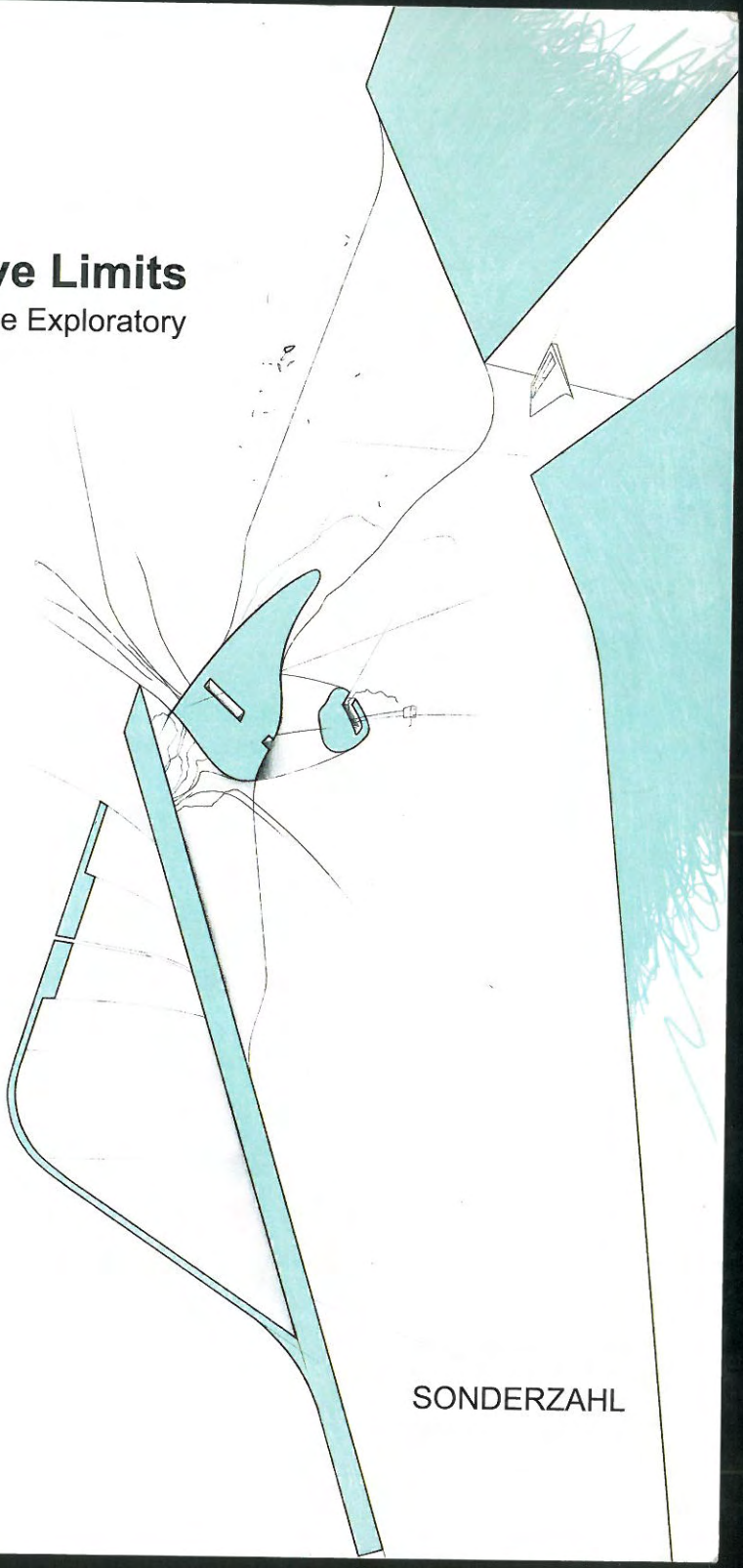
Productive Limits

Architects Gone Exploratory



SONDERZAHL

Productive Limits
Architects Gone Exploratory

Edited by Sophie Hochhäusl and Manuel Singer

SONDERZAHL

Content

Just Good Enough, is Good Enough for Architecture! Optimizing - a Task for the Architect

Richard Schaffranek

An Optimized Kitchen

A specific recipe prepared in a specific kitchen creates a specific path through that kitchen. If one parameter is changed, e.g. the layout of the kitchen or the instructions in the recipe, the length of the path created by this recipe in this kitchen changes. If one wants to optimise the process of cooking, one way is to change the spatial arrangement of the kitchen. Another way is to change the sequence of a recipe. For now let us say, we would like to "optimize" the recipe, by finding the shortest path that visits every function needed. This problem can be understood through a graph. The different functions are represented through the nodes. The edges represent the distance between the functions. This problem is referred to as the "traveling salesman problem" in computer science.

When a recipe needs to visit three functions A, B, C in a kitchen, and each function is connected with each other function, there is one solution AB, BC, CA. Are there four functions A, B, C, D there are already three possible sequences [AB, BD, DC, CA], [AD, DB, BC, CA] and [AB, BC, CD, DA]. To know which of these sequences is the shortest, all have to be calculated. With n = 4 all possible sequences can be computed[2] with ease but when n increases the amount of possible solutions quickly becomes infinite.[3] Different approaches to such problems have been developed. The search for the optimal solution is computed is discarded in favor of a good enough solution in a reasonable time. This is called "a heuristic" in computer sciences. Starting from such a heuristic the solution can be improved through applying different "optimization" algorithms. Some of these algorithms guarantee an improvement in quality of the solution, others don't (e.g.: Genetic Algorithms).

Another way to "optimize" a kitchen is to minimize the space occupied by the kitchen and thus shorten the distances between the elements within the kitchen, as

demonstrated by Margarete Schütte-Lihotzky with the Frankfurt Kitchen. If you "optimize" the spatial layout, the storage space available will also be minimized. The smaller the storage space, the more often you have to go shopping[4] and the more time you need to buy your supplies. But the more often you go shopping the less you will have to carry at once. Two things are important in every optimization processes, the definition of an optimum and the setup of a prototypical model. Both influence the outcome. In case of the Frankfurt Kitchen the prototypical model included all the elements needed in a kitchen, but it did not take into account the way to and from the store or the weight of the supplies.

What is the optimum of a kitchen? Is it a minimum of space occupied by the kitchen, is it a maximum of storage space provided within the kitchen or do the recipes have to be adapted in order to optimize the kitchen?

An Architect's Vision

Today computer scientista and mathematicians are taking over the field of "optimization" in architecture. Computer scientists concerned with the shortest path and the most efficient space usage test optimization algorithms on office layouts and space allocation planes in buildings.[5]

In fields like urban planning the idea of optimization manifests itself in other ways. Since relations are more complex, the computer gets used to analyse a suggested development (as happened with "Flugfeld Aspern")[6] based on statistical data or, as with the "Space Syntax" method[8], the geometrical layout of a development and its integration into the city. From this analysis conclusions can be drawn and the design can be modified.

At the conference "Advances in Architectural Geometry"[9] at the Vienna University of Technology in September 2009, many speakers, mostly mathematicians, computer scientists and building engineers, used the phrase "the architect's vision" to explain the architecture of the projects they were working on. The "architect's vision" mostly referred to some strange double curved shape, modelled with the help of various 3d programs. A reason why they referred to these projects as "the architect's vision," might be because their involvement with architecture mostly starts when the design is finished.

For Example the Centre Pompidou in Metz, design by Shigeru Ban, was already on site before DesignToProduction[10] was asked to script the fabrication plans for the double curved wooden roof structure. The carpenters, unable to produce the fabrication drawings for their CNC – machine, not the architect, contacted them.

A Simple Tool, a Simple Algorithm

New simple tools such as Processing, Grasshopper, VVVV and many more[11] have been developed over the last years. These tools make implementing algorithms easy, so that also architects can start to develop their own simple algorithms to design kitchens, buildings, even cities. An example for such a simple algorithm is: "how to place as many objects into a defined area so that at least y% of each object can be seen from a specific set of points along a line?"[12]

The optimum in this case is simply defined: "maximize the number of objects fitted into the area so that all objects can see at least y% of the coastline."

The more simply the optimum is defined, the harder it is to know what the optimum for a given area is. For the stated problem a simple heuristic can compute a valid solution for this problem:

1. Define a boundary in which the objects should be placed;
2. Define the viewpoints from which the objects should be seen;
3. Randomly generate a multitude of objects within the given boundary (different positions, different rotations);
4. Sort the generated objects according to their distance to the viewpoints;
5. Select from the first n generated objects, the object that shadows the smallest area from all viewpoints, but is seen at least from y% of the viewpoints;
6. Move this object from the list of generated objects into the list of selected objects;
7. Delete all objects from the list of generated objects that would block the view of the selected object;
8. As long as the list of generated objects is not empty continue at step 5.
9. The list of selected objects will hold a valid solution where each of the objects can be seen from at least y% of the viewports.

When looking at these different solutions (same y values: different number of generated objects, different number of considered objects [n] in step 5; different y values) one can easily see, that they all create similar spaces. No two objects touch each other, the objects are not aligned with the boundaries, the squares form a sort of courtyards and y% of each square can be seen from the viewports. When changing the y value the arrangement of objects starts to become different. It is unclear how close these solutions are to an optimal solution. How different in its spacial qualities an optimal solution would be as well.

If we simplify the problem a little further: "how to place as many equal squares into a given area so that 100% of each square can be seen from one point?" the optimal solution is well defined. Since 100% of each square must be seen from the viewpoint, an optimal solution is the solution where there are no gaps between the view cones of the different squares, and the angle of the view cones are as small as possible > the squares are arranged with the biggest distance to the viewpoint. In the case of a single viewpoint the quality of the solution with the given heuristic depends strongly on the number of considered objects [n] in step 5; the bigger the number of objects [n], the closer the solution is to an optimum. The bigger the number of objects [n], the longer is needed to compute the solution.

In the given example (Image 2) a maximum of 18 objects can be placed into the square. The heuristic seems to perform quite poorly. 9 objects are placed at first, but when playing with the parameters, the outcome can be increased to 13 objects (72%). When setting y to 50% (Image 3) the heuristic manages to place 16 objects out of 25 (64%). If y is set to 0% (Image 4) the heuristic only manages to place 48 out of 100 possible objects (48%; This also applies when considering multiple viewpoints).

It becomes apparent that the spatial difference between an "optimal" solution and a solution computed with the heuristic is negligible. If one wants to change the spatial arrangement, one has to change the rules, how the spatial arrangement is generated, e.g. "how to arrange as many objects so that each object has at least y hours of sun during a day?" If this constraint would be added to the heuristic, the spatial arrangement would change.

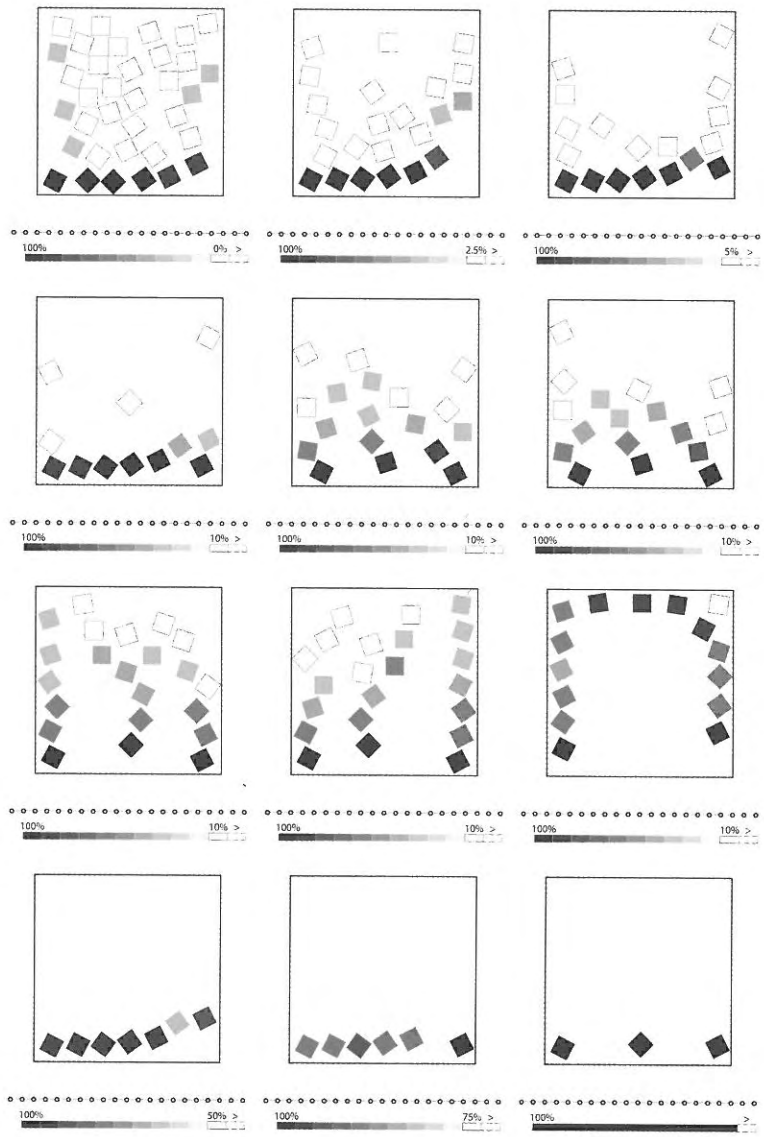


Image 1. Different generated solutions. The minimal y value is always the lower bound of the legend.

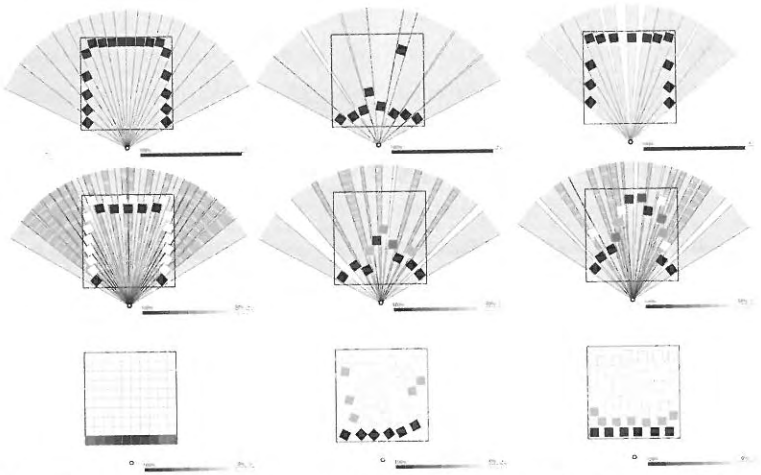


Image 2. y = 100%; optimal solution by hand; first generated solution; best found generated solution;
y = 50%; optimal solution by hand; first generated solution; best found generated solution;
y = 0%; optimal solution by hand; first generated solution; best found generated solution;

Prototypical Models – Generative Models

This proposed heuristic is a prototypical model, which can generate spatial layouts on different scales for different purposes (e.g.: planning a city layout, an exhibition space, etc.). This prototypical model only takes views on a 2D plane into account; any three-dimensional information is lost. The question stated at the beginning can then be reformulated: "How to arrange as many buildings along the coast so that a minimum of y% of the coastline can be seen from every building?" With this interpretation the 2D layout of a city can be planned in a way that would guarantee a view of the coastline and the sea for every floor in every building. The building height itself doesn't make any difference and no conclusions on the building height can be drawn from this interpretation of the prototypical model.

Other prototypical models, such as the calculations proposed by "Space Syntax" to analyse and predict the behaviour of people in new developments, can be used as generative tools.

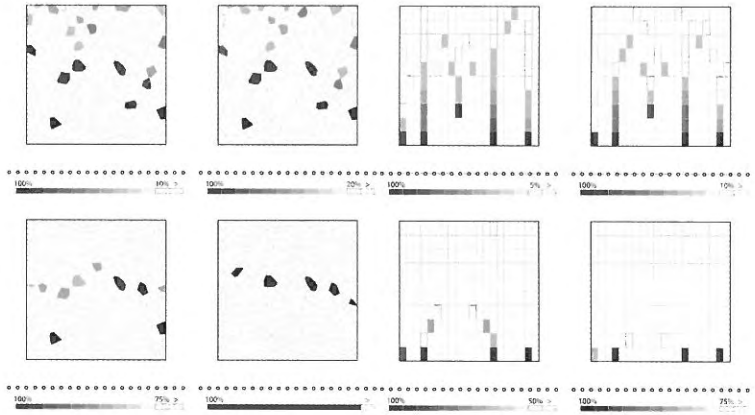


Image 3. Different generated solutions using a randomly generated Voronoi diagram / a simple grid. The minimal y value is always the lower bound of the legend.

One measurement proposed is called "choice". [13] If applied to the circulation of a building this can suggest where it is likely for people to meet spontaneously and start a conversation. The information content of such informal conversations can be larger than in any scheduled meeting. Especially in information driven working environments, e.g. "Google Offices," this form of communication is important and valued by their office designs.

This "choice" measurement can be used as feedback function for an algorithm generating the circulation and spatial arrangement of functions, of a building. This was tried with the project Matrjoschka,[14] a competition entry for an academic interchange.

In this project the vertical circulations and courtyards are randomly placed. Then a heuristic places the requested functions, as squares, based upon the distances to the entrances.

The circulation is then based on this layout. In a first optimization step the placement of the vertical circulations and the courtyards is changed so that rooms like the cafeteria should end up with a high "choice" value, others like an apartment

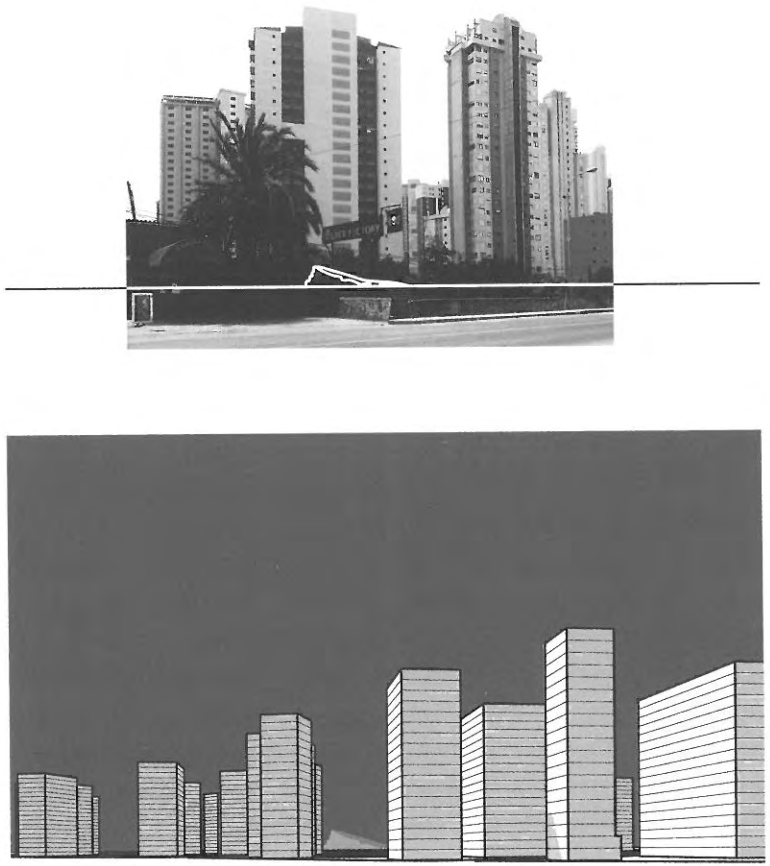


Image 4. Benidorm 2011, Benidorm imagined; At the moment many buildings are over 20 storeys tall in Benidorm. Mostly the view to the coastline is only possible from the upper storeys, so the next building built is always a little higher. With the proposed heuristic a view of the coastline would not only be provided for the upper storeys but the buildings would also only need to be 15 storeys tall in order to accommodate the same built mass.

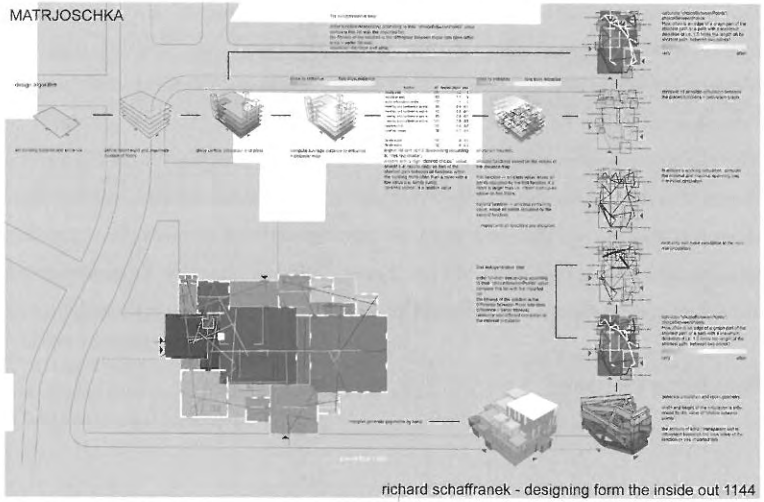


Image 5. Matrjoschka. Poster explaining the planning algorithm for the "Designing from the inside out" competition; design generated with Grasshopper and the SpiderWeb[15] plugin.

with a low "choice" value. In a second optimisation step, some of the possible circulation is removed in order to improve these differences even further. Through these optimisation steps it becomes possible to generate a working room-to-room circulation for a complex building, while the architecture itself does not change. The placement heuristic, which placed squares as representation of functions, defines the architectural form. The way the circulation was generated preferred a room-to-room layout, even though some corridors emerged.

The Task for the Architect

Optimizing architecture can't change architecture. Only changes to the rules that generate architecture will change the architecture. These rules can be due to the creative genius of the architect, leading to "the architect's vision," or due to simple instructions defining an algorithm.

We as architects have to be aware of such developments, and argue what optimization of architecture is about, in order to be a "user,"[16] rather than just a consumer. Then it is possible to go beyond simple optimisation ideas like the "shortest path."

The computer provides the possibility for architects to develop simple algorithms themselves. Those algorithms might not generate an optimal solution but they help to understand the possible outcomes, of the rules defined through the algorithm, which might not be optimised but good enough for architecture. Once setup, the algorithm can be altered and different planning scenarios can be explored.

Simulating Planning!

References

1. "Traveling Salesman Problem," accessed November 2, 2011, http://www.tsp.gatech.edu
2. To compute, to solve a problem by following a set of simple instructions, does not imply the use of a computer.
3. In an undirected graph where every node is connected to every other node the number of possible sequences [N] is given through the formula.
4. This thought was proposed by Prof. Manfred Wolff-Plottegg in one of his lectures at the Vienna University of Technology in 2010.
5. See for example: J A Bland, "Space-planning by ant colony optimisation," *International Journal of Computer Applications in Technology*, Vol. 12/6: 320-328. or. E.K. Burke, P. Cowling, J.D. Landa Silva, and Barry McCollum, "Practice and Theory of Automated Timetabling III," *Lecture Notes in Computer Science 2001*, Volume 2079/2001: 254-273. or. "Recotech," accessed November 2, 2011 http://www.recotech.de.
6. The software "Community Viz" was used after the competition, to analyze different design-decisions during the master-planning process.
7. Clemens Beyer, Gregory Telepak, Josef Lueger, Michael Rosenberger, Manfred Schenk, "Ressourcenplaner Flugfeld Aspern: Interaktive Entscheidungsunterstützung" accessed November 18, 2011, www.corp.at/archive/corp2007_LUEGER.pdf.
8. "Space Syntax Laboratory," accessed November 2, 2011, http://www.spacesyntax.org/.
9. Cristiano Ceccato, *Advances in architectural geometry*, (Wien: Springer, 2010).
10. "DesignToProduction," accessed November 2, www.designtoproduction.ch.
11. Processing is a simple JAVA based editor initiated by Ben Fry and Casey Reas, www.processing.org. VVVV is a simple graphic based editor, mostly for multimedia content, vvvv.org. Grasshopper is a simple graphic based editor for Rhinoceros 3d, www.grasshopper3d.com.
12. When there are no obstacles between a viewpoint and an object, then the object can be seen a 100%. The 100% refers to the angle of the view cone starting at the viewpoint and enclosing the object. The absolute value of the angle depends on the size of the object and the distance between object and viewpoint. If an object blocks the view so that the angle of the remaining view cone is reduced to half of the maximal angle then an objects visibility is reduced to 50%.
13. How often an edge of a graph is part of the shortest path between any two points in a graph. For more details please visit: "Space Group," http://www.space.bartlett.ucl.ac.uk/.

14. Entry for the "Designing from the Inside Out" competition, held by the "University of Bremen, Institut für Informatik und Gesellschaft Abteilung Kognitionswissenschaft" in 2011. For more details please visit: http://cognition.iig.uni-freiburg.de/martinb/inside-out/index.htm
15. SpiderWeb is a plugin for Grasshopper under development, providing simple graph calculations for Grasshopper. "Fakultät für Architektur und Raumplanung, Richard Schaffranek," accessed November 1, 2011, http://www.gbl.tuwien.ac.at/Archiv/digital.html.
16. In the movie "Tron" the main character Flynn is referred to as a user. Unlike the program MCP that rules the digital world, Flynn is capable of creative thought, which gives him super powers. Tron by Steven Lisberg, Walt Disney Productions, 1982.

Image Credits

1. Richard Schaffranek, 2011
2. Richard Schaffranek, 2011
3. Richard Schaffranek, 2011
4. Benidorm 2011, Benidorm Imagined, Richard Schaffranek, 2011
5. Matrjoschka, Richard Schaffranek, 2011

Architecture is more than design proper: it reaches out to related spatial disciplines, but it also lives in the realm of research and scholarship. This book records some of the conditions and modalities through which architects leave behind architecture design to engage with other domains of knowledge: research architecture, history of architecture, and architectural publishing, as well as literary sciences, philosophy, and informatics. It engages questions of disciplinary boundaries between scholarship and design and asks how those boundaries become productive. The authors of these collected essays open up domains of knowledge by pushing, stretching, transgressing and perforating boundaries. They engage productive limits.

The works here presented are a result of a colloquium that took place at the Academy of Fine Arts Vienna in December 2010. The essays result from an iterative process of analysis, intense debate and subsequent refinement that incorporates the fruits of scholarly dialogue within each writer's thinking. They have taken up the understanding that discussion with other scholars allows an intellectual expansion in which the scholar returns to the object of study, only to see it anew. Thus this book engages questions beyond those raised at the colloquium by extending them to new partners in debate; a series of interviews carries forward the concepts and thoughts previously accumulated. Thus, this book itself is a productive limit: it records the perpetual transgression of approaches and positions by opening them up to yet another audience.

SONDERZAHL
ISBN 978 3 85449 381 5