
CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE MATEMÁTICAS Y FÍSICA

**TESIS**

# COMPRESIÓN DE IMÁGENES DE DOCUMENTOS BASADA EN MÉTODOS CLÚSTER

PRESENTA

Luis Fernando Muñoz Pérez

PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS

COMITÉ TUTORAL

Tutor: Dr. Antonio Guerrero Díaz de León

Asesor: Dr. Hermilo Sánchez Cruz
Asesor: Dr. Rogelio Salinas Gutiérrez

Aguascalientes, Ags., 05 de abril de 2018

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. José de Jesús Ruiz Gallegos
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
PRESENTE

Por medio de la presente, en mi calidad de tutor designado del estudiante **LUIS FERNANDO MUÑOZ PÉREZ** con **ID 108852** quién realizó la tesis titulada: **COMPRESIÓN DE IMÁGENES DE DOCUMENTOS BASADA EN MÉTODOS CLÚSTER** y con fundamento en el Artículo 175, Apartado II del reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y, sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., 05 de Abril de 2018

Dr. José Antonio Guerrero Díaz de León

**c.c.p.- Interesado**
**c.c.p.- Secretaría de Investigación y Posgrado**
**c.c.p.- Jefatura del Depto. de Matemáticas y Física**
**c.c.p.- Consejero Académico**
**c.c.p.- Minuta Secretario Técnico**

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. José de Jesús Ruiz Gallegos
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
PRESENTE

Por medio de la presente, en mi calidad de asesor designado del estudiante **LUIS FERNANDO MUÑOZ PÉREZ** con **ID 108852** quién realizó la tesis titulada: **COMPRESIÓN DE IMÁGENES DE DOCUMENTOS BASADA EN MÉTODOS CLÚSTER** y con fundamento en el Artículo 175, Apartado II del reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y, sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., 05 de Abril de 2018

Dr. Hermilo Sánchez Cruz

**c.c.p.- Interesado**
**c.c.p.- Secretaría de Investigación y Posgrado**
**c.c.p.- Jefatura del Depto. de Matemáticas y Física**
**c.c.p.- Consejero Académico**
**c.c.p.- Minuta Secretario Técnico**

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. José de Jesús Ruiz Gallegos
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
PRESENTE

Por medio de la presente, en mi calidad de asesor designado del estudiante **LUIS FERNANDO MUÑOZ PÉREZ** con **ID 108852** quién realizó la tesis titulada: **COMPRESIÓN DE IMÁGENES DE DOCUMENTOS BASADA EN MÉTODOS CLÚSTER** y con fundamento en el Artículo 175, Apartado II del reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla, y así continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y, sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., 05 de Abril de 2018

Dr. Rogelio Salinas Gutiérrez

**c.c.p.- Interesado**
**c.c.p.- Secretaría de Investigación y Posgrado**
**c.c.p.- Jefatura del Depto. de Matemáticas y Física**
**c.c.p.- Consejero Académico**
**c.c.p.- Minuta Secretario Técnico**

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

**LUIS FERNANDO MUÑOZ PÉREZ**
**MAESTRÍA EN CIENCIAS CON OPCIÓN A LA COMPUTACIÓN Y MATEMÁTICAS APLICADAS**
**P R E S E N T E.**

Estimado alumno:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **"COMPRESIÓN DE IMÁGENES DE DOCUMENTOS BASADA EN MÉTODOS CLÚSTER",** hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

**A T E N T A M E N T E**
Aguascalientes, Ags., a 06 de abril de 2018
*"Se lumen proferre"*
**EL DECANO**

**M. en C. JOSÉ DE JESÚS RUÍZ GALLEGOS**

c.c.p.- Archivo.

## Agradecimientos

*"La raza humana necesita un desafío intelectual.*
*Debe ser aburrido ser Dios y no tener nada que descubrir."*
Stephen Hawking

Gracias

A la Universidad Autónoma de Aguascalientes por las puertas abiertas, sin distinción o restricción alguna, para quienes buscamos ampliar nuestras capacidades.

Al Centro de Ciencias Básicas por permitirme incursionar en la rama de las Matemáticas, para complementar las bases de mi formación académica profesional.

Al Conacyt por darme la oportunidad y el apoyo para realizar esta Maestría dentro del Programa Nacional de Posgrados de Calidad.

A mi tutor, el Dr. José Antonio Guerrero Díaz de León, por impregnarme su gran capacidad para sorprenderse.

A mis compañeros de generación, que me acompañaron a lo largo de la maestría.

## Dedicatorias

*"En las matemáticas no entiendes las cosas. Te acostumbras a ellas."*
Johann von Neumann.

Para mi esposa, por impulsarme a ser mejor, sin la cual no sería lo que soy.

A mi madre, por creer en mi antes que yo, sin la cual no sería.

A mis hermanos, que forjaron mi esencia, sin los cuales no sería lo que fui.

Y a mi hija, que espero con ansias.

# Índice

## Lista de Tablas

## Lista de Figuras

## Resumen

En la presente tesis se introduce un método eficiente para la compresión con pérdida de información diseñado para imágenes de documentos de texto binivel digitalizados. El método utiliza un diccionario conformado por representantes de clase que es generado utilizando un criterio de mínima entropía. El algoritmo identifica inicialmente los diferentes símbolos contenidos en la imagen de documento, y posteriormente los símbolos son agrupados en clases por medio de un algoritmo de clusterización, particularmente el agrupamiento jerárquico haciendo uso de una distancia de similaridad. Para cada clase, se selecciona un representante utilizando el principio de entropía mínima. La técnica crea un archivo de texto en el que cada objeto que pertenece a una clase es reemplazado por su representante de clase junto con su referencia. Finalmente, éste archivo resultante es comprimido con ayuda del archivador Paq8; compresor sin pérdida de información que usa un algoritmo de mezcla de contexto y así detectar la redundancia existente en el archivo. El rendimiento del algoritmo propuesto se evalúa utilizando archivos digitalizados de una base de datos estándar propuesta para la compresión de documentos por el Comité Consultivo para la Telegrafía y Telefonía Internacional (CCITT - Consultative Commitee for International Telephony and Telegraphy) en sus diferentes resoluciones. Se realizan comparaciones con otros algoritmos de última generación. Nuestros resultados establecen cuantitativamente que nuestra metodología propuesta es una técnica con una razón de compresión menor.

## Abstract

In the present thesis is introduced an efficient method for lossy compression of digitalized bilevel image documents. The method uses a dictionary which consists of class representative defined using a minimum entropy criterion. The algorithm initially identifies the different symbols contained in a image document, and then the symbols are grouped in classes by means of a clustering algorithm, particularly hierarchic clustering and suitable similarity distances. For each class, a representative is selected using the principle of minimum entropy. The technique creates a file in which every object belonging to a class is replaced by its class representative and his reference, as well. Finally, the resulting file is compressed with the archiver Paq8, a compressor lossy that uses a context mixing algorithm. The performance of the proposed algorithm is assessed using digitized files from a standard database for document compression along with different resolutions. Comparisons against other state-of-the-art algorithms are performed in this manuscript. The results establish quantitatively that the presented methodology is a more efficient technique.

# Capítulo 1

# Introducción

En la segunda mitad de los años cuarenta, en el Instituto de Tecnología de Massachusetts (MIT) se incorporó como estudiante de ingeniería Claude Elwood Shannon. En julio de 1948 Claude E. Shannon decidió recoger todas sus investigaciones en una obra y publicó un artículo en la revista Bell System Technical Journal bajo el título de "A Mathematical Theory of Communication", donde presentó los trabajos precedentes y algunas ideas sobre la medida de la información articulándolas dentro de una teoría conocida como teoría matemática de la información [21].

En 1949 este artículo junto a un prólogo de Warren Weaver fue publicado como libro con un título casi idéntico [22] por la University of Illinois Press. Esta obra se presentó divida en dos partes. En la primera parte, recogía la contribución de Weaver donde se incluía un modelo teórico que intentaba representar los elementos y las relaciones implicadas en el flujo de información. En la segunda parte, Shannon presentó y desarrolló una teoría que tenía como objetivo principal la definición matemática de todas aquellas magnitudes que intervienen en las situaciones en las que se produce un flujo o transmisión de información, y conseguir, a partir de esas definiciones un cálculo de la cantidad de información que puede ser transportada a través de un canal y la identificación, además, de las formas de maximizar la eficacia de ese proceso. A éste método es a lo que llamamos hoy en día entropía de la información o entropía de Shannon.

Una de las aplicaciones de la teoría de la información son las imágenes de documentos que se comprimen para su transmisión a través de un correo electrónico o como parte de los procedimientos de almacenamiento de datos. La compresión de los datos hace posible completar la transmisión en menos tiempo.

## 1.1 Compresores de imágenes

Un método de compresión de imágenes está normalmente diseñado para un tipo específico de imagen, por ejemplo nos encontramos con compresores de imágenes binivel, donde los píxeles pueden tener uno de dos valores (normalmente 0 o 1) . A continuación se describen a grandes rasgos los compresores de imágenes con pérdida de información más conocidos y/o usados.

**Grupo 4 (G4)** . Los documentos transferidos entre las máquinas de fax se envían como mapas de bits, por lo que, cuando esas máquinas llegaron a ser populares, se hizo necesario un método de compresión estándar de datos. La Unión Internacional de Comunicaciones ITU propuso y desarrolló varios métodos. Los primeros estándares de compresión de datos desarrollados por la ITU fueron el `T2` (también conocido como `Grupo 1`) y el `T3` (`Grupo 2`) [14]. Éstos se han quedado obsoletos y han sido sustituidos por el `T4` (`Grupo 3`) y el `T6` (`Grupo 4`) [8].

Una máquina de fax escanea un documento línea a línea, convirtiendo cada línea en pequeños puntos blancos y negros, llamados pels (Picture Element)[20]. Para obtener el código del Grupo 4, se cuentan las rachas de pels blancos y negros, y se utiliza el algoritmo de Huffman para asignar un código de tamaño variable a cada racha.

**JBIG2 y JB2** . `JBIG2` (Joint Bi-level Image Experts Group 2) es un ejemplo de un método de propósito especial [7, 25]. Fue desarrollado específicamente para la compresión progresiva de imágenes binivel. El término compresión progresiva significa que la imagen se guarda en varias capas en la secuencia de datos comprimidos a medida que las resoluciones son más y más altas, tal como se muestra en la Figura 1.1. Cuando se descomprime y se visualiza una de las capas, se visualiza primero una imagen de poca resolución (la primera capa, ver Figura 1.1a), seguida por versiones mejoradas de la misma (capas posteriores, como en la Figura 1.1).

Una característica importante de la definición de JBIG2 es que el funcionamiento del codificador no se define en detalle [10, 3]. Da por hecho que cualquier codificador que genera un archivo JBIG es un codificador JBIG válido.

**DjVu** . Los métodos de compresión de imágenes se diseñan normalmente para un tipo específico de imagen. JBIG, por ejemplo, fue diseñado para imágenes binivel y es

(a) (b) (c) (d) (e)

Figura 1.1: Representación de las capas almacenadas para una compresión progresiva de una imagen.

ahí donde `DjVu` [4] hizo uso de las ventajas de diferentes compresores de imágenes para desarrollar un compresor que contuviera las riquezas de cada uno de estos.

Tomando el hecho de que, el texto, los dibujos y las imágenes contenidas en un documento, tienen diferentes características visualmente hablando, el texto requiere una alta resolución dejando de lado la importancia del color, pues no existe gran variación de color. Las Imágenes y fondos, por otro lado, pueden ser codificados a menor resolución, pero con más bits por píxel. Por consiguiente, DjVu comienza descomponiendo el documento en tres componentes: máscara (mask), primer plano (foreground), y fondo (background). El fondo y el primer plano son comprimidos con un método basado en wavelets, mientras que la máscara es comprimida con JB2, desarrollado por AT&T.

**Refined Fixed Double Pass Binary Object (RDPD)** . Al igual que JBIG2 y JB2, *RDPD* [17] es un compresor diseñado para imágenes binivel. La formación del diccionario con doble pasada consiste en emplear el algoritmo Pattern Matching & Substitution (PM&S) que hace una clasificación de todos los objetos utilizando la distancia de Tanimoto como primer paso, mientras que en el segundo paso clasifica los patrones elegidos previamente en la formación del diccionario. En cada uno de los pasos se realiza el refinamiento de las clases generadas eligiendo el mejor miembro representante como patrón.

## 1.2 Planteamiento del problema

El cambio a mayores velocidades de transferencia de información no es tarea fácil, básicamente por razones como la no factibilidad de realizar cambios de infraestructura. Para conseguir mayores prestaciones de velocidad se debe recurrir a técnicas que les permitan superar de alguna manera las deficiencias físicas de la red. La técnica más

importante en este sentido es la compresión de datos. La compresión de datos es beneficiosa en el sentido de que el proceso de compresión – transmisión – descompresión es más rápido que el proceso de transmisión sin compresión. La compresión de datos no sólo es para la transmisión de datos sino también para el almacenamiento masivo. De igual manera, la necesidad de almacenamiento crece por encima de las posibilidades del crecimiento de los discos duros o memoria.

El problema no es resuelto pues nos encontramos con uno nuevo: decidir la forma más conveniente de representar información utilizando sólo los símbolos que representarán tal información (es decir, los símbolos que se pueden emplear para su codificación). Pero, la información es diferente y variada, y en consecuencia no se puede codificar cualquier tipo de información con los mismos símbolos, o lo que es lo mismo, ningún método de compresión puede comprimir de manera eficiente todo tipo de dato, ésta es la razón por la que continuamente se están desarrollando nuevos métodos de compresión de propósito especial.

## 1.3 Objetivo

En la presente tesis se diseña un método de compresión que supere los actuales estándares de compresión de imágenes de documentos, el cual tendrá entre sus características principales el control sobre el nivel de pérdida de píxeles y en consecuencia sobre el nivel de compresión.

## 1.4 Motivación

La necesidad actual del ser humano de procesar una gran cantidad de información, en particular información digital, genera un problema debido a que esta incrementa con el día a día. Tal información debe ser almacenada y transmitida, por lo que desde el punto de vista computacional en algún punto podremos sobrepasar la capacidad de almacenamiento y/o de transmisión, lo que se traduce en la necesidad de establecer un mecanismos que permitan aprovechar de manera eficiente los recursos computacionales disponibles.

## 1.5 Estructura de la tesis

En los siguientes capítulos se expondrán temas que nos permitirán comprender de mejor manera la técnica propuesta en la presente tesis, así como los puntos medulares de la

misma.

En el Capítulo 2 se definen el término "objeto" y cómo trabajar con este para agruparlos de la forma más homogénea posible, además de detectar aquellos rasgos que lo caracteriza y así poderlo representar de la mejor manera posible.

En el Capítulo 3 se muestra la disposición y codificación de la información que nos servirá para la construcción del archivo comprimido, así como para reconstruir una aproximación del documento original.

Finalmente, en el Capítulo 4 se presentan algunos métodos del estado del arte contra los que se compara el método propuesto, así como los documentos de prueba o "benchmarks" para su comparación. También se presentan algunas medidas para la comparación entre estos métodos en los aspectos: nivel de compresión y pérdida. Por último, se realiza el análisis de los diferentes resultados y de las ventajas y desventajas del método propuesto respecto a otros.

# Capítulo 2

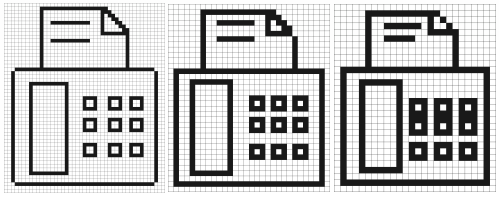
# Agrupamiento de objetos

Al tratar con imágenes nos encontramos con formas muy particulares según el tipo de documento pero que regularmente se repiten en él; en esta sección se mostrará la manera de manipular esas formas con la finalidad de aprovechar su repetitividad agrupándolas lo más homogéneamente posible con ayuda del análisis clúster.

## 2.1 Definiciones y conceptos

La formación de una imagen digital es el primer paso para cualquier procesamiento de imágenes digitales y consiste básicamente en el uso de algún tipo de sensor mediante el cual la imagen óptica se transforma en una *imagen raster* que permitirá su procesamiento digital.

Las imágenes raster están formadas por una rejilla de celdas donde a cada una de estas celdas, que se denominan pixeles, a los que se les asigna un valor de color. Por esto, cuando vemos todo el conjunto de celdas representadas en un monitor tenemos la ilusión de una imagen de tono continuo. Es importante hacer notar que el pixel es una unidad de información, no una unidad de medida, es decir, un pixel puede ser tan pequeño (`0.1 mm`) o tan grande (`1cm`) como sea necesario o posible (váse Figura 2.1).



(a) (b) (c)

Figura 2.1: Muestra de rejillas para la misma imagen a diferentes resoluciones.

(a) Binivel (b) Escala de grises (c) Tonos continuos

Figura 2.2: Ejemplos de imágenes a diferentes tipos de escalas.

El término "resolución" indica el número de pixeles por unidad de longitud de la imagen, por ejemplo puntos por pulgada (dpi, por sus siglas en inglés). Para el propósito de la compresión de imágenes es útil distinguir los principales tipos de imágenes [3]:

**Imagen binivel (o monocromática).** Ésta es una imagen donde los pixeles pueden tener uno de dos valores, normalmente referenciados como negro y blanco (1–pixeles y 0–pixeles, respectivamente). Cada pixel de dicha imagen se representa mediante un bit (ver Figura 2.2a).

**Imagen en escala de grises** . Un pixel en dicha imagen puede representarse con $n$ bits; admitiría $2^n$ posibles tonos de gris (o tonos de otro color) y normalmente los valores de los pixeles estarían comprendidos entre 0 y $2^n - 1$, un ejemplo de imagen en tonos de gris se muestra en la Figura 2.2b. El valor de $n$ es normalmente compatible con un tamaño de byte[1].

**Imagen de tonos continuos.** Este tipo de imágenes se distingue por contener una gran gama de colores, incluso tonos de grises; tan amplia puede ser esta gama de colores que es complicado distinguir entre ellos, en algunas ocasiones es más bien imposible si sus valores difieren poco. En consecuencia, este tipo de mallas producen zonas en la imagen que varían de manera uniforme (continua). Este tipo de imágenes pueden contener hasta tres componentes en el caso de imágenes a color como lo muestra la imagen de la Figura 2.2c, a diferencia de las imágenes binivel y en escala de grises que sólo emplean una componente.

[1] 4, 8, 12, 16, 24, o algún otro múltiplo conveniente

### 2.1.1 Conectividad

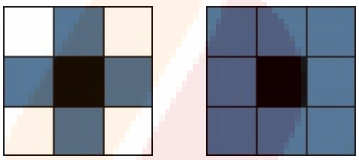
Un pixel $p$ de coordenadas $(x, y)$ tiene dos vecinos horizontales y dos vecinos verticales cuyas coordenadas están dadas por:

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1).$$

Este conjunto se le llama vecindad–4 de $p$ y se denota como $N_4(p)$ (véase Figura 2.3a). Similarmente, un pixel $p$ tiene cuatro vecinos en diagonal cuyas coordenadas están dadas por:

$$(x+1, y+1), (x-1, y+1), (x+1, y+1), (x-1, y-1).$$

Este conjunto se denota como $N_D(p)$. A este conjunto de puntos junto con los de la vecindad–4, se le llaman vecindad–8 de $p$ y se denotan como $N_8(p)$ (véase Figura 2.3b).



(a) Vecindad–4. (b) Vecindad–8.

Figura 2.3: Ejemplo de vecindades.

La conectividad entre pixeles [18] es un concepto utilizado para establecer los límites entre objetos y regiones de componentes en una imagen. Para establecer la conectividad entre dos pixeles, es necesario determinar si son adyacentes en sentido específico (si son vecinos) y si su nivel de gris satisface un criterio especificado de similitud (si son iguales). Por ejemplo, en una imagen binivel con valores 0 y 1, dos pixeles sólo se consideran conectados si comparten alguna de sus fronteras y además tienen el mismo valor.

**Definición 2.1.1.** A partir de estos conceptos, podemos definir dos tipos de conectividad:

1. Conectividad–4. Dos pixeles, $p$ y $q$, están conectados si $q$ pertenece a $N_4(p)$ (ver figura 2.4a).

2. Conectividad–8. Dos pixeles, $p$ y $q$, están conectados si $q$ pertenece a $N_8(p)$(ver figura 2.4b).

(a) Conectividad–4. (b) Conectividad–8.

Figura 2.4: Tipos de conectividad según su vecindad.

### 2.1.2 Etiquetado de componentes

Suponiendo un barrido en una imagen binivel de izquierda a derecha y de arriba a abajo, y asumiendo una conectividad–4, si $p$ denota el pixel en cualquier paso del proceso de barrido, con $t$ y $l$ los vecinos de arriba e izquierda, respectivamente, la naturaleza de la secuencia de barrido asegura que, cuando llegamos a $p$, los puntos $t$ y $l$ ya han sido etiquetados.

Si el valor de $p$ es 0, simplemente se mueve a la siguiente posición del barrido; si el valor es 1, se deben examinar $t$ y $l$; en caso de que ambos valgan 0, se asigna una etiqueta a $p$, mientras que si sólo uno de los dos vale 1, se asigna otra etiqueta a $p$. En caso de que ambos valgan 1 y tengan la misma etiqueta, se asigna esa etiqueta a $p$. Al finalizar el barrido, todos los puntos con valor de 1 han sido marcados.

Para conectividad–8, el procedimiento es similar, pero las dos vecindades diagonales superiores de $p$ se denotan como $s$ y $m$, también deben ser examinadas. La naturaleza del barrido asegura que estas vecindades han sido procesadas al llegar a $p$.

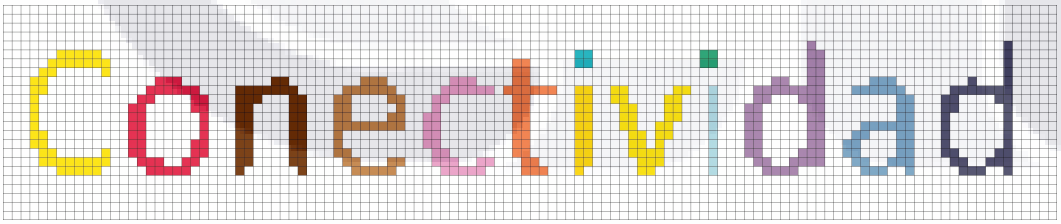



Figura 2.5: Ejemplo de imagen etiquetada, donde cada color representa una etiqueta y en consecuencia cada grupo de pixeles con el mismo color son objetos de la imagen.

**Definición 2.1.2.** Al subconjunto de pixeles con la misma etiqueta, se les llama componentes u objetos (ver Figura 2.5).

Una vez detectados los objetos contenidos, nos disponemos a agruparlos de manera que los integrantes de estos grupos sean lo más homogéneos, es decir, similares entre

ellos. Con este objetivo mostraremos una manera de hacerlo usando análisis clúster. Antes se debe introducir una forma para medir qué tanto un objeto se parece o difiere de otro.

## 2.2 Medidas de similaridad y disimilaridad

Existe un gran variedad de medidas de similaridad o disimilaridad, cada una tiene sus fortalezas y debilidades. Estas medidas se discuten en el contexto de dos problemas reales.

1. En el primer problema se da una secuencia observada y otras varias almacenadas y se requiere determinar la secuencia observada que mejor se adapta a la guardada.

2. El segundo problema consiste en ubicar sólo una porción de interés de la secuencia y buscarla dentro de la secuencia completa, en este caso es necesario encontrar la posición de mejor coincidencia de la plantilla dentro de la imagen observada, es decir, donde la similaridad se maximice.

Denotaremos la secuencia observada en el primer problema y la plantilla en el segundo problema por $O_2$ y denotaremos una secuencia guardada en el primer problema y una ventana dentro de la secuencia en el segundo problema por $O_1$. También supondremos que $O_1$ y $O_2$ contienen $n$ pixeles. Los dos problemas son similares en el sentido de que ambos requieren la determinación de la similitud entre dos secuencias o entre una plantilla y una ventana en una secuencia más grande.

Entre las medidas mayormente empleadas en lo que se refiere a imágenes binivel nos encontramos con las mencionadas a continuación.

**Distancia Euclidiana.** Esta distancia es la empleada en la geometría Euclidiana, que se define como la distancia directa entre dos puntos. La distancia Euclidiana entre dos objetos de imágenes binivel $O_1$ y $O_2$ está definida por:

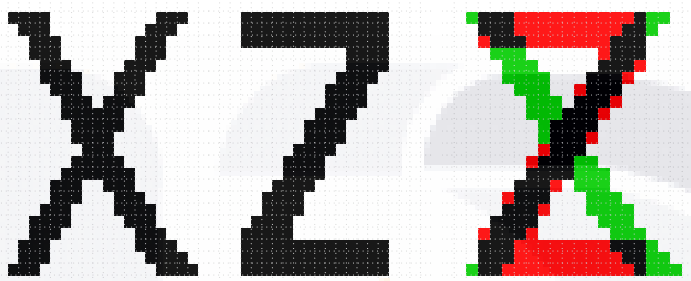
$$S_E(O_1, O_2) = \sqrt{n_1 + n_2 - 2n_{12}} \tag{2.2.1}$$

donde $n_1$ son los pixeles encendidos en el objeto $O_1$, $n_2$ los pixeles encendidos en el objeto $O_2$ y $n_{12}$ los pixeles encendidos en común, es decir, la raíz cuadrada del número de pixeles en los que que difieren ambos objetos.

**Distancia de Tanimoto.** La medida de Tanimoto [24] entre dos objetos de imágenes binivel $O_1$ y $O_2$ está definida por:

$$S_T(O_1, O_2) = \frac{n_1 + n_2 - 2n_{12}}{n_1 + n_2 - n_{12}}, \tag{2.2.2}$$

donde $n_1$, $n_2$ y $n_{12}$ son como en la distancia Euclidiana; dicho de otra forma, en



(a) Objetos binarios. (b) Empalme de los objetos 2.6a.

Figura 2.6: Muestra de empalme de dos objetos binarios. Donde los pixeles en negro son los pixeles encendidos en común, los pixeles en verde corresponden a los pixeles encendidos sólo en el primer objeto y los pixeles en rojo los pixeles encendidos sólo en el segundo objeto.Con $S_T = 0.6039$, $D_H = 122$ y $D_{HM} = 122$.

el numerador nos encontramos con el número pixeles que tienen en común ambos objetos (pixeles negros de la Figura 2.6b) y en el denominador con el número de pixeles en uno u otro objeto (pixeles negros, verdes y rojos de la Figura 2.6b). Este tiene el efecto de normalizar la medida con respecto a las escalas de $O_1$ y $O_2$, lo cual produce que esta medida se encuentre entre los valores 0 y 1,

**0** cuando sean el mismo objeto pixel a pixel; y

**1** cuando no tienen pixel alguno en común.

**Distancia de Hamming.** Dados los objetos $O_1$ y $O_2$, se define la distancia de Hamming [16] como el número de pixeles donde las imágenes $O_1$ y $O_2$ difieren, o dicho de otra manera, el mínimo de sustituciones requeridas para pasar de la imagen $O_1$ a la imagen $O_2$, definida como:

$$D_H(O_1, O_2) = n_1 + n_2 - 2n_{12} \tag{2.2.3}$$

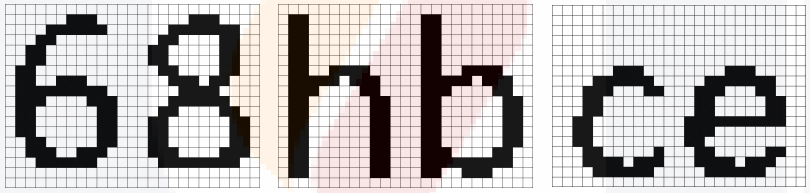
donde $n_1$, $n_2$ y $n_{12}$ son como en la distancia Euclidiana. A diferencia de la distancia de Tanimoto, la distancia de Hamming no estandariza su medida, en consecuencia

no está acotada superiormente sólo interiormente por 0, que es cuando los objetos $O_1$ y $O_2$ son el mismo.

Es importante mencionar que para nuestro método no se emplearon estas medidas debido a desventajas para nuestro propósito, pero que son mencionadas debido a que son de las medidas mayormente reconocidas cuando se habla de medidas de similitud de objetos. La principal desventaja es que en ocasiones al agrupar empleando alguna de estas distancias es posible agrupar objetos que evidentemente no deberían pertenecer al mismo grupo, ejemplos de ello son mostrados en la Figura 2.7. Para ello, debemos tener un control sobre la topología de los objetos, tema que se tocará a continuación.

## 2.3 Control de topología

Supongamos dos objetos con diferente número de hoyos (como los mostrados en la Figura 2.7), que aunque no son los mismos podemos llegar a agruparlos debido a que la distancia entre ellos, sea cual sea esta, es pequeña debido a su gran similitud.



(a) (b) (c)

Figura 2.7: Objetos binivel similares con diferente número de hoyos.

### Distancia de Hamming modificada

Para propósitos de esta tesis se diseñó una medida que logra no sólo medir la similaridad entre dos objetos sino que adicionalmente logra tener control sobre la topología de los objetos. La distancia de Hamming modificada entre dos objetos $O_1$ y $O_2$, como el nombre lo dice, es una variación de la medida de Hamming contabilizando el mínimo de sustituciones requeridas para pasar de la imagen $O_1$ a la imagen $O_2$ sólo en el caso de que la cantidad de hoyos $\gamma$ contenidos son el mismo, definida como:

$$D_{HM}(O_1, O_2) = \begin{cases} n_1 + n_2 - 2n_{12} & \text{si } \gamma(O_1) = \gamma(O_2) \\ \infty & \text{en otro caso} \end{cases} \tag{2.3.1}$$

Para calcular el número de hoyos $\gamma$ empleamos la generalización de la fórmula de Euler [11]:

$$V - A + C = 2 - 2\gamma, \tag{2.3.2}$$

o equivalentemente:

$$\gamma = \frac{2 - V + A - C}{2}, \tag{2.3.3}$$

donde $V$ es el número de vértices, $C$ el número de caras y $A$ el número de aristas.

El uso del número de hoyos $\gamma$ se debe a que queremos evitar que dos objetos con diferente número de hoyos caigan en el mismo clúster.

## 2.4 Análisis clúster

Es claro que en un documento de texto nos encontramos con objetos que si bien no son iguales sí son bastante parecidos, como consecuencia de ello nos es conveniente agruparlos como ya se mencionó en grupos homogéneos. Esto tendrá un impacto directo a la hora de su compresión pues sin duda alguna es mayormente conveniente almacenar un sólo objeto que represente de manera fiel a uno de estos grupos de objetos que a todos y cada uno de estos.

El problema de la clasificación puede ser complicado debido a varios factores, como la presencia de clases definidas de forma imperfecta, la existencia de categorías solapadas y posibles variaciones aleatorias en las observaciones. Una forma de tratar estos problemas, desde el punto de vista estadístico sería encontrar la probabilidad que tiene cada nueva observación de pertenecer a cada categoría existente. En este sentido, el criterio de clasificación más simple sería elegir la categoría más probable, mientras que pueden necesitarse reglas más sofisticadas si las categorías no son igualmente probables o si los costos de mala clasificación varían entre las categorías.

Análisis clúster es el nombre genérico de una amplia gama de algoritmos usados para clasificar elementos de estudio. Más concretamente, un método clúster es un procedimiento estadístico que comienza con un conjunto de datos que contienen información sobre una muestra de entidades e intenta reorganizarlas en grupos lo más homogéneamente posible, a estos se les llaman clústers.

El punto de partida para el análisis clúster es, en general, una matriz $D = (d_{ij})$ que proporciona la información para cada par de objetos, en nuestro caso esta información es la similaridad (disimilaridad) entre la imagen $O_i$ y la imagen $O_j$. La $i$-ésima fila de

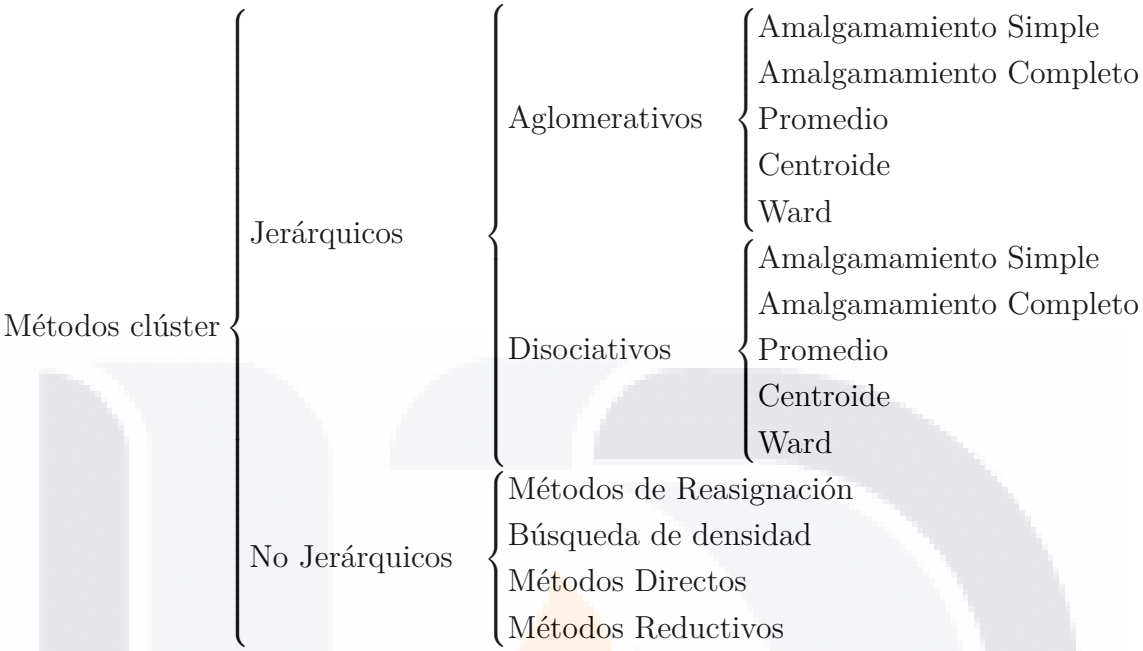



Figura 2.8: Mapa conceptual de los principales métodos clúster.

$D$ es el vector de distancias del objeto $O_i$ contra el resto de los objetos, y la $j$-ésima columna es el vector de distancias del objeto $O_j$ contra el resto de los objetos.

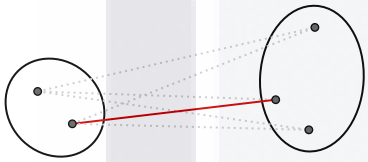
A grandes rasgos se distinguen dos categorías de métodos clústers: métodos jerárquicos y métodos no jerárquicos. En la Figura 2.8 se presentan algunos de los métodos clúster más importantes. Los métodos no jerárquicos [27], también conocidos como partitivos, tienen por objetivo realizar una sola partición de los individuos en $K$ grupos. Ello implica que se tiene que especificar de antemano la cantidad de grupos que deben ser formados.

Los métodos jerárquicos tienen por objetivo agrupar clústers para formar uno nuevo o bien separar alguno ya existente para dar origen a otros dos, de tal forma que se minimice alguna medida de disimilitud o bien una medida de similitud.

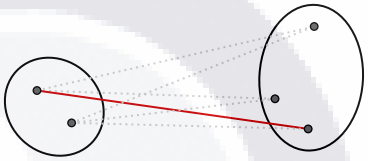
Los métodos jerárquicos se subdividen a su vez en aglomerativos y disociativos. Los aglomerativos comienzan el análisis con tantos grupos como individuos haya en el estudio. A partir de ahí se van formando grupos de forma ascendente, hasta que, al final del proceso, todos los casos están englobados en un mismo clúster. Los métodos disociativos realizan el proceso inverso al anterior: empiezan con un clúster que engloba a todos los individuos y a partir de este grupo inicial se van formando grupos cada vez más pequeños a través de sucesivas divisiones. Al final del proceso se tienen tantos

grupos como individuos en la muestra estudiada.

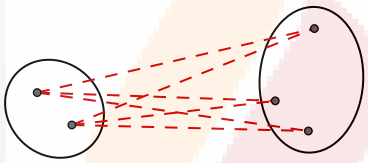
A continuación, vamos a presentar algunas de las estrategias que pueden ser empleadas a la hora de unir los clústers en las diversas etapas de un agrupamiento jerárquico. Ninguno de estos procedimientos proporciona una solución óptima para todos los problemas que se pueden plantear. El conocimiento del problema planteado y la experiencia, sugerirán el método más adecuado. De todas formas es conveniente usar varios procedimientos con la idea de contrastar los resultados obtenidos y sacar conclusiones, tanto como si hubiera coincidencias en los resultados obtenidos con métodos distintos como si no las hubiera.



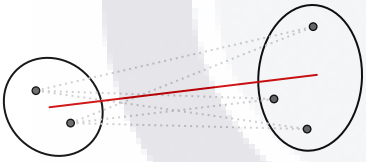
(a) Amalgamamiento Simple.



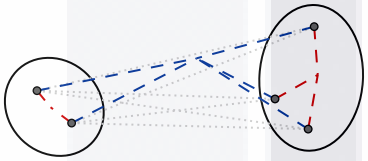
(b) Amalgamamiento Completo.



(c) Promedio.



(d) Centroide.



(e) Ward.

Figura 2.9: Representación gráfica de diferentes métodos de clusterización jerárquica.

**Amalgamamiento Simple (Single Linkage).** En este método se considera que la similitud (disimilitud) entre dos clústers viene dada, por la máxima similitud (o mínima disimilitud) entre sus componentes.

Así, si tras efectuar la etapa $k$-ésima tenemos ya formados $n-k$ clústers, la distancia entre los clústers $C_i$ (con $n_i$ elementos) y $C_j$ (con $n_j$ elementos) es:

$$s(C_i, C_j) = \max_{\substack{O_l \in C_i \\ O_m \in C_j}} \{s(O_l, O_m)\} \text{ o } d(C_i, C_j) = \min_{\substack{O_l \in C_i \\ O_m \in C_j}} \{d(O_l, O_m)\}. \qquad (2.4.1)$$

En la Figura 2.9a se muestra la interpretación gráfica del Amalgamamiento Simple.

**Amalgamamiento Completo (Complete Linkage)** . En este método se considera que la similitud (disimilitud) entre dos clústers se mide considerando a sus elementos más dispares, es decir, la similitud (disimilitud) entre clústers viene dada por la mínima similitud (máxima disimilitud) entre sus componentes.

Así, si tras efectuar la etapa $k$-ésima tenemos ya formados $n-k$ clústers, la distancia entre los clústers $C_i$ (con $n_i$ elementos) y $C_j$ (con $n_j$ elementos) es:

$$s(C_i, C_j) = \min_{\substack{O_l \in C_i \\ O_m \in C_j}} \{s(O_l, O_m)\} \text{ o } d(C_i, C_j) = \max_{\substack{O_l \in C_i \\ O_m \in C_j}} \{d(O_l, O_m)\}. \tag{2.4.2}$$

En la Figura 2.9b se muestra la interpretación gráfica del Amalgamamiento Completo.

**Promedio (Average).** Se considera como criterio de unión de dos clústers el promedio ponderado entre las similitudes (disimilitudes) de los componentes del clúster respecto a los del otro.

Si tomamos dos clústers $C_i$ y $C_j$, donde el clúster $C_i$ está formado, a su vez por otros dos clústers, $C_{i1}$ y $C_{i2}$ (con $n_{i1}$ y $n_{i2}$ elementos respectivamente, en consecuencia $n_i = n_{i1} + n_{i2}$ el número de elementos de $C_i$, y $n_j$ el número de elementos de $C_j$) . En términos de similitudes (disimilitudes), el promedio es:

$$\begin{aligned} s(C_i, C_j) =& \frac{1}{(n_{i1} + n_{i2})n_j} \sum_{l=1}^{n_{i1}+n_{i2}} \sum_{m=1}^{n_j} s(O_m, O_l) \\ =& \frac{n_{i1} s(C_{i1}, C_j) + n_{i2} s(C_{i2}, C_j)}{n_{i_1} + n_{i2}}. \end{aligned} \tag{2.4.3}$$

Es fácil mostrar la igualdad anterior (ver anexo A), mostrando así que la distancia $s(C_i, C_j)$ es el promedio ponderado de las distancias de cada uno de los dos clústers previos, $C_{i1}$ y $C_{i2}$ , con respecto al clúster $C_j$. En la Figura 2.9c se muestra la interpretación gráfica de la distancia promedio.

**Centroide.** En este método la semejanza entre dos clústers viene dada por la semejanza entre sus centroides, esto es, los vectores de medias de las variables medidas sobre los individuos del clúster.

Para medir la distancia entre los clústers $C_j$ (con $n_j$ elementos) y $C_i$ (formado a su vez por dos clústers $C_{i1}$ y $C_{i2}$, con $n_{i1}$ y $n_{i2}$ elementos, respectivamente). Sean $m_j$ , $m_{i1}$ y $m_{i2}$ los centroides de los clústers anteriormente mencionados. Así, el centroide del clúster $C_i$ vendrá dado por $m_i$, cuyas componentes son:

$$m_l^i = \frac{n_{i1}m_l^{i1} + n_{i2}m_l^{i2}}{n_{i1} + n_{i2}}. \tag{2.4.4}$$

Con ello, la distancia euclidiana cuadrática entre los clústers $C_i$ y $C_j$ vendrá dada por:

$$\begin{aligned} d_2^2(C_i, C_j) =& \sum_{l=1}^{n}(m_l^j - m_l^i)^2 \\ =& \frac{n_{i1}}{n_{i1} + n_{i1}} d_2^2(C_{i1}, Cj) + \frac{n_{i2}}{n_{i1} + n_{i1}} d_2^2(C_{i2}, Cj) - \frac{n_{i1}n_{i2}}{(n_{i1} + n_{i1})^2} d_2^2(C_{i1}, C_{i2}). \end{aligned} \tag{2.4.5}$$

Es fácil mostrar la igualdad anterior (ver anexo A). En la Figura 2.9d se muestra la interpretación gráfica de la distancia centroide.

**Ward.** El objetivo del método de Ward [26] es encontrar en cada etapa aquellos dos clústers cuya unión proporcione el menor incremento en la suma total de errores,

$$E = \sum_{k=1}^{K} E_k. \tag{2.4.6}$$

Si suponemos que existen $K$ clústers, se define $E_k$ como la distancia euclidiana al cuadrado entre cada individuo del clúster $k$ a su centroide:

$$E_k = \sum_{i=1}^{N_k}\sum_{j=1}^{N}(x_{ij}^k - m_j^k)^2. \tag{2.4.7}$$

Así cuando cada clúster está compuesto por un sólo individuo, cada individuo coincide con el centro del clúster y por lo tanto en este primer paso se tendrá $E_k = 0$ para cada clúster y con ello, $E = 0$.

Supongamos ahora que los clústers $C_i$ y $C_j$ se unen resultando un nuevo clúster

$C_u$. Entonces el incremento de $E$ es:

$$\begin{aligned}\Delta E_{ij} =& E_u - E_i - E_j \\ =& \frac{n_i n_j}{n_u} \sum_{l=1}^{n} (m_l^i - m_l^j)^2.\end{aligned} \tag{2.4.8}$$

Encontrando a cada etapa aquellos dos clústers cuya unión proporcione el menor incremento en la suma total de errores. En la Figura 2.9e se muestra la interpretación gráfica de la distancia Ward.

Una vez que se han agrupado los objetos según su similaridad o disimilaridad se procede a encontrar o generar un objeto que los represente de la mejor manera. A continuación se presentará un método que nos permitirá generar tal objeto usando el principio de mínima entropía.

## 2.5 Generación de representante

Un paso primordial en los métodos de compresión es la selección de un representante de clúster al cual debemos otorgar la propiedad de reducir la distancia que existe entre él y los integrantes del clúster, esto para reducir la pérdida de pixeles al momento de reconstruir la imagen original sustituyendo los objetos que originalmente se encontraban en ella por el representante mencionado. En este trabajo se hará uso del concepto de mínima entropía propuesto por Shannon [21] para su generación.

### 2.5.1 Entropía

En un sentido amplio, la entropía se interpreta como la medida del desorden de un sistema, lo cual le asocia un grado de homogeneidad. En física esto se aplica a la segunda ley de la termodinámica, la cual dice que los sistemas aislados tienden al desorden, es la magnitud que permite medir la parte no utilizable de la energía contenida en un sistema. Esto quiere decir que dicha parte de la energía no puede usarse para producir un trabajo. La entropía para la formación de un compuesto químico se establece midiendo la que conforma a cada uno de sus elementos constituyentes. A mayor entropía de formación más favorable será su formación.

Mientras que en la teoría de la comunicación el concepto de entropía es empleado como medida del grado de incertidumbre que posee un mensaje, es decir, la confianza que se tiene en que el mensaje transmitido es correcto.

**Definición 2.5.1.** Sea $C$ una variable aleatoria discreta de rango finito que no incluya valores de probabilidad nula, y sea $p(c) = P(C = c)$ su función de probabilidad. Definimos la entropía de Shannon de la variable aleatoria $X$ como:

$$H(C) = -\sum_{i=1}^{n} p(c_i) \log p(c_i). \tag{2.5.1}$$

Consideremos a $C$ un grupo de objetos binivel generado después de una clusterización, el método propuesto consta en un principio de dos pasos fundamentales: el primero de ellos es elegir de entre los objetos contenidos en $C$ el que en promedio tenga la menor distancia y tomarlo como base para nuestro segundo paso, que es empalmarlos uno a uno formando un histograma de tal suerte que la entropía del histograma sea la menor.

Para una mejor apreciación un ejemplo de $C$ es el formado por los objetos mostrados en la Figura 2.10 que son objetos similares entre si, además en la Tabla 2.1 se muestran las distancias de Hamming modificada que existen entre cada uno de estos objetos; en ella podemos observar que el objeto 2.10c tiene la menor distancia promedio, por lo tanto será el primer objeto del histograma.

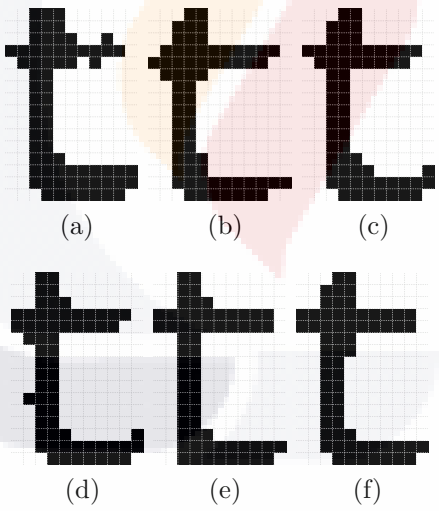



Figura 2.10: Objetos contenidos en un clúster para la generación de un representante empleando el principio de mínima entropía.

A continuación, se toman el siguiente objeto (según su aparición, en nuestro ejemplo el objeto de la Figura 2.10a) y se realiza el empalme con el objeto obtenido en el primer paso. Para realizar el empalme, tomamos como referencia inicial los centroides de ambos objetos y calculamos la entropía de este histograma resultante para después repetir el proceso variando el centroide, sólo del objeto a empalmar, sobre los ocho vecinos del

centroide de éste, para quedarnos finalmente con el empalme que haya arrojado la menor entropía (veáse Figura 2.11a). Reiterando el proceso con el resto de los objetos (objetos de las Figuras 2.10b, 2.10d, 2.10e, 2.10f) hasta objetener finalmente el histograma de la Figura 2.11j.

| | | Objeto (a) | (b) | (c) | (d) | (e) | (f) | Promedio |
|---|---|---|---|---|---|---|---|---|
| O | (a) | 0 | 18 | 9 | 15 | 16 | 16 | 12.33 |
| b | (b) | 18 | 0 | 9 | 11 | 10 | 6 | 9.00 |
| j | (c) | 9 | 9 | 0 | 8 | 9 | 7 | 7.00 |
| e | (d) | 15 | 11 | 8 | 0 | 5 | 9 | 8.00 |
| t | (e) | 16 | 10 | 9 | 5 | 0 | 6 | 7.67 |
| o | (f) | 16 | 6 | 7 | 9 | 6 | 0 | 7.33 |

Tabla 2.1: Distancia de Hamming modificada entre los objetos mostrados en la Figura 2.10.

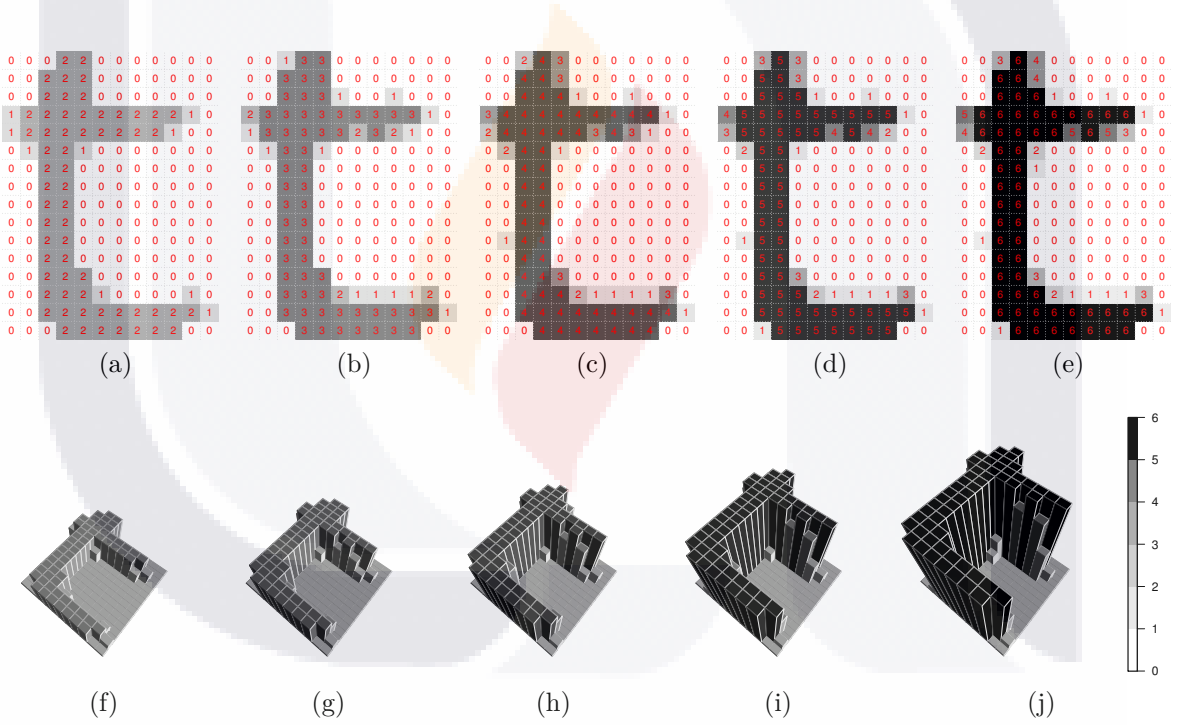
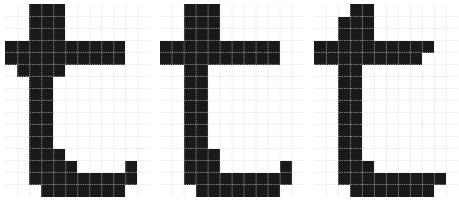



Figura 2.11: Proceso de empalme de los objetos en la Figura 2.10 usando el criterio de entropía mínima. En la parte inferior de la figura vemos el histograma visto de forma tridimensional y la parte superior el mismo histograma pero desde una vista superior con la altura de cada columna en color rojo.

Recordemos que las imágenes con las que trabajamos son binivel con valores no mayores a 1, pero nuestro histograma puede alcanzar valores mayores. Para solucionar este inconveniente, basta con dividir el histograma final por el número de objetos y



(a) $h = 1/3$ (b) $h = 1/2$ (c) $h = 2/3$

Figura 2.12: Efecto de diferentes umbrales $h$ para la elección del representante.

| | | Objeto | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | (a) | (b) | (c) | (d) | (e) | (f) | R | Promedio |
| O | (a) | 0 | 18 | 9 | 15 | 16 | 16 | 11 | 12.14 |
| b | (b) | 18 | 0 | 9 | 11 | 10 | 6 | 7 | 8.71 |
| j | (c) | 9 | 9 | 0 | 8 | 9 | 7 | 4 | 6.57 |
| e | (d) | 15 | 11 | 8 | 0 | 5 | 9 | 6 | 7.71 |
| t | (e) | 16 | 10 | 9 | 5 | 0 | 6 | 5 | 7.28 |
| o | (f) | 16 | 6 | 7 | 9 | 6 | 0 | 5 | 7.00 |
| | R | 11 | 7 | 4 | 6 | 5 | 5 | 0 | 5.43 |

Tabla 2.2: Distancia de Hamming modificada entre los objetos mostrados en la Figura 2.10 (primeras 6 columnas y 6 filas) más el representante obtenido con un umbral $h = 0.5$ mostrado en la Figura 2.12b (última fila y columna).

tomar sólo aquellos pixeles que superen un umbral $0 \leq h \leq 1$. En la Figura 2.12 se muestra el resultado de emplear diferentes umbrales sobre el histograma generado a partir del clúster de nuestro ejemplo.

Calculando de nueva cuenta la Tabla 2.1 pero ahora tomando como séptimo objeto el representante $R$ generado con un umbral de 0.5 se obtiene lo mostrado en la Tabla 2.2, que arroja una mejora disminuyendo la distancia incluso la del objeto 2.10c que fue el que mejor resultado obtuvo respecto al resto en un principio.

La elección del umbral afecta considerablemente tanto el nivel de compresión como el nivel de pérdida de pixeles. Lo anterior debido a que entre menor sea el umbral menor será la información para almacenar por cada objeto pero mayor la pérdida de pixeles y lo contrario para un umbral mayor, por ello se decidió tomar $h = 0.5$, con la finalidad de obtener aproximadamente el mismo número de pixeles cambiados (de 0 a 1, como de 1 a 0). Dando como resultado lo ya mencionado, la disminución de las distancias.

Así pues, en este capítulo se introdujo la metodología empleada para la creación del representante a partir de un grupo de objetos binivel, previamente generado empleando un análisis clúster controlando no agrupar en el mismo, objetos con diferente topología con ayuda de la distancia de Hamming modificada diseñada para este propósito.

# Capítulo 3

# Estructura del compresor

El siguiente paso en el proceso para la compresión de una imagen binivel es identificar aquella información que me va a permitir recuperar, de la manera más fiel, la imagen original y la forma más conveniente de almacenar dicha información. Una vez definidos los grupos de objetos similares y creados los representantes, es necesario crear el archivo comprimido en base a ellos. Recuérdese que la idea es almacenar solamente el representante y usarlo para sustituir a cada objeto que representa en el documento; por ello es necesario: identificar la posición de cada objeto en el documento o referenciarlo de alguna manera con respecto al representante que lo va a sustituir.

Otro punto importante es que necesitamos almacenar la información de cada representante, y esto lo debemos hacer de manera que ocupe la menor cantidad de bytes posibles. A continuación se detalla la forma en que serán almacenados los objetos binivel representantes de cada clúster así como la estructura del archivo comprimido final.

## 3.1 Ubicación en documento

Particularmente para nuestro propósito, en el caso de imágenes binivel donde están bien diferenciados los objetos (1–pixeles) del fondo (0–pixeles), definiremos las variantes que existen para identificar las coordenadas del píxel referencia de la posición del objeto que servirá a la hora de descomprimir y querer reconstruir la imagen binivel.

**Ubicación matricial.** Consideremos una imagen $I$ de tamaño $n \times m$ pixeles, donde $n$ es el número de filas de la imagen y $m$ el número de columnas de la imagen; hacemos referencia a un píxel en $I$ con la dupla $(i, j)$, donde $i$ es la fila y $j$ es la columna en la que se encuentra, comenzando el conteo de las filas de arriba a abajo y el conteo de las columnas de izquierda a derecha. Es claro de esta manera que el valor de las filas se encuentra entre 1 y $n$, y el valor de las columnas entre

1 y $m$.

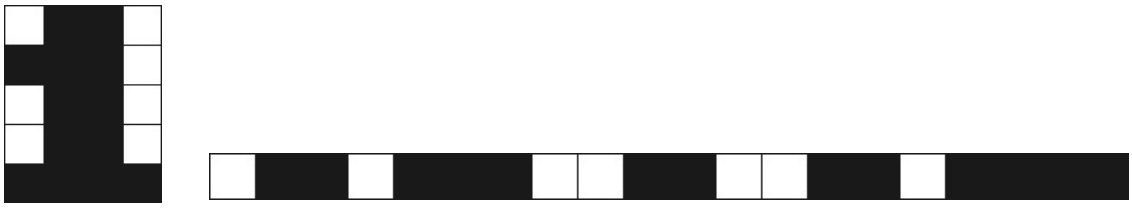
Para referenciar una componente conectada (como se mostró en la sección 2.1), se crea una submatriz que la contiene, de manera que dicha submatriz sea la menor posible. Así, usamos el píxel en la esquina superior izquierda de dicha submatriz para referencia a la componente. Por ejemplo, en la Figura 3.1 nos encontramos con cuatro componentes con forma de 'a', 'b','c' y 'd', respectivamente, hacemos referencia a su ubicación tomando el píxel superior izquierdo de la submatriz que contiene a cada componente. En la Figura 3.1 se marca con rojo a los pixeles que determinan sus ubicaciones originales en la imagen $I$. Esta ubicación matricial

Figura 3.1: Imagen binivel que presenta la ubicación de los objetos contenidos en ella. El píxel rojo representa tal ubicación para cada objeto contenido en la matriz de $40 \times 64$ pixeles. Con ubicación matricial $(3,3)$,$(3,38)$,$(4,19)$ y $(13,45)$; y ubicación vectorial 131,166,211 y 813 en orden de aparición si recorremos la imagen de izquierda a derecha y de arriba a abajo.

necesita de almacenar dos enteros para cada componente y, dado que el propósito final es comprimir el archivo de la imagen, sería conveniente reducir esa cantidad. La siguiente forma de ubicación es una opción para lograr esto.

**Ubicación vectorial.** Si "desdobláramos" la imagen $I$ (tal como se muestra en la Figura 3.2) de tal forma que nos quedara un vector de tamaño $nm$ , en lugar de trabajar con dos números $i$ y $j$ que nos indiquen la posición del objeto, tendríamos únicamente un número. Que no es más que el valor

$$l = (i-1) * m + j. \tag{3.1.1}$$



(a) (b)

Figura 3.2: En el lado izquierdo se muestra un objeto binivel en su forma matricial de tamaño $5 \times 4$ y del lado derecho el mismo objeto desdoblado de longitud 20.

donde $i$ y $j$ es la posición matricial del píxel. De la misma forma, si queremos convertir de la forma vectorial a la matricial a partir de $l$, hacemos

$$i = \text{mod}(l, m) + 1 \text{ y } j = \text{res}(l, m), \tag{3.1.2}$$

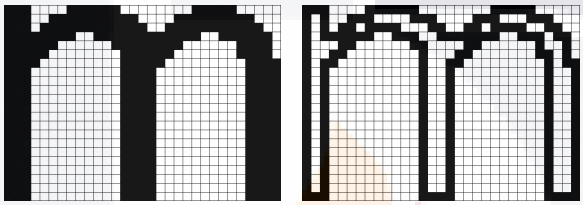
donde mod es la función módulo y res la función residuo, que nos arrojan la división y el resto en una división entera. Notemos que un valor indispensable, es el ancho $m$ de la imagen.

Notemos que la ubicación vectorial toma valores entre 1 y $mn$. Esta última cantidad puede volverse muy grande incluso para imágenes pequeñas. Por ejemplo, si nuestra imagen es de $100 \times 100$, $mn = 10000$. Por ello, introduciremos una alternativa a ello, la ubicación relativa, que es de gran ayuda cuando debemos referenciar más de un objeto.

**Ubicación relativa.** Como podemos ver, para propósitos de almacenar información, ambas formas de ubicar un objeto tienen desventajas: en el caso de la forma matricial la necesidad de almacenar dos números y la forma vectorial la necesidad de almacenar valores grandes. La ubicación relativa consiste en tomar la ubicación vectorial de un objeto a partir de la ubicación del anterior objeto detectado. En el caso del primer objeto detectado se toma la ubicación vectorial tal cual. En el ejemplo de la Figura 3.1, el primer objeto detectado es la letra 'a' con ubicación $l_a = 131$. A continuación la referencia del segundo objeto 'c' ya no sería 166 sino $l_c = 166 - l_a = 35$; para el tercer objeto 'b' sería $l_b = 211 - l_a - l_c = 45$ y así sucesivamente. Es importante mencionar que tales referencias jamás serán negativas por la forma en que se recorre la imagen (de izquierda a derecha y de arriba a abajo) para detectar las componentes.

## 3.2 Codificación de componentes

El contorno de un objeto contenido en una imagen es de gran importancia, pues preserva la información estructural de los límites del objeto. Debido a que almacenar la submatriz que contiene a cada componente necesita de una gran cantidad de bits, almacenaremos únicamente su contorno debido a que lo que se encuentra fuera de este son 0–pixeles y en el interior tenemos únicamente 1–pixeles. En la Figura 3.3 podemos observar tanto el objeto que se quiere almacenar (lado izquierdo) como su contorno que será almacenado (lado derecho).



(a) (b)

Figura 3.3: En el lado derecho observamos el contorno de la Figura 3.3a al lado izquierdo.

### 3.2.1 Extracción de contorno

El contorno de un objeto en una imagen se define como la variación entre pixeles vecinos de la imagen. Por tanto, un detector de contornos se puede formar a través de técnicas de diferenciación de imagen. En al caso de imágenes binivel, un píxel $p$ se define como contorno si este vale 1 y alguno de sus vecinos[1] es 0.

Para representar el contorno usaremos código de cadena, que son estructuras de datos que permiten la representación de un contorno mediante símbolos que representan uno o varios movimientos referentes al recorrido de los pixeles que conforman dicho contorno. Se ha implementado el código 3OT (Three Ortogonal Symbol chain code), el cual representa los contornos en una imagen binaria, con o sin hoyos, con tres símbolos. La razón de la elección de este código de cadena y no de otros como F4, F8, VCC, entre otros; es por el hecho de que se ha mostrado que es de las mejores opciones a la hora de comprimir este código de cadena [12].

[1]Según el tipo de conectividad

### 3.2.2 Código de cadena 3OT

Supongamos que hemos detectado el contorno de un objeto en una imagen binaria. El proceso de codificación 3OT [19] comienza detectando en la imagen binaria el primer 1–pixel $(x_1, y_1)$, siendo el escaneo fila por fila de izquierda a derecha. Una vez detectado ese píxel inicial se comienza a recorrer únicamente el borde de los pixeles de la imagen en sentido de las manecillas del reloj, esto teniendo en cuenta una vecindad 4-adyacente, es decir, los únicos movimientos que podemos realizar a partir de la esquina de un 1–pixel son: izquierda $(-\hat{i})$, derecha$(\hat{i})$, arriba $(-\hat{j})$ y abajo $(\hat{j})$; con $\hat{i}$ y $\hat{j}$ los vectores canónicos unitarios (ver Figura 3.4).

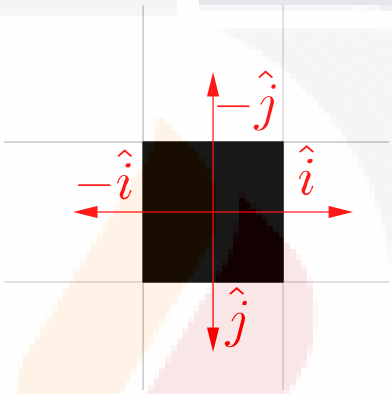



Figura 3.4: Movimientos válidos desde un pixel a otro para el recorrido del borde de un objeto binario, donde izquierda $(-\hat{i})$, derecha$(\hat{i})$, arriba $(-\hat{j})$ y abajo $(\hat{j})$ ; con $\hat{i}$ y $\hat{j}$ los vectores canónicos unitarios.

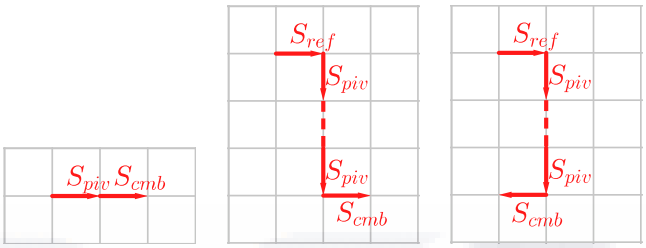
Para poder codificar el borde nos auxiliaremos de un vector de referencia $S_{ref}$, un vector pivote $S_{piv}$ y un vector de cambio $S_{cmb}$. El vector $S_{ref}$ se puede ver como el último cambio de dirección detectado, el vector $S_{piv}$ es la dirección actual, y por último el vector $S_{cmb}$ que representa la siguiente dirección.

Se comentó que éste método consiste de tres símbolos, que son 0, 1 y 2. Tales códigos son generados con ayuda de los vectores $S_{ref}$, $S_{piv}$ y $S_{cmb}$ y su interacción. Los cuales son generados de la siguiente manera (ver Figura 3.5):

**0:** No se ha detectado cambio de dirección del vector cambio $S_{cmb}$ respecto al vector pivote $S_{piv}$, (Figura 3.5a).

**1:** Si por el contrario se registra un cambio de dirección en el último paso que hemos dado, es decir, $S_{piv} \neq S_{cmb}$ (Figura 3.5b), pero sin embargo se mantiene la dirección del vector de referencia $S_{ref}$.

**2:** Además de darse un cambio de dirección en el último paso y su dirección es contraria a la del vector de referencia $S_{ref}$ (Figura 3.5c).

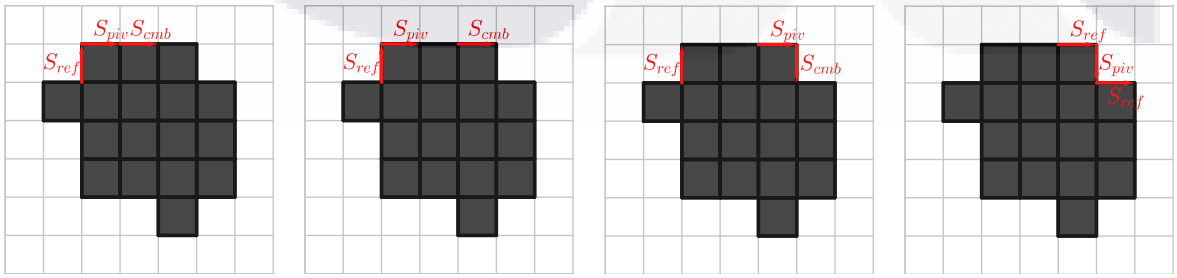



(a) Generación de un código "0". (b) Generación de un código "1". (c) Generación de un código "2".

Figura 3.5: Cambios de dirección para la codificación 3OT. El único código que no depende del vector de referencia $S_{ref}$ es el "0".

De esta manera se irán generando cada uno de los códigos conforme se recorre el borde del objeto. Sin embargo, una vez generado el código representante de uno de los bordes del pixel es necesario actualizar los vectores referencia $S_{ref}$ y $S_{piv}$. Así, cada vez que se genere un símbolo '1' o '2' en nuestro código, el vector pivote se convertirá en el vector referencia ($S_{ref} \leftarrow S_{piv}$) y el vector cambio será ahora nuestro vector pivote ($S_{piv} \leftarrow S_{cmb}$); en cambio si el símbolo es "0" los vectores quedan intactos.

Es importante tener en cuenta que al comenzar en el borde del pixel $(x_1, y_1)$, no contemplamos los vectores de referencia $S_{ref}$ y pivote $S_{piv}$ debido a que no se ha comenzado el recorrido. A pesar de ello, los podemos conocer desde un principio ya que al ser el primer pixel, según el orden de línea mayor, el vector referencia tiene una dirección hacia arriba ($S_{ref} = -\hat{j}$) y el vector pivote hacia la derecha ($S_{piv} = \hat{i}$), tal como se muestra en la Figura 3.6a.




(a) Primer paso. Código generado "0". (b) Segundo paso. Código generado "0". (c) Tercer paso. Código generado "2". (d) Cuarto paso. Código generado "1".

Figura 3.6: Ejemplo de los primeros pasos para la generación del código de cadena.

### Ubicación de hoyos

Como hemos observado en varias imágenes, por ejemplo en el caso de la Figura 3.1, los objetos pueden contener hoyos que corresponden a 0–pixeles rodeados totalmente de 1–pixeles. Al igual que hemos visto cómo detectamos objetos en una imagen, de la misma manera detectamos estos hoyos en un objeto. La idea básicamente es la misma, pues para obtener la ubicación del hoyo usamos a la submatriz que contiene al objeto (en lugar de a toda la imagen) como muestra la Figura 3.7, y se obtiene la referencia del hoyo como si se tratase de un objeto.

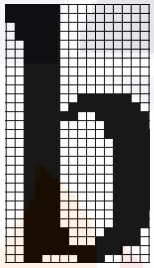


Figura 3.7: Objeto extraído de la Figura 3.1, con referencia matricial $(13, 7)$ y referencia vectorial 99.

## 3.3 Compresión a nivel bit

Hasta el momento se ha generado un archivo de texto con la información necesaria para la construcción de una aproximación a la imagen original, pero es necesario mencionar que el tamaño de este archivo de texto puede ser incluso mayor que el de la imagen original, pues lo que se hizo hasta el momento fue convertir el archivo original en otro con mayor redundancia y así poder aplicar una compresión a nivel bit. PAQ [13] es una familia de algoritmos de compresión estrechamente relacionados con el algoritmo Predicción por Coincidencia Parcial (PPM, por sus siglas en inglés) [5]. Los métodos de compresión de datos basados en PPM originalmente fueron una referencia en términos de compresión en la década de los 90, una gran cantidad de modelos que predicen de forma independiente el siguiente bit. Finalmente, estas predicciones se combinan usando una red neuronal y un algoritmo aritmético como el PPM.

El primer método de mezcla contextual que PAQ utilizó fue un modelo de mezcla de contexto ponderado. Donde cada modelo $m \in \{1, \ldots, n\}$, $n \in \mathbb{N}$, $n$ el número de modelos, tiene un peso fijo $w_m \in \mathbb{N}$ y las probabilidades se expresan como un par de números naturales. Debido a que lo que importa es la razón entre los pesos, la

normalización de los pesos (o las probabilidades) no es necesaria. Dada la distribución de probabilidad por el modelo $m$, sea $P_m = (P_m(0), P_m(1))$, entonces el símbolo 0 tiene una probabilidad

$$p_m(0) = \frac{P_m(0)}{P_m(0) + P_m(1)}, \tag{3.3.1}$$

y el símbolo 1

$$p_m(1) = \frac{P_m(1)}{P_m(0) + P_m(1)}. \tag{3.3.2}$$

En consecuencia la probabilidad final $P_*$ es la suma ponderada

$$P_* = \sum_{m=1}^{n} P_m w_m = (P_*(0), P_*(1)) \tag{3.3.3}$$

Generalmente, si el contexto tiene una longitud $k$, el peso se establece como $k^2$. Pues intuitivamente se puede pensar que entre mayor sea el contexto mayor conocimiento de los modelos se tiene, y en consecuencia los modelos que usan tales contextos podrían predecir con mayor precisión, lamentablemente esto no siempre es cierto.

Esto fue resuelto empleando un modelo de mezcla de contexto ponderado adaptativo. En este modelo, los pesos cambian en cada paso. La idea principal es suponer que la predicción ponderada final $P_*$ será más precisa que la mayoría de sus predicciones individuales $P_m$, $m \in \{1, \ldots, n\}$. Por lo tanto, cuanto mayor es el error de predicción y mayor es la desviación entre $P_*$ y $P_m$, más ajustes se deben hacer a $w_m$. Tal ajuste se calcula como

$$w_m = w_m + ed \tag{3.3.4}$$

donde $e = 1 - P_*(1)$ y $d = P_*(0)P_m(1) - P_*(1)P_m(0)$, donde debido a la forma de la ecuación se sugiere el uso de una red neuronal [6].

A modo de resumen, el método propuesto *EntD* consiste en agrupar cada uno de los objetos en $n$ grupos empleando un análisis clúster con ayuda de la medida de Hamming modificada, la cual nos permite controlar la topología de los objetos y así evitar juntar en un mismo clúster objetos con diferente número de hoyos. Posteriormente se genera un representante por clúster empleando el criterio de mínima entropía el cual es codificado empleando el código de cadena 3OT. Finalmente, son almacenados en un archivo de texto los representantes ya codificados junto con la posición de los hoyos, así como el representante de cada objeto y su posición relativa en el documento, para finalmente hacer una compresión a nivel bit de este archivo de texto con ayuda del compresor PAQ.

# Capítulo 4

# Resultados y Análisis

En este capítulo mostraremos los resultados obtenidos con el compresor de imágenes con pérdida de información propuesto para su comparación con algunos compresores del estado del arte. También se definen medidas de calidad de la compresión para tal comparación.

## 4.1 Medidas de calidad de compresión

Un algoritmo de compresión se puede evaluar de diferentes formas: podemos medir la complejidad relativa del algoritmo, la memoria requerida para implementar el algoritmo, la rapidez con la que el algoritmo funciona en una máquina determinada, el nivel de compresión en términos de bits ahorrados y qué tan cercana es la reconstrucción al original. En esta tesis nos enfocaremos en su complejidad y los dos últimos criterios dado que son los criterios estándar más usados al comparar este tipo de algoritmos [17, 7, 4].

Se utilizan las siguientes medidas estándar para expresar el rendimiento de un método de compresión, algunas de ellas son mencionadas a continuación:

**Razón de compresión.** Una forma de medir qué tan bien comprime un compresor es observar la razón entre el número de bits ocupados por el archivo original (antes de la compresión) y el número de bits del archivo comprimido,

$$\text{Razón de compresión} = \frac{\text{tamaño de la cadena de salida}}{\text{tamaño de la cadena de entrada}}, \tag{4.1.1}$$

que nos da un valor de razón de compresión entre 0 y 1, digamos $p$, significa que los datos ocupan, tras la compresión, un $100p\%$ de su tamaño original. Valores mayores que 1, implican un flujo de salida mayor que el de entrada (compresión negativa).

**Razón de pérdida.** Como ya se mencionó, en la compresión con pérdida la reconstrucción difiere de los datos originales. Por lo tanto, para determinar la eficiencia de un algoritmo de compresión con pérdida debemos tener alguna forma de cuantificar dicha diferencia. La razón de pérdida es la relación entre el número de pixeles invertidos (cambiados de 0 a 1 o viceversa) entre el producto del tamaño de la imagen ($m \times n$).

$$\text{Razón de pérdida} = \frac{\text{pixeles invertidos}}{mn} \tag{4.1.2}$$

Es claro que al medir el número de pixeles invertidos con esta medida siempre será menor a 1, pues el máximo de pixeles que podemos invertir es $m \times n$.

## 4.2 Análisis de los métodos jerárquicos según su calidad de compresión

Recordemos que el hecho de emplear análisis clúster jerárquicos para agrupar los objetos en clases homogéneas nos permite elegir la forma de construir los clústers, usando criterios mencionados en la Sección 2.4. Nos dimos a la tarea de comparar los diferentes métodos jerárquicos, con la finalidad de tener elementos para elegir el mejor de entre ellos.

### 4.2.1 Optimización multiobjetivo

En ocasiones, surgen problemas que requieren la optimización simultánea de más de un objetivo. Por ello, habrá que optimizar una función de la forma $f : S \longrightarrow T$, donde $S \subset R^n$ y $T \subset R^k$ . Tal problema radica en que normalmente no existe un elemento de $S$ que produzca un óptimo de forma simultánea para cada uno de los $k$ objetivos que componen $f$. Esto se deberá a la existencia de conflictos entre objetivos, que harán que la mejora de uno de ellos dé lugar a un empeoramiento de algún otro. El problema que atañe a nuestro método y en general a los métodos de compresión con pérdida, es el querer obtener la mayor compresión (razón de compresión pequeña) posible sacrificando la menor cantidad de información (razón de pérdida pequeña).

A diferencia de los problemas de optimización con un único objetivo ($k = 1$), el concepto de óptimo es ahora relativo, y será necesario decidir de alguna forma cuál es la mejor solución (o cuáles son las mejores soluciones) al problema [23].

**Definición 4.2.1** (Optimización multiobjetivo)**.** Encontrar un vector $x^* = (x_1^*, x_2^*, \ldots, x_n^*)^t$ que satisfaga las $k$ restricciones:

$$g_i(x) \geq 0 \text{ con } i = 1, 2, \ldots, k, \tag{4.2.1}$$

y optimice la función $f(x) = (f_1(x), f_2(x), \ldots, f_k(x))^t$, donde $x = (x_1, x_2, \ldots, x_n)^t$ es el vector de variables de decisión.

En otras palabras, se desea determinar la(s) solución(es) particular(es) $x^*$, del conjunto $S$ formado por todos los valores que satisfacen las restricciones (4.2.1), que dé lugar a los valores óptimos para todas las funciones objetivo.

Para tratar el problema comentado del conflicto entre objetivos existen diversos métodos, entre ellos:

**Métodos basados en la combinación de objetivos.** Dentro de estos métodos se puede mencionar el método de la suma ponderada [23], en el que se optimizará el valor obtenido mediante la suma de los valores correspondientes a los distintos objetivos, multiplicados cada uno por un coeficiente de peso. Estos coeficientes de peso establecerán la importancia relativa de cada objetivo. El problema de optimización multiobjetivo se transforma así en otro de optimización escalar, que para el caso de la minimización será de la forma:

$$\min \sum_{i=1}^{k} w_i f_i(x) \tag{4.2.2}$$

donde $w_i \geq 0$ es el coeficiente de peso correspondiente al objetivo $i$.

Existen variantes del método anterior, como el método de la programación por metas, en el que se establece una meta para cada objetivo y lo que se suma ahora (multiplicado por el correspondiente coeficiente) es la distancia de cada objetivo a su meta. Para un caso de minimización sería:

$$\min \sum_{i=1}^{k} w_i |f_i(x) - M_i| \tag{4.2.3}$$

donde $M_i$ representa la meta de $i$-ésimo objetivo.

**Métodos basados en la asignación de prioridades.** Estos métodos tienen en común que establecen unas prioridades entre los distintos objetivos, teniéndose en cuenta su importacia relativa durante el proceso de optimización [23].

**Métodos basados en el concepto de frente de Pareto.** Como ya se mencionó, un problema multiobjetivo no tiene una única solución eficiente, más bien, tiene un conjunto de soluciones eficientes que no pueden ser consideradas diferentes entre sí. A este conjunto de soluciones se le denomina Frente de Pareto [23].

**Definición 4.2.2** (Dominancia de Pareto)**.** Dado un vector $u = (u_1, u_2, \ldots, u_k)$, se dice que domina a otro vector $v = (v_1, v_2, \ldots, v_k)$, si y sólo si, para todo $i \in \{1, 2, \ldots, k\}$, $u_i \leq v_i$ y y existe $i_0 \in \{1, 2, \ldots, k\}$ talque $u_{i_0} < v_{i_0}$ (véase Figura 4.1).

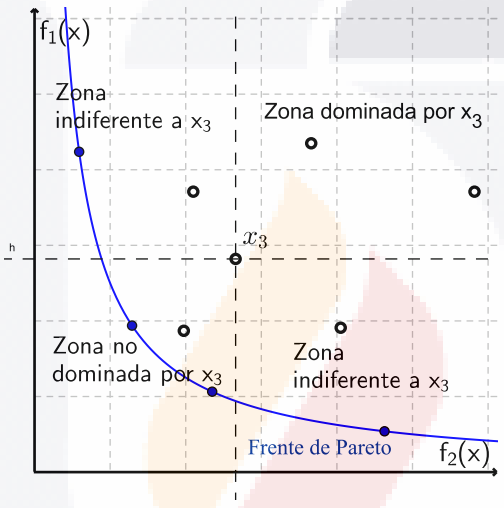



Figura 4.1: Los puntos que se encuentre en el cuadrante superior derecho delimitado por el punto $x_3$ son puntos dominados por $x_3$, los puntos que se encuentran en el cuadrante inferior izquierdo son puntos que dominan a $x_3$ y, finalmente, los dos cuadrantes restantes contienen a los puntos que no son ni dominados ni no dominados por $x_3$. La curva azul delimita el frente de Pareto.

**Definición 4.2.3** (Óptimo de Pareto)**.** Una solución $x^*$ se dice que es óptimo de Pareto, si y sólo si, no existe otro vector $x$ tal que $v = f(x) = (v_1, v_2, \ldots; v_k)$ domine a $u = f(x^*) = (u_1, u_2, \ldots, u_k)$.

La definición 4.2.3 dice que $x^*$ es un óptimo de Pareto si no existe un vector $x$ que haga mejorar alguno de los objetivos, respecto a los valores obtenidos para $x^*$, sin que empeore de forma simultánea alguno de los otros (véase Figura 4.1). Al conjunto de óptimos de Pareto es a lo que llamamos frente de Pareto.

Referente al tema de compresión de imágenes con pérdida y particularmente a nuestro método, tenemos que dependiendo del número de clústers obtenemos un número de

bytes ocupados por nuestro archivo comprimido y una pérdida de información (pixeles erróneos). Donde es de esperar, que entre menos clúster mayor será la compresión, pero también mayor será la pérdida de información.

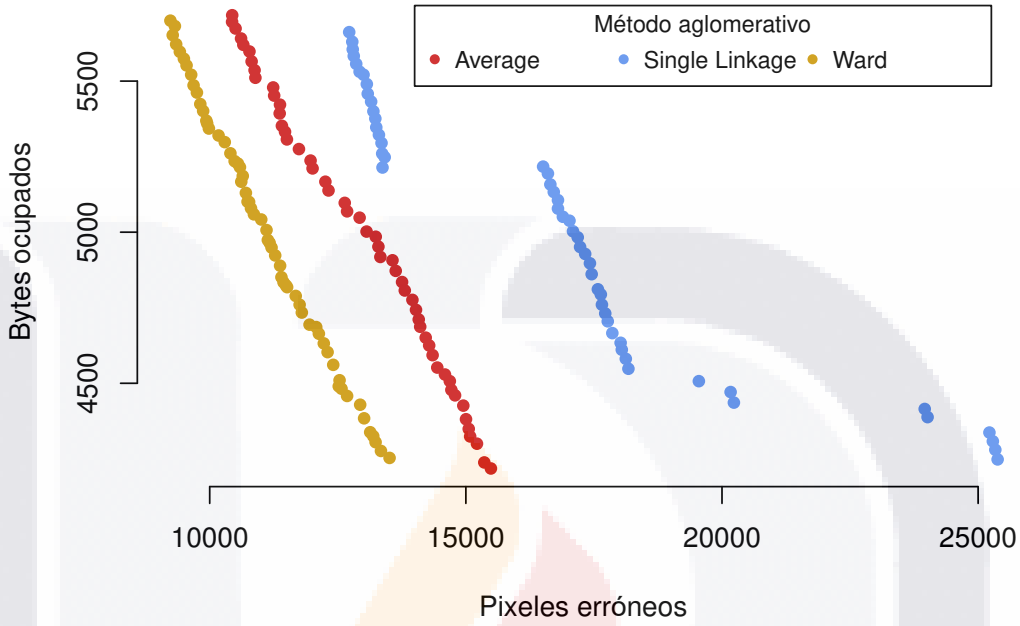



Figura 4.2: Resultados de la compresión de la Figura 4.3a comparando los bytes ocupados (en eje de las ordenadas) contra la pérdida de pixeles (en el eje de las abscisas) usando los métodos aglomerativos Promedio, Amalgamamiento Simple y Ward, con diferente número de clústers.

Debido a que tanto combinación de objetivos y asignación de proridades, son métodos un tanto subjetivos por el hecho de tener que elegir los pesos para cada objetivo se decidió emplear el frente de Pareto, pues este nos arroja toda una gama resultados de los cuales elegimos el que mejor nos parezca según nuestras necesidades.

Para poder determinar cuál de los método jerárquico presentados en la seción 2.4 arroja mejores resultados se aplicó el frente de Pareto sobre la imagen mostrada en la Figura 4.3a debido a que contiene texto e imágenen. Los resultados se muestran de forma gráfica en la Figura 4.2 exhibiendo la compresión de la imagen empleando los métodos Average, Single Linkage y Ward [1] para diferente número de clústers, mostrándose que el método Ward ayuda a nuestro compresor EntD dando mejores resultados en cuanto a bytes ocupados y pixeles erróneos, siendo este método el que nos proporciona en su totalidad los puntos óptimos de Pareto. Por esto y los resultados obtenidos en el

[1]Complete Linkage y Centroid fueron descartados debido a que arrojaban resultados deficientes comparado con estos.

resto del documento se trabajará con tal método.

### 4.2.2 Complejidad del algoritmo

Como ya se mencionó en el Capítulo 3, nuestro método EntD consiste en extraer los objetos de la imagen binivel ingresada, con los cuales se construye la matriz de distancias empleando la distancia de de Hamming modificada para posteriormente generar los $n$ representantes de los clústers generados a partir de dicha matriz. En lo anterior, observamos que la complejidad del algoritmo radica en la obtención de los $n$ clústers y en la obtención del representante de cada uno de ellos, ya que la extracción de los objetos es necesaria para la mayoría de los compresores de esta naturaleza [17, 9].

Así, dado que la obtención del representante consiste en ir empalmando objetos (de un mismo clúster) a pares, su complejidad es de orden $O(n)$. Sabemos que la complejidad del algoritmo de clusterización empleando el método Ward es $O(n^2)$ [15], obteniéndose que la complejidad de nuestro algoritmo es $O(n^2 + n) = O(n^2)$.

## 4.3 Análisis de error

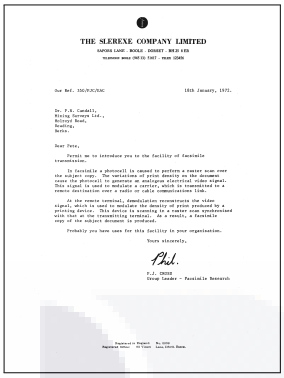
Se emplearon como benchmarks ocho imágenes propuestas por la ITU [2] (mostradas en la Figura 4.3) a diferente resolución, a saber 200, 300, 400 y 600 dpi. Los resultados obtenidos de comprimir tales imágenes se muestran en la Tabla 4.1, en la cual podemos remarcar que en cuanto a nivel de compresión estamos por encima en el 84% de las imágenes a las diferentes resoluciones, denotando una mejora significativa.

En esta sección, evaluamos la calidad de EntD utilizando dos medidas de calidad, que son el porcentaje de razón de compresión y el porcentaje de razón de pérdida. Nuestras comparaciones con otros métodos disponibles en la literatura[2] mostrarán que el método propuesto en este trabajo tiene un buen rendimiento de compresión con una pequeña pérdida de información.

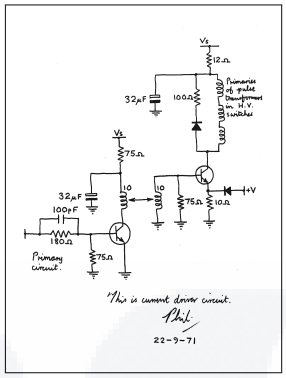
Para fines de comparación, utilizaremos como puntos de referencia las imágenes del CCITT mostradas en la Figura 4.3. Tales imágenes se han utilizado ampliamente en la literatura como pruebas para la compresión de imágenes binarias. En particular, han sido utilizados por la ITU, y se obtuvieron de la página de documentos de referencia de la ITU [2] considerando resoluciones de 200, 300, 400 y 600 dpi (que son tamaños

[2]Debido a que existe una gran variedad de variantes del los métodos JBIG2 y JB2, se usó el software proporcionado por la compañia Cuminas [1] para la obtencióin de los datos concentrados en la Tabla 4.1.

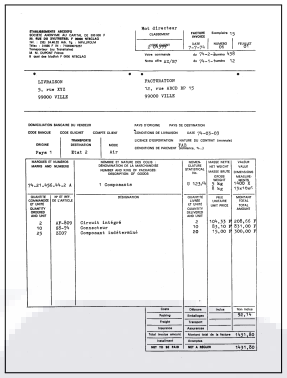
estándar en el trabajo diario) y métodos con y sin pérdida.



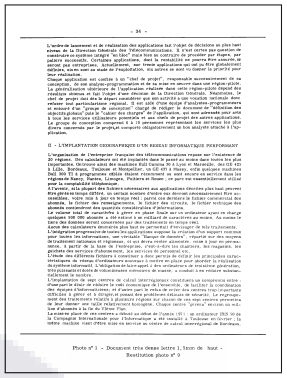
(a) F01: Carta en inglés.



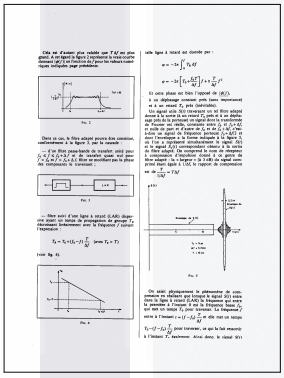
(b) F02: Circuito hecho a mano.



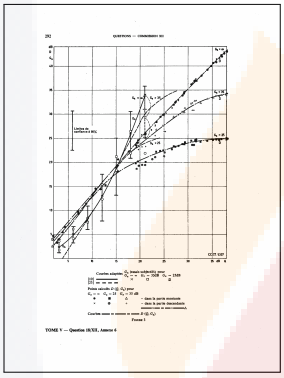
(c) F03: Factura en francés.



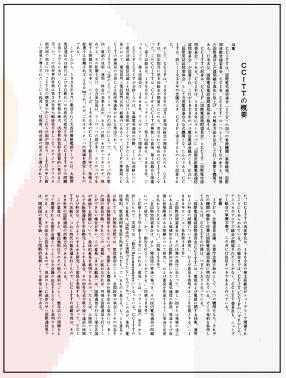
(d) F04: Texto en francés.



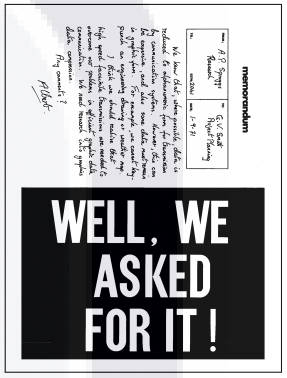
(e) F05: Texto en francés con figuras.



(f) F06: Gráfico en francés.



(g) F07: Kanji.




(h) F08: Memorándum manuscrito.

Figura 4.3: Imágenes de prueba obtenidas de la ITU.

**Razón de compresión.** El lado izquierdo de la Tabla 4.1 muestra el porcentaje la razón de compresión para las ocho imágenes de la Figura 4.3, usando las cuatro resoluciones. Referente a estos datos, vale la pena señalar que los porcentajes más bajos se obtienen en su mayoría con EntD superado por RDPD sólo en cinco del total de pruebas. Siendo sólo en dos de ellas, la eficacia de RDPD sustancialmente mejor que nuestra técnica, siendo esta la imagen de la Figura 4.3g para resoluciones de 300 y 400 dpi. Para una mejor apreciación, los resultados se presentan gráficamente en el lado izquierdo de la Figura 4.4, para resoluciones de imagen de 200, 300, 400 y 600 dpi.

**Razón de pérdida.** El lado derecho de la Tabla 4.1 presenta el porcentaje de pixeles invertidos totales para diferentes métodos con pérdida, con imágenes a resolu-

ciones de imagen de 200, 300, 400 y 600 dpi. Resultados mostrados de forma gráfica en el lado derecho de la Figura 4.4 correspondientes a las diferentes resoluciones ya mencionadas. El porcentaje de pérdida de información perdida es en promedio del 0.323% a una resolución de 600 dpi, lo que es bajo, teniendo en cuenta el rango de compresión alcanzado. EntD muestra mejoras de hasta 35% con respecto a JB2. Desafortunadamente, no existe un método con pérdida que se destaca como el mejor. Sin embargo, JBIG2 y EntD han mostrado las mejores actuaciones siendo la primera el mejor en la mayoría de las pruebas.

Resumiendo, nuestro método propuesto EntD muestra un mejor rendimiento de compresión con respecto a los del estado del arte. Por otro lado, los métodos JB2 y JBIG2 muestran un mejor rendimiento en términos de porcentajes de razón de pérdida que nuestro método y el RDPD. Sin embargo, es importante señalar que las razones de compresión de las técnicas JB2 y JBIG2 son considerablemente más altas que el método presentado en este documento. En general, hemos demostrado que la metodología actual es una técnica de compresión competitiva cuando se compara con otros métodos estándar disponibles hoy en día.

| Imagen | Porcentaje de razón de compresión | | | | | | | Porcentaje de razón de pérdida | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Imágenes a 200dpi (1728 × 2339 pixeles) | | | | | | | | | | |
| | DjVu | Paq8l | G4 | JBIG2 | JB2 | RDPD | EntD | JBIG2 | JB2 | RDPD | EntD |
| F01 | 63.493 | 78.551 | 54.043 | 24.043 | 19.793 | **15.261** | 15.267 | 0.293 | 0.284 | 0.289 | **0.284** |
| F02 | 41.509 | 73.373 | 33.893 | 24.564 | 24.477 | 17.892 | **17.836** | **0.007** | 0.010 | 0.093 | 0.075 |
| F03 | 53.550 | 72.688 | 46.395 | 24.183 | 21.858 | 13.037 | **12.209** | 0.360 | 0.369 | 0.565 | **0.357** |
| F04 | 79.988 | 89.446 | 70.962 | 21.920 | 18.616 | 12.880 | **12.578** | 1.283 | 1.270 | 1.254 | **1.170** |
| F05 | 59.576 | 79.450 | 51.047 | 20.699 | 19.472 | 12.867 | **12.785** | **0.516** | 0.562 | 0.587 | 0.602 |
| F06 | 39.219 | 56.104 | 33.893 | 22.765 | 22.209 | 16.820 | **15.976** | **0.069** | 0.069 | 0.128 | 0.151 |
| F07 | 77.432 | 94.398 | 69.924 | 34.232 | 28.678 | 25.336 | **24.377** | 0.975 | 0.955 | **0.865** | 1.287 |
| F08 | 35.330 | 59.114 | 31.150 | 21.654 | 21.991 | 17.122 | **17.084** | 0.031 | **0.029** | 0.117 | 0.109 |
| | Imágenes a 300dpi (2592 × 3508 pixeles) | | | | | | | | | | |
| | DjVu | Paq8l | G4 | JBIG2 | JB2 | RDPD | EntD | JBIG2 | JB2 | RDPD | EntD |
| F01 | 38.57 | 76.79 | 45.39 | 18.84 | 15.59 | 9.41 | **9.24** | **0.230** | 0.234 | 0.271 | 0.259 |
| F02 | 25.05 | 68.90 | 29.87 | 20.35 | 21.32 | 14.81 | **14.35** | **0.003** | 0.003 | 0.081 | 0.076 |
| F03 | 32.92 | 70.10 | 38.96 | 20.07 | 18.47 | 8.93 | **8.06** | **0.279** | 0.279 | 0.542 | 0.537 |
| F04 | 45.65 | 85.22 | 56.97 | 15.58 | 13.08 | **7.25** | 7.29 | 1.088 | **1.071** | 1.168 | 1.168 |
| F05 | 35.98 | 76.14 | 42.61 | 16.12 | 15.04 | 8.56 | **7.98** | 0.431 | **0.413** | 0.496 | 0.484 |
| F06 | 24.76 | 56.24 | 30.32 | 19.34 | 19.78 | 13.44 | **13.18** | **0.049** | 0.054 | 0.107 | 0.099 |
| F07 | 47.74 | 92.02 | 57.92 | 26.05 | 22.30 | **15.25** | 17.48 | 0.807 | **0.768** | 0.882 | 0.881 |
| F08 | 20.96 | 56.94 | 27.79 | 17.85 | 18.76 | 13.16 | **12.80** | **0.011** | 0.013 | 0.111 | 0.122 |
| | Imágenes a 400dpi (3456 × 4677 pixeles) | | | | | | | | | | |
| | DjVu | Paq8l | G4 | JBIG2 | JB2 | RDPD | EntD | JBIG2 | JB2 | RDPD | EntD |
| F01 | 55.18 | 72.10 | 38.76 | 15.70 | 13.13 | 6.79 | **6.51** | **0.187** | 0.190 | 0.244 | 0.240 |
| F02 | 40.36 | 65.12 | 27.23 | 17.90 | 18.92 | 12.68 | **11.74** | **0.002** | 0.003 | 0.078 | 0.072 |
| F03 | 49.99 | 69.36 | 35.07 | 18.85 | 17.27 | 7.41 | **6.96** | **0.224** | 0.239 | 0.486 | 0.480 |
| F04 | 67.22 | 81.43 | 48.87 | 12.75 | 10.65 | 5.41 | **5.15** | 0.935 | **0.918** | 1.054 | 0.934 |
| F05 | 53.68 | 73.22 | 37.54 | 13.55 | 12.84 | 6.62 | **6.37** | **0.359** | 0.360 | 0.419 | 0.401 |
| F06 | 38.81 | 53.95 | 27.12 | 17.46 | 17.86 | 11.37 | **11.07** | **0.0397** | 0.0405 | 0.104 | 0.101 |
| F07 | 71.31 | 89.16 | 50.29 | 21.47 | 18.50 | **11.49** | 14.63 | 0.672 | 0.676 | 0.798 | **0.658** |
| F08 | 34.09 | 52.27 | 24.69 | 15.50 | 16.47 | 10.79 | **10.69** | **0.004** | 0.007 | 0.112 | 0.107 |
| | Imágenes a 600dpi (5184 × 7016 pixeles) | | | | | | | | | | |
| | DjVu | Paq8l | G4 | JBIG2 | JB2 | RDPD | EntD | JBIG2 | JB2 | RDPD | EntD |
| F01 | 53.01 | 73.65 | 35.21 | 20.00 | 13.32 | 5.31 | **4.97** | **0.159** | 0.171 | 0.246 | 0.234 |
| F02 | 35.62 | 58.43 | 24.08 | 15.89 | 16.40 | 10.55 | **10.15** | **0.0005** | 0.001 | 0.077 | 0.062 |
| F03 | 45.53 | 65.12 | 31.30 | 18.36 | 17.00 | 5.86 | **5.24** | **0.161** | 0.161 | 0.461 | 0.510 |
| F04 | 59.52 | 77.54 | 43.94 | 14.74 | 11.62 | 3.47 | **3.41** | 0.800 | **0.821** | 1.022 | 0.920 |
| F05 | 48.46 | 69.96 | 33.61 | 14.54 | 12.56 | 5.10 | **4.63** | **0.276** | 0.286 | 0.428 | 0.399 |
| F06 | 39.20 | 56.50 | 26.57 | 18.80 | 17.72 | 10.08 | **10.00** | **0.025** | 0.0302 | 0.110 | 0.088 |
| F07 | 61.49 | 84.13 | 42.25 | 25.71 | 23.67 | **11.49** | 12.84 | 0.119 | 0.127 | 0.256 | **0.246** |
| F08 | 35.02 | 54.27 | 24.67 | 16.28 | 16.37 | 9.91 | **9.77** | **0.001** | 0.004 | 0.124 | 0.121 |

Tabla 4.1: Resultados de las imágenes mostradas en la Figura 4.3 después de la compresión con diferentes compresores.


Figura 4.4: Resultados de compresión de las imágenes mostradas en la Figura 4.3. A la izquierda se muestra el procentaje de razón de compresión y del lado derecho el porcentaje de razón de pérdida.

## Conclusiones

En este trabajo, hemos propuesto un método para la selección de representantes de clústers en la compresión de imágenes de documentos. La técnica presentada identifica inicialmente los diferentes objetos en una imagen. A su vez, los objetos son agrupados en clases a través del método jerárquico Ward, y los representantes de la clase son generados usando un criterio de mínima entropía. Varias pruebas fueron realizadas utilizando archivos de documentos estándar a diferentes resoluciones. Los resultados presentados en este trabajo muestran que la metodología propuesta tiene un mejor rendimiento de compresión al compararse con JBIG2, JB2 y otros compresores del estado del arte. Esto puede deberse al hecho de que el enfoque de entropía mínima produce mejores representantes para los clústers.

Otra ventaja del método presentado en este trabajo es que solo requiere un parámetro, a saber, el número de clases empleadas por el algoritmo. Este parámetro tiene un impacto directo en la compresión y los niveles de pérdida. De hecho, cuantos más clústers se utilicen, menor será el nivel de compresión y se perderá menos información. Por el contrario, cuanto menor sea el número de clústeres, mayor será el nivel de compresión y más información se perderá. El criterio de entropía permite crear iterativamente un patrón no binario que garantiza una mayor concentración de pixeles superpuestos en cada paso. La binarización de este patrón produce un representante de clúster con aproximadamente los mismos errores positivos (de pixeles blancos a negros) y errores negativos (de pixeles negros a blancos). Además, la presente técnica tiene control sobre el número de agujeros de los objetos, lo que facilita la identificación de las diferencias topológicas entre los símbolos. Esta característica mejora mucho los métodos como JB2 y RDPD.

## Referencias


[1] Cuminas. https://www.cuminas.jp/en, Noviembre 2017.

[2] Itu reference documents - bi-level images. https://www.itu.int/net/itu-t/sigdb/genimage/test24.htm, Noviembre 2017.

[3] Tinku Acharya and Ajoy K Ray. *Image processing: principles and applications*. John Wiley & Sons, 2005.

[4] Leon Bottou, Patrick Haffner, Paul G Howard, Patrice Simard, Yoshua Bengio, and Yann Le Cun. High quality document image compression with. *Journal of Electronic Imaging*, 7(3):410–426, 1998.

[5] John Cleary and Ian Witten. Data compression using adaptive coding and partial string matching. *IEEE transactions on Communications*, 32(4):396–402, 1984.

[6] Wolfgang Ertel. *Introduction to artificial intelligence*. Springer Science & Business Media, 2011.

[7] Paul G Howard, Faouzi Kossentini, Bo Martins, Søren Forchhammer, and William J Rucklidge. The emerging jbig2 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7):838–848, 1998.

[8] Roy Hunter and A Harry Robinson. International digital facsimile coding standards. *Proceedings of the IEEE*, 68(7):854–867, 1980.

[9] O Johnsen, J Segen, and GL Cash. Coding of two-level pictures by pattern matching and substitution. *Bell Labs Technical Journal*, 62(8):2513–2545, 1983.

[10] Markus Kuhn. Jbig-kit. *University of Cambridge.[Online]. Available: http://www. cl. cam. ac. uk/mgk25/jbigkit*, 1995.

[11] Simon Lhuilier. *De relatione mutua capacitatis et terminorum figurarum, geometricč considerata seu de maximis et minimis, pars prior elementaris*. M. Gröll, 1782.

[12] Hiram H López-Valdez, Hermilo Sánchez-Cruz, and Magdalena C Mascorro-Pantoja. Single chains to represent groups of objects. *Digital Signal Processing*, 51:73–81, 2016.

[13] Matthew V Mahoney. Adaptive weighing of context models for lossless data compression. Technical report, 2005.

[14] Kenneth R McConnell, Dennis Bodson, and R Schaphorst. Fax, digital facsimile technology and applications, artech house, 1992.

[15] Fionn Murtagh and Pierre Legendre. WardâĂŹs hierarchical agglomerative clustering method: which algorithms implement wardâĂŹs criterion? *Journal of classification*, 31(3):274–295, 2014.

[16] Mohammad Norouzi, David J Fleet, and Ruslan R Salakhutdinov. Hamming distance metric learning. In *Advances in neural information processing systems*, pages 1061–1069, 2012.

[17] Mario A Rodríguez-Díaz and Hermilo Sánchez-Cruz. Refined fixed double pass binary object classification for document image compression. *Digital Signal Processing*, 30:114–130, 2014.

[18] Azriel Rosenfeld and Reinhard Klette. Digital geometry. *Information Sciences*, 148(1-4), 2002.

[19] Hermilo Sánchez-Cruz, Ernesto Bribiesca, and Ramón M Rodríguez-Dagnino. Efficiency of chain codes to represent binary objects. *Pattern Recognition*, 40(6):1660–1674, 2007.

[20] T Series. Terminal equipments and protocols for telematic services. *Data Protocols for Multimedia Conferencing, International Telecommunication Union, ITU-T Telecommunication Standardization Sector of ITU*.

[21] Claude E Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:623–656, 1948.

[22] Claude E Shannon and Warren Weaver. The mathematical theory of communication. urbana, ill. *Univ. Illinois Press*, 1:17, 1949.

[23] Ricardo Smith. *Decisiones con múltiples objetivos e incertidumbre*. Universidad Nacional de Colombia, 1993.

[24] Taffee T Tanimoto. Ibm internal report. *Nov*, 17:1957, 1957.

[25] Dave AD Tompkins and Faouzi Kossentini. A fast segmentation algorithm for bi-level image compression using jbig2. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 1, pages 224–228. IEEE, 1999.

[26] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[27] Ramón Pedret Yerba. *Técnicas cuantitativas al servicio del marketing: Métodos descriptivos de análisis multivariable*. PhD thesis, Universitat de Barcelona, 1986.

# Anexo A

# Métodos jerárquicos

## Promedio (Average)

Si tomamos dos clústers, $C_i$ y $C_j$, donde el clúster $C_i$ está formado, a su vez, por otros dos clusters, $C_{i1}$ y $C_{i2}$ (con $n_{i1}$ y $n_{i2}$ elementos respectivamente, en consecuencia $n_i = n_{i1} + n_{i2}$ el número de elementos de $C_i$, y $n_j$ el número de elementos de $C_j$) . En términos de similitudes (disimilitudes), el promedio sería:

$$\begin{aligned}
s(C_i, C_j) =& \frac{1}{(n_{i1} + n_{i2})n_j} \sum_{l=1}^{n_{i1}+n_{i2}} \sum_{m=1}^{n_j} s(O_l, O_m) \\
=& \frac{1}{(n_{i1} + n_{i2})n_j} \sum_{l=1}^{n_{i1}} \sum_{m=1}^{n_j} s(O_l, O_m) + \frac{1}{(n_{i1} + n_{i2})n_j} \sum_{l=1}^{n_{i2}} \sum_{m=1}^{n_j} s(O_l, O_m) \\
=& \frac{n_{i1}}{(n_{i1} + n_{i2})n_{i1}n_j} \sum_{l=1}^{n_{i1}} \sum_{m=1}^{n_j} s(O_l, O_m) + \frac{n_{i2}}{(n_{i1} + n_{i2})n_{i2}n_j} \sum_{l=1}^{n_{i2}} \sum_{m=1}^{n_j} s(O_l, O_m) \\
=& \frac{n_{i1}}{n_{i1} + n_{i2}} s(C_{i1}, C_j) + \frac{n_{i2}}{(n_{i1} + n_{i2})n_{i2}n_j} s(C_{i2}, C_j) \\
=& \frac{n_{i1}s(C_{i1}, C_j) + n_{i2}s(C_{i2}, C_j)}{n_{i_1} + n_{i2}}.
\end{aligned}$$

## Centroide

La distancia euclídea cuadrática entre los clústers $C_i$ y $C_j$ vendrá dada por:

$$
\begin{aligned}
d_2^2(C_i, C_j) =& \sum_{l=1}^{n} (m_l^j - m_l^i)^2 \\
=& \sum_{l=1}^{n} \left[ m_l^j - \frac{n_{i1} m_l^{i1} + n_{i2} m_l^{i2}}{n_{i1} + n_{i2}} \right] \\
=& \sum_{l=1}^{n} \left[ (m_l^j)^2 - 2 m_l^j \frac{n_{i1} m_l^{i1} n_{i2} m_l^{i2}}{n_{i1} + n_{i2}} \right. \\
&+ \frac{(n_{i1})^2 (m_l^{i1})^2 + (n_{i2})^2 (m_l^{i2})^2 + n_{i1} n_{i2} (m_l^{i1})^2 + n_{i1} n_{i2} (m_l^{i2})^2}{(n_{i1} + n_{i2})^2} \\
&\left. + \frac{-n_{i1} n_{i2} (m_l^{i1})^2 - n_{i1} n_{i2} (m_l^{i2})^2 + 2 n_{i1} n_{i2} m_l^{i1} m_l^{i2}}{(n_{i1} + n_{i2})^2} \right] \\
=& \sum_{l=1}^{n} \left[ \frac{n_{i1} (m_l^j)^2 + n_{i2} (m_l^j)^2}{n_{i1} + n_{i2}} + \frac{n_{i1} (m_l^{i1})^2}{n_{i1} + n_{i2}} + \frac{n_{i2} (m_l^{i2})^2}{n_{i1} + n_{i2}} \right. \\
&\left. - 2 m_l^j \frac{n_{i1} m_l^{i1}}{n_{i1} + n_{i2}} - 2 m_l^j \frac{n_{i2} m_l^{i2}}{n_{i1} + n_{i2}} - \frac{n_{i1} n_{i2}}{(n_{i1} + n_{i2})^2} (m_l^{i1} - m_l^{i2})^2 \right] \\
=& \sum_{l=1}^{n} \left[ \frac{n_{i1}}{n_{i1} + n_{i2}} (m_l^j - m_l^{i1})^2 + \frac{n_{i2}}{n_{i1} + n_{i2}} (m_l^j - m_l^{i2})^2 - \frac{n_{i1} m_{i2}}{(n_{i1} + n_{i2})^2} (m_l^{i1} - m_l^{i2})^2 \right] \\
=& \frac{n_{i1}}{n_{i1} + n_{i1}} d_2^2(C_{i1}, Cj) + \frac{n_{i2}}{n_{i1} + n_{i1}} d_2^2(C_{i2}, Cj) - \frac{n_{i1} n_{i2}}{(n_{i1} + n_{i1})^2} d_2^2(C_{i1}, C_{i2}),
\end{aligned}
$$