Design Without Rigid Rules

Sotirios D. Kotsopoulos

Abstract Visual rules capture how a shape is perceived and what we choose to do with it. Shape decompositions reveal alternative sets of shape parts. In visual design, we can neither determine rigid modes of seeing nor doing or interpreting. Hence, rules and decompositions cannot be rigid. This paper examines the productive and interpretive process of visual calculation with shapes in the arts and architecture context. Four simple computations with squares are presented. Algebras of shape decompositions with lines are constructed from rules, and lattice diagrams reveal their order. An identity rule for squares is applied, and a minimal decomposition in parts of line segments is offered for the generated shapes, presenting all squares. Decompositions change erratically as emergent squares are presented. When the parts of a square change locally, new forms are identified globally, and the interpretation of the whole shape shifts arousing the mind in various inventions.

1 Introduction

Rules are associated with play, algorithms, and design. The rules of art and design are typically considered as the rules of seeing. They are serendipitous and self-prescribed invented without conscious attention [1]. Kant introduces the idea of free play in art, arguing that every original artwork '*discloses a new rule*.' Original artworks are unexpected and surprising to their producers. They are experienced retrospectively even by their creators, who do not know how they achieved them [2]. Csikszentmihalyi [3] agrees that: "*a playful attitude is typical of creative individuals*". Aristotle also [4] emphasized the association between *chance* (*τύχη*) and *art* (*τέχνη*): "*in a way chance and art are concerned with similar things… art loves chance and chance loves art.*" For similar reasons, the Greeks had placed players under the protection of Mercury (in Latin), or Hermes (*Ερμής* in Greek). Hermes was also the god of *interpretation* or *hermeneia* (*ερμηνεία*), which implies

S. D. Kotsopoulos (✉)
Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: skots@mit.edu

J. S. Gero (ed.), *Design Computing and Cognition'20*,
https://doi.org/10.1007/978-3-030-90625-2_7

that the interpretation of rules is involved in playing, and the interpretation of a course of action leads to the formation of rules.

This paper examines the playful interpretive process of calculating with shapes in the context of the visual arts and architecture. The presentation falls within the area of shape computation theory. The underlying premise is that visual rules capture how a shape is seen and how we act on it. Even if we choose to do nothing, identity rules determine its parts. Shape decompositions present sets of parts for alternative ways of seeing. In creative design, we can neither determine rigid modes of acting on shapes, nor seeing or interpreting their parts.

The title "Design Without Rigid Rules" acknowledges Wittgenstein's remarks on children's games, where rules and interpretations are made up and abandoned effortlessly [5]: "*In ball-games, there is winning and losing, but when a child throws his ball at the wall and catches it again, this feature has disappeared*". Children interpret the playground in unexpected ways. They play to enrich their experience. They advance into new rules as soon as they get tired or exploit a game. Pleasure does not depend on knowing where the game is leading. Sometimes it is pleasant to think of design as a playground, where designers do not commit to rigid rules. This proposition echoes the approach of Lionel March [6]: "*In a shape grammar, a rule is no fetter but, on the contrary, shape rules liberate. They provide the language in which the designer speaks*".

Four computations with squares, in the algebra U_{12}, are presented in this paper, and algebras of shape decompositions with lines are constructed from shape rules. The calculation starts from an initial square modified through addition and translation along the horizontal axis. An identity rule for squares is applied to the produced shape, and lattice diagrams expose the order of the decompositions, demonstrating that shape parts change erratically as emergent squares are presented. For each produced shape, a minimal decomposition in parts of line segments is provided to offer every square in the shape. Since the left-hand side of any visual rule designates an act of "matching", the provisions holding for identities hold for any rule, independently from what is on the right-hand side. The examples demonstrate that even with a narrow objective such as the recognition of emergent squares, free interaction with shapes is indispensable. Local interests determine global possibilities. When the parts of the square change in the local context, new shapes are identified in the global context. The interpretation of the whole shape changes globally each time the description of the square changes locally.

2 Background

The Roman historian Pliny (23–79 c.e.) mentions the Greek practice of throwing the sponge soaked with color at the canvas to produce unpremeditated forms. Pliny describes the struggle of Protogenes in painting the mouth of a panting dog [7]. A variant of this story exists in the 63rd Discourse "*On Fortune*" of Dio Chrysostom (40–115, c.e.). This time the painter is Apelles [8]: "*...finally in a fit of desperation,* *he hurled his sponge at the painting, striking it near the bit… Apelles was delighted by what Fortune had accomplished in his moment of despair and finished his painting, not through his art, but the aid of Fortune*". The tale describes a creative process that relies on accident, where seeing is of primary importance. In the Renaissance, artists used these same ancient ideas. Alberti [9] argued that art emerged by accident when people came across a gnarled tree trunk or a piece of clay, whose contours '*needed only a slight change*' to look like something else. For the artist of the Renaissance, the ordinary experience was a repository of forms if only one was patient enough to look for them. Mantegna secreted zephyrs in the billowing clouds of his images, Bellini's rocks hid human faces, and the folds in Dürer's drapery contained a camouflaged catalog of physiognomic types (left part of Fig. 1). Like the ancients, Botticelli and Leonardo would throw the sponge against the wall and contemplate the stains. Leonardo [10] made an explicit link between the ambiguous visual stimulus and creative imagination: "*I cannot forbear to mention among these precepts a new device for study which may seem trivial and ludicrous it is nevertheless extremely useful in arousing the mind in various inventions. And this is when you look attentively at a wall spotted with stains, or with a mixture of stones and veined marble of various colors, if you have to devise a scene, you may discover a resemblance to various compositions, landscapes beautified with mountains, rivers, rocks, trees, plains, wide valleys and hills in varied arrangements, or again you may see battles and figures in quick action or strange faces, and costumes, with an endless variety of objects which you could reduce to complete and well-drawn forms. And these appear on such walls confusedly, like the sound of bells in whose jangle you may find any name or form you choose to imagine*." Leonardo's attentive glance discovers emergent landscapes, faces, and more from the spots and stains of the wall. In the eighteenth century, a systematic approach based on Leonardo's idea of attentive looking began by the English landscape painter Alexander Cozens, who noticed the influence of the existing stains on the page: "*The stains, though extremely faint, appeared upon revisal to have influenced me, insensibly, in expressing the general appearance of a landscape*." Cozens' method on how to form landscape compositions from stains (center part of Fig. 1) with the aid of five rules was published in 1785 [11]. Making and interpreting random marks became a popular parlor game in the late 1850s, in which Victor Hugo was an enthusiastic amateur (right part of Fig. 1). Hugo's creations influenced the development of automatism in modern art. In 1921 the Swiss psychologist Hermann Rorschach, whose father was an art teacher, began subjecting his patients to an inkblot test. Influenced by the poet Justinus Kerner who made inkblots, drew them, and then wrote poems about what he saw, and Freud, Rorschach created ten symmetrical plates presented to the subject in a predetermined order. The doctor was asking: '*What might this be?*' The blots triggered associations and offered data that would enable the psychologist to make judgments of character.

In 1949 John von Neumann noticed [12] that interpreting a shape or picture does not differ from a Rorschach test: "*with respect to the whole visual machinery of interpreting a picture, of putting something into a picture, we get into domains which you certainly cannot describe in those terms* [of visual analogy]. *Everybody*

Fig. 1 Left: Study of drapery, by A. Dürer. Center: Landscape study by A. Cozens. Right: Inkblot drawing by V. Hugo

will put an interpretation into a Rorschach test, but what interpretation he puts into it is a function of his whole personality and his whole previous history, and this is supposed to be a very good method of making inferences as to what kind of a person he is."If shapes can be interpreted and structured infinitely, they are excluded from the permanent logical-visual description. In the end, von Neumann avoided dealing with visual description and preferred game theory, mathematical economics, quantum logic, and the mathematical models of the atomic bomb.

These references retrace playful processes of different historical ages in which seeing informs perception, art, and design when an observer distinguishes and interprets ambiguous visual appearances. A computational equivalent is presented in this paper, demonstrating that non-rigid modes of seeing are vital in design. Calculations with shapes and algebras of shape decompositions exhibit that even with a narrow target in the mind of the observer, the parts of shapes change erratically when emergent shapes are presented. When the parts of a selected subshape change locally, new forms are identified globally, and the interpretation of the whole scene shifts in front of our eyes, "*arousing the mind in various inventions*".

2.1 Computational Design

Computational design was introduced in the '60s to install generative methods in design and establish mathematical principles through which we can practice and explain the design process. Different algorithmic accounts of design engaged a variety of formal means, such as set theory in Alexander [13], graph theory in Steadman [14], Boolean algebra in March [15], computer programming languages in Eastman [16], and in Mitchell [17], formal syntax in Hillier et al. [18], and shape grammars in Stiny and Gips [19]. Process is arguably the most potent notion of computational design research. Rule-based computational systems were used in the description, prescription, or reference of design processes. In description, computational rules were used to map the actions of designers, to capture the productive steps of a process, and to affirm that a particular course of action produces the desired results. In prescription, rules were used as instructions to determine a norm of production and for implementing devices with strong generative capacities like computer scripts or programs. Finally, rule-based design systems enabled a detailed retrospective reference, analysis, and evaluation of a design process and its outcomes.

In the early years of computational research, computational design methods have found immediate application in engineering, especially in the decomposition and generation of designs from functions [20]. Functional decomposition involves breaking down the function of a device as finely and clearly as possible in a hierarchy of functioning parts. In the context of engineering, the novelty was often superseded by the requirements of economy and efficiency. Conceptual and visual ambiguity was generally treated as noise. In contrast, in the context of the fine arts and architecture, the individuality of expression always remained in demand, and ambiguity was, in most cases, valued as an enabler of creativity. Subsequent computational design research confirmed the importance of ambiguity as an enabler of creativity and invention and as a condition of computing intelligence. Suwa et al. [21] exhibit that the impetus for the invention of essential issues or requirements leading to creative design rests on "*unexpected discoveries*", the acts of attending to visual-spatial features in sketches, which can be unintentional. In comparing how visual artists and scientists interpret graphs and visual art, Kozhevnikov [22] shows that visual imagery supports diverse types of thinking. Visual artists interpret abstract art as abstract representation, while scientists and humanities professionals interpret it literally. In contrast, visual artists interpret graphs aspictures, while scientists interpret graphs abstractly. Tversky [23] claims that creating spatial representations requires both abstraction and ambiguity. Ambiguity enables the multiplicity of interpretation that is the foundation of creative thought. Making and revising these representations reflects the ongoing change of perspective.

The philosopher of computing Brian Cantwell Smith [24] realizes that current computational systems cannot deal with the ambiguity that results from the multiplicity of perspectives. "*Only recently have we begun to know how to build systems that support multiple perspectives on a single situation (even multiple perspectives of much the same kind, let alone perspectives in different, or even incommensurable, conceptual schemes).*" Brian Cantwell Smith notes that designers and artists drawn into computing are not just users of computation. They are participants in its invention. In this light, the input of artists and architects becomes invaluable. Digital, high-speed computing goes beyond the combinatorial capabilities of the human mind, extending variation to a point where it seemingly breaks in creative results, but the outcome relies entirely on what is rigidly encoded in the start. Shape grammar theory was introduced in the '70s to account for calculation with unstructured shapes and shape ambiguity without prescribed underlying representation for shapes.

2.2 Shape Grammars

Research in shape grammars has demonstrated that a corpus of designs can be treated formally by rules similar to a mathematical system. Shape grammar theory was developed for over four decades in a series of papers [25–29], and it was recapitulated in [30]. Stiny emphasizes the importance of seeing in shape grammars: "*seeing informs calculating in art and design so that calculating overtakes computers and what they hold, whether in logic or with data and learning*." Another characteristic of shape calculations is their ambiguous disposition: "*there's plenty of anxiety not knowing what's next, and plenty of pleasure and delight, too, trying to find out*." Visual excitement and delight arise when "*definite things go together in one way, only to change into different things arranged in another way*." Alternate modes of composition and decomposition result from the *embed-and-fuse cycle*, which allows spatial elements to fuse and rules to identify parts without imposing structural constraints. In the context of shape grammars, one is "*free to see, in an open-ended process in which nothing prior is given or known*."

A shape grammar is a rule-based system that generates a set of visual configurations (shapes) by capturing the interaction of elements of 0, 1, 2, or 3 dimensions. Shape grammars include a calculating and a syntactic-interpretive part. The calculating part offers an algebraic framework for computations with shapes. The syntactic-interpretive part consists of productive rule statements assigning structure and meaning to the computations.

The algebraic framework of shape grammars includes shape algebras. An algebra is a set of elements that is closed under a set of operations. Algebras can be distributive, associative, etc., while axioms can distinguish algebraic structures such as rings, lattices, and Boolean algebras. A sub-algebra is a subset of an algebra that is also an algebra.

Shape algebras compute with shapes. A shape algebra U includes the empty shape (0), and it is closed under the Boolean operations of sum (+), difference (–), and product (·). Each shape in U is a finite arrangement of shapes. An algebra U_{ij} includes i-dimensional shapes in a j-dimensional space, with $i \leq j$ and i = 0, 1, 2, and 3 for points, lines, planes and solids.

Euclidean transformations t, acting on shapes, are included in the algebras. Krstic [32] and [33] describes two alternatives for including the transformations in the algebras U_{ij}. The first includes transformations as operators in the set of operations acting on the set $\{U_{ij}\}$ of shapes. This turns algebras U_{ij}, into generalized Boolean algebras with infinite operators. The second proposition is to include the transformations T_j, in the set $\{U_{ij}\}$. This turns shape algebras into two-sorted algebras $\{U_{ij}, T_i\}$, with a Boolean and a group part that handle structure and symmetry respectively.

The part relation $\leq$ applies to shapes made out of lines, planes, or solids, while points can be only identical or discrete. The part relation $\leq$ is a formal relation capturing the fact that spatial elements of the same dimension with i > 0 can be embedded on one another without predetermined joints. The relation $\leq$ is an order relation and renders the sets U_{ij} of shapes, into relatively complemented lattices. The relation $\leq$ is antisymmetric, reflexive, and transitive. Further, each lattice U_{ij} is distributive.

For any two shapes in U_{ij} there is a least element the empty shape, but in all algebras—except U_{00}—there is no upper element since there is no shape containing all shapes. Although there is no upper element for shapes, complements are defined in a relative manner. Hence, each lattice U_{ij} is a relatively complemented one, and because the lattice is distributive, all relative complements are uniquely determined.

The lattice-theoretic operations of join $\cap$, meet $\cup$, and complement substituted with the operations of sum, product, and complement can form a Boolean algebra. The algebra U_{00} containing a single point is an example. The rest of U_{0j}, U_{1j}, U_{2j}, and U_{33}, algebras are not Boolean algebras because they are missing the unit element: there is no shape containing all shapes. Mendelson [34] notes that similar algebraic structures with two binary operations, product, symmetric difference, and 0, without unit, are Boolean rings. Birkoff [35] shows a one-to-one correspondence between Boolean algebras and Boolean rings with unit. Tarski [36] refers to generalized Boolean algebras as Boolean rings. Since for every shape $y \in U_{ij}$, (with $y \neq 0$), there are potentially infinitely many elements x divisible by y the ring is *atomless*.

The syntactic part of a shape grammar uses productive rule statements. Based on Post, [37] a production system associates a set of conditions to a conclusion that the conditions are said to produce. A production system consists of a set of initial strings of symbols and a set of instruction formulas or rules. Given a set of initial strings $\{\varphi_1,\ldots, \varphi_n\}$ with $n \geq 1$, and/or strings derived from them by the rules, a production indicating '$\{\varphi_1,\ldots, \varphi_n\}$ produces φ_{n+1}', or: '$\varphi_1,\ldots, \varphi_n \rightarrow \varphi_{n+1}$'. A shape grammar is a production system consisting of the initial shape and rules that apply to shapes to derive other shapes. As shown in [19, 25] for shapes A, B, C, a rule: $A \rightarrow B$ applies to an initial shape C to generate the shape C', whenever there is a transformation t that makes the shape t(A) part of the shape C. The left part of the rule plays the role of the condition of the rule statement. Since A is a shape and C is another shape, t(A) can match on several parts of C. If the t(A) $\leq$ C, the rule subtracts the shape t(A) from the shape C and adds the shape t(B) in its place. Concisely: $C' = [C\ t(A)] + t(B)$. The sequence of shapes generated by the rule $A \rightarrow B$ is noted: $C \Rightarrow C' \Rightarrow C'' \Rightarrow \ldots \Rightarrow C'\ldots$.

Given that the rule search has been carefully done, rule ordering does not introduce new information. Nonetheless, except generation, shape grammars can serve as formalized explanatory theories giving account for how rules apply to generate designs. A grammatical examination of stylistic change in design is found in Knight [38]. Knight emphasizes the importance of rules and the visual medium that grammars use to characterize the composition or syntax of designs: "*shape grammars are systems of rules for characterizing the composition or syntax of designs in spatial languages…the graphic medium in which shape grammars are expressed is like the medium in which designers and artists conceive and express their work*." Knight [39] focuses on the emergence of unanticipated shapes, apart from the subdivision, and proposes ways to predict them and define rules to handle them within grammars. Knight [40] examines if grammars can be creative and expressive devices: "*shape grammars were developed to be explanatory, to give insights into the designs they generate…the pictorial representation of shape rules endows them with expressiveness and transparency that provide opportunities for the kind of creativity described*." Recently, Knight [41] demonstrates how grammars can be adapted for calculations with material things. The specification of material things involves "*augmenting shapes with other descriptions*." It is defined as an ordered pair consisting of a shape specification and a material specification, with rules illustrating in parallel shape and material states.

Stiny [42] emphasizes the importance of a special computational rule: A → A. Applied to a shape C under a transformation t, it leaves C unchanged. The shapes (C-t{A)) + t[A) and C are identical. Rules of the form A → A are called identities. They lack constructive force, but they have observational value. By distinguishing the shape A within C, the identity organizes its parts in decompositions. A decomposition represents a shape as a finite, non-empty set of parts that sum to the shape. Shapes in U_{ij} with $i > 0$ do not have fixed decompositions. Hence, they enable shapes that look the same to be described by alternative point sets.

Calculation in a shape grammar starts with the initial shape, which is gradually changed by recursive rule application that adds, erases, or replaces some parts with others, or subdivides it in alternative ways.

3 Calculating with Shapes and Their Decompositions

Computational rules encapsulate how a shape is seen and how we act on it. If we decide to do nothing, a rule may still organize a shape in parts. Shapes with lines, planes, and solids can be divided in infinitely many ways. However, in design practice, we only need a finite number of divisions. A composition of shapes is a shape, a finite arrangement of basic elements of 0, 1, 2, or 3 dimensions organized in the space of 1, 2, or 3 dimensions—the parts of such a formation fuse in a unified whole without explicit subdivisions. Decomposition is the analysis of a shape into a finite, non-empty set of elements that sums to the shape. For shapes in algebras, U_{ij} with $i > 0$ decompositions contain shapes rather than collections of points, and computations with parts are carried out as computations with shapes. The two acts of decomposition and composition correspond to what Stiny [16] calls the "*embed and fuse cycle*". It allows spatial elements to fuse and rules to identify parts without structural constraints.

Krstic [43] distinguishes two families of structured decompositions, namely, discrete and bounded. Discrete decompositions offer a minimum structure because there is no overlap between their elements. Bounded decompositions contain the whole shape they analyze and, occasionally, other parts. In bounded decompositions, the parts are always seen in the context of the whole, while the empty shape implies the global context or surroundings. Both global and local contexts are useful in design practice. Discrete and bounded decompositions can be extended to more structured entities that handle continuity and the recasting of shape computations into computations with atoms. Stiny [28, 44] calculates decompositions through topology and Boolean algebra. Krstic [43] uses lattices, hierarchies, topologies, Boolean algebras, and Boolean algebras with operators.

Boolean algebra, topology, and lattices construct algebras of shape decomposition in the featured examples. If $B(\alpha)$ denotes the set of all parts of a shape α, then the set $B(\alpha)$ is a sub-algebra of the algebra in which α is defined. If α is a shape from U_{ij}, then $B(\alpha)$ is a Boolean algebra [33, 42]. The Boolean algebra $B(\alpha)$ is the maximal structure of the shape α, and it is a two-sorted shape algebra that is closed under the symmetry group of α. The algebra $B(\alpha)$ is a proper subalgebra of U_i. The shape α has parts that are shapes from U_{ij} closed under the Euclidean transformations and the Boolean algebra $B(\alpha)$ as upper bound.

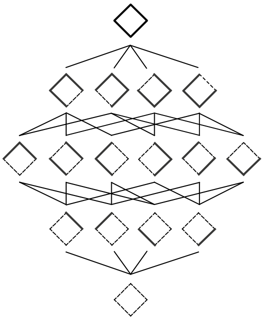
As in Krstic [33], the structure of a shape can be relativized to each of its parts. If α is a shape, A its decomposition, and shape $\beta \leq \alpha$, then β is implicitly structured in decomposition $B = \{\beta \cdot x \mid x \in A\}$, which is the relativization of the structure of α to β.

Jowers et al. [45] show that new shape parts created by a transformation in a design activity "*may decompose the shape into infinitesimal parts for which it is impossible to make provision*". The authors conclude that it is instructive to take a shape perspective on designing: "*choosing/constructing parts, transforming those parts, fusing them back together while revealing new parts*." In the four examples with squares that follow, algebras of shape decompositions with lines are constructed from an identity in the algebra U_{12}. The structure of the whole shape changes globally every time the description of the square is modified locally, and new shapes become noticeable as the parts change.

3.1 Four Compositions with Squares

The calculation of the example begins with a square. The most common decomposition of a square is four maximal lines [sqx] = 4 lines. The four maximal lines form a trivial topology for the square, including 15 subshapes and the empty shape. A Boolean algebra B(x) is defined. The set of finite subsets of B(x) that sum to x is the set of all decompositions of x.

A lattice organizes the subshapes based on the $\leq$ relation. At the top is placed the whole square and at the root the empty shape. The four maximal lines and their combinations occupy the leaves.



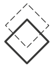
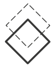
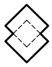
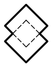
The initial square is modified through the addition and translation of squares along the horizontal axis. The translation of squares produces a grid from where an identity can retrieve new squares. This approach echoes March [6]: "*when two triangles* [squares] *intersect, there is the possibility that additional triangles* [squares] *will emerge. This characteristic of shapes gives rise to spontaneous creation (or destruction).*"

Composition 1 The additive shape rule (left) adds a square. The produced shape contains two squares. Based on the initial decomposition sqx = 4 lines, the new shape is described by a set of 8 maximal lines.

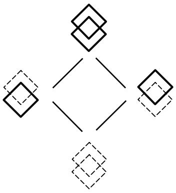
An identity rule for squares (left) is applied to retrieve the squares.
Three squares are distinguished in total. One of them is emergent (right).

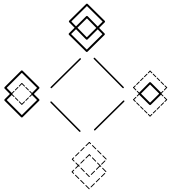
Stiny [42] shows that identities work as observational devices. The structure determined for a shape C is the record of their activity. An identity is applied under a transformation t, and the mapping h(x) = x for every part x of C. Continuity is preserved if the closure operations Γ and Γ′ for C before and after the identity is applied, satisfy two conditions: Γ(t(A)) = t(A), and Γ(x) = Γ′(x). The shape t(A) is closed, and the shape C retains the same topology before and after. The identity recognizes t(A) as a division of C, and a structure is determined for C. Table 1 presents the shapes C, t(A), and its complement C-t(A). It also presents the closure of the subshape that appears in the fourth column. Based on Earl [46], the closure structure associated with the elements of C gives closure of a shape x in C as the smallest element of C containing x.

Table 1 Complement and closure of a subshape A in C

C	t(A)	C-t(A)	Subshape	Closure
				
				

Alternative parts and their complements can form Boolean algebras, which organize the structures in equivalence classes. The two decompositions with squares form two distinct Boolean algebras.



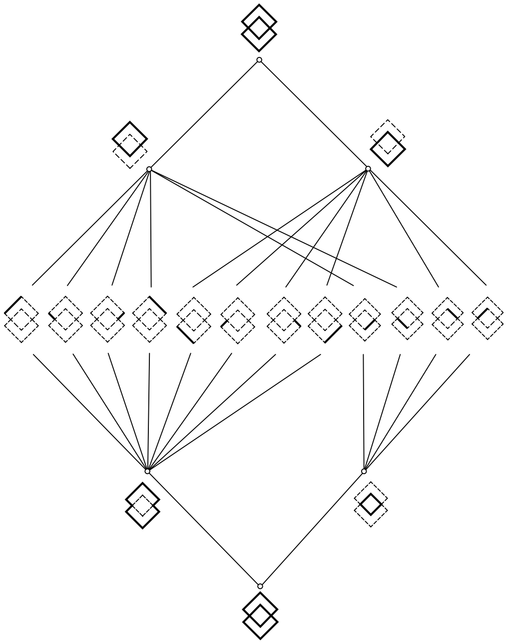


The structure of the shape can be relativized to each of its parts. The top row of Table 2 presents the initial decomposition of shape C, a set of 8 lines. The left column includes the recognized subshapes β and γ. The shapes $\beta \leq C$ and $\gamma \leq C$ are structured in decompositions $B = \{\beta \cdot x \mid x \in C\}$, and $\Gamma = \{\gamma \cdot x \mid x \in C\}$.

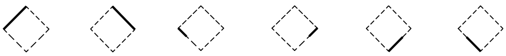
Table 2 Relativization of the two decompositions for C

•								
								
	ø	ø			ø	ø		

The lattice presents a decomposition for C that contains twelve lines.



In this composition, six line segments [sqx = 6] describe the initial square.



Two different decompositions (sums of parts) describe the square in C. They include four and six line segments apiece.

A topology and Boolean algebra, including the 2^{12} permutations of the twelve atoms and the empty shape, exhaust the sub-shapes of C. Even with the narrow objective of recognizing squares, structural change fuels new possibilities. New shapes are accessed globally as the parts of the square change locally. Examples of other shapes identified in the topology are:

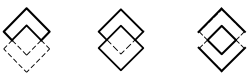
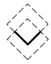


Table 3 A non-empty subshape α of C fills the algebraic conditions of an *ideal*

C	α	x	y	x + y	x • y
					
					

A non-empty subshape α of shape C (with α, C ∈ U_{ij} and i, j ≠ 0) fills the algebraic conditions of an *ideal* (Table 3):

(i) if x ≤ α and y ≤ α then x + y ≤ α;
(ii) if x ≤ α and y ≤ C, then x · y ≤ α.

(ii) is equivalent to the proposition (ii′): if x ≤ α and y ≤ x, then y ≤ α.

Proof Assume (ii) and let shape x ≤ α and shape y ≤ x. Since y ≤ x, it must also be y = x · y. Also, by (ii) y ≤ α. Conversely, assume (ii′) and let x ≤ α and y ≤ C. Since x · y ≤ x, it follows by (ii′) that x · y ≤ α.

The empty shape is part of every shape x. It is also part of every ideal.

Any subshape α of a shape C, with α < C (and α ≠ C), fills the algebraic conditions of a proper ideal. An algebra containing the empty shape as the only proper ideal is called simple. The Boolean algebra U_{00} containing a single point fills the conditions to be called simple.

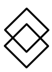
Dually, a non-empty subshape β of shape C (with β, C ∈ U_{ij} and i, j ≠ 0) fills the algebraic conditions of a *filter* (Table 4):

(i) if x ≤ β and y ≤ β, then x · y ≤ β;
(ii) if x ≤ β and y ≤ C, then x + y ≤ β.

(ii) can be replaced by (ii′): if x ≤ β and x ≤ y, then y ≤ β.

The equivalence of the conditions (ii) and (ii′) follows from the equivalence between x ≤ y and x + y = y. The reader is invited to check that these conditions hold, and the proof for *filters* follows dually from that of *ideals*.

Table 4 A non-empty subshape β of C fills the algebraic conditions of a *filter*

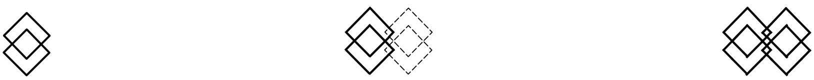
C	β	x	y	x • y	x + y
					

Given any subshape δ of a shape β (δ ≤ β) with δ, β ∈ U_{ij} and i, j ≠ 0, the intersection τ of all subshapes α of β containing δ (i.e., such that δ ≤ α), is itself a subshape of β that contains δ. There is at least one subshape of β containing δ, namely the shape β itself. Assume that the shapes x and y are parts of the intersection τ and that z is part of shape β. If α is any subshape of β containing δ, then x ≤ α and y ≤ α. Hence, the sum x + y ≤ α.

Likewise, x · z ≤ α. Thus, x + y and x · z are in τ, and therefore τ is a subshape of β and also an ideal.

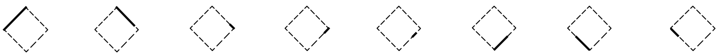
In any shape β ∈ U_{ij} with i, j ≠ 0, a set δ of subshapes of β organizes an inclusion chain of subshapes in β if and only if, for any subshape α_1 and α_2 in δ it is either $\alpha_1 \leq \alpha_2$ or $\alpha_2 \leq \alpha_1$. The relation ≤ orders δ. The union of such an inclusion chain of subshapes is again a subshape of shape β. Composition 1 is modified by addition as it is presented next.

Composition 2 A copy of shape (left) is added under the specified translation (center) to produce the next shape in the sequence (right).



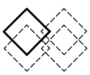
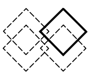
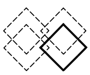
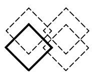
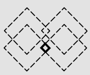
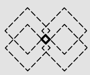
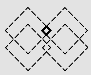
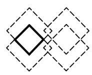
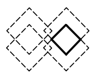
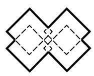
Nine squares are identified. With their complements, the whole shape, and the empty shape, they form two discrete, bounded decompositions. Three emergent squares are presented (Table 5).

A new decomposition is introduced for composition 2, enabling the recognition of all nine squares in the shape. In this context, eight line segments describe the initial square [sqx = 8 lines].



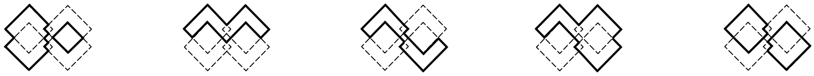
Three different decompositions describe the square in the composition.

Table 5 The nine squares and their complements are presented below. The three emergent squares are shaded.

Discrete Decomposition 1					
					
Discrete Decomposition 2					
					

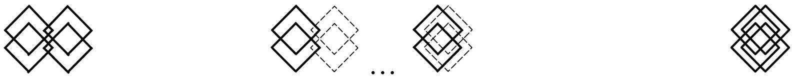
They include four and eight line segments apiece.

The sum of eight line segments per (initial) square yields a set of thirty-two atoms for composition 2. A Boolean algebra, including the 2^{32} permutations, exhausts its sub-shapes. The structural change introduced by recognizing the three emergent squares enables the recognition of other shapes. Examples of new shapes in the topology are the following:



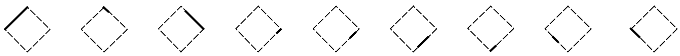
Composition 2 is modified by translating the second, added pair of large squares along the horizontal axis.

Composition 3 A part of composition 2 (left shape) is translated towards the left as shown (center shape) to produce the next shape in the sequence (right shape).



Eleven squares are identified in total. With their complements, the whole shape, and the empty shape, they form four discrete and bounded decompositions, more specifically, Boolean algebras (Table 6). Each decomposition determines alternative parts for the shape. Seven emergent squares are presented.

A new decomposition is introduced for composition 3. Ten atomic line segments describe the initial square [sqx = 10] and enable the presentation of all eleven squares in the shape.



Five alternative sums of parts describe different occurrences of squares in the shape.

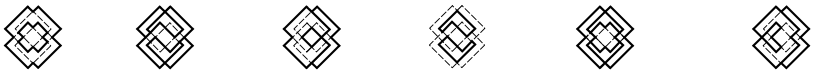
They include four, eight, and ten line segments apiece.

The sum of ten line segments per (initial) square yields a set of forty atoms for the description of composition 3. A topology and Boolean algebra, including the 2^{40} permutations of these atoms, exhaust its sub-shapes. The recognition of the

Table 6 The eleven squares and their complements are presented below. The seven emergent squares are shaded.

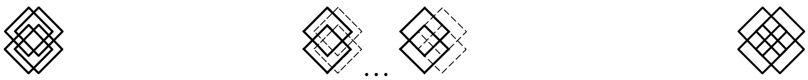
Discrete Decomposition 1				
				
Discrete Decomposition 2				
				
Discrete Decomposition 3				
				
Discrete Decomposition 4				
				

seven emergent squares enables the recognition of other shapes. Examples of new shapes in the topology are the following:



Composition 3 is modified again by translating the second pair of large squares along the horizontal axis.

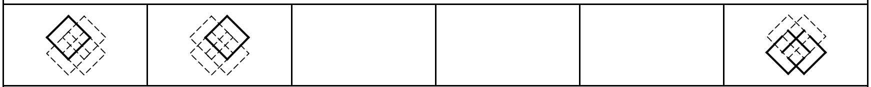
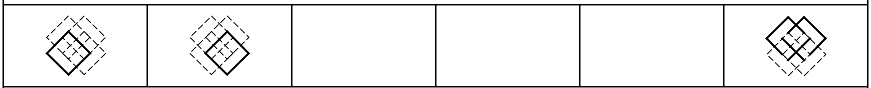
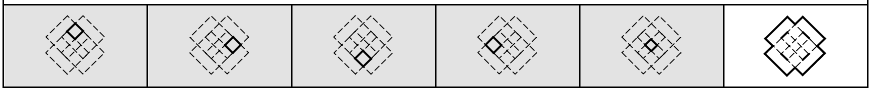
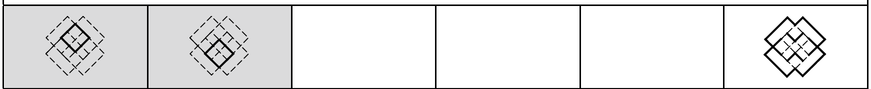
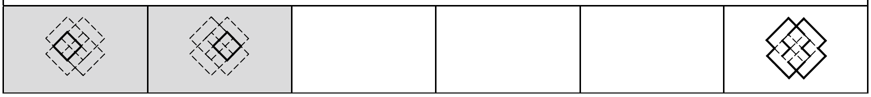
Composition 4 A part of composition 3 (left shape) is translated towards the right as shown (center shape) to produce the next shape in the sequence (right shape).



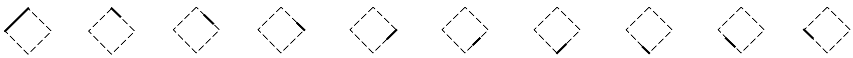
Thirteen squares are recognized in the new shape, with their complements, the whole shape, and the empty shape, they form five discrete, bounded decompositions, Boolean algebras (Table 7). Each decomposition presents alternative parts for the whole shape. The decompositions are reduced to three if symmetrical structures are neglected. Nine emergent squares are identified.

A new decomposition is introduced for the whole shape in composition 4. Ten atomic lines describe the initial square [sqx = 10], enabling the presentation of all

Table 7 The thirteen squares and their complements are presented below. The nine emergent squares are shaded.

Discrete Decomposition 1

Discrete Decomposition 2

Discrete Decomposition 3

Discrete Decomposition 4

Discrete Decomposition 5


thirteen squares in the shape. Although the number of parts is the same as in composition 3, the ten line segments differ in position and size.

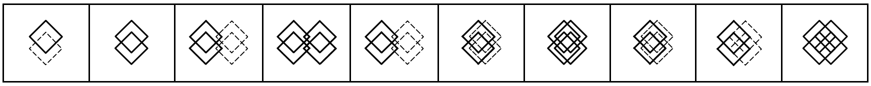


Four alternative sums of parts describe the different occurrences of squares in composition 4. They include four, eight, and ten lines apiece. In all four sets, the line segments differ in position and size from those distinguished in composition 3.

The sum of ten line segments per (initial) square yields a set of thirty-four atoms for composition 4. A topology and Boolean algebra, including the 2^{34} permutations of these atoms, exhaust its sub-shapes. The presentation of the nine emergent squares enables the recognition of other shapes. Examples of such shapes included in the topology are the following:

4 Results

Four computations with squares were presented. The sequence of additions and translations of squares appears next:



In the following Table 8, the top row presents the produced compositions 1, 2, 3, 4. It also presents their products with the initial square and the initial shape.

Table 9 Summarizes the modifications in the structure of the initial square before and after each step of the computation.

Table 8 The produced compositions 1, 2, 3, 4 (top row) and their products with the initial square and the initial shape.

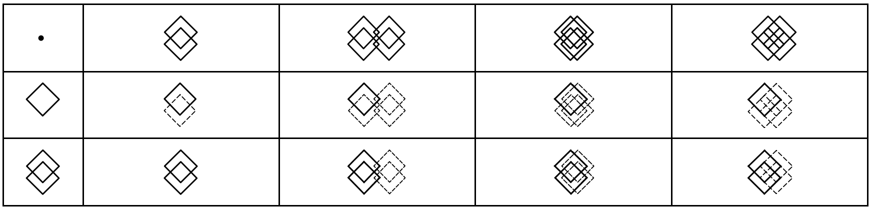
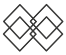


Table 9 Modifications in the structure of the initial square at the consecutive steps of the computation.

Shape	**# of initial squares**	**# of atoms per square before**	**# of emer-gent squares**	**# of squares after compu-tation**	**# of atoms per square after**	**# of atoms per design**	**Atoms per initial square [sqx]**
	1	4	--	1	4	4	
	2	4	1	3	6	8	
	4	6	3	9	8	32	
	4	10	7	11	10	40	
	4	8	9	13	10	34	

5 Discussion

Rules are associated with play, algorithms, and design. Shape rules encapsulate how a shape is seen and how we act on it. If we decide to do nothing, rules can still distinguish the parts of a shape and form a decomposition. In this paper, Boolean algebras, topology, and lattice diagrams were used to construct algebras of shape decompositions from rules, to encapsulate how a shape is seen. A Boolean algebra $B(\alpha)$ of a shape α is the maximal structure of the shape α that serves as its upper bound. The parts of α are also shapes in the same algebra. The Boolean algebras presented in this paper were proper sub-algebras of U_{12}.

Shapes are usually perceived unanalyzed. A composition of shapes involves spatial elements that fuse in a unified whole without explicit subdivisions in parts. However, in creative design and technical disciplines, we explore shape decompositions to satisfy various demands. A shape decomposition analyzes a shape into a finite, nonempty set of parts that sums up the shape. Shape decompositions recast computations with shapes into computations with sets of points. They have an important role because they emphasize diverse ways of seeing. The two acts of composition and decomposition of shapes correspond to the "*embed and fuse cycle*" [31], which allows rules to identify parts and spatial elements to fuse without constraints. Using unanalyzed shapes is of paramount importance for inventing new things. Providing a structure is necessary for assigning semantic, functional, material, and other properties. The interplay between shape and structure is dynamic. Shapes stand for anything we imagine, and having the ability to revise shape and structure to reflect a change in perspective is critical.

The featured calculations with shapes included adding and translating squares along the horizontal axis. An identity rule was used to present every square in each composition. Transformations like translations apply globally, while shape rules apply locally. Emergent squares were retrieved from the produced shapes with an identity rule. With a narrow, local interest in identifying squares, the examination of decompositions before and after the identity revealed that the sums of parts were changing erratically. When the decomposition of a square was changing locally, the parts of the overall shape were shifting globally. The interpretation of the whole shape was shifting in front of our eyes, arousing the mind in various visual inventions.

In design, shape transitions happen haphazardly, and logical structure and interpretation apply retrospectively. The conversion of unstructured shapes to structured sets and back is an essential component of the design process. Having a simultaneous view of both worlds is vital. This is a unique strength of shape grammars in algebras with elements of higher and zero dimensions. We break a shape into parts, provide an interpretation, and then fuse everything back to a unified whole and start anew. The process demands a fair amount of reflection and imagination that slows down progress. It differs from object-oriented programming or using a computer program. Shape, structure, and interpretation evolve slowly, and transparency is critical. Doing the same with a computer, in exchange for speed and efficiency, is an entirely different experience. The computer presents a "correct answer" and removes the seeing/thinking from the process. Design was not meant to be fast nor efficient in this way. More recently, a new generation of computational technology, the Shape Machine, presented in Economou et al. [47], promises to "*fundamentally redefine the way shapes are represented, indexed, queried and operated upon*" and resolve many of the long time overdue problems. Last, a sloppy description with which the designer might feel free to identify alternative modes of interpretation might be better than a meticulously visualized, polished one. The more determined is a description, the more rigid the rules that intervene in interpretation, and the more one has to adjust the action to certain predefined routines of operation. The less determined these details are, the more direct the way of interacting and interpreting in intuitive terms is, and the fewer predetermined rules intervene.

Future research on algebras of decompositions may focus on the requirements of subfields of design, such as architectural design, product design, or engineering. A parallel study on decompositions serving specific subdomains of a workflow could be advanced. An example is the inclusion of material properties, like in Knight's making grammars. The approach could be extended to include other properties, such as semantics, utility, time, or other. The sole provision to all the above is to adhere to the visual character of shape grammars. As Stiny [31] puts it: "*shape grammars from the start have been part and parcel of a full-fledged aesthetic enterprise in open-ended visual perception, and other modes of sense experience*."

References

1. Abrams MH (1953) The mirror and the lamp: romantic theory and the critical tradition. Oxford University Press Inc., p 198
2. Kant I (1790) Critique of the power of judgment (trans: Guyer P) (2000) Cambridge University Press, § 9, 17, 35, 49, 46, 51
3. Csikszentmihalyi M (1997) Creativity: flow and the psychology of discovery and invention. Harper Perennial, p 61
4. Aristotle Nicomachean ethics (trans: Rackham H) (1926) Harvard University Press, pp 1139a–1140b
5. Wittgenstein L (1953) Philosophical investigations (trans: von Wright GH, Rhees R, Anscombe GE). Basil-Blackwell, p 31e
6. March L (1996) Rulebound unruliness. Environ Plann B Plann Des 23:391–399
7. Pliny, Natural history. (trans: Rackham H) (1938) Heinemann, London, p 102
8. Dio Chrysostom (trans: Crosby LH) (1951) Harvard University Press, p 33
9. Alberti LB (1443) On painting and sculpture (trans: Grayson C) (1972), London, Phaidon, p 121
10. Richter JP (1970) The notebooks of Leonardo Da Vinci, (trans) Dover Publications, Inc, New York, p 254
11. Cozens A (1785) A new method of assisting the invention in drawing original compositions of landscape. In: Oppe AP (ed) (1952) Black AC, pp 20–31

12. Von Neumann J (1949) Theory and organization of complicated automata. In: Theory of self-reproducing automata (ed: Burks AW) (1966), University of Illinois Press, pp 46–47
13. Alexander C (1964) Notes on the synthesis of form. Harvard University Press
14. Steadman JP (1973) Graph-theoretic representation of architectural arrangement. In: March L (ed) The architecture of form, Cambridge, pp 94–115
15. March L (1972) A Boolean description of a class of built forms. March L (ed) The architecture of form, Cambridge, pp 41–73
16. Eastman CM (1970) Representations for space planning. Commun ACM 13:242–250
17. Mitchell WJ (1974) An approach to automated generation of minimum cost dwelling unit plans. University of California, Los Angeles
18. Hillier B, Leaman A, Stansall P, Bedford M (1976) Space syntax. Environ Plan B 3:147–185
19. Stiny G, Gips J (1972) Shape grammars and the generative specification in painting and sculpture. Freiman (ed), Information processing, N Holland, p 71
20. Ullman DG (1992) The mechanical design process. McGraw Hill, pp 120–172
21. Suwa M, Gero JS, Purcell T (1999) Unexpected discoveries and s-inventions of design requirements: a key to creative designs. In: Gero JS, Maher ML (eds) Computational models of creative design IV, key centre of design computing and cognition. University of Sydney, pp 297–320
22. Kozhevnikov M (2015) Visual-object versus visual-spatial representations: insights from studying visualization in artists and scientists. In: Gero JS (ed) Studying visual and spatial reasoning for design creativity. Springer, Dordrecht, pp 193–204. https://doi.org/10.1007/978-94-017-9297-4_11
23. Tversky B (2015) On abstraction and ambiguity. In: Gero JS (ed) Studying visual and spatial reasoning for design creativity. Springer, Dordrecht, pp 213–224
24. Smith BC (1996) On the origin of objects. The MIT Press, pp 23–25
25. Stiny G (1980) Introduction to shape and shape grammars. Environ Plann B Plann Des 7:243–251
26. Stiny G (1990) What is a design. Environ Plann B Plann Des 17:97–103
27. Stiny G (1992) Weights. Environ Plann B Plann Des 19:413–430
28. Stiny G (1993) Boolean algebras for shapes and individuals. Environ Plann B Plann Des 20:359–362
29. Stiny G (1994) Shape rules: closure, continuity, and emergence. Environ Plann B Plann Des 21:s49–s78
30. Stiny G (2006) Shape, talking about seeing and doing. The MIT Press
31. Stiny G (2020) Calculating beyond fancy in imagination's magical Realm (forthcoming)
32. Krstic D (1999) Constructing algebras of designs. Environ Plann B Plann Des 26:45–57
33. Krstic D (2005) Shape decompositions and their algebras. Artif Intell Eng Des Anal Manuf 19:261–276
34. Mendelson E (1970) Boolean algebra. McGraw Hill, p 138
35. Birkhoff G (1948) (1960) Lattice theory, vol XXV. American Mathematical Society, Colloquium Publications, p 154, 170
36. Tarski A (1956) Logic, semantics, meta-mathematics. Hackett, p 449
37. Post E (1943) Formal reduction of the general combinatorial decision problem. Am J Math 65:197–268
38. Knight T (1994) Transformations in design: a formal approach to stylistic change and innovation in the visual arts. Cambridge University Press
39. Knight T (2003) Computing with emergence. Environ Plann B Plann Des 30(1):125–155
40. Knight T (2005) Creativity rules. In Gero JS, Maher ML (eds) Computational and cognitive models of creative design VI. Key centre of design computing and cognition. University of Sydney, Australia, pp 155–174
41. Knight T (2015) Shapes and other things. Nexus Network Journal: Architecture and Mathematics, Birkhauser
42. Stiny G (1996) Useless rules. Environ Plann B Plann Des 23:235–237
43. Krstic D (1996) Decompositions of shapes. PhD Thesis. UCLA

44. Stiny G (2001) How to calculate with shapes. In: Antonson EK, Cagan J (eds) Formal engineering design synthesis. Cambridge, pp 24–60
45. Jowers I, Earl C, Stiny G (2019) Shapes, structures and shape grammar implementation. Comput Aided Des 111:80–92
46. Earl C (1997) Shape boundaries. Environ Plann B Plann Des 24:668–687
47. Economou A, Hong T C, Ligler H, Park J (2019) Shape machine: a primer for visual computation. In: Cultural DNA workshop 2019: computational studies on the cultural variation and heredity. KAIST, Daejeon, South Korea