A Hidden Markov Model based Dynamic Hand Gesture Recognition System using OpenCV

Rajat Shrivastava
Dept. of Electronics and CommunicationEngineering
Maulana Azad National Institute of Technology
Bhopal-462001, India
rajatshrivastavanitbpl@gmail.com

***Abstract*- In this paper we propose a novel and faster system for dynamic hand gesture recognition by using Intel's image processing library OpenCV. Many hand gesture recognition methods using visual analysis have been proposed: syntactical analysis, neural networks, the hidden Markov model (HMM). In our research, a HMM is proposed for hand gesture recognition. The whole system is divided into three stages detection and tracking, feature extraction and training and recognition. The first stage uses a more non-conventional approach of application of Lαβ colour space for hand detection. While the process of features extraction is the combination of Hu invariant moments and hand orientation. For the training, Baum-Welch algorithm using Left-Right Banded (LRB) topology is applied and recognition is achieved by Forward algorithm with an average recognition rate above 90% for isolated hand gestures. Because of the use of OpenCV's inbuilt functions, the system is easy to develop, its recognition rate is quite fast and so the system can be practically used for real-time applications.**

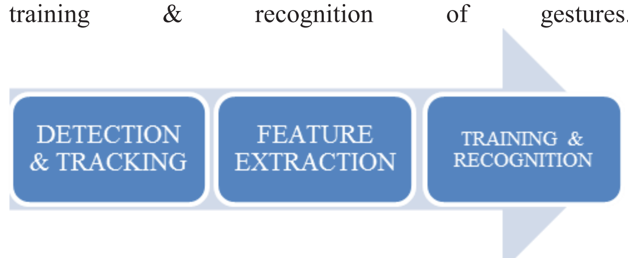
***Keywords*- Hand Gesture Recognition, Hidden Markov Model (HMM), Lαβ colour space, Hu Invariant Moments, OpenCV.**

I. INTRODUCTION

The goal of Human Computer Interaction (HCI) is to bring the performance of human machine i.e. computer similar to human-human interaction. Gestures play an important part in our daily life, and they can help people convey information and express their feelings. To say technically, a gesture is spatio-temporal pattern which may be static, dynamic or both [1]. Among different body parts, the hand is the most effective, general-purpose interaction tool. So, in the field of Human-Computer Interaction (HCI) hand gesture recognition is an active area of research. On the basis of current research work done, hand gesture recognition techniques are categorised into two: glove based method and vision based method. Glove-based gesture interfaces require the user to wear a cumbersome device, and generally carry a load of cables that connect the device to a computer. There are many vision-based techniques, such as model-based and state-based which have been proposed for locating objects and recognizing gesturers. Recently, there have been an increasing number of gesture recognition researches using vision-based methods. For a more natural interface, hand gesture must be distinguishable from visual images without the aid of any external devices. If we go through the vision based methods, a lot of techniques come into highlight such as syntactical analysis, neural network based approaches, Fuzzy systems and the HMM based recognition[2, 3]. For continuous hand gestures HMM algorithm comes out to be the best in the lot because HMM is a statistical model and is capable of modelling spatio-temporal time series where the same gesture can differ in shape and duration.

Our system uses OpenCV image processing library to perform the complete process from pre-processing to detection & tracking to feature extraction and finally training and testing by HMM algorithm. OpenCV based library makes the system easy to create due to the large amount of inbuilt functions of various image processing tasks like edge detection, feature tracking etc., also being a C++ based library the systems compatibility for real time applications is quite high with a fast processing speed.

In this paper, our system has been described in three stages: hand detection & tracking of the gesture, feature extraction from the tracked down image of hand and finally training & recognition of gestures.



II. DETECTION & TRACKING

In our system, the detection process i.e. hand localization and extraction is done using a combination of two methods:

A) Thresholding- To extract the moving object region, we can apply the thresholding on the frame difference to extract the possible moving region in complex background as per eq1 [4].

$$D_i(x,y) = \begin{cases} 1, & |F_i(x,y) - F_{i+1}(x,y)| \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

B) Skin Colour Segmentation- Skin can be easily detected using the colour constraint. Usually a skin colour is detected using HSV colour space but we got to see that it suffers through a lot of noise (e.g.: Fig 2) especially for a complex background. So, we decided to use Lαβ colour space because as we know that our skin colour mainly comprises a ratio of red and yellow colour and in case of Lαβ colour space, the α component represents the pixel components position between red and green while the β component represents between yellow and blue making it less vulnerable to noise (e.g.: Fig 3) making it most useful colour space for skin colour segmentation over HSV.

Using the above two methods we were able to localize and extract hand movement information. Then we used the 'AND' logic to combine the two extracted information together.

$$C_i(x,y) = D_i(x,y) \text{ and } E_i(x,y) \quad (2)$$

Here, $D_i(x, y)$ is image output after thresholding and $E_i(x, y)$ is the image output of skin colour segmentation.

The detection process is followed by the tracking of the extracted object i.e. the hand. For tracking we decided to use the CAMSHIFT (Continuously Adapted Mean Shift) algorithm. The mean shift algorithm operates on probability distributions. To track coloured objects in video frame sequences, the colour image data has to be represented as a probability distribution by using colour histograms. But in the case of video image sequences the colour distribution histogram varies a lot over time. So the original mean shift algorithm is modified to adapt dynamically for changing colour probability distributions resulting into CAMSHIFT algorithm [5]. The CAMSHIFT algorithm is as follows:

1. Choose the initial location of the search window. (For us is the window consists the extracted object after the detection process.)
2. Compute the mean position within the window.

$$p_i(w) = \frac{1}{|w|} \sum_{j \in w} p_j(w) \quad (3)$$

Fig 2: Result of Skin colour segmentation in HSV colour space

3. Centre the search window at the computed mean position in step 2.
4. Repeat the step 2 and 3 for one or many iterations.
5. Now find out the zeroth moment from above and set the search window size equal to the function of the zeroth moment.
6. Repeat step 4 and 5 till the mean position is moves less than the pre-set threshold value.

OpenCV proves to be a great help in the application of the CAMSHIFT algorithm for hand tracking. The library provides us an inbuilt function cvCamshift() which helps us to directly apply the algorithm just by providing the function the required search window, its colour histogram and number of iterations etc. and it applies the algorithm exempting us to write down the whole CAMSHIFT code from scratch.

III. FEATURE EXTRACTION

One should know that a gesture recognition system's efficiency depends on the features extracted. Now the features extracted to be considered good should match certain criteria like 1) they should remain invariant to rotation, translation and reflection, 2) easily computable and 3) they should not be getting replicated [4]. Now for hand gesture recognition a lot of features extractable are available but we decided to choose Hu invariant moments and hand orientation features.

Hu derived a set of 7 moments which are invariant to rotation and translation [6]. The equations of the moments are as follows:

$$\varphi_1 = n_{20} + n_{02} \quad (4)$$

$$\varphi_2 = (n_{20} - n_{02})^2 + 4n_{11}^2 \quad (5)$$

$$\varphi_3 = (n_{30} - 3n_{12})^2 + (3n_{21} - n_{03})^2 \quad (6)$$

Fig 3: Result of skin colour segmentation in Lαβ colour space

$$\varphi_4 = (n_{30} + n_{12})^2 + (n_{21} + n_{03})^2 \quad (7)$$

$$\varphi_5 = (n_{30} - 3n_{12})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] + (3n_{21} - n_{03})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] \quad (8)$$

$$\varphi_6 = (n_{20} - n_{02})[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] + 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03}) \quad (9)$$

$$\varphi_7 = (3n_{12} - n_{03})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] + (3n_{12} - n_{30})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] \quad (10)$$

These extracted are then normalized so to keep them within a particular range [8].

$$\varphi_i = (\varphi_i - \min(\varphi_{all}))/(\max(\varphi_{all}) - \min(\varphi_{all})) \quad (11)$$

Here φ_i is the ith Hu moment feature while min (φ_{all}) and max (φ_{all}) is minimum and maximum Hu moment feature respectively.

OpenCV comes in quite handy in this case too, providing us an inbuilt function in the library to find out the Hu moments i.e. cvHuMoments() which gives us all the 7 feature vectors as its output.

Orientation is quite a handy feature for hand gesture recognition as it's easily computable.The orientation is determined between two consecutive centroid points when drawing gesture path so, for this the centroid of the hand is calculated [7]. The orientation θ_t is computed as eqn 12.

$$\theta_t = \arctan\left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right); \; t = 1, 2, 3 \ldots\ldots T-1 \quad (12)$$

Here T is the length of the gesture path. The computed angles are quantized in a range of 1 to 20 by dividing them by 18 degrees, generating a vector of discrete code words fed directly to the HMM model shown in fig 4.

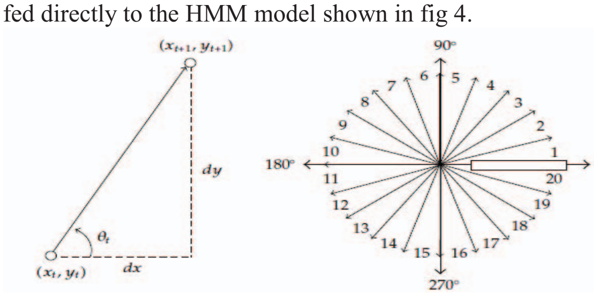


Fig 4: The orientation and its codewords

IV. TRAINING & RECOGNITION

After obtaining the features from the extraction process, the abstracted vectors are used for computing probability of each gesture type using HMM. The features obtained are described as vectors and supplied input to the HMM model constructed for each gesture.

An HMM can be considered as the simplest dynamic Bayesian network i.e., a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (*hidden*) states. The mathematics behind the HMM was developed by L. E. Baum and co-workers. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters.

In a *hidden* Markov model, the state is not directly visible, but output, dependent on the state, is visible. The adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model [8].

A discrete HMM's parameter set λ is represented by one vector π and two matrices A and B. HMM has three fundamental problems: evaluation, decoding and training.

1. Evaluation: The problem is to calculate an output observable symbol sequence or vector O given an HMM parameter set λ. The following problem is solved by using Forward-Backward algorithm [9].
2. Decoding: The problem is to determine an optimal state sequence which is associated with the given observable symbol sequence or vector O by a given HMM parameter set λ. The following problem is solved by using Viterbi algorithm [10].
3. Training: The problem is to maximize the output probability of generating an observable symbol sequence or vector. The following problem is solved by using Baum-Welch algorithm [9].

HMM has three topologies Fully Connected (i.e. Ergodic model) where any state can be reached LR such that each state can go back to itself or to the following states and LRB model in which each state can go back to itself or the next state only. We choose left-right banded model (fig.) as the HMM topology, because it is good for modelling-order constrained time-series and its properties also change over time in sequence [7] and the number of states are decided on the basis of complexity of a gesture.

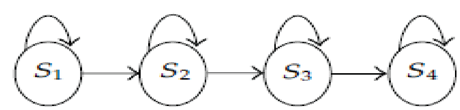


Fig.5 LRB Topology

After finishing the training process by computing the HMM parameters for each type of gesture, a given gesture is recognized corresponding to the maximal likelihood from a given gesture set [11]. The isolated and continuous gestures are recognized by its discrete vector and HMM Forward algorithm (fig.6)

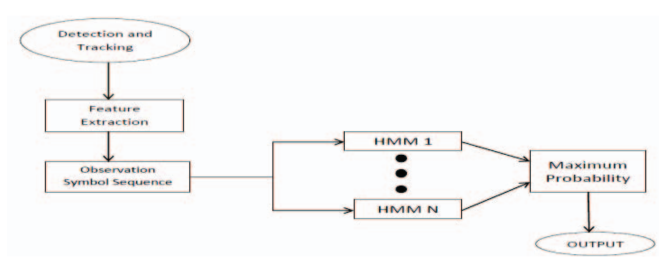


Fig.6 Recognition Process

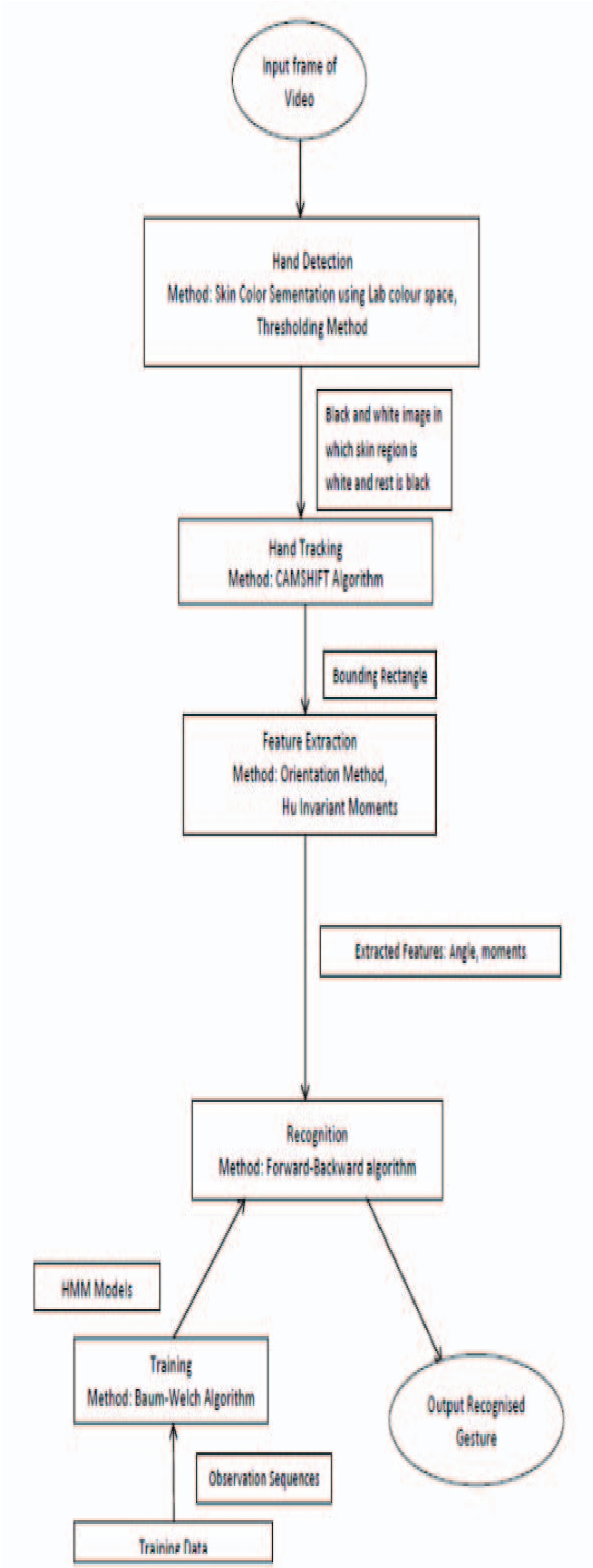


Fig.7 Flow chart describing complete hand recognition system

V. RESULTS

Our proposed system showed good results to recognize the real time gestures. The system was implemeted in C++ using Visual compiler. We had total 5 geatures and 10 training videos for each gesture. We got a recognition rate of 94.33% with each gesture model was defined by LRB topology with 5 states.

VI. CONCLUSION

This paper proposes an automatic recognition system that can recognize isolated gestures.The proposed system uses Hu invariat moments and hand motion tracjectory i.e. hand orientations as features which are fast and easily calculatable which is an advantage since using OpenCV. Our system is more focussed on the speed of recognition as compared to better recognition rate. Further experiments will be focussed on increasing the recognition rate of our system by improving the tracking techniques as it suffers very much from noise.

ACKNOWLEDGEMENT

I would like to acknowledge Dr. Anupam Agrawal, Profeesor at IIIT Allahabad for providing me this idea to work upon. I also acknowlede Siddharth Rautaray for his valuable ideas during project completion.

REFERENCES

[1] Elmezain M., Al-Hamadi A., Appenrodt J., Michaelis B., "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory," *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, vol., no., pp.1-4, 8-11 Dec. 2008. [2] T. Starner, A. Pentland, Visual recognition of American sign language using hidden Markov model, Proceedings of the International Workshop on Automatic Face-and Gesture-Recognition, Zurich, Switzerland, June 1995,pp.189-194.

[3] J. Yang, Y. Xu, C.S. Chen, Human action learning via hidden markov model, IEEE Trans. Systems, Man, Cybernet. 27 (1) (1997) 34-44.

[4] Feng-Sheng Chen, Chih-Ming Fu, Chung-Lin Huang, Hand gesture recognition using a real-time tracking method and hidden Markov models, Image and Vision Computing, Volume 21, Issue 8, 1 August 2003, Pages 745-758.

[5] Sulochana M. Nadgeri, S. D. Sawarkar, and A. D. Gawande. 2010. Hand Gesture Recognition Using CAMSHIFT Algorithm. In *Proceedings of the 2010 3rd International Conference on Emerging Trends in Engineering and Technology* (ICETET '10).

[6] M. Hu. “Visual Pattern Recognition by Moment Invariants,” in IRE Transaction on Information Theory, Vol. 8, No. 2, pp. 179-187, 1962.

[7] Xiaoyan Wang, Ming Xia, Huiwen Cai, Yong Gao, and Carlo Cattani, “Hidden-Markov-Models-Based Dynamic Hand Gesture Recognition,” Mathematical Problems in Engineering, vol. 2012

[8] A tutorial on hidden markov models and selected applications in speech recognition (1989) by Lawrence R. Rabiner.

[9] G.D. Forney, The viterbi algorithm, Proceedings of the IEEE 48 (1973) 268–278.

[10] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of The IEEE 77 (2) (1989) 257–285.

[11] Aditya Ramamoorthy, Namrata Vaswani, Santanu Chaudhury, Subhashis Banerjee, Recognition of dynamic hand gestures, Pattern Recognition, Volume 36, Issue 9, September 2003, Pages 2069-2081