

Voronoi Stippling Report

Stephen Arnold - G00864316

CS633 - Computational Geometry

1 Summary of the two methods

In this introductory section, the methods and algorithms employed by the two stippling techniques shall be reviewed.

1.1 voronoi method

The provided implementation of Voronoi Stippling makes use of Fortune's Algorithm for the calculation of the Voronoi Cells. Initially, the input parameters are read in and stored. Next, an initial random distribution of points is created within the boundaries of the original image. These data points are the initial configuration stored in the stippler data structure.

Using an iterative approach, a Voronoi Diagram is created from the set of initial points using Fortune's Algorithm. Next, the applied stipples are redistributed and the displacement from the new locations to the previous centroids is calculated. This process is repeated until the average change in displacements across all cell meets a minimum threshold. Once the centroids have settled, the image is stippled by placing circles of appropriate size centered at the centroids.

1.2 hedcuter method

Using an iterative approach, the hedcuter stippler uses simple rejection sampling to generate an initial distribution of points, or centroids, within the confines of a given image. The algorithm compares the grayscale value of the randomly selected location within the image and applies a gaussian scaling factor. The point is either accepted or rejected based on satisfying a predetermined threshold. Once the initial points are determined, a set of right cones are generated with apexes at the cell generator. Initially the cones have equal height. Each cone is also assigned a unique "color" to identify it later. The intersection of the cones in the z-direction will determine to which cone the pixels of the original image belong. In this way, the pixels are assigned a unique value which determines to which Voronoi Cell the pixels belong.

The weighted Centroidal Voronoi Tessellation (CVT) used by the hedcuter algorithm makes use of a density function to distribute, and realign, the centroids of the Voronoi Diagram. Darker areas (lines, edges, etc.) are assigned a higher density than lighter areas. As the algorithm iterates through the image, centroids tend to migrate towards higher density regions.

The stippling method invoked by the hedcuter places a variably sized disk, centered at the centroid of the cell. Two options exist within the code: uniformly sized disks and variably-sized disks. Various levels of gray may be obtained by uniform-sized disks at various spacing densities or by

varying the sizes of the placed disks. Colored disks may also be used in conjunction to create various tones. [1]

***** uses precomputed stipple "tiles" to fill a Voronoi cell. Each pixel in the image is examined and an appropriate stipple pattern applied

2 Comparison of the two methods

After compilation of both stippling methods, various test cases were run to compare each methods effectiveness as a stippling method. Those results shall be discussed here.

A single image was used to allow for simple comparison of the significant variables of interest. The image, seen in Figure 1, is a grayscale PNG-format image of Abraham Lincoln, America's 16th president, from the year 1865. Unless otherwise noted, this image was used for all variations.

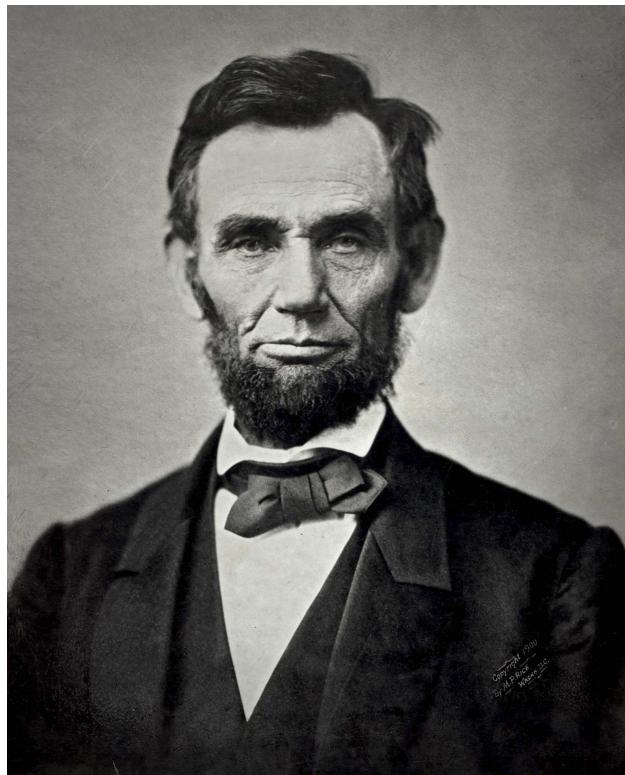


Figure 1: Abraham Lincoln, 1865.

2.1 Multipass Results

To determine if there are any statistically significant variations in running the same program on the same image multiple times, a script was written to test this condition. For VM, the results of each run using 4000 points, subpixel density 5, non-overlapping, and variable radius were extremely consistent. Minor variations in the duration of the runs were seen. Over 5 runs, times ranged from 56.78 seconds for

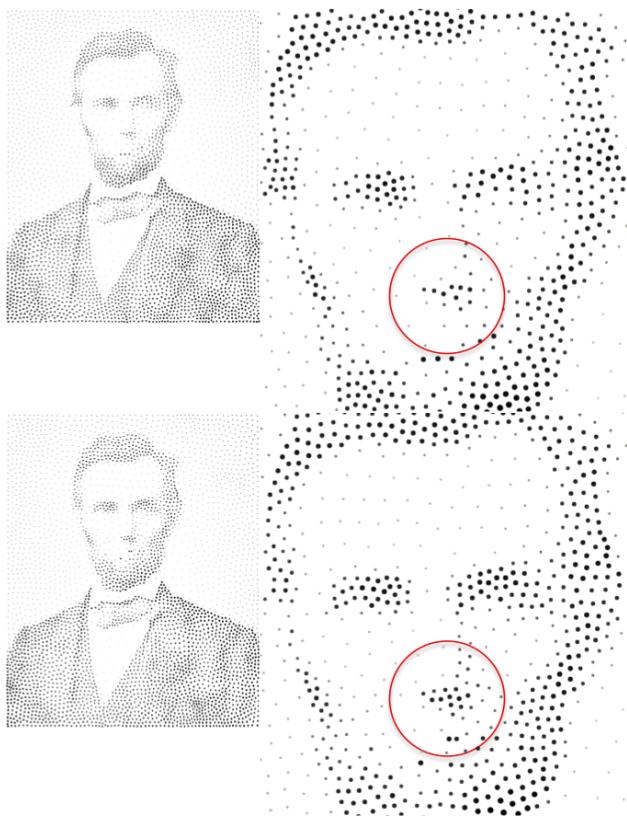


Figure 2: Pixel placement differences between hedcuter runs.

Figure 2 shows the pixel differences between different hedcuter runs.

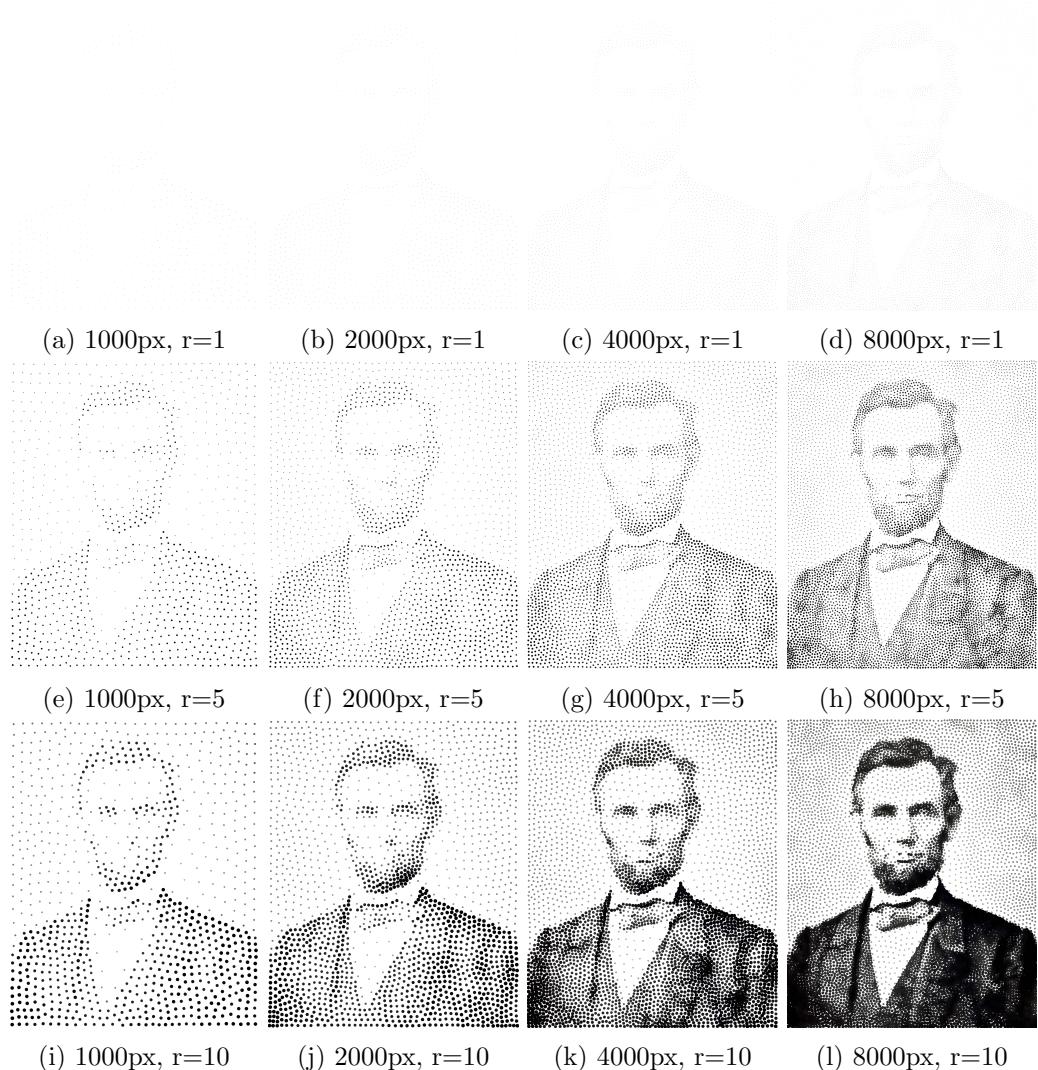


Figure 3: Hedcuter images varying number of disks. Radius = 1.

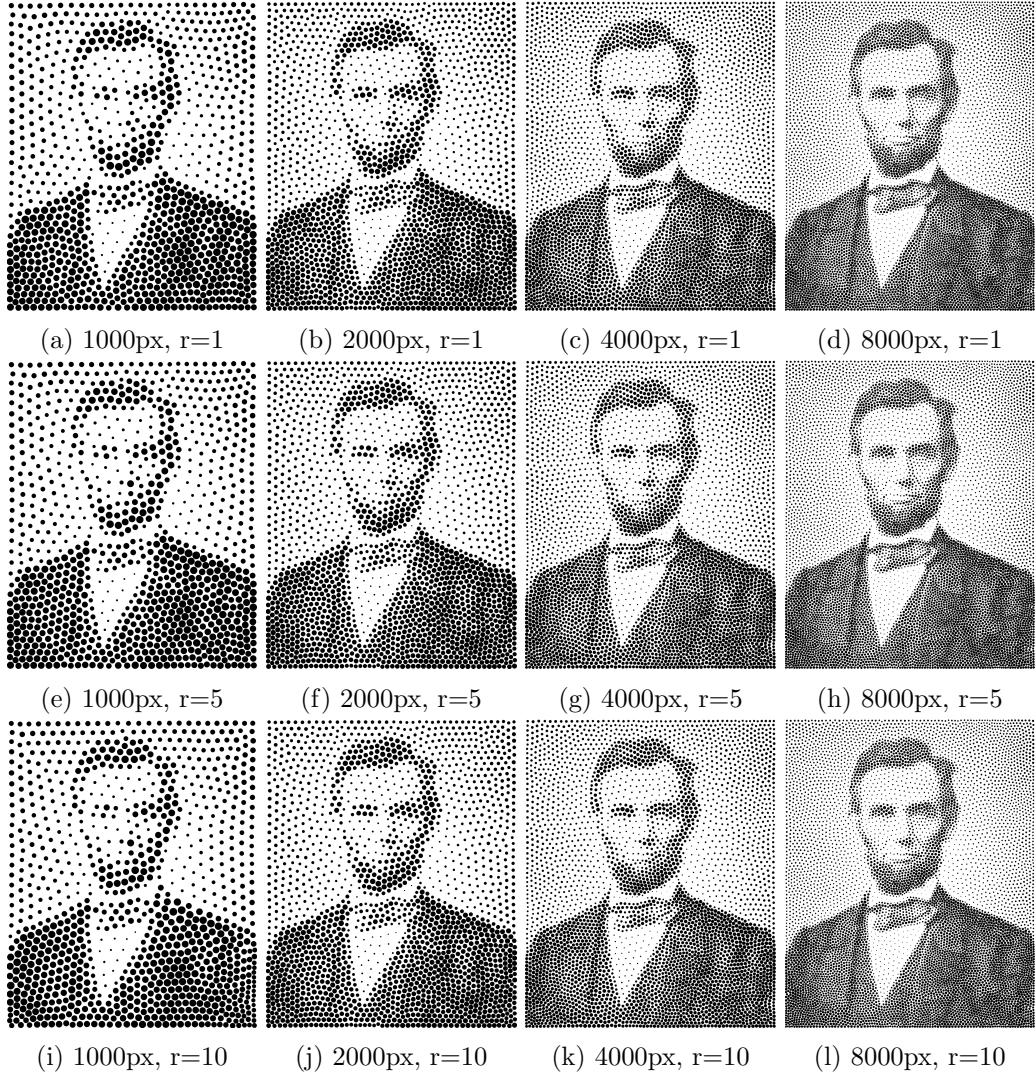


Figure 4: Hedcuter images varying number of disks. Radius = 1.

2.2 Variations on Number of Disks

2.3 Variations on Speed with Number of Disks

2.4 Differences in image type

2.5 Differences from WSJ Hedcuts

3 Improvement of hedcuter method

References

- [1] Adrian Secord. Weighted voronoi stippling. In *Proceedings of the 2nd International Symposium on Non-photorealistic Animation and Rendering*, NPAR '02, pages 37–43, New York, NY, USA, 2002. ACM. [1.2](#)