1 Color Space

In order to use linear algebra tools to apply to color space, we model it as a vector space. An example of a color space is RGB space, where colors are represented by the amount of red, green, and blue light they contain. Therefore, each color will have red, green, and blue values on the interval [0,1]. This is not a vector space, because there is no natural way to define addition and scalar multiplication so the system is closed under those operations. However, the following continuous transformation will make it a vector space. Let (r,g,b) be a 3-tuple signifying a color in RGB space such that r,g,b ∈ [0,1]. Let the corresponding color in RGB space be (r’,g’,b’) = \begin{pmatrix} \frac{r}{2} + \frac{1}{2} \\ \frac{g}{2} + \frac{1}{2} \\ \frac{b}{2} + \frac{1}{2} \end{pmatrix}

Notice that given a red value in R, r', we can find easily find r = \frac{2r' - 1}{2} for any real r', and similarly for the other two color channels. Since you can go in either direction by a continuous map, we see the two spaces are homeomorphic.

Now that we have any triple in R³ corresponding to a color, we can talk about color space as a genuine vector space.

2 Operations

We see that scalar multiplication in the above vector space has special meaning. If we multiply each color in a color image by a real number λ ≥ 1, then each pixel gets farther away from (0,0,0), which is middle gray. Therefore, the contrast between the light (positive values) and dark (negative values) will increase. Similarly, multiplication by λ ≤ 1 will decrease contrast. Multiplication by -1 will turn reds into cyans, greens into magentas, and blues into yellows, so multiplication by -1 is the same as inverting the colors.

We can also apply any linear transformation A. For example, given an image M such that the (i,j)th pixel color is given by the vector xᵢ, we see that

\[ A(M) = \{Axᵢ | (i,j) ∈ M \}\]

For example, A could represent an orthonormal matrix, and this would be a “rotation” of the colors of the image.

3 Aligning Colors

Any given image has many different rotations of its colors in color space. Given two images that we suspect are rotations of each other in color space, it would be useful to be able to find the rotation matrix A that takes one image to the other. To accomplish this task, we can use properties of the covariance matrix. Treating the set of K pixels as a population of vectors in R³, we define the covariance matrix as

\[ C = \frac{1}{K} \sum_{i=1}^{K} xᵢ xᵢ^T \]

where xᵢ is the color vector corresponding to pixel i, and mxᵢ is the mean of all the color vectors. C is a symmetric three by three matrix, where the ith entry represents the covariance between colors i and j (or the variance of color i if j = i). Since C is symmetric, it has a full set of orthogonal eigenvectors.

Let Cₓᵧ be the covariance matrix of the image that has been transformed by rotation matrix A. We can now find A simply by computing Cₓᵧ and Cₓᵧ. Theorem 1. For an orthonormal set of eigenvectors v₁,...,vₙ of Cₓᵧ there exists an orthonormal basis of eigenvectors of Cₓᵧ u₁,...,uₙ such that

\[ uᵢ \cdot Axᵢ = \lambdaᵢ xᵢ \]

Proof Let \( uᵢ \cdot Axᵢ = \lambdaᵢ xᵢ \).

One can verify that each \( uᵢ \) defined in this way is an eigenvector of Cₓᵧ. Since the \( xᵢ \cdot xᵢ \) is an orthogonal matrix, \( xᵢ \cdot xᵢ = 1 \). So we see

\[ uᵢ \cdot Axᵢ = \lambdaᵢ xᵢ \]

as desired.

Regardless of whether two images are rotations of each other, we can compute Cₓᵧ for both images and find a candidate for A. We can then rotate the first image by A and then check and see if they are the same image. If they are not, then we know the two images were not simple rotations of each other in color space.

4 Color Clashing

To the human eye, some color combinations look better than others. For example, one might think the look of blue and red together, where orange and teal just do not belong. While this is certainly subjective, there are many modern techniques in image processing, including the idea of using the eigenvectors of the covariance matrix to correct for rotations.

5 Color Variety

Suppose someone is designing a webpage using a certain set of colors. In what ratios should the colors be used to make the webpage as colorful as possible? To model this situation, again assume all colors are under consideration. Some shades of color are very close to each other, while others are radically different. Let gᵢ be some measure of how far the ith color is from the jth color. Then G = [gᵢ] is a matrix that represents these distances. Let \( \beta \) be a matrix of the amount of each color. Then, ideally, each color would contribute enough variety to justify its presence. To model this, we require \( \beta \) to satisfy

\[ \beta x ≤ G \cdot \beta \]  

Notice this is identical to Equation 1, except that the inequality is reversed.

Figure 3: The best combination of the three colors shown above according to this model.

We see G meets the same requirements as before, so again we can apply the Perron-Frobenius Theorem to determine that G must have a positive eigenvalue \( \lambda_{max} \). This is the associated positive eigenvector \( \beta_{max} \) that satisfies Equation 2. Clearly, any \( \beta ≤ \beta_{max} \) will satisfy it with the same \( \beta \). Can the inequality be satisfied with \( \lambda ≤ \lambda_{max} \)? Just as before, the answer is no.