www.cadanda.com
Computer-Aided Design
AND APPLICATIONS

Notes on Generative Modeling, Procedural Symmetry, and Constructability of Architectural Design

Shen-Guan Shih

National Taiwan University of Science and Technology, sgshih@mail.ntust.edu.tw

ABSTRACT

In most CAD systems, geometric entities are the focused objects for operations. Procedural information regarding how designers build up their models is merely kept at the most primitive level, for implementing the undo/redo function basically. Tools are hardly provided for the manipulation of the modeling process. With generative modeling, designers model the process that generates shapes.

Generative modeling enables designers to record and reuse the procedural information that would otherwise be lost if CAD tools cannot recognize and manipulate the generative process of shapes. The lost information could have been useful to disclosing procedural symmetry for generating complex forms, and reducing the cost of design operation and communication. In this paper, it is speculated that generative modeling may facilitate knowledge integration in early design stages. Architectural design cases that use generative modeling to integrate structure and construction domain knowledge to increase constructability are displayed and discussed.

Keywords: generative modeling, procedural symmetry, constructability.

1. INTRODUCTION

Prior to the development of industry, manufacturing was often done at homes or workshops with hand tools or basic machines. Designers were also fabricators of their own works. When it was necessary, design was expressed as fabrication processes, such as recipes for cooking. As design and fabrication were separated into different jobs, designers created specifications to the artifact, and fabricators manufactured in respect to the specification. The work of a designer was shifted from the creation of artifacts to making drawings that specify the artifact. The situation eventually built up to shift the paradigm over design thinking and education, most conspicuously, in architecture. Architects are trained to be experts of drawing drawings, instead of building buildings.

Drawings are abstract representations of artifacts. Architectural design drawings such as plan, elevation and section are abstractions based on the geometric symmetry of buildings along orthogonal axes. When someone draws a column in a building plan, it is assumed that the profile of the column would not be varied in other heights, unless it is specified differently in other plans or sections. The invariance of shapes under axial translation is presumed in architectural design drawings and architectural design thinking. In other words, symmetry, which can be referred as unchanged aspects of shapes or systems under certain transformations, greatly accounts for the very reasons of why and how drawings are used as media for conveying what are in designers' minds to explicit forms.

The ruler and compass are tools to ensure the translational and rotational symmetry of shapes. Computer-aided design systems also create and manipulate geometric objects based on symmetric features of forms. Symmetrical forms such as spheres and cubes are used as basic elements to construct more complicated shapes. Commands for shape manipulation such as move, rotate, scale, offset, as well as commands for shape creation such as copy, extrude, revolve, array, sweep and the likes are all based on geometrical symmetry. Designing consists largely in the act of realizing the symmetric features of the form in mind, and the act of realizing what is in mind into drawings with commands and tools that are based on symmetry.

In this paper, procedural symmetry is regarded as a kind of higher orders of symmetry than that of geometry. Generative modeling systems enable designers to model the generative process of modeling based upon the realizing and breaking of symmetry in higher orders. The speculation is

Computer-Aided Design & Applications, 11(5), 2014, 518–525, http://dx.doi.org/10.1080/16864360.2014.902681

that generative modeling would be a powerful tool to downsize the communication cost across various domains of professions. In addition to theoretical arguments on issues regarding symmetry, information processing, and communication, examples on generative modeling are displayed for explanations.

2. PROCEDURAL SYMMETRY

Processes are regarding transformations. A process consists of a sequence of actions that transform things. Procedural symmetry is defined as the property of actions that are unchanged under transformations in space and time. In other words, it is the invariant pattern of actions that we can find within the process of making things. In professor Leyton's generative theory of shapes [8], shapes are thoroughly derived by productions of symmetrical transformations. Leyton speculated that transformation sequences that comply with "maximization of transfer" and "maximization of recoverability" are essential to human cognition and reasoning of shapes. According to the theory, the keeping and breaking of symmetry is fundamental to all intellectual behaviors concerning with the recognition and reasoning of shapes.

A straight line is symmetrical under the group of translations along its direction. The first derivative of a straight line is invariant everywhere on the line. The symmetry of a conical curve, such as the parabola $y = x^2$, is not limited to symmetry under reflection over the Y axis of the coordinate system. Although the tangents of the parabola are variable along the curve, yet the first derivative of the curve, which can be represented as $dy/dx = 2x$, is a straight line with translational symmetry when represented graphically. The second derivative of a conical curve is a constant, which implies that "translational symmetry" can be found on the curve in a higher order. Derivatives in differential calculus are regarding changes, and changes of changes, and so on to the endlessly higher orders of meta-changes. Following Leyton's theory, we speculate that procedural symmetry can be viewed as higher orders of symmetry on geometry, similar to the relations among the first, second and higher orders of derivatives to curves in differential calculus.

In turtle geometry [1] shapes are defined with sequences of actions. A square can be drawn by iterative actions of drawing a line segment and then turning right. Using the symbols d for drawing a line of one unit length and r^{90} for turning right with a right angle, we can represent the process of drawing a square as $(dr^{90}dr^{90}dr^{90}dr^{90})$. Imagining that the list of actions be extended infinitely on both ends, the process remains invariant by shifting any even number of symbols in either directions. The list of symbols is symmetrical under the transformation of time, or more precisely, under the group of transformations that is generated by shifting two actions. The list is also symmetrical under the groups of reflection either with an arbitrary d or r^{90} as the center. If we refer the 90 degrees rotation of a square to the action shifting of the generative process, and reflections on orthogonal axes and diagonals of the square to the reflections on d and r^{90} of the process, we can define a one-to-one mapping between each member of the dihedral group of D4 to the members of the symmetry group for the procedural representation of the square. In this case, the symmetry in the geometrical representation of a square is converted to the symmetry in the procedural representation of the same square.

Fig. 1 shows a rectangular spiral form that is generated with the process $(d^1 r^{90} d^2 r^{90} d^3 r^{90} \ldots d^n r^{90})$. The list is no longer symmetrical on shifting or reflection, because the exponent of d increases as the process goes further with time. If we let symbol concatenation to be a kind of non-commutative multiplication, and let the exponent of d's as a variable t, then the process can be represented as the product of a sequence, which can be written as $\prod_{t=1}^{n} d^t r^{90}$. We can derive the necessary transformation that transfers one item of the sequence to the next one by dividing the ith item $d^i r^{90}$ with the $(i-1)$th item $d^{i-1} r^{90}$ of the sequence. With the following calculation, we get the invariant d.

$$d^i r^{90} (d^{i-1} r^{90})^{-1} = d^i r^{90} r^{-90} d^{-i+1} = d^i d^{-i+1} = d$$

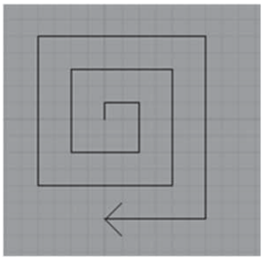

Fig. 1: A rectangular spiral form generated by $(d^1 r^{90} d^2 r^{90} d^3 r^{90} \ldots d^n r^{90})$.

In this case, the necessary transformation for transferring an action to the succeeding action is invariant throughout the generative process of the spiral form. We have found procedural symmetry in a higher order. Symmetry is the key to the reduction of information processing. Experienced drafts persons would learn that much work can be saved if they are able to disclose symmetry in the forms to be drawn. This might be the very reason why all CAD systems assist drafting by providing tools based on geometric symmetry. It is expected that CAD systems would be more helpful by providing tools to create and manipulate shapes on procedural symmetry in higher orders.

3. GENERATIVE MODELING AND PARAMETRIC DESIGN

Parametric design is concerned with procedural symmetry. In most CAD systems, procedural information regarding how designers build up their design models is merely kept at the most primitive level, for implementing the undo/redo function basically. Tools are hardly provided for the manipulation of processes. Script editor and interpreter are mostly provided for the extension and customization of the system; may not be properly engineered to be used by designers for modeling. Procedural symmetry is not recognized and recorded. As a model is getting more complicated, it would be much harder to edit. Generative modeling is an aged-old paradigm, within which processes are the focused objects of modeling, while shapes are by-products. It concerns "how" to generate the shape, rather than "what" the shape is composed of. Most CAD systems have adopted the "what" strategy instead of the "how" strategy for the modeling of buildings.

Generative modeling enables designers to record and reuse the information of shape generation that would otherwise be lost if the CAD tool can only manage what the designed shape is composed of. The lost information could have been useful to disclosing procedural symmetry, and reducing cost of design operation and communication. Generative Modeling Language (GML) [6] can be viewed as a 3D extension to PostScript, which is a programming language devised for specifying print layouts. As a kind of standard format, PostScript takes the advantage of being a fully expressive programming language that is interpreted upon printing. The data-exchange between print layout authoring tools and printers is extremely efficient and powerful. The efficiency of data-exchange between designers and 3D modelers are further realized by GML. It has been shown that with a GML file as small as 18kb in size, the interpreter can generate a 3D model of the Cologne Cathedral that consists of 70 tracery windows [5], and a single window in highest resolution contains about 7 million triangles. The GML file very efficiently encodes the information that is necessary for specifying the 3D model of the Cologne Cathedral. The efficiency cannot have been achieved without the disclosed procedural symmetry in the generative process of the complicated 3D form of the cathedral.

GML's ability for parametric design is another feature that makes it particularly apt for being used as a language for design modeling. For more examples, Generative component for Microstation™ and Grasshopper for Rhinoceros™ may also be regarded as being generative modeling based. They are tools frequently referred as parametric design systems. These systems enable designers with the ability to create shape variations by changing values of parameters. Parametric design has now been widely recognized as a kind of powerful tools for modeling curved and complicated forms. It is shown that GML has been used to define a large variety of parametric designs for buildings and products [5]. Fig. 2 shows a parametric model and its generative process of the Luce Memorial Chapel, a building designed by I.M. Pei. The model consists of10 parameters for the building mass with the curved envelop, and 11 parameters to specify the waffle panel structure. Fig. 2 shows the Grasshopper generative process. Fig. 3 shows the rendered images of the model generated with specific values for parameters so that the result resembles the actual building. The parametric model would have been helpful for the communication between the

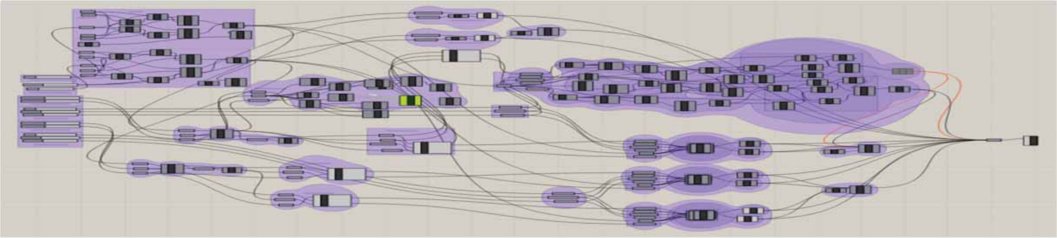

Fig. 2: A generative process that models the Luce Memorial Chapel in Taiwan.

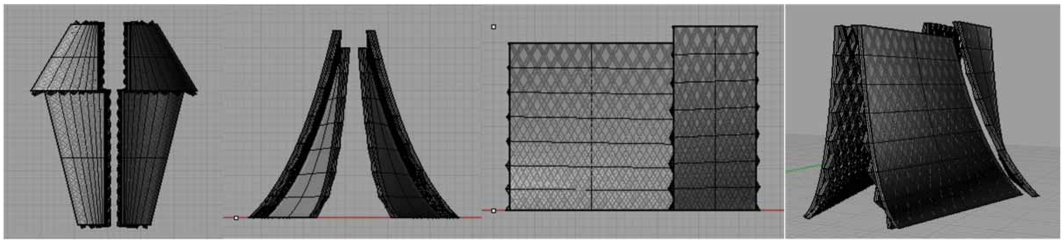

Fig. 3: The Luce Memorial Chapel model generated by the process.

architect, the client, and the structure engineer during the design process. The parametric model can be linked to Karamba, which is a software application for structure analysis, so that the designer can get prompt feedback upon any change to the parameters.

The cores of parametric shapes are the distinctions of the variants and the invariants in the generative process. The variants are the parameters that may be changed effortlessly in the design process, and the invariants are the processes that transmit parameters to the desirable form. Being apt for capturing procedural symmetry, generative modeling systems are by nature very powerful in defining and customizing parametric shapes. Here we use the turtle geometry notation defined in the prior section for further explanation. Mathematical function is a natural way to define processes that take variable input to perform more generic tasks. For example, squares of arbitrary sizes can be defined with the function $square(x) = (d^x r^{90})^4$, where x can be any positive number; the function $polygon(n) = (dr^{360/n})n$ defines the set of n-sided regular polygons; and the spiral form in Fig. 1 can also be defined as a recursive function *spiral(n)*, where

$$spiral(1) = dr^{90}, and\ spiral(n) = spiral(n-1)d^n r^{90}.$$

We can define another operator "+" so that enough number of turtles would be activated to perform all tasks that are separated with the "+" operator simultaneously. For example, the expression $(d + r^{90}d + r^{180}d + r^{270}d)$ would activate four turtles from the initial position, each of which would turn to the desirable direction and draw one of the four arms of the cross. The expression can be expanded from $(1 + r^{90} + r^{180} + r^{270})d$, where 1 is the identity element of multiplication and the turtle would simply do nothing when seeing this symbol. We can define a function to generate crosses with various sizes as

$$cross(x) = (1 + r^{90} + r^{180} + r^{270})d^x.$$

L-system [9] was devised to define biological forms, specialized but not limited to the generation of plant-like shapes. L-system can be regarded as a kind of generative modeling systems using rewrite rules to generate strings that are interpreted as geometric forms based on turtle geometry. With the above notation, we can define recursive functions that generate tree-like forms as rewrite rules in L-systems do. For example, the recursive function,

$$tree(1) = d,\ tree(n) = d^n(r^{90} + r^{-90})tree(n-1),$$

can be expanded to generate parametric tree-like shapes as shown in Fig. 4. We can evaluate the function to get the following instantiations with n equals to 2, 3 and 4:

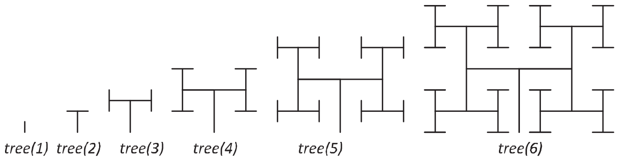

Fig. 4: Tree-like shapes generated with various values for the parameter *n*.

$$\begin{aligned} tree(2) &= d^2(r^{90} + r^{-90})tree(1) = d^2(r^{90} + r^{-90})d, \\ tree(3) &= d^3(r^{90} + r^{-90})tree(2) \\ &= d^3(r^{90} + r^{-90})d^2(r^{90} + r^{-90})d, \\ tree(4) &= d^4(r^{90} + r^{-90})tree(3) \\ &= d^4(r^{90} + r^{-90})d^3(r^{90} + r^{-90})d^2(r^{90} + r^{-90})d. \end{aligned}$$

If someone is to draw these shapes with geometric transformations such as copy, scale and rotate, symmetrical patterns of actions would be observed. Procedural symmetry can be defined by using functions as parameters to other functions. For example, the function $four(x) = (xr^{90})^4$ would generate a shape by executing the input expression x four times with a right turn inserted in between each pair of x's. For example, $four(dr^{90}dr^{90}dr^{-90}dr^{-90})$ would draw the shape shown in Fig. 5 (a).

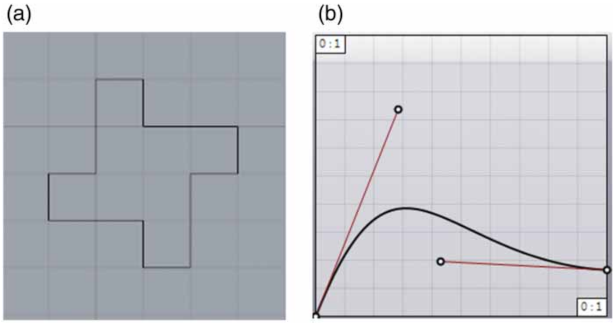

Fig. 5: (a) Left, The shape generated by $four(dr^{90}dr^{90}dr^{-90}dr^{-90})$. (b) Right, A cubic Bézier curve defined by 4 control points.

Parametric curves can be defined based on the same notation. The expression $(d^{\pi/n}r^{360/n})^n$ draws a unit circle as n approaches towards infinity. Modeling with procedural symmetry does not necessarily lead to trivial symmetrical forms. Symmetry exists in an infinite number of orders, much of which can hardly be recognized by even the smartest human minds. A cubic Bézier curve such as the one shown in Fig. 5 (b) is asymmetrical as it seems. The symmetry of the curve is yet to be uncovered after the third derivative of the three degree polynomial function that defines it. Many designers have the illusion

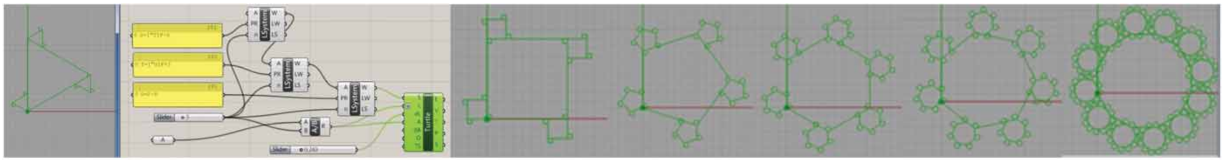

Fig. 6: Parameteric shapes created by a L-system.

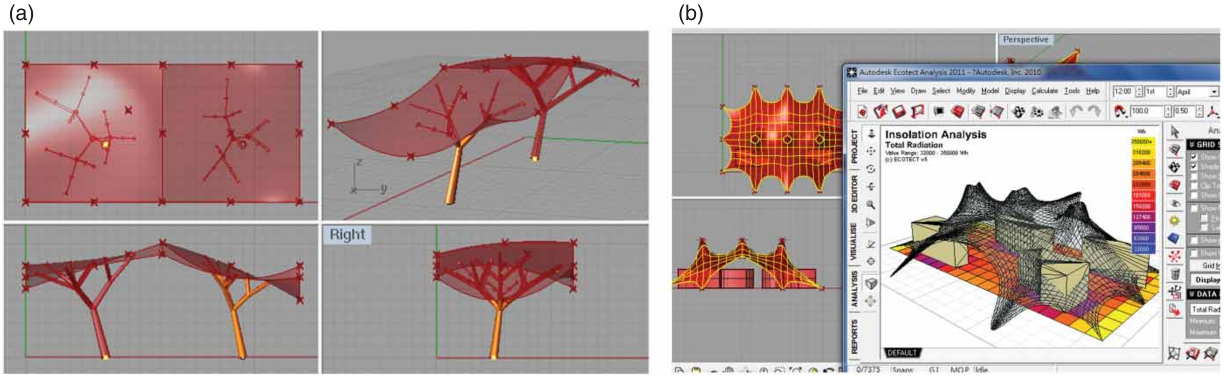

Fig. 7: Parametric shapes with generative modeling. (a) Parametric tree-like structure supporting a curve surface. (b) Parameteric design with tent structure linked to Ecotect for solar analysis.

that they can create and manipulate the so-called "freeform" with current CAD systems using B-splines. The illusion comes from the ignorance of symmetry in higher orders that lay behind the appearance. Designers must learn the lesson from mathematicians that symmetry can still be found within deviations that break symmetry. Once they do, the gap between design and construction would be down.

Fig. 6 and Fig. 7 show some examples of parametric design used as exercises in a design modeling course for undergraduate students in the department of architecture. Fig. 6 shows variations generated by a parametric shape defined with Grasshopper and Rabbit, which is an L-system extention to Grasshopper. Fig.7 (a) shows tree-like structures that automatically connect to the curve surface on top upon editting of parameters and the surface. Fig. 7 (b) shows parameteric design for simulating tent structure using Kangaroo, a kinetic simulator, and Gecko, that enables links to Ecotect for solar analysis.

4. INFORMATION, MESSAGE AND DESIGN COMMUNICATION

Design communication requires efficient media for exchanging information. In a paper that sets up the foundation for the theory of communication, Shannon [10] defines message as a sequence of recognizable patterns of signals that are transmitted with some communication channels, and information as the capability of differentiation enabled by the transmitted message, with which some certain situations can be identified from all possible situations that can be represented with the same set of message encoding. Shannon further suggested that information can be measured by calculating the uncertainty, or the unlikelihood, of the recognized situations with a logarithm function over probability. These definitions became the firm foundation for the development of the entire discipline of communicational theory in decades that followed.

Weaver [12] elaborated Shannon's theory by distinguishing three levels of communication in terms of the technical, the semantic, and the effectiveness. Research in architectural design communication concerns more with the semantic and effectiveness levels than the technical level. Although Shannon's discussion was focused on the technical level, Weaver concluded that the mathematical model proposed by Shannon can also be extended to the semantic and effectiveness levels of communication. As an example for communication within the semantic level, the amount of the information that is transmitted by a message consisting of the winning numbers for the lottery is much larger than the amount of information that is transmitted by a message of the same length consisting a number set that will not win. The message consisting of the winning set enables the message receiver to identify the very unlikely situation of winning the lottery. The message consisting of the other set does very little help by identifying one of the millions of ways that will not win, which is almost completely certain to the receiver even without the message. For design communication, drawings and models are messages that encode the necessary information for identifying the desirable design specifications between the sender and the receiver of the message. The lesson that we learned from Shannon is that the message sent by the designer does not need to include specifications that are certain to the receiver. The same amount of information can be sent by more compact message consisting of only the uncertain things that are unlikely for the receiver to derive without the message. Design communication between professionals takes this advantage so that compact representations such as abstract symbols and hatch patterns can be used to define construction specification, plan, and section to define 3D forms.

Operations on design drawings and models are costly and risky, for any change may induce cascades of necessary changes for maintaining the consistency of the entire set of construction documents, and any unresolved inconsistency may induce great losses in later stages of construction and operation. The organization of a design team has much to do with the cost for design communication. Design related works are divided by various disciplines of domain knowledge such as architects, interior designers, structure engineers, and other consultants. Each member of the design team would receive messages from other team members and feedback with messages that encode professional contribution. One of the difficult situations is that all contributions can be interdependent and vulnerable for changes from other parts of the design.

The structure of the team for a construction project minimizes the cost for interdisciplinary communication. A price to be paid is that the separation of design and construction in the industry has a negative effect on the constructability for the project. Building Information Modeling (BIM) [2] has been widely accepted as the bright avenue leading the industry out of the unsatisfactory situation. Most recognized is the promised land of Integrated Project Delivery (IPD), where project information can be exchanged across disciplines almost for free and retains its operability from source to destination.The magic is brought by advances of building information modeling platforms, with which the project team is supposed to integrate all necessary information including the form, the structure, the material, the MEP systems, and the construction altogether as a strongly interrelated data bank.The magic is further extended to every corner of the industry with Industry Foundation Classes (IFC), a universal file format for project information, and dialects that are developed through Information Delivery Manual (IDM) and Model View Definition (MVD) for exchanging information between specific professions within specific phases in the project lifecycle. [2]

An important issue needed to study is that how can the integrated project data bank be created and manipulated efficiently. In the backstage, how can the project information be encoded with a compact structure for storage, edit and retrieval, as well as for the maintenance of consistency? On the front stage, how can the authoring platforms of project information interact with project participants so that the rewards of integration outshine the increased work load for information processing? The standpoint is that generative modeling based parametric design may partially answer the front stage question, while the answer of the back stage question may need further insight to the fundamental theory of design modeling.

5. CONSTRUCTABILITY

Most construction projects are divided into phases, each of which is performed by teams of distinct professions. As professor Fischer [3] has stated, "*... The deliverables of one phase or discipline are often thrown over the wall to the participants performing the next task. Formal feedback loops usually do not exist ...*", in most construction projects, the desirable knowledge for constructability is often not available in early phases such as programming and design. However, the fragmentation of the construction industry could have been inevitable due to the inefficiency of interdisciplinary communication. Professor Fischer and his research team took up the challenge to formalize the knowledge for analyzing constructability of building design [4]. Recommended general contractors, designers, subcontractors, and suppliers of construction materials and equipment were interviewed for knowledge acquisition. The following design variables that contribute to the constructability of reinforced concrete structure were derived by the research team.

1. Dimensions of elements (e.g., height, depth, width, thickness and length)
2. Distances between elements (e.g., clear spans and story heights)
3. Changes in dimensions and distances (e.g., from floor to floor or from bay to bay)
4. Quantity and type of reinforcement
5. Concrete strength
6. Repetition of dimensions and distances, and modularity of layout

Among the six variables, the third, fourth, and the last are related to the geometrical symmetry of layout, element, and reinforcement of the structure system. Variables one and two are related to the compatibility of formwork systems that may greatly raise the constructability when applied properly to the construction. Most of such systems, such as tunnel form, flying form, and gliding form, are devised based on some symmetric features of the design. For example, the applicability of flying form would depend on the following three symmetric features of the structure system. First, as the flying form is moved out from the constructing structure after the concrete has been cured, the process requires at least a clearance with translational symmetry along the path based on the size and shape of the form. Second, the use of flying forms would be more efficient if they can be repeatedly used throughout the project. Third, the spaces between adjacent flying forms would require customized formwork if the void spaces are not of the same shape and size. It is obvious that the first, second, and sixth variables of constructability devised by Fischer can be characterized with the required symmetry described above. Other types of formwork systems might require other types of symmetry. An example is the jump form system used in a construction project in Hong Kong [11]. The formwork can be lifted floor by floor as the construction goes on to the top. It requires that the design possesses translational symmetry in vertical direction.

In early design phases, it is often unrealistic to analyze constructability of the design based on specific types of construction systems for that related decisions would not be set until later stages of the design phase. The analysis of constructability in early design phase requires reasoning on a higher level of abstraction. The five types of regular polyhedron, namely the tetrahedron, hexahedron, octahedron, dodecahedron and icosahedron, are of the highest symmetry among all polyhedron for that all vertices, edges and faces are interchangeable. There is no need of information to distinguish one vertex from another either in the design modeling or in the construction. If we are allowed to ignore asymmetrical factors such as gravity and weather, the constructability of a regular polyhedron would potentially be among the highest because all faces can be constructed with identical elements that are jointed with identical interfaces. Interchangeable vertices imply that if the form is constructed with linear elements along the edges of the polyhedron, the installation of all joints would probably take same machinery, information, skill and process to complete. Interchangeable edges imply that all panel elements can be installed in similar way. Interchangeable faces imply that same panel element can be used repeatedly and interchangeably. The symmetrical features of building design are strongly related to constructability. For building design with symmetrical shapes, dimensions and interfaces, systematic and prefabricated construction methods are more applicable; material and parts can be massively fabricated; engineers and workers take less effort in communication and control; tools, machinery and equipment can be reused without specific customization.

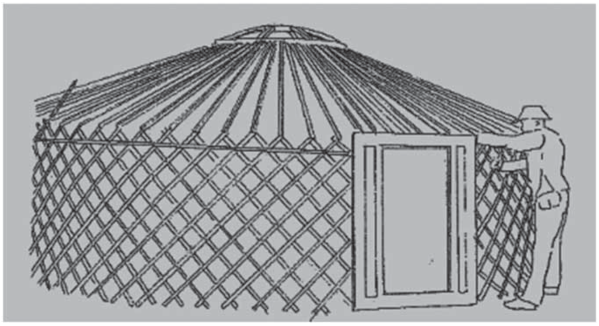

Fig. 8: A Mongolian Ger under construction.

The Ger, as a building type developed by pastoralists, for whom mobility is the very means to keep alive, is a good subject to learn about constructability. Fig. 8 shows a drawing of a Mongolian Ger under construction. The wall panels of a Ger are criss-crossed lattices that are made with sticks of the same size and length. The rafters for the roof are identical sticks notch into the top of the wall panels and the roof ring, in the same way for all of them. Procedural symmetry can easier be observed in the construction process. A family sized Ger can be assembled and made ready for living in half an hour by the family members. The form, the structure, and the construction are seamlessly integrated with their living environment and living style. It would be very hard, if possible, for any designer based on similar criteria to come up with better design alternatives that overtake traditional Ger by constructability. For modeling the Ger, a designer can take the most advantage out of the symmetrical form with symmetrical processes, and that is exactly what generative modeling is best of.

Fig. 9 shows two different ways to build an igloo. On the left, the igloo is built by laying snow bricks onto a sloped spiral curve to the top. The igloo on the right shows that snow bricks were arranged as layered circular rings on top of another with reduced sizes. According to a documented film in 1949 by D. Wilkinson [7], the Inuit build igloos with the spiral design. The circular design of an igloo might be more natural from a designer's standpoint, for its geometric symmetry is more apparent and is easier to draw and model with pencils and CAD systems. However, the spiral design of the igloo has the advantage over constructability, at least from the point of view of Inuit igloo builders. The spiral design sacrifices the

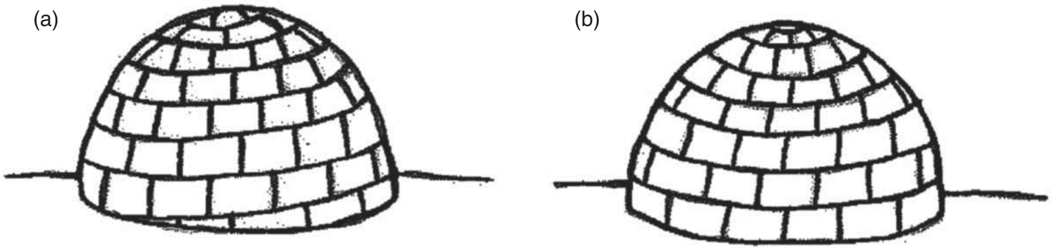

Fig. 9: (a)The spiral design of igloo. (b) The circular design of igloo.

geometrical symmetry of snow bricks for procedural symmetry. For every brick that is being laid, it is supported by bricks underneath and the brick on the down side of the spiral. With the circular design, procedural symmetry is broken whenever the first snow brick of each layer has to stand on the tilted top of the lower layer without getting any support from both sides. Procedural symmetry is broken again when the last snow brick of each layer has to be cut smaller for fitting into the space within blocks on both sides. It would require further effort and skill to ensure the air-tightness of the construction.

6. CONCLUSION

It is speculated that there might be relations between procedural symmetry in design modeling and constructability from the standpoint of information processing. For every task to be performed, be it an act for design modeling or for construction, there is some certain information that needs to be processed. Such information, which may either be derived from memory or from the working context, is required for appropriate actions upon the encountered situation. In our cases with turtle geometry, the only required information for the turtle to act is the input symbol that is being given to it right at the moment. The turtle needs no sensors to get information from the context, nor does it need to memorize anything. It is not difficult to define more sophisticated turtles that are capable of sensing the environment, or memorizing what they have done, and making decisions autonomously.In the field of computational theory, classes of autonomous devices such as finite state machines, push-down automata and Turing machines were defined based on levels of complexity for information processing. Although the classification was used to analyze computational complexity for tasks that are defined with mathematical constructions, it could also be applied to real world tasks as well. In design modeling or construction, work processes are composed of individual tasks performed by some actors such as drafts persons, designers, workers, engineers or robots. Actors in the design and construction process are analogous to the autonomous devices in computational theory. It is required that these actors are able to retrieve and process the necessary information so that they can make the right decision over what and how to perform the task, in ways similar to the autonomous devices that take information from input and process it for output. Constructability can be classified by the complexity of the information processing that is required for job site activities.

Procedural symmetry implies that there exists some ways to distinct the invariants from the variants so that complicated processes can be decomposed into simpler tasks that are symmetrical in the way information is retrieved and processed. Symmetry is the key to simplicity and complexity. Generative modeling, as a platform to formalize processes, could be used as an adequate design media for designers to build up design models based on higher orders of symmetry for better constructability.

ACKNOWLEDGEMENT

We gratefully acknowledge the support of the National Science Foundation, Taiwan, R.O.C. and the National Taiwan University of Science and Technology for the research presented.

REFERENCES

[1] Abelson, H.; diSessa, A.: Turtle Geometry, The MIT Press, 1986.

[2] Eastman, C. M.: BIM handbook: a guide to building information modeling for owners, managers, designers, engineers and contractors (2nd ed. ed.). Hoboken, N.J.: Wiley. 2011.

[3] Fischer, M.: Automating Constructibility Reasoning with A Geometrical and Topological Project Model, Computing Systems ni Engineering, 4 (2–3), 1993, 179–192.

[4] Fischer, M.; Tatum, C.B.: Characteristics of Design-Relevant Constructability Knowledge. Journal of Construction Engineering and Management, (1997) 123(3), 253–260.

[5] Havemann, S.; Fellner, D.W.: Generative Parametric Designof GothicWindow Tracery. The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, 2004.

[6] Havemann, S.: Generative Mesh Modeling, Doctoral Dissertation, der TechnischenUniversität-Braunschweig, 2005.

[7] How to Build an Igloo: http://www.nfb.ca/film/How_to_Build_an_Igloo/. National Film board of Canada

[8] Leyton, M.: A Generative Theory of Shape, Springer-Verlag Berlin Heidelberg, 2001

[9] Prusinkiewicz, P.; Lindenmayer, A.: The Algorithmic Beauty of Plants (The Virtual Laboratory). Springer-Verlag, 1990.

[10] Shannon, C.E.: The Mathematical Theory of Communication, in The Mathematical Theory of Communication, University of Illinois Press, 1949

[11] The Jump Form System: http://www.cityu.edu.hk/CIVCAL/production/advanced/jump_form.html, City University of Hong Kong.

[12] Weaver, W.: Some Recent Contributions to the Mathematical Theory of Communication, in The Mathematical Theory of Communication, University of Illinois Press, 1949