Title:
Developing Diagram-Based Computation for Algorithmic Modeling of Parametric Architectural Design

Authors:
Chieh-Jen Lin, t60011@mail.tut.edu.tw, Tainan University of Technology

Keywords:
Design Intention, Intention Diagram, Intention driven, Algorithmic Modeling, Algorithmic Framework

DOI: 10.14733/cadconfP.2020.348-352

Introduction:
The algorithmic design has become a new paradigm of architectural education and practice. Through the computational processes created by designers, performance simulation replaces the styling presentation, and the techniques move from the production of objects towards the generation of integrated systems [1]. Programmable algorithmic modeling tools provide the visual script editor that allows designers unfamiliar with programming to experiment with various theories and methods in a digital sandbox. Regardless of the generative, parametric or algorithmic design methods, all are the application of different levels of algorithms. For architectural design, the generative design focuses on generating forms by rules but doesn't necessarily involve issues other than geometric shapes. The parametric design implies algorithms have been defined and fixed and focuses on manipulating complex forms by modifying parameters. The algorithmic design applies existing algorithms or develops new algorithms to solve specific problems, such as applying genetic algorithms to optimize multiple objectives. No matter for parametric, generative, or algorithmic approaches, the applying algorithms for solving specific problems, rather than the intentions proposed by designers, drives the thinking and manipulating processes of algorithmic design.

Architectural design is not only the process of defining and solving problems, but also generating and learning knowledge to solve problems. Innovative intentions and beliefs often go beyond the scope of known problems that have existing algorithms for solving. Cognitive researches point out that the use of algorithmic modeling tools must play the dual role of designers, and the user's attempts usually are limited to geometry exploration [7]. Due to the lack of connection with design intentions beyond geometry and performance, parametric design often causes criticism of limited creativity and poor quality [2]. Since architectural problem-solving must be validated through geometric forms. Digital architecture is therefore often simplified into generating forms by software. In addition to generate forms, algorithmic modeling tools have the potential to explore "problem definitions" and "solving algorithms." The algorithm can not only input problem information as parameters and output building attributes as variables, but also can import design intentions of problem-solving to avoid the issue of “garbage in, garbage out.” Regardless of generating complex forms, multi-objective optimization, and predictions and evaluations of performances, all are well-defined design problems with known algorithms. Therefore, the algorithmic design is usually to apply visual scripts to implement known algorithms, rather than to openly interpret intentions and derive the steps for satisfying the intention. How to develop algorithms to input design intentions, such as sites’ contexts, functional requirements, building codes, aesthetic criteria, and other “soft data” [6] as parameters, based on specific design beliefs to derive building attributes, such as mass, façade, circulations, and other variables of a building, are still lacking sufficient investigations.

This paper proposes the approach for developing intention-driven algorithms by applying the narrative diagrams [5] which can visually narrate what design intentions are proposed, and how the intentions are reached rather than what only the geometric forms are generated. As the data-driven approach taking the data as input parameters, and the performance-driven approach taking the performance as output variables of building forms, the intention-driven approach of this paper aims to assist designers to represent their intentions as input parameters and to validate generated models whether satisfying the proposed intentions or not. By integrating semantic ontology [3] and applying topological algorithms of an algorithmic framework [4], this paper aims to help architects to associate the architectural knowledge with the algorithmic process of parametric design.

Main Ideas:

Famous architects apply diagrams as visual narratives of their design intentions where their proposals come from [5]. Those diagrams visually narrate the design stories, which usually are a serial procedure of geometric manipulations that response design issues such as sites' contexts, functional requirements, building codes, and aesthetic criteria. Apart from the descriptive texts and the associative colors, the narrative capabilities of diagrams are based on the gradual transformation of the geometries and introductive symbols, such as the lines, and arrows in the serial diagrams. The introductive symbols narrate how the geometries in diagrams were modified to respond the proposed intentions. For example, a straight-arrow presents the intention of retracting the building's mass in order to make spaces for outdoor activates, or a curved arrow presents the intention of reorienting the building's facade in order to face specific contexts such as open horizons or attractive landscapes. However, those diagrams usually are only symbolic representations of design intentions rather than actual generating procedures, in other words, the algorithm of generating geometries.

Unlike the "hard data" such as sunshine, rainfall, wind, and other climate data applying in the performance-driven approach, the design intentions proposed by architects may be the personal interpretations of "soft data" mentioned above. Architects have traditionally relied on sketches or diagrams to express their intentions and refine relevant proposals of these issues. But those diagrams usually only the representations of finalized ideas, rather than an operable tool for exploring possible results and refining proposals. Since the algorithmic modeling tools like Grasshopper can input geometries as parameters, it is possible to develop algorithmic modeling which is driven by the geometric representations of design intentions.

However, the overlapping previews of Grasshopper usually cannot easily distinguish relevant parameters and generative steps, and the "baked" geometries in Rhino inevitably lose algorithmic information. The narrative diagrams approach of generative algorithms, therefore, was proposed in the previous study in order to help architects for representing their modeling intentions [5]. By integrating semantic ontology of the algorithmic framework entitled STG*f* developed in previous studies [4], and applying visualizing algorithms, the narrative diagram approach can help architects to associate modeling intentions with the algorithmic process of parametric design. Based on the narrative diagram approach, this paper proposes a computational approach for helping architects to develop intention driven algorithms of parametric architectural design. Beyond the narrative of generating geometries, the intention driven approach aims to associate narrative diagrams with soft data mentioned above.

Diagrams as Parameters of Design Intentions

Just as architects prefer to express design intentions in sketches, most "soft data" applied in design intentions is possible to be visually represented by graphics or geometries like the introductive symbols in the narrative diagrams. For example, points can be used to represent existing trees' locations in the site, lines to represent the directions of attractive landscapes, curves to represent the human circulation, and closed outline to represent the intended outdoor space remained in the site. Those representations are easy to be understood by architects and to input as geometric parameters of generative algorithms in Grasshopper. With the help of semantic modules of the STG*f*, it is easy to attach semantic annotations with geometries for narrating the semantic relations of design intentions.

Therefore, the challenge becomes how to extract geometric features of those representations for developing available algorithms.

How can the points of trees' locations shape outdoor spaces within the site? How can the directional lines of attractive landscapes reorient the building's façade? If there are clear definitions of conceptual sketches provided by the architects, it may be easy to script algorithms for those who are familiar with Grasshopper based on those sketches. The scripting tasks still is difficult for architects who are not so familiar with Grasshopper, and the algorithmic ideas in those sketches may also be full of ambiguity. However, the graphic representations of design intentions provide the basic geometric features for further developing algorithms in order to satisfy those intentions.

Diagrams as Representation of Generative Algorithms

The visual narratives of diagrams are not only based on annotation tags, but also the geometric features of the introductive symbols. A point can easily be thought of as the center of forces or fields, a line as the direction and intensity of the forces, and a curve as rotation or bending of the forces. Architects therefore use these symbols to express their intentions of applying those forces to refine the proposals for responding design issues. Based on the features of geometries, three basic generative algorithms are proposed: (1) a circle around a given point represent the attraction or repulsion algorithms and the strength of the attraction or repulsion (Fig. 1), (2) a straight line and its length represent the offsetting or bending algorithms and its strength parameter (Fig. 2), and (3) the curve with segmented points present the twisting algorithms and their strength parameters (Fig. 3).

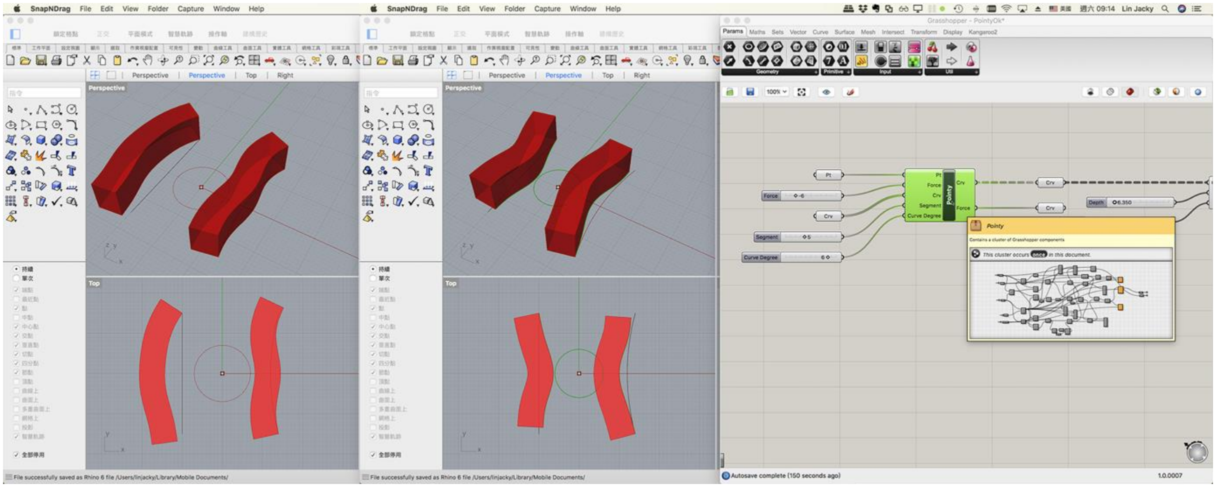

Fig. 1: Basic pointy diagram algorithm of the attraction/repulsion.

Unlike textual scripts can be easily interpreted themselves by naming parameters, functions, and classes, most graphic scripts of generative algorithms usually can only be recognized their behaviors and validate their results through the manipulations of input parameters. But three proposed algorithms can intuitively and visually predict their operating behavior and possible results by the geometric features of parametric diagrams. Since not all parameters are represented as operable geometric objects in Rhino, those diagrams therefore not only represent the parameters of the algorithms, but also the visual clues of their algorithms.

Proceedings of CAD'20, Barcelona, Spain, July 6-8, 2020, 348-352

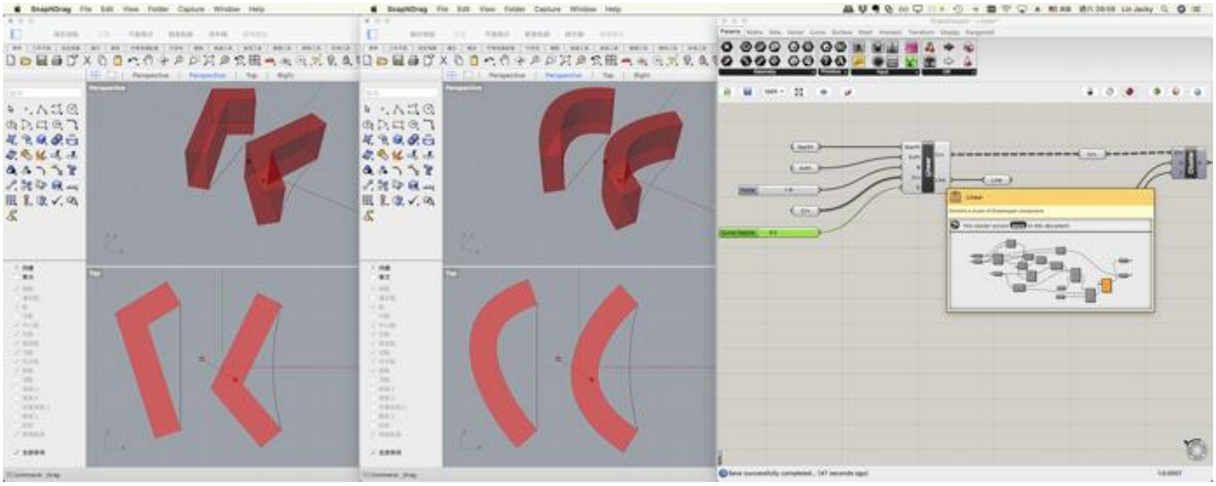

Fig. 2: Basic linear diagrams of the offsetting or bending algorithms.

Diagrams as Computable Design Intentions

The original purpose of developing narrative diagrams is to make the algorithmic idea of generative modeling easy to understand and communicate. In the case of architectural design, diagrams usually must narrate not only what the generated geometries are, but also how the algorithms generate the geometries. Although the narrative diagram approach proposes multiple narratives for help architects to exploring possible results by simultaneously generating results of different parameters. For narrating more complex intentions and associating architectural design criteria, however, it is therefore necessary to develop more complex narrations such as a serial of narrative diagrams. By combining three proposed algorithms, it is able to develop more narrative diagrams and more complex generative algorithms for specific design issues. For example, the curve algorithm is implemented by dividing a curve in order to generate points for placing the attraction/repulsion algorithms along the curve, and the visualization of the attraction/repulsion strength parameters is implemented by generating meta-balls from those parametric circles of the attraction/repulsion strength.

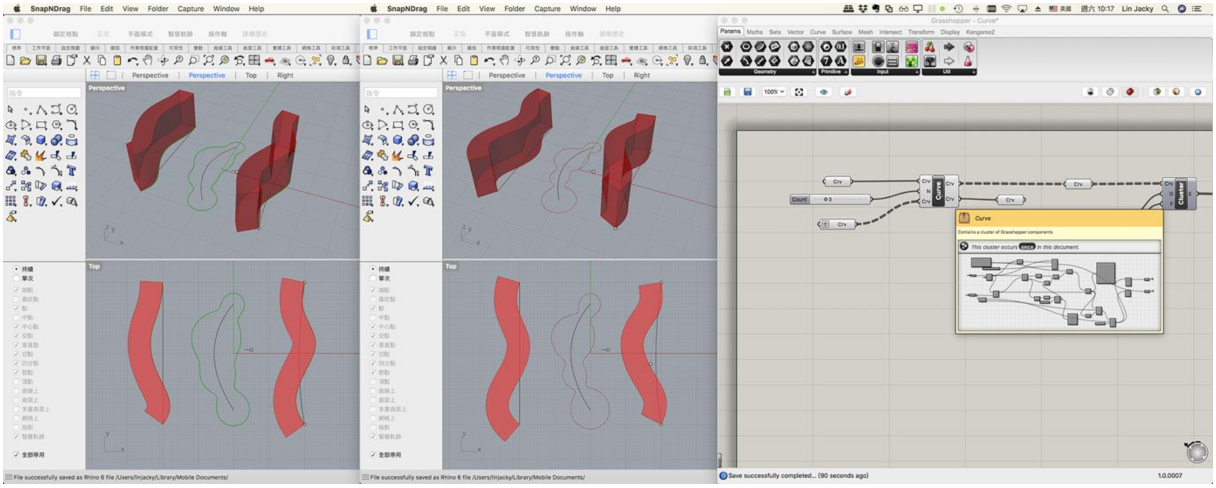

Fig. 3: The curve-twisting algorithms that places the pointy attraction/repulsion algorithms along the curve and visualizes the attraction/repulsion strength parameters by meta-ball curve.

Through combining three proposed algorithms, such as combining the pointy attraction/repulsion algorithms and the linear strength algorithms to generative curves for arranging attraction/repulsion strengths, this paper provides a visual approach for developing diagram-based computation for algorithmic modeling of parametric architectural design. By hocking the semantic modules of STG*f*, those parametric diagrams can easily associate with relevant design issues such as site contexts and other soft data applied in the intentions.

Conclusions:
Since the design intentions and their "soft data" may be the personal interpretations proposed by architects, how to represent them in order to communicate with stakeholders becomes even more important than the "hard data" approaches such as simulations of building performance. Although styling and aesthetics may be the essentials for winning an architectural design competition. However, the proposed building forms by architects often go beyond purely aesthetic attempts. Diagrams are a traditional and popular means for architects to narrate the design stories behind their proposals. However, those diagrams usually only the visual representations of finalized intentions rather than computable algorithms. Algorithmic modeling provides an opportunity for converting the graphic narrative of diagrams into the operable and computational representation of generative algorithms. By applying three proposed algorithms as the basic words or phrases for narrating design intentions, this paper demonstrates how to developing diagram-based computation for helping architects to explore possible results and available algorithms for validating their intentions.

Acknowledgments:
The Ministry of Science and Technology of Taiwan supported this paper under the 2019 research reward program of the Tainan University of Technology.

References:
[1] Ahlquist, S.; Menges, A.: Computational Design Thinking, in: A. Menges, S. Ahlquist (Eds.) Computational Design Thinking: Computation Design Thinking, John Wiley and Sons, London, 2011, pp. 10-29.
[2] Coyne, R.: What's wrong with parametricism, http://richardcoyne.com/2014/01/18/whats-wrong-with-parametricism/,
[3] Khan, M. T. H.; Demoly, F.; Kim, K.-Y.: Formal Ontology and CAD Integration with Macro Parametric Approach, Computer-Aided Design and Applications, 14(1), 2017, 24-32. https://doi.org/10.1080/16864360.2017.1308078
[4] Lin, C.-J.: The STG-framework: A Pattern-based Algorithmic Framework for Developing Generative Models of Parametric Architectural Design at the Conceptual Design Stage, Computer-Aided Design and Applications, 15(5), 2018, 653-660. https://doi.org/10.1080/16864360.2018.1441231
[5] Lin, C.-J.: Developing Narrative Diagrams for Algorithmic Modeling of Architectural Parametric Design, Computer-Aided Design and Applications, 17(2), 2019, 447-457. https://doi.org/10.14733/cadaps.2020.447-457
[6] Ottchen, C.: The Future of Information Modelling and the End of Theory: Less is Limited, More is Different, Architectural Design, 79(2), 2009, 22-27. https://doi.org/10.1002/ad.845
[7] Yu, R.; Gero, J.; Gu, N.: Architects' Cognitive Behaviour in Parametric Design, International Journal of Architectural Computing, 13(1), 2015, 83-102. https://doi.org/10.1260/1478-0771.13.1.83