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

Extended Diffeomorphism for Real-Time Motion Replication in Workspaces with Different Spatial Arrangements

Masaki Saito[1,2], Shunki Itadera[1] and Toshiyuki Murakami[2]

***Abstract*— This paper presents two types of extended diffeomorphism designs to compensate for spatial placement differences between robot workspaces. Teleoperation of multiple robots is attracting attention to expand the utilization of the robot embodiment. Real-time reproduction of robot motion would facilitate the efficient execution of similar tasks by multiple robots. A challenge in the motion reproduction is compensating for the spatial arrangement errors of target key-points in robot workspaces. This paper proposes a methodology for smooth mappings that transform primary robot poses into follower robot poses based on the predefined key points in each workspace. Through a picking task experiment using a dual-arm UR5 robot, this study demonstrates that the proposed mapping generation method can balance lower mapping errors for precise operation and lower mapping gradients for smooth replicated movement.**

I. INTRODUCTION

Robotic teleoperation plays an important role in hazardous environments and in areas that are inaccessible to humans. A fundamental functionality of a teleoperation system is to map the operator's input to the robot's movement. When a single operator controls multiple robots to perform a collaborative task, such as lifting a large object, the operator needs to repeat similar input to control each robot in the non-common action phase, e.g., guiding individual robot to the corresponding grasping point. Once the geometric errors among the robots become acceptable, the operator can send a common command to control all the robots simultaneously, e.g., lifting the table up.

Although multi-robot teleoperation is a promising method to expand the usage of robot embodiment, large workloads are required for precise control in the non-common action phase. Therefore, a reduction method of operational workload is important for practical multi-robot teleoperation. One approach is to develop an interface to easily select a target robot in the non-common action phase and a shared control surface in the common action phase [1]. While this approach reduces the operation workload for selecting robots, the operator still needs to control robots individually, which would lead to a decrease in work efficiency. In this study, we address a methodology to control multiple robots simultaneously to improve the efficiency in the non-common action phase. This paper tackles on real-time replication of a primary robot action for the other robots, considering the geometric difference among the robots' workspaces. This approach aims to allow the operator to focus on and control a primary robot, and the other robots automatically adjust the geometric errors and reach the target poses without time delay.

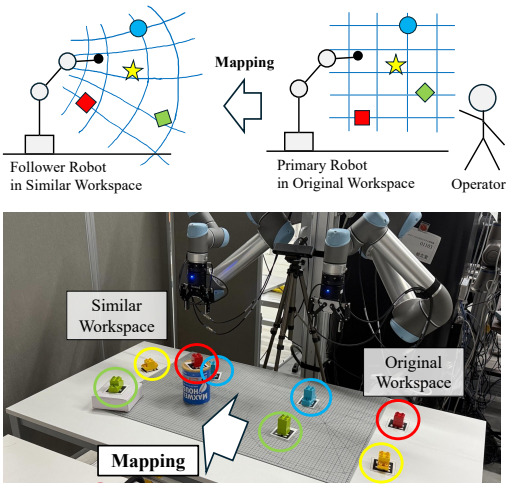


Fig. 1. Concept of our proposed diffeomorphism mapping.

There have been several studies on real-time motion replication for simultaneous multi-robot operation. For instance, variable scaling methods [3], [4] are used to adjust motion scaling factors for each robot to absorb geometric error in the corresponding environment. However, such scaling methods require updating the scaling coefficients when the geometric configuration of the workspace changes, leading to a discontinuous and non-transparent experience. To overcome the shortcomings of the scaling methods, a diffeomorphism-based spatial mapping method has been proposed [5]. The map is represented by a weighted composition function of radial basis functions (RBF), which is continuous and differentiable, and designed for minimizing the inter-environmental errors of position and orientation between the robots. However, in highly unstructured environments, the author reported that significant distortion occurs in the mapping due to limited compensation for large inter-environmental errors. Since excessive distortion in positional mapping causes a large robot velocity, the gradient of mapping needs to be low in practice. Nevertheless, the conventional methods are hard to achieve both a lower mapping gradient and a lower mapping error in some cases.

This paper aims to define a novel diffeomorphism to mini-

This work was supported in part by JSPS KAKENHI JP24K20826 and JST CRONOS JPMJCS24K6.

[1]National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan (e-mail: {saitou-aist312, s.itadera}@aist.go.jp)

[2]Department of System Design Engineering, Keio University, Yokohama, Japan (e-mail: mura@sd.keio.ac.jp)

mize both mapping gradient and mapping error, even in cases of highly geometric randomized workspaces. We propose two types of extended diffeomorphisms: Rotation Extended Diffeomorphism and Twisted Affine Extended Diffeomorphism. Rotation Extended Diffeomorphism is an extension of vector-based positional diffeomorphism to quaternion-based one. Twisted Affine Extended Diffeomorphism is an extension of the conventional method with the addition of a nonlinear mapping for workspaces with various deformations. Fig. 1 shows a concept image of our proposed mapping methodology. We demonstrate the effectiveness of the presented methodology through simulation and experiments.

II. Diffeomorphism for Motion Replication

We define a mapping formulation for real-time motion replication. The transformation from the primary robot position $\boldsymbol{p}$ and orientation $\boldsymbol{q}$ to another robot's position $\hat{\boldsymbol{p}}$ and orientation $\hat{\boldsymbol{q}}$ is represented as follows:

$$\hat{\boldsymbol{p}} = \hat{\boldsymbol{\Phi}}(\boldsymbol{p}),\ \hat{\boldsymbol{q}} = \hat{\boldsymbol{\Psi}}(\boldsymbol{p}, \boldsymbol{q}), \tag{1}$$

where $\hat{\boldsymbol{\Phi}}$ and $\hat{\boldsymbol{\Psi}}$ are diffeomorphic mapping functions for position and orientation, respectively. This paper address a novel design of the position mapping function $\hat{\boldsymbol{\Phi}}$.

A. Conventional Diffeomorphism

According to the prior work [5], we here define conventional diffeomorphic mapping functions $\hat{\boldsymbol{\Phi}}$ and $\hat{\boldsymbol{\Psi}}$ in eq. (1) as $\hat{\boldsymbol{\Phi}}_{\mathrm{DIFF}}$ and $\hat{\boldsymbol{\Psi}}_{\mathrm{DIFF}}$, which are written as:

$$\hat{\boldsymbol{\Phi}}_{\mathrm{DIFF}}(\boldsymbol{p}) = \boldsymbol{\Phi}_J \circ \boldsymbol{\Phi}_{J-1} \circ \cdots \circ \boldsymbol{\Phi}_2 \circ \boldsymbol{\Phi}_1(\boldsymbol{p}), \tag{2}$$

$$\hat{\boldsymbol{\Psi}}_{\mathrm{DIFF}}(\boldsymbol{p}, \boldsymbol{q}) = \boldsymbol{\Psi}_J \circ \boldsymbol{\Psi}_{J-1} \circ \cdots \circ \boldsymbol{\Psi}_2 \circ \boldsymbol{\Psi}_1(\boldsymbol{p}, \boldsymbol{q}), \tag{3}$$

where $\boldsymbol{\Phi}_j$ and $\boldsymbol{\Psi}_j$ $(j = 1 \cdots J)$ are j-th basic composition function for position and orientation mapping, and J is the total number of the composition. Note that the forms of the basic composition functions are unique, and we omit the j index for a simple expression. The basic composition functions ($\boldsymbol{\Phi}(\boldsymbol{p})$ and $\boldsymbol{\Psi}(\boldsymbol{p}, \boldsymbol{q})$) are defined as:

$$\boldsymbol{\Phi}(\boldsymbol{p}) = \boldsymbol{p} + k_f(\boldsymbol{p}|\rho_1, \boldsymbol{c}_1) \cdot \boldsymbol{v}_1, \tag{4}$$

$$\boldsymbol{\Psi}(\boldsymbol{p}, \boldsymbol{q}) = \boldsymbol{v}_2{}^{k_f(\boldsymbol{p}|\rho_2, \boldsymbol{c}_2)} \otimes \boldsymbol{q}, \tag{5}$$

where $\boldsymbol{v}_1 \in \mathbb{R}^3$ and $\boldsymbol{v}_2 \in \mathbb{H}$ are corresponding to the amount of linear transformation of key points in position and orientation. The transformation is weighted by RBF, which is represented by

$$k_f(\boldsymbol{p}|\rho, \boldsymbol{c}) = \exp(-\rho^2 ||\boldsymbol{p} - \boldsymbol{c}||^2), \tag{6}$$

where $\rho \in \mathbb{R}^1$ and $\boldsymbol{c} \in \mathbb{R}^3$ are variables determining the shape of the function, corresponding to the slope and center position of RBF, respectively. The previous study in [5] presented a sufficient condition for diffeomorphism as:

$$0 < \rho_1 < \frac{1}{\sqrt{2}||\boldsymbol{v}_1||} e^{\frac{1}{2}}. \tag{7}$$

While conventional mapping can absorb translational error, it is not compatible with workspace rotation and reflection cases. That is because the optimization parameter $\boldsymbol{v}_1$ is used for a locally weighted shift. In this paper, we present two extended diffeomorphisms to address the rotation, reflection, and swapping of key points in workspaces.

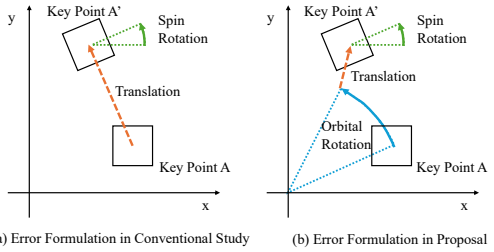


Fig. 2. Geometric error formulation among workspaces.

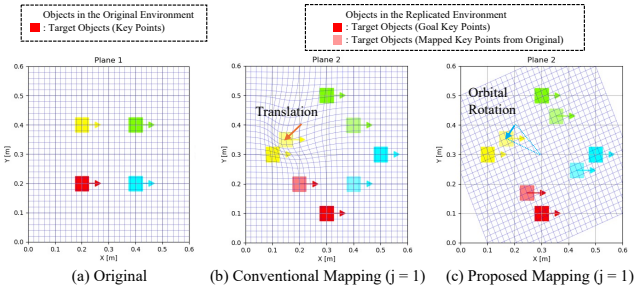


Fig. 3. Comparison of the initial mapping between DIFF and R-DIFF.

B. Rotation Extended Diffeomorphism

1) Orbital-Rotation for Error Representation: In order to improve the compatibility of the conventional diffeomorphic mapping with rotated situations, we redefine the geometric error formulation among the workspaces. Fig. 2 shows the error formulation when key point A shifts and rotates as key point A'. The conventional method shown in Fig. 2(a) represents the geometric error as translation and spin-rotation terms. Since consideration of the rotation of the key points group is not sufficient, we additionally consider the orbital-rotation term for position error, which provides a novel form of diffeomorphism compatible with orbital-rotational error as shown in Fig. 3. In the mathematical expression, we define the diffeomorphic function $\hat{\boldsymbol{\Phi}}$ in eq. (1) as $\hat{\boldsymbol{\Phi}}_{\mathrm{R-DIFF}}$ and modify the basic composition function for position in eq. (4) as follows:

$$\boldsymbol{\Phi}(\boldsymbol{p}) = \boldsymbol{v}_3{}^{k_f(\boldsymbol{p}|\rho_3, \boldsymbol{c}_3)} \cdot \boldsymbol{p}^{\boldsymbol{q}} \cdot \bar{\boldsymbol{v}}_3^{k_f(\boldsymbol{p}|\rho_3, \boldsymbol{c}_3)} + k_f(\boldsymbol{p}|\rho_1, \boldsymbol{c}_1) \cdot \boldsymbol{v}_1, \tag{8}$$

where $\boldsymbol{p}^{\boldsymbol{q}}$ is a pure quaternion form of $\boldsymbol{p}$. $\boldsymbol{v}_3$ is a quaternion representing the orbital-rotational error of key points.

2) Parameter Optimization: We present an algorithm for determining the parameters of our diffeomorphic mapping as shown in Algorithm 1. The sets of variable parameters for position and orientation mapping are composed of $\rho_{1,j}, \boldsymbol{c}_{1,j}, \boldsymbol{v}_{1,j}, \rho_{3,j}, \boldsymbol{c}_{3,j}, \boldsymbol{v}_{3,j}$ $(j = 1 \cdots J)$ and $\rho_{2,j}, \boldsymbol{c}_{2,j}, \boldsymbol{v}_{2,j}$ $(j = 1 \cdots J)$, respectively. In the optimization process, parameters $\boldsymbol{c}, \boldsymbol{v}$ are selected based on a geometric

relationship. After deciding $\boldsymbol{c}, \boldsymbol{v}$, a parameter ρ is selected to minimize the following cost functions:

$$F_{\Phi}(\rho_{1,j}, \rho_{3,j}, \boldsymbol{v}_{1,j}, \boldsymbol{v}_{3,j}) = \frac{1}{K}\sum_{k=1}^{K}\Big\|\boldsymbol{p}_{k,\text{goal}} - \Phi_j(\boldsymbol{p}_{k,j})\Big\|, \tag{9}$$

$$F_{\Psi}(\rho_{2,j}, \boldsymbol{v}_{2,j}) = \frac{1}{K}\sum_{k=1}^{K}\Big\|\log(\boldsymbol{q}_{k,\text{goal}} \otimes \overline{\Psi_j(\boldsymbol{q}_{k,j})})\Big\|, \tag{10}$$

where $\boldsymbol{p}_{k,\text{goal}}$ denotes the goal position of key points, $\boldsymbol{q}_{k,\text{goal}}$ denotes the goal orientation of key points, and $\boldsymbol{r}_{k,\text{goal}}$ denotes the goal orbital-rotation of key points. k is the index of key points. $\boldsymbol{q}_0$ is a null rotation quaternion. The composition iteration of RBF is terminated when the position and orientation errors fall below a predefined minimum threshold, or when the number of compositions j reaches the maximum J_{max}.

Algorithm 1 Diffeomorphism Parameter Optimization

Require: $\boldsymbol{P}_{\text{inits}} = \{\boldsymbol{p}_{k,\text{init}}, \boldsymbol{q}_{k,\text{init}}, \boldsymbol{r}_{k,\text{init}}\}_{k=1}^{K}$, $\boldsymbol{P}_{\text{goals}} = \{\boldsymbol{p}_{k,\text{goal}}, \boldsymbol{q}_{k,\text{goal}}, \boldsymbol{r}_{k,\text{goal}}\}_{k=1}^{K}$
Ensure: $\boldsymbol{\rho} = \{\boldsymbol{\rho}_{1,j}, \boldsymbol{\rho}_{2,j}, \boldsymbol{\rho}_{3,j}\}_{j=1}^{J}$, $\boldsymbol{C} = \{\boldsymbol{c}_{1,j}, \boldsymbol{c}_{2,j}, \boldsymbol{c}_{3,j}\}_{j=1}^{J}$, $\boldsymbol{V} = \{\boldsymbol{v}_{1,j}, \boldsymbol{v}_{2,j}, \boldsymbol{v}_{3,j}\}_{j=1}^{J}$

Initialize $\boldsymbol{P}_j \leftarrow \boldsymbol{P}_{\text{inits}}$
for error (9), (10) small enough **do**
  $m := \arg\max(||\boldsymbol{p}_{k,j} - \boldsymbol{p}_{k,\text{goals}}||)$, $k\in[1,\cdots,K]$
  $n := \arg\max(||\log(\boldsymbol{q}_{k,j} * \overline{\boldsymbol{q}_{k,\text{goals}}})||)$, $k\in[1,\cdots,K]$
  $o := \arg\max(||\log(\boldsymbol{r}_{k,j} * \overline{\boldsymbol{r}_{k,\text{goals}}})||)$, $k\in[1,\cdots,K]$
  $\boldsymbol{c}_{1,j}, \boldsymbol{c}_{2,j}, \boldsymbol{c}_{3,j} \leftarrow \boldsymbol{p}_{m,j}, \boldsymbol{p}_{n,j}, \boldsymbol{p}_{o,j}$
  $\boldsymbol{v}_{1,j}, \boldsymbol{v}_{2,j}, \boldsymbol{v}_{3,j} \leftarrow (\boldsymbol{p}_{m,\text{goal}} - \boldsymbol{p}_{m,j}), (\boldsymbol{q}_{n,\text{goal}} * \overline{\boldsymbol{q}_{n,j}}), (\boldsymbol{r}_{o,\text{goal}} * \overline{\boldsymbol{r}_{o,j}})$
  Optimize $\boldsymbol{\rho}_{1,j}$ to minimize $F_{\Phi}(\rho_{1,j}, 0, \boldsymbol{v}_{1,j}, \boldsymbol{q}_0)$
  Optimize $\boldsymbol{\rho}_{2,j}$ to minimize $F_{\Psi}(\rho_{2,j}, \boldsymbol{v}_{2,j})$
  Optimize $\boldsymbol{\rho}_{3,j}$ to minimize $F_{\Phi}(0, \rho_{3,j}, \boldsymbol{0}, \boldsymbol{v}_{3,j})$
  if $F_{\Phi}(0, \rho_{3,j}, \boldsymbol{0}, \boldsymbol{v}_{3,j}) < F_{\Phi}(\rho_{1,j}, 0, \boldsymbol{v}_{1,j}, \boldsymbol{q}_0)$ **then**
    $\rho_{1,j}, \boldsymbol{v}_{1,j} \leftarrow 0, \boldsymbol{0}$
  else
    $\rho_{3,j}, \boldsymbol{v}_{3,j} \leftarrow 0, \boldsymbol{q}_0$
  end if
  $\boldsymbol{p}_{k,j+1}, \boldsymbol{q}_{k,j+1} = \Phi_j(\boldsymbol{p}_{k,j}), \Psi_j(\boldsymbol{q}_{k,j})$
end for

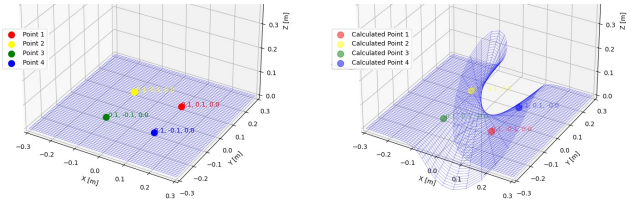
C. Twisted Affine Extended Diffeomorphism

1) Twisted Affine Transformation: In this paper, assuming that there are four key points, we present a twist transformation to address key points swapping. For instance, the twist transformation of the roll axis can be defined using a rotation matrix in the roll direction R_{roll} as:

$$\hat{\boldsymbol{p}} = R_{\text{roll}}(\phi_t) \cdot \boldsymbol{p}, \tag{11}$$

where the twist angle $\phi_t(x)$ is a sigmoid function, which can be written as:

$$\phi_t(x) = \phi_{t0} \cdot \frac{1}{1 + e^{-\rho_{\text{sig}} x}}, \tag{12}$$



(a) Original (b) Twist Transformed

Fig. 4. Our presented twist transformation of the xy-plane.

where ρ_{sig} is a parameter of the intensity of the twist. The result of applying the twist transformation of the roll axis to the xy-plane is shown in Fig. 4. In this paper, the twist transformation is defined in roll and pitch directions.

By combining the twist transformation with the affine transformation, the mapping can address key points' scaling, rotation, reflection, and swapping simultaneously. We present the mapping method as Twisted Affine Extended Diffeomorphism, written as:

$$\hat{\boldsymbol{\Phi}}_{\text{TA}}(\boldsymbol{p}) = R_{\text{TA}}\boldsymbol{p} + \boldsymbol{t}_{\text{TA}}, \tag{13}$$

where

$$\begin{aligned} R_{\text{TA}} = {} & R_{\text{scaling}}(a_x, a_y, a_z) \cdot R_{\text{shear}}(s_{xy}, s_{xz}, s_{yz}) \\ & \cdot R_{\text{rotation}}(\phi, \theta, \psi) \cdot R_{\text{reflection}}(r_x, r_y, r_z) \\ & \cdot R_{\text{twist}}(\phi_t(x), \theta_t(y)), \end{aligned} \tag{14}$$

$$\boldsymbol{t}_{\text{TA}} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T. \tag{15}$$

The definitions of each rotation matrix are provided in the Appendix. The parameters are shown in Table I.

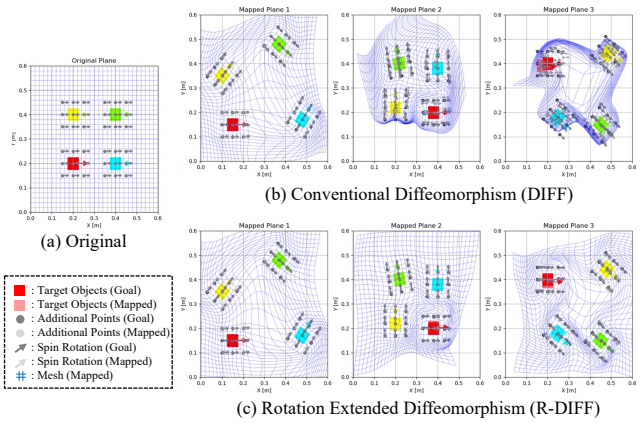
TABLE I
PARAMETERS OF THE TWISTED AFFINE TRANSFORMATION

Parameter	Meaning	Range
a_x, a_y, a_z	Scaling along x, y, z axes	$[0.5, 2]$
s_{xy}, s_{xz}, s_{yz}	Shear in xy, xz, yz planes	$[0, 1]$
ϕ, θ, ψ	Rotation around x, y, z axes	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
r_x, r_y, r_z	Reflection along x, y, z axes	$\{-1, 1\}$
ϕ_t, θ_t	Twist angles around x, y axes	$\{0, \pi\}$
t_x, t_y, t_z	Translation vector	$--$

Algorithm 2 Twisted Affine Parameter Optimization

Require: $\boldsymbol{P}_{\text{inits}} = \{\boldsymbol{p}_{k,\text{init}}, \boldsymbol{q}_{k,\text{init}}\}_{k=1}^{K}$, $\boldsymbol{P}_{\text{goals}} = \{\boldsymbol{p}_{k,\text{goal}}, \boldsymbol{q}_{k,\text{goal}}\}_{k=1}^{K}$
Ensure: $\boldsymbol{A} = \{a_x, a_y, a_z, s_{xy}, s_{xz}, s_{yz}, \phi, \theta, \psi\}$ $\boldsymbol{B} = \{r_x, r_y, r_z, \phi_{t0}, \theta_{t0}\}$ $\boldsymbol{T} = \{t_x, t_y, t_z\}$

$\boldsymbol{T} \leftarrow \sum_{k=1}^{K}(\boldsymbol{p}_{k,\text{goal}}) - \sum_{k=1}^{K}(\boldsymbol{p}_{k,\text{init}})$
for $\boldsymbol{B}$ choose candidates from TABLE I **do**
  Optimize $\boldsymbol{A}$ to minimize positional error F_{TA}
  if $F_{\text{TA}} < best\_objective$ **then**
    $best\_objective \leftarrow F_{\text{TA}}$
    $\boldsymbol{A}\_opt, \boldsymbol{B}\_opt \leftarrow \boldsymbol{A}, \boldsymbol{B}$
  end if
end for
$\boldsymbol{A}, \boldsymbol{B} \leftarrow \boldsymbol{A}\_opt, \boldsymbol{B}\_opt$

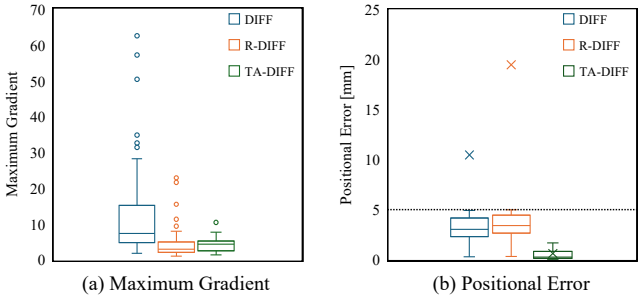


(a) Original

(b) Conventional Diffeomorphism (DIFF)

(c) Rotation Extended Diffeomorphism (R-DIFF)

Fig. 5. Simulation1: Comparison of the mapping shape between DIFF and R-DIFF.



(a) Maximum Gradient

(b) Positional Error

Fig. 6. Simulation2: Comparison of evaluation index between DIFF, R-DIFF, and TA-DIFF.

TABLE II

EVALUATION METRICS IN SIMULATION 2

	(a) DIFF	(b) R-DIFF	(c) TA-DIFF
Positional Error (Max)	145 mm	211 mm	4.73 mm
Success Rate	90 %	87 %	100 %
RBF Iteration (Average)	39	29	10

2) Parameter Optimization: Algorithm 2 shows the optimization process. The discrete parameters $\boldsymbol{B}$ are selected in order. The continuous parameters $\boldsymbol{A}$ are selected with the BFGS method to minimize the following cost functions:

$$F_{\mathrm{TA}} = \frac{1}{K}\sum_{k=1}^{K}\left\|\boldsymbol{p}_{k,\mathrm{goal}} - \Phi_{\mathrm{TA}}(\boldsymbol{p}_{k,\mathrm{init}})\right\|. \tag{16}$$

III. Manipulator Control

We simultaneously control multiple robots, which are placed in geometrically randomized workspaces, using our presented motion replication method. We input a target end-effector pose of the primary robot using a teleoperation interface. The other robots follow transformed target poses calculated from eq. (1). Then we obtain an augmented variable $\boldsymbol{x}_i$ of the target pose of i-th robot as $\boldsymbol{x}_i = [\boldsymbol{p}_i^T \ \boldsymbol{q}_i^T]^T$

The robot controller to follow the target poses is implemented using OpenHRC [6]. We apply a proportional control shown in eq. (17) and use its output as target end-effector velocities.

$$\dot{\boldsymbol{x}}_i = k_p(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i), \tag{17}$$

where $\bar{\boldsymbol{x}}_i$ denotes the current pose of the i-th robot. $k_p = 2.0$ is the proportional gain. The inverse kinematics that converts the desired end-effector velocity to the desired joint velocity is formulated as a QP (quadratic programming) optimization problem as shown in eq. (18).

$$\min_{\dot{\boldsymbol{\theta}}_i} \sum_{i}^{N}\left(\left|\boldsymbol{J}_i(\boldsymbol{\theta}_i)\dot{\boldsymbol{\theta}}_i - \dot{\boldsymbol{x}}_i\right|^2 + |\dot{\boldsymbol{\theta}}_\mathbf{i}|^2\right), \tag{18}$$

where N is the number of robots. $\boldsymbol{\theta}_i$ and $\boldsymbol{J}_i(\boldsymbol{\theta})$ is the joint angle and Jacobian matrix of i-th robot, respectively. The obtained joint velocity is sent to the robot via a Joint Velocity Controller.

IV. Simulation

A. Simulation 1

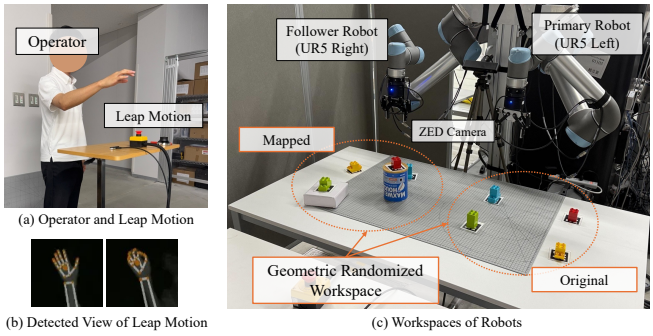
We conduct numerical simulations to compare the conventional Diffeomorphism (DIFF) and the proposed Rotation Extended Diffeomorphism (R-DIFF). In this simulation, multiple sub-key points around target objects are added to improve the consistency of geometric information around targets. The number of target objects K is set to 4. Sub key points around each target object are placed from a distance of 0.05 m in x,y,z directions. The total number of sub-key points K_s is 27. All points, including target objects and sub-key points around target objects, are used as key points $\boldsymbol{P}_{\mathrm{inits}}, \boldsymbol{P}_{\mathrm{goal}}$ in Algorithm 1. The number of replicated environments is 3. The maximum number of RBF iterations J_{max} is experimentally set to 100. Iteration ends when eq. (9) is under 5 mm.

Fig. 5 shows the results of simulation 1. In the case of DIFF, a steep gradient and positional error are confirmed in Planes 2 and 3. However, in the case of R-DIFF, the gradient is gentle and the positional error is nearly zero. In Plane 3, the positional error of the key points is 18.65 mm (DIFF), 5.58 mm (R-DIFF).

B. Simulation 2

Simulation 2 examines the reliability of the mapping in various situations. DIFF, R-DIFF, and Twisted Affine Diffeomorphism (TA-DIFF) are compared in the situation where target objects are placed randomly within the range $\boldsymbol{p}_{k,\mathrm{goal}} \in [0.1, 0.6]^3$. There are no sub-key points around target objects ($K_s = 0$). The number of replicated environments N is 100. The other conditions are the same as ones in simulation 1.

Fig. 6 is the boxplot showing the maximum gradient and positional error between target objects. The maximum gradient is calculated by the spectral norm of the Jacobian matrix for each grid point set at 0.02 m intervals on the map. The positional error between target objects is calculated using the value of eq. (9) at the end of the mapping algorithm. From the results, the proposed method was confirmed to

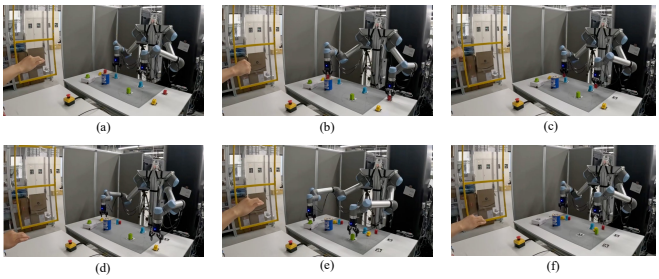


(a) Operator and Leap Motion

(b) Detected View of Leap Motion

(c) Workspaces of Robots

Fig. 7. Experimental setup.



(a) (b) (c) (d) (e) (f)

Fig. 8. Task flow in Experiment 2. (a) Initial state (b) Picking red target (c) Carrying red target (d) Picking yellow target (e) Picking green target (f) Finished all picking task.

generate smoother mappings with smaller velocity gradients, taking into account the geometric relationships between target objects. Regarding positional errors, TA-DIFF achieved less than 5 mm in all situations, but R-DIFF showed larger positional errors. This would be because eq. (8) for R-DIFF does not provide appropriate compensation in some cases. Conversely, TA-DIFF could roughly resolve the twist errors in the TA transformation step, which would result in remarkably low positional error. Table II summarizes the other evaluation values for each method. In DIFF and R-DIFF, there were approximately 10 % cases where the error did not fall below the desired value after 100 RBF superpositions. However, the proposed TA-DIFF guaranteed that the position error in the mapped environment would always fall below the target minimum error under all conditions. Furthermore, optimization calculation time has positive correlation to J and K.

V. Experiment

A. Setup

To verify the effectiveness of the proposed method, we conduct experiments using a dual-arm UR5 (Universal Robots) with 2F-85 grippers (Robotiq) [1] . A ROS2 velocity controller is implemented for the robot. The experimental setup of the dual-arm UR5 robot is shown in Fig. 7. The robot is connected to a PC running Ubuntu 22.04 with a PREEMPT_RT patched kernel. The right arm is offset in the y-direction by +0.6 m relative to the left arm.

[1]software:https://github.com/fzi-forschungszentrum-informatik/robotiq_2f_urcap_adapter

Table III lists the target positions used in each experiment. In Experiment 2, red, yellow, green, and blue blocks are used as target objects representing the robot's environment. Their positions and orientations are detected using AprilTag (AR marker) and a ZED 2i stereo camera. We employ a Leap Motion Controller 2.

TABLE III
Target positions in each environment

	Right Arm	Left Arm
Experiment 1	Preset Values	Preset Values
Target 1 ●	$(0.50, 0.20, 1.20)$	$(0.55, 0.16, 1.22)$
Target 2 ●	$(0.70, 0.20, 1.20)$	$(0.45, 0.41, 1.26)$
Target 3 ●	$(0.70, 0.40, 1.20)$	$(0.68, 0.24, 1.02)$
Target 4 ●	$(0.50, 0.40, 1.20)$	$(0.79, 0.48, 1.02)$
Experiment 2	Detected (AprilTag)	Detected (AprilTag)
Target 1 ●	$(0.492, 0.502, 0.898)$	$(0.746, 0.344, 1.061)$
Target 2 ●	$(0.488, 0.163, 0.919)$	$(0.482, 0.334, 0.912)$
Target 3 ●	$(0.721, 0.164, 0.927)$	$(0.849, 0.146, 0.961)$
Target 4 ●	$(0.756, 0.506, 0.905)$	$(0.564, 0.059, 0.896)$

B. Experiment 1

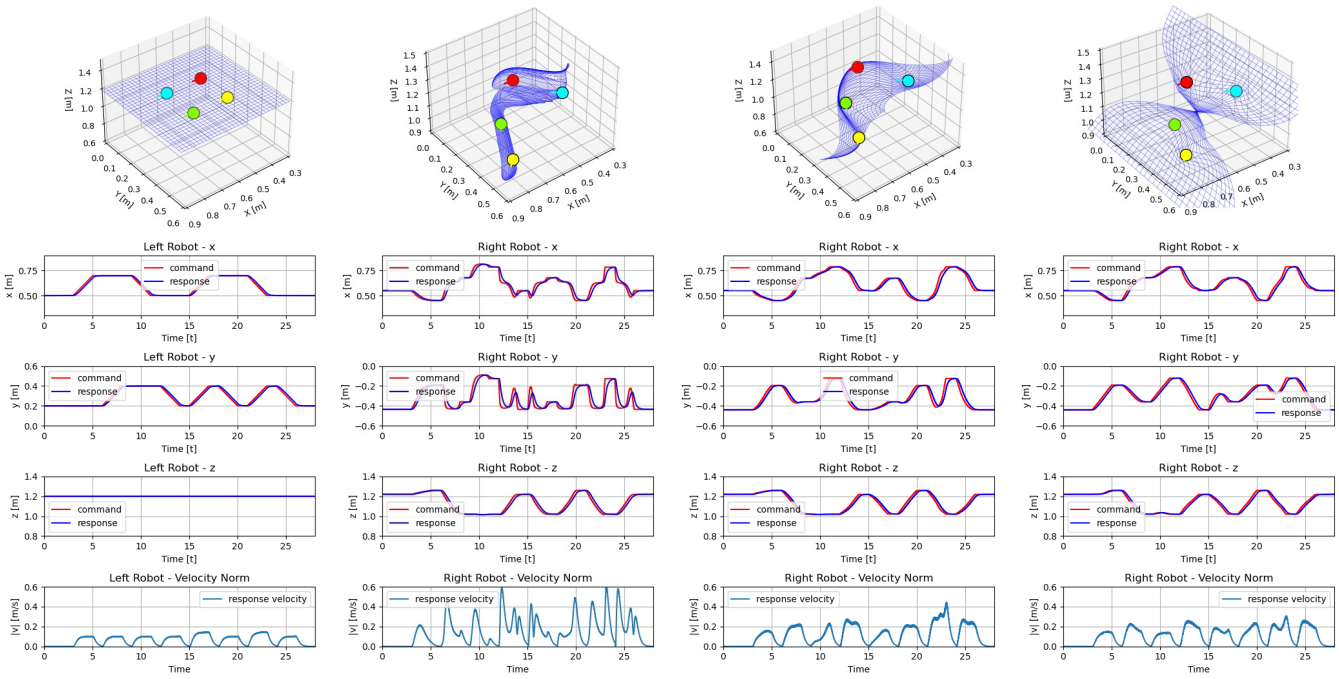
In Experiment 1, the primary (left) robot follows a trajectory in the order: Target 1 $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ 4 $\rightarrow$ 1 $\rightarrow$ 3 $\rightarrow$ 2 $\rightarrow$ 4. We assess the responses of the follower (right) robot under the following mapping methods: DIFF, R-DIFF, and TA-DIFF.

Fig. 9 shows the command and response position of the end-effector, and the velocity norm for each mapping. In DIFF, oscillatory behavior was observed in the position response when moving from Target 4 to 1. Besides, the generated motion in DIFF environment included a trajectory with a larger velocity norm than in the original environment. In contrast, with the proposed R-DIFF and TA-DIFF, the velocity deviation from the original trajectory was small, and the position response remained smooth. The evaluation indexes for each mapping are shown in Table IV. The proposed mappings significantly reduced the velocity gradient while achieving the required positional accuracy.

C. Experiment 2

In Experiment 2, the operator conducts a task grasping Targets 1–4 and moving them to the location of Target 2 in both the primary and follower environment at the same time, as shown in Fig. 8. The pinch and grab gestures of the operator's hand are used as triggers to activate the robot operation and open/close the mounted gripper, respectively. The operator conducts the task three times.

In the case of DIFF, the follower robot sometimes moved very quickly even though the commanded velocity for the primary robot was not so large. This caused an emergency stop because the follower robot reached its safety limiter. In the case of R-DIFF, no emergency stop of the follower robot occurred. However, the prediction of the movement of the follower robot would be difficult because the shape of the map is not intuitive to the human operator, which prolongs the task completion. In the case of TA-DIFF, thanks to the



(a) Original (b) Diffeomorphism (c) Rotation Diffeomorphism (d) Twisted Affine Diffeomorphism

Fig. 9. Results of each mapping method and position/velocity response of the robot in Experiment 1.

TABLE IV
EVALUATION METRICS IN EXPERIMENT 1

	(a) DIFF	(b) R-DIFF	(c) TA-DIFF
Positional Error	3.26 mm	3.36 mm	2.94 mm
Maximum Gradient	44.13	5.41	5.39
RBF Iteration	88	28	4
TA Optimization Time	-	-	0.45 s
DIFF Optimization Time	0.64 s	0.23 s	0.02 s

smooth and intuitive shape of the mapping, the operator was able to grasp all targets.

VI. DISCUSSION

The proposed R-DIFF is compatible with an infinite number of key points theoretically, which is advantageous when adding multiple sub-key points around target objects, as in simulation 1. Compared to DIFF, R-DIFF tends to improve the mapping gradient, but in some cases, the mapping error is higher than DIFF. TA-DIFF performs the best in terms of mapping error and mapping gradient, but since it assumes that the number of key points is four, further consideration is needed for more than 5 key points.

VII. CONCLUSION

In this paper, we presented the Rotation Extended Diffeomorphism and the Twisted Affine Extended Diffeomorphism, and their application as a mapping method for real-time motion replication. Through simulation, our approach is more compatible than the conventional method in terms of mapping gradient and mapping error. The experimental results show that the proposed mappings enabled smooth position mappings for robots in the replicated environment, while maintaining the smallest velocity deviation from a primary robot. These findings indicate that the proposed method would be effective for real-time motion reproduction in multi-robot teleoperation. Future work will address mappings at levels of velocity and acceleration to facilitate smooth environmental interaction of each robot.

APPENDIX

$$R_{\text{scaling}}(a_x, a_y, a_z) = \text{diag}(a_x, a_y, a_z) \tag{19}$$

$$R_{\text{shear}}(s_{xy}, s_{xz}, s_{yz}) = \begin{bmatrix} 1 & s_{xy} & s_{xz} \\ 0 & 1 & s_{yz} \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$$R_{\text{reflection}}(r_x, r_y, r_z) = \text{diag}(\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z}) \tag{21}$$

$$R_{\text{rotation}}(\phi, \theta, \psi) = R_{\text{yaw}}(\psi) \cdot R_{\text{pitch}}(\theta) \cdot R_{\text{roll}}(\phi) \tag{22}$$

$$R_{\text{twist}}(\phi_t(x), \theta_t(y)) = P \cdot R_{\text{pitch}}(\theta_t(y)) \cdot P \cdot R_{\text{roll}}(\phi_t(x)) + (I - P) \tag{23}$$

$$P = \text{diag}(1, 1, 0) \tag{24}$$

REFERENCES

[1] I. Ozdamar, M. Laghi, G. Grioli, A. Ajoudani, M. G. Catalano and A. Bicchi, "A Shared Autonomy Reconfigurable Control Framework for Telemanipulation of Multi-Arm Systems", IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 9937–9944, 2022.

[2] F. Conti and O. Khatib, "Spanning large workspaces using small haptic devices", In proceedings of World Haptics Conference, pp.183–188. 2005.

[3] H. Kobayashi and H. Nakamura, "A scaled teleoperation", In proceedings of IEEE International Workshop on Robot and Human Communication, pp. 269–274, 1922.

[4] L. Guanyang, G. Xuda, L. Lingzhi, and W. Yan, "Haptic based teleoperation with master-slave motion mapping and haptic rendering for space exploration", Chinese Journal of Aeronautics, vol. 32, no. 3, pp. 723-736, 2019.

[5] X. Gao, J. Silvério, E. Pignat, S. Calinon, M. Li and X. Xiao, "Motion Mappings for Continuous Bilateral Teleoperation", IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 5048–5055, 2021.

[6] "OpenHRC", https://github.com/OpenHRC/OpenHRC