Paper Title ***Marvel App***

Dzvezdana Arsovska Pavel Shumejko Ivan Iuschenko

University of Tartu

***Abstract* — This paper will present an application for human skin detection. The application also allows the user to change the color of the skin to green or other available colors. The project is realized in Python with the help of OpenCV and other necessary libraries. The skin detection will be done in the HSV color space. An option of changing the skin detection parameters and comparing it with skin detection in another color space is added. By using the GUI the user will be able to choose among various skin colors and display the world in the edge format.**

Keywords — HSV; Color space; Skin detection; Python; Skin color; Hue; Saturation; OpenCV;

I. INTRODUCTION

In image processing, skin detection represents the process of finding skin-colored pixels and regions in an image or a video. It usually is a process that starts at a pixel-level, and that involves a pre-process of color space transformation followed by a classification process. A color space transformation is assumed to increase separability between skin and non-skin classes, to increase similarity among different skin tones, and to bring a robust performance under varying illumination conditions.

The objective of the skin detection is to find out skin regions in an image. [1] Skin color detection is the process of separation between skin and non-skin pixels. Detection of human skin is an essential step in many computer vision applications such as: automatic face and skin recognition and detection, face or hand tracking, video surveillance, human computer interaction (HCI), filtering adult content on the internet and many other uses in video applications.

II. METHODS OF SKIN DETECTION

There are two types of skin detection: pixel based skin detection and region based skin detection. [1] In pixel based skin detection each pixel is classified as either skin or non-skin individually from its neighbor. The skin detection based on color fall in this category. [2] In the region based skin detection, skin pixels are spatially arranged to enhance the performance. This method requires additional information such as intensity or texture. Skin color has proven to be a useful and robust cue for skin detection and over the several past decade many pattern recognition based strategies have been proposed for achieving robust and accurate solution.

A. *Skin Color Based Methods - Advantages*

Pixel based methods that use skin color ranges allow fast processing and are highly robust to geometric variations of the face pattern. Also, the experience suggests that human skin has a characteristic color, which is easily recognized by humans. In addition this methods are robust under various circumstances and they are robust to resolution changes. Also they eliminate the need of slow or complicated tracking devices. The simplicity of this method makes it ideal from computational perspective and therefore it can be used for real time applications like video processing. [3]

B. *Skin Color Based Methods – Disvantages*

Using this method will result in a few challenges and difficulties. In a controlled background environment, skin detection can be sufficient to locate faces and skin regions in images. As we previously described, color processing is much faster than processing other facial and skin features and is highly robust to geometric variations of the face and skin patterns, it is typically used as a step in detecting regions in images that potentially have human faces and body limbs. However, many objects in the real world have some skin-tone colors which causes false detections in skin detection where different non-skin objects are recognized as skin. [4]

Uncontrolled backgrounds, other external factors such as change of illumination, the quality of the video/image (different cameras have different output for identical input), blurring caused by movement can challenge the skin detection process. In addition human skin comes in a wide variety of colors and it varies dramatically from one individual to another. Using wide range of skin colors will result in noise so the range of the values should be carefully chosen. Also another challenge will be creating a fast computational algorithm suitable for real time application.

C. *Factors to be considered*

In the skin color detection process, it is necessary to consider the following factors:

- The distinction or separation of skin and non-skin pixels in the image.
- The device for capture the image. For the same image, different cameras have different output.
- Whether illumination varies drastically in the image?
- Skin tones vary from one person to others.
- Movement of object degrades the quality of image due to blurring of colors.
- Shadows and lightness has a vital role to change the color of the image.
- The color space used for the detection or segmentation.

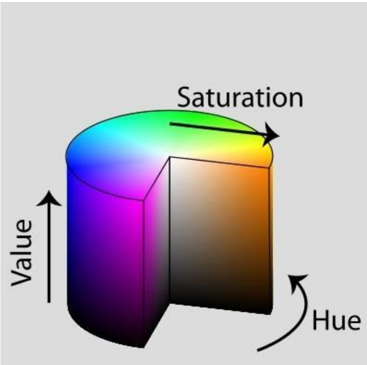
III. COLOR SPACE FOR COLOR DETECTION

An appropriate color space for skin detection needs to be chosen after careful consideration of the advantages that different color spaces have. Color space is a mathematical model to represent color information as three or four different color components. The purpose of color space is to facilitate the specification of colors in some standard, generally accepted way. In essence, a color space is a specification of a coordinate system and a subspace within the system where each color is represented by a single point. The color space has to increase the separability between skin and non-skin classes, decreased separability among skin tones and the cost of conversion for real time applications need to be considered. In terms of digital image processing the most commonly used color models in practice are RGB (red, green, blue) color model, CMY (cyan, magenta, yellow), CMYK (cyan, magenta, yellow, black) and HSV (hue, saturation, value).

Different color models are used for different applications such as computer graphics, image processing, TV broadcasting, and computer vision. Different color space is available for the skin detection. They are: RGB (red, green, blue) based color space (RGB, normalized RGB), Hue Based color space (HSI (hue, saturation, intensity), HSV, and HSL), Luminance based color space (YCbCr, YIQ, and YUV), and perceptually uniform color space (CIEXYZ, CIELAB, and CIELUV). Several color spaces have been proposed in the literature for skin detection applications. In this project, the HSV color space will be used for skin detection. HSV, which is a perceptual color space that have been used in many skin detection applications and has many advantages. One of it is that it allows the boundary of the skin color class to be intuitively specified by the users by manipulating the hue and saturation component. This color space has three components: the hue (H), the saturation (S) and the brightness (V). The brightness component is often not considered in order to reduce the illumination dependency of the skin color.

IV. HSV COLOR SPACE

Hue-saturation based color spaces were introduced when there was a need for the user to specify color properties numerically. They describe color with intuitive values, based on the artist's idea of tint, saturation and tone. 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. The ”intensity”, ”lightness” or ”value” is related to the color luminance. [3] The intuitiveness of the color space components and explicit discrimination between luminance and chrominance properties makes these color spaces popular in the works on skin color segmentation. Several interesting properties of Hue are that 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. However, several undesirable features of these color spaces exist, including hue discontinuities and the computation of ”brightness” (lightness, value), which conflicts badly with the properties of color vision.



Picture 1 HSV color space

The HSV color space owes its usefulness to two principal facts. First, the intensity component V is decoupled from the color information in the image. Second, the hue and saturation components are intimately related to the way in which human beings perceive color. These features make the HSV model an ideal tool for developing image processing algorithms based on the color sensing properties of the human visual system.

Color images in RGB color space are transformed in HSV color space using the following equations:

$$H = \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B > G \end{cases} \tag{1}$$

$$\theta = cos^{-1} \frac{\frac{1}{2}[(R-G)+(R-B))]}{[(R-G)^2+(R-B)(G-B)]^{1/2}} \tag{2}$$

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

$$I = \frac{1}{3}(R+G+B) \tag{4}$$

V. Algorithm for Skin Detection

Picture 2 illustrates the algorithm steps. The code was created using Python 2.7.2 and the OpenCV library. The Graphical User Interface (GUI) was created using Tkinter. First, the video frame which is represented in BGR color space is smoothed with a Gaussian low pass filter (kernel size of 3x3 of ones). Then the filtered video frame is converted from BGR to HSV color space. A binary mask is created using the upper and lower limits for the HSV values for the skin we want to detect. After the thresholding, the skin pixels are represented with ones while the non-skin pixels are shown as zeros. Additional noise removal is done using the morphological operation “Closing” on the binary mask (kernel size of 3x3 of ones and 2 iterations). Using the bitwise AND operation the segmented skin and its color representations are stored as outputs (Skin mask and colored skin masks in red, green and blue). Another operation which changes the pixels on the HSV frame is applied using the mask as a pointer which determines which skin pixels need to have their H value change to a value that corresponds to a green color. Also sliders for the upper and lower limits of the HSV values are added. This allows the user to change the mask in real time. Thus the optimal mask depending on the current conditions can be applied. In addition for comparison, the video frames are converted in YCbCr color space and a new binary mask is created using fixed upper and lower limits the YCbCr channels. The detected skin pixels in this mask are also represented with ones and the non-skin pixels as zeros. The same method which was used before on the HSV frame is applied now in order to color only the skin pixels with a green color. If the user chooses the option to represent the video in edge format, the video frames are represented in the edge format using canny edge detector. Also an option to color the edges in the 3 primary (RGB) colors is available in the GUI.

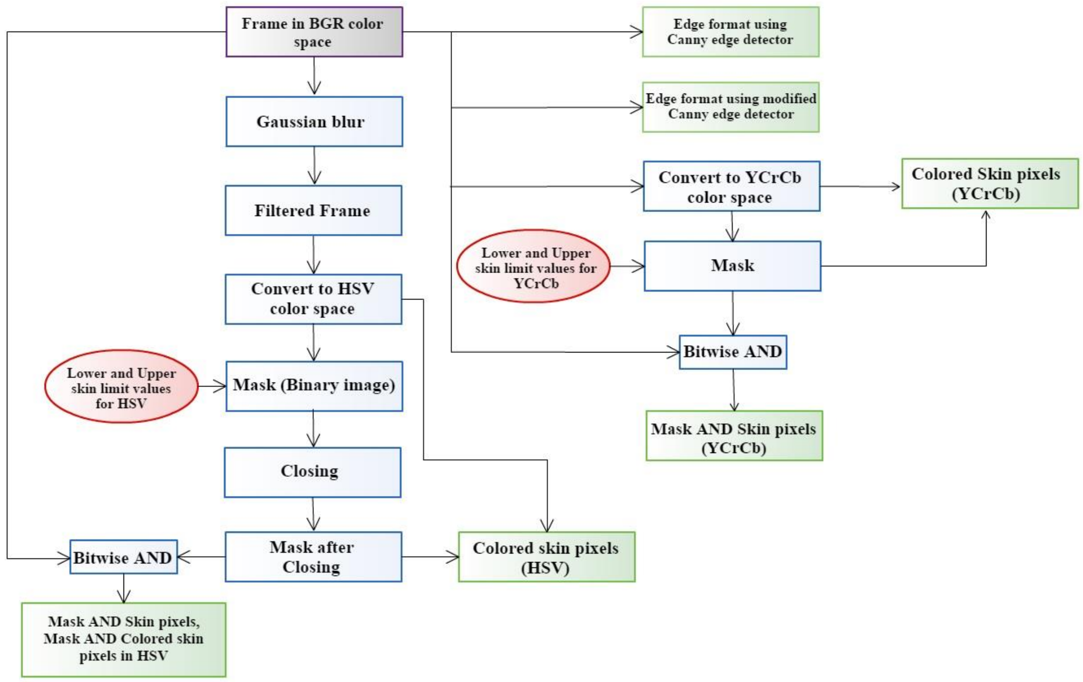
VI. Noise Filteringg

Due to the factors listed in section II.B the video is corrupted by noise which needs to be removed. Some of the noise is removed using Gaussian low pass filter and morphological operations.

A. *Gaussian filter*

First, the frames are smoothened using Gaussian low pass filter with kernel 3x3. The form of the filter in two dimensions is given by:

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \tag{5}$$



Picture 2 Application’s algorithm represented as a flow chart

where $D(u,v)$ is the distance from the origin of the Fourier is transform and D_0 is the cut off frequency. The Gaussian filter has the effect of reducing the image's high frequency components. This results in blurred image with reduced noise and reduced details. Gaussian filter is the first step in applying Canny edge detector for edge detection.

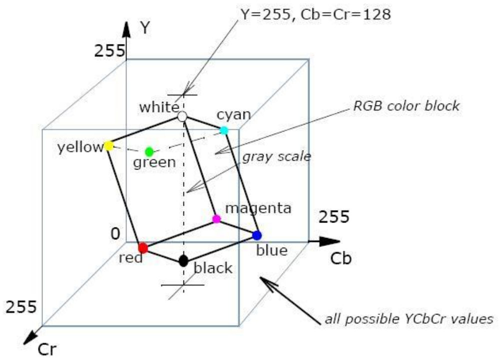
B. *Morphological operations*

After smoothing using Gaussian low pass filter we apply morphological operations in order to reduce the noise and improve the skin detection. For this purpose we apply dilation followed by erosion on the binary image using the same 3x3 structuring element for both operations. Dilation followed by erosion is known as "Closing" operation and it is represented using the following equation:

$$A \bullet B = \cup \{(B_z) | (B_z) \cap A \neq \emptyset\} \qquad (6)$$

Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape. In other words, the result of the closing operation is smoothing the edges and filling the small holds or connecting adjacent objects. Contrary to the opening operation, it can generally integrate the narrow gap, remove the small holes and fill the gaps on the contour.

VII. COMPARISON WITH YCBCR COLOR SPACE



Picture 3 3D model of YCbCr color space

YCbCr color space is consist of such components as Y (luma component) and CB and CR which are the blue-difference and red-difference chroma components. [6] For the image transformation from RGB color space to YCbCr color space the following equations should be used:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.400 \\ 1.000 & -0.343 & -0.711 \\ 1.000 & 1.765 & 0.000 \end{bmatrix} \begin{bmatrix} Y \\ Cb - 128 \\ Cr - 128 \end{bmatrix}$$

Both HSV and YCbCr color spaces allow separation of luminance component from chrominance component, so they can be used for color detection. Main difference is the coordinate system and way of calculation\transformation from RGB color space [6].

VIII. EDGE DETECTION USING CANNY EDGE DETECTOR

For edge detection we used the Canny edge detector. The Canny edge detector was developed way back in 1986 by John F. Canny. And it's still widely used today was one of the default edge detectors in image processing. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. [5]

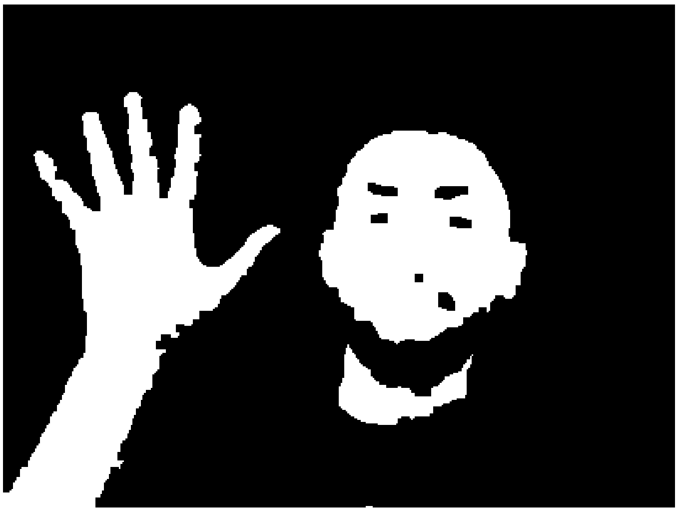
- Step 1: Smooth the image using a Gaussian filter to remove high frequency noise.
- Step 2: Compute the gradient intensity representations of the image.
- Step 3: Apply non-maximum suppression to remove "false" responses to to edge detection.
- Step 4: Apply thresholding using a lower and upper boundary on the gradient values.
- Step 5: Track edges using hysteresis by suppressing weak edges that are not connected to strong edges.

The problem is determining the optimal values for the lower and upper thresholds. The application allows two video feeds in the edge format. One of them uses directly the canny edge detector function with a wide threshold values (50 and 100) which gives the edges but also high frequency noise is present. The other uses a function that modifies the canny edge detector. This is done by taking the median of the image, and then upper and lower thresholds are constructed based on a percentage of this median. A lower value of sigma corresponds to a tighter threshold while a larger value gives a wider threshold. In our application a sigma=0.33 give satisfactory results.

IX. RESULTS

The following pictures represent the GUI and the obtained results:

Picture 3 Graphical User Interface



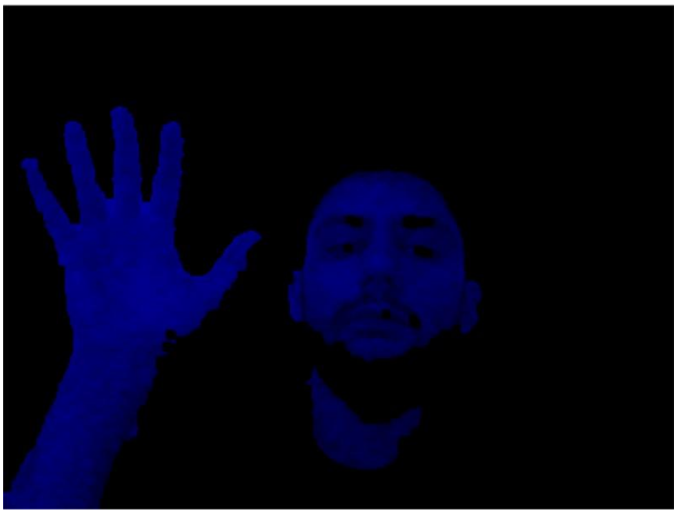
Picture 4 Binary mask using HSV

Picture 5 Green mask using HSV

Picture 6 Camera feed

Picture 7 Skin mask using HSV

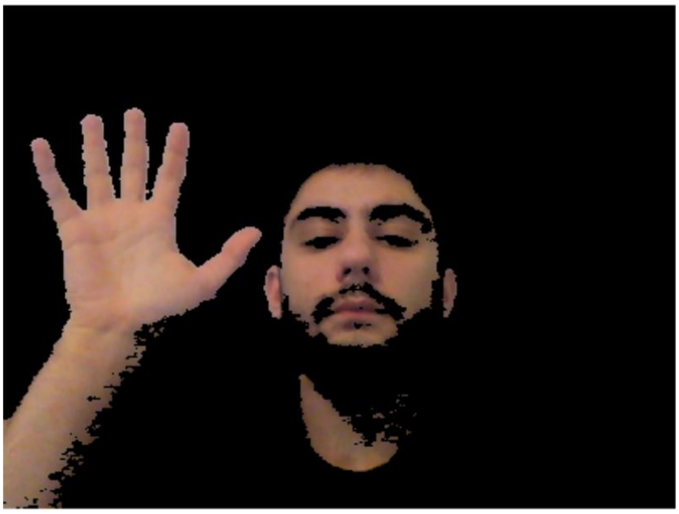
Picture 8 Red mask using HSV



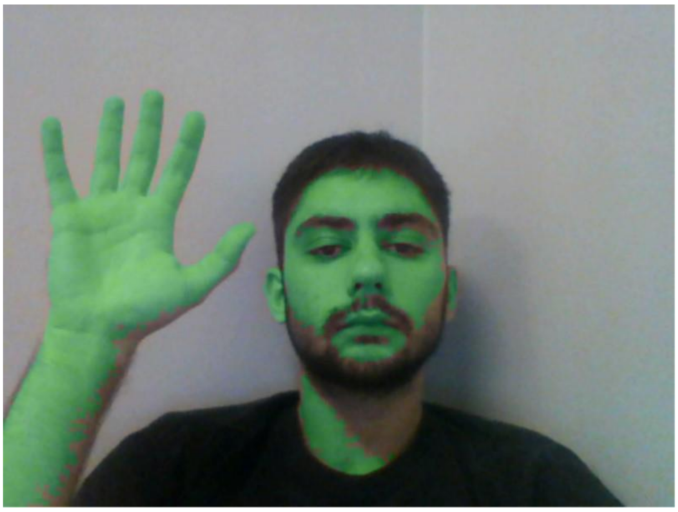
Picture 9 Blue mask using HSV

Picture 10 Colored skin using HSV

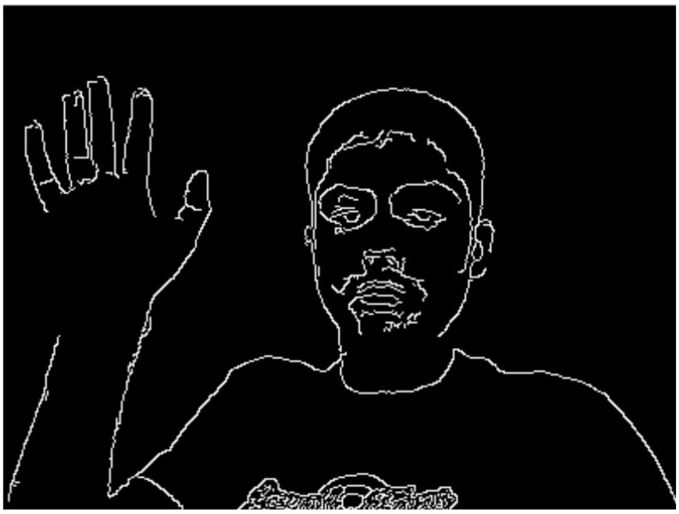
Picture 11 Edge detection using Canny edge detector



Picture 12 Skin mask using YCbCr



Picture 13 Colored skin using YCbCr



Picture 14 Edge detection using modified Canny edge detector

Picture 15 Colored edges

X. CONCLUSION

Skin detection and changing skin color can be done in both HSV and YCbCr color spaces. The result of skin detection using HSV or YCbCr color space is based on the selection of threshold value. As we can see from the equations the process of image transformation from RGB to HSV is more time consuming then transformation to YCbCr. That is why it better to use HSV based detection for images with uniform background. In case when image has a lot of fluctuation in color information YCbCr color space is suited better. For the created application and skin detection method we can say that:

- The simplicity of the used method makes it ideal from computational perspective and therefore it can be used for real time applications like video processing.

- In a controlled background and environment the application gives satisfactory results. Otherwise, some objects in the background with the similar skin-tone color may cause false detections.

- The usage of a better quality camera will result in a less noisy output.

- The result of HSV and YCbCr color space based skin color detection is based on the selection of threshold value.

- Tkinter is not the best Graphical user interface package because it requires numerous format conversions in order to display an image.

- The Canny edge detection function is simple and easy to implement in Python.

REFERENCES

[1] Ahmed Elgammal, Crystal Muang and Dunxu Hu, “Skin detection - a short tutorial”, unpublished.

[2] Vladimir Vezhnevets, Vassili Sazonov and Alla Andreeva “A survey of pixel based skin color detection techniques”, unpublished.

[3] V.A. Oliveira and A. Conci, “Skin detection using HSV color space”, unpublished.

[4] Dr. Aly A. Farag “Intruduction to Biometrics Lectures – Skin Detection”, unpublished

[5] Adrian Rosebrock, Zero-paratemeter, automatic Canny edge detection with Python and OpenCV, 09.11.2017, URL: www.pyimagesearch.com/2015/04/06/zero-parameter-automatic-canny-edge-detection-with-python-and-opencv/

[6] Khamar Basha Shaik, Ganesan P, V.Kalist, B.S.Sathish, J.Merlin Mary Jenitha, Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space, 12.01.2017, URL: http://ac.els-cdn.com/S1877050915018918/1-s2.0-S1877050915018918-main.pdf?_tid=89583eca-d908-11e6-bd9c-00000aab0f6c&acdnat=1484254316_b58dba5de9ff454446cc8be5e6dc3c09