Open Access
Full Length Article

ORIGINAL ARTICLE

TRACKING NON RIGID OBJECT LIKE HUMAN HAND WITH CLUTTERED BACKGROUND USING CASCADE OBJECT DETECTOR

Faisal Jamal Nasir[1], Muhammad Naeem[2], Asma Gul[3], Nasir Ahmad[1]

[1]Department of Computer Systems Engineering, University of Engineering & Technology Peshawar, Pakistan
[2]Department of Computer Science, University of Peshawar, Pakistan
[3]Department of Statistics, Shaheed Benazir Bhutto Women University, Peshawar, Pakistan

ABSTRACT

This paper presents an algorithm for initialization and tracking of hand in a video taken by an ordinary color camera. Hand detection is carried out by training the Viola-Jones classifier using a histogram of gradient features. The detected hand instances contain false positives which are further filtered using skin color based segmentation. For skin color detection, theTint, Saturation, and Lightening (TSL) color space has been used, which is designed specifically for the skin color detection. As skin color is a perceptual and subjective human concept and not a physical property, the classifier has been trained on the skin and non-skin data with an AdaBoost algorithm, which provides the distinction between skin and non-skin input. A tracking algorithm is then used to track the identified hand in successive frames of video.

Keywords: Skin Color Detection, Hand Detection, Hand Tracking, Cascade Detector

Author`s Contribution
[1] Data analysis, interpretation and manuscript writing, Active participation in data collection
[2] Conception, synthesis, planning of research and manuscript writing
[3] Interpretation and discussion

Address of Correspondence
Nasir Ahmad
Email: n.ahmad@uetpeshawar.edu.pk

Article info.
Received: March 03, 2017
Accepted: June 19, 2017

Cite this article: *Nasir Jamal F, Naeem M, Gul A, Ahmad N. Tracking Non-Rigid Object Like Human Hand With Cluttered Background Using Cascade Object Detector. J. inf. commun. technol. robot. appl.2017; 8(1):44-51.*

Funding Source: *Nil*
Conflict of Interest: *Nil*

INTRODUCTION

Considerable interest has been shown in Natural Human-Computer Interaction (HCI) in HCI research recently. In particular, communication with the computer system through hand-gestures and other natural modes of interaction has received a good degree of attention especially in the area of computer vision and pattern recognition. Literature in this respect abound. Here a review of selected work is presented. Ghotkar and Kharate[1] proposed a vision-based system for hand tracking. Their approach is based on simple image processing techniques and the camshaft algorithm and uses the Hue, Saturation, and Value (HSV) color space to detect skin color while the edge traversal algorithm to detect hand. The algorithm showed good performance on color gloves but was found to be sensitive to lightening conditions when hands are bare. The method proposed in utilizes mid-level semantics coupled with Bayesian probability network for hand-tracking during occlusion. In the method, a segmentation of the frames of sign language videos is performed with the help of skin color and descriptors extracted from the frames. Then the Gaussian distribution model of skin color is used to detect a hand, which is followed by removal of noise through a morphological opening operation. A limitation of this approach is a reliance on face features for detecting the hand and the inability to tackle the task of detecting the hand when face information is not available. Approaches based on Hidden Markov Model have also been proposed especially for real-time hand tracking. Nguyen used Kalman applies 2D HMM to motion descriptors extracted from hand blobs. Additionally, skin color information is also utilized in the detection process. An assumption in this method is that there is a single hand in the image. Consequently, the subject has to wear shirts with long sleeves to suppress the image of the arm from the image of hand obtained. Binh et al.[3] proposed an algorithm for recognition of dynamic hand gestures. The algorithm uses Kalman filter for tracking hand and HMM for characterization of hand-gesture. Wu et al.[4] finds an estimate of both and global hand posture using a divide and conquer approach. An estimate of the motion of figures is obtained from the global pose of the hand and the motion is then tracked with the Montie Carlo technique. Fingers of the hand can be either in fully straight or fully curled state. An assumption in this method is that there are no clutters present. Furthermore, the hand-initialization process is manual and unplanned rotation is not supported. In [5], a boosted tree-based classifier approach is used to solve the problem of hand detection. Shamovie and Sutherland [6] adopted a dynamic model based on skin color and grassfire algorithm that tracks bimanual movements of hand in a real time when there is occlusion. Misra et al.[7] proposed a method based on the calculation 2D color histogram of hand-image, which is then fed to a Gaussian Mixture Model. The method doesn't show robustness to background lightning and color reflections such as skin color reflections. The approach by Bao et al.[8] addresses the problem of recognizing the Vietnamese sign language with a method based on pseudo two-dimension hidden Markov model and using skin color for tracking and recognizing the hand gesture. Freeman and Weisman [9] proposed an algorithm for controlling television with a hand gesture and Sepehri et al.[10] investigated the use of the hand as an interface device in virtual and physical spaces. Finally, the works in [11-14] address the problem of recognizing hand-gestures in various situations of human and computer communication.

In this work, we propose a novel method for recognizing hand-gestures based on self-initialization of hand position. The method provides good performance even when the background is cluttered. Moreover, the user is relieved of the requirement to wear any data glove or hold a camera for capturing hand-image. As efficient hand tracking is a necessary step for hand gesture recognition (HGR), we also suggest a technique for hand recognition that applies a cascade of week classifiers to color based segmentation with the AdaBoost algorithm.

The remaining of this paper is organized as follows. In the following section, we give a model for hand tracking. Next, color models used in different algorithms are discussed. Following that, we describe the proposed method for hand detection and report the performance on a standard dataset. Finally, we give the conclusion and describe the future work.

HAND TRACKING ALGORITHM

The hand tracking algorithm proposed in this work consists of two stages, i.e. the initialization stage and tracking stage. These stages are discussed in details in the following sub-sections.

Initialization Stage:

The initialization stage of the hand tracking algorithm is performed in these steps.

- Capturing a single frame of video using web/laptop cam.
- Apply three types of detectors on the captured frame i.e. skin detector, Viola-Jones face detector, and cascade hand detector.
- Filtering faces which are wrongly detected as hands.
- Filtering out the hands which do not comply with the hand color of AdaBoost hand detector.
- If the hand is found, the count is incremented by one. Otherwise, the count is reset.
- If the count reaches five, the program switches from initialization mode to tracking mode.
- Otherwise, keep repeating step 1 to step 3 till a hand is found.

The whole process of hand tracking is depicted Figure 1.

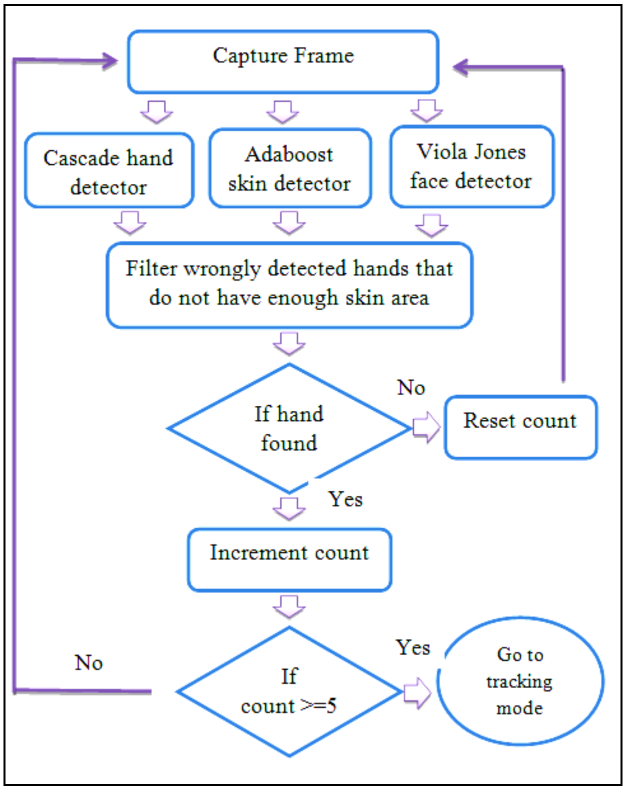

Figure 1 Block diagram for hand initialization algorithm

Tracking Stage:

The initialization stage is followed by the tracking stage. The step-by-step procedure of the tracking step is as follows.

- Set the cascade hand tracker size according to hand region found in initialization stage.
- Apply the cascade hand detector.
- If multiple hands are found filter them and select the right one on the basis of its size and position as compared to size and position of old hand region.
- Show detected hand on the screen and keep tracking new frame using step 1 to step 4.
- If cascade hand detector no hand fails to detect the hand, apply the skin detector.
- Find skin blobs in the frame. Apply erosion and dilation operations to refine blobs and remove noise.
- Filter detected hands such that if some arm region is also included with them, it is removed.
- Mark the blob nearest to old hand region and complying with its size as hand detected.
- Keep tracking hand using step 1 to step 4 again.
- If the hand is neither found by cascade detector nor by skin detection, increment misses hand count by one.
- Check miss hand count. If its value exceeds 5, switch to initialization mode.
- If miss hand count has not exceeded by 5 the mark old hand region as detected hand and go to step 1

The hand tracking process is summarized in Figure 2.

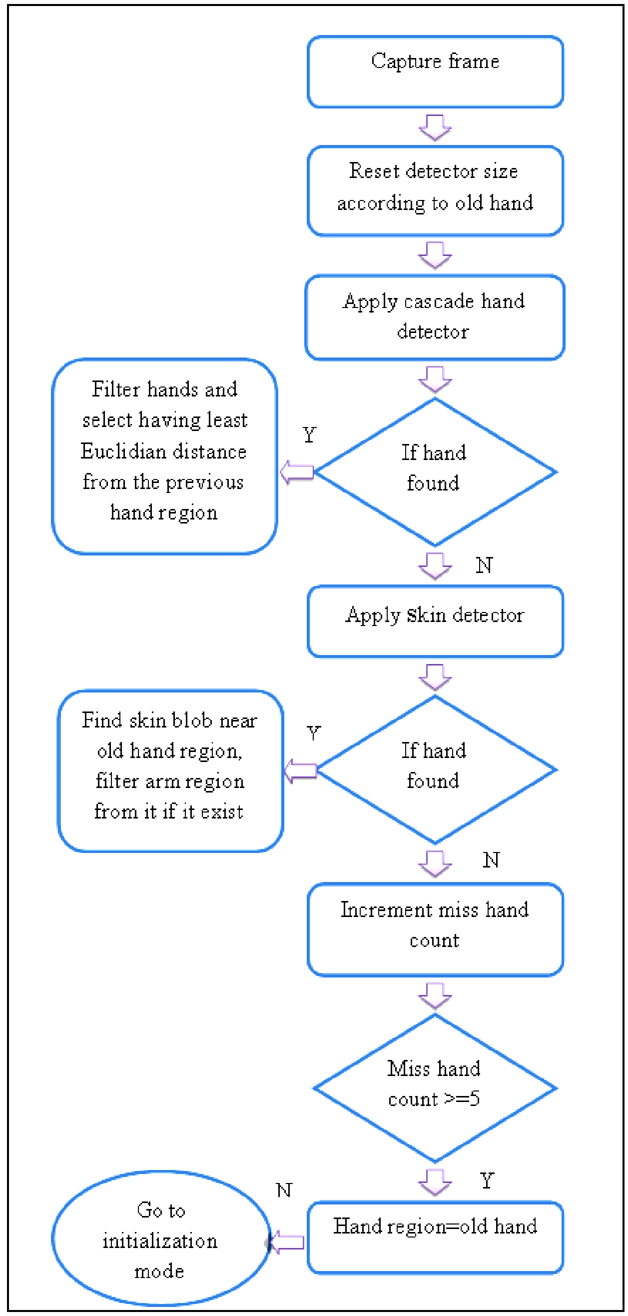

Figure 2 Block diagram for hand tracking algorithm

Our hand detection model uses dual layer detection scheme. The first hand is detected using cascade detector. But in some cases, hand detector fails to detect hand. In such case, another approach is used i.e. to detect hand using skin color. But during the initialization process hand must comply with both the skin color and hand shape detectable by hand detector.

Hand Detection Algorithm

A lot of techniques have been in practice to segment an image on the basis of skin color distribution. These techniques can model skin color using following methods.

Non-Parametric Methods – these methods determine the distribution of skin color from the training data without building an explicit model of the skin. It may be a Lookup the table, Histogram Model or Bayes Classifier.

Parametric methods – In these methods a parametric model is derived from the training set. An example of the parametric model is the Gaussian Model.

Different color models can be used to implement these techniques. A brief detail of these models is given in the following text.

Color Spaces:

In the proposed work we have tried to overcome difficulties and challenges of skin color detection. Skin color detection has been a research area of increased attention due to the trend of content-based image processing. It's important not only in bare hand detection but also gives away for face localization providing a good starting point for facial expressions studies. It would reasonable to say that estimating the area with skin color is a first vital step in most of the popular face detection algorithms.

Variability in skin color tone across different nations as well as with the change in illumination conditions makes skin color detection a very challenging task. Many algorithms using exploiting features of different color schemes have been introduced to cope with these challenges. Cai and Goshtasby[15] proposed a probabilistic color model based on CIE-Lab color scheme, for robustness against illumination. They trained a classifier on a set of skin color samples using Gaussian distribution on a* and b* values. Kurata et al.[16] used Bays model on HS color space in HSV system to minimize the illumination problems. Most of the research in skin color detection is based on RGB, YCbCr and HIS, HSV and LAB color spaces. This section briefly describes the color spaces.

RGB Color space:

The RGB color space consists of the three primary colors. These are red, green and blue. Other colors can be obtained by a combination of spectral combination of these primaries. The RGB model can be represented by a 3-dimensional cube with red, green and blue colors at the corners on each axis. The black color is at the origin while white is at the opposite end of the cube. A line from black to white can be used to represent grayscale colors. The RGB model simplifies the design of computer graphics systems but is not ideal for color-based segmentation because red, green and blue components have a high degree of correlation. This makes the execution of some image processing algorithms difficult. Many processing techniques, e.g., histogram equalization, process only an image's intensity components. A variant of this color scheme called the normalized RGB and used mostly for segmentation purposes is given by.

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B} \qquad b = \frac{B}{R+G+B}$$

YCbCr Color Space:

Basically, it is not a color space but a way of coding RGB values, given by.

$$\mathrm{Y} = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B}$$

$$\mathrm{Cr} = \mathrm{R} - \mathrm{Y}$$

$$\mathrm{Cb} = \mathrm{B} - \mathrm{Y}$$

YCbCr color space is well known for its abilities to handle digital video information. It belongs to the family of color spaces used for television transmission. Examples of such color spaces are YUV and YIQ. The difference of YCbCr with YUV and YIQ is that YUV and YIQ are analog spaces for the respective PAL and NTSC systems while YCbCr is a digital color system. In these schemes, the RGB (Red-Green-Blue) is separated into two components; the Y component, which represents the luminance information and the C component, which represents the chrominance information. They have used in compression applications although such a color specification is not very unintuitive. In the 8-bit (i.e., 0 to 255) model of YCbCr specified in the Recommendation 601, Y, which is the luminance component, has an excursion of 219 and an offset of +16. It uses the code 16 for black and code 235 for white. Thus, the signal processing foot room and bedroom in this color scheme are represented by the extremes of the range. On the contrary, the excursion and offset for the chrominance components Cb and Cr are +112 and +128 which defines a range of color values from 16 to 240.

HSI Color Space:

A color picture has basically three properties, hue saturation, and intensity and corresponding color model for that is HSI. Hue is adjusted to choose a color. To get different shades of color saturation is adjusted while intensity component is used to make an image lighter or darker.HSI is currently most prominently used color space for machine vision, where;

$$\mathrm{H} = \arccos\frac{\frac{1}{2}\left((\mathrm{R}-\mathrm{G}) + (\mathrm{R}-\mathrm{B})\right)}{\sqrt{(R-G)^2+(R-B)(G-B)}}$$

$$\mathrm{S} = 1 - 3\frac{\min(R,G,B)}{R+G+B}$$

TSL Color Space

TSL is a perceptual color space specially designed for skin color detection. T stands for Tint (Hue with white added), S is saturation (degree of colorfulness) and L is lightness (grayscale image). This color space was proposed by Jean Christophe Terrillon and ShigeraAkamatsu[17] for face detection. Conversion from RGB to TSL is straightforward, given by.

$$\mathrm{T} = \begin{cases} \frac{1}{2\pi}arctan\frac{r'}{g'}+\frac{1}{4} if g' > 0 \\ \frac{1}{2\pi}arctan\frac{r'}{g'}+\frac{3}{4} if g' < 0 \\ 0 \quad if g' = 0 \end{cases}$$

$$S = \sqrt{\frac{9}{5}(r'^2 + g'^2}$$

$$\mathrm{L} = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B}$$

Where

$$\mathrm{r}' = \mathrm{r} - \frac{1}{3} \qquad \mathrm{g}' = \mathrm{g} - \frac{1}{3} \qquad \mathrm{r} = \frac{R}{R+G+B}$$

$$\mathrm{g} = \frac{G}{R+G+B}$$

TSL is advantageous due to normalization in RGB to TSL transform. Normalization of R and G makes sensitivity of chrominance distribution reduced with the variability in skin color tones.

Terrillon[17] investigated the efficiency of skin detection for different color spaces. For this purpose, faces in 90 images with 133 faces and 59 subjects – 27 Asian, 31 Caucasian and 1 African were detected. TSL outperformed others schemes with 90.8% correct detection and 84.9% correct rejection as given in the table below. As a conclusion, we have decided to use TSL for skin detection task in our hand gesture recognition system. A brief comparison of different color coding schemes is also given in Table 1.

Table 1 Comparison of different color coding schemes

Name of the coding scheme	No of samples checked	Percentage of correct detection	Percentage of correct rejection
TSL	600	91.3	88.9
r-g	700	66.6	76.3
CIE-xy	650	68.2	83.5
CIE-DSH	800	67.5	75.0
HSV	750	77.7	84.7
YIQ	550	48.6	79.`1
YES	550	47.1	81.3
CIELUV	600	73.2	82.5
CIELAB	730	70.8	84.6

PROPOSED METHOD FOR SKIN COLOR DETECTION

The parametric method is used to detect skin color. We trained an AdaBoost classifier on TSL color space. People having different color tunes are selected for this purpose. A ratio higher by 20% to 40% than other skin tones for people with medium color gave best detection result. Because tan skin tunes cause noise during detection, their ratio is kept at the minimum level of 5% to 10%. Fine skin patches are cropped from these images using MATLAB built-in imfreehand function and color values are written in a text file in the RGB form. The resultant file contains 60% negative samples and 40 % positive samples. For training the classifier, this file converted to TSL color space. There are 363899 distinct skin color tones in the database. The T, S, and L color values are given by the first three columns of the text file.

The 1 or -1 in the last column indicate a skin or a non-skin sample.

NaN values are removed from the data and an AdaBoost classifier is trained which gave optimum results at 28 stages. As L represent luminance factor and have nothing to do with the color of skin so only Tint and Saturation values are used to train the AdaBoost classifier. Now, whenever we want to segment an image we first convert it into TSL color space and then it is segmented applying AdaBoost classifier. Some segmentation results are shown in the pictures given in Figure 3.

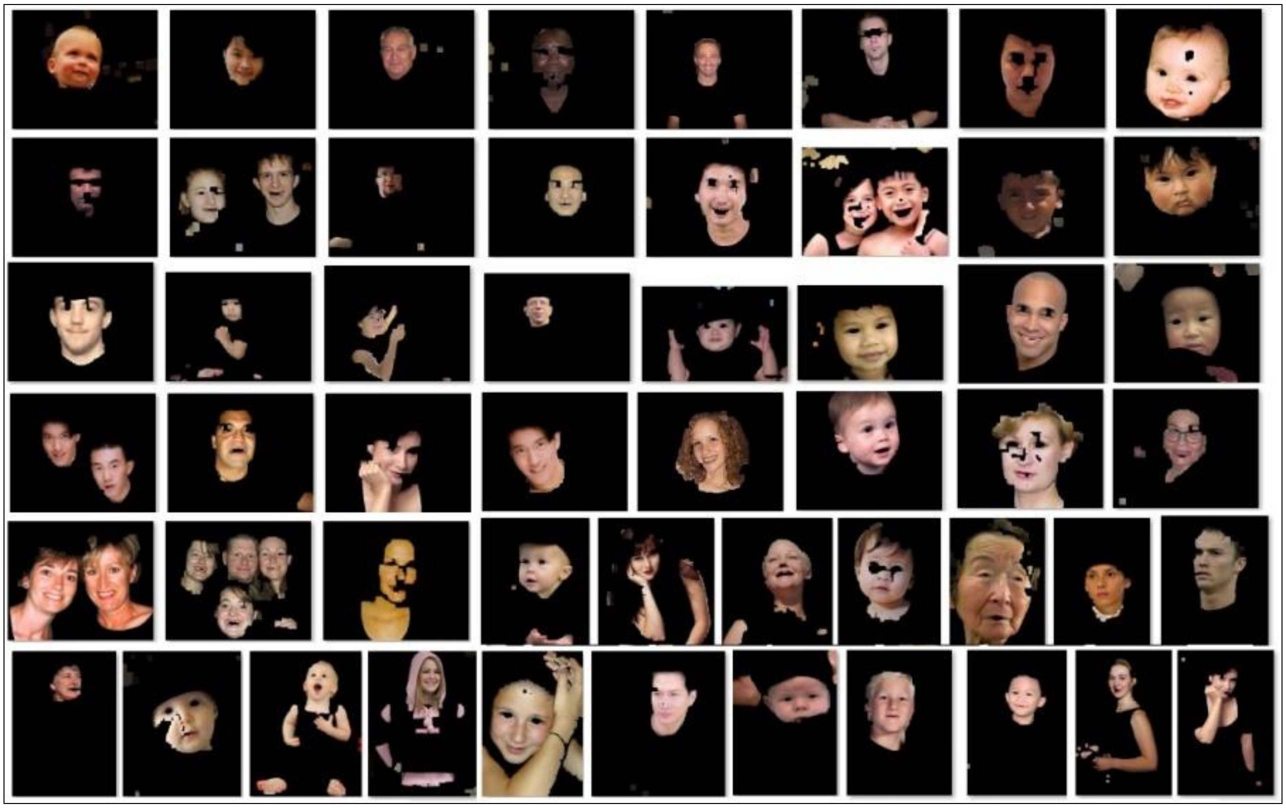

Figure 3 Application of skin detection algorithm on Lam Phung skin dataset

Training Cascade Hand Detector:

Cascade hand detector plays a vital role for hand detection in our algorithm. A lot of pre-trained models are already given in MatLab. But they are all applicable to rigid nondeformable objects. As the hand is consist of fingers that can change shape and turn around a large range of angles. So it is not simple to cover all those angles.

Cascade of detectors approach uses sliding window method in which a classifier is applied at most common positions, scales, and, in some cases, orientations of an image of a hand. It uses a collection of classifiers to efficiently locate a target object in the image regions. In each stage, an increasingly more complex binary classifier is used allowing the algorithm to exclude the regions that do not contain the target object. If at any stage of the cascade, the desired object is not located, the detector rejects the region immediately and further processing is terminated. In this manner, the invocation of computation-intensive classifiers further down the cascade is avoided.

For training the classifier we used 1209 positive samples. These hand samples belong to both men and women of different ages having various hand shapes while placing their hands in a semi-open or open mode with slight variations in orientations. For training, we used train cascade object detector function given in vision library of MATLAB version R2013b. The size of training samples is set to default that is each sample image is resized to 32 pixels resolution before training start. The result of training is an XML file containing a 12 stages classifier.

While applying this classifier the min and max size of detection is reset to ±20 % of the size of the previous hand detected. It is done taking an assumption that within two consecutive frames hand size will not change more than that. Or in the simple words very rapid movement of the hand is not allowed.

Hand Detection Process:

Hand detection process is divided into two parts, initialization, and tracking. In the first part presence of hand is confirmed by three different tests.

- By skin detection
- By confirming that it is not a face falsely detected as hand
- Cascade detector

Once a hand is detected it has to be almost still just for a movement so that at least five frames of it can be collected. After that, the detection process is complete and tracking process starts. During tracking some natural phenomena are kept in consideration. For example, new hand detected will surely be in some neighborhood of previously detected hand, its size will not vary a lot within consecutive frames and its orientation will not vary a lot. In case hand rotates out of our detector limit then the skin color detection tool is used to keep track of hand till it comes back to an orientation visible to the cascade detector shown in Figure 4.

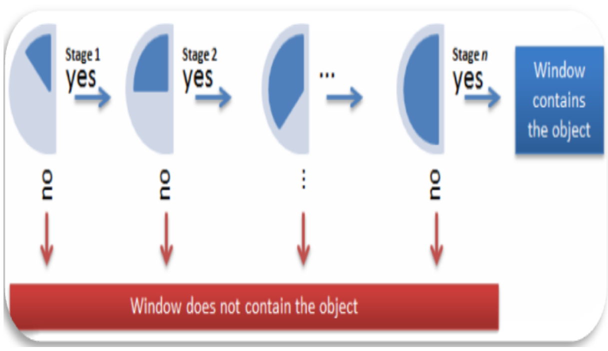

Figure 4 Cascade object detection block diagram

The problem of the face, mistakenly perceived as a hand is tackled with, using Viola and Jons face detector [14]. After hand detector is applied, a face detector is also applied to the same frame. If the hand region coincides with more than 50 % region of the face then it is presumed that it is not an original hand but is a face.

Another big problem with hand detection using skin color is that if the region of the arm is not covered with some cloth of non-skin color then it will also be detected with a hand while using skin detection. To solve this problem a very simple method is used i.e. the size of the detected hand is matched with the one previously detected by the detector. If there is a remarkable variation then the distance of the center of four corners of this detected region, cropped to the size of the previously detected region of hand is measured from the center of previously detected hand and the one with the least distance is select to be the cropped hand region.

RESULTS AND CONCLUSIONS

Experiments have shown that the cascade detector alone can detect the hand with a great variation in shape and orientation as shown in Figure 5.

Figure 5 Hand detected by alone cascade detector

When used in conjunction with a skin detector, improved results are obtained (See Figure 6) when the hand is tracked in a video clip captured by a simple webcam. An efficient detector for hand tracking was obtained by combining the two classifiers that demonstrated robustness to such conditions as skin color noise, rotation of the hand, brightness changes, the clutter of the background. The research suggests the possibility of detecting non-rigid objects with the classifier through the imposition of some limitations on the orientations of the classifier and training the classifier on the most conspicuous features of the object.

Figure 6 Hand detected by combination cascade detector and skin detector

An implementation of the technique for non-rigid objects can also be obtained. Furthermore, the same work combined with hand gesture recognition may also be used for ASL.

REFERENCES

1. S. Ghotkar, and G. K. Kharate, Hand segmentation techniques to hand gesture recognition for natural human-computer interaction, Int. Jour. of Human-Computer Interaction (IJHCI), 3(2012), 1-15.
2. D. I. Kosmopoulos, and I. G. Maglogiannis, Extraction of mid-level semantics from gesture videos using a Bayesian network. Int. Jour. of Intelligent Systems Technologies and Applications, 1(2006), 359-375.
3. D. N. Binh, S. Enokida, and T. Ejima, A New Approach Dedicated To Real-Time Hand Gesture Recognition. IPCV. (2006), pp. 481-488.
4. Y. Wu, J. Y. Lin, and T. S. Huang Capturing natural hand articulation, Proc. of 8th IEEE Int. Conf. on Computer Vision (ICCV), 2001, Vancour, Canada.
5. E. J. Ong, and R. Bowden. A boosted classifier tree for hand shape detection. Proc. of 6th IEEE Int. Conf. on Automatic Face and Gesture Recognition, 2004.
6. A. Shamaei and A. Sutherland, A dynamic model for real-time tracking of hands in bimanual movements. In: Camurri A., Volpe G. (eds) Gesture-Based Communication in Human-Computer Interaction. GW 2003. Lecture Notes in Computer Science, vol 2915. Springer, Berlin, Heidelberg.
7. A. Misra, A. Takashi, T. Okatani, and K. Deguchi, Hand gesture recognition using a histogram of oriented gradients and partial least square regression, MVA2011 IAPR Conference on Machine Vision Applications, 2011, Nara, Japan.
8. P. T. Bao, B. Nguyen, and T. D. Khoa, A new approach to hand tracking and gesture recognition by a new feature type and HMM, Fuzzy Systems and Knowledge Discovery, 6th Int. Conf. on. vol. 4. IEEE, 2009, Tainjin, China.
9. W. T. Freeman, and C. D. Weissman, Television control by hand gestures. Proc. of Int. Workshop on Automatic Face and Gesture Recognition, 1995.
10. A. Sepehri, Y. Yacoob, and LS Davis Employing the hand as an interface device. Journal of Multimedia 1(2006), 18-29.
11. J. Alon, V. Athitsos, Q. Yuan, S. Sclaroff, A unified framework for gesture recognition and spatiotemporal gesture segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2009), 1685-1699.
12. M. Maraqa, F. Al-Zboun, M. Dhyabat, and R. Abu-Zaiter, Recognition of Arabic Sign Language (ArSL) using recurrent neural networks, Journal of Intelligent Learning Systems and Applications, 4(2012), 41-52.
13. E. Stergiopoulou, K. Sgouropolous, N. Nikolaou, N. Papamarkus, and N. Mitianoudis, Real-time hand detection in a complex background. Engineering Applications of Artificial Intelligence, 35(2014), 54-70.
14. P. Viola, and M. J. Jones, Robust real-time face detection, International Journal of computer vision, 57(2004): 137-154.
15. J. Cai, and A. Goshtasby Detecting human faces in color images, Image, and Vision Computing 18 (1999) 63-75.
16. T. Kurata, T. Okuma, M. Kourogi, and K. Sakaue, The hand mouse: GMM hand-color classification and mean shift tracking. Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Proc. of IEEE ICCV Workshop, 2001, Vancour, Canada.
17. J. Terrillon, H. Fukamachi, S. Akamatsu, and M. N. Sherazi, Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. Automatic Face and Gesture Recognition, Proc. of 4rh IEEE Int. Conf., 2000.