International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 443-451 www.ijaem.net ISSN: 2395-5252

Creating Air Canvas Using Computer Vision

Adinarayana Salina[1], K. Kaivalya[2] K. Sriharsha[3] K. Praveen[4] M. Nirosha[5]

Department of CSE, Raghu Institute of Technology, Visakhapatnam, Andhra Pradesh, India
Student [2,3,4,5] B. Tech (COMPUTER SCIENCE ENGINEERING)
Raghu Institute of Technology, Visakhapatnam, Andhra Pradesh, India

Submitted: 01-06-2022 Revised: 05-06-2022 Accepted: 08-06-2022

ABSTRACT: -
Writing in air has been one of the best interesting and challenging progress in field of image processing and pattern designing now a days. It mainly contributes to the advancement of an automated process and can improve the interface between man and machine in various applications. In Several research areas, works have been focusing on new techniques and methods which helps in reducing the processing time and providing high recognition efficiency, accuracy. Object tracking is considered an important task in the field of computer vision.it involves first, detecting the object, secondly tracking its movement from frame to frame, and finally analysing the behaviour of that object. We will use computer vision to track the path of the finger. It will be a powerful means of communication and is an effective method that reduces the use to write.

As We all know that artists create paintings on a canvas. Then think, what if we can paint on air just by waving our hands. So, in this project, we are going to design an air canvas using OpenCV and Python.

Keywords: computer vision, air writing, hand and real time gesture control system, object detection

I. INTRODUCTION

In the emerging of digital world, traditional art of writing is being replaced by digital art nowadays.Digital art refers to the expression and transmission of an art form with a digital form. Traditional art refers to the art form created before digital art. From the receiver to the analysis, it can simply be divided into the visual arts, the audio arts, the audio-visual arts, and the audio-visual fantasy arts. Digital art and traditional art are linked and interdependent.

In the current situation, digital art and traditional art include a state of symbiosis, so understanding the Traditional methods include pen and paper, chalk and the blackboard writing method. The essential goal of digital art is to build a hand gesture recognition system for digital writing. Digital art includes many ways of writing, such as using keyboards, touchscreen surfaces, digital pens, styluses, using electronic gloves, and more. But in this system, we use hand gesture recognition with the use of computer vision ing using python programming, creating a natural human-machine interaction. With advancements in technology, the need to develop natural human-computer interaction (HCI) systems to replace traditional systems is increasing rapidly.

Color detection is an image processing technique where we can detect any color within a certain range of the HSV colorspace.Image segmentation is one of the process of labelling each pixel in an image, where each pixel shares characteristics.

Here in this project we used64bit operating system, x64based processor and the process configuration is Intel(R) Core(TM) i57200U CPU @ 2.50GHz 2.70 GHz with 8GB RAM and the software libraries used are Python,Numpy,Opencv

II. LITERATURE SURVEY

A. Powerful hand recognition with Kinect sensor In[3], the proposed system uses depth and color information from the Kinect sensor to detect the shape of the hand. For gesture recognition, with the Kinect sensor is a difficult problem to be noted. The resolution of this kind of Kinect sensor is only 640×480. It works well for tracking a large object, for example, the human body. But something as small as a finger is complicated.

B. Motion of an LED-equipped finger the authors in [4] have proposed a method in which an

LED is mounted on the user's finger and a web camera is used to track the finger. The plotted character is compared with the character present in the database. It returns all alphabets that matches the drawn pattern. It requires a pointed red LED light source attached to the finger. Also, assume that there are no red objects other than LED light in the focus of the web camera.

C. Enhanced Desktop Interface In [5], an enhanced segmented desktop interface approach for interaction was proposed. This system uses a projector and a charged device (CCD) camera for you to use your fingertips; users can use desktop applications. In this system, each part performs each uniquely different task. The left hand is used to select radial menus, while the right hand is used to select objects for manipulation. He achieves this by using an infrared camera. Determining the fingertip is computationally expensive, so this system defines search windows for the fingertip.

2.1 CHALLENGES IDENTIFIED

A. Fingertip detection is the system which only works with your fingers, and there are no such devices like highlighters, or any other related gadgets. Identifying and recognition an object like a finger from an RGB image without an advanced device like depth sensor is a great challenge.

B. Lack of pen up and down movement of the system uses RGB camera to write from starting. Since depth sensing is not possible, the pen's up and down movement cannot be tracked. Thus, the entire trajectory of the fingertip is tracked and the resulting image will be meaningless and not recognized by the model The difference identified among hand written and air written ‘G’ is shown in Figure

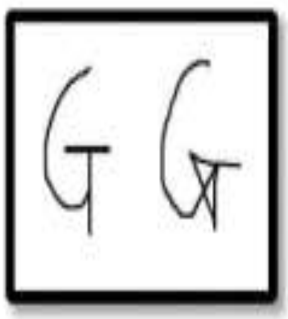


Figure 2.1 Canvas Based Drawing

C. Real-time system control Using real-time hand gestures to transition a system from one state to another requires a lot of code attention. In addition, the user must know many movements to master his plan fully.

2.2 PROBLEM STATEMENT

You never thought, raising your finger in the air can help us draw on a real picture. How wonderful that this aerial web works in Computer Vision Projects.

2.3 PROBLEM SOLUTION

Computer projects helps us to draw on a screen easily by waving our fingers with a color indicated, Open CV helps in more advancement many ways of writing, such as using keyboards, touchscreen surfaces, digital pens, styluses, using electronic gloves, and more. But in this system, we use hand gesture recognition with the use of machine learning algorithm using python programming, creating a natural human-machine interaction. With advancements in technology, the need to develop natural human-computer interaction (HCI) systems to replace traditional systems is increasing rapidly.

Our project mainly focuses on solving major problems –
1. Deaf people: Although we take listening and hearing as an easy task, communication using sign language. Most of the world cannot understand their feelings without a translator in the middle.
2. Smartphone abuse: They cause accidents, depression, distraction and other illnesses that we humans can still detect. While its portability and ease of use are deeply admired, its downside is life-threatening events.
3. Waste of paper is not uncommon. We waste a lot of paper when doodling, writing, drawing, etc.
4.This project is developed using openCV, Python

environment.

III. FEATURES

1.Can track any specific individual required color pointer.
2.Users can draw four different colors and even change them whenever required without any difficulties.
3.There is Clear option at the top of the display that helps to scrub the board at a time.
4.Once the program has started No need to involve the contact with the computer

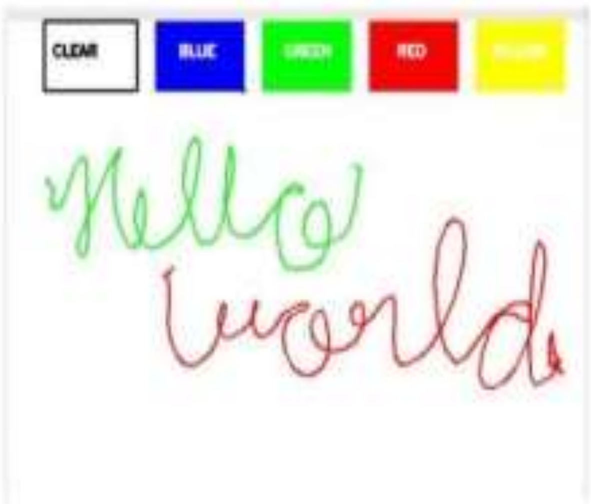


Figure 3.1 Structure of Air Canvas

IV. SYSTEM METHODOLOGY

The system needs a data set for finger detection model. The main purpose of the fingertip model is to record movement, this recorded movement is air characters

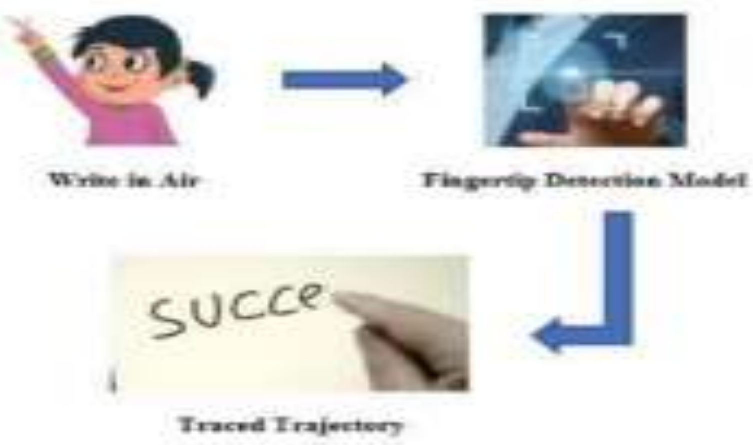


Figure 4.1 Methodology of Air Canvas

A. Fingertip sensor model: Air writing can be done simply by using a single-color stylus or air pen. However, the system uses fingertips. We believe that people can write in the air without having to wear a stylus. We used deep learning algorithms to detect the tip of the finger in each image, generating a list of coordinates.

Technique for creating finger recognition dataset:

Video to Frames: In this approach, two-second videos of a person's hand movements are captured in different regions. These videos are then divided into 30 individual images, as represented in diagram 3. 2000 images total.

This dataset was manually labelled using LabelImg. The best model on this dataset yields 99% accuracy. However, since all 30 frames are produced from the same video and region, the dataset is very monotonous. Therefore, the model does not perform well for discrete backgrounds from the backgrounds in the data set.

B. Take photos in separate backgrounds: To overcome the inconvenience caused by the previous method's lack of diversity, we introduced a new dataset. This time we are aware that we need a few gestures to make control the system. So, we collected four different hand poses, shown in Figure 4. The main intension was to make the model able to effectively recognize the fingertips of the four fingers.

This makes the system under the control of user by the number of fingers as he represents them.

They can now type quickly by showing their index finger, convert that writing motion into electronic text by showing two fingers, it also adds space by showing three fingers, press the backspace key

with how to display five fingers, make consecutive predictions by showing four fingers, then show 1, 2,3 fingers to choose 1st or 2nd or 3rd predictions as per requirement. To exit prediction mode, show five fingers. This dataset includes 1800 images approximately. by using this kind of script, the previous model is designed to automatically label this dataset. Next, we correct the mislabelled images and introduce another template. Accuracy reaches 94%. Unlike before, this model works well in different contexts.

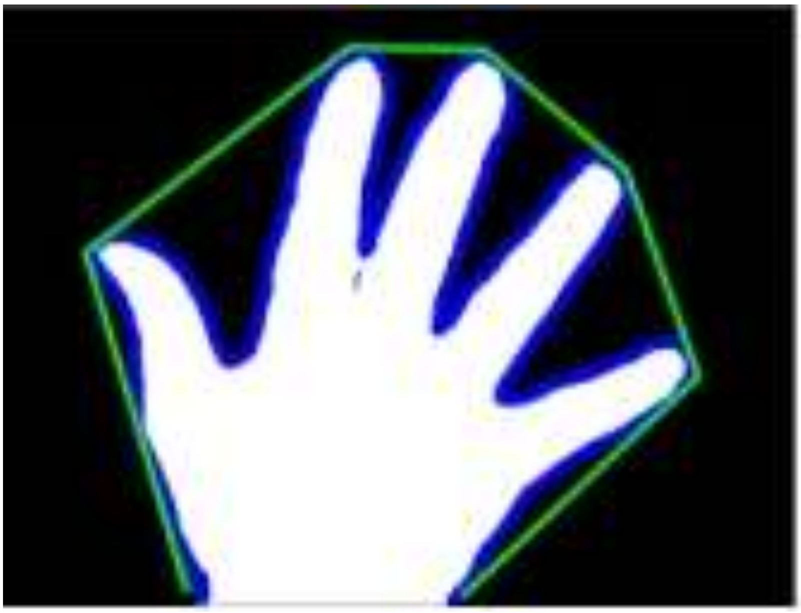


Figure 4.2 Detection of Finger Tips Using OpenCV

C. Training the finger recognition model: Once the dataset is ready and labelled, it is divided into training and development groups (85% to 15%). We used pre-trained faster Single Shot (SSD) and RCNN detector models to train our dataset. the Fast RCNN is best in accuracy than SSD. for information we must verify results. The SSD combines two standard detection modules - one that suggests regions and one that classifies them. This speeds the performance of objects that are detected in a single attempt. Usually It is used for real-time object detection. Faster RCNN uses the feature map output from Fast RCNN to compute area recommendations. They are evaluated by a network of regional recommendations and delivered to an area of the interest group class. The final result is given as two fully connected classes for bounding box classification and also regression. We adapted the final fully connected Fast RCNN layer for fingertip recognition in images.

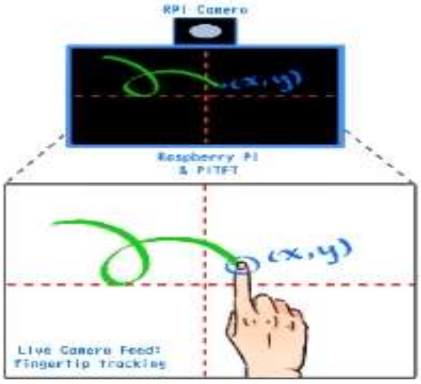


Figure 4.3 Training the finger recognition model

V. WORKFLOW ALGORITHM

This is the most interesting part of our system. features are introduced due to writing. Therefore, the number of gestures used to control the system is equal to the number

of actions involved. The basic features we have included in our system are

1. Writing Mode In this state the system will draw the fingertip coordinates and store them
2. Color Mode - User can change text color between different colors available.
3. Backspace mode - if any user does a mistake, we need to overcome the mistake for such situation we require a backspace. To add a backspace a gesture is introduced.

Description Points to develop air canvas using OpenCV:

- Import and install necessary packages and files.
- Capture frames from the webcam
- Invoke a canvas window for Drawing
- Detect and track the green color
- Draw on the canvas window.

VI. SYSTEM DESIGN

Have you ever wanted to capture your imagination just by lifting your finger in the air Here we will create an aerial canvas that can draw anything on it by capturing the movement of the color marker with the webcam. A colored object which is on the tip of the finger is used as a marker. OpenCV helps computer vision techniques in this project. The best preferred language is python as it has many libraries and easy to use syntax, but understanding the basics can be implemented in any language which are supported by OpenCV.

Here, color detection and track ing are used to achieve our goal. Color markeris detected and a mask is generated. It includes the next stagesof morphological activity on the product mask, erosion and expansion.

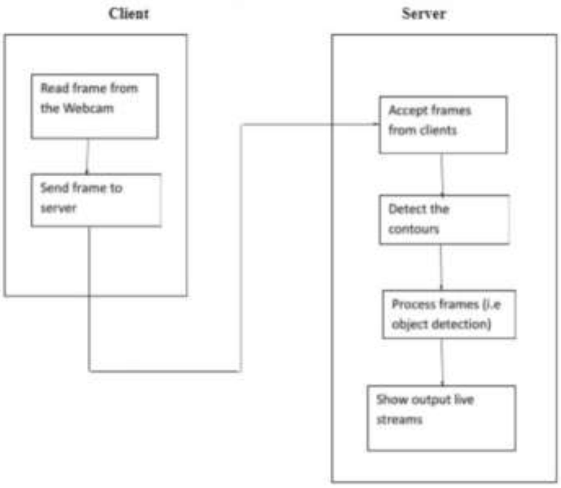

Client
Server
Read frame from the Webcam
Send frame to server
Accept frames from clients
Detect the contours
Process frames (i.e object detection)
Show output live streams

Figure 6.1 Block diagram of client and server

A.Color tracking of object at fingertips:

First, Captured image from the webcam must be converted to the HSV color space to detect colored objects.

The code below converts the incoming image to HSV space, which is great, and helps color space for color tracking. We will now create track bars to align the HSV values into the required color range of the colored object. that we have placed on our fingers. now, once track bars are configured, we get the real time value of the track bars and a range is created. This range is a numpy struct used to pass into the cv2.inrange() function fur further. This function returns the colored object and mask. This Mask is in a black and white image included with white pixels at the position of the required color.

B. Detecting the mask contour of the colored object:

Now after detecting the mask in the Air Canvas, Now it's time to identify its center to draw the line. Here, in the below code of ,we do some morphological manipulation on the mask, to make eliminate impurities and easy to detect contours

C. Drawing the Line by using the Contour position.

The real logic involved in this Computer Vision project; we will perform a python deque on of A data Structure. The deque will store the position of the contour on each frame and we will use these points to make a line using OpenCV functions. This position of the contour helps us to make decision, to click on a button or we want to draw on the sheet. We implemented some of the buttons on the top of Canvas, if the pointer points their area, it will help trigger their method. There are four buttons on the canvas created with OpenCV.

Clear the whole screen by deque.
Red: Changes the ink colour to red using the palette.
Green: Change the color to green using the palette.
Yellow: Change the color to yellow

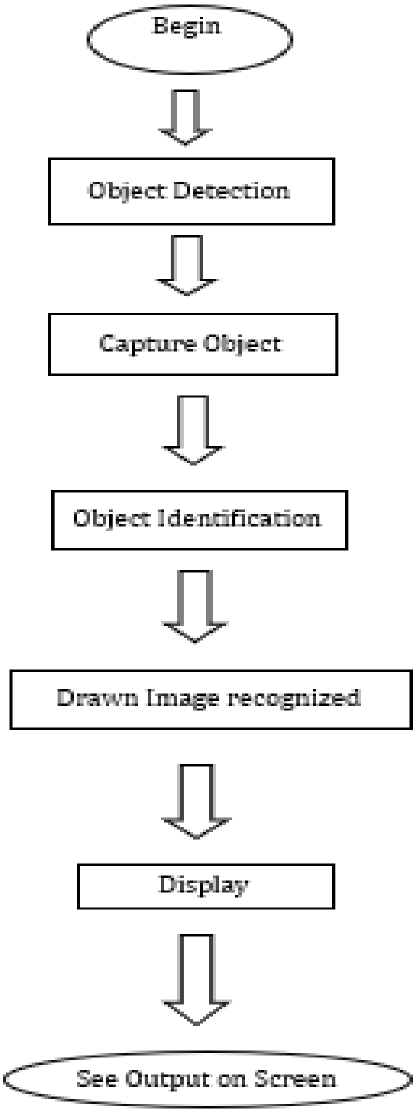
D. Points used:

We will now draw all the points on the locations stored in the deques, with the corresponding color.

6.1 PROJECT FLOW

1. Start image playback and convert captured image to HSV color space (Easy to spot color)
2. Make the canvas and place the corresponding colour buttons on it.
3. Adjust the trace bar values to find colour highlight mask.
4. Mask pre-treatment with morphological manipulations. (Erotion and Dilation)
5. Detect the contours, and identify the coordinates of the center of the largest contour and keep storing them in array
for the next frames. (Tables for drawing points on the canvas)
6. Finally, draw the points stored in the table.

6.2 IMPLEMENTATION OF FLOW CHART



VII. RESULT

Here are the results and output for our project. We used OpenCV module to run our code.
We have executed our code in anaconda ,Jupiter Notebook where we first installed required packages and run our code for output.

Here two windows are displayed one window is our camera page where our pen will be tracked using coordinates which we specified in our code and the second window is for drawing the recorded coordinates and we can use components like clear or change the colour

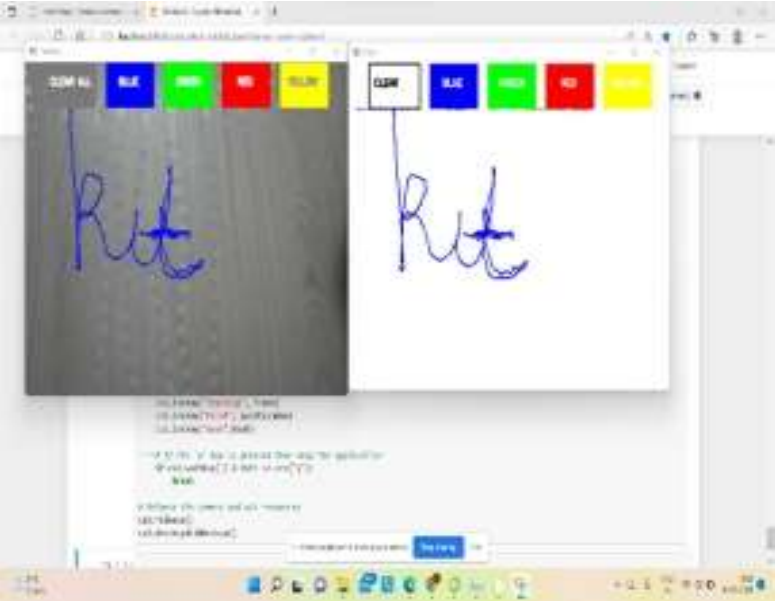


Figure 7 Diagram showing the result

VIII. CONCLUSION

The system is capable of challenging traditional writing methods. It eliminates the need to carry a cell phone in hand to take notes, providing an easy way to do the same on the go. It will also serve a great purpose of making it possible for people with special disabilities to communicate easily. Even the elderly or those who have difficulty using the keyboard can use the system with ease. By extending the functionality, the system can also be used for quick control of IoT devices. Drawing in the air can be done. The system will be a great piece of software for smart devices to use so that people can easily understand and interact with the digital world. Augmented reality can bring text to life. Some system limitations may be improved in the future. First, using handwriting recognition instead of character recognition will allow the user to write word-for-word, which will speed up writing. Second, hand gestures with pauses can be used to control the system in real time as is done in [1] instead of fingertips usage. Third, our system recognizes fingertips in its background state and changes their original state. The air writing system should only follow the owner's control gestures and must not be fooled by those nearby having a deep glance is much more important. Also using EMNIST dataset, which isn't a appropriate dataset of air character. In future the further object detection algorithms like YOLO v3 will improve accuracy and speed of fingertip recognition. In the future, advances in artificial intelligence will enhance the efficiency of aerial writing.

REFERENCES

[1]. Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016.

[2]. P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, pp. 1-6, 2016.

[3]. Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4, 2013 [4] Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006

[4]. H.M. Cooper, "Sign Language Recognition: Generalising to More Complex Corpora", Ph.D. Thesis, Centre for Vision, Speech and Signal Processing Faculty of Engineering and Physical Sciences, University of Surrey, UK, 2012

[5]. Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on

Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016

[6]. Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S.Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 19, NO. 7, JULY 1997, pp.677-695

[7]. A.D. Gregory, S.A. Ehmann, and M.C. Lin, "InTouch Interactive Multiresolution Modeling and 3D Painting with a Haptic Interface," Proc. IEEE Virtual Reality (VR'02), pp. 45-52, 2000.

[8]. T. Grossman, R. Balakrishnan, G. Kurtenbach, G.Fitzmaurice, A. Khan, and B. Buxton, "Creating Principal 3D Curves with Digital Tape Drawing," Proc. Conf. Human Factors Computing Systems (CHI' 02), pp. 121- 128, 2002.

[9]. Kenji Oka, Yoichi Sato, and Hideki Koike, "Real-Time Fingertip Tracking and Gesture Recognition," IEEE Computer Graphics and Applications, 2002, pp.64-71.

[10]. Erik B. Sudderth, Michael I. Mandel, William T. Freeman, Alan S. Willsky, "Visual Hand Tracking Using Nonparametric Belief Propagation", MIT Laboratory For Information & Decision Systems Technical Report P2603, Presented at IEEE CVPR Workshop On Generative Model-Based Vision, Pp. 1-9, 2004

[11]. Napa Sae-Bae, Kowsar Ahmed, Katherine Isbister, NasirMemon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," Proc. SIGCHI Conference on Human Factors in Computing System,2005, pp.977-986

[12]. W. Makela, "Working 3D Meshes and Particles with Finger Tips, towards an Immersive Artists' Interface," Proc. IEEE Virtual Reality Workshop, pp. 77-80, 2005.

[13]. T. A. C. Bragatto, G. I. S. Ruas, M. V. Lamar, "Real-time Video-Based Finger Spelling Recognition System Using Low Computational Complexity Artificial Neural Networks", IEEE ITS, pp. 393-397, 2006

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

[15]. R. Wang, S. Paris, and J. Popovic, "6D hands: markerless hand-tracking for computer-aided design," in Proc. 24th Ann. ACM Symp. User Interface Softw. Technol., 2011, pp. 549–558.

[16]. Maryam KhosraviNahouji, "2D Finger Motion Tracking, Implementation For Android Based Smartphones", Master's Thesis, CHALMERS Applied Information Technology,2012, pp 1-4

[17]. H.M. Cooper, "Sign Language Recognition: Generalising to More Complex Corpora", Ph.D. Thesis, Centre for Vision, Speech and Signal Processing Faculty of Engineering and Physical Sciences, University of Surrey, UK, 2012

[18]. S. Vikram, L. Li, and S. Russell, "Handwriting and gestures in the air, recognizing on the fly," in Proceedings of the CHI, vol. 13, 2013, pp. 1179–1184.

[19]. EshedOhn-Bar, Mohan ManubhaiTrivedi, "Hand Gesture Recognition In Real-Time For Automotive Interfaces," IEEE Transactions on Intelligent Transportation Systems, VOL. 15, NO. 6, December 2014, pp 2368-2377

[20]. Guo-Zhen Wang, Yi-Pai Huang, Tian-Sheeran Chang, and Tsu-Han Chen, "Bare Finger 3D Air-Touch System Using an Embedded Optical Sensor Array for Mobile Displays", Journal Of Display Technology, VOL. 10, NO. 1,JANUARY2014,pp.1