BESET

Ben's Evolutionary Shape Extraction Tool

Ben Poole

Summer 2007

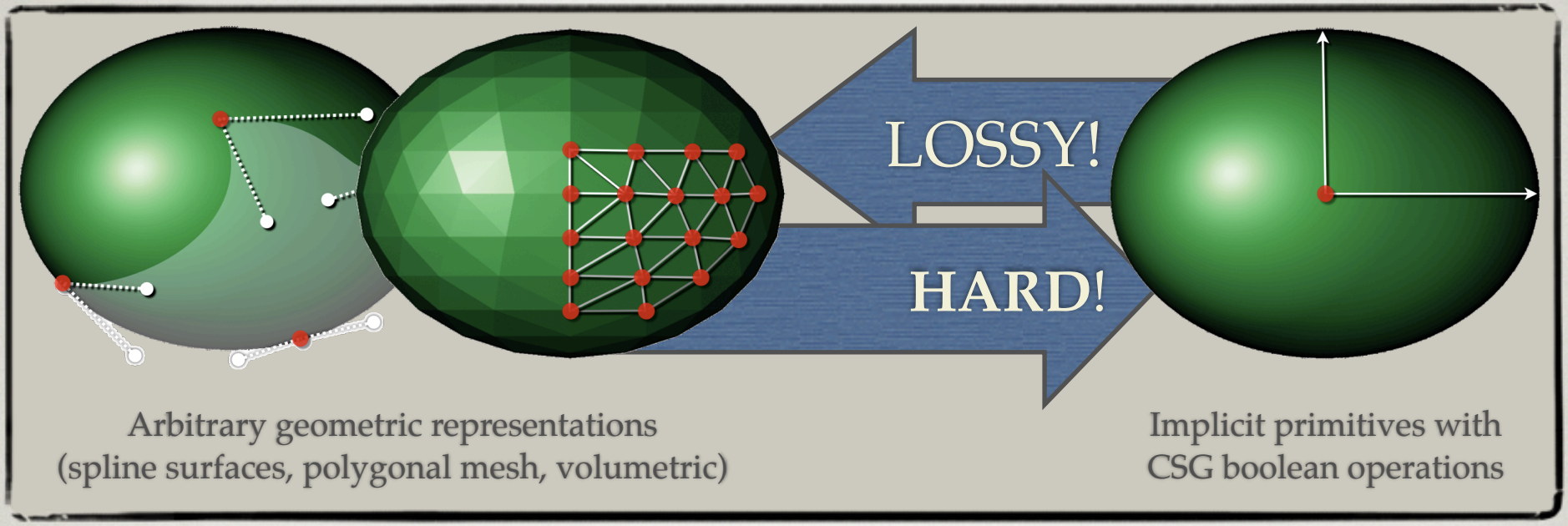
Beset Overview

Beset is an experimental **shape extraction** research project developed by summer student Ben Poole in 2007

Ben worked (remotely) on designing and implementing a genetic algorithm technique to “evolve” a CSG model that matches an arbitrary input geometry model

Why? ..
Some Potential Benefits

- Automatic reverse tessellation
  (e.g., go from a Bag of Triangles back to CSG implicits)
- Geometry compression (compact representation)
- Conversion from explicit to implicit geometry representation (good analysis properties)
- Topological structure, volume occupancy, and connectivity detection
- Advanced computer graphics research topic

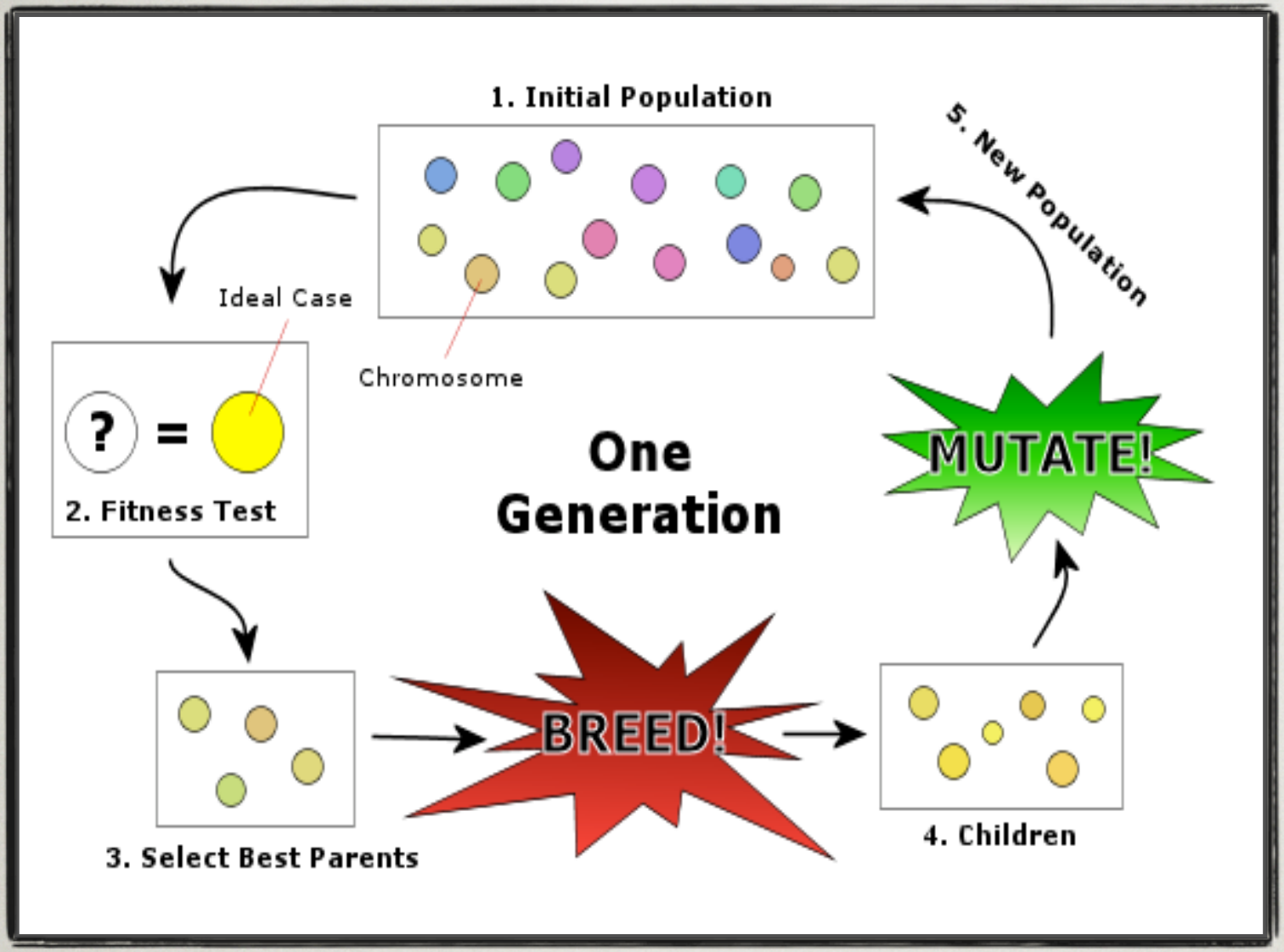


The Problem being Researched ....

Reverse tessellation and conversion from explicit geometry representations like spline surface boundary models to an implicit geometry model is a hard shape detection and geometry reconstruction problem with no guarantee of a reasonable solution.

The Beset Approach

- Implemented a genetic algorithm (GA) using an encoded form of a CSG hierarchy
- Chromosomes (aka genomes) consists of primitives and boolean operators
- Developed a fitness function that utilized BRL-CAD raytracing to evaluate and compare spatial occupancy of candidate solutions

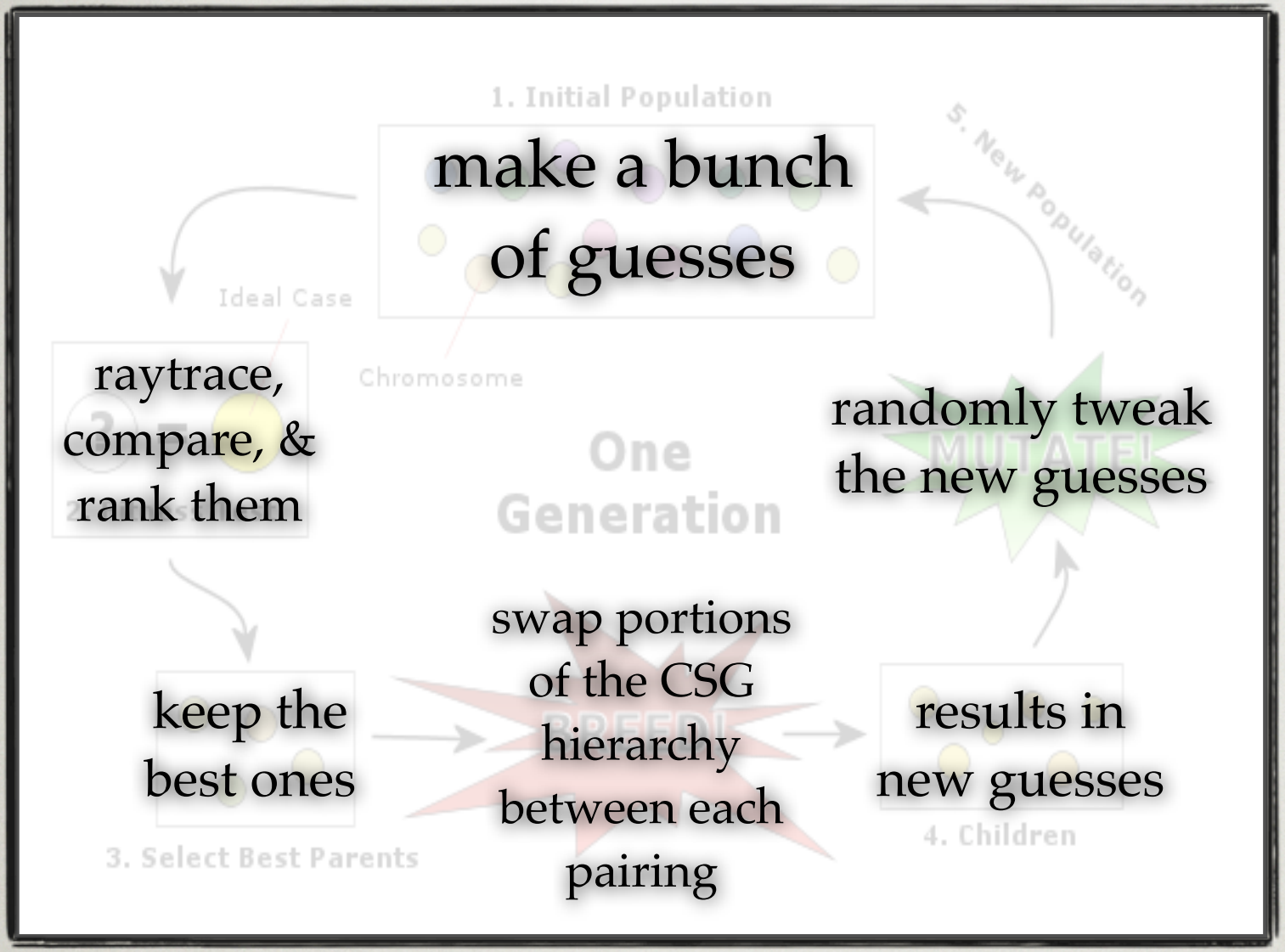


Genetic algorithms can have difficulty converging on an optimal solution and can require vast amounts of computing power. Most times they are able to converge on a near-optimal solution, however if the parameters of the genetic algorithm (such as mutation rate, crossover rate, population size) are not properly set, a near-optimal solution may never be reached. The GA stagnates on a local minima solution.

Genetic Algorithms

"survival of the fittest"

It's an "evolutionary algorithm" used as a computation method for finding solutions to optimization and search problems using global search heuristics.



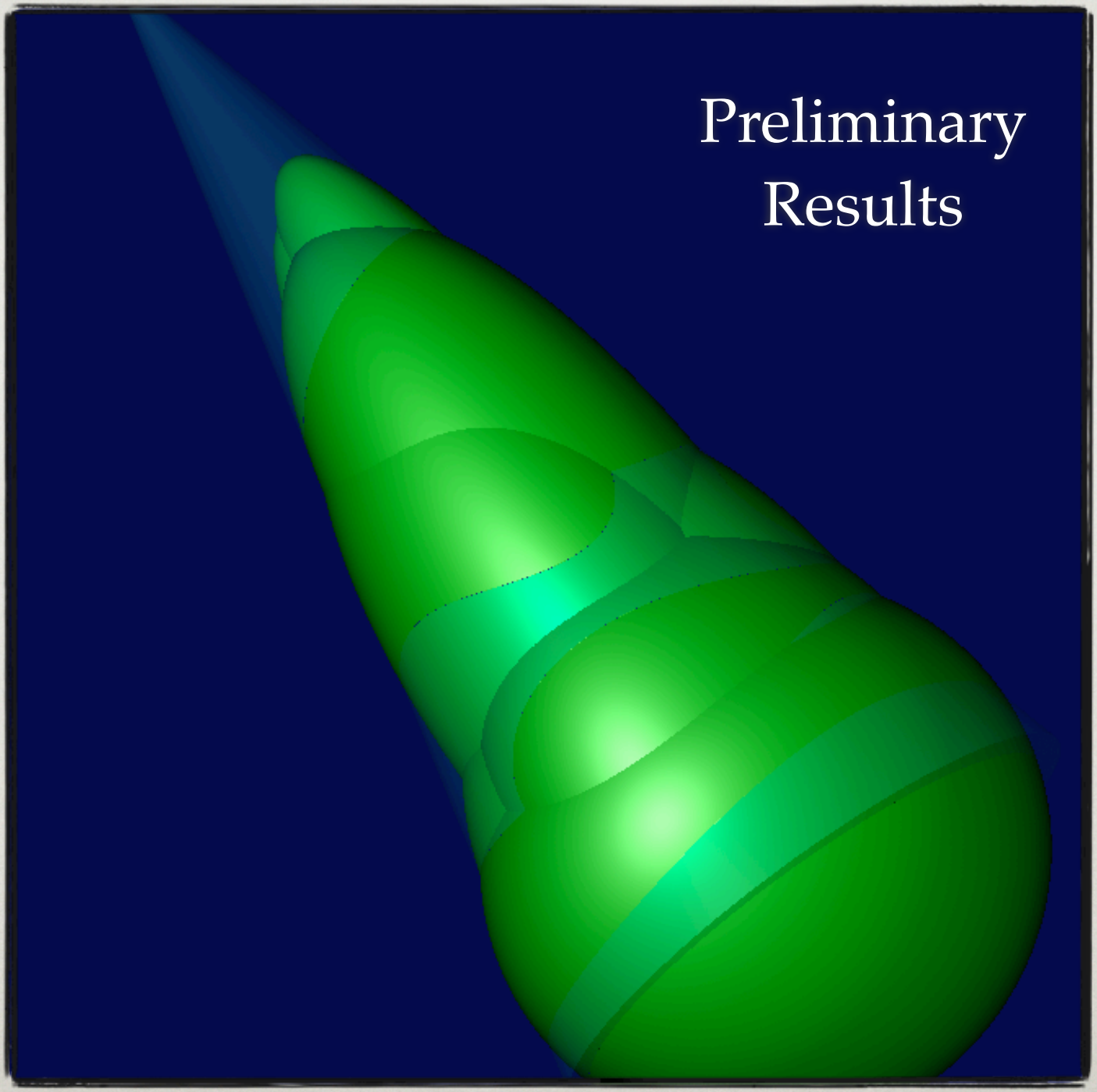
Genetic Algorithms

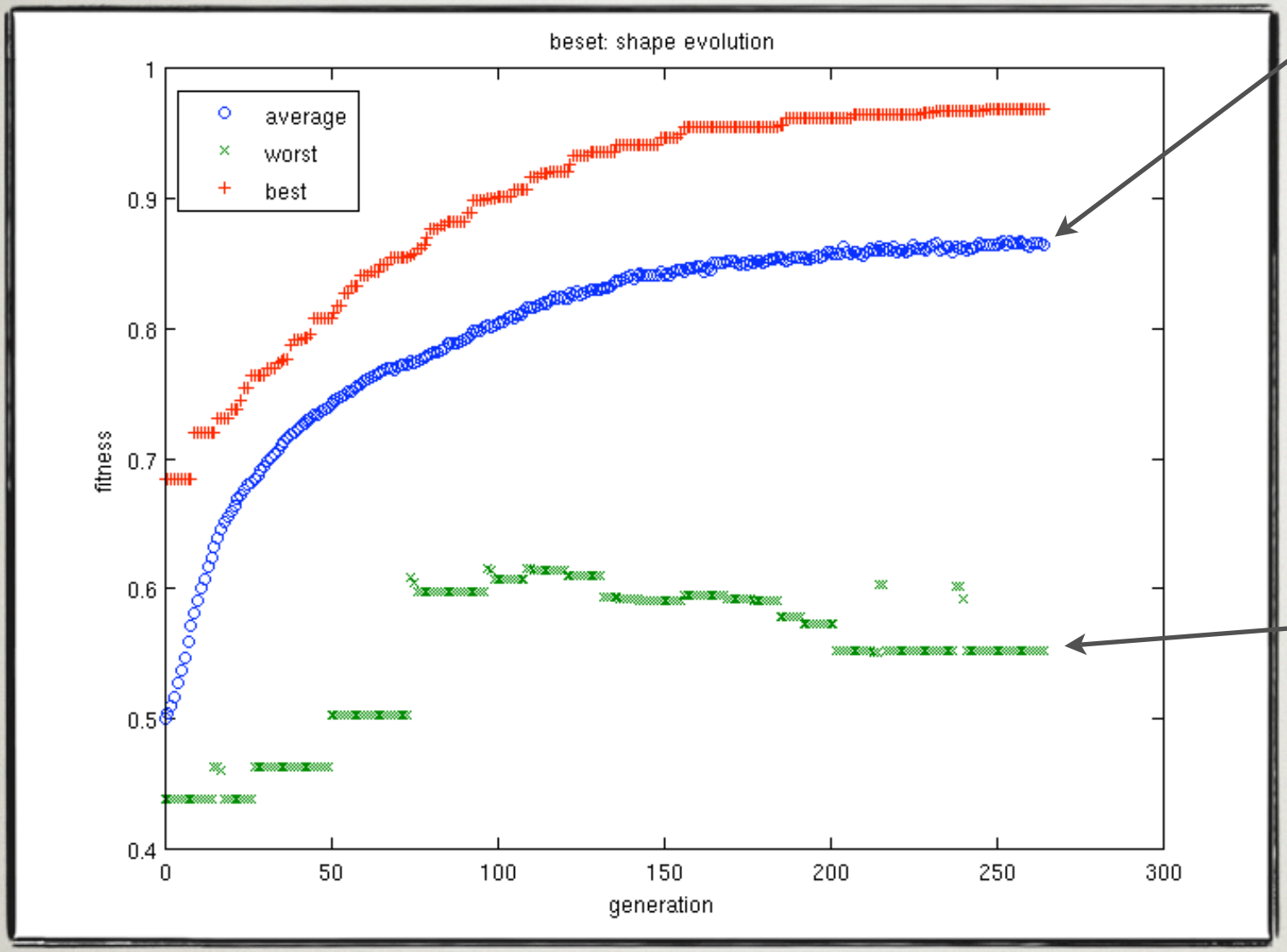
"survival of the fittest"

Applied to geometry and by using raytracing to evaluate a fitness function, we can drive the GA towards an adaptive solution to a given (arbitrary) input geometry.

Beset's Implementation

- Beset trees are made up of **primitive shapes** *(currently ellipsoids, arbs, and conics)* and **CSG boolean operators** *(union, intersection, and difference)*
- A random population is created where the bounding box of a new primitive being added intersects the bounding box of the input data
- Each candidate is raytraced and segments are compared
- Each individual is then evaluated based on their closeness to the input and this value is normalized between 0 and 1 *(this is the fitness measure)*
- The most fit individuals are selected to be passed on to the next generation, and certain genetic operations will be performed on them *(elite preservation, elimination, crossover, mutation, etc)*


Preliminary
Results



Exponential tapering off of the average fitness is expected but stagnates too rapidly here indicating that the population size probably needs to be considerably larger

Decreasing or stagnant worst case fitness is an indicator that the fitness function would probably benefit from additional metrics

THE GA SEARCHES FOR A MATCHING SHAPE

Input geometry was a simple truncated general cone but the GA was only allowed to construct guesses using ellipsoids and union CSG operators.

Results

- Needs additional research on allowing a more diverse sample set (e.g, multiple primitives & boolean operators)
- Need larger population sizes and improved fitness function metrics in order to evolve efficient solution candidates.
- Algorithm is probably insufficient by itself for matching low-probability features without fitness specialization (simulated annealing would probably help)
- There is considerable merit to the overall approach as a viable shape extraction method that it warrants further study