A Project Report

On

Contactless Gesture Recognition System using Infrared Proximity Sensors

BY

Anjali Goyal 2017A8PS0818H

Archana S. 2016B4A80535H

Harsha 2017A8PS1553H

Shailja Preum 2017A8PS0473H

Under the supervision of

Dr. Karumbaiah Chappanda and Dr. Ponnalagu R N

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

INSTR F311 and INSTR F312

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

HYDERABAD CAMPUS

(NOV 2019)

ACKNOWLEDGMENTS

We would like to thank Professor **Karumbaiah chappanda Sir and Ponnalagu R N** Ma'am for presenting us with this opportunity to be a part of a Project under them. It is a radical opportunity with immense scope to expand our theoretical and practical knowledge in our area of interest. We would like to take this chance to thank the HOD of our Phoenix department. Lastly, we would also like to thank the lab assistants and the TA assigned to us for their constant and helpful support.

Birla Institute of Technology and Science-Pilani,

Hyderabad Campus

Certificate

This is to certify that the project report entitled "Contactless Gesture Recognition System using Infrared Proximity Sensors" submitted by Ms. Anjali Goyal, Ms. Harsha, Ms. Shailja Preum, Ms. Archana S in partial fulfillment of the requirements of the course INSTR F311 Electronic Instruments and Instrument Technology and INSTR F312 Transducers and Measurement system, embodies the work done by them under my supervision and guidance.

Date: 21/11/19 **(SIGNATURE)**

BITS- Pilani, Hyderabad Campus

CONTENTS

ABSTRACT

We propose a contactless, vision-based gesture recognition system that uses **infrared proximity sensors** to detect gestures made. Whenever the hand of the user is near the system, the reflection of IR light is detected by the infrared sensor, and the intensity of the light increases when the distance of the object decreases. The infrared signals picked up by the sensor are then processed and classified using a **custom classifier** which will be tested on a custom dataset of different gestures. The setup configuration includes a proximity sensor that is constructed using **two IR LEDs and an IR receiver**. The **gesture classification algorithm** has been written using the **Arduino IDE**.

The gesture recognition system's power will mostly be from the power that is consumed by the control chip and the IR LED, and we estimate that the **power consumption** will be restricted to **less than 25 mW**, making this an **efficient, low power consuming**, real time 3-D gesture recognition system that can be easily deployed onto less complex devices.

INTRODUCTION

Gesture Recognition Systems are utilized as an interface between PCs and humans, along with being employed to control and interact with a wide variety of electronic instruments. These Gesture Recognition systems can fundamentally be classified into three classes as follows:

1. Motion-based: When the user holds a device or a controller that detects the gesture made.

2. Touch-based: When the system includes a touch-screen and the positions and directions of the finger or equivalent tool of the user are mapped, thus recognizing the gesture.

3. Vision-based: When the system makes use of image and signal processing to detect gestures made without touching any device.

The first two types of systems need the users to hold and contact certain devices, for the gesture recognition, and vision based systems use camera setups, image processing and techniques that involve computer vision. These systems are difficult to set up for small scale use and are also expensive and extremely power hungry. For building a system that needs to function when there are limited resources available, it is important that the setup cost, power consumption and ease and size of setup is taken into consideration. Keeping this in mind, we have built a contactless gesture recognition system that consists of a couple of digital infrared sensors, that have been programmed to do the gesture recognition using a custom algorithm, with an Arduino Uno Microcontroller.

Problem Solving Methodology

Components Used and Setup:

The components we used are:

1. Breadboard
2. Arduino Uno Microcontroller
3. 2 digital IR Sensors
4. Jumper wires
5. Laptop for interfacing

Languages used:

1. Arduino IDE (Based on C++)
2. Python (for interfacing sensor output with VLC)

We connected two IR sensors to the Breadboard, placed at a distance of approximately 3 cm from each other. These were then interfaced with the Arduino Uno, which was connected to the laptop.

Voltage is applied to the pair of IR LEDs, which in succession emit Infrared light. This light propagates through the air and once it hits the hand (or object), which acts as a hurdle, it is reflected back to the receiver. The LED on the diode glows, thus indicating that an object has been detected.

A digital sensor system consists of the sensor itself, a cable, and a transmitter. The sensor consists of electronic chip. The signal measured is directly converted into a digital signal. The data transmission through the cable is also digital.

Using the concept of states and delay as in Digital Design, we have created two states of the two sensors each placed on the left and right. These two states are defined and calibrated using a time delay of a few hundred microseconds in the gesture classification algorithm written using the Arduino IDE.

We have taken two states in the algorithm into consideration namely Q(t) and Q(t+d) where d is the delay defined. The algorithm is defined such that left sensor and right sensor digital values are checked first and then after the defined delay, both sensors are checked for their boolean values again and therefore the gesture is recognized and printed on the screen after running the code in the Arduino software. The chip on the Arduino Uno board plugs into the USB port of the computer and supports the laptop as a virtual serial port.

To make the gesture recognition feature interactive, we have interfaced the output with VLC Media player, so that we can pause, play and rewind/go forward with the playback. To do this interfacing, we have written a Python script, importing the Python library pyautogui, that provides functionality of control of the computer's keyboard.

State Table -

	Left		**Right**	
Gesture	Q(t)	Q(t+d)	Q(t)	Q(t+d)
Left	**0**	**1**	**1**	**0**
Right	**1**	**0**	**0**	**1**
Push	**0**	**1**	**0**	**1**
Pull	**1**	**0**	**1**	**0**

Results and Conclusions

We tested the gesture recognition system for accuracy by using the precision-recall matric. We took in 30 different samples for input.

The precision is calculated as TP/(TP+FP) , where TP denotes the number of true positives and FP denotes the number of false positives.

The precision of our system is =

The recall is calculated as TP/(TP+FN), where TP denotes the number of true positives, and FN denotes the number of false negatives.

The recall of our system is =

High precision implies that there is less chances of getting false alarms, and recall expresses the ability to find all relevant information from the dataset.

Practical Application:

The system is low-power, and able to recognize gestures with over 98% precision in real time. As mobile phones and tablets are becoming more and more important in everybody's lives, there is a growing need of an intuitive user interfaces for small-sized, resource-limited mobile devices. While the first two types of the gesture recognition system requires users to make contact with devices, a contactless system becomes more and more significant. Where vision-based systems using camera and computer vision techniques allow users to make intuitive gestures without touching the device. However, most vision-based systems are computationally expensive and power-consuming, which is undesirable for resource-limited mobile devices like tablets or mobile phones.

Further scope

1. Friendly user interface that can be easily understood by any user and eventually its application can be extended to more applications like PDF reader, video games etc.

2. Computationally inexpensive and low power consuming hardware and software setup, that makes it ideal for integrating with any device, both simple and complex.

Limitations

1. Ambient light obstructs the functioning as is the case with infrared sensors, as they are extremely sensitive. A proper optical barrier must be used to prevent this.
2. We have assumed values of time delays between gestures according to what worked well for our test dataset. This leads to the system being slightly inflexible with different speeds of gestures.

Arduino Uno Code snapshots -

```
int sensorL = 3;
int sensorR = 6;
bool r=false,l=false,m=false;
int iL,iR,n=0;

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600) ;

  pinMode(sensorL, INPUT);
  pinMode(sensorR, INPUT);

}

void loop() {
  Serial.begin(9600) ;

iL=digitalRead(sensorL);
iR=digitalRead(sensorR);
```

```
if(iL==0 && iR==1)
  {
    l=true;
    m=true;
  }

 if(l==true)
   {
    iR=digitalRead(sensorR);
    if(iR==0)r=true;

    }

   if(m==false && r==true && l==true ){
    Serial.println("Right swipe!");
    r=false;
    l=false;
    m=false;
    delay(100);
  }
```

```
   if(iR==0 && iL==1)
   {
     r=true;
     m=false;
   }
   if(r==true)
   {
     iL=digitalRead(sensorL);
     if(iL==0)
       {
         l=false;
       }
   }

   if(m==true && r==true && l==false ) {
     Serial.println("Left swipe!");
     r=false;
     l=false;
     m=false;
     delay(100);
   }
```

```
  if(iR==0 && iL==0){
    m=true;
    r=true;
    l=true;
  }
  if(m==true && r==true && l==true){
    Serial.println("Push");
    r=false;
    l=false;
    m=false;
    n=1;
   delay(800);
  }
 if(n==1)
  {
   iL=digitalRead(sensorL);
   iR=digitalRead(sensorR);
    if(iL==1 && iR==1){
      Serial.println("Pull");
      n=0;
         }
    }
    delay(200);
```

Python code

```
import serial #Serial imported for Serial communication
import time #Required to use delay functions
import pyautogui

ArduinoSerial = serial.Serial('com5',9600) #Create Serial port object called arduinoSerialData
time.sleep(2) #wait for 2 seconds for the communication to get established

while 1:
    incoming = str (ArduinoSerial.readline()) #read the serial data and print it as line
    print (incoming)

    if 'Push' in incoming:
        pyautogui.typewrite(['space'], 0.2)

    if 'Left swipe!' in incoming:
        pyautogui.hotkey('ctrl', 'left')

    if 'Left swipe!' in incoming:
        pyautogui.hotkey('ctrl', 'right')

    incoming = "";
```

https://forum.arduino.cc/index.php?topic=66868.0

https://circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing