Abstract

Human Computer Interaction (HCI) deals on how human and computer interacts each other, hence Human, Computer and Interaction are three major components to study under HCI. Users are diverse in nature according to their physical, mental, cognitive, social, economic conditions and backgrounds, so interaction with digital media should be developed in such a way that diverse users can use digital technology as per their capacity. Gesture is the non-verbal way of communication and plays a prominent role in interaction. With methods of Gaussian blur, Skin color detection, Contour and Convex Hull, the application is able to recognize thumbs up and down and zoom in and out the windows screen.

Keywords: HCI, Interactions, Gesture recognition, Skin color detection, Gaussian blur, Contour and Convex hull, Python, OpenCV

Table of Contents

Introduction

Human-Computer Interaction (HCI) is the study of how the people interact with computer and to what extent (Carroll). It is also the study of the computers are or are not developed for the successful interaction with human beings. HCI is about designing a computer system that supports people so that they can carry out their tasks safely and productively (Tomayess Issa, 2015). In HCI, there are major three components, which are the Users, Computer and the interaction. Users are the one of the components of the system who uses the computer, might be individual or a group. Users are diverse in nature according to their physical, mental, cognitive, social, economic conditions and backgrounds. The other component is the computer which can be ranging from desktop computers to the large scale of digital system(Jones). And the third part is the interaction which is the intuitive conversation between the users and the computer. Therefore HCI is about the understanding what it means to be a user of a computer system and to create the related products and services that work seamlessly(I. D. Foundation).

There are two means of communication exists in the world, which are verbal and non-verbal means of communication. Gesture is a non-verbal means of communication with the movement of limb or body as an expression of thought of feelings(Imran Hussain, 2014). The objective of the project is to detect the thumbs up and thumbs down of the hand and do the necessary functions such as emulating up and down arrow and zooming in and out of the image. So, this project relates to the non-verbal means of communication which is the gesture of hand. The gesture of hand can be divided into two types. First one is the global motion of the entire hand and second one is the local motion, i,e, only the fingers move (Imran Hussain, 2014). In HCI, main advantage of using visual input is the possibility of communication

without need of any physical contact with the device to be controlled (Imran Hussain, 2014). To achieve the objective of the project, following methods of HCI are used:

1. **Skin color detection:**

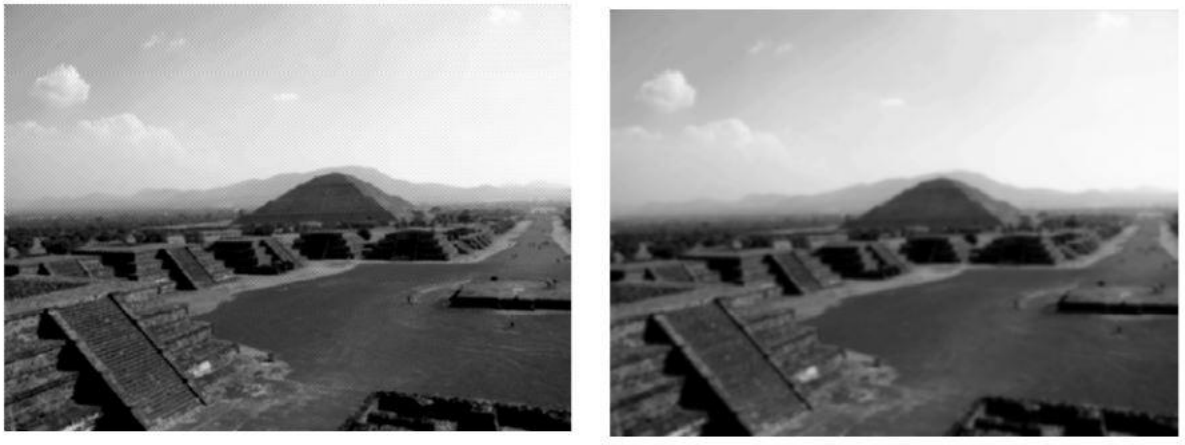
Skin color detection is the process of separation between skin and non-skin pixelto find out the skin regions in an image (Shaik, Ganesan, Kalist, Sathish, & Jenitha, 2015). The first or basic step to detect the human face and limbs is this method. The detection and segmentation of skin regions in an image is used in many applications like video surveillance, human motion monitoring, HCI, face detection, gesture detection, etc. There exist two types of skin detection methods which are, pixel and region based (Shaik et al., 2015). In the pixel-based skin detection method, each pixel is classified as either skin or non-skin individually from its neighbour, whereas in region-based method, the skin pixels are spatially arranged to enhance the performance (Shaik et al., 2015). One of the pixel-based skin detection methods is the skin color based segmentation. However it is very popular method of face detection and tracking, there occurs mainly three problems which are, what colorspace to choose, how exactly skin color distribution should be modelled and what will be the way of processing of color segmentation results for the face detection (Vladimir Vezhnevets, 2003).

2. **Gaussian Blur**

Gaussian blur also known as Gaussian Filtering is one of the technique in image processing to reduce noise and details by using the Gaussian function. Mathematical function in 2D form of Gaussian function is:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Since the Fast Fourier Transform (FFT) of Gaussian is another Gaussian, so the gaussian filter will reduce the high-frequency components of image so it is a Low Pass Filter (LPF). The Gaussian filter works by using the 2D distribution as a point spread function which is achieved by convolving the 2D Gaussian distribution function with the image, then need to produce a discrete approximation to the Gaussian function (Auckland, 2010). The distribution will approach very close to zero about three standard deviations from mean, 99% of the distribution lies within 3 SD, which means we can normally limit the kernel size to contain only values 23 (Auckland, 2010). It is used to remove the noise and details, but it is not effective to remove the “salt and pepper noise”. It is a common first step to edge detection.



Before application of Gaussian blur After application of Gaussian blur

figure 1: Gaussian blur

3. Contour and Convex hull

A contour is a curve for a two variables functions along which the function has a constant value (Amiraj Dhawan, 2013). A contour joins points above a given level and of equal elevation and a map illustrate the contour using lines which shows the steepness of slopes and valley and hills functions gradient is always perpendicular to the contour lines (Amiraj Dhawan, 2013). Contours are straight lines or curves describing the intersection of one or more horizontal planes with a real or

hypothetical surface with which is drawn around the white blob of the hand that is found out by thresholding the input image as there can be possibilities that more than one blob will be formed in the image due to noise in the background (Amiraj Dhawan, 2013). That's why the contours are drawn on such smaller white blobs and considering all blobs formed due to noise are small, thus the large contour is considered for further processing specifying it is the contour of hand (Amiraj Dhawan, 2013).

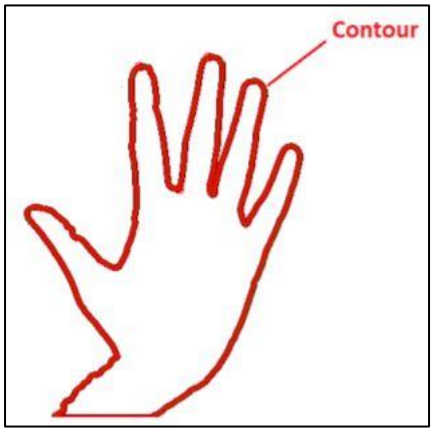


figure 2: Contour detection

The convex hull of a set of points in the Euclidean space is the smallest convex set that contains all the set of given points which is drawn around the contour of the hand, such that all contour points are within the convex hull which makes an envelope around the hand contour (Amiraj Dhawan, 2013).

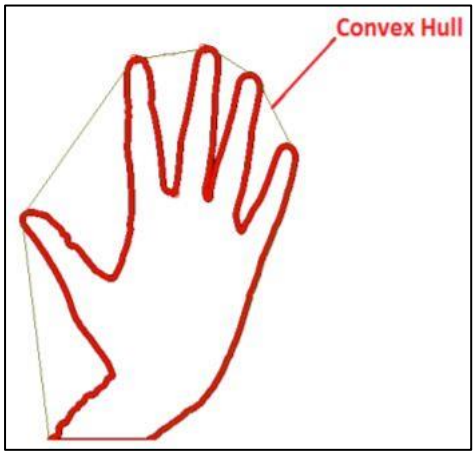


figure 3: Convex hull of human palm

Methodology

For this project which is the detection of human thumbs up and down and zooming the screen image when it is thumbs up and zoom out when it is thumbs down, we have used skin detection to detect the hand of the person, convex hull and contour detection method to distinguish the hand. Then we have used the angle determination to detect the thumbs up and down, after detection of thumbs up and down, then the display window is zoom in and zoom out. The detailed methodology is discussed as follows:

Step one:

Firstly, the camera of the operating device, in this case, Personal computer is opened which is done first while executing the code. The camera reads the object in real time or in other words, takes the video in its frame in real-time. The real-time image is converted into a grayscale image for further image processing.

```
camera = cv2.VideoCapture(0)
while camera.isOpened():
    reb, img = camera.read()
    cv2.rectangle(img, (300, 300), (150, 150), (120, 212, 120), 0)
    anonymous_img = img[150:300, 150:300]
    grey = cv2.cvtColor(anonymous_img, cv2.COLOR_BGR2GRAY)
    Am = (15, 15)
```

Step two:

After getting the video on its frame, and conversion of RGB video to the grayscale image, filtering is needed to be applied to filter out the noises of any kind. For this Gaussian filtering is applied. For any filtering, need to fix a threshold factor, for which it is as follows.

```
Bi = cv2.GaussianBlur(grey, Am, 0)
_, thresh = cv2.threshold(Bi, 100, 100, cv2.THRESH_BINARY_INV +
cv2.THRESH_OTSU)
```

Step three:

After filtering and thresholding, the program now has the image which is ready for the further actions. Now the contours are detected using the following functions and the convex hull is drawn around of the hand, such that all the contour points are within the convex hull, which makes the envelope around the hand contour.

```
on = max(contours, key = lambda x: cv2.contourArea(x))
hull = cv2.convexHull(on)
drawing = np.zeros(anonymous_img.shape, np.uint8)
cv2.drawContours(drawing, [on], 0, (0, 255, 0), 0)
cv2.drawContours(drawing, [hull], 0, (0, 0, 255), 0)
hull = cv2.convexHull(on, returnPoints = False)
defects = cv2.convexityDefects(on, hull)
countdefects = 0
cv2.drawContours(thresh, contours, -1, (0, 255, 0), 3)
```

Step four:

After drawing convex hull, next step is to determine the angle of the thumb. For this purpose, we used the following formula and used cosine method to determine the angle of the thumb and declared the angle of the thumb based on the position of the thumb. We have used value 90 to determine if the thumb is in up position of not.

```
for i in range(defects.shape[0]):
    a, b, c, d = defects[i, 0]
    start = tuple(on[a][0])
    end = tuple(on[b][0])
    far = tuple(on[c][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)
    Ang = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57
    if Ang <= 90:
```

```
        countdefects = countdefects + 1
if countdefects == 1:
```

Step five:

After declaring the angle of the thumb based on the thumb position, now the condition is applied for the zoom in and zoom out based upon the thumb angle. The zoom in and out of the video frame is based upon the following condition. So, whenever the camera detects the thumbs up, the video frame is zoomed in and when the thumb is down, the video frame is zoomed out respectively.

```
if countdefects == 1:
    cv2.putText(img, "Thumb Down Zoom out", (120, 310),
cv2.FONT_HERSHEY_PLAIN, 2,412)
    pts1 = np.float32([[0,0],[300,0],[0,300],[300,300]])
    pts2 = np.float32([[0,0],[300,0],[0,300],[300,300]])
    M = cv2.getPerspectiveTransform(pts1,pts2)
    dst = cv2.warpPerspective(img,M,(500,500))
else:
    cv2.putText(img, "Thumb Up Zoom in", (120, 310),
cv2.FONT_HERSHEY_PLAIN, 2, 412)
    pts1 = np.float32([[120,120],[380,120],[120,380],[380,380]])
    pts2 = np.float32([[0,0],[400,0],[0,400],[400,400]])
    M = cv2.getPerspectiveTransform(pts1,pts2)
    dst = cv2.warpPerspective(img,M,(600,600))
cv2.imshow('Thumb Recognition', dst)
all_img = np.hstack((drawing, anonymous_img))
if cv2.waitKey(1) & 0xFF == ord('q'):
```

In addition, it is significant to mention that just fingers with wrists must be shown in the rectangle with the white background. In case the wrist shown to the rectangle then pollex (the thumb) should be Up and Down in order to see the functionality of our application.

Platform used in this project.

For this project, we have used “python” as a programming language, “OpenCV” libraries and as for the Integrated Development Environment (IDE) the “PyCharm”.

The brief description of the platforms used in this project are as follows:

Python:

Python is an object-oriented, high-level programming language with dynamic semantics. It's high-level built in data structures combined with dynamic typing and binding, makes it very attractive for Rapid Application Development (RAD) and also for scripting to connect to the other components (P. S. Foundation, 2001). It supports modules, packages and encourages for code reuse. The python interpreter and the extensive library are available in source or binary form without charge for all the major platforms (P. S. Foundation, 2001). It started as the scripting language but now used both in desktop and web application. Just like other programming languages, it can be run in different IDEs, some popular and common are PyCharm, PyDev, Komodo IDE, Eric, Eclipse, etc.

OpenCV

Open Source Computer Vision (OpenCV) library is an open source computer vision and machine learning software library which was built to provide a common infrastructure for computer vision application (Team, 2018). It contains more than 2000 algorithms which can be used for image processing, detect objects, faces, extract 3D models and many other computer vision applications. It has Java, MATLAB, C++, interfaces and is platform independent which means run on Windows, Linux, Android and Mac OS (Team, 2018).

PyCharm

PyCharm is one of the popular IDE for python programming and for this project we have chosen to be programming in this IDE. PyCharm is the IDE developed by JetBrains. It is the company that developed tools and solutions for programmers and software companies. Very popular IDE, Android Studio solely for android application

development was developed by JetBrains with collaboration with Google. PyCharm contains Intelligent Coding Assistance, Built-in Developers Tools, Web Development Package and Scientific tools. It is a cross-platform IDE so can be installed and run in Windows, MacOS, Android and Linux environment.

User testing

The code is executed by the IDE without any errors and to check for the performance of the system, we have used the confusion matrix for the user testing. The confusion matrix helps to describe the performance of the model we have created. From confusion matrix, we can determine accuracy, precision, sensitivity, and specificity of the system. To determine these there are four factors that needed to be considered which are, True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).

For testing our system, we have recruited 5 participants to test the functionality of the system and each participant will test the system for 10 times.

Tasks: Thumbs Up for Zoom in and Thumbs down for Zoom out

Trials: 10 each

Tester: 5 participants

Variables:

True Positives (TP): Thumbs up and down detected and Zoom in and out respectively

True Negatives (TN): No Thumbs Up and Down detected and No Zoom in and out

False Positives (FP): Thumbs Up and Down detected but no Zoom in and Out occurred

False Negatives (FN): No Thumbs Up and Down detected but Zoom in and out occurred

To find:

a. Accuracy = (TP+TN)/ (TP+TN+FP+FN)
b. Precision: Positive Predictive Value (PPV) = (TP)/(TP+FP)
c. Sensitivity: True Positive Rate (TPR) = (TP)/(TP+FN)
d. Specificity: True Negative Rate (TNR) = (TN)/(TN+FP)

Guidelines for user testing

1. The background should be white with no objects behind.
2. Show just the wrist with closed palm inside the rectangular green frame.
3. Start with thumbs up (pollex) and down.

Observation and Finding of User testing

Confusion Matrix:

		Actual Value	
n = 50		Positive	Negative
Predicted Value	Positive	35 (TP)	0 (FP)
	Negative	15 (FN)	0 (TN)

Table 1: Confusion matrix for project

Hence,

a. Accuracy = (TP+TN)/ (TP+TN+FP+FN) = (35+0)/ (35+0+0+15) = 0.70 i.e. 70%
b. Precision: PPV = (TP)/ (TP+FP) = (35)/ (35+0) = 1 i.e. 100%
c. Sensitivity: TPR = (TP)/ (TP+FN) = (35)/ (35+15) = .70 i.e. 70%
d. Specificity: TNR = (TN)/ (TN+FP) = (0)/ (0+15) = 0.00 i.e. 0%

Conclusion and Future work

After the observation of user testing phase of the project, we could measure the performance of the system we created for recognizing thumbs up and down and zoom in and out of the video frame. The accuracy of the model is found to be 70%, precision 100%, sensitivity 70% and specificity 0%. We implemented the fundamentals of image processing and hand gesture recognition method in this project and able to create a gesture input system, and we are successful towards achieving our primary goal. With this project, we are able to gain intensive knowledge on image processing, gesture recognition and gestural input interaction system. Also, with this project we learn about Python programming language and working with OpenCV libraries for image processing.

In this project the functionality of gesture input is to zoom in and out of the real-time video frame. The further future work would be to zoom in and out of the image frame with the gestural input which is thumb up and down. This project is completed using the angle detection of the thumb, which means when the thumb angle is 90 degrees then the thumb is detected as Up and vice versa. In future, we would apply template matching to achieve this goal.

References

Amiraj Dhawan, V. H. (2013). Implementation of Hand Detection based Techniques for

Human Computer Interaction.

Auckland, T. U. o. (2010). Gaussian Filtering. Retrieved from https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf

Carroll, J. M. Human Computer Interaction. Retrieved from https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro

Foundation, I. D. Human-Computer Interaction (HCI) Retrieved from https://www.interaction-design.org/literature/topics/human-computer-interaction

Foundation, P. S. (2001). What is Python? Executive Summary. Retrieved from https://www.python.org/doc/essays/blurb/

Imran Hussain, A. K. T., Kandarpa Kumar Sarma. (2014). Hand Gesture Recognition System with Real-Time

Palm Tracking.

Jones, M. Retrieved from https://www.cs.bham.ac.uk/~rxb/Teaching/HCI%20II/intro.html

Shaik, K. B., Ganesan, P., Kalist, V., Sathish, B. S., & Jenitha, J. M. M. (2015). Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space. *Procedia Computer Science, 57*, 41-48. doi:https://doi.org/10.1016/j.procs.2015.07.362

Team, O. (2018). About. Retrieved from https://opencv.org/about.html

Tomayess Issa, P. I. (2015). *Sustainable Design, HCI, Usability and Environmental Concerns*: Springer-Verlag London.

Vladimir Vezhnevets, V. S., Alla Andreeva. (2003). A Survey on Pixel-Based Skin Color Detection Techniques.