Paint Inspired Color Compositing

Nathan Gossett*
University of Minnesota at Twin Cities

Baoquan Chen†
University of Minnesota at Twin Cities

Figure 1: **Photographs of paint mixing.** *Left:* A photograph of Red, Yellow and Blue paint mixing together. (The white spots are produced by light sources reflecting off wet paint.) *Right:* A photograph of overlapping Red, Yellow and Blue watercolor paint strokes.

Abstract

Color is often used to convey information, and color compositing is often required while visualizing multi-attribute information. This paper proposes an alternative method for color compositing. In order to present understandable color blending to the general public, several techniques are proposed. First, a paint-inspired RYB color space is used. In addition, noise patterns are employed to produce subregions of pure color within an overlapped region, and color region edges are emphasized. We show examples to demonstrate the effectiveness of our technique for visualization.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.3.m [Computer Graphics]: Miscellaneous;

Keywords: RYB, Color Mixing, Perception.

1 Introduction and Motivation

Color compositing is a commonly used operation in computer graphics. In the specific area of visualization, color is often used to convey information. In the event that spatially displayed data has overlapping regions of differing attributes, color compositing can be used to communicate multiple channels of information to the viewer. However, this only works if the viewer is able to discern the individual components that combine to form the final color image.

Due to the specifics of the human perceptual system and the subsequent design of computer displays and printers, most computer graphics applications make use of the RGB (for additive displays) or CMYK (for subtractive media) color spaces. However, although these spaces are easy to use and are correct predictors of color mixing for computer monitors and color printers, they do not line up with some of the commonly held assumptions of the general public about color mixing. Due to the use of subtractive color models based on pigment mixing in early childhood art training, many people have a mental model of color that is quite different than the RGB model. When colors are mixed in RGB, the resulting colors are often different than the color expected by the viewer. This confusion can even extend to the subtractive color mixing strategy of CMYK due to the close relationship between the rules for CMYK and RGB mixing (CMYK is essentially the complement of RGB). These inconsistencies can lead to confusion in the viewer of a visualization.

A potential solution to this problem is color weaving, proposed by Urness et al. [7]. In their technique, multiple overlapping color regions within a LIC image can be visualized by restricting each streamline to a single color. Thus, no actual compositing of colors occurs, but neighboring streamlines all contribute to an overall understanding of the coloration of a larger region. While this technique is quite effective for flow visualization, it does not trivially extend to other applications.

Our objective, then, is to develop an alternative color compositing strategy that takes advantage of assumptions that regular, untrained viewers are likely to make. Towards this end, we make two observations. First, unlike with RGB mixing, many people do not experience much confusion when presented with a painted area that is a mixture of multiple colors. Many people have had at least some experience with paint mixing (Figure 1 Left). In addition, some art formats such as watercolor can produce uneven color coverage due to variations in paper absorbance and brush technique. In these cases, regions with overlapping colors will still contain small subregions that are heavily biased towards one color, making identification easier (Figure 1 Right). We take these two experiences of painting as inspiration for an intuitive color compositing strategy. Note that we do not attempt to exactly duplicate the look of painted brush strokes, but instead attempt to extract desirable properties from these examples.

This technical report is an extension of our previous work [2]. Here, we propose the use of three techniques to aid in the proper

*email: gossett@cs.umn.edu
†email: baoquan@cs.umn.edu

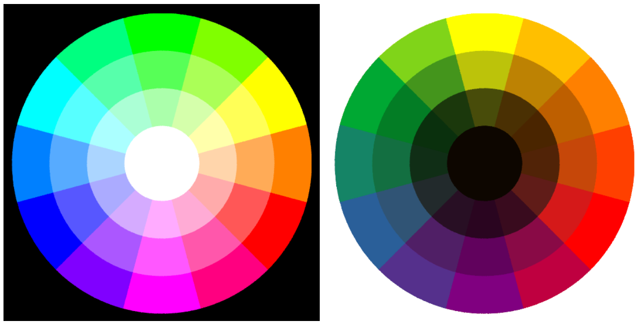


Figure 2: **Color wheels.** *Left:* RGB color wheel. 12 o'clock is pure Green, 4 o'clock is pure Red, and 8 o'clock is pure Blue. 2, 6 and 10 o'clock are two primaries at 100%. The rest of the positions are produced with one primary at 100% and another at 50%. Colors are saturated at the outside of the circle and desaturated at the center. *Right:* RYB with non-linear interpolation. This wheel is produced by using the RGB values from the left image as RYB values. The colors produced are approximately the colors suggested by Itten with the exception of Blue-Green (9 o'clock) which is darker in our model.

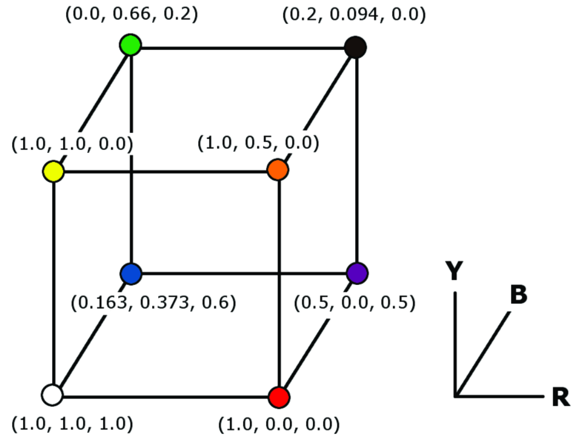


Figure 3: **RYB interpolation cube.** RGB coordinates are given for each corner of the RYB cube. RGB colors are chosen based on Itten's suggestions.

understanding of overlapping color regions.

1. Using a subtractive color space with Red, Yellow and Blue as primary colors
2. Using procedurally generated noise patterns to create subregions of easily identifiable colors within a mixed region
3. Emphasizing the boundaries of differing regions

Our aim is to improve visualization where color is used to convey multiple data properties.

2 Achieving Intuitive Color Mixing

2.1 RGB vs RYB

The RGB color space is a familiar and popular color space, particularly within the computer graphics community. By defining colors using Red, Green and Blue as primary colors, any color in a computer monitor's color gamut can be produced, which usually encompasses a majority of the colors the human visual system is capable of seeing. Color mixture in RGB is handled additively, so that Red and Green mix to form Yellow, Red and Blue mix to form Magenta, and Green and Blue mix to form Cyan (see Figure 2 Left). However, this results in color mixtures that might be confusing to a naive viewer. Yellow is not often thought of as a mixture of Red and Green, and few people would predict White as the result of mixing Yellow and Blue.

Instead, many people carry a mental model of color that is more in line with that endorsed by Johannes Itten [3]. Itten's model is still widely used in art education. In this model, Red, Yellow and Blue are used as "pure" primary colors. Red and Yellow mix to form Orange, Yellow and Blue mix to form Green, and Blue and Red mix to form Purple (see Figure 2 Right). These are the colors an untrained viewer would expect to obtain using children's paint (see Figure 1 left). In addition, many people do not think of White as the mixture of all colors, but instead as the absence of color (a blank canvas). A more common assumption would be that mixing many colors together would result in a muddy dark brown color.

To create color mixtures that more closely resemble the expectations of those not trained in the RGB color space, we propose the use of an idealized RYB system.

Realization Details:
Unlike RGB, the RYB color space is not based on any psychological perception model, but is instead based on experiences of idealized paint mixing. As such, it would be difficult to try to define a rigorous mathematical conversion from RYB to RGB for the purpose of display. However, a reasonable approximation may be obtained by defining a cube with each axis representing either Red, Yellow or Blue (see Figure 3). By defining appropriate RGB values for each of the eight colors represented by the corners of the cube, we can use trilinear interpolation to obtain suitable RGB values for any color defined in RYB (see Figure 2). For instance, if a color were to be defined as 100% Red, 50% Yellow and 25% Blue, it could be represented as (1.0, 0.5, 0.25) in RYB coordinates. By interpolating the RGB values defined at the 8 corners of the cube, this color would have RGB coordinates of (0.8375, 0.19925, 0.0625), producing a slightly muddy orange color as expected.

Taking into consideration the natural limits people have in dealing with multiple categories of information [4], and color categories specifically [1], we can limit the use of ambiguous colors in our implementation. A non-linear interpolation (a cosine function or something similar) may be used to bias the interpolated colors towards the 8 characteristic colors located at the corners of the cube. For instance, a linear interpolation by a factor of t between values A and B is $A+t(B-A)$. However, using $A+t^2(3-2t)(B-A)$ will bias the result towards A or B. In the above example, interpolating in this manner will produce RGB coordinates of (0.886719, 0.212891, 0.039063), which is a brighter, more identifiable orange. Biasing in this manner will ensure that the viewer will encounter a more limited number of easily identifiable colors, but still allows for smooth transitions between colors.

Example code for performing this conversion can be found in Appendix A.

2.2 Noise Patterns

Colors are more easily identifiable when separate colors are allowed to exist in separate spatial regions. A process such as Color Weaving [7] can be used to create regions of "pure" color that are large enough to be picked up by the viewer and correctly identified.

Similar effects can be achieved by using noise patterns to modulate the intensity of each color over a region. Regions with multiple colors overlapping can then have sub-regions where one color contributes more than the others, making that color easily identifiable. In sub-regions where multiple colors contribute equally, the RYB color blending described above provides intuition for component identification.

Realization Details:
Our goal when using noise patterns was to create a random variation in color intensity similar to the variations encountered in watercolor paintings. We found that the turbulence function proposed by Ken Perlin [6] produced adequate results. A 2-dimensional turbulence texture is created for each source color. This texture is then used to modulate each color at render time. Areas with no color overlap are modulated less than areas with color overlap. (See Figure 4 (c).)

It should be noted that if subregions of pure color are too small, the viewer will perceive additive mixture of the colors [5]. Effort must be made to ensure that subregions of pure color are large enough that additive mixing does not occur.

On the other hand, one potential problem with using low or moderate frequency turbulence functions is that regions of pure color can be relatively large and coherent, giving the false impression that the subregion is not part of an overlapping region. To avoid this, a sufficiently high frequency pattern can be used. Figure 5 shows a comparison of these two options in the middle and bottom images.

2.3 Edge Emphasis

Using a simple edge detector, the boundary of each color region can be identified. By saturating the color along this edge, identification of each color region is made easier. If clean edges are too distracting, the intensity can be blended towards the interior of the color region in a randomized manner. Figure 4 (c) demonstrates this effect.

Realization Details:
We render each source color separately and produce a gradient estimation using a Sobel edge detection filter. A map is then produced by starting at each non-zero gradient position and proceeding in the gradient direction a random distance. Points located further away from the detected edge are given smaller weights in the edge map. This map is then used to intensify each color in the final composite image.

3 Results and Discussion

We first demonstrate our color mixing technique with a simple example in Figure 4 a-c. When using traditional RGB compositing, certain combinations, such as mixing green and red to obtain yellow, may be unintuitive to naive viewers. Simply switching to RYB compositing creates mixtures that are much more intuitive. With the addition of noise patterns and edge emphasis, viewers can clearly identify the source colors that are mixed in any particular region. For instance, in the region where Blue and Yellow overlap, some subregions are clearly Blue, some are clearly Yellow, and some are Green (an intuitive mixing of Blue and Yellow). Even in areas where more than two colors overlap, the individual source colors can be identified.

Figure 5 provides an example of how our color compositing strategy can provide a more intuitive effect than standard RGB mixing. Using a map of the USA, we attempt to display the presence or absence of three traits (linguistic patterns for example) for each state. Note that for the RGB map, the state of Arizona is White. Rather than viewing this as having the meaning that Arizona contains all three traits, many viewers would view this as meaning Arizona carries none of the traits (i.e. Arizona is "blank"). In the two examples of our strategy, the viewer is able to see regions of Red, Blue and Yellow, along with regions of Brown.

A difficult situation arises when the color compositing is used to colorize another visualization, such as the LIC image seen in the bottom right of Figure 4. Here, the spatial frequency of the LIC image itself interferes with the spatial frequency of the noise pattern, making it more difficult to identify the "pure" subregions. Similarly, if a color region is too small (such as Long Island, New York in Figure 5), there may not be enough spatial resolution to produce a noise pattern. In these types of situations, the RYB color blending and edge emphasis are able to compensate for the lessened effect of the noise image.

The use of independent techniques allows for easy modification of the various algorithms to achieve different effects. For instance, although the maps in Figure 5 portray the presence of binary traits, strategies could be employed to convey relative percentages of traits. Using noise patters of different frequencies could convey additional information.

4 Conclusion

Although we are not advocating the use of our system as a universal replacement for the RGB system, we do present it as an alternative or supplemental system. Due to a significant portion of the population being mostly untrained in RGB additive color mixing, visualizations targeting these people must find some alternative mixing strategy. The RYB color mixing strategy has seen wide exposure, especially in the art community. By taking advantage of any exposure to RYB that the viewer may have had (particularly as a child), an intuitive color mixing strategy will improve visualization by communicating better to naive viewers.

In addition, the use of noise patterns allows subregions of "pure" colors to exist within overlapping regions. These subregions can allow a viewer to easily and accurately identify how many and which colors are overlapping within that region. By emphasizing edges, the boundary of each color region can be clearly identified by the viewer.

By using multiple complementary techniques, our color compositing strategy is better able to handle difficult compositing situations. Unlike other color compositing techniques [7], they require no knowledge of the problem domain and can be applied quickly and independently. All techniques described in this paper may be easily implemented using a combination of widely known algorithms. Future work on this topic will focus on testing the effectiveness of our system in comparison to other color mixing and compositing techniques for use in visualization.

5 Acknowledgements

We would like to thank the anonymous reviewers of the InfoVis version of this work for their helpful suggestions and improvements. We also wish to thank the University of Minnesota Graphics group for their discussions and criticisms that helped shape the direction of this project. We also thank Timothy Urness for providing the flow dataset used in Figure 4 d.

This work is supported by NSF CAREER ACI-0238486 and in part by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAD19-01-2-0014. Its content does not necessarily reflect the position or the policy of this agency, and no official endorsement should be inferred. Research space and administrative support provided by the University of Minnesota Digital Technology Center.

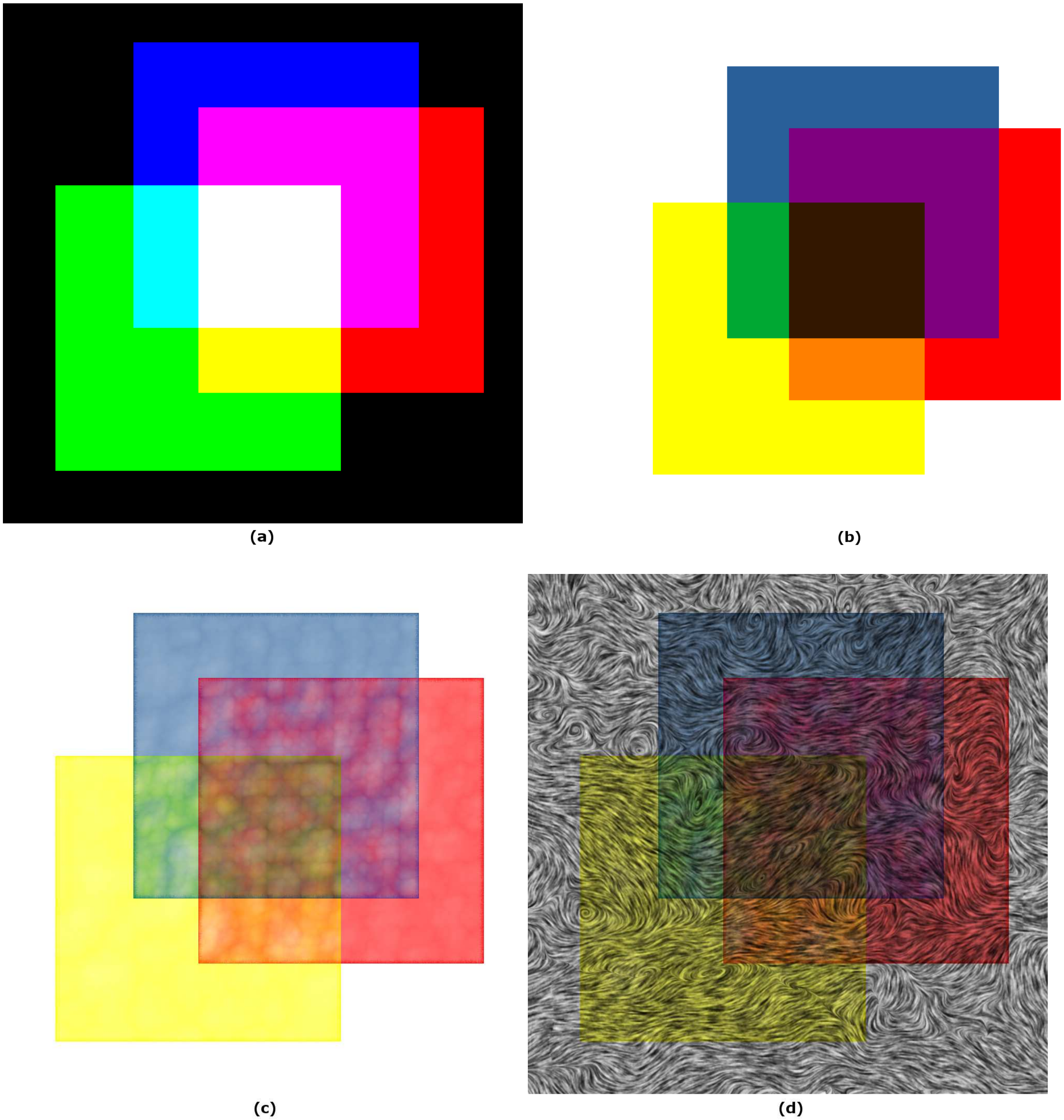
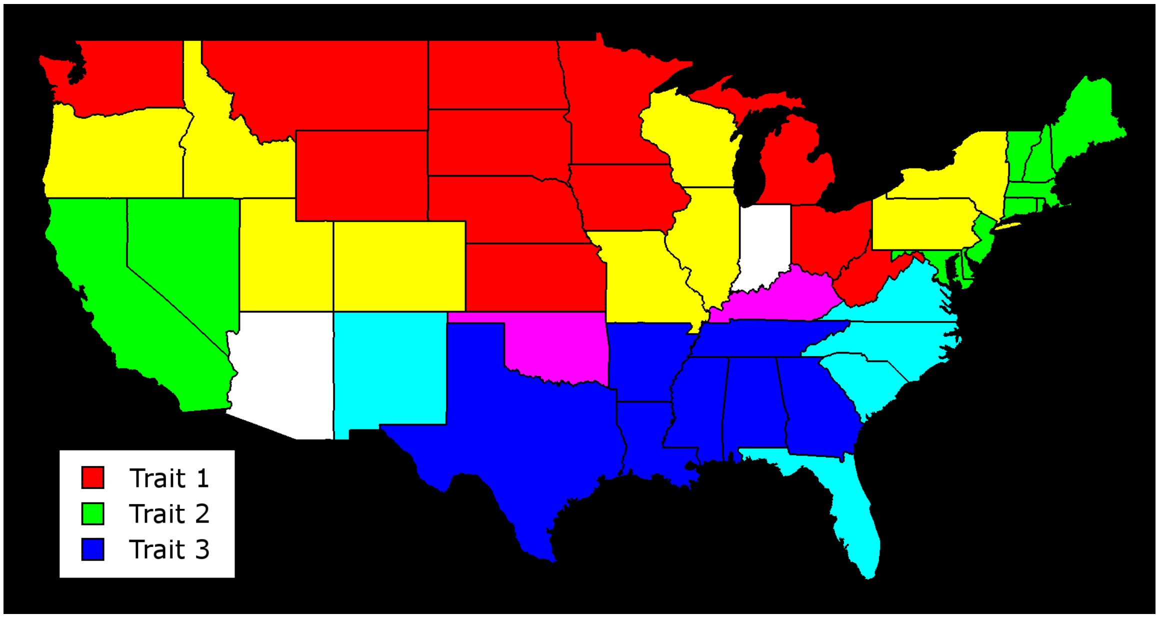
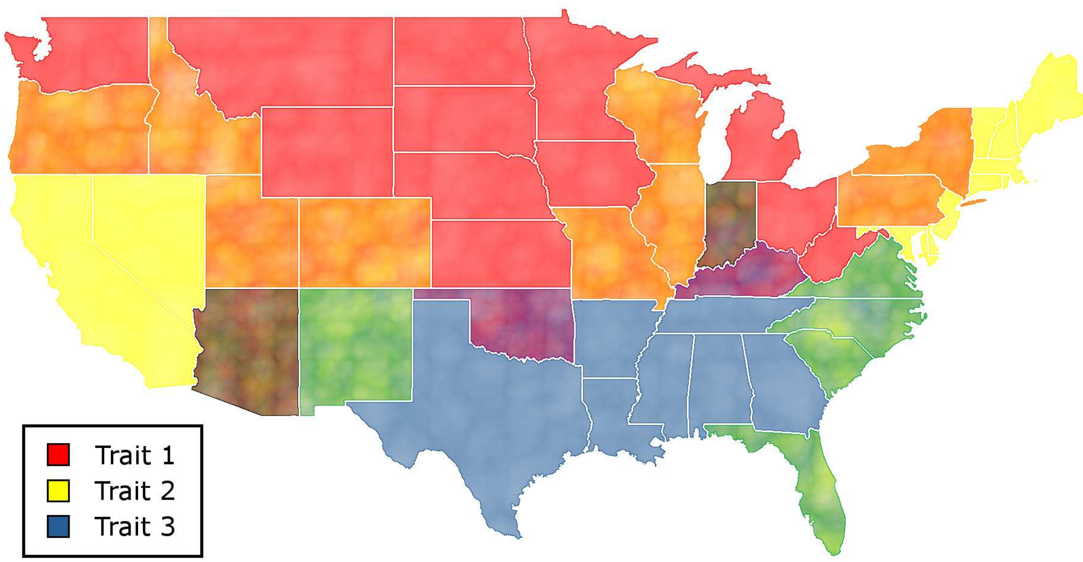
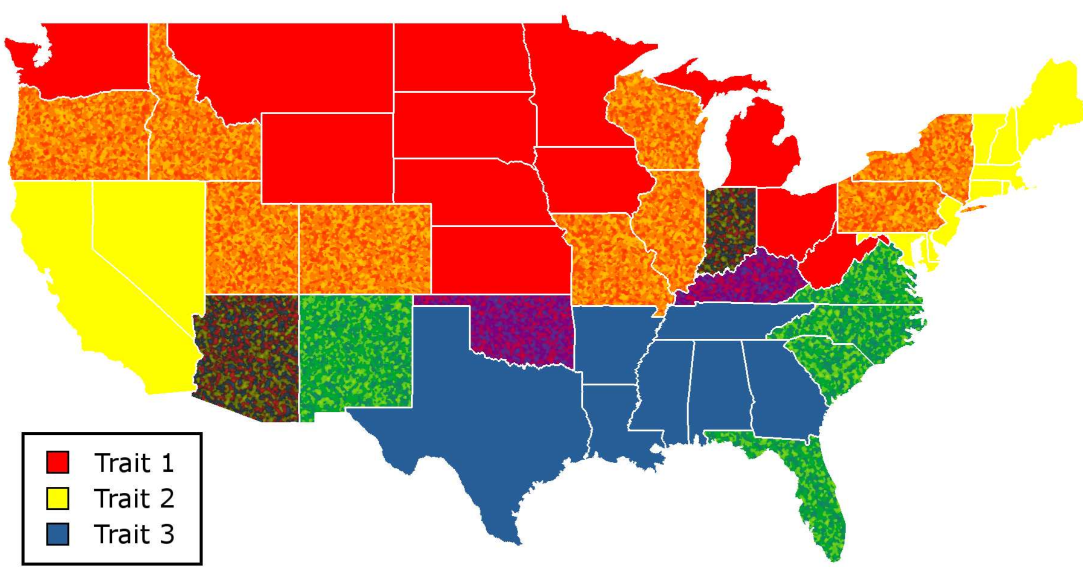


Figure 4: **Intuitive compositing.** *(a)* RGB color compositing, *(b)* RYB color compositing, *(c)* RYB compositing with Perlin noise and edge emphasis, *(d)* LIC image overlayed (Flow dataset provided by Timothy Urness)

Figure 5: *(Next Page)* **Using color compositing in maps.**
Top: Using RGB compositing to show three populations with Red, Green and Blue representing the three groups. Note that Cyan, Yellow, Magenta and White represent overlapping regions, yet they appear to be additional independent populations.
Middle: Using RYB and Perlin noise to show the same three populations with Red, Yellow and Blue representing the three groups. Note that states with multiple populations overlapping appear to be mixtures of colors rather than new, pure colors.
Bottom: Using RYB with more saturated colors and higher frequency noise.


Trait 1
Trait 2
Trait 3


Trait 1
Trait 2
Trait 3


Trait 1
Trait 2
Trait 3

References

[1] Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution.* University of California Press, Los Angeles, 1969.

[2] Nathan Gossett and Baoquan Chen. Paint inspired color mixing and compositing for visualization. In *IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 113–118, Washington, DC, USA, 2004. IEEE Computer Society.

[3] Johannes Itten. *The Art of Color*. Van Nostrand Reinhold, New York, 1961.

[4] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97, 1956.

[5] Stephen E. Palmer. *Vision Science: Photons to Phenomenology.* MIT Press, Cambridge, Massachusetts, 1999.

[6] Ken Perlin. An image synthesizer. In *Proceedings of SIGGRAPH85*, pages 287–296. ACM Press, 1985.

[7] Timothy Urness, Victoria Interrante, Ivan Marusic, Ellen Longmire, and Bharathram Ganapathisubramani. Effectively visualizing multi-valued flow data using color and texture. In *Visualization, 2003. IEEE*, pages 115–121, 2003.

A C Code demonstrating RYB to RGB Conversion

```
#include <stdio.h>
#include <math.h>

//Perform a biased (non-linear) interpolation between values A and B
//using t as the interpolation factor.
float cubicInt(float t, float A, float B){
    float weight = t*t*(3-2*t);
    return A + weight*(B-A);
}

//Given RYB values iR, iY, and iB, return RGB values oR, oG, and oB
void subinterp(float iR, float iY, float iB, float* oR, float* oG,
float* oB){
    float x0, x1, x2, x3, y0, y1;

    //red
    x0 = cubicInt(iB, 1.0f, 0.163f);
    x1 = cubicInt(iB, 1.0f, 0.0f);
    x2 = cubicInt(iB, 1.0f, 0.5f);
    x3 = cubicInt(iB, 1.0f, 0.2f);
    y0 = cubicInt(iY, x0, x1);
    y1 = cubicInt(iY, x2, x3);
    *oR = cubicInt(iR, y0, y1);

    //green
    x0 = cubicInt(iB, 1.0f, 0.373f);
    x1 = cubicInt(iB, 1.0f, 0.66f);
    x2 = cubicInt(iB, 0.0f, 0.0f);
    x3 = cubicInt(iB, 0.5f, 0.094f);
    y0 = cubicInt(iY, x0, x1);
    y1 = cubicInt(iY, x2, x3);
    *oG = cubicInt(iR, y0, y1);

    //blue
    x0 = cubicInt(iB, 1.0f, 0.6f);
    x1 = cubicInt(iB, 0.0f, 0.2f);
    x2 = cubicInt(iB, 0.0f, 0.5f);
    x3 = cubicInt(iB, 0.0f, 0.0f);
    y0 = cubicInt(iY, x0, x1);
    y1 = cubicInt(iY, x2, x3);
    *oB = cubicInt(iR, y0, y1);
}

void main(){
    float r1, r2, y1, g2, b1, b2;
    r1 = 1.0;
    y1 = 0.5;
    b1 = 0.25;
    subinterp(r1, y1, b1, &r2, &g2, &b2);
    printf("%f %f %f\n", r2, g2, b2);
}
```