A State Transition Based Approach to Recognize Gestures Using Multi Level Color Tracking

Suresh Alse, Shreekanthadatta Eligar
Department of Information Technology
National Institute of Technology Karnataka
Surathkal, India
{sur.10it106,sge.10it91}@nitk.edu.in

Shridhar G.Domanal, G. Ram Mohana Reddy
Department of Information Technology
National Institute of Technology Karnataka
Surathkal, India
{shridhar.domanal, profgrmreddy}@gmail.com

***Abstract*—Gesture recognition is one of the most challenging tasks in Human computer interaction and it has wide range of applications. Here we propose a gesture recognition system which does not involve training the machine in order to detect simple gestures. The proposed technique involves multi level color (color inside color) tracking where region of interest (ROI) is found with respect to the outer color and then with respect to the next inner color and so on. Then the technique involves State transition based approach to recognize gestures where the tracked data is broken down into a sequence of transitions which determine a gesture. This technique is used to develop Jarvis[8], an open source project to control Linux systems using gestures and object tracking.**

Keywords—Thresholding, HSV, Target object, Color sequence, Jarvis

I. INTRODUCTION

Today we can see the world wanting everything in their life to happen at the flick of their hand without putting much physical effort. Gesture recognition is one phenomenon which can help us go in this direction. It has several applications as mentioned in "Gesture Recognition: A Survey" [2]:

- Better Human Computer Interaction
- Recognition of sign language.
- Distance learning
- Forensics

Gesture recognition is not a new concept. It has been a part of us since a long time. However most of the techniques used to achieve this are complex. Systems dealing with simple gestures are also forced to use complex machine learning algorithms consuming more space and time. Hence we would like to propose a new technique to detect simple gestures using state transition based approach.

Our technique first involves multi level color tracking. Say the target object is concentric filled circles of various colors. Here the image is converted to HSV and then thresholding is done on the image based on the outer color of the target object. After that, the area bounded by the maximum and the minimum X and Y in Cartesian plane are found. Then the extracted region is again exposed to thresholding with respect to the next color and again area bounded by maximum and minimum X and Y in Cartesian plane are found. This goes on and on till we find the centroid of the object. Thus by keeping track of the center we can obtain the path taken by the gesture.

By this we can eliminate several similar colored objects in the background. Also it is now possible to ease the HSV range while thresholding.

After obtaining the path taken by the gesture we need to find the states attained by the path during the gesture. In this paper we are concentrating only on four possible states which are Horizontal Left to Right (HLR), Horizontal Right to Left (HRL), Vertical Top to Bottom (VTB) and Vertical Bottom to Top (VBT). Any gesture is represented as a sequence of transitions from one to another. For example:

$$\text{HLR} \rightarrow \text{VTB} \rightarrow \text{HLR (Z shape gesture)} \qquad (1)$$

For the set of points extracted from multi level color tracking the sequence of transitions is obtained in above fashion which uniquely represents a gesture.

In Jarvis[8] to handle gesture and non gesture patterns a flag object was used, i.e whenever an object of a particular color is detected we can safely say that gesture has started. When the flag object becomes invisible we can say that gesture action has stopped.

II. LITERATURE SURVEY

A. HMM-Based Threshold Model Approach for Gesture Recognition [3]

Hyeon-Kyu Lee and Jin H. Kim proposed a gesture recognition technique using Hidden Markov Model. 'In order to separate gesture and nongesture pattern a model called "threshold model" that calculates the likelihood threshold of an input pattern and provides a confirmation mechanism for the provisionally matched gesture patterns was proposed. The threshold model is a weak model for all trained gestures in the sense that its likelihood is smaller than that of the dedicated gesture model for a given gesture. Consequently, the likelihood can be used as an adaptive threshold for selecting proper gesture model. Then the states with similar probability distributions are merged, utilizing the relative entropy measure'.

978-1-4673-6217-7/13/$31.00 ©2013 IEEE

B. *Gesture modelling and Recognition using Finite state mechanics[4]*

Pengyu Hong, Matthew Turk and Thomas S Huang proposed a state based approach to gesture learning and recognition. 'Each gesture is defined as a sequence of states in spatial-temporal space. From training data corresponding to a gesture, spatial information is learnt and then they are grouped into segments. Then along with temporal information a Finite State Machine(FSM) recognizer is built. Each machine will have a FSM corresponding to it'.

C. *Soft Computing and Conectionist Approach [2]*

'Soft computing is a consortium of methodologies that works synergistically and provides flexible information processing capability for handling real-life ambiguous situations [9]. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions'.

As we can see most of the algorithms are based on machine learning which can be unnecessarily difficult, time consuming and less accurate in some cases.

III. METHODOLOGY

This section describes the steps to be followed in order to extract gesture using our approach.

In our approach the gesture video stream is the source. Once the gesture input has started, each of the frame in the video is subjected to Multi level color thresholding to obtain the point which needs to be tracked. Once found, the point is recorded to keep track of the gesture. When the gesture input stops, the tracked data is analyzed using our approach to obtain a sequence of state transitions which uniquely describes a gesture. Fig.1. Shows how the process flows in our approach.

(Multi level color thresholding → Tracking) → Gesture Recognition (2)

Steps involved in our approach are clearly described in the following sub sections.

a. *Multi level color thresholding*

As we know the drawback in every color based object tracking is that the target object needs to have a color different to that which exists in its background. One way to resolve this is to know the exact color range of the target object. However it is practically impossible as the color of the target object keeps changing on various physical conditions. Hence we propose a simple way to solve this problem, which is to make the target object composed of several colors one inside the other. For instance say the target object consists of yellow color on the outside and has green color on the inside as shown in Fig 2.
Note that there are objects which have different colors including yellow and green and we need only the object which follows the color sequence.

Color sequence: Yellow → Green (3)

Note that the length of the color sequence which is the number of colors in the color sequence increases the accuracy of our approach.

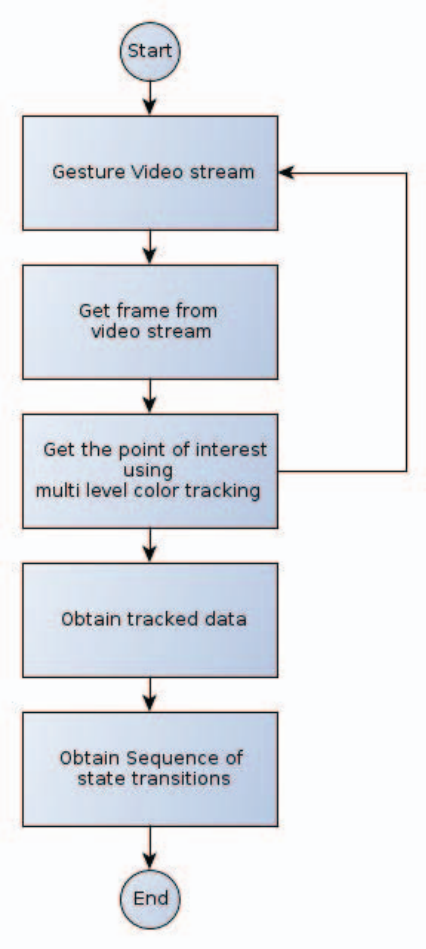

Fig. 1. Process diagram of our approach

Fig. 2. Snapshot of the original frame

Threshold the frame with respect to the first color in the color sequence. Here it filters out the parts of the frame which doesn't have yellow.

Fig. 3. Frame after thresholding with respect to yellow

Now get the maximum and minimum values of each of the clusters in the thresholded image which will give the region of interest for further processing.

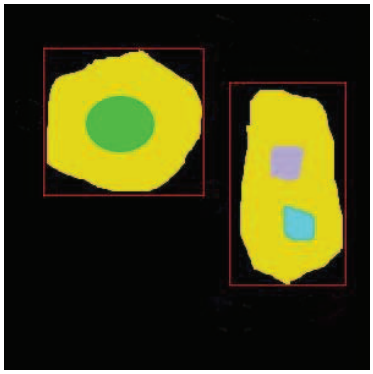

Fig. 4. Region of interest for further processing

Now thresholding is applied on the ROI with respect to the second color in the color sequence.

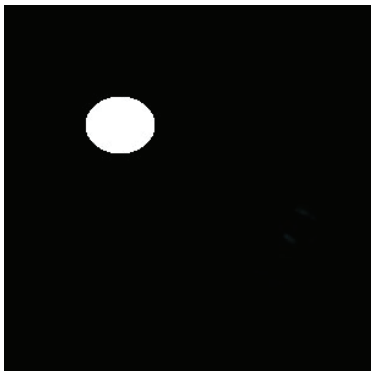

Fig. 5. Required region extracted

This process is repeated for all the colors in the color sequence till the required ROI is extracted. Now its centroid gives the point which has to be tracked.

This process is done for every frame during the tracking process. At the end of tracking we will have the tracked data consisting of an array of points. This is then passed into the gesture recognition system.

b. Gesture Recognition

In this step the tracked data is processed to obtain the gesture as a sequence of state transitions.

i. Grid construction

- First maximum and minimum X and Y values of the points in the tracked data found.
- A 5x5 grid is constructed of cell size:
  width=(maxX-minX) /5
  height=(maxY-minY)/5 (4)
  as shown in Fig. 6.

ii. States

- Start from the first point in the tracked data and see to which cell it belongs. In Fig.6 the first cell belongs to cell (1,1)
- Now consider the second point in the tracked data and repeat the above process. Continue doing this till you find the point which occurs in a different cell.
- If the previous cell was (i,j) and the current cell is (m,n):
  a. If (m,n) = (i+1,j) then the line is horizontal and is moving from left to right (HLR).
  b. If (m,n) = (i-1,j) then the line is horizontal and is moving from right to left (HRL).
  c. If (m,n) = (i,j+1) then the line is vertical and is moving from top to bottom (VTB).
  d. If (m,n) = (i,j-1) then the line is vertical and is moving bottom to top (VBT)
- Let S be the sequence of state transitions in the space defined by R={HLR,HRL,VTB,VBT} (5)

iii. Extract sequence of transitions using Grid

- Determine the first state of the gesture. In Fig. 6 it is HLR
- Then compute the states till you find a change in state. In Fig. 6 after HLR you find VTB
- Similarly compute all the changes in states to determine S. In Fig 6
  S = HLR → VTB → HRL → VBT
- The sequence of states S determines the gesture.
- There are certain exceptions to this method. When the size of the grid is very small (if camera resolution is 640x480, less than 25px should be considered as small):
  i) If the height is very small and x value is increasing then S = HLR
  For example: Fig.7.

ii) If the height is very small and x value is decreasing then S = HRL

iii) If the width is very small and y value is increasing then S = VTB
For example: Fig. 8.

iv) If the width is very small and y value is decreasing then S = VBT

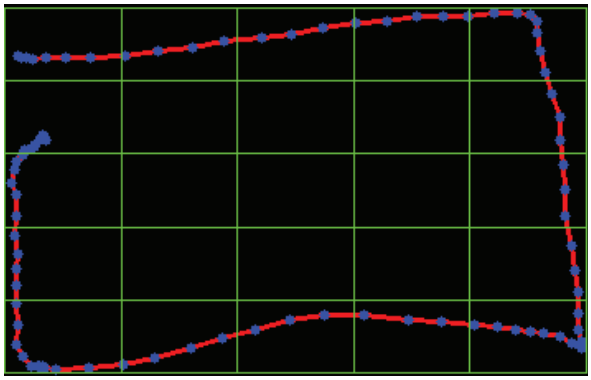

Fig. 6. Blue : Points in tracked data
Red : Line connecting the tracked points
Green : Grid

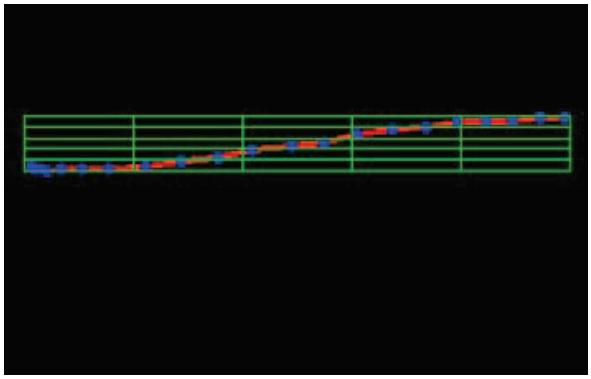

Fig. 7. Horizontal line Left to Right (HLR)

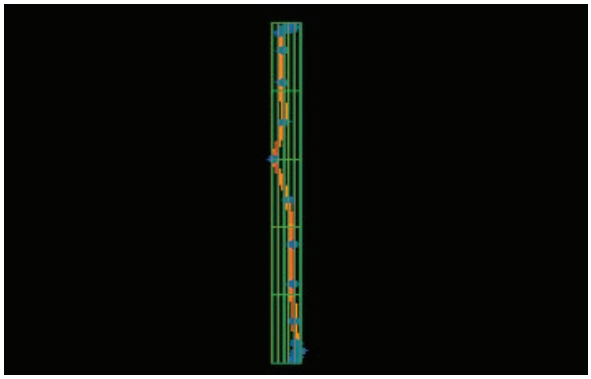

Fig. 8. Vertical line Top to Bottom (VTB)

c. Pseudo code

The following is the pseudo code for the implementation of our approach. The following code is given with comments exhaustively for better understanding and practical implementation purpose.

```
Start
Input: Series of frames in gesture video stream, F
Output: Sequence of state transitions, S
//First get tracked data in an empty array
Track_data=[]
//Every frame must be taken into consideration
for frame in F
do
    //ROI is initialized to the size of the frame
    ROI=size of frame
    //Every color from a color sequence is considered
    for color in (Color Sequence of frame )
    do
        //ROI is found and updated accordingly
        ROI=extract_region(frame,color,ROI)
    Done
    //Point of interest is the centroid of the final ROI
    point_of_interest=centroid(ROI)
    //Point of interest is stored as tracked data
    Track_data.append(point_of_interest)
Done
//Now we have tracked data
//Now obtain S. The gesture sequence
//Boundaries found.
//Grid number is set to 5
//Grid size threshold set to 25
minX, minY, maxX, maxY=find_info(track_data)
sizeX=(maxX-minX)/5
sizeY=(maxY-minY)/5
S=[]
//track_data.first refers to the first point in track data
//track_data.last refers to last point in track data
//*.X refers to x coordinate of the point.
//*.Y refers to y coordinate of the point.

//First perform checks for grid size less than threshold
//Check for horizontal left to right
If sizeX<25 and track_data.first.X<track_data.last.X
        S=HLR
        End

//Check for horizontal right to left
If sizeX<25 and track_data.first.X>track_data.last.X
        S=HRL
        End

//Check for Vertical bottom to top
If sizeY<25 and track_data.first.Y<track_data.last.Y
        S=VBT
        End

//Check for vertical top to bottom
If sizeY<25 and track_data.first.Y>track_data.last.Y
        S=VTB
```

```
        End

//If grid size is greater than grid size threshold
// Note that cluster here refers to the grid element

//Initialize old cluster with cluster of first track data
old_cluster=find_cluster(tracked_data.first)

//Loop through all the tracked points
for point in tracked_data
do
//new_cluster is the cluster of point under consideration
        new_cluster=find_cluster(point)
//Trend here refers to HLR or HRL or VTB or VBT
/*If new and old clusters are not the same then find trend of
the tracked data */
        if old_cluster!=new_cluster
                trend=find_trend(old_cluster,new_cluster)
//If there is a change in trend store it in S
        if trend and trend !=S.last
                S.append(trend)
        old_cluster=new_cluster
done
//Now S gives the gesture
End
```

IV. CONCLUSION

We proposed a state transition based approach to recognize gestures using multi level color tracking. The proposed algorithm first detects the point of interest by thresholding and extracting the ROI multiple times depending on the colors in the color sequence of the target object. Once the point of interest is fund, the response is recorded. The series of recorded response is the tracked data which gives the path taken by the gesture. Then state transitions are found for the tracked data which describes the gesture.

This technique is used and tested by us in Jarvis[8] which is a tool which can be used to control a Linux system using gestures. Many people were asked to use Jarvis and feedback was taken. Most of them gave a positive feedback, but some said that if the system were to detect diagonal lines it would have been better. For future work the algorithm can be extended to incorporate four more states representing diagonal lines viz, (Top→Botton, Left→Right), (Top→Botton, Right→Left), (Bottom→Top, Left→Right) and (Bottom→Top, Right→Left) respectively.

V. ACKNOWLEDGEMENT

The proposed approach is implemented in Jarvis[8] which is a human computer interaction project done under the guidance of G Ram Mohana Reddy and Shridhar G.Domanal from Department of Information and Technology in National Institute of Technology Karnataka, Surathkal

VI. REFERENCES

[1] A. F. Bobick and A. D.Wilson, "A state-based approach to the representation and recognition of gesture," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 12, pp. 1235–1337, Dec. 1997.

[2] Sushmita Mitra and Tinku Acharya, "Gesture Recognition: A Survey", IEEE transactions on systems,man, and cybernetics—Part C Applications and reviews, vol. 37, no. 3, may 2007

[3] Hyeon-Kyu Lee and Jin H. Kim, "HMM-Based Threshold Model Approach for Gesture Recognition".

[4] Pengyu Hong, Matthew Turk and Thomas S Huang, "Gesture modelling and Recognition using Finite state mechanics".

[5] K. Vaananen and K. Boehm, "Gesture Driven Interaction as a Human Factor in Virtual Environment-An Approach with Neural Networks" Virtual Reality Systems, R. Earnshaw, M. Gigante, H. Jones, eds., chapter 7, pp. 93-106. Academic Press,1993..

[6] C. Cedras and M. Shah, "Motion Based Recognition: A Survey,".

[7] J. Rittscher and A. Blake, "A Probabilistic Background Model for Tracking," Proc. Europen Conf. Computer Vision, vol. 2, 2000.

[8] http://www.github.com/alseambusherr/jarvis

[9] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," Commun. ACM, vol. 37, pp. 77–84, 1994.