A Human–Machine Collaborative Building Spatial Layout Workflow Based on Spatial Adjacency Simulation

Ximing Zhong[1](✉), Fujia Yu[2], and Beichen Xu[3]

[1] Aalto University, 02150 Espoo, Finland
ximing.zhong@aalto.fi
[2] Tampere University, 33100 Tampere, Finland
yufujia@tju.edu.cn
[3] University of Liverpool, L69 7ZN Liverpool, UK
B.Xu14@liverpool.ac.uk

Abstract. The space layout of a reasonable modular building prototype is a time consuming and complex process. Many studies have optimised automatic spatial layouts based on spatial adjacency simulation. Although machine-produced plans satisfy the adjacency and area constraints, people still need further manual modifications to meet other spatially complex design requirements. Motivated by this, we provide a human–machine collaborative design workflow that simulates the spatial adjacency relationship based on physical models. Compared with previous works, our workflow enhances the automated space layout process by allowing designers to use environment anchors to make decisions in automatic layout iterations. A case study is proposed to demonstrate that the solution generated by our workflow can initially complete different customised design tasks. The workflow combines the advantages of the designer's decision-making experience in manual modelling with the machine's ability in rapid automated layout. In the future, it has the potential to be developed into a designer-machine collaboration tool for completing complex building design tasks.

Keywords: Spatial adjacency simulation · Physical model · Responsive design process · Human–machine collaborative workflow · Real-time visualization

1 Introduction

The modular building spatial layout is a task involving many complex decision-making activities [1]. Designers need to spend a huge amount of time and make a great effort to consider the dimensions of the building modular and all possible relationships within and between the buildings. In addition, they need to dynamically modify the prototypes of individual modules and layout combinations to meet the user's changing needs and complex spatial adjacencies [2].

Since the 1960s, the rule-based approach has been widely used for automated spatial layout [3]. Several rule-based studies translate adjacency [4], density [5], and topological

© The Author(s) 2023
P. F. Yuan et al. (eds.), *Hybrid Intelligence*, Computational Design and Robotic Fabrication,
https://doi.org/10.1007/978-981-19-8637-6_2

and geometric constraints [6] into mathematical constraint objectives to optimize layout solutions for spatial automation. Based on 'Rule-based', various classical studies treat automated building layout and prototyping as a multi-objective optimisation process. However, not all design requirements can be easily translated into explicit mathematical constraints [7]. Moreover, due to the partial errors in adjacencies sometimes arising and the weak interactivity of the layouts generated by this method. It is hard for designers to make decisions in automatic layout iterations [8].

Many studies regarding automatic layout based on the physical model show the potential to simulate the human–machine collaborative design process, thus resulting in the advantage of being highly interactive [9]. Based on these methods, some studies demonstrate the feasibility of simulating the spatial layout by analogy spring [10] and gravitational and repulsive forces [4] to satisfy complex adjacency demand constraints. However, the results of the physical model calculations still require a significant amount of manual modelling work and filtering to complete customized architectural design tasks. In their studies, designers are mainly limited to filtering and modifying machine-generated plans after simulating but not adjusting the results via directly changing the physical model during the machine iteration. Many studies on human–machine collaboration in recent years have demonstrated the great advantages of human–machine collaboration. Humans make decisions based on experience; machines can solve problems quickly and automatically, with each side playing to its strengths [1]. Impacted by this, the aim of our workflow is:

1. Utilizing physical models based on simulating spatial adjacency relationships to automate spatial layouts to meet customized design tasks' needs (adjacency, area, geometric relationships, etc.)
2. Enable designers to manually control the nodes interactively. Designers can set and move the environment anchor in the system to determine the final layout generated by machines. The workflow can solve different prototype design tasks (PDT) initially.

2 Related Works

In 1989, Baraff proposed an approach to the automatic layout of space by simulating and correcting physical collisions with physics, an interactive process [11]. In 2002, Arvin automated the conceptual design process by applying the physics of motion to the elements of spatial planning [10]. This approach provides a responsive design process. In 2010, Hao, H. and Ting-Li, J presented the optimization of adjacency relations for floating bubbles as an analogy to spatial layouts. By assigning two primary forces to agents: attraction and repulsion, it achieved a responsive layout of spaces with more complex adjacency requirements constraints [4]. While in 2016, EISayed further proposed a theoretical approach to simulate the interactive layout process of collision and reorganization between the modified space and other prototype spaces that the designers provided in advance by simulating mechanical springs [8]. Furthermore, it demonstrated its highly feasible and interactive in the experiments used by Egyptian lay workers for custom modifications. However, 1. Conflicting adjacencies and geometric constraints sometimes make the system arise with conflicting partial layouts that need to be redone manually by architects. 2. Even though automated layouts based on physical layout models are more interactive than purely algorithmic simulation, it is still hard for designers to modify the physic model during the iteration.

Therefore, we are trying to explore a responsive building layout workflow based on space adjacency simulation with physical models by introducing the concept of environment anchors. Environment anchors in our paper are objects in modelling software that architects can define for different design tasks, such as sun, orientation, building main core, corridors, and public spaces.

3 Method

As shown in Fig. 1, the proposed workflow is divided into four major parts. I. User Input Component (UIC): Designers enter the Matrix of Space size and Matrix of Adjacent relationship (data of various spatial demands and complex adjacency requirements). II. Stimulating Component: Use physical models to achieve automatic layout by simulating the adjacency relationship between units. IV. Visualization Component: The computer provides real-time feedback with the results and evaluations (space adjacent, average distance) to designers. III. Designer Adjustment Component: Designers can interactively change the layout through moving environmental anchors, combing manual modifications with machine-generated solutions to generate an automated layout that satisfies the adjacency relationship.

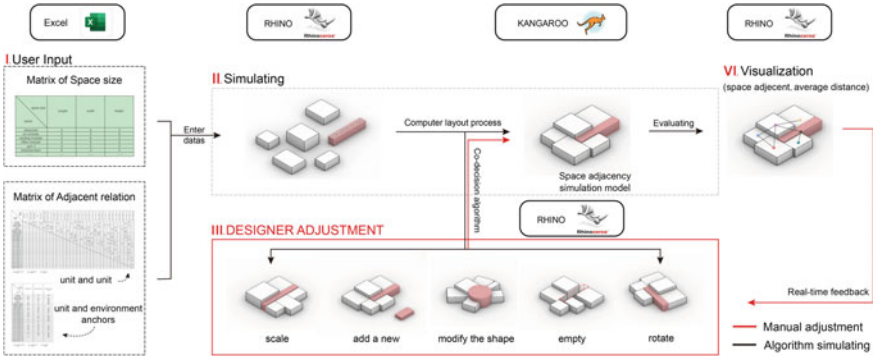


Fig. 1. The framework of human–machine collaborative workflow

3.1 UIC Component (Unit Size and Adjacency Matrix)

Unit Size Input: Designers record the size requirements for each unit by length, width, and height in an Excel file shown in Fig. 2a. These data are transferred to the Rhino modelling platform and then converted into a 3D physical model using Grasshopper. As Fig. 2b shows, each unit is generated automatically.

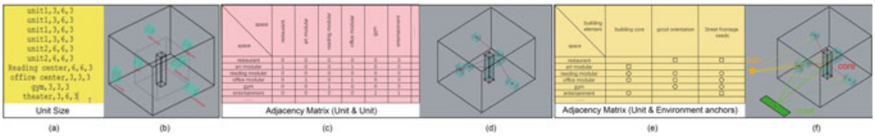


Fig. 2. UIC and preliminary results

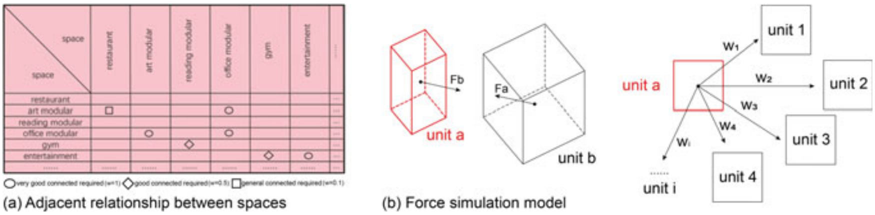
Adjacency Matrix Input: Designers record different adjacency requirements in the adjacency matrix. Figure 2c shows the adjacency relationship between units, and Fig. 2e shows the relationship between units and environment anchors. Moreover, the different shapes represent w = 0.1, w = 0.5, w = 1. In our case, parameter w is a constant from 0 to 1, which represents the degree of adjacent requirements between the different modules.

Designers need to set these adjacencies manually. When the setting is completed, the system converts the adjacency into lines between spaces, as shown in Fig. 2d, f. The adjacency relationship lines will then be put into the modules to simulate forces. Different adjacency relationships correspond to different weights of attraction and repulsion. When the adjacent relationship becomes stronger, the more increasing corresponding force performs.

3.2 Simulating Component

3.2.1 Spatial Adjacency Relations Simulating by Physical Model

We first introduce the simulation of the spatial adjacency relationship between all units. Arvin. S and House. D present a prototype for simulating spatial layout [10]. Different axial gravitational or repulsive forces are exerted between the two units depending on their different adjacency requirements. For unit a, the force on it in the system is equal to the sum of the forces of all the objects that have an adjacency requirement in Fig. 3b. Specifically, the combined force of the unit a is equal to the gravitational force of each object multiplied by the weight w (Fig. 3a) and then summed. It is noted that the initial gravitational force values in the system are all 1 N. The designer can control which units previously had a stronger attraction by adjusting w to control the final layout. Furthermore, we use the Kangaroo plugin to automatically simulate the force and layout of units based on data we can read from the UIC.



space \ space	restaurant	art modular	reading modular	office modular	gym	entertainment	......
restaurant							...
art modular	□			○			...
reading modular							...
office modular		○		○			...
gym			◇				...
entertainment					◇	○	...
......	......	......	......	......	......	......	...

○very good connected required (w=1) ◇good connected required (w=0.5) □general connected required (w=0.1)

(a) Adjacent relationship between spaces

(b) Force simulation model

Fig. 3. Layout simulation based on adjacency between unit and unit by physical model

We then introduce the simulation of the adjacency relationship between units and environment anchors. In our workflow, environmental anchors can be defined as any design-related custom factor, such as the sun direction, the position of the main core, street frontage needs, main core, corridors, platforms where units are located and other factors that can influence the layout of the building. Meanwhile, we set these anchors as the ‘object instance’ in the system. Hence the designer can be manually moved, deformed, and turned (Fig. 4) by any modelling software. It provides a good basis for combining manual modelling and automated scheduling workflows. Designers try to solve layout problems with environment anchors. Then the machine will automatically fill in the solution based on the adjacency relationship.

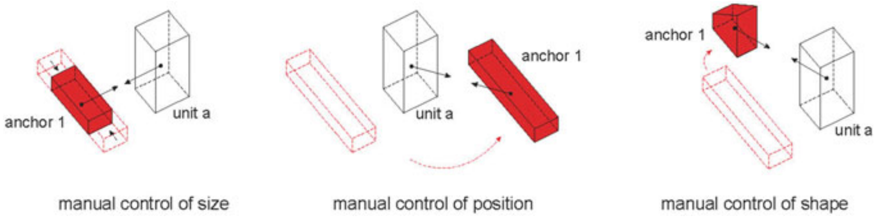


Fig. 4. Operability of environment anchors

Hao, H. and Ting-Li, J proposed a prototype of disjoint objects that generate repulsive forces when objects overlap and separate [4], ensuring that the space is geometrically constrained in the layout process. Based on their works, we use collision modules in Kangaroo to enable objects to be non-intersecting.

3.2.2 Simulating Process

The system automatically simulates the spatial layout based on UIC data and manual settings. Figure 5 shows a simulating process from random distribution (a) to the association (b) to aggregated state (c). In addition, the red lines in the diagram demonstrate the adjacent relationship between units that have adjacent requirements. A shorter length indicates higher value attractiveness and more optimized adjacency. The sum of spatial adjacency line distances can be seen as a parameter to measure and optimize the spatial adjacency of a solution [12]. As Fig. 5d shows, the curve shows how the sum of distances changes during the optimization process which can help designers to monitor the optimizing phase. The sum of the geometric distances among boxes with spatial proximity requirements tends to decrease over time and level off as the space reaches a more compact state. The system will give designers a completed calculated solution when the curve becomes stable.

3.3 Visualization Component

Turner proposed the Mean Shortest Path Length, which quantifies the accessibility of every location in the spatial system [13]. Based on his study, we define a colour range to represent different values of adjacency relationship to show the real-time average shortest path length visualisation. The lower values indicate closer adjacency and more

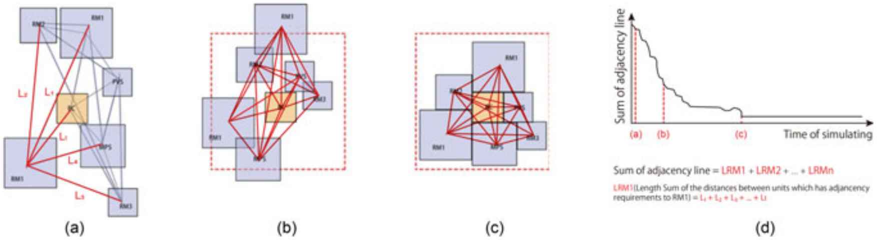

RM2
RM1
PVS
K
L₂
L₁
L₃
L₄
L₅
MPS
RM1
RM3
(a)
(b)
(c)
Sum of adjacency line
Time of simulating
(a) (b) (c)
Sum of adjacency line = LRM1 + LRM2 + ... + LRMn
LRM1(Length Sum of the distances between units which has adjancency requirements to RM1) = L₁ + L₂ + L₃ + ... + Li
(d)

Fig. 5. Dynamic change of adjacency lines distance and optimization phase in simulating process

optimised coloured cool colours (Fig. 6), whereas higher values are hot colours. Users can consider and minimise this total distance to improve overall fitness. Designers can adjust and re-optimise the specific unit space in real-time by manual adjustments when getting a redpoint.

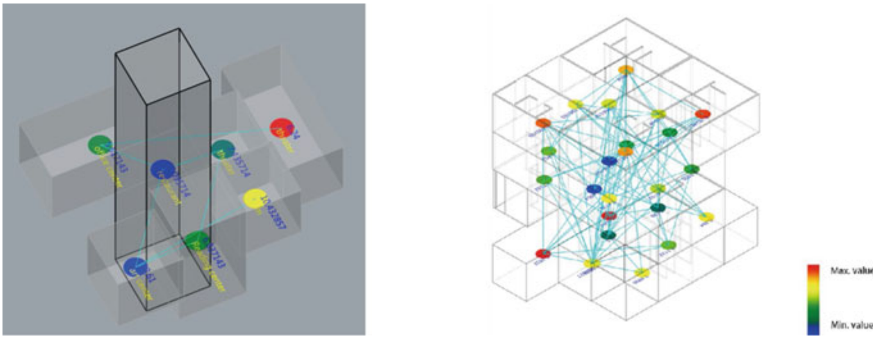


Fig. 6. Diagram of 3D spatial adjacency visualization

3.4 Designer Adjustment Component (Illustration of Interface)

During the simulation process, designers can manually move the different environment anchors to achieve different spatial layouts. Figure 7 shows an illustration of the designer adjustment interface. Users can change the scale, position, and shape of environment anchors' objects in the system. In addition, they can preview a real-time layout result of the unit's relocation under gravitational and repulsive forces after the adjustment. This workflow has been deepened into a tool called Autocat which is available on Food 4 Rhino.

4 Case Study

The Hong Kong social house project competition is selected as our case study. This is an architectural competition project completed by Archiford LTD in 2018 to satisfy complex area and spatial adjacency requirements by using a modular design approach.

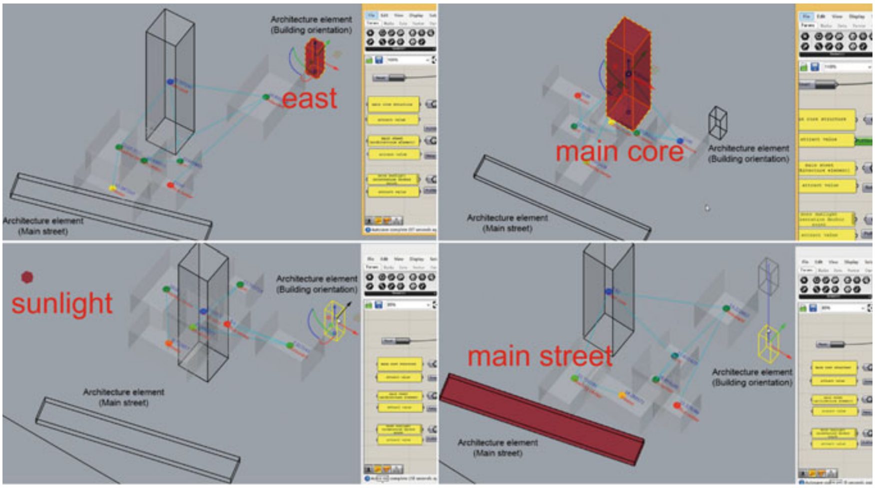


Fig. 7. Illustration of designer adjustment interface

We are trying to test our workflow based on the research and prototype design offered by Archiford LTD. In this case, we attempt to set different goals to verify whether our workflow can generate layouts that comply with the adjacency and set constraints. In addition, trying to verify if it is possible to guide the machine to generate the expected spatial layout by designers' manually adjusting the environment anchors.

Experimental rules: We expect each group to have a different layout and intention (details will discuss in the following setting section). We define different rules and expect these different rules to combine human decision-making and automated layout to decide the final layout results jointly.

Experimental Settings: Fig. 8a shows the base household type, and Fig. 8.b, c shows two sets of adjacency relations (unit and unit & unit and environment anchors).

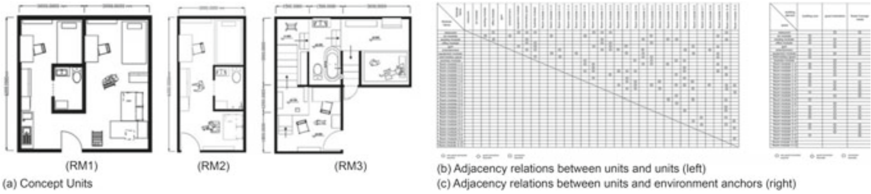
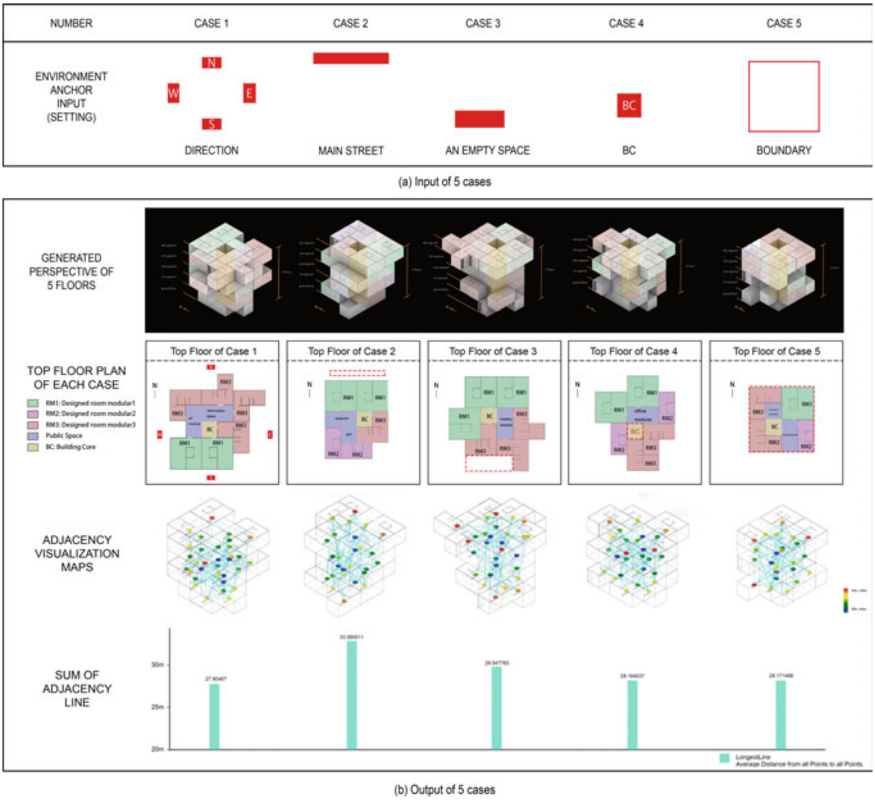

(RM1)
(RM2)
(RM3)
(a) Concept Units
(b) Adjacency relations between units and units (left)
(c) Adjacency relations between units and environment anchors (right)

Fig. 8. Basic prototype and user requirement data of hongkong social housing project offered by Archiford LTD

Subject to the building units satisfying the adjacencies in Fig. 8, we propose five different prototype design tasks (PDT). These requirements are for the top-floor plan, which makes it easier to observe. Specific PDTs and environment anchors settings (EAS) are described below, EAS are shown in Fig. 9a.



(a) Input of 5 cases

(b) Output of 5 cases

Fig. 9. EAS and the corresponding spatial layout results

Case 1: PDT: Try to make the RM3 unit lay on the north side of the plan. EAS: Design four direction anchors (North, South, West, and East) and increase the weights between RM3 and anchor 'North' (Fig. 9a).

Case 2: PDT: We intend the RM1 units to face or be further adjacent to the main street. EAS: Set the main street as the environment anchor. Adjust the position of the main street to check whether RM1 will change their positions accordingly (Fig. 9a).

Case 3: PDT: We intend to incorporate a terrace space in the southwestern part of the building. EAS: An 'empty' space. Check whether the terrace space will appear on the top floor of the building.

Case 4: PDT: Temporarily not adding other manual adjustments than the BC (Building core). EAS: BC. Use it to compare the effect of different environment anchor settings on the automated layout results.

Case 5: PDT: We expect it can be optimised to produce a plan-regular spatial layout. EAS: Boundary (Restrict the outline of the layout results). Check whether the layout of the top floor will be adapted to the boundaries we defined.

5 Result and Discussion

As shown in Fig. 10, the system achieves the rapid automated space layout, which the initial layout takes only 20 s to generate. From the result, we can find that most of the generated plans can satisfy the adjacency relationship, for example, the relationship between RM3 units and public space. Moreover, because of the manual modification, the accessibility to private units on each floor with building core and public spaces can also be satisfied.

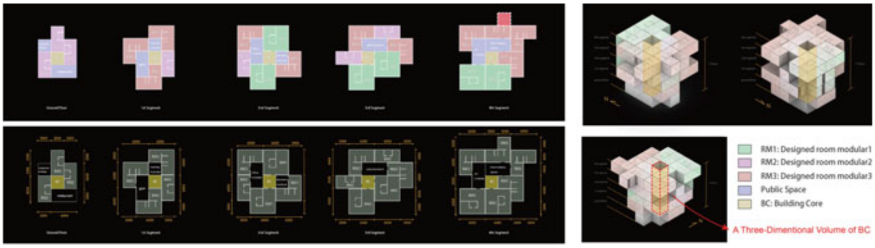

RM1: Designed room modular1
RM2: Designed room modular2
RM3: Designed room modular3
Public Space
BC: Building Core
A Three-Dimentional Volume of BC

Fig. 10. Quick 3D spatial prototype layout result

Figure 9b demonstrates the results of PDT. Our workflow can basically meet the initial layout of different tasks depending on environment anchors set by designers. For example, the RM3 units in Case1 are finally laid on the north side. The RM1 units in Case 2 are adjacent to the main street and are changed along with the location of the main street. The final layout of Case 5 generates a square boundary. Furthermore, we can note that compared with Case 4, Case 3 generates an appropriate terrace space on the south side by the designers' manual adjustment and machine coordination. However, these detailed tasks are challenging to achieve with only automated algorithms.

Figure 10 shows the spatial layout of a solution with a five-storey building, which proves the potential of a 3D layout. We directly achieve the prototype layout, satisfying the adjacency relationship in a three-dimensional by establishing horizontal traffic space and vertical core.

Limitation. Our work still has some limitations. Our approach can only control environmental anchor points rather than each building module to drive building layout results. To control each unit, we need new algorithms for calculating adjacency and position information about the unit in real-time in the modelling software Rhino in conjunction with the physics simulation in Grasshopper. It ensures that the unit is always available for editing (area, shape, position), enabling architects to solve more complex architectural tasks. In addition, in unassembled buildings design, the shape of each unit is fixed in advance, and it is difficult to produce a variety of architectural forms, such as the plans in Fig. 10. For this problem, we can add more optimization objectives (number of building sides, side lengths, volume, etc.) The multi-objective optimization approach can be combined with current workflows to generate more diverse and adaptable building solutions.

6 Conclusion and Future Work

In conclusion, our research focuses on building a human–machine collaborative workflow to assist in the rapid generation of spatial prototypes to meet the designer's customisation needs. Our workflow enables fast layouts based on adjacencies, saving a great deal of time for designers in an early design stage. As for the Building prototype generation, our workflow shows the ability to solve different design tasks initially. It allows designers to manually adjust some nodes to determine the final layout with the machine to meet the different design tasks in real-time. The limitation of each unit's boundary calculation and control ability needs to be enhanced to solve more complex spatial layout tasks. It will provide a hybrid-intelligent possibility of human–machine collaboration for solving complex building design tasks in the future.

References

1. Anderson C, Bailey C, Heumann A, Davis D (2018) Augmented space planning: using procedural generation to automate desk layouts. Int J Archit Comput 16(2):164–177
2. Homayouni H (2000) A survey of computational approaches to space layout planning (1965–2000). Department of Architecture and Urban Planning University of Washington
3. Fricker P, Hovestadt L, Braach M, Dillenburger B, Dohmen P, Rüdenauer K, Lemmerzahl S, Lehnerer A (2007) Organised complexity
4. Hao H, Ting-Li J (2010) Floating bubbles: an agent-based system for layout planning. In: Proceedings of the 15th CAADRIA conference, pp 175–183
5. White R, Engelen G (2000) High-resolution integrated modelling of the spatial dynamics of urban and regional systems. Comput Environ Urban Syst 24(5):383–400
6. Dapogny C, Faure A, Michailidis G, Allaire G, Couvelas A, Estevez R (2017) Geometric constraints for shape and topology optimization in architectural design. Comput Mech 59(6):933–965
7. Rahbar M, Mahdavinejad M, Markazi AH, Bemanian M (2022) Architectural layout design through deep learning and agent-based modeling: A hybrid approach. J Build Eng 47:103822
8. Elsayed M, Tolba O, Elantably A (2016) Architectural space planning using parametric modeling-egyptian national housing project. In: ASCAAD conference proceedings, pp 45–54
9. Veloso P, Rhee J, Krishnamurti R (2019) Multi-agent space planning: a literature review (2008–2017). In: CAADFutures
10. Arvin SA, House DH (2002) Modeling architectural design objectives in physically based space planning. Autom Constr 11(2):213–225
11. Baraff D (1989) Analytical methods for dynamic simulation of non-penetrating rigid bodies. In: Proceedings of the 16th annual conference on computer graphics and interactive techniques, pp 223–232
12. Boon C, Griffin C, Papaefthimiou N, Ross J, Storey K (2015) Optimizing spatial adjacencies using evolutionary parametric tools: using grasshopper and galapagos to analyze, visualize, and improve complex architectural programming. Perkins + Will Res J 7(2):25–37
13. Turner A (2001) Depthmap: a program to perform visibility graph analysis. In: Proceedings of the 3rd international symposium on space syntax, vol 31, pp 31–12

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.