

Comparison of Simulation Models for a Cleaner Robot

Rahul Ravindran
Computer Engineering
VESIT, University of Mumbai
Mumbai, India

ABSTRACT

Simulation algorithms like random walk and standard walk can be used to simulate the working of a cleaner robot assigned a task of cleaning the room. This paper represents a way to simulate and conclude which model would be best suited for a robot to clean a room efficiently. The algorithms implemented represent a stochastic simulations which results in random behavior, causing the results to fluctuate over many iterations, however after an analysis of large number of such test cases and statistical data, a conclusion can be made as to which model would best serve the purpose. The end result is a model that when applied to the robot with the given task would perform it effectively and quickly in the least possible time and any given condition.

Keywords:

Simulation, Random Walk, Standard Walk, Cleaner Robot, Stochastic.

1. INTRODUCTION

Simulation attempts to build an experimental device called a model, the intent of a model is to provide possible behaviors of the actual system being modeled. Al-most all simulation models are descriptive and not prescriptive as they provide information on how a system behaves under different conditions but not how to arrange the conditions to make the system work best. Most simulation models can be classified along three dimensions. One of the dimensions selected for simulating the working of the cleaner robot is a stochastic model, stochastic simulations include randomness which leads to different results each time the model is employed. The second dimension which fits our model is the discrete model where each robot is accounted for and not grouped together as a single entity. Simulation modeling allows designers and engineers to avoid repeated building of multiple physical prototypes to analyze designs for new or existing parts. Before creating the physical prototype, users can virtually investigate many digital prototypes. Using the technique, they can:

- Simulate part failure and identify the loading conditions that cause them
- Verify handy calculation
- Provide a prototype as to how the actual system will work in a given set of conditions.

One of the two algorithms applied in order to simulate the working of the cleaner robot is the random walk simulation model which has

the name suggests represents a truly stochastic model that does not provide any kind of concrete decision making to the robot, while on the contrary relies on the random nature of the algorithm. The second algorithm that is used to model the working of the cleaner robot is the standard walk model which restricts the movement of the robot until it reaches a certain guard condition which in this case would be the detection of the boundary of the room upon which the robot will change direction and proceed with its working. The use of these two simulation models can be effectively used to model the working of the cleaner robot in different conditions and provide a robust and efficient model to solve the problem of cleaning the room effectively and quickly.

2. RANDOM WALK

A random walk is a mathematical formulization of a path that consists of a succession of random steps. For example, the path traced by a molecule as it travels in a liquid or a gas, the search path of a foraging animal, the price of a fluctuating stock and the financial status of a gambler can all be modeled as random walks, although they may not be truly random in reality.

Random walks have been used in many fields: ecology, economics, psychology, computer science, physics, chemistry, and biology. Random walks explain the observed behaviors of processes in these fields, and thus serve as a fundamental model for the recorded stochastic activity.

One of the most fundamental example of random walk is the example of Brownian motion, the random motion of atoms within an element, until 1900 no concrete theory was posited to model such a stochastic phenomenon in the filed of physics. Random walks is widely used in many fields to provide working of many concepts under different conditions and to understand processes and systems more appropriately.

Implementing the random walk simulation model for the problem of a cleaner robot involved the idea of considering the random walk as a walk itself, Imagine a drunken farmer standing in the middle of the field and every second he takes a random direction what would be his expected position in say a time span of 1000 seconds, more likely if the farmer takes ever so many steps is he more likely to be at distance away from origin or moving around it.

Fig. 2 shows the rectangular grid depicting the filed on which the farmer is at a initial position O, at the very instant the farmer has an option to move to either positions A,B,C or D assuming he takes a

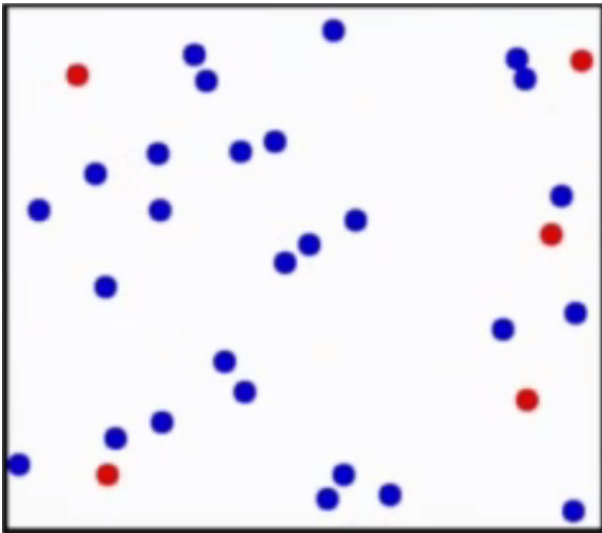


Fig. 1. Random motion of atoms(Brownian motion)

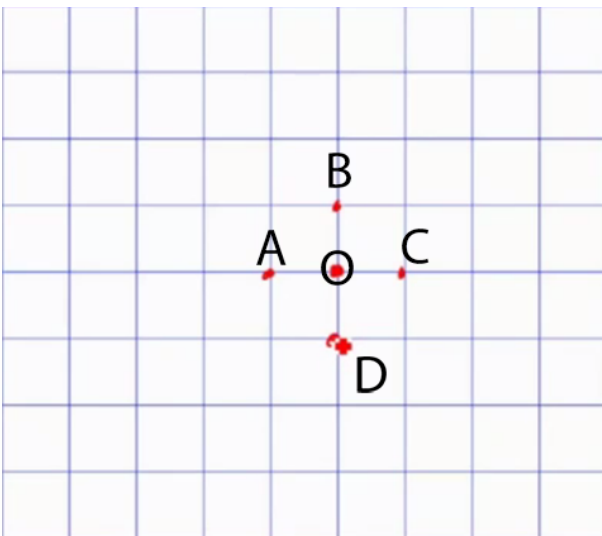


Fig. 2. Initial intuition for random walk simulation model

step of 1 at each time instant. When the farmer has taken his first move say C, at the next instant of time he has four more possibilities and he may choose any depending on that situation and therefore as we see the problem tends to become more difficult as it's complexity increases with each step and it becomes abstruse to find out the exact nature of the farmer after a certain interval of time. The first simulation attempts to achieve an operation similar to the above working but this time on a more practical example a cleaner robot. The rectangular field in Fig. 2 will now be used to depict a room and the cleaner robot will be simulated based on random walk to clean the room, the assumption will be that when the robot steps on a tile it has cleaned the tile. Thus using the random walk we can simulate the working of a cleaner robot and make some kind of judgment as to whether or not the room will be cleaned by the robot after a certain amount of time.

3. SETTING UP THE SIMULATION MODEL

The implementation will focus on rooms that are rectangular in shape and the room will be depicted using a rectangular 2D array of tiles.

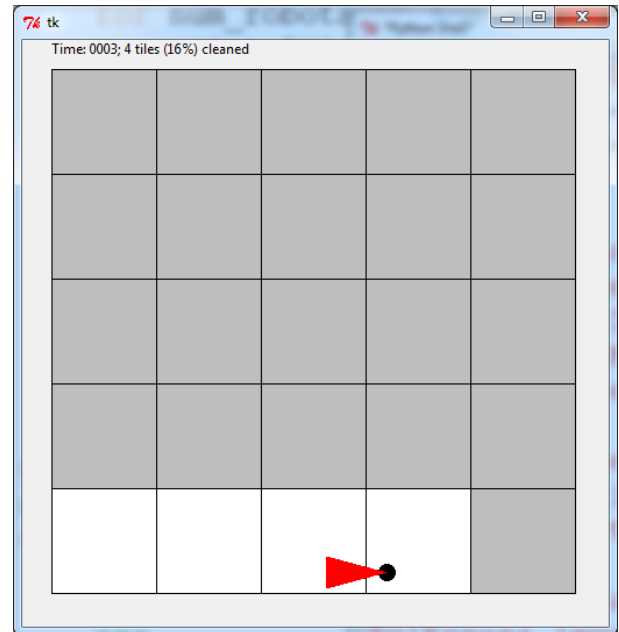


Fig. 3. Implementation of the rectangular room and the robot cleaning the tiles using random walk in python.

Fig. 3 shows the actual simulation model implemented in python, the rectangular array of squares represents the tiles of the rectangular room and the red arrow with a black dot at the front depicts the robot and it's movement around the room, the room initially has all it's tiles dirty and depicted in grey color, as the robot moves around the room it begins to clean the room, once a tile is cleaned the robot moves ahead as it's job is to clean the room entirely. The cleaned tiles are marked with white color and the robot is considered to have cleaned the tile once it has landed on it for the first time. The top left corner of Fig. 3 depicts the overall tiles cleaned at any particular instant of time and vice versa the time taken for cleaning a particular amount of the room. The simulation model will be used to test out the working of the robot using standard walk and the random walk simulation model and perform rigorous testing in order select the apt model for cleaning the room practically and also efficiently. Statistical information generated after the simulation ends will provide a further intuition as to which simulation model will be best useful for the cleaner robot.

4. RANDOM WALK SIMULATION

The first simulation involves the robot cleaning the room with the random walk model. The robot begins at any arbitrary position in the room. After each move the robot re-calibrates it's direction and begins to clean the room. The intuition is that more the steps the robot takes the higher is the chance that the robot would have traveled a larger distance from it's initial start position.

Fig. 4 shows the initial position of the robot in the room that is

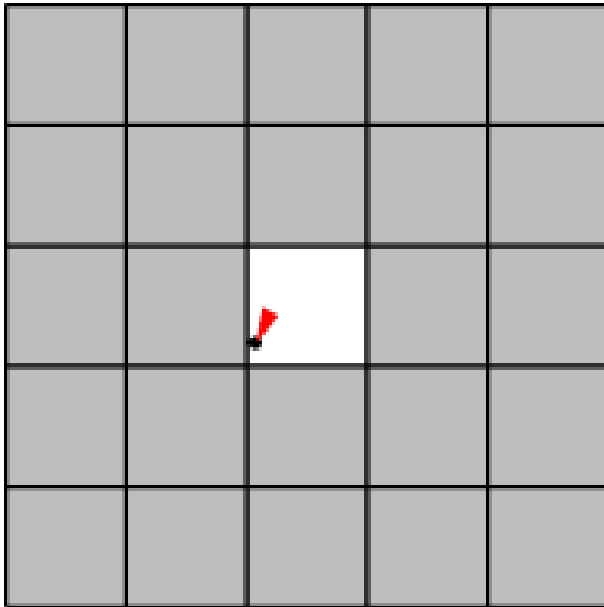


Fig. 4. Initial position of the robot at time $T=0$

required to be cleaned, each run of the simulation results in a different initial position of the robot in the room.

```

Require: Randomly initialize the position of the robots
repeat

    if position in room and tile is not cleaned then
        clean tile
        set new direction
    end if
    if position not in room then
        set new direction
    end if
until all tiles haven't been cleaned
    
```

The above algorithm successfully simulates the movement of the robot across the room using the random walk simulation model. After each move the robot recalibrates its position and continues the task of cleaning the room.

Although the random walk simulation can provide as a kind of simulation to help make the robot clean the room entirely it does not guarantee that the entire room will be cleaned due to its random nature, hence stochastic simulations usually have a tiny bit of controversy surrounding the working due to the fact that it cannot be affirmed that the simulation is true in all possible conditions.

Fig. 5 and fig. 6 represents the working of the model during time instance 2 and 3. Each Robot follows a certain number of rules as given below:

- Each robot has a position inside the room. We'll represent the position using coordinates (x, y) which are floats satisfying $0 \leq x < w$ and $0 \leq y < h$. In our program we'll use instances of the Position class to store these coordinates.
- A robot has a direction of motion. We'll represent the direction using an integer d satisfying $0 \leq d < 360$, which gives an angle in degrees.

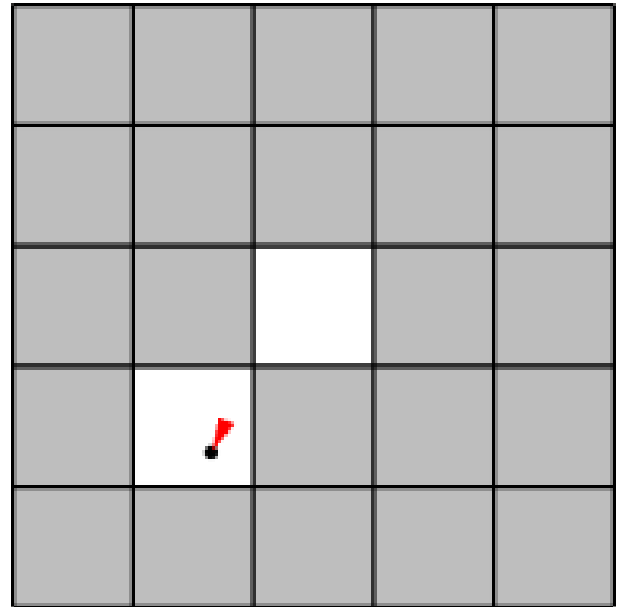


Fig. 5. position of the robot at time $T=1$

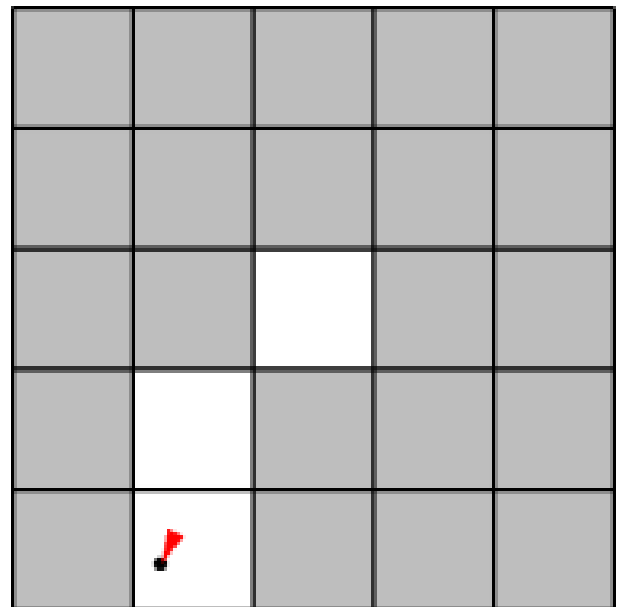


Fig. 6. position of the robot at time $T=2$

—All robots move at the same speed s , a float, which is given and is constant throughout the simulation. Every time-step, a robot moves in its direction of motion by s units.

—If a robot detects that it will hit the wall within the time-step, that time step is instead spent picking a new direction at random. The robot will attempt to move in that direction on the next time step, until it reaches another wall.

The simulation ends when a specified fraction of tiles have been cleaned or the entire room is cleaned.

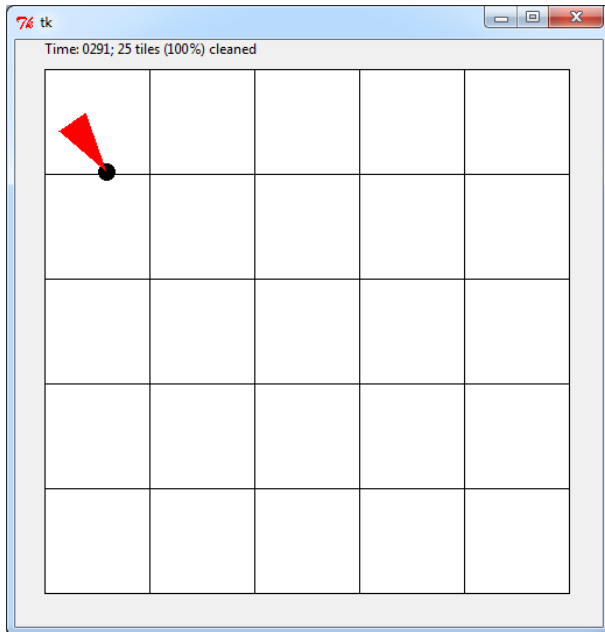


Fig. 7. Completion of random walk simulation on a 5*5 rectangular room.

5. STANDARD WALK SIMULATION

The robot starts out at some random position in the room, and with a random direction of motion. The illustrations below show the robot's position (indicated by a black dot) as well as its direction (indicated by the direction of the red arrowhead). The standard walk simulation begins with the robot being initialized with a random position, thereafter the robot's position and direction unlike random walk is not calibrated at each instance but is calculated only under a certain guard condition which in this case would be the robot not crossing the room i.e encountering the boundary of the room. The simulation consists of following details:

- Multiple Robots: In general, there are $N \geq 0$ robots in the room, where N is given. For simplicity, assume that robots are points and can pass through each other or occupy the same point without interfering.
- The room: The room is rectangular with some integer width w and height h , which are given. Initially the entire floor is dirty. A robot cannot pass through the walls of the room. A robot may not move to a point outside the room.
- Tiles: The area of the room is divided into 1×1 tiles (there will be $w * h$ such tiles). When a robot's location is anywhere in a tile, the entire tile is considered to be cleaned (as in the pictures above). By convention, we will refer to the tiles using ordered pairs of integers: $(0, 0)$, $(0, 1)$, ..., $(0, h-1)$, $(1, 0)$, $(1, 1)$, ..., $(w-1, h-1)$.
- The simulation ends with the robot cleaning the room entirely.

Fig. 8 shows the sequence of iterations followed in cleaning the room, the direction of the robot is calibrated at each instance it encounters a wall.

The algorithm for the standard walk simulation differs from the random walk simulation only in the case that direction is not calculated at each instance the robot is still in the room. Standard walk simulation model provides a better convergence rate as the time taken by the robot to clean the room is well within the time required to

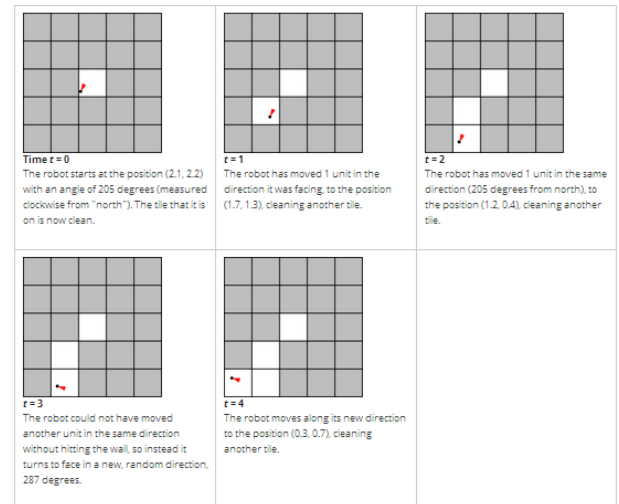


Fig. 8. Sequence of steps for the simulation of the robot

Require: Randomly initialize the position of the robots
repeat

if position in room and tile is not cleaned **then**
 clean tile

end if

if position not in room **then**
 set new direction

end if

until all tiles haven't been cleaned

clean the same room by a robot executing the random walk simulation model. However the direction in which the robot moves once it encounters the wall is still random thereby still not assuring that the simulation will end in termination however it is easy to construe that the distance traveled by the robot relative to its initial position will be more than that of the robot working under the random walk simulation model. The robot being simulated efficiently cleans the room entirely at a relatively faster pace than the robot which was operating in random walk simulation model.

Fig. 9 shows the termination of simulation of the robot, as expected at a rate faster than the random walk robot, statistical information further substantiates the fact that the robot using standard walk will perform faster in all given cases and take the least amount of steps.

6. COMPARISON OF RANDOM AND STANDARD WALK

A detailed comparison of both the models will help better understand why the standard walk model leads to a better result as compared to the random walk model. Random walk model relies totally on the random nature of the simulation to complete the action of cleaning the room, however the same random nature of the problem causes the robots to stray in path that is it may so happen that the robots linger in a certain space long enough to not complete the action of cleaning the room. Generation of graphs with the help of simulation of upto 10 robots cleaning the same rectangular room at the same time gives detailed analysis of how much speed up is brought about by the robots implementing the standard walk simulation model. Thus two graphs containing the simulations of not

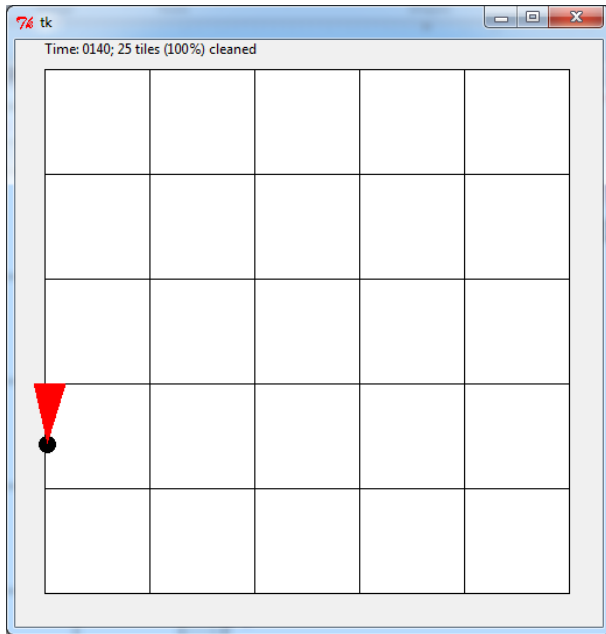


Fig. 9. Termination of simulation of a robot operating in standard walk simulation model

only different number of robots but also different size of rooms enable us to clearly pick out the most apt and suitable simulation model that can practically help the cleaner robot to efficiently clean the room and also do it in the least possible time.

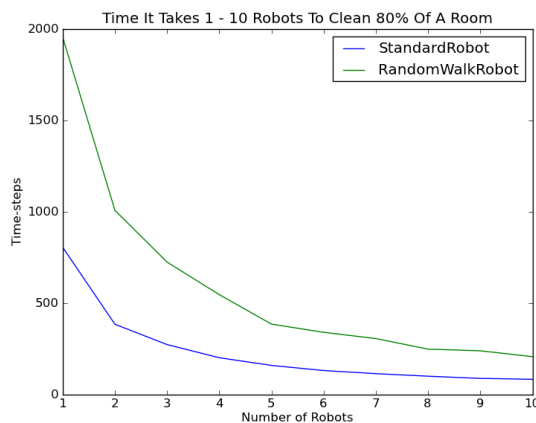


Fig. 10. Time taken by 1-10 robots to clean 80% of the room

Fig.10 shows the time-steps required by various number of robots to clean the room, standard walk robot takes the least amount of time and also if more number of robots were to be employed to clean the room the time required decreases significantly.

Fig. 11 shows the time taken by a robot to clean 80% of various shaped rooms like 20*40 and so on, the standard walk robot maintains a constant aspect ratio as compared to the random walk

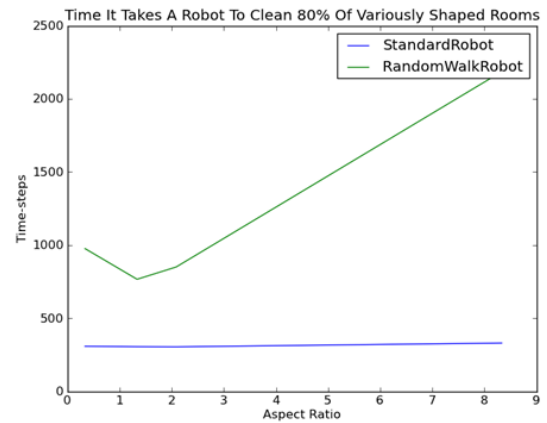


Fig. 11. Time taken by a robot to clean 80% of variously shaped rooms

robot., the time steps required by the robot increases significantly for the random walk robot as the random nature of the movement causes the robot to move around the room in a more frenzied manner as compared to that of the robot using standard walk simulation model.

7. CONCLUSION

The results of the graphs generated from the simulation models it is clear that the robot operating under the standard walk model performs much better than the robot operating under the random walk simulation model. The standard walk robot can not only clean the room efficiently but also perform the task with much greater speed as compared to the random walk robot[?].

8. REFERENCES

- [1] H. Choset. Coverage for robotics a survey of recent results." *Annals of Mathematics and Artificial Intelligence*", 31(1):113126, 2001.
- [2] L.H. Erickson, J. Knuth, J.M. OKane, and S.M. LaValle. "Probabilistic localization with a blind robot". In Proc. of the IEEE Intl.Conf On Robotics and Automation, 2008
- [3] K. Helsgaun. "An effective implementation of the lin-kernighan traveling salesman heuristic". *European Journal of Operational Research*, 126:106130, 2000
- [4] W.Y. Jeong and K.M. Lee. "Cv-slam: A new ceiling vision-based slam technique". *Proc. of Robotics: Science and Systems (RSS)*, 2005.
- [5] E. Kruse and F.M. Wahl. "Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning".
- [6] M. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain. "Auction-based multi-robot routing". In Proc. of Robotics: Science and Systems (RSS), 2005.
- [7] J.C. Latombe. "Robot Motion Planning". Springer Verlag, 1990.
- [8] G. Lawitzky. "A navigation system for cleaning robots". *Autonomous Robots*, 9(3):255260, 2000.

- [9] M. Luber, G.D. Tipaldi, and K. Arras. "*Place-dependent people tracking*". In Proc. of the Intl. Symposium of Robotics Research (ISRR),2009.
- [10] R. Mannadiar and I. Rekleitis. "*Optimal coverage of a known arbitrary environment*".In Proc. of the IEEE Intl. Conf.on Robotics and Automation(ICRA), 2010
- [11] H. Moravec and A. Elfes. "*High resolution maps from wide angle sonar*". In Proc. of the IEEE Intl. Conf.on Robotics and Automation(ICRA), 1985
- [12] B. Siciliano and O. Khatib, editors. Handbook of Robotics. Springer,2008.
- [13] S. Thrun, W. Burgard, and D. Fox.Probabilistic Robotics. MIT Press,2006.
- [14] Y. Zhang, J. Liu, G. Hoffmann, M. Quilling, K. Payne, P. Bose, and A. Zimdars. "*Real-time indoor mapping for mobile robots with limited sensing*". In Proc. of the 3rd Intl. Workshop on Mobile Entity Localization and Tracking , 2010.
- [15] R. Zlot, A. Stentz, M.B. Dias, and S. Thayer. "*Multi-robot exploration controlled by a market economy*".In Proc. of the IEEE Intl. Conf.on Robotics and Automation(ICRA), 2002