**Chapter 4**

**Hand Posture Detection**

**4.1 Introduction**

Object detection play very important role in my own data set. To analyze and extract relevant data about and object of interest from an image, one need to first get that object in the image. Hand posture detection refers to finding the place and size of hand within a sequence of images. Now days it is very popular problem in computer vision problem and has many numerous applications, such as gesture recognition, sign language recognition, computer graphics games, and human computer interaction (HCI).

Skin color [Bretzn02, Mckenna04, Imagawa00] is an important property to detect hand and tracking. However, Color image have different problem of removing other objects with similar color such as face and human arm. To solve this problem, we introduce a new method in [Dardas11] to detect hand postures only using face detection and subtraction, skin detection, and hand postures contours detection and comparison algorithm. The face was detection removed because the skin detection will detect the face and the face’s contours are very similar to the fist hand gesture contours. The detected face was subtracted by replacing the face area with a black color. After removing unwanted area of the face, I detected the skin area using the hue, saturation, value (HSV) color model since it has real-time performance and it is strong against alternations, scaling and lighting conditions. Then, the interested area of contours was compared with all the existing hand posture template contours to eliminate unwanted interest of area like objects existing in the image.

**3.2 Hand Detection Approaches**

There are different approaches for hand detection have been introduced in the literature that employ different visual features and, in many cases, their combination. These features are motion, skin color, shape, and 3D models of hands. Hand detection methods were discussed in [Zabulis09] and will be discussed later in this chapter.

**3.2.1 Color**

Skin color detection has been used in many hand gesture recognition projects. A main objective towards giving a model of skin color is the choice of the color space to be utilized. Different color spaces have been introduced such as RGB, normalized RGB, HSV, YCrCb, and YUV**.** Color spaces that efficiently divide the chromaticity from the luminance parts of color are typically regarded as preferable as I did in my approach by removing the Value (V) section in HSV model. This is because of employing chromaticity-dependent mechanisms of color only. **Generally,** hull color detection can be disordered by background objects that have a hull color distribution like human skin color.Some project has been done on this problem by using background subtraction [Rehg94b, Gavrila96].On the contrary, it was expected that unwanted background subtraction normally depends on the camera system that does not move with respect to a fix background.Another solution [Utsumi98, Blake99] has applied the dynamic modification of background compensation techniques.

**3.2.2 Shape Hand**

Shape property have been working to discover in frames**.** More details can be acquired by reducing the contours of objects in the frame.If the contour is perfectly detected, it provides good presentation of the hand gesture which is indirectly related to viewpoint, skin color, and lighting. Typically, contour extraction based on edge recognition uses many edges fitting to the hand image area but also to distinct background objects.Accordingly, sophisticated post-processing techniques are needed to develop the presentation of this method such as our approach in [Dardas11] by combining skin color detection with contours detection and comparison after face subtraction. A second method that has been used in fingertip finding is pattern matching. Patterns can be images of fingertips [Crowley95] or fingers [Rehg95] or generic 3D cylindrical models [Davis94]. These pattern matching methods can be upgraded by using extra image features such as contours [Rehg94b].

**3.3 My Approach for Hand Detection**

I propose an integrated system for detection, segmentation and tracking of the hand in a gesture recognition system using a single webcam.Some other methods that use color gloves **[Aran06, Tokatl05],** my method can detect the plain hand posture by integrating two useful features: skin color detection and contour matching.my proposed hand posture finding algorithm has real-time performance and is strong against rotations, scaling, a cluttered background, and lighting conditions. **Section 4.4.2 shows the robustness of our proposed hand posture detection algorithm based on comparison with other approaches.** Detecting the human hand in a cluttered background will boost the performance of hand gesture recognition systems used in Chapter 5 and 6 in terms of accuracy and speed. In this method, the speed and result of recognition will be the same for any frame size taken from a webcam such as 640×480, 320×240 or 160×120 and the system will be also robust against cluttered background because I process the detected hand posture area only.The small image size that holds the detected hand posture area only must be similar with the training images size of training stage as I will discuss the training and testing recognition systems stages in Chapter 5.

**In order to detect the hand posture in the image, a four steps system was designed according to our approach [Dardas11] and as shown in Figure 3.1. First, the templates of hand postures were loaded before capturing images from a webcam or video file. Second, the face was detected using the Viola&Jones algorithm [Viola04] and then subtracted with a black circle. Third, the skin color locus for the image was extracted for the user’s skin color after face subtraction. Then as the fourth step, the hand posture was detected by eliminating false positive skin pixels and identifying hand posture and other real skin color regions using contours matching with the loaded hand postures templates contours. Skin Detection Area Loading Hand Postures Templates Contours Face Detection and Subtraction Capturing Images from Webcam or Video file Templates Contours Comparison with Skin Area Figure 3.1: Hand posture detection stages**

**3.3.1 Loading Templates of Hand Postures**

**For detecting hand gesture using skin color, there are different methods including skin color based methods. In our case, after detecting and subtracting the face, skin detection and a contour comparison algorithm were used to search for the hand and discard other skin coloured objects for every frame captured from a webcam or video file. Before capturing the frames from a webcam, we loaded the templates of hand gestures as shown in Figure 3.2: fist, index, little and palm to extract their contours and saved the four for comparison with the contours of the skin detected area of every frame. After detecting the skin area for every frame captured, we compared the contours of the detected areas with the previously loaded hand posture template contours to eliminate other skin like objects existing in the image. If the contour comparison of the detected skin area complies with any one of the stored hand postures contours, a small image will enclose the hand posture area only. As shown in Chapter 4, a small image (50×50 pixels) will be used to extract the keypoints using the SIFT algorithm for hand posture recognition. In Chapter 5, PCA will be used for extracting the features of the small image (160×120 pixels) to recognize the hand posture.**

**3.3.3 Skin Detection**

**Skin detection is a useful technique for many computer vision applications such as face detection, recognition, tracking and facial expression extraction, or hand tracking and gesture recognition. There are well-known methods for skin color modeling and recognition that will allow to differentiate between skin and non-skin pixels based on their color. In order to get appropriate distinction between skin and non-skin regions, a color transformation is needed to separate luminance from chrominance [Zhu04]. The input images generally are in RGB format, which has the disadvantage of having components dependent on the lighting conditions. The confusion between skin and non-skin pixels can be decreased using color space transformation. Many skin detection methods use color components in other color spaces, such as HSV, YCbCr, TSL or YIQ to provide more robust parameter recovery under changes in lighting condition. Experiments have shown that skin colors of individuals cluster closely in the color space for all people from different ethnicities, for example, color appearances in human faces and hands vary more in intensity than in chrominance [Jun08, Kelly08, Yang98]. Thus, removing the intensity V of the original color space and working in the chromatic color space (H,S) provides invariance against illumination conditions. In [Zhu04], it had been noted that discarding the Value (V) component and only using the Hue and Saturation components, can still permit for the detection 96.83% of the skin pixels. In our implementation, we use the hue, saturation, value (HSV) color model since it has shown to be one of the most adapted to skin-color detection [Zarit99]. It is also compatible with the human color perception. Besides, it has real-time performance and it is more robust in cases of rotations, scaling, cluttered background, and changes in lighting condition. Therefore, our proposed hand posture detection algorithm is real-time and robust against the mentioned previous changes. The other skin like objects existing in the image are eliminated by contour comparison with the loaded hand postures template contours. The HSV color space is obtained by a non-linear transformation of the fundamental RGB color space. The transformation between RGB and HSV was described in [Ford98]. Hue (H) is a component that represents pure color such as pure yellow, orange or red, whereas saturation (S) provides a measure of the degree to which a pure color diluted by white light [Gonzal04]. Value (V) attempts to represent brightness along the grey axis such as white to black, but since brightness is subjective, it is thus difficult to measure [Gonzal04]. According to [Foley96] and Figure 3.7, Hue is measured in HSV color space by an angle with Red starting at 0 degrees, Green at 120 degrees and Blue at 240 degrees. The black line in the diagram at the lower left on the screen demonstrates the hue angle. Saturation, S, is a ratio that ranges from 0.0 along the center line of the cone (the V axis) to 1 on the edge of the cone. Value, V, ranges from 0.0 (dark) to 1.0 (bright).**

**According to [Jun08], the HSV model can be derived from non-linear transformation from an RGB model according to the following equations. H= (3.1) S = (3.2) V = (3.3) θ = arcos (3.4) From a classification point of view, skin-color detection can be considered as a two class problem: skin-pixel vs non-skin-pixel classification. There are many known classification techniques such as thresholding, Gaussian classifier, and multilayer perceptron [Nallap07, Greens01, Phung01]. In our implementation, we used a thresholding method that allows for higher computation speed when compared with other techniques, given our real-time requirements. The basis of this thresholding classification is to find the range of two components H and S in the HSV model as we discarded the Value (V) component. Usually a pixel can be viewed as being a skin-pixel when the following threshold ranges are simultaneous satisfied: 0° < H < 20° and 75° < S < 190°.**

**3.3.4 Contour Comparisons**

**Once the skin area is detected, the contours of the detected areas are recovered and then compared them with the contours of the hand posture templates. If the recovered contours are recognized as belonging to the hand posture contour templates, that area will be identified as a region of interest (by enclosing the detected hand posture with a rectangle) which will then be used for tracking the hand movements and saving the hand posture in a PGM-format small images as shown in Figure 3.7. These images will further be used to extract the features needed to recognize the hand postures in the testing stage as discussed in Chapters 4 and 5. The total time needed to detect the hand posture is around 15 milliseconds for every frame captured.**

**If there are two hand postures in the image, our system will alternate in detecting one of the two hands for every frame captured because the OpenCV function cvBoundingRect will enclose one rectangle only around the detected hand, which has the largest matching contours with the loaded hand posture templates contours. The single rectangle will enclose the detected hand posture for one frame and may enclose the other hand posture for the next frame if it has a larger matching contour.**

**3.4 Summary**

**In this chapter, we described the use of skin detection and the contour comparison algorithm to detect the hand posture. This approach can also be used to detect the face because the face has a skin color and its contours are similar to the hand fist posture contours. To eliminate the face area, we detected the face using the Viola and Jones method and then subtracted the face before applying the skin detection algorithm to detect the hand posture only by replacing the face area with a black circle for every frame captured. Skin detection and the contour comparison algorithms were used for detecting the human hands and discarding other skin-coloured objects after face detection and subtraction. Before capturing the frames from a webcam, we load the templates of hand postures: fist, index, little, and palm to extract their contours, and save them for comparison with the contours of the skin area of every frame captured. After detecting the skin area using the HSV color model, we compared the contours of that area with the loaded hand posture template contours to eliminate other skin-like objects existing in the image. If the contours of the detected skin area comply with any one of the loaded hand posture templates contours, a small image will enclose the hand posture area only. The small image will be used for extracting features to recognize the detected hand posture. Our approach will be used in Chapter 4, 5, and 6 for hand posture detection before recognition.**

American sing language has different gesture for each alphabet and numbers. I captured the images for each sign. In which consist unwanted noise and removed the backgrounds from each of the images using background-subtraction techniques.

**4.2 Computer Vision**

Open CV is an open source library (see: <http://opensouces.org>) computer vision library available <http://sourceForge.net/Projects/opencvlibrary>. The library is written in C an C++ and runs under Window, Linux, and Mac OS X. There is active development on interfaces for python,Ruby,Matlab and other language.

Open CV was designed for computational efficiency and with a strong focus on real time application. OpenCV is written in optimized C and can take advantage of multicore processor.

Open CV main goal is to provide a simple to use computer vision infrastructure that helps people build fairly sophisticated vision application quickly. OpenCV library contains over 500 function that span many area in vision, including factory product inspection,medical imaging,security, user interface,camera calibration and robotics. OpenCV also contain full,general purpose Machine learning library.

Computer vision is the transformation of data from still or video camera into either a decision or a new representation. All such transformation are done for achieving some particular goal. The input data may include some contextual information such as the camera is mounted in a car or laser range finder indicates an object is 1 meter way. The decision might be “there is a person in this scene “ or “there are 14 tumor cell on this slide”.A new representation might mean turning a color image to a grayscale image age or removing camera motion from an image sequence.

<https://books.google.ca/books?hl=en&lr=&id=seAgiOfu2EIC&oi=fnd&pg=PR3&dq=computer+vision+with+the+opencv+library&ots=hUG8afjBN9&sig=5-ygSPco-0195GM9VVVgMyIauks#v=onepage&q=computer%20vision%20with%20the%20opencv%20library&f=false>

chapter 1 : page 1 and 2

**https://pdfs.semanticscholar.org/edb8/858d07f51cca833acddb499efdc4abe477eb.pdf**