SMART GESTURE USING REAL TIME OBJECT TRACKING

Sumanth Bhat, N. Lavanya and M.A. Anusuya
Department of Computer Science and Engineering, Sri Jayachamarajendra College of Engineering, India

Abstract

Gesture can be used to interact with the computer without any physical contact. The use of keyboard and mouse can be minimized. Gesture can be of various types. One such type is movement of hand in a particular posture. To detect these type of gestures first it must be verified that the hand is present in frame and is present in the required posture. The first one is achieved by creating a mask of the frame considering the skin color range in the HSV color space. The later part involves shape matching with some template shape. The shape matching involves computing of central moments between the mask and the template shape. The hand posture defines the start and end of gesture. All the movement of hand between start and end of gesture is tracked and gesture is recognized from the tracked data. For the purpose of recognition, Convolution Neural Network is used. An application is built on recognition. Once a gesture is recognized an event will be triggered.

Keywords:

Contour Detection, Shape Matching, Hue Moments, Convolution Neural Network, Event Triggering

1. INTRODUCTION

Gesture Recognition can be seen as a way for computers to understand human body language, thus building a richer bridge between machine and human than primitive text user interface or even GUI’s. Motion gestures provide a complementary modality for general human computer interaction.

Motion gestures are meant to be simple so that a user can easily memorize and perform them. However, motion gestures themselves are not expressive enough to input text for motion-based control. Also, implementation of image based gesture recognition using different methodologies pose different concerns as the result may vary from camera to camera. The proposed method detects the gesture which is movement of hand in particular posture (index finger opened, other fingers closed). The proposed method mainly involves two phases: hand detection and pattern matching. Hand detection uses various image processing concepts like contour detection, color based masking, image matching etc. The later phase is done using Convolution Neural Network. The proposed system fails under unfavorable illumination and unfavorable background condition. Also as CNN model is being used for matching gesture, the size of training dataset plays an important role in determining accuracy of prediction. The dataset used in this system is a small one. Even though the model may not predict accurately all the time, most of the inaccurate predictions are ignored.

2. RELATED WORKS

Many researchers have contributed towards the area of gesture recognition. In [1], the author proposes an HMM-based method to recognize complex single hand gestures. Gesture images are gained by a common web camera. Skin color is used to segment hand area from the image to form a hand image sequence. Then a state-based spotting algorithm is used to split continuous gestures. After that, feature extraction is executed on each gesture. Features used in the system contain hand position, velocity, size, and shape. Data aligning algorithm is used to align feature vector sequences for training. Then an HMM is trained alone for each gesture.

In [2], the author proposes a depth and skeleton information from kinect are effectively utilized to produce marker less hand extraction. The hand shapes, corresponding textures and depths are represented in the form of super pixels, which effectively retain the overall shapes and color of the gestures to be recognized. Based on this representation, a novel distance metric, super pixel earth mover's distance (SP-EMD), is proposed to measure the dissimilarity between the hand gestures.

In [3], the author proposes a method to use K-means clustering algorithm to partition the input image for segmentation. The bounding box is used to find the orientation. For detection features like centroid, Euclidean distance are measured. Hand is represented by making use of five bits. First bit represents the presence of thumb in the hand gesture. Remaining four bits represents the four fingers.

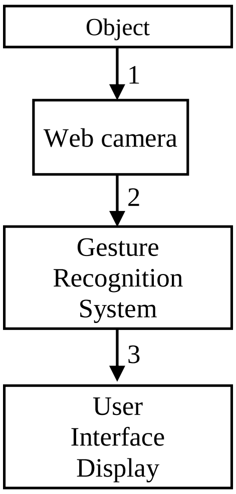
In [4], the author proposes a method of gesture-based HCI (Human Computer Interaction) system which runs with fixed frames per second is proposed. Every time an image is captured by the camera, the captured image is preprocessed and a hand detector tries to filter out the hand image from the captured image; the whole process terminates if there is nothing detected. Then, a CNN classifier is employed to recognize gestures from the processed image, while a Kalman estimator is employed to estimate the position of the mouse cursor according to the movement of a point tracked by the hand detector. Finally, the recognition and estimation results are submitted to a control center; a simple probabilistic model is used to decide what response the system should make.

In [2], the author proposes a method of triggering events after detecting gesture. The main parts of the proposed system are hand detection, finger detection, finger count and event triggering. Hand detection is the inception that involves hand segmentation. It is a color based hand gestures recognition so, different hand gesture color are first taken and then threshold is set manually. Finger identification involves identification of two types of points identified as convex points are defect points. Convex null points are used to detect the fingers. Finger count is performed with the convex hull points. Events triggering involves triggering an event when predefined number of fingers are detected

3. PROPOSED SYSTEM

The Fig.1 depicts proposed architecture with four modules:

- *User*: User is the one who performs the gesture which is to be recognized by the system.
- *Camera*: It records the gesture performed by the user and forwards it to the system.
- *Gesture Recognition System*: This is the main module of the system. It recognizes the gesture performed by the user.
- *User Interface*: User Interface is to make the user aware of the background inconsistencies that would affect the input to the system. Also it maps the gesture with an event and triggers the event corresponding to the gesture performed.



1- Hand Gesture
2- Image captured by the web camera
3- Action to be performed based on gesture

Fig.1. Architecture of Proposed System

The main part of the architecture is gesture recognition. The Fig.2 shows different phases of gesture detection.

- *Camera Input*: The gesture performed by the user is captured by the web camera and is given to the system frame by frame.
- *Hand Detection and Tracking*: In each frame it is checked whether hand is present, and also the posture of hand is compared with template posture.
- *Tracking Condition*: Based on the posture of hand it is decided whether or not tracking should be continued.
- *Feature Extraction and Gesture Recognition*: Once tracking is stopped, image is constructed from tracked data. Feature extraction is done on the image and gesture will be predicted.
- *Event Triggering*: Based on prediction corresponding event is triggered.

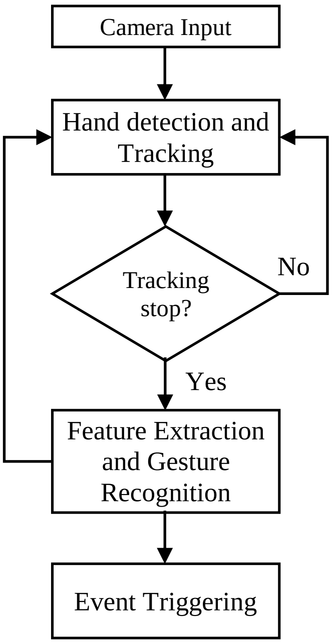

Camera Input
Hand detection and Tracking
Tracking stop?
No
Yes
Feature Extraction and Gesture Recognition
Event Triggering

Fig.2. Different phases of Gesture Recognition

4. METHODOLOGY

The overall process of the proposed gesture recognition system can be compared with that of those mentioned in [4] and [5]. The Fig.3 depicts the approach used.

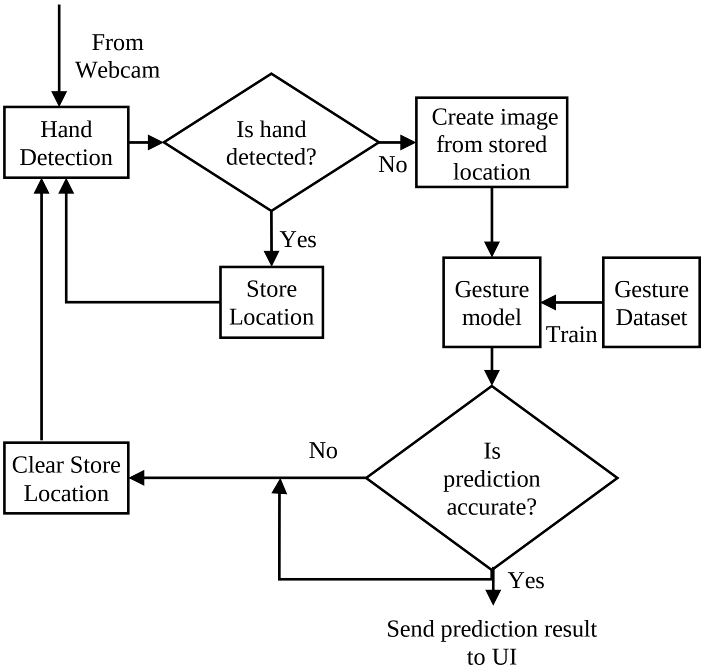


Fig.3. Flowchart depicting the proposed Methodology

The flowchart in Fig.3 can be summarized as follows:

Step 1: The gesture recognition system inputs frames from the camera continuously.

Step 2: In each frame it’s determined whether the frame contains the hand, and the hand is in required posture. The hand posture decides the starting and ending of a gesture.

Step 3: The system stores the location of the tip of the hand until the hand is in required posture.

Step 4: Once the posture changes the gesture is said to be stopped and it will be recognized.

Step 5: After recognizing the gesture event will be triggered only if the confidence of the prediction is high.

4.1 HAND DETECTION

Detecting whether the frame contains the hand is the first step in Gesture Recognition. Hand detection has mainly two steps in the proposed methodology.

4.1.1 Detecting Hand Region:

This is mainly color based hand detection. The frame is first converted to HSV color space. In the HSV image a mask of region having skin color is found. The mask is set using skin color range in HSV color space (both minimum and maximum) which is set manually. Once the mask is found contour detection is done. In the contours the one with maximum area is considered as hand.

OpenCV [10] provides methods to create mask, find contours which are used in the proposed methodology. It is assumed that hand will be the largest skin color shape in the frame.

4.1.2 Checking for the Posture of the Hand:

Once the mask of hand region is obtained, it's checked whether the hand region is in the required posture. This is done using a function matchshapes in OpenCV [10]. This function computes a type of central moments call Hue moments and using those values computes a value showing the closeness of the mask and the template image of the hand in required posture.

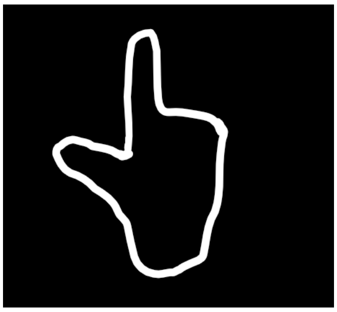


Fig.4. Required Shape of Hand

The value returned by the function is used to decide whether or not the hand is in required posture. The threshold value which is to be used for decision is found out manually through trial and error.

4.1.3 Termination Criteria:

The termination or delimiting criterion is important to the system since there is no pen-up motion, as in the case of traditional online handwriting. The posture of hand decides termination criteria for hand tracking. Once hand posture changes from template posture (Fig.4) termination criteria is said to be satisfied. Until termination criteria is satisfied the location of hand is stored. Once the termination criterion is satisfied, the stored location are used to create an image depicting the gesture drawn by user.

4.1.4 Feature Extraction and Gesture Recognition:

Once hand detection stops i.e. once the user changes hand posture to a different posture (termination criteria), the gesture will be recognized using the image generated. For this purpose Convolution Neural Network is used. The image is inputted to the CNN model and it predicts the result.

The CNN model first extracts number of features from the input image [9]. Using these features, it gives prediction upon the gesture performed.

Before using CNN model, it has to be trained. For training, a dataset which is created manually is used. To create the dataset a procedure similar to the gesture recognition is used. This image generated after termination criteria satisfied is saved. The dataset used contains 6 categories of image. The six categories are numbers 0-5. Each category contains 30-40 images.

Once the model is trained it is used for detection. The model outputs prediction of chances of the input image to be each one of the categories. The category with the highest chance is chosen as output.

4.1.5 Event Triggering:

Once the gesture is detected, all that is left is event triggering. Each gesture is mapped to a corresponding event and the event corresponding to detected gesture must be triggered. As the size of the model is very small, to prevent wrong event triggering a threshold is set for chances predicted by the model. Only if prediction confidence is high the event will be triggered.

5. RESULT AND CONCLUSIONS

The system implemented triggers event according to Table.1.

Table.1. Event Triggering

Gesture Drawn	Event Triggered
0	Decrease Brightness
1	Take Screenshot
2	Increase Brightness
3	Decrease Volume
4	Increase Volume
5	Disconnect

The Fig.5-Fig.10 shows different test cases of the implementation of proposed system.

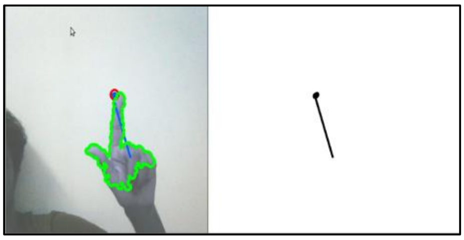


Fig.5. Start of Gesture Tracking

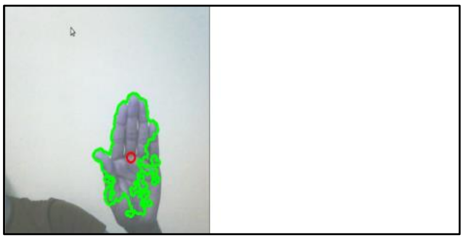


Fig.6. Stop of Gesture tracking

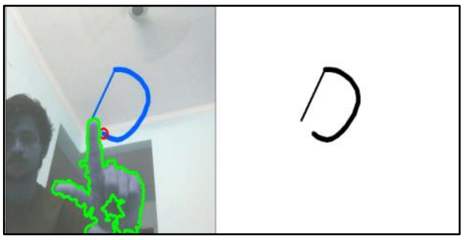


Fig.7. Gesture Recognition Test Case 1

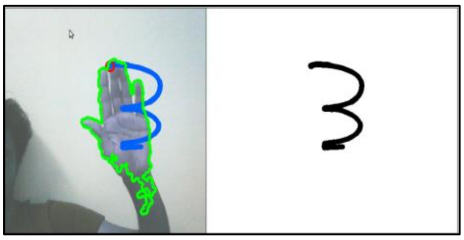


Fig.8. Gesture Recognition Test Case 2

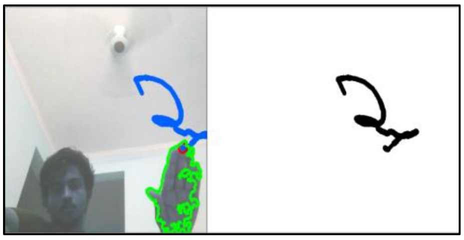


Fig.9. Gesture Recognition Test Case 3

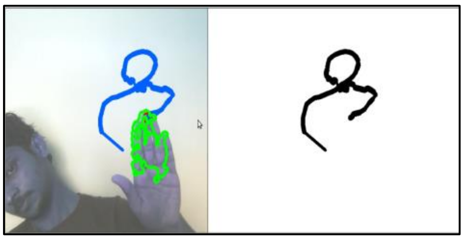


Fig.10. Gesture Recognition Test Case 4

The Fig.5 show the hand in required posture, which means gesture tracking should start. This is depicted by the blue line drawn towards the tip of finger (indicated by the red circle). The Fig.6 shows the hand not in required posture, which means the tracking should end. This can be seen in the image. The Fig.7-Fig.10 shows different cases of Gesture Recognition which is summarized in Table.2.

Table.2. Gesture Recognition Test Cases

Remarks	Outcome
Success in recognizing 0 confidently	Brightness Decreased
Success in recognizing 3 confidently	Volume Decreased
Recognized as 2 but confidence level is low	Gesture Discarded
Recognized as 0 but confidence level is very low	Gesture Discarded

- In [5], the authors proposes event triggering methodology of triggering event based on gesture recognition. The following are the comparisons made between methodology of [5] and proposed methodology:
- The type of gesture used in [5] is finger count, whereas the gesture in this proposed methodology is movement of hand in particular posture.
- The gestures in [5] can be said to be static in nature, i.e. there is not any movement of hand. This is not the case in the proposed work.
- For the purpose of feature extraction centroid calculation, thumb detection, and finding distance between fingers are done in [5]. In the proposed methodology feature extraction is done by CNN.
- In [5], the authors has implemented only using OpenCV [10], the proposed methodology is implemented using OpenCV [10] as well as Keras [11] and TensorFlow.

5.1 LIMITATIONS

The system fails mainly in two conditions:

- When the illumination is not proper.
- When the background is not proper.

As the input is through web camera proper illumination is required. This can be resolved by either using a better camera for input or using image processing concepts to enhance the input frame. The second condition is due to using color based detection. If the background contains a large portion having color belonging to skin color range then the hand will not be detected as the largest contour and shape matching fails. Therefore we can say that the accuracy of gesture identification is affected by the web camera and background.

Another drawback of the system is the size of the dataset used. As the CNN is trained on a smaller dataset accuracy of gesture recognition is not high. This can be increased by training the model with larger dataset with more variations to increase the accuracy of prediction.

6. CONCLUSIONS

In this study, we detect hand gestures hence the presence of hand and required posture must be verified in the frame. The first one is achieved by creating a mask of the frame considering the skin color range in the HSV color space. The later part involves shape matching with some template shape. The shape matching involves computing of central moments between the mask and the template shape. The hand posture defines the start and end of gesture. All the movement of hand between start and end of gesture is tracked and gesture is recognized from the tracked data. For the purpose of recognition, Convolution Neural Network is used. An application is built on recognition. Once a gesture is recognized an event will be triggered.

REFERENCES

[1] Z. Yang, Y. Li, W. Chen and Y. Zheng, "Dynamic Hand Gesture Recognition using Hidden Markov Models", *Proceedings of 7th International Conference on Computer Science and Education*, pp. 360-365, 2012.

[2] C. Wang, Z. Liu and S. Chan, "Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera", *IEEE Transactions on Multimedia*, Vol. 17, No. 1, pp. 29-39, 2015.

[3] M. Panwar and P. Singh Mehra, "Hand Gesture Recognition for Human Computer Interaction", *Proceedings of International Conference on Image Information Processing*, pp. 1-7, 2011.

[4] P. Xu, "A Real-Time Hand Gesture Recognition and Human-Computer Interaction System", *Proceedings of International Conference on Computer Vision and Pattern* Recognition, pp. 1-8, 2017.

[5] R. Parashar and R. Pareek, "Event Triggering using Hand Gesture using OpenCV", *International Journal of Engineering and Computer Science*, Vol. 5, No. 2, pp. 57-60, 2016.

[6] K.B. Shaik, P. Ganesan, V. Kalist, B. Sathish and J.M.M. Jenitha, "Comparative Study of Skin Color Detection and Segmentation in HSV and YCBCR Color Space", *Procedia Computer Science*, Vol. 57, pp. 41-48, 2015.

[7] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, Vol. 120, pp. 122-125, 2000.

[8] J. Brownlee, "Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras", Available at: https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/

[9] S. Dey, "CNN Application on Structured Data-Automated Feature Extraction", Available at: https://towardsdatascience.com/cnn-application-on-structured-data-automated-feature-extraction-8f2cd28d9a7e

[10] OpenCV 4.1.0, Available at: https://opencv.org/releases/

[11] Keras Documentation, "Conv1D", Available at: https://keras.io/layers/convolutional.