IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AIR SCRIPTING USING OPENCV

[1]Guluru Manoj Reddy, [2] R Murali Dhar, [3]Y Naveen Kumar, [4]E Sathiyanarayanan

[1]Student, [2]Student, [3]Student, [4]Assistant Professor
[1]Electronics and Communication Engineering,
[1]Madanapalle Institute of Technology and Science, Madanapalle, India

Abstract: In recent years, one of the most fascinating and challenging research areas in image processing and pattern recognition has been writing in the air. It provides a major contribution to the advancement of an automated process and can improve the human-machine interface in a variety of applications. Several studies have developed innovative methods and practices that would cut processing time while increasing recognition accuracy. Within the discipline of Computer Vision, object tracking is regarded as a critical problem. Deaf and partially disabled people can use Air Scripting to effectively communicate.

Index Terms - Image Processing, RGB and HSV Colour Model, OpenCV, NumPy

1. INTRODUCTION

The conventional art of writing is being overtaken by digital art in the digital age. Digital art refers to art forms that are expressed and transmitted in a digital format. The digital manifestation is distinguished by its reliance on modern science and technology. Traditional art refers to work that was created prior to the advent of digital art. Visual art, audio art, audio-visual art, and audio-visual imaginary art, which comprises literature, painting, sculpture, architecture, music, dance, theatre, and other works of art, can simply be split into visual art, audio art, audio-visual art, and audio-visual imaginary art. Traditional and digital art are inextricably linked and interdependent. Although social growth is not a result of the choice of the people, the demands of human life are the primary driving force. In art, the same thing happens. In the current situation, digital art and traditional art are both part of a symbiotic state, therefore we must thoroughly comprehend the fundamental understanding of the form between the two. Pen and paper, chalk, and a whiteboard are examples of conventional writing methods. The primary goal is to write digitally, a hand gesture recognition technology is used. Many different methods of writing are used in digital art, such as utilizing a keyboard, a touchscreen surface, a digital pen, a stylus, or electronic hand gloves. Etc. However, with our method, we recognize hand gestures.

Python is used to implement a machine learning method. Programming, which allows man and machine to connect in a natural way as well as the machine with the progress of technology, there is a greater need for Natural 'human–computer interaction (HCI)' development the use of [1] systems to replace traditional systems is on the rise. Online teaching learning platforms have become an integral part of life, during and post Covid-19 pandemic. In this regard, many teaching learning accessories have been developed for use. The work presented in this project is an application that helps draw one's imagination on screen by just capturing the motion of object of interest with a camera in air. OpenCV was used to construct a computer vision project.

2. LITERATURE REVIEW

2.1 ROBUST HAND RECOGNITION WITH KINECT SENSOR

In [2], the system proposed used the depth and color information from the Kinect sensor to detect the hand shape. As for gesture recognition, even with the Kinect sensor. It is still a very challenging problem. The resolution of this Kinect sensor is only 640×480. It works well to track a large object, e.g., the human body. But following a tiny thing like a finger is complex.

2.2 LED FITTED FINGER MOVEMENTS

Authors in [3] suggested a method in which an LED is mounted on the user's finger, and the web camera is used to track the finger. The character drawn is compared with that present in the database. It returns the alphabet that matches the pattern drawn. It requires a red colored LED pointed light source is attached to the finger. Also, it is assumed that there is no red-colored object other than the LED light within the web camera's focus.

2.3 RELATED WORK

A lot of work has been done on extending and simplifying the process of creating documents. Examples include using gesture-based input control [4]– [7], swipe-based input methods [8], voice-based input methods [9] and even interaction methods on imaginary surfaces [10]. Methods for establishing a Virtual Reality environment where the user explicitly inputs information are particularly pertinent to our work. These solutions are ineffective in contexts where the user must interact with real-world things due to their Virtual Reality nature. Such as a smart factory or a smart class. Our approach differs from these methods in that the user remains in the real world.

2.4 RELATED WORK

A lot of work has been done on extending and simplifying the process of creating documents. Examples include using gesture-based input control [4]– [7], swipe-based input methods [8], voice-based input methods [9] and even interaction methods on imaginary surfaces [10]. Methods for establishing a Virtual Reality environment where the user explicitly inputs information are particularly pertinent to our work. These solutions are ineffective in contexts where the user must interact with real-world things due to their Virtual Reality nature. Such as a smart factory or a smart class. Our approach differs from these methods in that the user remains in the real world.

3.SYSTEM MODEL

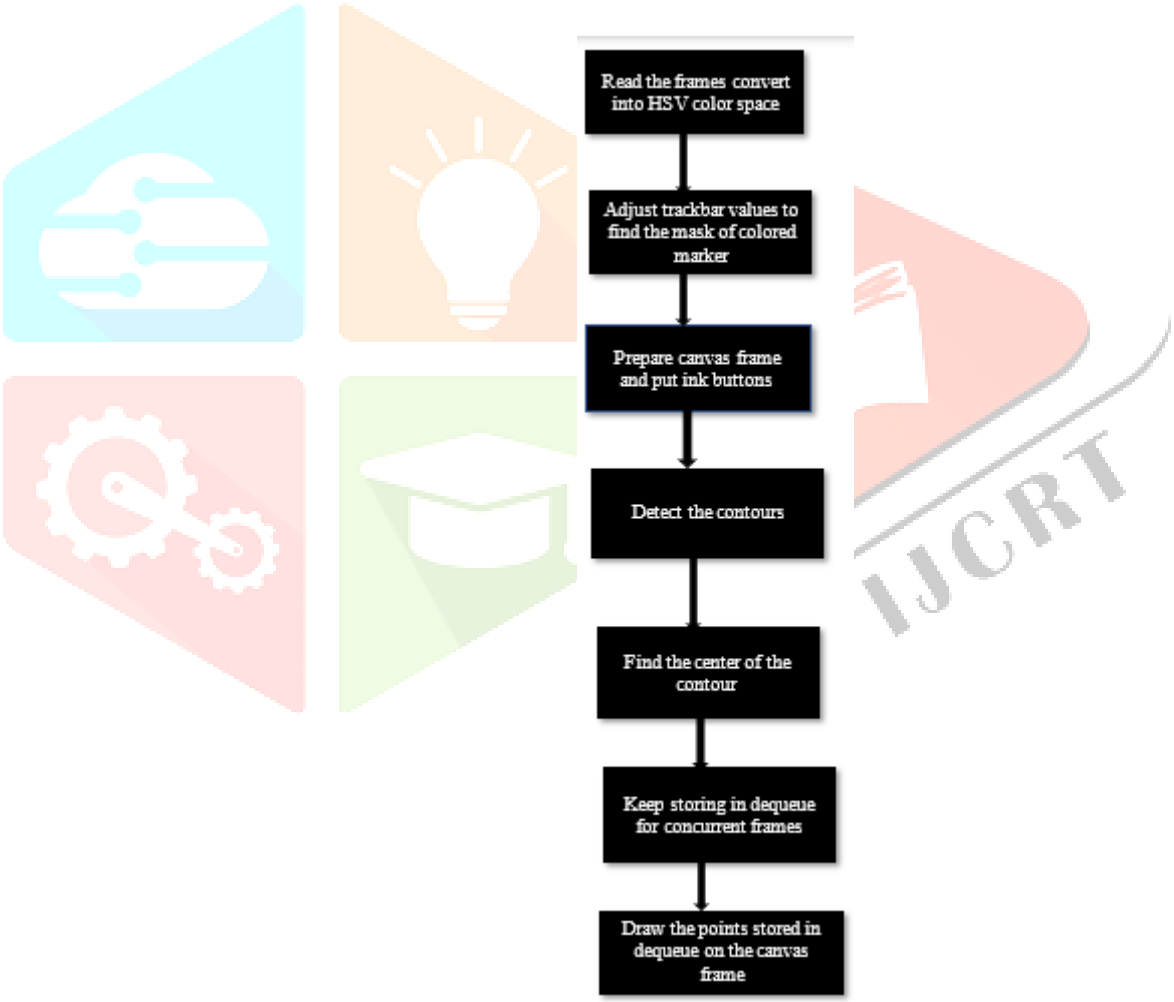

Figure 1.1 Air Scripting System Flow Chart

The whole operating flow system of this study is depicted in Figure 1.1. Reading of the frames from webcam converts into HSV color model then preparing canvas frame followed by keeping respective ink buttons on the canvas frame.

4. METHOD OF IMPLIMENTATION

Step 1 – Start reading the frames.
Step 2 -- Convert the collected frames to HSV colour format (easily detectable colours).
Step 3--Create a canvas frame and attach the appropriate ink buttons to it.
Step 4 – Now, adjust the track bar values to locate the coloured marker's mask.
Step 5 – The mask is pre-processed using morphological methods (Eroding & Dilation).
Step 6 – The following steps are as follows: Detecting the contours. determining the centre coordinates of a large contour and storing them in an array for future frames (Arrays for drawing points on the canvass).
Step 7 – Determine the centre coordinates of a large contour and keep them in the dequeue for subsequent frames (Arrays for drawing points on the canvass).
Step 8 – Draw the points into dequeue.
Step 9 –Save the paint window in the system.

5. SOFTWARE REQUIREMENT SPECIFICATION

a) PYTHON:

- Python3.8

b) LIBRARIES:

- Numpy
- OpenCV
- Collections

c) OPERATING SYSTEM:

- Windows

6. RESULT AND DISCUSSION

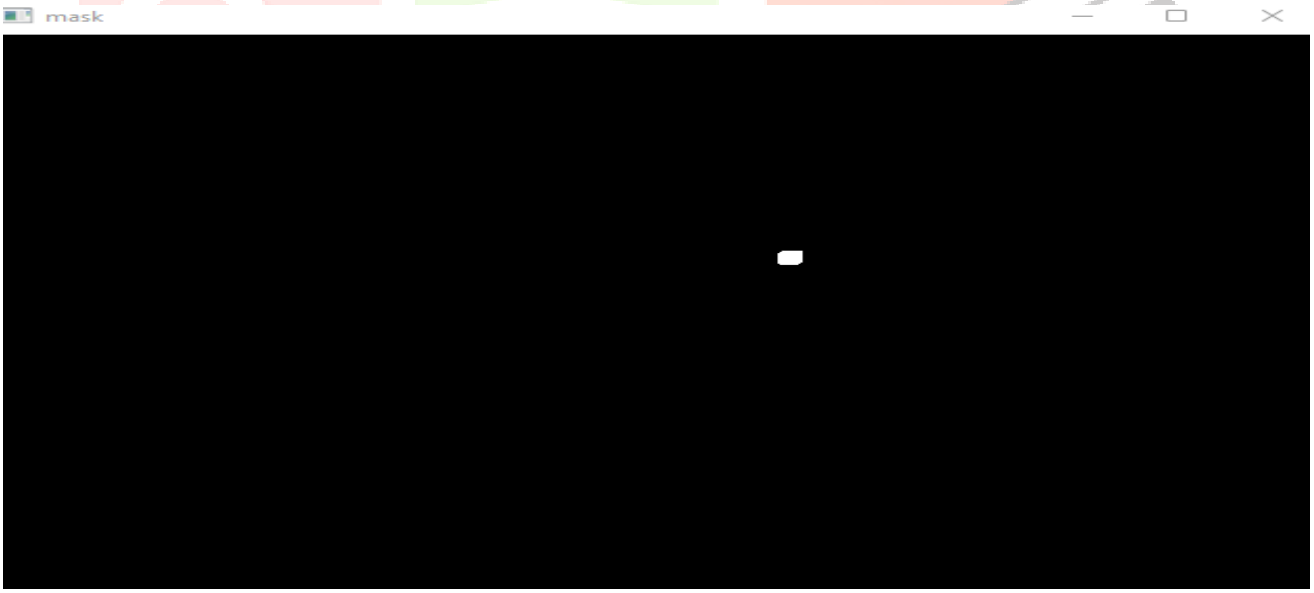

Figure 1.2 Mask Window

The figure 1.2 shows the mask window that is detected by web camera in that window the thick white rectangle shows the mask. Mask is a binary image which is combination of zero's and one's. In our case blue mask is used which acts as a colored marker.

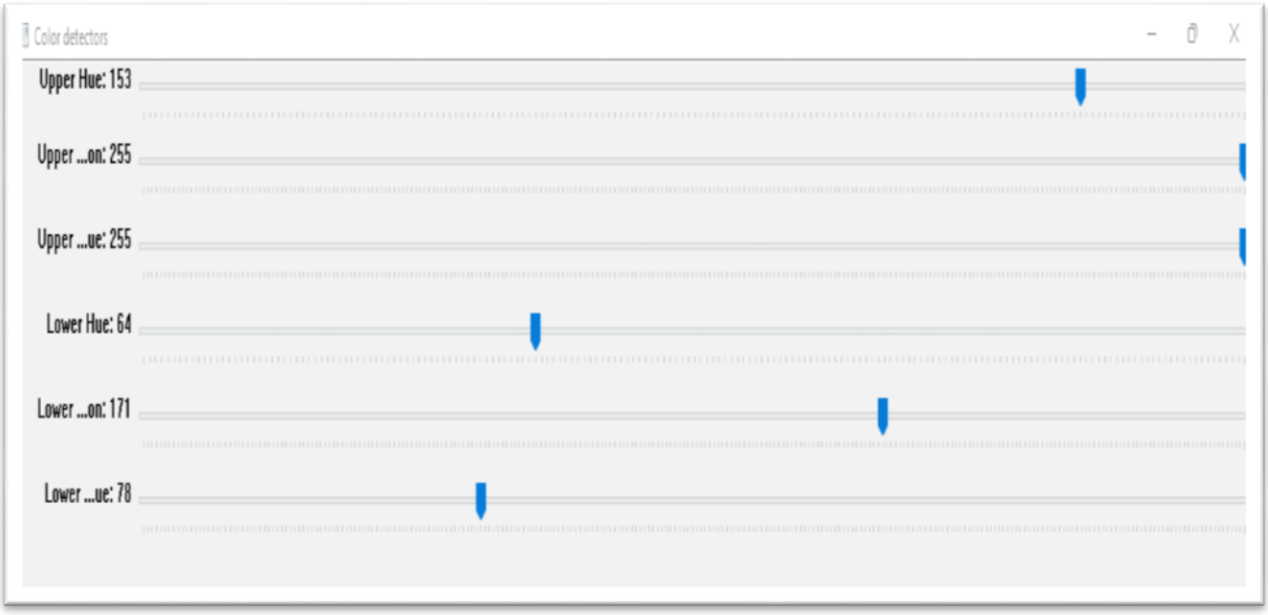

Figure 1.3 Colour Detector Window

The figure1.3 shows colour detector window which has HSV values. It has 6 different parameters to detect colour mask the blue colour mask values are listed in the table below.

1) Lower Hue
2) Upper Hue
3) Lower Saturation
4) Upper Saturation
5) Lower Value
6) Upper Value

TABLE 1.4 Values Set for Blue Colour Mask.

S.NO	PARAMETER NAME	VALUES
1	Upper Hue	(153/160)
2	Upper Saturation	(255/255)
3	Upper Value	(255/255)
4	Lower Hue	(064/180)
5	Lower Saturation	(171/255)
6	Lower Value	(078/255)

The table 1.4 shows the various values set for blue colour mask.

cv2.namedWindow("Color detectors") This function creates a window with name “Color Detectors”

cv2.createTrackbar("Upper Hue", "Color detectors", 153, 180, setValues)

This function creates the trackbar needed for adjusting the marker colour. We need to use six times in our code to generate a blue colour marker.
Loading the default webcam of PC using below mentioned function.
cap = cv2.VideoCapture(0)

Figure 1.5 Paint Window

The figure1.5 shows the paint window which has 5 colour buttons namely blue, green, red, yellow and a save button and a clear button which can remove data from dequeue. And the colour buttons are meant for selecting respective colours on the paint window. These six buttons are created using function cv2.rectangle(paintWindow, (160,1), (255,65), colors[0], -1). which creates rectangle shaped box in paint window, the range is defined within parenthesis and colors[0] gives white background also the function used for keeping the name inside the rectangle is cv2.putText(). All the six buttons are predefined with in its range. If our color marker goes in that particular range, then OpenCV takes those functionalities. The output windows can be displayed on the screen using cv2.imshow() function.

- Clear: Clears the screen by emptying the dequeues
- Red: Changes the marker colour to red
- Green: Changes the marker colour to red
- Yellow: Changes the marker colour to red
- Blue: Changes the marker colour to red
- Save: Saves the current paint window

The figure 1.6 shows the paint window with text written on it “Air Scripting” using red colour. Different colours can also be selected for writing.

Figure 1.6 Paint Window with text

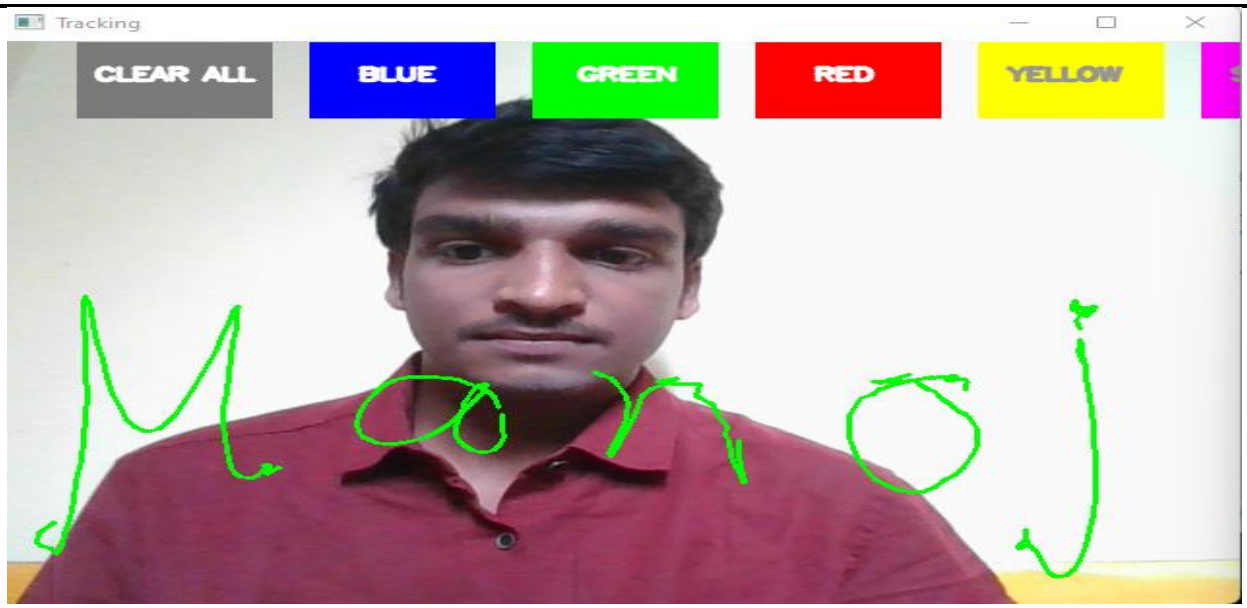

Figure 1.7 Tracking Window

The figure 1.7 shows tracking window with text written “Manoj” on it along with that person. The difference between canvass frame and tracking window is that person can be seen along with the text.

we performed the morphological operations are a set of operations that process images based on shapes. These apply a structuring element to an input image and generate an output image. The very basic morphological operations are two:

1. Erosion and
2. Dilation.

Erosion: -
- Eliminates away the boundaries of foreground object.
- Mainly used to diminish the features of an image.

Dilation: -
- Mostly increases the object area.
- Used to make the features get elevated

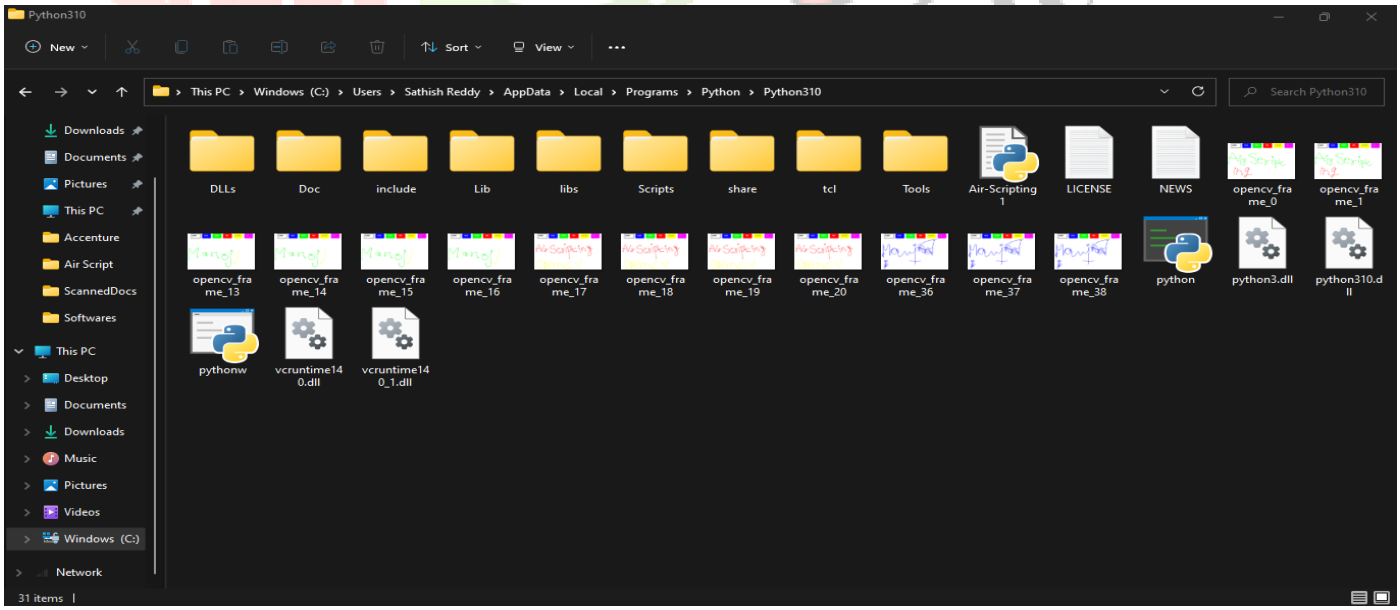

Figure 1.8 Window Saved

The figure 1.8 shows the location/path in which the paint window is saving, when we pressed the save button which is present on the window screen. The series of paint windows were saved in the path where python program is running, the series of frames are shown in the figure with name opencv_frame_2. The location where the window is saving is “C:\Users\Sathish Reddy\AppData\Local\Programs\Python\Python310”

7. FEATURES OF THE AIR CANVAS

1. Can track any specific-coloured pointer.
2. User can draw in four different colours and even change them without any Hussle.
3. Able to rub the board with a single location at the top of the screen.

8. TESTING AND VALIDATION

Table.1.9 Test Parameters

SL.NO	TEST CONDITIONS	SYSTEM BEHAVIOUR
1	High Intensity of Light	Not able to detect mask properly
2	Medium Light Intensity	Able to detect the mask
3	Poor Lightening Conditions	Sometimes detects, sometimes fails

The various testing conditions and system behaviour are included in table 1.9 The system's performance is assessed under a variety of different lighting. For better detection of the mask the lightening intensity should be medium otherwise system fails.
The System can be undergoing into various test cases in different conditions that can be carried out to check the system behaviour and to compare with the results with the expected output. For three test conditions the system behaviour and the expected outputs are matched with each other. So that the system efficiency is better than the existing system. And the Artificial Intelligence is growing exponentially, so the future prospective is very advanced.

9. CONCLUSION

In conclusion, Air Scripting is an interactive way to communicate with kids it reduces the use of laptops. It is an effective communication stands out for people who has no interest on laptops. This project provides the user with an interactive environment in which he can draw whatever he wants by selecting his desired colours from a palette of options. We've learned about colour detection, segmentation morphological techniques.

To enhance hand gesture tracking, we would have to delve more into OpenCV. Even in high intensity of light the system will work better. So that the system efficiency is better than the existing system. And the Artificial Intelligence is growing exponentially, so the future prospective is very advanced.

I. ACKNOWLEDGMENT

The authors gratefully acknowledge the Madanapalle Institute of Technology and Science for the resources provided as well as their unwavering encouragement for the project. Without their support, this project would not have been possible.

REFERENCES

[1] Ruiduo Yang, Sudeep Sarkar, "Coupled grouping and matching for sign and gesture recognition", Computer Vision and Image Understanding, Elsevier, 2008

[2] Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006

[3] Yuan-Hsiang Chang, Chen-Ming Chang, "Automatic Hand-Pose Trajectory Tracking System Using Video Sequences", INTECH, pp. 132- 152, Croatia, 2010

[4] S. Kumar and J. Segen, "Gesture-based input interface system with shadow detection," Sep. 23 2003, uS Patent 6,624,833.

[5] E. Tamaki, T. Miyaki, and J. Rekimoto, "Brainy hand: an ear-worn hand gesture interaction device," in CHI'09 Extended Abstracts on Human Factors in Computing Systems. ACM, 2009, pp. 4255–4260.

[6] P. Mistry, P. Maes, and L. Chang, "Wuw-wear ur world: a wearable gestural interface," in CHI'09 extended abstracts on Human factors in computing systems. ACM, 2009, pp. 4111–4116.

[7] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," IEEE transactions on visualization and computer graphics, vol. 18, no. 4, pp. 643–650, 2012.

[8] W. C. Westerman, H. Lamiraux, and M. E. Dreisbach, "Swipe gestures for touch screen keyboards," Nov. 15 2011, uS Patent 8,059,101.

[9] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates et al., "Deep speech: Scaling up end-to-end speech recognition," arXiv preprint arXiv:1412.5567, 2014.

[10] S. Gustafson, D. Bierwirth, and P. Baudisch, "Imaginary interfaces: spatial interaction with empty hands and without visual feedback," in Proceedings of the 23nd annual ACM symposium on User interface software and technology. ACM, 2010, pp. 3–12.