A Framework for Coverage Analysis of Large and Complex Structures by Multiple Autonomous Industrial Robots

Penglei Dai, Mahdi Hassan, Xuerong Sun, Ming Zhang, and Dikai Liu

***Abstract*— Coverage analysis is essential for many coverage tasks (e.g., robotic grit-blasting, painting and surface cleaning) performed by Autonomous Industrial Robots (AIRs). Coverage analysis enables (1) the performance evaluation (e.g., coverage rate and operation efficiency) of AIRs for a coverage task, and (2) the configuration design of a multi-AIR system (e.g., decision on the number of AIRs to be used). Multi-AIR coverage analysis of large and complex structures involves addressing various problems. Thus, a framework is presented in this paper that incorporates various modules (e.g., AIR reachability, AIR base placement, collision avoidance, and area partitioning and allocation) for appropriately addressing the associated problems. The modules within the framework provide the flexibility of utilizing different methods and algorithms, depending on the requirements of the target application. The framework is tested and validated by extensive analyses of 10 different scenarios with up-to 10 AIRs.**

I. Introduction

The fast development of robotic technologies is significantly improving the quality and efficiency of many industrial applications [1]. Unlike preprogrammed industrial robots that have been widely implemented for repetitive mass productions within structured and unchanging environments, Autonomous Industrial Robots (AIRs) are able to perform more complex and challenging tasks in unstructured environments. AIRs are industrial robots that have self-awareness and environmental awareness enabling them to operate autonomously in unknown or partially known environments [1], where prior map of the structure is not available and various changes may occur in the environment.

To improve productivity and efficiency, some applications may need multiple AIRs operating in a collaborative and effective manner. As an important category of these applications, coverage tasks may require multiple AIRs to process surfaces of a large and complex target object, e.g., collaborative grit-blasting operation [1]. Enabling multiple AIRs to conduct a coverage task on an object is a challenging problem [2]. The associated subproblems (e.g., AIR reachability to the surface, AIR base placement, collision avoidance, and area partitioning and allocation) need to be properly solved to achieve safe and efficient collaboration among AIRs. Most of the existing research works mainly focus on solving one of the above subproblems for a specific application. In this work, various relevant subproblems of coverage analysis are integrated as part of a framework that is particularly designed for large and complex structures. Note that, in this paper, the scope of coverage analysis mainly includes two aspects as follows:

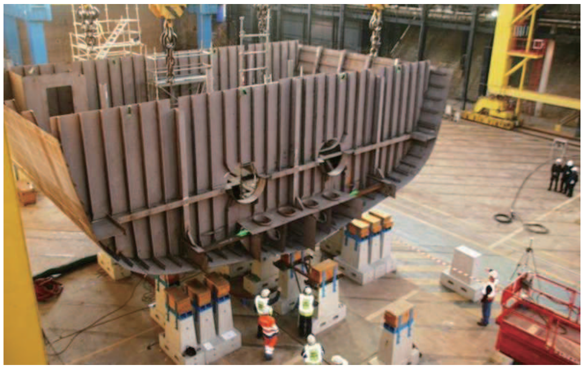

Fig. 1. Example of a ship-hull block [3]

- Coverage rate: the percentage of a structure's surfaces that can be covered by the AIRs.
- Operation efficiency: the operation time of the AIRs for achieving the coverage rate.

The manufacturing and maintenance of large and complex structures (e.g., airplanes, steel bridges and ship hulls) normally involve intensive labor and hazardous working environments. For example, in the rust removal process of a large ship-hull block (i.e., ship-hull section) as shown in Fig. 1, a dozen of workers may need to simultaneously conduct grit-blasting operations for several hours in an enclosed and dusty blasting room. Therefore, AIRs have great potential to replace human workers in performing coverage tasks (e.g., painting, surface cleaning and grit-blasting) on such structures. When utilizing multiple AIRs in such scenarios, it is essential to conduct coverage analysis to obtain critical information such as:

- the maximum coverage rate that can be achieved by the AIRs,
- the operation efficiency of completing the achievable coverage rate relative to the number of AIRs, and
- the surfaces that cannot be covered by the AIRs.

A proper coverage analysis enables: 1) the performance evaluation of AIRs (i.e., coverage rate and operation efficiency); 2) the configuration design of a multi-AIR system (e.g., the selection of AIRs, the decision on the number of

Penglei Dai, Mahdi Hassan and Dikai Liu are with the Centre for Autonomous Systems (CAS) at the University of Technology Sydney (UTS), Australia. e-mail: penglei.dai@uts.edu.au; mahdi.hassan@uts.edu.au; dikai.liu@uts.edu.au

Xuerong Sun and Ming Zhang are with China Merchants Heavy Industry (Jiangsu) Co., Ltd, China. e-mail: sunxuerong@cmhk.com; zhangming1@cmhk.com

AIRs to be used, and the mounting or locomotion strategy of the AIRs). However, to the best of authors' knowledge, there does not exist a framework that adeptly integrates relevant methodologies to obtain the above information for large and complex structures, which need to be covered by multiple AIRs.

The contributions of the framework proposed in this work are threefold:

- To the best of authors' knowledge, it is the only framework that can conduct coverage analysis for multiple AIRs performing coverage tasks on large and complex structures.
- This framework can be used for both performance evaluation of AIRs and configuration design of a multi-AIR system for coverage tasks.
- This framework is scalable and consist of critical modules. It has the flexibility of enabling various methodologies and algorithms to be implemented in these modules, according to specific target structures and requirements of different applications.

The paper is organized as follows: Section II presents the problem formulation. Then, a framework for coverage analysis is presented in Section III, which includes critical modules and associated methodologies. In Section IV, the framework is tested and validated by a real-world application, where 10 different scenarios with up-to 10 AIRs are considered for thorough analyses. Finally, conclusions are given in Section V.

II. Problem Formulation

For multiple AIRs performing coverage tasks, the problem of coverage analysis consists of several sub-problems as follows:

- How to represent a structure such that the structure's representations (e.g., surfaces or points) can facilitate efficient coverage analysis?
- Where to place the AIRs so as to achieve the maximum coverage rate of the structure, considering that the AIRs may need to be repositioned multiple times due to the large size of the structure?
- Given a location of an AIR, how to check whether the structure's surfaces can be reached by the AIR's kinematics?
- For the structure's surfaces that can be reached by an AIR, how to check whether these surfaces can be reached without any collisions (e.g., the collisions between the AIR body and the structure)?
- Considering safe and efficient collaboration among AIRs, how to partition the surfaces of the structure and appropriately allocate the partitioned surfaces to all AIRs?
- How to compute the operation efficiency of the AIRs for the coverage tasks?

For large and complex structures, some of the above sub-problems become more challenging due to the following reasons:

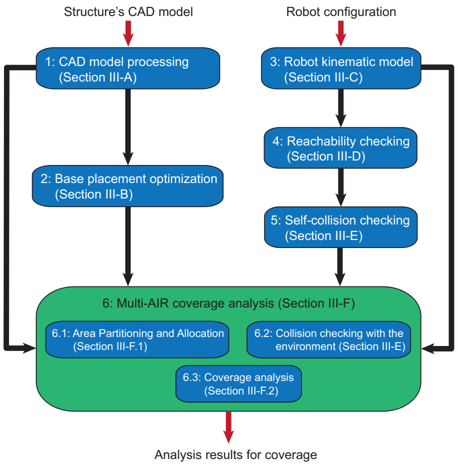
Structure's CAD model
Robot configuration
1: CAD model processing (Section III-A)
3: Robot kinematic model (Section III-C)
4: Reachability checking (Section III-D)
2: Base placement optimization (Section III-B)
5: Self-collision checking (Section III-E)
6: Multi-AIR coverage analysis (Section III-F)
6.1: Area Partitioning and Allocation (Section III-F.1)
6.2: Collision checking with the environment (Section III-E)
6.3: Coverage analysis (Section III-F.2)
Analysis results for coverage

Fig. 2. The framework for coverage analysis

- With the increase of structure's overall dimensions, more base placements are needed for AIRs to visit during coverage operation.
- The structure's large overall dimensions and geometric complexities (e.g., existence of beams, columns and uneven surfaces) may lead to high computational cost in the reachability checking and collision checking.
- The partitioning and allocation of the structure's surfaces to AIRs becomes more complex with the increase of the structure's overall dimensions and complexities.

Considering the sub-problems and challenges mentioned above, it is desired to have a framework which can comprehensively solve the problem of coverage analysis for large and complex structures. The framework is expected to generate the analysis results such as achievable coverage rate and operation efficiency, which are essential for the configuration design and performance evaluation of a multi-AIR system.

III. A Framework for Coverage Analysis

The proposed framework is illustrated by a flowchart in Fig. 2. The framework consists of different modules, including CAD model processing, base placement optimization, robot kinematic model, reachability checking, self-collision checking and multi-AIR coverage analysis. The multi-AIR coverage analysis module (i.e., the module within the green block shown in Fig. 2) includes Area Partitioning and Allocation (APA), collision checking with the environment and coverage analysis. The framework outputs the analysis results (i.e., coverage rate and operation efficiency) for the structure to be processed by multiple AIRs. More details of the modules shown in Fig. 2 are presented in the following sub-sections.

A. CAD model processing

In the design, manufacturing or construction of architectural and engineering structures, 3D CAD models are normally used. 3D CAD models normally represent a physical structure by various geometric entities (e.g., lines, triangles

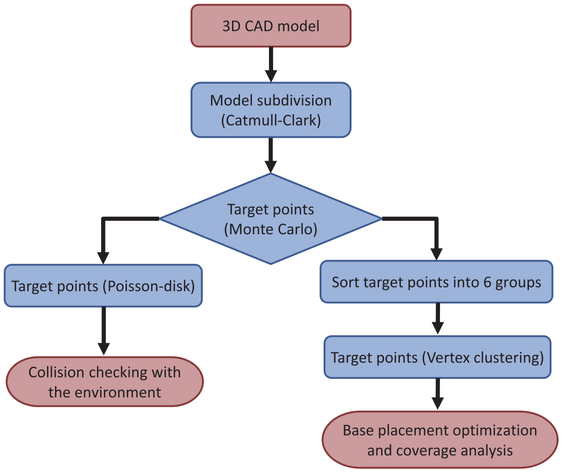
3D CAD model
Model subdivision (Catmull-Clark)
Target points (Monte Carlo)
Target points (Poisson-disk)
Sort target points into 6 groups
Target points (Vertex clustering)
Collision checking with the environment
Base placement optimization and coverage analysis

Fig. 3. Flowchart for model processing

and curved surfaces). However, representing the structure by the collection of various geometric entities may not be convenient and efficient in the coverage analysis of the structures, as different entities may need to be separately analyzed.

In the field of robotics, point clouds and meshes are widely used in representing the environments and objects that robots interact with. Only including one form of geometric entity (i.e., point), point clouds or meshes can be uniformly processed, which simplifies problems such as robot reachability checking and collision checking. Thus, in this work, CAD model processing (i.e., module 1 in Fig. 2) suggests converting the structure's 3D CAD model to a collection of target points for coverage analysis of a multi-AIR system interacting with a structure. For the sake of simplicity, in the rest of this paper, target points are refer to the points generated in the CAD model processing to represent the structure.

There exist many techniques and algorithms for converting 3D CAD models (e.g., STL model) to target points, including Monte Carlo sampling technique [4], poisson-disk sampling method [5] [6], vertex clustering method [7] and etc. Depending on specific requirements of different applications, these methods can be applied or integrated to generate target points. In this work, as shown in Fig. 2, the target points generated can be used in the modules of base placement optimization, collision checking with the environment and coverage analysis. To generate satisfactory target points for these different modules, in this work, a model processing approach is proposed and presented by a flowchart in Fig. 3.

For collision checking with the environment (i.e., the collision checking between the structure and the AIR body) presented in Section III-E, target points satisfying the following criteria are desired:

- The target points should have uniform density throughout all structure's surfaces, which can facilitate the accurate and efficient collision checking with the environment.
- Appropriate density of target points is desired, which can ensure collision-free operations of AIRs while not requiring high computational cost.

However, a structure may consist of many surfaces with different dimensions, which make it difficult to generate target points with uniform density throughout the whole structure. To solve this problem, as illustrated in Fig. 3, the 3D CAD model (e.g., STL model) of a structure is firstly subdivided using Catmull-Clark algorithm [8]. Subdividing each surface into small segments can help to represent the whole structure by segments with similar size, thereby facilitating to generate the target points with uniform density within all segments. Then, using Monte Carlo method [4], an initial group of target points with high density can be generated. A higher density of target points can achieve a higher accuracy in collision checking with the environment. However, it may lead to higher computational cost, as more target points need to be checked. To reduce the density, as shown in Fig. 3, the Poisson-disk sampling algorithm [5] is further implemented to regenerate the target points that are feasible for collision checking with the environment.

On the other hand, for base placement optimization and coverage analysis, more criteria may need to be considered in the generation of target points. To perform efficient and accurate coverage analysis, three key criteria are suggested:

- The target points should include normals, which indicate the orientations of a structure's surfaces (represented by target points) to be covered by AIRs.
- The target points should be uniformly distributed, i.e., the distance between any two adjacent points in horizontal or vertical direction should be identical. Given the AIR's operation speed, the uniformly distributed target points make it convenient to estimate the completion time of covering each individual surface, thereby facilitating the efficiency analysis for AIR's operation.
- The target points should be generated and distributed on all surfaces of a structure.

The target points generated using Monte Carlo method include the normals, which meet the first criterion listed above. To convert these dense target points to uniform target points satisfying the second criterion, one option is to apply vertex clustering approach [7]. Given the target points with dense distribution throughout the structure, vertex clustering utilizes cube-shaped 3D cells to contain the target points, thereby simplifying the representations of the structure by the centre points of 3D cells. Since the 3D cells are equally-sized and uniformly aligned in arrangement, the regenerated target points (i.e., centres of 3D cells) meet the second criterion listed above. Note that the horizontal or vertical distance between any two adjacent target points can be specified by the side length of the 3D cells.

In the generation of target points, whether the third criterion can be met depends on the size of each 3D cells as well as the minimum wall thickness of a structure. To be specific, if the side length of the 3D cells is larger than

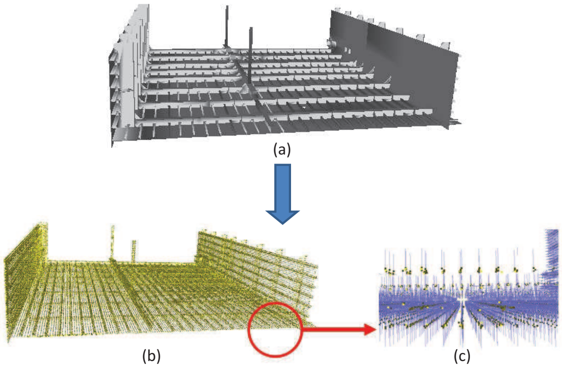

Fig. 4. Conversion from CAD model to uniform target points

the thickness of a wall, then the target points (i.e., initial target points generated using Monte Carlo method) on both surfaces of the wall will be contained by a single layer of 3D cells using vertex clustering method. This will result in a single layer of uniform target points representing the wall, which violates the third criterion. To solve this problem, as shown in Fig. 3, the initial target points generated by Monte Carlo method are firstly sorted into 6 groups, according to the orientations of points' normals in cartesian space. Then, these 6 groups of initial target points are separately processed by vertex clustering to form 6 groups of uniform target points, followed by the combination of these groups to obtain the uniform target points representing the whole structure. An example of CAD model processing is shown in Fig. 4. Given the 3D CAD model of a large and complex structure shown in Fig. 4 (a), the uniform target points meeting the above three criteria are generated and shown in Fig. 4 (b), with two layers of target points on each thin wall as illustrated by Fig. 4 (c).

B. *Base placement optimization*

Given a structure (e.g., the structure shown in Fig. 4) to be covered by multiple AIRs, another crucial problem is to find proper AIR base placements that can facilitate all AIRs to collaboratively cover the entire structure. A base placement is a fixed location for the base of an AIR, where the AIR can operate on the allocated area of the structure's surface. Due to the large size of the target structure (e.g., a structure with side length larger than 10 m), many base placements need to be selected for each AIR to visit during the operation, i.e., each AIR needs to be locomoted many times to complete its coverage task. The problem of AIR base placements needs to be addressed while considering the following criteria:

- Maximize the coverage rate of the entire structure.
- Minimize the overall completion time.
- Minimize the number of base placements for each AIR.
- Generate collision-free base placements for AIRs.
- Provide the visiting sequence of the base placements.

Considering the multiple objectives and constraints listed above, the base placement problem can be treated as a multi-objective optimization problem. Therefore, base placement optimization (i.e., module 2 in Fig. 2) is included to generate appropriate base placements, where the AIRs' bases can be placed to conduct coverage tasks.

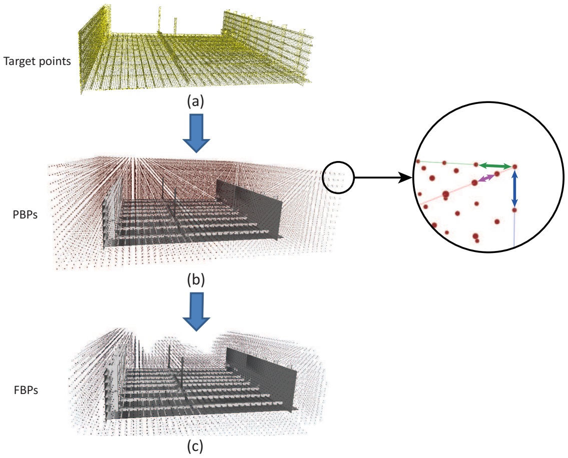

Fig. 5. Example of base placements generation

For the optimization of base placements, one efficient way is to firstly generate a collection of base placement candidates, from which the final base placements for AIRs to visit can be optimized. An example for the generation of base placement candidates is illustrated in Fig. 5. Given the uniform target points of a structure shown in Fig. 5 (a), a group of Possible Base Placements (PBPs) enclosing the structure can be first generated. As can be seen in Fig. 5 (b), all PBPs have uniform distribution throughout the whole workspace for AIRs (i.e., Cartesian space). However, PBPs include the base placements where AIRs may collide with the structure or can not reached to the surfaces of the structure. Thus, to further reduce the size of base placement candidates, additional criteria can be considered to sort out Favourite Base Placements (FBPs) from PBPs. FBPs refer to the base placement candidates where the AIRs are more likely to achieve the stated objectives and constraints. As shown in Fig. 5 (c), a collection of FBPs can be sorted out from PBPs, with the consideration of criteria as follows:

- Base placements with an above-threshold distance relative to the surfaces of the structure, which are less likely to cause collision between AIRs and the structure.
- Base placements where an AIR's workspace can cover above a certain number of uniform target points generated (see module 1 in Section III-A), so as to discard base placements with low potential coverage. Note that the number of target points inside the AIR's workspace (with appropriate normal) can indicate the potential coverage rate of the structure's surfaces.

After obtaining the FBPs, optimization can be conducted to select the final base placements for AIRs to visit during coverage task. There are some research works available for the optimization of robot's base placements in different applications [9] [10]. However, these works are only

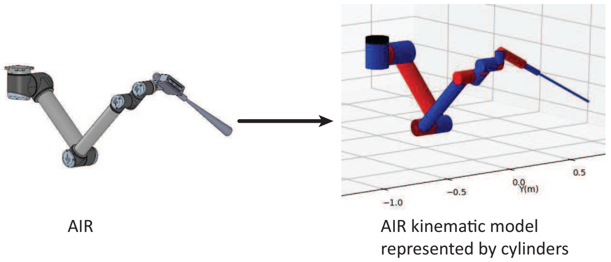

Fig. 6. AIR kinematic model represented by cylinders

applicable to a single robot. In the authors' previous works [11] [12] [1], optimization-based methods are proposed to determine the base placements for multiple AIRs. These methods are promising for target structures with small or medium dimensions (e.g., small components that can be covered without moving AIRs' bases, or a vehicle that can be covered by several base placements of two AIRs). With the increase of size for search space (e.g., the large-scale structures considered in this work), these methods may become computationally expensive, but can still be used by making some simplifications or modifications. Considering the stated objectives and constraints, some other methods such as genetic algorithm (GA) and greedy-based method can also be used to obtain the optimized base placements from FBPs.

C. Robot kinematic model

For the framework presented in this work, the robot kinematic model is essential for both reachability checking and collision checking. A robot kinematic model can be used to represent the robot's forward kinematics and derive the inverse kinematics.

The forward kinematics uses the robot kinematic equations to compute the pose of the robot's end-effector (EE) from specified values of the joint parameters [13]. Using Denavit-Hartenberg (DH) method [14], the forward kinematic model of an AIR can be built, which is represented by a series of cylinders shown in Fig. 6. Conversely, given a desired EE pose, the inverse kinematics can be used to compute whether there exists a feasible solution for the robot joints. There are two main kinds of methods for computing inverse kinematics, namely analytical methods [15] [16] and numeric methods [17] [18].

In the framework illustrated in Fig. 2, only forward kinematic model is included in the module 3. Depending on the methodologies applied in reachability checking (i.e., module 4 presented in Section III-D), module 3 can be used to:

- derive the inverse kinematics to check the AIRs' reachability to the uniform target points generated from module 1, or
- facilitate to build the lookup table that can be used for reachability checking.

In addition, the forward kinematic model in module 3 can also be used to conduct collision checking, including self-collision checking and collision checking with the environment (see Section III-E for more details).

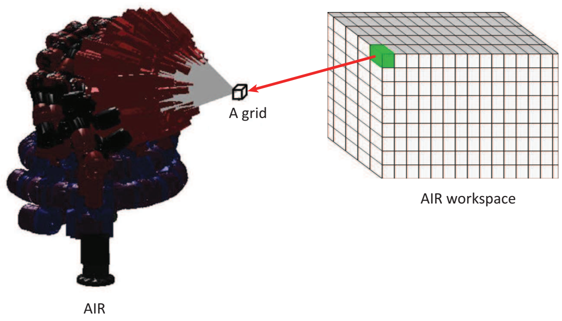

Fig. 7. A group of poses associated to a grid from AIR workspace [2]

D. Reachability checking

For base placement optimization (module 2) and coverage analysis (module 6.3), one task is to check the reachability to each target point representing the structure (module 1). Note that the AIRs' reachability can be affected by many factors, including the structure's geometric complexity, the placements of the AIRs' bases with regard to the structure, the AIR configuration, etc. Given the uniform target points representing the structure (module 1), optimized base placements (module 2) as well as the robot kinematic model (module 3), different reachability checking methods can be applied to figure out whether there exist feasible AIR poses to reach the target points with appropriate EE orientations and positions.

One option for checking AIR's reachability is to perform robot inverse kinematics. As presented in Section III-C, there exist two kinds of methods for computing inverse kinematics, including analytical methods and numeric methods. However, the former may be hard to obtain analytical solutions, whereas the latter usually comes with low computational efficiency [19]. In addition, after performing robot inverse kinematics, the self-collision checking (i.e., the collision checking between all links of an AIR) needs to be conducted to verify whether the AIR pose solution is collision-free. This will further reduce the efficiency in reachability checking. Note that collision checking is described in the next sub-section, including both self-collision checking and collision checking with the environment.

Another option is to construct the lookup table using AIR forward kinematics (module 3), which stores feasible robot poses for a set of discretized EE points within the robot workspace [20] [2]. Since coverage of large structures is considered in this paper, then a very large number of target points is needed to represent the structure (e.g., about 150,000 uniform target points are generated for the structure shown in Fig. 4). Thus, checking reachability to each target point can be computationally costly. As a lookup table is constructed off-line (only once) and queried on-line using Kd-trees, Quadtrees, Octrees or similar hierarchical data structures [21], it is computationally efficient and effective

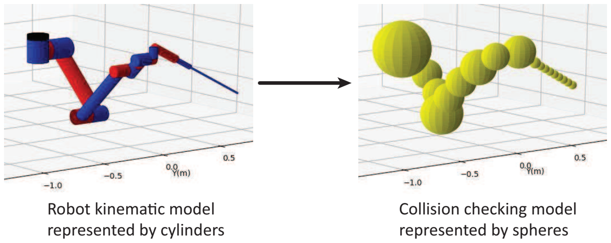
−1.0
−0.5
0.0
0.5
Y(m)
−1.0
−0.5
0.0
0.5
Y(m)
Robot kinematic model represented by cylinders
Collision checking model represented by spheres

Fig. 8. Collision checking model represented by spheres

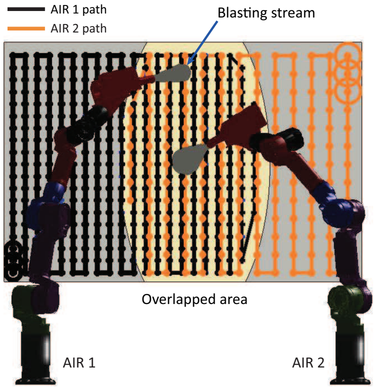

Fig. 9. Example of area Partitioning between two AIRs [33]

for real-time applications. However, the solution accuracy depends on the size of the lookup table (i.e., the amount of information stored), and a larger lookup table will need a larger memory space to store the data. Note that, the self-collision checking can be considered as part of the construction of lookup table, thereby saving computational cost during coverage analyses.

In the construction of lookup table, as shown in Fig. 7, the workspace of an AIR is decomposed into a large number of cube-shaped and equally-sized 3D grids. Each 3D grid is associated with many discrete robot poses that can reach the grid with different orientations. This is done by incrementally sweeping through all joints of the AIR and assigning each pose to the appropriate grid. These poses (within each grid) are further grouped according to the similarity in the orientation of the EE. Thus, if a target point representing the structure falls within one of the grids in the lookup table, then the AIR poses associated with the grid (and the relevant group within the grid) are obtained from the lookup table. Note that all robot poses stored in the lookup table are self-collision free, as self-collision checking has been considered while constructing the one-off lookup table.

E. Collision checking

In the reachability checking (module 4 in Fig. 2) and coverage analysis (module 6.3), the collision checking is essential for safe operation of AIRs. The robot collision checking normally includes self-collision checking (i.e., the collision checking between links of each AIR) and collision checking with the environment (i.e., the collision checking between individual AIRs, and objects in the environment). Using the lookup table (module 4 in Fig. 2), a set of AIR poses for each target point representing the structure is obtained. These poses do not self-collide and are within the joint limits. Then, these poses are checked one-by-one using module 6.2 to find one that is collision-free with the environment.

There are many methods and techniques for robot collision checking or detection in the literature. One effective way of achieving collision avoidance is to utilize hardware, including impedance actuators [22] [23], force and torque sensors [24] [25], depth sensor and camera [26] [27]. There are two main classes of algorithms for collision detection, namely Discrete Collision Detection (DCD) and Continuous Collision Detection (CCD) [28]. DCD algorithms normally have high detection speed and wide applications in real-time operations [29] [28], while CCD algorithms can overcome the potential undetected problem due to discrete robot configurations considered in DCD methods, but usually lead to high computational cost [30] [31]. The above methods are applicable for the scenarios where the operation environment is unknown or partially unknown (e.g., the target structure and environment need to be localized).

As for the scenarios with known environment (e.g., the location of target structure is given), one effective and fast way for collision checking is to simplify the AIR and represent it by spheres and capsules [32], as shown in Fig. 8. Using robot kinematic model (module 3), the configurations of spheres and capsules surrounding the robot body can be computed. Then, the self-collision checking can be simplified by computing whether the spheres and/or capsules overlap with each other. On the other hand, the collision checking with the environment can be performed by computing whether a target point representing part of a structure locates within a sphere or a capsules representing the AIR. All target points that are within a certain proximity to a sphere representing part of the AIR are checked, and data structures such as Kd-tree can be used for fast queries.

F. Multi-AIR coverage analysis

As illustrated in Fig. 2, multi-AIR coverage analysis (module 6 with green background) includes three sub-modules, namely Area Partitioning and Allocation (APA), collision checking with the environment (described in Section III-E) and coverage analysis. The APA aims to partition all surfaces of a target structure and allocate them to each individual AIR, whereas the coverage analysis is included to obtain the results such as overall coverage rate and operation efficiency. These two sub-modules are further discussed below.

1) Area Partitioning and Allocation: For coverage tasks to be conducted by multiple AIRs, another essential problem is how to partition the surface areas of the target structure and appropriately allocate the partitioned areas amongst all AIRs. To better understand this problem, a simple example is presented in Fig. 9, where two AIRs conduct grit-blasting operation on a wall. As highlighted in Fig. 9, without partitioning and allocation of the overlapped area (the area that can be reached by both AIRs), both AIR 1 and AIR 2 will cover the overlapped area, which may lead to low operation efficiency or even collision between these two AIRs. Given the optimized base placements generated by module 2 in Fig. 2, the base of each AIR needs to be moved and placed many times to complete the whole coverage task. This can not be efficiently and safely achieved without proper Area Partitioning and Allocation (APA). Therefore, the APA module is included in the framework shown in Fig. 2, with objectives or constraints listed as follows:

- Avoid missing surface areas of the target structure for AIRs to cover.
- Minimize the overall completion time of the coverage task through equitable APA.

It is challenging to simultaneously achieve the objectives listed above, especially when the object's surfaces are non-planar, complex in shape and unconnected from each other [34]. There exists some methods for area decomposition, partitioning and allocation, including convex decomposition methods [35], grid graph bisection method [36], and gradient based optimizations [37]. However, these methods only focus on solving part of the APA problem considered in this work. In one of the authors' previous works, Voronoi partitioning [38] and multi-objective optimization are combined to solve the APA problem of a multi-AIR blasting system [39]. However, with the increase in the size and geometric complexity of the target structure, simplifications or modifications may be needed to reduce the computational cost of this optimization-based method.

To perform coverage rate analysis of large and complex structures, one efficient solution of APA problem is to use First-Come First-Served (FCFS) method with a greedy base placement approach. Using a greedy-based placement approach (module 2 in Fig. 2), the base placement (i.e., one of FBPs) from which the largest surface coverage (i.e., maximum number of target points locating within the AIR's workspace) is obtained and allocated to an AIR. Then, the next base placement with the largest surface coverage is assigned to the next AIR, and so on. Every time, after finishing coverage at a base placement, the AIR will move to the next base placement with maximum potential coverage. At each base placement, the AIR will take all the points that it can cover (according to FCFS) even if some points may be covered at a later stage by another AIR at a different base placement. In this way, all surface areas of the structure can be partitioned and allocated without overlaps amongst the AIRs. Although this method may not provide an optimal solution, it is computationally efficient for performing coverage analysis for large structures.

2) Coverage analysis: After partitioning and allocating the surfaces (represented by target points generated by module 1 in Fig. 2) of a structure to the AIRs, the coverage analysis needs to be conducted, while performing the collision checking with the environment (see collision checking in Section III-E). In this work, as stated in Section I, the scope of coverage analysis includes two aspects, including coverage rate and operation efficiency. For the analysis of coverage rate, some basic criteria are listed as follows:

- Every surface of the structure should be analyzed to obtain the overall coverage rate.
- Coverage of the structure's surfaces should be checked by reachability checking (module 4).
- All reachable surfaces of the structure should be checked by self-collision checking (module 5) and collision checking with the environment (module 6.2).

Depending on the requirements of the application under consideration, some additional criteria may need to be considered, such as robot joint torque minimization and manipulability measure maximization [39].

To analyze the coverage of structure's surface areas allocated to an individual AIR, one efficient option is to only check the AIR's reachability to selective target points representing the surface areas. For instance, given a rectangular surface area, it is assumed that the AIR can cover the whole area only if the 4 target points at the four corners of the area can be reached by the AIR without any collisions. This option can greatly reduce the number of target points to be analyzed, thereby improving the efficiency in the analysis of coverage rate. However, if some other unchecked target points within the area cannot be reached by the AIR, then the AIR may not be able to cover the whole surface area. Therefore, this option may not provide accurate analysis results for coverage rate. To improve the accuracy of coverage analysis, another option is to check the AIR's reachability to all target points representing the surface areas. However, for large and complex structures (e.g., the structure represented by about 150,000 target points in Fig. 4 (b)), it may be computationally expensive to check AIR's reachability to every target point, while considering the criteria for coverage analysis listed above.

To conduct accurate and efficient coverage analysis, appropriate strategies and methods can be selected for the modules and sub-modules of the framework shown in Fig. 2. To ensure the analysis accuracy, all target points representing the structure need to be checked by module 4 (i.e., reachability checking), module 5 and module 6.2 (i.e., collision checking). Thanks to the CAD model processing, which can generate uniform target points throughout the whole structure with proper density. On the other hand, to improve the analysis efficiency, lookup table can be constructed for efficient reachability checking (module 4). Note that, the self-collision checking (module 5) can also be included in the one-off construction of lookup table, which can further enhance the computational efficiency of the whole framework. In addition, the fast optimization method such as greedy-based method can be applied to obtain the optimized base placements for all AIRs to visit (module 2), whereas the FCFS method can be implemented to achieve quick APA (module 6.1) for the collaboration of AIRs in coverage tasks. Finally, the target points that can be reached

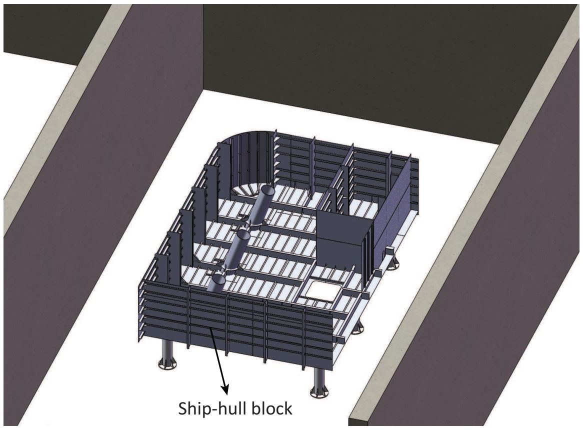

Fig. 10. A ship-hull block to be cleaned by a multi-AIR blasting system

without collisions are sorted out and allocated to all AIRs, thereby obtaining the coverage rate of the whole structure. In terms of operation efficiency, given the average operation speed at the AIR's EE, it is convenient to estimate the AIRs' operation time of achieving the coverage rate, thanks to uniform distribution of the target points.

IV. Analysis and Results

A. Real-world example application

In this work, a multi-AIR blasting system for cleaning ship-hull blocks (i.e., large-scale sections that compose an entire ship hull in shipbuilding industry) is considered as an example application. The operation environment of the multi-AIR blasting system is illustrated in Fig. 10, where a ship-hull block is placed within the blasting room for cleaning process. The system configuration and related assumptions are listed as follows:

- UR10 robot is considered as the platform of the blasting AIR as shown in Fig. 6.
- The underneath surfaces of the block are assumed to be cleaned by the AIRs locomoted by Autonomous Ground Vehicles (AGVs).
- The other surfaces of the block are assumed to be cleaned by the AIRs locomoted by some other facilities, e.g., gantries, cherry pickers and scissor lifts.
- Any base placement (see Section III-B) within the blasting room can be visited by the AIRs through above locomotion measures (e.g., AGVs and gantries).

The multi-AIR blasting system can replace human workers in the blasting room, where the dense dust, the loud noise, and human fatigue are hazardous for workers' health. However, the limited allowed duration for the cleaning process of the ship-hull blocks poses high requirements regarding coverage rate and cleaning efficiency that need to be met by the blasting system. Thus, before the deployment of such a system, it is essential to conduct coverage analysis to estimate the minimum number of AIRs that can satisfy the operation requirements. In addition, for each individual ship-hull block, the coverage analysis can also provide the maximum achievable coverage rate as well as the surfaces that can not be covered by the AIRs (these surfaces are to be cleaned through manual blasting).

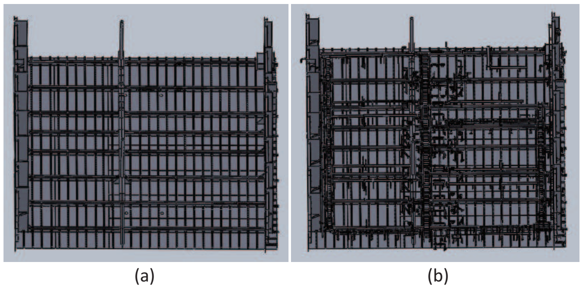

Fig. 11. CAD models of Block #1

As shown in Fig. 1, the ship-hull blocks have the following properties:

- Large overall dimensions (e.g., the length and width are within [10 15]m).
- Small local features (e.g., beams and columns with small lengths, widths, and/or heights).
- Thin walls (e.g., the wall thickness is within [12 20]mm).
- Existence of outfittings (e.g., pipes and components installed on the block).

In this work, for each ship-hull block, two different scenarios (i.e., block model without outfittings and block model with outfittings) are considered in coverage analysis. Fig. 11 presents the CAD models of Block #1, where the model without outfittings and the model with outfittings are shown in Fig. 11 (a) and Fig. 11 (b), respectively.

To test and verify the framework proposed in this work, extensive analyses are conducted for 10 different scenarios (i.e., 5 ship-hull blocks without and with outfittings, including Block #1 shown in Fig. 11). For each scenario, the analysis results include:

- Maximum achievable coverage rate.
- Coverage rate vs. operation time.
- Operation time vs. number of AIRs in completing maximum achievable coverage rate.

Note that, in this work, maximum achievable coverage rate is not only determined by the structure complexities of scenarios (i.e., ship-hull blocks) and the methodologies used in the framework (e.g., methodologies for base placement optimization and APA), but also limited by the configuration constraints of AIRs and end-effectors (EEs).

B. Setup of Framework Modules

This section presents the methods and algorithms that are used in the modules in Fig. 2 to generate the analysis results for the example application presented in Section IV-A.

Given the CAD models of ship-hull blocks, two different groups of target points can be generated using the approach illustrated in Fig. 3. As presented in Section III-A, one group of target points can be used for collision checking with the environment (i.e., module 6.2), whereas the other group of

TABLE I

MAIN PROPERTIES OF THE LOOKUP TABLE.

Parameter	Value
Number of 3D grids for AIR workspace	720192
Number of all AIR poses for 3D grids	83886080
Number of AIR pose groups for each grid	67

uniform target points can be used for both base placement optimization (i.e., module 2) and coverage analysis (i.e., module 6.3). As described in Section III-F.2 (i.e., module 6.3), to conduct accurate and efficient coverage analysis, the uniform target points with proper density are needed. In this work, the density of uniform target points is represented by the interval distance between any two adjacent target points. For the application illustrated in Fig. 10, it depends on the diameter of blasting area on the block surface, which is projected by the blasting stream from AIR's EE. To be specific, as shown in Fig. 9, if the interval distance of two adjacent target points is set to be the diameter of circular area projected by the blasting stream, then the surface area between these two target points can be assumed to be covered by the AIR. In this work, the normal distance between the blasting nozzle at AIR's EE and the block's surface is set as 500 mm, which results to the blasting circular area with the diameter of 200 mm on the block's surface. Therefore, by setting the interval distance of the two adjacent target points as 200 mm, a group of uniform target points (including about 150,000 points) is generated and shown in Fig. 4 (b).

In base placement optimization (i.e., module 2 presented in Section III-B), two criteria are given for the generation of FBPs. In this example application, the AIRs with working radius of 2.3 m are considered, and the settings of these two criteria are listed as follows:

- Threshold distance between base placements and the block's surfaces: 0.7 m.
- Threshold number of target points locating within the AIR workspace: 1.

Using the criteria listed above, the FBPs for each ship-hull block can be generated (e.g., the FBPs of Block #1 shown in Fig. 5 (c)). Note that a base placement with a larger number of target points locating within the AIR's work space only indicates that the AIR is likely to cover more surface areas of the block at this particular base placement. The reachability of these target points will be checked by reachability checking (i.e., module 4) and the collision checking with the environment (i.e., module 6.2). After obtaining the FBPs, the final base placements (i.e., the base placements for the AIRs to visit during blasting operation) need to be optimized and selected from FBPs. In base placement optimization, the APA (see Section III-F.1) needs to be considered to divide and allocate the overlapped surface areas amongst AIRs as shown in Fig. 9. In this work, to perform efficient coverage analysis, the greedy-based method and the FCFS method are combined to simultaneously solve the problems of base placement optimization and APA, which is presented in Section III-F.1.

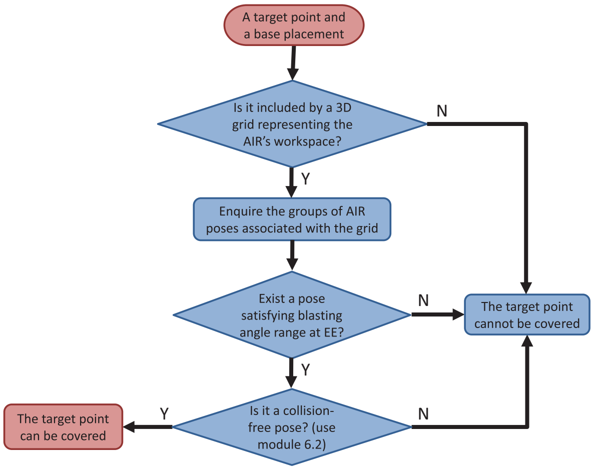
A target point and a base placement
Is it included by a 3D grid representing the AIR's workspace?
N
Y
Enquire the groups of AIR poses associated with the grid
Exist a pose satisfying blasting angle range at EE?
N
The target point cannot be covered
Y
The target point can be covered
Y
Is it a collision-free pose? (use module 6.2)
N

Fig. 12. Flow chart for querying the lookup table

In the example application presented in Section IV-A, UR10 robots are considered as the platforms of AIRs performing blasting operation on ship-hull blocks. Using DH method [14] and the configuration of UR10 robot, the robot kinematic model is built and represented by cylinders shown in Fig. 6. Based on the kinematic model, as shown in Fig. 8, the collision checking model is constructed and represented by spheres.

In terms of reachability checking (i.e., module 4 presented in Section III-D), considering the scales and complexities of ship-hull blocks presented in Section IV-A, a lookup table is constructed to achieve efficient coverage analyses. In this work, k-d tree [40] data structure is used in the construction of the lookup table. As can be seen from the lookup table properties listed in Table I, a huge number of AIR poses (i.e., 83886080 poses) are included to enhance the analysis accuracy. For each individual grid representing AIR's workspace, 67 groups of AIR poses are considered where each group contains EE poses with similar orientation. The flow chart in Fig. 12 illustrates the process of querying the lookup table. More details about lookup table construction are available in the authors' previous work in [2].

As presented in Section III-F.1, multiple objectives need to be optimized when solving an APA problem, which can be viewed as a multi-objective optimization problem. In one of the authors' previous works, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) is used to solve the APA problem for multiple AIRs [39]. However, in this work, the ship-hull blocks are characterized by large-scale dimensions (i.e., the side length is up to 10 to 15 m) and structure complexities (e.g., the existence of small beams, columns and outfittings), which make APA a challenging problem. For the sake of computational efficiency, as presented in

TABLE II

Number of target points for 5 blocks without outfittings.

Block part number	Number of target points
#1	152675
#2	149444
#3	172211
#4	101654
#5	151425

Section III-F.1, the FCFS method is combined with greedy-based method to simultaneously solve the problems of APA and base placement optimization.

In this framework, as presented in Section III-F.2, both coverage rate and operation efficiency need to be analyzed within coverage analysis (i.e., module 6.3). The coverage rate indicates the maximum percentage of the whole block area that can be covered by AIRs. Module 6.1 (i.e., APA) partitions and allocates the block's surface areas (represented by uniform target points) to all AIRs. However, some allocated target points may not be reached by an AIR, due to the AIR configuration constraints, structure complexities (e.g., small I-beams and L-beams), and existence of outfittings (e.g., pipes and small components installed on the block). To obtain accurate coverage rate, in module 6.3, the reachability of target points at the optimized base placements is checked using the lookup table. Meanwhile, module 6.2 (i.e., collision checking with the environment) is used to ensure that all reachable points are covered by the AIRs with collision-free poses. Thanks to the uniform target points generated, the coverage rate of a block's surfaces can simply be computed based on the number of reachable target points. To analyze the operation efficiency, two time-related assumptions are made as follows,

- At each base placement, the blasting speed (i.e., the speed for robot EE to move along reachable points) is considered constant: 0.07 m/s. This is a reasonable assumption for the application under consideration since the AIR's EE needs to move at a constant speed for uniform coverage.
- For each individual AIR, the movement time between any two base placements is considered constant: 15 s. However, the actual time based on a point-to-point planner can be used.

Given the reachable points at each base placement (i.e., obtained in the coverage rate analysis) as well as two assumptions listed above, the blasting operation time at each base placement and AIRs' movement time between differen base placements can be computed, thereby obtaining the operation efficiency of the whole multi-AIR blasting system.

C. Results

In this work, different real ship-hull blocks are analyzed using the framework setup presented in Section IV-B. For each block (e.g., Block #1 shown in Fig. 11), there are two different scenarios, including the block without outfittings (as shown in Fig. 11 (a)) and the block with outfittings (as shown in Fig. 11 (b)). This is to demonstrate how the existence of outfittings affect the coverage rate and operation efficiency to be achieved by a multi-AIR blasting system. Table II illustrates the number of uniform target points generated for 5 blocks without outfittings (i.e., blocks with part number #1, #2, #3, #4 and #5). Thus, using these 5 blocks with different part numbers, 10 different scenarios (i.e., 5 scenarios without outfittings and 5 scenarios with outfittings) are considered for thorough analyses.

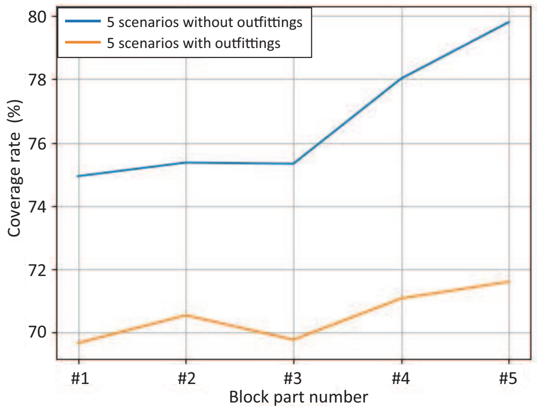

Fig. 13. Coverage rates of 10 scenarios

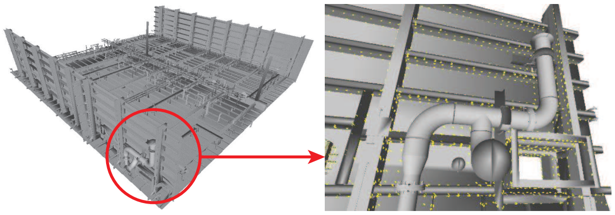

Fig. 14. Uncovered points on Block #1 with outfittings

The simulation results for coverage rates of 10 scenarios are presented in Fig. 13. Depending on the structure complexities of different block, 75% to 80% of each block can be covered for the scenario without outfittings, whereas about 69% to 72% of each block can be covered for the scenario with outfittings. It is interesting to note that the coverage rates are obtained under the constrains from both equipments (i.e., configuration constraints of AIRs and end-effectors) and the target structures (e.g., I-beams, L-beams and pipes on the blocks). To further increase the coverage rates, the AIRs and end-effectors (EEs) with higher maneuverabilities and flexibilities can be used to cover the hard to reach areas, which is not the interest of this work. To indicate the accuracy of the coverage rates, the uncovered areas highlighted by yellow points are shown in Fig. 14, which cannot be reached due to the kinematic limitations of the AIR shown in Fig. 6.

After obtaining the coverage rates of all scenarios, it is necessary to calculate the system's operation efficiency

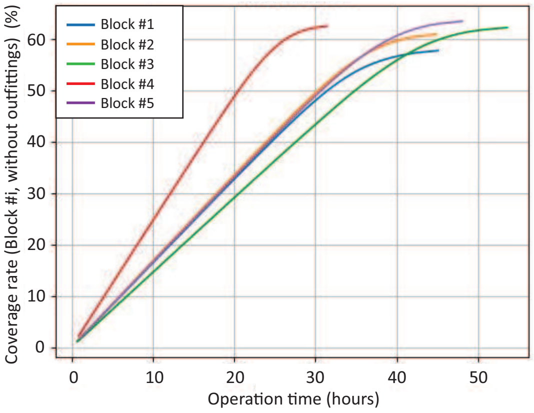

Fig. 15. Coverage rate vs. operation time (5 scenarios without outfittings)

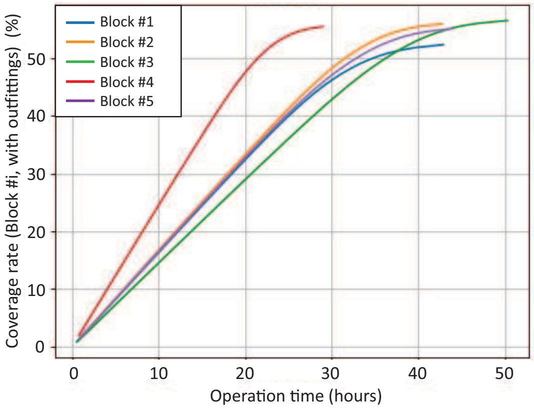

Fig. 16. Coverage rate vs. operation time (5 scenarios with outfittings)

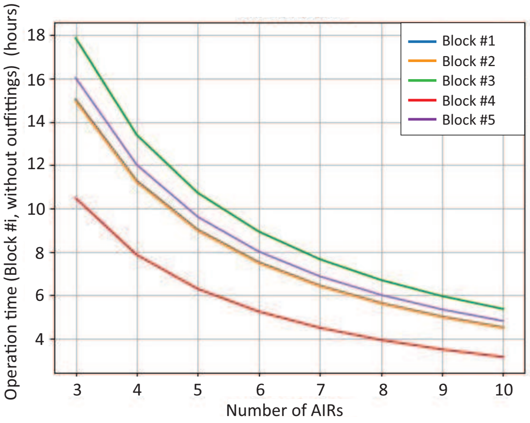

Fig. 17. Operation time vs. number of AIRs (5 scenarios without outfittings)

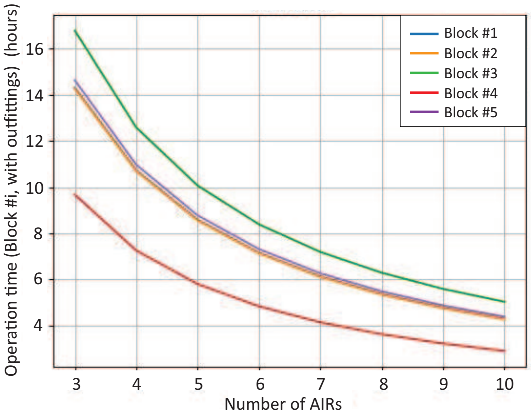

Fig. 18. Operation time vs. number of AIRs (5 scenarios with outfittings)

in completing all reachable areas of each scenario. As presented in Section IV-A, the AIRs on AGVs are assumed to blast the underneath flat surfaces of the block, whereas the other AIRs locomoted by some other facilities (e.g., gantries, cherry pickers and scissor lifts) are in charge of blasting the rest surfaces with complex structures. Thus, compared with AIRs on AGVs, the other AIRs are expected to take longer completion time in the coverage task. Therefore, in this work, the underneath surfaces of the block are not considered in efficiency analysis, assuming that the operation efficiency of the whole multi-AIR blasting system depends on the efficiency of AIRs locomoted by other facilities than AGVs.

The operation efficiency of a single AIR is firstly analyzed, i.e., how much time it takes a single AIR to achieve the coverage rate of each scenario presented in Fig. 13. The analysis results for the 5 scenarios without outfittings and the other 5 scenarios with outfittings are presented in Fig. 15 and Fig. 16, respectively. Depending on the dimensions and complexities of the block, it will take a single AIR 28 to 54 hours to cover all reachable area. With the increase in operation time, the increase in coverage rate becomes slower, as the base placements with fewer target points (i.e., the target points within the AIR's workspace) are visited by the AIR towards to the end of the process. As can be seen from Fig. 15 and Fig. 16, for each block, the scenario without outfittings is characterized by higher coverage rate (shown in Y-axis) and longer operation time (shown in X-axis), as compared with the scenario with outfittings. This clearly illustrates how the existence of outfittings affect the performance of a multi-AIR blasting system in both coverage rate and operation efficiency. Note that the differences in coverage rates shown in Fig. 13, Fig. 15 and Fig. 16 indicate the coverage rates to be achieved by the AIRs on AGVs, which are not needed in the analysis of operation efficiency.

In the configuration design of a multi-AIR blasting system, it is essential to figure out how many AIRs are needed to satisfy the required operation efficiency in practice. Using the FCFS method presented in Section III-F.1, the analysis results representing the relationship between operation time and number of AIRs can be obtained for all 10 scenarios, which are presented in Fig. 17 (i.e., 5 scenarios without outfittings) and Fig. 18 (i.e., 5 scenarios with outfittings), respectively. As can be seen from the analysis results, for each scenario, the operation time drops with the increase in the number of AIRs. According to the specific time duration for blasting operation in practice, proper number of AIRs can be selected in the multi-AIR blasting system.

V. Conclusions

This work proposed a framework for the coverage analysis of large and complex structures to be processed by multiple AIRs. This framework consists of different modules where relevant methodologies and algorithms are presented for each individual module, such that coverage analysis can be performed efficiently for large and complex structures. A real-world application of a multi-AIR blasting system for ship-hull blocks is considered. Using the ship-hull blocks with different scenarios (i.e., without outfittings and with outfittings), extensive analyses are conducted to generate the results such as coverage rate and operation efficiency. The results for coverage rate not only demonstrate the AIRs' capacity in covering different blocks, but also provide the block surfaces to be processed by human workers (i.e., the surfaces that can not be covered by AIRs). On the other hand, the results for operation efficiency are essential for the configuration design of a multi-AIR blasting system (e.g., number of AIRs for satisfying the required production efficiency in practice).

VI. Acknowledgments

This work was supported by the Project ”Development of Grit-blasting Robots for Commercial Cargo Ship-hull Cleaning”, which was sponsored by China Merchants Heavy Industry (Jiangsu) Co., Ltd (CMHI). A special thanks goes to all team members from CMHI who helped with various aspects of this work.

References

[1] M. Hassan, D. Liu, and G. Paul, “Collaboration of multiple autonomous industrial robots through optimal base placements,” *In Journal of Intelligent and Robotic Systems*, vol. 90, pp. 113–132, 2018.

[2] M. Hassan, “Enabling methodologies for optimal coverage by multiple autonomous industrial robots,” *Doctoral dissertation*, 2018.

[3] Naval technology. [Online]. Available: https://www.naval-technology.com/projects/gowind-2500-corvette/attachment/gowind-2500-corvette3/

[4] M. Corsini, P. Cignoni, R. Scopigno, and M. Corsini, “Efficient and flexible sampling with blue noise properties of triangular meshes,” *IEEE transactions on visualization and computer graphics*, vol. 18, no. 6, pp. 914–924, 2012.

[5] R. Bridson, “Fast poisson disk sampling in arbitrary dimensions,” *ACM SIGGRAPH*, 2007.

[6] J. Bowers, R. Wang, L. Wei, and D. Maletz, “Parallel poisson disk sampling with spectrum analysis on surfaces,” *ACM Transactions on Graphics*, vol. 29, pp. 166:1–10, 2010.

[7] K. L. Low and T. S. Tan, “Model simplification using vertex-clustering,” in *Proceedings of the Symposium on Interactive 3D Graphics*, 1997, pp. 75–81.

[8] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350–355, 1978.

[9] M. F. Aly, A. T. Abbas, and S. M. Megahed, “Robot workspace estimation and base placement optimisation techniques for the conversion of conventional work cells into autonomous flexible manufacturing systems,” *International Journal of Computer Integrated Manufacturing*, vol. 23, no. 12, pp. 1133–1148, 2010.

[10] J. Yang, W. Yu, J. Kim, and K. Abdel-Malek, “On the placement of open-loop robotic manipulators for reachability,” *Mechanism and Machine Theory*, vol. 44, no. 4, pp. 671–684, 2009.

[11] M. Hassan and D. Liu, “An approach to base placement and area partitioning for complete surface coverage by multiple autonomous industrial robots,” in *In 4th International Doctoral Symposium on Mechanical Engineering at Hokkaido University (IDSHU)*, 2015, pp. 89–94.

[12] M. Hassan, D. Liu, G. Paul, and S. Huang, “An approach to base placement for effective collaboration of multiple autonomous industrial robots,” in *In IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3286–3291.

[13] P. Richard, *Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators*. MIT Press, Cambridge, Massachusetts, 1981.

[14] D. Jacques and H. R. Scheunemann, “A kinematic notation for lower-pair mechanisms based on matrices,” *Trans ASME J. Appl. Mech.*, vol. 23, pp. 215–221, 1955.

[15] Y. J. Liu and T. Huang, “Inverse kinematics and trajectory planning of 6R serial manipulators,” *Mech. Eng.*, vol. 48, no. 3, pp. 9–15, 2012.

[16] W. Li, Y. Du, Z. Song, X. Zhao, and E. Mao, “An alternative inverse kinematics position analysis for the control of welding robot,” in *International Conference on Mechanical Design*, 2018, pp. 1323–1341.

[17] C. Hua, C. Weishan, and X. Tao, “Wavelet network solution for the inverse kinematics problem in robotic manipulator,” *J. Zhejiang Univ. Sci. A.*, vol. 7, no. 4, pp. 525–529, 2006.

[18] Z. Ren, Z. Wang, and L. Sun, “A hybrid biogeography-based optimization method for the inverse kinematics problem of an 8-dof redundant humanoid manipulator,” *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 7, pp. 607–616, 2015.

[19] Y. Wei, S. Jian, S. He, and Z. Wang, “General approach for inverse kinematics of nr robots,” *Mechanism and Machine Theory*, vol. 75, pp. 97–106, 2014.

[20] H. Halfar, “General purpose inverse kinematics using lookup-tables,” in *IEEE International Conference on Industrial Technology (ICIT)*, 2013, pp. 69–75.

[21] S. Peters, “Quadtree- and octree-based approach for point data selection in 2d or 3d,” *Annals of GIS*, vol. 19, no. 1, pp. 37–44, 2013.

[22] R. Wang and H. Huang, “An active-passive variable stiffness elastic actuator for safety robot systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3664–3669.

[23] Y. Ono, K. Shimamoto, T. Nogawa, H. Masuta, and H. Lim, “Passive collision force suppression mechanism for robot manipulator,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 280–285.

[24] D. Popov, A. Klimchik, and N. Mavridis, “Collision detection, localization & classification for industrial robots with joint torque sensors,” in *26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017, pp. 838–843.

[25] O. Sim, J. Oh, K. K. Lee, and J. H. Oh, “Collision detection and safe reaction algorithm for non-backdrivable manipulator with single force/torque sensor,” *Journal of Intelligent Robotic Systems*, vol. 91, no. 3, pp. 403–412, 2018.

[26] X. Wang, C. Yang, Z. Ju, H. Ma, and M. Fu, “Robot manipulator self-identification for surrounding obstacle detection,” *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 6495–6520, 2017.

[27] S. Morikawa, T. Senoo, A. Namiki, and M. Ishikawa, “Realtime collision avoidance using a robot manipulator with light-weight small high-speed vision systems,” in *IEEE International Conference on Robotics and Automation*, 2007, pp. 794–799.

[28] D. Han, H. Nie, J. Chen, and M. Chen, “Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection,” *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 98–104, 2018.

[29] H. A. Sulaiman, A. Bade, and N. M. Suaib, “Balanced hierarchical construction in collision detection for rigid bodies,” in *International Conference on Science & Social Research*, 2010, pp. 1132–1136.

[30] W. Zhao and Y. Lan, “A fast collision detection algorithm based on distance calculations between nurbs surfaces,” in *IEEE International Conference on Computer Science and Electronics Engineering*, 2012, pp. 534–537.

[31] M. Tang, D. Manocha, and Y. J. Kim, “Hierarchical and controlled advancement for continuous collision detection of rigid and articulated models,” *IEEE Trans. Visual. Comput. Graphics*, vol. 20, no. 5, pp. 755–766, 2013.

[32] V. Macagon and B. Wnsche, “Efficient collision detection for skeletally animated models in interactive environments,” *Image and Vision Computing NZ*, pp. 378–383, 2003.

[33] M. Hassan, D. Liu, S. Huang, and G. Dissanayake, “Task oriented area partitioning and allocation for optimal operation of multiple industrial robots in unstructured environments,” in *In 13th International Conference on Control Automation Robotics Vision (ICARCV)*, 2014, pp. 1184–1189.

[34] M. Hassan and D. Liu, “Performance evaluation of an evolutionary multiobjective optimization based area partitioning and allocation approach,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2018, pp. 527–532.

[35] O. V. Kaick, N. Fish, Y. Kleiman, S. Asafi, and D. Cohen, “Shape segmentation by approximate convexity analysis,” *ACM Trans. Graph.*, vol. 34, no. 1, pp. 1–11, 2014.

[36] D. Delling, D. Fleischman, A. V. Goldberg, I. Razenshteyn, and R. F. Werneck, “An exact combinatorial algorithm for minimum graph bisection,” *Mathematical Programming*, vol. 153, no. 2, pp. 417–458, 2015.

[37] A. Adaldo, S. S. Mansouri, C. Kanellakis, D. V. Dimarogonas, K. H. Johansson, and G. Nikolakopoulos, “Cooperative coverage for surveillance of 3d structures,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1838–1845.

[38] A. Okabe, B. Boots, K. Sugihara, S. N. Chiu, and D. G. Kendall, *Spatial tessellations:Concepts and applications of Voronoi diagrams*. London: Wiley, 2008.

[39] M. Hassan and D. Liu, “Simultaneous area partitioning and allocation for complete coverage by multiple autonomous industrial robots,” *Autonomous Robots*, vol. 41, no. 8, pp. 1609–1628, 2017.

[40] M. Berg, O. Cheong, M. V. Kreveld, and M. Overmars, *Orthogonal Range Searching*. Springer Berlin Heidelberg, 2008.