Genetic Algorithm for Mimicking Image

*Thomas Laverghetta*,   
Old Dominion University – Computational Modeling and Simulation Engineering

# Introduction

# Algorithms

In this section, I will be discussing the *Genetic Algorithm* used to generate mimicking source images using triangles. A genetic algorithm is a search heuristic that is inspired by Charles Darwin’s theory of natural evolution (Mallawaarachchi, 2017). The process of natural selection starts with fittest individuals having higher probability to survive, and therefore, will have offspring (inherit the characteristics of the parents and will be added to the next generation). Their offspring should be better than their parents and have better chances at survival. This process iterates until a steady state has been reached where the difference in fitness is minuet.

Algorithmically, the process of natural selection can be expressed into five phases: initial population generation, fitness function, selection, crossover, mutation. The initial population is the initial population to which natural selection can occur. Fitness function is a way of determining the likelihood of survival for individuals. Selection is the process of selecting individuals to have offspring. Crossover is the process of augmenting two individual’s chromosomes to generate a new chromosome (new offspring). Mutation is a random occurrence that modifies new offspring’s chromosome. This ensures that the population does not reach steady state too fast.

The following next sections will be pseudocode for fitness, selection, crossover, and mutation functions.

## Fitness Function

The fitness function, as stated before, calculates how fit an individual to survive (how close it is to the goal or source image for this application). The fitness score I will be using was taken from a genetic algorithm Peng Ding created and posted to GitHub (Ding, 2016). Since it is on GitHub, it is opensource. He calculated the root-mean-square difference between images. This method can also be found under example 19 of ProgramCreek’s Python PIL.ImageChops.difference() webpage (ProgramCreek, n.d.). The following is Ping Ding’s fitness that I used in my program (pseudocode mixed with Python syntax):

*Fitness (genetically generated image (img))*

*1. h = ImageChops.difference(source image, img).histogram()*

*2. fitness = sqrt(reduce(operator.add, map(lambda h, i: h \* (I ^2), h, list(range(256)) \* 3 / (area of source of image in pixels))*

*3. return fitness*

## Selection Function

The selection function, as mentioned previously, selects two parents to mate to create new offspring. To select the new parents, I did a weighted random parent selector. Where individuals (parent candidates) with

*Fitness (genetically generated image (img))*

*1. h = ImageChops.difference(source image, img).histogram()*

*2. fitness = sqrt(reduce(operator.add, map(lambda h, i: h \* (I ^2), h, list(range(256)) \* 3 / (area of source of image in pixels))*

*3. return fitness*