International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9, Issue-5

Dynamic Gesture Recognition System to Control Media

Asnath Victy Phamila Y, Raunak Gopal, Geetha S

Abstract: ***Gesture control technology is developing quickly and changing many aspects of daily life. Gesture control devices evolved from expensive primitive input devices to affordable devices capable of fine detail recognition. These devices are now used in a much wider range, from research experiments and prototypes to day-to-day commercial products. In this paper, a hand swiping algorithm to control media on personal computers is presented. The algorithm aims to be accurate, without the burden of high computational complexity. The paper begins with an introduction to gesture control and surveys existing work in the field. The main algorithm which tracks the users hand to control the volume is explained in detail in the methodology section. A Raspberry Pi board along with a Logitech HD webcam will be used to test the algorithm. The results are discussed in the experiment section, followed by the constraints and conclusion.***

Keywords: ***Accurate, Computational Complexity, Gesture, Swiping***

I. INTRODUCTION

In 2011, the world was introduced to the first mainstream commercial gesture controlled device – the Microsoft Kinect. The Kinect enabled the user to provide input to video games via body movement, which it picked up using its infrared camera, thus eliminating the need for a controller. While the Kinect secured high sales figures and opened the world to gesture technology, it was hardly the first gesture control device.

Initial research on gesture control began in the 1980's, with the invention of the data glove. The glove was wired, and incorporated certain sensors in key areas such as the joints of fingers. Movement of these joints were tracked and mapped to unique gestures, which was interpreted by the computer. Over the years, accelerometers, infrared cameras, fiberoptic bend sensors are technologies that have been used to enable faster, more accurate and also wireless gesture recognition. It must be noted that as gesture controls techniques become more diverse, the need for improved preprocessing techniques to extract fine details became necessary.Gesture control algorithms are classified into static and dynamic algorithms based on movement. Gestures such as the thumbs up, peace sign which do not require movement are classified as static gestures. Gestures which employ movement such as the palm swiping algorithm implemented in this paper are dynamic gestures. While it may not be as popular as touch or voice, gesture controls have become a commonplace occurrence in today's world. It's uses can be seen in smart devices, vehicles, home automation, gaming and in sign language translation. Gestures can be performed quickly while also feeling natural, making the technology appealing.

Revised Manuscript Received on May 15, 2020.
* Correspondence Author
Asnath Victy Phamila Y*, Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India, Email – asnathvicty.phamila@vit.ac.in
Raunak Gopal, Student, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India, Email – raunak.gopal2016@vitstudent.ac.in
Geetha S, Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India, Email - geetha.s@vit.ac.in

II. RELATED WORKS

Research into Gesture controls began in the 1980's but the technology has truly become popular in the last decade. This section will take a look at some of the work published by other researchers. IBM developed a camera based gesture interface to control home appliances for disabled people. In 2004, a cheap vison based input device - Visual Touchpad was introduced to control PCs, laptops, public kiosks using two handed gestures. One of the most recent advancements in gesture control systems comes with Google's project Soli, which uses an 8mm x 10mm radar chip to provide a wide array of gestures to control the new Pixel 4 smartphone.

A gesture control system to control a graphic editor tool which involves tracking hand movements has been proposed in [1]. [2] also proposes a method to control a graphic tool by implementing 12 different dynamic gestures which involve drawing common shapes like a rectangle, triangle, circle etc. [3] proposed a system unaffected by environmental changes, which implemented 3D pointing gestures from binocular view for Human Computer Interaction. In [4], a system to track hand gestures using multivariate Gaussian distribution was presented. Keskin et. al. [6] modelled a joint distribution to classify gestures by training an SVM classifier after dividing the hand into 21 regions. Zeng et. al. [7] built a system to control a wheel chair using 5 hand gestures and 3 compound states, which works both indoors and outside.

III. PROPOSED METHODOLOGY

The methodology section can be divided roughly into two parts – The first part explains the pre-processing techniques that were used to eliminate the background and isolate the user's hand. The second part of the section contains the main algorithm, which is used to track the movement of the isolated hand to increase/decrease the volume.

The system has been coded in Python 3.6 with OpenCV being the main image processing library. A Logitech HD webcam connected to a Raspberry Pi board has been used for Video Capture.

A. Preprocessing Techniques

The aim of this step is to extract the contour/outline of the user's hand i.e. to separate the hand from the surroundings. There are various techniques to do this, but the technique used will be combining two fairly known methods – HSV segmentation and Background subtraction. A video is a sequential collection of frames (images). The individual frames in the video captured by the camera are in the sRGB (standard Red Green Blue) color space. The color of every pixel in a frame is represented by a combination of these three primary colors. While RGB is the de facto model in most electronic media, it is heavily influenced by the luminance of a scene. This is where HSV comes in. HSV is a different color space where pixels are represented by a combination of their Hue (dominant color), Saturation (intensity) and Value (brightness). A unique characteristic is that one can differentiate image luminance from chroma/color information in HSV.

After obtaining the HSV frame, the next step is to extract all the pixels whose value is in the range of human skin tone. Doing this involves applying a mask with the min and max values of HSV values of the human skin tone to the frame. Now that the skin toned value pixels have been identified, the HSV image is converted to Greyscale, where ideally the hand pixels should be white and the background pixels should be black. However, the application of only HSV segmentation has one problem – It does not get rid of the background objects which are skin tone colored! To solve this problem, HSV segmentation is combined with Background Subtraction.

Background Subtraction is a technique to eliminate stationery objects. Since the camera is static, and the object of interest (the user's hand) is never completely still, the conditions to apply Background subtraction are satisfied. While there are several Background Subtractors, the one that is used is the Gaussian Mixture Based BackgroundSubtractorMOG2. Once the foreground image is obtained (in greyscale), a Bitwise-And is performed with the HSV segmentation frame and the foreground frame.

After the application of the mask, there may still be a couple of outliers (which cause holes in the hand). To fill these holes involves performing dilation followed by erosion. Dilation, as the term suggests is a process where a pixel is made high (1) if most of its neighboring pixels are high. The dimensions of the neighboring pixel matrix (kernel) is pre-determined (in this case it is 5 x 5 matrix). Erosion is the exact opposite of dilation, where a pixel is made low (0) if most surrounding kernel pixels are low. Dilation followed by Erosion is called Closing, and is performed using the morphologyEx function in OpenCV.

Once the Closing process is complete, the intermediate output is a zero defect greyscale image where only moving skin toned color objects are white. There can be multiple such object besides the user's hand (like another person in the frame, the user's face etc.), which have to be removed. To extract only the user's hand involved finding the contour of all such objects, and considering only the largest contour (the contour with the maximum number of points). Since the user's hand is the object closest to the camera, it's contour will have the most number of points.

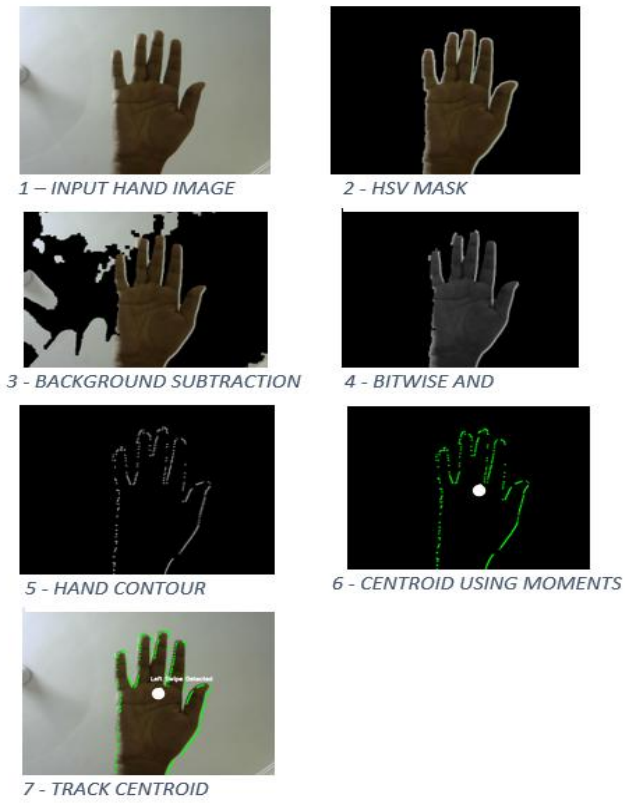


Figure 1. Diagram of the steps in the Algorithm

B. The Main Centroid Tracking Algorithm

Now that the user's hand contour is obtained, the goal is to track its movement. One way to approach this would be to track the movement of every single point in the contour, but would result in poor performance on less powerful computers. Since it is known that the human hand is a solid object, whose shape is fixed, tracking the centroid of the hand would give the exact direction in which the hand is moving in, while having low computational complexity. So the first part of this step is to find the co-ordinates of the centroid given the hand contour. To do this, one can use Image Moments, a Moment being a weighted average of pixel intensities. The Image Moments can be found using the inbuilt moment function in OpenCV. The centroid of the hand can be calculated form the Moments using the formula –

$$C_x = M_{10}/M_{00}$$
$$C_y = M_{01}/M_{00}$$

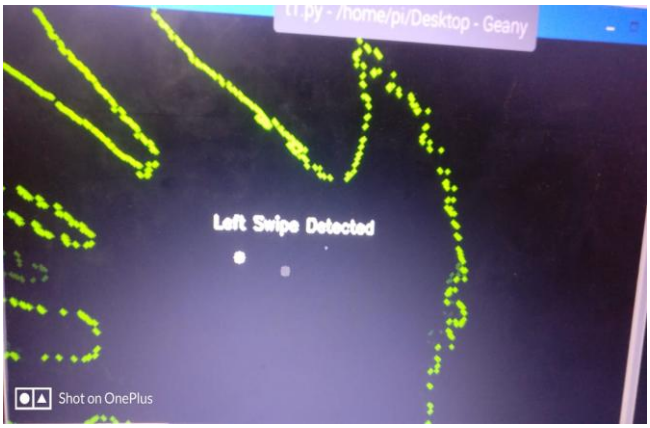
where C_x is the x-coordinate and C_y is the y-coordinate and M denotes the Moment The centroid can be calculated in this way for each new frame. Now the movement of the centroid needs to be tracked. To do this involves making use of the Queue (First In First Out) data structure, which has a fixed size (in this case a length of 10). When a new frame is captured, the co-ordinates of the hand's centroid is found and pushed it into the queue

(The leftmost element is popped out). For explanation purposes, the algorithm will be focusing only on the horizontal movement of the hand (left/right swipe only), and thereby using only the x-coordinates of the centroid. The queue will be iterated through starting with the second element (index number 1).

The x-coordinate of the current element in the queue is subtracted from the x-coordinate of the previous element, resulting in a Difference Value. If a Difference Value is positive, it means that the centroid/hand has moved to right from the previous frame and current frame and if it is negative it means the centroid moved to the left. This will provide 9 Difference Values in total (since the queue length is 10), which are all stored in a separate list. If majority of the Difference Values are positive, it is concluded the user has swiped right. If majority of Difference Values are negative, it is concluded the user has swiped left. The left and right swipe can be each mapped to a particular action, in our case to decrease and increase the volume respectively.

IV. EXPERIMENT

(a)



(b)

(c)

Figure 2: (a) Image of the Experiment Setup (b) Left Swipe Output Image (c) Right Swipe Output Image

For the experiment, a Logitech HD (720p60fps) webcam and portable earphones were connected to a Raspberry Pi – 3 Model B board running Linux. The left swipe of the hand was mapped to decrease the volume (by 2%) and a right swipe to increase (also by 2%). The algorithm was coded in Python 3.6 with OpenCV as the Image Processing library. The length of the Queue was set to 10. It must be noted that at least 10 frames must be captured, before a swipe is registered (With a 60fps camera, this results in a very small delay time of $1/6^{th}$ of a second). The program was started while a Youtube music video was running in the background. For testing 5 different people with different skin colors were used with varying background and adequate lighting. Out of a 100 swipes performed, 91 were detected and registered correctly and 9 were undetected. There was never a case where a left swipe was registered as a right swipe or vice versa.

V. CONSTRAINTS

The challenges/constraints of the implemented gesture control algorithm implemented are –

- The accuracy of the output of the algorithm may vary based on factors like age, gender, race, height and other physical factors. While this study does test out the algorithm in various conditions successfully, there probably is inherent bias in the pre-processing techniques which may result in skewed outputs.
- While the algorithm works in a low powered computer, it depends heavily on additional peripherals such as a good quality HD webcam. This may push away potential users who do not wish to make an investment
- The algorithm was tested on 5 people with varying skin colors in different conditions and was found to work accurately. While this shows the capability of the algorithm, a larger group of test subjects is required for mimicking the heterogeneous real world population
- There is a small startup time in the algorithm before a swipe is registered, which is proportional to the length of the Queue used and inversely proportional to the Fps of the camera.
- The gestures can be performed by anyone, providing no security and are also easy to perform accidently

VI. CONCLUSION

An extensive study of gesture controls was taken up. Using this knowledge, an accurate yet computationally cheap palm swiping algorithm by tracking the centroid was presented. An experiment was performed to map the horizontal swiping gestures to control the volume on a personal computer. Since the algorithm is light on resources, it can be extended to work on other computers such as those found in vehicles, gaming consoles etc. In its current state, the palm swiping algorithm only works for horizontal swipes. It can be easily made to work for vertical swipes by changing to the y-axis.

A future system can be designed to integrate horizontal (x) and vertical (y) swipes by additionally calculating the angle of the swipe. As stated before, the algorithm registers inputs irrespective of the person trying to perform it. A layer of security can be added to the algorithms by combining it with hand geometry identification systems, to ensure only members input gestures are considered.

REFERENCES

1. S. Mitra, and T. Acharya. (2007). "Gesture Recognition: A Survey" IEEE Transactions on systems, Man and Cybernetics, Part C: Applications and reviews, vol. 37 (3), pp. 311- 324, doi: 10.1109/TSMCC.2007.893280
2. Min B., Yoon, H., Soh, J., Yangc, Y., & Ejima, T. (1997). "Hand Gesture Recognition Using Hidden Markov Models". IEEE International Conference on computational cybernetics and simulation. Vol. 5, Doi: 10.1109/ICSMC.1997.637364
3. Guan, Y., Zheng, .M. (2008). "Real-time 3D pointing gesture recognition for natural HCI. IEEE Proceedings of the 7th World Congress on Intelligent Control and Automation WCICA 2008, doi: 10.1109/WCICA.2008.4593304
4. Mokhar M. Hasan, Pramod K. Mishra, (2012) "Features Fitting using Multivariate Gaussian Distribution for Hand Gesture Recognition", International Journal of Computer Science & Emerging Technologies IJCSET, Vol. 3(2).
5. Khan, Rafiqul Zaman & Ibraheem, Noor. (2012). Hand Gesture Recognition: A Literature Review. International Journal of Artificial Intelligence & Applications (IJAIA). 3. 161-174. 10.5121/ijaia.2012.3412.
6. C. Keskin, F. Kirac, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in Computer Vision Workshops (ICCV Workshops), pp. 1228-1234, 2011.
7. Zeng, Jinhua & Sun, Yaoru & Wang, Fang. (2012). A Natural Hand Gesture System for Intelligent Human-Computer Interaction and Medical Assistance. Proceedings - 2012 3rd Global Congress on Intelligent Systems, GCIS 2012. 382-385. 10.1109/GCIS.2012.60

AUTHORS PROFILE

Dr Asnath Victy Phamila Y

Dr.Asnath Victy Phamila Y, is Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India. She holds M.E and Ph.D degree in Computer Science and Engineering from Anna University. Her research area includes image processing, wireless sensor networks, network security and cyber physical systems. She has around 13 years of academic and 3 years of industry experience.She has several research papers to her credit which have been published in reputed journals. She also serves as a reviewer in reputed journals. Her fields of skills and expertise include Signal, Image and Video Processing, Medical and Biomedical Image Processing, Image Compression, Data Compression, Image Coding and Image Fusion.

Raunak Gopal

Raunak Gopal is a student who has completed his 3rd Year B.Tech, Computer Science & Engineering, at Vellore Institute of Technology (VIT, Chennai Campus). He has an excellent academic record and a keen interest in practical exposure in the field of Artificial Intelligence, Deep Learning, Analytics and Big Data.
Raunak is currently doing his internship in a Global Technology Company which is leader in Hardware, Software and Cognitive computing.
In 2019, Raunak was part of a winning team that worked on a Drone using a Pixhawk Controller for locating and identifying people in situations of emergency such as floods, earthquakes etc. This Project has been awarded "Best Project in Technical Answers for Real World Problems by VIT Chennai".
In 2018, in his summer internship at Indian Institute of Technology, Madras, he had worked on a Machine Learning Project for development of a new algorithm for "Hypergraph Representation Learning using Deepwalk".
In 2017, Raunak had worked on the development of a new algorithm for "Predictive Driver Behaviour" for "Instantaneous Driver Intent" at one of India's largest Commercial Vehicle Manufacturers as part of his summer internship.
He has developed serveral other models using Machine Learning and AI for predictive analysis at College.
Raunak is proficient in the use of the latest programming languages and wants to pursue a career in Machine Learning, Artificial Intelligence and Big Data to provide disruptive solutions to society. A voracious reader, Raunak also has interests in music, movies and history

Dr Geetha S

Dr.S.Geetha is a Professor in School of Computing Science and Engineering, VIT University, Chennai Campus, India. She has received the B.E., and M.E., degrees in Computer Science and Engineering from Madurai Kamaraj University, India in 2000 and Anna University of Chennai, India in 2004, Ph.D. Degree from Anna University in 2011, respectively. She has rich teaching and research experience. She has published more than 80 papers in reputed International Conferences and refereed Journals. Her research interests include steganography, steganalysis, multimedia security, intrusion detection systems, machine learning paradigms and information forensics. She is a recipient of University Rank and Academic Topper Award in B.E. and M.E. in 2000 and 2004 respectively. She also holds the Certificate of Appreciation from IBM in 2009, 2010 for Great Mind Challenge, Mentor IBM Academic Initiative Program. She is also the proud recipient of ASDF Best Academic Researcher Award 2013, ASDF Best Professor Award 2014, Research Award-2016 and High Performer Award – 2016, from VIT University, ISCA Best Poster Award - 2018.