

Gesture Based Commands to Interact with the OS

Chaitra B H¹, Anupama H S², Cauvery N K³, Lingaraju G M⁴

Assistant Professor, Department of CSE, RVCE, Bangalore^{1,2}

Professor and Head, Department of ISE, RVCE, Bangalore³

Professor, Department of ISE, MSRIT, Bangalore⁴

Abstract: Human beings interact with a lot of things around them. One such interaction happens between humans and computers. As the technology and software are becoming complex day-by-day, the modes of interaction with computers are also varying. It may be through speech to text conversation, using sensors like accelerometers and gyros worn on the body of the interactor, using facial gestures, computer interfaces like keyboard and mouse, or by using external device like smart-phones or using hand gestures. Gesture based recognition system involves tracking a person's movements and predicting what they may indicate. For every gesture recognized, an application program is launched which the user can use it to interact with the OS. The programs launched may be command prompt, notepad, and MS paint or file explorer. This hand gesture recognition algorithm uses AForge.Net framework for capturing frames from the webcam in real time. The frames captured are then processed for detection of the hand. The features are extracted and the gestures are identified to interact with the OS and launch different applications. The gestures include patterns like "V", "M", STOP and Triangle.

The application program is designed in C# with .Net Framework 4 and it uses threading support to implement a parallel version of the algorithm. For the program to be compiled the compiler is set to allow unsafe code to run which is needed to accelerate the image processing steps. A frame is captured per second, the region is identified, the features are computed and hand gesture is identified and if no gestures are found, a message will be displayed. The hand gesture recognition rate using the above algorithm is 88.16%.

General Terms: Hand gesture recognition, Human Computer Interaction, Hand Posture, Thread Allocation, and Optical Character Recognition.

Keyword: Binarization, Threshold Filtering, Feature Extraction, Framework, Parallel Version, Compactness, Protrusion Ratio.

1. INTRODUCTION

Human interactions with the machines have changed over time. Thanks to the developments in software and hardware technologies that have resulted in new applications. In today's world, there is no bound on the functionality of personal computers or mobile communication appliances. For instance, intelligent traffic system promise formerly unknown levels of safety. Humanoids have begun to make their presence in countries like Japan. Robotics has changed the way people interact with computers. Robots making their way into homes across nations show the effect of technology on humans.

As the interests in Robotics have increased tremendously, human computer interaction technology is witnessing major advances. Over the decades, man-machine interface design is widely recognized as a major challenge. It represents a key technology, which promises excellent payoffs. As a major marketing factor it also secures a competitive edge over contenders. In fact, software and hardware giants alike such as Apple, Microsoft, Asus, Samsung and HP have made human computer interaction, a commercial success. Multimodal interaction by mimics, gestures, speech, video based biometrics, interactions in virtual reality, with robots and the provision of user assistance is some of the forms of interactions that are

existent today. All of these are highly innovative and dynamically developing research areas with immense potential to change the way man interacts with machines. Devices such as Microsoft Xbox Kinect have been making waves in the field of HCI and gesture recognition technology.

The main aim of gesture recognition research is to create a system which can identify certain specific human gestures and use them for device control or to convey information [6]. Gestures are frequently used for communication and are also used for everything from pointing at a person to get their attention to conveying information about space and temporal characteristics. Gesturing does not simply beautify spoken language, but is part of the language generation process. There are five different kinds of gestures: Gesticulation, Language-like gestures, Pantomimes, Emblems and Sign Languages [1]. Despite the importance of these types of gestures in normal human-to-human interactions, most research to date in HCI, focus on gestures which are less ambiguous, less spontaneous and natural and more culture specific. Gestures are an easy way to give geometrical information to the robot [2][3]. They are being extensively used in areas like service robotics which is making way to households in several locations across globe.

This paper presents a static approach to recognize human gestures using hand gesture recognition algorithm [4][7] using AForge.Net framework for capturing frames from the webcam in real time. The frames captured are then processed for detection of the hand. The features are extracted and the gestures are identified to interact with the OS and launch different applications. As shown in the figure 1.1, the gestures include patterns like “V”, “M”, STOP and Triangle. The application program is designed in C# with .Net Framework 4 and it uses threading support to implement a parallel version of the algorithm.



Fig 1.1 the gestures (from left): Stop, M, V, Triangle

2. GESTURE RECOGNITION ALGORITHM

Gesture Recognition is used for Sign Language recognition [5]. In this hand gesture recognition algorithm, the hand is the major portion in the image. It uses AForge.Net framework for capturing frames from the webcam in real time. The frames captured are then processed for detection of the hand. The features are extracted and the gestures are identified to interact with the OS and launch different applications. The application program is designed in C# with .Net Framework 4 and it uses threading support to implement a parallel version of the algorithm.

Working of the algorithm is as shown in figure 2.1. The steps include:

i. Capturing of frames using AForge.Net Framework which is designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence – image processing, neural networks, genetic algorithms, machine learning, robotics, etc.

ii. Image Binarization converts an image up to 256 gray levels to a black and white image. Binarization is used as a pre-processor before OCR. The simplest way to use image Binarization is to choose a threshold value and classify all the pixels with the values above this threshold value as white and all other pixels as black. But finding a threshold compatible to the entire image is very difficult, therefore, adaptive image Binarization is needed where an optimal threshold value is chosen for each image area.

Threshold filtering of the binary image using one upper and lower value helps identify gestures effectively.

iii. Contour Extraction is also known as **border following** or **boundary following**; Contour tracing is a technique that is applied to digital images in order to extract their boundary. It is one among the many pre-processing techniques performed on digital images to extract the information about their general shapes. Once the contour of a given pattern is extracted, its characteristics are examined and used as features to classify a given pattern.

iv. Feature Extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and is suspected to be redundant, then the input data will be transformed into a reduced representation set of features (also named as feature vector). Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the feature set will extract relevant information from the input data in order to perform the desired task.

Gestures which are recognized are thus used for launching different applications or controlling various devices.

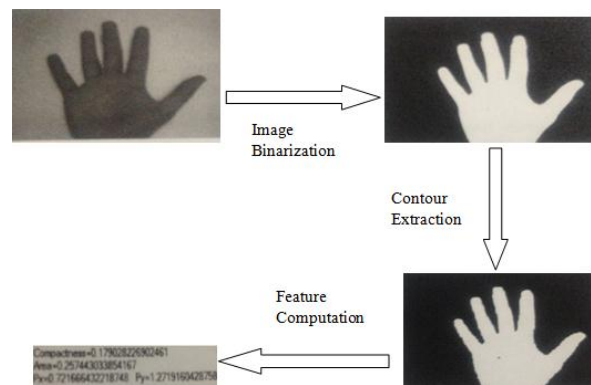


Fig 2.1: Gesture Algorithm steps

In this project, the two features that are used for identifying the gestures are: Compactness and Protrusion Ratio. Based on the values of these two parameters the hand gestures are identified.

3. PARALLEL IMPLEMENTATION

Four steps of the algorithm are implemented in a parallel form to meet the systems real time constraint. Out of which three are configured to detect the number of logical processors on the system and to start threads according to the number.

3.1 Thread Allocation for Image Binarization

To implement a parallel version of the Binarization process, which is an operation carried out on single pixels, the matrix containing the image is divided into lines and each thread processes one line. When a thread completes its task another thread is started and the process continues until the entire image has been converted. The same

technique is used to copy the resulting image back to the original bitmap and is then displayed on screen.

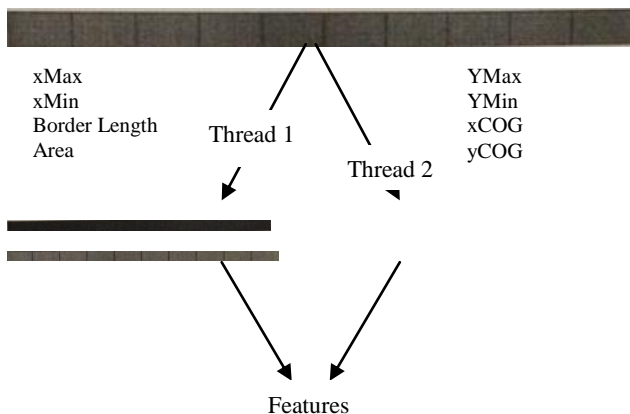
3.2 Thread Allocation for Binary Image Filtering

Some of the noise present in binary image is then filtered using two-threshold filter. A similar process to the one above is used to implement this step using a window.

3.3 Thread Allocation for Feature Computation

The features that are needed for the gesture recognition process are: Compactness and Protrusion Ratio. Compactness is a shape factor which is a dimensionless quantity, used in image analysis and microscopy. Protrusion ratio is a geometrical factor which tells us how much the image sticks out.

To compute these features, another set of features are used to first calculate the co-ordinates: xMax, yMax, xMin, yMin, length of the border, center of gravity position on the x-axis and y-axis. To accelerate this task, the set is split into groups of equivalent complexity and a thread is allocated for each one as shown in figure 3.1.



4. EXPERIMENTAL WORK

Certain gestures can be used to interact with the OS and launch pre-defined applications on the computing system. The gestures include patterns like ‘V’, ‘M’, Stop and Triangle. The actions taken upon recognition of gestures are:

1. Displaying of the recognized gesture in text format.
2. Running an application program.

The recognized text is displayed using AppendText() function. The string to be displayed is passed as a parameter to the function. The texts displayed are “Stop”, “V”, “M”, and “Triangle” depending on the gesture recognized. “No Gesture” is displayed in case no gesture is recognized.

An application program can be launched from a program using certain system calls or APIs. In Linux, this can be achieved using system () API. This system call takes a string parameter, which is the path to the executable to be launched. Similarly on Windows, we can use System () API in C++, which is similar to the Linux version. However, C# has a separate API

System.Diagnostics.Process.Start (). The .NET framework includes a class called Process, and it is included in the Diagnostics namespace. The namespace is then included using System.Diagnostics at the place where the application has to start by calling the Process.Start method. Process.Start () takes a single parameter which is the path to the executable to be launched. The syntax of the API is:

```
System.Diagnostics.Process.Start("PathToExe.exe");
```

The application Programs launched for various Gestures are as given in the Table 1.1.

Gesture	Application Program
Stop	Command Prompt
V	File Explorer
M	M S Paint
Triangle	Notepad

Table 1.1 Application Programs launched for various gestures

5. EXPERIMENTAL RESULTS

The application is written in C# using AForge.NET Framework 4 for interfacing with the webcam. For the program to be compiled the compiler must be set to allow unsafe code to run, which is required to accelerate the image processing steps. The program requires AForge references which consist of AForge.Control, AForge.Video, AForge.Video.DirectShow. Webcam drivers are installed utilizing AForge website and to run the algorithm, webcam must be selected from the list.

The start button starts the acquisition process; the image is binarized, filtered and displayed in the webcam as shown in the figure 4.1. The features are computed and the hand gesture is identified. Stopping the acquisition process is done with the stop button. For optimal results the hand should be in front of a white background and sufficient illumination is present to obtain a good contrast of colors. The threshold parameter allows tweaking the Binarization step so that only the hand is shown. The hand in the captured and the processed area should have a green outline to indicate that the correct region is identified. If the hand doesn't have the outline, then try moving the webcam, change the background or adjust the threshold.

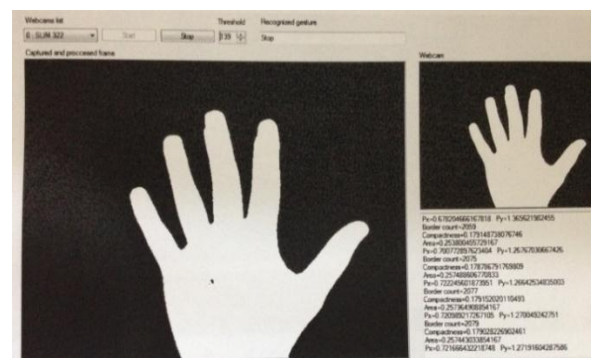


Fig. 4.1 Selection of the webcam for gesture recognition

The launching of command prompt using the ‘Stop’ gesture is as shown in the figure 4.2. The launching of MS paint using ‘M’ gestures is also shown in the figure 4.3. The launching of file explorer using the gesture ‘V’ is shown in figure 4.4.

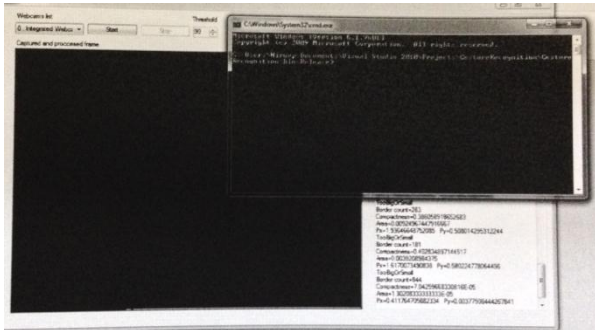


Fig. 4.2 Launching of command prompt using ‘STOP’ gesture

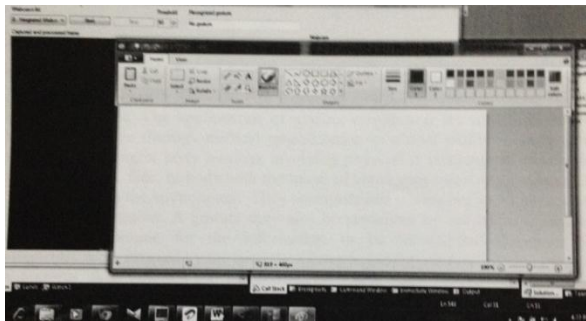


Fig. 4.3 Launching of M S Paint using ‘M’ gesture

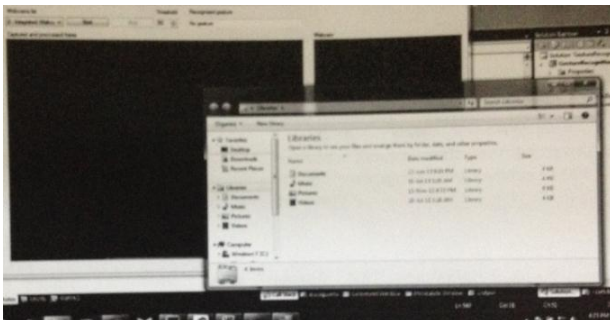


Fig. 4.3 Launching of file explorer using ‘V’ gesture

6. CONCLUSION

Gesture recognition has been a hot area of research and innovation since its inception. It is of utmost important in designing an intelligent and efficient human-computer interface. This paper presents a static approach to recognize human gestures using hand gesture recognition algorithm using AForge.Net framework for capturing frames from the webcam in real time. The frames captured are then processed for detection of the hand. The features are extracted and the gestures are identified to interact with the OS and launch different applications. The gestures include patterns like “V”, “M”, STOP and Triangle. The application program is designed in C# with

.Net Framework 4 and it uses threading support to implement a parallel version of the algorithm. This paper presents a small fraction of hand gestures that can be recognized for their use in controlling computer applications. These gestures thus can be used in intelligent ways in various applications including lie-detection, hearing aid for the disabled, tele-education, self-driving cars, etc.

REFERENCES

- [1] Gesture-Based Communication in Human Computer Interaction, International Gesture Workshop, GW’99 Gif-sur-Yvette, France, March 1999 Proceedings.
- [2] Waldherr, S,Thrun, S, Romero, R, and Margaritis,D, “Gesture Recognition”,1998, In Proceedings of AAAI-98.
- [3] Xiaolong Teng, Biani Wu, Weiwei Yu and Chongqing Liu. A Hand Gesture recognition system based on local linear embedding, April 2005.
- [4] Dipak Kumar Ghosh , Samit Ari , “A Static Hand Gesture Recognition Algorithm Using K-Mean Based Radial Basis Function Neural Network”, IEEE,2011.
- [5] S. Sarkar. Segmentation-robust representations, matching, and modeling for sign language recognition, June 2011. CVPR Workshop on Gesture Recognition, Co-authors: Barbara Loeding, Ruiduo Yang, Sunita Nayak, Ayush Parashar
- [6] Rafiqul Zaman Khan and Noor Adnan Ibraheem, “Hand Gesture Recognition: A Literature Review”, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.4, July 2012, PP. 161-174.
- [7] Devikarani Patil, Varalakshmi B.D, “Hand Gesture Recognition for MP3 Player using Image Processing Technique and PIC16F8779”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4 Issue-4, September 2014.