2017 13th IEEE Conference on Automation Science and Engineering (CASE)
Xi'an, China, August 20-23, 2017

Robot Programming for Manipulators through Volume Sweeping and Augmented Reality

Yasumitsu SARAI[1] and Yusuke MAEDA[2]

***Abstract*— Today's industrial robots require that human operators teach motions in advance. However, conventional methods for robot programming need deep knowledge and skills about robots or great effort for inputting information of working environment into computers. Therefore, a robot programming method in which everyone can easily teach robots "good" motions is demanded. For this purpose, our group proposed a robot programming method that uses manual volume sweeping by operators and automatic motion planning together to generate motion plans with short cycle times. Because a swept volume is the space through which the robot has passed without collision, it is movable space of the robot that can be used in motion planning. In this paper, we proposed using augmented reality in this programming method. We constructed a system in which operators can perceive obtained swept volumes and generated paths intuitively through augmented reality. Teaching experiments showed that non-skilled operators can make a robot move in shorter time than teaching/playback by direct teaching.**

I. Introduction

It is necessary for operators to teach motions that they want industrial robots to do in advance. This operation is called robot programming. Teaching/playback is a typical method of robot programming. In teaching/playback, a human operator teaches a robot a sequence of joint angles by moving it (for example, with a teaching pendant) and the robot plays it back. However, it is difficult for non-skilled operators to teach in this method because knowledge or experiences about robots are needed in order to teach "good" robot motions (for example, with short cycle times).

Direct teaching [1] or lead-through programming is a method to perform teaching/playback intuitively. In this method, human operators do not use teaching pendants but grasp the end-effectors of robots and move them directly. However, it is not different from the conventional teaching/playback in that operators have to think up robot motions by themselves. Thus it is still difficult for non-skilled operators to generate "good" robot motions. Although some other intuitive robot programming methods have been proposed [3], they have not been established yet. Therefore, a method in which everyone can easily teach "good" robot motions is demanded.

Offline programming [2] is also used for robot programming. In this method, actual robots are not needed and we can minimize the stop of production line. Automatic motion planning is actively studied (e.g., [4]–[7]), which can be used for offline programming. The optimal robot motion can be automatically generated by motion planning algorithms. Therefore, it can save the effort of operators to generate motion plans. However, this method is not easy to apply because there is a great care for inputting the exact information of obstacles around actual robots.

Hasegawa et al. proposed to regard the space through which a robot body passed as the movable space of the robot (swept volume) [8]. Our group applied Hasegawa's idea to programming industrial robots [9] [10]. In our method, an operator grasps the end-effector of a robot and moves it directly like direct teaching in order to obtain swept volumes. This method does not need knowledge and experience about the robot. The operator can generate "good" robot motions just by moving the robot without collisions with obstacles. Moreover, we constructed a system to perform calculation of swept volumes, display of them and motion planning totally online [10]. This enabled easier generation of the robot motions. However, because the swept volume is displayed on the PC screen as a CG model, it was difficult for operators to understand positional correspondence relation between the actual robot and swept volumes. As a result, redundant or insufficient volume sweeping tended to occur in our previous system.

In this paper, we propose using augmented reality (AR) to display swept volumes and generated paths to operators in the robot programming method through volume sweeping. This enables operators to perceive the shape of obtained swept volumes and generated paths intuitively and understand the relationship between these virtual entities and the actual robot.

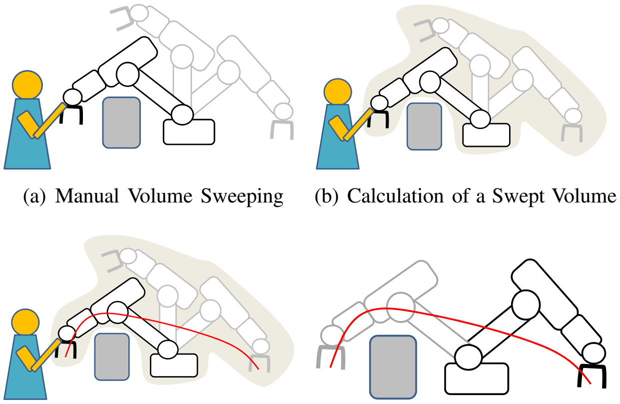
In the next section, we outline our robot programming method, robot programming by volume sweeping. In Section II, we describe the overview of our robot programming through volume sweeping. In Section III, we explain problems in our previous proposed method and benefits offered by introducing AR. In Section IV, we describe our experimental setup of robot programming. In Section V, we explain our implementation of AR. In Section VI, teaching experiments are carried out. In Section VII, we conclude this paper.

II. Robot programming method through volume sweeping

The procedure of the robot programming method by volume sweeping is as follows (Fig. 1):

1) An operator grasps the robot directly and moves it freely without making it collide obstacles (manual

[1]Y. SARAI is with Maeda Lab, Dept. of Mechanical Engineering, Div. of Systems Integration, Graduate School of Engineering, Yokohama National University, 79-5 Tokiwadai, Hodogaya-ku, Yokohama 240-8501 JAPAN.
[2]Y. MAEDA is with Division of Systems Research, Faculty of Engineering, Yokohama National University.



(a) Manual Volume Sweeping (b) Calculation of a Swept Volume

(c) Motion Planning (d) Execution of Planned Motion

Fig. 1. Overview of Proposed Robot Programming Method

volume sweeping). The robot is also taught its initial and goal configurations by the operator like direct teaching. Joint angles are recorded during manual volume sweeping (Fig. 1(a)).

2) A swept volume through which the whole body of the robot passed is calculated from the data of joint angles in 1) (Fig. 1(b)). Then, it is visualized on display as CG models.

3) The path from the initial configuration to the goal configuration is generated automatically based on the obtained swept volume by using a motion planning software (Fig. 1(c)).

4) The robot tracks the generated path (Fig. 1(d)).

In the system we developed, the motion planning software of 3) is activated periodically while 1) and 2) being done. Then, a robot motion is generated from the obtained swept volume, and its estimated cycle time is calculated. They are also displayed to the operator while the operator is moving the robot. The operator continues manual volume sweeping until he or she obtains a satisfactory motion plan. In this method, the operator can input the information of movable space of the robot to a computer implicitly as a result of manual volume sweeping and obtain a near-optimal path. Therefore, operators who have no knowledge and experience about robot programming can teach "good" robot motions easily.

III. Introducing Augmented Reality

In our previous implementation [10], an operator perceived obtained swept volumes and generated paths by seeing CG models on a PC monitor. Thus, the operator needed to understand where the space through which the robot body had passed was on the basis of the positional relation between the CG models of the robot and the swept volume. However, in such a way to perceive the obtained swept volume, it was not intuitive to understand the correspondence relationship between the actual robot and swept volumes in the virtual world. As a result, redundant and insufficient in volume sweeping tended to occur. Moreover, there were time and effort that they had to stop moving the robot to look at the PC monitor frequently.

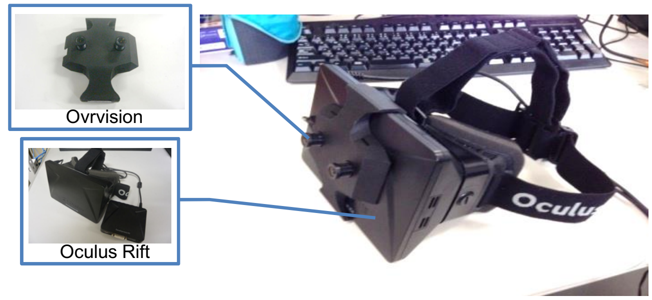


Fig. 2. Headset

In order to solve these problems, we implemented an AR display function by using a head mounted display (HMD) with a stereo camera to this robot programing system. In our proposed method, swept volumes and generated paths that overlay the actual robot on camera images are displayed on the screen of the HMD. This enables operators to understand the correspondence relationship between the actual robot and swept volumes easily and saves the effort of switching their gaze between the PC monitor and the robot frequently.

IV. Robot Programming Environment

A. Hardware Environment

We used Oculus Rift Development Kit 1 [16], an immersive HMD, and Ovrvision 1 [15], a stereo camera (the resolution is 640×480, the frame rate is 60 [fps]) dedicated to Oculus Rift, in order to display images to an operator by using AR. We constructed a video see-through headset by integrating them (Fig. 2). VE026A, a small 6-axis manipulator by DENSO WAVE (Fig. 3), was used as a target robot for programming. We used this small manipulator in our prototype system, unlike our previous paper [10] in order to ensure operators' safety in experiments. Necessary calculation for robot programming was performed on a Windows PC with a Intel Core i7-4790 (3.60GHz).

B. Software Environment

Motion planning is performed by Motion Planning Kit (MPK) [12]. MPK is a software using SBL (a single-query bi-directional probabilistic roadmap planner with lazy collision checking) as the motion planning algorithm [14]. A swept volume is used as the movable space of the robot and MPK generates a near-optimal path from the initial configuration to the goal configuration that an operator taught the robot during manual volume sweeping. A swept volume is expressed by an octree. The octree is obtained by collision check between each of robot bodies and a cuboid. We used collision check software SOLID3 [11] for calculating swept volumes.

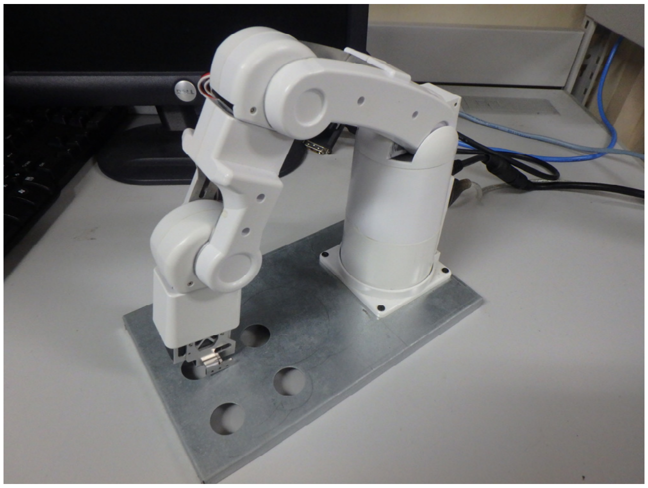


Fig. 3. Small 6-Axis Manipulator

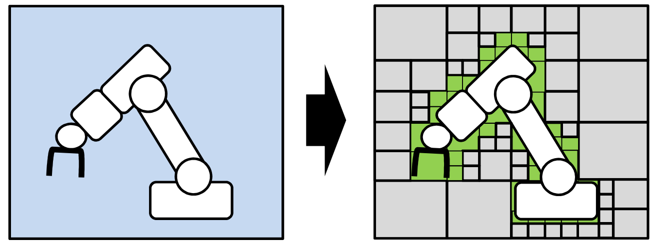


Fig. 4. Calculation of Swept Volume by Octree and Collision Detection

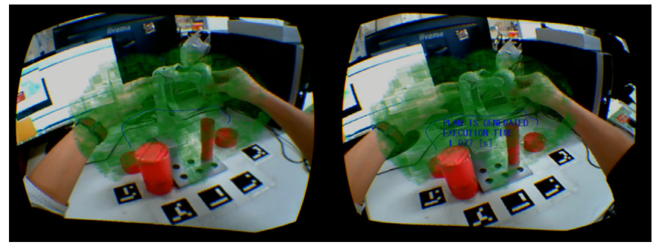


Fig. 5. Displayed Images for Headset (in AR Mode)

V. Displaying information by using AR

A. Displaying Swept Volumes by AR

We constructed a system that calculates swept volumes online from the data of joint angles of the robot obtained during manual volume sweeping and display the calculated swept volumes for the headset. This calculation is performed every time the configuration of the robot changes.

As we have described previously, a swept volume is represented by an octree. A cuboid that fully includes the workspace of the robot (580 [mm] $\times$ 580 [mm] $\times$ 430 [mm]) is used as the starting point in the calculation of swept volumes. The cuboid is divided into voxels recursively up to six times to represent a swept volume as an octree (Fig. 4). The swept volume is drawn for HMD as a virtual object of an aggregate of these voxels. Then, we display the obtained swept volume that overlays the actual robot by using AR like Fig. 5 so that operators are able to perceive the shape of the swept volume and where the obtained swept volumes are in the workspace of the robot. The right and left images in Fig. 5 are for right and left eyes, respectively, to enable stereoscopic views.

In order to display virtual objects by AR, it is necessary to compute the position and orientation of them in the real world. They are estimated by recognizing some AR markers allocated around the robot in the image of the camera. We used AR markers of ArUco [13]. Existence regions, edges, square corners and ID of the markers are extracted from binarized camera images. Then, the relative position and orientation between the camera and the markers are calculated from the coordinate values of corners in the image and intrinsic parameters of the camera, which was calibrated in advance. The relative position between the robot and each marker was also measured and inputted to our system in advance. Operators can perceive obtained swept volumes and perform manual volume sweeping simultaneously if any of markers is recognized by the system. Moreover, operators can understand swept volumes in conjunction with the real world.

B. Displaying generated paths by AR

We display paths generated by automatic motion planning so that they overlay the actual robot by using AR (Fig. 5). Motion planning is performed during manual volume sweeping. MPK is activated every 20 seconds to generate a path from the initial configuration to the goal configuration. The path is improved by smoothing three times. The generated path by MPK is displayed as a blue line. Moreover, an estimated cycle time of the generated path is displayed on the right-eye screen.

C. Using Virtual Reality

We implemented virtual reality (VR) mode in addition to the above-mentioned AR mode. Operators can always switch between the AR mode (Fig. 5) and the VR mode (Fig. 6) on the display of HMD. In the VR mode, we display not camera images but the virtual model of the robot. Operators can shift their viewpoint with a gamepad in the virtual world arbitrarily and change the orientation of the line of sight by changing their head orientation, which can be detected with Oculus Rift DK 1.

This VR mode complements the AR mode in that it enables operators to check swept volumes from physically-impossible viewpoints.

VI. Teaching experiment

A. Overview of Teaching Experiment

Teaching experiment was carried out to evaluate the usefulness of the proposed method. We compared it with the teaching/playback by direct teaching, in which everyone can program robot motions intuitively. The target task is pick-and-place: the robot picks an object at the initial position and places it at the goal position. However, we skipped making the hand tool open or close for the sake of brevity. Operators were eleven undergraduate and graduate students

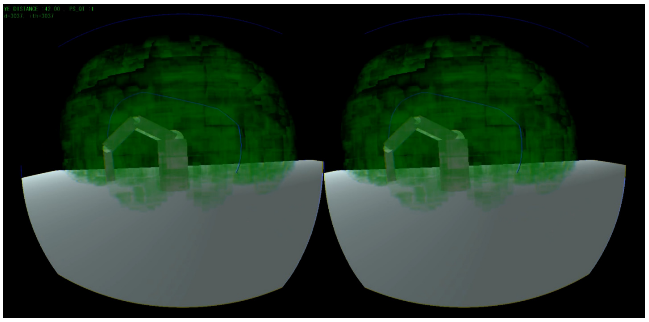


Fig. 6. Displayed Image for Headset (in VR Mode)

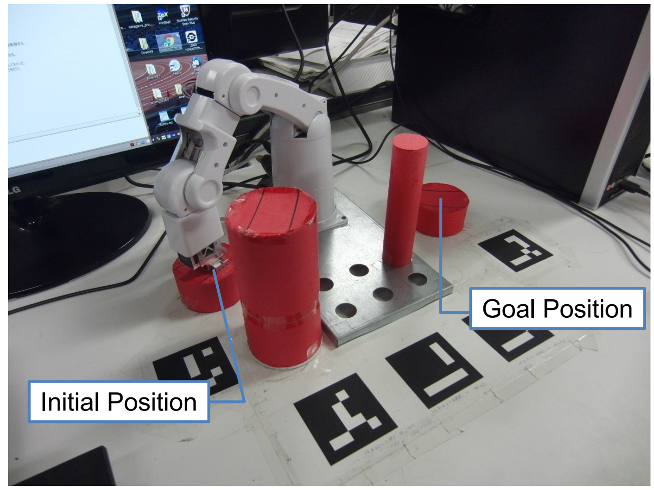


Fig. 7. Environment of experiment

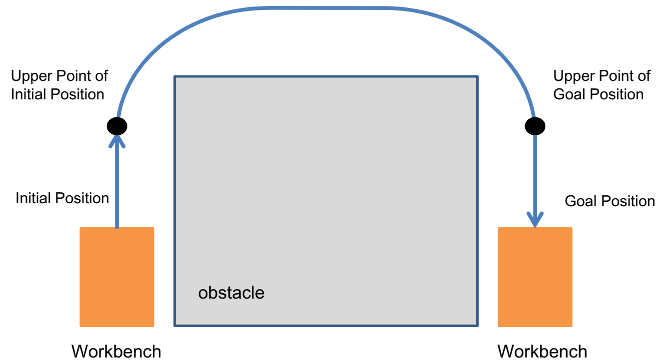


Fig. 8. Acquired Points in Proposed Method

who were not skilled at robot programing. The experiment was conducted in the environment showed in Fig. 7. In this environment, they had multiple options for programming a motion of the robot from the initial configuration to the goal. For example, the operators should consider whether they make the end-effector of the robot pass over the obstacles or by the side of them.

We compared robot programing with our proposed method and teaching/playback by direct teaching in term of the teaching time and the cycle time. Here the teaching time means the time to complete robot programming, and the cycle time means the time for execution of the programmed motion by the robot. Teaching operations were carried out for each operator in the following order: 1) direct teach 2) our proposed method 3) direct teach. The reason why direct teach was tested twice is that our proposed method may give the operators hints as to good robot motions. Therefore direct teach was carried out before and after our proposed method.

In teaching/playback by direct teaching, each of the operators grasped the end-effector of the robot and moved the robot from the initial position to the goal position along the path that is to be optimal subjectively. All the data of joint angles were recorded to reproduce them in playback.

In our proposed method, the operators inputted the joint angles of four positions: the initial position, the goal position, the upper point of the initial position and the upper point of the goal position, to the system at the beginning of teaching operation like Fig. 8. After that, the operator put on the headset and started manual volume sweeping (Fig. 9). Motion planning by MPK was carried out from the upper point of the initial position to the upper point of the goal position and motions were programmed by teaching/playback. This is to save time and effort of sweeping a volume near around the initial and goal positions, and ensure the accuracy of the position at the initial and goal configurations.

Operators decided when to finish volume sweeping based on generated paths and their estimated cycle times. After finishing manual volume sweeping, motion planning by MPK was performed ten times and the path with the shortest estimated cycle time was adopted to obtain stable results because the motion planning algorithm of MPK uses random numbers. Thus, we define the teaching time of the proposed method as a total of the time for ten-time motion planning and the time for manual volume sweeping.

B. Experimental Results

The result of the teaching times is shown in Fig. 10. The result of the cycle times is shown in Fig. 11. The teaching times of our proposed method were 26∼117 times longer (63 times on average) than those of the first teaching/playback by direct teaching and 29∼117 times longer (58 times on average) than those of the second teaching/playback by the direct teaching. However, the cycle times of our proposed method were 12∼49% shorter (30% on average) than those of the first teaching/playback by direct teaching and 8∼46% shorter (29% on average) than those of the second teaching/playback by direct teaching. There is clearly a difference between the teaching times of each method.

For the cycle time, we evaluated whether there are significant differences between the average cycle time in each teaching method or not by t-test with 1% level of significance.

As a consequence, no significant difference was found between the average cycle time of the first teaching/playback by direct teaching and that of the second teaching/playback by direct teaching. However, it was demonstrated that the average cycle time of our proposed method is significantly

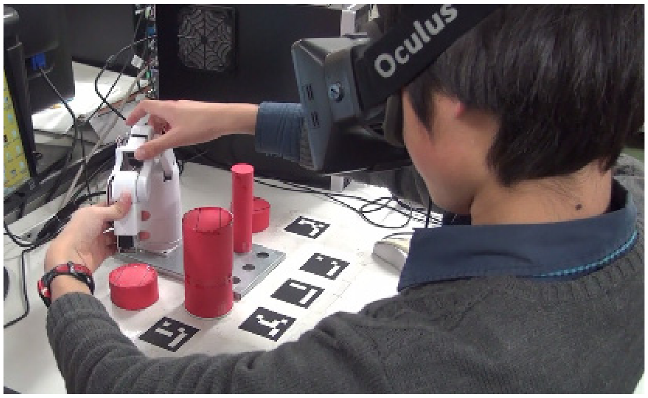


Fig. 9. Teaching Operation

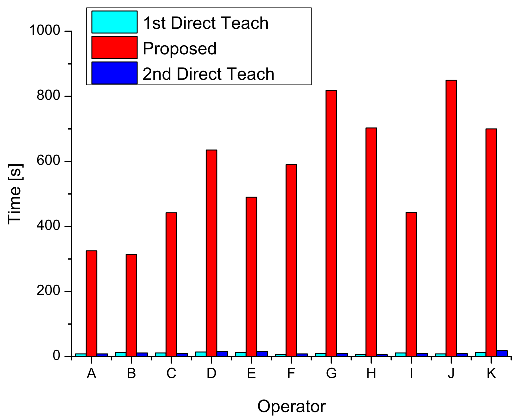


Fig. 10. Teaching Times

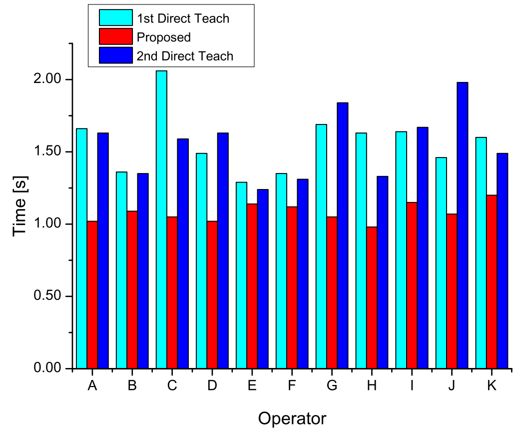


Fig. 11. cycle times

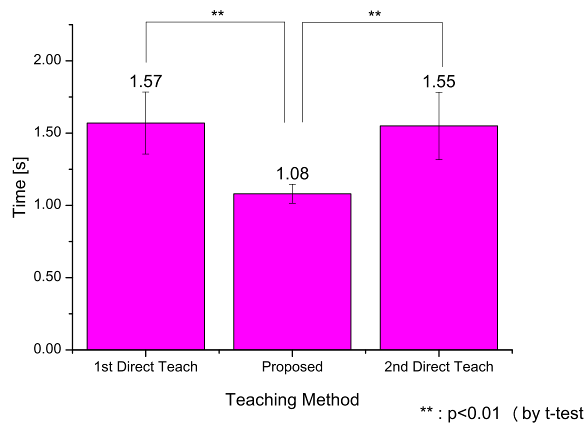


Fig. 12. Average of cycle time

shorter than those of the first and second teaching/playback by direct teaching (Fig. 12).

Table I is the result of the calculation of swept volumes in our proposed method. It shows the number of nodes in octrees and swept volumes in quantity in the end of volume sweeping.

C. Discussion

In our proposed method, the working time of the operators was much longer than teaching/playback by direct teaching. That is because the operators need to move the robot not to track only a single path but to sweep a necessary volume. However, the cycle time was shorter in our proposed method. Therefore, it can be said that our proposed method can generate "good" motions.

Fig. 13 and Fig. 14 show robot motions generated by the operator C in the first teaching/playback by direct teaching and in our proposed method, respectively. When the robot executed the motion generated in the first teaching/playback by direct teaching, the path of its end-effector from the initial position to the goal position was nearly the shortest. On the other hand, when the robot executed the motion generated in our proposed method, the robot moved over the obstacles as its end-effector went a longer way than Fig. 13. However, its cycle time was 49% shorter than that of teaching/playback by direct teaching. This is an example that the shortest end-effector path does not mean "good" motion with short cycle time. Thus it is difficult for non-skilled operators to generate "good" robot motion in conventional teaching/playback taking robot kinematics and joint specifications into consideration. On the other hand, even non-skilled operators can program robot motions with short cycle time in our proposed method.

Our proposed method requires improvement to reduce burdens to operators, by shortening the teaching time. One of the main reasons for such a long teaching time is that obtained swept volumes are slightly smaller than actual ones. We use a smaller robot model than actual in calculating swept volumes to avoid collisions between the robot and the obstacles. That is because the resolution of swept volumes is limited by a computing performance of the system. Therefore, if we can calculate swept volumes with higher resolution, we will be able to use a robot model as large as the actual robot and obtain larger swept volumes. Then the teaching time would be shorter. Therefore, it is necessary to speed up calculation of swept volumes in order to raise their resolution.

TABLE I
CALCULATED SWEPT VOLUMES

Operator	# of nodes in octree	swept volume [m^3]
A	59474	0.0333
B	48501	0.0272
C	37486	0.0211
D	59035	0.0279
E	39607	0.0223
F	44519	0.0250
G	49162	0.0276
H	49554	0.0278
I	52802	0.0296
J	37867	0.0214
K	52092	0.0292

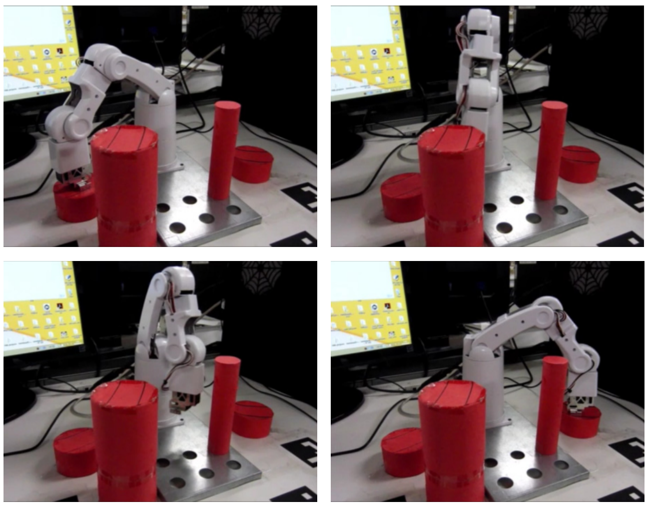


Fig. 13. Execution of Motion Programmed by Operator C (1st Teaching/Playback by Direct Teaching)

From Fig. 10 and Table I, longer teaching times do not always lead to larger swept volumes. From this result, it is thought that there were reduplications of volume sweeping. One of the reasons for this is that the non-swept volumes surrounded by swept volumes are hard to see in the present graphic model. Therefore, we are trying to make graphic models of AR easier to grasp such non-swept volumes.

In addition, we are trying to introduce markerless AR to the system in order to increase its usability.

VII. Conclusion

We proposed a method of robot programming through volume sweeping and augmented reality. The experimental results on our prototype system showed that non-skilled operators can generate "good" robot motions whose cycle times are short. In the future, it is necessary to improve the resolution of swept volumes and diminish burdens to operators in teaching operations.

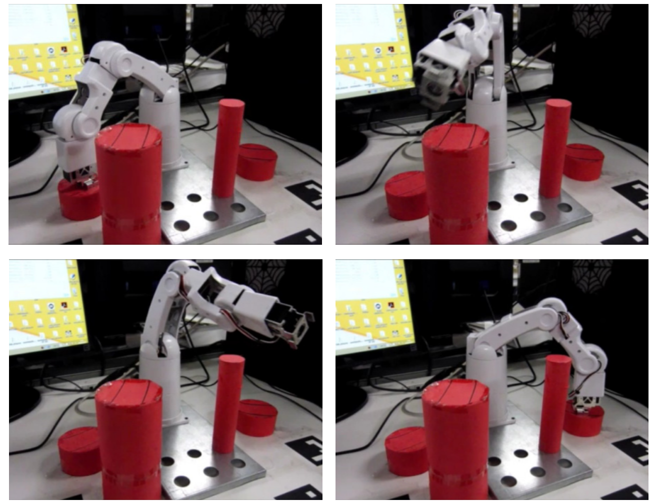


Fig. 14. Execution of Motion Programmed by Operator C (Proposed Method)

References

[1] H. Hirabayashi, K. Sugimoto, S. Arai, and S. Sakaue, "Virtual compliance control of multiple degree of freedom robot," *Trans. of Soc. of Instrument and Control Engineers*, vol. 22, no. 3, pp. 343-350, 1986, in Japanese.

[2] Y. F. Yong, J. A. Gleave, J. L. Green, and M. C. Bonney, "Off-line programming of robots," in *Handbook of Industrial Robotics*, S. Y. Nof, Ed. Wiley, 1985, pp. 366-380.

[3] C. Ioanes, and A. Chioreanu, "Current trends regarding the intuitive programming of industrial robots," *Acta Technica Napocensis Series: Applied Mathmatics and Mechanics*, vol. 55, no. 1, pp. 187–192, 2012.

[4] S. Bjorkenstam, et al., "Energy efficient and collision free motion of industrial robots using optimal control," in *Proc. of IEEE Int. Conf. on Automation Science and Engineering*, 2013, pp. 510–515.

[5] A. Gasparetto and V. Zanotto, "Optimal trajectory planning for industrial robots," in *Advances in Engineering Software*, vol. 41, no. 4, 2010, pp. 548–556.

[6] J. Schulman, et al., "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," in Robotics: science and systems, 2013.

[7] M. Zucker, et al., "CHOMP: Covariant Hamiltonian optimization for motion planning," in *The International Journal of Robotics Research*, 2013, vol. 32, no. 9-10, pp.1164–1193.

[8] T. Hasegawa, K. Nakagawa, and K. Murakami, "Toward on-line transition to autonomous teleoperation from master-slave manipulation," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 3715-3720.

[9] Y. Maeda, T. Ushioda, and S. Makita, "Easy Robot Programming for Industrial Manipulators by Manual Volume Sweeping," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 2234-2239.

[10] S. Ishii and Y. Maeda, "Programming of robots based on online computation of their swept volumes," in *Proc. of IEEE Int. Symp. on Robot and Human Interactive Communication*, 2014, pp. 385-390.

[11] "SOLID3", `http://www.dtecta.com/`.

[12] F. Schwarzer et al., "Motion Planning Kit", `http://ai.stanford.edu/~mitul/mpk/`.

[13] S. Garrido-Jurado et al., "ArUco," `http://www.uco.es/investiga/grupos/ava/node/26`.

[14] G. Sánchez and J.-C. Latombe, "A single-query bi-directional probabilistic roadmap planner with lazy collision checking," *Springer Tracts in Advanced Robotics*, vol. 6, pp. 403-418, 2003.

[15] "Ovrvision1," `http://ovrvision.com/setup/`.

[16] "Oculus Rift Development Kit 1," `https://developer.oculus.com/documentation/`.