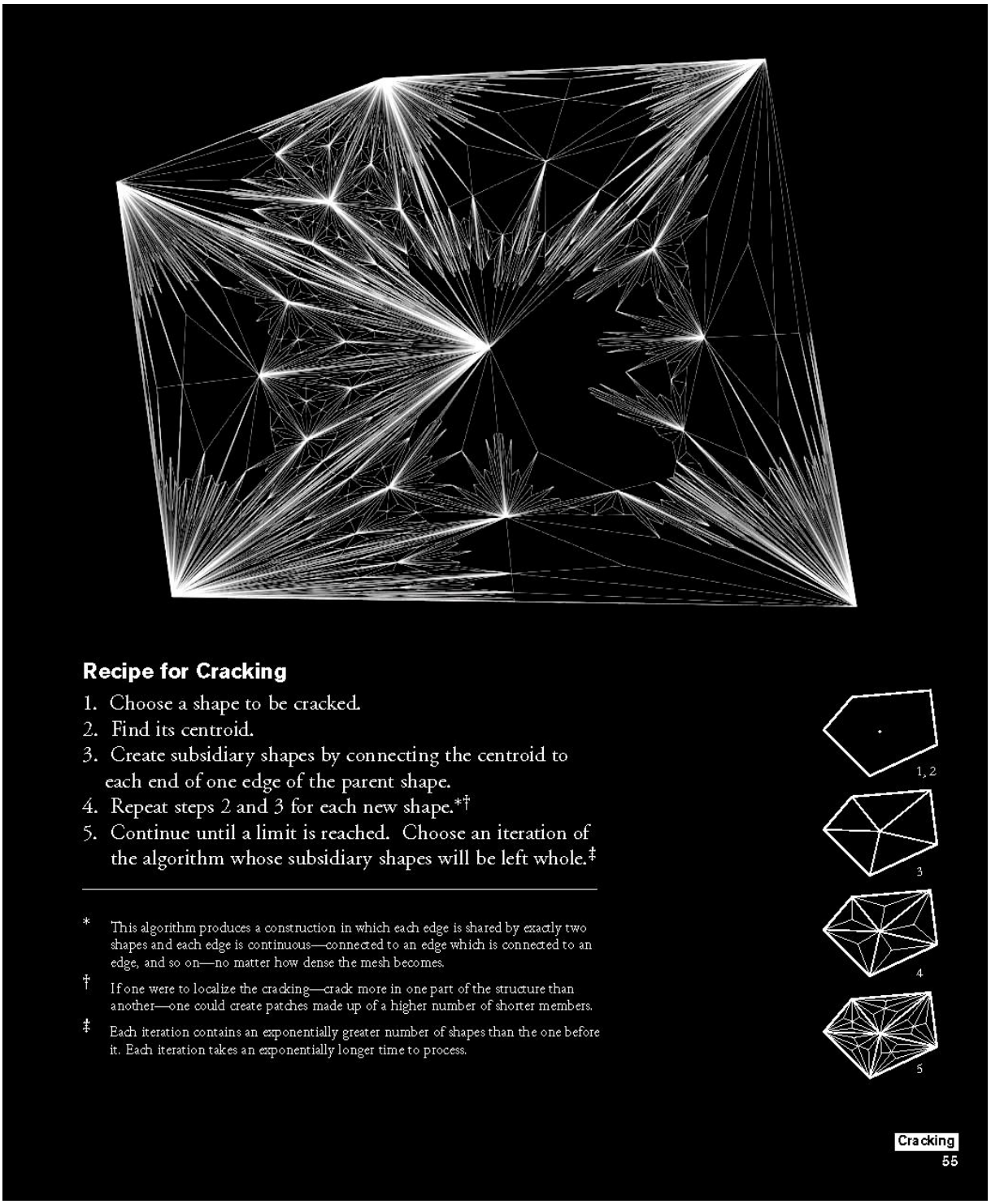
Assignment 6: Grasshopper 1 - Visual Coding and Algorithmic Manipulation



"*The term algorithm simply means a series of steps. Today, as modeling, representation and fabrication technologies shift from manual to automated process, this issue of algorithm is pressing precisely because it confronts the design of procedures themselves*." Aranda/Lasch (*Pamphlet Architecture 27: Tooling)*

Shapes are inherently driven by internal geometric properties; by manipulating these properties accessed through the algorithm, the architect is able to exploit certain formal predilections, allowing the liquefaction and manipulation of geometry.

The biggest challenge architecture faces when dealing with algorithms lies in the necessity to establish clear, concise and coherent rules that can be used to manipulate geometry. It is only once these rules are established that the task of designing can begin.

This assignment asks students to engage algorithms from both graphic and computational perspectives. By producing a graphic recipe that reduces complex geometry to a collection of *points, planes and vectors,* students will develop a diagrammatic pseudo-code. This pseudo-code will in turn be used to build a parameterized Grasshopper Component that will dynamically populate a fluid surface.

This assignment is split into two parts, with *Part A* being completed prior to Wednesday's tutorial. *Part B* will be handed out in tutorial and will have a final due date of 11:59 pm Sunday October 29, 2017.

Part A: Component Development

1. Design a graphic pseudo code for the production of a parametric component that can be manipulated through a minimum of three parameters. The pseudo code will illustrate how the geometry can be understood through hierarchical relationships between points, planes and vectors.

Deliverables Part A:

- Compile a graphic representation of your component, along with the graphic pseudo-code and diagrams that illustrate how parametric changes produce difference within the component into an 11" X 17" board. Post this board to your blog by 8:00 am Wednesday October 25, 2017.

For the Tutorial:

It is expected that each student will have Grasshopper 3D loaded and running on their computers for the tutorial on Wednesday October 25, 2017 (the latest release of Grasshopper can be downloaded at http://www.grasshopper3d.com/). It is also **strongly** encouraged that students familiarize themselves with the Grasshopper interface and the Grasshopper Primer before the tutorial (The Grasshopper Primer can be accessed at http://grasshopperprimer.com/en/index.html).

The best way to learn Grasshopper is through practice. You will not be able to effectively utilize this extremely powerful and useful tool simply through tutorials and the instructions of others – it requires effort, patience and getting your hands dirty. To this, you should try to have very specific questions for your TA's during tutorials. It is encouraged that all troubleshooting be undertaken through a process of sketching and understanding the problem through geometry (specifically points, planes and vectors). The importance of sketching to understand problems in algorithmic thinking cannot be over emphasized. Sketching leads to intention and intention is necessary to successfully execute any algorithm.

Part B: Grasshopper Component Development

1. Reproduce your parametric component in Grasshopper. Be sure that your Grasshopper component responds to variation in the (minimum) three parameters identified in Part A.
    a. **Hint:** As you reproduce your component within Grasshopper be sure to think about the importance of points, planes and vectors and the hierarchical relationship between these data streams. If you are having troubles in Grasshopper the best way to work through problems is through sketching the relationships and translations within your geometry – it might be helpful to produce additional pseudo-code to understand your original intentions.
2. Design a parametric surface (a surface developed in Grasshopper that responds to at least two parameters).
3. Develop a definition, using the box-morph command in Grasshopper to populate your parametric surface with your component. Explore how changes in the parameters that drive the component and the surface produce variation across the aggregate field.
4. Render the component aggregated across the surface within a particular context.

Deliverables Part A:

- Compile a graphic representation of your component, parametric surface, graphic pseudo-code, diagrams that illustrate parametric change, along with your render onto a 20" X 30" board.
- Post the board and your Grasshopper Definition to your blog by 11:59 pm Sunday October 29, 2017.