

# Generación de estructuras arbóreas tridimensionales basados en los Sistemas de Lindenmayer.

Simón Tagtachian, Cristina Argumedo.

Centro CAO. Facultad de Arquitectura, Diseño y Urbanismo (Proyecto UBACYT AR017);

Dto. de Computación, Facultad de Ciencias Exactas y Naturales.

Universidad de Buenos Aires.

Fax 782-9168, e-mail: [crisar@huiyin.cao.uba.ar](mailto:crisar@huiyin.cao.uba.ar)

Fax 816-5783, e-mail: [simon@gidef.uba.ar](mailto:simon@gidef.uba.ar)

Los sistemas de Lindenmayer (L-systems) fueron creados con el objetivo principal de simular el crecimiento y desarrollo de plantas. Estos sistemas se aplican aquí para obtener estructuras arbóreas tridimensionales y poder generar -en un futuro- bosques virtuales. Se buscaron, compararon y hallaron los parámetros que definen cada una de estas estructuras, se obtuvieron diferentes formas arbóreas y finalmente se las representó tridimensionalmente.

## Introducción

El biólogo Aristid Lindenmayer (1925-1989) fundó las bases de los modelos algorítmicos (determinísticos y estocásticos) que pueden describir fenómenos de crecimiento de estructuras. Particularmente, estos algoritmos, pueden ser utilizados en la simulación del crecimiento de plantas. En su honor, a estos procesos se los conoce como L-systems.

El concepto central de los L-systems es la "reescritura" o la recursividad de los procesos. El algoritmo consiste en una microgramática que posee símbolos y reglas de sustitución. A partir de formas simples, se construye una estructura compleja, la cual puede ser interpretada en términos gráficos y mostrada o representada como una estructura arbórea. La microgramática utilizada, genera una ristra de caracteres que luego puede ser interpretada en términos gráficos.

## Interpretación gráfica de las cadenas de caracteres

La idea básica de la interpretación de las ristas de caracteres es la "representación de la Tortuga". En estructuras bidimensionales, ésta consiste en suponer que estamos parados arriba de una tortuga. La tortuga puede avanzar o retroceder, rotar y -con una pluma atada- escribir mientras avanza o avanzar sin escribir -sin utilizar la pluma-. Estas son las ideas introducidas por el lenguaje LOGO. Luego, se puede decir que el 'estado de la tortuga' viene dado por  $(x,y,\alpha)$  donde  $(x,y)$  son las coordenadas cartesianas donde se halla la tortuga; el ángulo  $\alpha$ , se interpreta como la dirección en que se encuentra avanzando la tortuga. Dando la longitud del paso (la tortuga avanza en distancias discretas, o pasos) y el ángulo de incremento de rotación  $\theta$  (rotación discreta), podemos definir la trayectoria que recorrerá la tortuga.

Los siguientes son los 'comandos' principales que 'comprende' la tortuga en el espacio bidimensional:

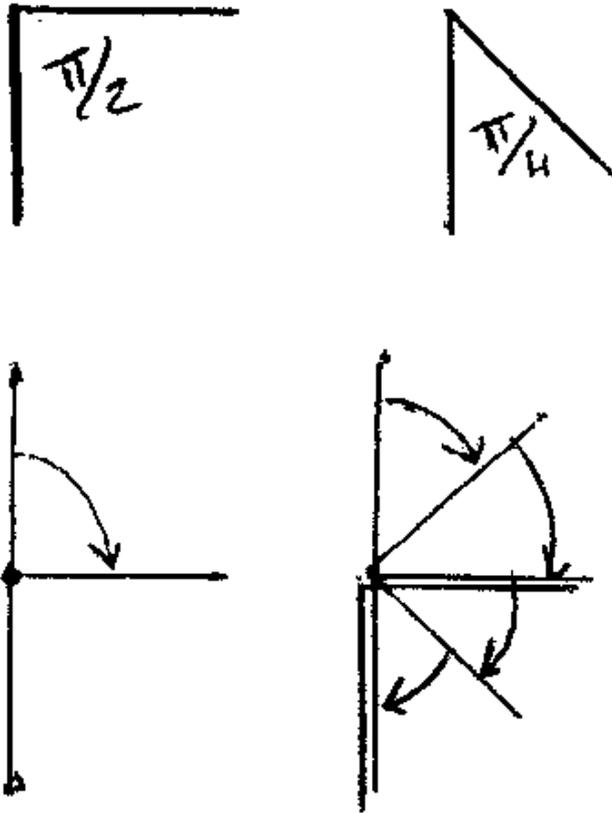
F La tortuga avanza -dibujando- un paso de longitud 'd'. El estado inicial de la tortuga es  $(x,y,\alpha)$  y el final es  $(x',y',\alpha')$ . La correspondencia se obtiene haciendo,  $x'=x+d*\cos(\alpha)$  ;  $y'=y+d*\sin(\alpha)$  y  $\alpha'=\alpha$ . Se traza una línea entre  $(x,y)$  y  $(x',y')$ .

f La tortuga avanza pero no dibuja. Las coordenadas se transforman de la misma manera que con F.

+ La tortuga rota a la izquierda en un ángulo  $\theta$ . Para obtener el estado final de la tortuga hacemos,  $x'=x$  ;  $y'=y$  ;  $\alpha'=\alpha+\theta$ .

- La tortuga rota a la derecha en un ángulo  $\theta$ . Para obtener el estado final de la tortuga hacemos,  $x'=x$  ;  $y'=y$  ;  $\alpha'=\alpha-\theta$ .

Dada la cadena inicial, el estado inicial de la tortuga  $(x_0,y_0, \alpha_0)$  y los parámetros -fijos-  $d$  y  $\theta$ , este tipo de interpretación nos permite tomar a la ristra como la secuencia de instrucciones que debe cumplir la tortuga.



En la ilustración 1, se puede ver las diferentes maneras de obtener un ángulo. Tener presente que un ángulo se puede lograr de diferentes maneras: si  $\theta = \pi/2$  entonces un ángulo de  $\pi/2$  se logra tomando como ristra  $F+F$ . Si  $\theta = \pi/4$  entonces la ristra es  $F++F$ .

Podemos ver en los siguientes ejemplos cómo es el funcionamiento de las ristas elementales. Como parámetros damos el valor de  $\theta$  y de la ristra de reemplazo. No es necesario usar una computadora para generar estos gráficos (sí es absolutamente necesario para ristas grandes). Aquí vemos unos ejemplos trazados a mano alzada. Generar los dibujos 'a mano' es muy útil en el proceso de elección y comprensión de la acción de las ristas principales.

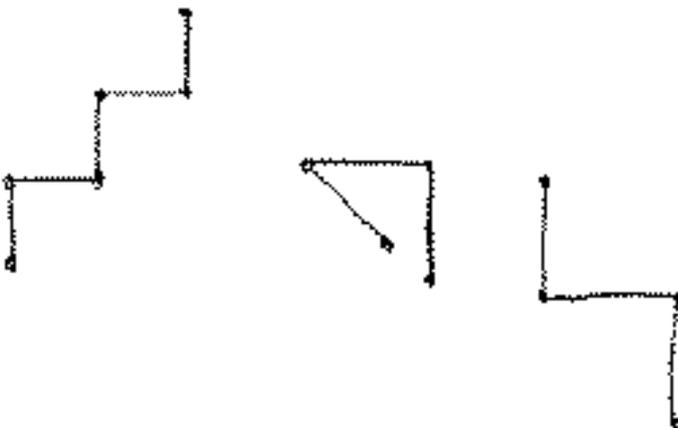


Ilustración 3:  $\theta = \pi/4$ ,  $F \rightarrow F++F+++F$

Ilustración 4:  $\theta = \pi/2$ ,  $F \rightarrow F+F-F$

Ilustración 2:  $\theta = \pi/2$ ,  $F \rightarrow F+F-F+F-F$

Ilustración 5:  $\theta = \pi/4$ ,  $F \rightarrow F+++F---F+++F---F+++F$

Ilustración 6:  $tita = \pi/3$ ,  $F \rightarrow F + F - F + F$



Ilustración 7:  $tita = \pi/4$ ,

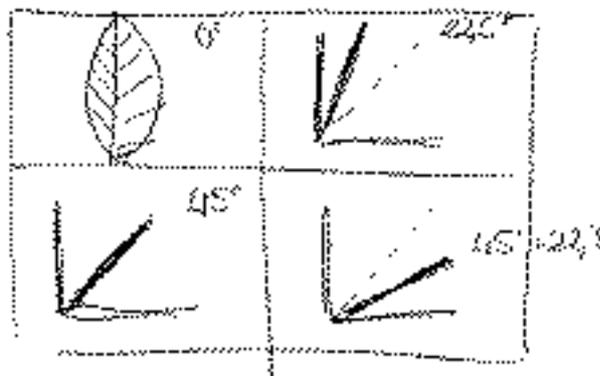
$F \rightarrow F + F + F - F - F - F - F + F + F$

**Proceso de mapeo de una imagen.**

El proceso de mapeo de una imagen, nos permite incluir una imagen real dentro de una imagen generada o de síntesis, obteniéndose de esta manera una imagen mixta, la cual posee componentes reales y componentes generados o virtuales.

El ejemplo más sencillo es la captura de una hoja de un rosal. En nuestro caso, tomamos una hoja y capturamos su imagen con una cámara de video, digitalizándola de manera que quede con su eje longitudinal ubicado verticalmente. Los parámetros fundamentales del archivo son su ancho y longitud y la cantidad de colores que posee. Con estos valores, y aplicando los algoritmos correspondientes, se pueden obtener tres tipos de efectos, los cuales logran, a partir de una imagen bidimensional, una simulación de imagen tridimensional.

La rotación de las hojas se realiza utilizando las matrices de rotación o transformaciones de coordenadas. Para ubicar la hoja digital en el lugar correcto, se debe determinar primero el punto de inserción de ésta dentro de la planta general, y según ese punto, se realiza la rotación. Para hacer esto usamos:



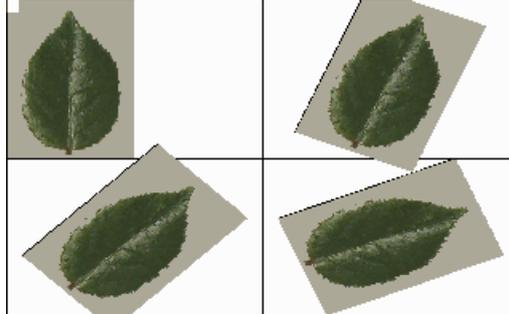
$$x' = x * \cos(tita) + y * \sin(tita)$$

$$y' = -x * \sin(tita) + y * \cos(tita)$$

Las rotaciones habitualmente las vamos a medir en grados o en radianes. La conversión esta dada por  $2 * \pi = 360$  grados.

Las matrices de rotación, pueden variar en un signo y esto influirá en el sentido de la rotación final.

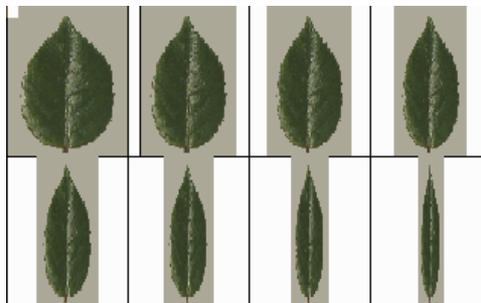
$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$



En las imágenes siguientes (ilustración 8) se pueden observar las rotaciones del anverso de una hoja de rosal. También se puede ver los esquemas de rotación y los valores en ángulos y radianes.

$[0, (2\pi)/16, (2\pi)/8, (2\pi)/8 + (2\pi)/16]$

Un proceso similar se sigue si se quiere simular la rotación de una hoja (alrededor del eje longitudinal). Se realiza una compresión en el eje x que simula ser una rotación en un ángulo alfa (ilustración 9).



Esta correspondencia se puede ver en la ilustración 10 y las equivalencias entre el porcentaje de compresión y la rotación en grados, se puede ver en la tabla 1.



Tabla 1

1.0 --> 0°	0.8 --> 36° 52'	0.7 -->45° 35'	0.6 -->53° 7'
0.5 --> 60°	0.4 --> 66° 25'	0.3 --> 72° 32'	0.2 -->78° 27'

De la misma manera, se puede realizar rotaciones 'hacia atrás' de la hoja. Esto en la práctica se logra comprimiendo en el eje y el grafico. Las correspondencias en rotaciones de ángulos, en este caso, son las mismas.

La técnica de mapeos de imágenes cobra importancia cuando se la inserta dentro de las estructuras de los sistemas L, dando entonces una imagen de síntesis mixta (en este trabajo no se realizó este mapeo sobre las hojas de los árboles). La ramificación y estructura general está dada entonces por los algoritmos, pero el aspecto de cada hoja 'es' el aspecto de una hoja real mapeado sobre el espacio destinado a las hojas en la generación del objeto.

### Ángulos de rotación simple.

Los utilizamos para rotar una hoja -o cualquier otro elemento de la planta- en el espacio. Según las tres diferentes matrices de rotación, se puede generalizar el movimiento espacial de la tortuga. Los símbolos de rotación utilizados en la confección e interpretación de las ristas son lo siguientes:

+: gira un ángulo delta usando  $R_u(+\delta)$  alrededor del eje z.

- : gira un ángulo delta usando  $R_u(-\delta)$  alrededor del eje z.

| : gira 180 grados usando  $R_u(\pi)$  alrededor del eje z

& : gira un ángulo delta usando  $R_l(+\delta)$  alrededor del eje y

^ : gira un ángulo delta usando  $R_l(-\delta)$  alrededor del eje y

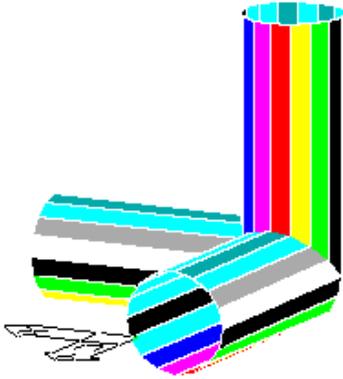
\ : gira un ángulo delta usando  $R_l(+\delta)$  alrededor del eje x

/ : gira un ángulo delta usando  $R_l(-\delta)$  alrededor del eje x



Las matrices de rotación son las siguientes:

$$\begin{pmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{pmatrix} = R_u(\delta)$$



$$\begin{pmatrix} \cos(\delta) & 0 & -\sin(\delta) \\ 0 & 1 & 0 \\ \sin(\delta) & 0 & \cos(\delta) \end{pmatrix} = R_l(\delta)$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta) & -\sin(\delta) \\ 0 & \sin(\delta) & \cos(\delta) \end{pmatrix} = R_h(\delta)$$

Cualquier objeto puede ser ubicado en el espacio usando una traslación más una función de rotación que incluya estas matrices.

Una manera más precisa de ubicar un sólido en el espacio es utilizando los ángulos de Euler. Estos ubican unívocamente en el espacio, un cuerpo sólido. Los ángulos se denotan con tita, phi, ji, y se comportan de manera similar a las matrices de rotación. Finalmente, los caracteres leídos de las ristas tendrán que interpretar los movimientos utilizando una compleja matriz de rotación y desplazamiento, cuya transformación final es la siguiente :

Coordenadas iniciales: mx, my, mz; corrimientos en los ejes: corrx, corry, corrz -desplazamientos-

Coordenadas finales: mmx, mmy, mmz

$$mmx = corrx + (\cos(ji) * \cos(phi) - \cos(tita) * \sin(phi) * \sin(ji)) * mx + (\cos(ji) * \sin(phi) + \sin(ji) * \cos(tita) * \cos(phi)) * my + (\sin(ji) * \sin(tita)) * mz$$

$$mmy = corry + (-\sin(ji) * \cos(phi) - \cos(ji) * \cos(tita) * \sin(phi)) * mx + (-\sin(ji) * \sin(phi) + \cos(ji) * \cos(tita) * \cos(phi)) * my + (\cos(ji) * \sin(tita)) * mz$$

$$mmz = corrz + (\sin(tita) * \sin(phi)) * mx + (-\sin(tita) * \cos(phi)) * my + (\cos(tita)) * mz$$

### Ramas de grosor variable.

En el crecimiento de las plantas generadas con L-S, es fundamental incluir alguna variable o método que permita valorizar la antigüedad de los segmentos que forman los troncos, para darles algún tipo de grosor. Los métodos pueden ser variados. El caso más sencillo toma en cuenta la cantidad de corchetes abiertos hasta ese momento (los corchetes "[" y "]" marcan los procesos de bifurcación). Puede ser confuso, pues se generan grosores no deseados. Esto se produce cuando la secuencia reescribe segmentos en el espacio ya ocupado. Otras dos alternativas son utilizar criterios posicionales -toman en cuenta la altura de la rama y la distancia al tronco-, o generar una ristra testigo que se implemente al mismo tiempo que la ristra principal y que conlleve la 'edad' de las ramas.

### Arboles tridimensionales generados por rotación.

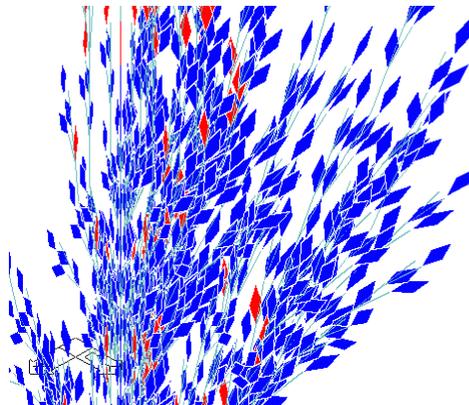
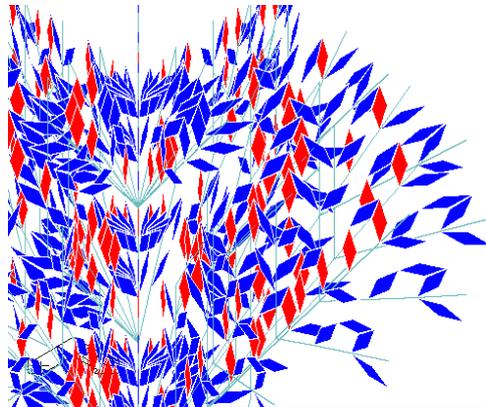
Una vez definidas todas estas herramientas, podemos generar árboles 3d por rotación. Los pasos son los siguientes:

- a) generar la estructura fundamental que representará el árbol con la ristra primaria en el plano xz
- b) agregar la capa de 'hojas' que dará mayor realismo utilizando una ristra secundaria que puede tener una secuencia diferente a la primaria.
- c) generar una rotación en el plano xy eligiendo convenientemente el ángulo de rotación y la determinar la porción del espacio a generar.
- d) una vez obtenido el archivo de coordenadas, que indica los comienzos y finales de las ramas, y la posición y orientación de la hojas, se lo transforma a DXF y se lo grafica y renderiza de la manera más conveniente.

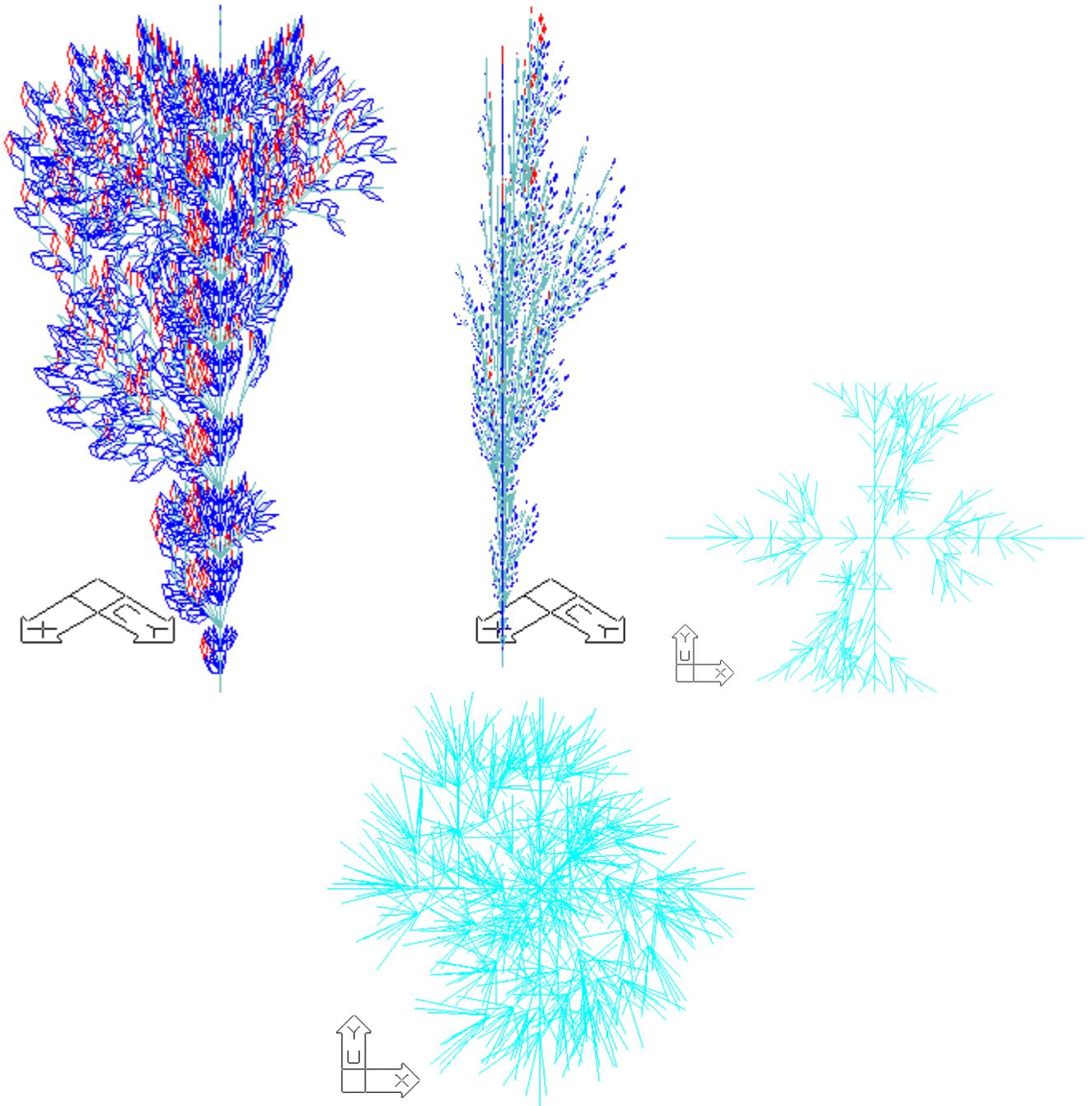
Todo este proceso, es una manera bastante sencilla, pero introduce un alto porcentaje de simetrías -la propia simetría de rotación- contrariamente a lo esperado, esta simetría es muy difícil de apreciar -a menos que se vea el árbol en el plano xy- debido a la gran cantidad de segmentos y a que la vista atraviesa todo el objeto imposibilitando la fijación de vistas.

### Arboles tridimensionales puros.

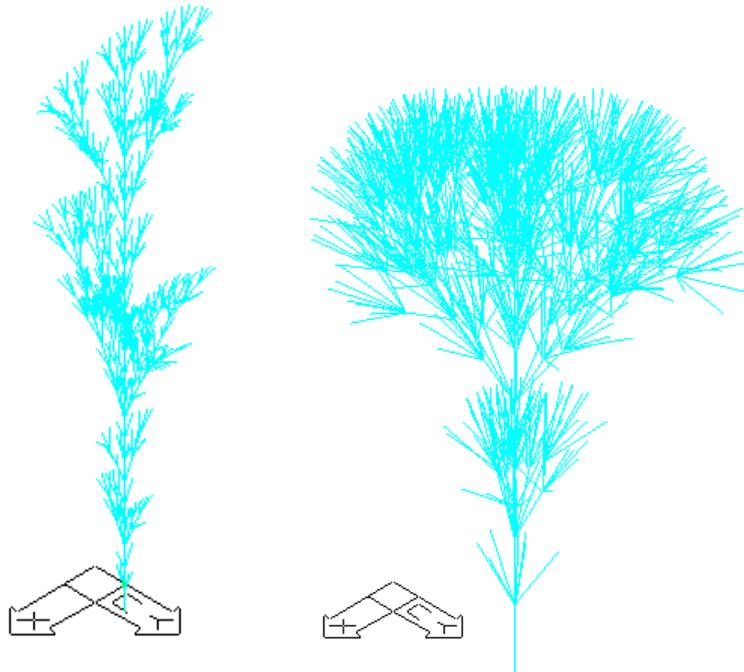
La manera más realista de generar un árbol por L-S es utilizando las matrices de rotación complejas antes descritas (por lo menos utilizarán 6 caracteres diferentes por cada sentido de rotación) combinándolas con ristras suplementarias para las hojas y flores (utilizarán todos los caracteres necesario para determinar el tipo de hoja o floración). Asimismo, es fundamental la generación de los troncos para lograr mayor realismo (en este trabajo, los troncos aun no tienen sección). Muchas son las ventajas de estos árboles donde ya no existe, a priori, ningún tipo de simetría de rotación. Además, entre árboles generados a partir de las mismas ristras, se pueden obtener ligeras diferencias estructurales utilizando componentes aleatorias -por ejemplo en la longitud de las ramas y orientación de las hojas-. Esto siempre aumentará el realismo de la imagen final.



Estas dos imágenes muestran detalles de árboles generados por rotación.



[Las dos imágenes superiores izquierdas son árboles generados por rotación; las tres imágenes de la derecha y la inferior, muestran árboles 3d puros]



#### A manera de conclusión y próximos pasos.

Se obtuvieron estructuras arbóreas de diferentes formas y de alto realismo. Se delinearón e implementaron las herramientas de búsqueda para lograr y diseñar estas estructuras. Queda aún pendiente lograr mayor realismo dándole volúmen a los troncos de los árboles, como así también realizar un renderizado de las imágenes. La composición final, utilizando ditintos tipos de árboles y fondos, es un trabajo que requiere paciencia y ensayo, teniendo especial cuidado en que no haya incompatibilidades entre los diferentes mapas de colores de las imágenes primarias. De todos modos, los resultados están a la vista y la utilización de estas estructuras en el campo de la computación gráfica y el arte, son muy amplias.

#### Bibliografía:

- [1] Barnsley M., Fractals Everywhere. Academic press, 1988
- [2] Barnsley M. y Sloan A., A better way to compress images. Byte - Enero 1988
- [3] Prusinkiewicz P.y Lindenmayer A., The Algorithmic Beauty of plants. Springer Verlag. 1990. New York.
- [4] Goldtein, Mecánica clásica, Editorial Aguilar, 1972.
- [5] Hofstadter D.; Godel, Escher y Bach. Consejo Nacional de Ciencia y Tecnología. Mexico, 1979.
- [6] S.Tagtachian, C.Argumedo. Generación de imágenes virtuales basadas en estructuras de Lindenmayer. 1er Congreso Internacional de Matemática y Diseño. FADU-UBA., 1995

Nota: las imágenes presentadas fueron generadas con programas propios escritos en Borland C++ 3.0 y Quik Basic 4.5. Se generaron archivos DXF que luego fueron importados en Autocad y así fueron logradas las imágenes finales mostradas.