Analysis and Evaluation of Gesture Recognition using LeapMotion

Pedro Miguel Oliveira Leitão

Student of Doctoral Program of Informatic Engineering
Faculty of Engineering, University of Porto
Porto, Portugal
pedromoleitao@gmail.com

Abstract Nowadays are emerging increasingly natural interaction devices, which use human body as a natural way of interacting with applications. To correctly interact with natural interfaces, also named NUI, there is a need to improve the recognition and performance evaluation of different gestures simultaneously in order to identify which configurations between gestures and settings best fit to get a greater efficiency on their recognition. The evaluation was performed based on real gestural attempts with two participants. Finally, the application got a gesture recognition average rate of 86.1% using only the minimum resources provided by the LeapMotion device.

Keywords: Natural User Interface, LeapMotion, Gesture Recognition

1 Introduction

The increasing interest for new interaction paradigms, combined with new emerging technologies, are originating new Natural User Interface (NUI) devices on the market.

Recent devices are aiming to bridge some existing limitations on human-machine interaction through gestures performed with hands, fingers and drawing tools. This kind of devices, such as LeapMotion, which will be described in the next section, are interesting to enthusiastic public and all community due to its simplicity, efficiency and numerous areas where it can be applied. As gestures have distinct characteristics, such as the way and even how fast movements are performed.

On work produced by Sharad Vikram, Lei Li and Stuart Russel [1] authors present an interface of online recognition, of gestures using NUI interaction, performing a very precise interpretation which makes it ideal to drawn words in real-time.

Over time are appearing different techniques associated to Human-Computer-Interaction, based on computational real-time vision, as described by Eshed Ohn-Bar and Mohan Manubhai [2]. They propose a robust system based on natural interaction, to recognize signs in real-time inside a vehicle. These devices used inside vehicles enable a decrease of driver visual charge, driving mistakes and have a high level of adaptation and usability by users.

This paper is organized in the following way: in sections 2 and 3 some NUI devices will be presented. They are equivalent to LeapMotion, but their functioning rely on different available technologies. In section 4 all work done and technical details, about it, will be described. In section 5 all evaluations related to this project will be presented. Finally, on sections 6 and 7, this paper will be concluded and presented some perspectives about future work are discussed.

2 LeapMotion

LeapMotion company was founded in 2010 by Michael Buckwald and David Holz. Working at nearly 300 frames per second, this device (Figure 1) has the capacity to collect hands movements simultaneously, with precision higher than 0.01mm [3].

Figure 1 – LeapMotion device [7].

2.1 Hardware

This device consists of 2 Infrared (IR) monochrome cameras and 3 IR LEDs. As Microsoft Kinect LED, they project a pattern of points along the area which will be captured by IR cameras, collect all data and transferring it to the software layer for analysis purposes. This software uses received data and generate a representation of mapped data, in a three dimensional space to compare with bi-dimensional frames and submit this bi-dimensional images, to an algorithm of edges detection [4].

In order to optimize interaction performance, process of transfer information over USB cable is submitted to a compression process in order to remove background light and unduly added noise. After that, three-dimensional data collected by IR sensors will be analysed and reconstructed for representation. Finally, to transfer information to LeapMotion API, an algorithm of tracking, looking for information about hands, fingers and tools representing them on three-dimensional and transferring this data into API layer responsible for communicate with high level software [5].

The biggest challenge of LeapMotion team still be the residual latency. This problem is related with captured images, at a moment that will suffer delay and will only show these images some milliseconds after the movement has completed.

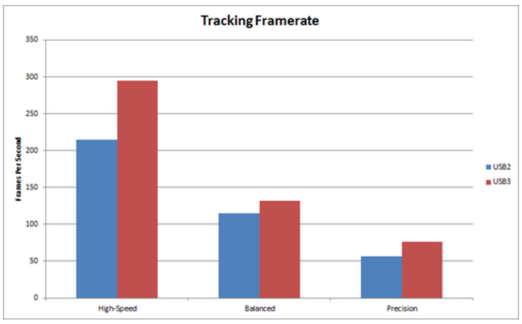
The better way to minimize this problem and improve processor response time should consider the follow settings [6]:

- Use USB3.0 cable to higher transfer rate;
- Use monitor with short response time;
- Initiate LeapMotion in 'High-Speed' mode.

2.2 Technical Details

The device LeapMotion, has 3 different operating modes presented on graphic (Graphic 1): 'High-Speed', 'Balanced' e 'Precision':

- The method 'High-Speed' is suitable for scanning fast movements. This mode increases a resolution of IR sensors, making them quadruple the number of captured images per second of data collection of movements.
- Next mode is 'Precision' mode, with this it is possible to decrease frame rate to 40% doing it lower than the normal value, without need to decrease resolution. It is ideal for capture of small movement variations.
- Finally, with 'Balanced' mode, it tries to reconcile recognition features as quickly and accurately, adapting referred modes in just one, balancing the resolution bandwidth and computational charge with the value of the frame rate.



Graphic 1 - Frame rates and operationing modes of LeapMotion using USB cables 2.0 and 3.0 [5].

One feature which should be considered is about data acquisition and what kind of cable used and processing capacity of used machine.

With USB cable 3.0, it is possible to transfer a higher data frame rate comparing with it is predecessor USB2.0. However, with this cable 3.0 which ensure higher transfer of data needs a high computer capacity to be able to process all received data, and use this higher amount of information to reduce of movements delay.

2.3 Planes of Interaction

To detect touch in deepness over a tri-dimensional interaction zone, 3 zones of detection are automatically defined (Figure 2). These 3 plans, represented with different colors, are bounding different zones of interaction.

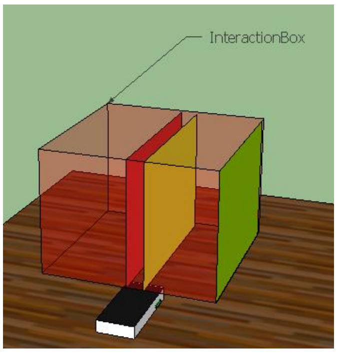


Figure 2 – Interaction zones to detecting touch on the areas.

With a selected object to interact with those zones and going forward over the interaction box, our object will find the following plans: 'None'(green), 'Hovering'(yellow) and 'Touching'(red) on (Figure 2). Starting at user position and entering into interaction box bounds, the first interaction zone is called 'None' bounded between 130 and 280 mm and is used to navigate over this bi-dimensional zone. Interaction zone 'Hovering' is between 70 and 130 mm is here where object navigator would be positioned over bi-dimensional coordinate which will be clicked. Last interaction zone is called 'Touching' settled between -100 and 70 mm and is used to seal action of touch in a coordinate selected on previous zone 'Hovering'.

2.4 Software

LeapMotion SDK is supported by Windows, Macintosh and Linux operating systems. First version of this SDK, offers applications to test as well as functions to calibrate device to required settings [8]. The second version of this SDK already exists on experimental version [9] which includes as main improvements the capability to represent all structure of a hands articulations simultaneously, more robustness to interference originated by sunlight and the implementation of new gestures such as pinch and grab objects [10].

3 Application Development

The scope of this project is intended to develop an evaluation application developed on .Net Framework 4, named Leap Tester composed by two interfaces to evaluate individual and a group of gestures, giving two types of analysis and evaluations. On first stage was started an individual analysis of each gesture type and some of their parameters. On second stage, was focus on a general evaluation of all gestures simultaneously, using better results of previous stage in order to get better results.

For each move, were implemented functions which instantiate device commands which could be integrated with different applications to associate this gesture command with different kinds of functions. We selected four types of gestures available with LeapMotion library, which use only one hand, right or left and create two new gestures one of them use two hands and the other with one hand right or left. The last ones were created to measure the recognition accuracy of gestures with greater movement amplitudes. Each function was associated with a device library gesture distinct type. So CLEAN move instance SWIPE command, CLICK DISPLAY is associated with SCREENTAP command, CIRCULAR move to both sides, right and left, are associated to CIRCLE command, at last CLICK function are related to KEYTAP command. The new type gestures implemented were APPROACH TWO HANDS and FIVE FINGERS.

3.1 Gestures Implementation

Of all recognizable signs embedded on LeapMotion library were modified the nature of SWIPE and CIRCULAR gestures in order to check their performance using different movement configurations (Figure 3).

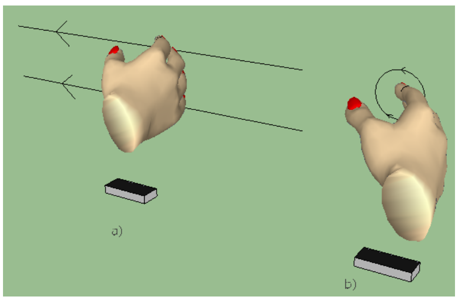


Figure 3 - a) SWIPE move performed horizontally from right to left. b) CIRCULAR move in both directions.

The SWIPE movement could be recognized as vertical or horizontal sliding move, using index finger of both hands. So to restrict this movement to only accept horizontal sweeps performed from right to left side. To bound this horizontal move absolute value of bi-dimensional vector generated by sweep move (1) on X axis should be bigger than vertically on Y axis. About movement direction, if value of D_x was bigger than 0 so, sign is produced on clockwise, otherwise if value is smaller than 0 movement is counter-clockwise.

$$H = |D_x| > |D_y| \quad (1)$$

About CIRCULAR movement, this move allows to do circular moves using both index fingers and production an circle with minimum radius of 5 mm and minimum arc length of 1.5 * π radians. To identify the way the circular movement was produced, calculating the angle of normal vector resultant of this move (2). In case of the angle value being less than 90º, movement was performed clockwise; otherwise it was performed counter-clockwise.

$$\frac{\mathrm{D} \cdot \mathrm{C}}{\|\mathrm{D}\| \, \|\mathrm{C}\|} \leq \frac{\pi}{2} \quad (2)$$

Beyond the referred modifications, new two gestures were implemented, the APPROACH TWO HANDS and FIVE FINGERS.

While first implementation of APPROACH TWO HANDS was configured to bound this gesture to be recognized when palm of hands being approached, at a distance less than 4 cm and vertically between them.

The gesture FIVE FINGERS is acknowledged through approximation of two hands of sensor in order to detect its fingers. As device don't know when should start to detect hand fingers, so when a hand is detected inside the interaction box, is performed immediately the function of move detection. This becomes inappropriate when we want the gesture to be submitted to another validation using another method. Based on this adversity, a function was developed to allow user to insert the hand inside the box. After inserting the hand a countdown variable is started performing the recognition of five fingers after elapsed time.

3.2 User Interface

Developed application have two different kind of interfaces to allow participants to make their experiments. This application is split into two groups (Figure 4): evaluation interface with gestures parameterization and in other hand, the interface of test generation with all gestures simultaneously.

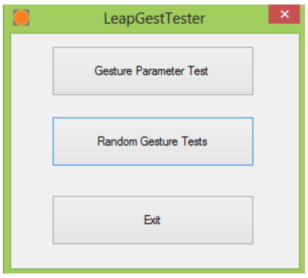


Figure 4 – Selection menu of evaluator.

On first group (Figure 5), user can select any gesture and change their parameters, generating values between a minimum and a maximum values defined by him. When each gesture is recognized, user needs to remove his hand from LeapMotion interaction area to avoid repeated moves and click over 'Next Gesture' button to get another attempt. This procedure is repeated about 50 trials, where user should answer correctly to all requested gestures.

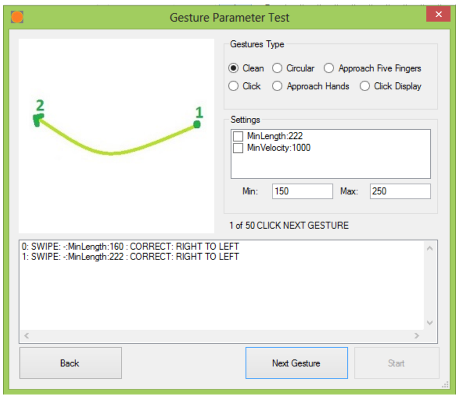


Figure 5 –Interface of evaluation by type.

Then, after finishing all individual sequences of assessment of each gesture and after analyze obtained results. The second group, (Figure 6), is about implementation of retrieved values with better results of previous experience and use these values to initiate a new and last stage of tests to evaluate all gesture types generated randomly.

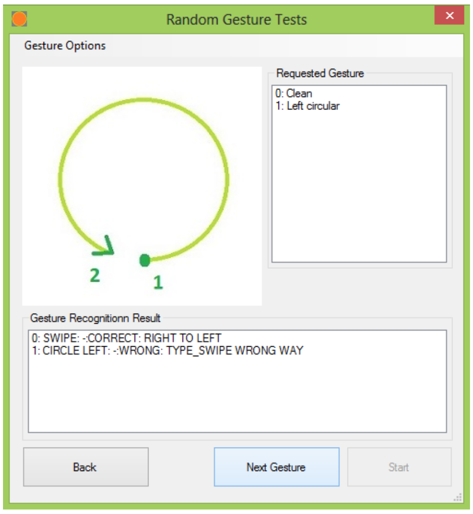


Figure 6 - Interface of general evaluation.

In both experiment interfaces, all acquired results of attempts are saved in text files to be used for further analysis.

4 Validations

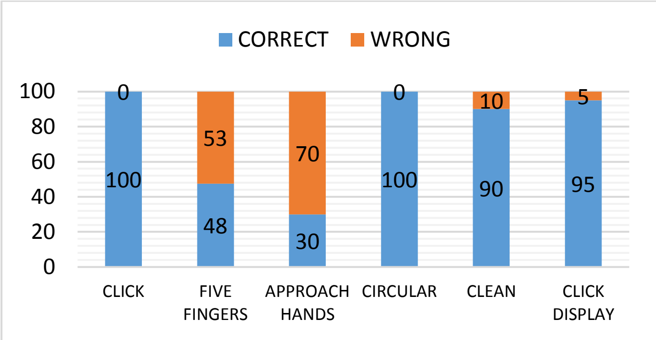
In order to verify the feasibility of the developed application, we proceeded to an informal evaluation with 2 participants with 20 and 27 years old which each of them did 200 attempts, using LeapMotion with USB cable 2.0 at 'Balanced' mode. The reason why we have chosen these settings to perform our experiments over worst conditions in order to optimize the latency problem of our application and under bad conditions try to achieve the best possible results of gestures recognition.

All evaluations, were performed using a machine equipped with a CPU Intel Core i7-4700MQ working at 2.40 Ghz with 4GB RAM DDR3, using USB cable 2.0 and working on operating mode 'Balanced' where it is frame rate was between 110 and 120 fps.

During the implementation of new gestures, and along of experimental stage of gesture FIVE FINGERS, this move was detected and instantly recognizing all fingers when user put his hand over device, entering on interaction box and crossing plans. To improve this situation and add another verification in order to give time to change his command or retreat his action. It was decided to give freedom of movement before recognizing five fingers, and allow the user to move his hand inside the interaction box. Only after a period of time of 5 seconds recognition process of fingers is launched.

In first evaluation stage, was applied general evaluation tester, and generated randomly 600 attempts. All gestures generated were defined to inhibit the remaining

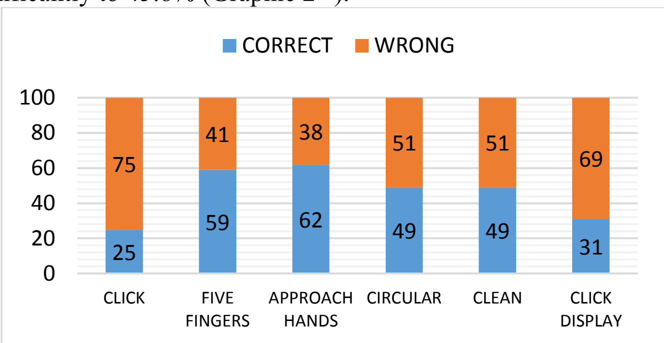
5 types of gestures. Every time, one gesture was asked all others would be disregarded. This test methodology got an average of correct gesture identifications of 77.1% (Graphic 1st).



Graphic 1st – Results of 1º evaluation of all gesture types.

During second evaluation was applied general evaluation tester as on previous stage. However, was removed the restriction about the recognition of one type of gestures in each test and enabled the ability to recognize all gesture types at same time. Now all gestures: CLICK, FIVE FINGERS, APPROACH HANDS, CIRCULAR, CLEAN and CLICK DISPLAY could be recognized on any attempt.

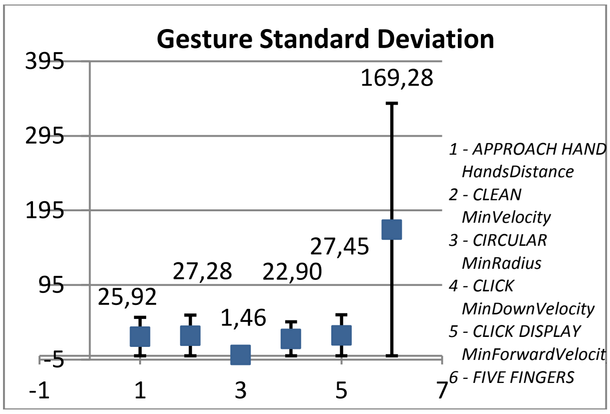
So, each gesture should be much distinct as possible to improve correctly identification. At the moment in which all gestures are accepted, makes CLEAN gesture wrongly recognized, because this move is applied in many others different moves, misunderstandings errors of recognition. Through elaboration of a set of more than 600 attempts was possible to verify the number of correct answers went down significantly to 45.8% (Graphic 2nd).



Graphic 2nd - Results of 2º evaluation of all gesture types.

Among second evaluation and the last, was considered individually each type of gesture changing some parameters of them and testing each one separately. For APPROACH TWO HANDS gesture, was changed distance between hands through parameter *HandsDistance;* generating randomly values between 100 and 200 mm and replacing them in order to find best value which was 142 mm. On CLEAN gesture, was changed swipe minimum velocity parameter *MinVelocity,* which changed between 150 and 250 mm/s, getting 209.2 mm/s as best result. Next gestures were two circular directions of CIRCULE which was randomly, changed the radius *MinRadius* of the circle move using finger replacing by values between 5 mm and 10 mm and getting 6.7 mm as better result. Follow gesture CLICK DISPLAY, was replaced value of minimum velocity of finger getting in on interaction box *MinForwardVelocity*, the speed variation was between 50 and 150 mm/s getting an optimal value of 93 mm/s. Next gesture was the CLICK, and parameter changed on him was click minimum velocity, replacing with values between 50 and 100 mm/s and getting 87.2 mm/s as best result. Finally, on detection of 5 FINGERS was replaced parameter *DetectHandFingers* with values between 1000 and 1500 ms having 1290.8 ms as optimal value.

On next graph (Graphic 2), are presented all types of gestures as well as dispersion of values collected based in the average of each gesture. From all gestures CIRCULAR as the one who had lower variability of values, on other hand FIVE FINGERS was the one who had more different correct values along the recognitions.



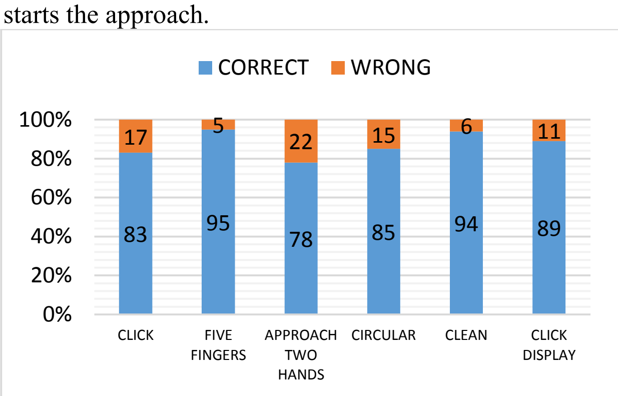
Graphic 2 - Standard deviations of the values of each parameter applied to each gesture.

At a final evaluation stage, was planned a strategy to improve recognition, enabling in each request gesture could be recognize another one. Combination of gestures as shown on table below (Table 1) for each required gesture, application can only identify the requested sign and another one with the exception of FIVE FINGERS and APPROACH TWO HANDS in these last, is possible to recognize all types of gestures. All these combinations are grouped based on gestures differences joining those with more differences between themselves.

Request	Gestures combinations			
CLEAN	CLEAN	CIRCULAR		
CIRCULAR	CIRCULAR	CLICK		
CLICK	CLICK	CLEAN		
CLICK DUSPLAY	CLICK DUSPLAY	CLICK		
FIVE FINGERS	CLEAN	CIRCULAR	CLICK	CLICK DISPLAY
APPROACH TWO HANDS	CLEAN	CIRCULAR	CLICK	CLICK DISPLAY

Table 1 – Combination of allowed gestures on 3º evaluation.

Through the collected values from individual experiments, last evaluating stage were 86.1% better using this method than others prior methods applied on previous evaluations. This method allows to reduce problems about affinities between signs and grouping each of them in order to improve signs recognitions as shown on graphic below (Graphic 3rd). APPROACH TWO HANDS and FIVE FINGERS are gestures which are very heterogeneous, comparing features between them and with all others gestures, so don't was necessary restrict them. Throughout the reviews, a common error was the detection of CLEAN sign who create interferences because the sliding movement some time is associated with a begin of an CLICK gesture when finger starts the approach.



Graphic 3rd - Results of 3º evaluation of all gesture types.

5 Conclusion and Future Work

The study of different configuration features applied to gestures allow us to identify a set of parameters which can perform a low recognition conflict between them and a good results with different features between their recognition working together. At the moment with experiments performed to people with different ages was possible to determine the feasibility of this method applied with adults. Although the results indicate the young public shows more difficult along the familiarization process with plans, comparing with the older participant. However, the younger participant shows to have on other hand greater adaptation capacity, gradually improving his coordination capacities.

Along the evaluations, acquired results allow us to find some of better values to improve recognition process of gestures in order to get better results on sign recognitions.

As perspective of future work, we would like to do more experiments in order to test more deeply our method. Furthermore add new gestures using new techniques and approaches in order to get better efficiency results and bridging ambiguity of signs. Another possibility would be the upgrade to evaluate new version of LeapMotion SDK using a stable release of this version to get the degree of efficiency, flexibility and robustness of this new library.

To conclude, with the improvement achieved on the last experiment, was possible to approach a possible implementation of gestures applied. On 3rd evaluation stage had a very good recognition rate, making these settings a serious candidate to integrate NUI application to draw and recognize characters written in the air [11].

References

1. Vikram S., Li L., Russel S. *Handwriting and Gesturein the Air, Recognition on the Fly*, University of California, Berkeley, 2013.
2. Ohn-Bar E.,Trivedi M. Hand *Gesture Recognition for Automotive Interfaces: A Multimodal Vision-based Approach and Evaluations*. IEEE Transactions on Intelligent Transportation Systems 2014, 2014.
3. LeapMotion. Leap Motion Controller Set To Ship May 13 for Global Pre-Orders, February, 2013.
4. Nguyen T., Schulze J., Kiyokawa K. Sculpting in Real-Time Using a Leap Motion, University of California, San Diego; Osaka University, 2013.
5. Bedikian R., 2014. Understanding Latency: Part 1. Leap Motion Blog. July 2013.
6. Bedikian R., 2014. Understanding Latency: Part 2. Leap Motion Blog. July 2013.
7. Westover B., 2013. Leap Motion Controller review, July 2013.
8. Pinto P., 2013. Leap Motion – Unboxing e Análise, April 2013.
9. Leap Motion Developer Portal, July 2014. <https://developer.leapmotion.com/>
10. mbuckwald, 2014. V2 Tracking Now in Public Developer Beta, May 2014 <https://community.leapmotion.com/t/v2-tracking-now-in-public-developer-beta/1202>
11. Pedro Leitão; João J. Pereira; António Castro (2012), Interface caligráfica de escrita no ar, 20º EPCG: IPVC.