

City Generator: GIS Driven Genetic Evolution in Urban Simulation

MING TANG

Savannah College of Art and Design

INTRODUCTION: GENETIC EVOLUTION AND ITS CONSTRAINTS IN URBAN SIMULATION

Since Charles Darwin proposed his theory stating that all species are generated via the process of Natural Selection, evolution and genetic approach are considered not only powerful enough to solve biological puzzles, but also effective in creating algorithms and simulating complexity and diversity. There are many applications and research aspects of "Evolutionary Design". Bentley defines genetic evolution (GE) as "using computers to generate the form of designs, the emphasis for this type of Evolutionary Design is on the generation of novelty and originality, and not the production of globally optimal solutions.¹" As a cross-disciplinary research method, GE borrows many concepts from biology such as genotype², phenotype, expression³ and reproduction⁴. It has been used in the field of urban design to achieve a higher degree of synthesis, such as "virtual architectural models⁵" designed by John Frazer, and the "City-Engine⁶" developed by Pascal Mueller. Today, cellular-automation⁷ and L-system⁸ are two popular form-seeking techniques closely associated with GE methodology. With cellular-automation, computer simulated evolution follows the same principle of survival of the fittest⁹. A large quantity of solutions are generated and compared in the computer-simulated environment¹⁰. While the other form-seeking technique, L-system, procedurally breeds complex offspring from a single axiom by applying reproduction and mutation principles. In many experimental projects, these two tech-

niques have shown great potentials of generating a large population of variations.

However, both of these techniques require the input of abstract scripts rather than spatial information. The expression of form generation is entirely interpreted by computer scripts, which are not integrated with any standard CAD software. The lack of comprehensive understanding of shape grammar becomes the bottleneck for architects and urban designers to explore the potential of GE in the design-orientated environment. After exploring both GE methods in several courses taught at the Savannah College of Art and Design (SCAD), it becomes obvious to me that the disconnection of an abstract script (genotype) and its corresponding form (phenotype) makes the current GE approaches hard to control, especially for the students without programming skills. Therefore, it is desirable to develop a new tool allowing designers to effectively implement GE techniques in the design-orientated environment without learning complex shape grammars. The goal is to develop a new tool to automatically construct 3D urban forms by applying certain evolution rules to the existing spatial data. With this tool, an urban model is generated as a synthetic creature, which is meaningful on various aspects far beyond its aesthetic values.

THE OBJECTIVE OF CITY GENERATOR

In the past four years, GIS program has been taught at SCAD to introduce students the use of

computers in the assessment and representation of the environmental landscape. Data collection, assessment and synthesis are incorporated as components of the urban design courses. Usually, 2D diagrams and GIS maps are generated at the end of the procedure of spatial data capture, management, manipulation, and analysis. However, these 2D images are usually only used during the early design stage such as site analysis, rather than being used as instruments for designers to study, experience, and revise their design in various design phases in the 3D environment.

Taking advantage of the powerful GE method, this paper describes an alternative and practical approach to use 2D GIS maps for instant 3D urban modeling, with urban design students as target audiences. Therefore, traditional 2D images, diagrams and maps can be directly associated with the 3D form of a city in various design phases. It allows designers to explore their design by using both 2D and 3D instruments in all design phases, rather than using 2D in the early stage, while 3D as a tool to simply visualize the outcome of the design. In an ideal situation, students can generate instant 3D urban models through very simple operations in the computer by creating 2D urban diagrams or manipulating GIS maps. The computation tool should automatically construct the 3D city based on the logic of 2D blue print.

Besides its application in urban design, this method can also be applied in the field of visual effects, game design or historic study for its effectiveness of generating large 3D city models.

Over the last two years, investigations were made for various software (ArcGIS, Maya, Max, Houdini) and algorithms (L-system, turtle script, cellular-automation, shape grammar) in the digital computing and urban design courses taught at SCAD, with the emphasis on their potentials of creating urban forms for the production of novelty and originality. The investigation started from the research on the L-system and Cellular-automation. As a result, a MEL / TCL¹¹ based Maya plug-in, which we named as *City Generator*, has been developed to integrate GIS into GE techniques. It can automatically generate 3D urban forms by interpreting 2D GIS data. Two modeling approaches have been applied for this purpose.

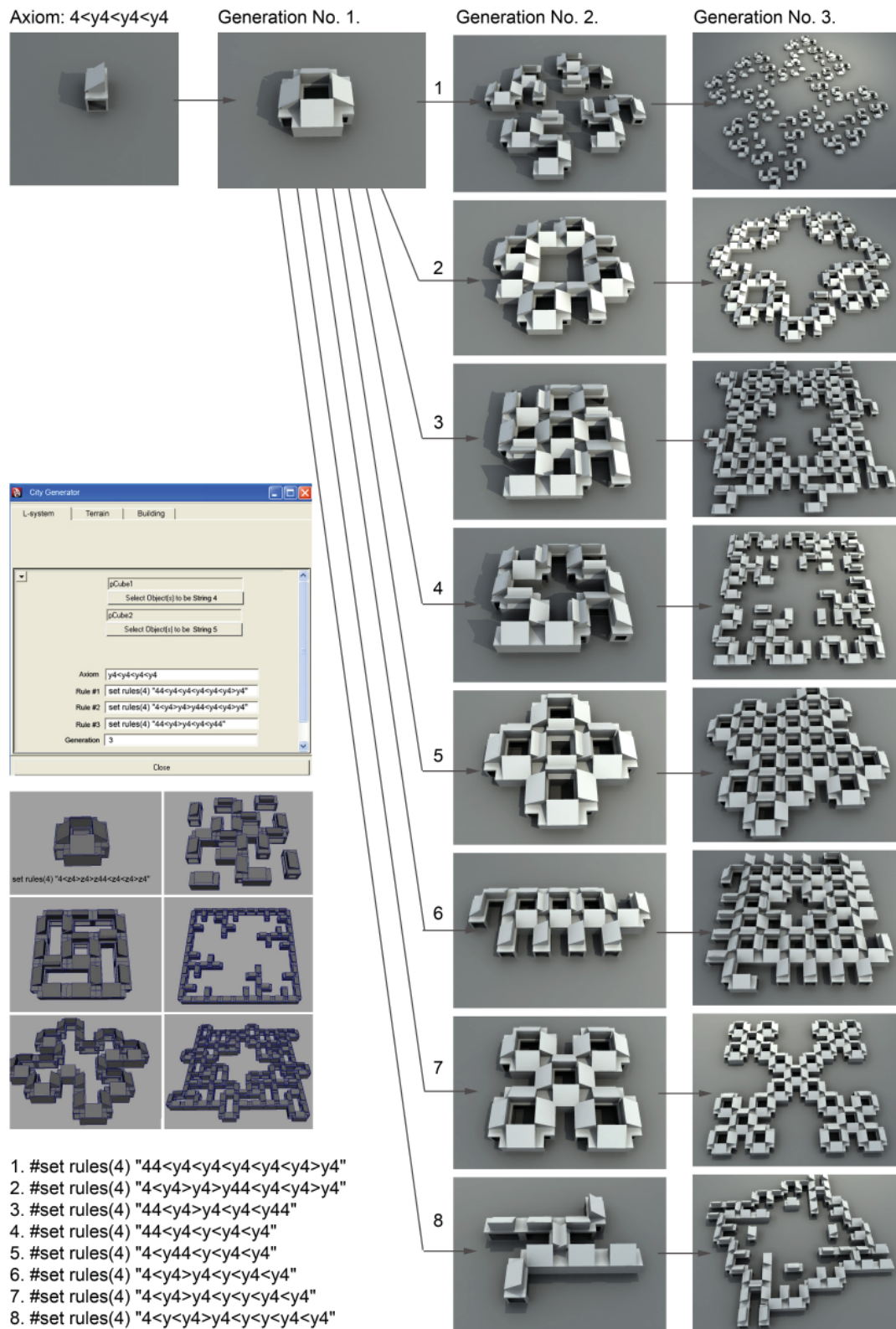
PROCEDURAL ARCHITECTURAL MODEL

Procedural modeling with shape grammars is increasingly implemented in genetic computing to create 3D models with a high degree of complexity, such as the urban simulation projects by Benjamin Watson, and *City Engine* by Pascal Mueller. Inspired by their research, as well as the procedural modeling and L-system in Houdini software¹², we have developed a graphic interface for *City Generator*, which utilizes a set of strings to generate simulated urban patterns. The goal is to maximize the spatial variations and explore the relationships among individual urban elements such as the streets, buildings, and open spaces. In an experimental project, a traditional Chinese courtyard house (axiom) is assembled and applied self-organization rules using a sequence of L-string (Figure 1).

Although the procedural model in *City Generator* does not provide a complete control over every detail of the courtyard house due to the limited number of dynamic parameters, it does allow the creation of a large population of complex urban patterns that is user directed. Similar as how sexual reproduction allows characteristics from parents to be transferred to their offspring, each generation of the courtyard houses delivers their features (genotypes) to the next generation, which allows desirable features to evolve independently and later be multiplied in a large community scale.

In this modeling approach, random noise is added into the L-System. Similar as the mutation in a DNA string, the noise normally has a subtle effect on the phenotype unless it has been amplified or accumulated through several generations. Controlled by a sequence of L-strings and the interpretation of each character, this type of mutation allows the complexity of a form to grow continuously as evolution proceeds. Since noise is performed on a random basis, models with low noise frequency may not trigger any mutations, while the ones with high noise frequency will abet an extraordinary mutation that can override their parents.

Although procedural modeling has shown its potential in providing a great variety of forms, it still requires designers to define the axiom and evolution rules by the abstract strings, which do not

Figure 1. L-system in *City Generator*

normally associate with any urban design vocabulary. Although a graphic interface has been developed to assist non-programmers to write abstract strings, the complex expression between genotype (string) and its phenotype (3D form) still frustrates our students. Therefore, we start to research how to replace this abstract input by GIS data input, which is directly related with more understandable spatial features.

SPATIAL OCCUPANCY (SO) URBAN MODEL

Inspired by the cellular automation and self-organization, the Spatial Occupancy (SO) modeling was investigated for its capability of representing complex geometries in the molecular level. It is effective to compile a set of contiguous discrete "chunks" of "voxels¹³" to define a SO model. Large quantities of voxels are arrayed in space and thus formulate the presented urban forms. The characteristics of the form are heavily relied on the voxel's spatial relationship and the internal logic among themselves. An experimental project, "Invisible Manhattan", has articulated this technique in the first version of *City Generator* (Figure 2). To maximize the potential scenarios of the urban form in Manhattan, internal parameters of each voxel are subjected to alterations controlled by a GIS population density map, which determines the height for each voxel. This technique of using a 2D map to control its corresponding 3D geometry has been widely used in the animation industry to control a digital character's bone system and deform the corresponding muscle appropriately. In our approach, each voxel's height value is controlled by its corresponding pixel's alpha value in the GIS population density map. The map is generated by ArcView according to Census 2000 population data on the block level. An amplifier is then added to its original 3D displacement value, following by the application of the density map to the controlling voxels in the simulated Manhattan model. The resolution of the 2D density map controls the quantity of voxels used to "build up" the 3D urban form (Figure 2).

It is evident that SO modeling has significant benefits in the evolution process since the entire form is modulated at the molecular level. When two SO models have identical voxel resolution, it is easy to mix their genetic features and reproduce a new form by simply blending the corresponding

voxels from both parent models. Inspired by the crossover of DNA strands that occurs in the reproduction of biological organisms, we can combine existing urban forms (parents) and created new urban forms (children) with some of their features inherited from each parent. For example, parent A is produced based on the population density data from the census bureau, while parent B is produced based on the land value data from real estate database, while parent C is produced based on the zoning map from local planning department. Groups of new urban forms are built up by "mixing" the gene of parent A, B, and C. The specific characteristic of each child is inherited from all parents through its voxels. Each voxel's spatial value carried in parents is passed to their children's related node with a randomly assigned weight by the "Blending Shape¹⁴" function in the 3D computer program.

This process can be described by two steps. First, various GIS data is utilized to create the first generation of urban forms. Then a prototype matrix is created to drive the deformation of target geometry mesh. For instance, 100 children are produced from each pair of "urban parents". The first child is identical to parent A, while the 100th child is identical to parent B. The other 98 children are the mixture of parent A and B with a different weight combination¹⁵.

INTEGRATION IN THE DESIGN CONTEXT

As a new tool for urban simulation, the central feature of the *City Generator* is a Genetic Evolution (GE) engine, which can quickly generate various urban forms by the combination of L-system controlled procedural model and the GIS map controlled SO model. The problem is, how could this tool capture the underlying urban design principles from which a range of potential solutions can be explored? It is believed that the solution relies on the human computer interaction. Instead of creating an instant city by a single data input, *City Generator* requires a user to be constantly involved in the evolution process during multiple computing phases. Designers play a significant role in optimizing the form mating, reproduction, mutation, and selection. (Figure 3)

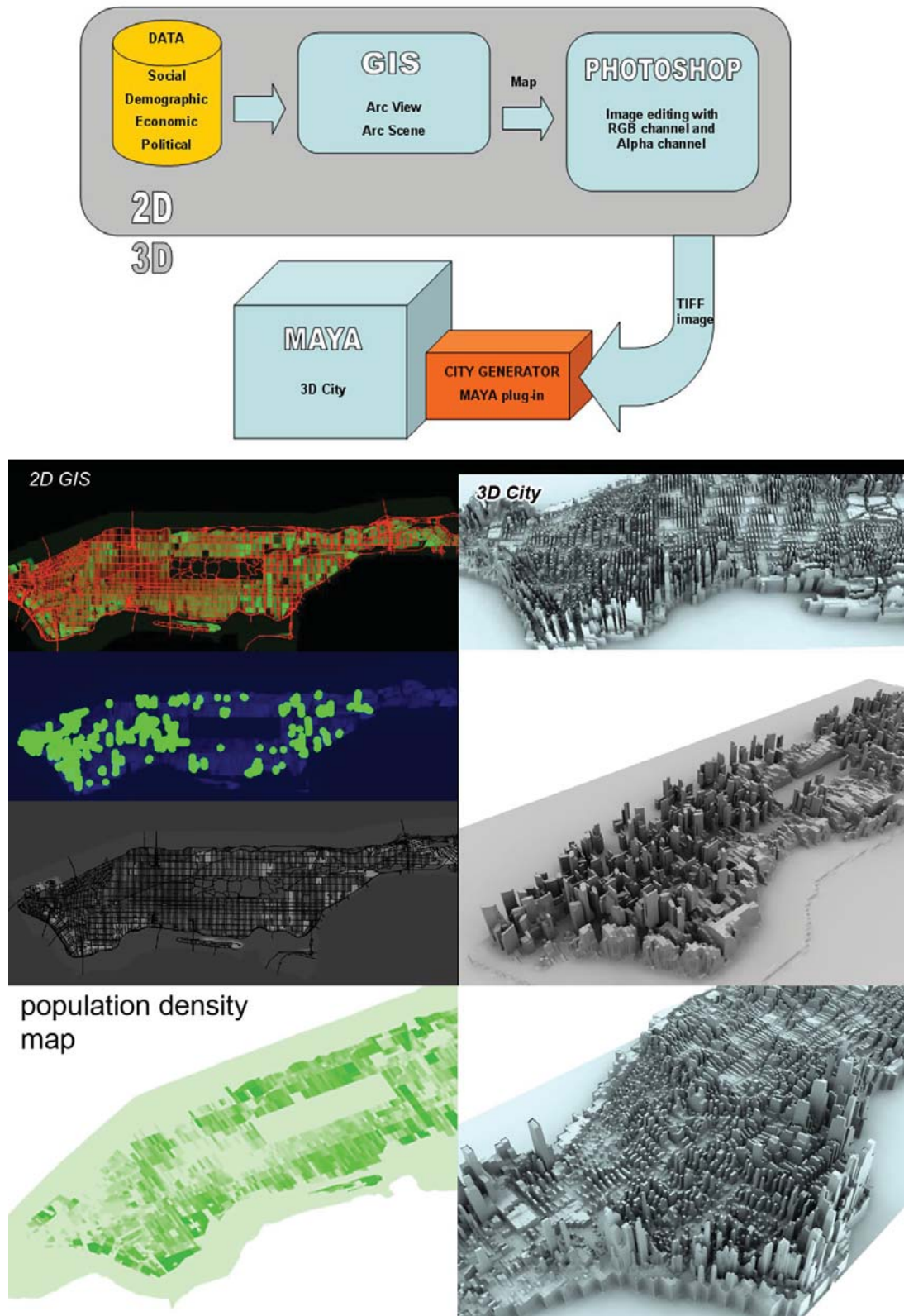


Figure 2. Top: Workflow for City Generator. Bottom: SO model of Manhattan.

Phase I: building generation (No GIS involved)

In the first phase, *City Generator* is used to “breed” selected architectural models across several generations. This process is not related with any urban context and there are no GIS inputs involved. The goal is to produce various building models from selected architectural types. Hundreds of different building models are generated at the end of this process and some ideal models will be selected and used in the urban context in the Phase III later. For instance, a group of ideal houses are selected from a large candidates’ pool based on the evaluation criteria. These “survivors” then mate with each other and generate the second generation. Frequently we introduce a brand new house type into the breeding process as a new genotype in order to stimulate more diverse outcomes.

We also periodically stimulate mutations in several members of the current population, and yield a new candidate solution. This new solution may be accepted or eliminated based on its fitness. Similar selection process as the natural selection is applied, in which the “most-fit” members of the population will survive, while the “least fit” will be eliminated. Because the complexity of fitness in urban design is far beyond the rules defined by computer algorithm such as cellular automation, evaluation and selection of the most-fit is exclusively operated by reviewers rather than artificial intelligence. In *City Generator*, the evaluation of “fitness” depends on the feasibility and functionality of a form in addition to its aesthetic value, which is subject to the reviewers’ decision. As a result, we breed a group of ideal houses, which are ready to be used in Phase III.

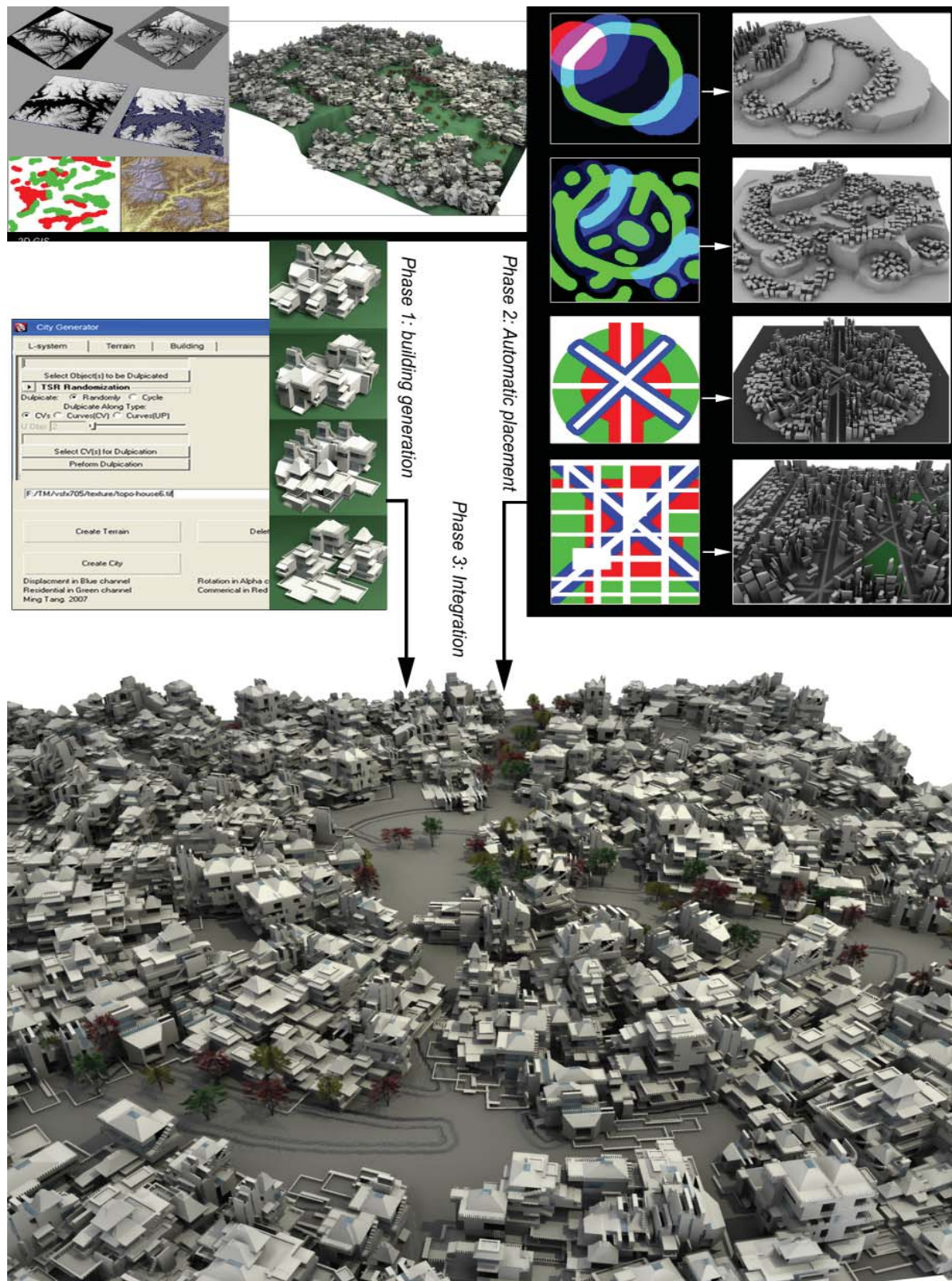
Phase II: automatic placement in the urban context (GIS involved)

In the second phase, *City Generator* uses 2D maps to generate 3D SO model. These 2D maps contain GIS data, which reflect zoning, population, transportation, and other spatial information. They can also be manipulated directly in an image-editing program such as Photoshop to represent an urban designer’s master plan. *City Generator* can read the 2D information embedded in GIS maps based on the RGBA¹⁶ value of each pixel and execute corresponding modeling operation. In another word,

during the automatic construction, the resultant 3D city model can inherit all the geographical, social, demographic and design information from the 2D GIS maps. For instance, a digital elevation model can be converted into a blue color TIFF image in GIS and later the blue color value can be read by *City Generator* to define the terrain elevation¹⁷. A commercial zoning map can be converted into a red color map in GIS and later the red color can be read by *City Generator* to construct a series of commercial buildings. Each building has a unique height correlated with the land value. A vegetation map can be converted into a green color map and later the green color can be read by *City Generator* to grow various types of plants. Overlapping multiple 2D GIS maps forms the potential mixed-use zones and complicated urban features, which can be continuously configured in a simple image editing program based on the design criteria and then generate instant 3D results. Therefore, instead of writing complex scripts or doing labor-intensive modeling, designers only need to create 2D maps, observe and evaluate the 3D urban modeling process, and control the result in a preferred direction. At the end of this phase, designers get an abstract urban model represented by a large quantity of voxels. Because the SO model only uses a simple proxy geometry (voxel) to represent a building (or a tree) instead of using a highly detailed procedural model, the processing time from 2D to 3D is short and the model is easy to be manipulated back and force.

Phase III: substitute proxy by procedural building models. (GIS involved)

In the final phase, each voxel in the SO model of Phase II is substituted by a detailed procedural building model survived in Phase I. The substitution rule is defined by the designer according to design criteria. For instance, the building’s local transformation node such as orientation, scale, position can follow its voxel’s node, or be re-defined by the information embedded within the RGBA value of new GIS maps such as view shed maps or slope maps. In an experimental project, a 2D map is originally generated based on the population density in GIS software, modified in Photoshop by the designer. This map is then read as the instruction menu by *City Generator* to automatically grow an urban form presented by thousands of voxels. Finally each voxel is “filled in” by an appropriate

Figure 3. Three phases of using *City Generator*

building model generated in Phase I. The height of each building is controlled by each voxel's population density value.

SWARM CITY PROJECT

City Generator has become a powerful instrument for us to produce new solutions and more exciting urban forms. The "Swarm City" project has explored the applications of this new tool in both large urban scale and small architectural scale.

The first part of the Swarm City project is to explore the transformative spatial layout of housing units. It is inspired by Proxy Paine's *Auto Sculpture Maker* (SCUMAK) in 1998¹⁸. In this installation, Paine used computer controlled machine to create polyethylene sculptures. Each sculpture was formed by rigidly scripted machine and identical parameters. However, due to the forces of chaos, the produced forms were all different. The artist created the contrast between his personalized mass production and the depersonalized industry manufacture pipeline. Similar as SCUMAK, *City Generator* is used to massively produce a large quantity of housing units with variations among each of them. However, the variation in *City Generator* is not controlled by the force of chaos, but the mutation in GE program. In nature, when two individuals mate, each passes half of its paired chromosomes onto its common offspring¹⁹. Following this nature analog, procedural houses "genetically" embed variability with a permutation of all the possible spatial arrangements (Figure 4).

First, five basic building units are modeled and imported into *City Generator*. Then we execute the breeding process and produce 3125 offspring in the first generation. This process is finished by an exhaustive combination of five original units' genotype. From these 3125 samples, only five ideal spatial arrangement solutions are selected by the reviewers and thus reserved as the genotypes for the next generation. To stimulate the mutation, a nonlinear deformation node is evolved independently in the evolution and then explicitly added to the process to yield more complex layout potentials. In addition, a central courtyard, a negative volume, is also introduced into the evolution as a "void unit" and blended with the selected layouts. As a result, the evolution process has created a high degree of complexity and maximized the

dynamic possibilities of spatial arrangement with relatively simple input. In this process, GE demonstrates an unlimited potential of form reproduction driven from a set of genetic parameters. The reviewer consistently selects the desired spatial layouts and mixes them in the new generation.

The second part of the Swarm City project is using the selected housing units to grow the entire city based on the GIS data. The goal is to simulate the possible form of urban fabric, a diagram city, which grows from an organic natural context, rather than following an imposing neutral gridiron pattern. The complex urban fabric relies on how individual building responds to the logics of the mass and landscape features such as contour, vegetation, and view shed. For this purpose, multiple SO models driven by 2D diagrams (painted by the designer) and GIS maps (from ArcView and DEM TOPO! plug-in) are generated and evaluated.

For example, a view-shed map is created in ArcView to control the placement of certain house with a big roof balcony and an observing tower. A family income map is generated to control the placement of public housing. A population density map is used to control the placement of community centers. A DEM elevation map is generated in GIS TOPO! to move all buildings up and down following the contour. These 2D maps using RGBA value to guide thousands of voxels to behave like swarm intelligence²⁰. Then each voxel is substituted by a detailed architectural unit. Finally, time, as the 4th dimension, is added to capture the growth into an animation. Expressions are altered and various spatial arrangements are produced as the value of time is smoothly animated.

CONCLUSION

In addition to the integration of 2D GIS maps into 3D design process, one of the important objectives of *City Generator* is to create an engaging experience that allows designers to control the evolution of 3D urban forms in a dynamically changing interface. In the experimental projects, *City Generator* can process hundreds, even thousands of urban forms in a relatively short period of time. As the automatic construction is generated very quickly, it is advantageous for designers to choose from a large candid pool instead of being binding in a few solutions.

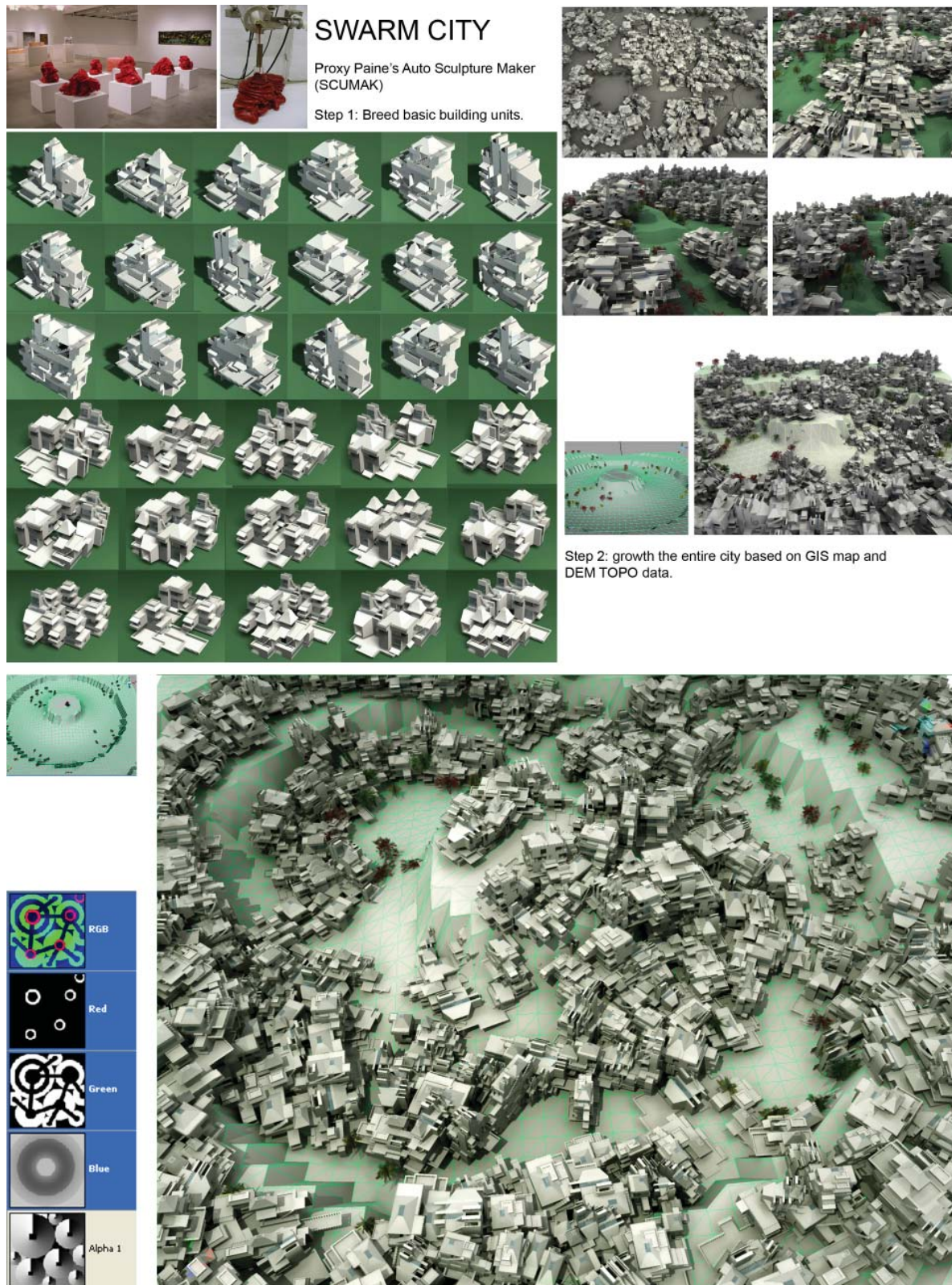


Figure 4. Swarm City project

This new design process is considered as a psychological change rather than simply a new visualization method. It assists designers "to observe various possibilities which may not have been thought about before" (Yuhuda Kalay. 2000), which may lead to creative design solutions. With the mature of artificial intelligence, it may even be considered as a system "which attempts to learn about human aesthetics from the user and apply fitness evaluation automatically. In either case, it will allow the user and computer to interactively work together in a new way to produce results that neither could easily produce alone." (Karl Sims. 1994)

However, there are several constraints in the current GE technique that prevents it from being widely adopted by researchers and mainstream practice. Other than the complex expression of the abstract script (genotype) and the unpredictable form it generated (phenotype), which have been discussed before, another constraint is the missing "shared body plan"²¹. As described by Manuel De Landa, "...if evolved architectural structures are to enjoy the same degree of combinatorial productivity as biological ones, they must also begin with an adequate diagram, an abstract building corresponding to the abstract vertebrate". However, this shared body plan can not be achieved across various building types or urban forms by the procedural modeling and shape grammars. *City Generator* can only apply GE in the voxel level without recognizing the urban components such as street pattern, building footprint, architectural elements, and etc. Neither the SO model nor the procedural model can provide a higher level of abstraction to form a "shared urban plan".

Another constraint is the evaluation of a large quantity of simulations generated from multiple independent GIS datasets. As argued by Peter Bentley, one severe difficulty for GE is its "complex evaluation routines" (Peter Bentley, 1999). In *City Generator*, GE engine can generate a large quantity of urban forms, each of which perfectly responds to one specific GIS input. For instance, one GIS map can guide computer to produce an ideal urban form matching the population density pattern. Another GIS maps can guide computer to produce a new urban form matching the family income pattern. Certainly *City Generator* can breed these two urban forms and generate many

offspring to reflect both social patterns. However, how to judge the balance between these two independent influences is a big challenge during the evaluation. Should we use 50% family income pattern plus 50% population density pattern, or 10% family income pattern plus 90% population density pattern? When we have more and more ideal urban forms responding to other social, political and economical features, it will be too overwhelming for a designer to make an appropriate evaluation.

We don't believe this constraint can be overcome by the growth of artificial intelligence in the near future. It is still necessary for *City Generator* to rely on the evaluation from human, and more importantly, the creativity of human. To create meaningful urban forms in an ideal design environment, designers should be able to continuously optimize computer's exhaustive permutation and modeling power, guide the evolution process with design concepts and rules. Once this is achieved, the computer can be used as a synthetic assistant and evolutionary force.

REFERENCES

- John Frazer. *An Evolutionary Architecture*. London: Architectural Association Publications, 1995.
- Manuel De Landa. "Deleuze and the Use of the Genetic Algorithm in Architecture," *Design for a digital world*, Wiley-Acamedy. New York: Wiley, 2001, pp117-120.
- Pascal Mueller, Gang Zeng, Peter Wonka and Luc Van Gool. "Image-based Procedural Modeling of Facades," *Proceedings of ACM SIGGRAPH 2007*. New York: ACM, 2007.
- Yoav Parish and Pascal Mueller. "Procedural Modeling of Cities," *Proceedings of ACM SIGGRAPH 2001*. New York: ACM, 2001.
- Karl Sims. "Evolving 3D Morphology and Behavior by Competition," *Proceeding of the Artificial Life IV conference*. Cambridge, Massachusetts: MIT Press, 1994.
- Bentley, Peter. J. *Evolutionary Design by Computers*. Massachusetts: Morgan Kaufmann, 1999.
- Rowland, Duncan. "Evolutionary Cooperative Design Methodology: The Genetic Sculpture Park," *Leonardo* 35:2. Cambridge, Massachusetts: April 2002.
- Pablo Funes, Jordan Pollack. Computer Evolution of Buildable Objects. *Fourth European Conference on Artificial Life*, P. Husbands and I. Harvey, eds. Cambridge, Massachusetts: MIT Press. 1997.
- Yehuda E kalay. *Architecture's New Media*. Cambridge, Massachusetts: The MIT press. 2004.
- Testa, P., O'Reilly AND U., Kangas, M. "MoSS: Morphogenetic Surface Structure — a software tool for design

exploration," *Proc. Digital Creativity Symp.* London: Univ. Greenwich, 2000, pp71-80.

Bettum, J. AND Hensel, M. "Channeling systems: dynamic processes and digital time-based methods in urban design", *Contemporary Processes in Architecture, Architectural Design*, 2000, p36-43.

Peter Wonka. *Procedural Modeling of Architecture. Course procedural modeling of urban environments.* SIGGRAPH 2006.

Neil Leach. "Swarm Tectonics," *Digital Tectonic, Wiely-Academy.* New York: Wiely, 2004.

Endnotes

1. Evolutionary Design could encompass four distinct aspects: "Evolutionary Design Optimization, Creative Evolutionary Design, Evolutionary Art, and Evolutionary Artificial Life Forms". Bentley, Peter. J. *Evolutionary Design by Computers.* Massachusetts: Morgan Kaufmann, 1999.

2. In biological systems, genotype is the genetic code for creating an individual, which is normally composed of DNA. In simulated evolutions there are many possible representations of genotypes, such as strings of binary digits, sets of procedural parameters, or symbolic expressions.

3. Expression is the process by which the phenotype is generated from the genotype. Expression can be a biological developmental process that reads and executes the information from DNA strands, or a set of procedural rules that utilize a set of genetic parameters to create a simulated structure.

4. Reproduction is the process by which new genotypes are generated from existing genotypes. For evolution to progress there must be variation and mutation in new genotypes with some frequency. In biology, mutation could change the phenotype such as the color of fur, the size of wing, etc. Certain phenotype is more adapt to certain environment than others, which makes the creature with this phenotype more competitive and be able to successfully pass its genotype to the next generation. In a computer-simulated evolution, selection is either controlled by human beings or artificial intelligence.

5. John Frazer developed an alternative method of generating architecture by evolving a computer model in a simulated environment. "Within the virtual world of the computer, this evolutionary process is compressed in time and space - a large number of iterative cycles can be performed very quickly." John Frazer, *An Evolutionary Architecture.* London: Architectural Association Publications, 1995.

6. Mueller proposed a system using a procedural approach based on L-systems to model cities. "From various image maps given as input, such as land-water boundaries and population density, this system generates a system of highways and streets, divides the land into lots, and creates the appropriate geometry for the buildings on the respective allotments." Yoav Parish and Pascal Mueller. "Procedural Modeling of Cities,"

Proceedings of ACM SIGGRAPH 2001. New York: ACM, 2001.

7. A cellular automaton is a discrete model studied in computability theory, mathematics, and theoretical biology. It consists of a regular grid of cells, each in one of a finite number of states. Every cell has the same rule for updating, based on the values in this neighborhood. Each time the rules are applied to the whole grid a new generation is created.

8. Lindenmayer system is a formal grammar (a set of rules and symbols) most famously used to model the growth processes of plant development, but also able to model the morphology of a variety of organisms.

9. In nature, fitness is simply the ability of an organism to survive and reproduce. In the digital computation, computer program is used to evaluate the outcomes and eliminate the "unsuccessful" one.

10. Within the evolution hierarchy all individual features are condensed and passed to the next generation. Here variations can simply take place by changing the crossover rule in hierarchy, or introducing a new genotype into the system.

11. MEL is a programming language within Autodesk Maya program. TCL, stand for Tool Command Language, is an interpreted language with programming features.

12. HOUDINI is 3D software developed by Side Effects Software, Inc. It provides procedural 3D animation and special effects tools for film, broadcast, entertainment and visualization production.

13. Each voxel (a portmanteau of the words volumetric and pixel) is a volume element, representing a value in 3D space. This is analogous to a pixel, which represents 2D image data. A 3D grid occupied by the object is used to contain all voxels. These volumetric voxels are fixed-size cubes.

14. This technique has been used in the computer generated facial animation for many years. Blending shape allows the animator to morph facial expression based on multiple individual target. For instance, a "happy" impression and a "sad" impression could pass their influence to the same face. The result is not just a neutral impression, but unlimited blending impressions based on the weight shifted between "happy" and "sad".

15. For instance, the second child had 99% influence from A and 1% influence from B, the third child had 98% influence from A and 2% from B, etc.

16. Each pixel in a digital image has Red, Green, Blue and Alpha channel. Each channel contains values from 0 to 256.

17. The strategy is similar as using 2D image to create 3D Digital Elevation Model (DEM) in GIS. Large quantity of spatial information are classified and converted into the RGB values.

18. Paine's machine made sculptures with various forms driven by the mutation, similar as the identical

embryos can be developed into different cells. This genetic-transformation philosophy can be traced back to his earlier projects like *New Fungus Crop*, and *Amanita Field*. Although they were created with totally different approach (SCUMAK is Mechanical and *New Fungus Crop* is botanical), these projects share a similar configuration that is heightened by examine the relation between individual and group, the contrast between the various outcome and predefined procedure. In Paine's sculpture making machine, auto-mechanical and computer controlled manufacturing technology resembles nature such as Chaos theory, which was executed in the computer programming level to produce unpredictable geometric forms. See Tan Lin, "Grand Arts: Proxy Paine," Grant Arts,

<http://www.grandarts.com/exhibits/RPaine.html>

19. The chromosomes combine to form new pairs, which lead to a unique new individual with phenotypes inherited from both parents. Individuals with more adapted genotypes will survive in the evolution process while others will eventually be eliminated.

20. The swarm intelligence can also be described with a metaphor of a flock of birds (or fish). "The flock veers, dives, soars in a fairly uniform movement-uniform in the sense that each individual bird is more or less conforming to the overall pattern of the group. However, there is no leading bird taking control. Each bird is responding individually to those around it, obeying simple commands. The net result of these individual is logic of swarm behavior." Neil Leach, "Swarm Tectonics," *Digital Tectonics*, Wiley-Academy. New York: Wiley, 2004.

21. A common ancestor could pass its shared body plan to its offsprings which could be either a monkey, a human or a gorilla. In another word, by changing the proportion of the componets in the shared body plan, we can generate various types of creatures. As described by Manuel De Landa, "a kind of 'abstract vertebrate' which, if folded and curled in particular sequences during embryogenesis, yields an elephant, twisted and stretched in another sequence yields a giraffe, and in yet other sequences of intensive operations yields snakes, eagles, sharks and humans." Manuel De Landa. "Deleuze and the Use of the Genetic Algorithm in Architecture," *Design for a digital world*, Wiley-Acamemy. New York: Wiley, 2001, pp. 117-120.