The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Applying a Path Planner based on RRT to Cooperative Multi-robot Box-pushing

Takahiro Otani[1] and Makoto Koshino[2]

[1]Electronic and Mechanical Engineering Course
[2]Dept. of Electronics and Information Engineering
Ishikawa National College of Technology
Kitacyujo, Tsubata, Kahoku-gun, Ishikawa, 929-0392

Abstract

Considering the robot system in the real world, the multi-robot system that multiple robots work simultaneously without colliding with each other is more practical than the single-robot system that only one robot works. Therefore, solving the path planning problem on the multi-robot system is very important.

In this study, we develop a path planner based on the Rapidly-exploring Random Tree (RRT) which is a data structure and algorithm designed for efficiently searching for multi-robot box-pushing, and make experiments in real environments. A path planner must construct the plan without colliding with obstacles or other robots. Moreover, robots must collaborate with other robots to transport the box without colliding with obstacles in some cases. Our proposed path planner constructs the box-transportation plan and path plans of each robot in consideration of above.

Experimental results show that our proposed planner can construct the multi-robot box-pushing plan without colliding obstacles and that robots can execute tasks according to the plan in real environments. We also check out that multiple robots can perform tasks on the problem that only one robot cannot transport the box to the goal.

1 Introduction

Considering the robot system in the real world, the multi-robot system that multiple robots work simultaneously without colliding with each other is more practical than the single-robot system that only one robot works. Additionally, in the multi-robot system, each robot that collaborate with other robots positively can perform more advanced tasks than single-robot can do. Therefore, solving the path planning problem on the multi-robot system that is the most primal element task is very important. In this study, we develop a path planner based on the Rapidly-exploring Random Tree (RRT) for multi-robot box-pushing, and make experiments in real environments. The RRT is a data structure and algorithm that is designed for efficiently searching. The data structure is constructed incrementally in a way that quickly reduces the expected distance of a randomly-chosen point to the tree. The method is particularly suited for path planning problems that involve obstacles and differential constraints, and that have been applied to various path planning or motion planning problems for robots.

Traditionally, some path planners or behavior-based robot architectures for multi-robot box-pushing have been proposed. Kamio and Iba proposed the multi-agent cooperation path planning algorithm for object-transportation based on RRT, and succeeded to construct plans that three robots transport object cooperatively[1]. However, it was made experiments only in simulation environments with simplified model. Yamada and Saito proposed the behavior-based robot architecture with adaptive action selection without explicit communication for multi-robot box-pushing, and succeeded to transport the box to the goal in real environments[2]. However, the method unconsidered presence of obstacles in the environment.

A path planner must construct the plan without colliding with obstacles or other robots. Moreover, robots must collaborate with other robots to transport the box without colliding with obstacles in some cases. Our proposed path planner constructs the box-transportation plan and path plans of each robot in consideration of above.

We made experiments using compact circular mobile robot called " e-puck. " Experimental results show that our proposed planner can construct the multi-robot box-pushing plan without colliding obstacles and that robots can execute tasks according to the plan in real environments. We also check out that multiple robots can perform tasks on the problem that

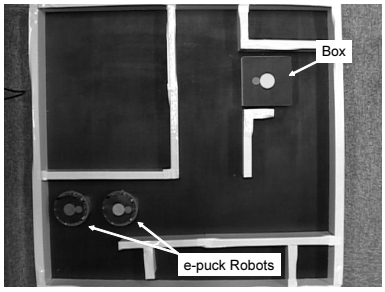

Figure 1: The environment used in this study.

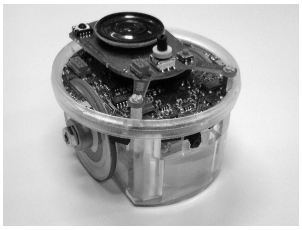

Figure 2: e-puck robot.

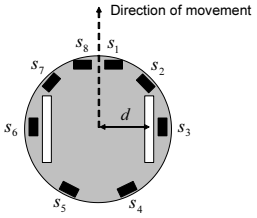

Figure 3: The model.

only one robot cannot transport the box to the goal.

2 Targeted Problem

The problem targeted in our research is the task that two robots collaboratively transport the box from the start position to the goal position. Figure 1 is the environment used in this study. We consider transporting the box located in upper right position to lower left position robot currently located in.

We use compact circular mobile robot called "e-puck" as the box-pushing robot. Figure 2 is the photograph of the robot, and Figure 3 is the model of it. The robot has eight infrared proximity sensors ($s_1 \ldots s_8$) and two wheels connected stepping motor directly, but it has not the dragging mechanism such as a gripper.

3 Proposed Planner based on RRT

In this study, we develop a path planner based on the Rapidly-exploring Random Tree (RRT)[3] for multi-robot box-pushing. The planner use RRT-ConCon[4] that is the extended RRT algorithm described in Figure 4. The scheme of the proposed planner is described in Figure 5. This method executes three steps below.

```
RRT-ConCon(x_init, x_goal)
  for k = 1 to K do
    x_rand := RANDOM_STATE();
    if not CONNECT(a, x_rand) = Trapped then
      if CONNECT(b, x_new) = Reached then
        return Path(a, b);
    SWAP(a, b)
  return Failure
```

Figure 4: The scheme of RRT-ConCon.

```
PLAN_MULTI-ROBOT_BOX-BUSHING(b, R)
 if PLAN_BOX_PUSHING(b) = Success
   T := GET_SUB_TASKS()
    for i = 1 to |T| do
      r := ALLOCATE_TASK(R, T, i)
      if CREATE_ACTION(r, T, i) = Success
        Add new action to r.actions
      else
        return Failure
    EXECUTE_PLAN()
    return Success
  else
    return Failure
```

Figure 5: The scheme of the proposed planner

(1) Constructing the Box-pushing Plan and Finding Sub-tasks

First, RRT customized for box-transportation construct the box-transportation plan from the start to the goal, and find sub-starts which is the position a robot starts to box-pushing and sub-goals which is the position a robot pass the box on to the other robot.

(2) Allocating Sub-tasks and Constructing path plans

Next, the planner allocates sub-tasks which is the box-pushing task from a sub-start to the next sub-goal to each robot, and constructs path plans from current position of robots to sub-starts using RRT-ConCon.

(3) Executing the constructed plans

Finally, robots execute tasks according to the constructed plans in the real environment. The robot executes movement behavior to the sub-start and box-pushing behavior to the sub-goal.

The movement behavior is executed as follows. The robot follows planned line according to the equation below.

$$\begin{cases} v_R = v + \Delta v \\ v_L = v \quad \Delta v \\ \Delta v = K_L L + K_{LD}\dot{L} + K_t\theta + K_{tD}\dot{\theta} \quad \Delta v_o \end{cases} \tag{1}$$

v_R is the speed of the right wheel, and v_L is the speed of the left wheel. L is the distance between the center

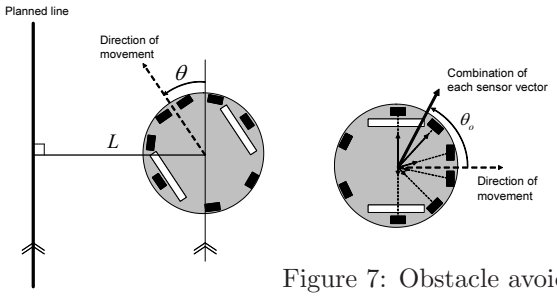

Figure 6: Line tracing.

Figure 7: Obstacle avoiding.

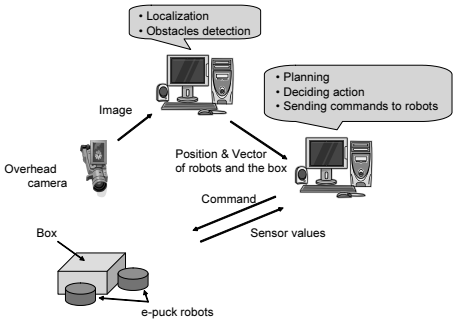

Figure 8: The system architecture.

of the robot and planned line, and θ is the angle formed by planned line and direction of movement. Figure 6 illustrates these parameters. K_L, K_{LD}, K_t, and K_{tD} are coefficients of each term. Δv_o is the offset value to avoid obstacles represented as the equation below.

$$\Delta v_o = \begin{cases} K_o\theta_o & \max(s_1, s_2, s_3, s_6, s_7, s_8) > \\ 0 & \text{otherwize} \end{cases} \quad (2)$$

θ_o is the angle formed by combination of each forehand sensor vector and direction of movement, and is illustrated in Figure 7. The sensor vector is defined as the vector from center of the robot to the sensor, and the magnitude is the output value of the proximity sensor. K_o is the coefficient of the offset value. s_i is the output value of proximity sensor i, and is the threshold. In addition, the robot adjust its position and angle when it is located in near the goal.

The box-pushing behavior is started after finishing of the adjacent box-pushing behavior. The robot wait ready at the sub-start position till then. The box-pushing behavior is executed in the same manner as the movement behavior. However, the offset value Δv_o in equation (1) is always 0.

4 Experiment

The system architecture is illustrated in Figure 8. The system consist of an overhead camera, two personal computers, and robots. The overhead camera and a personal computer are used to locate robots and the box, and to detect obstacles. The resolution of the camera is 640 480 pixels. Incidentally, Figure 1 is the image recorded by the camera. The maker mounted on each robot and the box is used to localize them. Additionally, the other personal computer constructs the box-transportation plan and path plans of each robot, and sends action-commands to each robot according to constructed plan. The robot support Bluetooth communication, so the personal computer sends action-commands through it.

Table 1: Success rate of the proposed planner.

	Percentage
Succeeded	98.7%
Failed to plan box-pushing	0.1%
Failed to plan paths	1.2%

4.1 Evaluation of planning paths

We made experiment to evaluate the performance of the proposed planner in the environment figured in Figure 1. The experiment was performed on a personal computer with a Core 2 Duo 1.8GHz processor and 1GB RAM, and was trialed 1000 times. The limit of the iterations in RRT-ConCon was set to 100000. We checked out the success rate of the proposed planner, length of paths of robots, number of sub-tasks, and computation time of planning.

Table 1 presents the success rate of the planner. This result shows that our proposed planner has a high probability of constructing executable plans in the environment used in this study. Table 2 presents each experimental result of executable plans. Incidentally, an example of the box-pushing plan is figured in Figure 9, and Figure 10 is an example of the path plan of the robots. The length of the path is 2120.9 pixels, and the number of sub-tasks is 6.

4.2 Evaluation of executing plans

We made experiment to check out whether the robot can execute constructed plans in real environment. Coefficients of equation (1) and (2) were set as follows; $v = 100$, $K_L = 10$, $K_{LD} = 50$, $K_t = 150$, $K_{tD} = 800$, and $K_o = 150$. The experiment was tri-

Table 2: Experimental results of planning paths.

	Av	SD	Min	Max
Length of paths (pixels)	2041.6	68.2	1843.4	2350.4
Number of sub-tasks	6.0	0.3	6.0	10.0
Computation time of planning box-pushing (ms)	1082.9	5220.9	36.0	67200.0
Computation time of planning paths (ms)	54.0	14.7	25.0	128.0

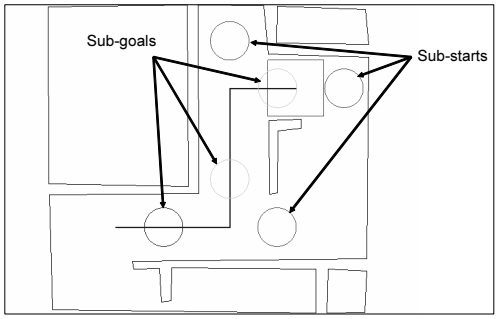

Figure 9: A box-pushing plan.

aled 10 times using the plan figured in Figure 9 and 10, and we checked out actual length of paths of robots and execution time.

Table 3 presents each experimental result of executing plan. The actual path's length is longer than constructed plan. The reason is that robots do not run on constructed line absolutely, but snake their way along the line according to equation (1). Incidentally, one of the actual paths of robots is figured in Figure 11.

5 Summary

In this study, we develop a path planner based on the RRT for multi-robot box-pushing, and make experiments in real environments. Experimental results show that our proposed planner can construct the multi-robot box-pushing plan without colliding obstacles and that robots can execute tasks according to the plan in real environments.

Table 3: Experimental results of executing plans.

	Av	SD	Min	Max
Length of paths (pixel)	2178.0	23.2	2140.7	2212.9
Execution time (s)	177.6	4.1	168.0	183.3

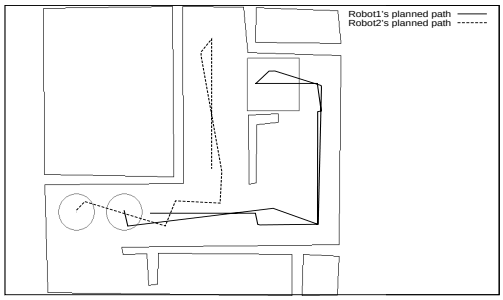

Figure 10: A path plan of two robots.

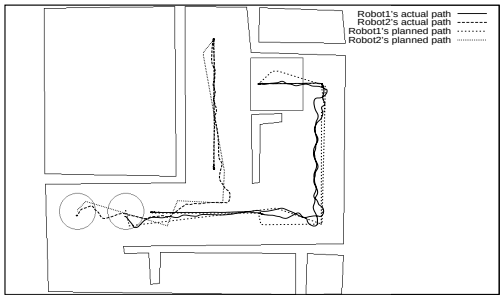

Figure 11: An actual path of robots.

References

[1] S. Kamio and H. Iba, "Multi-Agent Cooperation Path Planning Algorithm Based on Random Sampling," *The IEICE transactions on information and systems*, Vol. J89-D, No. 2, pp. 250-260, 2006.

[2] S. Yamada and J. Saito, "Adaptive Action Selection without Explicit Communication for Multi-Robot Box-Pushing," *Journal of the Robotics Society of Japan*, Vol. 17, No. 6, pp. 818-827, 1999.

[3] S.M. LaValle and J.J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and Computational Robotics: New Directions*, ed. B.R. Donald, K.M. Lynch, and D.Rus, pp. 293-308, A K Peters, Wellesley, MA, 2001.

[4] J. Kuffner and S.M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 995-1001, 2000.