Random Inspection Tree Algorithm in Visual Inspection with a Realistic Sensing Model and Differential Constraints

Přemysl Kafka, Jan Faigl *Member, IEEE*, Petr Váňa

***Abstract*— In this paper, we consider existing asymptotically optimal inspection planning algorithm in coverage path planning with realistic visibility constraints of standard cameras. Although the existing approach is able to provide an optimal solution with omnidirectional sensing and limited sensing range, it is prohibitively computationally expensive for problems with only few objects to be covered and limited field of view. Based on the analysis of the utilized sampling-based strategy, we propose a heuristic approach to decrease computational requirements in problems with restricted viewing frustum, which is a more realistic model of a digital camera. In addition, we also consider a minimal distance and angle under which the object to be covered is captured by the forward looking camera to make a snapshot of the object with the required details.**

I. Introduction

In mobile robotics, the inspection path planning can be considered as a problem to determine a cost efficient path along which a given vehicle covers (sees) all requested objects of interest with its sensor system while considering motion and sensing constraints of the vehicle. This problem can also be found as the coverage path planning (CPP) problem [1] in the literature. Strictly speaking, the CPP differs from the inspection planning in the way how the environment or objects of interest are covered. In the CPP, it is usually requested to visit each point of the environment by a tool to perform some task, e.g., clean a floor [2], while the inspection planning, mostly range measurements taken by cameras, sonars and laser scanners are considered. However, both terms can represent the same class of problems.

Two fundamental approaches have been proposed for inspection path planning. The first type of approaches are decoupled methods in which the problem is decomposed into two parts solved independently: the sensor placement followed by motion planning [3]. Thus, particular sensing locations to cover the objects of interest are determined prior finding a cost efficient path connecting the locations. In this way, a minimal number of locations can be found by algorithms for solving the art gallery problem with sensing constraints arising from robotic applications, e.g., limited field of view or incident angle of the laser beam with the surface of the scanned object [4]. Then, a cost efficient path is determined using motion planning technique or the problem is treated as the multi-goal path planning problem [5], which leads to the traveling salesman problem (TSP) [6].

The presented work has been supported by the Czech Science Foundation (GAČR) under research projects No. 15-09600Y and No. 16-24206S.

J. Faigl is with the dept. of Computer Science, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic faiglj@fel.cvut.cz

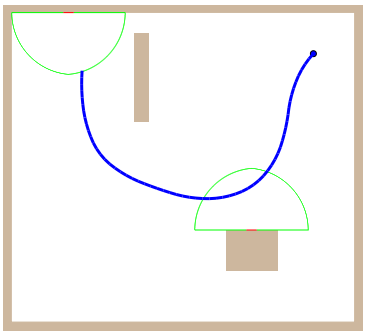


Fig. 1. An example of found inspection path for a nonholonomic system to inspect the object of interest (red parts of the obstacles with the size 5 cm) under visibility constraints to have the object of interest captured with at least 10 pixels in the image.

Although, the decoupled approach provides a feasible solution of problems with a high number of potential sensing locations, e.g., in coverage and surveillance missions [7], [8], its main drawback is that sensing locations are determined independently on the path and motion constraints of the inspection vehicle. Thus, it may happen that the cost of a particular path to connect two locations would be very high or would not be feasible at all. Therefore, a direct approach is a more suitable for non-holonomic robots.

A pure geometrical variant of the direct inspection planning can be formulated as the watchman route problem (WRP) which stands to determine a shortest path from which all points of the environment are covered. The WRP is known to be NP-hard for the polygonal map with obstacles [9], and therefore, approximate and heuristic algorithms have been proposed [10], [11]. However, found solutions are formed only from straight line segments and thus, these methods are not suitable for non-holonomic vehicles.

Authors of [12] proposed randomized sampling-based motion planning to directly provide an inspection path in configuration space of the vehicle with differential constraints. Moreover, the authors exploit the property of the asymptotic optimality of the Rapidly-exploring Random Graph (RRG) [13] algorithm and propose asymptotically optimal Random Inspection Tree Algorithm (RITA) for inspection planning with differential constraints. The authors reported results for relatively simple problems with an omnidirectional sensor with limited range for which solutions close to optimal are found in tens of minutes and up to two hours.

In this paper, we consider RITA in inspection path planning with sensing constraints of the forward looking camera with limited sensing range and field of view. Moreover, we consider the minimal distance and angle under which the object of interest is captured with sufficient details in the

image. Due to these sensing constraints, the area of v in the (k, v)-inspectionProblem (see [12]) is smaller than for an omnidirectional sensing. Thus, the problem may be a more difficult to solve by randomized sampling-based approaches and finding a solution can be more computationally demanding than the problem with omnidirectional sensing considered by the original RITA [12]. Therefore, we propose heuristics for improving the success rate to find a solution within a given runtime limit. In addition, we experimentally verified that found paths are feasible for a real mobile robot. The considered contributions of the paper are as follows:

- Deployment of RITA in problems with realistic sensing constraints of the forward looking camera;
- Improved performance of the roadmap expansion;
- Evaluation of RITA and the proposed modifications in visual inspection problems;
- Practical verification of the planned path with a real mobile robot in an open loop control.

The paper is organized as follows. The problem is formally defined in Section II together with a brief overview of the original RITA [12] to make this paper a more self-contained. A camera model and the proposed model of the sensing constraints are presented in Section III and the proposed RITA modifications are in Section IV. Evaluation results and experimental validation with a real robot are reported in Section V. Concluding remarks are dedicated to Section VI.

II. Problem Statement

The proposed approach is directly based on the original RITA [12], and therefore, we can consider the identical problem formulation. However, a 2D workspace is considered for simplicity and better readability, albeit the proposed sampling strategy is valid in 3D problems as for RITA.

The robot workspace $\mathcal{W} \subset \mathbb{R}^2$ is represented as a set of polygonal obstacles inside a simple boundary polygon. It is assumed that models of the environment and robot are known and they define the robot configuration space $\mathcal{C}$. The robot motion constraints can be described by $\dot{q} = g(q, u)$, where $q \in \mathcal{C}$ and u is the control input from a set of possible inputs $u \in \mathcal{U}$. The function g is a smooth function and for simplicity we consider a car-like robot with $q = (x, y, \theta)$, where x and y stand for a position of the robot in the world and θ is its orientation, and the control $u = (v_f, \varphi_s)$ consists of the forward velocity v_f and the steering angle φ_s.

A rigid body of the robot $\mathcal{A}$ is defined in $\mathcal{W}$ and the obstacle space $\mathcal{C}_{obstacle} \subset \mathcal{C}$ is a set of all configurations q for which the robot body $\mathcal{A}(q)$ at the configuration q is in a collision with obstacles $\mathcal{O}$, i.e., $\mathcal{C}_{obstacles} = \{q | q \in \mathcal{C} \text{ and } \mathcal{A}(q) \cap \mathcal{O} \neq \emptyset\}$. Then, the collision free part of $\mathcal{C}$ can be defined as $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obstacles}$.

A continuous function $\sigma : [0, 1] \to \mathbb{R}^2$ of the bounded variation $TV(\sigma) < \infty$ is called a path, where $TV(\sigma)$ is the total variation $TV(\sigma) = \sup_{\{0=\tau_0<\tau_1\ldots<\tau_n=s\}} = \sum_{i=1}^{n} |\sigma(\tau_i) - \sigma(\tau_{i-1})|$ [13]. In particular, we are interested in a collision-free path, i.e., $\sigma(\tau) \subset \mathcal{C}_{free}$ for $\tau \in [0, 1]$.

Regarding the motion constraints, a trajectory γ can be defined as the time-parametrized path induced by the control function $u : [0, T] \to \mathcal{U}$ for the motion model $\dot{q} = g(q, u)$. Having a feasible control input u, a feasible trajectory can be defined as $\gamma : [0, T] \to \mathcal{C}_{free}$ with respect to u.

Each particular object of interest $\mathcal{I}_i$ to be covered is defined by a straight line segment of the length d_i and its normal pointed at the half-plane from which it is requested to see the object. Then, a set of all points of the object $\mathcal{I}_i$ to be covered is denoted as $S(\mathcal{I}_i)$. We assume the value of d_i is such that there always exists a non-empty set of configurations $A(S(\mathcal{I}_i)) \subseteq \mathcal{C}_{free}$ from which $\mathcal{I}_i$ can be completely covered from some configuration q, $q \in A(S(\mathcal{I}_i))$.

In addition to feasibility of the path, we also request that all objects of interest $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$ are visible (inspected) from the trajectory according to the sensing constraints. We denote the set of all points of the object $\mathcal{I}_i$ that are inspected from q with respect to the sensing constraints as $V_{\mathcal{I}_i}(q)$, detailed in Section III. For simplicity, we assume that the whole object $\mathcal{I}_i$ can be covered from a single configuration q and thus, the object $\mathcal{I}_i$ is covered from a feasible trajectory γ if there exist at least one configuration q_x of the trajectory $q_x \in \gamma$ such that $S(\mathcal{I}_i) \subseteq V_{\mathcal{I}_i}(q_x)$, i.e., there exists a time τ_x for which $q_x = \gamma(\tau_x)$ and the robot covers $\mathcal{I}_i$ from q_x using its sensor system. We call a feasible trajectory γ as an admissible trajectory if all objects of interest are covered: $\bigcup_{i=1}^{n} S(\mathcal{I}_i) \subseteq \bigcup_{\tau=0}^{T} V(\gamma(\tau))$.

Based on the presented preliminaries, we can define the inspection path planning problem. Having an initial configuration q_{init} we aim to find the shortest trajectory γ^* such that the trajectory γ^* is a collision-free, feasible, and admissible. Let the length of a trajectory γ be $\mathcal{L}(\gamma)$, then we aim to determine $\gamma^* = \operatorname{argmin}_{\gamma \in \Gamma} \mathcal{L}(\gamma)$, where Γ is a set of all possible admissible trajectories. In particular, we aim to find the control function $u^*(t)$ that induces γ^* [12].

A. Overview of RITA

The proposed approach is directly based on RITA that has been introduced in [12], and therefore, only a brief overview is presented here. RITA is schematically depicted in Algorithm 1 and it works as follows. The algorithm starts from some initial configuration q_{init} and iteratively expands the roadmap represented as the graph (tree) $G = (V, E)$, where each node $v \in V$ represents a particular configuration of $\mathcal{C}_{free}$ and each edge $e \in E$ represents a feasible trajectory between two configurations with the associated control $u \in \mathcal{U}$. The roadmap is expanded by the *ExpandTree* procedure. In original RITA, the procedure expands the tree from a configuration that is randomly selected according to the probability that is proportional to the inverse number of the nodes within its neighbourhood:

1) Determine the number of nodes in the neighbourhood: $D_n(q), \forall q \in V$;
2) Determine the probability to chose a node q:

$$P_n(q) \propto 1/D_n(q), \forall q \in V. \quad (1)$$

After the node selection, a random control input $u \in \mathcal{U}$ is applied and the procedure returns a new configuration q_n, its parent configuration q_p, and a trajectory $\sigma(q_n, q_p)$ for the applied u. If $\sigma(q_n, q_p)$ is collision free and q_n improves the

Algorithm 1: RITA

Input: q_{init} – an initial configuration of the robot
Input: i_{max} – the maximal number of iterations
Output: $G = (\boldsymbol{V}, \boldsymbol{E})$ – the motion planning roadmap
Output: q_{best} – the best found solution
begin
  Initialization: $G \leftarrow \{\boldsymbol{V} \leftarrow \emptyset, \boldsymbol{E} \leftarrow \emptyset\}$;
  $\boldsymbol{V} \leftarrow q_{init}$; $bestCost \leftarrow \infty$; $q_{best} \leftarrow \emptyset$; $i \leftarrow 0$;
  while $i < i_{max}$ **do**
    $\langle q_n, q_p, \sigma(q_n, q_p), u \rangle \leftarrow ExpandTree(G)$;
    if *isCollisionFree*$(\sigma(q_n, q_p))$ **then**
      $cost \leftarrow Cost(q_p) + Cost(\sigma(q_n, q_p))$;
      if $cost < bestCost$ **then**
        $\boldsymbol{V} \leftarrow \boldsymbol{V} \cup q_n$; $\boldsymbol{E} \leftarrow \boldsymbol{E} \cup (q_p, q_n)$
        $vis \leftarrow Visibility(q_n) \cup q_p.vis$
        if $q_n.unseen = \emptyset$ **then**
          $bestCost \leftarrow cost$
          $q_{best} \leftarrow q_n$
    $i \leftarrow i + 1$

cost of the current solution, q_n is added to the roadmap. In addition, if a trajectory from q_n provides a new admissible solution with a lower cost, q_n becomes q_{best}, from which the final trajectory is determined using its parents.

III. Camera Model and Sensing Constraints

The studied problem is motivated by inspection missions with a single forward looking camera utilized for capturing images of objects of interest. From a pinhole camera model, we can establish relation of the object height y and its distance to the camera as $d = (yd')/y'$, see Fig. 2. However, an effective distance depends on the pixel resolution, sensor size, and also on the size of the object to be covered.

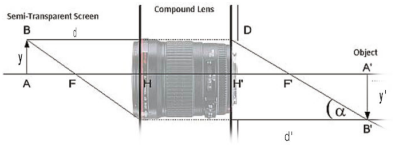


Fig. 2. Pinhole camera model

Let the desired number of pixels the object has to occupy in the image be N_{pix}, then the maximal effective distance d_{max} can be derived as

$$d_{max} = \frac{d' \cdot y}{N_{pix} \cdot p_{size}}, \tag{2}$$

where p_{size} is the real size of the pixel. Similarly, the minimal effective distance to capture the whole object of the size y can be derived as:

$$d_{min} = \frac{y}{tan(\alpha/2)}, \tag{3}$$

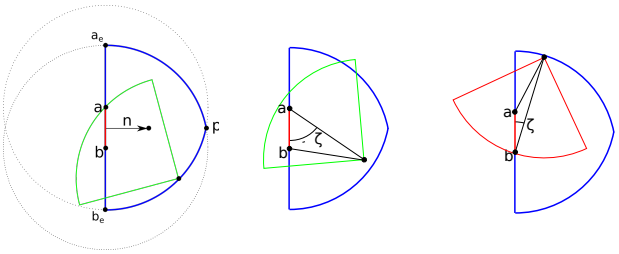
where $\alpha/2$ is the required field of view to capture the object of size y that is centered to the optical axis. α depends on the camera lens and for the focal length f and the size of the object in the image sensor y', it can be established as

$$\alpha = 2 \arctan\left(\frac{y'}{2f}\right). \tag{4}$$

In the rest of this paper, we consider 78.6° field of view and the maximal view distance 2 m.

A. Sensing Constraints

A geometrical model of the part of $\mathcal{C}_{free}$ from which an object of interest can be covered is desirable for a quick evaluation of the sensing constraints satisfaction during inspection planning. However, its exact geometrical shape is too complex to be used, because the robot yaw and its distance to the object have to be considered.



(a) (b) (c)

Fig. 3. Geometric relation of the camera coverage area (green and red) and the area from which the object $\mathcal{I}_i$ (defined as the segment a–b) can be covered (blue). Examples of the object (a–b segment) observability for which the angle sensing constraints between the camera axis and object to be covered: (**b**) are satisfied; (**c**) are not satisfied. The red outline of the camera coverage area does not satisfy the angle constraint and the segment a–b is not covered from that position.

Based on [14], we propose approximation of the area from which an object of interest $\mathcal{I}_i \in \mathcal{I}$ can be covered. The geometric definition of the area is depicted in Fig. 3a. The object $\mathcal{I}_i$ is represented as the straight line segment a–b with the accompanied normal vector $\boldsymbol{n}$. The area from which $\mathcal{I}_i$ can be covered is defined by two circles c_a and c_b centered at a and b, respectively, with the radius d_{max}. The intersection point p of c_a and c_b (at the side of $\boldsymbol{n}$) together with a_e and b_e form approximation of the geometrical shape of the covering area, see the blue outline in Fig. 3a.

Although the proposed geometrical primitive is not a precise shape of the area containing all configurations from which $\mathcal{I}_i$ can be covered, it allows a fast evaluation to quickly discard not promising configurations. Notice, regarding the angle constraints to have sufficient details of the object in the image, a configuration inside the area must be further evaluated for the current robot yaw and distance to $\mathcal{I}_i$.

B. Object Observability Test

The object $\mathcal{I}_i$ is considered to be covered if its snapshot in the image provides sufficient details. The introduced geometric shape provides the first initial evaluation if a configuration q can provide sufficient coverage of $\mathcal{I}_i$. Then, additional tests are incrementally performed to further evaluate coverage of the object. The evaluation of the coverage of $\mathcal{I}_i$ can be summarized as follows:

1) The initial test directly uses the geometric shape of the camera coverage area, see Fig. 3a. This test is fast and it filters a majority of false positive covering attempts.

2) The visibility distance constraint consists of d_{max} and d_{min}, which is easy to check using Euclidean distance of the camera position and the object end points a and b.

3) The orientation of the object to the camera has to provide sufficient details of the captured object. An evaluation of this constraint needs to consider the incident angle and thus, it depends on the camera orientation and its distance to the object. A situation with and without satisfied incident angle is depicted in Fig. 3b and Fig. 3c, respectively.

4) The above tests do not cover the case when an obstacle is in the field of view, which is the most computational expensive test. However, the object of interest is considered as a segment and its end points together with the camera position form a triangle. Hence, the same collision detection mechanism as in RITA can be utilized, e.g., using RAPID library [15].

IV. Improving RITA Performance

Although the proposed sensing constraints can be directly applied to the original RITA [12], the angle constraint together with the restricted field of view significantly reduce the number of admissible trajectories, e.g., see Fig. 4. Moreover, we found out that the most time consuming operation of the original RITA is the tree expansion. In particular, the determination of the probability to select the node for an expansion (1), which consists of neighbor search and update of the individual node probability to be selected.

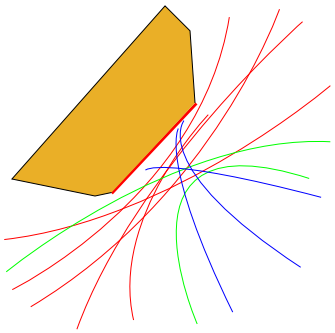
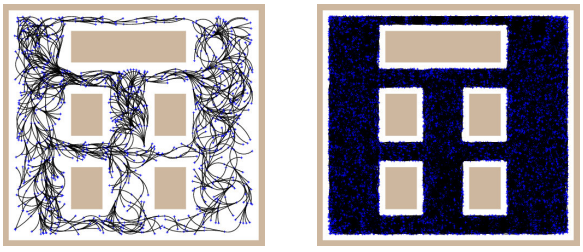


Fig. 4. An example of the object attached at the obstacle (red segment at the yellow obstacle) with many feasible trajectories (in green, red, and blue) while only two trajectories are admissible (in green). Notice, the upper green trajectory is admissible only for a direction from right to left.



(a) k=2 000 (b) k=50 000

Fig. 5. Example of roadmap evolution after k expansion steps of RITA

The computation of p_{select} is computationally demanding because basically each node of the roadmap $G = (V, E)$ carries the information about its probability p_{select}. Therefore, we have investigated how the roadmap and probabilities evolve during the roadmap construction. We found out that after 50 000 iterations almost all space is covered by samples and we can suppose that differences in particular probabilities p_{select} of the individual nodes are decreasing with the increasing number of iterations, see Fig. 5. Hence, we considered the maximal squared error ϵ^2 of the individual probabilities from the mean value of the actual probabilities p_{select} of all nodes in the roadmap. An evolution of ϵ^2 during the roadmap construction is shown in Fig. 6.

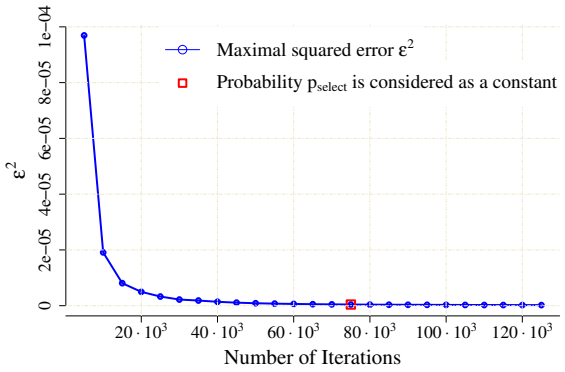


Fig. 6. An evolution of ϵ^2 during the roadmap construction

The evolution indicates that the individual values of p_{select} become similar and thus, it may not be necessary to compute individual probabilities after each successful roadmap expansion. Therefore, we propose a modified variant of RITA with *Constant Selection Probability* called RITA$_{\text{CSP}}$ in which we compute ϵ^2 at each predefined iteration, e.g., after each 5 000 iterations, and once its value converges to a constant value, the probability p_{select} is considered as constant, i.e., the uniform distribution, for the rest of the expansions.

A. Prioritization of Covering Configurations

The proposed constant value of p_{select} does not need a computationally intensive determination of the probability values. However, it does not provide any mechanism to prefer nodes from which the next expansions can cover further objects. This motivated us to bias individual probabilities to prefer selection of nodes covering the objects as

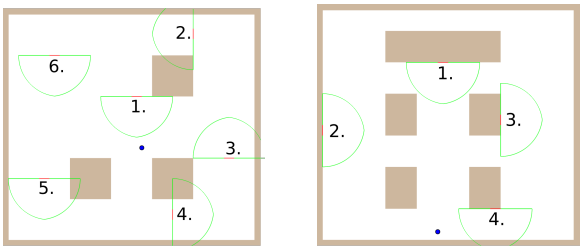
$$\tilde{p}_{pri} = p_{select}(Covered(v_i) + 1)^2, \qquad (5)$$

where p_{select} is the value determined in RITA$_{\text{CSP}}$ and $Covered(v_i)$ is the number of objects covered along the path from the root to the node v_i. After the prioritization, the probabilities are normalized, i.e., particular "slices" of the roulette wheel used for the node selection are adjusted appropriately. We call this modification as RITA$_{\text{pri}}$.

V. Results

Performance of RITA and the proposed modifications has been evaluated for a forward looking camera attached to a vehicle with the minimal turning radius 0.5 m moving with the constant (unit) forward velocity. Two 12×11 meters large environments called *squares* and *rectangles* have been considered with different numbers of objects $\mathcal{I}$, see Fig. 7.

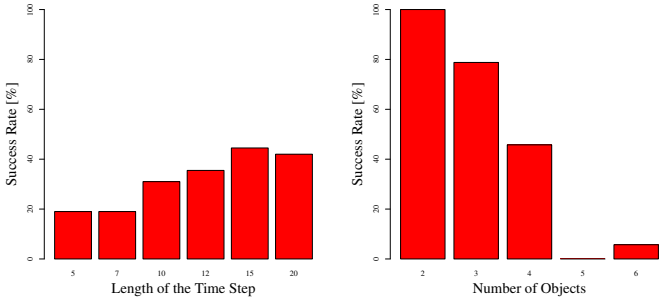
Each object is 5 cm long and it is requested it occupies at least 10 pixels in the captured image. The objects labels indicate which objects have to be covered in a particular scenario, e.g., for the *squares* environment and 5 objects of interest, the objects 1. to 5. are requested to be covered.



(a) environment *squares* (b) environment *rectangles*

Fig. 7. Environments with the objects of interest. Blue point represents the initial robot configuration with the yaw equals to 90°.

The main performance indicator is the *success rate* computed as the ratio of the number of trials in which an admissible solution has been found to the total number of the performed trials for a particular scenario and the considered value of i_{max}. The reported computational time has been obtained for a single core C++ implementation and Intel Xeon X5660 running at 2.8 GHz with 48 GB RAM.



(a) (b)

Fig. 8. Success rate of finding an admissible path by RITA according to: (**a**) the length of the time step; and (**b**) the number of objects of interest.

Prior a comparison of the original RITA and the proposed modifications, we determine the most suitable time step for the roadmap expansions, because it is a crucial parameter of the sampling-based motion planning methods. In general, a lower value causes the generated tree expands in more various ways, but the expansion is slower than for higher values which provide faster expansion but finding a path through narrow passages can be more demanding. The environment *squares* with 2–6 objects has been used with the time step from the set {5, 7, 10, 12, 15, 20}, which gives 40 different scenarios. For each scenario, 5 trials have been evaluated with the maximal number of iterations set to i_{max}=10^6. The overall success rate for increasing time step is depicted in Fig. 8a from which we selected the time step 15 to support a high success rate of the original RITA approach. Besides, the influence of increasing number of objects to be covered to the success rate is depicted in Fig. 8b.

A. Performance Improvements

The computational time depends on the number of nodes in the roadmap which increases with the number of performed iterations. Therefore, the CPU time at the particular iteration has been recorded for the *squares* map with 6 objects and all three variants of RITA, see Fig. 9. Based on these results, RITA$_{\text{CSP}}$ provides superior performance.

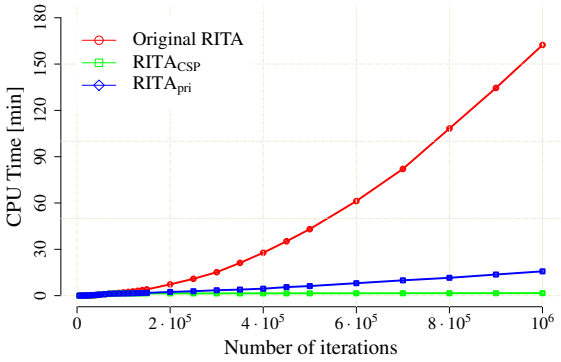


Fig. 9. Required CPU time according to the number of performed iterations for the original RITA and the proposed RITA$_{CSP}$ and RITA$_{pri}$ modifications in the *squares* map and 6 objects of interest

RITA$_{\text{pri}}$ is more demanding because of probability updates (5), but it is still less demanding than original RITA.

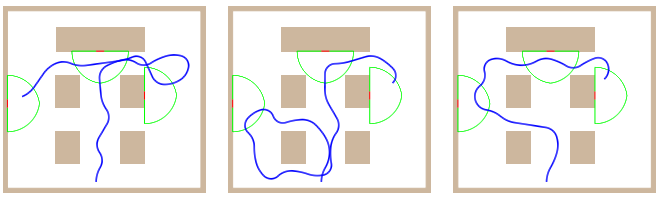
Although the proposed modifications decrease the computational burden and thus they may generate a denser roadmap within the same computational time as original RITA, the most important aspect of the inspection planning is the ability to find an admissible solution.

TABLE I

SUCCESS RATE – *squares* SCENARIOS

Number of objects of interest	**Original RITA** Success rate **[%]**	**RITA$_{\text{CSP}}$** Success rate **[%]**
4	100%	100%
5	20%	80%
6	0%	0%

Results depicted in Fig. 8b indicate that the original RITA needs a high number of iterations to find an admissible inspection path for limited sensing capability of the robot. Therefore, the maximal number of iterations has been increased to i_{max}=10^7 for the *squares* environment and different number of objects of interest. The success rate from 10 trials is depicted in Table I, where RITA$_{\text{CSP}}$ provides better performance than original RITA. For 5 objects, RITA needs about 3 hours to find an admissible solution after approximately 2·10^6 iterations. Both approaches need almost the same number of iterations to find a solution; however, the proposed RITA$_{\text{CSP}}$ is much faster. Examples of found admissible inspection paths are in Fig. 10.



(a) k=73 137 (b) k=1 055 823 (c) k=1 634 645

Fig. 10. Example of the inspection path after k expansions in the *rectangles* environment with three objects of interest and the proposed RITA$_{\text{CSP}}$ modification

Due to a poor performance of the original RITA, the

rectangles environment has been considered only with the proposed modifications. The average number of the roadmap nodes, the average length of the final found solution (denoted as L), and the average required computational time (denoted as T_{CPU}) from 10 trials for 3 and 4 objects of interest are listed in Table II. Notice, for $RITA_{CSP}$, the maximal number of iterations has been set to $2{\cdot}10^7$; otherwise a solution for 4 objects is not found by $RITA_{CSP}$ albeit solutions for 3 objects of interest are found with a high success rate for i_{max}=$2{\cdot}10^6$. For the greedy expansion of $RITA_{pri}$, it is sufficient to set i_{max} to 10^6 to find an admissible solution.

TABLE II
PERFORMANCE OF $RITA_{CSP}$ AND $RITA_{PRI}$ IN *rectangles* SCENARIOS

Method	**Objects**	**Success Rate [%]**	**Number of Nodes**	**L** [m]	$\mathbf{T_{CPU}}$ [s]
$RITA_{CSP}$	3	100%	7 517 782	20.8	274
$RITA_{CSP}$	4	20%	11 114 477	30.2	326
$RITA_{pri}$	4	100%	220 525	29.0	944

The preference of covering nodes in $RITA_{pri}$ provides a roadmap with a significantly less number of nodes. A higher computational time of $RITA_{pri}$ indicates that many expansions are not successful. On the other hand, $RITA_{CSP}$ has a lower success rate despite of a denser roadmap. The proposed greedy preference of the roadmap nodes from which objects are covered steers the expansions towards coverage of the further objects and thus, $RITA_{pri}$ is able to provide admissible solutions in spare roadmaps. Although it may result in stucking in a local minima without finding a solution, the presented results (for the considered scenario) indicate that the performance of RITA can be improved by a more sophisticated p_{select}.

B. Use case – Experiment with a Real Robot

The proposed $RITA_{CSP}$ has been also verified in real deployment to further validate that found trajectories are directly executable by a mobile robot. Because at the time of writting this paper, a real car-like robot has not been available for experimenting, we considered a hexapod walking robot that has been restricted to the car-like kinematics using a limited set of motion primitives [16]. Thus, the robot was able to move only forward with a particular turning radius according to the steering angle as a car-like vehicle. Snapshots of the robot performing the planned inspection path are depicted in Fig. 11.

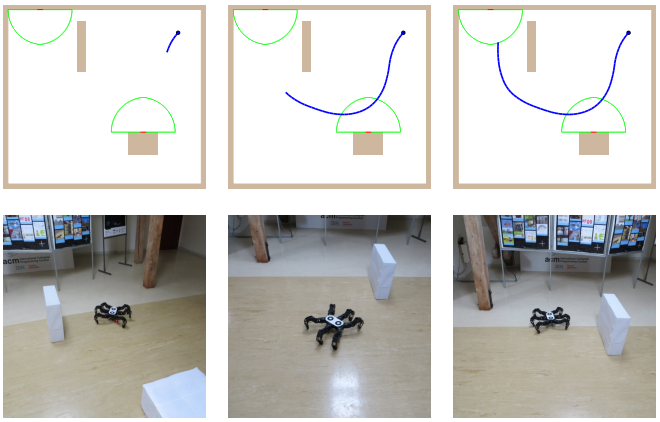


Fig. 11. Hexapod walking robot considered with car-like kinematics performing the planned inspection path in an open loop control. Circular markers on the body of robot serve for the robot localization.

VI. CONCLUSION

Asymptotically optimal inspection planning algorithm called RITA has been investigated in problems with more realistic sensing constraints of the forward looking camera. Although RITA is a general approach to solve inspection planning for vehicles with differential motion constraints, its practical performance is poor because it is too computationally demanding even in problems with few objects of interest.

The proposed modifications decrease the computational burden while they also improve success rate of finding admissible solution for a limited number of roadmap expansions. On the other hand, the evaluated problems can be still considered as relatively small, and therefore, additional improvements have to be proposed to solve larger problems, which is a subject of our future work.

REFERENCES

[1] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[2] E. Park, K. Kim, and A. del Pobil, "Energy efficient complete coverage path planning for vacuum cleaning robots," in *Future Information Technology, Application, and Service*, ser. Lecture Notes in Electrical Engineering, 2012, vol. 164, pp. 23–31.

[3] H. González-Baños, D. Hsu, and J.-C. Latombe, "Motion planning: Recent developments," in *Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications*, S. Ge and F. Lewis, Eds. CRC Press, 2006, ch. 10.

[4] H. González-Baños and J.-C. Latombe, "Planning Robot Motions for Range-Image Acquisition and Automatic 3D Model Construction," in *AAAI Fall Symposium*, 1998.

[5] T. Danner and L. E. Kavraki, "Randomized Planning for Short Inspection Paths," in *ICRA*, 2000, pp. 971–976.

[6] S. Alatartsev, S. Stellmacher, and F. Ortmeier, "Robotic task sequencing problem: A survey," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 2, pp. 279–298, 2015.

[7] B. Englot and F. S. Hover, "Three-dimensional coverage planning for an underwater inspection robot," *Int. J. Rob. Res.*, vol. 32, no. 9-10, pp. 1048–1073, 2013.

[8] P. Janousek and J. Faigl, "Speeding up coverage queries in 3d multi-goal path planning," in *ICRA*, 2013, pp. 5082–5087.

[9] F. Li and R. Klette, "An approximate algorithm for solving the watchman route problem," in *Robot Vision*, ser. Lecture Notes in Computer Science, 2008, vol. 4931, pp. 189–206.

[10] E. Packer, "Computing multiple watchman routes," in *Experimental Algorithms*. Springer, 2008, pp. 114–128.

[11] J. Faigl, "Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1668–1679, 2010.

[12] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using system with differential constraints," in *ICRA*, 2013, pp. 4111–4118.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[14] B. Englot and F. Hover, "Sampling-base coverage path planning for inspection of complex structures," vol. 22, 2012, pp. 29–37.

[15] "Rapid - robust and accurate polygon interference detection," [cit. 11-09-2015]. [Online]. Available: http://gamma.cs.unc.edu/OBB/

[16] P. Vanek, J. Faigl, and D. Masri, "Multi-goal trajectory planning with motion primitives for hexapod walking robot," in *ICINCO*, 2014, pp. 599–604.