ALGORITHMIC FOREST: A Study to Generate >Lighting-Revealing= Structure by Algorithm

JAE HWAN JUNG
C.A.A.D.
Swiss Federal Institute of Technology Zurich (ETH), Switzerland.
jungja@student.ethz.ch

Abstract. In today's world of architectural design the use of computational technology as drawing and modelling tools is ubiquitous. However, the typical use of the computer as a design tool generally limits the creative aspect of architectural design. Incorporating algorithm can enhance traditional ''manual'' methods of CAD-based design, as well as furthering human intellect in the field of architectural design. The research to be presented will demonstrate the potential benefits of algorithms by using them to design and generate a structure that creates variable lighting effects similar to those created by natural light shining through trees in a forest.

1. Introduction

In recent years, as computer technology has become commonplace in the world, its use as a drawing and modelling tool has also become widespread in architectural design. However, the typical use of the computer as a design tool generally limits the creative aspect of architectural design. An alternative process does exist that can take advantage of the computer as a computation machine and can assist in the creative aspects: incorporating algorithms. This involves the creation of scripted programs to generate space and form from the logic inherent in the architectural design process. Applying algorithms can enhance traditional ''manual'' methods of CAD based design, as well as furthering human intellect in the field of architectural design.

The research to be presented will demonstrate the potential benefits of algorithms by using them to design and generate a structure, specifically a light-revealing structure. Light is one of the many considerations in architecture, it reveals the building, its place, form, space, and meaning. Light reveals architecture and, in the best instances, architecture also reveals light. Moreover, light and structure are intertwined. Louis I. Kahn said, ''Structure is the maker of light. When you decide on the structure, you are deciding on light.'' Particularly, this presentation will focus on a structure which creates variable lighting effects similar to those created by natural light shining through trees in a forest.

2. Analysis

2.1. PRINCIPLE ROLES_VARIABLE LIGHTING EFFECTS AROUND A TREE

Figure 1 is an illustration of variable lighting effects. The analysis of these effects determines that they are made of various individual or overlapping layers. Some layers, which are made of open space, let in all the sunlight. Other layers, which are made of individual or overlapping leaves, filter to varying degrees or even block out the sunlight. In addition to the variables created by open space, individual leaves and overlapping, the unique shape and position of each leaf is also a factor. It is the combination of these different factors that produces the final variable lighting effects.

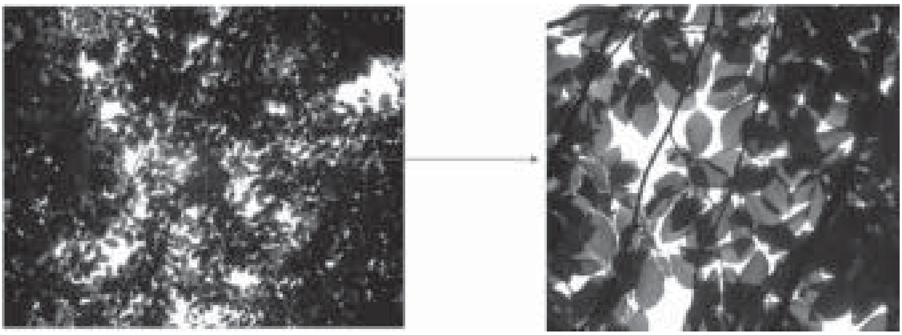


Figure 1. Variable lighting effects.

2.2. GEOMETRIC PATTERN—BRANCHES & LEAVES

Figures 2 illustrates the process by which a geometric representation of branches and leaves is created.

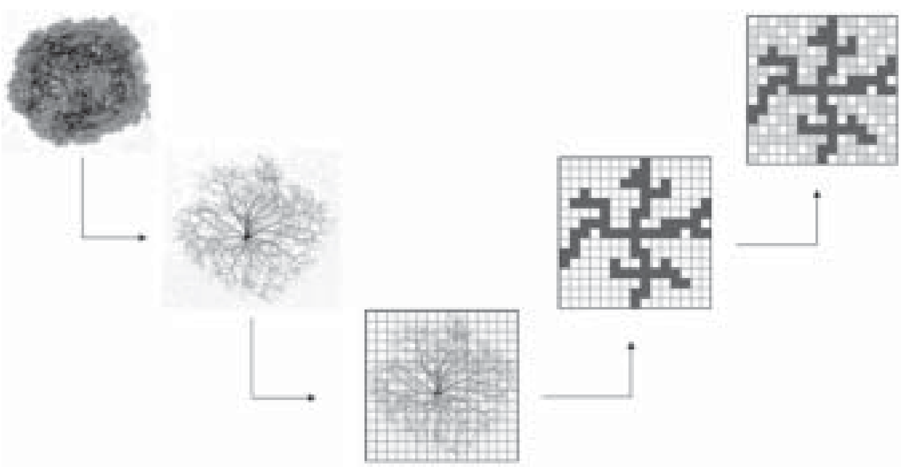


Figure 2. Geometric representation of branches and leaves.

Within this figure in 2-01 we see a photograph of a tree as seen from above; in 2-02 this same view is seen but without leaves so that the image shows only branches, in 2-03 we see Figure 02 with an overlying grid; 04 is a manual recreation of 02 using a mosaic pattern to represent the branches. In 05 we see the final geometric recreation of figure using the branch mosaic from figure four and adding a new mosaic to represent leaves.

2.3. GEOMETRIC PATTERN—TRUNKS

Figure 3 illustrates the process by which a geometric representation of tree trunks is created.

01 02

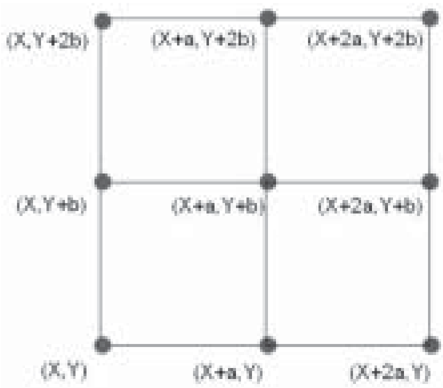
Figure 3. Geometric representation of tree trunks.

In 01 we see a photograph of tree trunks. In particular, we see clearly the natural curvature of the trunks. In 02 is shown the image in 01 with five geometric representations of tree trunks superimposed. These geometric trunks are created using straight lines and angles to represent the trunks natural curvature. The blank circles identify the various angles, of which there are no more than five per trunk, that simplify the geometric recreation of the trunks natural curvature.

3. Generation

3.1. PROGRAMMING ROOF FRAMES

The roof frame is one of the basic elements of the structure. In the structure shown in Figure 4 there are two roof frames that are positioned overlapping each other. They are generated by a regular grid algorithm and a random grid algorithm.



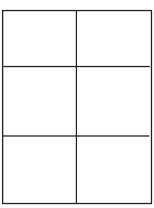

```
for ($x=0; $x<3; $x++)
{
for ($y=0; $y<3; $y++)
{
$aax[$x][$y]=($x*a);
$aay[$x][$y]=($y*b);
$aaz[$x][$y]=0;
}
```

a,b : an invariable number

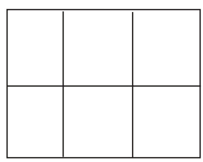
Figure 4. Overlapping roof frames.

The algorithm used to create the roof frames is a variation of a regular grid algorithm. In Figure 4 we see an example of a regular grid algorithm. In this example there are nine points, each given a fixed variable that results in four rectangles of equal dimension.

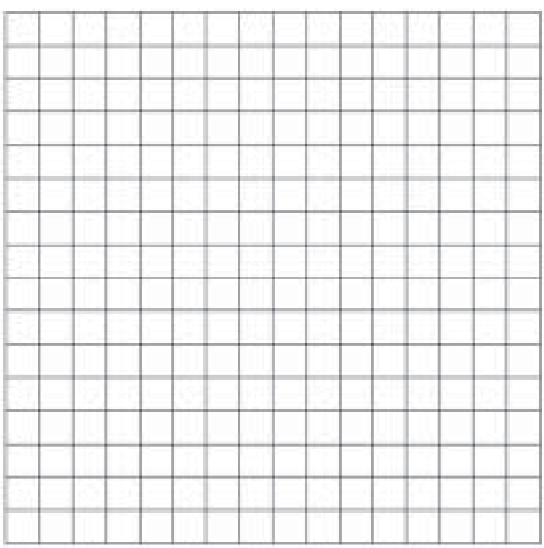
In Figure 5 we see examples of a regular grid algorithm created using different variables.



```
for ($x=0; $x<3; $x++)
{
for ($y=0; $y<4; $y++)
{
$aax[$x][$y]=($x*10);
$aay[$x][$y]=($y*10);
$aaZ[$x][$y]=0;
}
}
```



```
for ($x=0; $x<4; $x++)
{
for ($y=0; $y<3; $y++)
{
$aax[$x][$y]=($x*10);
$aay[$x][$y]=($y*10);
$aaz[$x][$y]=0;
}
}
```



```
for ($x=0; $x<17; $x++)
{
for ($y=0; $y<17; $y++)
{
$aax[$x][$y]=($x*5);
$aay[$x][$y]=($y*5);
$aaz[$x][$y]=0;
}
}
```

Figure 5. Regular grid algorithm using variables.

For the purposes of this project a variation of the regular grid algorithm was created. It is called a random grid algorithm and is used to generate the roof frames. In Figure 6 we see an example of the random grid algorithm.

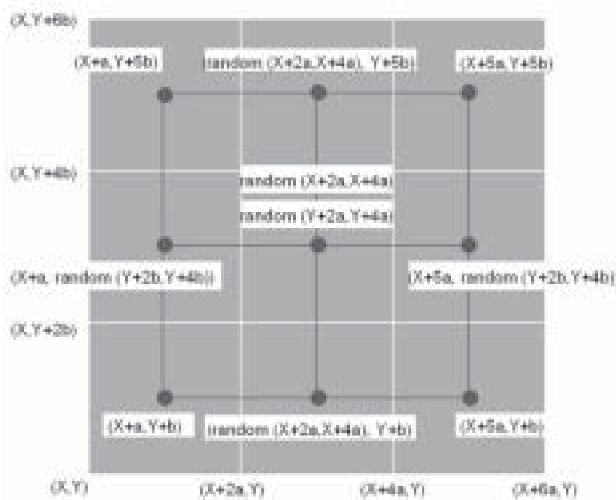


Figure 6. Random grid algorithm.

In this example there are nine points, four with we fixed corners that result in a rectangle. Another four are arranged randomly one point per axis of the rect. These points are given a limited range on the axis along which they can be positioned. The interior point is arranged randomly but also within a limited area. The limited

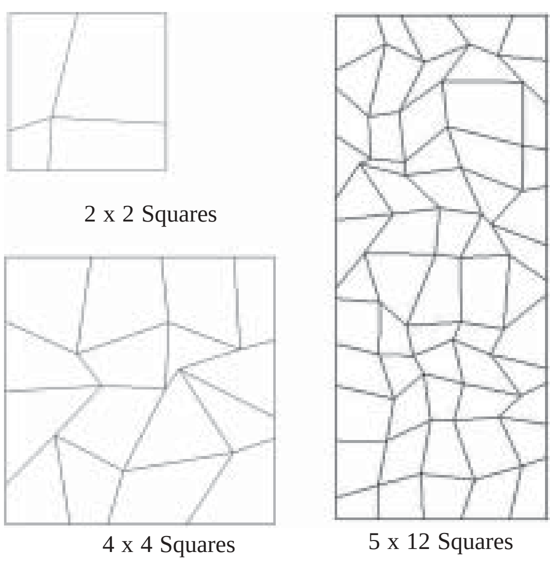


Figure 7. Examples of random grid algorithm.

range and area of the points positions guarantees that the resulting form is always 4-sided. The algorithm creates a grid of 4-sided forms with greatly varied dimensions.

In Figure 7 we see examples of random grid algorithm created using different variables.

3.1. PROGRAMMING ROOF INTERSTITIAL CONNECTOR

The two roof frames are jointed by a roof interstitial connector. The roof connector is generated by a random fractal algorithm. In Figure 8 we see an example of how a random fractal is generated.

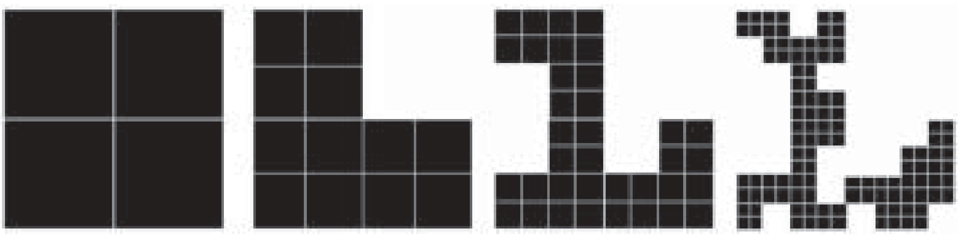


Figure 8. Generating a random fractal.

Start with a filled-in square, divide it into four equal squares and randomly remove one of these squares. Next, divide the three remaining squares into four equal squares and randomly remove one square from each of the three. Continue

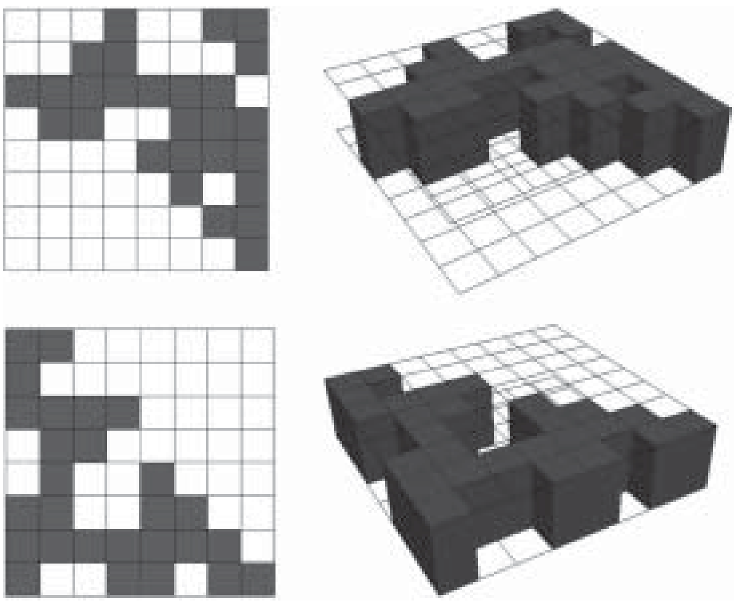


Figure 9a. 8 x 8 Squares.

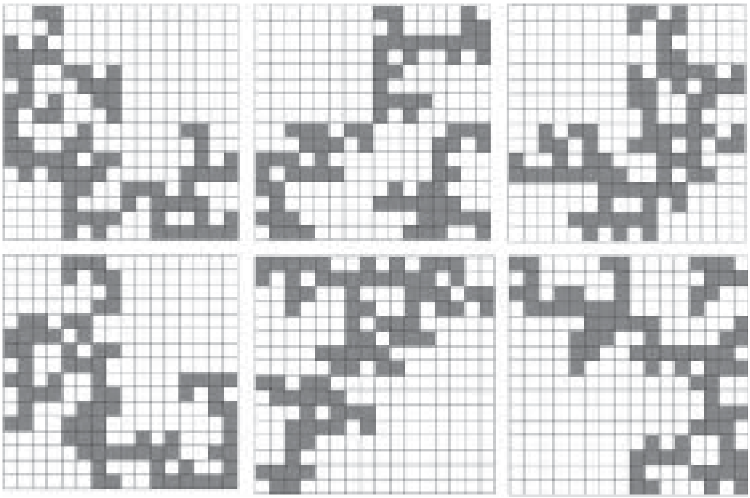


Figure 9b. 16 x 16 Squares.

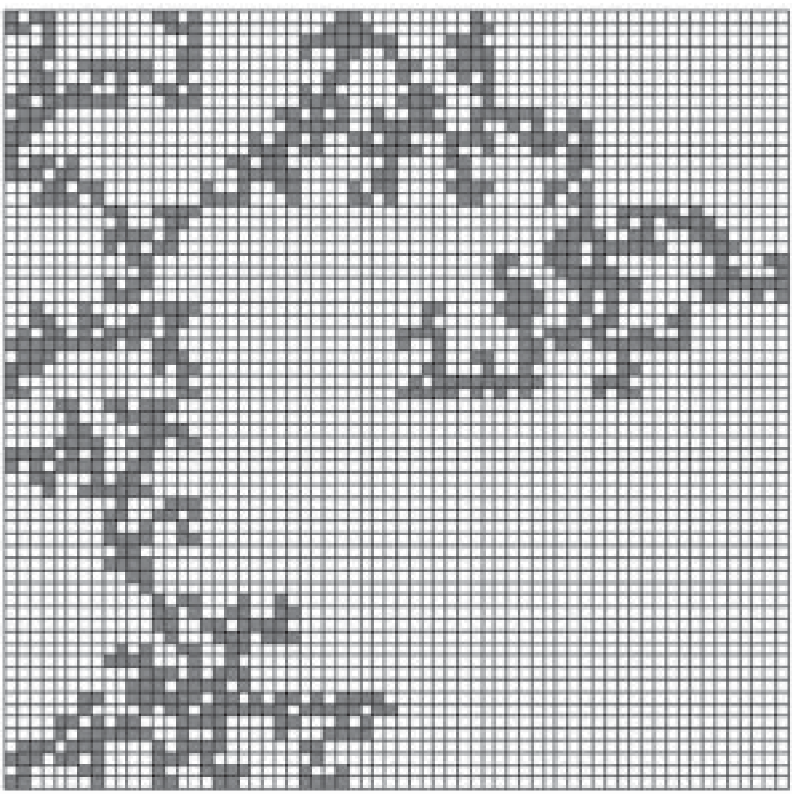


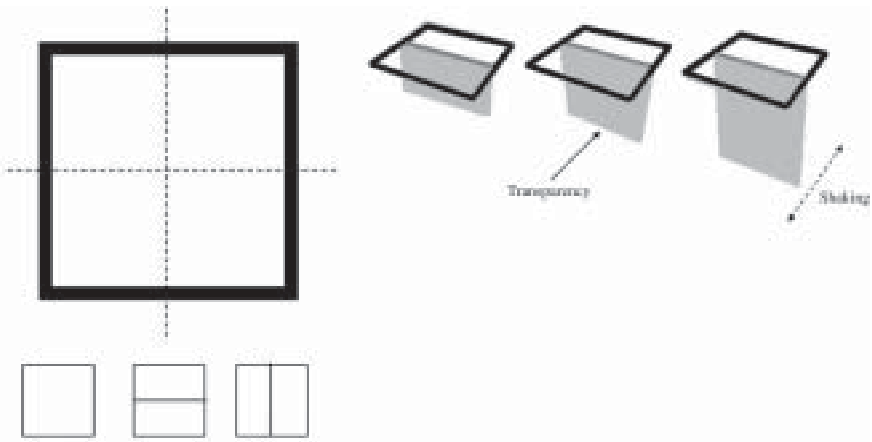
Figure 9c. 64 x 64 Squares.

this process through as many steps as are desired or possible based on grid parameters.

The purpose of the pattern of the roof connector is to represent the geometric pattern pf tree branches. As in the roof frame algorithm many instances are possible when different variables are used. In Figure 9 (a, b and c) we can see examples of roof connectors created using the random fractal algorithm.

3.1. PROGRAMMING ROOF SURFACES

The roof surfaces hang from the two overlapping roof frames and are generated from a random face algorithm. The random face algorithm determines if a unit of the roof frame will have a roof surface and if it does which position and what length it will have. Figure 10 illustrates the possible examples.) The random face algorithm also prevents roof surfaces and roof connector from occupying the same roof frame unit.



01 02

Figure 10.

The purpose of the roof surfaces is to represent the geometric pattern of tree leaves, so that the structure recreates variable lighting effects similar to those created by natural light shining through trees in a forest.

3.2. PROGRAMMING COLUMNS

The columns' forms are generated according to the random growing algorithm. There are two types of random growing algorithm, each generating a different type of column; one generating main columns, the other sub columns. Together they support and stretch like the trunks of trees in a forest. In Figure 12 we see grids positioned at different levels. The random growing algorithm determines that each

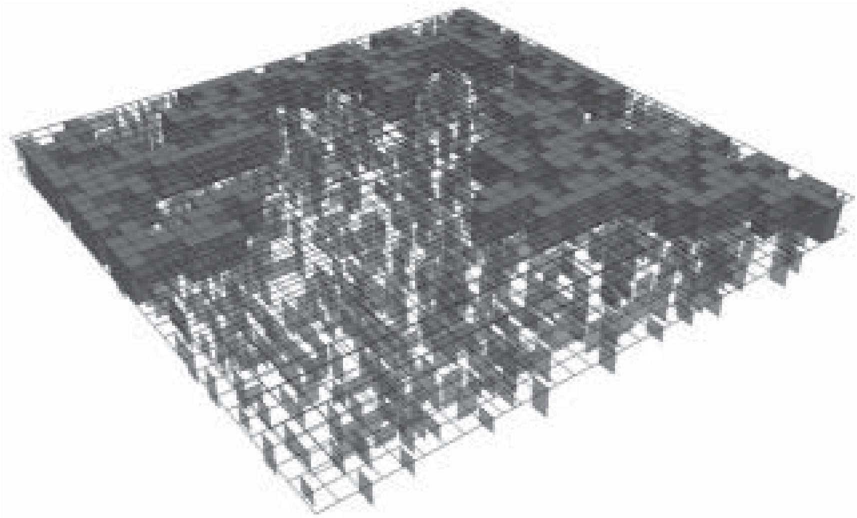


Figure 11. 32 x 32 Squares.

column begins in the middle of the top grid. Next the algorithm randomly chooses a position on the grid below to which the column will extend. Then the algorithm chooses a position on the next lower grid to which the column will extend. The algorithm for the main columns have one angle, the sub columns two. For the main columns the algorithm generates an angle, for the sub columns two angles are generated.

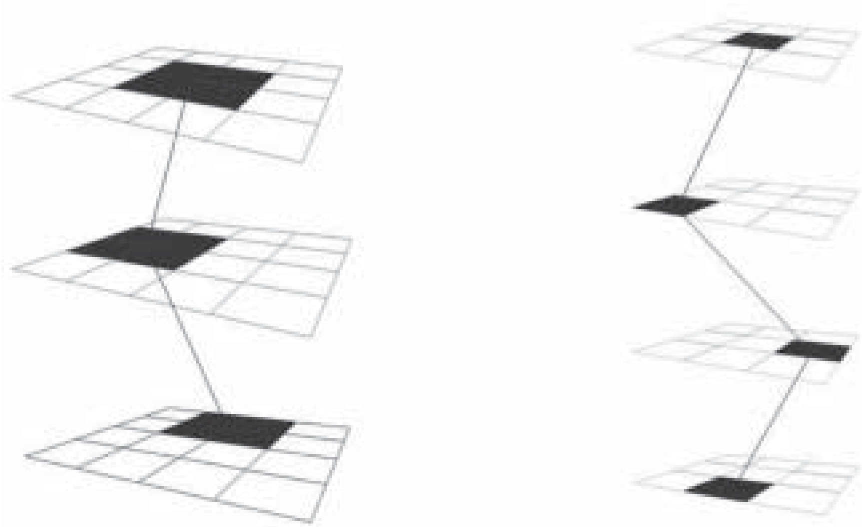
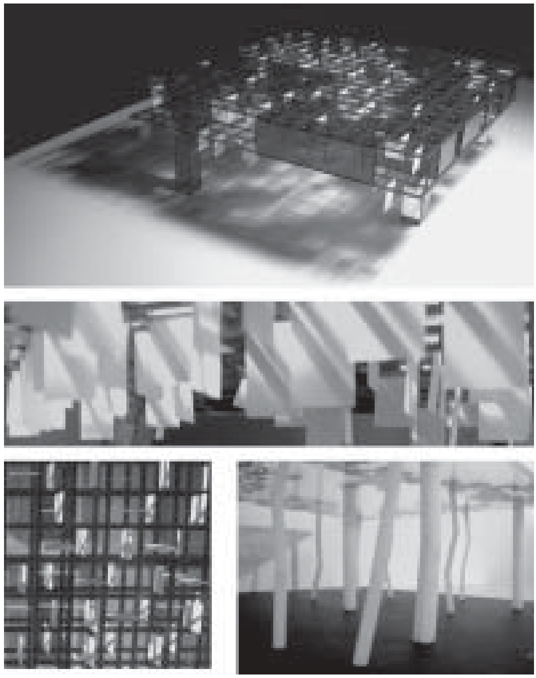


Figure 12. Grids at different level.

3.2. PROTOTYPE



4. Conclusion

The use of algorithm as a computation method in the field of architectural design can be expected to bring a number of benefits. One is that results can be unpredictable and unexpected, inspiring architects to create innovative solutions. Another benefit is that results are adaptable, allowing architects to easily alter designs according to the parametric demands of the project. Additionally, results can be numerous, enabling architects to quickly evaluate and choose the most suitable results. The conclusion of this research indicates that architects using algorithms can augment the intellectual process of design with the capabilities of computation, resulting in a symbiotic process that reveals new facets of architectural design.

References

Cecil, Balmond. 2002, Informal, Prestel.

Fractal Geometry. http://classes.yale.edu/fractals/Welcome.html

Kostas, Terzidis. 2003, Expressive Form: a Conceptual Approach to Computational Design. Spon Press.

Marietta, S. Millet. 1996, Light Revealing Architecture. VRN

Wurman, Richard Saul. 1986, What will be has always been: the words of Louis I. Kahn. New York: Rizzoli, p. 63.