Multi-Robot Planning for Perception of Multiple Regions of Interest

Tiago Pereira[1][2][3], A. Paulo G. M. Moreira[1][2], and Manuela Veloso[3]

[1] Faculty of Engineering, University of Porto, Portugal
{tiago.raul, amoreira}@fe.up.pt
[2] INESC TEC, Porto, Portugal
[3] Carnegie Mellon University, Pittsburgh, USA

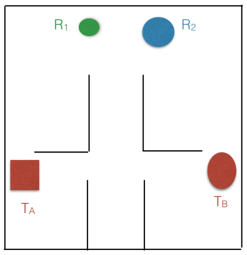
Abstract. In this paper we address the allocation of perception tasks among a set of multiple robots, for tasks such as inspection, surveillance, or search in structured environments. We consider a set of target regions of interest in a mapped environment that need to be sensed by any of the robots, and the problem is to find paths for the robots that cover all the target regions with minimal cost. We consider not only sensing range when determining paths for the robots to perceive the targets, but also a sensor cost function that can be adapted to each robot's sensor. Thus the planning has to search for paths with minimal motion and perception cost, instead of the traditional approach where line-of-sight is the only requirement in a motion cost minimization problem. Our contribution is to use planning to determine possible perception positions for every robot, which we cluster and then use as possible waypoints that can be used to construct paths for all the robots. Given the combinatorial characteristics of path determination in this setting, we contribute a construction heuristic to find paths that guarantee full coverage of all the feasible perception target regions, while minimizing the overall cost. We assume robots are heterogeneous regarding their geometric properties, such as size and maximum perception range. We consider simulated scenarios where we show the benefits of our approach, enabling multi-robot path planning for perception of multiple regions of interest.

Keywords: multi-robot, perception, planning

1 Introduction

In this work we consider multiple heterogeneous robots that have to plan together in order to perceive a set of target regions of interest. The robot's physical characteristics are considered when planning their paths in a structured environment that has been mapped before. For a given environment, not all target positions can be perceived by all robots. Using the intrinsic differences of each robot in problems such as task allocation also allows for more efficient planning, by reducing the combinatorial possibilities of the search space.

We consider a 2D gridmap of obstacles to represent the environment, and mobile robots that are heterogeneous in regard to geometric properties, such as size and sensing range. As shown in Figure 1, we assume there is a set of heterogeneous robots (Rs) and target regions of interest (Ts) that need to be perceived. The target regions can represent areas that need to be covered by the robot's sensors for inspection or search. The regions of interest could also represent location uncertainty around a point that needs to be perceived, and the target regions can have any shape and size.



(a) Problem

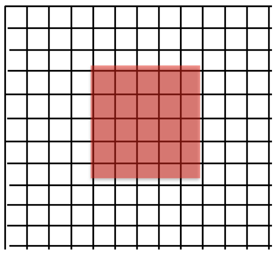

(b) Target Region in Gridmap

Fig. 1. Environment with obstacles represented in black, circular robots R and target regions of interest T that need to be perceived; in the right image, a target region with a given shape, size and position in a gridmap which results in discretization of Ts.

In traditional multi-robot path planning for perception tasks, an infinite perception range is a common assumption, or even a finite maximum range. However, the cost of perception should also be included when determining paths for robots executing perception tasks. Therefore, we introduce the following problem, where the goal is to find paths for each robot that minimize the total cost of motion and perception, given by

$$\text{cost} = \sum_{R} C_R + \lambda \sum_{T} C_T \tag{1}$$

where C_R is the path size for robot R, C_T is the cost of perception of target region T, and λ is the trade-off parameter between perception cost and motion cost. We assume all target regions have to be observed.

The cost of perception of a target region T perceived from a robot depends on its path ρ, and we assume it is the average of perception cost for the grid points inside the region of interest

$$C_T(\rho) = \frac{1}{\#T} \sum_{\mathbf{t} \in T} \min_{\mathbf{p} \in \rho} c_p(||\mathbf{p} - \mathbf{t}||) \tag{2}$$

The number of points of the gridmap inside the target region is represented by $\#T$. For multiple robots, C_T uses the minimum of the perception cost not only for ρ, but the paths of all robots.

The perception cost function, c_p, models sensor accuracy and it is function of perception distance d_p. As an example, if the sensing error increases quadratically with distance, then the perception cost is a quadratic function.

$$c_p(d_p) = d_p^2 \tag{3}$$

Given the problem with robots Rs and targets Ts, a planner is used to find the paths for each robot such as all the target regions are perceived by at least one robot, and the overall cost function is minimized. We assume the overall motion cost to be a weighted sum of all paths' sizes, thus minimizing the energy spend to move the robots by using appropriate weights for each robot.

The approach we contribute has a first step to determine perception points for each target grid point. For that we use PA* [7], a technique to determine from a given initial position the optimal perception position to perceive a target, assuming some perception cost function and the λ parameter. We then cluster the perception points, and use the clusters as new initial positions from where to run PA* again. Our algorithm is then able to obtain a set of clusters that can be used as waypoints for path planning.

In the second step, the planner uses the set of waypoints to construct paths for each robot. Given the combinatorial nature of our problem, we use a constructive heuristic to iteratively add new waypoints to the robots' paths, and construct a solution that covers all the targets that need to be perceived, while minimizing the overall cost. We contribute an algorithm that can be used to find paths to perceive target regions of interest both for single and multi-robot teams.

In the next sections we describe our proposed method in more detail.

2 Perception Clusters from PA*

We start by considering first a single robot scenario. For each target grid point $\mathbf{t}$ inside target regions of interest, we run PA* to find a path to perceive $\mathbf{t}$ from initial robot position $\mathbf{r}$, optimizing for both motion and perception costs using λ as the trade-off parameter, as shown in Figure 2. PA* returns the optimal path with minimal cost, where the final position is the optimal perception point. PA* search results in a perception point $\mathbf{p}_\mathbf{t}^\mathbf{r}$ for each $\mathbf{t}$.

We should note that this perception position is optimal only for the local scenario of a robot starting at $\mathbf{r}$ to perceive $\mathbf{t}$, but it is not necessarily optimal in the multiple target regions scenario. However, we use these points as an initial step for constructing paths for the robots to perceive those regions.

The robots' paths can then be obtained as a combinatorial solution of the determined perception points. Unlike the traveling salesman problem (TSP), not all perception points need to be visited, and the robot does not need to return to the initial position. In order to avoid a combinatorial explosion for the path planning, we cluster perception points based on distance. The point closer to each cluster's center of gravity is the one used as waypoint in the path planning, and the perception cost for each $\mathbf{p}_\mathbf{t}^\mathbf{r}$ associated with the respective cluster.

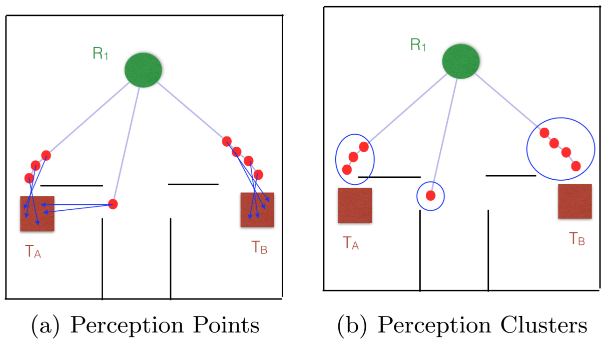


Fig. 2. When running PA* from the robot initial position to each point inside target regions, the search returns an optimal perception point, shown as a red dot; in order to reduce the number of possible combinations, the perception points are clustered in groups that can be used as single waypoints in path planning, as shown in (b).

The proposed approach does not find all needed perception points, as the optimal paths from PA* depend on the initial position. So, the PA* search to targets $\mathbf{t}$ needs to be re-run again from each cluster centroid, resulting in new perception points $\mathbf{p}_{\mathbf{t}}^{\mathbf{q}}$. New clusters might appear from each iteration when running PA* from new initial positions, as shown in Figure 3(a). If a new cluster's centroid is close to an existing one they can be merged, with the robot radius being the merging threshold. Cost of perception of target point $\mathbf{t}$ in cluster P_i is

$$c_{\mathbf{t}}^{i} = \min_{\mathbf{p}_{\mathbf{t}}^{\mathbf{q}} \in P_i, \mathbf{q} \in \{\mathcal{Q} \bigcup \mathbf{r}\}} c_p(||\mathbf{p}_{\mathbf{t}}^{\mathbf{q}} - \mathbf{t}||) \tag{4}$$

where $\mathcal{Q}$ is the set of cluster centroids.

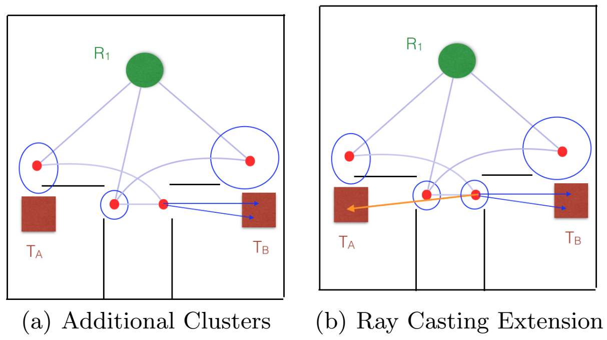


Fig. 3. When running PA* from cluster centroids, new perception points might result in new clusters, as shown in (a); from that clustering strategy some clusters might only be associated with certain targets, and additional perception feasibility to other target points can be obtained using ray casting to test for line-of-sight, as shown in (b).

Clusters are generated by running PA* to target points $\mathbf{t}$ from different initial positions, but $c^i_{\mathbf{t}}$ is only determined if PA* searches to $\mathbf{t}$ result in perception points that are clustered to P_i. Nevertheless, other target points might still be observable from cluster P_i, even if PA* finds the cluster position non-optimal to perceive those points. In Figure 3(b), for every cluster centroid, ray tracing is used to determine line-of-sight and perception cost to other target points $\mathbf{t}$ whose cost was not previously determined as $c^i_{\mathbf{t}}$. Ray tracing determines perception feasibility from a cluster centroid to any other target point, and the respective distance is used to associate a perception cost to the tuple centroid-target point.

3 Path Construction

Even though there might not be any connections between some pair of clusters initially, we still consider them in the heuristic path construction, as shown in Figure 4, because PA* is optimal locally for each target point but is globally sub-optimal in the general multi-target path planning setting.

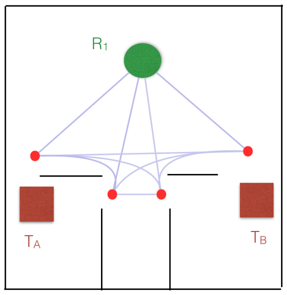


Fig. 4. Map with robot and target regions of interest, with red dots as cluster centroids and lines connecting all of them showing all the path's combinatorial possibilities.

The clusters centroids can be used as waypoints when determining the path for a robot to perceive all the target points. Pairwise distances between all cluster centroids and initial robot position can easily be determined with A*. The waypoints are $\mathbf{q}_j$, with $0 \leq j \leq m$ where m is the number of clusters and $\mathbf{q}_0 = \mathbf{r}$ is the initial position. The path ρ can be represented as a sequence $\{s_i\}$, with $0 \leq i \leq L$ (L is path length in terms of number of clusters covered) and $1 \leq s_i \leq m$ for $i \geq 1$ and $s_0 = 0$. The path cost is then given by:

$$\mathrm{cost}(\rho) = \sum_{i=1}^{i \leq L} \mathrm{dist}(\mathbf{q}_{s_{i-1}}, \mathbf{q}_{s_i}) + \lambda \sum_{T} \left(\frac{1}{\#T} \sum_{\mathbf{t} \in T} \min_{1 \leq i \leq L} c^{s_i}_{\mathbf{t}} \right) \tag{5}$$

Any point can be visited more than once, but that would be redundant. Moreover, not all points need to be visited. Given the combinatorial characteristics of this problem, solving it optimally for any $m > 10$ is already very time consuming. Therefore, we use a construction heuristic to iteratively construct a path from the initial position that covers all the target points with the robot's sensor. Examples of constructive heuristics used in the TSP are the nearest neighbor, nearest insertion, cheapest insertion, and farthest insertion.

Improvement heuristics could be used to improve the solution once a feasible path is found. Examples are point removal, k-opt moves, and metaheuristics.

At each iteration, and for each point i that can still be inserted in the robot's path, the added motion cost is given by the cheapest insertion, which finds the best position in the current path to insert the new point.

$$\begin{aligned}\text{cost}_m(i) = \min\Big(&\min_{1\le j\le L} \text{dist}(\mathbf{q}_{s_{j-1}}, \mathbf{q}_i) + \text{dist}(\mathbf{q}_i, \mathbf{q}_{s_j}) - \text{dist}(\mathbf{q}_{s_{j-1}}, \mathbf{q}_{s_j}),\\ &\text{dist}(\mathbf{q}_{s_L}, \mathbf{q}_i)\Big)\end{aligned} \tag{6}$$

For each point to be inserted, there is also a possible gain associated with the improvement in perception cost from sensing from a closer distance.

$$\text{gain}_p(i) = \lambda \sum_T \frac{1}{\#T} \sum_{\mathbf{t}\in T} \max\left(\min_{0\le j\le L}\left(c_{\mathbf{t}}^{s_j}\right) - c_{\mathbf{t}}^{i}, 0\right) \tag{7}$$

We use for $c_{\mathbf{t}}^0$ the maximum perception cost, $\lambda c_p(r_p)$, where r_p is the maximum perception range. The bigger $c_{\mathbf{t}}^0$, the highest priority is given to points that perceive previously unseen target points, which is a behavior similar to the farthest heuristic. Points are considered valid if gain positive, or if it adds visibility to any previously unseen target. Otherwise the planner might not add to the path the only positions that can observe some far away target, even though we want complete coverage. The overall base method is shown in Algorithm 1.

Algorithm 1 Base Path Construction from Cluster Centroids

```
Require: List of points to insert: ${1..m}$
 1: while There is valid points to choose from do
 2:     for all points not yet inserted do
 3:         Find added motion cost, $\text{cost}_m(i)$ as cheapest insertion of point $i$
 4:         Find gain in perception cost, $\text{gain}_p(i)$
 5:         $\text{gain}(i) = \text{gain}_p(i) - \text{cost}_m(i)$
 6:         if $\text{gain}(i)$ is valid then
 7:             Add $i$ to list of points to consider for insertion in this iteration
 8:         end if
 9:     end for
10:     Choose point that maximizes gain
11:     Insert point in path according with cheapest insertion
12:     Update path perception cost to each target point
13: end while
14: Return Path
```

3.1 Avoiding Local Minima

As shown in Figure 5, the base algorithm presented before can very easily get stuck in local minima, as it is based on a greedy heuristic. In the figure's example, in the first iteration cluster 1 has the highest gain and is added to the robot's path, but as we show that point is not even part of the optimal path.

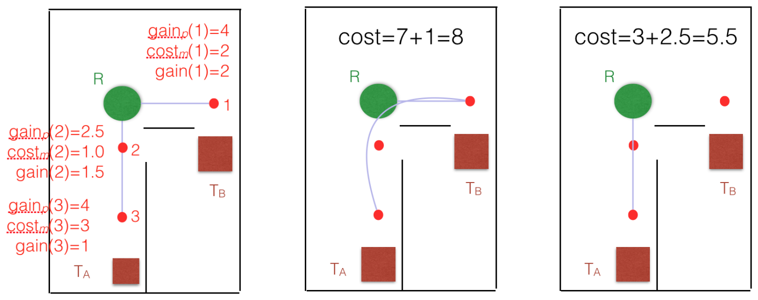


Fig. 5. In the left image, the robot can move to three cluster centroids from its initial position, and cluster 1 has the highest gain, considering a quadratic perception cost and $\lambda = 0.5$; in the middle figure we show a path that moves through cluster 1 and then to cluster 3 in order to perceive both T_A and T_B, with motion cost 7 and perception cost of 1 $(2 \times 0.5 \times 1^2)$; in the last image, we show the optimal path that moves through cluster 2 and then cluster 3, perceiving both targets with a lower overall cost, motion cost equal to 3 and perception cost of 2.5 $(0.5 \times 2^2 + 0.5 \times 1^2)$.

To help avoid local minima, we contribute a n-level depth search for the greedy constructive heuristic. Instead of looking only one step ahead, it looks at the insertion of n points, and chooses the one with minimal cost. For that purpose we use Algorithm 2, where we contribute a recursive function that implements the n depth search and testing combinations of n points to insert. This function is called once in each iteration, returning the best point to insert in the path at each time, until there is no points to insert in the robot's path.

Because we consider combinations of n points and we use the cheapest insertion heuristic, a 2-level search that inserts first cluster i and then the cluster centroid j has the same gain as the reverse, inserting first cluster j and then i. As a tiebreaker rule, we insert first the point with the highest gain in the top level of the recursive search (variable determined on line 7 of Algorithm 2).

4 Multi-Robot

The extension of the previous n-depth heuristic from the single robot approach to the multiple robot setting is now straightforward. We build clusters of perception

Algorithm 2 Recursive function used in n-depth heuristic for path construction

```
Require: List of points to insert: $\{1..m\}$
 1: function SEARCH(dists, $c_{\mathbf{t}}$'s, path, $n$)
 2:     if n==0 then return $< -1, 0 >$
 3:     end if
 4:     for all points not yet inserted do
 5:         Find added motion cost, $\text{cost}_m(i)$ as cheapest insertion of point $i$
 6:         Find gain in perception cost, $\text{gain}_p(i)$
 7:         $\text{gain}(i) = \text{gain}_p(i) - \text{cost}_m(i)$
 8:         if $\text{gain}(i)$ is valid then
 9:             Create new temporary path, $\text{path}_i$, updated with insertion of point $i$
10:             Find the gain from next (n-1)-depth search:
11:             $< j, \text{nextgain}(i) >=$ SEARCH(dists, $c_{\mathbf{t}}$'s, $\text{path}_i$, $n-1$)
12:             $\text{overall\_gain}(i) = \text{gain}(i) + \text{next\_gain}(i)$
13:         end if
14:     end for
15:     if no valid point then
16:         return $< -1, 0 >$
17:     end if
18:     Choose point that maximizes overall gain
19:     return $< i, \text{overall\_gain}(i) >$
20: end function
```

points from PA* for all the robots. Then the construction heuristic considers multiple lists of cluster centroids and at each search level can choose to add any of those points to the respective robot's path. Insertion on paths at different depth levels of the recursive search might be for different robots.

The complexity of the n-level heuristic search in the multi robot scenario is $M!/(M-n)!$ in each iteration, where M is the total number of cluster centroids over all robots. In each iteration one cluster is added to a robot's path.

However, new inefficiencies of the heuristic arise in the multi-robot scenario, as shown in Figure 6. In that example, either cluster centroids 1 or 2 can be added to the respective robot's paths. From point 2, all target points can be observed, but from point 1 only part of T_A can be observed. Using constructive heuristic with a 1-level search, adding point 1 to R_1 path has a higher gain, even though in the next iteration R2 will still have to move to point 2 in order to perceive the yet unseen parts of T_A, resulting in sub-optimal path construction. In some cases this inefficiency can be solved with higher n, as here a 2-level search would already avoid this problem. Nevertheless, for big problems with multiple targets and robots, n has to be small in order to reduce the search complexity, and might not be enough to solve this inefficiency.

4.1 Unfeasibility subsets

There are target points that can be perceived by all robots, and others that can only be observed by a subset of robots. Therefore, the idea is, at each iteration

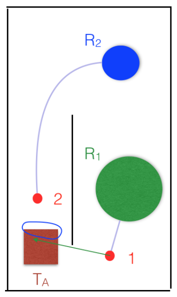


Fig. 6. Inefficiency arising from different robots being able to perceive targets from different locations; here R_2 is able to move to cluster 2 and perceive all points in T_A, but R_1 can only move to the first cluster where it only perceives a part of the target.

of the path construction phase, to consider first cluster centroids that are the only ones that can observe some target points. We start by centroids that are associated with targets that are perceived by one robot only, then by two, and so on, until the only remaining are the ones that can be observed by any robot. Using this approach solves the problem in Figure 6 without increasing n. The separation of cluster centroids by subsets of unfeasibility can be accomplished by adding a component proportional to the number of robots that cannot perceive a target, and the maximum gain, $K\lambda c_p(r_p)$, where K is the number of regions.

Our complete contribution using unfeasibility sets is shown in Algorithm 3.

4.2 Results

We show in Figure 7 the resulting paths for the planning problem of 2 heterogeneous robots perceiving 3 regions of interest, for a large λ that makes robots move close to the target regions. We consider two test scenarios with a changing position for one of the target regions, and we show how it impacts the resulting plan. The smaller robot 1 can get into the region where the changing target is, and observe it from a close distance. However, the bigger robot 2 can only perceive this region from a distance. Therefore, when the target moves closer to the opening from where it is perceived, the perception cost for the bigger robot reduces and the planner moves this robot such has it perceives two target regions, while the first robot moves to perceive the target that can only be observed by the first robot. Nevertheless, when the changing target moves away from the opening, the quadratic perception cost for robot 2 increases significantly, and as a result there is a point from where it is worth for the robot 1 to move forth and back to observe all the target regions from a closer distance.

For scenarios with cluster lists up to 10 centroids per robot, we also run a brute-force algorithm to test all possible combinations and compare with our heuristic. In the simulated environment we used, shown in Figure 7, but with varying targets' sizes and positions, the heuristic always returned the same paths

Algorithm 3 Recursive n-level constructive heuristic with unfeasibility subsets

```
Require: List of points to insert: $\{1..m\}$
 1: function SEARCH(dists, $c_{\mathbf{t}}$'s, paths, $n$)
 2:     if n==0 then return $< -1, -1, 0 >$
 3:     end if
 4:     for all $< r, i >$ all robot and cluster points not yet inserted do
 5:         Find $\text{cost}_m(r, i)$, as cheapest insertion of point $i$ in path of robot $r$
 6:         Find gain in perception cost, $\text{gain}_p(r, i)$
 7:         for all $\mathbf{t}$ do
 8:             if $\mathbf{t}$ is not yet observed by any robot path then
 9:                 $\text{unfeas\_gain}(r, i, \mathbf{t}) = \#(\text{Robots that cannot perceive } \mathbf{t}) \times (K\lambda c_p(r_p))$
10:             end if
11:         end for
12:         $\text{gain}(r, i) = \text{gain}_p(r, i) - \text{cost}_m(r, i) + \max_{\mathbf{t}}(\text{unfeas\_gain}(r, i, \mathbf{t}))$
13:         if $\text{gain}(r, i)$ is valid then
14:             Create new temporary paths, $\text{paths}_i$ updated with insertion of point $i$
15:             Find the gain from next (n-1)-depth search:
16:             $< s, j, \text{next\_gain}(r, i) >=$ SEARCH(dists, $c_{\mathbf{t}}$'s, $\text{paths}_i$, $n - 1$)
17:             $\text{overall\_gain}(r, i) = \text{gain}(r, i) + \text{next\_gain}(r, i)$
18:         end if
19:     end for
20:     if no valid point then
21:         return $< -1, -1, 0 >$
22:     end if
23:     Choose point that maximizes overall gain
24:     return $< r, i, \text{overall\_gain}(r, i) >$
25: end function
```

as the brute-force algorithm, but with lower computation time, in the order of seconds, proving its efficiency. For bigger cluster lists, we could only use the heuristic approach for the path planning. For the problems in Figure 7, in a map with 200 by 200 pixels, and a total of 5 clusters for the two robots, the cluster determination took around 30 seconds, and the path construction 5 milliseconds.

5 Related Work

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 correct object detection and classification [9]. Another class of problems for visibility 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 [1], which has been extended to 3D [3].

PA* was proposed to optimally solve the planning for perception of a single target position in 2D gridmaps, given motion and perception costs [7]. It was also shown how to improve search efficiency with robot-dependent information [5].

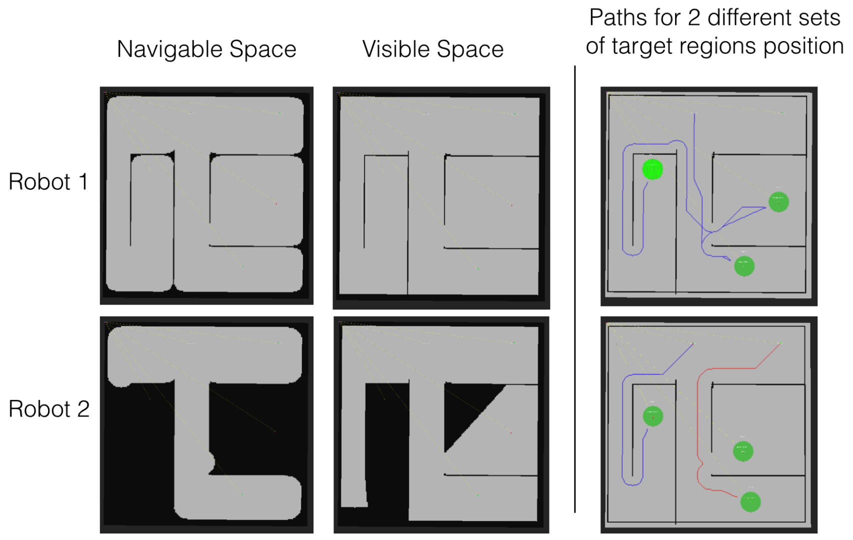


Fig. 7. Planning scenario with two heterogeneous robots and 3 regions of interest; In the first column we show the space where robots can move, and in the middle column the associated visible space of each robot; in the last column, the resulting paths for the robots when one of the regions of interest changes its position.

Planning sequences of perception points to cover regularly all interest points in the environment is also relevant for multi-robot patrolling [8], where a probabilistic strategy was used for a team of agents to learn and adapt their moves to the state of the system at the time, using Bayesian decision rules and distributed intelligence. When patrolling a given site, each agent evaluates the context and adopts a reward-based learning technique that influences future moves.

Other relevant work focuses on the sensing horizon, and how to opportunistically plan navigation and view planning strategy in order to anticipate obstacles with look-ahead sensing [4]. Candidate positions are considered based on the possibility of anticipating obstacles, and used as waypoints. In the same topic, it has also been shown that perception planning and path planning can be solved together [2], selecting the most relevant perception tasks depending on the current goal of the robot, thus successfully solving navigation and exploration tasks together. The sets of unfeasibility have also been used before in heterogeneous multi-robot planning, but for actuation-based tasks [6].

6 Conclusion

In this work we contribute a constructive heuristic for path planning, to use with heterogeneous multi-robot settings in the problem of perception of multiple regions of interest. The solution can be used in inspection, surveillance or search in robotics. We introduce mechanisms to avoid local minima of the proposed heuristic, such as considering sets of unfeasibility, and n-depth search. We were able to successfully generate paths for multiple robots in simulated environments, in a novel problem that considers both motion and perception cost.

Acknowledgments. This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme 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 project TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020, financed by the North Portugal Regional Operational Programme (NORTE 2020), under PORTUGAL 2020 Partnership Agreement, and Carnegie Mellon Portugal Program grant SFRH/BD/52158/2013.

References

1. Faigl, J.: Approximate solution of the multiple watchman routes problem with restricted visibility range. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council **21**(10), 1668–79 (2010). DOI 10.1109/TNN.2010.2070518. URL http://www.ncbi.nlm.nih.gov/pubmed/20837446
2. Gancet, J., Lacroix, S.: Pg2p: A perception-guided path planning approach for long range autonomous navigation in unknown natural environments. In: Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on
3. Janousek, P., Faigl, J.: Speeding up coverage queries in 3D multi-goal path planning. Proceedings - IEEE International Conference on Robotics and Automation (1), 5082–5087 (2013). DOI 10.1109/ICRA.2013.6631303
4. Nabbe, B., Hebert, M.: Extending the path-planning horizon. The International Journal of Robotics Research **26**(10), 997–1024 (2007)
5. Pereira, T., Moreira, A., Veloso, M.: Improving heuristics of optimal perception planning using visibility maps. In: Autonomous Robot Systems and Competitions (ICARSC), 2016 International Conference on, pp. 150–155. IEEE (2016)
6. Pereira, T., Veloso, M., Moreira, A.: Multi-robot planning using robot-dependent reachability maps. In: Robot 2015: Second Iberian Robotics Conference, pp. 189–201. Springer (2016)
7. Pereira, T., Veloso, M.M., Moreira, A.P.: Pa*: Optimal path planning for perception tasks. In: ECAI'16, pp. 1740–1741 (2016)
8. Portugal, D., Rocha, R.P.: Cooperative multi-robot patrol with bayesian learning. Autonomous Robots **40**(5), 929–953 (2016)
9. Potthast, C., Sukhatme, G.S.: A probabilistic framework for next best view estimation in a cluttered environment. Journal of Visual Communication and Image Representation **25**(1), 148–164 (2014). DOI 10.1016/j.jvcir.2013.07.006. URL http://dx.doi.org/10.1016/j.jvcir.2013.07.006