Metaverse 2024, 5(2), 3020.
https://doi.org/10.54517/m3020

Article

Integration of Leap Motion sensor with camera for better performance in hand tracking

Khadijeh Mahdikhanlou, Hossein Ebrahimnezhad*

Computer Vision Research Lab, Department of Electrical Engineering, Sahand University of Technology, Tabriz 51, Iran
*** Corresponding author:** Hossein Ebrahimnezhad, ebrahimnezhad@sut.ac.ir

CITATION

Mahdikhanlou K, Ebrahimnezhad H. Integration of Leap Motion sensor with camera for better performance in hand tracking. Metaverse. 2024; 5(2): 3020.
https://doi.org/10.54517/m3020

ARTICLE INFO

Received: 22 October 2024
Accepted: 27 November 2024
Available online: 10 December 2024

Abstract: In this paper, we propose a framework for hand tracking in human-computer interaction applications. Leap Motion is used today as a popular interface in virtual reality and computer games. In this study, we evaluated the merits and drawbacks of this device. The limitations of this device restrict the user's free movement. The purpose of this study is to find an optimal solution for using Leap Motion. We propose a framework to estimate the pose of the hand in a bigger space around Leap Motion. Our framework uses an integration of Leap Motion with a camera that are placed in two different places to capture the information of the hand from various views. The experiments are designed based on the common tasks in the human-computer interaction applications. The finding of this study demonstrates that the proposed framework increases the precise interaction space.

Keywords: Leap Motion; hand tracking; human-computer interaction; 3D hand pose dataset; deep learning

1. Introduction

Advances in computer vision and machine learning have increased the use of hand gestures as a new form of interface in different fields. Converting hand gestures from sign language to other content, such as text or speech, provides more media for the deaf community. Deaf people use their hands to communicate with others. These hand gestures are called sign language and are learned by deaf individuals. The majority of hearing people are unfamiliar with sign language, so deaf people are isolated from society. In order to address this problem, researchers are developing hand gesture and sign language recognition systems to translate gestures from video to text or speech. So, these systems help deaf communities to access various media [1,2]. Hand gestures can play an important role in providing more engaging and enjoyable experiences for users of computer games. In computer-aided design applications, users can interact with objects and computer models using their hands more intuitively and naturally [3–5]. Human-robot interaction is another important field where humans can control a remote robot by hand gestures [6,7].

Hand-based interfaces can be classified coarsely into two groups, including wearable sensor-based methods and vision-based methods [1,8]. Vision-based systems include RGB or Infra-Red (IR) depth cameras with markerless or marker-based methods [3,5]. IR sensors can provide spatial information, but they are sensitive to light and noise. The main issue in the RGB cameras is the spatial information, and the occlusion is the problem of both types of cameras.

Leap Motion was released in 2016 for hand tracking and today is exploited extensively in various technologies involving virtual reality [9]. The popularity of

Leap Motion raises concerns about the accuracy of its measurements. In this paper, we exploit Leap Motion and a camera to estimate the 3D hand pose for human-computer interaction applications. Our study begins with an analysis of the limitations of Leap Motion. The results of the conducted experiments show that Leap Motion can work effectively only in a limited space, while human-computer interaction tasks require a larger space for users to move their hands around freely. This paper is an effort to increase the space of interfacing. For this purpose, we propose to use an integration of an RGB camera with Leap Motion. We use a deep learning-based architecture that estimates the 3D hand pose from a single RGB image captured from the camera. For training the proposed system, we created a large and challenging dataset that contains more than 50,000 real RGB images. In order to collect the dataset, a camera was placed next to Leap Motion, and a user performed different hand gestures near Leap Motion. The 3D coordinates of the hand's joints are provided by Leap Motion. The captured images from the camera are used as the input for the deep learning architecture, and the corresponding Leap Motion information is used as the ground truth. The samples in which the hand positions are computed wrongly by Leap Motion are removed from the dataset. We designed an experiment to show that the RGB image-based method is not able to distinguish the precise direction of the hand in some cases. Leap Motion can detect the overall direction of the hand precisely because of embedded infrared LEDs. Therefore, in human-computer interaction applications, we can use Leap Motion to detect the hand location, 3D coordinates of the hand's joints, and direction of the hand in 3D space, and the proposed deep learning-based architecture is used to estimate the hand pose in cases where the hand is occluded relative to Leap Motion.

Our main contributions are summarized as follows:

- We studied the limitations of Leap Motion related to the zone of the interaction.
- We evaluated the measurements of Leap Motion without comparing them with a reference device.
- We combine Leap Motion with a camera to increase the accurate zone of interfacing in human-computer interaction applications.
- We used a new and indirect way to annotate the 3D coordinates of the joints on the RGB hand image.

The paper consists of six sections. Section 2 reviews the related research works. Section 3 describes the proposed system. Experimental results are reported in section 4. Section 5 discusses the paper. Finally, the conclusion is presented in section 6.

2. Related work

In this section, we review studies that evaluated the accuracy of Leap Motion. We also mention existing datasets for hand pose estimation.

2.1. Leap Motion measurements

Razo et al. [10] compared the measurements of Leap Motion with a coordinated measuring machine (CMM) as the reference device. They created a cylinder that simulated the finger. The cylinder was moved across a set of marked points, and then its coordinates were measured by Leap motion and CMM. Their findings

demonstrated that the mean error of Leap Motion is 9.6 millimeters. Oropesa et al. [11] placed a laparoscope at marked positions inside a box and measured its coordinates by Leap Motion. In their study, four experiments were conducted to test the dynamic, static, long, and short-term precision of Leap Motion. The obtained results showed that the static precision of Leap Motion, both in the long term and short term, is less than 2.5 mm, and the dynamic precision ranges between 2 and 15 mm. Quesada et al. [12] evaluated the precision of Leap Motion through recognizing the sign language gestures. In their experiment, the participants performed sign language alphabets in two different places relative to Leap Motion. They evaluated the number of attempts that users performed until the device recognized the gesture correctly. The results were compared with those obtained by Intel RealSense as another device for hand tracking. In their study, the Support Vector Machine (SVM) was used to classify the signs, but it is not clear which features were extracted from the input. Mahdikhanlou et al. [1] combined the information of Leap Motion with a set of hand-crafted features extracted from RGB images for recognizing gestures in sign language. Ponraj et al. [13] fused Leap Motion and flex sensors to compensate for the errors of Leap Motion. They measured the coordinates of the fingertips at the thumb, middle, and index fingers. In their study, the participants performed four gestures. At first, the fingertips faced directly toward Leap Motion and gradually turned away from it until they were entirely occluded by the side of the hand. Guna et al. [14] selected 37 reference locations above and around Leap Motion sensory space for performing a static measurement. For static evaluation, a plastic arm model simulating a human hand was used, and for dynamic measurement, they moved a V-shaped tool above Leap Motion and measured the distance between two points of the detected tool. According to their results, in the static experiments, the standard deviation was less than 0.5 mm, but there was a significant increase in the standard deviation when moving away from Leap Motion. In their dynamic scenario, a significant drop in accuracy was reported for samples taken from more than 250 mm above Leap Motion space. Mahdikhanlou et al. [3] studied the occlusion issue arising from approaching two hands where Leap Motion fails to detect one or both hands. Ameur et al. [15] mentioned the effect of orientation on the performance of Leap Motion in non-desk situations. They studied temporal information existing in the sequential hand gesture in time series. Jiang et al. [16] investigated the occlusion problem of Leap Motion in a virtual grasp application. They combined signals acquired from force myography (FMG), a muscular activity-based hand gesture recognition device, with data from Leap Motion. In their experiments, participants performed the act of grasping virtual objects with virtual reality goggles on their heads, an FMG band on their wrist, and a Leap Motion positioned either on the desk or on the goggles. Their gestures involve six grasping interactions with small virtual objects, including pinching a ball using different numbers of fingers (G1–G3), lateral key-pinching a thin disk (G4), and hand-wrapping a slender cylinder (G5 and G6). Wang et al. [17] used multiple Leap Motions to enlarge the interaction space. They used five Leap Motions, and both hands were involved in their studies. In their setting, the front Leap Motion was set to be the reference camera. In their experiments, users were asked to move their hands very slowly (less than 10 millimeters per second). Then, the difference between measurements conducted by the central Leap Motion and the lateral Leap Motions in

the overlapping tracking area was reported. They simplified the hand-tracking problem to a palm-joint tracking problem. All samples in their experiments were captured from an egocentric view.

2.2. Datasets for 3D hand pose estimation

Existing datasets in the field of hand pose estimation can be classified based on the data type, dimension, existence of an object, and viewpoint. Some datasets provide hand poses only in two dimensions [18]. Datasets can be collected from an egocentric view (first-person) or a third-person view, and they exist in RGB format or depth type [19–22]. Egocentric datasets are mostly used in AR/VR applications where the camera is mounted on the headset. EgoHands contains 48 videos of two people sitting face to face and playing different games such as cards and chess [23]. Garcia et al. [22] created an egocentric RGB-D dataset aimed at recognizing daily activities in the kitchen. Other studies, including one by Mueller et al. [19] and one by Rogez et al. [24], have focused on the interaction between the hand and the objects in egocentric view. The three-dimensional pose of the hand can be computed by multi-view images [20]. A wide range of studies has used depth information of the hand for hand pose estimation [25]. Compared with depth-based systems, collecting datasets for 3D RGB-based hand pose estimation systems is more challenging due to the annotation issue. Some depth datasets have been collected and annotated by attaching tiny bending sensors to the hand so that they do not influence the depth [22]. Zimmermann and Brox collected a large synthetic dataset [21]. Cai et al. [26] used weakly labeled RGB-D images. Then, they computed an adaptation between RGB-D and synthetic images.

3. Proposed method

In order to replace a mouse with hand gestures, we need to capture both the pose and the location of the hand. Spatial information of the hand is required to navigate the pointer on the desktop, and each pose of the hand can be interpreted as a special intention of the user. Leap Motion contains three infrared LED sensors that are used for finding the location of the hand in space. Three-dimensional coordinates of the hand's joints can be computed by Leap Motion's built-in SDK. However, the results of the experiments show that Leap Motion is not reliable in distinguishing the hand's small components, such as joints, at far distances and in some angles of view. So, we propose a deep learning-based network for 3D hand pose estimation from the images captured by a camera placed at a distance farther away from Leap Motion. **Figure 1** shows the places of Leap Motion and the camera in the proposed system. The camera in **Figure 1** is shown in red color. The overall location and direction of the hand are computed by Leap Motion, and then, when the distance or direction of the hand is larger than a threshold, the hand pose is computed by the deep learning-based architecture.

Figure 1. The positions of Leap Motion and the camera in the proposed interaction space. The camera is depicted in red color.

3.1. Dataset creation

In this study, we collected a big dataset. For this purpose, a camera was placed next to Leap Motion, and then the participants performed a set of gestures around Leap Motion. The information provided by Leap Motion is used as the ground truth for the captured image by the camera. Each time, the user maintained their hands in a special gesture and moved their hand around Leap Motion to capture the data at different distances and different views. In other words, the positions of the hand's joints relative to each other were constant during a sequence of frames. For each sequence, the first few frames were investigated visually to determine whether Leap Motion recognized the pose of the hand correctly or not, and then one of them was selected as the reference frame for the sequence. The angles of the joints in all the frames were compared to the reference frame, and the samples in which the difference between the joints and the corresponding joints in the reference frame was bigger than a threshold were removed from the dataset.

3D hand pose estimation needs a large amount of data due to the complex structure of the hand and its ability to perform various poses. Furthermore, deep learning models are prone to the overfitting issue. Overfitting occurs when the model achieves high accuracy on the training data while it does not generalize well on new and unseen data [27]. One of the best ways to reduce overfitting is to train the model on a large and diverse dataset. The human hand is an articulated object with high degrees of freedom composed of six parts (fingers and palm) that can occlude each other easily. Images of the same hand gesture from diverse viewpoints are quite different. So, the complex anatomy of the hand and its self-occlusion make it very challenging to extract the 3D pose of the hand. In this paper, we collected a large dataset of hand images, and some techniques were employed to prevent overfitting. The dataset samples have been collected from eight participants. Each participant performed between 27 and 44 gestures. 21 joints have been marked on each image. The dataset covers multiple hand shapes, self-occlusion, and articulations. All the images were captured from the right hand. So, the images and the corresponding ground truth were flipped in order to generate samples for the left hand. We also rotated all the images and their ground truth from 10° to 350°.

3.2. 3D hand pose estimation from a single RGB image

Given an RGB image $I \in R^{300\times300\times3}$ showing a single hand, we want to estimate its three-dimensional pose. The 3D pose of a hand is defined as a set of coordinates $p_i = (x_i, y_i, z_i)$ that denotes the coordinates of the i th joint of the hand. In this paper, the pose of the hand is defined by 21 joints ($i \in [1, 21]$).

The proposed architecture for hand pose estimation consists of two parts (See **Figure 2**). The first part is an encoder-decoder network that produces 21 heat maps, each of them belongs to one of the 21 joints. The value of each pixel on the heat map shows the probability of locating the 2D coordinates of the joint at that pixel. We use the Mean Square Error (MSE) between the ground truth heat map Φ_{WH}^{gt} and the estimated heat map Φ_{WH}. The loss of the network is defined as Equation (1) where W and H are the width and height of the heap map, respectively. The ground truth is defined as a Gaussian heat map with a standard deviation $\sigma = 1$.

$$Loss_{heatmap} = \sum_{W} \sum_{H} \left(\Phi_{WH} - \Phi_{WH}^{gt}\right)^2 \quad (1)$$

The second part of the proposed network is a CNN that determines the type of hand in the image. Three-dimensional hand pose estimation from the two-dimensional joints is an ill-posed problem. It can extract at least two hand poses for a set of 2D joints. So, we must provide more information for the network to interpret only a unique 3D pose for a given image. We propose to use the information about the type of hand in the image. So, the image of the hand is first classified to determine whether the given image belongs to the left hand or the right hand.

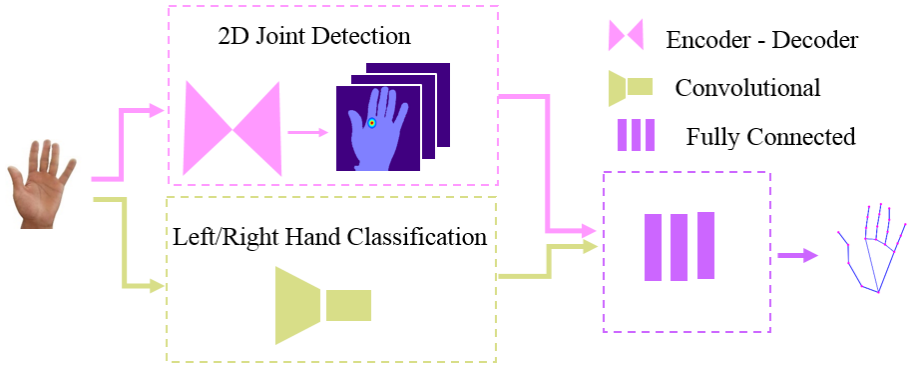

Figure 2. The proposed architecture for estimating the 3D hand pose from an RGB image.

4. Experiments

We conducted a set of experiments to evaluate different parts of this study. The first experiment investigates the effect of the distance of the hand and the field of view on the reliability of Leap Motion. The second experiment indicates that the absence of depth information in the RGB images is what causes errors in the direction of the hand. The last experiment was developed based on tasks that are prevalent in human-computer interaction applications, such as translating objects.

4.1. Reliability of Leap Motion

Figure 3 shows Leap Motion's scope of vision. In order to assess the reliability of Leap Motion, the participants moved their hands above Leap Motion while keeping their hands in a special gesture. During this experiment, the users were asked to keep the angle of the joints constant. The participants performed each task for 48 seconds, and a total of 9408 frames were captured. For determining the ground truth of each sequence, some of the frames in the sequence were investigated visually, and then one of them that was very similar to the performed gesture was selected as the ground truth. In each frame, the position of the palm and the angles at the joints were recorded and compared with the ground truth. In each sample, if the difference between the recorded angle and the ground truth at least at one of the joints is more than 60°, the sample is considered a falsely recognized gesture. The angle of the hand's view is computed as Equation (2), where z is the height of the hand (see **Figure 4**). Correctly recognized samples are illustrated in green color in **Figure 5**. The red points in **Figure 5** show the samples that were recognized falsely. **Figure 5a** indicates that false recognitions by Leap Motion have mostly occurred in heights larger than 400 millimeters and angles bigger than 45 degrees, and **Figure 5b** shows that by combining a camera and Leap Motion, the rate of false recognitions decreases in this area.

$$r = \sqrt{x^2 + y^2 + z^2} \tag{2}$$

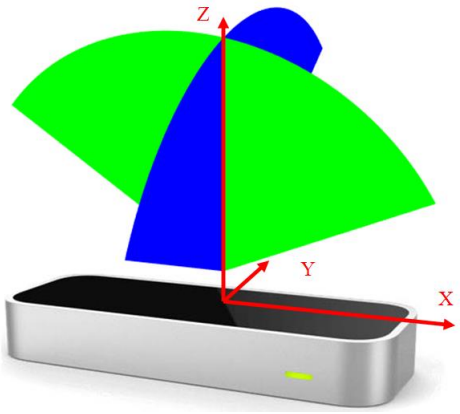

Figure 3. Vision angles of Leap Motion in *x* (green) and *y* (blue) directions.

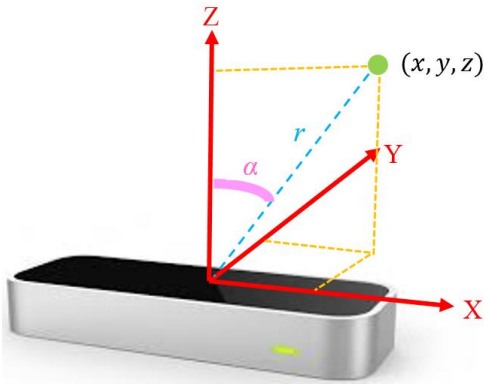

Figure 4. Computing of the angle between Leap Motion and the hand. The green point denotes the position of the palm.

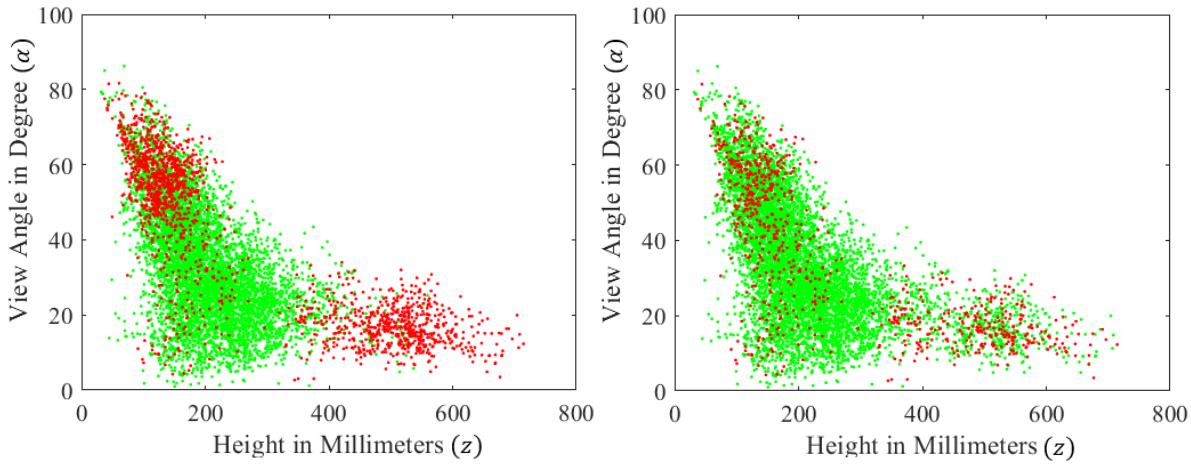

Figure 5. (a) The results of the hand gesture recognition for different heights and angles of view by Leap Motion, and **(b)** integration of Leap Motion with a camera.

4.2. Image Information for computing the direction of the hand

In three-dimensional space, the direction of the palm is determined by the blue vector illustrated in **Figure 6a**, and it is computed as the normal vector of the plane that passes through three blue points in **Figure 6b**. The red vector can be considered as the direction of the wrist, and it is equivalent to the direction of a part of the middle finger depicted by a red line in **Figure 6b**. One of the cases where the lack of depth information causes error is when the palm direction is perpendicular to the camera view. In this situation, if the hand is rotated around the yellow color vector (cross product of the red and blue vectors) gradually, the RGB pattern does not change very significantly, and consequently, the precise direction of the palm or wrist is not perceived by an image-based network such as the proposed deep learning-based architecture. **Figure 7** shows a sequence of hand images in which the palm direction is changing from left to right. The patterns of the leftmost and rightmost samples are very different. So, the image-based network can distinguish them from each other easily. However, the changes in the image pattern in every two successive samples are not very sensible for the network. **Figure 8** shows the ground truth and the predicted direction of each sample using image information and Leap Motion information. For computing the ground truth direction, the direction of the hand was computed using samples that were captured from a different view. It is clear from **Figure 8** that the image-based network has failed to recognize the correct direction of the hand in the middle samples. The smooth changes of the ground truth (red curve) have not appeared in the predicted direction using the image information (green curve). The image-based network has discriminated the different samples very roughly such that a sharp drop has resulted in the curve.

(a) **(b)**

Figure 6. (a) The direction of the hand can be determined by red and blue vectors; **(b)** the red part of the skeleton in (b) can be considered as the red vector in (a), and the normal vector of the plane passing through three blue points in (b) can be considered as equivalent to the blue vector in (a).

Figure 7. In a sequence of hand images, from left to right, the direction of the palm is changing from 90° to 30°, but the change in the pattern of the successive images is not very perceptible.

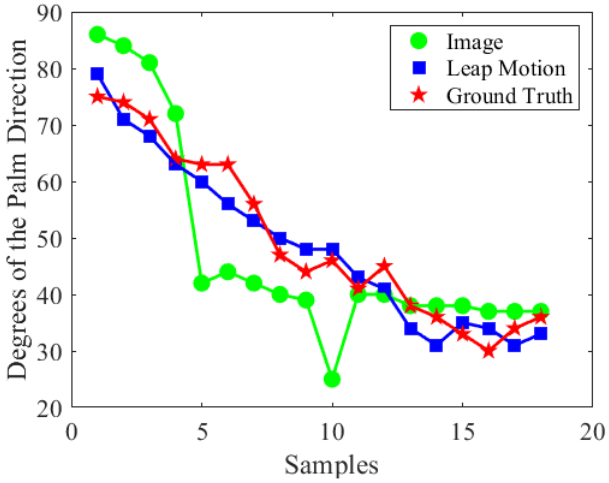

Figure 8. Ground truth and the predicted direction of the palm for samples shown in **Figure 7**.

A user study was carried out to clarify the issue. In this study, the users were asked to maintain their wrists in a fixed location and plot a quarter-circle through rotating their hands around the wrist (see **Figure 9**). In the beginning, the direction of the palm was perpendicular to the view of the camera. The task was repeated by different participants and for different gestures. The samples in which the location of the wrist has been changed were removed from the evaluation, and a total of 100K samples were collected. The image of the hand is captured by two cameras from two different views. **Figure 9** shows the location of the two different cameras in red and green colors. We expect that the direction computed from the view of the red camera

will be more precise; then, the information from the red camera is considered the ground truth. The direction of the palm is computed by Leap Motion and image-based networks separately for each sample. The number of samples per angle is counted and shown by a color spectrum in **Figure 10**. As the figure indicates, the predicted directions of the palm in different samples by image information are not distributed uniformly, and most of them are focused on very small or very large angles.

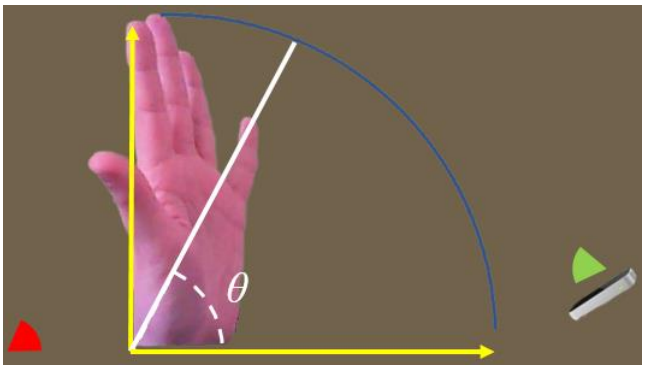

Figure 9. The users had to plot a quarter-circle through rotating their hand around the wrist in front of a camera (green) and Leap Motion. The ground truth direction of the samples is computed using the images that are captured by the red camera.

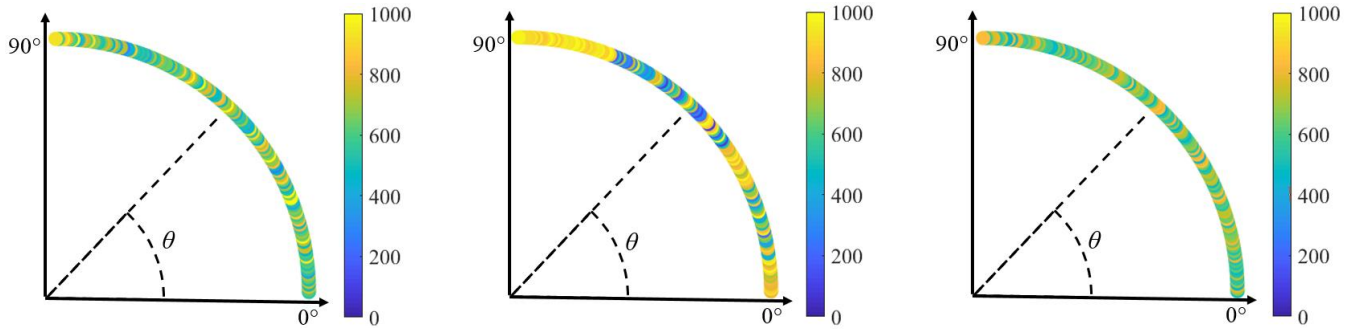

Figure 10. The computed direction of the palm using images captured from **(a)** the red camera; **(b)** image information captured from the green camera and; **(c)** by Leap Motion.

4.3. Integration of Leap Motion and the camera in a user study

In this section, a user study is described that is designed to compare the performance of the proposed framework and Leap Motion. The users were asked to perform a gesture and simultaneously move a pointer on a set of curves shown in **Figure 11**. The pointer is moved based on the coordinates of the index fingertip. When the distance between the index fingertip and the curve is less than a threshold, the angles at the hand's joints are recorded and compared with the ground truth. **Figure 11** shows the results of the user study. Each participant repeated the task twice, and the number of correct gestures at each point of the curve was counted and is shown by the color spectrum.

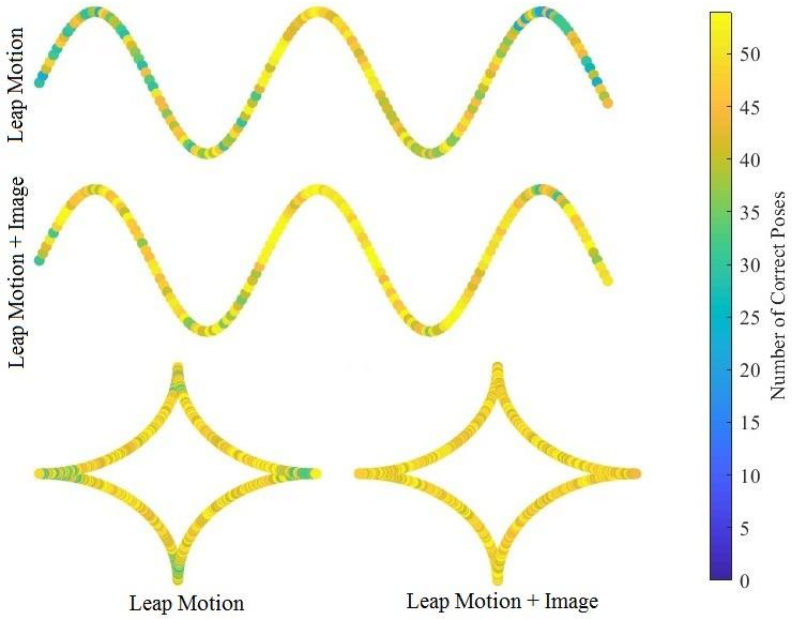

Figure 11. Results of the user study. Users had to perform a set of gestures and at the same time move a pointer through the curve. At each point, the histogram of the correctly recognized gestures using the information of Leap Motion and using the combination of the information of the image with Leap Motion is recorded.

5. Discussion

The main focus of this paper was to design a framework for human-computer interfacing applications through the integration of Leap Motion with a camera. The camera is located in a different location farther away from Leap Motion and captures images of the hand from a different view. By fusing a camera into the interaction space, we alleviated the limitations of Leap Motion and expanded the user's freedom during interfacing with a desktop. The built-in algorithm of Leap Motion has not been published, but we investigated its limitations by conducting a set of experiments. In some studies [10,11], the measurements of Leap Motion were evaluated by comparing them with a reference device. In these studies, measurements are limited to being done on a set of single points. In this paper, the users performed a number of gestures in different places above Leap Motion, and the capability of Leap Motion was investigated by measuring the stability of the measures in different places and different views. Another issue that we dealt with was the creation of a dataset for 3D hand pose estimation from a single RGB image. Annotating 3D coordinates on the RGB images is not possible. Some researchers created synthetic datasets. Some of the studies used stereo images to compute the 3D pose of the images. We used an indirect method to compute the 3D pose of the images. We placed a camera next to Leap Motion and collected images and their 3D coordinates of the joints; then, we removed samples where Leap Motion computed the joints wrongly. All the samples in our dataset are real. The results of the first experiment can matter for designers who exploit Leap Motion in human-computer interaction fields. Oropesa et al. [11] conducted their experiments in a very simple environment (a simple box). The background color was monochromatic, fixed, and without any pattern. The scenario lacked the inclusion of more complex elements. Leap Motion tracked only a simple instrument with a single

color. Their instrument is much simpler than the anatomy of a hand. In our tests, there were users with different colors of clothes, and the environment was more complex. In the study conducted by Oropesa et al. [11], for the dynamic experiment, the instrument was moved only 100 mm along the Z-axis, but we evaluated the performance of Leap Motion in the range of 700 mm along the Z-axis. Quesada et al. [12] considered 26 American Sign Language (ASL) alphabets for evaluation. They performed gestures in two positions to determine whether Leap Motion was able to recognize all signs. They performed signs on top of Leap Motion, which is very close to our experiment. We compare our results in this position. In fist-like gestures, data provided by Leap Motion is not very reliable. Letters in ASL, including "A", "M", "N", "O", "P", "S", and "T" are very similar, and the only difference among them is the position of the thumb, which is out of the Leap Motion vision range. Leap Motion cannot estimate the coordinates of the thumb joints precisely in these gestures (**Table 1**). **Table 2** shows a comparison between our results and the results of the study conducted by Jiang et al. [16].

Table 1. Comparison of our results with results of experiments conducted by Oropesa et al. [12].

Letter	Oropesa et al. [12]	Ours	Letter	Oropesa et al. [12]	Ours	Letter	Oropesa et al. [12]	Ours
A	No	Yes	J	No	Yes	S	No	Yes
B	No	Yes	K	No	Yes	T	No	Yes
C	No	Yes	L	Yes	Yes	U	Yes	Yes
D	Yes	Yes	M	No	Yes	V	Yes	Yes
E	No	Yes	N	No	Yes	W	Yes	Yes
F	Yes	Yes	O	Yes	Yes	X	No	Yes
G	Yes	Yes	P	No	Yes	Y	Yes	Yes
H	Yes	Yes	Q	No	Yes	Z	Yes	Yes
I	No	Yes	R	Yes	Yes			

Table 2. Comparison of our results with results of experiments conducted by Jiang et al. [16].

	G1	G2	G3	G4	G5	G6
Jiang et al. [16]	94	92	95	93	97	90
Ours	98.03	99.45	99.34	99.00	99.23	99.50

All experiments were conducted under normal room lighting conditions. Leap Motion warns in poor lighting conditions. Further research is needed to investigate the effect of the light condition on the performance of Leap Motion. The importance of these studies is due to the embedded LEDs in Leap Motion that are sensitive to the light of the environment. The scope of our study is limited to the presence of a single hand in the interaction space. Interfacing with both hands can raise more challenges. Our dataset's samples have been collected from a third view, and the results of the experiments cannot be generalized to the egocentric view applications.

6. Conclusion

The goal of this study was to improve the performance of Leap Motion for human-computer interaction applications. For this purpose, we first design different experiments to investigate the limitations of Leap Motion. Conducted experiments showed that Leap Motion can perform perfectly when the hand is located at a distance between 100 mm and 400 mm with a view angle of less than 45°. The human-computer interaction applications involve tasks that require larger movements of the user's hand. By combining the information of Leap Motion and the proposed deep learning-based architecture, we provided a larger space for users to interact with the computer. We proposed to use a camera in addition to Leap Motion and estimate the 3D pose of the hand using a deep learning architecture. The camera captures the image of the hand from a different view. The results show that combining the information of Leap Motion and a camera can lead to achieving a more accurate estimation of 3D hand pose at farther distances. However, the overall direction of the hand and palm can be computed more precisely using Leap Motion. In other words, the pattern of the RGB image is not informative enough to estimate the global direction of the hand, but Leap Motion can compute the distance and the overall direction of the hand due to containing three Infra-Red LEDs. In comparison with other works, we trained the deep learning model with a real dataset containing samples in various directions. Improvements could be made both in the dataset and deep learning architecture. The proposed deep learning was trained on the dataset collected from the third view. In the future, the deep learning model can be trained on egocentric view datasets.

Author contributions: Conceptualization, KM and HE; methodology, KM; software, KM; validation, KM, HE; formal analysis, KM; investigation, KM; resources, KM; data curation, KM; writing—original draft preparation, KM; writing—review and editing, KM, HE; visualization, KM; supervision, HE; project administration, HE. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References

1. Mahdikhanlou K, Ebrahimnezhad H. Multimodal 3D American sign language recognition for static alphabet and numbers using hand joints and shape coding. Multimedia Tools and Applications. 2020; 79(31–32): 22235–22259. doi: 10.1007/s11042-020-08982-8
2. Elakkiya R, Selvamani K. Subunit sign modeling framework for continuous sign language recognition. Computers & Electrical Engineering. 2019; 74: 379–390. doi: 10.1016/j.compeleceng.2019.02.012
3. Mahdikhanlou K, Ebrahimnezhad H. Object manipulation and deformation using hand gestures. Journal of Ambient Intelligence and Humanized Computing. 2021; 14(7): 8115–8133. doi: 10.1007/s12652-021-03582-2
4. Feng Z, Yang B, Tang H, et al. Behavioral-model-based freehand tracking in a Selection-Move-Release system. Computers & Electrical Engineering. 2014; 40(6): 1827–1837. doi: 10.1016/j.compeleceng.2014.05.014
5. Chattaraj R, Khan S, Roy DG, et al. Vision-based human grasp reconstruction inspired by hand postural synergies. Computers & Electrical Engineering. 2018; 70: 702–721. doi: 10.1016/j.compeleceng.2017.10.018
6. Yongda D, Fang L, Huang X. Research on multimodal human-robot interaction based on speech and gesture. Computers & Electrical Engineering. 2018; 72: 443–454. doi: 10.1016/j.compeleceng.2018.09.014
7. Cui Y, Song X, Hu Q, et al. Human-robot interaction in higher education for predicting student engagement. Computers and Electrical Engineering. 2022; 99: 107827. doi: 10.1016/j.compeleceng.2022.107827

8. Gupta R, Kumar A. Indian sign language recognition using wearable sensors and multi-label classification. Computers & Electrical Engineering. 2021; 90: 106898. doi: 10.1016/j.compeleceng.2020.106898
9. Połap D, Kęsik K, Winnicka A, et al. Strengthening the perception of the virtual worlds in a virtual reality environment. ISA Transactions. 2020; 102: 397–406. doi: 10.1016/j.isatra.2020.02.023
10. Curiel-Razo YI, Icasio-Hernández O, Sepúlveda-Cervantes G, et al. Leap motion controller three dimensional verification and polynomial correction. Measurement. 2016; 93: 258–264. doi: 10.1016/j.measurement.2016.07.017
11. Oropesa I, de Jong TL, Sánchez-González P, et al. Feasibility of tracking laparoscopic instruments in a box trainer using a Leap Motion Controller. Measurement. 2016; 80: 115–124. doi: 10.1016/j.measurement.2015.11.018
12. Quesada L, López G, Guerrero L. Automatic recognition of the American sign language fingerspelling alphabet to assist people living with speech or hearing impairments. Journal of Ambient Intelligence and Humanized Computing. 2017; 8(4): 625–635. doi: 10.1007/s12652-017-0475-7
13. Ponraj G, Ren H. Sensor Fusion of Leap Motion Controller and Flex Sensors Using Kalman Filter for Human Finger Tracking. IEEE Sensors Journal. 2018; 18(5): 2042–2049. doi: 10.1109/jsen.2018.2790801
14. Guna J, Jakus G, Pogačnik M, et al. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. Sensors. 2014; 14(2): 3702–3720. doi: 10.3390/s140203702
15. Ameur S, Ben Khalifa A, Bouhlel MS. Chronological pattern indexing: An efficient feature extraction method for hand gesture recognition with Leap Motion. Journal of Visual Communication and Image Representation. 2020; 70: 102842. doi: 10.1016/j.jvcir.2020.102842
16. Jiang X, Xiao ZG, Menon C. Virtual grasps recognition using fusion of Leap Motion and force myography. Virtual Reality. 2018; 22(4): 297–308. doi: 10.1007/s10055-018-0339-2
17. Wang Y, Wu Y, Jung S, et al. Enlarging the Usable Hand Tracking Area by Using Multiple Leap Motion Controllers in VR. IEEE Sensors Journal. 2021; 21(16): 17947–17961. doi: 10.1109/jsen.2021.3082988
18. Jawahar CV, Li H, Mori G, et al. Computer Vision—ACCV 2018. Springer International Publishing; 2019.
19. Mueller F, Bernard F, Sotnychenko O, et al. GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In: Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18–23 June 2018; Salt Lake City, USA. pp. 49–59.
20. Zhang J, Jiao J, Chen M, et al. A hand pose tracking benchmark from stereo matching. In: Proceedings of 2017 IEEE International Conference on Image Processing (ICIP); 17–20 September 2017; Beijing, China.
21. Zimmermann C, Brox T. Learning to Estimate 3D Hand Pose from Single RGB Images. In: Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV); 22–29 October 2017; Venice, Italy.
22. Garcia-Hernando G, Yuan S, Baek S, et al. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In: Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18–23 June 2018; Salt Lake City, USA.
23. Bambach S, Lee S, Crandall DJ, et al. Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In: Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV); 7–13 December 2015; Santiago, Chile. pp. 1949–1957.
24. Rogez, G., Khademi, M., Supančič, J.S., et al. 3D Hand Pose Detection in Egocentric RGB-D Images. In: Agapito, L., Bronstein, M., Rother, C. Computer Vision - ECCV 2014 Workshops. Lecture Notes in Computer Science, vol 8925. Springer International Publishing; 2015.
25. Yuan S, Ye Q, Stenger B, et al. BigHand2.2M Benchmark: Hand Pose Dataset and State of the Art Analysis. In: Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 21–26 July 2017; Honolulu, USA.
26. Cai, Y., Ge, L., Cai, J., et al. Weakly-Supervised 3D Hand Pose Estimation from Monocular RGB Images. In: Ferrari, V., Hebert, M., Sminchisescu, C., et al. Computer Vision – ECCV 2018. Lecture Notes in Computer Science, vol 11210. Springer International Publishing; 2018.
27. Bengio Y, Goodfellow I, Courville A. Deep learning. MIT press; 2017.