Low-Cost Teleoperation with Haptic Feedback through Vision-based Tactile Sensors for Rigid and Soft Object Manipulation

Martina Lippi*[1], Michael C. Welle*[2], Maciej K. Wozniak[2], Andrea Gasparri[1], Danica Kragic[2]

***Abstract*— Haptic feedback is essential for humans to successfully perform complex and delicate manipulation tasks. A recent rise in tactile sensors has enabled robots to leverage the sense of touch and expand their capability drastically. However, many tasks still need human intervention/guidance. For this reason, we present a teleoperation framework designed to provide haptic feedback to human operators based on the data from camera-based tactile sensors mounted on the robot gripper. Partial autonomy is introduced to prevent slippage of grasped objects during task execution. Notably, we rely exclusively on low-cost off-the-shelf hardware to realize an affordable solution. We demonstrate the versatility of the framework on nine different objects ranging from rigid to soft and fragile ones, using three different operators on real hardware.**

I. Introduction

Tactile sensors are designed to replicate the sense of touch in machines, enabling robots to gather information about the physical properties of objects they come in contact with, including texture, shape, and softness. This tactile capability is especially critical when executing precise and delicate manipulations of fragile objects, such as handling food items or brittle materials. Among the various tactile sensor technologies [1], camera-based tactile sensors have recently gained considerable interest within the research community thanks to their high resolution, robustness, circuit simplicity, and low cost [2] compared to electronic ones. Such sensors generally comprise a deformable elastomer integrated with a camera positioned behind to capture high-resolution images of the deformation occurring during contact. To name a few applications, in-hand manipulation [3], volumetric mesh reconstruction [4], and manipulation of soft and brittle objects [5] have been accomplished in the literature using autonomous robots equipped with vision-based tactile sensors. However, in many real-world applications, human guidance remains indispensable to ensure successful outcomes. This necessity arises in scenarios such as recovering from unexpected failures, operating in highly unstructured environments, or recording example manipulations.

Motivated by the above observations, our study delves into a teleoperation scenario where we aim to combine human guidance skills with the tactile information gathered from vision-based tactile sensors mounted on the robot end effector. More specifically, we propose a novel Tactile-to-Haptic (T2H) algorithm where we translate the observed tactile information to vibrations received on the user's teleoperation controller. Although haptic feedback has been largely applied in the literature, e.g., see [6], [7], [8], to the best of our knowledge, there exist no readily deployable solutions for bringing camera-based tactile sensor data into affordable consumer-grade hardware that can be used to teleoperate a robotic manipulator. Our framework aims to fill this need by providing an easy-to-deploy way for teleoperating a Franka Panda manipulator, equipped with a gripper and two DIGIT tactile sensors [9], using an Oculus Quest 2 controller. The 6D velocities from the hand-held controller are mimicked by the robot end effector, while the trigger buttons are used to close and open the gripper. The tactile sensing is translated into the amplitude of the perceived vibration of the controller giving the user an intuitive feeling of if, and how strongly, an object is grasped. Furthermore, we equip the framework with a partial autonomy behavior where the robot automatically detects and prevents slippage once an object is being grasped. In detail, our contributions are:

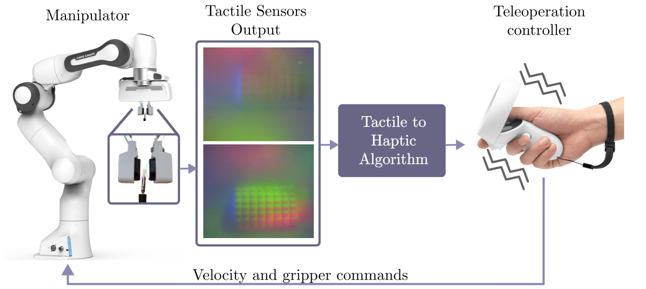

Fig. 1. Overview of the proposed T2H teleoperation framework: the data obtained with camera-based tactile sensors are elaborated by the T2H algorithm, the resulting haptic feedback is provided to the operator via the teleoperation controller, and the commands for the robot are generated.

- A novel T2H teleoperation framework deployable on low-cost hardware that translates tactile sensor readings into haptic vibration feedback, while providing partial autonomy for slippage prevention.
- Versatile demonstrations by three different operators for the use of such haptic feedback with and without partial autonomy for rigid and soft object manipulation.
- Public available source code and setup instructions[1].

II. Related Work

Remote robot control, often referred to as teleoperation is an active area of research and there are various approaches on how to do it most efficiently and accurately [10]. Xie et al. discussed the use of joysticks as a control interface [11].

*These authors contributed equally (listed in alphabetical order).

[1]Roma Tre University, Rome, Italy {*martina.lippi,andrea.gasparri*}@*uniroma3.it*

[2]KTH Royal Institute of Technology Stockholm, Sweden, {*mwelle,maciejw,dani*}@*kth.se*

[1] https://vision-tactile-manip.github.io/teleop/

Ishiguro et al. delved into the application of exoskeletons [12], while Gliesche's work compared keyboard and mouse with kinesthetic devices [13]. Nandikolla et al. showed that another approach to control the robots can be via brain-computer interfaces (BCI) as a means to command robots using users' brain signals, allowing them to interact with robots in an innovative way [14]. While these approaches offer advanced teleoperation, it is important to note that they often involve expensive equipment, and require specialized knowledge and training. Additionally, most of them are heavily tailored to specific robotic platforms.

On the other hand, Virtual, Augmented, and Mixed (VAM) technologies offer advanced and immersive applications that can be used to manipulate robots via position or velocity controllers, and can be easily adapted to different robotic platforms [15], [16], [17], [18], [19], [20]. Moreover, VAM frameworks have been introduced to facilitate human-robot collaboration by visualizing the states and intentions of robots in delivery tasks [21], [22]. These frameworks have found applications in industrial contexts as well [23], [24], along with interactive robot programming scenarios [25].

However, to the best of our knowledge, none of these VAM frameworks focuses on manipulating soft and fragile objects using haptic feedback. Our method uses a VAM framework controller to provide the user with a sense of touch recorded via tactile sensors. This sensorial capability comes at a low hardware cost while enhancing the ease of manipulation for the user, especially when handling fragile objects.

A. *Haptic feedback in human-robot interaction*

Haptic feedback plays an important role in achieving precision in robot manipulation. It encompasses various feedback options and controllers that contribute to improving the user's understanding of the robot's state and movement. Various feedbacks are classified as haptics. Vibrotactile feedbacks are vibrations that can provide sensory information about whether the robot is closer or further from an obstacle [26], [27] while squeezing bands can simulate the sensation of grasping objects [28]. Six DoF controllers apply resistance forces to guide the robot's movements, intervening, for instance, if the robot deviates from the intended path [29], [30]. However, those solutions are highly specialized and expensive. Additionally, the implementation of a *guiding force* as haptic feedback can sometimes lead to confusion and conflicts between the user and the robot, and finding the balance is an active area of research [31]. Moreover, haptic devices include gloves equipped with resistance plates to simulate the sensation of holding objects [32], wrist-worn haptic devices that offer controller-less haptic feedback for virtual and augmented reality experiences [33], and devices designed for simulating interactions with virtual objects, with haptic feedback concentrated on the wrist [34], [35].

While these applications focus on the haptic feedback to improve the precise manipulation, there are few works on using visual sensors (such as DIGIT [9], Minsight [36], or GelSight [37]) as a driver for the haptic feedback. Cao et al. [38] use visual sensors employed in the robotic arm to assess material properties, surface characteristics, roughness, and friction, and translate it to the haptic screen that reflects the properties of this material.

In our paper, we take a different approach. First of all, our solution uses VAM controllers and visual sensors that are commercially available, affordable, and not robot-specific. Next, we use these visual sensors to correlate the haptic feedback with how much force is applied to the object the robotic arm is grasping. This helps the operator to understand how much force is applied to the object, preventing unnecessary object compression that may cause damage to it, but also preventing *slippage* when not enough force is applied.

III. Problem statement and method overview

The primary objective of this study is to develop a teleoperation framework that leverages readily available and cost-effective hardware to manipulate objects that can possibly be delicate. Specifically, the framework should achieve the following goals: *i) cost-effective teleoperation,* i.e., realizing a system based on low-cost and commercial hardware components, enhancing its affordability; *ii) delicate object manipulation*, i.e., enabling the precise and gentle manipulation of fragile objects; *iii) user-friendly interface*, i.e., designing an intuitive interface for users, ensuring ease of use, and reducing the learning curve for operators of varying expertise levels; *iv) real-time control*, i.e., providing control and feedback mechanisms with no delay.

To achieve the above objectives, we propose the T2H framework depicted in Fig. 1. Specifically, we consider a robotic manipulator equipped with off-the-shelf vision-based tactile sensors on the gripper, and a controller able to provide vibration feedback to the user. In our setup, we employ the following commercial and low-cost devices: DIGIT tactile sensors [9] for the robot and an Oculus Quest 2 controller for teleoperation. The tactile sensor data are translated into vibration feedback based on low-computational image processing, allowing the user to perceive the sense of the applied pressure on the object under manipulation. Concurrently, the controller's linear and angular velocities are transformed into velocity commands for the robot's end effector, providing a reactive and intuitive interface for the users. In addition, we design a partial autonomy behavior to proactively prevent object slippage during manipulation. In practice, this entails the automatic tightening of the grasp whenever autonomous slippage is detected, ensuring a secure hold on the object. The synergy between the haptic feedback and the autonomous behavior empowers users with the ability to manipulate delicate objects with ease (as shown in Sec V), reducing the risk of excessive compression of these objects. Note that the proposed framework can be adapted to any vision-based tactile sensor and teleoperation controller with vibration feedback.

IV. T2H Framework

The proposed T2H framework comprises the following key modules which are detailed in the rest of the section: *i)* mapping from tactile sensors to teleoperation controller, *ii)*

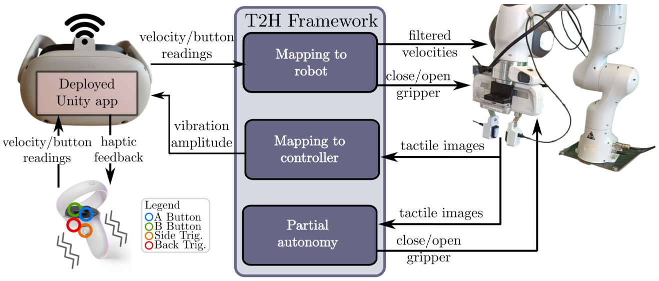

Fig. 2. Communication architecture for the T2H teleoperation framework.

mapping from teleoperation controller to robot commands, and *iii)* partial autonomy for slippage prevention.

The communication architecture, along with the information flow, is illustrated in Fig. 2 and is fully integrated in ROS middleware. We leverage the Unity Robotics toolbox[2] which enables communication between Unity and ROS. Specifically, it allows us to create a custom Unity application that is deployed on the Oculus headset and is capable of sending/receiving information to/from ROS. In this way, haptic feedback can be provided to the teleoperation controller, while also allowing commands from the controller to be forwarded to the robot. In the proposed framework, the operator has a direct view of the operating environment and is not required to wear the headset. However, the development of an additional graphical user interface (GUI) to further augment human perception is planned for future work.

A. Mapping from tactile sensors to haptic feedback

To determine the haptic feedback given the tactile sensors' images, we expand upon the processing methodology introduced in our prior work [5]. The basic idea is to compute the pixel-wise variation for each sensor image with respect to a background reference image, which can be set by the operator at any time, and then relate the vibration feedback with the count of altered pixels. In general, as the grasping force intensifies, the proportion of pixels differing from the background image also increases, resulting in amplified haptic feedback that should be conveyed to the operator.

We refer to the pixel-wise variation in each sensor as variation image and compute it as reported in Algorithm 1. First, we evaluate the pixel-wise difference between the current sensor image and the background as $D^s_k = |T^s_k - R^s|$, (line 1) where T^s_k is the RGB image of the tactile sensor s, with $s \in \{1,2\}$, at time step k, R^s is the RGB reference image of the sensor, and D^s_t is the resulting RGB difference image. Then, the three RGB channels are averaged to produce a one-dimensional difference image $\bar{D}^s_k$ (line 2), which undergoes a thresholding operation to reduce noise in the tactile sensor data. More precisely, we set pixel values to 0 if their intensity falls below an automatically tuned noise threshold $\eta^s \in [0,1]$, and to 1 otherwise. In this way, a binary difference image B^s_k is generated (line 3) which captures the significant differences between the current tactile image and the reference image. Details about the computation of the background reference image and the tuning of threshold noise are provided later in the section. For a more reliable (but slower) touch detection process, the variation image V^s_k is obtained with the element-wise (or Hadamard) product of c consecutive binary difference images (lines 4-7), i.e.,

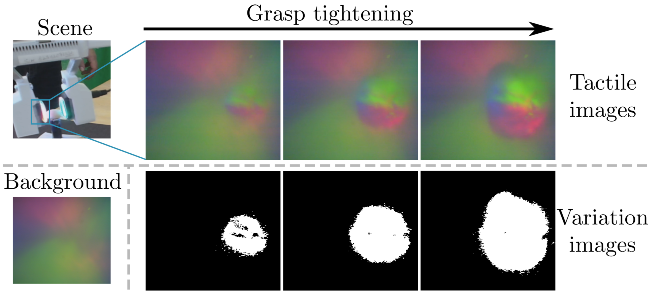

Fig. 3. Representation of the tactile images (first row) with respective variation images (second row) as the grasp is tightened on a pistachio nut. The scene and the background are shown on the left.

$$V^s_k = B^s_k \odot B^s_{k-1} \dots \odot B^s_{k-c+1},$$

with $\odot$ denoting the Hadamard product, i.e., to classify a pixel as a variation from the reference image, it must exhibit this variation for a minimum of c consecutive frames. The percentage of variation pixels, i.e., equal to 1, is finally computed and normalized in the range $[0,1]$ (line 8). Figure 3 shows the tactile images (first row) and the respective variation images (second row) obtained as the grasp is tightened on a pistachio nut. The background is reported on the left.

Algorithm 1 Tactile variation detection

```
Require: $T^s_k, [B^s_{k-c+1}, \dots, B^s_{k-1}], R^s, \eta^s$
1: $D^s_k = |T^s_k - R^s|$
2: $\bar{D}^s_k$ =average channels($D^s_k$)
3: $B^s_k$ = thresholding($\bar{D}^s_k$, $\eta^s$)
4: $V^s_k = B^s_k$
5: for $i \in \{k-c+1, \dots, k\}$ do:
6:     $V^s_k = V^s_k \odot B^s_i$
7: end for
8: $p^s_k$ =percentage variation pixel($V^s_k$)/100
9: return $p^s_k, B^s_k$
```

Based on the above, our T2H algorithm, reported in Algorithm 2, generates the vibration feedback that is provided to the user. More specifically, at each time step k, the tactile images are retrieved (line 3), and the variation ratios p^s_k are computed, with $s = 1,2$ (lines 4-5). At this point, the haptic feedback f_k, representing the controller vibration intensity, is determined with the following logarithmic function (line 7)

$$f_k = \log_{10}(1 + \alpha p_k) / \log_{10}(1 + \alpha),$$

where p_k is the average variation ratio (line 6) and α is a positive constant playing a critical role in the feedback behavior as shown in Fig. 4: as α increases, the haptic feedback becomes more sensitive to small changes in p_k, leading to a steeper incline for low values of p_k. Conversely, as α decreases, a flatter slope for low values of p_k is obtained. Finally, the haptic feedback is sent to the teleoperation

[2] https://github.com/Unity-Technologies/Unity-Robotics-Hub

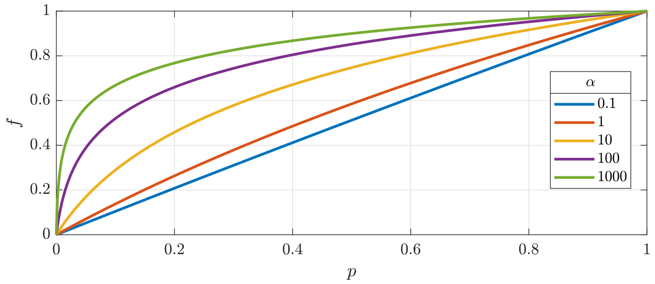

Fig. 4. Representation of the haptic feedback with different gains α.

controller (line 8) and the lists of previous binary difference images are updated (lines 9-10).

```
Algorithm 2 Tactile to Haptic (T2H) mapping
Require: $R^s, \eta^s$ with $s = 1,2$, $\alpha, c$
 1: $L^1 = \emptyset, L^2 = \emptyset$
 2: for time step k do
 3:     $T_k^1, T_k^2$ =collect tactile images
 4:     $p_k^1, B_k^1$ =tactile variation detection($T_k^1, L^1, R^1, \eta^1$)
 5:     $p_k^2, B_k^2$ =tactile variation detection($T_k^2, L^2, R^2, \eta^2$)
 6:     $p_k = (p_k^1 + p_k^2)/2$
 7:     $f_k = \log_{10}(1 + \alpha p_k)/\log_{10}(1 + \alpha)$
 8:     send haptic feedback($f_k$)
 9:     $L^1$ =add image to fixed-size list($B_k^1, c$)
10:     $L^2$ =add image to fixed-size list($B_k^2, c$)
11: end for
```

To establish the reference images R^s and automatically set the noise thresholds η^s with $s = 1,2$, we ask the operator to press the B button (marked in green in Fig. 2) before starting the manipulation (i.e., in no contact condition). This preliminary step triggers the collection of K consecutive tactile images for each sensor: the simple average of the first N images forms the reference image, while the average maximum value across the one-dimensional difference images $\bar{D}_k^s$ establishes the noise threshold, i.e., $\eta^s = \frac{1}{K}\sum_{i=k}^{k+K} \max(\bar{D}_i^s)$.

B. Mapping from teleoperation controller to robot

The purpose of the module for mapping from the teleoperation controller to the robot is to replicate with the robot the movements executed by the operator. This translation of commands persists as long as the A button remains pressed, i.e., the robot receives zero-velocity commands in the case the button is not pressed. When the button A is pressed, both the linear and rotational velocities of the teleoperation controller are mapped into inputs for the low-level Cartesian controller of the manipulator. For safety reasons, a sliding window averaging filter with length w is additionally applied to smooth the velocity inputs, and each component of the linear velocity is bounded within the range $[-v_{max}^l, v_{max}^l]$, with v_{max}^l a positive constant. Similarly, the angular velocity components are restricted to the range $[-v_{max}^r, v_{max}^r]$, with v_{max}^r a positive constant. Moreover, velocity components within the range of $[-v_{min}^l, v_{min}^l]$ for linear motion and $[-v_{min}^r, v_{min}^r]$ for rotational motion, with v_{min}^l and v_{min}^r positive constants, are filtered out and are set to zero. This ensures that the controller does not respond to minor velocity changes. To command the gripper, the back trigger (in red in Fig. 2) is employed to close the gripper at a constant velocity of v^g, while the side trigger (in orange in Fig. 2) is used to open the gripper similarly. Note that the A button's function to enable/disable the robot motion serves as a critical safety feature, preventing unintended movement by the operator. Furthermore, it grants the operators the flexibility to adjust their hand position should they approach the limits of their workspace. Once repositioned, they can resume providing velocity commands to the robot (see accompanying video).

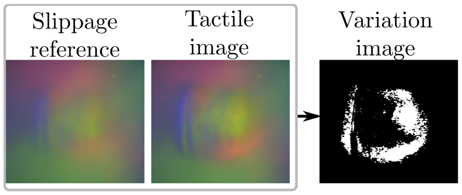

Fig. 5. Example of the slippage reference image (left), tactile data (middle), and respective variation image (right) during a pistachio nut manipulation.

C. Partial autonomy for slippage prevention

The partial autonomy behavior assists the operator in preventing object slippage. Briefly, in the event of automatic slip detection, the robot takes autonomous action to secure the grasp and prevent the object from slipping out of its grip. To realize the slip detection, we adopt a similar approach to the one described earlier for computing variation images. More specifically, when touch is detected on both sensors at time step k, i.e., $p_k^1 > \varepsilon_t$ and $p_k^2 > \varepsilon_t$, with ε_t a positive constant, the autonomous behavior, reported in Algorithm 3 is activated and an additional reference image $\bar{R}^s$ for each sensor s, referred to as slippage reference image, is computed by averaging N consecutive tactile images (lines 3-4). Note that this reference image does not depict the background; instead, it represents the tactile data of the object currently being grasped as shown in Fig. 5. When the autonomous behavior is active, we calculate at each time step k the variation ratios, denoted as $\bar{p}_k^s, \forall s$, with respect to the slippage reference images. This calculation is performed in accordance with Algorithm 1 using the slippage reference images $\bar{R}^s$ (lines 6-7). A slippage event is detected if either $\bar{p}_k^1 > \zeta^s$ or $\bar{p}_k^2 > \zeta^s$ is observed, where ζ^s, $\forall s$, is a time-varying threshold updated as follows (lines 5 and 14):

$$\zeta^s = 2^{\#slip}\varepsilon_t,$$

with $\#slip$ representing the number of slippages detected since the autonomous behavior activation.

If slippage is detected, the grasp is automatically tightened by a displacement δ_s (line 9). After tightening, new slippage reference images are acquired (lines 10-11), and the thresholds ζ^s are increased (line 12). The reason for increasing the threshold is to dynamically adjust the sensitivity of slip detection over time [5]. As the object is grasped more tightly, a higher tolerance for slip detection is necessary. Without this adaptation, the system might continually detect slippage for minor changes, hindering the successful manipulation of the object. When the gripper is opened, i.e., the side trigger is pressed, the slippage prevention is deactivated.

Algorithm 3 Partial autonomy

```
Require: $T_k^s, \bar{R}^s, \bar{L}^s = [B_{k-c+1}^s, ..., B_{k-1}^s], \eta^s$ with $s = 1, 2$, $\varepsilon_t, \delta_s, N$, initialize
if initialize then
    $\bar{R}^1$ = compute reference image($N$)
    $\bar{R}^2$ = compute reference image($N$)
    $\#slip = 0$
    $\zeta^1 = \varepsilon_t$, $\zeta^2 = \varepsilon_t$
end if
$\bar{p}_k^1, \bar{B}_k^1$ =tactile variation detection($T_k^1, \bar{L}^1, \bar{R}^1, \eta^1$)
$\bar{p}_k^2, \bar{B}_k^2$ =tactile variation detection($T_k^2, \bar{L}^2, \bar{R}^2, \eta^2$)
if $\bar{p}_k^1 > \zeta^1$ or $\bar{p}_k^2 > \zeta^2$ then
    tighten grasp($\delta_s$)
    $\bar{R}^1$ = compute reference image($N$)
    $\bar{R}^2$ = compute reference image($N$)
    $\#slip = \#slip + 1$
    $\zeta^s = 2^{\#slip} \varepsilon_t, \forall s$
end if
```

V. Experimental results

The aim of the experiments is to teleoperate the robot in order to manipulate several objects, moving them from their initial positions to a bowl, as illustrated in Fig. 6. An additional obstacle object (a yellow mustard bottle) is introduced into the scene and must be navigated around. The objects under consideration consist of soft food items, that are lime, plum, grape, and tomato, as well as more rigid items, that are AUX connector, Tetra Pak box, gel plastic bottle, plastic cup, and pistachio nut. The selected objects have distinct sizes, shapes, textures, and material properties, such as brittleness, presenting varied challenges for manipulation. The initial configurations of all objects are shown in Fig. 7. It is worth highlighting that in the initial configuration, the AUX connector is fully inserted in the respective port, requiring a certain force to be exerted for the initial unplugging phase. Similarly, the grapes and tomatoes are attached to the respective stems, requiring delicate manipulation to pick the fruit berry and detach it from the stem. While detaching, it is crucial to exert enough force to successfully separate the fruit, but it is equally important not to grasp too forcefully in order to prevent any damage to the delicate object. All the remaining objects do not require any unplugging/detaching operation before transportation. Three distinct operators participated in the experiments, one of whom was an experienced operator (specifically, an author of the paper). For each object, the corresponding manipulation task was executed three times by the experienced operator and twice by an external operator without prior experience in the task. This setup allows us to assess the usability of the T2H framework, even with individuals who may not be as familiar with it. Additionally, each manipulation task was executed both with and without enabling the partial autonomy behavior to validate its effectiveness, resulting in a total of 90 experiments. All experimental videos, code, and setup instructions are available on the project website[1]. Furthermore, the accompanying video reports examples of manipulation for all objects.

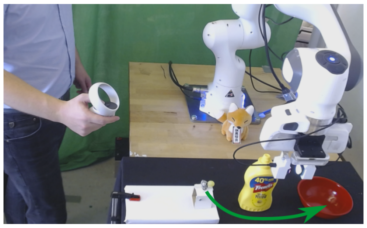

Fig. 6. Experimental setup for T2H framework validation. The desired motion is highlighted with an arrow.

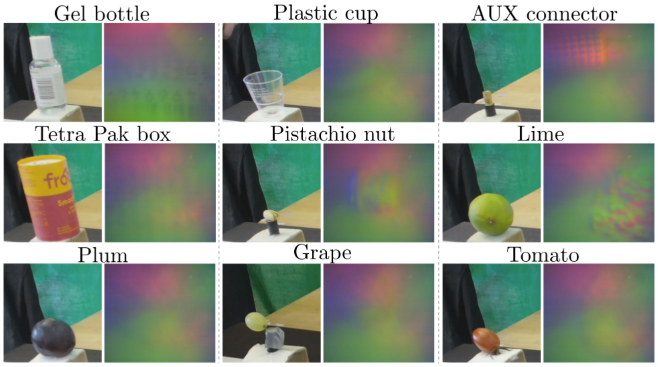

Fig. 7. Set of objects along with the respective tactile data.

Implementation details: The parameters of the T2H framework for the experiments were selected as follows. For the mapping to teleoperation controller and partial autonomy, we used gain $\alpha = 1000$, generated the reference images using a frame count of $N = 10$, tuned the automatic noise threshold over $K = 100$ frames, set the touch threshold to $\varepsilon_t = 0.01$, and considered $c = 2$ consecutive frames for variation detection. For the mapping to robot inputs, the maximum linear velocity was set to $v_{max}^l = 0.1$ m/s, and the maximum rotational velocity to 1 rad/s, while we disregarded linear velocities below 0.005 m/s and rotational velocities below 0.05 rad/s. Furthermore, we used $w = 5$ for the velocity averaging filter. The closing gripper speed was set to $v_g = 0.005$ m/s, while the displacement in the event of slippage was $\delta_s = 0.001$ m.

Results: As demonstrated in the experiment videos available on the project website, all three operators were able to successfully complete all teleoperation tasks with various objects, including delicate ones, without causing any damage. Table I summarizes the results obtained on the manipulation of all objects with (white cells) and without (grey cells) partial autonomy (PA) for slippage prevention enabled. Specifically, we report the duration, the minimum gripper opening, and the number of slippages (when PA is enabled only) achieved by the experienced (left) and non-experienced (right) operators. Average and standard deviation values are shown. Among the considered metrics, the minimum gripper opening serves as an indicator of object compression, with a larger opening implying less compression applied. Note that this metric is an absolute value and is not normalized relative to the (unknown) object width. Therefore, it must be compared on a per-object basis. For the experienced user,

TABLE I

RESULTS OBTAINED BY THE EXPERIENCED (LEFT) AND NON-EXPERIENCED (RIGHT) USERS. DURATION, MINIMUM GRIPPER OPENING, AND #*slip* WITH (GREY CELLS) AND WITHOUT (WHITE CELLS) PA ENABLED ARE REPORTED (#*slip* IS ONLY RECORDED WITH PA).

	Experienced user						Non-experienced users					
	Duration [s]		Min. opening [cm]		#*slip*		Duration [s]		Min. opening [cm]		#*slip*	
Object	✗ PA	✓ PA	✗ PA	✓ PA	✗ PA	✓ PA	✗ PA	✓ PA	✗ PA	✓ PA	✗ PA	✓ PA
Bottle	29.4±2.1	28.1±5.2	3.16±0.52	2.64±0.51	*N.A.*	2.7±0.9	49.0±7.6	43.3±4.3	3.24±0.33	2.27±0.03	*N.A.*	1.0±0.0
Cup	31.4±3.1	30.8±3.6	3.96±0.09	4.04±0.14	*N.A.*	1.7±0.5	47.4±5.7	36.4±0.5	3.75±0.13	4.25±0.13	*N.A.*	1.5±0.5
AUX	41.3±7.6	36.9±3.5	0.90±0.05	0.89±0.00	*N.A.*	5.0±0.8	39.8±11.8	44.0±9.3	0.72±0.03	0.83±0.05	*N.A.*	6.5±1.5
Box	33.0±2.5	26.6±1.4	5.04±0.34	5.19±0.06	*N.A.*	1.0±0.0	28.8±5.9	40.5±0.4	5.09±0.09	4.61±0.58	*N.A.*	1.0±0.0
Nut	35.4±1.8	26.1±2.3	1.53±0.22	1.53±0.34	*N.A.*	2.7±1.2	52.7±13.2	50.7±3.1	1.07±0.17	1.44±0.20	*N.A.*	2.5±0.5
Lime	28.4±3.0	25.5±3.0	4.95±0.09	4.83±0.14	*N.A.*	1.0±0.0	38.3±0.1	45.1±0.4	4.44±0.02	4.75±0.00	*N.A.*	1.0±0.0
Plum	27.6±2.3	27.7±4.3	4.59±0.20	4.33±0.19	*N.A.*	1.0±0.0	43.4±1.0	37.7±0.8	3.78±0.08	3.98±0.02	*N.A.*	1.0±0.0
Grape	44.3±3.4	38.0±3.2	1.68±0.11	1.75±0.18	*N.A.*	1.0±0.0	102.5±22.1	33.7±2.0	1.30±0.00	1.56±0.09	*N.A.*	1.0±0.0
Tomato	33.8±5.2	31.8±1.0	2.73±0.28	3.20±0.10	*N.A.*	1.0±0.0	60.0±29.3	35.1±1.6	2.48±0.65	2.56±0.15	*N.A.*	1.0±0.0

we can observe that the AUX connector and the grape berry require the longest manipulation times due to the challenging initial unplugging/detaching operations and the delicate nature of the grape. Across all objects, the autonomous slippage prevention behavior consistently contributes to the reduction of the time to complete the manipulation task. This is particularly evident for the AUX connector, pistachio nut, Tetra Pak box, and grape berry, with the AUX connector exhibiting the highest frequency of slippage prevention activation (i.e., showing the highest value of #*slip*). This is motivated by the requirement of applying a certain amount of force to unplug the connector, and insufficient force can lead to slippage. In such instances, the autonomous slippage detection and grasp tightening mechanisms significantly enhance the efficiency of task completion. In addition, we can observe that the PA behavior also leads to a reduction in object compression. This is evident in the fact that, on average, the minimum gripper opening required to complete the task is either lower or comparable. As far as the non-experienced users are concerned, similar trends to those observed for the experienced user are obtained, only resulting in average longer completion times and higher object compression. These results highlight the usability of the T2H framework as well as the benefits conferred by the slippage prevention autonomous behavior.

Figure 8 reports two illustrative examples of the haptic feedback f obtained by the experienced operator during the manipulation of the grape berry, both with (bottom) and without (top) enabling the partial autonomy behavior. Relevant screenshots depicting the manipulation process are included. In particular, in the absence of partial autonomy, the operator initiates the grasping of the berry at approximately $t = 24$ s, and a respective increase in haptic feedback amplitude is observed. However, as an attempt is made to move upwards, the berry slips from the grasp (at approximately $t = 27$ s), as indicated by the drop in the feedback signal. Subsequently, the operator makes a second attempt to secure the berry (at about $t = 33$ s), ultimately completing the manipulation (as shown in the screenshot at $t = 38$ s). When the partial autonomy is active, the operator performs the grasping of the berry at about $t = 24$ s, resulting in a corresponding increase in haptic feedback. However, in this case, automatic slippage detection occurs at approximately $t = 24.8$ s, triggering an autonomous tightening of the grasp. This intervention effectively prevents any slippage of the grape berry and enables the operator to complete the manipulation task (as shown in the screenshot at $t = 30$ s) without having to re-grasp the grape berry. Further examples are available in the accompanying video and on the project website.

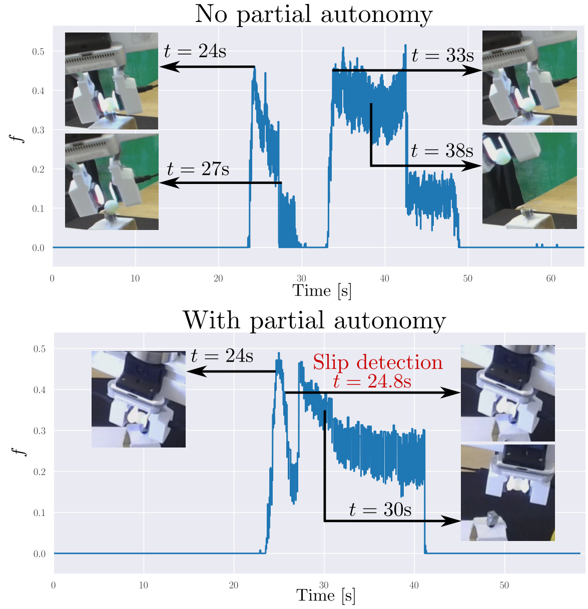

Fig. 8. Haptic feedback obtained during the grape manipulation by the expert user without (top) and with (bottom) partial autonomy enabled.

VI. CONCLUSION

In this study, we proposed the novel T2H framework, providing haptic feedback based on vision-based tactile sensors for teleoperation. We realized T2H with low-cost consumer-grade hardware and made the source code public. The framework leverages pixel-wise variation in tactile images compared to background reference images to provide vibration feedback, enhancing operator perception and control during object manipulation. Additionally, a partial autonomy feature is designed to prevent slippage during manipulation tasks. The system's effectiveness was demonstrated across a range of objects with diverse physical properties, including fragile items, without causing damage. Usability was confirmed across operators with varying experience levels. Future work aims to introduce proactive behaviors by predicting human intentions, integrate a GUI with the headset and conduct extensive user studies involving multiple operators.

References

[1] A. Yamaguchi and C. G. Atkeson, "Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision?," *Advanced Robotics*, vol. 33, no. 14, pp. 661–673, 2019.

[2] S. Zhang, Z. Chen, Y. Gao, W. Wan, J. Shan, H. Xue, F. Sun, Y. Yang, and B. Fang, "Hardware technology of vision-based tactile sensor: A review," *IEEE Sensors Journal*, vol. 22, no. 22, pp. 21410–21427, 2022.

[3] A. Yamaguchi and C. G. Atkeson, "Tactile behaviors with the vision-based tactile sensor fingervision," *International Journal of Humanoid Robotics*, vol. 16, no. 03, p. 1940002, 2019.

[4] X. Zhu, S. Jain, M. Tomizuka, and J. Van Baar, "Learning to synthesize volumetric meshes from vision-based tactile imprints," in *International Conference on Robotics and Automation (ICRA)*, pp. 4833–4839, 2022.

[5] M. C. Welle, M. Lippi, H. Lu, J. Lundell, A. Gasparri, and D. Kragic, "Enabling robot manipulation of soft and rigid objects with vision-based tactile sensors," in *International Conference on Automation Science and Engineering (CASE)*, 2023.

[6] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, 2023.

[7] I. El Rassi and J.-M. El Rassi, "A review of haptic feedback in tele-operated robotic surgery," *Journal of medical engineering & technology*, vol. 44, no. 5, pp. 247–254, 2020.

[8] A. Bolopion and S. Régnier, "A review of haptic feedback teleoperation systems for micromanipulation and microassembly," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 496–502, 2013.

[9] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, *et al.*, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.

[10] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, 2023.

[11] Y. Xie, X. Hou, and S. Wang, "Design of a novel haptic joystick for the teleoperation of continuum-mechanism-based medical robots," *Robotics*, vol. 12, no. 2, p. 52, 2023.

[12] Y. Ishiguro, T. Makabe, Y. Nagamatsu, Y. Kojio, K. Kojima, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit tablis," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6419–6426, 2020.

[13] P. Gliesche, T. Krick, M. Pfingsthorn, S. Drolshagen, C. Kowalski, and A. Hein, "Kinesthetic device vs. keyboard/mouse: a comparison in home care telemanipulation," *Frontiers in Robotics and AI*, vol. 7, p. 561015, 2020.

[14] V. Nandikolla and D. A. Medina Portilla, "Teleoperation robot control of a hybrid eeg-based bci arm manipulator using ros," *Journal of Robotics*, vol. 2022, 2022.

[15] M. Wozniak, C. T. Chang, M. B. Luebbers, B. Ikeda, M. Walker, E. Rosen, and T. R. Groechel, "Virtual, augmented, and mixed reality for human-robot interaction (vam-hri)," in *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 938–940, 2023.

[16] C. Barentine, A. McNay, R. Pfaffenbichler, A. Smith, E. Rosen, and E. Phillips, "A vr teleoperation suite with manipulation assist," in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 442–446, 2021.

[17] S. Xu, S. Moore, and A. Cosgun, "Shared-control robotic manipulation in virtual reality," *arXiv preprint arXiv:2205.10564*, 2022.

[18] M. Moletta, M. K. Wozniak, M. C. Welle, and D. Kragic, "A virtual reality framework for human-robot collaboration in cloth folding," *arXiv preprint arXiv:2305.07493*, 2023.

[19] F. Kennel-Maushart, R. Poranne, and S. Coros, "Multi-arm payload manipulation via mixed reality," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 11251–11257, IEEE, 2022.

[20] M. K. Wozniak, R. Stower, P. Jensfelt, and A. Pereira, "What you see is (not) what you get: A vr framework for correcting robot errors," in *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 243–247, 2023.

[21] K. Chandan, V. Kudalkar, X. Li, and S. Zhang, "Arroch: Augmented reality for robots collaborating with a human," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3787–3793, IEEE, 2021.

[22] M. K. Wozniak, R. Stower, P. Jensfelt, and A. Pereira, "Happily error after: Framework development and user study for correcting robot perception errors in virtual reality," *arXiv preprint arXiv:2306.14589*, 2023.

[23] Q. Wang, Y. Cheng, W. Jiao, M. T. Johnson, and Y. Zhang, "Virtual reality human-robot collaborative welding: a case study of weaving gas tungsten arc welding," *Journal of Manufacturing Processes*, vol. 48, pp. 210–217, 2019.

[24] J. Zubizarreta, I. Aguinaga, and A. Amundarain, "A framework for augmented reality guidance in industry," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 9, pp. 4095–4108, 2019.

[25] M. Ostanin and A. Klimchik, "Interactive robot programing using mixed reality," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 50–55, 2018.

[26] M. Lorenzini, S. Ciotti, J. M. Gandarias, S. Fani, M. Bianchi, and A. Ajoudani, "Performance analysis of vibrotactile and slide-and-squeeze haptic feedback devices for limbs postural adjustment," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 707–713, IEEE, 2022.

[27] Z. Chua and A. M. Okamura, "A modular 3-degrees-of-freedom force sensor for robot-assisted minimally invasive surgery research," *Sensors*, vol. 23, no. 11, p. 5230, 2023.

[28] A. Smith, B. Ward-Cherrier, A. Etoundi, and M. J. Pearson, "Feeling the pressure: The influence of vibrotactile patterns on feedback perception.," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 634–640, IEEE, 2022.

[29] R. Luz, A. Pereira, J. Corujeira, T. Krueger, J. Beck, E. den Exter, T. Chupin, J. L. Silva, and R. Ventura, "Feeling the slope? teleoperation of a mobile robot using a 7dof haptic device with attitude feedback,"

[30] R. Pocius, N. Zamani, H. Culbertson, and S. Nikolaidis, "Communicating robot goals via haptic feedback in manipulation tasks," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 591–593, 2020.

[31] Z. Al-Saadi, Y. M. Hamad, Y. Aydin, A. Kucukyilmaz, and C. Basdogan, "Resolving conflicts during human-robot co-manipulation," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 243–251, 2023.

[32] R. Hinchet, V. Vechev, H. Shea, and O. Hilliges, "Dextres: Wearable haptic feedback for grasping in vr via a thin form-factor electrostatic brake," in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pp. 901–912, 2018.

[33] J. E. Palmer, M. Sarac, A. A. Garza, and A. M. Okamura, "Haptic feedback relocation from the fingertips to the wrist for two-finger manipulation in virtual reality," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 628–633, IEEE, 2022.

[34] M. Coffey and A. Pierson, "Collaborative teleoperation with haptic feedback for collision-free navigation of ground robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8141–8148, IEEE, 2022.

[35] A. A. Stanley and K. J. Kuchenbecker, "Evaluation of tactile feedback methods for wrist rotation guidance," *IEEE Transactions on Haptics*, vol. 5, no. 3, pp. 240–251, 2012.

[36] I. Andrussow, H. Sun, K. J. Kuchenbecker, and G. Martius, "Minsight: A fingertip-sized vision-based tactile sensor for robotic manipulation," *Advanced Intelligent Systems*, p. 2300042, 2023.

[37] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[38] G. Cao, J. Jiang, N. Mao, D. Bollegala, M. Li, and S. Luo, "Vis2hap: Vision-based haptic rendering by cross-modal generation," *arXiv preprint arXiv:2301.06826*, 2023.