Human-Aided Cleaning Algorithm for Low-Cost Robot Architecture*

Seungyong Kim, Kiduck Kim, and Tae-Hyung Kim

Department of Computer Science and Engineering, Hanyang University
1271 Sa 1-Dong, Ansan, Kyunggi-Do, 426-791, South Korea
{kimsy,kdkim,tkim}@cse.hanyang.ac.kr

Abstract. This paper presents a human-aided cleaning algorithm that can be implemented on low-cost robot architecture while the cleaning performance far exceeds the conventional random style cleaning. We clarify the advantages and disadvantages of the two notable cleaning robot styles: the random and the mapping styles, and show the possibility how we can achieve the performance of the complicated mapping style under the random style-like robot architecture using the idea of human-aided cleaning algorithm. Experimental results are presented to show the cleaning performance.

Keywords: Cleaning robots, Random style cleaning, Mapping style cleaning, Human-robot interaction.

1 Introduction

Since the application of household service robots has been restricted to nonessential tasks, the proliferation of the robot markets has been limited so far contrast to that of the industrial robot applications. In order to be of practical value, we may have to answer a basic question, “what does the robot do?” Recently, an autonomous robotic vacuum cleaner has a clear answer to the question and becomes one of the most successful killer applications that widen the horizon of household service robots. It simply cleans floors. Albeit simple, the cleaning robot delivers a practical importance to consumer electronics industry, but it challenges technical and economic problems [1]. Thus, the robot cleaner would not be able to be successful in the market, without having a proper compromise between the two problems. If we stress on a technical problem, we may end up with a high-cost intelligent robot, but of little economic value. If we pursue an economic value, the final product may not clean well enough.

Traditionally and academically, many researches on the cleaning algorithm are based on the existence of navigation maps for a cleaning floor [2, 3]. Creating and managing such a map would play a pivotal role for the robot to find its cleaning paths and particular locations to move. Robots should recognize proper landmarks using an expensive vision sensor, stores those image points to map database with coordinates.

* This research has been supported by the Ministry of Education and Human Resources Development, S. Korea, under the grant of the second stage of BK21 project.

J. Jacko (Ed.): Human-Computer Interaction, Part III, HCII 2007, LNCS 4552, pp. 366–375, 2007.

The map building process is complicated enough. Moreover, although the mapping and navigation process is completed, the method raises a localization problem [4, 5], which is more difficult one because it is expensive to compensate the moving aberration due to friction, slipping, and unorthodox floor conditions. Dead-reckoning [4], Markov localization [6], and Monte Carlo localization [7] are some of approaches to deal with the problem. The mapping and navigation based approach would be technically possible to construct a truly intelligent cleaning robot in unknown and unstructured environments, albeit expensive.

On the other hand, if we decide to sacrifice the utmost cleaning efficiency, which is not that critical to housekeeping chores like cleaning, we may build a robot at a modest price, where the architecture is equipped with a modest microprocessor, minimum number of motors, and with a limited set of sensors like bump sensor, cliff and wall sensors. What about maps? No need. Its mobility is based on a random walk. Simply to be random, we may not need an expensive vision sensor or complex actuators for navigating maps. Roomba [8, 9] by iRobot, Inc., is a pioneering product in this random style robot, and Roboking by LG Electronics and Trilobite by Electrolux are one of those as well. The complexity is intentionally minimized yet they do clean well up to the users' expectations. A home service robot in this style was recorded as the first significant commercial success due to its simplicity in terms of cleaning operations at modest price.

Nonetheless, they used to clean the same area repeatedly while other part of the area remained unclean for a quite long time. In this paper, we present a human-aided cleaning algorithm for the better cleaning efficiency like map-based approach yet with low-cost robot architecture like Roomba. Navigation map building, autonomous path finding, and automatic localization problems are all important computation problems for machine intelligence, but the higher cost and complexity of such system is too much for floor cleaning robots. Note that human beings can do better for the area partitioning than machine. Moreover, the users are familiar with the interior shape of their home. In this work, we can parameterize the operational behavior of cleaning robot. At one extreme of human-robot interactions, if we do not teach anything about the cleaning space, the robot is supposed to move around the space randomly as Roomba does. At another extreme, if we teach every details about the moving information, the robot repeat what people do to clean their floor; the robot mimics housemaids. There is a wide spectrum of robot operations between the two extremes. For example, people may point out every one and each of turning points in a polygonal cleaning space, instead of dragging the robot to the entire cleaning space, and then the robot would be able to do the rest by automatically filling the space defined by the given set of edges. Designing the cleaning algorithm for the division of works between human beings and robots, we can achieve the two conflicting goals of low-cost robot architecture and cleaning efficiency as much as we can. The heart of our approach is to make a division of roles between human being and machine.

The remainder of the paper is organized as follows. Section 2 briefly describes the background of robot driving systems and their required set of sensors. In Section 3, we identify the set of robot instructions in our *human-aided cleaning algorithm* and

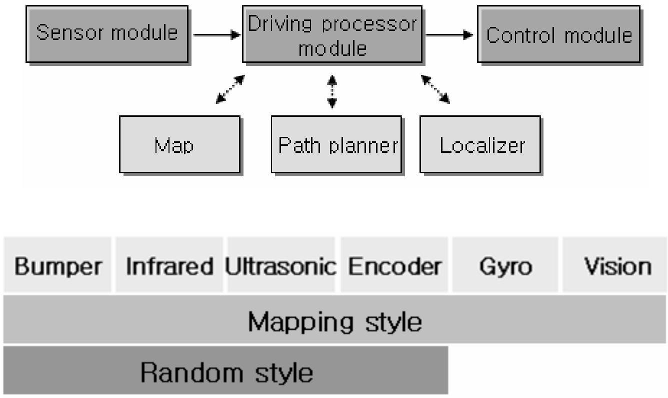


Fig. 1. Robot driving system and various sensors

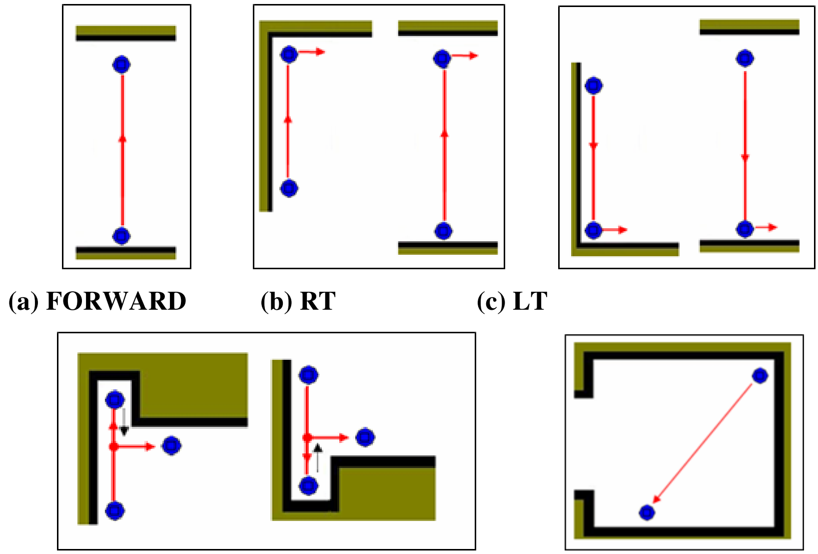
present our approach in general. To show the behavioral performance of our approach, we conducted experiments for diverse shapes of cleaning space with proper information from human beings, and compared them with that of Roomba. We used the Player/Stage simulator [10, 11] developed at the University of Southern California on Linux Fedora 3.0 with kernel version 2.6.9-1.667. The experimental results are discussed in Section 4. We conclude the paper in Section 5.

2 Background

There are two contrasting robot driving systems as shown in Fig. 1: the random style and the mapping style. The cleaning robot comprise a *sensor module* that manages a plurality of sensors, a *driving processor module* that processes cleaning algorithms based on the inputs from various sensors, and a *control module* that executes control instructions from the processor module via an actuator. The random style driving system consists of these three components. The mapping style robot additionally comprises a map that is built and used for navigation, a path planner that routes a whole path in a cleaning area, and a localizer that periodically estimates and corrects the current position when friction and slipping happen. The number and the type of the sensors are quite different among various cleaning robots, but gyro and vision sensors are normally employed only for the mapping style. Our approach essentially requires an ultrasonic[1] sensor for distance measurement, a gyro sensor for angle measurement, and an encoder for moving distance measurement. Possibly an infrared sensor can be included for cliff and staircase detection. We do not need an expensive vision sensor that is only used for map building and localization. Normal sensor configuration is summarized in Fig. 1, and our approach needs a gyro sensor additionally, comparing to the random style.

---

[1] Normally, a laser sensor is more accurate for the measurement, but more expensive.



(d) BLT & BRT **(e) GOTO**

Instruction	Description	Instruction	Description
FORWARD	Go straight	**BLT**	Make backwardly left turn
RT	Make right turn	**BRT**	Make backwardly right turn
LT	Make left turn	**GOTO**	Go to the next work point

Fig. 2. Basic robot control instructions

3 Human-Aided Cleaning Algorithm

As presented in the previous section, there are two robot driving styles: random and mapping. The random style robot can be built on inexpensive components and operate in simple but creative ways [8]. The main thrust of the human-aided cleaning algorithm is to perform like the mapping style robot, but with almost same robot architecture as the random style; our robot algorithm does not provide a mechanism for path planning and localization facilities but get provided the essential intelligence with a proper interaction from the human users who are familiar with the cleaning floor situations. The seemingly magical high performance comes right from the human-robot interactions. In this section, we present the overall architecture of our approach and clarify what human beings should provide to the robot.

3.1 Overall Architecture

The human-aided cleaning algorithm consists of three components: basic control instruction sets, region filling algorithm from the mapping style [3], and a list of necessary hardware components.

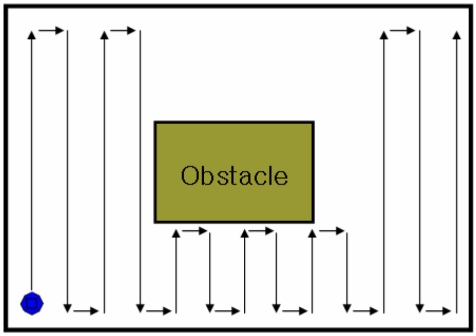


Fig. 3. Basic robot movement based on a region filling algorithm

Fig. 2 shows our six basic robot control instructions and portrays one and each of their operational semantics. The FORWARD instruction is executed until blocked by an obstacle. RT instruction is for making a right turn when the robot is blocked by a left and front side obstacle. Similarly, LT instruction is to make a left turn. If there is only a front side obstacle, the robot is able to execute RT and LT according to a given control program implementation. BLT is to make a left turn backwardly, that is effectively to make a right turn from the original moving direction after having run into a narrow alley. BRT is to make a right turn backwardly in order to escape from an alley. Finally, the GOTO instruction is to move directly from an endpoint of a cleaning section to a starting point of another one without performing a cleaning job. Cleaning sections are partitioned for the purpose of cleaning efficiency based on information from human users.

These six instructions can suffice to implement the cleaning robot movement that is similar to a region filling algorithm as shown in Fig. 3. From a specific starting point, the robot keeps moving up and down until it reaches to an obstacle. After it gets into an area that is limited by an obstacle, the movement is confined to the below of the obstacle. The upper part is remained not traveled. The pseudo-code representation for the region filling algorithm is shown in Fig. 4. The default instruction is FORWARD so the program does not have to specify it to the robot explicitly. Region filling algorithm is only used within a particular cleaning section. Human users provide the robot with the most reasonable section partitioning information so we can safely assume that it knows how the entire cleaning area is comprised of smaller cleaning sections that is to be cleaned based on the region filling algorithm.

3.2 Human-Robot Interaction

Region filling algorithm based robot movement is performed based on the space information that is provided by a human user: the starting point of the current cleaning section, the relative coordinate information for the cleaning space, and the moving distance and angle for the next cleaning section. A cleaning space can be partitioned to multiple cleaning sections for higher cleaning performance. A cleaning section is always rectangular. Thus, if there is a cleaning section partitioned with an obstacle like in Fig. 3, there can be a region that is not covered by our cleaning robot

```
// F:front, L:left, R:right
enum collision_type{
  Nothing =0,
  F,
  FL,
  FR,
  FLR
};

enum movement_template{
  Forward =0,
  RT,
  LT,
  BRT,
  BLT,
  GoTo,
};

for( ; ;)
{
  if( is_this_region_complete() )
  {
    if( next_region_check() )
    {
      movement = Goto;
    }
    else{
      break;
    }
  }
switch(movement){
  case Forward:
    switch(collision){
      case Nothing:
        /* straight forward */
        break;
      case F:
      case FR:
      case FL:
      case FLR:
        if ( /* upside? */){
          if ( /* FR || FLR */ ){
            movement = BLT;
          }else
            movement = RT;
        }else { /* down side */
          if ( /* FL || FLR */ ){
            movement = BRT;
          }
          else
            movement = LT;
        }
        break;
    }
    break;
      .
      .
      .
} // end switch
// end for
```

Fig. 4. Pseudo-code representation for a region filling algorithm

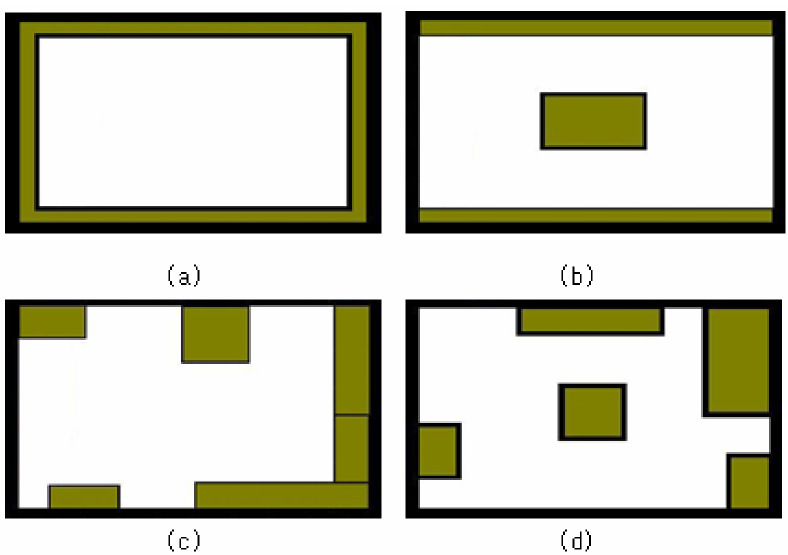


Fig. 5. Four different maps used in the simulation experiment

since the region filling algorithm is applied to one and each of those sections. Human users are responsible for the most reasonable partitioning for cleaning efficiency.

Completed at a section, the robot is directed to move directly to the starting point of the next cleaning section using the GOTO instruction with distance and angle parameters. The number of repetitions in a cleaning section can be decided arbitrarily depending on dirtiness.

Unlike the mapping method, our cleaning robot does not have to make computational efforts to come up with the cleaning space information, and the information is provided by a human user. A creative way of human-robot interaction is required to conveniently let the robot know about the cleaning space information consisting of the rectangular shape of sections and the cleaning sequence of those sections. The design of such interfacing method is beyond the scope of this paper, since the primary purpose of our research in this paper is to corroborate the usefulness of the human-aided cleaning algorithm, as shown in the next section.

4 Experimental Results

In our simulation experiment, we use four different maps from the simplest form without any obstacles to a complicated one with randomly placed obstacles as shown in Fig. 5. The cleaning area is represented by white areas in the figure. The room size excluding the area by obstacles is 8.2m X 4.5m. Cleaning performance will be represented by the progress percentage with regard to the cleaning area in a given time span and by the elapsed time to finish a given cleaning task.

4.1 Experiment Environment

We have implemented our simulation program on the Player/Stage simulator [10] developed at the University of Southern California, in order to confirm the efficacy of the human-aided cleaning algorithm comparing to the commercially successful Roomba cleaning algorithm. The working platform is Linux Fedora 3.0 with kernel version 2.6.9-1.667 on Pentium 4 CPU with 1GB memory. Our simulation relies on the functionalities of many virtual devices like laser, infrared, bumper sensors and so forth, that is supported and confirmed by Player 2.0 simulator [11]. The moving velocity is set to 0.3 m/sec.

We conducted four different simulation types in terms of the degree of human-robot interaction to the four maps in Fig. 5. Type 0 is without having any extra information from human users, that is alike pure Roomba algorithm. Type 1 is still without enough helps from human users but the cleaning space is bipartite using only two sets of virtual walls like in conventional Roomba-based cleaning. In type 2 simulation, the cleaning space is partitioned by human users but only with the very first starting point. In the last type, the robot is provided with the starting point of the starting section and a series of the next starting points of the next cleaning sections.

4.2 Results

Fig. 6. shows the progress percentage of cleaned area for four different types of maps that are applied to by four different human-robot interaction modes. We just limit the time interval up to 10 minutes because there is not much progress after then. For

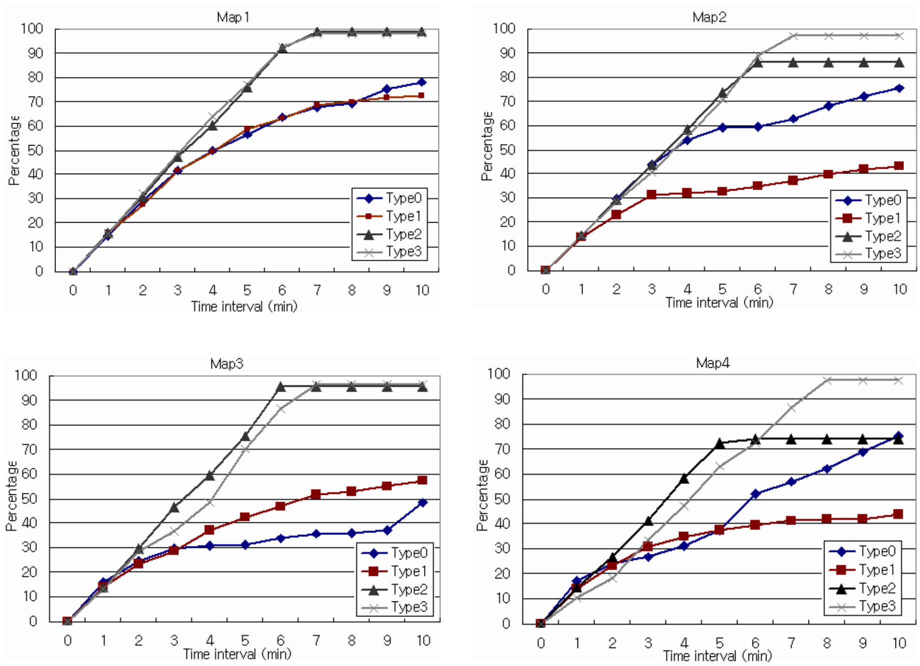


Fig. 6. Progress percentage of cleaned area

map 1	Complete coverage (%)			
	30	60	90	95
Type0	2	6	20	82
Type1	3	6	18	31
Type2	2	4	6	7
Type3	2	4	6	7

map 2	Complete coverage (%)			
	30	60	90	95
Type0	3	6	21	48
Type1	3	33	42	70
Type2	2	5	don't	don't
Type3	2	5	7	7

map 3	Complete coverage (%)			
	30	60	90	95
Type0	3	12	55	93
Type1	4	21	24	29
Type2	2	4	6	6
Type3	3	5	7	7

map 4	Complete coverage (%)			
	30	60	90	95
Type0	4	8	33	90
Type1	3	36	52	58
Type2	2	4	don't	don't
Type3	3	5	7	8

Fig. 7. Elapsed time to complete given coverages

example, as we can see in the figure, Type 2 and Type 3 operations are finished within 8 minutes while Type 0 and Type 1 show very gentle slope after 7 or 8 minutes so they need more than 60 minutes to reach 95% coverage of the given cleaning area.

We also measured the elapsed time to reach certain amount of cleaned area. As shown in Fig. 7, the random style algorithm (Type 0 and Type 1) and human-aided algorithm spend almost same amount of time to reach less than 30% of cleaning coverage but the elapsed time is growing exponentially if the required coverage is more than 50%. In the case of the most complicated map (d) in Fig. 5, Type 2 does not finish its task.[2] Since there is no guidance for the next cleaning section, the robot lost its way to go further, therefore the performance becomes worse than the pure random style algorithm. Also note that the elapsed time depends on how to bisect the area using two virtual walls in Type 1; we chose the better one in our experiment.

5 Conclusion

We have presented the human-aided cleaning algorithm and provided the simulation results that the behavioral performance of our style exceeds much higher than the random style cleaning algorithm like Roomba. The advantage of the random style cleaning robots is that we can implement them on a low-cost architecture comparing to the mapping style while the cleaning performance is practically useful. In this paper, we can build a cleaning robot with similar architecture but its behavioral performance can be greatly improved if human users can provide the cleaning space information to the robot. Based on our preliminary experimental results, we believe it is very possible to build a cleaning robot free from all the technological and economical difficulties like path planning and localization problems for the mapping style cleaning robots. We are going to further explore the simplification of basic robot instruction sets that can be implemented with the minimized sensing and driving modules. We are also working on the most convenient way of human-robot interactions that may lead to high cleaning performance.

References

1. Fiorono, P., Prassler, E.: Cleaning and Household Robots: A Technology Survey. Autonomous Robots 9, 51–92 (2000)
2. Elfes, A.: Sonar-Based Real-World Mapping and Navigation. IEEE Journal of Robotics and Automation RA-3(3), 249–265 (1987)
3. Oh, J., Choi, Y., Park, J., Zheng, Y.: Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map. IEEE Transactions on Industrial Electronics, 51(3), 718–726 (2004)
4. Tsai, C-C.: A Localization System of a Mobile Robot by Fusing Dead-reckoning and Ultrasonic Measurements, IEEE Instrumentation and Measurement Technology Conference, pp. 144–149 (1998)
5. Neira, J., Tardos, J., Horn, J., Schmidt, G.: Fusing Range and Intensity Images for Mobile Robot Localization. IEEE Transactions on Robotics and Automation 15(1), 53–76 (1999)
6. Fox, D., Burgard, W., Thrun, S.: Active Markov Localization for Mobile Robots, Robotics and Autonomous Systems, vol. 25 (1998)
7. Dellart, F., Fox, D., Burgard, W., Thrun, S.: Robust Monte Carlo Localization for Mobile Robots. In: Proc. Of National Conference on Artificial Intelligence, vol. 128 (2001)

[2] In map 2 (Fig. 5 (b)) map 4 (Fig. 5 (d)), Type 2 terminates its task at 86.4% and 73.82%, respectively.

8. Jones, J.: Robots at the Tipping Point: The Road to the iRobot Roomba. IEEE Robotics and Automation Magazine 13, 76–78 (2006)
9. Fortizzi, J., DiSalvo, C.: Robots Service Robots in the Domestic Environment: A Study of the Roomba Vacuum in the Home, ACM Annual Conference on Human Robot Interaction, pp. 258–265 (2006)
10. Gerkey, B., Vaughan, R., Howard, A.: The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: Proceedings of the International Conference on Advanced Robotics, pp. 317–323 (2003)
11. Collett, T., MacDonald, B., Gerkey, B.: Player 2.0: Toward a Practical Robot Programming Framework. In: Proceedings of the Australasian Conference on Robotics and Automation (2005)