2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Toward Following Changes in a Human's Posture: Stroking Motion Generation Using a Mobile Manipulator and an RGB-D Camera

Akishige Yuguchi, Sora Nii, Naoyuki Aikawa, and Yoshio Matsumoto

***Abstract*— While the conventional stroking motions using robot arms in physical human-robot interaction were planned from the pre-recognized shape of the target, it's not practical because the movements of the target body during stroking are not considered. In this paper, we propose a stroking motion generation method using a mobile manipulator and an RGB-D camera to follow changes in the target human's posture. Specifically, we first extract the target subject with image recognition and segmentation, and point cloud processing. Next, we generate the target part's 3D model and a motion trajectory on the model. Finally, we repeatedly update the trajectory by following changes in the target part's posture using an Iterative Closest Point (ICP) algorithm. For evaluation, the proposed method was implemented on a mobile manipulator HSR. Then, stroking motions were generated on a human-shaped robot's back with various movements, and the alignment success score and the alignment error were measured. From the results, we confirmed that the motion generation was highly successful and occlusion by the manipulator's arm affected the alignment.**

I. Introduction

Stroking the human body is a fundamental motion in physical care, such as washing the body, massage, and skincare. Several studies have reported that touching and stroking human subjects in physical care, *e.g.*, massage can produce positive effects, such as a reduction in pain [1] and improvements in sleep quality [2].

On the other hand, robots have recently been expected to perform stroking motions due to labor shortages in physical care. In previous studies on robotic stroking motions, Ishikura *et al.* [3], [4] reported that the stroking motion on a linear trajectory on the human back using a robot arm provided pleasant feelings in younger and older participants, similar to those provided by a human. Yuguchi *et al.* [5] suggested that generating stroking motions using a robot arm that follows arbitrary target positions on the shape of the human back observed by an RGB-D camera had the potential to provide a more pleasant feeling than stroking motions on a linear trajectory using the robot arm. As a similar technique, Tsumaki *et al.* [6] realized stroking motions using a robot arm to apply ointment to the human back by estimating arbitrary target positions on the curved shape of the back using stereo vision.

In addition, as a closer way to stroking motions using robot arms on the human body, King *et al.* [7] achieved autonomously generating wiping movements using robot arms to remove dirt from the skin by estimating the distance to the human body using a laser range finder. Hawkins *et al.* [8] created an ellipsoidal 3D model of the care-receiver's head and achieved the shaving task using a mobile manipulator by having the end-effector follow arbitrary target positions on the model. Bolotnikova *et al.* [9] achieved autonomously generating a motion using a humanoid robot, *Pepper*, such as placing the robot's hand on the care-receiver's arm, simulating the start of the care-receiver's transfer support by planning the target contact point from a surface model of the human body's point clouds. Although these systems could generate motions following arbitrary positions on the target human body part, modeling the body part and planning the motion trajectory was performed only once before the motion generation. Hence, they are unable to generate motions in response to changes in the posture of the care-receiver's target part, such as breathing and scratching oneself, and there are still issues to be addressed before they can be performed in real-world environments. Therefore, it is essential for practical applications to generate stroking motions that adjust the pose of the care-receiver's movements.

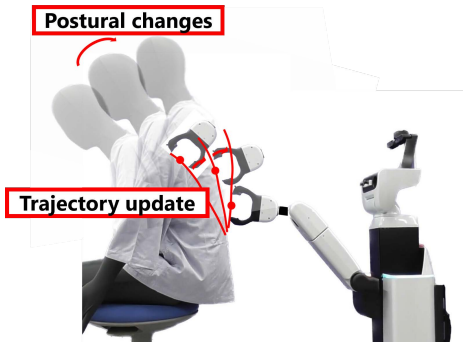

Fig. 1. This work's target: stroking motion following changes in the target posture using a mobile manipulator

In this study, we propose a stroking motion generation method using a mobile manipulator and an RGB-D camera to follow changes in the target human's pose, as shown in Fig. 1. Specifically, we first extract the target subject's part with image recognition and segmentation, and point cloud processing from the mobile manipulator's RGB-D camera. Next, we generate the target part's 3D model and a motion trajectory on the model. Finally, we repeatedly update the trajectory by following changes in the target part's pose using an ICP algorithm. For experimental evaluation, we implement the proposed method using a mobile manipulator HSR and an RGB-D camera attached to the HSR's head, generate stroking motions on a human-shaped robot's back

Akishige Yuguchi, Sora Nii, Naoyuki Aikawa, and Yoshio Matsumoto are with the Graduate School of Advanced Engineering, Tokyo University of Science, Tokyo 125-8585, Japan {akishige.yuguchi, ain, yoshio.matsumoto}@rs.tus.ac.jp

978-1-6654-5784-2/26/$31.00 ©2026 IEEE

with various movements, and measure the alignment success score and the errors as quantitative indices.

The rest of this paper is organized as follows: Section II describes the proposed method. Section III describes the implementation and experiment of the stroking motion generation using a mobile manipulator for the human-like robot's back with various movements. Section IV describes the quantitative evaluation on the alignment success and error, and their result and discussion. Finally, Section V concludes this paper.

II. Proposed Method

A. Overview

The stroking motion in this study refers to a stroking motion in a direction from the upper human back to the lower human back as well as the aforementioned previous studies [3], [4], [5]. The proposed method consists of two processes: generating the initial stroking motion trajectory and repeatedly updating the stroking motion trajectory.

In the first process, we first extract the point cloud data of the target human body part from the mobile manipulator's RGB-D camera and perform appropriate downsampling of the extracted point cloud using voxel downsampling (hereafter referred to as the initial point clouds P) to reduce the amount of calculation. Next, we generate a 3D mesh model of the target part from the initial point clouds P, and a stroking motion trajectory (hereafter referred to as the initial stroking motion trajectory) on the model's surface.

In the second process, we first extract the current observed point cloud data (hereafter referred to as the current point clouds Q) of the target part from the RGB-D camera using the same extraction method in the above process. Next, we estimate the pose change of the target part from the poses between the initial point clouds P and the current point clouds Q. Third, we update the initial stroking motion trajectory based on the pose change estimation. Finally, we repeat this process at every frame of the RGB-D camera.

In this section, we explain the three elemental methods in these processes, *i.e.*, the extraction of the target human body part's point clouds, the generation of the initial stroking motion trajectory, and the repeated trajectory update.

B. Extraction of the Target Human Body Part's Point Clouds

We use BodyPix[1] to recognize the target part of the human body. BodyPix is a real-time human body part segmentation that can recognize the human body in an image by dividing it into 24 parts, such as the back, the upper and lower arms and legs, and the face. As shown in Fig. 2, we first obtain the target part's mask image using the segmentation on the RGB image captured by the RGB-D camera. Next, we segment the target part's RGB-D image using this mask image. Third, we generate the target part's point clouds from the RGB-D image with the downsampling. Finally, we obtain the point clouds Q by performing the downsampling in each frame of the RGB-D camera.

[1]BodyPix, https://github.com/tensorflow/tfjs-models/tree/body-pix-v2.0.4/body-pix

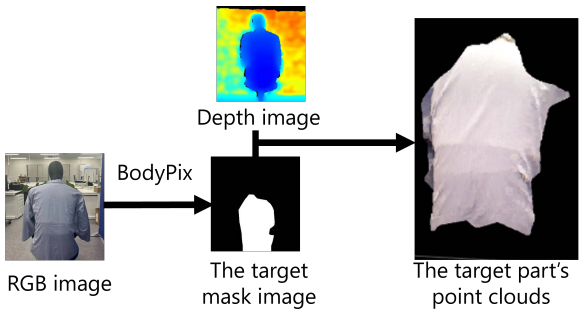

Fig. 2. Extraction of the target human body part's point clouds

C. Generation of the Initial Stroking Motion Trajectory

To generate a 3D mesh model and the initial stroking motion trajectory on the model, we estimate the normal vectors of all points in the initial point clouds P, as follows.

1) Search the neighboring points of each point $\mathbf{p}$ in the initial point clouds P.
2) Perform principal component analysis on the neighboring points' positions and eigendecomposition by calculating the variance-covariance of the neighboring points.
3) Select the eigenvector with the smallest eigenvalue as the normal vector.
4) Unify the direction of the normal vectors so that the z component of all normal vectors becomes a negative value.

We first generate a 3D mesh model with the calculated normal vectors using the Ball-Pivoting algorithm [10]. Next, we convert the mesh model into a graph where each node has the vertex position and the normal vector, and each edge has the distance between the connected vertices. Finally, we search for a trajectory that connects an arbitrary starting node (S) and an arbitrary ending node (G) on the graph using the Dijkstra algorithm [11] and generate this trajectory as the initial stroking motion trajectory. Note that the initial stroking motion trajectory is generated only once before the motion generation.

D. Repeated Trajectory Update

To update the stroking motion trajectory repeatedly, we estimate the pose change of the target part at every frame of the RGB-D camera. We adopt the Iterative Closest Point (ICP) algorithm [12] for the pose change estimation. In this case, we align the initial point clouds P with the current point clouds Q and estimate the transformation matrix $\boldsymbol{T}$ of the pose change between the point clouds P and Q. Specifically, we perform the ICP algorithm in the following steps.

1) From each point $\mathbf{p}$ in the initial point clouds P, find the nearest point $\mathbf{q}$ within the distance threshold in the current point clouds Q, and create correspondence pairs, $\mathcal{K} = \{(\mathbf{p}, \mathbf{q})\}$. Hereafter, the point $\mathbf{q}$ corresponding to the point $\mathbf{p}$ is referred to as the corresponding point.
2) Update the transformation matrix $\boldsymbol{T}$ to minimize the objective function $E(\boldsymbol{T})$ in the following equation:

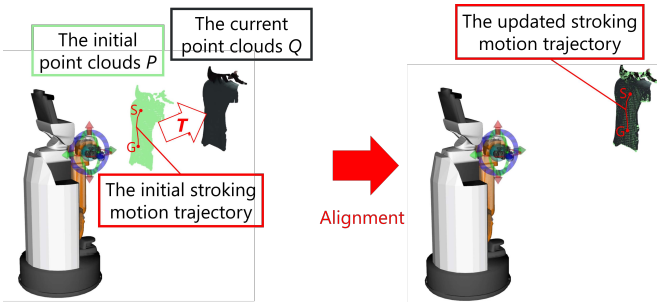

Fig. 3. Alignment from the initial point clouds P to the current point clouds Q and the stroking motion trajectory update using an ICP algorithm in the proposed method

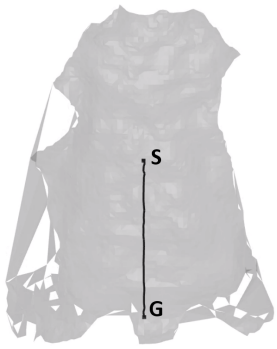

Fig. 4. The initial stroking motion trajectory on an example of the 3D mesh model of the human back

$$E(\boldsymbol{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \|\mathbf{q} - \boldsymbol{T}\mathbf{p}\|^2. \tag{1}$$

Steps 1) and 2) are repeated until the transformation matrix $\boldsymbol{T}$ converges. Finally, we update the stroking motion trajectory from the initial stroking motion trajectory with the estimated transformation matrix $\boldsymbol{T}$. Fig. 3 shows this trajectory update's overview.

III. Motion Generation Experiment

A. Implementation

We implemented the proposed method on a mobile manipulator Human Support Robot (HSR) [13], produced by Toyota Motor Corporation, to generate stroking motions. This HSR has 5 DOFs in the arm, 3 DOFs in the base, and 2 DOFs in the head (pan and tilt). To reach an arbitrary target pose by the HSR's arm, the HSR has to move both the arm and the base. We attached an RGB-D camera, Intel RealSense D435, to the HSR's head. In addition, the HSR's hand was made to grasp a Foaming Net Ball produced by Ryohin Keikaku Co., Ltd., as the end effector for stroking motions. The software system of the HSR was built on Robot Operating System (ROS)[2]. We used MoveIt [14] for stroking motion generation and Open3D [15] for point cloud processing. We set the initial stroking motion trajectory to one direction from the upper back to the lower back in this experiment, as well as our previous works [3], [4], [5], as shown in Fig. 4.

[2]Robot Operating System (ROS), https://wiki.ros.org/

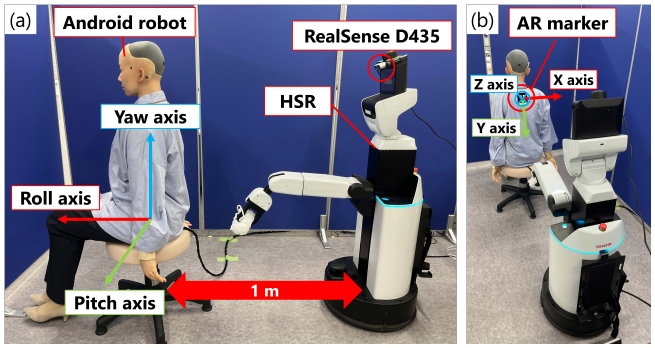

Fig. 5. Experimental environment

TABLE I
The patterns in the average angular speed of the android robot's waist movements [deg/s]

	Lower speed	Middle speed	Higher speed
Pitch axis	0.50	0.65	0.94
Yaw axis	0.55	0.74	1.06

To prevent any unexpected motion generation, we implement the safety mode on the HSR. Specifically, we calculate the value $fitness$, which represents the ratio of the number of corresponding points $N_{\mathcal{K}}$ to the number of points in the target point clouds N_Q, as follows.

$$fitness = \frac{N_{\mathcal{K}}}{N_Q}. \tag{2}$$

The corresponding points $\mathcal{K}$ represent a pair of a source point and a target point, and points within the specified distance threshold are considered to be corresponding points. When the value $fitness$ calculated by the point cloud alignment is under 0.4 and the distance threshold for corresponding points for the alignment is over 0.025 [cm], it is judged that the alignment has failed, and then the HSR's operation automatically stops. Note that the HSR initially has a safety mode that automatically stops its operation when an abnormality occurs due to excessive pressure detected by the installed force sensors.

B. Experimental setup

For the stroking target, we used an android robot (AL-G109ST-F) with an adult female human appearance as a human-shaped robot produced by A-Lab Co., Ltd and set its back as the target part (See Fig. 5 (a)). This robot is driven by pneumatic actuators with an 8-bit input controller and can control the pitch and yaw axes around the waist to express movements such as bowing and twisting the body. The android robot has an advantage to generate the same motion every time and be lower risk to be damaged by stroking motion generation failures, in contrast to human subjects.

For actual motion generation, we prepare three speed patterns (lower, middle, and higher) in the pitch axis, the yaw

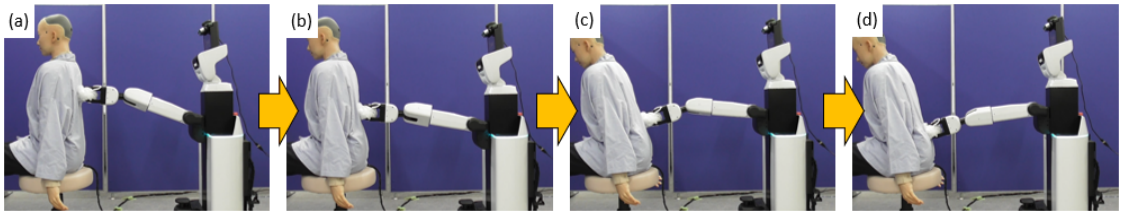

Fig. 6. A successful example of the stroking motion generation using the HSR when the android robot's waist moves on the pitch axis at the lower speed

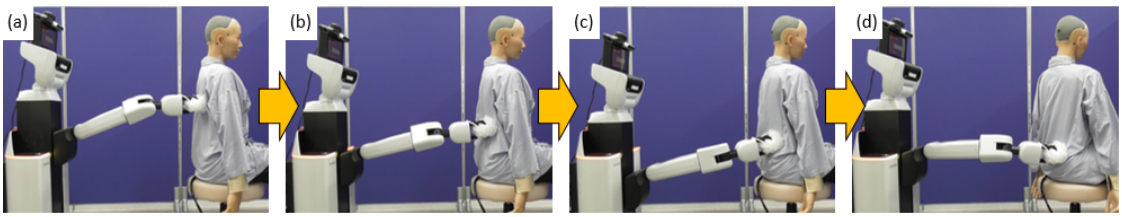

Fig. 7. A successful example of the stroking motion generation using the HSR when the android robot's waist moves on the yaw axis at the lower speed

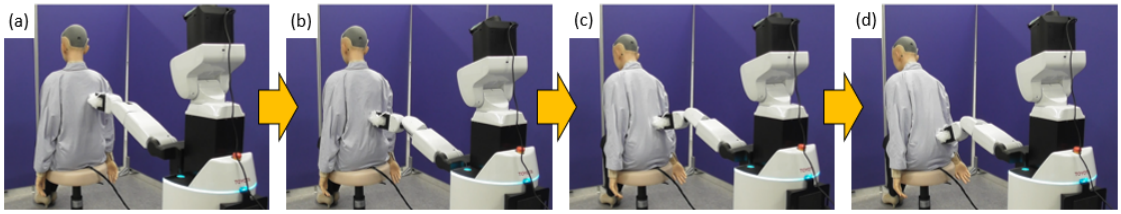

Fig. 8. A successful example of the stroking motion generation using the HSR when the android robot's waist moves on the pitch and yaw axes at the lower speed

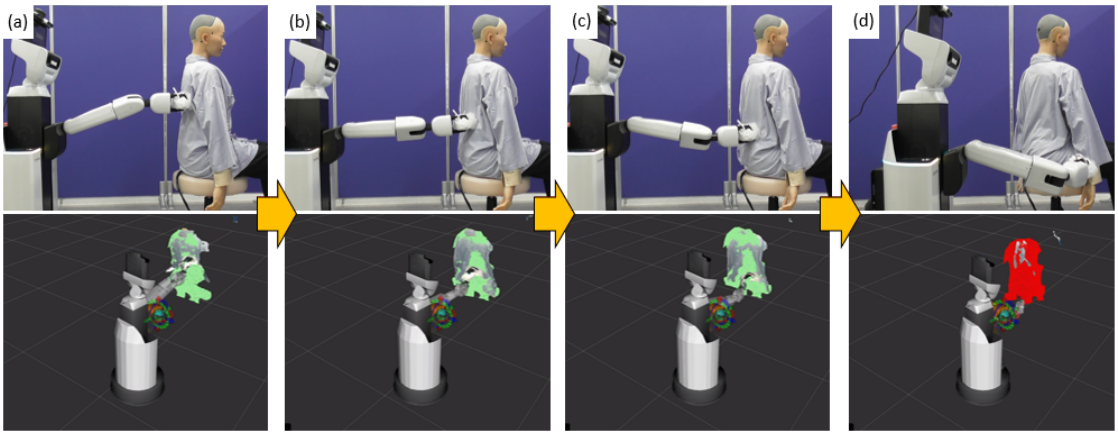

Fig. 9. A failure example of the stroking motion generation using the HSR when the android robot's waist moves on the yaw axis at the lower speed due to an alignment failure. The bottom figures visualize the alignment results. The alignment success is painted in green. The alignment failure is painted in red.

axis, and both axes of the android robot's waist, as shown in Table I. The total movements are 9 patterns. The initial position of the HSR was set at about 1 m behind the android robot so that the entire back of the android robot, which was dressed in a patient's wear[3] used in the previous work [5] could be captured by the RealSense D435 on the HSR's head.

[3]SG1441, Nagai Leben, Inc.

C. Result

We conducted the motion generation five times in each pattern. When the android robot's waist moved on the pitch axis and the pitch and yaw axes, the success rate of the stroking motion generation was over 80%. On the other hand, when the android robot's waist moved on only the yaw axis, the success rate of the stroking motion generation was less than 20%. The failure counted the case that the motion generation didn't finish. Thus, we confirmed that the proposed method could follow these pose changes and highly

succeeded in the stroking motion generation on the back.

Fig. 6, 7, and 8 show a successful example of the stroking motion generation using the HSR when the android robot's waist moves at the lower speed on the pitch and yaw axes. From Fig. 6, 7, and 8, the end-effector moves to follow the change in the back's pose, and the stroking motion was successfully generated by following the back.

Fig. 9 shows a failure example of the stroking motion generation using the HSR when the android robot's waist moves on the yaw axis at the lower speed due to an alignment failure. The upper figures show the end-effector moves away from the android robot's back and the HSR stops at the position away from the back to the right. The bottom figures in Fig. 9 visualize the alignment of the initial point clouds P with the current point clouds Q. The point clouds painted in green show the initial point clouds P after alignment. The point cloud painted in red indicates an alignment failure, as shown in the bottom right figure.

The failures were mainly caused by an alignment failure around the end of the stroking motion generation. The main factor in the alignment failure was thought to be the significant change in the shape of the back's surface between the start and end of the stroking motion. Since the android robot's waist movement on the pitch axis is like bowing, the shape of the back's change seems to be relatively little. On the other hand, since the android robot's waist movement on the yaw axis involves twisting the android robot's torso, the shape of the back's change seems to be significant. For this reason, we thought that the alignment failure at the end of the stroking motion generation was caused by significantly changing the shape of the back from the start and increasing the error in the corresponding points.

IV. Quantitative Evaluation

A. Quantitative Indices

We also conducted a quantitative evaluation of the proposed method under some conditions. In this experiment, we measured the alignment success score and the alignment error as quantitative indices.

To obtain the alignment success score, we adopted the $fitness$ (See Eq. (2)). When the $fitness$ is higher, it means that the alignment success is higher, and then the trajectory update becomes more successful.

To obtain the alignment error, we calculate an orientation error between the point clouds Q and the AR marker attached to the stroking target. The AR marker's orientation is the true orientation of the stroking target. When the alignment error is larger, it means that the pose error between the stroking target and the stroking motion trajectory is larger, and then the pose error between the actual and ideal trajectories is larger.

B. Experimental setup

The experiment was conducted in the same environment in Section III. Note that the android robot's waist moved at 0.45 [deg/s] in the pitch axis or at 0.50 [deg/s] in the yaw axis, close to the lower speed. In addition, we attached an AR marker (0.07 [m] $\times 0.07$ [m]) to the android robot's back as shown in Fig. 5 (b) to calculate the alignment error.

TABLE II

Experimental conditions in the quantitative evaluation

	Stroking motion generation	The android robot's waist's orientation change
C1	-	-
C2	-	Pitch axis
C3	-	Yaw axis
C4	✓	-
C5	✓	Pitch axis
C6	✓	Yaw axis

TABLE III

Measurement results of the alignment success score

	Mean (SD) value	Minimum value
C1	0.999 (0.002)	0.989
C2	0.998 (0.002)	0.990
C3	0.957 (0.043)	0.864
C4	0.560 (0.059)	0.485
C5	0.533 (0.053)	0.440
C6	0.582 (0.440)	0.480

Table II shows all the conditions for this experiment. C1 was set as the standard condition that eliminated the two movement factors, *i.e.*, the target part's pose changes and the stroking motions. Specifically, in C1, the proposed method without the stroking motion generation was performed with no change in the target part's pose. In C2 and C3, the android robot was made to change its pose around the pitch and yaw axes, respectively, but the stroking motion was not generated. In C4, the android robot's pose was fixed and the stroking motion was generated. In C5 and C6, the stroking motion was generated and the android robot's pose was changed around the pitch and yaw axes, respectively. In each condition, we once measured the value $fitness$ by every frame of the RGB-D camera after 35 [s], which we set to have the android robot start moving.

C. Results of Alignment Success Score

Table III shows the mean (SD) and minimum values of the measured $fitness$ in all conditions. Compared to C1, where the android robot's waist did not move, and C2, where the waist moved in the pitch axis, in C3, where the waist moved in the yaw axis, the mean of $fitness$ decreased slightly.

In addition, in C4-C6, where the stroking motions were generated, the mean of $fitness$ decreased relatively. Hence, compared to C1, we observed that the alignment success score decreased slightly in C3 and decreased in C4-C6. From these results, we thought that the occlusion by the manipulator's arm mainly caused the alignment success score decrease.

D. Result of Alignment Error

Table IV shows the measurement results of the alignment error. The AR marker's coordinate system is shown in Fig 5 (b). The results show the mean (SD) and maximum of the alignment error in each axis in each condition.

TABLE IV
MEASUREMENT RESULTS OF THE ALIGNMENT ERROR [DEG]

	X axis		Y axis		Z axis	
	Mean (SD) value	Maximum value	Mean (SD) value	Maximum value	Mean (SD) value	Maximum value
C1	0.68 (0.33)	1.40	0.12 (0.39)	0.91	0.07 (0.09)	0.30
C2	0.52 (1.70)	3.52	4.06 (2.63)	8.01	1.57 (0.94)	3.55
C3	0.29 (2.56)	5.75	0.72 (3.36)	6.78	0.30 (0.79)	1.92
C4	2.98 (2.51)	7.59	0.80 (2.06)	5.30	1.19 (1.29)	0.89
C5	1.41 (1.16)	4.34	0.69 (1.65)	4.55	0.33 (1.01)	1.43
C6	4.15 (3.71)	11.42	1.15 (1.60)	4.29	3.62 (3.72)	2.74

Compared to C1, where the android robot's waist did not move, and C3, where the waist moved in the yaw axis, in C2, where the waist moved in the pitch axis, the alignment error was larger. This result also means that the stroking target (*i.e.*, the android robot's back)'s orientation change in C2 was larger than in C3.

In addition, in C4-C6, where the stroking motions were generated, compared to C1, the mean of the alignment error was larger. Especially, the alignment error, including the maximum error, was larger in C6. From these results, we thought that the occlusion by the manipulator's arm mainly caused the alignment error increase, as mentioned in Section IV-C.

Through all experiments, we confirmed that the alignment success score and error were larger due to occlusion by the manipulator's arm. We will need to consider where to attach an RGB-D camera to a mobile manipulator.

E. Experimental Limitations

The number of the quantitative evaluations was limited, so that, for example, we could not discuss the effect of the proposed method in each condition on wrinkles in the patient's wear on the entire back of the android robot. Furthermore, the type of the stoking motion trajectory was limited to one direction from the upper back to the lower back, so that we could not discuss evaluation results for other types of motion trajectories.

V. CONCLUSION

In this paper, we proposed a stroking motion generation method using a mobile manipulator and an RGB-D camera to follow changes in the target human body part's pose. For evaluation, we implemented the proposed method on a mobile manipulator HSR with an RGB-D camera RealSense D435, generated stroking motions on the human-shaped robot's back with various movements, and measured the alignment success score and the alignment error. From the results, we confirmed that the motion generation was highly successful and occlusion by the manipulator's arm affected the alignment in the proposed method.

As future work, we would like to adopt the proposed method to generate stroking motions with various trajectories on a human subject and/or other target body parts. Furthermore, we will consider where the camera should be placed on the manipulator to minimize occlusion by the manipulator's arm.

ACKNOWLEDGMENT

We appreciate the technical support from the Human Support Robot (HSR) User Community, Japan.

REFERENCES

[1] P. Goldstein, S. G. Shamay-Tsoory, S. Yellinek, and I. Weissman-Fogel, "Empathy predicts an experimental pain reduction during touch," *The Journal of Pain*, vol. 17, no. 10, pp. 1049–1057, 2016.

[2] K. Andersson, L. Törnkvist, and P. Wändell, "Tactile massage within the primary health care setting," *Complementary therapies in clinical practice*, vol. 15, no. 3, pp. 158–160, 2009.

[3] T. Ishikura, Y. Kitamura, W. Sato, J. Takamatsu, A. Yuguchi, S.-G. Cho, M. Ding, S. Yoshikawa, and T. Ogasawara, "Pleasant stroke touch on human back by a human and a robot," *Sensors*, vol. 23, no. 3-1136, pp. 1–11, 2023.

[4] T. Ishikura, W. Sato, J. Takamatsu, A. Yuguchi, S.-G. Cho, M. Ding, S. Yoshikawa, and T. Ogasawara, "Delivery of pleasant stroke touch via robot in older adults," *Frontiers in Psychology*, vol. 14, no. 1292178, pp. 1–8, 2024.

[5] A. Yuguchi, T. Ishikura, S.-G. Cho, J. Takamatsu, and T. Ogasawara, "Robotic stroke motion following the shape of the human back: Motion generation and psychological effects," in *Proceedings of the ICRA 2024 Workshop on Nursing Robotics*, no. 33, 2024, pp. 1–5.

[6] Y. Tsumaki, T. Kon, A. Suginuma, K. Imada, A. Sekiguchi, D. N. Nenchev, H. Nakano, and K. Hanada, "Development of a skincare robot," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 2963–2968.

[7] C.-H. King, T. L. Chen, A. Jain, and C. C. Kemp, "Towards an assistive robot that autonomously performs bed baths for patient hygiene," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 319–324.

[8] K. P. Hawkins, P. M. Grice, T. L. Chen, C.-H. King, and C. C. Kemp, "Assistive mobile manipulation for self-care tasks around the head," in *Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Robotic Rehabilitation and Assistive Technologies (CIR2AT)*, 2014, pp. 16–25.

[9] A. Bolotnikova, S. Courtois, and A. Kheddar, "Multi-contact planning on humans for physical assistance by humanoid," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 135–142, 2020.

[10] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[11] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[12] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[13] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of human support robot as the research platform of a domestic mobile manipulator," *ROBOMECH Journal*, vol. 6, no. 1, pp. 1–15, 2019.

[14] D. Coleman, I. Şucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *Journal of Software Engineering for Robotics*, vol. 5, no. 1, pp. 3–16, 2014.

[15] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv preprint arXiv:1801.09847*, 2018.