Algorithmic Generation of Path Fragment Covers for Mobile Robot Path Planning

Jan Willemson[1] and Maarja Kruusmaa[2]

[1] Tartu University, Department of Computer Science,
Liivi 2, Tartu, Estonia, jan@ut.ee
[2] Halmstad University, Electrical and Electronic Engineering,
BOX 823, Halmstad, Sweden, Maarja.Kruusmaa@ide.hh.se

Abstract. This paper describes a framework for mobile robot path planning in dynamic environments where the environments are represented by grid maps. The planning heuristics considered in the paper is to generate the minimal set of paths so that all the possible path fragments of two edges are covered. We show that the number of required paths is linear in the dimensions of the grid (thus making application of the approach realistic and scalable). The main contribution of the paper is the description of all the minimal covers by means of an efficient algorithm. We prove its correctness and conclude that there are $2^{(m-1)(n-2)+(m-2)(n-1)}$ minimal path fragment covers in an $m \times n$ grid.

1 Introduction

Mobile robot applications often require repeated traversal in a changing environment between predefined start and goal points. For example, a mobile robot could be used to transport details and sub-assemblies between a store and production lines; this task implies repeated traversal between the store and the production cells. A mobile robot can also be used for surveillance; this task implies visiting certain checkpoints on a closed territory in a predefined order.

Real environments where these kind of mobile robots have to operate are dynamic by nature. From the point of view of mobile robot navigation, it means that unexpected obstacles can appear and disappear. The nature and density of the obstacles is usually unknown or too difficult to model.

At the same time, a mobile robot in a dynamic environment has to fulfill its assignment as fast and safely as possible. This means choosing paths between target points that are most likely unblocked and where the robot does not spend too much time maneuvering between obstacles.

A straightforward approach in robotic applications has been to choose the shortest paths to the goal. However, in dynamic environments one needs a more flexible approach as some previously satisfactory path may suddenly become inconvenient to traverse. When such a situation occurs, the robot needs a strategy for planning a new path, preferably completely different from everything it has tried in past.

Our basic approach for building such a strategy is described in [5]. The environment is represented by means of a rectangular grid (see Subsection 3.1). The paths are planned on the grid and information about the best paths (e.g. in the sense of traversability) is stored in a case base memory. If the robot needs a new path, it may try to use its knowledge about previously best paths. If this attempt fails, a new path has to be generated. As the grid map is mostly more or less imprecise and the environment is changing, we do not try to compile paths from parts previously known to be good, but rather try to generate and evaluate the path as a whole.

In this paper we concentrate on finding a good path generation mechanism that generates paths, as dissimilar as possible, from a given start to a given goal thus permitting the robot to try as many alternative approaches as possible in its search for better paths. The rest of the paper is organized as follows. Section 2 presents current status of the research, Section 3 introduces the idea of path fragment covers, Section 4 discusses the cardinality of such covers. Section 5 presents the main contribution of the paper – an algorithm for generating all minimal path fragment covers – and Section 6 draws some conclusions together with directions for future work.

2 State of the art

In the context of mobile robotics, path planning means finding a collision-free path from start to goal (see e.g. [6] for an overview of intelligent robotics and path planning techniques).

Very few research studies reported so far consider the problem of path selection in dynamic environments [2, 1]. Unlike these approaches, we do not assume that the structure of the environment is known a priori. Therefore our approach is applicable also in cases where very little is known about the environment and where the structure of the environment may change.

In our own previous work, supported by extensive tests in simulated environments as well as on real robots in real environments, we have used a heuristic algorithm for generating new innovative paths [4]. The tests have revealed that the robot is able to adapt quickly to the changes in the environment. As a result, risk of collisions, time delays and traveled distances are minimized.

However, the tests also pointed out the shortcomings of our approach. We observed that many paths that the heuristic algorithm generated were similar to each other. It also appeared that sometimes the robot got trapped to local minima – some paths that would have been easy to follow were never generated.

The first attempt to overcome this problem is made in [5]. In that paper, first a Hausdorff-like [3] distance measure between the paths is defined. Then, a construction is given for a seed path set such that for every possible path from start to goal there exists a path in the seed set that is not more than of distance δ away from the original path. This corresponds to covering the whole path space with path balls of radius δ. In this way we guarantee that when traversing all the seed set paths, a close match is traversed for every possible path.

It is proved in [5] that for a $m \times n$ grid the seed path set can not be made smaller than

$$\binom{\left\lfloor \frac{m}{2\delta+1} \right\rfloor + \left\lfloor \frac{n}{2\delta+1} \right\rfloor}{\left\lfloor \frac{m}{2\delta+1} \right\rfloor}.$$

The size of the seed set is reasonable for relatively large values of the similarity threshold δ, but when this threshold is decreased, the seed set cardinality increases very fast.

On the other hand, while being distant with respect to the distance measure, many paths in the constructed seed set overlap significantly. This consideration leads to another kind of interpretation for space covering introduced in the current paper. We will not try to generate a close match for every possible path (as implied by [5], this is hopeless anyway), but rather divide the whole area into small fragments and try to design such path sets that cover all the fragments. We will see that this approach leads to a much more scalable solution, namely the number of required paths is linear in grid dimensions.

A crucial question is how to choose the fragments to consider. The simplest solution would be to take all the grid points and select a path set to cover all of them. This idea does not work well for the following reason. If for example $m \geq n$, the minimal path set covering all the grid nodes contains $n + 1$ paths organized as shown in Figure 1 for $m = 4$, $n = 3$. If an obstacle occurs on the left boundary of the grid, many paths will be disregarded as unsuitable and as a result, many grid points never get traversed.

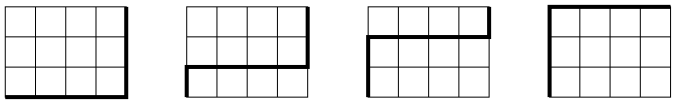

Fig. 1. Node cover for $m = 4$, $n = 3$

3 Covering path fragments

3.1 Grid and paths

Let $[a, b]$, $a \leq b \in \mathbb{Z}$ denote the set $\{a, a+1, \ldots, b\}$. We will consider a robot moving on a rectangular grid $[0, m] \times [0, n]$ allowing only right and up moves. We will also refer to such a grid as $m \times n$ grid, although it in fact contains $(m+1) \times (n+1)$ points.[3] The robot starts from the point $(0, 0)$ (which we consider to be the bottom left corner of the grid) and it must reach the point (m, n).

Definition 1. *By a* path *on the grid* $[0, m] \times [0, n]$ *we mean a sequence of grid points* $((x_0, y_0), (x_1, y_1), \ldots, (x_{m+n}, y_{m+n}))$ *such that*

1. $x_0 = y_0 = 0$, $x_{m+n} = m$, $y_{m+n} = n$;
2. *for each* $i \in [0, m+n-1]$, *the condition*

$$(x_i = x_{i+1} \ \&\ y_i + 1 = y_{i+1}) \vee (x_i + 1 = x_{i+1} \ \&\ y_i = y_{i+1})$$

holds.

The set of all paths on the grid is denoted by $\mathcal{P}_{m,n}$

3.2 Path fragment cover

In order to ensure higher probability for every grid point to be potentially traversed we need a more complicated condition than just the requirement to visit all the grid points. In this paper we use the requirement to cover all the *path fragments of length* 2 instead.

Definition 2. Path fragment of length 2 *(or* path fragment *for short) is a sequence of grid points* $((x_1, y_1), (x_2, y_2), (x_3, y_3))$ *where one of the following holds:*

- $x_1 = x_2 = x_3 \ \&\ y_1 + 1 = y_2 = y_3 - 1$;

[3] This is mainly due to visual reasons as the graphical representations of the grids used throughout this paper consist of $m \times n$ squares.

- $x_1 + 1 = x_2 = x_3 - 1$ & $y_1 = y_2 = y_3$;
- $x_1 + 1 = x_2 = x_3$ & $y_1 = y_2 = y_3 - 1$;
- $x_1 = x_2 = x_3 - 1$ & $y_1 + 1 = y_2 = y_3$.

The four possible path fragments are depicted in Figure 2 and later on they will be called path fragments of type one, two, three and four, respectively.

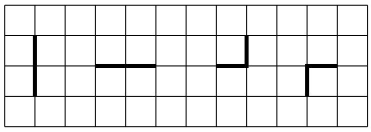

Fig. 2. Path fragments of length 2

Definition 3. *A subset $X \subseteq \mathcal{P}_{m,n}$ is called a* path fragment cover *if for every possible path fragment (c_1, c_2, c_3) there exists a path $P \in X$ such that $c_1, c_2, c_3 \in P$.*

Locally speaking, there are (at most) four different possibilities to traverse an inner grid point. Choosing path fragment cover as our seed path set guarantees that all these four possibilities are (at least theoretically) tried.

Now we can state the main problem of the current paper.

> Find the minimal size of a path fragment cover for a grid $[0, m] \times [0, n]$. Describe all the path fragment covers of minimal size and design the means for generating them.

4 Minimal path fragment cover

Definition 4. *Denote by $\tau(m, n)$ the minimal cardinality of path fragment cover of a $[0, m] \times [0, n]$ grid.*

Theorem 1.

$$\tau(m,n) = \begin{cases} 1, \text{ if } m = 0 \vee n = 0; \\ m + n, \text{ if } (m = 1 \,\&\, n \geq 1) \vee (n = 1 \,\&\, m \geq 1); \\ 2m + 2n - 2, \text{ if } m, n \geq 2. \end{cases}$$

Proof. The proof of the first two cases is trivial. In case $m, n \geq 2$ we first consider the left boundary of the grid (without the point $(0, 0)$). We say that a path fragment of length 2 *leaves the left boundary* if it has the form $((0, a), (1, a), (b, c))$, where $a \neq 0$. Now we compute the number of such fragments. It is clear that only the fragments of types two and three can leave the left boundary. For each one of the points $(0, 1), (0, 2), \ldots, (0, n-1)$ there exist fragments of both types that begin in these points (note that here we use the assumption $m \geq 2$). From the point $(0, n)$ only the fragment of type two can begin; altogether we get $2(n-1)+1 = 2n-1$ path fragments leaving the left boundary.

Symmetrically, we define a path fragment to *leave the bottom boundary* if it has the form $((a, 0), (a, 1), (b, c))$ (where $a \neq 0$). Similar to above, we find that there are $2m-1$ path fragments leaving the bottom boundary.

It is clear that every path can leave the boundary of the grid only once and hence there must be at least $(2n-1)+(2m-1) = 2m+2n-2$ paths to cover all the path fragments. On the other hand the construction showing that $2m+2n-2$ paths are sufficient is quite simple. All the fragments of type one can be covered with $m+1$ paths. For covering all the still uncovered fragments of type two $n-1$ paths are enough. Finally, the remaining fragments of types three and four can be covered using $m+n-2$ paths in a "diagonal" way. Altogether we have $(m+1)+(n-1)+(m+n-2) = 2m+2n-2$ paths.

For the case $m = 4$, $n = 3$ this construction is depicted in Figure 3.

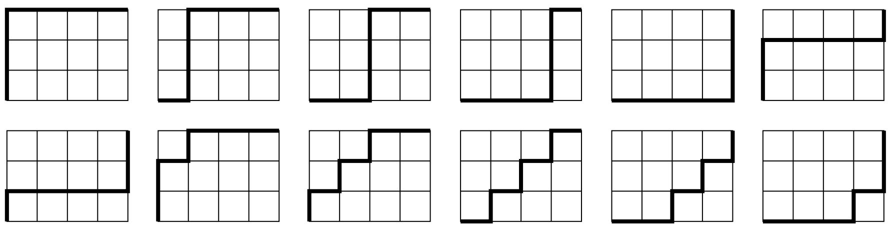

Fig. 3. Minimal cover for $m = 4$, $n = 3$

5 Generating the minimal covers

The last construction given in the proof of Theorem 1 and shown in Figure 3 will later be referred to as *trivial cover*. It is easy to see that generally (i.e. when $m \geq 2$, $n \geq 2$ and $m+n \geq 5$) this cover is not unique.

Intuitively we can also say that the trivial cover is not a very good one. Although we have all the fragments of length 2 covered, we still have not achieved our original goal of trying as different paths as possible; for example two two neighboring "diagonal" paths are rather close. In order to compare different minimal path fragment covers suitable measures must be introduced (e.g. based on the Hausdorff-like distance defined in [5]) and analyzed.

The analysis itself remains the subject for future research, but the rest of the current paper discusses the problem of generating all the minimal path fragment covers needed for the analysis.

We will show that a minimal path fragment cover can be fully described by its behavior on the following subset of grid edges:

$$\mathcal{S}_{m,n} = \{((i,j),(i,j+1)) : 0 < i < m,\, 0 < j < n-1\} \cup \\ \cup \{((i,j),(i+1,j)) : 0 < i < m-1,\, 0 < j < n\}.$$

Stating it otherwise, $\mathcal{S}_{m,n}$ consists of the edges of the grid that have no endpoint on the grid's boundary. Clearly, $|\mathcal{S}_{m,n}| = (m-1)(n-2) + (m-2)(n-1)$.

Next we "color" all the elements of $\mathcal{S}_{m,n}$ with one of two bits, i.e. 0 or 1. All the possible "colorings" can be considered as the elements of the function set

$$F_{m,n} = \{f : \mathcal{S}_{m,n} \to \{0,1\}\}.$$

Now we take a "coloring" function $f \in F_{m,n}$ and construct a minimal path fragment cover based on it. Note that in every such cover, there are four paths that contain no edges from the set $\mathcal{S}_{m,n}$ (e.g. in Figure 3 they are the paths number 1, 5, 8 and 12, respectively); all the other $2m+2n-6$ paths have at least one edge from this set. It follows from the proof of Theorem 1 that all these $2m+2n-6$ paths are identified by the first path fragment of length 2 that leaves the (left or bottom) boundary of the grid. Order these fragments in some order $r_1, r_2, \ldots, r_{2m+2n-6}$. It is the case for every $r_i = (c_i^1, c_i^2, c_i^3)$ that $(c_i^1, c_i^2) \notin \mathcal{S}_{m,n}$ and $(c_i^2, c_i^3) \in \mathcal{S}_{m,n}$.

In order to produce a suitable cover, we apply Algorithm 1.

Theorem 2. *Algorithm 1 produces a minimal path fragment cover.*

Proof. In order to prove the claim, we first argue that every edge of $\mathcal{S}_{m,n}$ gets traversed exactly twice during Algorithm 1. This claim is clear for the edges $((1,1),(1,2))$ and $((1,1),(2,1))$ and further we can use induction on $x_1 + y_1$ (where $e = ((x_1,y_1),(x_2,y_2))$).

Algorithm 1 Producing a minimal path fragment cover

Require: $f \in F_{m,n}$

Include the four compulsory paths into the cover
for $i = 1$ to $2m + 2n - 6$ **do**
  Take the fragment r_i and include all the edges from the grid's bottom left corner to the fragment r_i (the edges of fragment r_i included) into the new path.
  while the last edge of the current path part belongs to $\mathcal{S}_{m,n}$ **do**
    Let $e = ((x_1, y_1), (x_2, y_2))$ be the last edge of the current path
    if ($f(e) = 0$ and the edge e has not been traversed before) **or** ($f(e) = 1$ and the edge e has been traversed before) **then**
      Let the next edge on the path be $((x_2, y_2), (x_2, y_2 + 1))$
    else
      Let the next edge on the path be $((x_2, y_2), (x_2 + 1, y_2))$
    end if
  end while{Completing this loop we have reached the boundary of the grid}
  Complete the path to the top right corner of the grid
  Add the completed path to the cover
end for

Induction step is proved easily by considering different possible cases. For example, take an edge $e = ((x_1, y_1), (x_1, y_1 + 1)) \in \mathcal{S}_{m,n}$ that has no endpoint on the the left nor on the bottom boundary of the subgrid consisting of the edges of $\mathcal{S}_{m,n}$. Then both of the edges

$$e_1 = ((x_1 - 1, y_1), (x_1, y_1)) \quad \text{and} \quad e_2 = ((x_1, y_1 - 1), (x_1, y_1))$$

belong to $\mathcal{S}_{m,n}$, and by induction hypothesis they both get traversed two times. Consider further the case $f(e_1) = 1$ and $f(e_2) = 0$. The edge e is then traversed after the second traversal of e_1 and after the first traversal of e_2. This proves the induction step for this particular choice of edge e and function f, all the other cases are considered similarly.

Developing the given argument it is easy to see that all the fragments of length 2 (with no edge on the boundary of the grid) get traversed exactly once. E.g. in the example considered both the fragment consisting of edges e_1 and e, and the fragment consisting of edges e_2 and e get traversed. As we produced $2m+2n-2$ paths altogether, we must traverse each of the fragments of length 2 (with no edge on the boundary of the grid) exactly once and thus have a minimal path fragment cover.

In order to show that we actually achieve every possible cover exactly once, we give a construction of a suitable function f based on a minimal path fragment cover, which can be easily seen to induce the inverse mapping to the one of Algorithm 1.

Let us have a minimal path fragment cover. It contains four trivial paths, and all the other paths can be ordered by first boundary-leaving fragments $r_1, r_2, \ldots, r_{2m+2n-6}$ as done above. Let $P_1, P_2, \ldots, P_{2m+2n-6}$ be the corresponding ordering of the paths. Note that every edge $e \in \mathcal{S}_{m,n}$ is traversed by exactly two of these paths.

We define the function $f \in F_{m,n}$ as follows.

$$f(e) = \begin{cases} 0, & \text{if for the path } P_i \text{ (with smaller possible index } i) \\ & \quad \text{that traverses the edge } e, \text{ the next edge after} \\ & \quad e \text{ is upward,} \\ 1, & \text{otherwise.} \end{cases}$$

It is easy to see that the minimal path fragment cover built by Algorithm 1 and based on the defined function f results in the same cover as the one we began with.

Theorem 3. *There are* $2^{(m-1)(n-2)+(m-2)(n-1)}$ *minimal path fragment covers in an* $m \times n$ *grid.*

Proof. We showed above that minimal path fragment covers are in one-to-one correspondence with the elements of the set $F_{m,n}$. Hence their number is equal to

$$|F_{m,n}| = 2^{|\mathcal{S}_{m,n}|} = 2^{(m-1)(n-2)+(m-2)(n-1)}.$$

We illustrate the work of Algorithm 1 for the case $m = 4$, $n = 2$. Out of $2 \cdot 4 + 2 \cdot 2 - 2 = 10$ paths in every minimal cover there are 7 that belong to every one of these covers (they are exactly the paths that have no edge in $\mathcal{S}_{4,2}$). $2^{(4-1)(2-2)+(4-2)(2-1)} = 4$ variations of the remaining three paths are shown in Figure 4.

Time complexity of Algorithm 1 can be computed as follows. We need to generate $2m + 2n - 6$ paths, each path generation takes $m + n$ steps (including both steps on the grid boundary and the while-loop on lines 4–11). Hence, the overall time complexity is $O((m+n)^2)$. Note that practically the same time is needed just for the result printout and thus there can be no substantially faster algorithm for this particular task.

Space complexity of Algorithm 1 is low as well. We need $(m-1)(n-2)+(m-2)(n-1)$ bits to store the function f and additional $(m-1)(n-2) + (m-2)(n-1)$ bits to store information about the traversedness of edges. Hence, the overall space complexity is $O(mn)$.

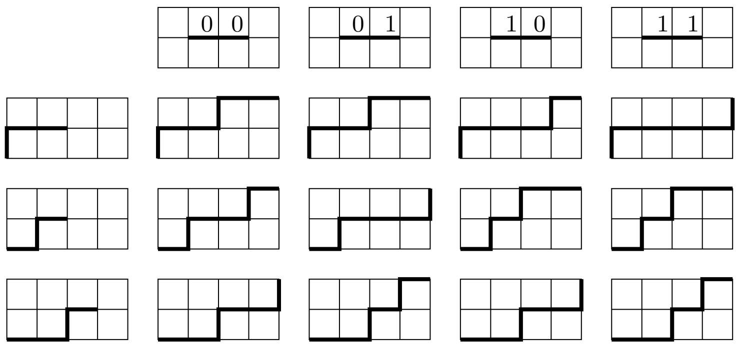

Fig. 4. Constructing all the covers for $m = 4$, $n = 2$

6 Conclusions and further work

In this paper we have presented a framework for mobile robot path generation in unknown and dynamically changing environment. Our approach was to represent the area as a grid map and design a special set of paths (called path fragment cover). Following the paths from this set, the robot traverses all grid points in all possible (up to four) ways.

We showed that the minimal number of paths required to cover $m \times n$ grid is in general case equal to $2m + 2n - 2$ and hence, such a solution is highly scalable. Still, minimal path fragment cover is not unique and the number of suitable covers grows exponentially. We described all of them by means of a very efficient algorithm and proved that the algorithm is correct.

Intuitively it seems that not all the possible minimal covers are not equally good. In order to determine the best one from them, it is necessary to define some quantitative measure on covers and try to optimize that. This measure should take into account the relative positions of all pairs of paths. For example, one could define the measure of minimal cover $\mathcal{C}_{m,n}$ as $\min_{P_1,P_2 \in \mathcal{C}_{m,n}} d_g(P_1, P_2)$, where d_g is an appropriate distance measure between the paths (see e.g. [5]), and maximize this quantity. At the moment, it is not clear at all how such a quantity behaves and where does it achieve its maximum, so this remains the subject for future research.

References

1. Karen Zita Haigh and Maria Manuela Veloso. Planning, execution and learning in a robotic agent. In *AIPS-98*, pages 120 – 127, June 1998.

2. H. Hu and M. Brady. Dynamic Global Path Planning with Uncertainty for Mobile Robots in Manufacturing. *IEEE Transactions on Robotics and Automation*, 13(5):760–767, October 1997.
3. D. Huttenlocher, D. Klanderman, and A. Rucklige. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.
4. Maarja Kruusmaa. *Repeated Path Planning for Mobile Robots in Dynamic Environments*. PhD thesis, Chalmes Univeristy of Technology, Gothenburg, Sweden, 2002.
5. Maarja Kruusmaa and Jan Willemson. Covering the Path Space: A Casebase Analysis for Mobile Robot Path Planing. In Max Bramer, Alun Preece, and Frans Coenen, editors, *Research and Development in Intelligent Systems XIX*, BCS Conference Series, pages 1–16. Springer, December 2002.
6. Robin R. Murphy. *An Introduction to AI Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 1st edition, 13 November 2000.