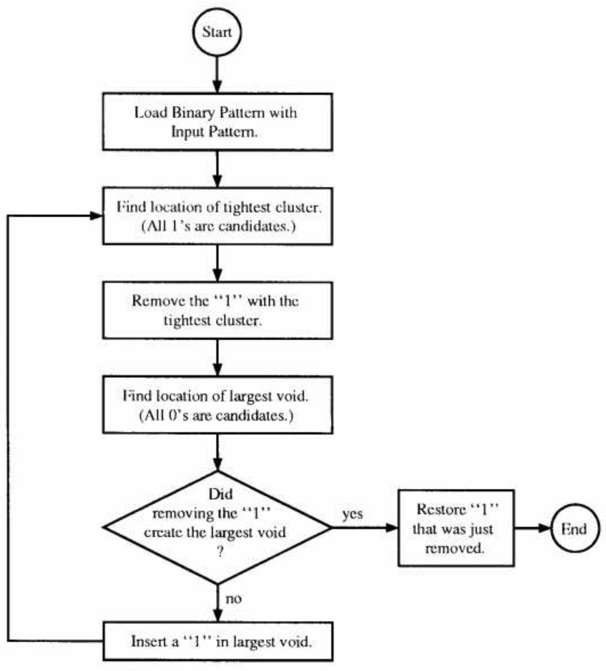
ENGI 7894/9869 Assignment 4: GPU Halftoning

Due: Thursday, July 21 at 11:59 pm

Halftoning refers to rendering a continuous tone image with only monochromatic dots, from which the illusion of continuous tones is created by varying the locations and spacing (or sometimes the size) of the dots. An example of halftoning an image is shown below:

In this assignment, you are asked to implement the first step of a popular method for halftoning called the void-and-cluster method, on a GPU.

The first step consists of generating a pattern called the Initial Binary Pattern where the dots are roughly evenly spaced, yet not forming patterns that are too "regular".


Start
Load Binary Pattern with Input Pattern.
Find location of tightest cluster. (All 1's are candidates.)
Remove the "1" with the tightest cluster.
Find location of largest void. (All 0's are candidates.)
Did removing the "1" create the largest void ?
yes
Restore "1" that was just removed.
End
no
Insert a "1" in largest void.

Consider the flow chart above given in the paper "The void-and-cluster method for dither array generation" by R. Ulichney (see cv.ulichney.com/papers/1993-void-cluster.pdf).

You may assume the Input Pattern in the first step above is a two-tone 128x128 pixel image (black and white), where roughly 10% of the pixels are white. For each pixel in the Input Pattern, a random floating-point number between 0 and 1 is generated. If that number is greater than 0.1, the pixel's intensity is set to 0, otherwise, it is set to 255.

Otherwise, a void-and-cluster finding filter is used to find the locations of the largest void (set of pixels mostly black) and the largest cluster (set of pixels mostly white). This filter is defined as follows:

$$DA(x,y) = \sum_{p=-64}^{63} \sum_{q=-64}^{63} BP(p',q') f(p,q),$$

where $p' = (128 + x - p) \bmod 128$ and $q' = (128 + y - q) \bmod 128$, and $f(p,q) = e^{-\frac{p^2+q^2}{4.5}}$.

The largest void is determined as being at the pixel at location $(x,\ y)$ which gives the minimum value of DA, while the largest cluster is determined as being located where DA attains its maximum value. You are asked to implement the method in the flow chart above on a GPU using either OpenCL or Cuda to generate the Initial Binary Pattern and output it to a BMP file using the utility routines given in utils.h and utils.cpp provided in D2L.