RESEARCH PAPER | Engineering | Volume : 5 | Issue : 1 | Jan 2015 | ISSN - 2249-555X

Hand Gestures Recognition For Computing Utilities

KEYWORDS

Prof. PARIMALA. R
Assistant Professor, Department of Information Science & Engineering, P E S Institute of Technolog, 100 ft Ring Road, BSK 3rd stage, Hosakerehalli, Bangalore – 560085, Karnataka, India.

Prof. Manjunath A E
Assistant Professor, Department of Computer Science & Engineering, R V College of Engineering, Vidyanikethan Post, Mysore Road, Bangalore-560067, Karnataka, India.

ABSTRACT *In this paper we propose a hand gesture recognition method to control some of the application utilities in Ubuntu platform. A new approach is used for predicting the real-time gestures for HCI. Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. We have used the OpenCV library for handling and manipulating input from the webcam. Motion Detection is done by finding the difference between subsequent frames. Noise elimination is done so that only the moving object (hand or finger) will be interpreted. The interpreted gestures are then scanned against a set of known gesture to find which gesture matches the best. The action that can be either a system command or a keystroke, is then performed accordingly. Application utilities that can be controlled using hand gestures are - VLC Media Player, Libre Office Presentation, File Conversion and Opening – Processing – Closing few applications.*

I. INTRODUCTION

Gesture recognition enables humans to communicate with the machine and interact naturally without using any mechanical input devices. The concept of gesture recognition is used, to point a hand/finger at the computer screen so that the cursor will move accordingly. This will potentially make the conventional input devices such as mouse, keyboards and touch screens redundant. However controlling media player using hand gestures have always been a challenge. Media player is usually controlled using keyboard or mouse.

But controlling it without keyboard and mouse is difficult task. Many hand gesture recognition methods have been used to capture the gestures and control the media player. There are, however, fundamental limitations to most current systems for gesture detection based off training on a set of pre-defined gestures. Non-uniform lighting conditions, less-than-ideal camera resolution and accuracy of possible gesture classifications are some of the limitations.

This is a Linux-based application for live gesture recognition using webcam input. The C++ language is used in the Open Computer Vision Library. This project is a combination of live motion detection and gesture identification/ recognition. This application uses the webcam to read and detect the gesture made by the user and perform specific operations. The user will perform a particular gesture and using webcam this gesture will be captured. Then identification and recognition (against a set of known gestures) of the same is done. Finally the action corresponding to that gesture will be performed. The application can be running in the background while the user runs other programs and applications. This concept is useful for a hands-free approach. It is useful in media player, Libre office Presentation control, file conversion, opening, closing few applications and navigation of documents or files. A simple gesture can pause or play the movie/song or increase the volume even while sitting afar from the computer screen.

II. DESIGN

This work is divided into four parts:

1. Reading the input gesture and converting it into Greyscale gesture image:

The first stage involves in reading real time hand gesture as an input from the webcam and converting it into a form that can be processed easily. The code continuously streams video from the webcam and processes the frames of the video for recognition of the gesture. This can be done by the subtraction of RGB value of the pixels of the previous frame from the RGB values of the pixels of the current frame. Then this image is converted into another colour space i.e., octachrome (8 colours only - red, blue, green, magenta, cyan, yellow, black, white). This makes most of the pixels neutral or grey. After this greying of those pixels not surrounded by 20 non-grey pixels will be done. The non-grey pixels that remain represent proper motion and noise is eliminated and processed further.

2. Intercepting the gesture from the input of the webcam:

Difference between subsequent frames in the video stream are calculated to detect motion. Then the code further eliminates noise so that only the moving object (hand or finger) in the frames are interpreted.

3. Recognising the gesture from a database of gestures:

A database is used for the recognition purpose, which contains a set of points for each gesture and code for the same. When an user performs a particular gesture, a set of points are generated (using the average of x & y coordinates of non-grey pixels in each frame) which are matched with the gestures in the databases to find the gesture the best matches the input query gesture. To match the gestures the points are appropriately scaled as per their standard deviation and then corresponding points of the user's gesture and that from the database are compared for matching purpose. The gesture in the database which has the least sum of squares of the differences between the corresponding points is returned as the match for the gesture.

4. According to the intercepted gesture, perform specific operation:

The interpreted gesture is scanned against a set of known gesture in the database to find which gesture matches the best. Then an action that is a system command or a keystroke, associated with the keystroke is then performed accordingly.

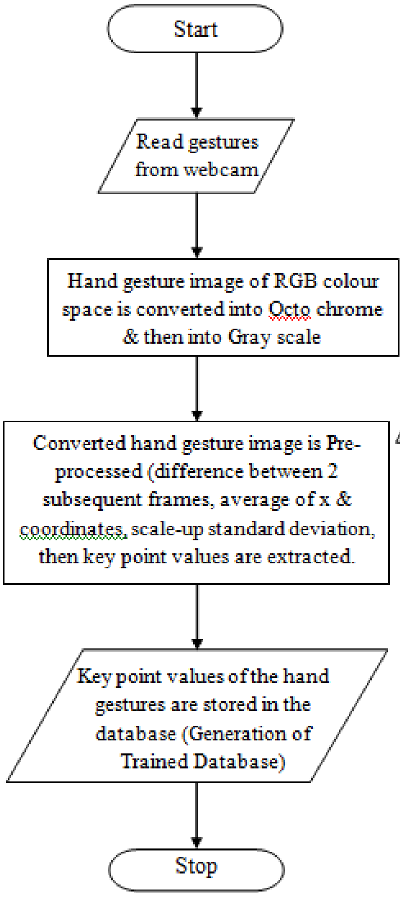
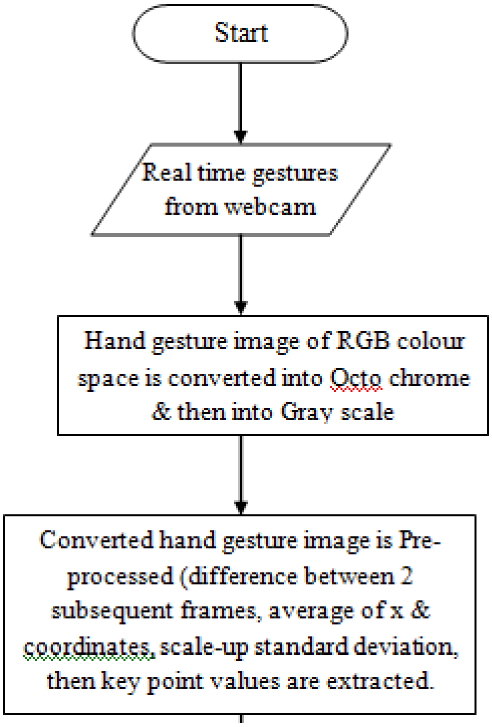

Start
Read gestures from webcam
Hand gesture image of RGB colour space is converted into Octo chrome & then into Gray scale
Converted hand gesture image is Pre-processed (difference between 2 subsequent frames, average of x & coordinates, scale-up standard deviation, then key point values are extracted.
Key point values of the hand gestures are stored in the database (Generation of Trained Database)
Stop

Figure 1. Flow chart for Training Stage.


Start
Real time gestures from webcam
Hand gesture image of RGB colour space is converted into Octo chrome & then into Gray scale
Converted hand gesture image is Pre-processed (difference between 2 subsequent frames, average of x & coordinates, scale-up standard deviation, then key point values are extracted.

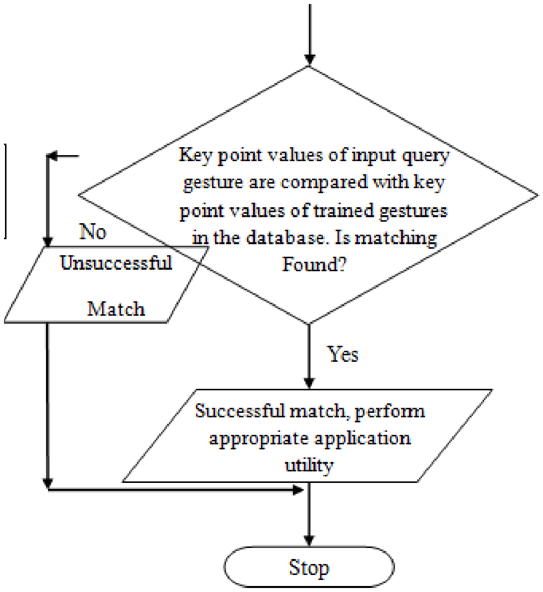

Key point values of input query gesture are compared with key point values of trained gestures in the database. Is matching Found?
No
Unsuccessful Match
Yes
Successful match, perform appropriate application utility
Stop

Figure 2. Flow chart for Testing Stage.

III. IMPLEMENTATION

OpenCV library is used for implementing the input gesture handling and manipulation from the webcam. The various OpenCV packages are used to code the application using C++ language in Ubuntu platform. The code continuously streams the real time hand gesture from the user. Number of frames per second (fps) that need to be accepted per gesture from the webcam is adjusted based on computer's memory.

Each functionality of the project is implemented as a separate module. The module for addition of new gesture is responsible for accepting the new gesture from the user along with a specific character and operation that is controlled by that gesture. Checking and deletion of an existing gesture is done.

The difference between subsequent frames are used to detect motion. Noise elimination is done so that only the moving object (hand or finger) may be interpreted in the gesture. The interpreted gesture is scanned against a set of known gestures in the database to find which gesture matches the best. The action, a system command or a keystroke, associated to the gesture is then performed accordingly.

IV. VARIOUS FEATURES IMPLEMENTED

Different features implemented in this work are as follows:
Detection of any kind of gesture which is provided in the database.

Background Elimination is done & better results if operated in a place where there is no much movement in the background.

Code is optimized to reduce noise due to change in light.

The head movements while performing the gesture are eliminated.

The above mentioned method is implemented to control the following application utilities:

1. VLC media player with following features being targeted:
Play the media.

Pause it.

Select next or previous media.

Increase or decrease the volume.

2. Controlling Libre Office Presentation with the following features:
Open an existing Libre office presentation in full screen mode.

Select next or previous slide in the presentation.

Close the presentation.

3. File Conversion:
Converting an existing document and/or presentation into pdf format.

Opening a pdf converted file.

Scroll up/down the pdf document.

Close the file.

4. Opening & Closing few Applications:
Opening & Closing of Firefox web browser, dashboard, files & folders.

V. RESULTS AND DISCUSSIONS

The implemented code detects any kind of hand gestures which is provided in the training database. Hand gestures which are not in the training data set will not be understandable to the system. In such cases the output cannot be predicted & either of the following will happen i.e., sometimes displays the message invalid gesture & sometimes accepts this as some valid gesture by performing the specific utility in the system.

Background subtraction is done & only skin color pixels are detected to make sure only hand gestures are considered. Code is not optimized to reduce noise due to change in light. So lighting variations can majorly contribute to the improper working of the code.

Analysis of the application's performance is shown in Receiver Operating Characteristics ROC) curve.

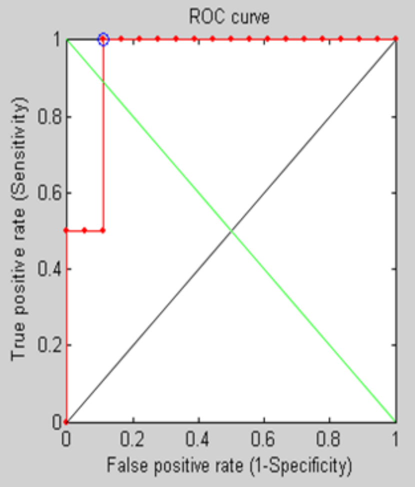


Figure ROC curve with 94% accuracy for VLC Media Player Utility of Hand Gesture Recognition.

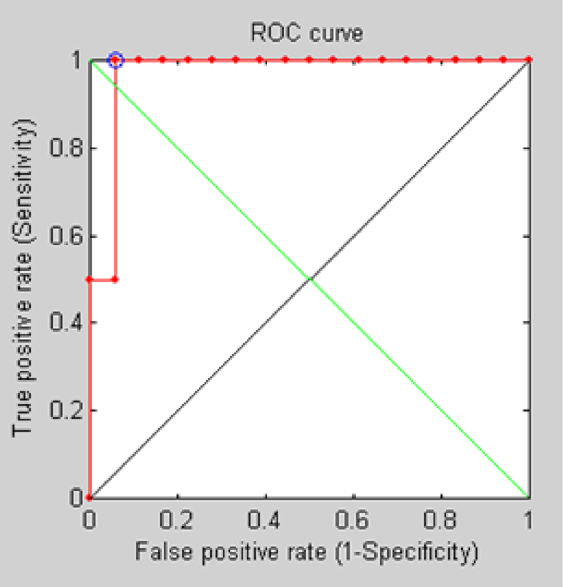


Figure ROC curve with 97% accuracy for VLC Media Player Utility of Hand Gesture Recognition.

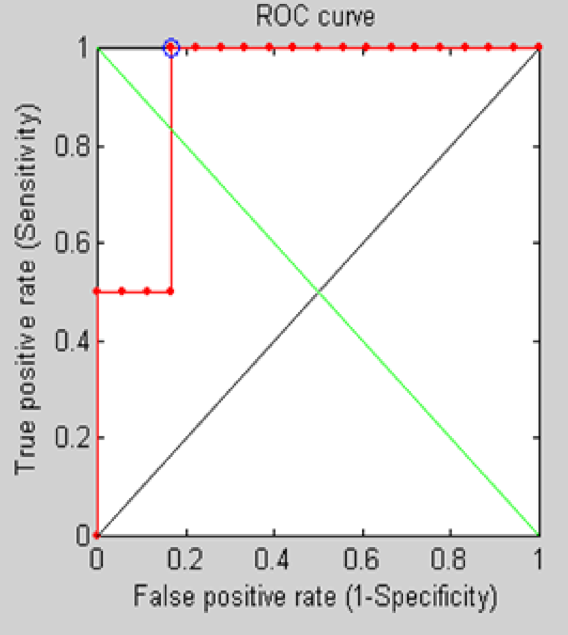


Figure ROC curve with 91% accuracy for VLC Media Player Utility of Hand Gesture Recognition.

V. FUTURE WORK & LIMITATIONS

This project has a vast arena of development. The code can be extended to incorporate mouse movements as well as still gestures. The main limitation is lighting conditions. Gestures cannot be captured when lighting conditions are too high. Similar gestures like ' c ' and ' o ' are not easily distinguishable and the program may give incorrect result in certain cases.

Further tweaks can be incorporated in the code to increase the efficiency of the gesture recognition process. Improving the code for better interpretation and recognition of the gestures and newer gestures maybe incorporated for more functionality. The graphical or natural user interface for adding and checking gestures as well as running the program can be improved, e.g. an interactive GUI can be provided rather than using terminal commands. Overall performance is good although it has some limitations.

REFERENCE [1] "2W09_live-gesture-recognition", Aamod Kore, Kshitij Singh, Nisheeth Lahoti, S S Kausik, Piyush Kumar, IIT, Bombay, 2013. | | [2] Nasser H. Dardas, Nicolas D. Georganas, Fellow, IEEE (2011), "Real-Time Hand Gesture Detection and Recognition using Bag-of-features and Support Vector Machine Techniques", IEEE Transactions on Instrumentation and Measurement, Vol. 60, No.11 | | [3] paper 2 singap J. Geethapriya and A. Srinivasan "A New Framework for Real-time Hand Gesture Detection and Recognition", MNM Jain Engineering College, Thoraipakkam, Chennai. 2012 International Conference on Information and Network Technology (ICINT 2012) IPCSIT vol. 37 (2012) © (2012) IACSIT Press, Singapore.ore | | [4]. Siddharth Swarup Rautaray, Anupam Agrawal, "A Vision based Hand Gesture Interface for Controlling VLC Media Player" Indian Institute of Information Technology Allahabad, International Journal of Computer Applications (0975 – 8887) Volume 10– No.7, November 2010 | | [5] http://www.kafu-academic-journel.info /journel /4/119/ | [6] http://blog.cedric.ws/opencv-simple-motion-detection. |