Human hand motion retargeting for dexterous robotic hand

Jedrzej Orbik[1], Shile Li[1], Dongheui Lee[1,2]

***Abstract*— One way to achieve dexterous manipulation autonomously with natural input is through learning by demonstration. Unfortunately, grasping an object with a complex dexterous hand is a complicated task. To facilitate the demo acquisition process, we propose a low-cost framework to map the human hand motion from a single RGB-D camera using inverse kinematics. This framework has been implemented in a CoppeliaSim simulation environment. We evaluate two multi-task handling methods and a low-pass filter using two obtained trajectories. Empirically, the proposed framework can successfully perform grasping task imitations. An exemplary video of the object manipulation is presented on the project website: `https://sites.google.com/view/retargeting-for-dexterous-hand`**

I. Introduction

Correct imitation of the human hand motion with robotic hand model is a challenging task. The model of human hand has a high number of degrees of freedom, which makes tracking and retargeting difficult. There are discrepancies between the human and robotic hand and errors occur during tracking and estimation of the human hand pose. The additional challenge is that the pose estimation may cause an anatomically infeasible position, and this effect is reinforced by the noise from the tracking data. This had to be considered when calculating desired joint positions during motion.

The current state-of-the-art Hand Pose Estimation (HPE) methods have significantly improved tracking capabilities from single images, and complicated motion tracking systems can be replaced with a single depth camera. That is because of the emergence of new discriminative methods based on deep neural networks, which can deal with the self occlusions of the human hand and correctly segment the hand from the environment [1], [2]. The state-of-the-art approaches still suffer from the occlusions when interacting with the objects, but this problem can be mitigated after a wider variety of hand-object interactions dataset becomes available. This allows us to take the step further and use these pose estimation for training the robot motion in complex dexterous manipulation tasks.

This research aims to create the bridge between hand motion tracking and imitation learning. Imitation learning is the approach that helps to the tedious work when each move of the robot during task execution had to be precisely engineered. Here, the agent attempts to emulate human behavior based on the provided demonstrations. The common approaches incorporate Hidden Markov Models [3], [4], dynamic movement primitives [5], deep neural networks [6], [7] or reinforcement learning methods [8], [9].

We created the retargeting algorithm of the human hand motion to the anthropomorphic robotic hand. We execute the imitation learning method in the simulated environment and provide the interface for future applications with hardware. The applied HPE algorithm [10] provides the position of each of 21 joints of the human hand as depicted in Fig. 1. We focused on the retargeting to the five-fingered robotic hand, since mapping to robotic hand models of different morphology would require a consideration of the function of each finger and readjustment of the kinematics accordingly.

The presented work makes following contributions:

1) Provides a low-cost framework for human hand trajectory acquisition for learning by demonstration.
2) Motion retargeting for hand structure taking into account the complex kinematics in the global frame.
3) Mitigation of the lack of smoothness in the input pose estimation using low-pass filter and point pattern matching.

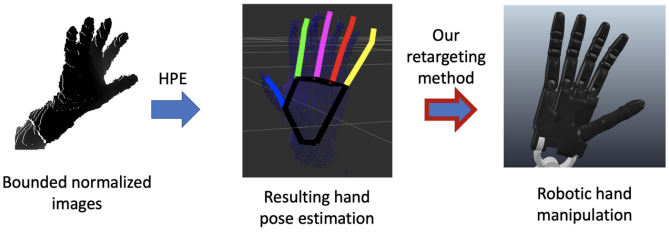

Fig. 1. General view on the hand motion retargeting architecture. The bounded pre-scaled depth images are being fed to the neural network architecture [10] to infer the current hand pose. The positions of the hand joints are used as the input to our retargeting method to replicate the trajectory with the dexterous robotic hand.

II. Related work

The hand pose estimation is currently the subject of intensive research. This is a very challenging problem because of the inherent high dimensionality of the hand model action space (more than 20 DOFs), self-occlusions of the hand parts, or occlusions by the other objects during hand object interaction, which make the problem especially difficult. Since the advent of depth-sensing devices at a consumer level, a lot of depth image-based hand pose estimation methods appeared. Current state-of-the-art methods [11], [12] use Convolutional Neural Networks (CNN) to regress the hand pose directly from a single image.

[1]Jedrzej Orbik, Shile Li and Dongheui Lee are with the Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany, Germany

[2]Dongheui Lee is also with Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany

To tackle the above mentioned difficulties, traditionally, researchers have used tracking based methods [13], [14], which rely on sequence of data and a good initialization pose for this problem, then each frame's estimation is optimized from the previous frame's estimates. In recent years, deep learning proves to be successful in different research fields. Hand pose estimation is no exception, as state-of-the-art results are all coming from deep learning based methods [10], [11], [12]. Relying on large annotated datasets, deep learning methods are suitable to find out a highly non-linear mapping function between a single input image and the hand pose output. In our framework, we will use the HPE method from [10] to obtain hand pose from a depth camera. The task of motion retargeting has been solved in the previous works for the whole human body using the inverse kinematics, which is a difficult task due to different morphology of the human and robot body and redundancy of the controller. Early work by Nakamura et al. has coped with redundancy by the introduction of multi-task prioritized controllers. Ayusawa and Yoshida [15] have approached the problem of adjusting the morphology between the subject human and robot using a morphing function.

The motion retargeting of dexterous hands has been addressed in a number of different works. Kumar et al. [16] have created a professional human hand motion capture system using an expensive CyberGlove. In the method by Rossel et al. [17] the complexity of the retargeting of dexterous robotic hand has been minimized by the use of principal component analysis. Handa et al. [18] have developed a system for human hand motion retargeting, but it required multiple cameras to ensure a smooth trajectory of the robot motion. A recent work by Garcia-Hernando et al. uses residual reinforcement learning trained on the interactions with the environment which limits its application in practice. In the work by Li et al. [19] the camera input has been coupled with the inertial measurement unit reading to allow image-to-image translation producing the expected robot movement.

Our method does not require any additional sensors apart from a ubiquitous RGB-D camera, such as a Kinect camera. It is lightweight and comparing to other methods, does not require task-specific training prior to the task execution. It can be deployed on a variety of real-world and simulated environments based on the camera tracker only.

III. Method

Fig. 2 shows the overview of the proposed retargeting framework and the detailed processing of the HPE diagram is presented in Fig. 3.

In the proposed approach, the HPE method by Li and Lee [10] is being used. Hand pose tracking provides 21 Cartesian positions of the joints inferred from the depth image stream from the RGB-D camera. The estimated pose is given with an altering frequency, which influences the smoothness of the trajectory and imposes the use of low-pass pose filtering.

The position of the end effector defined by the task descriptor $\mathbf{x} \in \mathbb{R}^m$ is conveniently calculated from the joint vector $\mathbf{q} \in \mathbb{R}^n$ according to equation of the forward kinematics [20]:

$$\mathbf{x} = f(\mathbf{q}), \tag{1}$$

where $k_{\mathbf{x}}(\mathbf{q})$ is non-linear mapping function, m is the dimension of the task descriptor $\mathbf{x}$ (commonly equals 6 for the given pose $\mathbf{p} \in \mathbb{R}^3$ and angles $\theta \in \mathbb{R}^3$) and n is the dimension of the task-space (amount of the DOFs in the manipulator configuration).

A. Low-pass filtering

Each new inference of the HPE algorithm underlies filtering with the low-pass filter. The smoothing of the motion is necessary because each result of pose estimation is independent of the preceding ones. This may result in sudden changes, which are kinematically implausible.

The low-pass filter is implementing according to the formula:

$$\mathbf{x}_{d,\text{new}} = \mathbf{x}_{\text{HPE}}\alpha + \mathbf{x}_{d,\text{old}}(1-\alpha), \tag{2}$$

where each x denotes the Cartesian positions of the joints, and x_d describes the desired position.

The smoothing parameter α, which takes values in range $(0, 1)$ has been selected to a low value 0.2, which increased the response time but still contributes to the steadiness of the mapping.

B. Global hand pose estimation

In order to execute the inverse kinematics of the hand, it is convenient to transform the inferred HPE to the hand's homogeneous position and execute the control in the hand frame only. We propose to use the least-squares estimation described by Umeyama [21] to estimate the global 6D hand pose by the calculation of transformation parameters between two point patterns with known correspondences.

It defines a closed-form solution to the absolute orientation problem [22]:

$$e^2(\mathbf{R}, \mathbf{t}, c) = \frac{1}{n}\sum_{i=1}^{n} \|y_i - (c\mathbf{R}x_i + \mathbf{t})\|^2, \tag{3}$$

where $\mathbf{R}$ is the rotation matrix, $\mathbf{t}$ is the translation vector, c is the scaling factor, x_i is the position of the joint with index i in the homogenous global frame, and y_i is the corresponding target position of the joint inferred by HPE. Scaling c is adjusted in the subsequent step to accommodate the finger lengths for successful tracking.

The solution to least squares estimation is found by solving:

$$\mathbf{R}, \mathbf{t}, c = \arg\min_{\mathbf{R},\mathbf{t},c} e^2. \tag{4}$$

The $\mathbf{R}$ matrix is found using lemma proven in the paper by Umeyama [21] using the singular value decomposition:

$$\mathbf{U}\mathbf{D}\mathbf{V}^T = \text{SVD}(\mathbf{X}\mathbf{Y}^T), \tag{5}$$

with X being a set of points in the world frame and Y a set of corresponding points in the homogeneous frame. The

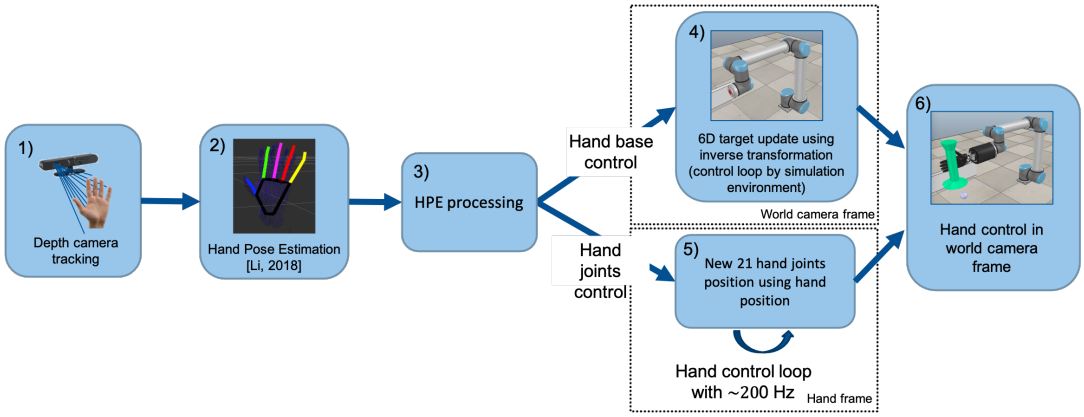

Fig. 2. Overview of the retargeting framework. 1) Single RGB-D camera stream 2) HPE algorithm [10], 3) HPE output processing presented closer in Fig. 3, 4) Base control from extracted hand base pose in world frame 5) Inverse kinematics control loop for the fingers in hand frame 6) This results in the dexterous hand control in the world frame.

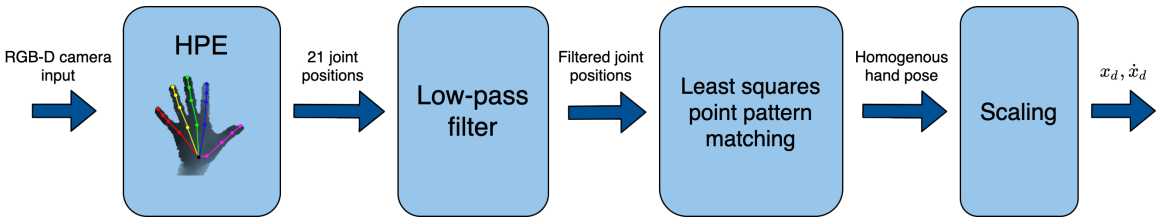

Fig. 3. Diagram of HPE output processing. Scaling is performed after after the global pose of hand frame has been estimated. This facilitates the hand scaling process.

result of SVD (singular value decomposition) provides the rotation matrix:

$$\mathbf{R} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{6}$$

if the patterns consist of at least 3 noncollinear point pairs. The matrix $\mathbf{S}$ is provided by:

$$\mathbf{S} = \begin{cases} \mathbf{I} & \text{if } \det(\mathbf{U})\det(\mathbf{V}) = 1 \\ \text{diag}(1, 1, ..., 1, -1) & \text{if } \det(\mathbf{U})\det(\mathbf{V}) = -1. \end{cases}$$

The calculation of optimal $\mathbf{t}$ and c is not described here for brevity please refer to the paper by Umeyama.

To find the position of the hand in the world camera coordinate system, we need to calculate the inverse transformation:

$$\mathbf{T}^{-1} = \begin{bmatrix} c\mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{c}\mathbf{R}^T & -\frac{1}{c}\mathbf{R}^T\mathbf{t} \\ 0 & 1 \end{bmatrix}, \tag{7}$$

using the property of the rotation matrices $\mathbf{R}^{-1} = \mathbf{R}^T$.

The target hand base pose is updated with each HPE reading using the same low-pass filter as introduced in (2). The inverse kinematics for the used UR-10 robotic arm is calculated using built-in functions of the CoppeliaSim simulation environment.

C. Scaling

Before the computation of the inverse kinematics, the last step of scaling from the human hand to robotic hand finger lengths is necessary. We are considering each of the fingers beginning with the finger's MP joint ($\mathbf{x}_1$) at the palm and match their position to the robotic hand's MP joints in the hand frame (refer to Fig. 4. Then we follow to the finger's tip $\mathbf{x}_{N+1}$ according to the equation:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} d_i, \tag{8}$$

with index i taking values from 1 to N, N as the number of joints in the finger. d_i corresponds to the length of the link from the Cartesian positions of the joint x_i, and:

$$\mathbf{v}_i = \mathbf{x}_i - \mathbf{x}_{i-1}. \tag{9}$$

The scaling allows the correct inverse kinematics calculation as long as reaching the desired task descriptor positions would not exceed the joint limits putting the manipulator in singularity.

D. Closed Loop Inverse Kinematics algorithm

In this work, inverse kinematics were performed based on the Closed Loop Inverse Kinematics algorithm (CLIK) which is widely used in robotics for the computation of the joint configuration from the desired position and orientation of the task descriptor. It properly deals with the singular configurations, which emerge when the target lies beyond the robot's reach (case when there are no solutions to the

problem) or when the robot is redundant with a higher number of DOF than the dimension of the task descriptor (case of infinitely many possible solutions).

For the simplification of the problem, the trajectory tracking of a single finger will be investigated.

The task descriptor denotes the position in the 3-dimensional Cartesian coordinates system corresponding with the x, y, z coordinates (orientation of the joints is not provided by the HPE and therefore is ignored):

$$\mathbf{x} \in \mathbb{R}^3, \ \mathbf{x} = [p_x, p_y, p_z]^T. \tag{10}$$

Configuration vector in joint space of single finger consists of 4 elements, one for each of the angles DIP and PIP joint, and 2 for MP joint angles (see Fig. 4).

$$\mathbf{q} \in \mathbb{R}^4, \ \mathbf{q} = [q_1, q_2, q_3, q_4]^T. \tag{11}$$

This implies $m = 3$, $n = 4$. Taking into consideration required dimensionality we can utilize the CLIK algorithm, which utilizes pseudo-inverse Jacobian regularization:

$$\mathbf{J}^* = \mathbf{W}_1^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}_1^{-1}\mathbf{J}^T + \mathbf{W}_2)^{-1}. \tag{12}$$

Here, $\mathbf{W}_1$ is a positive definite weight matrix $m \times m$, $\mathbf{W}_2$ is the damping matrix $n \times n$, which is necessary if $\mathbf{J}$ is ill-conditioned, imposing rank-deficiency of the Jacobian matrix, when lying in a singular configuration. Setting $\mathbf{W}_2 = 0$ and $\mathbf{W}_1 = \mathbf{I}^{m\times m}$ arrives at the usual formula of the original generalized pseudo-inverse:

$$\mathbf{J}^\dagger = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}. \tag{13}$$

The calculation of the required angular velocities $\dot{\mathbf{q}}$ is subject to a numerical optimization based on the given desired task velocity and suffers from numerical drift. To prevent this, a feedback correction is introduced by computation of the current position error e. The inverse kinematics is then described as:

$$\dot{\mathbf{q}} = \mathbf{J}^*(\dot{\mathbf{x}}_d + \mathbf{K}\mathbf{e}), \tag{14}$$

where $\mathbf{K}$ is positive definite gain matrix, and $\mathbf{e}$ is a vector that expresses the position error $[e_x, e_y, e_z]^T$. The position error is then specified as:

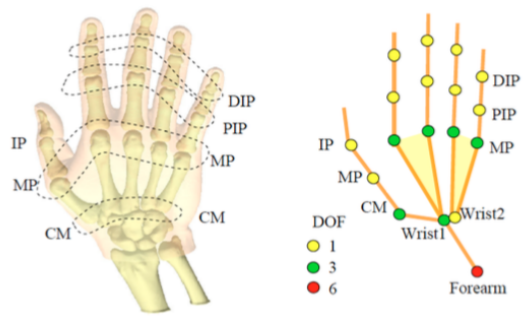

Fig. 4. The human hand skeleton with depicted positions of the joints align with the estimated 21 joint poses from HPE algorithm. [23]

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x}. \tag{15}$$

In the algorithm, the weight matrix W_1 is adjusted depending on the motion direction in the vicinity of the joint limits. That encourages the mechanism movements which escape from the constrained configurations. Specifically, the entries of the diagonal weight matrix are updated according to the following rule:

$$\theta_i = \begin{cases} 1 + \left|\frac{\partial H}{\partial q_i}\right| & \text{if } \Delta\left|\frac{\partial H}{\partial q_i}\right| \geq 0 \\ 1 & \text{if } \Delta\left|\frac{\partial H}{\partial q_i}\right| < 0, \end{cases} \tag{16}$$

where $H(q)$ is the performance criterion to prevent the motion beyond the joint limits and allow the uninterrupted movement away from them. $\frac{\partial H(q)}{\partial q_i}$ is its gradient specified as:

$$\frac{\partial H(q)}{\partial q_i} = \frac{(q_{i,max} - q_{i,min})^2(2q_i - q_{i,max} - q_{i,min})}{4(q_{i,max} - q_i)^2(q_i - q_{i,min}^2)}. \tag{17}$$

The gradient $\frac{\partial H(q)}{\partial q_i}$ is equal 0 in the middle of the joint movement range and equals infinity at its limit.

E. Handling multiple tasks

As the task descriptor of a single finger is of a dimension of 3 and we are controlling the finger, which has 4 DOFs, one of the dimensions is not going to be constrained. In order to avoid this, we create 2 subtasks for each finger: one for the finger tip and one for the DIP joint of each respective finger. Two methods for handling multiple tasks were taken into consideration:

1) Task augmentation
2) Task prioritization

After the evaluation, task prioritization was selected as the optimal solution. The comparison of the two approaches is presented in section IV-C.

1) Task augmentation: This attempts to satisfy all tasks in parallel and thus exploits all the available DOFs. In our case, it was not delivering acceptable results as the desired dimension of the task descriptors (2xm) was too high to satisfy using the available number of joints n in the mechanism. This leads to the failure of all the tasks during trajectory targeting.

2) Task prioritization: This method allows the definition of task descriptors as the subtasks with the order of priority. The prioritized solution is then given by the procedure defined in Algorithm 1. For more details on the task prioritization and augmentation methods, we refer to the work by Dariush et al. [24].

Task prioritization allows the movement of the mechanism in the lower priority in the null-space of the higher priority tasks and exploits all the available DOFs in the process. Here, the main task was selected at the top and the secondary task at the joint DIP of each finger.

Algorithm 1: Solving multi-tasking IK with task prioritization [24]

Input: Tasks i = 1,...,k each defined by its Jacobian matrix $\mathbf{J}_i$ and task descriptor $\dot{\mathbf{x}}_{d,i}$

Output: $\dot{\mathbf{q}}$ joint velocities

$\mathbf{N}_0 = \mathbf{I}$

for *i = 1..k* **do**

$\mathbf{v}_i = \dot{x}_{d,i}$

$\hat{\mathbf{J}}_i = \mathbf{J}_i \mathbf{N}_{i-1}$

$\hat{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{J}_i \sum_{j=i}^{i-1} (\mathbf{J}^{\dagger} \hat{\mathbf{v}})$

$\dot{\mathbf{q}} = \sum_{i=1}^{k} (\hat{\mathbf{J}}_i \hat{\mathbf{v}}_i) + \mathbf{N}_k \mathbf{z}$,
with $\mathbf{z}$ being an arbitrary vector

IV. Experiments

To assess the validity of our approach we use the well-established CoppeliaSim. This simulator has been widely used in previous methods [25], [26], [27]. An exemplary scene used for the evaluation of grasping is depicted in Fig. 5.

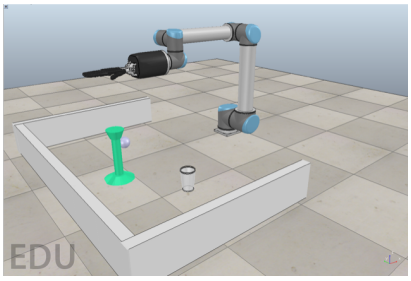

Fig. 5. CoppeliaSim simulation environment scene used during the experiments. Shadow Dexterous Hand model was mounted on the UR-10 robotic arm.

For the experiments, 2 trajectories with arbitrary finger motion of roughly 20 seconds were collected for the comparison of the parameters and methods of the mapping algorithms. The experiments were conducted with α value of 0.2. Errors are calculated as the sum over Euclidean distances between the desired position of all of the finger joints (including finger tips) and their corresponding mapped positions and sampled with 10 Hz frequency.

A. Motion imitation

Fig. 6 and Fig. 7 depict the results of two of these trajectories with task prioritization and two task handles. In Fig. 6 there is only one sudden position change when the fingers are bent. In turn, the fingers on trajectory in Fig. 7 are in constant motion.

It can be seen, that the sum of the errors for fingers converges below 0.1 m with the error of each of the fingers converging below 1 cm, as long as the finger joint angles were possible to satisfy by the robotic hand. For example, in Fig. 6 the middle finger position cannot be satisfied at the end of the trajectory mapping and the error stays close to 2 cm. It can be also observed, that the error is smallest

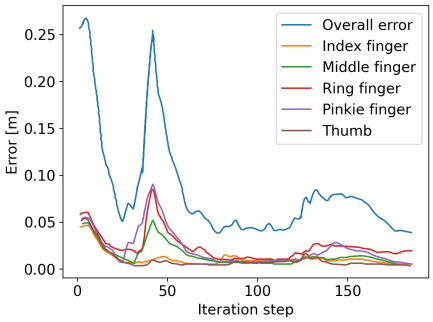

Fig. 6. Errors for each of the fingers with slow finger movement trajectory.

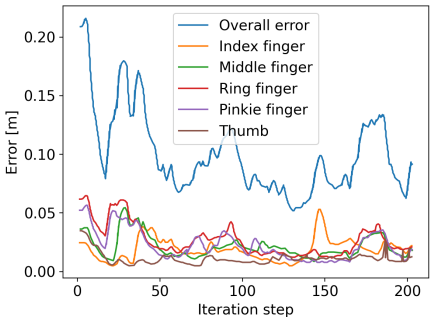

Fig. 7. Errors of pose mapping for faster finger movements.

on both trajectories for the thumb, which has 5 DOFs in comparison to the other fingers with 3 DOFs for each. This gives the additional freedom which can be exploited to achieve motion that closely follows the target. The reaction time can be estimated to roughly 1.3 seconds (10 Hz of the error calculation loop) and is due to the low-pass filtering of the HPE results. The slow reaction time could be improved by increasing the rate of the HPE algorithm on the more efficient hardware. Otherwise, the additional tuning of the low-pass filter α parameter could be necessary.

It is visible that it takes approximately 10 iterations at the beginning of the trajectory tracking until the error value decreases. This is induced by the high values of the weight matrix for the initiatory fully straightened out hand, which has to be updated before the movement of the joint is possible. An improved algorithm of the dynamic weight matrix parameter calculation could improve the convergence speed.

B. Selection of α parameter

In order to select the optimal α parameter, the calculation of error over two trajectories with different parameter values was performed. The results are presented in Table I. All the results are the mean of 2 runs of each of the configurations. It can be seen from the presented table, that the optimal α parameter depends on the frequency of the HPE. For the lower HPE frequency, the higher value of α is preferable, which shifts the task descriptor position closer to the current HPE readings. It is in line with our expectations since this prevents adding to the response delay, which would be more severe for lower frequency tracking input.

trajectory	α = 0.1	α = 0.2	α = 0.3	α = 0.4
slow 5Hz	20.02	**17.54**	18.90	22.17
faster 5Hz	22.88	21.79	**20.14**	21.78
slow 2.5Hz	18.34	17.20	**16.62**	17.21
faster 2.5 Hz	23.69	23.32	19.60	**19.53**

TABLE I

RESULTS OF THE EVALUATION OF DIFFERENT VALUES FOR LOW-PASS FILTER α PARAMETER IN TWO EXEMPLARY TRAJECTORIES. THE VALUES REPRESENT MEAN TRACKING ERROR GIVEN IN MM OVER ALL JOINTS.

	error over all joints	error of finger tips
Task augmentation	**15.54**	0.4513
Task prioritization	19.38	**0.3699**
Single task	21.20	0.3987

TABLE II

EVALUATION OF DIFFERENT MULTI-TASK METHODS. TABLE PRESENTS MEAN TRACKING ERROR GIVEN IN MM TESTED ON THE SLOW TRAJECTORY.

C. Selection of task handling method

During the experiments 3 inverse kinematics task handling methods were evaluated taking into consideration the error over all the joints (coherence at trajectory mapping) and the finger tip alone (critical for the successful grasping performance):

1) Two task descriptors with task augmentation
2) Two task descriptors with task prioritization
3) Single task descriptor

The results are depicted in Fig. 8 and Fig. 9 and summarized in Table II. Task augmentation would be the optimal solution if we aim for minimal error over all the joints, but it does not allow as good tracking of the task descriptor on the finger tip as the other two methods. The task prioritization with the main task set to the finger tip outperforms both other methods. Averaging over both tasks in task augmentation leads to failure at satisfying both the tasks at the same time and leads to its inferior performance. This should be taken into consideration when selecting a multi-task handling method.

D. Qualitative results

During the experiments with grasping the smoothness of the trajectory tracking is essential. This gives the required stability when handling the object and induces a higher success rate at task execution. The lower α values which imply the stronger filtering were preferable. The success rate when attempting to grasp and raise a prepared object in the simulation was approximately 1 in 10. An example of the successful task execution is presented in Fig. 10.

The lack of haptic feedback and any method, which would assist the tracking to ensure a firm contact with the object's surface, still makes the successful grasping with the demanding objects difficult. Additional work for the improved contact with the objects can be dedicated to improving the grasping performance. One of the possible solutions would be the optimization with particle swarm optimization [28].

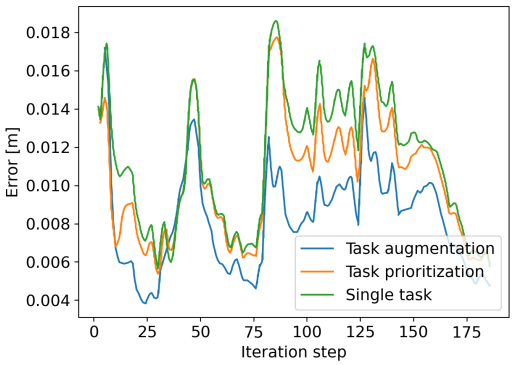

Fig. 8. Errors of mapping of index finger tip and 2 top joints (evaluation of the smoother trajectory). Task augmentation outperforms the other two methods, but task prioritization still outperforms single task handling, since it constraints all the DOFs of finger.

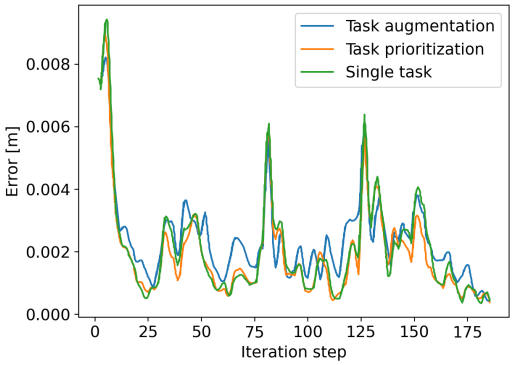

Fig. 9. Errors of mapping of index finger tip (evaluation of the smoother trajectory). Task prioritization performs on par with single task handling for finger tip control.

V. CONCLUSIONS

We have presented a framework for the motion retargeting of the complex human hand motion to the dexterous robotic hand from a single depth camera stream. The advantages of different multi-tasking handling methods have been presented. The platform has been used to successfully grasp an object in the simulation environment.

For the successful mapping of the trajectory, the low-pass filtering and the correspondence point adjustment were necessary. This allowed the calculation of the hand entirely in its local frame independent of the used simulation environment. The use of task prioritization allows the high accuracy of the finger tip tracking while lowering the error of the positions of otherwise unconstrained DOFs. The promising results encourage further research of the computer simulations as the possible environment to perform the conventionally very expensive autonomous learning with the robot models.

The findings can open the door to the low-cost fully autonomous learning process of dexterous manipulation also in a simulator. In future work, we would like to evaluate the captured demonstrations of dexterous hand manipulation tasks with a variety of learning algorithms.

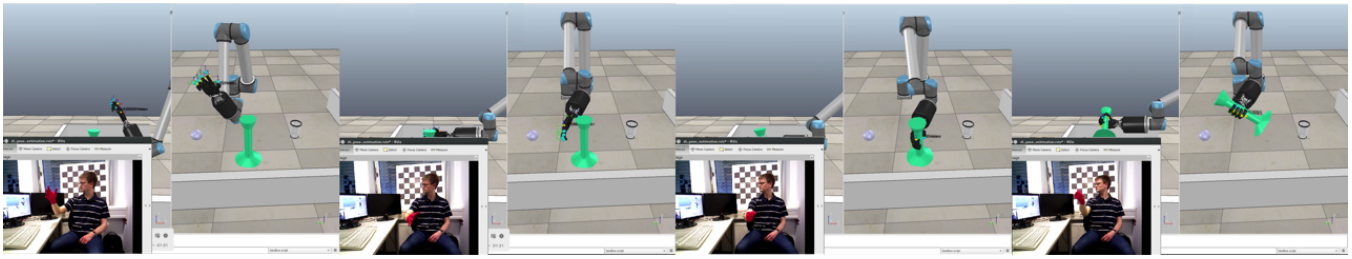

Fig. 10. Example of a successful grasping imitation in the simulation environment. The subject executing the task is visible in the lower left corner in each frame.

ACKNOWLEDGMENT

This research was partially supported by the Helmholtz Association.

REFERENCES

[1] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," in *Proceedings of International Conference on Computer Vision (ICCV)*, vol. 10, 2017.

[2] C. Choi, S. Ho Yoon, C.-N. Chen, and K. Ramani, "Robust hand pose estimation during the interaction with an unknown object," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3123–3132.

[3] L. Rozo Castañeda, P. Jimenez Schlegl, and C. Torras, "Sharpening haptic inputs for teaching a manipulation skill to a robot," in *1st IEEE International Conference on Applied Bionics and Biomechanics*, 2010, pp. 331–340.

[4] L. Rozo, P. Jiménez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden markov models," in *9th International Workshop on Robot Motion and Control*. IEEE, 2013, pp. 227–232.

[5] A. Pervez, A. Ali, J.-H. Ryu, and D. Lee, "Novel learning from demonstration approach for repetitive teleoperation tasks," in *2017 IEEE World Haptics Conference (WHC)*. IEEE, 2017, pp. 60–65.

[6] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," *arXiv preprint arXiv:1709.04905*, 2017.

[7] T. Zhang, Z. McCarthy, O. Jowl, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[8] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2169–2176.

[9] J. Ho and S. Ermon, "Generative adversarial imitation learning," *arXiv preprint arXiv:1606.03476*, 2016.

[10] S. Li and D. Lee, "Point-to-pose voting based hand pose estimation using residual permutation equivariant layer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 927–11 936.

[11] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, p. 169, 2014.

[12] M. Oberweger and V. Lepetit, "Deepprior++: Improving fast and accurate 3d hand pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 585–594.

[13] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool, "Tracking a hand manipulating an object," in *2009 12th International Conference on Computer Vision*. IEEE, 2009.

[14] 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*. IEEE, 2011, pp. 2088–2095.

[15] K. Ayusawa and E. Yoshida, "Motion Retargeting for Humanoid Robots Based on Simultaneous Morphing Parameter Identification and Motion Optimization," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1343–1357, Dec. 2017.

[16] V. Kumar and E. Todorov, "Mujoco haptix: A virtual reality system for hand manipulation," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 657–663.

[17] J. Rosell, R. Suarez, C. Rosales, J. A. Garcia, and A. Perez, "Motion planning for high DOF anthropomorphic hands," in *2009 IEEE International Conference on Robotics and Automation*. Kobe: IEEE, May 2009, pp. 4025–4030.

[18] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox, "DexPilot: Vision Based Teleoperation of Dexterous Robotic Hand-Arm System," *arXiv:1910.03135 [cs]*, Oct. 2019.

[19] S. Li, J. Jiang, P. Ruppel, H. Liang, X. Ma, N. Hendrich, F.-C. Sun, and J. Zhang, "A mobile robot hand-arm teleoperation system by vision and imu," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10 900–10 906, 2020.

[20] S. Chiaverini, G. Oriolo, and I. D. Walker, "Kinematically redundant manipulators," *Springer handbook of robotics*, pp. 245–268, 2008.

[21] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.

[22] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.

[23] M. Kouchia, M. Tadaa, N. Miyataa, and M. Mochimarua, "Evaluation of estimated hand measurements for creating digital hands," in *Proceedings 19th Triennial Congress of the IEA*, vol. 9, 2015, p. 14.

[24] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, "Whole body humanoid control from human motion descriptors," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2677–2684.

[25] J. Fajardo, A. Lemus, and E. Rohmer, "Galileo bionic hand: semg activated approaches for a multifunction upper-limb prosthetic," in *2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)*. IEEE, 2015, pp. 1–6.

[26] S. D. Han, N. M. Stiffler, A. Krontiris, K. E. Bekris, and J. Yu, "High-quality tabletop rearrangement with overhand grasps: Hardness results and fast methods," *arXiv preprint arXiv:1705.09180*, 2017.

[27] E. De Coninck, S. Bohez, S. Leroux, T. Verbelen, B. Vankeirsbilck, P. Simoens, and B. Dhoedt, "Dianne: a modular framework for designing, training and deploying deep neural networks on heterogeneous distributed infrastructure," *Journal of Systems and Software*, vol. 141, pp. 52–65, 2018.

[28] D. Antotsiou, G. Garcia-Hernando, and T.-K. Kim, "Task-oriented hand motion retargeting for dexterous manipulation imitation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.