

A New System for Extracting and Detecting Skin Color Regions from PDF Documents

Tarek Abd El-Hafeez

Department of Computer Science,
Faculty of Science, Minia University
El-Minia, Egypt

Tarek.Hemdan@Science.MiniaUniv.edu.eg

Abstract— There is no tool to block pornographic images included in the PDF documents. So, this paper proposes a system to extract images from PDF documents, and detect the skin color regions in the extracted images as a step to filter undesirable images from PDF documents. We studied two famous skin color detection techniques. The first technique is based on the log opponent color space and the second technique is based on the HSV color space. We modified the two techniques to overcome the failure of detecting complex image's background by combining each of the two techniques with the YIQ skin detection technique to obtain accurate skin detection results. Finally, the results of comparing the two skin detection techniques in terms of the number of pixels detected are presented.

Keywords— PDF Documents, Log opponent and HSV color spaces, PDF Image extraction, Skin detection.

1. Introduction

When PDF first came out in the early 1990s, it was slow to catch on. At that time, not only did the only PDF creation tools of the time (Acrobat) cost money, but so did the software to view and print PDF files. Early versions of PDF had no support for external hyperlinks, reducing its usefulness on the web. Additionally, there were competing formats such as Envoy, Common Ground Digital Paper, DjVu and even Adobe's own PostScript format (.ps). Adobe soon started distributing the Acrobat Reader (now Adobe Reader) program at no cost, and continued to support PDF through its slow multi-year ramp-up. Competing formats eventually died out, and PDF became a well-accepted standard [9].

In 2005 Microsoft presented a similar fixed-layout document format named XML Paper Specification (XPS). XPS is based on XAML, and is distributed along a royalty-free license. XPS support is scheduled to be included in Microsoft Windows Vista [11].

With the development of WWW, the dramatically falling cost of data storage and the advancing in coding technology are generating dazzling array of photography, animation, graphics, sound and video. Filtering the adult images is very important for search engines to avoid offensive content on the web. Now, there are some ways to stop naked images

arriving on computers, such as blocking unwanted sites, identifying images depicting naked or scantily dressed people. "X-Stop" is a tool to block pornographic sites. Forsyth and Fleck designed software to detect naked people [7]. This software begins by analyzing the color and texture of a photograph. When it finds matches for skin colors, it runs an algorithm that looks for cylindrical areas that might be corresponds to an arm or leg. It then seeks other flesh-colored cylinders, positioned at certain angles, which might confirm the presence of limbs. There is no tool to block pornographic images included in the PDF documents.

Skin color has proven to be a useful and robust cue for face detection, localization and tracking. Image content filtering, content aware video compression and image color balancing applications can also benefit from automatic detection of skin in images. Numerous techniques for skin color modeling and recognition have been proposed during the past years. Most existing skin segmentation techniques involve the classification of individual image pixels into skin and non-skin categories on the basis of pixel color [18]. It is not an easy task to extract regions of specific color from a given color image, since the color of an object varies with changes in illumination color, illumination geometry, and miscellaneous sensor parameters [5]. A few papers comparing different approaches have been published [1], [19], [23].

This paper proposes a system that scans the PDF document, extracts all images and saves them. Then it converts the extracted images to jpeg format. The system interface contains a button called "skin recognition" that is employed to detect the skin color region in human images. We studied two skin color images detection techniques. The first one is based on the log opponent color space and the second one is based on the HSV color space. We modified the two techniques to overcome the failure of detecting complex image's background by combining each of the two techniques with the YIQ skin detection technique to obtain accurate skin detection results. Finally, the results of comparing the two skin detection techniques in terms of the number of pixels detected are presented. This system is a part of a large system that is being developed to filter undesirable images from PDF documents.

This paper is organized as follows: section 2 describes the Portable Document Format (PDF). Section 3 describes the PostScript documents. In section 4, a description of the proposed system is presented. Section 5 describes the two skin color detection techniques. Finally, the experimental results are presented in section 6.

2. Portable Document Format (PDF)

Portable Document Format (PDF) is an open file format created and controlled by Adobe Systems, for representing two-dimensional documents in a device independent and resolution independent fixed-layout document format. Each PDF file encapsulates a complete description of a 2D document (and, with the advent of Acrobat 3D, embedded 3D documents) that includes the text, fonts, images, and 2D vector graphics that compose the document. PDF files do not encode information that is specific to the application software, hardware, or operating system used to create or view the document. This feature ensures that a valid PDF will render exactly the same regardless of its origin or destination (but depending on font availability when fonts are not encapsulated in the file) [11].

Anyone may create applications that read and write PDF files without having to pay royalties to Adobe Systems; Adobe holds a number of patents relating to the PDF format but licenses them on a royalty-free basis for use in developing software that complies with its PDF specification [10].

PDF files are most appropriately used to encode the exact look of a document in a device-independent way. While the PDF format can describe very simple one page documents, it may also be used for many pages, complex documents that use a variety of different fonts, graphics, colors, and images.

PDF is primarily the combination of three technologies:

- A sub-set of the PostScript page description programming language, for generating the layout and graphics,
- A font-embedding/replacement system to allow fonts to travel with the documents, and
- A structured storage system to bundle these elements and any associated content into a single file, with data compression where appropriate.

3. PostScript

PostScript is a computer language — more precisely, a page description language — that is run in an interpreter to generate an image. This process requires a fair amount of resources [14].

PDF is a file format instead of a programming language and for that reason it doesn't need to be interpreted. For instance,

flow control commands like if and loop are removed, while graphics commands such as *lineto* remain.

That means that the process of turning PDF back into a graphic is a matter of simply reading the description, rather than running a program in the PostScript interpreter. However, the entire PostScript world in terms of fonts, layout and measurement remains intact.

Often, the PostScript-like PDF code is generated from a source PostScript file. The graphics commands that are output by the PostScript code are collected and tokenized; any files, graphics or fonts the document references are also collected; and finally everything is compressed into a single file [11].

As a document format, PDF has several advantages over PostScript [14]:

- **Single file:** A PDF document resides in a single file, whereas the same document in PostScript may span multiple files (graphics, etc.) and probably occupies more space.
- **Already interpreted:** PDF contains already-interpreted results of the PostScript source code, so it is less computation-intensive and faster to open, and there is a more direct correspondence between changes to items in the PDF page description and changes to the resulting appearance of the page.
- **Object transparency:** PDF (starting from version 1.4) supports true transparency while PostScript does not.
-
- **Font substitution:** If displayed with Adobe Reader, a font-substitution strategy ensures the document will be readable even if the end-user does not have the "proper" fonts installed. PDF also allows font embedding to ensure that the "proper" fonts are displayed. While this is possible with PostScript, such files cannot normally be distributed freely because of font licensing agreements.
-
- **Independent pages:** PostScript is an imperative programming language (with an implicit global state), so instructions with one page can affect the appearance of any following page. It is therefore necessary to interpret all the preceding pages in order to determine the appearance of any given page. Each page in a PDF document is unaffected by any others.

Types of content

A PDF file is often a combination of vector graphics, text, and raster graphics, as shown in Figure 1. The basic types of content in a PDF are [12]:

- Text stored as such that can be easily copied to another program
- Vector graphics for illustrations and designs that consist of shapes and lines
- Raster graphics for photographs and other types of image

- Write custom converters.
- Color separate PDF.
- Perform automatic and preflight operations on PDF.
- Rotate pages, change the crop and media box.
- Embed TrueType, Type1, Type3, or composite fonts
- Merge/edit form data with PDF.
- Edit contents of PDF pages. Add/remove/edit images, text, and vector graphics. Change color spaces, fonts, line styles, and other attributes.
- Solid foundation for PDF viewers, editors, and software RIPs (Raster Image processors)

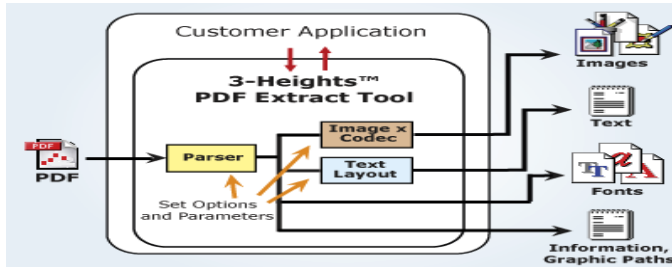


Figure 1: PDF document contents.

Tools exist, such as pdfimages (bundled with Xpdf) to extract the raster images from a PDF file. This can be useful if the PDF is a collection of scanned pages. Xpdf can also make raster images from complete pages that include vectors.

4. The Proposed System Description

PDFNet is a high-quality, industry-strength PDF library meeting requirements of most demanding and diverse applications introduced by *PDFTron Systems Inc.* Using *PDFNet* namespace, available at www.pdftron.com, we can perform the following tasks [13]:

- Append and assemble PDF documents. Merge specific PDF pages from multiple documents. Delete or rearrange pages.
- Change security settings. We can set new security handlers and edit or remove existing security.
- Create PDF from scratch. PDF can be created dynamically and delivered on the fly (e.g. on web and database servers). The library can also be integrated with client applications in order to provide enhanced PDF output that can't be produced using generic PDF writers (such as PDF print drivers or PostScript converters)
- Extract text.
- Extract fonts or use an easy to use API in order to access glyph outline data.
- Extract images. Images can be automatically converted and normalized to a target color space such as device RGB.
- Read bookmarks, annotations, and document metadata.

We use *PDFNet* namespace to extract the images from the PDF Document as a first step of our system. The second step of the proposed system is detecting the skin color regions in the extracted images.

The main purpose of the proposed system is extracting the images from the PDF documents and then detecting the skin color regions of these images. Using the proposed system, we can obtain various data about the PDF document such as the document title, document name, document author, and pages of the document. Also, we can obtain various data about the extracted images such as the number of the image in the document, the image height, and the image width.

The proposed system interface contains five buttons with captions "Open PDF Document", "Image Data", "Thumbnail Images", "Skin Recognition", and "Exit". Figure 2 shows the proposed system interface. The implementation of the proposed system is based on C#. Net.

The purpose of the "Open PDF Document" button is to browse and open the PDF document to be analyzed, then extract images from this document and convert these images to jpeg format. The purpose of "Thumbnail Image" button is to display all the extracted images from the document as a photo album, as shown in Figure 4. The purpose of "Skin Recognition" button is to recognize the skin in every image of the document. We use this button to determine the skin percentages of each image of the document as shown in Figure 5. Some filter techniques can be used with these images to prevent the undesirable images from displaying. The purpose of the "Image Data" button is to display the Image Data window as shown in Figure 3. Finally, the purpose of the "Exit" button is to exit the system.

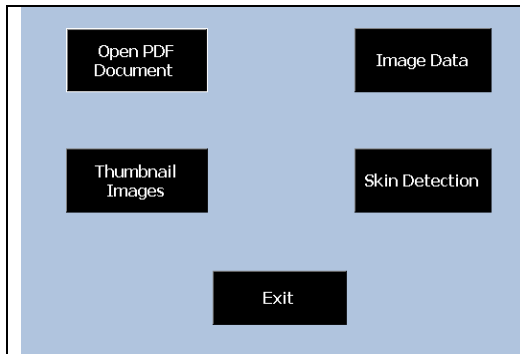


Figure 2: The proposed system interface.



Figure 3: The data of the extracted image.

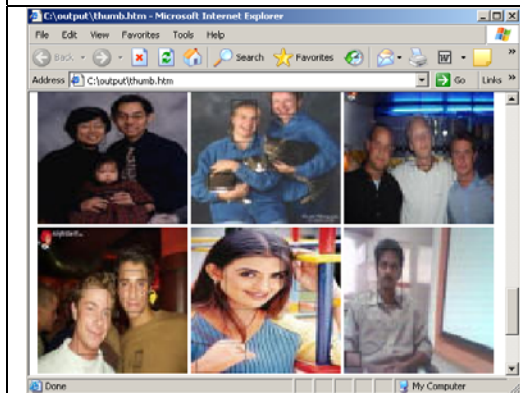


Figure 4: The image thumbnail of the extracted images.

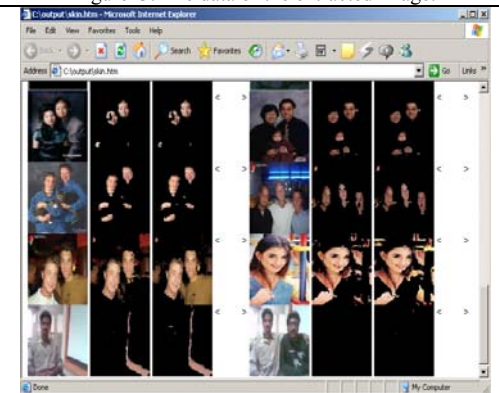


Figure 5: The skin detection window.

Figure 6 shows the analyzed PDF document considered in this paper.

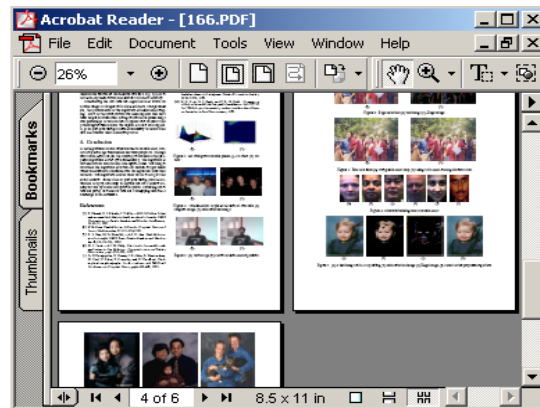


Figure 6: The analyzed PDF document.

The proposed system performs the following tasks:

1. Call the *Selection()* method to browse and choose the PDF document to be analyzed.
2. Call the *ImageExtract ()* method.
3. Convert the extracted images to the jpeg format.
4. For each image in the document do
 call the *Log_Skin()* method, or call the *HSV_Skin()* method, to detect skin regions.
5. Show the Images Data using "Image Data" button.

6. Show the extracted images as thumbnail using "Thumbnail Images" button.
7. Show the skin regions detected in step 4 using "Skin Detection" button.
8. Choose another PDF document to be analyzed or Exit using "Exit" button.

Figure 7 shows the algorithm for selecting the PDF document, to be analyzed by the *selection()* method, and extracting images from the selected document by the *ImageExtract()* method.

```

static void selection()
{
    PDFDoc doc = new PDFDoc(input); //input is the selected PDF
    document.
    for (PageIterator itr = doc.PageBegin(); itr != doc.PageEnd();
    itr.Next())
    {
        reader.Begin(itr.Current());           // reader is an object of
        ElementReader class.
        ImageExtract(reader);
        reader.End();
    }
}

static void ImageExtract(ElementReader reader)
{
    Element element;
    while ((element = reader.Next()) != null)
    {
        switch (element.GetType())
        {
            case Element.Type.e_form: // Process form XObjects
            {
                reader.FormBegin();
                ImageExtract(reader);
                reader.End();
                break;
            }
            case Element.Type.e_image:
            case Element.Type.e_inline_image:
            {
                //Construct the image data
                //Construct the Thumbnail images
                //Call two skin recognition Methods.
            }
            break;
        }
    }
}

```

Figure 7: The algorithm for selecting the PDF documents and extracting images from the selected document.

5. Skin Color Detection

Skin color detection is a very important step in many vision systems, like gesture recognition, hand tracking, video indexing, region of interest, face detection, etc. Pixel based skin detection can narrow the search space prior to high-level layers, however this is not an easy task. Skin pixels can vary with ambient light, such as color lamps acting as filters, brightness and specularities, shadows, daylight, etc. Since different cameras return different values for the same scene, pixel-based skin detection becomes a cumbersome task [2].

The main goal of skin color detection is to build a decision rule that will discriminate between skin and non-skin pixels. Identifying skin colored pixels involves finding the range of values for which most skin pixels would fall in a given color space. The purpose of a color space is to facilitate the specification of colors in some standard, generally accepted manner.

A color space is a specification of a coordinate system and subspace within a system where each color is represented by a single point. Most color spaces today are oriented toward

hardware such as color monitors or toward applications where color manipulation is a goal such as the creation of color graphics or animation [8]. Various color spaces are used for processing digital images. For some purposes, one color space may be more appropriate than others.

For skin detection, researchers do not quite agree on whether the choice of color space is critical to the overall performance of the detection system. One obvious reason for this is the lack of a standard set of images that can be used as a benchmark for algorithms using different color spaces. This may be as simple as explicitly classifying a pixel as a skin pixel if $R > G$ or $R > B$ or both [3] or may be as complex as models using neural networks and Bayesian methods [4]. In general, a good skin color model must have a high detection rate and a low false positive rate. That is, it must detect most skin pixels while minimizing the amount of non-skin pixels classified as skin. This section describes two known techniques for detecting skin color as a first step to filter undesirable images.

5.1 The Log Opponent Technique

The log opponent colour space are inspired by the human visual system that may be expressed in terms of the two opponent hues, yellow-blue and green-red, which cancel each other when superimposed [18]. A log opponent colour space for skin colour detection was suggested by Fleck et al. [7]:

$$L(x) = 105 \log_{10} (x + 1 + n)$$

$$I = L(G)$$

$$R_g = L(R) - L(G)$$

$$B_y = L(B) - (L(G) + L(R)) / 2$$

where n is a random noise value generated from a distribution uniform over the range $[0,1]$ and the constant 105 is used to scale the range to the interval $[255,255]$.

An alternative way of hue and saturation computation using log opponent values was introduced in [7], where additional logarithmic transformation of RGB values aimed to reduce the dependence of chrominance on the illumination level. While the meanings of hue and saturation remain unchanged, they are found using log opponent values. First, the RGB values are transformed into the log-opponent values intensity I , R_g and B_y by the following Equations:

$$I = \log(G)$$

$$R_g = \log(R) - \log(G)$$

$$B_y = \log(B) - \frac{\log(G) + \log(R)}{2}$$

The green channel represents intensity, because sometimes the red and blue channels from cameras have poor spatial resolution. The log transformation is intended to make the R_g and B_y values independent of illumination level. Finally, the hue and saturation are found using the Equations:

$$hue = \arctan(R_g, B_y)$$

$$saturation = \sqrt{R_g^2 + B_y^2}$$

The human skin color lies in the range $I < 6$ and $100 \leq hue \leq 150$.

This technique fails to cover complex backgrounds. We modified this technique by merging the Duan et al. method [6] with the log opponent color space to obtain more accurate results. First conversion from RGB to YIQ may be accomplished using the color matrix:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ 0.5957 & -0.2745 & -0.3213 \\ 0.2115 & -0.5226 & 0.3111 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where I is the red-orange axis, and Q is roughly orthogonal to I. The less I value means the less blue-green and the more yellow.

Duan et al. [6] found that the combination of YUV and YIQ color space is more robust than each other. The most people's skin varies in the range from 20 to 90 in the term of I. Our modification combine the two techniques to get more accurate regions of skin.

Hence, the modified method considers the skin color region lies in the range $20 \leq I \leq 90$ from YIQ color space and $100 \leq hue \leq 150$.

Figure 8 shows the results obtained by the log opponent technique before modification in the left column and after modification in the right column and Figure 9 shows the algorithm of the modified log opponent technique for detecting skin color regions.

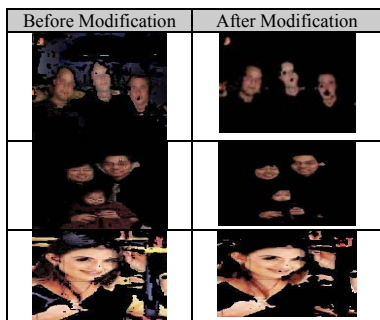


Figure 8: The left column shows the log opponent technique results and the right column shows our modified technique results.

```
public static void Log_Skin(Bitmap original1, ref Bitmap modified1)
{
    //detects skin. loops through the pixels in the image, if the pixel is skin
    then it leaves that pixel unchanged, or else it will
    // color that pixel black. (Modified LOG). First, convert RGB color
    space to IRgBy color space using this formula:
```

```
I = L(G)
Rg = L(R) - L(G)
By = L(B) - [L(G) + L(R)] / 2
double I = 0.596 * c.R - 0.274 * c.G - 0.322 * c.B;
//20 <= I <= 90 from YIQ Color Space
hue = atan2(Rg, By) * (180 / 3.141592654)
if (I >= 20 && I <= 90 && (hue >= 100 && hue <= 150))
{
    //skin;
}
else
{
    //non skin
}
}
```

Figure 9: The algorithm of the modified log opponent technique for detecting skin color regions.

5.2 Human Perceptual Color Spaces

The hue-saturation-intensity (HSI) family represents user-oriented color spaces, which are primarily used in computer graphics. The components are better suited for human interaction than, e.g., RGB, because they relate to the human perception of color, saturation, and luminance. The conversion from RGB to the HSI family is non-linear. An example on the conversion from RGB to HSV is given below.

The HSV (Hue, Saturation, Value/Intensity/Luminance) color space is defined as follows [15]:

$$V = \text{Max}(R, G, B)$$

$$S = \frac{D}{V}$$

$$H = \begin{cases} \frac{G-B}{2-R+B} & R=V \\ \frac{6D}{4-G+R} & G=V \\ \frac{6D}{6D} & B=V \end{cases} \quad (1)$$

where $D = \text{Max}(R, G, B) - \text{Min}(R, G, B)$

The HSV color space describes color with intuitive values, based on the artist's idea of tint, saturation and tone. This was introduced when there was a need to specify color properties numerically. Hue defines the dominant color as described by wavelength, for instance the distinction between red and yellow. Saturation measures the colorfulness of an area in proportion to its brightness such as the distinction between red and pink. Value refers to the color luminance, the distinction between a dark red and a light red. For skin detection, the value component is discarded to eliminate the undesirable effect of uneven illumination.

The transformation is defined by:

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

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

$$V = \frac{1}{3} (R + G + B) \quad (2)$$



Figure 10: The left column shows the HSV technique results and the right column shows our modified technique results.

Some studies show that HSV is invariant to highlights at white light sources, to matte surfaces, and ambient lighting. However, hue discontinuities and the computation of the luminance component conflict badly with the properties of color vision. The cyclic nature of Hue-Saturation spaces also makes it inconvenient for parametric skin color models that need a tight cluster of skin performance for optimum performance [21]. Tsekeridou and Pitas showed, in [20], that human skin-colours are positioned in a small cluster of the HSV space. Their suggestions to appropriate thresholds for hue saturation and value values were defined like this:

$$V \geq 0.4$$

$$0.2 < S < 0.6$$

$$0^\circ < H < 25^\circ \quad \text{or} \quad 335^\circ < H < 360^\circ$$

The selected range of H restricts segmentation to reddish colors and the saturation range selected ensures the exclusion of pure red and very dark red colors, both of which are caused by small variations in lighting conditions. The threshold on V is introduced to discard dark colors. These values for HSV are modified by Wang and Yuan [22], the new range values are:

$$0.35 < V < 1$$

$$0.20 < S < 0.68$$

$$0 < H < 50$$

This technique fails to cover complex backgrounds. We modified this technique by merging the Duan et al. method [6] with the HSV color space to obtain more accurate results. Duan et al. [6] found that the combination of YUV and YIQ color space is more robust than each other. The most people's skin varies in the range from 20 to 90 in the term of I.

Hence, the modified method considers the skin color region is in the range (20 ≤ I ≤ 90) from YIQ color space and (0.20 ≤ S ≤ 0.75), (V > 0.35) and (0 ≤ H ≤ 25).

Figure 10 shows the HSV technique results before modification in the left column and after modification in the right column. Figure 11 shows the algorithm of the modified HSV technique for detecting skin color regions.



```
public static void HSV_Skin (Bitmap original1, ref Bitmap modified1)
{
    //detects skin. loops through the pixels in the image, if the pixel is skin
    then it leaves that pixel unchanged, or else it will color that pixel black.
    double I = 0.596 * c.R - 0.274 * c.G - 0.322 * c.B;
    //20>=I>=90
    double V = Math.Max(c.R, Math.Max(c.G, c.B));
    double D = Math.Max(c.R, Math.Max(c.G, c.B)) -
    Math.Min(c.R, Math.Min(c.G, c.B));
    double S = D / V; double H = 0;
    if (c.R == V) H = (c.G - c.B) / (6 * D); if (c.G
    == V) H = (2 - c.R + c.B) / (6 * D);
    if (c.B == V) H = (4 - c.G + c.R) / (6 * D);
    if ((S >= 0.20 && S <= 0.75) && V > 0.35 && H >= 0 &&
    H <= 25 && I <= 90 && I >= 20)
    {
        //skin;
    }
    else
    {
        //non skin
    }
}
```

Figure 11: The algorithm of the modified HSV technique for detecting skin color regions.

6. Experimental Results

In this section, we present a comparison between the two skin recognition techniques described in section 5. For each method, we present a histogram showing the number of images (IMAGES axis) and the number of pixels detected (DETECTION axis) as skin or non-skin. Each technique is applied on a wide variety of images extracted from PDF documents taken under different lighting conditions and with different backgrounds. The images also have areas containing skin from other parts of the body like hands, and areas with color very similar to that of skin. These areas get classified as skin and they form skin regions accordingly.

Figure 12 describes the first technique (the modified Log Opponent) results for some variety of images, and Figure 13 describes the second technique (the modified HSV) results for the same images.

■ The skin pixels detected.

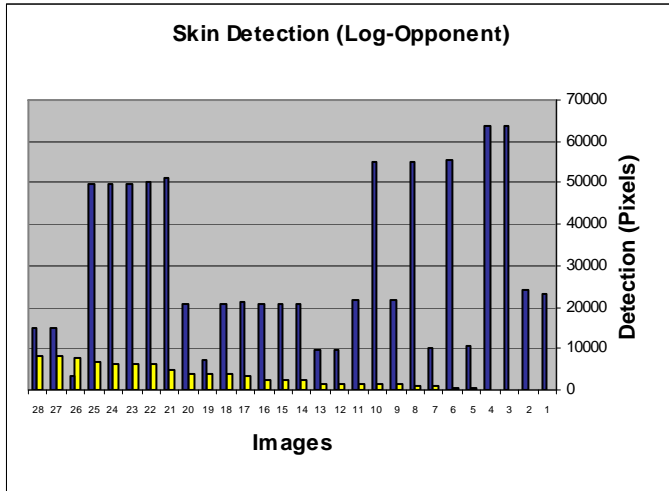


Figure 12: The first technique (the modified Log Opponent) results for various images.

■ The non-skin pixels detected.

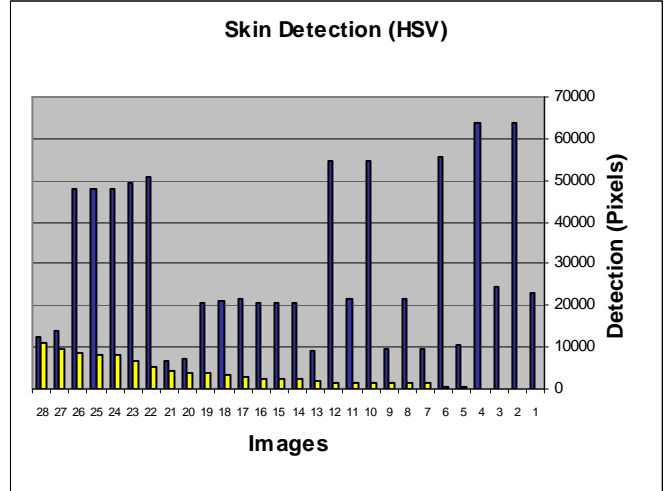


Figure 13: The second technique (HSV) results for the same images considered in Figure 10.

Figure 14 shows sample of accurate results of the modified (Log Opponent) and (HSV) techniques.

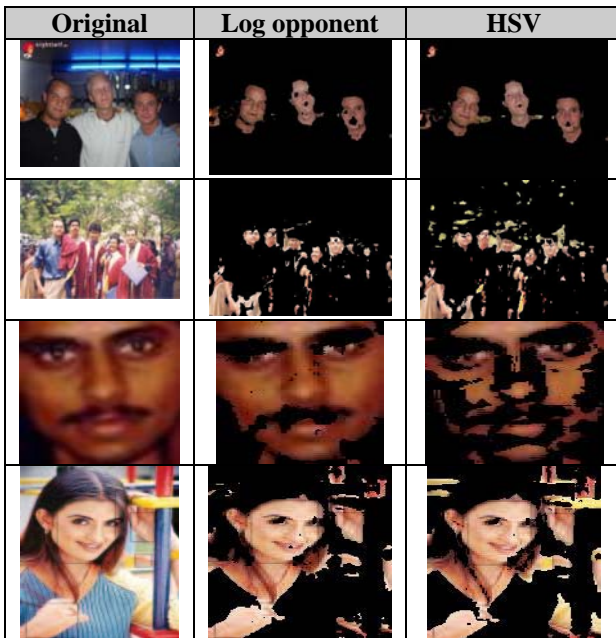
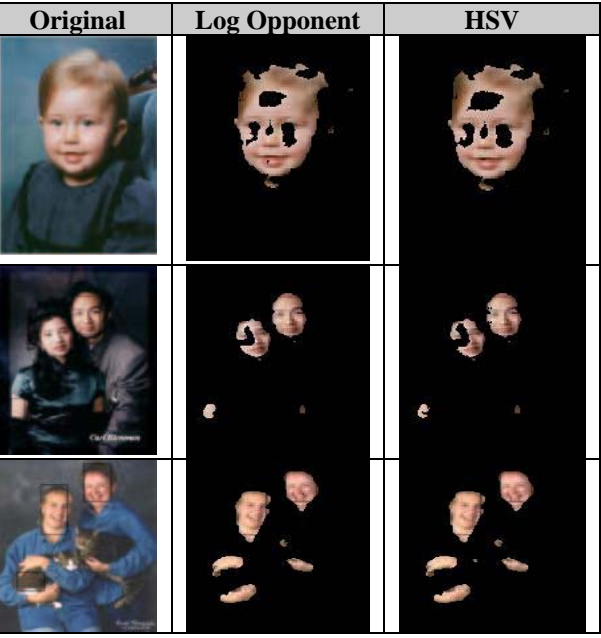


Figure 14: Samples of accurate results of the modified (Log Opponent) and (HSV) techniques



3- The modified Log Opponent technique is more efficient than the HSV technique in terms of skin pixels detected.

Thus, a skin color model should not treat light and dark skin in the same manner. This may be more computationally expensive but it allows for more skin pixels to be labeled correctly.

As can be seen in Figures 12, 13 and 14 :

- 1- Using a single color space may limit the performance of the skin color filter. This study shows that better performance can be achieved by using two or more color spaces.
- 2- Contrary to some studies, experimental results in this study indicate that chrominance values still significantly differ for light and dark skin even if the

CONCLUSION

This paper proposed a system to extract images from PDF documents and then detect the skin color regions of these images. We studied two famous skin color detection techniques. The first technique is based on the log opponent color space and the second technique is based on the HSV color space. We modified the two techniques to overcome the failure of detecting complex image's background by combining each technique with the YIQ skin detection techniques to obtain accurate skin detection results. Finally, the results of comparing the two skin detection techniques in terms of the number of pixels detected are presented. Future extension of this work includes more studies on the images background and lighting similar to skin color pixels to increase the accuracy of the skin color detection. Another extension of this work consists of using the proposed system as a filter to prevent displaying undesirable images in PDF documents.

REFERENCES

- [1] Brand, J., and Mason, J., "A Comparative Assessment of Three Approaches to Pixel level Human Skin- Detection". In Proc. of the International Conference on Pattern Recognition, vol. 1, 1056-1059, 2000.
- [2] Brand, J., Mason, S., Roach, M., Pawlewski, M., "Enhancing face detection in colour images using a skin probability map". Int. Conf. on Intelligent Multimedia, Video and Speech Processing, pp. 344-347, 2001.
- [3] Brown, D., Craw, I., and Lewthwaite, J., "A SOM Based Approach to Skin Detection with Application in Real Time Systems". In Proc. Of the British Machine Vision Conference, 2001.
- [4] Chai, D. and Bouzerdoum, A., "A Bayesian Approach to Skin Color Classification in YCbCr Color Space". In Proc. Of IEEE Region Ten Conference, vol. 2, 421- 4124, 1999.
- [5] Cho, M., Jang, H., and Hong, S., "Adaptive Skin Color Filter", Pattern Recognition, Vol. 34, pp. 1067-1073, 2001.
- [6] Duan, L., Cui, G., Gao, W., and Zhang, H., "Adult image detection method base-on skin color model and support vector machine". In Asian Conference on computer Vision, pages 797-800, Melbourne, Australia, 2002.
- [7] Fleck, M., Forsyth, D., and Bregler, C., "Finding Naked People". In Proc. of the ECCV, Vol. 2, 592-602, 1996.
- [8] Gonzales, R., and Woods, E., "Digital Image Processing," Prentice Hall, Inc, New Jersey, 2002.
- [9] "Portable Document format From Wikipedia, the free encyclopedia", http://en.wikipedia.org/wiki/PDF_format
- [10] "Legal notices for Developers", http://partners.adobe.com/public/developer/support/topic_legal_notices.html
- [11] "Portable Document format From Wikipedia, the free encyclopedia", <http://www.pdf-file-writer.com/>
- [12] "3-Heightstm PDF Extract API", <http://www.pdf-tools.com/asp/products.asp?name=EXPA>
- [13] "PDFNet SDK API reference", <http://www.pdftron.com/net/apiref.html>
- [14] "Evaluation Software download", <http://www.pdftron.com/downloads.html>
- [15] James, D. , Andries, V., Steven, K., and John, F., "Fundamentals of interactive computer graphics". Addison-Wesley, 2 edition, 1994.
- [16] Moritz Storrang, "Computer Vision and Human Skin Colour", Ph.D. Dissertation, Aalborg University, Faculty of Engineering and Science, 2004.
- [17] Peer, P., Solina, F., "An automatic human face detection method", in Proc. 4th Computer Vision Winter Workshop (CVWW), Rastendorf, Austria, Feb. 1999, pp. 122-130.
- [18] Phung, L., Bouzerdoum, A., and Chai, D., "Skin Segmentation Using Color Pixel Classification: Analysis and Comparison" , IEEE Tran. On Pattern Analysis and Machine Intelligence, Vol. 27, No. 1, Jan. 2005.
- [19] Terrillon, C., Shirazi, N., Fukamachi, H., and Akamatsu, S., "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, 2000.
- [20] Tsekeridou, S. and Pitas, I., "Facial Feature extraction in Frontal Views Using Biometric Analogies", Proc. of the IX European Signal Processing Conference, Vol. I, 315-318 (1998).
- [21] Vezhnevets, V., Sazonov, V., and Andreeva, A., "A survey on Pixel-Based Skin Color Detection Techniques". In Proceedings Graphicon '03, pp: 85-92, September 2003.
- [22] Wang, Y., Yuan, B., "A novel approach for human face detection from color images under complex background", Pattern Recognition, 34(10), 1983-1992, 2001.
- [23] Zarit, D., Super, J., and Quek, H., "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, 1999.