Image Chromatic Adaptation using ANNs for Skin Color Adaptation

P. Kakumanu† S. Makrogiannis† R. Bryll† S. Panchanathan‡ N. Bourbakis†

†ITRI/Department of Computer Science and Engineering
Wright State University
Dayton Ohio 45435 USA
{kpraveen, smakrogi, rbryll, bourbaki} @cs.wright.edu

‡Department of Computer Science and Engineering
Arizona State University
Tempe Arizona 85284 USA
panch@asu.edu

Abstract
The goal of image chromatic adaptation is to remove the effect of illumination and to obtain color data that reflects precisely the physical contents of the scene. We present in this paper an approach to image chromatic adaptation using neural networks (NN) with application for detecting - adapting human skin color. The network is trained on randomly chosen color images containing human subject under various illuminating conditions, thereby enabling the model to dynamically adapt to the changing illumination conditions. The proposed network predicts directly the illuminant estimate in the image so as to adapt to the human skin color. The comparison of our method with Gray World, White Patch and Neural Network on White Patch algorithms is presented. We also present our results on detecting skin regions in NN color corrected test images. The results are promising and suggest a new approach for adapting human skin color using NN’s.

Keywords
Image Chromatic Adaptation, Skin Color Adaptation, Neural Networks, Face Detection, CMCCAT2000.

1 Introduction
Color is the perception of light in the visible spectrum of electromagnetic radiation, incident upon the retina. The color of an object in an image is determined by the amount of light reflecting the surface of the object (spectral reflectance) and the amount of light incident on that surface (spectral radiance) [1]. Thus, a change in the level of illumination on an object changes the color of the object in that image. Human visual system can dynamically adapt to the varying lighting conditions and can approximately preserve the actual color of the object. This ability of humans to reduce the effect of light on the color of the object and to retain a stable perceptual representation of the surface color is referred to as color constancy or chromatic adaptation. However, unlike humans, image capturing devices such as digital cameras are not capable of adapting to the various illumination sources across scenes. This poses a problem for vision systems based on color information. Various applications such as content based image recognition systems [2, 3, 4] need to represent color constantly across different illumination conditions. Consider the recognition of human faces from images. Face recognition is the primary task in facial expression analysis, audio visual speech processing, facial biometrics and in many other human computer interaction domains. Skin color information can be considered a very effective tool for identifying/classifying facial areas provided that the underlying skin color model can only be properly adapted to various lighting conditions [5, 6]. Hence, an important step in recognizing objects using color information is to remove the effects of illumination from the object under consideration.

As a step towards accomplishing this goal, we present in this paper a simple approach to adapt human skin color using neural networks. The neural network is trained on a database of real images and estimates the illuminant of the skin color in the image. The proposed methodology uses the CMCCAT2000 chromatic adaptation transform [7] and the transformation is applied in LMS cone space. The skin regions in the NN stabilized images were successfully detected using a simple Euclidean distance between the pixel color values and the gray value in RGB space. The remainder of the paper is as follows: Section 2 gives an overview of the related work in image chromatic adaptation and skin color adaptation. Section 3 describes the general strategy to color correction, given a good estimate of the illuminant. Section 4 describes the proposed NN approach for skin color adaptation and three other approaches (Gray World, White Patch and Neural Network on White Patch) for the same. Sections 6 and 7 describe the results of the proposed method on skin stabilization and skin detection on a database of test images.
2 Related work

Estimating the illuminant is critical in solving the image chromatic adaptation problem. A number of strategies have been proposed to estimate the image illuminant. All these algorithms are based on the assumptions of either the existing camera characteristics or the illuminant properties or the distribution of the color values. Gray World algorithms [8, 9] assume that the average reflectance of the image(s) is gray and the illuminant is estimated as the color shift from the average gray value of the image. Retinex algorithms [10, 11, 12] try to model the human color perception system and estimate the illuminant by comparing the average color value at each pixel to the maximum value found by looking at a larger area (a patch) in the image. The gamut mapping algorithms [13] consider all the possible mappings between the set of color values under the known and unknown illuminants. The set of all possible color values under any illuminant is referred to as gamut. The unique solution set to this mapping can be obtained in different ways [13, 14, 15, 16]. In Bayesian color constancy [17, 18, 19] a maximum likelihood approach is used to determine the illuminant estimate so to maximize the likelihood of the observed data. In color balancing by image statistics [20], image histogram is used first to identify the images that require color correction and those that do not. A modified white balance algorithm is then used to color balance the images. In neural network based method [21], a NN is used to estimate the chromaticity of the illuminant. The input to the network is a binary value indicating the presence of sampled chromaticity. The output of the network is the expected chromaticity.

Skin color information has been an important cue in identifying human faces and body in images. To model skin color, a number of approaches use Gaussian Mixture Models (GMM) [22, 23, 24, 25]. The parameters in the GMM are normally estimated using the Expectation–Maximization (EM) algorithm. To detect skin pixels and non-skin pixels, [22, 23, 24] essentially apply a thresholding technique on the pixel’s skin color probability. Zhu et al. [25] used a Support Vector Machine (SVM) to classify the GMM as skin or non-skin pixels. In [6], Hsu et al. uses first a color correction step before applying skin detection. Both the color correction and skin detection were done in YCbCr space. A version of white patch algorithm is used for color correction. To detect faces, the skin color clusters are modeled by shape functions.

In this paper, our approach to skin detection consists of two stages. The first stage is to color correct the image in LMS cone space for skin color adaptation. The illuminant estimate is predicted by a Neural Network trained on image data by a back propagation algorithm. The second stage is to classify the skin pixels and non-skin pixels using a simple thresholding technique in RGB space on the achromatic value of the color corrected images.

3 Image Chromatic Adaptation Approach

Image chromatic adaptation transforms an image under an unknown illuminant to the corresponding colors under a known canonical illuminant. This consists of estimating the illuminant color and then to correct the image pixel-wise based on the estimate of the illuminant. In section 4, we describe four different algorithms (Gray World, White Patch and Neural Network on white patch and Neural Network on skin) for estimating the illuminant. Given a good estimate of the illuminant, color correcting an image is done using von Kries diagonal model of chromatic adaptation [26]. According to this model, the mapping of an image under an unknown illuminant to an adapted illuminant is obtained simply by scaling each channel independently as given by the following equation:

\[
\begin{bmatrix}
L' \\
M' \\
S'
\end{bmatrix} =
\begin{bmatrix}
K_L & 0 & 0 \\
0 & K_M & 0 \\
0 & 0 & K_S
\end{bmatrix}
\begin{bmatrix}
L \\
M \\
S
\end{bmatrix}
\]  

(1)

where \((L, M, S)\) and \((L', M', S')\) are the tristimulus values of the Long, Medium and Short cone responses of the unknown and adapted illuminants respectively and \((K_L, K_M, K_S)\) are the corresponding scaling coefficients.

The scaling coefficients can be expressed as the ratio of the cone responses to a white patch under the reference canonical illuminant and to those of unknown illuminant. In practice, the LMS values are approximated by the CIE XYZ tristimulus values or the image RGB values. However, it has been shown that the von Kries diagonal model holds in an optimal sense for chromatic adaptation, only if the sensor responses are expressed in LMS cone space. Hence a series of transformations as described below are applied to transform the image RGB values to LMS cone space.

The data obtained from a digital camera is in standard color space (sRGB). sRGB input values are non-linear gamma corrected RGB values [27, 28]. To obtain CIE XYZ tristimulus values from sRGB, we first need to linearize these values. Linearization of normalized sRGB values (in the range 0 to 1) is obtained by the following transformation:

\[
sRGB' = \begin{cases} 
sRGB/12.92 & \text{if } sRGB \leq 0.04045 \\ ((sRGB + 0.055)/1.055)^{2.4} & \text{otherwise} \end{cases} 
\]  

(2)

The conversion from sRGB’ to CIE D65 XYZ can be obtained by the following matrix transformation.

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
0.4124 & 0.3576 & 0.1805 \\
0.2126 & 0.7152 & 0.0722 \\
0.0193 & 0.1192 & 0.9505
\end{bmatrix}
\begin{bmatrix}
R_{sRGB'} \\
G_{sRGB'} \\
B_{sRGB'}
\end{bmatrix}
\]  

(3)
The conversion from CIE XYZ to LMS is obtained by the following CMCCAT2000 [29, 30] transformation matrix.

\[
\begin{bmatrix}
L \\
M \\
S
\end{bmatrix} =
\begin{bmatrix}
0.7982 & 0.3389 & -0.1371 \\
-0.5918 & 1.5512 & 0.0406 \\
0.0008 & 0.0239 & 0.9753
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

(4)

Once we obtain the LMS cone space responses for the corresponding sRGB values, color correcting the image is done by applying the scaling coefficients in equation 1. The sRGB values corresponding to these new color corrected cone space responses can then be obtained by the inverse application of equations 4, 3 and 2 respectively. Whatever features are used to represent the sensor responses, estimating the illuminant i.e., estimating the coefficients in equation 1 is the preliminary step in image chromatic adaptation. Estimating these coefficients for adapting human skin color is the central theme of this paper and is described in detail in sections 4-5.

4 Methodologies in the Framework

4.1 Gray World (GW) Method

The Gray World algorithm [8] is one of the simplest and widely used algorithms for estimating the illuminant of an image. It assumes that given an image with sufficiently varied colors, the average reflectance of the surfaces in the image is gray. Hence, any shift from gray of the measured sensor responses in the image, i.e., the illuminant. The scaling coefficients in equation 1 are averages of the sensor response \(s\) correspond to the color of the image. It assumes that given an image with sufficiently varied colors, the average reflectance of the surfaces in the image is gray. Hence, any shift from gray of the measured sensor responses correspond to the color of the illuminant. The scaling coefficients in equation 1 are therefore set to compensate this shift. To compute the sensor responses of the illuminant, we simply average all the sensor responses in the image, i.e.,

\[
R_{avg} = \frac{1}{N} \sum_{k=1}^{N} R_{im}(k), \quad G_{avg} = \frac{1}{N} \sum_{k=1}^{N} G_{im}(k), \quad B_{avg} = \frac{1}{N} \sum_{k=1}^{N} B_{im}(k)
\]

(5)

where \(N\) represents the number of pixels in the image, \(R_{avg}, G_{avg}, B_{avg}\) represents the average color value of the image and \(R_{im}, G_{im}, B_{im}\) represents the set of image pixels in RGB color space. The average gray value, \(gray_{avg}\) and then the scaling coefficients for the entire image are calculated using equations 6 and 7. Once the scaling coefficients were calculated, the color correction is applied in the normalized RGB space.

\[
gray_{avg} = 0.299 \times R_{avg} + 0.587 \times G_{avg} + 0.114 \times B_{avg}
\]

(6)

\[
\begin{bmatrix}
K_R \\
K_G \\
K_B
\end{bmatrix} =
\begin{bmatrix}
gray_{avg} / R_{avg} \\
gray_{avg} / G_{avg} \\
gray_{avg} / B_{avg}
\end{bmatrix}
\]

(7)

4.2 White Patch (WP) Method

The White Patch algorithm searches for a white patch in the image under the assumption that the brightest patch in the image is white. The chromaticity of the illuminant is the chromaticity of the white patch. There are many different versions of White Patch algorithms [10, 11, 12]. The version of the algorithm implemented here searches for the white patch reference in the image by selecting the pixel values over a fixed threshold value. The average sensor response of this white patch is adjusted to gray. The scaling coefficients for the entire image are obtained as the ratio of the average gray value of the selected white patch to that of the independent average responses of the white reference by using equations 6 and 7. The color correction is then applied in the normalized RGB space.

4.3 Neural Network (NN) Method

In this method, a neural network is trained to learn the estimate of the chromaticity of the illuminant. The network used is similar to that described by Cardei [21]. The network is a multilayer perceptron with two hidden layers. The input layer consists of 1600 neurons, the first hidden layer has 48 neurons, the second hidden layer has 8 neurons and the output layer has 2 neurons. The normalized input sRGB space is first transformed into rg chromaticity space, \(r = R / (R + G + B)\) and \(g = G / (R + G + B)\). The input space \((r, g)\) is divided into 40 * 40 (1600) discrete bins, each \((r, g)\) histogram bin corresponding to one of the input neurons. The input to the neuron is represented by 1 or 0 indicating that the chromaticity corresponding to the \((r, g)\) histogram bin is either present or not present in the image. The output of the network is the expected \((r, g)\) chromaticity of the illuminant in the image. The network is trained using back-propagation algorithm with learning and momentum rates of 10 and 1 respectively. The error function was the Euclidean distance in \((r, g)\) chromaticity space between the network’s estimate and the provided expected estimate of the image illuminant. The network inputs \((r, g)\) histogram bins) that have a non-zero input were marked active and only those active inputs were used during training. This pruning of the network [21] reduces the noise that may occur if the inactive inputs during training become active during testing.

We have two different network models. In the first model (NN on white patch), the expected output estimate of the illuminant is the color of the reference white patch. This model is trained to estimate the color of the white patch in the image. In the second model (NN on skin patch), the expected output estimate of the illuminant is the color of the facial skin in the image. This model is trained to estimate the skin color present in the image.
5 Experimental Results

The experimental data set consists of 326 images with the subjects face always in the upper left corner and with a white patch (to facilitate a reference for white patch). These images were captured by a digital camera, in and around the campus at WSU over a number of days and during various timings of the day to include various illuminating conditions. Out of these, 255 images were randomly selected to form the training set and the remaining 71 images form the test set. The image data is expressed in 2D normalized rg chromaticity space.

During training the neural network, the pixels with all the three sensor (color) values in the range 11-254 were selected. This filtering is done to remove too dark and too bright pixels. To provide adequate data for training the NN models, a random set of 30,000 pixels was drawn 20 times from each image in the training set. These random sets of pixels form the training sequence for the NN, creating an overall training set of 5100 (255 * 20) “images”. The training of the NN consisted of 200 epochs through the entire training sequence. The network is trained to estimate the color of the face (NN on skin patch). In this way, the NN algorithms were trained to bring either the white patch to perfect grey or to bring the average face color to perfect grey in each training image. The pruning of the network as described in section 4.3 resulted in 350 and 352 active neurons out of the 1600 inputs in each of these models respectively.

5.1 Skin Color Stabilization

Our goal firstly is to stabilize the skin color in the images. The Gray World algorithm brings the average color of the image to gray. The White Patch algorithm brings the average color of the reference white patch in the image to gray. The NN algorithms were trained to bring either the white patch or the average face color to gray. The performance of these algorithms is expressed in terms of the average color value \( \langle R_{\text{avg}}, G_{\text{avg}} \rangle \), average color distance \( d^2 \) and average pair-wise distance \( pd^2 \) over all the 71 test images. \( d^2 \) is measured as the Euclidean distance of the image pixel values from the average skin patch value, averaged over the entire test data. \( pd^2 \) is measured as the Euclidean distance of the image pixel values from all other possible image pixels in the entire test data, averaged over the entire test data set. These measures are defined by the following equations:

\[
R_{\text{avg}} = \frac{1}{N} \sum_{k=1}^{N} R_{im}(k), \quad G_{\text{avg}} = \frac{1}{N} \sum_{k=1}^{N} G_{im}(k) \tag{8}
\]

\[
d^2 = \frac{1}{N} \sum_{k=1}^{N} \left[ (R_{im}(k) - R_{\text{avg}})^2 + (G_{im}(k) - G_{\text{avg}})^2 \right] \tag{9}
\]

\[
pd^2 = \frac{1}{N^2} \sum_{i,j=1}^{N} \sqrt{(R_{im(i)} - R_{im(j)})^2 + (G_{im(i)} - G_{im(j)})^2} \tag{10}
\]

where \( N \) represents the total number of pixels in the skin patch over the entire dataset, \( R_{\text{avg}}, G_{\text{avg}} \) represents the average skin patch color value, and \( R_{im}, G_{im} \) represents the skin patch pixels in rg chromaticity space. Table 1 shows the performance of the various algorithms for skin stabilization. Column 2 represents the above described parameters for the original, unprocessed test images. As expected, the color variability is the greatest in the original images, since no color correction is applied. It is clear from the table that all the algorithms described for color correction improves the stability of the skin patch with respect to the original images. Column 4 represents the performance of the white patch algorithm. For this model, the parameters were computed after applying color correction on the test images by manually selecting the known white patch in these images. This method provides the best skin color stability as evident by the lowest value for parameters \( d^2 \) and \( pd^2 \), and hence is treated as reference data for other algorithms. This is the best possible color balance of the test images that can be obtained semi automatically. The NN trained on white patch (column 5) improves the performance by approximately 50% over the unprocessed images. However, the skin color stabilization is not as good as the reference white patch method. The last column represents the performance of the NN trained on skin patch. In this model, the NN is trained to bring the skin colored face patch to perfect gray in each image. The average color of

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Uncorrected Data</th>
<th>Gray World</th>
<th>White Patch</th>
<th>NN on White Patch</th>
<th>NN on Skin Patch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. color ( \langle R_{\text{avg}}, G_{\text{avg}} \rangle )</td>
<td>0.46, 0.29</td>
<td>0.422, 0.31</td>
<td>0.444, 0.302</td>
<td>0.44, 0.295</td>
<td>0.321, 0.334</td>
</tr>
<tr>
<td>Avg. color distance ( d^2 )</td>
<td>0.108</td>
<td>0.063</td>
<td>0.041</td>
<td>0.055</td>
<td>0.048</td>
</tr>
<tr>
<td>Avg. pairwise distance ( pd^2 )</td>
<td>0.152</td>
<td>0.091</td>
<td>0.059</td>
<td>0.081</td>
<td>0.07</td>
</tr>
</tbody>
</table>
the skin area is very close to perfect gray (the average r and g values for gray are 1/3, 1/3). Also, $d^2$ and $pd^2$ values are the second lowest in the table and although higher, are very close to the reference white patch method. From table 1, it is clear that the NN trained on skin patch produces better performance over the rest of the methods. Figure 1 and Figure 2 show some of the corresponding images obtained after applying the color correction. However, we need to note that applying color correction in LMS cone space resulted in perceptually natural images, though the quantitative results are the same as that of the scaling in normalized RGB space.

### 5.2 Skin Detection

From the discussion in section 5.1, it is clear that the NN color adaptation produces better performance for stabilizing the facial skin color. In this method, when the skin area is used as reference, the stabilization will attribute gray values to the skin area. This property can be used to detect the skin regions in the image. To detect

<table>
<thead>
<tr>
<th>UNCORRECTED IMAGES</th>
<th>GRAY WORLD</th>
<th>WHITE PATCH</th>
<th>NN ON WHITE PATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="UNCORRECTED IMAGES" /></td>
<td><img src="image2" alt="GRAY WORLD" /></td>
<td><img src="image3" alt="WHITE PATCH" /></td>
<td><img src="image4" alt="NN ON WHITE PATCH" /></td>
</tr>
</tbody>
</table>

**Figure 1** Color Corrected images obtained by different algorithms
the skin region as the achromatic region, we measure the color variance at each pixel in the color adapted image. The variance value at each pixel is calculated as follows:

$$\text{Var} = \frac{1}{3} \sum_{k=1}^{3} (I(k) - \bar{I})^2$$  \hspace{1cm} (11)

where $I(k)$ denotes the normalized color value in R, G and B channels for $k = 1, 2, 3$ respectively and $\bar{I}$ denotes the average color value. This variance measure denotes the Euclidean distance of the pixel color value from the axis of gray values in the RGB space and hence small color variance values denote the gray (or achromatic) pixels that correspond to the skin area. These variance values are normalized and inverted, and a thresholding operation is applied to separate skin pixels from non-skin pixels. Skin was detected by the color variance discussed above in equation 11 and with a thresholding value of 30. Figure 2 shows the skin regions detected by applying the above discussed thresholding technique on NN stabilized images.

<table>
<thead>
<tr>
<th>UNCORRECTED IMAGE</th>
<th>NN ON SKIN PATCH</th>
<th>DETECTED SKIN AREAS</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image16.png" alt="Image" /></td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
<td><img src="image21.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 2  Color Corrected images obtained by NN on skin patch and the detected skin regions


6 Conclusions and Future Directions

Skin color information is an important cue in various human face detection and human computer interaction applications. In this present work we have proposed a Neural Network approach to estimate the skin color. Training the NN to stabilize skin color has produced significantly better performance than the Gray World algorithm or the NN trained on white patch algorithms. A simple skin detection approach presented indicates that NN color adaptation can be used for skin color adaptation. The results presented here are an integral part of a larger project aimed at synthesizing perceptually realistic facial expressions. It is expected that the simple thresholding technique based on color information alone is not sufficient for face detection. For this reason, we are currently investigating other techniques such as clustering and LG graph procedures along with shape information for face detection. This particular capability of the proposed methodology for detecting human shape information for face detection. This particular such as clustering and LG graph procedures along with this reason, we are currently investigating other techniques

7 Acknowledgements

This research was partially supported by awards from NSF-ASU, AIIS Inc. and WSU/DAGSI.

8 References