

**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD ZACATENCO  
DEPARTAMENTO DE CONTROL AUTOMÁTICO**

**Compresión de Imágenes Digitales mediante Fractales  
generados por Sistemas de Funciones Iteradas**

**TESIS**  
que presenta

**MANUEL ALEJANDRO OJEDA MISSES**

para obtener el grado de

**MAESTRO EN CIENCIAS**

en la Especialidad de

**CONTROL AUTOMÁTICO**

Directora de Tesis: **Dra. Petra Wiederhold Grauert**

México, D.F.

SEPTIEMBRE 2015

# Agradecimientos

“Te quiero... gracias porque vives, porque ayer me dejaste tocar tu luz más íntima y porque dijiste con tu voz y tus ojos lo que yo esperaba toda mi vida”

*Frida Kahlo*

## A MI FAMILIA

A mi abuelita, a mis hermanos Anilú, Miguel y Karina, a mi mamá y a mis tíos por lograr otro gran ciclo en mi vida. A ti Leopoldo Zetina y a tu familia por tu gran apoyo.

## A LA DRA. PETRA WIEDERHOLD

Por su dedicación, paciencia y apoyo académico.

## A MIS AMIGOS

A mis amigos Daniel Muñoz, Edgar González y Juan Torres por su apoyo.

## A MIS SINODALES

A la Dra. Matha Rzedowski y al Dr. Alberto Soria por sus valiosos comentarios y tiempo dedicado para la revisión de esta tesis.

## AL CINVESTAV Y EL DEPARTAMENTO DE CONTROL AUTOMÁTICO

Por permitirme estudiar y continuar con mi desarrollo académico, y por su alto nivel.

## AL CONACYT

Por su gran apoyo económico, y poder continuar con mis estudios.



# Resumen

En esta tesis se estudian los fundamentos matemáticos de fractales generados por sistemas de funciones iteradas y su aplicación a la aproximación y compresión de imágenes digitales binarias de dimensión dos. Dos métodos básicos son analizados, el primero para generar fractales mediante sistemas de funciones iteradas llamado algoritmo de la máquina copiadora. El segundo método pretende comprimir y aproximar imágenes reales mediante fractales generados por sistemas de funciones iteradas, donde estas funciones son construidas y estimadas con ayuda del teorema del collage o montaje. Ambos métodos fueron implementados. Se realizaron amplios experimentos con el algoritmo de la máquina copiadora. Luego, en base al teorema del collage se estiman funciones de similitud y más generalmente, afines, para formar sistemas de funciones iteradas y con ellos generar fractales que logran aproximar y comprimir algunas imágenes binarias reales.

## Abstract

This thesis studies the mathematical foundations of fractals generated by iterated function systems, and their application to the approximation and compression of digital two-dimensional binary images. Two basic methods are analyzed, the first one is called photocopy machine algorithm and generates fractals using iterated function systems. The second method aims to compress and to approximate real digital binary images via fractals generated by iterated function systems, where the functions are estimated with the help of the Collage Theorem. Both methods were implemented. Several experiments were performed using the Photocopy Machine Algorithm. Then, for some real digital binary images, similarity and, more generally, affine functions are estimated to form iterated function systems in order to generate fractals which approximate and compress the given images.



# Índice general

Agradecimientos	I
Resumen	III
<b>1. Introducción</b>	<b>1</b>
1.1. ¿Por qué estudiar a los fractales?	1
1.2. Modelado de imágenes	3
1.3. Imagen digital	4
1.4. Compresión de imágenes digitales	6
1.5. Objetivos	8
1.6. Estructura de la tesis	8
<b>2. Bases matemáticas para fractales</b>	<b>11</b>
2.1. Espacio métrico	11
2.2. Espacio de Hausdorff o Espacio $H$	13
2.3. Mapeo de contracción	16
2.4. Transformaciones afines	19
2.5. Sistemas de funciones iteradas (IFS)	21
2.6. Teorema del collage o teorema del montaje	22
<b>3. Aplicaciones prácticas de fractales</b>	<b>27</b>
3.1. Aplicación de un IFS: algoritmo de la máquina copiadora	27
3.2. Aplicación del teorema del collage o montaje a la compresión de imágenes	32
<b>4. Implementación y resultados de experimentos</b>	<b>41</b>
4.1. Experimentos: algoritmo de la máquina copiadora	42
4.2. Experimentos: compresión de imágenes con transformaciones de similitud	47
4.3. Experimentos: compresión de imágenes con transformaciones afines	57

<b>5. Conclusiones y trabajos futuros</b>	<b>69</b>
5.1. Conclusiones . . . . .	69
5.2. Trabajos futuros . . . . .	70

# Índice de figuras

1.1.	Ejemplo de una figura autosimilar . . . . .	2
1.2.	Soporte de una imagen: $\square = \{(x, y) \in \mathbb{R}^2 : a \leq x \leq b, c \leq y \leq d\}$ . . . . .	4
1.3.	Ejemplo de punto $p$ en $\mathbb{Z}$ y un disco centrado en él . . . . .	5
1.4.	Imagen binaria 2-dimensional . . . . .	6
2.1.	Distancia entre punto y conjunto . . . . .	14
2.2.	Distancia del conjunto $A$ al conjunto $B$ . . . . .	14
2.3.	Distancia del conjunto $B$ al conjunto $A$ . . . . .	15
2.4.	Ejemplo de una sucesión $\{A_n\}_{n=1}^{\infty}$ en $(H(\mathbb{R}^2), h)$ que converge a un helecho . . . . .	15
2.5.	Imagen original y resultado de aplicación de una dilatación con constantes $r_1, r_2$ distintas	19
2.6.	Imagen original y resultado de aplicación de una erosión con constantes $r_1, r_2$ distintas	20
2.7.	Imagen original (izquierda parte inferior), la reflexión con respecto al eje $x$ (izquierda parte superior) y su reflexión con respecto al eje $y$ (a la derecha) . . . . .	20
2.8.	Imagen original (inferior) y resultado de aplicar una similitud (superior), que es una combinación de escalamiento, rotación antihoraria y traslación . . . . .	21
2.9.	Imagen original y el resultado de aplicar una transformación de sesgado . . . . .	21
2.10.	Ejemplos de aproximaciones de un mismo fractal, obtenidas después de $n$ iteraciones de aplicación de un IFS . . . . .	23
2.11.	Conjuntos $[0, 1]$ y $M$ . . . . .	25
3.1.	Ejemplo del algoritmo de la máquina copiadora . . . . .	28
3.2.	Aproximaciones del triángulo de Sierpinski; la última resultado de 10 iteraciones . . .	29
3.3.	Aproximaciones de un helecho; la última resultado de 15 iteraciones . . . . .	29
3.4.	Aproximaciones de un árbol fractal; la última resultado de 10 iteraciones. . . . .	30
3.5.	Aproximaciones de un dragón fractal; la última resultado de 15 iteraciones . . . . .	31
3.6.	Aproximaciones de un mosaico fractal; la última resultado de 10 iteraciones . . . . .	31
3.7.	Triángulo de Sierpinski como unión de tres partes autosemejantes . . . . .	33
3.8.	Aproximación del triángulo de Sierpinski obtenido como atractor de un IFS estimado con el teorema del collage . . . . .	34
3.9.	Imagen de una pirámide [1] como unión de cuatro partes autosemejantes . . . . .	34



3.10. Aproximación de la pirámide obtenida como atractor de un IFS estimado con el teorema del collage . . . . .	35
3.11. Imagen fractal con cinco partes autosemejantes . . . . .	36
3.12. Aproximación obtenida como atractor de un IFS estimado con el teorema del collage .	36
3.13. Imagen del triángulo de Sierpinski rotado como unión de tres partes autosemejantes .	36
3.14. Aproximación del triángulo de Sierpinski rotado obtenido como atractor de un IFS estimado con el teorema del collage . . . . .	37
3.15. Imagen de un castillo [1] como unión de cuatro partes autosemejantes . . . . .	38
3.16. Aproximación del castillo obtenido como atractor de un IFS estimado con el teorema del collage . . . . .	38
4.1. Aproximación del triángulo de Sierpinski con 7 iteraciones . . . . .	44
4.2. Aproximación de un helecho con 8 iteraciones . . . . .	45
4.3. Aproximación de un árbol fractal con 9 iteraciones . . . . .	45
4.4. Aproximación de un dragón fractal con 12 iteraciones . . . . .	46
4.5. Aproximación de un mosaico fractal con 10 iteraciones . . . . .	47
4.6. Conjunto $L$ con $w_i(a) = a'$ , $w_i(b) = b'$ , $w_i(c) = c'$ similitudes . . . . .	48
4.7. Conjunto $L$ con similitud $w_i(L)$ rotada respecto a la figura original $L$ . . . . .	48
4.8. Menú en <i>Matlab</i> para estimación de los mapeos del IFS . . . . .	49
4.9. Imagen del triángulo de Sierpinski con puntos $a, b, c$ de la imagen total (figura $L$ ) y puntos $a', b', c'$ generados por la similitud $w_1$ . . . . .	51
4.10. Imagen del triángulo de Sierpinski original e imagen del fractal aproximado . . . . .	52
4.11. Imagen de un castillo con puntos $a, b, c$ de la imagen total $L$ y puntos $a', b', c'$ generados por la similitud $w_3$ . . . . .	54
4.12. Imagen del castillo original e imagen del fractal aproximado . . . . .	55
4.13. Imagen del fractal original (izquierda) e imagen del fractal aproximado (derecha) . .	56
4.14. Imagen de la hoja fractal original (esquina superior izquierda) y hoja fractal aproximada con tres (esquina superior derecha), cuatro (esquina inferior izquierda) y cinco transformaciones de similitud (esquina inferior derecha) . . . . .	57
4.15. Imagen del triángulo de Sierpinski con puntos $P_1, P_2, P_3$ del conjunto original y puntos $P'_1, P'_2, P'_3$ del conjunto imagen de la transformación afín $w_1$ . . . . .	58
4.16. Collage empleado para el triángulo de Sierpinski . . . . .	62
4.17. Imagen de un helecho de la naturaleza, y su aproximación mediante un fractal generado por un IFS de cinco transformaciones afines . . . . .	62
4.18. Collage empleado para un helecho real . . . . .	63
4.19. Imagen de un conjunto de hojas de un árbol de la naturaleza, y su aproximación mediante un fractal generado por un IFS de cuatro transformaciones afines . . . . .	64
4.20. Collage empleado para un conjunto de hojas de un árbol . . . . .	64

4.21. Imagen de una hoja de la naturaleza, y su aproximación mediante un fractal generado por un IFS de cuatro transformaciones afines . . . . .	65
4.22. Collage empleado para una hoja real . . . . .	66
4.23. Collage empleado para una hoja real . . . . .	66
4.24. Collage empleado para una hoja . . . . .	67
4.25. Imagen de un árbol de la naturaleza, y su aproximación errónea mediante un fractal generado por un IFS de cinco transformaciones afines . . . . .	68
4.26. Collage empleado para las hojas de un árbol . . . . .	68



# Índice de Tablas

4.1. Tabla de transformaciones para el triángulo de Sierpinski . . . . .	44
4.2. Tabla de transformaciones para un helecho . . . . .	44
4.3. Tabla de transformaciones para un árbol fractal . . . . .	45
4.4. Tabla de transformaciones para un dragón fractal . . . . .	46
4.5. Tabla de transformaciones para un mosaico fractal . . . . .	46
4.6. Tabla de transformaciones de similitud estimadas para el triángulo de Sierpinski . . . . .	50
4.7. Tabla de transformaciones de similitud originales para el castillo . . . . .	53
4.8. Tabla de transformaciones de similitud estimadas para el castillo . . . . .	53
4.9. Tabla transformaciones de similitud estimadas para una pirámide . . . . .	55
4.10. Tabla con tres transformaciones de similitud estimadas para una hoja fractal	56
4.11. Tabla con cuatro transformaciones de similitud estimadas para una hoja fractal . . . . .	56
4.12. Tabla con cinco transformaciones de similitud estimadas para una hoja fractal . . . . .	56
4.13. Tabla con cinco transformaciones afines estimadas para un helecho real. .	63
4.14. Tabla con cuatro transformaciones afines estimadas para el conjunto de hojas de un árbol real . . . . .	65
4.15. Tabla con cuatro transformaciones afines estimadas para una hoja real .	66
4.16. Tabla con cuatro transformaciones afines estimadas para una hoja . . . . .	67
4.17. Tabla con cinco transformaciones afines estimadas para un árbol . . . . .	67



# Capítulo 1

## Introducción

### 1.1. ¿Por qué estudiar a los fractales?

“Dios crea círculos, ondas y fractales”

*Benoit B. Mandelbrot* [10]

En el medio ambiente, la mayor parte de los cuerpos guardan una geometría que describe su forma, éstos son formados con figuras regulares o irregulares en el plano o el espacio conteniendo puntos, rectas, curvas, polígonos y en algunos casos con fragmentos y/o irregularidades, generando geometrías inexplicables a simple vista. Hoy en día existen herramientas para llevar a cabo el estudio y descripción de estos objetos o figuras complejas logrando simplificarlas y modelarlas de manera sencilla y eficiente. Gran parte de ellas tiene propiedades de simetría y/o autosimilaridad, es decir, generalmente un pequeño fragmento de ellos contiene figuras que con un cambio de escala es idéntica a la figura completa.

Muchos objetos conservan su estructura aunque se observen de lejos, es decir, esta estructura tiene el mismo aspecto independientemente de su escala. Por ejemplo, si observamos la figura 1.1 (de la izquierda) y realizamos un acercamiento, visualizamos que una pequeña parte de  $b$ ) representa totalmente la figura  $a$ ), permitiendo deducir que  $b$ ) está formada por 8 copias de la figura  $a$ ). Es claro que son imágenes diferentes y similares a la vez, guardando el mismo patrón, rasgo o estructura interna. Esto define que si cualquier parte de una figura por pequeña que sea, contiene información relativa a la misma, es llamada comúnmente *fractal*.

En otros ejemplos, estas copias pueden ser deformadas, es decir, pueden existir en versiones trasladadas, rotadas o distorsionadas. Ésta es la idea de autosimilaridad, una

base fundamental e intuitiva para generar y definir un fractal.

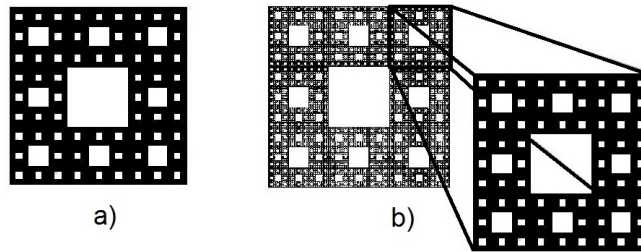


Figura 1.1: Ejemplo de una figura autosimilar

La palabra fractal proviene del adjetivo “fractus” en latín, cuyo significado es un objeto geométrico cuya estructura es fragmentada o irregular y se repite a diferentes escalas. Mandelbrot usó el término fractal por primera vez en la década de los 70’s, específicamente para estudiar fenómenos temporales o espaciales continuos que exhiben correlación dentro de un rango de escalas, el término se justifica porque un fractal no tiene dimensión entera [10].

Claramente se pueden tener  $n$  objetos de la figura *a*) en la figura *b*), para llevar a cabo su estudio, se requiere una geometría útil y efectiva para modelarlos. Algunos fenómenos naturales pueden ser modelables mediante fractales, debido a que su estructura y dimensión matemática son fuera de lo común, intuitivamente son objetos curiosos con curvas o superficies infinitas, indeterminables o aisladas. Para describir a tales objetos, las herramientas de la geometría tradicional no son apropiadas y suficientes, sino que se necesitan nuevos enfoques para su análisis matemático, representativo y computacional. Algunas características de los fractales que motivan su investigación son las siguientes [13], [14]:

1. Aunque tienen estructura compleja aúnan hacia la sencillez.
2. Conservan una estructura fina a toda escala.
3. Una gran cantidad de fractales son generados mediante transformaciones afines.
4. Una estructura fractal puede ser generada con algoritmos simples y de carácter recursivo.
5. En la mayor parte de los fractales no se reconoce un inicio o fin, pero en la etapa de construcción se pueden visualizar aproximaciones obtenidas mediante un número finito de iteraciones.

6. Un fractal posee autosimilaridad, lo que permite aproximarlos de manera más sencilla.
7. Los fractales no son descritos mediante herramientas geométricas tradicionales, puesto que no cumplen con propiedades comunes, por ejemplo, no tienen longitud finita ni dimensión entera.

Los fractales son objetos geométricos cuya estructura es fragmentada, irregular o abstracta. Por estas razones para su estudio en esta tesis, será necesario definir e introducir cómo es modelada una imagen, qué es un espacio métrico, una transformación afín, la métrica de Hausdorff, un sistema de funciones iteradas (IFS) y el teorema de contracción.

## 1.2. Modelado de imágenes

Tradicionalmente, una imagen se ha entendido como una señal de dimensión dos, modelada por una función de dos variables, la cual representa información visual, información visible por el humano y captada por algún sistema óptico. De manera general, una imagen es una representación de información sobre alguna propiedad (por ejemplo física, química, biológica o estadística) de la materia, la cual es modelable matemáticamente por un número finito de variables.

Una representación es un sistema formal para hacer explícitas ciertas entidades o tipos de información, conjuntamente con una especificación de cómo este sistema es descrito. El resultado de la aplicación de una representación a una entidad dada es su descripción.

Concretamente, una imagen es modelada por una función de  $n$  variables reales y con valores en  $\mathbb{R}$  o en  $\mathbb{R}^m$ . En esta tesis consideramos solamente el caso  $n = 2$  (define una imagen de dimensión dos, imagen dos-dimensional o imagen 2D) y  $m = 1$  (denota una imagen monocromática) donde los valores de la función son interpretados como valores de gris o intensidades. Más generalmente, una imagen a color puede ser modelada por una función  $f : \mathbb{R}^n \rightarrow \mathbb{R}^3$ , donde  $f(x) = (y_1, y_2, y_3)$ , cada valor  $y_i$  representa la intensidad de luz dentro de uno de los canales de colores en el sistema RGB (Red, Green, Blue).

Denotamos por  $\mathcal{R}$  al conjunto de las imágenes 2D monocromáticas  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . Para cada imagen  $f \in \mathcal{R}$  suponemos las siguientes propiedades axiomáticas, según [2]:

**Propiedad i.**  $f$  tiene un dominio de definición acotado y rectangular,  $\square = [a, b] \times [c, d] \subseteq \mathbb{R}^2$  (figura 1.2), el cual es llamado soporte de  $f$ .



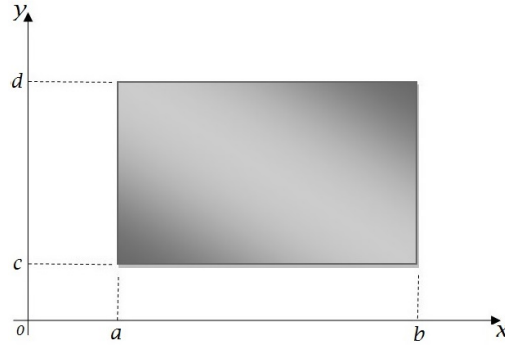


Figura 1.2: Soporte de una imagen:  $\square = \{(x, y) \in \mathbb{R}^2 : a \leq x \leq b, c \leq y \leq d\}$

**Propiedad ii.**  $f$  tiene valores reales los cuales representan sus valores cromáticos, frecuencias e intensidades de luz o alguna otra propiedad del mundo real, o en un caso especial pueden ser creados de manera artificial.

**Propiedad iii.** El autor enfatiza que cada imagen de  $f$  en  $\mathcal{R}$  sea independiente de la resolución. Eso significa que  $f$  puede ser representada mediante una versión digitalizada la cual fue obtenida por una discretización con una resolución arbitrariamente alta.

**Propiedad iv.** El conjunto  $\mathcal{R}$  es cerrado bajo la operación de recorte (*clipping*).

**Propiedad v.**  $\mathcal{R}$  es cerrado bajo las operaciones de estiramiento (*stretching*) y encogimiento (*shrinking*).

**Propiedad vi.**  $\mathcal{R}$  es cerrado bajo la operación de reflexión (*reflection*).

**Propiedad vii.**  $\mathcal{R}$  es cerrado bajo operaciones de recorte de una versión rotada de una región rectangular de la imagen de  $f$ .

**Propiedad viii.**  $\mathcal{R}$  es cerrado bajo transformaciones afines invertibles.

Estas propiedades son propuestas como axiomas para las imágenes reales [2]. Cabe notar que la *propiedad i* determina que toda imagen tenga soporte acotado (es decir, de tamaño finito) y de forma rectangular. La *propiedad ii* no presenta limitaciones, puesto que  $f$  es función de  $\mathbb{R}^2$  en  $\mathbb{R}$ , es decir asigna valores de dimensión e intensidades a cada imagen. La *propiedad iii* en otras palabras denota que cada píxel corresponde a una porción arbitrariamente pequeña del plano. Mientras tanto las propiedades restantes denotan al conjunto  $\mathcal{R}$  cerrado bajo distintos tipos de operaciones o transformaciones.

### 1.3. Imagen digital

Una imagen que representa alguna propiedad de la materia, es llamada *imagen real* en [2] puesto que refleja algún aspecto de la realidad. En contraste, bajo una imagen

artificial se entiende una imagen creada por la computadora o por el humano.

Una imagen es digitalizada cuando es procesada por un discretizador de señales, de esta manera es convertida en una estructura especial de datos, cuyos elementos pertenecen a un conjunto discreto y se le asignan valores discretos (por ejemplo números enteros dentro de un intervalo, o solamente 1 y 0), esta estructura puede ser memorizada y tratada en una computadora.

Un conjunto discreto en  $\mathbb{R}^n$  es un subconjunto  $D \subseteq \mathbb{R}^n$  el cual forma un subespacio discreto del espacio euclidiano  $\mathbb{R}^n$ . Eso significa que para todo punto  $p \in D$  existe un disco abierto alrededor de  $p$  que no contiene a otros puntos de  $D$ . Se observa como el conjunto  $\mathbb{Z}$  en la figura 1.3 está conformado por los números enteros, y para cualquier  $p \in \mathbb{Z}$ , existe un disco abierto centrado en  $p$  que no contiene otros puntos de  $\mathbb{Z}$ . Por ello,  $\mathbb{Z}$  es determinado como discreto.

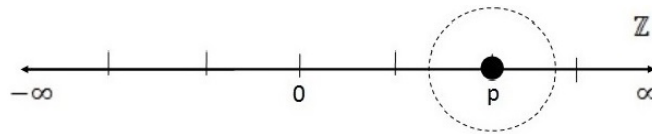


Figura 1.3: Ejemplo de punto  $p$  en  $\mathbb{Z}$  y un disco centrado en él

Una *imagen digital  $n$  – dimensional* es una función  $f_d : D \rightarrow \mathbb{Z}$ , donde  $D$  es un conjunto discreto en  $\mathbb{R}^n$  y  $\mathbb{Z}$  denota al conjunto de los números enteros, considerando que tiene un soporte acotado de forma rectangular y un intervalo en  $\mathbb{Z}$  siendo el dominio de valores también acotado. Una imagen digital 2-dimensional *binaria* es una función  $f_d : D \rightarrow \{1, 0\}$ , donde  $D$  es un conjunto discreto en  $\mathbb{R}^2$ .

Las imágenes binarias (figura 1.4) pueden representar imágenes (de valores en gris o de color) que ya fueron segmentadas, es decir donde los objetos de interés ya son detectados y memorizados en la computadora. La función imagen asigna a todo elemento de  $D$  el valor 1, si el elemento pertenece al objeto de interés, en otro caso, le asigna el valor 0. Éstas son el tipo de imágenes que se estudian en esta tesis.

El proceso de convertir al dominio de definición de una imagen  $f$  en el dominio de definición  $D$  de la imagen digital  $f_d$ , es llamado discretización, mientras que se habla de cuantización en referencia al proceso de convertir al dominio de valores de una imagen  $f$  en el dominio de valores de la imagen  $f_d$ .

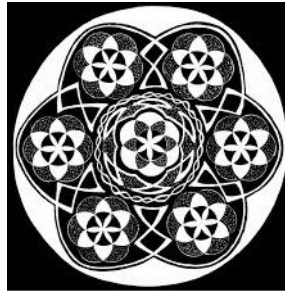


Figura 1.4: Imagen binaria 2-dimensional

## 1.4. Compresión de imágenes digitales

Existen importantes aplicaciones que relacionan a las imágenes y los fractales, entre ellas el modelado de formas naturales, sistemas dinámicos y la compresión de imágenes, que permite el almacenamiento o transmisión de manera eficiente. Según González y Wood [6], la compresión de imágenes es el proceso de reducción del volumen de datos para representar una cantidad de información. Es decir, un conjunto de datos puede contener datos redundantes, repetitivos o similares, los cuales si se identifican harán más sencillo el proceso de compresión.

En general las técnicas de compresión de imágenes se pueden clasificar en dos clases [3]: *a)* Sin pérdida de información, como los métodos de Run-length encoding [9], codificación de Huffman [8], codificación aritmética [3], éstos emplean codificación de bits y el algoritmo Lempel-Ziv [15] que realiza un análisis de cadenas o palabras asignándoles una propabilidad de recurrencia y *b)* Con pérdida de información, por ejemplo la codificación por transformaciones de Fourier [6] y vectores de cuantificación [3] ambas en el dominio de frecuencias, finalmente la compresión fractal [7] que consiste en transmitir imágenes autosemejantes comprimiéndolas mediante funciones.

Las aplicaciones y selección de algún método dependen del tipo de imagen que se esté tratando, es decir, en algunas no es permisible la pérdida de información, como en imágenes biológicas, medicas o legales. Mientras que en otras es posible permitir cierto grado de error, manteniendo la calidad de la imagen.

El objetivo de la compresión de imágenes es reducir los datos de la imagen con la menor pérdida posible de información para permitir su almacenamiento o transmisión de forma eficaz y generar la mejor calidad de imagen en una tasa de bits dada (o tasa de compresión). Enseguida, se presentan otros puntos importantes en los esquemas de

compresión [5]:

1. Escalabilidad. Eso se refiere a la reducción de calidad lograda por la manipulación de los datos del archivo (sin descomprimir y comprimir nuevamente), en otras palabras, se refiere a la codificación progresiva o tramas de bits embebidos. La escalabilidad es útil para previsualizar las imágenes mientras se descargan (por ejemplo, en un navegador web) o para proporcionar accesos de calidad variable (por ejemplo, en una base de datos).
2. Codificación de región de interés. Ciertas partes de la imagen se codifican con una calidad mayor que otras, representando las regiones de interés con la mayor calidad.
3. Meta de información. Los datos comprimidos pueden contener información sobre la imagen que puede ser utilizada para categorizar, buscar o clasificar las imágenes, dicha información puede incluir estadísticas de color, pequeñas imágenes previsualizadas, información encriptada o información de derechos de autor.
4. Recursos de procesamiento. Los algoritmos de compresión requieren diferentes cantidades de recursos, como son el espacio de almacenamiento y el tiempo de procesamiento para codificar y decodificar la información.

La calidad de un método de compresión se mide a menudo por la cantidad de ruido (generado durante dicho proceso o contenido en la imagen), la pérdida de información y mediante la comparación de la aproximación con la imagen real.

La principal motivación de esta tesis es la compresión de imágenes mediante fractales, denotado como un método sencillo, debido a que emplea sólo funciones en  $\mathbb{R}$  o  $\mathbb{R}^2$ . Otra gran ventaja es el estudio preliminar sobre fractales, permitiendo analizar, implementar y comprender la compresión de imágenes grandes, complejas y autosimilares. Algunas razones por las que no se optó por alguno de los otros métodos de compresión es que requieren el estudio de ciertas áreas o temas específicos tales como sistemas de codificación, transformada de Fourier, cuantificación de señales y sistemas aleatorios.

Justificado el método de compresión de imágenes mediante fractales, es importante mencionar que su estudio se basa en el teorema del collage o montaje. Para llevar a cabo el estudio, implementación y validación del teorema del collage será necesario también estudiar un método práctico para generar fractales, denotado algoritmo de la máquina copiadora, y algunos conceptos como son la métrica de Hausdorff y los sistemas de funciones iteradas.

## 1.5. Objetivos

El objetivo general de esta tesis es estudiar los métodos básicos de compresión basados en el algoritmo de la maquina copiadora (photocopy machine algorithm) y en el teorema del collage o montaje (collage theorem).

Los objetivos particulares de este trabajo están relacionados con la compresión de imágenes mediante fractales generados por sistemas de funciones iteradas, son enlistados a continuación:

1. Conocer los fundamentos matemáticos para los fractales que son generados por sistemas de funciones iteradas.
2. Comprender y experimentar la generación de fractales por sistemas de funciones iteradas.
3. Estudiar y experimentar un método básico interactivo de compresión basado en el teorema del collage.
4. Aplicar la compresión basada en el teorema del collage en imágenes digitales para obtener modelos fractales.

Para cumplir con los objetivos, es importante estudiar algunas definiciones, teoremas y algoritmos matemáticos, los cuales son tratados más adelante en este documento de tesis.

## 1.6. Estructura de la tesis

En el Capítulo 1 se describe la justificación para el estudio de los fractales e intuitivamente cómo están constituidos, así como los principales términos y razones que motivan su investigación. Además se presenta la definición de imagen real y digital, su dominio de definición útiles para llevar a cabo el modelado de imágenes. Por último, se detalla el proceso de compresión de imágenes como una técnica de gran utilidad para aproximar imágenes dadas mediante fractales, señalándolo como la motivación fundamental que origina el desarrollo de esta tesis y sus objetivos planteados.

En el Capítulo 2 se expone el concepto de espacio métrico, sucesiones y el espacio métrico euclidiano como un espacio completo. Además se explica el espacio de Hausdorff donde son modelables los fractales. El objetivo de esta unidad es familiarizarse con los fundamentos matemáticos para generar fractales por medio de sistemas de funciones iteradas (IFS) y el teorema del collage o montaje para la compresión de imágenes mediante

fractales.

En el Capítulo 3 se presentan las dos aplicaciones fundamentales de esta tesis, por un lado la aplicación de un IFS llamado el algoritmo de la maquina copiadora, cómo funciona y algunos ejemplos de aproximaciones fractales. Por otro lado se presentan aproximaciones de algunas transformaciones para un IFS estimado con el teorema del collage o montaje.

El Capítulo 4 presenta la etapa de implementación para la estimación de los IFS, así mismo se presentan los resultados obtenidos con transformaciones afines y transformaciones de similitud. También se comparan los resultados, es decir, las aproximaciones obtenidas con las imágenes reales.

Finalmente, en el Capítulo 5 se mencionan las conclusiones que resultaron durante la investigación de esta tesis y trabajos futuros.



# Capítulo 2

## Bases matemáticas para fractales

Como fue mencionado en el introducción, un fractal tiene propiedades particulares, por ejemplo autosemejanza o autosimilaridad y la conservación de su estructura a toda escala. En [10] se define un fractal como un objeto geométrico irregular y fragmentado, lo cual intuitivamente significa que es un subconjunto de un espacio métrico formado por partes o pedazos. Para definir formalmente a los fractales considerados en esta tesis, son necesarios algunos conceptos matemáticos que recordamos a continuación.

### 2.1. Espacio métrico

**Definición.** Un *espacio métrico*  $(X, d)$  es un conjunto no vacío  $X$ , junto con una función con valores reales  $d : X \times X \rightarrow \mathbb{R}$ , que mide la distancia entre los puntos  $x$  y  $y$  en  $X$ , entonces  $(X, d)$  se llama espacio métrico siempre y cuando se cumplan las siguientes propiedades:

- i)  $d(x, y) \geq 0$ ,  $d(x, y) = 0 \iff x = y$ ,  $\forall x, y \in X$  (positividad).
- ii)  $d(x, y) = d(y, x)$ ,  $\forall x, y \in X$  (simetría).
- iii)  $d(x, y) + d(y, z) \geq d(x, z)$ ,  $\forall x, y, z \in X$  (desigualdad del triángulo).

**Ejemplos:**

1.  $(\mathbb{R}, d_{\mathbb{R}})$  con  $d_{\mathbb{R}}(x, y) = |x - y| = \sqrt{(x - y)^2}$  valor absoluto de la diferencia.
2.  $(\mathbb{R}^2, d_{\mathbb{R}^2})$  con  $d_{\mathbb{R}^2}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$  llamada métrica euclidiana.
3.  $(\mathbb{R}^2, d_1)$  con  $d_1(x, y) = |x_1 - y_1| + |x_2 - y_2|$  llamada métrica  $d_1$  o de Manhattan.
4.  $(\mathbb{R}^2, d_{\text{máx}})$  con  $d_{\text{máx}}(x, y) = \text{máx}\{(x_1 - y_1), (x_2 - y_2)\}$  llamada métrica  $d_{\text{máx}}$ .



La geometría fractal está concentrada en la descripción, clasificación, análisis y observación de subconjuntos de un espacio métrico  $(X, d)$ , pero estos no siempre tienen un carácter geométrico simple, su descripción requiere de conceptos topológicos. En nuestro estudio de fractales nos encontraremos con el concepto de espacio métrico completo, al cual definiremos más adelante.

El concepto de sucesión es importante, intuitivamente es una lista ordenada de elementos u objetos, matemáticamente una sucesión de  $X$  es un mapeo  $f : \mathbb{N} \rightarrow X$ , es decir, para todo  $n \in \mathbb{N}$ ,  $f(n) = x_n \in X$ , y se denota  $\{x_n\}_{n=1}^{\infty}$ .

**Definición.** Sea  $\{x_n\}_{n=1}^{\infty}$  una sucesión en un espacio métrico  $(X, d)$ ;

- \*  $\{x_n\}_{n=1}^{\infty}$  se llama *sucesión de Cauchy* si dado cualquier  $\varepsilon > 0$  existe  $N \in \mathbb{N}$ , tal que  $d(x_n, x_m) < \varepsilon, \forall n, m > N$ ;
- \*  $\{x_n\}_{n=1}^{\infty}$  se llama *sucesión convergente* a un punto  $x \in X$ , si dado  $\varepsilon > 0$  existe  $N \in \mathbb{N}$ , tal que  $d(x_n, x) < \varepsilon, \forall n > N$ . El punto  $x$  es llamado el límite de la sucesión, esto es denotado  $\lim_{n \rightarrow \infty} x_n = x$ .

Es claro que toda sucesión  $\{x_n\}_{n=1}^{\infty}$  convergente en  $X$ , es de Cauchy. Sin embargo, en algunos espacios hay sucesiones de Cauchy que no son convergentes. Por ejemplo, en el espacio métrico de los números racionales con el valor absoluto usual existen sucesiones de números racionales que convergen hacia un número irracional.

**Definición.** Un espacio métrico  $(X, d)$  es *completo*, si toda sucesión de Cauchy  $\{x_n\}_{n=1}^{\infty}$  converge en  $X$ , es decir, tiene un límite  $x \in X$ .

Tenemos en mente el caso  $X = \mathbb{R}^2$  con la métrica euclidiana sobre  $X$ , por lo que en consecuencia, el siguiente teorema será de gran utilidad.

**Teorema.** *El espacio métrico euclidiano  $\mathbb{R}^n$  es un espacio métrico completo.*

Para terminar esta sección, recordamos algunos conceptos básicos de la topología dados en [11]. Sea  $(X, d)$  un espacio métrico, se define un disco abierto para  $x \in X$  y  $\varepsilon \in \mathbb{R}, \varepsilon > 0$ ,  $U_\varepsilon(x) = \{y \in X : d(x, y) < \varepsilon\}$ . Para el subconjunto  $M \subseteq X$  recordemos lo siguiente:

- \*  $M$  es *compacto*, si cada sucesión  $\{x_n\}_{n=1}^{\infty}$  en  $M$  contiene una subsucesión convergente con límite en  $M$ .
- \*  $M$  es *acotado*, si existe un punto  $a \in X$  y un número positivo  $R$  tal que  $d(a, x) < R$

para todo  $x$  de  $M$ .

- \*  $cl(M) = \{y \in X: \text{para todo } \varepsilon > 0, U_\varepsilon(y) \cap M \neq \emptyset\}$  (cerradura).
- \*  $M$  es *cerrado* si  $cl(M) = M$ .
- \*  $int(M) = \{m \in M: \text{existe } \varepsilon > 0 \text{ tal que } U_\varepsilon(m) \subset M\}$  (interior).
- \*  $fr(M) = cl(M) \setminus int(M)$  (frontera).
- \*  $x$  es un *punto de acumulación o punto límite* de  $M$ , si  $x \in cl(M \setminus \{x\})$ .

La familia  $\tau$  de todos los subconjuntos abiertos de un espacio métrico  $(X, d)$  tiene las siguientes tres propiedades: El conjunto total  $X$  y el conjunto vacío  $\emptyset$  pertenecen a  $\tau$ , la intersección de cada dos conjuntos pertenecientes a  $\tau$  es nuevamente un elemento de  $\tau$ , y la unión arbitraria de conjuntos que están en  $\tau$  nuevamente pertenece a  $\tau$ . Si  $X$  simplemente es un conjunto y  $\tau$  una familia de subconjuntos de  $X$  con estas tres propiedades, entonces  $\tau$  se llama una *topología* sobre  $X$  y  $(X, \tau)$  se llama un *espacio topológico*. En consecuencia, todo espacio métrico también es un espacio topológico.

## 2.2. Espacio de Hausdorff o Espacio $H$

Un espacio de Hausdorff es definido como un espacio topológico  $(X, \tau)$  en el cual cualesquiera dos puntos distintos  $x, y \in X$  son separables en el siguiente sentido: existen abiertos disjuntos  $M, N \subseteq X$ , es decir,  $M, N \in \tau$ ,  $M \cap N = \emptyset$  tales que  $x \in M$  y  $y \in N$ . Resulta que cualquier espacio métrico  $(X, d)$  es un espacio de Hausdorff: si tomamos  $x, y \in X$  dos puntos distintos entonces su distancia  $d(x, y)$  es un número positivo distinto del cero. Si usamos ahora la mitad de este número como radio  $r$  de cada uno de los discos abiertos  $U_r(x), U_r(y)$ , entonces ciertamente estos discos abiertos son disjuntos y cada uno contiene su punto centro.

Un espacio métrico especial, el cual naturalmente es de Hausdorff, resulta ser útil como modelo del espacio de fractales. Para eso se necesita la métrica de Hausdorff la cual es introducida en esta sección. Esta métrica mide la distancia entre conjuntos y es una clave fundamental para comprender la naturaleza de los fractales.

**Notación.** Si  $(X, d)$  es un espacio métrico completo, entonces  $H(X)$  denota al conjunto cuyos puntos son los subconjuntos no vacíos y compactos de  $X$ .

**Definición.** Sean  $(X, d)$  un espacio métrico completo,  $x \in X$  y  $B \in H(X)$ , se define:

$$d(x, B) = \min\{d(x, b) : b \in B\},$$

$d(x, B)$  es llamada la distancia del punto  $x$  al conjunto  $B$ .

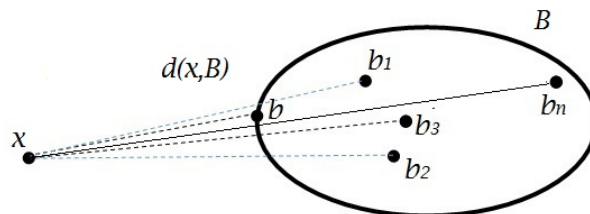


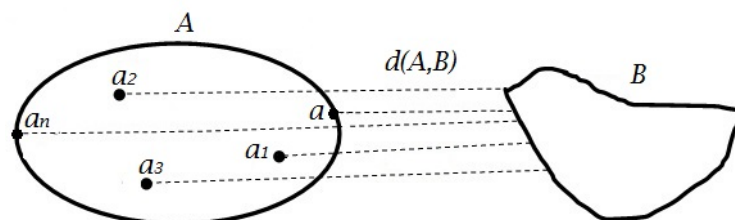
Figura 2.1: Distancia entre punto y conjunto

En la figura 2.1 se muestra gráficamente la distancia del punto  $x$  al conjunto  $B$ , donde se seleccionan los puntos  $b, b_1, b_2, b_3, \dots, b_n$  y con la definición, la distancia  $d(x, B)$  es dada por el mínimo de las distancias euclidianas entre el punto  $x$  y los puntos de  $B$ , en particular, para este ejemplo es mínima para el punto  $b$  de la frontera del conjunto  $B$ .

**Definición.** Sean  $(X, d)$  un espacio métrico completo, y los subconjuntos de  $X$ ,  $A$  y  $B$  que pertenecen a  $H(X)$ . Se define:

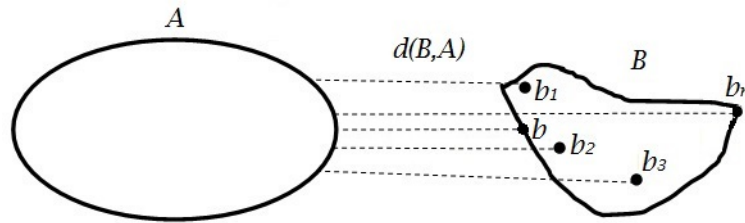
$$d(A, B) = \max\{d(a, B) : a \in A\},$$

$d(A, B)$  es llamada la distancia desde el conjunto  $A \in H(X)$  al conjunto  $B \in H(X)$ .

Figura 2.2: Distancia del conjunto  $A$  al conjunto  $B$ 

La figura 2.2 muestra gráficamente la distancia del conjunto  $A$  al conjunto  $B$ , donde se seleccionan los puntos  $a, a_1, a_2, a_3, \dots, a_n$  de  $A$ . Con el mismo principio de medir la distancia entre punto y conjunto, y de forma análoga se determina la distancia de  $B$  a  $A$ , denotada  $d(B, A)$ , ejemplificada en la figura 2.3.

Éste ejemplo muestra como la distancia  $d(A, B)$  no cumple los axiomas de una métrica, puesto que no cumple con la propiedad de simetría, es decir,  $d(A, B) \neq d(B, A)$ . Por lo tanto es necesario definir una nueva métrica, la métrica de Hausdorff.

Figura 2.3: Distancia del conjunto  $B$  al conjunto  $A$ 

**Definición.** Para cualquier espacio métrico completo  $(X, d)$ , la *métrica de Hausdorff* entre los conjuntos  $A, B \in H(X)$ , es definida como sigue:

$$h(A, B) = \max \{d(A, B), d(B, A)\}.$$

Se puede demostrar que si  $(X, d)$  es un espacio métrico completo, entonces la distancia  $h(A, B)$  es una métrica sobre el conjunto  $H(X)$  de todos los conjuntos no vacíos y compactos de  $X$ [1].

**Teorema.** Si  $(X, d)$  es un espacio métrico completo, entonces el espacio métrico  $(H(X), h)$  es completo también. Además, si  $\{A_n\}_{n=1}^{\infty}$  es una sucesión de Cauchy en  $(H(X), h)$ , tal que  $A = \lim_{n \rightarrow \infty} A_n \in H(X)$ , entonces  $A$  es caracterizado de la siguiente manera:

$$A = \{x \in X : \text{existe una sucesión de Cauchy } \{x_n\}_{n \geq 1} \text{ con } x_n \in A_n \text{ la cual converge a } x\}.$$

Figura 2.4: Ejemplo de una sucesión  $\{A_n\}_{n=1}^{\infty}$  en  $(H(\mathbb{R}^2), h)$  que converge a un helecho

**Definición.** Al espacio métrico  $(H(X), h)$  se le llama *espacio de fractales*.

Resumiendo, el espacio de fractales  $H(X)$  es un espacio métrico completo, y como espacio topológico es de Hausdorff. Sus elementos son subconjuntos compactos de  $X$ .

## 2.3. Mapeo de contracción

Una contracción o aplicación contractiva es una herramienta en la solución de problemas, por ejemplo, el análisis y modelado de sistemas eléctricos, mecánicos y dinámicos, en el estudio de modelos económicos, en modelos de crecimiento de población y en modelos caóticos. Las contracciones son usadas también en el estudio de algoritmos iterativos empleados en métodos numéricos. Incluso determinados fractales son puntos fijos de ciertas contracciones, lo cual es de interés en esta tesis.

Una contracción es una función de un espacio métrico en sí mismo que acorta las distancias y reduce los tamaños, en otras palabras, es una función que tiene una constante de Lipschitz estrictamente menor a uno y por lo tanto, es continua.

**Definición.** Una función  $f : X \rightarrow X$  en un espacio métrico  $(X, d)$ , es llamada *contracción o mapeo contractivo* si existe una constante  $0 \leq s < 1$  tal que:

$$d(f(x), f(y)) \leq s \cdot d(x, y) \forall x, y \in X,$$

donde  $s$  es llamado el factor de contracción de  $f$ .

**Ejemplo 1:** Sea la función  $f : \mathbb{R} \rightarrow \mathbb{R}$ , dada por  $f(x) = \frac{1}{2}x + 5$ , el factor de contracción es determinado:

$$\begin{aligned} d(f(x), f(y)) &= \left| \frac{1}{2}x + 5 - \left( \frac{1}{2}y + 5 \right) \right| \\ &= \frac{1}{2} \cdot |x - y| = \frac{1}{2} \cdot d(x, y) \\ &\leq \frac{1}{2} \cdot d(x, y) \end{aligned}$$

por lo que,  $s = \frac{1}{2}$  es el factor de contracción.

**Ejemplo 2:** Sea la función  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , dada por:

$$\begin{aligned} f \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} = \frac{1}{2}z + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} ; z = \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned}$$

Luego, el factor de contracción es determinado:

$$\begin{aligned} d(f(z), f(w)) &= \left| \frac{1}{2}z + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} - \left( \frac{1}{2}w + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right) \right| \\ &= \frac{1}{2} \cdot |z - w| = \frac{1}{2} \cdot d(z, w) \\ &\leq \frac{1}{2} \cdot d(z, w) \end{aligned}$$

por lo que,  $s = \frac{1}{2}$  es el factor de contracción.

El siguiente resultado es llamado el teorema del punto fijo, el cual garantiza que toda contracción sobre un espacio métrico completo tiene un punto fijo único. Eso es de gran importancia para la geometría fractal, pues es justamente el resultado que garantiza la existencia y unicidad de un fractal.

Un punto  $x \in X$  es un punto fijo de la función  $f : X \rightarrow X$ , si  $f(x) = x$ . El teorema usa la siguiente notación:

**Notación.** Si  $f : X \rightarrow X$  es una función en un espacio métrico, entonces  $f^{\circ n} : X \rightarrow X$  es la  $n$ -ésima aplicación iterativa de  $f$ , es decir, para  $x \in X$ :

$$f^{\circ 0}(x) = x, f^{\circ 1}(x) = f(x), f^{\circ 2}(x) = f(f^{\circ 1}(x)) = f(f(x)), \dots, f^{\circ n}(x) = f(f^{\circ(n-1)}(x)), \forall n \geq 1.$$

Análogamente si  $f$  es invertible y  $f^{-1}$  es la función inversa, entonces  $f^{\circ(-m)}(x) : X \rightarrow X$  es definida de forma recursiva:

$$f^{\circ(-1)}(x) = f^{-1}(x), f^{\circ(-m)}(x) = (f^{\circ m})^{-1}(x), \forall m \geq 1.$$

**Teorema.** *Teorema del punto fijo/Teorema de contracción.* Todo mapeo de contracción  $f : X \rightarrow X$  en un espacio métrico completo  $(X, d)$ , tiene un punto fijo  $x_f \in X$  único. Además la sucesión  $\{f^{\circ n}(x)\}_{n \geq 1}$  converge a  $x_f$ , es decir,

$$\lim_{n \rightarrow \infty} f^{\circ n}(x) = x_f, \forall x \in X.$$

**Ejemplo:** Consideremos  $f : \mathbb{R} \rightarrow \mathbb{R}$ , dada por  $f(x) = \frac{1}{2}x + 5$ . El punto fijo es  $x_f = 10$ :

$$x_f = \frac{1}{2}x_f + 5, \quad x_f - \frac{1}{2}x_f = 5, \quad \frac{1}{2}x_f = 5, \quad x_f = 10.$$

Aplicando el teorema, calculamos el punto fijo de manera iterativa:

$$\begin{aligned}
 f^{\circ 1}(x) &= \frac{1}{2}x + 5 \\
 f^{\circ 2}(x) &= \frac{1}{2} \left( \frac{1}{2}x + 5 \right) + 5 = \frac{1}{4}x + 5 \left( 1 + \frac{1}{2} \right) \\
 f^{\circ 3}(x) &= \frac{1}{2} \left( \frac{1}{4}x + 5 \left( 1 + \frac{1}{2} \right) \right) + 5 = \frac{1}{8}x + 5 \left( 1 + \frac{1}{2} + \frac{1}{4} \right) \\
 &\vdots \\
 f^{\circ n}(x) &= \frac{1}{2^n}x + 5 \sum_{i=1}^{n-1} \left( \frac{1}{2} \right)^i = \frac{1}{2^n}x + \frac{1 - \left( \frac{1}{2} \right)^n}{1 - \frac{1}{2}} = \frac{1}{2^n}x + 5 \cdot 2 \cdot \left( 1 - \left( \frac{1}{2} \right)^n \right)
 \end{aligned}$$

Finalmente se obtiene:

$$\begin{aligned}
 x_f &= \lim_{n \rightarrow \infty} f^{\circ n}(x) \\
 &= \lim_{n \rightarrow \infty} \left( \frac{1}{2^n}x + 10 \cdot \left( 1 - \left( \frac{1}{2} \right)^n \right) \right) \\
 &= \lim_{n \rightarrow \infty} \left( \frac{1}{2^n}x \right) + \lim_{n \rightarrow \infty} \left( 10 \cdot \left( 1 - \left( \frac{1}{2} \right)^n \right) \right) \\
 x_f &= 0 + 10(1 - 0) = 10
 \end{aligned}$$

Recordando que el espacio métrico  $(H(X), h(d))$  de los subconjuntos compactos no vacíos de  $X$  con la métrica de Hausdorff  $h(d)$ , también es un espacio completo. Aplicando el teorema de contracción, un fractal se define en esta tesis como el punto fijo de un mapeo contractivo en  $(H(X), h(d))$ . Para construir un mapeo de contracción definido sobre  $H(X)$ , observamos primero que todo mapeo de contracción es continuo, lo cual es una consecuencia inmediata de la definición de contracción. Luego, se sabe de la topología [11] que toda imagen continua de un compacto es nuevamente un conjunto compacto, es decir, si  $w : X \rightarrow X$  es continuo, entonces  $w$  mapea a  $H(X)$  en sí mismo. Eso implica lo siguiente:

**Lema.** Sea  $w : X \rightarrow X$  una contracción en un espacio métrico  $(X, d)$ , con su factor de contracción  $s$ , entonces el mapeo  $w : H(X) \rightarrow H(X)$  definido por:

$$w(B) = \{w(x) : x \in B\}, B \in H(X)$$

también es una contracción con el mismo factor de contracción  $s$ , donde  $H(X)$  es equipado con la métrica de Hausdorff.

En consecuencia se requiere la construcción de contracciones definidas en  $X$ , para nuestro caso en  $\mathbb{R}^2$ , para obtener contracciones en el espacio  $H(X)$ . Una clase interesante de contracciones se obtiene de las transformaciones afines.

## 2.4. Transformaciones afines

Las transformaciones afines son aplicadas en los sistemas de funciones iteradas (Iterated Function Systems), los cuales son usados para generar fractales, y en consecuencia, también para la compresión de imágenes.

**Definición.** Una *transformación afín* en el plano euclidiano es un mapeo  $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  de la siguiente forma:

$$w(x, y) = (ax + by + e, cx + dy + f)$$

donde  $a, b, c, d, e$  y  $f$  son números reales llamados parámetros. Equivalentemente, de forma matricial tenemos:

$$w(x, y) = w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = Tx + V$$

Así que, la transformación afín  $w$  consiste en una transformación lineal  $T$ , y una traslación especificada por el vector  $V$ . Veamos algunos ejemplos:

1. La transformación  $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , definida por  $w(x, y) = (x, y)$  es el mapeo *identidad*.
2. Una transformación dada por  $w(x, y) = (r_1x, r_2y)$ , donde  $r_1, r_2$  son constantes, es una *dilatación* en el caso que  $|r_1|, |r_2| \geq 1$ , y es una *erosión* cuando  $|r_1|, |r_2| < 1$ .



Figura 2.5: Imagen original y resultado de aplicación de una dilatación con constantes  $r_1, r_2$  distintas





Figura 2.6: Imagen original y resultado de aplicación de una erosión con constantes  $r_1, r_2$  distintas

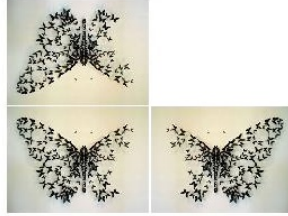


Figura 2.7: Imagen original (izquierda parte inferior), la reflexión con respecto al eje  $x$  (izquierda parte superior) y su reflexión con respecto al eje  $y$  (a la derecha)

3. Las transformaciones  $w_1(x, y) = (x, -y)$  y  $w_2(x, y) = (-x, y)$  corresponden a la *reflexión*, respecto al eje  $x$  y al eje  $y$ , respectivamente.
4. Una transformación  $w(x, y) = (x + f, y + g)$ , donde  $f, g$  son constantes reales, es llamada una *traslación*.
5. Una transformación afín  $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  es llamada *similitud*, si tiene una de las siguientes formas ( $w_1$  con rotación antihoraria y  $w_2$  con rotación horaria):

$$w_1(x, y) = w_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cos \theta & -r \operatorname{sen} \theta \\ r \operatorname{sen} \theta & r \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = T_1 x + V_1$$

$$w_2(x, y) = w_2 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cos \theta & r \operatorname{sen} \theta \\ r \operatorname{sen} \theta & -r \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = T_2 x + V_2$$

para números reales  $r, e, f$  con  $r \neq 0$  y algún ángulo  $\theta$ ,  $0 \leq \theta \leq 2\pi$ , llamado el ángulo de rotación, mientras que  $r$  es llamado el factor de escala o de escalamiento.

6. Una transformación  $w(x, y) = (x + by, y)$  o  $w(x, y) = (x, cx + y)$ , donde  $b, c$  son constantes reales, es llamada *distorsión* o *sesgado*.

Finalmente observamos que no toda transformación afín es contractiva. Una contracción es obtenida mediante las transformaciones de erosión o similitud con factor de escala  $r$ , siempre y cuando cumpla la condición  $|r| < 1$ .



Figura 2.8: Imagen original (inferior) y resultado de aplicar una similitud (superior), que es una combinación de escalamiento, rotación antihoraria y traslación



Figura 2.9: Imagen original y el resultado de aplicar una transformación de sesgado

## 2.5. Sistemas de funciones iteradas (IFS)

En esta sección se aborda el estudio del mecanismo más usual para construir fractales. Esto nos permite establecer una definición formal de la noción de autosimilaridad, que se discutió de manera intuitiva en la introducción y que constituye una de las propiedades esenciales de los fractales.

Si se aplica una sola transformación afín contractiva a un conjunto compacto, el resultado será un compacto más pequeño. Si repetimos la aplicación muchas veces, el tamaño del resultado será reducido posiblemente hasta obtener un punto. Por lo tanto es necesario llevar a cabo varias transformaciones a la vez, las cuales lleven los puntos hacia nuevas posiciones, para obtener un resultado interesante. Ésa es la idea fundamental de un sistema de funciones iteradas (IFS).

**Teorema.** Sean  $(X, d)$  un espacio métrico y  $w_n : n = 1, \dots, N$  contracciones de  $X$  en  $X$  con sus factores de contracción  $s_n$ . Entonces la transformación  $W : H(X) \rightarrow H(X)$  definida por

$$W(B) = w_1(B) \cup w_2(B) \cup w_3(B) \cup \dots \cup w_N(B) = \cup_{n=1}^N w_n(B), \quad B \in H(X)$$

es una contracción en  $(H(X), h(d))$  con factor de contracción  $s = \max\{s_n : n = 1, \dots, N\}$ .

Como resultado del teorema del punto fijo, el único punto fijo  $A \in H(X)$  del mapeo

$W$  cumple que  $A = W(A) = \cup_{n=1}^N w_n(A)$  y es dado como  $A = \lim_{n \rightarrow \infty} W^{on}(B)$  para cualquier  $B \in H(X)$ . Es importante observar que el conjunto inicial  $B$  para las aplicaciones iterativas de  $W$  puede ser cualquier subconjunto compacto no vacío de  $X$ . Este teorema es la motivación para la definición de un sistema de funciones iteradas que consiste de un número finito de transformaciones afines definidas en el plano.

**Definición.** Un *sistema de funciones iteradas* (IFS) de un espacio métrico completo  $(X, d)$  consiste en un conjunto finito de mapeos de contracción  $w_n : X \rightarrow X$ , con factores de contracción  $s_n$  para  $n = 1, \dots, N$  y es denotado por  $\{X; w_n, n = 1, \dots, N\}$ .

En esta tesis nos interesan solamente los casos donde  $X$  es la línea real o el plano euclidiano o un subconjunto apropiado de éstos. La aplicación de un IFS consiste en aplicar cada transformación  $w_i$  a cualquier subconjunto compacto inicial  $B = A_0$  de  $X$  (por ejemplo un punto), formando la unión de dichos resultados obteniendo un nuevo conjunto  $A_1$ :

$$A_1 = w_1(A_0) \cup w_2(A_0) \cup w_3(A_0) \cup \dots \cup w_N(A_0)$$

Luego es aplicado el mismo principio, generando un nuevo conjunto  $A_2$ :

$$A_2 = w_1(A_1) \cup w_2(A_1) \cup w_3(A_1) \cup \dots \cup w_N(A_1)$$

El proceso aplicado es iterativo, el cual genera nuevos conjuntos  $A_3, A_4, A_5, \dots$ , el resultado de infinitas iteraciones es un fractal. En la práctica los ejemplos generados en la computadora sólo se efectúan con un número finito de iteraciones, dando como resultado una aproximación de un fractal.

**Definición.** El punto fijo  $A \in H(X)$ , descrito en el teorema anterior, es decir,  $A = \lim_{n \rightarrow \infty} W^{on}(B)$  para  $B \in H(X)$  y  $W$  dado por los mapeos  $w_1, w_2, \dots, w_N$  del IFS, es llamado el *atractor o fractal* generado por el IFS.

## 2.6. Teorema del collage o teorema del montaje

El siguiente teorema es la base central en el diseño de un IFS. Dada una imagen  $L$  con un dominio de definición acotado,  $L \in H(X)$ , se supone que existe un IFS  $w_1, w_2, \dots, w_N$ , tal que  $\cup_{n=1}^N w_n(L)$  da un conjunto muy cercano o similar a  $L$ . En otras palabras, el IFS genera un atractor  $A$  muy similar o casi igual a  $L$ . El teorema del collage o montaje [2] tiene como objetivo llevar a cabo la compresión de imágenes, sin embargo, pueden aproximarse diferentes IFS.

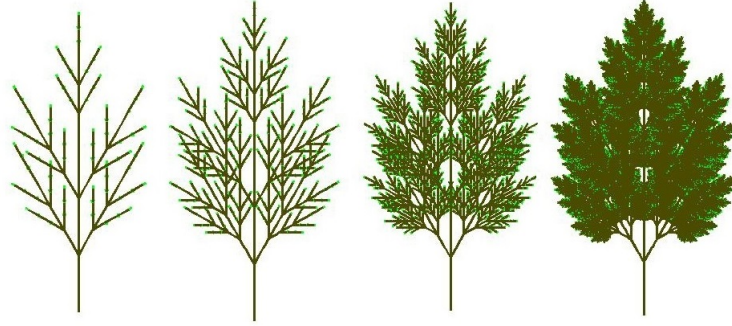


Figura 2.10: Ejemplos de aproximaciones de un mismo fractal, obtenidas después de  $n$  iteraciones de aplicación de un IFS

**Teorema.** *Teorema del collage/Teorema del montaje.* Sean  $(X, d)$  un espacio métrico completo,  $L \in H(X)$  y  $\varepsilon \geq 0$ . Además, supongamos que un IFS  $\{X; w_0, w_1, w_2, \dots, w_N\}$  con factor de contracción  $s = \max\{s_0, s_1, s_2, \dots, s_N\}$  cumple que

$$h\left(L, \bigcup_{n=1}^N w_n(L)\right) \leq \varepsilon,$$

donde  $h$  denota la métrica de Hausdorff sobre el conjunto  $H(X)$ . Entonces vale que

$$h(L, A) \leq \varepsilon/(1 - s),$$

donde  $A$  es el atractor del IFS. Equivalentemente,

$$h(L, A) \leq (1 - s)^{-1} h\left(L, \bigcup_{n=1}^N w_n(L)\right) \quad \forall L \in H(X).$$

El objetivo fundamental se basa en encontrar similitudes  $w_1, w_2, \dots, w_N$  tales que la unión  $\bigcup_{n=1}^N w_n(L) = w_1(L) \cup w_2(L) \cup \dots \cup w_N(L)$  la cual es justamente el montaje o collage de  $L$ , da un conjunto muy similar o casi igual a  $L$ . Entonces se deduce que el IFS dado por  $w_1, w_2, \dots, w_N$  genera un atractor  $A$  muy parecido a  $L$ . El número  $\varepsilon$  denota al factor de error entre el conjunto compacto dado  $L$  y su montaje o collage. Algunos ejemplos se presentan a continuación:

**Ejemplo 1:** El conjunto  $L = [0, \frac{3}{4}] \cup [\frac{3}{4}, 1] = [0, 1]$  puede ser representado mediante dos transformaciones afines  $w_1([0, 1]) = [0, \frac{3}{4}]$  y  $w_2([0, 1]) = [\frac{3}{4}, 1]$ . El problema fundamental es estimar o aproximar cuáles son las transformaciones  $w_1, w_2$  que construyen al conjunto  $L$ , es decir, tales que  $w_1([0, 1]) \cup w_2([0, 1]) = L$ .

Una solución es hacer que  $w_1$  cumpla las siguientes condiciones  $w_1(0) = 0$  y  $w_1(1) = \frac{3}{4}$ . Eso es posible satisfacer mediante la función

$$w_1(x) = \frac{3}{4}x, \quad x \in [0, 1].$$

De manera análoga para  $w_2$ , exigiendo  $w_2(0) = \frac{3}{4}$  y  $w_2(1) = 1$ , se obtiene

$$w_2(x) = \frac{1}{4}x + \frac{3}{4}, \quad x \in [0, 1].$$

Por lo tanto, dichas transformaciones afines generan al conjunto  $L$  mediante el collage  $L = [0, \frac{3}{4}] \cup [\frac{3}{4}, 1]$ . Así que,  $L$  no solamente es similar a su collage sino igual. Por el teorema del collage entonces vale que  $h(L, A) = 0$  para el atractor  $A$  del IFS  $\{[0, 1]; w_1, w_2\}$ , pero este atractor es el mismo intervalo  $[0, 1]$ .

**Ejemplo 2:** En el siguiente ejemplo se desea aproximar al conjunto  $L = [0, 1]$  mediante dos transformaciones, considerando un error  $\varepsilon$ , el objetivo es ajustar los parámetros de las transformaciones  $w_1 = ax + b$  y  $w_2 = cx + d$ , donde  $a, b, c, d \in \mathbb{R}$  para  $x \in [0, 1]$ . El propósito final es obtener un atractor  $A$  muy similar a  $L$ .

Como primer paso se eligen aleatoriamente los parámetros  $a = 0.51, b = 0.1, c = 0.47, d = 0.53$ , los cuales definen las transformaciones afines:

$$w_1 = 0.51x - 0.01, w_2 = 0.47x + 0.53$$

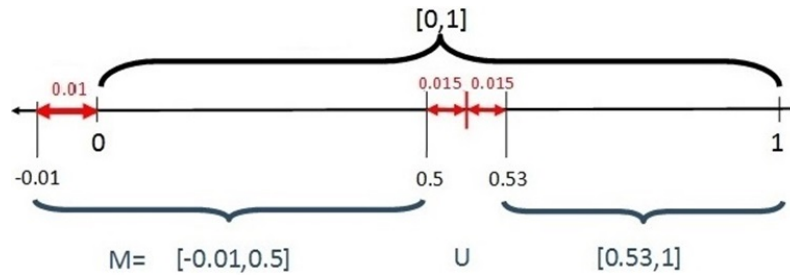
y al IFS  $\{[0, 1]; w_1, w_2\}$  con factor de contracción  $s = \max\{s_1, s_2\} = 0.51$ .

A continuación se calcula la distancia entre el conjunto  $L = [0, 1]$  y su collage  $M = w_1([0, 1]) \cup w_2([0, 1])$ . En la figura 2.11 se representan los conjuntos  $[0, 1]$  y  $M = [-0.01, 0.5] \cup [0.53, 1]$ , y se calcula  $h([0, 1], M) = \max\{d([0, 1], M), d(M, [0, 1])\}$  empleando la métrica de Hausdorff.

Obviamente  $d(x, M) = 0$  para  $x \in [0, 1] \setminus [0.5, 0.53]$ , pero para  $x \in [0.5, 0.53]$  vale que  $d(x, M) \leq \frac{|0.53-0.5|}{2} = 0.015 = d(0.015, 0.5)$ . Eso implica que  $d([0, 1], M) = \max\{d(x, M) : x \in [0, 1]\} = 0.015$ .

Por otro lado, todo  $m \in M$  con  $m \geq 0$  cumple que  $d(m, [0, 1]) = 0$ . Pero para  $m = -0.01$  y  $x = 0$  tenemos  $d(m, x) = 0.01$ . Por lo tanto  $\max\{d(m, [0, 1]), m \in M\} = d(M, [0, 1]) = 0.01$ . En consecuencia  $h([0, 1], M) = \max\{0.015, 0.01\} = 0.015$ .

Tenemos  $h(L, w_1(L) \cup w_2(L)) = 0.015 < \varepsilon$ , para  $\varepsilon = 0.02$  por ejemplo (la distancia es menor a  $\varepsilon$  para cualquier  $\varepsilon > 0.015$ ) además  $s = 0.51$ . Entonces, por el teorema de

Figura 2.11: Conjuntos  $[0, 1]$  y  $M$ .

collage se sigue  $h(L, A) \leq \frac{0.02}{0.49} \approx 0.04$ , es decir, el error entre el conjunto dado  $L = [0, 1]$  y el atractor  $A$  del IFS  $\{[0, 1]; w_1, w_2\}$  es acotado por  $0.04$ .

Una vez definido el concepto de atractor o fractal, el espacio donde son modelables y el teorema fundamental que garantiza la existencia y unicidad de los fractales. En el siguiente capítulo se describen dos grandes aplicaciones, un algoritmo que emplea los sistemas de funciones iteradas para la aproximación de los fractales llamado algoritmo de la máquina copiadora y el teorema del collage, fundamental para la compresión de imágenes mediante fractales.



# Capítulo 3

## Aplicaciones prácticas de fractales

Recordando que los fractales tienen entre sus características la autosimilaridad, donde un pequeño fragmento contiene rasgos, estructuras o patrones del mismo. Existen diferentes métodos para generar la aproximación de un fractal basados en iteraciones. Para comprender como están constituidos y aproximados, debido a que son abstractos se recurre a un método práctico mediante IFS, conocido como el algoritmo de la máquina copiadora [2].

Por otro lado, el objetivo fundamental de esta tesis es aplicar la compresión de imágenes basado en el teorema del collage o montaje, donde se busca estimar un IFS el cual genera un atractor  $A$  muy similar a una imagen dada (la cual es modelada como la aproximación de un fractal).

### 3.1. Aplicación de un IFS: algoritmo de la máquina copiadora

Una vez determinado en la sección 2.5 el teorema para la definición de un IFS, que consiste en una transformación  $W : H(X) \rightarrow H(X)$  en el espacio  $(H(X), h(d))$ , donde  $W$  es dado por transformaciones  $w_1, \dots, w_N$ , se aplica cada transformación  $w_i$  a un subconjunto inicial  $A_0$  y se forma la unión de los resultados obteniendo un nuevo conjunto  $A_1$ . El proceso es aplicado iterativamente, generando a los conjuntos  $A_2, \dots, A_n, \dots$ , obteniendo al atractor  $A = \lim_{n \rightarrow \infty} W^{on}(A_0)$ .

Los ejemplos se efectúan con un número finito de  $n$  iteraciones que dan como resultado la aproximación de un fractal. Como conjunto inicial  $A_0$  utilizamos un solo punto del plano.



**Definición.** Sea  $\{\mathbb{R}^2; w_1, \dots, w_N\}$  un IFS, se selecciona un conjunto compacto inicial  $A_0 \subseteq \mathbb{R}^2$  y un número de iteraciones  $n$ . El *algoritmo de la máquina copiadora* es determinado por el IFS y los parámetros  $A_0, n$  y consiste en la realización de  $n$  aplicaciones iterativas del IFS:

$$A_1 = W(A_0), A_{k+1} = W(A_k) = \bigcup_{j=1}^N w_j(A_k) = W^{o k+1}(A_0), \text{ para } 0 \leq k \leq n-1.$$

El algoritmo genera los conjuntos  $A_1, A_2, \dots, A_n \in H(\mathbb{R}^2)$ , los cuales son aproximaciones del fractal  $A = \lim_{n \rightarrow \infty} (A_n)$ . Dicho algoritmo realiza computacionalmente la aplicación iterativa de las transformaciones afines dadas en el IFS, para un número finito  $n$  de iteraciones, generando como resultado final una aproximación  $A_n$  del fractal  $A$ .

La forma de poner en práctica el algoritmo, es aplicando las transformaciones  $w_1, w_2, \dots, w_N$  a una hoja o documento  $A_0$ , el proceso construye una sucesión de imágenes  $A_1, A_2, \dots, A_n$ . En la figura 3.1, cada conjunto  $A_{k+1}$  es generado por dos copias contraídas, rotadas y trasladadas de  $A_n$ . La transformación que realiza la unión de estas dos copias, es aplicada iterativamente al documento  $A_0$  un número finito  $n$  veces, para obtener una aproximación  $A_n$  de  $A$ . Para generar una aproximación del atractor  $A$ , como conjunto inicial usamos un sólo punto.

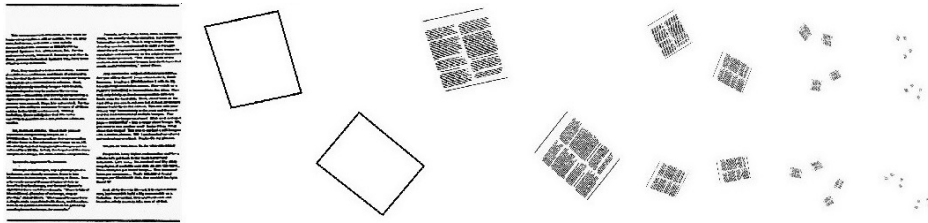


Figura 3.1: Ejemplo del algoritmo de la máquina copiadora

**Ejemplo 1:** En este ejemplo se genera una aproximación del triángulo de Sierpinski [1]. Este fractal puede ser obtenido a través de distintos IFS, tiene propiedades interesantes, por ejemplo, su carácter autosimilar, simétrico y que contiene patrones iguales a distintas escalas. Es empleado el IFS con tres transformaciones afines contractivas, dadas por:

$$w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 50 \\ 1 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 50 \\ 50 \end{pmatrix}$$

Por lo que, la aproximación del triángulo de Sierpinski (figura 3.2) sólo muestra un número finito de  $n$  iteraciones.

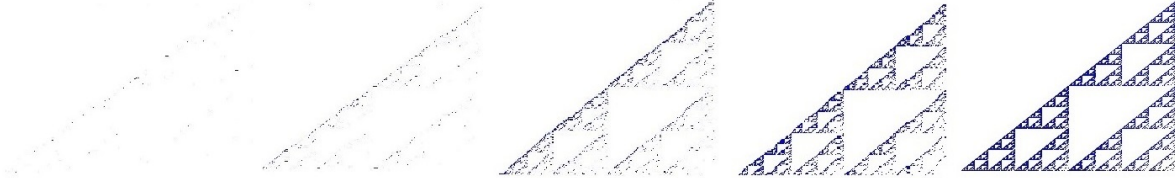


Figura 3.2: Aproximaciones del triángulo de Sierpinski; la última resultado de 10 iteraciones

**Ejemplo 2:** El siguiente ejemplo usa un IFS de cuatro transformaciones afines en  $\mathbb{R}^2$ . El resultado de la aplicación iterativa del IFS es la aproximación de un helecho [1] (figura 3.3).

$$w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0.16 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.2 & -0.26 \\ 0.23 & 0.22 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$$

$$w_4 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.44 \end{pmatrix}$$



Figura 3.3: Aproximaciones de un helecho; la última resultado de 15 iteraciones

**Ejemplo 3:** Consideremos el IFS dado por las contracciones:

$$\begin{aligned}
 w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
 w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.42 & -0.42 \\ 0.42 & 0.42 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.2 \end{pmatrix} \\
 w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.42 & 0.42 \\ -0.42 & 0.42 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.2 \end{pmatrix} \\
 w_4 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0.2 \end{pmatrix}
 \end{aligned}$$

El proceso de obtención de la aproximación al atractor del IFS es similar a los anteriores, la siguiente figura 3.4 muestra el resultado después de un número finito de iteraciones que aproxima un atractor conocido como árbol fractal [1].

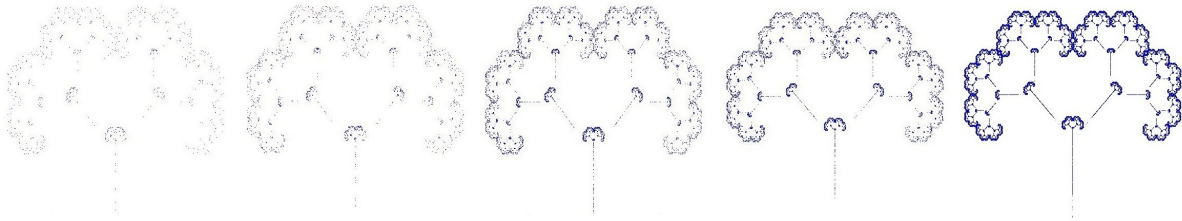


Figura 3.4: Aproximaciones de un árbol fractal; la última resultado de 10 iteraciones.

**Ejemplo 4:** En este ejemplo se genera una aproximación de un dragón fractal dado en [2]. Este fractal es obtenido mediante el IFS con dos transformaciones afines contractivas, dadas por:

$$\begin{aligned}
 w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.44 & 0.32 \\ -0.07 & 0.61 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -3 \\ 70 \end{pmatrix} \\
 w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} -0.82 & 0.16 \\ -0.16 & -0.81 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 137 \\ 14 \end{pmatrix}
 \end{aligned}$$

Finalmente la aproximación de este dragón fractal es presentada en la figura 3.5, donde se muestra un número finito de 15 iteraciones.

**Ejemplo 5:** En este ejemplo se presenta la aproximación de un mosaico fractal dado en



Figura 3.5: Aproximaciones de un dragón fractal; la última resultado de 15 iteraciones

[2], mediante tres contracciones  $w_1, w_2, w_3$ :

$$w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 0.53 \\ -0.48 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 118 \\ 150 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 24 \\ 0 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 24 \\ 64 \end{pmatrix}$$

El proceso de aproximación del atractor se muestra en la figura 3.6 después de 10 iteraciones.

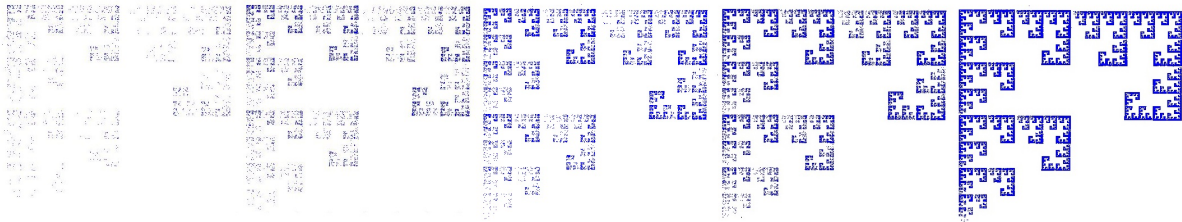


Figura 3.6: Aproximaciones de un mosaico fractal; la última resultado de 10 iteraciones

En general, se presenta la evolución de la aproximación de cada fractal obtenido después de  $n$  iteraciones de la aplicación de un IFS, la cual se detallará en el siguiente capítulo de implementación y resultados. Así mismo se describirá cómo se obtuvieron, cómo fueron implementados en el software, cuántas iteraciones se llevaron a cabo en cada uno de ellos y el tamaño de las imágenes resultantes.

## 3.2. Aplicación del teorema del collage o montaje a la compresión de imágenes

El método para compresión de imágenes esta basado en el teorema del collage [2], el cual no exige que los conjuntos  $w_i(L)$  sean mutuamente disjuntos o que sólo se toquen, sino que también pueden traslaparse. Sin embargo, será mejor la estimación del IFS entre menos se traslapen estos pedazos del collage, aunque claro que  $w_1(L) \cup w_2(L) \cup \dots \cup w_N(L)$  deberá cubrir totalmente a  $L$ . El método de compresión de imágenes mediante fractales se basa en tres puntos [2]:

1. Una imagen digital en el espacio  $\mathcal{R}$  de las imágenes 2D monocromáticas  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . En esta tesis, la imagen es binaria, y el conjunto de píxeles de interés es identificado con el conjunto compacto  $L \subseteq \mathbb{R}^2$ . El objetivo es modelar a  $L$  mediante un fractal dado como atractor de un IFS.
2. Una métrica  $d$  en el espacio  $L$ , tal que  $(L, d)$  es un espacio métrico completo. En esta tesis  $L \subseteq \mathbb{R}^2$  y  $d$  es la métrica Euclidiana.  $L$  es un compacto no vacío en  $\mathbb{R}^2$  y por lo tanto es a la vez un elemento del espacio métrico completo  $(H(\mathbb{R}^2), h)$ , donde  $h$  es la métrica de Hausdorff.
3. Mapeos de contracción  $w_1, w_2, \dots, w_N$  de  $\mathbb{R}^2$  en  $\mathbb{R}^2$ , para cada  $w_i$  existe un factor de contracción  $s_i$ , tal que  $0 \leq s_i < 1$  y cumple:

$$d(w_i(x), w_i(y)) \leq s_i \cdot d(x, y) \forall x, y \in L.$$

Dado el conjunto  $L$  de píxeles de interés de una imagen digital, el primer objetivo es encontrar transformaciones  $w_1, \dots, w_N$ , tal que la unión  $w_1(L) \cup \dots \cup w_N(L)$  genera un conjunto similar o cercano a  $L$ , donde la cercanía es medida por la métrica de Hausdorff. Dicha unión es llamado el *montaje o collage* de  $L$ .

Entonces, como resultado del teorema del collage el sistema de funciones iteradas dado por  $w_1, \dots, w_N$  genera un atractor  $A$  aproximadamente igual al conjunto  $L$ . En los ejemplos siguientes se ilustrará la aplicación para comprimir imágenes cuyo conjunto de píxeles de interés es un fractal  $L$ . El objetivo es re-descubrir al IFS que lo genera como atractor  $A$ . Para generar cada aproximación del atractor  $A$ , usamos un solo punto como conjunto inicial.

**Ejemplo 1:** La compresión de una imagen que contiene al triángulo de Sierpinski se basa en encontrar un IFS que conforma la figura  $L$  en cada una de sus partes autosemejantes. En este ejemplo se estiman tres contracciones  $w_1, w_2, w_3$ . Es decir, se busca representar

a  $L$  mediante su collage  $w_1(L) \cup w_2(L) \cup w_3(L)$ , donde  $w_1(L)$  es la parte azul,  $w_2(L)$  es la parte amarilla y  $w_3(L)$  es la parte verde de la figura 3.7. Cabe mencionarse que el observador puede estimar otras contracciones, es aceptable siempre y cuando el resultado sea semejante.

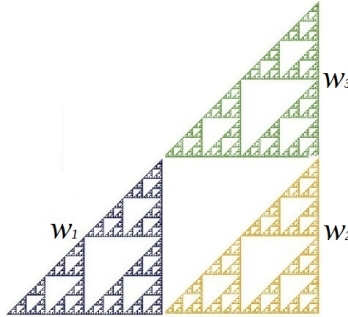


Figura 3.7: Triángulo de Sierpinski como unión de tres partes autosemejantes

La información sobre la imagen quedará estipulada en el IFS estimado, la aplicación reiterada de dichas transformaciones debe ser capaz de obtenerse mediante el algoritmo de la máquina copiadora. Para este ejemplo se estimó un IFS dado por las siguientes contracciones y el resultado de la aproximación para 10 iteraciones con el algoritmo de la máquina copiadora es mostrado en la figura 3.8.

$$w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.4972 & 0 \\ 0 & 0.4972 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0.9946 \\ 0.9946 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.5014 & 0 \\ 0 & 0.5014 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 176.053 \\ 0 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.4957 & 0 \\ 0 & 0.4957 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 176.053 \\ 175.059 \end{pmatrix}$$

**Ejemplo 2:** El siguiente ejemplo se presenta la imagen de una pirámide, en la que se estiman cuatro similitudes  $w_1, w_2, w_3, w_4$ . Es decir, representamos a  $L$  mediante su collage  $w_1(L) \cup w_2(L) \cup w_3(L) \cup w_4(L)$ , donde  $w_1(L)$  es la parte amarilla,  $w_2(L)$  es la parte azul,  $w_3(L)$  es la negra y  $w_4(L)$  es la parte verde, respectivamente (ver figura 3.9).

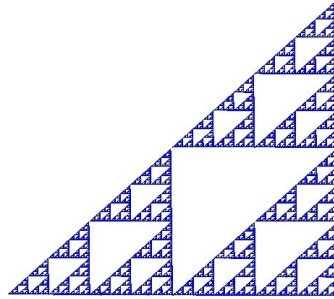


Figura 3.8: Aproximación del triángulo de Sierpinski obtenido como atractor de un IFS estimado con el teorema del collage

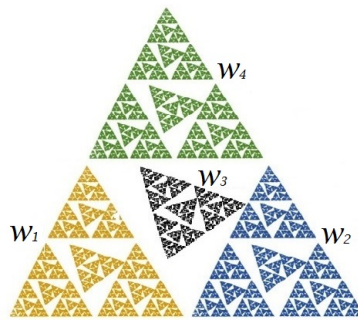


Figura 3.9: Imagen de una pirámide [1] como unión de cuatro partes autosemejantes

Los siguientes mapeos estimados hacen posible la compresión de la figura de la pirámide:

$$w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0,4995 & 0 \\ 0 & 0,4995 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0,4941 & 0 \\ 0 & 0,4941 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 142,471 \\ 0,823 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0,2287 & -0,1964 \\ 0,1964 & 0,2287 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 121,059 \\ 46,117 \end{pmatrix}$$

$$w_4 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0,4955 & 0 \\ 0 & 0,4955 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 61,764 \\ 147,412 \end{pmatrix}$$

Llevando a cabo el algoritmo de la máquina copiadora para este ejemplo, se obtiene la siguiente aproximación con 8 iteraciones (figura 3.10).

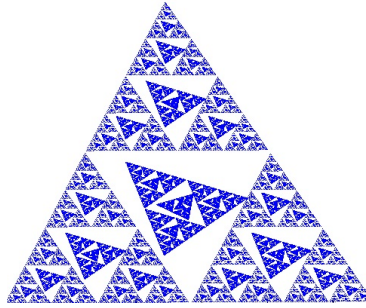


Figura 3.10: Aproximación de la pirámide obtenida como atractor de un IFS estimado con el teorema del collage

**Ejemplo 3:** El siguiente ejemplo presenta una imagen con cinco similitudes  $w_1, w_2, w_3, w_4, w_5$ . Es decir, representamos la imagen  $L$  mediante el collage  $w_1(L) \cup w_2(L) \cup w_3(L) \cup w_4(L) \cup w_5(L)$ , donde  $w_1(L)$  es la parte verde,  $w_2(L)$  es la parte azul,  $w_3(L)$  es la negra,  $w_4(L)$  es la parte roja y  $w_5(L)$  es la parte amarilla respectivamente (ver figura 3.11).

Los siguientes mapeos hacen posible la compresión de la figura 3.11 con cinco contracciones:

$$\begin{aligned} w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.32 & 0 \\ 0 & 0.32 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.32 & 0 \\ 0 & 0.32 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 930 \\ 0 \end{pmatrix} \\ w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.31 & 0 \\ 0 & 0.31 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 465 \\ 465 \end{pmatrix} \\ w_4 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.31 & 0 \\ 0 & 0.31 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 930 \\ 930 \end{pmatrix} \\ w_5 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.32 & 0 \\ 0 & 0.32 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 930 \end{pmatrix} \end{aligned}$$

Llevando a cabo el algoritmo de la máquina copidora para este ejemplo, se obtiene la figura 3.12 aproximada con 9 iteraciones.

**Ejemplo 4:** El siguiente ejemplo presenta la imagen del triángulo de Sierpinski rotado, se estiman tres similitudes  $w_1, w_2, w_3$ . Se representa el triángulo  $L$  mediante su collage  $w_1(L) \cup w_2(L) \cup w_3(L)$ , donde  $w_1(L)$  es la parte verde,  $w_2(L)$  es la parte azul y  $w_3(L)$  es la negra mostrado en la figura 3.13.



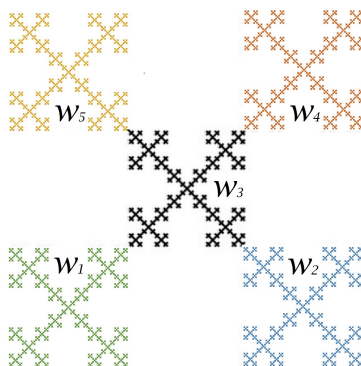


Figura 3.11: Imagen fractal con cinco partes autosemejantes

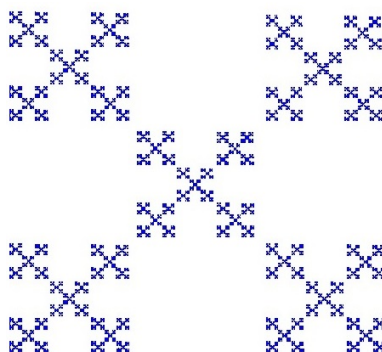


Figura 3.12: Aproximación obtenida como atractor de un IFS estimado con el teorema del collage

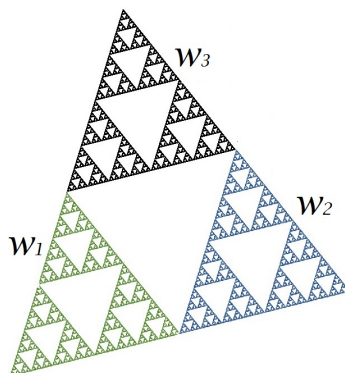


Figura 3.13: Imagen del triángulo de Sierpinski rotado como unión de tres partes autosemejantes

Los siguientes mapeos hacen posible la compresión de la figura del triángulo de Sierpinski con rotación:

$$w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.487576 & 0 \\ 0 & 0.487576 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.486692 & 0 \\ 0 & 0.486692 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 300 \\ 75 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.483271 & 0 \\ 0 & 0.483271 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 100 \\ 325 \end{pmatrix}$$

Llevando a cabo el algoritmo de la máquina copiadora para el triángulo rotado se obtiene la siguiente aproximación con 8 iteraciones mostrada en la figura 3.14.

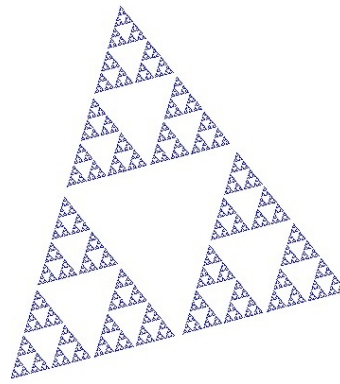


Figura 3.14: Aproximación del triángulo de Sierpinski rotado obtenido como atractor de un IFS estimado con el teorema del collage

**Ejemplo 5:** El siguiente ejemplo presenta una imagen de [1] p. 185, se estiman cuatro similitudes  $w_1, w_2, w_3, w_4$ . Es decir, representamos a  $L$  mediante su collage  $w_1(L) \cup w_2(L) \cup w_3(L) \cup w_4(L)$ , donde  $w_1(L)$  es la parte verde,  $w_2(L)$  es la parte azul,  $w_3(L)$  es la negra y  $w_4(L)$  es la parte verde respectivamente (ver figura 3.15).

El objetivo es obtener la compresión de la imagen, es decir, estimar dichas transformaciones que generan al atractor aproximadamente igual a la figura dada. Los siguientes mapeos hacen posible la compresión:

$$w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.498 & 0 \\ 0 & 0.498 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$w_2 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.498 & 0 \\ 0 & 0.498 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 279 \\ 0 \end{pmatrix}$$

$$w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.3978 & 0 \\ 0 & 0.3978 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 333 \\ 346 \end{pmatrix}$$

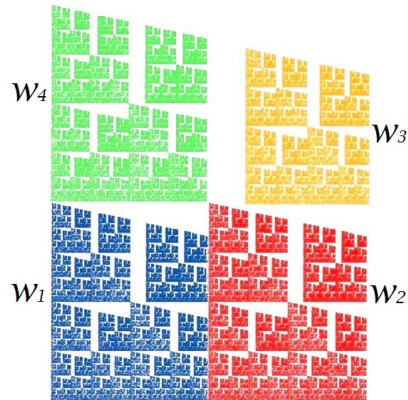


Figura 3.15: Imagen de un castillo [1] como unión de cuatro partes autosemejantes

$$w_4 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.498 & 0 \\ 0 & 0.498 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 348 \end{pmatrix}$$

Cabe mencionarse que los resultados en las aproximaciones son similares, sólo varían los vectores de traslación. En [1] el autor presenta los IFS, sin embargo, los vectores de traslación no tienen unidades de medición, por lo que es justificable que en nuestra implementación los vectores de traslación son medidos en pixeles. La figura 3.16 representa una aproximación del castillo con 8 iteraciones mediante el algoritmo de la máquina copiadora.

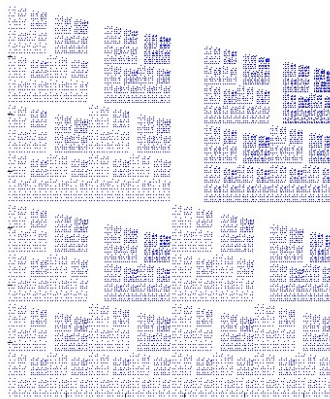


Figura 3.16: Aproximación del castillo obtenido como atractor de un IFS estimado con el teorema del collage

La implementación y resultados del método de compresión de imágenes, son detallados en el siguiente capítulo. Así mismo se explicará cómo se obtuvieron los ejemplos de las figuras 3.8, 3.10, 3.12, 3.14 y 3.16, se describirán los resultados obtenidos, el proceso de estimación, las limitaciones que se presentaron, los errores calculados entre los IFS y los errores en las escalas y mediciones durante el proceso de compresión.



# Capítulo 4

## Implementación y resultados de experimentos

La etapa de implementación es una de las más variadas, debido a que pueden generarse distintas maneras de resolver un problema. En general un algoritmo utiliza herramientas matemáticas y lógicas, un algoritmo según [4] es definido como un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Para que un algoritmo sea implementado es necesario que cumpla las siguientes características: ha de ser preciso, debe indicar el orden en que se ejecutan las instrucciones, debe estar bien definido, es decir, si se realiza dos veces con los mismos datos de entrada se obtiene el mismo resultado, y ha de ser finito, esto es, el algoritmo debe tener un fin, es por esto que en la aproximación de fractales aunque tengan carácter infinito, su implementación debe ser finita.

En general, un algoritmo funciona en tiempo discretizado, es decir, paso a paso, definiendo así una secuencia de estados computacionales por cada entrada válida (la entrada son los datos que se le suministran al algoritmo antes de comenzar).

La ventaja de una implementación es que se puede poner a prueba un algoritmo con distintos datos, de esta forma validar los resultados y detectar fácilmente posibles errores durante dicho algoritmo. En la siguiente sección se describe el algoritmo de la máquina copidora y otro algoritmo para la compresión de imágenes basado en el teorema del collage o montaje los cuales fueron ambos implementados en *Matlab R2012b*, algunas características son:

1. Puede representar imágenes mediante matrices, facilitando su procesamiento.

2. Ofrece un fácil manejo en operaciones y procesos con matrices y vectores.
3. Tiene la capacidad de desplegar gráficos de manera sencilla y flexible.
4. Posee un lenguaje de programación de alto nivel que incluye funciones para realizar operaciones matemáticas de manera sencilla.
5. Es un software de carácter común, que ya ha sido empleado en el departamento por su capacidad de procesamiento de datos y sencilla programación.
6. Tomando en cuenta el tiempo de realización de la tesis, el conocimiento y la sencilla implementación se optó por dicho software.

Sin embargo, el autor de [2] emplea *Turbo C* para el algoritmo de la máquina copiadora y presenta sus resultados. Dando paso a poder validar y comparar los resultados de [2] con los generados en esta tesis.

## 4.1. Experimentos: algoritmo de la máquina copiadora

El algoritmo de la máquina copiadora tiene como objetivo generar los conjuntos  $A_1, \dots, A_n$  los cuales aproximan un fractal  $A$  mediante  $n$  aplicaciones iterativas de un IFS. El algoritmo presenta los siguientes puntos importantes:

- \* El algoritmo usa como conjunto inicial  $A_0$ , un punto en el plano. Sin embargo, el usuario podría definir el conjunto inicial, por ejemplo, cualquier conjunto en el plano, un rectángulo, un círculo, una recta, etc.

- \* Se declaran las transformaciones mediante vectores y matrices. Se representa cada transformación contractiva en una tabla donde cada renglón representa a cada una de ellas con los parámetros  $a, b, c, d, e, f \in \mathbb{R}$ .

- \* El algoritmo se ejecuta mediante un ciclo desde 1 hasta  $n$  iteraciones, donde  $n$  es proporcionado por el usuario.

- \* En cada paso se aplican las transformaciones  $w_i$  al conjunto inicial  $A_0$  aproximando al conjunto de la siguiente manera  $A_1 = w_1(A_0) \cup w_2(A_0) \cup \dots \cup w_n(A_0)$ . El algoritmo aplica nuevamente las transformaciones al conjunto  $A_1$  obteniendo  $A_2 = w_1(A_1) \cup w_2(A_1) \cup \dots \cup w_n(A_1)$  así hasta  $n$  iteraciones aproximando al conjunto  $A_n = w_1(A_{n-1}) \cup w_2(A_{n-1}) \cup \dots \cup w_n(A_{n-1})$ . Las transformaciones definen si las copias serán contraídas, dilatadas, reflejadas, trasladadas o de similitud.

\* Se gráfica la aproximación  $A_n$  de  $A$ . Las coordenadas de la figura van desde  $x = 0, y = 0$  hasta las coordenadas  $x_i, y_i$ , representan el tamaño de la imagen aproximada.

Pueden existir diferentes maneras de implementar el algoritmo, sin embargo, es importante ordenar los pasos indicando los puntos de inicio y fin del proceso. El código presentado es una descripción generalizada y utiliza las convenciones estructurales de un lenguaje de programación *.m* para la implementación del algoritmo de la máquina copiadora:

```

INICIO
% Se borran las variables del espacio de trabajo y limpia la pantalla.
% Se define el conjunto inicial A(:, :, 1)=[ 0 ; 0 ].
% Se ingresan el numero de iteraciones num_it=8
    y el numero de transformaciones num_trans=2.
% Se define cada transformacion afin, mediante su matriz y vector:
matriz_trans(:, :, 1)=[0.498 0;0 0.498];
vector_trans(:, :, 1)=[0;0];
matriz_trans(:, :, 2)=[0.498 0;0 0.498];
vector_trans(:, :, 2)=[279;0];
% Inicializa un ciclo calculando los puntos a generar:
A(:, :, num_trans^num_it)=[ 0 ; 0 ];
    for cont_1=1:num_it
        A_copy=A;
        cont_ALM=0;
        for cont_2=1:( num_trans^(cont_1-1) )
            for cont_3=1:num_trans
                % Almacena en la variable cont_ALM los puntos generados:
                cont_ALM=cont_ALM+1;
                % Aplica el total de puntos a generar al conjunto inicial
                A(:, :, cont_ALM)=matriz_trans(:, :, cont_3)*A_copy(:, :, cont_2)
                    +vector_trans(:, :, cont_3);
            end
        end
    end
% Se guardan en la variable P los puntos generados.
P=A(:, :, cont_ALM);
x=P(1); y=P(2);
% Se conservan en la grafica puntos generados anteriormente
hold on;
% Grafica el atractor aproximado:
plot(x,y)
FIN

```



**Ejemplo 1:** Para la aproximación del triángulo de Sierpinski se usan las transformaciones representadas en la tabla 4.1. Como resultado de la implementación se obtiene una aproximación con 7 iteraciones mostrada en la figura 4.1 con tamaño de  $456 \times 582$  píxeles.

w	a	b	c	d	e	f
1	0.5	0	0	0.5	1	1
2	0.5	0	0	0.5	50	1
3	0.5	0	0	0.5	50	50

Tabla 4.1: Tabla de transformaciones para el triángulo de Sierpinski

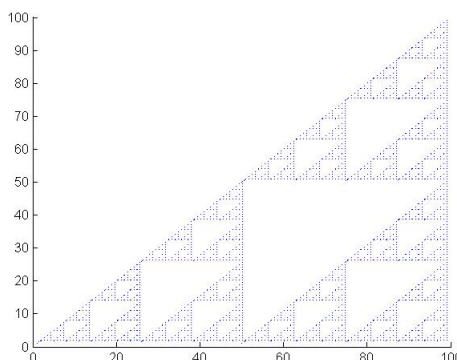


Figura 4.1: Aproximación del triángulo de Sierpinski con 7 iteraciones

**Ejemplo 2:** Las transformaciones para la aproximación de un helecho son dadas por la tabla 4.2. Para este ejemplo, mostrado en la figura 4.2, se realizaron 8 iteraciones obteniendo como resultado una figura de  $454 \times 570$  píxeles.

w	a	b	c	d	e	f
1	0	0	0	0.16	0	0
2	0.85	0.04	-0.04	0.85	0	1.6
3	0.2	-0.26	0.23	0.22	0	1.6
4	-0.15	0.28	0.26	0.24	0	0.44

Tabla 4.2: Tabla de transformaciones para un helecho

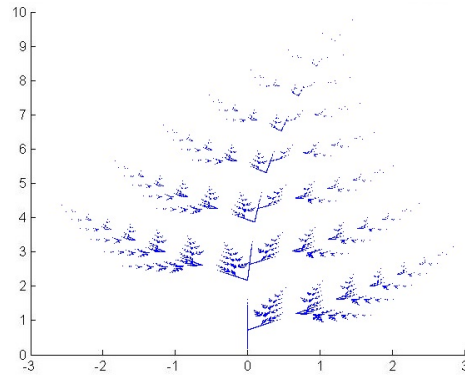


Figura 4.2: Aproximación de un helecho con 8 iteraciones

**Ejemplo 3:** La siguiente tabla 4.3 contiene las transformaciones afines para un árbol fractal. El algoritmo es realizado para 9 iteraciones obteniéndose como resultado la figura 4.3 de  $405 \times 502$  píxeles.

w	a	b	c	d	e	f
1	0	0	0	0.5	0	0
2	0.42	-0.42	0.42	0.42	0	0.2
3	0.42	0.42	-0.42	0.42	0	0.2
4	0.1	0	0	0.1	0	0.2

Tabla 4.3: Tabla de transformaciones para un árbol fractal

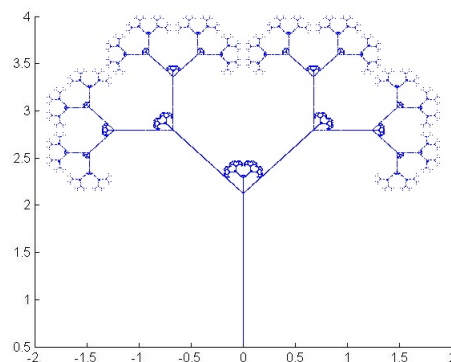


Figura 4.3: Aproximación de un árbol fractal con 9 iteraciones

**Ejemplo 4:** Para la aproximación de un dragón fractal de la figura 4.4 se empleó la

tabla 4.4 con dos transformaciones. El algoritmo llevo a cabo 12 iteraciones obteniendo como resultado una figura de  $504 \times 672$  pixeles.

w	a	b	c	d	e	f
1	0.44	0.32	-0.7	0.61	-3	70
2	-0.82	0.16	-0.16	-0.81	137	14

Tabla 4.4: Tabla de transformaciones para un dragón fractal

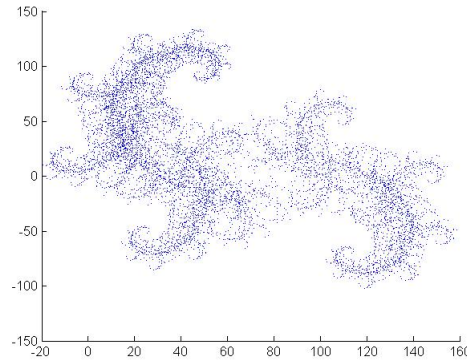


Figura 4.4: Aproximación de un dragón fractal con 12 iteraciones

**Ejemplo 5:** Para la aproximación de un mosaico fractal en la figura 4.5 se empleó la tabla 4.5 con tres transformaciones. El algoritmo empleó 10 iteraciones obteniendo como resultado una figura de  $506 \times 671$  pixeles.

w	a	b	c	d	e	f
1	0	0.53	0	0.5	118	150
2	0.5	0	0	0.5	24	0
3	0.5	0	0	0.5	24	64

Tabla 4.5: Tabla de transformaciones para un mosaico fractal

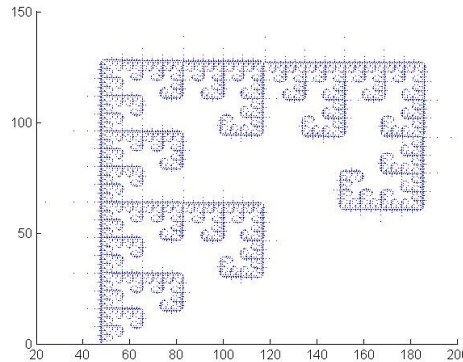


Figura 4.5: Aproximación de un mosaico fractal con 10 iteraciones

## 4.2. Experimentos: compresión de imágenes con transformaciones de similitud

Ahora es momento de presentar los principales puntos para la implementación y la parte experimental para la compresión de imágenes en esta tesis la cual se basa en aplicar el Teorema del Collage o Montaje para generar una aproximación  $A$  de un conjunto dado  $L$ , donde  $A$  es un fractal.

◊ Se descompone al conjunto  $L$  en un collage  $L_1, L_2, \dots, L_k$  donde  $L \approx L_1 \cup L_2 \cup \dots \cup L_k$  y cada  $L_i$  puede ser modelado como imagen de  $L$  bajo una transformación contractiva  $w_i$ , es decir  $L_i = w_i(L)$ . Luego se trata de estimar los parámetros de cada  $w_i$ . En esta sección se emplean transformaciones  $w_i$  de similitud.

◊ Se recomienda que  $L$  contenga sólo similitudes de reflexión, rotación, traslación y escalamiento (transformaciones de similitud), y que sean a lo más cinco.

◊ Para estimar cada  $w_i$  se seleccionan tres puntos del conjunto imagen de la similitud y sus tres puntos correspondientes del conjunto  $L$ . En la figura 4.6  $a, b, c$  son los puntos de  $L$ ,  $a', b', c'$  son los puntos correspondientes a  $w_i(L)$ , es decir, literalmente  $w_i(a) = a', w_i(b) = b', w_i(c) = c'$ .

◊ El factor de escala para cada conjunto imagen de la similitud  $w_i$  se estima empleando la métrica euclidiana. Para la figura 4.6 se miden las longitudes  $\overline{ab}$  y  $\overline{a'b'}$  y el factor de escala es determinado como  $r = \overline{a'b'} / \overline{ab}$ .

◊ Se mide el ángulo de inclinación en el conjunto imagen de la similitud denotado como  $\theta$  (figura 4.7), es decir, si la figura  $w_i(L)$  se encuentra rotada un ángulo  $\theta$  respecto al eje  $x$  (a la figura original  $L$ ).

◊ Para estimar el vector de traslación  $V$ , para el ejemplo de la figura 4.6 se supone

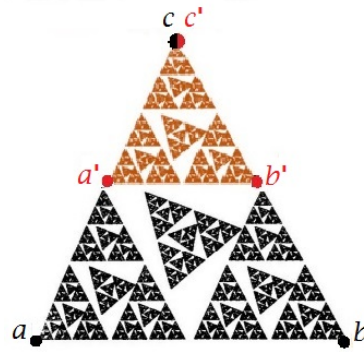


Figura 4.6: Conjunto  $L$  con  $w_i(a) = a'$ ,  $w_i(b) = b'$ ,  $w_i(c) = c'$  similitudes

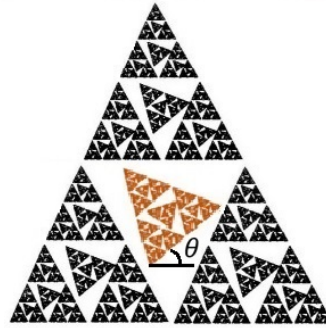


Figura 4.7: Conjunto  $L$  con similitud  $w_i(L)$  rotada respecto a la figura original  $L$

que el punto  $a$  está en la posición  $(0, 0)$ . Por lo tanto para estimar la traslación hasta el punto  $a'$  del conjunto imagen de la similitud se obtiene con las coordenadas del punto  $a'$  que representa la traslación en  $(x', y')$  medida en píxeles.

◊ Una vez estimados los parámetros  $r, \theta$  y  $V$  se obtiene la estimación de la transformación de similitud  $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  del IFS (definida en la sección 2.4) con la función:

$$w(x, y) = \begin{pmatrix} r \cos \theta & -r \sin \theta \\ r \sin \theta & r \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = Tx + V$$

◊ Una vez obtenidas las estimaciones anteriores, se aproxima un IFS con las transformaciones  $w_1, \dots, w_N$  según sea el caso, y se aplica el algoritmo de la máquina copiadora aproximando un atractor  $A_n$ .

◊ Finalmente se compara la figura dada  $L$  y el atractor  $A_n$  generado por el IFS estimado. Se mide la calidad de la estimación de los mapeos, mediante el error absoluto promedio con los parámetros de las matrices  $T_i$  de los mapeos verdaderos y los parámetros

de la matrices  $T'_i$  de los mapeos estimados:

$$e_{prom} = (|a_1 - a'_1| + |b_1 - b'_1| + |c_1 - c'_1| + |d_1 - d'_1| + \dots \\ + |a_N - a'_N| + |b_N - b'_N| + |c_N - c'_N| + |d_N - d'_N|)/4N$$

donde  $a_1, b_1, c_1, d_1, \dots, a_N, b_N, c_N, d_N$  son los parámetros de las matrices de transformación  $T_1$  hasta la matriz  $T_N$  de los mapeos verdaderos y  $a'_1, b'_1, c'_1, d'_1, \dots, a'_N, b'_N, c'_N, d'_N$  son los parámetros de las matrices de transformación  $T'_1$  hasta la matriz  $T'_N$  de los mapeos estimados y  $N$  denota el número total de mapeos del IFS.

La etapa de implementación y experimentos para la compresión puede variar dependiendo del observador, es decir, debido a que pueden estimarse diferentes transformaciones afines, en otras palabras, distintos IFS. Para esto se implementó en *Matlab R2012b* un menú (ver figura 4.8) para estimar las transformaciones del IFS. Posteriormente con el IFS estimado se valida la compresión de imágenes mediante fractales generados por sistemas de funciones iteradas con el algoritmo de la máquina copiadora.

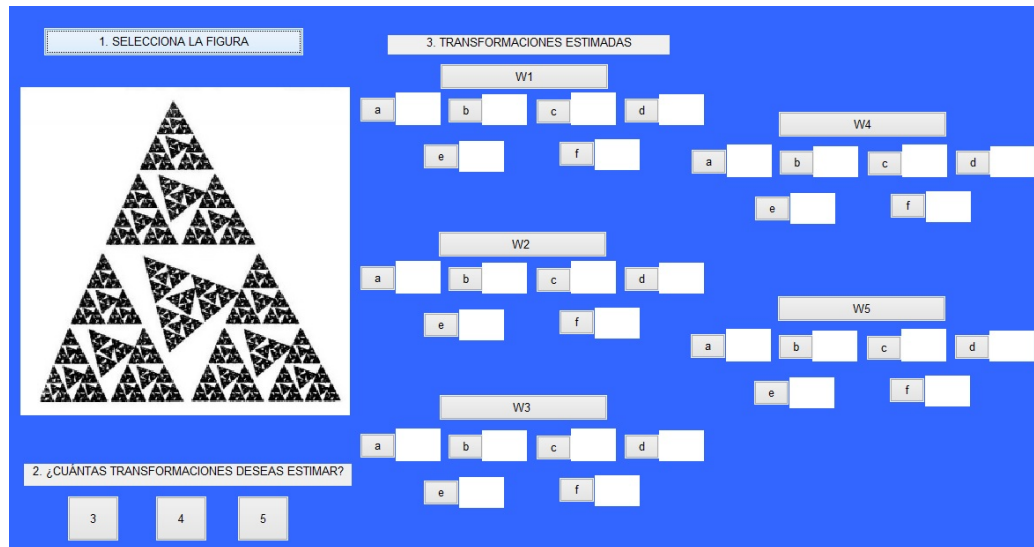


Figura 4.8: Menú en *Matlab* para estimación de los mapeos del IFS

Para llevar a cabo la estimación de las transformaciones de similitud es necesario seguir los siguientes pasos:

1. Se presiona el botón de seleccionar figura, automáticamente permitirá direccionar a la carpeta donde se encuentre la figura o imagen que se desea aproximar mediante una compresión.

2. Se selecciona al número de transformaciones que se desean estimar. En la estimación sólo se permiten 3, 4 o 5 (se recomienda que no sean más de 5, ya que la aproximación tardaría más en procesar los datos, sin embargo, en los casos que hay más transformaciones la aproximación es más detallada).
3. Para estimar cada transformación es necesario seleccionar tres puntos del conjunto total  $L$  y los tres puntos correspondientes del conjunto imagen de la similitud  $w_i$ .
4. Será importante encontrar transformaciones  $w_1, \dots, w_N$  tal que la unión  $w_1(L) \cup \dots \cup w_N(L)$  cubra por completo al conjunto  $L$ , de esta manera generará un conjunto similar o igual al conjunto  $L$ .
5. Una vez estimadas las transformaciones  $w_1, w_2, \dots, w_N$  se desplegarán en el menú con los parámetros  $a, b, c, d, e, f$ , para cada una de ellas.
6. Se valida la compresión mediante el IFS estimado, se implementa en el algoritmo de la máquina copiadora, obteniendo una aproximación del fractal.
7. Se compara visualmente la aproximación de la compresión con la imagen real y se calcula el error absoluto promedio de los términos de la matriz  $T$  de cada transformación del IFS original y del IFS aproximado. Algunos ejemplos son presentados a continuación:

**Ejemplo 1:** Para la compresión de la imagen del triángulo de Sierpinski se estimaron tres contracciones  $w_1, w_2, w_3$  dadas en la siguiente tabla:

w	a	b	c	d	e	f
1	0.4972	0	0	0.4972	0.9946	0.9946
2	0.5014	0	0	0.5014	176.053	0
3	0.4957	0	0	0.4957	176.053	175.059

Tabla 4.6: Tabla de transformaciones de similitud estimadas para el triángulo de Sierpinski

Éstas estimaciones se obtuvieron para la figura 4.9, se considera como la figura total  $L$  y se visualizan en ella solamente tres similitudes. A continuación, se seleccionan tres puntos de la imagen total  $(a, b, c)$  y tres puntos correspondientes para la primer similitud  $(a', b', c')$ , mostrados en la figura 4.9.

Se calcula la distancia euclidiana en Matlab, notese que las distancias euclidianas calculadas entre pixeles son correctas bajo la suposición que los pixeles pertenecen a la

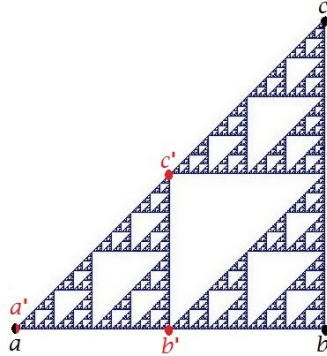


Figura 4.9: Imagen del triángulo de Sierpinski con puntos  $a, b, c$  de la imagen total (figura  $L$ ) y puntos  $a', b', c'$  generados por la similitud  $w_1$

retícula cuadrática donde cada dos pixeles vecinos horizontales o verticales tienen distancia 1,  $\overline{a'b'} = 176 = R'$  y  $\overline{ab} = 354 = R$  y como consecuente se calcula el factor de escala de la similitud respecto a la imagen total  $r = R'/R = \overline{a'b'}/\overline{ab} = 0.4972$ .

Para determinar el ángulo de inclinación  $\theta$  se usa como herramienta la función *seno*, para este ejemplo se mide mediante los puntos  $a'$  y  $b'$ , es decir, se mide si existe una variación en su altura en este caso ambos puntos están alineados respecto al eje  $y$ . Y con la función  $\text{sen}^{-1}$  se mide el ángulo de apertura respecto al eje  $x$  en sentido antihorario en grados. Para esta similitud se estima un ángulo de inclinación  $\theta = 0^\circ$ .

Para estimar el vector de traslación suponemos que el punto  $a$  se encuentra en la coordenada  $(0, 0)$ . Por lo tanto para estimar la traslación se mide la distancia euclidiana del punto  $a$  hasta el punto  $a'$ , en este caso específico se tiene que  $a = a'$  por lo que la traslación resulta  $V = [0; 0]$ . Obtenemos la estimación de la primera contracción:

$$\begin{aligned} w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} r \cos \theta & -r \sin \theta \\ r \sin \theta & r \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \\ &= \begin{pmatrix} 0.4972 \cos(0) & -0.4972 \sin(0) \\ 0.4972 \sin(0) & 0.4972 \cos(0) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0.9946 \\ 0.9946 \end{pmatrix} \\ w_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.4972 & 0 \\ 0 & 0.4972 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0.9946 \\ 0.9946 \end{pmatrix} \end{aligned}$$

De manera similar se estiman las transformaciones  $w_2$  y  $w_3$  que también son contracciones con el mismo factor de similitud. La única variante son las posiciones de



traslación respecto al punto  $a$  con coordenadas  $(0,0)$ . Para  $w_2$  el punto  $b'$  esta situado en la coordenada  $(176.053, 0)$  y para  $w_3$  el punto  $c'$  con coordenada  $(176.053, 175.059)$ . Por lo tanto se justifican las estimaciones de las contracciones presentadas en la tabla 4.6.

Se calcula el error absoluto promedio que mide el nivel de calidad de la compresión.  $N = 3$  representa el número total de mapeos para la aproximación del triángulo de Sierpinski, y los parámetros  $a_1 = 0.5$ ,  $b_1 = 0$ ,  $c_1 = 0$ ,  $d_1 = 0.5$ ,  $a_2 = 0.5$ ,  $b_2 = 0$ ,  $c_2 = 0$ ,  $d_2 = 0.5$ ,  $a_3 = 0.5$ ,  $b_3 = 0$ ,  $c_3 = 0$ ,  $d_3 = 0.5$  son los parámetros de las transformaciones del fractal verdadero (tabla 4.1) y los parámetros  $a'_1 = 0.4972$ ,  $b'_1 = 0$ ,  $c'_1 = 0$ ,  $d'_1 = 0.4972$ ,  $a'_2 = 0.5014$ ,  $b'_2 = 0$ ,  $c'_2 = 0$ ,  $d'_2 = 0.5014$ ,  $a'_3 = 0.4957$ ,  $b'_3 = 0$ ,  $c'_3 = 0$ ,  $d'_3 = 0.4957$  del fractal estimado (tabla 4.6) respectivamente, el error es calculado:

$$\begin{aligned}
 e_{prom} &= (|a_1 - a'_1| + |b_1 - b'_1| + |c_1 - c'_1| + |d_1 - d'_1| \\
 &\quad + |a_2 - a'_2| + |b_2 - b'_2| + |c_2 - c'_2| + |d_2 - d'_2| \\
 &\quad + |a_3 - a'_3| + |b_3 - b'_3| + |c_3 - c'_3| + |d_3 - d'_3|)/4N \\
 &= (|0.5 - 0.4972| + |0 - 0| + |0 - 0| + |0.5 - 0.4972| \\
 &\quad + |0.5 - 0.5014| + |0 - 0| + |0 - 0| + |0.5 - 0.5014| \\
 &\quad + |0.5 - 0.4957| + |0 - 0| + |0 - 0| + |0.5 - 0.4957|)/4 * 3 \\
 e_{prom} &= (0.0056 + 0.0028 + 0.0086)/12 = 0.0015
 \end{aligned}$$

Finalmente se considera un error pequeño y pueden observarse los resultados en la figura 4.10 entre el fractal original y el fractal aproximado.

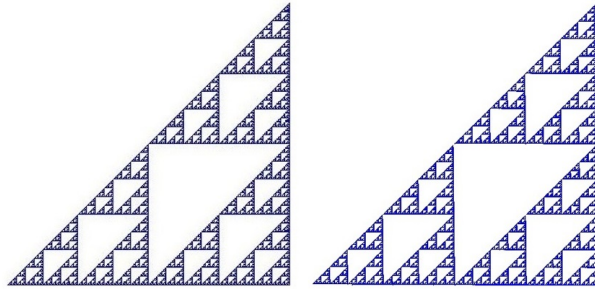


Figura 4.10: Imagen del triángulo de Sierpinski original e imagen del fractal aproximado

**Ejemplo 2:** El siguiente ejemplo es la imagen de un castillo obtenida de [1] p. 185 con su tabla de transformaciones con cuatros contracciones (tabla 4.7), sin embargo

nosotros estimamos con la metodología implementada las contracciones de la tabla 4.8. A continuación se detalla cómo se obtuvieron las estimaciones y verificamos su error.

w	a	b	c	d	e	f
1	0.5	0	0	0.5	0	0
2	0.5	0	0	0.5	2.79	0
3	0.4	0	0	0.4	3.33	3.48
4	0.5	0	0	0.5	0	3.48

Tabla 4.7: Tabla de transformaciones de similitud originales para el castillo

w	a	b	c	d	e	f
1	0.4987	0	0	0.4987	0	0
2	0.4982	0	0	0.4982	279	0
3	0.3978	0	0	0.3978	333	346
4	0.4979	0	0	0.4979	0	348

Tabla 4.8: Tabla de transformaciones de similitud estimadas para el castillo

Para este ejemplo se considera la imagen del castillo como al conjunto  $L$  y se estiman cuatro similitudes. Se seleccionan tres puntos de la imagen total  $L$  y los tres puntos correspondientes a la similitud.

Para calcular el factor de escala de  $w_3$  medimos los segmentos  $\overline{a'b'} = 222$  y  $\overline{ab} = 557$  para obtener  $r = \overline{a'b'}/\overline{ab} = 0.3977$  respecto a la imagen total (ver figura 4.11).

El ángulo de inclinación  $\theta$  es estimado mediante los puntos  $a'$  y  $b'$ , es decir, se observa que ambos se encuentran alineados respecto al eje  $y$ . Por lo que con la función  $\text{sen}^{-1}$  se mide el ángulo de apertura respecto al eje  $x$ , se estima un ángulo  $\theta = 0^\circ$  (medido en sentido antihorario).

Para el vector de traslación, el punto  $a$  se encuentra en  $(0,0)$ . Puede observarse que  $a' = w_3(a)$  se encuentra en  $(333, 346)$ , obteniéndose el vector de traslación  $T = [333 ; 346]$ .

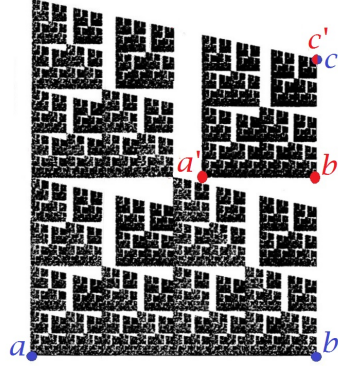


Figura 4.11: Imagen de un castillo con puntos  $a, b, c$  de la imagen total  $L$  y puntos  $a', b', c'$  generados por la similitud  $w_3$

Por lo tanto la contracción es estimada de la siguiente manera:

$$\begin{aligned}
 w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} r \cos \theta & -r \operatorname{sen} \theta \\ r \operatorname{sen} \theta & r \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \\
 &= \begin{pmatrix} 0.3978 \cos(0) & -0.3978 \operatorname{sen}(0) \\ 0.3978 \operatorname{sen}(0) & 0.3978 \cos(0) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 333 \\ 346 \end{pmatrix} \\
 w_3 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} 0.3978 & 0 \\ 0 & 0.3978 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 333 \\ 346 \end{pmatrix}
 \end{aligned}$$

De manera similar se estiman los mapeos  $w_1, w_2, w_4$ . El error absoluto promedio resulta de las siguientes operaciones:

$$\begin{aligned}
 e_{prom} &= (|0.5 - 0.4987| + |0 - 0| + |0 - 0| + |0.5 - 0.4987| \\
 &\quad + |0.5 - 0.4982| + |0 - 0| + |0 - 0| + |0.5 - 0.4982| \\
 &\quad + |0.4 - 0.3978| + |0 - 0| + |0 - 0| + |0.4 - 0.3978| \\
 &\quad + |0.5 - 0.4979| + |0 - 0| + |0 - 0| + |0.5 - 0.4979|)/4 * 4
 \end{aligned}$$

$$e_{prom} = (2(0.0013) + 2(0.0018) + 2(0.0023) + 2(0.0021))/16 = 0.0107$$

Se concluye que los mapeos estimados muestran un resultado eficiente y el error es pequeño y aceptable. En la figura 4.12 se muestra el fractal original y el fractal aproximado estimado, puede observarse que las imágenes son similares casi iguales, por lo que puede concluirse que la compresión fue correcta.

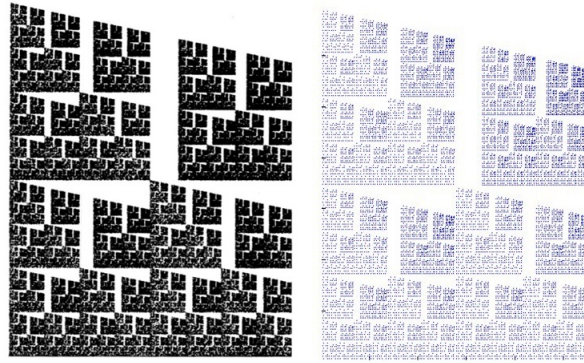


Figura 4.12: Imagen del castillo original e imagen del fractal aproximado

Los siguientes ejemplos corresponden a imágenes sin tablas originales de transformaciones. En estos casos sólo se comparan de forma visual las aproximaciones obtenidas con las imágenes originales.

**Ejemplo 3:** La figura 4.13 presenta la imagen de una pirámide obtenida de [1] p. 187. Se estima el siguiente IFS dado en la tabla 4.9:

w	a	b	c	d	e	f
1	0.4995	0	0	0.4995	0	0
2	0.4941	0	0	0.4941	142.471	0.823
3	0.2287	-0.1964	0.1964	0.2287	121.059	46.117
4	0.4955	0	0	0.4955	61.764	147.412

Tabla 4.9: Tabla transformaciones de similitud estimadas para una pirámide

En este ejemplo no se calcula el error debido a que en [1] no se proporciona ninguna tabla con transformaciones. Sólo se observa el resultado comparando la imagen original y la aproximada en la figura 4.13. Puede observarse que son muy similares, casi iguales.

**Ejemplo 4:** La figura 4.14 presenta la imagen de unas hojas fractales (esquina superior izquierda). Para este ejemplo se comprime la imagen mediante diferentes IFS, presentando en cada caso su aproximación correspondiente. A continuación se presentan las tablas junto con las aproximaciones de un mismo fractal, con diferentes IFS.

Puede observarse en la figura 4.14 cómo con el IFS con tres transformaciones la aproximación es menos detallada y con menor calidad.

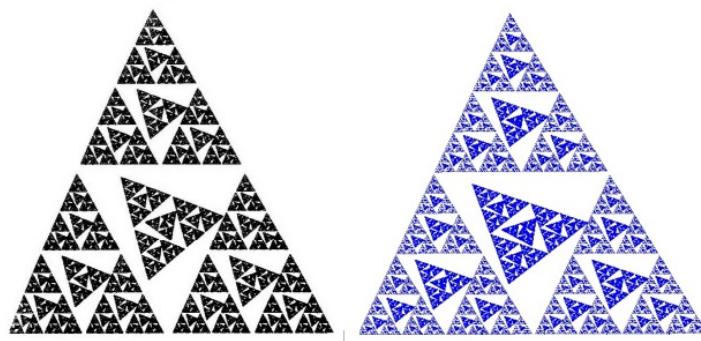


Figura 4.13: Imagen del fractal original (izquierda) e imagen del fractal aproximado (derecha)

w	a	b	c	d	e	f
1	0.69	-0.013	0.013	0.69	2.5	0
2	0.344	-0.364	0.364	0.344	356	12
3	0.013	-0.5	0.5	0.013	536.5	197.32

Tabla 4.10: Tabla con tres transformaciones de similitud estimadas para una hoja fractal

w	a	b	c	d	e	f
1	0.34	-0.013	0.013	0.34	4.8	0
2	0.23	-0.25	0.25	0.23	180	9.6
3	0.33	-0.35	0.35	0.33	252.5	9.6
4	0.47	-0.5	0.5	0.47	360	12

Tabla 4.11: Tabla con cuatro transformaciones de similitud estimadas para una hoja fractal

w	a	b	c	d	e	f
1	0.34	-0.013	0.013	0.34	4.8	0
2	0.23	-0.25	0.25	0.23	180	9.6
3	0.33	-0.35	0.35	0.33	252	9.6
4	0.33	-0.35	0.35	0.33	360	12
5	0.013	-0.49	0.49	0.013	534.22	194.92

Tabla 4.12: Tabla con cinco transformaciones de similitud estimadas para una hoja fractal



Figura 4.14: Imagen de la hoja fractal original (esquina superior izquierda) y hoja fractal aproximada con tres (esquina superior derecha), cuatro (esquina inferior izquierda) y cinco transformaciones de similitud (esquina inferior derecha)

En este ejemplo se presentan a lo más cinco transformaciones, el tiempo de procesamiento es mayor que cuando se aplican sólo tres transformaciones. Sin embargo, puede observarse en la esquina inferior derecha de la figura 4.14 cómo la aproximación es más fina y detallada.

### 4.3. Experimentos: compresión de imágenes con transformaciones afines

Se llevó a cabo la compresión de imágenes con transformaciones de similitud. Sin embargo, existe una manera más general mediante transformaciones afines aplicadas en un IFS para aproximar una imagen real. Para esto recordamos de la sección 2.4 una transformación afín definida como un mapeo  $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ :

$$w(x, y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = (ax + by + e, cx + dy + f)$$

donde  $a, b, c, d, e$  y  $f$  son números reales llamados parámetros.

Este tipo de transformaciones afines son definidas de forma generalizada y tendrán fundamental uso en la compresión de imágenes reales, es decir, imágenes que no tienen una geometría perfecta, no son simétricas o no son fractales, simplemente cualquier objeto con característica autosimilar que se desea comprimir y aproximar mediante un fractal por IFS.

Esta metodología se basa en seleccionar tres puntos del conjunto original  $L$  y tres puntos correspondientes al conjunto de imagen de la transformación afín. Por lo tanto es necesario ejemplificarlo y calcular de manera analítica las estimaciones de los mapeos. En la figura 4.15 se presenta el triángulo de Sierpinski, donde se tienen tres puntos del conjunto original  $P_1, P_2, P_3$  y tres puntos del conjunto imagen de la transformación afín  $P'_1, P'_2, P'_3$ . Para estimar cualquier mapeo de una contracción es necesario estimar los parámetros  $a, b, c, d, e, f$  de la transformación  $w(x, y) = (ax + by + e, cx + dy + f) = (x', y')$ .

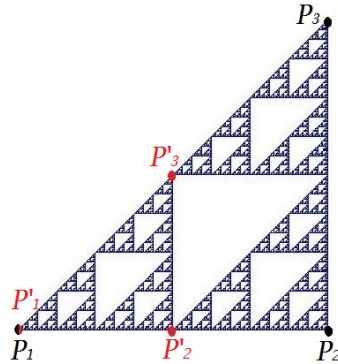


Figura 4.15: Imagen del triángulo de Sierpinski con puntos  $P_1, P_2, P_3$  del conjunto original y puntos  $P'_1, P'_2, P'_3$  del conjunto imagen de la transformación afín  $w_1$

En otras palabras este método usa las coordenadas de los puntos  $P_1, P_2, P_3, P'_1, P'_2, P'_3$  para estimar cada transformación:

$$\begin{aligned} (ax_1 + by_1 + e, cx_1 + dy_1 + f) &= (x'_1, y'_1) \\ (ax_2 + by_2 + e, cx_2 + dy_2 + f) &= (x'_2, y'_2) \\ (ax_3 + by_3 + e, cx_3 + dy_3 + f) &= (x'_3, y'_3) \end{aligned}$$

Por lo tanto se obtiene el siguiente sistema de ecuaciones lineales donde los parámetros

$a, b, c, d, e, f$  son desconocidos:

$$\begin{aligned} ax_1 + by_1 + e &= x'_1 \\ ax_2 + by_2 + e &= x'_2 \\ ax_3 + by_3 + e &= x'_3 \\ cx_1 + dy_1 + f &= y'_1 \\ cx_2 + dy_2 + f &= y'_2 \\ cx_3 + dy_3 + f &= y'_3 \end{aligned}$$

Se representa el sistema de ecuaciones en forma matricial y se observa que el problema es encontrar la solución de un sistema de la forma  $Ax = b$ , para una matriz  $A$ ,  $x$  y  $b$ :

$$\begin{pmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ x_2 & y_2 & 0 & 0 & 1 & 0 \\ x_3 & y_3 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ 0 & 0 & x_2 & y_2 & 0 & 1 \\ 0 & 0 & x_3 & y_3 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ y'_1 \\ y'_2 \\ y'_3 \end{pmatrix}$$

donde  $Ax = b$  es llamada ecuación lineal en  $x$ . A continuación se reporta un método basado en el Teorema de Cramer [12] para resolver este sistema:

**Teorema.** *Teorema de Cramer.* Sea  $A$  una matriz de tipo  $(n, n)$  y  $b \in \mathbb{R}^n$ :

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

El sistema de ecuaciones lineales  $Ax = b$  tiene una única solución  $x$  si  $\det A = |A| \neq 0$ , y esta solución tiene la forma

$$x = \begin{pmatrix} l_1 \\ \vdots \\ l_n \end{pmatrix}, l_j = \frac{\begin{vmatrix} a_{11} & \dots & a_{1j-1} & b_1 & a_{1j+1} & \dots & a_{1n} \\ \vdots & & & \ddots & & & \vdots \\ a_{n1} & \dots & a_{nj-1} & b_n & a_{nj+1} & \dots & a_{nn} \end{vmatrix}}{\det A}$$

donde el numerador de la  $j$ -ésima coordenada  $l_j$  del vector solución  $x$ , es el determinante de la matriz  $A_j$  generada al sustituir la  $j$ -ésima columna de  $A$  por el vector  $b$ .



Resolviendo el sistema de ecuaciones se estiman los parámetros  $a, b, c, d, e, f$  que definirán a cada transformación afín. Para obtener la solución de forma numérica y general se implementó un programa basado en el método de Cramer.

Este programa funciona ingresando las coordenadas de los tres puntos del conjunto imagen  $L$  y los tres puntos correspondientes del conjunto imagen de la transformación afín:

```
% ESTIMACION DE UNA TRANSFORMACION AFIN BASADO EN EL METODO DE CRAMER

clc; clear all; close all;
msgbox('INSERTA LOS 3 PUNTOS DEL CONJUNTO IMAGEN L "EJEMPLO:P=[X Y]"')
P1=input('\n Inserta la coordenada de P1:');
P2=input('\n Inserta la coordenada de P2:');
P3=input('\n Inserta la coordenada de P3:');
msgbox('INSERTA LOS 3 PUNTOS DEL CONJUNTO IMAGEN
      DE LA TRANSFORMACION AFIN "EJEMPLO: P=[X Y]"')
p1=input('\n Inserta la coordenada de p1:');
p2=input('\n Inserta la coordenada de p2:');
p3=input('\n Inserta la coordenada de p3:');

%CONSTRUYE MATRIZ A Y VECTOR b DEL SISTEMA => Ax=b
A=[P1(1) P1(2) 0 0 1 0;P2(1) P2(2) 0 0 1 0;P3(1) P3(2) 0 0 1 0;
   0 0 P1(1) P1(2) 0 1;0 0 P2(1) P2(2) 0 1;0 0 P3(1) P3(2) 0 1];
b=[p1(1);p2(1);p3(1);p1(2);p2(2);p3(2)];
detbase = det(A);
C = A; % guardo valor original de matriz A

%CALCULA LA SOLUCION DEL SISTEMA DE ECUACIONES
disp(sprintf('LA SOLUCION DEL SISTEMA ES'))
cont=1;
N=6;
while(cont<=N)
    C(:,cont)= b; % reemplazo columna i-esima por vector b
    incognita(cont) = det(C)/detbase;
    disp(sprintf(' x(%d) = % .4f',cont,incognita(cont)))
    C = A; % vuelvo a la matriz original
    cont=cont+1;
end
```

Finalmente se calculan los parámetros desconocidos del sistema de ecuaciones, de esta manera se estimará un IFS con el cual se podrá comprimir y aproximar una imagen deseada. Algunos ejemplos son presentados a continuación:

**Ejemplo 1:** En este ejemplo se presenta la estimación y validación de la transformación  $w_1$  de forma generalizada del triángulo de Sierpinski (figura 4.15). Se estima con los puntos del conjunto  $L$ , los cuales son  $P_1 = (1, 1)$ ,  $P_2 = (100, 1)$ ,  $P_3 = (100, 100)$  y los puntos del conjunto imagen de la transformación afín  $P'_1 = (1, 1)$ ,  $P'_2 = (50, 1)$ ,  $P'_3 = (50, 50)$ .

Ahora con la estructura de una transformación afín  $w(x, y) = (ax+by+e, cx+dy+f) = (x', y')$  y los puntos  $P_1, P_2, P_3, P'_1, P'_2, P'_3$  se obtiene el siguiente sistema de ecuaciones

$$\begin{aligned} e &= 1 \\ 100a + e &= 50 \\ 100a + 100b + e &= 50 \\ f &= 1 \\ 100c + f &= 1 \\ 100c + 100d + f &= 50 \end{aligned}$$

Por lo tanto resolviendo el sistema con el método de Cramer, se obtienen los parámetros  $a = 0.5, b = 0, c = 0, d = 0.5, e = 0$  y  $f = 0$  definiendo así la similitud  $w_1$  para el triángulo de Sierpinski tal como se obtuvo con anterioridad con transformaciones de similitud:

$$w_1(x, y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

De forma similar se estiman las transformaciones restantes presentadas en la tabla 4.1, las cuales conforman el collage de la figura 4.16.

Ahora se presentan ejemplos de objetos naturales, es decir, figuras que no son verdaderamente fractales sino imágenes reales de la naturaleza que serán comprimidas y aproximadas mediante un fractal.

**Ejemplo 2:** En el siguiente ejemplo se estima el IFS para un helecho real (figura 4.17) con transformaciones afines y los puntos  $P_1 = (0, 0)$ ,  $P_2 = (7, 7.5)$ ,  $P_3 = (-0.5, 16)$ ,  $P'_1 = (0, 5.5)$ ,  $P'_2 = (4, 10)$ ,  $P'_3 = (-0.5, 16)$  para la transformación  $w_1$ .

Sustituyendo en la transformación afín los puntos correspondientes se obtiene el si-

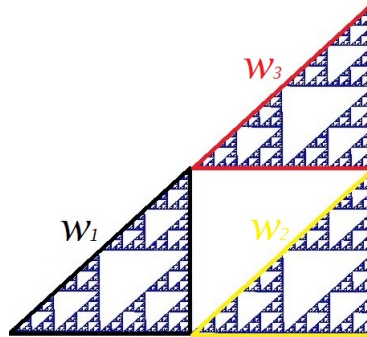


Figura 4.16: Collage empleado para el triángulo de Sierpinski



Figura 4.17: Imagen de un helecho de la naturaleza, y su aproximación mediante un fractal generado por un IFS de cinco transformaciones afines

guiente sistema de ecuaciones

$$\begin{aligned}
 e &= 0 \\
 7a + 7.5b + e &= 4 \\
 -0.5a + 16b + e &= -0.5 \\
 f &= 5.5 \\
 7c + 7.5d + f &= 10 \\
 -0.5c + 16d + f &= 16
 \end{aligned}$$

Resolviendo el sistema, se calculan los parámetros para la transformación  $w_1$ ,  $a = 0.5853$ ,  $b = -0.013$ ,  $c = -0.0583$ ,  $d = 0.6544$ ,  $e = 0$  y  $f = 5.5$ :

$$w_1(x, y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} 0.5853 & -0.013 \\ -0.0583 & 0.6544 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 5.5 \end{pmatrix}$$

De manera similar se estiman los parámetros para las transformaciones  $w_2$ ,  $w_3$ ,  $w_4$  y  $w_5$  que conforman el collage de la figura 4.18, aproximando el IFS de la tabla 4.13.

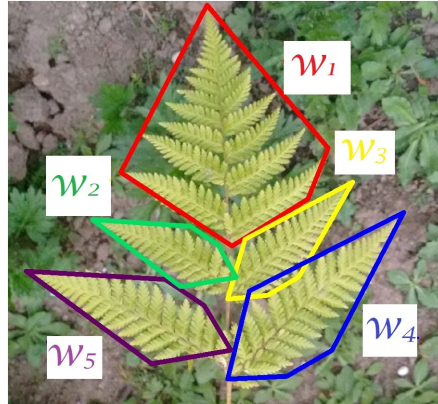


Figura 4.18: Collage empleado para un helecho real

w	a	b	c	d	e	f
1	0.5853	-0.013	-0.0583	0.6544	0	5.5
2	-0.047	0.311	-0.2549	0.3045	0	4
3	0.2181	-0.3369	-0.0821	0.1901	0	4.5
4	-0.0389	0.4363	-0.2829	0.3974	0	1
5	0.3477	-0.4579	0.0864	0.2527	0	1.5

Tabla 4.13: Tabla con cinco transformaciones afines estimadas para un helecho real.

**Ejemplo 3:** En el siguiente ejemplo se estima un IFS para el conjunto de hojas de un árbol real mostrado en la figura 4.19 mediante transformaciones afines.

Se estima la transformación  $w_1$  con parámetros  $a = 0.63$ ,  $b = -0.32$ ,  $c = 0.32$ ,  $d = 0.65$ ,  $e = 21.7$  y  $f = 3.8$ :

$$w_1(x, y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} 0.63 & -0.32 \\ 0.32 & 0.65 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 21.7 \\ 3.8 \end{pmatrix}$$

De forma similar se estiman los parámetros para las transformaciones  $w_2$ ,  $w_3$  y  $w_4$  que en unión definen el collage de la figura 4.20. De esta manera se aproxima un IFS que lleva a cabo la compresión de la imagen real del conjunto de hojas del árbol (tabla 4.14).



Figura 4.19: Imagen de un conjunto de hojas de un árbol de la naturaleza, y su aproximación mediante un fractal generado por un IFS de cuatro transformaciones afines

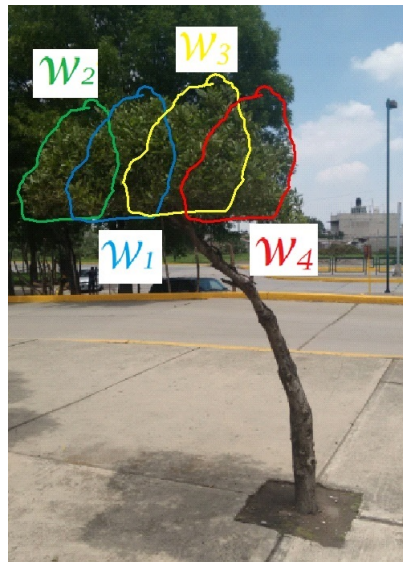


Figura 4.20: Collage empleado para un conjunto de hojas de un árbol

Finalmente se presenta la aproximación del conjunto de hojas del árbol en la figura 4.17, pueden compararse concluyendo que existe similitud entre ambas imágenes.

**Ejemplo 4:** El siguiente ejemplo es para la compresión y aproximación de una hoja real, se estima un IFS para la figura 4.21 con transformaciones afines.

w	a	b	c	d	e	f
1	0.63	-0.29	0.32	0.61	21.7	3.8
2	0.4	-0.25	0.35	0.49	6.2	8.5
3	0.36	-0.26	0.36	0.38	21.7	7
4	0.519	-0.31	0.27	0.5	50.5	7

Tabla 4.14: Tabla con cuatro transformaciones afines estimadas para el conjunto de hojas de un árbol real



Figura 4.21: Imagen de una hoja de la naturaleza, y su aproximación mediante un fractal generado por un IFS de cuatro transformaciones afines

Se estima la transformación  $w_2$  con parámetros  $a = 0.43$ ,  $b = 0.52$ ,  $c = -0.45$ ,  $d = 0.5$ ,  $e = 1.49$  y  $f = -0.75$ :

$$w_2(x, y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} 0.43 & 0.52 \\ 0.45 & 0.5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1.49 \\ -0.75 \end{pmatrix}$$

De forma análoga se estiman las transformaciones  $w_1$ ,  $w_3$  y  $w_4$  (ver collage en la figura 4.22) aproximando el IFS de la tabla 4.15. En la figura 4.21 se muestra la aproximación mediante un fractal generado por un IFS muy similar a la imagen deseada.

Con estos ejemplos se ha validado el método basado en el Teorema del Collage para la compresión de imágenes reales. La eficiencia del método es aceptable debido a la calidad y similaridad de las imágenes aproximadas. Con los resultados expuestos se demuestra que el método es satisfactorio tanto en la teoría y como en la práctica.

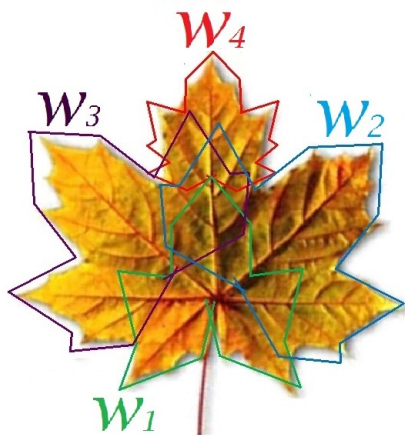


Figura 4.22: Collage empleado para una hoja real

w	a	b	c	d	e	f
1	0.14	0.01	0	0.51	-0.08	-1.3
2	0.43	0.52	-0.45	0.5	1.49	-0.75
3	0.45	-0.49	0.47	0.47	-1.62	-0.74
4	0.49	0	0	0.51	0.02	1.62

Tabla 4.15: Tabla con cuatro transformaciones afines estimadas para una hoja real

**Ejemplo 5:** El siguiente ejemplo es un caso erróneo y fallido para la compresión y aproximación de una hoja mediante transformaciones afines. Las transformaciones estimadas  $w_1$ ,  $w_2$  y  $w_3$  aproximan el IFS de la tabla 4.16.



Figura 4.23: Collage empleado para una hoja real

En la figura 4.23 se muestra la aproximación errónea mediante un fractal generado por el IFS completamente diferente a la imagen deseada y en la figura 4.24 se muestra el

w	a	b	c	d	e	f
1	0.8	-0.1	0.03	0.78	16.6	17.6
2	0.36	0.4	-0.41	0.4	142.78	3.2
3	0.5	0.4	-0.34	0.54	6.4	0

Tabla 4.16: Tabla con cuatro transformaciones afines estimadas para una hoja

collage empleado, sin embargo, no presenta un resultado efectivo.

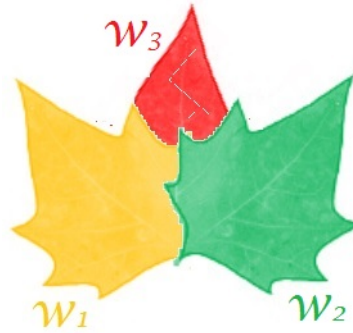


Figura 4.24: Collage empleado para una hoja

**Ejemplo 6:** El último ejemplo fallido es para la compresión y aproximación de las hojas del árbol de la figura 4.25 mediante transformaciones afines.

Se estiman las transformaciones  $w_1$ ,  $w_2$ ,  $w_3$  y  $w_4$  aproximando el IFS de la tabla 4.17. En la figura 4.25 se muestra la aproximación errónea mediante un fractal generado por un IFS completamente diferente a la imagen real.

w	a	b	c	d	e	f
1	0.23	0.6	-0.79	0.33	2.4	4.4
2	0.24	0.54	0.54	0.24	0.2	53.2
3	0.1	0.5	0.4	0.06	6.5	34.5
4	0.34	0.1	0	0.6	0.5	36.5
5	0.33	-0.6	0.5	0.33	-1.2	3.4

Tabla 4.17: Tabla con cinco transformaciones afines estimadas para un árbol



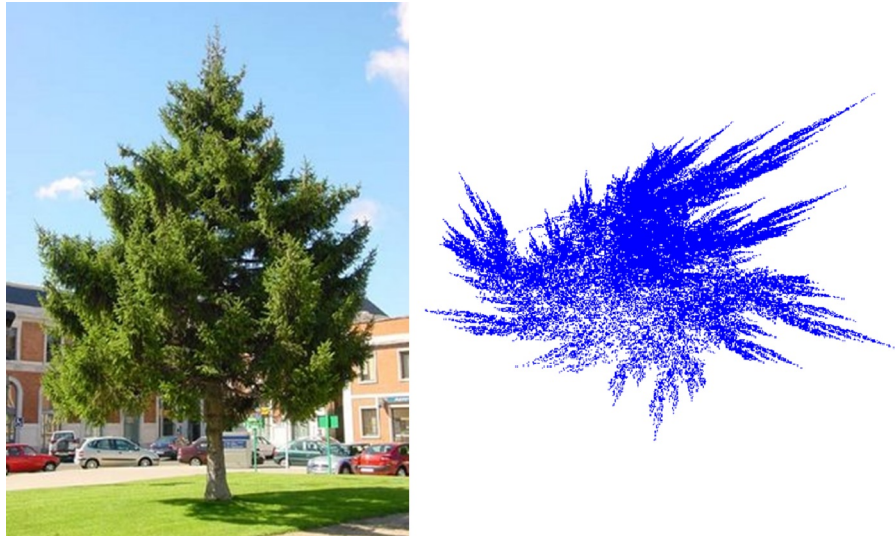


Figura 4.25: Imagen de un árbol de la naturaleza, y su aproximación errónea mediante un fractal generado por un IFS de cinco transformaciones afines

Finalmente en la figura 4.25 se muestra el collage empleado para el árbol. En este ejemplo no se logra una aproximación efectiva.

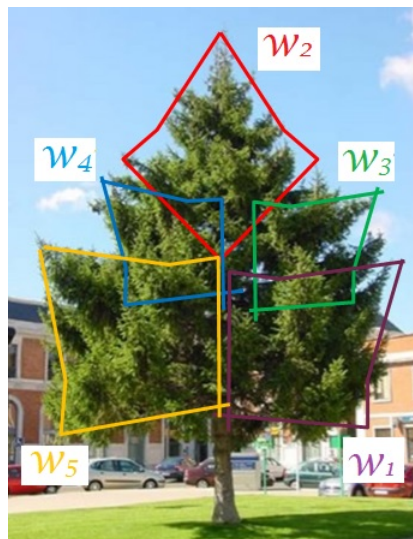


Figura 4.26: Collage empleado para las hojas de un árbol

Los ejemplos presentados, resultados erróneos, fallidos y no aceptables, debido a que las aproximaciones obtenidas no tienen similitud, ni semejanza a las imágenes reales.

# Capítulo 5

## Conclusiones y trabajos futuros

### 5.1. Conclusiones

En esta tesis se logró estudiar y comprender las bases matemáticas que definen a un fractal, así mismo los fundamentos para generarlos por sistemas de funciones iteradas.

En la investigación se realizaron experimentos que generan fractales o atractores a través del algoritmo de la máquina copiadora comprendiendo que son efectuados con un número finito de iteraciones, obteniendo solamente una aproximación de ellos.

Nuestra implementación del algoritmo de la máquina copiadora se considera con flexibilidad, porque el usuario puede fijar, modificar y redefinir las transformaciones afines que se desea aplicar al conjunto inicial y accesible ya que podrá proporcionar un número de iteraciones  $n$  para el IFS.

Se implementó un menú para estimar transformaciones de un IFS sólo con tres puntos del conjunto original y tres puntos del conjunto imagen de cada similitud. Las estimaciones se validan con el algoritmo de la máquina copiadora generando un atractor  $A$ , dando como resultado aproximaciones similares o casi idénticas a las imágenes que se deseaban comprimir.

Para comparar la aproximación lograda de la imagen original mediante un fractal generado por un IFS, se observa que las figuras resultantes conservan los mismos patrones, rasgos o estructuras internas a las imágenes originales. Mientras tanto, de acuerdo al error absoluto promedio se concluye que los ejemplos presentan errores muy pequeños, consiguiendo estimaciones similares a las representaciones originales.

Los primeros experimentos de compresiones de imágenes sólo fueron realizados con transformaciones de similitud de verdaderos fractales. Después se trató de comprimir objetos de la naturaleza (no necesariamente fractales) mediante transformaciones de similitud, y se obtuvieron resultados fallidos. Para poder aproximar a estas figuras mediante un fractal generado por un IFS; fue necesario emplear transformaciones contractivas afines más generales que similitudes. De esta manera se logró aproximar y comprimir la imagen dada con éxito.

En resumen, se cumplieron los objetivos planteados en esta tesis. Incluso se lograron comprimir y aproximar objetos o figuras de la naturaleza como plantas y árboles.

## 5.2. Trabajos futuros

Como trabajos futuros que pueden complementar o ampliar el alcance de esta tesis, se mencionan las siguientes ideas:

1. Esta tesis sólo experimenta con imágenes binarias. Una propuesta de trabajo futuro puede ser la modelación de imágenes de gris, es decir, imágenes monocromáticas, o de imágenes de color mediante fractales. Sin embargo, para eso se deben primero estudiar extensiones del concepto del fractal.
2. Sería interesante validar los resultados del teorema del collage mediante la implementación de la medición de la distancia de Hausdorff entre la imagen dada  $L$  y el atractor  $A$  del IFS.
3. Los fractales generados como atractores de sistemas de funciones iteradas afines constituyen solamente una clase de fractales. Se podrían estudiar y experimentar con fractales generados por el algoritmo de iteración aleatoria o por el algoritmo del tiempo de escape relacionado con sistemas dinámicos, o con fractales dados como conjuntos de Julia o conjuntos de Mandelbrot.

# Bibliografía

- [1] Barnsley, M., *Fractals Everywhere*, Academic Press, USA, 2012.
- [2] Barnsley, M., Hurd, L. P., *Fractal Image Compression*, AK Peters Ltd., USA, 1993.
- [3] Blelloch, G., *Introduction to data compression*, Carnegie Mellon University, 2001, USA.
- [4] Cobo-Yera, Ángel, *Programar desde un punto de vista científico*, Visión Libros, España, 2009.
- [5] Díaz-Oramas W., Loret de Mola G., *Geometría fractal y compresión de imágenes*, Soluciones avanzadas, 1996, Vol. 5, No. 38 , pp. 54-56.
- [6] González, R., Wood R., *Tratamiento digital de imágenes*, Addison-Wesley, 2008, USA.
- [7] Guérin, E., Tosan, E., Baskurt A., *A fractal approximation of curves fractals*, World Scientific, Vol. 9, No.1, pp. 95-103, 2001.
- [8] Huffman, D., *A method for construction of minimum redundancies codes*, ACM Computing Surveys, 1952, IRE 40,9.
- [9] Lelewer, Debra., A., Hirschberg, D., *Data Compression*, ACM Computing Surveys, 1987, Vol. 19, No. 3.
- [10] Moreno-Marín, Juan C., *Experiencia didáctica en Matemática; construir y estudiar fractales*, SUMA: revista sobre la enseñanza y aprendizaje de las matemáticas, 2005, Vol. 6, No. 12, pp. 1-14.
- [11] Munkres, J.R., *Topology: A First Course*, Prentice Hall, USA, 1975.
- [12] Poznyak, A.S., *Advanced mathematical tools for automatic control engineers* Volume One, Elsevier Ltd., UK and Netherlands, 2008.

- [13] Quezada-Len Ariel Osvaldo, *Fractales, más allá de 1D, 2D o 3D*, Revista digital universitaria-UNAM, México, 2002, No. 40, pp. 254-264.
- [14] Valdez-Cepeda Ricardo David, Olivares-Sáenz Emilio, *Geometría fractal en la ciencia del suelo*, Terra Latinoamericana, 1998, Vol.16, No. 3 , pp. 277-288.
- [15] Ziv, J., Lempel, A., *Compression of individual sequences via variable-rate coding*, IEEE Transactionn on Information Theory, 1978.

**FIRMAS DEL JURADO DEL DEPARTAMENTO DE CONTROL  
AUTOMÁTICO APROBANDO LA TESIS**

---

Dra. Petra Wiederhold Grauert

---

Dra. Martha Rzedowski Calderón

---

Dr. Alberto Soria López