From Idea to Shape, From Algorithm to Design:
A Framework for the Generation of Contemporary Facades

Inês Caetano[1], Luís Santos[2], António Leitão[1]

[1]INESC-ID/Instituto Superior Técnico, Universidade de Lisboa
[2]UC Berkeley – College of Environmental Design
ines.caetano@ist.utl.pt
luis_sds82@berkeley.edu
antonio.menezes.leitao@ist.utl.pt

Abstract. Nowadays, there is a growing interest in buildings' envelops presenting complex geometries and patterns. This interest is related with the use of new design tools, such as Generative Design, which promotes a greater design exploration. In this paper we discuss and illustrate a structured and systematic computational framework for the generation of facade designs. This framework includes (1) a classification of facades into different categories that we consider computationally relevant, and (2) an identification and implementation of a set of algorithms and strategies that address the needs of the different designs.

Keywords: generative design, facades, algorithms.

1 Contemporary Facades

The history of Architecture provides many examples of styles that were adopted, rejected, and then re-adopted in a similar or changed form. Before Modernism, buildings' facades were the canvas where architectural style was celebrated. On this canvas architects imprinted their personal interpretation of the current cultural stylistic models, with their metrics and canons. However, with the birth of Modernism, and its hygienic and austere aesthetic, composing a facade was an architectural task that lost some of its prestige. After Modernism (or since Post-modernism), we witness an increasing interest in facade composition and, nowadays, designing a facade is reassuming an important role in architecture practice due, in part, to the support of digital technologies [1].

This trend of highly textured building envelopes celebrates again the ornament in architecture and the composition of architectural facades. There are historical and cultural reasons for this renewed interest, such as the reinterpretation of Modernist aesthetics, the reintroduction of symbolism and historical precedent by Post-Modernism [2], and the diligent look and revisit of vernacular precedent proposed by Critical Regionalism [3]. However, there is also a technological reason: algorithmic approaches made it easier to conceive, deploy and adapt the design of architectural surfaces with complex and intricate textures, and differentiated levels of porosity. As

adfa, p. 1, 2011.

an example, consider the highly sophisticated Erwin Hauer "Continua series" sunscreens patterns (Fig. 1), which nowadays can be treated as a programming exercise in some parametric design courses.

Unfortunately, new facade designs still require a lot of effort to invent, experiment, and produce. It is important, then, that this effort be as small as possible. The work presented here proposes a systematic methodology for the development and composition of algorithmically-based facade patterns. As we will show, our methodology promotes the design exploration of facades and simplifies its adaptation to the ever-changing design process conditions.

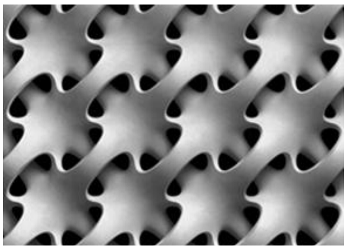


Fig. 1. Continua Screen, design 1 - pattern developed by Erwin Hauer in the 1950's [4].

2 Algorithmic-based Processes in Architecture

Creativity is characterized by unconsciousness and inaccuracy [5] and, thus, is better served by a design process that embraces change. Traditional tools do not easily support change because they require too much time and effort to modify models. On the other hand, the computer became a very important tool for the design process, which changed, and is still changing, the way architects design [6]. Recent technologies allow design exploration to go far beyond the traditional possibilities, thus promoting the development and proliferation of complex shapes, new patterns and advanced production techniques. Computers "*do not eradicate human imagination but rather extend its potential limitations...it provides the means for exploration, experimentation, and investigation in an alternative realm*" [7].

Generative design (GD) is an approach to design which creates shapes through algorithms [7]. This approach enables the generation of various solutions in a short period of time, avoiding the tedious and repetitive tasks needed when the modeling work is done manually, even with state-of-the-art CAD/BIM (Computer Aided Design/Building Information Modeling) software. In addition, GD enables and facilitates the manufacturing of complex solutions, by extracting information directly from the model to CAM (Computer Aided Manufacturing) or Digital Fabrication. Through GD, instead of going directly from the idea to the design, architects produce an intermediate algorithmic-based description of a design [8]. Parametric Design is a GD approach in which the parameters of a particular design are declared, rather than its shape [6]. This approach has the ability to generate different instances of a design where each instance represents a unique set of transformations based on the actual parameters [9], allowing the designer to freely explore a larger solution space of the design briefing/program. Ultimately, this leads to the assessment of solutions that would be difficult to generate with traditional design methods. An algorithmic-based design method can easily accommodate changes in the proposed solutions, as the dynamics of the design process alter the state of the design brief and its programmatic nature.

3 A Framework for the Generation of Contemporary Facades

In this paper, we discuss the development of a computational framework for the design of facades. Our work started with an analysis of a large corpus of contemporary facades. The wide variety of contemporary facades has already promoted several different classifications, which were based on different concepts, such as Depth and Affect [10] and facade's Articulation [1]. However, as our framework aims to help the designers with the *generation* of facades, we propose a different classification which is more helpful to the designer that intends to use a computational approach.

In our work, facades are classified into different categorical dimensions that we consider computationally relevant. This multidimensional classification guides the designer towards a library of functional operators, each addressing the generation of different designs of facades. By using this library, designers match their ideas with the categorical dimensions, which guides them in the selection of the most appropriate computational approaches for the generation of the idealized facades.

Given that facades are, in many cases, a composition of different designs, it is not reasonable to expect that this matching process yields a complete computational solution. Instead, the designer is responsible for the division of the whole design into parts, for establishing the dependencies between them, for instantiating and combining the different computational functions that handle each part, and for the additional scripting that might be needed to handle specific circumstances of the design brief.

4 Design Stages

There are several stages in the design of facades and the presented framework takes them into account. These stages are in accordance with the computational logic of the facade design and each one corresponds to one or more dimensions of our classification. The stages are:

1. The definition of the facade's geometry.
2. The generation of the facade's elements, which includes the definition of their geometry, type of deformation and size variation.
3. The distribution of the elements, which is responsible for mapping and rotating the elements on the facade.
4. The generation of the facade's final appearance, which produces the type of facade's finish and selects the material or color to apply.

In the next sections we discuss each categorical dimension and its role in the different steps of the facade generation.

5 Categorical Dimensions

The framework is organized in eight categorical dimensions: (1) Facade's Geometry, (2) Element's Geometry, (3) Element's Distortion, (4) Element's Size, (5) Element's Distribution, (6) Element's Rotation, (7) Material & Color, and (8) Facade Articulation, where each dimension corresponds to a set of related computational functions. This classification generates a multi-dimensional space where parts of a facade can be located. The important result of our work comes, then, from the identification and implementation of a set of algorithms and strategies that address the needs of the different dimensions of this space. Some of the locations in this multi-dimensional space can use a specific computational approach that is adequate for the creation of the designs that match the intended facade. Other locations, representing less common kinds of facades, might not have a specific computational solution, but our experience shows that is possible, using the range of tools that we developed, to quickly implement the particular solution required by that facade.

Table 1. Classification Synthesis: the dimensions are at the top of each box and below there are the corresponding algorithms**.**

Dimension	Algorithms
FACADE'S GEOMETRY	STRAIGHT CYLINDRICAL SPHERICAL UNDULATE S-ELLIPSE TORUS FREE-FORM
ELEMENT'S ROTATION	HORIZONTAL VERTIVAL PICTORIAL RANDOM
ELEMENT'S GEOMETRY	CIRCULAR CYLINDRICAL SPHERICAL OVAL TRIANGULAR PYRAMIDAL SQUARED RECTANGULAR CUBOID HEXAGONAL PENTAGONAL STRIPES PICTORIAL
ELEMENT'S DISTORTION	TWISTED UNDULATED INTERLACED BENDED
ELEMENT'S SIZE	FIXED INCREASING ATTRACTED PICTORIAL RANDOM
ELEMENT'S DISTRIBUTION	1D: IN COLUMNS IN ROWS ALTERNATED: COLUMNS OR ROWS 2D: REGULAR GRID CHESS GRID ALTERNATED GRID RECURSIVE GRID PICTORIAL GRID 3D
MATERIAL & COLOR	WOOD METAL CONCRETE ONE COLOR MULTI-COLOR PICTORIAL RANDOM COLOR
FACADE ARTICULATION	PERFORATED APPLIED STACKED JUXTAPOSED LAYERED

Table 1. shows, synthetically, all the categorical dimensions. Each box describes one dimension and the corresponding algorithms. In the next sections we discuss each of these dimensions.

5.1 Facade's Geometry

Given that designers want their models to be flexible, when they define the underlying principle of a geometry, they want to control and change it easily, so that many design instances can be generated within the same geometrical principle. This idea guides our first dimension, named Facade's Geometry. For each different geometry, our framework provides a $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ parametric function that describes the shape of the facade. For example, $f(u, v) = XYZ(u \times 5, 0, v \times 10)$, where XYZ is the Cartesian coordinate function, represents a five-by-ten rectangle on the XZ plane. Naturally, other coordinate systems can be used, such as the Cylindrical, represented by function CYL, and the Spherical, represented by the function SPH, to which can be applied transformations, such as translation, rotation, etc. The coordinate system transformations are related to a spatial location of reference, capable of codifying the transformed referential, which, for brevity, we will omit. To simplify the presentation, each parametric function $S(u, v)$ will range over the domain $0 \leq u \leq 1, 0 \leq v \leq 1$.

To make the framework more flexible, we also rely on the use of anonymous functions, i.e., functions which do not have a name, and higher-order functions (HOFs), i.e., functions that receive other functions as arguments and/or compute other functions as results [11].

As an example, consider the facade of the Formstelle Office Building (Fig. 2), which is completely planar. This is classified in the Facade's Geometry dimension as Straight, which, depending on a width *w* and height *h* of the facade, is defined by (1):

$$Straight(w, h) = \lambda(u, v).XYZ(u \times w, 0, v \times h) \quad (1)$$

Note that Straight is a HOF that returns an anonymous parametric function that represents a delimited region on the XZ plane. The λ symbol is the λ-calculus notation for an anonymous function [11].

Although it is perfectly possible to explicitly define functions such as Straight, our framework goes deeper than that by providing a set of more fundamental functional operators that can be arbitrarily combined. One such operator represents a one-dimensional linear variation: $linea(a,b)=\lambda(t).a+(b-a)t$. Another, represents a (paradoxical) constant "variation": $constan(c)=\lambda(t).c$.

Given that the facade geometry domain is two-dimensional, it is also useful to extend the domain of the above one-dimensional variations into $\mathbb{R}^2$. To this end, we define the HOFs $dim_u(f)=\lambda(u,v).f(u)$ and $dim_v(f) = \lambda(u, v).f(v)$. A final but important operator is the generalized composition of functions

$$\circ(f, g_1, \cdots, g_n) = \lambda(x_1, \cdots, x_m).f\big(g_1(x_1, \cdots, x_m), \cdots, g_n(x_1, \cdots, x_m)\big) \quad (2)$$

In order to allow a simplified notation, we define $a \otimes b = dim_a\big(linear(0, b)\big)$, we treat all numbers n that occur in a function context as $constant(n)$, and we treat any ordinary first-order function f that is used with functional arguments $g_1, \cdots, g_n$ as $\circ(f, g_1, \cdots, g_n)$. This means, e.g., that $sin \times cos$ is the same as $\circ(\times, sin, cos)$.

The use of HOFs allow us to move from the numeric space, where numbers are combined using numeric operators, into the function space, where functions are combined using functional operators. In this space, the Straight function presented above can be equivalently defined as

$$Straight(w, h) = XYZ(u \otimes w, 0, v \otimes h) \quad (3)$$

For a different example, consider the Suzhou SND District Urban Planning Exhibition Hall (Fig. 3), which, for radius *r* and height *h*, is described by the following function (4):

$$Cylindrical(r,h) = CYL(\,r, u \otimes 2\pi, v \otimes h) \quad (4)$$

Fig. 2. Facade Geometry: Straight Facade - Formestelle Office Building, in Töging am Inn, Germany.

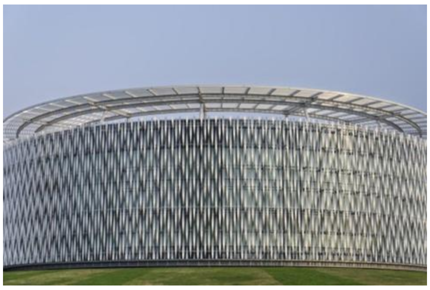


Fig. 3. Facade Geometry: Cylindrical Facade - Suzhou SND District Urban Planning Exhibition Hall, in Jiangsu, China.

Finally, consider the sinusoidal facades which are very common in recent architecture (see Figs 4 and 5). The sinusoidal HOF is: $sinusoid(a,\omega,\phi) = \lambda(x).\, a \times \sin(2\pi\omega x + \phi)$, where a is the amplitude of the sinusoid, ω is the angular frequency, i.e. the number of cycles per unit length, and ϕ is the phase. However, there are more than one type of sinusoidal surfaces. Some, such as the one in Fig. 4, have the undulation in the XY plane, thus producing a horizontal wave. This type of surface is defined by function (5):

$$Sb(w,h,a,\omega,\phi) = XYZ(u \otimes w, u \otimes sinusoid(a,\omega,\phi), v \otimes h) \quad (5)$$

Others, such as the Boiler house at the Guy's Hospital, in London (Fig.5) have the undulation along two axes and are defined by (6):

$$Se(w,h,a,\omega,\phi) = XYZ(u \otimes w, u \otimes sinusoid(a,\omega,\phi) \times v \otimes sinusoid(a,\omega,\phi + \pi/2), v \otimes h) \quad (6)$$

On the other hand, there are facades with completely irregular shapes, like the Selfridges Building in Birmingham (Fig. 6), which are classified in the Facade's Geometry dimension as Free-Form. In this last case, the designer creates the shape manually, and imports it into our framework where it is represented as another parametric function that results from an interpolation process.

Fig. 4. Facade Geometry: Facade with horizontal waving - Apartment house in Tokyo.

Fig. 5. Facade Geometry: Sinusoidal and co-sinusoidal Facade - Boiler House at Guy's Hospital, in London, UK.

We will now look into the Elements' Geometry, the next relevant dimension of our classification.

Fig. 6. Facade Geometry: Free Form Facade - Selfridges Building in Birmingham, UK.

5.2 Elements' Geometry

There are several examples of facades where a particular kind of element is repeated, as is visible in Fig. 7. As we saw in the previous section, the facade's geometry defines surface on which the elements will be placed, but before considering the placement of the elements, we need to describe the algorithms that shape them. This dimension is called Element's Geometry and, as it happens with the Facade's Geometry, it provides several pre-defined functions representing geometric shapes. In many cases, these elements can be described by the same functions that describe the facade geometry but, here, the geometry is more standardized, allowing us to provide pre-defined functions such as circle, triangle, square, hexagon, etc.

Contemporary facades with round elements are also very common and can be classified as cylindrical, spherical, circular, etc. The New Center for Manufacturing Innovation (Fig. 7), is an example of a facade with circular elements. Another example is the facade of the Hanjie Wanda Square (see Fig. 8), which is covered by several metallic spheres.

Facades with quadrangular elements can be classified as squared, rectangular, cuboid, etc. Elements shaped as regular-polygons are also common in contemporary facades and can be classified as hexagonal, pentagonal, hexagonal-prism, etc. An example of a facade with hexagonal elements is The Cube in Milano (see Fig. 9). Facades with striped elements are classified as stripes, producing continuous elements along the whole facade (Fig. 10).

Fig. 7. Element's Geometry: Circular Elements - New Center for Manufacturing Innovation, in Monterrey, Mexico.

Fig. 8. Element's Geometry: Spherical elements - *Hanjie Wanda Square*, in China.

Fig. 9. Element's Geometry: Hexagonal Elements - The Cube, in Milan, Italy.

Fig. 10. Element's Geometry: Stripes Elements - Aspen Art Museum, in Aspen, USA.

Besides the regular geometries, this dimension provides algorithms for more specific geometries, which are classified as Pictorial. These algorithms generate elements whose shape is determined by an image. An example of a facade with pictorial elements is Mayfair House, in London (Fig. 11).

After selecting the facade's geometry and the element's geometry, it is time to combine them to define the complete facade. One of the advantages of the functional representation is that it makes this combination a trivial composition of functions. As an example, a facade such as the one in Fig. 7 (circular elements on a straight facade), is defined by $\circ(circle, Straight(w, h), constant(r))$ or, in simplified notation, $circle(Straight(w, h), r)$).

Fig. 11. Element's Geometry: Pictorial Elements - Mayfair House, in London, UK.

It is important to note that the previous function is a continuous function that generates an infinity of circles in a delimited rectangle on the XZ plane. This means that we are not yet representing the actual distribution of circles, a topic that will be described in a later section.

5.3 Elements' Distortion

Distortion is a type of transformation where the natural form of the element is changed, for example, by twisting it along some dimension. To this end, it is possible to use pre-defined operations, such as Sweeping, which displaces the element's section along a curve, possibly rotating it and/or scaling it.

Twisted elements, such as those visible in the Huaxin Business Center (Fig. 12), result from a helical movement around their own axis. These particular elements are described by the following function, where c is a curve in the facade's surface, w and e are the dimensions of the cross-section of the element and, finally, ϕ is the twisting angle:

$$\lambda(c).sweep(c, rectangle(w, e), \phi) \tag{7}$$

Another type of distortion is named Undulated, which is a particular case of sweeping, where the guiding curve is sinusoidal (Fig. 13).

Interlaced elements use the same method described above but the elements are strategically placed so that they are weaved (Fig. 14). This is achieved by alternating the value of the sinusoid's phase, in both vertical and horizontal elements.

Finally, Bended is a deformation similar to a Zig-Zag, which flexes the elements according to the angles defined by the user (Fig. 15).

Fig. 12. Element's Distortion: Twisted Elements: Huaxin Business Center, in Xuhui, China.

Fig. 13. Element's Distortion: Undulated Elements - Visitor Pavilion National Museum Palace in Het Loo, Apeldoorn, Netherlands.

Fig. 14. Element's Distortion: Interlaced Elements - Argul Weave Building, in Bursa, Turkey.

Fig. 15. Element's Distortion: Bended Elements - Pan American Health Organization Building, Washington DC , USA.

5.4 Elements' Size

As is visible in Fig. 16, it is common to find modern facades where the element has a size that varies along the facade. For this dimension-Element's Size-we have predefined a set of functions capable of producing the most common types of variations. For example, if we assume that we have a straight facade where the element is a circle whose radius changes linearly from r_0 to r_1 along the v dimension, the corresponding function is given by

$$circle(Straight(w,h), dim_v(\, linear(r_0, r_1))) \quad (8)$$

In another frequent example the size varies according to the distance to a point or a curve (Fig.17). These cases are classified as Attracted and we provide a set of corresponding functions for their computation.

Fig. 16. Element's Size: Increasing - The Tourist Office and Landscaping of Quinta do Aido, Portugal.

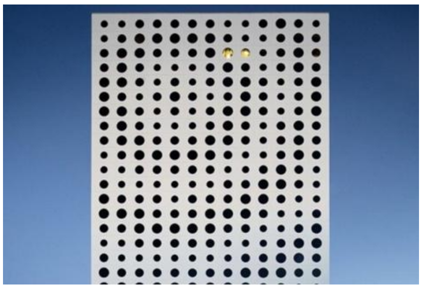


Fig. 17. Element's Size: Attracted - Quality Hotel Friends, in Sweden.

Elements with a size variation that reproduces an image are classified as Pictorial (Fig.18). For these cases, an image is provided as input, where each pixel color value controls the size of the element.

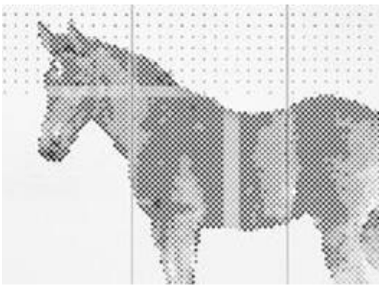


Fig. 18. Element's Size: Pictorial - Hästsportens Hus, in Sweden.

Fig. 19. Element's Size: Random - Cascais House, in Portugal.

Finally, elements with a random size variation (Fig. 19), are classified as Random because the element function has a size parameter controlled by a random function. As an example, we can modify the linear size variation illustrated above to use randomness instead:

$$circle(Straight(w,h), \lambda(u,v).random(r_0,r_1)) \quad (9)$$

5.5 Elements' Distribution

So far, we have described functions and functional operators that allow the construction of a functional description of the facade that includes its geometry, the element's shape, its transformation and its size variation. As previously mentioned, this description is a continuous function. However, most facades are discretized, in the sense that the function is not evaluated in its entire domain but, instead, in a sampling of its domain. It is this sampling that characterizes the Element's distribution, that is, the placement of the elements along the facade. In other words, the functional description of the facade is mapped, not on a continuous domain, but on a discrete domain. This is accomplished by a discretization function that, given a continuous domain and the intended number of samples, computes a set containing the discretization of the domain. Depending on the dimensionality of the domain, we might also need to compute the Cartesian product of the discretization of the independent domains.

The use of discretization functions allows different patterns of mapping. For example, in the Huaxin Business Center (Fig. 12) the elements are distributed in alternated columns with opposite rotations. Another example is the Quality Hotel Friends (Fig.17) where elements are disposed in a regular grid, i.e. aligned horizontally and vertically. A third example occurs when there is an overlapping of two grids, as it happens with the Hanjie Wanda Square (Fig. 8). We classify this type of distribution as Alternated Grid

A Recursive Grid is an interesting example of a composed distribution: it occurs when a surface is divided into a regular grid which is further randomly subdivided into sub-grids (Fig. 20). Another composed distribution, classified as Pictorial Grid, maps the elements according to the design or image provided by the designer, i.e. the elements' distribution outlines the geometry in the picture (Fig. 21). The last composed two-dimensional distribution, pre-defined within this sub-group, is classified as Random-Grid, precisely because the element's distributions are based on randomness (see Fig. 19).

Fig. 20. Element's Distribution: Recursive Grid- The Cube, in Birmingham, UK.

Fig. 21. Element's Distribution: Pictorial Grid - Podcetrtek Sports Hall, in Podcetrtek, Slovenia.

In the last sub-group, 3D distribution, the mapping of the elements is done along three dimensions. Two of them belong to the surface geometry and the third one represents an additional spatial or temporal dimension. In this last case, the placement of the elements varies with the course of time (Fig. 23).

The functionality here described is summarized in Fig. 22 for the one-dimensional and two dimensional discretization cases.

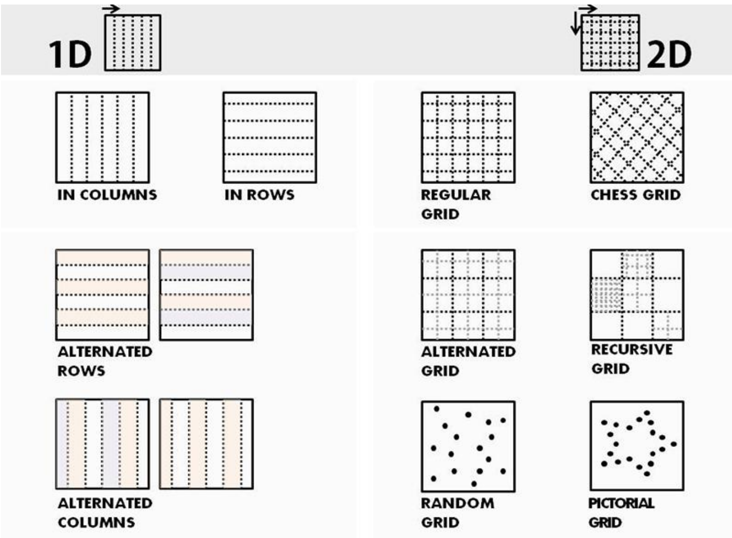


Fig. 22. Explanatory scheme of the available types of distributions. The set on the left represents one-dimensional distributions (1D), and the set on the right the two-dimensional distributions (2D).

5.6 Element's Rotation

In some facades, such as the one in Fig. 12, elements can be distinguished according to its rotation. This categorical dimension, Element's Rotation, is responsible for defining the rotation angle to be applied to each element. The most common types of rotation are pre-defined in our framework and include elements horizontally rotated and elements vertically rotated. Facades whose element's rotation produces a general image or pattern are classified as Pictorial-Rotation. The mechanism is similar to the dimension Pictorial-Size, but uses rotation angles instead. The last type of rotation is classified as Random-Rotation, and it applies a random rotation angle to each element.

Fig. 23. Element's Distribution: 3D distribution - MegaFaces Pavilion Sochi 2014 Winter Olympics, in Russia.

Fig. 24. Element's Rotation: Pictorial Rotation - Winery Gantenbein, in Switzerland.

5.7 Facade's Articulation

Articulation is a method or manner of jointing that makes the united parts clear, distinct, and precise in relation to each other [12]. The relation between the facade's parts can be done in different ways, thus providing facades designs with different appearances. Thus, Facade's Articulation directly addresses the tectonics of the facade composition, i.e. the way different elements connect and relate to assemble a specific architectural effect.

In Perforated facades, the elements are subtracted from the whole surface, thus requiring a Boolean subtraction. Here, the elements locate and shape the holes that constitute the facade. In Applied facades, the elements are united to the façade's surface. Facades of Stacked elements and facades with elements Juxtaposed also have a relation of union between the parts but, in the first case, the elements have to be distributed so as to be placed right next to each other, while in the second case the elements are not applied on the surface, because they themselves establish the building's skin, by their juxtaposition. Lastly, facades consisting of an overlapping of two or more layers are classified as Layered. The layers may have characteristics of the previous articulations, which are then overlapped and unified, constituting a unique facade.

5.8 Material & Color

The last categorical dimension is in charge of giving the materiality to the facade's model. If the facade has the material in sight, the layer of the facade will have the name of the chosen material and will also present the chosen materiality. If the facade has colors in its final appearance, the process is the same but with the color name. For more specific uses of colors, if there is an apparent use of random colors, the layer name will have a certain randomness that, consequentially, produce random colors - Random Color. If there is an apparent use of color to produce an overall image or pattern, the name of the layer will be submitted to the same process as the coordinates already explained, pictorial-size and pictorial-rotations, but this time receiving tones - Pictorial Color.

6 Practical Application

In practical terms, the end result of our research is a library of functional primitives and functional operators usable in different programming languages and a set of guidelines that helps a designer select and combine the most useful operators to implement a design for a particular facade.

As an example, consider a facade with Straight geometry and with Juxtaposed elements. The elements have a Pictorial geometry and a linearly increasing size variation. The distribution of the elements is in a Regular Grid, and the color of the facade is classified as White. For each of these classifications, we can select the appropriate function, which we will combine using the functional operators. The end result is visible in Fig. 25. The chosen classification is highlighted in the following table:

FACADE'S GEOMETRY	ELEMENT'S GEOMETRY	ELEMENT'S SIZE	ELEMENT'S DISTRIBUTION 2D	FACADE ARTICULATION	MATERIAL COLOR
STRAIGHT	CIRCULAR	FIXED	**REGULAR GRID**	PERFORATED	METAL
CYLINDRIC.	SQUARED	**INCREASING**	CHESS GRID	APPLIED	CONCRETE
SPHERICAL	HEXAGONAL	ATTRACTED	ALTERNATE GRID	PRINTED	…
UNDULATED	TRIANGULAR	PICTORIC	RECURSIVE GRID	**JUXTAPOSED**	**WHITE**
FREE-FORM	**PICTORIAL**	RANDOM	PICTORIC GRID	LAYERED	BLACK

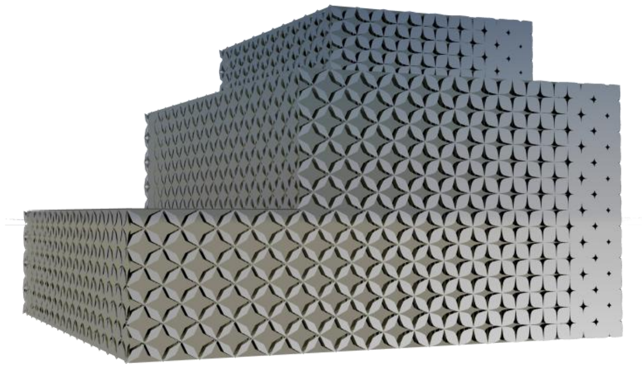


Fig. 25. An example of a facade generated through the framework operations: Straight facade; pictorial elements with increasing sizes; regular-grid distribution; Color white and juxtaposed surface.

Now, imagine that we replace the size variation, from linearly increasing to randomized. This change produces the facade visible in Fig. 26.

FACADE'S GEOMETRY	ELEMENT'S GEOMETRY	ELEMENT'S SIZE	ELEMENT'S DISTRIBUTION 2D	FACADE ARTICULATION	MATERIAL COLOR
STRAIGHT	CIRCULAR	FIXED	**REGULAR GRID**	PERFORATED	METAL
CYLINDRIC.	SQUARED	INCREASING	CHESS GRID	APPLIED	CONCRETE
SPHERICAL	HEXAGONAL	ATTRACTED	ALTERNATE GRID	PRINTED	…
UNDULATED	TRIANGULAR	PICTORIC	RECURSIVE GRID	**JUXTAPOSED**	**WHITE**
FREE-FORM	**PICTORIAL**	**RANDOM**	PICTORIC GRID	LAYERED	BLACK

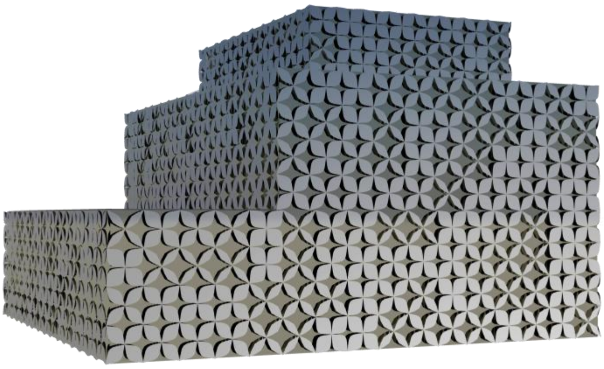


Fig. 26. An example of a facade generated through the framework operations: Straight facade; pictorial elements with random sizes; regular-grid distribution; Color white and juxtaposed surface.

Now, if we change the type of element's distribution to become a Recursive Grid, and keep the type of size variation classified as linearly increasing, we generate the facade in Fig. 27.

FACADE'S GEOMETRY	ELEMENT'S GEOMETRY	ELEMENT'S SIZE	ELEMENT'S DISTRIBUTION 2D	FACADE ARTICULATION	MATERIAL COLOR
STRAIGHT	CIRCULAR	FIXED	REGULAR GRID	PERFORATED	METAL
CYLINDRIC.	SQUARED	**INCREASING**	CHESS GRID	APPLIED	CONCRETE
SPHERICAL	HEXAGONAL	ATTRACTED	ALTERNATE GRID	PRINTED	…
UNDULATED	TRIANGULAR	PICTORIC	**RECURSIVE GRID**	**JUXTAPOSED**	**WHITE**
FREE-FORM	**PICTORIAL**	RANDOM	PICTORIC GRID	LAYERED	BLACK

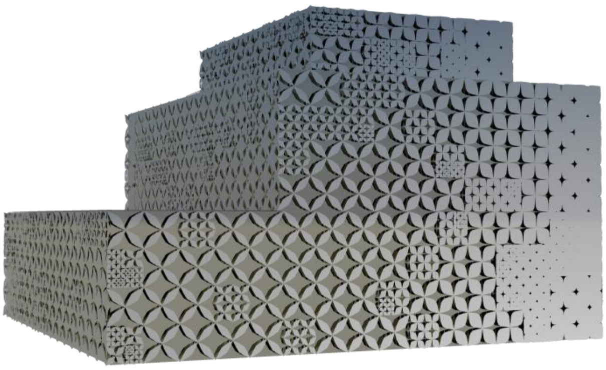


Fig. 27. An example of a facade generated through the framework operations: Straight facade; pictorial elements with increasing sizes; recursive-grid distribution; Color white and juxtaposed surface.

Lastly, imagine that this facade has now the distribution of the elements in a Chess Grid, but keeps the geometry and the size variation of the elements as in the previous example. The facade articulation is Juxtaposed but also Layered, where the first layer is classified by the color White and the second layer by the color Black. After exploring this pattern, it is applied on a surface with undulated geometry. The generated facade is represented by Fig. 28.

FACADE'S GEOMETRY	ELEMENT'S GEOMETRY	ELEMENT'S SIZE	ELEMENT'S DISTRIBUTION 2D	FACADE ARTICULATION	MATERIAL COLOR
STRAIGHT	CIRCULAR	FIXED	REGULAR GRID	PERFORATED	METAL
CYLINDRIC.	SQUARED	**INCREASING**	**CHESS GRID**	APPLIED	CONCRETE
SPHERICAL	HEXAGONAL	ATTRACTED	ALTERNATE GRID	PRINTED	…
UNDULATED	TRIANGULAR	PICTORIC	RECURSIVE GRID	**JUXTAPOSED**	**WHITE**
FREE-FORM	**PICTORIAL**	RANDOM	PICTORIC GRID	**LAYERED**	**BLACK**

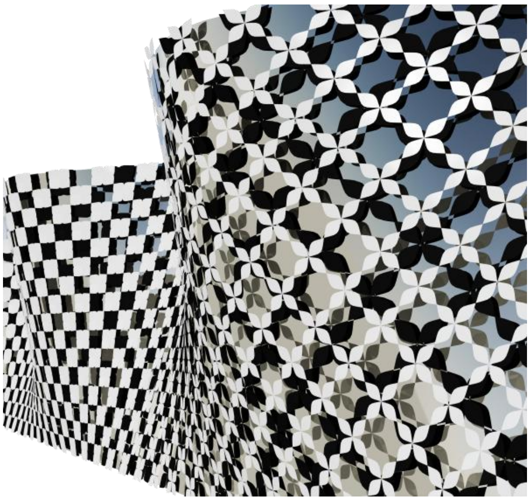
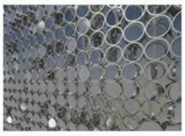
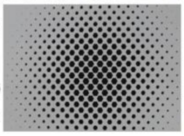
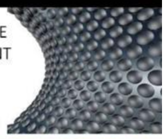
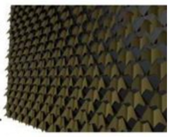
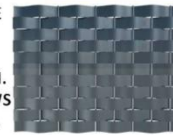


Fig. 28. An example of a facade generated through the framework operations: Layered facade with undulated geometry on top of a reflective surface. Each layer is composed by juxtaposed pictorial elements with increasing sizes and chess-grid distribution. The outer layer uses color white and the inner layer, simulating a shadow of the outer layer, uses color black.

7 Evaluation

In order to evaluate our framework we used it to reproduce some existing facades. Table 2 presents (on the left) a selection of projects which we classified (in the middle) according to our categorical dimensions and which we modeled (on the right) using the functional representation suggested by the classification.

Table 2. Table synthesis with some of the framework's applications

PROJECT	CLASSIFICATION	MODEL
	STRAIGHT FACADE PICTORIAL ELEM. FIXED SIZE REGULAR-GRID LAYERED ARTICU.	
	STRAIGHT FACADE CYLINDRICAL ELE. FIXED SIZE REGULAR-GRID JUXTAPOSED ART.	
	STRAIGHT FACADE HEXAGONAL ELE. ATTRACTED SIZE ALTERNATED-GRID PERFORATED ART.	
	FREE-FORM FACADE SPHERICAL ELEMENT FIXED SIZE ALTERNATED-GRID APPLIED ARTICULA.	
	STRAIGHT FACADE PICTORIAL ELEM. FIXED SIZE ALTERNATED-GRID APPLIED ARTICULA.	
	STRAIGHT FACADE STRIPE ELEMENT FIXED SIZE UNDULATED ELEM. ALTERNATED ROWS JUXTAPOSED ART.	

We can conclude, from the comparison between each project and the corresponding model generated using our framework, that we can achieve a high degree of fidelity. Equally important is the effort required to use our framework. Our empirical evaluation shows that the classification step requires between five and ten minutes, while the selection, composition, and testing of the functions suggested by the classification takes between fifteen minutes and one hour, depending on the complexity of the facade.

8 Related Work

There are already some tools that attempt to solve the problems here described, such as the Paneling Tools plug-in for Rhino and Grasshopper, the Lunch Box add-on to Grasshopper, and ParaCloud Gem, a stand-alone toolkit that adds generative capabilities to any CAD system that supports *.obj, *.stl, *.collada, and *.dxf file formats. All of them are capable of creating grids of points on a surface, mapping elements in different ways, applying attractors to control elements size, etc.

The nature of their limitations is three-folded: (1) its use is entirely manual, thus mainly promoting iterative user-driven processes, which can be tiresome and error-prone; (2) when using such toolkits in the context of an Application Programming Interface (API) or as plug-ins to a domain-specific programming language, such as Grasshopper, a certain level of automation is obtained, however, the designer is always bound to the specific functionalities provided by the tool, thus limiting its agency in exploring different combinations of operations and extending the capabilities of the tool's pre-defined operations; (3) these tools are more used for generic panelization, subdivision, and population of surfaces thus, although they have been used to generate complex facade patterns, they are not fully architectural oriented which means that they do not directly address relevant concepts in facade design such as materiality or the tectonic relation between the facade elements

We should also mention recent domain-specific programming languages, such as Dynamo for Revit and Grasshopper for Rhino, which allow users to implement the functionalities proposed in this paper. In addition, some of the pre-defined components have similar purpose to some of the HOFs which we presented. However, the freedom of connection allowed by these tools becomes difficult to manage in complex facades [13]. In these cases, a more structured and systematic approach like the one we propose is more manageable.

In summary, with the current framework the architect is limited by the non-domain specificity of existing tools or in order to extend their capabilities he needs to build from scratch the necessary functionalities or use a mix of different tools that most of the time are not compatible. This work extends the state-of-the-art by: (1) systematizing and structuring, in an architectural-oriented framework, the parametric generation of a wide range of facade typologies, and by (2) operationalizing it resorting to a simple algorithmic approach that uses and combines different functional operators that directly implement facade design concepts.

9 Conclusions

The exploration of architectural facades is not new. However, by resorting to recent digital technologies, architects can once again focus on facade design, promoting a growing interest in the exploration of complex patterns and geometries.

In this paper we presented a methodological framework that helps designers generate different facade designs, through the use of a set of functional operators. The current implementation of the framework was done using the Rosetta IDE [14], allowing its exploration in different programming languages.

In order to systematize the use of the framework, we proposed a classification of facades based on several categorical dimensions which we considered to be computationally relevant. These categorical dimensions guide the selection of the functional algorithms that handle each part of the facade. These might then be used directly, or might be combined using functional operators, promoting a systematic exploration of designs which ultimately aims to a higher productivity by: (1) improving the time of scripting tasks, and (2) adding flexibility to the designers' workflow. Due to the simplicity of the functional composition, this framework accommodates the ever-changing nature of a design process by facilitating the test of several design concepts, or instantiations of the same idea, in any design stage.

In the near future, we plan to expand the set of functional algorithms and operators, covering a wider range of facades. In order to make this framework more usable, we are particularly interested in conducting a wider field study of its application, to identify weaknesses of the proposed processes and opportunities for extensions.

10 Acknowledgments

This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013, and by the Rosetta project under contract PTDC/ATP-AQI/5224/2012 and by the PhD grant under contract SFRH/BD/98658/2013.

11 References

1. B. Pell, The Articulate Surface - Ornament and technology in Contemporary Architecture, Birkhauser, 2010.
2. R. Venturi, Complexity and Contradiction in Architecture, 1966.
3. K. Frampton, "Towards a Critical Regionalism: Six Points for an Architecture of Resistence," *The Anti-Aesthetic: Essays on Postmodern Culture*, 1983.
4. E. Hauer, Erwin Hauer: Continua, Architectural Screens and Walls, Princeton Architectural Press, 2004.
5. F. A. Bukhari, "A Hierarchical Evolutionary Algorithmic Design (HEAD) System for Generating and Evolving Buildings Design Models," Queensland University of Technology, 2011.
6. B. Kolarevic, " Architecture in the Digital Age - Designing and Manufacturing," London, UK, 2003.
7. K. Terzidis, "Introduction," em *Expressive Form - A conceptual approach to computational design*, Spoon Press, 2003, pp. 1 - 8.
8. A. Leitão, "Teaching Computer Science for Architecture," 2013.
9. C. Barrios, "Transformations on Parametric Design Models," 2005.
10. F. Moussavi e M. Kubo, The function of Ornament, Actar, 2006.
11. A. Leitão, "Improving Generative Design by Combining Abstract Geometry and Higher-Order Programming," em *CAADRIA - Conference on Computer-Aided Architectural Design Research in Asia*, Kyoto, 2014.
12. B. Borson, "Life of an Architect," 08 November 2010. [Online]. Available: http://www.lifeofanarchitect.com/words-that-architects-use/.
13. A. Leitão, L. Santos e J. Lopes, "Programming languages for generative design: A comparative study," *International Journal of Architectural Computing*, vol. 10, n.º 1, pp. 139-162, 2012.
14. J. Lopes e A. Leitão, "Portable Generative Design for CAD Applications," em *ACADIA*, Calgary, Canada, 2011.