International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue (4) July, Technovision-2014, ISSN 2249–071X

Technovision-2014: 1st International Conference at SITS, Narhe, Pune on April 5-6, 2014

# Algorithms in Architectural Design

**Dr. (Mrs.) Pankaja G. Bagul**
H.O.D, M. Arch-Computer Applications
STES Sinhgad College of Architecture, Pune, India
Email: pankajabagul@gmail.com

**Dr. Nilesh J. Uke**
Associate Professor, Dept. of Information Technology
STES Sinhgad College of Engineering, Pune, India
Email: njuke.scoe@sinhgad.edu

*Abstract* – **In recent years, algorithms in architectural design have been able to transcend their role as frameworks of formalization and abstraction. This has been made possible in a large part by the integration of scripting languages into CAD programs. 'Algorithms' output can now be directly visualized, and through digital fabrication methods this output can be built. This opens up a new role for algorithms as a design tool. As such, they provide the benefits of depth and breadth. On the one hand, their computational power can address processes with a scale and complexity that precludes a manual approach. On the other hand, algorithms can generate endless permutations of a scheme. A slight tweaking of either the input or the process leads to an instant adaptation of output. When combined with an evaluative function, they can be used to recursively optimize output on both a functional and aesthetic level. Yet beyond this, a computational approach to architecture enables the generation of the previously unseen forms that can longer be conceived of through traditional methods become possible, thus opening up new realms.**

*Keywords* – **Computer Science, Algorithms, Architectural Design, Computation, Digital Modeling.**

## I. Introduction

Historically algorithms have been used quite extensively in architecture. While the connotation of an algorithm may be associated with computer science, nonetheless the use of instructions, commands, or rules in architectural practice are in essence algorithms. The rationalization of the design process necessarily involves the use of structured, discrete, and welldefined instructions for the accomplishment of design projects. While many definitions and models of design exist, most agree that "Design is a process of inventing physical things which display new physical order, organization, form and function. However, since no formula or predetermined steps exist which can translate form and function into a new physical entity, design has been held as an art rather than a science. It is considered to be an iterative, 'trial and error' process that relies heavily on knowledge, experience and intuition. The problem with this is not necessarily in the lack of objective criteria, but in lack of rational consistency. If design is to be studied as a process then a series of reasonable justifiable and consistent steps should be established. In contrast, another set of theories defines the design process as a problem solving process as such it can be conceived as a systematic, finite, and rational activity. Problem solving is characterized as a process of searching through alternative solutions to find one or several which meet certain goals. Alternatively a problem space does not always necessitate the identification of a solution as a target, but instead may involve addressing the problem for possible alternative solutions that are not known in advance.

## II. Algorithms

An algorithm is a process of addressing a problem in a finite number of steps using logical operations. It can also be a rationalized version of human thinking. According to Kostas Terzidis, "An algorithm is not only a computer implementation, a series of lines or code in a program, or a language, it is also a theoretical construct with deep philosophical, social, design and artistic repercussions." [1]

While most algorithms are designed with a specific solution in mind to a problem, there are some problems whose solution is unknown, vague, or ill-defined, in which case algorithms become the means for exploring possible paths that may lead to potential solutions. Thus theoretically as long as a problem can be defined in logical terms, a solution may be produced that will address the problem's demands. [2] (Kostas Terzidis 28) Algorithms are represented either in diagrams as Flow Charts or by using computer languages in the form of Scripts or programs.

To put it more precisely, an algorithm is a linguistic expression of the problem and as such it is composed of linguistic elements and operators arranged into spelling, and grammatically and syntactically correct statements. The linguistic articulation serves the purpose not only to describe the problem's steps but also to communicate the solution to another agent for further processing. In the digital medium, the agent is the computer itself.

• Thus an algorithm can be seen as a mediator between the human mind and the computer's processing power.
• The ability of the algorithm to serve as a translator can be interpreted as bi-directional: either as a means of dictating to the computer how to go about solving the problem, or as a reflection of a human thought into the form of an algorithm.

## III. The Computational Turn

The power of computation which involves vast quantities of calculations, combinational analysis, randomness, or recursion, to name a few, point out to new thought processes which may not have ever occurred to the human mind. These 'idea generators' which are based on computational schemes have a profound ability not only to expand the limits of human imagination but also to point out the potential limitations of the human mind. Hence what was inconceivable once is possible even enhanced by computer-augmented human thinking. [1] We are now witnessing a 'computational turn' that countermands the reduction of architectural praxis to the mindless perfection of modelling and rendering techniques. Many architects find the prepackaged design environments to have some limitations. Most architects now use computers and interactive software programs as exploratory tools. All their work is informed by, and thus dependent on the software they are using, which inscribes its logic onto their everyday routines. Such users of software packages have little or no knowledge of the algorithms powering the programs they employ. Most of the interactivity is reduced to a manipulation of displayed forms on the screen, unaware the underlying mathematical calculations behind them. According to Ingeborg M Rocker all of this – even though implemented on computers – has little to do with the logics of computation. [3] (Rocker Ingeborg 16–25)

However architects are now seen devoted to code, in the form of scripting algorithms. While previously architects were obsessed with the reduction of complexity through algorithms, today they are trying to explore complexities based on the generative power of algorithms and computation. For architects and artists like Karl Chu, Kostas Terzidis, George Liaropoulos-Legendre, Mike Silver and CEB Reas, scripting is the means to develop their own design tools and environments. The computer is no longer used as a tool for representation, but as a medium to conduct computations.

Architecture emerges as a trace of algorithmic operations. Regardless of their complexity, the tasks and decisions involved can be formalized as an algorithm. As such, algorithms provide a framework for articulating and defining both input data and procedures. This formalization can promote structure and coherency, while systemically maintaining full traceability of all input..

## IV. Algorithmic Processes in Design

Before Algorithmic architecture has opened up a new field of explorations that aims at a better understanding and exploration of computation's genuine processes and their potential for the production of architecture. It is fascinating to understand how complex architectures emerge from simple rules and models.

### *A. Case study 1: (Studio Rocker22 in spring 2004))*

Define Cellular automata and the Game of Life became the architect's basis for experimentation. The moment a cell turns active, the project code is realised, and thus becomes realisable. Recursive procedures that repeat indefinitely are reading and writing code according to preset rules: line by line, generation by generation. Hereby, each generation impacts the next generation and consequently all following ones. Patterns of code appear and disappear.

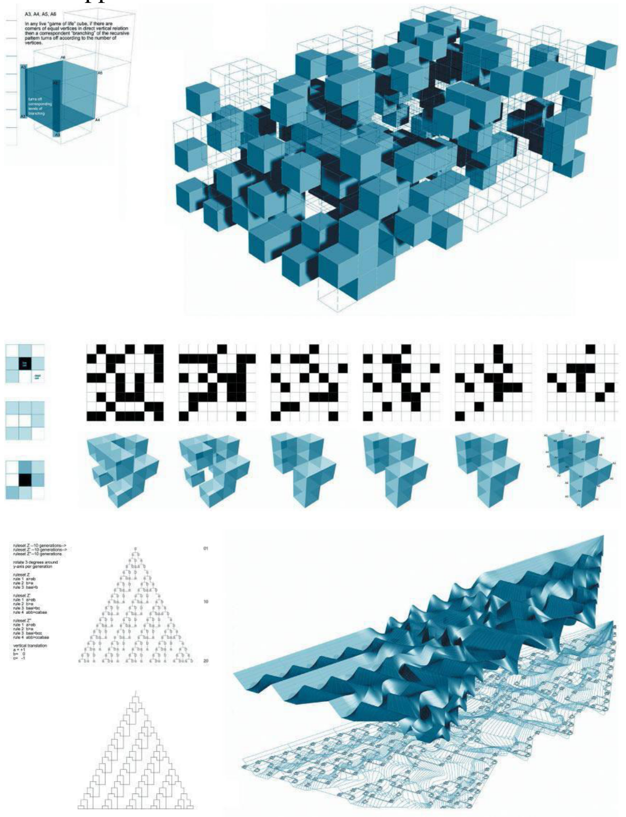


Fig 1, Brandon Williams/Studio Rocker, Recursions, 2004 (Rocker Ingeborg: 23, 24)

Quite different to the Turing Machine, which only uses a one-dimensional tape, Brandon Williams's design is a two dimensional surface. Modes of transposition determine how the abstract code, consisting of As and Bs, realises and thus becomes realisable as surface and structure. Obviously, the chosen mode of transposing code into its expression is just one of many possibilities. Any code's expression is thus always just one of an infinite set of possible realisations. We just have realised the incompleteness of realisation. [3]

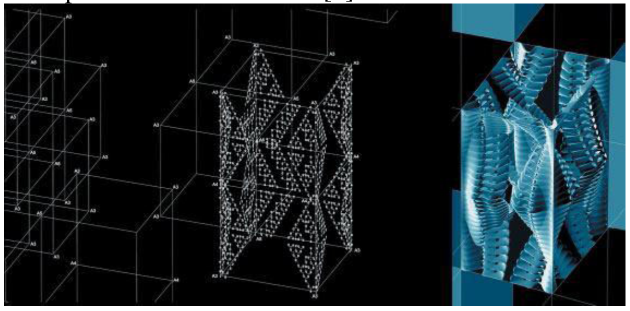
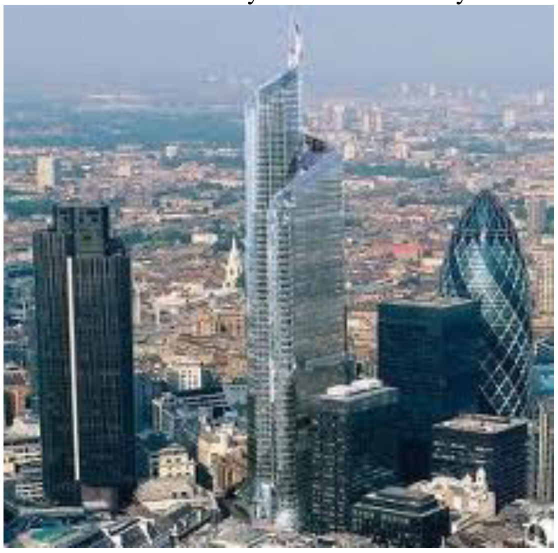


Fig.2. Brandon Williams/Studio Rocker, Expression of code, 2004 (Rocker Ingeborg: 25)

### *B. Case study 2: Double-curved, 'snake skin' façade*

Mathematics is used to create an exchange between schematic design and production. The transition involves the rationalization of complex forms by fundamental geometries since cost is always a crucial limitation.

The design of a double-curved, 'snake skin' façade for the Pinnacle using a single, yet flexible module type in order to avoid wastage in fabrication is an example of how a rather common problem was solved. The designers, Kohn Pedersen Fox, developed a novel approach by embedding analytical algorithms within the design process. This enabled them to integrate optimization routines for constructability and cost efficiency.



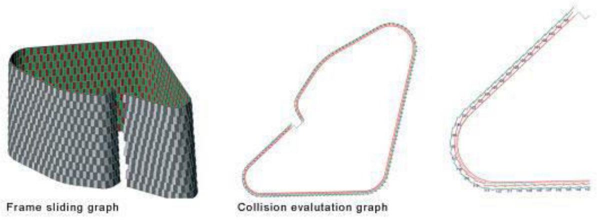
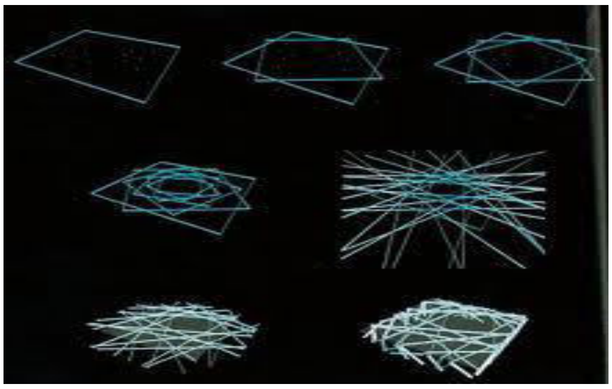


Fig.3. Panel configuration diagram and graphical evaluation representations. (Maria Bessa: 21)

Geometric elements were equipped with 'software sensors' and the parametric model developed 'geometric indicators' around the 'irritated areas' where panel clashes occurred. This analytical problem-solving approach was incorporated into the design process, rather than being a process of validation applied to a finished design. The application of algorithms for the organization of proliferated material components over a predefined form is increasingly used in the design of structural skins. In this way of working, the most critical factor to be incorporated in the design process is structural stability. Algorithms have been developed that inform the design in a feedback loop as the design is developed. [4]

### *C. Case study*

The prime objective of the Serpentine Pavilion of Toyo Ito and Cecil Balmond was the integration of a structural system as an integral function of the skin. Since the form of the structure was given as a simple rectangular box, it was necessary to find a technique for subdividing the skin in order to create a structure. Fractals are chosen as the process for subdivision since they enable the design of a structural system based on a square shape that could be propagated to infinity.



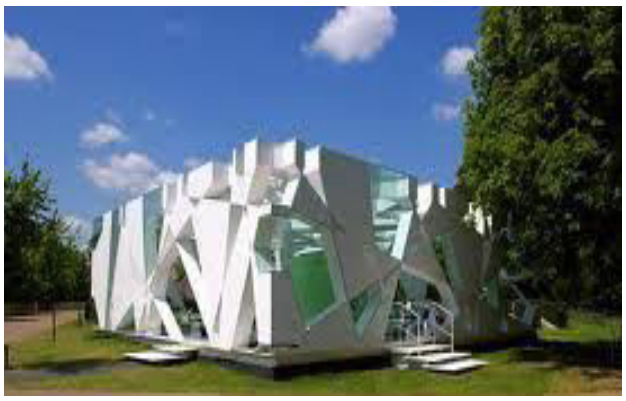


Fig.4. Serpentine Pavilion algorithm (Cecil Balmond 132)

The output of the algorithm was a two-dimensional pattern, and the thickness of each beam was defined according to the distribution of stresses along the surface. [5]

### *D. Case study 4*

The algorithm developed by Chris Bosse for the design of the National Aquatics Centre in Beijing, the 'Water cube', moves the algorithmic process one step further. In this design a single material system produces structure and at the same time defines space. Structural stability is a priori assured by the design choice itself –the formation of a stable configuration of the geometry of bubble packing that also occurs spontaneously in many natural systems. [6]

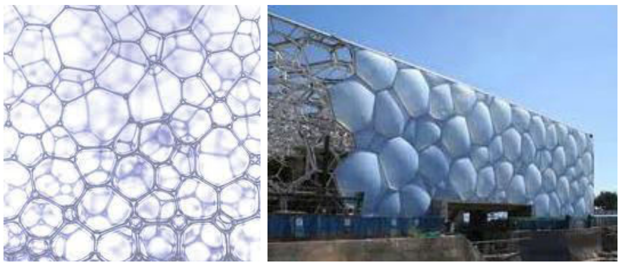


Fig.5. National Aquatics Centre in Beijing, the Watercube (Chris Bosse 2008)

There are few known examples of architectural forms that are completely generated from scratch by algorithms. Marc Fornes' research interests lie in the generation of built prototypes that are originated entirely from a set of codes. Fornes is seeking to evolve new algorithms for a more exploratory 'non-deterministic' generative process. His Aperiodic Vertebrae explores complex and aperiodic stacking by successively subdividing four primitives over many generations. The description of form by the primitives provides flexibility when it comes to manufacturing. Depending on fabrication requirements, the primitives can be described by planes or even double-curved components. This process also looks for efficiency in assembly and in manufacturing cost, and a nested three-dimensional array of identical pieces was the optimum solution. [4].

## V. Discussion

Matteo Lo Prete finds an analogy between architectonic planning – architectural design and the use of algorithms. In a sense that there are central ideas and, starting from them, the final solution is totally free from any conditioning. According to him Shape, final objective of a calculus algorithm, is an entity devoid of contamination, free from faults and answering at best to initial criteria. Surely, from some point of view, this could seem to be an expected process, free from any interest concerning creativity, sterile from the cultural point of view, uniquely based on calculus. On the contrary, due to the presence of what Weinstock defines as "Emergence", the unexpected solution is always there. And this is the true potential, the real innovative advancement in the use of algorithms for generating shape. As algorithms do not base themselves on predefined sets of geometrical shapes but on simple rules, the formal result is unexpected, similar to that reached through a creative process. [7] (Matteo Lo Prete, Noemalab)

Arup is among the international practices that are moving in this direction. Within Arup there is a small team named Advanced Geometry Unit consisting of an architect (Charles Walker, representing the head of the team work), three engineers, a mathematician and a physicist. The aim of the group is to study a solution for projects where particularly complex geometries are required, or where they pursue a more specific approach in the use of algorithms for architecture.

Probably this is the emerging stereotype of the new architects' studio, where a large number of employees, designers, consultants are replaced by a few specialized number of persons, coming from very different fields. The main activity is to identify deep problems of a project, leading to the development of application tools for their solution. Programming is another fundamental part of the new architectural planning, basing itself on rules established from time to time with the contribution of the other two sciences. The future of architecture, thus, is a multidisciplinary and united group where the architect recognizes he does not have the sufficient knowledge in some fields and agrees on cooperating with a series of more qualified professionals.

## VI. Conclusion

Nearly forty years after the arrival of computer in the field of architectural planning, we can say that it's time for a new stage, where architects can act with a more mature and rigorous attitude towards tools which have slowly refined as time went by. If during the first testing period, coinciding with years around the end of the millennium, it was possible to understand certain approaches, now it is necessary to consolidate what has been tested into a compact and rigorous technical and methodological fund. Thus hence forth, digital modelling will not be seen as a process where the architect selects the geometric typology more suitable for his purposes, using this tool to elaborate shape, without being aware of the true potential. Processes generating shape must be the object of studies which is always more profound and conscious. This will give modelling control and conscious decision making back to architects. As in the earlier phases one had to choose from the library set up by the creators of the software, who often did not know the specific needs of every user. Greg Lynn has aptly said, Thus algorithmic architecture should be seen as a new stage for architectural design, where the architect must necessarily be aware of underlying logic of tools he uses. This seems to be the only way that architects will be able to get back the control of the design process.

The algorithmic design approach can deal with a large number of competing constraints simultaneously, and can be used to explore numerous differently weighted design solutions within time frames that are just not economically feasible by more traditional methods. Though algorithms are becoming widespread in many design and fabrication industries, perhaps their best use is in architectural design, where they can enable designers to work in intuitive and nondeterministic ways. Thus new and innovative designs can be produced that achieve structural and environmental performances that were once considered to be post-design optimization processes.

The promises and limits of such explorations are diverse. The question remains open as to whether the turn to computation will reconfigure architecture to such an extent that a new kind of architecture will emerge.

## REFERENCES

[1] Kostas Terzidis, Algorithmic architecture, London, Architectural Press, 2006. 18 Print
[2] Kostas Terzidis, Algorithmic architecture, London, Architectural Press, 2006, 28 Print
[3] Rocker Ingeborg M, When code matters, Programming Cultures, Architectural Design, Vol 76/4, , July/August 2006. 16–25 Print
[4] Bessa Maria, Theoretical Meltdown, Algorithmic Design, Special Issue:, Architectural Design, Volume 79/ 1, 2009 120–123,. Print
[5] Cecil Balmond, Geometry, Algorithm, Pattern: The Serpentine Pavillion 2002, Digital Tectonics, ed Neil Leach, London, Wiley-Academy, 2004. 132 Print [6] Lynn Greg "Programming cultures: art and architecture in the age of software", Mike Silver(ed), Architectural Design: London, Wiley-Academy. 2006, 55. Print
[7] Matteo Lo Prete , Generative algorithms , Pier Luigi Capucci, Noemalab, web 07/2008
[8] Berlinski David, 'Introduction', The Advent of the Algorithm, New York, Harcourt, 2000, 9. Print
[9] Kostas Terzidis, Expressive Form: A Conceptual Approach to Computational Design, London, Spon Press , 2003, 71. Print
[10] Stiny, G., and Gips, J., Algorithmic Aesthetics: Computer Models for Criticism and Design in Arts, Berkeley and Los Angeles, University of California Press, 1978. 16. Print
[11] Cardoso Daniel Llach, Shift+Design: Scripts and Other Design Artifacts, International journal of architectural computing issue 01, volume 08, Jan 2010, 15. Print
[12] Magdy M. Ibrahim and Robert J. Krawczyk, "Generating Fractals Based on Spatial Organizations", Computer graphics & geometry, Volume: 8.2, 2006 : 3-15. Web. 3-9-2010
[13] MindTools.com., "Flow Charts", Rebuilding Morale. 2011. Web: 12 Jan 2012.