

Illusion Workshop Project I: Random Dot Stereogram

PSYCH 30: Introduction to Perception

Alejandrina G.R.

October 12, 2020

Introduction

To generate a Random Dot Stereogram (RDS) for this project, I created a simple program in JavaScript. My program works the following way:

1. Input image selection
2. Image threshold processing
3. Generating random dot pattern from image
4. Generating a red dots layer
5. Generating a cyan dots layer
6. Generating a combined red and cyan dots image

The steps will be described in further detail in the sections that follow.

1 Input Images

For my project, I wanted to create a 3D Snoopy because I love beagles. But I also included a Stanford logo to double check the effectiveness of my code.



Figure 1: Snoopy image used.



Figure 2: Stanford logo image used.

2 Threshold Image Processing

Threshold image processing, so that if given a color image it partitions the image's contrast to binary black and white format. Hence, this step is also why I also included the red Stanford logo, to make sure threshold processing contributed to the overall process. Furthermore, this step makes it easier for later steps when generating red and cyan dot pairs.



Figure 3: Threshold processing of Snoopy image.



Figure 4: Threshold processing of Stanford logo.

3 Random Dot Pattern Image

From each image, the program selects random pixel (x, y) coordinates of the threshold image. Each randomly picked pixel is analysed to check if it is a black colored pixel or a white colored pixel from the threshold image in order to see how much horizontal Δx shift should be added between a pair of red and cyan dots for that random location. For instance, white pixels have twice the space in between red and cyan dots than black pixels, since we want binary black and white sections of the image to have different perceptual depths.

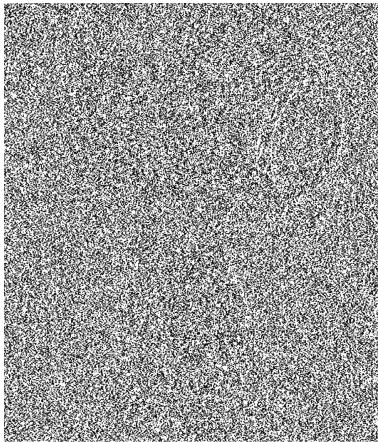


Figure 5: Random dots of Snoopy image.

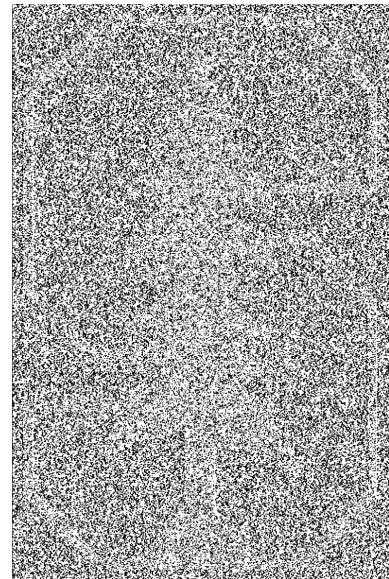


Figure 6: Random dots of Stanford logo.

4 Red Dots Images

Below are layers of red random dots for each input image. A red dot's position is determined by $(x + \Delta x, y)$, where (x, y) is the position of the randomly picked pixel. And where $\Delta x = -2 * \text{Dot's Width}$ if the pixel is white; or $\Delta x = -\text{Dot's Width}$ if the pixel is black

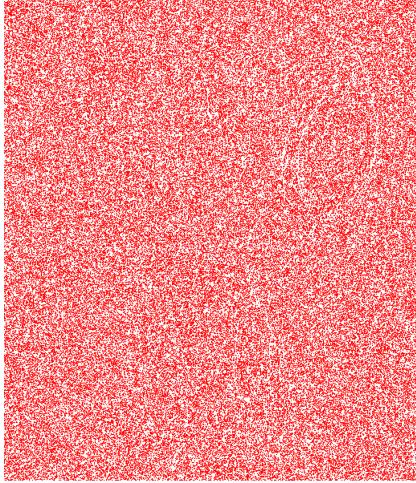


Figure 7: Red layer of Snoopy image.

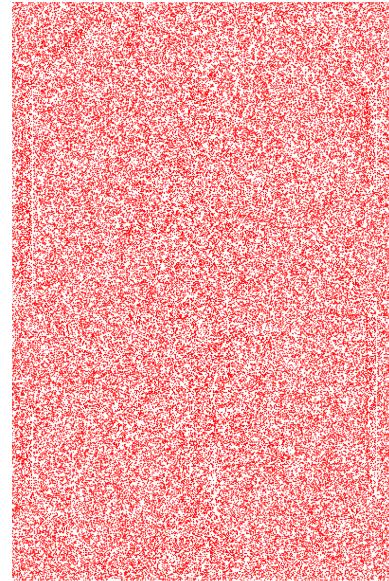


Figure 8: Red layer of Stanford logo.

5 Cyan Dots Images

Below are layers of cyan random dots for each input image. A cyan dot's location is determined by $(x + \Delta x, y)$, where (x, y) is the position of the randomly picked pixel. And where $\Delta x = 2 * \text{Dot's Width}$ if the pixel is white; and $\Delta x = \text{Dot's Width}$ if the pixel is black.

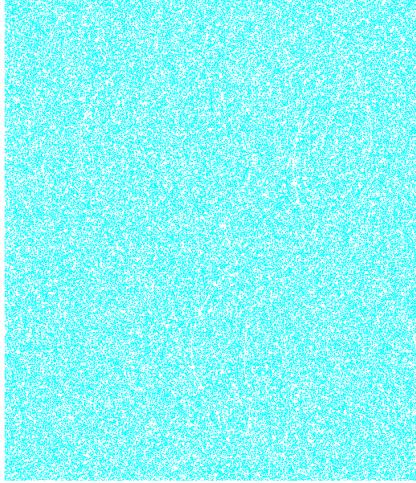


Figure 9: Cyan layer of Snoopy image.

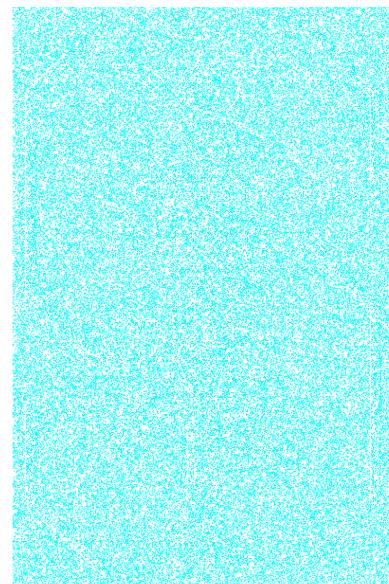


Figure 10: Cyan layer of Stanford logo.

6 Final RDS Images

Finally, below are the final random dot stereograms for the Snoopy image and the Stanford logo that were generated from the steps described above.

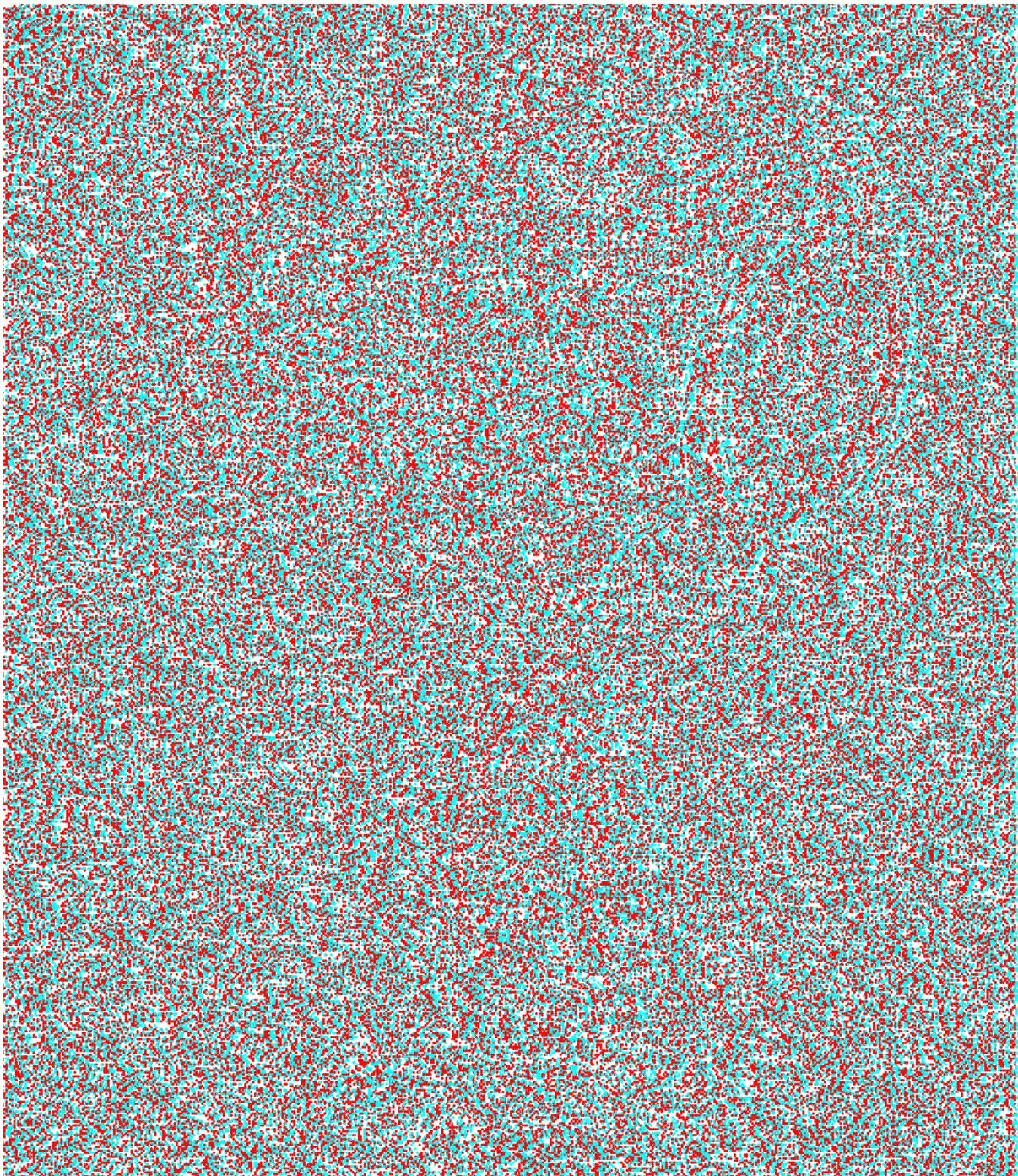


Figure 11: Final RDS of Snoopy image.

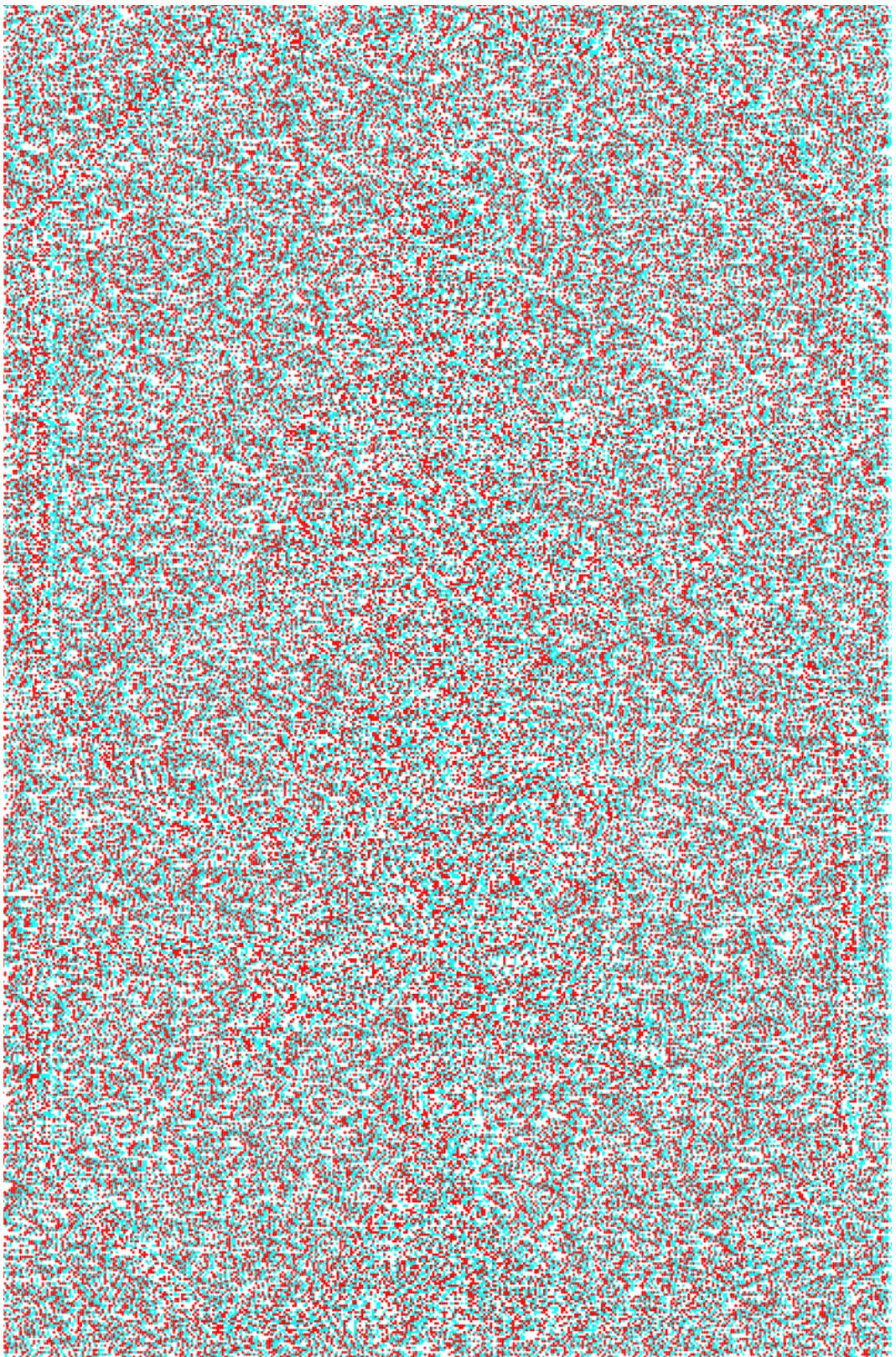


Figure 12: Final RDS of Stanford logo.

Conclusion

Overall, this was a fun and interesting project.

The code for my project can be found in this [Google Drive folder link](#). The folder of the code also includes a folder with the resulting RDS images and the step images included in this report.

Thanks to the PSYCH 30 teaching team for a great class.