

Designing with Space Syntax

A configurative approach to architectural layout, proposing a computational methodology

Pirouz Nourian¹, Samaneh Rezvani², Sevil Sariyildiz³

^{1,3}TU Delft, the Netherlands, ²Politecnico di Milano, Italy

¹http://wiki.bk.tudelft.nl/designInformatics/Pirouz_Nourian, ²<http://www.linkedin.com/pub/samane-rezvani/28/383/777>, ³http://wiki.bk.tudelft.nl/designInformatics/Main_Page.
¹p.nourian@tudelft.nl, ²samaneh.rezvani@mail.polimi.it, ³i.s.sariyildiz@tudelft.nl

Abstract. *This paper introduces a design methodology and a toolkit developed as a parametric CAD program for configurative design of architectural plan layouts. Using this toolkit, designers can start plan layout process with sketching the way functional spaces need to connect to each other. A tool draws an interactive bubble diagram and a set of tools reveal feasible geometric interpretations of the proposed bubble diagram in terms of plan layout graphs. Offering real-time Space Syntax analyses at the same time, the tools provide feedback on the spatial performance, which is translatable into the likely social performance of the plan layout patterns.*

Keywords. *Architectural configuration; graph theory; space syntax; spatial performance; plan layout.*

BACKGROUND

Space Syntax theory (Hillier and Hanson, 1984; Hillier, 2007) has established a methodological body of knowledge on spatial qualities of architecture as distinguished from its over-highlighted formal aspects. From an analytical point of view, Space Syntax theory provides a comprehensive and consistent framework for understanding spatial arrangements and their likely human effects, which we can term as social performance of buildings. From another perspective, in a study of building types as social constructs, John Habraken (Habraken, 1988) categorizes three major aspects of building types as social constructs: spatial organization, physical structure, and stylistic systems. He suggests that the one most intimately related to our behavior is the 'spatial or-

ganization'; he specifically mentions that a social role certain space has within a building is very much dependent on its 'position' as to the transition from public to private.

On the other hand, from a computational design perspective, the issue of 'plan layout' has been mostly addressed from various optimization points of views (Lobos and Donath, 2010); most of which deem configuration as an order that can be 'found' through thousands of trials and errors in putting spaces together in different ways in order to maximize certain qualities. This approach to plan layout is in deep contradiction to viewing architectural design as an intellectual activity initiated with 'proposing' configurative ideas. "Architectural and urban

design, both in their formal and spatial aspects, are seen as fundamentally configurational in that the way the parts are put together to form the whole is more important than any of the parts taken in isolation" (Hillier, 2007, p. 1). "Configuration as the way spaces are related to each other in order to serve a functional purpose is the very nature of architecture" (Hillier, 2007, p. 67); and yet we find very little about the way design can be systematically started through dealing with such a matter.

What is primarily missing in the literature about computational layout is a methodological approach rooted in consideration of "how designers think" (Lawson, 2005); likewise, a comprehensive consideration of social implications of configurations is absent. Specifically, in the mentioned optimization approaches to plan layout, it is often neglected to relate to design processes as practiced by designers. Designers do not seek to reach an order through thoughtlessly trying out random arrangements of spaces; on the contrary, they usually start with an 'idea' as to how spaces should be put together to function in a certain desired way. Such configurative ideas convey the understanding of architects from what is 'socially' considered as desirable.

SUMMARY

The initial idea behind the proposed design methodology, supported by our toolkit, was to give life to the bubble diagrams conventionally used for spatial arrangement, to allow for communication of configurative ideas between designers and computers. This idea brought about the following questions: How can a computational system interpret configurative ideas, put in the form of a bubble diagram, to plan layout patterns? Does a certain configurative diagram have only a single corresponding layout or more? If there are many, how can we systematically find the fundamentally distinct ones? Moreover, which design qualities result from the proposed connectivity patterns; and how can we study them methodically?

We have addressed these questions from a graph theoretical point of view and proposed a

computational design methodology (a structured collection of computational methods) embedded in a design toolkit in response. It begins with an abstract configurative arrangement of spatial entities by a human designer; follows with provision of interactive bubble diagram; resumes by systematic exploration of feasible geometric interpretations of the configurative inputs as plan layout patterns; and ends with dimensional specification of them according to the design brief (this feature is still under development).

A single connectivity graph, as an abstract entity, is interpretable to various geometric configurations all of which share the same pattern of interconnectivity although they may vary in size and shape from one another. Using this methodology, designers can sketch how the spaces are to interlink, and then they can use the toolkit as follows. A tool reads these interlinks and interprets them as a graph that captures the important spatial properties of a building; another tool finds a planar topological embedding of this graph; and a set of tools perform Space Syntax analyses such as depth (visualized in justified graphs), integration, control, choice, and difference factor (Hillier and Hanson, 1984; Hanson, 1998).

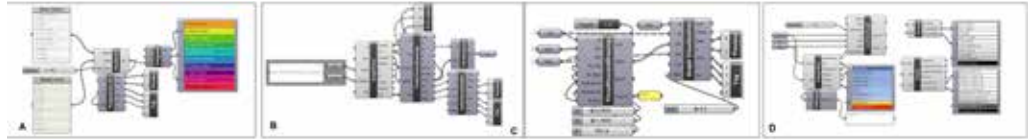
DESIGN WORKFLOW

The design process put forward by our toolkit is indeed a reflective cycle of see-move-see or "reflection-in-action" (Schon, 1987), rather than an automated problem solving procedure. The whole design workflow proposed by our methodology is about going from an abstract graph description of spatial connections to a topological planar embedding of that graph, analyzing that graph in real-time and finding feasible geometric cell configurations that admit the proposed graph of connections.

Our design methodology is an innovative fusion of what was proposed by Steadman and March (Steadman, 1983, pp. 69-75; March and Steadman, 1974); a Tutte (1963) convex drawing algorithm; our innovative force directed algorithm inspired by that of Eades (1984); a set of real-time Space Syntax analyses (these tools are the first real-time Space Syntax

Figure 1

Our toolkit: A) graph formation tools, B) graph reading, convex drawing and plan layout tools, C) force-directed bubble diagram tools, D) our Space Syntax toolbox.



tools integrated with a parametric architectural design workflow); and a set of algorithms for finding plan layouts inspired by Steadman and (Roth and Hashimshony, 1988).

We have developed a parametric design plugin in VB.NET that is installed as an add-on for Rhinoceros® and Grasshopper© [1] [2]. Our tool suite is developed as a plugin that is installed on Grasshopper, and it is undergoing final tests before release

(Figure 1).

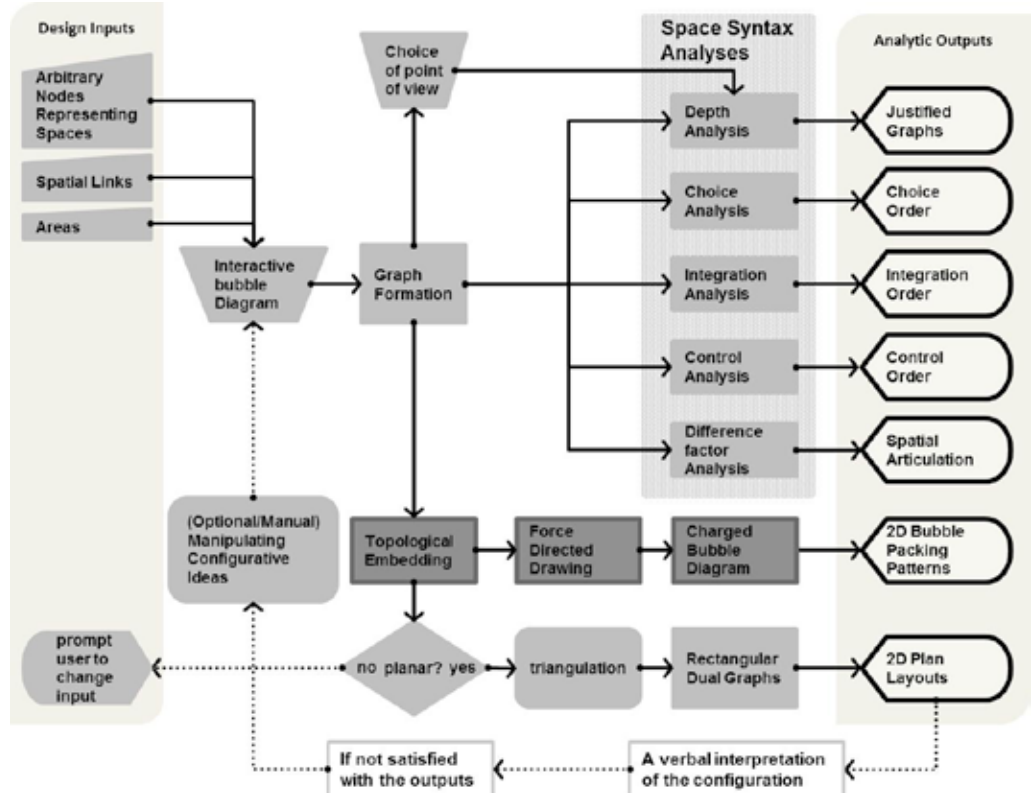
Technically, the course of actions suggested by our proposed design methodology (Figure 2) is as described below.

Step 1: Preparing the input

Designer starts with making a number of arbitrary points as for defining the center of functional spaces, a corresponding list of (rough or exact) area val-

Figure 2

A flowchart describing our proposed design methodology. We could wrap the whole set of tools in this way, but we chose to let the curious user try different tool configurations.



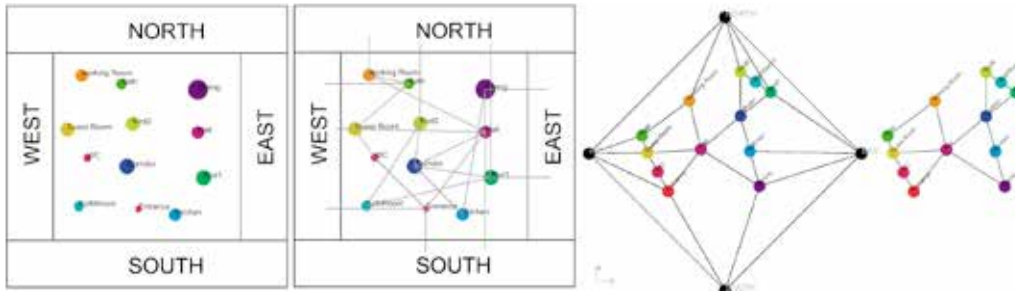


Figure 3
From left to right, the sketchpad, a sample configuration drawn by a designer, the unique untangled planar drawing of the configuration including and excluding the nominal North-East-South-West sides of the configuration..

ues, and a list of spatial labels (names) for them. A tool assigns rainbow colors to the functional spaces to make them more recognizable. To make it easy for the designer to link the nodes, a “graph reader” tool puts circles of sizes specified by the area values around all center points. The graph reader tool provides a sketchpad with the nominal North-South-East-West sides for the user to draw the connections (Figure 3).

Step 2: Producing a Connectivity Graph

According to their configurative idea, designer draws a line between every pair of points (circles representing functional spaces) that they think

should be directly linked. These links eventually would need the rooms to be adjacent to one another in order to be accessible immediately. This is to say that a set of connectivity requirements can be thought of as a subset of an adjacency requirements set. However, designers usually do not think of adjacencies in advance. It makes more sense to start with a set of required connections, even though it is more difficult to formalize. In our approach, it is easy for a designer to add an adjacency link and even distinguish between adjacency links and connectivity ones. This has important consequences on the ultimate floor plans. This point will be clarified in the explanation of next steps. Designer adds a few links to

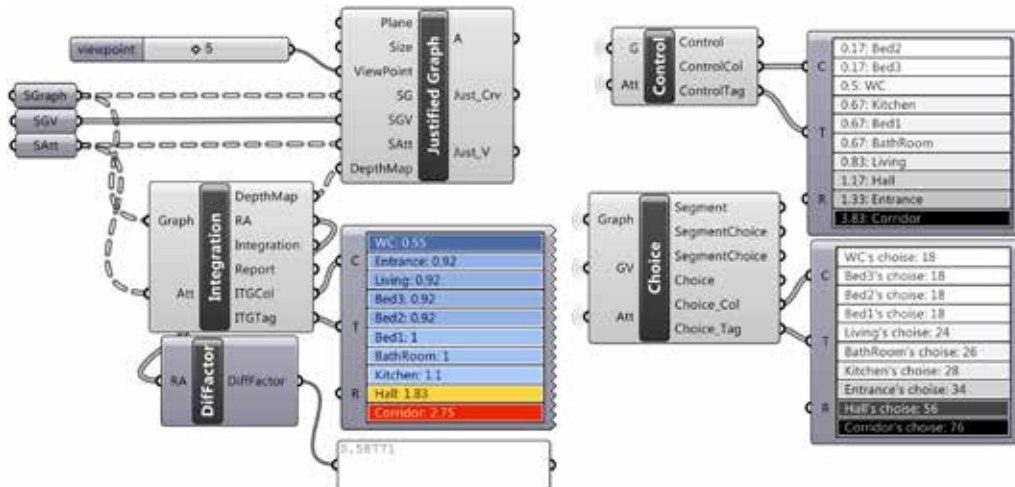
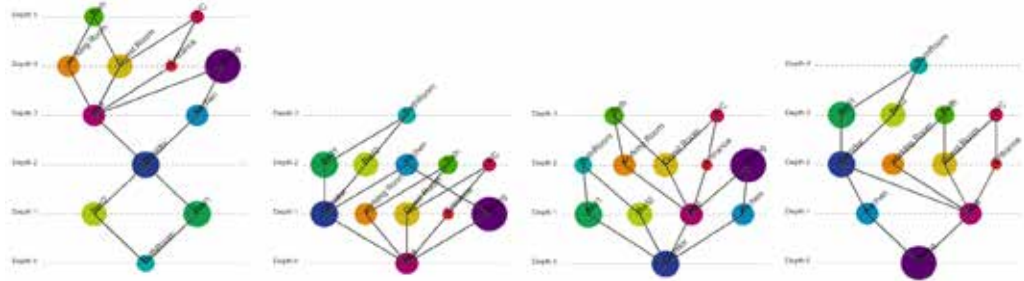


Figure 4
our architectural Space Syntax analysis tool suite in action.

Figure 5
A set of justified graphs drawn automatically (user can change the point of view) by our Justified graph tool.



relate some of the spaces to the nominal Northern, Southern, Eastern, or Western frontiers of their plan.

- The graph reader interprets the input links and points and their “label and area and color” attributes as a graph (Figures 2 and 4).
- It provides the user with a verbal interpretation of links between spaces.
- It tells the user whether there can ever be a plan in one floor with such connections (corresponding to a planar graph).

Step 3: Space Syntax Analyses

The theory of Space Syntax was initiated as a theory of architecture, seeking to explain the meaning of spatial configurations as to their social functions. Although it has been mostly used in urban analysis, it is still an architectural theory, and its basic examples are architectural. In simple terms, the theory of space syntax is focused on how spatial units relate to one another in buildings and built environments. In this context, the terms syntax and morphology are used practically in their linguistic senses. We could consider meanings for spatial arrangements, analogous to the way we do for verbal statements. While studying syntactic issues we look at how spaces relate to each other as a whole. Whereas, from another point of view, we could look at the individual spaces and focus on their morphological aspects, and their geometrical state of being. This is to say, loosely speaking, that the former concerns the topology and the latter concerns the geometry of built environments. In our tool suite, we have implemented a few of space syntax measures including:

- *Depth* (Automatically Visualized in Justified Graphs)

The first thing we need to know about a configuration is how many topological steps a single space is away from another one. A distance measured between two nodes on a graph is called the graph theoretical distance between them. We have developed an automated “Justified Graph” drawing tool that visualizes such distances on depth levels. In any configuration, one can choose a point of view to look at their proposed configuration literally from different points of views (Figure 5).

- *Integration* (Hillier and Hanson, 1984)

Integration (1) is a measure of centrality that indicates how likely it is for a space to be private or communal. The more integrated a space, the shallower it is to all other nodes in a configuration. Integration is calculated by computing the total depth of a node when the depths of all other nodes are projected on it. It is formalized as in (1) in which k denotes the number of nodes, TD is the total depth as explained above, and D_k , the so-called diamond value, is obtained from (2). It indicates how an individual space is private or communal within a configuration.

$$I = \frac{D_k(k-2)(k-1)}{2(TD-k+1)} \quad (1)$$

$$D_k = \frac{2\left(k\left(\log_2\left(\frac{k+2}{3}\right)-1\right)+1\right)}{(k-1)(k+1)} \quad (2)$$

- *Difference Factor* (Hanson, 1998)
As a measure of spatial articulation for a whole configuration, the difference factor indicates how differentiated the space are within a configuration. It is calculated according to (3), (4), (5) and (6).

$$RA = \frac{2(TD - k + 1)}{(k - 2)(k - 1)} \quad (4)$$

$$a = \text{the maximumm RA}, b = \text{mean RA}, c = \text{minimum RA} \\ t = a + b + c$$

H: Unrelativized Difference factor

$$H = - \left\{ \left[\frac{a}{t} \ln \left(\frac{a}{t} \right) \right] + \left[\frac{b}{t} \ln \left(\frac{b}{t} \right) \right] + \left[\frac{c}{t} \ln \left(\frac{c}{t} \right) \right] \right\} \quad (5)$$

Relativized Difference factor

$$H^* = \frac{H - \ln 2}{\ln 3 - \ln 2} \quad (6)$$

- *Control* (Hillier and Hanson, 1984; Hillier et al, 1987)

Control value (7) intuitively indicates how strongly a vertex in a graph (a space in a configuration) is linked to other points in a superior manner. It is computed by (7) in which D_i is the degree of a 'neighbor' node, and n is the number of all neighbor nodes.

$$\text{Control} = \sum_{i=1}^n \frac{1}{D_i} \quad (7)$$

- *Choice* (Originally introduced as Betweenness by Freeman (1977))

Choice or Betweenness is a measure of importance of a node within a configuration. That literally tells how many times a node happens to be in the shortest paths between all other nodes. It can also be computed for the links connecting the nodes in a similar way. It is computed by (8) in which $\sigma_{jk}(P_i)$ is the number of shortest paths between nodes P_j and P_k which contain node P_i , and σ_{jk} the number of all geodesics between P_j and P_k .

$$C_B(P_i) = \sum_j \sum_k \frac{\sigma_{jk}(P_i)}{\sigma_{jk}} \quad (j < k) \quad (8)$$

Step 4: Producing a Unique Convex Embedding of the Connectivity Graph

A very important tool in our tool suite is for untangling a connectivity pattern that still has an abstract meaning. This tool produces a unique topological embedding of that pattern on a plane. It is implementing the Tutte algorithm for convex drawing (Tutte, 1963). The valuable point is that once this (linear-time) algorithm converges into an embedding (usually in a small fraction of a second) we are certain that it is unique. Therefore, that means that no matter how we provide the connectivity input, we always get the one embedding that corresponds to that single graph of connectivity. A topological embedding indicates how the vertices of a graph are connected to one another on a surface. It is usually expressed in terms of 'face' descriptions. There is only one convex embedding of a planar graph, which is revealed by Tutte algorithm. The convex drawing algorithm reveals the unique planar topology of the connectivity graph, given that it is linked in a particular way to the nominal "North, East, West, and South" (NEWS). A topological description is in between an abstract connectivity description and a concrete geometry. This is exactly the breakthrough of our computational methodology that it uses a Tutte embedding for generating geometric graph drawings and plan layout patterns. This tool also performs a planarity test and tells the user if a floor plan is admissible for the set of connectivity requirements; provides an ordering for automated justified graph drawing; and distinguishes a sub graph of the whole connectivity graph (excluding NEWS vertices). This sub graph, its vertices and its attributes will be used further on (Figure 3). This tool also generates error messages when the connectivity graph is not planar. The tutte algorithm, however, could deliver result with poor geometric resolution in some cases. To overcome this drawback we introduced our force-directed drawing tool in addition.

Step 5: Force-Directed Graph Drawing

This tool contains our force-directed graph-drawing algorithm and makes a "kissing disk" drawing of the

Figure 6

A few samples of interactive bubble diagrams (user can change the areas in real-time) produced by our force-directed graph drawing algorithm. The links are colored according to their betweenness importance values.



bubble diagram. This algorithm works by a set of attractive and repulsive forces (as in (9)) acting recursively on graph vertices, seeks a 'relax' situation for a graph, and reaches to a graph drawing. This tool is quite intuitive and shows in real-time bubble diagrams neatly according to the specified areas and the connectivity graph (Figure 6).

Attraction: $AF_{ij} = k_a \Delta x_{ij}$ for all linked (i, j)

Repulsion: $RF_{ij} = \frac{k_r}{x_{ij}}$ for all (i, j)

Step 6: Revealing Dimension-Less Plan Layout Patterns

A convex drawing found in the fourth stage can be 'triangulated' so as to give rise to dual graphs that can represent a cell configuration admitting the connectivity graph in itself. While triangulating, we may add links that were not proposed as connectivity links, but they simply imply adjacencies that may arise out of compactness and enclosure geometric constraints. If we confine the triangulations to a particular type of triangulations, then we may get rectangular dual graphs that can be viewed as dimension-less plan-layout patterns (Figures 7 and 8). These "dimension-less dissections can be later dimensioned by means of two algorithms introduced

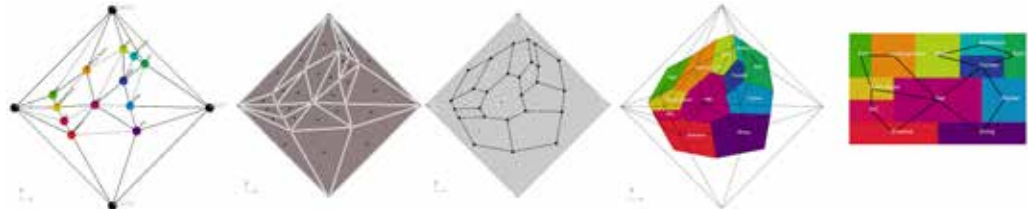
in (Steadman, 1983; Roth and Hashimshony, 1988; March and Steadman, 1974). We are still developing the dimensioning process and so far, the tool goes until delivering dimensionless patterns.

ANALYSIS AND EVALUATION OF SPATIAL PERFORMANCE

Space Syntax measures and their distributions are qualitatively interpretable into concepts such as privacy and community (Hillier, 2007, p. 22). In case of residential plans for instance, the various representations and measures of Space Syntax show how domestic space manifests life styles, social meanings, and identities of different sub-groups within society (Hanson, 1998). Using Space Syntax methodology, the system interprets spatial arrangement from the very moment it is drawn as a bubble diagram, and gives qualitative feedback on the implications of this diagram to the designer as spatial performance measures. According to the design context, designers are free to interpret these spatial performance measures into the 'likely' social performance of their ideas. As a result, performance analysis is automated by the system; but performance evaluation, i.e. judging the relative goodness of design alternatives, due to the intellectual complicacy of the matter and es-

Figure 7

The course of computational procedures for triangulating a connectivity graph by adding adjacency links, finding a dual graph and a rectangular dimension-less plan layout pattern.



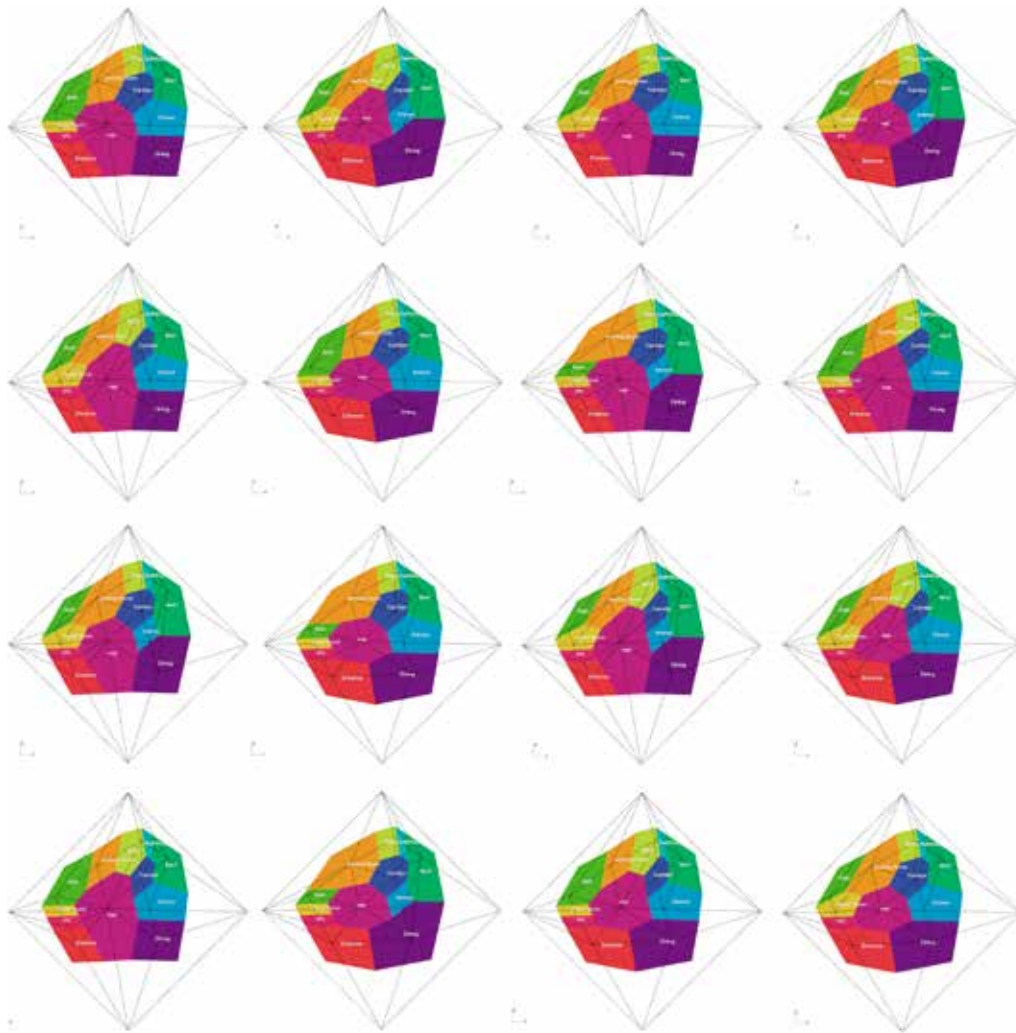


Figure 8
The 16 feasible plan-layout patterns of a sample connectivity graph revealed and enumerated by our tools exhaustively.

pecially because of its contextual essence, is intentionally left for human designers using the system.

DISCUSSION

It is left for the designers to decide on how they want to alter their ideas during the design process, but the tools always provides them with automatic

feedback on the properties of what they design; while showing them their own ideas, literally, from different points of view. It is important to note that these ideas usually evolve during the course of design process, as problem formulations and solutions evolve together (Dorst and Cross, 2007). Viewing a justified graph, designer can choose from which

space the other spaces are seen, say from different points of views, and analyze it in terms of syntactic measures. This helps designers see if what they have proposed in terms of a bubble diagram actually matches with their initial ideas on privacy/community, spatial articulation and other spatial qualities. We argue that through the design process put forward by this 'tool and methodology' package, designers have full intellectual control over the spatial qualities of their designs; they can benefit from computation in seeing their own ideas from different angles; and they receive objective feedback on the spatial qualities of their designs and indications on likely social performance of their designs. Our design toolkit allows for interactive diagrammatic design by human designers and suggests them multiple possible interpretations of their own configurative ideas.

REFERENCES

- Dorst, K and Cross, N 2007, 'Co-evolution of Problem and Solution Spaces in Creative Design', *Computational Models of Creative Design*, Key Centre of Design Computing and Cognition, Sydney.
- Eades, P 1984, A Heuristic for Graph Drawing, *Congressus Numerantium*, 42(11), p. 149–160.
- Freeman, L 1977, 'A set of measures of centrality based upon betweenness', *Sociometry*, 40, p. 35–41.
- Habraken, NJ 1988., *Type as a Social Agreement*, s.n., s.l.
- Hanson, J 1998, *Decoding Homes and Houses*, Cambridge University Press, Cambridge, UK.
- Hillier, B and Shinichi, I 2007, *Network and Psychological Effects in Urban Movement*, Springer, Melbourne.
- Hillier, B and Hanson, J 1984, *The Social Logica of space*, Cambridge University Press, Cambridge, UK.
- Hillier, B, Burdett, R, Peponis, J and Penn, A 1987, Creating Life: Or, Does Architecture Determine Anything? *Architecture et Comportement/Architecture and Behaviour*, 3(3), p. 233 – 250.
- Hillier, B 2007, *Space is the Machine*, Cambridge University Press, London.
- Lawson, B 2005, *How designers think*, 4th ed, Architectural Press, Elsevier, Burlington.
- Lobos, D and Donath, D 2010, 'The problem of space layout in architecture', *arquitecturavista*, 6(2), pp. 136-161.
- March, L and Steadman, P 1974, *The Geometry of Environment: An Introduction to Spatial Organization in Design*, MIT Press, s.l.
- Roth, J and Hashimshony, R 1988, 'Algorithms in graph theory and their use for solving problems in architectural design', *computer-aided design*, 20(7), pp. 373-381.
- Schon, D 1987, *The Reflective Practitioner*, Cambridge University Press, Cambridge, UK.
- Steadman, P 1983, *Architectural Morphology: An Introduction to the Geometry of Building Plans*, Taylor and Francis, s.l.
- Tutte, WT 1963, *How to draw a graph*. s.n., s.l.
- [1] <http://www.rhino3d.com/>
- [2] <http://www.grasshopper3d.com/>