Leap Motion Technology

**Dr. USHA S.M**  **Pawan Pratap Singh Chauhan**  **Mannala Sharath Chandra** *Electronics and communication Electronics and communication Electronics and Communication*

*JSSATEB JSSATEB JSSATEB*

*Bangalore, Karnataka Bangalore, Karnataka Bangalore, Karnataka*

[ushasm@jssateb.ac.in](mailto:ushasm@jssateb.ac.in) [chauhanpawan056@gmail.com](mailto:chauhanpawan056@gmail.com) [sharathchandramannala@gmail.com](mailto:sharathchandramannala@gmail.com)

**Krishnakanth Reddy Mypati Sujith**

*Electronics and communication Electronics and communication*

*JSSATEB JSSATEB*

*Bangalore, Karnataka Bangalore, Karnataka*

[*m.kittamkd@gmail.com*](mailto:m.kittamkd@gmail.com)[*mypatisujith@gmail.com*](mailto:mypatisujith@gmail.com)

***ABSTRACT -*** *The Leap Motion controller is a device that reads the precise position of a person’s hands in the space above it, and relays that data to a program running on the person’s computer. The purpose of this project is to create a prototype of an application that uses the Leap Motion controller to create and run a presentation. The hand and gesture recognition of the Leap Motion device facilitates an intuitive application that enables presenters to interact less with their computer and more with their audience. The application is written in Java 8, and is heavily based on the JavaFX graphics library. While there remain imperfections in the application, the application is a working prototype that demonstrates the strengths of gesture-based presentation software and demonstrates how an existing software task can be enhanced with new technology.*

**1. Introduction**

In this project, we will show you how to design an Arduino based gesture control that can be controlled by hand to a computer. Hence worth recently Gesture controlled Laptops or computers are getting very famous. This technique is called Leap motion this project enables us to control certain functions on our computer/Laptop by simply waving our hand in front of it. It is very cool and fun to do it, but these laptops are really priced very high. So, in this project let us try building our own Gesture control Laptop/Computer by combining the Power of Arduino and Python We will use two Ultrasonic sensors to determine the position of our hand and control a media player (VLC) based on the position.[1]

Gesturing is a natural part of human communication and becomes more and more important in AR/VR interaction. The Leap Motion Controller is a new device developed for gesture interaction by Leap Motion. The device has a small dimension of 0.5x1.2x3 inches. To use the Leap Motion Controller, the user need to connect it to a computer by USB. Then the users put hands on top of the Leap Motion Controller. It gives an example of how to use the Leap Motion Controller. Leap Motion Usage. The small object in the middle is Leap Motion Controller connecting to the Mac on the right. Hand on top of the Leap Motion is tracked and interacted with virtual objects.

The Leap Motion Controller could detect palm and fingers movements on top of it. The tracking data which contains the palm and fingers’ position, direction, velocity could be accessed using its SDK. According to [Weather et al. 2013], the Leap Motion Controller provides a detection accuracy of about 200 µm. The newest version of Leap Motion Controller Orion currently do not provide gesture recognition. We are trying to implement the gesture recognition.[2]

For many hearing impaired individuals, sign language is the primary means of interaction. Sign language considers Different modes of gestures such as lip movements, facial expression, eye brow movements and hand gestures [1].These gestures can be classified as manual and non-manual gestures. Manual gestures are the ones that consider only hand movements. Whereas non-manual are the ones that include other features such as facial expressions and lip Movements. Gestures can also be classified as static and dynamic gestures. Former are the ones that can be Represented in a single image frame. Later are the ones that constitute continuous movement of hands to complete the gesture.

**2.1. Description**

The Leap Motion Controller is a natural user interface for digital devices that allows touch free control, using hands gestures as input. It is a small and simple device dedicated to hands and fingers tracking. It has three infrared LEDs which emit light with wavelength of 850 nanometers (outside the visible spectrum) and two cameras that capture the reflected light in this spectrum.

The device is compact, small (a few centimeters) and light (32 grams), making it easy to be used and transported everywhere. Because of its simplicity, it is affordable, with the price of 56 USD as we write. The distance monitored by the LMC starts from a few centimeters up to almost one meter.

**2.2. Functioning**

The LMC uses a USB 2.0 connection to transfer data to the computer. The device has a sensing space of near a quarter of a cubic meter with the form of an inverted pyramid with the maximum working distance of 80 cm. The data takes the form of a sequence of pairs of grayscale stereo images in the near infrared light spectrum representing mainly the objects illuminated by the LMC IR LEDs, but also ghosts produced by other sources of light in that spectrum. The Leap Motion has two different tracking modes - standard (on table) or as a head-mounted device, useful when it is used together with VR devices like Oculus Rift.[3]

The main difference between the Leap Motion and Kinect, a popular natural interface device produced by Microsoft initially for gaming, is that the Leap is able to track particular small objects (hands and pen like things) in a smaller volume, but with much higher accuracy, making high-precision drawing, manipulating 3D objects and pinch-to-zoom gestures on maps possible. The Kinect is focused on capturing general full objects, making it less sensitive to small movements, like moving the fingers or a pen. Unlike Microsoft Kinect, Leap Motion does not provide access to raw data in the form of a cloud of points. The captured data is processed by proprietary drivers supplied by vendor and accessible through a dedicated API, making it just a HCI interface, not a global 3D scanner. The LMC data is presented as a series of snapshots called frames. One frame consists of objects like Hands, Fingers, Pintables (objects longer and thinner than fingers) and additional information, like frame timestamp, rotation, translation, scaling and other data. The frames frequency can depend on browser (60 frames per second maximum) or on device (where it and can reach 200 fps). Even if the Leap Motion claimed that the device will be more accurate than a mouse, as reliable as a Keyboard and more sensitive than a touch screen (2012), there is still some distance to that performances. The usage of the LMC as a standard input device for a PC in 2D pointing tasks is limited compared to a normal mouse because of the large movement times, the high overall effort ratings and the larger error rate. (Bachmann, Weichert & Rinkenauer, 2015). Thereby, a deviation between a desired 3D position and the average measured positions below 0.2 mm has been obtained for static setups and of 1.2 mm for dynamic setups, minimal in reducing the LMC outcomes since the involuntary hand tremor amplitude varies between 0.4 mm ± 0.2 mm (Weichert, Bachmann, Rudak & Fisseler, 2013).[4]

**2.3. Programming Leap Motion**

One of the big features of the device is the availability of open-source software. The community of developers started from the 12.000 that received test units in October 2012 in the Software Developer Program to over 200.000 people now. The enthusiasts benefit of well documented APIs and libraries, accessible on the dedicated portal at https://developer.leapmotion.com. In February 2016, Leap Motion released Orion, a major update to its core software, designed specifically for hand tracking in Virtual Reality. The Leap Motion SDK contains a library in C++ and one in C that define the API to the LMC tracking data. Wrapper classes for these libraries define language bindings for C# and Objective-C, while bindings for Java and Python use SWIG, an open source tool. Excluding the leap.js client JavaScript library, all the files needed to develop applications (libraries, header files, plugins, etc.) are included in the Leap Motion SDK, downloadable from the portal. The supported languages are JavaScript, C#, C++, Java, Python and Objective-C. There are general classes, like Controller, Frame, Hand, Finger, Pointable, Vector, which are present in all the languages APIs. There is also a pack of assets dedicated to Unity and an independent plugin for Unreal.

The users contribute with libraries for integration with different programming languages, like Processing or with code to improve the performance of the existing features. For example, Nowicki, Pilarczyk, Wasikowski and Zjawin (2014) use Support Vector Machines and Hidden Markov Models to create a gesture recognition library for real-time processing.

**3.1. Applications**

Thereare many applications that are using the LMC for learning purposes and their number is continuously increasing. The students from the Faculty of Engineering in Foreign Languages of the University POLITEHNICA of Bucharest are working with the device in various instances, preparing themselves for the next generation of HCI devices. They are working with it in interdisciplinary project based learning (Pavaloiu, Petrescu & Dragomirescu 2014) or use it for VR based learning applications. In the following we will demonstrate the usefulness of this concept presenting some of their applications.

There is presented a controller for an anthropomorphic robotic hand, used as an example of an interdisciplinary robotics diploma project. The LMC reads the position of the fingers and transfer it to the computer. The corresponding angles are encoded and sent to an Arduino device that controls the positions of the fingers using 5 servos. The code is developed in JavaScript.[5]

The PiRover robot, a rover type machine which is controlled using gestures. The hand movements are read by the device, interpreted and the corresponding commands are sent to the robot by a wired or wireless connection (Bluetooth or Wi-Fi). The command segment of the robot, made by a Raspberry Pi 2 board, receives the commands and controls the rover tracks correspondingly. The position of the palm above the device imposes the direction – forward, backward, left or right or combinations of those. A close fist is the signal for stop. The code is developed in Python.

Some of our efforts are dedicated in create VR applications for e-learning and a favorite development environment is Unity. The Leap Motion Development Portal integrates well the LMC in Unity through a plugin and several assets (2015). Pavaloiu, Sandu, Grigorescu, Ioaniţescu and Dragoi (2016) describes the use of VR for dentistry learning, while Pavaloiu,, Sandu, Grigorescu and Dragoi (2016) show the advantages of inexpensive devices like LMC used for VR training. A major drawback of non-haptic devices is the fact that they are not efficient in learning medical procedures that implies force feedback, like drilling or cutting. There are still several operations that can be practiced and improved in VR, like visualization, choosing the right treatment, making the resins, etc. In Fig. 4 there is shown an application that challenges the student to discover caries using a mouth mirror. The code is developed in Unity.

**3.2. Future of Leap-Motion type devices**

Natural user interfaces will dominate in the next years the HCI. The simplicity and reduced cost of Leap-Motion make it a prominent candidate for the future mobile interfaces. It is relatively simple to integrate the hardware of the LMC into a smartphone and to obtain a companion natural interface to the surrounding digital devices. Because of the large computing power the smartphones are showing, they will perform all the data processing and they will also adapt the recognition to the owner characteristics.

**3.3. Objectives**

* The main objective of this project is to allow for a spatial tracking of a user's hands. It is specifically designed to interact with computers by using natural hand movements and gestures.
* It describes the possibilities of this device and to outline suitable areas of applications, in particular for augmented reality-based applications.

**4. Methodology**

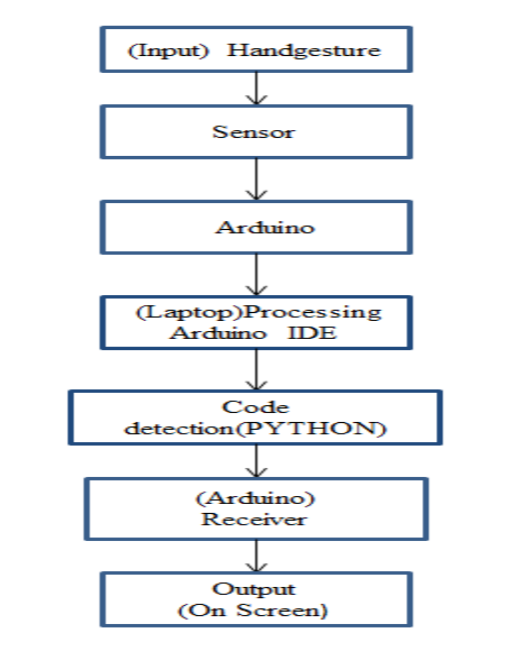


Fig 1: leap motion technology

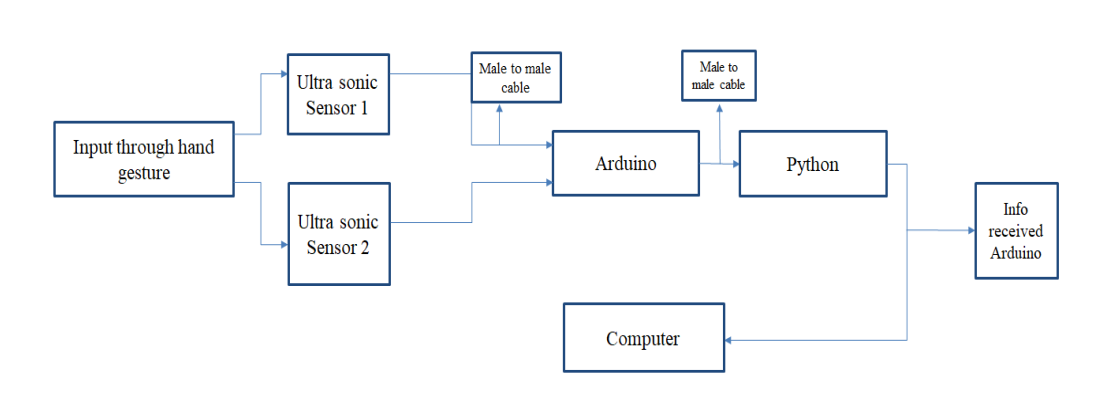


Fig 2 : Functional Representation of leap motion

**5. Hardware Components**

* Arduino
* Two Ultrasonic Sensors (HC-SR04)
* Some wires
* USB Cable (for Arduino)

**Software Tools:**

* Arduino IDE
* Python IDLE
* PySerial library (We will use to communicate with serial ports)
* PyAutogui library (We will use to perform actions)

**6. Results & Analysis**

In designing and implementing the Leap Motion Presenter, we identified several important strategies for designing an application that uses a completely new control scheme. The most crucial area of focus when designing this application was the user experience and interactions between the user and the application. With a new medium of interaction, new user expectations for interaction arise. A good example is using the grab gesture to initiate movement of objects, as the relationship between grabbing an object in the real world is mapped to grabbing an object in the presentation. Most of the other gestures, on the other hand, do not map to an action performed in daily life, but still successfully map to expectations of users through other established applications.

Many desktop applications follow common design elements, such as clickable buttons that appear pressed when clicked to signal a button press to the user or toolbars at the top of the application for the bulk of menu items. These applications follow these design patterns as users have been accustomed to using applications that make use of these interface elements, and the same can be done with Leap gestures. The resize gesture was inspired by mobile applications that support zooming in and out through pinching and squeezing. By slightly modifying it to work with two hands, but still retaining the idea of pinching and squeezing, we created a way of resizing objects that the user is familiar with. Being able to tie in gesture actions with other relatable actions, whether it be mimicking real life actions or existing implementations, translates to actions that users will be able to perform and remember with ideally as little frustration as possible.

Along with gestures, menu design needs to ensure the actions that can be performed on 29.

The device are utilized to successfully navigate through user options, rather than attempting to mimic current desktop menu trends that have been designed with the mouse and keyboard in mind. One of the earliest examples of this are the specialized toolbars, as selections are not made by having the user push downward on the desired menu item with their hand. This type of design attempts to utilize the device’s strengths in hand tracking as well as use a gesture that falls into the expectations of the user to a certain degree.

**7.Discussion**

The paper presents a scheme to recognize static gestures from ISL using a Leap motion controller. Acquisition of input frame through a leap motion device helps to meet the requirements of efficient gesture recognition in a 3-D space. The sensor tackles the major issues in earlier vision based systems such as skin color, lighting variations and hand orientation relative to the device.

In this work, distance features are used for training and classification. The system is tested using four similarity measures namely Euclidean Distance measure, Cosine similarity, Dice similarity and Jaccard similarity. Analysis of these four distance measure shows that cosine distance gives maximum accuracy of 90%.

The accuracy can be increased by using more features such as finger angles and hand orientation with classifiers such as Bayes and HMM for dynamic gestures. The Leap Motion controller can be integrated with kinect camera to achieve recognition of all body gestures for different sign languages.

***REFERENCES***

1. ***Control your Computer with Hand Gestures using Arduino (circuitdigest.com)***
2. ***Arduino based Hand Gesture Control of Your Computer (electronicshub.org)***
3. ***Literature Survey Reference***
4. ***Godwin Ponraj (PDF) Hand Gesture Recognition Control for Computers Using Arduino (researchgate.net)***
5. ***Patsakorn Rittitum (PDF) Hand Gesture Recognition Control for Computers Using Arduino (researchgate.net)***
6. ***Lucy Dodakian Lucy Dodakian's research works | University of California, Irvine, CA (UCI) and other places (researchgate.net)***
7. ***Meenakshi Panwar Meenakshi Panwar's research works | Centre for Development of Advanced Computing, Pune (CDAC) and other places (researchgate.net)***

***CONFERENCE PAPERS***

1. ***“SYSTEMS AND METHODS OF FREE-SPACE GESTURAL INTERACTION”***

***Publication number: 20190258320***

***Type: Application***

***Filed: May 2, 2019***

***Publication date: August 22, 2019***

***Applicant: Leap Motion, Inc.***

***Inventors: Hua Yang, Leonid Kontsevich, James Donald, David S. Holz, Jonathan Marsden, Paul Durdik.***

1. ***“DETERMINING THE ORIENTATION OF OBJECTS IN SPACE”***

***Publication number: 20190257914***

***Type: Application***

***Filed: May 3, 2019***

***Publication date: August 22, 2019***

***Applicant: Leap Motion, Inc.***

***Inventor: David HOLZ***

1. ***“USER-DEFINED VIRTUAL INTERACTION SPACE AND MANIPULATION OF VIRTUAL CAMERAS IN THE INTERACTION SPACE”***

***Publication number: 20190250719***

***Type: Application***

***Filed: April 29, 2019***

***Publication date: August 15, 2019***

***Applicant: Leap Motion, Inc.***

***Inventors: Isaac COHEN, Maxwell SILLS***

1. ***“SYSTEMS AND METHODS OF FREE-SPACE GESTURAL INTERACTION”***

***Patent number: 10281987***

***Type: Grant***

***Filed: September 3, 2014***

***Date of Patent: May 7, 2019***

***Assignee: Leap Motion, Inc.***

***Inventors: Hua Yang, Leonid Kontsevich, James Donald, David S Holz, Jonathan Marsden, Paul Durdik***

1. ***“VELOCITY FIELD INTERACTION FOR FREE SPACE GESTURE INTERFACE AND CONTROL”***

***Publication number: 20190113980***

***Type: Application***

***Filed: December 7, 2018***

***Publication date: April 18, 2019***

***Applicant: Leap Motion, Inc.***

***Inventors: Isaac COHEN, David S. HOLZ, Maxwell SILLS***