Task Autocorrection for Immersive Teleoperation

Chenyang Wang[1], Simon Huber[2], Stelian Coros[2], Roi Poranne[3]

***Abstract*— Teleoperating robotic arms is a challenging task that requires years of training to master. It is mentally demanding, as the operator must internally compute transformations, or rely on muscle memory, to perform even the simplest tasks. Alternative methods that rely on *embodiment* – the immersive, first person experience of controlling the robot from its point of view are recently becoming more popular, thanks to the emergence of mixed reality devices. These methods create an intuitive experience by tracking the users motions, and *retargetting* them to the robot. However, even recent hardware fails at achieving total immersion, due to inherent discrepancies such as latency, imperfect tracking, and the differences between human and robot motor systems. Thus, performing even simple pick-and-place tasks with these systems, while more intuitive, is still cumbersome, and far from the level of human performance.**

In this paper we propose an immersive system that aims to bridge this gap. The system tracks the user's motion and retargets them to the robot as usual, but it also detects the user's intent, that is, the task they wish to perform. Based on this knowledge, the system can *autocorrect* the motion when it is about to fail, in a *seamless* manner, such that the task is successfully performed. We evaluate the efficacy of our autocorrection system in a user study. The results show a statistically significant performance improvement in terms of operation accuracy and time.

I. Introduction

Recent years saw major advancements in autonomous robotics, that greatly enhanced robot decision making capabilities. However, though AI-capable robots are no longer perceived as a science fiction pipe dream, neither are they an attainable goal in the near future. The transition to full autonomy will not happen suddenly, but rather gradually, through a coherent consolidation of human and robot intelligence. Recent successful approaches to teach robots to perform tasks are based on *imitation learning*, where an expert demonstrates a task, and a policy to perform this task is inferred from the demonstration. Gathering demonstration data can be done by manipulating the robot directly or by teleoperating, to perform the task, and recording the motion. This however, can be cumbersome and time consuming, without a proper interface.

Teleoperation of robotic arms by e.g. *jogging* is notoriously difficult to master, and to a novice, it can be mentally draining. Fully controlling a robotic hand, and not simply switching between preconfigured grasps, is virtually impossible. In recent years, due to the growing availability of mixed reality devices, more and more teleoperation systems rely on immersive, first person *embodiment* of the robot. We argue that with these approaches, generating demonstrations can be done much more efficiently. These systems attempt to create an intuitive experience by presenting the operator with a stereo video feed of robot's perspective via a Head Mounted Display (HMD) and by tracking the operator's motions, and *retargetting* them to the robot. Despite the many advances, a totally immersive experience will not be possible in the foreseeable future. This is due to issues such as latency and imperfect tracking, but also due to the inherent differences between human and robot motor systems. As is evident from demonstrations, even the highest-end systems, which include tactile feedback for example, are far from exhibiting human dexterity. The solution, we argue, lies in slight, ideally unobservable, modifications of the motions in a way that makes it easier to perform tasks, a process we term *task autocorrection*.

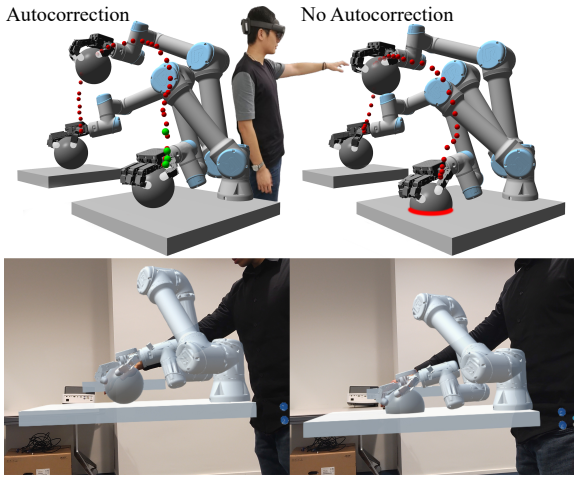

Fig. 1. An operator with HoloLens 2, teleoperating a robotic arm to place a ball on the desk. Without autocorrection the ball is placed through the desk, while with it, it is placed perfectly well. The red dots indicate the trajectory of user input. The green dots indicate the corrected input trajectory.

The effect of the autocorrection can be demonstrated in Fig 1 and the accompanying video; There, using an HMD with hand tracking capabilities, the user attempts to control a virtual robot which tracks their hand. They attempt to pick and place a virtual ball on the table. While without autocorrection, they fail to properly grasp the ball, and place it *through* the table, the autocorrection motion is valid. Inspired by the work of Dragan and Srinivasa [1], who presented the concept of *assistive* teleoperation, the task autocorrection framework comprises of two main components: a task predictor and a motion planner. The intent predictor is a neural network trained to predict the intent of

[1]Chenyang Wang is with the Dept. of Information Technology and Electrical Engineering, ETH, Zurich, Switzerland. wangche@ethz.ch;

[2]The authors are with the Department of Computer Science, ETH, Zurich, Switzerland. simon.huber@inf.ethz.ch; scoros@inf.ethz.ch

[3]Roi Poranne is with the Department of Computer Science, University of Haifa, Haifa, Israel. roiporanne@cs.haifa.ac.il

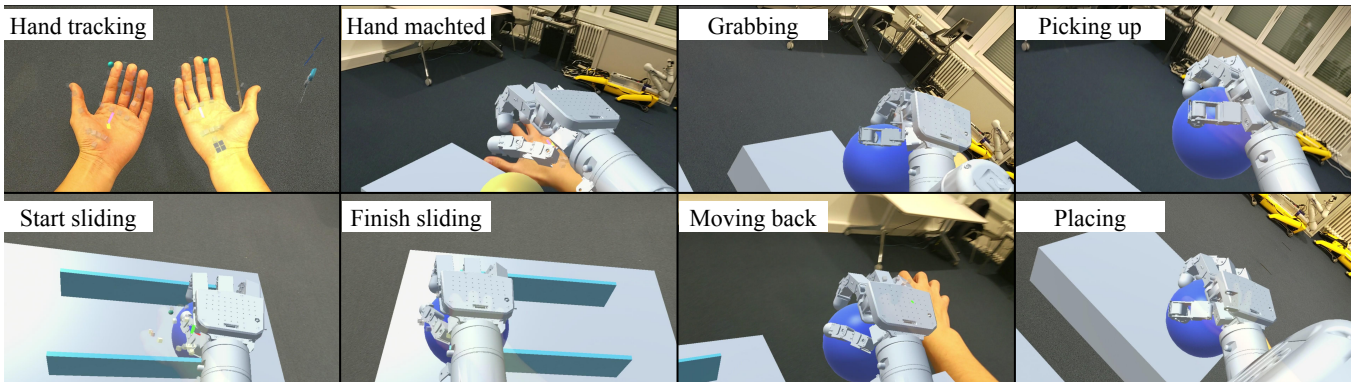

Fig. 2. Teleoperating a virtual robot in MR shown as a sequence of snapshots from left to right and top to bottom. After the participant matched the input hand with the robot hand, he grabbed the yellow ball and then slide it between the open slots. At last, the ball was placed back on the desk. (Once the ball is stably grasped, the ball would turn to blue color.)

the operator. The motion planner, based on the predicted task, computes the optimal robot motion, and blends the user's motion with the optimal one, based on the confidence level of the prediction. This blending is called *arbitration* in [1].

We present our approach for immersive teleoperation. Our contribution is threefold:

1) A holographic immersive teleoperation playground using Microsoft's HoloLens 2 (Fig 2).
2) An extendable autocorrection system for teleoperation.
3) A user study to evaluate the efficacy of autocorrection.

II. Related work

Immersive teleoperation is a topic that has been researched for years, but recently saw a resurgence due to an increasing reliability and availability of HMDs. Recent approaches combine of different hardwares as input device. We mention a few examples: Fritsche et al. [2] proposed a system composed of Microsoft Kinect, Oculus Rift and haptic *SensorGlove* to teleoperate the robotic arm of *iCub*. Similarly, [3] used a Vive VR headsets and hand controllers to teleoperate PR2 robot, with the same goal of obtaining demonstrations suitable for imitation learning. Liang et al. [4] used palm position from a Leap Motion to teleoperate a 6 DoF manipulator, and additionally visualized the 3D model of tracked hand in an HMD. In [5], an *augmented virtuality* system was created to teleoperate UR10 with a mounted Robotiq-85 gripper. The recent vision-based system, DexPilot [6], was used to teleoperate a robotic hand-arm system by observing human hand via 4 RGB-D cameras. In addition to tracking, a recent partnership called the Converge Robotics Group created the commercial Tactile Telerobot, a teleoperated hand with a tactile feedback glove, providing the user with an increased level of immersion. The hardware, however, is very expensive in comparison to the alternatives, and therefore not suitable for data collection from many users.

Central to the question of intuitive teleoperation is the question of motion *retargetting*, that is, how to *map* human motions to robots motions. At the most basic level, it works by solving an IK problem, where the target end-effector pose matches the user's hand pose. Rakita et al. [7] relaxed this mapping by considering additional goals, such as smooth motion and limited joint velocity. Other factors such as joint space discontinuities and self-collisions were addressed in [8]. Eric et al. [9] showed that Mixed Reality feedback outperformed physical feedback when the operator commands the robot via speech, pointing, and eye gaze to let it distinguish different objects. Finally, we mention that the problem has been of great interest to the computer graphics community, as of the the fundamental problems is how to transfer motions from one character to another (see e,g, the seminal work by Gleicher [10]).

Intent Prediction is the process of understanding the action that a subject is about to perform. It is closely related to the action recognition problem, which has been the topic of extensive research in the computer vision community. While the focus has been centered on third-person human action recognition (see e.g. [11] for a review), first-person is becoming more prominent [12, 13]. To collect data, [14] developed RoboTurk, a crowd-sourcing platform where users can manipulate a simulated robot in order to accomplish specific tasks, which is intended for imitation learning.

Assistive Teleoperation, as described in [1], is the process of arbitrating the operators motion and a learned optimal policy. Teleoperation and full autonomy are both extremely challenging, and assistive teleoperation seeks to bridge the gap and lie somewhere in between these two extremes. One of the first instances of assistive teleoperation was proposed by [15], in which a robot could turn cranks based on imprecise operator inputs. Since then, many different methods were described, and we sample just a few here [16, 17, 18, 19, 20, 21, 22].

As mentioned, we follow the approach introduced in [1], which consists of two main components: prediction of user intent and its arbitration with the user's input. In our work, we constantly predict the user's intent, and based on that intent, we compute an optimal trajectory to perform the action. We do not execute the optimal trajectory, however, as this would be too intrusive to the user. Instead we blend the user's retargetted motion and the optimal one together, informed by the confidence level of the intent prediction. Similar to [23], our system could also change to new context of correction if a new user intent is predicted.

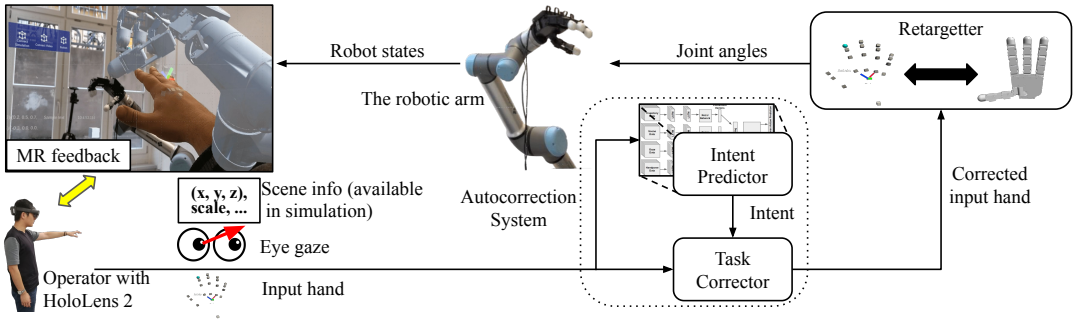

Fig. 3. System overview: Mixed Reality teleoperation framework and autocorrection system

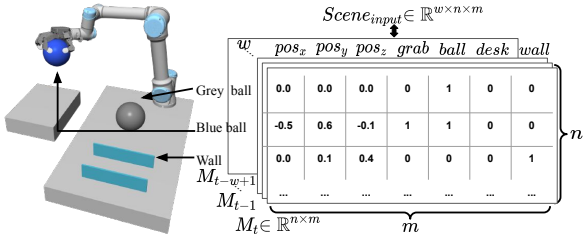

Fig. 4. An example of the scene: 6 objects (two balls, two walls and two desks) ($n = 6$). Each row of M represents an m dimensional state vector of object ($m = 7$). The first three elements indicate the position of the object. *grab* denotes whether the object is grabbed, and the last three elements are one-hot coded vectors representing the type of the object. The scene input is stacked by such w scene matrices M.

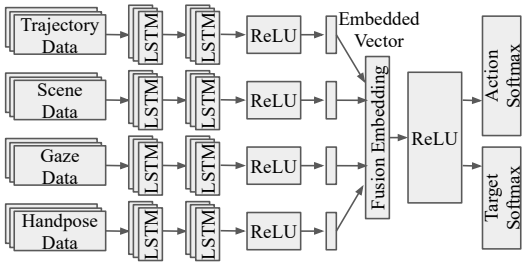

Fig. 5. The architecture of the multi-modal and multi-task action prediction neural network

III. Method

A system overview is shown in Fig 3. The teleoperation system consists of three parts: A data generating input device, a *retargetter* that maps the input data to joint angles, and the robot itself. Our robotic setup consists of a Universal Robots [24] UR5 robot arm with 6 *DOFs*, and a mounted Allegro Hand from Wonik Robotics with 4 fingers and 4 joints per finger (16 *DOFs*) [25]. A hologram of this setup is displayed using a HoloLens 2 HMD. The HoloLens 2 provides a 26-joint articular hand model and a gaze pointer (origin and direction). The hand model is used as data for the *retargetter*, which maps it to joint angles. We use a basic inverse kinematics model, where the operator's fingertips are mapped to the robot's fingertips. Formally:

$$q^* = argmin_q \sum_{A_i,P_i} \frac{1}{2} ||T(A_i,q) - P_i||^2 \qquad (1)$$

where q is the joint angles, A_i, P_i being the corresponding robot and human fingertips) and T the forward kinematics function. We solved this problem using Newton's method.

A. The Autocorrection System

Our autocorrection system consists of two parts. An *intent predictor* that estimates the user's intent and the *task corrector* that changes the data based on the predicted intent.

1) Intent Prediction: By working in a virtual environment, we can easily generate and label data, which is collected to train a multi-modal multi-task neural network to predict the user's intent. The input data of the network is the hand palm trajectory, the trajectories of the 26 hand joints, the gaze (origin and direction) trajectory and the stacked scene data tracked with the same window size as other trajectories. The scene object transformation M_t and scene input are organised as Fig 4. To distinguish between the action (e.g., grabbing) and the object (e.g., blue ball) we define intent as a tuple of action and target. Two soft-max outputs are added to the network to predict action and target respectively.

a) Network Architecture: Given the input and the output structure in Fig 5, the input is transformed into four equal-size embedded vectors by forward propagating along two stacked Long Short Term Memory (LSTM) layers and one followed ReLU network in each branch. The first LSTM layer in the scene branch iterates over the axis representing the number of objects n instead of the time axis w because it is designed to handle a varying amount of objects in the scene. The embedding vectors are fused together by learnable weighted averaging and then fed into a final ReLU network before computing the soft-max output. The output of action prediction branch is a d dimensional vector, which represents the probability for each action ($d = 4$, *Pick*, *Place*, *Slide-Between-Walls* and *None*). The target prediction output gives more details for the pick and place action: A value of 1 for the first entry indicates that the object will be picked up. A value of 1 for the second entry indicates that the operator will drop the object. An additional *None* entry is also added to indicates all the other cases. Thus, the dimension of the target prediction output is $2n_{max} + 1$, where $n_{max} = 8$ is the maximum number of the objects in the scene.

b) Training: The training dataset was generated by asking participants to accomplish one of two tasks: pick a ball and place it on a desk randomly or pick a ball and slide it between two walls. In the context of action prediction, the annotated labels are shifted forward by different time span, which represents how far into the future the robot could anticipate the beginning of the user's action. We gathered around 5 hours of training data with this method. The trained

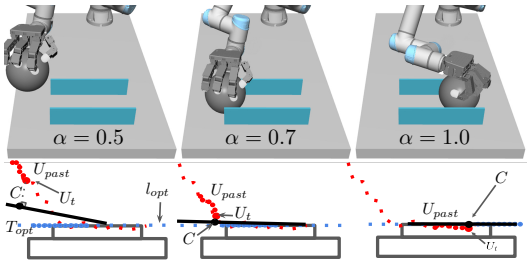

Fig. 6. Top: slide task with the final arbitration value α evolving over time based on confidence score. Bottom: abstract view that shows the user trajectory in red, the relevant points U_{past} for the line fitting in fat red, the optimal line l_{opt} in blue and the optimal trajectory T_{opt} in fat blue. U_t is the most recent input point and C the corrected input.

model achieves 89.48 % accuracy for action prediction and 85.02 % accuracy for target prediction on our test set.

2) Task Correction: For each action, we define a different set of physical requirements. For example, when placing a ball on a desk, it is required that the ball does not penetrate the desk. Sliding a ball through the thin slot between two walls, must be in a perfect linear motion. Considering the varying task difficulty, the aggressiveness of assistance also varies accordingly. The uncertainty given by the intent predictor complicates things even more, i.e. prediction confidence. To accommodate different physical requirements and the varying degree of assistance per task we propose a modular task correction framework with an arbitration value that allows to tweak the aggressiveness of the correction. To integrate the uncertainty into the correction framework we use the minimum between the two maximum probabilities from each soft-max layer of the intention prediction network as a confidence score.

a) Correction Framework: We employ a hierarchical system to be able to stack or switch out different elements. As an example, imagine at the top the sliding task stacked on a ball grasping task. For this specific example, the first corrector would keep the trajectory and the second one would correct for the placement of the fingertips. In our case, a controller selects the suitable correctors this way such that the system can have arbitrary complexity. Every corrector takes the user's input hand U, possibly already processed by another corrector, the result of the intent prediction, and the confidence score as input. It estimates the task-specific optimal trajectory T_{opt} and the arbitration value α that specifies the aggressiveness of the task correction and then interpolates T_{opt} with U based on α. The evolution of the arbitration value, the optimal trajectory and the hand trajectory can be seen in Fig 6.

b) Implementation of Correctors: We implemented three correctors: pick, place, and slide. The place corrector assists in achieving an optimal place action. The optimal trajectory is a target point P, i.e. $T_{opt} = \{P\}$, which is exactly r cm (radius of the ball) above the projection on the nearest desk. The arbitration value α, is a linear combination between the confidence of intent prediction network $p \in [0,1]$ and a regularizer $\sigma = 1 - 1/(1 + exp(-k(d - d_0)) \in [0,1]$, which depends on the estimated distance d between the ball center and the target point. In the experiments conducted the values were set as $k = 50$ and $d_0 = 0.5$. A point-wise linear interpolation method is applied here and the corrected user input $C = (c_x, c_y.c_z)$ is computed as:

$$C = \alpha P + (1 - \alpha)U \tag{2}$$

With increasing α, the system would correct increasingly aggressive towards the optimal position.

The slide corrector assists the operator to slide the ball in a straight line between the two walls, i.e. optimal line l_{opt} in Fig 6. Consequently, the optimal trajectory of the slide corrector consists of a series of optimal positions over the next a few timesteps, $T_{opt} = \{P_1, P_2, ..., P_n\}$, where $n = 20$ is the number of future timesteps (around 30 fps). The first position P_1 is the projection point on l_{opt} of the current user input hand position at time t. The followed points $\{P_2, ..., P_n\}$ are equally spaced on l_{opt}, where the spaced distance and direction are determined by the previous observations of user input $U_{past} = \{U_{t-p}, U_{t-p+1}, ..., U_t\}$. The value $p = 10$ determines the window size. The interpolation process is more complex for the slide corrector. It is designed by fitting a line l_{fit} via weighted PCA on the point set $U_{past} \cup T_{opt}$. The weight w_{past} assigned to U_{past} is 1, while the weight w_{opt} for T_{opt} depends on arbitration value α, where $w_{opt} = \gamma(1 + \alpha)$. The $\gamma \in [1, \infty]$ is a parameter to tune the weights between them. If the weight w_{opt} is larger than w_{past}, the correction is biased towards the optimal line l_{opt}. Finally, the corrected user input C can be determined by projecting U_t on l_{fit}.

Lastly, the ball grasp corrector assists the operator with performing a valid grasp of a ball. In our case, this is a grasp that accurately touches the ball, without penetrating it. The optimal trajectory T_{opt} is an hand pose H_{opt} that will grip the sphere entirely. They are computed by finding the nearest contact points on the sphere's surface corresponding to the input tip position of the users hand. The interpolation is to move the user input tip position to this optimal position H_{opt} once the ball is detected to be grasped. From the corrector-framework perspective, it is as a point-wise linear interpolation with arbitration value $\alpha = 1$.

IV. Evaluation

To evaluate the autocorrection approach, we designed a user study where participants were invited to teleoperate a life sized robot *hologram* in MR with or without the assistance of autocorrection (Fig. 7). The autocorrection system is equipped with aforementioned place corrector, slide corrector and ball grasp corrector. The simulation is done on a laptop with a *Nvidia GTX* 1060 graphic card, an Intel i7-8750H CPU and 16 GB of memory.

A. Tasks

Two tasks were targeted in the user study, namely, *Pick-And-Place* and *Slide*. The scene contained the robot, which was placed on a virtual desk at 1m high, two balls colored grey and yellow, both with a diameter of 16 cm, an additional "shelf" was place slightly above the table to provide an alternative surface. Additionally, two "walls", 8 cm of height and 16 cm apart were placed on the desk, to form a tight

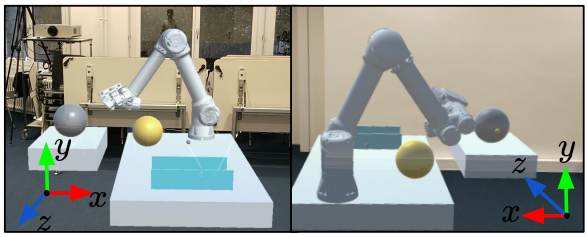

Fig. 7. The experimental scene as viewed in the HoloLens 2

slot for the balls. And they are upright and aligned with x direction as Fig 7.

In *Pick-And-Place* task, the participants were required to grab the grey ball first, and place on a spot marked by an indicator (shown as a small sphere) and then grab the yellow ball and place it next to another indicator. The participants were told to try to place the ball *properly*, that is, it should not placed in the air or through the surface of the desk. In *Slide* task, the requirement was to pick the grey ball first and drag along the slot from the right to the left, in the perspective of the robot, and then drag it back. This was repeated twice. The participants were told to keep the ball as low as possible without colliding with the desk or the walls.

B. Experimental Setup

1) Participants: We invited 11 participants to take part in the user study (two females, M = 24.73 years old, SD = 2.15). Five of them wore glasses and two of them had the experience of teleoperation in MR before. Each participated in 8 trials and all recorded data was used for evaluation.

2) Experimental Procedure: Each participant was asked to wear the HoloLens 2, perform the eye calibration routine and follow a short operation introduction. The instructions and goal for the study were then explained. Before starting the experiment itself, a warm-up trial was conducted to let the participants become familiar with basic operations. For each trial, the participants were asked to perform either a *Pick-And-Place* task or a *Slide* task once with or without enabling autocorrection. And the participant did not know whether autocorrection system was enabled or not. In total, there were four trials with autocorrection and four trials without autocorrection for each participant. The two tasks were also equally distributed among 8 trials, and the order of trials was random. In each trial, a separate program was running at the same time to evaluate and record the experiment. After each trial, the participant would have a break and complete a questionnaire survey.

C. Performance Evaluation

1) Quantitative Measure: The recorded experiments were evaluated using two quantitative metrics during simulation: *Violation Distance* (VD) and *Operation time*.

The violation distance is defined as the overlapped distance between two objects along specific direction. For each place action in *Pick-And-Place* task, it is computed as the vertical distance (y direction as Fig 7) between the ball center with its optimal position, which is 8 cm (radius) higher above the nearest desk surface. The score for one trial is the average

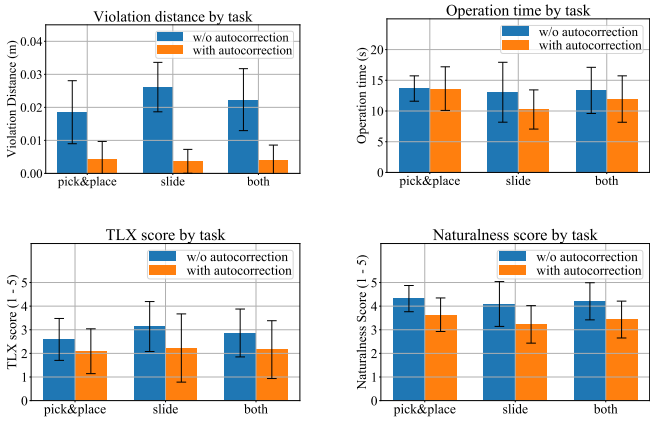

Fig. 8. Evaluation results: the height of each bar represents the mean value and standard deviation is indicated by the error bar on it. The mean and standard deviation are computed across trials for each type of task.

value among all place actions over time. For the *Slide* task, the violation distance is evaluated as the sum of overlapped distance between the ball and the wall and between the ball and the surface (z and y direction respectively). The score for one trial is the average value integrated over the entire slide motion.

The operation time is defined as the amount of time required to accomplish each task. For the *Pick-And-Place* task, the timing began as soon as the robot picked the grey ball up and ended when the yellow ball was released. For the *Slide* task, timing started when the ball entered the slot and ended when it left the slot after the repeated motion.

2) Questionnaire Survey: In addition to the quantitative evaluation during each trial, after the end of it, the participant was asked to answer two questions to evaluate the task load and the degree of *naturalness* during control for each trial, both of spanning score values of 1-5 in 1 point increments.

Question 1: Which level of the difficulty do you feel when you execute each teleoperation task?

A Task Load Index (TLX) was rated by participants for this question referring to the *Effort* rating of the NASA Task Load Index (NASA-TLX) [26], where we rescaled it from 1, meaning "Very easy", to 5 meaning "Very hard".

Question 2: Which level of the naturalness do you feel when you teleoperate the robot irrespective of the result of task?

Participants also need to answer this question by rating with *naturalness score*. It aims to subjectively measure the level of autonomy discrepancies [27], which indicates the user's sense of the intervention arising from the autocorrection system. It was scaled from 1 being "very unnatural", to 5 being "very natural".

D. Result

We considered hypotheses that, the autocorrected motion will outperform the non-autocorrected on *Pick-And-Place* task, *Slide* task or both tasks in terms of three metrics: lower violation distance, shorter operation time and lower TLX, but with a lower naturalness score. The mean and

TABLE I

MEAN AND STANDARD DEVIATION ACROSS TRIALS FOR RESULT WITH AND WITHOUT AUTOCORRECTION.

Task	*VD (cm)*	*Time (s)*	*TLX*	*Naturalness*
Pick&Place w/o	1.85±0.95	13.66±2.06	2.59±0.88	4.32±0.55
Pick&Place with	0.42±0.54	13.65±3.55	2.09±0.95	3.64±0.71
Slide w/o	2.61±0.75	13.05±4.88	3.14±1.06	4.09±0.95
Slide with	0.37±0.36	10.25±3.19	2.23±1.44	3.23±0.79
Both w/o	2.23±0.94	13.36±3.76	2.86±1.01	4.20±0.79
Both with	0.39±0.46	11.95±3.78	2.16±1.22	3.43±0.78

standard deviation of respective metric by task are summarized in Fig 8 and Table I by comparing in settings with or without autocorrection. To determine how the dependent variables, i.e. four metrics differ for the independent variable (with/without autocorrection) separately, four univariate ANOVA analysis are conducted for each task.

1) Pick-And-Place task: Firstly, for *Pick-And-Place* task the results revealed that, compared to the no autocorrection system, the system with autocorrection has statistically lower violation distance, $F(1,42) = 35.586$, $p < .0005$, partial $\eta^2 = .459$, whose mean value is decreased by 77.3%. However, they did not show statistically significant difference on operation time, $F(1,42) = .000$, $p = .989$, partial $\eta^2 = .000$. Moreover, regarding the results of questionnaire survey, the ANOVA analysis showed that the autocorrection system could decrease TLX rating with statistical support, $F(1,42) = 3.110$, $p = .085 < .1$, partial $\eta^2 = .069$, which dropped down by 19.3% in average. But it could also lead to lower naturalness score, $F(1,42) = 12.023$, $p = .001 < .005$, partial $\eta^2 = .223$ (decreased by 15.7% in average).

2) Slide task: Likewise, same ANOVA tests are applied for *Slide* task. The results indicated that the autocorrection system improved the quantitative measures significantly, i.e. lower violation distance ($F(1,42) = 153.508$, $p < .0005$, partial $\eta^2 = .785$) and shorter operation time ($F(1,42) = 4.873$, $p = .033 < .05$, partial $\eta^2 = .104$), which were decreased by 85.8% and 21.5% (2.8 s) respectively regarding the mean value. Moreover, for questionnaire survey, the ANOVA test revealed that participants rated the trials with the assistive autocorrection system with lower TLX rating ($F(1,42) = 5.419$, $p = .025 < 0.05$, partial $\eta^2 = .114$) and lower naturalness score ($F(1,42) = 10.231$, $p = .003 < .005$, partial $\eta^2 = .196$), which dropped down by 28.7% and 21.0% for each metric in average.

3) Both tasks: If we do not distinguish two tasks, the results would give more support on our hypotheses with larger sample size. Firstly, across all trials for both tasks, the results revealed that autocorrected motion has statistically lower violation distance, $F(1,86) = 132.813$, $p < .0005$, partial $\eta^2 = .607$. Its average value is decreased by 82.5%. They also showed statistically significant difference on operation time with shorter mean time by 10.6% ($F(1,86) = 3.011$, $p = .086 < 0.1$, partial $\eta^2 = .034$). Furthermore, the autocorrected motion was rated with lower TLX with statistical support, $F(1,86) = 8.456$, $p = .005 < .01$, partial $\eta^2 = .090$, which decreased by 24.5% in average. The overall naturalness score was also lower with autocorrection, $F(1,86) = 20.939$, $p < .0001$, partial $\eta^2 = .196$ (decreased by 18.3% in average).

V. DISCUSSION

The quantitative measures and questionnaire survey verified the feasibility and usability of our autocorrection system. Based on the performance across all trials of user study irrespective of the task type, it can be concluded that the autocorrection system is able to improve operational accuracy and save operation time with statistical support. In particular, the system can decrease the violation distance significantly (over 75% for either *Pick-And-Place* or *Slide* task). In term of user experience, the autocorrection can reduce the user's sense of challenge, but it does increase the sense of intervention. Naturally, the operator would be aware of stronger intervention from the autocorrection system if the user input is corrected more aggressively.

For a specific task the results show a similar trend regarding lower violation distance, lower subjective task difficulty and the sense of naturalness. They also show that autocorrection system is capable of decreasing the operation time for *Slide* task. However, it did not show a significant difference for *Pick-And-Place*. This may stem from the fact that only place corrector is helpful for *Pick-And-Place* task and the place action usually happens instantaneously, where the correction can not contribute a lot in term of time efficiency. Additionally, it can be noticed that there is a high standard deviation in Fig 8 for each task, which may partially result from the wrong prediction of the intent.

While our work shows promise for teleoperating a humanoid robot arm, there are many directions for future work. We intend to evaluate the performance of our system in the real world which would show the strengths of an autocorrection system even more. Next, the current intent predictor has the perfect location of all objects in the scene and it would be interesting to integrate additional methods to extract the scene matrix from the cameras that are observing surrounding. Furthermore, only several examples of domain-specific corrector are explored in this work and especially only two levels. We will add more levels to our hierarchy to start integrating more properties, e.g., the material of the objects we interact with. The strategy of arbitration needs to be studied in-depth to trade off better operation performance against more human autonomy. The optimal trajectory calculated in the correctors did not consider obstacles. Therefore, we plan to add an obstacle considering path planning algorithm.

ACKNOWLEDGMENT

We would like to thank Mugeeb Al-Rahman Hassan for the initial implementation of our algorithm, and to the volunteers who participated in the experiment. Chenyang Wang was supported by the China Scholarship Council. This work was supported in part by the Swiss National Science Foundation NCCR Digital Fabrication (Agreement 51NF40-182887).

REFERENCES

[1] A. Dragan and S. Srinivasa, "Formalizing assistive teleoperation," in *Proceedings of Robotics: Science and Systems*, July 2012.

[2] L. Fritsche, F. Unverzag, J. Peters, and R. Calandra, "First-person tele-operation of a humanoid robot," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 997–1002.

[3] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," 10 2017.

[4] C. Liang, C. Liu, X. Liu, L. Cheng, and C. Yang, "Robot teleoperation system based on mixed reality," in *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2019, pp. 384–389.

[5] D. Sun, A. Kiselev, Q. Liao, T. Stoyanov, and A. Loutfi, "A new mixed-reality-based teleoperation system for telepresence and maneuverability enhancement," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 1, pp. 55–67, 2020.

[6] 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," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9164–9170.

[7] D. Rakita, B. Mutlu, and M. Gleicher, "A motion retargeting method for effective mimicry-based teleoperation of robot arms," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, 2017, pp. 361–370.

[8] D. Rakita, B. Mutlu, and M. Gleicher, "Relaxedik: Real-time synthesis of accurate and feasible robot arm motion," 06 2018.

[9] E. Rosen, D. Whitney, M. Fishman, D. Ullman, and S. Tellex, "Mixed reality as a bidirectional communication interface for human-robot interaction," 09 2020.

[10] M. Gleicher, "Retargetting motion to new characters," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: Association for Computing Machinery, 1998, p. 33–42. [Online]. Available: https://doi.org/10.1145/280814.280820

[11] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *ArXiv*, vol. abs/1806.11230, 2018.

[12] S. Singh, C. Arora, and C. Jawahar, "First person action recognition using deep learned descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2620–2628.

[13] B. Tekin, F. Bogo, and M. Pollefeys, "H+o: Unified egocentric recognition of 3d hand-object poses and interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *CoRL*, 2018.

[15] R. Goertz, "Manipulators used for handling radioactive materials," human factors in technology, 1963.

[16] A. Fagg, M. Rosenstein, R. Platt, and R. Grupen, "Extracting user intent in mixed initiative teleoperator control," 09 2004.

[17] Y. Demiris and G. Hayes, "Imitation as a dual-route process featuring prediction and learning components: A biologically plausible computational model," 09 2020.

[18] E. You and K. Hauser, "Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input," 06 2011.

[19] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012, pp. 1–8.

[20] J. Kofman, Xianghai Wu, T. J. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1206–1219, 2005.

[21] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, "Assistive teleoperation of robot arms via automatic time-optimal mode switching," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2016, pp. 35–42.

[22] J. Crandall and M. Goodrich, "Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation," vol. 2, 02 2002, pp. 1290 – 1295 vol.2.

[23] Y. Duan, M. Andrychowicz, B. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," 03 2017.

[24] U. Robots. Ur5. [Online]. Available: https://www.universal-robots.com/de/produkte/ur5-roboter/

[25] W. Robotics. Allegro hand. [Online]. Available: http://www.wonikrobotics.com/Allegro-Hand.htm

[26] S. Hart and L. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Advances in psychology*, vol. 52, pp. 139–183, 1988.

[27] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, "Common metrics for human-robot interaction," vol. 2006, 03 2006, pp. 33–40.