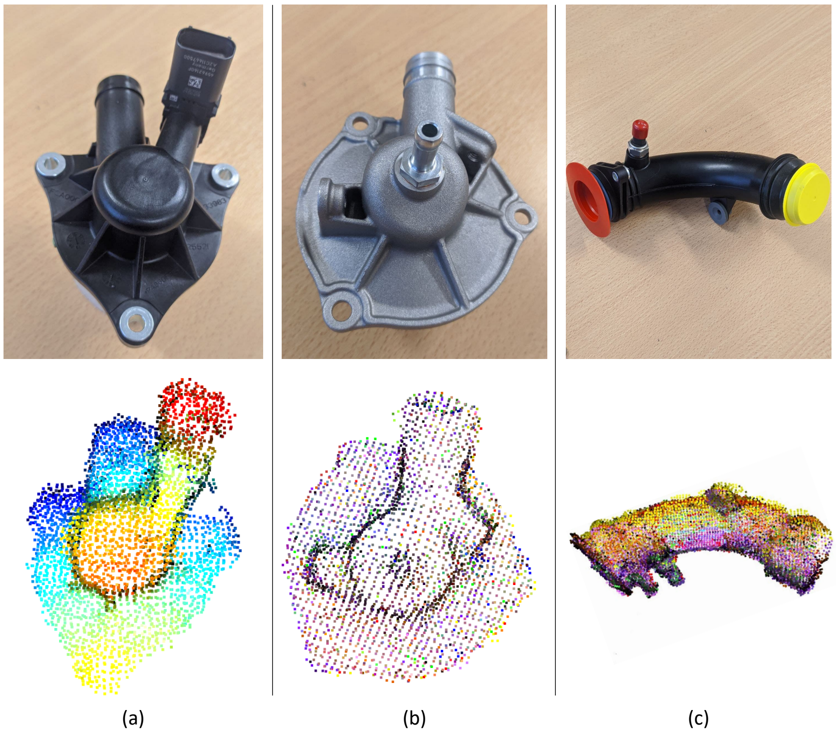


Figure 6. Mechanical workpieces and relative generated models: (a) plastic oil separator crankcase, (b) metal oil separator crankcase, (c) air pipe.

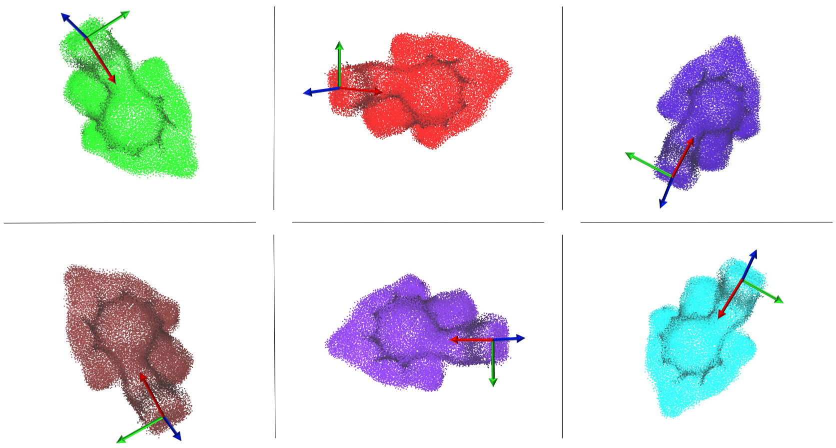
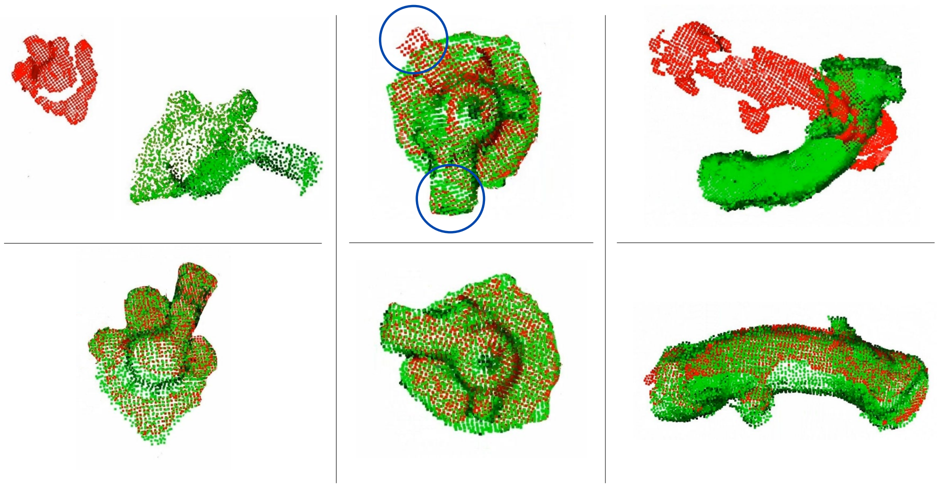


Figure 7. Example of the various models with different orientations for the plastic oil separator crankcase. The red, green and blue arrows represent the x, y and z axis, respectively.

compared with the model point cloud with eight different orientations, in order to find the best matching. Fig. 7 shows the models with different orientations.
In order to evaluate our approach the following procedure has been implemented:

- for each model orientation, a maximum number of 100 iterations was established;
- two thresholds for the fitness are defined: threshold f_l below which the overlap is considered failed and threshold f_h above which the overlap is considered good enough;
- during the overlapping, if threshold f_h is exceeded, the algorithm stops and no further comparisons are made;
- if no overlap exceeds the threshold f_h, the one with the highest fitness is considered;
- if no overlap exceeds threshold f_l, the algorithm reports a failure.



(a) (b) (c)

Figure 8. Examples of overlap failure (top row) and successful (bottom row) for three objects: (a) plastic oil separator crankcase, (b) metal oil separator crankcase, (c) air pipe. The current point clouds acquired by the depth sensor are in red, while the model point clouds are in green. The blue circles highlight the non-overlapping for the metal oil separator crankcase by indicating the same object part not aligned.

Examples of correct and incorrect overlapping for the three considered workpieces are reported in Fig. 8.

In the case of failure, the robot manipulator moves the camera around the object in order to acquire a new image from a different point of view. Then, the whole procedure is restarted.

After the above procedure, the labeled grasp pose can be projected on the current object, that is referred with respect to the robot base frame. After a further transformation, by using the camera-end-effector calibration matrix, the robot can be commanded to perform the object grasp.

Regarding the plastic and metal oil separator crankcases (see Fig. 6a and Fig. 6b), the algorithm was able to find the match and the robot was able to grasp the object.

Define the estimation grasping position and orientation errors as

$$\boldsymbol{e}_p^o = \boldsymbol{p}^o - \hat{\boldsymbol{p}}^o, \tag{5}$$

$$\boldsymbol{e}_\phi^o = \boldsymbol{\phi}^o - \hat{\boldsymbol{\phi}}^o, \tag{6}$$

where $\boldsymbol{p}$ ($\boldsymbol{\phi}$) is the actual grasping point position (orientation, expressed as a triple of Euler angles [33]) while $\hat{\boldsymbol{p}}$ ($\hat{\boldsymbol{\phi}}$) are the estimates provided by the visual algorithm. The superscript ^{o} means that all the variables are expressed in the object frame $\mathcal{F}_o$.

Tables 1-2, report the errors for the plastic oil separator crankcase and the metal one, respectively. A set of snapshots of the grasping procedure for the two objects is detailed in Fig. 9.

As can be observed, on 18 successfully tests, conducted in different light conditions due to the presence of natural light in the environment, the mean error is about 3.82 mm (0.15 radians) for the plastic oil separator crankcase and 4.64 mm (0.06 radians) for the metal one. However, a wide deviation is experienced in the different tests, as witnessed by the values of the standard deviations in the Tables. This is mainly due to the adoption of low-detailed point clouds. More in general, regarding the whole experimental campaign, a success rate of 88.3% has been experienced for the plastic oil separator crankcase and 84.8% for the metal one.

Although the model has been well-built, for the air pipe (see Fig. 6c) the experiments show that the search for the best match was not successful. This is probably due to the symmetry of object and the model not very accurate. The algorithm was not able to find

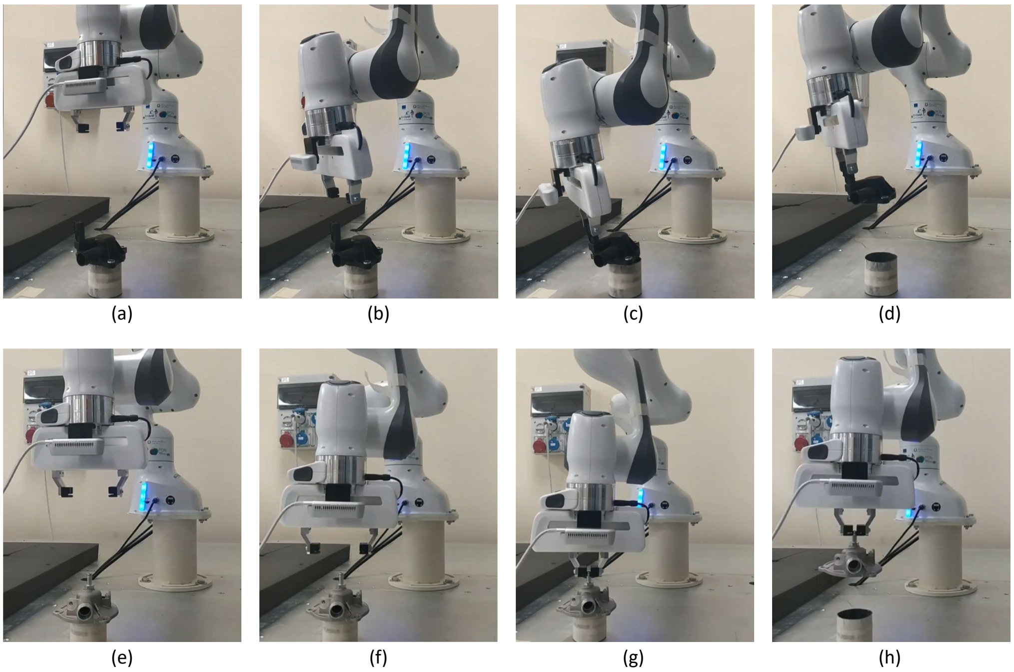


Figure 9. Snapshots of the grasping procedure for two different objects: (a)-(e) a point cloud is acquired; (b)-(f) the robot approaches the object close to the estimated grasping point; (c)-(g) the end effector grasps the object; (d)-(h) the object is raised by the robot.

the match because many portions of the object are quite similar. Correct overlaps were found only when the object orientations are close to that considered for the model. In this case, only a success rate of 32.7% has been experienced.

The obtained results show that the proposed method can be promising for the grasping of partially known objects in the absence of a CAD model, but it requires a further investigation in order to better analyze the features required for a correct execution of the registration and make it working also on symmetric components.

A video of the execution can be found at https://web.unibas.it/automatica/machines.html while the code is available in the GitHub repository at https://github.com/sileom/graspingObjectWithModelGenerated.git.

6. Conclusions

A method to handle the problem of grasping partially known objects in unstructured environment has been proposed. The approach can be used in absence of accurate object models and consists of a comparison between a point cloud of the object and a model built from a set of point clouds previously acquired. The experiments, conducted on a set of mechanical workpieces used in real world automotive factories, show that the method is applicable in case of objects with particular shapes, but not in the case of objects with symmetric shape. Camera features influence the overall performance: a more accurate sensor could allow to build a more detailed model to improve the performance and robustness of the approach. Future work will be devoted to extend the method to any kind of components.

Author Contributions: Conceptualization, M.Sileo, D.D.Bloisi and F.Pierri; Methodology, M.Sileo, D.D.Bloisi and F.Pierri; Software, M.Sileo and D.D.Bloisi; Investigation, M.Sileo; Validation, M.Sileo; Formal analysis, M.Sileo, D.D.Bloisi; Writing - original draft, M.Sileo; Writing - review & editing, D.D.Bloisi and F.Pierri; Supervision, D.D.Bloisi and F.Pierri; Project administration, D.D.Bloisi and F.Pierri; Funding acquisition, D.D.Bloisi and F.Pierri. All authors have read and agreed to the published version of the manuscript.

Table 1. Test results for the plastic oil separator crankcase. The position errors are in *mm* while the orientation errors are in *rad*.

Test	e_{p_x}	e_{p_y}	e_{p_z}	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
1	1.964	5.316	5.858	0.293	0.084	0.273
2	4.759	1.076	4.006	0.176	0.123	0.233
3	8.460	1.400	1.040	0.048	0.116	0.237
4	0.600	0.380	0.000	0.304	0.072	0.142
5	2.260	2.300	8.200	0.392	0.263	0.130
6	0.310	3.320	6.433	0.169	0.489	0.083
7	5.011	1.637	5.641	0.133	0.171	0.164
8	4.989	1.151	4.758	0.219	0.146	0.230
9	1.240	1.331	5.402	0.202	0.065	0.155
10	4.373	0.095	10.515	0.016	0.170	0.107
11	5.966	1.398	3.785	0.101	0.019	0.179
12	1.442	4.579	5.702	0.104	0.052	0.189
13	4.529	1.033	10.044	0.048	0.115	0.099
14	2.042	1.057	3.729	0.092	0.033	0.113
15	3.533	1.509	2.023	0.110	0.088	0.166
16	2.202	5.128	3.623	0.084	0.154	0.043
17	4.798	3.496	7.289	0.066	0.019	0.216
18	7.122	2.755	13.521	0.012	0.254	0.146
Mean error	**3.644**	**2.164**	**5.643**	**0.143**	**0.135**	**0.161**
Standard deviation	**2.219**	**1.542**	**3.299**	**0.103**	**0.110**	**0.059**

Table 2. Test results for the metal oil separator crankcase. The position errors are in *mm* while the orientation errors are in *rad*.

Test	e_{p_x}	e_{p_y}	e_{p_z}	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
1	1.143	3.243	13.409	0.039	0.015	0.133
2	3.157	2.066	10.148	0.005	0.051	0.146
3	0.852	6.250	4.402	0.013	0.042	0.291
4	8.063	7.524	0.843	0.024	0.017	0.018
5	6.522	4.445	7.708	0.012	0.019	0.053
6	2.922	0.093	5.704	0.024	0.005	0.068
7	1.727	2.234	6.592	0.013	0.002	0.096
8	0.379	1.559	11.784	0.013	0.026	0.009
9	7.802	1.887	7.063	0.060	0.091	0.054
10	8.114	0.677	8.383	0.023	0.054	0.028
11	2.548	1.079	8.268	0.002	0.040	0.079
12	2.814	2.884	6.487	0.169	0.003	0.183
13	0.816	0.460	13.407	0.017	0.024	0.157
14	5.874	0.002	7.367	0.070	0.189	0.073
15	9.727	0.415	2.165	0.001	0.105	0.000
16	5.761	0.153	9.530	0.038	0.101	0.012
17	4.702	3.785	0.587	0.091	0.082	0.027
18	5.360	2.653	7.143	0.002	0.029	0.051
Mean error	**4.349**	**2.301**	**7.277**	**0.034**	**0.050**	**0.082**
Standard deviation	**2.845**	**2.077**	**3.616**	**0.041**	**0.047**	**0.073**

Funding: This research has been supported by the project ICOSAF (Integrated collaborative systems for Smart Factory - ARS01_00861), funded by MIUR under PON R&I 2014-2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code is available in the GitHub repository reachable at the following link https://github.com/sileom/graspingObjectWithModelGenerated.git.

Acknowledgments: The authors would like to thank Ms. Marika Colangelo for her help with software implementation.

Conflicts of Interest: No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

References

1. Karabegović, I. Comparative analysis of automation of production process with industrial robots in Asia/Australia and Europe. *International Journal of Human Capital in Urban Management* **2017**, *2*, 29–38.
2. Weiss, A.; Wortmeier, A.K.; Kubicek, B. Cobots in industry 4.0: A roadmap for future practice studies on human–robot collaboration. *IEEE Transactions on Human-Machine Systems* **2021**, *51*, 335–345.
3. Ozkul, T. Equipping legacy robots with vision: performance, availability and accuracy considerations. *International Journal of Mechatronics and Manufacturing Systems* **2009**, *2*, 331–347.
4. Sahbani, A.; El-Khoury, S.; Bidaud, P. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems* **2012**, *60*, 326–336.
5. Bicchi, A.; Kumar, V. Robotic grasping and contact: A review. In Proceedings of the Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). IEEE, 2000, Vol. 1, pp. 348–353.
6. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-driven grasp synthesis–a survey. *IEEE Transactions on Robotics* **2013**, *30*, 289–309.
7. Zhang, H.; Tan, J.; Zhao, C.; Liang, Z.; Liu, L.; Zhong, H.; Fan, S. A fast detection and grasping method for mobile manipulator based on improved faster R-CNN. *Industrial Robot: the international journal of robotics research and application* **2020**.
8. Morrison, D.; Corke, P.; Leitner, J. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. In Proceedings of the Robotics: Science and Systems (RSS), 2018.
9. Morrison, D.; Corke, P.; Leitner, J. Learning robust, real-time, reactive robotic grasping. *The International journal of robotics research* **2020**, *39*, 183–201.
10. Sileo, M.; Bloisi, D.D.; Pierri, F. Real-time Object Detection and Grasping Using Background Subtraction in an Industrial Scenario. In Proceedings of the 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI). IEEE, 2021, pp. 283–288.
11. Bloisi, D.D.; Pennisi, A.; Iocchi, L. Background modeling in the maritime domain. *Machine vision and applications* **2014**, *25*, 1257–1269.
12. Asif, U.; Tang, J.; Harrer, S. GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices. In Proceedings of the IJCAI, 2018, pp. 4875–4882.
13. Dune, C.; Marchand, E.; Collowet, C.; Leroux, C. Active rough shape estimation of unknown objects. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008, pp. 3622–3627.
14. Kraft, D.; Pugeault, N.; Başeski, E.; POPOVIĆ, M.; Kragić, D.; Kalkan, S.; Wörgötter, F.; Krüger, N. Birth of the object: Detection of objectness and extraction of object shape through object–action complexes. *International Journal of Humanoid Robotics* **2008**, *5*, 247–265.
15. Detry, R.; Ek, C.H.; Madry, M.; Piater, J.; Kragic, D. Generalizing grasps across partly similar objects. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 3791–3797.
16. Dang, H.; Allen, P.K. Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 1311–1317.
17. Schmidt, P.; Vahrenkamp, N.; Wächter, M.; Asfour, T. Grasping of unknown objects using deep convolutional neural networks based on depth images. In Proceedings of the 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, pp. 6831–6838.
18. Eppner, C.; Brock, O. Grasping unknown objects by exploiting shape adaptability and environmental constraints. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013, pp. 4000–4006.
19. Fischinger, D.; Vincze, M.; Jiang, Y. Learning grasps for unknown objects in cluttered scenes. In Proceedings of the 2013 IEEE international conference on robotics and automation. IEEE, 2013, pp. 609–616.
20. Fischinger, D.; Vincze, M. Empty the basket-a shape based learning approach for grasping piles of unknown objects. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 2051–2057.

21. Klingbeil, E.; Rao, D.; Carpenter, B.; Ganapathi, V.; Ng, A.Y.; Khatib, O. Grasping with application to an autonomous checkout robot. In Proceedings of the 2011 IEEE international conference on robotics and automation. IEEE, 2011, pp. 2837–2844.
22. Bellekens, B.; Spruyt, V.; Berkvens, R.; Penne, R.; Weyn, M. A benchmark survey of rigid 3D point cloud registration algorithms. *Int. J. Adv. Intell. Syst* **2015**, *8*, 118–127.
23. Nigro, M.; Sileo, M.; Pierri, F.; Genovese, K.; Bloisi, D.D.; Caccavale, F. Peg-in-hole using 3D workpiece reconstruction and CNN-based hole detection. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 4235–4240.
24. Sutton, M.A.; Orteu, J.J.; Schreier, H. *Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications*; Springer Science & Business Media, 2009.
25. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image and vision computing* **1992**, *10*, 145–155.
26. Valgma, L.; Daneshmand, M.; Anbarjafari, G. Iterative closest point based 3d object reconstruction using rgb-d acquisition devices. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU). IEEE, 2016, pp. 457–460.
27. Shuai, S.; Ling, Y.; Shihao, L.; Haojie, Z.; Xuhong, T.; Caixing, L.; Aidong, S.; Hanxing, L. Research on 3D surface reconstruction and body size measurement of pigs based on multi-view RGB-D cameras. *Computers and Electronics in Agriculture* **2020**, *175*, 105543.
28. Lin, W.; Anwar, A.; Li, Z.; Tong, M.; Qiu, J.; Gao, H. Recognition and pose estimation of auto parts for an autonomous spray painting robot. *IEEE Transactions on Industrial Informatics* **2018**, *15*, 1709–1719.
29. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *38*, 2241–2254.
30. Yang, H.; Shi, J.; Carlone, L. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics* **2020**, *37*, 314–333.
31. Choi, S.; Zhou, Q.Y.; Koltun, V. Robust reconstruction of indoor scenes. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5556–5565.
32. Nigro, M.; Sileo, M.; Pierri, F.; Bloisi, D.; Caccavale, F. Assembly task execution using visual 3D surface reconstruction: An integrated approach to parts mating. *Robotics and Computer-Integrated Manufacturing* **2023**, *81*, 102519.
33. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics – Modelling, Planning and Control*; Springer: London, UK, 2009.
34. Zhou, Q.Y.; Park, J.; Koltun, V. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847* **2018**.
35. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE international conference on robotics and automation. IEEE, 2009, pp. 3212–3217.
36. Li, H.; Qin, J.; Xiang, X.; Pan, L.; Ma, W.; Xiong, N.N. An efficient image matching algorithm based on adaptive threshold and RANSAC. *IEEE Access* **2018**, *6*, 66963–66971.
37. Tsai, R.Y.; Lenz, R.K.; et al. A new technique for fully autonomous and efficient 3 D robotics hand/eye calibration. *IEEE Transactions on robotics and automation* **1989**, *5*, 345–358.
38. Franka Emika Panda. https://www.franka.de/.