A Method of Online Motion Generation Using Swept Volumes Collected in Advance

Rui Zhu
Dept. of Engineering
Grad. School of Sci. and Tech.
Shinshu University
4-17-1 Wakasato, Nagano, Japan
Email: 18w4802g@shinshu-u.ac.jp

Kotaro Nagahama
Mech. Sys. Engineering
Faculty of Engineering
Shinshu University
4-17-1 Wakasato, Nagano, Japan
nagahama@shinshu-u.ac.jp

Keisuke Takeshita
Toyota Motor Corp.
1 Toyota-Cho, Toyota,
Aichi, Japan

Kimitoshi Yamazaki
Mech. Sys. Engineering
Faculty of Engineering
Shinshu University
4-17-1 Wakasato, Nagano, Japan
kyamazaki@shinshu-u.ac.jp

***Abstract*— This paper describes a method to efficiently generate a reaching motion to the target posture by using the experience of the past motion. The key point of the proposed method is the practical use of swept volumes. As pre-processing, various target postures of the end-effector are given under various environments, and robot movements to reach to the posture are generated. In each reaching motion, swept volume, which is the volume that the robot occupies in its workspace during the reaching motion, is recorded combined with the target posture of the end-effector. When a new target posture of the end-effector is given, past end-effector postures that are close to the new posture are searched. Then swept volumes associated with each posture are remembered, and the interference between each volume and the current environment is checked. This enables to select a feasible reaching motion with only one collision check procedure per one motion sequence. The effectiveness of the proposed method was confirmed by the experiments with a daily assistive robot in a simulated environment.**

***Index Terms*— Motion planning, Swept volume.**

I. Introduction

A robot equipped with an articulated manipulator can be expected to perform various tasks. In order to make good use of such a robot effectively, several functions are required. One of them is the function to decide the motion of the robot up to the given end-effector posture. In this paper, we deal with the method to obtain the joint angle sequence of the robot to reach the target posture of end-effector.

When a robot works to support people in daily life, it is desired to achieve a pick-and-place task required by them. Cleaning of dishes and the removal of items from storage cabinets are examples of such routine tasks. In most cases, these tasks should be performed in a situation that there is not only the item to be manipulated but also other objects exist. It is not always the case that the same objects are placed in the same number and the same positions every time. Therefore, in order to grasp the target item, it is necessary not only to simply plan the grasping posture for the item but also to obtain the motion sequence of the robot so as to avoid collision with the surrounding objects.

The purpose of this study is to establish a motion generation method that can be used in the situation described above. Although the same problem setting can be found in many past studies [1][2], we focus especially on speeding up motion decision processing in this study. Therefore, we propose a method that can quickly determine the reaching motion to achieve the target end-effector posture by making use of the experience of past motions. In the proposed method, the robot has experienced various reaching motions stored them as knowledge in advance. Then, as a target posture of end-effector is given, an appropriate motion is quickly determined on the spot, referring to the motion experienced in the past.

In general, collision check processing is essential in motion planning [3]. In the process, after the shape of the robot body and the shape of the surrounding environment are acquired, the collision state between them is investigated. This process requires more computational resources as the number of links and obstacles increases, or as the surrounding environment becomes more complex. For example, if an environmental map is provided in units of objects such as a table or a cup, the computational load on the motion generation increases as the number of objects increases.

The key word of the proposed method is “swept volume”. For example, let assume that a robot consisting of rotational joints and a base unit with wheels have moved from an initial state to a goal state. Swept volume is the sum of the space where the robot existed even once during the transition of the initial state to the goal state. Since collision check can be achieved by simply calculating “AND” of swept volume with the surrounding environment shape, it is very useful to know swept volume when performing motion selection. Several methods using swept volume have been proposed. For example, a method which to check the collision state of the robot using polyhedral swept volume can be found in [4]. Spherical swept volume has been also used for collision check [5]. As an extended method which to check the collision state can be more safely by expanding the swept volume with the buffer radius [6]. These methods have a problem that the computational cost is high. Therefore, a method to represent swept volume as cubic voxels and to check the collision state by combining the recursive division of voxels and hierarchical interference check [7] has been proposed. However, its experiments were limited to the simulation environment. To integrate the system with visual processing and to apply it to the actual work environment are future tasks.

As mentioned above, a motion is generated by referring to the motion performed in the past in this study. Swept volumes are calculated for every past reaching motion, and they are used during generating a motion for the present reaching task. If there are target postures in the past which are close to the present target posture, the purpose can be achieved by selecting a past motion and reproducing it. The important point is that it is sufficient only by a few collision checks between the swept volume and the present surrounding environment. Namely, the conventional procedure that generates robot posture at each time and performs collision check for each posture becomes unnecessary.

The contributions of this paper are as follows:

- We propose a method of motion generation for reaching task. Feasible robot motion can be found quickly.
- We represent a simple approach using the swept volume. It enables to reduce collision check which is the most burdensome process for motion planning.
- We clarified the effectiveness and the remaining issue of the proposed method through experiments using a mobile manipulator.

The structure of this paper is as follows. The next section describes related work, and Section III introduces the approach of this study. Section IV explains the proposed method in detail, and Section V presents the experimental results and discussion about the proposed method. Finally in Section VI, we conclude this study.

II. Related work

The approach to generate robot motion is called motion planning, many methods have been proposed [8]. In general motion planning, a robot motion that can avoid collision with surrounding environments is generated. Probabilistic Roadmaps (PRM) [9] and Rapidly-exploring Random Trees (RRT) [10] are famous and conventional methods. These methods have been extended actively, then various methods are proposed [11]. As these methods enable to find feasible solution at high-dimensional configuration space, they are suitable for robots having many DoFs such as articulated manipulators.

The conventional methods described the above assume that a feasible robot motion is generated from scratch at the search space which is generated for the present environment. Meanwhile, there are extended approaches that use past motion experience effectively. Leven et al. [12] and Kuwata et al. [13] proposed methods reusing or adjusting the result of past PRM. Perez et al. [14] and Ferguson et al. [15] also proposed methods reusing or adjusting the result of existing Trees generated by RRT. Yoshida et al. [16] proposed an online replanning method by parallelizing motion planning and actual motion of a robot.

There were also approaches that aim to learn robot motion from demonstration [17][18]. Most of them require to collect training data depending on a given task [19]. One advantage is that feasible robot motion is easy to obtain. To reduce the dependence to a given task, methods using dynamic planning or recurrence planning have been proposed [20]. However, these methods tend to increase computational load when surrounding environment is complex. On the other hand, another approach such as Hauser et al. [1] succeeded to reduce computational burden by confining collision check process in local area. On the alternate approach, a robot motion is built by combining motion primitives generated in advance [21]. It enables to reduce calculation costs. However, such approach involves another difficulty when decides the connection of the primitives. This needs additional computational cost. Recently, a method using deep learning have been proposed, and some researchers succeeded to plan a path for wheeled mobile robot [22][23]. A state-of-the-art method enables to generate motion for multi-DoFs articulated manipulator [24].

III. Approach

A. Problem setting and method outline

It is assumed that a target object for picking up is placed on a table. The challenge of this study is how quickly to determine the reaching motion of a robot that can move in a workspace including the target object. For this purpose, we propose a method to generate a motion of the robot using motion experienced in the past. There are mainly two processes; pre-process for data collection, and motion selection process for the present environment.

As an important pre-process for that strategy, reaching motions are planned at a variety of environments, and the results are registered in advance. The procedure is as follows. First, the position of a target object is variously changed, and robot motions are planned in each environment. Here, a conventional motion planning method is used. Then, the swept volume is generated from each motion. These results are represented by voxels. Finally, the target pose of an end-effector, a reaching motion, and a swept volume are registered as one group. In this study, we assume to use a mobile manipulator. This means that a motion consists of joint angles of a manipulator and a trajectory of a mobile base unit.

Meanwhile, the process for motion selection is as follows. First, a target object is randomly placed on a table. This determines the target pose of end-effector. Then, a 3-D point cloud of the object and surrounding environment is obtained by a 3-D range image sensor etc., and it is converted to a voxel map. These two results are input to the motion selection process, and a motion data to reach the target end-effector pose is selected from the registered dataset. As this process just searches a feasible motion from the dataset, it enables to obtain a feasible motion quickly.

The approach mentioned above is inspired by the following fact. For instance, when a human tries to pick up an object placed on a table, he/she will generate a reaching motion based on his/her past experience. Even if the placement of the object is the first look, and there are other obstacles, he/she can achieve the picking task by modifying some similar motions that he/she experienced in the past. That is, if the situation is the same, then a similar motion is always generated. Such

motions are kinematically sophisticated, such repeatability in itself is very important.

It is the same for situations to use robots. We will have many cases that similar robot manipulations are needed in similar situations. For the above reason, this study emphasizes the repeatability in motion planning. We aim to establish a method to obtain a feasible motion: if a robot has experienced an optimal or suboptimal motion in the past, the motion can be actively used.

B. Representation of Swept Volume

Swept volume is one representation to define an occupied space in a 3-D workspace. This representation largely influences the performance of methods using it. As mentioned in the above subsection, we introduce voxel representation. One conventional representation is elevation map which is called Digital Elevation Map (DEM) or 2.5D map [25][26]. However, it is difficult for DEM to represent 3-D concave shape, e.g. tunnels. As improvement of DEM, Triebel et al. [27] proposed MLSM (Multi-Level Surface Map). There are some extensions such as TCCM [28]. These representations allow infinite resolution in the vertical direction, and also make it possible to create 3-D concave shapes. On the other hand, collision check tends to complicate comparing with voxel representation.

As introduced in Section I, conventional form of swept volume is detailed mesh representation by polygons. While this is the approach prioritizing accuracy, it takes time to generate. On the other hand, although representing the occupied space of a robot by a voxel group is less accurate, the advantages of the approach are the compressible data form and easy for interference check. That is, if the environmental map is also expressed by voxels and the reference coordinate system of the robot and the environment map is the same, the interference check is completed only by examining whether there is an overlap of voxels. In this study, we choose voxel representation with emphasis on such ease. In addition, this representation is extensible to Octree representation [29]. It makes possible to speed up interference check and to increase the compression ratio.

IV. Motion Generation Method

A. Variables representations for the proposed method

Let Q be a robot motion obtained by motion planning. Q is a representation that the robot motion from start posture to goal posture is divided into M postures in time direction. It is represented as $Q = (\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_M)$. $\mathbf{q}$ which is data of each time is a vector consisting of posture parameters: e.g. $\mathbf{q} = (\theta_1, \theta_2, \cdots, \theta_N)$ for an articulated manipulator. As described in the experiment section, we consider using the proposed method with a mobile manipulator. In this case, let $\mathbf{q}$ be a vector connecting the posture parameters of mobile base unit for moving on the floor and joint angles of the manipulator.

Let $\mathbf{x} = (x, y, z, \theta, \phi, \psi)$ be a posture of end-effector. Environment map map is represented by binarized voxel space.

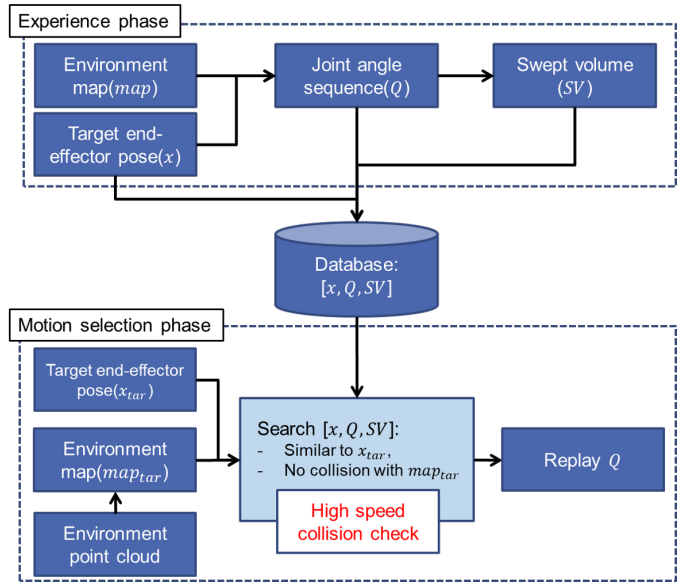

Fig. 1. The overview of the proposed method

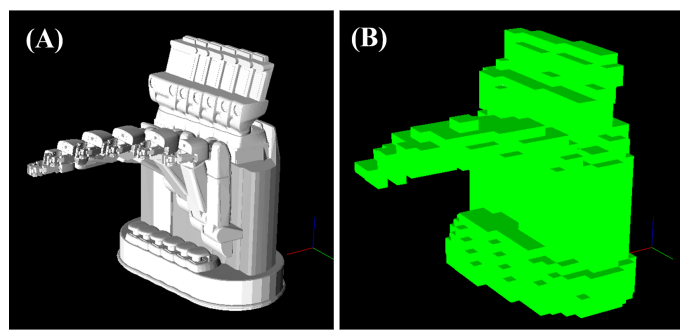

Fig. 2. Calculation of swept volume. The green voxels in figure (B) shows the SV

That is, from the 3-D point cloud obtained by measuring the surrounding environment, a voxel map is generated in which 1 is substituted for the part where the measurement point is included, and 0 otherwise. Furthermore, swept volume SV, which is the key of the proposed method, is also represented by a voxel group.

As already mentioned above, one key point to efficiently generate a motion Q is to make the origin of the reference coordinates of map and that of SV the same. Also, let the voxel composing SV be the same size as the voxel for the map. These settings permit to make direct interference check without coordinate conversion.

B. Training phase

Figure 1 shows the structure of the proposed method. It is composed of two phases: motion experience phase and motion selection phase. In this subsection, the former phase shown in upper side of Fig.1 is explained in detail.

First, environment map map and a target posture of end-effector $\mathbf{x}$ is given. Then, motion planning using a conventional method is applied in the situation, and a robot motion ensured to have no interference with the surrounding

environment is obtained. Next, the motion obtained is divided into M number of postures $\mathbf{q}$, then Q is obtained as a sequence of them. After that, $\mathbf{q}$ is sent to robot in order, an occupied space by the robot posture is calculated (Fig. 2 (A)). Finally, all of the occupied spaces are summed up then one swept volume SV is obtained (Fig. 2 (B)). This means that one swept volume is generated from one motion. The resulted from $(\mathbf{x}, Q, SV)$ is registered as one group.

The above procedure can also be performed in the simulation environment. That is, the robot model and the surrounding environment are virtually constructed, and the above processing is performed while changing map and $\mathbf{x}$ variously. This enables to collect a large amount of data without conducting actual experiments.

C. Motion selection phase

The lower part of Fig. 1 shows the flow of the motion selection phase. First, depending on the task, $\mathbf{x}_{tar}$ is given. Meanwhile, since a 3D point cloud obtained by measuring the current surrounding environment is given, a voxel map map_{tar} is generated. Next, using $\mathbf{x}_{tar}$ as a query, groups of data with similar $\mathbf{x}$ among the data collected in the previous section are picked up. Then SV is read out from the set of the data, and interference check with map_{tar} is performed. If there is no interference, the motion selection process is terminated by outputting Q linked the data.

Among the above-mentioned data, there may be multiple combinations of data to be selected by the search. The reason is that it is sufficient to choose the one which does not make interference among them. In addition, kd-tree is constructed before performing the search processing. By this way, it is possible to efficiently search for $\mathbf{x}$ which is closest to $\mathbf{x}_{tar}$.

D. Pose interpolation for accuracy compensation

In the proposed method, the quality of the obtained motion depends on the motions collected in the experience phase. That is, if the target posture of end-effector is set at small intervals and sufficiently detailed data collection is performed, it is possible to select an accurate reaching posture regardless of what kind of an end-effector posture is given in the motion selection phase. However, in such a case, the amount of data to be collected is large, which is undesirable from the viewpoint of memory consumption. On the other hand, if the interval of the end-effector posture is roughly made in the experience phase, the amount of data will be reduced. However, it is not possible to generate an appropriate reaching motion.

In this study, we assume to use a mobile manipulator. As such robot platforms permit to divide movable parts into a manipulator and a mobile base unit, the following fine-tuning method is possible for such trade-off problems. First, for a given end-effector posture $\mathbf{x}_{tar}$, $\mathbf{x}_{ref}$ nearby that is searched as a reference posture. Then, the difference $\hat{\mathbf{x}} = \mathbf{x}_{tar} - \mathbf{x}_{ref}$ is calculated. Let $\mathbf{q}_M^{ref}$ be the final posture of the robot for reaching to $\mathbf{x}_{ref}$. One of the tips here is to choose $\mathbf{x}_{ref}$ closer to the initial end-effector pose than $\mathbf{x}_{tar}$.

Then, a motion planning to achieve the transition from $\mathbf{x}_{ref}$ to $\mathbf{x}_{tar}$ is performed. In this planning, the conventional method used in the experience phase can also be used. As a result, the motion from $\mathbf{q}_M^{ref}$ to $\mathbf{q}^{tar}$ is smoothly connected.

In the above procedure, it takes more processing time than a simple selection of motion from the experience because the time for existing motion planning is added. However, $\hat{\mathbf{x}}$ is not so large. Therefore, the amount of motion required in the planning is small. Therefore, the motion planning will end quickly. This means that it is possible to obtain the reaching motion faster than directly using the existing motion planning method from the initial posture to the final posture.

V. Experiments

A. Settings

The proposed method was evaluated using the Human Support Robot (HSR, [30]) produced by Toyota Motor Corp. HSR has an arm with four degrees of freedom, a two-fingered end-effector, and a moving base with mechanism equivalent to omnidirectional mobile base unit. Xtion PRO LIVE, an RGB-D camera, is mounted on its head.

This experiment was performed using a simulation environment. In the initial state of the experiment, HSR was located at the initial position on the floor, and a table was placed in front of HSR. The height of the table was 470 mm. The target object was placed on the table, and HSR could observe the target object using its RGB-D camera. The target task for HSR was to reach, grasp and lift the object which was instructed by a user. For the robot-user interaction, the following interface was constructed and used for the experiment. The color image from the RGB-D camera of HSR was presented for the user. The user could instruct the target to the robot by clicking on the object in the color image. After the target object was instructed, HSR performs the above-mentioned process of motion selection phase. Namely, HSR selected the motion to reach an object without hitting the obstacle was selected from the stored motions, the motion was modified as described in Sec. IV-D, and the motion was executed.

The environmental map map_{tar} and the swept volume SV, which were used during the motion selection phase, were expressed as binary voxel maps. The environmental map map_{tar} was created at the beginning of the motion selection phase as a voxel containing at least one point of the 3D point cloud from the RGB-D camera was set as 1, and otherwise the voxel was set as 0. On the other hand, swept volume SV was created in the motion experience phase as follows: voxels intersecting with the swept volume of HSR were created as 1 and the other voxels were created as 0. In each case, the length of one side of the voxel was 100 mm, and the size of map_{tar} and SV was 1700 mm $\times$ 1400 mm $\times$ 1200 mm. However, the environmental map generated from the 3D point cloud in the motion selection phase contained the points of the target object itself. This meant, the target object could be judged as an obstacle, and motion selection could not be performed correctly. Therefore, the voxels corresponding to

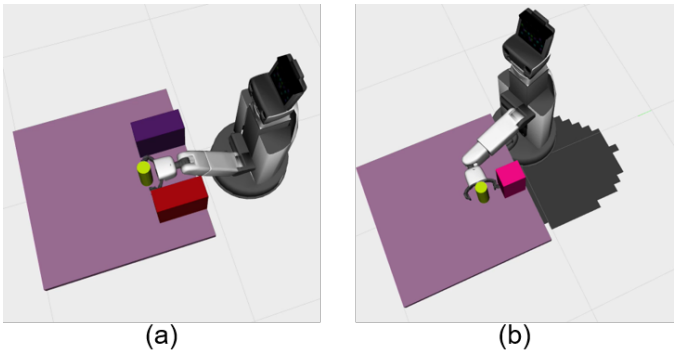

Fig. 3. The environments and robot postures during the training procedure

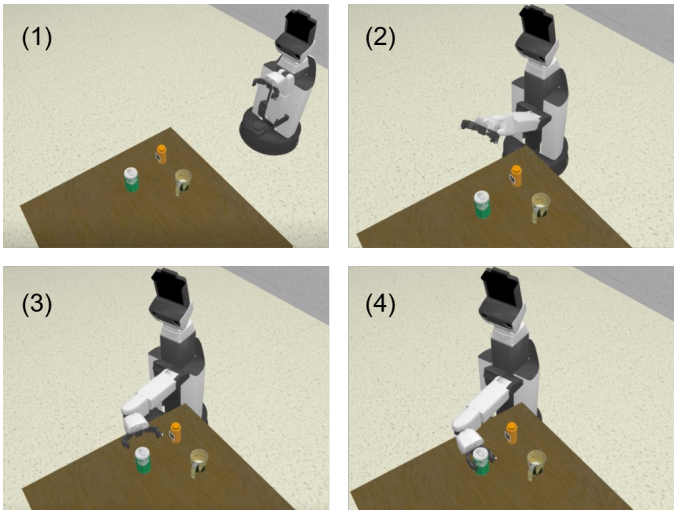

Fig. 4. An example of the reaching motion during the motion selection phase

the target area and 27 voxels around it were set to 0 before the motion was selected.

B. Training data collection

During the motion experience phase, the system used the simulation environment to generate motions to reach the target object located at various places on the table, and collected a large number of sets of $(\mathbf{x}, Q, SV)$. The target object had a cylindrical shape with a radius of 30 mm, and for each target object at each position, two types of motion were generated: (a) the motion in which the hand is moved from the robot's side to the back during the reaching, and (b) the motion in which the hand is moved from the right to left during reaching. The image (a) and (b) in Fig. 3 show the motion type (a) and (b), respectively. Note that prior to the motion experience (a), obstacles were placed on the 90 mm left and 90 mm right of the target object. Prior to the motion experience in (b), obstacles were placed 70 mm in front of the target object. In each case, the motion to reach the target object while avoiding obstacles was generated using RRT. Various motion data can be collected by generating motions in which the robot avoids obstacles near the target object in this way. This enables to select an action that avoids the obstacle in the motion selection phase. The position of the target object was varied by every 20 mm on the horizontal surface of the table top, and 500 pieces of data were acquired for each gripping method (a) and (b). This means a total of 1000 pieces of data were collected.

C. Motion generation results

First, 50 trials of reaching and grasping were conducted to evaluate the reliability of the method described in Sec. IV-C. At each trial, the user clicked and instructed the target object and HSR reached and grasped the target object. Figure 4 shows one of the trials in the experiment. The green can in the figure was the target object; the table, the yellow bottle, and the cup were obstacles. Figure 4 (1) is the initial state, and HSR succeeded in reaching and grasping the green can from (2) to (4). In this case, the proposed system was proved to select the trajectory that did not collide with the table or other objects. In the experiment, it was judged as successful if the target object was picked up without collision with an obstacle. When HSR collided with an obstacle during reaching, or HSR was not possible to pick up the target, it was regarded as a failure. As a result, 49 trials out of 50 trials were successful; the success rate was 98%.

Second, the computational time of (i) the conventional method using RRT, (ii) the proposed method described in Sec. IV-C, and (iii) the proposed method with pose interpolation described in Sec. IV-D were evaluated and compared. In this experiment, 50 trials of reaching and grasping were conducted to evaluate each method (150 trials all). As a result, the average time taken to calculate the appropriate motion was (i) 467 msec, (ii) 34 msec, and (iii) 169 msec, respectively.

D. Discussion

From experimental results introduced the above, it was clear that a fast motion generation for a mobile manipulator can be accomplished by the proposed method. Grasping a target object was accomplished with high success rate. This result tells us that the proposed method is useful for such situation. On the other hand, one failure occurred in 50 trials. Although grasping in itself was successful, the robot collided with other objects around the target object.

There seem to be two reasons. One is that obstacles are too close to the target object. As mentioned in Section V-A, voxels surrounding the target objects are set to 0 in motion selection phase. This has an effect to avoid a mistake that the target objects are regarded as obstacles. However, it is also a cause of ignoring obstacles in the vicinity of the object. Such failures can be reduced by reducing the size of voxels and generating a more detailed environment map.

The second cause is the motion accuracy of the robot. HSR is a mobile manipulator and involves movement of the mobile base unit during a reaching motion. In this motion, it is not always possible to move exactly along the target trajectory due to an error in self-localization and initial positioning error of the mobile base unit. Therefore, it is difficult to completely eliminate the possibility of collision. Such failures can be reduced by selecting a motion that allows for a sufficient distance between the robot and surrounding obstacles.

Furthermore, in the motion selection phase of this experiment, motion data was collected by the method of grasping from the front and that of from the right, respectively. By collecting additional motion data that grasps from the top, it is possible to adapt to more diverse environments.

In the second experiment, the motion selection with the proposed method described in Sec. IV-C was about 13 times faster than the conventional method. The calculation time with the method proposed in Sec. IV-D was only 2.7 times faster than the conventional method. However, the half of its calculation time (169 msec - 34 msec = 135 msec) was thought to be the time to calculate the interpolation of $\hat{\mathbf{x}}$. Note that the motion generation for $\hat{\mathbf{x}}$ and the motion execution from the initial pose to the $\mathbf{q}_M^{ref}$ can be executed in parallel. Because the motion execution requires much more time than the motion generation for $\hat{\mathbf{x}}$, the time for the robot to stop for the motion selection/generation for the method (iii) would be substantially 34 msec. With the proposed method, fast and delicate motion generation would be possible.

VI. Conclusion

In this paper, we described a motion generation method for mobile manipulators. The proposed method consists of two phases: the motion experience phase and the motion selection phase. The key of the method is swept volume. Swept volumes are calculated at various situations given at experience phase, and they are used for generating a motion for the present reaching task. If there are target postures in the past close to the present target posture, the purpose can be achieved by selecting a past motion and reproducing it. The important point is that it is sufficient only by a few collision checks between the swept volume and the present surrounding environment. The effectiveness of the proposed method was confirmed through a reaching task that a mobile manipulator grasps an object placed on a table. In our implementation, the time for motion generation was about 34 msec on average.

Future work includes performing experiments at the more complex environment. Compressing of motion experience data is also an important issue.

References

[1] K. Hauser: "Lazy Collision Checking in Asymptotically-Optimal Motion Planning," in Proc. of IEEE Int'l Conf. on Robotics and Automation, pp. 2951–2957, 2015.

[2] R. Menasri, A. Nakib, B. Daachi, H. Oulhadj, and P. Siarry: "A Trajectory Planning of Redundant Manipulators Based on Bilevel Optimization," Mathematics and Computation, pp. 934–947, 2015.

[3] F. Schwarzer, M. Saha, and J. C. Latombe: "Adaptive Dynamic Collision Checking for Single and Multiple Articulated Robots in Complex Environments," in IEEE Transactions on Robotics, vol. 21, No. 3, pp. 338–353, 2005.

[4] J. H. Tsai: "Collision Distance Detection Based on Swept Volume Strategy for Optimal Motion Plan," J. Adv. Mech. Des. Syst. Manuf., 3 (3), pp. 212–223, 2009.

[5] M. Tang et al.: "C2A: Controlled conservative advancement for continuous collision detection of polygonal models," in Proc. of the IEEE Int'l Conf. on Robotics and Automation, pp. 849–854, 2009.

[6] H. Tăubig: "Real-time swept volume and distance computation for self collision detection," in Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 1585–1592, 2011.

[7] T. Hasegawa, K. Nakagawa, K. Murakami: "Collision-free path planning of a telerobotic manipulator based on swept volume of teleoperated manipulator," in Proc. of IEEE Int'l Symposium on Assembly and Task Planning, pp. 259–263, 2003.

[8] M. Elbanhawi and M. Simic: "Sampling-Based Robot Motion Planning: A Review," in IEEE Access, vol. 2, pp. 56-77, 2014.

[9] L. E. Kavraki et al.: "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Transactions on Robotics and Automation, 12 (4), pp. 566–580, 1996.

[10] S. M. LaValle: "Rapidly-exploring random trees: A new tool for path planning," Technical Report. Computer Science Department, Iowa State University, pp. 1–4, 1998.

[11] J. J. Kuffner and S. M. LaValle: "RRT-Connect: An Effcient Approach to Single-Query Path Planning," in Proc. IEEE Int'l Conf. on Robotics and Automation, pp. 995–1001, 2000.

[12] P. Leven and S. Hutchinson: "A framework for real-time path planning in changing environments," Int'l J. of Robotics Research, vol. 21, no. 12, pp. 999–1030, 2002.

[13] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How: "Motion planning for urban driving using RRT," in Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 1681–1686, 2008.

[14] D. Ferguson et al.: "Replanning with RRTs," in Proc. 2006 IEEE Int. Conf. on Robotics and Automation, pp. 1243–1248, 2006.

[15] D. Ferguson and A. Stent: "Anytime RRTs," in Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 5369–5375, 2006.

[16] E. Yoshida, K. Yokoi, and P. Gergondet: "Online replanning for reactive robot motion: Practical aspects," in Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 5927–5933, 2010.

[17] A. Ude, C. G. Atkeson and M. Riley: "Planning of Joint Trajectories for Humanoid Robots Using B-Spline Wavelets," in Proc. of IEEE Int'l Conf. on Robotics and Automation, vol. 3, pp. 2223–2228, 2000.

[18] D. Costantinescu and E. A. Croft: "Smooth and Time-Optimal Trajectory Planning for Industrial Manipulators along Specified Paths," J. of robotic systems, 17(5), pp. 233–249, 2000.

[19] N. Melchior and R. Simmons: "Graph-based Trajectory Planning through Programming by Demonstration," in Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 1929–1936, 2012.

[20] J. Luo and K. Hauser: "Robust Trajectory Optimization Under Frictional Contact with Iterative Learning," Robotics: Science and Systems, vol. 41, Issue 6, pp. 1447–1461, 2015.

[21] H. Kato et al.: "Development of Robot Motion Performance Platform for Auto Generation of Robot Motion Planning," Proc. of SICE Annual COnf., pp. 1685–1690, 2012.

[22] W. Gao, D. Hsu, W. S. Lee, S. Shen, K. Subramanian: "Intention-Net: Integrating Planning and Deep Learning for Goal-Directed Autonomous Navigation," in Proc. of the 1st Annual Conference on Robot Learning, pp. 185–194, 2017.

[23] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart and C. Cadena: "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in Proc. of IEEE Int'l Conf. on Robotics and Automation, pp. 1527–1533, 2017.

[24] A. H. Qureshi, M. J. Bency, M. C. Yip: "Motion Planning Networks," arXiv:1806.05767, 2018.

[25] M. Herbert,. C. Caillas, E. Krotkov, I.S. Kweon, and T. Kanade: "Terrain mapping for a roving planetary explorer," Proceedings of International Conference on Robotics and Automation, pp. 997–1002, 1989.

[26] F. Nashashibi, et al.: "3-D Autonomous Navigation in a Natural Environment," Proceedings of International Conference on Robotics and Automation, pp.433–439, 1994.

[27] R. Triebel, et al.: Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2276–2282, 2006.

[28] K. Yamazaki, T. Mori, T. Yamamoto, M. Inaba: "Cuboid-Based Workspace Mapping and Plane Detection using Time-Series Range Data," in Proc. of IEEE Int'l Conf. on Robotics and Biomimetics, pp.583–590, 2011.

[29] N. Fairfield et al.: "Real-time SLAM with octree evidence grids for exploration in underwater tunnels," J. of Field Robotics, pp. 3–21, 2007.

[30] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase: "Development of the Research Platform of a Domestic MobileManipulator Utilized for International Competition and Field Test," in Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp. 7675–7682, 2018.