Realizing an assembly task through virtual capture

Damien Petit[1*] Ixchel G. Ramirez-Alpizar[2]
Wataru Kamei[1] Qiming He[1] Kensuke Harada[1,2]

***Abstract*— Modern manufacturing strategy requires the robotic infrastructure to be able to adapt to new products or to accomplish new tasks quickly. In order to respond to this demand, teaching a robot to realize a task by demonstration has regained popularity in recent years, especially for dual-arm or humanoid robots. One of the main issues using this method is to adapt the captured motion from the human demonstration to the robot's specific kinematics and control. In this paper we present a method where the motion and grasping adaptation is tackled during the capture. We demonstrate the validity of this method with an experiment where a humanoid robot realizes an assembly previously demonstrated by a user wearing a Head Mounted Display (HMD) performing an assembly task in a virtual environment.**

I. Introduction

Robots have been widely used in manufacturing to perform repetitive and tiresome unique tasks since the beginning of mass manufacturing. In recent years, multipurpose robots have become more and more efficient and affordable. This allowed an evolution in modern manufacturing, where not only highly repetitive unique tasks but also tasks containing variability can be accomplished by the same robot [1]. The new challenge brought by this evolution is to be able to teach the robots to accomplish those new tasks quickly and efficiently.

Programming a robot to realize a new task from beginning to end, taking into account motion planning, control, etc., can be expensive and time consuming. One of the most natural ways for a worker to teach a robot to realize a task is to demonstrate this task so that the robot can replicate it. This process called programming by demonstration [2] [3] has been approached in different ways and applied in wide areas. Lau et al. [4] approached this problem from a machine learning point of view to adapt applications. Aleotti et al. [5], [6] used a virtual environment to make a motion demonstration to be later perform on the real robot. The major difference with our work is that the demonstration by the user is made without taking into account the robot kinematics, which we take into account. Also in the work of Aleotti et al. ([5], [6]), the robot's motion is generated by analyzing the obtained data through a task recognition process which is not needed with our approach.

[1] Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama-cho, Toyonaka, 560-8531, Japan.
[2] Automation Research Group, Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo, 135-0064, Japan.
* Corresponding author e-mail: damien.petit@hlab.sys.es.osaka-u.ac.jp

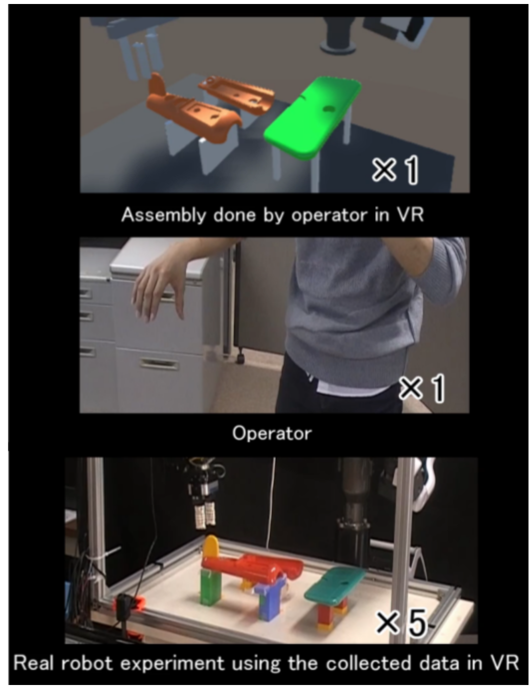

Fig. 1: This is the main setup used to capture the user motion in the virtual environment. The captured motion is then later played back onto the real robot.

In this paper, we developed a virtual environment containing the parts to be assembled and the humanoid robot used to realize the task. The main set-up is shown in Figure 1. The virtual humanoid robot has been carefully modelled so that it has the same kinematics as the real robot used to realize the assembly in the real world, which is one of the main contributions of this work. It is also worth noting that the humanoid robot is equipped with a two-finger gripper. Therefore another contribution of this work is to adapt the hand grasping motion of the user (captured through a visual sensor device) to the grasping motion of the two-finger gripper, where a special feature called "virtual grasping" has been developed to render the grasping of the virtual object more realistic. Finally, we confirmed the validity of this approach with an experiment where the user equipped with a Head Mounted Display (HMD) demonstrates the assembly of two parts of a toy airplane in a virtual world by controlling the virtual humanoid robot. The motion is set to be captured when a play button is pressed by the user, then, the captured motions is played back onto the real robot using the real toy airplane for realizing the desired task.

This paper is organized as follows. In section II the main contributions of this paper are explained. First we explain how we capture the human motion, then we describe the model of the virtual robot, after we explain how to adapt the human grasp to the two-finger gripper grasp and how to make the virtual grasping realistic. Finally, we show how to use the captured data to realize the experiment on the real robot. In section III, we describe the experiment carried out to validate our approach. Finally, in section IV, we discuss the results of the experiments and future work to improve this approach and its results.

II. Framework

In this section, we describe the main functionalities and contributions of this work. First, we describe the virtual robot model and its kinematics to obtain a faithful virtual rendering of the real robot appearance, dimensions and motions. Then, we describe how we adapted the human grasping motion to the two-finger gripper of the real robot. Finally, we explain how we played back the captured motion into the robot to realize the same assembly task.

A. Virtual capture

The capture of the user's hand motion during the demonstration is done using the vision capture device Leap Motion Controller (LMC)[a]. In order to improve the performance of the capture and make the user feel more comfortable, the capture device is attached on the Head Mounted Display (HMD) wore by the user. The user wearing the different devices and a rendering of the virtual robot can be seen in Figure 2. In order to facilitate the integration of the captured device and the deployment of our software after completion, the virtual environment has been developed using the unity framework[b].

With our approach only the hands and fingers' motions are necessary to capture. Thus, the hardware needed is easy to setup and the user fatigue is minimum during the demonstration.

During the demonstration, the motion data relative to the virtual robot are recorded to be later played back onto the real robot.

B. Virtual robot

In our previous work [7], we built a framework where the captured data corresponded only to the hands and fingers motion of the user without taking into account the kinematics of the robot. The adaptation from the human motion to the humanoid motion was done afterwards through the Choreonoid framework[c]. The results were mixed due to the difficulties of adapting the human motion to a motion compliant to the kinematics and control of the humanoid robot. That is why in this work, we chose to adapt the demonstrated motion during the capture in order to played the motion back to he humanoid robot in a straightforward way. In order to realize this, the virtual robot must have the same kinematics as the real robot.

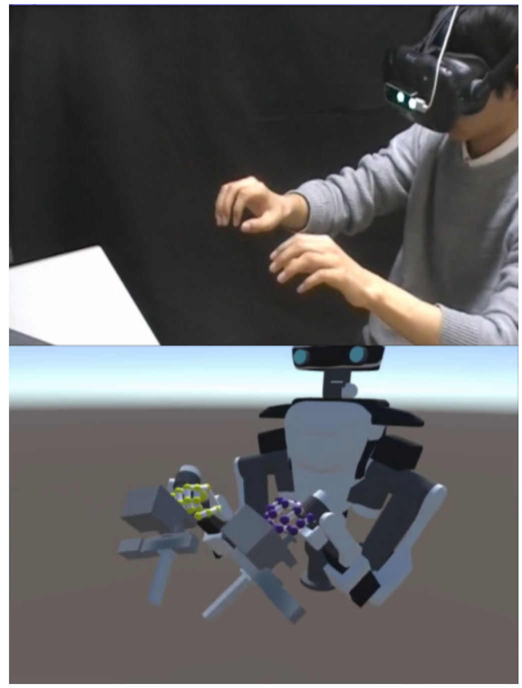

Fig. 2: On the top part of the figure the user is wearing an HMD and a LMC attached to the HMD. In the bottom part the virtual robot and virtual hands of the user (yellow and purple joints) are shown.

C. From human grasp to 2 fingers gripper virtual grasp

One of the advantages of using a virtual environment to capture the grasping motion is to avoid the occlusion problem of tracking hand-held objects and their grasping points using computer vision techniques such as the work of Oikonomidis et al. [8]. However, a few features are necessary to perform grasps in a virtual environment when using a virtual gripper. In this section, we explain the different features that we developed to perform the virtual grasp using the user's hand as input.

1) Virtual finger gripper control: To facilitate the control (open/close action and motion) of the virtual gripper by the user, we chose to model the control of the two-finger gripper as represented in Figure 3. The position of the gripper fingers is defined by the position of the thumb finger's tip (point B), the index finger's tip (point A) and the middle point (C) of the straight line connecting these points. The virtual gripper's fingers tip are located at points A′ and B′ (the crossing points between the hand's palm normal and its perpendicular line passing through point C), as shown in Figure 3. The orientation of the gripper is defined by the normal direction of the hand's palm. The virtual gripper adapted to the user's virtual hand is illustrated in Figure 4.

2) Virtual grasping: In order to realize the grasping in the virtual environment it was necessary to develop a fast and reliable method for defining a stable virtual grasp between

[a]https://www.leapmotion.com/
[b]https://unity.com/
[c]https://www.choreonoid.org/en/

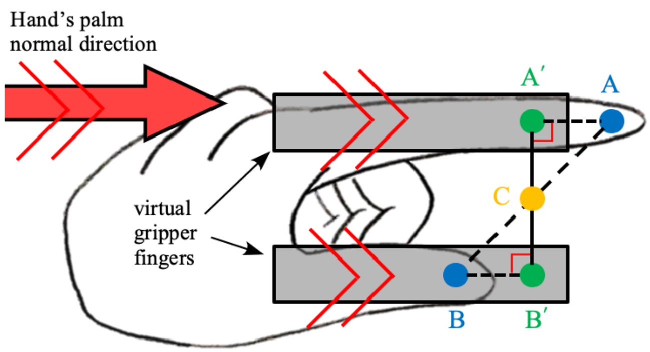

Fig. 3: The virtual gripper is controlled by the user's hand based on the fingers' tips and the orientation of the hand's palm normal.

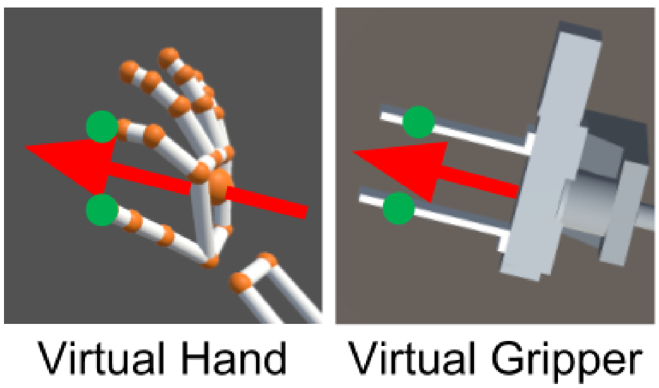

Fig. 4: Adaptation from the user's virtual hand to the robot's virtual gripper.

the virtual gripper and the virtual object which would also be stable with the real gripper and object parts.

To do so, we developed a simple algorithm based on the contact area between the gripper and the object. To determine the contact area we generate a point sampling of the CAD model of the finger's gripper and object part according to a Poisson-disk distribution using the algorithm of Corsini et al. [9], as shown in Figure 5. In this algorithm, a threshold is empirically determine for each pair of object-gripper. The grasp is then determined as stable if the contact point area between the gripper and the object is superior to this threshold, as illustrated in Figure 6.

D. Virtual assembly

To realize an assembly in the real world, the user obviously uses his sight but also its tactile sense. In the set-up described in this paper no haptic feedback has been used. In order to compensate the lack of haptic feed back, we developed a visual cue to help the user realize the virtual assembly has been completed. The method has been developed in our previous work [7]. The algorithm is based on the relative position between the object to be assembled and a certain threshold empirically determined to allow for some small degrees of freedom (DOF) to perform the assembly.

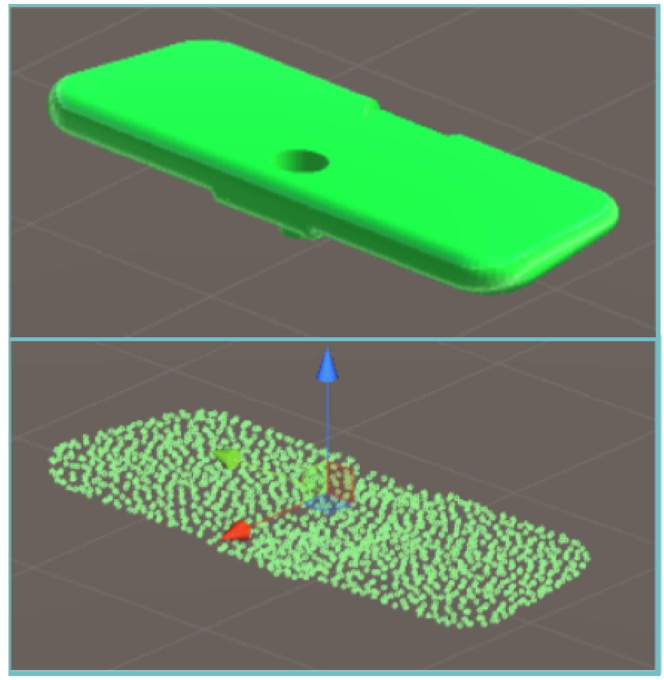

Fig. 5: The object's parts to be assembled are converted from a CAD model into a point based distribution in order to be used by the virtual grasping method that we developed. This virtual grasping method is used to determine if a virtual object is grasped or not by the user's virtual hand.

Once the algorithm successfully detects a virtual assembly, the objects that constitute the assembly change color, therefore helping the user performing the assembly.

E. Robot control

To reproduce the motion performed by the virtual robot onto the real robot, we use the controller developed by Harada et al. [10] which takes as input the joint's angles.

III. Experiment

In order to validate our approach, we performed an experiment where the user was asked to realize an assembly of a toy airplane in a virtual environment. During the demonstration the necessary data were recorded from the virtual robot to be later played back on the real robot, as seen in Figure 8.

A. Set-up

- To perform the demonstration, the user is wearing an HMD HTC Vive[d] which has a per eye resolution of 1080 x 1200 pixels.
- The hand (and fingers) tracking is realized with the the LMC (placed on the HMD as seen in Figure 2) and the orion framework[e] which offers a higher accuracy and frame rate.
- The object to be assembled is a three parts toy airplane as seen in Figure 1.

[d]https://www.vive.com/us/
[e]https://developer.leapmotion.com/orion

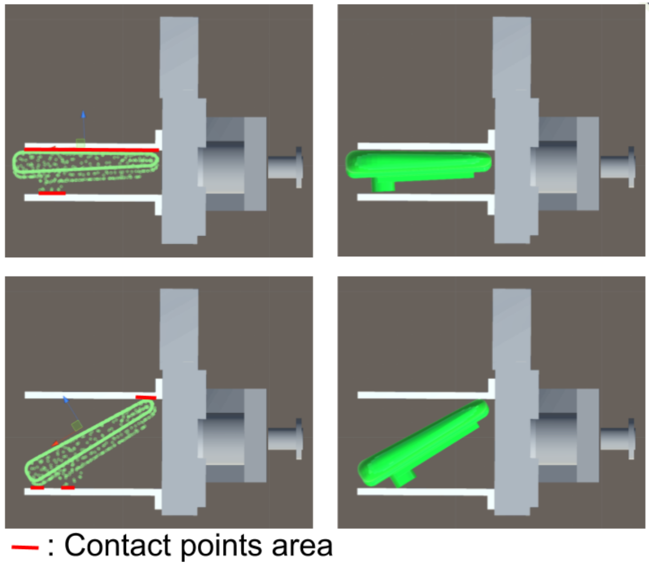

Fig. 6: In the upper part of the image, the number of contact points is superior to the defined threshold. Therefore the contact is judged stable and the virtual object is grasped by the virtual gripper.
In the lower part of the image, the number of contact points is below the defined threshold and therefore the grasping is not established. The user will need to move the gripper to increase the number of contact points to be able to realize the virtual grasp.

- The integration of the HMD and the LMC as well as the virtual scene and robot was made with the unity framework[f].
- The developed software ran on a desktop PC equipped with an i5 processor.
- During the demonstration the user was standing up in a room with plenty of space to move freely.

B. User's Demonstration

As mentioned previously the demonstration by the user is performed in a virtual environment which can be seen at each step of the assembly task in Figure 8. The numbers 1 to 8 correspond to each step of the assembly. Images with the same number have a matching time-stamp. For example the images 3A, 3B, 3C correspond to the same time-stamp.

The initial pose of the virtual robot and virtual toy airplane have been carefully set-up to match the initial pose of the real robot and the toy airplane, as seen in Figure 8-1B and Figure 8-1C. In this step, the fingers of the user are open in order to open the virtual gripper and grasp the upper part of the toy airplane in the next step.

In Figure 8-2A, 2B, 2C, we can observe the user placing the virtual gripper on top of the upper part of the toy airplane and close his fingers to grasp the part. The virtual grasp conditions are described in section II-C.2.

[f]https://unity.com/

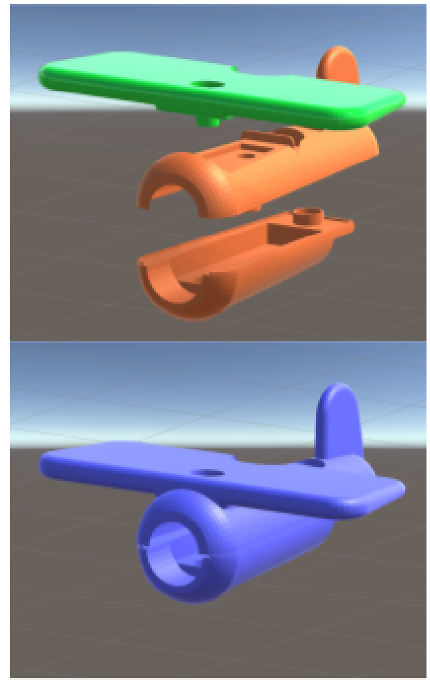

Fig. 7: A visual cue is used to inform the user that the virtual assembly was performed successfully. In this case the assembled object changed color and became blue.

In step 3, the upper part of the toy airplane is transported on top of the bottom part while the user's fingers are still holding the upper part.

In step 4, the upper part is released by the user to realize the assembly with the bottom part. It is worth noting that in the virtual environment a visual indication has been made to signal the user that the two parts have been virtually assembled and that it is safe for the user to release the grasped part. This visual cue is necessary to help the user realize a correct assembly since there is no haptic feedback.

From step 5 the left arm of the robot is used to realize the assembly. In this arm, a bigger gripper was attached to the hand in order to perform appropriately the assembly of the toy airplane's wing to its upper body. Following the same procedure as steps 1-4, in step 5 the user opens the gripper and places it in a suitable part of the wing to realize the grasp in the next step.

In step 6, the user closes the gripper to realize the grasp on the wing part of the toy airplane.

In step 7, the wing part is transported towards the upper body part of the toy airplane for its assembly.

In step 8, the visual cue indicates the user that the virtual assembly has been successful and that the grasped part can be released.

Along all of these steps, the joint's values of the virtual robot are recorded in order to be later played back on the real robot.

C. Robot's performance

After the user's demonstration and the recording of the joint's values of the virtual robot is finished, the experiment

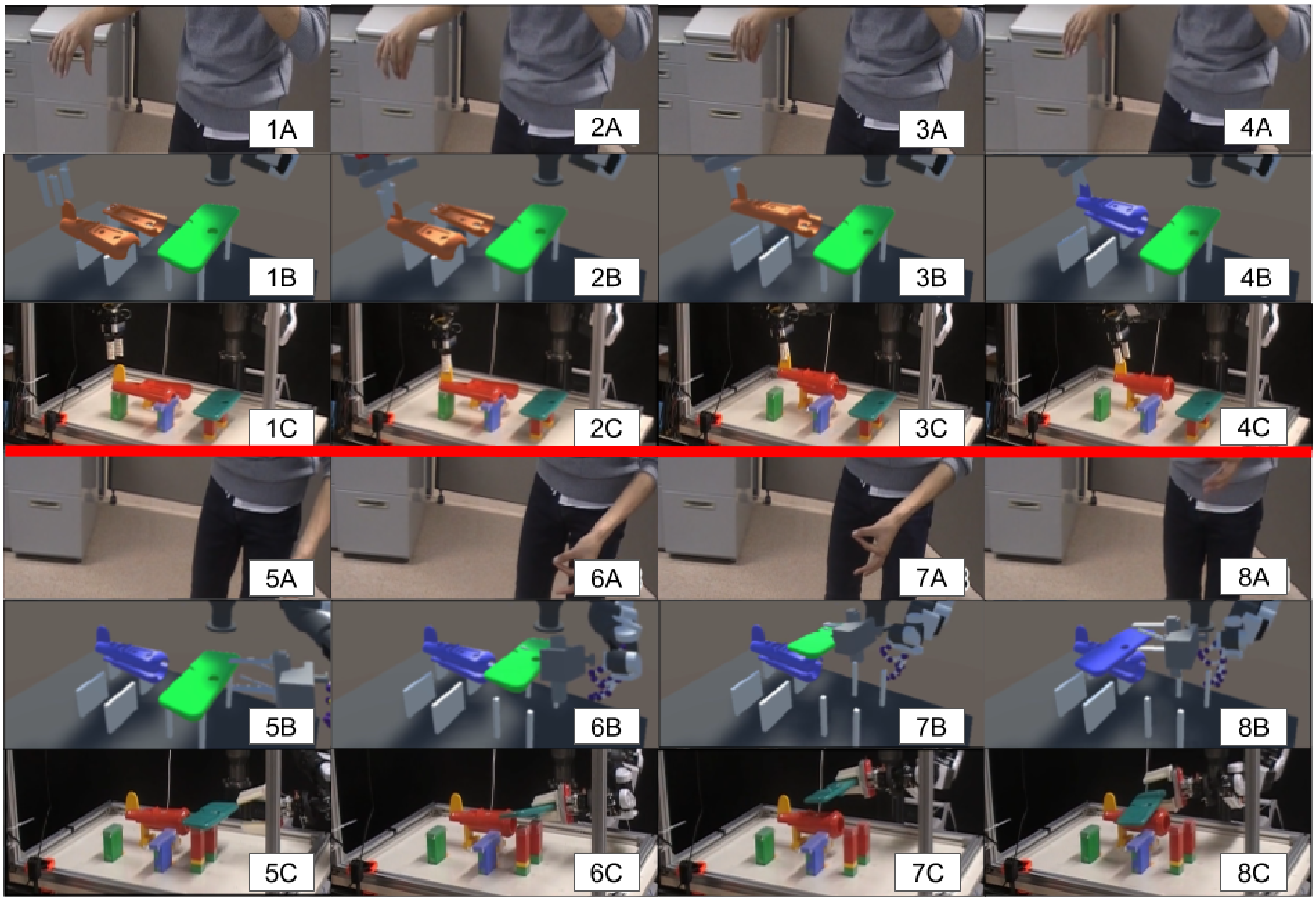

Fig. 8: The sequence of the experiment used to validate the method presented in this paper has 8 steps. Each step is illustrated with the user performing the demonstration (represented by the letter A); the virtual environment (represented by the letter B) controlled by the user during the demonstration; the robot performing the experiment demonstrated by the user (represented by the letter C).

is realized on the real robot. The recorded joint data are then feed in to the joint control developed in the graspPlugin[g] framework [10] to realize the experiment. Since the virtual robot model and its kinematics have been carefully modelled after the real robot, no adaptation of the recorded joints values is necessary. As a safety measure, the speed of execution of the recorded joints has been reduced.

IV. Discussion

The successful real assembly of a toy airplane by a dual-arm robot based on the demonstration of the user realized in a virtual environment validates the framework developed in this paper. The use of virtual reality makes it easy, safe and simple for the user to realize the assembly. The lack of haptic feedback is compensated with visual cues to communicate information to the user relative to the state of the assembly. The joints data are recorded and then directly played back on the real robot to realize the real assembly.

However, this is just the first stage of development, and a few points of this approach can be improved. One crucial point of this approach is having a virtual environment as close as possible to the real environment. Especially the starting pose of the object parts and the robot. In the current state, the virtual environment is set-up manually by measuring the real set-up manually. In the future, we would like to speed up this process by creating the virtual environment directly from the real environment using the object parts CAD model, AR markers [11], computer vision techniques and an optimized iterative closest point (ICP) algorithm such as the one proposed by Li and Song [12]. This will improve the reliability of the virtual environment and speed up the process of set-up of a new assembly.

The virtual grasp detection method developed in this paper works well for flat surfaces and finger grippers but could

[g]http://www.hlab.sys.es.osaka-u.ac.jp/grasp/en

be improved using the interaction forces simulated by the physics engine available in the unity framework. In this case, a more realistic virtual grasp detection could be implemented based on algorithms integrating interaction forces such as the one presented by Harada et al. [13]. Which could also be expanded for the use of a suction gripper.

Even though visual cues indicating the success of the virtual assembly to the user are very useful, further improvements are necessary to increase the realism of the virtual assembly. The use of haptic feedback in virtual reality has been shown to improve the realism of the virtual environment and object interaction [14]. In the future, we would like to develop a light haptic feedback placed on the fingertips of the user to increase the realism of the interaction. This could improve the speed and accuracy of the assembly made by the user in the virtual environment.

Finally, with the recent increase in use of machine learning methods comes the need of realistic data generation [15]. We believe that a virtual environment, such as the one presented in this paper, could help to generate the data used in machine learning algorithms that need a human feedback such as some reinforcement learning algorithms [16].

V. Conclusion

In this paper, we presented a method to realize an assembly through the virtual capture of a user demonstrating an assembly task. The data recorded during the demonstration of the task are then used to realize the assembly on a real robot, without the need of transforming the recorded data. We showed experimental results to demonstrate the validity of the proposed method. Future works have been discussed to improve the realism and performance of the assembly such as an automatic virtual environment generation and improvement of the virtual grasp by taking into account interaction forces generated by a physics engine.

Acknowledgment

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, 2016.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*, pp. 1371–1394, Springer, 2008.

[3] S. Calinon, *Robot programming by demonstration*. EPFL Press, 2009.

[4] T. Lau, S. A. Wolfman, P. Domingos, and D. S. Weld, "Programming by demonstration using version space algebra," *Machine Learning*, vol. 53, no. 1-2, pp. 111–156, 2003.

[5] J. Aleotti, S. Caselli, and M. Reggiani, "Toward programming of assembly tasks by demonstration in virtual environments," in *Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003. The 12th IEEE International Workshop on*, pp. 309–314, IEEE, 2003.

[6] J. Aleotti, S. Caselli, and M. Reggiani, "Leveraging on a virtual environment for robot programming by demonstration," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 153–161, 2004.

[7] D. Petit, I. G. Ramirez-Alpizar, Q. He, and K. Harada, "A virtual capture framework for assembly tasks," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 1618–1623, IEEE, 2018.

[8] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *2011 International Conference on Computer Vision*, pp. 2088–2095, IEEE, 2011.

[9] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 6, pp. 914–924, 2012.

[10] K. Harada, N. Yamanobe, W. Wan, K. Nagata, I. G. Ramirez-Alpizar, and T. Tsuji, "Motion-data driven grasp/assembly planner," *Journal of Robotics, Networking and Artificial Life*, vol. 5, no. 4, pp. 232–235, 2019.

[11] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*, vol. 51, 10 2015.

[12] W. Li and P. Song, "A modified icp algorithm based on dynamic adjustment factor for registration of point cloud and cad model," *Pattern Recognition Letters*, vol. 65, pp. 88–94, 2015.

[13] K. Harada, T. Tsuji, S. Uto, N. Yamanobe, K. Nagata, and K. Kitagaki, "Stability of soft-finger grasp under gravity," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 883–888, IEEE, 2014.

[14] H. G. Hoffman, "Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments," in *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No. 98CB36180)*, pp. 59–63, IEEE, 1998.

[15] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[16] D. Arumugam, J. K. Lee, S. Saskin, and M. L. Littman, "Deep reinforcement learning from policy-dependent human feedback," *arXiv preprint arXiv:1902.04257*, 2019.