Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Abascal Jiménez, Alejandro; Serra-Sagristà, Joan, dir. Compresibilidad de las imágenes. 2023. (958 Enginyeria Informàtica)

This version is available at https://ddd.uab.cat/record/272807

under the terms of the  license

Compresibilidad de las imagenes

Alejandro Abascal Jiménez - 1460774

Resumen– El trabajo consiste en tres partes diferentes pero relacionadas entre sí. Primero se profundiza el estudio de la compresibilidad de datos 1D y 2D. Esto incluye entender qué es la compresibilidad y por qué la entropía de Shannon no sirve a nuestros objetivos. Segundo, se define una métrica de compresibilidad para datos 2D. También se implementan dos métricas de compresibilidad para datos 1D ya existentes (Lempel-Ziv y Transformada de Burrows-Wheeler). Por último, se realizan experimentos para comparar el rendimiento de las métricas con diferentes conjuntos de imágenes y comprobar si la nueva métrica está bien definida o captura bien la compresibilidad de las imágenes.

Palabras clave– Compresión de imágenes, compresión de texto, Lempel-Ziv, Transformada de Burrows-Wheeler, métricas compresibilidad, fuentes de compresibilidad, estudio comparativo

Abstract– The work has been divided into three different but related parts. Firstly, an in-depth study of 1D and 2D data compressibility is carried out. This includes understanding what is compressibility and why Shannon's entropy is not useful for our purposes. Secondly, a 2D data compressibility metric is defined. Also, two existing 1D data compressibility metrics are implemented (Lempel-Ziv and Burrows–Wheeler transform). Finally, a series of experiments are run in order to compare the metrics perfomances with several sets of images and to test whether the new metric is well defined and is able to capture the compressibility of the images.

Keywords– Image compression, text compression, Lempel-Ziv, Burrows-Wheeler Transform, compressibility metrics, compressibility sources, comparative study

1 Estructura del Informe

La estructura de este informe es la siguiente:
En el segundo capítulo se describe el estado del arte, en el tercer capítulo se exponen los objetivos del trabajo, en el cuarto capítulo se describe la metodología seguida, en el quinto la planificación, en el sexto capítulo se explican conceptos relacionados con la linealización de las imágenes, en el séptimo se detallan los conceptos teóricos de la métrica propia y detalles técnicos tales como el esquema del algoritmo que implementa la métrica, el octavo capítulo trata la preparación de los experimentos, el noveno analiza los experimentos y sus resultados, y para acabar tenemos una pequeña conclusión y algunas consideraciones sobre trabajo futuro y limitaciones del trabajo. En el anexo se incluyen las imágenes utilizadas para los experimentos y los resultados de la aplicación de las métricas.

- E-mail de contacto: alejanro.abascal@gmail.com
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Joan Serra Sagristà
- Curso 2022/23

2 Introducción

2.1 Cantidad de datos generados actualmente y por qué debemos comprimir

En los últimos años, el volumen de datos que se generan en el mundo ha aumentado a niveles muy altos. Entre las fuentes que generan estos volúmenes encontramos redes sociales como Twitter o Youtube y científicas como la genómica y la astronomía. También se espera, como es natural, que estas fuentes crezcan de forma considerable en los años que vienen. [8] Otra característica a remarcar es la repetitividad que algunas de estas fuentes muestran. Por ejemplo, dos secuencias de genoma de humanos diferentes se diferencian en 0.1%. [1]

A su vez, los datos, o el procesamiento de estos son de interés, ya sea empresarial o científico. Y como bien sabemos, previo paso al procesamiento está el almacenamiento de los datos. Atendiendo al

volumen de datos generados actualmente, la necesidad de almacenar los datos antes de procesarlos y a que los recursos para almacenar datos son finitos, podemos ver la utilidad de la compresión. Entendida la compresión como la representación de datos utilizando el menor espacio posible es fácil ver que repercute en uso reducido de espacio y consumo de ancho de banda.

2.2 Por qué la entropía de Shannon no captura la repetitividad

Muchos algoritmos de compresión se basan en la entropía de Shannon [7] (codificación Huffman y codificación aritmética). Esta entropía sirve de límite inferior para el ratio de compresión conseguible para una fuente que emite información de acuerdo a un modelo probabilístico. Esta entropía mide la compresibilidad de un texto basándose en la frecuencia de los símbolos, lo que hace que no sea capaz de "capturar" la repetitividad [9]. Para ver esto, basta ver el siguiente ejemplo: si concatenamos dos textos idénticos, la estructura estadística de la concatenación es prácticamente la misma que la de los textos por separado. Por lo que la entropía será la misma. De esto se extrae que son necesarios otros enfoques para comprimir textos repetitivos. Consecuentemente, se han desarrollado métricas que intentan cuantificar esta repetitividad.

2.3 Utilidades o beneficios de tener métricas de compresibilidad

Explicada la necesidad de la compresión de datos y de otros enfoques para comprimir eficientemente textos repetitivos, creemos necesario explicar cuál podría ser un beneficio de las métricas de compresibilidad. El hecho de no haber una métrica estándar para cuantificar la compresibilidad de un texto, de que las métricas no comprimen igual dependiendo del dominio de los datos a comprimir o de ligeras variaciones en el texto (e.g. lectura de izquierda a derecha o derecha a izquierda) y que la carga computacional de las métricas es diferente, nos es natural pensar que dependiendo del tipo de datos habrá un esquema de compresión más adecuado que otros.
Por lo tanto, calcular la compresibilidad de un texto según diferentes esquemas de compresión puede ayudarnos a decidir si comprimir o no, elegir el método más adecuado o comprobar si ciertas modificaciones pueden aumentar la compresión. [4]

2.4 La repetitividad puede no ser la única fuente de compresibilidad

Algunos autores [9] consideran que pueden existir otras fuentes de compresibilidad diferentes a la repetitividad como podría ser la autosimilitud o autosemejanza, y de hecho, han desarrollado medidas utilizando ese concepto. A los autores de este informe les parece, a priori, una fuente más adecuada para la compresión de imágenes o datos 2D. Es esta una característica que queremos explotar en nuestro trabajo.

2.5 Métricas 1D

El campo de las métricas de compresibilidad en datos unidimensionales o textos está más avanzado que su contraparte de datos 2D. Y se basa casi únicamente en medir la repetitividad de un texto. A continuación, una breve explicación de algunas de estas métricas.

2.5.1 Lempel-Ziv

Abraham Lempel y Jacob Ziv propusieron evaluar la complejidad de una secuencia finita de texto como el tamaño del vocabulario generado (número de frases) por la aplicación de un algoritmo que escanea de izquierda a derecha el texto [10]. Estas frases se definen como una subsecuencia o subcadena de carácteres consecutivos que no ha aparecido previamente.

Este método se encuadra dentro de la llamada *compresión por diccionario*, que consiste en la definición de un conjunto de palabras y el posterior análisis del texto de acuerdo a este conjunto.
Su uso está bastante extendido debido a su simpleza y tiene una complejidad computacional baja, lo que lo hace una buena elección para obtener buenas compresiones a bajo coste.
Ejemplo Lempel-Ziv Si *alabaralalabarda* fuese la secuencia la métrica tendría un valor de 6 ya que obtendríamos las siguientes frases: *(a,l,ab,ar,alal,abard)*

2.5.2 Bidirectional Macro Schemes

Esta métrica [14] se puede considerar una extensión de Lempel-Ziv, por lo tanto se encuadra también en las técnicas de *compresión por diccionario.*

El problema de Lempel-Ziv es que realiza únicamente un escaneo de izquierda a derecha. No hay razón para pensar que un escaneo de derecha a izquierda no pueda producir una mejor compresibilidad (i.e. una complejidad más baja). Esta asimetría se resuelve en los *bidirectional macro schemes*, que calculan la complejidad como el vocabulario de menor tamaño generado por escaneos en los dos sentidos (de izquierda a derecha y de derecha a izquierda). La complejidad computacional de este algoritmo es más alta, lo que hace que sea menos popular aún siendo más óptimo. Los *bidirectional macro schemes* aseguran que la métrica de compresibilidad será igual o menor a la generada por Lempel-Ziv.

Ejemplo Bidirectional Macro Schemes Si *alabaralalabarda* fuese la secuencia la métrica tendría un valor de 9 ya que obtendríamos las siguientes frases: *(ala,b,a,r,a,l,alabar,d,a)*.

2.5.3 BWT

La transformación de Burrows-Wheeler [19] se define como una transformación reversible que se aplica a una secuencia y que aumenta la eficiencia de los algoritmos de compresión. La nueva secuencia se consigue generando y ordenando lexicográficamente todas las permutaciones en la secuencia original. Una vez realizado este paso, extraemos la última

columna de cada permutación y de esta forma habremos obtenido la nueva secuencia. La efectividad de esta transformación para mejorar la compresión de ciertos algoritmos se debe a que tiende a juntar los símbolos de forma que la probabilidad de encontrar el mismo carácter en posiciones cercanas aumenta de forma considerable. La métrica de complejidad calcula el número de repeticiones de carácteres consecutivos (1 o más) que aparecen en la secuencia transformada.

Ejemplo BWT Si *aabcddr* fuese la secuencia transformada la métrica calculada sería 5 (aa,b,c,dd,r).

2.5.4 Compresión gramática

La compresión gramática [18] se basa en el siguiente concepto: se genera una gramática a partir de una secuencia de texto. Esta gramática es especial en el sentido de que la totalidad del lenguaje que puede generar es la secuencia de texto en cuestión. La compresión se realiza sobre la gramática y no el texto. La métrica toma en cuenta el tamaño de las reglas de producción de la gramática para calcular su valor. Existen muchas variaciones sobre este concepto.

Ejemplo gramáticas La cadena de texto *alabaralalabarda* se puede generar mediante una gramática con 3 reglas de producción: *A = al , B = Aabar , C = BABda*

2.6 Métricas 2D

El campo, como hemos comentado con anterioridad, no está tan desarrollado como su contraparte y no hay tanta diversidad de métricas. A continuación, una breve explicación de los enfoques utilizados:

2.6.2 Gramáticas 2D

Salvando las distancias, a estas gramáticas [16] podemos considerarlas una extensión de las gramáticas unidimensionales. Se han investigado muchas variantes de este concepto.

Al ser una gramática, su métrica se deriva del tamaño de las reglas de producción de la gramática.

2.6.3 Linearización (curva de orden Z) + aplicación métricas 1D [12]

Este enfoque es el más ingenuo. Se basa en aplicar la curva de orden Z, que convierte datos multidimensionales en datos unidimensionales preservando la localidad de los datos, y posteriormente aplicar métricas de compresibilidad unidimensionales. Es ingenua porque presupone que los conceptos o fuentes de compresibilidad para datos unidimensionales (textos) son los mismos que para datos bidimensionales (imágenes).

3 OBJETIVOS

A continuación sigue una enunciación sobre los principales objetivos del trabajo. Posteriormente se tratará cada objetivo por separado, de forma que tendremos una explicación sobre su razón de ser.

Los objetivos de este trabajo son los siguientes:

Como primer objetivo nos hemos propuesto profundizar en el estudio de la compresibilidad tanto de datos 1D como de datos 2D.

El segundo objetivo consiste en intentar definir una nueva métrica de compresibilidad para datos 2D o mejorar alguna métrica ya existente.

El tercer objetivo consiste en estudiar la nueva métrica o, en su defecto, estudiar las métricas actuales.

3.1 Profundizar en el estudio de la compresibilidad tanto de datos 1D como de datos 2D.

Creemos que este primer objetivo es fundamental y condiciona el éxito de los objetivos siguientes, ya que si se quiere mejorar algo es necesario comprenderlo antes. En nuestro caso concreto, primero debemos entender qué se entiende por compresibilidad, por qué la entropía de Shannon no captura la repetibilidad, en qué se basan las métricas de compresibilidad tanto en datos 1D como datos 2D, qué concepto se enlaza con la compresibilidad en cada tipo de datos, etc.

3.2 Definir una nueva métrica de compresibilidad para datos 2D o mejorar alguna métrica ya existente.

Es el objetivo principal del trabajo. Tanto el objetivo anterior como el siguiente se relacionan con este en cuanto sirven de apoyo para su realización. Su razón de ser la hallamos en el hecho de que, como se ha explicado en el apartado de Introducción, las métricas de compresibilidad tienen gran utilidad, y el estudio de métricas de compresibilidad para datos 2D o imágenes no está tan avanzado como el estudio de métricas de compresibilidad para datos 1D o cadenas de texto. El resultado de este objetivo será la definición y descripción de la implementación de una nueva métrica, de la mejora o actualización de una existente, o en caso negativo, no se habrá podido mejorar una métrica existente ni definir una nueva métrica.

3.3 Estudio de la nueva métrica o, en su defecto, estudio de las métricas actuales.

Este objetivo es el primer y único de carácter práctico. Este consiste en comparar la métrica definida o la métrica mejorada con las métricas de compresibilidad actuales. Para llevar a cabo este objetivo implementaremos el resultado, de haberlo, del objetivo anterior junto a las métricas de compresibilidad actuales. Las métricas de compresibilidad actuales que se implementarán son: alguna (que

sea reachable y que no sea NP) métrica 2D y la linearización de datos 2D (curva de orden Z) + dos métricas 1D (métricas Lempel-Ziv y Transformada de Burrows-Wheeler). Posteriormente, analizaremos sus rendimientos con diferentes tipos de imágenes para poder llevar a cabo un estudio comparativo.

4 METODOLOGÍA

En este apartado describimos las metodologías escogidas para la realización del trabajo y el por qué de tales elecciones.

Este trabajo puede dividirse en dos grandes categorías: parte teórica y parte práctica. Esta diferencia en su naturaleza hace que sean necesarias o justificables metodologías diferentes entre sí.

Dadas las características de las tareas a realizar en la parte teórica (carácter de revisión, exploratorio, creativo e innovador), hemos decidido no seguir una metodología específica ya que no lo consideramos necesario por la simpleza de las tareas a realizar(revisión de la literatura, reflexión sobre ella, exploración de literatura) y por el hecho de que se pueden solapar.

En cuanto a la parte práctica sí que hemos escogido una metodología específica. Teniendo en cuenta que en esta parte se desarrolla software y se realizan experimentos, consideramos oportuno que sea una metodología relacionada con el desarrollo de software.

El software a desarrollar (implementación de varias métricas de compresibilidad) y lo que se pretende hacer con él (aplicar las métricas sobre diferentes conjuntos de imágenes) es bastante simple (su complejidad es baja y no nos importa mucho la experiencia del usuario ya que es un programa científico no lucrativo) y todo su funcionamiento estará claramente definido antes de empezar a programar.

Creemos que un proyecto con estas características (simple, funcionamiento claramente definido, nulo interés por la experiencia de usuario) encaja perfectamente con la metodología de desarrollo en cascada ya que no pretendemos mejorar progresivamente un producto si no implementar lo que ya tenemos en mente.

5 PLANIFICACIÓN

En esta planificación se tiene en cuenta el trabajo ya realizado (revisión del estado del arte) pero, en cambio, no se tiene en cuenta la redacción del informe inicial. Para la planificación también se ha tenido en cuenta la planificación propuesta para el desarrollo del Trabajo de Final de Grado. También han sido atendidas las características de algunos de los objetivos del trabajo a desarrollar (es decir, el carácter creativo de los primeros objetivos y la clara delimitación de los subobjetivos del último objetivo (estudio de la nueva métrica).

Primero se describen algunas consideraciones sobre la naturaleza de los objetivos a tener en cuenta al realizar la planificación. Posteriormente se realiza una descripción de las tareas propuestas para alcanzar los objetivos del trabajo.

Las consideraciones son las siguientes:

- Los dos primeros objetivos pueden enmarcarse dentro de lo que podríamos llamar la fase teórica del trabajo y, como se ha explicado antes, tienen ciertas características que dificultan una planificación muy detallada y estricta. Consideramos que se pueden solapar cuando la revisión del estado del arte está algo avanzada.

- El último objetivo se compone de varias fases que podrían considerarse independientes (no dependen de la finalización de otras fases u objetivos) entre los que encontramos la selección de métricas a desarrollar, su implementación, la selección de los conjuntos de imágenes y experimentación con ellas, siempre que sean las métricas con las que queremos comparar nuestra innovación o mejora. En cuanto a nuestra innovación o mejora (implementación, experimentación), sí que dependemos de la realización de los dos primeros objetivos.

Las tareas (o ausencia de estas) propuestas, desglosadas por objetivo, son las siguientes:

5.1 Profundizar en el estudio de la compresibilidad tanto de datos 1D como de datos 2D.
Este objetivo no está dividido en tareas ya que lo consideramos innecesario dada su naturaleza. Le caracteriza su naturaleza exploratoria y de revisión de la literatura científica al respecto.

5.2 Definir una nueva métrica de compresibilidad para datos 2D o mejorar alguna métrica ya existente.
Consideramos necesario aclarar que, al tratarse de un trabajo teórico y por su naturaleza de revisión, innovadora y creativa, no nos ha resultado fácil descomponer este objetivo en otros subobjetivos y que su ejecución se solapa con la del objetivo anterior (profundizar en el estudio de métricas de compresibilidad). Aún así, creemos que para llevar a cabo este objetivo, como mínimo se deben realizar las siguientes tareas:

- Análisis de las métricas ya existentes y entender sus fortalezas y debilidades

- Análisis de nuevas posibles fuentes de compresibilidad

5.3 Estudio de la nueva métrica o, en su defecto, estudio de las métricas actuales.
Este objetivo se descompone en las siguientes subobjetivos:

- Selección de las métricas de compresibilidad a implementar

- Diseño e implementación de las métricas escogidas

- Selección de los conjuntos de imágenes a utilizar

- Experimentación con las métricas

- Estudio comparativo de las métricas

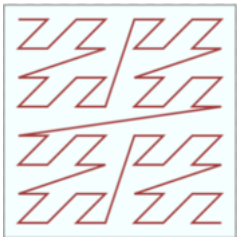


Fig. 1: Curva de morton para imagen 8x8

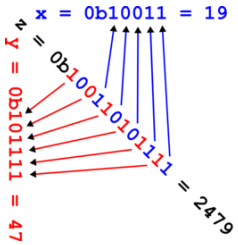


Fig. 2: Bit interleaving

6 LINEALIZACIÓN

Para llevar a cabo el objetivo C (Estudio de la nueva métrica o, en su defecto, estudio de las métricas actuales.), y poder así comparar algunas métricas con la métrica desarrollada, se decidió por utilizar el enfoque del punto 1.6.1: se aplica una linealización a la imagen y, posteriormente, se aplican las métricas de compresibilidad unidimensionales.

Las métricas elegidas han sido Lempel-Ziv y la de la Transformada de Burrows-Wheeler. Para la linealización hemos elegido la curva de orden Z o curva de Morton [1].

Finalmente se ha decidido utilizar unas implementaciones ya existentes de las métricas [2][3] escritas en Python. Para el cómputo de la curva de orden Z se ha utilizado el método del bit interleaving [4].

De forma breve, la curva de Morton (Fig. 1) pertenece al conjunto de space-filling curves, curvas que mapean en una única dimensión un espacio multidimensional y que, a su vez, preservan la localidad de los datos. Para el cómputo de los valores de la curva se ha utilizado el método del bit interleaving (Fig. 22). Es un método computacionalmente simple que funciona para coordenadas 2D de números enteros y produce un único número mediante el entrelazado de los bits que será la posición en la curva. También decir que su orden resultante es equivalente al que obtendríamos al hacer una búsqueda en profundidad de un quadtree.

El método funciona para imágenes que cumplan una serie de condiciones: han de ser cuadradas y tanto la anchura como la altura han de ser potencias de 2 y múltiplos de 4. Esto supone un problema ya que muchas imágenes no cumplen estas condiciones. Para solventar este obstáculo se utilizó el siguiente procedimiento:

Primero calculamos las dimensiones de la primera imagen que cumpla las condiciones y en la que quepa nuestra imagen original. Calculamos la curva de orden Z para la nueva imagen. Siguiendo el orden establecido por la curva, creamos otra curva en la que solo estarán los puntos que pertenecen a la imagen original.

De esta forma preservamos el orden establecido por la curva de orden Z y la conseguimos adaptar a formas no cuadradas y que no sean potencias de 2 ni múltiplos de 4. En la figura se puede observar el proceso para una imagen de 3x3 (Fig. 3).

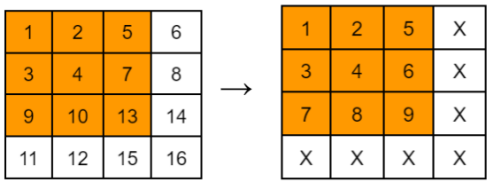


Fig. 3: Ejemplo proceso linealización imagen 3x3

7 MÉTRICA PROPIA

Una vez revisado el contexto del tema a tratar, vimos como insatisfactorias las medidas existentes, ya que pensamos que no se enfocan en las fuentes de compresibilidad adecuadas para tratar imágenes.

Es por eso que decidimos intentar crear una métrica en la cual se tuviese en cuenta otra fuente de compresibilidad completamente diferente. Basándonos en nuestra propia experiencia de percepción visual, y de forma muy intuitiva, nos dimos cuenta de que al visualizar imágenes, el proceso que seguimos puede ser entendido como una discriminación de objetos, en el que ver consiste en identificar todos los objetos o figuras que conforman la imagen.

En relación a esta fuente de compresibilidad también

encontramos útiles algunas ideas de Armando J. Pinho y Paulo J. S. G. Ferreira [6], en el cual tratan el problema de encontrar patrones desconocidos a priori que aparecen múltiples veces en una imagen, y el trabajo de Roberto C. Alamino [5], en el que se computa la complejidad de una imagen basándose en la cantidad promedio de simetría rota de la imagen.

Otra fuente de inspiración a la hora de crear o desarrollar la métrica fueron algunas métricas unidimensionales, específicamente Lempel-Ziv y las basadas en gramáticas. Lempel-Ziv por el hecho de que evalúa la complejidad de una secuencia como el tamaño del vocabulario generado (número de frases) y las basadas en gramáticas porque evalúan la complejidad de una secuencia como el tamaño de las reglas de producción de la gramática que genera el texto.

La relación que encontramos entre nuestra métrica y las expuestas en el último párrafo es que consideramos,en cierta manera, nuestra métrica como una extensión de las métricas unidimensionales aplicada a imágenes. Así como las métricas unidimensionales pretenden hallar las unidades mínimas (frases en el caso de Lempel-Ziv y reglas de producción en las basadas en gramáticas) con las que generar la secuencia en cuestión, nosotros también pretendemos hacer lo mismo con las imágenes. Y la unidad mínima que hemos considerado son regiones contiguas de píxeles con el mismo valor.

Teniendo en cuenta lo expuesto anteriormente, definimos la métrica como la cardinalidad del conjunto de objetos que conforman la imagen. Un objeto se define como un conjunto de píxeles contiguos que tienen el mismo valor. Dos píxeles son contiguos si ambos se encuentran en la 8-adyacencia del otro, o lo que es lo mismo, si sus coordenadas (cada una por separado) no difieren en más de una unidad (e.g. p(1,1) y q(0,2) son contiguos ya que no difieren en más de una unidad en ninguna de las coordenadas). En la Figura 4 se aprecia la 8-adyacencia.

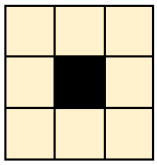


Fig. 4: 8-adyacencia de un píxel, los cuadrados amarillos son los vecinos del píxel negro

Por último, consideraremos como iguales dos objetos si podemos obtener la misma representación matricial tras aplicar ciertas transformaciones geométricas. Las transformaciones geométricas consideradas son la rotación con ángulo θ y la reflexión sobre el eje de abscisas. Remarcar que no consideramos la simetría en el eje Y ya que está cubierta por las transformaciones consideradas. Podemos representar estas transformaciones matricialmente de la siguiente manera:

1. Reflexión sobre el eje de abscisas: $M_X = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
2. Rotación con ángulo θ: $M_\theta = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta & \cos(\theta) \end{pmatrix}$

Dado un punto con coordenadas (x_0, y_0), obtendremos el punto transformado al multiplicar la matriz de la transformación deseada por el vector de coordenadas:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = M \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \text{ con } M \in \{M_X, M_\theta\}$$

Es posible aplicar varias transformaciones a la misma figura, para esto deberemos tener en cuenta que la multiplicación de matrices no es conmutativa ($AB \neq BA$) y que, por lo tanto, deberemos realizar las operaciones en el orden en el que aparecen, de izquierda a derecha, lo que implica un orden inverso a la aplicación de las transformaciones.

Por lo tanto, aplicar primero una rotación y luego una reflexión sobre un punto equivale a lo siguiente:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = M_\theta \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = M_X \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

Que equivale a :

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = M_X M_\theta \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

A continuación se muestra un ejemplo de dos figuras equivalentes y las transformaciones realizadas para ello (Fig. 5).

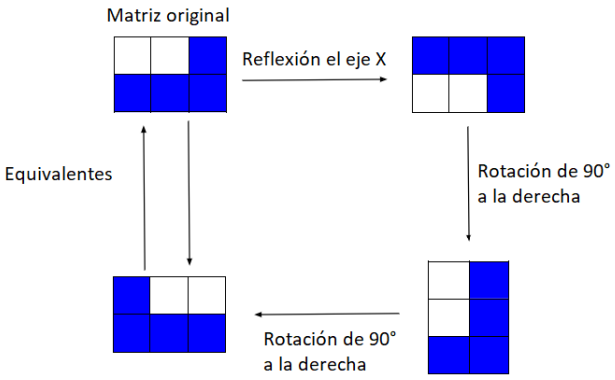


Fig. 5: Figuras equivalentes

A continuación, se describe de forma esquemática el funcionamiento algorítmico de la métrica.

Primero se realiza la fase de exploración o descubrimiento de figuras. Para ello se coge un píxel y se miran los contiguos, si alguno tiene exactamente el mismo valor se añade a la lista de coordenadas del objeto y se sigue buscando entre los píxeles contiguos del píxel añadido, los puntos solo se añaden una vez a la lista de coordenadas. Repitiendo este proceso para todos los valores diferentes de la imagen resultará en tener todos los objetos de la

imagen (tal y como los hemos definido) separados y localizados.

Una vez tenemos las listas de coordenadas, se calcula el rectángulo mínimo que contiene a la figura (esto se hace para facilitar los cálculos y facilidad a la hora de programar) y se transforman las coordenadas de forma que la figura comience en el origen de coordenadas (0,0). Las figuras se representan con matrices.

El último paso consiste en realizar transformaciones geométricas a las figuras y comprobar si son equivalentes al resto. Para poder comparar dos figuras estas deben tener las mismas dimensiones (en este caso la altura y la anchura son intercambiables, una figura de 2x3 se podría comparar con una de 2x3 y una de 3x2) ya que las únicas transformaciones que admitimos son la reflexión y la rotación, y deben estar representadas en forma matricial binaria (1 si hay píxel y 0 si no lo hay) después de haber sido transformadas.

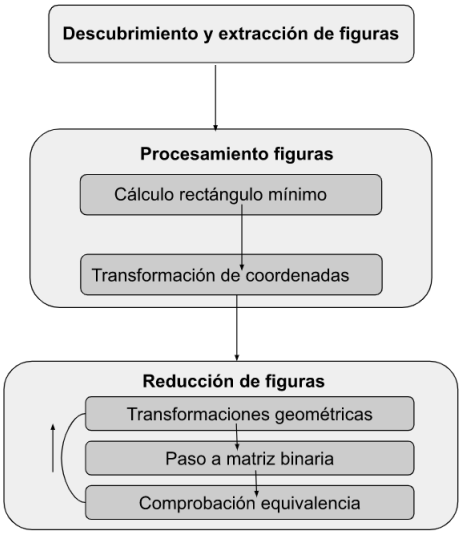


Fig. 6: Esquema algoritmo métrica

Para terminar, un ejemplo del resultado que daría la aplicación de la métrica sobre una imagen artificial muy simple (Fig. 10). En este caso la métrica tendría un valor de 5. Ya que con esas 5 figuras y aplicando las transformaciones necesarias es posible generar el resto.

8 EXPERIMENTOS

Con tal de experimentar con nuestra métrica y ver también cómo se comporta respecto a las métricas escogidas, primero hemos de seleccionar los conjuntos de imágenes sobre los que habremos de aplicar las métricas.

Para poner a prueba a nuestra métrica, primero escogemos imágenes de tamaño reducido (matrices de 3x3 a 7x7) y con muy pocos valores posibles para los píxeles (utilizamos, por simplicidad, una escala

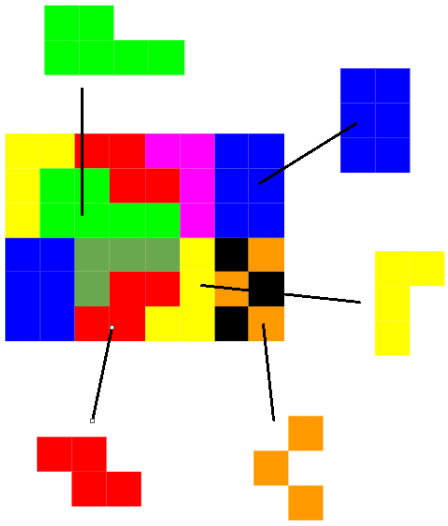


Fig. 7: Ejemplo aplicación métrica

de colores acorde al tamaño de estas matrices y al número de figuras que contiene y no un esquema de colores tradicional (e.g. RGB, grayscale, CMYK)) y con formas simples. Un conjunto de imágenes con estas características son aquellas que se consiguen mediante el relleno de rectángulos usando tetrominós o poliminós (i.e. figuras geométricas que se consiguen uniendo un número determinado de cuadrados iguales de forma que cualquier cuadrado de la figura debe compartir un lado con otro.).

Las imágenes de esta fase han sido generadas manualmente por el autor mediante la creación de matrices (Fig.9) .



Fig. 8: Imagen generada manualmente

Una vez hayamos acabado esta fase de la experimentación pasaremos a una fase con imágenes más complejas, más grandes y con esquema de color tradicional. Un conjunto de imágenes que cumple estas características y que, aún siendo más complejas, no comportan un aumento radical en la complejidad, son las banderas de países. Las banderas suelen presentar un fondo bastante simple, normalmente formado por figuras simples y pocos colores, junto a un logo más complejo, en el que las figuras son más elaboradas y aparecen más colores. También suelen presentar patrones o figuras que se repiten.

Para esta fase se han utilizado las banderas de la Unión Europea, los Estados Unidos de América, Sudáfrica, el Reino Unido (Fig.9) y la del movimiento LGBT. Sus tamaños están alrededor de 150x100 píxeles, lo que supone un aumento significativo en la

complejidad de las imágenes. Al principio queríamos utilizar sus tamaños originales (1200x900) pero computacionalmente nos resultaban demasiado costosas por lo que tuvimos que reducir su tamaño para poder ser más tratables. Lo mismo nos sucedió con el esquema de color a utilizar. Originalmente nuestra intención era experimentar con varios esquemas (RGB, CMYK) pero finalmente nos decidimos por utilizar la escala de grises ya que solo utiliza un valor por píxel, a diferencia de RGB (3) y CMYK (4), lo que aumenta la complejidad computacional considerablemente (temporal y espacial).

Consideramos que estos dos factores han provocado una pérdida de calidad en las imágenes que ha repercutido en la pérdida de calidad de los experimentos.



Fig. 9: Bandera Reino Unido

Por último, se utilizaron dos imágenes del mismo tamaño que las de la segunda fase pero en las que la complejidad, al menos perceptual, es mayor. Se trata de la imagen de fondo del sistema operativo Windows XP (Fig. 10) y de la imagen de un bosque. Estas imágenes adolecen de los mismos problemas que las imágenes de la segunda fase.



Fig. 10: Fondo de pantalla Windows XP

9 Análisis y resultados

En esta sección se procede a la presentación de los resultados obtenidos tras realizar los experimentos y al análisis de estos. Se utiliza una tabla resumen (Fig. 11) para presentar los resultados y hacer más fácil el seguimiento del análisis.

Analizando los resultados de la fase 1, podemos ver que realmente son imágenes bastante simples en cuanto a su complejidad (comparense los valores de las métricas unidimensionales para estas imágenes con los que dan para imágenes de la fase 2 o 3) y que nuestra métrica es capaz de captar tanto la baja complejidad de estas imágenes como de captar diferencias de grado, aunque mínimas, en la complejidad de dos imágenes relativamente parecidas (7x7-multiple y 7x11-multiple).

Las 2 últimas imágenes de la fase 1 (3x3-all-different y 4x4-all-same) se escogieron para ejemplificar una consecuencia lógica de la definición de nuestra métrica, que es el hecho de que una imagen cuyas figuras tienen un único píxel de tamaño tiene la misma complejidad que una imagen en la cual solo hay una figura que tendrá el tamaño de la imagen. Esto no es un problema, y cuadra con nuestra idea de que la imagen 3x3-all-different en sí no es compleja porque la unidad mínima que la forma se puede pensar como un cuadrado al igual que 4x4-all-same (aunque de diferentes tamaños).

Para la fase 2 ya podemos ver que se ha producido un aumento en la complejidad (más bien por el tamaño de las imágenes y la naturaleza propia de las métricas unidimensionales (i.e. tratan las imágenes como secuencias de texto) que no resulta tan notorio para nuestra métrica. Creemos que imágenes con más resolución hubiesen provocado que el resultado de aplicar nuestra métrica hubiera sido más bajo (e.g. la imagen LGBT debería tener 1 de complejidad). Lo que muestra que nuestra métrica es capaz de captar la estructura de la imagen (consideramos esto una fuente de compresibilidad) para imágenes que tienen pocas figuras y las presentan de forma repetida.

	LZ	BWT	Métrica propia
Fase 1			
7x7-multiple	24	32	5
7x11-multiple	40	63	8
3x3-all_different	9	11	1
4x4-all_same	5	4	1
Fase 2			
europe	2511	2248	24
south_africa	3788	1162	12
us_flag	5729	1369	27
united_kingdom	4797	3325	23
lgbt	3584	144	4
Fase 3			
forest	9141	27379	182
windows	9058	19333	326

Fig. 11: Resultados experimentos

Los resultados de la fase 3 nos indican que, efectivamente, se trata de imágenes que exhiben una complejidad mucho más elevada que las de las fases anteriores (tanto para el ojo humano como para las métricas). De estos resultados se desprende que la métrica aumenta de forma más o menos acorde a la complejidad de la imagen. Estas imágenes sufren de los mismos problemas que las de la fase 2.

10 Conclusiones

Hemos podido comprobar la utilidad que tendría una métrica de compresibilidad de imágenes debido al

gran volumen de datos generados actualmente y al coste computacional que requieren las compresiones actuales.
También hemos observado que, si bien el campo de las métricas de compresibilidad para datos unidimensionales es un campo activo y sobre el cual se siguen publicando artículos, el campo de las métricas de compresibilidad bidimensionales se encuentra mucho menos desarrollado.

En cuanto a la métrica desarrollada creemos que puede servir de inspiración para quien se esté enfrentando a un problema similar y que, pese a ser muy rudimentaria y simple en su definición, es capaz de capturar la repetitividad y cierta estructura de la imagen.

Los experimentos muestran un comportamiento similar entre las 3 métricas (i.e ven complejidad alta en imágenes complejas y viceversa.) aunque de no haber sido por lo problemas comentados en las secciones de análisis y experimentos los resultados de los experimentos de la fase 2 para nuestra métrica podrían haber sido mucho más bajos. Creemos que este hecho hubiese dado más peso a la hipótesis de que las imágenes tienen fuentes de compresibilidad diferentes a los textos y que una de estas puede ser la forma de las figuras (tal y como las hemos definido).

11 Trabajo futuro y Limitaciones

Si bien consideramos que nuestro trabajo supone una aportación interesante en el campo de las métricas de compresibilidad para imágenes, creemos también que, dadas las características de este proyecto (tiempo, recursos,alcance, conocimientos), algunos experimentos, consideraciones y posibles continuaciones o mejoras se han quedado fuera del trabajo.

Entre las que encontramos: intentar expandir el concepto de figura y no limitarlo únicamente a nuestra definición, extender las transformaciones posibles (e.g. resizing de la imagen), comparar la métrica con otras diferentes a las del estudio y, a poder ser, que sean métricas para imágenes, optimizar el código, ejecutar la métrica sobre imágenes más grandes y con mayor resolución, y por último, creemos que sería interesante desarrollar un algoritmo de compresión que se base en las ideas desarrolladas en este trabajo.

En cuanto a limitaciones del trabajo, queremos resaltar las indicadas en las secciones de Experimentos y Análisis y resultados. De forma breve, el hecho de haber tenido que reducir el tamaño de las imágenes y convertirlas a escala de grises ha provocado una pérdida de calidad de las imágenes que ha repercutido en la calidad de los resultados de los experimentos.

Agradecimientos

Quiero dar las gracias a todos los que han hecho posible este trabajo: mi tutor, Joan, por introducirme en este campo tan interesante y por sugerir el tema del trabajo. También por su infinita paciencia y buena fe. Y en especial a Emma, por apoyarme incondicionalmente y siempre creer en mí, y a Borja, por su curiosidad infinita y por su ayuda invaluable.

Bibliografia

[1] Anónimo, *Z-order curve*. adr.: https://en.wikipedia.org/wiki/Z-order_curve.

[2] Anónimo, *Burrows-Wheeler Transform Python Documentation*. adr.: https://pypi.org/project/burrowswheeler/#description.

[3] Anónimo, *Lempel-Ziv Complexity Python Documentation*. adr.: https://pypi.org/project/lempel-ziv-complexity/.

[4] Anónimo, *Interleaving bits*. adr.: http://graphics.stanford.edu/~seander/bithacks.html#InterleaveTableObvious.

[5] R. C. Alamino, "Measuring Complexity through Average Symmetry," març de 2015. adr.: http://arxiv.org/abs/1503.07218.

[6] A. J. Pinho i P. J. S. G. Ferreira, "FINDING UNKNOWN REPEATED PATTERNS IN IMAGES."

[7] C. E. Shannon, "A Mathematical Theory of Communication," pàg. 623 - 656.

[8] Z. D. Stephens, S. Y. Lee, F. Faghri et al., "Big data: Astronomical or genomical?" *PLoS Biology*, vol. 13, 7 2015, ISSN: 15457885. DOI: 10.1371/journal.pbio.1002195.

[9] C. U. Gallegos, "New approaches to compressibility and repetitiveness."

[10] A. Lempel i J. Ziv, "On the Complexity of Finite Sequences," *IEEE Transactions on Information Theory*, vol. 22, pàg. 75 - 81, 1 1976, ISSN: 15579654. DOI: 10.1109/TIT.1976.1055501.

[11] S. Raskhodnikova, D. Ron, R. Rubinfeld i A. Smith, "Sublinear Algorithms for Approximating String Compressibility," juny de 2007. adr.: http://arxiv.org/abs/0706.1084.

[12] G. Navarro i C. Urbina, "On Stricter Reachable Repetitiveness Measures*," maig de 2021. adr.: http://arxiv.org/abs/2105.13595.

[13] S. Kreft i G. Navarro, "On compressing and indexing repetitive sequences," vol. 483, abr. de 2013, pàg. 115 - 133. DOI: 10.1016/j.tcs.2012.02.006.

[14] J. A. Storer i T. G. Szymanski, "Data Compression via Textual Substitution," 1982.

[15] G. Navarro, C. Ochoa i N. Prezza, “On the Approximation Ratio of Ordered Parsings,” març de 2018. adr.: http://arxiv.org/abs/1803.09517.

[16] M. Pradella, A. Cherubini i S. C. Reghizzi, “A unifying approach to picture grammars,” oct. de 2009. adr.: http://arxiv.org/abs/0910.2829.

[17] B. Langmead, “High order entropy.” adr.: http://math.cornell.edu/~mec/2003-2004/cryptography/subs/digraphs.html.

[18] J. C. Kieffer i E. H. Yang, “Grammar-based codes: a new class of universal lossless source codes,” *IEEE Transactions on Information Theory*, vol. 46, pàg. 737 - 754, 3 maig de 2000, ISSN: 00189448. DOI: 10.1109/18.841160.

[19] M Burrows i D. J. Wheeler, “A Block-sorting Lossless Data Compression Algorithm,” 1994.

[20] U. A. de Barcelona., “Citacions bibliogràfiques segons el model IEEE,” 2022. adr.: https://ddd.uab.cat/record/128946.

A ANEXOS: IMÁGENES UTILIZADAS PARA LOS EXPERIMENTOS

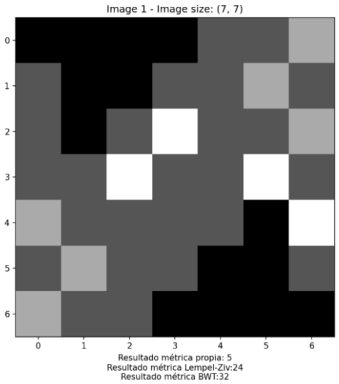


Fig. 12: Imagen 1

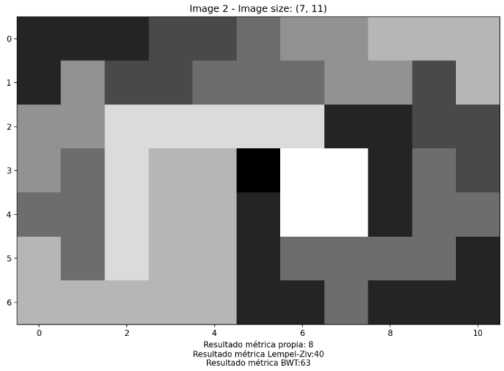


Fig. 13: Imagen 2

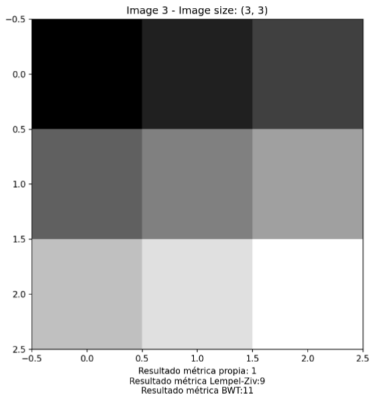


Fig. 14: Imagen 3

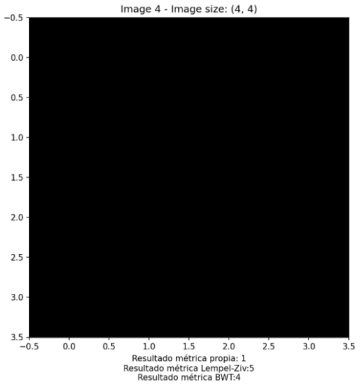


Fig. 15: Imagen 4

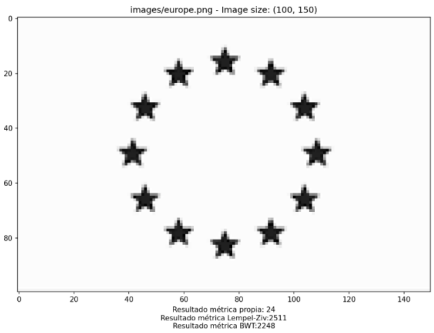


Fig. 16: Europa

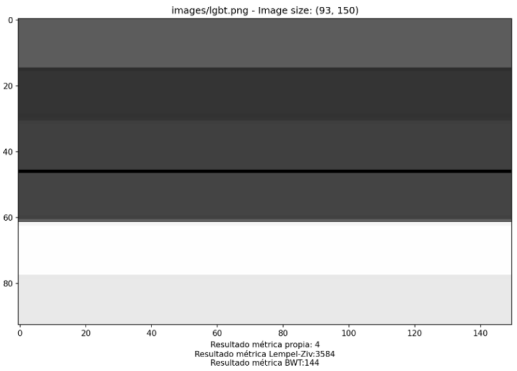


Fig. 17: LGBT

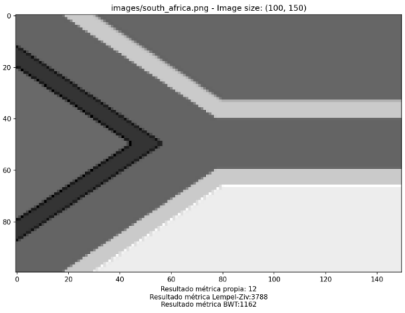


Fig. 18: Sudáfrica

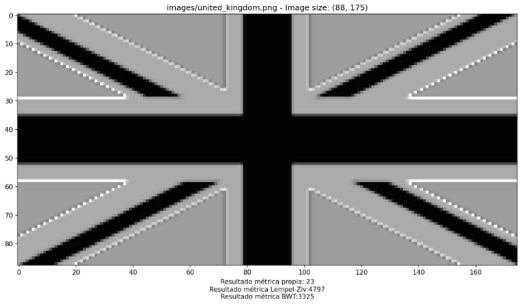


Fig. 19: Reino Unido

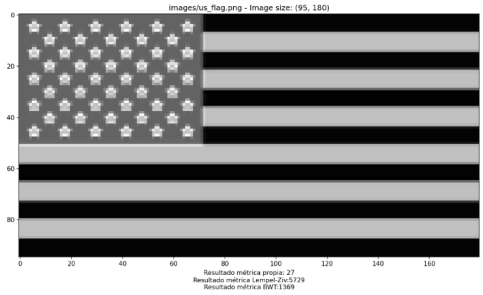


Fig. 20: Estados Unidos de América

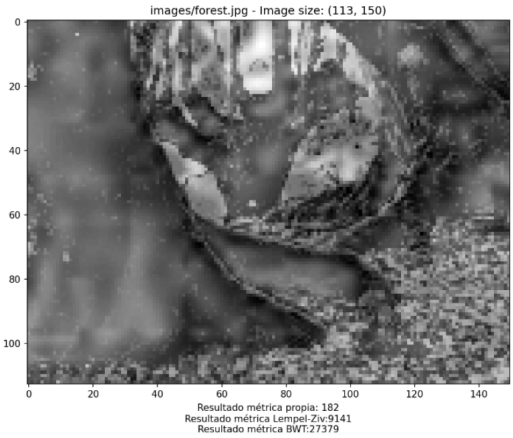


Fig. 21: Bosque

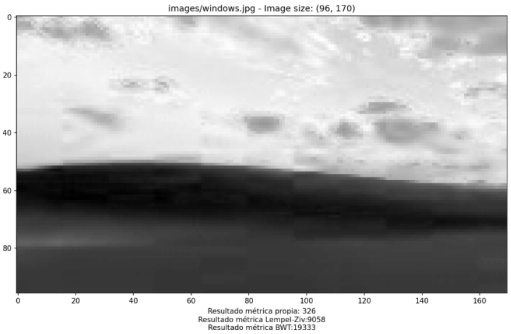


Fig. 22: Windows XP