Coverage Path Planning with Targeted Viewpoint Sampling for Robotic Free-From Surface Inspection

Emile Glorieux[a,*], Pasquale Franciosa [a], Dariusz Ceglarek [a]

[a] Digital Lifecycle Management (DLM), WMG, University of Warwick, CV4 7AL Coventry (UK)

Abstract

Surface metrology systems are increasingly used for inspecting dimensional quality in manufacturing. The gauge of these measurement systems is often mounted as an end-effector on robotic systems to exploit the robots' high degrees of freedom to reposition the gauge to different viewpoints. With this repositioning flexibility, a planning methodology becomes necessary in order to carefully plan the viewpoints, as well as the optimal sequence and quickest path to move the gauge to each viewpoint. This paper investigates coverage path planning for robotic single-sided dimensional inspection of free-form surfaces. Reviewing existing feasible state-of-the-art methodologies to solve this problem led to identifying an unexplored opportunity to improve the coverage path planning, specifically by replacing random viewpoint sampling strategy. This study reveals that a non-random targeted viewpoint sampling strategy significantly contributes to solution quality of the resulting planned coverage path. By deploying optimisation during the viewpoint sampling, an optimal set of admissible viewpoints can be obtained, which consequently significantly shortens the cycle-time for the inspection task. Results that evaluate the proposed viewpoint sampling strategy for two industrial sheet metal parts, as well as a comparison with the state-of-the-art are presented. The results show up to 23.8% reduction in cycle-time for the inspection task when using targeted viewpoints sampling.

Keywords: 3D surface inspection, robot motion planning, free-form surface, dimensional metrology

1. Introduction

Industrial surface metrology technologies, deployed in manufacturing systems for dimensional quality inspection of free-form surfaces, often use robotic solutions for positioning the measurement gauge to different viewpoints in order to cover the surfaces that are to be inspected. These then provide the ability of having an automated inspection station that can be placed either near-line in proximity to the production system or in-line within the production line [1, 2]. Automated inspection of free-form surfaces near- or in-line helps to significantly reduce the mean-time-to-detection of defects.

The *coverage path planning problem* is the associated robot path planning problem to determine the viewpoints from where to measure the part's surfaces, the sequence to visit the viewpoints, as well as the collision-free paths to travel to each viewpoint. Several criteria need to be considered when planning the coverage path, including full coverage of the targeted surfaces but also the resulting cycle-time for the inspection task. It should be noted that similar path planning problems need to be considered for other applications in manufacturing such as non-destructive testing [3], surface quality inspection [4], and on-machine inspection [5].

The cycle-time for near- or in-line dimensional quality inspection is typically of high importance. A short cycle-time enables to more frequently inspect more dimensional quality features on the parts with a single robotic metrology gauge, thus improving the resource utilisation. The cycle-time is the combination of the travel time for the robot to move the gauge to each viewpoint and the inspection time for the measurement by the gauge at each viewpoint. It is important to minimise both in order to effectively reduce the inspection task cycle-time. The relative importance of minimising the number of viewpoints and minimising the robot travel time depends on the characteristics of the metrology technology and robot system.

The objectives of the work presented in this paper is to understand the capabilities and limitations of existing coverage path planning techniques for dimensional quality inspection, as well as exploring opportunities for improvements. The specific focus is on single-sided dimensional quality inspection for free-form surfaces such as sheet metal parts. It can typically be assumed that the thickness of sheet metal parts is constant across the geometry. Therefore, when covering the single-side surface of the sheet metal part, all dimensions are known. Sheet metal parts are often produced in high-volumes at fast production rate, which makes having a short cycle-time for the dimensional quality inspection important to support effective quality control. Near- or in-line metrology systems typically have a lower accuracy compared to contact point-based coordinate measurement machines, however dimensional deviation for sheet metal parts resulting from shrinkage, wrinkling and spring-back are usually of a higher degree than the accuracy of these systems and are thereby suitable for dimensional inspection [6].

The main contribution of this paper is the identification and

*Corresponding author
Email address: e.glorieux@warwick.ac.uk (Emile Glorieux)

evaluation of the proposed targeted viewpoint sampling strategy within the investigation of coverage path planning for robotic dimensional quality inspection of sheet metal parts. It is proposed to deploy an optimisation technique to iteratively find viewpoints with maximum coverage of the remaining uncovered area of the to-inspect surfaces and minimum travel time, and repeat this in order to obtain a redundant set of admissible viewpoints. A subset of these are then used for the resulting coverage path. This is evaluated for the two real-world industrial sheet metal parts. This paper demonstrates that using a targeted strategy for viewpoint sampling instead of random sampling gives significant cycle-time reductions.

The remainder of this paper is organised as follows; the coverage path planning problem for dimensional quality inspection of sheet metal parts is discussed in Section 2. Section 3 presents the identified state-of-the-art coverage path planning methodologies during the performed literature study. The motivation and presentation of the proposed targeted viewpoint sampling strategy can be found in Section 4. The two considered case studies are presented in Section 5, followed by the feasibility analysis of the state-of-the-art coverage path planning methodologies in Section 6. Finally, an in-depth evaluation of the proposed targeted viewpoint sampling strategy is presented together with the comparison of its performance against the feasible state-of-the-art methodologies in Section 7.

2. Coverage path planning

Coverage path planning problem, within the scope of this work, is concerned with planning the viewpoints for surface metrology gauge to measure the to-inspect surfaces and the collision-free path connecting those viewpoints in the optimal sequence. This work focusses on discrete coverage path planning problems and it is assumed that reference geometries of the to-inspect surfaces, workspace and obstacles are available as well as that the workpiece and robot placement are predefined. According to Almadhoun et al. [7], discrete coverage path planning problems are typically decomposed into two subproblems:

1. Coverage sampling: to find the smallest set of viewpoints that provides full coverage,
2. Multi-goal path planning: to find the optimal sequence and collision-free robot paths that connect all viewpoints.

The solution to the first sub-problem determines the number of measurements by the gauge in order to fully cover the to-inspect surfaces, and thereby the required measurement time. The solution to the second sub-problem determines the length of the robot path to reposition the gauge to each viewpoint, and thereby the required robot motion time. The time for completing the inspection task, i.e. the cycle-time, is then the combination of the measurement time and the robot motion time. In this work, the objective for coverage path planning problem is to minimise the total time to complete the inspection task.

2.1. Coverage sampling

The coverage sampling problem is concerned with generating a set of viewpoints that provide full coverage of the to-inspect surfaces [8]. The coverage sampling problem can be formulated using a set system (S, Q) where S is the finite set of geometric primitives s_i, which are relatively small-sized geometric elements (e.g. nodes, triangles, rectangles, etc.), that comprise the surfaces that need to be inspected, and Q is the robot configuration space. A viewpoint is defined by feasible configuration $q_j \in Q$ and has a corresponding set of observed primitives, i.e. a specific subset of S. Solving the coverage sampling problem is done by finding the minimum number of viewpoints, with feasible configuration q_j, so that each primitive s_i is included in at least one subset of observed primitives associated with the viewpoints. When the geometry of the surfaces is known as well as the properties of the metrology system (e.g. field-of-view, image overlapping, shadow effects), as assumed in this work, a *model-based* methodology can be used [7].

In general terms, the coverage sampling typically starts with generating a discrete representation of the to-inspect surfaces as a set of geometric primitives (e.g. mesh of triangular faces). Note that this discretisation will affect the accuracy of the inspection. The next step is the viewpoint sampling, which employs a specific sampling strategy in order to generate the set of admissible viewpoints. The purpose is to obtain a set of admissible viewpoints, in a systematic way, that guarantees full coverage but might include unnecessary redundant viewpoints. The final step is to find the selected viewpoints that constitute the smallest subset in the admissible set that provides full coverage.

2.2. Multi-goal path planning

The *multi-goal path planning* problem, the second subproblem for the coverage path planning problem, is concerned with finding the sequence and path connecting the selected viewpoints in order to minimise the *travel cost* (i.e. the robot motion time to position the gauge at each viewpoint). The multi-goal refers to that there are multiple goals, i.e. viewpoints, that need to be reached along the path. In other words, the objective is to find the shortest kinematic feasible path for the robot to position the gauge at each viewpoint exactly ones, without colliding with any obstacle in the workspace. Finding a feasible collision-free path between the viewpoints can require one or more intermediate via-locations for the path so that the robot moves around the obstacles in the workspace. Hence, the multi-goal path planning methodology requires an obstacle avoidance technique to plan such intermediate via-locations.

3. Existing methodologies

This section discusses the existing state-of-the-art coverage path planning methodologies that have been considered in this work.

3.1. Coverage sampling

The viewpoint sampling strategy to obtain the viewpoints for the admissible set, together with the method to solve the uni-cost set-covering problem to select the optimal subset, are the two components that influence the solution of the resulting set of viewpoints for the coverage path. Several sampling strategies for the admissible viewpoints have been proposed.

Gonzalez-Banos [9] proposed a randomised art-gallery algorithm for coverage sampling whereby the admissible viewpoint set is generated by random sampling the workspace around the to-inspect surfaces. However, this sampling strategy does not guarantee full coverage of the to-inspect surface. The probability to achieve full coverage increases with the number of randomly sampled admissible viewpoints. A similar viewpoint sampling strategy is adopted in the methodology by Bircher et al. [10] and is proposed to be integrated within the proposed rapidly exploring "random tree of trees" path planning algorithm.

Raffaeli et al. [6] proposed a strategy that first clusters the primitives based on distance and surface normal direction in order to group primitives that can be covered from the same viewpoint. For each group, a viewpoint is sampled randomly that covers all primitives in the group and included in the coverage path. This significantly reduces the number of viewpoints however it struggles to guarantee full coverage.

Dornhege et al. [11] proposed a viewpoint sampling strategy that includes two different steps. The first step incorporates preliminary random viewpoint sampling, with relaxed constraints for evaluating the primitives' observability, in order to identify the most promising areas in the workspace for high-quality viewpoints. The second step again includes random sampling restricted to one of the identified most promising areas and using complete visibility constraints, to obtain admissible viewpoints. The number of admissible viewpoints is then equal to the number of identified promising areas, or thus to the threshold for an area to be considered promising.

Bircher et al. [12] proposed a strategy based around iteratively randomly resampling to find viewpoints that are closer to each other in order to reduce the robot motion time. It is assumed that a separate viewpoint is necessary for each geometric primitive of the discrete representation of the to-inspect surface. Consequently, the sole objective for the resampling strategy is to reduce the travel-cost between the viewpoints.

Vasquez-Gomez et al. [13] proposed a viewpoint sampling strategy where admissible viewpoints are obtained from an equidistant grid placed on the surface of a sphere around the to-inspect surfaces. For each node of the grid a viewpoint is generated and included in the admissible set.

Danner and Kavraki [14] proposed *dual sampling* strategy. For each primitive, m candidate admissible viewpoints that are able to observe that specific primitive are sampled. Next, the best viewpoint of these m candidates is selected for admissible set. The selection criteria is based on the number of primitives that can be observed from the viewpoints. In this way, there is at least one viewpoint in the admissible set for each primitive and full coverage is guaranteed.

In the work by Englot and Hover [15], an alternative strategy is proposed that also randomly samples m viewpoints for each primitive. It is proposed to include all m viewpoints in the admissible set since the solution quality improves by having a higher level of redundancy within the admissible viewpoint set.

Finding the smallest subset that provides full coverage within the admissible viewpoint set can typically be done by formulating this problem as a uni-cost set-covering problem, which is an integer-linear programming optimisation problem [16]. Depending on the number of primitives and the number of admissible viewpoints generated for each viewpoint, solving the set-covering problem can become challenging. When this is the case, heuristic optimisation methods (i.e. Greedy covering [16]) can be used instead to find an approximate solution.

Vasquez-Gomez et al. [13] proposed a greedy approximation method to find the viewpoints for the selected viewpoint set within the admissible set, as well as the sequence for the selected viewpoints. The greedy selection criteria include the number of observable primitives, travel cost from previous viewpoint, etc. The best n viewpoints according to these criteria (combined using linear scalarisation) are included in the set of selected viewpoints, until the $(n+1)^{th}$ viewpoint does not observe any unobserved primitives.

Jing [17] combines the set-cover problem to select the set of viewpoints from the sampling admissible viewpoints with the multi-goal path planning problem by formulating the two together as a special type of sequencing optimisation problem. This is to avoid the decoupling of the set-cover problem to select the optimal subset of admissible viewpoints and the multi-goal path planning problem.

3.2. Multi-goal path planning

The multi-goal path planning problem is concerned with finding the optimal sequence and collision-free paths to visit all selected viewpoints. This problem can be formulated as a travelling salesman problem, an extensively studied problem in combinatorial optimisation [16, 18]. Solving a travelling salesman problem is known to be particularly challenging with an increasing number of selected viewpoints to visit. Smaller instances of the travelling salesman problem can be formulated and solved as an integer linear programming problem. When the number of viewpoints to visit increases, this approaches becomes impractical. More powerful alternatives for solving these harder instances of the travelling salesman problem have been developed and implemented. Among the state-of-the-art are the Lin-Kernighan-Helsgaun heuristic proposed by Helsgaun [19] and the TSP-Concorde library [20].

It should be noted that in order to solve the travelling salesman problem, it is required to know the path and corresponding travel cost between all pairs of selected viewpoints. Global path planner such as sampling-based path planning methods to generate collision-free paths connecting the viewpoints are preferred for coverage path planning [14]. However for certain specific scenarios alternative methods can be more efficient and provide more elegant solutions [21]. Among the sampling-based path planning methods, the Rapidly-exploring Random Tree (RRT) method proposed by LaValle and Kuffner [22] and its variants are popular methods for coverage path planning [8, 17, 10] due to their performance for a variety of path planning problems.

Hence, when the number of admissible viewpoints increases, the number of viewpoint-pairs rapidly increases and calculating all corresponding travel cost can become impractical. In the work by Englot and Hover [8], calculating the travel cost for all viewpoint-pairs for the travelling salesman problem is circumvented. This is done by iteratively solving the travelling salesman problem initially for an optimistic lazy approximation of the travel cost between the viewpoint-pairs that is very easy to calculate. In each iteration, the approximated travel cost for all viewpoint-pairs included the planned multi-goal path is updated with the travel cost for the exact planned path.

4. Random versus targeted viewpoint sampling

One of the observation made during the literature study was that all existing coverage path planning methodologies rely on some form of random sampling in order to obtain the (admissible) viewpoints. The random sampling is done either within the entire robot workspace or within a smaller subspace of the workspace around one or more primitives. The quality of the planned coverage path is strongly affected by the sampled (admissible) viewpoints. In order to have a reliable viewpoint sampling methodology, a vast number of randomly sampled viewpoints are necessary. However, having many admissible viewpoints quickly becomes problematic for solving the set-covering problem that selects the optimal subset of viewpoint for the coverage path, even for approximate solutions.

Based on this, it is proposed to replace the random viewpoint sampling strategy with a targeted viewpoint sampling strategy. The motivation for the proposed targeted viewpoint sampling strategy is to generate a relatively small set of admissible viewpoints, according to a heuristic procedure, that still reliably yields a high quality coverage path. Such a targeted viewpoint sampling strategy is based around a search problem that is solved by optimisation. The search problem is formulated in order to find the "best" viewpoint based on the previously sampling viewpoints, and is then reformulated and solved iteratively in order to obtain a set of viewpoints. In other words, the work presented in this paper aims at developing a novel viewpoint sampling strategy that allows integrating objectives and constraints for sampling the admissible viewpoints so that these aid achieving the objective of the coverage path planning problem, e.g. minimising the cycle-time. In the remainder of this section, the search problem to find the optimal next viewpoint based on the previously sampled viewpoints is first presented and thereafter the proposed targeted viewpoint sampling strategy.

4.1. Targeted viewpoint sampling problem formulation

This section formulates the targeted viewpoint sampling problem, including the objectives in order to define the adopted meaning of an optimal viewpoint. Furthermore, the different constraints to consider in order to obtain feasible viewpoints are also presented. For the formulation of the viewpoint sampling problem, let $Q_i = \{q_i^1, q_i^2, \ldots, q_i^J\}$ be a robot pose specifying the position for each of the J robot joints for viewpoint i, and $s_i \in S$ be the subset of primitives that can be observed from viewpoint i. The set of primitives that represents the to-inspect surfaces will be written as S.

Objectives

There are two criteria that play a role to plan the quickest coverage path. Consequently, these two criteria translate into two objectives. The first criterion looks at having viewpoints that are able to observe many primitives at once since this will contribute to minimising the number of viewpoints to fully cover the surfaces. It is also important that each viewpoint covers a different subset of primitives than the other viewpoints. It is therefore proposed to only consider the primitives that have not been covered by the previously sampled viewpoints. The first objective maximises the number of not yet covered primitives that the gauge can observe from the viewpoint and is formulated as follows

$$f_{obs}(Q_i, S \setminus \{s_1 \ldots s_{i-1}\}) = \sum_{k=1}^{|S \setminus \{s_1,\ldots,s_{i-1}\}|} Pr(Q_i, s_k) \tag{1}$$

$$\text{with } Pr_k(Q_i, p_k) = \begin{cases} 1 & \text{if primitive } s_k \text{ is covered from } Q_i \\ 0 & \text{else} \end{cases}$$

with $f_{obs}(Q_i, S \setminus \{s_1 \ldots s_{i-1}\})$ being the function that evaluates the coverage of the given viewpoint Q_i by determining the number of primitives that can be observed by the gauge at that viewpoint, $\{s_1 \ldots s_{i-1}\}$ are the subsets of covered primitives by the previously sampled viewpoints $1 \ldots i-1$, s_k is the k^{th} primitive in the considered set of primitives $S \setminus \{s_1 \ldots s_{i-1}\}$, $Pr_k(Q_i, s_k)$ is the function that evaluates if primitive s_k is covered from viewpoint Q_i. This function utilises a model representation of the characteristics (e.g. field-of-view, depth-of-view, sensitivity to reflectivity) of the used metrology gauge and also requires geometric information of the primitives beyond the location, such as the surface normals.

The second criterion for the coverage path planning is concerned with having viewpoints that are located close to each other. The logic being that reducing the distance between the admissible viewpoints will aid to reduce the travel cost for the gauge to visit selected viewpoints. The second objective is thus the duration of the motions to move from the viewpoint Q_i to all previously sampled viewpoints. It is proposed to consider the motion duration for moving to all other viewpoints because the viewpoint sequence is not known at this stage. The second objective function is formulated as follows

$$f_{trav}(Q_i, Q_{1\ldots i-1}) = \sum_{j=1}^{i-1} f_{dur}(Q_i, Q_j) \tag{2}$$

where $f_{trav}(Q_i, Q_{1\ldots i-1})$ being the function that evaluates the travel cost indicator for the given viewpoint Q_i, $f_{dur}(Q_i, Q_j)$ is the function that calculates the duration of the motion to travel from viewpoint Q_i to viewpoint Q_j. The travel cost is equal to the sum of the duration of the motion to travel from the viewpoint Q_i under evaluation to each previously sampled admissible viewpoint. It should thus be noted that this does not take into account any sequence for visiting the sampled viewpoints, but is used as an approximate indicator for the travel costs. This function utilises algorithms to generate the motion for the point-to-point motions between viewpoints as well as for determining the timing of the trajectory. For this, a certain simplification can be made since the indicator is only used to compare different candidate viewpoints with each other.

Since there are two conflicting objectives, the viewpoint sampling optimisation problem is a multi-objective optimisation problem and it becomes necessary to determine how to handle the multiple objectives. On the one hand, to have a viewpoint that observes many not yet covered primitives, it needs to be significantly different from the previously sampled admissible viewpoints. On the other hand, to have a significantly different viewpoint, it needs to be located far away from the previously sampled admissible viewpoints, which results in a high travel-cost. In order to handle the multiple objectives, it is necessary to specify the desired trade-off between them. It is proposed to use the *linear-scalarisation* technique, also called the *weighted sum method* in order to combine the two objective functions in a scaled and balanced way. The combined multi-objective function can be written as follows

$$\begin{aligned} f_{obj}(Q_{1\ldots i}, S) = c_1 \cdot f_{obs}(Q_i, S \setminus \{s_1 \ldots s_{i-1}\}) \\ + c_2 \cdot f_{trav}(Q_i, Q_{1\ldots i-1}) \end{aligned} \tag{3}$$

with c_1 and c_2 being weighing factors to scale the values obtained by the two objective functions f_{obs} and f_{trav} as well as to balance the trade-off as desired for the specific problem at hand. Please note that, in the case of minimisation optimisation, the first of the two weighing factors needs to be negative ($c_1 < 0$) as the first objective function needs to be maximised and the other needs to be minimised.

Constraints

During the viewpoint sampling, several constraints need to be taken into account in order to guarantee that the sampled viewpoints are feasible considering the used robotic surface metrology technology for the dimensional inspection. A first constraint is concerned with the kinematics of the robotic system. When optimising the search problem to find the next optimal viewpoint during the sampling, it is necessary to evaluate that the viewpoint's pose Q_i is feasible in terms of the robot's kinematics. This constraint is formulated as follows

$$g_{kin}(Q_i) \geq 1 \tag{4}$$

with $g_{kin}(Q_i)$ the function to verify the robot kinematics and which returns the number of feasible robot poses for viewpoint Q_i. It involves solving the inverse kinematics for the gauge position and orientation in order to determine the feasible robot joint configurations to position the gauge accordingly.

The second constraint is concerned with the collisions between robotic metrology system and the obstacles in the workspace as well as with the to-inspect object. This constraint is formulated as follows

$$g_{col}(Q_i) \geq 1 \tag{5}$$

with $g_{col}(Q_i)$ being the function to verify that the robot poses for viewpoint Q_i are collision-free and which returns the number of collision-free robot-poses found. This involves performing a collision detection simulation based on the geometric models for the robot links, end-effector, part(s), fixture, obstacles, floor etc. and using interference calculation algorithms.

4.2. Targeted viewpoint sampling

This section presents the proposed targeted viewpoint sampling strategy. This strategy iteratively solves the formulated viewpoint sampling problem in order to obtain a set of high quality viewpoints for the admissible set. The required input includes: set of geometrical primitives that represents the to-inspect surfaces, kinematic model of the robotic system, geometrical models of obstacles in the workspace, as well as the characteristics of the used surface metrology system. The strategy iteratively reformulates and solves the search problem to find the optimal next viewpoint. It is necessary to reformulate the search problem each time a new viewpoint is obtained, i.e. after each iteration, since the formulation depends on the previously sampled viewpoints. This is repeated until the required full coverage of the to-inspect surfaces is reached with the sampled viewpoints. Englot and Hover [15] show that sampling more admissible viewpoints on top of a set that guarantees full coverage gives higher quality coverage paths. Based on this, the termination criterion for the viewpoint sampling becomes that each primitive is covered at least k times (i.e. $k \in \mathbb{N}^+$). The number k will be referred to as *'coverage-redundancy'* in this paper. The output is the smallest subset within the set of admissible viewpoints that provides full coverage of the to-inspect object with the desired coverage-redundancy k.

Workflow

This section presents the workflow for the proposed targeted viewpoint sampling strategy, which is schematically illustrated in Figure 1. Throughout the iterative sampling, the role of the set S_p is to continuously keep track of the primitives that need to be targeted by the next viewpoint. The set S_p is continuously updated to be the subset of primitives that have been covered the least number of times. Initially, the set S_p includes all primitives in S.

The first step (i.e. Step 1 in Figure 1) is to derive the search-space W_p to find the new viewpoint Q_i with i being the iteration-number. The search space $W_p \subseteq \mathbb{R}^6$ is the six-dimensional space specifying all positions (i.e. x, y, z-coordinates) and orientations (i.e. α, β, γ-angles) for the metrology gauge mounted as end-effector on the robot to cover the primitives in S_p. The search space is hereby adaptively reduced for the primitives in S_p, which significantly helps to solve the search problem for finding the next optimal viewpoint.

Next (i.e. Step 2 in Figure 1) is to deploy the optimisation algorithm to find the optimal new viewpoint Q_i that maximises

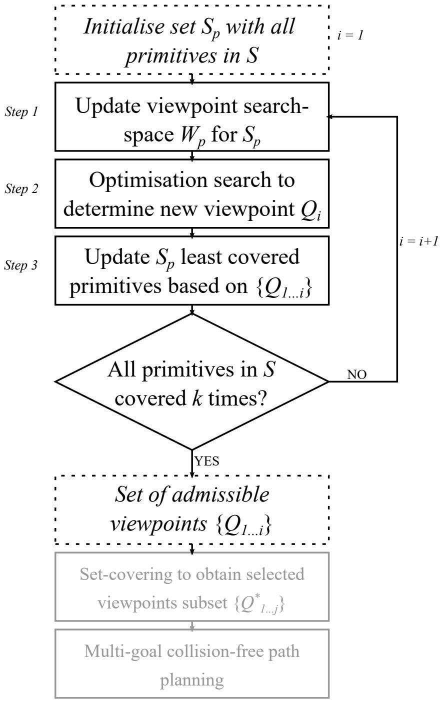

Figure 1: Flowchart of the proposed viewpoint sampling strategy, combined by set-covering and multi-goal path planning (in grey) into coverage path planning methodology

the number of covered primitives in S_p, as formulated in (1), while minimising the travel cost to the pose of the previously sampled viewpoints $\{Q_1 \dots Q_{i-1}\}$, as formulated in (2). The robot kinematics constraint from (4) and the collision avoidance constraint from (5) are also considered during the optimisation.

In the last step (i.e. Step 3 in Figure 1), the set S_p is updated for covered primitives by the new admissible viewpoint in order to include the subset of primitives that have been covered the least number of times by previously sampled admissible viewpoints $\{Q_1 \dots Q_i\}$. Only including the least-covered viewpoints in S_p, instead of all viewpoints that have not been covered k times, is crucial in order to avoid sampling k times the same viewpoint.

As long as all primitives in S have not been covered the required k times, these three different steps are repeated iteratively in order to find admissible viewpoints to cover the primitives in S_p. Finally, when this termination criterion has been reached, the generated set of admissible viewpoints is provided to formulate the set-covering problem in order to select the smallest subset within the admissible set that provides full coverage. This subset with selected viewpoints are then used for the coverage path.

As shown in Figure 1, the next step is to formulate the set-covering problem in order to find the smallest subset within the set of admissible viewpoints in order to achieve full coverage of the to-inspect surfaces. Two alternatives have been identified for this work. The first technique formulates the set-covering problem as an integer programming problem [23]. However, this becomes impractical when the number of admissible viewpoints increases (> 1000). In order to handle the case with a higher number of viewpoints, a greedy method is used to provide an approximate solution for set-covering problem.

Finally, the multi-goal path planning problem needs to be solved in order to obtain a feasible path that visits all the selected viewpoints $\{Q_1 \dots Q_i\}$, as shown in Figure 1. It is proposed to use the iterative solution for the travelling salesman problem [15].

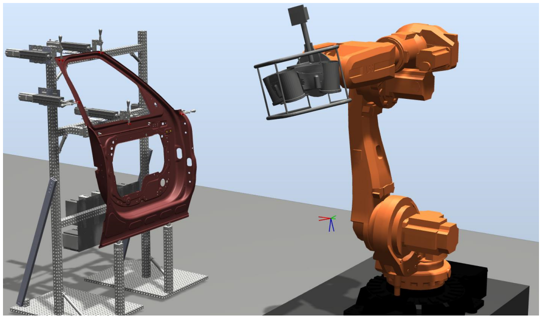

Figure 2: Robotic inspection station used in the considered case studies

5. Case studies

This section describes the case studies concerning dimensional inspection of sheet metal parts that were considered during the work for this paper. The used surface metrology technology is the CogniTens WLS400a white-light stereo-vision system from Hexagon Manufacturing Metrology mounted on a 6-DoF industrial manipulator arm robot of the type ABB IRB6620-150. The robot inspection cell is shown in Figure 2. The duration for taking a single observation at a viewpoint is around 4 seconds.

In order to evaluate the proposed strategy and compare against the state-of-the-art, tests have been performed for two different case studies considering different to-inspect surfaces. The surfaces are sheet metal parts that are real-world industrial components for the sub-assembly of an automotive door. In the first case study, the to-inspect sheet metal part is an inner door panel. Figure 3 shows the considered inner door panel as well as the geometric primitives (i.e. points) that provide the discrete representation of the part as black markers. In the second case study, the to-inspect sheet metal part is a window reinforcement frame for the inner door panel made. The window reinforcement frame panel is displayed in Figure 4 as well as the geometric primitives (i.e. points) that provide the discrete representation of the part are shown as black markers. For both case studies, the goal is to obtain viewpoints (and a path connecting them) that provide full coverage of the to-inspect sheet metal parts, i.e. covers all primitives representing the surface, displayed in Figures 3 and 4.

6. Feasibility analysis

The existing methodologies that were identified during the literature study, which was presented in Section 3, were analysed to investigate their feasibility for dimensional inspection

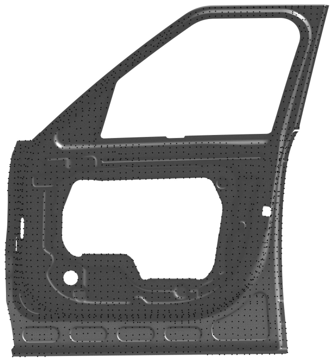

Figure 3: The inner door panel considered as to-inspect object in the first case study, the black markers show the 2283 geometric primitives that provide the discrete representation

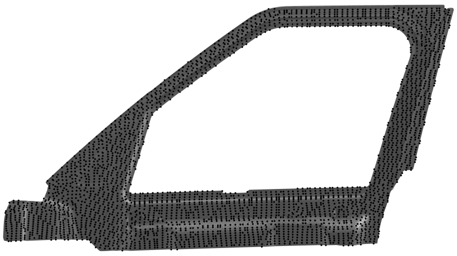

Figure 4: The window reinforcement panel considered as to-inspect object in the second case study, the black markers show the 3287 geometric primitives that provide the discrete representation

of sheet metal parts. For a coverage path planning methodology to be considered as feasible, it needs to be able to address the objectives and constraints presented in Section 2, which means that the number of viewpoints and the travel cost for the planned coverage path are minimised, the planned robot paths are kinematically feasible and collision-free, as well as full coverage of the to-inspect surfaces is guaranteed. This section presents the investigation and discusses the results summarised in Table 1, which columns refer to these four different criteria that have been used to evaluate the feasibility of the coverage path planning methodologies.

Several of the existing methodologies [9, 13, 11, 10] failed to achieve full coverage of the to-inspect surfaces of the sheet metal parts in the considered case studies. Many of these rely on randomly sampling the robot workspace around the to-inspect surfaces to generate admissible viewpoints. For the sheet metal parts, such as the ones considered in the case study, it was found that even when an impractically large number (>500,000) of admissible viewpoints are generated by random sampling, these still do not provide full coverage. Each time, the same subset of primitives remained uncovered. It turned out that these primitives can only be covered from a few specific viewpoints in the workspace. The probability that these specific viewpoints are found by random sampling is unreasonably low.

The methodologies proposed by Bircher et al. [12] manages to provide full coverage however the number of selected viewpoints was 300 times larger compared to other successful methodologies [14, 15]. The resulting total time for the inspection task will always be much longer, even when optimising the path using the proposed iterative resampling. The assumption that there is no need to minimise the number of viewpoints is thus unrealistic for dimensional inspection of sheet metal parts.

Table 1: Summary feasibility analysis of existing coverage path planning methodologies in terms of minimising the number of viewpoints (i.e. f_{obs} in (1)), minimising the travel cost (i.e. f_{trav} in (2)), guaranteeing kinematic feasible collision-free paths (i.e. g_{kin} in (4) and g_{col} in (1)), and guaranteeing full coverage (FC) of the to-inspect surfaces

Method	f_{obs}	f_{trav}	g_{kin}, g_{col}	FC
Vasquez-Gomez et al. [13]			x	
Bircher et al. [10]		x	x	
Gonzalez-Bonas et al. [9]	x			x
Raffaeli et al. [6]	x		x	
Dornhege et al. [11]	x	x	x	
Bircher et al. [12]		x	x	x
Danner and Kavraki [14]	x	x	x	x
Englot and Hover [8]	x	x	x	x
Jing et al. [17]	x	x	x	x
This paper	x	x	x	x

The viewpoint sampling strategy proposed by Raffaeli et al. [6] that is based around clustering the geometric primitives in groups based on distance and surface normal was also tested in this work. The results showed that it fails to provide full coverage of the to-inspect surfaces. This appears to be due to that there is no guarantee that all primitives in a group can be observed from a single viewpoint since the clustering criteria to group the primitives ignores the ray-tracing as well as the robot kinematics and collision-avoidance.

The methodologies proposed by Danner and Kavraki [14] and Englot and Hover [15], where an admissible viewpoint is sampled (randomly) for each primitive, manage to provide full coverage. It can thus be concluded that this viewpoint sampling strategy is the most suitable for coverage path planning for the dimensional inspection of sheet metal parts. Based on the performed analysis, it has been found that the iterative approach for solving the multi-goal path planning problem by Englot and Hover [8] the most suitable methodology for coverage path planning in the context of dimensional inspection of sheet metal parts. The main benefit of this methodology is that avoids having to generate collision-free paths to connect all possible viewpoint pairs.

The analysis of the methodology proposed by Jing [17] showed to be inefficient for the coverage path planning problem in the considered case studies. It took one hour of computations to obtain a solution of similar quality compared to five seconds of computations with a methodology by Englot and Hover [15] that uses the decoupled approach. This methodology is thus feasible for coverage path planning for the dimensional inspection of sheet metal parts. It was however not be further considered in this work due to its computational inefficiency.

Based this feasibility analysis, it was concluded that a methodology that decomposes the coverage path planning problem into three individual subproblems, i.e. (1) admissible viewpoint sampling, (2) set-covering for selected viewpoint subset, (3) multi-goal path planning, turns out to be most suitable for dimensional quality inspection of sheet metal parts. These three subproblems are considered individually in the feasible state-of-the-art methodologies proposed by Danner and Kavraki [14] and Englot and Hover [8].

7. Evaluation and Comparison

This section presents the evaluation of the proposed viewpoint sampling strategy for the two case studies and the comparison with existing state-of-the-art feasible coverage path planning methodologies. First, the implementation of the proposed targeted viewpoint sampling strategy is presented and the performed tests for the evaluation are described. Thereafter, the results of the comparison are presented and discussed.

7.1. Implementation

The proposed targeted viewpoint sampling strategy requires the characteristics of the surface metrology technology as input in order to determine the primitives that can be observed from a viewpoint, as formulated in (1). These include the field-of-view, ray-tracing, and light scatter. Firstly, the primitives that are within the truncated pyramid corresponding to field-of-view of the gauge when it is at the viewpoint are determined. This evaluation will be represented by the function f_{fov}. Secondly, for each of primitive in the field-of-view, ray-tracing is used to evaluate whether the line of view between the gauge and the primitive is no obstructed by other primitives. This evaluation will be represented by the function f_{ray}. Thirdly, the angle between the line of view and the surface-normal of the primitive is calculated in order to check that this is within the limitations of the gauge. This evaluation will be represented by the function f_{scat}. This objective can thus be formulated using the following function:

$$f_{obs}(Q_i, \{Q_{1...i-1}\}, S) = f_{scat}(f_{ray}(f_{fov}(Q_i, \{Q_{1...i-1}\}, S))) \quad (6)$$

with Q_i being the viewpoint that is currently being evaluated, $\{Q_{1...i-1}\}$ is the set of previously sampled viewpoints.

In order to implement the linear scalarisation objective function from (3), the weighing factors c_1 and c_2 to scale and balance the two objectives need to be specified. In this work, the trade-off is balanced so that it always prioritise the gains in number of observed primitives (i.e. f_{obs}) over travel cost (i.e. f_{trav}).

The Self-Adaptive Differential Evolution (SADE) algorithm [24] was used for the optimisation to find the best next viewpoint in Step 2 (see Figure 1) in the proposed viewpoint sampling strategy. The SADE algorithm was tuned until it manages to consistently find a good enough solution (i.e. 90%), which corresponds to 1000 viewpoint evaluations.

During the viewpoint sampling, the travel cost from the evaluated viewpoint to all previously sampled viewpoints is approximated since performing collision-free path planning for all these viewpoint-pairs becomes computationally expensive. This is approximated by considering the travel cost to move to the average robot-pose, i.e. difference between viewpoint-pose joint angle with the average joint-angles of all previously sampled viewpoints, is used instead. In this way, only a single travel-cost needs to be calculated when evaluating a viewpoint and the lazy travel-cost approximation is easy to compute.

The used multi-goal path planning method was adopted from [8]. It starts with calculating the lazy travel cost approximation for each viewpoint-pair. The travelling salesman problem (TSP) for finding the optimal sequence of viewpoints to minimise the travel cost is then solved. The TSP is solved as an integer programming problem [23], initially without subtour elimination constraints in order to reduce the size of the problem when the number of viewpoints increases. When the obtained optimal path solution includes subtours, specific constraints to eliminate those subtours are added to the formulation of the integer programming problem. The problem with its new formulation is solved again to obtain update the solution excluding those subtours. This is repeated until the solution is free of subtours. When a subtour-free path is found, the next step is to verify the travel-cost by path planning with collision-avoidance. This verified collision-free path planning is performed using the Rapidly-exploring Random Trees (RRT) [22] and post smoothing of the obtained collision-free path. In the next iteration, the TSP solved with these updated verified travel costs. The method iteratively continues to until the planned multi-goal path returns a path for which all included viewpoint-pairs already have exactly calculated travel-costs, thereby confirming that the solution is the optimal solution.

An alternative termination criterion was used, i.e. a maximum number of viewpoint evaluations, then k is unspecified and the method continues to find new admissible viewpoints to observe the least covered primitives until this maximum number of viewpoint evaluations is reached. The motivation for this is to control the number of viewpoint evaluation in order to have a fair comparison with the existing state-of-the-art feasible coverage path planning methodologies.

7.2. Tests

Several tests have been conducted to evaluate the performance of the proposed viewpoint sampling strategy, which will be referred to as *targeted viewpoint sampling (TarSamp)*. Its performance is compared with existing state-of-the-art methodologies that were found to be feasible within the context of dimensional inspection of sheet metal parts, as discussed on Section 6. This includes the methodology proposed by Englot and Hover [8], which will be referred to as the *redundant viewpoint sampling (RedunSamp)*, as well as the methodology proposed by Danner and Kavraki [14], which will be referred to as the *dual viewpoint sampling (DualSamp)*.

The comparison evaluates the number of viewpoints as well as the travel cost for the path connecting all viewpoints since both need to be minimised in order to reduce the cycle-time of the inspection task, as discussed Section 2. The travel cost for the coverage paths is presented as a time-duration that is estimated based on the travel distance for each robot joint to follow the sub-tour path and the maximum robot joints' velocities, while neglecting the robot dynamics by assuming instantaneous accelerations. This simplification does not affect the test results for the purpose comparing the different viewpoint sampling strategies. Two different travel cost duration indicators are presented. The first one, i.e. the approximated travel cost, is solely based on the estimated duration for the subtour-free coverage path, while neglecting collision avoidance. This is included because it is a more direct indication of viewpoint sampling performance, since only the approximated travel cost is considered during viewpoint sampling. The second travel cost indicator is the verified travel cost that is based on the estimated duration of the planned subtour-free collision-free path, i.e. via-location have been included where necessary to avoid collisions between the robot, end-effector, floor, part, fixture, obstacles, etc.

As discussed in Section 4.2, two different techniques to solve the set-covering problem are considered in order to select the smallest subset in the admissible set that provides full coverage. The first is by solving the set-covering problem as an unicost integer linear programming (ILP) problem, however it was found this became prohibitively expensive when the number of admissible viewpoints exceeds 1000. In those cases, the greedy approximation method to solve the set-covering problem was used. Solving the set-covering problem as an ILP problem gives better solutions than the greedy approximation. The number of admissible viewpoints is influenced by the used viewpoint sampling strategy but is in the first place dependent on the to-inspect surfaces. In order to perform an even-handed comparison, two sets of results are presented for the cases where there were less than 1000 viewpoints in the admissible set, i.e. one using ILP and another using the greedy approximation.

The presented results are averages of multiple repetitions (i.e. 30 for case study 1 and 50 for case study 2), and an ANOVA study was performed to evaluate whether there are significant difference between the average results for the different methods and test configurations. This is necessary due to the stochastic character of the viewpoint sampling methodologies. Multiple test configurations for the viewpoint sampling are considered which differ in maximum number of viewpoint evaluations during sampling in order to investigate how the number of viewpoint evaluations, and consequently the number of sampled viewpoints affect the results. Changing the maximum number of viewpoint evaluations corresponds to changing the admissible viewpoint sampling redundancy. For RedunSamp, this corresponds to changing the number of admissible viewpoints sampled for each primitive. For DualSamp, this corresponds to change the number of viewpoints samples generated for each primitive and from which the best one is then selected for the admissible set.

7.3. Results

This section presents the results of testing the proposed targeted viewpoint sampling strategy TarSamp for the two case studies and the comparison with the feasible state-of-the-art methodologies, RedunSamp and DualSamp.

Case study 1

The results for the coverage path to inspect the inner door panel (see Figure 3) are shown in Table 2. The second and third column in Table 3 shows the used set-covering method, i.e. greedy approximation (i.e. greedy approx.) or integer linear programming (i.e. ILP), and the configuration for the test. Three different test configurations are considered, with a maximum number of viewpoint evaluations of 250,000; 125,000; 60,250, corresponding to sampling 10, 5, 3 feasible viewpoints for each primitive with RedunSamp and DualSamp. The fourth column in Table 2 shows the number of viewpoints included in the coverage path. The fifth and sixth column show the approximated travel cost (i.e. ATC) and verified travel cost (i.e. VTC).

Starting the evaluation of TarSamp with greedy approximation set-covering by investigating its performance for minimising the number of viewpoints. It can be seen in Table 2 that it outperforms RedunSamp and DualSamp. Across the three test configuration, TarSamp gives on average around 8.1 % and 17.2 % fewer viewpoints respectively compared to RedunSamp and DualSamp. Based on these results, it can be said that TarSamp provides admissible viewpoints with a better coverage of the to-inspect surfaces and thereby allows to reduce the number of viewpoints in the coverage paths.

When looking at the travel cost results, it can be seen in Table 2 that TarSamp gives significantly better results compared to RedunSamp and DualSamp. Across the three test configurations, the approximated travel cost with TarSamp is 19.9 % (3.0 s) and 22.1 % (3.5 s) shorter respectively compared to RedunSamp and DualSamp. Similarly, the results in Table 2 show that the verified travel cost with TarSamp is 20.1 % (3.3 s) and 22.7 % (3.7 s) shorter respectively compared to RedunSamp and DualSamp. It can thus be said that TarSamp contributes significantly to reducing the travel cost of the resulting coverage paths.

For this case study, the to-inspect surfaces were represented by 2283 geometric primitives. This meant that with RedunSamp and DualSamp, there were always (at least) 2283 admissible viewpoints. It was therefore not feasible to use ILP for the set-covering, and only the greedy approximation could be used. TarSamp generated 60 up to 250 admissible viewpoints depending on the specific test configuration, which does allow using ILP for the set-covering.

The results presented in Table 2 also allow to compare the performance of the greedy approximation (greedy approx.) for the set-covering using ILP for TarSamp. The results show that using ILP gives on average 5.5 % fewer viewpoints than the greedy approximation, across the three test configurations. The analysis showed that there is however no significant difference for the travel cost results. It should be noted that, on the one hand, for the test configuration with the highest viewpoint sampling redundancy, ILP performs around 11.6 % better than the greedy approximation. On the other hand, there is no significant difference between ILP and the greedy approximation for the results with the lowest viewpoint sampling redundancy configuration. This is further investigated for case study 2. It should

Table 2: Results for case study 1: number of viewpoints (# VPs), approximated travel cost (ATC), and verified travel cost (VTC) of the coverage paths

Viewpoint sampling method	Set-covering method	# VP-evals	# VPs mean	# VPs std	ATC [s] mean	ATC [s] std	VTC [s] mean	VTC [s] std
TarSamp	greedy approx.	250,000	60.87	2.446	11.95	0.509	12.32	0.519
RedunSamp 10	greedy approx.	250,000	**66.63**	1.732	**15.26**	1.681	**15.86**	1.796
DualSamp 10	greedy approx.	250,000	**76.20**	2.809	**15.91**	1.855	**16.41**	1.941
TarSamp	ILP	250,000	**53.80**	1.400	11.18	0.463	11.51	0.527
TarSamp	greedy approx.	125,000	63.43	2.609	12.35	0.489	12.74	0.623
RedunSamp 5	greedy approx.	125,000	**68.73**	2.258	**15.33**	1.131	**16.00**	1.269
DualSamp 5	greedy approx.	125,000	**76.40**	2.908	**15.77**	1.455	**16.30**	1.635
TarSamp	ILP	125,000	**60.10**	2.155	11.90	0.504	12.24	0.637
TarSamp	greedy approx.	60,250	64.57	1.775	12.44	0.452	12.75	0.504
RedunSamp 3	greedy approx.	60,250	**70.13**	1.525	**15.25**	1.029	**15.93**	1.266
DualSamp 3	greedy approx.	60,250	**75.63**	2.659	**15.49**	1.110	**16.19**	1.416
TarSamp	ILP	60,250	64.77	1.888	12.46	0.444	12.75	0.476

Statistically significant different results compared to corresponding TarSamp with greedy approx. result are highlighted in **bold**.

also be noted that when looking at the results for the different test configurations for each individual methodology, there is no significant indication that the viewpoint sampling redundancy has an effect on the minimisation of the travel cost.

When comparing the performance of TarSamp with ILP against the other methods RedunSamp and DualSamp, it can be seen in Table 2 that there is an even more significant improvement compared to using the TarSamp with the greedy approximation for the minimisation of the number of viewpoints in the coverage path. The average difference is 13.2 % compared to RedunSamp and 21.7 % compared to DualSamp.

Case study 2

The results for the coverage path to inspect the window reinforcement frame panel (see Figure 4) are presented in Table 3. The table is organised in the same way as Table 2. Six test configurations are considered with maximum number of viewpoint evaluations of $660,000$; $495,000$; $330,000$; $165,000$; $110,000$; $38,000$, corresponding to sampling $20, 15, 10, 5, 3, 1$ feasible viewpoints for each primitive with RedunSamp and DualSamp. The test configuration (i.e. 38,000) without any viewpoint sampling redundancy is included to provide results in order to substantiate the conclusions concerning the effects of viewpoint sampling redundancy. For the same reason, two additional configurations are included with a higher redundancy.

Starting the evaluation of TarSamp with the greedy approximation set-covering by analysing its performance for minimising the number of viewpoints. It can be seen in Table 4 that TarSamp provides significantly better solutions. Across the six test configurations, TarSamp performs 3.7 % better than RedunSamp and 6.9 % better than DualSamp. This further confirms that TarSamp significantly outperforms RedunSamp and DualSamp concerning the minimising the number of viewpoints in the coverage path.

The results concerning reducing the travel cost also confirm that TarSamp performs significantly better than RedunSamp and DualSamp. Looking at the approximated travel cost, TarSamp gives 9.2 % (0.6 s) and 11.4 % (0.8 s) reduction on average across the six test configurations, respectively compared to RedunSamp and DualSamp. Similarly, TarSamp gives a 9.6 % (0.7 s) and 11.5 % (0.8 s) reduction in verified travel cost on average across the six test configurations, respectively compared to RedunSamp and DualSamp.

For this case study, the to-inspect surfaces were represented by 3287 geometric primitives, which meant that there were (at least) the same number of viewpoints in the admissible set generated by RedunSamp and DualSamp. It was therefore not possible to use ILP set-covering in combination with these two methodologies, and instead only results with the greedy approximation set-covering method are presented. For TarSamp, the number of viewpoints in the admissible set was 660 at maximum and ILP set-covering could be used for all test configurations. Results for TarSamp in combination with both greedy approximation as well as ILP set-covering methods are presented in Table 3.

Comparing TarSamp with the greedy approximation set-covering against ILP set-covering, a more pronounced difference was observed than in case study 1. For all test configurations, there is a significant difference between the results obtained with two different set-covering methods combined with TarSamp. ILP set-covering gives 23.8 % (3.5) fewer viewpoints compared to the greedy approximation, on average across the six test configurations. In contrast to case study 1, there is a significant difference between the travel cost results. On average across the six test configurations, ILP gives 3.3 % (0.2 s) lower in both for the approximated and verified travel cost in comparison with the greedy approximation set-covering.

As mentioned earlier, additional test configurations have been included for this case study in order to investigate the effect of admissible viewpoint sampling redundancy. Across the six test configurations, there is a significant trend indicating that the number of viewpoints in the coverage decreases when the admissible viewpoint sampling redundancy increases. It can thus be concluded that admissible viewpoint sampling redun-

Table 3: Results for case study 2: number of viewpoints (# VPs), approximated travel cost (ATC), and verified travel cost (VTC) of the coverage paths

Viewpoint sampling method	Set-covering method	# VP-evals	# VPs		ATC [s]		VTC [s]	
			mean	std	mean	std	mean	std
TarSamp	greedy approx.	660,000	18.32	0.621	6.073	0.141	6.073	0.140
RedunSamp 20	greedy approx.	660,000	**18.56**	0.861	**6.694**	0.472	**6.767**	0.479
DualSampl 20	greedy approx.	660,000	**19.52**	1.216	**7.161**	0.345	**7.180**	0.439
TarSamp	ILP	660,000	**14.00**	0.000	**5.864**	0.280	**5.876**	0.273
TarSamp	greedy approx.	495,000	18.12	0.435	6.026	0.128	6.026	0.128
RedunSamp 15	greedy approx.	495,000	**18.60**	0.808	**6.736**	0.492	**6.816**	0.479
DualSampl 15	greedy approx.	495,000	**19.72**	1.161	**7.080**	0.456	**7.131**	0.537
TarSamp	ILP	495,000	**14.00**	0.000	**5.919**	0.314	**5.926**	0.326
TarSamp	greedy approx.	330,000	18.36	0.851	6.189	0.280	6.217	0.342
RedunSamp 10	greedy approx.	330,000	**18.92**	0.966	**6.725**	0.389	**6.818**	0.413
DualSampl 10	greedy approx.	330,000	**19.94**	1.077	**6.893**	0.439	**6.897**	0.458
TarSamp	ILP	330,000	**14.18**	0.388	**5.949**	0.249	**5.949**	0.247
TarSamp	greedy approx.	165,000	18.42	1.012	6.217	0.315	6.243	0.360
RedunSamp 5	greedy approx.	165,000	**19.36**	1.139	**6.785**	0.405	**6.795**	0.419
DualSampl 5	greedy approx.	165,000	**19.76**	1.254	**6.862**	0.431	**6.897**	0.473
TarSamp	ILP	165,000	**14.96**	0.450	**5.987**	0.307	**5.999**	0.305
TarSamp	greedy approx.	110,000	18.70	0.995	6.219	0.325	6.235	0.340
RedunSamp 3	greedy approx.	110,000	**19.54**	0.973	**6.871**	0.385	**6.887**	0.407
DualSampl 3	greedy approx.	110,000	**19.98**	1.078	**6.875**	0.359	**6.874**	0.361
TarSamp	ILP	110,000	**15.46**	0.676	**6.024**	0.309	**6.023**	0.308
TarSamp	greedy approx.	38,000	18.96	1.160	6.266	0.425	6.272	0.407
RedunSamp 1	greedy approx.	38,000	**20.18**	1.155	**6.918**	0.380	**6.933**	0.408
DualSampl 1	greedy approx.	38,000	**20.18**	1.155	**6.918**	0.380	**6.933**	0.408
TarSamp	ILP	38,000	**17.38**	1.276	**6.068**	0.381	**6.110**	0.426

Statistically significant different results compared to corresponding TarSamp with greedy approx. result are highlighted in **bold**.

dancy contributes to reducing the number of viewpoints in the coverage path. However, for all three viewpoint sampling methods in the comparison, there is no significant trend for the first three test configurations, i.e. 660,000; 495,000; 330,000. This indicates that there is an upper limit for the admissible viewpoint sampling redundancy above which there is no further improvement in terms of minimising the number of viewpoints. Furthermore, the results also show that there is no significant trend concerning the effect of the admissible viewpoint sampling redundancy on the minimisation of the travel cost.

7.4. Computation time

Upon request of the reviewers, this section presents a comparison of the computation time for the different coverage path planning methodologies that are evaluated in this work. In order to attribute any differences in computation time to particular step(s) in the methodologies, the different subproblems (i.e. viewpoint sampling, set-covering, multi-goal path planning) are considered individually in this comparison. Starting with the viewpoint sampling, it is most important to note that the computation time is determined by the viewpoint evaluation procedure. For the implementation used in this work, the time for a single viewpoint evaluations was always closely around 18 milliseconds. The number of viewpoint evaluations is the same for all methodologies within each test configuration, which was chosen to perform a fair comparison between the different viewpoint sampling methodologies. This has as a consequence that there is no significant difference between the computation time for the viewpoint sampling of the different methodologies. The computation time is around 11 minutes for the test configuration with 38,000 viewpoint evaluations, and up to 3 hours and 20 minutes for the test configuration with 660,000 viewpoint evaluations.

Two different methodologies have been used for solving the set-covering to select the viewpoints for the coverage path, i.e. integer linear programming (ILP) and using a greedy approximation methodology (i.e. greedy approx.), the former only with TarSamp. The results for the comparison of the computation time are shown in Table 4. On the one hand, it can be seen that the computation time for the greedy approximation is always very closely around 1 second, across all test configurations. It can thus be said that the number of admissible viewpoints does not influence the computation time of the greedy approximation for solving the set-covering. On the other hand, it can be seen that the average computation time for solving the set-covering by integer linear programming is significantly larger and is influenced by the number of admissible viewpoints. It starts from 15.6 seconds, i.e. for the test configuration with the lowest number of admissible viewpoints, up to 1062 seconds with the highest number of admissible viewpoints.

Table 4: Comparison of the computation time for solving the set-covering problem for case study 2 using a greedy approximation (greedy approx.) versus by integer linear programming (ILP)

Configuration	Computation time [s]			
	greedy approx.		ILP	
	mean	std	mean	std
38,000	0.98	0.02	15.6	24.7
165,000	0.98	0.04	167	168
330,000	1.04	0.04	224	197
660,000	1.06	0.04	1062	1712

Finally, the same multi-goal path planning methodology was used for all different coverage path planning methodologies in the comparison in this work. Hence, no comparison of computation time has been performed for this. To give an indication of the computation time, the multi-goal path planning for case study 2 never took longer than 150 seconds.

7.5. Discussion

It can be seen that TarSamp performs significantly better than the other methods in the comparison. It is important to note that the advantage of TarSamp is two-fold. On the one hand, there is the improvement in solution quality for the coverage path when using TarSamp to generate the set of admissible viewpoints. This refers to the significantly better coverage paths, both in terms of minimising the number of viewpoints and the travel cost, when using TarSamp (with greedy approximation set-covering) compared to RedunSamp and DualSamp. These results are obtained using the same coverage path planning methodology, except the different viewpoint sampling differs. Therefore, the improved solution quality of the final coverage path can be attributed to proposed targeted viewpoint sampling strategy integrated in TarSamp.

On the other hand, the proposed targeted viewpoint sampling strategy integrated in TarSamp needs to be credited for generating significantly smaller number of viewpoints for the admissible set. Across the different case studies and test configurations, the number of viewpoints in the admissible set generated by TarSamp is at least 98 % smaller compared to RedunSamp, and 79.92 % up 98.84 % smaller compared to DualSamp. Having fewer but better admissible viewpoints is beneficial since set-covering problem can then be solved more effectively, and thereby having a further reduction of the number of viewpoints in the coverage path. This is a secondary advantage of the proposed viewpoint sampling strategy.

It is however necessary to make a remark about the dependency of the proposed viewpoint sampling methodology TarSamp on the convergence of the optimisation algorithm deployed during the viewpoint sampling and the ability to reformulate the search space for each iteration. The advantage of TarSamp over the other methods in the comparison goes hand in hand with the number of viewpoint evaluations that are required to find the viewpoint with maximised coverage of the least covered primitives. The ability to reformulate the objectives, constraints as well as the search space for finding the viewpoint at each iteration helps significantly to reduce the required number of viewpoint evaluations.

Interestingly, it was found that for both case studies the viewpoint sampling redundancy only contributes to minimising the number of viewpoints, and not to minimise the travel cost. Even for the cases where there was a significant reduction in the number of viewpoints, the travel cost remains similar with an increase of the viewpoint sampling redundancy.

From the investigation of the computation time, it can be concluded that the sampling of the admissible viewpoints is the main contributor to the computation time of the coverage path planning methodology. Whereas solving the set-covering problem (particularly when using the greedy approximation method) and the multi-goal path planning can be computed relatively quickly compared to the iterative viewpoint sampling. This reveals that improving the viewpoint sampling is critical in order to minimise the computation time for the coverage path planning, which adds to the relevance of the proposed targeted viewpoint sampling strategy.

8. Conclusions

The goal of the presented work in this paper was to investigate coverage path planning for dimensional quality inspection of sheet metal parts. First, the details of the coverage path planning problem in the context for this specific application were presented, including the objective for minimising the cycle-time for the inspection task as well as the relevant constraints for the robot kinematics, collision-avoidance and full coverage of the to-inspect surfaces. A summary of the performed literature study to identify existing state-of-the-art methodologies is given. Each of these are analysed to evaluate their feasibility for solving the coverage path planning for dimensional quality inspection of sheet metal parts considering the specific objectives and constraints. This showed that the identified feasible methodologies adopt an approach that includes decomposing the coverage path planning problem into three individuals subproblems: (1) viewpoint sampling, (2) set-covering to select viewpoint subset, (3) multi-goal path planning.

During the analysis of the existing coverage path planning methodologies, it was also observed that all rely on some form of random viewpoint sampling. Based on this, it was set out to investigate whether a targeted viewpoint sampling strategy, instead of random sampling, contributes to improving the coverage path solution quality. A targeted viewpoint sampling strategy is proposed and evaluated by comparing its performance against the identified feasible state-of-the-art methods. The results showed that the proposed targeted viewpoint sampling strategy generates significantly better quality solutions for the coverage path, both in terms of minimisation of the number of viewpoints as well as travel cost for the planned path to move to each viewpoint in an optimal sequence. The proposed targeted viewpoint sampling strategy performs 3.7 % up to 23.8 % better in minimising the number of viewpoints and 9.6 % up to 22.7 % better in minimising the travel cost, compared to state-of-the-art methods.

On the one hand, the presented work identifies the most suitable coverage path planning methodology for dimensional inspection of sheet metal parts with guaranteed full coverage, collision-free robot paths and minimised cycle-time. On the other hand, it was shown that adopting a targeted viewpoint sampling strategy gives significantly better coverage paths compared to using random viewpoint sampling. Future work could include extending the proposed targeted viewpoint sampling strategy for adaptive autonomous positioning for robot vision systems [25] as well as the multi-goal path planning for multi-robot system (i.e. multiple inspection robots and/or robotic material handling) and integrate trajectory optimisation to minimise energy consumption [26].

Acknowledgements

This work was performed at the *Digital Lifecycle Management (DLM)* research group of *WMG* at the University of Warwick in Coventry (UK) and supported by the *UK EPSRC* project EP/K019368/1: *Self-Resilient Reconfigurable Assembly Systems with In-process Quality Improvement*.

References

[1] Y. Ding, P. Kim, D. Ceglarek, J. Jin, Optimal sensor distribution for variation diagnosis in multistation assembly processes, IEEE Transactions on Robotics and Automation 19 (4) (2003) 543–556.

[2] M. Babu, P. Franciosa, D. Ceglarek, Spatio-Temporal Adaptive Sampling for Effective Coverage Measurement Planning during Quality Inspection of Free Form Surfaces using Robotic 3d Optical Scanner, Journal of Manufacturing Systems.

[3] P. Olivieri, L. Birglen, X. Maldague, I. Mantegh, Coverage path planning for eddy current inspection on complex aeronautical parts, Robotics and Computer-Integrated Manufacturing 30 (3) (2014) 305–314.

[4] J. Molina, J. E. Solanes, L. Arnal, J. Tornero, On the detection of defects on specular car body surfaces, Robotics and Computer-Integrated Manufacturing 48 (2017) 263–278.

[5] B. Li, P. Feng, L. Zeng, C. Xu, J. Zhang, Path planning method for on-machine inspection of aerospace structures based on adjacent feature graph, Robotics and Computer-Integrated Manufacturing 54 (2018) 17–34.

[6] R. Raffaeli, M. Mengoni, M. Germani, F. Mandorli, Off-line view planning for the inspection of mechanical parts, International Journal on Interactive Design and Manufacturing 7 (1) (2013) 1–12.

[7] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, G. Cai, A survey on inspecting structures using robotic systems, International Journal of Advanced Robotic Systems 13 (6) (2016) 1–18.

[8] B. Englot, F. S. Hover, Sampling-Based Coverage Path Planning for Inspection of Complex Structures., in: Twenty-Second International Conference on Automated Planning and Scheduling, 2012, pp. 29–37.

[9] H. Gonzalez-Banos, A Randomized Art-gallery Algorithm for Sensor Placement, in: Proceedings of the Seventeenth Annual Symposium on Computational Geometry, SCG '01, ACM, New York, NY, USA, 2001, pp. 232–240.

[10] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, R. Siegwart, Receding horizon path planning for 3d exploration and surface inspection, Autonomous Robots 42 (2) (2018) 291–306.

[11] C. Dornhege, A. Kleiner, A. Kolling, Coverage search in 3d, in: 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics, 2013, pp. 1–8.

[12] A. Bircher, M. Kamel, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, R. Siegwart, Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots, Autonomous Robots 40 (6) (2016) 1059–1078.

[13] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, E. Lopez-Damian, Volumetric Next-best-view Planning for 3d Object Reconstruction with Positioning Error, International Journal of Advanced Robotic Systems 11 (10) (2014) 1–13.

[14] T. Danner, L. E. Kavraki, Randomized planning for short inspection paths, in: 2000 IEEE International Conference on Robotics and Automation, Vol. 2, 2000, pp. 971–976 vol.2.

[15] B. Englot, F. Hover, Planning Complex Inspection Tasks Using Redundant Roadmaps, in: Robotics Research, Springer Tracts in Advanced Robotics, Springer, 2011, pp. 327–343.

[16] J. Lundgren, M. Rnnqvist, P. Vrbrand, Optimization, 1st Edition, Studentlitteratur AB, Lund, 2010.

[17] W. Jing, Coverage Planning for Robotic Vision Applications in Complex 3d Environment, PhD Thesis, Carnegie Mellon University (2017).

[18] D. L. Applegate, R. E. Bixby, V. Chvatal, W. J. Cook, The Traveling Salesman Problem A Computational Study, Princeton Series in Applied Mathematics, Princeton University Press, 2006.

[19] K. Helsgaun, An effective implementation of the LinKernighan traveling salesman heuristic, European Journal of Operational Research 126 (1) (2000) 106–130.

[20] D. L. Applegate, R. E. Bixby, V. Chvatal, W. J. Cook, Concorde TSP Solver (2006).
URL `http://www.tsp.gatech.edu/concorde/`

[21] L. E. Kavraki, S. M. LaValle, Motion Planning, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer Berlin Heidelberg, 2008, pp. 109–131.

[22] S. M. Lavalle, J. J. Kuffner, Rapidly-Exploring Random Trees: Progress and Prospects, in: Algorithmic and Computational Robotics: New Directions, 2000, pp. 293–308.

[23] S. P. Bradley, A. C. Hax, T. L. Magnanti, Chapter 9. Integer Programming, in: Applied Mathematical Programming, Reading, MA: Addison-Wesley Publishing Company, 1977, pp. 272–319.

[24] J. Brest, A. Zamuda, I. Fister, B. Boskovic, Some Improvements of the Self-Adaptive jDE Algorithm, in: IEEE Symposium on Differential Evolution (SDE), 2014, pp. 1–8.

[25] M. A. Montironi, P. Castellini, L. Stroppa, N. Paone, Adaptive autonomous positioning of a robot vision system: Application to quality control on production lines, Robotics and Computer-Integrated Manufacturing 30 (5) (2014) 489–498.

[26] E. Glorieux, S. Riazi, B. Lennartson, Productivity/energy optimisation of trajectories and coordination for cyclic multi-robot systems, Robotics and Computer-Integrated Manufacturing 49 (2018) 152–161.