

## Research Article

# Real-time Planning Robotic Palletizing Tasks using Reusable Roadmaps

Takumi Sakamoto<sup>1,\*</sup>, Kensuke Harada<sup>1,2</sup>, Weiwei Wan<sup>1,3</sup><sup>1</sup>Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka 560-8531, Japan<sup>2</sup>Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-3-36, Aomi, Koto-ku, Tokyo 135-0064, Japan<sup>3</sup>Intelligent Systems Research Institute, Advanced Industrial Science and Technology (AIST), 1-1 Umezono, Tsukuba 305-8560, Japan

## ARTICLE INFO

### Article History

Received 07 December 2019

Accepted 18 December 2019

### Keywords

Palletizing  
de-palletizing  
motion planning  
path planning  
PRM  
RRT\*

## ABSTRACT

This paper focuses on robotic motion planning for performing the palletizing or de-palletizing tasks. In such tasks, a robot usually iterates similar pick-and-place for several times. Considering such feature of the tasks, we propose two motion planning approaches named reusable Probabilistic Roadmap Method (PRM) and reusable Rapidly-exploring Random Tree Star (RRT\*) where both methods utilize the previously constructed roadmaps in the conventional PRM and RRT\*, respectively. We experimentally confirm that both methods significantly save the calculation time needed for motion planning compared to the conventional planning methods.

© 2020 The Authors. Published by Atlantis Press SARL.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## 1. INTRODUCTION

In a logistic process, human workers loads cargo boxes onto a pallet and to unload them from a pallet where such tasks are often called the palletizing and de-palletizing tasks, respectively (Figure 1). Since the palletizing tasks are known to be very labor intensive, this research proposes a robotic motion planning method for automating such tasks.

When a robot loads a lot of cargo boxes onto a pallet, a robot performs the pick-and-place for a number of times with slightly changing the initial and target configurations of the robot (Figure 2). Let us consider planning the motion of a robot performing the palletizing task by using a conventional sampling based motion planning method such as the Probabilistic Roadmap Method (PRM) and Rapidly-exploring Random Tree (RRT). Although the configurations of the robot and the environment around it are similar for each pick-and-place, a motion planner usually constructs a number of roadmaps and the constructed roadmaps are not used in the next pick-and-place. This results in unnecessarily increasing the calculation time needed for motion planning. Moreover, the calculation time of the random-sampling based motion planners such as PRM and RRT changes depending on each pick-and-place. It means that the calculation time may sometimes become extremely large if the roadmaps are placed at the wrong places in the initial planning phase.

To cope with this problem, this paper proposes an approach for robotic motion planning methods suitable for the palletizing tasks

by re-using the roadmaps constructed in the previous pick-and-place. We propose two planning methods where one is named the Re-usable PRM (R-PRM) and the other is the Re-usable RRT Star (R-RRT\*). In both methods, we consider reducing the calculation time by re-using the previously constructed roadmaps. In R-PRM, we construct a graph of roadmap in the first pick-and-place. After the second pick-and-place, we just modify the roadmaps located around the initial and the target object pose. In the R-RRT\*, we execute the RRT\* in the first pick-and-place. After the second pick-and-place, we use the solution path obtained in the first pick-and-place. We confirm that both methods enable us to significantly reduce the calculation time after the second pick-and-place.

The rest of the paper is organized as follows: after explaining the related works in Section 2, we explain the proposed methods in Section 3. In Section 4, we confirm the effectiveness of our approach through experiments by using an industrial robot.

## 2. RELATED WORKS

In robotic motion planning, random-sampling based methods such as the PRM and RRT have been widely used [1–4]. To plan a smooth path, post processors such as Geraerts and Overmars [5], Hauser and Ng-Thow-Hing [6] and the motion planning methods considering the asymptotic optimality such as the RRT\* [4] have been proposed.

While the conventional PRM and RRT assume the static obstacles, some methods on motion planner can be used under dynamically

\*Corresponding author. Email: [sakamoto@hlab.sys.es.osaka-u.ac.jp](mailto:sakamoto@hlab.sys.es.osaka-u.ac.jp)

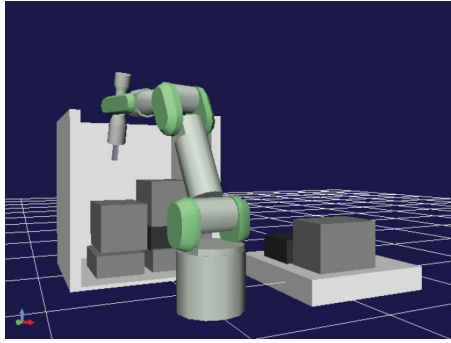


Figure 1 | Image of palletizing and de-palletizing tasks.

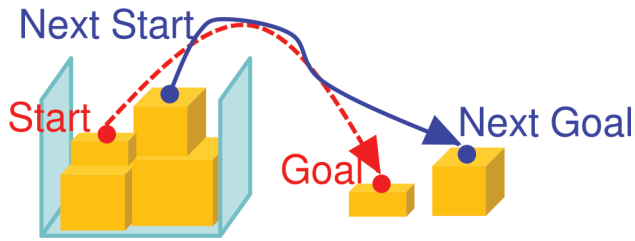


Figure 2 | In a robotic palletizing task, a robot iterates similar pick-and-place for a number of times.

changing environments. We can roughly divide the researches into two groups. In the first group, the roadmap graph is modified according to the change of environments [7–12]. Ferguson and Stent [10] proposed the anytime-RRT where they update an invalid part of the tree structure according to the change of environment. Anytime-RRT has been also used for the path planning of mobile robots [11,12]. On the other hand, the second group, the solution path is locally modified according to the change of environment [13,14]. Quinlan and Khatib [13] proposed a method for modifying the solution path according to the gradient of energy field. Combining the above two approaches, Yoshida et al. [15] proposed a method of humanoid motion planning that can deal with the change of environments.

On the other hand, this paper proposes a method of robotic motion planning for changing environments. Different from the previous approach, we consider utilizing a feature of the palletizing tasks where the environment only the position of the target object changes for each pick-and-place. Assuming such small change of environment, we consider modifying the roadmap graph.

### 3. PROPOSED MOTION PLANNERS

In this research, we propose two methods on robotic motion planning re-using the previously constructed roadmaps, i.e., R-PRM and R-RRT\*. First, we explain a method for collision checking introduced in this research in Subsection 3.1. Then, we explain the R-PRM and R-RRT\* in Subsections 3.2 and 3.3, respectively.

#### 3.1. Collision Detection

We tackle the motion planning problem of a robot moving in a dynamically changing environment. As an example of a dynamically

changing environment, we consider a situation where a robot performs the pick-and-place for a number of times. In each pick-and-place, a robot picks an object from the stack and put it at a designated place. After a robot performs a pick-and-place, the configuration around the robot changes only due to the moved object. Since the change of configuration is not large, it is not efficient to construct the roadmap graph from scratch for each pick-and-place. Rather, taking into account the configuration of moved object, we consider slightly modifying the previously constructed roadmaps. This subsection explains how to check collision around the part of the configuration space where the change occurred.

Figure 3 shows how to check collision and to modify the previously constructed roadmap graph where the newly appeared obstacle in the configuration space is marked in gray. Since it is almost impossible to precisely obtain the shape of newly appeared obstacle in the configuration space, we first assume a hyper sphere in the configuration space as shown by the green circle. We set the radius of the hyper sphere large enough to include the newly appeared obstacle. We consider checking the collision of the roadmaps constructed in the previous pick-and-place included in this hyper sphere. If the collision is detected, we consider removing the corresponding roadmaps and edges connecting these roadmaps from the roadmap graph.

#### 3.2. Re-usable Probabilistic Roadmap Method

We first explain the R-PRM for reusing the previously constructed roadmap graph. The overview of the algorithm is shown in Figure 4 where white and gray denote the free space and obstacle, respectively. Red and blue lines denote the previously found path and the path to be planned, respectively.

##### (i) Planning Initial Path

To plan the motion of a robot in the first pick-and-place, we use conventional PRM and memorize the constructed roadmap graph as shown in Figure 4a. The box placed at the designated place is regarded as the newly appeared obstacle when planning the next pick-and-place. By using the collision checking method explained in the previous subsection, we consider removing the inefficient edges and nodes included in the roadmap graph as shown in Figure 4b.

##### (ii) Planning Second and Subsequent Pick-and-place

To plan the motion of a robot in the second and subsequent pick-and-place, we consider re-using the previously constructed roadmap graph for the purpose of shortening the

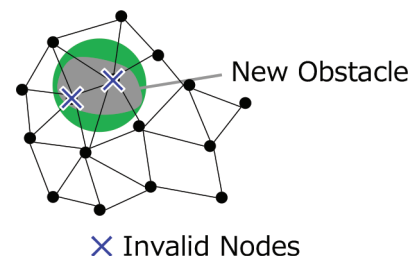


Figure 3 | Collision checking of roadmaps included in the hyper sphere.

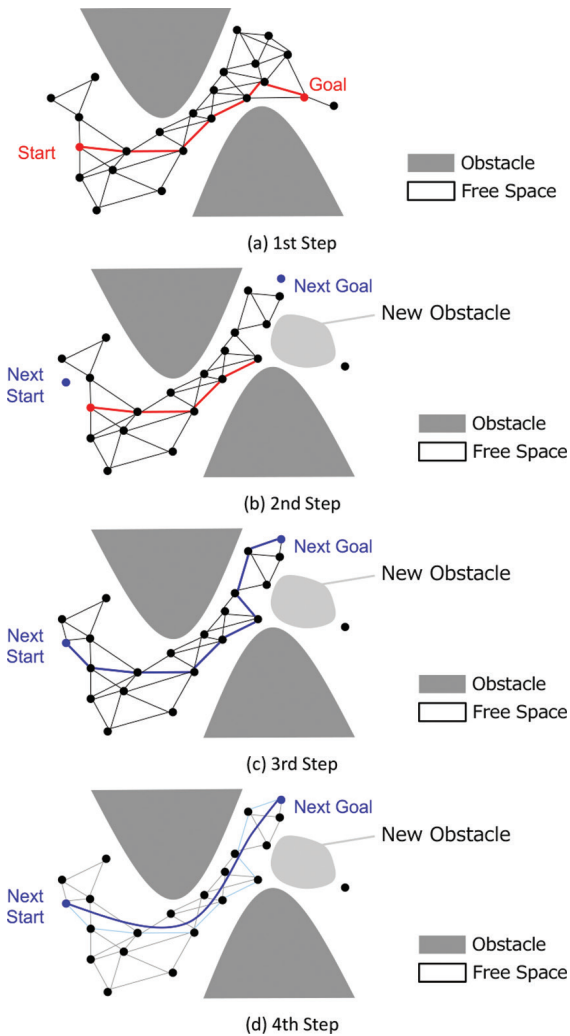


Figure 4 | Path planning using R-PRM.

calculation time. Connecting the new start and goal configurations to the roadmap graph, the solution path is searched as shown in Figure 4c.

### (iii) Path Smoothing

After finding the solution path, we apply the shortcut operation [6] for predefined times to smooth the solution path as shown in Figure 4d.

## 3.3. Re-usable Rapidly-exploring Random Tree Star

We next explain a single-query motion planner that re-uses the previously constructed roadmaps based on the RRT\* with satisfying the asymptotic optimality.

### (i) Planning Initial Path

To plan the motion of a robot in the first pick-and-place, we use the conventional RRT\* as shown in Figure 5a. The roadmap tree is constructed from both initial and goal configurations. After finding the solution path, we remove the initial and the target configurations from the solution path.

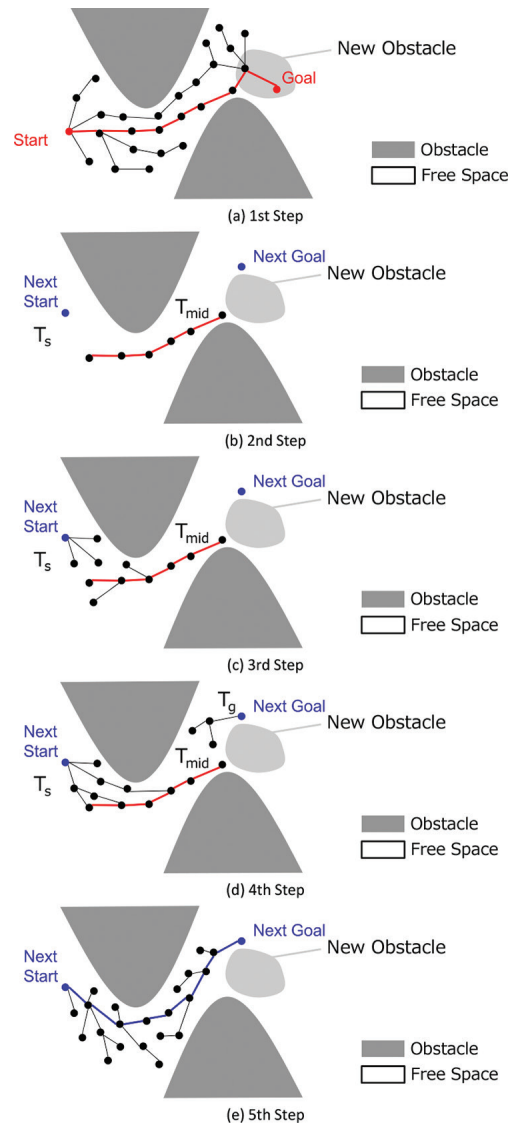


Figure 5 | Path planning using R-RRT\*.

Let  $T_{mid}$  be the obtained path. When planning the next pick-and-place, the object placed in the previous pick-and-place is regarded to be an obstacle. We remove the roadmaps causing collision from  $T_{mid}$  due to the newly appeared obstacle as shown in Figure 5b.

### (ii) Planning Second and Subsequent Pick-and-place

After the second and subsequent pick-and-place, we first connect  $T_{mid}$  with the new initial configuration, and then connect  $T_{mid}$  with the new target configuration. Let  $T_s$  and  $T_g$  be the roadmap trees rooted at the initial and the target, respectively. First, by expanding both  $T_s$  and  $T_{mid}$ , we try to connect  $T_s$  and  $T_{mid}$  by using the RRT\* as shown in Figure 5c. Then, by expanding both  $T_g$  and  $T_{mid}$  by using the RRT\* as shown in Figure 5d.

While we can guarantee the asymptotic optimality of the solution path by using the conventional RRT\*, our proposed R-RRT\* can approximately realize the optimal path if the new initial and target configurations are close enough to the previous initial and target configurations, respectively.

## 4. EXPERIMENTS

This section shows experimental results of our proposed methods.

### 4.1 Experimental Conditions

We performed the following three palletizing experiments.

- EX1 Arrange the picked objects from the back to the front as shown in [Figure 6](#).
- EX2 Arrange the picked objects from the front to the back as shown in [Figure 7](#).
- EX3 Place objects placed in a cage on the table under the table as shown in [Figure 8](#).

Snapshot of robot motion in EX2 is shown in [Figure 9](#).

In each experiment, we tested PRM, R-PRM, RRT\*, and R-RRT\*. In case of R-PRM and R-RRT\*, we planned the first pick-and-place by using RRM and R-RRT\* and planned the second and subsequent pick-and-place by using R-PRM and R-RRT\*, respectively. In each experiment, we conducted experiments on placing six objects for



**Figure 6** | Overview of EX1.



**Figure 7** | Overview of EX2.



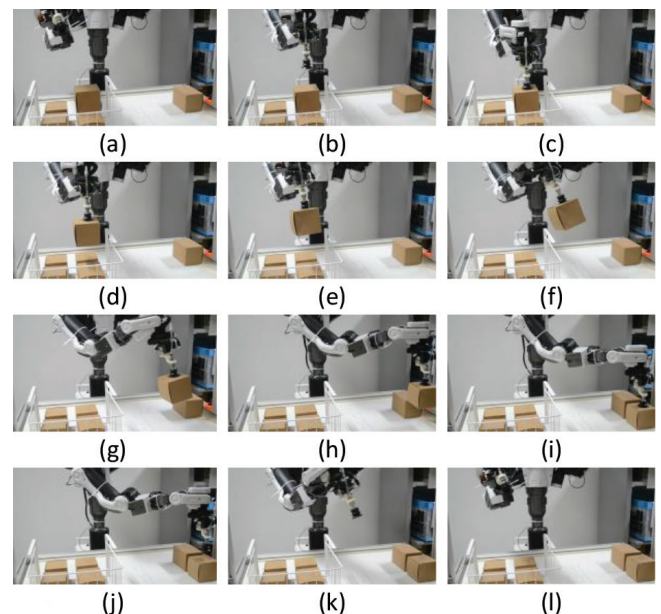
**Figure 8** | Overview of EX3.

five times. We especially show the result of motion planning from the grasping configuration ([Figure 9d](#)) to the placing configuration ([Figure 9h](#)). We used the uniform sampling in EX1 and EX2 while we used the Gaussian sampling in EX3 since a large number of obstacles are included. When planning the motion by using PRM and R-PRM, we sampled the configuration space until we obtain 50 milestones. We smooth the solution path by applying the shortcut operation [6]. When planning the motion by using RRT\* and R-RRT\*, we terminated the planning algorithm if 10 feasible paths are planned or the path is updated for 100,000 times in EX1 and EX2 and for 300,000 times in EX3.

We used the industrial robot Nextage [16] equipped with a suction gripper in the right hand. To pick a box, the suction pad is placed at the upper surface of a box. We coded the planning algorithms on the robot simulation environment Choreonoid [17] and sent the motion command from Choreonoid to Nextage. To plan the robot motion, we used the PC with eight 3.10 GHz CPUs.

### 4.2. Results

[Figures 10](#) and [11](#) show the comparison of calculation time between PRM and R-PRM and between RRT\* and R-RRT\*, respectively. In each figure, a–c show the result of EX1, EX2 and EX3, respectively. The vertical and the horizontal axes show the calculation time and the number of pick-and-place trial, respectively. We can see from these figures that the calculation time after second and subsequent pick-and-place has been improved in case of R-PRM and R-RRT\* where the calculation time is <400 (ms) for all the experiments. Variation of the calculation time appeared in PRM and RRT\* is reduced in R-PRM and R-RRT\*. In EX1 and EX2, all the algorithms did not fail in finding a solution. On the other hand, in EX3, PRM failed in finding a solution for nine times out of 30 times pick-and-place while R-PRM never failed in finding a solution. This is because R-PRM and R-RRT\* re-uses the previous roadmaps and likely to successfully find a solution.



**Figure 9** | Snapshot of robot motion in EX2.

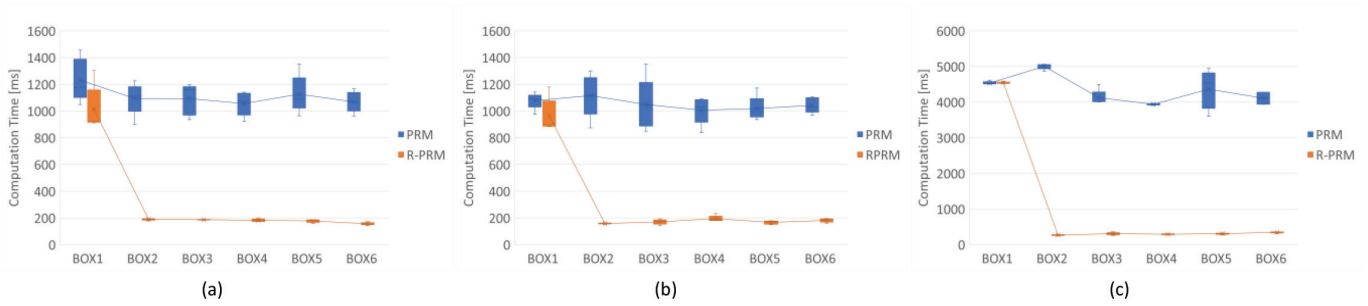


Figure 10 Results of PRM and R-PRM\*. (a) EX1, (b) EX2, (c) EX3.

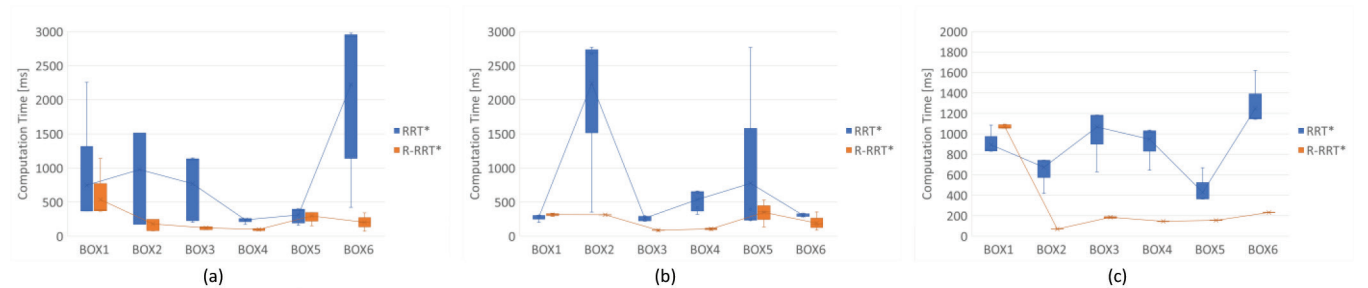


Figure 11 Results of RRT\* and R-RRT\*. (a) EX1, (b) EX2, (c) EX3.

## 5. CONCLUSION

In this paper, we proposed a robotic motion planning algorithms re-using the previously constructed roadmaps which is well applied for robotic palletizing tasks. We proposed two motion planning algorithms, i.e., R-PRM and R-RRT\* based on PRM and RRT\*, respectively. Through experiments, we confirmed that we can significantly reduce the calculation time and can reduce the variation of the calculation time if we use our proposed algorithms. Application of our motion planning algorithms to other tasks is considered to be our future research topic.

## CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

## REFERENCES

- [1] L.E. Kavraki, P. Svestka, J.C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Trans. Robot. Autom.* 12 (1996), 566–580.
- [2] S.M. LaValle, J.J. Kuffner, Randomized kinodynamic planning, *Int. J. Robot. Res.* 20 (2001), 378–400.
- [3] J.J. Kuffner, S.M. LaValle, RRT-connect: an efficient approach to single-query path planning, *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, IEEE, San Francisco, CA, USA, 2000, pp. 995–1001.
- [4] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, *Int. J. Robot. Res.* 30 (2011), 846–894.
- [5] R. Geraerts, M.H. Overmars, Creating high-quality paths for motion planning, *Int. J. Robot. Res.* 26 (2007), 845–863.
- [6] K. Hauser, V. Ng-Thow-Hing, Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts, *2010 IEEE International Conference on Robotics and Automation*, IEEE, Anchorage, AK, USA, 2010, pp. 2493–2498.
- [7] P. Leven, S. Hutchinson, A framework for real-time path planning in changing environments, *Int. J. Robot. Res.* 21 (2002), 999–1030.
- [8] L. Jaillet, T. Simeon, A PRM-based motion planner for dynamically changing environments, *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, IEEE, Sendai, Japan, 2004, pp. 1606–1611.
- [9] J. van den Berg, D. Ferguson, J. Kuffner, Anytime path planning and replanning in dynamic environments, *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, IEEE, Orlando, FL, USA, 2006, pp. 2366–2371.
- [10] D. Ferguson, A. Stent, Anytime RRTs, *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Beijing, China, 2006, pp. 5369–5375.
- [11] Y. Yang, O. Brock, Elastic roadmaps: globally task-consistent motion for autonomous mobile manipulation in dynamic environments, *Robotics: Science and Systems*, 2007.
- [12] Y. Kuwata, G.A. Fiore, J. Teo, E. Frazzoli, J.P. How, Motion planning for urban driving using RRT, *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Nice, France, 2008, pp. 1681–1686.
- [13] S. Quinlan, O. Khatib, Elastic bands: connecting path planning and control, *Proceedings IEEE International Conference on Robotics and Automation*, IEEE, Atlanta, GA, USA, 1993, pp. 802–807.
- [14] O. Brock, O. Khatib, Elastic strips: a framework for motion generation in human environments, *Int. J. Robot. Res.* 21 (2002), 1031–1052.
- [15] E. Yoshida, K. Yokoi, P. Gergondet, Online replanning for reactive robot motion: practical aspects, *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Taipei, Taiwan, 2010, pp. 5927–5933.
- [16] KAWADA Robotics Inc. Available from: <https://www.kawadarobot.co.jp>.
- [17] Choreonoid. Available from: <http://choreonoid.org/ja>.

## AUTHORS INTRODUCTION

### Mr. Takumi Sakamoto



Currently, he is a master student of Graduate School of Engineering Science, Osaka University.

### Dr. Kensuke Harada



He received the doctor of engineering from Graduate School of Engineering, Kyoto University in 1997. From 1997 to 2002, he was a research associate at Graduate School of Engineering, Hiroshima University. In 2002, he joined the National Institute of Advanced Industrial Science and Technology as a research scientist.

For one year from 2005, he was a Visiting Scholar of Computer Science Department, Stanford University. From 2016, he has been a professor at Graduate School of Engineering Science, Osaka University.

### Dr. Weiwei Wan



He received his PhD in Robotics from the Department of Mechano-Informatics, The University of Tokyo in 2013. He was affiliated with the Japan Society for the Promotion of Science (JSPS) from 2013 to 2015 and did his postdoc research at the Manipulation Lab in the Robotics Institute, Carnegie Mellon University.

He was on a tenure-track position at the Manipulation Research Group, Intelligent System Research Institute, National Institute of Advanced Industrial Science and Technology (AIST) during 2015–2017. From 2017, he has been an associate professor at the Graduate School of Engineering Science, Osaka University, Japan.