Beyond Surface
Aspects of UVN world in Algorithmic Design

Aya Okabe
The University of British Columbia
ayaokabe@interchange.ubc.ca

Tsukasa Takenaka
The University of British Columbia
ttsukasa@interchange.ubc.ca

Jerzy Wojtowicz
The University of British Columbia
jerzy@post.harvard.edu

Abstract
The need for architects to develop their own computational tools is becoming increasingly evident. In this paper, we introduced our design tool named ‘UVN generator’ which is based on the algorithmic process combining scripting potentiality and flexibility of traditional 3D surface modeling. Our attempt on combining the two served us well to explore the new ground for design. New conditions were explored and observed in the three case studies which are named ‘on a surface’, ‘between surfaces’ and ‘on a new ordered surface’, referring to place where the scripts were run. In design projects presented in our case studies, we focus on the system behind the generation of complex, expressive, biomimetic, yet humanistic shape. This challenge to find a new ground for computational design enables us to pose our critical question ‘What could be algorithmic design potential may lay beyond basic surfaces?’.

1. Introduction
Introduction of algorithmic design process into architecture continues to produce interesting results. This type of architectural design “might be aligned with neither formalism nor rationalism, but with intelligent form and traceable creativity”. Algorithm is generally understood as a sequence of exact instructions proceeding from an initial state, through a successive transformation, up to its termination. Due to its prescriptive character, some see algorithmic method restricts the design creativity. However, the algorithmic design process is not necessarily predictable, or deterministic. Through the deployment of variables, the algorithm offers explosion of possibilities, while the explorations of randomness, or fuzzy logic can contribute to the new, unpredictable and rich design conditions. “The intellectual power of an algorithm lies in its ability to infer new knowledge and to extend certain limits of human intellect”.

Our paper builds on the initial premise of Kostas Terzidis that an algorithmic design radically differs from the conventional design. “The human designer may be constrained by quantitative complexity and may be unable to construct unpredictability since that would negate a designers intellectual control”. By reconsidering, challenging and extending algorithmically conceived surfaces in architecture we hope to contribute to the emergence of space-edge notion in new, algorithmic design. Combining scripting with traditional surface modeling we illustrate new and intuitive aspects of algorithmic design. We discuss our UVN geometry generator that facilitates running given scripts on any selected objects as well as related design method and illustrate its deployment in three schematic case study projects.

2. Methodology
2.1. Why UVN?
The need for architects to develop their own tools is increasingly evident. For those who translate design ideas into a precise geometric form using traditional modeling application there is a growing operational ease, but “behind the scenes” there are always series of scripts making it all possible. Scripts can easily repeat dynamics that compiles many components and complex relationships. However, when you generate architectural form with scripts, you need to write often complicated algorithm using exact mathematical formula. Sooner, or later, without broad understanding and engagement of algebraic, or mathematical discipline you confront your limitations in drawing your imagined, complex form. Apart from the possible “generate and test” strategy, to enter design critical loops we need to apply digital tools more simply and immediately.

Is there any way that we can use both application flexibility of making complex shapes and utilize potential of scripting? How these two could combine in the design process?

In response to those questions we proposed the idea of the “UVN Generator”. We named it for its main task of shape translation – from XYZ coordinate system into our UVN coordinate system. This is very important, as it facilitates running given scrips on the selected object using the UVN coordinate system.

With this hybrid approach, we can now generate extremely complex shapes in a familiar way without constraints; then, we can construct complex geometric relationships and parameter values on the object using scripts. Our assumption is that we can broaden shape possibilities by running scripts on the UVN coordinates usually running on XYZ coordinates. (Figure 1. A script running on XYZ coordinates) In other words, running scripts on the space edge that allows us to give new expressions. Furthermore, we could design space-edge as a ’space design’, not just ’surface design’ by adding N coordinate to UV coordinates.

2.2. Technique for UVN generator
The UVN coordinates are based on UV coordinates generated by a perspective projection system, where we applied the N coordinates as the normal direction from an arbitrary point on the UV surface. Using this notion, we developed the script for the purpose of translation from XYZ coordinates to UVN coordinates, and call it ‘UVN generator’. This is illustrated in the alignment of cubes script that is running initially on XYZ coordinates and then translated to UVN coordinates. (Figure 2. A script running on UVN coordinates)
3. Case Studies

We explored the three types of process applied to space-edge, a basic process 'On a surface', 'Between surfaces', and more advanced process which is based on the above two 'On a new ordered surface'. We will present those processes with a few variations indicating its potential for architectural design.

3.1. Case One : On a Surface

3.1.1 Proceeding with script

The first case study illustrates a process with series of scripts running 'on a surface'. After you form a space, the algorithmic process starts with the 'UVN Generator' working to translate points on your drawn space to points on the UVN coordinates. After those points data are stored in a point-cloud database, the programming can take next steps, 'Points relations and Unit design' for running scripts on a UVN surface.

The process can be seen in a demonstration project, Studium-X in Warsaw where a script was run on the toroidal form of enclosure surface. By running the script, the surface will be a media, not only a wall function itself, reflecting the imagination of architects grasping from dynamic, energetic and clamor images of a stadium. (Figure 3. Running scripts on a simple surface) (Figure 4. StadiumX-Warsaw by Tsukasa Takenaka, vdsEURO, 2007)
3.1.2 How to work scripts on a surface?
The ‘Point relations and unit design’ step offers many components and complex relationships to a surface by adding selected ‘unit function’ code. ‘Unit function’ in turn can store any of

![Diagram of point relations and unit design](image)

Point relations:
- \( P_t = \) Base point
- \( P_{t2} = P_t + U \) distance + \( N \) distance
- \( P_{t3} = P_t + V \) distance + \( N \) distance
- \( P_{t4} = P_t + U \) distance + \( V \) distance
- \( P_{t5} = P_t + \frac{U \text{ distance}}{2} + \frac{V \text{ distance}}{2} + N \) distance
- \( P_{t6} = P_t + \frac{U \text{ distance}}{2} + V \) distance
- \( P_{t7} = P_t + U \) distance
- \( P_{t8} = P_t + U \) distance + \( V \) distance

![Figure 5. Basic unit variations](image)

![Figure 6. Wall-study by Aya Okabe, UBC, 2008](image)
your 2D/3D unit drawings based on a point ‘Pt’ value that comes from point-cloud database. 
(Figure 5. Basic unit variations)

The test project is demonstrated below. Our intention was to make a curved wall surface into unpredictable and rich design condition using the simple ‘Leaf’ unit. The ‘Leaf’ unit is made by nine points connected by curves and surfaces.

The result is repeating and aggregating order with complex geometric relationships of units. The wall character becomes unique. Its screen like perforated surface is expressive, yet with intuitive softness of transitions found only in nature. It is radically different from of conventional, inorganic forms produced with predictable logic of conventional design. This biologically inspired system suggests potential parallel between design creativities and creations in nature. 
(Figure 6. Wall-study by Aya Okabe, UBC, 2008)

3.2. Case Two : Between Surfaces
The main difference between first and second case is the location where scripts are run. (Figure 7. UV coordinates and β coordinates) In Case One, scripts run directly on a single surface, while in Case Two, series of scripts run between two or more surfaces which are not necessarily identical, as they can have distinct shapes. The field where scripts run is defined by UV coordinates and a new β coordinates that has direction between surfaces, instead of N coordinates.

(Figure 7. UV coordinates and β coordinate)

(Figure 8. Structural-study by Tsukasa Takenaka, vdsEURO, 2007)
The second hypothetical case fosters communications and interactions between surfaces in design to create expressive surface behavior. The results was explored through a structural study illustrated in figure 8. *(Figure 8. Structural-study by Tsukasa Takenaka, vdsEURO, 2007)* In this study, the space between outside and inside was designed with ring shaped ‘structural elements’ and both inner and outer ‘facade elements’ which were designed using human scale components.

This type of process allows to integrate and combine structures with inner and outer facades. Since all elements and their relationships are determined by scripts, the rich and diverse range of possible structural solutions could be rapidly explored. The process is initiated by selecting both guiding surfaces, followed be generation of space structures.

### 3.3. Case Three : On a New Ordered Surface

#### 3.3.1 Embed new order on a surface

This final case study is based on two previously presented cases. In order to develop surface behavior further, we take following steps in the script design.

a. select any object
b. make UV grid with N or ß coordinates
c. give new order to the grid generated in step b

*Figure 9. A new ordered surface*

*Figure 10. Rhythm variations*

*Figure 11. A study for 'Biennale Pavilion' by Tsukasa Takenaka, 2008*
d. run a selected unit script on the new ordered surface
To achieve this, we developed a script for making ‘new ordered grid’ which introduced notion of rhythm to a surface. This function is controlled by any rhythm running on a UVN grid or a UVß grid by generation of deformation for each direction.

In this way, interesting results can be clearly observed in the difference between a normal UVN surface and a new ordered surface. Even the surface shapes and the script for running on those surfaces are the same, different expression of outcomes can be generated. (Figure 9. A new ordered surface) Besides, this process can embed a new order on any selected surfaces not only on simple surfaces, but also on a very complex one. This means, we can ultimately generate new ordered surface behavior beyond predictable.

The study project for ‘Biennale Pavilion’ was to derive hidden rhythm from the landscape and embed it on the surface for tracing out by our unit script. (Figure 10. Rhythm variations) It is of central significance for a sense of design process that this type of new ordered surfaces can respond to a rhythm from the site environment. This is particularly interesting point as outcome of design can have an underlying harmonious relationship to its site. (Figure 11. A study for ‘Biennale Pavilion’ by Tsukasa Takenaka, 2008)

3.3.2 Random-grid generation
On the other hand, there is a powerful potential for disorder, ‘randomness’ in design. Feature of randomness can have parametric range between order and disorder and control resulting appearance. Using those parameter, we can generate and express surface behavior beyond predictable. Another aspect of randomness is that random occurrence is depending on time. But using a database function to store the randomness as point-cloud automatically allows to reoccur the same randomness again. Therefore, with the technique of handling randomness, ‘randomness’ seems to become one of the materials we can now use in design extensively.

The roof study used randomness as one of the design element with a mathematical formula. The design intention for the project is to make a continuous roof located between urban and forested area. The study was started from demonstrating occurrence of randomness in U, V and UV directions. (Figure 12. Roof studies with random deformation) To controlled randomness occurrence more precisely, the project used mathematical formulas to control its occurrence.
The result of the study shows inspiration for a continuous transformation from a perfectly aligned ordered space to an ambiguous and expressive space that can connect and progressively negotiate between urban area and nature. (Figure 14. A roof-study by Aya Okabe, UBC, 2008)

4. Future Work
There are few central and potentially critical characteristics tangent to this method to be investigated further in the near future. The multiplicity of possible designs calls for exploration of ‘generate and test’ strategy in order to limit explosion of possibilities. With deployment of ‘Beyond Surface’ scripts as design tools there is a potential for interpretations of intuit biological process. Further investigation of nature inspired systems in architecture could lead to emergence of organic and potentially biomorphic forms and textures in our designs. Process of self organization, nested structures and patterns in a variety of programmed interpretations of intuit biological process needs much more work. The digitally controlled ordered surfaces responsive to and interactive with environment should also be more explored.

5. Summary and Conclusion
The need for architects to develop their own computational tools is becoming increasingly evident. For those who translate design ideas into precise geometric forms using traditional modeling application there is clear operational ease; yet “behind the scenes” there are always a series of scripts making it all possible. Architects like painters who develop their own style of brush, should step into managing their scripting language and customize own digital design tools. Our attempt on combining scripting potentiality and flexibility of the traditional surface modeling enables us also to explore a new ground - beyond the conventional surface.

We observed the results of surface behavior while constructing our projects. We, as designers, amplified our imagination using this computational method, and the surface could reflect and
restore our imagination to make it more expressive, biomimetic, and yet not without humanistic complexity. As the outcome, observers can be now aware of the fundamental essence and creative quality beyond the basic physical reality.

Apart from technical interests and functional possibilities, what attracted us was that algorithmically made surfaces gave us back to pay attention to its hidden sense behind its shape. As the results, it made ordered surface into the new ground, as well as, into the new design media reflecting environments and perhaps starting to connect architecture of natural and those of artificial.

6. Endnotes
1 i.e. book> see Terzidis K., Algorithmic Architecture, Oxford: Architectural Press, 2006, p.XII
2 i.e. book> see Terzidis K., Algorithmic Architecture, Oxford: Architectural Press, 2006, p.65
3 i.e. book> see Terzidis K. Algorithmic Architecture, Oxford: Architectural Press, 2006, p.103

7. References
Rutten D., Rhinoscript101, Robert McNeel & Associates