

Design and Development of an Interactive Software for the Teaching-Learning of Fractal Compression of Images

Edgar Miguel Vargas Chaparro, M.Sc.¹, Yulian Cedenoo Ocampo, M.Sc. (c).¹

¹Universidad Nacional de Colombia, Colombia, emvargasc@unal.edu.co, ycedenoo@unal.edu.co

Abstract— In the study of courses with complex topics, as is the case of data compression, students experience difficulty in their learning due to the abstract nature of these theories and the absence of interactive digital resources. Fractal image compression seeks to deduce a set of functions, called iterated function system (IFS), which by iterations reconstructs the image to be compressed; In this way, compression levels are achieved that are not achieved using other techniques, because the image is considered compressed in the description of the functions that generate it, this is what is stored, and to recover the image, the functions are simply iterated. This article describes the design and development of interactive software that supports the teaching-learning of the fundamental concepts of fractal compression of images: contractive affine transformations, iterated function systems (IFS), IFS encoding (compression) and decoding (decompression) IFS.

Keywords — Fractal compression, images, interactive software, teaching, learning.

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2022.1.1.429>

ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

Diseño y Desarrollo de un Software Interactivo para la Enseñanza-Aprendizaje de la Compresión Fractal de Imágenes

Edgar Miguel Vargas Chaparro, M.Sc.¹, Yulian Cedenoo Ocampo, M.Sc. (c).¹

¹Universidad Nacional de Colombia, Colombia, emvargasc@unal.edu.co, ycedenoo@unal.edu.co

Abstract— En el estudio de cursos con temáticas complejas, como es el caso de la compresión de datos, los estudiantes experimentan dificultad en su aprendizaje debido a lo abstracto de dichas teorías y la ausencia de recursos digitales interactivos. En la compresión fractal de imágenes se busca deducir un conjunto de funciones, llamado sistema de funciones iteradas (IFS), que por iteraciones reconstruya la imagen que se desea comprimir; así se alcanzan niveles de compresión que no se logran utilizando otra técnica, porque se considera comprimida la imagen en la descripción de las funciones que la generan, esto es lo que se almacena, y para recuperar la imagen simplemente se iteran las funciones. El presente artículo describe el diseño y desarrollo de un software interactivo que apoya la enseñanza-aprendizaje de los conceptos fundamentales de la compresión fractal de imágenes: transformaciones afines contractivas, sistemas de funciones iteradas (IFS), codificación (compresión) IFS y decodificación (descompresión) IFS.

Palabras Claves—Compresión fractal, imágenes, software interactivo, enseñanza, aprendizaje.

I. INTRODUCCIÓN

La compresión de datos es el proceso de transformar una cadena de datos de entrada (los datos originales) en otra cadena de datos (la salida o cadena comprimida), cuyo tamaño es menor al de la cadena de entrada. Se interpreta como un proceso que elimina la complejidad innecesaria (redundancia) de la información y así maximiza la sencillez, conservando su poder descriptivo y no redundante [1].

La esencia de la compresión de datos consiste en entender y modelar la fuente de información, y luego codificar los datos teniendo como referencia el modelo de la fuente. Esto es exactamente lo que hacen los esquemas de análisis/síntesis. Estos esquemas se basan en la disponibilidad de un modelo paramétrico para la generación de salida de la fuente. Cuando tal modelo existe, el transmisor analiza la salida de la fuente y extrae los parámetros del modelo, los cuales son transmitidos al receptor. El receptor utiliza el modelo, junto con los parámetros transmitidos, para sintetizar una aproximación a la salida de la fuente. La compresión fractal de imágenes encaja en estos esquemas, ya que también utiliza la idea de representar los datos con un proceso para generarlos, y el receptor regenera la salida de la fuente utilizando las “instrucciones” del transmisor; es decir, lo que se almacena o transmite no son las muestras de la salida de la fuente, sino un

método para sintetizar la salida [2].

En el estudio de temáticas abstractas, como es el caso de la compresión de datos, los estudiantes encuentran dificultad en su aprendizaje debido a la ausencia de recursos educativos digitales interactivos [3]. Este hecho se evidencia en [4] y [5], donde se determina que los estudiantes tienen dificultades a la hora de comprender los conceptos y aplicaciones de dicha temática. Existe una gran cantidad de contenidos digitales para la enseñanza y aprendizaje de los conceptos de la compresión de datos a diferentes niveles de profundidad, muchos de los cuales están disponibles principalmente en formatos de texto tales como PDF, y algunos pocos, están plasmados en videos ilustrativos que ofrecen de manera no formal contenidos, lo cual demuestra la ausencia de material pedagógico interactivo para la enseñanza de la compresión de datos, siendo ésta una temática de gran importancia considerando los grandes volúmenes de datos que se manejan en la sociedad actual.

Por lo anterior, en el presente trabajo se describe el análisis, diseño y desarrollo de un software interactivo para la enseñanza-aprendizaje de los conceptos fundamentales de la compresión fractal de imágenes: transformaciones afines contractivas, sistemas de funciones iteradas (IFS), codificación (compresión) IFS y decodificación (descompresión) IFS.

Para cumplir este objetivo se hace uso de una metodología inspirada en el modelo de diseño instruccional ADDIE, cuyo nombre es un acrónimo de los pasos clave que conforman este modelo, a saber [6]: Análisis, Diseño, Desarrollo, Implementación y Evaluación. Dado el alcance de este artículo no se expondrán los pasos de Implementación y Evaluación.

Este artículo está organizado de la siguiente manera: en la sección II se exponen los aspectos disciplinares, o contenido de enseñanza, del software interactivo. En la sección III se describe el análisis del software, y en la sección IV se presenta su diseño. En la sección V se expone el desarrollo del software, el prototipo obtenido y los resultados de las pruebas funcionales al prototipo. Finalmente, la sección VI presenta las conclusiones y el trabajo futuro que se deriva del presente trabajo.

II. ASPECTOS DISCIPLINARES

Los fractales han sido populares desde la década de los 70's y tienen muchas aplicaciones. Una de estas aplicaciones, relativamente poco utilizada, es la compresión de datos. La aplicación de los fractales en la compresión de imágenes se efectúa mediante *sistemas de funciones iteradas (IFS)*.

El principio de la compresión fractal de imágenes, es considerar que, dado que la forma que se quiere representar es una imagen, lo que se busca es deducir el conjunto de funciones, llamado código IFS, que por iteraciones reconstruya esta imagen. Existen sistemas en la actualidad que aplican esta técnica y que pueden comprimir una imagen hasta 10.000 a 1; estos niveles de compresión que no se logran utilizando otra técnica, son posibles porque básicamente se considera comprimida la imagen en la descripción de las funciones que la generan, esto es lo que se almacena, y para recuperar la imagen simplemente se iteran las funciones y así se reconstruye la imagen almacenada.

A. Transformaciones afines contractivas

Las transformaciones afines se pueden describir como combinaciones de rotaciones, escalados, esfuerzos cortantes y traslaciones de los ejes coordenados en un espacio n -dimensional. Para los propósitos de este trabajo se considerarán solo transformaciones afines en \mathbb{R}^2 .

Una transformación afín en \mathbb{R}^2 está compuesta por una transformación lineal y una traslación. La forma general para una transformación afín es:

$$w \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \quad (1)$$

donde los coeficientes a, b, c, d, e y f son números reales. La transformación lineal está representada por la matriz que se muestra a continuación:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad (2)$$

con la cual se puede realizar escalado en cada eje, rotación, esfuerzo cortante en el eje x y en el eje y de la imagen. La traslación desplaza el origen a las coordenadas (e, f) (ver Fig. 1).

Un ejemplo en dos dimensiones de una transformación afín es:

$$w(x,y)=(1/2x + 1/4 y + 1, 1/4 x + 1/2 y + 2) \quad (3)$$

Esta transformación mueve el punto $(0,0)$ a $(1,2)$ y mueve $(-1,0.5)$ a $(0.625,2)$. Se nota esta transformación con w . $w(S)$ denota la imagen por w del conjunto de puntos S .

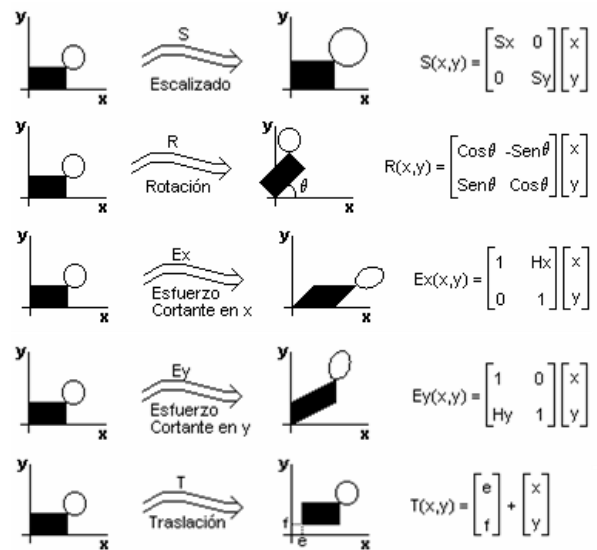


Fig. 1 Transformaciones afines en \mathbb{R}^2 . Tomada de [7]

Es fácil ver que las transformaciones descritas anteriormente pueden cambiar longitudes y ángulos. El escalado cambia las longitudes de los objetos, la rotación y los esfuerzos cortantes cambian los ángulos. Sin embargo, una propiedad que se conserva es el paralelismo de las líneas: un par de líneas paralelas permanecerán así después de aplicar cualquiera de estas transformaciones. Una transformación que conserva el paralelismo se llama *afín*.

En la Fig. 2 se muestra qué le hace w a una imagen de una cara sonriente F , que yace en el plano xy . El resultado es una nueva cara comprimida $w(F)$. La transformación afín deformó y trasladó la cara. Nótese que los ojos en la cara transformada $w(F)$ están más cerca que en la imagen F . Se dice entonces que la transformación es *contractiva* porque ella siempre mueve los puntos de tal forma que se encuentren más cerca.

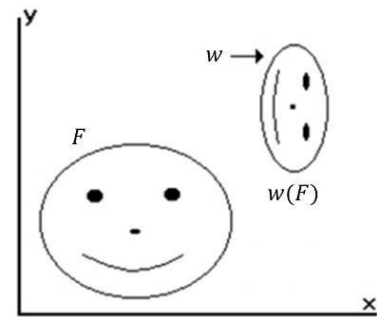


Fig. 2 Transformación afín contractiva.

La conclusión clave de esta sección es que cualquier transformación afín bidimensional puede ser completamente especificada por solo seis números.

B. Sistemas de funciones iteradas (IFS)

Un sistema de funciones iteradas (IFS) es una colección de funciones que se aplican iterativamente sobre un conjunto inicial. Para propósitos del presente trabajo se consideran IFS de la forma

$$\{w_i: i=1,2,\dots,N\}, \tag{4}$$

donde las w_i son transformaciones afines contractivas en \mathbb{R}^2 . La descripción de un IFS se llama un *código IFS*, el cual puede ser almacenado en una tabla que contiene los coeficientes de las transformaciones.

El siguiente es un ejemplo de un IFS en \mathbb{R}^2 compuesto de tres transformaciones afines contractivas y su correspondiente código IFS (Ver Tabla I):

$$\begin{aligned} w_1 \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ w_2 \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} \\ w_3 \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} \end{aligned} \tag{5}$$

TABLA I
CÓDIGO IFS DE TRANSFORMACIONES

w	a	b	c	d	e	f
w_1 :	1/2	0	0	1/2	0	0
w_2 :	1/2	0	0	1/2	0	1/2
w_3 :	1/2	0	0	1/2	1/2	0

Las tres transformaciones reducen una imagen a la mitad de su tamaño original; además, w_2 y w_3 colocan dos copias de la imagen en los desplazamientos relativos (0, 1/2) y (1/2, 0), como se muestra en la Fig. 3.

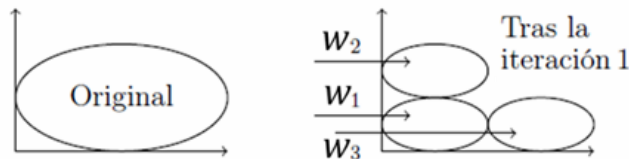


Fig. 3 Transformaciones afines generadas por el IFS de la Ecuación (5)

La Fig. 4 muestra la aplicación del IFS de ejemplo a tres imágenes iniciales distintas: un cuadrado, un círculo y una Z. Cuando el IFS se aplica algunas veces a una imagen, todavía es posible discernir las copias individuales de la imagen original. Sin embargo, cuando se aplica varias veces, el resultado es el triángulo de Sierpiński, por lo que el resultado final no depende de la forma de la imagen inicial. La imagen final obtenida en el límite, después de aplicar un IFS un número infinito de veces, se llama *atractor del IFS* [1].

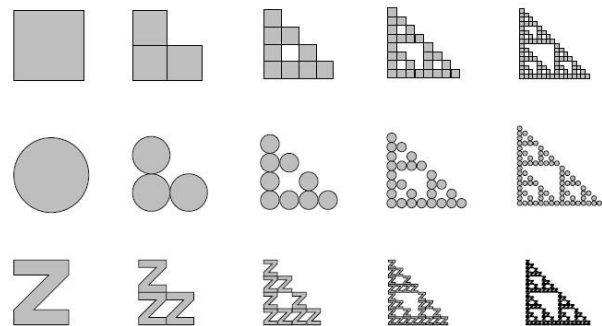


Fig. 4 Aplicación del IFS de ejemplo a tres imágenes iniciales distintas [1]

El resultado de cada iteración del IFS es una imagen que contiene todas las imágenes de todas las iteraciones anteriores. Si se aplica el mismo IFS muchas veces, es posible ampliar el resultado para magnificarlo, y todavía ver los detalles de las imágenes originales. En principio, si se aplica el IFS un número infinito de veces, el resultado final mostrará los detalles a cualquier aumento, será un *fractal*. Los objetos fractales con frecuencia son estadísticamente *autosimilares* o invariantes con el cambio de escala, es decir que al hacer ampliaciones de un objeto fractal se obtienen objetos tan complejos como el original (partes del objeto son idénticas a todo el objeto).

Matemáticamente, el efecto conjunto de todas las transformaciones que conforman el IFS se puede representar como la transformación $W(\cdot)$, que puede verse como la unión de las transformaciones w_i :

$$W = \bigcup_i w_i \tag{6}$$

Entonces la transformación W tiene un único punto fijo A tal que:

$$A = W(A) = \bigcup_{i=1}^N w_i(A) \quad \text{y} \quad A = \lim_{n \rightarrow \infty} W^n(I_0) \tag{7}$$

para toda imagen inicial I_0 . El punto fijo A es el atractor del IFS o fractal determinístico generado por el IFS.

C. Codificación IFS

Lo anterior sugiere una posible manera de comprimir imágenes reales: si se logra dividir una imagen en partes de tal manera que cada parte sea idéntica (o al menos muy cercana) a la imagen completa, entonces se puede comprimir la imagen mediante un IFS, averiguando las transformaciones que transforman la imagen completa en cada parte. La transformación para cada parte se expresa mediante unos pocos números, que se convierten en el *stream* comprimido.

Lo anterior está formalizado en el Teorema del Collage de Barnsley [8], que establece que para encontrar un IFS cuyo atractor es “cercano” a una imagen I según cierta métrica (esto es su atractor se “ve” como I), se debe encontrar un conjunto de transformaciones contractivas tal que la unión (collage) de las imágenes de I bajo las transformaciones sea cercana a I . En la Fig. 5 se ve un collage de una hoja de olmo, se tienen cinco transformaciones afines de la hoja de olmo que unidas se ven como la hoja original, entonces con esas cinco transformaciones se puede almacenar la forma de la hoja de olmo.

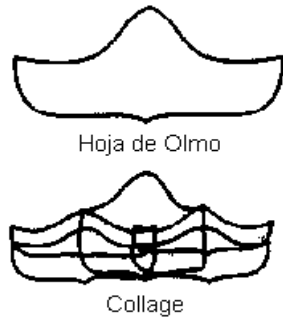


Fig. 5 Hoja de olmo y collage.

El codificador funcionará de la siguiente manera [9]:

1. Establecer las imágenes contractivas $\{w_i; i=1,2,\dots,N\}$ tal que $I \approx W(I) = \bigcup_{i=1}^N w_i(I)$.
3. Transmitir (o almacenar) los coeficientes que identifican el código IFS correspondiente.

D. Decodificación IFS (Algoritmo de iteración determinística)

El decodificador usará los coeficientes del código IFS para reensamblar W y reconstruir su punto fijo, la imagen (el atractor A o fractal generado por el IFS) a partir de una imagen arbitraria I_0 [7]:

1. Hacer:

- $T = \emptyset$
 - Para i desde 1 hasta N :
 - o $T = T \cup w_i(A)$
 - $A = T$
 - Despliegue A
- Mientras (no se interrumpa)

Este algoritmo parte de una imagen arbitraria I_0 y aplica a esta imagen cada una de las transformaciones w_i , uniendo tales imágenes en una sola imagen llamada T ; luego repite el mismo proceso sobre T en cada iteración. Este algoritmo es lento debido a que se calcula la imagen de cada punto para cada transformación, en cada iteración. Además, se debe disponer de memoria para almacenar dos imágenes, la que se tiene en cada momento y la que se está componiendo en la respectiva iteración.

E. Decodificación IFS (Algoritmo de iteración aleatoria - Chaos Game)

Un IFS con probabilidad es un IFS $\{w_i; i=1,2,\dots,N\}$ en el cual se asigna a cada transformación contractiva w_i una probabilidad de aplicación p_i , tal que:

$$\sum_{i=1}^N (p_i) = 1, \quad p_i > 0 \quad i = 1, 2, \dots, N \quad (8)$$

Un IFS con probabilidad se nota

$$\{w_i; i = 1, 2, \dots, N; p_i; i = 1, 2, \dots, N\} \quad (9)$$

Partiendo de un punto arbitrario x_0 , el algoritmo de generación mediante iteración aleatoria es [7]:

1. $x = x_0$
 2. Hacer:
 - Seleccionar un número aleatorio k en $\{1, 2, 3, \dots, N\}$, con p_i la probabilidad de que k sea i .
 - $x = w_k(x)$
 - Dibujar x
- Mientras (no se interrumpa)

Este algoritmo parte de un punto arbitrario x_0 , escoge de manera aleatoria una de las transformaciones de acuerdo con las probabilidades asignadas a ellas, calcula el punto imagen de x_0 por la transformación seleccionada, lo llama x y lo dibuja. Repite este proceso sobre x . La sucesión de puntos x forma una imagen que se aproxima al fractal. Este algoritmo es muchas veces más rápido que el de iteración determinística debido a que se calcula solo la imagen de un punto en cada iteración. Además, requiere memoria para almacenar solo una imagen que contiene los puntos que se han generado. Este algoritmo se llama en la literatura el juego del caos (chaos game) debido a que se están simulando sistemas dinámicos caóticos que tienen un atractor extraño.

El resultado de la aplicación del algoritmo siempre genera el mismo objeto fractal (al considerar infinitas iteraciones), es decir la componente aleatoria del algoritmo no afecta el fractal descrito. La convergencia del algoritmo de iteración aleatoria es independiente de la asignación de probabilidades [10].

III. ANÁLISIS DEL SOFTWARE INTERACTIVO

La metodología seleccionada para el desarrollo del software interactivo está inspirada en el modelo de diseño instruccional ADDIE, cuyo nombre es un acrónimo de los pasos clave que conforman este modelo: Análisis, Diseño, Desarrollo, Implementación y Evaluación [11].

En esta sección se describe el análisis, que consiste en la caracterización del público objetivo, la definición de los objetivos de aprendizaje y el contenido de enseñanza, la descripción del enfoque pedagógico y una descripción general de los requerimientos del software.

A. Caracterización del Público Objetivo

Este proyecto tuvo como público objetivo dos grupos de la asignatura “Teoría de la información y sistemas de comunicación”, cada uno conformado por aproximadamente 34 estudiantes de pregrado de la Universidad Nacional de Colombia. Como parte de las temáticas relacionadas a la teoría de la información, se estudian los conceptos básicos de la compresión de datos. Dicha asignatura está adscrita a la Facultad de Ingeniería de la universidad, motivo por el cual los estudiantes que la inscriben pertenecen a carreras exclusivas del área de ingeniería (especialmente Ingeniería de Sistemas y Computación), cursan entre séptimo y décimo semestre, y tienen edades entre los 20 y 33 años.

Como resultado de la observación del público objetivo se identificó que:

- Los estudiantes no contaban con un recurso educativo digital que les permitiera comprender y visualizar más fácilmente la compresión fractal de imágenes.
- Todas las clases teóricas y prácticas se realizaban en el salón asignado a la asignatura, que cuenta con computadores de escritorio en los que se puede ejecutar el software interactivo.

B. Definición de los Objetivos de Aprendizaje

- Experimentar la aplicación de transformaciones afines contractivas a imágenes monocromáticas (siluetas o contornos), con el fin de entender este tipo de transformaciones e identificar sus efectos.
- Identificar la relación existente entre una colección de transformaciones afines contractivas, un sistema de funciones iteradas (IFS) y un código IFS.
- Comprender el proceso de codificación (compresión) IFS logrado mediante la aplicación del Teorema del Collage de Barnsley.
- Comprender el proceso de decodificación (descompresión) IFS generado por el algoritmo de iteración aleatoria (chaos game).

C. Definición del Contenido de Enseñanza

Con base en los objetivos de aprendizaje definidos se establece el contenido de enseñanza, que se corresponde con los aspectos disciplinares de la sección I y está conformado por las siguientes temáticas:

- Transformaciones afines contractivas.
- Sistemas de funciones iteradas (IFS).
- Codificación IFS.
- Decodificación IFS.

D. Descripción del Enfoque Pedagógico

Debido a sus características particulares, un software interactivo es considerado como una tecnología que puede apoyar un enfoque de aprendizaje constructivista. Dicho enfoque se basa en la construcción de conocimiento a partir de actividades basadas en experiencias, considera al estudiante como un ser autónomo, capaz de guiar y evaluar su propio

aprendizaje, mientras que al docente lo concibe como un facilitador y orientador del proceso formativo.

De los múltiples enfoques constructivistas, el aprendizaje inmersivo se adecúa a las características del presente proyecto, ya que este tipo de aprendizaje permite aprovechar los beneficios de nuevas tecnologías para crear experiencias de aprendizaje, transmitir las enseñanzas, y crear ambientes de concentración y contacto con la temática que se enseña. De este modo se genera la posibilidad de que el estudiante haga un mayor uso de sus sentidos para adquirir los nuevos conocimientos [12].

E. Requerimientos Iniciales de Alto Nivel del Software Interactivo

El “Software Interactivo para Compresión Fractal de Imágenes” es un aplicativo gráfico interactivo. En este aplicativo el usuario puede realizar la compresión fractal de una imagen monocromática y posteriormente su descompresión.

Para comprimir una imagen, el aplicativo debe permitir recubrirla con un conjunto de subimágenes. Cada elemento del conjunto se obtiene mediante la aplicación de transformaciones a un duplicado de la imagen original. Este conjunto de subimágenes está representado por un código IFS que es hallado por el aplicativo y que posteriormente puede ser utilizado para regenerar la imagen, puesto que el código es la representación de la imagen comprimida.

Un código IFS consiste en una serie de números reales y por lo tanto puede ser almacenado en un archivo de texto. El aplicativo permite regenerar una imagen descomprimiendo el código IFS hallado en la sesión corriente o cargando un código IFS previamente hallado. El usuario también puede crear un código IFS utilizando un editor de texto y creando un archivo de texto que contenga una serie de números reales dispuestos en cierto orden; posteriormente, puede cargar dicho archivo y proceder a la descompresión del código.

Las transformaciones que están disponibles en el aplicativo son:

1) *Escalado*: Es una transformación lineal que altera el tamaño de una imagen, aplicando un factor en x que altera el tamaño de la imagen en la dirección del eje x , y un factor en y que altera el tamaño de la imagen en la dirección del eje y .

2) *Rotación*: Es una transformación lineal que rota los puntos de una imagen un ángulo, conocido como ángulo de rotación.

3) *Esfuerzo Cortante en X*: Es una transformación lineal que distorsiona una imagen produciendo un efecto de doblez o de corte, aplicando un factor que afecta los valores de las coordenadas en el eje x .

4) *Esfuerzo Cortante en Y*: Es una transformación lineal que distorsiona una imagen produciendo un efecto de doblez o de corte, aplicando un factor que afecta los valores de las coordenadas en el eje y .

5) *Traslación*: Una traslación es el movimiento en línea recta de una imagen de una posición a otra. Los puntos de la

imagen se trasladan agregando distancias de traslación en el eje x y en el eje y .

Se pueden realizar las transformaciones especificando parámetros en una caja de diálogo o utilizando el mouse, caso en el cual se debe habilitar la transformación que se desea haciendo click en el botón correspondiente de la barra de herramientas de transformaciones.

Al comenzar una sesión con el software interactivo se mostrará su ventana de aplicativo con los elementos típicos de Windows asociados a este tipo de ventana, los cuales son: Cuadro del Menú Control, Barra de Título, Título de la Ventana, Borde de la Ventana, Botón Minimizar, Botón Maximizar, Barra de Menús, Barras de Herramientas, Esquina de la Ventana, Punto de Inserción, Puntero del Mouse, Área de Trabajo del Aplicativo.

IV. DISEÑO DEL SOFTWARE INTERACTIVO

Con base en los resultados obtenidos en la etapa de análisis, se procede a especificar el diseño del aplicativo a desarrollar, conformado por: el diseño de la arquitectura de alto nivel y el diseño de la interfaz de usuario.

A. Diseño de la Arquitectura de Alto Nivel

En la Fig. 6, se ilustra el diseño didáctico del software interactivo, constituido por actividades de aprendizaje relacionadas al contenido de enseñanza, presentadas al estudiante mediante un enfoque de aprendizaje inmersivo, considerando que dicho contenido cumpla con las 10 heurísticas de usabilidad para diseño de interfaces de usuario de Nielsen [13] y con los 12 principios de aprendizaje por multimedia de Mayer [14], procurando así el diseño de una experiencia de aprendizaje satisfactoria para el estudiante.

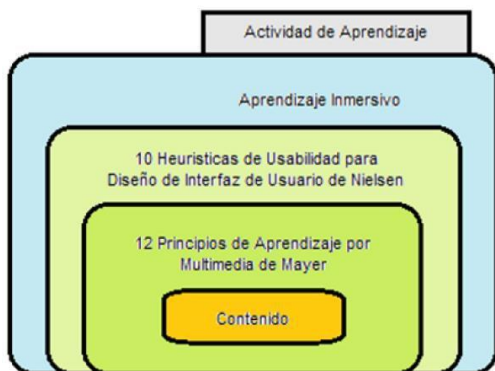


Fig. 6. Diagrama de diseño.

Con respecto a los estados, y las transiciones entre ellos, que se requerirán para la programación de la lógica del aplicativo, se propone el diagrama mostrado en la Fig. 7, a modo de diseño de alto nivel, donde se muestran las transiciones entre los estados del aplicativo.

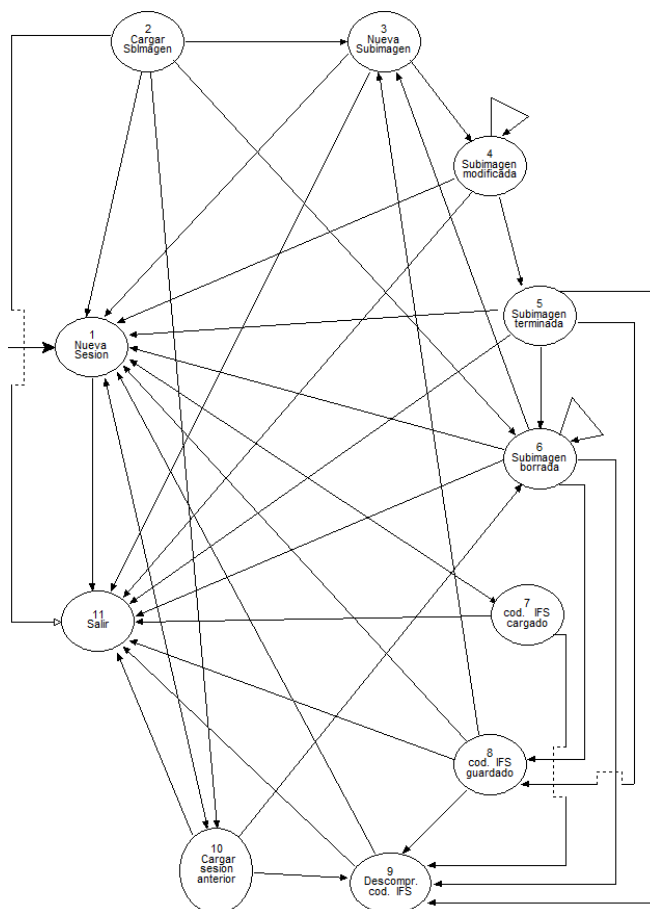


Fig. 7. Diagrama de transiciones entre estados del aplicativo.

B. Diseño de la Interfaz de Usuario

A continuación, se hace una descripción de la Barra de Menús requerida para que el aplicativo tenga la funcionalidad deseada (ver Fig. 8). La Barra de Menús del aplicativo está conformada por:

- 1) *Menú File*. Este menú permite la manipulación de archivos con extensiones .bmp (imagen) y .ifs (código IFS).
- 2) *Menú View*. Con los comandos de este menú el usuario puede personalizar la forma de presentación del aplicativo sobre la pantalla.
- 3) *Menú Transformation*. En este menú se encuentran los comandos que ejecutan sobre una imagen las transformaciones disponibles en el aplicativo.
- 4) *Menú Subimage*. En este menú se encuentran los comandos de creación, edición y finalización de una subimagen.
- 5) *Menú Algorithms*. Este menú contiene opciones que permiten ejecutar ciertos algoritmos que son de interés para el usuario:

- El algoritmo que halla las distancias de Hausdorff [8] entre la imagen original y el conjunto de subimágenes hallado, y entre la imagen original y el atractor del código IFS. Si dos imágenes están a una distancia de Hausdorff

h , significa intuitivamente que si se expande cualquiera de ellas h unidades en todas las direcciones cubrirá completamente a la otra.

- El algoritmo conocido como el “*juego del caos*”, el cual descomprime el Código IFS creado en la sesión actual.

6) *Menú Options*. Permite al usuario definir opciones del ambiente de trabajo del aplicativo (los colores de las imágenes que se trabajan).

7) *Menú Help*. Permite al usuario consultar sus dudas respecto al manejo del aplicativo. Este menú contiene los elementos básicos y opera de forma similar a cualquier Ayuda de un aplicativo estándar de Windows.

File	View	Transformation	Subimage	Algorithms	Options	Help
Load Bmp...	Coordinates	Scale...	New Subimage	Hausdorff...	Colors...	Contents
Load Ifs...	Tool Bars...	Rotation...	Edit Subimage...	Uncompress...		About...
Uncompress Ifs...	Matrix	Stretch a In X... In Y...	End Subimage			
Save		Translation...				
Save As...						
New Session						
Exit						

Fig. 8. Barra de Menús del aplicativo.

V. DESARROLLO DEL SOFTWARE INTERACTIVO

En esta etapa se desarrolló el software interactivo con base en los resultados obtenidos en las etapas anteriores de análisis y diseño, haciendo uso de técnicas de *desarrollo ágil de software* (desarrollo de software iterativo e incremental en el que los requerimientos del proyecto y su solución evolucionan y varían con el tiempo). Por lo tanto, en esta sección se expone la evidencia del prototipo inicial desarrollado y se describen los resultados de las pruebas de funcionalidad realizadas al prototipo.

El aplicativo desarrollado es un software interactivo ya que acepta entradas del usuario durante su ejecución, mediante el uso de una interfaz gráfica que utiliza un conjunto de imágenes y objetos gráficos para representar la información y las acciones disponibles. Debido a estas características, el software desarrollado puede apoyar un enfoque de aprendizaje constructivista, basado en la construcción de conocimiento a partir de actividades experienciales, permitiendo al estudiante autonomía en su proceso de aprendizaje.

A. Prototipo del Software Interactivo

En la Fig. 9, se observa la ventana principal del aplicativo con los tres componentes principales que permiten que tenga la funcionalidad esperada: Barra de Menús, Barra de Herramientas Estándar, Barra de Herramientas de Transformaciones y Área de Trabajo del Aplicativo.

La Barra de Herramientas Estándar (ver Fig. 10) se presenta por defecto al usuario cuando entra al aplicativo. Contiene botones para ejecutar rápidamente los comandos más comunes de la Barra de Menús, como son: File|Load Bmp..., File|Load Ifs..., File|Save, File|Exit, Subimage|New Subimage, Subimage|Edit Subimage..., Subimage|End Subimage,

Algorithms|Hausdorff..., Algorithms|Uncompress... y Help|Contents.

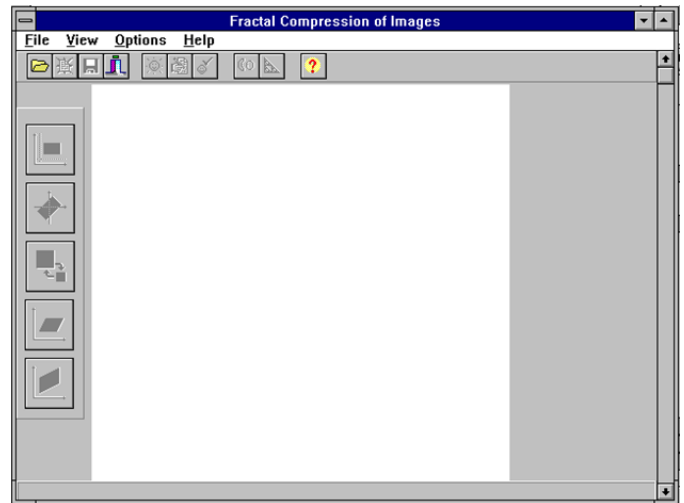


Fig. 9. Ventana principal del aplicativo.



Fig. 10. Barra de Herramientas Estándar.

Si se desea ocultar la Barra de Herramientas Estándar se debe ejecutar el comando View|Tool Bars... La caja de diálogo “ToolBars” aparecerá. Haciendo click en la caja de verificación “Standard”, de tal forma que quede sin marca, y haciendo click en el botón OK, se ocultará la Barra de Herramientas Estándar.

La Barra de Herramientas de Transformaciones (ver Fig. 11) se presenta por defecto al usuario cuando entra al aplicativo. Contiene botones para aplicar a una imagen en las transformaciones disponibles en el aplicativo a través del mouse, así:



Fig. 11. Barra de Herramientas de Transformaciones.

- *Escalado*: Hacer click en el botón de escalado de la Barra de Herramientas de Transformaciones. Escalar la subimagen arrastrando alguno de los puntos de control que aparecen.

- *Rotación*: Hacer click en el botón de rotación de la Barra de Herramientas de Transformaciones. Rotar la subimagen arrastrando el punto de control que aparece.

- *Esfuerzo Cortante en X*: Hacer click en el botón de esfuerzo cortante en x de la Barra de Herramientas de Transformaciones. Arrastrar el punto de control que aparece.

- *Esfuerzo Cortante en Y*: Hacer click en el botón de esfuerzo cortante en y de la Barra de Herramientas de Transformaciones. Arrastrar el punto de control que aparece.

- *Traslación*: Hacer click en el botón de Traslación de la Barra de Herramientas de Transformaciones. Arrastrar la subimagen a la posición deseada.

Si se desea ocultar la Barra de Herramientas de Transformaciones se debe ejecutar el comando View|Tool Bars... La caja de diálogo "ToolBars" aparecerá. Haciendo click en la caja de verificación "Transformations", de tal forma que quede sin marca, y haciendo click en el botón OK, se ocultará la Barra de Herramientas de Transformaciones.

B. Resultados de las Pruebas al Prototipo

A continuación, se muestran los resultados de una serie de pruebas funcionales realizadas al prototipo, para cada una de ellas se tiene la imagen a comprimir, los códigos IFS asociados a dos conjuntos de subimágenes diferentes con los que se recubrió la imagen, y el atractor generado por dichos códigos.

1) PRUEBA No. 1: Compresión de un cuadrado

En la Fig. 12 se muestra la imagen a comprimir, consistente en un cuadrado de color negro. En la Tabla II está el primer código IFS para el cuadrado, correspondiente al conjunto de subimágenes mostrado en la Fig.13. La Fig. 14 muestra el atractor generado por este primer código IFS.

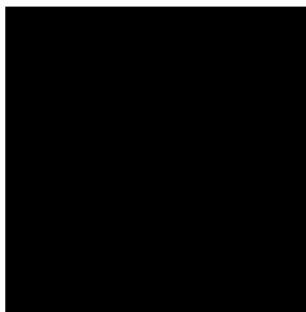


Fig. 12. Imagen a comprimir prueba No. 1.

TABLA II
PRIMER CÓDIGO IFS DE LA PRUEBA No.1

w	a	b	c	d	e	f	$prob.$
w_1	1	0	0	0.5304	0	36.6263	0.4143
w_2	0.6033	0	0	0.6174	27.7693	117.8435	0.2909
w_3	0.6157	0	0	0.6130	119.9014	119.1836	0.2948

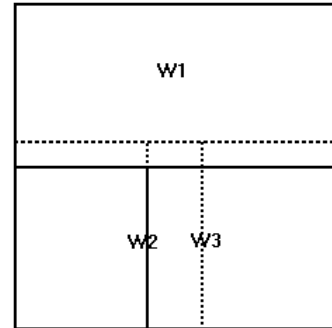


Fig. 13. Conjunto de subimágenes del primer código IFS.

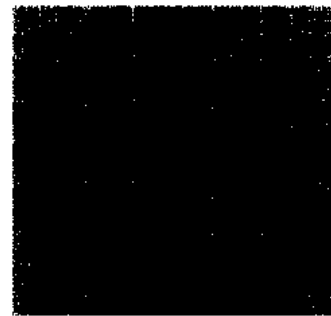


Fig. 14. Atractor generado por el primer código IFS de la prueba No. 1.

En la Tabla III está el segundo código IFS para el cuadrado, correspondiente al conjunto de subimágenes mostrado en la Fig.15. La Fig. 16 muestra el atractor generado por este segundo código IFS.

TABLA III
SEGUNDO CÓDIGO IFS DE LA PRUEBA No.1

w	a	b	c	d	e	f	$prob.$
w_1	0.5	0	0	1	35	0	0.5
w_2	0.5	0	0	0.5	156	39	0.25
w_3	0.5	0	0	0.5	156	154	0.25

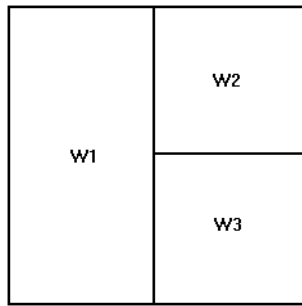


Fig. 15. Conjunto de subimágenes del segundo código IFS.



Fig. 16. Atractor generado por el segundo código IFS de la prueba No. 1.

2) PRUEBA No. 2: Compresión de una hoja

En la Fig. 17 se muestra la imagen a comprimir, consistente en una hoja de color negro. En la Tabla IV está el primer código IFS para la hoja. La Fig. 18 muestra el atractor generado por este primer código IFS.

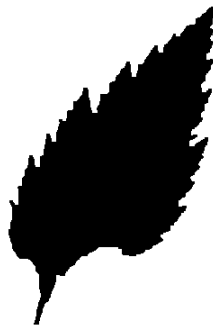


Fig. 17. Imagen a comprimir prueba No. 2.

TABLA IV
PRIMER CÓDIGO IFS DE LA PRUEBA No.2

w	a	b	c	d	e	f	prob.
w ₁	0.5256	-0.1621	0.1621	0.5256	140.8432	103.7557	0.2408
w ₂	0.5677	0.1941	-0.1941	0.5677	-19.3751	154.4923	0.2865
w ₃	0.2000	-0.0545	0.0000	0.5000	105.4180	169.5000	0.0796
w ₄	0.1002	-0.0701	0.0207	0.4457	111.0917	206.3631	0.0367
w ₅	0.6706	0.0000	0.0000	0.6678	102.1434	4.0508	0.3564



Fig. 18. Atractor generado por el primer código IFS de la prueba No. 2.

En la Tabla V está el segundo código IFS para la hoja. La Fig. 19 muestra el atractor generado por este segundo código IFS.

TABLA V
SEGUNDO CÓDIGO IFS DE LA PRUEBA No.2

w	a	b	c	d	e	f	prob.
w ₁	0.7167	0	0	0.7000	86.4165	0.2001	0.3781
w ₂	0.5261	-0.1602	0.1602	0.5261	139.6955	100.1424	0.2280
w ₃	0.6047	0.2384	-0.2384	0.6047	-33.4316	150.6186	0.3185
w ₄	0.1974	-0.0805	0.0322	0.4935	106.9834	179.2168	0.0754



Fig. 19. Atractor generado por el segundo código IFS de la prueba No. 2.

A continuación, se exponen algunas consideraciones producto de un análisis a los resultados arrojados por las pruebas de funcionalidad:

- Si una subimagen del conjunto creado tiene un área mucho mayor que las demás, la descompresión del código IFS hallado genera una imagen con una mayor densidad de puntos en el área que cubre dicha subimagen. Esto se debe a que la probabilidad de una transformación afin contractiva en un código IFS es proporcional al área que ella representa.

- Cuando una porción de la imagen que se desea comprimir se recubre con la intersección de dos o más subimágenes del conjunto creado, al descomprimir el código IFS hallado se observa una mayor densidad de puntos en algunas áreas de la imagen generada. Esto es

consecuencia de que hay una mayor densidad de probabilidad en el área de intersección.

- Cuando una porción de la imagen que se desea comprimir queda sin recubrir, esta irregularidad se reproduce durante el proceso de descompresión del código IFS hallado, generando una imagen con áreas que presentan una densidad de puntos muy baja. La explicación de este comportamiento obedece a las razones expuestas acerca de la asignación de probabilidades.

VI. CONCLUSIONES Y TRABAJO FUTURO

El estudio de la compresión de datos conlleva una gran importancia para el desarrollo de nuevas tecnologías en el ámbito de las telecomunicaciones, pues permite el análisis tangible de la información en los procesos de almacenamiento, transmisión, aseguramiento y encriptación de esta, revelando así un sin número de escenarios de optimización y mejoras para los algoritmos y sistemas actuales de procesamiento de la información.

La compresión de datos es una temática con un grado de dificultad considerable y los recursos educativos disponibles están compuestos en su mayoría por libros tradicionales o videos que no aprovechan las características de los dispositivos electrónicos masificados hoy en día, y no ofrecen posibilidad alguna de interacción con el material de aprendizaje.

Por lo anterior, en el presente trabajo se presentó el diseño y desarrollo de un software interactivo para la enseñanza-aprendizaje de la compresión fractal de imágenes, que permite un aprendizaje mediante actividades dinámicas y con un alto grado de interactividad, esperando lograr una mayor apropiación del contenido de enseñanza.

El software interactivo desarrollado, provee a los profesores de un recurso educativo digital que apoya la comprensión de un tema abstracto como lo es la compresión fractal de imágenes, permitiendo la estimulación cognitiva y despertando la curiosidad de los estudiantes, lo que promueve un mayor aprendizaje de la información expuesta [15].

Como trabajo futuro se propone llevar a cabo la implementación y evaluación del software interactivo con estudiantes de asignaturas relacionadas a la teoría de la información, la teoría de la codificación o materias afines. Esto con el fin de poder evaluar la capacidad del software para llevar a cabo la acción formativa y mejorar el proceso de enseñanza-aprendizaje de dichos temas.

REFERENCIAS

- [1] D. Salomon y G. Motta, *Handbook of Data Compression*, vol. Fifth Edition, New York: Springer, 2010.
- [2] K. Sayood, *Introduction to Data Compression*, San Francisco CA: Elsevier Inc, 2017.
- [3] A. García-Valcárcel, *Recursos digitales para la mejora de la enseñanza-aprendizaje*, Salamanca: [https://gredos.usal.es/bitstream/handle/10366/131421/Recursos digitales.pdf](https://gredos.usal.es/bitstream/handle/10366/131421/Recursos%20digitales.pdf), 2016.

- [4] F. G. Guerrero y L. A. Perez, *A Software for Learning Information Theory basics with Emphasis on Entropy of Spanish*, *Energía Y Computación*, 2008.
- [5] M. Hamada y M. Hassan, *A multimedia learning environment for information theory*, *Proceedings of 2016 IEEE International Conference on Teaching, Assessment and Learning for Engineering*, 2017.
- [6] W. Allen, "Overview and Evolution of the ADDIE Training System", *Adv. Dev. Hum. Resour.* 8, 2006, pp.430-441.
- [7] G. Hernandez y L. F. Niño, *Geometría Fractal en Compresión de Imágenes Digitales*, Bogotá: Universidad Nacional, 1992.
- [8] M. F. Barnsley, *Fractals Everywhere: Third Edition*, New York: Dover Publications, 2012.
- [9] G. Bielech, *Introduction to Data Compression*, Pittsburgh PA: Carnegie Mellon, 2013.
- [10] J. A. Elton, *Simultaneously Contractive Remetrization Theorem for Iterated Function Systems*, Georgia: Institute of Technology Preprint, 1988.
- [11] A. L. Davis, "Using instructional design principles to develop effective information literacy instruction: The ADDIE model", *College & Research Libraries News*, vol. 74, núm. 4, pp. 205-207, 2013.
- [12] N. Barrio, *Aprendizaje Inmersivo, una nueva estrategia de Aprendizaje* [Internet]. *Revista digital INESEM*. 2016 [citado el 8 de junio de 2020]. Disponible en: <https://revistadigital.inesem.es/educacion-sociedad/aprendizaje-inmersivo/>
- [13] J. Nielsen, *10 Usability Heuristics for User Interface Design* [Internet]. Nielsen Norman Group. 1995. Disponible en: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [14] H. Niegemann, y S. Heidig, *Multimedia Learning*. *Encycl Sci Learn*. 2002;41:27-9.
- [15] C. Toca y J. Carrillo, *Los entornos de aprendizaje inmersivo y la enseñanza a ciber-generaciones*, Brasil: *Educação e Pesquisa*, 2019.