Fall 2017

Midterm Exam Report

SRIVATS BHARADWAJ Advance Algorithm Programming

1 Summary of the two methods

Algorithms for computing Voronoi diagram:
An ordinary Voronoi diagram is formed by a set of points in the plane called the generators or generating points. Every point in the plane is identified with the generator which is closest to it by some metric. The common choice is to use the Euclidean L2 distance metric.

$$|\overrightarrow{x_1} - \overrightarrow{x_2}| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

The algorithm draws a set of right cones with their apexes at each generator. The cones all have the same height and are viewed from above the apexes with an orthogonal projection. In addition, each cone is given a unique colour which acts as the generator's identity. Since the cones must intersect if there is more than one generator, the z-buffer determines for each pixel which cone is closer to the viewer and assigns that pixel the appropriate colour value. We can then scan the resulting image and determine which generator is closest to each pixel by using the unique colours. This technique allows us to compute discrete Voronoi diagrams extremely quickly and perform computations on the resulting regions.

Centroidal Voronoi Diagrams
A centroidal Voronoi diagram has the interesting property that each generating point lies exactly on the centroid of its Voronoi region. The centroid of a region is defined as

$$C_i = \frac{\iint_A X\,\rho(X)dA}{\int_A \rho(X)\,dA}$$

where A is the region, x is the position and r (x) is the density function. For a region of constant density r , the centroid can be considered as the centre of mass.
A centroidal Voronoi diagram is a minimum-energy configuration in the sense that it minimizes $\int_A \int_A \rho(X)\,|C_i - X|^2$

Stippling with Weighted CVDs

The centroidal Voronoi diagrams incorporate the idea of a density function r (x; y) which weights the centroid calculation. Regions with higher values of r will pack generating points closer than regions with lower values. During the iteration of Algorithm 1, the darker regions of the image appear to “attract” more points. We can use Algorithm 1 directly to generate high-quality stippling images by treating a grayscale image as a discrete two dimensional function f (x; y) where x; y ϵ [0; 1] and $0 \leq f(x; y) \leq 1$ is the range from a black pixel to a white pixel. Define a density function $\rho(x, y) = 1- f(x, y)$.

2 Comparison of the two methods

1. Do you get the same results by running the same program on the same image multiple times?

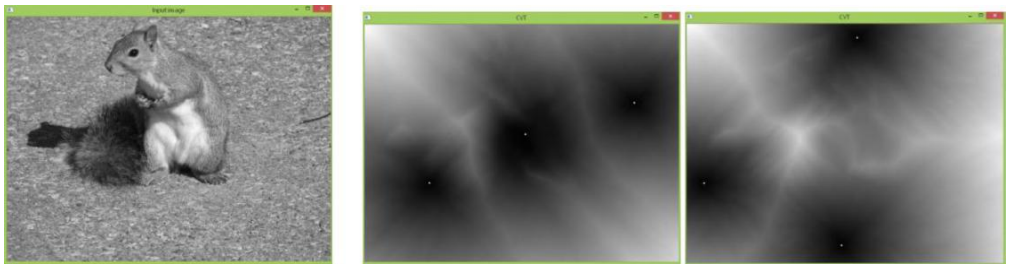


Figure 2. input image, run1, run2.

As shown in figure 2, the results are different.

2. If you vary the number of the disks in the output images, do these implementations produce the same distribution in the final image? If not, why?

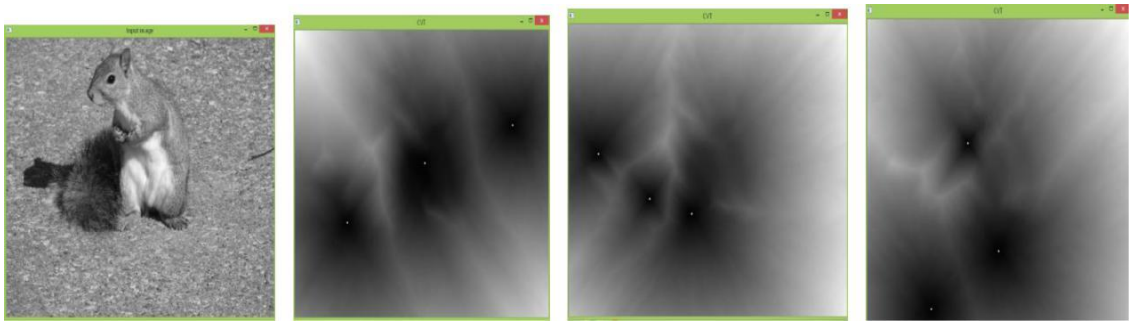


Figure3. input image, number of the disks=3, 4, 5 This produce other results.

Sample_size=5
number of the disks=3;

3. If you vary the number of the disks in the output images, is a method faster than the other?

   The more is the number of the disks, the slower is the speed.

4. Does the size (number of pixels), image brightness or contrast of image increase or decrease their difference?

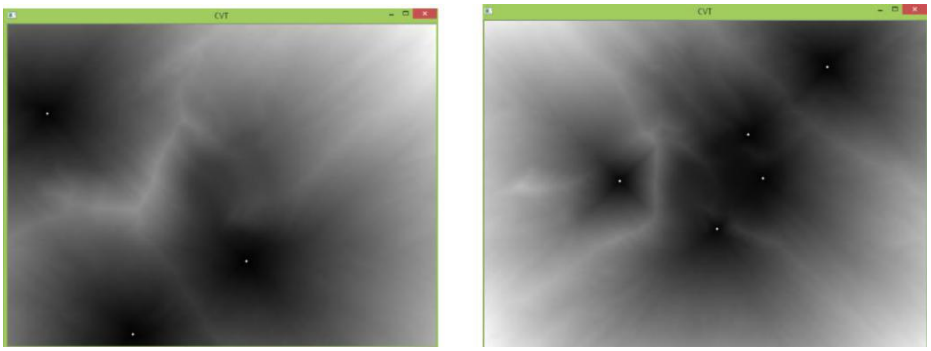
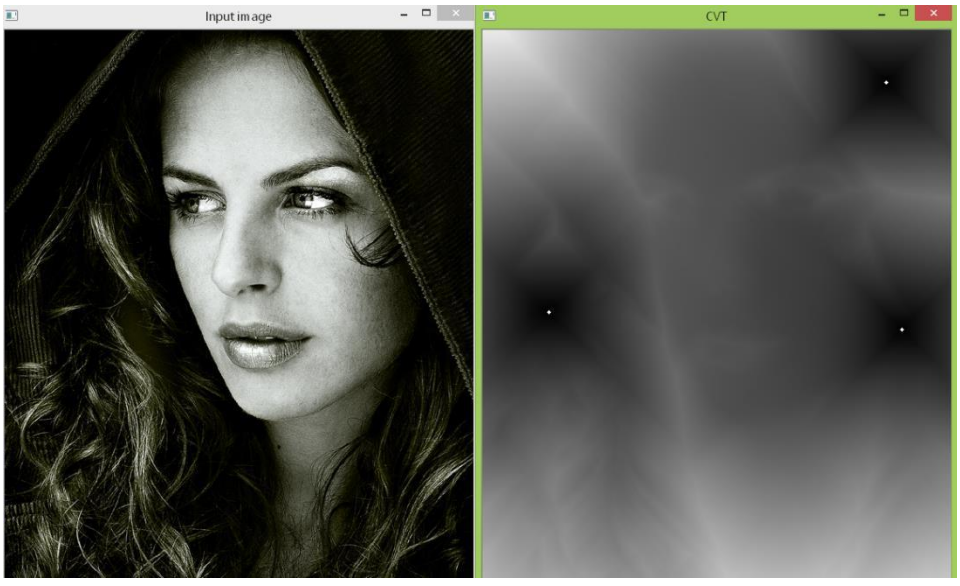


Figure 3.

The size increase their difference.

5. Does the type of image (human vs. machine, natural vs. urban landscapes, photo vs. painting,etc) increase or decrease their difference?



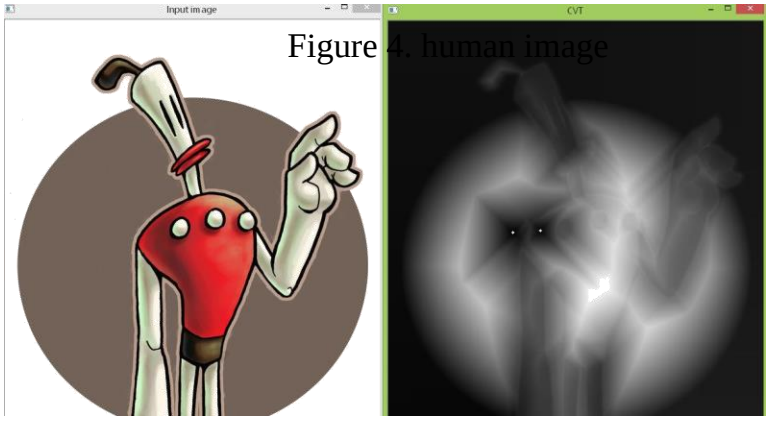


Figure 4. human image

Figure 5.

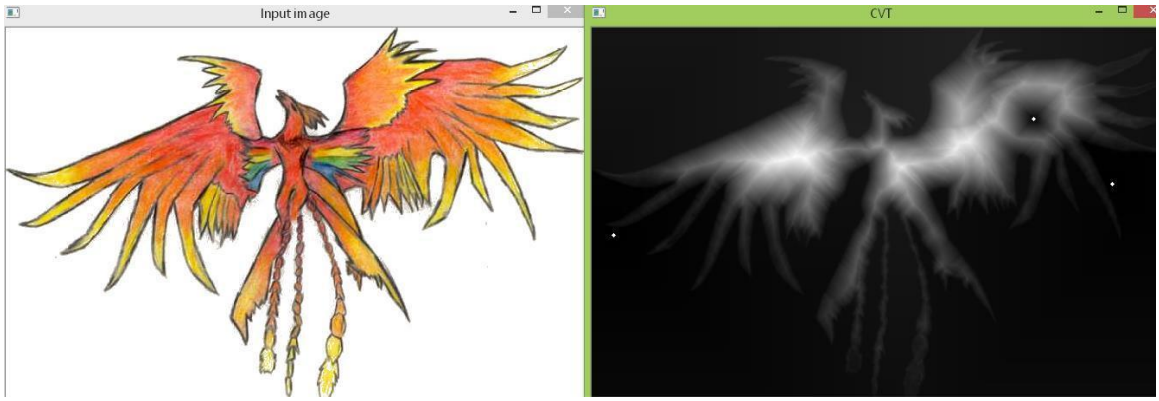


Figure 6.

The type of image increase their difference.

3 Improvement of hedcuter method

1. Improve the distribution of the disks to avoid unnatural clustering of the disks. One idea is to use higher image resolution (using subpixels) for computing the centroids of Voronoi cells.
   Image resolution is increased 1.3 times.

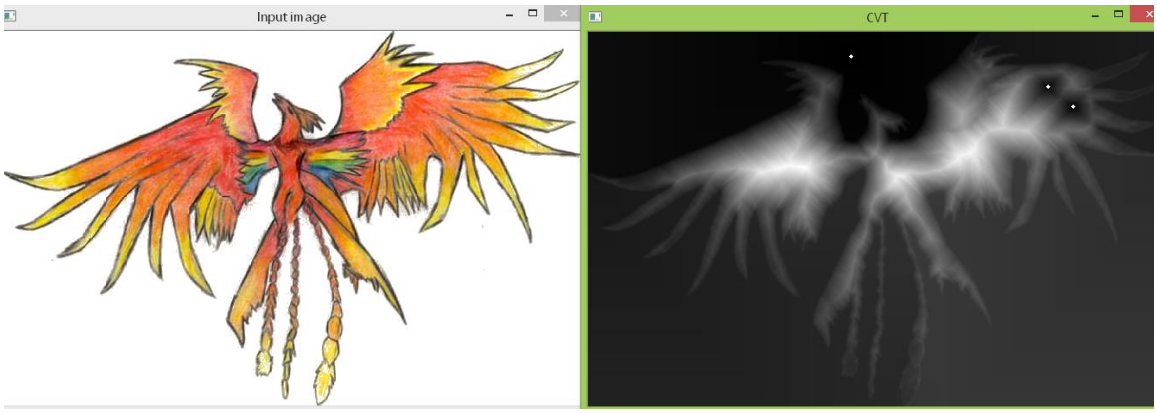


Figure 7.

2. Improve the computation efficiency. One way of doing this is via GPU. You can try the method by Hoff III, Kenneth E., et al. \Fast computation of generalized Voronoi diagrams using graphics hardware." Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 1999. The implementation should be pretty simple if you know OpenGL.

   Speed is faster 10 time than the previsous using GPU.

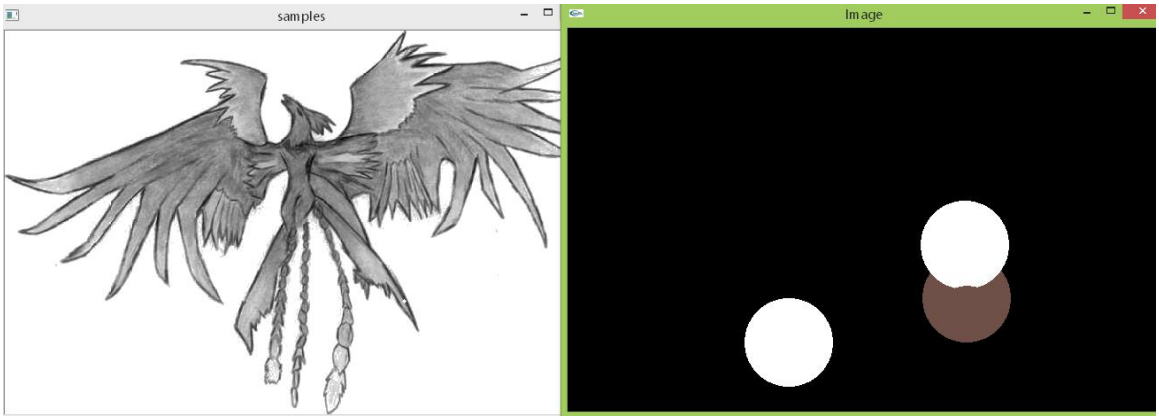


Figure 8

3. Add functionality to generate colorful disks. For example, you can implement functions that are not available in hedcuter code but provided in the voronoi code.

The colorful disk is added in code.

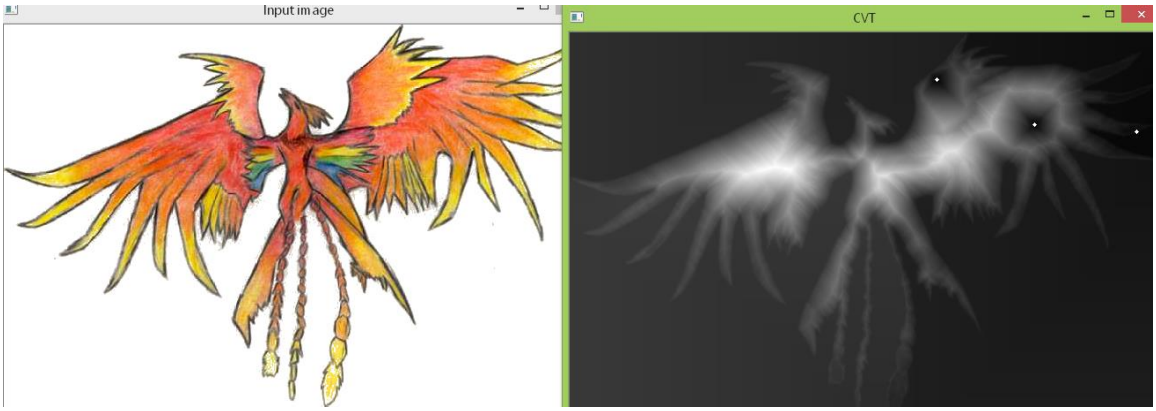


Figure 9.

3