

GENERATING FRACTALS OF REGULAR FORM BY PICTURE LANGUAGES

Vasile PREJMEREAN*, Simona MOTOGNA*, Vasile CIOBAN*

Dedicated to Professor Emil Muntean on his 60th anniversary

Received January 31, 1994

AMS Subject Classification 68U05, 68U10, 68Q25, 68T10

Rezumat. Generarea fractalilor de formă regulată prin limbaje picturale În această lucrare este prezentată o modalitate de generare a fractalilor de formă regulată, utilizând șiruri de comenzi pentru desenarea acestora. Aceste șiruri de comenzi sunt generate prin funcții care permit dezvoltarea, prelucrarea și recunoașterea acestor variațiuni geometrice numite fractali.

The fractal textures are more and more often used in computer graphics since they can model properly 3D-figures and natural forms and they have the advantage of representing the models on a plane surface. Fractals can have regular forms, based on repetition of a motif (primary detail) or randomizing forms, which are defined probabilistic. They can be built starting from curves, surfaces or figures and they are defined either by a function or by a construction rule [2].

Regular form fractals can be generated using a language of commands for drawings (images) [1]. Given a set of graphical primitives (corresponding to a set of drawing commands), from which one or more starting primitives (primary detail) are chosen, we will apply a transformation to this initial set, then to the obtained set of primitives we will apply again the same transformation and so on, as many times as we want. Finally, the resulting set

* "Babeș-Bolyai" University, Faculty of Mathematics and Computer Science, 3400 Cluj-Napoca, Romania

of primitives represents the desired fractal, respectively if the result is a string of commands then the fractal will be obtained executing these commands

Let's consider the set $C = \{c_1, c_2, \dots, c_p\}$, containing the graphical primitives which form a fractal family. These primitives can be drawn by corresponding graphical commands. So, we can achieve a drawing executing a string of such commands

Let $S = \{x_1 x_2 \dots x_m \mid m > 0, x_1, x_2, \dots, x_m \in C\}$ be the set of command strings obtained by concatenation of the elements from the set C . A family of variable fractals of regular type is a set of fractals which have been obtained starting from a primary detail $D \in S$, which is developed step by step according to a transformation rule $f: C \rightarrow S$, rule which is applied to every graphical primitive (command). This transformation gives the development rule for a primitive and is specific to every fractals family

The development function which allows a fractal to be transformed entirely with an iteration is

$$t: S \rightarrow S, t(x_1 x_2 \dots x_m) = \begin{cases} f(x_1), & \text{if } m=1 \\ f(x_1) f(x_2) \dots f(x_m), & \text{if } m>1 \end{cases}$$

(where uv represents the concatenation of u and v)

The development of a fractal after n iterations (n "years") is described by the function

$$t^n: S \rightarrow S, t^n(X) = \begin{cases} t(X), & \text{if } n=1 \\ t^{n-1}(t(X)), & \text{if } n>1 \end{cases}$$

We can say that $t^n(D)$ returns the commands string representing the fractal in the n -th step of the development. This observation makes us think that we can define a fractals family specifying the primitives set C , the primary detail D , and the development rule f

$$F_n = (C, D, f), n=1, 2,$$

The definition of the transformation rule f will use the following function

GENERATING FRACTALS OF REGULAR FORM

$$\text{Next } C \rightarrow C, \text{ Next}(c_i) = \begin{cases} c_{i+1}, & \text{if } i > p \\ c_1, & \text{if } i = p \end{cases} \quad (C = \{c_1, c_2, \dots, c_p\})$$

Since this function is bijective, we can construct the reverse function, which will simplify the definition of the function f

$$\text{Next}^{-1} C \rightarrow C, \text{ Next}^{-1}(c_i) = \begin{cases} c_{i-1}, & \text{if } i > 1 \\ c_p, & \text{if } i = 1 \end{cases}$$

Example 1 Let's consider the following set $C = \{r, u, l, d\}$ where

$$(r, u, l, d) \text{ is } (\rightarrow, \uparrow, \leftarrow, \downarrow) \quad [3]$$

and $f(x) = x \text{ Next}(x) x \text{ Next}^{-1}(x) x, \forall x \in C$

(x concatenated with $\text{Next}(x)$, concatenated with x ,) and $D=r$

Then $t^1(D), t^2(D), t^3(D), \dots$ represent the following family

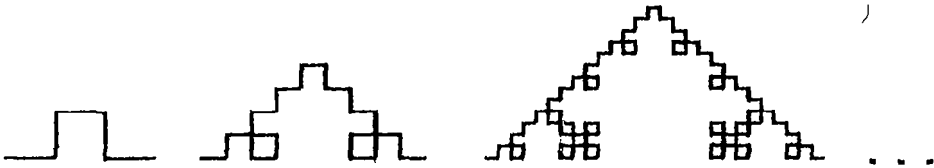


Figure 1

Either in constructing or in recognizing a fractals family defined in the way we described, the following property of the development function is very useful

$$t^n(XY) = t^n(X) t^n(Y)$$

The above property can be proved by complete induction as follows

$$\begin{aligned} \text{for } n=1 \quad t^1(XY) &= t(XY) = t(x_1x_2 \dots x_{m_1}y_1y_2 \dots y_{m_2}) = \\ &= f(x_1)f(x_2) \dots f(x_{m_1}) f(y_1)f(y_2) \dots f(y_{m_2}) = \\ &= t(x_1x_2 \dots x_{m_1}) t(y_1y_2 \dots y_{m_2}) = t(X)t(Y), \end{aligned}$$

assuming that $t^{n-1}(XY) = t^{n-1}(X) t^{n-1}(Y)$

then $t^n(XY) = t^{n-1}(t(XY)) = t^{n-1}(t(X)t(Y)) = t^{n-1}(t(X))t^{n-1}(t(Y)) =$
 $= t^n(X)t^n(Y)$

This property allows us to draw a fractal successively on sections and also to analyze a fractal reducing it at its subfractals which compounds it

We consider now another two examples

Example 2 Let $C=\{r,e,u,f,l,g,d,h\}$ be the set with

$$(r,e,u,f,l,g,d,h) = (\rightarrow, \nearrow, \uparrow, \nwarrow, \leftarrow, \searrow, \downarrow, \swarrow) ,$$

and $f(x) = x \text{ Next}(x) \text{ Next}^{-1}(x) x$, and $D=\text{urdl}$

Then $t^1(D)$, $t^2(D)$, $t^3(D)$, represent the following family

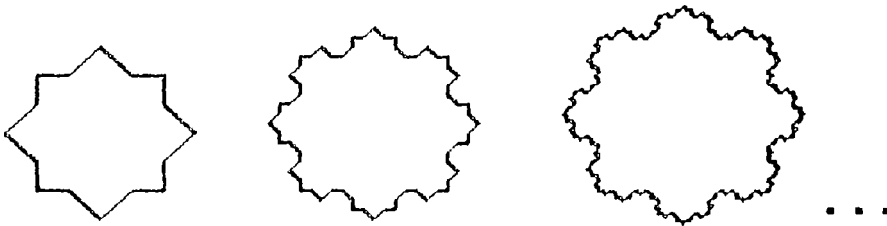


Figure 2

Example 3 Let $C=\{a,b,c\}$ be the set with

$$(a,b,c) = (\nearrow, \nwarrow, \leftarrow) ,$$

and $f(x) = x \text{ Next}(x) \text{ Next}^{-1}(x) x x$, and $D=\text{abc}$

Then $t^1(D)$, $t^2(D)$, $t^3(D)$, generate the family

The set C might contain even compound primitives (2D-figures or 3D-figures), and in this situation when we represent the fractals we should achieve a projection of the structure

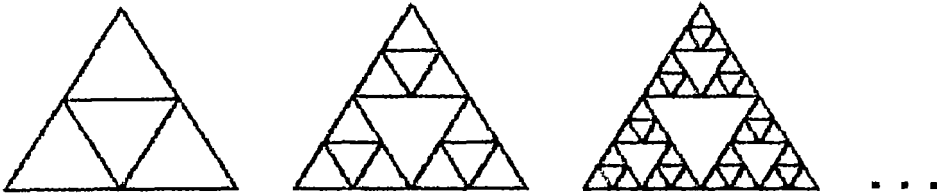


Figure 3

[4]

Some fractal transformations can be achieved modifying the commands string $t^n(D)$ through some operations

1) **Scaling** the figure can be increased k times, applying the following function

$$s^k : S \rightarrow S, s^k(x_1 x_2 \dots x_m) = x_1^k x_2^k \dots x_m^k,$$

to the string $t^n(D)$

2) **Translation** a figure can be translated k times on the direction of a primitive c if we apply the function

$$d_c^k(X) = c^k X$$

to the string $X = t^n(D)$

3) **Rotation** the function ρ^k defined below achieves k rotations of the fractal

$$\rho^k(x_1 x_2 \dots x_m) = \text{Next}^k(x_1) \text{Next}^k(x_2) \dots \text{Next}^k(x_m), k \in \mathbb{Z}$$

The rotation can be performed with an angle multiple of 45, 60 or 90 grades, depending on the selection of the primitives. Since the order of the primitives in the set C is important, we must respect the anticlockwise direction. We notice that the function ρ has the following property $\rho^k(t^n(X)) = t^n(\rho^k(X))$. This property simplifies the computation in fractals generation, fractals which have a primary detail compound from more than one primitive

Let's consider again the second example

$$\begin{aligned}
 t^n(\text{urdl}) &= t^n(u)t^n(r)t^n(d)t^n(l) = \\
 &= t^n(u)t^n(\text{Next}^{-2}(u))t^n(\text{Next}^4(u))t^n(\text{Next}^2(u)) = \\
 &= t^n(u)t^n(\rho^{-2}(u))t^n(\rho^4(u))t^n(\rho^2(u)) = \\
 &= t^n(u)\rho^{-2}(t^n(u))\rho^4(t^n(u))\rho^2(t^n(u))
 \end{aligned}$$

We notice that we can compute $t^n(u)$ one time, then the commands are executed, eventually transformed by the rotations ρ^{-2} , ρ^4 and ρ^2

If we have to recognize a fractals family then we will proceed as follows we find the primitives, construct the corresponding commands string, we decompose the commands string in substrings of length equal with $f(x)$, then every substring $f(x)$ is substituted with x . We apply the same action to the resulting string until we obtain D [5]. The commands string $t^n(D)$ from the second example,

rurdr uluru rurdr drdlld rurdr, is transformed by f^{-1} in

rurdr \rightarrow **r**, **uluru** \rightarrow **u**, **rurdr** \rightarrow **r**, **drdlld** \rightarrow **d** and **rurdr** \rightarrow **r**, then the resulting string **rurdr**, which is $f(r)$, is substituted by **r**

REFERENCES

- 1 F J Brandenburg, M P Chytil, On Picture Languages Cycles and Syntax - Directed Transformations, Technische Berichte der Fakultat fur Mathematik und Informatik Universität Passau, MIP-9020, 1990
- 2 D Dogaru, Metode noi în proiectare, Editura Științifică și Enciclopedică, București 1988
- 3 H A Maurer, G Rozenberg, E Welzl, Using String Languages to Describe Picture Languages, Information and Control, Vol 54, Nr 3, 1982
- 4 M Valda, A Posea, I Nistor și alții, Grafică pe calculator limbajele Pascal și C, Editura Tehnică București, 1992
- 5 R Vancea, S Holban, D Ciubotaru, Pattern Recognition, Editura Academiei, București 1969