

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/256471417>

Parametric Urban Patterns – Exploring and integrating graph-based spatial properties in parametric urban modelling

Conference Paper · January 2012

DOI: 10.13140/RG.2.1.2045.6163

CITATIONS

16

READS

1,709

3 authors:



Martin Bielik

Bauhaus-Universität Weimar

25 PUBLICATIONS 134 CITATIONS

[SEE PROFILE](#)



Sven Schneider

Bauhaus-Universität Weimar

61 PUBLICATIONS 252 CITATIONS

[SEE PROFILE](#)



Reinhard Koenig

Bauhaus-Universität Weimar

139 PUBLICATIONS 572 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



ESUM: Investigating the relationships between the energetic and the social performance of urban fabrics [View project](#)



Water Based Urban Design (WaBuD) [View project](#)

Parametric Urban Patterns

Exploring and integrating graph-based spatial properties in parametric urban modelling

Martin Bielik¹, Sven Schneider² and Reinhard König³

Bauhaus-University Weimar, Germany

<http://infar.architektur.uni-weimar.de>, <http://www.decodingspaces.de>

¹*martin.bielik@uni-weimar.de*, ²*sven.schneider@uni-weimar.de*, ³*reinhard.koenig@uni-weimar.de*

Abstract. The article presents a graph-based spatial analysis toolset (“decoding spaces” components) which we have recently developed as an extension of the visual scripting language Grasshopper3D for Rhino. These tools directly integrate spatial analysis methods into CAD design software which can have a significant effect on current design workflows. However, grasshopper doesn’t only enable the results of analyses to be used in the standard Rhino modelling environment. It also makes it possible to integrate spatial analysis into a parametric design approach as discussed in this paper. The functionality of this toolset is demonstrated using a simple urban design scenario where we introduce the idea of parametric patterns based on graph-measures.

“Keywords” spatial analysis, parametric modelling, urban layout, design process, decoding spaces.

Introduction

Graph based spatial analysis is a method which was first introduced in architecture and urban planning by Bill Hillier and his colleagues in the late 1970s as Space Syntax theory (Hillier & Hanson 1984). Because tests have shown that there are correlations between graph-based measures and functional aspects of a spatial configuration, the method has the potential to help architects in forecasting the socio-economic effects of their designs. The core principle of the methodology is to represent space (e.g. an urban or floor plan) as a configuration of single elements (e.g. streets, rooms) and to analyse their mutual relations. Three different representations are used to analyse space in terms of its basic elements: isovists, convex spaces and axial lines. For the software concept described below, we are particularly interested in the latter. An axial line is basically a line of sight in an environment. Based on the assumption that people use lines as a mental concept to orient and move through cities, axial maps represent a model of urban space as essentially a network of linear spaces. An axial map is a set of axial lines which cover the open space of a city completely. This map can be analysed using graph-based methods. For this the map needs to be converted into a graph. The lines represent the nodes, while the interconnections between the lines represent the edges of a graph. There are two important measures which can be calculated based on this graph: between-ness, closeness and connectivity. Closeness (or integration) measures the average distance from one element to all other elements (global integration) or to elements within a certain radius (local integration). The distance calculation is undertaken by counting the steps necessary to move from one element to another. The measure indicates, globally or locally, the

topological centre of a city, i.e. the part or parts of a city where most people reside and where the highest density of buildings with retail functions is to be found. Betweenness (or choice) measures how often an element is passed if all the shortest paths in the graph (of each element to all other elements) are traversed. The elements with the highest closeness value are more frequently passed. The concept of axial maps was later extended by another linear spatial representation, so called segment maps (Hillier & Iida, 2005). This is also based on the line network, but its basic element is a line segment. A segment occurs between the intersections of axial lines. For calculating the distance between two segments, not the number of segments, but the angle between the segments is taken into account. Compared with axial maps, segment maps offer two advantages: firstly, by using angles instead of steps the analysis results correlate more strongly with movement patterns; and secondly, since the method uses smaller elements it offers a much finer scale of configurational analysis.

Since the analysis methods above focus solely on the geometrical arrangement of spatial elements and require no additional data on land use or traffic for their calculations, they are particularly suitable for examining design alternatives in terms of their spatial characteristics. There are several software packages available to run such analyses, but creating and changing the geometry has to be done with specialised CAD software. Since the design process is an ongoing iterative process of improvement based on the creation of ideas and their evaluation (Lawson, 2006), it is beneficial for the design to explore as many iterations as possible in the design process. With the conventional workflow and file formats, this is problematic: each iteration requires that the user exports and imports data back and forth between the design and analysis tools. This hinders the design workflow significantly and acts as a disincentive for the designer to explore a wide range of variants.

For exploring different variants, parametric modelling represents a new approach to creating complex forms in architecture and urban design. In parametric modelling the final geometry is a result of a modelling process driven by algorithms and certain input parameters (Woodbury, 2010). The advantage of this approach is the ability to easily change the input parameters and generate new variants of a design. In parametric urban modelling, one can consider a multitude of factors for defining a final shape. The aim of this modelling process is to integrate the multidimensional character of real world situations, helping the designer to create sustainable environments. Geometrical and data parameters such as site morphology, height regulations, composition guidelines and various role and density urban indicators can be used to shape the city (Beirão, 2011). However, up to now there has been no support for including spatial analysis in this model.

By coupling these two methods we can contribute towards creating a more effective design process that employs spatial analysis and parametric modelling. This would make it possible to effectively analyse variants on the one hand and to incorporate the analysis results in the modelling process on the other.

“Decoding spaces” components

In order to link together the two methods more directly, we developed an extension for Grasshopper3D for Rhino, a well-known parametric modelling system. Grasshopper can be described as a visual programming language used mainly to generate geometries. The

language consists of so-called components which can process predefined data types (e.g. lines, surfaces, numbers) as their input and return processed data as their output. The data processing can be anything from simple mathematical operations to complex geometrical transformations. A parametric model is built up by creating interconnections between these components, whereby the output of one component forms the input of another one. Since Grasshopper is a powerful tool for geometric modelling and is widely used among architects, we decided to use this tool as a basis for our graph-based analysis tool.

Grasshopper can be extended by programming additional components. An SDK is available for the development of Grasshopper components that provides mechanisms for the exchange of data (input and output of the component). The components described in this section were developed in C# using the .NET Framework 3.5.

The newly developed components are a toolset which makes it possible to run a graph analysis on parametric line structures and to use the results of the analysis for further modelling. The toolset is named “decoding spaces”, where the term “decoding” relates to the analysis of space, and the term “coding” to its geometric modelling. For the design of this toolset, we have chosen a modular structure (see Figure 1) which improves the efficiency of calculation and the facilitation for future expansion. The different components can be grouped into four categories: preparation for analysis, calculation of graph measures, visualisation of the analysis results and special modelling tools.

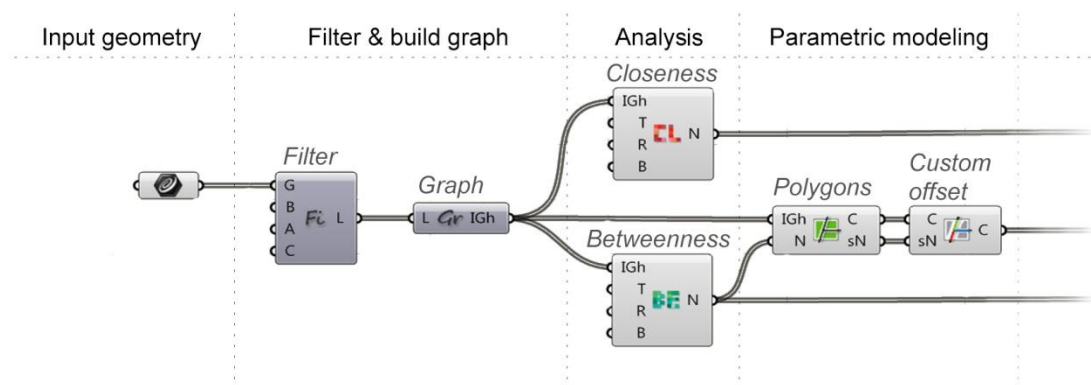


Figure 1
Modular concept of the spatial analysis framework for Grasshopper

As an analysis method, we have decided to use segment map analysis as described in the first section. This form of representing spatial structures can be easily mapped in a parametric model because it relates directly to the road network. On the other hand, it has been proven that the outcome of this method of analysis strongly correlates with pedestrian movement and distribution, and it is therefore of direct practical benefit for the design of cities and neighbourhoods.

In the following section, we describe the functionality of the different components we have developed.

ConvertToSegmentMap component

The segment analysis can only be undertaken on line segments. Any axes, curves and splines used by the designer must therefore first be converted into segments. This step is

important to ensure that designers retain formal freedom as splines are easier to control than single lines made up of composite curves. The component reads the geometry (any set of lines, polylines, curves and splines) and converts it into a segment map. The curves and splines are first subdivided into a finite number of segments, then segments are created between the intersections of all lines. Finally, all duplicates are removed (see Figure 2).

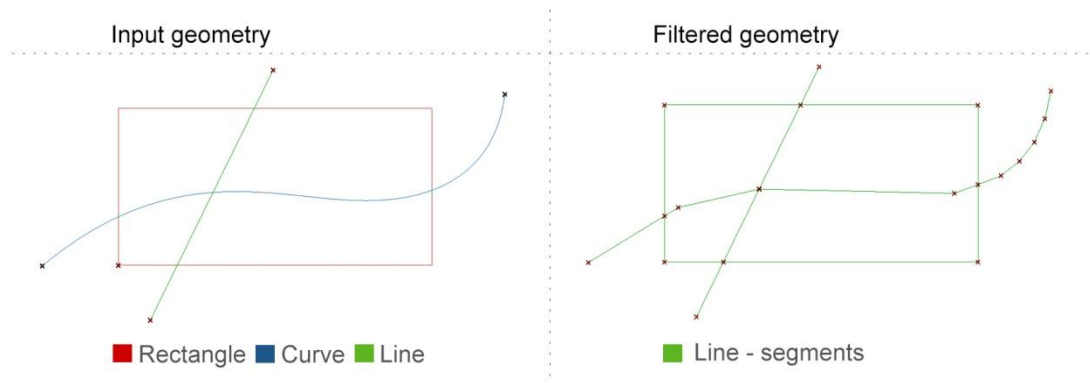


Figure 2
Filter Component converting geometry into line segments

The ConvertToGraph component

The ConvertToGraph component converts the geometry resulting from the ConvertToSegmentMap component (a segment map) into a graph. The graph is formed through interpreting each segment as a node and connections to other segments (in the case of equal endpoints) as edges. The weightings of the edges of the graph are based on the angles between intersection points of the segments. This is because it is assumed that humans prefer to take routes through a city that deviate as little as possible from the intended direction (i.e. minimal angular deviation). An angle of 90° is therefore weighted high, while an angle of 0° between two segments has no weight.

Closeness and Between-ness components

After a graph has been created from the segment map using the ConvertToGraph component, various analysis-components can be applied to calculate different graph-measures. Components have been developed for two major measures: Between-ness (choice) and Closeness (integration). These components can also be used to colour the geometry for display in the Rhino viewport. In addition, they can be analysed at different radii; e.g. they evaluate movement patterns on both a global and local scale.

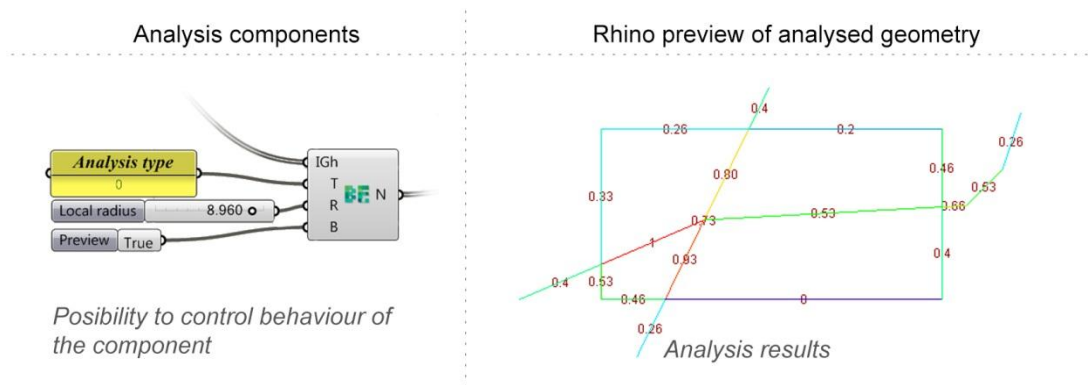


Figure 3
Analysis Component and its Rhino display

Analysis-related parametric modelling components

In addition to the pure analysis, we have implemented some useful modelling tools based on graphs, which are well suited for urban parametric models. These components include the generation of plots from street networks (polygons from graph) and their shaping through analysis results (custom-offset & custom-extrude).

Parametric urban patterns – form follows graph-based measures

In the following section we demonstrate the applicability of the above methods in a simple test scenario. We create an urban district based on parametric algorithms, driven by graph-based measures. These algorithms bear relation to the idea of using patterns in architecture and urban planning, which can be traced back to Christopher Alexander's book "A Pattern Language" (1977). In his book, Alexander explains the use of patterns to create a design in a manner similar to the way we use words to create sentences. Every pattern describes a typical design problem, a way to solve this problem and other related patterns which have to be considered when applying a pattern.

Today, parametric modelling makes it possible to rethink this idea of patterns and transform them into a new way of modelling designs. These algorithms are similar to Alexander's patterns in that they represent a solution for recurring problems in the environment. Using parametric modelling such a solution is described in the form of a computational algorithm capable of generating geometry. To demonstrate this idea, we developed three exemplary parametric patterns that use graph-based measures as parameters and applied them to a fictitious urban scenario. The fictitious urban grid has itself been generated using a simple subdivision algorithm implemented by default in Grasshopper (substrate component, see Figure 2).

This scenario leverages a very basic idea that underlines any parametric design – a single algorithm can generate any number of results (e.g. urban district) simply by changing the parameters (e.g. street grid). Patterns driven by spatial properties are on their own not sufficient to provide solutions for complex multidimensional real world problems where lots of other "patterns" also need to be considered. But they are definitely extremely important in shaping our environment as discussed in the introduction, and could play a crucial role in this kind of algorithmic pattern language. It should be noted that the patterns and their combinations described in the following are

examples and serve only to illustrate the concept of parametric patterns driven by spatial properties.

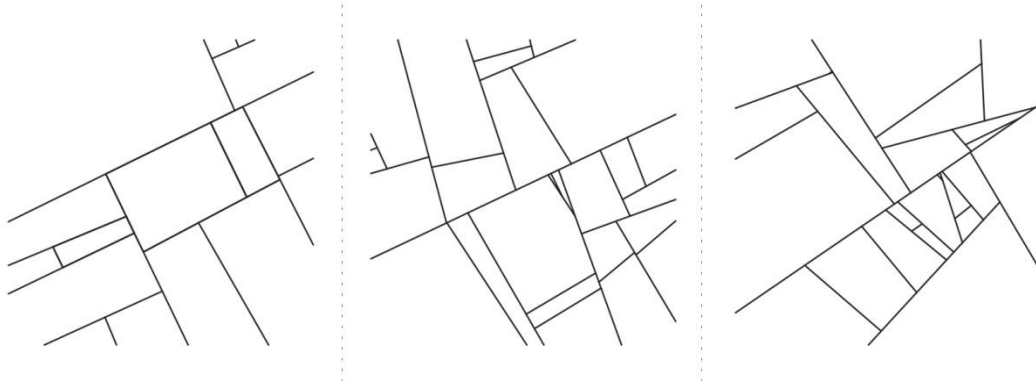


Figure 4
Fictitious urban district

Generating street width

In the first pattern, we use closeness analysis which has proved to be a good indicator of traffic frequency. The width of the roads is associated with the closeness value according to the principle that the more frequented a road segment is, the more space it should provide for pedestrians/cars. In order to do this, we need to recognize distinctive plots from a given street network and create a single closed polygon for every one of them. This is done using our own modelling component named “Extract polygon”, which uses the graph interpretation of the current line network delivered by the “graph component”. After this step, we need to offset each polygon edge by a distance that corresponds to the analysis value of the respective street segment. Since there is not a default option to offset each edge of a polygon a different amount, we built our own “custom offset” component. This component takes a polygon along with a list of offset values for each edge of the polygon as input parameters and outputs an offset polygon. Variants of the application of this pattern are shown in figure 3.

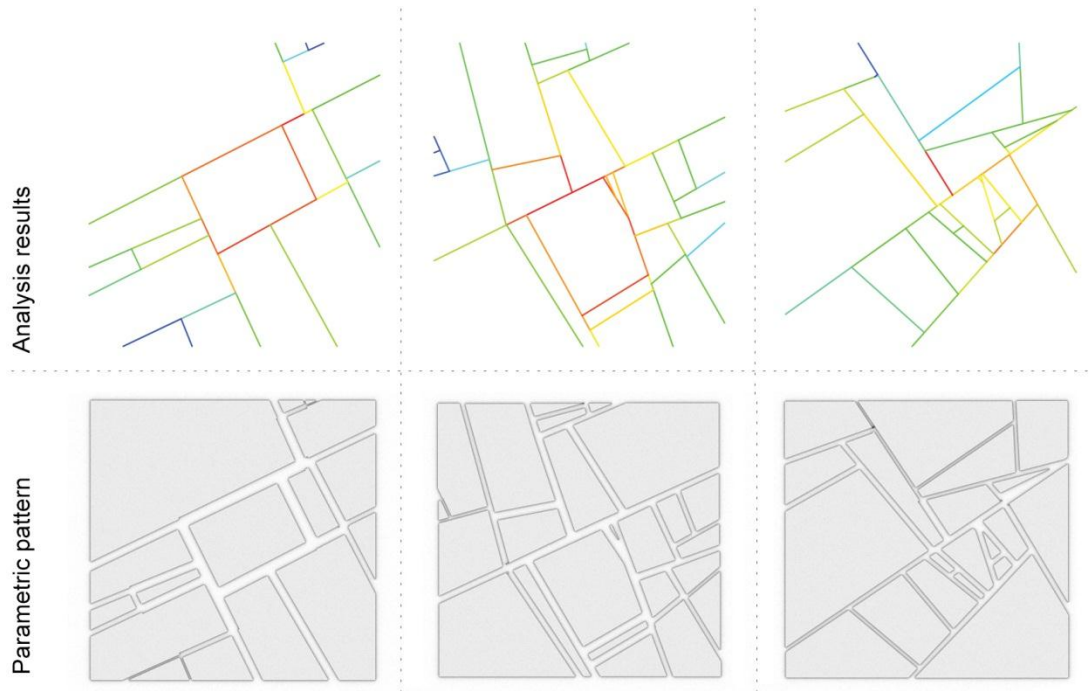


Figure 5
Street width generated using closeness analysis

Generating building height

The second pattern links the height of the building to the Closeness centrality. This analysis reveals local and global topological centres as explained in the introduction. Here the idea is that in the centre of cities, there is greater demand for housing and office space than in peripheral areas, which results in increasing building heights as one grows nearer to the centre. Here we use standard grasshopper modelling tools incorporated in the algorithm that generate a mountain-like massing model over the network of lines where the peaks and values reflect the betweenness of street segments below (see Figure 4).

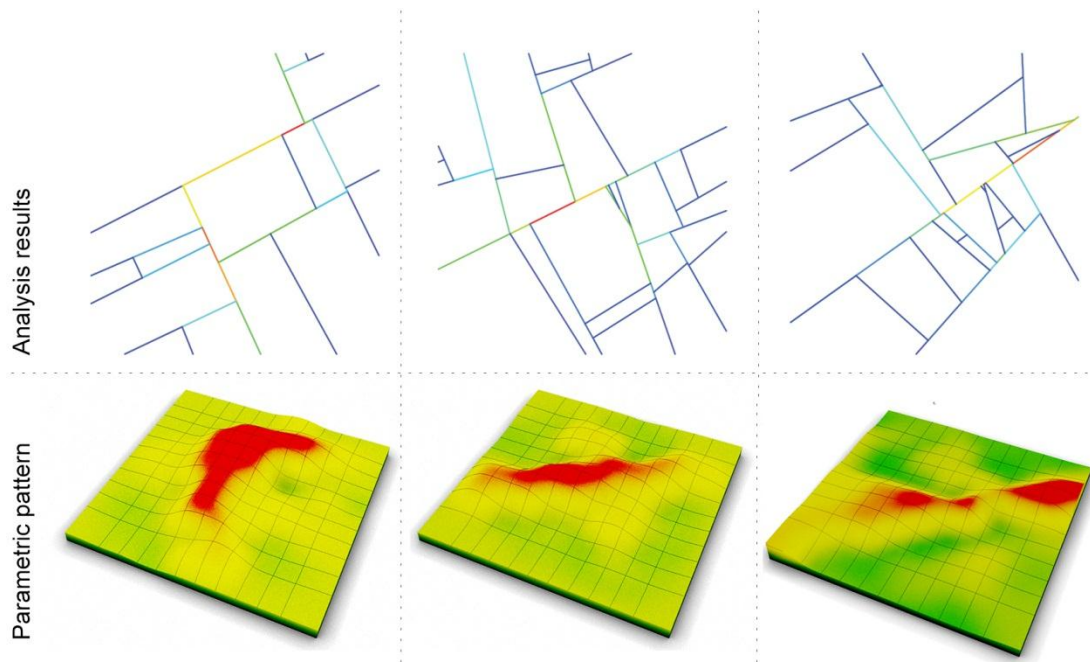


Figure 6
Massing model generated using Closeness analysis

Generating public space

The third pattern is used to create reasonable public spaces. The topology of the street network is the main factor that predetermines the distribution of inhabitants within the network and the function can either support or weaken this predisposition. The idea behind this pattern is that public spaces function well if they lie on integrated and well frequented streets. The public space component uses the results of the analysis, user defined ratios between free and built-up plots and their minimal distance as inputs and suggests where to allocate public spaces (see Figure 5). To create these public spaces, the respective plots are filtered out of the list of plots.

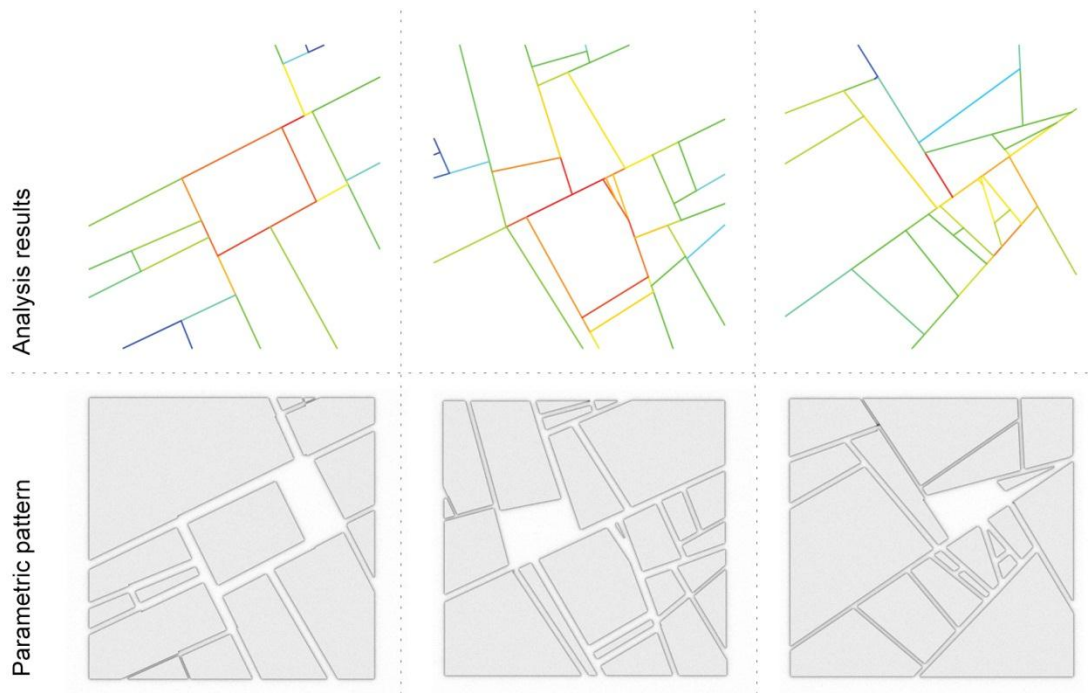


Figure 7
Choosing where public spaces should be

Putting it all together and examining variants

The last example shows how the rules can be combined. By combining patterns one is able to build up complex urban models. Since the model is parametric in nature it is possible to generate and evaluate lots of alternative solutions by changing different input parameters (see Figure 6). In this case, the chosen patterns criteria did not contradict one another which made it easy to combine them.

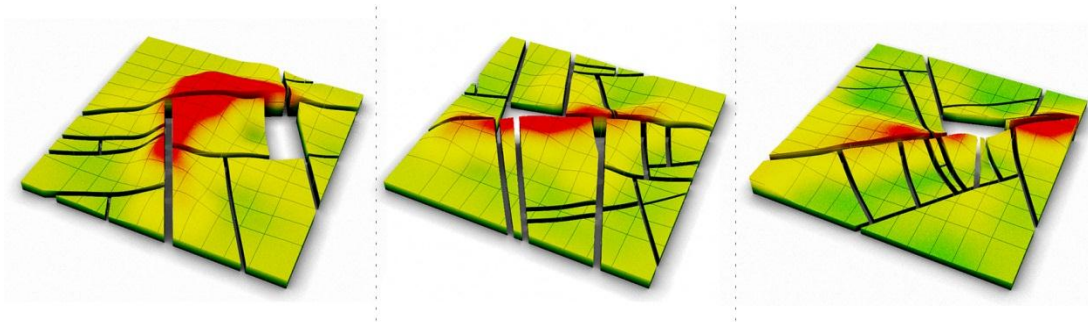


Figure 8
All patterns combined together

Conclusion & Outlook

In this article software components have been presented that make it easy to couple parametrical models with graph-based spatial analysis (closeness, between-ness). This offers two main advantages: firstly, the analysis can be run directly in a CAD

environment (using Rhino as a geometric modeller), which makes removes hurdles in the design workflow; and secondly, the results of the analysis can be used directly as parameters for the parametric model. This opens up new ways of thinking about how one can incorporate spatial analysis into the creation of forms. One possible way was outlined by introducing the concept of parametric urban patterns. The algorithmic structure of these patterns makes it possible to apply computational technologies to design tasks. The use of computational methods represents a significant enhancement in response time between the choice of a pattern and its application, and this is one of the greatest advantages of this new interpretation of design patterns.

If we look at the design approach proposed by Alexander (1977), we will find one important property that is not yet integrated in our model: It is the ability to effectively combine the patterns without any prescribed order. However, the combination of many different patterns presents an algorithmic challenge. The degree to which multi-objective optimization processes can be utilized to overcome the weakness of our current approach needs to be addressed in further studies. One direction seems to be clear: to replace the current linear workflow with a cyclic one, solutions can be explored by weighting different criteria until one finds a satisfactory solution.

The newly developed components for Grasshopper, as well as videos and tutorials can be downloaded from www.decodingspaces.de.

Acknowledgement

This work is supported by the German Research Foundation (DFG), reference number DO 551/18-2.

References

Alexander, Christopher et al.: 1977. A Pattern language, Oxford University Press, New York.

Beirão, J. and Nourian, P., Mashhoodi, B.: 2011. Parametric urban design: An interactive sketching system for shaping neighborhoods, *Proceedings of the eCAADe Conference*, Ljubljana, Slovenia, pp. 225-234

Hillier, B and Hanson, J.: 1984. The Social Logic of Space. University Press. Cambridge.

Hillier, B. and Iida, S.: 2005. Network effects and psychological effects: a theory of urban movement. Proceedings of Spatial Information Theory: International Conference, N.Y., U.S.A. pp. 475-490.

Lawson, B. 2006. How Designers Think. (4th Edition) Architectural Press Oxford.

Woodbury, R.: 2010. Elements of Parametric Design. Routledge. London.