A Project Report

on

HAND GESTURE RECOGNITION

Technical Skilling-2

By

Navadeep Reddy	2010030313
Siddharth	2010030475
Vipul Reddy	2010030502
Manoj Peravali	2010030503

under the supervision of

Ms. P.SREE LAKSHMI

Assistant Professor

Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500075, Telangana, India.

April, 2022

DECLARATION

The Project Report entitled "**HAND GESTURE RECOGNITION**" is a record of bonafide work of **NAVADEEP REDDY(2010030313), SIDDHARTH (2010030475), VIPUL REDDY(2010030502), MANOJ PERAVALI(2010030503)** submitted in partial fulfillment for the award of B.Tech in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/University/Institute.

NAVADEEP REDDY

VIPUL REDDY

MANOJ PERAVALI

SIDDARTHA SRI SAI

CERTIFICATE

This is to certify that the Project Report entitled "**HAND GESTURE RECOGNITION"** is being submitted by **Navadeep Reddy (2010030313), Siddharth (2010030475), Vipul Reddy (2010030502), Manoj Peravali (2010030503)** submitted in partial fulfillment for the award of B.Tech in CSE to the K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/institutes.

Signature of the Supervisor

Ms. P. SREE LAKSHMI

Assistant Professor

Signature of the HOD

Signature of the External Examiner

ACKNOWLEDGEMENT

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for completing this project successfully.

We take a grateful opportunity to thank our beloved Founder and Chairman who has given constant encouragement during our course and motivated us to do this project. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgment & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike** for her exemplary guidance, monitoring, and constant encouragement throughout the course of the project. We thank **Ms.P. Sree Lakshmi** of our department who has supported us throughout this project holding the position of supervisor.

We wholeheartedly thank all the teaching and non-teaching staff of our department without whom we won’t have made this project a reality. We would like to extend our sincere thanks, especially to our parents, our family members, and our friends who have supported us to make this project a grand success.

ABSTRACT

Hand gesture recognition system has developed excessively, reason being its ability to cooperate with machine successfully. Gestures are considered as the most natural way for communication among humans.

PCs in virtual framework. We often use hand gestures to convey something asit is non-verbal communication which is free of expression.

In this application, ourPC's camera records a live video, from which a preview is taken with the assistance of its functionalities or activities.

TABLE OF CONTENTS

I. List Of Chapters

1. INTRODUCTION

Gesture recognition is an area of active current research in computer vision. It brings visions of more accessible computer system. In this paper we focus on the problem of hand gesture recognition using a real time tracking method with Pseudo two-dimensional hidden Markov models (P2-DHMMs). We have considered singlehanded gestures, which are sequences of distinct hand shapes and hand region. Gesture is defined as a motion of the hand to communicate with a computer. They are capable to handle 3D worlds, making it natural and realistic, and provide in some implementation's tactile sensations. Regrettably, their cost is usually very high, and thus the user acceptance confined, hence making them more desirable for professional applications such as a flight simulator or remote surgery equipment. However, these interactions are often limited and non-rational, while the devices are awkward, unmanageable, and prone to distortion from the physical environment. We are interested in formulating an alternative, natural interface that more intimately models the way we interact with the real world. The user should be able to reach out, grab, point and move 3D objects just as we do with real objects.

These challenges open a new direction for human-computer interaction which combined with computer vision techniques, and it is possible to build an advanced input device. The computer vision devices can be implemented and upgrade to the new input devices in the future. It gives the input command to the computer rather than just a function of taking photo or record video. We can do more implementation to transform the computer vision devices to become an input command device to reach the function as keyboard or mouse. One of the ways to give signal to computer vision devices is by using hand gesture. More specifically hand gesture is used as the signal or input modality to the computer. Certain signal can be recognized by computer as an input of what computer should do. These will benefit the entire user without using a direct device and can do what they want as long as the computer vision device can sense it. These make computer user easier than using the keyboard or mouse. The future computer or laptop may eliminate the use of keyboard and mouse by substituting with a vision-based interpretation device.

2. LITERATURE SURVEY

ARTICLE NAME-Hand Gesture Recognition with Depth Images: A Review

AUTHOR-Jesus Suarez* and Robin R. Murphy, Member, IEEE

DATE-September 9-13, 2018

This paper presents a literature review on the use of depth for hand tracking and gesture recognition. A total of 10 methods are commonly used for hand tracking and gesture recognition in the papers reviewed (2 for segmentation, 3 for tracking, and 5 for classification). Hand segmentation is most commonly accomplished using depth thresholding or region growing techniques.

ARTICLE NAME- A REAL-TIME HAND GESTURE RECOGNITION METHOD

AUTHOR- Yikai Fang1, Kongqiao Wang2, Jian Cheng1 and Hanqing Lu1

Date- May 2018

vision based hand interaction is more natural and efficient. In this paper, we proposed a robust real-time hand gesture recognition method. ether our method combines fast hand tracking, hand segmentation and multi-scale feature extraction to develop an accurate and robust hand gesture recognition method. It takes advantage of color and motion cues acquired during tracking to implement adaptive hand segmentation

ARTICLE NAME- Vision Based Hand Gesture Recognition

AUTHOR- Pragati Garg, Naveen Aggarwal and Sanjeev Sofat

Date- June 2018

Vision based Hand Gesture Recognition techniques for human computer interaction, consolidatingthe various available approaches, pointing out their general advantages and disadvantages.. Visionbased Hand Gesture Recognition techniques for human computer interaction, consolidating the various available approaches, pointing out their general advantages and disadvantages

ARTICLE NAME- Hand Gesture Recognition Systems:
AUTHOR- Arpita Ray Sarkar, G. Sanyal, S. Majumder,
Date- 15, May 2013
. It is the best method to interact with the computer without using other peripheral devices, such as keyboard, mouse. Researchers around the world are actively engaged in development of robust and efficient gesture recognition system, more specially, hand gesture recognition system for various applications. Hand gesture recognition is finding its application for nonverbal communication between human and computer, general fit person and physically challenged people, 3D gaming, virtual reality etc. With the increase in applications, the gesture recognition system demands lots of research in different directions

3. HARDWARE AND SOFTWARE REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

Device name : LAPTOP-V48CJAV1

Processor : Intel(R) Core(TM) i7-9750HF CPU @ 2.60GHz 2.59 GHz

Installed RAM : 8.0 GB

Device ID : 7FA81F64-4E35-4D56-A8CC-28D79875044B

Product ID : 00327-35902-95915-AAOEM

System type **:** 64-bit operating system, x64-based processor

3.2 SOFTWARE REQUIREMENTS

Software : PyCharm Community Edition

Programming Language: Python

4. FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

Web-Browser

Math

Numpy

Cv2

4.2 NON-FUNCTIONAL REQUIREMENTS

Threshold must be set properly

Camera must focus properly

Hand must be recognized correctly

Gaussian blur must be applied properly

5. PROPOSED SYSTEM

5.1 PROPOSED ALGORITHM DESIGN TECHNIQUE

5.1.1 HAND ADAPTIVE ALGORITHM:

Most existing gesture recognition algorithms have low recognition rates under rotation, translation, and scaling of hand images as well as different hand types

Formulae to calculate Hand AdaptiveAlgorithm :L1=A1-A3 L2=A1=A2 L3=2*(A0-A3)

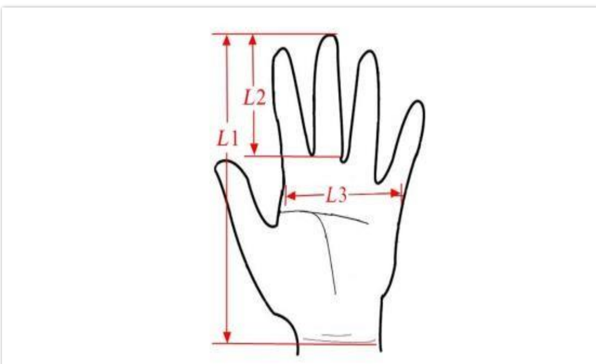


Fig 5.1.1.1

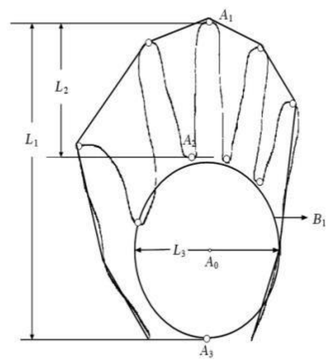


Fig 5.1.1.2

5.1.2 Otsu’s method for image thresholding:

The algorithm iteratively searches for the threshold that minimizes the within-class variance, defined as a weighted sum of variances of the two classes (background and foreground). The colors in grayscale are usually between 0-255 (0-1 in case of float). So, If we choose a threshold of 100, then all the pixels with values less than 100 becomes the background and all pixels with values greater than or equal to 100 becomes the foreground of the image.

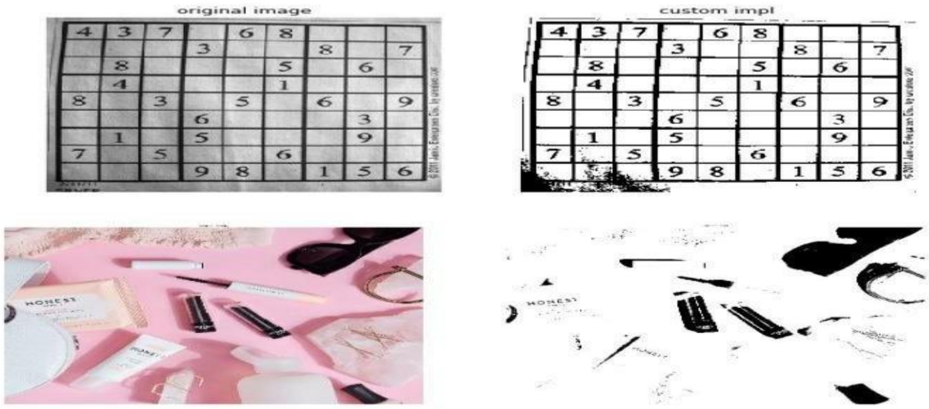


Fig:5.1.2.1

5.1.3 Gaussian blur:

In Gaussian Blur operation, the image is convolved with a Gaussian filter. The Gaussian filter removes the high-frequency components are reduced.

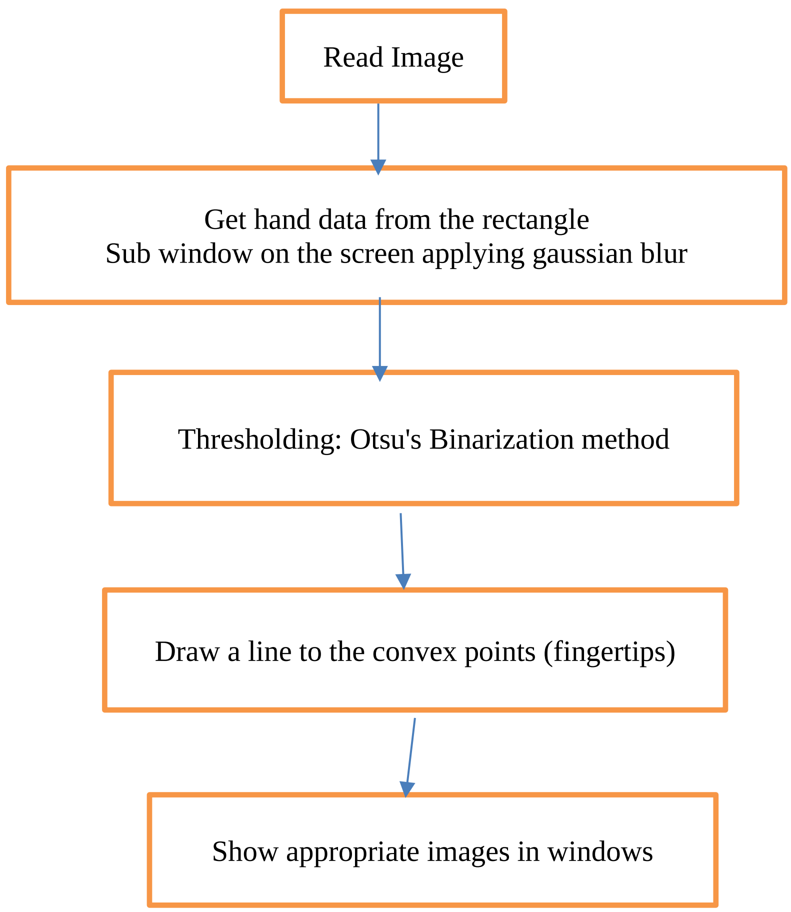
We can perform this operation on an image using the Gaussianblur()

Fig:5.1.3.1 Fig:5.1.3.2

6. IMPLEMENTATION

6.1 WORKFLOW

Steps Chart



6.2 CODE

```
import cv2
import numpy as np
import math
import webbrowser as wb
import os

print("Enter full website for")
print("\n2 fingers")
fingers2 = input()

print("\n3 fingers")
fingers3 = input()

print("\n4 fingers")
fingers4 = input()

tabs = 0
count = 0
cap = cv2.VideoCapture(0)

while (cap.isOpened()):
    ret, img = cap.read()
    cv2.rectangle(img, (1000, 1000), (100, 100), (0, 255, 0), 0)
    crop_img = img[100:400, 100:400]
    grey = cv2.cvtColor(crop_img, cv2.COLOR_BGR2GRAY)
    value = (35, 35)
    blurred = cv2.GaussianBlur(grey, value, 0)
```

```
    _, thresh1 = cv2.threshold(blurred, 127, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
    cv2.imshow('Thresholded', thresh1)
    (version, _, _) = cv2.__version__.split('.')
    if version == '4':
        contours, hierarchy = cv2.findContours(thresh1.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    elif version == '2':
        contours, hierarchy = cv2.findContours(thresh1.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    cnt = max(contours, key=lambda x: cv2.contourArea(x))
    x, y, w, h = cv2.boundingRect(cnt)
    cv2.rectangle(crop_img, (x, y), (x + w, y + h), (0, 0, 255), 0)
    hull = cv2.convexHull(cnt)
    drawing = np.zeros(crop_img.shape, np.uint8)
    cv2.drawContours(drawing, [cnt], 0, (0, 255, 0), 0)
    cv2.drawContours(drawing, [hull], 0, (0, 0, 255), 0)
    hull = cv2.convexHull(cnt, returnPoints=False)  # return point false to find convexity defects
    defects = cv2.convexityDefects(cnt, hull)
    count_defects = 0
    cv2.drawContours(thresh1, contours, -1, (0, 255, 0), 3) # to draw all contours pass -1
    for i in range(defects.shape[0]):
        s, e, f, d = defects[i, 0]
        start = tuple(cnt[s][0])
        end = tuple(cnt[e][0])
        far = tuple(cnt[f][0])
        a = math.sqrt((end[0] - start[0]) ** 2 + (end[1] - start[1]) ** 2)
        b = math.sqrt((far[0] - start[0]) ** 2 + (far[1] - start[1]) ** 2)
        c = math.sqrt((end[0] - far[0]) ** 2 + (end[1] - far[1]) ** 2)
        angle = math.acos((b ** 2 + c ** 2 - a ** 2) / (2 * b * c)) * 57
        if angle <= 90:
            count_defects += 1
            cv2.circle(crop_img, far, 1, [0, 0, 255], -1)
        cv2.line(crop_img, start, end, [0, 255, 0], 2)
    if count == 0:
        cv2.putText(img, "Wait for it :p", (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, 3)
    if count_defects == 1 and count != 2 and tabs <= 8:
        wb.open_new_tab('http://www.' + fingers2 + '.com')
        tabs = tabs + 1
        cv2.putText(img, "2." + fingers2, (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, (255, 0, 0), 3)
        count = 2
    elif count_defects == 2 and count != 3 and tabs <= 8:
        wb.open_new_tab('http://www.' + fingers3 + '.com')
        tabs = tabs + 1
        cv2.putText(img, "3." + fingers3, (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, (0, 0, 255), 3)
        count = 3
    elif count_defects == 3 and count != 4 and tabs <= 8:
```

```
        wb.open_new_tab('http://www.' + fingers4 + '.com')
        cv2.putText(img, "4." + fingers4, (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, (255, 165, 0),
3)
        tabs = tabs + 1
        count = 4
    elif count_defects == 4 and count != 5:
        cv2.putText(img, "5.Close Web browser", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 3, 3)
        os.system("taskkill /im chrome.exe /f")
        tabs = 0
        count = 5
    else:
        cv2.putText(img, "", (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, 3)

    if count == 2:
        cv2.putText(img, "2." + fingers2, (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, (255, 0, 0)3)
    elif count == 3:
        cv2.putText(img, "3." + fingers3, (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, (0, 0, 255), 3)
    elif count == 4:
        cv2.putText(img, "4." + fingers4, (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, (255, 165, 0),
3)
    elif count == 5:
        cv2.putText(img, "5.WebBrowser close", (50, 100), cv2.FONT_HERSHEY_SIMPLEX, 3, 3)
    cv2.imshow('Gesture', img)
    all_img = np.hstack((drawing, crop_img))
    cv2.imshow('Contours', all_img)
    k = cv2.waitKey(10)
    if k == 27:
        break
```

6.4 CODE EXPLANATION

Get hand data from the rectangle sub window on the screen

Applying gaussian blur

Thresholding: Otsu's Binarization method

Draw a line to the convex points (finger tips)

Show appropriate images in windows

We have to show fingers based on our app need to be opened

6. RESULT DISCUSSION

7.1 CONSOLE OUTPUT

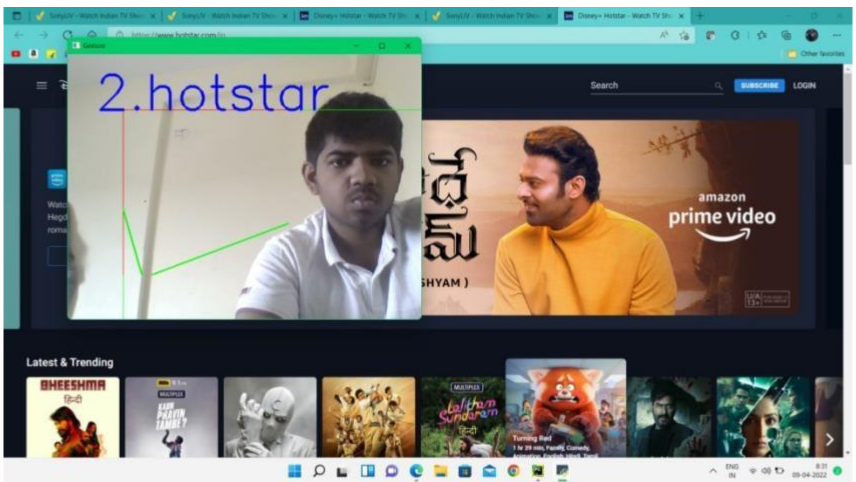


Fig 7.1.1: IMAGE ON HOTSTAR

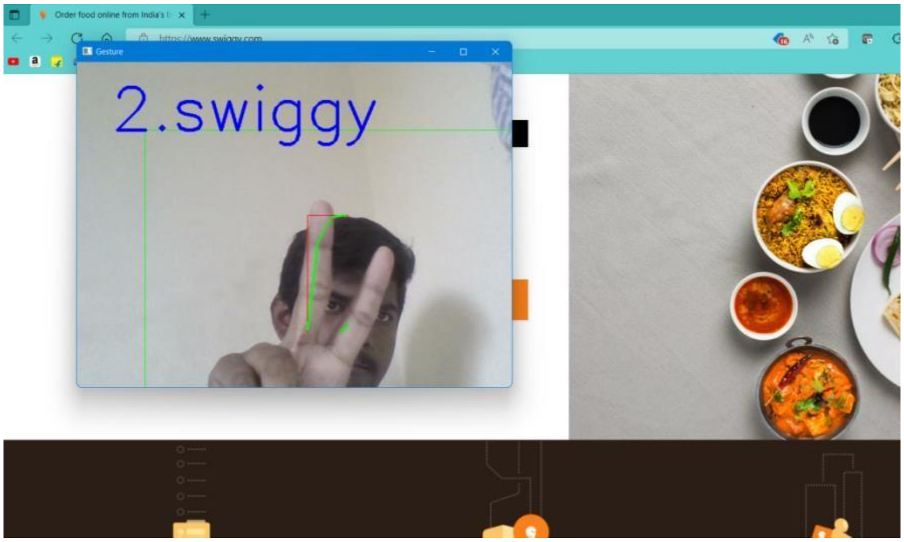


FIG 7.2.2: IMAGE ON SWIGGY

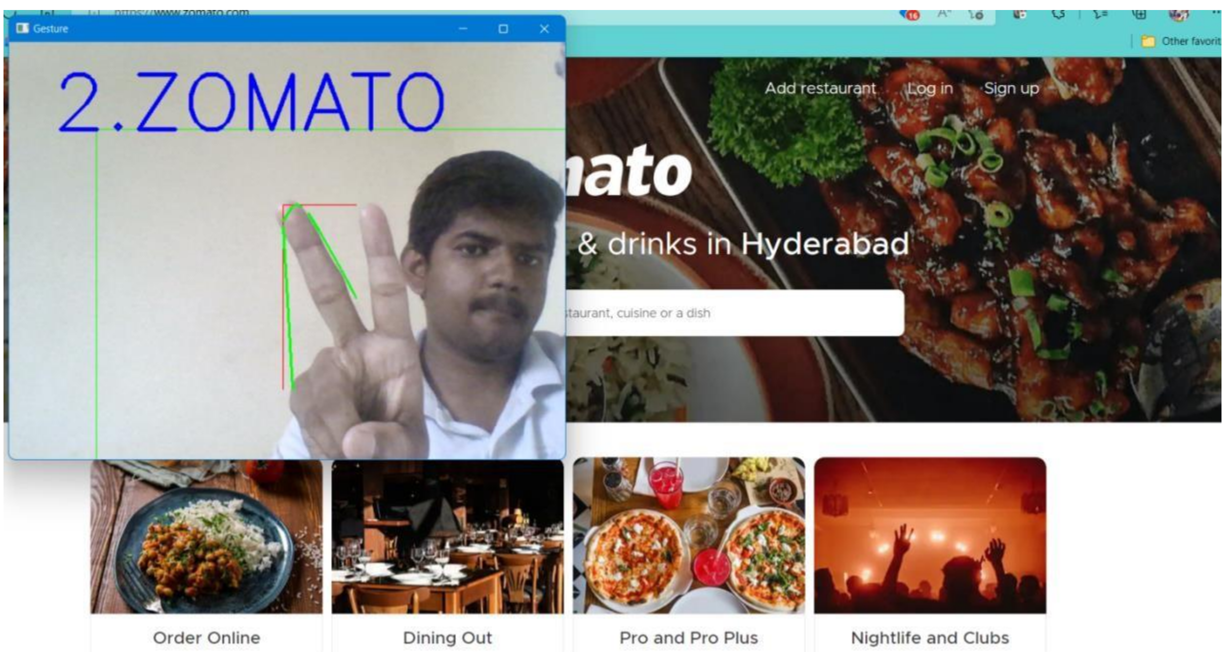


FIG 7.2.3: IMAGE ON ZOMATO

8. CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

We have developed a gesture recognition system that is shown to be robust for ASL gestures. The system is fully automatic and it works in real-time. It is fairly robust to background cluster. The advantage of the system lies in the ease of its use. The users do not need to wear a glove, neither is there need for a uniform background. Experiments on a single hand database have been carried out and recognition accuracy of up to 98% has been achieved. We plan to extend our system into 3D tracking. Currently, our tracking method is limited to 2D. We will therefore investigate a practice 3D hand tracking technique using multiple cameras. Focus will also be given to further improve our system and the use of a lager hand database to test system and recognition.

8.2 FUTURE SCOPE

The present application though seems to be feasible and more user friendly in comparison to the traditional input modes but is somewhat less robust in recognition phase. An attempt to make the input modes less constraints dependent for the user’s hand gestures has been preferred. But robustness of the application maybe increased by applying some more robust algorithms that may help to reduce noise and blur motion to have more accurate translation of gestures into commands.

9. REFERENCES

[1] Conic, N., Cerseato, P., De & Natale, F. G. B., (2007), “Natural Human- Machine Interface using an Interactive Virtual Blackboard”, In Proceeding of ICIP 2007, pp.181-184.

[2] Ismail, N. A., & O’Brien, A., (2008), “ Enabling Multimodal Interaction in Web-Based Personal Digital Photo Browsing”, Proceedings of the International Conference on Computer and Communication Engineering , Kuala Lumpur, Malaysia, May 13-15, pp. 907-910.

[3] Pang, Y. Y., Ismail, N. A., & Gilbert, P. L. S., (2010), “ A Real Time Vision-Based Hand Gesture Interaction”, Fourth Asia International Conference on Mathematical Analytical Modelling and Computer Simulation, pp. 237-242.

[4] Kortum, P., (2008) “HCI Beyond the GUI: Design for Haptic, Speech, Olfactory, and Other Nontraditional Interfaces” Morgan Kaufmann Publishers, pp. 75-106.

[5] Viola & Jones, (2001), “Rapid object detection using boosted cascade of simple features", In Proceedings of Computer Vision and Pattern Recognition, pp. I-511 - I-518. [6] Chen, Q., Coredea, M. D., Petriu, E. M., Varkony, A. R., Koczy, I., & Whalen, T.E., (2009), “ Human Computer Interaction for Smart Applications Using Hand Gesture and Facial Expressions,” International Journal of Advanced Media and Communication, vol. 3c.1/2, pp. 95-109.