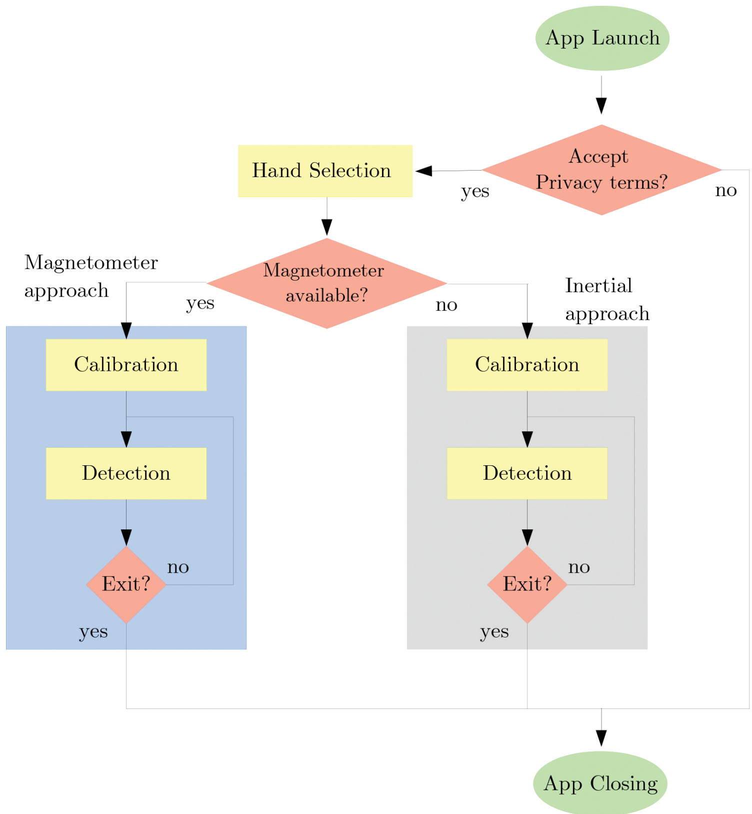

Figure 4.5: No Face-Touch application workflow.

Algorithm 2). After a calibration procedure (see Subsect. 4.1.3), the application starts the main loop exploiting the desired algorithm. The software continuously monitors hand proximity to face and, if necessary, alerts the user. The application runs also in background, until the *"EXIT"* button is pressed.

Algorithms

In this subsection we report the pseudo-code implementation of the algorithms detailed in Subsect. 4.1.3.

Algorithm 1 Detection with magnetometer.

Initialization:

$\theta = 0, \phi = 0, \Phi = 0,\ \bar{\Phi} = 0,\ \hat{\Phi} = 0,\ \alpha = 0,\ N = 200,\ M = 50,\ buffer = \emptyset,\ alert = False$

Calibration:

Phase 1: ▷ arm far from the magnets

```
    while time ≤ 2seconds do
        [m_x m_y m_z] ←read magnetometer
        Φ ← sqrt(m_x^2 + m_y^2 + m_z^2)
        if i < N then
            buffer.appendLast(Φ)
        else
            buffer.removeFirst()
            buffer.appendLast(Φ)
        end if
        i ← i + 1
    end while
```

$\bar{\Phi} \leftarrow \frac{1}{N}\sum_{i=1}^{N} buffer(i)$

$\sigma_\Phi \leftarrow \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(buffer(i) - \bar{\Phi})^2}$

Phase 2: ▷ move the watch closer to magnets

```
    while time ≤ 3seconds do
        [m_x m_y m_z] ←read magnetometer
        Φ ← sqrt(m_x^2 + m_y^2 + m_z^2)
        if Φ > Φ̂  then
            Φ̂ ← Φ
        end if
    end while
```

$\alpha \leftarrow (|\hat{\Phi} - \bar{\Phi}|)/\sigma_\Phi$

Monitoring:

loop

- $[a_x\ a_y\ a_z] \leftarrow$read accelerometer
- $\theta \leftarrow \text{arctan2}(a_y, a_z) \cdot 180/\pi$
- $\phi \leftarrow \text{arctan2}(-a_x, \sqrt{a_y^2 + a_z^2}) \cdot 180/\pi$
- $[m_x\ m_y\ m_z] \leftarrow$read magnetometer
- $\Phi \leftarrow \sqrt{m_x^2 + m_y^2 + m_z^2}$
- **if** $\theta_{min} < \theta < \theta_{max}$ and $\phi_{min} < \phi < \phi_{max}$ **then**
    - $safeOrientation \leftarrow False$
- **else**
    - $safeOrientation \leftarrow True$
    - *buffer*.removeFirst()
    - *buffer*.appendLast(Φ)
    - $\bar{\Phi} \leftarrow \frac{1}{N}\sum_{i=N-M}^{N}$ *buffer(i)*
    - $\sigma_\Phi \leftarrow \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(buffer(i) - \bar{\Phi})^2}$
- **end if**
- **if** !*safeOrientation* and $(|\Phi - \bar{\Phi}|/\sigma_\Phi) > \alpha$ **then**
    - $alert \leftarrow True$
- **else**
    - $alert \leftarrow False$
- **end if**

end loop

Algorithm 2 Detection without magnetometer.

Initialization:
$\theta = 0$, $\phi = 0$, $\dot{\phi} = 0$, $\beta = 0$, $N = 200$, $M = 50$, $i = 0$, $buffer = \emptyset$, $slope = \emptyset$, $rising = False$, $safeOrientation = True$, $alert = False$

```
Calibration:                                          ▷ arm still during calibration
  while time ≤ 2seconds do
    [a_x a_y a_z] ←read accelerometer
    θ ← arctan2(a_x, a_y) · 180/π
    φ ← arctan2(−a_x, √(a_x² + a_y²)) · 180/π
    φ̇ ←compute φ̇
    if i < N then
      buffer.appendLast(φ̇)
    else
      buffer.removeFirst()
      buffer.appendLast(φ̇)
    end if
    i ← i + 1
  end while
```

$\bar{\dot{\phi}} \leftarrow \frac{1}{N}\sum_{i=1}^{N} buffer(i)$

$\sigma_{\dot{\phi}} \leftarrow \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(buffer(i) - \bar{\dot{\phi}})^2}$

$\beta \leftarrow 3 \cdot \sigma_{\dot{\phi}}$

```
Monitoring:
  loop
    [a_x a_y a_z] ←read accelerometer
    θ ← arctan2(a_y, a_z) · 180/π
    φ ← arctan2(−a_x, √(a_y² + a_z²)) · 180/π
    φ̇ ←compute φ̇
    slope.deleteFirst()
    if φ̇ > β then
      slope.appendLast(1)
    else
      slope.appendLast(-1)
    end if
    slope_bar ← (1/N) Σ_{i=N−M}^{N} slope(i)
    if slope_bar > 0 then
      rising = True
    else
      rising = False
    end if
    if θ_min < θ < θ_max and φ_min < φ < φ_max then
      safeOrientation ← False
    else
      safeOrientation ← True
    end if
    if !safeOrientation and rising then
      alert ← True
    else
      alert ← False
    end if
  end loop
```

4.2 Wristband Display and Haptic Feedback for Mobile Augmented Reality

In the context of Mobile Augmented Reality (MAR), hand-based interaction with virtual environment is currently very limited. Apart from interaction mediated by the touch-screen, the user cannot manipulate the virtual environment with her/his own hands. In fact, virtual objects projected on the smart-phone screen are on a different layer than the camera-captured images, hence the user's hand cannot get in contact with virtual entities. This section presents a system designed to overcome the aforementioned issues, that is especially aimed to promote the use of Augmented Reality in daily life (*e.g.* industrial scenarios). The interaction of the real hand with virtual entities is enabled by the tracking of the hand pose and by representing it in the virtual environment through a virtual avatar, so that interactions can be seen (through smartphone screen) and measured. The finger-worn haptic interface displays the interaction forces through tactile stimuli on the fingertip to augment the user's perception abilities. For the purpose of deploying the AR system in daily life scenario, we focused on minimizing the encumbrance on the user's hands. We designed a forearm support to house the smartphone during the task, so that the user can keep both hands free, except for the ring used for tracking and haptic rendering.

This section presents the design of the first prototype and its further improvement, that promotes the wearability through a compact wireless haptic interface. The capability and usability of the refined system were investigated in a user study, through a comparison of our approach with the standard hand-held montage. Quantitative and qualitative results show that our system was positively received by the subjects.

4.2.1 Motivation

Augmented Reality (AR) enriches the real world with virtual objects, animations, and sounds. Even if AR systems have been developed for decades, only the most recent devices are bringing AR applications into our everyday life. Such fast growth suggests that Augmented Reality will become an essential technology in the future. For instance, the Industry 4.0 initiatives leverage Augmented Reality to facilitate training of technicians, ease product assembly [154], and lower cognitive load and error rate during task execution [155]. Despite its potentiality, AR is still not fully adopted for daily use. Many AR apps lack the functionality to justify the burden of wearing a Head-Mounted Display (HMD) or hand-holding a smartphone for a prolonged time. A system capable of providing *on-demand* support may prove more versatile for daily activities, by removing the major sources of concern that operators face when wearing HMD, such as isolation and loss of

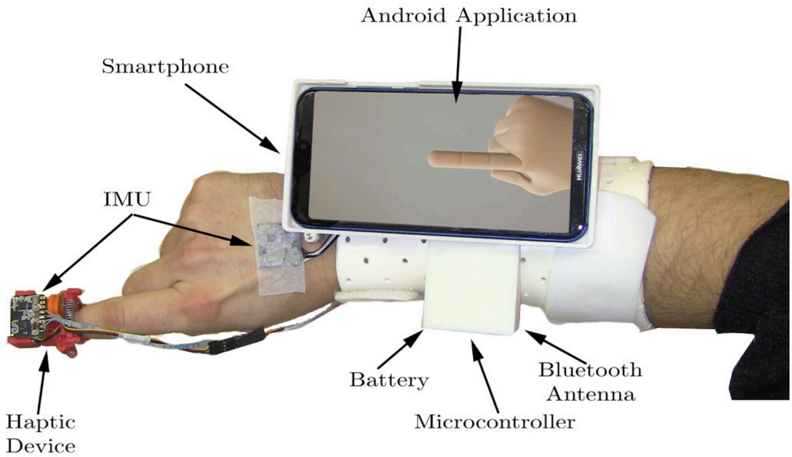

Figure 4.6: First prototype. An IMU-based system tracks the motion of the index finger while the haptic thimble provides the user with cutaneous stimuli. The proposed device creates the sensation of making/breaking contact with virtual objects. An Arduino microcontroller, powered by a Li-Po battery, is in charge of collecting data from the sensors and control the haptic device. The smart-phone shows the augmented environment to the user, computes the hand posture, and evaluates the fingertip contact force.

situational awareness. Between the approaches published in literature, Qian *et al.* presented a hand-held MAR system based on a smart-phone and an external portable processor to track the hand in real-time using a Leap Motion, obtaining good results in terms of user experience [156]. Although the user's hand is not constrained by wearing hardware, this solution cannot provide haptic feedback and always occupies one hand to hold the phone. Minseok *et al.* used a similar setup for developing a hand-held system based on touch and hand gestures [157]. The use of Leap Motion required a computer to detect gestures and forward them to the phone. Other strategies leveraged on using instrumented gloves to retrieve the hand pose from hand joint angles and provide vibrotactile feedback [158], or tracking the index finger using the smart-phone rear camera [159].

Conversely, our approach minimizes the required hardware by exploiting a smartphone for visual input and processing, and a fingertip interface for tracking and haptic rendering. In order to manipulate virtual entities, the user's hand

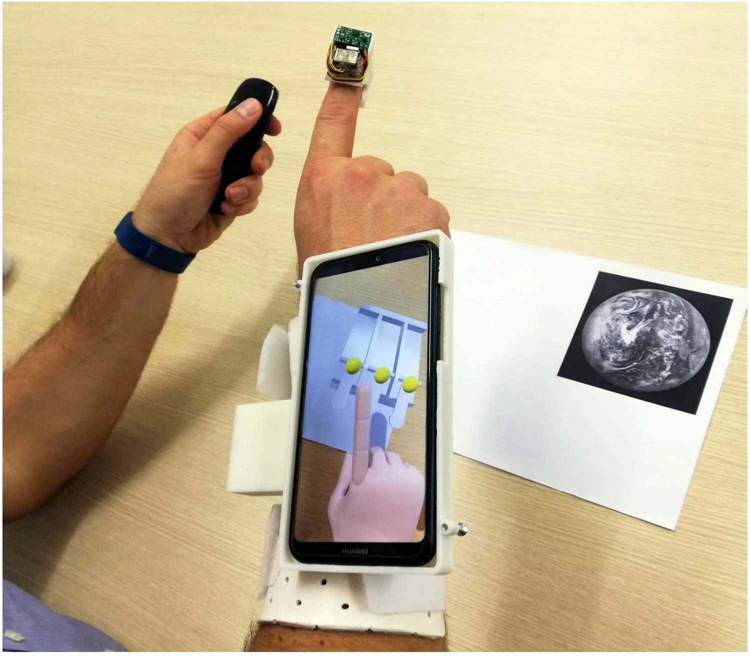

Figure 4.7: Second prototype. The haptic device worn on the fingertip handles tracking and haptic rendering. It it connected to the smartphone, placed on the wrist-support, using BLE communication. In this picture the left hand is free and holds a remote controller.

has to collide with the objects mesh, displayed on the phone screen. Instead of capturing the hand pose with a camera approach, we decided to leverage inertial measurements to reconstruct the hand posture. This way, we avoid the limitation of always having the real hand in the camera field of view, and we free the other hand from the burden of holding the device. The problem of locating the hand in the 3D space is addressed by positioning the phone on a wrist-mounted support. In fact, the ARcore library used to develop the app estimates the distance of the fiducial markers from the rear camera, that is translated of a fixed amount from the hand. Thus, the finger pose estimated with inertial measures and its position in the 3D space are integrated on a virtual avatar projected on the screen, which can interact with virtual objects.

The two versions of the prototypes we developed for testing consist of two wearable pieces of equipment: a display located on the forearm and a small fingertip interfaces to virtually map the finger and to generate haptic cues (see

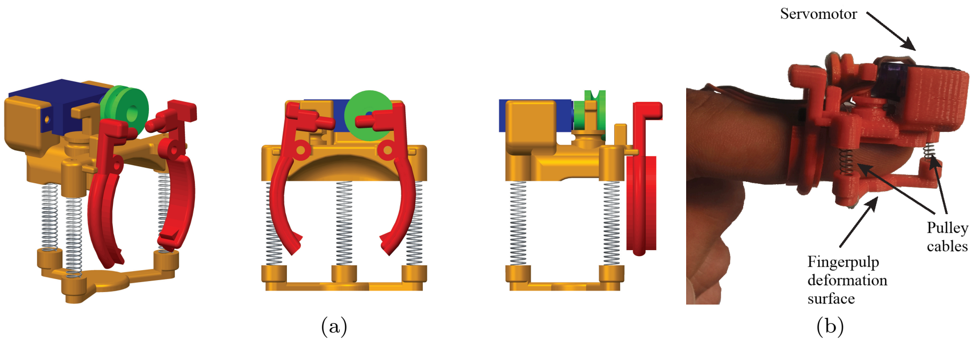

Figure 4.8: Haptic display used in the first prototype system. The CAD model is depicted in (a). The servo-motor exploits a pulley mechanism with three threads to move the flat surface and exert forces on the fingerpulp, producing the sensation of making/breaking contact with virtual objects. The haptic interface worn on the fingertip is shown in (b). During experimental trials, the IMU is attached on top of the servomotor for finger tracking.

Fig. 4.6 and Fig. 4.7). The final implementation of our system will feature a flexible display worn on the wrist to minimize encumbrance, and small fingertip haptic interfaces that have minimum impact on the user's manipulation capabilities.

With the proposed system we aim at addressing the following limitation: *(i)* keeping at least one hand free, *(ii)* enhancing the experience with haptic rendering, *(iii)* providing an on-demand AR experience, and *(iv)* designing a low-budget wearable system based on smart-phone.

We conducted an experimental evaluation to compare our approach with the standard hand-held montage. The functionality and usability of the new setup were tested in experimental trials involving 18 people, where the haptic feedback provided by the fingertip interface was essential for the task completion. Participants were asked to explore an augmented reality setting and solve a puzzle relying on visual and tactile information. Qualitative and quantitative indices were evaluated in two conditions to test the efficacy of the system condensed on a single arm, with respect to the standard hand-held montage. Results are presented as performance measurements retrieved during the experiments and users' personal evaluations from an *ad hoc* questionnaire.

4.2.2 System overview

In this subsection, we present a novel, cheap, and wearable setup that combines a visual display for Augmented Reality and an acting part, capable of rendering forces as cutaneous feedback. The system was tested under two different layouts: the hand-held (HH) and the wrist-mounted (WM) version, respectively.

Our efforts were channeled in a twofold manner: *i)* designing the hardware necessary to collect the input data and display visual and tactile cues; *ii)* developing an Android software to collect measurements, process data, and calculate the appropriate feedback. The smartphone is the central processing unit, but also serves as visual input source. It is secured to the arm using a 3D-printed ABS support attached to a thermoplastic splint. The finger avatar reproduces the index finger pose estimated by an Inertial and Magnetic Unit (IMU), while the interaction forces are rendered through a tactile display on the fingertip (Fig. 4.9).

Visual Display

Initially, we imagined to use a flexible LCD display, unfortunately such technology is not mature enough to be commercially available and ready for developers. Thus, we decided to use an off-the-shelf Android smartphone and build an *ad-hoc* forearm support, shown in Fig. 4.11a. We designed the support to hold the phone at a fixed distance from the wrist (for the hand tracking) avoiding motion constrains of the hand and camera occlusion. Three thermoplastic splints have been designed to accomodate different body sizes. Splint and support are secured through screws to have a firm grip. The support can house the smart-phone in two layouts, landscape and portrait, rotated 90° with respect to each other. The support has been designed to make sure that the camera field of view was not blocked by the user's hand or forearm, by introducing an offset between the position of the real and virtual hand. The splint is fastened to the arm using two Velcro strips. The smartphone app is displayed in panoramic layout, and was active for the entire duration of the trial. The support was used only in the wrist-mounted (WM) condition, while the smartphone was hand-held in the corresponding condition (HH).

First Prototype

The first prototype we developed is composed of a wristband display and a wearable fingertip interface (see Fig. 4.6). A smart-phone (P20 lite, Huawei Technologies Co. Ltd., CN) is the central processing unit, but also serves as visual input source and visual display. It is secured to the arm using a 3D-printed ABS support attached to a thermoplastic splint. The fingertip interface is equipped with: i) an Inertial and Magnetic Unit (MPU6050, Invesense Corp. US) to estimate the

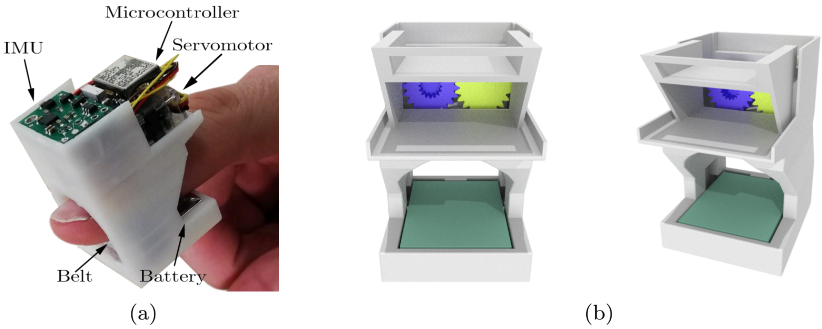

Figure 4.9: The wearable haptic device is composed of a small servomotor and two gears that move a flexible belt towards and away from the finger pulp. A micro controller and a Bluetooth antenna are in charge of controlling and connecting the device, respectively. A 100 mA h Li-Po battery, placed at the bottom of the case, powers the system. The total weight of thimble is 20.8 g. A user wearing the device is depicted in (a), while the rendered 3-D models from two different points of view are reported in (b).

index finger orientatio with respect to the back of the hand, and ii) a servomotor (HS-35HD Ultra Nano, HITEC Inc. USA) to render interaction forces through a pulley mechanism. A clip system enables users to easily fasten the device on the finger. The servomotor can provide a maximum torque of $0.8\,\mathrm{kg\,cm^{-1}}$. The cutaneous force is generated by the device considering the force generated by the servomotor and the resistance given by the three springs and the human skin (see Fig. 4.8). Interested reader is referred to [98] for further details on the force feedback generation.

On top of the servomotor is attached an IMU sensor board that contains a triaxial accelerometer/gyroscope and an I^2C interface. To estimate the relative motion of the finger with respect to the hand, we placed an additional IMU on the back of the hand. We decided to use a simplified kinematic model of the hand, which requires a reduced number of sensors to estimate the finger pose. To reconstruct the finger pose, the developed Android app combines the orientation estimated by two sensor boards and biomechanical constraints [149, 160]. An ATMega 328p microcontroller (Arduino Pro Mini 3.3 V) is in charge of communicating through a Bluetooth connection with the smart-phone to control the servomotor and to transmit data coming from the inertial sensors.

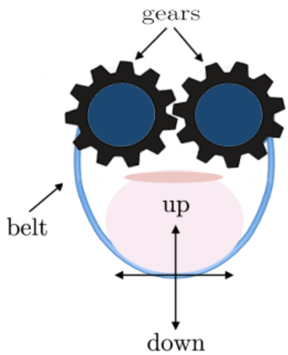

Figure 4.10: The actuating solution adopted in the proposed haptic thimble. One of the two gears is directly connected to the servomotor shaft. When the motor rotates, the two gears rotate in opposite directions, and the belt is pulled up/down, providing a force normal to the finger.

Second Prototype

Haptic rendering comes with a variety of interaction modalities, in particular we are interested in wearable actuators. A complete survey of wearable haptic interfaces is depicted in [88]. Among the different typologies of tactile stimulation, we selected skin indentation, which is the most intuitive for perceiving the sensation of making and breaking contact with objects. Starting from the results presented in [161], we designed a system capable of generating forces along a single axis, which is enough to perceive contact pressures. To reduce the device encumbrance, we employed a single servo-motor attached to a flexible belt. The torque generated by the motor determines the direction of rotation of a master gear that consequently moves a slave gear. Such coupling results in opposite spinning direction of the gears, that move the belt along the vertical axis (as depicted in Fig. 4.10). Subjects were asked to wear the interface on the index finger by placing the fingertip over the belt. The force exerted by the belt simulates the virtual contact forces by deformating the skin. The maximum belt translation in the vertical direction can be selected according to the external diameter of the gears, the length of the belt, and the rotation range range of the servo motor. We selected the maximum range available for the device, considering that also fingertips bigger than the average should fit without always being in contact with the flexible belt. We set the maximum range at 23 mm. Considering that the servo-motor maximum rotating range is 120°, the optimal belt length and gears external diameter to minimize the mechanism size were respectively 65 mm and 11 mm. The mechanism is integrated in a wearable interface worn at the fingertip, that is also used to estimate the index finger orientation with respect to the back of the hand (Fig. 4.10). The fingertip device is controlled by a microcontroller based on an ARM Cortex-M0, that is in charge of communicating with the smartphone via BLE, reading the IMU, and controlling the servo-motor. The Android app, presented in Subsect. 4.2.2, estimates the contact force and sends a corresponding value mapped in the useful range (defined during the calibration phase). The control of the haptic display is based on the assumption of linear relationship between the force generated and fingertip deformation. Hence. the force exerted by the center of the belt can be expressed as:

$$F_f = \left(\frac{\tau}{r} - k_f \Delta l\right),$$

where F_f is the force exerted on the fingertip, τ is the motor torque (max $0.8\,\mathrm{kg\,cm^{-1}}$ for the servomotor used), $r = 5.5 \times 10^{-3}\,\mathrm{m}$ is the pulley radius, Δl is the finger pulp indentation. The stiffness of the fingertip refers to a simplified model. In accordance with [162], we assume the relationship between the belt displacement and the generated normal force to be linear. We consider a stiffness k_f $=0.5\,\mathrm{N\,m^{-1}}$ as elastic behavior of the finger pulp. So that, we can compute the desired displacement for generating a requested force:

$$\Delta l = K_f^{-1} F_f,$$

where Δl is the displacement of the belt with respect to its initial position, *i.e.*, when the belt is in contact with the finger pad without producing any skin deformation. Thus, the evaluation of the motor angle results straightforward. Because of the design of the device, the maximum force that can be provided to the human's finger pad is about 5.0 N.

The useful servo range in each experiment was decided depending on the subject fingertip size. During the manual calibration, the operator defined the servomotor motion range according to the point corresponding to the maximum force exerted by the motor and the minimum contact point. These values were then inserted in the app settings interface, which communicated the useful range to the microcontroller. Every time the app detects contact, it sends a value to the microcontroller which performs the mapping in the useful servo range.

Hand Tracking

A necessary step to achieve and measure the interaction with virtual objects is to project a representation of the hand in the virtual environment reference system. In order to measure the distance between the hand representation and the fiducial marker (thus with the virtual objects) without additional external hardware,