

Texturing Coloured Images in Black and White

Hank Guss

Student of Mathematics and Computer Science, Oberlin College

35 E 85th St, New York, NY

henryguss@gmail.com

Abstract

I present a strategy for representing colour images using only white lines on a black background. This method maps colour channels to distinct line directions, varying the width of the lines based on the magnitude of the corresponding colour channel.

Introduction

There are any number of methods for greyscale reproduction of images, and each seems to solve a different problem. We can do much to re-create images in black and white more accurately—we can use dithering and error-diffusion to make optimal use of a small set of brightness values [1], we can weight colours of an image by our eye's natural preference [2], and we can highlight contrast between regions [3]. While highlighting contrast is useful for identifying region borders that would ordinarily be difficult to locate in a greyscale image, it does nothing to relate regions of similar colours. By associating directions with colour channels, we can represent an image in black and white while still visually informing the viewer of not only the contrast between the original colours, but also colour similarities between objects. To demonstrate the effectiveness of this strategy, I have included an image below wherein every pixel has the same average brightness value, and a corresponding tiling.

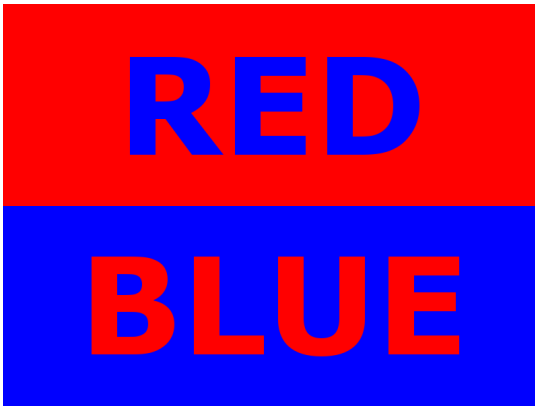


Figure 1 : *A pure red and blue image.*

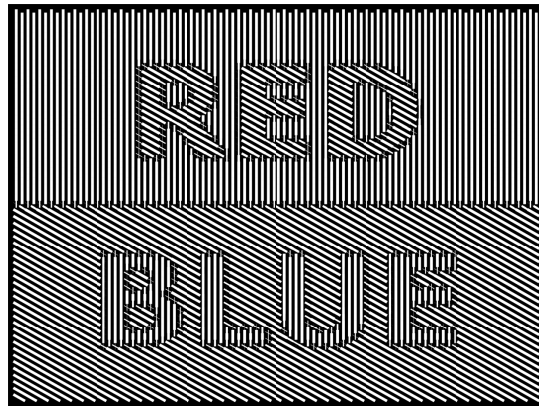


Figure 2 : *Figure 1 coloured directionally.*

Figure 2 is the result of ‘colouring’ this image using my procedure. Despite a flat greyscale image (by a direct averaging method), the variant line directionality produces recognizable edges between regions of contrasted colour, and informs the viewer that the lettering on the top half is coloured the same as the background on the bottom half and vice-versa. It is important to note that though most printers use a luminous averaging of colour channels to more closely match how the average human eye detects light, one could simply use the same weighting to create a similar image that would under a luminous greyscale conversion appear flat, and this algorithm could weight colours in the same convention. For the purposes of this paper, I will be assuming an unweighted greyscale method.

The Method

Images stored in computers by and large use red, green, and blue as primary colour channels. In order to represent these channels separately, I have chosen to portray them with line segments extending in three different directions (one for each colour channel). In order to keep the angles between these lines consistent, and to keep the lines from one tile to the next continuous, I have opted to use a hexagonal grid to tile my image reproductions, because the lines can be drawn to connect opposite edges the hexagon. A sample tile is shown in Figure 3.

To produce these directionally coloured images, the first step is to overlay the image with a hexagonal grid, and then to assign each pixel to the closest overlain hexagon. An example of this step of the process can be seen in Figure 7. Once this is done, each hexagon can be assigned an average value for each colour channel, where the average is taken over all pixels assigned to the hexagon. From here, the original image is no longer necessary. The hexagons are then divided by a constant number of lines (3, in Figure 3) in each direction. The thickness of these lines is proportional to the square (starting from a range [0,1]) of the corresponding colour value for the hexagon. I have also included a minimum colour channel value (threshold), such that if there is very little of a particular colour within a region, the hexagonal tile will exclude lines in the corresponding direction. In the Mondrian-esque piece below (Figure 4), the bold red, green, and blue regions are replaced with lines that run only in a single direction—in Figure 5, the regions can be identified to be the same colour by having lines in the same direction. When two colours are present in a region, cross-hatching occurs. Secondary colours can be identified by the rotational angle of the resulting rhombuses. Viewing the Mondrian on the left in grayscale, it can be difficult to match some rectangles with the rectangle of the same colour. However, a glance at the representation on the right reveals that the top left and top right are the same colour, and that the middle left and middle right rectangles are the same colour. This shows that regions and objects of no relation or proximity can be identified as having similar colours through this process, even though the resulting image is pure black and white.

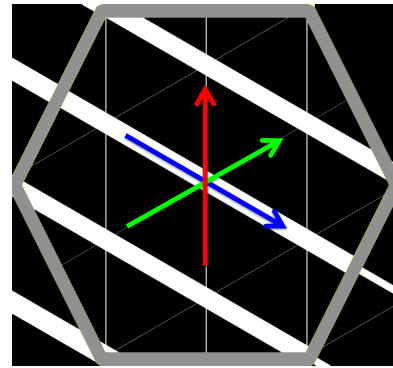


Figure 3 : *A hexagonal tile with three sets of lines corresponding to colour channels.*

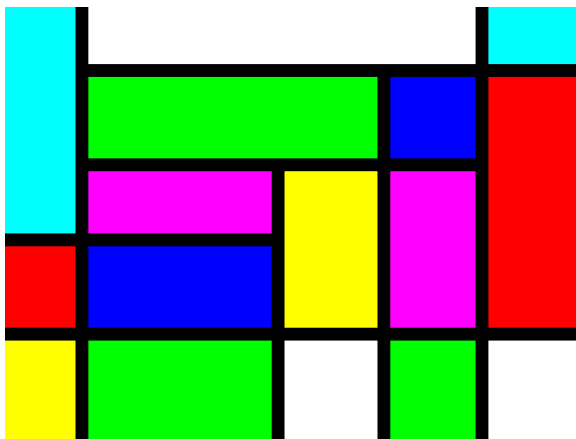


Figure 4 : *A Mondrian style piece in RGB.*

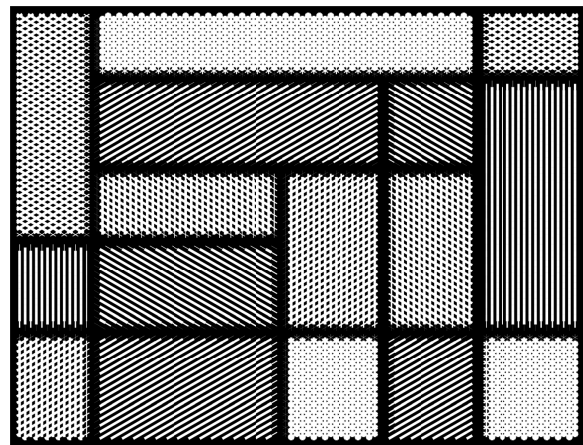


Figure 5 : *The same piece coloured directionally.*

This can also be seen in Figure 6 (a colour wheel I created using an earlier project of mine called Coral [4]), and Figure 8, its corresponding colouring. This example shows the result of applying this procedure to a gradation of colour magnitudes for each of the primary and secondary colours of light. Note here the range of thickness available for the lines, and that the thickness is independent between colour channels.

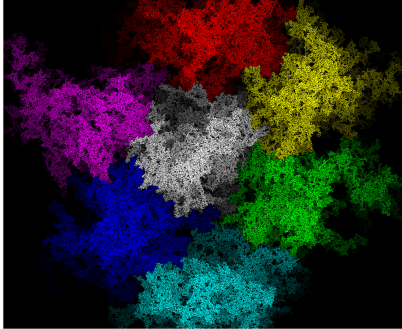


Figure 6 : *A colour wheel in RGB*

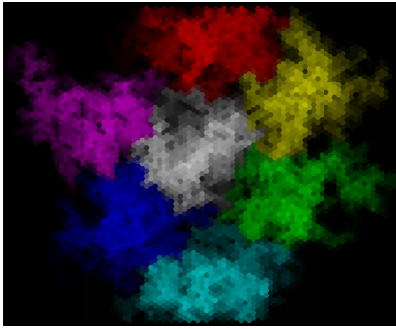


Figure 7 : *Lossy Segmentation*

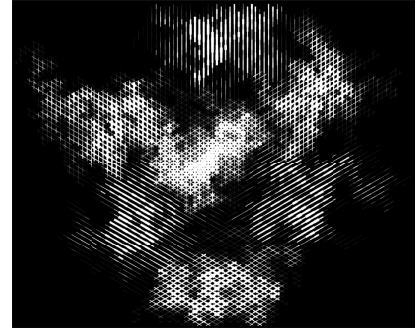


Figure 8 : *Directional 'colouring'*

In Figure 9, we can see a similarity in colour between the green background and the green on the parrot's wing. In Figure 10, we can see that those same regions have the same texture. In the tiled image we can see that the two branches are textured the same, and we can even see that the red and white of another parrot's beak on the right side is textured the same as the one in the center. The directionality of these lines draws attention to similarities that are lost in most conversions to greyscale.



Figure 9 : *A parrot, eating. [6]*

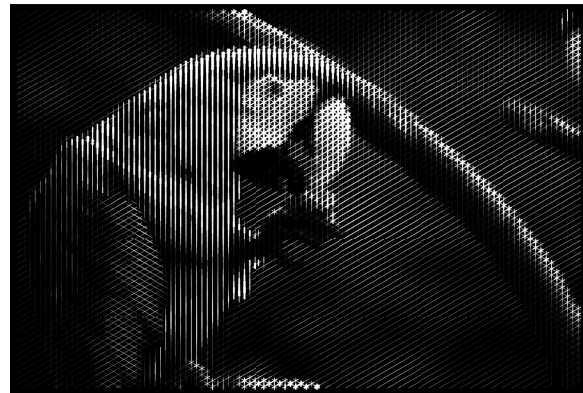


Figure 10 : *The same piece coloured directionally.*

While this algorithm preserves a sense of relationships between regions, its usefulness decreases as the resolution of the tiling increases. As the tiles become small, the intersections become hard to identify and the algorithm approaches a simple greyscale rendering of the original image. Figure 12 shows an example of this algorithm at high resolution, where it is hard to make a map between similarly textured regions. Still, closely examining this image reveals a similarity between the water and the air, as these were coloured similarly in the source image, as well as between the land in the foreground and the land on the horizon. It is also worth noting that if the resolution is increased such that a single hexagonal tile corresponds to a single pixel of the source image, the representation created is *not lossy*. It fully encodes the colour values at every pixel (though it still displays them on a hexagonal grid) and given a relation between brightness values and line thickness, a machine could reproduce the source image from its textured representation.



Figure 11 : *A lake. [5]*



Figure 12 : *The same piece coloured directionally.*

References

- [1] Ulichney, Robert. Digital Halftoning. Cambridge, MA: MIT, 1987. Print.
- [2] T. Helland, Seven Grayscale Conversion Algorithms (2011), <http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/> (as of Apr. 29, 2016)
- [3] Gooch, Amy A., Sven C. Olsen, Jack Tumblin, and Bruce Gooch. "Color2Gray: Saliency-Preserving Color Removal." Northwestern University. <http://www.cs.northwestern.edu/~ago820/color2gray/color2gray.pdf>. (as of 27 Apr. 2016)
- [4] H. Guss, Coral (2015), <http://www.hankguss.com/coral> (as of Mar. 15, 2016)
- [5] Unsplash, Lake (2015), (as of Mar. 15, 2016)
<https://pixabay.com/en/lake-water-nature-landscape-clear-768659/>
- [6] Condesign, Parrot (2015), (as of Mar. 15, 2016)
<https://pixabay.com/en/parrot-bird-colorful-ara-plumage-855493/>