

Shape computations without compositions

Iestyn Jowers¹, Chris Earl¹ and George Stiny²

¹ The Open University, Milton Keynes, UK

² MIT, Cambridge MA, USA

iestyn.jowers@open.ac.uk

Abstract. Parametric CAD supports design explorations through generative methods which compose and transform geometric elements. This paper argues that elementary shape computations do not always correspond to valid compositional shape structures. In many design cases generative rules correspond to compositional structures, but for relatively simple shapes and rules it is not always possible to assign a corresponding compositional structure of parts which account for all operations of the computation. This problem is brought into strong relief when design processes generate multiple compositions according to purpose, such as product structure, assembly, manufacture, etc. Is it possible to specify shape computations which generate just these compositions of parts or are there additional emergent shapes and features? In parallel, combining two compositions would require the associated combined computations to yield a valid composition. Simple examples are presented which throw light on the issues in integrating different product descriptions (i.e. compositions) within parametric CAD.

Keywords: Shape Computation, Composition, Embedding, Parametric CAD

1 Introduction

Design is often formalised in a hierarchical process; as Woodbury states ‘*designers organize their work as...recursive systems of parts with limited interactions between parts*’ [1]. Computer-aided design (CAD) replicates this approach, with design representations constructed hierarchically as assemblies of parts, and in parametric CAD the interactions between the parts are made explicit. Instead of defining single instances of a design concept, parametric models define a range of concepts with varying interactions between parts. Using parametric tools, designers dynamically explore these concepts by modifying parameters that control the interactions between parts, either manually or algorithmically [1]. Accordingly, Aish [2] identifies three themes that characterise a parametric approach to design: geometry, composition and algorithmic thought. *Geometry* includes the primitives used to describe a shape, i.e. points, lines, planes, curves, etc., and the relationships between these. *Composition* defines the hierarchical structure of the geometry, so that a concept is organised according to parts and sub-parts. *Algorithmic thought*, or *Logic* for short, is the process by which concepts are derived from a flow of interactions between geometry, within a defined composition.

The final publication is available at http://dx.doi.org/10.1007/978-981-10-5197-5_19

These three elements, geometry, composition and logic are common generic components of a range of creative processes although articulated differently across specific processes. The differences can be exemplified by comparing creative processes in parametric CAD and sketching. The way that design concepts are explored using pencil and paper in sketching is different from parametric CAD [3]. The primary difference lies in composition. When sketching, designers often change composition during exploration of a design concept to accommodate new ways of seeing, and to develop new avenues to explore [4]. Such switches in interpretation during sketching are visually intuitive, but computationally are difficult to achieve. Parametric tools are powerful at changing geometry, perhaps through explicit parameter changes which modify primitives individually, or in transforming the relationships between primitives. However, when compared with sketching, these parametric tools are less adept at changing composition.

The differences between parametric CAD and sketching, however, are not just a matter of degree, in terms of facility with compositions. The informal computations, combining logic and geometry, effected through sketching do not depend on specific compositions while the computations effected by parametric CAD take place within the context of explicit compositions. The former offers the possibility of computation without composition whilst the latter invokes composition as necessary for its computations. For sketching, although compositions are not necessarily required, they are often used to provide context and direction to see-move-see episodes [5].

This paper addresses the middle ground in design processes between the informal sketch and parametric CAD. It uses a model of shape computation based on shape rules [6] which covers characteristics of both sketching operations and the formal generative transformations of parametric CAD [7]. It demonstrates that these types of shape computation do not necessarily depend on compositions. Furthermore, such computations do not necessarily generate compositions. This perspective exemplifies the difference between sketching and parametric CAD. Compositions are defined in terms of geometry and logic which together constrain modifications of composition. This makes selecting a suitable composition an important step in the construction of parametric models, because a model requires an appropriate set of parts to accommodate exploration of design concepts. Subsequently, changing parts by changing compositions, although possible, essentially resets the design process. More significantly, perhaps, the paper demonstrates that changing compositions may just create a more elaborate chimera, monstrous and wildly imaginative but lacking substance or practicality.

This paper explores this problem of shape composition from the perspective of shape computation, where logic is formalised, not as a flow of interactions between geometry, but as a process of applying shape replacement rules. There is a distinction between these two types of logic, most notably in the way that compositions arise, because shape computations generate compositions. Shape computations mimic the sketching process, and rules are used to accommodate new ways of seeing and new ways of exploring [4]. As such, application of a rule changes the structure of a shape according to recognised parts. Fig. 1 presents an illustrative example, reproduced from Stiny [6]. The shape rule in Fig. 1a, rotates squares through 45° , and is applied to the shape in Fig. 1b by recognising any of the squares embedded in the shape. Applying

the rule gives rise to a network of shapes in a visual shape computation, as illustrated in Fig. 1c. The shape computation is finite, involving only seven distinct shapes, but there are three different compositions of the shape, and these result from how the rule is applied. Applying the rule to the small square gives rise to a different composition than applying it to the large squares, and a third composition results from rotating the two large squares so that line segments overlap.

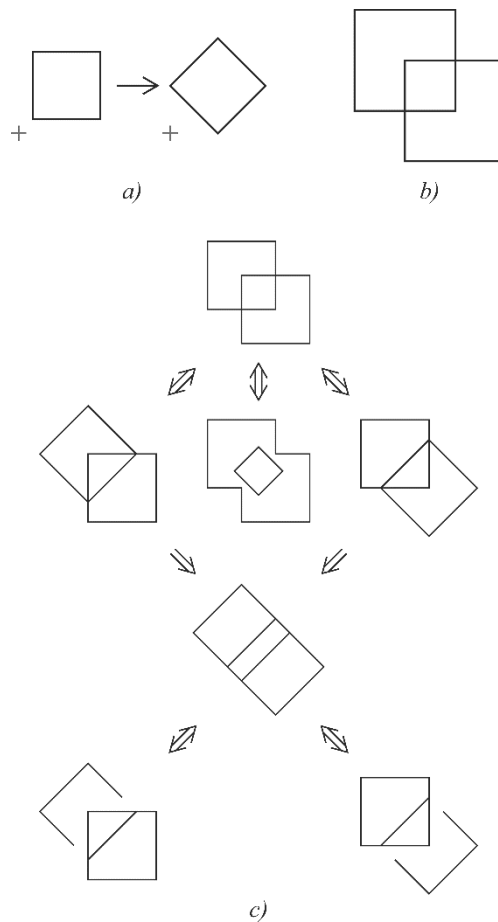


Fig. 1. An elementary shape computation

In shape computation, reinterpretation of compositions is supported by the continuously changing topology of shape [7], however implementing shape computations using parametric CAD is problematic, because changes in composition are difficult to achieve. Instead, it is desirable to identify part structures for shapes that will accommodate all necessary compositions, so that all the parts of interest can be identified and all the shapes in a computation can be derived. This paper explores how such part structures can be identified, and builds on results presented by Stiny [6]. It focusses on the shape

computation illustrated in Fig. 1, which Stiny showed gives rise to a part structure where the maximal edges of the shape in Fig. 1b are given by $ababaababa$, where a and b are both line segments, and $b = a\sqrt{2}$. The paper also considers a slightly more complicated shape computation. To scale the shape computation in Fig. 1, the obvious next step is to add another square to the initial shape, as illustrated in Fig. 2. It might be expected that the addition of the extra small square will result in a part structure that is comparable to that found for the computation in Fig. 1. But, finding a part structure that includes all compositions so that all shapes in the computation can be derived seems to be impossible.

The reason why this is impossible is not obvious, but is intrinsically linked to the periodic palindromic structure that Stiny identified for the computation in Fig. 1. This structure arises as a result of the embedding relation between line segments as described in [8], and in Section 2 key results concerning the part structures that result from embedded lines are reviewed. In Section 3 these results are applied to derive the periodic palindromic structure identified by Stiny for the shape computation in Fig. 1, and Section 4 uses the same results to show that there is no single structure that will accommodate all compositions necessary to carry out the computation if the shape in Fig. 2 is used as the initial shape. This result has implications with respect to the construction of parametric models, and the selection of composition. It suggests that predicting future manipulations is not the only barrier to identifying the correct composition for a particular parametric exploration. Some explorations are impossible to carry out using a single composition, and it is not necessarily possible to combine compositions in a single hierarchical structure.

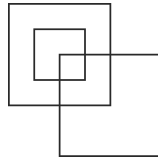


Fig. 2. Three overlapping squares

2 The structure of embedded lines

The part structures necessary to accommodate lines embedded in lines were explored in detail in [8]. There, the discussion was framed around a simple shape computation involving the recognition of parts of a shape. The identity rule in Fig. 3a, was used to recognise squares embedded in shapes composed of two squares sharing a common edge, an example of which is illustrated in Fig. 3b.

Visually, the computation is trivial because the two squares are easily identified as parts of the shape, as illustrated in Fig. 3c. Here, the compositions that result from recognising the either the small or large square are illustrated, but identifying a part structure for the shape that accommodates both these compositions is not trivial. The shared edge of the two squares is the key characteristic of the shape computation, and consequently the problem of identifying a part structure that accommodates both squares is

equivalent to identifying part structures that accommodate the elementary operation of embedding a line inside another line, while taking into consideration the reflective symmetry of the lines. Fig. 4 illustrates the necessary part structure.

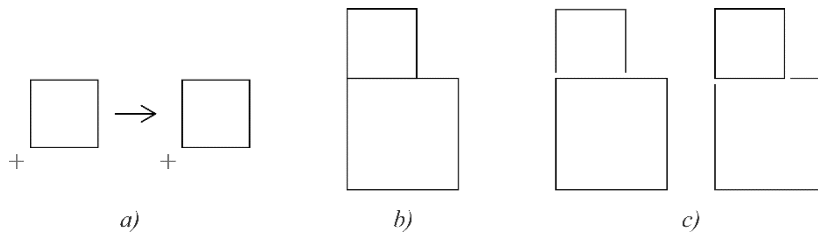


Fig. 3. A trivial shape computation

The triangles are included on the common edge to illustrate part structures while highlighting their symmetry. Each triangle correlates with a line segment embedded in an edge, and these are subdivided into finer structures, representing lines embedded in lines. Embedded lines associated with triangles are symmetrical, and their subdivision into embedded parts is symmetrical; in this sense, the triangles represent the structure of the edges as visual palindromes. The structures of the decomposed line segments are also illustrated in the decomposition of the top edge of the smaller square and the bottom edge of the larger, into line segments. A consequence of this decomposition is that the edges of the larger square and the smaller square have a different part structure, but shape computations that result from applying the rule in Fig. 3a can accommodate this by having two versions of the rule.

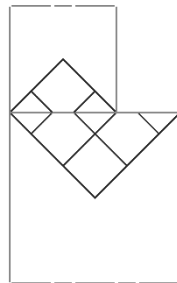


Fig. 4. Part structures resulting from recognition of squares

The structure illustrated in Fig. 4 is the simplest structure that accommodates the short edge being embedded in the long edge while retaining the symmetric properties of the two squares. Intuitively, it might be expected that embedding a short edge as part of a longer edge would simply result in a decomposition of the longer edge to accommodate the shorter. But this is not what is shown in Fig. 4, and the reason for this becomes apparent in the process for deriving the part structures of the edge. This process is illustrated in Fig. 5, and involves resolving the symmetries of the visual palindromes corresponding to the part structure of the edges.

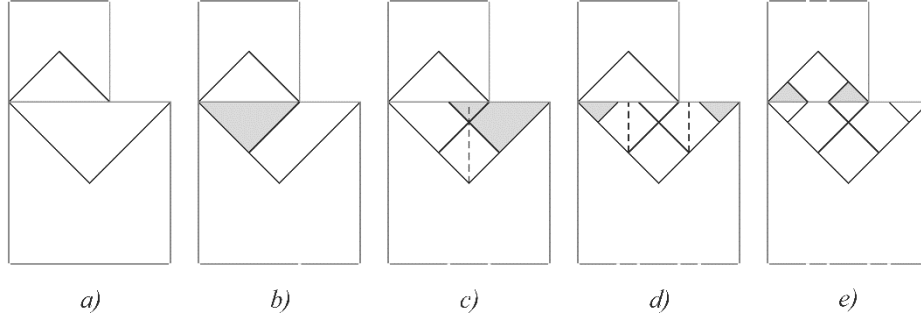


Fig. 5. Deriving the part structure of embedded lines by resolving symmetries

Fig. 5a illustrates the high-level structure of the edges where, to account for the symmetric properties of the squares, each edge is identified as a visual palindrome, represented by a triangle. Fig. 5b illustrates the embedding of the shorter edge in the longer; the structure of the longer edge now incorporates an embedded line that is the length of the shorter edge, represented by the triangle highlighted in grey. This new structure breaks the symmetry of the longer edge, which is addressed in Fig. 5c by reflecting the smaller triangle in the illustrated axis of symmetry of the larger triangle. A new triangle is defined by the overlap, and this represents further subdivision of the visual palindrome; it is this emergent form that requires a finer decomposition of the edges than might be intuitively expected. Fig. 5d resolves the symmetry of the longer edge by reflecting the emergent triangle in the illustrated axes of symmetry. Finally, in Fig. 5e, the structure of the shorter edge is subdivided according to the structure of the longer edge. The resulting part structure accounts for the symmetric properties of both squares, and allows the edge of the smaller square to be embedded in the edge of the larger square. The result is a periodic palindromic structure where the shorter edge can be described by the string uvu and the longer edge can be described by the string $uvuvu$, where u and v represent line segments of different lengths, determined by the ratio of the lengths of the edges of the squares.

Fig. 6 explores the part structures that arise when different initial shapes are used in this shape computation. In all of these, the shapes are composed of two squares sharing a common edge, constrained such that the edge of the smaller square is embedded in the larger, with both sharing an end point, and l , the edge length of the larger squares, is kept constant while n , the edge length of the smaller squares, increases from Fig. 6a to 6h. The shape rule simply recognises the square parts of the shape, and again, triangles are included to illustrate the necessary part structures, whilst highlighting their symmetry. This structure is also reflected in the decomposition of the top edge of the smaller square and the bottom edge of the larger. In Fig. 6a, $n < \frac{1}{2}l$ and embedding the shorter edge in the longer edge results in the part structure that is intuitively expected: the structure of the shorter edge remains unchanged and the structure of the longer edge includes the shorter edge as an embedded part. As a result, the structure of the shorter edge can be described by the string u , where u represents a line of length of $u = n$, and the structure of the longer edge can be described by the string uvu where v represents a line segment of length $v = l - 2n$. Increasing the edge length of the smaller square results

in an increase in the length u , and a decrease in the length v . Specifically, as $n \rightarrow \frac{1}{2}l$, $u \rightarrow \frac{1}{2}l$ and $v \rightarrow 0$, and, in Fig. 6b, when $n = \frac{1}{2}l$, $v = 0$ and the longer edge can be described by the string uu .

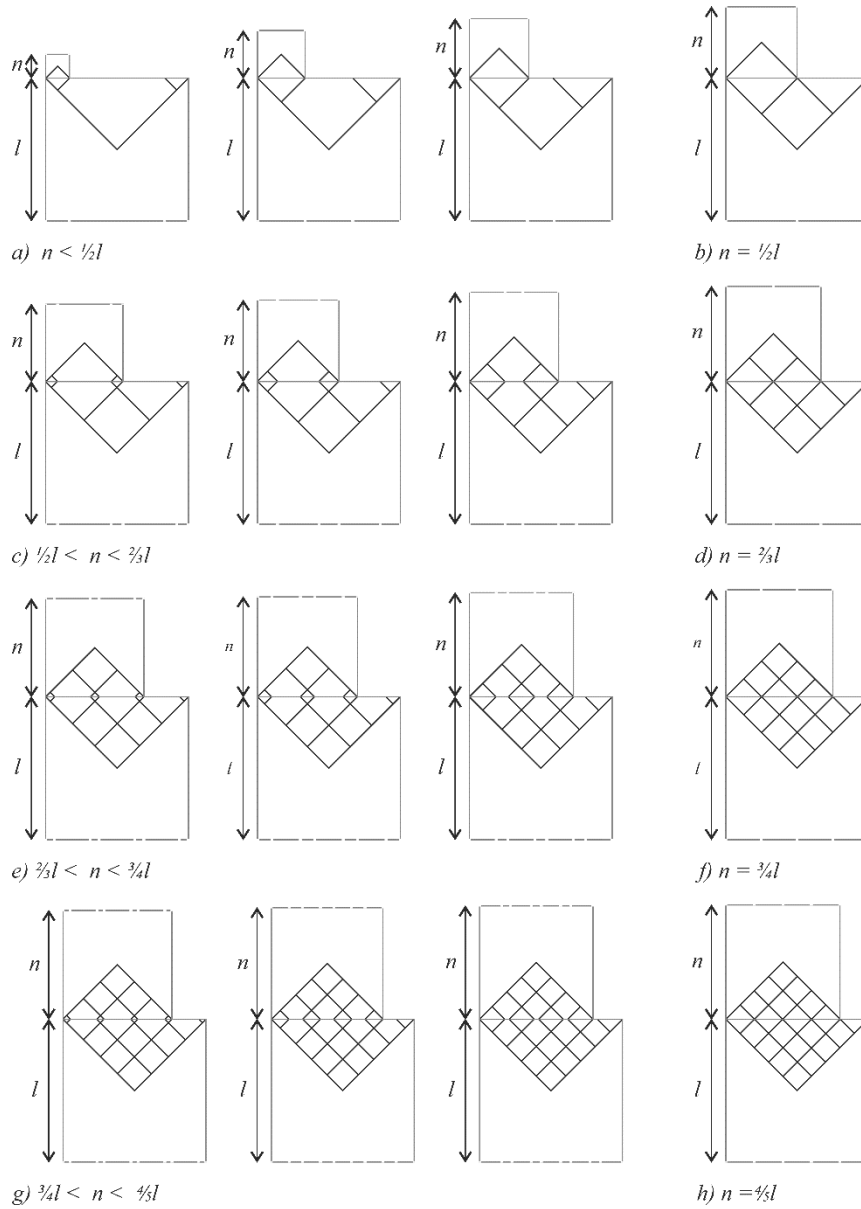


Fig. 6. Part structures resulting from two squares sharing a common edge

In Figs 6c-h, $n > \frac{1}{2}l$ and the embedded shorter edges overlap resulting in the emergence of more complicated structures, similar to Fig. 5. When $n > \frac{1}{2}l$ embedding the shorter edge in the longer edge results in a decomposition of both edges, and as n increases the symbolic descriptions of the resulting part structures can be categorised according to the following cases:

- In Fig. 6c, $\frac{1}{2}l < n < \frac{2}{3}l$, the short edge can be described by uvu and the long edge by $uvuvuv$. As $n \rightarrow \frac{2}{3}l$, $u \rightarrow \frac{1}{3}l$ and $v \rightarrow 0$
- In Fig. 6d, $n = \frac{2}{3}l$, $u = \frac{1}{3}l$ and $v = 0$, the short edge can be described by uu and the long edge by uuu
- In Fig. 6e, $\frac{2}{3}l < n < \frac{3}{4}l$, the short edge can be described by $uvuvu$ and the long edge by $uvuvuvu$. As $n \rightarrow \frac{3}{4}l$, $u \rightarrow \frac{1}{4}l$ and $v \rightarrow 0$
- In Fig. 6f, $n = \frac{3}{4}l$, $u = \frac{1}{4}l$ and $v = 0$, the short edge can be described by uuu and the long edge by $uuuu$
- In Fig. 6g, $\frac{3}{4}l < n < \frac{4}{5}l$, the short edge can be described by $uvuvuvu$ and the long edge by $uvuvuvuvu$, and as $n \rightarrow \frac{4}{5}l$, $u \rightarrow \frac{1}{5}l$ and $v \rightarrow 0$
- In Fig. 6h, $n = \frac{4}{5}l$, $u = \frac{1}{5}l$ and $v = 0$, the short edge can be described by $uuuu$ and the long edge by $uuuuu$

The pattern identified here continues, tending towards the limiting case where $n = l$ and the two squares are the same size, with the edges of both squares represented by a single line. But, as $n \rightarrow l$, $u \rightarrow 0$, and the part structure of the edges gets get finer and finer with the number of line segments increasing. This structure is always defined according to line segments of two alternating lengths, and it can always be described as a periodic palindrome over u and v . In general, the structure of the shorter edge can be described by the string $(uv)^k u$, and the structure of the longer edge can be described by the string $(uv)^{k+1} u$, where u and v represent lines of length u and v , respectively, and k is a positive integer.

To make this explicit, let W represent a line of length l , A represent a line of length m , and B represent a line of length n . Embedding B in W , such that W and B retain their reflective symmetry, gives rise to a palindromic periodic structure, such that

$$\begin{aligned} A &= uv \\ B &= (uv)^k u \\ W = AB &= uv(uv)^k u = (uv)^{k+1} u \end{aligned}$$

where u is a line of length u , v is a line of length v and k is given by $\lceil n/m \rceil - 1$, where $\lceil \cdot \rceil$ is the ceiling function. The period of this structure is $u + v = m$, and given that $W = AB = (uv)^{k+1} u$ and $B = (uv)^k u$, the lengths l and n can be written

$$\begin{aligned} l &= (k+2)u + (k+1)v \\ n &= (k+1)u + kv \end{aligned}$$

and it follows that

$$\begin{aligned} u &= (k+1)n - kl \\ v &= (k+1)l - (k+2)n \end{aligned}$$

For example, if $n = \frac{5}{3}l$, then $m = \frac{3}{2}l$ and $k = \lceil \frac{5}{3} \rceil - 1 = 1$, $u = \frac{1}{4}l$, $v = \frac{1}{6}l$. This confirms observations of Fig. 6c, where the edge of the larger square is composed of three line segments of length u and two line segments of length v , so that $l = 3u + 2v$, and the edge of the smaller square is composed of two line segments of length u and one line segment of length v , so that $n = 2u + v$.

3 The structure of two overlapping squares

Using the results summarised in the previous section, it is possible to derive the palindromic periodic structure identified by Stiny [6] for implementing the computation in Fig. 1. The full computation, including all seven possible shapes, requires three distinct compositions, illustrated in Fig. 7, which result from recognising the small square, the large squares, and from rotating the two large squares so that line segments overlap. To account for these compositions in the part structure of the shape it is necessary that the edges of the large squares include lines of length equal to the edges of the small square and the overlap, as embedded parts. In Fig. 7, these lengths are represented by x , y and p respectively, with $y = 2x$ and $p = (2 - \sqrt{2})x$.

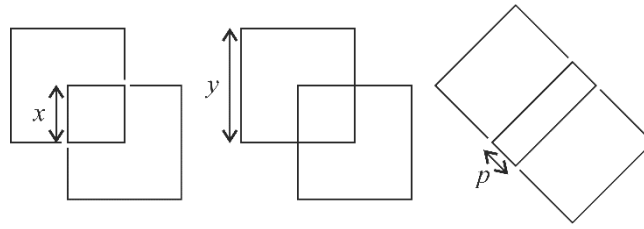


Fig. 7. Compositions necessary to implement the shape computation in Fig. 1

In the previous section it was shown that the structure that results from embedding one line as part of a second line is given by $(uv)^{k+1}u$. For a line of length $x = \frac{1}{2}y$ embedded in a line of length y , k is given by $\lceil x/(y-x) \rceil - 1 = 0$, and the resulting structure is therefore $u_x v_x u_x$, with subscripts used to identify the embedded line. The lengths of line segments u_x and v_x are respectively given by

$$\begin{aligned} u_x &= (k+1)x - ky = x \\ v_x &= (k+1)y - (k+2)x = y - 2x = 0 \end{aligned}$$

and the resulting part structure is therefore $u_x u_x$, as illustrated in Fig. 8a. Similarly, for a line of length $p = (2 - \sqrt{2})x$ embedded in a line of length y , k is given by $\lceil p/(y-p) \rceil - 1 = 0$ and the resulting part structure is therefore $u_p v_p u_p$, as illustrated in Fig. 8b. The lengths of line segments u_p and v_p are respectively given by

$$\begin{aligned} u_p &= (k+1)p - ky = p = (2 - \sqrt{2})x \\ v_p &= (k+1)y - (k+2)p = y - 2p = 2(\sqrt{2} - 1)x \end{aligned}$$

The structures in Fig. 8 accommodate the three compositions illustrated in Fig. 7, and in order to implement the full computation in Fig. 1, a structure needs to be identified that incorporates both of these structures.

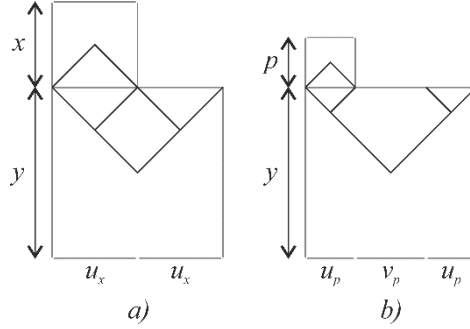


Fig. 8. Part structures resulting from compositions in Fig. 7

Representing the structures symbolically, it is required that

$$u_x u_x = u_p v_p u_p$$

This is shown visually in Fig. 9a, where u_x , u_p and v_p are represented by rectangles. Incorporating $u_x u_x$ in $u_p v_p u_p$ means that the overall part structure includes two distinct but equal parts which can be accommodated by separating v into front and back halves, so that $v_p = v_f v_b$, as illustrated in Fig. 9b.

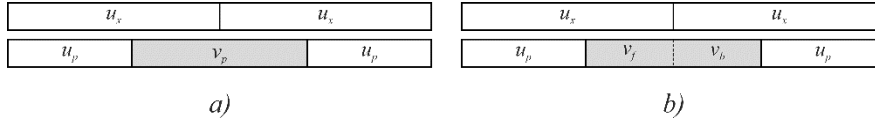


Fig. 9. Combinatorial structure, with lines represented by rectangles

This gives $u_x = u_p v_f$ and $u_x = v_b u_p$, which means that

$$u_p v_f = v_b u_p \quad (1)$$

Here, two structures are equated which share a common part, u_p , which is identified as the prefix of one structure and the suffix of the other. This composition is well understood in the mathematics of combinatorics, and Lyndon and Schützenberger [9] prove the following Lemma:

Lemma 1. If $AB = BC$ and $A \neq \varepsilon$, then $A = UV$, $B = (UV)^k U$ and $C = VU$ for some U , V and some integer $k \geq 0$.

In [8] it was shown that the integer k is given by $\lceil B/A \rceil - 1$, where $\lceil \cdot \rceil$ is the ceiling function, and the lengths of line segments U and V are given by

$$\begin{aligned} U &= (k+1)B - k(A+B) \\ V &= (k+1)(A+B) - (k+2)B \end{aligned}$$

From Lemma 1, the structures $AB = BC$ can be decomposed into a finer structure, $(UV)^{k+1}U$. This is the source of the periodic palindromic line structures identified by Stiny [6] and arises due to the reflective symmetry of the line: lines with identified parts are equal to their mirror images, which themselves contain the same parts. Applying Lemma 1 to equation (1) gives

$$\begin{aligned} v_b &= ab \\ u_p &= (ab)^k a \\ v_f &= ba \end{aligned}$$

and the structure for the edges of the large square is given by $(ab)^{k+1}a$, with k given by $[u_p/v_f]-1 = [(2-\sqrt{2})x/(\sqrt{2}-1)x]-1 = 1$, and the length of line segments a and b are given by

$$\begin{aligned} a &= (k+1)u_p - ku_x = (3-2\sqrt{2})x \\ b &= (k+1)u_x - (k+2)u_p = (3\sqrt{2}-4)x \end{aligned}$$

The resulting structure is $ababaababa$, with $b = a\sqrt{2}$, confirming the result found by Stiny [6]. This structure is illustrated in Fig. 10, where Fig. 10a shows the structure as it applies to the three lines x , y and p , while Fig. 10b shows the structure applied to the three compositions illustrated in Fig. 7.

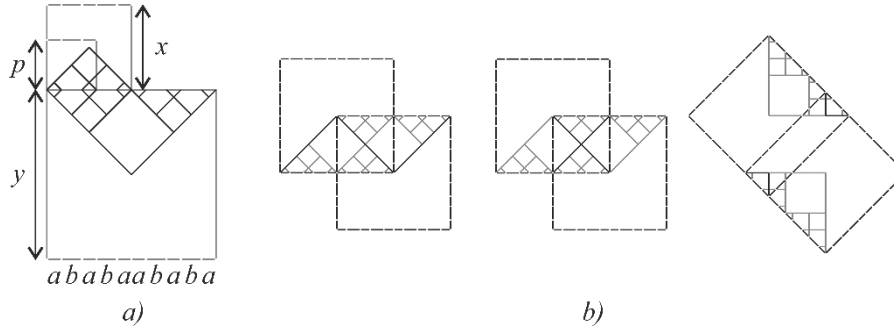


Fig. 10. The part structure of two overlapping squares

4 The structure of three overlapping squares

Applying the shape rule in Fig. 1a to the shape in Fig. 2 also gives rise to a network of shapes in a visual shape computation, as illustrated in Fig. 11. The shapes produced are stylistically very similar to those in Fig. 1c, but with more squares embedded in the shape there are more opportunities to apply the rule. The resulting computation is again finite, involving only twenty-four distinct shapes, and these require five distinct compositions of the shape, as illustrated in Fig. 12. Applying the rule to the small square gives rise to a different composition than applying it to the medium squares, or the large squares, and further compositions results from rotating the two medium squares or the two large squares, so that line segments overlap.

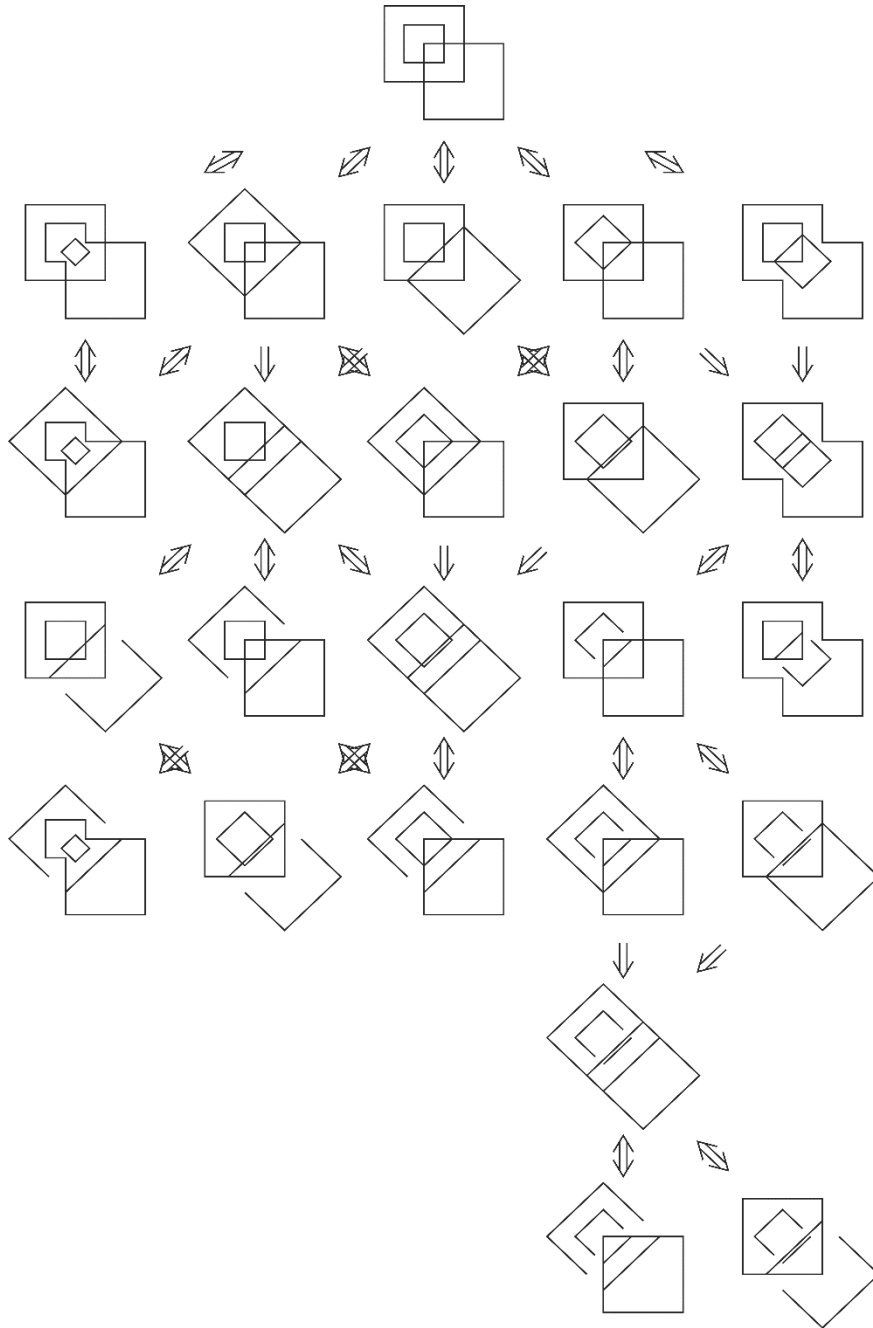


Fig. 11. The result of an elementary shape computation

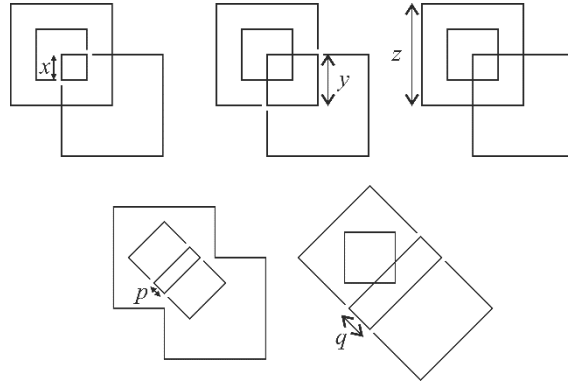


Fig. 12. Compositions necessary to implement the shape computation in Fig. 11

To account for these compositions in the part structure of the shape it is necessary that the edges of the large squares include as embedded parts, lines of length equal to the edges of the small and medium squares as well as the small and large overlap. In Fig. 12, these lengths are represented by x , y , z , p , and q , with $y = 2x$, $z = 2y = 4x$, $p = (2 - \sqrt{2})x$, and $q = 2p = 2(2 - \sqrt{2})x$. As in the previous section, the part structure necessary to accommodate the full computation can be explored by considering the structures that arise from pair-wise combinations of the line segments, x , y , z , p , and q , with lines embedded as parts of other lines. There are ten different line-in-line combinations to consider, and the resulting part-structures are illustrated in Fig. 13. For the sake of legibility, these are not drawn to scale.

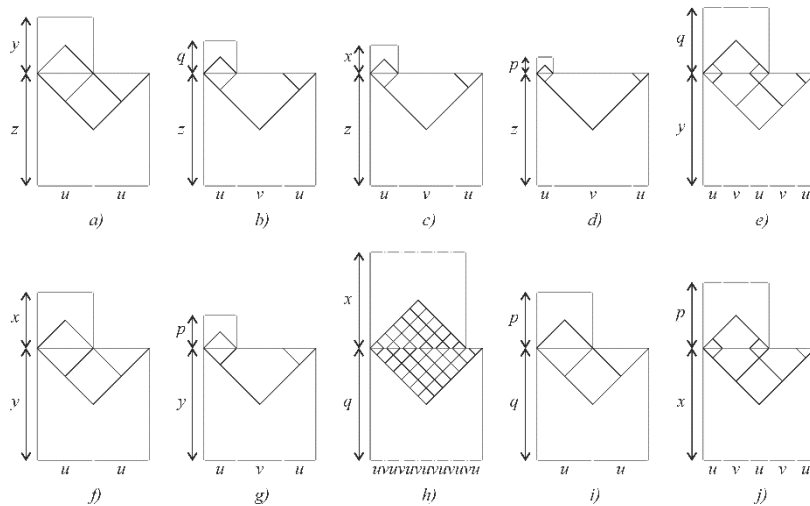


Fig. 13. Part structures resulting from compositions in Fig. 11

The existence of scaling relations between x , y , z , p , and q means that there is similarity between some of the structures, i.e. they are the same under isotropic scaling. Specifically, Fig. 13a is similar to Fig. 13f and Fig. 13i; Fig. 13b is similar to Fig. 13g; Fig. 13e is similar to Fig. 13j. This is because $z = 2y$, $y = 2x$, and $q = 2p$. Using these relations, it is possible to combine structures to accommodate the merger of the different compositions necessary to implement the computation in Fig. 11. For example, Fig. 14 illustrates two such combinations.

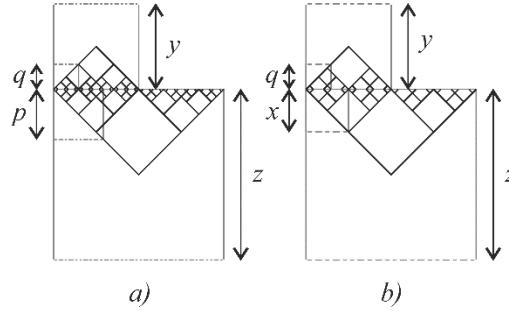


Fig. 14. Example part structures

Fig. 14a combines compositions that enable recognition of the large and medium squares and the rotation of both large and medium squares. It includes lines of length y , p and q , embedded as parts of the line of length z , but it does not include the line of length x which comes from the composition that enables recognition of the small squares. The resulting structure of z is $abaabababaababaababaabababaababa$, with $b = a\sqrt{2}$. Similarly, Fig. 14b combines the compositions that enable recognition of the large, medium and square squares and the rotation of the medium squares. It includes lines of length x , y and q , embedded as parts of the line of length z , but it does not include the line of length p which comes from the composition that enables rotation of the large squares. The resulting structure is $ababaababaababaababa$, with $b = a\sqrt{2}$.

In order to accommodate the full computation in Fig. 11, a part structure needs to be identified that incorporates both structures illustrated in Fig. 14. However, such a merger is impossible because the two structures are incommensurable. This can be shown by attempting to merge the structures illustrated in Fig. 13e and Fig. 13f, which would result in a structure where lines of length x and q are embedded in a line of length y . Representing the structures symbolically, it is required that

$$xx = uvuvu$$

where, $u = 2(3 - 2\sqrt{2})x$ and $v = 2(3\sqrt{2} - 4)x$. This is illustrated visually in Fig. 15a where x , u and v are represented by rectangles. Incorporating xx in $uvuvu$ means that the overall part structure includes two distinct but equal parts which can be accommodated by separating u into front and back halves, so that $u = u_f u_b$, as illustrated in Fig. 15b.

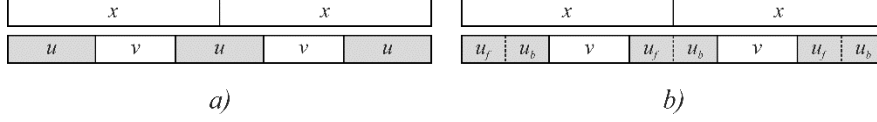


Fig. 15. Combinatorial structure, with lines represented by rectangles

This gives $x = u_f u_b v u_f$ and $x = u_b v u_f u_b$, which means that

$$u_f u_b v u_f = u_b v u_f u_b \quad (2)$$

Here, u_f and u_b are the same length, and equating the prefixes of $u_f u_b v u_f$ and $u_b v u_f u_b$ gives $u_f = u_b = u'$ and equation (2) reduces to

$$u'v = vu' \quad (3)$$

Here, two structures are equated which share the same two parts. This composition is also well understood in the mathematics of combinatorics, and Lyndon and Schützenberger [9] prove the following Lemma:

Lemma 2. If $AB = BA$ then A and B are powers of a common element

Lemma 2 suggests that the structures $u'v = vu'$ can be decomposed into a finer structure defined according to a common element, i.e. $u' = a^i$ and $v = a^j$ for some integers $i \geq 0$ and $j \geq 0$. For such a structure the lengths of u' and v are given by ia and ja respectively, indicating that the ratio v/u' is a rational number j/i . This is clearly not true because $u' = (3 - 2\sqrt{2})x$ and $v = 2(3\sqrt{2} - 4)x$ and $v/u' = 2\sqrt{2}$, which is irrational. There is a contradiction, suggesting that there is no structure that will accommodate all the compositions of the shape illustrated in Fig. 12. This confirms Stiny's [6] suggestion that there is no single structure that will support the shape computation in Fig. 11 and supports the argument that shape compositions cannot be defined prior to carrying out shape explorations.

The impossibility of finding a structure that accommodates the computation in Fig. 11 can be further emphasised by considering the result of applying Lemma 1 to equation (3). Applying Lemma 1, the structures $u'v = vu'$ can be decomposed into a finer structure, to give

$$\begin{aligned} u' &= a_1 b_1 \\ v &= (a_1 b_1)^i a_1 \\ u' &= b_1 a_1 \end{aligned}$$

for some integer i and line segments a_1 and b_1 . But here we have $a_1 b_1 = u' = b_1 a_1$, so that

$$a_1 b_1 = b_1 a_1$$

Applying Lemma 1 again, a_1 and b_1 can be decomposed into a finer structure, to give

$$\begin{aligned}
a_1 &= a_2 b_2 \\
b_1 &= (a_2 b_2)^j a_2 \\
a_1 &= b_2 a_2
\end{aligned}$$

for some integer j and line segments a_2 and b_2 . But here we have $a_2 b_2 = a_1 = b_2 a_2$, so that

$$a_2 b_2 = b_2 a_2$$

and Lemma 1 can be applied again. The same structure continues to reoccur over and over, as u' and v are decomposed into finer and finer parts in a process that tends towards the infinitesimals of Newtonian calculus. The process can only end when $a_\omega = 0$ or $b_\omega = 0$, for some integer ω . But the lengths of a_ω and b_ω are defined following iteration of

$$\begin{aligned}
a_i &= (i+1)u' - i(u'+v) \\
b_i &= (i+1)(u'+v) - (i+2)u'
\end{aligned}$$

and consequently are linear combinations of u' and v . It follows that $a_\omega = 0$ or $b_\omega = 0$ only when $v = iu'$ or $v = (i+1)u'$, i.e. when the length of v is an integer multiple of the length of u' , and u' and v are powers of a common element, as described in Lemma 2. But, this is not the case here, and it can be shown that for each subdivision $b_\omega = a_\omega \sqrt{2}$, and again there is a contradiction.

This argument has explored the possibility that the two part structures in Fig. 14 can be combined in a single composition. The existence of the contradiction proves that this is not possible, and consequently also proves that there is no structure that accommodates all the compositions illustrated in Fig. 12, and supports the shape computation illustrated in Fig. 11. Despite its visual simplicity it is not possible to implement the computation using a static hierarchy of predefined parts.

5 Discussion

Shape computations can be formulated as shape operations analogous to the conventional computations on word strings composed of vocabulary elements. These shape computations can transform shapes in a design process. However, word strings and shapes differ fundamentally. The former have an explicit and well defined composition of vocabulary elements. The latter lacks an explicit composition. Shapes are not without compositions, however these require construction.

The history of construction of a shape can provide one possible composition [6] which will be familiar in parametric CAD. Other compositions arise from the ways that the shapes themselves are perceived with different views corresponding to different compositions. Both construction and perception provide compositions. Consider the example of the shape computation in Fig. 1. This can be viewed as a set (lattice) of constructions using the rule (Fig. 1a), starting with the shape (Fig. 1b). The constructions in the computation serve to pick out specific parts of the eight maximal lines in the shape. They also create a set of new shapes (Fig. 1c).

A shape can have unlimited possible compositions and one way to consider perceptual views is to equate each perception with a specific composition. Any finite set of parts (covering the whole shape) represents a possible composition and thus a possible perceptual view. However, two issues arise. First, the sets of parts will be augmented by sums, differences and intersections of parts. Second where does each composition come from? Or in other words what is the computation which created that composition? One approach to addressing these issues, especially the second, is to consider the compositions arising from the computations. A question might remain about whether each composition has a generating shape computation. However, more basically – and the focus of this paper – there remains a question about whether generating shape computations necessarily create valid compositions; which among other characteristics will have finite sets of parts. The compositions of a shape are descriptions of that shape in terms of its parts. Multiple descriptions of design pose significant problems in applications of parametric CAD in several areas of design. In Architectural Design and Construction, BIM models require consistency across different descriptions, for example structural, layout, service and environmental [10]. In Engineering Design two issues emerge. First, generating and coordinating descriptions in different domains for product development; for example, product structures (bill of materials), manufacturing specifications, assembly, supply, service and maintenance; present significant issues for computational support [11, 12]. The second issue is given two distinct descriptions is there a consistent minimal ‘covering’ description?

The shape computation perspective on compositions (and descriptions) in CAD which is developed in this paper has revealed, albeit in simple examples, a critical problem. This is that even elementary shape computations do not necessarily create possible compositions. This has implications for CAD descriptions such as BoMs (Bills of Materials). If such descriptions are constructed ad-hoc for specific projects, then each associated composition can be considered as the result of a shape computation. However, putting two descriptions together may create a shape computation which does not yield a valid composition. In other words, although there may be a description in terms of the computation (and its associated rules), this description will not correspond to a finite set of parts.

The two squares example (Fig. 1) and its successors (Fig. 2, etc.) provide a striking example of scaling and its impossibility in simple, identifiable cases in principle. Common difficulties with scaling are not a question of logical impossibility, as they are for the successors of the two squares example, but because there is a lack of resources – material or time – to carry them out practicably. Merging the structures in Fig. 14 proves to be impossible because they are incompatible in terms of any underlying, finite set of common, non-zero elements. This highlights the special properties of shape computations that puts them beyond standard analytical techniques in CAD that require explicit, finite compositions beforehand – in shape computation, composition is an outcome of calculating, not a prerequisite – and it suggests both the difficulty of shape computation and the importance of re-examining algorithmic approaches to design that go beyond what is common practice today in parametric modelling and BIM. Shape computation presents a host of challenges both for visual design and for calculating beyond composition.

References

1. Woodburry, R.: Elements of Parametric Design. Routledge, London, (2010)
2. Aish, R.: From Intuition to Precision, in: J. P. Duarte, G. Ducla-Soares, S. A. Zita (eds.), Digital Design: The Quest for New Paradigms, 23rd eCAADe Conference Proceedings, Lisbon, Portugal, 21-24, (2005)
3. Prats, M. and Earl, C. F.: Exploration through drawings in the conceptual stages of product design, in J. Gero (ed.), Design Computing and Cognition '06, Springer, 83–102 (2006)
4. Prats, M., Lim, S., Jowers, I., Garner, S., Chase, S.: Transforming shape in design: Observations from studies of sketching, Design Studies, 30, 503-520, (2009)
5. Schön, D.A., Wiggins, G.: Kinds of seeing and their functions in designing, Design Studies., 13, 135-156, (1992)
6. Stiny, G.: Shape: Talking about Seeing and Doing, MIT Press, Cambridge, (2006)
7. Stiny, G.: Shape rules: closure, continuity, and emergence, Environment and Planning B: Planning and Design, 21, 49-78, (1994)
8. Jowers, I., Earl, C.: Structures in shapes: a perspective from rules and embedding, in Cultural DNA Workshop 2017, KAIST Graduate School of Culture Technology, Republic of Korea, (2017)
9. Lyndon, R.C., Schützenberger, M.P.: The equation $aM = bNcP$ in a free group. Michigan Mathematical Journal, 9, 289-298, (1962)
10. Eastman, C., Teicholz, P., Sacks, R., Liston, K.: BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. Wiley, New Jersey, (2011)
11. McKay, A., Stiny, G.N., de Pennington, A.: Principles of the definition of design structures, International Journal of Computer Integrated Manufacturing, 29, 1-14, (2016)
12. Behera, A. K., McKay, A., Chau, H.H., Robinson, M.A.: Embedding multiple design structures into design definitions: A case study of a collision avoidance robot. In proceedings of International Design Conference, Dubrovnik, Croatia, 119-128, (2016)