Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-017-0768-2

Optimal Perception Planning with Informed Heuristics Constructed from Visibility Maps

Tiago Pereira[1,2,3] · António Moreira[2,3] · Manuela Veloso[1]

Received: 19 November 2016 / Accepted: 27 December 2017

Abstract
In this paper we consider the problem of motion planning for perception of a target position. A robot has to move to a position from where it can sense the target, while minimizing both motion and perception costs. The problem of finding paths for robots executing perception tasks can be solved optimally using informed search. In perception path planning, the solution when considering a straight line without obstacles is used as heuristic. In this work, we propose a heuristic that can improve the search efficiency. In order to reduce the node expansion using a more informed search, we use the robot Approximate Visibility Map (A-VM), which is used as a representation of the observability capability of a robot in a given environment. We show how the critical points used in A-VM provide information on the geometry of the environment, which can be used to improve the heuristic, increasing the search efficiency. The critical points allow a better estimation of the minimum motion and perception cost for targets in non-traversable regions that can only be sensed from further away. Finally, we show the contributed heuristic with improvements dominates the base PA* heuristic built on the euclidean distance, and then present the results of the performance increase in terms of node expansion and computation time.

Keywords Perception planning · Heuristic search · Visibility maps

1 Introduction

In this work we deal with motion planning for perception tasks, where both the motion and sensing costs have to be considered in order to find an optimal path.

As we show in Fig. 1, a path has motion cost $cost_m$, proportional to the distance traveled, and a perception cost $cost_p$, which is a function of the perceiving distance between the final path position and the target. Here we assume the sensing cost is a function of the minimum distance between the path and the target. As a result, the final position of the path has the minimum distance to the target. We consider any perception cost function, as long as it is monotonically increasing with the perception distance. We use λ as the parameter that trades-off motion and perception cost. The path selection changes depending on the relative costs of motion and perception minimizing the overall cost.

Using an informed search algorithm, such as PA* [15], it is possible to explore the space and find the optimal path while reducing the number of expanded nodes compared to breadth-first search. However, the basic algorithm searches the environment without considering any information from the world. As in the common motion planning problem with A*, the heuristic in a certain node is determined using the solution for the straight line perception task between the node and the target, without considering obstacles. Assuming that no obstacle information is given, PA* searches the state space optimally, finding the optimal path given a perception function and parameter λ.

We contribute a new heuristic that can reduce the number of expanded nodes when the perception target location lies in non-traversable regions, where there is a minimum perception distance. For that purpose, we use the Visibility

✉ Tiago Pereira
tpereira@andrew.cmu.edu

António Moreira
amoreira@fe.up.pt

Manuela Veloso
mmv@cs.cmu.edu

1 Carnegie Mellon University, Pittsburgh, PA, USA
2 Faculty of Engineering, University of Porto, Porto, Portugal
3 INESC Technology and Science, Porto, Portugal

Published online: 12 February 2018

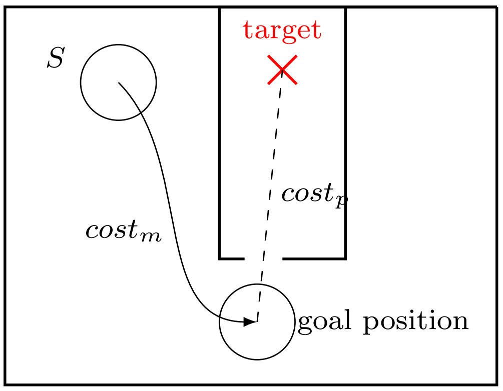


Fig. 1 The cost of a path is given by the sum of the motion cost and the perception cost; an informed heuristic search can find the optimal path; in this example, the target is inside an unreachable region, so there is always a lower bound to the minimum perception distance independent of the optimal path

Map [16], a transformation on the original map that can be used to solve the observability problem, i.e., determining which regions are visible by a circular robot from any point reachable from the initial robot position. The parameters of this transformation are the robot size and maximum sensing range.

The Visibility Map shows what regions are visible by the robot. So, if a perception target lies in a region that is not visible from the initial robot position, our proposed algorithm can return a failed plan immediately without spending any time in search. This is an important feature, because when there is no feasible path the base PA* algorithm always explores the entire search space looking for a solution.

There can be other gains in the number of node expansion if more information is used for the heuristic. In the Approximate Visibility Map, critical points are used in order to estimate the visibility inside non-traversable regions, while reducing the computation time compared to the brute-force algorithm. In this work we prove that critical points, by definition the points in the configuration space that generate the frontiers of motion reachability, are the closest to any target position inside unreachable regions. Therefore, we use the distance from the target to each critical point as an estimate of the minimum perception distance. The distance to the critical point can be used to create an admissible heuristic that outperforms the heuristic in PA*, which is only a function of the distance between the current node and the target. The critical point can also be used to have an estimate of the minimum motion distance.

We prove that our contributed heuristic that considers the Visibility Map is a dominant and consistent heuristic compared to the one from the original PA*, yielding a faster convergence to the optimal path.

We present related work, and then review the methods of PA* and Approximate Visibility Maps. Then we describe the new heuristic used in the search algorithm of motion planning for perception tasks, proving it is admissible and dominant over the base heuristic. We show results yielding a faster convergence to the optimal path. We show results on the increased efficiency in terms of node expansion, and then present our conclusions and future work directions

2 Related Work

Many robotic applications consider perception separately from planning, with both being computed interleaved [19]. It has been used for tasks as varied as SLAM [2], robot localization [1], in exploration to guide robots to unexplored regions [8], and for object recognition [4].

In our work, perception is the task that drives planning, and we develop a framework for motion planning to be able to perceive a target, while considering both the motion and perception costs. There is also a work where planning and perception are integrated and unified in a manipulation task, using a belief space of probability distribution over states as a planning framework [9]. Others plan for perception, but not to sense a target location. Instead they plan in unknown environments, so the perception is performed as a task to increase accuracy, being a part of the set of possible actions [7].

Perception got recently a more active role in planning. An example is object detection, where the next moves of the robot should be planned to maximize the likelihood of both correct object detection and classification [18, 20]. In [3], probabilistic active perception is planned for realistic environments, with arbitrary object positions. The planning problem is designed as a partially observable Markov decision process. An active planning tackles the problem of occlusions by reasoning about model and state transition uncertainties.

Another class of problems for perception planning is the inspection problem. In order to determine a path that can sense multiple targets, a neural network approach was used to solve the NP-hard Watchman Routing Problem. In order to do so efficiently, a fast method was proposed to answer visibility queries [6]. In that work, the solution is not optimal due to the existence of multiple targets.

PA*, a heuristic search, was proposed to solve the motion planning problem for perception of a target position in 2D, given motion and perception costs [15].

It was also proposed in the past that robots maintain reachabilities and visibilities information, both of a robot and a human partner in a shared workspace. However, it uses non-mobile robotic platforms [12]. Visibility graphs are considered in [10], but the focus is on generating points for a patrolling motion plan. It assumes vectorial obstacles, so visibility can easily be calculated using ray casting at the extremes of lines. However, most maps are obtained as detailed 2D gridmaps, and we do not want to limit the problem with a vectorial representation, allowing obstacles to take any shape.

Morphological operations have also been used in robotics to determine the actuation space of a robot [13], which is later used to coordinate multi-robot teams. In our work we use a similar technique, but we can determine visibility for a sensing range bigger than robot size. We use the idea of critical points to increase the efficiency of computation, and also as a tool to improve the heuristic of PA*. It is also possible to use a similar approach to find visibility for any-shape robots [16], by discretizing orientation as well.

We already used visibility maps as a pre-processing step for planning, improving search efficiency for perception tasks by introducing dominant heuristics [14].

Techniques borrowed from image processing have already been used for map transformation, e.g. automatically extracting topology from an occupancy grid [5]. They robustly find the big spaces in the environment like humans would, separating it into regions.

3 Optimal Path Planning for Perception Tasks

PA* is a heuristic search for motion planning that returns the optimal path to perceive a target, considering both motion and perception cost [15].

In our perception scenario, we have to find a path ρ that not only minimizes distance traveled, but also minimizes the perception cost. The path ρ is a sequence of adjacent positions in a grid, $\{s_0, s_1, ..., s_n\}$. The robot starts from the initial position $s_0 = S$ and moves through connected cells of the discretized configuration space to a final robot position, $s_n = F$, such as the sensing target T is perceived from some position in the path ρ. The total cost of path ρ is given by

$$\text{cost}(\rho) = \text{cost}_m(\rho) + \lambda \text{cost}_p(\rho, T) \tag{1}$$

where λ is a weight parameter that trades-off the motion cost $\text{cost}_m(\rho)$, and the perception cost $\text{cost}_p(\rho, T)$ in the overall cost function.

Usually the motion cost cost_m is proportional to the distance traveled from S to F.

$$\text{cost}_m(\rho) = \sum_{i=1}^{n} ||s_i - s_{i-1}|| \tag{2}$$

Here we assume that $\text{cost}_p(\rho, T)$ is a function of the minimum distance between the path ρ and target T, making the perception cost a function of the minimum sensing distance from the path to the target.

$$\text{cost}_p(\rho, T) = c_p\Bigg(\min_{\substack{s_i \in \rho, \\ s_i \text{ with line-of-sight to } T}} ||s_i - T||\Bigg) \tag{3}$$

with c_p being a continuous function that depends on the sensor model. In the perception tasks, one of the goals is to increase the accuracy of perception. In our work, we represent the accuracy of perception as a cost function, where a low sensing accuracy corresponds to a high perception cost, and vice-versa. Therefore, the cost of perception is a function of the sensing distance. We assume that the further away the robot is from the target, the lower the sensing accuracy, because the probability of inaccurate measurements increases. Thus the cost function for perception, c_p, increases with distance.

The optimal path is given by

$$\begin{aligned} \rho^* &= \underset{\rho \in \mathcal{P}}{\text{argmin}}\, \text{cost}(\rho) \\ &= \underset{\rho \in \mathcal{P}}{\text{argmin}}\, \text{cost}_m(\rho) + \lambda \text{cost}_p(\rho, T) \end{aligned} \tag{4}$$

where $\mathcal{P}$ is the space of all possible paths.

Theorem 1 *For the optimal path ρ^*, the position that minimizes the distance from the path to sensing target T is the final position of the path, F.*

Proof If there were another position s_i in the middle of the path that had the smallest distance to the target, then there would be a different path ending in s_i with minimal cost, contradicting the hypotheses that the path ρ^* from S to F is the one that minimizes the overall cost. □

The space of all possible paths $\mathcal{P}$ is a general notation for representing the minimization problem to find the optimal path, but our proposed algorithm does not implement any search in the path space. We are going to show in the following sections how to use the A* architecture to do an informed search that builds the optimal path by moving between neighbor cells from the initial position S to the final goal position F. We can use heuristic search for this problem because the first theorem states that the perception cost is a function only of the distance of the last point of the path to the target, and as such it can be included in the heuristic estimate of each node.

3.1 Informed Search for Perception Planning

A* is a graph search algorithm that finds the lowest cost path from a given initial node to a goal node. It also works with discretized representations of the environments such as gridmaps, where each grid position represents a node, and the graph connectivity can be either based on 4 or 8-neighborhood. A* explores the environment by computing a heuristic cost function for each visited node that estimates the cost to reach the target. It moves through the search space by selecting the nodes with lower overall cost.

In traditional motion planning, a node is tested as goal position just by checking if the coordinates of the current node are the same as the coordinates of the target. Moreover, the total cost estimate is given by the path cost from S, the starting position, to the current node n, plus a heuristic of the cost from n to the target position T.

$$f(n) = g(S, n) + h(n, T) \tag{5}$$

If the heuristic used is admissible, i.e., always less or equal than the true value, then the path returned is guaranteed to be optimal. Therefore, the usual choice for the heuristic in motion planning (MP) is just the euclidian distance between the current node and the target, without considering any obstacles

$$h_{mp}(n, T) = ||n - T|| \tag{6}$$

As in the A* algorithm, in PA* the total cost estimate is also given by the sum of $g(S, n)$, the path distance from the starting position S to the current node n, and $h(n, T)$, which here is a heuristic of both the motion and perception costs from n to T. In order for the heuristic to be admissible, it is based on the euclidean distance between the current node and the target, without considering any obstacles.

The heuristic in PA* still uses the straight line between current node and sensing target, without considering obstacles, as A* does. Nevertheless, it now considers the expected cost of both approaching the target and sensing from a smaller distance.

$$h_{pp}(n, T) = \min_{q} \Big(||n - q|| + \lambda c_p(||q - T||) \Big) \tag{7}$$

We assume that from position n the robot can still approach the target by moving to other location q, from where it senses the target. There is a trade-off between the possible increase of motion cost, and the decrease of perception cost. We take the distance between points n and q as the approaching cost.

3.2 Optimal Heuristics for PA*

With the previously presented problem formulation, it is possible to solve the perception planning problem with A*, introducing a new heuristic that takes into account the perception cost. For that purpose, we will consider the problem of finding the optimal sensing position in a continuous straight line scenario without obstacles, as presented in Fig. 2. The idea is to use the solution for the straight line problem without obstacles as an admissible heuristic for the perception problem.

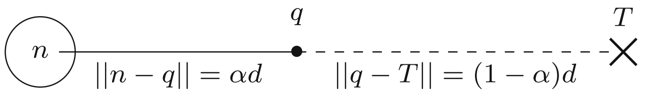


Fig. 2 Given a robot at position n and a perception target T at distance d in a scenario without obstacles, the optimal sensing goal position lies in the straight line connecting those two points. The image shows a solution with motion of αd and sensing distance equal to $(1 - \alpha)d$

With $||n - T|| = ||n - q|| + ||q - T|| = d$, the cost of motion and perception from node n to the target T is:

$$\text{cost}(\alpha, d) = |\alpha d| + \lambda c_p\big(|(1 - \alpha)d|\big) \tag{8}$$

where $\text{cost}(\alpha, d)$ is a continuous function of the overall cost when the robot is at a distance d to the target. The variable α represents the percentage of the distance the robot can approach the target, and $1 - \alpha$ the percentage of the distance d that is sensed. In order to have the optimal solution, we need to find α that minimizes cost.

$$\alpha^* = \underset{\alpha}{\text{argmin}}\, |\alpha d| + \lambda c_p(|(1 - \alpha)d|) \tag{9}$$

Therefore, the first step is to find the percentage that minimizes the cost of approaching and sensing the target in a straight line. This can be done analytically or numerically, depending on the function c_p. Later we will show optimal solutions for specific polynomial functions.

Lemma 1 *If c_p is a positive and monotonically increasing function, then $0 \leq \alpha^* \leq 1$, i.e., the optimal solution for the sensing goal position in the straight line perception task lies between the current node n and T.*

Proof The cost of motion is always positive and increases with traveled distance. Using the previous definition for perception cost, it is a function of sensing distance, thus it is positive and increases monotonically. Thus, the assumption for the perception cost function holds. Assuming a straight line distance with size d, the optimal decision of approaching the target by $\alpha^* d$ and sensing from distance $(1 - \alpha^*)d$ has a cost that is minimal. By definition,

$$\begin{aligned} \forall \alpha, d \in \mathbb{R} \quad & \text{cost}(\alpha, d) = |\alpha d| + \lambda c_p(|(1 - \alpha)d|) \leq \\ & |\alpha^* d| + \lambda c_p(|(1 - \alpha^*)d|) = \text{cost}(\alpha^*, d) \end{aligned} \tag{10}$$

If we take $\alpha > 1$, with a positive and monotonically increasing function c_p

$$\begin{aligned} \forall \alpha > 1 \quad & \text{cost}(\alpha, d) = \alpha d + \lambda c_p(d\alpha - d) > \\ & d + \lambda c_p(0) = \text{cost}(1, d) \end{aligned} \tag{11}$$

Using Eqs. 10 and 11, we prove that approaching the target by more than d has always a greater cost than approaching just by a distance d, showing that $\alpha^* \leq 1$. On other words, it is always preferable to approach the target by d and sense from that position than approach by a bigger distance while increasing the perception distance as well, by sensing the target from further away. The same analysis can be done for $\alpha < 0$,

$$\forall\alpha < 0 \quad \mathrm{cost}(\alpha, d) = -\alpha d + \lambda c_p(d - \alpha d) > 0 + \lambda c_p(d) = \mathrm{cost}(0, d) \tag{12}$$

proving that $0 \leq \alpha^* \leq 1$. □

There is an intuitive explanation for the previous lemma. If $\alpha^* > 1$, then the robot would move pass the target and sense it from behind, increasing both motion and perception costs, which contradicts the initial assumptions. If that was the case, there would be other solutions where the robot could sense from the same distance, but with smaller cost of approaching the target, thus having an overall smaller cost. That solution would also contradict the first theorem, as there would be a point in the middle of the path with smaller distance to the target. The same reasoning can be applied when $\alpha^* < 0$, which also represents a counter-intuitive situation, because the robot would be moving away from the target, increasing both the motion and perception cost, and again contradicting the first theorem.

The previous results are expected if we use positive and increasing cost functions, showing that to sense a target in a straight line, the optimal solution is to approach the target while moving to a position in between the current node and the target. Moving beyond the target or moving back always yields solutions with a higher cost.

In order to find the solution for the straight line problem, we have to solve the minimization problem

$$\alpha^* = \underset{\alpha}{\mathrm{argmin}}\, \mathrm{cost}(\alpha, d) \tag{13}$$

for $0 \leq \alpha^* \leq 1$. This can be done by setting the gradient to zero to find extrema for the interior region, and if any point exists, comparing it to the cost values on the boundary to find the point with minimum cost (for $\alpha = 0$ the cost is $\lambda c_p(d)$ and for $\alpha = 1$ the cost is $\lambda c_p(0) + d$). If the function c_p has a gradient which is strictly increasing, then the overall cost is a convex function with only one minimum, and in that case the optimal solution is either $\alpha^* = 0 \vee \alpha^* = 1 \vee \alpha^* = \alpha_c$, where α_c is given by

$$\frac{d\big(|\alpha_c d| + \lambda c_p(|(1 - \alpha_c)d|)\big)}{\alpha_c} = 0 \tag{14}$$

After finding α^* as the optimal solution for the straight line problem, the heuristic in PA* is

$$h_{pp}(n, T) = \alpha^*||n - T|| + \lambda c_p\big((1 - \alpha^*)||n - T||\big) \tag{15}$$

and if we find the optimal approach point q^* such as $||q^* - T|| = \alpha^*||n - T||$ (where q^* has to be in the straight line distance between n and T), then the heuristic can also be defined as

$$h_{pp}(n, T) = ||n - q^*|| + \lambda c_p(||q^* - T||) \tag{16}$$

In order to use the straight line solution as the best heuristic for PA* when not considering obstacles, as in the common A* formulation, we need to prove that the straight line solution yields the smallest cost.

Considering the scenario in Fig. 3, the direct distance between robot position and target is d, as in the example before. However, because the approaching is not in a straight line with the target, $d' = ||n-q||+||q-T|| > d = ||n-T||$. Given the non-straight line assumption, $d' = d+\epsilon$, with $\epsilon \geq 0$. The robot moves a percentage of this path $\alpha d'$, and senses the rest $(1 - \alpha)d'$, where $\alpha = \frac{||n-q||}{||n-q||+||q-T||}$.

Theorem 2 *If the heuristic in PA* uses the straight line solution, the heuristic is admissible and consistent iff the perception cost function $c_p(d)$ is zero for $d = 0$.*

Proof The cost of the solution with $0 \leq \alpha \leq 1$ and $d' > 0$ is

$$\begin{aligned}\mathrm{cost}(\alpha, d') &= \alpha d' + \lambda c_p((1 - \alpha)d') \\ &= \alpha(d + \epsilon) + \lambda c_p((1 - \alpha)(d + \epsilon))\end{aligned} \tag{17}$$

Again, if the perception cost function is monotonically increasing

$$\mathrm{cost}(\alpha, d') = \alpha(d + \epsilon) + \lambda c_p((1 - \alpha)(d + \epsilon)) \geq \alpha d + \lambda c_p(d - \alpha d) \geq \mathrm{cost}(\alpha^*, d) \tag{18}$$

proving that a motion out of the straight line always has an higher cost than the optimal solution for the straight line, and as such the straight line solution can be used as an heuristic for the informed search in perception planning.

As we assumed that $c_p(0) = 0$, then it is trivial to show that $h(T, T) = 0$.

Furthermore, if we consider a successor node n' with an optimal approach point q'^* and a cost to move from n to n' as $c(n, n') = ||n - n'||$, then

$$c(n, n')+h(n', T) = ||n-n'||+||n'-q'^*||+\lambda c_p(||q'^*-T||) \tag{19}$$

Using first the geometric triangle inequality and then the definition of the heuristic,

$$\begin{aligned}c(n, n') + h(n', T) &\geq ||n - q'^*|| + \lambda c_p(||q'^* - T||) \\ &\geq ||n - q^*|| + \lambda c_p(||q^* - T||) = h(n, T)\end{aligned} \tag{20}$$

and because $h(T, T) = 0$, this heuristic is consistent. □

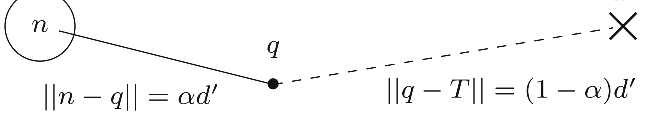


Fig. 3 Robot approaches target in a non-straight line to position q, with approach distance of $\alpha d'$, and the sensing distance of $(1 - \alpha)d'$, with $d' > d$, where d is the straight line distance without obstacles

We use $||.||$ as the euclidean distance. As we have showed here, the heuristic for PA*, h_{pp}, depends only on the distance to the target $d = ||n - T||$ and the parameter α^*, which depends on the perception cost function c_p and the distance d as well. Regarding the cost function $g(S, n)$, it is the same as standard A* is used, with g being the minimum cost for a robot to move from the initial position S to node n.

We consider the perception quality depends only on the distance to the target, because we assume robots can rotate in place and sense the target from the most favorable direction. Furthermore, the perception cost function c_p can be any monotonically increasing function, allowing flexibility to represent the cost of multiple perception models.

3.3 Perception Functions

For any specific perception cost, it is possible to find the optimal sensing position q^* and the parameter α^* as a function of the distance $||n - T||$ and the function c_p. And with α^* known before-hand, the heuristic h_{pp} becomes only a function of n and T, and easy to compute during search.

We give in this section two examples for the function c_p, where the perception cost is either a linear or quadratic function of the distance to the target. In our model we assume circular omnidirectional sensing, with a limited range r_p. We then formulate the heuristic $h_{pp}(n, T)$ in a way that it can be used for multiple perception cost functions, while being easily computed for each specific perception function.

First, we define the optimal sensing distance as $d_s^* = ||q^* - T|| = (1 - \alpha^*)||n - T||$, when $||n - T|| \rightarrow \infty$. This is useful when the overall cost functions are convex, with a perception cost whose gradient is strictly increasing, which is true for the functions we exemplify in this section. In that case, the optimal sensing distance is a constant, and a unique solution that only depends on the function c_p, not depending on the distance $||n - T||$. In this scenario, the real sensing distance used in the heuristic h_{pp} is either d_s^* or the boundary solution $||n - T||$ if $||n - T|| < d_s^*$.

In the linear case,

$$c_p(d) = \begin{cases} |d| & |d| \leq r_p \\ \infty & |d| > r_p \end{cases} \tag{21}$$

there are two cases for the optimal sensing distance d_s^*:

- $\lambda < 1$: Cost of motion is greater than cost of sensing, so robot minimizes motion by sensing form as far apart as possible (limited by maximum sensing range r_p);
- $\lambda \geq 1$: Cost of sensing is greater than cost of motion, so robot moves as close to the target as possible.

The optimal sensing distance d_s^* for linear perception is:

$$d_s^* = \begin{cases} r_p & \lambda < 1 \\ 0 & \lambda \geq 1 \end{cases} \tag{22}$$

With a quadratic sensing cost, $c_p(||q - T||) = ||q - T||^2$, we can solve for α^* using Eq. 14. In order to find the minimum, we find the point α_c with zero derivative

$$\frac{d\big(\alpha^* d + \lambda(d - \alpha^* d)^2\big)}{\alpha^*} = 0 \Leftrightarrow \alpha^* = 1 - \frac{1}{2d\lambda} \tag{23}$$

Therefore, ideally the robot would move to a fixed distance $1/(2\lambda)$ of the target to sense it optimally.

Again, the optimal sensing distance d_s^* for the quadratic perception cost function also depends on r_p:

$$d_s^* = \begin{cases} \frac{1}{2\lambda} & 1/(2\lambda) \leq r_p \\ r_p & 1/(2\lambda) > r_p \end{cases} \tag{24}$$

We can interpret the quadratic c_p as a function that represents a cost that changes little with close distances. When the robot is already close enough to the target, changes in the distance to the target have a small impact on the perception cost. On the other hand, the further away the robot is from the target, the greater the impact of the distance in the perception cost. Therefore, it is reasonable to think of a fixed optimal sensing distance in that case, which is a function of λ, the parameter that weights motion and perception cost.

Having the optimal sensing distance d_s^* and checking the boundary condition, the heuristic becomes:

$$h_{pp}(n, T) = \begin{cases} (||n - T|| - d_s^*) + \lambda c_p(d_s^*) & ||n - T|| \geq d_s^* \\ \lambda c_p(||n - T||) & ||n - T|| < d_s^* \end{cases} \tag{25}$$

This heuristic equation can also be used for other perception cost functions with similar properties (convex functions with unique solutions), with the only requirement of finding d_s^*.

The solution to the perception planning problems depends greatly on the sensor used. That is the reason we approached this problem in a more general perspective first, introducing a heuristic that works for any perception cost function, as long as it is an increasing function with distance. The specific cases shown in this section were only presented as examples of what needs to be done to determine the heuristics for specific cost functions. It is possible to adapt the cost functions to the problem in hand (e.g., sensor properties), and then determine the heuristic for that specific problem using our approach.

3.4 Underlying Graph

In the solution presented until now, we assume there is a grid-like discretization of the world which is used as the underlying graph. Every grid point is a node in the graph, and two nodes are connected if their respective grid positions are neighbors, assuming 8-connectivity as the adjacency rule. Due to the existence of obstacles between grid positions, not all of them are reachable by the robot.

As explained before, nodes are expanded using a heuristic that estimates how much the robot should approach the target in order to have an optimal path. If there were no obstacles in the PA* problem, the search would expand nodes until reaching the final grid position F. In that case, unless $F = T$, the heuristic for node F has an approaching distance equal to zero, i.e. $\alpha^* = 0$, which could be used as a stopping condition. However, in that case the heuristic in the final expanded state would be non-zero if $F \neq T$.

Furthermore, if we also consider the existence of obstacles, other problem arises. As the search progresses, assuming in current node n the optimal α^* is greater than zero, the algorithm has to continue search because the possibility of approaching the target yields a solution with lower cost. The meaning of this situation is that at the current node n the solution is not guaranteed to be optimal and there is the change to find another node with a lower overall cost for both motion and perception. As search progresses, the nodes that could in principle have a lower cost might be blocked by obstacles, or not have line of sight with the target.

Therefore, it is important that the algorithm can go back to the previous node n if it has the global lower cost after exploring other alternatives. While for this node n there will always be a positive optimal approaching distance, after exploring the rest of the graph and increasing the lower bound for the overall cost, this will be the node minimizing the cost as all the other possibly better alternatives have already been tested and are not valid final positions. However, the capability of backtracking to previous nodes is not possible in the graph described so far.

We thus propose an alternative extension to the original underlying graph that is built from the grid discretization, as shown in Fig. 4. On this extension one more node, T_e, is added to the original graph. The cost between nodes in the original graph is $c(n, n') = ||n - n'||$, and they stay the same in the extended graph. However, new connections are added between every node and the new node T_e such as their cost is $c(n, T_e) = \lambda c_p(||n - T_e||)$. The heuristic value of this node is zero, $h_{pp}(T_e, T) = 0$. Therefore, when expanding node n, not only its grid neighbors are added to the priority queue with the respective priority $f(n')$, but

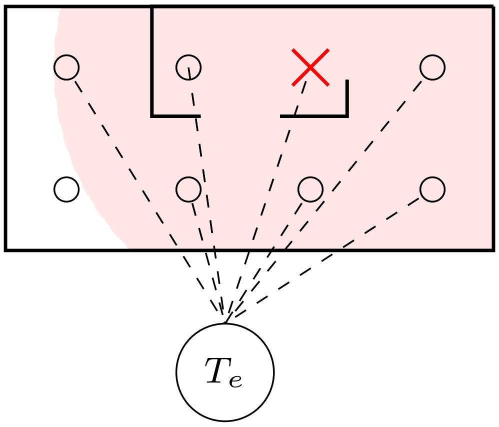


Fig. 4 Extending the underlying graph with an additional node T_e, which must be reached to finish search

also the node T_e is added with priority $f(T_e) = g(S, T_e) = g(S, n) + \lambda c_p(||n - T_e||)$, where n is the last grid position expanded.

3.5 Stopping Condition

Using the extended graph described before, the stopping condition becomes trivial. In order for search to stop with a feasible path for the robot, the node T_e has to be expanded. In that case, search stops after expanding a node with heuristic equal to zero. Moreover, until now obstacles which occlude the target have not been considered. So, when expanding the node T_e it is necessary to test with ray casting if there is line of sight between the last position of the path in the grid and the target. If the evaluation is successful, search stops because the optimal path has been found.

As there can be multiple entries in the priority queue for the node T_e which come from connections with different grid positions, it is necessary to memorize for each one what was the last grid position m visited. When the ray casting test results in a non-obstructed line of sight to the target, the position m becomes F, and the g values are used to find that optimal path from S to F, exactly as done in A* for motion planning.

Furthermore, it is possible that the ray casting fails and search has to continue. In that case, even though the heuristic is consistent, the node T_e can be expanded multiple times. All the other nodes from the original graph built from the grid can never be expanded more than once, and as such are added to a closed list.

Finally, in the new extended graph the heuristic in the *goal state* is always zero, so we can drop the constraint $c_p(0) = 0$ in order to have a consistent heuristic. The multiple expansion of T_e may seem contradictory to the fact

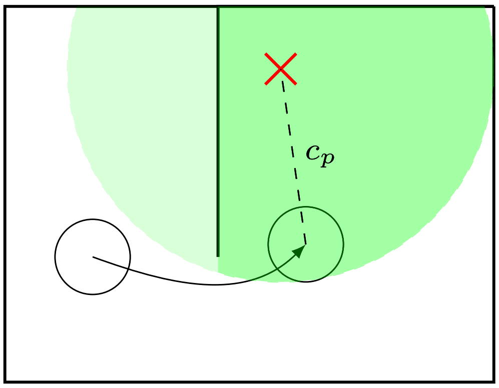


Fig. 5 Given a sensing target and a maximum perception range r_p, there is a set of candidate goal positions (green circle with radius r_p). From those positions, not all are feasible because obstacles can block line of sight to the target. The darker green represents feasible goal positions

that the heuristic is consistent, but can be explained as an *error* in the cost of connections to the state T_e. When ray casting fails from node n to T, the cost of the connection from node n to node T_e should have been infinity instead of $\lambda c_p(||n - T||)$. Therefore, this expansion of T_e should not have existed, and the first valid expansion will terminate search.

As we show in Fig. 5, there is candidate and feasible final path positions. For all the nodes outside of the circle, the connection to node T_e is infinity. For the grid positions inside the circle, the cost of the connection to T_e depends on the distance to the target. This is known and used when building the extended graph. Obviously, when the distance to the target is more than the maximum sensing range r_p, the connection to T_e is infinity and therefore unfeasible sensing distances are never considered during search.

The feasible positions are determined using ray casting. That operation could be done in the beginning too in order to have a correct cost in the connections to T_e (again, cost is infinity to non-feasible final path locations). However, ray casting is an expensive operation, so we only test for feasibility after expanding T_e, updating the cost to infinity

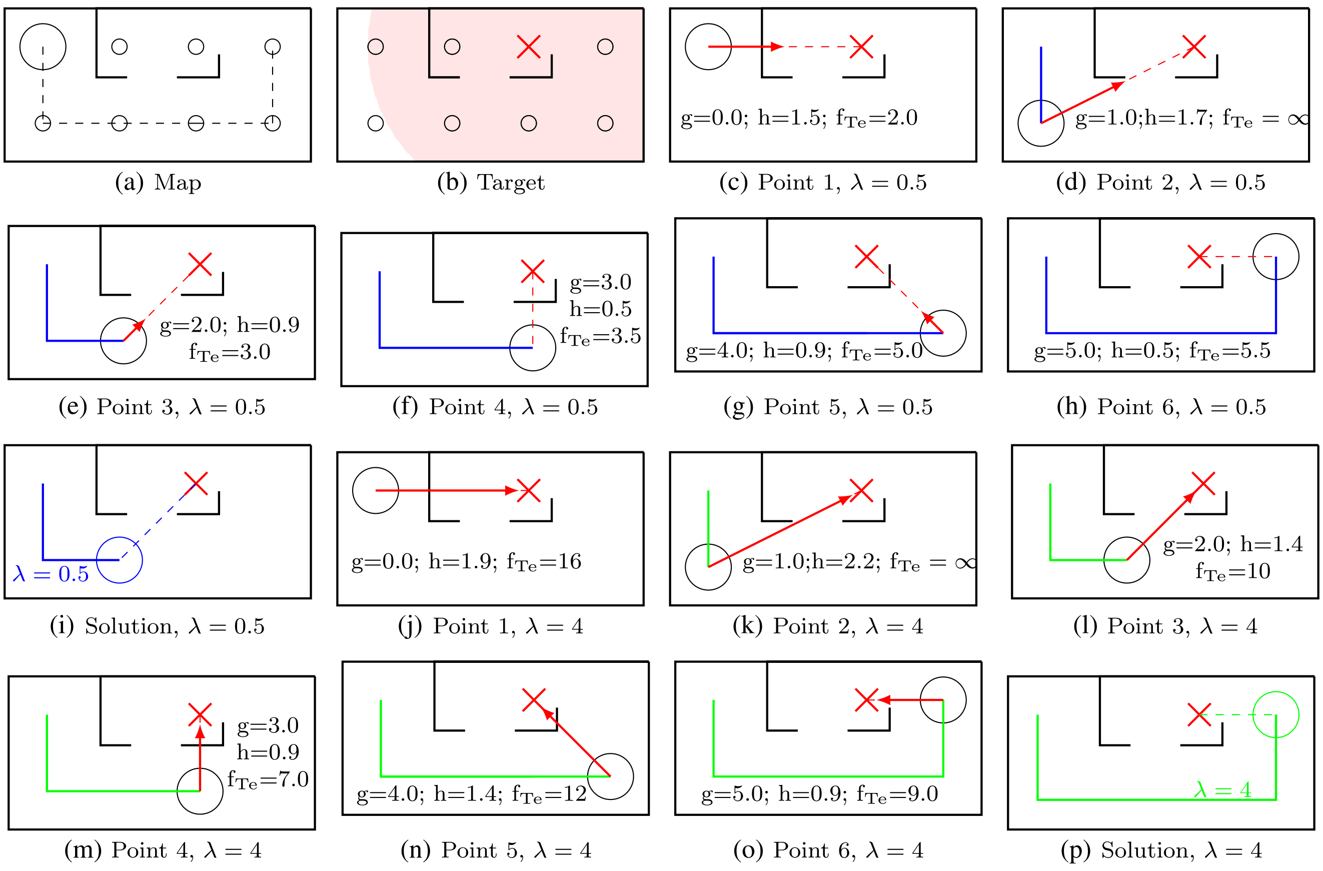

(a) Map
(b) Target
g=0.0; h=1.5; f$_{Te}$=2.0
(c) Point 1, $\lambda = 0.5$
g=1.0;h=1.7; f$_{Te}$ = ∞
(d) Point 2, $\lambda = 0.5$
g=2.0; h=0.9 f$_{Te}$=3.0
(e) Point 3, $\lambda = 0.5$
g=3.0 h=0.5 f$_{Te}$=3.5
(f) Point 4, $\lambda = 0.5$
g=4.0; h=0.9; f$_{Te}$=5.0
(g) Point 5, $\lambda = 0.5$
g=5.0; h=0.5; f$_{Te}$=5.5
(h) Point 6, $\lambda = 0.5$
$\lambda = 0.5$
(i) Solution, $\lambda = 0.5$
g=0.0; h=1.9; f$_{Te}$=16
(j) Point 1, $\lambda = 4$
g=1.0;h=2.2; f$_{Te}$ = ∞
(k) Point 2, $\lambda = 4$
g=2.0; h=1.4 f$_{Te}$=10
(l) Point 3, $\lambda = 4$
g=3.0 h=0.9 f$_{Te}$=7.0
(m) Point 4, $\lambda = 4$
g=4.0; h=1.4; f$_{Te}$=12
(n) Point 5, $\lambda = 4$
g=5.0; h=0.9; f$_{Te}$=9.0
(o) Point 6, $\lambda = 4$
$\lambda = 4$
(p) Solution, $\lambda = 4$

Fig. 6 PA* search for 2 values of λ and quadratic perception cost: map has 8 grid positions, from which 6 are reachable by the robot; connectivity between feasible robot positions is represented with dashed black lines; the maximum sensing range is shown with the red circle around the target (red cross), so all grids points except one are close enough to perceive the target; each grid point, which is a graph node, has a $g(S, n)$ cost and a heuristic estimate $h(n, T)$; each grid point is connect to the node T_e of the extended graph; the priority value of T_e with a connection to each point shown as f_{T_e} in the respective image; red arrows represent the optimal approach distance for the heuristic in each point and the red dashed line the optimal perception distance in the heuristic for each point

in case ray casting fails during search, meaning there is no line of sight from that position to the target.

In conclusion, the stopping criteria can be stated as reaching node T_e (with a connection from node n) when ray casting from n to T returns a feasible line of sight from grid position n to the target, with n becoming the final path position F.

3.6 Example with Multiple Feasible Sensing Positions

We show in Fig. 6 two PA* searches with different values for the trade-off parameter λ. In Fig. 6a we have a simple environment where black lines represent obstacles. We assume the robot can only be in 8 positions (represented with circles, where the big circle in the upper left corner is the robot in its initial position). The vertical and horizontal distance between grid points is 1 unit. From those 8 points only 6 are reachable, which are connected to the initial position with the dashed lines. In Fig. 6b the target position is shown with a red cross, and the red region represents a circle with radius r_p, the maximum sensing range. Therefore, all the points inside the red region could perceive the target if there were no obstacles. The bottom left grid point is the only position from where the target cannot be sensed.

The perception function is quadratic, with $c_p(||n-T||) = ||n - T||^2$ when $||n - T|| < r_p$, and infinity otherwise. In the first example, the weight parameter $\lambda = 0.5$, so the optimal sensing distance is $d_s^* = 1$. For each point the heuristic cost represents the minimal weighted sum of an approach distance and the cost of a perception distance. The approach distance, $\alpha^* d$, is represented with a red arrow, and the perception distance for the heuristic, $(1 - \alpha^*)d$, is represented with a red dashed line.

The reachable points are all connected in a line, and as shown in Fig. 6 they are Point 1 to Point 6, which are nodes n_1 to n_6 in the graph representation. In order to find the optimal path from n_1 to perceive the target, the algorithm starts by expanding n_1 in Fig. 6c. This node is not the goal because search only stops after expanding the node T_e in the extended graph. The successors of n_1 are node n_2 with priority $f_2 = 2.7$ $(= g_2 + h_2 = 1 + 1.7)$, as shown in Fig. 6d, and node T_e with priority $f_{T_e} = 2$, as shown in Fig. 6c. These two nodes are added to the priority queue. Because T_e has lower priority, it is the next expanded node, and the stopping criteria is tested with ray casting, but fails because there is no line of sight to the target from n_1 as shown in Fig. 6c.

In the next iteration the only node in the priority queue is n_2, which is expanded. The next node added to the priority queue is n_3 with priority $f_3 = g_3 + h_3 = 2 + 0.9 = 2.9$, because it is the only grid neighbor to n_2 that was not expanded before. The connection to T_e from n_2 has infinite cost because it is beyond the maximum sensing range to the target. As such, T_e is not added to the priority queue this time.

The next and only node in the priority queue is n_3, shown in Fig. 6e, which adds T_e to the priority queue again, now with priority (3.0). In this iteration, the node n_4 is also added with priority (3.5). There is now 2 nodes in the priority queue, and the first expanded is again T_e because it has the lowest priority. Using ray casting, the algorithm finds that n_3 has no obstructions and can perceive the target. Therefore, the stopping criteria is met and search stops. Node n_3 is the final position F, and the optimal path to perceive this target with $\lambda = 0.5$ is $\rho = \{n_1, n_2, n_3\}$, as shown in Fig. 6i, with cost of 3 units. Due to the specific parameters used, nodes n_4 to n_6 shown in Fig. 6f, g and h are never expanded.

As a side note, it is possible to see in Fig. 6f and h that the heuristic distance for perception takes the whole distance from the node to the target, and as such $f = g + h = f_{T_e}$.

In the second example, $\lambda = 4$, making the perception cost bigger in comparison to the motion cost, and as such it is expected for the optimal solution to move closer to the target in order to reduce the perception distance. Again, in order to find the optimal path from n_1 to perceive the target, the algorithm starts by expanding n_1 in Fig. 6j. This node has successors node n_2 with priority $f_2 = 3.2$ $(= g_2 + h_2 = 1 + 2.2)$, as shown in Fig. 6k, and node T_e with priority $f_{T_e} = 16$, as shown in Fig. 6j. These two nodes are added two the priority queue.

Because n_2 has lower priority, it is the next expanded node. The next node added to the priority queue is n_3 with priority $f_3 = g_3 + h_3 = 2 + 1.4 = 3.4$, because it is the only grid neighbor to n_2 that was not expanded before. The connection to T_e from n_2 has infinite cost again, not being added to the priority queue.

The next node in the priority queue with lowest priority is n_3, shown in Fig. 6l, which is expanded and adds T_e to the priority queue again, now with priority 10. In this iteration, the node n_4, the only not expanded neighbor, is also added with priority 3.9. There is now 3 nodes in the priority queue, n_4 with priority 3.9 and T_e twice with priorities 16 and 10. The next expanded is n_4, shown in Fig. 6m, which adds again T_e with priority 7.0, and n_5 with priority 5.4, as shown in Fig. 6n.

In the next iteration n_5 is expanded, adding T_e to the priority queue with priority 12 and node n_6 with priority $f_6 = 5.0 + 0.9 = 5.9$. Again all the nodes T_e added to the priority queue have higher cost, so the next expanded node is n_6, from Fig. 6o, and T_e is added one last time, with priority 9. At this point the priority queue has 5 nodes, all of them being T_e, with priorities 7, 9, 10, 12 and 16. The first expanded is T_e with a connection from n_4, shown in

Fig. 6m, but the stopping criteria fails with ray casting being blocked by obstacles. Finally, the node T_e with connection from n_6 and priority 9 is expanded, and search stops as n_6 has line of sight with the target.

Having met the stopping criteria, node n_6 is the final position F, and the optimal path to perceive this target with $\lambda = 4$ is $\rho = \{n_1, n_2, n_3, n_4, n_5, n_6\}$, as shown in Fig. 6p, with cost of 9 units. As we can see, the priority of expanded nodes always increases, which is a characteristic of consistent heuristics.

For $\lambda = 3$, both solutions with paths stopping in n_3 and n_6 have the same cost, with 8 units.

4 Approximate Visibility Maps

In this section we will summarize a technique for effective determination of the observability capabilities of robots in 2D gridmaps. We show how to use morphological operations in order to build approximate Visibility Maps [16]. These maps can then be used to obtain information that, when considered in the heuristic of perception planning, can improve the search efficiency.

We assume robots have a circular shape, and a maximum sensing range, r_p. The goal of visibility maps is to efficiently determine the observability of a robot in a certain environment, i.e., determine what regions can be sensed from any point that is reachable from the initial robot position. The algorithm is a function of robot size and sensing range. We show in Fig. 7 a simulated environment with obstacles, and the resulting Approximate Visibility Map (A-VM).

Robots move along the environment, which we assume is represented as a gridmap. Given the robot's geometric properties, there will be some regions that are accessible to the robot, and some regions that will be unaccessible. Furthermore, robots can use their sensors to perceive inside unaccessible regions.

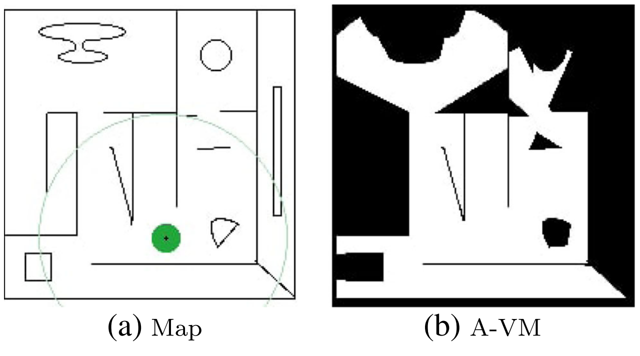


Fig. 7 Given a black and white gridmap, an omnidirectional circular robot (green), and a sensing range (green circumference), the A-VM determines what can be sensed from reachable positions

Considering maps are discrete representations of the environment, there is a duality between images and maps, because both of them are a discrete sampling of the world. Our algorithm performs purely geometric operations on the map in order to obtain all the regions that can be sensed by the robot, using image oriented techniques. The first step of our algorithm is to transform an occupancy grid map (with probabilities of occupation in each cell) into a binary map of free and obstacle cells, using a threshold. The occupancy grid can be obtained through SLAM methods.

In order to determine the robot-dependent map, we use the partial morphological closing operation, which can be applied on images using a structuring element with a given shape to represent the robot. Here the structuring element is a circle representing a circular robot. The domain is a grid of positions G. The input is a black and white binary image representing the map (Fig. 8a), with M being the set of obstacle positions, and with every pixel corresponding to a node in the underlying graph. The structuring element, R, represents the robot. The morphological operation dilation on the obstacle set M by R is

$$M \oplus R = \bigcup_{z \in R} M_z \tag{26}$$

where $M_z = \{p \in G \mid p = m + z, m \in M\}$, i.e., the translation of M by z over the grid G.

Applying the dilation operation to the obstacles in the map (black pixels in the map image), the algorithm inflates the obstacles by the robot radius, which can be used to find the free configuration space in Fig. 8b.

$$C^{free} = \{p \in G \mid p \notin M \oplus R\} \tag{27}$$

The same approach is used in most motion planning algorithms, with obstacle inflation being used to find the configuration space and do path planning while avoiding collisions. Indeed, inflation is the solution used, for example, in the ROS navigation package. [11].

The configuration space shows where the robot center can be, representing the feasible positions for the robot center, but not giving any information of what regions can be actuated or sensed by the robot.

From C^{free}, it is possible to find the points in the free configuration space that are reachable from the initial robot position S, which belong to the reachable set $Reach(S)$.

$$Reach(S) = \{p \in C^{free} \mid p \text{ connected to } S\} \tag{28}$$

The reachability set $Reach(S)$, in Fig. 8d, is the set of points in the free configuration space that are connected to the initial position S, i.e., there is always a path between points in $Reach(S)$ and S through adjacent cells in the free configuration space. The reachable space can easily be obtained using the image processing technique *labeling*.

The partial morphological closing is obtained by applying the second morphological operation (erosion) to a subset of the original dilated image. Thus here the erosion is applied to the complement of the reachable set $Reach(S)$ instead of being applied to the overall image with dilated obstacles $M \oplus R$. Because dilation and erosion are dual operations, the partial morphological closing was computed by dilating the reachable set instead. With this operation only positions reachable from the starting position are considered.

$$A(S) = Reach(S) \oplus R \tag{29}$$

Using the partial morphological operation we determine the actuation space, $A(S)$, which represents the regions a circular robot can touch with its body, given its radius and an initial position.

If we think of a robot that actuates in the world, applying the partial morphologic operation to an image map results in the *Actuation Map*, in Fig. 8e, which represents the regions of the environment that can be actuated by a robot, assuming the actuation radius is equal to the robot size. If the actuation range is smaller than the robot radius, this approach can still be used to determine the actuation map $A(S)$ by using a smaller circle for the second dilation operation instead of R. The regions outside the actuation space, $U(S)$, cannot be reached by the robot body, and thus cannot be actuated.

$$U(S) = \{p \in G \mid p \notin A(S) \wedge p \notin M\} \tag{30}$$

As an example, we can consider a vacuum cleaning robot. The configuration space represents the possible center positions, and $A(S)$ represents the regions the robot can clean. Finally, the unreachable regions are the parts of the environment the robot cannot clean. The unreachable regions might result from the circular shape of the robot, that makes it impossible for it to clean corners, as shown in Fig. 8.

Furthermore, if we consider the problem of perception, a *Visibility Map* represents the regions of the input map that are visible by the robot from some reachable position. The sensing range is expected to be bigger than the robot radius.

The easier approach to solve the visibility problem for robots in a certain environment is to use a brute-force algorithm. For each position in the map to be tested for visibility, we only need to find a feasible robot position in $Reach(S)$ that has line of sight to the point being tested. For that purpose we use the ray casting technique. This approach is however very inefficient. If the robot moves in an environment with unexpected and dynamic obstacles, the visibility map needs to be updated frequently. Therefore, using the brute-force would lead to high idle times.

4.1 Visibility from Critical Points

As an alternative to using the brute-force approach, we can take the actuation space $A(S)$ and consider it as a first approximation of the Visibility Map, because it shows the visible points assuming a sensing radius equal to or smaller than the robot size. However, in order to have a transformation from the input image map to the complete Visibility Map, we have to consider sensing beyond the robot radius.

The Approximate Visibility Map is built incrementally from the first approximation given by $A(S)$. The unreachable regions $U(S)$ are divided in a set of different disconnected components $U^l(S)$, which is useful because it allows to determine additional visibility inside each one independently. Each region $U^l(S)$ has its unique openings to the actuation space, from where visibility inside $U^l(S)$ is possible. These openings are the frontiers, defined as the points of the unreachable space that are adjacent to $A(S)$, as shown in Fig. 9a.

$$F^l(S) = \{p \in U^l(S) \mid \exists p' : p' \text{ is adj. to } p \wedge p' \in A(S)\}. \tag{31}$$

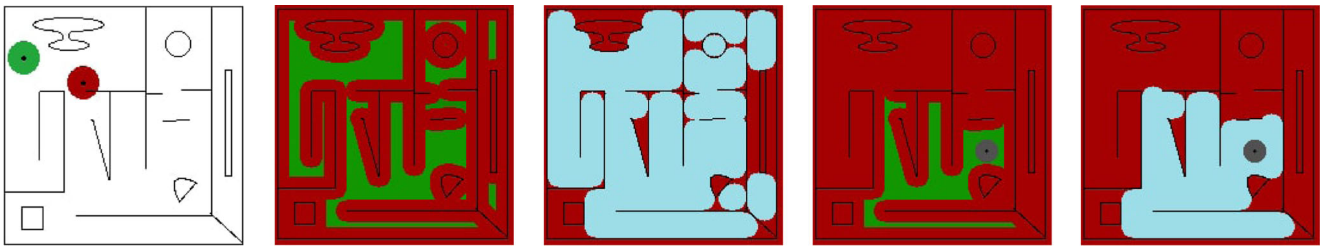


Fig. 8 In (**a**) two possible positions for the robot are shown, the green one being feasible, while in the red the robot overlaps with obstacles. These positions correspond to green and red points in the configuration space in (**b**), obtained by application of the morphological operation dilation to the map (C^{free} is set of green regions). The morphological closing is shown in (**c**). From the configuration space, the connected parts to the initial position (gray robot) are determined, which results in the reachable space presented in (**d**). If the second dilation operation is only applied to the reachable space instead of all C^{free}, applying the partial morphological operation to the reachable space, it is possible to determine the *actuation space*, in (**e**)

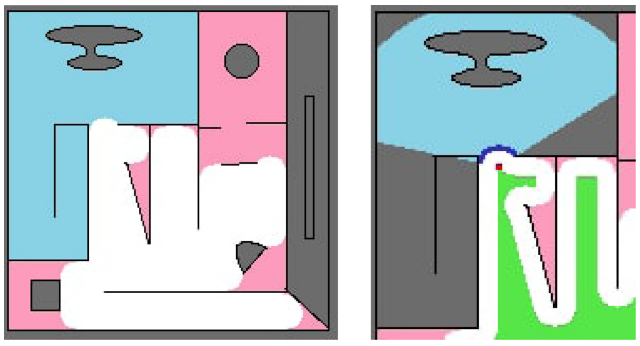

(a) Unreachable Regions (b) Critical Point

Fig. 9 In **a** we show $A(S)$ in white, in pink the unreachable regions that connect with $A(S)$, and in blue an example of a disconnected unreachable region $U^l(S)$. In **b** we highlight that disconnected region, showing in dark blue the frontier segment points $F^{li}(S)$, and in red the critical point $c^*_{li}(S)$, and the expected visibility $V_e^{cli}(x)$ from a critical point is shown in light blue, estimating visibility inside an unreachable region through a frontier. Gray points of the unreachable region cannot be visible from the critical point. The points from the Reachable Space are shown in green

The frontier can be composed of disjoin segments $F^{li}(S)$, and visibility inside the unreachable region should be determined for each segment independently.

When determining visibility for a sensing range greater than robot size, the additional visibility always comes from points inside of $U^l(S)$ with line of sight through $F^{li}(S)$. Every viable line of sight inside the unreachable regions passes through points in the frontier. Therefore, regions without frontiers do not need to be checked, because they are disconnected from the observable world by walls or obstacles. Checking for the existence of frontiers reduces the search of visibility only to the sensing-accessible parts of the world.

There are multiple candidate positions that can sense inside $U^l(S)$, and all of them have to be in $Reach(S)$, the feasible positions for the robot center. In order to have the true visibility map, all points from $Reach(S)$ should be considered.

The complete solution is computational expensive, so an alternative was proposed, where the visibility inside unreachable regions through each frontier segment is considered only from one point of the reachable space.

As only one point is being used, the final visibility map is an approximation of the ground-truth. In order to obtain a good approximation, the point chosen has to maximize the expected visibility inside the unreachable region. Given a point p and a frontier $F^{li}(S)$ it is possible to determine an expected visibility inside $U^l(S)$, $V_e^{pli}(S)$, as the area of an annulus sector defined by the robot size, sensing radius, and the frontier extremes.

In order to maximize the expected utility, a point closer to the frontier is chosen, which is equivalent to having a deeper and wider expected visibility inside $U^l(S)$. For this ideal point the sum of the distances to all frontier points is minimized:

$$c^*_{li}(S) = \underset{p \in Reach(S)}{\operatorname{argmin}} \sum_{\zeta \in F^{li}(S)} \|p - \zeta\|^2 \tag{32}$$

where $c^*_{li}(S)$ is called a *critical point*, which is explained in Fig. 9b. For each frontier $F^{li}(S)$ there is one critical point. If we determined visibility inside unreachable regions from all the points in the reachable space, we would obtain the true visibility map.

In order to deal with occlusions, we determine the true visibility of $V_e^{cli}(S)$ from critical points $c^*_{li}(S)$ using ray casting, considering a maximum sensing range.

Although the method is not efficient to determine the full visibility, here we only calculate visibility from a small set of points, with most having a high probability of being visible.

By transforming the original large-scale brute-force task in a set of small problems solved only using the critical points, it is possible to use ray casting more efficiently to determine visibility.

From each pair of frontier $F^{li}(S)$ and critical point $c^*_{li}(S)$, the algorithm determines the expected visibility $V_e^{cli}(S)$. Then, in order to determine which points are part of the visible regions from the critical point, we consider occlusions using ray casting.

The algorithm determines the true visibility from the critical point and through the corresponding frontier inside an unreachable regions. Those points of true visibility define the set $V_t^{cli}(S) \subseteq V_e^{cli}(S)$.

The additional visibility $V_t^{cli}(S)$ corresponds to the points inside $U^l(S)$ visible from the critical point $c^*_{li}(S)$

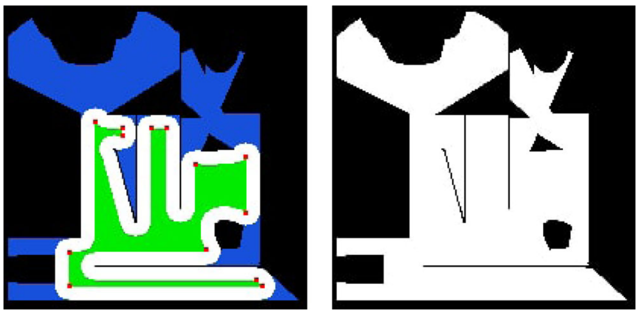

(a) Visibilities From Critical Points (b) Visibility Map

Fig. 10 In this figures the green region is the reachable set, $Reach(S)$. Critical points are represented as red dots. All the critical points and respective extended visibility regions are presented in (**a**). The final visibility map, shown in (**b**) is the union of the *actuation space* and the extended visibility from all critical points

through the frontier $F^{li}(S)$. Finally, the Approximate Visibility Map is given by

$$V(S) = A(S) \bigcup_{li} V_t^{cli}(S) \tag{33}$$

After analyzing unreachable regions $U^l(S)$ independently, we were able to determine the visibility inside each one. The overall visibility for the all map is then simply given by the union of the *actuation space* with the individual visibilities $V_t^{cli}(S)$ obtained for each region $U^l(S)$, as shown in Fig. 10. We show the overall technique in Algorithm 1.

Algorithm 1 Approximate Visibility Map: Creating visibility maps from grid maps

```
Require: Occupancy Map $M_{occ}$, robot initial position $S$, shape $R$, maximum sensing range $r_p$
 1: $M \leftarrow$ im2bw($M_{occ}$)    ▷ b&w image from gridmap
 2: $C^{free} \leftarrow$ dilation($M, R$)    ▷ inflation of obstacles
 3: $Reach(S) \leftarrow$ labeling($C^{free}, S$)    ▷ reachable space from $S$
 4: $A(S) \leftarrow$ dilation($Reach(S), R$)    ▷ partial morph. closing
 5: $V(S) \leftarrow A(S)$    ▷ visibility initialized with $A(S)$
 6: $U(S) \leftarrow$ unreachable($A(S), M$)    ▷ unreachable regions
 7: $\{U^l(S)\} \leftarrow$ labeling($U(S)$)    ▷ find disconnected regions
 8: for each $U^l(S)$ do
 9:     $F^l(S) \leftarrow$ frontier($U^l(S), M$)    ▷ find frontiers
10:     $\{F^{li}(S)\} \leftarrow$ clustering($F^l(S)$)    ▷ disconnected frontiers
11:     for each $F^{li}(S)$ do
12:         $c_{li}^*(S) = \underset{z \in Reach}{\text{argmin}} \sum_{\zeta \in F^{li}} \|z - \zeta\|^2$    ▷ critical point
13:         $V_e^{cli}(S) \leftarrow$ expected_visibility($U^{li}, F^{li}, c_{li}^*, r_p$)
14:         $V_t^{cli}(S) \leftarrow$ ray_casting($V_e^{cli}(S), c_{li}^*(S)$)
15:         $V(S) \leftarrow V(S) \bigcup V_t^{cli}(S)$
16:     end for
17: end for
18: Return $V$
```

5 Perception Planning with A-VM

In this section we show that with an initial fixed cost of building the Approximate Visibility Map (A-VM), it is possible to use the critical points from the A-VM to improve the search heuristic of PA* [14].

The visibility map gives information on the feasibility of perception, while not giving any information about the positions from where targets can be perceived. Nevertheless, the transformation provides structured information about the environment, and it is possible to separate grid points into three categories:

1. **Reachable Space**: points that can be reached by the robot center, $Reach(S)$;
2. **Actuation Space**: points that can be “touched” by the robot body, $A(S)$;
3. **Unreachable Regions**: points the robot cannot cover with its body and motion only, because they lie in positions not traversable by the robot, $U(S)$.

The second category is a superset of the first. While there is no perception information for targets in the first category, it is possible to gain information about points in the second category only, i.e., $T \in$ (Actuation Space) $\setminus$ (Reachable Space). We know these targets T have a distance to the reachable set not bigger than the size of the robot. Therefore, with a small search bounded by the robot size, it is possible to find the closest point p in the reachable set minimizing distance to the target T. The distance $||p - T||$ is a lower bound for the perception distance. However, in this specific case the gained information will probably have only negligible effects on the search efficiency.

The third category of points, those that belong to the unreachable regions, is the category with the greatest benefits in terms of information gain from the Visibility Map transformation. Targets in the unreachable regions have associated a critical point, which gives information about a possible position from where they can be sensed. Furthermore, the distance between a target in region $U^l(S)$ and a critical point $c_{li}^*(S)$ can be used as a better estimate of the perception distance in the heuristic for perception planning. Therefore, we will focus our discussion only to points that belong to the third category, *Unreachable Regions*.

Considering the base heuristic of PA*, independently of the perception cost function, we know it is associated with the cost of approaching the target in order to sense it from a better sensing position, reducing the perception cost. Moreover, the heuristic does not consider obstacles, and the best sensing position lies in the straight line between the current node n and the target T. We assume we can solve the heuristic minimization problem (Eq. 7) for a specific cost function c_p, and find the optimal sensing distance d_s^*. Assuming $0 \le d_s^* \le ||n - T||$, the heuristic can be given as

$$h_{pp}(n, T) = ||n - T|| - d_s^* + \lambda c_p(d_s^*) \tag{34}$$

The visibility map gives information about the minimum sensing distance from any point in the reachable space to a point in the unreachable region, which can be used in the heuristic instead of the optimal sensing distance d_s^* given by the straight line solution, as shown in Fig. 11. When using the Approximate Visibility Map, and being the

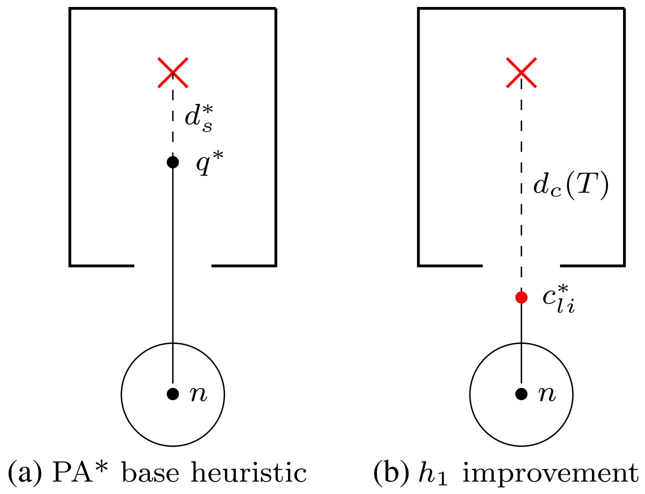


Fig. 11 Impact of using the distance to the critical point, $d_c(T)$, as the guaranteed minimum perception distance on the improved heuristic h_1

distance from critical point to $T \in U^l(S)$ $(\equiv T^l)$ equal to $d_{li}^c(T) = ||T - c_{li}^*(S)||$, the heuristic becomes:

$$h^1(n, T^l) = \begin{cases} ||n - T^l|| - d_l^c(T^l) + \lambda c_p(d_l^c(T^l)) \\ \qquad\qquad\qquad \forall ||n - T^l|| \geq d_l^c \\ \lambda c_p\left(d_l^c(T^l)\right) \quad ||n - T^l|| < d_l^c \end{cases} \tag{35}$$

where $d_l^c(T^l) = \min_i d_{li}^c(T^l)$. This applies to the case $d_l^c(T^l) \geq d_s^*$, otherwise the original heuristic h_{pp} is used. Here we are not considering the possibility that d_l^c is bigger than r_p, but in that case the target would not be visible. In order to use this heuristic as admissible and guarantee an optimal path, we only need to prove the distance from unreachable T^l to any other point in the reachable space is bigger than $d_l^c(T^l)$.

In the unreachable regions, the minimum sensing distance is the smallest distance from the target to the corresponding critical point. Again, that distance can be used as the minimum sensing distance in the PA* heuristic to improve the search speed.

Theorem 3 *Distance of points inside unreachable regions to the critical point is minimal in comparison to distance to any other point in the Reachable Space.*

Proof We assume only one critical point and frontier, for sake of simplicity. As shown in Fig. 12, we consider only the frontier extremes, the two obstacles at points $O_1(0, -\zeta)$ and $O_2(0, \zeta)$, with $\zeta < R$, being R the robot radius. The frontier is between those two obstacles. If the robot starts at some point with $x > 0$, then the unreachable region consists of points with $x \leq 0$. If there were other obstacles besides the points O_1 and O_2, there might be unreachable points with $x > 0$, but those are not relevant to this proof, and as such we kept the minimum number of obstacles.

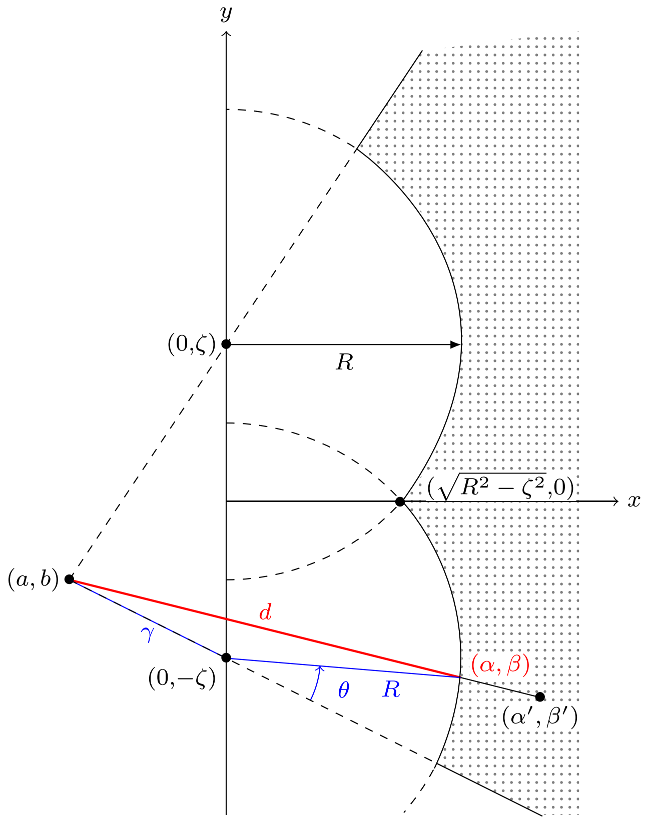


Fig. 12 Given two obstacles at positions $(0, \zeta)$ and $(0, -\zeta)$, the set of points in the reachable space that can sense the point (a, b) is represented with the filled region; the critical point $(\sqrt{R^2 - \zeta^2}, 0)$ is the point with the minimum distance to any (a, b) in the unreachable region

Using the same reasoning, only points of the reachable space with $x > 0$ are interesting because those are the only ones that can be used to have visibility inside the unreachable region.

Following this description, the critical point results as the point that is at R distance from both obstacles, $(\sqrt{R^2 - \zeta^2}, 0)$. For any point (a, b), with $a < 0$, the distance to the critical point has to be the minimum distance between (a, b) and any point in the reachable space, (α', β'), with $\alpha' > 0$. As we can see in Fig. 12, for any point (α', β') there is a point (α, β) in the border of reachability that has lower distance to (a, b). And the distance between (a, b) and (α, β) is given by

$$\begin{aligned} d^2 &= (\gamma + R\cos\theta)^2 + (R\sin\theta)^2 \\ &= \gamma^2 + R^2\cos^2\theta + 2\gamma R\cos\theta + R^2\sin^2\theta \\ &= \gamma^2 + R^2 + 2\gamma R\cos\theta \end{aligned} \tag{36}$$

As we can see from the equation, the distance is minimized when increasing the angle θ, and the angle θ is maximized at the critical point. Thus, we prove the distance to any unreachable target is minimized by the critical point. □

Only minimal errors exist due to discretization. While in the continuum space there is only one point that minimizes the distance to all frontiers, in the gridmap, there might be two points that minimize the distance, with the same cost. If we take into account the discretization error when determining the minimal distance, then it is possible to use the distance to the critical point to still obtain an admissible heuristic. That can be accomplished by subtracting the quantization error from the distance to the critical point.

With heuristic h_1 we can also use the distance to the critical point, d^c, for the stopping condition. Given that d^c is proved to be minimal, we can change the cost of connections to node T_e if their distance is less than the critical point distance to the target, reducing the number of points to be tested with ray casting.

$$c(n, T_e^l) = \infty \quad , \quad ||n - T^l|| < d_l^c(T^l) \tag{37}$$

However, it is still possible to improve the proposed heuristic. Instead of using the critical point to have only a lower bound estimate on the perception distance, we can use it to estimate a lower bound for motion cost as well, as shown in Fig. 13.

At first we assume the optimal sensing distance d_s^* is lower than the distance to the critical point, d_{li}^c. Thus, from any position with line of sight to the target (robot at point $x > 0$ in Fig. 14), the robot will move to a point as close as possible to the unreachable target, i.e., a border of the reachable space. Therefore, we know that the minimum motion cost will be the distance between the current node n and the closest point in the border of the Reachable Space. From Fig. 14, we can see that in the worst case scenario, the distance between the critical point and any other point of the Reachable Space border, with line of sight to the target, is $2R$. Thus, the new admissible heuristic becomes:

$$h^2(n, T^l) = \min_i \Big(\max(||n - c_{li}^*(S)|| - 2R, 0) + \lambda c_p(d_{li}^c(T^l)) \Big) \tag{38}$$

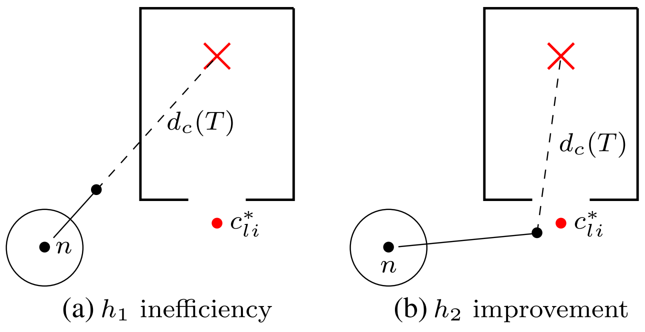


Fig. 13 Impact of using the critical point location for a better estimate of the motion cost on the improved heuristic h_2

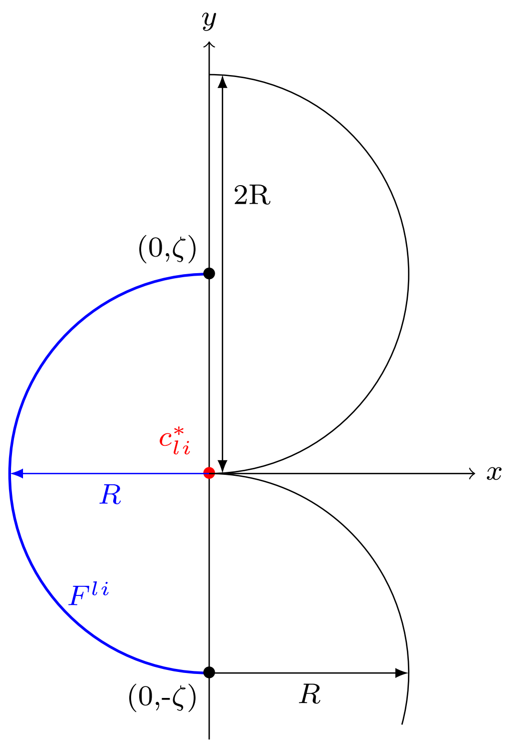


Fig. 14 Worst case scenario for the distance between critical point and the border of reachable space (the two half circumferences with $x > 0$), from Fig. 12: for the worst case, the distance between obstacles points is exactly the diameter of the robot, and the further point in the border of the reachable space is at distance $2R$ from the critical point

We can also update the heuristic to consider the case $d_s^* > d_{li}^c(T^l)$, using $\delta = \max(d_s^* - d_{li}^c(T^l), 0)$.

$$h^2(n, T^l) = \min_i \Big(\max(||n - c_{li}^*(S)|| - 2R - \delta, 0) + \lambda c_p(d_{li}^c(T^l)) \Big) \tag{39}$$

Finally, the first heuristic h^1 might be a better estimate in cases there is line of sight between n and T, so in order to always use the best heuristic, we choose the one closest to the true value, considering they are both admissible.

$$h^{AVM}(n, T^l) = \max(h^1(n, T^l), h^2(n, T^l)) \tag{40}$$

Theorem 4 *Heuristic using Approximate Visibility Map dominates original heuristic in PA*.*

Proof The original heuristic $h_{pp}(n, T)$ in PA* is always less than the real cost, because it uses the optimal solution for the euclidean distance without any obstacles, assuming optimal motion and perception distances. The heuristics using the A-VM replace the perception and motion costs by better estimates, but still underestimating the real cost as shown in Figs. 12 and 14 and Theorem 3. Thus the estimates $h^1(n,T)$ and $h^2(n,T)$ are always greater or equal than $h_{pp}(n,T)$, because we proved the sensing distance to the critical point is the minimal perception distance. Moreover, if both $h^1(n,T)$ and $h^2(n,T)$ are admissible heuristics, the maximum operation keeps that property. Therefore, $h_{pp}(n,T) \leq h^{AVM}(n,T)$. And because $h^{AVM}(n,T)$ is admissible, it is also dominant over $h_{pp}(n,T)$. $\square$

Like we did in Eq. 37, we can use the added information of the critical point location to update the cost of connecting nodes to T_e, filtering the clearly unfeasible positions. That allows not only to reduce the size of the priority queue that manages the PA* search, but also the number of ray casting operations. Nodes n from where perception is not feasible are updated such as $c(n, T_e) = \infty$.

Given a target T^l in an unreachable region, we consider the distance from the target to the critical points, $d^c_{li}(T^l)$. We know that frontiers F^{li} are generated by the robot body, which is a circle with radius R. Therefore, the distance between frontier points and the critical point c^*_{li} is R. Using this information, we can determine an annulus sector from the target to the possible frontier points with a distance R around the critical point, and guarantee that any feasible position with line of sight to the target has to be in that annulus sector. So, determining the maximum possible angle range between target and frontier points allows us to filter the feasible points for perception.

Assuming a critical point c^*_{li} and target aligned with the x axis, as in Fig. 15, the maximum angle to the frontier is given by:

$$\theta^f_{li}(T^l) = \max_{\phi} \operatorname{atan}\left(\frac{R\sin(\phi)}{d^c_{li}(T^l) - R\cos(\phi)}\right) \tag{41}$$

Because $d^c_{li}(T^l) > R$, we can solve the equation and find the optimal ϕ^*

$$\cos(\phi^*) = \frac{R}{d^c_{li}(T^l)} \tag{42}$$

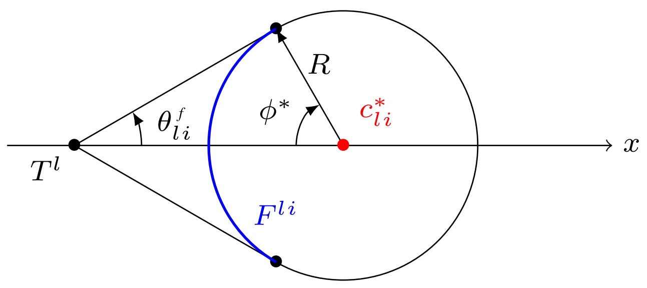


Fig. 15 Determining the maximum angle range from the target to the frontier points, considering the distance to the critical point and the robot size R

and the maximum angle to the frontier θ^f becomes

$$\theta^f_{li}(T^l) = \operatorname{atan}\left(\frac{R}{\sqrt{\left(d^c_{li}(T^l)\right)^2 - R^2}}\right) \tag{43}$$

Then, we are able to filter the feasible nodes n for perception of the target using the angle between n and the target T^l, θ^n, and the angle between the target and the critical points c^*_{li}, θ^c_{li}.

$$c(n, T_e) = \begin{cases} \lambda c_p(\|n - T^l\|) & \|n - T^l\| < d^c_l(T^l) \wedge \\ & \exists i : \theta^c_{li} - \theta^f_{li} \leq \theta^n \leq \theta^c_{li} + \theta^f_{li} \\ \infty & \text{otherwise} \end{cases} \tag{44}$$

6 Results

In this section we present several experiments that show the benefits of each improvement proposed for the PA* heuristics. Only target points that are located in unreachable regions are considered for this analysis. Those are the only ones associated with critical points, thus being the regions where it is possible to use structured information from visibility maps to help the performance of PA* by improving its heuristics. For targets in other regions, the algorithm uses just the base PA* heuristic, resulting in no negative impacts on efficiency. We consider 5 variants of PA*: base PA*, PA* with improved heuristic h_1 (PA1), PA* with h_1 and use of Eq. 37 (PA1R), PA* both improved heuristics h_1 and h_2 and Eq. 37 (PA1R2), and finally PA* with both h_1 and h_2 and Eq. 44 (PA1R2A). The color meaning of figures in this section is explained in Table 1. Search colors are applied only on top of the white regions of the map representation (free configuration space or reachable space, depending on PA* version in consideration), because white regions represent all the possible nodes that can be visited in search. Point colors are

Table 1 Color meaning for map, search and points

		Base PA*	PA* variants
Map Colors	⬛	Obstacles	
	⬜	C^{free}	$Reach(S)$
	▓	Free Space$\backslash C^{free}$	$U(S)\backslash V(S)$
	▒	-	$V(S)\backslash Reach(S)$
Search Colors	🟦	Nodes on Open List	
	🟦	Expanded Nodes on Closed List	
	🟪	Expanded Nodes Feasible for Perception	
	🟥	Nodes Tested as Goal Position	
Point Colors	🟦	Starting Position	
	🟩	Target Position	
	🟥	Final Path Position for Perception	

used on top of the layer with search colors. In the map colors layer, light gray is used only for the improved PA* versions to represent the visible parts of the unreachable space. For the layer with search colors, purple represents the expanded nodes that are considered feasible locations for perception ($c(n, T_e)$ has a finite cost), and orange represents the nodes tested as goal position using ray casting.

First of all, we evaluate the impact of the first heuristic improvement h_1, from Eq. 35. In this heuristic the distance to the critical point d^c is used as an indication of what is the minimum perception distance to the target. If the distance to the critical point is greater than the optimal sensing distance from the base PA*, d_s^*, the heuristic can use this distance to make a more realistic estimate of the cost to perceive the goal. The impact of this heuristic improvement is greater when the difference to the base heuristic is bigger, i.e., when the λ parameter is higher (lower optimal sensing distance of PA*, d_s^*), and the distance to the critical point greater. In those cases, for targets in unreachable regions whose real minimum perception distance is large, the base PA* algorithm will reach the critical point but not consider it as the final position, and search will continue expanding nodes with lower f values, assuming it might be possible to perceive the target from a smaller and better distance. Then, only after having explored a large portion of the space, the algorithm will find the critical point to be the right perception position, and test it with ray casting in the stopping condition. However, using the information from the visibility maps, i.e., the distance to the respective critical point (or minimum distance in case of multiple critical points from where the target can be perceived), the improved heuristics does not consider an unrealistic perception distance. Therefore, it converges much faster to the solution, because it knows shortly after reaching the critical point that no other nodes can have lower cost, and node expansion is greatly minimized. Figures 16b, 17, 18b, 19 and 20b show the great impact of this heuristic for small d_s^* in comparison to d_l^c, while Figs. 21b, 22, 23, 24 and 25b show the much smaller impact of this heuristic for similar d_s^* and d_l^c.

Moreover, when using the improved heuristic h_1 it is also possible to use the distance to the critical point, d^c, as an additional improvement on efficiency by filtering the elements that are tested with the stopping condition (PA-1R). Given the distance to the critical point being the minimum perception distance possible, we know that nodes with lower distances to the target cannot be the last position of the path, and obstacles are guaranteed to be in between, not allowing line-of-sight. In the base PA* algorithm, all expanded nodes are connected to the final node T_e if their distance is less than the maximum perception range. However, in our improved heuristic, using Eq. 37, only nodes with a distance to the target of at least d_l^c are connected to T_e, thus reducing

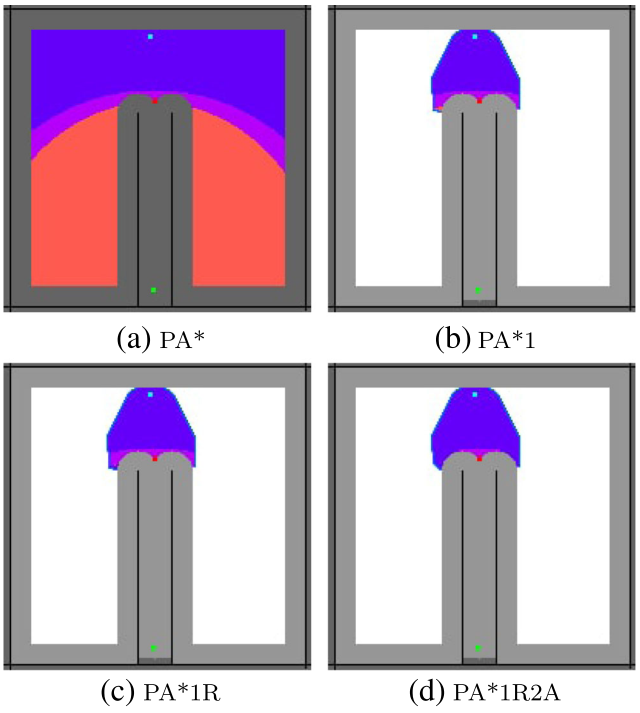


Fig. 16 S1: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.04$, quadratic perception cost function, $d_s^* = 12.5$, and target inside unreachable region with large distance to critical point

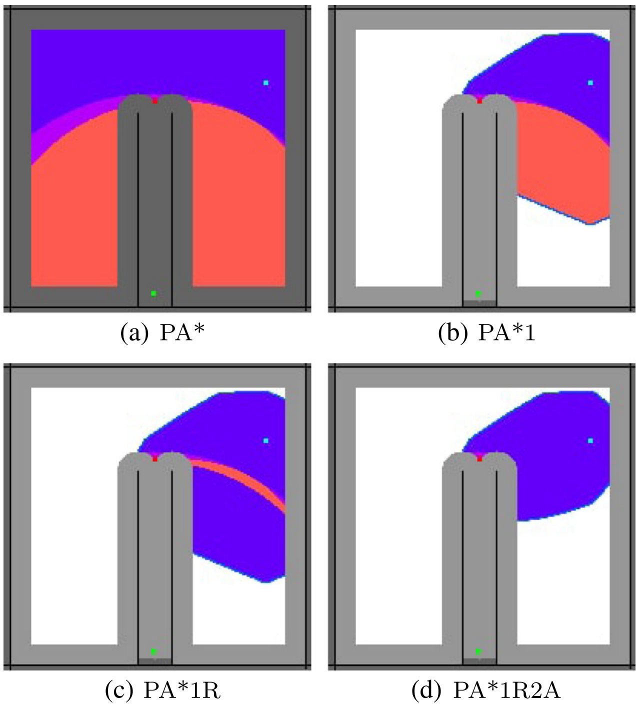


Fig. 17 S2: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.04$, quadratic perception cost function, $d_s^* = 12.5$, and target inside unreachable region with large distance to critical point

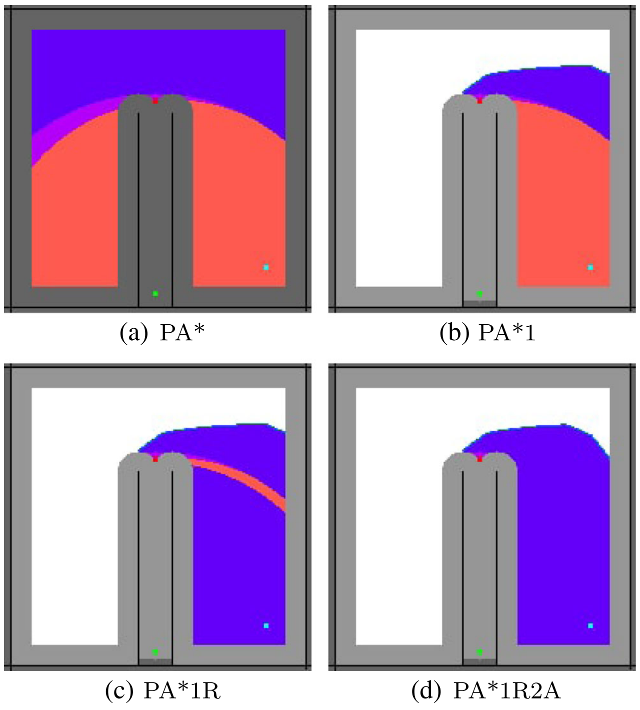

(a) PA*
(b) PA*1
(c) PA*1R
(d) PA*1R2A

Fig. 18 S3: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.04$, quadratic perception cost function, $d_s^* = 12.5$, and target inside unreachable region with large distance to critical point

the number of nodes tested with ray casting as feasible final positions of the path. This is an important contribution, because ray casting is expensive to compute, and enables us to improve the search time. As seen in all figures, the cloud of points that are expanded remains the same as before, but this feature reduces greatly the number of nodes considered feasible, and as such it reduces the number of ray casting operations to test for line of sight (the location of ray casting tests is represented as orange in the figures). While the h_1 heuristic has small impact in some scenarios, the PA-1R variant has a good impact independently of λ and d_l^c, as confirmed in Figs. 23c to 25c.

As showed in the previous section, h_1 only updates the perception distance estimate, being agnostic to the critical point positions. Therefore, many times h_1 makes the search expand nodes in undesirable directions, possibly contrary to the critical points, not directing search into the only regions from where the target is observable. Again, the expansion with h_1 might still result in a waste of computation resources, even tough it is better than the original heuristic. For that purpose, we contributed h_2, in Eq. 39, which directs the node expansion toward the points from where targets are observable, i.e., the critical points. As shown before, this new improvement can result in a great search efficiency boost. Besides being useful for expanding directly toward the best critical point if there are multiple

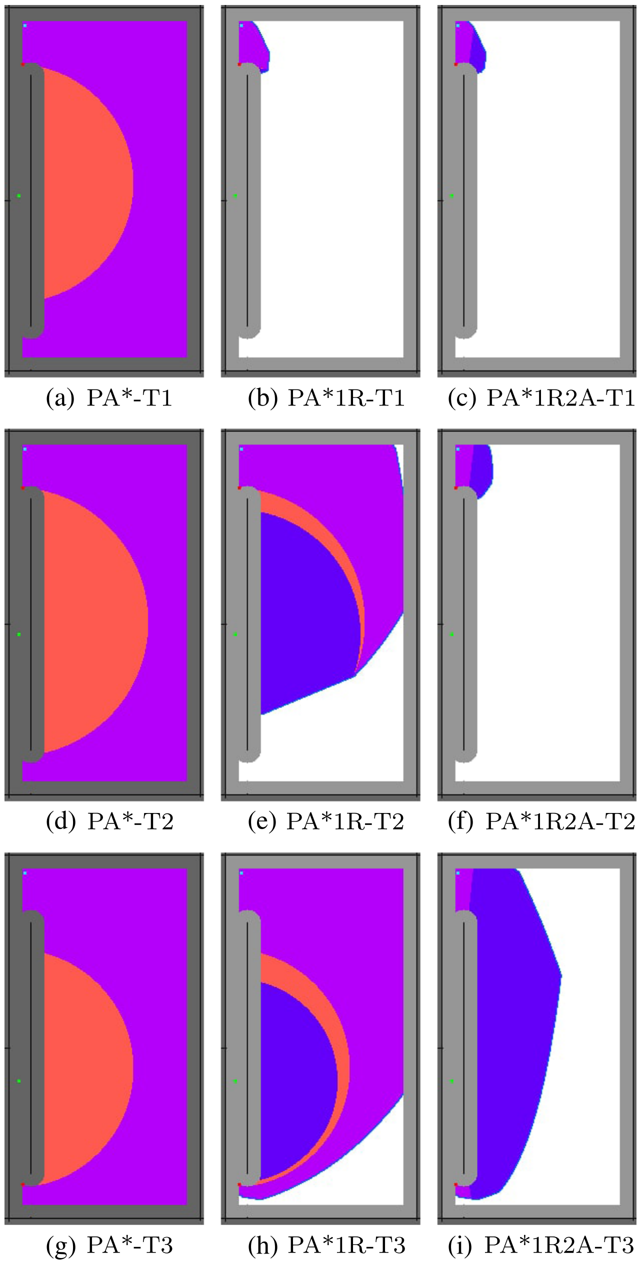

(a) PA*-T1
(b) PA*1R-T1
(c) PA*1R2A-T1
(d) PA*-T2
(e) PA*1R-T2
(f) PA*1R2A-T2
(g) PA*-T3
(h) PA*1R-T3
(i) PA*1R2A-T3

Fig. 19 S4, S5 and S6: Node expansion and ray casting results for search on 375x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.04$, quadratic perception cost function, $d_s^* = 12.5$, and target inside unreachable region with large distance to critical points, for 3 different target positions, T1, T2 and T3

ones (Figs. 26 and 19), this heuristic is also useful to direct search toward the feasible regions for perception of the target, i.e., the critical point, even when there is only one. For the special case where the target, initial position and critical point are aligned, as shown in Fig. 16, the h_1 heuristic is dominant, and h_2 has very little impact. However, as shown in many other figures, the improvements from h_2 can have a great impact on search efficiency. Even

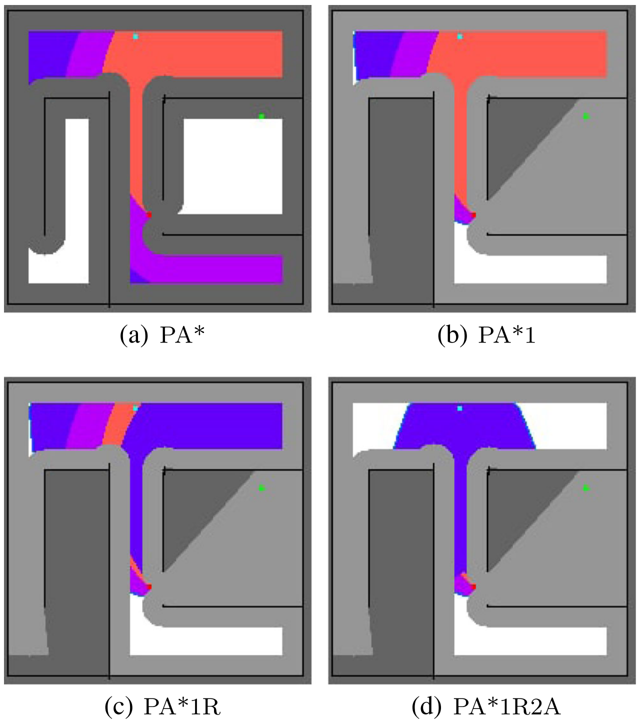


Fig. 20 S7: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.04$, quadratic perception cost function, $d_s^* = 12.5$, and target inside unreachable region with large distance to critical point

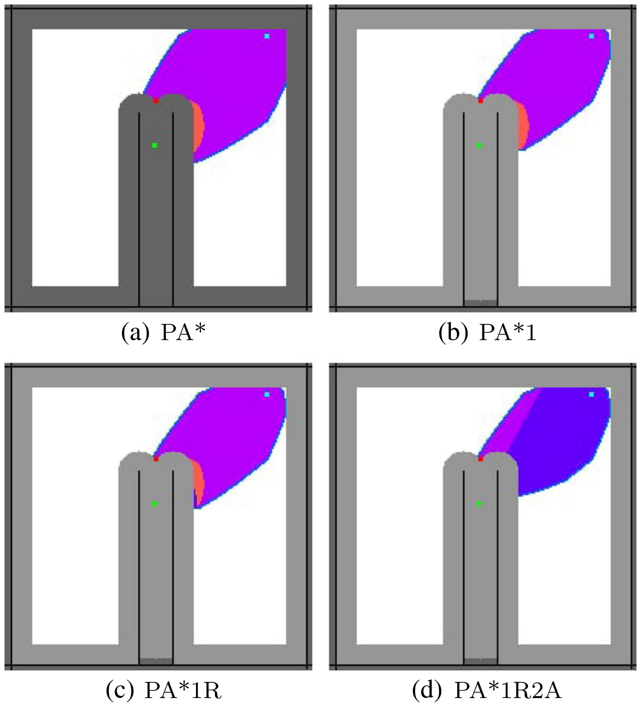


Fig. 22 S9: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.04$, quadratic perception cost function, $d_s^* = 12.5$, similar to S2, but target inside unreachable region with small distance to critical point

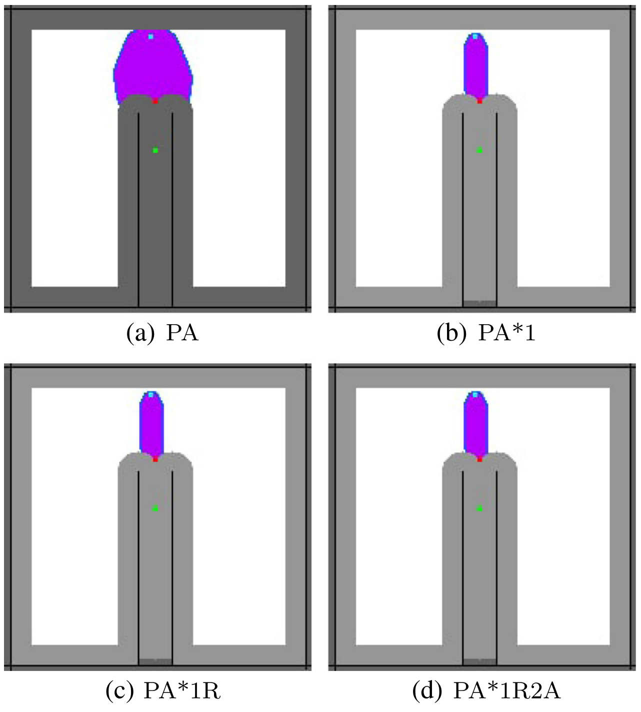


Fig. 21 S8: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.04$, quadratic perception cost function, $d_s^* = 12.5$, similar to S1, but target inside unreachable region with small distance to critical point

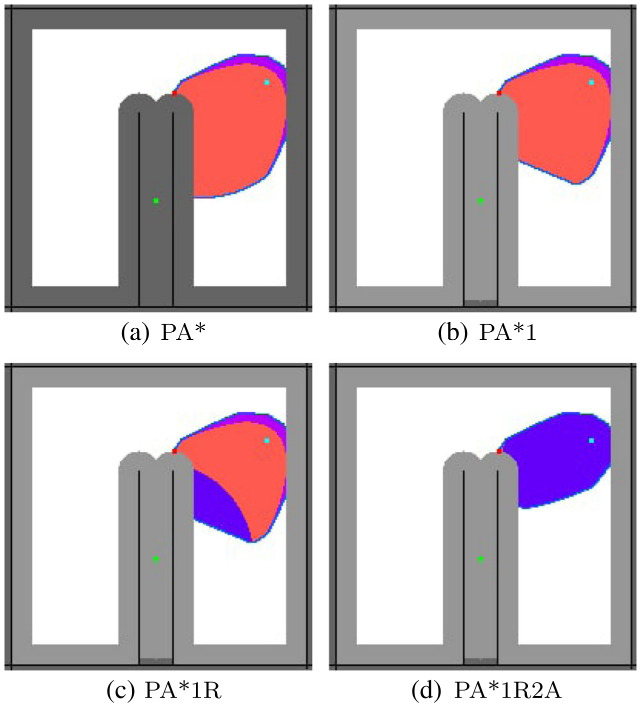


Fig. 23 S10: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13$, $r_p = 130$, $\lambda = 0.007$, quadratic perception cost function, $d_s^* = 71.4$, and target in unreachable region with $d_l^c(T) \approx d_s^*$

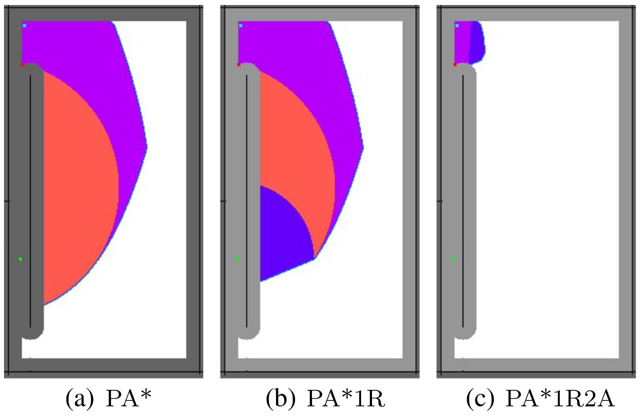


Fig. 24 S11: Node expansion and ray casting results for search on 375x200 gridmap, $R = 13, r_p = 130, \lambda = 0.007$, quadratic perception cost function, $d_s^* = 71.4$, and target in unreachable region with $d_l^c(T) \approx d_s^*$

though for Fig. 18 the PA1R and PA1R2A variants have a similar cloud of expanded nodes at the end of search, an analysis of the initial expansion behavior, from Fig. 27, is useful for understanding what differentiates each variant. While PA* just starts expanding toward the target position, even though it is blocked by a wall, PA1 expands more uniformly, using the information that the target cannot be observed from a small distance. The PA1R variant expands exactly like PA1, but it does not use unnecessary ray casting

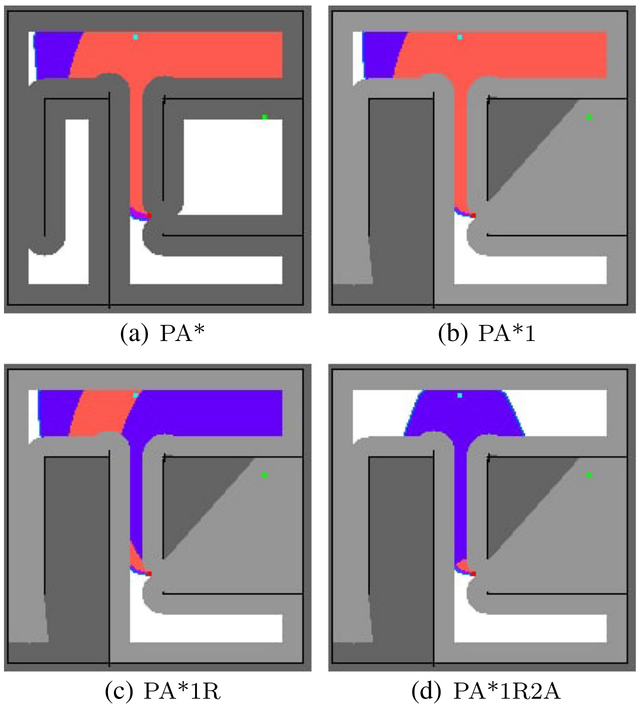


Fig. 25 S12: Node expansion and ray casting results for search on 200x200 gridmap, $R = 13, r_p = 130, \lambda = 0.007$, quadratic perception cost function, $d_s^* = 71.4$, and target inside unreachable region with $d_l^c(T) > d_s^*$

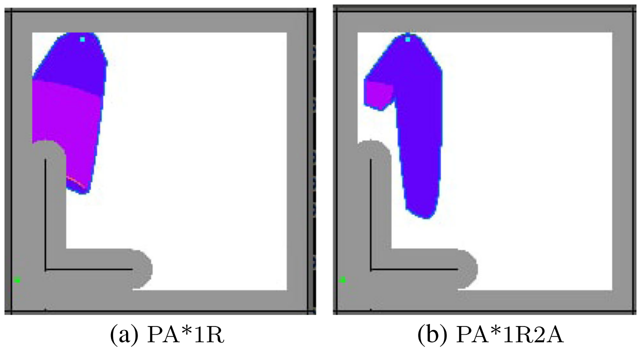


Fig. 26 Progression of node expansion with multiple critical points

operations, resulting in a faster expansion. Finally, PA1R2 and PA1R2A use the critical point location in the heuristic, and as such start expanding in the direction of the critical point. In the Fig. 20 example, the impact of h_2 is visible even at the final search step.

This new heuristic is also capable of being of great help for lower λ, where the d_s^* and d^c might be similar and where h_1 introduces less gains, as shown in Figs. 24 and 25. Heuristic h_2 improvements are less dependent on λ, because the heuristic does not improve only the estimate for the perception distance, but also the motion cost estimate, using the critical points.

When there are multiple critical points, h_2 can produce great differences. As shown in Fig. 19, for the T1 target positions, the target, optimal and closest critical point and initial position are aligned, so h_1 and h_2 produce similar

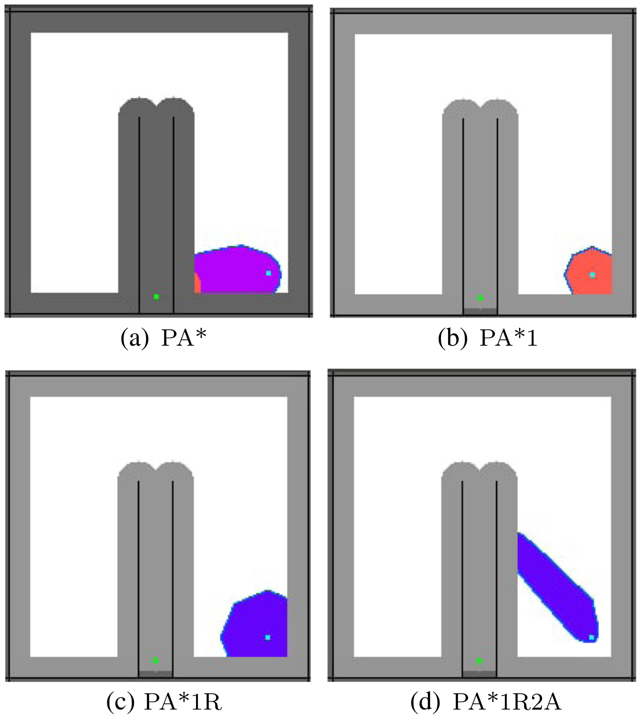


Fig. 27 Initial stage of search for problem S3, Fig. 18

Table 2 Number of expanded nodes (Exp.), number of stopping condition tests (SC) and ray casting operations, and computation time for the 12 test scenarios and the 5 studied variants of PA*

	Test	Exp. (#)	SC (#)	Time (ms)
S1	PA	33312	11795	63
	PA1	1963	11	1
	PA1R	1954	2	2
	PA1R2	1947	2	3
	PA1R2A	1946	1	3
S2	PA	33789	12272	72
	PA1	11076	3694	24
	PA1R	7765	383	8
	PA1R2	5348	329	9
	PA1R2A	5020	1	5
S3	PA	33813	12296	66
	PA1	15604	6684	29
	PA1R	9324	404	8
	PA1R2	9110	404	10
	PA1R2A	8707	1	9
S4	PA	65092	15349	159
	PA1	1122	25	1
	PA1R	1101	4	1
	PA1R2	1096	4	1
	PA1R2A	1093	1	2
S5	PA	70138	20395	150
	PA1	51628	17425	105
	PA1R	36571	2368	68
	PA1R2	1652	39	2
	PA1R2A	1614	1	2
S6	PA	65108	15365	114
	PA1	58165	15365	106
	PA1R	46809	4009	65
	PA1R2	24147	2614	43
	PA1R2A	21534	1	24
S7	PA	11822	4069	18
	PA1	9741	4069	14
	PA1R	6126	454	6
	PA1R2	3620	454	6
	PA1R2A	3185	19	4
S8	PA	1792	1	1
	PA1	532	1	0.5
	PA1R	532	1	0.5
	PA1R2	532	1	0.5
	PA1R2A	532	1	0.5
S9	PA	5143	168	6
	PA1	3961	165	8
	PA1R	3941	145	5
	PA1R2	3778	123	6
	PA1R2A	3656	1	5
S10	PA	9348	4503	21
	PA1	8288	3973	18
	PA1R	7366	3051	16
	PA1R2	5310	2314	14
	PA1R2A	3003	7	3
S11	PA	35430	12681	87
	PA1	33638	11806	90
	PA1R	30128	8296	80
	PA1R2	1157	1	12
	PA1R2A	1157	1	13
S12	PA	10271	4729	16
	PA1	10159	4729	17
	PA1R	6560	1130	8
	PA1R2	3493	854	4
	PA1R2A	2686	47	2

results. On the other hand, for T3, the bottom critical point is optimal, and h_2 is better at directing search toward it. However, more interesting here is the case of T2, where the optimal critical point is the top one, while the minimum perception distance to the target is the distance to the bottom critical point. In that case, h_1 will expand nodes using the distance between the target and the bottom critical point in the heuristic, resulting in a lot of unnecessary expansions while search tries to find a solution with a smaller perception distance. On the other hand, h_2 uses the trade-off between perception and motion cost, and immediately stops search at the top critical point, knowing it is not worth to expand more and thus greatly reducing the number of node expansions.

Table 3 Computation time to construct the Visibility Maps of each 12 test scenarios

Test (#)	Time (s)
S1	0.90
S2	0.99
S3	0.95
S4	1.32
S5	1.32
S6	1.32
S7	0.76
S8	0.92
S9	0.94
S10	0.93
S11	1.33
S12	0.89

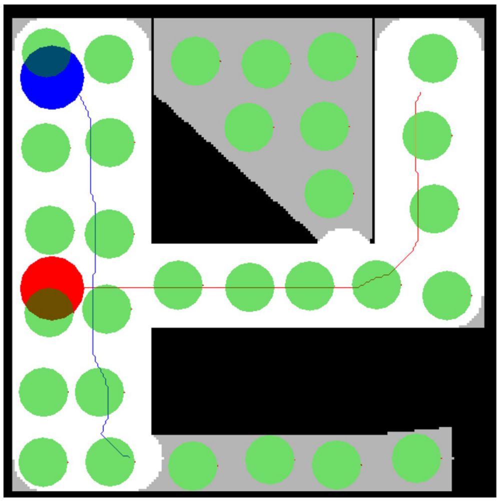


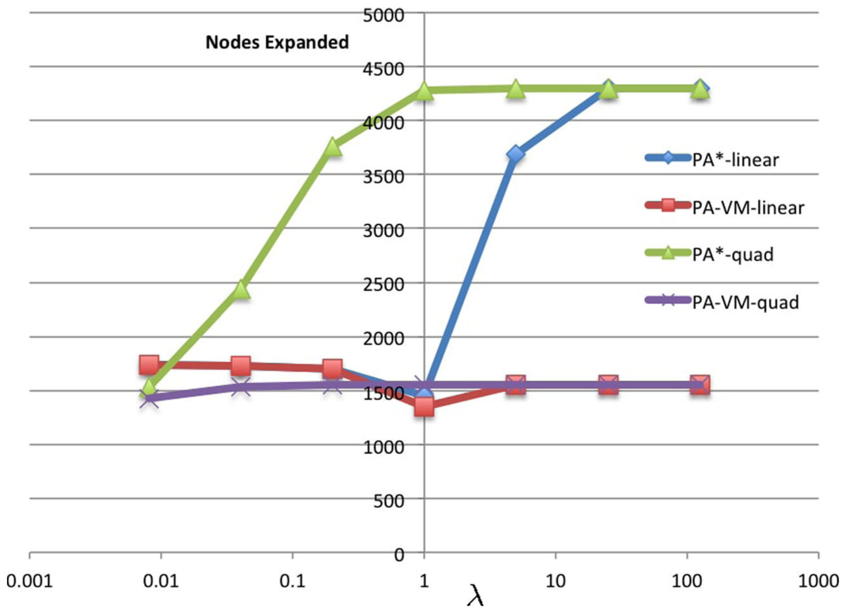
Fig. 28 Path planning for two robots (blue and red) which have to perceive a set of regions of interest (green), minimizing both motion and perception cost, in a 200x200 gridmap, with both robots having size 13, $r_p = 130$ and $\lambda = 0.04$; white represents actuation space, gray the visible space, and black the obstacles or non-visible space

Finally, we consider the last variant, PA1R2A, using the rule from Eq. 44. This case does not need much explanation, as the results speak for themselves. Considering the geometry of the environment, this variant limits the feasible perception positions to the ones in front of a frontier, resulting in a huge reduction of the nodes tested with ray casting (orange in the figures), with great benefits in terms of computation time. As a note, in the same way that PA1 and PA1R had the same node expansion cloud and were only different in the number of expanded nodes that were tested with ray casting, PA1R2 and PA1R2A have the same expansion cloud and again they are only different in terms of the number of ray casting operations, also having a big impact on scenarios with low λ and large optimal perception distance d_s^*.

All the tested scenario's details can be found in Tables 2 and 3, which shows that the time spent building the visibility maps can be gained back with 10 to 15 searches. The visibility maps calculation can also be done once before-hand, as an offline pre-processing of the map, while PA* searches can use its information for faster real-time operation. Moreover, operations to build the visibility map are highly parallelizable, so it is possible to reduce its computation times significantly.

We also run an experiment with multi robot path planning for perception tasks, shown in Fig. 28. This problem has two robots, and a set of regions of interest defined by the user, that have to be observed by any of the robots, while minimizing both the motion and perception cost with the trade-off parameter λ. An algorithm to solve this problem was already proposed [17], which is based on running PA* from different locations and clustering the perception positions in waypoints that can be used with a constructive heuristic to find paths for the robots. This method is heavily dependent on PA*, and we evaluated the impact of our heuristic improvements. As shown in Fig. 28, we chose 30 regions of interest with equal size, out of each only 10 are inside unreachable regions and can benefit from our contributions, while the other 20 will just run with the base PA*. Moreover, we use $\lambda = 0.04$, which results in a small $d_s^* = 12.5$, but the 10 regions that benefit from our heuristics are evenly distributed in space, thus some of them have large distances to the respective critical points, while others have small perception distances from the critical points.

Fig. 29 Comparison of average number of nodes expanded by PA* and PA* with Visibility Maps (PA-VM) as a function of λ, for both linear and quadratic perception cost function



The first part of the algorithm runs PA* multiple times and clusters the perception points, taking 2439 seconds (41 minutes) when using the base heuristic, and only 1088 seconds (18 minutes) when using our contributed PA1R2A variant, reducing more than half the total computation time of this phase. Moreover, the time to compute the visibility maps was only 2 seconds, which is completely negligible compared to the total time to complete all the PA* searches and clustering. The second phase, with the constructive heuristic, took 321 seconds to compute the robot paths, shown in the figure, as a combination of waypoints determined from the first phase.

We also tested in a simulated map of Figure 8a the performance of our proposed algorithm with a dominant heuristic (PA-VM), against the original PA* heuristic, comparing the number of expanded nodes and presenting the results in Fig. 29. The change in efficiency was analyzed as a function of changes of the weight parameter λ, with 7 values ranging between 0.008 and 125. Eight different initial robot positions distributed uniformly in the reachable space were considered, and 25 different target positions, also uniformly distributed, resulting in 1400 different search instances over the range of λ. Because the only difference is for points in $U(S)$, we only tested targets in the unreachable regions. As shown in Fig. 29, for λ greater than one (small optimal sensing distances d^*), there is a great improvement in the average number of nodes expanded, with our method expanding only 35% of nodes expanded by PA*. The results depend highly on the environment topology, having a high variance. Depending on the target position, the node expansion percentage can change from almost 0 to 90%, for large λ. Even for low λ, where both heuristics have similar average results, there were some instances with a gain of 50% in the number of nodes expanded.

7 Conclusions

We reviewed both PA* and Approximate Visibility Map algorithms. The first is an informed search method to find optimal paths for perception tasks. The latter is a map transformation that represents the observable regions in a 2D environment by a given robot. Adding information about the structure of environment can be used to improve the heuristics in PA*, resulting in a reduced search, expanding less nodes. So, the critical points from the Visibility Map were used to create better estimates of the motion and perception costs, while proving they can be used as an admissible and dominant heuristic compared to the one proposed for PA*. In this work only circular robots were considered, with perception cost functions that are rotation invariant. In the future we would like to consider more complex motion and perception cost functions for any-shape robots.

Acknowledgements This work is financed by the ERDF - European Regional Development Fund through the Operational Program for Competitiveness and Internationalization - COMPETE 2020 Program within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013, and also by the FCT grant SFRH/BD/52158/2013 through the CMU-Portugal program.

References

1. Biswas, J., Veloso, M.M.: Localization and navigation of the cobots over long-term deployments. Int. J. Robot. Res. **32**(14), 1679–1694 (2013)
2. Carlone, L., Ng, M., Du, J., Bona, B., Indri, M.: Rao-Blackwellized particle filters multi robot slam with unknown initial correspondences and limited communication. In: IEEE International Conference on Robotics and Automation (ICRA) (2010)
3. Eidenberger, R., Scharinger, J.: Active perception and scene modeling by planning with probabilistic 6D object poses. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2010)
4. Ekvall, S., Jensfelt, P., Kragic, D.: Integrating active mobile robot object recognition and slam in natural environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2006)
5. Fabrizi, E., Saffiotti, A.: Extracting topology-based maps from Gridmaps. In: IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE, vol 3, pp 2972–2978 (2000)
6. Faigl, J.: Approximate solution of the multiple watchman routes problem with restricted visibility range. IEEE Trans. Neural Netw. **21**(10), 1668–79 (2010)
7. Gancet, J., Lacroix, S.: Pg2p: A perception-guided path planning approach for long range autonomous navigation in unknown natural environments. In: 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings (2003)
8. Howard, A., Parker, L., Sukhatme, G.: Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection. Int. J. Robot. Res. **25**(5-6), 431–447 (2006)
9. Kaelbling, L.P., Lozano-Pérez, T.: Unifying perception, estimation and action for mobile manipulation via belief space planning. In: 2012 IEEE International Conference on Robotics and Automation (ICRA) (2012)
10. LaValle, S.M.: Planning algorithms. Cambridge University Press, Cambridge (2006)
11. Lu, D.V., Smart, W.D.: Towards more efficient navigation for robots and humans. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1707-1713. IEEE, Cambridge (2013)
12. Pandey, A.K., Alami, R.: Mightability maps: a perceptual level decisional framework for co-operative and competitive human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2010)
13. Pereira, T., Veloso, M., Moreira, A.: Multi-robot planning using robot-dependent reachability maps. In: Robot 2015: Second Iberian Robotics Conference (2015)
14. Pereira, T., Moreira, A.P., Veloso, M.: Improving heuristics of optimal perception planning using visibility maps. In: The IEEE International Conference on Autonomous Robots Systems and Competitions, ICARSC'16 (2016)
15. Pereira, T., Veloso, M., Moreira, A.P.: PA*: Optimal path planning for perception tasks. In: The European Conference on Artificial Intelligence, ECAI'16 (2016)

16. Pereira, T., Veloso, M., Moreira, A.P.: Visibility maps for any-shape robots. In: The IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'16 (2016)
17. Pereira, T., Moreira, A.P., Veloso, M.: Multi-robot planning for perception of multiple regions of interest. In: The third Iberian Robotics Conference, accepted at ROBOT'2017 (2017)
18. Potthast, C., Sukhatme, G.S.: A probabilistic framework for next best view estimation in a cluttered environment. J. Vis. Commun. Image Represent. **25**(1), 148–164 (2014). https://doi.org/10.1016/j.jvcir.2013.07.006
19. Rusu, R.B., Şucan, I.A., Gerkey, B., Chitta, S., Beetz, M., Kavraki, L.E.: Real-time perception-guided motion planning for a personal robot, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2009)
20. Velez, J., Hemann, G., Huang, A., Posner, I., Roy, N.: Planning to perceive: Exploiting mobility for robust object detection. Icaps pp 266–273 (2011)

Tiago Pereira is a Ph.D. candidate in the CMU-Portugal Program, being a student in the Electrical and Computer Engineering Department both at Carnegie Mellon University and Faculty of Engineering of University of Porto. He is also a research assistant at INESC TEC. He received an Integrated B.S and M.S. in Electrical and Computer Engineering from University of Porto in 2012. His research interests include motion planning and coordination of heterogeneous multi-robot systems.

António Moreira is currently an Associated Professor in Electrical Engineering, developing his research within the Robotic and Intelligent Systems Centre of INESC TEC (Centre Coordinator). He graduated with a degree in Electrical Engineering from the University of Porto in 1986. He then pursued graduate studies at the University of Porto, completing a M.Sc. degree in Electrical Engineering - Systems in 1991 and a Ph.D. degree in Electrical Engineering in 1998. From 1986 to 1998 he also worked as an assistant lecturer in the Electrical Engineering Department of the University of Porto. His main research areas are Process Control and Robotics.

Manuela Veloso is the Herbert A. Simon University Professor in the School of Computer Science at Carnegie Mellon University. She is the Head of the Machine Learning Department. She researches in Artificial Intelligence, Robotics, and Machine Learning. She founded and directs the CORAL research laboratory, for the study of autonomous agents that Collaborate, Observe, Reason, Act, and Learn, www.cs.cmu.edu/~coral. Professor Veloso is AAAI, Fellow, ACM Fellow, AAAS Fellow, and IEEE Fellow, and the past President of AAAI and of RoboCup. Professor Veloso and her students research a variety of autonomous robots, including mobile service robots and soccer robots. See www.cs.cmu.edu/~mmv for further information, including publications.