Gesture-Based Drawing Application: A Survey

Leena I. Sakri, Vijeta Kerur, Gautam Shet, Sohail Mokashi, and Abhishek Patki

Abstract In the software sector, there is a wide range of painting software. Each of these painting applications has its own set of benefits and drawbacks. The common disadvantage in all these programs is that each of them requires a pointing device to draw. Our project is centered on the creation of a system that draws on a screen using human hand gestures. The user doesn't have to make use of any external hardware device to draw. This system makes use of different hand gestures to draw on the screen. This survey paper presents a comparison of various methodologies proposed by different authors on "hand gesture-based drawing" system. These systems make use of Human Hand Gestures for writing characters, words, or free drawing in free space. There are systems that make use of a stylus, digital pen, electronic hand gloves, and other external equipment to interact with computers and there are certain systems that do not make use of such external equipment but, make use of only hand gestures to interact with computers. Each of these methods has its own advantages and disadvantages. The aim of this survey paper is to present the most efficient and accurate method out of the below-elaborated methodologies.

Keywords Hand gesture · Computer vision · Machine learning · Myo-armband · Writing in the air · Fingertip · Human–computer interaction (HCI)

1 Introduction

In this digital world, the traditional arts of writing with pen and paper, chalk, and board are replaced by digital art. Digital and traditional art are both interdependent and interrelated. A typical painting software program utilizes hardware pointer tools or a touchscreen for interactivity. To interface with the software system, we usually need a hardware medium. Natural human–computer interaction may be enhanced by using the hands as an input device directly. This has progressed from "text-based" interfaces to "graphical-based interfaces."

L. I. Sakri (✉) · V. Kerur · G. Shet · S. Mokashi · A. Patki
Department of Information Science and Engineering, SDM College of Engineering and Technology, Dharwad, Karnataka 580002, India
e-mail: leenaignatiusakri@gmail.com

H. K. Deva Sarma et al. (eds.), *Machine Learning in Information and Communication Technology*, Lecture Notes in Networks and Systems 498,
https://doi.org/10.1007/978-981-19-5090-2_27

Digital art will be a powerful means of communication for deaf and dumb individuals, increasing the effectiveness of communication. There are many different approaches to building this hand gesture system. It includes “Background Subtraction”, “2-DifViz”, “Convolutional Neural Network (CNN)”, “Gated Recurrent Unit (GRU)”, “Gaussian blurring”, “thresholding algorithms”, “Haar Cascade Algorithm”, “HSV Color Model”, “Open CV”, and “Numpy” in Python, etc., which gives various accuracy values. Because a finger can be used as a pointer on a computer screen, traditional input devices such as mice, keyboards, and even touch screens will become obsolete. The majority of people are familiar with various painting software, but not with software that allows you to paint or draw using your fingertips (for drawing the shapes, writing the characters, words, etc.). Gesture recognition is the first step towards making computers understand human body language. Hand gestures are used to manage the computer system, making it more user-friendly.

The remainder of the paper is structured as follows: this section provides an overview of the hand gesture. Section 2 discusses the related work. Section 3 examines the methodology, and Sect. 4 concludes with some commentary.

2 Literature Survey

In Ref. [1], the method makes use of a “2-DifViz” algorithm and a “Myo-armband”. In Ref. [2], the system detects hand movements by using a specific object, and it recognizes written text by using CNN. In Ref. [3], the technology uses the axis determined by the hand’s endpoint and centroid to augment a pen on the hand. In Ref. [4], the goal of the project is to create a “motion-to-text” converter. It might be used as a software program in smart wearables that allow people to write in the air. This project is a record of gestures. The finger’s route will be traced using computer vision.

Reference [5] describes a technology that allows you to write in the air without the use of gloves or sensors. Using a fingertip tinted with a certain color, we can create letters and sentences in free space. The user places the color marker on the tip of their fingers. Python programming was used to do this. The user’s drawings can be saved on any platform that the user specifies. In Ref. [6], for pointing, the method employs hand motions. As a result, it adds some originality and improves the software’s user-friendliness. Using hand gestures coupled with specific activities, it will create a variety of shapes on the paint screen. This device simply requires a laptop camera or webcam and hand motions. In Ref. [7], using webcam footage as input, this study tackles the problem of mid-air finger writing. This research shows how to recognize air-writing hand postures by looking at geometrical hand features. It then uses the faster “R-CNN” framework to identify hands, segment them, and count the number of lifted fingers.

In Ref. [8], hand gestures are used to draw on the screen with this technique. To draw, the user does not require an external hardware pointer device. Rather than that, various finger motions are employed to produce drawings. This system is intended to be a “primary-level software” product that provides user amusement. The software can be improved further by enhancing existing functionality. In Ref. [9], the construction of an algorithm that uses hand gestures is an intuitive technique. This research presents a dynamic hand gesture recognition method for elderly people. The program uses optical flow and blob analysis to track six dynamic hand motions and categorise their meanings in a vision-based hand gesture identification system. In terms of detection, tracking, and classification, the experiment yielded positive results for all six hand motions. In Ref. [10], the approach employs a Senz3D commercial depth + RGB camera, which is inexpensive and simple to obtain when compared to other depth cameras. This technique works by evaluating 3D data in real-time and classifying the number of convexity defects into gesture classes using a set of classification criteria. This produces real-time results and eliminates the need for any training data. The suggested approach provides respectable results while consuming very little CPU power.

In Ref. [11], the goal of the technique was to recognize an English character printed in the air with our finger. The LED-fitted finger motions are captured via a web camera and printed onto the display screen when identified. The ultimate purpose of the developed system is to provide a cost-effective solution capable of converting performed English alphabet finger movements to text in a text editor and serving as an efficient means of human–computer interaction. They created the ego-fingertip dataset and proposed the bi-level cascaded convolutional neural network pipelines attention-based hand detector and multi-point fingertip detector. They presented a camera free, three-dimensional, and interactive virtual touch device using only their bare hands in Ref. [12]. A flat panel can detect pictures reflected off a bare finger by incorporating optical sensors within the display pixels and connecting angle scanning illuminators to the display’s edge. In Ref. [13], they propose a vision-based method for mid-air handwriting recognition. Combining handwriting recognition with multi-camera 3D tracking of the hands is what led to this.

3 Methodologies

The following papers show a survey of different approaches used for hand gesture-based drawing systems (Table 1).

AirScript-Creating Documents in Air [1] talked about a technique called AirScript. They presented the 2-DifViz algorithm. It turns a hand in the air (as caught by a Myo-armband) into a collection of x and y coordinates on a “2D cartesian coordinate system” that may be shown on a canvas. They came up with a two-step plan for making papers in the air. It breaks down the task into two separate parts:

Phase I (“2-DifViz”): The hand movement becomes a tangible representation of the number that is being written in the air.

Phase-II “(HDRA)”: A fused classifier is used to recognize handwritten digits. This Airscript allows the user to move autonomously while simultaneously delivering real-time visual feedback on the typed characters, resulting in natural interaction. The

Table 1 Comparison of different methods

Serial no	Research paper	Year of publication	Accuracy (%)
1	AirScript-Creating Document in Air	2017	96.70
2	Text Recognition by air drawing	2009	98.24
3	Interactive Drawing Based on Hand Gesture	2015	Not Provided
4	Aircanvas application using OpenCV and Numpy in python	2021	94.00
5	A Research on Digital Art using Machine Learning Algorithm	2020	93.00

recognition module in this Airscript automatically guesses the information of written documents in the air. They demonstrated a deep learning approach that makes use of sensor data and 2-DifViz's visualization. This module is made up of two components: a "Convolutional Neural Network (CNN)" and a "Gated Recurrent Unit (GRU)." The results of these networks are combined to provide an overall prediction of the letters written in the air. Applications of "Airscript" include smart classrooms, smart factories, smart laboratories, etc., where people can add text without any reference surface. The model was shown to be 91.7% accurate in in-person independent evaluations and 96.7% accurate in in-person dependent evaluations (Fig. 1).

Text recognition by air drawing [2]: In this paper, a method is proposed where text is drawn by the user in the air which is captured by a camera. Here, a particular object is used based on its color to detect any movement made by the user. The color of the object is captured by the lower and upper bounds of HSV. This leads to object detection at every instant. CNN is used in training the model. This model has an accuracy of 98.64% for training and 98.24% for testing. Initially, the camera captures live video. Then, for object detection, the frame, which is in RGB format, is converted to HSV format. Then, a bitwise AND operation is applied between the same frames. A mask is applied while doing bitwise AND. This mask contains the lower and upper bound HSV of the object.

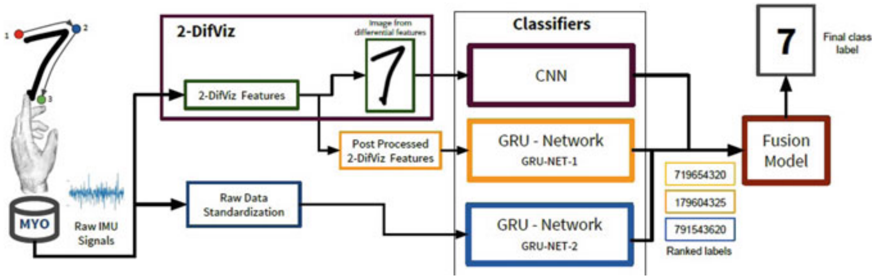

Fig. 1 Flowchart of the system

The frame generated after the bitwise AND operation contains only the specific object. A rectangular contour is drawn over the object to identify it. OpenCV's built-in functions are used to find the contour. To avoid detection of unwanted contours, the area of the object itself is specified. Then the center of the desired contour is determined. To train the CNN model, the "A–Z HandWritten" dataset from Kaggle is used. After training and saving the model, it is used to recognize the character. In this model, since it is trained to recognize the alphabets of the English language, in order to recognize the text, rectangular contours are drawn on the individual characters of the text. The RGB frame is converted to a grayscale frame and then OpenCV's built-in functions are used to create contours. The model is trained five times to get accurate predictions. The overall accuracy of the trained model is 98.64% and the tested model is 98.24%.

Interactive drawing based on hand gesture [3]: The technique suggested in this study is based on a picture captured by a live camera. First, the "Background Subtraction" operation is carried out. The camera image and the reference image are first converted to greyscale images in "Background Subtraction." Here, the initial frame of the camera is taken first, and then "background subtraction" is used. This picture is then used as the reference image. The "background subtraction" procedure has two steps. The first step is to remove the camera input image from the reference image. In a subsequent phase, the reference image is then subtracted from the camera image. The final image data is obtained by using the OR logic operation on the resulting (subtracted) image from the two preceding operations. A "skin area detection" algorithm has been applied to the picture that has undergone "background subtraction". To detect the skin, the input image is first subjected to "background subtraction". Following this, "skin region detection" is carried out in order to extract the hand. The "RGB" color image is transformed into a "YCbCr" color image to detect the skin region. Cb and Cr readings are utilized as references for detecting skin areas. Since the "skin color detection" algorithm is used in detecting a person's hand, objects that have a color similar to a person's hand are also detected. Hence, "labeling" and "post-treatment" procedures are carried out. Here, labeling is done using the "ONE-PASS run-length" algorithm. In order to carry out "one-pass run-length" labeling, a conversion step is required for the final picture that has undergone "skin region detection". The determined regions are arranged depending on their size in the image for which labelling was performed, with the hand being the largest sized region. After detecting the hand image region, the center of gravity and the hand's endpoint are determined. Then, using a certain formula, a spot between those two points is found. This point serves as a reference for drawing on the camera. This also becomes the coordinate for the augmented pen.

The Python Air Canvas application with OpenCV and NumPy [4]: The workflow for this system includes writing in the air, followed by a fingertip detection model, and then tracing a trajectory. Writing in the air is followed by a fingertip detection model and a traced trajectory in this system's workflow. The main goal of the fingertip detection model is to figure out how the air character moves. Normally, a stylus or any air pen with a colored nib is used in air writing. But, this model recognizes the tip of a finger. A dataset is required for this model. A dataset can be created by either importing a video and breaking it into multiple separate images, or directly adding the images. The model can be trained in such a way that if an index finger is shown, then writing mode is on. If two fingers are used, then space is added, three fingers, then backspace, etc. This technology enables the user to operate the machine using his fingertips. They used “Single Shot Detector (SSD)” and “Faster RCNN” pre-trained models in training the dataset. The accuracy of a faster RCNN was higher than that of an SSD. SSDs have two standard object detection models: one to propose regions and the other to classify them, which are used to identify objects in real-time. Faster “RCNN” computed region proposals using the output feature map and sent them to the region of interest pooling layer. Faster “RCNN’s” last linked layer was turned to identify the tip of the finger. This system gave an accuracy of 94%, working well in different backgrounds.

Digital art research using a machine learning algorithm [5]: This system has a workflow of fingertip detection, data collection, processing the captured image, and saving it in a folder. It used an ARM11 Raspberry Pi device, a camera, and a projector. The user’s fingertip has a colored mark. This colored tape makes the camera recognize the gestures of the hand. In this system, they used a shiny green color placed at the tip of the finger of the user. The image captured is transferred to the ARM11 Raspberry Pi for processing. The projector receives the processed info and projects it on a specific screen, which is considered the output by the user. The complete process is managed by the Python language with the help of OpenCV. Camera interfacing is mandatory. An algorithm detecting the gestures indicates that it is functional and processing will be done. By pressing “s” and “q” on the keyboard, an image detected gets saved and the program will be stopped, respectively. Along with Python, the Haar cascade algorithm (HCA) and HSV color model are used. HCA is to detect only green-shiny tips and mask other colors. HCA has four stages, which are: “Haar Feature Selection,” “Integrated Image Creation,” “Adaboost Training,” and “Cascading Classifiers.“ HSV was used to set the min and max values of green on a fingertip. This is achieved by converting BGR into hue, saturation, and value (HSV). Hue is measured in degrees from 0 to 360. Saturation is the percentage of gray in a color that ranges from 0 to 100%. From 0 to 100%, the value describes the brightness or intensity of a color, from 0 to completely black. About 82% overall recognition accuracy is achieved. For four gestures only, an accuracy of 93% is achieved in recognition.

4 Conclusion

After examining the aforementioned publications, we conclude that precision can be attained without the use of an external pointing device by utilizing machine learning approaches (digital pens, styluses, etc.). The use of external pointing devices increases accuracy, but these devices are costly. So, practically, not everyone will be able to use this system for daily use. In our paper, we have selected six best-proposed papers that have outputs similar to our project. From the above-listed methods, we conclude that the “Air Canvas application using OpenCV and NumPy in Python” having an accuracy of 94.00% is best suited for our project. Although “AirScript-Creating Documents in Air” has an accuracy of 96.70%, it uses Myo-ArmBand, which is more expensive, and “Text Recognition by Air Drawing” has an accuracy of 98.24%, since this method uses an object for motion detection and drawing.

References

1. Dash A, Sahu A, Shringi R, Gamboa J, Afzal MZ, Malik MI, Dengel A, Ahmed S (2017) AirScript—creating documents in air. In: International conference on document analysis and recognition
2. Patel J, Mehta U, Panchal K, Tailor D, Zanzmera D (2009) Text recognition by air drawing. In: Conference on computational intelligence and communication technologies (CCICT)
3. Kim T-E, Interactive drawing based on hand gesture
4. Saoji SU, Dua N, Choudhary AK, Phogat B (2021) Air canvas application using opencv and numpy in python. Int Res J Eng Technol (IRJET)
5. Beedkar MS, Wadhekar A (2020) A research on digital art using machine learning algorithm. J Eng Sci (JES)
6. Rokade G, Kurund P, Ahire A, Bhagat P, Kamble V (2020) Paint using hand gesture. Int Res J Eng Technol (IRJET)
7. Ahmed SA, Dogra DP, Kar S, Roy P (2018) Fingertip detection and tracking for recognition of air-writing in videos. https://www.researchgate.net/publication/327570570
8. Bose V, Roy R, Nadirsha M, Raj B, Ajesh M, Jacob PM (2016) Gesture-based painting system. IEEE
9. Ganokratanaa T, Pumrin S (2017) The vision-based hand gesture recognition using blob analysis. In: IEEE international conference science and technology
10. Agrawal R, Gupta N (2016) Real-time hand gesture recognition for human-computer interaction. In: IEEE 6th international conference on advanced computing
11. Ramasamy P, Prabhu G, Srinivasan R (2016) An economical air writing system is converting finger movements to text using a web camera. In: International conference on recent trends in information technology (ICRTIT), Chennai, pp 1–6
12. Wang G-Z, Huang Y-P, Chang T-S, Chen T-H (2014) Bare finger 3D air-touch system using an embedded optical sensor array for mobile displays. J Disp Technol 10(1)
13. Schick A, Morlock D, Amma C, Schultz T, Stiefelhagen R (2012) Vision-based handwriting recognition for unrestricted text input in mid-air. In: Proceedings of the 14th ACM international conference on Multimodal interaction. ACM, pp 217–220