Gesture Based Virtual I/O System

Pravinkumar Sonsare[1], Aryan Gupta[2], Shefali Jindal[3], Shruti Chittora[4], Sumeet Gedam[5] and Sushant Dhote[6]

Shri Ramdeobaba College of Engineering and Management, Nagpur

Abstract
This paper presents an advancement in the technology by replacing mechanical keyboard with Virtual Keyword. The objective is to develop a virtual keyboard using hand/finger recognition which tracks the finger position on the keyboard layout. Real time system is being developed by using camera to capture the stream of hand gestures on keyboard printout. ArUco marker is then used to locate the orientation of keyboard in space in camera's field of view. OpenCv library is further used for perspective transformation of frames. The frames obtained are then used to detect the finger position with respect to the keyboard using Mediapipe. Keystroke are then detected by the hover duration.

Keywords:*Virtual Keyboard, Hand and Finger Detection, Keystroke Detection, ArUco Marker, Mediapipe, OpenCV*

Introduction

Input devices for text entry in computers are keyboards. Technology advances have resulted in a variety of computer devices, such as tablets and smart phones. (Yousaf, Muhammad Haroon et al. 2013). The contemporary keyboard for personal computers was derived from a typewriter keyboard(Nikhil Koul, Pranav Nawathe, 2014). Although the layout has not changed, the computer keyboard detects key presses by forming and breaking electrical contacts(Nikhil Koul, Pranav Nawathe, 2014). The main problem of this concept is that it necessitates a large amount of physical capacity to handle the keyboard, reducing the availability for applications such as mobile phones, in which the screen size is limited(Nikhil Koul, Pranav Nawathe, 2014). Touchscreens were created to address this flaw by integrating the input method into the display directly. However, many people find typing on touchscreens problematic due to the small size of the buttons. There are also potential security concerns because the touchscreen typing keypads are incorporated into the computer software. To address both of these issues, a virtual keyboard was created(Nikhil Koul, Pranav Nawathe, 2014).

A touch-typing device with no physical representation of the sensor regions is known as a virtual keyboard. A virtual keyboard is just another example of technological advancement. Computing is no longer restricted to desktop computers; it may also be found in our mobile phones (Yousaf, Muhammad Haroon & Habib, Hafiz Adnan & Hussain, Fawad & Rizwan, 2013). However, the keyboard interface, i.e., the QWERTY keyboard, has been constant since the beginning(Gade, V.D., Gaikwad, S.S., Gajmal, P.R., Kandhare, N.P. and Rahane, 2016). The virtual keyboard is the most recent advancement in this sector. The virtual keyboard device uses a camera and image processing techniques to allow users to type with a paper keyboard on any flat surface. Virtual keyboards allow us to design keyboards in nearly any language on almost any platform. Because of its tiny size and ease of use, virtual keyboards are an excellent choice for text input on a variety of systems. Because of its tiny size and ease of use, virtual keyboards are an excellent choice for text input on a variety of systems. Virtual keyboards may be classified based on the following features. Touchscreen keyboard layouts or sensing regions on virtual keyboards Keyboard layouts or similar groupings of "keys" or sensing regions projected optically Human hand and finger actions are optically sensed. Depending on whatever device the keyboard is used on, online virtual keyboards for several languages that do not require changing OS settings are available.

This paper shows how gesture detection may be used to improve the design of a virtual keyboard. The virtual keyboard we're showing is an adaptation of a real keyboard that detects keystrokes with the help of OpenCV media pipe and ArUco.

Literature Review

Muhammad Haroon Yousaf. proposed a revolutionary advance in virtual keyboard design for computer device users. The input was to record a hand video in order for the virtual keyboard to work. Then, in the lower portion of the screen, this hand video is supplemented with a backdrop keyboard to offer feedback to the user for smooth interaction with the virtual keyboard. The hand movement is then detected using a video sequence. Finger joints trajectory-based keystroke detection and identification merged with enhanced keys spatial information knows which key is hit. By combining a USB camera with a tablet computer, a real-time system is created. Results are obtained by attempting to employ a low-cost system.

A projected picture of a keyboard is displayed for the user's reference, and the user's input is recognised using a camera and digital image processing techniques. A projector, camera, and a working environment with MATLAB R2010a software were all required(Nikhil Koul, Pranav Nawathe, 2014).

Dr. Nishant Jain (Gade, V.D., Gaikwad, S.S., Gajmal, P.R., Kandhare, N.P. and Rahane, 2016) In order to develop a virtual keyboard for security considerations, I used photo inspection systems to create one. To increase security, it was suggested that the normal password typing keypads used in these systems be replaced with a regular security camera located in the security lock room. With the help of this circle estimation, the red circle is positioned towards the point of touch, informing us of the position of the fingertip and the key pressed accordingly. Image processing methods are used to monitor the user's touch, and output keystrokes are created.

(Yin, Xiaoming & Xie, 2007) introduced various novel methods for segmenting hand images, recognising 2D hand postures, and reconstructing 3D hand postures. Hand-image segmentation, hand feature extraction, and hand gesture identification were all employed. They used a colour segmentation technique based on the RCE Neural network to segment hand pictures, and then retrieved topological properties of the hand from the image of the segmented hand region, analysing these features to distinguish 2D hand positions. To showcase the use of our hand position detection techniques, we created a human-robot interaction system.

(Lombardi, L., & Porta, 2007)have developed a vision-based interface for recognising and interpreting hand gestures made near the mouse as input commands. These motions will have no effect on how you use your mouse normally, but they will speed up the process. A two-dimensional "appearance-based" technique is employed with a quick recognition algorithm. There are two stages to gesture recognition. To acquire and identify the skin pixels, a colour test is conducted first, and then probable markers of each action are explored as part of picture segmentation. Essentially, the article has shown that this technology, which makes use of a standard camera, is adequate for PC input and, in general, does not conflict with standard mouse input.

A overview of modern hand gesture recognition systems is provided(Khan, Rafiqul Zaman & Ibraheem, 2012). A review of modern postures and gestures recognition systems is also provided. A summary of hand gesture research results, databases, and a comparison of the key gesture recognition phases are also provided. Finally, the benefits and downsides of the aforementioned systems' hand gesture-based recognition systems are explored.

(Gade, V.D., Gaikwad, S.S., Gajmal, P.R., Kandhare, N.P. and Rahane, 2016) image processing technique, a virtual keyboard is developed. The system includes one low-quality camera which records RGB images of a user's fingertips on a level surface in order to obtain keystrokes. The performance of the three key image processing phases is evaluated first, followed by a comparison of the two algorithms for locating the user's fingertip in an image. After then, the system's sensitivity to varying illumination and frame rate is evaluated, and the user is given the results.
(Yousaf, M.H., Habib, 2014)demonstrated a virtual keyboard that employed a keystroke detection and recognition technique based on finger joint tracking. The activities of the user's hands are collected in a video sequence for finger joint localization and tracking. Averaging background subtraction, linked component classification, and contour analysis are used to discover the hands region and localise the finger joints. Probabilistic regional density-based kernel tracking is used to estimate finger joint trajectories. The calculated trajectories are subsequently inferred into corresponding feature vectors based on spatial and pathic information. These feature vectors are supplied into a logic-based approach and a Dynamic Bayesian Network for classification, which leads to the detection and identification of keystrokes. Experimentation findings show that with an actual video data collection and real-time implementation, accurate keystroke detection and identification for 28 keys is possible.
(Y. Zhang, 2017)developed a virtual keyboard programme with two subsystems. The processing of custom keyboard input and movies is handled by these two subsystems. Fingertip recognition system and customized keyboard recognition system are the two subsystems in concern. The image is flattened with a Gaussian filter before being separated into RGB channels and binarized in the fingertip identification subsystem. The end of the fingertip is frequently identified within the application at any point in time, implying that the software continuously inspecting the fingertip at all times. Character recognition uses feature extraction, which tries to recall each character and their instant location on paper keyboard. The study report was primarily concerned with the accuracy of a virtual paper keyboard system under various lighting situations, including Amber glow, workplace lights, shaded spaces, flat surfaces, and rising surfaces are all examples of lighting.

(Adajania, Y., Gosalia, J., Kanade, A., Mehta, H. and Shekokar, 2010),(Pucik, 2013) presented a virtual keyboard that extracts the user's fingertips and detects whether or not they have touched the keyboard using a normal web camera and a shadow analysis algorithm.

Methodology

We used OpenCV library and media pipe for hand detection. OpenCV can process images and videos to identify objects and faces. The basic idea was to first map the paper layout of the keyboard. The user's fingers are then mapped to physical keys in the next stage. This is a simple 2D mapping depending on the evidence in the current frame (x, y coordinates) and the relative position of the fingertip to the keyboard's endpoints. Using OpenCV and Media Pipe, we created hand/finger detection. Hands by Media Pipe is a high-resolution hand and finger tracking system. Media Pipe Hands makes use of a machine learning pipeline that consists of numerous models that operate together: An orientated hand bounding box is returned by a palm detection model that acts on the whole picture. To distinguish the hand from the backdrop frame, a histogram-based technique is used. To achieve the best results, thresholding and filtering methods are utilised for background cancellation.

Firstly, we map the keys in a ratio to convert according to the size of extracted image of keyboard. Then we separate the keyboard image from background in given frame, for this we used contour detection. We can recognise the edges of objects and simply pinpoint them in a picture using contour detection. Many intriguing applications, such as image-foreground extraction, simple-image segmentation, detection, and identification, use it as a starting point. The problem with this was that the key controls were detected along with the border of keyboard and also varying lighting conditions made it difficult to identify so we manually took input from the cursor in case of stable webcam where the image is steady. To automate for dynamic case, we tried orb object detection for keyboard layout but it was taking too much computational power and wasn't much efficient for hand detection which works as follows:

- Convert a picture to grayscale using a query.
- Now Initialize the ORB detector and detect the key points in query image and scene.
- Compute the descriptors belonging to both the images.
- Match the key points using Brute Force Matcher.
- Show the matched images.

Finally, we used ArUco markers for corner detection of the layout and it was faster and more accurate than others we used before. ArUco markers are used to determine the location and orientation of the keyboard image in camera's field of view.

Once we got the corner points, we warp transform that frame using inbuilt cv2 warp transform so that we can get a birdeyeview or top eye view for any angle. In this processed image we then did hand detection and focussed on index finger coordinates so this index finger coordinates were compared to the mapping which we did previously and for accomplishing this we used the concept of finding the nearest centre of keys from the index finger coordinates. The mapping of the keyboard and hand detection is illustrated in figure 1.

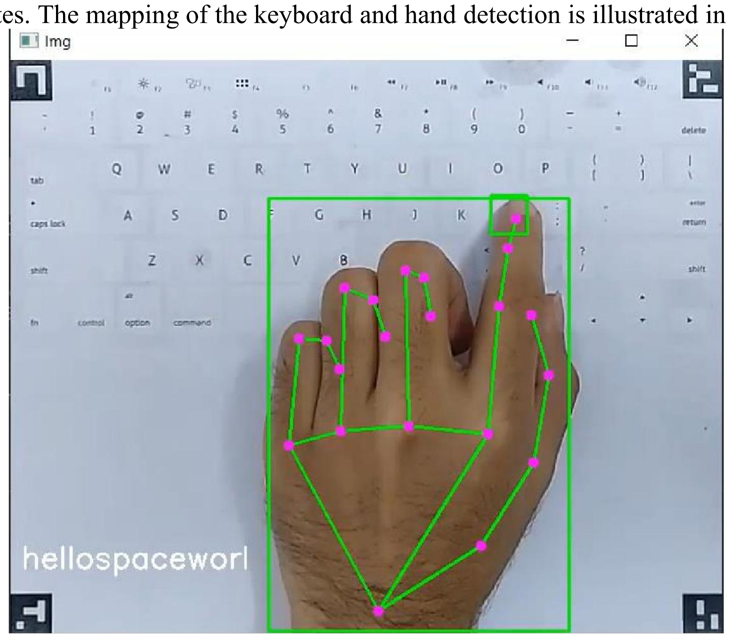

Figure 1:*Mapping of keyboard and hand detection.*

The problem occurring was we were getting redundant keys as input which was occurring due to key detection due to the movement from one valid key to another.

Lastly to eliminate redundant keys the concept was to use that the movement from one key to another the keys occurring between the movements have less frequency than that of the valid keys. So, to eliminate each redundant key, an activation threshold (time) was given so that the redundant keys do not get register. The entire flow of the system is depicted in figure 2.
Another thing for special keys we did was that we gave them the functionality of toggle keys instead of hold and wait and visually represented the status on the windows.

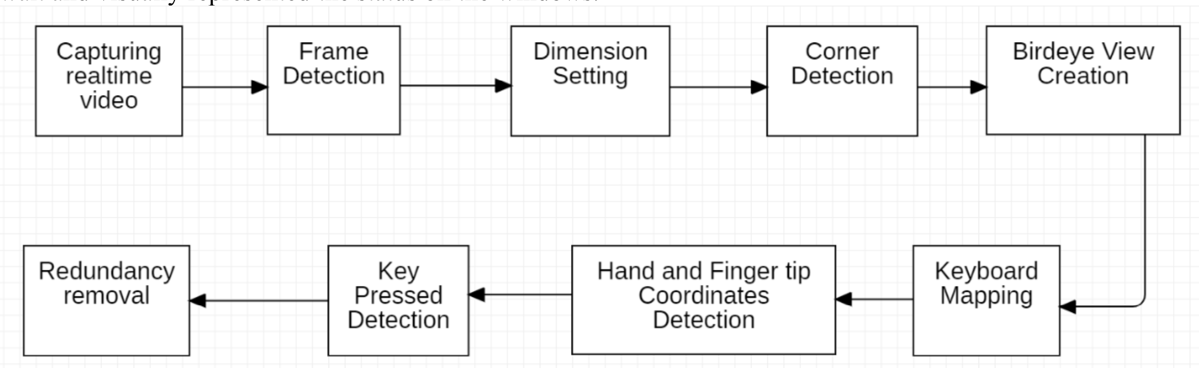

Figure 2:*Flow Diagram*

Result and Discussion

As a result, we proposed a system which shows reliable result and we tested it repeatedly by taking an own input and testing it accuracy. The system's performance is measured in terms of each finger's ability to detect keys and each key's ability to recognise keystrokes.
We tested the above system to analyse its accuracy by typing a paragraph of around 150 words using our keyboard and found that around 35 words have a minor or major typing error.
The accuracy of our system was calculated in two different ways:

1. Considering the word correctness- Accuracy (where accuracy is calculated as number of correct words observed / total number of observations *100) which came out to be around 77%.
2. Considering the character correctness where accuracy was checked character by character which came out to be around 84%.

This was compared with the previous research done in this field and is considered is acceptable value. It can be further improved using different approaches to map the keys and use other efficient algorithms. We also implemented special which demonstrated the functioning of caps lock, ctrl (copy and paste commands), backspace and space bar keys. The below table shows the sample output after testing the system for a given set of words.

Input Word	**Observed word**
The	The
final	Final
application	Xpolication
was	Was
designed	Cesiged
with	Wwith
several	Weveral
goals	Goals

in	in
mind	Mind

Table 1:*Different between input and observed word*

Conclusion

In conclusion, the virtual keyboard application is not a particularly efficient option, but more study and technology breakthroughs might help it develop further. This concept may be developed further in order to better comprehend Augmented Reality. We may use a camera to recognise a keyboard created on a sheet of paper and to detect the fingertip location in the keyboard, anticipating whether or not a touch has been made and generating the appropriate result.

The virtual keyboard, which represents a next generation of human computer interaction and human mobile devices in the development of virtual worlds, is the topic of this study. The desire for small, easy-to-handle mobile devices is increasing. The keyboard device's lightness and flexibility are critical here. The convenience of compression is provided by the virtual keyboard. This keyboard may be customised and resized based on user needs, with only a minor update to the application. As a consequence, the virtual keyboard programme may be inferred that it will make typing easier, more dependable, and secure for the user.

The suggested system can be enhanced in the future in terms of flexibility, user friendliness, and robustness. For keystroke detection and identification, the system's accuracy may be improved further. Once the tracking is perfected, it may be extended to include musical instruments such as the piano and drums.

The system's time and processing complexity are still being optimised, and research is ongoing.

References

Adajania, Y., Gosalia, J., Kanade, A., Mehta, H. and Shekokar, N. (2010) ‘Virtual keyboard using shadow analysis’, *Emerging Trends in Engineering and Technology*, pp. 163–165.

Gade, V.D., Gaikwad, S.S., Gajmal, P.R., Kandhare, N.P. and Rahane, W. (2016) ‘Virtual Keyboard using Image Processing’, *International Journal of Engineering Technology, Management and Applied Sciences*, 4(4).

Khan, Rafiqul Zaman & Ibraheem, N. (2012) ‘Hand Gesture Recognition: A Literature Review.’, *International Journal of Artificial Intelligence & Applications*, 3.

Lombardi, L., & Porta, M. (2007) ‘Adding Gestures to Ordinary Mouse Use: A New Input Modality for Improved Human-Computer Interaction’, in *14th International Conference on Image Analysis and Processing*. Available at: https://doi.org/10.1109/ICIAP.2007.4362821.

Nikhil Koul, Pranav Nawathe, P. T. (2014) ‘Virtual Keyboard’, *International Journal of Scientific & Engineering Research*, 5(4), p. 650.

Pucik, M. H. and J. (2013) ‘Fingertip detection for virtual keyboard based on camera’, in *23rd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 356–360.

Y. Zhang, W. Y. and A. N. (2017) ‘A virtual keyboard implementation based on finger recognition’, *International Conference on Image and Vision Computing*, pp. 1–6.

Yin, Xiaoming & Xie, M. (2007) ‘Hand Posture Segmentation, Recognition and Application for Human-Robot Interaction’, in *Human Robot Interaction*.

Yousaf, M.H., Habib, H. A. (2014) ‘Virtual Keyboard: Real-Time Finger Joints Tracking for Keystroke Detection and Recognition’, *Arab J Sci Eng*, 39, pp. 923–934.

Yousaf, Muhammad Haroon & Habib, Hafiz Adnan & Hussain, Fawad & Rizwan, M. (2013) ‘VISUAL GESTURE BASED AUGMENTED VIRTUAL KEYBOARD SYSTEM.’, *The Nucleus*, 50, pp. 381–386.