About Digital Images and Lindenmayer Systems

SILVIU DUMITRESCU
Department of Informatics
University Transilvania of Brasov
Iuliu Maniu 50
ROMANIA
silviu.dumitrescu@unitbv.ro

Abstract: - In this paper, we are studying deterministic context free Lindenmayer Systems used to describe commands to a device that generates black and white digital images. Instead of well-known methods of drawing, we will paint squares, not lines. In the final part of the paper, we will discuss some important properties of growth functions of D0L-systems. In addition, we turn the discussion to gray scale or color digital image generation.

Key-Words: - Deterministic Context Free Lindenmayer Systems (D0L-systems), Digital Pictures, Growth Functions, Turtle

1 Introduction

The concepts and techniques of picture processing have arisen from many different disciplines, among them mathematics, computer science [4], engineering or biology. L-systems are suitable tools for drawing images of real life structures. This is due to their ability to model biological growth. Traditionally, the strings of symbols generated by L-systems are interpreted as images either using vector interpretation, or so-called turtle geometry interpretation. In this paper, we consider the second approach, where the symbols are translated into commands to a ‘turtle’ – a simple device moving on the plane used in drawing pictures.

In the first section, we present the main concepts used in this paper: context free Lindenmayer systems, a special case of them called deterministic systems (D0L-systems), and digital pictures, which are formed by units. As we will see, a unit is a square defined by position, length and color.

In the second section, we consider the turtle device, which works as a drawn device. The commands for turtle are described by a D0L-system. Considering Sierpinski triangle, we give an example of its generation. The main difference between this method and the others is that instead of drawing lines we are filling units. The problem with turtle, in this case, is that we have to scale the dimension of the unit in order to draw the image, with more details in the same frame.

In the main section of this paper, we will study some important properties of growth functions defined for those D0L-systems that are describing turtle movements. We discuss about a recursive formula based on the Cayley-Hamilton theorem.

Finally, we extend the discussion at gray-scale or color images generation. This is doing by introducing new attributes in the tuple that defines the turtle state.

2 Problem Formulation

2.1 Definitions and notations

We start this section with basic definitions involved in this paper.

Definition 1: [6] An 0L-system (context free Lindenmayer system) represents an ordered tuple \( (V, P_0, F) \), where \( V \) is the finite, nonempty set of symbols, usually called alphabet, \( P_0 \) is a nonempty word over \( V \), called initial word or axiom, and \( F \) represents a set of ordered pairs called productions.

\[
F = \{(a, P) \mid a \in V, P \in V^* \}
\]

We usually denote a production by \( a \rightarrow P \).

The main three differences between L-systems and well-known string grammars are:

- there is a unique alphabet (without separation between terminal and nonterminal symbols)
- for each symbol, \( a \in V \), exists at least a production in \( F \)
- a step in the derivation chain represents replacement of all symbols of current word by productions in \( F \).

1) For a set \( V \), \( V^* \) denotes the set of all strings over \( V \), including the empty string \( \lambda \); \( V^* = V^* \setminus \{\lambda\} \) denotes the set of all strings over \( A \) except the empty string \( \lambda \).
The prefix 0, in 0L-systems, denotes the property of context freeness. Thus, the replacement of a symbol does not care about the context where the symbol exists inside of the word.

In this paper, we consider a special case of 0L-systems, called D0L-systems, deterministic systems, which mean that for each symbol, we have exactly one production in $F$.

Definition 2: A digital image $\Sigma$ is a finite rectangular array whose elements are called units. Each unit, $U = (x, y, l)$, of $\Sigma$ is a square defined by a pair of Cartesian coordinates $(x, y) \in \mathbb{R}^2$ (down-left corner) and a length $l \in \mathbb{R}$, which is constant.

A unit, $U$, in a digital image $\Sigma$ has two types of neighbors:
- its four horizontal and vertical neighbors: $U_l = (x_l, y_l, l)$, $U_r = (x_r, y_r, l)$, $U_u = (x_u, y_u, l)$, $U_d = (x_d, y_d, l)$ such that $|x-x_i|+|y-y_i| = 1$, $i \in \{l, r, u, d\}$
- its four diagonal neighbors: $U_ul = (x_ul, y_ul, l)$, $U_ur = (x_ur, y_ur, l)$, $U_dl = (x_dl, y_dl, l)$, $U_dr = (x_dr, y_dr, l)$ such that $|x-x_i| = |y-y_i| = 1$, $i \in \{ul, ur, dl, dr\}$

We shall refer to the neighbors of first type as the 4-neighbors of $U$ and the neighbors of both types, collectively, as the 8-neighbors of $U$. The former neighbors are said to be 4-adjacent to $U$, and the latter, 8-adjacent. Note that if $U$ is on the border of $\Sigma$, some of its neighbors may not exist.

2.2 The drawn device
As we saw, small units compose digital images and we need a device to draw them.

We call this device “turtle”. A turtle has mobility to move inside of the Euclidean plan. The turtle state is given by the triple $(x, y, \alpha)$, where $(x, y) \in \mathbb{R}^2$ are the plan coordinates and $\alpha \in \{0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4\}$ is the direction of turtle movement.

A simple move of the turtle means advance one unit in the direction given by $\alpha$. The simple move can be with or without drawing depends, the turtle is up or is down.

In the first step, we will draw only black and white digital images. For this, we will consider background colored in white and the image colored in black.

The movements of the turtle, in order to draw a black and white digital image, can be described by a D0L-system, $D = (V, P_0, F)$, where $V = \{U, u, +, -\}$ and $P_0 = \{U\}$. The symbols of $V$ have the following meaning:
- $U$, simple move with drawing (we say that the turtle is down)
- $u$, simple move without drawing (we say that the turtle is up)
- $+$, rotate the drawing direction with angle $\alpha$, counterclockwise
- $-$, rotate the drawing direction with angle $\alpha$, clockwise.

In addition, the initial state of the turtle can be modified for each individual case, but most likely it is $(0, 0, \pi/2)$.

Note that the only change to the original turtle interpretation by Prusinkiewicz is the interpretation of $U$: Instead of drawing lines, we paint black squares.

Example 1: The D0L-system that describes the Sierpinski triangle is defined by the tuple $D = (V, P_0, F)$, where $F = \{(U, U-u-u-U+u-U), (u, uu), (+, +), (-, -)\}$. The initial state of the turtle is the usual one. A derivation in three steps using these productions looks as follows:

\[U \Rightarrow U-u-u+U+u-U\Rightarrow U-u-u+U+u-U-u-u-u+U-u+u-U \Rightarrow U-u-u+U+u-U-u-u-u+U-u+u-U \Rightarrow U-u-u+U+u-U-u-u-u+U-u+u-U\]

During derivation process the word $w_k$, obtained after $k \geq 1$ steps ($w_0 = U$), will be represented inside of a white square formed by $2^k \times 2^k$ units. When $k$ increases, the image details increase. However, when $k$ increasing the image will not be properly represented because it will have huge dimension. The simple solution is to scale the image at each step, which means reducing turtle unit dimension. The minimum dimension of the turtle unit is one pixel.

When $k$ approaches infinity, the image is full detailed, but in real representation, this cannot be doing, so we have to consider the finite number of steps until the turtle unit approaches one pixel.

In previous example, we start with the word $w_0$. This is representing by a black square inside of a frame by $q \times q$ pixels. After first derivation step, the turtle unit will be scaled such as we can draw four units inside of the frame. $w_1$ is represented as following:

\[
\begin{array}{cccc}
\text{U} & \text{U} & \text{U} & \text{U} \\
\text{U} & \text{U} & \text{U} & \text{U} \\
\text{U} & \text{U} & \text{U} & \text{U} \\
\text{U} & \text{U} & \text{U} & \text{U} \\
\end{array}
\]

$w_2$ is formed by 16 units. The process is simple. Each black unit is scaled such as instead of one
initial unit we create four new small units colored by the same pattern. The new image, after two steps, is looking as following:

The scale factor is $\frac{1}{2}$.

Continuing the derivation process, after $k$ steps, initial frame is formed by $2^k \times 2^k$ turtle units.

We say about a point $(x, y) \in [0, q] \times [0, q]$ that is in the image generated if it is inside of a unit square colored in black.

We define the image generated on step $k$ by:

$$T(w_k) = \{(x, y) \mid (x, y) \text{ is in a black unit of } 2^k \times 2^k \text{ units}\}$$

If the sequence $T(w_0), T(w_1), \ldots$ converges in the standard Hausdorff metric, the limit

$$T(D) = \lim_{k \to \infty} T(w_k)$$

is the infinite resolution image defined by the D0L-system $D$ with scaling defined previously.

### 3 Growth functions of systems that generates images

An important question when we deal with the D0L-systems is the length of their words. The growth function $f_D : \mathbb{N} \to \mathbb{N}$ of the D0L-system $D$ is defined by:

$$f_D(k) = |w_k|$$

and means the length of the word $w_k$.

By studying growth functions, we determine which types of biological growth D0L-system is capable of modeling.

Here we discuss a special matrix representation of homomorphism introduced [6] to help growth function calculation.

**Definition 3:** The incidence matrix of a homomorphism $F$ is defined by the following square matrix (the dimension of the matrix is equal to number of symbols in alphabet):

$$M(F) = (m_{a,b})_{a,b \in \mathcal{V}}, \quad m_{a,b} = |F(b)|_a$$

The element $m_{a,b}$ represents the number of occurrences of symbol $a$ in the production where $b$ is the left hand side member.

In example 1, we have matrix $M$ defined as following:

$$M(F) = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix}$$

Recall now the Parikh vector, which describes the letter distribution of a word. Let $\eta$ be a column vector with all elements 1 and $\Pi$ the Parikh vector of the axiom $P_0$. We can now discuss an important identity regarding the growth function of D0L-systems:

$$f_D(k) = \eta^T M(F)^k \Pi^T$$  \hspace{1cm} (1)

that calculates the length of the word $w_k$, after $k$ steps, starting with the axiom.

Remark: in the previous formula we used an important result proofed in [7]: $M(F^k) = M(F)^k$, which means that the incidence matrix after $k$ derivations equals the incidence matrix after one derivation on power $k$.

Now we define a recursive formula for the growth function starting from the Cayley-Hamilton theorem. This says that every square matrix over the real or complex field satisfies its own characteristic equation.

Therefore, for a matrix of dimension $n$, we can calculate de coefficients, $c_i \in \mathbb{R}$, $i = 1, n$, such as the Cayley-Hamilton formula is:

$$M^n = c_1 M^{n-1} + \ldots + c_{n-1} M + c_n M^0$$

where $M^0 = I_n$.

If we consider that $M$ is the incidence matrix, we apply previous formula to (1)

$$\eta^T M(F)^k \Pi^T = c_1 \eta^T M(F)^{n-1} \Pi^T + \ldots + c_{n-1} \eta^T M(F) \Pi^T + c_n \eta^T M(F)^0 \Pi^T$$

that means:

$$f_D(n) = c_1 f_D(n-1) + \ldots + c_{n-1} f_D(1) + c_n f_D(0),$$

where $n = |\mathcal{V}|$.

We conclude the previous results into next theorem.
Theorem 1: For the growth function of D0L-systems, \( D = (V, P_0, F) \), we can find a recursive formula given by:

\[
f_D(n+i) = c_1 f_D(n+i-1) + \ldots + c_{n-1} f_D(i+1) + c_n f_D(i),
\]

where \( n=|V| \), \( i \geq 0 \).

For example 1 the previous formula coefficients are \( c_1 = -7 \), \( c_2 = 17 \), \( c_3 = -17 \), \( c_4 = 6 \) and the recursive formula is:

\[
f_D(i+4) = -7 f_D(i+3) + 17 f_D(i+2) - 17 f_D(i+1) + 6 f_D(i),
\]

\( i \geq 0 \) with \( f_D(3), f_D(2), f_D(1), f_D(0) \) given values.

To draw gray-tone images using L-systems we can modify the turtle state by adding the weight, \( g \). The weight can be an arbitrary real number, initially 1. Instead of painting always a black square, the turtle paints a grey square, whose darkness is giving by the current weight of the turtle. The local grayness function \( f: \mathbb{R}^2 \rightarrow \mathbb{R} \) is describing the darkness of every point of the plan. Initially the plane is completely white, that is, the local grayness function is \( f(x, y) = 0 \), for all, \( x, y \in \mathbb{R} \). The weight of the turtle is simply adding to the darkness of the part \( f'(x, y) = f(x, y) + g \). All concepts described for black and white images remain unchanged.

4 Conclusions

Digital images can be decomposing in units, and can be drawn with special devices. The movements of those devices can be described by D0L-systems. We can find a recursive formula for growth functions that describes the word generated during the derivation. We can draw not only black and white images, but grayscale or colored images. Drawing of color images is based on weight turtles. We can introduce the forth member in the tuple that describes the state of the turtle for grayscale images and two more attributes to describe color images, corresponding to RGB codification of colors.

References:

MATHEMATICAL METHODS, COMPUTATIONAL TECHNIQUES, INTELLIGENT SYSTEMS

ISSN: 1790-2769

51