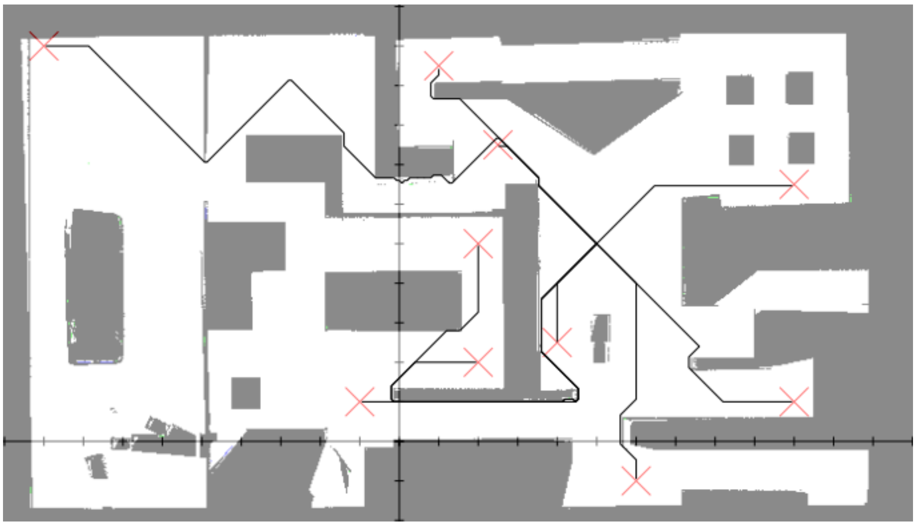
Universiteit van Amsterdam

Bachelor Thesis Artificial Intelligence

Using path planning to grade the quality of a mapper



Author:
Christiaan Walraven
cwalrave@science.uva.nl

Supervisor:
Arnoud Visser
a.visser@uva.nl

June 26, 2009

Abstract

In the Robocup Urban Search And Rescue League, one of the objectives is to make a map of the surroundings of a disaster struck environment. To grade the quality of such a map is not a trivial problem. This paper introduces a way to evaluate such a map by using computer vision techniques on the map and then using path planning to end up with an objective grade for the map.

Contents

1 Introduction

The RoboCup is an annual competition in the field of robotics [8]. It started out in 1993 as a robotic soccer competition with beating the world champions of soccer by the year 2050 as their final goal. Since then it has grown out to become much more than that by creating different competition leagues in robotics, aiming to promote and stimulate research and education in the field of artificial intelligence by competing against teams from all over the world.

One of the leagues in which can be participated is the Urban Search And Rescue (USAR) Virtual Robot Rescue League, in which a virtual team of robots searches for victims in a simulated situation where help is needed [1]. In this league a team of robots has to enter an area where a disaster has happened to explore the situation as thoroughly as possible and return with useful information about the surroundings of the disaster struck environment. The simulation is done in USARSim [7].

In this league, the teams have a limited amount of time to drive around in the virtual world and make a map (such as in figure 1) of the environment, after which they will be graded on three different criteria:

1. the size of the part of the map the robots cleared
2. the number of victims found
3. the quality of the map

The cleared space of the map is easily determined by calculating the size of the surface of the cleared space (the green part in figure 1). The number of victims found is also straightforward to determine, by counting the number of victims the team of robots have found and checking if the locations of the found victims are correct. Unfortunately, the quality of the map can not be measured as easily as the first two criteria. This resulted in appointing a human referee to determine the number of points given for this criterium, leaving the grading subjective. Where some people might grade a map that looks like the actual world as the best map, others might consider the usefulness of a map of far greater value than it looking like the actual world.

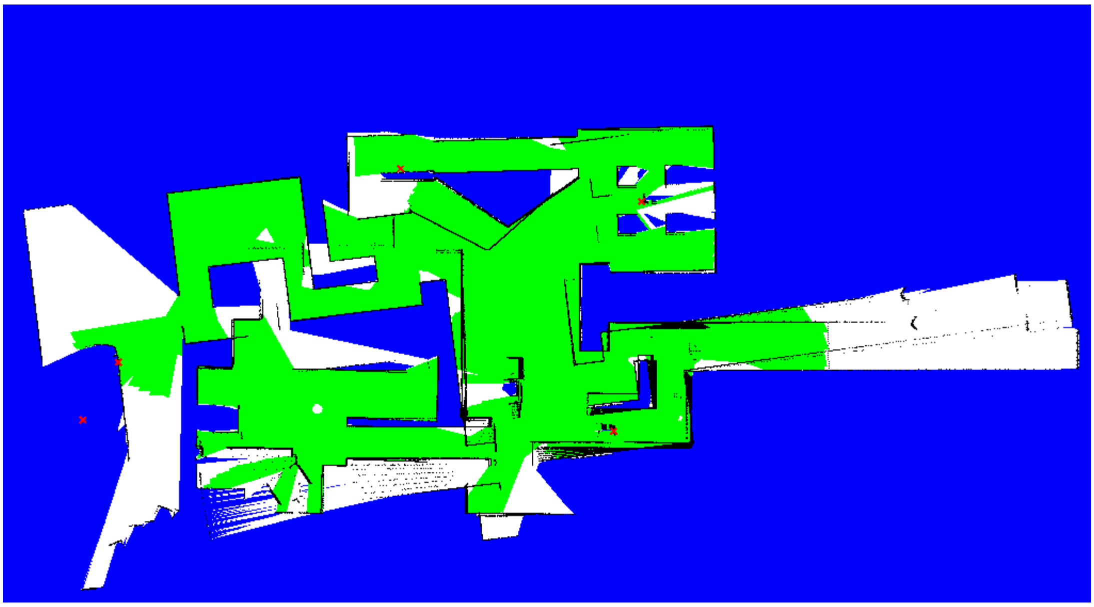


Figure 1: A typical map made by a robot

If there would be a objective way to determine the quality of the map, the help of a referee would be superfluous, making the grading process both more efficient and fair. In this paper one way to do this will be presented and discussed. The presentation of this method for giving a grade to a map is done in the next chapter, after which the results are discussed in chapter 3. A discussion about the pro's and con's of this method is given in chapter 4, concluding in chapter 5 with possible future work.

2 Method

In an upcoming paper by Balaguer, Balakirsky, Carpin and Visser [3], a way to evaluate the quality of a map by planning a number of paths to predefined reachable locations [4] is suggested. The quality of the map will then be calculated by appointing a fraction of the grade to each one of the reachable locations, resulting in the complete grade by summing these fractions. Off course it would be best if the path planning algorithm comes up with the optimal path to the goal, but there are a few ways in which this might go wrong [2]:

1. The goal state is outside of the cleared path
2. Every path to the goal state is blocked, due to the robot seeing obstacles that are not actually there
3. The path that is found is shorter than the optimal path due to obstacles that are actually there, but were not seen by the robot (a shortcut)
4. The path that is found is longer than the optimal path due to the robot seeing obstacles that are not actually there (a detour)

While the first three will end in total failure, the last one is only a detour and should be given a small penalty.

Using this idea, a step-by-step method can be made to evaluate the quality of a map:

1. Take a correct map of the environment which was mapped and choose a number of points that are reachable on this map.
2. Plan paths from the start position (which was used by the mapping robots as the position from where the mapping task took place) to each of the reachable points and check the length of each of these paths.
3. Check if the points are reachable on the map that has to be evaluated by using path planning to these locations.
4. Check if every step in the found paths is a free space on the actual map.
5. Check the lengths of the found paths and compare this to the lengths of the optimal paths that were found using the actual map to get the appropriate penalty.

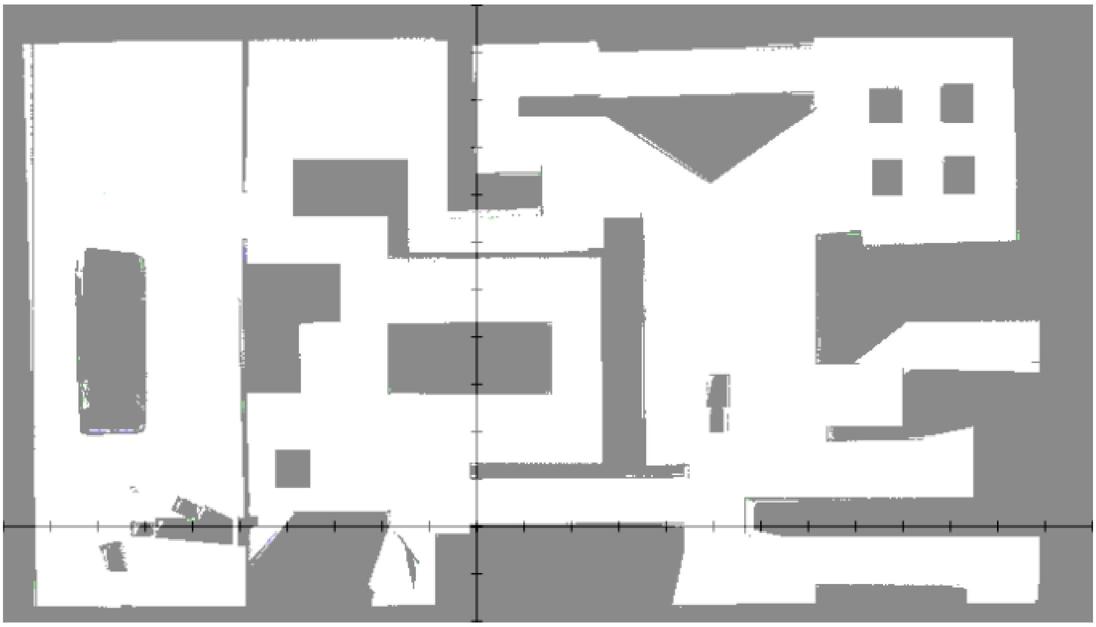


Figure 2: The actual map of the mapping world

The following formula is then used to calculate the grade G that will be given to the map:

$$G = \sum_{i=1}^{n} \frac{g}{n} p_i \tag{1}$$

with n the number of reachable points, g the highest reachable grade and $p_1 ... p_n$ the penalty for each found path.

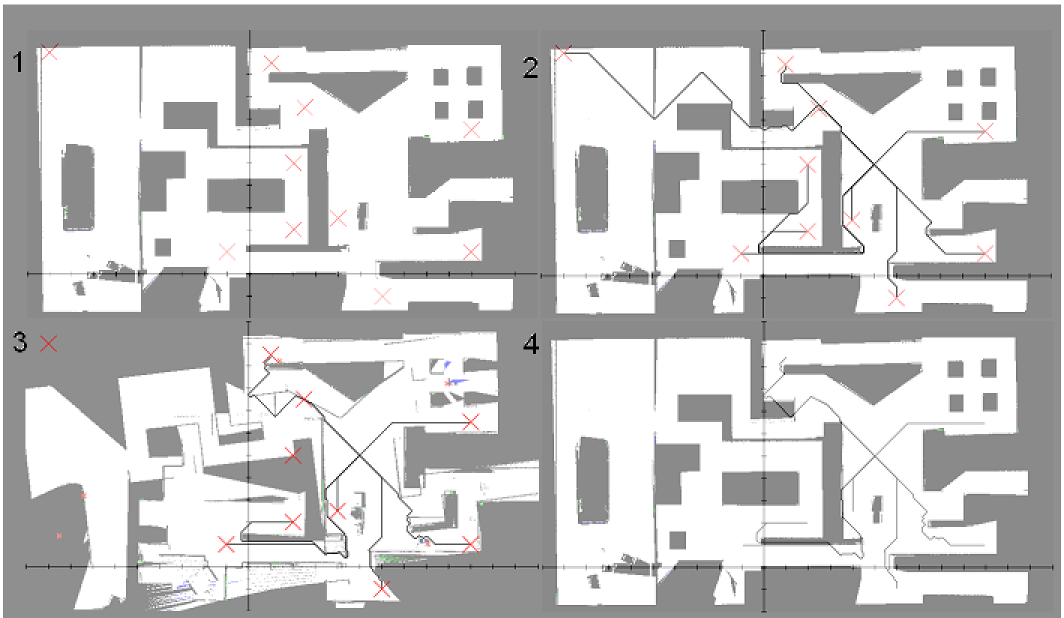
p_i will be 0 when the target point is not reached and 1 when the optimal path is found.

Figure 3: A screenshot of the mapping world

3 Results

In this paper, the Mapping World (Figure 3) was used in Unreal Tournament 2004 [9]. The A* algorithm [6] was used to do the path planning. There were ten reachable points chosen on the actual map. The perfect map would be awarded 50 points. Figure 4 shows the five steps that were described in the *Method* section of this paper using figure 2 as the actual map and figure 1 as the map that has to be graded.

1. The reachable points are chosen on the actual map
2. The optimal paths are calculated
3. Paths are planned on the map that has to be graded
4. These paths are checked to be consistent with the actual map
5. Check the length of the found paths and compare this to the length of the optimal paths that were found using the actual map to get the appropriate penalty.



5

n	1	2	3	4	5	6	7	8	9	10
a_i	218	77	75	41	38	87	90	142	194	168
m_i	--	--	75	41	38	94	90	150	--	--
p_i	0	0	1	1	1	0.86	1	0.9	0	0

Figure 4: The five steps as described in the Method section of this paper

The total grade G was calculated using formula (1) where

$$p_i = \left(\frac{a_i}{m_i}\right)^2 \qquad \text{if the target point is reached} \tag{2}$$

$$p_i = 0 \qquad \text{if the target point is not reached} \tag{3}$$

with $a_1...a_n$ the length of each optimal path and $m_1...m_n$ the length of each measured path.

The standard penalty is squared to give a bit more weight to detours, resulting in a bigger penalty for small detours, for these are the most common. This way a path that is twice as long as the optimal path will be awarded only $\frac{1}{4}$, which seems quite fair as well.
Because there are 50 points to be awarded, there are 5 points for every reached location. Together with the penalties, the formula gives the following grade to the map in figure 1:

$0 + 0 + 5 + 5 + 5 + 4.3 + 5 + 4.5 + 0 + 0 = 28.8$ out of 50.

4 Discussion

The method described in this paper seems to work quite well on the quality measurement problem. It gives a definite answer to the question of map quality and can be used with bigger maps as well. In the case reported in this thesis, all reachable points are quite close to the robots start position, but as we can all imagine, robots will quickly become better mappers over the course of the next few years and the points can be put farther away from the start position.

One could also argue that we should already include points that are farther away. In that way there will be lower scores in the next few years, but those scores will be comparable with scores that are given out in the future. To do this effectively, the most obvious choice would be to just add more points on the actual map, or even place a target at every reachable square meter.

Another fact that might be seen as an obstacle is the fact that the maps are treated as images and analyzed using computer vision. This results in a number of unwanted problems. For instance, the paths on the map that has to be graded are tested on the actual map in step 4 of the method. Because of this, the two maps have to be of the same size, with the same orientation and location of the starting point. This is off course a disadvantage, for maps that are rotated in their entirety, might be much better, but will be graded very low, since few of the target points can probably be reached. Off course it may be argued that a robot that does not know it's orientation in space will be utterly useless, but still this could be seen as a disadvantage in grading a map.

5 Future Work

There is a lot of work still to be done in the field of automatically grading maps. As a first, it would be very efficient to have a way to automatically change the scale, location and rotation of the map to fit on the actual map. So far, this has to be done by hand. Once that is done, tests could be done on any given map in no time, resulting in more data to check how good a map really is.

Also, a lot of has to be done to completely automatize the whole system. These are mostly small tasks that are done by hand at the moment, but should not take too much time to implement. These include:

- create target points in a target point file by simply clicking on the target locations on the map.
- automatically loading both the actual map and the map that has to be graded into the system, making it possible to directly check for step 4.
- automatically do step 5 by loading the optimal path lengths and immediately calculate the penalties.

Furthermore, the path planning algorithm can be made a lot faster, by using something else than A*. Since a lot of paths start in the same way, this first section is calculated numerous times, it would be better to have some way points in small corridors and on corners from where new paths can be planned. A breadth first solution that keeps on going until the whole map has been covered would also be an option, but the space complexity will most likely get in the way here.

6 References

[1] Stephen Balakirsky, Stefano Carpin, Alexander Kleiner, Michael Lewis, Arnoud Visser, Jijun Wang and Vittorio Amos Ziparo, "Towards heterogeneous robot teams for disaster mitigation: Results and Performance Metrics from RoboCup Rescue", Journal of Field Robotics, volume 24(11-12):pp. 943-967, November 2007.

[2] Thomas Collins and J.J. Collins and Conor Ryan, "Occupancy Grid Mapping: An Empirical Evaluation", Mediterranean Conference on Control & Automation (MED '07)", pages 1-6, June 2007, 10.1109/MED.2007.4433772.

[3] Benjamin Balaguer, Stephen Balakirsky, Stefano Carpin and Arnoud Visser, "Evaluating Maps Produced by Urban Search and Rescue Robots: Lessons Learned from RoboCup"

[4] O.Hachour, "Path planning of Autonomous Mobile robot", International Journal of Systems Applications, Engineering & Development, Issue 4, Volume 2, 2008

[5] Shane O'Sullivan, "An Empirical Evaluation Of Map Building Methodologies in Mobile Robotics Using The Feature Prediction Sonar Noise Filter And Metric Grid Map Benchmarking Suite", Master Thesis, University of Limerick, 2003

[6] P. E. Hart, N. J. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", 1968 IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2): 100107. doi:10.1109/TSSC.1968.300136

[7] S. Carpin, M. Lewis, J. Wang, S. Balarkirsky and C. Scrapper, "USARSim: a robot simulator for research and education", In Proceedings of the IEEE International Conference on Robotics and Automation, pages 14001405, 2007.

[8] The Official RoboCup website, "http://www.robocup.org/"

[9] The Official Unreal Tournament 2003 And 2004 website
"http://www.unrealtournament2003.com/"