Background and Skin Colour Independent Hand Region Extraction and Static Gesture Recognition

Prakhar Mohan, Shreya Srivastava and Garvita Tiwari
Department of Electronics and Communication
Indian Institute of Information Technology Allahabad
Allahabad, India
{prakharmohan699, shreya94srivastava, garvita.tiwari9}@gmail.com

Citation: P. Mohan, S. Srivastava, G. Tiwari, R. Kala (2015) Background and skin colour independent hand region extraction and static gesture recognition. In Proceedings of the 2015 Eighth International Conference on Contemporary Computing, Noida, India, pp.144-149.

Final Version available at: http://ieeexplore.ieee.org/abstract/document/7346669/

***Abstract*—Hand extraction and gesture recognition has always been a challenging problem in its general form. In this paper, we consider a fixed set of standard gestures and a reasonably structured environment and develop three effective procedures for extracting hand from the image, two of which are for plain non-complex static background and one for complex static background making it independent of the skin and background colours. The second part is of recognizing the gesture and making it scale invariant. For hand extraction, the three basic concepts used are 1. Gaussian distribution, 2. K-Mean classification and 3. Simple background subtraction and consecutive frame subtraction to find the palm region in the complete image. In gesture recognition, we extracted some features like centre of hand region, no. of fingers and the distance between the fingers. Using these features, the gestures are classified into seven standard hand gestures.**

Keywords—*Gaussian distribution; K-Mean classification; Background Subtraction, Sequential frames subtraction, Gesture Recognition; Computer Vision; Hand extraction*

I. INTRODUCTION

Object recognition has always been a hot area of research and still has a lot of potential for advancements in technologies. From an eclectic store of recognisable things, Gesture Recognition is gaining importance as it finds its usage in a large number of applications. But before coming on to gesture recognition, extracting hand (whose gesture is to be recognized) from images with complex backgrounds has also been a tough task and needs a lot of technical advancements. Often hand gesture recognition uses image processing to extract hand region from the image. Most of the techniques rely on recognition through markers (e.g. using gloves) or extracting hand using colour. Thus extraction becomes very much circumscribed by the colour of skin, clothes and background.

Detection of hand region in successive image frames taken from a video captured from a camera uses several types of features like skin colour [1], [3], [5], motion of hand, simple thresholding [1] etc.. Simple thresholding can be used for simple static background but such situation hardly occurs in real time scenarios.

Colour based techniques generally include training using skin pixel data in various colour spaces or using look up tables or simply setting the threshold levels for different colour planes[5]. Some techniques use background subtraction [1], [2] along with colour based segmentation [3] to detect the hand region. Methods like clustering have also been used for this purpose [4]. The method initially locates k clusters in the image and then every pixel is classified to their nearest cluster. This method has low time complexity but false detection increases. The main drawbacks with colour based segmentation is that the skin colour varies from person to person. Also, if the background has some components similar to the skin colour of the user, the method results in over-segmentation. Our proposed method for hand extraction is independent of the skin and background colours.

In various previous works, gesture recognition is done using techniques like principal component analysis [6], neural network [7], Fourier descriptors [8], Hidden Markov Model [9]. In neural networks approach, 3D Euclidean binary space is created from binary frames of captured video and is fed to *feed forward neural network*. PCA is used to determine Eigen space to extract features and then data is fed to neural network. These techniques are avoided as they require large dataset for training, produce delay and have less accuracy, hence real-time applications becomes difficult to implemented efficiently.

Geometrical features based on shape representation can be used to classify gestures. These can be contour based like Perimeter, Shape Signature, Wavelet Descriptor etc. or region based shape properties like Convex Hull, Media Axis, Area, Euler Number, Moments etc. Centre of gravity (COG) and extreme points are obtained and hence number of fingers in hand gesture are obtained [5], [10]. Other geometrical feature like convex hull and convexity defects are used to find hand gestures [11]. In our model, the features: number of fingers,

distance between adjacent fingertips and Centroid are used for classification. The method is fast, computationally less complex and requires no dataset storage.

II. PROPOSED METHODOLOGY

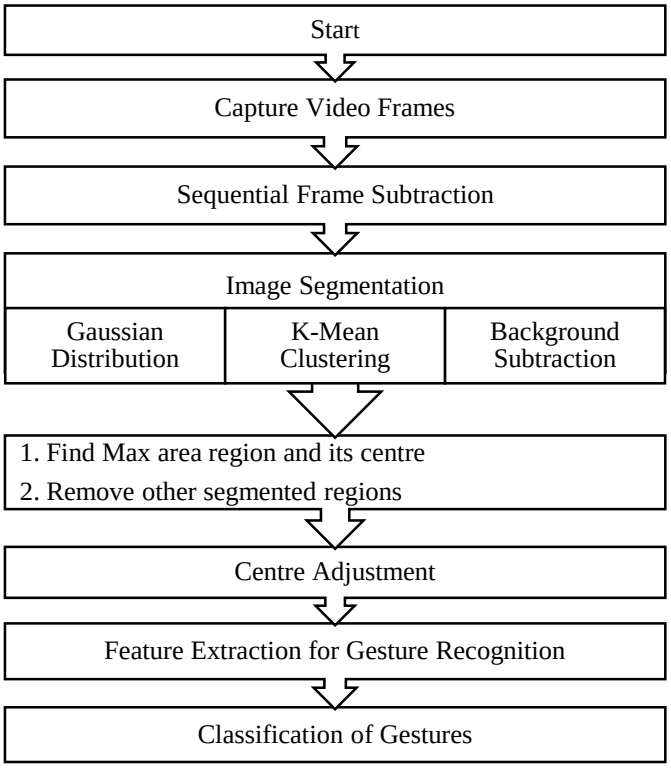

Fig. 2. Process flow of proposed methodology

The process flow of the proposed methodology is shown in Fig. 1. The frames or images from which the hand gesture is to be recognised are captured continuously using a camera. These frames are pre-processed in which the RGB frame is either converted to grayscale or YCbCr colour space depending on the image segmentation method used. After pre-processing, hand region is extracted using sequential frame subtraction and one of the three methods of image segmentation i.e. Gaussian Distribution, K-Mean Clustering and Background Subtraction. Once the hand region is obtained, features for gesture recognition are extracted and accordingly the gestures are classified. These features include the centre of hand, number of fingers and the normalised distance between adjacent finger tips.

Unlike many other developed techniques, our method of extraction and recognition is independent of skin and background colour and to obviate these restrictions we need to impose certain constraints on the environment or the background. These constraints are:

a) The background should be strictly static i.e. there should be no movement in the background.
b) The hand while making gestures should not be strictly static i.e. slight movements in hand is necessary.
c) The background colour should be different from the skin colour but their exact colours are not necessary.

Fig. 1. a) The background frame F(0), b) A regular frame F(n), c) The previous regular frame F(n-1), d) Final result of background subtraction, e) Final result of sequential frames subtraction.

III. HAND REGION EXTRACTION

A). Image Subtraction

For background subtraction, the background image is found by averaging the first five frames captured by the camera. This averaged background images is referred to as F(0). No gestures are made in these frames. The frames captured from now on are referred to as regular frames denoted as F(n) and contains the gestures to be recognised. These are subtracted from F(0) as:

$$I_{sub1} = \text{threshold}\,[\text{abs}\,\{F(n) - F(0)\}] \quad (1)$$

Here, the two frames are subtracted and thresholded on the basis of the absolute subtracted value. The thresholding is binary in nature i.e. all the pixels with values above the threshold are represented as a white pixel and the one lower than the threshold are represented as a black pixel. After thersholding, the binary images are made free of noise by using morphological dilation and erosion [13] and is referred to as I_{sub1}.

In sequential frame subtraction, the consecutive frames F(n) and F(n-1) are subtracted as given by (2).

$$I_{sub2} = \text{threshold}\,[\text{abs}\,\{F(n) - F(n\text{-}1)\}] \quad (2)$$

Here, the operations are same as those of background subtraction explained above. The final result of sequential subtraction obtained after subtracting, thresholding and noise removal is referred to as I_{sub2}. The results for the two subtraction techniques are shown in Fig. 2.

From this figure it can be noted that the background subtraction gives a sure hand region along with many other unwanted regions whereas the sequential frame subtraction gives a rough idea where the hand is in the frame. Thus rather than processing the complete image, the region obtained in I_{sub2}

Fig. 3. a) Bounding box drawn on I_{sub2}, b) Mask P_{mask} obtained using eq.3, c) Masked grayscale image of frame shown in Fig. 2b.

has to be processed. This region is estimated by a bounding box drawn as given in (3).

$$P_{mask}(x, y) = \begin{cases} 255 & x \in [x_1, x_2], y \in [y_1, y_2] \\ 0 & otherwise \end{cases} \quad (3)$$

where (x_1, y_1) and (x_2, y_2) are diagonal vertices of the bounding box. The resulting bounding box is shown in Fig. 3. The images (c) shows the masked portions of the frame shown in Fig. 2b. On these masked images we apply image segmentation techniques of Gaussian Distribution or K-Mean Clustering.

B). Image Segmentation

For segmentation, the three kind of scenarios to be dealt with are: 1). Hand region is lighter than immediate background, 2). Hand region is darker than immediate background and 3). Immediate background is made of two components, one darker than hand region while other lighter than the hand region.

1). Gaussian distribution: Gaussian distribution or Normal distribution is a common continuous probability distribution. Assuming our masked images to follow Gaussian distribution, a frame is segmented into hand and non-hand region by using the mean ($\boldsymbol{\mu}$) of all pixel values in its masked image and their standard deviation ($\boldsymbol{\sigma}$). For Gaussian Method, the three scenarios are shown in fig. 4. From these figures, it is easy to analyse that the mean pixel value behaves as a threshold to separate the hand and background region. If **p** be the pixel value of a particular pixel, then the hand region for different scenarios are given as:

$$I_{gauss} = \begin{cases} 0 < (p - \mu) < 2 * \sigma & scenario\ 1 \\ 0 > (p - \mu) > -2 * \sigma & scenario\ 2 \\ -\sigma < (p - \mu) < \sigma & scenario\ 3 \end{cases} \quad (4)$$

We see that the hand region obtained for three scenarios have different conditions. For scenario 1, the hand pixel values are larger than the background pixel values, in scenario 2, the hand pixel values are smaller than the background pixel values and in scenario 3 the hand pixel values lies between the values of the darker and the lighter portions of the background. Hence making it work for all the scenarios together becomes very difficult. Also depending on the number of pixels in hand and non-hand region, the mean value ($\boldsymbol{\mu}$) gets biased and the segmentation becomes less efficient.

2). K-Mean Clustering: K-Mean clustering is a common technique used for image segmentation. The region is divided into k clusters and the pixels are then classified to the cluster nearest to them. In our approach, we classify the image into two clusters. The masked portion of the frames (as shown in fig. 3 and 4) are used to train and find the means of these clusters. The two mean values referred to as $\boldsymbol{\mu_0}$ and $\boldsymbol{\mu_1}$ are initially assigned the lowest and the highest pixel value from the masked portion. The pixels are then classified according to their distances from μ_0 and μ_1. These mean values are then recalculated by averaging their respective pixels and again the pixels are classified. This is done till the two mean values converge. One of the two values thus obtained correspond to hand region and other correspond to the background region.

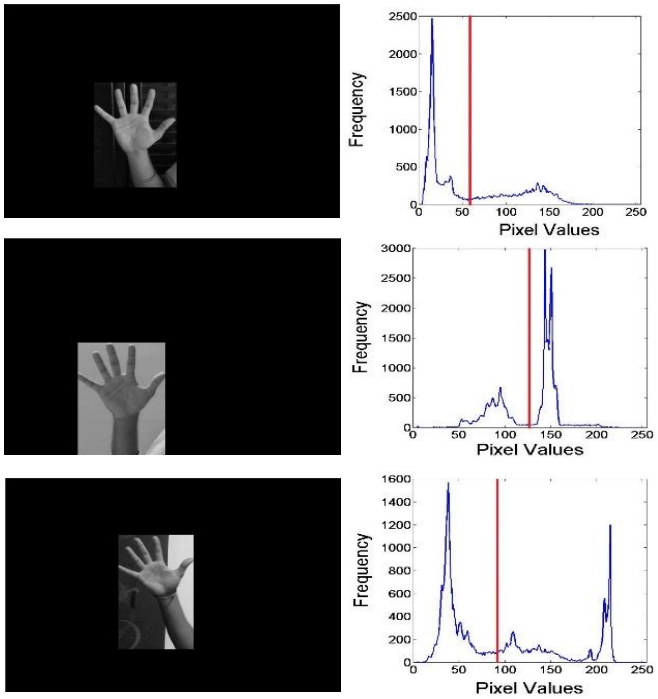

Fig. 4. Masked images for three scenarios and their pixel analysis. (a, b) Scenario 1 with μ = 59.16 and σ = 52.43, (c, d) Scenario 2 with μ = 126.67 and σ = 32.62, (e, f) Scenario 3 with μ = 91.9 and σ = 65.82.

The cluster corresponding to the hand region can be found by using a sure hand pixel. Finding this sure hand pixel is a challenging task. Assuming that the majority portion of the masked image is our hand region, it can be said that the centre of this rectangular masked region will fall on the hand region. The cluster closer to this pixel value becomes the hand region and is represented by a white pixel. The method is not very stable because this centre may lie outside the hand region because of disturbances or if the hand region is small due to larger distance from the camera.

3). Background Subtraction and Removing of Over-Segmented Regions: Both the Gaussian method and the K-Means Clustering have some limitations and fail to give results in at least one of the three scenarios stated above. We now propose a very simple and analytical method to do the same task that works well for all the three scenarios. Consider Fig. 5, left columns shows the I_{sub1} images for some set of frames, middle column shows the I_{sub2} images bounded by a bounding box and right column shows the masked portion of I_{sub1} generated using the bounding box as explained earlier along with centre of the max area region drawn on it.

Masking the I_{sub1} this way removes many of the over-segmented regions. By analysing many of such masked images, it was found that the hand is a part of the region with maximum area and the other regions can be removed leaving behind only the region with the hand. This can be seen in Fig 5(c and i). But it may happen that for some frames the region having the hand is not the one with maximum area. This condition is shown in

Fig. 5. (a, d, g) I_{sub1} for 3 regular frames, (b, e, h) I_{sub2} for respective frames along with the bounding box. (c, f, i) Masked portion of I_{sub1}.

Fig. 5(f). Since the frames are captured continuously, if for a frame the max area region found was correct and the centre was calculated but in the next frame the max area region changes, there would be a sudden change in its centre. This abrupt change can be easily detected using a threshold distance and such frames can be obviated. Doing this, the hand region is successfully extracted from a complex frame following the constraint stated above.

C). Centre Adjustment

Having extracted the hand region by using one of the above methods, its centre has to be at the proper position as it is one of the key features used for recognition. The extracted region may or may not have the arm portion (shown in fig 6). If the arm portion is extracted along with the hand (shown in fig 6(c)), the calculated centre is located far below than that of the hand. This is an undesirable situation and the centre needs to be adjusted. For this, the white pixel at position (x_1, y_1) above the centre (x, y) and having the maximum distance from it is calculated. If this distance (**d**) is more than a threshold (**D**), the centre is adjusted as:

$$y' = y - (d - D)\frac{(y - y1)}{d} \quad (5a)$$

$$x' = x - (d - D)\frac{(x - x1)}{d} \quad (5b)$$

where (**x'** , **y'**) is the new centre's coordinates. This is repeated till all the white pixels above the centre lie within the radius of **D** from this centre.

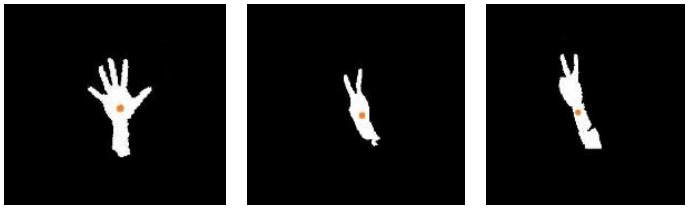

Fig. 6. (a and b) Hand region with no or very little arm portion. (c) Hand region along with a long arm portion.

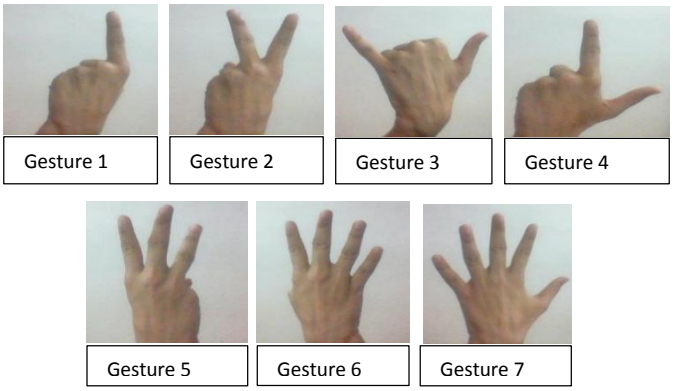

Fig. 7. Seven standard gestures our model recognises.

IV. GESTURE RECOGNITION

Our model recognizes 7 standard gestures as described in Fig. 7. These gestures are classified using features like centre of hand, no. of fingers and distance between adjacent fingers.

A). Feature Extraction

Following features have been used for gesture recognition:

1*). Centroid:* Centroid is the centre of gravity of image, which is calculated using image moments. A moment is a specific quantitative measure of the shape of a set of points. Moment of order (p + q) as defined in [12] is

$$M_{pq} = \sum_0^{m-1} \sum_0^{n-1} x^p y^q f(x, y) \quad (6)$$

where m and n are binary image dimensions, f(x, y) for binary image is either 1 or 0. Centroid or centre of hand is given by the first order moments i.e. $\{x, y\} = \{M_{10}/M_{00}, M_{01}/M_{00}\}$, where x and y are the coordinates of the centroid.

2). Number of Fingers: In our proposed method to find the number of fingers, the first step is to remove the portion below the centroid. To do this, a horizontal cut is made through the centroid. If this cut intersects the hand region at exactly two places as shown in Fig. 8, the area below the centroid can be removed immediately. In case it intersects the hand at more than two places as shown in Fig. 9, the cut is shifted few pixels down from its current position. This is repeated till the cut intersects the hand at exactly two positions after which the region below the cut can be removed.

A circle with centroid as its centre is now drawn such that it intersects only the fingers [10], [11] (shown in Fig. 10). The radius of this circle is obtained as:

$$R = 0.65 \times ((x_f - c_x)^2 + (y_f - c_y)^2)^{1/2} \quad (7)$$

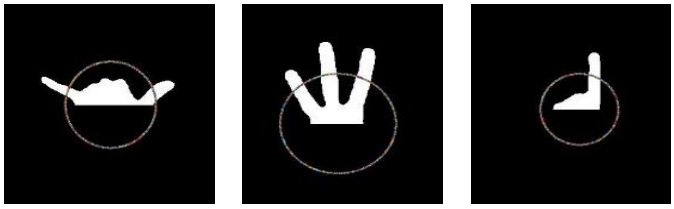

Fig. 10. Circles intersect hand only at fingers.

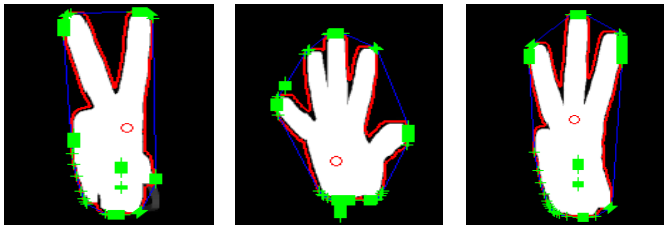

Fig. 11. Polygon fitting around hand region using convex hull

here R is the radius, (X_f , Y_f) is the farthest white pixel above the centroid, (C_x , C_y) is the centroid. The constant 0.65 was found by manually analysing a large number of gestures. Half of the number of these intersections gives the number of fingers.

3). Separation between finger tips: A polygon is fitted around the hand region using convex hull [11]. A convex hull of a set S is the smallest convex set of continuous points that encloses the complete set S. Polygon fitting through convex hull is shown in Fig. 11. This polygon fitting gives more vertices than the actual because of noise in the segmented image. The correct vertices are estimated by setting a minimum distance between two adjacent vertices. Vertices closer than this minimum distance are considered to be same and are ignored. The vertices which lie above the centroid are considered as finger tips. The distance between these adjacent vertices when normalized by the width of the extracted hand is used for classifying gestures with same number of fingers i.e. gesture 2, 3 and 4 shown in Fig. 7.

Features like Differential Angle and Polygon Area have also been used for classification in other works [5]. In Differential angle, lines joining two adjacent vertices with the centroid form an angle. Measure of this angle forms a basis for classification. Similarly in polygon area technique, the area of triangle formed by joining the two fingertip vertices and the centroid is used for classification. But these techniques are not very efficient and often results in misclassification.

B). Classification

There are five main classes defined for classification of seven gestures listed in table 1. These classes are demarcated only on the basis of number of fingers. But class 2 has three different gestures which cannot be differentiated by number of fingers. For them features like separation between finger tips, differential angle and the polygon area were tested. From table 2, it can be seen that the best results were provided by the normalised separation between finger tips. Differential angle and polygon area, as can be seen from the table, are not able to classify the gestures alone. When used together, will create a further two tier model, which will increase computational complexity. So, the feature to be used is the normalised separation between fingertips.

TABLE 1. FIVE MAIN CLASSES FOR SEVEN DIFFERENT GESTURES.

Class	Number of Fingers	Gesture Number
Class 1	1	1
Class 2	2	2, 3, 4
Class 3	3	5
Class 4	4	6
Class 5	5	7

V. RESULTS and DISCUSSIONS

Above we discussed three methods for extracting hand region from the image. Here we discuss their efficiency. The method to be used should work for all the three scenarios defined above. When using Gaussian distribution method, the mean pixel value acts as a threshold. For each scenario, the hand region is given by eq. 4. We see that for each scenario the condition for hand region is different. Also, the data forms a two or more modal distribution which leads to biasness of the mean value. This problem is more dominant in scenario 3 where the hand region lies in between the two background components. Thus when using Gaussian Method, it becomes difficult to realise the scenario and extract the region accordingly.

The K-Mean clustering classifies a pixel into one of the two classes, 1 or white for pixels similar to hand region and 0 or black for pixels different from hand region. When the number of classes or clusters are two, the method works well for

TABLE 2. FEATURE THRESHOLDS FOR CLASSIFYING CLASS 2 GESTURES I.E. GESTURE NO. 2, 3 AND 4.

Class 2 Gestures	Differential Angle	Normalized Separation between Fingertips	Polygon Area
Gesture 2	< 1 rad	< 0.4	Random range
Gesture 3	~ 4 rad	> 0.6	$> 4.5x10^{04}$
Gesture 4	~ 4 rad	> 0.4 & < 0.6	$> 2x10^{04}$ & $<4x10^{04}$

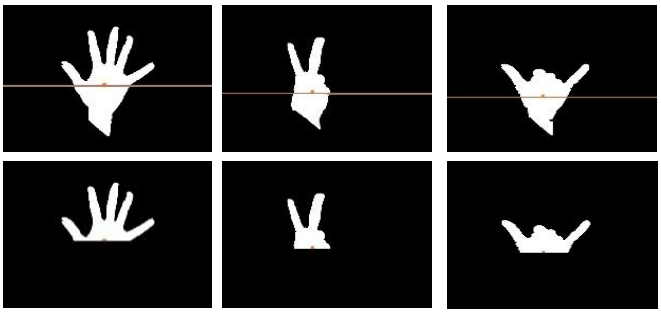

Fig. 8. Steps to remove the region below the centroid

Fig. 9. Adjustment of horizontal cut to remove the unwanted region

Fig. 12. Segmentation results of Gaussian Distribution method (2nd row), K-Mean Clustering method (3rd row) and Background Subtraction method (4th row).

scenarios 1 and 2 but fails for scenario 3. This is because while initializing the two means, $\mathbf{mu_0}$ is assigned the lowest pixel value which fall under the darker region and $\mathbf{mu_1}$ is assigned the highest pixel value which fall under the lighter region. So, on converging, the two means gets accumulated somewhere at the interface of the hand region with the other two regions. Thus while classifying, the hand region gets classified into both the classes and the efficiency falls drastically. One solution to this is using more than 2 clusters but on doing so, the initialization of cluster means becomes difficult and sometimes leads to convergence problems.

The background subtraction technique works fine with all the three scenarios. Also this technique is faster than the Gaussian distribution and the K-Mean Clustering techniques. Results for the image segmentation are shown in Fig. 12. First row shows the original RGB frames of three scenarios. Second row is the segmentation output of Gauss distribution method using the condition of the first scenario. Third row is the segmentation results of K-Mean clustering and the forth row shows the results of background subtraction. Further hand region is identified using the masking technique and finding the centre as explained above. This figure very well explains the limitations of Gaussian method and K-mean clustering in one or more scenarios while the background subtraction technique works well for all the three scenarios.

TABLE 3. RESULTS SHOWING THE NUMBER OF GESTURES RECOGNISED SUCCESSFULLY AND THEIR RECOGNITION RATES.

Gesture Number	Number of test images	Successful Recognition	Recognition Rate
Gesture 1	36	36	100 %
Gesture 2	36	33	91.66 %
Gesture 3	36	29	80.55 %
Gesture 4	36	33	91.66 %
Gesture 5	36	34	94.44 %
Gesture 6	36	35	97.22%
Gesture 7	36	35	97.22 %
Overall	252	235	93.2%

For gesture recognition, the proposed model is applied on a sequence of frames consisting of different gestures generated by the background subtraction method. Table 3 shows the results of experiment performed on these frames. Recognition rate was obtained by finding number of gestures in each class recognised correctly. Out of 252 hand gesture tested, this model was able to correctly recognised 235 of them giving a success rate of **93.2%.** In our methodology, the recognition is scale invariant and does not depend on the distance of the user with the camera. Still a large distance degrades the efficiency because for larger distance, the hand region in the image is very small and is not extracted properly. The method is also rotation invariant and works well as long as the fingers are clearly segmented. This is because each class is demarcated by the number of fingers in the gesture and class 2 gestures are further classified by the normalised separation between the adjacent fingertips.

VI. CONCLUSION

The method we presented in this paper works well as long as the constraints are followed strictly. The extraction of rand region is made easier more efficient and less computational expensive but it has some limitations too. The background should be strictly static. Though it is flexible to small head movements or slight momentarily disturbances, continuously moving objects like fans, tree leaves etc. causes the method to malfunction. Also the background immediately behind the hand should be different in colour though the exact colours are not important.

The method proposed for recognition is both scale and rotation invariant. The recognition is not effected by the distance from the camera but for too large distances, the hand portion in the image is very small and is not extracted properly. Also the recognition is efficient as long as the fingers are extracted properly. The rotation of hand is allowed up to the point where the fingers captured by a single camera do not overlap. Gestures with improper finger spacing reduces the efficiency of the method. This is the reason for low recognition efficiency for gestures with same number of fingers. Here, only a small gesture set is considered but the method can be extended to recognise a large set of gestures.

REFERENCES

[1] Randive, A. A., H. B. Mali, and S. D. Lokhande. "Hand Gesture Segmentation." International Journal of Computer Technology and Electronics Engineering, Vol. 2, No. 5, pp. 125-129, (2012).

[2] H. Khaled, S. Sayed, E. S. Mostafa, H. Ali, "Hand Gesture Recognition Using Average Background and Logical Heuristic Equations ", International Journal of Computers and Technology, Vol. 11, No. 5, pp. 2634-2640, 2013.

[3] S. N. Krishna, V Lathasree, "Fusion of Skin Color Detection and Background Subtraction for Hand Gesture Segmentation", International Journal of Engineering Research & Technology, Vol. 3, No. -2, pp. 2278-0181, 2014.

[4] M Panwar, P S Mehra. "Hand Gesture Recognition for Human Computer Interaction", Proceedings of the IEEE International Conference on Image Information Processing (ICIIP 2011), Waknaghat, India, November 2011, pp. 1-7.

[5] Gaurav Modanwal, Satish K. Singh, "Writing Support System For Blinds Using Gesture Recognition", a Thesis of Master of Technology in Electronics Engineering, Indian Institute of Information Technology Allahabad, July-2014.

[6] Trong-Nguyen Nguyen, Huu-Hung Huynh, and Jean Meunier, "Static Hand Gesture Recognition Using Principal Component Analysis Combined with Artificial Neural Network," Jounal of Automation and Control Engineering, Vol. 3, No. 1, pp. 40-45, February, 2015.

[7] Murthy, G.R.S.; Jadon, R.S., "Hand gesture recognition using neural networks," Advance Computing Conference (IACC), 2010 IEEE 2nd International, vol., no., pp.134, 138, 19-20 Feb. 2010

[8] Gamal, H.M.; Abdul-Kader, H.M.; Sallam, E.A., "Hand gesture recognition using fourier descriptors," Computer Engineering & Systems (ICCES), 2013 8th International Conference on , vol., no., pp.274,279, 26-28 Nov. 2013

[9] Vieriu, R.-L.; Goras, B.; Goras, L., "On HMM static hand gesture recognition," Signals, Circuits and Systems (ISSCS), 2011 10th International Symposium on , vol., no., pp.1,4, June 30 2011-July 1 2011.

[10] Malima, Asanterabi, Erol Ozgur, and Müjdat Çetin. "A fast algorithm for vision-based hand gesture recognition for robot control." Signal Processing and Communications Applications, 2006 IEEE 14th. pp. 1-4, IEEE, 2006.

[11] Nayana P B and Sanjeev Kubakadd. "Implementation of Hand Gesture Recognition Technique for HCI Using Open CV ". International Journal of Recent Development in Engineering and Technology SSN 2347 -6435 (Online) Volume 2, Issue 5, pp. 17-21, May 2014

[12] Palaniappan, R., P. Raveendran, and Sigeru Omatu. "Improved Moment Invariants for Invariant Image Representation." Invariants for Pattern recognition and Classification' (World Scientific Publishing Co., Singapore, 2000): 167-187.

[13] Jamil, N.; Tengku Mohd Tengku Sembok; Bakar, Z.A., "Noise removal and enhancement of binary images using morphological operations," Information Technology, 2008. ITSim 2008. International Symposium on , vol.4, no., pp.1,6, 26-28 Aug. 2008.