C. M. Herr, N. Gu, S. Roudavski, M. A. Schnabel, *Circuit Bending, Breaking and Mending: Proceedings of the 16th International Conference on Computer-Aided Architectural Design Research in Asia,* 493-502.

DATA FLOW AND PROCESSING IN THE COMPUTATIONAL FRACTAL ANALYSIS METHOD

MICHAEL J. OSTWALD,[1] JOSEPHINE VAUGHAN[2] and STEPHAN K. CHALUP[3]
The University of Newcastle, Callaghan, NSW, Australia,
1. michael.ostwald@newcastle.edu.au,
2. josephine.vaughan@newcastle.edu.au,
3. stephan.chalup@newcastle.edu.au

Abstract. One of the few quantitative methods available for the consistent analysis of architectural form is the 'box-counting' approach to determining the approximate fractal dimension of a plan or elevation. In its computational form this method has been used to analyze the plans and facades of a wide range of buildings. The data points produced are synthesized by the software into a series of fractal dimension (*D*) values that are in turn compiled in various ways to produce a series of composite results describing a complete building. Once this process is complete the data may be coded with additional information producing a set of mathematical results that describe the form of a building. This paper offers the first complete description of this important analytical process from the point of view of information flow, algorithmic operations, review options and data magnitude. No previous paper has detailed the full scope of the data used in the computational method, or the way in which various stages produce different types of outcomes. The purpose of this paper is to elucidate the way in which this particular computational method, drawing its inspiration from the complexity in natural systems, may be used to process different types of information and produce various forms of quantitative data to support architectural design and analysis.

Keywords. Fractal analysis; computational analysis.

1. Introduction

There are relatively few methods available to support the consistent and quantifiable analysis of architectural form. While a small number have been proposed over the last few decades (Krampen, 1979; Stamps III, 1999) only one, the fractal analysis method, has been adopted by a wide range of scholars and has been applied in more than 50 cases. There are two related versions of this analytical approach both of which share a common algorithmic core. The first, the "manual method" (Bovill, 1996; 1997) which is produced by hand on transparent overlays, identifies a limited range of results for an isolated image. The second method, the "computational" variation (Ostwald, Vaughan, Tucker, 2008; Ostwald and Vaughan, 2009a) produces an individual result for an image from a comprehensive review of the information present in a plan or elevation. These results can then be combined together in various ways to analyze complete buildings or sets of works. While there are several other variations which merge aspects or qualities of the two (Lorenz, 2003) these remain the most widely published methods. However, despite their relatively widespread application, a detailed account of the computational process which underpins them – from the point of view of the type of information required, its method of processing and its output forms – has never been published. In response to this situation, the present paper provides a brief overview of the theory and background to fractal analysis, before developing a detailed, information-based account of the operations in the computational variation. This description not only charts the influence of several limitations in the process, it identifies the scale and complexity of the data present in different stages and during key operations in the process.

2. Background

Fractal geometry may be used to describe irregular or complex lines, planes and volumes that exist between whole number integer dimensions. This implies that, instead of having a dimension, or *D*, of 1, 2 or 3, fractals might have a *D* of 1.51, 1.93 or 2.74 (Mandelbrot, 1982). There are a range of methods for determining the approximate fractal dimension of an object and one of the most common, the box-counting approach, was developed by Bovill (1996; 1997) for the analysis of architecture. Bovill's work demonstrated that the fractal dimensions of line drawings of elevations of different buildings could be determined using this approach. In the last 15 years the manual variation of this method has been widely used for the analysis of architectural and urban designs (Bechhoefer and Appleby, 1997; Makhzoumi and Pungetti, 1999; Burkle-Elizondo, Sala and Valdez-Cepeda, 2004). Ostwald, Vaughan and Tucker (2008) developed a computational variation of this method which has since been consistently applied to the analysis of more than 40 famous designs by a wide range of architects including Frank Lloyd Wright, Eileen Gray, Peter Eisenman and Kazuyo Sejima (Ostwald and Vaughan, 2009a; 2009b; Vaughan and Ostwald, 2009).

3. Computational fractal analysis method

The computational fractal analysis method may be visualized as a process diagram delineated by four types of information flows or states (inputs, processes, operations and outputs) (see Figure 1). The procedure itself occurs over two, three or four stages, depending on the type of results required. Thus, for example, at each of the three main procedural stages {1.0, 2.0, 3.0} there is an opportunity to generate different types of results {1.4, 2.2, 3.3}.

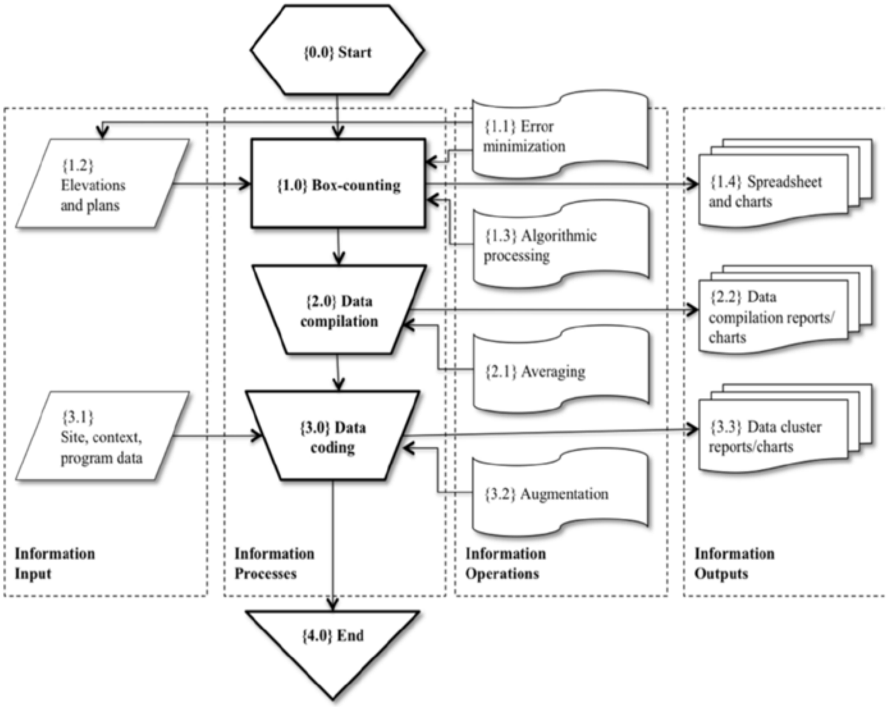

Figure 1. Computational Fractal Analysis process flow chart.

The first stage {1.0} in the process involves the application of the box-counting procedure to a selected image (input data). However, before this has even occurred there is a critical error minimization process {1.1}. The fractal analysis method has several known limitations that can cause errors in the results. The four common problems with the method are associated with balancing "white space" and "starting image" proportion, line width, scaling coefficient and moderating statistically divergent results (Lorenz, 2003, Foroutan-Pour, Dutilleul and Smith, 1999; Ostwald and Vaughan, 2009a). No single solution exists to these problems but by closely controlling the quality of the input data (plans and elevations) and by consistently applying these rules, three of the four limitations are minimized to such an extent that they have little or no impact on the results. In practice, this means that the input data frequently has to be redrawn or reconstructed using the same graphic conventions for all images to ensure that the results are accurate and consistent. The impact of the fourth limitation, statistical divergence, is currently checked after the box counting stage to see if it is significant {1.1}. If it is significant, it is possible to manually adjust the number of grids used for the comparison to ensure that the results are valid.

The box counting procedure {1.0} may be undertaken either by specially authored software *(Archimage)* or commercial software *(Be-noit)* or a combination of both. In each case, the software uploads the input data {1.2}, which is usually a plan or elevation of a building. A grid is then placed over the drawing and each box in the grid is analysed to determine whether any lines from the façade are present in it (see Figure 2). Those grid boxes that have some detail in them are then marked. This data is then processed {1.3} using the following numerical values:

$_{(s)}$	=	the size of each box in the grid
$N_{(s)}$	=	the number of boxes containing some detail
$1/_{s}$	=	is the number of boxes at the base of the grid

Next, a grid of smaller scale is placed over the same façade and the same determination is made of whether detail is present in the boxes of the grid. A comparison is then made of the number of boxes with detail in the first grid ($N_{(s1)}$) and the number of boxes with detail in the second grid ($N_{(s2)}$). Such a comparison is made by plotting a log-log diagram ($\log[N_{(s)}]$ versus $\log[1/_{s}]$) for each grid size. This leads to the production of an estimate of the fractal dimension of the façade; actually it is an estimate of the box-counting dimension (D_b) which is sufficiently similar that most researchers don't differentiate between the two. The slope of the line (D_b) is given by the following formula:

$$D_b = \frac{[\log(N_{(s2)}) - \log(N_{(s1)})]}{[\log_{(1/s2)} - \log_{(1/s1)}]}$$

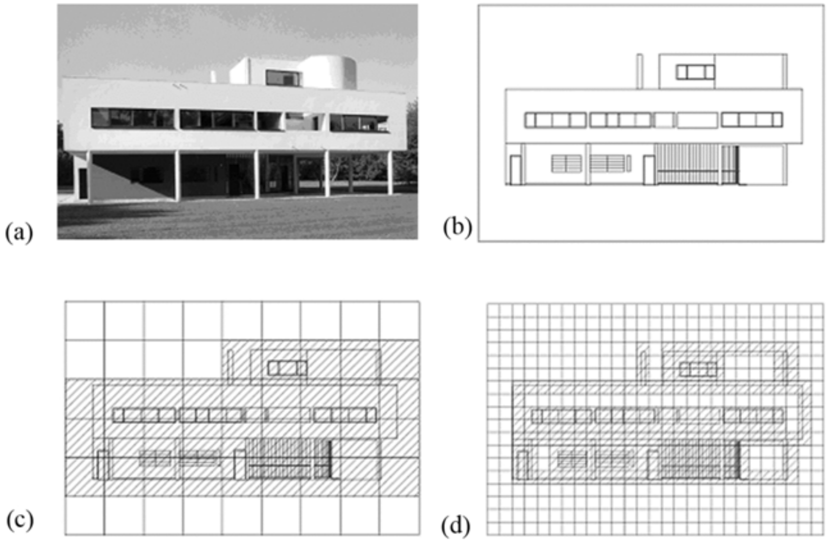

Figure 2. Example of the start of the box-counting approach (a) Villa Savoye, (b) façade elevation, (c) stage 1, grid, (d) stage 2 grid.

For the analysis of a house, say Le Corbusier's Villa Savoye, each elevation will require around 13 or 14 iterations (grid comparisons) to produce a reasonable result whereas for a large historic building (say the gothic revival British Houses of Parliament) over 100 grid comparisons may be needed for each façade. Once this is complete, a spreadsheet is then automatically produced (see Table 1) containing all of the collected data on the elevation or plan. In the example, Table 1, the smallest grid size used for the analysis was 5 pixel boxes, and there were 5583 of the boxes in this grid size containing data. The largest grid size was 425 pixel boxes, and there were only 6 data points found in this grid. The results in the table are then charted in a log-log scale producing the fractal dimension results *D* (see Figure 3) {1.4}.

Table 1. Example results for box-counting the Villa Savoye façade: $D_{(Elev)} = 1.441$.

Iteration	Box Size	Box Count	Iteration	Box Size	Box Count	Iteration	Box Size	Box Count
1	5	5583	**6**	27	610	**11**	150	53
2	7	3681	**7**	38	379	**12**	213	29
3	9	2605	**8**	53	234	**13**	301	16
4	13	1539	**9**	75	157	**14**	425	6
5	19	955	**10**	106	82	-		

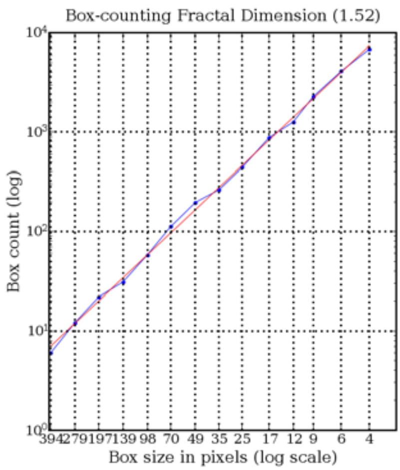

Figure 3. Example Log-Log diagram for the Villa Savoye façade: $D_{(Elev)}$ = 1.441.

The following stage {2.0} of the procedure is concerned within taking isolated D results for an individual elevation (or plan) and combining them to produce a set of average or typical fractal results which reflect characteristic levels of two dimensional visual complexity in a complete design. In the first part of this process each elevation is analysed using *Archimage* and *Benoit* programs producing, respectively, a $D_{(Archi)}$ and a $D_{(Benoit)}$ outcome. Then, the $D_{(Archi)}$ and $D_{(Benoit)}$ results for each elevation are averaged together to produce a separate $D_{(\mathrm{Elev})}$ result which moderates the different software packages respective tendencies to under-count or over-count detail.

For a typical house elevation, this $D_{(\mathrm{Elev})}$ result represents a summary of more than 26 separate mathematical comparisons each of around 7,000 data points, and using two variations on the box-counting approach (*Archimage* and *Benoit*) to produce an optimal result {1.3 & 2.1}. Next, the $D_{(\mathrm{Elev})}$ results for each of the four elevations of a building are averaged together producing a $D_{(\mathrm{Comp\ Elev})}$ result; being the average, two dimensional visual complexity of a complete dwelling as represented by its elevations {2.1}. This $D_{(\mathrm{Comp\ Elev})}$ result is typically the culmination of over 100 calculations and 28,000 automated data detection decisions.

The same procedure is then repeated for the plans of the building, initially producing a $D_{(\mathrm{Plan})}$ result for each level that is then combined into a $D_{(\mathrm{Comp\ Plan})}$ outcome. Finally, the $D_{(\mathrm{Comp\ Plan})}$ and $D_{(\mathrm{Comp\ Elev})}$ results may be averaged to create a single $D_{(\mathrm{Comp\ Plan\ +\ Elev})}$ result which is the best reflection of the two dimensional formal and spatial characteristic visual complexity of a building (see Figure 4). Each of these three forms of output, $D_{(\mathrm{Comp\ Plan})}$, $D_{(\mathrm{Comp\ Elev})}$ and

$D_{(\text{Comp Plan + Elev})}$ may be reported at this stage if required {2.2}; alternatively, this stage {2.0} can end with the production of the $D_{(\text{Elev})}$ results which are coded {3.0} in the final stage of the process.

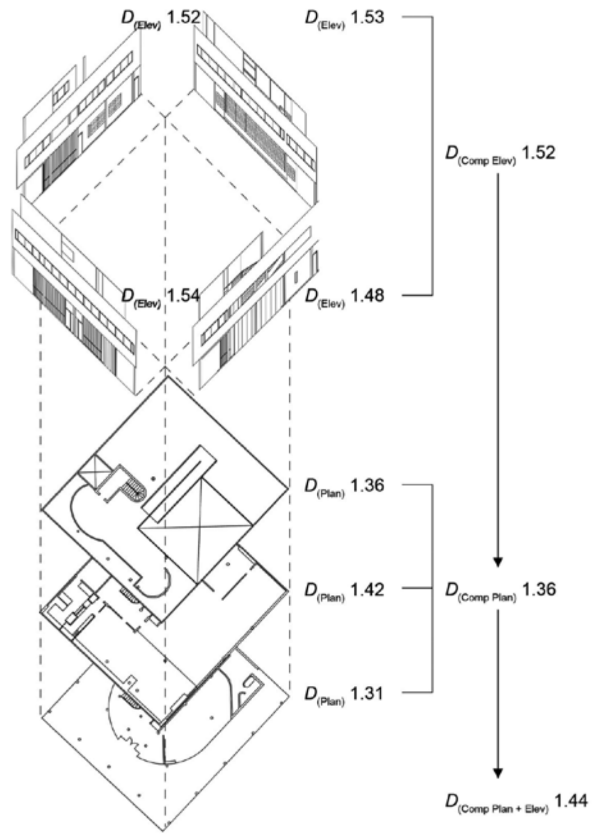

Figure 4. Diagrammatic representation of the "data compilation" stage {2.0}.

One of the practical challenges with applying the data developed from the computational fractal method is that it produces a single numerical result for each case being considered. This means that the data is in the form of isolated numbers on a linear scale and within a fixed range (2.00 > D > 1.00). This is akin to producing a chart which has a single "x axis" along which points are placed. The application of this type of data is essentially limited to comparisons between, for example, buildings that have higher levels of complexity and those that have lower levels. Or alternatively, this data can be used for comparisons between different architects' works or different periods in an architect's body of work. While such comparisons have been used to confirm a range of intuitive readings of architectural form, the application of fractal analysis is limited without its augmentation with an additional dimension; in effect, producing a "y axis" to compliment the current, isolated, "x axis". This is why the manual coding {3.0} of fractal data is so important.

The four current reports {3.3} that can be generated from the coding stage {3.0} are concerned with, respectively, orientation on site, approach to the building, permeability (the number of windows in each façade) and program (a variation of permeability which requires the programmatic function associated with an opening to be charted against the number of windows). In each case, the process uses additional information in the form of site plans, annotated floor plans and the expertise to read and interpret them {3.1}. In this way each $D_{(\mathrm{Elev})}$ result may be coded to reflect up to 8 different qualities of orientation and approach, and a wide range of permeability and program results {3.2} (see Table 2). The typical outcome of this final stage is a set of graphs of charts comparing the relative visual complexity of a façade and its architect's design strategies for sitting, address and program {3.3} (see Figure 5).

Table 2. Example coding of $D_{(\mathrm{Elev})}$ data for the Villa Savoye against three criteria

House: Villa Savoye	D(Elev)	Orientation	Approach	Permeability
Elevation 1	1.512	N	F	27
Elevation 2	1.4855	E	L	21
Elevation 3	1.5175	S	B	17
Elevation 4	1.450	W	R	12

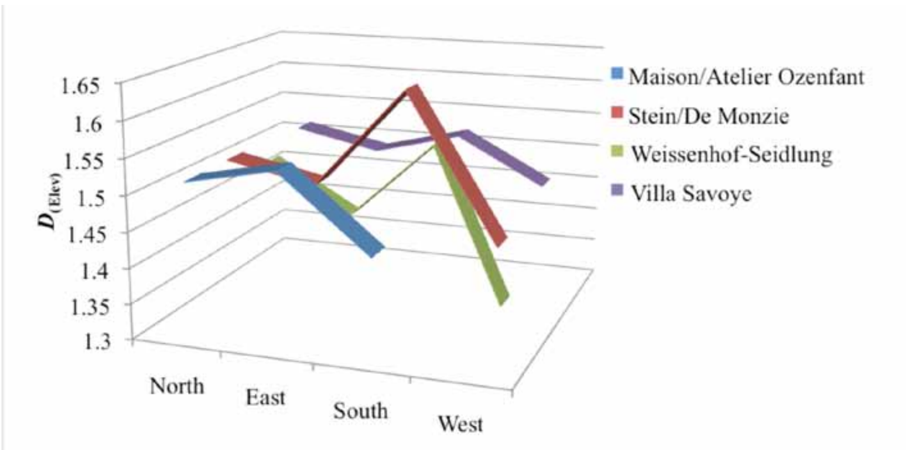

Figure 5. Example $D_{(Elev)}$ results for the Villa Savoye charted by orientation against the same results for three other houses by Le Corbusier.

4. Conclusions

The scientist Mandelbrot (1983) famously developed his system of non-Euclidean geometry in response to an observation that natural systems had a degree of complexity that defied conventional analytical techniques. However, with the rise of the computer, complex processes could suddenly be modelled for the first time. As a result of this, the architectural community now has access to a technique for analyzing the full complexity of architectural form and an appropriate set of computational tools for dealing with the large amounts of data they produce. Nevertheless, while computers may enable this process to occur, it is critical for researchers, scholars and designers to understand how this method works. This is why the present paper sets out to both trace and explain the processes that make up this important analytical method and the different forms of output it is capable of producing.

In the case of the example used throughout this paper, the Villa Savoye, the computational fractal analysis technique commenced with seven initial pieces of information (three plans and four elevations) from which it identified almost 42,000 relevant data points which were processed through around 182 iterative mathematical comparisons. This data was then synthesized to produce 10 results; 4 x $D_{(\mathrm{Elev})}$, 3 x $D_{(\mathrm{Plan})}$ results, 1 x $D_{(\mathrm{Comp\ Elev})}$, 1 x $D_{(\mathrm{Comp\ Plan})}$ and 1 $D_{(\mathrm{Comp\ Elev + Plan})}$. The 4 x $D_{(\mathrm{Elev})}$ results were then coded against 8 closed variables (that is, variables with a fixed range) and two open variables leading to more than 40 critical comparison points for the complete house. While the Villa Savoye is a relatively small house, the present authors have used the computational technique to begin to analyze a major historic building and in doing so they produced more than 2 million data points from over 1000 mathematical operations; the scale of the information processing required exceeded the capacity of the average desktop computer and required extensive dedicated processing time to complete. As the method is increasingly applied to more complex cases it becomes even more important that the people using or considering using this method are aware of its methodological and conceptual strengths and limitations. This paper provides the background for this knowledge, and further research will provide a critical assessment of the fractal analysis of architecture.

Acknowledgements

An ARC Fellowship (FT0991309) and two ARC Discovery Grants (DP0770106, DP1094154) supported the original authoring and testing process along with the most recent analysis undertaken in the present paper.

References

Bechhoefer, W. and Appleby, M.:1997, Fractals, music and vernacular architecture: an experiment in contextual design, *in* N. AlSayyad (ed.), *Critical methodologies in the study of traditional environments*, *Vol. 97*, University of California at Berkeley, Berkeley, unpaged.

Bovill, C.: 1996, *Fractal geometry in architecture and design*, Birkhäuser, Boston.

Bovill, C.: 1997, Fractal calculations in vernacular design, *in* N. AlSayyad (ed), *Critical methodologies in the study of traditional environments*, **97**, University of California at Berkeley, Berkeley, unpaged.

Burkle-Elizondo, G.; Sala, N. and Valdez-Cepeda R. D.: 2004, Geometric and complex analyses of Maya architecture: some examples, *in* K. Williams and F. Delgado-Cepeda (eds.), *Nexus V: architecture and mathematics*, K. W. Books, Florence, 57–68.

Foroutan-Pour, K.; Dutilleul, P. and Smith, D. L.: 1999, Advances in the implementation of the box-counting method of fractal dimension estimation, *Applied mathematics and computation,* **105**(2), 195–210.

Krampen, M.:1979, *Meaning in the urban environment*, Pion, London.

Lorenz, W.: 2003, *Fractals and fractal architecture,* Vienna University of Technology, Vienna.

Makhzoumi, J. and Pungetti, G.: 1999, *Ecological landscape design and planning: the Mediterranean context*, E & FN Spon, London.

Mandelbrot, B.: 1982, *The fractal geometry of nature*, WH Freeman and Company, San Francisco.

Ostwald, M. J.; Vaughan, J. and Tucker, C.: 2008, Characteristic visual complexity: fractal dimensions in the architecture of Frank Lloyd Wright and Le Corbusier, *in* K. Williams (ed.), *Nexus: architecture and mathematics*, K. W. Books and Birkhauser, Turin, 217-232.

Ostwald, M. J., and Vaughan, J.: 2009a, Visual qualities in early Modern and late Modern architecture: a mathematical comparison of formal complexity in the houses of Gray and Sejima, *ANZASCA 2009,* Newcastle, 9–32.

Ostwald, M. J., and Vaughan, J.: 2009b, Calculating visual complexity in Peter Eisenman's architecture: A computational fractal analysis of five houses (1968-1976), *CAADRIA 2009*, Taiwan, 75–84.

Vaughan, J. and Ostwald, M. J.: 2009, A computational comparison between the pre-modern (1905-1912) and early modern (1922-1928) architecture of Le Corbusier, *Design Principles and Practices*, **3**(4), 359–371.

Stamps III, A.: 1999, Architectural detail, Van der Laan septaves and pixel counts. *Design Studies,* **20**, 83–97.