Recibido: 14 de septiembre del 2011 Aceptado: 3 de enero del 2012
Publicado en línea: 8 de enero del 2013

Color spaces YCH and YScH for color specification and image processing in multi-core computing and mobile systems

Yuriy Kotsarenko, Fernando Ramos

Tecnológico de Monterrey, Campus Cuernavaca

ykot@inbox.com, fernando.ramos@itesm.mx

Resumen. En este trabajo dos nuevos espacios de color se describen para especificación de colores y procesamiento de imágenes utilizando la forma cilíndrica del espacio de color YIQ. Los espacios de colores clásicos tales como HSL y HSV no toman en cuenta la visión humana y son perceptualmente inexactos. Los espacios de colores perceptualmente uniformes como CIELAB y CIELUV son muy costosos computacionalmente para aplicaciones interactivas de tiempo real y son difíciles de implementar. Las alternativas propuestas, por otro lado, tienen un balance entre uniformidad perceptual, desempeño y simplicidad de cálculo. Estos espacios modelan colores de forma más exacta y son rápidos de calcular. Los resultados experimentales en este trabajo comparan espacios de colores clásicos con los propuestos en términos de uniformidad, riqueza de colores y desempeño, incluyendo numerosas pruebas de rapidez en procesadores de varios núcleos y sistemas móviles tales como ultra portátiles y los tablets tipo iPad. Los resultados evidencian que los espacios de colores propuestos son mejores alternativas para la industria de computación donde actualmente se utilicen los espacios de colores clásicos.

Abstract. Two novel color spaces are described for color specification and image processing using cylindrical variants of YIQ color space. The classical color spaces HSL and HSV do not take human perception into account and are perceptually inaccurate. Perceptually uniform color spaces such as CIELAB and CIELUV are computationally expensive for real-time interactive applications and are difficult to implement. Proposed alternatives in this work provide a reasonable balance between perceptual uniformity, performance and calculation simplicity. They model colors more accurately are fast to compute. Experimental results are provided, where the classical color spaces are compared to the proposed ones in terms of perceptual uniformity, color richness and performance, including numerous benchmarks on multi-core processors and mobile systems such as ultraportable computers and tablets such as iPad. The results provide evidence that the proposed color spaces are better alternatives for computer industry where classical color spaces are currently being used.

KEYWORDS: COLOR SPACE, PERCEIVED BRIGHTNESS, COLOR SELECTION, IMAGE PROCESSING, MOBILE SYSTEMS, PARALLEL COMPUTING, MULTI-CORE PROCESSORS.

Introduction

In image editing and manipulation software it is common to use HSV and HSL color spaces based on hue and lightness for image manipulation and intuitive color selection. These color spaces are so popular that they made their way into CSS3 specification [1]. The HSV and HSL color spaces were originally proposed by A. R. Smith back in 1978 [2]. They have an elegant way of color specification – using *hue*, *saturation* and *lightness* (or *value* in HSV). This specification makes it easy to define colors both numerically and visually using sliders, rings and so on. The values of *hue*, *saturation* and *lightness* can be directly calculated from the *red*, *green* and *blue* components. The conversion process was initially described by A. R. Smith [2] and can be found in popular books [3], [4].

However, the main drawback of the HSV and HSL color spaces is that they do not model colors as they are seen by the human eye. According to Charles Poynton [5], these color spaces are "useless for conveyance of accurate color information", suggesting that they should be abandoned. In his works [5], [6], Charles Poynton describes HSV and HSL models as flawed in respect to color vision and that they do not define the color objectively. The major drawback of the HSL color space is the *lightness* (or similarly *value* in HSV) component does not take into account the perception of the color brightness as it is perceived by the human eye.

In the context of color perception more uniform color spaces have been proposed

such as CIE XYZ, CIELAB and CIELUV [7]. Several formal definitions are necessary to compute the color components in these spaces from the *red*, *green* and *blue* values, such as the primaries specifying the RGB color space and white point coordinates [6], [7]. Other candidates for perceptually uniform color spaces include Guth's ATD95 Color Model [8] and DIN99 color space [9], [10]. All these color spaces share several characteristics – having the advantage of being more perceptually uniform with the major drawback of being cumbersome to compute and difficult to implement. The mathematical complexity of the aforementioned color spaces makes them a very expensive alternative for real-time processing, especially in mobile applications such as those running on ultraportable PCs, smartphones and tablets such as iPad. Another drawback is that the chroma component in the aforementioned models has range of values that varies depending on other values. This makes it difficult, if not impossible, to display the color wheel given a specific value of brightness because some areas of the color wheel will have invalid colors, being either out of range or not visible by the human eye at all.

An alternative color space inspired by HSV and HSL called HCL Color Space has been recently introduced by M. Sarifuddin and R. Missaoui [11]. The authors have taken into account the fact that the human eye reacts in a logarithmic manner to color intensity, making the color model more closely related to color perception. However, the HCL color space shares the drawbacks with the HSL and CIELAB color spaces and none of their advantages. The "luminance" (marked in quotes because according to Poynton the term *luminance* is commonly misused [6]) component *L* in the HCL color space is calculated using minimum and maximum RGB values, similarly to how it is done in HSL color space, making it inherently perceptually incorrect. The chroma component *C* has a non-normalized variable range, depending on the given "luminance" *L*, making its selection unintuitive. Last but not least, the conversion between RGB and HCL is cumbersome with some parameters not explained in sufficient detail (such as the origin of Y_0 parameter and the tuning of parameter γ).

The colors in the HCL color space plotted at constant values of *L*, namely 25%, 50% and 75% of maximum value, are illustrated on Fig. 1. The chroma at each position is calculated as $C = \sqrt{x^2 + y^2}$ and the hue is calculated as $H = \text{atan2}(y, x)$. The *atan2* function is an arctangent of *y*/*x* with the resulting angle in range of [-π, π] [12].

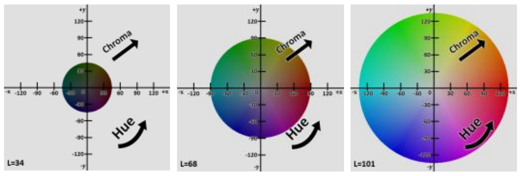

Fig. 1. Colors plotted in the HCL color space with *L*=34 (a), *L*=68 (b) and *L*=101 (c). Bright gray indicates that the resulting RGB values are out of the valid range.

It can be seen on the Fig. 1 that HCL color space looks similar to HSV, where different hue shades have different perceived brightness. That is, green appears brighter than red and blue appears darker. For a reference, the colors in the CIELUV space [7], [13] are displayed in Fig. 2 for the same percentages of the maximum value of luminance *L**.

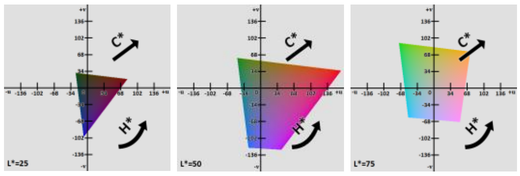

Fig. 2. Colors plotted in the CIELUV color space with *L**=25 (a), *L**=50 (b) and *L**=75 (c). Bright gray indicates that the resulting color is outside of sRGB color gamut.

The above figures show that different shades of color like green versus blue produce colors with similar perceived brightness (note that the printed version may slightly differ depending on the color space used by the printer). Comparing the figures of HCL and CIELUV color spaces it is evident that colors with the same *L* value in HCL color space do not appear with the same brightness. This problem occurs also in HSV and HSL color spaces, but not in CIELAB, CIELUV and DIN99, and their cylindrical variants. However, the range of chroma *C** in these color spaces (the distance from the center to any point on diagram) varies a lot – that is, with the specific values of chroma C^* and luminance L^*, different values of H^* may produce colors either inside or outside the *sRGB* color gamut [14].

The proposed coordinate system

Many different television standards share common characteristics, one such standard is the YIQ color space, introduced by the National Television System Committee (NTSC) [3], [15]. The particularities of this color space are that its *Y*, *I* and *Q* components can be calculated quickly right from RGB color values, and that the component *Y* called luma is proportional to the gamma-corrected luminance of a color [5], [6], [15]. That is, the luma models the perceived brightness more precisely than the "brightness" component in color models such as HSV, HSL and HCL.

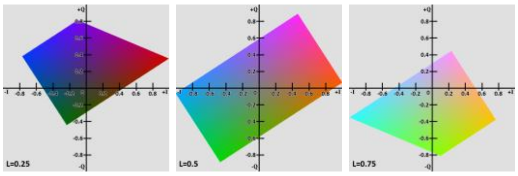

Fig. 3. Colors plotted in the YIQ color space with *Y*=0.25 (a), *Y*=0.5 (b) and *Y*=0.75 (c). The bright gray indicates that the resulting RGB values are outside of valid range.

Fig. 3 shows colors plotted in the YIQ color space with different values of the luma. The colors on each diagram have roughly the same perceived brightness, looking similar to the results obtained with perceptually uniform spaces such as CIELUV and CIELAB. In terms of hue, the diagram shows that a wide palette of colors is available; the question is how a user can choose a color in the YIQ color space using common terms such as *hue* and *saturation*.

As shown previously on Fig. 2 the cylindrical version of CIELUV uses terms of *chroma* and *hue* based on the angle between *u* and *v*. A similar approach can be applied to YIQ, introducing the terms *Y*', *C*' and *H*':

$$Y' = Y \qquad (1)$$

$$C' = \sqrt{I^2 + Q^2} \qquad (2)$$

$$H' = \operatorname{atan2}(Q, I) \qquad (3)$$

The parameters described by the equations (1), (2) and (3) form a new color space denoted *YCH*, with a cylindrical coordinate system. The parameter *Y*' is defined in the range [0, 1], *H*' is defined in the range [-π, π] and *C*' values range from 0 to 0.7925 in typical applications. The reverse transformation from *YCH* to YIQ color spaces is the following:

$$Y = Y' \qquad (4)$$

$$I = C' \cdot \cos(H') \qquad (5)$$

$$Q = C' \cdot \sin(H') \qquad (6)$$

However, as mentioned previously, the *YCH* color space suffers similar drawback as other color spaces such as cylindrical variants of CIELAB and CIELUV. Different values of hue with constant values of luma and chroma may produce both valid and invalid RGB colors. This is illustrated on Fig. 4.

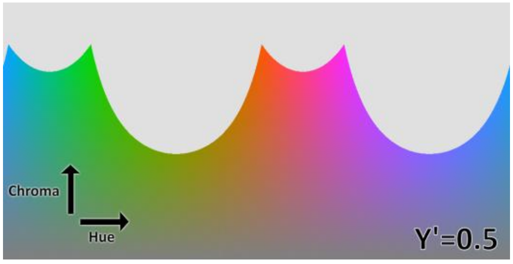

Fig. 4. Colors plotted in the *YCH* color space showing the dynamic range of chroma for a constant value of luma and different values of the hue.

Note that the shape shown on Fig. 4 looks different with other values of *Y'*. In order to display the color wheel or speak in terms of *saturation*, it would be convenient for the chroma to be normalized within a fixed range for different values of hue and luma. This way, the saturation will range from 0 to 1, for all the values of the hue and the luma, eliminating the area of invalid RGB colors. This issue is further discussed in the next section.

The Valid Chroma Algorithm

The normalization of valid chroma values, so that they fall within a fixed range, requires calculating the valid lower and upper bounds of the chroma values for the given set of luma and hue values. The lower chroma bound is zero, while the upper bound needs to be determined. This can be solved either analytically or numerically. Due to the fact that every color space is calculated differently, it is difficult to develop an algorithm that can be used in each scenario. It is possible to calculate the bounds using a brute force approach, but it would be too slow to be considered for practical applications. An alternative described in this work is an algorithm that is more efficient than the pure brute force search, but can still work with any given color space. It is based on the popular branch and bound algorithm [16], [17]. This algorithm can quickly determine the approximated upper chroma limit by continuously dividing the space in half until the reasonable precision is achieved. The algorithm can be described as a series of steps:

1. Set *L* to zero (lower bound), set *R* to one (upper bound).
2. Set *Result* to zero.
3. If $(R-L)<\varepsilon$, then go to step 9.
4. Set *M* to $(L+R)/2$.
5. If Valid(*L*) and not Valid(*M*) and not Valid(*R*), then set *R* to *M*, set *Result* to *Left*, go to step 3.
6. If not Valid(*L*) and not Valid(*M*) and Valid(*R*), then set *L* to *M*, set *Result* to *Right*, go to step 3.
7. If Valid(*L*) and Valid(*M*) and Valid(*R*), then set *L* to *M*, set *Result* to *Middle*, go to step 3.
8. If not Valid(*L*) and Valid(*M*) and Valid(*R*), then set *R* to *M*, set *Result* to *Middle*, go to step 3.
9. Return *Result*.

In the algorithm described above, *L*, *R* and *M* are variables of floating-point type, the function *Valid*(*x*) returns *true* if the color with the chroma *x* and constant values of *hue* and *luma* has a valid RGB entry, and *false* otherwise; ε is the desired precision factor. For the image resolution of 1024 by 512 pixels shown on Fig. 4, the color curve is seamlessly predicted with ε equal to 0.0079, requiring at most 7 cycles from step 3 to 8 in the worst case. The returning value, denoted as C_{max}, is the maximum value of chroma that in combination with hue and luma can be properly displayed within the sRGB color gamut or has valid RGB values if formal RGB color space is not defined. It is evident that the algorithm does not only work for the *YCH* color space, but can also be applied to other cylindrical coordinate variants such as those based on CIELAB, CIELUV and DIN99, although the function *Valid*(*x*) in these color spaces might be quite computationally expensive, making the algorithm less attractive in terms of performance.

The *saturation* of color defined in the range of [0, 1] can be calculated once C_{max} is known:

$$S_c = \frac{C'}{C'_{\max}} \quad (7)$$

where *C*' is the *chroma* of the given color in the *YCH* color space and $C'_{\max}$ is the value calculated using the valid chroma algorithm. If the value of *saturation* is given, the chroma can also be calculated as:

$$C' = S_c \cdot C'_{\max} \quad (8)$$

The equations (7) and (8) allow the color to be defined using the values of *Y*', *H*' and S_c, which always produce valid RGB values, assuming that values of *Y*', *H*' and S_c are within the proper ranges. In the context of this work, the color space with *Y*', *H*' and S_c coordinates is referred to as YScH.

The pseudo code for the proposed algorithm when applied to *YCH* color space is the following:

```
Left  = 0;
Right = 1;

while (Right - Left > Epsilon) do
 begin
  Middle = (Left + Right) / 2;

  IsLeft   = ChromaValid(Y, Left, H);
  IsRight  = ChromaValid(Y, Right, H);
  IsMiddle = ChromaValid(Y, Middle, H);

  if (IsLeft)and(not IsMiddle)and(not IsRight) then
   begin
    Right  = Middle;
    Result = Left;
   end;

  if (not IsLeft)and(not IsMiddle)and(IsRight) then
   begin
    Left   = Middle;
    Result = Right;
   end;

  if (IsLeft)and(IsMiddle)and(not IsRight) then
   begin
    Left   = Middle;
    Result = Middle;
   end;

  if (not IsLeft)and(IsMiddle)and(IsRight) then
   begin
    Right  = Middle;
    Result = Middle;
   end;
end;
```

Listing 1. The pseudo code for the valid chroma algorithm.

In the pseudo code in Listing 1 the function *ChromaValid* determines whether the color specified using the coordinates *Y*, *C* and *H* is a valid color. This can be achieved, for instance, by converting the color back to *RGB* coordinates and then verifying whether these coordinates are within the valid range. In the pseudo code it is assumed that the *Y* and *H* color components are valid, and remain constant.

Visual Characteristics

The color specification using *Y*', *H*' and *S*' coordinates provides an alternative way for color selection or image modification. The classical color wheel can easily be displayed with the variable hue at constant saturation and luma, as illustrated in Fig. 5. Note that, depending on the printing device and/or the monitor, the image may have different perceptual uniformity in hue – some areas appear smoother on paper while other look smoother on a computer monitor. This is due to the fact that different output devices using different color spaces to display images may reduce the visible palette, which is already limited, since all the colors visible by humans cannot be represented using valid RGB coordinates.

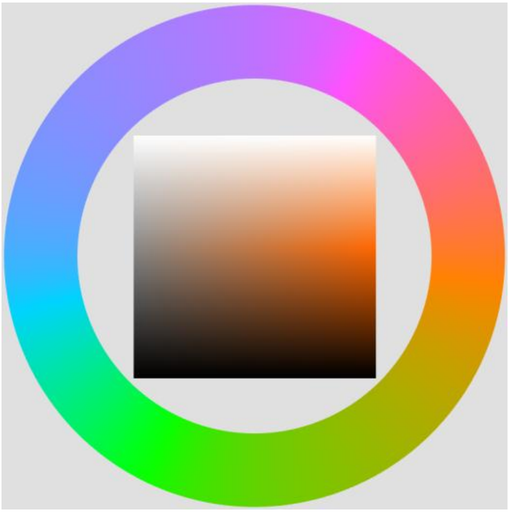

Fig. 5. Color wheel displayed using the YScH color space where *Y'*=0.6 and S_c=1.0. The rectangle in the middle has *H'*=0.

In Fig. 5 it can be seen that the blue and red shades have their perceived brightness closer to the green one as opposed to the classical color wheels of HSV and HSL. These color wheels are not shown in this work, but can be easily found in many popular image editing applications. The rectangle in the middle shows that given a certain color and saturation, changing the luma coordinate gives more perceptually uniform results than HSV and HSL color spaces.

Previously in Fig. 4, a diagram illustrating the variations of C and H at constant Y was characterized by areas of invalid RGB color values. A similar diagram for the YScH color space is shown in Fig. 6.

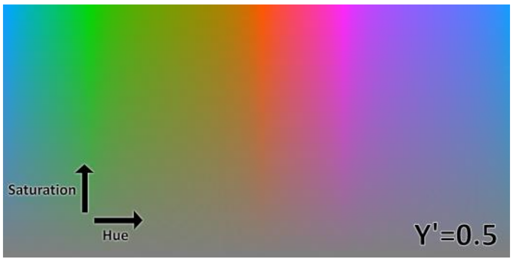

Fig. 6. Colors plotted in the YScH color space with variable hue and saturation, and constant luma.

It is evident in Fig. 6 that all colors specified in YScH are valid and, with a constant value of luma, have similar perceived brightness (something that does not happen with HSV and HSL color spaces, as can be seen in the next section). In addition, a more detailed color disks are shown in the next section, which display the same shades as those on Fig. 5 and Fig. 6 in a different arrangement, where it can be compared to HSV and HSL more interactively.

Analysis of Perceptual Uniformity

It was mentioned previously that the valid chroma algorithm can be also applied to other color spaces that use cylindrical coordinate system for specifying the color hue, the chroma and the brightness. The newly introduced color spaces YCH and YScH are fast to compute but their perceptual uniformity still remains to be established. Therefore, it is necessary to compare the results obtained with the proposed color spaces and with the classical ones. One experiment consists in drawing color circles divided into segments of the same size, each filled with a unique color. The hue is chosen by means of the radial angle of the segment and the brightness (luma, luminance or similar component depending on the color space used) is chosen as the distance of the color block to the center. Two images are displayed for each color space, one with a full saturation, the other with the half saturation. The evaluation is made by considering the perceived changes in brightness in the circle from the outside to the center, as well as the changes in hue between individual blocks. The smoother the change, the better is the result. In the color spaces such as CIELUV and DIN99 the valid chroma algorithm was used. In order to properly display the color disks for these color spaces, the white point D_{65} is assumed and the final color is represented in *sRGB* color gamut.

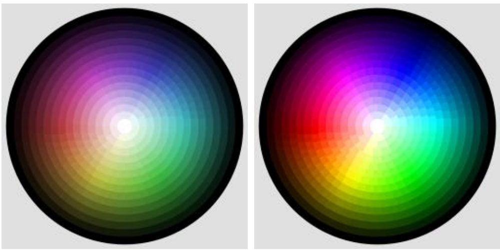

Fig. 7. The color disks plotted using the HSL color space with S=0.5 and S=1.0 respectively.

The first set of images shown in Fig. 7 shows the color disks using the HSL color space. The fully saturated image shows that the change in brightness and even in hue is quite non-uniform – there are three large spots of perceptually the same color (red, blue and green) where both the brightness and hue don't change much. There are noticeable jumps in hue, especially near the blue area.

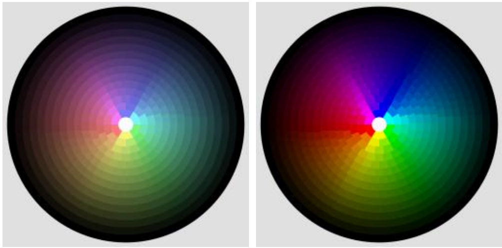

Fig. 8. The color disks plotted using the HCL color space with S_c=0.5 and S_c=1.0 respectively.

The images in Fig. 8 show the disks obtained with the recently introduced HCL space. They are very similar to those obtained with the HSV color space, not shown here, with the same problems as those pointed for the HSL color space.

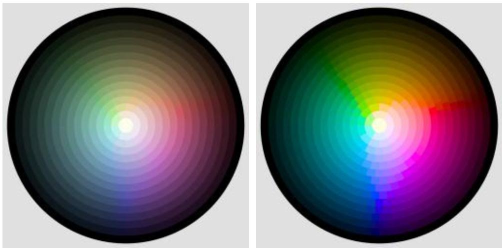

Fig. 9. The color disks plotted using the CIELUV color space with S_c=0.5 and S_c=1.0 respectively.

The disks obtained when using the CIELUV and DIN99 color spaces are shown on Fig. 9 and Fig. 10 respectively. The disk obtained with the CIELAB is not shown as appears quite similar. The images make it evident that both color spaces handle the perceived brightness very well, and better than the HSL and HCL color spaces. The change in hue is smoother too, although a large area on the disk is dedicated to blue, making the cylindrical variants of CIELUV and DIN99 (CIELAB is also included in this list) not suitable for showing color wheels as it dedicates too much space on the color wheel for a specific color, being, for this reason, insufficient.

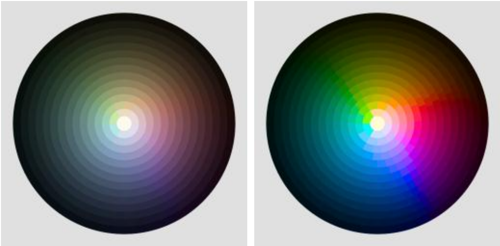

Fig. 10. The color disks plotted using the DIN99 color space with S_c=0.5 and S_c=1.0 respectively.

Finally, the color disks obtained with the proposed *YCH* (YScH) color space are shown in Fig. 11. The images show a good balance between the perceived brightness and the changes in hue, although an apparent rotation of red and blue shades can be seen in the darker areas.

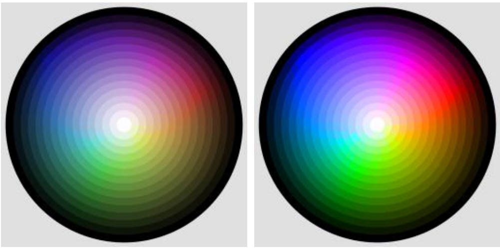

Fig. 11. The color disks plotted using *YCH* color space with S_c=0.5 and S_c=1.0.

If the above figures are viewed on a monitor that conforms to sRGB standard, a troubling observation can be made: the blue area in color spaces such as CIELAB, CIELUV, DIN99, and even *YCH,* appears to be actually brighter than the rest of hues with the same brightness component and saturation. This observation suggests that the discrimination of the blue in Rec. 709 [14] for the calculation of perceived brightness is too high, in other words, the blue does not appear as dark as red and green.

The observation of the HSL color disk shows that the colors corresponding to representative points positioned on the same circle of the color disk have different perceived brightness, which varies even more on the HCL color disk. This does not happen in the CIELUV and DIN99 color disks. The problem is very apparent in the area of blue shades, but it appears to be reduced in the YCH color disk, where the blue colors appear to have a perceived brightness similar to that of the other colors corresponding to representative points positioned on the same circle in the color disk.

The User Study

The experiments described in the previous section were backed up with the user study, where the color disks were shown to each of the participants either on a large TFT LCD screen, a computer monitor or on a printed paper. The display and the printing devices were carefully chosen to be fully compliant with the sRGB color space (at least according to their manufacturers) to ensure that the colors are properly displayed and the discrepancy between the brightness and hue on the different media is minimal. Each participant observed the color disks shown in the previous section and had to answer 19 questions in the quiz, 4 of which were control questions to verify the consistency of the answers.

In the quiz there were 3 categories that are classified as *color variety*, *color uniformity* and *general category*. The *color variety* category is mainly about the richness of the colors on the given color disk: the score depends on the number of colors that the participant can differentiate, so the disks with more perceived colors scored higher. The *color uniformity* category is used to determine which of the color disks appears more uniform in perceptual terms to each participant. The final *general category* is used to determine how well each of the color disks can be used for the color selection, being the source color palette for picking up colors.

In the user study a total of 52 people participated with the age ranging from 11 to 48 years old, with an average age of 25.7 and the standard deviation of 9.16. According to the people who specified their sex in the quiz, 23 were females and 22 males. The specialization that was specified by each participant included but not limited to administration, agriculture, arts, business, chemistry, commerce, computer science, education, electronics, foreign languages, history, industrial engineering, information technology, psychology and robotics.

The final results are shown in Table 1. It is important to note that although there was general consensus in the answers of the participants, there were few with the controversial results, where people gave higher scores to the *HCL* color space and lower scores to the rest. The participants did not know what color space was shown on each color disk, in the quiz they were marked by letters from A to E in the order as they appear in the table.

It is evident from the results that the *YCH* color space proposed in this work received higher scores on average in every category, with the *HSL* color space being in the second place. An interesting observation is that although the color spaces *CIELUV* and *DIN99* were developed to be perceptually uniform, the human observers gave higher scores in the *uniformity* category to the *HSL* and the *YCH* color spaces.

Performance Benchmarking

The classical color spaces were compared to the proposed *YCH and YScH* color spaces in terms of real-time performance on numerous computer systems using multi-threading approach, including 32-bit and 64-bit processors, including latest Intel Core i7 ® processors, ultraportable and embedded processors such as Intel Atom ® and embedded processors used in tablets such as iPad. In the experiments it was assumed that the source pixel of an arbitrary image is given using the *RGB* components. The conversion from the *RGB* components to the given color space was made; the *valid chroma algorithm* was applied whenever possible. The resulting color was then converted back to the *RGB* components. The listing of all systems used in experiments is shown in Table 2.

The first set of experiments involved 64-bit CPUs, which required operating system support and the application to be compiled for that target. The results shown in **¡Error! No se encuentra el origen de la referencia.** indicate that the proposed color spaces although marginally slower than HSL are significantly faster than classical perceptually uniform color spaces such as CIELUV and DIN99. The calculation times for 0.3 MP image indicate that the proposed color spaces can be used successfully in real-time on the majority of 64-bit systems.

It is important to note that HSL (and first variant of HCL) color space does not need valid chroma algorithm to be applied so it cannot be compared reliably to other color spaces. The *valid chroma algorithm* cannot be applied to the HSL color space because its saturation is always valid.

The second set of experiments involved 32-bit CPUs, including embedded and ultra-portable CPUs such as Intel Atom ® and tablet processor used in Apple iPad 2 ®. The results are shown in Table 4. Although more powerful processors present the performance close to the previous 64-bit experiments, less powerful processors show that YScH color space cannot be used in real-time, even though it does not take long to process (note that the results indicate that classical perceptually uniform color spaces are still out of question even for static image processing in these smaller CPUs). However, YCH color space can still be used in real-time in these situations. The difficulty regarding to YScH in such cases can be mediated by modifying the error factor to make it faster at the expense of precision.

Conclusions and Future Work

In this work, new color spaces *YCH* and *YScH* were proposed as alternatives for color specification and applications that do image processing on multi-core processors and mobile systems in real-time. These color spaces are based on YIQ transmission color space transformed into cylindrical coordinate system. The advantage of the proposed alternatives over classical color spaces such as HSV and HSL is better perceptual accuracy, while being significantly faster than other well-known color spaces such as CIELAB and CIELUV. The benefits of using the proposed alternatives are evident in terms of perceptual quality as backed up by statistical user study presented in this work, where contestants had to qualify each of the presented color disks in several categories; the results have shown in average the preference in all three categories for one of the presented alternatives instead of the classical color spaces. The calculation of YCH and YScH coordinates is easy and simple to implement as opposed to classical alternatives such as CIELAB, CIELUV and DIN99 among others, which reduces development time and costs.

The valid chroma algorithm described in this work uses a branch and bound technique to constrain the saturation value to stand within a predefined interval, making the color specification more user-friendly. Although the algorithm can work with other color spaces that use cylindrical coordinate system, it is particularly useful for the proposed *YCH* color space (therefore converting it to YScH) as the performance hit is relatively low when compared to well-known color models such as CIELUV and DIN99. Furthermore, it can be used with CIELAB, CIELUV, DIN99 and other color spaces to expand chroma component to a fixed range for convenient usage.

The benchmarking experiments described in this work show that when using multi-threaded approach on multi-core processors and mobile systems the proposed alternatives are excellent candidates as large images can be processed in real-time (or very close to it) even on ultraportable computers and ultrathin tablets. Therefore, the proposed color spaces are recommended as replacements in future applications and those that already use classical color spaces such as HSV and HSL.

Finally, an important observation is that the color disks shown in this work and their user study suggest that CIELUV and DIN99 color spaces are not very well suited for color specification since they received lower scores in almost every category in the quiz, being worse than *YScH* and *HSL* color spaces. It appears that CIELUV and DIN99 color spaces on their disks dedicate too much space for blue color tint. This can be a drawback when working with color specification and when processing image pixels as the bias will be shifted into blue segment. Further experiments are required to confirm this observation. The Guth's ADT color model and HSP color space proposed by Darel Rex Finley [18] and its derivatives [19] should be also evaluated both in terms of quality and performance to see how they compare with the results presented in this work.

Acknowledgements

Special thanks to M. Sc. Rafael Rojano Caceres for his invaluable help with the user study, who applied the majority of the quiz in the "Universidad Veracruzana (UV)" in the city of Xalapa-Veracruz, México.

References

[1] Pemberton, Steven, and Brad Pettit. "CSS Color Module Level 3." *W3C Official Site.* Edited by Tantek Çelik, Chris Lilley and L. David Baron. World Wide Web Consortium. July 21, 2008. http://www.w3.org/TR/css3-color/ (accessed November 11, 2008).

[2] Smith, Alvy Ray. "Color Gamut Transform Pairs." *SIGGRAPH 78 Conference Proceedings.* 1978. 12-19.

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

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

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

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

[7] Schanda, Janos. *Colorimetry: Understanding the CIE system.* Wiley Interscience, 2007.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Microsoft Corporation. "atan, atan2." *MSDN Library.* 2008. http://msdn.microsoft.com/en-us/library/aa272018(VS.60).aspx (accessed 11 24, 2008).

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

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

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

[16] Forrest, J. J. H., and J. A. Tomlin. "Branch and bound, integer, and non-integer programming." *Annals of Operations Research* (Springer Netherlands) 149, no. 1 (2007): 81-87.

[17] Borchers, Brian, and John E. Mitchell. "An improved branch and bound algorithm for mixed integer nonlinear programs." *Computers and Operations Research* (Elsevier Science Ltd.) 21, no. 4 (1994): 359-367.

[18] Finley, Darel Rex. "HSP Color Model — Alternative to HSV (HSB) and HSL." *The WWW Homepage of Darel Rex Finley.* 2006. http://alienryderflex.com/hsp.html (accessed November 25, 2008).

[19] Kotsarenko, Yuriy, and Fernando Ramos. "Simple perceptual color space for color specification and real-time processing." General Congress of the International Commission for Optics (ICO), Puebla, Mexico, August 2011. Proc. SPIE 8011, 80119D; doi: 10.1117/12.901997.

About the authors

Dr. Yuriy Kotsarenko has a Ph.D. and M. Sc. in Computer Sciences and Engineer Degree in Computer Systems. His research is focused in areas of computer vision, computer graphics, optics and software engineering; he also collaborates in several research works related to robotics with interests in seismology, agriculture and ecology. Currently he is working on a project for Embarcadero Technologies (Scotts Valley, California), as a frameworks arquitect.

Dr. Fernando Ramos Quintana is a senior researcher at ITESM, Campus Cuernavaca. He was the head of the Graduate School on Computer Science and Engineering at the campus. He holds a Masters and Ph.D, Degrees in Robotics from Universite de Franche Comte, Besancon, France, and a Bachelor degree in Electronic Engineering from the Mexican National Polytechnic Institute. He is also a member of National System of Researchers from CONACyT, Mexico. His research interests include the development of interaction models in multi-agent systems, computer assisted education and biology systems

.

Supplemental Tables

The following tables contain data from user study and performance benchmarking experiments and are included separately from the document due to size constraints.

Category	HSL	HCL	CIELUV	DIN99	YScH
Variety	8.34	5.17	7.15	5.54	8.38
Uniformity	7.19	4.25	6.89	6.17	8.70
Selection	8.18	5.57	7.24	5.75	8.53
Overall	7.90	4.99	7.09	5.82	8.53

Table 1. The average scores given by the participants in the quiz in the scale of 1 to 10, with 1 being the worst and 10 being the best.

Abbrev.	Machine Specification
Eee10HE	Asus Eee PC 1000HE, Intel Atom N280 1.66 Ghz, DDR2-667
Eee15PN	Asus Eee PC 1015PN, Intel Atom N570 1.66 Ghz, DDR3-1333
LatD830	Dell Latitude D830, Intel Core 2 Duo T7700 2.4 Ghz, DDR2-667
Prec4500	Dell Precision M4500, Intel Core i7 Q840 1.86 Ghz, DDR3-1333
AMD3310	Samsung NP305V4A-A01MX, AMD Dual-core A4-3310MX APU, DDR3-1333
Quad24	Desktop Core 2 Quad Q6600 2.4 Ghz, DDR2-800 (performance RAM)
Cel1400	Desktop Celeron Dual-Core E1400 2.0 Ghz, DDR3-800
Cel3300	Desktop Celeron Dual-Core E3300 2.5 Ghz, DDR2-800
iPad2	Apple iPad 2, CPU Apple A5 Dual-core 1 Ghz, DDR2

Table 2. The description of different systems used in benchmarking experiments.

	HSL	HCL	HCL^2	LUV	DIN99	YCH	YScH
Machine	**64-bit Mode (time in *ms*), 0.3 MP**						
LatD830	23.35	75.05	2171.79	3019.55	4557.05	47.31	572.55
AMD3310	27.36	81.87	2416.23	3746.55	5885.67	52.86	757.51
Prec4500	7.85	27.28	747.50	1053.17	1720.04	17.29	259.31
Cel1400	25.88	92.17	2643.42	3621.8	5558.39	56.27	697.31
Quad24	12.84	38.89	1090.59	1496.03	2286.2	25.03	289.21

Table 3. Benchmarking results while running in 64-bit CPU mode using multi-threading approach while processing 0.3 MP (resolution 640x480) image. The time is given in milliseconds and represents how long it took to process a single image; HCL^2 refers to HCL color space with valid chroma algorithm applied.

	HSL	HCL	HCL²	LUV	DIN99	YCH	YScH
Machine	**32-bit Mode (time in *ms*), 0.3 MP**						
Eee10HE	155.34	360.79	11310.79	12622.98	21891.04	138.75	2810.77
Eee15PN	78.76	184.41	5991.28	6398.83	10886.28	68.08	1385.32
LatD830	32.63	102.35	3090.48	3274.03	5882.4	35.32	634.21
AMD3310	42.22	119.40	3932.05	3951.06	6993.24	54.33	912.3
Prec4500	12.86	34.83	1048.01	1158.91	2046.56	13.62	256.85
Cel1400	39.22	122.80	3747.53	3978.59	7144.34	42.84	768.5
Quad24	17.34	51.84	1546.84	1645.34	2950.31	18.5	316.41
Cel3300	26.26	84.30	2588.54	2834.01	5261.72	30.65	588.61
iPad2	139.13	353.00	11213.38	13927.12	21536.13	215.63	3714.38

Table 4. Benchmarking results while running in 32-bit CPU mode using multi-threading approach while processing 0.3 MP (resolution 640x480) image. The time is given in milliseconds and represents how long it took to process a single image; HCL^2 refers to HCL color space with valid chroma algorithm applied.