2007-09-01

Image-Based Color Schemes

Bryan S. Morse
morse@byu.edu

Daniel Thornton

See next page for additional authors

Original Publication Citation

B. Morse, D. Thornton, Q. Xia, and J. Uibel, "Image-based color schemes," in IEEE International Conference on Image Processing (ICIP), vol. III, pp. 497-5, September 27.

Authors
Bryan S. Morse, Daniel Thornton, Qing Xia, and John Uibel

IMAGE-BASED COLOR SCHEMES

Bryan S. Morse[1], *Daniel Thornton*[1], *Qing Xia*[1], *and John Uibel*[2]

Brigham Young University
[1]Department of Computer Science
[2]Center for Instructional Design
Provo, UT 84602

ABSTRACT

This paper presents a novel method for generating color schemes based on images intended to anchor color designs. This has wide applicability for web pages, printed materials, or other applications where images are used as a key part of the overall design. Unlike methods that are variants of color quantization and try to pixel-wise approximate the image, this method draws on graphic-design principles by emphasizing hue selection first, weighting effects of color by saturation, and considering the local spatial coherency in order to determine the overall visual impact of a color. Results demonstrate that the method generalizes to a wide range of images, choosing visually important elements of the coloration of the source image.

***Index Terms*—** Image processing, image color analysis

1. INTRODUCTION

Designs for online or printed materials often center around images that anchor the overall graphic design. These include pages specifically for displaying photographs but also those for corporate sites, weblogs (blogs), online instruction, or other thematically consistent materials. In some cases, the services of skilled graphic designers may be available to design the color schemes and layout. In others, designers may create basic templates that can be adapted to the colors of individual images chosen for each site, group, or page. At our university for example, graphic designers create basic course templates intended to be anchored by a single graphic or image, but these may then be customized for each course based on specific images.

This paper presents a method for generating color schemes from images intended to anchor designs, such as in Figure 1. This not only speeds up the time to design such pages but puts adaptation of basic site or page designs into the hands of content specialists who may not have skills in color design.

2. BACKGROUND AND RELATED WORK

The idea of using images to generate color schemes is not in itself new. The simplest approaches follow those found in color quantization, which itself has a long history [1, 2, for example]. The connection between quantization and color scheme generation would seem to be obvious, but there are important differences between a quantization palette and a color scheme. The goal of quantization is to minimize the overall error when replacing each pixel by the nearest member of the quantized palette, or in some cases trying to minimize the error in the appearance of a dithered result [3]. As a result, quantization often biases the resulting palette towards more commonly occurring colors in a utilitarian fashion rather than considering the overall visual effect of those colors. Color schemes, though, are concerned with creating *other* content that harmonizes with the image, perhaps picking up on small, coherent, saturated parts of it.

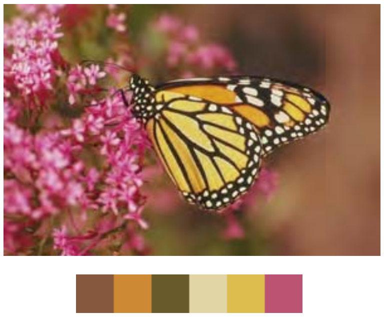


Fig. 1. An automatically generated color scheme

The method presented in [4] uses a segmentation approach to the problem, first segmenting the image then identifying the colors of the segmented regions. This is a powerful approach, but it relies on accurate segmentation and does not always work as well in cases of highly textured regions. Most other approaches, as with the one proposed here, use histograms to look at global image color properties rather than requiring accurate color segmentation.

There are a limited number of commercial or web-based tools for manually generating (and in some cases, sharing) color schemes [5, 6, 7], some of which are also capable of generating color schemes from images [5, 7, 8]. Although the methods behind these tools are not always publicly available, they either use or appear to use quantization [8] or randomized sampling [7] to generate the schemes.

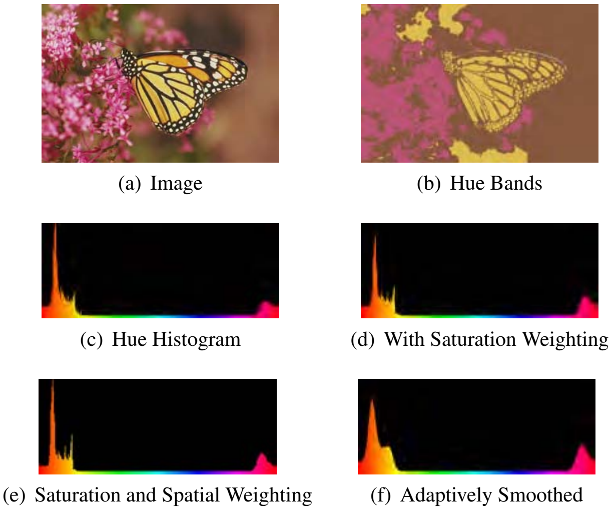
An obvious connection with the aims of this work can be made with the Color Harmonization work of Cohen-Or, et al. [9], which can recolor images to better harmonize their colors. It can also harmonize text, banners, or other content added to the image as part of the overall harmonization, but this isn't the same as generating schemes for generating large amounts of additional content.

The method presented in this paper uses a multistage subdivision approach, emphasizing first the color hues, then the saturations, then the lightness values, thus closely modeling the approach and priorities a graphic designer might use.

3. METHODS

Consultation with professional graphic designers revealed that for them it was most important to "get the hues first". Getting the right hues from the image is far more important from a perceptual or design standpoint than trying to match saturations or intensities.

1-4244-1437-7/07/$20.00 ©2007 IEEE



(a) Image (b) Hue Bands

(c) Hue Histogram (d) With Saturation Weighting

(e) Saturation and Spatial Weighting (f) Adaptively Smoothed

Fig. 2. Hue selection process. After adaptive smoothing of a saturation-weighted, spatial-coherence-weighted histogram of the hues (c–f), three distinct bands of non-gray hues are identified (b,f).

3.1. Identifying Dominant Hues

After converting the image to an HLS color space, with $H(x,y)$, $L(x,y)$, and $S(x,y)$ denoting respectively the hue, lightness (intensity or brightness), and saturation of pixel (x,y), we begin with a histogram of the hues in the image (Figure 2c). If we denote the set of pixels with hue h (in degrees) as the set P_h:

$$P_h = \{(x,y) : H(x,y) = h\} \tag{1}$$

the hue histogram $D(h)$ can be written simply as the cardinality of each respective set P_h:

$$D(h) = |P_h| \tag{2}$$

We have found that simply using the hue histogram alone isn't sufficient, as it treats equally pixels with different saturations. A strong red is given the same weight as a barely (and perhaps imperceptibly) pinkish gray. We thus further refine the hue histogram by weighting each pixel by its saturation:

$$D(h) = \sum_{(x,y)\in P_h} S(x,y) \tag{3}$$

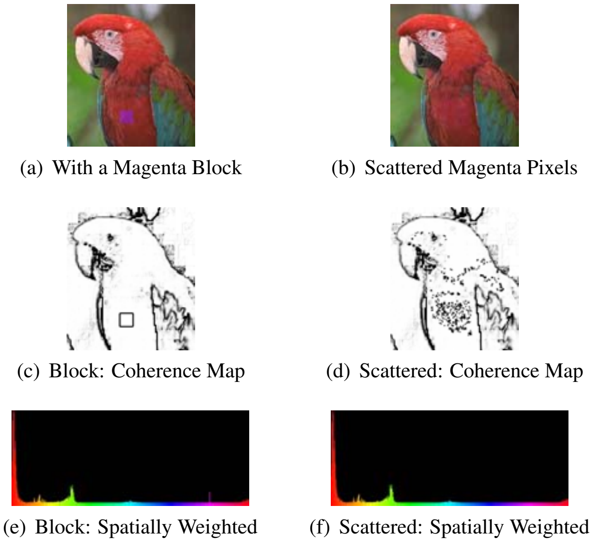
The effects of saturation weighting can be seen in Figure 2d. Notice the lower relative weighting in the histogram of the large area of low-saturation orange to the right side of the source image.

Another point that came out in discussion with graphic designers is the important of coherent blocks of color. We calculate the local spatial color coherence $C(x,y)$ at each pixel by first calculating the sum of the absolute differences[1] between the pixel's hue and those of its eight-connected neighbors:

$$N(x,y) = \frac{\sum_{i=1}^{-1} \sum_{j=1}^{-1} |H(x,y) - H(x+i,y+j)|}{8} \tag{4}$$

and modeling the coherence as the likelihood of this local neighborhood difference relative to the estimated noise in the image:

$$C(x,y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-N(x,y)^2/2\sigma^2} \tag{5}$$



(a) With a Magenta Block (b) Scattered Magenta Pixels

(c) Block: Coherence Map (d) Scattered: Coherence Map

(e) Block: Spatially Weighted (f) Scattered: Spatially Weighted

Fig. 3. Effects of spatial coherence weighting. The solid block of magenta added to (a) has strong internal coherence (c) and appears clearly as a spike at that color in the corresponding spatially weighted hue histogram (e), while the same number of randomly scattered pixels of the same color (b) have little coherence (d) and thus little effect (f).

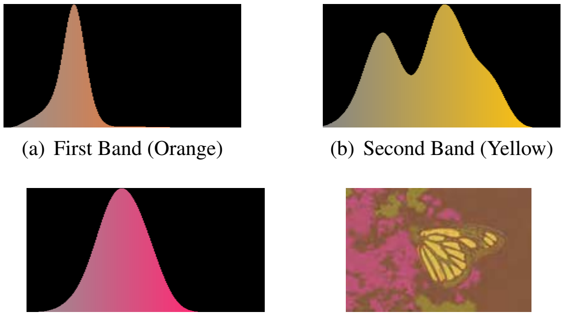
We then use this spatial coherence to further weight (Figure 2e) the saturation-weighted hue histogram:

$$D(h) = \sum_{(x,y)\in P_h} S(x,y)\; C(x,y) \tag{6}$$

The effects of weighting by spatial coherence can be seen in Figure 3. To evaluate and demonstrate the effect of weighting by spatial coherence, we took the well-known "parrots" image and added a 20 × 20 block of magenta pixels (Figure 3a). We also took the original image and added the same number (400) of randomly scattered magenta pixels (Figure 3b). As can be seen visually, the magenta block in the first image is quite noticeable and adds to the overall perception of colors in the image, while the scattered pixels are almost imperceptible. Other than the pixels in the original image that were overwritten by the magenta pixels, one would thus expect the saturation-weighted histograms for these two images to be the same. Any method using histograms alone without spatial coherence weighting would thus produce the same results for these two images. By including spatial coherence weighting (Figure 3c,d), the block of magenta has significant effect (Figure 3e), while the effects of the randomly scattered pixels of the same color are negligible.

The spatial-coherence-weighted, saturation-weighted hue histogram is then smoothed, again modulo 360 degrees, to remove secondary peaks (Figure 2f). The smoothing window is adjusted until a user-specified maximum number of peaks (local maxima) and minimum distance between peaks are reached. Local minima between the peaks are identified, and the range of hues between these local minima constitute *hue bands* B_k for each significant hue in the image. In the example in Figure 2, three distinct dominant hues are identified: orange, yellow, and (reddish) magenta. A spatial map of the pixels in these bands can be seen in Figure 2b.

[1] All addition and subtraction of hues is performed modulo 360 degrees due to the wrap-around nature of the color wheel.



(a) First Band (Orange) (b) Second Band (Yellow)

(c) Third Band (Magenta) (d) Hue-Saturation Bands

Fig. 4. Further subdivision of each band of hues by saturation. Smoothed histograms of the saturation within each previously identified band of hues (a–c) is used to further subdivide by saturation. In this case, further subdivision of the orange and yellow bands results in a total of five hue-saturation bands (d).

3.2. Identifying Saturations and Intensities within Hues

Once the dominant hues are identified, the image is partitioned into regions according to the hue bands, and the pixels for each band are further subdivided by saturation. This is done by constructing a histogram of saturations for the pixels in each hue band. If we denote the set of pixels in hue band k with saturation s as

$$P_{k,s} = \{(x,y) : H(x,y) \in B_k \text{ and } S(x,y) = s\} \qquad (7)$$

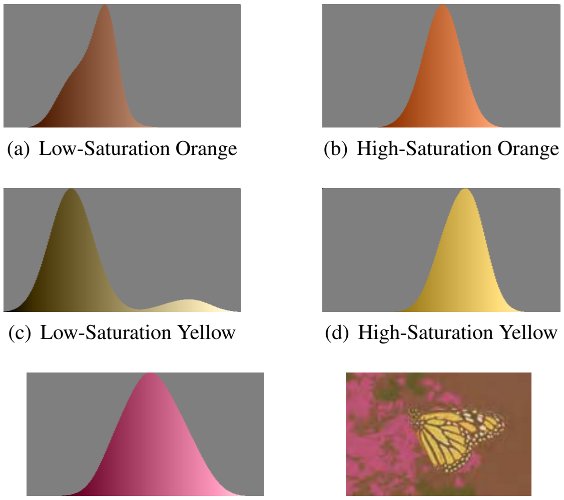
the weighted saturation histogram $D_k(s)$ for each band can be written in a manner similar to Eq. 6:

$$D_k(s) = \sum_{(x,y)\in P_{ks}} S(x,y)\, C(x,y) \qquad (8)$$

These saturation histograms for each hue band are then adaptively smoothed with the maximum number of peaks and minimum distance between peaks adjusted based on the percentage of the image in that hue band, and the peaks and valleys defining each band are again identified. The smoothed saturation histograms for the hue bands from Figure 2 are shown in Figure 4. A single dominant saturations is identified for the magenta band. Two dominant saturations are identified in the orange band, dividing these into a lower-saturation orange band and a (small but distinct peak) higher-saturation orange band. Two dominant saturations are also identified in the yellow band, dividing these into a lower-saturation yellow band and a medium-saturation yellow band. A spatial map of these bands can be seen in Figure 4d.

A similar process is repeated for the intensities in each hue-saturation band, the results of which for the example in Figures 2 and 4 are shown in Figure 5. In this case, a single dominant lightness is found for each of the low-saturation orange, high-saturation orange, high-saturation yellow, and magenta bands. Two dominant intensities are identified for the low-saturation yellow band, one darker (greenish brown) and one brighter (the off-white spots on the wings). A spatial map of these final bands can be seen in Figure 5e.

Once the image has been subdivided by hue, then saturation, then lightness, the color values for the pixels within each hue-saturation-lightness band are then averaged to determine the palette for the color scheme.



(a) Low-Saturation Orange (b) High-Saturation Orange

(c) Low-Saturation Yellow (d) High-Saturation Yellow

(e) Magenta (f) Final Color Bands

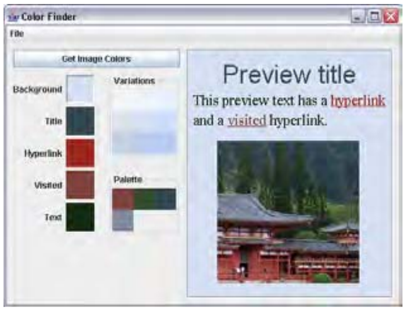
Fig. 5. Final subdivision of each hue-saturation band by lightness. Smoothed histograms of the intensities within each previously identified hue-saturation band (a–d) are used to further subdivide by lightness. In this case, further subdivision of the low-saturation yellows results in a total of six hue-saturation bands (e). Compare the original image and the resulting color scheme found in Figure 1.

3.3. Variations

In addition to this core method, we have also implemented several optional variations that we have found useful. The first of these is to exclude extremely bright and/or extremely dark colors from consideration. On many monitors, for example, extremely dark colors may appear uniformly black to the user. If one assumes that pure blacks and whites will be available to the user as part of their color schemes, one can avoid including consideration of near-blacks and/or near-whites when determining the dominant hues and the corresponding saturation/lightness subdivisions.

Another variation is to consider complementary or triad colors when determining the dominant hues. Each pixel's hue may be included in the histogram not only for its own hue but (to a lesser degree) for complementary or triad colors. This has the effect of merging the combined contributions to the histogram of complementary or triad combinations. For example, a strong blue may contribute to the values for corresponding green or yellow triad colors. If there are a large number of near-green pixels close to the green triad color for the blue, their effects combine when the histogram is smoothed. Thus, the near-green peak may be pulled slightly from the actual hue in the image closer to the green triad color suggested by the blues. Rather than simply picking complementary colors or triads then, the result is driven towards colors that appear in the image *and* are color-scheme compatible with other colors in the image.

One may also consider the uses of the selected colors when designing the scheme for a particular intended application. For example, if the intent is to create an image-driven color scheme for a web page, one can draw from web design principles suggesting that darker text on lighter backgrounds is preferable. Our method can automatically select the lightest color, lighten and desaturate it further, and use that for the background. Similarly, the darkest color can be further darkened to use as a suitable text color. One could also select the hue closest to the text color, darken that suitably, and use



Preview title

This preview text has a hyperlink and a visited hyperlink.

Fig. 6. Interactive interface for selecting colors from the scheme

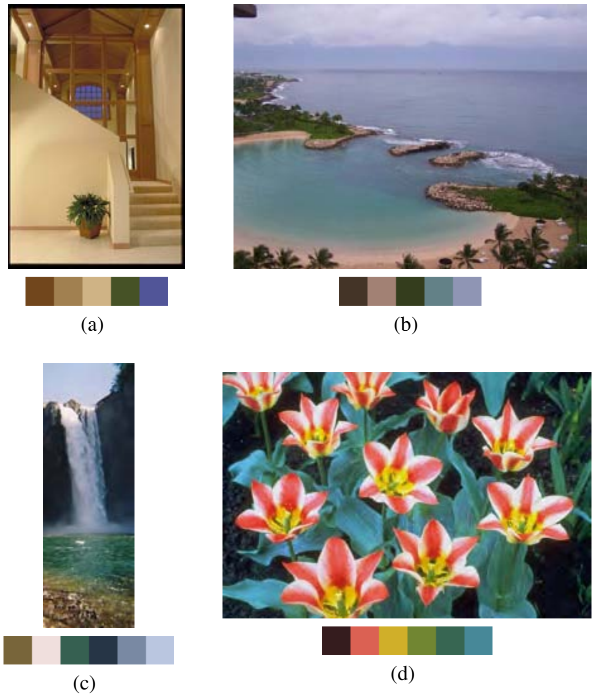


Fig. 7. Additional examples of automatically generated color schemes based on their respective images

it for the “visited links” color. Such application-specific designs are enhanced by primarily identifying dominant hues then selecting variations of the saturation or brightness values for specific uses. In our implementation, we also present the user with saturation/lightness variations on each dominant hue (Figure 6).

4. RESULTS

Additional example of the results of automatically generating color schemes using the method presented here can be found in Figure 7.

In Figure 7a, it is important to note that the color scheme picks up on the green of the plant and the blue of the sky showing through the window, even though these areas are relatively small compared to the overall tan and brown of the room. This is different from typical quantization-based methods, which would inordinately weight the more common colors since these methods are designed to minimize the overall error in replacing pixels by entries from the palette, not in identifying the overall most visually prominent colors.

Many of the other figures show similar results, often picking up on small though coherent and visually significant areas of color rather than simply picking multiple variations of the colors from larger, more spatially dominant regions.

5. CONCLUSION

This paper has presented a novel method for generating color schemes from images. Although the idea of generating color schemes from images is not new, the methods proposed here produce better results than those produced by quantization approaches or by random sampling of the image. By focusing first on identifying the significant hues, then the saturation/lightness combinations within bands of those hues, the algorithm approaches the problem in a fashion similar to that employed by graphic designers. This also means that as the user adapts the saturations and intensities of the colors in the scheme to fit the intended application, the hue-driven scheme is still compatible with the image.

One obvious limitation of the approach presented here is that it assumes that the colors in the image are themselves aesthetically harmonious (why else would the user have picked it to anchor the design?), and the basic method does not try to “enforce” harmony in the generated scheme. In other words, the generated scheme may be no more harmonious that the original image. However, [9] notes that most images of nature are often already harmonious.

Encouraging harmonious schemes may be done by accentuating complementary or triad colors in the selection process as presented here, though these employ only simple notions of harmony. One might also employ more sophisticated models of color harmony such as those proposed in [10] and employed in [9].

A web-based version for selecting HTML color schemes is available at `http://rivit.cs.byu.edu/ColorSchemes`.

6. REFERENCES

[1] Paul Heckbert, “Color image quantization for frame buffer display,” in *Proceedings of SIGGRAPH ’82*, 1982, pp. 297–307.

[2] Xiaolin Wu, “Color quantization by dynamic programming and principal analysis,” *ACM Transactions on Graphics*, vol. 11, no. 4, pp. 348–372, 1992.

[3] Bryan Morse, Thomas Howard, Stephen Larson, Michael Bastian, and Eric Mortensen, “Color quantization and dithering gamuts,” in *Proceedings of the International Conference on Signal and Image Processing (SIP’99)*, 1999.

[4] Pere Obrador, “Automatic color scheme picker for document templates based on image analysis and dual problem,” Tech. Rep. HPL-2006-10, Hewlett Packard Laboratories, 2006.

[5] “Color schemer,” http://www.colorschemer.com.

[6] Adobe Labs, “Kuler,” http://kuler.adobe.com/.

[7] Lloyd Dalton, “Colr,” http://www.colr.org/, 2005.

[8] Steven DeGraeve, “Color palette generator,” http://www.degraeve.com/color-palette/.

[9] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu, “Color harmonization,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 624–630, July 2006.

[10] Y. Matsuda, *Color design*, Asakura Shoten, 1995.