The MiranScript
Intuitive Calculations

Stylianos Dritsas
Massachusetts Institute of Technology, Cambridge, USA
http://dritsas.net

Abstract. *This paper documents the implementation of a computational solution for a design project and presents a series of thoughts around the topic of architectural design as an act of intuitive calculation.*

Keywords. *Design Computation, Calculation, Scripting, Rapid Prototyping, Digital Fabrication.*

Introduction

The role of calculation, as an aspect of the design process, is becoming increasingly significant due to the progressive integration of computation into the architectural practice. The pragmatic reasons for this emergent need can be traced in the concurrent expansion of the possibilities for more complexity, which computation keeps pumping into the design domain and its rapid accumulation by the community. The nowadays casually complex geometries illustrate a situation where computation broadened the domain of design expression, yet simultaneously imposed a regime of esoteric computational laws upon for the architecturally trained designer. Calculation in this case became either a life-vessel wrapped in graphics or a necessary evil to be bared with, ignored or voided by all means.

Yet calculation was and still is an integral part of design thinking: in a rather more implicit nature. The need for an explicit understanding of calculation cannot be but another domain for research in design. The goal set for this project was to explore the potential of integrating calculation intuitively into design. The attempt did not target to hide calculation into the visually driven process of design –even though this was proven occasionally necessary– but rather to allow calculation to inform, influence and augment the thinking process.

The topic of design and/as calculation has been explored from various perspectives which significantly stimulated the design of the computational solution presented in this paper. Namely, “Design by Numbers” (Maeda, 1999) illustrated the modes of employing computation as an expressive

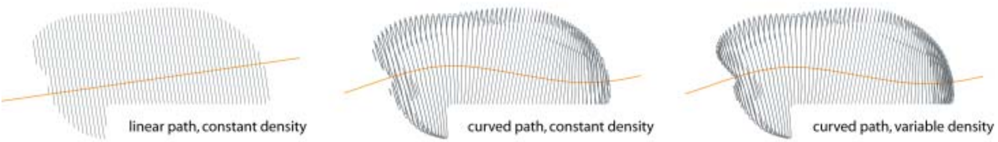

Figure 1. The MiranScript conceptual diagram.

Figure 2. The prototype produced by five axes CNC milling (Photo: courtesy of dECOi Architects).

and intuitive medium for design. “How to calculate with shapes” (Stiny, 2000) presented a highly provocative relationship between design and calculation as being essentially the same process. The work of Axel Kilian (2004) in bidirectional design systems exemplified the methods of bridging between design and computation by linking multiple design domains. Finally, the architectural work of Mark Goulthorpe (2000) hinted for intuitive modes of computational design.

Project description

MiranScript was a computation solution developed for the design the “Miran Galerie”, part of an interior renovation of a Parisian fashion atelier (Goulthorpe, 2004), commissioned to dECOi Architects. The script was introduced in an intermediate phase of the design, for automating the processes of rapid prototyping and digital fabrication (Figure 2).

The design can be described as a series of encapsulated envelopes: the original hosting space, the main gallery hull and a floating exhibition cell.

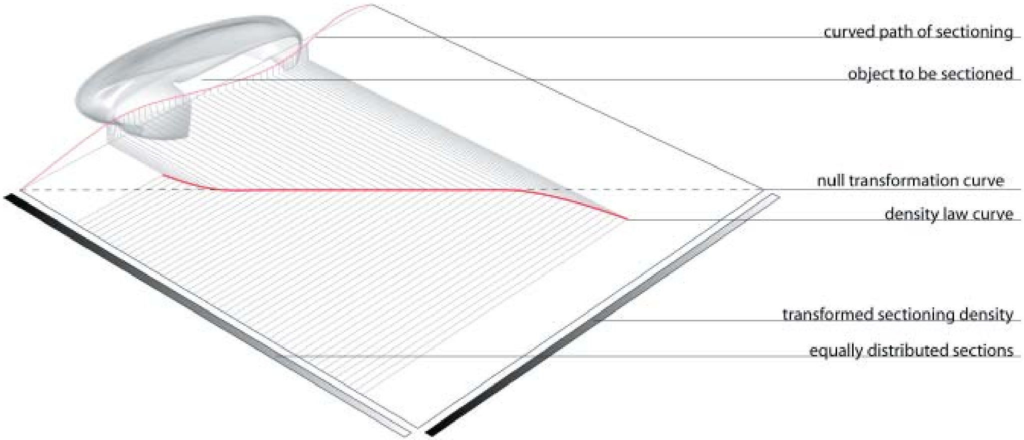

Figure 3. The geometric inputs of MiranScript: the initial shell, the sectioning path and density curve.

The design had already a well defined tectonic and fabrication methodology Both the main and the floating hulls were to be fabricated in plywood by means of CNC milling of the flat rings and arcs produced by contouring the original envelopes. Spacing between the contours was identified as an effective method of controlling the cost of fabrication while integrating the technique with a unique aesthetic expression (Goulthorpe, 2004).

Design & Implementation

MiranScript was founded on the scripting platform produced during the implementation of a previous rapid prototyping script (Dritsas and Kashyap, 2005). In its initial implementation it also shared the same target of bridging between digital sketches and physical artifacts by means of computation. The script, or better the collection of scripts, was written in Visual Basic™ scripting language, hosted by the Rhinoceros™ CAD application.

In essence, the script's role was to eliminate digital draftsmanship by collapsing the process of geometric modeling. This process involved the decomposition of the original envelopes by contouring, thickening the generated members, appending construction details and preparing them for a specific machining platform. The script tackled each step of the process individually and eventually linked them all together in a digital production line.

The nature of the original script was one of pre-calculation since, once it was implemented; it already contained the resolution of the whole process. The scripted black-box requested a set of NURBS surfaces and a list of user preferences, which defined the inputs (Figure 3), and produced a series of specifications fitted to the peculiarities of a fabrication machine, which defined its outputs.

Areas of experimentation

Yet as an application of computation for design, the initial implementation of the MiranScript seemed to be not that innovative because in essence it capitalized on the raw power of computation and left the architectural and computational design processes intact, or at most indirectly linked. After realizing the hiatus between these modes of design, the next implementation of MiranScript set as its main goal to make the process more creative and fuse design and computation. For attempting this, the whole process and its initial abstractions had to be reconsidered.

Revisiting the process evolved on two levels of inquiry: a conceptual and a technical. With the knowledge gained from the results of the first implementation, it seemed that the method of contour cutting as a design and fabrication process proved to be troublesome for multiple reasons. While contouring seemed to be the most cost effective and controllable process, compared to molding making and casting curved patches (Goulthorpe, 2004), it also proved to be problematic in accurately reproducing the original complex geometry. For instance, the final reconstructed-by-contours form was rather distorted, because of the curvature characteristics at the end parts of the gallery.

Simultaneously, the method contouring seemed to be unrelated to the dynamism of original design. Yet, the first step of integration was made by the introduction of spacing between the ribs. Spacing established a relationship between the original form, its visual feedback/experience, the fabrication technique and cost control. This was a conceptual area identified for further development and experimentation.

Procedural abstraction

Contouring is a degenerative geometric transformation which is founded on the properties of the section. A section effectively removes a component for a scalar dimensional vector of a given set. In our case, the three dimensional geometry was decomposed in series of two dimensional profiles. A section as an operation has at least two operands, even if one or both of them are void, which do not need to have equal dimensional order. Contouring in turns is defined by the following essential parameters: the sectioning operands, a path along which sections are taken and a spacing function that controls the distances between consecutive sections.

This generalization allowed us to rethink the project and become more conscious of our methodologies. MiranScript used this abstract definition and employed the original geometry as the first operand, a generalized spatial curve as sectioning path, a non-linear spacing function and either a plane or a rectilinear volume as the second sectioning operand. The sudden jump backwards in abstraction proved the CAD tools in our disposal unable to tackle this new problem and necessitated the design of novel computational system for allowing us move forward.

Script description

The script used a generic NURBS curve and along its parametric domain laid a series of registration points. The tangent vectors along the registration points on the curve were used as normal vectors for the cutting planes. The planes were used for extracting profiles from the original geometry which were later extruded for creating solid ribs (Figure 4).

In a subsequent enhancement, instead of generating single planes, the script created pairs of offset planes, describing the boundaries of infinite rectilinear volumes, and used them for extracting pairs of profiles from the original geometry. The ribs generated by this process had curved side surfaces in both directions as opposed to the first extrusion-flavored ribs (Figure 4).

The distribution of the registration points on the contouring path was controlled by an adapted version of a law curve. The same script with same parameters was used for generating rod-joints between the ribs, special registration flooring for accurately positioning the ribs and eventually handled the necessary steps for bringing the design in a fabrication ready stage (Figure 4).

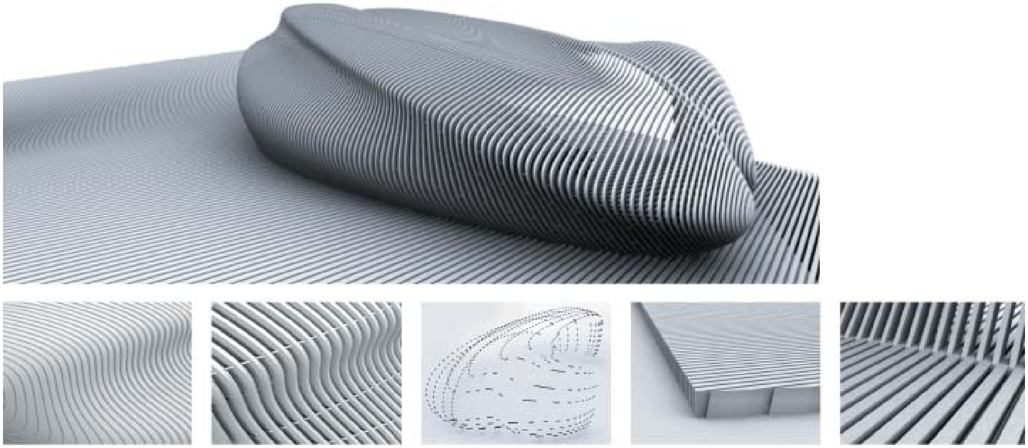

Figure 4. Details of generated geometry.

Implementation Issues

The main problem of employing degenerative, as opposed to constructive, geometric processes can be found in the issue of multiplicity of outputs. In essence, the operation of section proved to be highly adventurous, because both the amount and nature of the output geometry were variable. For instance, a section of a sphere and a plane will always produce a single circular profile. This is not always true with more complex geometries, as in our case. Apart from the fact that there were both cases of singular and multiple generated profiles from a single section; there was also the possibility of each being of different geometric type: a combination of a curve, a line and even a point.

Moreover, the operation of sectioning NURBS surfaces is usually performed by approximation which proved to be highly troublesome in producing "dependable" geometry. Subsequent transformations applied on the generated profiles occasionally produced numeric errors which unnecessarily stalled the process.

Controlling the density of the sectioning planes on their curved path was an area of interface design. The initial implementation exploited the interpreted nature of the programming language and requested a mathematical parametric expression which was evaluated by the script for spacing the plane registrations along the axis of contouring. This approach even though it was the most accurate method of inputting a distribution function in the script, meant to be extremely unintuitive. The next version of the script replaced the formula with a graph. The xy-plane of the default coordinate system of the CAD application was used as a graphing-board, and the user could simply draw the distribution function. The graph was certainly more architecture-friendly, but still it was rather visually detached from the contouring path. Eventually, the solution that was used in the final implementation further reduced the accuracy and generality of for the distribution method but became much more intuitive and effective. The final custom made law curve was drawn based on the proportions and orientation of the contouring path. A projection of it was generated on the xy-plane and its boundaries were used for creating a control box. The user had to simply draw a distribution curve just by the contouring axis.

Eventually, fabrication issues had to be tackled. Initially the script was written for handling prototyping by means of three dimensional printing, such as SLA. For this purposes the script was suited for producing solid watertight models. The problems related with this process were all produced by the numeric instability of the sectioned parts. These were resolved by quality control procedures on the section products and local corrections. In a later phase the script had to be expanded for being able to handle CNC machining of the flat profiles. Our target platform was a five axes machine capable of accurately cutting the curved edge of the ribs. The script had to incorporate the machining tolerances of the specific platform, as for instance to produce double layered machine paths and rectify the joint registrations to fit the angular tolerance of the mechanic drill (Figure 5).

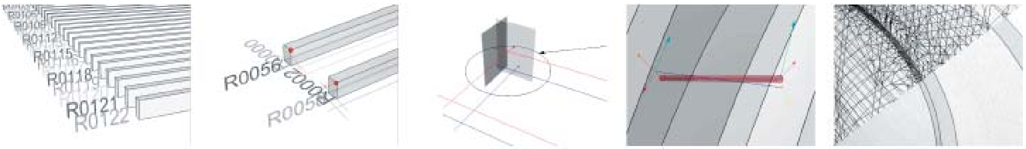

Figure 5. Prototyping and fabrication moments.

Evaluation

The effects of employing computation for the design of Miran gallery were multiple in number and variable in nature. A rough categorization would distinguish between the effects of employing computation for its executive benefits and those in effect of involving computation in the design thinking process. This segregation of modes though became increasingly difficult to sustain as the process became more integrated.

Initially, the time between the phases of sketch design and detailing was collapsed. The regeneration of the geometry from start to end took only a few minutes in contrast to days of hand modeling. The complexity of the architectural design project though, was proportional to the complexity of its computational implementation. The nature of this relationship cannot be characterized either as linear or as exponential because it is highly dependent on qualitative characteristics of the design. Nevertheless, the implementation of a computational solution for a design project was time consuming, especially because of the intention of massively attacking the problem space from the beginning till the end. The approach of solving sub-domains of the problem and eventually linking them was sought as the most flexible and time-effective.

The process was simplified and the complexity of modeling curved geometry was minimized, if not eliminated. The script requested a number of parameters and produced the geometry with little or no exceptions. The effects of black-boxing the process were rather cognitive in this case. A level of abstraction was encapsulated and rendered as solved, requiring little or no further thought investment. Subsequently, the procedural wrapper allowed either higher level or different kind of abstractions to be investigated.

A possible problem of this methodology though may appear on the occasion of a desired redefinition of the initial abstractions captured in the form of a computational application. It is not always possible to estimate beforehand whether rewriting a script is preferable rather than modifying an existing one. In cases of complex hierarchies of computational constructs it usually becomes more difficult, if not impractical, to rewind. For instance, in the specific implementation of MiranScript certain aspects of the NURBS geometry sub-system were proven to be insufficient for our purposes; going backwards and implementing a custom geometric system was not an affordable option. This phenomenon can be described as "computational dissonance": the biases inscribed in consecutive implementations are carried along derivative work.

The process became explicit and therefore stable and measurable. The script would always exhibit the same level of stability in each execution. It also operated on the geometry on a different level than a CAD modeler, by being able of tracking complex numeric data, such as curve quality; measure and correct non-visually identifiable geometric discrepancies; and report explicit information when exceptional conditions were met. The absence of direct metric in complex geometries underlines the necessity of calculation in contemporary digital design projects. Most of the problems but also most of the novel solutions created for this project originate from the inscribed visual mentality of architectural design which may be enhanced, leveraged or assisted by non-visual or numerically driven methods of computation.

The design, as a system, became reusable and it could be applied in any arbitrary initial geometry and produce either its prototyping or fabrication specifications. This effect has an impact on the notion of the architectural design product as a singular object. With this potential in hand, design comes closer to a mass producible, yet locally adaptable, product rather than a unique master piece. Of course the point of this remark is not related to the aesthetic of the design, which cannot be diminished in a computational constructs.

Though, the script encapsulated a blurry region or a design family, rather than a single instance.

Eventually, the domain of the design space became more accessible. The rapidness of automated modeling allowed investing more time in exploring among design variations. After the first stable implementation the script became an exploratory tool while retaining its executive characteristics. The exploratory nature of the script was not one of a random search in the dark but rather an informed or informeable one, due to its underlying fabrication knowledge encoded as geometric rules. In essence, we learned how the system behaved and which conditions were feasible or not. Eventually, the same script allowed us to increase the resolution of the search, lock down to a solution subspace and fine tune the final design candidate.

Conclusion

In retrospect, the most interesting aspect of MiranScript was that it opened up the design exploration, instead of locking it. The expansion though was not one without control or logic. The numerically driven process shed light in possible design paths and simultaneously hinted about the implications of moving in one direction or the other. Even though the script was modestly introduced as an automation operator, it eventually allowed us to employ it as an expressive tool for forming a unique aesthetic result. Whether these effects are result of the creative nature of design or evoked by the computational solution is difficult to categorically state. Nevertheless, if there was a single point to be kept from this research that would be this indeterminate condition or better stated an almost unavoidable creative co-evolution between calculation and intuition. In this paper this computational design methodology or, if you may, aesthetic was delineated as one of the multiple approaches for exploiting the potential of computation and intuitively incorporating fragments of calculation in the process.

Aspects of design that MiranScript did not have the chance to touch define some possible areas for future research. For instance, incorporating economic hints and allowing dynamic cost control can be an area for further development. A great challenge would also be to incorporate live feedback mechanisms in the process such as the original sketch is informed and adapted according to the results produced by iterative processing and evaluation. These modes of incorporating the element of calculation in the design process seem as natural steps in the development of architectural computation.

Acknowledgements

I would like to thank Mark Goulthorpe (dECOi Architects & Associate Professor of Architecture, MIT) for his invaluable encouragement and remarkable intuition that contributed in the development of MiranScript. I would also like to express my gratitude to Alexandros Tsamis (SMArchs scholar in Design and Computation, MIT) for his always sharp remarks and accurate feedback that helped making the application more design sensible.

References

Dritsas, S., Kashyap, S., (2005), Scripted Mockups: Bridging Digital and Physical through computation, CAADRIA 2005, 10th International Conference of the Association for Computer Aided Architectural Design Research In Asia, India.

Goulthorpe, M., (2004), Miran Gallery, FEIDAD 2004 Competition Entry, http://decoi.mit.edu/miran/

Maeda, J., (1999), Design by Numbers, Cambridge MA, MIT Press.

Kilian, A., (2003), Fabrication of partially double-curved surfaces out of flat sheet material through a 3d puzzle approach, ACADIA 2003, Indianapolis.

Kilian, A., (2004), Linking hanging chain models and fabrication, ACADIA 2004, Cambridge ON,

Canada.
Stiny, G., (2000), How to Calculate with Shapes, Cambridge MA, MIT.