EAST WEST UNIVERSITY

UNDERGRADUATE PROJECT REPORT

ON

An Efficient Algorithm to Detect the Skin Pixel Based on its RGB Values

Submitted By:

Tasnim Hosen

Yeasir Arafat

Supervised By:

Md. Shamsujjoha

Senior Lecturer

Department of Computer Science and Engineering,

East West University

Dhaka, Bangladesh

A project submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering to the Department of Computer Science and Engineering

January, 2016

DECLARATION BY CANDIDATES

We hereby declare that this project entitle "An Efficient Algorithm to Detect the Skin Pixel Based on its RGB Values" is our own work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

This original work submitted to the "East West University". The work was done under the guidance of "Md. Shamsujjoha, Senior Lecturer, East West University".

Signature of the candidates

....	....
Tasnim Hosen	Yeasir Arafat

ACCEPTANCE

This project entitle “An Efficient Algorithm to Detect the Skin Pixel Based on its RGB Values” is submitted by Tasnim Hosen, ID No: 2011-2-60-030 and Yeasir Arafat, ID No: 2011-2-60-022 to the Department of Computer Science and Engineering, East West University, Dhaka-1212, Bangladesh is accepted as satisfactory for partial fulfillment of the requirement for the degree of Bachelor of Computer Science and Engineering on Jan 2016.

Supervisor: Md. Shamsujjoha, Senior Lecturer

Signature: ……………………………..

Date: …………………………………..

Chairperson: Dr. Shamim Hasnat Ripon, Associate Professor

Signature: ……………………………..

Date: …………………………………..

Acknowledgment

At first, we are thankful to almighty for giving us the ability to do the work. We are also thankful to many individual who’s giving us the support. We are highly grateful to our supervisor Md. Shamsujjoha for his valuable guidance and regular supervision and also providing valuable information without this we cannot finish our work.

Besides our respective supervisor we would like to thank Dr. Taskeed Jabid and other faculty member of the Computer Science and Engineering Department for their support.

Abstract

This thesis work represents an efficient RGB pixel based skin color detection technique. To find identify human and surveillance skin detection is necessary. 2D color image classified as RGB color histogram and using this histogram create some RGB condition to find the skin color of human. These methods determine skin color of human with accuracy up to 91.63% which is better than existing techniques. The proposed method will be applicable for any 2D color image.

Contents

List of Figure

List of Table

Chapter 1

Introduction

At present, skin detection is widely use application in computer system. Our works based on the RGB numerical values. RGB values caries the information of skin and non skin vale. We use this RGB value to find the skin color. Using RBG color histogram we solve skin detection problem. We developed matching algorithm to find the accuracy of our thesis work to compare with other. Lots of work have done in previous, to find the skin image form a color image. To solve the skin detection lots of method have been applied like RGB, HSV, FREQUENCY, YCrCb, TSL [1],[2],[3],[4].In computer lots work have been done to solve skin detection problem using RGB[1][5][6][7].

In this chapter, we discuss about the skin detection problem using RGB numerical value. We discuss about why we use RGB numerical value to solve skin detection problem in section 1.1. The motivation of our work discuss in section 1.2. The objective of our work discuss in section 1.3. The overview of our work discuss in section 1.4.Our contribution to solve skin detection problem will discuss in section 1.5. The summary of our work discuss in section 1.6.

1.1Purpose of Skin Detection

We have been used establish algorithm to solve skin detection problem. But add some new RGB condition to increase the accuracy of skin detection. That gives us the output. It is easy to implement then other method. We find this condition by analyzing the RGB value of 500 color image. By using our method we easily describe the skin color in a color image. Because, it is base on RGB value of color images. We easily describe the statistic of RGB color image.

1.2 Motivation

Now a day the world is dominated by technology. Technologies are increasing day by day. Many difficult problems are solving by machine learning and image processing technique like face detection, facial expression recognition, Gender recognition and gesture recognition. It is easy way to find the face recognition from 2D color image. Our method is too much easy for this reason we are motivated to do this work. There are lots of applications of face recognition like surveillance, criminal investigation, homeland and private security, and easy people tagging these types of application motivated us to do the work. These are the common problems now a day. To solve these types of problem we need face recognition. For this reason we are motivated to do this work.

Without this, our supervisor Md. Shamsujjoha and our teacher Dr. Tasked Jabid motivated us to develop a work in that is easily understandable.

It is challenging to find skin color from 2D color image using RGB value. Because there are lots of value in 2D color image skin value, non skin value, background image value. From this, it is difficult to find skin color. We use some condition to find the skin color from 2D color image. Find out this condition is challenging. That is motivated us.

We did not use any establish algorithm to solve skin detection problem. We used RGB numerical value to solve skin detection problem. Then we used matching algorithm to find the accuracy. Then we compare with other work. Sometimes its gives us better accuracy than other. Sometimes it's not better than other.

1.3 Objective

The objective of our work gives better performance for skin color detection. We use 2D color image to find hit count value of R, G, and B in particular pixel [0 to 255]. Using this value we create histogram of R_ value count, G_ value count and B_value count respectively. Using this we find the condition for skin detection. Then we used matching algorithm to find the accuracy.

Using this accuracy we compare our work with other work. There are more than 1000 color image we used to find our work accuracy. We use different types of 2D color image like bright image, dark image.

1.4 Our Contribution

Our contribution for skin detection is increasing the accuracy of skin detection. For increasing the accuracy we proposed some new method. This is also our contribution. Then we developed matching algorithm to find the accuracy.

In our method using RGB color histogram we collect information from 2D color image. Using this histogram we find the RGB condition of skin color pixel. All work done by using MATLAB.

1.5 Outline

In chapter 2 we describe the previous work on skin detection and different type of method of skin detection technique.

In chapter 3 we describe our proposed methodologies on skin detection.

In chapter 4 we describe our experimental result. And we also analyze the performance of our proposed system.

In chapter 5 we conclude our research work by mentioning some limitations of our work.

Chapter 2

Existing Work Review

In past decades, lots of work has been done for face detection and tracking. Many heuristic and pattern reorganization based method have been proposed to solve the face detection problem. By using this method they are achieving their goal. Some of methods are very good some are not good enough. To solve skin color detection problem, they have been used color based feature of 2D color image. Because, 2D color image pixel carries the information of skin color and non skin color. By separating skin and non skin pixel easily find skin color.

When we have been used skin color as a feature for face detection. We faced three problems.

1. What colorspace we used.
2. How exactly the skin color distribution should be modeled.
3. What will be the way of processing of color segmentation result for face detection?

Pixel-based skin detection methods, separate pixel as a skin or non- skin pixel independently. There are lots of techniques to find skin color and non skin color pixel like region-based method [8][9][10], pixel-based skin detection[2], two non- parametric skin modeling[11], Bayes skin probability map[11]. By reading this we collected idea about skin pixel separation.

In this chapter, we will be discussing about previous face detection techniques. In section 2.1 we discuss about colorspaces used for skin detection. In section 2.2 we discuss about skin modeling. In section 2.3 we discuss about face detection using RGB. In section 2.4 we discuss about the summary of chapter 2.

2.1 Colorspaces used for skin detection

There are lots of colorspace. Properties of this colorspaces are different. In computer graphics and video signal transmission standards introduce this colorspace. This colorspace have been used to solve the skin detection problem. In this section we will discuss about this colorspace that have been used for skin detection.

2.1.1 RGB

An RGB color space is any additive color space based on the RGB color model. A particular RGB color space is defined by the three chromaticities of the red, green, and blue additive primaries, and can produce any chromaticity that is the triangle defined by those primary colors. It is one of the most widely used colorspace for processing and storing of 2D color image. To solve skin detection problem most of the time RGB colorspace have been used [12],[13].

2.1.2 Normalized value of RGB

Normalized RGB is a representation of RGB colorspace. Normalized RGB is created from the GB. Using below equation:

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

The sum of the three normalized components is known ($r + g + b = 1$), the third component does not hold any important information and can be absent, sinking the space dimensionality. The remaining components are often called "pure colors", for the dependence of r and g on the brightness of the source RGB color is diminished by the normalization. A remarkable property of this representation is that for matte surfaces, while ignoring ambient light, normalized RGB is invariant (under certain method) to changes of surface orientation relatively to the light source

[14]. This, together with the transformation simplicity helped this colorspace to gain popularity among the researchers [4], [2], [15], [16], [9].

2.1.3 HSI, HSV, HSL – Hue Saturation Intensity

When researcher needs to specify color properties numerically, they were introduced Hue-Saturation based colorspaces. Hue- Saturation based colorspace is customizes colorspace. Because many color used in this colorspace. In RGB colorspace red, green, blue in fixed. But in this colorspace many color like red, green, purple, yellow are used.

Hue defines the dominant color (such as red, green, purple and yellow) of an area; saturation measures the colorfulness of an area in proportion to its brightness [17]. The “intensity”, “lightness” or “value” is related to the color luminance. The intuitiveness of the colorspace components and explicit discrimination between luminance and chrominance properties made these colorspaces popular in the works on skin color segmentation [2], [18], [19], [20], [21]. Several interesting properties of Hue were noted in [14]: it is invariant to highlights at white light sources, and also, for matte surfaces, to ambient light and surface orientation relative to the light source.

Using below equation we find out the HSV value.

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

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

$$V = 1/3(R+G+B)$$

There are two different way to represent the Hue-saturation computation. One is using log opponent values was introduced [9], where additional logarithmic transformation of RGB values amied to reduce the dependence of chrominance on the illumination level. Other way, is the polar coordinate system of Hue-Saturation spaces.

2.1.4 TSL

TSL (Tint, Saturation, lightness) is a transformation of the normalize RGB into more instinctive values, close to hue saturation in their meaning.

S= $[9/5(r'^2 + g'^2)]^{1/2}$

$$T = \begin{cases} \frac{\arctan\left(\frac{r'}{g'}\right)}{2\pi} + \frac{1}{4}, g' > 0 \\ \frac{\arctan\left(\frac{r'}{g'}\right)}{2\pi} + \frac{3}{4}, g' < 0 \\ 0, g' = 0 \end{cases}$$

L= 0.299R + 0.587G + 0.114B

Where r'= r-1/3, g'=g-1/3 it's come from [4]. They argue that normalized TSL space is superior to other colorspaces.

2.1.5 YCrCb

YCrCb is an encoded nonlinear RGB signal, commonly used by European television studios and for image compression work. Color is represented by luma (which is luminance, computed from nonlinear RGB [17]), constructed as a weighted sum of the RGB values, and two color difference values Cr and Cb that are formed by subtracting luma from RGB red and blue components.

$$Y = 0.299R + 0.587G + 0.114B$$
$$C_r = R - Y$$
$$C_b = B - Y$$

The transformation simplicity and explicit separation of luminance and chrominance components makes this colorspace attractive for skin color modeling [22], [2] [23], [24], [25], [26].

2.1.6 Other colorspaces

There are other colorspace like YCrCb, YUV, YIQ, and CIE-xyz. This colorspaces are used to find the skin detection. This colorspaces are different from one to another. Using this colorspaces many work have been done. The properties of this colorspaces are different. We are working with RGB colorspace in our thesis. For our thesis we used RGB colorspace.

2.2 Skin detection modeling

Now, the next step is to build up a system. By using this system we find the skin color. For this we use previous colospace pixel data. There are lots of method have been developed in previous to solve skin detection problem. In this section we will discuss about this method that have been used for skin detection.

2.2.1 Explicitly define skin region

This is very simple method for skin detection. Using this condition we easily find skin color from 2D image. In this method have been used some RGB condition to find the skin cluster. Using this condition they find the skin color. The condition given below:

(R, G, B) is classified as skin if:

$R > 95$ and $G > 40$ and $B > 20$ and $\max\{R, G, B\} - \min\{R, G, B\} > 15$ and $|R-G| > 15$ and $R > G$ and $R > B$

For the simplicity of this method many researchers working with this method [27], [25], [28], [21].The great advantage of this method is simplicity of skin detection rules that leads to construction of a very rapid classifier. But it is difficult to achieving high recognition rates with this method are the need to find both good colorspace and adequate decision rules empirically. To solve this problem many researchers have been proposed many methods. One of the methods that used normalizes RGB colorspace [29].The researchers start with a normalized RGB space and then apply a constructive induction algorithm (see[29] for details) to create a number of new sets of three attributes being a superposition of r, g, b and a constant 1/3, constructed by basic arithmetic operations. A decision rule, similar to (10) that achieve the best possible recognition is estimated for each set of attributes. The authors prohibit construction of too complex rules, which helps avoiding data over-fitting that is possible in case of lack of training set representativeness.

Using this method many researchers find skin color. But the accuracy of the method is not good enough.

2.2.2 Enhanced Skin Color Classifier Using RGB Ratio Model

In this method they are develop previous paper of kovac [1]. Then they are adding some condition and they are get better accuracy. In this paper they are add two more condition. That condition given below:

(R, G, B) is classified as skin if:

$R > 95$ and $G > 40$ and $B > 20$ and $\max\{R, G, B\} - \min\{R, G, B\} > 15$ and $|R-G| > 15$ and $R > G$ and $R > B$

New two conditions that they are add is:

$0.0 <= \frac{R-G}{R+G} <= 0.5$ and $\frac{B}{R+G} <= 0.5$

They are adding this condition because previous method could not detected dark and yellow color. Previous method problem is the range of R-value is from 4 to 255, the range of B-value is from 1 to 200, and the range of G value is from 1 to 255. It shows some agreement with Kovac's rule which is that a pixel is considered as skin pixel if R > G and R > B and B < 200. However, this rule is still unable to detect some dark skin colour and yellow like colour which is detected as skin colour.

By considering the aforementioned issues, a new method has been developed based on painting colour concepts and colour ratio, which is based on colours mixing to produce new colour. This means some ratio of RGB has been taken to develop a new skin colour rule. The sum of R, G,B and distance between R and G, and B values were observed based on ratio. The histogram of ratio of difference between R and G over the sum of R and G, and the ratio of B over sum of R and G are plotted from skin pixel of training dataset. Then they are adding two conditions.

$0.0 <= \frac{R-G}{R+G} <= 0.5$ and $\frac{B}{R+G} <= 0.5$

But in this paper adding this they are overcome previous paper problems. After adding this, this method gives better accuracy than previous paper.

2.3 Nonparametric Skin detection model

Nonparametric skin detection model is to estimate skin color distribution from the training data without deriving an explicit model of the skin color. Using this key idea nonparametric method find the skin color. It is very useful method for face reorganization. The result of these methods sometimes is referred to as construction of Skin Probability Map (SPM) [12], [29] assigning a probability value to each point of a discredited colorspace. In this section we discuss about this. Two clear advantages of the non-parametric methods are i. they are fast in training and usage and ii. They are theoretically independent to the shape of skin distribution. The disadvantages are much storage space required and inability to interpolate or generalize the training data.

2.3.1 Normalized lookup table

Normalized lookup table worked on histogram of 2D and 3D color image. The histogram of 2D and 3D image called normalized lookup table (LUT). The colorspace (usually, the chrominance plane only) is quantized into a number of bins, each corresponding to particular range of color component value pairs (in 2D case) or triads (in 3D case). Bins of histogram stores the number of times this particular color occurred in the training skin images. After training, the histogram counts are normalized, converting histogram values to discrete probability distribution. Many researchers work with this algorithm [2],[15],[20],[19],[30].

$P_{skin}(c) = \frac{skin(c)}{Norm}$

In this equation skin value is skin(c). From histogram bin they get skin(c) value. Then they normalize this value. The normalized values of the lookup table bins constitute the likelihood that the corresponding colors will correspond to skin.

2.3.2 Bayes Classifier

To find skin color, Bayes classifier use normalized lookup table (LUT) histogram bins. In this method, skin value computed from skin and non skin pixel value. Using this value they are computed skin pixel.

The value of Pskin(c) computed in (LUT) is actually a conditional probability P(c|skin) - a probability of observing color c, knowing that we see a skin pixel. A more appropriate measure for skin detection would be P(skin|c) - a probability of observing skin, given a concrete c color value. To compute this probability, the Bayes rule is used:

$$P(skin|c) = \frac{P(c|skin)P(skin)}{P(c|skin)P(skin)+P(c|\neg skin)P(\neg skin)}$$

P(c|skin) and P(c|¬skin) are directly computed from skin and non-skin color histograms of normalized lookup table(LUT). The prior probabilities P(skin) and P(¬skin) can also be estimated from the overall number of skin and non-skin samples in the training set [13], [2], [26]. An inequality P(skin|c) >=Θ, where Θ is a threshold value, can be used as a skin detection rule [13]. Receiver operating characteristics curve [31] shows the relationship between correct detections and false detections for a classification rule as a function of the detection threshold. For computing threshold in Bayes classifier used below equation:

$$\frac{\mathrm{P(c|skin)}}{\mathrm{P(c|\neg skin)}} > \text{Threshold}$$

Where threshold computed from below equation:

$$\text{Threshold} = \mathrm{K} * \frac{1-P(skin)}{P(skin)}$$

This shows, why the choice of prior probabilities does not affect the overall detector behavior for any prior probability P(skin) it is possible to choose the appropriate value of K, that gives the same detection threshold Θ. It is also clear, that maximum likelihood (ML) and maximum a posteriori (MAP) Bayes classification rules compared in [2] are equivalent to with different threshold values.

Using Bayes we can easily find the face region. Bayes classifier gives higher accuracy then other skin detection work. It is a good method for skin detection.

2.3.3 Self Organizing Map

Self-Organizing Map (or SOM), devised by Kohonen in 80's is now one of the most popular types of unsupervised artificial neural network. In [4] a SOM-based skin detector was proposed. Two SOM's - skin-only and skin + non-skin were trained from a set of about 500 manually labeled images. The detectors performance was tested on the authors training/test images set and famous Compaq skin database [13]. Several colorspaces (normalized RGB, Hue-Saturation, cartesian Hue-Saturation and chrominance plane of TSL) were tested with SOM detector. The results have shown that SOM skin detectors do not exhibit vivid performance change when using different colorspaces. The SOM performance on the authors dataset is marginally better than Gaussian mixture model, while for the Compaq database the SOM performance is inferior to the RGB histograms used in [13]. The authors stress out that SOM method needs considerably less resource than histogram and mixture models and is efficiently implemented for run-time applications by the means of SOM hardware.

It has mainly been used to find patterns in and classify high dimensional data, although it works equally as well with low dimensional data. The basic SOM consists of a 2-dimensional lattice *L* of neurons. Each neuron η_i € L has an associated codebook vector μ_i € R^n . In what follows *n*=2 although in other applications *n* is often much larger. The lattice is either rectangular or hexagonal, with connections within *L* determining the neighborhoods of a given neuron. Training the SOM involves first randomly initializing all the codebook vectors and then sequentially presenting each training sample. We first fix a metric ton *L*, usually Euclidean or Manhattan.

Each $V \in R^n$ is presented as an input vector to all neurons in the network, and twinning neuron η_c with codebook vector μ_c is determined so that

$$\| V - \mu_c \| = \min_{1 \leq i \leq k} \| V - \mu_i \|$$

Where, *k* is the number of neurons in the network. The neurons in specific neighborhoods of the winning neuron then have their codebook vectors adjusted to be closer to the input vector according to a parameterized learning function. As training progresses, the learning rate and the size of the affected neighborhoods are decreased, and the lattice gradually forms a topologically ordered mapping (or feature map) of the training data. If necessary, a calibration phase then takes place, where labeled training data is sequentially presented to the SOM and, each time, the data label and index of the winning neuron recorded. Each neuron is then assigned the label of the type for which it 'fired' the most. Classification can then take place by presenting data and labeling with the label of the winning neuron each time.

2.4 Parametric skin detection mode

The most popular histogram-based non-parametric skin models require much storage space and their performance directly depends on the representativeness of the training images set. The need for more compact skin model representation for certain applications along with ability to generalize and interpolate the training data stimulates the development of parametric skin distribution models.

2.4.1 Single Gaussian

Results reported in the literature indicate that a single bivariate gaussian probability density function (pdf) can be used successfully as a model for the skin color, even when multiple ethnic groups are considered [32],[33],[34],[35]. The model can be obtained via the maximum likelihood criterion, which looks for the set of parameters (mean vector and covariance matrix) that maximizes the likelihood function. The likelihood function for a multivariate gaussian pdf has a single maximum, and the estimates JL and 1: for the mean vector and the covariance matrix are obtained analytically and have well-known values given by the equation[36].

$$\mu = \frac{1}{n} \sum_{k=1}^{n} x_k$$

$$\Sigma = \frac{1}{n} \sum_{k=1}^{n} (x_k - \mu)(x_k - \mu)^t$$

Where, JL is the estimated mean vector, sum is the estimated covariance matrix, n is the number of observations in the sample set, and xk is the k th observation. The resulting gaussian pdf that fits the data is then [36].

$$P(x|\ \mu,\Sigma\ ) = \frac{1}{(2\pi)^{1/2} (\det(\Sigma\ ))^{1/2}} * \exp(-\frac{1}{2} D^2)$$

Where

$$D^2 = (x - \mu) \Sigma^{-1} (x - \mu)^t$$

Is the square Mahalanobis distance and d is the dimensionality of the gaussian function (d == 2 in our particular case). Figure 2 shows a plot of the function estimated from the sample data set with the application of this equation.

2.4.2 Mixture of Gaussians

A more sophisticated model, capable of describing complex-shaped distributions is the Gaussian mixture model. It is the generalization of the single Gaussian, the pdf in this case is:

$$p(c|skin) = \sum_{i=1}^{k} \pi_i \cdot p_i(c|skin)$$

In this equation *k* is the number of mixture components, p*i* are the mixing parameters, obeying the normalization constraint$\sum_{1}^{k} \pi i$ =1 andp*i* = 1 and pi(*c*|*skin*) are Gaussian pdfs, each with its own mean and covariance matrix. Model training is performed with a well-known iterative technique called the Expectation Maximization (EM) algorithm, which assumes the number of components k to be known beforehand. The details of training Gaussian mixture model with EM can be found, for example in [9], [11]. The classification with a Gaussian mixture model is done by comparing the $p(c|skin)$ value to some threshold. The choice of the components number k is important here. The model needs to explain the training data reasonably well with the given model on one hand, and avoid data over-fitting on the other. The number of components used by different researchers varies significantly from 2 [11] to 16 [13]. A bootstrap test for justification of k =2 hypothesis was performed in [9], in [11] k = 8 was chosen as a good compromise between the accuracy of estimation of the true distributions and the computational load for thresholding [16], [18] have also used Gaussian mixture models.

2.4.3 Multiple Gaussian Clusters

Approximation of skin color cluster with three 3D Gaussians in YCbCr space is described in [22]. A variant of *k*-means clustering algorithm for Gaussian clusters performs the model training. The pixel is classified as skin, if the Mahalanobis distance from the c color vector to the closest model cluster center is below a pre-defined threshold.

2.4.4 Elliptic Boundary Model

By examining skin and non-skin distributions in several colorspaces Lee and Yoo [37] have concluded that skin color cluster, being approximately elliptic in shape is not well enough approximated by the single Gaussian model. Due to asymmetry of the skin cluster with respect to its density peak, usage of the symmetric Gaussian model leads to high false positives rate. They propose an alternative they call an elliptical boundary model which is equally fast and simple in training and evaluation as the single Gaussian model and gives superior detection results on the Compaq database [13] compared both to single and mixture of Gaussians . The elliptical boundary model is defined by below equation.

$$\Phi(c) = (c - \phi)^T \ \Lambda^{-1} (c - \phi)$$

The model training procedure has two steps - first, up to 5% of the training color samples with low frequency are eliminated to remove noise and negligible data. Then, model parameters (Φ and L) are estimated by

$$\Phi = \frac{1}{n}\sum_{i=1}^{n} c_i \;; \qquad \Lambda = \frac{1}{N}\sum_{i=1}^{n} f_i \,.\, (c_i - \mu)(c_i - \mu)$$

$$\mu = \frac{1}{N}\sum_{i=1}^{n} f_i\, c_i; \qquad N = \sum_{i=1}^{n} f_i$$

Where *n* is the total number of distinctive training color vectors *ci* of the training skin pixel set (not the total samples number!), and *fi* is the number of skin samples of color vector *ci*. Pixel with color *c* is classified as skin in case when Φ(*c*) < θ, where q is a threshold value. The authors claim that their model approximates the skin cluster better, because the data skew does not affect the model centroid Φ calculation.

All described parametric methods operate in colorspace chrominance plane, ignoring the luminance information. Of course, since an explicit distribution model is used, a question of model validation arises. Obviously, the goodness of fit is more dependent on the distribution shape, and therefore colorspace used, for parametric than for non-parametric skin models. This is clearly visible in the results of [11], [37], where the model performance varies significantly from colorspace to colorspace. Only several authors have included theoretical justification for the validity of models they used. [38] has shown that skin color distribution of a single person under fixed lighting conditions in normalized RGB space obeys Gaussian distribution. [9] have justified the hypotheses of skin data normality in CIELuv space and validity of two-component Gaussian mixture model by statistical tests. Others relied whether on the observation of nearly elliptic shape of the skin chrominances cluster in the colorspace they used (to employ single Gaussian model or similar), or its clearly non-elliptical shape (to employ mixture of Gaussians or several Gaussian clusters) with further model performance evaluation as the acceptance criterion [11], [37], [18].

2.5 Dynamic skin detection model

A family of skin modelling methods was designed and tuned specifically for skin detection during face tracking. This task makes skin detection different from the static images analysis in several aspects. First, in principle, the skin model can be less general (more specific) - i.e. tuned for one concrete person, camera or lighting. Second, initialization stage is possible, when the face region is discriminated from background by different classifier or manually. This gives a possibility to obtain skin classification model that is optimal for the given conditions (person, camera, lighting, background). Since there is no need for model generality, it is possible to reach higher skin detection rates with low false positives with this specific model, than with general skin color models, intended to classify skin in totally unconstrained images set [13]. On the other hand, skin color distribution can vary with time, along with lighting or camera white balance change, so the model should be able to update itself to match the changing conditions. Also, model training and classification time becomes extremely important here, for the skin detection system must work at real-time, consuming little computing power.

To summarize the most important properties of skin color model for face tracking: first, it should be fast in both training and classification and second, it should be able to update itself to changing conditions. Minding these aspects, many researches turn to simple parametric skin modelling - it is easily updated to distribution change, is acceptably fast (except for many-component mixture of Gaussians) and needs little storage space. The high false positives rate - a usual companion of parametric skin modelling, is less a problem here. The need for specific, not general skin color model permits achievement of good classification performance. Among non-parametric models, the histogram-based LUT is popular for face tracking tasks, thanks to its simplicity and high training and working speed.

A number of methods for skin color distribution recalculation were proposed: online Expectation Maximization [16], dynamic histograms [15], [39], [19], Gaussian distribution adaptation [9]. Several authors have investigated how the color of a single person should be modeled and how it varies with lighting change. The hypothesis of unit model Gaussian distribution of one person’s skin color under fixed lighting was justified in [9]. A special study on skin color change under different lighting conditions was made by [40] and [41].

Another promising method appeared recently, which is not included in this table, is automatic construction of a colorspace and an explicitly defined skin cluster in it [42], [43]. The authors have achieved results that outperform Bayes SPM classifier in RGB space for their dataset, giving significantly lower false positives rate (around 6% against 22%) and almost equal false negatives (around 5%).

2.6 Other Work

There are lots of work have been done in previous about skin detection like using frequency, Logical Binary Pattern.

2.7 Summery

In this chapter we discuss about the previous work about skin detection. We discuss about how they detect skin color. We discuss about the all of method previous. Then we develop our idea about skin detection.

Chapter 3

Description of our proposal work

In this chapter, we have described the computer vision based 2-D skin recognition system .Section

3.1 represented our process of our proposed work.

3.2 represented our algorithm

3.3 represented where we develop this algorithm

3.4 represented why we develop this algorithm

3.5 represented how we have compared the accuracy between our algorithms to other establish algorithm for finding skin values.

3.1 Process of Our Proposed Method

3.1.1 General Description

Skin value is very important to find out the human identity individually. There are so many algorithms to find out the skin value. But sometimes it more difficult to execute for others or more difficult to understand. But our algorithm is easier than others algorithm which is easy to execute. We can define skin value and non skin value from an image.

Here we generated our code using Mathlab7. It makes easy to find values which are needed to our experiment with low time complexity.

We had worked a 2-D images data set.

3.1.2 Find RGB value

In this section, we had found the RGB values of our data set which contains 120 images. Then we found out which pixel was how much hit. For example in Red, 122 number pixels are hited 425 times that means at that point for general image we can get skin. For Green, 211 number pixels are hited 321 times that means at that point for general image we can get skin. For Blue, 159 number pixels are hited 245 times that means at that point for general image we can get skin. Then we make a excel sheet to see that which pixel how much hited which helps to compare the probability which pixels are skin and which pixels are non skin.

3.1.3 Fixed skin range

When we found the r, g, b values and which pixels how much hit . After analysis the excel sheet we tried to fix the range which can help us to find out the skin value. We applied different types of range to fix the skin range.

For example

{

R<100 and G<20 and B <15

Then we can tell it skin pixels

)

3.2Our Algorithm

INPUT: RGB 2-D color image

OUTPUT: Skin image for original image

Begin :

for (i=0 TO row) **do**

    for (j= 0 TO colum) **do**

        if (satisfy RGB range condition) **then**

```
New set =[R,G,B]
    if (difference R&& G > limit ) then
        if (ration between G and (R+G) < limit ) then
            if (ration between  B and (R+G) < limit ) then
                        Skin image (i,j,white)
                    end if
                end if
            end if
        end if
    end for
end for
```

3.2.1 Explanation Algorithm to Detect Skin

1. Load an image;

2. Declare an array for R, G, B values which are not considered for finding skin value according to experimental sheet;

3. Make an image which contains zero values with row and Colum;

4. Then find R, G, B values from our original image with its co ordinates (x – axis and y –axis);

5. set our condition with range;

6. Set an new image according to skin value which was found after giving condition

3.2.2 Matching algorithm

Input: Skin image from data set and our applying method skin image

Output: Accuracy percentage respect to data set skin image

Begin:

```
    get co- ordinate =(find  data _set_image = =1)
     get co- ordinate =(find  data _set_image = =1)
            Count =0
          for     i=0 to numel  (x )  do
            if ~isempty(find   (matching "1 ") then
                  Count ++
              end if
        end for
return  [ count / numel] *100
```

Description:

1. Load a particular skin image from our data set and our new skin image;

2. Find out the value "1" from both images;

3. Check the similarity according to axis (x – axis and y –axis);

4. Then counted "1" which had been matched;

5. Total accuracy;

3.2.3 How to our algorithm described by an image

Step #1

Load an image from data set

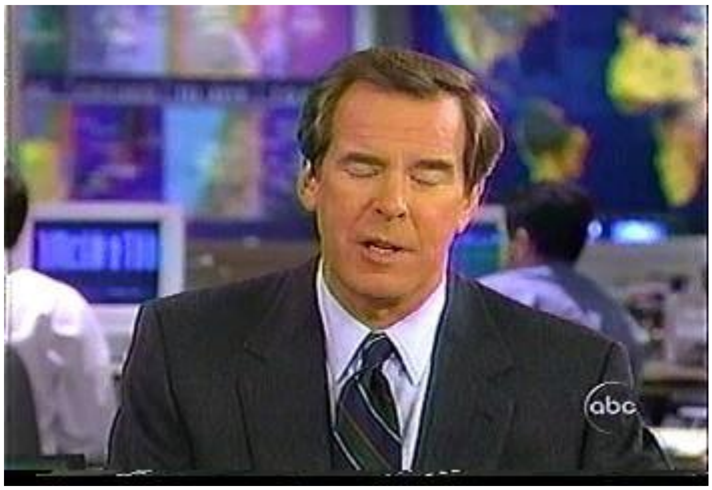

Figure 3.1: 2D color image

Step #2

Set it black

Figure 3.2: Black image of 2D color image

Step #3

Set skin value only

Figure 3.3: 2D color image applying our method

3.2.4 How to work our matching algorithm described by an (11 *10) matrix

Matrix 1

Table 3.1: Suppose it's a database skin image matrix

1	1	1	1	1	0	0	0	1	1
1	1	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0	0
0	1	0	1	1	1	1	1	1	0
0	1	1	0	1	1	1	1	0	1
0	1	0	1	1	1	1	1	0	0
1	1	1	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	1	1
0	1	0	0	1	1	0	1	0	0
0	0	0	0	0	1	1	1	1	1
0	0	1	1	0	1	1	0	0	0

Matrix 2

Table 3.2: Suppose it's our skin image matrix

0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0	0
0	1	0	1	1	1	1	1	1	0
0	1	1	0	1	1	1	1	0	1
0	1	0	1	1	1	0	0	0	0
1	1	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	1	1
0	1	0	0	1	1	0	1	0	0
0	0	0	0	0	1	1	1	1	1
0	0	1	1	0	0	0	0	0	0

Then our matching algorithm compare between two matrixes by axis. Where it's finding 1 with same co ordinate of two matrixes then it's counted. And find accuracy

3.3 Where we develop this algorithm

3.3.1 Previous algorithm condition

In this segment we have described why we developed our algorithm. Generally we developed some previous algorithm and give some new condition. At first we developed the The Kovac model [1], Swift's rule [6], Ghazali Osman [7] ,saleh [5] Here is those his algorithm

The Kovac model

Pixel is skin color pixel if:

Rule 1: $R > 95$ and $G > 15$ and $B > 20$ and

Rule 2: Max (R, G, B) – Min$(R, G, B) > 15$ and

Rule 3: $|R - G| > 15$ and

Rule 4: $R > G$ and $R > B$

Swift's rule:

Pixel is not skin colour pixel if:
$B > R, G < B, G > R, B < \frac{1}{4}R$ or $B > 200$

Ghazali Osman model:

He just added a new condition with the Kavoc algorithm.

The new condition is

$$0.0<((R-G)/(R+G))<= .5 \; \& \; (B/(R+G)<= .5)$$

Saleh model

Finally, a very simple rule was introduced by Saleh which consider only the value of *R* and *G*. This rule defines that a pixel is skin pixel when $R - G$ is greater than 20 and less than 80. That means the range of *R* is from 21 to 255, while the range of *G* is from 0 to 234. This rule Does not consider a present of *B*-value which contributed to the whitish color. This rule is also unable to detect dark skin color or skin cover under shadow, and yellow like color and redder colour problems which is detected as skin pixel.

3.3.2 Our condition

```
R > 100 && G > 20 && B>20
        V=[R,G,B];
          ((Mean (v) - min (v)) > 15)
            (abs(R-G) > 15 && R > G && R > B)
              if (abs(R-G)<=120 && B<R && G<R)
              (0.0<((R-G)/(R+G))<= .5 & (B/(R+G)<= .5))
                (R~=Red_not_count_pixel  )
                  (G~=Green_not_count_pixel  )
                  (B~=Blue_not_count_pixel  )
```

3.4 Why we develop this algorithm

3.4.1 Range changed

In our algorithm we at first changed the range of R, G, B. we give a condition like this R > 100 && G > 20 && B>20. Because we saw from our training sheet result that for RED in below 100 pixels and for Green in below 20 and for Blue in below 20 every pixels are hit but its total value are not significant .That means those pixels possibility is low to get found a skin values .so we modified it .Another interesting part for this range here we can see Green and Blue both are also greater than 20 .The reason is from our experimental result, we can see the change of pixel value and hit of pixels values are nearly same . If we fix this range then we can maximize our skin pixel values

3.4.2 Condition changed

In this condition, we use mean value instead of max value. If we use max value of RGB then here create some problems .Because when use max function to find the maximum value for the RGB and subtract its minimum value then there are maximum range value included in the skin value pixels which are not found generally as skin pixel in different image. But when we use mean or average value function then what value we get that difference of minimum value is less than the previous condition. That's why we get specific value of skin which has maximum counted as skin pixels value.

3.4.3 Our proposed rule

We developed the previous condition and also give new rules to find skin value which has a significant effect on our accuracy. Our proposed rules are

(R~=Red_not_count_pixel)

(G~=Green_not_count_pixel)

(B~=Blue_not_count_pixel)

Red_not_count_pixel : This is an array which contains

Red_not_count_pixel =[232,233];

That's pixel value.

Green_not_count_pixel : This is an array which contains

Green_not_count_pixel=[41,42,43];

Blue_not_count_pixel: This is an array which contains

Blue_not_count_pixel=[24,25,26, 241,242,243,244,245,246,247];

We remove this values from our RGB array .Because we saw that the after fix the range that values insert our skin value array though it has no significant effect our skin image .But that values takes down our accuracy blew the previous accuracy .for that value show no skin color as a skin color .For that our algorithm significantly affected and didn't find out the skin image preciously. For that we created that condition and add this condition in our code and we get a very good result and good accuracy also which very much good than other skin image accuracy.

3.4 Compute the accuracy

At first we find out the "1" from our data set image and the find out the "1" from our new skin image. Then its compare with the axis. Where get the similar value then its store an array which is automatic update when its gets the similar values with similar axis. Generally there are matching algorithm to find out a particular skin image accuracy .By this algorithm we find out the true positive and true negative number .It also can find out false positive number and false negative number .Anyone who want to find out the accuracy an binary image can use our algorithm . It helps a lot to anyone .Because it's very easy logic and its time complexity is very low which is non-considerable.

Chapter 4

Experimental Performance and Result Analysis

In this chapter we will discuss about experimental performance and result analysis. In section 4.1 we will discuss about the image and image dataset that we use in our work. In section 4.2 we will discuss about count RGB value from 2D color image and histogram of this value. In section 4.3 we will discuss about our work result analysis. In section 4.4 we will discuss about the performance and compare with other method.

4.1 Image and dataset

For our work we use 2D color image. In below figure we show the information of an image that we use in out thesis.

```
>> whos
  Name              Size                            Bytes  Class

  B                  1x1                                1  uint8 array
  G                  1x1                                1  uint8 array
  I                180x150x3                        81000  uint8 array
  R                  1x1                                1  uint8 array
  final_image      180x150                         216000  double array
  i                  1x1                                8  double array
  j                  1x1                                8  double array
  v                  1x3                                3  uint8 array

Grand total is 108008 elements using 297022 bytes

>> |
```

Figure 4.1: Information that show all details

Fig describes the size, class and space for a single image that we use for our thesis. It all matrix size we use for single image. It also describes class and space of this image.

For our method we use “UChile” database data. Using this database image we complete our work.

4.2 RGB value count and Histogram

From 127, 2D color image we collect RGB hits value using MATLAB7. Using this value we generated histogram.

4.2.1 R_value Count and Histogram

Using matlab we count R pixel hits. Then we create histogram to find our condition.

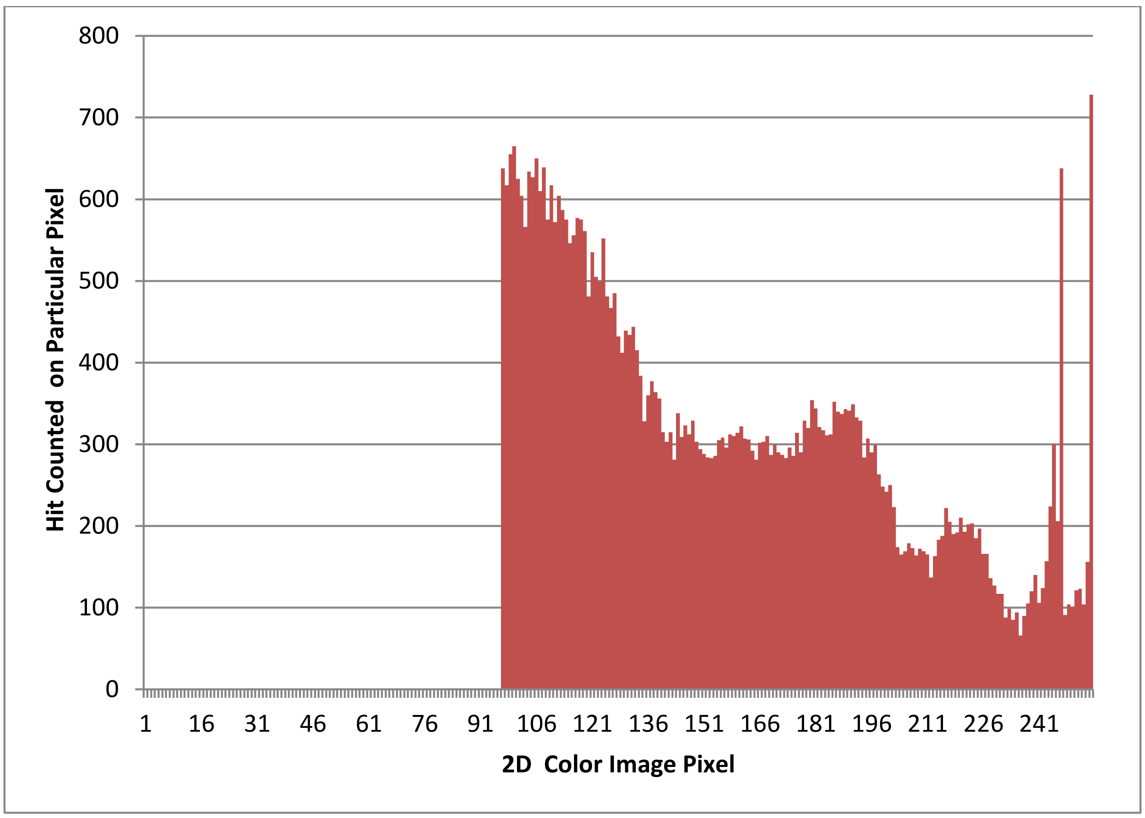

Figure 4.2: Histogram of R pixel hit count

In this histogram we see that hits in pixel 1 to 96 is zero. For this we skip this pixel and the hit in pixel 236 is very low. So we also skip this pixel. Then we develop an algorithm according this.

4.2.2 G_value Count and Histogram

Using matlab we count G pixel hits. Then we create histogram to find our condition

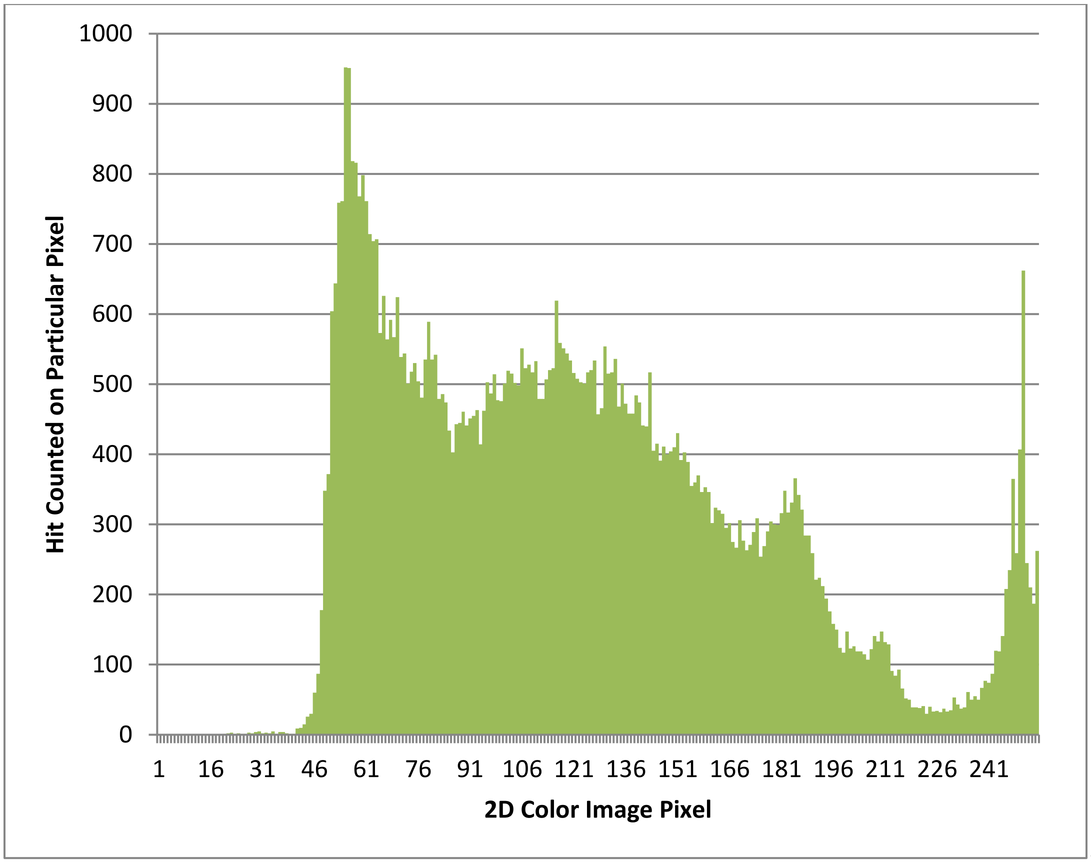

Figure 4.3: Histogram of G pixel hit count

In this histogram we see that hits in pixel 1 to 40 is zero. For this we skip this pixel and the hit in pixel 41, 42, 43 is very low. So we also skip this pixel. Then we develop an algorithm according this.

4.2.1 B_value Count and Histogram

Using matlab we count R pixel hits. Then we create histogram to find our condition.

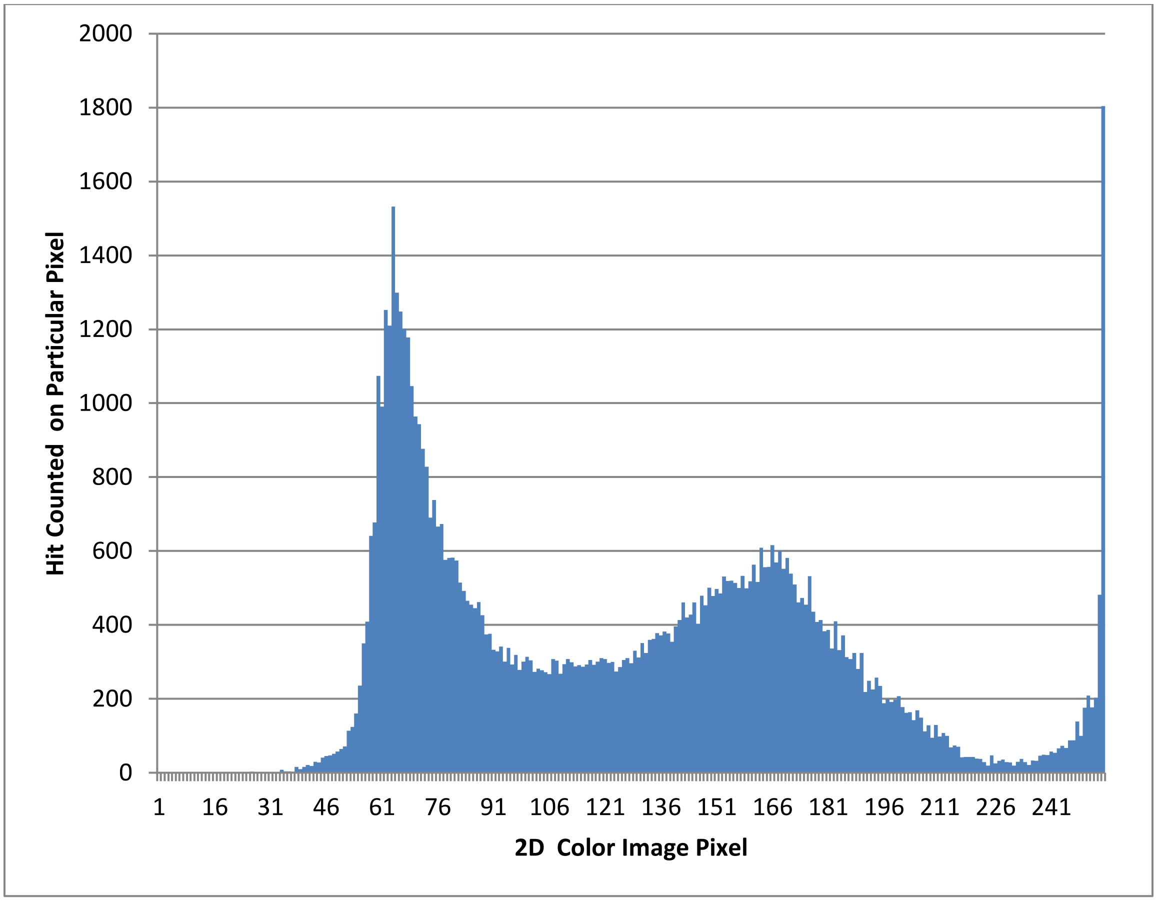

Figure 4.4: Histogram of B pixel hit count

In this histogram we see that hits in pixel 1 to 23 is zero. For this we skip this pixel and the hit in pixel 23 to 36 is very low. So we also skip this pixel. Then we develop an algorithm according this.

4.3 Result Analysis

For 33 image of the data set we get below accuracy of our method. For find accuracy we build a new algorithm. Using this algorithm we find the accuracy of our method and previous method. Then we compare our result with other method. It will discuss in section 4.4. In this section we will discuss about accuracy of our method.

Below table describe our method true positive result:

Table 4.1: Result

Rule	TP(%)	FP(%)
Kovac	81.46	16.76
Saleh	84.40	19.08
Swift	87.67	40.24
Ghazali Osman	91.22	37.84
Bayes SPM in RGB	80	8.5
Maximum Entropy Model in RGB	80	8
SOM in TS	78	32
Our propose method	91.63	42.36

4.4 Compare and Performance

Our method gives better result than other method. In our method we work on UChile dataset. Other method also works on UChile dataset. Compare to other rule our rule gives better result.

The performance of our method is better than other method. We checked it applying other method. The overall performance of our method is better than Kovac, Saleh, Swift, Ghazali Osman rule.

Example image set:

Figure 4.5: Set of image

Applying our method

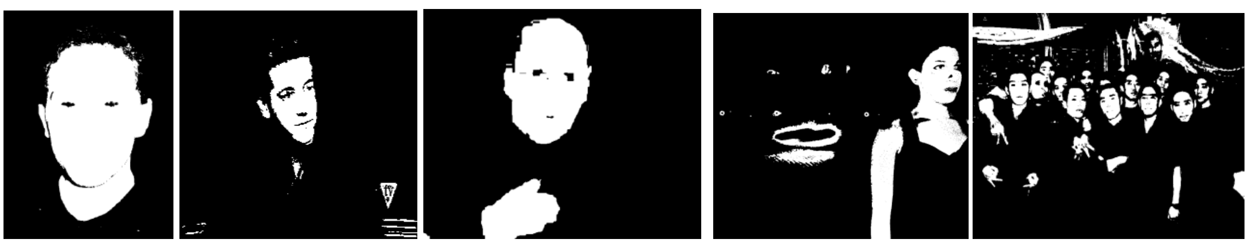

Figure 4.6: Set of image after applying our method

Chapter 5

Conclusion

In this chapter, we have summarized our work. This chapter contains two sections. In section 5.1 we will discuss summery. In section 5.2 represents future work

5.1Summery

Here, we have described skin recognition pattern and its accuracy. We have used 2D image for experiment .we used mathlab to find out the experimental value and its accuracy .we used different images to represent skin image where each image has different skin value. But we tried to fix a RGB range which helped us to detect skin image for all images. We developed some algorithm and also included a new rule to fix a skin image.

For fix the range we had to find out the RBG count list that mean which pixels how much hit and made a excel sheet by them and histogram. Then we used different range to fix skin .We generally Fix the range based on significant hits values pixel.

Then use fixed a suitable range for RGB to detect skin and make new image according to the value we got after those condition. And we get satisfactory accuracy.

Sometimes our algorithm cannot fix the perfect skin image those problems are created when in an images contain more Red values or yellow values. When background color are same with skin color in this case our method not good like other image.

5.2 Future work

Now we have tried to fix where we get more Red values or yellow values so that our algorithm cannot take it as a skin value if we fix it then our accuracy will be more than 95 %. In future we try to solve yellow color problem which is not skin but detected as a skin color. We also try to solve background problem. When background color and skin color are same in this case our method cannot perform well. In future we solve this problem. In medical application, in future we try to find Jaundice disease using skin detection.

References

[1] J. Kovac, P. Peer, and F. Solina, "Human skin colour clustering for face detection," in Proceeding of Eurocon 2003. Computer as a Tool. The IEEE Region 8, 2003, pp. 144-148.

[2] ZARIT, B. D., SUPER, B. J., AND QUEK, F. K. H. 1999. Comparison of five color models in skin pixel classification. In ICCV'99 Int'l Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems, 58–63.

[3] Wei Ren Tan, Chee Seng Chan, Member, IEEE, Pratheepan Yogarajah, and Joan Condell, "A Fusion Approach for Efficient Human Skin Detection," in proceedings of IEEE Transactions on Industrial Informatics, vol. 8, no. 1, pp. 138-147, February 2012.

[4] BROWN, D., CRAW, I., AND LEWTHWAITE, J. 2001. A som based approach to skin detection with application in real time systems. In Proc. of the British Machine Vision Conference, 2001.

[5] A. A.-S. Saleh, "A simple and novel method for skin detection and face locating and tracking," in Asia-Pacific Conference on Computer-Human Interaction 2004 (APCHI 2004), LNCS 3101, 2004, pp. 1-8.

[6.] D. B. Swift, "Evaluating graphic image files for objectionable content," US Patent US 6895111 B1, 2006

[7]Ghazali Osman, Muhammad Suzuri Hitam and Mohd Nasir Ismail "ENHANCED SKIN COLOUR CLASSIFIER USING RGB RATIO MODEL" International Journal on Soft Computing (IJSC) Vol.3, No.4, November 2012

[8] KRUPPA, H., BAUER, M. A., AND SCHIELE, B. 2002. Skin patch detection in real-world images. In Annual Symposium for Pattern Recognition of the DAGM 2002, Springer LNCS 2449, 109–117

[9] YANG, M.-H., AND AHUJA, N. 1998. Detecting human faces in color images. International Conference on Image Processing (ICIP), vol. 1, 127–130.

[10] JEDYNAK, B., ZHENG, H., DAOUDI, M., AND BARRET, D. 2002. Maximum entropy models for skin detection. Tech. Rep. XIII, Universite des Sciences et Technologies de Lille, France.

[11] TERRILLON, J.-C., SHIRAZI, M. N., FUKAMACHI, H., AND AKAMATSU, S. 2000. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In Proc. of the International Conference on Face and Gesture Recognition, 54–61.

[12] BRAND, J., AND MASON, J. 2000. A comparative assessment of three approaches to pixellevel human skin-detection. In Proc. of the International Conference on Pattern Recognition, vol. 1, 1056–1059.

[13] JONES, M. J., AND REHG, J. M. 1999. Statistical color models with application to skin detection. In Proc. of the CVPR ’99, vol. 1, 274–280

[14] SKARBEK, W., AND KOSCHAN, A. 1994. Colour image segmentation – a survey –. Tech. rep., Institute for Technical Informatics, Technical University of Berlin, October.

[15] SORIANO, M., HUOVINEN, S., MARTINKAUPPI, B., AND LAAKSONEN, M. 2000. Skin detection in video under changing illumination conditions. In Proc. 15th International Conference on Pattern Recognition, vol. 1, 839–842.

[16] OLIVER, N., PENTLAND, A., AND BERARD, F. 1997. Lafter: Lips and face real time tracker. In Proc. Computer Vision and Pattern Recognition, 123–129.

[17] POYNTON, C. A. 1995. Frequently asked questions about colour. In ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz

[18]MCKENNA, S., GONG, S., AND RAJA, Y. 1998. Modelling facial colour and identity with gaussian mixtures. Pattern Recognition 31, 12, 1883– 1892.

[19]SIGAL, L., SCLAROFF, S., AND ATHITSOS, V. 2000. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, 152–159.

[20]BIRCHFIELD, S. 1998. Elliptical head tracking using intensity gradients and color histograms. In Proceedings of CVPR ’98, 232–237.

[21]JORDAO, L., PERRONE, M., COSTEIRA, J., AND SANTOS-VICTOR, J. 1999. Active face and feature tracking. In Proceedings of the 10th International Conference on Image Analysis and Processing, 572–577

[22]PHUNG, S. L., BOUZERDOUM, A., AND CHAI, D. 2002. A novel skin color model in ycbcr color space and its application to human face detection. In IEEE International Conference on Image Processing (ICIP’2002), vol. 1, 289–292.

[23]MENSER, B., AND WIEN, M. 2000. Segmentation and tracking of facial regions in color image sequences. In Proc. SPIE Visual Communicationsand Image Processing 2000, 731–740

[24]HSU, R.-L., ABDEL-MOTTALEB, M., AND JAIN, A. K. 2002. Face detection in color images. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 5, 696–706.

[25]AHLBERG, J. 1999. A system for face localization and facial feature extraction. Tech. Rep. LiTH-ISY-R-2172, Linkoping University.

[26]CHAI, D., AND BOUZERDOUM, A. 2000. A bayesian approach to skin color classification in ycbcr color space. In Proceedings IEEE Region Ten Conference (TENCON'2000), vol. 2, 421–424

[27]PEER, P., KOVAC, J., AND SOLINA, F. 2003. Human skin colour clustering for face detection. In submitted to EUROCON 2003 – International Conference on Computer as a Tool

[28]FLECK, M., FORSYTH, D. A., AND BREGLER, C. 1996. Finding naked people. In Proc. of the ECCV, vol. 2, 592–602.

[29]GOMEZ, G., AND MORALES, E. 2002. Automatic feature construction and a simple rule induction algorithm for skin detection. In Proc. of the ICML Workshop on Machine Learning in Computer Vision, 31–38.

[30]SCHUMEYER, R., AND BARNER, K. 1998. A color-based classifier for region identification in video. In Visual Communications and Image Processing 1998, SPIE, vol. 3309, 189–200.

[31]TREES, H. L. V. 1968. Detection, Estimation, and Modulation Theory, vol. I. Wiley.

[32] J.G. Wang and E. Sung. "Frontal-view face detection and facial feature extraction using color and morphological operators". Pattern Recognition Letters 20 (1999)1053-1068.

[33] Y. Wang and B. Yuan. "A novel approach for human face detection from color images under complex background". Pattern Recognition 34 (2001) 1983-1992.

[34] J. Cai and A. Goshtasby. "Detecting human faces in color images". Image and Vision Computing 18 (1999) 63-75.

[35] E. Saber and A.M. Tekalp. "Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions". Pattern Recognition Letters 19 (1998) 669-680

[36] R.O. Duda and P.E. Haft. Pattern Classification and SceneAnalysis. John Wiley& Sons, New York, 1973

[37]LEE, J. Y., AND YOO, S. I. 2002. An elliptical boundary model for skin color detection. In Proc. of the 2002 International Conference on Imaging Science, Systems, and Technology.

[38]YANG, J., LU, W., AND WAIBEL, A. 1998. Skin-color modeling and adaptation. In Proceedings of ACCV 1998, 687–694.

[39]STERN, H., AND EFROS, B. 2002. Adaptive color space switching for face tracking in multi-colored lighting environments. In Proc. of the International Conference on Automatic Face and Gesture Recognition,249–255.

[40]M. STORRING, H. ANDERSEN, E. G. 1999. Skin colour detection under changing lighting condition. In Araujo and J. Dias (ed.) 7th Symposium on Intelligent Robotics Systems, 187–195.

[41]MARTINKAUPPI, B., AND SORIANO, M. 2001. Basis functions of the color signals of skin under different illuminants. In 3rd Intl conference on Multispectral Color Science, 21–24.

[42]MARTINKAUPPI, B., LAAKSONEN, M., AND SORIANO, M. 2001. Behavior of skin color under varying illumination seen by different cameras at different color spaces. In Machine Vision Applications in Industrial Inspection IX, Proceedings of SPIE, vol. 4301, 102–112.

[43]G. Mclachlan and T. Krishnan. The EM algorithm and extensions. John Wiley & Sons, New York, 1997.

Appendix

RGB hit pixel count

```
clc;

clear all;


RR  = [];

GG = [];

BB = [];

for i=1:44
    s = strcat(num2str(i));

    impath = strcat(s,'.png');

   IO = imread(impath) ;

   RR =  IO( : , : , 1);

   GG =  IO( : , : , 2);

   BB  =  IO( : , : , 3);

end


m = [];
for I = 1:255

m = [m numel(find(RR==i))];

end

A = find(m>0);

B =  m(A);

C =  [A' B'];

dlmwrite('textred.csv',C);
```

```
d= [];
for i = 1:255

d=[d numel(find(GG==i))];

end

Aa=find(d>0);

Bb= d(Aa);

Cc =[Aa' Bb'];

dlmwrite('textGreen.csv',Cc);

dd=[];

for i= 1:255

dd=[dd numel(find(BB==i))];

end

Aaa=find(dd>0);

Bb= dd(Aaa);

Ccc =[Aaa' Bb'];

dlmwrite('textBlue.csv',Ccc);
```

Skin pixel detection

For general purpose:

```
clear all;
clc;

I=imread('1.png');

final_image = zeros(size(I,1), size(I,2));

    for i = 1:size(I,1)

        for j = 1:size(I,2)

            R = I(i,j,1);

            G = I(i,j,2);

            B = I(i,j,3);

            if(R > 100 && G > 20 && B > 20)

                v = [R,G,B];

                if((mean(v) - min(v)) > 15)

                    if(abs(R-G) > 15 && R > G && R > B)

                        if (abs(R-G)<=120 && B<R && G<R)

                        if(0.0<((R-G)/(R+G))<= .5 & (B/(R+G)<= .5))

                        final_image(i,j) = 1;

                        end

                        end

                    end

                end

            end

        end
```

```
    end
    figure, imshow(final_image);
     imwrite(final_image,'1b1.png');
```

Some special case

```
clear all;
clc;

I=imread('1.png');
% figure,imshow(I);
 Red=[232,233];
 Green=[41,42];
Blue=[24,25,241,242,243,244,245,246,247];
final_image = zeros(size(I,1), size(I,2));
    for i = 1:size(I,1)
        for j = 1:size(I,2)
            R = I(i,j,1);
            G = I(i,j,2);
            B = I(i,j,3);
            if(R > 100 && G > 20 && B > 20)
                v = [R,G,B];
                if((mean(v) - min(v)) > 15)
                    if(abs(R-G) > 15 && R > G && R > B)
```

```
            if (abs(R-G)<=120 && B<R && G<R)
            if(0.0<((R-G)/(R+G))<= .5 & (B/(R+G)<= .5))
              if(R~=Red )
                if(G~=Green)
                if(B~=Blue)
            final_image(i,j) = 1;
                end
                end
              end
            end
            end
          end
        end
      end
    end
  end
  figure, imshow(final_image);
   imwrite(final_image,'1b1.png');
```

Ture Positive

```
clear all;
clc;

I  =  imread('9b.png');
```

```
I  =  im2bw(I);

I1 =  imread('9b2.png');

I1  =  im2bw(I1);

[x,y]=find(I = = 1 );

[x1,y1]=find(I1 = = 1);


flag =0;

for  i=1:numel(x1)

if  ~isempty(find (x==x1(i) & y==y1(i), 1))

flag = flag+1;

end

end

disp((flag/numel(x1)) *100);
```

False Positive

```
clear all;

clc;

I =  imread('37.png');

I =  im2bw(I);

I1=  imread('37b1.png');

I1 =  im2bw(I1);

[x,y] = find(I = = 0);

[x1,y1]=find(I1==1);
```

```
flag=0;
for  i=1:numel(x1)
  if  ~isempty(find (x==x1(i) & y==y1(i), 1))
    flag = flag+1;
   end
end
disp((flag/numel(x1)) *100)
```

References

[1] J. Kovac, P. Peer, and F. Solina, "Human skin colour clustering for face detection," in Proceeding of Eurocon 2003. Computer as a Tool. The IEEE Region 8, 2003, pp. 144-148.

[2] ZARIT, B. D., SUPER, B. J., AND QUEK, F. K. H. 1999. Comparison of five color models in skin pixel classification. In ICCV'99 Int'l Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems, 58–63.

[3] Wei Ren Tan, Chee Seng Chan, Member, IEEE, Pratheepan Yogarajah, and Joan Condell, "A Fusion Approach for Efficient Human Skin Detection," in proceedings of IEEE Transactions on Industrial Informatics, vol. 8, no. 1, pp. 138-147, February 2012.

[4] BROWN, D., CRAW, I., AND LEWTHWAITE, J. 2001. A som based approach to skin detection with application in real time systems. In Proc. of the British Machine Vision Conference, 2001.

[5] A. A.-S. Saleh, "A simple and novel method for skin detection and face locating and tracking," in Asia-Pacific Conference on Computer-Human Interaction 2004 (APCHI 2004), LNCS 3101, 2004, pp. 1-8.

[6.] D. B. Swift, "Evaluating graphic image files for objectionable content," US Patent US 6895111 B1, 2006

[7]Ghazali Osman, Muhammad Suzuri Hitam and Mohd Nasir Ismail "ENHANCED SKIN COLOUR CLASSIFIER USING RGB RATIO MODEL" International Journal on Soft Computing (IJSC) Vol.3, No.4, November 2012

[8] KRUPPA, H., BAUER, M. A., AND SCHIELE, B. 2002. Skin patch detection in real-world images. In Annual Symposium for Pattern Recognition of the DAGM 2002, Springer LNCS 2449, 109–117

[9] YANG, M.-H., AND AHUJA, N. 1998. Detecting human faces in color images. International Conference on Image Processing (ICIP), vol. 1, 127–130.

[10] JEDYNAK, B., ZHENG, H., DAOUDI, M., AND BARRET, D. 2002. Maximum entropy models for skin detection. Tech. Rep. XIII, Universite des Sciences et Technologies de Lille, France.

[11] TERRILLON, J.-C., SHIRAZI, M. N., FUKAMACHI, H., AND AKAMATSU, S. 2000. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In Proc. of the International Conference on Face and Gesture Recognition, 54–61.

[12] BRAND, J., AND MASON, J. 2000. A comparative assessment of three approaches to pixellevel human skin-detection. In Proc. of the International Conference on Pattern Recognition, vol. 1, 1056–1059.

[13] JONES, M. J., AND REHG, J. M. 1999. Statistical color models with application to skin detection. In Proc. of the CVPR '99, vol. 1, 274–280

[14] SKARBEK, W., AND KOSCHAN, A. 1994. Colour image segmentation – a survey –. Tech. rep., Institute for Technical Informatics, Technical University of Berlin, October.

[15] SORIANO, M., HUOVINEN, S., MARTINKAUPPI, B., AND LAAKSONEN, M. 2000. Skin detection in video under changing illumination conditions. In Proc. 15th International Conference on Pattern Recognition, vol. 1, 839–842.

[16] OLIVER, N., PENTLAND, A., AND BERARD, F. 1997. Lafter: Lips and face real time tracker. In Proc. Computer Vision and Pattern Recognition, 123–129.

[17] POYNTON, C. A. 1995. Frequently asked questions about colour. In ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz

[18]MCKENNA, S., GONG, S., AND RAJA, Y. 1998. Modelling facial colour and identity with gaussian mixtures. Pattern Recognition 31, 12, 1883– 1892.

[19]SIGAL, L., SCLAROFF, S., AND ATHITSOS, V. 2000. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, 152–159.

[20]BIRCHFIELD, S. 1998. Elliptical head tracking using intensity gradients and color histograms. In Proceedings of CVPR '98, 232–237.

[21]JORDAO, L., PERRONE, M., COSTEIRA, J., AND SANTOS-VICTOR, J. 1999. Active face and feature tracking. In Proceedings of the 10th International Conference on Image Analysis and Processing, 572–577

[22]PHUNG, S. L., BOUZERDOUM, A., AND CHAI, D. 2002. A novel skin color model in ycbcr color space and its application to human face detection. In IEEE International Conference on Image Processing (ICIP'2002), vol. 1, 289–292.

[23]MENSER, B., AND WIEN, M. 2000. Segmentation and tracking of facial regions in color image sequences. In Proc. SPIE Visual Communicationsand Image Processing 2000, 731–740

[24]HSU, R.-L., ABDEL-MOTTALEB, M., AND JAIN, A. K. 2002. Face detection in color images. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 5, 696–706.

[25]AHLBERG, J. 1999. A system for face localization and facial feature extraction. Tech. Rep. LiTH-ISY-R-2172, Linkoping University.

[26]CHAI, D., AND BOUZERDOUM, A. 2000. A bayesian approach to skin color classification in ycbcr color space. In Proceedings IEEE Region Ten Conference (TENCON'2000), vol. 2, 421–424

[27]PEER, P., KOVAC, J., AND SOLINA, F. 2003. Human skin colour clustering for face detection. In submitted to EUROCON 2003 – International Conference on Computer as a Tool

[28]FLECK, M., FORSYTH, D. A., AND BREGLER, C. 1996. Finding naked people. In Proc. of the ECCV, vol. 2, 592–602.

[29]GOMEZ, G., AND MORALES, E. 2002. Automatic feature construction and a simple rule induction algorithm for skin detection. In Proc. of the ICML Workshop on Machine Learning in Computer Vision, 31–38.

[30]SCHUMEYER, R., AND BARNER, K. 1998. A color-based classifier for region identification in video. In Visual Communications and Image Processing 1998, SPIE, vol. 3309, 189–200.

[31]TREES, H. L. V. 1968. Detection, Estimation, and Modulation Theory, vol. I. Wiley.

[32] J.G. Wang and E. Sung. "Frontal-view face detection and facial feature extraction using color and morphological operators". Pattern Recognition Letters 20 (1999)1053-1068.

[33] Y. Wang and B. Yuan. "A novel approach for human face detection from color images under complex background". Pattern Recognition 34 (2001) 1983-1992.

[34] J. Cai and A. Goshtasby. "Detecting human faces in color images". Image and Vision Computing 18 (1999) 63-75.

[35] E. Saber and A.M. Tekalp. "Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions". Pattern Recognition Letters 19 (1998) 669-680

[36] R.O. Duda and P.E. Haft. Pattern Classification and SceneAnalysis. John Wiley& Sons, New York, 1973

[37]LEE, J. Y., AND YOO, S. I. 2002. An elliptical boundary model for skin color detection. In Proc. of the 2002 International Conference on Imaging Science, Systems, and Technology.

[38]YANG, J., LU, W., AND WAIBEL, A. 1998. Skin-color modeling and adaptation. In Proceedings of ACCV 1998, 687–694.

[39]STERN, H., AND EFROS, B. 2002. Adaptive color space switching for face tracking in multi-colored lighting environments. In Proc. of the International Conference on Automatic Face and Gesture Recognition,249–255.

[40]M. STORRING, H. ANDERSEN, E. G. 1999. Skin colour detection under changing lighting condition. In Araujo and J. Dias (ed.) 7th Symposium on Intelligent Robotics Systems, 187–195.

[41]MARTINKAUPPI, B., AND SORIANO, M. 2001. Basis functions of the color signals of skin under different illuminants. In 3rd Intl conference on Multispectral Color Science, 21–24.

[42]MARTINKAUPPI, B., LAAKSONEN, M., AND SORIANO, M. 2001. Behavior of skin color under varying illumination seen by different cameras at different color spaces. In Machine Vision Applications in Industrial Inspection IX, Proceedings of SPIE, vol. 4301, 102–112.

[43]G. Mclachlan and T. Krishnan. The EM algorithm and extensions. John Wiley & Sons, New York, 1997.

Appendix

RGB hit pixel count

```
clc;

clear all;


RR  = [];

GG = [];

BB = [];

for i=1:44
    s = strcat(num2str(i));

    impath = strcat(s,'.png');

   IO = imread(impath) ;

   RR =  IO( : , : , 1);

   GG =  IO( : , : , 2);

   BB  =  IO( : , : , 3);

end


m = [];
for I = 1:255

m = [m numel(find(RR==i))];

end

A = find(m>0);

B =  m(A);

C =  [A' B'];

dlmwrite('textred.csv',C);
```

```
d= [];
for i = 1:255

d=[d numel(find(GG==i))];

end

Aa=find(d>0);

Bb= d(Aa);

Cc =[Aa' Bb'];

dlmwrite('textGreen.csv',Cc);

dd=[];

for i= 1:255

dd=[dd numel(find(BB==i))];

end

Aaa=find(dd>0);

Bb= dd(Aaa);

Ccc =[Aaa' Bb'];

dlmwrite('textBlue.csv',Ccc);
```

Skin pixel detection

For general purpose:

```
clear all;
clc;

I=imread('1.png');

final_image = zeros(size(I,1), size(I,2));

    for i = 1:size(I,1)

        for j = 1:size(I,2)

            R = I(i,j,1);

            G = I(i,j,2);

            B = I(i,j,3);

            if(R > 100 && G > 20 && B > 20)

                v = [R,G,B];

                if((mean(v) - min(v)) > 15)

                    if(abs(R-G) > 15 && R > G && R > B)

                        if (abs(R-G)<=120 && B<R && G<R)

                        if(0.0<((R-G)/(R+G))<= .5 & (B/(R+G)<= .5))

                        final_image(i,j) = 1;

                        end

                        end

                    end

                end

            end

        end
```

```
    end
    figure, imshow(final_image);
     imwrite(final_image,'1b1.png');
```

some special case

```
clear all;
clc;

I=imread('1.png');
% figure,imshow(I);
 Red=[232,233];
 Green=[41,42];
Blue=[24,25,241,242,243,244,245,246,247];
final_image = zeros(size(I,1), size(I,2));
    for i = 1:size(I,1)
        for j = 1:size(I,2)
            R = I(i,j,1);
            G = I(i,j,2);
            B = I(i,j,3);
            if(R > 100 && G > 20 && B > 20)
                v = [R,G,B];
                if((mean(v) - min(v)) > 15)
                    if(abs(R-G) > 15 && R > G && R > B)
                        if (abs(R-G)<=120 && B<R && G<R)
```

```
            if(0.0<((R-G)/(R+G))<= .5 & (B/(R+G)<= .5))
              if(R~=Red )
                if(G~=Green)
                if(B~=Blue)
            final_image(i,j) = 1;
                end
                end
              end
            end
            end
          end
        end
      end
    end
  end
  figure, imshow(final_image);
   imwrite(final_image,'1b1.png');
```

Ture Positive

```
clear all;
clc;

I  =  imread('9b.png');
I  =  im2bw(I);
```

```
I1 =  imread('9b2.png');

I1  =  im2bw(I1);

[x,y]=find(I = = 1 );

[x1,y1]=find(I1 = = 1);


flag =0;

for  i=1:numel(x1)

if  ~isempty(find (x==x1(i) & y==y1(i), 1))

flag = flag+1;

end

end

disp((flag/numel(x1)) *100);
```

False Positive

```
clear all;

clc;

I =  imread('37.png');

I =  im2bw(I);

I1=  imread('37b1.png');

I1 =  im2bw(I1);

[x,y] = find(I = = 0);

[x1,y1]=find(I1==1);

flag=0;
```

```
for  i=1:numel(x1)
  if  ~isempty(find (x==x1(i) & y==y1(i), 1))
    flag = flag+1;
   end
end
disp((flag/numel(x1)) *100)
```