International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

AN OPTIMIZED APPROACH FOR DEAF AND DUMB PEOPLE USING AIR WRITING

[1]Nithya.S, [2]Chandru.S, [3]Krishnakanth.G, [4]Pavithran.P

[1]Assistant Professor (Sr.G), Department of ECE, KPR Institute of Engineering and Technology, Coimbatore, Tamilnadu, India
[2,3,4] Students, Department of ECE, KPR Institute of Engineering and Technology, Coimbatore, Tamilnadu, India

---

Abstract - *The word communication is defined as the impacting or exchanging of information by speaking, writing or using some other medium. This paper presents an effective way of communication for deaf/dumb people. However, motion gestures themselves are not expressive enough. This paper provides effective way of communication using air writing with keyword recognition and it enables user to convey many messages with little effort. This is the method we find the motion of the particular object we going draw or write the exact mean of that particular motion. In existing system of this concept they use analyze the motion and draw approximate output in screen. But our method we draw the exact motion and we also used this method for the keyword detection and the output in the format of audio as well as video.*

Key Words*: *character detection, OCR, comparison method, keyword detection

1. INTRODUCTION

Air-writing refers to writing of linguistic characters or words in a free space by hand or finger movements. Air-writing differs from conventional handwriting; the latter contains the pen-up-pen-down motion, while the former lacks such a delimited sequence of writing events. Motion gestures provide a complimentary modality for general human–computer interaction. Motion gestures are meant to be simple so that a user can easily memorize and perform them. However, motion gestures themselves are not expressive enough to input text for motion-based control. We define "air-writing" as writing letters or words with hand or finger movements in a free space. Air-writing is especially useful for user interfaces that do not allow the user to type on a keyboard or write on a track pad/touch screen, or for text input for smart system control, among many applications. Isolated letters written in the air involve a sequence of hand or finger movements. Although any snapshot of such movements can be considered a realization of a motion gesture, air-writing is more complicated than gesture recognition because of the interdependence among the involved "gestures." In conventional handwriting, a sequential discrete stroke structure is made. A stroke is an isolated writing trajectory between the pen-up/pen-down events. In contrast, air-writing is rendered on a virtual plane without visual feedback and lacks the delimited sequence of writing events. Air-writing is also more complex for automatic recognition than cursive style writing on paper due to the lack of a concrete anchoring or reference position; the person who performs air-writing can only use an imaginary co-ordinate to guide the writing motion.

2. EXISTING SYSTEM

2.1 Hand gesture recognition

- Images of the hand gestures are taken using a Nokia N900 cell phone and matched with the images in the database and the best match is returned. Gesture recognition is one of the essential techniques to build user-friendly interfaces.

Figure-1: Hand Gestures

- The hand gesture recognition enables the users to communicate with limited predefined gestures
- And also Knob based voice outputs helps the user to select the information to be conveyed it is also limited and insufficient.

3. PROPOSED SYSTEM

3.1 .BLOCK DIAGRAM

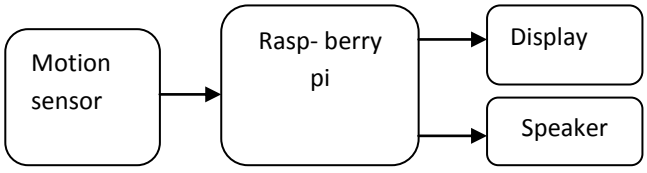

Figure-2: Block Diagram

3.2 CAPTURING IMAGE FRAMES

A video is basically a collection of continuous images displayed at a certain rate (generally 30 frames per second). To extract the frames from video first we need to attach this video to the input stream and then extract those as and when required. To attach the video to input stream we use the following function

CvCapture*capture=cvCreateFileCapture("file_name.avi");

3.3 COLOUR DETECTION

i)Open CV

The application written for this thesis relies heavily on computer vision, image processing and pixel manipulation, for which there exists an open source library named OpenCV (Open Source Computer Vision Library), consisting of more than 2500 optimized algorithms. Uses range from facial recognition, object identifying, classifications of human actions in videos, achieved with filters, edge mapping, image transformations, detailed feature analysis and more. Having Linux support, this is the perfect choice for developing an application specifically for a Raspberry Pi based system. Another positive aspect of this library is that it's written natively in C++ and therefore can be very smoothly implemented in a C/C++ application.

While there are numerous methods and algorithms contained within OpenCV, the most important benefit of this library for the purposes of this thesis are its basic data structures like *Mat*, which can be used to store pixel values of an image in an *n*-dimensional array, *Scalar* and *Point*, which respectively contain pixel values and coordinates of up to 3 dimensions .

The functions provided by this library are also necessary in the development process of the object tracking application. There are numerous options, but following the scope of this thesis, the focus is set on grabbing frames from a live camera feed [8], image threshold using HSV color space ranges, finding blobs and using their detected contours in a binary image and, in case a graphical user interface is enabled, displaying of image frames and a control panel for changing parameters during run-time.

ii) HSV VALUE

The HSV color space consists of three different descriptors, which is always the minimum number to classify a color, first of which is *hue* that describes a color that the human eye can see. The second is *saturation,* which describes the purity of a color or, in computer vision, how much that respective color is mixed with white and third is *value* (or luminosity, brightness), which on the contrary represents the magnitude of black in said color. Therefore if a pixel's saturation is high, it looks more rich, while a low saturation looks dull .A color with a low *value* appear darker or simply black.

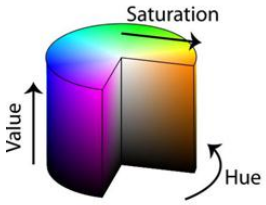

Figure-3: HSV color cylinder

The reason for using the HSV pixel format here instead of RGB , is because the HSV color space separates color information (*saturation*) from light intensity (*value*). When ignoring *value* in the threshold parameters, it is still possible to detect any color from an image, given the right *hue* and *saturation* values. Another important characteristic is the circular progression of *hue* values for different colors, which means that the highest *hue* value is simultaneously the lowest, because the space is circular not linear, and should be kept in mind when using image threshold algorithms in an OpenCV based application.

iii) PREPARING IMAGE FOR CHARACTER DETECTION

The first step towards making the image easily readable is converting the image into the HSV color space, which is much easier to threshold. The second step is threshold, which in this case is simply iterating through a pixel array of the captured image containing HSV values and setting the values of those pixels to 0 if the their values are below a lower boundary or 255 when the values are above a provided higher boundary, effectively creating a binary image. The boundaries are specified by *scalars* containing the currently set boundaries of hue, saturation and value.

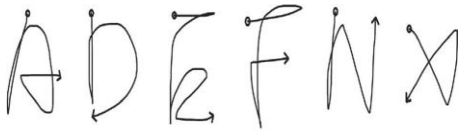

Figure-4: letters (a) A. (b) D.(c) E. (d) F. (e) N. (f) X.

iv) PROGRAM

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

while(1):

    # Take each frame
    _, frame = cap.read()

    # Convert BGR to HSV
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # define range of blue color in HSV
    lower_blue = np.array([110,50,50])
    upper_blue = np.array([130,255,255])

    # Threshold the HSV image to get only blue colors
    mask = cv2.inRange(hsv, lower_blue, upper_blue)

    # Bitwise-AND mask and original image
    res = cv2.bitwise_and(frame,frame, mask= mask)

    cv2.imshow('frame',frame)
    cv2.imshow('mask',mask)
    cv2.imshow('res',res)
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break

cv2.destroyAllWindows()
```

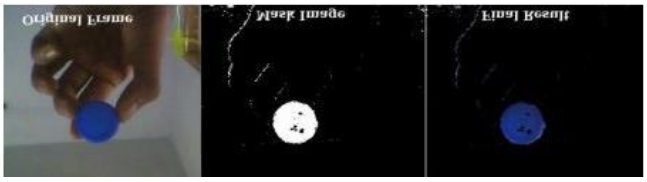

Figure-5: Color Detection

3.4 USING APPLICATION ON RASPBERRY PI

The application can be run on a Linux operating system, in this case a Raspberry Pi Linux distribution. It is preferably run from command line and can be provided parameters by the user. It is recommended the user try using the application without any parameters first to get a glimpse of how these settings change what the program is doing, especially using the color range sliders. The first step for the user is to check whether or not the camera is facing whatever direction should be scanned for objects. Next, the color ranges must be set correctly to separate a certain color (range) from the rest of the image. This is done via threshold and the resulting binary image can be displayed when enabled by the user on the control panel.

3.5 KEYWORD DETECTION

i) OCR method

This method of algorithms is an efficient method for detecting the characters from the image by using optical character recognition. Optical character recognition (also optical character reader, OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast). It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text to speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision. After that the keyword should be displayed based on the users wish.

ii) Comparison method

After detecting character image, it should be compared with database images then the characters should be recognized. After that the keywords should be displayed based on the users wish. For example character "A" displayed the word "welcome you all"

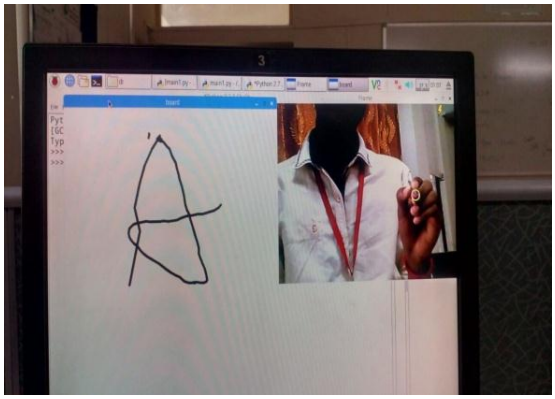

Figure-6: Character detection

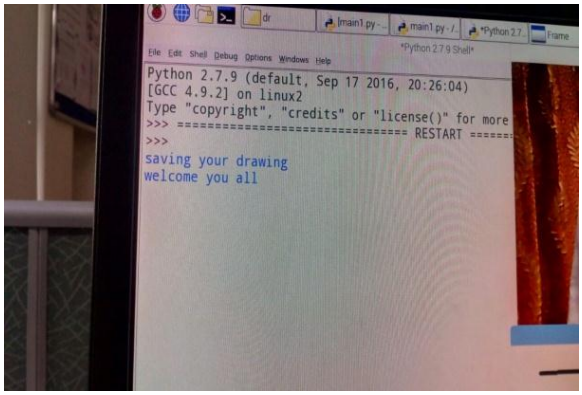

Figure-7: Keyword Detection

4. CONCLUSION

- This project overcomes the limited number of messages by using keywords in which recognizing two characters gives 520 possibilities.
- Using webcam instead of KINECT sensor and LEAP motion sensor makes this cost efficient
- After getting our expected results in color detection and tracking focus is to be made on optical character recognition

5. FUTURE SCOPE

- Taking this paper more compact and wearable, so it can be used in real time communication.
- Creating an interface which enables the user to customize their own set of information.

6. REFERENCES

[1] Feature and Processing Of Recognition of Characters, Words & Connecting Motions - R Dharma lingam – 2017

[2] Air-Writing Recognition—Part I: Modeling and Recognition of Characters, Words, and Connecting - Motions M Chen, G AlRegib, BH Juang – 2016

[3] Air-writing recognition—Part II: Detection and recognition of writing activity in continuous stream of motion data- M Chen, G AlRegib, BH Juang -2016

[4] Air-writing characters modelling and recognition on modified CHMMS Xu, Y Xue – 2016

[5] Human computer interface using hand gesture recognition based on neural network- Hamid A. Jalab, Herman. K. Omer – 2015

[6] A new writing experience: Finger writing in the air using a kinect sensor - X Zhang, Z Ye, L Jin, Z Feng, S Xu – 2013

[7] Hand Gesture Recognition System for Deaf and Dumb People Using PCA -Manisha U. Kakde1, Amit M. Rawate2 – 2013

[8] Vision-based handwriting recognition for unrestricted text input in mid-air - A Schick, D Morlock, C Amma, T Schultz – 2012[9] Hand gesture recognition based on shape parameters - Meenakshi panwar- 2012