International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 05 May 2024
Page No: 1248 - 1254
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0172

Optical Gesture Recognition for Virtual Cursor Navigation

Mr. Suresh K[1], Sanjay R[2], Vinoth M[3], Sanjay S[4]

[1,2,3,4]Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India.

Emails: *vinothmurugesan18@gmail.com[1,3], 20cs148@kpriet.ac.in[2], 20cs149@kpriet.ac.in[4]*

Abstract

In the rapidly evolving landscape of technology, there is a constant quest for innovations that set themselves apart. Gestures emerge as a highly coveted means of communication with machines. The significance of Human-Computer Interaction becomes evident when harnessing human gestures to govern computer applications. While conventional controls like mice, keyboards, and laser pointers have been commonplace, recent technological strides have introduced more effective techniques for application control. Numerous Gesture Recognition methods, leveraging image processing, have been explored in the past. However, the challenge of recognizing gestures in noisy backgrounds has persistently posed difficulties. In our proposed system, we intend to employ a technique known as Augmentation in Image processing to command a Media Player. This involves recognizing gestures to manipulate operations on the Media Player. Augmentation proves advantageous, particularly in the realm of virtual reality, as it is not constrained by background limitations. Additionally, it eliminates dependencies on specific accessories like gloves or color pointers for gesture recognition. The proposed system caters to users seeking a simplified and enhanced interaction with their computers.

Keywords: *Virtual Cursor Navigation, Hand Gesture, Tracking module, Interface module, Engine module.*

1. Introduction

Hand gestures are among the most natural ways to engage and communicate in any situation; they may be used for anything from carrying out tasks to interacting with others. Nonverbal communication, especially hand gestures, is recognized for its higher effectiveness than spoken communication in some circumstances. Making use of these findings is beneficial and has potential for the field of computer-human interaction research [6]. There is a growing interest in integrating hand gestures and poses into graphical user interfaces (GUIs) and virtual environments due to the investigation of new input methods in the context of modern user interfaces for personal computers, mobile devices, and the developing fields of virtual and augmented reality. While motion-capturing sensors or cameras were a major component of earlier bare-hand tracking techniques, recent trends have leaned towards camera-based approaches to enhance accessibility for a broader user base. Unlike motion-capturing methods that demand specialized hardware like gloves with multiple sensors, camera-based approaches capitalize on the widespread availability of cameras, allowing a diverse user group to experiment with applications sans dedicated equipment [2]. But having widely available hardware by itself does not solve the varied needs of a gesture-based interface's development and run-time stages. Iterative development stages require a large investment of time and effort in order to create a hand gesture interface that is expandable, customizable, and in line with user preferences. Furthermore, proficiency in computer vision and machine learning is needed to achieve precise hand gesture identification. In order to address these issues, Krupka et al. introduced development tools that make use of a basic vocabulary to describe hand motions and positions. Their method, which is based on the finite-state machine (FSM) paradigm, makes it easier for users to customize the interface to meet their unique demands by facilitating the smooth mapping of hand motions to actions. Building upon this framework, our work presents a hand gesture interface that integrates hand motions with the ability to makes use of their unique qualities [2].

2. Previous System

Gesture Recognition strives to decipher human gestures through the application of mathematical algorithms. Currently, various techniques are under exploration, including but not limited to color pointer techniques and numerous other approaches [1].

2.1.Hand Gesture

The MediaPipe framework is utilized for hand motion detection and hand tracking, On the other hand, computer vision uses the OpenCV library. The algorithm uses machine learning techniques to track and identify hand gestures and advice. [3]. MediaPipe, MediaPipe is an open-source tool from Google that can be used in a pipeline for machine learning. Because time series data was used in the construction of the MediaPipe architecture, it is beneficial for cross-platform programming. The MediaPipe architecture may be used with a range of audio and video formats since it is multimodal. The MediaPipe framework is used by developers to create systems for apps as well as to create and analyze systems via graphs. The MediaPipe system's phases are carried out in the pipeline configuration. The developed pipeline is platform-neutral, enabling scalability across desktop and mobile devices. The MediaPipe framework is composed of three main components: a framework for obtaining sensor data, an assessment of performance, and a set of reusable parts known as calculators. [4]. A pipeline is a graph made up of units called calculators, each of which is connected to other units via streams that allow data packets to pass through. Anywhere on the graph, developers can define or swap out custom calculators to create their own unique application. Real-time hand or palm detection and recognition is achieved by employing a single-shot detector model. The MediaPipe utilizes a single-shot detector model. Due to the simplicity of training palms compared to other surfaces, the hand detection module initially trains for a palm detection model. Additionally, non-maximum suppression demonstrates significantly improved performance on small objects such as fists or palms [3].

The OpenCV computer vision library includes methods for object recognition in pictures. Applications of computer vision in real time can be made with OpenCV, a computer vision library for the Python programming language. For analysis tasks like object and face detection as well as image and video processing, the OpenCV library is used in Figure 1. [5].

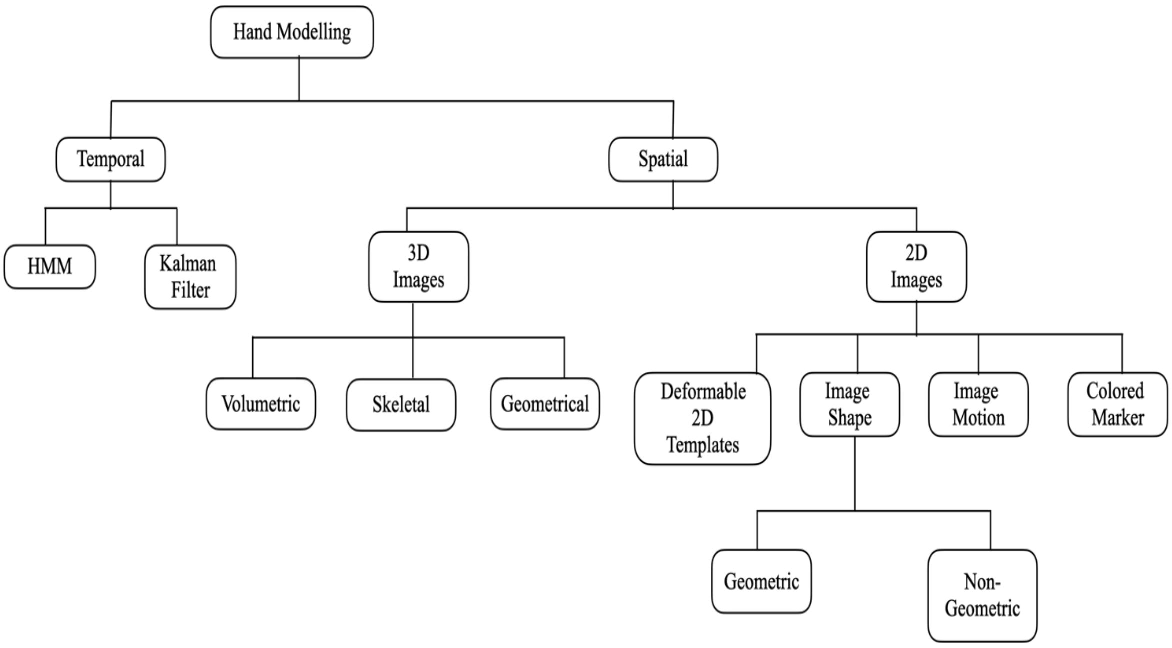

Figure 1 Hand Modeling Methods

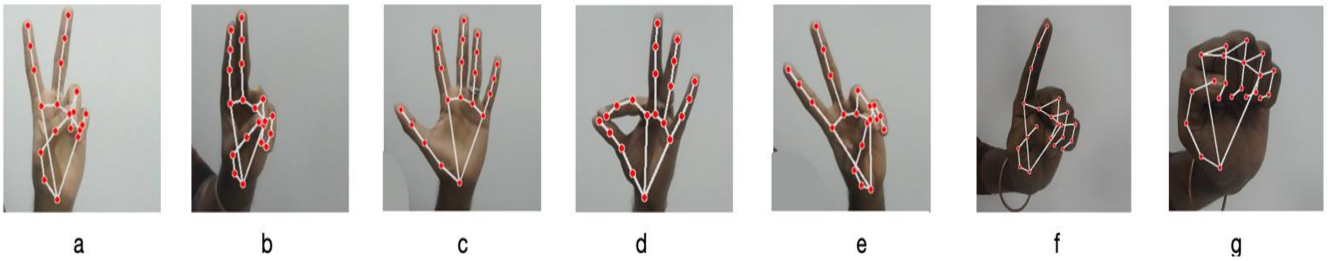

Figure 2 Hand Gestures

Hand gestures are represented as, [7]: (a) Navigate cursor; (b) select option; (c) Neutral gesture; (d) Zoom control; (e) Navigate cursor motion; (f) Double Click; and (g) Drag option in Figure 2.

We examine numerous color pointer approaches, which entail the use of different colored pointers. For hand gesture recognition, these colored points are applied to the fingers [8]. Pranav Mistry on the Sixth Sense has discussed a fairly well-known case of this. The wearable glove, which tracks fingers and hands using computer vision-based recognition, is another extremely well-liked technology. There is technology based on optic light that allows the user to interact with the gadget using basic hand gestures and motions rather than physically touching anything [2]. The RGB served as the foundation for all previous color models. However, there are limitations to all of these solutions. For example, data gloves require constant wear and are more expensive [8]. Color pointers must be carried around, and selecting an object may become confusing if the same color appears inside the recognition area [8]. Additionally, there are skeleton-based techniques where noise hand interpretation is a highly challenging task [8]. Moreover, this approach affects illumination conditions [8].and when a foundation image is needed, all of them are static in nature. scale issues arise when hands of various sizes enter the frame; background is also crucial [9]. Therefore, we need to develop new ideas that significantly reduce all of these drawbacks; in such scenario, augmented reality is a preferable option.

3. Augmented Reality

Gaming systems and medical are just two industries that use augmented reality, which combines the real world with virtual settings [10]. It successfully resolves earlier constraints due to its basic system architecture [11] is show in Figure 3.

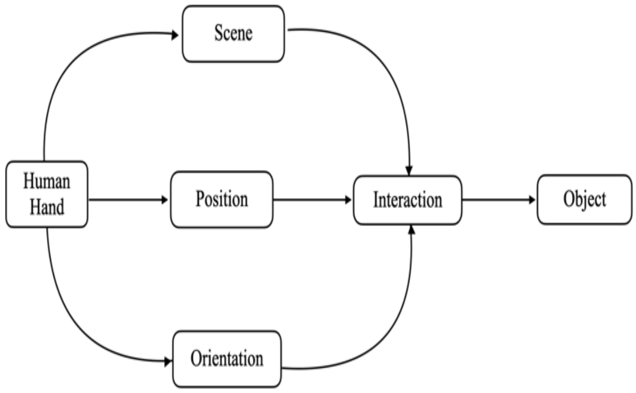

Figure 3 Architecture

4. Proposed System for Hand Detection

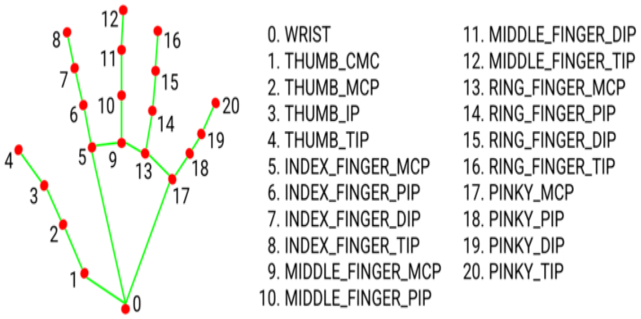

Figure 4 Hand Landmark Detection Model

The main goal of the suggested system is to employ a webcam—which may be an externally connected or built-in camera—to identify the user's dynamic motions in Figure 4.

MediaPipe is used to employ a combination of advanced computer vision algorithms and functions, such as feature extraction, key point detection, and geometric transformations. These techniques enable precise analysis of visual data, facilitating tasks such as hand tracking, pose estimation, facial landmark detection, and gesture recognition with high accuracy and efficiency [12].

The following is the sequence of algorithms that were used in the Model:

- **Hand Tracking:** MediaPipe facilitates hand tracking using a pre-trained neural network model. It accurately locates and tracks the user's hands in real-time, enabling precise gesture recognition.

- **Pose Estimation:** With MediaPipe's pose estimation capabilities, the system accurately captures the user's body poses and movements. This functionality enables the recognition of complex gestures and actions performed by the user [13].
- **Facial Landmark Detection:** MediaPipe includes facial landmark detection algorithms, allowing the system to identify key facial landmarks such as eyes, nose, and mouth. This information enhances the system's ability to interpret facial expressions and gestures.
- **Gesture Recognition:** By analyzing the tracked hand movements, body poses, and facial expressions, the system can recognize and interpret various gestures and actions performed by the user in real-time.

5. Application

The technology uses augmented reality to identify motions that correspond to different factions within the application. The gestures are captured by the live stream, which then presents a virtual menu for choosing an action. By pointing directly at buttons or other activities, users can engage in dynamic interaction. The system is not constrained by the background when utilizing gloves, bare hands, or any other object, demonstrating the effectiveness of augmentation [14].

6. System Description

The following steps constitute the approach:

- The system takes input from a camera which is then processed frame by frame to detect and recognize hand gestures.
- The first step is to detect the presence and location of hands in each frame of the input to accurately identify hands in various orientations, sizes, and lighting conditions.
- Once hand gesture is detected, the landmarks are estimated on the detected hands. These landmarks represent specific points such as fingertips, knuckles, and wrist joints.
- After obtaining the landmarks for each hand, the system analyzes the spatial configuration and movement patterns of these landmarks to recognize specific gestures.
- Once a gesture is recognized, the system generates an appropriate output, which would perform an action in an application for the respective hand gesture.

7. Real-Time Video Capture

The webcam on the computer is the sensor that the system uses to identify the movements of the user's hands. The webcam records live video at a preset frame rate and resolution that are set by the camera hardware. The system allows you to change the resolution and frame rate settings as needed. To process individual frames for gesture analysis and recognition, the recorded video is then split into image frames according to the camera's frames per second (FPS). As is show in Figure 5.

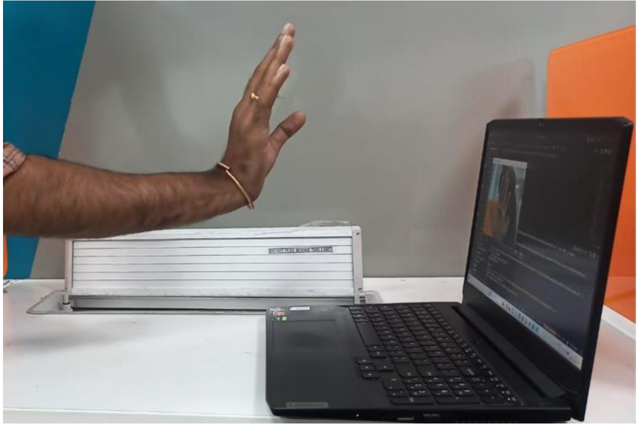

Figure 5 Real-Time Video Capture

8. Tracking Module

The Tracking module has two main objectives. First, by utilizing image region attributes, it seeks to pinpoint the precise region inside the image that matches the guiding hand. Based on the results from the FIZI module, which creates a skin area binary delineation, the region is first segmented into connected segments. The distinct zones are then classified according to a number of attributes, such as position (like the center of gravity and global image location) and size (like area). These unique qualities are then used to categorize these regions. Second, the Tracking module is in charge of updating the features of the chosen region with each frame of the video sequence while continuously monitoring it. Numerous techniques for tracking regions have been put out in the literature. Using the previously indicated region selection method and a comparison with the prior frame, tracking takes place in this scenario frame by frame [15]. The tracking module's final output is an image region that matches the guiding hand and the characteristic traits that go along with it.

9. Mouse Module

The Tracking module's output, which designates a particular region of interest that corresponds to the image's guiding hand, is connected to the Interface module through the Mouse module. Its primary objective is to translate this area of focus into a pinpoint or cursor on the displayed interface. This means that the Mouse module and the Interface module must have a close working relationship. There are several mapping strategies that can be used to connect the Tracking module's output to the Interface module.

Absolute mapping: This method establishes the relationship by using basic ratios. When the interface and picture frame buffer sizes are comparable, it works well.

Linear relative mapping: This technique moves the interface pointer using relative displacements. It may wear the user out after prolonged use and be sensitive to noise.

Non-linear relative mapping: This method uses a non-linear displacement function, just as linear mapping, to ensure minimal shifts for gradual hand movements and greater shifts for faster hand motion, providing proportional displacements. Noise has less of an impact on this strategy.

Every one of these mapping techniques has benefits and applications of its own. Direct System Integration and Interface Control are the two main applications of absolute mapping. On the other hand, non-linear relative mapping is frequently chosen in situations where noise resistance is an important factor. It is possible to attain the best system performance and user experience by carefully choosing the right mapping strategy.

10. Interface Module

The management of the visible interface falls under the purview of the Interface module. Among its primary duties are:

- Presenting and managing the virtual interface in accordance with its XML language specifications.
- Working in tandem with the Mouse module to control user interactions and related activities.

The main reason for selecting the XML language for interface description is its ease of usage. Users can now define interfaces in accordance with their own needs thanks to this. The initial stage involves importing and parsing the XML file that describes the different interface zones and how they communicate with the system. To illustrate, a designated area could be a little square with a text label that appears when the user clicks inside it, mimicking the hitting of a key (like 'A'). Upon loading the XML file, computations for interactions and displays are promptly executed to enhance processing efficiency per frame, which leads to a little decrease in processing time. In addition, a lookup database is designed to rapidly identify the zone corresponding to the user's interaction position, the system concludes its initialization computations, rendering the Interface module ready for display. It takes as input the position of the mouse pointer that the Mouse module sent, calculates the pertinent zone related to the current interaction, gets the actions that go along with it, and sends them to the Engine module to be performed. This all-encompassing integration of XML-based interface development, optimized interaction management, and effective computing procedures highlights the critical role that the Interface module plays in enabling smooth user-system interaction.

11. Engine Module

The final segment encompasses the Engine, which is responsible for completing the designated tasks. The Interface module forwards processed actions to the Engine module for execution. Interestingly, the Engine module's specifications depend on the operating system that is being used. The Engine module keeps a simple structure in spite of its reliance on the operating system: it sends the system calls that match the instructions it has received. Three integers define these directives: one indicates the type of action, and the other two are parameters. The Engine module's streamlined methodology guarantees a smooth and effective conversion of user inputs into commands at the system level. The

Engine module is essential to closing the gap between user inputs and system reaction because it complies with the needs of the underlying operating system and makes it easier for operations to be executed smoothly. This helps the system function more smoothly and responsively as a whole.

12. Implementation

To process perceptual data, we have integrated the MP and Open-MP libraries into our build of the framework. Because OpenCV offers a wide range of capabilities that facilitate smooth image processing and manipulation, it forms the basis for the kernel responsible for image processing. We use the CB lob library to streamline the labeling process for the resultant zones following FIZI processing. Additionally, we make use of the Tiny Xml library's features to perform parsing and loading the interface description from an XML file efficiently. An object-oriented approach is used throughout the code to enable a streamlined and controllable development process by placing an emphasis on modularity and reusability. This careful blending of several libraries and development tools into the Python framework demonstrates our dedication to utilizing state-of-the-art technology and industry best practices to provide a reliable, scalable, and effective solutions. Through the use of an object-oriented programming paradigm and the utilization of well-known libraries, our objective is to optimize development endeavors and guarantee the smooth incorporation of various features into the framework.

13. Experimental Results

There are two main user interfaces offered by the system. The first interface includes full mouse capabilities, including common functions like single and double clicks, vertical mouse wheel scrolling, and cursor movements. With all the regular keys found on a standard keyboard and special keys like space, backspace, and return, the second interface provides a full keyboard experience. Notably, the keyboard's keys are arranged across several screens to provide an intuitive user experience that requires little hand movement. The visitor needs to go to the right page in order to choose a particular letter.

Conclusions

This work offers a novel method for utilizing computer vision and machine learning approaches to develop a contactless human interface. The technique generates a functional virtual mouse and keyboard by means of an image acquisition device. It is feasible to obtain high-quality images by using machine learning, and a parallel implementation ensures that the data collected is analyzed rapidly. We have implemented a machine learning model to execute a native method for accurately recognizing gestures and controlling a virtual mouse, enabling seamless interaction and operation. A custom dataset is curated for training and testing of the model. The dataset comprises 10,000 images captured using a 720p FaceTime HD camera from about 10 individuals to contribute hand gesture samples across 10 distinct class labels to augment the training dataset. Based upon this result, the mediapipe and OpenCV frameworks are implemented in the system for operation. Consequently, this setup enables real-time communication between users and the computer without requiring physical contact. Such a device could find application in numerous other domains, such as surgery, where a touchless interface is required for enhanced precision and safety.

References

[1]. Waghmare Amit, B., S. Sonawane Kunal, A. Chavan Puja, and B. Kadu Nanasaheb. "Augmented Reality for Information Kiosk." system 5, no. 2 (2014).

[2]. Cannan, James, and Huosheng Hu. "Human-machine interaction (HMI): A survey." University of Essex (2011): 27.

[3]. Lin, Wanhong, Lear Du, Carisa Harris-Adamson, Alan Barr, and David Rempel. "Design of hand gestures for manipulating objects in virtual reality." In Human-Computer Interaction. User Interface Design, Development and Multimodality: 19th International Conference, HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part I 19, pp. 584-592. Springer International Publishing, 2017.

[4]. Garg, Pragati, Naveen Aggarwal, and Sanjeev Sofat. "Vision based hand gesture recognition." International Journal of Computer and Information Engineering 3,

no. 1 (2009): 186-191.

[5]. Pavlovic, Vladimir I., 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 19, no. 7 (1997): 677-695.

[6]. Hasan, Mokhtar M., and Pramod K. Mishra. "Hand gesture modeling and recognition using geometric features: a review." Canadian journal on image processing and computer vision 3, no. 1 (2012): 12-26.

[7]. Kaaniche, Mohamed Becha. "Human gesture recognition." PowerPoint slides (2009).

[8]. Ren, Zhou, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. "Robust part-based hand gesture recognition using kinect sensor." IEEE transactions on multimedia 15, no. 5 (2013): 1110-1120.

[9]. Nishi, Takahiro, Yoichi Sato, and Hideki Koike. "Interactive object registration and recognition for augmented desk interface." In CHI'01 Extended Abstracts on Human Factors in Computing Systems, pp. 371-372. 2001.

[10]. Ariyana, Yoki, and Aciek Ida Wuryandari. "Basic 3D interaction techniques in Augmented Reality." In 2012 International Conference on System Engineering and Technology (ICSET), pp. 1-6. IEEE, 2012.

[11]. Zhang, Qijian, Junhui Hou, and Yue Qian. "PointMCD: Boosting Deep Point Cloud Encoders Via Multi-View Cross-Modal Distillation for 3D Shape Recognition." IEEE Transactions on Multimedia (2023).

[12]. Minh, Vu Trieu, Nikita Katushin, and John Pumwa. "Motion tracking glove for augmented reality and virtual reality." Paladyn, Journal of Behavioral Robotics 10, no. 1 (2019): 160-166.

[13]. Ertugrul, Egemen, Ping Li, and Bin Sheng. "On attaining user-friendly hand gesture interfaces to control existing GUIs." Virtual Reality & Intelligent Hardware 2, no. 2 (2020): 153-161

[14]. Lin, Wanhong, Lear Du, Carisa Harris-Adamson, Alan Barr, and David Rempel. "Design of hand gestures for manipulating objects in virtual reality." In Human-Computer Interaction. User Interface Design, Development and Multimodality: 19th International Conference, HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part I 19, pp. 584-592. Springer International Publishing, 2017.

[15]. Tran, Dinh-Son, Ngoc-Huynh Ho, Hyung-Jeong Yang, Soo-Hyung Kim, and Guee Sang Lee. "Real-time virtual mouse system using RGB-D images and fingertip detection." Multimedia Tools and Applications 80 (2021): 10473-10490.