Journal of Intelligent & Robotic Systems (2022) 104: 16
https://doi.org/10.1007/s10846-021-01558-0

REGULAR PAPER

Robot End-Effector Mounted Camera Pose Optimization in Object Detection-Based Tasks

Loris Roveda[1] · Marco Maroni[2] · Lorenzo Mazzuchelli[2] · Loris Praolini[2] · Asad Ali Shahid[1] · Giuseppe Bucca[2] · Dario Piga[1]

Received: 19 January 2021 / Accepted: 10 December 2021 / Published online: 30 December 2021

Abstract
Robots equipped with the vision systems at the end-effector provide a powerful combination in industrial contexts, allowing to execute a wide range of manufacturing tasks, such as inspection applications. While many works are dedicated to machine vision algorithms, the optimization of the vision system pose is not properly addressed. Optimizing the sensor pose, in fact, can increase the object detection performance, avoiding occlusions and collisions in the real working scene. Therefore, the development of an approach capable of optimizing the pose of a vision system is the main objective of this paper. A complete pipeline for such optimization is proposed, composed of the following main components: working scene reconstruction, robot-environment collisions modeling, object detection, sensor pose optimization (exploiting Bayesian Optimization, a state of the art methodology), and collision-free robot motion planning. To validate the proposed approach, experimental tests have been executed considering two object detection-based tasks. A Franka EMIKA Panda robot equipped with an Intel© RealSense D400 at its end-effector has been employed as a robotic platform. Achieved results show the high-fidelity reconstruction of the real working environment for an offline optimization (i.e., performed simulations), as well as the capabilities of the employed Bayesian Optimization-based approach to define the sensor pose. The proposed optimization methodology has been compared with the grid point approach, showing an improved performance for camera pose optimization purposes. An additional experiment has been performed in order to show the possibility to exploit a digital twin (if available) of the working scene instead of the environment reconstruction (to reduce the computational resources and to avoid measurements noise in the 3D reconstruction). Obtained results show the feasibility of the proposed pipeline employing such a digital twin.

Keywords Camera pose optimization · Bayesian optimization · Vision systems · Industry 4.0 · Industrial robotics · Inspection tasks

1 Introduction

1.1 Context

Industry 4.0 paradigm [1] has put a huge importance on the autonomy of robotic platforms inside production plants. Having the robot able to self-adapt to its working environment is becoming crucial in many applications, such as assembly [2] and human-robot interaction [3] tasks. Machine vision is enhancing such autonomy [4–6], providing the manipulator with the capability to sense its working environment. Many robotic applications

✉ Loris Roveda
loris.roveda@idsia.ch

Marco Maroni
marco2.maroni@mail.polimi.it

Lorenzo Mazzuchelli
lorenzo.mazzuchelli@mail.polimi.it

Loris Praolini
loris.praolini@mail.polimi.it

Asad Ali Shahid
asadali.shahid@idsia.ch

Giuseppe Bucca
giuseppe.bucca@polimi.it

Dario Piga
dario.piga@supsi.ch

[1] Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA), Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Università della Svizzera Italiana (USI), 6928 Manno, Switzerland

[2] Department of Mechanical Engineering, Politecnico di Milano, 23900 Lecco, Italy

benefit from machine vision to improve the task performance, such as high-accuracy assembly tasks [7], human-robot collaboration [8], mobile robot operations [9], etc. In particular, object detection-based tasks, such as parts manipulation [10] and quality inspection [11], are becoming extremely powerful. A robot mounted with the camera at the end-effector is positioned for the localization of complex parts, even in the cluttered environments [12, 13]. In such scenario, the robot kinematics can be exploited to find the most suitable pose of the camera, maximizing the object detection performance and, therefore, the task success. Although the performance of machine vision algorithms has improved in difficult situations [14–16], there is still a lack of optimization algorithms to manage the pose of vision system in such tasks. Therefore, considering the described object detection-based tasks, this paper aims to propose a methodology for the optimization of the (end-effector mounted) camera pose to maximize the target part detection performance (e.g., to be exploited in quality inspection applications).

1.2 Related Work

The correct object detection and its pose estimation are currently hot-research topics [17–20]. A deep analysis of the state of the art object detection methodologies can found in Appendix 1. In general, the main drawback of the object detection approaches is that the optimization of the vision system pose to maximize the object detection performance is not considered. Indeed, such capability is of tremendous importance in many real robotic applications where the correct and optimized object detection is required in order to complete the task. In many real working conditions, in fact, the obstacles and occlusions are present, reducing drastically the chances to have a clear view of the part [21, 22]. Therefore, the correct positioning of the vision system in the working environment becomes necessary. With this aim, some works can be identified from the literature review. Ercan et al. [23] proposes the optimal placement of multiple cameras and the selection of the best subset of cameras for a single-object localization in the framework of sensor networks. Olague and Mohr [24] proposes a methodology to optimize the placement of a camera in order to minimize the detection error in the 3D measurements, treating the issue as an optimization problem. In [25] a method for automatic sensor placement for model-based object detection is described. The optimal sensor placements and the shortest path through these viewpoints is determined. In fact, by moving the vision system from one pose to another around the object to observe all features of interest, it is possible to correctly detect the target object. During the sensor planning, object features are re-sampled as individual points attached with surface normals. The optimal sensor placement graph is constructed by a genetic algorithm in which a min-max criterion has been used for the evaluation. The problem of camera placement for automated visual inspection tasks is studied under a multi-objective framework in [26], exploiting an evolutionary based technique. McGreavy et al. [27] proposed an approach for the computation of the next best view for the object detection purposes. Firstly, the visibility of the object candidate from a set of candidate views that are reachable by a robot is analyzed. Then, the visibility of the object features by projecting the model of the most likely object into the scene is analyzed. In such a way, the next best view for the object detection purposes is performed. In [28] a framework for the definition of the vision system positioning is proposed that uses an analysis of the object stable poses, together with dynamic simulation to predict the probability distribution of initial object poses. A scalable approach to determine a small number of well-placed camera viewpoints for optical surface inspection planning is proposed in [29]. By defining a set of geometric feature functionals, an adaptive, non-uniform surface sampling (sparse in geometrically low-complexity areas, and dense in regions of higher complexity) is performed. The main drawback of the above described approaches is that occlusions are not considered. If an object is not (at least) partially visible, the proposed algorithms are not able to compute the pose of a vision system. In addition, the cluttered environments are not considered, where the likelihood of the robot colliding with other objects in the scene is inevitable. Moreover, the vision system positioning criteria is not always related to the maximization of the object detection performance.

1.3 Paper Contribution

Within the above described context, the main objective of this paper is to develop an approach capable of optimizing the pose of the vision system mounted at the robot's end-effector. In fact, due to the high flexibility demanded for robotic cells (e.g., parts are provided to the robot in - partially - unknown conditions, especially in the case of human-robot collaboration scenarios [30]), the robot needs to autonomously locate the target component, thus, manipulating it with high precision. Therefore, the optimization of the sensor pose is required, due to the fact that the preliminary scanning of the operating environment might not result in a sufficient accuracy to properly locate the target component. Indeed, a complete pipeline for the optimization of the sensor pose to maximize object detection performance is proposed, composed of the following main components: working scene reconstruction, robot-environment collisions modeling, object detection, sensor pose optimization, and

collision-free robot motion planning. More in details, the optimization of the sensor pose is performed offline (i.e., in a simulation environment) exploiting the reconstructed working scene, including collisions modeling. Bayesian Optimization (BO) [31] is employed to optimize the sensor pose with proper definition of a cost function taking into consideration the maximization of the object detection performance, and the reachability requirements (w.r.t. both robot kinematics and collisions constraints). Such an optimization methodology, in fact, is known to be efficient in terms of the number of function evaluations for derivative-free optimization in comparison with other methods, such as Particle Swarm Optimization and Genetic Algorithms [32, 33]. Once the optimization is performed, the online collision-free robot motion planning is generated in order to position the sensor in an optimized pose, exploiting the reconstructed working scene and the collisions modeling. The target object detection is then finally performed, identifying the object and estimating its pose within the real working scene. The main novelty of the proposed paper are therefore:

- the capability to reconstruct a high-fidelity simulation environment based on measured real data, including both the modeling of occlusions and collisions;
- the capability to optimize the sensor pose offline to maximize the object detection performance, while guaranteeing occlusions and collisions avoidance. Exploiting the BO, the proposed approach allows to minimize the simulations iterations, making it possible to reduce the computational time;
- the capability to transfer the offline optimized sensor pose to the real robot, generating a collisions-free motion plan online to execute the target object detection-based task.

To validate the proposed approach, experimental tests have been executed in two different scenarios, in which a target part has to be detected: a controlled-conditions environment (i.e., without the presence of random positioned obstacles/occlusion elements) and an industrial-like environment (i.e., representing the real industrial working scene). An industrial use-case has been selected due to the fact that, nowadays, the production environment is becoming more and more flexible [34, 35], requiring the robot to operate in an unstructured environment. In fact, operators can work in parallel on the same task tackled by the robot, positioning the parts in unpredictable positions and requiring the robot to localize them to continue its work [8]. In addition, the robotic systems can be placed inside a (partially) new working scene, making it necessary to locate themselves and the target parts to perform their applications [21]. Other industrial applications that can benefit from the proposed pipeline are bin-picking tasks, where parts are randomly positioned inside the robot's work space and require the robot to define the sensor pose to maximize the object detection performance while avoiding collisions with the surrounding environment [22]. The proposed approach can be also exploited in the service or assistive robotics tasks, without requiring the change in proposed pipeline. A Franka EMIKA Panda robot has been employed as a robotic platform, equipped with an Intel© RealSense D400 at its end-effector. Achieved results show the high-fidelity reconstruction of the working environment for an offline optimization (i.e., performed simulations), making it possible to model the camera occlusions and collisions between the robotic system and the environment. The performance of the proposed approach for the definition of the sensor pose is analyzed and compared with a grid point sampling method. The proposed BO-based approach requires drastically reduced number of simulations/experiments to compute the optimized sensor pose (i.e., reducing the processing time). The improved results exploiting the proposed methodology are highlighted, showing the importance of an optimized camera positioning to achieve high performance for critical object detection-based tasks (such as grasping and inspection robotic applications). An additional experiment has been performed in order to replace the scene reconstruction component with the digital twin of the operating scene. In fact, if a modeling of the working environment is available (such as for the inspection task within the aerospace industry defined in the H2020 CS2 ASSASSINN project (https://makerfairerome.eu/it/espositori/?edition=2020&exhibit=200113), it is possible to exploit such digital twin to perform the sensor pose optimization offline (reducing the computational resources and eliminating the measurements noise in the working scene reconstruction). Achieved results show the capabilities of the proposed approach to exploit a digital twin of the operating scene and provide an effective solution applicable to real industrial tasks.

Therefore, the proposed pipeline exploiting the reconstruction of the operating environment is effective in the following cases:

- in the situation where several candidate positions are available;
- in the situation where the object detection performance achieved in the reconstruction phase are not accurate enough for the high-precision manipulation of the object;
- a digital twin of the operating environment is not available/not reliable due to the high complexity of the working scene.

The proposed pipeline exploiting the digital twin of the operating environment is effective in the following cases:

- the modeling of an operating environment is available;

– the sensor pose can be managed in order to maximize the object detection performance.

The following assumptions are made in order to exploit the proposed pipeline:

– the target object is available in the operating scene. This assumption is valid in many applications, e.g., quality inspection tasks;
– the reconstructed environment must contain the target object to be detected. An initial (rough) estimation on the positioning of the part is required (commonly, such position is roughly known in industrial applications [22]). The proposed pipeline allows the user to define the expected location of the target part (with a considered acceptable approximation of ± 15 cm based on the performed tests, this tolerance is acceptable for the most quality inspection tasks where installation errors are of the order of mm) to run the proposed optimization pipeline;
– a single part to be detected is considered in the operating environment. If multiple parts have to be detected, the proposed pipeline can be run from scratch for every single component, properly restricting the operating environment around the specific inspection location.

For the sake of completeness, it is worth mentioning the other works applying Bayesian optimization in robotics applications. Cully et al. [36] describes a trial-and-error algorithm that allows robots to adapt their behaviour in the presence of a damage; [37] employs constrained BO to select three force-controller parameters for combined position/interaction tasks; [38] proposes a methodology to achieve automatic tuning of optimal parameters for whole-body control algorithms, iteratively learning the parameters of sub-spaces from the whole high-dimensional parametric space through interactive trials; [39] adapts the robot trajectory by means of force measurements-guided Bayesian Optimization; [40] adopts Bayesian Optimization in order to regulate the robot control gains to maximize trajectory tracking performance; [2] optimizes the compliant controller gains and reference trajectories to maximize the performance of an assembly task. The proposed contribution in this paper differs from the works mentioned above since it is tailored towards vision-based robotics applications, where Bayesian Optimization has not been used before. In addition, a complete pipeline is proposed in order to integrate such optimization framework into a real industrial task, making it possible to acquire, process and manage the sensory data to enhance the capabilities of a robotized system.

To the best knowledge of the authors, no previous work developed a complete pipeline addressing the optimization of the (robot end-effector mounted) sensor pose for object detection performance maximization. As detailed in Section 1.2, existing works only deal with the sub-problems constituting the full application described in this paper. Therefore, no comparison with existing methodologies is provided. The grid point sampling method has been indeed used as a comparison algorithm in order to prove the improved performance of the BO-based pipeline, capable of limiting the optimization iterations (i.e., saving setup time and computational resources).

1.4 Paper Outline

The paper is structured as follows: Section 2 describes the proposed pipeline for an optimal sensor pose definition, providing the details of all the defined components; Section 3 describes the application of the Bayesian Optimization; Section 5 describes the controlled-environment case-study for the validation of the proposed approach; Section 6 describes the real conditions case-study for the validation of the proposed approach; Section 7 describes the pipeline experiment employing the digital twin of the operating scene instead of its 3D reconstruction; Section 8 states the conclusions.

2 Methodology

Given the proposed context in Section 1.1, the aim of this paper is to propose a complete pipeline for the optimization of the (robot end-effector mounted) camera pose allowing to maximize the target object detection performance, while guaranteeing the pose reachability w.r.t. both the robot kinematics and the collisions constraints within the reconstructed operating environment. To do that, the following pipeline is proposed. Five main blocks composing the pipeline are identified. A *scene reconstruction* block is defined, to acquire the point cloud for the modeling of an operating environment. This point cloud will be used to perform the object detection inside the operating scene. A *collision modeling* block is used to define the environmental constraints. The environmental constraints will be used to compute both the collisions-free optimized camera pose and the collisions-free robot motion. An *object detection algorithm* block is defined, to perform the detection of the target object inside the working scene. On the basis of the target part CAD (in .stl format) file and exploiting the point cloud available from the *scene reconstruction* block, the adopted algorithm gives as an output an index quantifying the performance of the object detection procedure. This index is then used to optimize the camera pose inside the operating scene. A *camera pose optimization* block is used to optimize the camera positioning inside the working scene, maximizing the object detection performance while guaranteeing the reachability of the target pose of the sensor. The object detection index from

the *object detection algorithm* block is exploited to perform this optimization. A *collision-free motion planner* block is defined, to plan and execute the robot motion, positioning the camera in the optimized pose. The optimized camera pose from the *object detection algorithm* block is exploited, together with the environmental constraints from the *collision modeling* block.

The *scene reconstruction*, *collision modeling*, *object detection algorithm*, and the *camera pose optimization* blocks run offline in a simulation environment, recreated on the basis of the acquired data from the working environment. In this way, it is possible to consider the real operating conditions and exploit high-fidelity simulations to optimize the target object detection task. The proposed approach allows to avoid any risk (e.g., collisions) related to the operation of a real robot, eliminating the need of experimental trails for the camera pose optimization purposes (i.e., minimizing the optimization time). The optimized camera pose is then transferred to the real robot. The *collision-free motion planner* runs online in order to reach the target camera pose, executing the object detection task exploiting the *object detection algorithm* block (this time, executed online).

In the following, each component of the pipeline is described in details.

2.1 Scene Reconstruction

The *scene reconstruction* block is used in order to create the operating scenario in which the robot is operating. This modeling is then used by the *collision modeling* block and by the *object detection algorithm* block in order to avoid robot-environment collisions and perform the target object detection, respectively.

Reconstructing the operating scene by means of sensory data is required in many real industrial applications [41]. In fact, a complete and representative CAD modeling of the target working scene is not always available. Therefore, an algorithm able to reproduce the robot's operating environment is required. For this purpose, multiple scans of the operating scene are acquired and combined using the vision system mounted at the robot's end-effector in order to recompose the working environment.

The positions in which to acquire such scans can be selected manually or on the basis of a pre-defined path. On the one hand, the robot is manually operated by the human operator, who decides the best poses and orientations for the scans acquisitions. On the other hand, a pre-defined robot motion can be automatically computed to define a path around the target part to be detected. The position of the target parts is usually (roughly) known in industrial settings (such as in grasping applications or assembly tasks [22]). For both the cases, the point cloud is acquired for each scan position, in the base reference frame of the robot.

Once all the scans are acquired, the main issue to resolve is related to the management of overlapping point clouds. In fact, the overlapping might result in elaboration difficulties for the *object detection algorithm* block. To solve this issue, the combined point clouds are elaborated in order to eliminate the overlaps by defining the size of a nominal cube (i.e., cube's side length) in the point cloud space. The point cloud space is discretized using the defined nominal cube. For each of the discretized cubes, the point clouds are merged into a single point cloud by averaging their data on the basis of the distance between the acquisition points. It should be noted that the nominal cube size affects the accuracy of the reconstruction of the point cloud and the computational efficiency: increasing the cube size, the computational resources and the reconstruction accuracy decreases; decreasing the cube size, on the other hand, results in higher computational resources and an increased reconstruction accuracy. Therefore, a trade-off between required computational resources and reconstruction accuracy needs to be found on the basis of the available hardware (i.e., available sensor, its characteristics such as resolution), available detection algorithm performance, target part to be detected (e.g., level of details required to properly model it), etc.

2.2 Collision Modeling

The *collision modeling* block is used in order to define the areas of the working environment that are occupied by obstacles, therefore, allowing to model the collision objects. For this purpose, the *Octomap*© library [42] has been employed.

The *Octomap*© library allows to model the work space volume with the collision objects by means of the reconstructed point cloud data (i.e., combining all the acquired point clouds) acquired with the *scene reconstruction* block, thus making it possible to avoid collisions between the robot and the environment using the *collision-free motion planner* block. The *Octomap*© library performs a (tunable) down sampling of the reconstructed point cloud, and for every point in the data creates a cube (with tunable size) for the occupied space volume, resulting in the collision objects.

2.3 Object Detection Algorithm

The *object detection algorithm* block is used to detect the target part inside the operating environment, exploiting the reconstructed point cloud data provided by the *scene reconstruction* block.

Making use of the *MVTec*© HALCON library, `find_surface_model` allows to achieve the best match for a target part in a 3D working environment, employing both its CAD model and the acquired point cloud describing the real operating scene. `find_surface_model` is based on the three main components:

- *approximate matching*: the poses of the approximate surface model inside the scene are searched. Firstly, scene points are sampled uniformly, controlling the sampling distance through the parameter *RelSamplingDistance*. Then, a set of keypoints is selected from the sampled scene points with the parameter *KeyPointFraction*. For each selected keypoint, the optimum pose of the surface model is computed under the assumption that the keypoint lies on the surface of the object. This is done by pairing the keypoints with all the other sampled scene points, and finding the point pairs on the surface model that have a similar distance and orientation. The sampled scene pose for which the largest number of points lies on the object is considered to be the best pose for this keypoint. The number of sampled scene points on the object is considered to be the score of the pose. From all keypoints, the poses with the best scores are selected and used as approximate poses;
- *sparse pose refinement*: the approximated poses are further refined. This increases their accuracy and the significance of the matching score values. The sparse pose refinement uses the sampled scene points from the approximate matching. The pose is optimized such that the distances from the sampled scene points to the plane of the closest model point are minimal. The plane of each model point is defined as the plane perpendicular to its normal. Since each keypoint produces one pose candidate, the total number of pose candidates to be optimized is proportional to the number of keypoints. The score of each pose is recomputed after the sparse pose refinement;
- *dense pose refinement*: accurately refines the previously founded poses. The distances between the scene points and the planes of the closest model points are minimized, similarly to the sparse pose refinement component. The difference is that only the poses with the best scores from the sparse pose refinement are refined and all points from the original scene are used for the refinement. Taking all points from the scene increases the accuracy, however requires more computational resources.

The output matching score (between the CAD model and the acquired point cloud) quantifying the part detection performance is in a range between 0 (no match) and 1 (perfect match). The matching accuracy is affected by a lot of phenomena, e.g., the shape of the model, the number of points in the scene, the measurement noise, the visibility of the part, etc. The complete description of the employed `find_surface_model` function for object detection purposes is available at the following link: https://www.mvtec.com/doc/halcon/12/en/find_surface_model.html, while an extensive evaluation of its performance is available in [43]. The matching score will be used by the *camera pose optimization* block in order to perform the maximization of the object detection algorithm performance.

2.4 Camera Pose Optimization

The *camera pose optimization* block is used in order to optimize the camera pose maximizing the object detection performance (i.e., the matching score as described in Section 2.3).

The proposed optimization algorithm exploits Bayesian Optimization [32], defining a cost function J with following input quantities: the matching score, and the reachability index (considering both kinematics constraints and the presence of collisions in the target camera pose). The output of the BO is (for each iteration) the camera pose to be evaluated w.r.t. the above defined quantities. For each optimization iteration, the camera is positioned (inside the reconstructed operating scene) based on the BO-suggested pose. The matching score and the reachability index, are computed to calculate the related cost function J value for each optimization iteration, guiding the evolution of an optimization algorithm.

The complete optimization algorithm and the procedure are thoroughly described in the next Section 3.

2.5 Collision-free Motion Planner

The *collision-free motion planner* block is used in order to achieve the optimized camera pose in the working environment. This block makes use of the output of the *camera pose optimization* block (i.e., the optimized camera pose), along with the information from the *scene reconstruction* block and the *collision modeling* block. In this way, it is possible to program the robot motion capable of reaching the target pose of the camera without colliding with the surrounding environment.

The reconstruction of the operating scene, and the collisions modeling (output of the blocks described in Sections 2.1 and 2.2, respectively) are necessary information to be used in this block.

2.6 Complete Workflow

The complete workflow of the proposed methodology is schematized in Fig. 1. The offline part of the methodology is used in the simulation environment to avoid extra experiments with the real robot, while the online part is used with the real robot, being necessary to perform the target task. The offline part of the methodology (including the scene reconstruction, the collision modeling, the object detection algorithm, and the camera pose optimization) is highlighted,

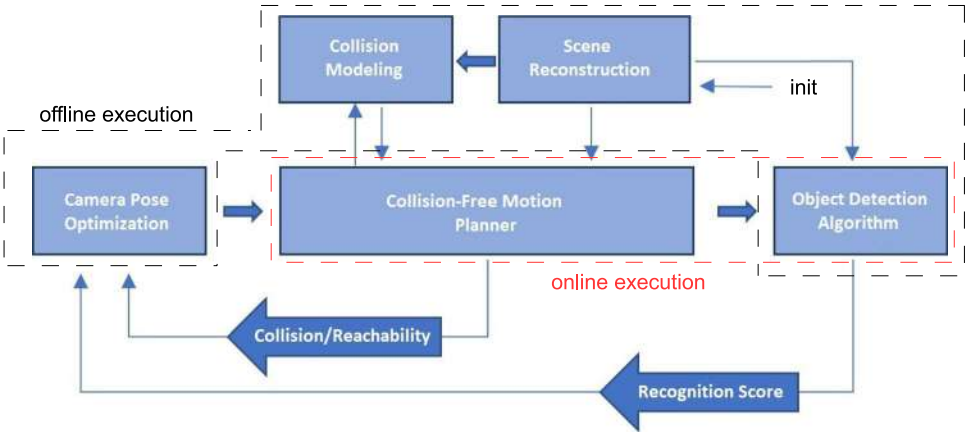

Fig. 1 The complete workflow of the proposed framework is shown, highlighting the connections between all the defined blocks. The approach starts with the initialization of the methodology (*init* arrow). The offline part of the methodology (including the scene reconstruction, the collision modeling, the object detection algorithm, and the camera pose optimization) is highlighted, alongside the online part (including the object detection algorithm and the collision-free motion planner)

alongside the online part (including the object detection algorithm and the collision-free motion planner).

Remark 1 It should be underlined that, if required, the *scene reconstruction* and the *collision modeling* blocks can run again before the *collision-free motion planner* execution to verify the operating conditions. Such update in the scene reconstruction block could be required if any change in the operating scene happens (e.g., the robot operates in a dynamic environment).

3 Camera Pose Optimization with Bayesian Optimization

The main algorithmic contribution of this paper is the use of the Bayesian Optimization to maximize the performance of the object detection algorithm. In fact, in order to achieve a good object detection (i.e., the highest matching score as described in Section 2.3) it is necessary to properly position the vision system (mounted at the robot end-effector) w.r.t. the target part.

3.1 Camera Pose Definition in the Operating Scene

The camera pose (i.e., the robot end-effector pose) affects the performance of the object detection algorithm. The camera needs to point to the target object in order to make it visible and perform the detection. Commonly, the position of the target object is (roughly) known in the industrial applications [22]). Therefore, it is possible to define the camera orientation to point directly to the object. In addition, on the basis of the adopted vision system, the field of view of the sensor needs to be considered, so that the target part is positioned in its range of visibility (i.e., not very near or far from the sensor).

On the basis of these considerations, admissible camera poses are generated in the reconstructed operating scene to lie on the surface of a sphere centered at the (expected) target position of an object, with a radius r (i.e., the distance between the sensor and the target part); thus maximizing the performance of the adopted sensor. The orientation of each pose is defined such that the camera points towards the center of the sphere (i.e., to the target part). In this way, two angles ϕ and θ (polar and azimuth angles, respectively) can be used as independent variables in order to describe the position on the given sphere surface (spherical coordinates, Fig. 2).

The angles ϕ and θ then define the optimization variables vector $\mathbf{x}_{opt} = [\phi, \ \theta]^T$, to be optimized by the Bayesian Optimization in order to maximize the detection of the part, i.e., the matching score described in Section 2.3.

Remark 2 It should be underlined that, by adopting the proposed spherical definition of the admissible camera poses, the number of optimization variables is reduced to 2 (from 6, that is the number of degrees of freedom describing a 3D position/orientation of object in the space). In addition, the camera is always pointing towards the target part, therefore, the camera poses not capable of detecting the part are reduced to the ones in which occlusions/bad visibility is present.

3.2 Cost Function

In order to optimize the camera pose on the proposed nominal sphere described in the Section 3.1 (i.e., optimize the polar and azimuth angles θ and ϕ) for maximizing the part detection performance (i.e., the matching score described in Section 2.3), a cost function J guiding the optimization needs to be defined. In this paper, the following cost function is proposed:

$$J = K_s(S-1) + K_r(S - \min(S, S_r)) - K_k K - K_c C - K_p(\max(S, S_p) - S). \quad (1)$$

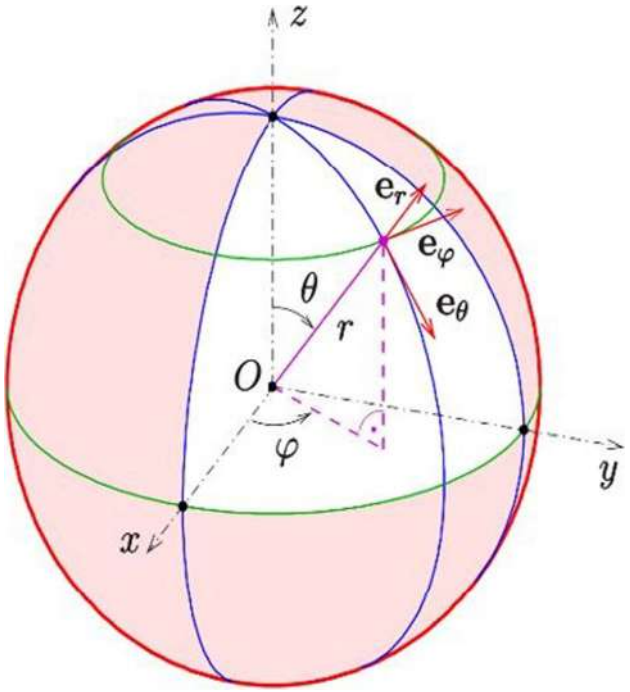

Fig. 2 Admissible camera poses lay on the reference sphere with radius r maximizing the field of view of the sensor and angles ϕ and θ defining the specific position pointing to the sphere center

The cost function (to be maximized, with values in the range $]-\infty,\ K_r(1-S_r)]$) is composed of the reward terms and the penalty terms. W.r.t. the reward terms, the following components can be identified: $K_s(S-1)$ and $K_r(S-\min(S,S_r))$. S is the matching score described in Section 2.3 with values between 0 (no match) and 1 (perfect match), K_s is the reward gain, S_r defines a threshold for the matching score S to achieve an additional reward, and K_r is the additional reward gain. $K_s(S-1)$ is an absolute reward based on the matching score S. $K_r(S-\min(S,S_r))$ is a relative reward that is enabled if $S-\min(S,S_r)>0$, i.e., $S>S_r$, that is the matching score S is over the defined threshold S_r. W.r.t. the penalty terms, the following components can be identified: K_kK, K_cC, and $K_p(\max(S,S_p)-S)$. K is the reachability index defining if the target position on the sphere is kinematically reachable having binary values 0 (target position reachable) or 1 (target position not reachable), K_k is the reachability penalty gain, C is the collisions penalty index defining if collisions are present at the target position on the sphere having binary values 0 (no collisions are present) or 1 (collisions are present), K_c is the collisions penalty gain, S_p defines the threshold for the matching score S to produce an additional penalty, and K_p is the penalty gain associated with low matching scores. K_kK is an absolute penalty related to the reachability of the target pose on the sphere that exploits the motion planner described in Section 2.5. K_cC is an absolute penalty related to the presence of collisions in the target pose on the sphere that exploits the collisions modeling described in Section 2.2. $K_p(\max(S,S_p)-S)$ is a relative penalty that is enabled if $\max(S,S_p)-S>0$, i.e., $S_p>S$, that is the matching score S is lower than the defined threshold S_p.

The cost function J in (1) is therefore capable of guiding the optimization in order to achieve the reachable and collisions-free camera pose maximizing the matching score, i.e., the object detection performance.

Remark 3 The complete workflow shown in Fig. 1 provides all the connections between the different blocks. To perform the described optimization, it should be noted that the collisions modeling, the robotic platform kinematics, and the matching score are implemented and evaluated for each iteration considering the reconstruction of the real environment. The proposed approach, therefore, integrates all the required tools to ensure the optimized task execution on a manipulator in a real application.

3.3 Bayesian Optimization

The cost function J (1) in Section 3.2 defines a metric to tune the camera pose parameters $\mathbf{x}_{opt}=[\phi,\ \theta]^T$.

Collecting all the design parameters in a vector $\boldsymbol{\gamma}$, the tuning task then reduces to the maximization of the cost $J(\boldsymbol{\gamma})$ with respect to $\boldsymbol{\gamma}$, within a space of admissible values $\boldsymbol{\Gamma}$. However, a closed-form expression of the cost J as a function of the design parameter vector $\boldsymbol{\gamma}$ is not available. Furthermore, this cost cannot be evaluated through numerical simulations as the robot dynamics are assumed to be partially unknown. Instead, it is possible to perform experiments on the robot and measure the cost J_i achieved for a given controller parameter vector $\boldsymbol{\gamma}_i$, and thus run an optimization algorithm driven by measurements of J. The peculiar nature of the optimization problem at hand restricts the class of applicable optimization algorithms. Indeed,

- (*i*) the measured cost J_i consists of a noisy observation of the "true" cost function, namely $J_i=J(\boldsymbol{\gamma}_i)+n_i$, with n_i denoting measurement noise and possibly intrinsic process variability;
- (*ii*) no derivative information is available;
- (*iii*) there is no guarantee that the function $J(\boldsymbol{\gamma})$ is convex;
- (*iv*) function evaluations may require possible costly and time-consuming experiments on the robot.

Features (*i*), (*ii*) and (*iii*) rule out classical gradient-based algorithms and restrict the problem to the class of gradient-free, global optimization algorithms. Within this class of algorithms, *Bayesian optimization* (BO) is generally the most efficient in terms of the number of function evaluations [32] and it is therefore, the most promising approach to deal with (*iv*).

In BO, the cost J is simultaneously learned and optimized by performing sequential experiments on the robot. Specifically, at each iteration i of an algorithm, the experiment is performed for a given controller parameter $\boldsymbol{\gamma}_i$ and the corresponding cost J_i is measured. Then, all the past parameter-cost observations $\mathcal{D}_i = \{(\boldsymbol{\gamma}_1, J_1), (\boldsymbol{\gamma}_2, J_2), \dots, (\boldsymbol{\gamma}_i, J_i)\}$ are processed and a new parameter $\boldsymbol{\gamma}_{i+1}$ is computed for the next experiment according to the approach discussed in the following. Additional details related to the *surrogate model*, *acquisition function*, and *algorithm outline* used for Bayesian Optimization can be found in [40].

Remark 4 As described in the Section 2, the camera pose optimization is performed offline by exploiting the reconstructed working scene. In this way, it is possible to avoid experimental trials and transfer the optimized camera pose to the real robot for the online task execution. The proposed procedure has the advantages of avoiding (possible) dangerous robot motions, while minimizing the time required by the optimization (i.e., the required number of experiments).

4 System Description

In order to test the proposed framework for the camera optimization pose in object detection tasks, a Franka EMIKA Panda robot has been employed as a robotic platform, equipped at its end-effector with an Intel© RealSense D400 (attaching it by means of an *ad hoc* designed flange, Fig. 3).

The hand eye calibration (to relate the camera's reference frame to the reference frame of the robot's end-effector) has been performed exploiting the `easy_handeye` ROS package [44] and apriltags [45].

For easier parameters setting during the experimental tests execution, a graphic user interface (GUI) has been implemented. Exploiting the proposed GUI it is possible to setup the optimization within the considered operating environment, and to verify the correct setup of the nominal sphere that defines the admissible camera poses; thus ensuring the correct reconstruction of the operating scene. The proposed GUI allows to define the expected part location in order to initialize the optimization with the positioning of the nominal sphere for the definition of the camera poses (i.e., setting the center of such nominal sphere). Figure 4 shows the implemented GUI.

The object dection has been performed making use of the *MVTec*© HALCON library, in particular exploiting the `find_surface_model` function. It has to be noted that other (open-source) software can be implemented into the *object detection algorithm* block. In particular, the Point Pair Feature algorithm [46, 47] has been also extensively tested in the authors' master thesis [48, 49]. However, the Surface Matching libraries from *MVTec*© HALCON provided better performance w.r.t. the Point Pair Feature algorithm. Therefore, only the Surface Matching libraries from *MVTec*© HALCON are employed in this work.

Fig. 3 Franka EMIKA panda robot's end-effector equipped with the Intel© RealSense D400 sensor

The collision-free motion planner has been implemented making use of the ROS Moveit! tool [50].

The proposed optimization procedure described in Section 3 has been compared with the grid point approach. The nominal sphere defining the admissible camera poses is discretized and all the resulting poses are evaluated. In this way, it is possible to compute the cost function J in (1) for the complete sphere to identify the optimal camera pose (the one that maximizes the matching score while satisfying the constraints on reachability and collisions). The achieved results are used in order to evaluate the optimal camera pose obtained by the proposed approach in Section 3. The discretization in Table 1 (resulting in 400 points) has been employed for the sphere gridding.

Remark 5 It is important to underline that the discretization step of the grid point approach affects the results of the approach. In order to correctly model the cost function J, the discretization step has to be small enough to capture its changes.

The Bayesian Optimization has been implemented exploiting the `limbo` library [51], imposing the parameters provided by Table 2 (defined in Section 3).

All the proposed components in Section 2 have been implemented in ROS, in order to provide a flexible framework capable of operating with real robotic systems in industrial tasks. As shown in Fig. 1, all the implemented blocks are interconnected, exchanging information during the optimization procedure and task execution.

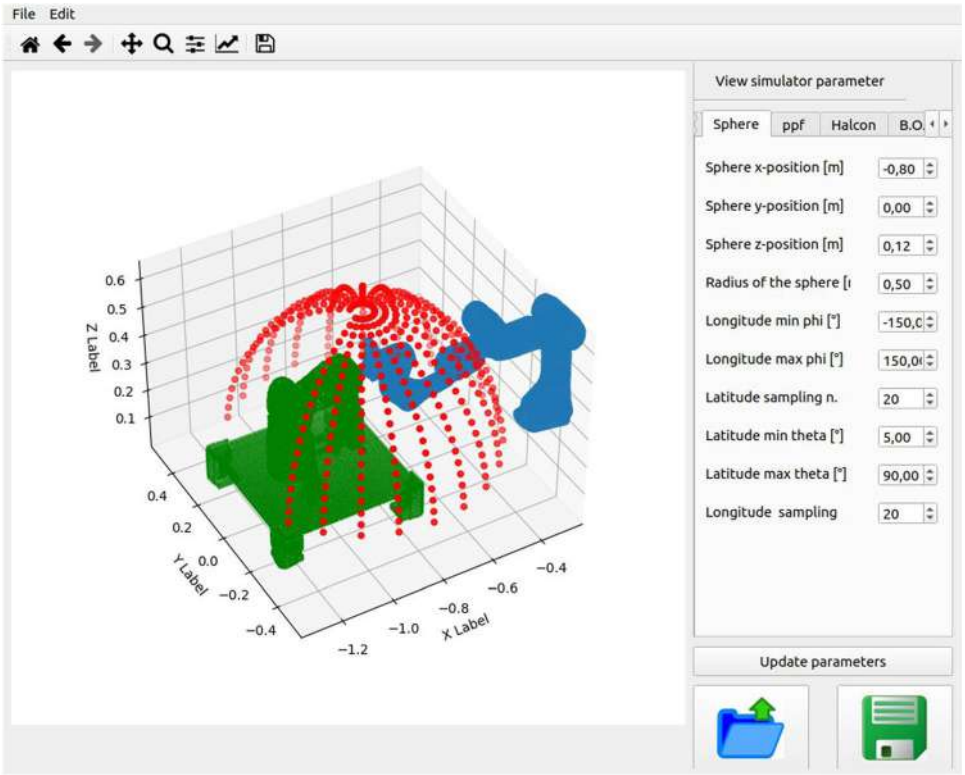

Fig. 4 The implemented GUI is shown. The robot representation (in blue), the reconstructed operating scene (in green), and the camera poses laying on the nominal sphere (in red) are visualized

Remark 6 It should be noted that the maximum number of iterations for the BO have been imposed in order to show sufficient number of iterations. An exiting criteria can be defined to stop the BO as soon as satisfying results are achieved.

The proposed framework has been implemented on a Dell OptiPlex 7070 Mini Tower pc. The employed OS is Ubuntu 20.04 with real-time kernel (version 5.9.1) for the communication between the ROS environment and the Franka EMIKA panda robot.

5 Validation in Controlled Environment

The proposed framework has been validated in a controlled environment, i.e., without the presence of random objects. In this way, a complete analysis of the performance is possible. In the following, the operating scenario (describing both the target part to be detected and the working scene) and the achieved results are detailed.

5.1 Operating Scenario

The target part to be detected is shown in Fig. 5. The proposed controlled environment validation scenario is shown in Fig. 6. The part visibility is occluded by the environment, limiting the admissible camera poses to solve the object detection task.

5.2 Scene Reconstruction

The scene reconstruction is performed using the methodology described in Section 2.1. Figure 7 shows the reconstructed operating environment exploiting the combined point clouds acquired with the Intel© RealSense D400 sensor. As it can be seen, the proposed approach allows to reproduce the target working scene for the purposes of the paper as described in Section 2 (i.e., collision modeling, object detection, camera pose optimization, and collision-free motion planning).

5.3 Optimization Results

Exploiting the grid point approach to discretize the nominal sphere, it is possible to evaluate the performance (i.e., the

Table 1 Employed parameters for sphere gridding optimization (Fig. 2 describes the adopted notation for the sphere parameters)

Parameter	Value	Units
θ_{min}	5	°
θ_{max}	90	°
θ *samples*	20	
ϕ_{min}	-150	°
ϕ_{max}	150	°
ϕ *samples*	20	
r	0.5	m

Table 2 Bayesian Optimization parameters setting

Parameter	Value
K_s	4000
K_r	500
S_r	0.4
K_k	1000
K_c	1000
K_p	2000
S_p	0.2
Initial random samples	10
Max BO iterations	50

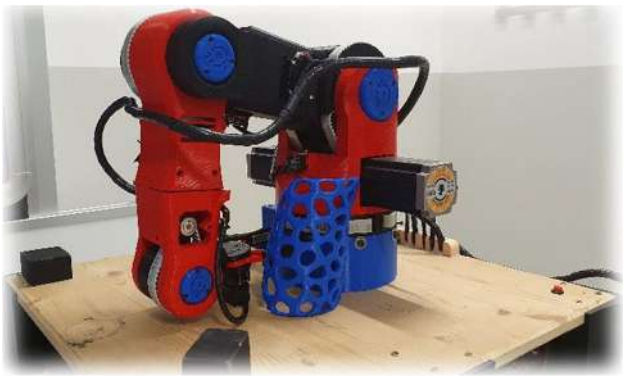

Fig. 6 Controlled environment validation scenario

matching score) of the object detection algorithm described in Section 2.3. Figure 8 shows the matching score S values as a function of the ϕ and θ angles. As it can be seen, the performance of the object detection algorithm vary with the pose of the camera. Therefore, optimizing the pose of the sensor is fundamental in order to properly detect the target part. The matching score information are indeed used to compute the cost function J (1). The Figure also shows the achieved optimized camera pose for both the grid point approach (green dot) and BO approach (yellow dot). As it is highlighted in the Figure, these two positions are in the same region with comparable matching scores, proving that the proposed approach is capable of achieving the optimal camera pose in a limited number of iterations (50 iterations are performed with the BO approach, while 400 iterations are performed with the grid point approach). All the BO iterations are shown in the Figure, to underline the behavior of the proposed approach.

Figure 9 allows to better visualize the evaluated camera poses for both the grid point approach and BO approach. As it can be seen, the grid point approach requires to exploring the complete sphere in order to find the optimal camera pose, while the BO approach requires less experiments to find the optimized camera pose. The Figure highlights that the BO approach selects the camera poses that tend to concentrate in the optimal region after some exploration is performed.

Figure 10 shows the reachable (marked in green) and unreachable (marked in red) poses on the nominal sphere. Unreachability is related to kinematics limits and/or collisions. Reachability needs to be taken into account in order to compute the optimized camera pose that conforms to the execution with the real robot. Reachability information are indeed used to compute the cost function J (1).

Figure 11 shows the cost function J (1) (to be maximized) values as a function of the ϕ and θ angles (exploiting both matching score and reachability data for the evaluated camera poses). As it can be seen, the camera poses

Fig. 5 CAD modeling and real part to be detected in the proposed controlled environment

Fig. 7 Reconstructed operating scenario

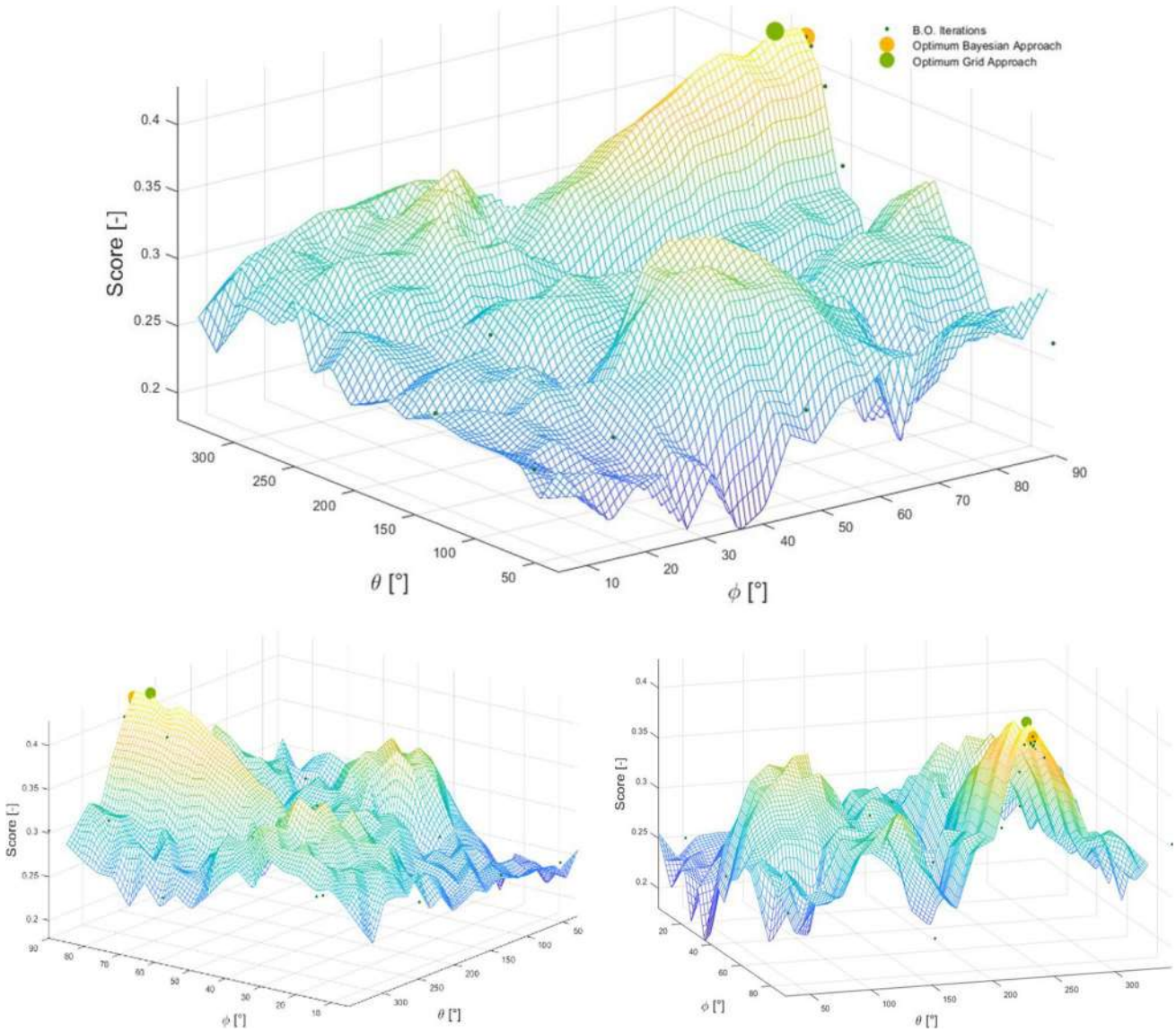

Fig. 8 Matching score S values as a function of the ϕ and θ variables for the object detection algorithm performance evaluation. The values related to the optimal camera pose for both the grid point approach (green dot) and the BO approach (yellow dot) are highlighted, alongside all the iterations performed by the BO approach

that are not reachable/presenting collisions are penalized. The optimized cost function value for both the grid point approach (green dot) and BO approach (yellow dot) are highlighted, alongside the cost function values related to all the BO iterations.

Remark 7 It should be underlined that, as explained in Section 4, the complete framework shown in Fig. 1 has been implemented in ROS, with intercommunication happening between the components. The proposed optimization, therefore, exploits the complete information available from all the other components.

5.4 Task Execution

Once the optimization of the camera pose is performed (offline), the robot motion can be planned (online) in order to execute the target object detection task, considering the collisions modeling to achieve a collision-free motion. The *Octomap*© library is exploited to define the collision objects as described in Section 2.2. Figure 12 shows the reconstructed environment, including the collision modeling. Exploiting Moveit!, it is then possible to plan the robot motion to reach the optimized camera pose for executing the object detection task.

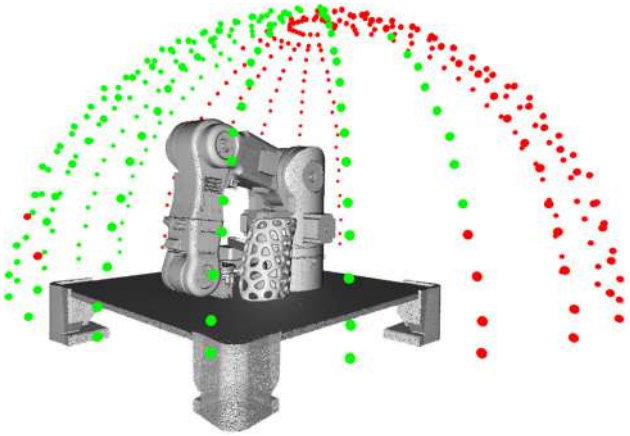

Fig. 10 Graphical representation of the reachability (considering both kinematics constraints and collisions). Reachable poses are marked in green, while unreachable poses are marked in red

The offline computed (i.e., exploiting the reconstructed operating scene) matching score S can be compared with the online one (i.e., in the real operating scene, after the camera is positioned in its optimized pose). The offline matching score is 0.552, while the online matching score is 0.545. The online matching score is slightly lower than

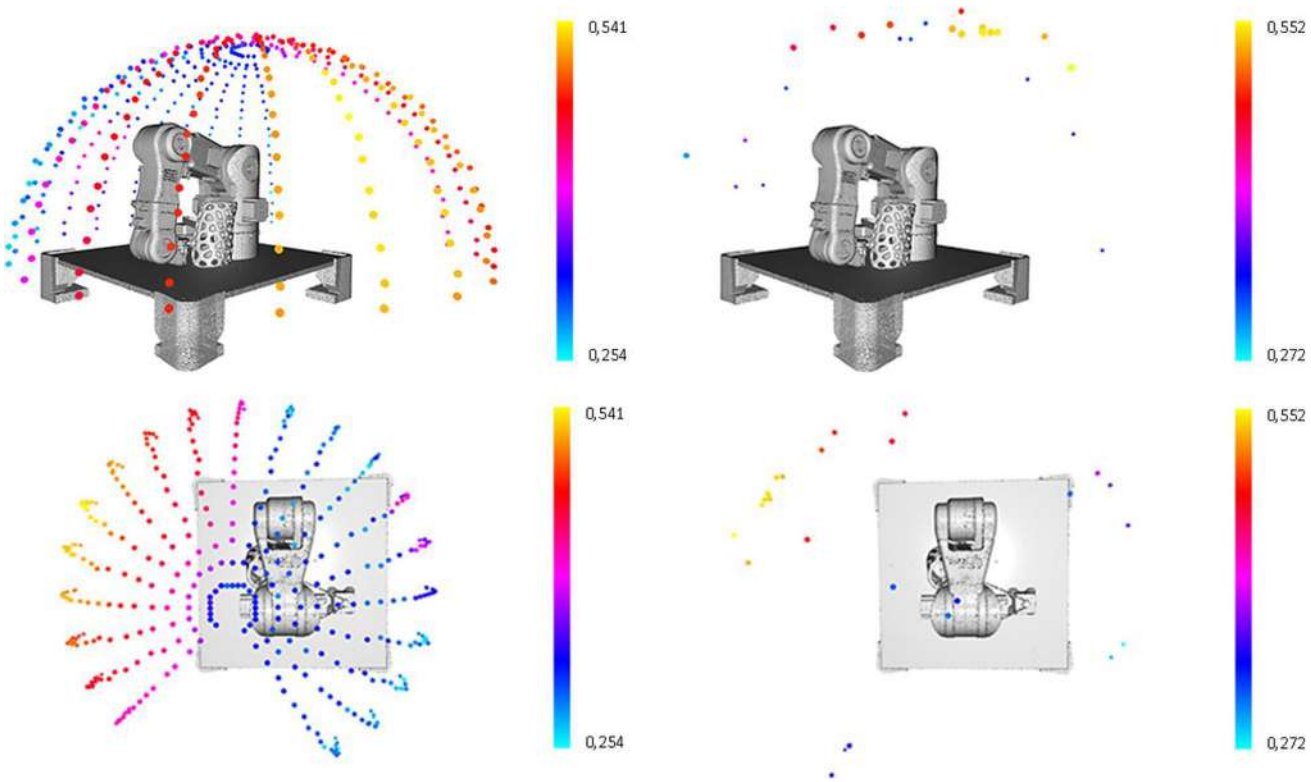

Fig. 9 Graphical representation of the matching score S for the evaluated camera poses exploiting the grid point approach (left), and the BO approach (right)

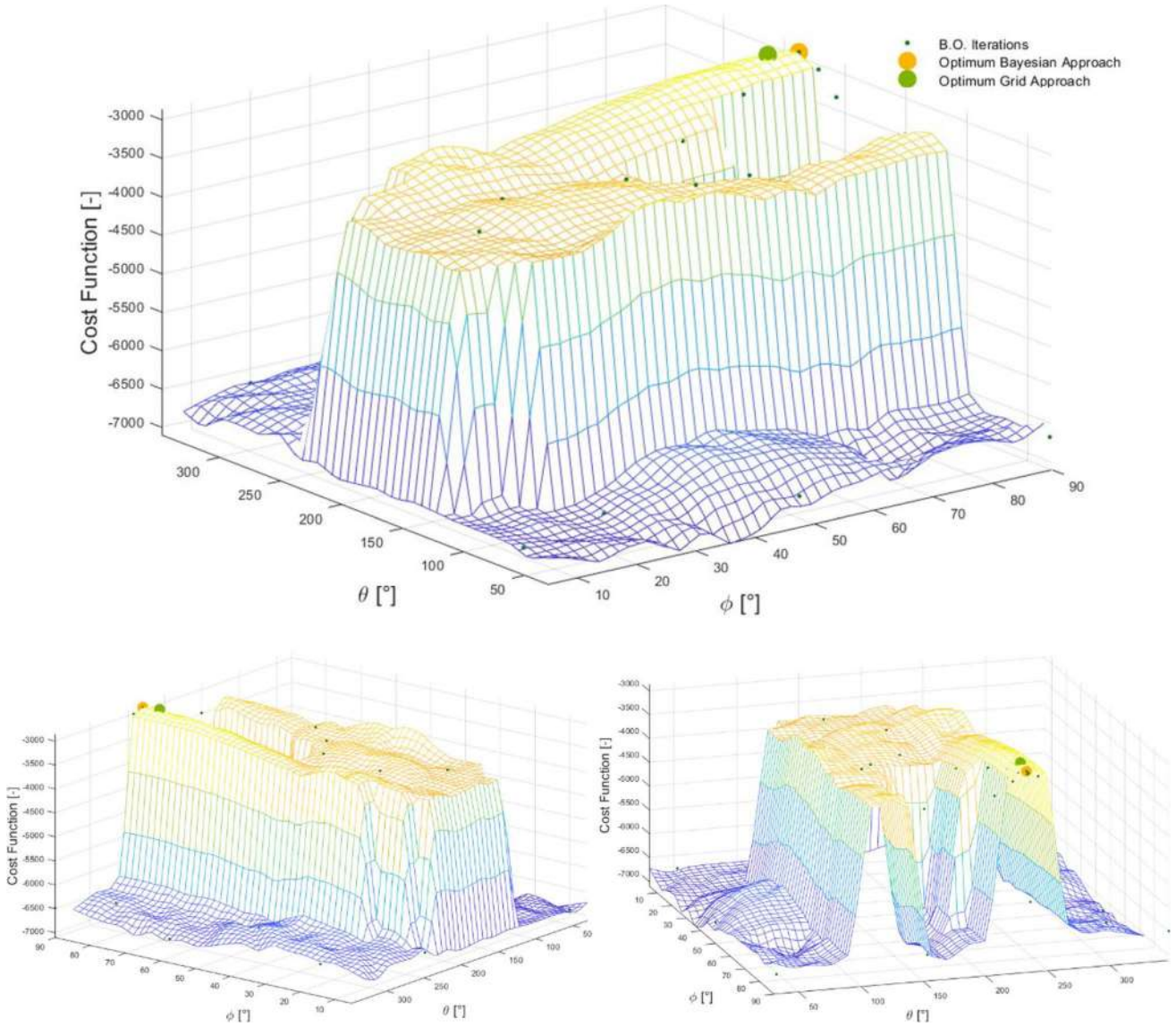

Fig. 11 Cost function J values as a function of the ϕ and θ variables. The values related to the optimal cost function J for both the grid point approach (green dot) and the BO approach (yellow dot) are highlighted, alongside all the iterations performed by the BO approach

the offline one. This is related to (very limited) modeling differences (e.g., related to light conditions, noise on the measured point clouds, etc.). However, the deviation of the online matching score w.r.t. the offline matching score is negligible, making the proposed approach effective (for both the operating scene reconstruction and the camera pose optimization).

6 Validation in Real Conditions

In order to prove the effectiveness of the proposed framework in real conditions, an additional test has been executed to consider realistic environments (i.e., higher level of occlusions, higher number of obstacles, presence of random objects). In the following, the proposed operating scenario and the corresponding results achieved are described,

6.1 Operating Scenario

The target part to be detected is a driller, as shown in Fig. 13. The proposed real environment validation scenario is shown in Fig. 14. The part is inside a toolbox, with many other random parts positioned around. Upper toolbox drawers are open, so that a higher-level of occlusion is achieved (i.e., only a small portion of the nominal sphere defining the admissible camera poses allows to properly detect the target part). Therefore, the optimization of the camera pose is highly important to execute the target object detection task.

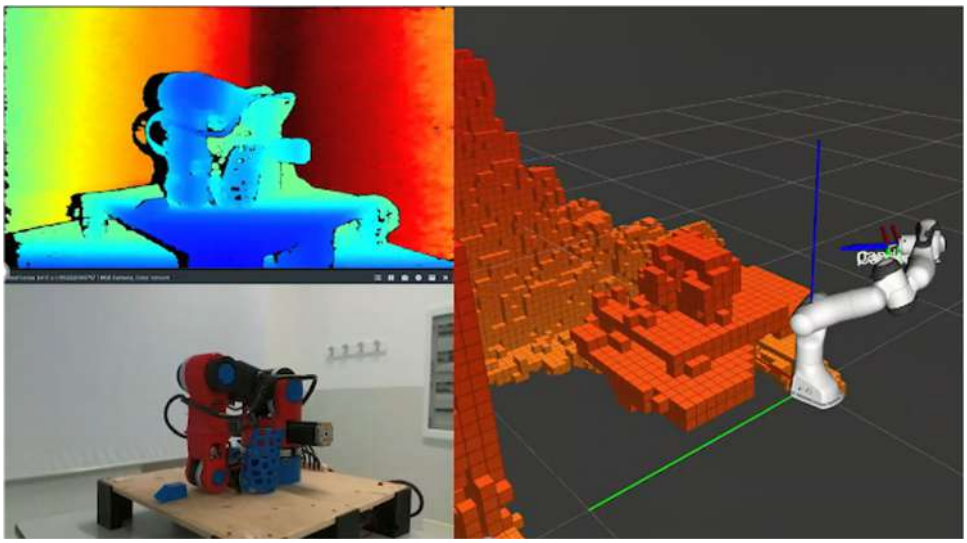

Fig. 12 Collision modeling reconstruction exploiting the *Octomap*© library

6.2 Results

The achieved results are shown in the video available at https://www.youtube.com/watch?v=B9KXQ2wixrY&t=12s. In the presented video, the complete procedure described in the Section 2 and the task execution are shown.

The proposed framework allows to model the operating scene, reconstructing the collision objects, exploiting such information in order to optimize (offline) the camera pose for the object detection task and to execute (online) the collision-free robot motion to the goal position.

Figure 15 shows the robot executing the object detection in the real operating environment, exploiting the optimized camera pose (based on the BO approach).

To demonstrate the increased performance achieved by the object detection algorithm exploiting the proposed camera pose optimization methodology, the a-type uncertainty on the estimated object pose (i.e., on both translational and rotational degrees of freedom - DOFs) has been computed and compared against the results obtained with non-optimized camera pose-based object detection and the object detection performance achieved during the operating scene reconstruction phase (i.e., exploiting the acquired data used for the working scene reconstruction). Table 3 shows the achieved results, highlighting that the optimized camera pose allows to improve the performance related to the estimation of the target object pose (extremely important in industrial applications, such as inspection or grasping tasks).

Remark 8 The developed open source software, including all the components described in Section 2, is available at the GitHub repository https://github.com/LorisR/BO_best_view.

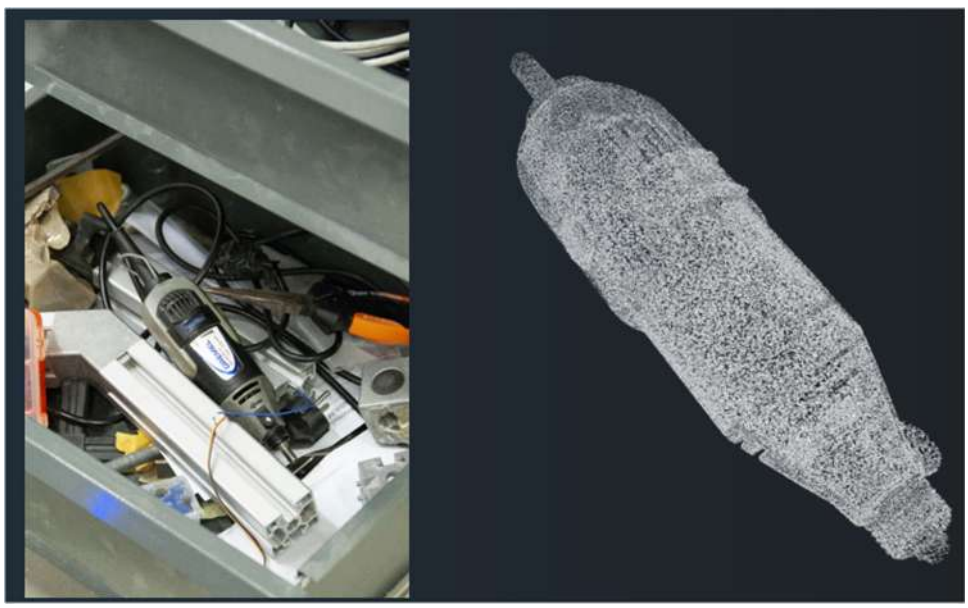

Fig. 13 Real part (driller) to be detected in the proposed real environment

Fig. 14 Real environment validation scenario

7 Employing the Operating Scene Digital Twin in the Pipeline

In this Section, the digital twin of the operating scene is adopted to substitute the working scene reconstruction block. For some applications, in fact, such digital twin of the real working environment is available [21]. Examples of such applications are e.g., quality inspection tasks, in which the installation of the target parts has to be checked. In such cases, the locations of the installed parts are known. A digital twin can, therefore, be generated and used for the optimization purposes addressed in the paper. A real example of such a use-case is shown in the H2020 CS2 ASSASSINN project https://makerfairerome.eu/it/espositori/?edition=2020&exhibit=200113, where the quality of the installation of target components has to be verified inside the aircraft fuselage. By exploiting a digital twin of the operating environment it is possible to reduce the computational resources required by the pipeline, additionally reducing the effect of the noise measurements that affect the acquired point clouds for the reconstruction of the operating environment.

Figure 16 shows the operating environment and its digital twin within the considered testing environment. By comparing Figs. 7 and 16, it can be noted that the measurement noise is eliminated in the operating environment reconstruction. However, it is important to underline that, to be effective, the digital twin needs to represent the target operating environment with high fidelity. If not, the digital twin cannot be used for the purpose of this paper. Figure 17 shows the graphical representation of the matching score S achieved by exploiting the proposed BO-based optimization. As it can be seen, the proposed approach is capable of maximizing the matching score S even when employing the digital twin of the reconstructed operating environment. By transferring the optimized sensor pose to the real manipulator, the achieved object detection performance has been validated, guaranteeing the maximum accuracy in the object localization (i.e., minimizing the measurement uncertainties as described in Section 6).

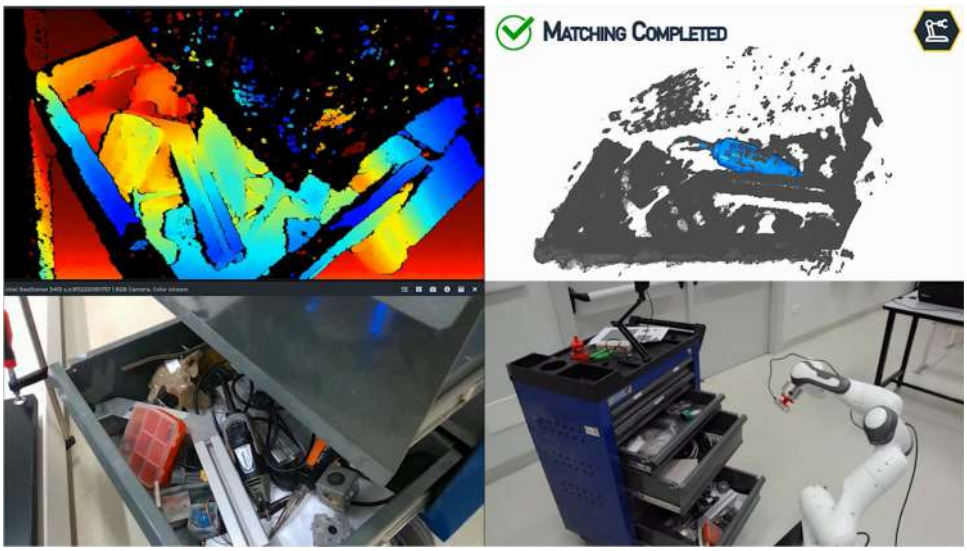

Fig. 15 Object detection task in the real operating environment. On the top-right side of the Figure, the correct detection of the part is highlighted

Table 3 A-type uncertainty for the estimated target object pose exploiting the BO-based camera pose, a non optimized camera pose, and the scene reconstruction camera pose

Camera Pose	Translational Uncertainties [mm]	Rotational Uncertainties [°]
BO-based	0.86	0.088
Not Optimized	6.18	38.02
Reconstruction phase	5.21	32.43

The a-type uncertainty is computed exploiting 20 values for the estimated object pose (for both translations and rotations) and for the three camera poses

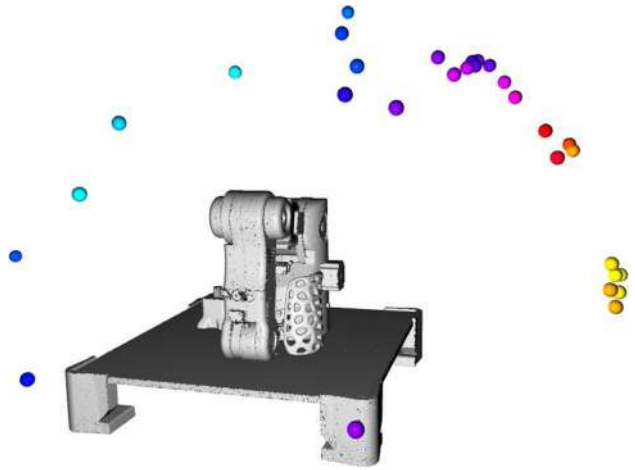

Fig. 17 Graphical representation of the matching score S for the evaluated camera poses exploiting the BO approach. These results are obtained exploiting the digital twin of the operating environment

8 Conclusions

In this paper, a complete pipeline for the optimization of the (robot's end-effector mounted) camera pose for the object detection purposes is proposed. The provided approach relies on the following main components: working scene reconstruction, robot-environment collisions modeling, object detection, sensor pose optimization (exploiting Bayesian Optimization), and collision-free robot motion planning. The proposed components have been integrated into the ROS framework to ensure the flexible application structure, capable of working with the real manipulators in industrial tasks. The main algorithmic contribution of this paper is related to the development of a Bayesian Optimization-based procedure that allows to reduce the number of iterations required to identify the most suitable pose of the camera. For the experimental validation purposes, a Franka EMIKA Panda robot has been employed as a robotic platform, equipped with an Intel© RealSense D400 at its end-effector. Two object detection tasks have been proposed to validate the proposed approach, demonstrating the improved capabilities of the proposed framework. The Bayesian Optimization-based approach has been compared against a grid point approach for the camera pose optimization, highlighting the limited number of optimization iterations required to achieve the same level of performance. In addition, an experiment employing the digital twin of the operating environment has been proposed, highlighting the possibility to use the available modeling of the working environment in the proposed pipeline. As shown in the provided video, the proposed pipeline allows to perform the optimization in a limited time, being suitable for implementation in real industrial applications.

Future work will be devoted to extend the algorithm to simultaneously deal with multiple objects. By integrating the transfer learning capabilities in the Bayesian optimization algorithm, it will be possible to exploit all the available data to optimize the camera pose for each target part, maximizing the detection performance for all the components separately. The centering of the nominal sphere (i.e., the 3 x, y, and z coordinates related to the expected positioning of the target object) that defines the camera poses will be also taken into consideration for an optimization (i.e., with 3 more optimization variables). This could give higher flexibility to the pipeline (i.e., by relaxing an assumption on the tolerance related to the initial definition of the target part location). In addition, human-in-the-loop optimization techniques will be considered to reduce the computational resources required by the optimization algorithm. The proposed pipeline employing the available digital twin of the operating scene will be further tested within the H2020 CS2

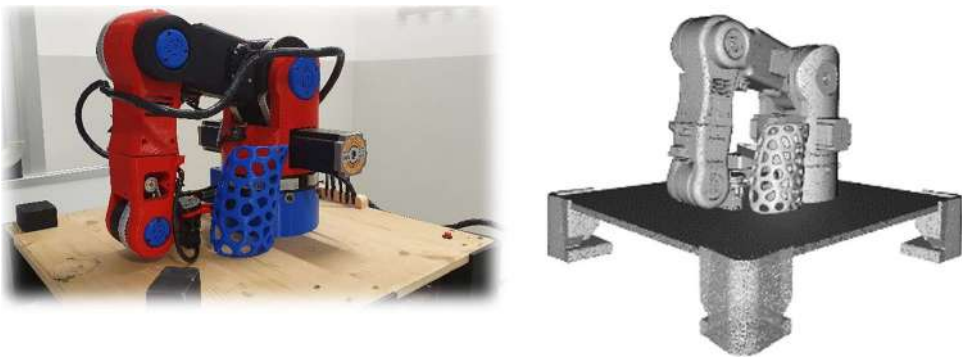

Fig. 16 Real operating scene vs. its digital twin employed instead of the working scene reconstruction block for the offline sensor pose optimization

ASSASSINN project in a real aerospace context by performing a quality inspection task (https://makerfairerome.eu/it/espositori/?edition=2020&exhibit=200113).

Appendix A: Object Detection State of the Art

Many object detection approaches are making use of specific datasets (created either by human annotation or incrementally placing one object in the scene and using foreground masking) for the detection of parts in operating environments [52–54]. Other approaches, instead, are making use of the CAD files of the target parts to be detected. Some of these works use synthetic datasets generated by rendering 3D CAD models of the target objects with different viewpoints, avoiding manual labeling [55]. In this case, an offline training (i.e., in simulation environments) for detection and pose estimation purposes is performed. However, many issues are still present in the proposed methods, making it difficult to transfer the trained behavior from simulation to the real task: modeling differences between the virtual training environment and the real testing scenario (i.e., the training can be not suitable for the target application), the generation of training objects' poses that are not necessarily physically realistic (i.e., increasing the processing time without providing useful information to the algorithm), the presence of occlusions that are usually treated in a simplified manner (i.e., unrealistic scenes resulting in possible failures when moving to the real task) [56, 57]. The possibility to (partially) overcome such issues by exploiting an autonomous process for training a Convolutional Neural Network for object detection and pose estimation has been proposed in [58]. By employing a physics engine to generate synthetic but physically realistic images, the proposed approach makes use of multiple views to perform the object detection and its pose estimation. Other CAD-based approaches make use of the real data acquired from the operating scene to perform the object detection and its pose estimation [59], using both the 2D images [60] or the RGB-D data [61]. In such a context, feature-based methods [62], in which the object detection is based on the use of 3D data, are some of the most popular solutions adopted in many robotic applications [41], that can be divided in two main groups: local feature-based [63] and global feature-based [64] methods. Local features-based approaches are based on matching the descriptors of local surface characteristics, including three main stages [65]: 3D keypoints detection, local surface feature description, and surface matching. The first phase is the most important one, in which a set of points are labelled as keypoints as a function of the exploited detection method (e.g., surface sparse sampling, mesh decimation, fixed-scale, adaptive-scale, etc. [66, 67]). These points will be the ones on which object detection will be based. Once a keypoint has been detected, geometric information of the local surface around the keypoint can be extracted and encoded into a feature descriptor. According to the approaches employed to construct the features descriptors, it is possible to classify the existing methods into three broad categories [68]: signature based, histogram based, and transform based methods. Finally, the surface matching step establishes a set of feature correspondences between the operating scene and the target model, by matching the scene features against the model features. A comprehensive survey of these existing methods is proposed in [69]. Global feature-based methods, instead, follow a different pipeline for which the whole object surface is described by a single or small set of descriptors. Global point cloud descriptor is described extensively in [70]. Local features-based techniques are more robust considering cluttered environments and partial occlusions, that are frequently present in the real-world applications. Global features-based methods are instead more suitable for model retrieval and 3D shape classification, especially considering the weak geometric structures. Alhamzi et al. [71] describes an approach based on the exploitation of both local features and global features techniques, based on the PCL library [72]. Other approaches have been developed for CAD-based object detection and pose identification purposes. Template matching techniques have been proposed exploiting RGB-D data as in [73], in which a method based on quantized surface normal as depth cue is proposed. In a similar way, recently, [74] applied the concept of multimodal matching of [73] on an efficient cascade-style evaluation strategy. Even techniques based on supervised machine learning have been used for object detection and pose estimation exploiting RGB-D data. In [75] a review of classification techniques used in supervised machine learning is described, explaining how the goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features and the different classification techniques. One of the methods to perform the pose estimation is represented by a deep learning approach to category-level 3D object pose tracking on RGB-D data with the use of key points [76]. This algorithm tracks novel object instances of known object categories such as bowls, laptops, and mugs in real time. It learns to compactly represent an object by a handful of 3D key points, based on which the inter frame motion of an object instance can be estimated through key point matching. Another algorithm for real-time 6 DOF pose estimation and tracking of rigid 3D objects uses a monocular RGB camera [77]. The key idea is to derive a region-based cost function using temporally consistent local color histograms. While such region-based cost functions are commonly optimized using first-order gradient descent techniques, in this paper a Gauss-Newton optimization scheme is proposed, which gives rise to drastically faster convergence and highly accurate and robust tracking performance. In numerous preliminary experiments performed by the authors [48, 49], it has been demonstrated that the proposed Gauss-Newton approach outperforms existing approaches in

the presence of cluttered backgrounds, heterogeneous objects and partial occlusions. HybridPose [78], instead, leverages on multiple intermediate representations to express the geometric information in the input image for pose estimation. In addition to key points, this type of algorithm integrates a prediction network that outputs edge vectors between adjacent key points. As most objects possess a partial reflection symmetry, HybridPose also utilizes predicted dense pixel-wise correspondences that reflect the underlying symmetric relations between pixels. Another work demonstrated that Neural Networks coupled with a local voting-based approach can be used to perform reliable 3D object detection and pose estimation in a cluttered environment showing occlusions [79].

Author Contributions Methodology: L. Roveda, M. Maroni, L. Mazzuchelli, L. Praolini; software implementation: M. Maroni, L. Mazzuchelli, L. Praolini, L. Roveda; experimental tests: M. Maroni, L. Mazzuchelli, L. Praolini, L. Roveda; work supervision: G. Bucca, D. Piga; funds acquisition: L. Roveda; paper editing: L. Roveda, A. A. Shahid, M. Maroni, L. Mazzuchelli, L. Praolini.

Funding The work has been developed within the project ASSASSINN, funded from H2020 CleanSky 2 under grant agreement n. 886977.

Data Availability Statement Open source code is available at https://github.com/LorisR/BO_best_view, providing all the developed components explained in the paper.

Declarations

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent to publish Authors consent to publish the here presented work.

Competing interests Not applicable.

References

1. Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., Hoffmann, M.: Industry 4.0. Business & Information Systems Engineering **6**(4), 239–242 (2014)
2. Roveda, L., Magni, M., Cantoni, M., Piga, D., Bucca, G.: Human-robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via bayesian optimization. Robot. Auton. Syst., pp 103711 (2020)
3. Roveda, L., Maskani, J., Franceschi, P., Abdi, A., Braghin, F., Tosatti, L.M., Pedrocchi, N.: Model-based reinforcement learning variable impedance control for human-robot collaboration. J. Intell. Robot. Syst., pp 1–17 (2020b)
4. Pérez, L., Rodríguez, Í., Rodríguez, N., Usamentiaga, R., García, D.F.: Robot guidance using machine vision techniques in industrial environments: A comparative review. Sensors **16**(3), 335 (2016)
5. Vozel, K.: The details of vision guided robotics. Quality, pp 38–40 (2020)
6. Shamsfakhr, F., Bigham, B.S.: Gsr: geometrical scan registration algorithm for robust and fast robot pose estimation. Assembly Automation (2020)
7. Nerakae, P., Uangpairoj, P., Chamniprasart, K.: Using machine vision for flexible automatic assembly system. Procedia Computer Science **96**, 428–435 (2016)
8. Roveda, L., Castaman, N., Ghidoni, S., Franceschi, P., Boscolo, N., Pagello, E., Pedrocchi, N.: Human-robot cooperative interaction control for the installation of heavy and bulky components. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 339–344. IEEE (2018)
9. Balatti, P., Kanoulas, D., Tsagarakis, N., Ajoudani, A.: A method for autonomous robotic manipulation through exploratory interactions with uncertain environments. Autonomous Robots **44**(8), 1395–1410 (2020)
10. Zhihong, C., Hebin, Z., Yanbo, W., Binyan, L., Yu, L.: A vision-based robotic grasping system using deep learning for garbage sorting. In: 2017 36th Chinese Control Conference (CCC), pp. 11223–11226. IEEE (2017)
11. Frank, D., Chhor, J., Schmitt, R.: Stereo-vision for autonomous industrial inspection robots. In: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2555–2561. IEEE (2017)
12. Militaru, C., Mezei, A.-D., Tamas, L.: Object handling in cluttered indoor environment with a mobile manipulator. In: 2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), pp. 1–6. IEEE, (2016)
13. Kragic, D.: Free space of rigid objects: Caging, path non-existence, and narrow passage detection. In: Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics, vol. 14, p. 19. Springer Nature (2020)
14. Nair, D., Pakdaman, A., Plöger, P.G.: Performance evaluation of low-cost machine vision cameras for image-based grasp verification. arXiv:2003.10167 (2020)
15. Cheng, S., Leng, Z., Cubuk, E.D., Zoph, B., Bai, C., Ngiam, J., Song, Y., Caine, B., Vasudevan, V., Li, C., et al.: Improving 3d object detection through progressive population based augmentation. In: European Conference on Computer Vision, pp. 279–294. Springer (2020)
16. Pi, Y., Nath, N.D., Behzadan, A.H.: Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. Advanced Engineering Informatics **43**, 101009 (2020)
17. Arnold, E., Al-Jarrah, O.Y., Dianati, M., Fallah, S., Oxtoby, D., Mouzakitis, A.: A survey on 3d object detection methods for autonomous driving applications. IEEE Transactions on Intelligent Transportation Systems **20**(10), 3782–3795 (2019)
18. Zou, Z., Shi, Z., Guo, Y., Ye, J. Object detection in 20 years: A survey. arXiv:1905.05055 (2019)
19. Du, G., Wang, K., Lian, S.: Vision-based robotic grasping from object localization, pose estimation, grasp detection to motion planning: A review. arXiv:1905.06658 (2019)
20. Chen, J., Zhang, L., Liu, Y., Xu, C.: Survey on 6d pose estimation of rigid object. In: 2020 39th Chinese Control Conference (CCC), pp. 7440–7445. IEEE (2020)
21. Roveda, L., Ghidoni, S., Cotecchia, S., Pagello, E., Pedrocchi, N.: Eureca h2020 cleansky 2: a multi-robot framework to enhance the fourth industrial revolution in the aerospace industry. In: Robotics and Automation (ICRA), 2017 IEEE Int Conf on, Workshop on Industry of the Future: Collaborative, Connected, Cognitive. Novel approaches stemming from Factory of the Future and Industry 4.0 initiatives (2017)

22. Vicentini, F., Pedrocchi, N., Beschi, M., Giussani, M., Iannacci, N., Magnoni, P., Pellegrinelli, S., Roveda, L., Villagrossi, E., Askarpour, M., et al.: Piros: Cooperative, safe and reconfigurable robotic companion for cnc pallets load/unload stations. In: Bringing Innovative Robotic Technologies from Research Labs to Industrial End-users, pp. 57–96. Springer (2020)
23. Ercan, A.O., Yang, D.B., El Gamal, A., Guibas, L.J.: Optimal placement and selection of camera network nodes for target localization. In: International Conference on Distributed Computing in Sensor Systems, pp. 389–404. Springer (2006)
24. Olague, G., Mohr, R.: Optimal camera placement for accurate reconstruction. Pattern Recognition **35**(4), 927–944 (2002)
25. Chen, S.Y., Li, Y.F.: Automatic sensor placement for model-based robot vision. IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics) **34**(1), 393–408 (2004)
26. Dunn, E., Olague, G.: Pareto optimal camera placement for automated visual inspection. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systemss, pp. 3821–3826. IEEE (2005)
27. McGreavy, C., Kunze, L., Hawes, N.: Next best view planning for object recognition in mobile robotics. CEUR Workshop Proceedings (2017)
28. Iversen, T.M., Kraft, D.: Optimizing sensor placement: A mixture model framework using stable poses and sparsely precomputed pose uncertainty predictions. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6652–6659. IEEE (2018)
29. Mosbach, D., Gospodnetić, P., Rauhut, M., Hamann, B., Hagen, H.: Feature-driven viewpoint placement for model-based surface inspection. Machine Vision and Applications **32**(1), 1–21 (2020)
30. Ajoudani, A., Zanchettin, A.M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., Khatib, O.: Progress and prospects of the human-robot collaboration. Autonomous Robots **42**(5), 957–975 (2018)
31. Pelikan, M., Goldberg, D.E., Cantú-Paz, E., et al.: Boa: The bayesian optimization algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, vol. 1, pp. 525–532. Citeseer (1999)
32. Brochu, E., Cora, V.M., De Freitas, N.: A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv:1012.2599 (2010)
33. Letham, B., Karrer, B., Ottoni, G., Bakshy, E., et al.: Constrained bayesian optimization with noisy experiments. Bayesian Analysis **14**(2), 495–519 (2019)
34. Schleicher, T., Bullinger, A.C.: Assistive robots in highly flexible automotive manufacturing processes. In: Congress of the International Ergonomics Association, pp. 203–215. Springer (2018)
35. Ciszak, O.: Industry 4.0–industrial robots. In: MMS 2018: 3rd EAI International Conference on Management of Manufacturing Systems, pp. 52. European Alliance for Innovation (2018)
36. Cully, A., Clune, J., Tarapore, D., Mouret, J.-B.: Robots that can adapt like animals. Nature **521**(7553), 503 (2015)
37. Drieß, D., Englert, P., Toussaint, M.: Constrained bayesian optimization of combined interaction force/task space controllers for manipulations. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 902–907. IEEE (2017)
38. Yuan, K., Chatzinikolaidis, I., Li, Z.: Bayesian optimization for whole-body control of high degrees of freedom robots through reduction of dimensionality. IEEE Robot. Autom. Lett. (2019)
39. Rozo, L.: Interactive trajectory adaptation through force-guided bayesian optimization. arXiv:1908.07263 (2019)
40. Roveda, L., Forgione, M., Piga, D.: Robot control parameters auto-tuning in trajectory tracking applications. Control Engineering Practice **101**, 104488 (2020)
41. Roveda, L., Castaman, N., Franceschi, P., Ghidoni, S., Pedrocchi, N.: A control framework definition to overcome position/interaction dynamics uncertainties in force-controlled tasks. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 6819–6825. IEEE (2020d)
42. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: An efficient probabilistic 3d mapping framework based on octrees. Autonomous Robots **34**(3), 189–206 (2013)
43. Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al.: Bop: Benchmark for 6d object pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 19–34 (2018)
44. https://github.com/IFL-CAMP/easy_handeye. Last visit in January 2021
45. Wang, J., Olson, E.: Apriltag 2: Efficient and robust fiducial detection. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4193–4198. IEEE (2016)
46. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. (2010)
47. Hinterstoisser, S., Lepetit, V., Rajkumar, N., Konolige, K.: Going further with point pair features. (2016)
48. Maroni, M., Praolini, L.: Best view methodology enhanced by bayesian optimization for robotic motion planning in quality inspection tasks. Master's thesis, Politecnico di Milano (2020)
49. Mazzuchelli, L.: Robotized quality inspection approach enhanced by bayesian optimization through point cloud based sensors. Master's thesis, Politecnico di Milano (2020)
50. Chitta, S., Sucan, I., Cousins, S.: Moveit![ros topics]. IEEE Robotics & Automation Magazine **19**(1), 18–19 (2012)
51. Cully, A., Chatzilygeroudis, K., Allocati, F., Mouret, J.-B.: Limbo: A fast and flexible library for bayesian optimization. arXiv:1611.07343 (2016)
52. Singh, A., Sha, J., Narayan, K.S., Achim, T., Abbeel, P.: Bigbird: A large-scale 3d database of object instances. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 509–516. IEEE (2014)
53. Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S.: Objectnet3d: A large scale database for 3d object recognition. In: European Conference on Computer Vision, pp. 160–176. Springer (2016)
54. Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., Katz, B.: Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In: Advances in Neural Information Processing Systems, pp. 9453–9463 (2019)
55. Su, H., Qi, C.R.., Li, Y., Guibas, L.J.: Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2686–2694 (2015)
56. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3d models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1278–1286 (2015)
57. Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. How useful is photo-realistic rendering for visual learning? In: European Conference on Computer Vision, pp. 202–217. Springer (2016)
58. Mitash, C., Bekris, K.E., Boularias, A.: A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 545–551. IEEE (2017)
59. Tushar, J., Sardana, H.K., et al.: Mechanical cad parts recognition for industrial automation. In: Smart Computing and Informatics, pp. 341–349. Springer (2018)
60. Ben Abdallah, H., Jovančević, I., Orteu, J.-J., Brèthes, L.: Automatic inspection of aeronautical mechanical assemblies by matching the 3d cad model and real 2d images. Journal of Imaging **5**(10), 81 (2019)

61. Song, K.-T., Wu, C.-H., Jiang, S.-Y.: Cad-based pose estimation design for random bin picking using a rgb-d camera. Journal of Intelligent & Robotic Systems **87**(3–4), 455–470 (2017)
62. Murphy, K., Torralba, A., Eaton, D., Freeman, W.: Object detection and localization using local and global features. In: Toward Category-Level Object Recognition, pp. 382–400. Springer (2006)
63. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
64. Czajewski, W., Kołomyjec, K.: 3d object detection and recognition for robotic grasping based on rgb-d images and global features. Foundations of Computing and Decision Sciences **42**(3), 219–237 (2017)
65. Sukanya, C.M., Gokul, R., Paul, V.: A survey on object recognition methods. International Journal of Science, Engineering and Computer Technology **6**(1), 48 (2016)
66. Castellani, U., Cristani, M., Fantoni, S., Murino, V.: Sparse points matching by combining 3d mesh saliency with statistical descriptors. In: Computer Graphics Forum, vol. 27, pp. 643–652. Wiley Online Library (2008)
67. Digne, J., Cohen-Steiner, D., Alliez, P., De Goes, F., Desbrun, M.: Feature-preserving surface reconstruction and simplification from defect-laden point sets. Journal of Mathematical Imaging and Vision **48**(2), 369–382 (2014)
68. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision **73**(2), 213–238 (2007)
69. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J.: 3d object recognition in cluttered scenes with local surface features: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(11), 2270–2287 (2014)
70. do Monte Lima, J.P.S., Teichrieb, V.: An efficient global point cloud descriptor for object recognition and pose estimation. In: 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 56–63. IEEE (2016)
71. Alhamzi, K., Elmogy, K., Barakat, S.: 3d object recognition based on local and global features using point cloud library. International Journal of Advancements in Computing Technology **7**(3), 43 (2015)
72. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4. IEEE (2011)
73. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: 2011 international conference on Computer Vision, pp. 858–865. IEEE (2011)
74. Hodaň, T., Zabulis, X., Lourakis, M., Obdržálek, Š., Matas, J.: Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4421–4428. IEEE (2015)
75. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering **160**(1), 3–24 (2007)
76. Wang, C., Martín-Martín, R., Xu, D., Lv, J., Lu, C., Fei-Fei, L., Savarese, S., Zhu, Y.: 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 10059–10066. IEEE (2020)
77. Tjaden, H., Schwanecke, U., Schömer, E., Cremers, D.: A region-based gauss-newton approach to real-time monocular multiple object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(8), 1797–1812 (2018)
78. Song, C., Song, J., Huang, Q.: Hybridpose: 6d object pose estimation under hybrid representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 431–440 (2020)
79. Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N.: Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016, p. 205–220. Springer International Publishing, Cham. ISBN 978-3-319-46487-9 (2016)

Ph.D. Loris Roveda received the M.Sc.(2011) and the Ph.D. (2015) in Mechanical Engineering at Politecnicodi Milano. Currently, he is a senior researcher at SUPSI-IDSIA,working on AI and ML techniques applied to industrial robotics (suchas robot control, human-robot collaboration, dynamics identification). He has been involved in many national and Europeanprojects, and he is now coordinating the EUROBENCH STEPbySTEP, XSPINE, and REMOTe_XSPINE projects, and he is the P.I. of the H2020CS2 ASSASSINN project.

Eng. Marco Maroni Marco Maronireceived the M.Sc. (2020) in Mechanical Engineering at Politecnico diMilano. During his master thesis he worked on robotics and machinevision algorithms for industrial applications.

Eng. Lorenzo Mazzuchelli LorenzoMazzuchelli received the M.Sc. (2020) in Mechanical Engineering atPolitecnico di Milano. During his master thesis he worked on roboticsand machine vision algorithms for industrial applications.

Eng. Loris Praolini Loris Praolinireceived the M.Sc. (2020) in Mechanical Engineering at Politecnico diMilano. During his master thesis he worked on robotics and machinevision algorithms for industrial applications.

Asad Ali Shahid Asad Ali Shahid received the M.Sc. degree in mechanical engineering from the Politecnico di Milano, Milan, Italy, in 2020. During his master's degree, he also studied at ETH Zürich, Zürich, Switzerland, for one semester. He is now a researcher at IDSIA USI/SUPSI. His research interests lie around robotics and machine learning. In particular, he is interested in endowing industrial robots with capabilities to learn skills from people (imitation learning) and on their own (reinforcement learning).

Prof. GiuseppeBucca Giuseppe Bucca is Associate Professor of Applied Mechanics atPolitecnico di Milano. He graduated inMechanical Engineering in 2001. In 2005 he received the Ph.D. inMechanical Systems Engineering at Politecnico di Milano and in 2007he joined the academic permanent staff in the same university. Hisresearch activity and teaching activity are focused on the dynamicsand control of mechanical systems. He participated and participatesto several research project funded by EU and he was and isresponsible of research and technical projects funded by privatecompanies.

Ph.D. Dario Piga Dario Piga received his Ph.D. in Systems Engineering from thePolitecnico di Torino (Italy) in 2012. He was a Postdoctoral-Researcher at the Delft University of Technology (The Netherlands) in2012 and at the Eindhoven University of Technology (The Netherlands)in 2013. From 2014 to early 2017 he was Assistant Professorat the IMTSchool for Advanced Studies Lucca (Italy) and since March 2017 he hasbeen Senior Researcher at the IDSIA Dalle Molle Institute forArtificial Intelligence in Lugano (Switzerland) and Lecturer at theSUPSI University of Applied Sciences and Arts of SouthernSwitzerland. His main research interests include systemidentification, robust control, Bayesian filtering and non-convexoptimization, with applications to process control and smartmanufacturing.