The fractal dimension is a ratio that determines how patterns scale as they are measured in different ways. For example, a 2-D object scales by a power of two, 3-D by three, and a line (1-D) just scales by a power of one. Fractals are useful when we don’t have such neat objects to examine. One of the most famous examples was Mandelbrot using Fractal’s to predict the measurement of the coast of England. In general, Fractals give us a way to measure scale of natural objects that are generally not ‘smooth’, something that comes in handy when we end up dealing with natural shapes.



In the Fractal Dimension code for this project, we implement two methods for determining the fractal dimension of a particular dimensional object. We first use the box-counting method. The method is iterative in that we begin with one box the size of the array/image we are interested in, and then partition the box into smaller and smaller scales still covering the entire image. At each step, we count the number of boxes that land on borders of the image and store the data. After n iterations, we plug our data into a best fit line and the resulting slope is our fractional dimension, up to a specific error term.

The second process we use is similar to box counting, reticular cell counting. It is similar to box-counting in that we partition our array into a grid of boxes and then count the boxes based on their underlying grayscale count. This will give us an idea on how many onject pixels are in each box. From this we once again plug the data into a linear model to obtain the slope which is the negative of the fractal dimension we care about.

The generalized fractal dimension formula is N = (1/r)D where N represents the number of boxes we count, 1/r is the ratio that we are scaling down by, and D determines the fractional dimension of our array. Thus, D = -log(N)/log(R).