

Design and Implementation of Automated Skin Detection using FPGA

Syed Khalid Hussain

Department of Electronic & Communication Engineering
CMRIT Engineering College,
Bangalore-560041

Prof. Harsha B K

Asst. Professor
Department of Electronic & Communication Engineering
CMRIT Engineering College, Bangalore-560041

Abstract:- Skin color detection is applied to many applications as hand tracking, human-machine interactions, face detection and searching people. However, detecting skin color requires repetitive operations on all pixels in the image, similar to other vision-based applications. The skin color is usually in the form of RGB color space mode which is a category of an explicitly defined skin region model and is converted to YUV color model and again YUV is converted to grey scale color model. Applying thresholding on grey scale values on binary image, i.e. an image whose pixels are either 1 or 0; 1 if it is a skin pixel and 0 if non skin pixel after skin detection algorithm, this binary image is obtained from the Matlab code and the proposed method flattens the image surface of objects in the scene by replacing the pixel value with the mean of its similar neighborhoods to remove the color noise and gradient detection is used to obtain the edges of an image. Finally median filter is used for image enhancement. The problem arises from many objects in the real world that causes similarity in skin-tone color such as sand, leather, hair and wood. Moreover, skin color is differing from a person to another. Finally, image enhancement is performed to improve the image quality.

Skin color detection is implemented on FPGA vertex2 pro board using Xilinx System Generator based on Hardware/Software Co-simulation and experimental results proved the accuracy and effectiveness, even under varying conditions of lights, skin colors and facial poses. All the hardware implementation is done in real time with minimal efforts and thus suitable for low powered applications.

I. INTRODUCTION

Image processing is the most scientific application, which can be used in many domains. Some of these applications are human tracking, face detection, hand gesture recognition Human identification, filtering image contents on the web searching, image retrieval and visual tracking for surveillance and many others. One of the most popular applications where it can be used is skin detection and tracking has been an extensive research for the several past decades, as computer systems are becoming more and more complex, more sophisticated human machine interfaces are required because of the mentioned person specific and context specific variations, skin color looks very differently throughout the images. Skin is the largest organ of human body and it is a soft outer covering of human's muscles, bones, ligaments, and internal organs. Skin color is produced by a combination of melanin, hemoglobin and carotene. A programmer is able to do that adaptation manually by evaluating the image and the

visible face, but most face interpretation systems must run automatically and obtains the image specific conditions of skin color. In order to extracts a small number of skin color pixels within those facial regions and in order to set up a skin color model that is representative for the person and context conditions. The performance of skin classifier was measured based on true positive (TF) and false positive (FP) indicator.

Since the image is input which cannot run be directly implemented on FPGA board so some matlab coding is required to convert an image to a binary pixel values. The skin color model is the RGB type model because the skin color classifiers result a skin color image that represents the information from the original RGB image that is relevant for fitting a model. The RGB model is able to reduce face detection that caused by reddish objects color as well as be able to detect darkened skin and skin covered by shadow and further this color skin model is converted to YUV model n we can see the output of YUV model converted image on the monitor and again the YUV model is converted to grey scale model to see the output image of grey scale model. Conversion of YUV to grayscale is not unique it requires different weighting of color channels which effectively represent the effect of black-and-white pixels with different-colored photographic filters In order to remove the background noise inversion is performed. Gradient detection is performed as a next step to determine the edges of an image and to improve it. Since the quality of the image is poor so image enhancement is done by means of median filter to improve the quality and the output of skin detection is displayed on the monitor as final output with minimum optimize power dissipation and area. The architecture for automated skin detection is shown below.



Figure 1: The Skin Detection Architecture

In real world applications, it is desirable to have a stand-alone, embedded skin recognition system. The reason is that such systems provide a higher level of robustness, hardware optimization, and ease of integration. The matlab code is utilized as a part of the Xilinx programming to play out the skin discovery by method for putting away the pixel values in a double port RAM, which are produced from the matlab from an information picture .The Integrated Software Environment (ISE) is the Xilinx plan

programming suite that permits you to take your configuration from outline passage through Xilinx gadget programming. The ISE Project Navigator oversees and forms your outline through the accompanying strides in the ISE plan stream. In this the initial step performed is configuration passage amid outline section, you make your source records in light of your outline destinations. You can make your top-level outline record utilizing a Hardware Description Language (HDL, for example, VHDL, Verilog, or ABEL, or utilizing a schematic. You can utilize various arrangements for the lower-level source records in your outline. After configuration passage and discretionary reenactment, you run amalgamation. Amid this progression, VHDL, Verilog, or blended dialect outlines get to be netlist documents that are acknowledged as contribution to the usage step. After amalgamation, you run outline usage, which changes over the sensible configuration into a physical record arrange that can be downloaded to the chose target gadget. From Project Navigator, you can run the execution procedure in one stage, or you can run each of the usage forms independently. Execution forms differ contingent upon whether you are focusing on a Field Programmable Gate Array (FPGA) or a Complex Programmable Logic Device (CPLD). You can confirm the usefulness of your outline at a few focuses in the configuration stream. You can utilize test system programming to confirm the usefulness and timing of your outline or a bit of your configuration. The test system translates VHDL or Verilog code into circuit usefulness and showcases consistent aftereffects of the depicted HDL to decide right circuit operation. Reenactment permits you to make and confirm complex capacities in a generally little measure of time. You can likewise pursue in-circuit check programming your gadget. Subsequent to producing a programming record, you arrange your gadget. Amid arrangement, you produce design records and download the programming documents from a host PC to a Xilinx gadget. As such, we have chosen the FPGA as a reconfigurable platform to carry out our implementation. Ultimately, the stand alone system may be implemented on ASIC, a dedicated processor, or even an FPGA chip, depending on the trade-offs in speed, portability, and reconfigurability.

II. LITERATURE SURVEY

Existing color-based skin segmentation techniques take advantage from the observation that skin-tone color has common properties which can be defined in various color spaces. In general, skin color detectors rely on rule-based or statistical skin modeling. A thorough survey comparing various color-based skin detection approaches was presented by Kakumanu et al. (2007).

There are a number of methods which are based on fixed decision rules defined in various color spaces after analyzing skin-tone distribution. The rules are applied after color normalization to determine if a pixel color value belongs to the skin. Skin-tone color was modeled in the HSV color space by Tsekeridou and Pitas (1998). Kovac et al. (2003) proposed a model defined in the RGB color

space. An approach introduced by Hsu et al. (2002) takes advantage of common skin color properties in nonlinearly transformed YCbCr color space using an elliptical skin color model. A technique operating in multiple color spaces to increase the stability was described by Kukharev and Nowosielski (2004). Cheddad et al. (2009) proposed reducing the RGB color space to a single dimension, in which the decision rules are defined.

Statistical modeling is based on analysis of skin pixel values distribution for a training set of images, in which skin and non-skin areas are identified and annotated. This creates a global model of skin color, which allows determining the probability that a given pixel value belongs to the skin class. Skin color can be modeled using a number of techniques, including the Gaussian mixture model (Greenspan et al., 2001) and the Bayesian classifier (Jones and Rehg, 2002). The latter method, given more attention was used in the research reported here to generate the skin probability maps.

There are a number of adaptive models that are designed to decrease the impact of overlaps between skin and non-skin pixels in a color space, which improves the segmentation accuracy for a presented scene. In general, adaptive approaches can be divided according to the information source used for the adaptation, i.e. An easy and most often used method is the definition of classifiers that build upon the approach of skin clustering is thresholding of different color space coordinates is used in many approaches that explicitly defines the boundaries of the skin clusters in a given color space. The underlying hypothesis here is that skin pixels have similar color coordinates in the chosen color space, which means that skin pixel are found within a given set of boundaries in a color space. The main drawback of this method is not the resulting true positives, but a comparably high number of false detections. We are able to compensate for this issue in our approach by using a multiple adaptive model approach. skin-like object tracking, whole-image analysis, face and hand detection, and the model type being adapted, i.e. threshold-based, histogram analysis, Gaussian mixture models.

Lee et al. (2007) proposed a method based on a multi-layer perception extracting lighting features from an analyzed image to adjust the skin detector. An approach for adapting the segmentation threshold in the probability map based on the assumption that a skin region is coherent and should have homogenous textural features was introduced by Phung et al. (2003). This method was further extended by Zhang et al. (2004) by involving the artificial neural network (ANN) for estimating an optimal acceptance threshold. The ANN was also used for adaptation by Yang et al. (2010). A method for a dynamic model adaptation based on observed changes in the histogram extracted from a tracked skin region was proposed by Soriano et al. (2000). Motion detectors for the skin color model adaptation were explored by Dadgostar and Sarrafzadeh (2006). Analysis of facial regions for effective adaptation of the skin model to local conditions was investigated by Fritsch et al. (2002); Stern and Efros, 2002; Kawulok, 2008; Kawulok et al., 2013 and Yogarajah et al., 2012.

Analysis of textural features extracted from an input image was applied to improve the performance of color-based methods. In the approach proposed by Wang et al. (1985) segmentation in the RGB and YUV color spaces is enhanced by analyzing various textural features, including contrast, entropy, homogeneity and more, extracted using the gray-level co-occurrence matrix (GLCM). The experimental results reported in the original work showed that the method is competitive for complex background detection, but the skin detection rate was significantly worse compared to color-based approaches. Additionally, the time complexity of calculating GLCM is proportional to $O(g^2P)$ (Clausi and Jernigan, 1998), where g is the number of gray levels of the input grayscale image.

An interesting algorithm incorporating color, texture and space analysis was given by Jiang et al. (2007). Initially, skin probability map color filter with a low acceptance threshold is applied in the RGB color space. Then, textural features are extracted using the Gabor wavelets from an input color image converted to the grayscale. The obtained response is subject to a threshold HT , which produces a binary texture mask. The aim of applying the texture mask is to reduce the false positive rate by filtering out the regions with large values of the texture feature, i.e. those that are not as smooth as skin, but were improperly classified as skin by the color filter. Finally, skin regions are grown using the watershed segmentation with well-defined region markers to exploit the spatial information. It was shown that the method reduced the false positive rate (from 20.1% to 4.2%) with simultaneous increase of the true positive rate (from 92.7% to 94.8%) compared to the color filtering for a data set containing 600 images. However, the authors did not provide any sensitivity analysis and it seems that a different threshold value is applied to every image, which makes it difficult to get satisfactory results for a larger number of images. Additionally, if human skin is not smooth (e.g. in case of the elders), then it may also be filtered out by the texture filter. On the contrary, if the skin-like background is smooth, then the misclassified pixels are not filtered out.

Simple textural features were used to boost the performance of a number of skin detection techniques and classifiers, including the ANN (Taqa and Jalab, 2010), non-parametric density estimation of skin and non-skin classes (Zafarifar et al., 2010), Gaussian mixture models (Ng and Pun, 2011), and many more (Forsyth and Fleck, 1999, Conci et al., 2008, Fotouhi et al., 2009, Abin et al., 2009). Generally, the analysis of texture in an input image helps reducing the number of pixels misclassified by pixel-wise color-based detectors. However, the roughness of skin and non-skin regions can vary among images, which in turn makes the response of a texture-based segmentation algorithm difficult to generalize for real-life data sets. In none of the mentioned methods the skin probability maps were analyzed with regards to their textural features.

Although the color-based skin models can be efficiently adapted to a given image, it was proved by Zhu et al. (2004) that it is hardly possible to separate skin from non-skin pixels using such approaches. It is easy to see that skin pixels are usually grouped into blobs whereas the non-

skin false positives are scattered around the spatial domain. A number of skin segmentation techniques emerged based on this observation: Kruppa et al. (2002) assumed that the skin blobs are of an elliptical shape, a threshold hysteresis was applied by Argyros and Lourakis (2004) and recently by Baltzakis et al. (2012). Conditional random fields were used by Chenaoua and Bouridane (2006) to exploit spatial properties of skin regions. An approach based on the cellular automata for determining skin regions was proposed by Abin et al. (2009).

The analysis of skin probability map domain for skin segmentation using a controlled diffusion was proposed by del Solar and Verschae (2004). Here, the diffusion seeds are extracted at first. They are formed by those pixels, whose skin probability, extracted from the pixel-wise skin probability maps, exceeds the seed threshold (P_a). Then, the skin regions are built according to the criteria of the diffusion process. A neighboring pixel p_j is adjoined to the source pixel p_i if a distance between p_i and p_j in the diffusion domain is smaller than a given diffusion threshold (D) and the skin probability of p_j is larger than the propagation threshold (P_b). The main drawback of this method is its performance in case of blurry region boundaries, since the diffusion process does not stop if the transitions between skin and non-skin pixels are smooth. In our earlier research (Kawulok, 2010) we introduced an energy-based technique for skin blobs analysis. The pixels are adjoined to the skin regions depending on the amount of the energy which is spread over the image according to the local skin probability. Recently, we proposed to use the distance transform in a combined domain (DTCD) of hue, luminance and skin probability (Kawulok, 2013). The algorithm was proved to be very competitive and outperformed our energy-based method and the method proposed by del Solar and Verschae (2004). We overcame the most significant shortcoming of the latter approach, i.e. misbehaving in case of smooth transitions between skin and non-skin regions, by taking advantage of the cumulative character of the distance transform. This method is exploited in the research reported here as

$$P(V/C_x) = C_x(V)/N_x \quad (2.1)$$

where $C_x(V)$ is the number of v -colored pixels in the class x and N_x is the total number of pixels in that class. Maximal number of histogram bins depends on the pixel bit-depth and for most color spaces it equals $256 * 256 * 256$. However, it was reported beneficial (Phung et al., 2005; Kawulok et al., 2014) to reduce the number of bins per channel, thus, in our research we used 64 bins per each channel in the RGB color space.

It may be expected that a pixel presents the skin, if its color value has a high density in the skin histogram. Moreover, the chances for that are larger, if the pixel color is not very frequent among the non-skin pixels. Taking this into account, the probability that a given pixel value belongs to the skin class is computed using the Bayes rule

$$P(C_x/V) = (P(V/C_x) * P(C_s)) \quad (2.2)$$

$$(P(V/C_x) * P(C_x) + P(V/C_n) * P(C_n)) \quad (2.3)$$

where a priori probabilities $P(C_s)$ and $P(C_{ns})$ may be estimated based on the number of pixels in both classes, but very often it is assumed that they both equal $P(C_s) = P(C_{ns}) = 0.5$. The learning phase consists in creating a skin color probability look-up table, which maps every color value in the color space domain into the skin probability. After training, using the look-up table, an input image is converted into a skin probability map, in which skin regions may be segmented based on an acceptance threshold (Pacc). The threshold value should be set to provide the best balance between the false positives and false negatives, which may depend on a specific application. An approach for detecting skin in this kind of content has therefore to be fast, reliable and needs to be stable against noise and artifacts caused by compression. Additionally, it must be very flexible against varying lighting conditions so we use thresholding method with haar feature filtering and spatial filtering for skin detection and is implemented on fpga kit so as to get the fast and accurate results.

III. PROBLEM STATEMENT

The main problem of skin color detection is to develop a skin color detection algorithm or classifier that is robust to the large variations in color appearance. Some objects may have almost similar skin-tone color which easily confused with skin color. A skin color can be vary in appearance base on changes in background color, illumination, and location of light sources, and other objects within the scene may cast shadows or reflect additional light.

Secondly, there are no specific methods or techniques that have been proposed to robust skin color detection arise under varying lighting conditions, especially when the illumination color changes. This condition may occur in both out-door and in-door environments with mixture of day light and artificial light.

Thirdly, many non-skin color objects are overlapping with skin color, and most of pixel-based method proposed in the literature cannot solve this problem. This problem is difficult to be solved because skin-like materials are those objects that appear to be skin-colored under a certain illumination condition.

In order to enable skin color detection to cope with the above-mentioned problems, the objective of this study is to enhance a skin color detection system by using RGB model. To achieve the aforementioned objective, a new skin images dataset have been developed for training and testing. Besides that, a benchmark skin images from Testing Database for Skin Detection and Skin image dataset from FPGA board with minimum power dissipation and area.

IV. SKIN COLOR MODEL

Color model is a technique by which color can be indicated, made, and envisioned. A human characterized a color by its qualities of intensity, tint, and brightness. A computer depicted a color utilizing measures of red, green, and blue phosphor outflow required to coordinate a color. The decision of color space can be considered as the

essential step in skin-color arrangement. The RGB color space is the default color space for most accessible picture positions. The other color space can be acquired from a direct or non-straight change from RGB. The color space change is accepted to diminish the cover amongst skin and non-skin pixels in this way helping skin-pixel arrangement and to give strong parameters against differing enlightenment conditions. It has been watched that skin hues vary more in force than in chrominance. Thus, it has been a typical practice to drop the luminance part for skin order. A color is typically determined utilizing three co-ordinates or parameters. These parameters portray the position of the color inside the color model is being utilized. Be that as it may, it is still not clear which are the best color model to be utilized for skin detection. For numerous color model restrictions the most reasonable color model for their skin color recognition technique is picked. Along these lines, diverse color models have been utilized for various skin color dissemination models, for example, RGB , RGB standardized, proportion sort RGB , CIE-XYZ, HSV , YCbCr , YIQ , YES , YUV , CIE-XYZ, and CIE-LUV.

4.1 RGB Color Model

The RGB color model is determined as far as the three essential hues: red (R), green (G), and blue (B). It is begun from Cathode Ray Tube (CRT) shows the application when it is advantageous to show color as a blend of three hued beams (red, green, and blue).The RGB color model is a standout amongst the most broadly utilized color model for preparing and putting away of advanced picture information. It likewise utilized for web pictures. There are high relationship between's channels, critical perceptual non-consistency blending of chrominance and luminance information make RGB not an exceptionally positive decision for color investigation and color based acknowledgment calculations.

One fundamental point of preference of the RGB color model is its effortlessness, yield entirely fulfilling execution and pace managing web pictures and as a rule skin color identification should be possible straightforwardly on pixel esteem without color model transformation. The luminance of a given RGB pixel is a straight blend of the R, G, and B values. In this manner, changing the luminance of a given skin patch influences all R, G, and B parts. At the end of the day, the estimation of the RGB will vary in view of the force of the light.

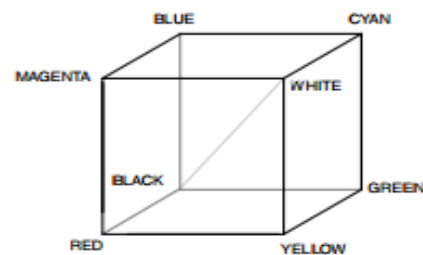


Figure 4: Representation of RGB color model

4.2 Orthogonal Color Models (YCbCr, YIQ, YUV, YES)

The YCbCr and YIQ are orthogonal color models. The YCbCr is an advanced color framework, while YIQ are simple space for the NTSC (National Television System Committee) frameworks. The YCbCr color model is here and there alluded to as the CCIR 601. These gadget subordinate color models are have a place with the group of TV transmission color model. This color model was characterized in light of expanding requests for computerized approaches in taking care of video data and has been utilized broadly as a part of advanced video.

YCrCb is an encoded nonlinear RGB signal, usually utilized by European TV studios and for picture pressure work. Color is spoken to by luma (which is luminance, registered from nonlinear RGB, developed as a weighted aggregate of the RGB qualities, and two color distinction values Cr and Cb that are shaped by subtracting luma from RGB red and blue parts.

$$Y = 0.299R + 0.587G + 0.114B \quad (4.1)$$

$$Cr = R - Y \quad (4.2)$$

$$Cb = B - Y \quad (4.3)$$

where R, G, and B are the estimation of red, green, and blue, individually. The orthogonal color spaces decrease the excess present in RGB color channels and speak to the color with measurably free segments (as autonomous as could be allowed). As the luminance and chrominance segments are expressly isolated, these spaces are a great decision for skin discovery. The YcbCr space is a standout amongst the most prominent decisions for skin discovery. This new color space contrasts from YCbCr in the use of Cg color part rather than the Cb segment and was accounted for to be superior to anything YCbCr. Other comparable color spaces in this class incorporate YIQ, YUV and YES, and speak to color as luminance (Y) and chrominance.

4.3 Gray scale model

Grayscale is a scope of shades of dim without clear shading. The darkest conceivable shade is dark, which is the aggregate nonappearance of transmitted or reflected light. The lightest conceivable shade is white, the aggregate transmission or impression of light at all unmistakable wavelengths. Transitional shades of dim are spoken to by equivalent splendor levels of the three essential hues (red, green and blue) for transmitted light, or equivalent measures of the three essential colors (cyan, fuchsia and yellow) for reflected light.

On account of transmitted light (for instance, the picture on a PC show), the splendor levels of the red (R), green (G) and blue (B) parts are each spoken to as a number from decimal 0 to 255, or double 00000000 to 11111111. For each pixel in a red-green-blue (RGB) grayscale picture, $R = G = B$. The softness of the dim is specifically relative to the number speaking to the splendor levels of the essential hues. Dark is spoken to by $R = G = B = 0$ or $R = G = B = 00000000$, and white is spoken to by R

$= G = B = 255$ or $R = G = B = 11111111$. Since there are 8 bits in the paired representation of the dark level, this imaging strategy is called 8-bit grayscale.

On account of reflected light (for instance, in a printed picture), the levels of cyan (C), fuchsia (M), and yellow (Y) for every pixel are spoken to as a rate from 0 to 100. For every pixel in a cyan-fuchsia yellow (CMY) grayscale picture, each of the three essential shades are available in equivalent sums. That is, $C = M = Y$. The gentility of the dark is conversely relative to the number speaking to the measures of every color. White is along these lines spoke to by $C = M = Y = 0$, and dark is spoken to by $C = M = Y = 100$.

In a few frameworks that utilization the RGB shading model, there are 2^{16} , or 65,536, conceivable levels for every essential shading. At the point when $R = G = B$ in this framework, the picture is known as 16-bit grayscale on the grounds that the decimal number 65,536 is proportionate to the 16-digit twofold number 11111111111111. Similarly as with 8-bit grayscale, the daintiness of the dark is straightforwardly corresponding to the number speaking to the brilliance levels of the essential hues. As one may expect, a 16-bit computerized grayscale picture expends much more memory or capacity than the same picture, with the same physical measurements, rendered in 8-bit advanced grayscale.

In a few frameworks that utilization the RGB shading model, there are 2^{16} , or 65,536, conceivable levels for every essential shading. At the point when $R = G = B$ in this framework, the picture is known as 16-bit grayscale on the grounds that the decimal number 65,536 is proportionate to the 16-digit twofold number 11111111111111. Similarly as with 8-bit grayscale, the daintiness of the dark is straightforwardly corresponding to the number speaking to the brilliance levels of the essential hues. As one may expect, a 16-bit computerized grayscale picture expends much more memory or capacity than the same picture, with the same physical measurements, rendered in 8-bit advanced grayscale.

Now and again, as opposed to utilizing the RGB or CMY shading models to characterize grayscale, three different parameters are characterized. These are tint, immersion and splendor. In a grayscale picture, the tint (evident shading shade) and immersion (obvious shading power) of every pixel is equivalent to 0. The gentility (obvious brilliance) is the main parameter of a pixel that can fluctuate. Delicacy can go from at least 0 (dark) to 100 (white).

V. DESIGN AND IMPLEMENTATION

In this project the input is an image, which cannot be directly run on Xilinx software which is implemented on FPGA vertex 2 pro board so here some matlab coding is required to convert the input image to a binary pixel values. The matlab code resize the input image to a 100*100 image so that it can fit into monitor and can be processed easily with 10000 pixels vales which is future stored in a dual port RAM with each port consisting of 10000 pixels. The use of dual port RAM is to store the input image in two ports A and B. Port A is directly connected to an output so

that it gives an output same as input image on a monitor and the second port that is port B is applied to an skin detection algorithm.

In matlab code three variable are used i,j and k where i indicates row, j indicates column and k to store the RGB values. To get an image 'UIgetfil' inbuilt function is used to read an image and the matlab code stored in the form of .coe file that contains the converted binary pixel values from an input image. This .coe file is used in xilinx software for skin detection algorithm.

The image consists of three form of color space RGB where each color consists of 8 bits so total 24 bits are used as a width of an image in dual port RAM thus 10000*24 size of an input image is stored in memory as length and width. Here in this execution two counter are used one for portA and portB to read different pixel values from the memory locations with different address values. After reading each pixel values counter is incremented and jumped to the different memory address. These pixels values are applied to the skin detection algorithm to detect the skin part. The pixels values which are stored in the form of RGB are converted to YUV color model by using the following equation

$$H=(R+2*G+B)/4 \tag{5.1}$$

$$U=R-G \tag{5.2}$$

$$V=B-G \tag{5.3}$$

YUV is a shading space commonly utilized as a major aspect of a shading picture pipeline. It encodes a shading picture or video considering human discernment, permitting decreased transfer speed for chrominance parts, in this manner ordinarily empowering transmission blunders or pressure ancient rarities to be more proficiently covered by the human observation than utilizing a "direct" RGB-representation. Other shading spaces have comparable properties, and the fundamental motivation to actualize or examine properties of YUV would be for interfacing with simple or advanced TV or photographic hardware that adjusts to certain YUV gauges. YUV signs are regularly made from RGB (red, green and blue) source. Weighted estimations of R, G, and B are summed to create Y, a measure of general shine or luminance. U and V are registered as scaled contrasts amongst Y and the B and R values. After converting RCB to YUV next YUV is converted to grey scale color model so that YUV values can be evaluated separately and so that is considered as grey scale model, thus grey scale is applied to the gradient detection to find the edges of the image using the following equation and the grey scale values is considered as R_o, G_o and B_o.

$$R_o = \frac{393*R}{66} + \frac{769*G}{64} + \frac{189*B}{64} \tag{5.4}$$

$$G_o = \frac{349*R}{64} + \frac{686*G}{64} + \frac{168*B}{64} \tag{5.5}$$

$$B_o = \frac{272*R}{64} + \frac{534*G}{64} + \frac{131*B}{64} \tag{5.6}$$

The widely used edge detection process is gradient operator which has many variations. Mathematically, for a

image function, f(x,y), the gradient magnitude is g(x,y) and the gradient direction is Θ(x,y) and is expressed as

$$G(x,y) = (\Delta x^2 + \Delta y^2)^{\frac{1}{2}} \tag{5.7}$$

$$\Theta(x,y) = \text{atan} \frac{\Delta y}{\Delta x} \tag{5.8}$$

Where, Δx = f(x+n,y) – f(x-n,y) and Δy = f(x,y+n) – f(x,y-n)

n is a small integer, that is unity. For example, the simple implementation of this is convolve the following mask with the image data, aligning the mask with the x and y axes to solve the values of Δx and Δy

$$-1|0|1$$

Since the quality of image is of low quality and some noise exist thus this noise is removed by inverting to the pixel values so to improve the quality of image and thus image enhancement is also performed by adding the extra values that is by multiples some different values to pixels using median filter. Median filters reduce the impulse noise level from corrupted images and also remove the salt-and-pepper noise. The median filter is a nonlinear smoothing operation that takes a median value of the data which is present inside a moving window of finite length. Median filter is also used to evaluate the mean value of filter. A methodology based on median filter to remove the noise is by its detection followed by filtering in binary images this whole procedure is called automated skin detection algorithm. The skin pixel value is finally decided on the basic of following condition.

$$10 < U < 74 \text{ and } 40 < V < 111$$

If the pixel value exists between this than it is a skin pixel which is indicated by 1 else it is a non skin which is indicated by 0. This skin detection algorithm which is implemented in this project with minimum power dissipation of 0.103W, the optimized area used is 6% and the delay of 3.90ns. The skin detection algorithm is shown below

Here PC, monitor and FPGA vertex 2pro board are used where PC and a monitor is connected to FGPA vertex2 pro kit, after execution of a program in PC using Xilinx 10.1, the output is displayed on monitor screen. Monitor screen is further divided into two parts where two outputs are present, the output1 is connected to PortA which is displays the input image whereas output2 is connected to portB which displays the skin detection output. As the name itself indicates automatic skin detection so output2 displays a skin detected image without being reset. The FPGA vertex2 pro Board there are four switches available that are used to see the different outputs in which switch1 is for reset the program, switch2 for YUV output image, switch 3 for edge detection image and switch4 for image enhancement output image on output2 screen in addition to that there are five more dip switches to perform the same operation. The user constraint file is used to assign the pins to this switches and in which 8 pins are used for R color, 8 pins for G color, 8 pins for B color and two extra pins for horizontal and vertical swings that is

for rows and column and another two pins for clock and reset.

Since the frequency of monitor is 50MHz and frequency of FPGA vertex 2 pro kit is 100MHz which are different from one another so we are using DCM(digital clock manager) so as to synchronize FPGA kit and monitor frequency by dividing the FPGA kit frequency by 2.

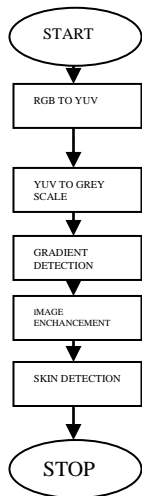


Figure 5: The skin detection algorithm

VI. EXPERIMENTAL RESULTS AND DISCUSSION

The designed automated skin detection algorithm which is implemented on the FPGA vertex 2 pro board gives directly the skin detected output image on the monitor screen which is obtained from the RGB color model. The RGB color model as an input image is converted to YUV color model and the output of the YUV color model image with the comparison of the input image is shown in figure 6.

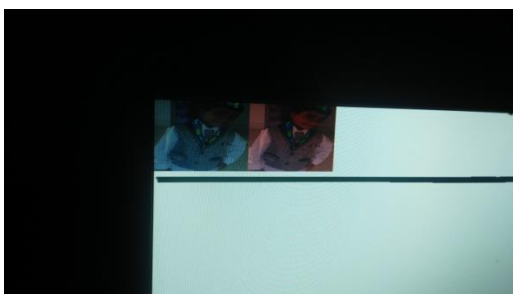


Figure 6: The YUV color model image

The YUV color model is converted to grey scale model and to this model skin detection algorithm is applied in order to detect the skin part in an image. The grey scale image contains some noise in it is in order to remove the background noise inversion is performed and the following result obtained is shown in figure 6.1.



Figure 6.1: The Inverted image

After removing the noise gradient detection is performed on the grey scale model to detect the edges of the image. The edge detection output image obtained using gradient detection is shown in figure 6.2.



Figure 6.2: The Edge detection output image

In order to improve the image quality image enhancement is performed using median filter and the image enhancement output is shown in figure 6.3

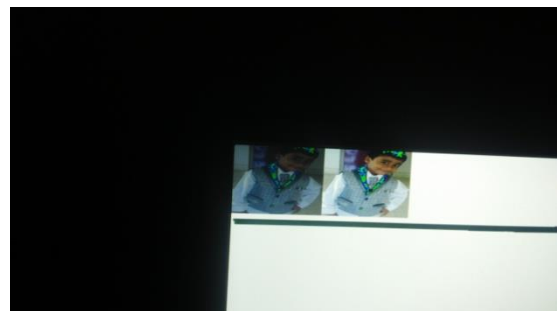


Figure 6.3: The image enhancement output

Finally after this all steps of skin detection algorithm, the final output image obtained is the skin detected output image is shown in figure 6.4.

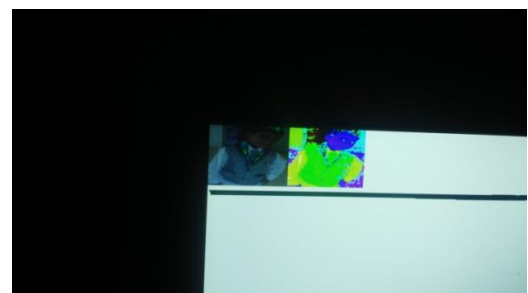


Figure 6.4: The skin detection output image

VII. CONCLUSION

In order to detect the skin part in a colored images based on skin detection algorithm of RGB color model with good accuracy requires lots of efforts. It has been employed in conjunction with skin color segmentation. As a result the occurrence of false negative has been greatly reduced. The work presented in this project report has been implemented on the FPGA vertex 2 pro board using Xilinx ISE 10.1 version software. Rigorous testing needs to be done with selecting different values of grid size and the threshold. We plan to automate the selection of grid size and threshold value based on the amount of skin color present in the image. The testing will be done using some standard datasets to compare our result with prevailing skin detection techniques. False detection is also low and this shows that it is able to distinguish between actual skin and background color. It is also seen that our algorithm is capable of classifying skin region in complex colored image. The robustness of the algorithm against the variance of illumination, focus and scales has been checked for a number of sample images. The result obtained is with minimum power dissipation and area.

This skin detection algorithm can be further improved with high accuracy and can improve the quality of the skin part image and further delay can be reduced.

REFERENCES

1. B. K. Harsha, M. L. J. Shruthi, "Non-Parametric Histogram Based Skin Modeling For Skin Detection", IEEE, 978-1-4799-1597-2/13, 2013.
2. S Nadimi, B Bhanu, Physical models for moving shadow and object detection in video. IEEE Trans. Pattern Anal. Mach. Intel. 26(8), 1079–1087 (2004)
3. B Ni, AA Kassim, S Winkler, A hybrid framework for 3D human motion tracking. IEEE Trans. Circuits Sys. Video Technol. 18(8), 1075–1084 (2008).
4. V. Vezhnevets, V. Sazonov and A. Andreeva, "A survey on pixel-based skin color detection techniques", GRAPHICON03, (2003),.
5. P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods", Pattern recognition, vol. 40, no. 3, (2007),
6. Amit Kumar and Shivani Malhotra, " Pixel-Based Skin Color Classifier", ISSN: 2005-4254, International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.8, No.7 (2015).
7. Zhu, Q., Cheng, K.-T., Wu, C.-T., Wu, Y.-L. 2004. Adaptive learning of an accurate skin-color model. In: Proceedings of IEEE FG, ISBN 0-7695-21223.
8. Tsekeridou, S., Pitas, I. 1998. Facial feature extraction in frontal views using biometric analogies. In: Proceedings of EUSIPCO.
9. Moghaddam, B., & Pentland, A. (1997). Probabilistic visual learning for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence.