

# Discovering Emergent Shapes using a Data-Driven Symbolic Model

John S. Gero  
Min Yan<sup>1</sup>

Design Computing Unit  
Department of Architectural and Design Science  
University of Sydney NSW 2006 Australia

*This paper presents a model for discovering emergent shapes based on the concept of shape hiding and data-driven search. It is founded on representing polyline bounded shapes using infinite maximal lines, an extension of the concept of maximal lines. A process model of shape emergence is presented and a number of examples given which demonstrate the utility of both the representation and the model.*

*Keywords: shape, emergence, symbolic model, data-driven search*

*In searching for a new solution Leonardo projected new meanings into the forms he saw in his old discarded sketches. (Gombrich, 1966)*

## 1 Introduction

Shapes play an important role in representing ideas, concepts and possible physical worlds. Shapes are the way we begin to understand the visual world our visual sense brings to us (Marr, 1982). Shapes play a dominant role in various design domains and particularly in architectural design where they are used not simply as the representation of an idea but also as a representation open to reinterpretation. This reinterpretation is the basis of emergence.

In the conceptual aspects of designing this reinterpretation of what has been drawn appears to play a critical role (Schön, 1983). It provides opportunities for designers to conceptualise what has been drawn differently from what was intended

---

<sup>1</sup>*On leave from Research Institute of Engineering Mechanics, Dalian University of Technology, Dalian 116023, China, Fax: +86-411-471 009*

when it was drawn.

Current CAD systems have not been used extensively during conceptual design for a variety of reasons, one of which is that they freeze the shape being represented and do not allow any other interpretations. Most CAD systems use geometric representations of shape based on line segments and their endpoints. Line segments are grouped together to form a shape. Shape emergence is concerned with finding other shapes derivable from the initial shape. Shape emergence clearly plays an important role in those design domains which use shapes to represent concepts.

Shape emergence in humans has been studied for some time (Gottschaltdt, 1926; Reed, 1974) and is a recognised phenomenon experienced by virtually all humans. Symbolic models related to shapes and to a lesser extent to shape emergence have been presented extensively by Stiny (1980, 1981, 1986, 1990), by Krishnamurti (1980, 1981) and by Krishnamurti and Earl (1992). Tan (1990) presents a limited approach to shape emergence as do Edmonds and Soufi (1992).

In this paper we present two contributions which assist with the discovery of emergent shapes. The first is concerned with a representation which extends the Stiny definition of maximal lines to extended and then infinite maximal lines. The second is concerned with a general process model of shape emergence. We wish to discover emergent shapes derivable from the initial shapes under the following conditions:

- the initial shapes are all closed (bounded) and the boundaries of emergent shapes are boundaries (or parts thereof) of initial shapes;
- the initial shapes are all closed (bounded) and the boundaries of emergent shapes are not necessarily boundaries (or parts thereof) of initial shapes; and
- the initial shapes include unbounded shapes and the boundaries of emergent shapes are not necessarily boundaries (or parts thereof) of initial shapes.

There are two fundamental approaches to discovery applicable to discovering emergent shapes:

- hypothesis-driven search; and
- data-driven search.

In *hypothesis-driven search* a schema is predefined using the representation and the database is searched to determine whether the hypothesised schema can be matched to the data. For example in a geometric representation of shape a triangle might be defined as

$$\text{triangle} = \{v_i, v_j, v_k, s_{ij}, s_{jk}, s_{ki}\}$$

where  $v_i$  = vertex  $i$

$s_{ij}$  = line segment between vertices  $v_i$  and  $v_j$

and the database of vertices and line segments searched for matches.

In *data-driven search* a data element (often called a cue or feature) or combination of data elements is used to traverse the database until some form of closure is satisfied. For example, a feature might simply be a vertex,  $v_i$ , and any line segment attached to it,  $s_{ij}$ . This feature is used to traverse the database until it returns to itself

(possibly subject to side constraints such as no overlaps). Shapes discovered this way do not need a predefined schema.

This paper proceeds by presenting definitions for the concepts used and then provides a new representation for shapes prior to describing a process model of shape emergence. This is followed by examples of shape emergence using a data-driven symbolic model.

## 2 Definitions

A *primary shape* is a shape that is represented explicitly, initially, and thus can be input and manipulated by specifying its behaviours. An *emergent shape* is a shape that exists only implicitly in a primary shape, and is never explicitly input and is not represented at input time (Mitchell, 1992). An *emergent visual shape* is an emergent shape that is cognitively recognised as an emergent shape (rather than simply symbolically constructed). Figure 1 shows examples of primary shape, emergent shape and emergent visual shape. The process of recognizing emergent shapes and emergent visual shapes from a primary shape is called *shape emergence*.

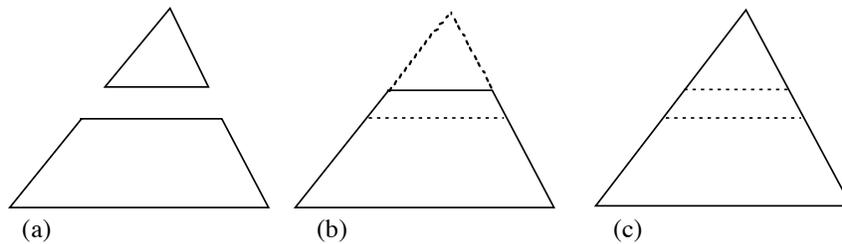
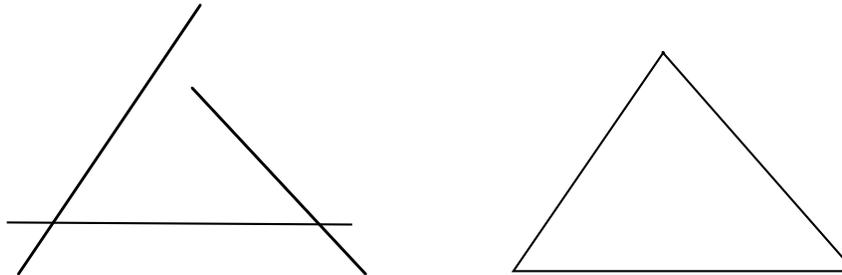


Fig. 1. (a) primary shape, (b) emergent shape, and (c) emergent visual shape.

A *polyline shape* is composed of a set of straight lines. A *bounded polyline shape* is an enclosed polyline shape, for any point on the boundary of which there exists at least one circuit composed of line segments which starts from and ends at that point without covering any line segment more than once. Figure 2 shows examples of an unbounded polyline shape and a bounded polyline shape. In the following, we call a bounded polyline shape a shape for short.



(a) (b)

Fig. 2. (a) unbounded polyline shape and (b) bounded polyline shape.

There are four different concepts of interest about lines: line segment; maximal line; extended maximal line; and infinite maximal line.

A *line segment*, denoted as  $l^s$ , is a part of a line between two points. A *maximal line*, denoted as  $l^m$ , is a line segment which embeds at least one line segment (Stiny, 1980). An *extended maximal line*, denoted as  $l^e$ , is a line segment within which at least one maximal line is embedded. An *infinite maximal line*, denoted as  $l$ , is the infinite line in which an extended maximal line is embedded. Figure 3 illustrates the concepts of line segment, maximal line, extended maximal line and infinite maximal line.

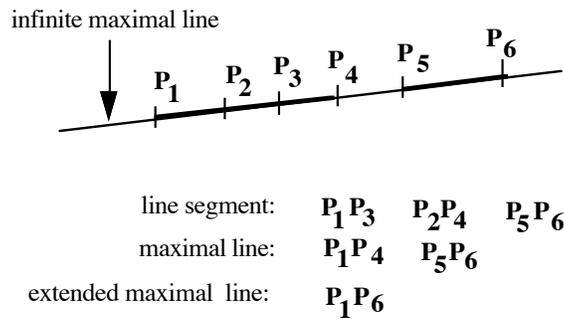


Fig. 3. Line segment, maximal line, extended maximal line and infinite maximal line;  $P_i$  is an endpoint of a line segment.

### 3 A Symbolic Representation for Shapes

#### 3.1 Representation

A conventional way to represent a shape is to use point coordinates as primitives (Stiny 1980). In this way, a line segment is described by the coordinates of its two endpoints. A shape as a set of line segments is represented through a set of point coordinates. Shape recognition based upon this coordinate representation is coordinate computation in which geometric properties of shapes are not explicitly applied, and are dependent upon calculating accuracy (Tan, 1990).

In this section, we use infinite maximal lines as the representation primitive to construct a symbolic representation of shapes to support shape recognition by symbolic reasoning.

Using infinite maximal lines as representative primitives, the general form of the symbolic representation of shapes is (Gero, 1992)

$$S = \{ N ; \text{constraints} \} \quad (1)$$

where  $N$  is the cardinality, i.e., the number of infinite maximal lines constituting shape  $S$ , and the constraints constrain behaviours or properties resulting from the infinite maximal lines, based upon which particular shapes are defined.

### 3.2 Behaviours of lines

Behaviours of infinite maximal lines include three kinds of properties: topological properties, geometrical properties and dimensional properties. Two main topological properties of a set of infinite maximal lines are intersection and segment. Intersection is a behaviour about the contiguity of two lines. If we assume that two parallel lines intersect at the infinite, then any two lines have an intersection. When more than two lines are contiguous with more than one intersection, the part of a line between two intersections forms a segment. The geometrical property we are interested in is the slope of an infinite maximal line. The dimensional property we are concerned with here is the length of a line segment which embeds in a infinite maximal line.

Let the symbol  $i_{kj}$  denote the intersection of two infinite maximal lines  $l_k$  and  $l_j$ , and  $i_{kjm}$  to denote the common intersection of three lines  $l_k$ ,  $l_j$  and  $l_m$ . Obviously, the order of the subscripts of  $i_{kj}$  is not significant, i.e.,  $i_{kj}$  and  $i_{jk}$  represents the same intersection.

The intersection behaviour of two lines  $l_k$  and  $l_j$  is represented as

$$(i_{kj}) \quad (2)$$

The segment behaviour of three lines  $l_k$ ,  $l_j$  and  $l_q$  is represented as

$$(i_{kj}, i_{qj}) \quad (3)$$

where  $k$  is not equal to  $q$ , and (3) represents a line segment embedded in the infinite maximal line  $l_j$ . The length of a segment  $[i_{kj}, i_{kp}]$  is represented as

$$d(i_{kj}, i_{kp}) \quad (4)$$

### 3.3 Constraints

There are three classes of constraints on infinite maximal lines: topological constraints on intersections, geometrical constraints on their slopes and dimensional constraints on the lengths of line segments. Therefore, the representation (1) can be extended into (5).

$$S = \{ N; \text{topological constraints}; \text{geometrical constraints}; \text{dimensional constraints} \} \quad (5)$$

*Topological constraints*

Topological constraints concern the structures within which intersections and segments are organized. They are represented as groups of intersections. There are three kinds of intersection groups: ordinary group, adjacent group and enclosed group, which respectively specify three kinds of topological structures organising intersections and line segments in different ways.

An ordinary group is represented by pairs of parentheses: "(" and ")", in which any two intersections may represent a line segment if the two intersections satisfy (3). The order of intersections in an ordinary group is of no significance. An adjacent group is represented by a pair of angle brackets, "<" and ">", in which only two adjacent intersections can represent a line segment if they satisfy (3). It is stipulated here that the first and the last intersections in an adjacent group is adjacent to each other. Obviously, the order of intersections in an adjacent group is significant. The same set of intersections with different adjacent orders represent different sets of line segments. An enclosed group, represented by pairs of square brackets, "[" and "]", is a subcase of an adjacent group, any two adjacent intersections of which must satisfy (3). An enclosed group represents a circuit of line segments, i.e. a bounded polyline shape. For example,

$$(i_{12}, i_{23} i_{13}) \quad (6)$$

the representation (6) represents a topological structure which organizes three intersections and three segments:

- intersection  $i_{12}$  of  $l_1$  and  $l_2$  ;
- intersection  $i_{23}$  of  $l_3$  and  $l_2$  ;
- intersection  $i_{13}$  of  $l_1$  and  $l_3$  ;
- segment  $(i_{12}, i_{13})$  embedded in  $l_1$ ;
- segment  $(i_{12}, i_{23})$  embedded in  $l_2$ ;
- segments  $(i_{23}, i_{13})$  embedded in line  $l_3$ .

Topological constraints may be represented as more than one group of intersections. In this case, only those line segments exist which are composed of two intersections that come from the same group.

#### *Geometrical constraints*

Three geometrical constraints on infinite maximal lines are parallel, perpendicular and skew.

The parallel constraint on two lines  $l_1$  and  $l_2$  is denoted as  $l_1 \parallel l_2$ .

The perpendicular constraint on two lines  $l_1$  and  $l_2$  is denoted as  $l_1 \perp l_2$ .

The skew constraint on two lines is denoted as  $l_1 \times l_2$ . Where  $l_1$  and  $l_2$  are neither parallel nor perpendicular, they are said to be skew to each other.

### *Dimensional constraints*

Examples of dimensional constraints are as follows.

$$\begin{aligned}d(i_{kj}, i_{pj}) &= d(i_{sj}, i_{rj}) \\d(i_{kj}, i_{pj}) &< 0.1d(i_{kj}, i_{rj})\end{aligned}$$

### 3.4 Cardinality

Considering a shape as a set of figure elements isolated from a background (Granovskaya et al., 1987), the cardinality in (1) is a measure of complexity of the background. The more lines in the background, the more shapes and the more complicated the shapes which can be formed from it. For instance, from three lines we may recognize a triangle, but we can never find a quadrilateral. Furthermore, more triangles can be found from five lines than from four lines, because more intersections can be formed from five lines than from four lines. The dependency relation between cardinality and number of subshapes is described by the cardinality theorem (Gero and Yan, 1992).

Given a shape defined by an intersection group, the minimum cardinality of the shape can be determined by a function Ca:

$$\text{Given } S = \{ N; [ i_{kj}, i_{km}, \dots, i_{pq} ] \},$$

$$N = Ca(S) \tag{7}$$

To end this section, we give some examples of symbolic representations of shapes as follows.

*Triangle:*

$$S1 = \{ 3; [i_{jk}, i_{kp}, i_{jp}] \} \tag{8}$$

*Four-sided shape:*

$$S2 = \{ 4; [i_{jk}, i_{kp}, i_{pq}, i_{jq}] \} \tag{9}$$

*Parallelogram:*

$$S3 = \{ 4; [i_{jk}, i_{kp}, i_{pq}, i_{jq}]; l_k // l_q, l_j // l_p \} \tag{10}$$

*Square:*

$$\begin{aligned}S11 &= \{ 4; [i_{jk}, i_{kp}, i_{pq}, i_{jq}]; l_k // l_q, l_j // l_p, l_j \perp l_k; \\&\quad d(i_{jk}, i_{kp}) = d(i_{pq}, i_{jq}) \} \tag{11}\end{aligned}$$

## 4 A Process Model of Shape Emergence

Psychologists have reported that, when perceiving a picture visually, a person

attaches to it a certain organization (Granovskaya et al., 1987). This organization involves dividing everything in the visual field into a figure and a background, and grouping elements in the figure into structures. Thus, shape recognition involves two steps: (a) isolating a figure from a background; and (b) structuring elements of the figure.

However, shape recognition involving the above two steps is not enough for emergent shape recognition which commences with a structured figure rather than an unstructured background. Emergent shape recognition restructures an existing structured figure. It is more difficult to build a new structure from an already structured figure than from an unstructured background, because the existing structure results in fixation (Purcell and Gero, 1991) that encumbers the establishment of a new structure.

Therefore, emergent shape recognition involves breaking or hiding existing constraint structures to remove fixation, and restructuring constraints to explicitly represent emergent shapes. We propose a process model of shape emergence, described in Figure 4, which includes two fundamental steps: shape hiding and shape emergence.

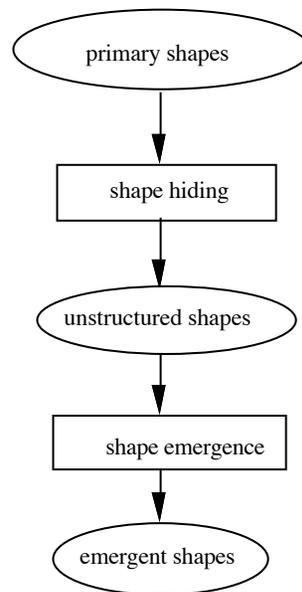


Fig. 4. A process model for shape emergence.

#### 4.1 *Shape hiding*

Shape hiding makes shapes which are explicitly represented become implicit through a change in representation. When a shape is explicitly represented, the constraints defining it are grouped or structured. When the constraint structure is destroyed or relaxed, the explicitly represented shape becomes implicit.

Based upon the symbolic representation of shapes we propose, shape hiding is

carried out by an ungrouping of intersections operation. Ungrouping intersections is carried out by an operator, Ug, which combines intersections from different groups into a single group.

As has been described in Section 3.3, an intersection group represents certain line segments, and a line segment does not exist if the two intersections defining it come from different groups. The operator Ug combines intersections from different groups into one group, so it relaxes topological constraints on the existence of line segments to the existence of extended maximal lines. That is, after ungrouping intersections, a line segment may be defined by two intersections that were in different groups in the primary shape.

As we shall see in the next section the single group of intersections resulting from shape hiding will be augmented by intersections of infinite maximal lines which did not exist in the primary shape.

#### 4.2 Shape Emergence

Shape emergence is the process of discovering possible shapes that were not explicitly represented in the primary shape. It consists of two steps: constraint derivation and shape discovery. Constraint derivation derives, from the constraints which exist in the symbolic representation, new constraints which do not exist in that representation. The following are the basic reasoning rules for constraint derivation.

$$R1: \quad l_j // l_k \wedge l_k // l_p \Rightarrow l_j // l_p \quad (12)$$

$$R2: \quad l_j \perp l_k \wedge l_k \perp l_p \Rightarrow l_j // l_p \quad (13)$$

$$R3: \quad l_j \perp l_k \wedge l_k // l_p \Rightarrow l_j \perp l_p \quad (14)$$

$$R4: \quad l_j // l_k \wedge l_k \perp l_p \Rightarrow l_j \perp l_p \quad (15)$$

$$R5: \quad l_j \perp l_k \Rightarrow l_j \perp l_k \quad (16)$$

$$R6: \quad l_j \perp l_k \Leftrightarrow i_{jk} \quad (17)$$

$$R7: \quad l_j // l_p \wedge l_k // l_q \wedge l_j \perp l_k \Leftrightarrow d(i_{jk}, i_{jq}) = d(i_{kp}, i_{pq}), \\ d(i_{jk}, i_{kp}) = d(i_{jq}, i_{pq}) \quad (18)$$

where  $A \Rightarrow B$  implies IF A THEN B;  $A \Leftrightarrow B$  implies IF A THEN B and IF B THEN A.

There are two strategies for shape discovery: hypothesis-driven search and data-driven search. Hypothesis-driven search has been investigated in Gero and Yan (1992). Here we introduce a data-driven search method for emergent shape discovery.

In data-driven search, shapes are discovered by applying the following rules to search suitable intersections until an intersection group is formed which represents a bounded polyline shapes. In the following, **I** stands for a set or group of intersections.

$$R8: \quad (i_{kj}) \Rightarrow \langle i_{kj}, i_{km} \rangle; i_{km} \neq i_{kj} \quad (19)$$

$$R9: \quad \mathbf{I} \perp \langle i_{kj}, i_{km} \rangle \Rightarrow \mathbf{I} \perp \langle i_{kj}, i_{km}, i_{pm} \rangle; i_{pm} \text{ does not belong to } \mathbf{I} \quad (20)$$

$$R10: \quad (i_{pm}) \perp \mathbf{I} \perp \langle i_{kj}, i_{km} \rangle \Rightarrow \langle i_{pm} \rangle \perp \mathbf{I} \perp \langle i_{kj}, i_{km}, i_{pm} \rangle;$$

$$i_{pm} \text{ does not belong to } \mathbf{I} \quad (21)$$

$$\text{R11: } \langle i_{pm}, i_{ps} \rangle \in \mathbf{I} \wedge \langle i_{kj}, i_{km} \rangle \Rightarrow \langle i_{pm}, i_{ps} \rangle \in \mathbf{I} \wedge \langle i_{kj}, i_{km}, i_{ps} \rangle; \quad (22)$$

$$i_{ps} \text{ does not belong to } \mathbf{I} \quad (22)$$

$$\text{R12: } \langle i_{kj}, i_{pj}, \dots, i_{nq}, i_{nk}, i_{kj}, i_{pj} \rangle \Rightarrow [ i_{kj}, i_{pj}, \dots, i_{nq}, i_{nk} ] \quad (23)$$

Where R8 is the starting rule for the data-driven search. It selects an intersection  $i_{kj}$  from the unstructured representation as an endpoint to construct a line segment which embeds in the infinite maximal line  $l_k$ . R9 constructs line segments with a new intersection which has not been used in the search process. R10, R11 and R12 are ending rules for the construction of a bounded polyline shape .

If an intersection group includes new intersections which do not exist in the primary shape, some dimensional constraints may be applied to it to check if its visual tension (Garrett, 1967) is sufficient for it to be recognised as an emergent visual shape rather than only an emergent shape. This problem is discussed in Gero and Yan (1992), and is not described further here.

## 5 An Example

In this section, we use an example to illustrate how emergent shapes are discovered using the data-driven approach.

The primary shape **S** shown in Figure 5 is composed of two triangles **T1** and **T2**. One symbolic representation of the shape in Figure 5 is (24).

$$\mathbf{S} = \{ 5; [ i_{12}, i_{23}, i_{13}], [ i_{45}, i_{35}, i_{34}]; l_1 // l_4, l_2 // l_5 \} \quad (24)$$

Through shape hiding by using the ungrouping operator, Ug, (24) becomes (25) which is a unstructured representation where no shape is explicitly represented; triangles **T1** and **T2** are hidden.

$$\mathbf{S} = \{ 5; (i_{12}, i_{23}, i_{13}, i_{45}, i_{35}, i_{34}); l_1 // l_4, l_2 // l_5 \} \quad (25)$$

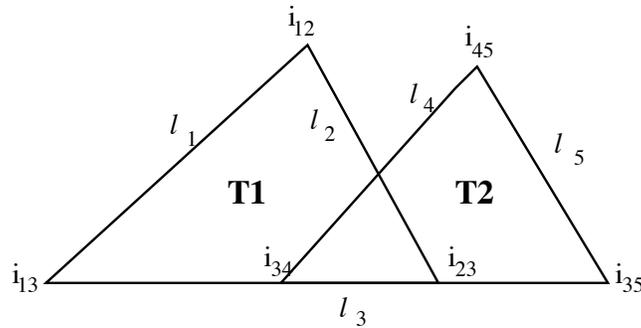


Figure 5. A primary shape composed of two overlapping triangles **T1** and **T2** , where  $l_i$  are infinite maximal lines and  $i_{jk}$  are intersections of  $l_j$  and  $l_k$ .

Shape emergence starts by applying the rules R1 to R7 to the unstructured representation (25):

$$\begin{aligned}
 i_{12} &\Rightarrow l_1 \sqcap l_2 ; && \text{by R6} \\
 l_1 \sqcap l_2, l_1 // l_4 &\Rightarrow l_2 \sqcap l_4 ; && \text{by R4} \\
 l_2 \sqcap l_4 &\Rightarrow i_{24} ; && \text{by R6} \\
 l_4 \sqcap l_5, l_1 // l_4 &\Rightarrow l_1 \sqcap l_5 ; && \text{by R4} \\
 l_1 \sqcap l_5 &\Rightarrow i_{15} ; && \text{by R6}
 \end{aligned}$$

The results of the above reasoning are two new intersections  $i_{24}$  and  $i_{15}$  , and (25) then becomes (26). One possible shape defined by (26) is shown in Figure 6.

$$\mathbf{S} = \{ 5; (i_{12}, i_{23}, i_{13}, i_{45}, i_{35}, i_{34}, i_{24}, i_{15}); l_1 // l_4, l_2 // l_5 \} \quad (26)$$

With representation (26), shape emergence continues as a data-driven search, rules R8 to R12 are applied to (26) as follows.

$$\begin{aligned}
 ( i_{13} ) &\Rightarrow \langle i_{13}, i_{12} \rangle ; && \text{by R8} \\
 \langle i_{13}, i_{12} \rangle &\Rightarrow \langle i_{13}, i_{12}, i_{24} \rangle ; && \text{by R9} \\
 \langle i_{13}, i_{12}, i_{24} \rangle &\Rightarrow \langle i_{13}, i_{12}, i_{24}, i_{34} \rangle ; && \text{by R9} \\
 \langle i_{13}, i_{12}, i_{24}, i_{34} \rangle &\Rightarrow \langle i_{13}, i_{12}, i_{24}, i_{34}, i_{13} \rangle ; && \text{by R10} \\
 \langle i_{13}, i_{12}, i_{24}, i_{34}, i_{13} \rangle &\Rightarrow \langle i_{13}, i_{12}, i_{24}, i_{34}, i_{13}, i_{12} \rangle ; && \text{by R11} \\
 \langle i_{13}, i_{12}, i_{24}, i_{34}, i_{13}, i_{12} \rangle &\Rightarrow [i_{13}, i_{12}, i_{24}, i_{34}] && \text{by R12}
 \end{aligned}$$

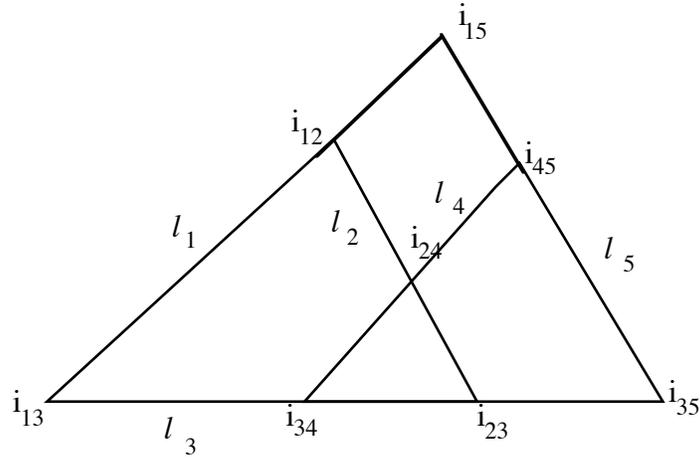


Fig. 6. A shape defined by symbolic representation (26)

The result is finding of a shape **Q1** composed of intersections and line segments represented by

$$[ i_{13}, i_{12}, i_{24}, i_{34} ] .$$

Its cardinality can be determined by function Ca:

$$Ca ( [ i_{13}, i_{12}, i_{24}, i_{34} ] ) = 4,$$

so its symbolic representation is given by (27).

$$\mathbf{Q1} = \{ 4; [ i_{13}, i_{12}, i_{24}, i_{34} ] \} \quad (27)$$

From (27), we can see that there are four intersections required, and **Q1** is composed of four infinite maximal lines, and the shape defined by (27) is a four-sided shape. The emergent shape **Q1** is shown in Figure 7.

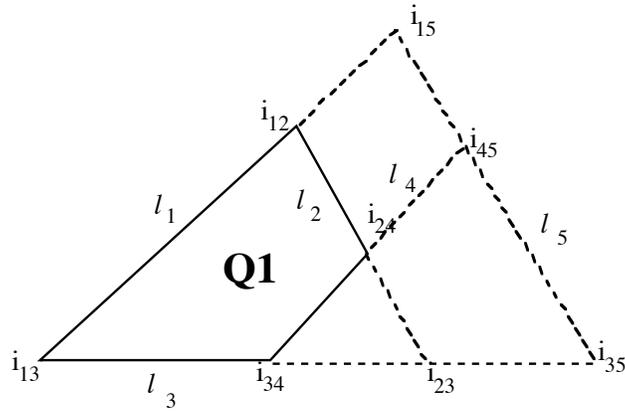


Fig. 7. An emergent quadrilateral, **Q1**, discovered by data-driven search.

Obviously, different emergent shapes can be discovered by applying the same rules R8 to R12 to the same unstructured representation (26). For example:

$$\begin{aligned}
 (i_{13}) &\Rightarrow \langle i_{13}, i_{15} \rangle; && \text{by R8} \\
 \langle i_{13}, i_{15} \rangle &\Rightarrow \langle i_{13}, i_{15}, i_{35} \rangle; && \text{by R9} \\
 \langle i_{13}, i_{15}, i_{35} \rangle &\Rightarrow \langle i_{13}, i_{15}, i_{35}, i_{13} \rangle; && \text{by R10} \\
 \langle i_{13}, i_{15}, i_{35}, i_{13} \rangle &\Rightarrow \langle i_{13}, i_{15}, i_{35}, i_{13}, i_{15} \rangle; && \text{by R11} \\
 \langle i_{13}, i_{15}, i_{35}, i_{13}, i_{15} \rangle &\Rightarrow [ i_{13}, i_{15}, i_{35} ] && \text{by R12}
 \end{aligned}$$

The result of the above data-driven search is discovery of the large triangle **T3**

$$\mathbf{T3} = \{ 3 ; [ i_{13}, i_{15}, i_{35} ] \} \quad (28)$$

which is shown in Figure 8.

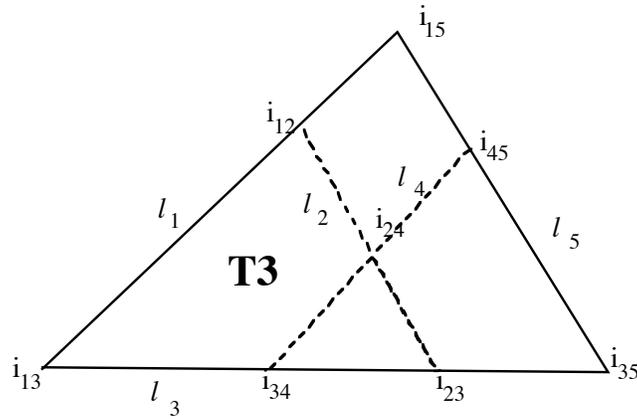


Fig. 8. An emergent triangle, **T3**, discovered by data-driven search.

The following search discovers the five-sided shape **F1** which is shown in Figure 9, and its symbolic representation is given by (29).

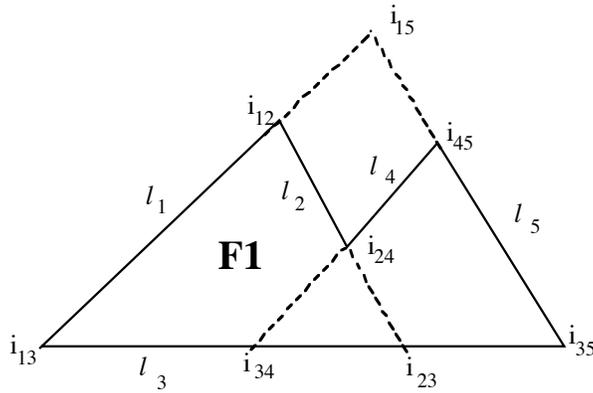


Fig. 9. An emergent five-sided shape, **F1**, discovered by data-driven search.

$$\begin{aligned}
 (i_{13}) & \Rightarrow \langle i_{13}, i_{12} \rangle; \text{by} && \text{R8} \\
 \langle i_{13}, i_{12} \rangle & \Rightarrow \langle i_{13}, i_{12}, i_{24} \rangle; && \text{by R9} \\
 \langle i_{13}, i_{12}, i_{24} \rangle & \Rightarrow \langle i_{13}, i_{12}, i_{24}, i_{45} \rangle; && \text{by R9} \\
 \langle i_{13}, i_{12}, i_{24}, i_{45} \rangle & \Rightarrow \langle i_{13}, i_{12}, i_{24}, i_{45}, i_{35} \rangle; && \text{by R9} \\
 \langle i_{13}, i_{12}, i_{24}, i_{45}, i_{35} \rangle & \Rightarrow \langle i_{13}, i_{12}, i_{24}, i_{45}, i_{35}, i_{13} \rangle; && \text{by R10} \\
 \langle i_{13}, i_{12}, i_{24}, i_{45}, i_{35}, i_{13} \rangle & \Rightarrow \langle i_{13}, i_{12}, i_{24}, i_{45}, i_{35}, i_{13}, i_{12} \rangle; && \text{by R11} \\
 \langle i_{13}, i_{12}, i_{24}, i_{45}, i_{35}, i_{13}, i_{12} \rangle & \Rightarrow [ i_{13}, i_{12}, i_{24}, i_{45}, i_{35} ] && \text{by R12}
 \end{aligned}$$

$$\mathbf{F1} = \{ 5 ; [ i_{13}, i_{12}, i_{24}, i_{45}, i_{35} ] \} \quad (29)$$

As a final example, consider the following application of the rules which discovers a six-sided emergent shape, **F2**, derived from only five infinite maximal lines. The shape is shown in Figure 10 and its symbolic representation is given by (30).

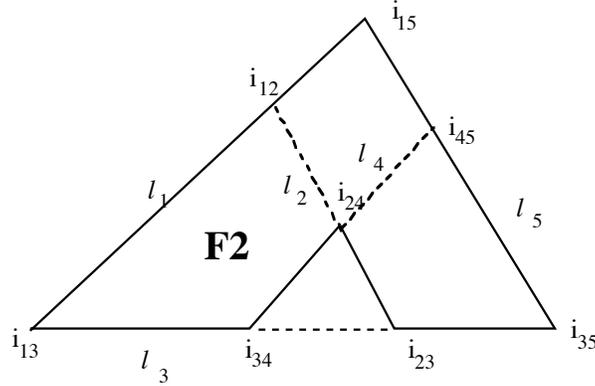


Fig. 10. An emergent six-sided shape, **F2**, discovered by data-driven search.

$$\begin{aligned}
 (i_{13}) & \Rightarrow \langle i_{13}, i_{15} \rangle; & \text{by R8} \\
 \langle i_{13}, i_{15} \rangle & \Rightarrow \langle i_{13}, i_{15}, i_{35} \rangle; & \text{by R9} \\
 \langle i_{13}, i_{15}, i_{35} \rangle & \Rightarrow \langle i_{13}, i_{15}, i_{35}, i_{23} \rangle; & \text{by R9} \\
 \langle i_{13}, i_{15}, i_{35}, i_{23} \rangle & \Rightarrow \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24} \rangle; & \text{by R9} \\
 \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24} \rangle & \Rightarrow \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34} \rangle; & \text{by R9} \\
 \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34} \rangle & \Rightarrow \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34}, i_{13} \rangle; & \text{by R10} \\
 \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34}, i_{13} \rangle & \Rightarrow \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34}, i_{13}, i_{15} \rangle & \text{by R11} \\
 \langle i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34}, i_{13}, i_{15} \rangle & \Rightarrow [ i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34} ] & \text{by R12}
 \end{aligned}$$

$$\mathbf{F2} = \{ 5 ; [ i_{13}, i_{15}, i_{35}, i_{23}, i_{24}, i_{34} ] \} \quad (30)$$

## 6 Conclusion

This paper has developed a model for shape emergence based on shape hiding using a representation of shapes founded on infinite maximal lines. It has shown how a symbolic model of shapes can be used as the basis of discovering emergent shapes using a data-driven symbolic model. The emergent shapes discovered need not have boundaries which embed into the boundaries of the primary shape. It is possible to discover ‘phantom’ shapes.

The ability to discover emergent shapes readily offers opportunities to develop design-oriented graphics systems which may be more amenable to augment designers during the conceptual stage of design. It has ramifications for collaborative design where two designers share the same (computational) workspace synchronously (Maher and Saad, 1992) but one 'sees' different shapes to those drawn by the other designer. The two schemas, the original and the emergent, could co-exist so that both designers have the same image in front of them but 'see' different things in those images. It becomes possible for designers to have different co-existing functions for the same design with computational support for each of them.

## References

- Edmonds, E. and Soufi, B., 1992. "The computational modelling of emergent shapes in design." In J. S. Gero and F. Sudweeks, F. (Eds), *Preprints of the Second International Round-Table Conference on Computational Models of Creative Design*. Sydney: Department of Architectural and Design Science, University of Sydney, pp.173-190.
- Garrett, L., 1967. *Visual Design: A Problem-Solving Approach*. New York: Reinhold.
- Gero, J. S., 1992. "Shape emergence and symbolic reasoning using maximal lines." *Unpublished Notes*, Design Computing Unit, Department of Architectural and Design Science, University of Sydney.
- Gero, J. S. and Yan, M., 1992. "Shape emergence by symbolic reasoning." *Working Paper*, Sydney: Design Computing Unit, Department of Architectural and Design Science, University of Sydney.
- Gombrich, E. H., 1966. *Norm and Form: Studies in the Art of the Renaissance*. Oxford: Phaidon.
- Gottschaldt, K., 1926. "Über den Einfluss der Erfahrung auf die Wahrnehmung von Figuren." *Psychologische Forschung* No. 8, pp. 261-317.
- Granovskaya, R. M., Bereznaya, I. Y. and Grigorieva, A. N., 1987. *Perception of Forms and Forms of Perception*, Hillsdale, New Jersey: Lawrence Erlbaum.
- Krishnamurti, R., 1980. "The arithmetic of shapes." *Environment and Planning B* 7, pp. 463-484.
- Krishnamurti, R., 1981. "The construction of shapes", *Environment and Planning B* 8, pp. 5-40.
- Krishnamurti, R. and Earl, C. F., 1992. "Shape recognition in three dimensions." *Environment and Planning B: Planning and Design* 19, pp. 585-603.
- Maher, M. L. and Saad, M., 1992. "Computer support for synchronous collaborative design." *Conference on the Impact of Computers on Design*, Sydney: Key Centre of Design Quality, University of Sydney, pp.61-76.
- Marr, D., 1982. *Vision*, San Francisco: Freeman.
- Mitchell, W. J., 1992. "A computational view of design creativity." In J. S. Gero and M. L. Maher (Eds), *Modeling Creativity and Knowledge-Based Creative Design*,

- Hillsdale, New Jersey: Lawrence Erlbaum (to appear).
- Purcell, A. T. and Gero, J. S., 1991. "The effects of examples on the results of a design activity." In J. S. Gero (Ed.), *Artificial Intelligence in Design '91*. Oxford: Butterworth-Heinemann, pp.525-542.
- Reed, S. K., 1974. "Structural descriptions and the limitations of visual imagery." *Memory and Cognition* 2, No.2, pp. 329-336.
- Schön, D., 1983. *The Reflective Practitioner*, New York: Basic Books.
- Stiny, G., 1980. "Introduction to shape and shape grammars." *Environment and Planning B* 7, pp. 343-351.
- Stiny, G., 1981. "A note on the description of designs." *Environment and Planning B* 8, pp. 257-267.
- Stiny, G., 1986. "A new line on drafting systems." *Design Computing* 1, pp. 5-19.
- Stiny, G., 1990. "What is a design?" *Environment and Planning B: Planning and Design* 17, pp. 97-103.
- Tan, M., 1990. "Saying what it is by what it is like—describing shapes using line relationships." In M. McCullough, W. J. Mitchell, and P. Purcell (Eds), *The Electronic Design Studio*, Cambridge: MIT, pp.201-214.