An Empiric Model of Face Detection based on RGB Skin Tone Color

Robin1, Ferawaty1, Jusin2, Hita1, Syanti Irviantina1, Saut Dohot Siregar3, Wenripin Chandra1

1STMIK Mikroskil, Medan, Indonesia  
2Universitas Pelita Harapan, Medan, Indonesia  
3 Universitas Prima Indonesia, Medan, Indonesia

E-Mail: robin80huang@gmail.com

Abstract. Modern smartphone has been society’s lifestyle where every smartphone has a high quality digital camera with all the digital image processing feature. One of those features are face detection. Face detection is the most basic process of any face processing operations. Most digital images are stored in the form of RGB (Red, Green, Blue) data. In this research, detection of human face features is done by using different RGB values in digital images. After applying skin tone color segmentation on digital images, detection area will be optimized using human head properties by eliminating non-human face skin tone area. The experiment shows that the detection is mostly accurate for images but there is issue on low light face skin color captured by digital cameras. From the experiment of our model using 10 sample face images, face can be detected on 9 of them, while in 1 image, the face cannot be detected at all because of low light condition.

1. Introduction
Modern smartphone such as iPhone or Android phone has been society’s lifestyle where every smartphone is now has integrated with a high quality digital camera along with all digital imaging technology. Actually there are a set of technology feature and component has integrated into smartphone camera where the improvement not only on hardware feature but the improvement also happens on software feature. The existance and improvement of digital imaging technology has affected the improvement digital image processing field. One of many field in digital image processing is face detection. Face detection is technique to identify human face in digital image from its surrounding. Many methods can be utilised to achive positive detection such as neural network, color-based, Gaussian Compound-based or feature-based [1].

Color-based is one of the most popular method used in face detection because of its speed in training as well as usage and independence to the distribution of neighboring color [2]. In this method, face detection is done by differing the color of face from non-face. Based on the color space used, there are several models in ths category: RGB model, YCbCr model and its variation YCgCr [3], and HSI model. Utilising color to detect human face brings its own challenges because the color depends on factors such as light condition, object movement, etc [4].
RGB color model is representation of image as three color components that is red, green and blue color. This model is very popular for storing and representing digital images [2]. RGB is based on Cartesian coordinate system, as shown in Figure 1 [2].

![RGB Color Model](image)

Figure 1. RGB Color Model.[2]

This model consists of three primary colors (red, green and blue) and any other colors are obtained by mixing these three primary colors. Each color is represented by value ranging from 0 to maximum. The maximum value is depended to the color depth and changing the dose (value) of each base color, any other colors can be created [2].

Detecting human face using RGB color is to determine the the human face color value. The accuracy of the detection is determined by RGB value perceived as human face and method in using those values.

Some researches in this area has been conducted using these model where the R, G and B in sequence represent the red channel value, green channel value and blue channel value of an digital image picture element (pixel). [2]

\[
R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and } \\
(\max\{R,G,B\} - \min\{R,G,B\}) > 15 \text{ and } \\
R-G > 15 \text{ and } R > G \text{ and } R > B
\]  
(1)

Or [3]

\[
(R>50) \text{ and } (G>40) \text{ and } (B>20) \text{ and } \\
(\max\{R,G,B\} - \min\{R,G,B\}>10) \text{ and } \\
(R - G \geq 10) \text{ and } (R>G) \text{ and } (R>B)
\]  
(2)

The models above use statistical operations min and max in addition to some comparison functions. These will tax the computer especially when processing large images. In this paper, we propose new model which eliminate those statistical operations while retain the accuracy of previous models. We will also try to increase the accuracy of the model by selecting the best value for RGB channel. The model is more simple and light in processing and empiric in nature.

2. Methodology
The premises used here is human skin color contain less greenish and bluish color, but dominated by color contains reddish color. In order to distinguish human face from the surrounding, we need to lower green and blue value in RGB. Regarding green and blue component, we assume that green component is more prevalent in human skin, therefore we took green component higher than blue component. So the basic of this proposed model would be R-G and G-B.
Due to human eye capability to distinguish about 10 million colors \[5\], we assume each color component has value between 0 to 255, corresponding to 8-bit color representation in digital image. Value 0 will be darkest, while 255 will be brightest.

Human skin color should not be too dark or too bright, so the value of each color component must not too low or too high. Based on test, we settle the threshold value for the model. The value along with the model used for image segmentation is constructed as

\[
(B\geq50 \text{ and } B\leq215) \text{ and (G-B}\geq10 \text{ and } G-B\leq65) \text{ and (R-G}\geq10 \text{ and } R-G\leq85) \tag{3}
\]

While applying image segmentation, human face holes pixels must be considered. In the experiment, tolerable black pixel of human face holes for image segmentation is 50 pixels in horizontal direction and 100 pixels for vertical direction.

According to human head anatomy, human head’s width and height proportion is in scale of 2:3. Therefore besides human skin color segmentation, we also have to eliminate non-face area of the image according to the proportion of human head properties. \[6,7,8\]

These are steps of the experiment process.

1. Input a digital image into the model.
2. Applying image segmentation into a binary type image based on skin tone color using our model for each x-axis horizontal and y-axis vertical pixel. Skin tone color detected will be truncated into 1 which is known as white color value while non skin tone color will be truncated into 0 which is known as black color value.
3. Find and calculate the rectangular position of a human face gained from the white area of the segmented binary image by determining the starting left-top and ended with right-bottom coordinates.
4. Resize the rectangular size to match human face properties.
5. observe and determine whether the detection result accurate or not.

3. Experiment

We construct software to perform experiment to verify the effectiveness of the proposed model. After determining the detection result, these steps will be repeated for each image used in the experiment. The accuracy result was categorized as one of: face detected accurately, almost accurately, less accurate and no face detected at all.

As discussed in methodology, the segmentation process uses the following pseudocode.

FOR \( y = 0 \) TO OriginalImage.Height-1
FOR \( x = 0 \) TO OriginalImage.Width-1
OriginalImage.GetRGB(x, y, R, G, B)
IF \( B\geq50 \text{ AND } B\leq215 \) THEN
GSubB = G-B
RSubG = R-G
IF \( (GSubB\geq10 \text{ AND } GSubB\leq65) \text{ AND (RSubG}\geq10 \text{ AND } RSubG\leq85) \) THEN
Segmented.SetPixel(x, y, WHITE)
ELSE
Segmented.SetPixel(x, y, BLACK)
ENDIF
ELSE
Segmented.SetPixel(x, y, BLACK)
ENDIF
END FOR
END FOR

These pseudocode below is to find and calculate the rectangular position of human face.

\( y=0 \)
WHILE \( y<\text{Segmented.Height AND NOT FaceFound} \)
\begin{verbatim}
x=0 
WHILE x<Segmented.Width AND NOT FaceFound
  Pixel=Segmented.GetByte(x, y)
  IF Pixel=WHITE THEN
    y1=y  tmpy=y
    BlackCount=0
    LoopStatus=TRUE
    WHILE LoopStatus AND tmpy>0
      IF Segmented.GetByte(x, tmpy)=WHITE THEN
        y1=tmpy
      ELSE
        BlackCount=BlackCount+1
      END IF
      IF BlackCount>50 THEN LoopStatus=FALSE
      tmpy=tmpy-1
    END WHILE
  END IF
  y2=y  tmpy=y
  BlackCount=0
  LoopStatus=TRUE
  WHILE LoopStatus AND tmpy<Segmented.Height
    IF Segmented.GetByte(x, tmpy)=WHITE THEN
      y2=tmpy
    ELSE
      BlackCount=BlackCount+1
    END IF
    IF BlackCount>50 THEN LoopStatus=FALSE
    tmpy=tmpy+1
  END WHILE
  x1=x2=x
  FOR tmpy=y1 TO y2
    tmpx=x1-1
    BlackCount=0
    LoopStatus=TRUE
    WHILE tmpx>=0 AND LoopStatus
      IF Segmented.GetByte(tmpx, tmpy)=WHITE THEN
        IF x1>tmpx THEN x1=tmpx
      ELSE
        BlackCount=BlackCount+1
      END IF
      IF BlackCount>100 THEN LoopStatus=FALSE
      tmpx=tmpx-1
    END WHILE
    tmpx=x2+1
    BlackCount=0
    LoopStatus=TRUE
    WHILE tmpx<Segmented.Width AND LoopStatus
      IF Segmented.GetByte(tmpx, tmpy)=WHITE THEN
        IF x2<tmpx THEN x2=tmpx
      ELSE
        BlackCount=BlackCount+1
      END IF
      IF BlackCount>TolerateBlackCountHoriz THEN LoopStatus=FALSE
      tmpx=tmpx+1
    END WHILE
  END FOR
END WHILE
\end{verbatim}
IF x1+50<x2 AND y1+50<y2 THEN
    FaceFound=TRUE
ELSE
    x1 = x2 = y1 = y2 = -1
END IF
END IF
END FOR
x=x+1
END WHILE
y=y+1
END WHILE

Also these pseudocode is applied to match the human head properties.
IF FaceFound THEN
    DetectedWidth=x2-x1+1
    DetectedHeight=y2-y1+1
    DH=_detectedHeight / 6
    Scale=DetectedHeight*100/DetectedWidth
    BlackCount=0
    WhiteCount=0
    FOR y=y1 TO y1+DH
        FOR x=x1 TO x2
            IF Segmented.GetPixel(x, y)=WHITE THEN
                WhiteCount=WhiteCount+1
            ELSE
                BlackCount=BlackCount+1
            END IF
        END FOR
        IF BlackCount>=WhiteCount THEN y1=y1+DH
    END FOR
    IF Scale>140 THEN y2=y1+DetectedHeight*100/Scale
x=x1                //decrease left margin
LoopStatus=TRUE
WHILE x<x2 AND LoopStatus
    BlackCount=0
    WhiteCount=0
    FOR y=y1 TO y2
        IF Segmented.GetPixel(x, y)=WHITE THEN
            WhiteCount=WhiteCount+1
        ELSE
            BlackCount=BlackCount+1
        END IF
    END FOR
    IF BlackCount>WhiteCount THEN x1=x ELSE LoopStatus=FALSE
    x=x+1
END WHILE
x=x2                //decrease right margin
LoopStatus=TRUE
WHILE x>x1 AND LoopStatus
    BlackCount=0
    WhiteCount=0
    FOR y=y1 TO y2
        IF Segmented.GetPixel(x, y)=WHITE THEN
            WhiteCount=WhiteCount+1
        ELSE
            BlackCount=BlackCount+1
        END IF
    END FOR
    IF WhiteCount>BlackCount THEN x1=x ELSE LoopStatus=FALSE
    x=x+1
END WHILE
WhiteCount=WhiteCount+1
ELSE
    BlackCount=BlackCount+1
END IF
END FOR
IF BlackCount>WhiteCount THEN x2=x ELSE LoopStatus=FALSE
    x=x-1
END WHILE

DetectedWidth=x2-x1+1
DetectedHeight=y2-y1+1
Scale=DetectedHeight*100/DetectedWidth
IF Scale<140 THEN
    y2=y1+DetectedHeight*100/Scale
ELSE
    y2=y1+DetectedWidth*1.2
IF y2>=ImgData.Height THEN y2=ImgData.Height-1
END IF

DrawDetectionRectangle(x1, y1, x2, y2)

There are 15 face images sample to test in this research. Many of them are photographs of STMIK Mikroskil and Universitas Prima Indonesia high school students and a few of them is provided by google from this URL https://www.google.co.id/search?safe=strict&biw=1280&bih=725&tbm=isch&sa=1&ei=aUZAXNfQLcnc9QPtioDQAg&q=face+detection+sample+images&oq=face+detection+sample+images&gs_l=img.3...49510.50423.50559...0.0.133.708.3j4.....0...1..gws-wizimg......0i19j0i7i30i19j0i8i30i19.9vsC11bSC20 access date Oct 18th 2018.

4. Result and Discussions
We use 16 images labelled as Face00 through Face15 as sample. Each image contains a face. These images are in pristine condition and do not pre-processed before. We apply the proposed model to each image. Original images and respective results are presented in Figure 2. Qualitative result of the experiment shown in Table 1.

| Image   | Resolution | Result       |
|---------|------------|--------------|
| Face00  | 120 x 150  | Accurate     |
| Face01  | 101 x 150  | Accurate     |
| Face02  | 100 x 150  | Accurate     |
| Face03  | 100 x 150  | Accurate     |
| Face04  | 112 x 150  | Less Accurate|
| Face05  | 100 x 150  | Accurate     |
| Face06  | 100 x 150  | Accurate     |
| Face07  | 251 x 193  | Accurate     |
| Face08  | 320 x 400  | Accurate     |
| Face09  | 200 x 250  | Almost Accurate|
| Face10  | 500 x 500  | Less Accurate|
| Face11  | 1000 x 1000| Almost Accurate|
| Face12  | 249 x 193  | Accurate     |
| Face13  | 250 x 193  | Less Accurate|
Figure 2. Image Segmentation and Face Detection Result on Face00 to Face07.
The experiment gives variety result ranging from accurate (face area detected perfectly) to less accurate (face area detected is smaller than actual face) to no face detected. As shown in Table 1, detection results of Face00, Face01, Face02, Face03, Face5, Face06, Face07, Face08 and Face12 are perfectly acceptable, while Face09, Face10, Face11 and Face13 is also acceptable while Face04 is less acceptable. The face in Face15 cannot be detected at all.

The result of Face04 is less acceptable. The face includes the neck region that has almost-similar tone to the face. That’s why this model confuses to determine correctly the face region.

In Face15, the face cannot be detected at all. We can deduce this result caused by the brownish background color is too similar to the face skin tone color. Color-based model detection which rely solely to detect the pre-determined face color shows its weakness in this case.

![Figure 3. Experiment result on Face15](image)

5. Conclusions
In this paper, we proposed a model to detect human face based on RGB. We have done experiment on some images. Looking at the result, the model seems to have shortfall in detecting face with underexposure. From the detection result of our model using 10 sample face images, 6 of them accurate, 3 of them is less accurate and only 1 image cannot be detected at all which is caused by low light condition. In ideal condition, the model will be able to detect face accurately. The future work will focus on better detection in low light image and ideal edge detection on human face on live security camera.

References
[1] Kumar GP, Shashidhara H, 2014 Real Time Detection and Tracking of Human Face Using Skin Color Segmentation and Region Properties, Engineering and Technology Publishing
[2] Kolkul S, Kalbande D, Shimpli P, Bapat C, Jatakia K, 2016 Human Skin Detection Using RGB, HSV and YCbCr Color Models ICCASP/ICMMD-2016
[3] Ghazali KM, Ma J, Xiao R, 2018 An Innovative Face Detection based on Skin Color Segmentation International Journal of Computer Applications Vol 34 No 2
[4] Zangana HM, Al-Shaikhli IF, 2013 A New Algorithm for Human Face Detection Using Skin Color Tone IOSR Journal of Computer Engineering, Vol 11
[5] Thakur S, Sayantanu P, Ankur M, Das S, Abraham A, 2011 Face Detection Using Skin Tone Segmentation IEEE

[6] Singh Raghuvanshi D, Agrawal D, 2012 Human Face Detection by using Skin Color Segmentation, Face Features and Regions Properties, International Journal of Computer Applications 0975-8887, Vol 38, No. 9

[7] Aznaveh MM, Mirzaei H, Roshan E, Saraee MH 2009 A New and Improved Skin Detection Method Using Mixed Color Space AISC-60

[8] Mohanty R, Raghunadh, 2016 Skin Color Segmentation based Face Detection using Multi-Color Space IJARCCE Vol 5 No 5

[9] Pujol FA, Pujol, M, Jimeno-Morenilla A, Pujol MJ, 2017 Face Detection Based on Skin Color Segmentation Using Fuzzy Entropy

[10] Gupta N, Jain V, Mishra S, 2015 Enhanced Face Detection Using Skin Color Model: A Review IJARCSEE Vol 4 No 11

[11] Buza E, Akagic A, Omanovic S, 2017 Skin Detection Based on Image Color Segmentation with Histogram and K-Means Clustering, IEEE

Acknowledgement
This paper is funded by STMIK Mikroskil