Title:
Topological Vision: Applying an Algorithmic Framework for Developing Topological Algorithm of Architectural Concept Design

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

Keywords:
Parametric Architectural Design, Algorithmic Modeling, Geometric Intention, Topological Relationship, Algorithmic Framework

DOI: 10.14733/cadconfP.2018.220-224

Introduction:
Parametric design is defined as the exploration of the associative relationships among geometric intentions [9]. With the growing popularity of algorithmic modeling tools, apart from their use in generating complex geometries, however, these tools have also inspired new models of architectural design thinking and strategies. According to Oxman, as an evolutionary result of algorithmic thinking and scripting, parametric design thinking is located at the intersection of three types of knowledge, which are the cognitive model of architectural design, the process model of digital design, and construction order of fabrication design [7]. At the early and conceptual design stages, which usually do not yet involve material and construction requirements, the associative relationships between architectural design knowledge and algorithmic processes of digital design become more critical for parametric architectural design.

Oxman indicates two types of cognitive model in architectural design, namely typological and topological knowledge, and suggests that the topological versions and visions represent a seminal theoretical and operative methodological concept in parametric design thinking [7]. Unfortunately, algorithmic modeling tools do not help users to associate architectural design knowledge, either typological or topological, with information processes of involving geometric intentions. Oxman concludes that knowledge of how to manipulate and explore the associative relationships and dependencies of topological geometries is a prerequisite key to parametric design thinking [7]. However, these manipulations and explorations must rely on algorithmic thinking and scripting the skills and techniques. At a time when parametric design and algorithmic modeling tools are becoming important, scripting and tool-making skills and knowledge should therefore be a core of architectural design education and practice.

One of the reasons for the popularity of algorithmic modeling tools is that most algorithms, including complex geometries formulas [6], metaheuristic multiple-objective optimization algorithms [10], and building performance prediction and evaluation formulas [8], were developed and validated in relevant disciplines. However, cognitive research has revealed that designers prefer to apply algorithms only as means of exploring geometric intentions, and tend to apply known solutions and design patterns in the case of non-geometric intentions [11]. Since algorithmic modeling was developed to accelerate 3D modeling tasks by applying algorithms, designers unsurprisingly prefer to apply known solutions rather than to develop or implement algorithms by themselves. However, if the design intentions represented by those known solutions and design patterns, especially when involving topological design knowledge, can be converted into algorithmic models, parametric design should be more useful as a means of exploring non-geometric intentions.

Based on the algorithmic framework termed STG*f* [5] proposed in previous studies, this paper applies this framework to helping architects for modeling topological algorithms by inputting geometric intentions. Through the assistance of rewritable example scripts and adjustable algorithmic modules, this paper aims to help architects to associate topological knowledge with algorithmic processes of parametric design.

Main Ideas:
Based on the STG pattern proposed in a previous study [4], which is the semantic-topological-geometric conversion pattern of BIM information schema [1], the STG*f* framework can be used to implement conceptual design algorithms by applying Grasshopper and the GhPython plugin as an algorithm-aided design tool [5]. By dividing conceptual design algorithms into three parts: (1) a semantic module helping architects to indicate geometric objects and to infer their semantic relationships; (2) a topological module that recognizes and validates the topological relationships of input geometric objects; and (3) a geometric module that manipulates, and visually validates the definitions of semantic and topological modules. By providing rewritable example scripts and adjustable topological modules, which are editable clusters of topological algorithms in Python language, this algorithmic framework aims to help architects to develop topological algorithms by inputting their geometric intentions at an early architectural design stage.

An algorithmic framework like STG*f* can not only serve as a design assistant thein development of algorithmic models, but can also provide a method of converting design knowledge into algorithms. The basic idea of the STG*f* framework involves three development steps in the form of a loop; these steps consist of definition of semantic models, development of topological controllers, and the final step of validation of geometric views. However, since the three algorithmic modules corresponding to these steps have been separated in the STG*f* framework, it is possible for a user to begin from any one of these modules. As studies have indicated, designers prefer to apply algorithms when exploring geometric intention [11]. And it has also been found in previous studies involving similar approaches that users usually applied geometric representations of design intentions, which are sketches or diagrams of design concepts, before they retrieved or applied relevant design criteria [3]. This paper therefore proposes another approach to the application of the STG*f* framework to the conversion of topological knowledge and criteria derived from the geometric intentions.

Geometric Features as Representation of Design Intentions
The initial purpose of the geometric module in the STG*f* framework was to demonstrate how to input geometric objects from Rhino into semantic and topological modules, and then to provide visual clues for the validation of users' design intentions. For users who have design concept sketches or diagrams, the geometric module of the STG*f* framework can be applied to help users to retrieve design knowledge based on geometric features.

Architects generally use sketches and diagrams of design concepts as a means of representing and communicating their intentions. In this context, the conventional method of retrieving design knowledge usually involves protocol analysis of designers' recollections. Thanks for the powerful processing ability of modern tools, it is very easy nowadays for users to photo or scan their sketches and diagrams into modeling tools such as Rhino. However, this approach can not only convert 2D analog images into 2D/3D digital geometries, but also provide computable geometric representations of design intentions. Since designers prefer to apply known solutions and design patterns in the case of non-geometric issues [11], further steps will be needed to analyze and retrieve non-geometric intentions and design knowledge from these representations.

Semantic Ontology as Parametric Schema of Design Intensions
At early design stages, design intentions usually consist of abstract, textual descriptions concerning various design objects and their relationships. Although essential semantic information regarding building components has been predefined in the BIM and Industry Foundation Classes (IFC) schema, however, designers' intentions should not be limited to the domain of these schemas. But Rhino has no predefined semantic schema concerning building information, not to mention definitions outside BIM or IFC. In addition, free and open definitions of a semantic schema may encounter conflicts with similar identity names. A contextual semantic ontology of design intentions is therefore needed to allow architects to define and interpret their unique objects and relationships [3].

A semantic ontology is a computational format for representing, storing, and validating a semantic ontology of domain knowledge. The initial purpose of the semantic module in the STG*f* framework was to apply ontological techniques to assist designers to capture the semantic logic of abstract design knowledge and intentions, and thereby establish parametric schema for validating algorithms. By hooking input geometric features with a defined semantic ontology of design knowledge and intentions, the STG*f* framework can help architects to associate their abstract design knowledge with geometric intentions in order to develop topological controlling algorithms for exploring non-geometric intentions.

Topological Algorithms as Generative Controllers of Design Intentions

A topology consists of the mathematical relations among different objects, and is the critical information in the validation of the conceptual consistency of design intentions. Because there is no unanimous definition of necessary topological information in the AEC domain, the BIM and IFC schema both ignore most of the topological information concerning relationships of different building components [1]. Especially at the early design stages, however, architects usually care more about spatial topologies, such as adjacency, overlapping, surrounding, and separation, et al. [2], than other types of mathematical relationships. Unlike generative algorithms focusing on generating geometric forms, there should be at least two kinds of topological algorithms for a given spatial topology, which are validating and adapting algorithms. A validating algorithm should be able to validate whether the input objects are consistent with a given topology or not. An adapting algorithm should be able to modify the input objects in order to satisfy a given topology.

More study is needed concerning how to automatically modify input objects in order to satisfy a given topology, however, and dealing with a topology that involves more than two objects still faces technological challenges. The initial function of the topological module in the STG*f* framework therefore only provides example scripts of validating algorithms for basic spatial topologies. By visually validating results based on defined semantic ontologies and input geometric features, this paper proposes topological vision as an assistant to help architects for developing algorithmic models of parametric architectural design (Fig. 1).

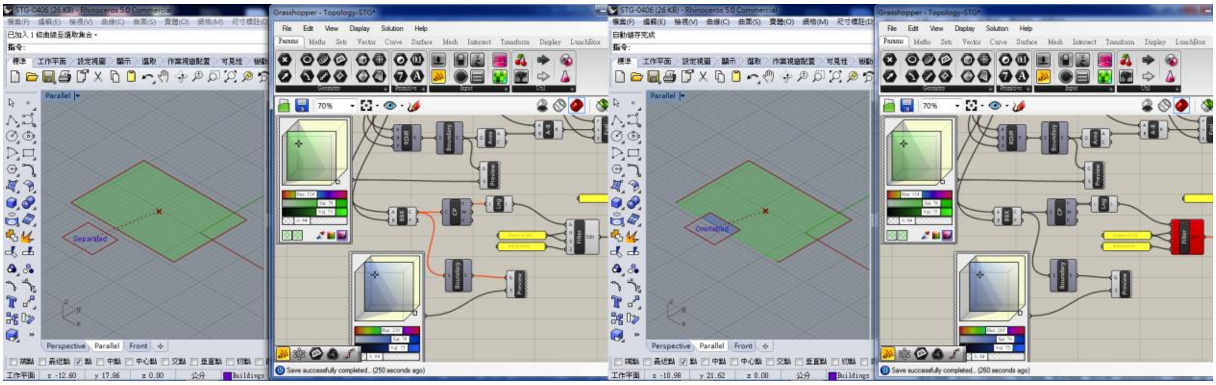

Fig. 1: A demonstration of topological vision involving geometric intentions.

Initial Testing and Evaluation of Topological Vision

In the previous studies, three recent architect qualification examinations in Taiwan were examined for encoding abstract design concepts in the early design stages. Since the contexts of the three sites specified in these examinations had no geometric features for retrieving relevant design criteria, there were no obvious clues that would constrain candidates' geometric intentions. Apart from for the interior contexts of the sites, such as existing trees, the "community center on a historic street" featured in the 2017 examination provided explicit geometric contexts next to the site (Fig. 2). A row of baroque-style, one-story, classic street houses is located along the west street (Fig. 2b), and a temple to the local land god is located on the east street (Fig. 2a).

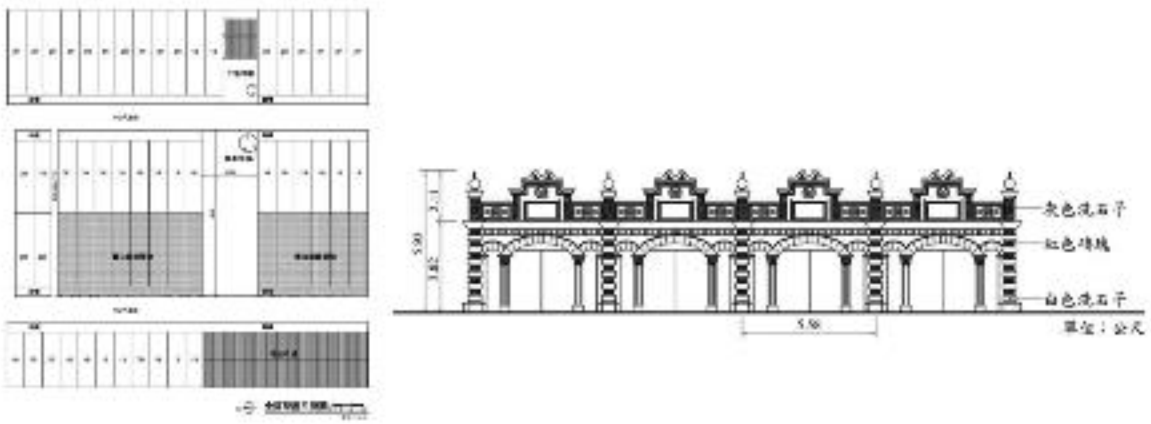

Fig. 2: The site contexts on the 2017 architect qualification exam in Taiwan: (a) a temple to the local land god is located on the east street, and (b) a row of baroque-style, one-story, street houses is located along the west street.

The baroque street houses suggest that the façade design along the historic street should reflect the geometric patterns of these classic street houses, which are 5.58 meters wide and 5.93 meters high. After inputting the geometric features of the existing street houses, trees, and the temple into the STG*f* framework, STG*f* can help architects to explore their topological intentions involving the given geometric features and the site contexts at an early design stage, including such aspects as the outdoor spaces shaped by a building and its surroundings, which are usually ignored in the BIM and IFC information schema. In addition, arrangement of the topological relations among the geometric features of the community center and its existing contexts so that they facilitate community activities still leaves much room for interpretation by architects.

Conclusions:

As mentioned above, designers prefer to apply known solutions and design patterns when expressing their non-geometric intentions. STG*f* seek to help architects convert topological knowledge embodied in known solutions and design patterns into algorithmic models for exploring further possible solutions. By providing rewritable example scripts and adjustable algorithmic modules, which are editable clusters of algorithmic components in Grasshopper, this paper aims to help architects associate topological knowledge with the algorithmic process of parametric design.

Acknowledgements:

The Ministry of Science and Technology of Taiwan supported this paper under grant number MOST 106-2221-E-165-001-MY2.

References:

[1] Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K.: BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd ed., John Wiley & Sons Inc., Hoboken, N.J., 2011. https://doi.org/10.1002/9780470261309

[2] Lin, C.-J.: Architectural Knowledge Modeling: Ontology-Based Modeling of Architectural Topology with the Assistance of an Architectural Case Library, Computer-Aided Design and Applications, 12(4), 2015, 497-506. https://doi.org/10.1080/16864360.2014.997647

[3] Lin, C.-J.: Design Criteria Modeling - Use of Ontology-Based Algorithmic Modeling to Represent Architectural Design Criteria at the Conceptual Design Stage, Computer-Aided Design and Applications, 13(4), 2016, 549-557. https://doi.org/10.1080/16864360.2015.1131551

[4] Lin, C.-J.: The STG pattern – application of a "Semantic-Topological-Geometric" information conversion pattern to knowledge-based modeling in architectural conceptual design, Computer-

Aided Design and Applications, 14(3), 2017, 313-323. https://doi.org/10.1080/16864360.2016.1240452
[5] Lin, C.-J.: STG-framework: A Pattern-Based Algorithmic Framework for Developing Generative Model of Parametric Architectural Design at Conceptual Design Stage, in: Proceedings of the CAD Conference and Exhibition 2017 (CAD'17), CAD Solutions, Okayama, 2017, 288-292. https://doi.org/10.14733/cadconfP.2017.288-292
[6] Miller, N.: Parametric Strategies in Civic Architecture Design, in: Proceedings of the 29th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA 2009), Chicago 2009, 144-152.
[7] Oxman, R.: Thinking difference: Theories and models of parametric design thinking, Design Studies, 52, 2017, 4-39. https://doi.org/10.1016/j.destud.2017.06.001
[8] Roudsari, M.S.; Pak, M.: Ladybug: A Parametric Environmental Plugin for Grasshopper to Help Designers Create an Environmentally-Conscious Design, in: Proceedings of the 13th International conference of Building Performance Simulation Association, Chambery, France, 2013.
[9] Woodbury, R.: Elements of Parametric Design, Routledge, New York, 2010.
[10] Wortmann, T.; Nannicini, G.: Black-Box Optimisation Methods for Architectural Design, in: 21st International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2016), Melbourne, 2016, 177-186.
[11] 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