1990

Hands-on exploration of recursive forms

Christopher Carlson
Carnegie Mellon University

Robert F. Woodbury

Carnegie Mellon University.Engineering Design Research Center.

Hands-On Exploration of Recursive Forms

by

Christopher Carlson, Robert F. Woodbury

48-19-90

Hands-On Exploration of Recursive Forms

Christopher Carlson, Robert Woodbury
Department of Architecture, Carnegie Mellon University
Pittsburgh, Pennsylvania

Abstract

We describe an extremely simple paradigm that we have used to explore the graphic qualities of *recursive forms*, forms developed by recursively replicating a motif. Our emphasis is on form *exploration*, as opposed to form *generation*, which has been the focus of related work in this area. In spite of the simplicity of the paradigm, an enormous variety of forms and phenomena can be explored with it, including spirals, branching structures, plane symmetries and tilings, "reptiles", iterated function systems, space-filling curves, "squigs", meanders, textures, phylotaxis and organic forms.

The components of this method are a representation for 2 1/2-dimensional recursive forms, an algorithm for drawing them, a "hands-on" method of performing affine transformations, and a guiding principle of interaction. The synthesis of these components yields a tool for form exploration that is not merely quantitatively, but qualitatively better than has been reported to date. This paradigm has been used effectively on personal computers, and yet it scales well to take advantage of more computing power when it is available.

CR Categories and Subject Descriptors

F.4.2 [**Mathematical Logic and Formal Languages**]: Grammars and Other Rewriting Systems - *Parallel rewriting systems*; I.3.5 [**Computer Graphics**]:

This work has been supported by the Engineering Design Research Center, an NSF Engineering Research Center, and by NSF grant MSM8717307.

Computational Geometry and Object Modeling - *Hierarchy and geometric transformations;* 1.3.6 **[Computer Graphics]:** Methodology and Techniques - *Interaction techniques;* J.5 **[Computer Applications]:** Arts and Humanities - *Arts, fine and performing;* K.3 [Computing Milieux]: Computers and Education; K.8 **[Computing Milieux]:** Personal Computing.

Additional Key Words and Phrases

computer imagery, affine transformation, fractal, symmetry, reptile, tree, spiral, tiling, recursive form, self-similarity

Introduction

This paper is founded on two premises: (1) that an exceedingly simple representation suffices to describe a broad spectrum of forms of interest in computer graphics, and (2) that these forms can be effectively explored with minimal computing resources. We discuss the rationale behind the design of a program that we have used to conduct such explorations and exhibit the results of several investigations.

Rule-based formalisms for the generation of form will be familiar to readers of the computer graphics literature. Many authors have described systems for creating complex images by the repeated action of rules on simple sets of data [Fou82, Kaw82, Smi84, Dem85, Opp86, Bar88, Vie89]. In contrast to these efforts, which have focused on image *generation,* the emphasis of our work is on image *exploration.* We combine a simple representation, an efficient generation algorithm, an intuitive mode of graphic interaction, and a guiding principle of interaction to yield a qualitatively better tool for exploring the graphic qualities of a fundamental class of rule-based forms than has been reported to date. This paradigm, implemented as the program *discoverForm,* has been used effectively on low-end Apple® Macintosh™ computers, not merely to draw forms, but to learn about them **and discover new ones.** The forms we present in this paper were all discovered in the course of explorations using Macintosh Plus and Macintosh II computers.

The paradigm we present distinguishes itself from similar work by its combination of generality with interactive implementation on low-power hardware. Most formalisms of similar intent are limited to a particular domain of forms, such as terrain models [Fou82] or branching forms [Kaw82, Smi84, Op86, Vie89]. Other formalisms encompass a variety of forms but require high-performance hardware for interactive response [Sti8O, Kri80, Kri81, Dem85, Bar88]. In spite of the simplicity of the *discoverForm* paradigm, an enormous variety of forms and phenomena can be explored with it, including spirals [Coo79], branching structures [Man83], plane symmetries and tilings [Ste81, Grii87], "reptiles" [Gar63], iterated function systems [Dem85, Bar88], space-filling curves, "squigs", and meanders [Man83], textures, phylotaxis [Coo79], and organic forms [Ste74].

Recursive forms

We are concerned with 2 1/2-dimensional *recursive forms,* forms that are developed by recursively copying a 2-dimensional *motif* according to a single replication rule. The resulting motif copies arc arranged in layers that arc conceptually ordered along a third dimension; the limited use of the third dimension gives rise to the term "2 1/2-dimensional". The replication rule specifies the spatial relationships of a fixed number of motif copies, called *clones,* to the original motif. Recursive application of a rule to a motif produces successive *generations* of motif copies, the first generation being the motif itself, the second generation the clones, the third generation the copies of the clones, and so fonh (see figure 1). A form is completely determined by a motif, a replication rule, and the recursive depth to which the rule is applied.†

In formal linguistic terms, the forms we describe are expressions of context-free parallel rewriting systems. They are akin to the *0L-systems* [Hop79] that Lindenmayer proposed to model developmental processes in biology, with the important difference that where *0L*-systems deal with the adjacency relationships of symbols in strings, recursive forms deal with the spatial relationships of motif copies in compositions.

The spatial relationships of the clones to the original in a replication rule give rise, through the recursive application of the rule to a motif, to the spatial structure of a form. These relationships are specified as plane transformations that map the motif onto the clones. We are concerned here with affine transformations (translation, scaling, rotation, reflection, shear, and strain), since they suffice to explore a wide range of basic structures and since they are easily specified by intuitive graphic interactions.

We represent a recursive form as an ordered display list of motif primitives and clones. Primitives are either lines, represented by their endpoints, or polygons, represented by vertex lists. We limit motif primitives to linear elements because the line-preserving property of affine transformations makes them easy to transform: the image of a line segment is the segment connecting the images of its endpoints. Other primitives, for example ellipses, could be handled equally efficiently, but linear primitives have proven entirely adequate for form explorations in practice. The graphic image of a motif is obtained by drawing the primitives in list order, thus primitives at the tail of the list are drawn on top of primitives at the head. Interspersed with the motif primitives are clones, represented by

† Some forms are depicted as a superposition of all generations, others as only the last generation. This distinction is clarified later on.

3x3 homogeneous transformations. A clone marks a place in the list at which a transformed copy of the motif is drawn.

To generate a form, the display list is traversed recursively and each motif primitive or clone is drawn as it is encountered. Informal pseudocode for the generation algorithm is shown in figure 2. The parameter dispiayTransf orm functions like a transform stack onto which clone transformations are pushed and popped; it is the identity when the generation routine is initially invoked. Motif primitives are transformed by the value of dispiayTransform before they are drawn. The flag allGans determines whether all generations of a form are drawn, or only the last generation. For forms such as trees and spirals, it is appropriate to draw a superposition of all of the generations of the form. For forms such as squigs and space-filling curves that involve recursive refinement of the parts of the form to smaller parts, only the last generation is of interest

Figure 3 shows the correspondence of a display list to the form it generates. The motif consists of several filled polygons that together depict orthogonal intersecting planes. Three octants of the motif are occupied by half-size clones. The spatial relationships of the clones to the motif give rise to a recursive partitioning of the cube; the order of polygons and clones in the display list ensures that the parts of the form nest properly to depict a three-dimensional form.

A sense of the variety of recursive forms is given by figure 4. Some of the depicted forms are chosen to demonstrate the relationship of recursive forms to the work of other researchers. Iterated function system transformations based on simple polygonal shapes [Bar88, pp 134-135] are easily obtained by interactively transforming clones of a polygon so that they approximately cover the original polygon. Substituting a point for the polygonal motif then yields an approximation to an iterated function system with equallyweighted transformations. Form (d) is such a form, obtained by self-tiling a randomly drawn polygon. Some 3-dimensional forms, such as Kawaguchi's biomorphs [Kaw82] and Reynold's "Monument to Recursion" [Rey81], can be explored by means of 2 1/2-dimensional mock-ups ((b), (e), and (p)). Many of the constructions in Mandelbrot's *The Fractal Geometry of Nature* [Man83] are recursive forms ((f) and (k)) and many of the line and plane symmetries [Ste81, Grii87] can be formulated as such (g). In spite of the strict regularity of the mechanism by which they are developed, recursive forms need not have a geometrically regular appearance. Smith has pointed out that deterministic mechanisms suffice to generate forms whose appearance is convincingly random [Smi84]. As fonp&> (d), (h), and (i) illustrate, his observation holds even in the limiting case of a single context -free replication rule.

The enormous variety of forms that recursive display lists can represent is a consequence of their geometric, rather than topological, basis. The representation does not prescribe a particular topological class, such as trees, to which forms must belong, and consequently, explorations of forms are not restricted by the boundaries between topological classes. To be sure, our perception supplies topological interpretations to some spatial configurations of parts; we perceive in figure 1 (b), for example, a branching structure rather than an assembly of unrelated parts. However, such topological interpretations are not embedded in the recursive display list representation. In fact, a fixed set of spatial relationships between the parts of a form may give rise to considerably different perceived topologies, depending on how its spatial structure is articulated by a motif. Recursive display lists provide a clean separation between the geometric structure of a form (determined by the transformations of the clones) and the motif through which the structure is expressed; they are *geometric* abstractions of structure, in contrast to *topological* abstractions, exemplified by Smith's graftals [Smi84].

One is tempted to call recursive forms "self-similar", but this is, strictly speaking, a misnomer. Mandelbrot uses the term "self-similar" to refer to point sets that are completely covered by non-overlapping transformed copies of themselves, admitting similarity transformations only ([Man83], pp 349-350). Our forms do not conform to his definition for.at least three reasons: (1) motif primitives are treated as atomic units rather than as point sets, (2) we admit non-similarity transformations, and (3) a form may not completely cover itself, leaving what Mandelbrot calls a "residue". The cumbersome term "self-affine with residue" is more precise, but even this description requires amending Mandelbrot's definition of "self-affine" to include shear transformations. We prefer to call these for simply "recursive", a term that captures their essence.

Affine transformations

To construct a form, one draws the parts of the motif, clones the motif, and affinely transforms the clones. Part of the effectiveness of the *discoverForm* paradigm derives from the immediacy of interaction between the user and the forms he is exploring.[3] An illusion of manipulating forms "hands-on" is provided by a model of interaction based on the fixed points of affine transformations.

Under a rule of parsimony, an affine transformation is uniquely determined by specifying the points that it fixes Gcaves unchanged) plus the image of one additional point. The rule of parsimony states that when more than one transformation maps these points as desired, we choose the simplest one. Transformations that fix no points are thus assumed to be pure translations, without, for example, rotation or scaling components. Similarly, transformations that fix exactly one point are assumed to be combinations of rotation and non-negative scaling about the fixed point, without an additional reflection component If

[3] We regretfully use the generic "he" throughout this paper without wishing to imply that the *discoverForm* user is necessary masculine.

an affine transformation fixes two distinct points, then it fixes the line passing through the points pointwise. If an affine transformation fixes three or more non-colinear points, it is the identity. Consequently, at most two fixed points are needed to specify a non-identity affine transformation (of the plane).

These properties map nicely onto a method of interactively effecting affine transformations in which *tacks* hold fixed points while an additional point is dragged to its image. To affinely transform an object, a user positions up to two tack icons on the screen image of a form. Then the user selects parts of the form, either motif primitives or clones, and drags a point on them to a destination point. As the point is dragged, the relationship of its current position to its initial position determines, together with the points fixed by the tacks, a unique affine transformation. The form is regenerated while the user drags so that it remains consistent with the transformation determined by the position of the cursor. With the generation level set to an appropriately low level, current generation personal computers can compute and redraw forms quickly enough to convey to the user the illusion that he is manipulating a physical model of a form.

The effects of the zero-, one-, and two-tack affine transformations on a square are shown in figure 5. Translations (a) are effected without tacks by dragging a point to its image. Rotation (b) and scaling (c) are effected by tacking a point and dragging about it with a cursor. As the user drags around the tack, the object rotates; as he drags toward and away from the tack, the object scales down and up. Shear (d) and strain (e) require two tacks, which fix the points on the axis passing through them. Shear is effected by dragging parallel to the tacked axis and strain by dragging perpendicular to it. Reflection (f) is a significant special case of strain. Because reflection about an axis and reflections and halfturns that interchange tacked points are often used in form explorations, the user is provided with menu items to perform these transformations directly.

Exploring recursive forms

Our work is motivated by our desire to explore the universe of recursive forms, a multi-dimensional conceptual space whose dimensions are calibrated by motif shape, number of clones, and clone transformations. To use a two-dimensional metaphor, we seek pinnacles of visual interest among the hills and valleys of the terrain of recursive forms. And we seek paths between these which will develop our understanding of the lay of the land. Our vehicle for exploration is a computer program, built upon the recursive display list representation, that implements hands-on affine transformations of motif primitives and clones. To clarify the nature of this task, we ask three questions: "What does the terrain look like?", "How are we able perceive and comprehend it?", and "How does our vehicle of exploration move about in it?".

With the benefit of hindsight, having explored a great deal of the terrain already, we can say that the landscape of recursive forms is rich in visual heights and depths, having ridges, peaks, plateaus, basins, and valleys in all directions. It is often precipitous. Small steps in some directions may plummet one from exhilarating heights to the depths of visual chaos (or inversely). As the number of clones increases, plateaus and ridges give way to more sparsely scattered peaks, since the likelihood that all of the components of a form will be in just the right relative positions to yield a visually meaningful result decreases with their number. However, the peaks of such rare coincidences are often correspondingly high.

Our means of moving about in the landscape and our perception of it are closely linked to the complexity-compounding effect of recursion, a phenomenon that has become known as "database amplification". Subtle changes to a form's motif or clones may cause uncorrespondingly large changes in the form that they generate. In terms of our vehicle of exploration, database amplification means that we move through the landscape extremely fast; small motions at the controls may send us rocketing through the visual countryside. Or, equivalently formulated, our speed effectively compresses the landscape into a small area, which has the effect of exaggerating the roughness of an already irregular terrain. This compression has a paradoxical effect on exploration. We are able to cover a lot of territory quickly, but the peaks and valleys fly by so fast that we are bound to miss a lot of scenery on the way.

A further consequence of database amplification is shown in figure 6, which illustrates how the recursive application of a replication rule to a simple motif may give rise to unexpectedly intricate forms. In all but the simplest instances, the human mind is not equipped to predict the outcome of such recursion. It is all but impossible to predict from the motif and clones of figure 6 (a) that they will give rise to the pattern in (b), the boundary in (c), or the mottled shading in (d), even though the mechanism by which those forms are developed is deterministic and immediately comprehensible. This shortcoming cannot be ascribed to a mere lack of experience with recursive forms; our several years of working with *discover Form* and its predecessors have developed our intuition of where to look for interesting forms, but not our ability to predict the appearance of a form from its motif and replication rule. Rather, the prediction problem seems to transcend the fundamental computing power of the human mind, which has not evolved to deal with deep recursion. An interesting consequence of this is that, as we noted above, forms with a random appearance may be generated by deterministic mechanisms. If the forms are sufficiently deeply recursive, we are unable to detect the underlying regularity.

In terms of the terrain metaphor, we cannot determine our height in the landscape (corresponding to the visual interest of a form) from our position in it (indexed by the generation level, motif shape, and spatial relationships of the clones). We can evaluate a

form by looking at it, but not by performing thought experiments. Worse yet, we often cannot extrapolate the visual interest of a form to its neighbors due to the roughness of the terrain. A visually dissonant form may be brought into splendid resonance by a slight adjustment of a clone, and yet the direction of the adjustment is not suggested by the dissonant form. In effect, we arc extremely nearsighted explorers of the landscape, able to perceive and evaluate only those forms that we manage to steer into direct view on the computer screen.

Exploration is venturing forth without a definite goal in the hope of discovering interesting phenomena. The explorer seizes on serendipitous events that suggest new paths of exploration. The success of an exploration depends a great deal on such opportunism. Imagine the difficulty of exploring a rough continent if your visibility was an arm's length and at each step of the way you had to say how far and in what direction you intended to go, and then go there *blindfolded.* You would miss interesting sights along the way and, more importantly, opportunities that you could not have anticipated. As myopic explorers of recursive forms, the productivity of our explorations depends less on where we intend to explore and more on the incidental forms we see along the way, since, given our shortsightedness with respect to recursive forms, the only opportunities for exploration we are likely to recognize are the ones that appear on the screen before us. In order to present these opportunities to the *discoverForm* user, we follow a simple principle of interaction: all changes to motifs and clones are immediately reflected in the forms they generate.

Adherence to this principle implies that as one draws new parts of a motif, the parts are drawn simultaneously in every copy of the motif; as parts of the motif are transformed, they are simultaneously transformed in every copy of the motif; and as clones are transformed, every motif copy whose compound transformation depends on the clone's transformation moves accordingly. Ruled out are interactions whose effect on the form is

not apparent until they are completed. It is not permissible, for example, to refrain from updating a form until the user has finished drawing a motif part. The program thus functions as a kind of super-kaleidoscope whose pattern structure (determined by the spatial relationships of the clones) can be changed as well as the scene viewed through it (the motif). In our experience with *discoverForm*, the largest source of ideas for new explorations and of insight into the connections between forms have been the incidental, unexpected forms we have encountered while performing motif and clone transformations.

Interactive response is achieved with even minimal computing power by adjusting the generation level of a form downward. Because of the approximate self-similarity of recursive forms, most features of interest in higher-level forms are also apparent in lower generations of the forms. We have found it useful to explore forms at low generation levels, increasing the level to flesh out forms that prove interesting. This strategy permits a great deal of terrain to be investigated effectively with relatively low-power hardware. As more powerful hardware becomes available, higher-order features of recursive forms, such as the boundary and shading effects illustrated in figure 6, can be explored with the same interactive paradigm.

Some explorations

We now turn to some sample explorations. As figure 7 illustrates, the simplest manipulations of recursive forms quickly suggest paths of exploration leading to diverse structures. At the center of the figure is a binary tree, a form generated by recursively attaching two clone branches to a trunk motif. The forms that encircle the tree are derived from the basic tree by rotating the branch fork about its attachment point to the trunk. In one cycle of rotation, we encounter explorations leading to the golden ratio and a generalization of it, diverse spiral forms, a plane symmetry, space-filling canal structures, and a dragon curve.

Starting with **the** symmetric tree (a), a rotation of 45° clockwise yields a tree whose branches enclose squares (b). Scaling down the branches just slightly brings the overlapping and jumbled squares into coincidence (c) so that each square has exactly two adjacent squares above it and to the right. Anyone familiar with the standard construction of the golden rectangle [Ghy77, Ch. II] will recognize that the lengths of the sides of the squares are related by the golden ratio, O. Form (f) is a three-dimensional expression of the same structure in which the squares are replaced by cubes. Scaling the branches further yields an infinite series of visual resonances in which squares have an integer numbero£ adjacent squares (d). The limiting case (e) is reached when the ratio of sides of successive squares is 0.5. Taking the side length of the largest square as 1, the ratios at which the resonances occur can be read off of the forms as the positive real solutions of the equations in the sequence $1 = r$, $1 = r + r^2$, $1 = r + r^2 + r^3, \ldots$, namely, 1,0, 0.5437,____The golden ratio is seen to be the first non-trivial ratio of this series.

As the branches are rotated further (g-n), the embeddedness of spirals in trees and of trees in spirals becomes evident Forms (h), (j), and (o) are derived from forms (g), (i), and (n) by scaling the clones to bring higher-order branches of the tree into coincidence and then using the tree as a template to construct the implied spiral structure. These constructions make dear that the visually distinct forms (h), (j), and (o) are simple parametric variations of a common theme. Form (j), nested squares, suggests two avenues of further explorations, one of changing the nesting relationship of the squares (m) and the other of nesting higher-order regular polygons in the same fashion (k, 1).

The stepwise transformation of one fonn to another by manipulating the motif and clones is a powerful technique for discovering connections between diverse forms. Although forms

(f) and (h) are visually quite disparate, they share similar structures, a commonality which is made evident by tracing the path of forms (fHc)-(b)-(g)-(h).

Continuing the cycle of rotation, at the halfway point (p) the tree is a compact structure whose branches lie on a regular rectilinear grid. In fact, the last-generation branches of this form obey the p4mm symmetry of the plane [Ste81, Ch. 34], as do the last-generation branches of form (q). By displaying only the last generation of the latter form (r) and replacing the trunk with an asymmetric arrow motif (s), we can use the form as a kaleidoscope through which to explore the visual richness of the p4mm symmetry greup. Rotating and translating the motif within this structure gives rise to patterns (t) and (u).

Shortly before the branch rotation comes full-cycle, the tree assumes the shape of a rectangle (v), the highest-generation branches evenly spaced within it. The white spaces between the branch groups suggest trying to position the motif so that the branches of the tree are non-intersecting. Rotating and translating the trunk motif reveals that this is indeed possible, and yields a canal structure that evenly waters the rectangle (w). Further exploration of motif positions reveals a related canal structure (x) in which the motif is centrally placed and the clone branches radiate from either end The symmetry of the clones in this form suggests exploring other angular relationships of the clones to the motif. One of the resulting forms (y) has an interesting boundary, which the high-order branches of the tree appear to fill evenly. By displaying only the last generation of the form and increasing the generation level (z), we are able to confirm that the form is the twindragon described by Mandelbrot [Man83, pp 66-67].

Articulating the structure of a form by varying its motif is powerful technique of exploration whose use is further illustrated in figure 8, which depicts variations on a theme by Hilbert. The Hilbert curve is developed by recursively replacing an inverted "IT* motif

by copies of the motif centered at the vertices of the "U" (a). The copies are joined with connecting segments to form a connected path (b). Ignoring the connecting segments, Hilbert's form is recursive. We replace the inverted "U" with an asymmetric motif (c) to reveal the form's structure in terms of the orientations of the motif copies. It is this structure that we are interested in exploring.

Using the form as a kaleidoscope as we have done before, we begin to explore the form's structure by translating vertical line motifs. This quickly reveals a structure of regions that are delimited by connected paths through the form (d). Rotating the vertical line motif by 90° yields a structure of branching paths that are orthogonal to the region boundaries. The structure of this form is elegantly expressed by a parallelogram motif ((e), shown one generation level lower than form (d)). Tracing three sides of the parallelogram yields a slanted "U" motif, which when appropriately translated gives rise to an illusory form (0 that appears to be 3-dimensional, but is not realizable. We did not anticipate, and indeed could not have anticipated, that any such structures as forms (d), (e), and (f) lurk in the Hilbert curve. Yet we were quickly and naturally led to their discovery by the visual coincidences we observed when we more or less randomly transformed motifs while viewing them through the structure of the curve.

In a final example, we recount the exploration that led to the discovery of the space-filling curve shown in figure 6. The curve is a "squig", a term introduced by Mandelbrot [Man83, Ch. 24] to describe paths that are developed by recursively replacing the linear segments of an initial path by scaled copies of the path. Some interesting squigs are based on paths through the points of regular grids; we were exploring these when we encountered the form.

Our exploration is diagrammed in figure 9. We began with a three-segment motif based on a square grid (a). Cloning the motif and positioning the clones on the path segments yielded an initial squig (b). Each of the three path segments can be replaced by a clone in four distinct ways, for a total of $4^3 = 64$ distinct squigs. We explored these more or less randomly, on the alert for interesting patterns, by tacking the endpoints of clones and transforming them with the reflection and halfturn operations described earlier. One configuration yielded a space-filling curve (c) with an unusual boundary. We investigated the structure of this curve by varying the motif as in the Hilbert curve exploration above. When we mirrored the last two segments of the path about the axis passing through their endpoints, the awkward curve (c) was transformed into the elegant form (d). From our previous experience with squigs, we had conjectured that every space-filling curve fills an area that is self-tiling. To determine if the conjecture held in the present case, we traced an approximation to the boundary of (d). This indeed revealed a reptile and its self-tiling scheme (e).

Conclusion

We have demonstrated that a diverse class of visually interesting forms can be profitably explored with minimal computing resources. By using a 2 1/2-dimensional, geometrically-based representation, we are able to describe a wide variety of forms, and more importantly, to discover connections between diverse forms by interactively transforming one to another. By putting the user in direct contact with the forms he is exploring and by providing immediate response to his manipulations, we foster the discovery of unanticipated avenues of inquiry, an important aspect of creative exploration.

The success of this paradigm rests on the recursiveness of recursive forms. Small changes to a motif or replication rule often cause uncorrespondingly large changes to the form they generate, due to the complexity-compounding effect of recursion. This enables the explorer to cover **a** large **and** diverse territory with relatively simple manipulations. Recursion also means that features of interest in higher-generation forms are often visible in lower-generations of the same forms. Hence, most explorations can be carried out at low generation levels where interactive response is almost always possible.

We would like to suggest three extensions to the *discoverForm* paradigm that would open up even greater domains of form to exploration. First, generalization of the representation and generation algorithm to three dimensions is straightforward. We chose not to investigate three-dimensional forms in order to avoid the difficult problem of designing intuitive manipulations of three-dimensional objects on a two-dimensional medium. However, our experience with *discoverForm* has led us to believe that the additional effort to design a 3-dimensional extension would be justified by the results of the explorations that could be conducted with it

Second, we were surprised that in spite of the simplicity of the recursive display list representation, it suffices to explore such a broad range of forms. It is the simplest representation of its sort, admitting only one motif and one context-free replication rule. Extensions to multiple motifs and multiple rules would permit equally efficient explorations of more complex systems such as Penrose tilings [Grii87, §10.3]. The admission of context-sensitive rules would permit modeling cellular automata, although the generation algorithm for context-sensitive forms might be considerably less efficient than the algorithm that we have presented here.

Finally, although our goal in designing *discoverForm* was to foster creative search, we. have not included in it explicit support for keeping track of the course of an exploration. The shape of an exploration and the connections between forms that it reveals is at least as interesting as the forms themselves (see, for example, figure 7). The *discoverForm* user must construct such a map in his head or on paper, although the computer could semi-automatically chart the paths the user explores. Mechanisms for maintaining a search map, attaching annotations to it, and presenting it to the user as an aid to planning future explorations and resuming old ones could contribute a great deal to the productivity of his explorations.

References

[Bar88] Barnsley, MP_M Jacquin, A., Malassenet, R, Reuter, L., Sloan, A.D., 1988, "Harnessing Chaos for Image Synthesis" *Computer Graphics* 22:4 131-140

[Blo85] Bloomenthal, J., 1985, "Modeling the Mighty Maple" *Computer Graphics* 19:3 305-311

[Coo79] Cook, T.A., 1979, *The Curves of Life* (Dover, New York)

[Dem85] Demko, S., Hodges, L., Naylor, B., 1985, "Construction of Fractal Objects with Iterated Function Systems" *Computer Graphics* 19:3 271-278

[Fou82] Fournier, A., Fussell, D., and Carpenter, L., 1982, "Computer rendering of stochastic models" *Communications of the ACM* 25:6 371-384

[Gai63] Gardiner, M., 1963, "Mathematical Games" *Scientific American* 208:5 154-164

[Ghy77] Ghyka, M., *The Geometry of Art and Life* (Dover, New York)

[Grii87] Griinbaum, B., and Shephard, G.C., 1987, *Tilings and Patterns* (W.H. Freeman, New York)

[Kaw82] Kawaguchi, Y., 1982, "A morphological study of the form of nature*[9] *Computer Graphics* 16:3 223-232

[Kri80] Krishnamurti, R., 1980,[4]The arithmetic of shapes" *Environment and Planning B* 7463-484

[Kris81] Krishnamurti, R., 1981, "The construction of shapes" *Environment and Planning B*7463-484

[Hop79] Hopcroft, JJE. and Ullman, J.D., 1979, *Introduction to Automata Theory, Languages\ and Computation* (Addison- Wesley, Reading, Massachusetts)

[Man83] Mandelbrot, B., 1983, *The Fractal Geometry of Nature* (W.H. Freeman and Co., San Fransisco)

[Opp86] Oppenheimer, P.E., 1986, "Real Time Design and Animation of Fractal Plants and Trees" *Computer Graphics* 20:4 55-64

[Rey81] Reynolds, C.W., 1981, *Computer Graphics* 15:3 (cover graphic)

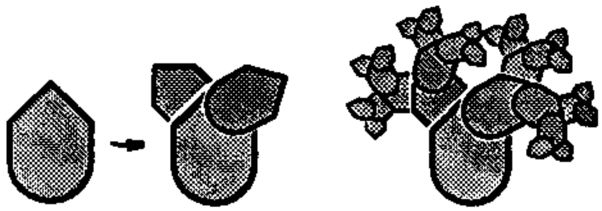
[Smi84] Smith, A.R., 1984, "Plants, fractals, and formal languages" *Comuter Graphics* 18:3 1-10

[Ste74] Stevens, P.S., 1974, *Patterns in Nature* (Little, Brown and Co., Boston)

[Ste81] Stevens, P.S., 1981, *Handbook of Regular Patterns* (MIT Press, Cambridge, Mass)

[Sti80] Stiny, G., 1980, "Introduction to shape and shape grammars" *Environment and Planning B 1* 343-351

[Vie89] Viennot, X.G., Eyrolles, G., Janey, N. and Arqu6s, D., 1989, "Combinatorial Analysis of Ramified Patterns and Computer Imagery of Trees" *Computer Graphics* 23:3 31-40



(a) (b)

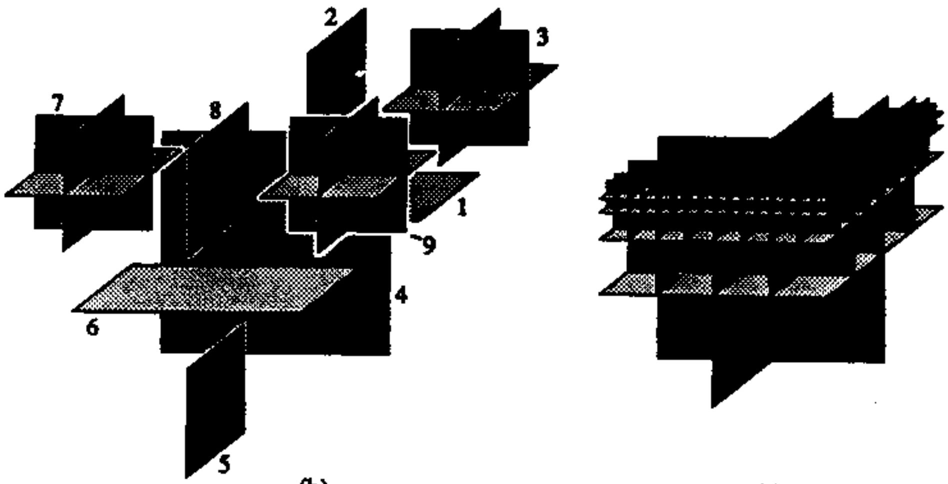
Figure 1. A recursive form: (a) replication rule and (b) fifth generation form.

```
generate(displayList, displayTransform, depth, allGens):
   if depth ≥ 1
      for item in displayList
         if primitive(item)
            if allGens or depth=1
               draw(item, displayTransform)
         else /* item is a clone transformation */
            generate(displayList,
                          concat(displayTransform, item),
                          depth-1, allGens)
```

Figure 2. The generation algorithm.

```
displayList :- {
   polygon 1,
   polygon 2,
   don* 3,
   polygon 4,
   polygon 5,
   polygon 6,
   don« 7,
   polygon 8,
   done 9
}
```

(a)



(b) (c)

Figure 3. A recursive display list and the form that it generates, (a) Display list, (b) motifpfimitives and clones (exploded view), and (c) fourth generation form.

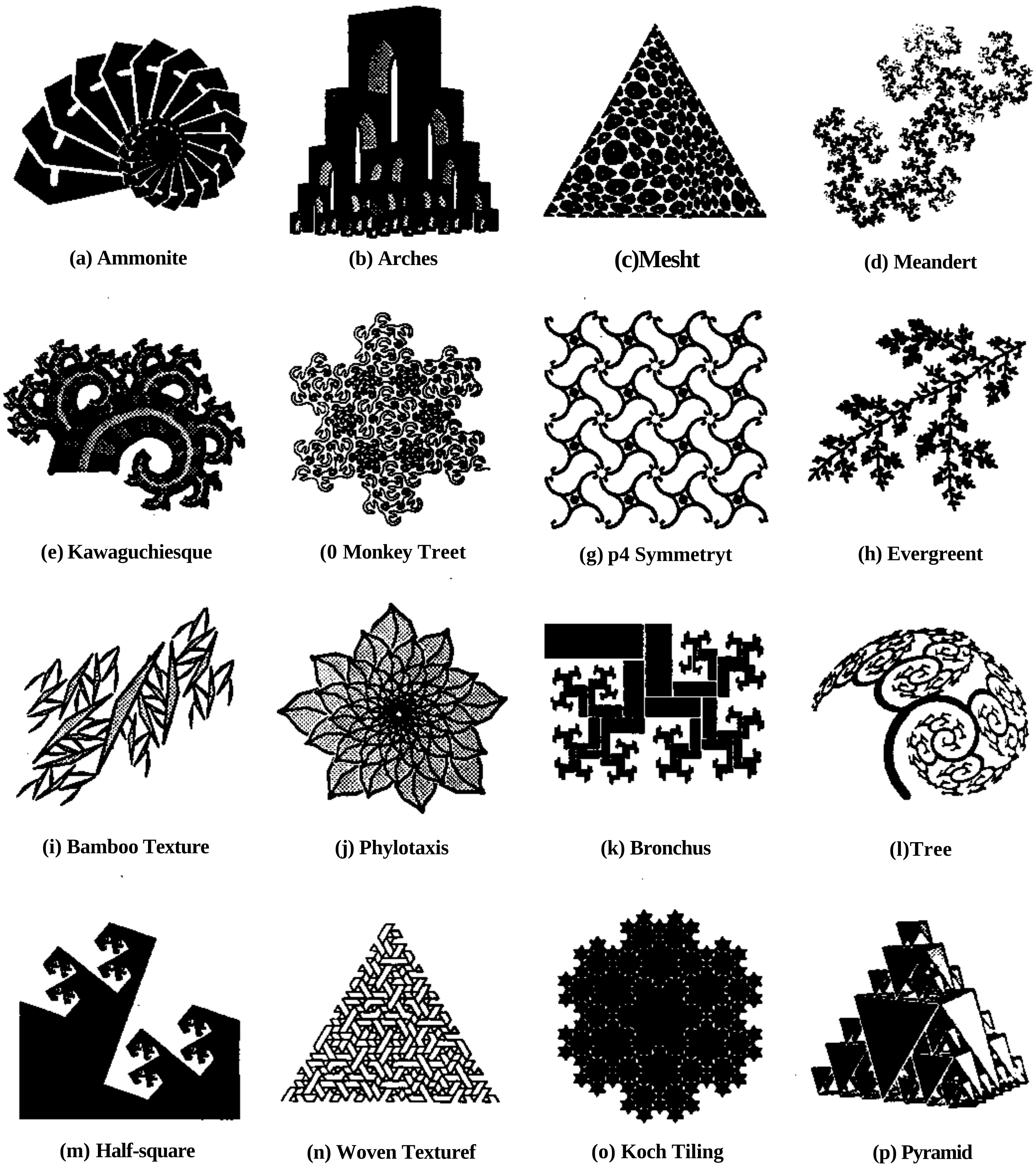


Figure 4. Some recursive forms. Forms marked with a dagger (f) are shown last generation only.

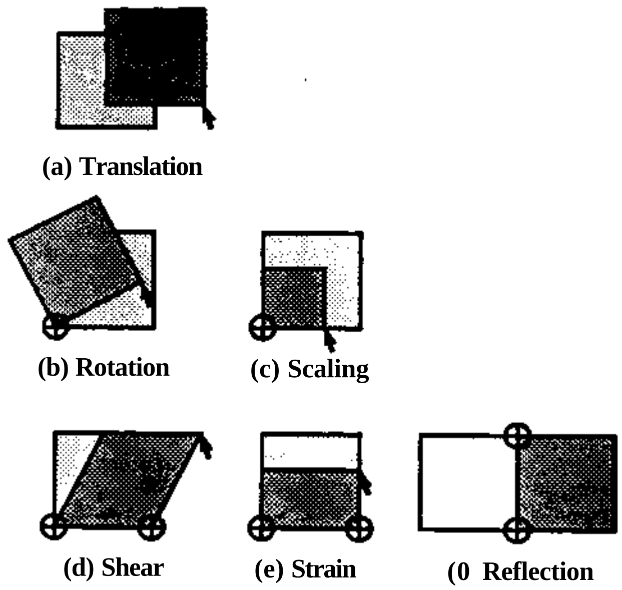
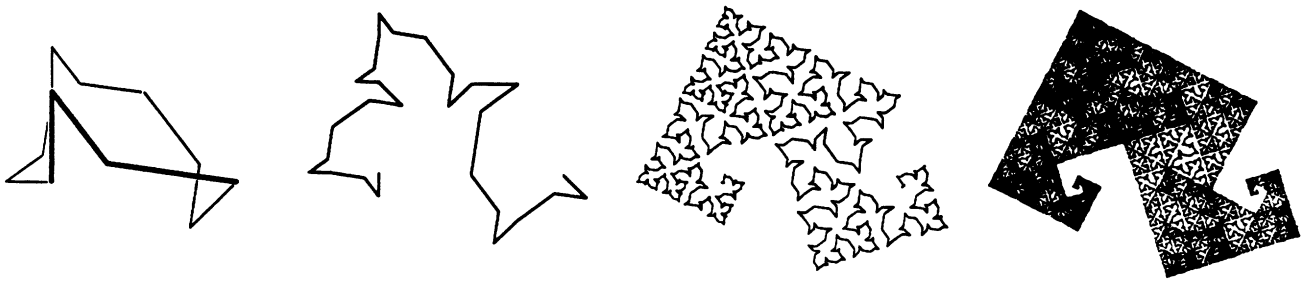


Figure 5. Performing affine transformations of the plane by tacking fixed points and dragging.



(a) Motif and clones (b) Third generation (c) Sixth generation (d) Ninth generation

Figure 6. Recursion compounds complexity. (a) Motif and clones, (b) resulting pattern, (c) emergence of a boundary, (d) emergence of shading effects.

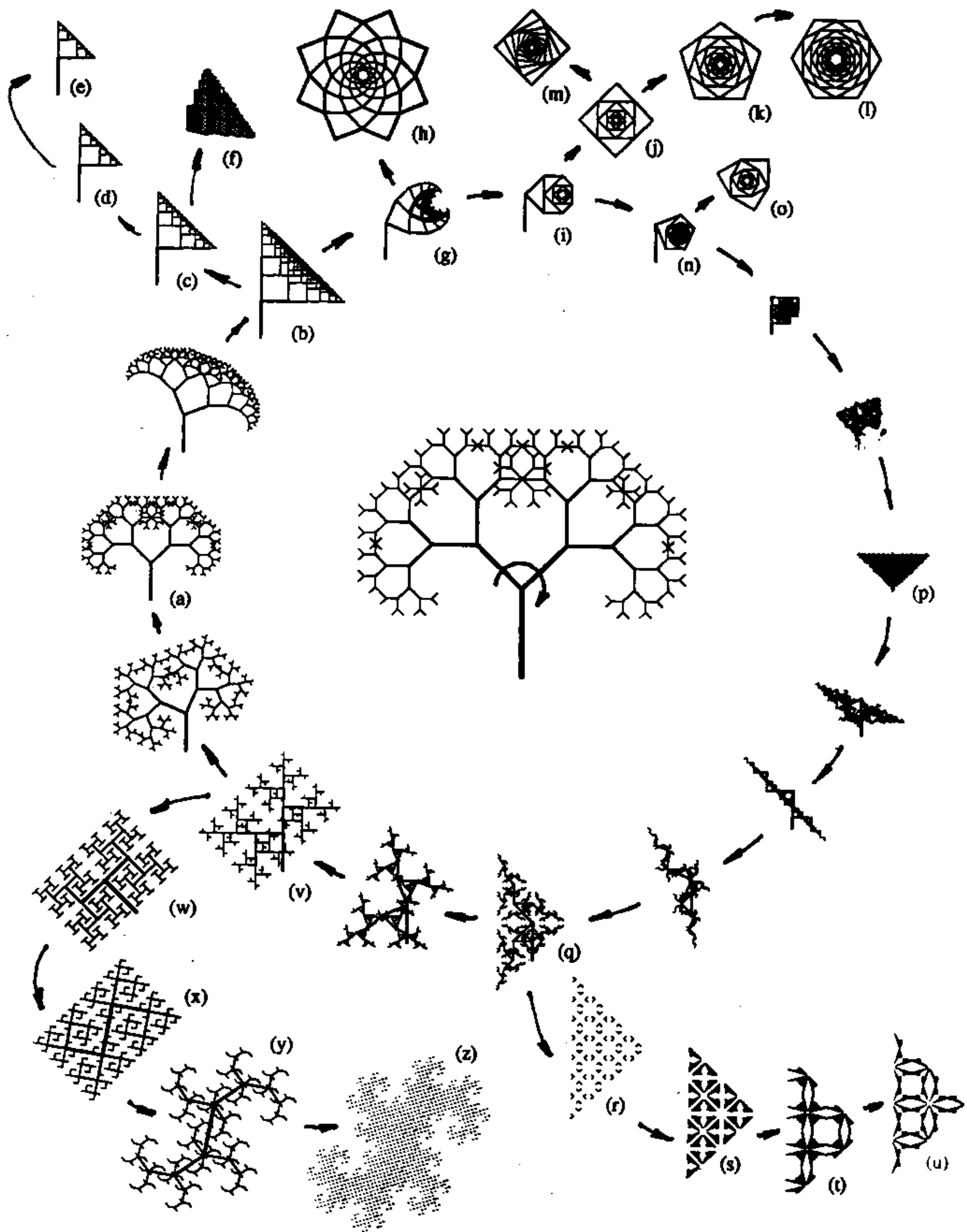


Figure 7. Exploring the binary tree.

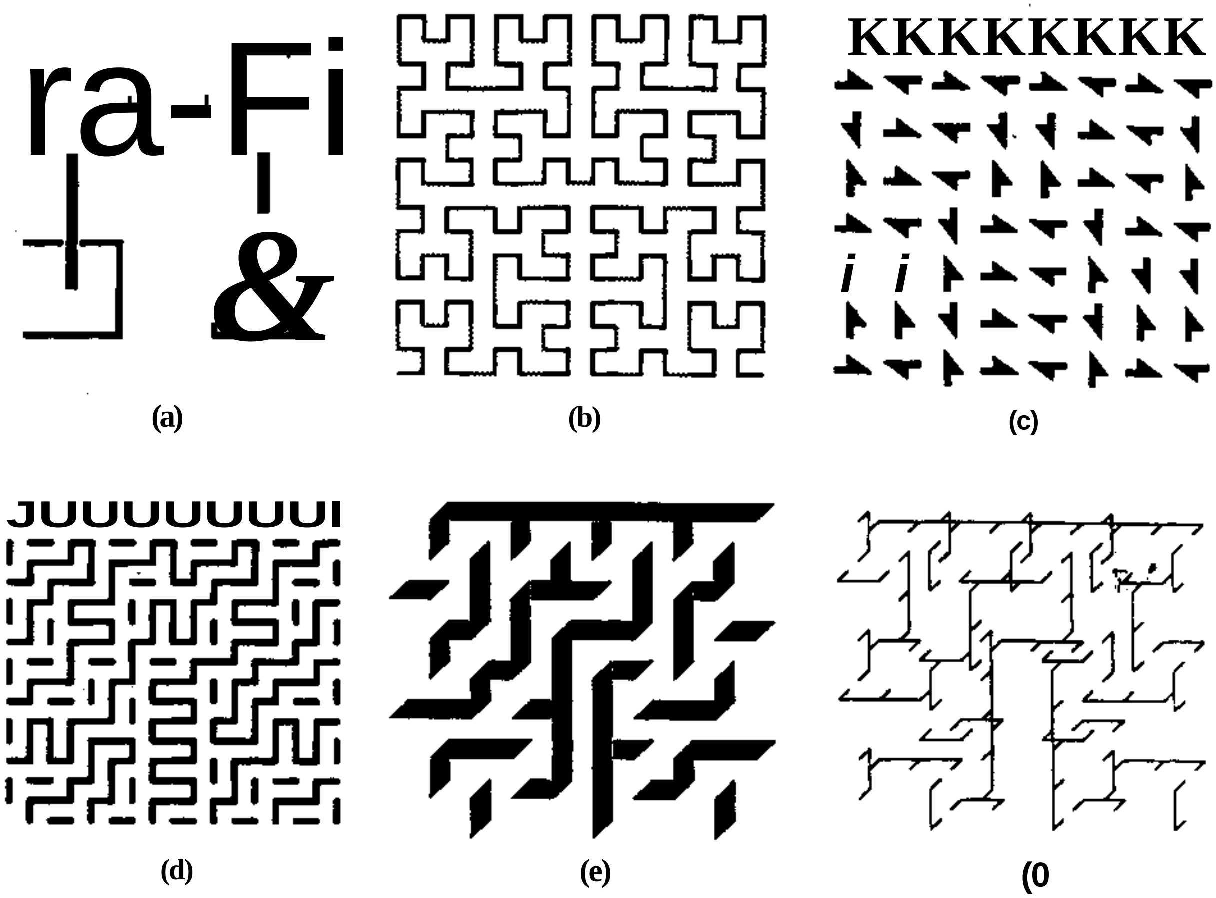


Figure 8. Variations on a theme by Hilbert: articulating the structure of a form by varying its motif.

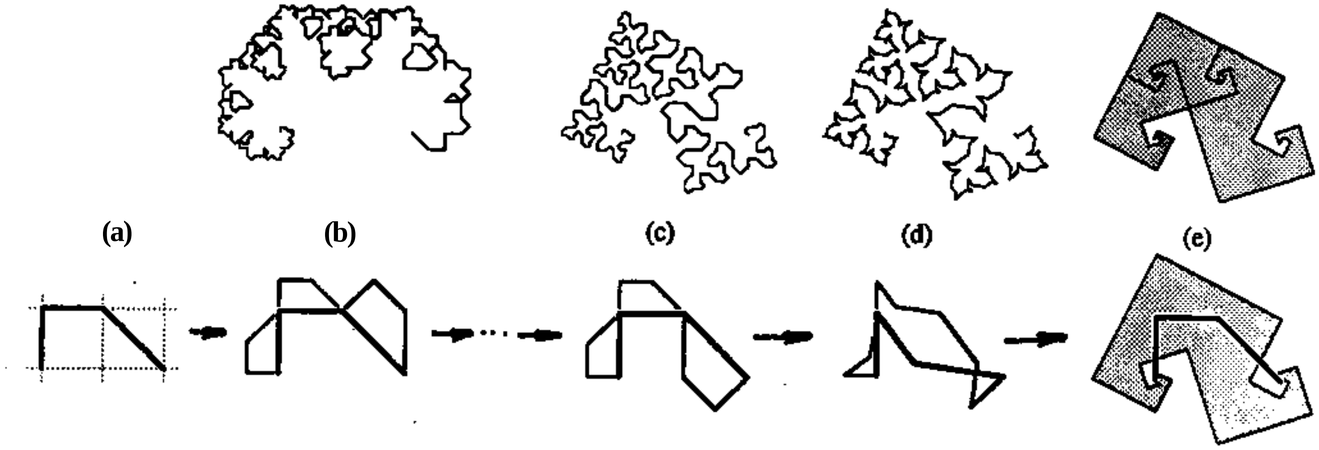


Figure 9. Discovery of a space-filling curve and reptile.