Frontiers of Architectural Research (2017) 6, 53–62

Higher Education Press

Available online at www.sciencedirect.com

Frontiers of Architectural Research

www.elsevier.com/locate/foar

SOUTHEAST UNIVERSITY

RESEARCH ARTICLE

Evolutionary approach for spatial architecture layout design enhanced by an agent-based topology finding system

Zifeng Guo*, Biao Li

School of Architecture, Southeast University, Nanjing 210096, China

Received 7 June 2016; received in revised form 22 October 2016; accepted 7 November 2016

KEYWORDS
Spatial allocation problem;
Layout planning;
Evolutionary strategy;
Multi-agent system;
Computer-aided architectural design

Abstract
This paper presents a method for the automatic generation of a spatial architectural layout from a user-specified architectural program. The proposed approach binds a multi-agent topology finding system and an evolutionary optimization process. The former generates topology satisfied layouts for further optimization, while the latter focuses on refining the layouts to achieve predefined architectural criteria. The topology finding process narrows the search space and increases the performance in subsequent optimization. Results imply that the spatial layout modeling and the multi-floor topology are handled.

1. Introduction

Building layout design is regarded as one of the major tasks in architecture design. It determines the shapes, dimensions, and positions of internal building spaces to satisfy architectural criteria. This task becomes complicated for human designers when the topology relationships of rooms are complex. Thus, the automated design technique attracts significant attention because it may be a potential way for researchers to find novel solutions for spatial design or quickly plan complex architecture programs, such as hospitals, schools, and laboratories. The idea of introducing such a technique does not mean replacing architects, but rather developing powerful tools to find fast solutions, verify ideas, and choose the best tool for furthering design.

The automated layout design problem is referred to as a space allocation problem in the field of computer-aided architectural design. Various approaches represent architectural layouts as two-dimensional (2-D) geometries and multistory buildings as the combination of multiple layers. Galle (1981) adopted a grid system for layout representation and proposed an approach to enumerate all arrangements of rooms within the grid. The exponential growth in the number

*Corresponding author.
E-mail address: 1281224531@qq.com (Z. Guo).
Peer review under responsibility of Southeast University.

http://dx.doi.org/10.1016/j.foar.2016.11.003

of possible arrangements makes this approach infeasible. A similar grid-based strategy can be found in Rosenman (1996), Lopes et al. (2010), and Peng et al. (2014). The methods adopted for layout generation are different in these approaches, which range from genetic algorithms (Rosenman, 1996) and procedural modeling (Lopes et al., 2010) to integer programming (Peng et al., 2014). The latter two have addressed the generation of multi-story building layouts, where vertical spaces, such as stair cases, were also involved. However, in Peng's approach, these spaces were manually positioned instead of optimized by computer. Another model widely used for layout representation is 2-D polygons. Michalek et al. (2002) represented rooms as 2-D rectangles and adopted mathematical optimization to decide the positions and the aspect ratios. Martin (2006) represented rooms as rectangles, and the positioning process in his approach was carried out by procedural modeling. Neither of these approaches addressed the multi-floor layout. Similar strategies can also be found in Goetschalckx and Irohara (2007), Afrazeh et al. (2010) and Ramtin et al. (2010). Doulgerakis (2007) generated internal layouts from the polygon that represents the boundary of the building. Rooms were generated by dividing the original polygon into small parts, and the division operations were encoded using genetic algorithms. Multi-floor layouts were addressed in this approach, but vertical spaces, such as a staircase and a two-floor-high living room, were not mentioned. Rodrigues et al. (2013a, 2013b, 2013c) and Merrell et al. (2010) also achieved the generation of multi-floor layouts. Staircases were addressed in the former, while other vertical spaces, such as two-floor-high living rooms, were addressed in the latter.

The above investigation indicates that discrete and continuous models are common for layout representation. Most approaches adopt a 2-D layout representation. Approaches that directly modify three-dimensional (3-D) spaces are rare. This trend may be explained by conventional and technical reasons. First, generated layouts represented by 2-D geometries can easily be converted into common architectural drawings. Second, 2-D computer graphics algorithms are stronger and faster than 3-D algorithms. Generating 2-D layouts is faster because the rooms in most buildings have the same height and can be directly extruded from the plan. However, such strategy may limit the solution space when designers intend to find novel combinations of building spaces. The strategy also encounters difficulties in dealing with situations such as multi-story buildings with different room heights or vertical rooms that cross several floors. Thus, we proposed an approach for spatial architectural layout design to directly modify architectural spaces. Our approach to the modeling of spatial layout improves the efficiency of topology finding. The major challenges of this work include (1) introducing a model for spatial layout representation, because few approaches directly modify architectural spaces, and (2) guaranteeing the efficiency of the program, because the number of possible arrangements in 3-D graphics is greater than that in 2-D ones. These challenges are tackled through the combination of a multi-agent system, which is inspired by Li (2012), and an evolutionary strategy. The former enhances the latter by providing it with topology-satisfied layouts. The latter, which is based on a grid system, improves the layouts to satisfy user-specified architectural criteria.

2. Agent-based topology finding

Unlike other constraints involved in the design process, topology relationships are fundamental requirements for layout designs. Changes in the dimensions or shapes of rooms do not lead to unusable rooms. A certain degree of redundancy exists for the entire layout to accept such deviations. However, compared with improper sizes or shapes, the missing connections between rooms lead to incorrect topology relationships and are disastrous for the layout design. The chance of this outcome happening grows higher when the number of rooms increases. Moreover, starting with layouts that have proper topology relationships can narrow the search space and may improve the efficiency of the evolutionary process. Thus, the purpose of introducing the topology finding process is to reduce the probability of incorrect topology relationships that occur in the latter optimization process by providing it topologically satisfied layouts.

2.1. Agent model

The topology finding process is implemented through a multi-agent system, where rooms are represented as bubble-like agents and connections as strings linking the agents. Bubble agents are divided into two types based on the type of room. Fig. 1 shows that the first type is a sphere-like agent that consists of a center point and a buffer distance. The second type is a capsule-like agent that consists of a segment and a buffer distance. Common rooms are represented as sphere-like agents and linear spaces, such as corridors, staircases, and rooms, which are represented as capsule-like agents across several floors. For horizontal capsule-like agents, the endpoints of their segments are affected by other agents. Thus, the endpoints can adjust their length and orientation automatically during the interaction between agents (Fig. 2). For vertical capsule-like agents, which usually represent staircases and tall rooms, the endpoints of its segment are

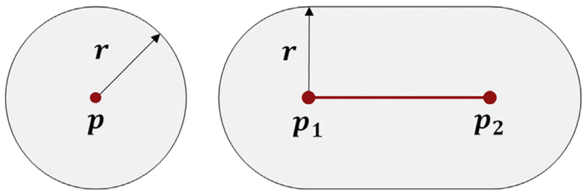

Fig. 1 Left: sphere-like agent. Right: capsule-like agent.

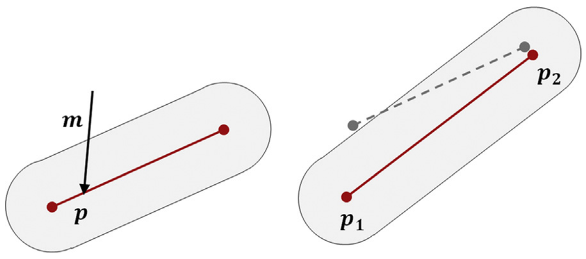

Fig. 2 Effect of the push operation on a horizontal capsule-like agent.