Any-angle Global Path Planning for Skid-Steered Mobile Robots on Heterogeneous Terrain

Submitted: 25th January 2016; accepted: 20th April 2016

Piotr Jaroszek

DOI: 10.14313/JAMRIS_2-2016/15

Abstract:
The paper is concerned with selection of the algorithm of path planning for a skid-steered mobile robot operating on heterogeneous terrain. Methods of path searching were reviewed and their applicability to particular kinematic structure of a robot was assessed. The Theta graph search algorithm was selected, because of its property of returning any-angle paths. Because in this method variable terrain type is not considered, necessary changes in algorithm structure were proposed to check homogeneity of the terrain. In order to enable choice of arbitrary optimization criterion, the model of cost dependent on terrain properties was introduced, which includes both longitudinal motion and turning. Operation of the modified algorithm with the introduced cost model was verified by means of simulation against A* reference algorithm often used in path planning tasks.*

Keywords: *spath planning, skid-steered mobile robot, Theta*, any-angle path planning, A**

1. Introduction

One of the most important tasks associated with autonomy of robot motion is global path planning. Every type of robot requires individual approach to this problem, because robot motion capabilities depend on kinematic structure of the mobile platform. Another equally important component of the global path planning is adequate representation of the environment. Most often the robot operates in heterogeneous environment, which can be characterized by diverse properties, and taking this fact into account is definitely beneficial in the path planning process. A desirable solution returns path optimal according to the adopted criteria and possible for realization by the robot in its workspace. There is no single method of path planning which would be appropriate for all tasks in robotics, and thus in order to get the best results for a given problem, the methods have to be modified.

In this article, the approach of optimal path planning is presented for one of the more frequently used types of mobile robots – the platform with non-steered wheels. To this end, the any-angle path planning method with original modifications that include terrain non-homogeneity and generic robot motion model was used.

2. Selection of Path Planning Method for a robot with Non-steered Wheels on Heterogeneous Terrain

There exist several methods of global path planning, whose usefulness varies depending on target application. The potential field methods [1] are fast, but they suffer from the local minima problem, the genetic methods have big potential [2], but are significantly slower and difficult in description and implementation. Both those groups are characterized by frequent lack of algorithm convergence, so sometimes the returned path is not optimal. Additionally, serious difficulties in representation of complex models of the environment and robot can be encountered. The alternative are graph search algorithms [3], where belong admissible non-heuristic algorithms, e.g. Dijkstra [4], and heuristic algorithms like A* [5]. The search space in those methods has to be discretized to the form of a weighted graph whose nodes and edges represent respectively available locations and possible movements between them. Moreover, the search algorithms make use of the strategy of choice of the search direction, which can be modified in order to achieve desirable motion behaviors that are possible for the chosen kinematic model of the robot.

In case of the graph search algorithms, the terrain where the robot operates is most often discretized to the grid of elements of identical size, based on which the graph is constructed. Weight of each edge of the graph depends in an unique way on properties of nodes which it connects. When the nodes represent locations, most often edge weight is equal to a distance between the nodes, though in general the travel cost does not have to be associated with length. By appropriately choosing search politics and cost models, it is possible to obtain solution of the optimal path planning task for any criterion. The essential problem which remains to be solved is how the path looks in the discretized terrain, which directly depends on graph representation, and the path appearance is often different than would be expected in continuous environment.

The path found on the terrain grid is optimal from the point of view of the graph, so it usually looks unrealistic and is not the shortest one in the real continuous environment. In [6], author presented analysis of the problem of path length dependency on various environment models and search methods on regular grids. Example of difference in the path length found with A* method, and the shortest one possible for the continuous environment is shown in Fig. 1.

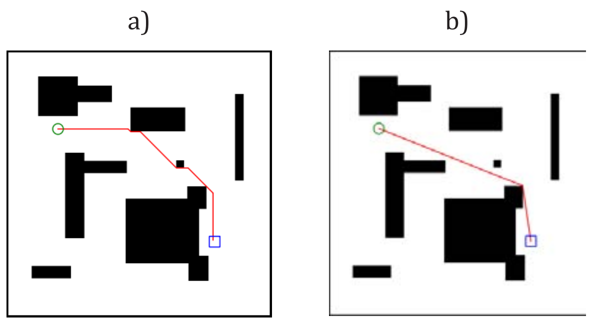

***Fig. 1.** The shortest path found with A* method in case of discrete environment (a) and in case of continuous environment (b)*

As one may notice, the path found by the A* algorithm (Fig. 1a), despite being optimal in the graph space (assuming path length as optimality criterion) does not reflect in a good enough degree the path which is optimal in the real continuous environment (Fig. 1b). The path, let alone it is longer than the shortest one in the continuous environment, comprises also several unnecessary turns which may introduce additional cost of robot motion. The robot will waste time and energy by turning in place, and time and energy affect the majority of optimality criteria traditionally used in robotics. It is possible to claim undoubtedly that for a skid-steered robot, turning in place has to be considered because of significant amounts of energy which are necessary for this maneuver. Therefore, the solution may consist in change of the environment representation or modification of the planning method to obtain paths more alike the real ones. The change of environment discretization, for example, to the visibility graph would solve the problem of path outlook, but it would also make representation of heterogeneity of the environment more difficult and would noticeably extend the computation time of the algorithm [7].

This problem can be also resolved by modification of the algorithm rather than the representation of the data. Example of this approach are any-angle path planning methods [6], that is, the methods for which the ultimate path outlook does not depend strictly on edges of the graph based on which the path was found.

Many approaches to this problem were proposed, from smoothing paths found by the A* to more complex modifications, that is, Theta* [7], Block A* [8], Field D* [9]. Based on [8] and [10] one may come to the conclusion that out of the previously mentioned methods, the Theta* algorithm will be the most appropriate for the mobile robot global path planning task. If path length is the chosen optimality criterion, then the found path is usually the shortest one and has smaller (or comparable) number of turns compared to other A* family algorithms. The Theta* algorithm yields to the other algorithms mainly as far as speed of computation is concerned, however, in the present case speed is not the most important factor. The important factor, besides optimality of found paths, is the possibility of representation of heterogenous terrain which directly affects the cost according to each of the assumed optimality criteria. Author believes that Theta* is one of the most suitable algorithms for the mobile robot global optimal path planning task because:

- it returns the optimal path if it exists,
- shape of the found path complies with the assumption of continuous environment,
- it works fine with discrete representation of terrain.

Authors in [11] presented generalization of the Theta* toward maps with non-uniform cost of each cell. The introduced modifications are general purpose in the sense, they do not take into account model of robot kinematics.

The aim of this work is modification of the Theta* method so as to solve the problem of finding optimal path for a robot with all wheels non-steered according to any cost model that includes longitudinal motion of a robot and its (pivot) turning on the known heterogeneous terrain.

3. Robot Model and Environment Representation

The map. The search space is represented in the form of a weighted graph with nodes as the admissible robot locations. The following assumptions concerning terrain map discretization were introduced:

- the terrain map is divided into square-shaped elements, which form the so-called occupancy grid,
- length of side of an individual element $l = 0.1$ m,
- the search space has the form of a graph with m nodes n_j $j=1,2,3...m$,
- each node n_j stores information about terrain properties $T(x,y) = \mu_{xy}$ and about location on the map (x,y) which it represents,
- the cost value is assigned to every edge depending on the properties of nodes connected by this edge.

The robot. The assumed mobile platform is equipped with non-steered or caster wheels and has the capability of:

- moving along a straight line,
- (pivot or in-place) turning through arbitrary angle.

It is assumed that during motion the robot does not turn along an arc, that is, forward motion and turning do not occur simultaneously. In the autonomous operation of this kind of robots it is favorable to avoid the combined motion, because the combined motion additionally increases possibility of wheel slip and other unpredicted motions. The point-to-point motion is the simplest to realize.

State of the robot at the time instant t is described as:

$$x = \begin{bmatrix} p_x^t \\ p_y^t \\ \theta^t \end{bmatrix}, \tag{1}$$

where p^t_x, p^t_y – are respectively x and y coordinates of robot position on the map, θ^t – robot orientation with respect to the Cartesian coordinate system of the map at the time instant t (Fig. 2).

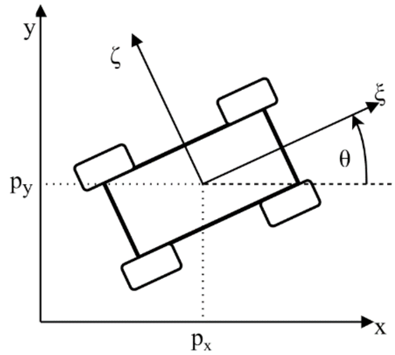

Fig. 2. Robot pose in map coordinates

The cost of robot motion can be represented in various ways, depending on the assumed criteria of path optimality. In order to allow arbitrary choice of the criteria, the total cost of motion C includes both longitudinal motion along a straight line and the pivot turning:

$$C = C_R + C_M, \tag{2}$$

where: C_R – turning cost and C_M – forward motion cost.

Motion along the path of length s can be divided into S elements of lengths Δs^i for which elementary costs C^i will be equal to:

$$C^i = C_R^i + C_M^i. \tag{3}$$

After assuming that the elementary length Δs^i is identical with the graph edge i, one may write that the total cost of path described on this graph is equal to:

$$C = \sum_{i=0}^{S-1}(C_R^i + C_M^i). \tag{4}$$

The cost of longitudinal motion and turning can be further determined based on terrain properties and the motion realized by the robot. Terrain property μ^i for the graph edge is equal to the value assigned to the node which was encountered earlier during robot motion. Use of only one node for the elementary motion along Δs^i for a dense grid introduces a small error, which was allowed for the sake of significant simplification of the cost calculation:

$$\begin{cases} \sum_{i=0}^{S-1}(C_R^i + C_M^i) \\ C_R^i = R(\Delta\theta^i, \mu^i)\,, \\ C_M^i = M(\Delta s^i, \mu^i) \end{cases} \tag{5}$$

where μ^i is the function parameter describing terrain property assigned to the start node of the i-th graph edge, R is the cost function for turning whose value depends on change of the angle of orientation of the robot $\Delta\theta^i$, M is the cost function for robot forward motion whose value depends on the travelled distance Δs^i. The turning cost for a robot with steered wheels can be much smaller than for the skid-steered robot. The μ^i value is equal to the property of the discrete element of coordinates X^i and Y^i on the map corresponding to the graph node deemed the start node for a given edge. The start node of the edge is defined by the search strategy and it depends on the previous position of the robot. After assuming that at the time instant t the robot has the state x^t and it starts motion from the node $n(p_x^t,p_y^t)$ along the edge i and ends motion at the node $n(p_x^{t+1},p_y^{t+1})$, one can write:

$$x^{\mathrm{i}} = x^{\mathrm{t}}, \tag{6}$$

$$\mu^i = \mu^t = T(p_x^t, p_y^t), \tag{7}$$

where $T(p_x^t,p_y^t)$ is terrain type property mentioned earlier. Thus, the weights assigned to edges depend directly on: nodes they connect, direction of robot motion and its orientation. In view of that, partial cost of robot motion between graph nodes can be written as:

$$C^i = R(\Delta\theta^i, \mu^t) + M(\Delta s^i, \mu^t), \tag{8}$$

where elementary length Δs^i and change of orientation $\Delta\theta^i$ with respect to terrain map are equal to:

$$\begin{cases} \Delta s^i = \sqrt{(p_x^i - p_x^{i-1})^2 + \left(p_y^i - p_y^{i-1}\right)^2}, \\ \Delta s^0 = 0 \end{cases} \tag{9}$$

$$\begin{cases} \Delta\theta^i = \theta^{\,i} - \theta^{\,i-1} \\ \Delta\theta^0 = \theta_{\,0} \end{cases}, \tag{10}$$

where $S > i > 0$ and for $i = 1$, the p_y^0 p_x^0 and θ^0 are values of the robot initial state.

Equation (8) can be transformed into general form of the partial cost as a function of the robot variable state:

$$\overline{C(x^{t-1}, x^t)} = R\left(x^{t-1}, x^t, T\left(p_x^{t-1}, p_y^{t-1}\right)\right) + M\left(x^{t-1}, x^t, T\left(p_x^{t-1}, p_y^{t-1}\right)\right). \tag{11}$$

Functions R and M can have arbitrary forms. For energy optimization of the path, they can be, for instance, models of energy consumption by robot drives during straight-line motion and during turning. Additionally, in order to keep the algorithm admissible, both functions have to be linear with respect to Δs^i and $\Delta\theta^i$, which are time dependent.

4. Global Path Planning with the Modified Theta* Method

Starting from a certain start node, the A* algorithm searches the state space graph, successively "closing" its "visited" nodes which are situated at the so-called frontier. During visiting the node n_j, the cost of path necessary to reach this node is calculated based on the cost function F:

$$F(n_j) = G(n_j) + H(n_j), \tag{12}$$

$$\begin{cases} G(n_j) = \sum_{j=1}^{k} g(n_{j-1}, n_j) \\ G(n_0) = 0 \end{cases} \quad k = 1,2,3\ldots, \tag{13}$$

where: $G(n_j)$ – is the cost from the start node to the n_j node, $g(n_{j-1}, n_j)$ – partial cost of robot motion between nodes n_{j-1} and n_j, l – number of nodes which form the path from n_0 to n_j, $H(n_j)$ – estimation of the cost of the remaining not-yet-found path from the node n_j to the target.

Nodes which are subsequently visited from the currently closed location (i.e., node) depend exclusively on the chosen search policy and most often they are the neighboring nodes. For a graph based on the occupancy grid, the algorithm – if the node is not at the map boundary – most often tries to visit from 4 to 8 neighbors of the current node (including possibilities of diagonal motion). Each node additionally stores information about its parent, from where it was visited. If at the moment of visiting the node, the total cost $F(n_i)$ is smaller than previously determined (the same nodes can be visited from different directions), the pointer to the parent of this node is updated. If the node considered target node becomes closed, then the path is found and it is generated based on the pointers to parents.

Assuming that the robot at a given time instant can be in one location only (a graph node), the introduced earlier partial cost can be now assumed as follows:

$$g(n_{i-1}, n_i) = \overline{C(x^{t-1}, x^t)}. \tag{14}$$

```
Main()
  open := closed := ∅;
  g(s_start) := 0;
  parent(s_start) := s_start;
  open.Insert(s_start, g(s_start) + h(s_start));
  while open ≠ ∅ do
    s := open.Pop();
    if s = s_goal then
      return "path found";
    closed := closed ∪ {s};
    foreach s' ∈ nghbr_vis(s) do
      if s' ∉ closed then
        if s' ∉ open then
          g(s') := ∞;
          parent(s') := NULL;
        UpdateVertex(s, s');
  return "no path found";
end

UpdateVertex(s, s')
  g_old := g(s');
  ComputeCost(s, s' );
  if g(s' ) < g_old then
    if s' ∈ open then
      open.Remove(s' );
    open.Insert(s', g(s') + h(s'));
end

ComputeCost(s, s')
  if lineofsight(parent(s), s') then
    if g(parent(s)) + c(parent(s), s') < g(s') then
      parent(s') := parent(s);
      g(s') := g(parent(s))+c(parent(s), s');
  else
    if g(s) + c(s, s') < g(s') then
      parent(s') := s;
      g(s') := g(s) + c(s, s');
end
```

Fig. 3. Pseudocode of the reference Theta [7]*

```
ComputeCost(s, s')
  if IsTerrainUniform(parent(s), s') then
    if g(parent(s)) + rot(parent(s), s' ) + lin(parent(s), s' ) < g(s')
then
      parent(s') := parent(s);
      θ(s') := θ(parent(s));
      g(s') := g(parent(s)) + rot(parent(s), s') + lin(parent(s),
s' );
  else
    if g(s) + c(s, s') < g(s') then
      parent(s') := s;
      θ(s') := θ(s);
      g(s') := g(s) + rot(s, s') + lin(s, s');
end

IsTerrainUniform(s, s')
  nodeList := bresenham(s,s');
  prev := ∅;
  foreach s' in nodeList
    if prev = ∅ then continue
    prev := s';
    if isnotuniform(s,s') return false;
  return true;
end

rot(s, s')
  return C_R(s, s');
end

lin(s, s')
  return C_M(s, s');
end
```

Fig. 4. Theta for the robot with non-steered wheels in the heterogeneous environment*

The main idea and difference of Theta* as compared to A* is determination of node parents based on mutual visibility by means of the line-of-sight check (the pseudocode shown in Fig. 3, described in detail in [7]). During searching the graph, the parent-child relations are updated for any successive nodes making the path that mutually "see" each other. Pointer to parent is set at every successive node to the furthest visible node which belongs to the path. This leads to a frequent situation where the parent is not neighbor to its child in the sense of being connected by the graph edge. Consequently, it is not possible to use the cost model like that directly in the Theta* algorithm, because partial value in the heterogeneous terrain is not identical along the whole path between parent and child. When the cost is not homogeneous during traverse, the visibility ceases to be a sufficient condition of optimality of the found path – replacement of the visibility condition with the terrain homogeneity condition becomes necessary (Fig.4). This condition is checked by iterative review of all cells lying on a line between the start point and end point under test (Fig. 5). Selection of the mentioned cells is carried out using the Bresenham algorithm [12]. If any of the cells is untraversable or type of terrain of the successive cells is not the same, then the terrain along the line connecting the chosen endpoints is heterogeneous. Otherwise, when the terrain is homogeneous, it is possible to use the derived earlier cost model because the μ parameter is constant.

In Fig. 4 necessary changes in the *ComputeCost* function within Theta* algorithm as well as additional formulas are shown, which include arbitrary models of cost for longitudinal motion and turning as well as terrain heterogeneity testing. Example longitudinal motion C_M and turning C_R cost models are shown in Fig. 6. In those models, the example cost of longitudinal motion depends on travelled distance and the cost of turning, on absolute value of change of the angle of robot orientation. The M and R quantities are auxiliary constants that modify values of the appropriate costs.

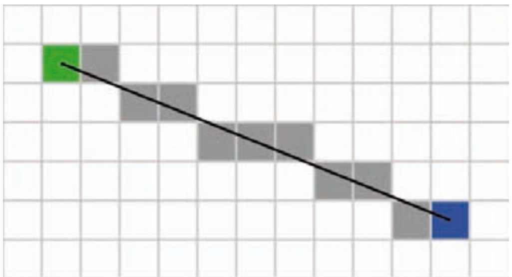

Fig. 5. Cells found using Bresenham algorithm for a terrain homogeneity check

```
C_M(s, s')
  return M * sqrt((py(s') – py(s))² + (px(s') – px(s))² ) * T(px(s), py(s))
end

C_R(s, s')
  return R * abs((θ(s')-θ (s)) * T(px(s), py(s));
end
```

Fig. 6. Example motion cost models for longitudinal motion C_M and turning C_R

5. Simulation Results

A number of simulations of path planning using the proposed Theta* modification and the standard version of A* with the same models of cost and environment in order to compare quality of the obtained results were conducted. On a map of dimensions 512 x 512 cells (51.2 m x 51.2 m) shown in Fig. 7, 100 trials of searching of paths for each of three different sets of terrain parameters were carried out. Values of terrain properties for three variants are shown in Table 1 and they are assigned to cells of the map according to the colors shown in Fig. 7.

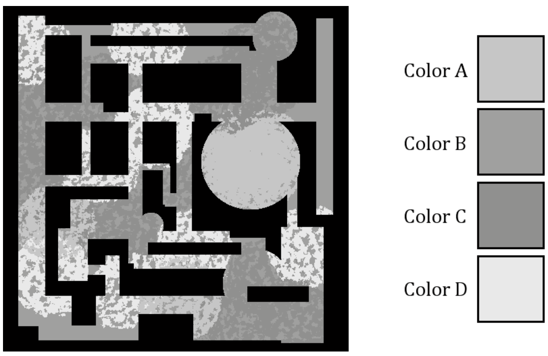

Fig. 7. A map used for algorithm testing

Values of R and M constants in the cost models are assumed respectively R = 5 and M = 1. Example paths found are shown in Fig. 8. The noticeable large number of turns in the path found by the modified Theta* is caused by high diversity of terrain and by not including costs associated with acceleration and stopping. Summary of results from 300 trials in total is shown in Tables 2 and 3. Gain in cost value and path length in case of Theta* as compared to the reference version of A* is presented.

Table. 1. Sets of terrain properties used for algorithm testing and their assignment to areas on the map

Color	Set of μ			Traversability
	#1	#2	#3	
A	0.1	0.1	0.5	Yes
B	0.3	0.4	2.5	Yes
C	0.5	0.8	5	Yes
D	0.8	1.6	12	Yes
Black	N/A	N/A	N/A	No

a) b)

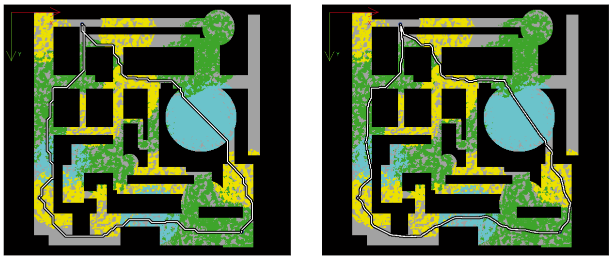

Fig. 8. Example paths found by A (a) and Theta* (b)*

Table. 2. Summary of results of path cost

	Path cost		
Set of μ	Average gain	Maximum gain	Minimum gain
#1	5.69%	10.73%	1.28%
#2	6.84%	12.84%	2.24%
#3	7.22%	12.07%	2.67%

Table. 3. Summary of results of path length

	Path length		
Set of μ	Average gain	Maximum gain	Minimum gain
#1	1.84%	8.86%	-20.63%
#2	1.76%	8.30%	-13.30%
#3	1.51%	8.13%	-8.83%

6. Conclusion

Use of the appropriate motion cost models enables solution of the global path planning task during operation of real mobile robots according to arbitrary criterion which depends on heterogeneous terrain. Adjustment of the any-angle algorithm to the considered robot model results in reduced cost of the found path as compared to the standard A* algorithm and the differences may reach over a dozen percent. It was also noticed that the gain is greater on the terrain where differences of properties are larger. Moreover, the length of path found is shorter on average, despite the used optimality criterion was disparate from the shortest path criterion. The changes proposed in the Theta* algorithm give opportunity of finding optimal path on heterogeneous terrain which can be realized by a robot with non-steered or caster wheels.

AUTHOR

Piotr Jaroszek – Industrial Research Institute for Automation and Measurements (PIAP), Warsaw, 02-486, Poland.

E-mail: pjaroszek@piap.pl

REFERENCES

[1] Hwang, Y. K., Ahuja, N., "A potential field approach to path planning", *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, 1992, 23–32. DOI:10.1109/70.127236.

[2] Chung, W. K., Xu, Y., "A generalized 3-D path planning method for robots using Genetic Algorithm with an adaptive evolution process". In: *2010 8th World Congress on Intelligent Control and Automation (WCICA)*, July 2010, 1354–1360. DOI:10.1109/WCICA.2010.5554851.

[3] Cormen, T. H., Stein, C., Rivest, R. L., Leiserson, C. E., "Introduction to Algorithms", 2nd ed. McGraw-Hill Higher Education, 2001.

[4] Dijkstra, E. W., "A note on two problems in connexion with graphs", *Numerische Mathematik*, vol. 1, no. 1, 1959, 269–271. DOI:10.1007/BF01386390.

[5] Hart, P. E., Nilsson, N. J., Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, 1968, 100–107. DOI:10.1109/TSSC.1968.300136.

[6] Nash, A., "Any-angle path planning". University of Southern California, 2012.

[7] Nash, A., Daniel, K., Koenig, S., Feiner, A., "Theta*: Any-angle Path Planning on Grids". In: *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, Vancouver, British Columbia, Canada, 2007, 1177–1183.

[8] Yap, P. K. Y., Burch, N., Holte, R. C., Schaeffer, J., "Abstract: Block A* and Any-Angle Path-Planning". In: *Fourth Annual Symposium on Combinatorial Search*, July 5, 2011.

[9] Ferguson, D., Stentz, A., "Field D*: An Interpolation-Based Path Planner and Replanner". In: *Robotics Research*, 2007, 239–253. DOI:10.1007/978-3-540-48113-3_22.

[10] Uras, T., Koenig, S., "An Empirical Comparison of Any-Angle Path-Planning Algorithms". In: *Eighth Annual Symposium on Combinatorial Search*, May 14, 2015.

[11] Choi, S., Yu, W., "Any-angle path planning on non-uniform costmaps". In: 2011 IEEE International Conference on Robotics and Automation (ICRA), May 2011, 5615–5621. DOI:10.1109/ICRA.2011.5979769.

[12] Bresenham, J. E., "Algorithm for computer control of a digital plotter", *IBM Systems Journal*, vol. 4, no. 1, 1965, 25–30. DOI:10.1147/sj.41.0025.