2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

On Distance Utility in the Exploration Task

Miroslav Kulich, Jan Faigl, and Libor Přeučil

***Abstract*— Performance of exploration strategies strongly depends on the process of determination of a next robot goal. Current approaches define different utility functions how to evaluate and select possible next goal candidates. One of the mostly used evaluation criteria is the distance cost that prefers candidates close to the current robot position. If this is the only criterion, simply the nearest candidate is chosen as the next goal. Although this criterion is simple to implement and gives feasible results there are situations where the criterion leads to wrong decisions. This paper presents the distance cost that reflects traveling through all goal candidates. The cost is determined as a solution of the Traveling Salesman Problem using the Chained Lin-Kernighan heuristic. The cost can be used as a stand-alone criterion as well as it can be integrated into complex decision systems. Experimental results for open-space and office-like experiments show that the proposed approach outperforms the standard one in the length of the traversed trajectory during the exploration while the computational burden is not significantly increased.**

I. INTRODUCTION

The exploration can be understood as a process of autonomous navigation of a mobile robot in an unknown environment in order to build a model of the environment. An exploration algorithm can be defined as an iterative procedure consisting of a selection of a new goal and a navigation to this goal. Such an algorithm is terminated whenever the defined condition (mission objective) is fulfilled. In this paper, the mission objective is building of a complete map of the environment. Besides, the usage of resources (e.g. the exploration time, the length of the trajectory) is optimized. In other words, the exploration strategy determines the next robot goal in each exploration iteration (one exploration step) with respect to the actual robot position, the current knowledge of the environment, and a selected optimization criterion.

Several exploration strategies have been proposed over last decades. The strategies differ in the way how candidates for the next goal are generated and in the criterion how the best candidate is selected. Yamauchi [1] introduced a frontier-based strategy that guides the robot to the nearest frontier, i.e. the boundary between a free and an unexplored space. It has been shown [2], [3] that this strategy produces reasonably short trajectories for graph-like environments with upper bound $O(|V|\log(|V|))$, where $|V|$ is the number of vertices of the graph. The authors of [4] discussed two simple heuristics improving Yamauchi's approach. The first one uses Voronoi diagrams to prefer exploration of the whole room in office-like environments before leaving it, while the second one repetitively re-checks whether the currently approached goal is still a frontier. When it is not, a new goal is determined. A strategy selecting the leftest candidate according to a robot position and orientation with a defined distance to obstacles is described in [5].

Other works generate several candidates in a free space (typically near to frontiers) and combine the distance cost (the utility evaluating effort needed to reach the goal) with other criteria. This concept has been introduced in [6] where measure $A(q)$ of an unexplored region of the environment, which is potentially visible from the candidate q, is combined with the distance cost $L(q)$ to get the overall utility of q:

$$g(q) = A(q)e^{-\lambda L(q)},$$

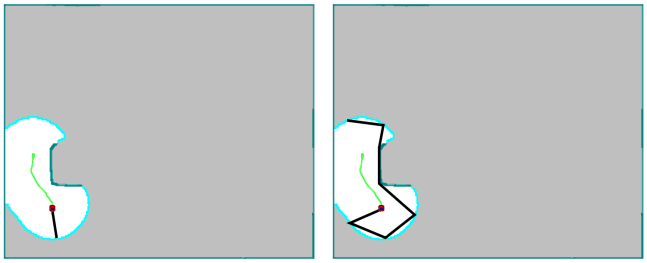
where λ is a positive constant. A utility of the next action as the weighted sum of the distance cost and expected information gain computed as a change of entropy after performing the action is presented in [7]. Another strategy taking into account the distance cost and the information gathered (based on the relative entropy) is introduced in [8] together with solid mathematical foundations. The strategy in [9] samples points near each candidate and filters samples according to selected criteria. The candidate with the highest number of samples that passed the filters is then chosen. Moreover, the localization utility can be integrated into the overall utility to prefer places traveling to them improves information about the robot pose [10]. Criteria forming the overall utility are not typically independent. General approach that reflects dependency among the criteria based on multi-criteria decision making is used in [11].

The aforementioned approaches evaluate the distance cost simply as the length of the trajectory from the current robot position to the next goal position. Such defined cost prefers candidates close to the robot without considering subsequent actions. In this paper, we present more sophisticated approach that is based on the observation that the robot should pass (or go nearby) all the goal candidates and define the distance cost for a candidate q as a minimal length of the path starting at the current robot position, continuing to the candidate q at first and then to all other candidates. We show that the introduced cost can reduce the exploration time significantly and leads to more feasible trajectories.

A similar approach is described in [12] where several exploration steps ahead are also considered. The state space of all possible paths consisting of several exploration steps is searched for the best alternative using the branch and bound algorithm. The branch and bound is a general technique that greedily searches relatively large state spaces without

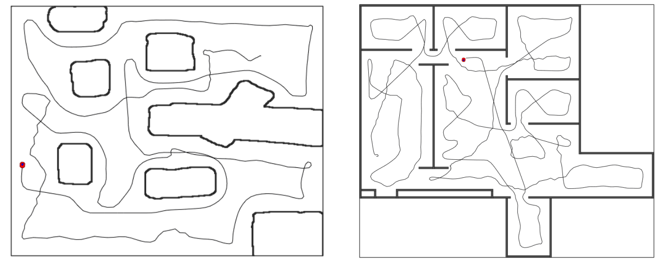
The authors are with Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technicka 2, 166 27 Prague 6, Czech Republic {kulich, xfaigl, preucil}@labe.felk.cvut.cz

978-1-61284-385-8/11/$26.00 ©2011 IEEE



(a) (b)

Fig. 1: Next goal selection by (a) the greedy approach, (b) taking into account all goal candidates.



(a) (b)

Fig. 2: Typical trajectories for the greedy approach.

a priory information about the solved problem. It can be therefore time consuming and the quality of found solutions heavily depends on the defined depth of pruning.

In our approach, we define the distance cost as the Traveling Salesman Problem (TSP). The problem formulation is described in the next section. Fast evaluation of the proposed distance cost is addressed by a heuristic algorithm for the TSP that finds a feasible solution quickly. Thus, the required computational time to solve the TSP is negligible in comparison to other parts of the exploration as it is shown in Section III presenting the experimental results. Finally, the concluding remarks are presented in Section IV.

II. Exploration with the TSP distance cost

Let the robot be equipped with a distance sensor with a fixed range (e.g. laser rangefinder) and the map the robot builds during exploration be modeled as the occupancy grid. The proposed exploration strategy is based on Yamauchi's frontier based approach. The key idea of the approach is to detect *frontier cells*, i.e. the reachable free grid cells (the cells representing free regions) that are adjacent with at least one cell that has not been explored yet. The *frontier* is a continuous set of frontier cells such that each frontier cell is a member of exactly one frontier. Once all frontiers are detected, the most appropriate frontier cell is selected as a new robot goal according to the defined criteria. This process is executed repeatedly at defined time steps until there is a frontier cell reachable by the robot.

As mentioned above, the current approaches compute the distance cost as the length of the path from the current robot position to the next goal, which can lead to selection of an inappropriate goal. An example of such a selection is demonstrated in Fig. 1a). In the shown situation, the robot moves down (the trajectory represented by the green curve) and a new goal has to be determined. The greedy approach selects the nearest frontier cell using the path showed as the black straight-line segment. It is obvious that in this situation a much better selection is to travel to the left first and then continue as it is illustrated in Fig. 1b).

Unfortunately, the described situation is not rare, and therefore the greedy approach produces superfluously long trajectories, see Fig. 2. To avoid this behavior, we propose a more informed approach to the distance cost using the TSP distance cost and consisting of two steps.

At first, frontier cells are filtered to get a set of representatives approximating the frontier cells such that each frontier cell is detectable by the robot sensor from at least one representative. This guaranties that all frontiers will be explored (i.e. it will be detected whether frontier lies in a free space or in any obstacle) after visiting all representatives. An algorithm for selection of representatives based on k-means is depicted in Algorithm 1.

Algorithm 1: Representatives selection for occupancy grids

Input: $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ - the set of frontiers
Input: D - the range of the used sensor (in grid cells)
Output: $\mathcal{R} = \{r_1, \ldots, r_m\}$ - the set of representatives

$\mathcal{R} = \{\}$
foreach $\mathcal{Q}_I$ **do**
  Set an appropriate number of representatives:
$$N = 1 + \frac{|\mathcal{Q}_I|}{2D}$$
  Find N means using k-means clustering:
$$\mathcal{A} = \texttt{k-means}(\mathcal{Q}_I, N)$$
  $\mathcal{R} = \mathcal{R} \cup \mathcal{A}$

Having the representatives, the second step is to decide in which order they will be visited with respect to the minimal length of the traveled trajectory. Let the robot position be s_0, the set of representatives $\mathcal{S} = \{s_1, \ldots, s_n\}$, and $d(a, b)$ denote the length of the path between cells a and b. The aim is to find a permutation $\Pi = \{\pi_1, \ldots, \pi_n\}$ of $\mathcal{I} = \{1, \ldots, n\}$ such that $d(s_0, s_{\pi_1}) + \sum_{i=1}^{n-1} l(s_{\pi_i}, s_{\pi_{i+1}})$ is minimal over all permutations of $\mathcal{I}$. The representative s_{π_1} is then selected as the next robot goal.

The problem to find the best permutation is similar to the TSP that is known to be NP-hard [13]. While finding an optimal solution of the TSP can be computationally demanding, there are many approximate algorithms. One of the most powerful ones is the Chained Lin-Kernighan heuristic, which gives near-optimal results (up to 1% of the optimum) in a reasonable time [14]. The TSP can be defined on a graph $G(V, E)$, where V is a set of vertices and E are edges connecting the vertices and representing the connection cost. The objective is to find a closed tour with the minimal cost connecting all vertices in V.

A straightforward approach to formulate the permutation problem as the TSP is to construct $G(V, E)$, where V is the set of representatives, and E is the set of the all shortest paths between the representatives. E can be computed by n calls of Dijkstra's algorithm on the adjacency graph of cells in the occupancy grid. Note that the TSP is formulated to find the best closed tour on the graph while a sequence of representatives ending in an arbitrary representative is requested for the distance cost. This discrepancy can be addressed by adding a fictive vertex s_∞ to V together with edges to all other vertices in V, whereas $d(s_\infty, s_0) = 0$ and $\forall i \in \{1, n\} : d(s_\infty, s_i) = \omega$, and ω is a large number (i.e. larger than the longest possible tour). This ensures that the TSP solver finds a solution where s_0 and s_∞ are neighbors in the tour. A solution of the permutation problem is found as a part of the tour starting from the s_0 and removing s_∞ and both its adjacent edges.

Algorithm 2: Frontier based exploration with the TSP distance cost.

```
repeat
    Get the updated map built from sensor readings
    Detect all frontiers from the actual map
    Select representatives
    Build the graph G(V, E)
    Solve the TSP on G(V, E)
    Set the neighbor of s_0 other than s_∞ as the next
    goal
until accessible frontier found
```

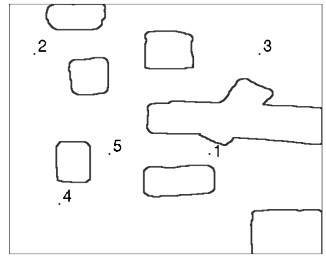
If the TSP distance cost is the only criterion for the goal selection it can be used as described above. An overview of the exploration procedure with this cost is depicted in Algorithm 2. In the case, the distance cost is combined with other costs (localizability, information gain) like in [10] [15], evaluation of the TSP distance cost is needed for each representative. Therefore a solution of the TSP for each representative s_j has to be found in order to compute the distance cost. The cost for s_j is determined using the graph $G_j(V_j, E_j)$ that is constructed in the following way:

1) $V_j = \{s_1, \ldots, s_n, s_\infty\}$, i.e. V_j does not contain s_0.
2) $d(s_\infty, s_j) = 0$ and $\forall i \in \{1, n\} \setminus \{j\} : d(s_\infty, s_i) = \infty$.
3) Costs of all other edges represent shortest paths between adjacent vertices.

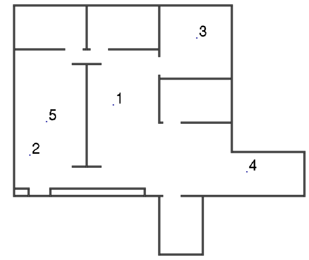
In other words, G_j is constructed similarly to the previous case, however s_j has the role of s_0. The distance cost for s_j is simply computed as the sum of $d(s_0, s_j)$ and the length of the sequence created from the TSP result in the similar way as in the previous case.

III. EXPERIMENTAL RESULTS

Performance of the proposed distance cost has been evaluated and compared to the standard greedy approach in two types of environments in simulations using the Player/Stage framework [16]. The first one is an open space represented by the *cave* map, which has been scaled to 25×20 m. The second one is an office-like environment represented by a map of the Autonomy Lab (*autolab*) scaled to 35×35 m. The environments are visualized in Fig. 3. Five positions where the robot starts the exploration have been chosen for each environment as shown in Fig. 3 and described in Tab. I.



(a) Cave



(b) Autolab

Fig. 3: Testing environments. The numbers correspond to the starting positions presented in Table I

All experiments were performed within the same computational environment: a workstation with the Intel®Core2 Duo CPU E6850 at 3 Ghz, 4 GB RAM running Sabayon 5.2 operating system with the Linux kernel 2.6.35. The algorithms have been implemented in C++ as client programs for the Player/Stage in version 3.0 and compiled by the GCC 4.4.2 with -O2 optimization flag. Simulation of the Pioneer 2DX robot equipped with SICK LMS200 with 180° field of view has been used as the robotic platform, while the occupancy grid with cell size 0.1×0.1 m has been chosen to represent the working environment. VFH+ algorithm [17] implemented in the Player has been used to control the robot motion and to avoid obstacles. The TSP solver used is the Chained Lin-Kernighan heuristic from the Concorde package [13].

TABLE I: Description of robot start positions in testing environments. The positions are in meters, the orientation of the robot is 0° for all positions.

Map	1	2	3	4	5
Cave	$[16, 8]$	$[2, 16]$	$[20, 16]$	$[4, 4]$	$[8, 8]$
Autolab	$[12, 18]$	$[2, 12]$	$[22, 26]$	$[28, 10]$	$[4, 16]$

The algorithm for the new goal selection (i.e. the body of the loop in Algorithm 2) is run every 1000 ms. It means that a new goal can be selected before the old one is reached. Moreover, the sensor range has been limited to 2, 3, and 5 meters. For each experimental setup consisting of the map, the starting robot position and the range, 30 runs have been performed for both the greedy approach and the proposed distance cost, which gives 1800 experiments in total. The particular experiment run took from 4 minutes for the *cave* map with 5 m range up to about 15 minutes for the *autolab* map and 2 m sensor range.

The experimental results are depicted in Tab. II for the *cave* map and in Tab. III for the *autolab* map. The solution quality is measured as the ratio of $avg_{TSP}/avg_{Greedy} \cdot 100\%$. Furthermore, the best found solutions of the proposed algorithm are shown in Fig. 6 and Fig. 7. The results show that the proposed TSP based approach outperforms the greedy selection in all cases. Generally speaking, the best improvement is achieved for smaller sensor ranges. The only exception is for the open space and high sensor range where the trajectories generated by the TSP approach can be shorter up to 72% comparing to the greedy approach. The greedy approach is not able to explore the space systematically, it leaves some places unexplored and they have to be visited later, which is not the case of the TSP based approach. Another interesting observation is that the standard deviation for the TSP case is significantly smaller than for the greedy approach. It is an expected result, because the TSP is more robust to small local changes in representatives' positions as demonstrated in Fig 5.

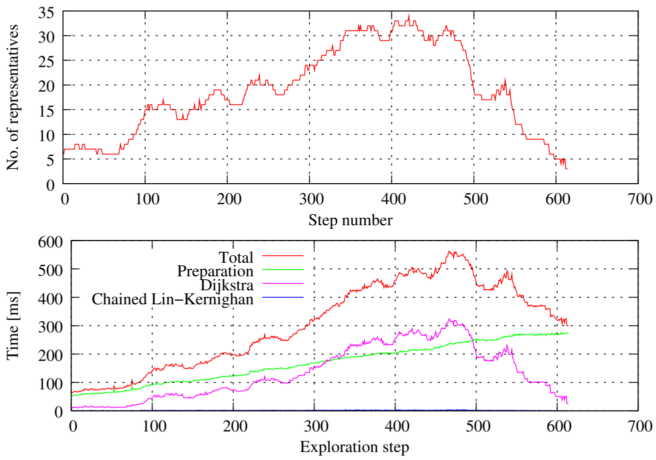


Fig. 4: The number of frontiers (upper) and the required computational time for the particular parts of the algorithm (bottom) during the exploration.

The required computational time of particular parts of the exploration algorithm in the *autolab* map and 2 m range is shown in Fig. 4. The blue curve, almost identical to the x-axis, denotes the computational time of the TSP solver. Regarding the times, the solution time of the TSP is negligible to other parts of the exploration algorithm.

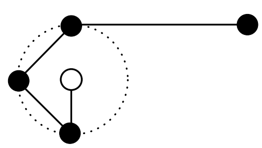


Fig. 5: If several goal candidates (the black disks) are in the similar distance to the robot and another one is far enough, small changes in goal candidates' positions do not change the shape of the TSP solution and thus the next goal will be preserved.

IV. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper, we present a novel approach to determine the distance cost in the exploration task. The key idea of the proposed approach is to select an appropriate set of goal candidates (representatives), which are expected to be visited by the robot. Then, the near optimal tour connecting all the representatives is found from which the next robot goal is determined. Although the introduced evaluation of the distance cost is primarily intended as a standalone criterion, a variant of the cost to be used as one of many criteria in complex systems has been presented as well.

A huge set of experiments has been performed in two different environments. These show that the presented method provides better results then the widely used greedy approach. The approach was presented for occupancy grids as the working environment representation. However a modification for a geometrical representation is straightforward.

B. Future Works

To the best of our knowledge there is no comprehensive comparison of exploration strategies in literature. Some attempts were made for example in [4], [11], [18], and [19]. Unfortunately, experiments in these papers are performed in different environments, with different robots and sensors and the number of experimental runs is relatively small. Moreover, the description of the experimental setup is not complete in many cases, which does not allow to repeat and compare described experiments. Therefore a detailed comparison of the current approaches (including the one presented in this paper), which can be repeated and enhanced by everyone, is one of our future goals.

The presented cost has been designed for a single-robot exploration. The next natural step is to extend the cost evaluation for the case of multiple robots that leads to solving a variant of the TSP called the Multiple Traveling Salesman Problem with MinMax criterion.

Finally, we would like to verify the results obtained in a simulation by experiments performed with real robots in the SyRoTek system [20].

V. ACKNOWLEDGMENTS

This work has been supported by the Ministry of Education of the Czech Republic, under the Project No. 6840770038.

REFERENCES

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*. IEEE Comput. Soc. Press, 1997, pp. 146–151.

[2] S. Koenig, C. Tovey, and W. Halliburton, "Greedy mapping of terrain," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 4, 2001, pp. 3594 – 3599 vol.4.

[3] S. Koenig, "Improved analysis of greedy mapping," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Las Vegas, NV*, 2003, pp. 3251–3257.

[4] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the Efficiency of Frontier-Based Exploration Strategies," in *Proc. of the Joint Conf. of the 41st Int. Symposium on Robotics and the 6th German Conference on Robotics*, Munich, Germany, Jun. 2010, pp. 36–43.

[5] Y. Mei, Y. hsiang Lu, C. S. G. Lee, and Y. C. Hu, "Energy-efficient mobile robot exploration," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, May 2006.

[6] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, Oct. 2002.

[7] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Proc. of Robotics: Science and Systems*, Cambridge, MA, USA, 2005.

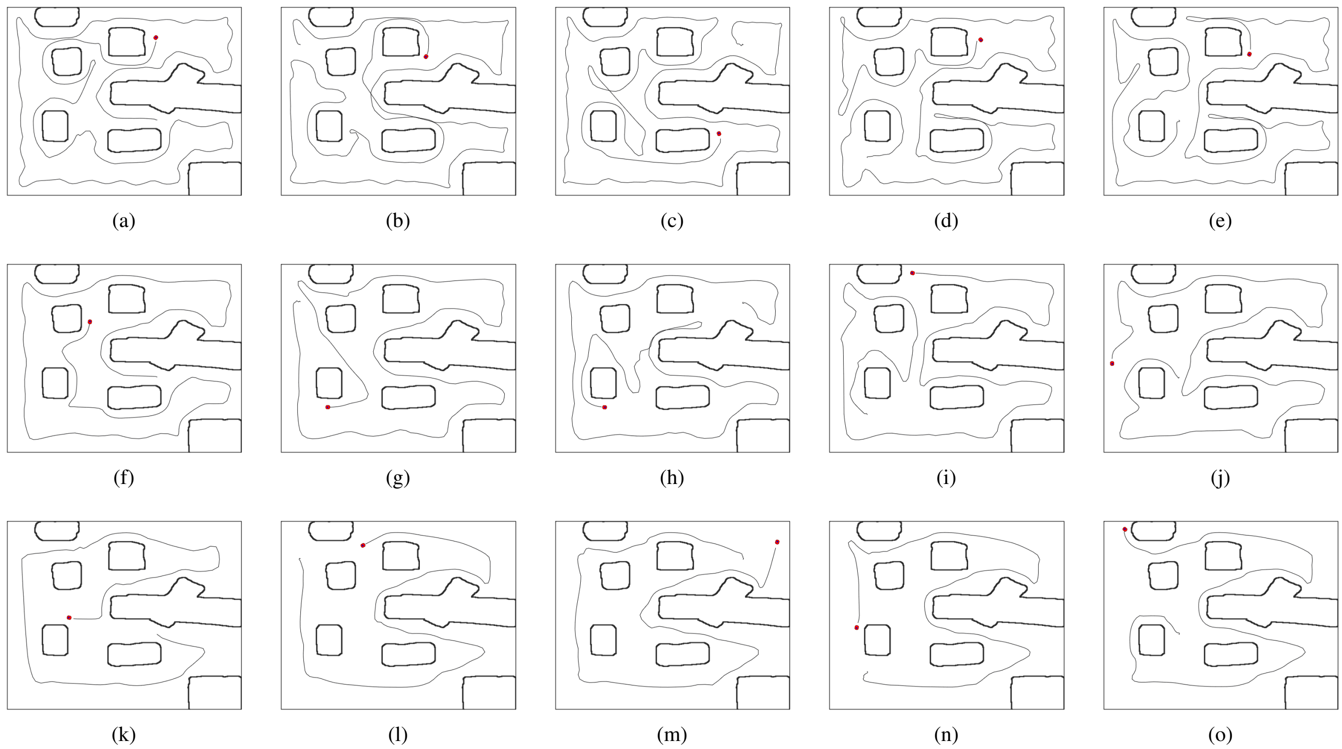


Fig. 6: The best results found with the TSP distance cost in the open space environment, the first row for $r = 2$, the second row for $r = 3$ and the third row for $r = 5$. The columns number equals to the position numbers.

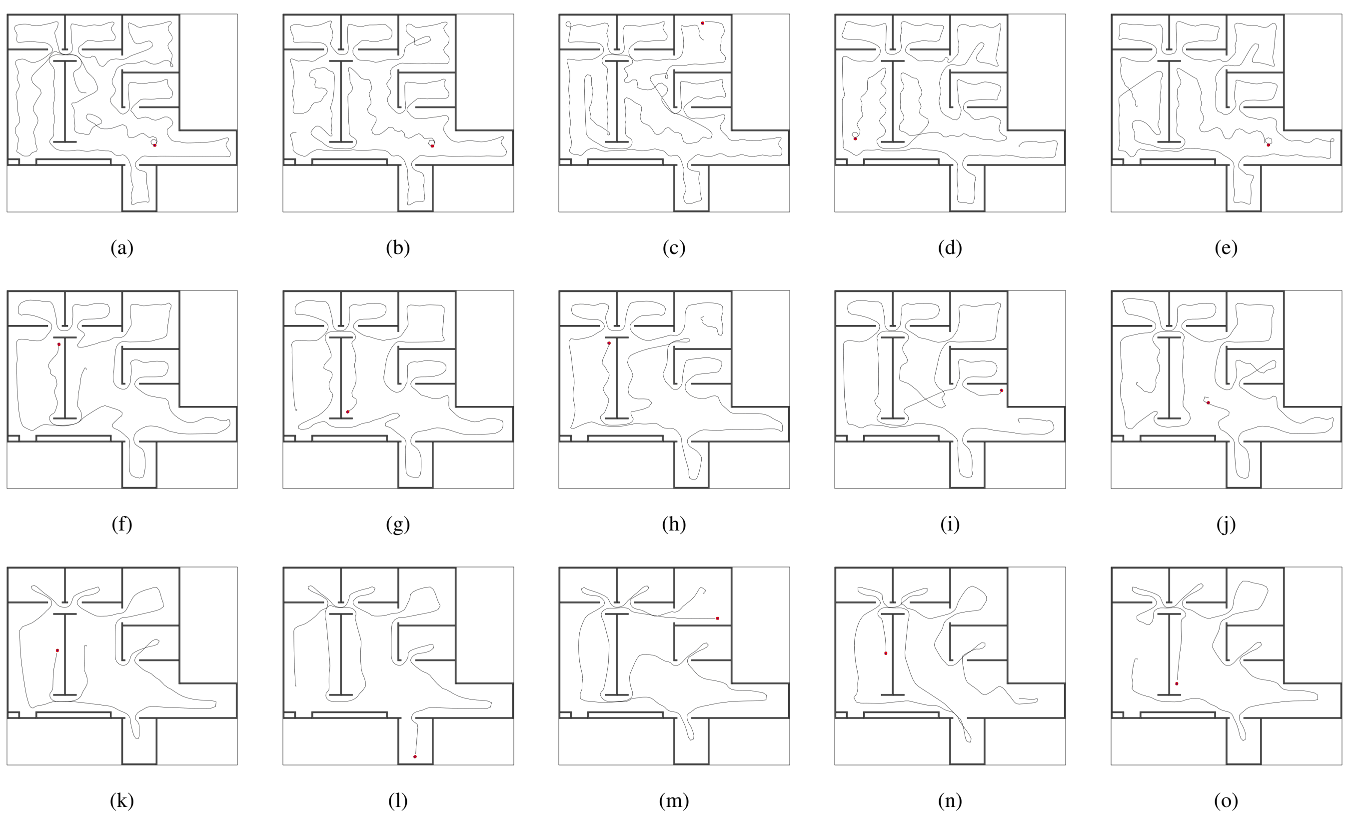


Fig. 7: The best results found with the TSP distance cost in the office-like environment, the first row for $r = 2$, the second row for $r = 3$ and the third row for $r = 5$. The columns number equals to the position numbers.

TABLE II: Comparison of the greedy approach and the proposed algorithm for cave environment. Pose numbers correspond to Table I.

Range [m]	Pose number	Greedy avg[m]	Greedy min[m]	Greedy max[m]	Greedy stdev[m]	TSP avg[m]	TSP min[m]	TSP max[m]	TSP stdev[m]	Ratio %
2.0	1	212.13	184.94	239.39	13.49	174.66	168.18	181.30	3.77	82.34
	2	200.41	184.39	227.08	10.32	177.09	171.62	185.95	3.76	88.36
	3	199.14	183.68	224.79	9.40	181.04	175.49	191.31	3.78	90.91
	4	206.98	187.35	226.92	9.99	184.86	172.13	195.83	5.65	89.31
	5	201.42	185.41	216.47	9.13	176.41	170.28	185.10	4.06	87.58
3.0	1	146.59	125.72	166.32	10.94	129.49	122.91	136.52	2.69	88.33
	2	129.84	108.03	160.68	10.72	125.53	118.34	137.34	5.16	96.68
	3	148.93	133.23	163.06	8.12	137.79	126.22	147.43	4.37	92.52
	4	148.21	121.36	183.64	14.16	132.06	125.69	143.60	5.87	89.10
	5	134.95	118.90	162.45	11.35	125.93	115.17	136.44	5.53	93.32
5.0	1	116.43	99.87	129.80	6.05	84.01	80.69	89.77	2.08	72.16
	2	108.49	90.12	125.40	10.30	78.45	77.50	79.79	0.67	72.30
	3	112.19	94.99	133.52	11.27	89.01	87.18	91.32	0.86	79.34
	4	110.69	99.64	124.45	10.95	88.32	80.66	104.97	5.87	79.79
	5	116.53	97.75	135.01	8.94	86.91	84.68	90.63	1.48	74.59

TABLE III: Comparison of the greedy approach and the proposed algorithm for autolab environment. Pose numbers correspond to Table I.

Range [m]	Pose number	Greedy avg[m]	Greedy min[m]	Greedy max[m]	Greedy stdev[m]	TSP avg[m]	TSP min[m]	TSP max[m]	TSP stdev[m]	Ratio %
2.0	1	345.57	307.87	383.82	19.37	301.76	293.63	314.45	5.97	87.32
	2	355.02	315.52	404.69	21.32	300.44	281.57	314.96	7.57	84.63
	3	351.16	320.93	385.65	16.98	311.48	302.10	320.74	4.97	88.70
	4	354.13	318.52	393.98	16.38	299.03	276.92	314.20	11.53	84.44
	5	343.95	311.52	378.76	17.37	295.06	277.32	308.13	8.27	85.79
3.0	1	244.08	222.14	274.80	13.36	217.71	212.03	226.03	3.86	89.20
	2	248.44	233.99	267.84	10.07	219.82	215.36	224.43	2.24	88.48
	3	247.51	222.44	276.87	13.92	231.25	221.22	237.13	3.36	93.43
	4	254.97	220.45	275.90	10.50	227.56	223.34	236.42	2.90	89.25
	5	251.67	235.38	284.67	11.94	223.20	217.37	234.58	4.78	88.69
5.0	1	167.72	151.19	192.29	10.46	159.82	156.45	163.40	1.72	95.29
	2	165.77	150.70	183.28	9.71	157.57	151.89	171.67	4.46	95.06
	3	182.63	164.36	198.81	10.12	165.42	162.81	168.11	1.56	90.58
	4	187.53	168.57	199.01	8.13	167.36	159.17	172.70	4.46	89.25
	5	163.26	147.85	185.09	10.93	153.77	149.32	167.67	5.12	94.19

[8] F. Amigoni and V. Caglioti, "An information-based exploration strategy for environment mapping with mobile robots," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 684 – 699, 2010.

[9] P. M. Newman, M. Bosse, and J. J. Leonard, "Autonomous feature-based exploration." in *IEEE Int. Conf. on Robotics and Automation*, Taiwan, Sep. 2003.

[10] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *IEEE/RSJ Int. Conf. on Intelligent Robots and System*. IEEE, 2002, pp. 534–539.

[11] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for an autonomous mobile robot," in *Proc. of the Fourth European Conference on Mobile Robots*. KoREMA, 2009, pp. 259–264.

[12] B. Tovar, L. Muñoz-Gómez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy, and S. Hutchinson, "Planning exploration strategies for simultaneous localization and mapping," *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 314 – 331, 2006.

[13] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. (2010, Sep.) Concorde TSP Solver.

[14] D. Applegate, W. Cook, and A. Rohe, "Chained Lin-Kernighan for large traveling salesman problems," *Informs J. on Computing*, vol. 15, no. 1, pp. 82–92, 2003.

[15] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, "A practical, decision-theoretic approach to multi-robot mapping and exploration," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 4, Oct. 2003, pp. 3232 – 3238 vol.3.

[16] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proc. of the 11th Int. Conf. on Advanced Robotics*, 2003, pp. 317–323.

[17] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1998, pp. 1572–1577.

[18] N. Basilico and F. Amigoni, "On evaluating performance of exploration strategies for an autonomous mobile robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems*, 2008.

[19] F. Amigoni, "Experimental evaluation of some exploration strategies for mobile robots," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 2818–2823.

[20] J. Faigl, J. Chudoba, K. Košnar, M. Kulich, M. Saska, and L. Přeučil, "SyRoTek - A Robotic System for Education," in *Proc. of Int. Conf. on Robotics in Education, Bratislava*. Bratislava: Slovak University of Technology in Bratislava, 2010, pp. 37–42.