

Sobre la compresión de imágenes y atractores de sistemas

Manuel Murillo Tsijli*

Resumen

En este trabajo se hace una breve introducción al tema de la compresión de imágenes y el formato. FIF. Se presentan algunos sistemas iterados de funciones y sus atractores.

Introducción

La compresión de imágenes juega un papel importante en el almacenamiento de imágenes con un menor costo, así como en la rápida transmisión de datos. Claro que al comprimir y luego descomprimir estos datos se puede perder información; esta información perdida muchas veces no se percibe por el ojo humano o es redundante, y los métodos de compresión eficientes son los que logran extraer el "espíritu" de la imagen, para después de descomprimir, reproducir una imagen muy cercana a la original. En la naturaleza muchos objetos mantienen una autosemejanza que se puede representar como el *atractor* de un sistema

iterado de funciones, tal es el caso de las nubes, helechos, plantas, árboles, arbustos, etc. La teoría de los *fractales* tiene aplicaciones en la compresión de imágenes, en el sentido de que podemos pensar que un objeto de esta naturaleza se puede transmitir o guardar eficientemente, reproduciéndolo en el momento que se desee.

En 1987, dos matemáticos del Instituto Tecnológico de Georgia, Michael Barnsley y Alan Sloan, formaron Iterated Systems Inc., una compañía con base en Atlanta, en donde se usa el desarrollo de la teoría *fractal* aplicada a la compresión. Uno de los productos más conocidos es la enciclopedia *Encarta*, publicada por la compañía Microsoft, que incluye en un CD-Rom miles de artículos, fotografías a color, mapas, etc. Otro producto es el *Fractal Imager*¹, un programa que recibe un archivo, preferiblemente en formato. JPG, y lo comprime en otro. FIF (*fractal image format*), que se puede descomprimir

* Profesor Asociado, Escuela de Matemática, Instituto Tecnológico de Costa Rica. Direcciones electrónicas: mmurillo@itcr.ac.cr, mmurillo@costarricense.cr.

¹ En Internet se pueden conseguir en <http://www.iterated.com>, junto con una gran variedad de fotos comprimidas en formato .FIF.

y ver con el *Fractal Viewer*, aumentando secciones de ella a cualquier nivel.

El primer esquema de compresión de este tipo publicado, fue la tesis doctoral de Arnaud Jacquin, estudiante de Barnsley, en 1989. Otros aportes importantes los dieron Y. Fisher, R. D. Boss y E. W. Jacobs, entre otros.

Una imagen es guardada en el computador como una colección de valores que indican el nivel de gris o color en cada punto (*pixel*) de la imagen inicial. Los archivos en formato .BMP, abreviado de BitMaP, son formatos de mapas de bits para Windows y tienen extensión .BMP, y algunas veces llevan la extensión .RLE, iniciales de *run length encoding*, que indica que los datos del mapa de caracteres fueron comprimidos usando uno de los dos métodos de compresión .RLE. Los archivos .BMP comprimen la información del color usando 1, 4, 8 ó 16 bits por *pixel* (bpp). El cálculo de los bpp determinan el máximo número de colores que la imagen puede tener, y está dado por 2^{bpp} . Si la imagen es un 1-bpp hay 2 colores posibles,

si la imagen es un 8-bpp, lo más usual, hay 256 colores, mientras que si es 24-bpp tendríamos más de 16 millones de colores. La estructura de cada uno de los diferentes formatos que aparecen en el Cuadro 1 y como guardan la información se puede encontrar en [13].

Comprimiendo imágenes con IFS's

El teorema 1 garantiza que el conjunto invariante puede ser una buena aproximación del conjunto inicial si la unión de copias pequeñas están cerca de este, pues en este caso tendríamos que $D(E,F)$ se acercaría a cero. En este sentido y como consecuencia de este teorema, el siguiente corolario nos dice que un conjunto compacto se puede aproximar tanto como se quiera, bajo la métrica Hausdorff, por un conjunto autosemejante e invariante de un sistema iterado de funciones. La demostración de ellos se puede encontrar en Falconer (1990, p. 134). Si el conjunto es más complicado, este se puede ver como la superposición de varios conjuntos invariantes de varios sistemas de funciones.

CUADRO 1
Diferentes formatos gráficos

Formato	Máx. bpp	Número máximo de colores	Tamaño máximo de la imagen	Método de compresión
.BMP	24	16 777 216	65 535 × 65 535 pixeles	RLE
.GIF	8	256	65 535 × 65 535 pixeles	LZW
JPEG	24	16 777 216	65 535 × 65 535 pixeles	JPEG
.PCX	24	16 777 216	65 535 × 65 535 pixeles	RLE
.PNG	48	281 474 976 710 656	2 147 483 647 × 2 147 483 647 pixeles	deflación
TIFF	24	16 777 216	4 294 967 295 pixeles en total	LZW, RLE, otros

Teorema 1 (Collage). Suponga que se tienen $S_i: \mathbb{R}^n \rightarrow \mathbb{R}^n$ funciones de crecimiento acotado $\forall i=1, \dots, m$, y suponga que para todo i se tiene $|S_i(x) - S_i(y)| \leq c|x - y|$ para todo $x, y \in \mathbb{R}^n$, con $c < 1$. Sea $E \subset \mathbb{R}^n$ compacto no vacío, entonces

$$D(E, F) \leq D\left(E, \bigcup_{i=1}^m S_i(E)\right) \frac{1}{1-c} \quad (1)$$

donde F es el conjunto invariante de los S_i y D la métrica Hausdorff.

Corolario 2. Sea $E \subset \mathbb{R}^n$ compacto no vacío. Para todo $\delta > 0$ existe un sistema iterado de funciones de crecimiento acotado S_1, \dots, S_n con conjunto invariante F tal que $D(E, F) < \delta$.

Podemos pensar que una imagen en una escala de gris, con todos los tonos de gris entre el blanco y el negro, se puede ver como $z = f(x, y)$ que representa el nivel de gris en el *pixel* (x, y) , donde el menor valor de f es blanco y el mayor es negro. Asumiremos que $f: I \times I \rightarrow I$, donde $I = [0, 1]$. Los métodos de compresión de imágenes se pueden evaluar usando la *razón de compresión*, que es la razón de la memoria requerida para guardar la imagen como colección de *pixeles* y la memoria requerida para guardar la representación de la imagen en forma comprimida. Un archivo de alrededor de 720KB se puede comprimir en FIF en 12KB, para una razón de compresión de 720:12, es decir, 60:1.

Las aplicaciones afines son, en general, rotaciones, reflexiones, traslaciones, contracciones o cambios en los ejes, que se pueden representar por

$$v_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}. \quad (2)$$

Se pueden usar diferentes métricas para medir la distancia entre dos imágenes, dos de ellas son la métrica *rms* y la métrica *sup* que es más sencilla:

$$d_{rms}(f, g) = \sqrt{\int_{I^2} (f(x, y) - g(x, y))^2 dx dy} \quad (3)$$

$$d_{sup}(f, g) = \sup_{(x, y) \in I^2} |f(x, y) - g(x, y)|. \quad (4)$$

Sean $f_0, f_1, f_2: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ tal que

$$f_0((x, y)) = \frac{1}{2}(x, y)$$

$$f_1((x, y)) = \frac{1}{2}(x, y) + \frac{1}{2}(1, 0) \quad y$$

$$f_2((x, y)) = \frac{1}{2}(x, y) + \frac{1}{2}(0, 1)$$

Con esto definimos un sistema iterado de funciones (IFS) que realizan la lista contractiva de razones $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. El *atractor* de este IFS es el triángulo de Sierpiński. Podemos pensar en este sistema iterado de funciones como una máquina de fotocopiar con tres lentes, la primera lente forma una copia reducida al 50% en la parte izquierda de la figura, la segunda lente forma una copia reducida al 50% en la parte derecha de la figura, y la tercera lente forma una copia reducida al 50% en la parte superior centrada de la figura. Ahora tomamos una figura cualquiera, le sacamos la primera copia en nuestra máquina y esta la volvemos a copiar, hacemos esto varias veces y debemos obtener una figura similar al *atractor* del sistema. Una exposición muy detallada se puede encontrar en [1], [14].

Esta máquina de copiado de reducción múltiple (MCRM), se puede representar como en la Figura 1, donde se indica

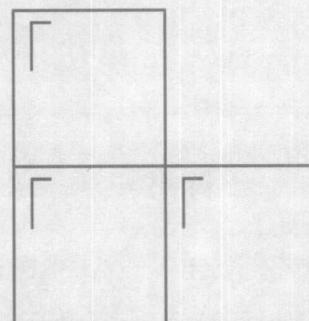


FIGURA 1
MCRM cuyo *atractor* es el triángulo de Sierpiński

además la orientación de la copia. El *atractor* de este MCRM es el triángulo de Sierpiński, con un cambio de ejes. En total hay 8^3 posibles MCRM, tomando en cuenta las posibles rotaciones, reflexiones, traslaciones y contracciones a escala $1/2$, análogas a la Figura 1. Un análisis de la simetría y conexidad de ellas se puede ver en [12, sec. 5.3].

A estas MCRM, se les puede permitir algún estiramiento de la copia, así obtenemos gran variedad de sistemas iterados, con *atractores* muy variados: Helecho de Barnsley, Figura 3 parte (b); árbol, Figura 4. Estos *atractores*, a pesar de ser tan naturales, se obtienen de la misma forma que algunos “monstruos” como la curva de Koch, el triángulo de Sierpiński, el Twindragon etc.

Descomprimiendo imágenes con IFS's

Con este esquema de guardar solamente el sistema iterado de funciones y reproducir una aproximación de su *atractor* en el momento que deseamos, pueden presentarse algunos problemas en cuanto al tiempo de descompresión, aunque la razón de compresión es muy buena, en algunos de estos el tiempo de descompresión no lo es.

Para la descompresión del helecho de Barnsley se usó el sistema de funciones descrito en el Cuadro 2. Aplicando este sistema iterado de funciones a x_0 obtenemos 4 imágenes, es decir, $\{f_1(x_0), f_2(x_0), f_3(x_0), f_4(x_0)\}$, y al aplicar de nuevo el sistema obtenemos 16 imágenes, así sucesivamente. En este procedimiento el programa que se usó, [10], representó 47 puntos, pero muchos de ellos están cerca de su imagen, sobre todo los que se obtienen

CUADRO 2
Funciones generadoras

		<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Figura 2	f_1	0,849	0,037	-0,037	0,849	0,075	0,183
	f_2	0,197	-0,226	0,226	0,197	0,4	0,049
	f_3	-0,15	0,283	0,26	0,237	0,757	-0,084
	f_4	0	0	0	0,16	0,5	0
Figura 4	f_1	0,195	-0,488	0,344	0,443	0,4431	0,2452
	f_2	0,462	0,414	-0,252	0,361	0,2511	0,5692
	f_3	-0,058	-0,07	0,453	-0,111	0,5976	0,0969
	f_4	-0,035	0,07	-0,469	-0,022	0,4884	0,5069
	f_5	-0,637	0	0	0,501	0,8562	0,2513

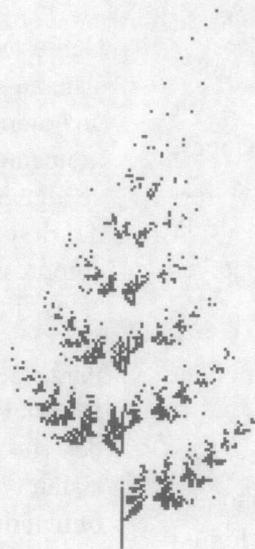


FIGURA 2
Helecho de Barnsley aplicando el IFS
descrito en el Cuadro 2

al aplicar f_4 y la resolución que se obtiene no es muy buena, Figura 2.

Pensando en este problema, podemos modificar la aplicación del sistema usando el conocido *Juego del Caos*, en donde en cada paso no aplicamos todas las funciones, sino solamente una función de manera aleatoria, todas con la misma probabilidad. Esto quiere decir que si x_0 es el punto inicial, tenemos que $x_n = f_{i_k}(x_{n-1})$ escogiendo i_k con igual probabilidad del conjunto $\{1,2,3,4\}$. Con este procedimiento se obtuvo la Figura 3 parte (a). Aun así, el Juego del Caos descrito aquí se puede modificar tomando diferentes probabilidades p_i para cada función f_i , este proceso recibe el nombre de *sistema iterado recurrente de funciones* (RIFS); una excelente exposición sobre la existencia, unicidad, convergencia y caracterización del *atractor* se puede encontrar en [2]. La aplicación a compresión de imágenes lo ilustra John Hart en [7], con algunos ejemplos. Un buen índice, para δ pequeño, está dado por

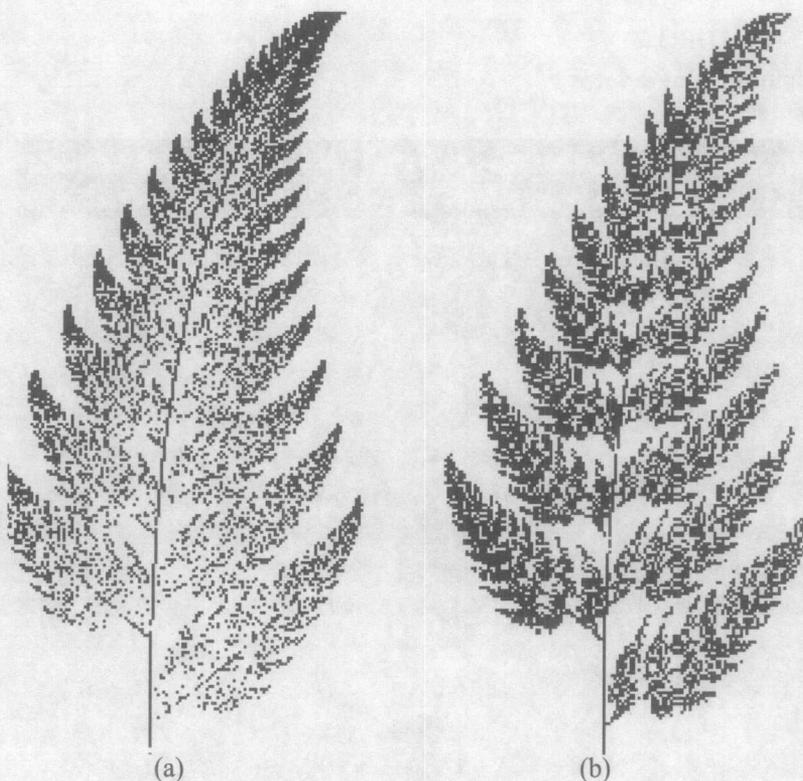


FIGURA 3
(a) Juego del Caos
(b) Juego del Caos con probabilidades diferentes

$$p_i = \frac{\max(\delta, |C_i|)}{\sum_{i=1}^n \max(\delta, |C_i|)}, \quad (5)$$

donde $|C_i| = \det C_i = \begin{vmatrix} a_i & b_i \\ c_i & d_i \end{vmatrix}$ y se tiene que

$\sum p_i = 1$. Con $\delta = 0,01$, ver [10], se obtuvo la Figura 3 parte (b), que tiene una mejor resolución con la misma cantidad de puntos que las Figuras 2 y 3 parte (b).

Otros ejemplos de descompresión de un RIFS, aplicado a un punto como imagen inicial, son el árbol de la Figura 4 izquierda, para el sistema de funciones descrito en el Cuadro 2 y la rama de la Figura 4 derecha.

Como un ejemplo interesante, consideremos como base el número complejo $b = -1+i$ que provee una representación

binaria de todos los números complejos [9], [11]. Al conjunto de todas estas fracciones se le llama usualmente Twindragon, Figura 5 izquierda. Una construcción geométrica de este se encuentra en [6]; sin embargo, para los efectos de este artículo, este conjunto es el *atractor* para el sistema iterado de funciones definido por $f_0(x)=b^{-1}x$ y $f_1(x)=b^{-1}+b^{-1}x$. Se puede cubrir el plano con una cantidad numerable de conjuntos

idénticos a este conjunto, de la forma $T+\omega$ donde ω es un entero gaussiano, y como todos los enteros gaussianos se pueden representar en esta base, se puede obtener un mosaico o “teselación” *fractal* del plano, Figura 5 derecha. Se dice fractal pues las fronteras de estos conjuntos son *fractales*, [1], [11].

Si ahora consideramos dígitos complejos, por ejemplo $D = \{0, 1, \omega, \omega^2\}$ donde $\omega = \frac{1}{2}(-1 + i\sqrt{3})$ y $b = -2$ da una representación de todos los números complejos. Al conjunto de todas las fracciones se le llama Fracciones de Eisenstein, Figura 6. Este conjunto es el atractor para el sistema iterado de funciones definido por $f_0(x) = -\frac{1}{2}x$, $f_1(x) = -\frac{1}{2}(1+x)$, $f_2(x) = -\frac{1}{2}(\omega+x)$ y $f_3(x) = -\frac{1}{2}(\omega^2+x)$.

Por otro lado, si a la máquina de fotocopiado de la que hablamos anteriormente, le agregamos el control del contraste y del brillo, entonces se pueden comprimir imágenes en una escala de gris. Para esto consideremos las aplicaciones afines v_i sobre \mathbb{R}^2 dadas en (2) con una dimensión más que controla el contraste y el brillo, a

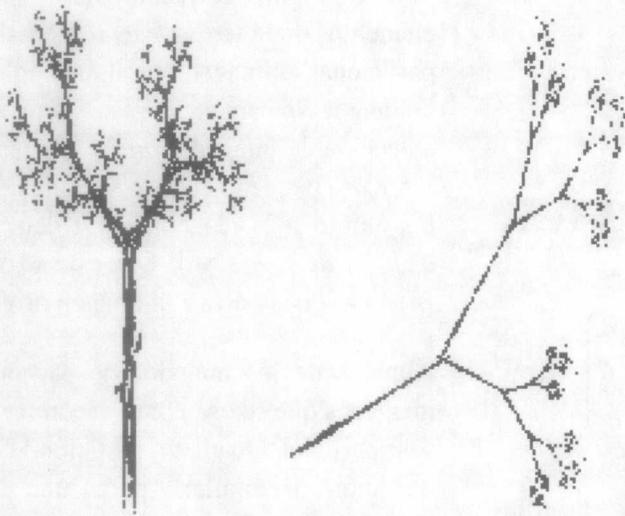


FIGURA 4

Un árbol y una rama como *atractores*

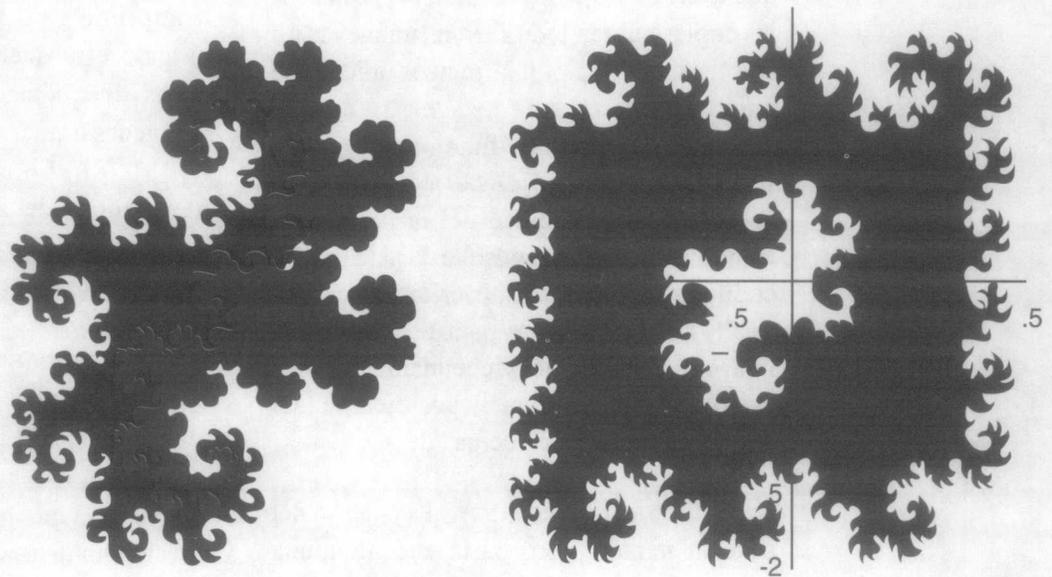


FIGURA 5
Twindragon

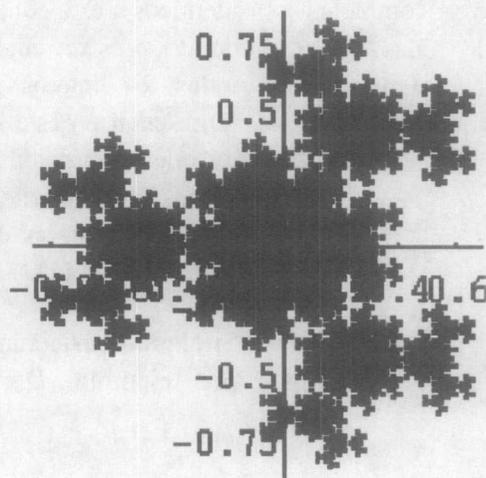


FIGURA 6
Fracciones de Eisenstein

estas aplicaciones las llamamos w_i ; es decir:

$$w_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ o_i \end{pmatrix} \quad (6)$$

donde s_i controla el contraste y o_i el brillo de w_i . Sean D_i y R_i el dominio y el rango de w_i , es decir, $v_i(D_i)=R_i$. Pedimos, además, que $\cup R_i=I^2$ y $R_i \cap R_j = \emptyset$ para $i \neq j$, es decir, las copias cubren todo I^2 y las imágenes no se traslapan. Sea f_0 una imagen inicial, se quiere encontrar el *atractor* $f=x_W$ del sistema iterado de funciones, o sea $W(f)=w_1(f) \cup w_2(f) \cup \dots \cup w_N(f)=f$. Para la existencia y unicidad del *atractor* para cualquier f_0 inicial consulte [1], [5]. Para codificar una imagen, necesitamos encontrar los D_i , R_i y w_i . Y para decodificar la imagen, aplicaríamos repetidamente W a cualquier imagen inicial, por ejemplo un punto hasta que esté "cerca" de x_W .

Lo óptimo sería encontrar el *atractor* del sistema iterado, pero basta encontrar una imagen f' , que en algún paso de la iteración

esté cerca de f , es decir, $d_{rms}(f',f)$ sea pequeño. Para esto es claro que se necesita minimizar la distancia entre $f \cap (R_i \times I)$ y $w_i(f)$, la distancia entre la porción de la imagen y su imagen, es decir, encontrar el mínimo de

$$d_{\text{sup}}(f \cap (R_i \times I), w_i(f)), \quad \forall i=1, \dots, N \quad (7)$$

así el problema se resume en encontrar los R_i y los D_i y por consiguiente los w_i .

De lo anterior, para codificar una imagen usando algún esquema que use los elementos de la teoría *fractal*, necesitamos particionar la imagen por alguna colección de rangos R_i ; para cada R_i se fija de alguna colección de imágenes un D_i que tenga un error *rms* pequeño cuando se aplica en R_i y por último de la ecuación (6) se determinan $a_i, b_i, c_i, d_i, e_i, f_i, s_i$ y o_i . Con esto se obtiene $W=\cup w_i$ que codifica la imagen original.

Algunos de los métodos de partición de imágenes que se usan para encontrar los R_i son: partición Quadtree, partición HV y la partición triangular, para una mejor comprensión de estos métodos puede consultar [5], [10]. En [5, Apéndice A] se da un programa, en lenguaje C, para comprimir y descomprimir imágenes usando el esquema HV. Existen, además, otros esquemas de compresión de imágenes usando la teoría fractal, [5, Caps. 9, 13], [8]. En la práctica, el que mejores resultados de compresión da es el particionamiento por el esquema HV, en donde una imagen comprimida en JPEG con una razón de compresión de 54,3:1 se puede comprimir usando el esquema HV con una razón de 58,1:1, y la resolución es mucho mejor, [5, Apéndice D]. En estos esquemas, la razón de compresión es muy buena sin embargo, el tiempo de descompresión no lo es; este es uno de los mayores problemas que enfrenta.

Bibliografía

- [1] Alfaro, M.; Murillo, M. & Soto A. Tesis de Licenciatura, Universidad de Costa Rica, 1997.
- [2] Barnsley, Michael F.; Elton John H. & Hardin Douglas P. "Recurrent Iterated Function Systems". *Constructive Approximation*, Vol. 5, pp. 3-31, 1989.
- [3] Edgar, Gerald A. *Measure, Topology, and Fractal Geometry*, Springer-Verlag, New York, 1990.
- [4] Falconer, Kenneth. *Fractal Geometry*, John Wiley & Sons, England, 1990.
- [5] Fisher, Yuval. *Fractal Image Compression*, Springer-Verlag, New York, 1994.
- [6] Gilbert, William J. "Fractal Geometry Derived from Complex Bases". *The Mathematical Intelligencer*, Vol. 4, pp. 78-86, 1982.
- [7] Hart, John C. "Fractal Image Compression and Recurrent Iterated Function Systems". *IEEE Computer Graphics*, Vol. 16, N° 4, 1996.
- [8] Jacquin, Arnaud E. "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations". *IEEE Transactions on Image Processing*, Vol. 1, N° 1, pp. 18-30, 1992.
- [9] Murillo T. Manuel & Soto A. Alberto, "Sobre Bases Complejas y Dragones". Memorias del V Encuentro Centroamericano de Investigadores en Matemática, pp. 230-235, Liberia, Costa Rica, 1997.
- [10] Murillo T. Manuel, "La Teoría Fractal Aplicada a la Compresión de Imágenes". Memorias del VI Encuentro Centroamericano de Investigadores en Matemática, pp. 164-172, Managua, Nicaragua, 1998.
- [11] Murillo T. Manuel & Soto A. Alberto, "Representaciones Numéricas en Bases Complejas". *Revista del Profesor de Matemática*, Año 4, N° 2, pp. 36-46, Chile, 1999.
- [12] Peitgen, H.-O.; Saupe, D. & Jürgens, H., *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, New York, 1992.
- [13] Proise, Jeff. "A Look Inside Bitmap Files". *PC Magazine*, Vol. 15, N° 21, pp. 322-324, diciembre 3, 1996.
- [14] Soto A., Alberto. "El Fascinante Triángulo de Sierpiński". Memorias del V ECADIM, Liberia, Costa Rica. 1997.