



# Data integration by two-sensors in a LEAP-based Virtual Glove for human-system interaction

Giuseppe Placidi<sup>1</sup> · Danilo Avola<sup>2</sup> · Luigi Cinque<sup>2</sup> · Matteo Polsinelli<sup>1</sup> · Eleni Theodoridou<sup>1</sup> · João Manuel R. S. Tavares<sup>3</sup>

Received: 25 May 2020 / Revised: 5 October 2020 / Accepted: 22 December 2020 /

Published online: 17 February 2021

© The Author(s) 2021

## Abstract

Virtual Glove (VG) is a low-cost computer vision system that utilizes two orthogonal LEAP motion sensors to provide detailed 4D hand tracking in real-time. VG can find many applications in the field of human-system interaction, such as remote control of machines or tele-rehabilitation. An innovative and efficient data-integration strategy, based on the velocity calculation, for selecting data from one of the LEAPs at each time, is proposed for VG. The position of each joint of the hand model, when obscured to a LEAP, is guessed and tends to flicker. Since VG uses two LEAP sensors, two spatial representations are available each moment for each joint: the method consists of the selection of the one with the lower velocity at each time instant. Choosing the smoother trajectory leads to VG stabilization and precision optimization, reduces occlusions (parts of the hand or handling objects obscuring other hand parts) and/or, when both sensors are seeing the same joint, reduces the number of outliers produced by hardware instabilities. The strategy is experimentally evaluated, in terms of reduction of outliers with respect to a previously used data selection strategy on VG, and results are reported and discussed. In the future, an objective test set has to be imagined, designed, and realized, also with the help of an external precise positioning equipment, to allow also quantitative and objective evaluation of the gain in precision and, maybe, of the intrinsic limitations of the proposed strategy. Moreover, advanced Artificial Intelligence-based (AI-based) real-time data integration strategies, specific for VG, will be designed and tested on the resulting dataset.

**Keywords** Virtual Glove (VG) · Hand rehabilitation · Orthogonal LEAP motion sensors · Human-system interaction · Data-integration

## 1 Introduction

In recent years, computer vision is becoming increasingly important in addressing a wide range of application areas, including human action recognition [4, 16], aerial image

---

✉ Giuseppe Placidi  
giuseppe.placidi@univaq.it

processing [5, 31], person re-identification [33, 37], and human-system interaction [14, 32]. Concerning the latter, its goal is to improve the communication between users and computers, virtual reality environments, electromechanical devices, and robots. With the use of highly sophisticated sensors, a lot of critical applications on remotely operating systems, e.g., driving robots, rovers, or performing medical procedures [10, 19, 25, 38, 39], are becoming possible. Tele-operated systems are expensive, neither replicable nor quickly replaceable. The results of long-planned, critical, costly, and challenging operations depend on their proper use, that requires precise recording and reproduction of the operator's hand and finger movements.

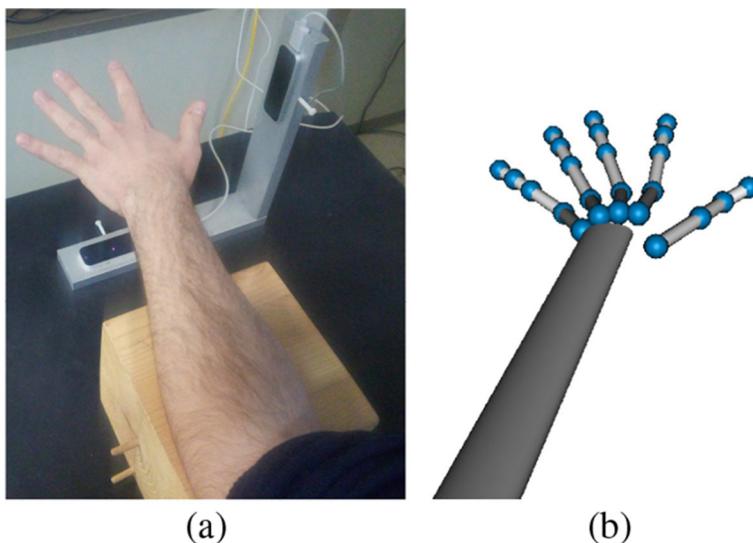
Both non-vision and vision-based gesture recognition are usually employed to finely track the hand and all its joints. The non-vision approaches utilize wearable devices, such as wired gloves, for the detection of finger movements [7, 20, 34], while vision-based approaches use the interpretation of video-collecting devices, usually sensors operating also in the infrared (IR) range, placed at a certain distance from the subject [2, 9, 10, 27, 28, 39, 41]. The key advantage of vision-based systems is that no physical contact is required and the movements are free and natural, being the hand unforced to wear anything, and it could naturally be used to grip specialized tools in order to carry on a procedure (for example, surgical devices). However, in order to be used to control remotely operating systems, the movements must be identified with good spatial precision (few millimetres) and in real-time (at least 30 frames per second, fps, are necessary). In the last few years, the use of immersive Virtual Reality (VR) interfaces driven by natural hand movements for remote control, is growing up thanks to the development of innovative optical 3D vision-based systems for gesture recognition [12] and the range of applications that benefited from them is increased, as is occurring in rehabilitation [3, 30]. One of the most recent optical 3D sensors, based on stereo vision, is the LEAP motion controller (LEAP®<sup>1</sup>). LEAP is a high-resolution 3D hand-sensing device which allows the freehand natural interaction, crucial for the implementation of real-time, realistic VR systems [6, 30]. It uses 3 IR light sources and two detectors to obtain 3D visual information saved and reproduced almost simultaneously (more than 60fps) from the server. It has been successfully integrated with VR environments in rehabilitation and neuroergonomics [8, 26, 30], and also used as a tool for touchless interfaces, such as 3D writing recognition systems [18]. One of its advantages is that it is appropriate for different hand sizes (adults and children), as well as for different hand shapes (healthy people and patients with residual infirmities). However, if objects have to be handled, e.g., a joystick or the controller of a remotely operated vehicle, they can produce occlusions. Even parts of the hand itself often cross over with the view of the sensor (self-occlusions). Thus, LEAP, such as most vision-based systems, can fail to correctly reproduce the hand trajectory because the spatial position of some joints of the hand, invisible to the sensor, are guessed, thus resulting in inaccurate and unstable representations. That could be negligible when just raw gestures need to be reproduced, but crucial when finer movements are used in tele-operated applications, such as tele-surgery or operations in dangerous environments. Recently, several works have been published with the aim of improving hand tracking accuracy by combining LEAP data with those of other devices or data from multiple LEAPs [17, 21–23, 40]. In particular, in [21] a LEAP is supported by an RGB webcam to improve the quality of the recognition of symbols in the 3D American sign language datasets. Aim of the proposed system is to reduce the ambiguities, due to occlusions, in gesture recognition: the RGB webcam is used as an auxiliary system, being

---

<sup>1</sup><https://www.ultraleap.com/>

it unable to furnish specific spatial information. The same gesture recognition problem for identification of American sign language and Handicraft-Gesture is solved accurately with just one LEAP [40]. In [22, 23] a LEAP is supported by a Depth camera. The system has very good accuracy (regarding gesture recognition) but low frame rate (15fps) making it not suitable for applications that requires higher frequency (30fps or greater) to track natural hand movements. Moreover, due to the occlusions between fingers, the method can perform well only when the hand is in ideal orientations/positions. Kiselev et al. [17] use three LEAPs for gesture recognition. The Authors show that by increasing the number of sensors, the accuracy also increases due the fact that the number of occlusions decreases. Moreover, the use of multiple sensors of the same type greatly improves the performance of the data integration strategy due to the easiness in comparing similar models. However, since just one LEAP at time can be driven by a single operating system, the used client/server architecture described in the paper suggests that at least three different computers have been used (cheap and critical in terms of synchronization). In addition, as two of the three LEAPs are coplanar, they mostly contribute to increase the active region but have low influence in reducing occlusions. Finally, the performance of the system, in terms of frame rate, has not been discussed. Shen et al. [35], solve the problem of occlusions in gesture recognition by proposing the use of three LEAPs placed with their long axes on the medium points of the sides of an equilateral triangle. Though the paper deeply discuss on the system assembly, calibration, data-fusion and results in terms of position/orientation accuracy, no mention is dedicated to the resulting efficiency of the system in terms of fps.

Virtual Glove (VG) is a system based on the synchronized use of two orthogonal LEAPs (Fig. 1) for reducing the probability of occlusions [30]. Better results regarding occlusions reduction could have been obtained with three LEAPs on a equilateral/equiangular configuration, as in [35], but we would have had serious problems with the real-time maintenance (at least 30fps) on a low-cost computer. Though the paradigm of VG [29] is applicable



**Fig. 1** Hand positioning inside VG **a** and the corresponding numerical model **b** reconstructed by using 24 joints (4 for the thumb and 5 for each of the other fingers)

to any number of sensors placed in any angular configuration, the choice of two orthogonal sensors represents a good compromise between optimization/positioning of the region of interest (ROI), precision and efficiency. In fact, through project-related considerations and qualitative measurements regarding position/dimensions of the ROI and precision with respect to the angle, it can be argued that: 1) An acute angle between the sensors planes [15], though useful to approach the ROI to sensors and to maximize precision of the sensors individually, would reduce the space between sensors that implies a reduction of the hand movements inside the system and, consequently, a reduction of the ROI. Moreover, IR reciprocal interferences between sensors would increase, thus resulting in a reduction of stability, reliability and, hence, of the final precision of the tracking; 2) An obtuse angle between the sensors planes, though increasing the space between sensors, would move away the ROI from the sensors surface, thus reducing the precision of the system. In the original embodiment of VG, data coming only from one of the LEAPs were used at each time instant by mutual exclusion: the one having the most favourable orientation with respect to the hand palm was chosen. Though simple and efficient, this solution did not solve many cases of occlusions and, to increase efficiency, a lot of useful information coming from the orthogonal sensor got wasted.

To integrate data from both LEAP sensors and to solve the problem of data wasting, we have also considered the possibility offered by Machine Learning [24], ML, and Deep Learning [36], DL, but, though very effective, they could be either too slow or too computationally expensive to be used in a low cost machine (VG system is imagined for accurate and, in the same time, low-cost human-system interaction [20]). Besides, we would face the difficulty of getting sufficiently populated labelled datasets to be used for training and composed by the spatial positions of the hand joints (collected by a position indicator and considered the ground truth) and the corresponding spatial positions measured by both LEAP sensors while moving the hand inside the VG. This last task, necessary for using ML and DL strategies, is a long process that, to be carried on, requires the usage of an advanced, mini-invasive (LEAP sensors have to view the hand and its joints) and precise position indicator, such as one of those produced by VICON<sup>®2</sup>, to be installed on the hand.

Aim of the present paper is to design and test a completely different data integration approach for VG, a good trade-off between simplicity, efficacy and efficiency without the requirement of any training datasets. The rest of the manuscript is structured as follows: Section 2 reviews VG assembly (both hardware and data collection strategy). Section 3 details the proposed data-integration method. Section 4 presents experimental measurements, results, and discussion. Finally, Section 5 concludes the manuscript and delineates future work and developments.

## 2 The VG assembly

### 2.1 Design, calibration, and sensors management

The VG hardware consists of a rigid support, equipped with lodges for the orthogonal LEAP sensors Fig. 1a. The sensors are fixed inside the lodges through plastic screws to avoid vibrations and movements. The center of each LEAP is positioned at 18.5 cm from the internal

---

<sup>2</sup><https://www.vicon.com/hardware/>

corner of the support: these measurements were optimized for maximizing the signal into a 21 cm side, while also reducing VG’s dimensions.

Both sensors were calibrated to a common right-handed Cartesian coordinate system, the center of which lies on the LEAPs plane, at the intersection of their vertical axes. Calibration was performed by accurately measuring, with a high precision positioning system<sup>3</sup> (spatial precision 0.01mm) the position of a tip of a stick on a set of  $m$  points inside the region of interest of the VG. On the same points, spatial measurements were collected by both LEAPs (one sensor at a time) and by calculating the transformation matrix [11]. Given the cloud of points measured by the two LEAPs each with its proper reference system,  $A = \{\mathbf{a}_i : 1 \leq i \leq m\}$  and  $B = \{\mathbf{b}_i : 1 \leq i \leq m\}$ , it finds the rotation matrix  $R$  and the translation vector  $\mathbf{s}$  that minimizes the error:

$$err = \frac{1}{m} \sum_{i=1}^m \|\mathbf{R}\mathbf{a}_i + \mathbf{s} - \mathbf{b}_i\|^2 \tag{1}$$

The center of mass for both sets,  $\mathbf{C}_A = \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i$  and  $\mathbf{C}_B = \frac{1}{m} \sum_{i=1}^m \mathbf{b}_i$ , are calculated and used to center the sets on the origin:

$$\mathbf{A}^+ = \{\mathbf{a}_i^+ : \mathbf{a}_i^+ = \mathbf{a}_i - \mathbf{C}_A\} \quad \mathbf{B}^+ = \{\mathbf{b}_i^+ : \mathbf{b}_i^+ = \mathbf{b}_i - \mathbf{C}_B\} \tag{2}$$

This allows to compute the cross-covariance matrix:

$$H = \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i^+ \mathbf{b}_i^{+T} \tag{3}$$

and to apply the Singular Value Decomposition (SVD) to decompose  $H$  in the vector  $[U, S, V] = SVD(H)$ , such that  $H = USV^T$  where  $U$  and  $V$  are orthogonal matrices and  $S$  is a non-negative diagonal matrix. In VG the rotation  $R$ , can be computed by  $R = UV^T$  and the translation by  $\vec{s} = -RC_B + C_A$ .

The resulting transformation, in homogeneous coordinates, is:

$$W = \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} & s_1 \\ R_{2,1} & R_{2,2} & R_{2,3} & s_2 \\ R_{3,1} & R_{3,2} & R_{3,3} & s_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

Regarding the operation of two sensors at the same time, the software development kit of the LEAP (SDK)<sup>4</sup> does not allow the use of two devices on the same operating system and an architecture based on the use of virtual machines is necessary. In our architecture, two virtual machines (slaves) are installed on the physical machine (master) and each of them manages one of the two sensors. Data provided by the SDK through the websocket are captured by a javascript router and returned to a server hosted on the master machine. In the same way, the server sends data from both devices to one or more clients running on the master. The server receives data from the routers and elaborates them by performing the coordinate transformation and by constructing, and representing on a virtual environment, the numerical hand model. The hand model structure could variate depending on the specific programming language SDK. In fact, VG uses Javascript API and the computations are performed by employing the bone class: Given a Hand instance, it has access to the Arm (bone class) and to the Finger classes. Each Finger accesses to its bones (i.e.,

<sup>3</sup>GUALDONI Mod. 49 FU 80, 1995, Milan - Italy

<sup>4</sup><https://developer.leapmotion.com/setup/desktop>

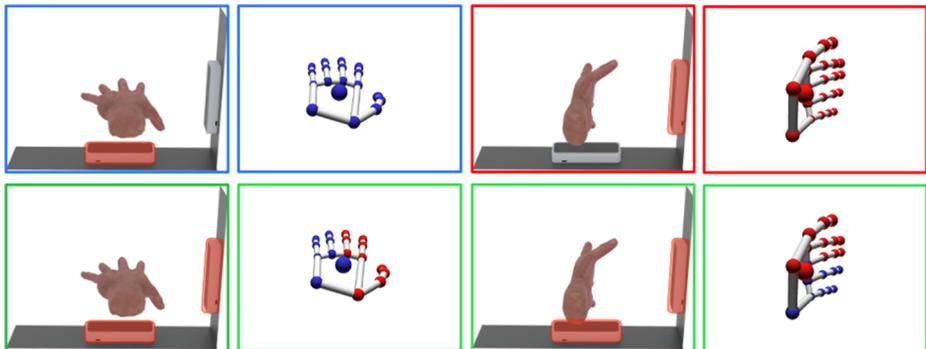
metacarpal, proximal, intermediate, and distal) and joints (i.e., attributes `carpPosition`, `mcpPosition`, `pipPosition`, `dipPosition`, `btipPosition`). Fig. 1 shows the hand inside the VG (a) and the corresponding numerical model (b).

## 2.2 Original data collection strategy

The original hand tracking strategy is based on a switching approach, i.e., at any given time instant  $t$ , only one sensor, and the same for all joints, is used to track the hand. In fact, both sensors are switched on but only one LEAP at a time is active and furnishes data (Fig. 2, top row). To determine which LEAP is active (the “favorite” sensor), the palm’s normal vector  $p$ , (a vector orthogonal to the palm of the hand), is used to find the angle between the X-axis of the horizontal LEAP reference system and the projection of  $p$  on the X-Y plane. If the angle is between  $225^\circ$  and  $315^\circ$  (the palm is facing downwards) or between  $45^\circ$  and  $135^\circ$  (the palm is directed upwards) the horizontal LEAP is active, while data from the vertical sensor are ignored. Out of these ranges, the role of the sensors is inverted: the vertical LEAP becomes active and data from the horizontal LEAP are ignored. Though this approach is very efficient (just hand orientation is necessary to choose which model to use) and capable to fix occlusions caused by the hand’s palm, it performs poorly when the hand is not perfectly oriented toward one of the sensors and/or when the hand is bending and some fingers obscure the others (this could occur in any orientation of the hand). In fact, with mutual exclusion, just data coming from one sensor are used each time instant, and, by discarding those of the other, a lot of potentially useful information is lost. These effects are accentuated when occlusions increase because of the handling of an object. In what follows we describe the new strategy we propose to exploit data from both LEAPs at the same time, thus improving the VG’s capability of reducing occlusions.

## 3 The proposed data integration strategy

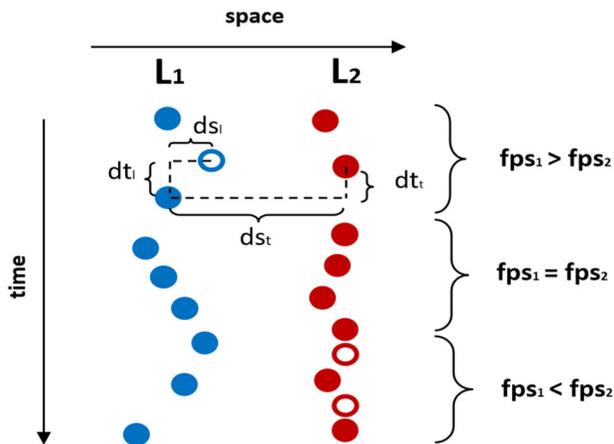
We aim at using data coming from both sensors. In particular, the role of “favourite view” to the sensor which has the palm of the hand oriented toward it, as in the mutual exclusion



**Fig. 2** Running example. The switching modality (top row) and the proposed integration strategy (bottom row). In the top row, the switching modality selects data from just one of the LEAPs. With the proposed modality, second row, the joints are selected according to their stability and data coming from both LEAPs are fused in the same model. Notice that, blue, red, and green are referred to horizontal LEAP, vertical LEAP, and the proposed data integration strategy, respectively

strategy, is maintained just at the beginning of acquisition but, after that, data from both sensors are checked for each joint and just those from one of the sensors, each time, are selected in terms of stability and used to track that joint, i.e., at any time  $t$ , different joints of the model could be associated to different sensors (Fig. 2, bottom row). The reason of this choice is two-fold: the hand is a dynamic structure and, during time, a joint could be alternately obscured or visible; when a joint is lost by one sensor, its guessed position could be very far from the correct value, correctly represented by the other sensor (in this case, merging data from both sensors would reduce position errors). In fact, when LEAP is tracking the hand, it correctly represents the joints that it sees and guesses those that it does not see due to occlusions.

When LEAP loses a joint, first it becomes temporally unstable (forming high-frequency flickers and shakes) and then it stabilizes the guessed position and maintains it still until the joint becomes visible again: at that point, the position is updated to the right one (also this update occurs abruptly). This produces jumps and spikes on the trajectory that could also consist of errors of centimetres (see the Section 4 below). A LEAP hand model contains data for all joints and for each time, even if some of them are invisible to the sensor. In this last case, the positions of invisible joints are guessed on the basis of the hand shape and previous temporal view (a proprietary LEAP strategy). The strategy we propose is to check, for each joint, the data flow coming from both sensors and to choose those coming from the more stable of the two. Joint's stability is inversely proportional to its velocity: when the model is unstable, spikes and jumps are produced in the trajectory and velocity is high. Since we have data of the same joint from both sensors, the velocity is computable and finite for both LEAPs and the corresponding values can always be compared: for each time instant, data are selected from the LEAP showing lower velocity, in module. The data flow, from both sensors, is shown in Fig. 3. Each sensor is collecting data with varying frame rate, also different between LEAPs, and data from closest times are compared. Two velocity values are calculated: one is the velocity along the same sensor, that we call Internal (intra-sensor) velocity, and the other is the velocity "produced" by skipping from one sensor to the other,



**Fig. 3** Example of data production from both LEAPs (blue and red dots), along time (vertical axis) and space (horizontal axis). The fps can change with time: always the slower is used to produce data. Empty dots, from the faster sensor, are discarded. The parameters (space and time changes) for the calculation of Internal and External velocities are also shown. The reported example is just representative (spatial discrepancies between sensors have been enlarged for graphical purposes)

that we call External (inter-sensors) velocity. External velocity is usually present because of the spatial differences between the two sensors (see Fig. 3). We first define these velocities and then we describe the stabilization strategy. For each joint ( $i = 1, 2, \dots, 24$ ) of the sensor  $L_j$  ( $j = 1, 2$ ), we calculate the Internal velocity:

$$|vI_i(t)|_{L_j} = \sqrt{\left(\frac{dx_{i,L_j}}{dt_{L_j}}\right)^2 + \left(\frac{dy_{i,L_j}}{dt_{L_j}}\right)^2 + \left(\frac{dz_{i,L_j}}{dt_{L_j}}\right)^2} \tag{5}$$

and the External velocity (it does not depend on one specific sensor):

$$|vE_i(t)| = \sqrt{\left(\frac{dx_i}{dt}\right)^2 + \left(\frac{dy_i}{dt}\right)^2 + \left(\frac{dz_i}{dt}\right)^2} \tag{6}$$

By indicating with  $C_{L,i}$  the LEAP currently used for the joint  $i$ , the resulting integration algorithm is the following:

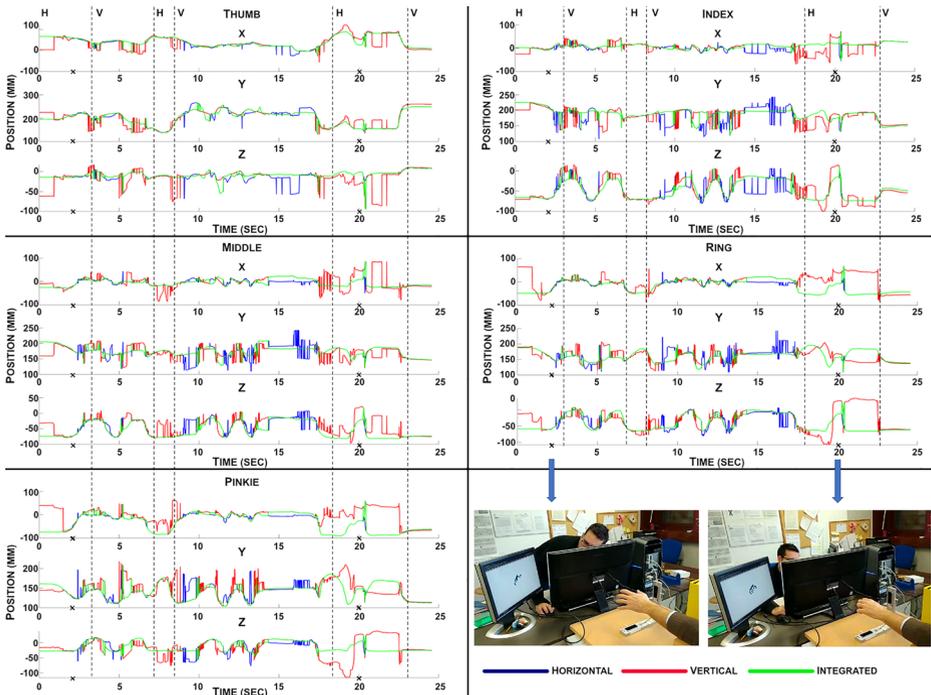
1. At the starting time, take the hand model from the favourite LEAP for all the joints and update  $C_L$  (here the  $i$  is missing because the LEAP is the same for all the joints);
2. Step to the following time  $t$  (that of the slower LEAP);
3. For each joint  $i$ :
  - (a) For each LEAP  $L_j$ :
    - (i) Calculate  $|vI_i(t)|_{L_j}$  and  $|vE_i(t)|$ ;
4. Verify conditions in Table 1, take the data from the appropriate LEAP and update  $C_{L,i}$  accordingly;
5. Go to step 2.

The conditions in Table 1, a truth table, allow to define from which LEAP we have to select data for the joint  $i$  at the time  $t$ . As it can be noticed, we first define the lowest value for the Internal velocity and, if a change of sensor is necessary with respect to the current  $C_{L,i}$ , we also check whether the External velocity is lower than the current, Internal, one. If this condition is met (data across sensors are more stable than those into the current one), a sensor data skip is allowed, otherwise data are collected from the original sensor. As it can be noticed, time also affects the LEAP choice because the number of fps changes with time for both sensors, and the two fps could even be always different. However, we always use the lower fps. In Fig. 4 an example is illustrated. The derivative calculation is obviously discrete.

The resulting hand model is a mixture of joints tracked from both sensors, resulting in a smoother train of points. The compared values refer to the same joint. No velocity threshold

**Table 1** Truth table indicating, inside the cells, the sensor that has to be chosen if the logical conditions (row and column) are met at the same time (logical AND)

	$ vI_i(t) _{L_1} <  vI_i(t) _{L_2}$	$ vI_i(t) _{L_1} \geq  vI_i(t) _{L_2}$
$C_{L,i} = 1$	$L_1$	$ vE_i(t)  \leq  vI_i(t) _{L_1}$ $ vE_i(t)  >  vI_i(t) _{L_1}$ $L_2$ $L_1$
$C_{L,i} = 2$	$ vE_i(t)  \leq  vI_i(t) _{L_2}$ $ vE_i(t)  >  vI_i(t) _{L_2}$ $L_1$ $L_2$	$L_2$



**Fig. 4** Fingertip trajectories on x, y, and z axes for a 25 sec free-hand moving exercise. Blue and red lines represent the trajectories obtained, respectively, by the horizontal and the vertical sensor; while the green trajectory is the one produced by the new proposed approach. Dashed lines indicate LEAP switch points in the original switching strategy. Two screenshots of the video, occurring at about 2.0 sec and 20.0 sec, respectively, are also reported (bottom right), to indicate: the experimental conditions; the position of the hand; and the real-time representation of the model on the screen of a PC. X symbols on the time axes indicate screenshots occurrences

is needed: no constraint needs to be set regarding the maximum velocity of the hand. In fact, having two sensors to register the same joint, it can be supposed that, if the joint is moving, the smoother track is the more precise between the two. To further improve precision, it could be useful to verify how to merge data when both sensors are operating correctly. In that case, however, additional calculations are necessary to verify the correctness of the data but that could preclude real-time.

## 4 Experimental measurements and results

### 4.1 Data collection

To demonstrate the effectiveness of the proposed strategy in comparison with the usage of a single sensor, measurements were collected while a subject moved the left hand inside the VG (the usage of the right hand would have been the same). We performed the experiment with the hand free, without grabbing any object, in order to highlight that: 1) the instability effects on the reproduced trajectories are due to the loss of the signal (occlusions) and not to disturbances due to the grabbed object; 2) also in a free-hand mode the number of

occlusions during the tracking process is high. The experiment started with the hand still oriented toward the horizontal LEAP, followed by wrist rotations alternated to a sequence of hand open-close operations. The number of wrist rotations was 5, corresponding to 6 hand positions with respect to the LEAPs (these are important to establish the changes of orientation of the hand with respect to both sensors, as clarified below). The duration of the sequence was 25 seconds for a total of 961 4D positions ( $x$ ,  $y$ ,  $z$ , and  $t$  all referred to the world system of VG). The hand model reconstructed in real-time by the proposed strategy was shown on a computer screen and saved into a database (DB). Apart from the reconstructed model, also original models obtained by each LEAP were stored into the same DB.

The whole experiment was also recorded using an external video camera and time was monitored by a stopwatch. Conditions of the room were maintained normal in order to avoid favorable conditions: no particular attention was paid to maintain external interferences low (controlled light, temperature, electromagnetic disturbances, and so on) and to maintain the background free of objects.

## 4.2 Results and discussion

Data obtained with the proposed method and those from each single LEAP have been recovered from the DB to be shown into the same plots. To this aim, the trajectories of just the 5 fingertips, organized by axes, are presented in Fig. 4 where three lines are reported: data from the horizontal LEAP (blue line), data from the vertical LEAP (red line) and data obtained with the proposed integration strategy (green line). As it can be observed, the green curve follows alternatively values from the blue or from the red curves by remaining on the smoother one. In fact, the green curves are smoother than blue and red ones and, in that way, also spikes and jumps, obviously representing tracking losses or outliers, are reduced. A summary of the outliers' reduction by using the proposed solution with respect to the switching approach is reported in Table 2. Though most of the outliers are removed, some of them remain mostly where both sensors are unstable at the same time.

Obviously, the data collection strategy originally used in VG would maintain all the outliers occurring for the actually active sensor, since the only information used to get data from one of the LEAPs was the orientation of the hand. Fig. 4 also indicates, with vertical dashed lines, the instants where the switching between sensors occurred in the original procedure, because the limit angles were overgone. Further, additional discontinuities could

**Table 2** Outliers. The number of outlier spikes using the old switching approach (before) in comparison to the new proposed method (after), and their reduction's percentage

Finger	x			y			z		
	Before	After	% Reduction	Before	After	% Reduction	Before	After	% Reduction
Thumb	19	6	68,4	18	13	27,8	21	12	42,9
Index	35	11	68,6	54	19	64,8	49	17	65,3
Middle	23	9	60,9	58	21	63,8	35	17	51,4
Ring	15	8	46,7	34	16	52,9	31	12	61,3
Pinkie	11	6	45,5	37	19	48,6	29	13	55,2

be produced by the transition from one LEAP to the other, as it can be observed at both sides of the vertical lines in the plots. Another important aspect to be noticed is that the selection of the blue or the red trajectory by the green one depends on the specific joint and not on the orientation of the hand palm (in that way, at a given time instant, a joint uses data by looking at its own smoother trajectory which could be different from that used by another joint).

Particular attention should be paid at the difference between spatial representations of the two LEAPs for the same joint that, in some traits, can be very high and continues to be high for a long time. As said before, this depends on the behavior of the sensor: when a LEAP has to guess the position of a joint, it chooses the best estimated position and maintains it until it sees the joint again. During this time, the positions indicated by the two sensors could greatly differ and this justifies the usage of data coming from just one of them instead of merging data from both. Such an effect also becomes evident in Fig. 4, especially for the time interval between  $t=1$  sec and  $t=3$  sec, where pinkie finger is shadowed by the rest of the hand with respect to the vertical sensor: its guessed positions by the vertical LEAP are quite different from those collected (correctly) by the horizontal one. When data are collected by the proposed integration strategy, the correct position is selected.

Figure 4 also shows relevant snapshots of the experiment on which some fingertip occlusions are highlighted: thanks to the proposed identification strategy, the trajectories are smoothed, as it can be observed on the plots, and the final reconstructed hand model is correctly reproduced on the computer screen in real-time. As noticeable, the system can follow and sufficiently record hand movements without any special preparation needed. The presented results have been acquired under normal conditions, which indicates that the VG system is capable to perform well also outside a laboratory and, after further development, makes it a good candidate for future applications in external environments. Based on these qualitative results, the model shape resembled the real hand accurately and, most importantly, the model followed the hand movements in real-time when operated on a PC with Intel I7, 32Gb Ram, NVIDIA GE force GTX 1080. The results confirmed that the model was represented on the screen at 47fps (demonstrated by registering the timestamps of the presentations on the screen) which is about 1.5 times the frequency required to consider human-systems interaction useful for real-time (about 30fps). This high frequency image acquisition is what allows us to combine and synchronize the two LEAP systems, even when they do not work on the same fps.

Table 2 shows that outliers in all five finger trajectories have been reduced to more than their half. In particular, the average reduction is 58% for the x, 51.6% for the y, and 55.2% for the z direction. To conclude the analysis, we want to remark that, both from Fig. 4 and from Table 2, certain fingers show a lower number of outliers than others. Thumb has the less outliers in both approaches (its averages on the three axes are 19.3 and 10.3 for the old and the new method respectively) while pinkie comes second using the old method (25.7 outliers on average) and ring finger has the second less spikes with the new method (12 on average). This is obviously due to their more favorable position (external). To provide an overview of outlier average values for both fingers and axes, Table 3 is also presented. Moreover, two other considerations can be made: 1) the x coordinate is more stable than other coordinates (there is a not evident explanation to that behavior and we have to explore it); 2) there is a different behavior between the two LEAPs of the VG (the horizontal one was more stable than the vertical). This point is probably due to internal hardware differences between the two sensors and it is not an environment effect. In fact, by rotating the VG of 90°, the two LEAPs maintain the same behaviour.

**Table 3** Outlier Average Values

Finger	Before	After	% Reduction	Axis	Before	After	% Reduction
Thumb	19,3	10,3	46,4	x	20,6	8,0	58,0
Index	46,0	15,7	66,2	y	40,2	17,6	52,0
Middle	38,7	15,7	58,7	z	33,0	14,2	55,2
Ring	26,7	12,0	53,6				
Pinkie	25,7	12,7	49,8				

Average values of the plot outliers with the old and the new data integration method, together with their % reduction percentage, separated by finger and grouped by axis (first four columns) and separated by axis and grouped by finger (the remaining four columns)

## 5 Conclusion

Occlusions are one of the biggest and most studied issues in hand movement tracking. The LEAP system, with its low cost and simple setup, offers the opportunity of significantly reducing the problem, by using more than one detector to multiply the visual information. We presented a new strategy for selecting data coming from both sensors forming the VG, a system composed by two vertically placed LEAP detectors to provide 4D hand tracking in real-time. The proposed strategy has made it possible to reduce occlusions, to avoid outliers and false position indications (errors) with respect to using data from just one sensor, and to increase the stability of VG. These are the necessary conditions by which VG, being a touchless system which leaves the hand free to perform natural movements, could be effectively used for reproducing hand and finger movements with good spatial and temporal resolution and, hence, to drive systems remotely with high accuracy. However, the proposed results are only capable to demonstrate qualitative improvements with respect to the original mutual exclusion strategy. Future work will be dedicated to organize measurements from which it would be possible to obtain also quantitative evaluations and to study possible countermeasures to the residual instabilities produced when some views are obscured with respect to both sensors at the same time (the usage of a third detector, as in [35], could help but the real-time conditions have to be checked). Moreover, advanced data integration strategies based on AI will be designed and tested to improve VG precision and stability, while maintaining a sufficiently low computational load for a low-cost machine to maintain real-time. In particular, we aim to reach this goal by using AI-based approaches, such as those in [1, 13, 40], applied to the temporal trajectories described by each joint of the hand.

**Acknowledgements** The work has been financially supported by the Italian Minister of the University and Research (Dottorato di Ricerca innovativo a caratterizzazione industriale borsa n.2, PON 2014-2020).

**Funding** Open Access funding provided by Università degli Studi dell' Aquila

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Ameer S, Ben Khalifa A, Bouhleb MS (2020) A novel hybrid bidirectional unidirectional lstm network for dynamic hand gesture recognition with leap motion. *Entertain Comput* 35:1–10
2. Ankit C, Jagdish RL, Karen D, Sonia R (2011) Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey. *Int J Comput Sci Eng Survey* 122–133
3. Avola D, Spezialetti M, Placidi G (2013) Design of an efficient framework for fast prototyping of customized human–computer interfaces and virtual environments for rehabilitation. *Comput Methods Prog Biomed* 110(3):490–502
4. Avola D, Bernardi M, Foresti GL (2019) Fusing depth and colour information for human action recognition. *Multimed Tools Appl* 78(5):5919–5939
5. Avola D, Cinque L, Foresti GL, Pannone D (2020) Homography vs similarity transformation in aerial mosaicking: Which is the best at different altitudes? *Multimed Tools Appl* 79:18387–18404
6. Bachmann D, Weichert F, Rinkenauer G (2014) Evaluation of the leap motion controller as a new contact-free pointing device. *Sensors* 15(1):214–233
7. Battaglia E, Bianchi M, Altobelli A, Grioli G, Catalano MG, Serio A, Santello M, Bicchi A (2016) ThimbleSense: A fingertip-wearable tactile sensor for grasp analysis. *IEEE Trans Haptics* 9(1):121–133
8. Carrieri M, Petracca A, Lancia S, Moro SB, Brigadoi S, Spezialetti M, Ferrari M, Placidi G, Quaresima V (2016) Prefrontal cortex activation upon a demanding virtual hand-controlled task: A new frontier for neuroergonomics. *Front Hum Neurosci* 10(53):1–13
9. Chen L, Wei H, Ferryman J (2013) A survey of human motion analysis using depth imagery. *Pattern Recogn Lett* 34(15):1995–2006
10. Chen S, Ma H, Yang C, Fu M (2015) Hand gesture based robot control system using leap motion. In: *Proceedings of the intelligent robotics and applications (ICIRA)*, pp 581–591
11. Eggert DW, Lorusso A, Fisher RB (1997) Estimating 3-d rigid body transformations: A comparison of four major algorithms. *Mach Vision Appl* 9(5–6):272–290
12. Erden F, Çetin AE (2014) Hand gesture based remote control system using infrared sensors and a camera. *IEEE Trans Consum Electron* 60(4):675–680
13. Iacoviello D, Petracca A, Spezialetti M, Placidi G (2016) A classification algorithm for electroencephalography signals by self-induced emotional stimuli. *IEEE Trans Cybern* 46(12):3171–3180
14. Imran J, Raman B (2020) Deep motion templates and extreme learning machine for sign language recognition. *Vis Comput* 36(6):1233–1246
15. Jin H, Chen Q, Chen Z, Hu Y, Zhang J (2016) Multi-leapmotion sensor based demonstration for robotic refine tabletop object manipulation task. *CAAI Trans Intell Technol* 1
16. Khan MA, Akram T, Sharif M, Muhammad N, Javed MY, Naqvi SR (2020) Improved strategy for human action recognition; experiencing a cascaded design. *IET Image Process* 14(5):818–829
17. Kiselev V, Khlamov M, Chuvilin K (2019) Hand gesture recognition with multiple leap motion devices. In: *2019 24th conference of open innovations association. (FRUCT)*. IEEE, pp 163–169
18. Kumar P, Saini R, Roy PP, Pal U (2018) A lexicon-free approach for 3d handwriting recognition using classifier combination. *Pattern Recogn Lett* 103:1–7
19. Liu Y, Zhang Y (2015) Toward welding robot with human knowledge: A remotely-controlled approach. *IEEE Trans Autom Sci Eng* 12(2):769–774
20. Luzhnica G, Simon J, Lex E, Pammer V (2016) A sliding window approach to natural hand gesture recognition using a custom data glove. In: *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, pp 81–90
21. Mahdikhanelou K, Ebrahimnezhad H (2020) Multimodal 3d american sign language recognition for static alphabet and numbers using hand joints and shape coding. *Multimed Tools Appl* 79(31):22235–22259
22. Marin G, Dominio F, Zanuttigh P (2015) Hand gesture recognition with leap motion and kinect devices. *IEEE International Conference on Image Processing, ICIP 2014* pp 1565–1569
23. Marin G, Dominio F, Zanuttigh P (2016) Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimed Tools Appl* 75(22):14991–15015
24. Mehryar M, Afshin R, Talwalkar A (2018) *Foundations of machine learning*, 2nd edn. MIT Press, Cambridge
25. Mizera C, Delrieu T, Weistroffer V, Andriot C, Decatoire A, Gazeau J (2020) Evaluation of hand-tracking systems in teleoperation and virtual dexterous manipulation. *IEEE Sensors J* 20(3):1642–1655
26. Moro SB, Carrieri M, Avola D, Brigadoi S, Lancia S, Petracca A, Spezialetti M, Ferrari M, Placidi G, Quaresima V (2016) A novel semi-immersive virtual reality visuo-motor task activates ventrolateral prefrontal cortex: A functional near-infrared spectroscopy study. *J Neural Eng* 13(3):1–14
27. Placidi G (2007) A smart virtual glove for the hand telerehabilitation. *Comput Biol Med* 37(8):1100–1107

28. Placidi G, Avola D, Iacoviello D, Cinque L (2013) Overall design and implementation of the virtual glove. *Comput Biol Med* 43(11):1927–1940
29. Placidi G, Cinque L, Petracca A, Polsinelli M, Spezialetti M (2017) A virtual glove system for the hand rehabilitation based on two orthogonal leap motion controllers. In: *Proceedings of the 6th international conference on pattern recognition applications and methods - Volume 1: ICPRAM, INSTICC, SciTePress*, pp 184–192
30. Placidi G, Cinque L, Polsinelli M, Spezialetti M (2018) Measurements by a leap-based virtual glove for the hand rehabilitation. *Sensors* 18(3):1–13
31. Prasad MG, Akula SP, Vemula A, Chandran S (2019) Mosaicing of multiplanar regions through autonomous navigation of off-the-shelf quadcopter. *IET Cyber-systems and Robotics* 1(3):81–92
32. Quintas J, Menezes P, Dias J (2017) Information model and architecture specification for context awareness interaction decision support in cyber-physical human–machine systems. *IEEE Trans Human-Machine Syst* 47(3):323–331
33. Rui S, Qiheng H, Wei F, Xudong Z (2020) Attributes-based person re-identification via cnns with coupled clusters loss. *J Syst Eng Electron* 31(1):45–55
34. Rusák Z, Antonya C, Horváth I (2011) Methodology for controlling contact forces in interactive grasping simulation. *Int J Virt Reality* 10(2):1–10
35. Shen H, Yang X, Hu H, Mou Q, Lou Y (2019) Hand trajectory extraction of human assembly based on multi-leap motions. In: *2019 IEEE/ASME international conference on advanced intelligent mechatronics (AIM)*, pp 193–198
36. Shi Z (2019) *Advanced Artificial Intelligence*, 2nd edn. World Scientific
37. Tang Y, Xi Y, Wang N, Song B, Gao X (2020) Cgan-tm: A novel domain-to-domain transferring method for person re-identification. *IEEE Trans Image Process* 29:5641–5651
38. Wang Z, Wang D, Zhang Y, Liu J, Wen L, Xu W, Zhang Y (2020) A three-fingered force feedback glove using fiber-reinforced soft bending actuators. *IEEE Trans Ind Electron* 67(9):7681–7690
39. Wei LJ, Sen LW, Sani ZM (2015) Leap motion underwater robotic arm control. *Jurnal Teknologi* 74(9):153–159
40. Yang L, Chen J, Zhu W (2020) Dynamic hand gesture recognition based on a leap motion controller and two-layer bidirectional recurrent neural network. *Sensors* 20:2106–2123
41. Zhang W, Cheng H, Zhao L, Hao L, Tao M, Xiang C (2019) A gesture-based teleoperation system for compliant robot motion. *Appl Sci* 9(24):1–18

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Giuseppe Placidi<sup>1</sup>**  · **Danilo Avola<sup>2</sup>** · **Luigi Cinque<sup>2</sup>** · **Matteo Polsinelli<sup>1</sup>** · **Eleni Theodoridou<sup>1</sup>** · **João Manuel R. S. Tavares<sup>3</sup>**

Danilo Avola  
avola@di.uniroma1.it

Luigi Cinque  
cinque@di.uniroma1.it

Matteo Polsinelli  
matteo.polsinelli@graduate.univaq.it

Eleni Theodoridou  
eleni.theodoridou@graduate.univaq.it

João Manuel R. S. Tavares  
tavares@fe.up.pt

<sup>1</sup> A2VI-Lab, Department of Life, Health & Environmental Sciences, University of L'Aquila, 67100, L'Aquila, Italy

<sup>2</sup> Department of Computer Science, Sapienza University, 00198, Rome, Italy

<sup>3</sup> Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, 4200-465, Porto, Portugal