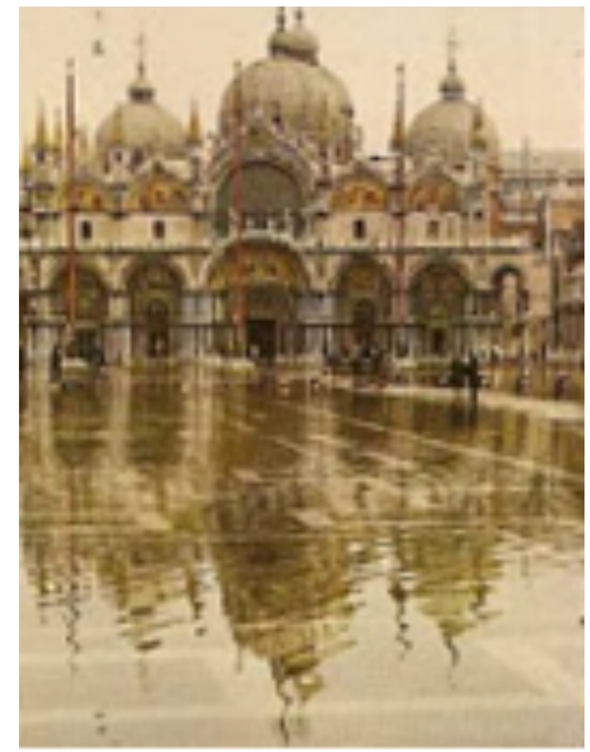
Pin-up #1 for jury review. 7/18/18, Driscoll PhD dissertation

Fractal dimension as objective function in genetic algorithms for application in architectural design

John Driscoll B.Arch, MS Systems Science
PhD candidate
Portland State University

Dissertation Committee:
Wayne Wakeland, Chair
Joe Fusion, Co-chair
Corey Griffin
Antonie Jetter, GO Rep



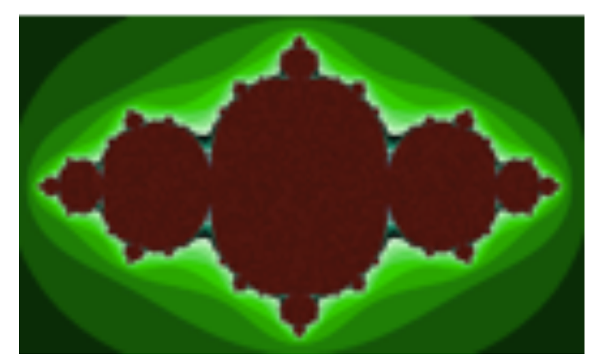
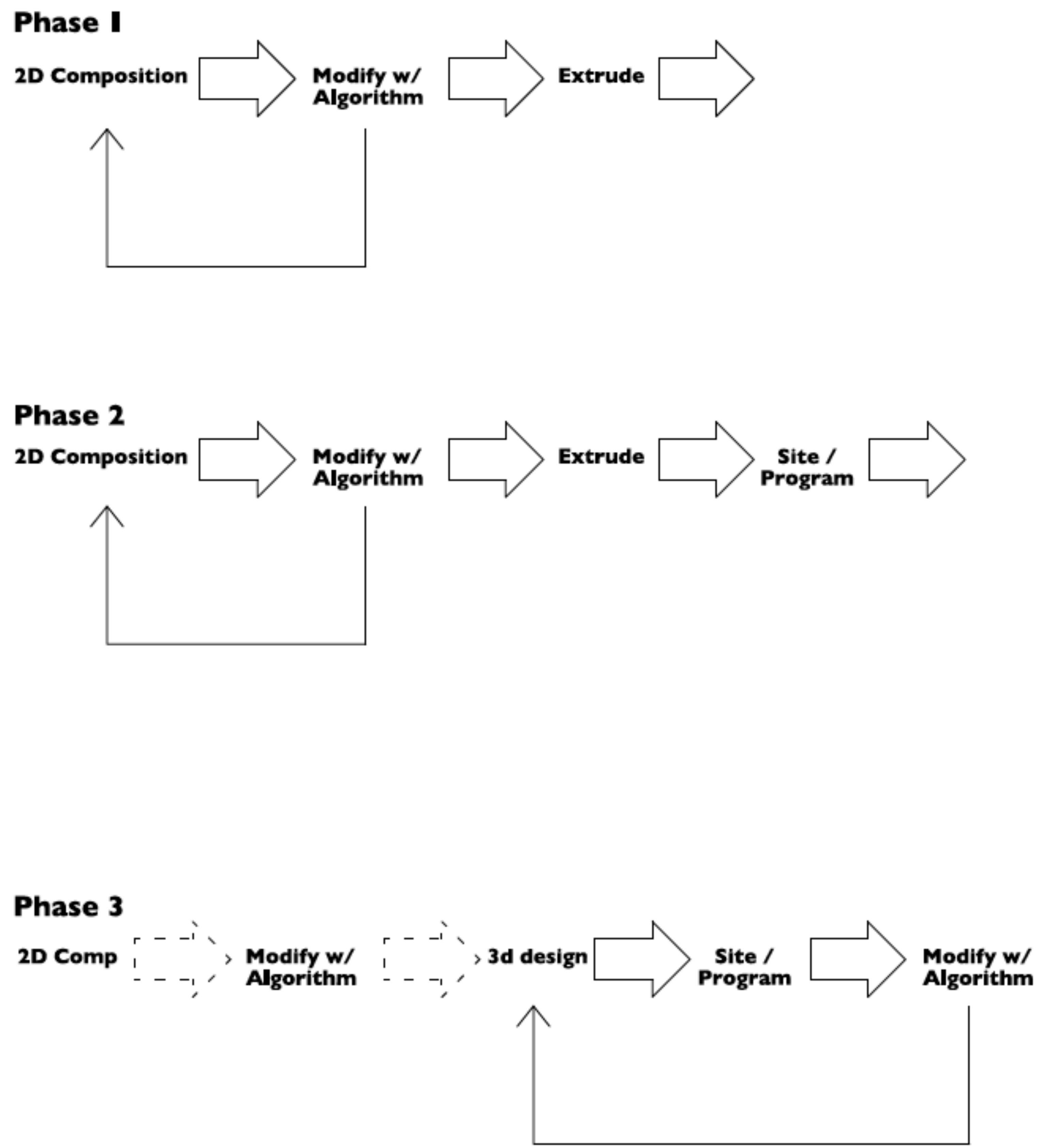


FIG 1. San Marcos Church in Venice reflected. Below: San Marcos Julia fractal, J(-3/4, 0).

driscoll.john92@gmail.com

Gallery: Pin Up #1 (Phase 1)

This review is the first of 3 reviews or project *phases* which are shown in the diagrams to the right. For now, the algorithm we are coding only creates designs based on parametric inputs and running a genetic algorithm [GA] to select exemplars with the highest fractal dimension [FD]. It is not necessary to be familiar with GAs or FD to participate in reviewing this project.



The jury is being asked to assess the following images as if they were part of a traditional pin-up in an academic setting. Please consider this review to be a design exercise where the designs represent compositions of lines and masses. The exercise is based on Dean Bryant Vollendorf’s [DBV] ’3 lines on a page’ method for teaching basic elements of composition such as: focal point, secondary focal point, counter point, rhythm, terminal, rectangular motion of the eye, etc. For more information on DBV’s method as well as more: gallery images and background materials and the Python code, use this link to access the DropBox folder titled, Pin-ups:

https://www.dropbox.com/sh/iqiv84hn9exocom/AABDBG2w4D24R93TR-P8aOIla?dl=0

driscoll.john92@gmail.com Portland State University

Case study project: Green Street Garage Renovation

https://www.cityofithaca.org/DocumentCenter/View/7614/Green-St-Garage-Redevelopment-IURA-RFP-Revised-5718

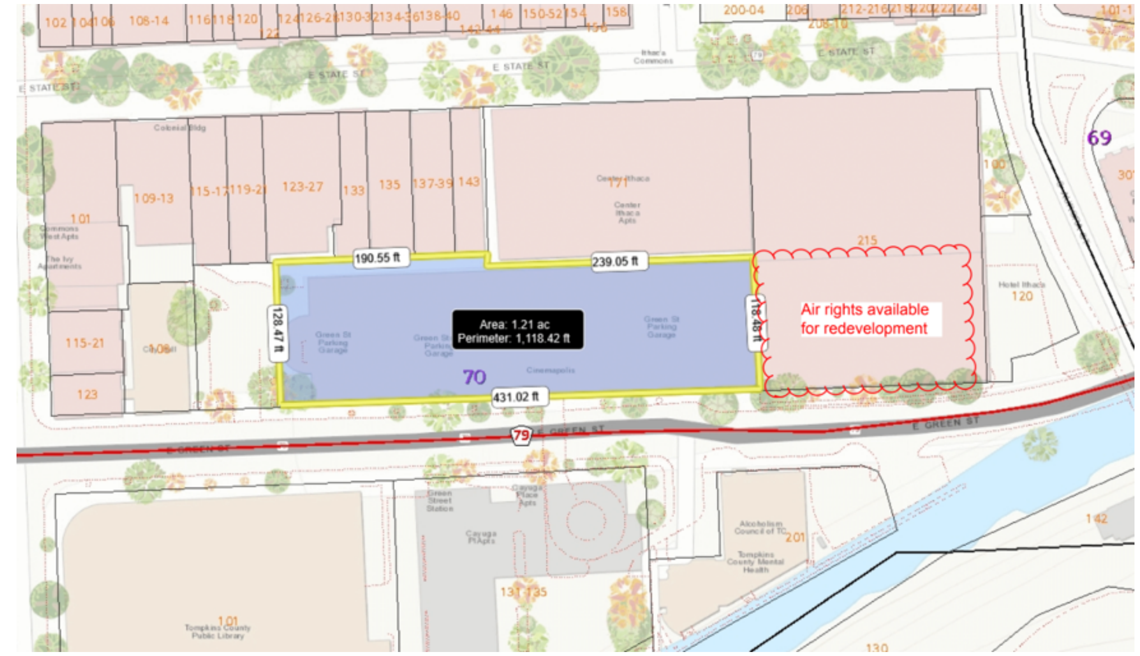


FIG 3. This page outlines the case study project that will be added to the design process in phase 2. More details will be provided for the project and its program in pin-up #2 . The project is for the Green Street parking garage renovation in downtown Ithaca, NY that the author is currently developing a proposal for in collaboration with others. The site is roughly 24,000 sq. ft. of surface area. We will be proposing 3 to 4 new levels of parking deck and a new facade across the south elevation as well as a new marquee for the movie theater (Cinemapolis) currently occupying the first floor floor level.

driscoll.john92@gmail.com Portland State University

Sample of 2D design outputs

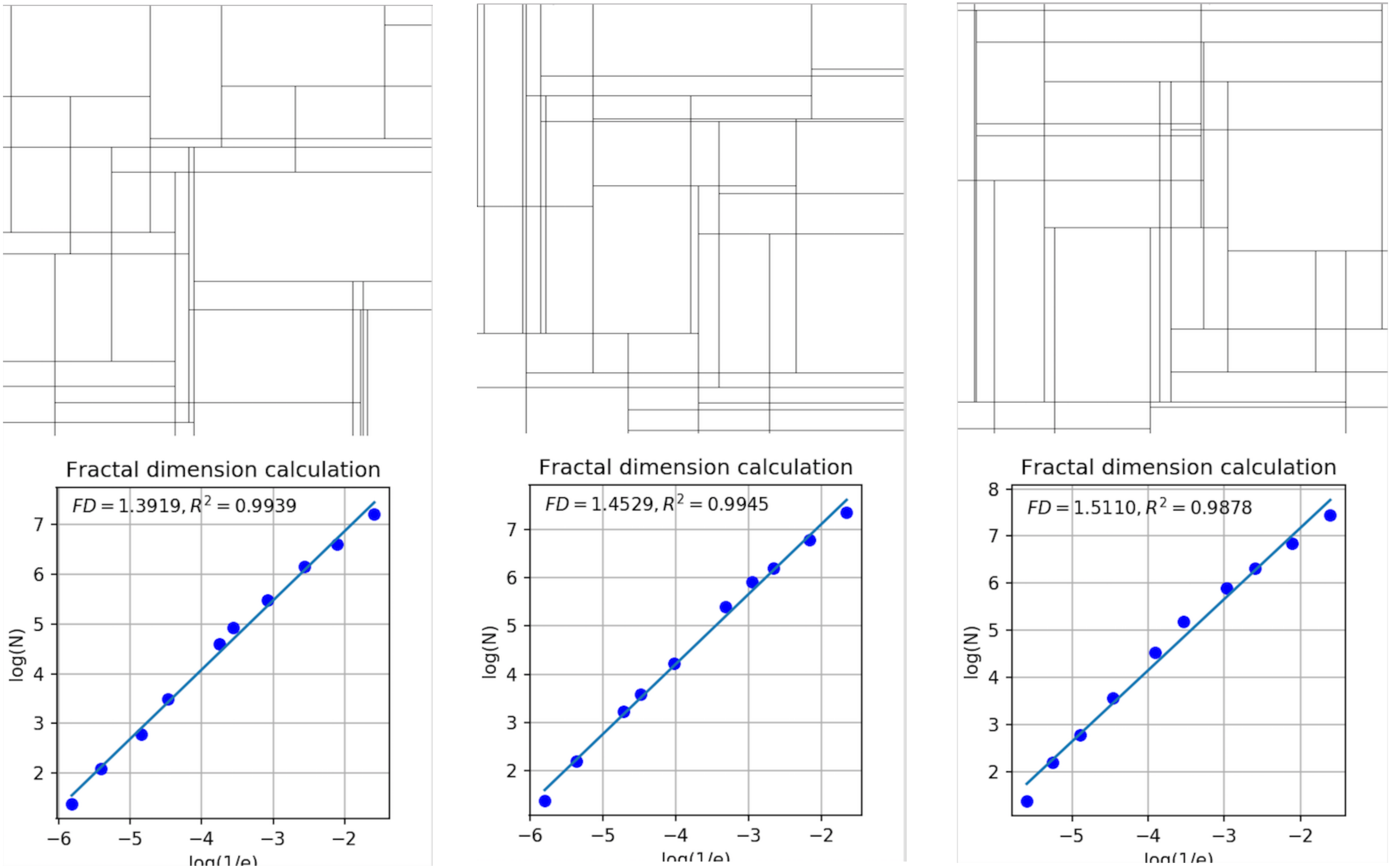


FIG 4. The images above show basic line compositions generated randomly with respect to a set of adjustable parameters such as the number of vertical and horizontal lines and a range for how many times they cross. These are (3) 30 line compositions with FD determined using box-counting dimension and shown in the graphs below.

driscoll.john92@gmail.com Portland State University

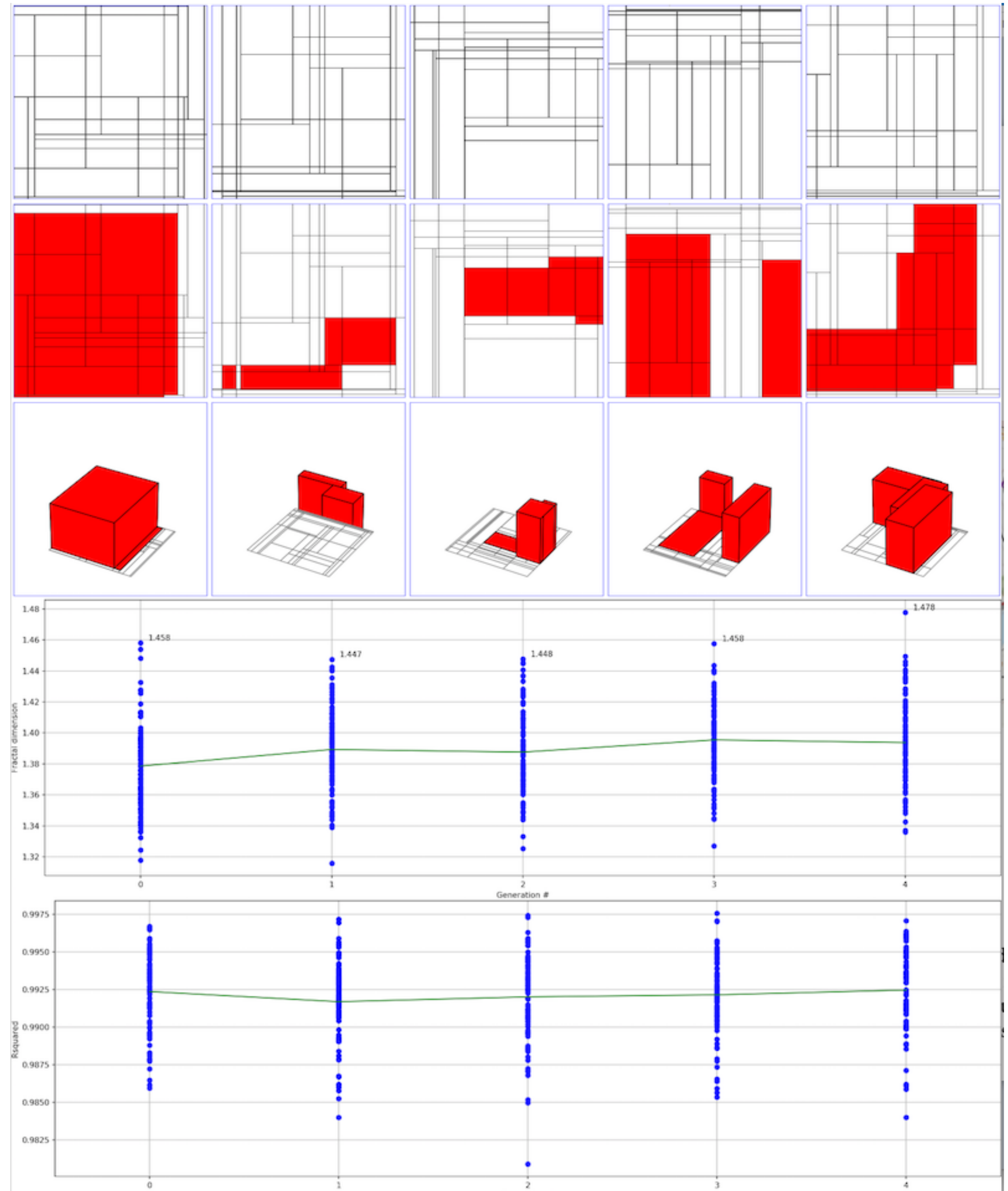
Timeline

FIG 5. The image to the right shows a timeline with the older designs to the left and the newer ones to the right. The 2D compositions are across the top. These are created randomly with a defined number of horizontal and vertical lines that are allowed to cross within a parametric range. Populations of designs breed and randomly mutate for a number of generations (5 in this case). Exemplars are selected based on the highest *fractal dimension* [FD] in the population. Over time, the FD generally increases.

The next set of images down show the exemplar composition but now with a number of rectangles or *masses* chosen at random. Masses are allowed to overlap and any number of masses may be set in the corresponding parameter.

The next set of images show the masses extruded into the third dimension to a random height within a parametrized threshold.

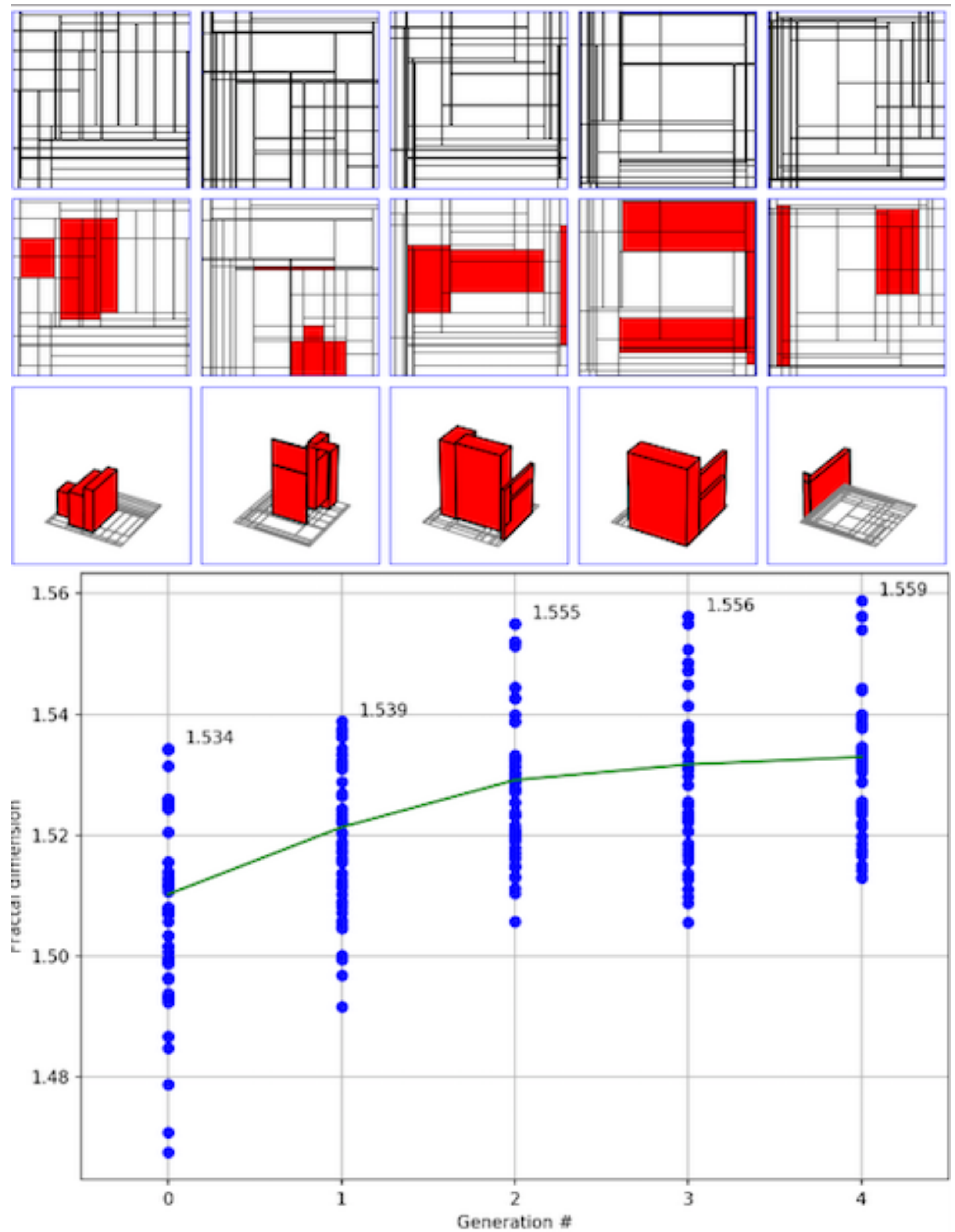
The graphs at the bottom display the FD and the r^2 of the box-counting dimension. Box-counting dimension is a technique for determining FD when an object is not regular. The FD measures only the 2D designs at this stage (phase 1). Subsequent phases will measure the 3D model for FD as well.



driscoll.john92@gmail.com
Portland State University

Timeline

FIG 6. This image again shows a timeline of five generations of designs with the exemplar design for each generation displayed at the top. FD for exemplars ranges from 1.534 to 1.559.

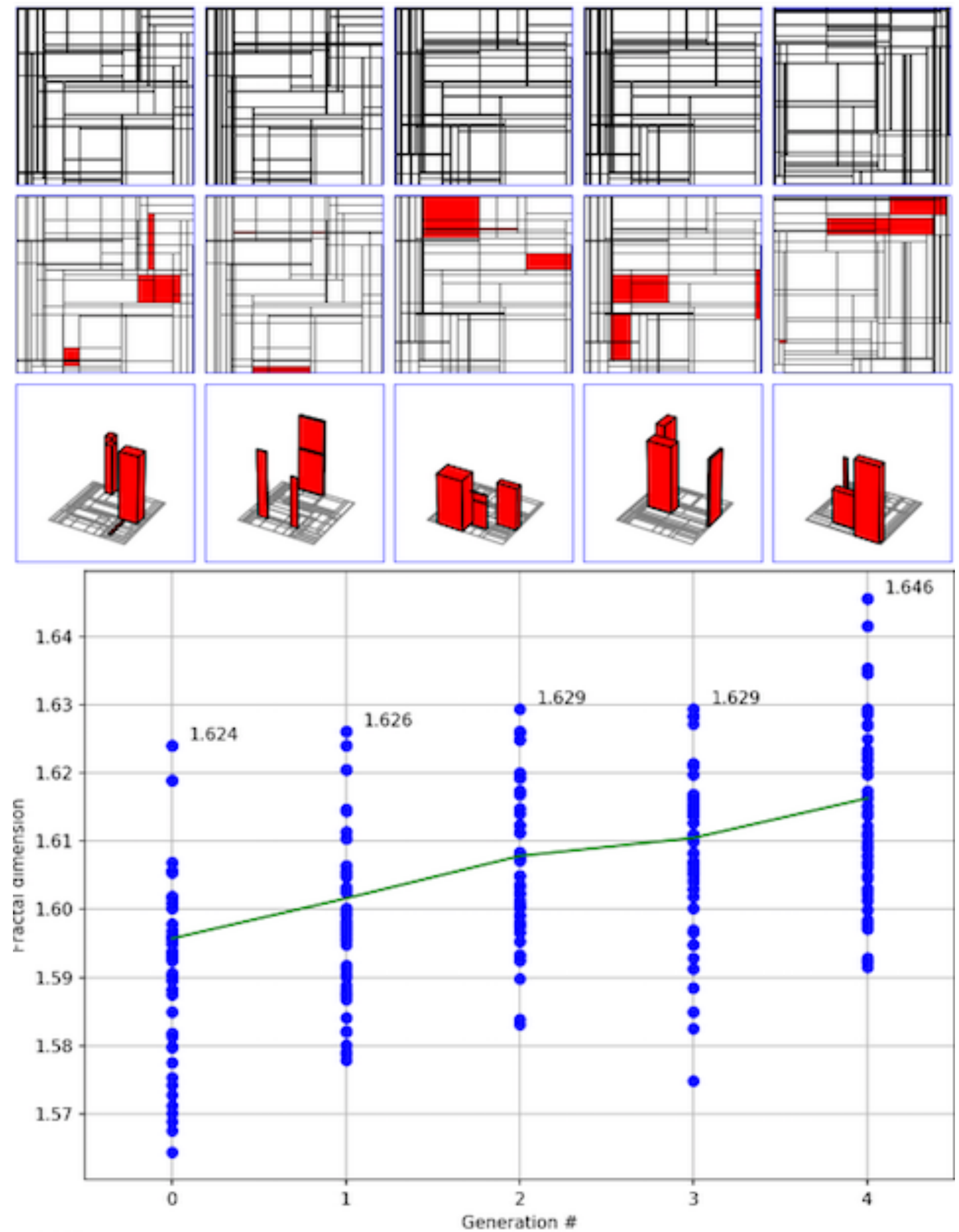


driscoll.john92@gmail.com

Portland State University

Timeline

FIG 7. This image again shows a timeline of five generations of designs with the exemplar design for each generation displayed at the top. FD for exemplars ranges from 1.624 to 1.646.



driscoll.john92@gmail.com
Portland State University

Timeline

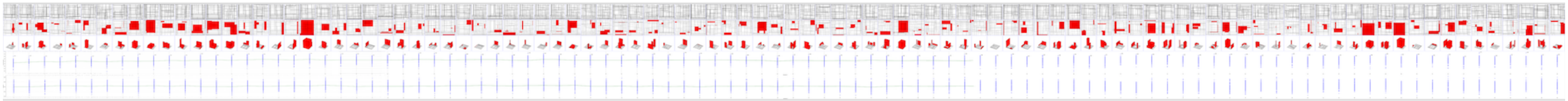
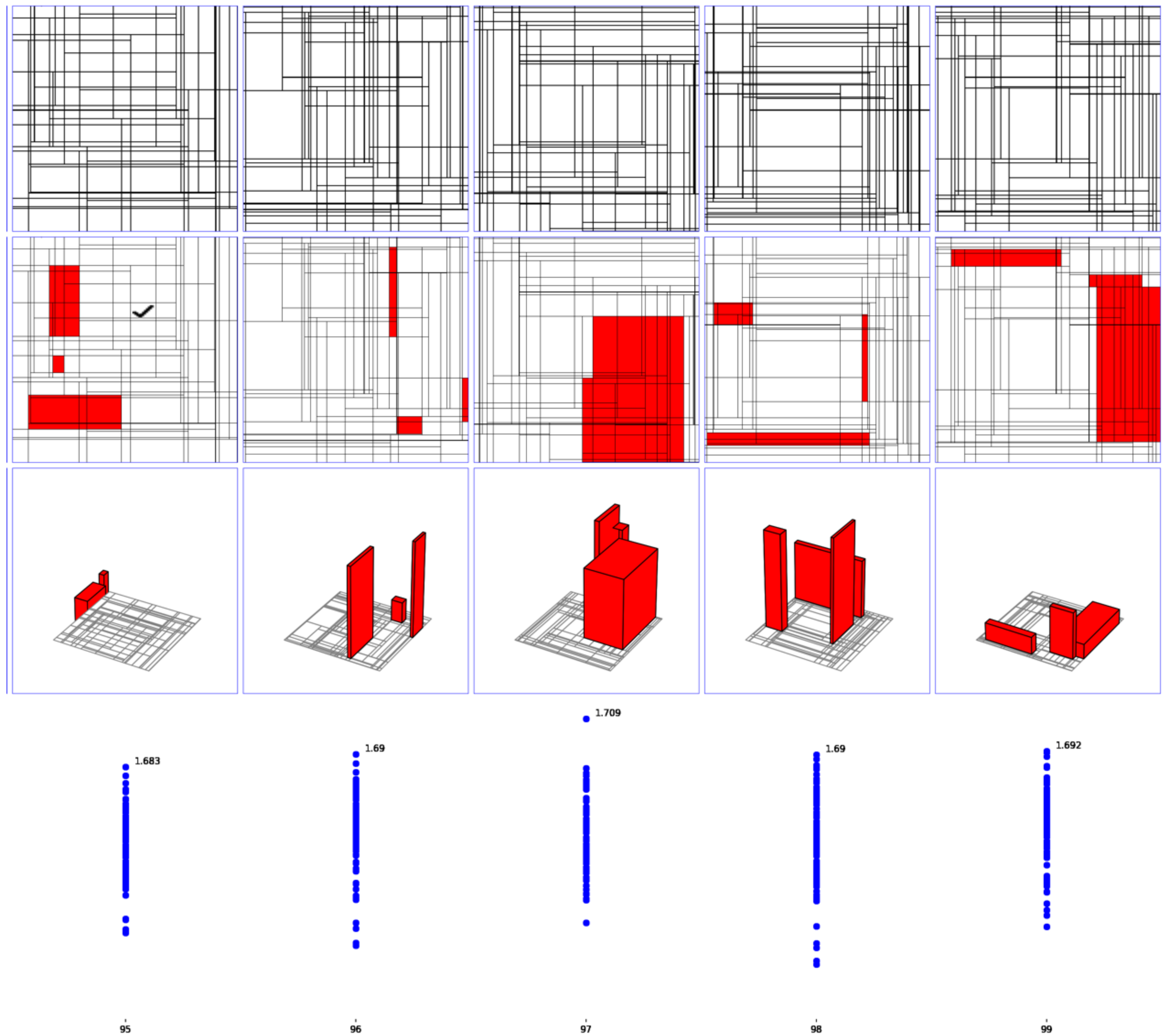


FIG 8. The image above shows a long run of 100 generations for a design using 50 lines. The last five generations are shown to the right. The FD for exemplars ranges from 1.683 to 1.709.

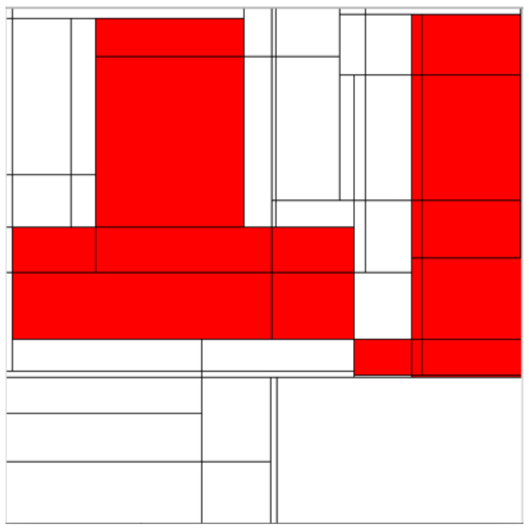


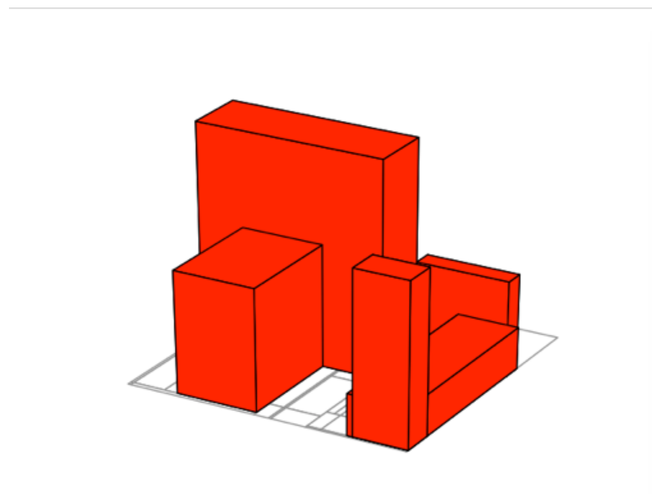
driscoll.john92@gmail.com

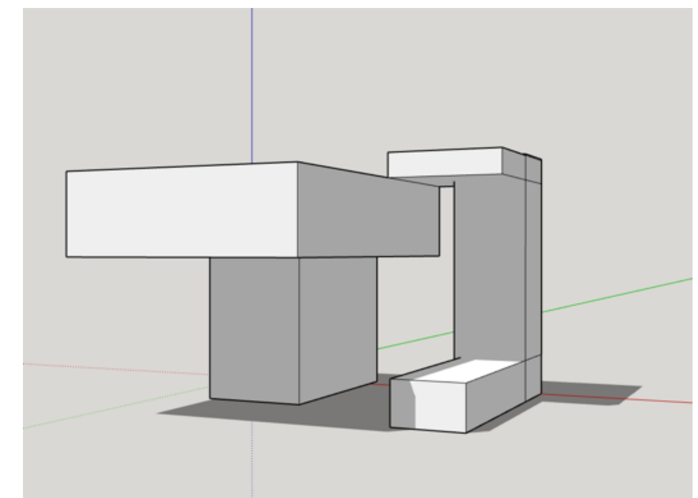
Portland State University

From computer model to architectural model

FIG 9. This image illustrates how the composition and extrusion may be incorporated into a traditional CAD environment and be considered in terms of its architectural qualities.







2D composition, 'exemplar' from genetic algorithm with masses added randomly.

Extrusion of masses with random z coordinate.

Export as .Stl file.

Import into Sketchup or other 3D modeler.

driscoll.john92@gmail.com

Portland State University

Example of scale and context added to raw output

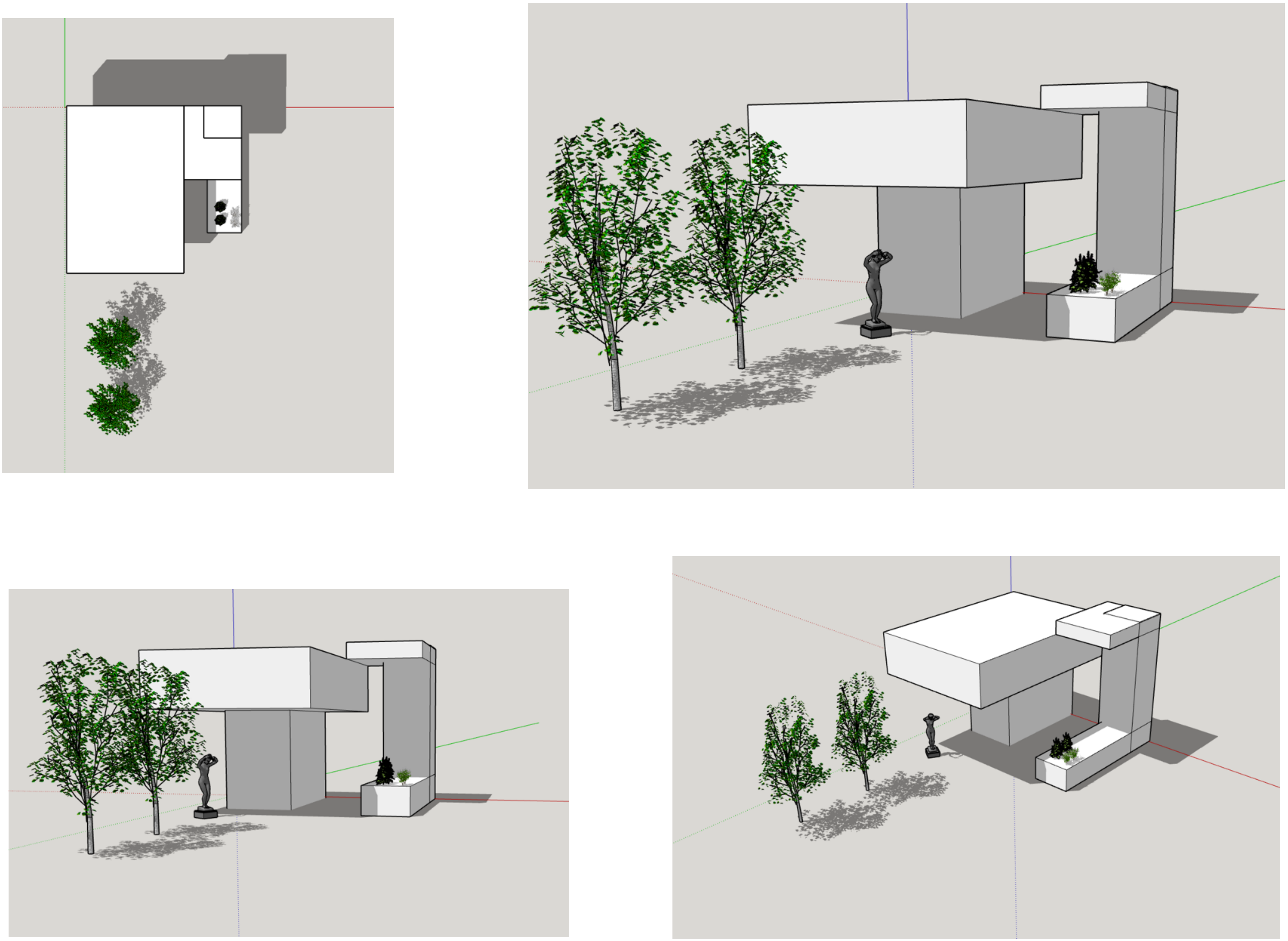


FIG 10. This image illustrates how basic elements and context might be added to the model giving it scale and function.

driscoll.john92@gmail.com

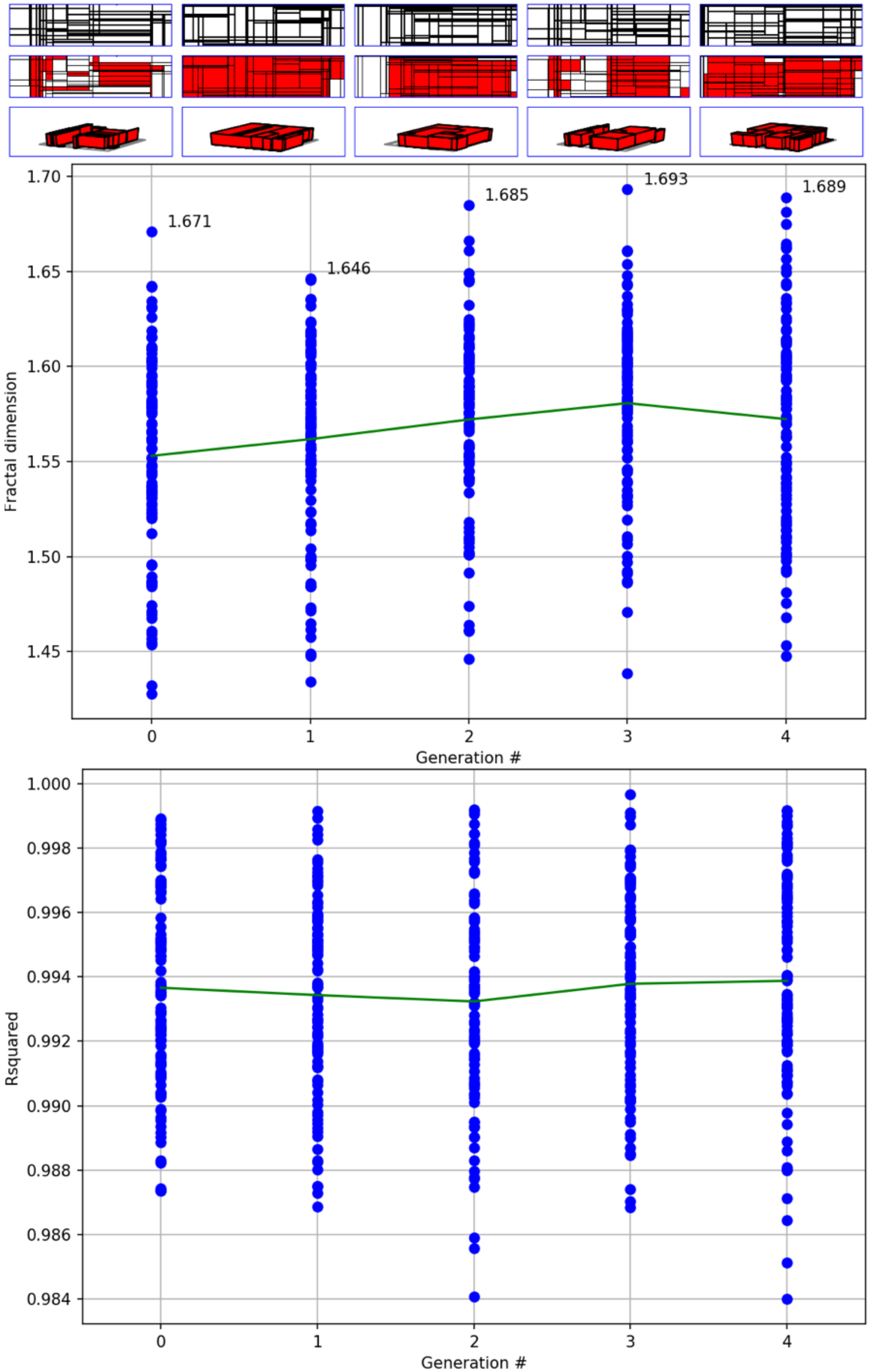
Code parameters

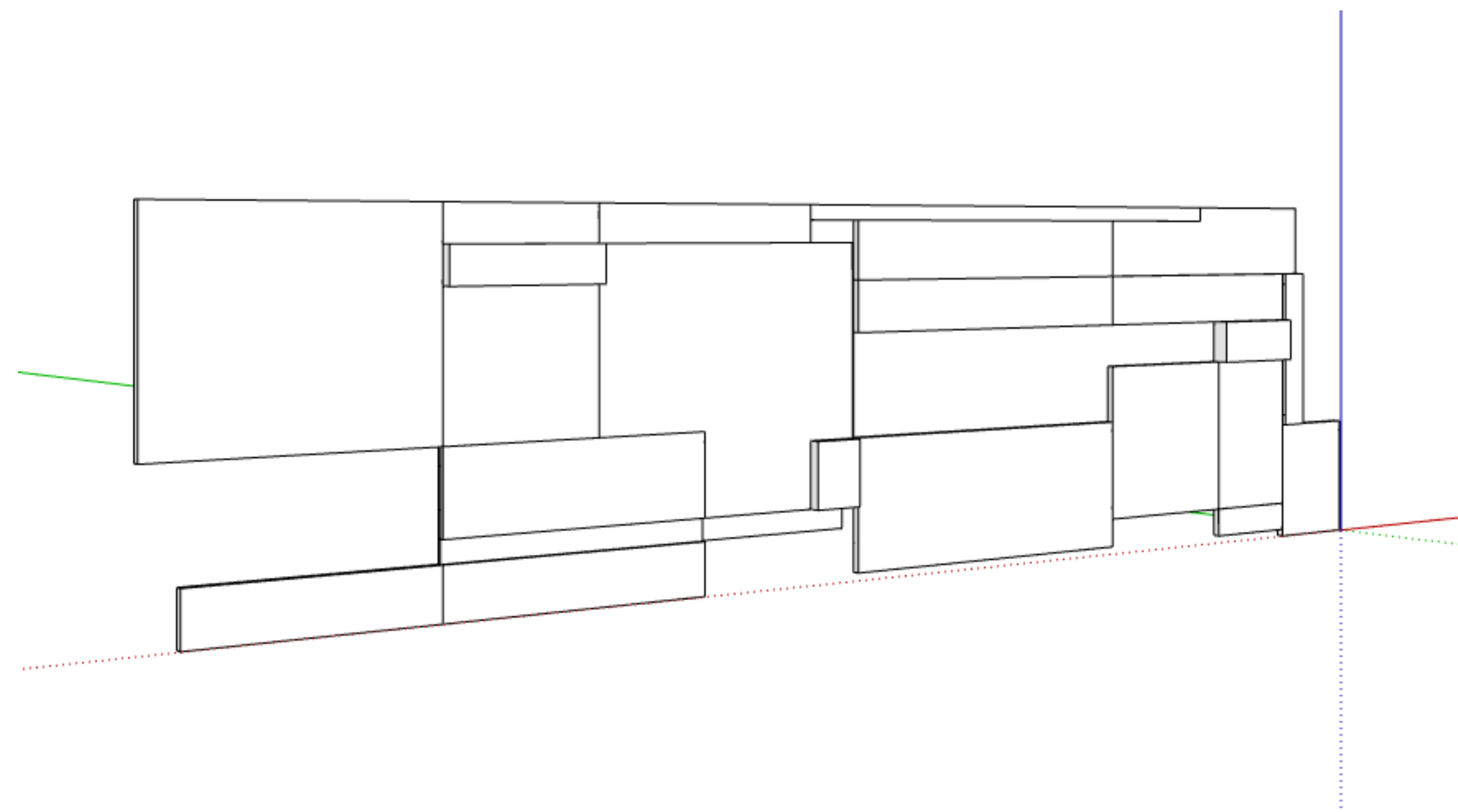
```
testGA(
    individualClass=gArch.design,                 # the object class that will serve as individuals to evolve
    numGenerations=5,                            # Number of generations to run GA
    populationSize=100,                           # Number of individuals in each generation
    fitnessFunction=fitnessFunction,              # The function to use to score each indvidual
    mutationRate=5,                               # Average # of mutations per individual per generation
    crossoverRate=0.1,                            # Average # of crossovers per individual per generation
    tournamentSize=3,                             # # of individuals in each competition group to determine survivors
    saveChampion=True,                            # Save the final champion of the evolution
    showChampion=True,                            # Display the final champion of the evolution
    saveTimeline=True,                            # Save timeline to file
    showTimeline=True,                            # Output timeline view of genetic algorithm results
    saveChampion3D=True,                          # Save the 3D final champion design to an stl file
    showOutputGraphs=False,                       # Show separate matplotlib graphs of FD, R2, and fitness vs. generation
    saveBaseFilename="SHELL_1",    # Filename base to use to save various files from the GA output (do not include a file extension)
    canvasSize=(200, 50),                        # Size of the design canvas.
    canvasHeight=2,                              # The height of the 3D volume that the design masses get extruded into.
    N_Lines=40,                                    # Number of lines with which to initialize the designs
    N_FD_Points=5,                                # Number of box scales to use in FD calculation
    maxCrossings=5,                              # Maximum # of crossings for lines in designs
    randomizeMaxCrossings=True,                   # Whether or not to randomize crossing # of lines between 2 and maxCrossings
    permanentEndpoints=[],                        # List of pairs of endpoints representing arbitrary permanent lines on the design.
    addProb=0.3,                                  # Probability that mutation will add a line
    changeProb=0.4,                               # Probability that mutation will change a line
    removeProb=0.3,                               # Probability that mutation will remove a line
    conserveLineNumber=True,                      # During mutation, should the # of lines be conserved?
    equalHorizontalVertical=True,                 # when adding/removing lines, should the design keep equal # of horiz. & vert. lines?
    pctBoxCountScaleJitter=0.1,                   # Maximum percent random deviation of box scales from a perfect logarithmic series. Default = 0.15
    verbose=False,                                # Show more or less debugging info?
    innerBorderThickness=1,                       # Inner border thickness for design images in timeline
    innerBorderColor='blue',                      # Inner border color for design images in timeline
    outerBorderThickness=6,                       # Outer border thickness for design images in timeline
    outerBorderColor='white',                     # Outer border color for design images in timeline
    rectangleSamplingMethod='sampleAllRectangles',     # How should rectangles be sampled when choosing masses?
                                                  #   'dartboard' = pick a point at random, whichever rectangle it lands in is chosen
                                                  #   'sampleMinimalRectangles' = identify all rectangles that do not overlap any lines, choose one at random
                                                  #   'sampleAllRectangles' = identify all rectangles, overlapping or otherwise, choose from one at random
    N_Masses=20,                                   # Number of masses to randomly select
    massColor='red',                              # Color of the selected masses in output image
    massLineColor='black',                        # Color of the design lines in output image
    massLineWidth=1,                              # Width of the design lines in output image
    massBaseOnGround=True                       # If true, all 3D masses have a base on the ground (at z=0). If false, the bottom height is random
    )
```

FIG 11. Here are the parameters for the model which can be set by the designer (or created by the designer) and control the algorithm.

driscoll.john92@gmail.com Portland State University

Shells





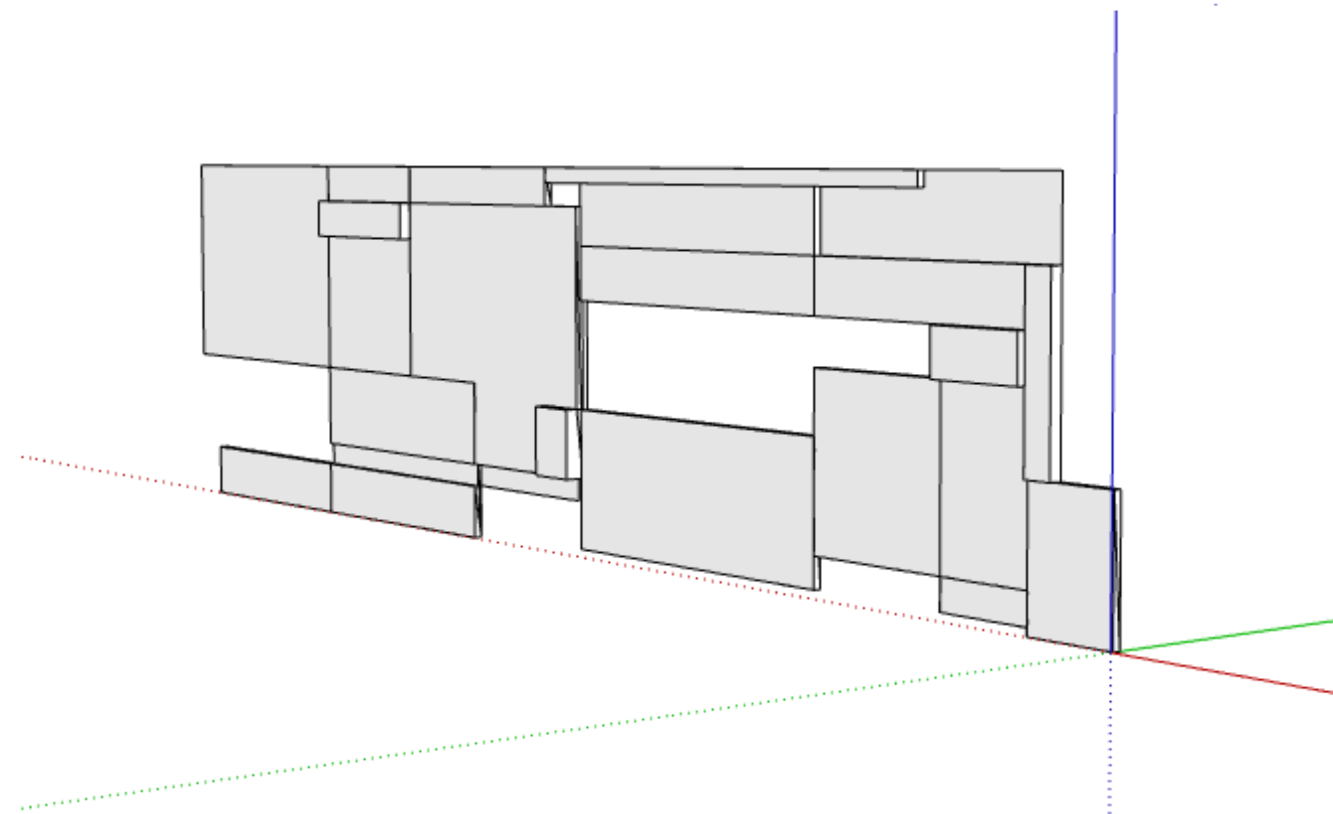
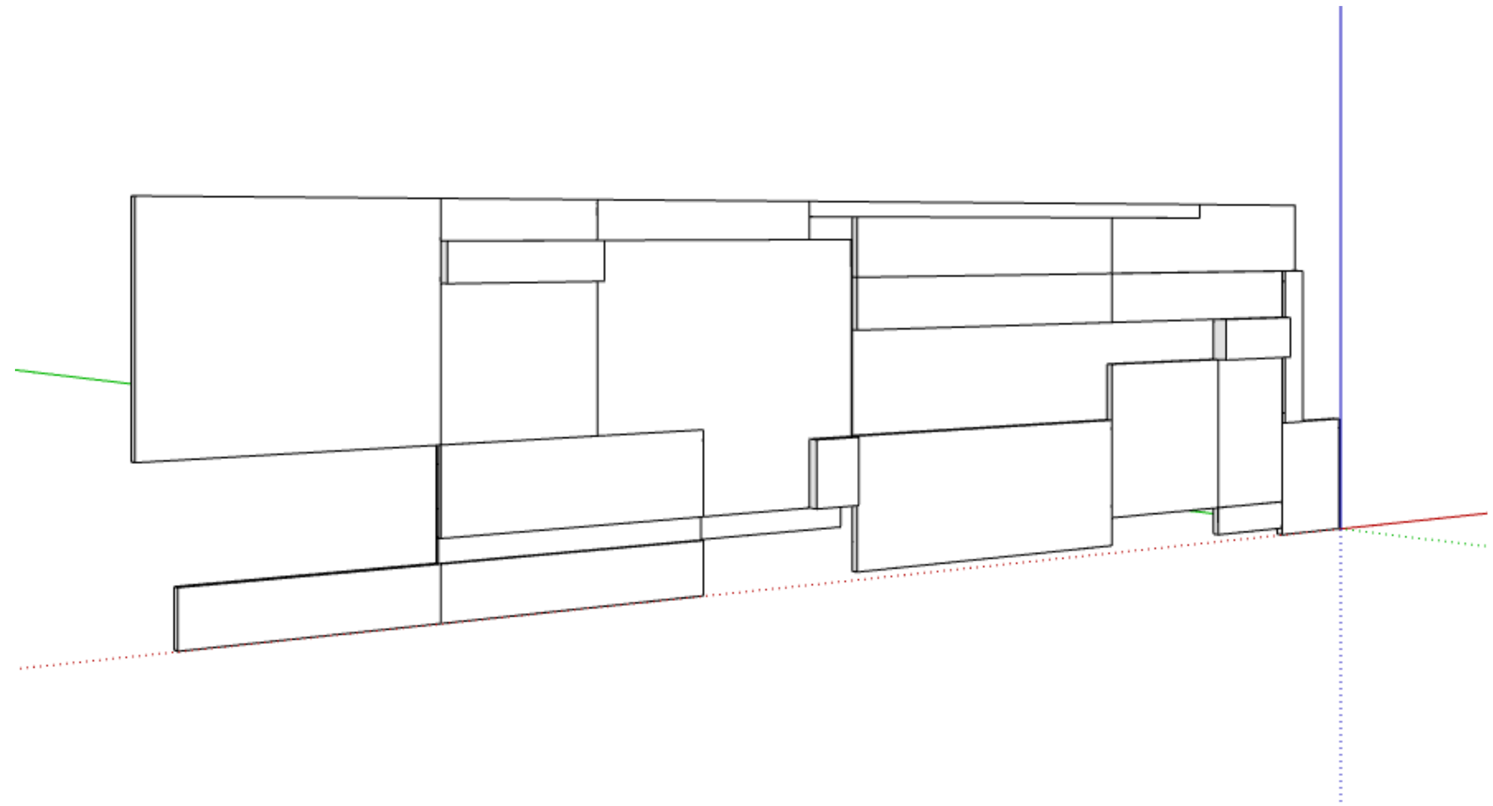
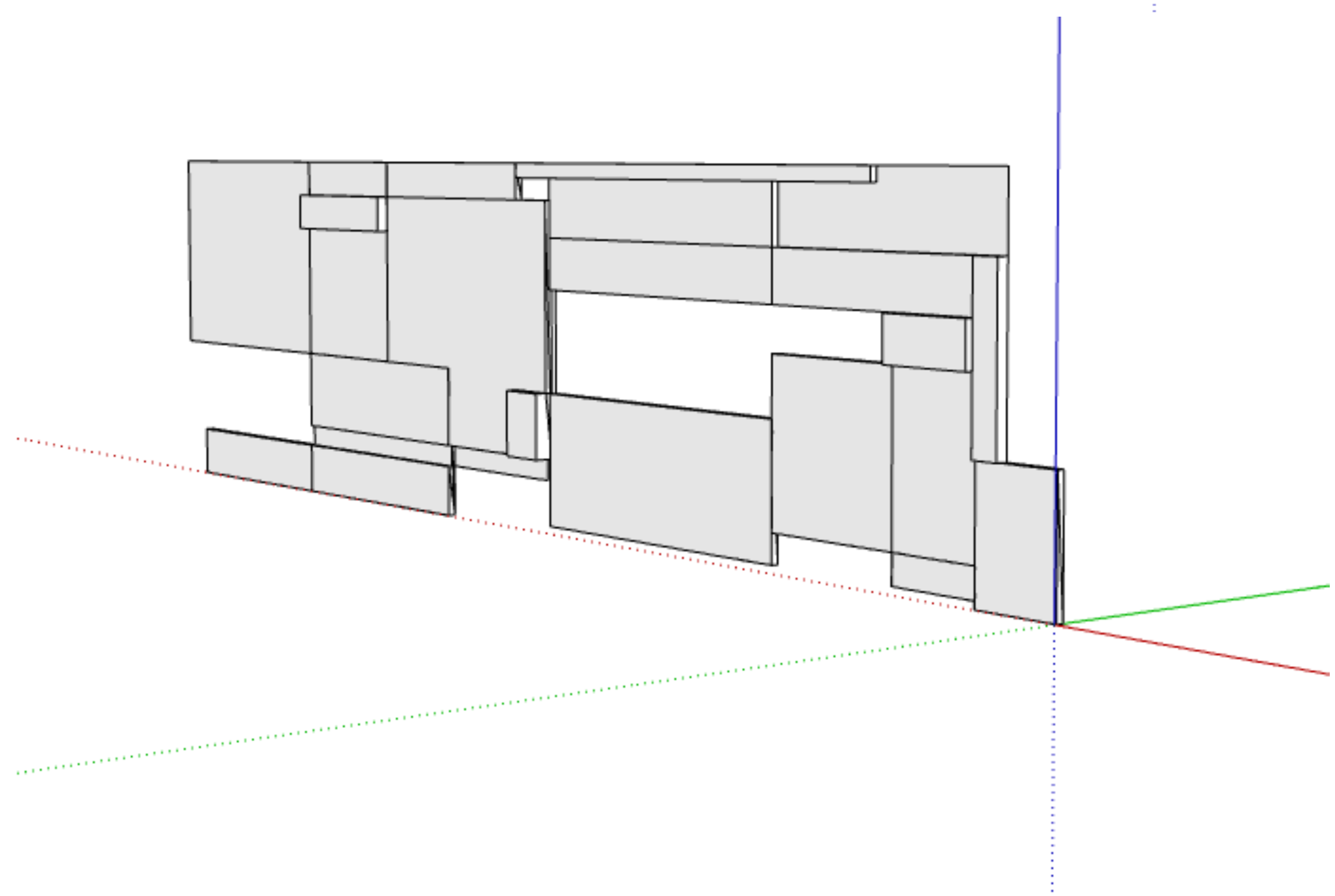


FIG 12. This page starts to play with the parameters to create different types of outputs. Here the canvas size and canvas height were modified.

Shells

FIG 13. These images represent modifications of the parameters shown in Fig. 11 to create *shells* which are not very thick and can cover a large area. The intention is to eventually use the iterative capacity of the algorithm to create many variations of designs that can be assessed both qualitatively as well as quantitatively using fractal dimension and/or other quantitative methods. FD is focused on here stemming from studies that have shown a relationship between multi-scale complexity as measured with FD and natural forms such as landscapes and urban environments.





driscoll.john92@gmail.com

Green Street Garage Project (algorithmic design applied)

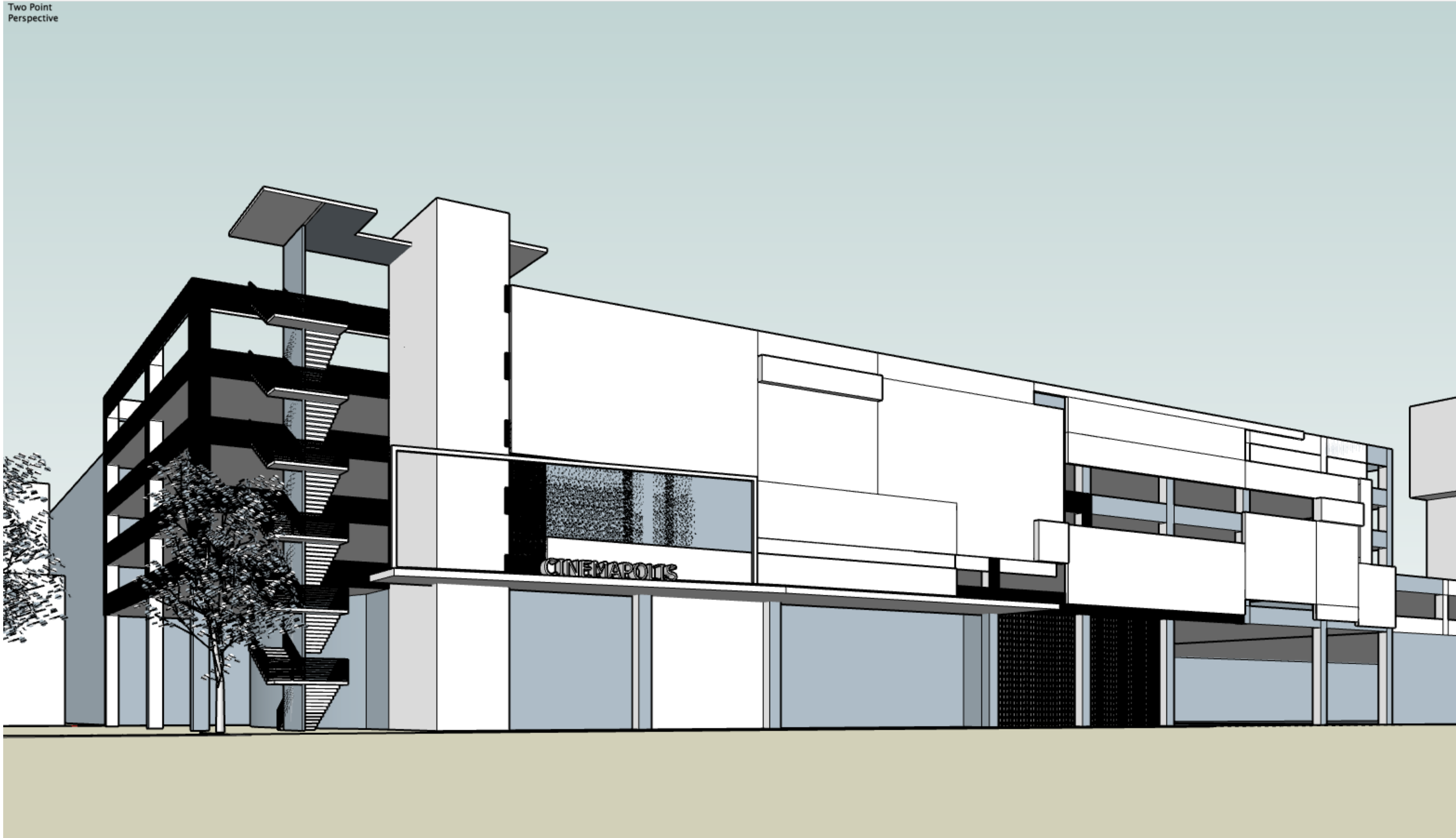


FIG 14. This image represents how the algorithm may be *fit* in a fast and loose way to a specific project such as the Green Street Garage project in downtown Ithaca, NY. The next phase of this project, Pin-up #2, will explore this potential application in more detail, including: context, scale and program. The algorithm will also be modified in phase 3 to allow the designer to make alterations in the form and run this new model through the algorithm, closing the loop so to speak. This feedback between designer and algorithm may ultimately happen as many times as is necessary to converge on an acceptable solution.

driscoll.john92@gmail.com Portland State University

Thank You!

driscoll.john92@gmail.com