Recibido: 17 de Agosto de 2010 Aceptado: 25 de Noviembre de 2010 Publicado en línea: 30 de Diciembre de 2010

Measuring perceived color difference using YIQ NTSC transmission color space in mobile applications

Yuriy Kotsarenko, Fernando Ramos
TECNOLOGICO DE DE MONTERREY, CAMPUS CUERNAVACA.

Resumen: En este trabajo varias fórmulas están introducidas que permiten calcular la medir la diferencia entre colores de forma perceptible, utilizando el espacio de colores YIQ. Las formulas clásicas y sus derivados que utilizan los espacios CIELAB y CIELUV requieren muchas transformaciones aritméticas de valores entrantes definidos comúnmente con los componentes de rojo, verde y azul, y por lo tanto son muy pesadas para su implementación en dispositivos móviles. Las fórmulas alternativas propuestas en este trabajo basadas en espacio de colores YIQ son sencillas y se calculan rápidamente, incluso en tiempo real. La comparación está incluida en este trabajo entre las formulas clásicas y las propuestas utilizando dos diferentes grupos de experimentos. El primer grupo de experimentos se enfoca en evaluar la diferencia perceptible utilizando diferentes fórmulas, mientras el segundo grupo de experimentos permite determinar el desempeño de cada una de las fórmulas para determinar su velocidad cuando se procesan imágenes. Los resultados experimentales indican que las formulas propuestas en este trabajo son muy cercanas en términos perceptibles a las de CIELAB y CIELUV, pero son significativamente más rápidas, lo que los hace buenos candidatos para la medición de las diferencias de colores en dispositivos móviles y aplicaciones en tiempo real.

Abstract: An alternative color difference formulas are presented for measuring the perceived difference between two color samples defined in YIQ color space. The classical color difference formulas such as CIELAB, CIELUV and their derivatives require many arithmetic transformations of the input values commonly given using red, green and blue components, making these formulas cumbersome to implement on mobile devices. The alternative formulas based on YIQ color space presented in this work are lightweight and can be quickly calculated, even in real-time. The comparison is made between the newly introduced formulas and the classical ones using two different sets of experiments. The first set of experiments is focused on evaluating the perceived difference of colors using different formulas for picking colors. The second set of experiments allows benchmarking of the color difference formulas to evaluate their performance when processing images. The experimental results show that the newly introduced formulas are close in perceived terms to CIELAB and CIELUV but are significantly faster, making them good candidates for measuring color difference on mobile devices and applications even in real-time.

KEYWORDS: COLOR DIFFERENCE, COLOR METRIC, YIQ, PERCEPTUAL UNIFORMITY, COLOR SPACE

Introduction

There are many applications where two or more colors need to be compared for some purpose. The examples of such applications include but are not limited to stereo vision algorithms, recognition algorithms used in surveillance, textile industry and image processing. As the number of mobile devices is growing, there is an increasing range of devices, each with unique set of processing capabilities. In the typical software applications running on the aforementioned systems the colors are usually defined by their respective red, green and blue values, also called RGB color space. A comparison of two sample colors usually involves calculating the distance or *difference* between them. If the sample colors have their respective coordinates specified as (r_1', g_1', b_1') and (r_2', g_2', b_2'), then the difference between the two colors can be calculated as:

$$E_{rgb} = \sqrt{(r_2' - r_1')^2 + (g_2' - g_1')^2 + (b_2' - b_1')^2} \qquad (1)$$

$$E_{rgb} = \sqrt{(\Delta r')^2 + (\Delta g')^2 + (\Delta b')^2} \qquad (1.1)$$

The values of r_1', g_1', b_1' and r_2', g_2', b_2' are specified in the range of [0, 1]. In some performance critical applications, the square root is omitted and the value of ΔE_{rgb}^2 is used instead.

The equation (1.1) is both easy to implement and fast to compute. However, the arbitrary variations of values in RGB space are not perceived as uniform by human eye. This fact led to the development of more perceptually uniform color spaces described later on in this work. In addition, not all colors visible to human eye can be modeled by the RGB color space [1]. The RGB color space itself can be defined in many ways depending on the three chosen primaries making the space dependent on the particularly used device. In the context of this work, the values of RGB color space are specified according to sRGB standard described in Rec. 709 [2].

In an informal article “Colour Metric” posted on CompuPhase web site on Internet [3], Thiadmer Riemersma suggested a variation to the equation (1), which apparently gives better results in perceptual terms and justifies it by the fact that the modification is used in their products such as EGI, AniSprite and PaletteMaker. The proposed modification is the following:

$$\Delta E_{ri} = \sqrt{\left(2 + \frac{\bar{r}}{256}\right) \cdot \Delta r^2 + 4\Delta g^2 + \left(2 + \frac{255 - \bar{r}}{256}\right) \cdot \Delta b^2} \qquad (2)$$

where $\bar{r} = (r_1 + r_2)/2$ and r_1, g_1, b_1, r_2, g_2 and b_2 values are given in the range of [0, 255]. According to Thiadmer Riemersma, the formula was derived from several tests using a program which displayed 64 colors palette and a sample color, that user had to match to an existing color on the palette; in addition, the program also suggested a matching color depending on one of the available color difference equations.

The equation (2) has an elegant form and can actually accept integer values for RGB components – a good point for mobile applications, but it lacks experimental data to justify the selection of weight coefficients and the program used by the author to deduce the equation (2) displayed a very limited palette to the users, making the reliability of the proposed formula inconclusive.

A color space that can encompass all the colors visible to the human eye has been proposed - the CIE 1931 XYZ color space, which is considered to be CIE standard. The selection of XYZ functions and the development of the color space were described by J. Schanda [4]. The coordinates specified in RGB color space can be transformed to CIE XYZ as the following [2], [5]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (3)$$

where X, Y and Z are the new coordinates in CIE XYZ color space and r, g and b are the linear RGB coordinates [5], [6]. The transformation from non-linear r', g' and b' coordinates to their linear counterparts is made by applying *gamma correction* [2], [6]:

$$L = \begin{cases} \frac{1}{12.92} V'_{sRGB}, & 0 \le V'_{sRGB} \le 0.03928 \\ \left(\frac{V'_{sRGB} + 0.055}{1.055} \right)^{2.4}, & 0.03928 < V'_{sRGB} \le 1 \end{cases} \quad (4)$$

where V'_{sRGB} should be replaced by the r', g' and b' values to obtain the linear values of r, g and b. The coefficients used in the equation (4) are specified according to Rec. 709 standard [7], but other specifications exist [2].

The CIE 1931 XYZ color space is primarily used for specifying other perceptually uniform color spaces. One of such color spaces that is also considered a CIE standard - the CIE 1976 L^*, a^*, b^* (CIELAB) color space, which can encompass all the colors visible to human eye and is somewhat perceptually uniform [2], [9]. The coordinates in CIELAB color space can be calculated from CIE XYZ values as the following [2], [4]:

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16 \quad (5)$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad (6)$$

$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \quad (7)$$

$$f(t) = \begin{cases} t^{1/3}, & t > (6/29)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29}, & \text{otherwise} \end{cases} \quad (8)$$

where $f(t)$ is a temporary transform function, X_n, Y_n, and Z_n are the coordinates of the particular white point used in the calculations. According to Rec. 709 [5] the white point D_{65} has the coordinates of X_n=0.9505, Y_n=1 and Z_n=1.0891 (calculated from their x and y coordinates of 0.3127 and 0.329 respectively on the CIE chromaticity diagram [1], [6], [7] [9]). The values of a^* and b^* can vary in [-100, 100] range for typical images, while the value of L^* is usually defined in the range of

[0, 100]. The color difference between a pair of two samples can therefore be calculated as:

$$\Delta E^*_{Lab} = \sqrt{\left(\Delta L^*\right)^2 + \left(\Delta a^*\right)^2 + \left(\Delta b^*\right)^2} \qquad (9)$$

There are many color difference formulas derived from CIELAB color difference (for instance, several variants are measured in [10]) and in many cases they can be generalized to the following form [10], [11]:

$$\Delta E^*_{gen} = \sqrt{\left(\frac{\Delta L}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2 + \Delta R} \qquad (10)$$

where ΔR according to M. R. Luo, G. Cui and B. Rigg [11] is an interactive term between chroma and hue differences (in many variants it is set to zero [10]), k_L, k_C, and k_H are parameters given by the user that according to M. Melgosa, J. J. Quesada, and E. Hita [10] depend on the experimental conditions based on the environment of the application; S_L, S_C, and S_H are the correction factors or so-called weighting functions that apparently improve the perceptual uniformity of the equation. The parameters ΔC^* and ΔH^* are variations in chroma and hue respectively, they can be calculated according to J. Schanda [4] (in a variant such as CIEDE 2000 they are calculated in similar way [12]):

$$C^*_i = \sqrt{\left(a^*_i\right)^2 + \left(b^*_i\right)^2} \quad i=1,2 \qquad (11)$$

$$\Delta C^* = C^*_2 - C^*_1 \qquad (12)$$

$$h^*_i = \tan^{-1}\left(b^*_i / a^*_i\right) \quad i=1,2 \qquad (13)$$

$$\Delta H^* = 2\sqrt{C^*_2 \cdot C^*_1} \cdot \sin\left[\left(h^*_2 - h^*_1\right)/2\right] \qquad (14)$$

The simplified equation using the aforementioned cylindrical values of *C** and *h** according to J. Schanda [4] is the following:

$$\Delta E_{LCH} = \sqrt{\left(\Delta L\right)^2 + \left(\Delta C\right)^2 + \left(\Delta H\right)^2} \qquad (15)$$

The equations (9), (10) and (15) represent the human perception of colors more closely (unless CIELAB is not really perceptually uniform [9]) and they provide a way of calculating color differences for all colors that can be seen by human eye (as opposed to RGB-based color difference formulas described earlier). However, they require three transformations from original RGB coordinates to CIELAB and equation (15) requires further transformation of the coordinates to cylindrical coordinate system, making the CIELAB equations cumbersome to implement on mobile devices, where arithmetical capabilities can be quite limited. The generalized equation (10) for the CIELAB color difference variants can be difficult to implement for the end user, who may not know the details of the development of the equation to be able to properly compute arguments ΔR, k_L, k_C,, k_H, S_L, S_C, and S_H. Thus, the equation (10) is not only cumbersome to implement, but also difficult to use. The equations based on CIELAB color space have successfully been used in still images but are not suitable for real-time processing [2].

An alternative perceptually uniform color space has been proposed as CIE standard – the CIE *L**, *u**, *v** (CIELUV) color space. The color coordinates in CIELUV color space can be calculated from CIE XYZ values as the following [2], [4]:

$$L^* = \begin{cases} 116(Y/Y_n)^{1/3} & Y/Y_n > (6/29)^3 \\ (29/3)^3 (Y/Y_n) & Y/Y_n < (6/29)^3 \end{cases} \quad (16)$$

$$u^* = 13L^*(u' - u'_n) \quad (17)$$

$$v^* = 13L^*(v' - v'_n) \quad (18)$$

where *u*' and *v*' are the parameters from CIE 1976 UCS Diagram [4], u'_n and v'_n are the coordinates of the reference white point. The coordinates *u*' and *v*' (and their respective white point values) can be calculated according to J. Schanda [4] as:

$$u' = 4X/(X + 15Y + 3Z) \quad (19)$$

$$v' = 9Y/(X + 15Y + 3Z) \quad (20)$$

The values of *u** and *v** may vary in [-100, 100] for typical images, while *L** is usually defined in the range of [0, 100]. Note that *L** in CIELUV space refers to the same *L** in CIELAB color space. The equation (16) for calculating *L** is slightly different from the equation (5) as they have been taken from different sources, but it can be determined graphically that they represent practically the same curve. In the works of C. Poynton [2] and J. Schanda [4] it can be verified that *L** parameter in both CIELAB and CIELUV is the same.

The color difference in CIELUV space can be determined as [13], [14]:

$$E^*_{Luv} = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2} \quad (21)$$

The CIELUV color space can also be defined using cylindrical coordinates similar to CIELAB as described earlier in this work, with its accompanying color difference equation but is not described here because of size constraints.

The equation (21) shares similar characteristics with the CIELAB color difference equation (9). It predicts the color difference between the samples close to how it can be seen by human eye, but it has the same weakness being difficult to implement on mobile devices with limited mathematical capabilities. In fact, the CIELUV equation (21) has implementation more complex than the CIELAB equation (9) because of intermediary values *u*', *v*' and their respective equivalents for the white point. According to E. M. Granger [9], the CIELUV space is a better alternative for measuring perceived color differences than CIELAB, suggesting further improvements to the luminance parameter *L**.

A more complex color difference metrics have been developed, such as DIN99 color space [15] with its own color difference formula and its derivatives [16]. The DIN99 color space is considered a German standard; its coordinates are calculated by transforming CIELAB coordinates using logarithmic functions and rotation. The DIN99 based color difference equations are more complex and require more mathematical operations, making them worse candidates for mobile devices and applications.

An alternative color model proposed by S. L. Guth called ATD Model for Color Vision [17] has a different approach to measuring color difference using two separate formulas: *short color difference* and *large color difference*. These two formulas are calculated by a series of transformations, starting with *x*, *y* and *z* chromaticity coordinates on CIE diagram (not CIE XYZ values), which are consecutively transformed to new set of coordinates in roughly five steps, after which the aforementioned color difference formulas can be calculated. The entire process uses a lot of multiplications and temporary variables. The color difference formulas based on ATD Model have their own advantages, such as being closer to human perception as suggested by E. M. Granger [9] and closely predicting MacAdam results [18] (parts on CIE chromaticity diagram which are perceived by human eye as the same color), making them good alternative when perceptually uniform color distance measure is critical. However, the intermediary calculations required for computing ATD color difference formulas make a significant obstacle for using ATD Model on mobile devices, especially in real-time.

Another approach in modeling colors and calculating the color difference has been described by M. Sarifuddin and R. Missaoui [19]. According to the authors, their new HCL color space is perceptually uniform (although further studies are required to justify this) and its coordinates can be calculated from non-linear (presumably) RGB coordinates:

$$Q = e^{\alpha\gamma} \tag{22}$$

$$\alpha = \frac{\min(r', g', b')}{\max(r', g', b')} \cdot \frac{1}{Y_0} \tag{23}$$

$$L_{saf} = \frac{Q \cdot \max(r', g', b') + (1 - Q) \cdot \min(r', g', b')}{2} \tag{24}$$

$$C_{saf} = \frac{Q \cdot (|r' - g'| + |g' - b'| + |b' - r'|)}{3} \tag{25}$$

$$H_{saf} = \tan^{-1}\left(\frac{g' - b'}{r' - g'}\right) \tag{26}$$

where Q is a luminosity tuning parameter, α is a temporary variable, $\gamma=3$ and $Y_0=100$. The values of r', g' and b' are given in range of [0, 255]. The value of L_{saf} is thought to have values in the approximate range of [0, 216], while H_{saf} takes values in the range of [$-\pi/2$, $\pi/2$] (although the authors provide a solution for redefining H_{saf} in the full rage of [$-\pi$, π]). The range of C_{saf} is not explained in the work. Therefore the color difference described in [19] is calculated as:

$$\Delta E_{HCL} = \sqrt{[A_L \Delta L_{saf}]^2 + A_{CH}[C_1^2 + C_2^2 - 2C_1C_2 \cdot \cos(\Delta H)]} \tag{27}$$

where $A_L=1.4456$ and $A_{CH} = \Delta H + 0.16^\circ$.

The main advantage of the equation (27) is that its underlying HCL color coordinates can be calculated directly from non-linear RGB values. The HCL color space also attempts to model the luminance *L* (although according to C. Poynton [2] it is misspelled, as the term *luminance* is better described by the approximation of L^*) in a way that it's perceived by human eye. However, according to C. Poynton [8], the components of red, green and blue having the same value do not generate the same perceived brightness. The latter means that L_{saf} is not actually the perceived lightness of the color, making the equation (27) perceptually non-uniform (that is, small changes in the resulting color difference does not represent small changes in perceived colors). The usage of trigonometric functions is another drawback of the equation as it may be difficult to compute on mobile devices.

The alternative color metric

In television a color space denoted YIQ has been introduced by the National Television System Committee (NTSC) [20], [19] for the video broadcast. The color space is composed of the component *Y* called luma [2], [8], [21], which is proportional to the gamma-corrected luminance of a color and two other components *I* and *Q*, the combination of which describes both the hue and saturation of color [21].

According to W. K. Pratt [21], the reasons for transmitting the YIQ components in television were that the *Y* signal alone could be used with existing monochrome receivers to display black and white video and that *I* and *Q* signals could have the transmission bandwidth limited without noticeable image degradation.

The values of *Y*, *I* and *Q* can be directly calculated from the non-linear *r*', *g*' and *b*' components in the following way [21]:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.29889531 & 0.58662247 & 0.11448223 \\ 0.59597799 & -0.27417610 & -0.32180189 \\ 0.21147017 & -0.52261711 & 0.31114694 \end{bmatrix} \cdot \begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} \qquad (28)$$

The reverse transformation from YIQ color space back to RGB is accomplished as:

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} 1.00000000 & 0.95608445 & 0.62088850 \\ 1.00000000 & -0.27137664 & -0.64860590 \\ 1.00000000 & -1.10561724 & 1.70250126 \end{bmatrix} \cdot \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \qquad (29)$$

The value of luma *Y* is defined in range of [0, 1], while *I* and *Q* are defined in ranges of approximately [-0.5, 0.5] for typical images. It is important to note that luma component *Y* although directly related to color brightness does not represent the luminance and should not be confused with *L** in the previously described CIELAB and CIELUV color models.

The color difference in YIQ color space would be the distance between two color points, which can be calculated as the following:

$$\Delta E_{yiq} = \sqrt{(\Delta Y)^2 + (\Delta I)^2 + (\Delta Q)^2} \qquad (30)$$

where $\Delta Y = Y_2 - Y_1$, $\Delta I = I_2 - I_1$ and $\Delta Q = Q_2 - Q_1$. However, in the above equation all three terms *Y*, *I* and *Q* are not equally proportional in perceived terms – the range of the possible values is different for each one of the components and the eye is more sensitive to the brightness of the color than to chroma [20]. In order to compensate for these irregularities three coefficients are introduced:

$$\Delta E'_{yiq} = \sqrt{k_Y (\Delta Y)^2 + k_I (\Delta I)^2 + k_Q (\Delta Q)^2} \qquad (31)$$

where k_Y, k_I and k_Q are the weighting coefficients to compensate for the proportions of the *Y*, *I* and *Q* coordinates.

The NTSC has allocated different bandwidth values for each of the signals [20] – approximately 4 MHz, 1.4 Mhz and 0.6 Mhz to *Y*, *I* and *Q* respectively. Based on that conclusion, the experimental weighting coefficients can be specified as $k'_Y = 4$, $k'_I = 1.4$, and $k'_Q = 0.6$. The first step would be processing the coefficients so that they sum to unity (this way their proportions can be seen):

$$k''_Y = \frac{k'_Y}{k'_Y + k'_I + k'_Q} \approx 0.6667 \qquad (32)$$

$$k''_I = \frac{k'_I}{k'_Y + k'_I + k'_Q} \approx 0.2333 \qquad (33)$$

$$k''_Q = \frac{k'_Q}{k'_Y + k'_I + k'_Q} \approx 0.1000 \qquad (34)$$

Since finding the above coefficients is a matter of tuning, during the experiments it was found that taking the square root of the above values gives better perceptual results. The resulting color difference equation with the weighting coefficients normalized is the following:

$$\Delta E''_{yiq} = \sqrt{0.5053 \cdot (\Delta Y)^2 + 0.299 \cdot (\Delta I)^2 + 0.1957 \cdot (\Delta Q)^2} \qquad (35)$$

In the performance critical applications, the color difference equation (35) can have its square root omitted and instead the value of $\left(\Delta E''_{yiq}\right)$ to be used. If each component of *Y*, *I* and *Q* is stored using 8 bits in the range of [0, 255], the alternative fast color difference equation using the fixed-point coefficients is the following:

$$\Delta E^2_{fixed} = \left[129 \cdot (\Delta Y_i)^2 + 76 \cdot (\Delta I_i)^2 + 50 \cdot (\Delta Q_i)^2\right] >> 8 \qquad (36)$$

where $Y_i = \mathrm{trunc}(Y * 255)$, $I_i = \mathrm{trunc}(128 + 256 * I)$ and $Q_i = \mathrm{trunc}(128 + 256 * Q)$. The values of I_i and Q_i are clamped to fit in the range of [0, 255] in the case of overflow.

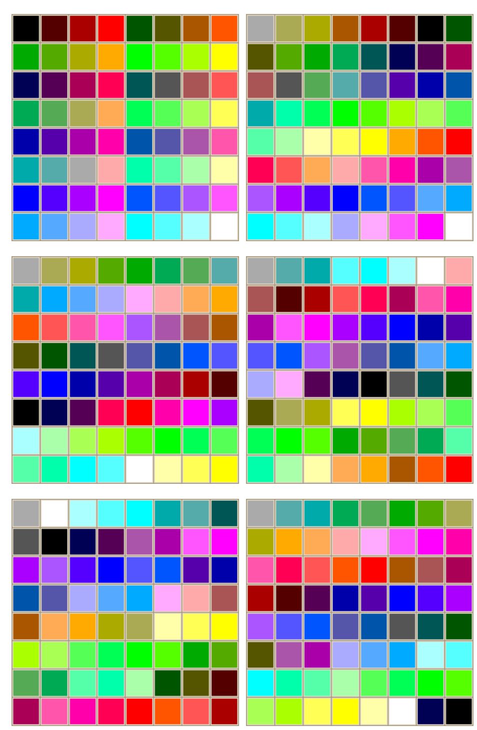
The equations (35) and (36) have a very compact form and are easy to compute on mobile devices. The general equation (31) and its specific variant (35) are to be implemented on systems powerful enough to work with real numbers, while the equation (36) is recommended for systems with severe arithmetic limitations. The resulting quality and performance of the equations is described below.

Analysis of Perceived Color Difference

The first set of experiments was focused on evaluating the perceived difference in colors calculated by the different formulas. The experiments consisted in two techniques: ordering a set of given colors with the minimal perceived difference to produce the softer gradient and construction of color maps by picking the closest color to the existing neighbors to produce a soft looking image. It is important to note that there is no exact solution to the tests described in this work and the color selection is subjective even to humans, so a reasonable balance is looked for. The judge of the experiments is, of course, the human eye. In the tests each of the equation is called either by the name of its color space (e.g. RGB, LAB, LUV, YIQ and so on), by its name (e.g. CIEDE2000) or by the author's name and the color space (e.g. Riemersma RGB).

The initial test was made using uniform palette of colors in RGB space with the combination of each *R*, *G* and *B* colors in four steps each. The original unsorted palette and the newly generated palettes are shown on the figure 2. The algorithm sorts the palette starting from the middle gray and continues by finding the next closest color match using one of the color difference formulas. Note that as the algorithms progresses, there are less colors left in the palette and thus less chance of success at the end. The resulting palette is a single horizontal row, which is, however, split in multiple rows on the illustrated images.

Judging the resulting palettes on the figure 2, it appears that Riemersma RGB is much better than RGB, while DIN99, ATD95 and YIQ are the best of the group. LAB, LUV and HCL made some good and bad choices, LUV being slightly better than LAB and HCL. ATD95 arranged colors perceptually quite well even although failed slightly in darker areas, while DIN99 made some mistakes in hue selection; YIQ made mistakes when comparing purple-red shades, while handling blue, green and orange shades quite well. CIEDE2000 performed slightly worse than ATD95, DIN95 and YIQ by making mistakes both in perceived brightness and hue.



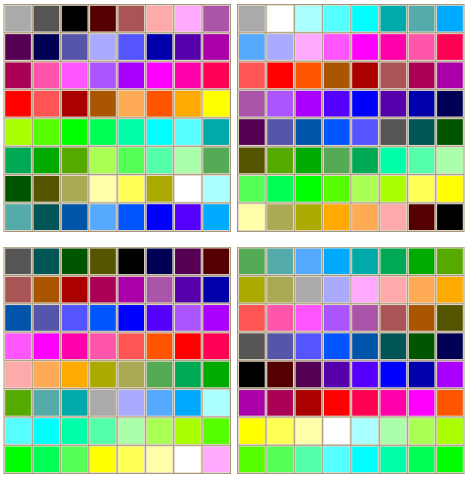


Figure 1. Sorting 64 colors in uniform palette set: original unsorted (a), RGB (b), Riemersma RGB (c), LAB (d), LUV (e), DIN99 (f), HCL (g), CIEDE2000 (h), ATD95 (i) and YIQ (h).

The next set of images has been made using a random palette of 64 colors using uniform random distribution. The resulting palettes are shown on the figure 3. The Riemersma RGB gives again better results than the original RGB, performing well with less saturated colors, while LAB and LUV give better results on less saturated colors but perform better with the proper hue ordering. DIN99 gives some really good choices as well as ATD95, with CIEDE2000 falling slightly behind. HCL properly detected color hues but didn't order them properly and performed worse with perceived brightness. YIQ handled well the changes of hue and perceived brightness, making mistakes, again, with colors having slightly different hues sufficient enough to be noticed, although in overall it performed quite well. The random palette made the color ordering more difficult by giving more saturated colors and unbalanced palette in terms of pure colors.

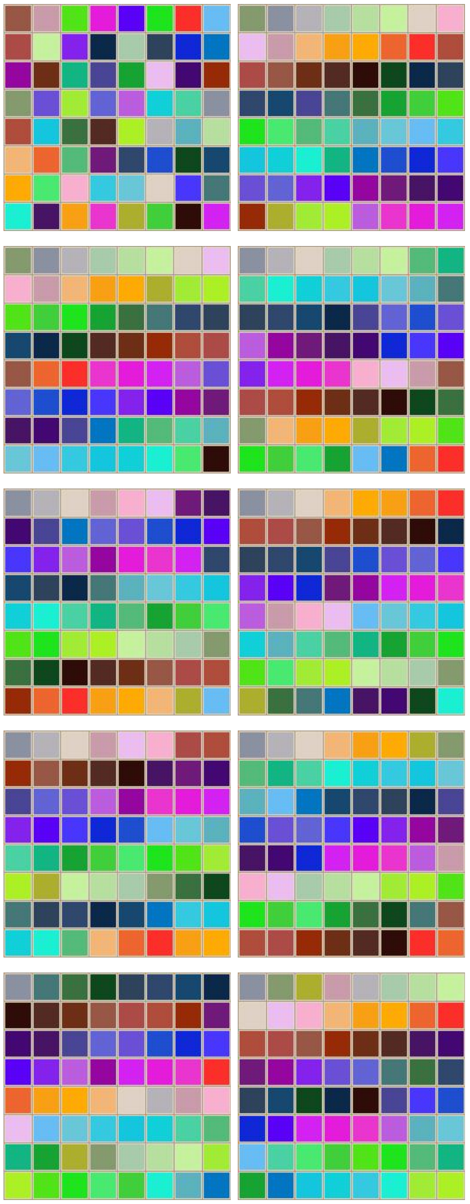


Figure 2. Sorting 64 colors in random palette set: original unsorted (a), RGB (b), Riemersma RGB (c), LAB (d), LUV (e), DIN99 (f), HCL (g), CIEDE2000 (h), ATD95 (i) and YIQ (h).

The next experimental technique consisted in hexagonal map, which is filled by picking colors from the source palette and placing them on the map [22]. The process begins in the center by gradually filling the map in circles. Although there might be more colors in the palette, the algorithm stops when the entire map has been filled. The first set of images has been generated using a random palette of 288 colors using uniform random distribution and the starting target sample in the center is black. The results are shown on figure 4.

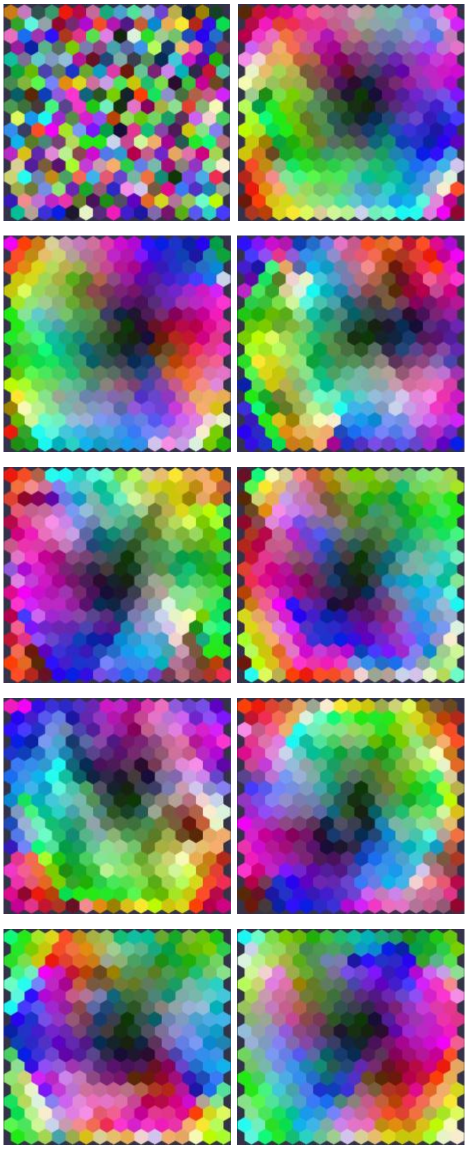


Figure 3. Accommodation of a randomly generated palette of 288 colors on the hexagonal map: original unsorted (a), RGB (b), Riemersma RGB (c), LAB (d), LUV (e), DIN99 (f), HCL (g), CIEDE2000 (h), ATD95 (i) and YIQ (h).

The uniqueness of this experiment is that the placed color is tested against all the already filled neighbors using color difference formula to achieve the less perceived difference, which shows the color difference between multiple colors instead of sample color pairs. From the results shown on the figure 4 it can be seen that RGB although performs better with several colors involved, still made some poor choices in the colors with the same perceived hue, while Riemersma RGB improved it quite a bit. LAB, LUV and HCL although grouped colors with the similar hue, the accommodated colors are quite different in perceived terms, making the outcome quite poor. DIN99 and CIEDE2000 performed quite well in most cases with the error growing gradually in the end. ATD95 in this scenario performed poorer especially in the darker colors. In this case, YIQ performed quite well also, especially in red and green hues, making some mistakes in the area of blue. In fact, LUV is the only formula that handled well blue colors on average.

The next set of images has been generated using uniform RGB palette with each of the component having five steps, making 125 colors in total. There are fewer colors in this experiment, creating bigger difference between colors in perceived terms. In addition, the initial color in the middle is white instead of black. The resulting set of images is illustrated on the figure 5.It can be seen that both RGB and Riemersma RGB performed well in the beginning with Riemersma RGB giving better results in the area of blue and cyan colors. Both LAB and LUV performed poorly with very little good choices. DIN99, CIEDE2000 and ATD95 performed surprisingly poorer than RGB and Riemersma RGB making some good and bad choices, where CIEDE2000 made the best choices in green-blue area. YIQ performed well in the area of perceived brightness (where many other algorithms failed), making mistakes with the colors having similar brightness but somewhat different hue. The last set of tests was particularly difficult for the color difference formulas, where the majority was guided by the perceived color hue resulting in large perceived mistakes.

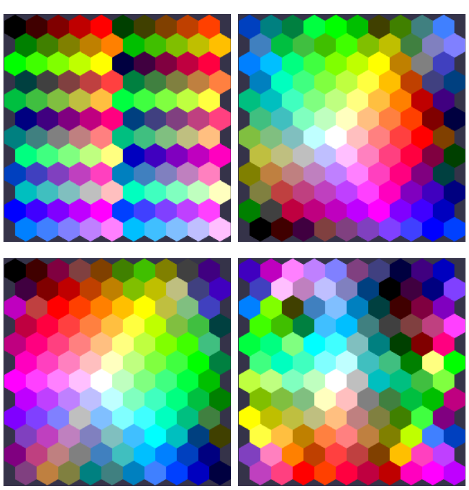


Figure 4. Accommodation of a uniform RGB palette of 125 colors on the hexagonal map: original unsorted (a), RGB (b), Riemersma RGB (c), LAB (d), LUV (e), DIN99 (f), HCL (g), CIEDE2000 (h), ATD95 (i) and YIQ (h).

From the results of the previously mentioned experiments it can be concluded that many color difference formulas perform well when used for the sample pair of colors but perform poorer when used in larger groups of colors. On average, LAB and LUV color difference formulas performed poorly when compared to the others such as DIN99, CIEDE2000 and ATD95. RGB and Riemersma RGB work better in groups than in sample color pairs. Apparently in the experiments, DIN99 was the best performer in terms of perceived differences. Surprisingly, YIQ in many cases performed quite well sometimes being one of the best in the group, although its weakness are colors with same perceived brightness and small hue difference sufficient enough to be perceived.

Performance Analysis

The color difference formulas used in the previous set of experiments have been also subjected to the performance evaluation. The evaluation consisted in ten image pairs with the size of 640x480 pixels filled with random color data. The algorithm called color difference formula between the pixels of each image pair calculating the time it takes to process all ten images. The pixel data of each image had colors specified both in sRGB and CIE XYZ color spaces – if the algorithm required a different color space (e.g. CIE LUV), it had to perform the necessary conversion on per pixel basis. The results are shown on the table 1.

Table 1. Benchmarks of different color difference formulas.

Formula	Total Duration (ms)	Frame Rate (640x480) Frames/second
RGB	258	**38.83**
Rie. RGB	297	**33.73**
LAB	3307	**3.02**
LUV	1763	**5.67**
DIN99	6299	**1.59**
HCL	2098	**4.77**
CIEDE2000	7231	**1.38**
ATD95	4430	**2.26**
YIQ	386	**25.91**
YIQi	125	**80.00**

The experiments have been performed on a workstation with the processor of *Intel Core 2 Quad Q6600* on bus of 1066 Mhz and memory *DDR2 800 Mhz RAM* with an average of 5 ms access time. The application ran in a single thread. From the results described on table 1 it can be seen that many color difference formulas can hardly be used for real-time processing considering that even on a relatively powerful machine the resulting frame rate is quite low even for a small resolution. The frame rate shown on the aforementioned table might be much less on mobile devices and their applications. The performance of RGB, Riemersma RGB and YIQ is quite acceptable, with integer variation of YIQ being the best candidate for real-time processing. It is important to note that integer variants of RGB and Riemersma RGB were not tested, but they are most likely to outperform the integer variation of YIQ.

Conclusions and future work

In this work a new color difference formula based on YIQ color space has been introduced. The formula has a balance between giving perceptually uniform results and being fast to compute. The classical color difference formulas such as those based on CIE LAB and CIE LUV are either too complex to compute on mobile devices, or do not produce the results in perceptual terms. The newly proposed color difference formula comes in two variations each with its own purpose – the high-precision variant useful in general computer systems and the integer variant using fixed-point arithmetic for computing the color difference quickly on more restricted processing equipment.

The experiments described in this work show that RGB based color difference does not produce adequate results in perceptual terms for practical use. The color difference formulas directly applied in CIE LAB and CIE LUV space are better than RGB, but fail in some circumstances. More complex formulas such as DIN99 color difference, CIEDE2000 formula and ATD95 large color difference produce better results in perceived terms with DIN99 color difference being the best on average in the group. The proposed color difference formula based on YIQ color space results in a balance between a perceptual quality being close to CIEDE2000 and DIN99 but is very fast to compute. It is proposed as a good candidate for mobile devices and applications.

There are areas where the newly proposed color difference formula can be improved. Specifically, the weighting coefficients k_Y, k_I and k_Q have been derived experimentally and analytically, but may not produce the optimal results in a particular set of conditions. More experiments are required for fine-tuning the weighting coefficients to produce better results.

The proposed color difference based on YIQ color space makes mistakes with colors having similar perceived brightness but hue different enough to be noticed. A potential deformation of I and Q components may improve the accuracy of the color difference formula.

References

[1] Hill, Francis S. *Computer Graphics using OpenGL.* Prentice Hall, 2000.

[2] Poynton, Charles. *Digital Video and HDTV Algorithms and Interfaces.* Morgan Kaufmann, 2003.

[3] Riemersma, Thiadmer. "Colour metric." *CompuPhase.* May 26, 2008. http://www.compuphase.com/cmetric.htm (accessed October 26, 2008).

[4] Schanda, Janos. *Colorimetry: Understanding the CIE system.* Wiley

Interscience, 2007.

[5] Daniel Malacara-Hernandez. “Color Vision and Colorimetry: Theory and Applications”, April 2002. SPIE Press Book.

[6] Henry R. Kang. “Computational Color Technology”. May 2006. SPIE Press Book.

[7] "Recommendation ITU-R BT.709-5." *Parameter values for the HDTV standards for production and international programme exchange.* International Telecommunication Union, 2008.

[8] Poynton, Charles. "Frequently-Asked Questions about Color." *Charles Poynton Web Site.* November 28, 2006. http://www.poynton.com/ColorFAQ.html (accessed October 22, 2008).

[9] Granger, E. M. "Is CIE L*a*b* Good Enough for Desktop Publishing?" *Proceedings of SPIE, the International Society for Optical Engineering* (Society of Photo-Optical Instrumentation Engineers) 2170 (1994): 144-148.

[10] Melgosa, M., J. J. Quesada, and E. Hita. "Uniformity of some recent color metrics tested with an accurate color-difference tolerance dataset." *Applied Optics* (Optical Society of America) 33, no. 34 (1994): 8069-8077.

[11] Luo, M. R., G. Cui, and B. Rigg. "The Development of the CIE 2000 Colour-Difference Formula: CIEDE2000." *Color Research & Application* 26, no. 5 (2001): 340-350.

[12] Sharma, Gaurav, Wencheng Wu, and Edul N. Dalal. "The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations." *Color Research and Application* 30, no. 1 (February 2005): 21-30.

[13] Imai, F. H., N. Tsumura, and Y. Miyake. "Perceptual color difference metric for complex images based on Mahalanobis distance." *Journal of Electronic Imaging* 10, no. 2 (2001): 385-393.

[14] Carter, Robert C., and Ellen C. Carter. "CIE L*u*v* Color-Difference Equations for Self-Luminous Displays." *Color Research & Application*, 1983: 252-253.

[15] Büring, Hendrik. "Eigenschaften des Farbenraumes nach DIN 6176 (DIN99-Formel) und seine Bedeutung für die industrielle Anwendung." *8. Workshop Farbbildverarbeitung der German Color Group*, October 2002: 11-17.

[16] Cui, G., M. R. Luo, B. Rigg, G. Roestler, and K. Witt. "Uniform Colour Spaces Based on the DIN99 Colour-Difference Formula." *Color Research & Application* 27, no. 4 (2002): 282-290.

[17] Guth, Lee S. "Further Applications of the

ATD Model for Color Vision." *Proc. SPIE* (The International Society for Optical Engineering) 2414 (April 1995): 12-26.

[18] MacAdam, David L. "Visual Sensitivities to Color Differences in Daylight." (Journal of the Optical Society of America) 32, no. 5 (May 1942): 247-273.

[19] Sarifuddin, M., and Rokia Missaoui. "A New Perceptually Uniform Color Space with Associated Color Similarity Measure for Content-Based Image and Video Retrieval." ACM SIGIR Workshop on Multimedia Information Retrieval, Salvador, Brazil, 2005.

[20] Hearn, Donald, and Pauline M. Baker. *Computer Graphics, C Version.* Prentice Hall, 1996.

[21] Pratt, William K. *Digital Image Processing.* 3rd Edition. Wiley-Interscience, 2001.

[22] Kotsarenko, Yuriy. “Quality measurement of existing color metrics using hexagonal color fields.” 7º Congreso Internacional de Cómputo en Optimización y Software, Cuernavaca, Mexico, 2009.

M. en Sc. Yuriy Kotsarenko se graduó como Ingeniero de Sistemas Computacionales en el ITESM, Campus Cuernavaca en Diciembre del 2004. Ahí mismo se graduó como Maestro en Ciencias Computacionales en el Diciembre del 2006 con la tesis “Perceptual Color Similarity Measurement”. Actualmente es alumno de posgrado, candidato a graduación como Doctor en Ciencias Computacionales en la misma institución. Su tesis doctoral esta asesorado por el Dr. Fernando Ramos Quintana. Su especialización incluye gráficos computacionales, visión computacional y óptica. También es experto en el desarrollo de software corriendo su propia empresa de desarrollo de software y video juegos.

Dr. Fernando Ramos Quintana es Director del Programa de Graduados en Informática y Computación del ITESM, Campus Cuernavaca. Su especialización incluye optimización combinatoria, comunicaciones inalámbricas, robótica cooperativa y bioinformática. Dirige múltiples proyectos de investigación, incluyendo investigación de modelos computacionales formales para la regulación genética, bioremediación de suelos, aguas y cultivos contaminados con plaguicidas utilizando la enzima esterasa recombinante de la garrapata Boophilus Microplus en la levadura Kluyveromyces Lactis, robótica médica entre otros.