2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Visual Quality Inspection Planning: A Model-Based Framework for Generating Optimal and Feasible Inspection Poses

Vanessa Staderini[1], Tobias Glück[1], Philipp Schneider[1] and Andreas Kugi [1,2]

***Abstract*— Automatic visual quality inspection is pivotal in both computer vision and robotics. It plays a crucial role in manufacturing, where robotic systems are increasingly employed to enhance the speed and efficiency of visual quality assessments. Several inspection planning methodologies have been developed; however, they often address the inspection challenge from a singular perspective of robotics or computer vision. This work introduces a comprehensive approach that synergistically integrates principles from both domains. We present an innovative algorithm designed to generate optimal inspection poses by considering the interplay between the inspected object's geometry and the kinematics of the robotic setup used for inspection. This is accomplished by taking advantage of the concept of visibility. The effectiveness of our algorithm is demonstrated through simulations and experiments, revealing complete coverage for diverse geometries and materials with a small number of inspection poses. Moreover, we benchmark our framework against box constraints and workspace sampling techniques to generate feasible inspection poses. The results indicate superior performance in achieving extensive coverage and reducing the number of required optimal inspection poses, enhancing the overall inspection process.**

I. Introduction

As the manufacturing industry progresses, the demand for high-quality products grows, requiring quality assurance in line with product quality standards. To this end, quality inspection at different stages of the production line is essential, where manual inspection remains the preferred option. However, manual inspection is time-consuming and prone to human errors caused mainly by fatigue and inattention due to the repetitiveness of the manual inspection procedures. Although the rapid development of 3D sensor technology and robotics enables automatic inline inspection, automating visual inspection poses challenges due to complex 3D geometries and the variety of possible defects for the parts to be assessed.

It is necessary to develop an inspection plan by addressing the coverage path planning problem, see [2], which consists of solving (i) the viewpoint planning problem (VPP) to find a set of optimal camera poses (optimal viewpoints) from which to inspect the target object, and (ii) the path planning problem (PPP) to define a time-optimal and collision-free trajectory for a given robotic setup that connects the optimal viewpoints. This is true in the case of a moving camera. In the case of a fixed camera and a moving object, the concept of viewpoints is replaced by that of objectpoints,

[1]The authors are with the Center for Vision, Automation and Control at AIT Austrian Institute of Technology GmbH, Vienna, Austria vanessa.staderini@ait.ac.at

[2]Andreas Kugi is also with the Automation and Control Institute, Technische Universität Wien, Vienna, Austria

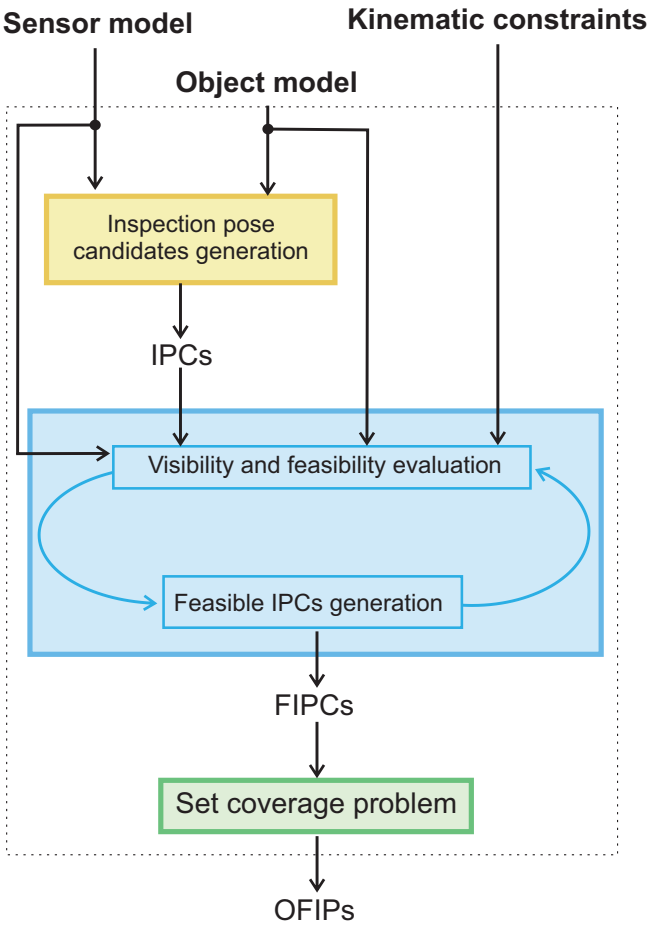


Fig. 1: Pipeline for the generation of optimal and feasible inspection poses (OFIPs). Given the model of the sensor and the 3D mesh of the target object, inspection pose candidates (IPCs) are generated [1]. Based on these IPCs, a two-stage assessment process is performed. The first stage involves conducting visibility and feasibility evaluations on the IPCs. This assessment accounts explicitly for the object, the sensor model, and the kinematic constraints of the robotic system. The second stage employs an optimization to refine the unfeasible inspection pose candidates and generate feasible inspection pose candidates (FIPCs). The procedure terminates solving the set coverage problem (SCP), which determines the final set of OFIPs.

represented by the poses that the object must assume in order to be covered by the camera. This is a common scenario in the manufacturing sector, as the object to be inspected often moves on a conveyor belt or is held by a robot that manipulates it during different production steps. In this work, we use the term *inspection poses* to refer to viewpoints or objectpoints.

To address the VPP, optimal inspection poses are often determined without accounting for kinematic constraints. Typically, feasibility checks for robotic configurations are conducted only after deriving these poses. Considering kinematic constraints is crucial, especially for robots with limited degrees of freedom. Such robots are cost-effective for tasks requiring a restricted range of motion. Additionally, if precision and speed are priorities, robots with fewer degrees of freedom and simpler designs may surpass more complex counterparts. Ignoring kinematic constraints in VPP, treated solely as a computer vision issue, diminishes inspection effectiveness and achievable coverage. To overcome this, we introduce an algorithm that simultaneously integrates the requirements from a computer vision and a robotics view. This approach systematically incorporates the geometry of the object being inspected, the sensor model, and the robot's kinematic constraints, as illustrated in Fig. 1.

The paper is organized as follows: Section II presents the state of the art of inspection pose generation with a focus on viewpoint generation methods. The formulation we adopt in this manuscript and our new method are introduced in Section III. The experiments are presented in Section IV, and the conclusions are drawn in Section V.

II. Related works

Research on solving the VPP has been ongoing for over four decades, but many challenges remain open. Methods for viewpoint generation can be divided into model-based or model-free methods.

Model-based methods assume that the object to be inspected is known, and candidate viewpoints are usually generated by sampling the object's surface or a volume containing it, as outlined in [3]. Then, the optimal viewpoints are retrieved by solving the set coverage problem by linear-integer programming or by applying heuristic methods, see [4]. The main disadvantage of model-based methodologies is that they generally do not consider kinematic constraints during viewpoint generation. As a result, some optimal inspection poses will be unfeasible for a given robotic system. The common approach here is to apply small, transformations to the viewpoints, such as rotations around the viewing direction or translations along it. If a feasible inspection pose cannot be found by applying these transformations, the viewpoint is discarded [5], [6]. In [7], the authors list the constraints that must be met during viewpoint planning: (i) geometric, (ii) optical, (iii) reconstructive, and (iv) environmental, where the latter includes the constraints of the kinematic setup, limiting the poses of the sensor and the inspected part that can be realized. Unfortunately, nothing is said about how to satisfy the environmental and, more specifically, the kinematic constraints and how to handle situations where a sampled viewpoint is not feasible. In [8], Mavrinac et al. propose a semiautomatic planning framework with particle swarm optimization to define some constrained camera parameters (e.g., position) after manually initializing the number of cameras and their orientation. An objective function is optimized subject to the inspection system's design constraints, using a coverage metric for the entire system at a specific task point on the object's surface. Although the object's geometry is known, it was not directly used to establish camera poses, but rather to compute the performance metric across all task points. In [9] and [10], viewpoints are calculated based on the geometry of the inspected object, and viewpoints outside the camera workspace are discarded. Subsequently, the viewpoints are connected by generating a path for the robot, and a feasibility analysis is performed so that the viewpoints that the robot cannot reach are modified through an optimization routine minimizing positioning and orientation errors. If a new viewpoint is not found for which the pose error is less than a certain threshold and the candidate child's pose falls within the lower and upper limits of the robot joint, the viewpoint is considered unfeasible and therefore excluded from the camera path planning. This approach does not guarantee that the modified viewpoint will provide the same visibility as the original one. A different possible approach to address the VPP is to sample the robot workspace according to user-defined importance distributions, as done in [11], for optimal placement of multiple visual sensors to maximize coverage. This has the advantage of generating feasible viewpoints, but these are not generated based on the geometry of the inspected object making challenging the inspection of complex geometries.

Model-free methods, such as next-best-view (NBV) planning [12], do not assume any knowledge of the object to be inspected. Given an initial viewpoint, the method selects the NBV as the one that maximizes a reward function and complies with the robot's workspace constraints, see [13]–[15]. In recent years, reinforcement learning has been employed to find the NBV, as in [16], in which an agent is trained to choose a camera pose and plan the robot's movement toward this pose. During sampling, any pose located outside the robot's workspace is discarded. Each feasible view pose is evaluated in terms of a reward that considers both the previously scanned surface and the unseen areas. The objective is to maximize coverage while minimizing the number of views required. Model-free methods are the right choice when the object is unknown or when there are no special requirements on the inspection time. However, they should be avoided in industrial applications for which the CAD model of the target object is known and the time constraints are strict.

A different approach for generating feasible viewpoints is proposed in [17]. In this case, the authors define topological spaces in which viewpoints can lie to satisfy all observation constraints of a specific object feature. The authors assume that the direction of the viewpoints is fixed and that the robot's workspace is known. This approach is based on

synthesis modeling and works well when there are only a few features to observe and when there is no particular need to reduce the number of optimal viewpoints required to perform the inspection since all viewpoints within each topological space are equally weighted. Furthermore, there is no guarantee that the viewpoint can be reached in the required orientation, and nothing is said about how such a situation will be handled. In [18], Do and Pham propose a model-based method to generate feasible robot configurations for cleaning tasks. Each ideal configuration is computed based on a Voronoi tessellation of the object and inverse kinematics (IK) solutions are determined to find feasible configurations for which the robot's end-effector is closely aligned with the desired one. Unfortunately, the method is not directly applicable to visual inspection applications, as visibility is not considered.

Addressing the unresolved challenges in the VPP for inspection tasks, we introduce a new algorithm to create feasible inspection pose candidates. This algorithm leverages knowledge of the target object and accounts for the kinematic constraints of the inspection setup. Optimal configurations are identified by resolving the set coverage problem (SCP). Subsequently, the most efficient sequence for visiting these configurations is determined by solving the traveling salesman problem, as detailed in [19].

III. Methodology

Fig. 2 depicts the inspection setup involving a fixed camera and a robot manipulating an object. These are defined by the camera and object frames $\mathcal{C}$ and $\mathcal{O}$, respectively. The inspection requires the robot to align $\mathcal{O}$ with the j-th objectpoint frame $\mathcal{O}_j$. The base and end-effector frames are denoted as $\mathcal{B}$ and $\mathcal{E}$. The world frame is $\mathcal{W}$.

A. Kinematic planning problem

The forward kinematics of the end-effector frame $\mathcal{E}$ w.r.t. the robot base frame $\mathcal{B}$ of a robot with n degrees of freedom is given by

$$\begin{bmatrix} \mathbf{p}_{\mathcal{B}}^{\mathcal{E}} \\ \mathbf{o}_{\mathcal{B}}^{\mathcal{E}} \end{bmatrix} = \mathbf{h}(\mathbf{q}), \tag{1}$$

where the joint configuration $\mathbf{q}^\top = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$ contains the generalized coordinates of the robot. The pose (1) consists of the Cartesian position vector $\mathbf{p}^\top = \begin{bmatrix} x & y & z \end{bmatrix}$ and the orientation is expressed as unit quaternion $\mathbf{o}^\top = \begin{bmatrix} \eta & \boldsymbol{\epsilon}^\top \end{bmatrix}$, with scalar part η and vector part $\boldsymbol{\epsilon}$. Similarly, the pose (1) can also be expressed as homogeneous transformation

$$\mathbf{H}_{\mathcal{B}}^{\mathcal{E}} = \begin{bmatrix} \mathbf{R}_{\mathcal{B}}^{\mathcal{E}} & \mathbf{p}_{\mathcal{B}}^{\mathcal{E}} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2}$$

with rotation matrix $\mathbf{R}_{\mathcal{B}}^{\mathcal{E}} \in SO(3)$ associated with the orientation $\mathbf{o}_{\mathcal{B}}^{\mathcal{E}}$. The notation $(\cdot)_{\mathcal{A}}^{\mathcal{B}}$ denotes the geometric relation of frame $\mathcal{B}$ with respect to frame $\mathcal{A}$, expressed in $\mathcal{A}$. Consequently, we can compose homogeneous transformations to express the object frame $\mathcal{O}$ w.r.t. the world frame $\mathcal{W}$ as

$$\mathbf{H}_{\mathcal{W}}^{\mathcal{O}} = \mathbf{H}_{\mathcal{E}}^{\mathcal{O}} \mathbf{H}_{\mathcal{B}}^{\mathcal{E}} \mathbf{H}_{\mathcal{W}}^{\mathcal{B}}. \tag{3}$$

B. Candidate inspection poses planning problem

In accordance to what is commonly done for the inspection planning, N viewpoints $\mathbf{v}_j$ with $j = \{1, \ldots, N\} \in \mathbb{N}$ are computed as candidates assuming a fixed object and a moving camera. Each viewpoint is defined by a frame $\mathcal{C}_j$, see Fig. 3. The viewpoints are often expressed with respect to the object frame, and the corresponding homogeneous transformation is $\mathbf{H}_{\mathcal{O}}^{\mathcal{C}_j}$.

In our work, we generate the viewpoint candidates adopting the model-based method presented in [1], [20]. Given a 3D model of the object to be inspected as mesh, Poisson disk sampling is applied to the object's surface. Starting from each sample, a standoff correction is executed along the normal of the object surface corresponding to that sample, as in Fig. 3. This way, we guarantee evenly spaced viewpoints and the object to be in the field of view (FOV) and depth of field (DOF) of the sensor. The camera center is positioned at the end point of the translation and is oriented with its heading direction (z-axis) as the normal vector along which the translation was performed. The y-axis is the up vector, and the x-axis is naturally determined. Given the viewpoints, it is possible to determine the corresponding objectpoints for a setup with a fixed camera and a moving object as in our configuration. This is done by forcing the equality

$$\mathbf{H}_{\mathcal{C}}^{\mathcal{O}_j} = \mathbf{H}_{\mathcal{C}_j}^{\mathcal{O}}. \tag{4}$$

Therefore, we can write the homogeneous transformation that describes the j-th objectpoint in the world frame when the camera is fixed and the object moves as

$$\mathbf{H}_{\mathcal{W}}^{\mathcal{O}_j} = \mathbf{H}_{\mathcal{C}}^{\mathcal{O}_j} \mathbf{H}_{\mathcal{E}}^{\mathcal{C}} \mathbf{H}_{\mathcal{B}}^{\mathcal{E}} \mathbf{H}_{\mathcal{W}}^{\mathcal{B}}. \tag{5}$$

In the following, we refer to inspection pose (IP) $\boldsymbol{\iota}_j = \{\mathbf{v}_j, \mathbf{o}_j\}$ as our framework can be applied to both viewpoints and objectpoints generation methods since they are related as shown in (4) and (5). Each inspection pose has associated a frame $\mathcal{I}_j = \{\mathcal{C}_j, \mathcal{O}_j\}$. The method is intuitive to implement, not time-intensive and produces a model-based set of inspection poses (IPs) capable of covering the entire object surface.

C. Feasible inspection poses planning problem

Given the forward kinematics of a manipulator (1), an inspection pose $\boldsymbol{\iota}_j = \mathbf{h}(\mathbf{q}_j)$ is called (kinematically) feasible [21] if the joint coordinates $\mathbf{q}_j[l]$, with $l = 1, ..., n$, are within the joint limits, i.e., $\mathbf{q}_j[l] \in [\underline{q}_l, \overline{q}_l]$, where $\underline{q}_l$ and $\overline{q}_l$ are the lower and upper limits of the *l-th* joint, respectively and if the robot is not in self-collision or colliding with the environment. For a given set of inspection pose candidates (IPCs) $\boldsymbol{\iota}_j,\ j = 1, \ldots, N$, we assess their feasibility under the given kinematic constraints and compile a set Υ consisting of indices corresponding to those configurations deemed unfeasible, such that $|\Upsilon| \leq N$. The indices corresponding to feasible configurations are stored in Φ, ensuring that

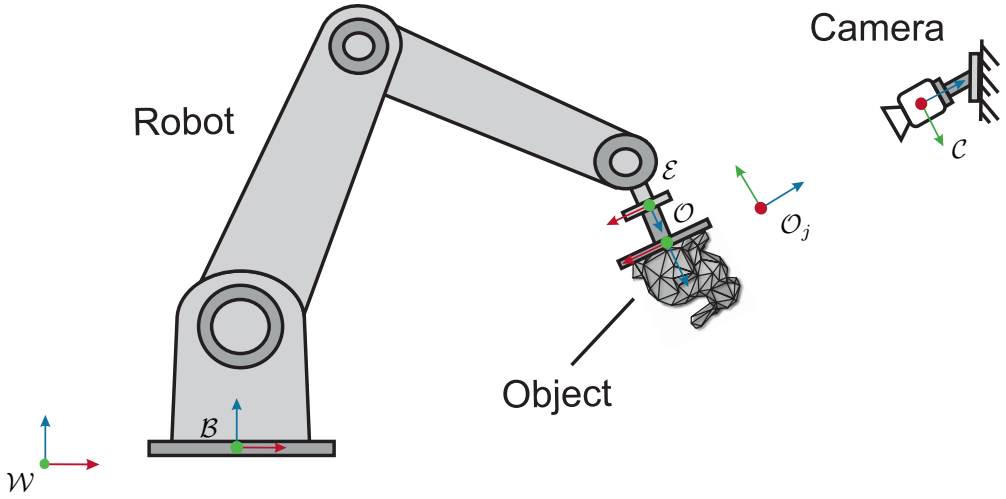


Fig. 2: Illustration of a robot inspection setup. The robot moves the object defined by frame $\mathcal{O}$ in front of a fixed camera with frame $\mathcal{C}$. The base and end-effector frames of the robot are $\mathcal{B}$ and $\mathcal{E}$, respectively. The global reference is the world frame $\mathcal{W}$. The inspection task involves aligning the object frame $\mathcal{O}$ with the j-th objectpoint frame $\mathcal{O}_j$.

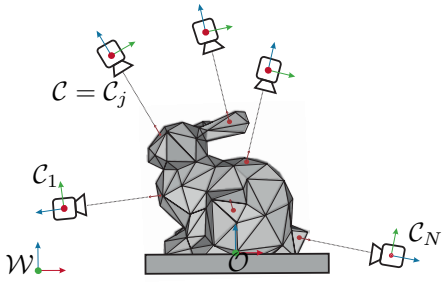


Fig. 3: Viewpoint candidates with frame $\mathcal{C}_j$ with $j = 1, \ldots, N$ computed w.r.t. the frame $\mathcal{O}$ of a fixed object. The inspection task consists in aligning the camera frame $\mathcal{C}$ with the j-th viewpoint frame $\mathcal{C}_j$. The viewpoints were computed based on Poisson disk sampling and standoff correction. The heading direction is indicated in blue, and the up vector is indicated in green. The world frame is reported as $\mathcal{W}$.

$|\Upsilon| + |\Phi| = N$. Subsequently, we formulate an optimization problem and request the following properties:

1) We want to determine a feasible configuration that covers at least the same faces that were covered from the unfeasible configurations $\mathbf{q}_j$ with $j \in \Upsilon$.
2) We want to maximize the coverage achieved with each inspection pose.
3) We want to cover those faces that are not covered by the already determined feasible configurations $\mathbf{q}_j$ with $j \in \Phi$.

The unfeasible configurations are corrected by solving an optimization problem in which solutions are sampled inside the workspace of the setup $\mathbf{q}_j \in [\underline{\mathbf{q}}, \overline{\mathbf{q}}]$. An initial solution can be determined by orthogonal projection of the unfeasible configurations onto the feasible joint limit sets $[\underline{\mathbf{q}}, \overline{\mathbf{q}}]$. The optimization comprises three main parts, explained below:

1) We introduce the visibility matrix $\mathbf{V} \in \mathbb{R}^{M \times N}$, with M the number of surface elements of the mesh and, N the number of inspection poses. For the j-th inspection pose $\boldsymbol{\iota}_j$ and i-th surface element s_i of the inspected object provided as mesh, the matrix is element-wise defined as

$$\mathbf{V}(s_i, \boldsymbol{\iota}_j) = V_{i,j} = V[i,j] = \\ = \begin{cases} 0 & \text{if } s_i \text{ is not visible for } \boldsymbol{\iota}_j, \\ 1 & \text{if } s_i \text{ is visible for } \boldsymbol{\iota}_j. \end{cases} \tag{6}$$

In addition, we can introduce the visibility column vector

$$\mathbf{c}(\mathbf{q}_j) = \mathbf{V}[:, j], \; j \in \Upsilon \tag{7}$$

associated with the j-th unfeasible configuration $\mathbf{q}_j$. In order to determine a feasible configuration for covering the faces that were covered in the unfeasible inspection pose, a naive approach could be to solve the optimization problem

$$\min_{\mathbf{w}_j \in [\underline{\mathbf{q}}, \overline{\mathbf{q}}]} \left\| \mathbf{c}(\mathbf{w}_j) - \mathbf{c}(\mathbf{q}_j) \right\|_2^2 \tag{8}$$

to infer feasible configurations $\mathbf{w}_j$ from the initially unfeasible configuration $\mathbf{q}_j$, $j \in \Upsilon$. However, this approach is ineffective because the visibility column vector contains zeros and ones. The optimization would attempt to determine a feasible configuration for which the faces that were previously covered are covered and those unseen remain unseen. This is not what we want. Therefore, we define an index set

$\Gamma_j = \{i|\mathbf{V}(s_i, \boldsymbol{\iota}_j) = 0\} \subseteq \{1, 2, ..., M\}$ containing all the indices relative to the surface elements s_i that cannot be viewed in the unfeasible configuration $\mathbf{q}_j$, $j \in \Upsilon$. Additionally, we define an operator $\pi(\cdot)$ that is applied to vectors or matrices such that when $\pi_{\Gamma_j}(\cdot)$ is applied to $\mathbf{c}(\mathbf{q}_j)$, all the rows with the number corresponding to an entry in Γ_j are removed from $\mathbf{c}(\mathbf{q}_j)$, resulting in a reduced all-one vector

$$\pi_{\Gamma_j}[\mathbf{c}(\mathbf{q}_j)] = \mathbb{1}. \tag{9}$$

Accordingly, using the objective term

$$\mathcal{F}_1 = \left\|\pi_{\Gamma_j}[\mathbf{c}(\mathbf{w}_j)] - \mathbb{1}\right\|_2^2, \tag{10}$$

only initially visible surface elements are considered within the optimization process.

2) We want to maximize the coverage offered by the new feasible configuration $\mathbf{w}_j$. The coverage due to a single inspection pose $\boldsymbol{\iota}_j$ is given by

$$\mathrm{COV}(\boldsymbol{\iota}_j) = \frac{1}{M}\sum_{i=1}^{M}\mathbf{V}_{i,j}, \tag{11}$$

with M the number of surface elements of the mesh, and $\mathbf{V}_{i,j}$ element associated to the visibility matrix as in (6). We recall that $\boldsymbol{\iota}_j = \mathbf{h}(\mathbf{w}_j)$ holds and we introduce the objective term as $\mathcal{F}_2 = \mathrm{COV}(\mathbf{w}_j)$ with $\mathrm{COV}(\boldsymbol{\iota}_j) = \mathrm{COV}(\mathbf{w}_j)$.

3) The two objective terms introduced so far concern the optimization of the single configuration $\mathbf{w}_j$ and do not consider any information about the feasible configurations $\mathbf{q}_j$ with $j \in \Phi$. Moreover, the second objective term $\mathcal{F}_2$ may favor the emergence of identical feasible configurations, determining some faces to be covered in many inspection poses and other faces being uncovered. For this reason, we introduce a third term that attempts to increase the coverage of faces that are unseen with the already determined feasible configurations $\mathbf{q}_j$, $j \in \Phi$.
This is achieved by initializing a matrix $\bar{\mathbf{V}}$ defined column-wise as

$$\bar{\mathbf{V}}[:, j] = \bar{\mathbf{V}}_{:,j} = \begin{cases}\mathbf{c}(\mathbf{q}_j) & \text{if } j \in \Phi,\\ \mathbf{0} & \text{if } j \in \Upsilon.\end{cases} \tag{12}$$

Every time a new feasible configuration $\mathbf{q}_j^*$ is determined starting from $\mathbf{q}_j$ with $j \in \Upsilon$, the matrix $\bar{\mathbf{V}}$ is updated by replacing the j-th null column by $\mathbf{c}(\mathbf{q}_j^*)$ as in (7). Moreover, the sets Φ and Υ are updated so that $\Phi \to \Phi \cup \{j\}$ and $\Upsilon \to \Upsilon \setminus \{j\}$.
Based on this, every time we solve the optimization problem for an unfeasible configuration $\mathbf{q}_j$ with $j \in \Upsilon$, we define an index set $\Delta_j = \{i|\sum_{k=1}^{N}\bar{\mathbf{V}}_{i,k} > 0\} \subseteq \{1, 2, ..., M\}$ containing the indices of the surface elements covered by current feasible configurations, if any. Then, the operator $\pi_{\Delta_j}(\cdot)$ defined above is applied to filter out from $\mathbf{c}(\mathbf{w}_j)$ those elements that are already covered, i.e., element corresponding to the rows with index in Δ_j. This matrix is then used to compute the *reduced coverage* as

$$\widetilde{\mathrm{COV}}(\mathbf{w}_j) = \frac{\left\|\pi_{\Delta_j}[\mathbf{c}(\mathbf{w}_j)]\right\|_1}{\dim(\pi_{\Delta_j}[\mathbf{c}(\mathbf{w}_j)])} \tag{13}$$

and the objective term is

$$\mathcal{F}_3 = \widetilde{\mathrm{COV}}(\mathbf{w}_j). \tag{14}$$

Finally, each unfeasible configuration is corrected solving the optimization problem

$$\mathbf{q}_j^* = \arg\max_{\mathbf{w}_j \in [\underline{\mathbf{q}}, \overline{\mathbf{q}}]} \mathcal{F}(\mathbf{w}_j) \text{ for } j \in \Upsilon, \tag{15a}$$

with the objective function

$$\mathcal{F}(\mathbf{w}_j) = a\mathcal{F}_3 + b\mathcal{F}_2 + \frac{1}{\mathcal{F}_1}. \tag{15b}$$

Here, $a > 0$ and $b > 0$ are tuning parameters. To solve the optimization problem (15) numerically, Bayesian optimization [22]–[24] is used, see Fig. 4.

Once the optimization problem has been solved to refine each unfeasible inspection pose, the visibility matrix of the feasible configurations is used to solve the set coverage problem and determine the optimal inspection poses $\mathcal{I}^*$ with coverage computed as

$$\mathrm{COV}(\mathcal{I}^*) = \frac{1}{N}\sum_{j=1}^{N}\mathrm{COV}(\boldsymbol{\iota}_j). \tag{16}$$

The term $\mathrm{COV}(\boldsymbol{\iota}_j)$ is determined as in (11).

IV. Experiments

Our novel algorithm was evaluated in simulation and validated in lab experiments. The experimental setup featured a fixed camera observing an object in motion on a robotic platform with three degrees of freedom, as depicted in Fig. 5. Various objects, each with distinct geometries and material properties (as illustrated in Fig. 6), were considered for testing. These included a 3D-printed convex rabbit, an industrial component referred to as 'bracket' in both metal and 3D-printed plastic, a 3D plastic-printed parallelepiped, a gear, and a hirth[1]. This diverse selection aimed to assess the algorithm's performance on different geometries and materials as the rabbit is characterized by a smooth surface while the parallelepiped has sharp edges. Furthermore, the inclusion of the gear and hirth models, publicly available and previously utilized in other works for inspection planning [1], [3], [20], was deliberate. These represent objects common in research and industry and present several self occlusions. Finally, the bracket is provided both in plastic and aluminum alloy to better asses the capabilities of our framework on materials with different properties. Our aim was to explore the performance of the algorithm when applied in the lab to industrial and research components with different geometries and materials. The acquisition setup, as illustrated in Fig. 5,

[1] https://owncloud.fraunhofer.de/index.php/s/H8jV9rwGN84knzP (Accessed July 17, 2024)

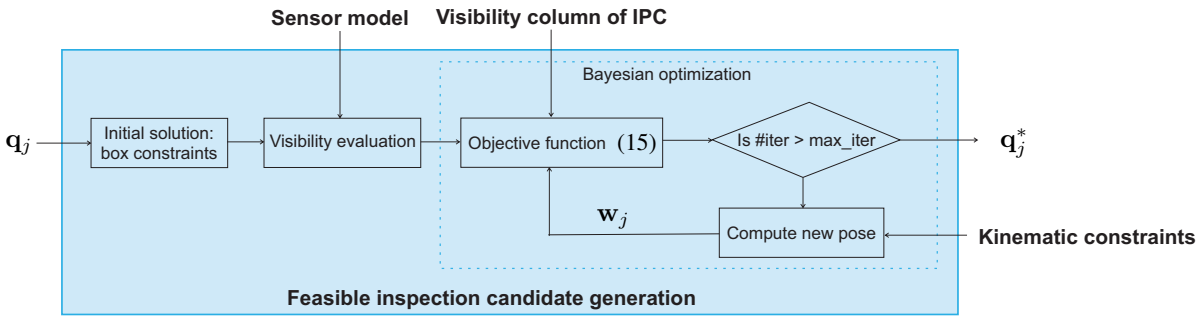


Fig. 4: Pipeline for the refinement of unfeasible configurations. The corresponding robotic configuration is computed for each inspection pose candidate (IPC). If no feasible configuration exists, the inspection pose is modified via a Bayesian optimization aiming at finding a new configuration that maximizes (15).

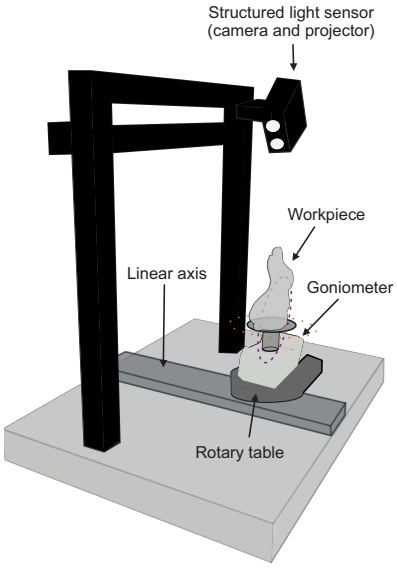


Fig. 5: Acquisition setup: Fixed structured light sensor and moving object with three degrees of freedom.

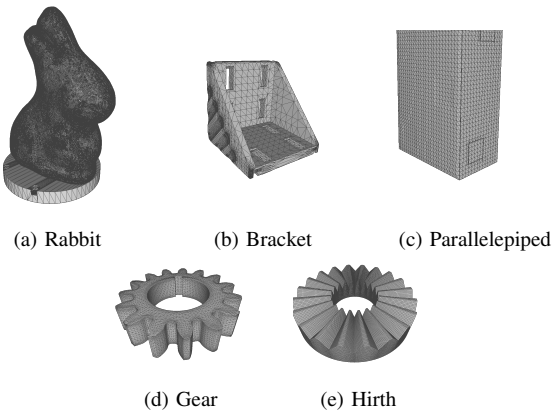


Fig. 6: 3D meshes of the objects used in the experiments.

has a fixed camera and allows movement of the object with three degrees of freedom.

Although our framework is versatile and can be adapted for any set of inspection poses, we specifically show the results obtained in simulation and experimentally when the method proposed in [1] is used to generate the inspection pose candidates.

A. Simulation

In our simulations, we employed a sensor with a depth of field (DOF) ranging from 350 mm to 700 mm. The camera sensor featured a dimension of 1920 × 1200 pixels with a field of view (FOV) of $38.70° \times 24.75°$. The projector sensor comprised an array of 1140 × 912 micro-mirrors, with a FOV of $45.23° \times 24.75°$. The projector was tilted by $15°$ around the up axis of the camera, and the baseline between the camera and the projector was set to 137 mm. These specifications were inspired by the data sheet of a structured light sensor (Zivid One+ S[2]), which was utilized in our lab experiments. The IBM ILOG CPLEX Optimizer [25] was employed to solve the SCP and the traveling salesman problem [26].

To assess our method, we compared our novel methodology based on Bayesian optimization (BO) with two common approaches: (i) box constraints (BC) and (ii) workspace sampling (WS). In the case of *box constraints*, unfeasible configurations were transformed into feasible ones through orthogonal projection, as detailed in Section III, without incorporating Bayesian optimization. For *workspace sampling*, inspection poses were randomly determined with a uniform distribution within the workspace, disregarding any information about the object. The 3D mesh of the object was then used to evaluate candidates and define optimal inspection poses.

Table I presents the results of generating optimal and feasible inspection poses (OFIPs) for BC, WS, and BO across five different objects. The number of inspection pose candidates

[2]https://www.zivid.com

TABLE I: Comparison of OFIPs generation methods.

Object	Method	#IPCs	#OFIPs	Coverage (%)
Parallelepiped	BC	50	10	98.70
	WS	50	8	99.77
	BO	50	**6**	**100**
Rabbit	BC	50	19	60.75
	WS	50	17	95.02
	BO	50	20	**99.52**
Bracket	BC	100	36	55.92
	WS	100	29	97.55
	BO	100	**28**	**99.60**
Gear	BC	50	18	68.11
	WS	50	20	97.11
	BO	50	21	**99.80**
Hirth	BC	50	9	72.82
	WS	50	8	98.55
	BO	50	**7**	**99.77**

(#IPCs) is chosen accordingly to the complexity and dimension of the objects. The reported performance represents average values over five executions. We utilized an open-source library [24] for Bayesian optimization with expected improvement as a utility function, domain reduction to accelerate the search, and 60 iterations per inspection pose. Our method consistently achieved higher coverage (%), aligning with the primary objective of our work.

Moreover, the number of optimal and feasible inspection poses (#OFIPs) for BO remains low, thanks to the goodness of the original candidate configurations and the optimization procedure. The optimization problem (15) is partially influenced by the choice of the inspection pose used to initiate the optimization process. Despite this dependency, conducting a time-intensive optimization cycle to decide from which configuration to begin the optimization was considered unnecessary as it increased coverage as low as 0.2%. Workspace sampling (WS) showed comparable performance to the box constraint method (BC) when dealing with objects with simple geometry, and the number of candidates is small; see the results presented in Table I for the parallelepiped. However, WS outperformed BC in the case of higher candidate numbers, as BC tended to concentrate new feasible inspection poses in the same locations, i.e., at the boundaries of the workspace. Although WS achieved over 95% coverage for all five objects, increasing the number of candidates was required to match the performance of the BO method. With 1000 initial candidates, WS achieved 98.6% and 98.4% coverage for the hirth and the gear, respectively, with similar results for the other objects. This makes finding the solution of the SCP, formulated as an integer linear programming problem, challenging as the input of the SCP is the visibility matrix (6), and its size is affected both by the number of surface elements of the mesh and the number of candidates. On the contrary, a key strength of our algorithm lies in its ability to achieve high coverage with a low number of candidates, enhancing efficiency in solving the SCP. While increasing the number of Bayesian iterations extends computation time, this is inconsequential for industrial applications with offline inspection planning.

B. Real-world applications

In the simulation phase, our method, based on Bayesian optimization (BO), demonstrated a remarkable capability to achieve higher coverage compared to alternative methods such as workspace sampling (WS) and box constraints (BC). Furthermore, coverage exceeded 99% for all five objects, showcasing the efficacy of our approach in virtual environments.

To assess the practical utility and validity of our framework in real-world applications, we conducted experiments using our method to autonomously generate optimal and feasible inspection poses. In these experiments, the Zivid One+ S structured light sensor was fixed in position, and diverse target objects were mounted on a robotic setup with three degrees of freedom. The setup included a kinematic chain comprising a Physik Instrumente (PI) LS-180 linear axis, and a PI PRS-200 rotary table stage incorporating a PI WT-120 goniometer, as depicted in Fig. 5. An initial calibration stage for the camera and setup facilitated reaching planned object poses, and rough alignment of acquired point clouds was achieved through forward kinematics.

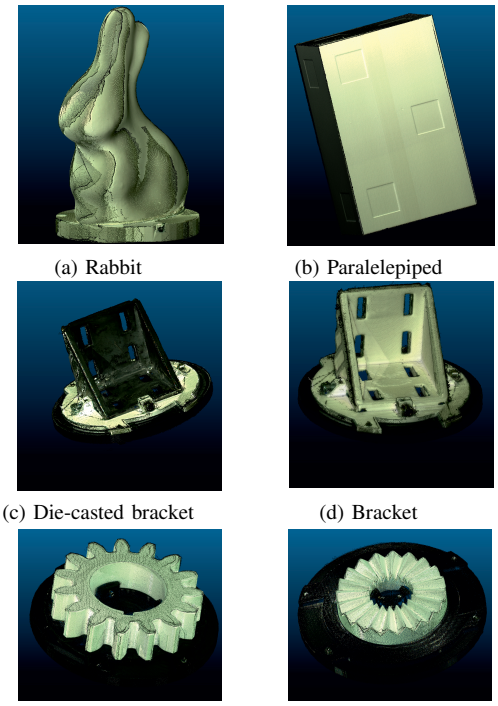
The alignment process involved saving initial joint positions, calculating travel between current and initial positions for each joint, adjusting the rotating stage axis, rotating the tilt stage (i.e., the goniometer), and applying translations along the linear axis to correct positional misalignment. Starting from this initial alignment, a refinement step using the iterative Closest Point algorithm (ICP) [27] is necessary to address errors between partial point cloud scans.

The acquired point clouds are shown in Fig. 7. The noise was removed by applying silhouette masking [28]. We suggest adopting Global ICP [29] to improve the quality of the point clouds and fully align them.

Despite the challenges in inspecting objects with different geometries and material properties, our pipeline was able to fully cover the objects in experimental settings as outlined in the simulations. The ability to adapt to diverse object geometries and materials underscores the versatility and practicality of our method for real-world inspection scenarios.

V. Conclusions

This paper presents a novel framework for generating optimal and feasible inspection poses for inspection planning. Unlike prior works, our approach leverages the object's geometry to generate seed inspection poses, refined through Bayesian optimization to ensure feasibility while evaluating coverage. The key innovation lies in computing feasible inspection poses that cover at least the same surface area as their unfeasible counterparts. This is attributed to the inherent capability of the initial seed inspection pose to provide complete coverage. Simulations demonstrated the superiority of our method, while our new framework was also experimentally validated. The results underscore the potential of our framework in optimizing objectpoint or



(a) Rabbit (b) Paralelepiped (c) Die-casted bracket (d) Bracket (e) Gear (f) Hirth

Fig. 7: Experimentally acquired point clouds. A denoising step was executed via silhouette masking. The surface of the objects was fully covered.

viewpoint generation for diverse applications, both for challenging object geometries and material properties. A possible future roadmap involves replacing Bayesian optimization with reinforcement learning to teach an agent to rectify unfeasible configurations and compare the performance of the two methods. Additionally, we will enhance the proposed framework to optimize globally all the inspection pose candidates. Furthermore, the presented methodology can be extended to a two-robot configuration with a moving camera and a moving object.

REFERENCES

[1] V. Staderini, T. Glück, P. Schneider, R. Mecca, and A. Kugi, "Surface sampling for optimal viewpoint generation," in *IEEE 13th International Conference on Pattern Recognition Systems (ICPRS)*, 2023.

[2] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai, "A survey on inspecting structures using robotic systems," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, pp. 1–18, 2016.

[3] P. Gospodnetić, D. Mosbach, M. Rauhut, and H. Hagen, "Viewpoint placement for inspection planning," *Machine Vision and Applications*, vol. 33, no. 1, pp. 1–21, 2022.

[4] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning with a registration constraint," in *IEEE Third International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 127–134.

[5] W. R. Scott, "Model-based view planning," *Machine Vision and Applications*, vol. 20, no. 1, pp. 47–69, 2009.

[6] W. Jing, C. F. Goh, M. Rajaraman, F. Gao, S. Park, Y. Liu, and K. Shimada, "A computational framework for automatic online path generation of robotic inspection tasks via coverage planning and reinforcement learning," *IEEE Access*, vol. 6, pp. 54 854–54 864, 2018.

[7] S. Chen and Y. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 393–408, 2004.

[8] A. Mavrinac, X. Chen, and J. L. Alarcon-Herrera, "Semiautomatic model-based view planning for active triangulation 3-d inspection systems," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 799–811, 2014.

[9] R. Malhan, R. Jomy Joseph, P. M. Bhatt, B. Shah, and S. K. Gupta, "Algorithms for improving speed and accuracy of automated three-dimensional reconstruction with a depth camera mounted on an industrial robot," *Journal of Computing and Information Science in Engineering*, vol. 22, no. 3, p. 031012, 2022.

[10] R. K. Malhan and S. K. Gupta, "Planning algorithms for acquiring high fidelity pointclouds using a robot for accurate and fast 3d reconstruction," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102372, 2022.

[11] E. Hörster and R. Lienhart, "On the optimal placement of multiple visual sensors," in *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, 2006, pp. 111–120.

[12] C. Connolly, "The determination of next best views," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 432–435.

[13] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next-best-view planning for 3d object reconstruction with positioning error," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, 2014.

[14] S. Wu, W. Sun, P. Long, H. Huang, D. Cohen-Or, M. Gong, O. Deussen, and B. Chen, "Quality-driven poisson-guided autoscanning," *ACM Transactions on Graphics*, vol. 33, no. 6, pp. 1–12, 2014.

[15] C. Mineo, D. Cerniglia, V. Ricotta, and B. Reitinger, "Autonomous 3d geometry reconstruction through robot-manipulated optical sensors," *The International Journal of Advanced Manufacturing Technology*, vol. 116, pp. 1895–1911, 2021.

[16] C. Landgraf, B. Meese, M. Pabst, G. Martius, and M. F. Huber, "A reinforcement learning approach to view planning for automated inspection tasks," *Sensors*, vol. 21, no. 6, p. 2030, 2021.

[17] A. Magaña, J. Dirr, P. Bauer, and G. Reinhart, "Viewpoint generation using feature-based constrained spaces for robot vision systems," *Robotics*, vol. 12, no. 4, 2023.

[18] V.-T. Do and Q.-C. Pham, "Geometry-aware coverage path planning for depowdering on complex 3d surfaces," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5552–5559, 2023.

[19] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119 310–119 342, 2021.

[20] V. Staderini, T. Glück, R. Mecca, P. Schneider, and A. Kugi, "Spatial resolution metric for optimal viewpoints generation in visual inspection planning," in *International Conference on Computer Vision Systems*. Springer, 2023, pp. 197–207.

[21] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[22] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[23] A. Agnihotri and N. Batra, "Exploring bayesian optimization," *Distill*, 2020, access July 17, 2024. [Online]. Available: https://distill.pub/2020/bayesian-optimization

[24] F. Nogueira *et al.*, "Bayesian optimization: Open source constrained global optimization tool for python," *URL https://github.com/fmfn/BayesianOptimization*, 2014, access July 17, 2024.

[25] I. I. Cplex, "V12. 1: User's manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[26] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.

[27] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[28] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[29] P. Glira, N. Pfeifer, C. Briese, and C. Ressl, "A correspondence framework for als strip adjustments based on variants of the icp algorithm," *Photogrammetrie, Fernerkundung, Geoinformation*, vol. 2015, no. 4, pp. 275–289, 2015.