Course Project - CSE 518 Human-Computer Interaction

Hand-Gesture Based Browser Control

Abhinav Sharma (114950392)

1 Abstract

A hand-gesture-based human computer interface can simplify the use of browsers on computers and gadgets by, enabling users to perform repetitive browser tasks by just waving their hands in the air. Existing solutions have typically included complex configurations restricted due to resource constraints or working limited in controlled environments. There have been attempts to use 3D camera for the said task but the insufficient data on which the image processing was performed is one reason why the gesture recognition algorithms currently in use are not fit practical use. This study introduces an efficient technique for computer interfaces. The aforementioned technology is based on use of the pre-existing camera on the device being used integrated with light-weight computer vision algorithms. These algorithms will help in precise detection of hand motions, leading the user to accurately operate the browser.

2 Introduction

Gesture recognition is a branch of computer science that uses various mathematical methods to interpret human gestures. Most body action or state can serve as the source of a gesture. Gestures are physical actions involving the hands or arms that convey significant information about the gesture. To interpret these motions, numerous methods have been developed using sensors, cameras, and computer vision algorithms. Today, mechanical devices like keyboards and mouse are the foundation of most human-computer interactions. Due to its capacity to recognize human movements in a natural way, a class of techniques based on computational vision has attracted increasing attention in recent development in human-computer interaction. The images or videos captured by a camera are used as input by these techniques. The main aim of these algorithms is to identify the hand configuration at a moment to trigger a device function.

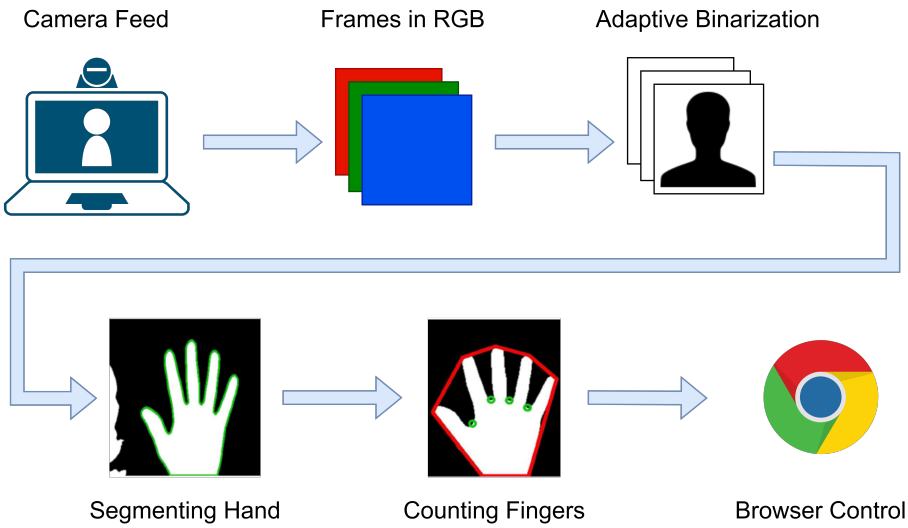


Figure 1: System Archtecture

3 Motivation

With the aim to continuously simplify and improve user experience, using hand-gestures to control system features is increasing by the day. Web browser is one of the most common applications we use everyday. When contrasted to the use of actual hardware, gesture control has many benefits. The majority of gesture control systems, however, need additional sensors or depth cameras to identify or capture the movement of gestures before a relevant signal can be sent to trigger a specific task. As a result, gesture-based system has not yet become widely used or are yet to gain popularity.

4 Problem Definition

In order to create dynamic gesture control on the computer, this research suggests a technique for a hand gesture control system that makes use of an object identification algorithm called the convex-hull algorithm together with predefined set of rules that will be triggered when a gesture is detected. For hand gesture recognition, this project makes use of a single RGB camera that is already built into the user device. The convex-hull algorithm works on the principles of computational geometry and hence does not depend on dataset for gesture recognition, hence eliminating the errors caused due to low or biased data. Algorithms to integrate hand-gesture commands with browser is through the Operating System and Selenium module in Python. Hand-gestures can be used to open or close browser windows and tabs, switch tabs. The set of operations assigned to a specific gesture can be customized i.e., the user can choose to change these operations upon initiation of system. Figure 1 is the architecture of the proposed system.

5 Related Work

In-depth research has been done on how the brain interprets hand gestures in the literature. A camera and computer vision were used in the work conducted by Freeman and Weissman [1] to identify a user's hand from across a room that enables the user to operate a television. When a user displays an open hand, an on-screen hand icon that may be used to change other graphical controls, such as a volume slider, will appear. Users preferred this substitute for the real remote control, and the authors found that the on-screen hand's input was helpful in guiding the user. Users discovered that holding their hand up for extended periods of time to engage the various controls was exhausting. Gorilla arm is a term used to describe the user weariness that is typical of gesture-based interfaces.

Various methods have depended on the creation of a 3D picture utilizing many cameras in order to identify and track hand movements [2] [3] [4]. These systems required a complex installation procedure that needed to be done with care since calibration factors like the separation between the cameras were crucial to the triangulation algorithms being employed. Since a lot of video data had to be analyzed in real-time and stereo-matching frequently fails on scenes with little to no texture, these techniques were also computationally costly. Such systems would ultimately be useless outside of their unique lab settings.

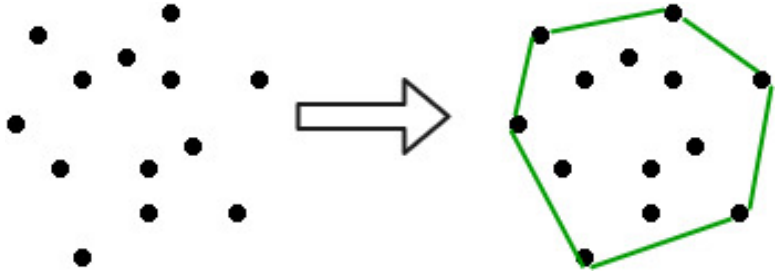


Figure 2: Mathematical depiction of Convex-Hull Algorithm

6 Methodology

On receiving the video feed from the camera of user device, first step to detect hand-gestures is for the Python program to separate this video feed into frames. The number of frames a video can be segregated into depends on the quality of camera, for example a 30 Frames-per-second camera will divide a 5 second video into 150 frames (5 * 30). Although, we are using image processing techniques to improve the image quality, hence eliminating the need of superior quality cameras.

We have a region-of-interest on the video camera feed, by region-of-interest we mean a box. When a user wants the program to trigger an action for browser control, the user is supposed to bring their hand inside this region-of-interest for the algorithm to detect hand and then proceed with further execution. As the frames are initially in the Red-Gree-Blue color format, to reduce noise first these frames are converted into black and white and then adaptive binarization algorithm is used to eliminate the disturbance caused though background and focus on the hand of user. Also, we declare a pixel range for hand of the user, this helps in better detection of hand inside the region-of-interest box and also this pixel range helps to avoid use of other Deep Learning algorithms to detect hand. These Deep Learning algorithms would have been Convolution Neural Network based and processing each frame of video through this model would be computationally expensive and in-turn would have slowed down the system.

Then the hand is segmented from the complete frame, as we only need that part for further processing. We have a pre-defined set of rules for number of fingers and customizable rules for the website that the user wants to open. Convex- Hull is used to count the fingers and then depending on the count one of the fingers the algorithm detects a task is executed like opening or closing a website or scrolling up or down. Selenium Python module is used to trigger these tasks. Selenium is a tool for controlling web-browsers through programs and performing browser automation.

Convex-Hull Algorithm

The smallest convex polygon that includes all the given points is known as a convex hull. This is depicted in the Figure 2.

Figure 3: Input image from camera feed

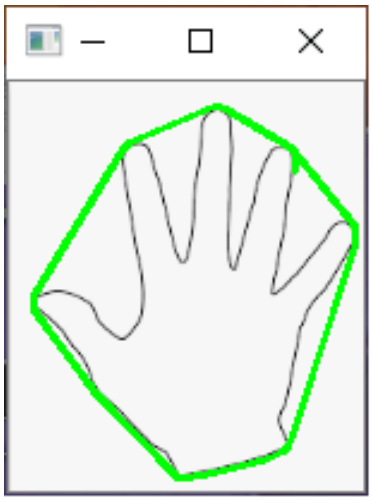


Figure 4: Result of Convex-Hull Algorithm on Figure 3

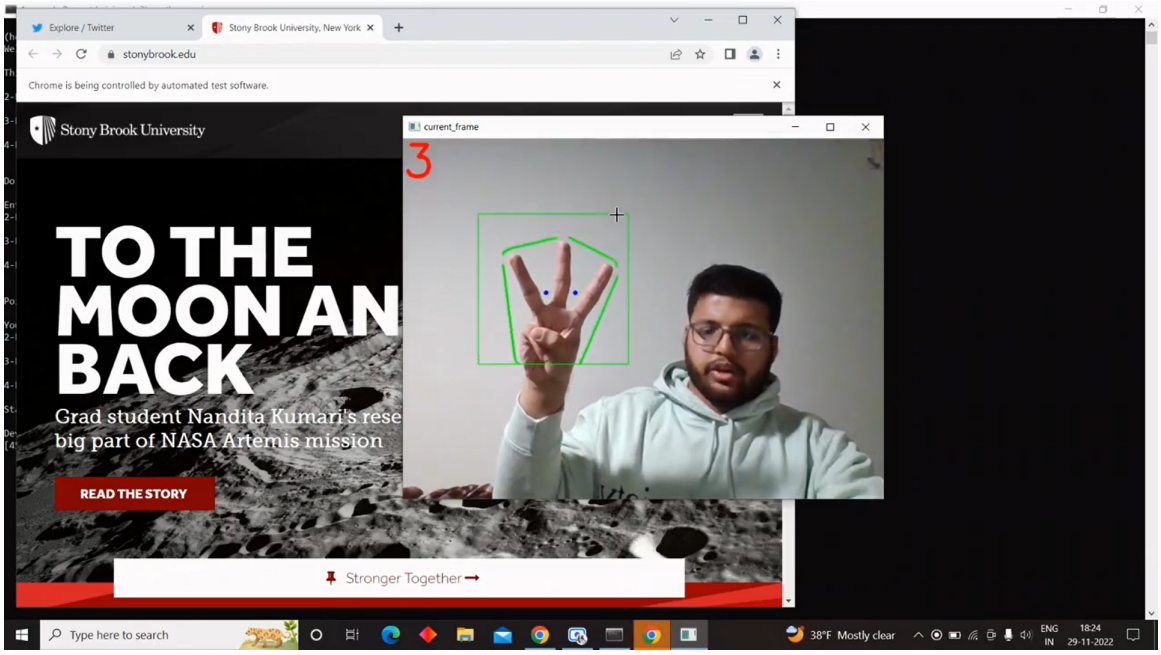
Now as we already had a grayscale image of segmented hand (see Figure 3), we then use OpenCV's internal modules to get the Canny Edges of this hand. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. The next step is to find the contours of this image which is performed though the Find Contours function of OpenCV. We only consider the external contours of the image, as we aim to find the number of fingers in our image and not the internal contours that are within the area of hand. After this we iterate to find all the possible contours in the image and start creating convex-hull for all these contours. This is depicted in Figure 4.

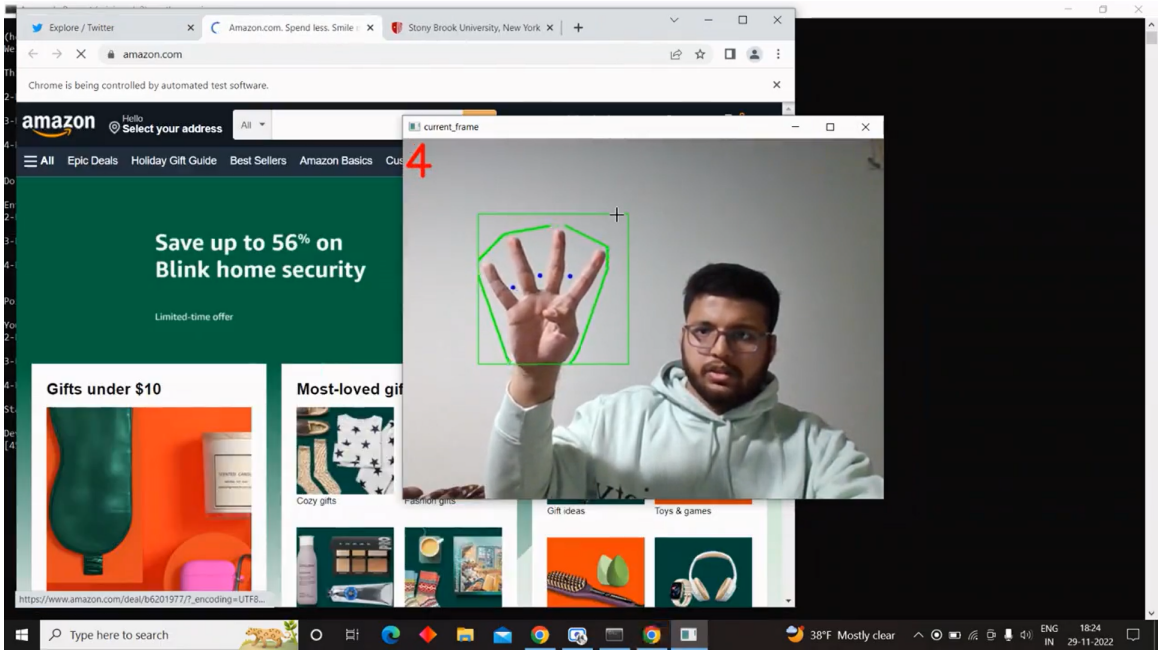
Any divergence of the contour from its convex hull is known as the convexity defect. So, the gap between our any two fingers would be a convexity detect and so we count these number of defects in order to find the number of open fingers. Then we add one to this number of defects and pre-define the tasks we need the browser to perform.

Once the system is initiated the user gets the option to continue with the previous user defined gesture set or input a new one. The system has optimizations like switching tab if the user tries to open a website that is already running on browser. The table below mentions the set of gesture for a given user.

Number of Fingers	**Task**
0	Scroll-Down
1	Scroll-Up
2	Open "stonybrook.edu"
3	Open "amazon.com"
4	Open "twitter.com"
5	Close current tab

User Interface





7 Validation

For validation, I asked 7 students to use the system as a test to robustness under different conditions. The experiments performed was to test mainly two things:

1. Detection and Tracking of hands.
2. Opening and closing of multiple tabs on browser.
3. Frequently switching hand gestures.

Experiments:

1. For this task the users sat in different backgrounds and different lightning conditions. Conversion of image to grayscale and using adaptive binarization made sure that any background other than plain background like a window or wardrobe and dim lighting conditions did not affect the detection of hands or counting of fingers. Even if the user kept the position of hand such like that the hand only occupied less than half area of the region-of-interest the detection of accurate. These three tests made sure that the hand-gesture detection and tracking module of the system has completely accurate output and in real-time.
2. Every user in the test had the option to select their preferred set of websites on initiating the system and test any combinations of sequence of hand-gestures. These combinations include, opening a website when it is already running on another tab, switching tabs, re-opening a website after closing it once and re-opening the browser after closing the browser once.
3. This is one of the most important tests, as triggering a task on every gesture detected will result in stack overflow and crashing of the system. So, the system included a counter that will ensure that if a particular gesture is continually detected for fifty frames, only then a task will be triggered. This helps to eliminate false positives of finger counting and problem of frequently changing hand gesture.

8 Future Work

1. Using an image processing technology or hand gesture control, we can also enable the system to play/pause any audio or video that is now playing on the page. So, using only a few hand motions, the video or audio on the current page can be played.

2. People with visual impairments and physical limitations would benefit greatly from a speech recognition system combined with hand gestures since it would fully eliminate the need for a keyboard, but would be very computationally expensive.

9 Conclusion

The control of computer applications, such as browsers, may be considerably enhanced by the use of gestural inputs. Using gestures is comparatively easier and faster than using mouse and point on a button and clicking. Using gestures as a human-computer interface has many advantages over other traditional input devices, such as keyboard or mouse or joystick, as it will no longer damage the device life as there is no device wear and tear, the input will be independent from the user's known language, and an intuitive human-computer interface. Therefore, we implement use of hand-gestures for browser control.

References

[1] W. T. Freeman and C. D. Weissman, "Television Control by Hand Gestures", Proc. of Int. Workshop on Automatic Face and Gesture Recognition. IEEE Computer Society, pp. 179-183, 1995.

[2] Q. Cai and J. Aggarwal, "Tracking Human Motion Using Multiple Cameras", IEEE Computer Society, vol. 3, pp. 68-72, August 1996.

[3] A. Utsumi, T. Miyasato and F. Kishino, "Multi-Camera Hand Pose Recognition System Using Skeleton Image", IEEE Computer Society, pp. 219-224, July 1995.

[4] Z. Jun, Z. Fangwen, W. Jiaqi, Y. Zhengpeng and C. Jinbo, "3D Hand Gesture Analysis Based on Multi-Criterion in Multi-Camera System", IEEE Computer Society, pp. 2342-2346, September 2008.