Universidad de Valladolid

Escuela Técnica Superior de Ingenieros de la Telecomunicación

Grado en Tecnologías de la Telecomunicación

Herramienta para la generación y análisis de imágenes con contenido esteganográfico

Autor:
Justino Rafael Rodríguez-Galván

Tutores:
Dr. Federico Simmross-Wattenberg
Dr. Santiago Aja-Fernández

Marzo de 2019

Resumen

La esteganografía es la ciencia de esconder mensajes ocultos en medios portadores «inocentes», mientras que el estegoanálisis es su opuesto, la detección de posibles mensajes ocultos disfrazados en dichos soportes. En este trabajo, se han estudiado estas ciencias aplicadas a la manipulación digital de imágenes.

Dada la naturaleza de la modificación de las imágenes, el mensaje se convertirá en ristras de bits que pueden ir desde información hasta código ejecutable y es en este último caso en el que más nos centraremos, puesto que podría explotar las vulnerabilidades del intérprete de imágenes o *parser* y ser ejecutado sin conocimiento del usuario.

En este trabajo se ha estudiado la capacidad de inserción de datos en una imagen mediante la propuesta de tres métodos, cuyos resultados se han discutido en base a tres baremos, QILV, SSIM y PSNR, así como un método de estegoanálisis para la comprobación de la robustez de uno de los métodos de esteganografiado propuestos.

El primer método propuesto está basado en la inserción de los datos en las zonas de alta frecuencia espacial, utilizando dos variables para adaptación a la densidad de datos a esteganografiar. La primera variable es la dimensión del *kernel* que servirá para realizar el filtro de desviación típica y el segundo será un umbral normalizado, para determinar qué valores son susceptibles de ser esteganografiados. El segundo, por su parte, utilizará el dominio transformado DCT, así como el algoritmo de compresión del formato JPEG, modificando la matriz de cuantización del mismo. Finalmente, el último método presentado estará basado en el dominio transformado SVD.

En cuanto al método de estegoanálisis presentado, se ha basado en ataques de fuerza bruta de texto claro con mensaje desconocido y texto claro con mensaje conocido para comprobar cuán robusto es el primero de los métodos que se presentan.

Agradecimientos

La primera persona a la que quiero dedicar un agradecimiento es a mi abuela Paulina, la mujer con la que más tiempo he pasado en mi infancia, puesto que se ocupaba de mí mientras mis padres trabajaban. A ella he de agradecerle el cariño con el que me crió y todas las enseñanzas que me transmitió. Si bien ella ya no va a poder ver cómo presento este trabajo, está muy presente en él, y puedo sentir su orgullo al titularme finalmente así como también recibo su fuerza para enfrentar los retos venideros.

También tengo que mencionar a mis padres, Viky y Tino, puesto que sin ellos esta travesía no hubiera llegado a buen puerto. Debo agradecer la confianza que depositaron en mí y todo el apoyo recibido durante todos estos años, así como su cariño, que suavizó la mayoría de tragos amargos y todas sus lecciones, que han forjado mi carácter.

A mi tía Yoli, por invitarme a comer un día a la semana a comer en su casa e interesarse tanto por el devenir de mis estudios, así como las sobremesas de descanso viendo alguna película, con una interesante conversación.

A mis tutores, y mentores, Federico Simmross y Santiago Aja, estandartes de esta profesión y grandes profesores que han tenido a bien dirigir este proyecto y me han brindado su ayuda, conduciéndolo desde su experiencia para solventar aquellos baches que fueron surgiendo y corrigiéndome los fallos que se sucedían por mi inexperiencia.

A Raquel, por estar ahí, en las idas y en las venidas; sigo preguntándome como una casualidad puede marcar tanta diferencia y convertirse en tanta felicidad. Tengo que agradecer que quieras construir una vida conmigo, que estés dispuesta a aguantarme y a consolarme durante este camino, que si bien estará lleno de baches, habrá momentos —como éste— en los que podremos parar a tomar aire y al echar la vista atrás, conteplarlos con una sonrisa. Tampoco puedo olvidarme de tu inestimable ayuda revisando estos *ladrillos*, demostrando el amor que me profesas.

Por último, me gustaría agradecer a todas las personas que me han ayudado a llegar hasta este punto, que me han ofrecido su compañía y conocimientos y con los que he disfrutado de buenos momentos durante estos años de carrera, ya fuera para celebrar logros o para olvidar tragos amargos.

Tabla de Contenidos

Capítulo 1

Introducción

1.1. Motivación

Desde la antiguedad, la protección de la información ha sido esencial. La ocultación de las tácticas ante el enemigo era básica para tener alguna posibilidad de alcanzar la victoria, para ello, la información se cifraba mediante un código secreto utilizando diferentes métodos, según la época. Por otra parte, el enemigo, consecuentemente, quería descifrar dichas tácticas para obtener una ventaja. Estas técnicas se engloban en el arte de la **criptología**, que, según [Rea17], es el estudio de los sistemas, claves y lenguajes ocultos y secretos. Esta ciencia, a su vez, se subdivide en otras cuatro que pasamos a describir ahora:

- La **Criptografía** es el arte de escribir con clave secreta o de un modo enigmático [Rea17]. Los métodos de ocultación de la información son muy variopintos, i.e. reordenar las palabras de una oración, sustituir letras por símbolos, sustituir letras por otras letras... en definitiva, proteger un mensaje de terceras personas alterando el mismo mediante un código sólo conocido por los interesados. El ejemplo más gráfico de criptografía lo encontramos en la Segunda Guerra Mundial. En ese momento, el ejército británico interceptaba las comunicaciones del Reich, pero sólo encontraba un galimatías compuesto por una sucesión de caracteres sin sentido aparente. En efecto, los alemanes contaban con un artefacto que cifraba sus comunicaciones con mucha eficacia, la máquina Enigma [Mar00].

- El **Criptoanálisis** es la parte de la criptología que, según [Rea17], se define como el arte de descifrar criptogramas. Siguiendo con el ejemplo anterior, el equipo británico que estudió (y descifró) las dichas comunicaciones podrían comportar el equipo con más importancia de esta índole de la historia. El equipo, comandado por Alan Turing, consiguió desencriptar el cifrado Enigma mediante la computación, utilizando las máquinas *Colossus* y *The Turing bombe*. Cabe destacar la importancia de esta operación, puesto que tras descifrar dicho código, el equipo de Turing se dedicó a aplicar modelos probabilísticos para decidir qué actos se podían llevar a cabo sin que el ejército del Reich sospechara de sus comunicaciones, dando cuenta de la importancia de la información [Mar00].

- La siguiente parte de la criptología que se introducirá es la **Esteganografía**. Esta parte ya no comparte la misma raíz, *kryptos*, sino que es $\sigma\tau\eta\gamma\nu\circ\varsigma$ *steganos*, que también significa oculto. La principal diferencia entre la criptografía y la esteganografía es que en ésta última se pretende que la comunicación pase desapercibida. El ejemplo por antonomasia es la escritura con zumo de limón, que pasa desapercibida a todos menos al destinatario del mensaje que lo revelará con tinta china o acercando una vela. Finalmente, el **Estegoanálisis** es la parte análoga al criptoanálisis, pero esta vez referida a la esteganografía.

El ejemplo escogido anteriormente para la criptografía no ha sido azaroso, puesto que compone un hito en el que una máquina de computación vence a una máquina mecánica a la hora de la encriptación de las comunicaciones, por tanto, desde ese momento se buscarían formas de encriptación utilizando esta nueva y potente tecnología. Si avanzamos en el tiempo a nuestra época, comprobaremos que el mundo ha cambiado enormemente. Vivimos en un mundo en línea, siempre conectados, enviando datos personales (sensibles en muchas ocasiones), a través de la red. De esta difusión de información personal, así como de la extensión de servicios distribuidos por Internet, surgió el concepto de **Seguridad Informática** [Sta17]:

La Seguridad Informática se aplica a todos los procedimientos necesarios para preservar la **Intregridad, Disponibilidad y Confidencialidad** de los recursos de un sistema informático. A continuación se procederá a explicar los elementos de esta tríada:

- La **Confidencialidad** es la propiedad que debe asegurar la privacidad de los usuarios, así como la información de los mismos, considerando como una pérdida la divulgación de dichos recursos de forma no autorizada. [Sta17]
- La **Integridad** consiste en la imposibilitación de la destrucción o modificación, incluyendo autenticidad y norepudio, de los recursos de una estación, siendo en esete caso una pérdida la consecución de las acciones antes descritas [Sta17].
- Finalmente, la **Disponibilidad** se corresponde con un acceso ágil y fiable a los recursos. La imposición de trabas a la hora de recibir los recursos se considera una pérdida [Sta17].

En este sentido, la criptografía se ha centrado en esconder la información que se envía desde los ordenadores hacia Internet, para protegerla de cualquiera que esté captando el tráfico que circula por una red. También se utiliza para preservar las Redes Inalámbricas de terceros, es decir, para impedir que desconocidos roben la clave de una red inalámbrica privada. Si utilizamos un Sistema Operativo GNU/Linux, podremos comprobar también que la información relativa a las contraseñas de usuario se almacena de forma segura y cifrada en el fichero `/etc/shadow`...

Por otra parte, el sendero tomado por la esteganografía ha sido el de modular ficheros aparentemente inocentes, tales como ficheros de audio o imagen para embeber en ellos mensajes secretos o códigos. Por ejemplo, en [Vij17] se comenta el uso de la esteganografía para esconder datos que se han robado en ataques de ciberespionaje o en [Vij18] se comenta el uso de *Memes* que circulan por Twitter para enviar instrucciones a un *malware* de tipo Caballo de Troya para tomar capturas de pantalla u otras funciones maliciosas.

Sin embargo, la esteganografía esconde más utilidades. Dada la naturaleza de las páginas web, en las que se descargan automáticamente las fotografías incrustadas, el artículo [Bol17] pone en relieve el compromiso en el que se podría poner un dispositivo en el caso de que interpretara el código malicioso oculto en una imagen. Según este artículo, las defensas ante estos posibles ataques estarían aún en sus primeras fases de desarrollo, por lo que se pretende estudiar este campo para determinar cuán factibles son las proposiciones del mismo y, de ser así, tratar de encontrar un método para salvaguardar la Seguridad Informática, previamente descrita.

1.2. Objetivos

El objetivo principal de este proyecto es el que da nombre al mismo: la **creación de una herramienta para la generación y análisis de imágenes con contenido esteganográfico**, si bien deberemos dividirlo en subobjetivos:

- Creación de diversos métodos de esteganografiado, utilizando técnicas diferentes, tales como modificación del bit menos significativo y dominios transformados, tratando de que uno de ellos sea el de la DCT por su relación con el formato JPEG. Implícitamente, cada algoritmo debe ser reversible, en el sentido de que se deben poder extraer los datos intactos. Cabe destacar que el formateo del mensaje a la hora de la recuperación excede las competencias de este trabajo. Esta especificación se recordará en la definición de los métodos.
- Evaluación de los anteriores métodos mediante diversos baremos. Éstos serán medidas de similitud estructural entre la imagen portadora y la *estegoimagen* en relación con la tasa de modulación de la portadora. Además, se valorará la posibilidad o imposibilidad de formatear los datos obtenidos a una extensión conocida.
- Implementación de una herramienta de estegoanálisis genérica, así como el simulacro de un ataque de fuerza bruta; esto es, una tentativa de extracción de los datos, suponiendo en primer lugar mensaje desconocido y posteriormente conocido, aplicando *ceteris paribus* a todos los parámetros de esteganografiado salvo a uno, sobre el que se hará un barrido.

1.3. Fases y métodos

La metodología utilizada para llevar a cabo los objetivos previamente expuestos será:

1. En primer lugar se estudiará qué métodos son los más idóneos para el esteganografiado, teniendo en cuenta cómo impacta la modulación a la percepción de la imagen, en primera instancia, así como la posible aplicación de una transformación previa, etc. En este fase también se llevará a cabo un estudio sobre los diferentes formatos de imagen, pues en ellos veremos que se aplican diferentes transformaciones y algoritmos de compresión que afectarán a la modulación, contando también con la extensión de su uso.
2. Tras la elección de los diversos métodos, se procederá a la creación de un primer prototipo que implemente las diferentes fases de modulación, i.e. la modificación de la imagen original con la consiguiente comprobación de que tal modificación se ha hecho patente. En esta fase se prescindirá de información cromática para facilitar la misma, y se prescindirá de ella hasta haber avanzado notablemente en el diseño del prototipo. El mensaje modulado se compondrá de una ristra de bits generados de forma pseudoaleatoria.
3. La siguiente fase consistirá en generar un prototipo que recupere la información incrustada, comprobando que no existan errores a la hora de la recuperación.
4. Una vez completados los anteriores puntos, se evualuará utilizando diversos métodos para cuantificar el impacto de la modulación sobre la imagen original. Los métodos, que serán posteriormente descritos, tratarán de emular la percepción humana.
5. Tras la consecución de todos los anteriores puntos, se completará el prototipo añadiendo características, como color (decidiendo previamente qué base sería la óptima), y efectuando cambios en la cantidad de datos a embeber para comprobar la adaptabilidad de los diferentes algoritmos a la cantidad de información y el consiguiente impacto visual.

Seguidamente, se procederá a desglosar las fases de la aplicación de estegoanálisis:

1. De forma análoga a la parte anterior, se estudiarán aquellas zonas y metodologías más propicias para ser esteganografiadas, prescindiendo nuevamente del color. Dado que en esta ocasión trataremos de descubrir si existe o no modulación, se compararán partes similares de la imagen, la compresión existente por zonas, etc.

2. Habiendo concluido el estudio anterior, nuestra siguiente meta será el diseño de un prototipo de un analizador de imágenes con un decisor que indique si existe esteganografía. En esta fase del proyecto, se utilizarán imágenes de prueba exageradamente modificadas para asegurar el correcto funcionamiento del algoritmo.

3. El siguiente avance que se buscará será el de afinar las zonas del detector. Se tratará de encontrar los umbrales de falsa alarma y no detección procurando esta vez imágenes con modulaciones más sutiles y de diferentes métodos.

4. Seguidamente, se tratará de evaluar el prototipo, alternando imágenes esteganografiadas y meras portadoras, para evaluar la efectividad del detector.

5. Finalmente, y de forma análoga al último subobjetivo de la anterior parte, se modificará el prototipo para atender a la información cromática de la imagen.

1.4. Estructura del documento

El presente trabajo contará con la estructura que se describe a continuación.

En primer lugar, en el capítulo 2, se pretende contextualizar el estudio que se va a realizar en el resto del trabajo, referido, como ya se ha expuesto, a la esteganografía y al estegoanálisis. En él podremos encontrar, a modo introductorio, en la sección 2.1, conceptos básicos de cómo interpretamos las imágenes, las células sensibles y algunas definiciones básicas de las componentes de las mismas, tales como la Luminancia, el Color, etc. En ella también se expondrán diferentes experimentos que se desarrollaron para caracterizar la visión humana como si de un sistema se tratara. La siguiente sección de este capítulo, y principal, se centrará ya en el título de la misma, definiendo, en primer lugar, sistemas de procesamiento, centrándonos en los Lineales e Invariantes en el Espacio, para después definir la convolución discreta bidimensional. Seguidamente, expondremos dos transformaciones que se pueden aplicar a las imágenes para, finalmente, concluir con la sección 2.3, que, valiéndose de los modelos anteriormente expuestos, tratará de explicar qué es la compresión y cómo la utilizan diferentes formatos de imagen.

El siguiente capítulo, estará referido al arte que nos ocupa, en este caso el esteganografiado y el estegoanálisis, por lo que estará compuesto por las secciones 3.2 y 3.3 referidas a ambos, respectivamente. En esta sección se podrá encontrar la síntesis de trabajos previos, que serán reproducidos para ser evaluados y comparados con los métodos propuestos en el capítulo **??**.

El capítulo 4 estará compuesto por una sección referida a cada método de esteganografiado y estegoanálisis compuesto en este trabajo con una definición de cada uno de ellos, así como un desglose de su funcionamiento.

Seguidamente, se discutirá el rendimiento de nuestra propuesta en comparación con las otras vistas, y se mostrará una cuantificación de los resultados.

El último capítulo, se expondrán las conclusiones sacadas tras el desarrollo de la herramienta, buscando los puntos débiles del mismo y aquellas características que se podrían incluir en un futuro.

Capítulo 2

Modelado matemático de imágenes

2.1. Percepción de la imagen

Dada la naturaleza de este proyecto, nuestro primer cometido será la de estudiar cómo nuestros ojos reciben la información de imagen, puesto que así podremos determinar de qué forma podemos aplicar las transformaciones pertinentes para incrustar un mensaje, de la forma que el impacto visual sea el mínimo posible.

A lo largo de esta sección se mostrarán diversos experimentos que han dado lugar a la caracterización de nuestro sistema visual, así como la representación de la misma.

2.1.1. Definiciones formales

Antes de comenzar a explicar las diversas transformaciones que hay que aplicar a una imagen, resulta evidente que hay que explicar cómo manipular dicha imagen. Dada su naturaleza bidimensional, el álgebra de matrices resulta óptimo, pero, ¿qué representarán los valores que residan en una matriz de estas características? Para responder a esta cuestión, empezaremos definiendo algunas propiedades de la imagen:

- La **luminancia** o **intensidad** de un objeto con una distribución lumínica $I(x, y, \lambda)$ como [Jai89]:

$$L_v(x, y) = \int_0^\infty I(x, y, \lambda) V(\lambda) d\lambda \tag{2.1}$$

  Definiendo $V(\lambda)$ como la función de eficiencia de luminosidad relativa del SVH, caracterizada como la sigiente figura [Jai89]:

  Este parámetro definirá cuan claro u oscuro es un tono de gris de forma objetiva y, en el caso de contar con una solo valor para cada celda, será esta de forma predeterminada.

- El **brillo** (o luminancia percibida) es el primer componente del color que definiremos. La definición es la misma que para la luminancia, salvo que en este caso las luminancias adyacentes afectan a la percepción de la misma[1].

- **Tinte *Hue***. Característica referida al «color» propiamente dicho, es decir, si es más rojo, amarillo, morado, etc. El tinte es función de la longitud de onda [Jai89].

[1] Este tema se retomará en la siguiente sección.

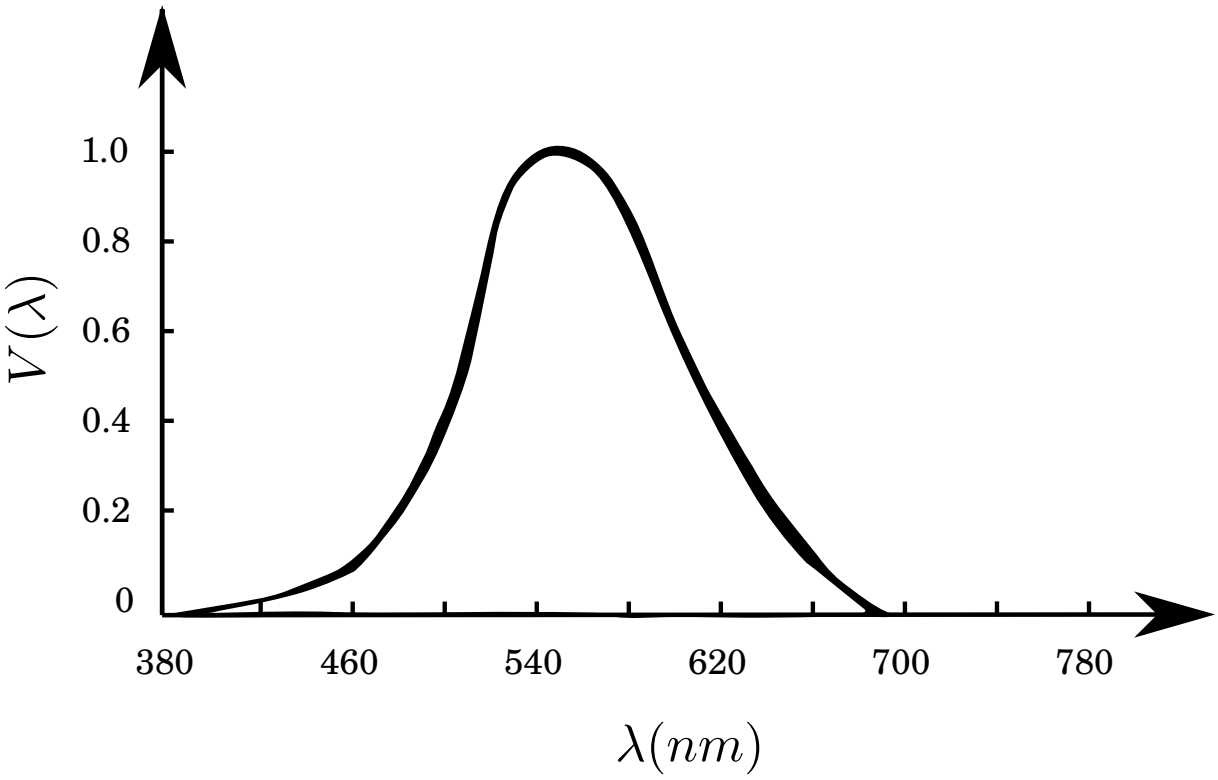


Figura 2.1: Función típica de eficiencia luminosa [Jai89].

- **Saturación *Saturation***. Define la «pureza», que aumenta conforme se le va añadiendo blanco[2] [Jai89]. Para una mejor comprensión de estos aspectos, la figura 2.2 representa al funcionamiento de lo anteriormente exupesto. El uso de la notación * utilizada para el brillo no debe ser confundida con el complejo conjugado. El uso del mismo es para mantenerla consistente con los símbolos utilizados comunmente para las coordenadas del color.

- La MTF *(Modulation Transfer Function)* es el módulo de la OTF *Optical Transfer Function* que se define como [Jai89]:

  $$OTF = \frac{H(\xi_1, \xi_2)}{H(0,0)}$$

  y por tanto

  $$MTF = |OTF| = \left| \frac{H(\xi_1, \xi_2)}{H(0,0)} \right|$$

- La **profundidad de bit** es la cantidad de bits con los que se representará cada píxel de una imagen, la profundidad predeterminada es `uint8`, que cuenta con 256 valores (entre 0 y 255), aunque en algunos casos tendremos que utilizar el tipo `double` que cuenta con una profundidad mucho mayor, del orden de 2^{64} valores, entre 0 y 1.

Una vez definidas las bases podemos proseguir a la siguiente sección.

[2] Cabe destacar que existen otras formas de representar el color, que se verán en la siguiente sección.

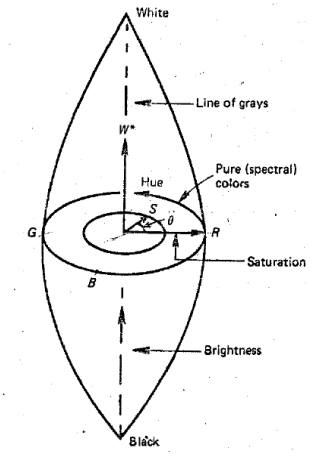


Figura 2.2: Representación del esquema HSV de color. El brillo, W^*, varía a lo largo del eje vertical, el tinte θ, sobre la circunferencia y la saturación S, con respecto al radio [Jai89].

2.1.2. Sistema visual humano

La presente subsección tendrá como objetivo caracterizar los atributos de la imagen y, en relación a éstos, una caracterización medible de los mismos, en relación con el SVH. Primeramente se procederá a explicar la composición de éste, para después comentar su funcionamiento.

El SVH concentra el sentido de la vista en los ojos, órganos que recopilan la información que utilizará nuestro cerebro para formar la imagen. Por una parte tenemos el cristalino, lente biconvexa que variará su convexidad en función de la distancia al objetivo para enfocar el mismo, la pupila, que regula la cantidad de luz (filtrando el color verde) que entra en la parte fotosensible, la retina, compuesta a su vez por dos tipos de células que se expondrán a continuación [Jai89]:

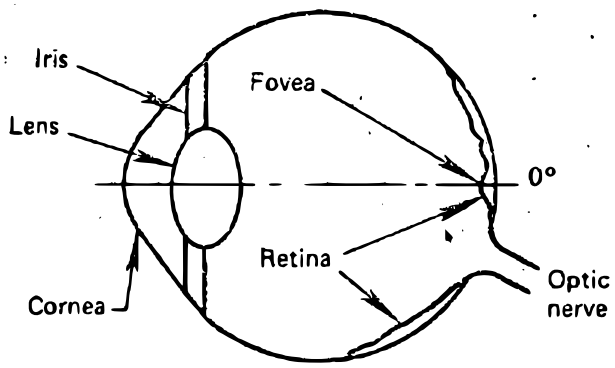


Figura 2.3: Sección de un ojo [Jai89].

Las células fotosensibles de que se compone la retina son dos:

- Los bastones, llamados así por su forma alargada y estrecha, son los encargados de la visión escotópica, es decir, la visión en condiciones de baja luminosidad. Su concentración es de unos 100 millones, siendo la parte más importante de la retina. Son capaces de distinguir cambios en la luz [Jai89].

- Los conos, por su parte, suponen un número de 6,5 millones, concentrados en la parte central de la retina, llamada *fóvea*. A diferencia de los bastones, su funcionamiento requiere de más luminosidad (visión fotópica). Son los encargados de diferenciar los colores, existiendo un tipo para cada color primario, rojo, azul y verde [Jai89].

Dada la naturaleza de este estudio, nos centraremos en la visión fotópica, puesto que asumiremos una buena iluminación a la hora de ver las imágenes.

2.1.3. Contraste simultáneo

El primer experimento que pasamos a definir es el de contraste simultáneo. Éste da cuenta de la diferencia entre luminancia y brillo. Como se puede ver en 2.4 los recuadros del centro, pese a ser de igual luminancia, aparentan ser tonos diferentes de gris por el fondo distinto [Jai89]

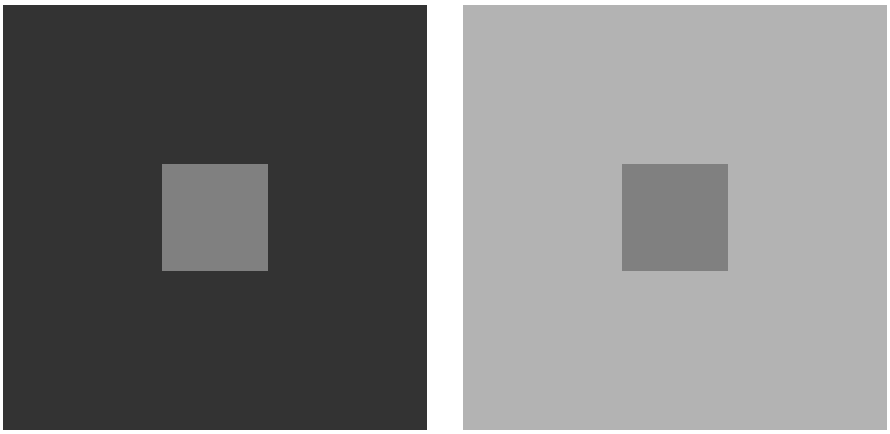


Figura 2.4: Contraste simultáneo.

Otro resultado que se encontro tras el estudio de este experimento, teniendo en cuenta la ley de Weber [Jai89]:

$$\frac{|I_s - I_0|}{I_0} = \text{constante} \tag{2.2}$$

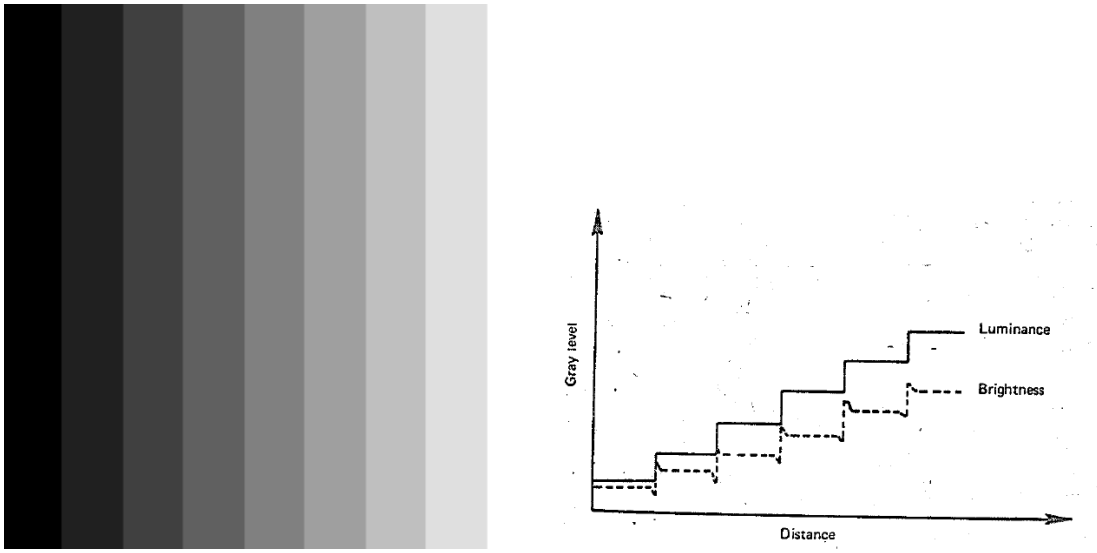
Que, si reescribimos $I_0 = I$, $I_s = I + \Delta I$, donde ΔI es pequeño para luminancias en las que la diferencia es tenuamente percibible (JND, *Just Noticeable Difference*), la anterior ecuación puede escribirse como [Jai89]:

$$\frac{\Delta I}{I} \simeq d(\log I) = \Delta c \quad \text{(constante)} \tag{2.3}$$

Tras la experimentación, se llegó a la conclusión de que $\Delta c = 0,2$, es decir, que al menos son necesarios 50 niveles de contraste en una escala de 0 a 1 [Jai89]

2.1.4. Experimento de las bandas de Mach

El segundo experimento que se procederá a explicar es el de **el efecto de las bandas de Mach**. Éste nos muestra que el brillo no es una función monótona de lumniancia. Tomando la barra de tonos de grises de la figura 2.5, en la que cada una de ellas se corresponde con un nivel constante de luminancia se puede observar que el brillo aparente no es uniforme a lo largo de la barra. Por el contrario, en cada transición, cada barra parece ser más brillante en el lado derecho y más oscura en el izquierdo [Jai89].



(a) Escala de grises. (b) Luminancia vs Brillo percibido [Jai89].

Figura 2.5: Efecto de las bandas de Mach [Jai89].

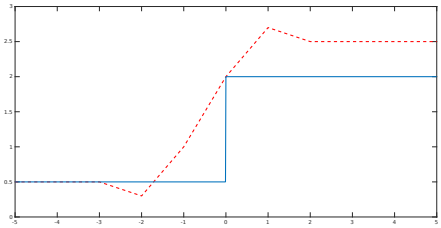
Utilizando el ejercicio 3.5 de [Jai89] obtenemos una simplificación de la respuesta del ojo, en la que se muestra el fenómeno de *inhibición lateral*. Los valores de la respuesta mostrada en la figura 2.6 representan la ponderación espacial relativa del contraste por los receptores (bastones y conos). Los lóbulos negativos indican que la señal neural, es decir, procesada tras la recepción, es inhibida por algunos de los receptores localizados lateralmente [Jai89].

Si bien con este efecto podemos medir la respuesta visual en coordenadas visuales, si aplicamos la transformada de Fourier, obtendremos la respuesta frequencial a través de la cuál podremos calcular la **MTF *Modulation Transfer Function*** es mediante la consideración de un *grating* sinusoidal de una variación del contraste (razón del máximo al mínimo de intensidad) y la frecuencia espacial. La observación de la figura 2.7(a) a una distancia de un metro dará como resultado la gráfica expuesta en 2.7(b) [Jai89].

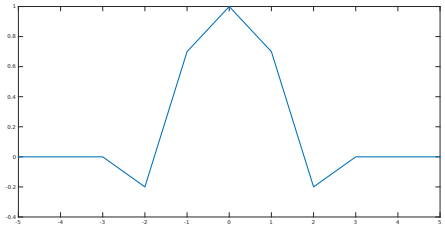
La anterior exposición de datos sirve para entender que, al menos en los métodos que se sirvan de la imagen en bruto o, al menos aquellos métodos que no purguen información, deberán esconder la información en aquellos lugares con frecuencias espaciales más altas.

2.1.5. Color

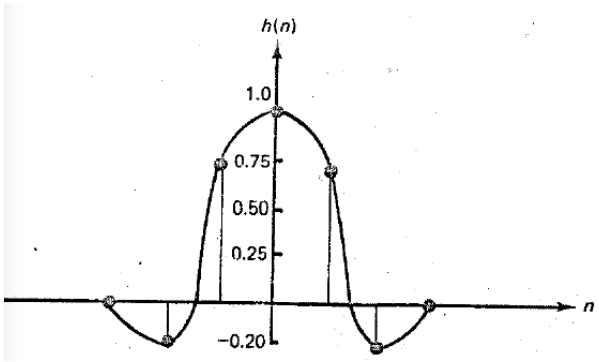
La representación del color está basada en la teoría clásica de Thomas Young (1802), quien demostró que cualquier color se puede reproducir mediante la mezcla en las proporciones adecuadas de los tres colores primarios. Tras esta deducción, se demostró que existen tres tipos diferentes de conos, cuyos



(a) Enunciado ejercicio 3.5.

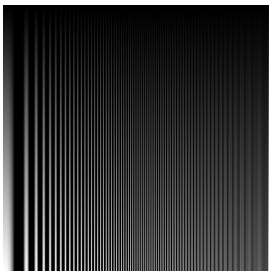


(b) Respuesta al impulso simplificada.

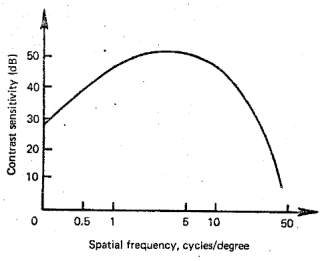


(c) Respuesta al impulso natural.

Figura 2.6: Respuesta al impulso del SVH.



(a) Escala de grises con grating sinusoidal.



(b) Gráfica del MTF [Jai89].

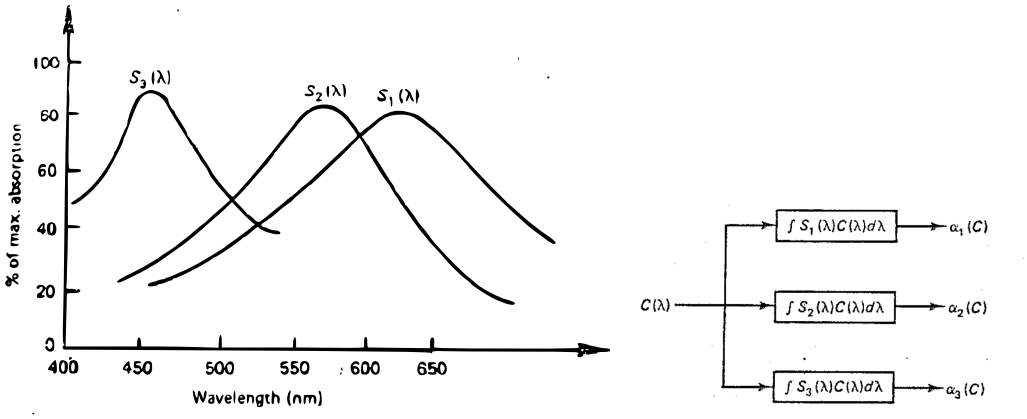
Figura 2.7: MTF del sistema visual humano.

espectros de absorción son $S_1(\lambda), S_2(\lambda)$ y $S_3(\lambda)$, donde $\lambda_{min} \leq \lambda \leq \lambda_{max}$, $\lambda_{min} \simeq 380$nm, $\lambda_{max} \simeq 780$ nm. Los picos de estas respuestas se encuentran en las regiones amarilla-verde, verde y azul, respectivamente, del espectro electromagnético, como se puede ver en la figura 2.8 [Jai89].

Según la teoría de los tres colores, la distribución espectral de energía de una luz «coloreada», $C(\lambda)$, producirá una sensación de color que puede ser descrita por las repuestas espectrales como [Jai89]:

$$\alpha_i(C) = \int_{\lambda_{min}}^{\lambda_{max}} S_i(\lambda)C(\lambda)d\lambda \qquad i = 1, 2, 3 \tag{2.4}$$

La ecuación (2.4) debe ser interpretada como una ecuación de la representación del color. Si $C_1(\lambda)$



(a) Típicas curvas de sensibilidad para S_1, S_2, S_3.

(b) Modelo de tres receptores para la representación del color.

Figura 2.8: (a) Típico espectro de absorción de los tres tipos de cono en la retina humana; (b) Modelo de tres colores para representación del color [Jai89].

y $C_2(\lambda)$ son dos distribuciones espectrales tales que producen respuestas $\alpha_i(C_1)$ y $\alpha_i(C_2)$ y se cumple:

$$\alpha_i(C_1) = \alpha_i(C_2), \qquad i = 1, 2, 3 \tag{2.5}$$

Entonces los colores C_1 y C_2 se perciben como idénticos. Cabe destacar que ambos pese a resultar idénticos, pueden tener distribuciones espectrales distintas [Jai89].

2.2. Modelo matemático

2.2.1. Sistemas lineales e invariantes en el espacio

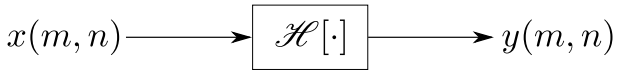


Figura 2.9: Un sistema.

Como ya se ha adelantado, para el estudio matemático de las imágenes utilizaremos el álgebra de matrices, una parte importante del Álgebra Lineal. Partiendo de esta premisa, resultará evidente la utilización de **Sistemas Lineales** para la manipulación de las mismas. En primer lugar, definiremos un sistema $\mathscr{H}$ como un *artefacto* que modifica una entrada $x(m, n)$ dejando a $y(m, n)$ como salida, fruto del procesamiento del mismo. Esto es [Jai89]:

$$y(m, n) = \mathscr{H}\left[x(m, n)\right] \tag{2.6}$$

El sistema será lineal si y sólo si cualquier combinación lineal de dos entradas $x_1(m, n)$ y $x_2(m, n)$ produce la misma combinación de las salidas $y_1(m, n)$ y $y_2(m, n)$, e.g. si utilizamos dos constantes arbitrarias a_1 y a_2 [Jai89]

$$\begin{aligned} \mathscr{H}[a_1x_1(m, n) + a_2x_2(m, n)] &= a_1\mathscr{H}[x_1(m, n)] + a_2\mathscr{H}[x_2(m, n)] \\ &= a_1y_1(m, n) + a_2y_2(m, n) \end{aligned} \tag{2.7}$$

superposición lineal. Cuando la entrada es una delta de Kronecker bidimensional situada en (m', n'), la salida en (m, n) se define como [Jai89]:

$$h(m, n; m', n') \triangleq \mathscr{H}[\delta(m - m', n - n')] \tag{2.8}$$

a lo que llamaremos la **respuesta al impulso** del sistema. Para un sistema de imagen, es el valor en el plano de salida debido a un punto fuente localizado en (m', n') en el plano de entrada. En nuestra notación, el punto y coma (;) ha sido utilizado para distinguir los pares de coordenadas de entrada y salida [Jai89].

La respuesta al impulso se denomina **función de dispersión** (PSF) cuando entrada y salida representan una cantidad positiva tal y como la intensidad de la luz. Si se utiliza la denominación genérica, respuesta al impulso, permite incluir cantidades negativas e incluso complejas. La *región de soporte* para la respuesta al impulso es la región cerrada más pequeña en el plano m, n fuera del cual la respuesta al impulso es cero. Se dice de un sistema que es FIR, de **respuesta al impulso finita** o IIR, de **respuesta al impulso infinita** si la región de soporte es finita o infinita, respectivamente [Jai89].

La salida de cualquier sistema lineal puede obtenerse de su respuesta al impulso y la entrada aplicando la regla de superposición de (2.7) para la representación de

$$\left.\begin{aligned} x(m, n) &= \sum_{m', n'=-\infty}^{\infty}\sum x(m', n')\delta(m - m', n - n') \\ \sum_{m,n=-\infty}^{\infty}\sum \delta(m, n) &= 1 \end{aligned}\right\}$$

como sigue:

$$\begin{aligned} y(m, n) &= \mathscr{H}[x(m, n] \\ &= \mathscr{H}\left[\sum_{m'}\sum_{n'} x(m', n')\delta(m - m', n - n')\right] \\ &= \sum_{m'}\sum_{n'} x(m', n')\mathscr{H}[\delta(m - m', n - n')] \\ \Rightarrow \quad y(m, n) &= \sum_{m'}\sum_{n'} x(m', n')h(m, n; m', n') \end{aligned} \tag{2.9}$$

Si la traslación de la entrada provoca la translación de la salida, el sistema se denomina **invariante en el espacio**. Continuando con la definición de (2.9), si el impulso se da en el origen, tendremos

$$\mathscr{H}[\delta(m, n)] = h(m, n; 0, 0)$$

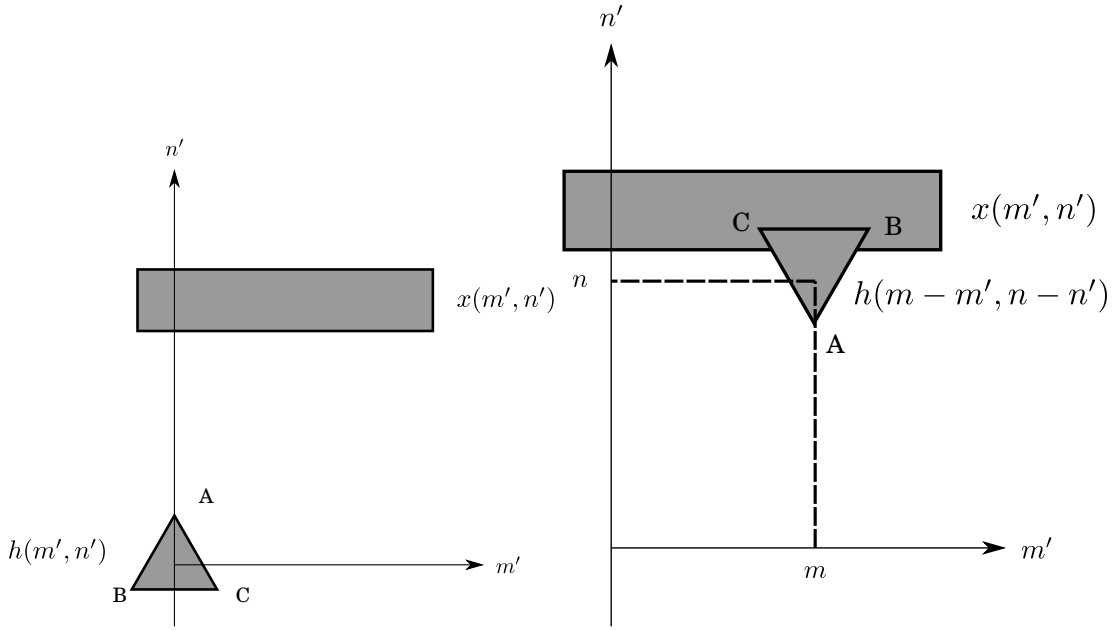
por lo que, si es invariante en el espacio, debe cumplirse [Jai89]:

$$\begin{aligned} h(m, n; m', n') &\triangleq \mathscr{H}[\delta(m - m', n - n')] \\ &= h(m - m', n - n'; 0, 0) \\ \Rightarrow \quad h(m, n; m', n') &= h(m - m', n - n') \end{aligned} \tag{2.10}$$

i.e., la respuesta al impulso es función únicamente de dos variables de desplazamiento. Esto significa que la forma de la respuesta al impulso no cambia cómo se mueve el impulso sobre el plano m, n. Un sistema se denomina *variante en el espacio* cuando no se cumple (2.10)

En el caso de cumplirse, la salida se convierte en [Jai89]:

$$y(m, n) = \sum_{m', n'=-\infty}^{\infty}\sum h(m - m', n - n')x(m', n') \tag{2.11}$$



(a) Respuesta al impulso.

(b) La salida en el punto (m,n) es la suma de los productos en el área de superposición.

Figura 2.10: Convolución discreta en dos dimensiones.

denominada **convolución** de la entrada con la respuesta al impulso. En la figura 2.10 se muestra una interpretación gráfica de esta operación. El array de la respuesta al impulso es rotado 180º con respecto al origen y girado entonces por (m, n) y superpuesto sobre el array $x(m', n')$. La suma del producto de los arrays $\{x(\cdot, \cdot)\}$ y $\{h(\cdot, \cdot)\}$ en las regiones superpuestas devuelve el resultado en (m, n). Se utiliza el símbolo $*$ para denotar esta operación. Esto es [Jai89]:

$$\begin{aligned} g(x, y) &= h(x, y) * f(x, y) \\ &\triangleq \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - x', y - y') f(x', y') dx', dy' \\ y(m, n) &= h(m, n) * x(m, n) \\ &\triangleq \sum_{m', n' = -\infty}^{\infty} \sum h(m - m', n - n') x(m', n') \end{aligned} \tag{2.12}$$

2.2.2. Transformada de Fourier

La tranformada de Fourier, típicamente presentada en funciones temporales, se encarga de trasladar dichas funciones del plano temporal al plano frecuencial. Este cambio se hace en base a la **Serie de Fourier**, que convierte cualquier señal (periódica y no periódica) a la superposición de señales sinusoidales. Las ecuaciones de síntesis y análisis son, respectivamente:

$$X(\omega) \triangleq \mathscr{F}[x(t)] \triangleq \int_{-\infty}^{\infty} x(t) \exp(-j2\pi\omega t) dt \tag{2.13}$$

$$x(t) \triangleq \mathscr{F}^{-1}[X(\omega)] = \int_{-\infty}^{\infty} F(\omega) \exp(j2\pi\omega t) d\omega \tag{2.14}$$

A continuación se definirán las ecuaciones de análisis y síntesis para el caso bidimensional. Dado que, para las señales temporales se utilizaba una notación distinta (x para las funciones para no ser confundida con la f de la frecuencia temporal en Hertzios utilizando la variable t para el tiempo y ω para la velocidad angular), en este caso al tratarse de variables espaciales y frecuencias de la misma índole, adoptaremos la notación oportuna [Jai89]:

$$F(\xi_1, \xi_2) = \int_{-\infty}^{\infty} f(x, y) \exp[-j2\pi(x\xi_1 + y\xi_2)]dxdy \tag{2.15}$$

$$f(x, y) = \int_{-\infty}^{\infty} F(\xi_1, \xi_2) \exp[j2\pi(x\xi_1 + y\xi_2)]d\xi_1 d\xi_2 \tag{2.16}$$

Propiedades de la transformada de Fourier

A continuación se exponen las diferentes propiedades de esta transformada [Jai89]:

1. **Frecuencia espacial**. Si $f(x, y)$ denota la luminancia y x e y las coordenadas espaciales, entonces ξ_1 y ξ_2 son las frecuencias espaciales que representan los cambios de luminancia con respecto a las distancias espaciales. Las unidades ξ_1 y ξ_2 son las recíprocas de x e y, respectivamente. Algunas veces, las coordenadas x e y son normalizadas sobre la distancia de visionado de la imagen $f(x, y)$. Entonces, las unidades ξ_1 y ξ_2 son ciclos por grado (del ángulo de visión).

2. **Unicidad**. Para funciones continuas, $f(x, y)$ y $F(\xi_1, \xi_2)$ son únicas una respecto de la otra. No hay pérdida de información si, en vez de preservar la imagen, su transformada de Fourier se preserva. Este hecho se utiliza en una técnica de compresión de datos denominada *codificación en transformada*.

3. **Separabilidad**. Por definición, el núcleo de la transformada es separable, por lo que se puede escrbir como una transformación separable en x y en y, esto es:

$$F(\xi_1, \xi_2) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x, y) \exp(-j2\pi x\xi_1)dx \right] \exp(-j2\pi y\xi_2)dy$$

   Esto significa que la transformada bidimensional puede interpretarse como una sucesión de dos transformaciones en una dimensión a lo largo de las coordenadas espaciales.

4. **Respuesta frecuencial y autofunciones de sistemas invariantes en el espacio**. Una autofunción de un sistema se define como su función de entrada que se reproduce a la salida sólo pudiendo cambiar en amplitud. Una propiedad fundamentalde un sistema linea e invariante en el esapacio es que las autofunciones son dadas por la exponencial compleja $\exp[j2\pi(\xi_1 x + \xi_2 y)]$. Tal como se muestra en la Figura 2.11, para cualquier (ξ_1, ξ_2) fijados, la salida del sistema LSI debería ser

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - x', y - y') \exp[j2\pi(\xi_1 x + \xi_2 y)]dx'dy'$$

   Donde si hacemos el cambio de variables $\tilde{x} = x - x', \tilde{y} = y - y'$ y simplificando el resultado, obtenemos

$$g(x, y) = H(\xi_1, \xi_2) \exp[j2\pi(\xi_1 x + \xi_2 y)]$$

   La función $H(\xi_1, \xi_2)$, que es la transformada de Fourier de la respuesta al impulso, también denominada **respuesta frecuencial** del sistema. Representa la amplitud (compleja) de la respuesta del sistema en la frecuencia espacial (ξ_1, ξ_2).

5. **Teorema de la convolución**. La transformada de Fourier de la convolución de dos funciones es el producto de sus transformadas de Fourier i.e.:

$$g(x,y) = h(x,y) * f(x,y) \Leftrightarrow G(\xi_1,\xi_2) = H(\xi_1,\xi_2)F(\xi_1,\xi_2) \tag{2.17}$$

Este teorema sugiere que la convolución de dos funciones puede ser evaluada mediante la transformación inversa del producto de la transformada de Fourier de ambas funciones. La versión discreta de este teorema conduce al algoritmo FFT (Fast Fourier Transform).

Por otra parte, la transformada de Fourier del producto de dos funciones es la convolución de la transformada de cada una de ellas.

El resultado del teorema de la convolución, también puede extenderse a la *correlación espacial* entre dos funciones reales $h(x,y)$ y $f(x,y)$, que es definido como

$$c(x,y) = h(x,y) \star f(x,y) \triangleq \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x',y')f(x+x',y+y')dx'dy' \tag{2.18}$$

Un cambio de variables muestra que $c(x,y)$ es también la convolución $h(-x,-y)*f(x,y)$, expresada como

$$C(\xi_1,\xi_2) = H(-\xi_1,-\xi_2)F(\xi_1,\xi_2) \tag{2.19}$$

6. **Presevación del producto interno**. Otra propiedad importante de la transformada de Fourier es que el production interno de dos funciones es igual al productio interno de sus transformadas de Fourier, i.e.:

$$I \triangleq \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)h*(x,y)dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\xi_1,\xi_2)H*(\xi_1,\xi_2)d\xi_1 d\xi_2 \tag{2.20}$$

Si igualamos $h = f$, obtenemos la ampliamente conocida **formula de conservación de la energía de Parseval**:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2 dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |F(\xi_1,\xi_2)|^2 d\xi_1 d\xi_2 \tag{2.21}$$

7. **Transformada Hankel**. La transformada de Fourier de una función simétrica circular es también simétrica circular y viene dada por la llamada *Transformada de Hankel*.

2.2.3. Resultados en teoría de matrices

Vectores y matrices

Habitualmente secuencias de una y dos dimensiones son representadas por vectores y matrices, respectivamente. Una columna vector u que contiene N elementos se denota como [Jai89].

$$\boldsymbol{u} = \{u(n)\} = \begin{bmatrix} u(1) \\ u(2) \\ \vdots \\ u(N) \end{bmatrix} \tag{2.22}$$

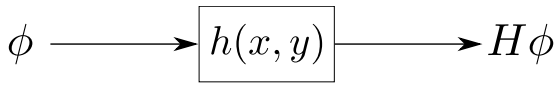


Figura 2.11: Autofunción.

El n-ésimo elemento del vector $\boldsymbol{u}$ se denota como $u(n)$, u_n o $[\boldsymbol{u}]_n$. Salvo que se especifique lo contrario, todos los vectores se considerarán columna de manera predeterminada. Un vector columna de tamaño N también se denomina como de dimensiones $N \times 1$ [Jai89].

Una matriz $\mathbf{A}$ de dimensiones $M \times N$ tiene M filas y N columnas y se define como: [Jai89].

$$\boldsymbol{A} \triangleq \{a(m,n)\} = \begin{bmatrix} a(1,1) & a(1,2) & \cdots & a(1,N) \\ a(2,1) & & & \\ \vdots & \vdots & \vdots & \\ a(M,1) & a(M,2) & \cdots & a(M,N) \end{bmatrix} \tag{2.23}$$

El elemento en la fila m-ésima y en la columna n-ésima de la matriz $\mathbf{A}$ se denota como $[\boldsymbol{A}]_{m,n} \triangleq a(m,n) \triangleq a_{m,n}$. La columna n-ésima de $\mathbf{A}$ se denota como a_n, cuyo elemento m-ésimo se escribirá como $a_n(m) = a(m,n)$. Cuando el índice inicial de la matriz no sea $(1,1)$, deberá indicarse de forma explícita [Jai89].

$$\boldsymbol{A} = \{a(m,n) \qquad 0 \leq m,n \leq N-1\}$$

representa una matriz $N \times N$ con un índice inicial (0,0).

En dos dimensiones, es usualmente útil visualizar una imagen como una matriz. La representación matricial es simplemente una rotación de 90º en sentido horario de la representación bidimensional Cartesiana convencional:

Ordenación de filas y columnas

Algunas veces es necesario escribir una matriz en forma de vector, por ejemplo, cuando se almacena una imagen en un disco. Si tenemos

$$\boldsymbol{x} \triangleq \mathscr{O}\{x(m,n)\}$$

será una ordenación uno a uno de los elementos del array $\{x(m,n)\}$ en el vector $\boldsymbol{x}$. Para una matriz $M \times N$, un mapeo que se utiliza habitualmente es el denominado *lexicográfico* o *ordenamiento de diccionario*. Este es un *vector ordenado por filas* y se define como:

$$\begin{aligned} \boldsymbol{x}^T &= [x(1,1)x(1,2)\ldots x(1,N)x(2,1)\ldots x(2,N)\ldots x(M,1)\ldots x(M,N)]^T \\ &\triangleq \mathscr{O}_r\{x(m,n)\} \end{aligned} \tag{2.24}$$

Teniendo en cuenta que, $\boldsymbol{x}^T$ es el vector fila obtenido apilando cada fila a la derecha de la fila de $\boldsymbol{X}$. Otro mapeo úil es el apilamiento de columna por columna, cuyo resultado es el *vector ordenado por columnas* como:

$$\begin{aligned} \boldsymbol{x}^T &= [x(1,1)x(2,1)\ldots x(M,1)x(1,2)\ldots x(M,2)\ldots x(M,N)]^T \\ &= \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix} \triangleq \mathscr{O}_c\{x(m,n)\} \end{aligned} \tag{2.25}$$

Reglas de transposición y conjugación

1. $\boldsymbol{A}^{*T} = [\boldsymbol{A}^T]^*$
2. $[\boldsymbol{AB}] = \boldsymbol{B}^T\boldsymbol{A}^T$

3. $[\boldsymbol{A}^{-1}]^T = [\boldsymbol{A}^T]^{-1}$
4. $[\boldsymbol{AB}]^* = \boldsymbol{A}^*\boldsymbol{B}^*$

En la literatura de teoría de matrices, habitualmente la **transpuesta conjugada** se denota como $\boldsymbol{A}^*$, pero, en nuestro caso, deberemos distinguir entre $\boldsymbol{A}, \boldsymbol{A}^*, \boldsymbol{A}^T$ y $\boldsymbol{A}^{*T}$, por lo que denotaremos la transpuesta conjugada de la última de las maneras expuestas.

Matrices de Toeplitz y circulante

Una **Matriz de Toeplitz** $\boldsymbol{T}$ es una matriz que tiene elementos constantes sobre la diagonal principal y las subdiagonales. Esto significa que los elementos $t(m,n)$ dependen únicamente de la diferencia $m-n$, i.e., $t(m,n) = t_{m-n}$. Si bien una matriz de Toepliz $N \times N$ tendría la forma:

$$\boldsymbol{T} = \begin{bmatrix} t_0 & t_{-1} & & \cdots & t_{-N+1} \\ t_1 & t_0 & t_{-1} & & t_{-N+2} \\ t_2 & \ddots & & & \vdots \\ \vdots & & t_1 & t_0 & t_{-1} \\ t_{N-1} & \cdots & t_2 & t_1 & t_0 \end{bmatrix} \tag{2.26}$$

y está completamente definida por los $(2N-1)$ elementos $\{t_k, -N+1 \leq k \leq N-1\}$. Las matrices de Toeplitz describen las transformaciones de entrada-salida de sistemas LSI unidimensionales y matrices de correlación de secuencias estacionarias.

Una matriz $\boldsymbol{C}$ será **circulante** si cada una de sus filas o columnas es un desplazamiento circular de la fila o columna previa:

$$\boldsymbol{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & c_1 & \cdots & c_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & & c_{N-1} & c_0 & c_1 \\ c_1 & c_2 & \cdots & c_{N-1} & c_0 \end{bmatrix} \tag{2.27}$$

Las matrices circulantes son un caso particular de las de Toeplitz y

$$c(m,n) = c((m-n) \mod N) \tag{2.28}$$

Las matrices circulantes describen el comportamiento de entrada-salida de sistemas periódicos lineales unidimensionales y la matrices de correlación de secuencias periódicas. Estas matrices tienen un especial significado, dado que con ellas podremos simplificar las operaciones de la convolución y convolución circular:

Convolución lineal operando con una matriz de Toeplitz

La salida de un sistema LSI con respuesta al impulso $h(n) = n,\ -1 \leq n \leq 1$ y con entrada $x(n)$, que es cero fuera del intervalo $[0,4]$, viene dada por la convolución

$$y(n) = h(n) * x(n) = \sum_{k=0}^{4} h(n-k)x(k)$$

Como $y(n)$ vale cero fuera del intervalo $-1 \leq n \leq 5$, esta operación se puede escribir como una operación de una Matriz Toepliz 7×5 operando sobre un vector 5×1, de la forma:

$$\begin{bmatrix} y(-1) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \end{bmatrix}$$

Convolución circular como operación de una matriz circulante

La convolución de dos secuencias periódicas también será periódica y se puede representar como

$$y(n) = \sum_{k=0}^{N-1} h(n-k)x(k), \quad 0 \leq n \leq N-1$$

donde $h(-n) = h(N-n)$ y N es el periodo. Por ejemplo, si $N = 4$ y $h(n) = n + 3 \mod 4$. En notación vectorial, esto ofrece:

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 0 & 3 & 2 & 1 \\ 1 & 0 & 3 & 2 \\ 2 & 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

Por tanto, la transformación entrada-salida de una convolución circular viene descrita por una matriz circulante.

Matrices ortogonales y unitarias

Una **matriz ortogonal** es una tal que su inversa es igual a su transpuesta, es decir:

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^T \tag{2.29}$$

o

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{I} \tag{2.30}$$

Una matriz se dice unitaria si su inversa es igual a su traspuesta conjugada:

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{*T}$$

o

$$\boldsymbol{A}\boldsymbol{A}^{*T} = \boldsymbol{A}^{*T}\boldsymbol{A} = \boldsymbol{I} \tag{2.31}$$

Una matriz real y ortogonal también es unitaria, pero una matriz unitaria no necesita ser ortogonal. Las definiciones precedentes, implican que *las columnas (o filas) de una matriz unitaria $N \times N$ son ortogonales y forman un conjunto completo de vectores base en un espacio vectorial N-dimensional.*

Definición positiva y formas cuadráticas

Una matriz $\boldsymbol{A}$ hermítica positiva $N \times N$, se denomina **definida positiva** o **semidefinida positiva** si la forma cuadrática

$$Q \triangleq \boldsymbol{x}^{*T}\boldsymbol{A}\boldsymbol{x}, \quad \forall x \neq 0 \tag{2.32}$$

es positiva (> 0) o no negativa (≥ 0) respectivamente. De forma similar, $\boldsymbol{A}$ es *negativa definida* o *negativa semidefinida* si $Q < 0$ o $Q \leq 0$, respectivamente. Una matriz que no satisfaga cualquiera de las condiciones anteriormente expuestas es *indefinida*

es positiva (> 0) o no negativa (≥ 0) respectivamente. De forma similar, $\boldsymbol{A}$ es *negativa definida* o *negativa semidefinida* si $Q < 0$ o $Q \leq 0$, respectivamente. Una matriz que no satisfaga cualquiera de las condiciones anteriormente expuestas es *indefinida*.

Si $\boldsymbol{A}$ es definida simétrica positiva (no negativa) entonces todos sus autovalores $\{\lambda_k\}$ son positivas (no negativas) y el determinante de $\boldsymbol{A}$ satisface la inecuación:

$$|\boldsymbol{A}| = \prod_{k=1}^{N} \leq \prod_{k=1}^{N} a(k,k) \tag{2.33}$$

Formas diagonales

Para cualquier matriz Hermítica $\boldsymbol{R}$ existe una matriz unitaria $\boldsymbol{\Phi}$ tal que:

$$\boldsymbol{\Phi}^{*T}\boldsymbol{R}\boldsymbol{\Phi} = \boldsymbol{\Lambda} \tag{2.34}$$

donde $\boldsymbol{\Lambda}$ es una matriz diagonal que contiene los autovalores de $\boldsymbol{R}$. Una forma alternativa a la ecuación anterior es

$$\boldsymbol{R}\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Lambda} \tag{2.35}$$

cuyo conjunto de ecuaciones de autovalores es:

$$\boldsymbol{R}\boldsymbol{\phi}_k = \lambda_k \phi_k \quad k = 1, \ldots, N \tag{2.36}$$

donde $\{\lambda_k\}$ y $\{\phi_k\}$ son autovalores y autovectores, respectivamente, de R. Para matrices Hermíticas, los autovectores correspondientes a distintos autovalores son ortogonales. Para autovalores repetidos, sus autovectores de un subespacio qeu pueden ser ortogonalizados hasta completar un conjunto de autovectores ortogonales. La normalización de estos autovectores conduce a un conjunto ortonormal, i.e., la matriz unitaria $\boldsymbol{\Phi}$, cuyas columnas son autovectores. La matriz $\boldsymbol{\Phi}$ también es llamada la *automatriz* de $\boldsymbol{R}$

2.2.4. Transformada discreta del coseno

La DCT es una transformada útil a la hora de la compresión de imágenes, dado que agrupa la energía en los primeros elementos, haciendo prácticamente irrelevantes el resto, permitiendo comprimir la imagen sin perder calidad. Esta faceta se explorará más en profundidad en la sección referida a la compresión.

La definición que seguirá está tomada de [Jai89], el cual utiliza una normalización para una matriz de dimensiones $N \times N$. La matriz de transformación $\mathbf{C} = \{c(k,n)\}$ se define como:

$$c(k,n) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 0, 0 \leq n \leq N-1 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{\pi 2n+1)k}{2N}\right], & 1 \leq k \leq N-1, 0 \leq n \leq N-1 \end{cases} \tag{2.37}$$

La DCT unidimensional de una secuencia $\{u(n), 0 \leq n \leq N-1\}$ se define como

$$v(k) = \alpha(k) \sum_{n=0}^{N-1} u(n) \cos\left[\frac{\pi(2n+1)k}{2N}\right], \qquad 0 \leq k \leq N-1 \tag{2.38}$$

donde

$$\alpha(0) \triangleq \sqrt{\tfrac{1}{N}}, \quad \alpha(k) \triangleq \sqrt{\tfrac{2}{N}} \quad \text{para} \quad 1 \leq k \leq N-1 \tag{2.39}$$

La transformación inversa se obtendrá mediante

$$u(n) = \sum_{k=0}^{N-1} \alpha(k) v(k) \cos\left[\frac{\pi(2n+1)k}{2N}\right], \qquad 0 \leq n \leq N-1 \tag{2.40}$$

Como se puede ver en la figura 2.12, se comprueba la utilidad anteriormente enunciada de esta transformación, la agrupación en los primeros miembros de la energía de la señal.

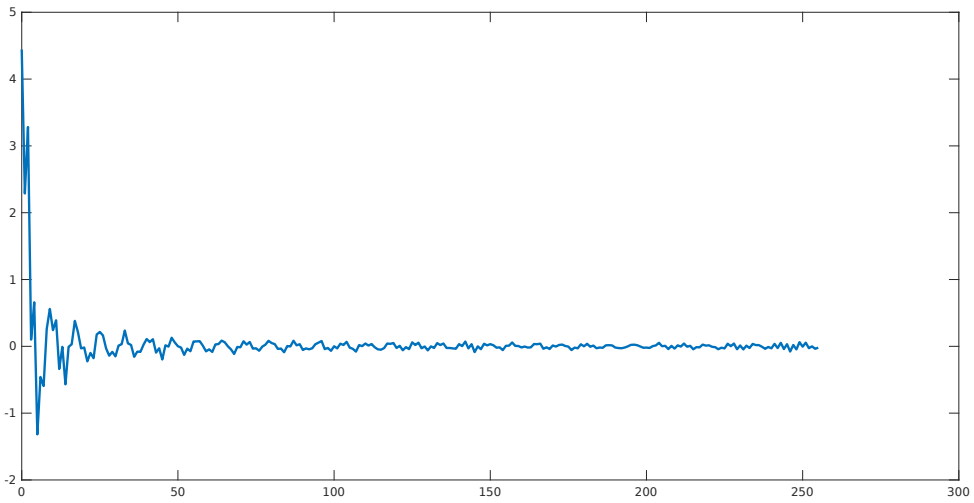


Figura 2.12: DCT de una línea de una imagen.

La transformada bidimensional, tal y como define [Jai89], se resuelve mediante la sustitución de $\boldsymbol{A}$ y $\boldsymbol{A^*}$ por C en las siguientes ecuaciones referentes a transformaciones unitarias separables.

$$v(k,l) = \sum_{n,m=0}^{N-1} \sum a(k,m) u(m,n) a(l,n) \leftrightarrow \boldsymbol{V = AUA^T} \tag{2.41}$$

$$u(m,n) = \sum_{k,i=0}^{N-1} \sum a^*(k,m) v(k,l) a^*(l,n) \leftrightarrow \boldsymbol{U = A^{*T} V A^*} \tag{2.42}$$

Propiedades de la DCT

1. La DCT es real y ortogonal, es decir,

$$C = C^* \Rightarrow C^{-1} = C^T \tag{2.43}$$

2. La transformada del coseno no es la parte real de la DFT *Discrete Fourier Transform*. Esto se puede ver mediante la inspección de **C** y de la matriz de la DFT **F**, explicada en la sección 4 del quinto capítulo de [Jai89], aunque está relacionada con la extensión simétrica de la DFT.

3. La transformada del coseno es una transformada rápida. Teniendo un vector de N elementos, ésta puede ser calculada en $O(N \log_2 N)$ operaciones mediante una FFT *Fast Fourier Transform* de N puntos, mediante una reordenación de los elementos pares e impares. Para una explicación más extensa, nos volvemos a remitir a [Jai89].

4. La DCT comporta una excelente aglomeración de energía para datos muy correlados. Esto es debido a las siguientes propiedades.

5. Los vectores base de la DCT, es decir, las filas de **C**, son autovectores de la matriz simétrica tridiagonal $\boldsymbol{Q_c}$, definida como:

$$\boldsymbol{Q_c} = \begin{bmatrix} 1-\alpha & -\alpha & & \mathbf{0} \\ -\alpha & 1 & & \\ & & \ddots & -\alpha \\ \mathbf{0} & & -\alpha & 1-\alpha \end{bmatrix} \tag{2.44}$$

6. La transformada $N \times N$ es muy cercana a la transformada KL de una secuencia Markov de primer orden de longitud N cuya matriz de covarianza dada por

$$\boldsymbol{R} = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & & & & \vdots \\ \rho^2 & & \ddots & & \rho^2 \\ \vdots & & & & \rho \\ \rho^{N-1} & \dots & \rho^2 & \rho & 1 \end{bmatrix} \tag{2.45}$$

cuando el parámetro de correlación ρ es próximo a 1. La razón es que $\boldsymbol{R}^{-1}$ es una matriz simétrica tridiagonal, la que, para un escalar $\beta^2 \triangleq \frac{1-\rho^2}{1+\rho^2}$ y $\alpha \triangleq \frac{\rho}{1+\rho^2}$ satisface la relación:

$$\beta^2 \boldsymbol{R}^{-1} = \begin{bmatrix} 1-\rho\alpha & -\alpha & & & \mathbf{0} \\ -\alpha & 1 & & & \\ & & \ddots & & \\ & & & 1 & -\alpha \\ \mathbf{0} & & & -\alpha & 1-\rho\alpha \end{bmatrix} \tag{2.46}$$

Esto da la aproximación

$$\beta^2 R^{-1} \equiv \boldsymbol{Q}_c \qquad \text{para} \qquad \rho \simeq 1 \tag{2.47}$$

Dado que los autovectores de $\mathbf{R}$ y los autovectores de $\boldsymbol{Q}_c$, esto es, la transformada del coseno, estarán muy cerca. Esta propiedad junto con que es una transformada rápida, ha hecho de la misma una sustituta útil para la transfomada KL de secuencias de Markov de primer orden de alta correlación.

El siguiente método que se presentará será el SVD, cuya documentación también estará a cargo de [Jai89].

2.2.5. Descomposición en valores singulares: SVD

Para la siguiente teoría de transformación, cosideraremos una imagen $N \times M$ que denominaremos **U**, que será un vector en un espacio vectorial NM-dimensional. Sin embargo, es posible representar cualquier imagen en un subespacio r-dimensional donde r es el rango de la matriz **U**.

Asumamos que la imagen es real y $M \leq N$. Las matrices UU^T y U^TU son no negativos, simétrocos y tienen autovalores idénticos, $\{\lambda_m\}$. Teniendo en cuenta que $M \leq N$, hay al menos $r \leq M$ autovectores. Es posible encontrar r autovectores ortogonales $M \times 1$ $\{\phi_m\}$ de U^TU y r autovectores ortogonales $\{\psi_m\}$ de UU^T, esto es:

$$U^TU\phi_m = \lambda_m\phi_m, \qquad m = 1, \ldots, r \tag{2.48}$$

$$UU^T\psi_m = \lambda_m\Psi_m, \qquad m = 1, \ldots, r \tag{2.49}$$

La matriz U se representa como

$$U = \Psi\Lambda^{1/2}\phi^T \tag{2.50}$$

$$= \sum_{m=1}^{r} \sqrt{\lambda_m}\psi_m\phi_m^T \tag{2.51}$$

donde Ψ y Φ son matrices $N \times r$ y $M \times r$ cuyas m-ésimas columnas son los vectores ψ_m y ϕ_m, respectivamente, y $\Lambda^{1/2}$ es una matriz diagonal $r \times r$ definida como:

$$\Lambda^{1/2} = \begin{bmatrix} \sqrt{\lambda_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sqrt{\lambda_r} \end{bmatrix} \tag{2.52}$$

La ecuacion (2.51) se denomina *representación espectral*, *expansión del producto matricial externo* o la ***Descomposición en Valores Singulares (SVD)*** de U, puesto que los autovalores no nulos, λ_m, también son llamados valores singulares. Si $r \ll M$, la imagen que contenedora de NM muestras puede ser representada por $(M + N)r$ muestras de los vectores $\{\lambda_m^{1/4}\psi_m, \lambda_m^{1/4}\phi_m; m = 1, \ldots ..., r\}$.

Como Ψ y Φ tienen columnas ortogonales, según (2.53) la *transformada SVD* de la imagen U se define como:

$$\Lambda^{1/2} = \Psi^TU\Phi \tag{2.53}$$

la cual es una transformada separable que diagonaliza la imagen dada.

Propiedades de la transformada SVD

A continuación pasaremos a definir las propiedades de esta transformada, apoyándonos una vez más en [Jai89].

1. Si $\phi_m, m = 1, \ldots, r$ son conocidos, los autovectores ψ_m pueden ser determinados como

$$\psi_m \triangleq \frac{1}{\sqrt{\lambda_m}}U\phi_m, \qquad m = 1, \ldots, r \tag{2.54}$$

Se puede demostrar que ψ_m son autovectores ortonormales de $\boldsymbol{UU^T}$ si ϕ_m son autovectores ortonormales de $\boldsymbol{U^TU}$.

2. La transformada SVD definida por (2.53) no es una transformación unitaria. Este hecho está determinado porque tanto $\boldsymbol{\Psi}$ como $\boldsymbol{\Phi}$ son matrices rectangulares. en cualquier caso, se pueden incluir en $\boldsymbol{\Phi}$ o $\boldsymbol{\Psi}$ autovectores ortogonales ϕ_m y ψ_m que satisfagan

$$\boldsymbol{U}\phi_m = 0, \quad m = r+1, \ldots, M$$

y

$$\boldsymbol{U^T}\psi_m = 0, \quad m = r+1, \ldots, N$$

tales que esas matrices sean unitarias y la *transformada SVD unitaria sea:*

$$\begin{bmatrix} \boldsymbol{\Lambda^{1/2}} \\ 0 \end{bmatrix} = \boldsymbol{\Psi^T U \Phi} \tag{2.55}$$

3. La imagen $\boldsymbol{U_k}$ generada por la suma parcial

$$\boldsymbol{U_k} \triangleq \sum_{m=1}^{k} \sqrt{\lambda_m}\psi_m\phi_m^T, \qquad k \leq r \tag{2.56}$$

es la mejor aproximación de mínimos cuadráticos de rango k para $\mathbf{U}$ si λ_m está en un orden decreciente de magnitud[3]. Para cualquier $k \leq r$, el mínimo error cuadrático

$$\epsilon_k^2 = \sum_{m=1}^{M}\sum_{n=1}^{N} |u(m,n) - u_k(m,n)|^2, \qquad k = 1, 2, \ldots, r \tag{2.57}$$

se reduce a

$$\epsilon_k^2 = \sum_{m=k+1}^{r} \lambda_m \tag{2.58}$$

Si asumimos que $L \triangleq NM$ se podrá escribir siempre una representación de la transformada bidimensional unitaria como una expansión del producto matricial externo en un espacio L-dimensional, a saber

$$\hat{\boldsymbol{U}} = \sum_{l=1}^{L} w_l \boldsymbol{a_l} \boldsymbol{b_l^T} \tag{2.59}$$

donde w_l son esaclares y $\boldsymbol{a_l}$ y $\boldsymbol{b_l}$ son secuencias de vectores de las bases ortonormales de dimensiones $N \times 1$ y $M \times 1$ respectivamente. El mínimo error cuadrático entre $\mathbf{U}$ y cualquier suma parcial será

$$\hat{\boldsymbol{U}}_k \triangleq \sum_{l=1}^{k} w_l \boldsymbol{a_l} \boldsymbol{b_l^T} \tag{2.60}$$

está minimizado para cualquier $k \in [1, L]$ cuando la anterior expansión coincide con (2.56), es decir, $\hat{\boldsymbol{U}}_k = \boldsymbol{U}_k$.

Esto significa que la energía concentrada en los coeficientes transformados $w_l, l = 1, \ldots, k$ está maximizada por la transformada SVD para la imagen en cuestión. Si se compara con la transformada KL[4] que maximiza la energía media en un número dado de coeficientes transformados, la media debe ser tomada sobre el conjunto para el cual la función de autocorrelación está definida. Por lo tanto, en una base imagen a imagen, la transformada SVD concentrará más energía en el mismo número de coeficientes, aunque habrá que calcular la misma para *cada* imagen. Por otra parte, la transformada KL necesita ser calculada sólo una vez para todo el conjunto de imágenes. Entonces, mientras que existe la capacidad para encontrar una aproximación de transformada rápida para la KL, no se espera un sustituto para la SVD.

[3] Este aspecto condicionará el método que hace uso de esta transformación.

[4] Esta transformada no se explorará en el presente documento, pero se puede profundizar en ella en la sección 5.11 de [Jai89].

2.3. Compresión y formatos de imagen

En la siguiente sección se encontrarán las claves del estudio de los diversos dominios transformados que se han sucedido en el documento. El uso de los mismos está determinado por la necesidad de reducir el tamaño de almacenamiento de las imágenes en un disco duro. Si a su vez un método concreto está relacionado con un formato concreto, propiciará un algoritmo de esteganografiado capaz de introducir datos en ese tipo de formato.

2.3.1. Compresión

Para explicar la compresión, de una forma simple nos valdremos de la explicación dada en [MM97], en la que se definen la **relación de compresión** y la **relación de redundancia**. La primera se define como:

$$C = \frac{n_1}{n_2} \tag{2.61}$$

en la que C se definiría como la relación de compresión, y n_1 y n_2 serían dos cantidades de datos diferentes que representan la misma información. Es evidente que si $n_1 > n_2$ la relación será superior a la unidad y se reducirá el monto del paquete de datos. Seguidamente, procederemos a definir la otra relación:

$$R = 1 - \frac{1}{C} = 1 - \frac{n_2}{n_1} \tag{2.62}$$

En este caso, veremos representada por R la cantidad de información de la información realmente «útil», es decir, R estará cercana a 0 cuando no exista redundancia y, por tanto, no podrá reducirse demasiado el tamaño de su representación. Si por el contrario R se acerca a la unidad, se podrá prescindir del envío de la mayoría de la información. Cabe destacar que habrá que escapar a la situación límite en la que se envíe más información que la original, es decir $n_2 > n_1$.

La definición de la compresión que da en este caso [Jai89] se basa en la **entropía**, que es la representación de la tasa media de información de una fuente con L símbolos independientes posibles con probabilidades $p_i,\ i = 0, \ldots, L-1$, tal y como podría ser un conjunto de datos de imágen en bruto:

$$H = -\sum_{i=0}^{L-1} p_i \log_2 p_i \quad \text{bits por símbolo} \tag{2.63}$$

Además, de acuerdo con el **Teorema de Codificado sin Ruido** de Shannon, es posible codificar sin distorsión una fuente de entropía de H bits por símbolo utilizando $H + \epsilon$ bps, donde ϵ es una cantidad positivia aribrariamente pequeña. Entonces, la máxima compresión C posible viene definida por:

$$C = \frac{B}{H + \epsilon} \tag{2.64}$$

donde B representa la tasa binaria promedio de los datos en bruto y $H + \epsilon$ representa lo mismo para los datos codificados.

Volviendo a [MM97], y al concepto de redundancia, en el caso de las imágenes se pueden explotar tres tipos de la misma, **de codificación**, para la que se estudia la cantidad de veces que aparecen los diferentes valores y se utilizan códigos del tipo Huffman para reducir la codificación, utilizando menos símbolos para aquellos valores más repetidos, **entre píxeles**, referida a cómo agrupar la información, es decir, se busca la agrupación de la información. Para llevar a cabo estos menesteres se puede proceder a transformar el dominio, siendo el de la DCT el idóneo por sus propiedades de concentración de energía. Finalmente, está la **redundancia psicovisual**, que, utilizando la forma en que nuestro cerebro interpreta la información visual, se puede eliminar información de la imagen referida a partes de un mismo objeto que cuenten con pocas variaciones de luminancia que son imperceptibles a nuestro sistema visual [MM97].

2.3.2. Formatos de imagen

Para hablar de formatos de imagen, primero hay que explicar qué es un fichero. Un fichero es una colección de datos formateados de una manera tal que un programa sea capaz de interpretarlo. En algunas ocasiones el formateo es mínimo, por ejemplo en el formato WAV apenas se modifican los datos en bruto, aunque se antepone una cabecera para que los reproductores usuales sean capaz de interpretar esos datos.

A partir del esbozo que se ha presentado, se puede presentar un formato de imagen como la modificación de los datos en bruto de imagen para dar una estructura de fichero que sea capaz de interpretar un visor de imágenes. Existen diferentes tipos de formatos, en los que no entraremos en detalle, salvo en dos aspectos. Cómo construyen la imagen y cómo comprimen la información visual.

Empezando por los primeros:

- En primer lugar se definirá el modelo ***raster*** o **mapa de bits**. Este modelo es el típico de los dispositivos de captura de imágenes, tales como cámaras, que dividen el continuo que supone la realidad en una cuadrícula, cuyo tamaño de cuadro será más pequeño conforme gane en resolución en megapíxeles. La expresión «pixelarse», viene del efecto de bloques que se produce al aumentar las fotografías, que deja entrever la cuadrícula sobre la que se ha asentado la realidad. Los principales formatos asociados a este modelo son PNG, TIFF o JPEG.
- En contraposición al primer modelo, encontramos el ídem **vectorial**, que consiste en dividir la imagen en puntos que se unirán con una línea. En contraposición al anterior, no encontraremos un efecto de bloques al ampliar la imagen. Algunos ejemplos de este modelo son SVG, PDF o EPS.

Teniendo en cuenta que se pretende utilizar imágenes usuales, se ahondará en los formatos del primer modelo. A continuación se expondrán los tipos de compresión:

- La primera manera de compresión, **sin pérdidas** o *lossless*, del inglés, se limita a utilizar los tipos de redundancia que no impiden la recuperación total de la información comprimida. Como ejemplos tendríamos PNG o TIFF.
- La otra forma es **con pérdidas** o *lossy*. En este tipo de compresión, a diferencia de la anterior, no es posible recuperar toda la información, y se producirá una degradación poco significativa de la calidad de la imagen. El ejemplo principal de formato que utiliza este tipo de compresión es JPEG.

Este aspecto es el más importante en términos de nuestro trabajo, puesto que dependiendo de la forma de esteganografiado, si se produce una degradación de la información, puede perderse la modulación efectuada. También cabe destacar que uno de los formatos más utilizado es JPEG, por lo que es de un gran interés el buscar una forma de introducir información en ficheros formateados de esta forma, por lo que profundizaremos más en el proceso de creación de una imagen de este formato.

El formato JPEG

El documento sobre el que se sustenta el estándar de este formato [Wal92], data de 1992. En el se explica la creación de un nuevo estándar internacional por parte de la JPEG *(Joint Photographic Experts Group)*, (Grupo Conjunto de Expertos en Fotografía) para la compresión de imágenes fijas de tono continuo.

El estándar abarca dos tipos posibles de la reducción de capacidad necesaria para el almacenamiento de la información. Por una parte, soporta una compresión con pérdidas, basada en la transformada DCT y, por otro, admite la utilización de un método predictivo basado en la entropía *lossless*.

A continuación, detallaremos los pasos que sigue una codificación basada en la DCT, siguiendo los pasos de la figure 2.13:

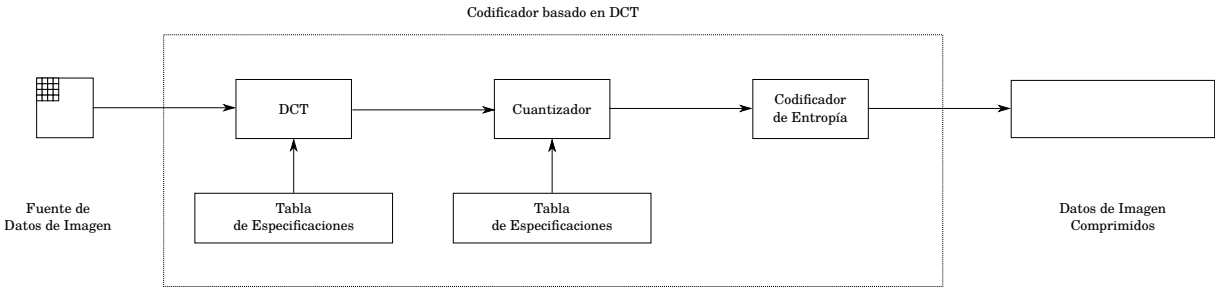


Figura 2.13: Codificador JPEG basado en DCT.

1. División de la imagen en bloques de 8×8 píxeles.
2. Aplicación de una reducción de 128 niveles a cada elemento de cada bloque.
3. Transformación DCT de cada bloque.
4. Cuantización de cada elemento del bloque mediante una (o varias, dependiendo de si es en escala de grises o en color) matriz de cuantización Q.
5. Redondeo de los elementos de cada bloque.
6. Codificación Huffman de los bloques, diferenciando el primer elemento (superior izquierdo) como valor «de continua» y el resto de los mismos como valores «de alterna», aplicando a cada tipo una tabla de transducción diferente. Cabe destacar que el ordenamiento para esta codificación se sigue un orden en *zigzag*.
7. Codificación de entropía.

El paso más importante de este proceso es la cuantización, definida como

$$F^Q(u,v) = Redondeo\left(\frac{F(u,v)}{Q(u,v)}\right) \tag{2.65}$$

Existen dos matrices diferentes de cuantización, una de ellas para los valores de luminancia y otra para los valores de crominancia:

$$Q_Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \tag{2.66}$$

$$Q_C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 24 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix} \tag{2.67}$$

Las anteriores matrices de cuantización son las ofrecidas por el IJG *Independent JPEG Group*, que equivalen a mantener un factor de calidad del 50 % con respecto a la imagen original. Es lógico, tras observar la ecuación (2.65), llegar a la conclusión de que un mayor valor en las matrices de cuantización llevará a una degradación mayor de la imagen original. El método que al que la misma entidad llegó para mejorar el factor de calidad es el siguiente [Cio18]:

$$S = \begin{cases} \frac{5000}{Q_f} & \text{si } Q_f < 50 \\ 200 - 2Q_f & \text{si } Q_f \geq 50 \end{cases}$$
$$Q_{i,j} = \left\lfloor \frac{50 + S + D_{i,j}}{100} \right\rfloor \tag{2.68}$$

Siendo Q_f el factor de calidad deseado en porcentaje entre 0 y 100.

Este tema será retomado en la sección 3.3.

La recuperación de los datos de imagen comprimidos se harán de la siguiente forma:

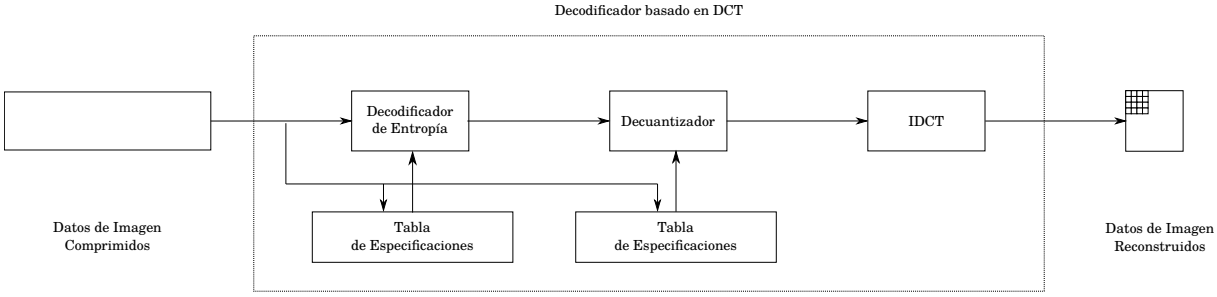


Figura 2.14: Decodificador JPEG basado en DCT.

1. Se recuperan las especificaciones de las tablas de los datos comprimidos.
2. Se decodifica el código de entropía.
3. Se hace la codificación inversa Huffman utilizando el *zigzag*.
4. Se decuantizan los datos.
5. Se calcula la transformada del coseno inversa.
6. Se añaden los 128 niveles al bloque de 8×8.

Capítulo 3

Trabajos previos en esteganografía y estegoanálisis

3.1. Introducción

Tras la explicación del modelo mátemático que rodeará nuestro método, recapitularemos los aspectos más importantes del mismo y la repercusión de los mismos a la hora de implementar métodos esteganográficos en imagen.

En primer lugar, como la esteganografía se basa en la ocultación de la información para que resulte imperceptible a aquellos que no sean receptores directos del mensaje, hemos caracterizado el Sistema Visual Humano (SVH), concluyendo que:

- La sensibilidad a variaciones cromáticas es mayor que a la referida a escala de grises, dado que sólo somos capaces de diferenciar 50 niveles de éstos frente a millones de colores.
- Mediante el experimento de las bandas de Mach se comprobó que la percepción del brillo en variaciones abruptas de luminancia se distorsiona, notando en la transición el valor menor como mayor del que realmente tiene y viceversa con el mayor.
- Utilizando una variación de *grating* sinusoidal, se dedujo la MTF del SVH, llegando a la conclusión de que la sensibilidad del mismo se reduce conforme aumenta la frecuencia espacial.

De estos resultados podremos, entonces, extrapolar el rumbo que habría de llevarse, siendo este el de modificar las partes de una imagen de mayor frecuencia espacial.

Una vez evaluado el SVH, fijamos nuestro objetivo en el análisis de las imágenes, indagando en los modelos matemáticos que las gobiernan para aplicar los resultados deducidos del estudio del SVH, por tanto, podemos sintetizar las partes más relevantes de dicho apartado en:

- Nos apoyamos en la teoría de matrices para el análisis matemático de la información de las imágenes, dada la condición bidimensional de las mismas.
- Para su procesamiento nos basamos en la teoría referente a sistemas lineales, utilizando la convolución como principal operación para llevar a cabo dicho procesamiento (o filtrado).
- Introducimos la transformada de Fourier para un análisis espectral de la imagen, discerniendo la frecuencia espacial referida a las componentes de la imagen en grados por ciclo.

Una vez sentadas estas bases, procedemos a explicar dos de los dominios transformados que más se utilizan para la compresión de información de esta índole, la transformada discreta del coseno y la descomposición en valores singulares. La primera se escogió porque forma parte del algoritmo de compresión del formato JPEG y la segunda por su capacidad de almacenar información con poco impacto sobre la imagen.

Una vez contextualizado nuestro estudio, procederemos a ver un par de ejemplos de métodos existentes para la esteganografía y uno para el estegoanálisis. Sendos métodos esteganográficos se basan en dos de los conceptos explicados previamente, el de la inclusión de la información en las zonas de alta frecuencia espacial, ideando además un algoritmo para agrupar los datos, y otro basado en la DCT. En cuanto al metodo de estegoanálisis inspeccionado, se basa en la comparación del error entre la imagen y una recompresión de la misma a una calidad diferente, siempre y cuando la imagen esté formateada como JPEG.

3.2. Esteganografía

A lo largo de esta sección, detallaremos diversos métodos existentes para la inclusión de datos disfrazados en imágenes. En primer lugar presentaremos el método MBNS, que opera en el dominio real, y a continuación desglosaremos diversos métodos basados en el formato JPG, que detallan la evolución del método F5, partiendo desde F3 y utilizando Jsteg de base.

3.2.1. MBNS

El primer método que evaluaremos, [Zha05], en esta parte se refiere a la esteganografía teniendo en cuenta las limitaciones del SVH para esconder la información en las imágenes. El método que sigue es el de un sistema con múltiples bases, denotando

$$x = (d_{n-1}d_{n-2}\dots d_1 d_0)_{b_{n-1}b_{n-2}\dots b_1 b_0} \qquad \text{donde} \quad 0 \leq d_i < b_i,\ i = 0, 1, \dots, n-1$$

cuyas fórmulas de síntesis y análisis serían las siguientes:

$$x = d_0 + \sum_{i=1}^{n-1} \left(d_i \prod_{j=0}^{i-1} b_j \right) \tag{3.1}$$

$$\begin{aligned} d_0 &= \text{ mód } (x, b_0) \\ d_k &= \text{ mód } \left\{ \frac{1}{\prod_{j=0}^{k-1} b_j} \left[x - d_0 - \sum_{i=1}^{k-1} \left(d_i \prod_{j=0}^{i-1} b_j \right) \right], b_k \right\}, \quad k \geq 1 \end{aligned} \tag{3.2}$$

Una vez transformados los valores se procederá a introducirlos en los diferentes píxeles. Éstos se escogerán de un conjunto S_1 que incluye todos los de la la imagen a excepción de la fila superior y la columna izquierda, que serán los primeros integrantes del conjunto S_0.

Siguiendo una clave (que deberá compartirse en el otro extremo), se irán escogiendo más píxeles que serán introducidos en el conjunto S_0. Una vez seleccionados, se procederá a la inclusión de la secuencia d ebits d ela siguiente manera:

1. Dividir la secuencia en segmentos de l bits.

2. Convertir cada segmento en un entero positivo «x» y asignar el valor 1 a la variable u, que determinará el número de símbolos necesarios en el sistema de múltiples bases anteriormente expuesto.

3. Calcular la desviación típica $\sigma(i,j)$ de los valores de los píxeles esteganografiados $p'(i-1,j), p'(i-1,j-1), p'(i,j-1)$. Cabe destacar que hay que seguir el orden marcado por H para no corromper los datos incluidos con anterioridad. La base se calcula de la siguiente forma:

$$b(i,j) = \text{mín}\left(\left\lceil \frac{\sigma(i,j)}{\Delta} \right\rceil, 16\right) \tag{3.3}$$

   con $\lceil \bullet \rceil$ refiriéndose al entero mayor más próximo (redondeo por exceso) y Δ una constante inversamente proporcional a la cantidad de símbolos introducidos por base y directamente proporcional a la distorsión provocada en la imagen original.

4. Transformar el valor del dígito en el sistema de múltiples bases siguiendo (3.2) con la base: $b(i,j)$,

$$d(i,j) = \quad \text{mód } [x, b(i,j)]$$

5. Modificar el valor del píxel $p(i,j)$ de la siguiente forma:

$$p'(i,j) = \underset{v\in[0,255],}{\arg\text{mín}} \underset{\text{mód } [v,b(i,j)]=d(i,j)}{} |v - p(i,j)| \tag{3.4}$$

   escogiendo entre los valores

$$p'_1 = \left\lfloor \frac{p(i,j) - d(i,j)}{b(i,j)} \right\rfloor b(i,j) + d(i,j)$$

   o

$$p'_2 = \left\lfloor \frac{p(i,j) - d(i,j)}{b(i,j)} + 1 \right\rfloor b(i,j) + d(i,j)$$

   donde $\lfloor \bullet \rfloor$ representa el entero menor más cercano (truncamiento). Por tanto, el *estegovalor* es escogido entre estos dos.

6. Actualizar el parámetro u como $u \leftarrow ub(i,j)$. si $u < 2^l$. Si $u < 2^l$, significa que el segmento no se ha representado totalmente. En ese caso, hay que volver al paso 4, tras actualzar el valor de x de la siguiente forma:

$$x \leftarrow \frac{x - d(i,j)}{b(i,j)} \tag{3.5}$$

   En otro caso, se vuelve al paso 3 para modular otro segmento. De esta forma, cada segmento estará embebido en diferentes píxeles de la imagen portadora.

En el lado de la extracción, la clave secreta y el parámetro Δ se utilizan para recuperar el mensaje modulado. Tras obtener la secuencia H de la clave, las bases $b(i,j)$ pueden ser ordenadas, calculándolas de la imagen esteganografiada utilizando (3.3), y el valor de $d(i,j)$ se obtiene cuando $b(i,j)$ es mayor que 1.

$$d(i,j) = \quad \text{mód } [p'(i,j), b(i,j)] \tag{3.6}$$

Tomando una serie de $b(i,j)$ mayor que 1, consecutivamente, en el orden indicado por H, denotado como $b(t)$, $t = 1, 2, \ldots, T$, donde $T \leq (M-1)(N-1)$. Reescribiendo $\{b(1), b(2), \ldots, b(T)\}$ como

$\{b(1), b(2), \ldots, b(T_1+1), b(T_1+2), \ldots, b(T_2), \ldots, b(T_k+1), b(T_k+2), \ldots, b(T_{k+1}), \ldots\}$ bajo la siguiente condición:

$$\begin{aligned} &\prod_{t=T_{k+1}}^{T_{k+1}-1} b(t) < 2^l \\ &\prod_{t=T_{k+1}}^{T_{k+1}} b(t) \geq 2^l, \quad k = 0, 1, 2, \ldots;\ T_0 = 0 \end{aligned} \tag{3.7}$$

Entonces, cada segmento binario modulado puede ser extraido de los correspondientes símbolos $d(T_k + 1), d(T_k + 2), \ldots, d(T_{k+1})$.

3.2.2. Diversos métodos basados en algoritmo JPEG

Los siguientes trabajos que definiremos, recogidos en [Wes01], tienen en común que utilizan el algoritmo de compresión de JPEG para embeber los datos en la estegoimagen. Partiremos por el primer método explicado en dicho documento.

Jsteg

El primer método explicado que es presentado como el punto de partida, modifica el LSB de los coeficientes en frecuencia por el mensaje secreto en aquellos componentes superiores a 1.

El problema relatado en [Wes01] es que, si bien no presenta anomalías visuales, si ejecutamos un ataque estadístico contra un esteganografiado de este tipo encontraremos indicios de que la imagen ha sido modificada para incrustar en ella un mensaje. El ataque estadístico se verá en profundidad en la sección 3.3.

Método F3

El método F3, dos versiones inferior a F5, no se basa en la sobrescritura de bits, sino que decrementa los valores absolutos en el caso de que el LSB no se corresponda, salvo para los casos en que el valor absoluto del coeficiente sea cero.

En este caso también se pueden observar anomalías en las frecuencias de diversos coeficientes, viéndose afectados en este caso los valores pares que aparecerán más frecuentemente.

Método F4

F4 trata de reparar dos debilidades de F3, éstas son:

1. Debido al estrechamiento de los ceros esteganográficos, F3 incrusta más ceros que unos, y produce, así como JSTEG, peculiaridades estadísticas en el histograma.
2. El histograma de los ficheros JPEG contiene más elementos impares que pares, excluyendo el cero. Por tanto, la portadora sin cambios, desde el punto de vista de Jsteg o F3, contendrá más valores impares que pares.

Para solventar estas deficiencias, el algoritmo F4 hace uso del *mapeo* de coeficientes negativos al valor esteganográfico inverso, es decir, los valores negativos pares representarán un uno esteganográfico, un impar un cero y viceversa para los positivos.

Método F5

Finalmente, en [Wes01], se explica el algoritmo F5, siendo la última evolución de los dos anteriores.

El primer cambio que se detalla de F5 con respecto a los anteriores, es que distribuye los cambios por toda la imagen, en vez de agrupar todos los cambios en el inicio de la imagen. Además, para prevenir ataques, la función de incrustación debería embeber los datos con una densidad similar por todas partes.

La implementación de este algoritmo es la que sigue:

1. Se inicia la compresión JPEG y se para tras la cuantización de los coeficientes.
2. Se inicializa un generador de números aleatorios criptológicamente fuerte, con la clave derivada de la contraseña que introduzca el usuario, que permitirá la recuperación de datos.
3. Instanciación de la permutación con dos parámetros: un generador aleatorio y número de coeficientes.
4. Se determina el parámetro k de la capacidad del medio portador, y la longitud del mensaje secreto.
5. Cálculo de la longitud de la palabra clave $n = 2^k - 1$.
6. Se inserta el mensaje secreto con una matriz de codificación $(1, n, k)$ de la siguiente forma:
   a) Llenado de un búfer con n coeficientes distintos de cero.
   b) Generación de un valor *hash* con k espacios de bit.
   c) Se ejecuta una operación XOR bit a bit para añadir los siguientes k bits del mensaje al valor *hash*.
   d) Si la suma es 0, el búfer se conserva sin cambios, en otro caso , la suma es el índice del búfer $1 \ldots n$, el valor absoluto de cada elemento debe ser decrementado.
   e) Se hace un test para el *estrechamiento*, es decir, cada vez que se produce un cero. Si se da el caso, hay que ajustar el búfer (eliminando el 0 leyendo un coeficiente distinto de cero más, lo que significa repetir el paso 6a desde el mismo coeficiente). Si no sucede este fenómeno, se avanza hacia nuevos coeficientes tras el búfer actual. Si aún quedan datos, también se procede de nuevo al paso 6a.
7. Continuar con la compresión JPEG (codificación Huffman, etc.).

3.3. Estegoanálisis

En la siguiente sección se detallarán dos aproximaciones diferentes para la búsqueda de información oculta en imágenes. En primer lugar veremos el método de PhotoForensics, *ELA* y después veremos diferentes técnicas para realizar ataques estadísticos.

3.3.1. Análisis del nivel de error

Para hacer un estudio de si una imagen ha sido modificada, se pueden aplicar diferentes procesamientos. En nuestro caso de estudio, utilizaremos un método en concreto que se puede encontrar en [Kra07]. En este artículo se detallan métodos forenses para deducir las manipulaciones a las que se han podido ver sometidas diferentes fotografías. Además, en este estudio se encuentra un método propio para deducir el nivel de compresión del formato JPEG.

El primer método que propone [Kra07], se basa en la inspección de los metadatos, es decir, los datos que hacen a un formato, dicho formato (cabecera, etc.). En este trabajo en particular no entraremos en este tipo de análisis, puesto que nos centraremos en los propios datos de imagen.

El siguiente método propuesto, PCA, se basa en la inspección de las tres varianzas que surgen a través de las tres dimensiones de la distribución de colores de los píxeles de una imagen. Este método no tiene tan buenos resultados como el siguiente que se expondrá, que además da nombre a esta subsección.

El método que se presentará es el ELA *(Error Level Analysis)*. Este método se basa en hallar el error de la diferencia entre una imagen y una compresión de calidad muy alta. El principio de funcionamiento de este método consiste en la igualación de los píxeles adyacentes con las sucesivas compresiones, lo que significa que, con cada compresión efectuada, los niveles de error serán cada vez menores. Si a una imagen que ya ha sido comprimida se le aplica una transformación por medio de algún programa de edición, estos montajes se percibirán como una parte de error muy alto en comparación con el resto de la imagen. A continuación mostraremos varios ejemplos.

Como se puede ver, en la imagen (b) de la figura 3.1,los niveles de error son altísimos porque nunca se ha comprimido la imagen, procedente del formato TIFF, en las dos siguientes, (d) y (f), el error disminuye y en la (h), se puede comprobar cómo la luna agregada destaca sobre el resto del error de la imagen.

3.3.2. Ataques estadísticos

Como se adelantó anteriormente, en [Wes01] también encontramos un método de estegoanálisis basado en la estadística. Para ello nos valdremos de los métodos `Jsteg` y `F3` vistos en la anterior sección 3.2

Ataque basado en χ^2

En el caso de Jsteg, el reemplazamiento de bits produce una dependecia entre los valores de frecuencia de aparición, que sólo difieren en la posición de bit, en este caso LSB. La influencia producida recae en la frecuencia de aparición de pares de coeficientes adyacentes, como se puede ver en la figura 3.2. Si c_i es el histograma de los coeficientes JPEG la asunción de una imagen modificada es que las frecuencias adyacentes $c_{2i} y c_{2i+1}$ son similares. Tomando la media aritmética:

$$n_i^* = \frac{c_{2i} + c_{2i+1}}{2} \tag{3.8}$$

para determinar la distribución esperada y comparar frente a la distribución observada:

$$n_i = c_{2i} \tag{3.9}$$

La diferencia entre ambas distribuciones dada como:

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - n_i^*)^2}{n_i^*} \tag{3.10}$$

con $k-1$ grados de libertad, los cuales són el número de diferente de categorías en el histograma menos uno.

En la figura 3.3 se muestra el ataque estadístico en una estegoimagen Jsteg (con un 50 % de la capacidad usada). El diagrama presenta la probabilidad de inserción:

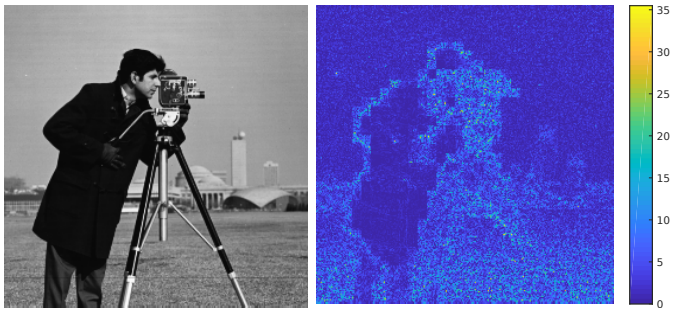
$$p = 1 - \frac{1}{2^{\frac{k-1}{2}}\Gamma\left(\frac{k-1}{2}\right)} \int_0^{\chi^2} t^{\frac{k-1}{2}-1} \exp\left(-\frac{t}{2}\right) dt \tag{3.11}$$

como una función de muestra incremental, es decir, inicialmente, la muestra comprime el primer 1 % de los coeficientes JPEG, después el 2, el 3 %, etc. La probabilidad es de uno hasta el 54 % y 0,45 al 56 %. Una muestra de 59 % y más contiene suficientes coeficientes sin cambiar que produce una probabilidad que prácticamente será 0.

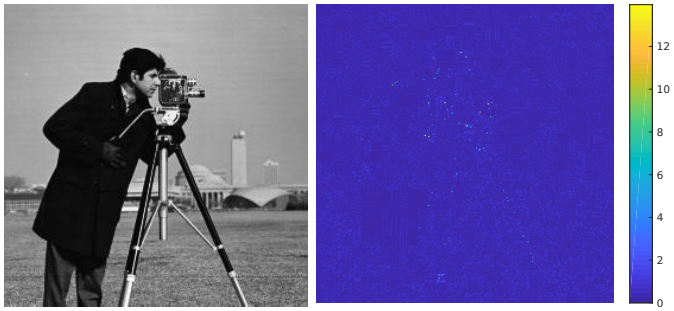
Ataque basado en la observación del histograma

Si bien en el anterior apartado hemos podido comprobar que en el histograma se apreciaban las modificaciones, en este caso, el diseño de F3 no llega a tal extremo de proporcionar una forma de calcular la probabilidad de incrustación de mensajes.

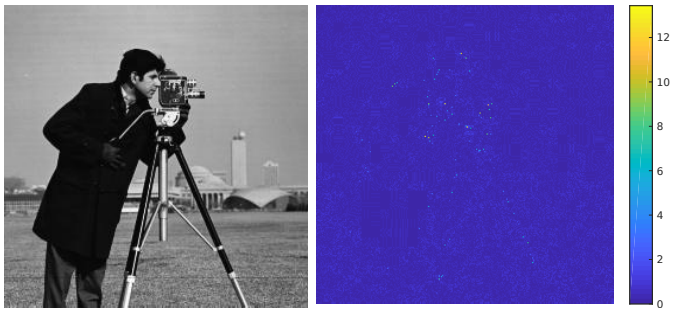
En el caso de F3, dado su funcionamiento de decrementar el valor absoluto en uno, se producirá una simetría entre los valores positivos y negativos en el histograma que nos hará sospechar de una inserción de datos, como se puede observar en la figura 3.4.



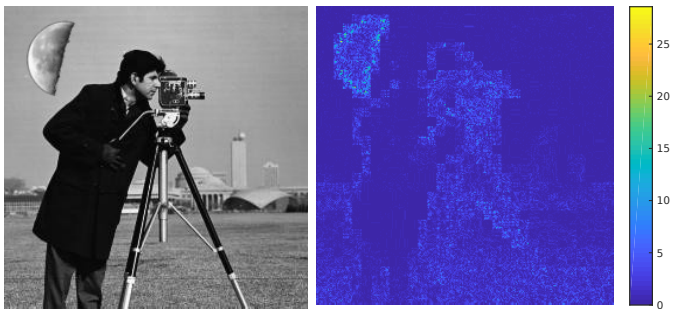
(a) Imagen original (b) ELA imagen original.



(c) Imagen comprimida al 75 %. (d) ELA imagen comprimida al 75 %.



(e) Imagen recomprimida al 75 %. (f) ELA imagen comprimida al 75 %.



(g) Imagen modificada. (h) ELA imagen modificada.

Figura 3.1: Comparación variación de niveles de error entre la imagen original «cameraman» y versiones modificadas o comprimidas.

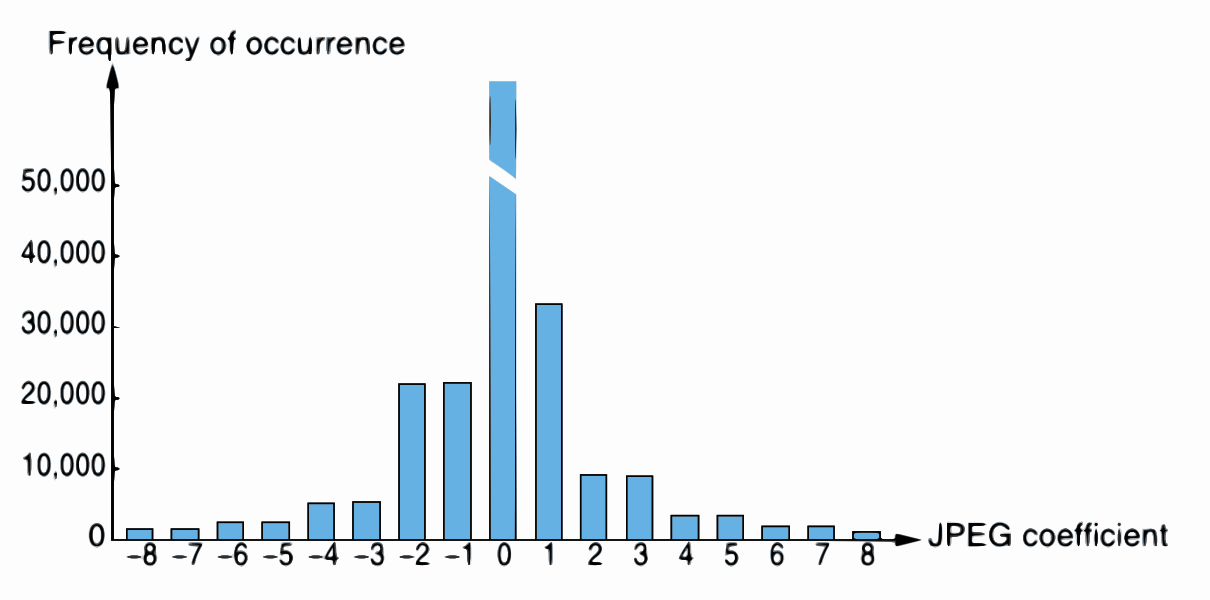


Figura 3.2: Histograma de una estegoimagen Jsteg.

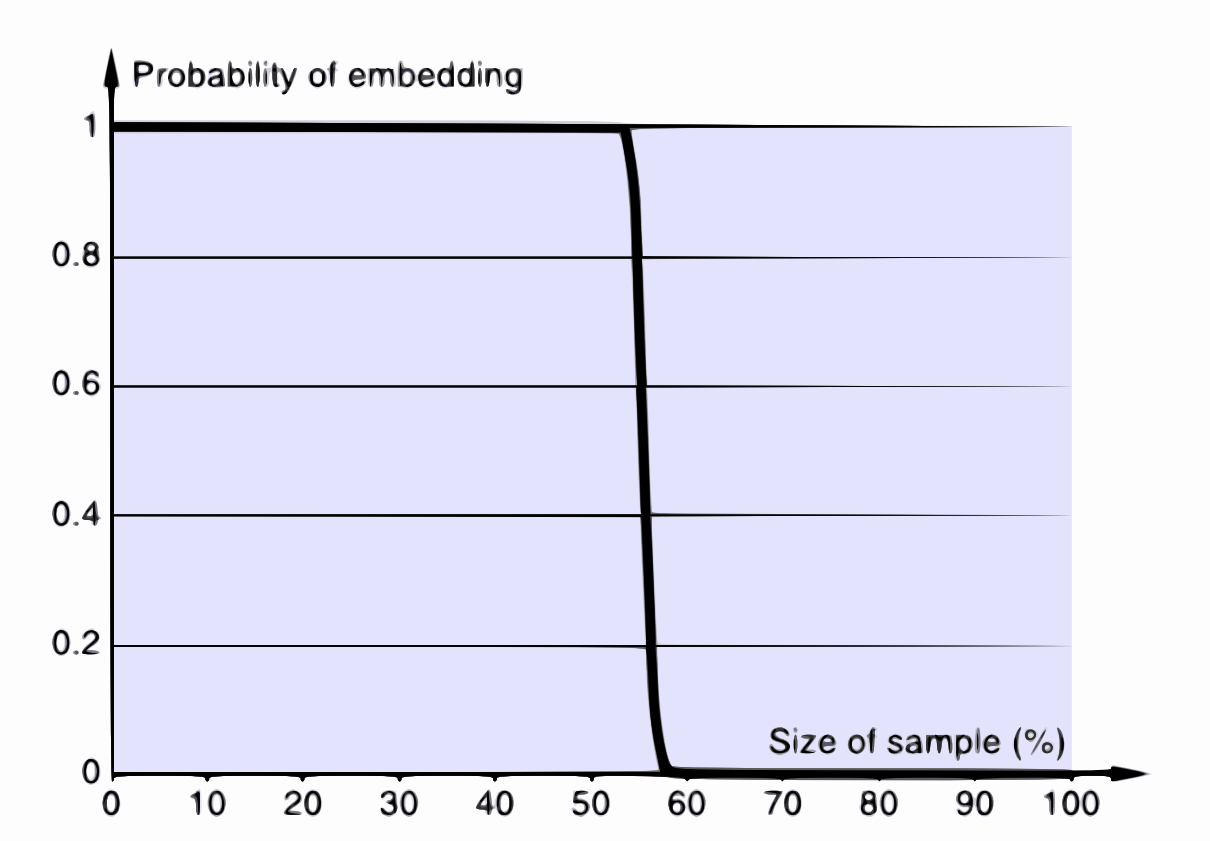


Figura 3.3: Distribución de probabilidad de inserción de datos.

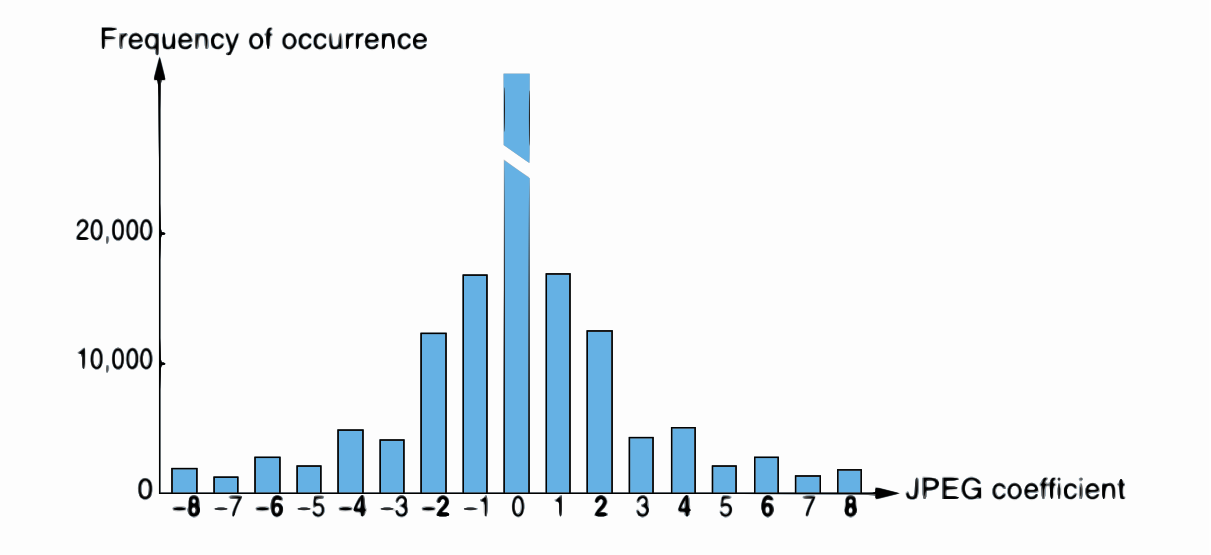


Figura 3.4: Histograma de una estegoimagen F3.

Capítulo 4

Propuesta

4.1. Esteganografiado adaptativo en zonas de alta frecuencia espacial

Nuestra primera propuesta está basada en la incrustación de la información en el bit menos significativo de los píxeles designados. Como se introduce en el título de esta sección, utilizaremos las zonas de alta frecuencia espacial, dado que será en éstas en las que la percepción de nuestro sistema visual se vuelve menos sensible y, por tanto, más imperceptible se volverá la información escondida.

La determinación de las zonas de alta frecuencia espacial viene dada por el filtrado de la imagen para hallar las diferentes varianzas locales, para lo que se utilizará un *kernel* de tamaño $N \times N$ de tamaño seleccionable, cuyos elementos valdrán $\frac{1}{N^2}$. Una vez seleccionado el tamaño del mismo hallaremos las diferentes varianzas locales, tras lo que aplicaremos un umbral λ. La utilización de estos dos elementos de tamaño variable será lo que al final nos dé la adaptación al tamaño del texto.

Dado que la condición fundamental de cualquier algoritmo de esteganografiado es la capacidad de extraer el mensaje embebido, si utilizamos nuestra imagen original para el cálculo de las varianzas locales, al modificar los valores de los bits menos significativos de estos píxeles, podremos modificar las varianzas locales y perder el mensaje de forma irreversible. La solución a esta problemática pasa por calcular una copia de nuestra imagen, truncando el valor del bit menos significativo.

Una vez obtenida nuestra imagen «truncada», deberíamos comenzar a calcular las varianzas locales, pero si utilizamos la imagen en bruto, al filtrar, los bordes se interpretarán como una zona de alta frecuencia espacial. Para dar solución a este problema, se procederá a replicar la imagen de forma simétrica por todos sus márgenes, para suavizar al máximo este efecto y, tras el procesamiento recuperaremos la parte de interés.

Finalmente, realizaremos la raíz cuadrada del valor absoluto de nuestra matriz de varianzas locales para obtener las desviaciones típicas locales. Este paso se lleva a cabo dado que la varianza aglomera todos los valores en torno al 0, puesto que no da margen a *outliers*, mientras que la desviación típica ofrece un mayor dispersión. Tras obtenerla, normalizaremos los valores para utilizar un umbral entre 0 y 1.

Tras la aplicación del umbral, como resultado obtendremos una matriz «mapa» con unos en los valores susceptibles de ser modificados y con ceros en el caso contrario. A partir de aquí, el proceso de

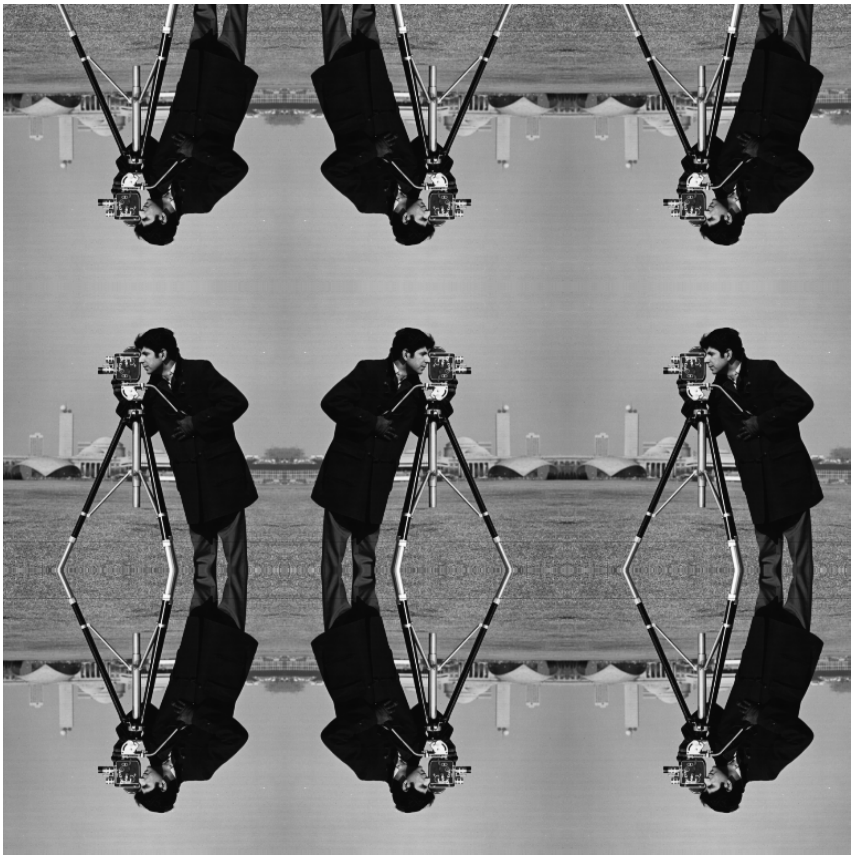


Figura 4.1: Imagen replicada.

incrustación será el siguiente:

$$\begin{array}{ll} \text{Si } d(i)=1 & \begin{cases} I(i) \text{ impar} \longrightarrow S(i)=I(i) \\ I(i) \text{ par} \longrightarrow S(i)=I(i)+1 \end{cases} \\ \text{Si } d(i)=0 & \begin{cases} I(i) \text{ impar} \longrightarrow S(i)=I(i)-1 \\ I(i) \text{ par} \longrightarrow S(i)=I(i) \end{cases} \end{array} \tag{4.1}$$

En la parte de la recepción, se utilizarán los mismos valores para λ y el mismo tamaño del *kernel*, siguiendo los mismos pasos, salvo que recuperaremos un 1 de los valores marcados con el «mapa» si son impares y un 0 para los pares.

De forma operativa, se seguirán los siguientes pasos:

1. Tras escoger nuestra imagen portadora, realizamos un procesamiento para crear una copia que obvie el último bit, es decir:
$$I' = 2\left\lfloor \frac{1}{2} I \right\rfloor \tag{4.2}$$

2. El siguiente paso consiste en replicar la imagen volteada 8 veces para anular el efecto de bordes a la hora de filtrar la imagen.

   Se puede ver claramente que, de saltarnos este paso, el efecto de bordes sería palpable, como se puede ver en la figura 4.2.

3. A continuación procederemos a filtrar la imagen mediante el filtro h cuyas características son las siguientes:

$$h = \begin{pmatrix} \frac{1}{N^2} & \cdots & \frac{1}{N^2} \\ \vdots & \ddots & \vdots \\ \frac{1}{N^2} & \cdots & \frac{1}{N^2} \end{pmatrix}, \qquad h \in (N \times N) \tag{4.3}$$

Filtraremos tanto la imagen de partida, I, como a ella misma con sus elementos al cuadrado, con

$$I_2 = \begin{pmatrix} i_{1,1}^2 & \cdots & i_{1,M}^2 \\ \vdots & \ddots & \vdots \\ i_{N,1}^2 & \cdots & i_{N,M}^2 \end{pmatrix}$$

de la siguiente manera:

$$\begin{aligned} M &= h * I \\ M_2 &= h * I_2 \end{aligned} \tag{4.4}$$

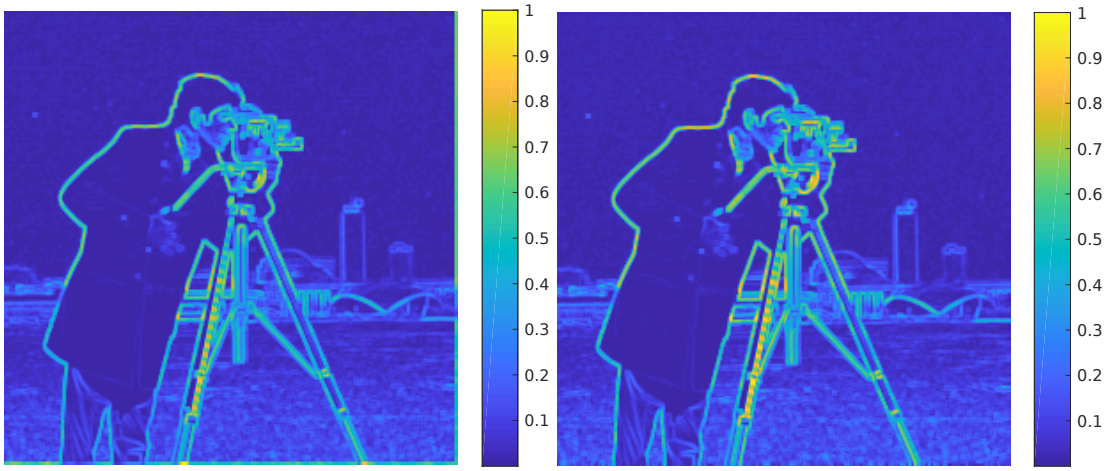
4. Una vez conseguida la matriz de varianzas, calcularemos la raíz cuadrada de su valor absoluto para utilizar la desviación típica en detrimento de la varianza, dado que ésta está más distribuida. Acto seguido normalizaremos el resultado para tener todos los valores en el intervalo$[0, 1]$.

5. El siguiente paso, una vez procesada la imagen, es aplicar un umbral, λ, para conseguir una máscara de aquellos píxeles que serán modificados.

6. Finalmente, modularemos la información introduciéndola en los píxeles seleccionados tras aplicar la máscara. El algoritmo de esteganografiado será el siguiente:

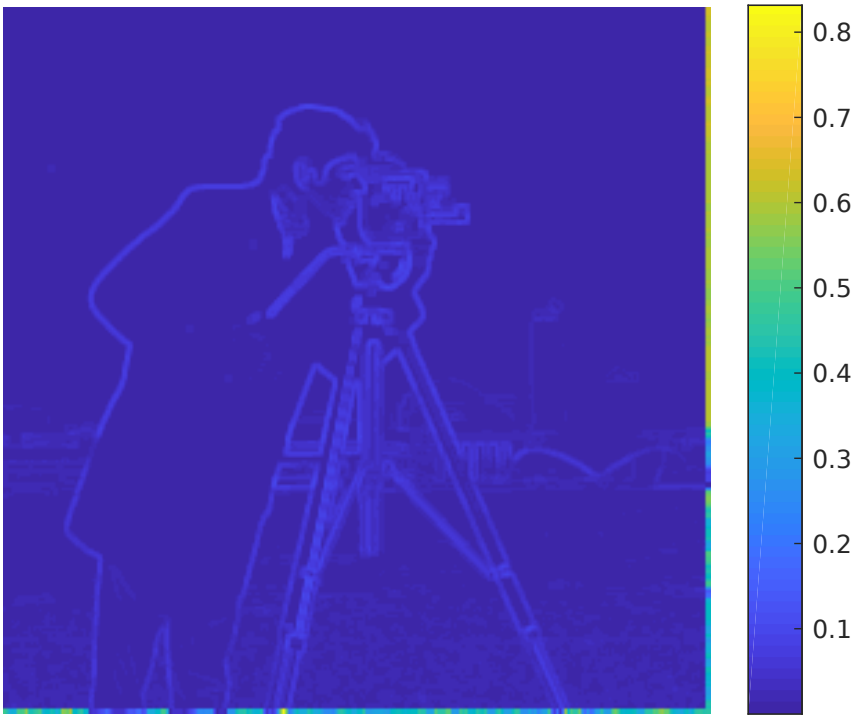
$$\begin{aligned} &\text{Si } d(i) = 1 \quad \begin{cases} I(i) \text{ impar} \longrightarrow S(i) = I(i) \\ I(i) \text{ par} \longrightarrow S(i) = I(i) + 1 \end{cases} \\ &\text{Si } d(i) = 0 \quad \begin{cases} I(i) \text{ impar} \longrightarrow S(i) = I(i) - 1 \\ I(i) \text{ par} \longrightarrow S(i) = I(i) \end{cases} \end{aligned} \tag{4.5}$$

En la figura 4.4 se puede ver cómo se han introducido datos en la imagen.



(a) Desviaciones típicas locales imagen no replicada.

(b) Desviaciones típicas locales imagen replicada.



(c) Diferencia de desviaciones típicas locales.

Figura 4.2: Error de bordes en cálculo de desviaciones típicas locales.

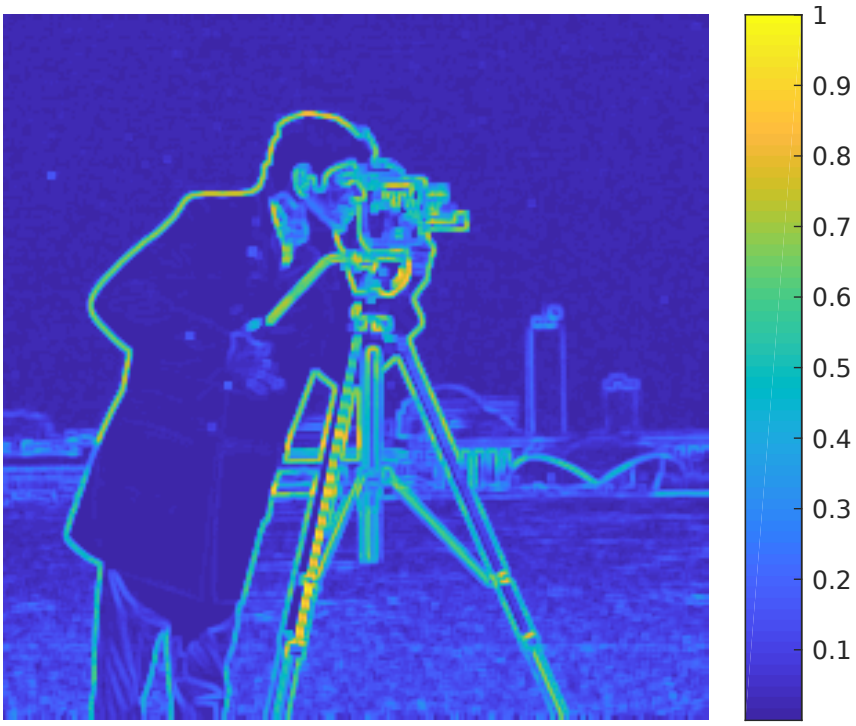
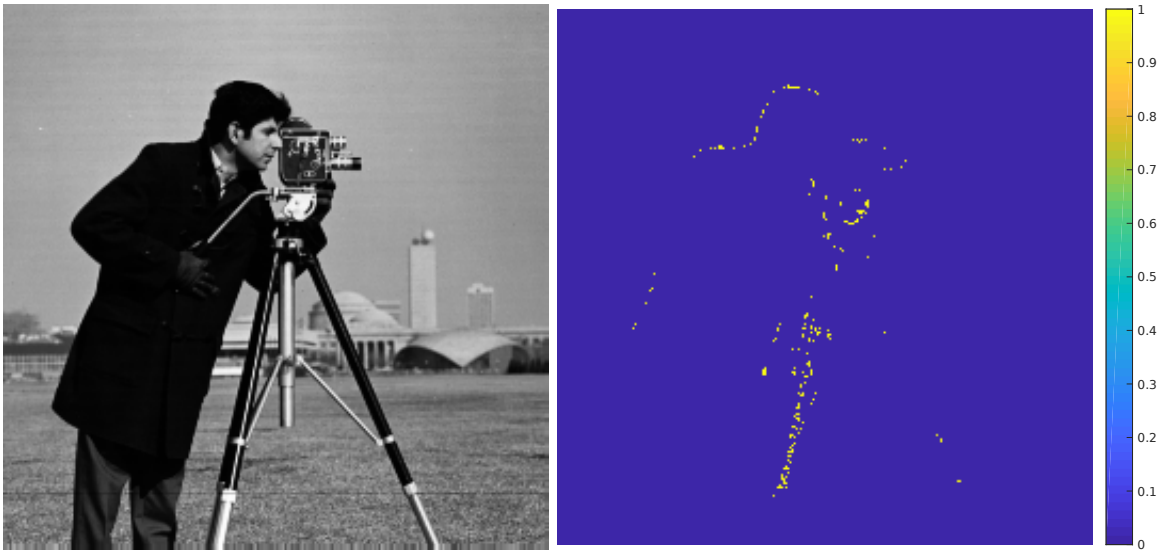


Figura 4.3: Valores de desviación típica normalizados. Los valores más altos se corresponden con los cambios más bruscos, normalmente bordes.



(a) Estegoimagen. (b) Imagen diferencia.

Figura 4.4: Estegoimagen junto con variación frente a original.

Para obtener la secuencia modulada en la imagen portadora, será necesario, además de la imagen, conocer la dimensión del filtro de varianzas y el umbral utilizado. La extracción se torna trivial: una vez repetidos los pasos del 1 al 4 del algoritmo anterior, sólo hay que poner en orden los valores que quedan tras aplicar la máscara, se sustituirán los valores impares por unos, y los pares por ceros.

4.2. Esteganografiado basado en JPEG

El siguiente método que presentaremos estará basado en el dominio de la Transformada Discreta del Coseno, dado que es la transformación utilizada por el algoritmo de compresión JPEG. Además de esta transformada, también utilizaremos la división en bloques de 8×8, así como la cuantización característica de este formato. El método, básicamente, consiste en modificar los n últimos valores de cada bloque, teniendo en cuenta el orden en zigzag utilizado por el algoritmo JPEG, por los n bits que correspondan del mensaje a enviar. El mensaje, por tanto, se distribuirá entre varios bloques.

El *modus operandi* seguido empieza por la modificación de la matriz de cuantización que se vaya a utilizar. En nuestro caso, será la matriz Q_Y presente en [Wal92]. La modificación, denotada como Q' afectará a los últimos n elementos, siguiendo el orden inverso al zigzag utilizado en el formato. El nuevo valor que tendrán los valores de Q' será unitario, para que no elimine los valores introducidos. El siguiente paso será la consecución de la matriz correspondiente a todos los bloques transformados siguiendo el algoritmo JPEG con nuestra matriz Q'. Una vez obtenida dicha matriz, se irán escogiendo los diferentes bloques para incluir los n bits del mensaje correspondientes. En el lado de la recepción, nos valdremos del conocimiento de los n bits que se embeberán por bloque para construir una matriz Q' con la que poder recuperar los valores correspondientes a esas posiciones, y así conseguir reconstruir el mensaje original. Dado el amplio uso que se da al formato JPEG, hemos desarrollado un algoritmo para esteganografiar imágenes. Cabe destacar que la explicación que se seguirá estará aplicada sólo a un canal, pero sería similar para una imagen con información cromática. Para este caso, huelga decir que es imprescindible una transformación previa al espacio YCBCR, que es donde se aplican las matrices de cuantización de bloques del estándar JPEG, como se puede ver en [Wal92]. El algoritmo en cuestión es el que sigue:

1. En primer lugar, aplicaremos el algoritmo usual de JPEG, previamente descrito, reduciendo en 128 niveles cada píxel antes de efectuar la DCT.
2. Seguidamente, haremos uso del ordenamiento en zigzag, en nuestro caso en orden inverso, para incluir el valor de los n bits correspondientes a cada bloque en los últimos elementos.
3. El siguiente paso es modificar la matriz de cuantización Q para que no destruya los n bits embebidos, sustituyendo el valor original por la unidad.
4. Finalmente, se procederá a la construcción de la estegoimagen, utilizando nuestra matriz Q' modificada, sin olvidar añadir los 128 niveles tras aplicar la transformación DCT inversa.

A la hora de la recepción, bastará con aplicar el primer punto anterior con la matriz Q', y recoger los n bits en el orden indicado.

4.3. Esteganografiado basado en SVD

El último método que presentaremos estará fundamentado en la descomposición de la matriz imagen en sus valores singulares, siguiendo el proceso visto anteriormente. Como ya sabemos, el resultado de la aplicación de esta tranformada nos dejará que $I = UDV^T$, siendo D una matriz diagonal. Si bien en

los anteriores métodos contábamos con un número de «huecos» *a priori* iguales al número de píxeles de la imagen, en este caso contaremos sólo con una de las dimensiones de la matriz $M \times N$. Para solventar este problema, se agruparán los bits del mensaje de b en b para poder incluir más datos.

Lógicamente, se sobrescribirán los n últimos valores del vector diagonal de la matriz D, dando lugar a una nueva matriz D'. Para proceder con este algoritmo, es necesario un estudio previo de los valores alojados en dicha matriz D antes de su sobrescritura, dado que por su construcción, la transformación siempre ordena los valores en orden descendente[1], por lo que habrá que comprobar qué valores tienen los elementos que no serán sobrescritos, dado que de aglomerar demasiados bits por símbolo, podrían suplantar sus posiciones y degradar en gran medida la integridad de la imagen.

Una vez obtenida nuestra nueva matriz D', obtendremos la estegoimagen S como

$$S = UD'V^T \tag{4.6}$$

La estaganografía basada en SVD se desarrollará de la siguiente forma:

1. Cálculo de los valores singulares de la matriz de la imagen cumpliéndose la igualdad:
$$I = UDV^T$$
2. Agrupación de los bits del mensaje de b en b.
3. Obtención del valor decimal de dichos fragmentos.
4. Sustitución de los N últimos valores del vector formado por la diagonal principal de la matriz D por el vector que forman los valores decimales de los fragmentos del mensaje.
5. Creación de la matriz diagonal D' mediante el vector modificado.
6. Recuperación de la estegoimagen mediante la aplicación de
$$S = UD'V^T$$

Una vez se tiene la estegoimagen S, para la recuperación de la información modulada bastará con repetir el paso 1, donde se recuperará la matriz D' de la que se obtendrá un vector compuesto por los elementos de la diagonal principal de esta matriz y, una vez recuperado, comprobar los N últimos valores convenidos en donde se encuentra la información. Cabe destacar que este algoritmo ordena automáticamente de forma descendente los valores, por lo que será necesario implementar un algoritmo de ordenación del mensaje que, dadas sus características, excede las competencias de este trabajo.

4.4. Estegoanálisis: ataque a texto claro con mensaje desconocido

En esta primera propuesta de estegoanálisis, trataremos de extraer un mensaje de una estegoimagen creada mediante nuestro primer método propuesto, asumiendo que conocemos la ventana de filtrado, es decir, teniendo como única variable el umbral λ. Nuestra propuesta se podría resumir como un ataque de fuerza bruta, utilizando diferentes umbrales, entre dos límites, definiendo cuál es el número de pasos que se dan entre ambos márgenes.

4.5. Estegoanálisis: ataque a texto claro con mensaje conocido

En este caso, añadiremos un comparador al caso anterior para comprobar si se ha encontrado el mensaje conocido.

[1] Dada la premisa de este trabajo, en el que se pretende esconder información, entendemos que la ordenación excede los objetivos del mismo.

Capítulo 5

Resultados

En el presente capítulo se procederá a evaluar el rendimiento de los diferentes métodos propuestos, aportando todos los datos pertinentes, tales como las imágenes originales, las imágenes esteganografiadas, y una imagen diferencia para comprobar en qué difieren las imágenes.

Los métodos de evaluación de las imágenes serán SSIM [Wan04] y QILV [AF06], para comprobar cómo de parecidas son las imágenes, midiendo su similitud estructural, y la clásica PSNR, para medir el pico de su ratio señal a ruido, en función del Error Cuadrático Medio (MSE).

El orden será el acostumbrado, empezando por el primer método de esteganografiado presentado.

5.1. Esteganografiado adaptativo en zonas de alta frecuencia espacial

En esta sección evaluaremos el primer método, utilizando la imagen «cameraman», imagen en escala de grises de tamaño 256×256 píxeles y la imagen «lenna», en color con un tamaño de 512×512 píxeles. En cuanto a los datos presentados en la tabla 5.1, hay que precisar que el campo «Capacidad Imagen» se refiere a la capacidad de la imagen teniendo en cuenta los parámetros λ y tamaño de ventana, así como que el porcentaje de cambio se refiere a la totalidad de los píxeles de la imagen. Finalmente, se ha de hacer una última precisión con respecto a la fila correspondiente al primer canal del segundo esteganografiado en el espacio YCBCR, en el que no se aplica ningún cambio pero sí hay cambios en la estructura en la parte de evaluación; esto se debe a que la comparación es una vez retornado al espacio RGB, sobre el que se ven las imágenes. Finalmente, hay que precisar que la conversión a YCBCR y su posterior reconversión a RGB puede hacer que se pierda parte del mensaje (lo que podría hacerlo inservible) y, aunque a efectos prácticos dicha esteganografía sería inválida, para nuestro propósito (ilustrar el rendimiento de un espacio frente al otro) es perfectamente válido.

Las imágenes modificando el umbral se pueden ver en la figura 5.1, cuyas imágenes diferencia se corresponden con la figura 5.2. Ídem para la variación de la ventana figuras 5.3 y 5.4 y color, figuras 5.5 y 5.6. Los datos extraídos de ellas se pueden encontrar en la tabla 5.1.

(a) Imagen original.

(b) Estegoimagen de 500 bits.

(c) Estegoimagen de 1000 bits.

(d) Estegoimagen de 2000 bits.

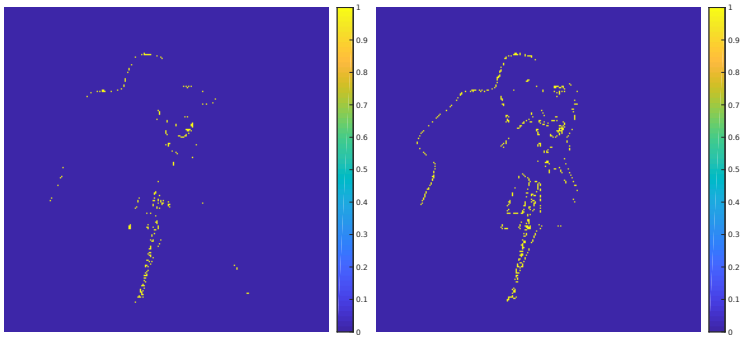
(e) Estegoimagen de 5000 bits.

(f) Estegoimagen de 10000 bits.

(g) Estegoimagen de 25000 bits.

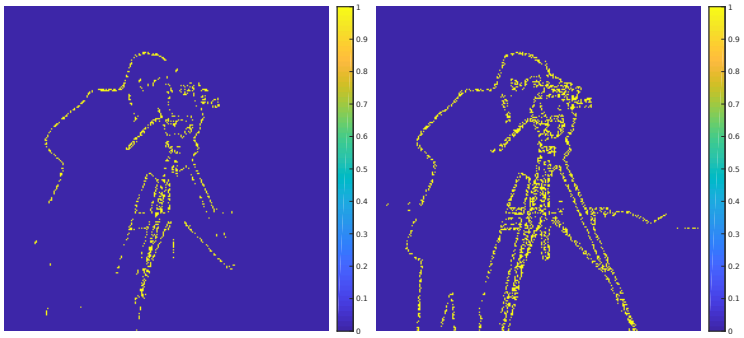
(h) Estegoimagen de 50000 bits.

Figura 5.1: Comparativa imagen original vs. estegoimágenes variación de umbral.



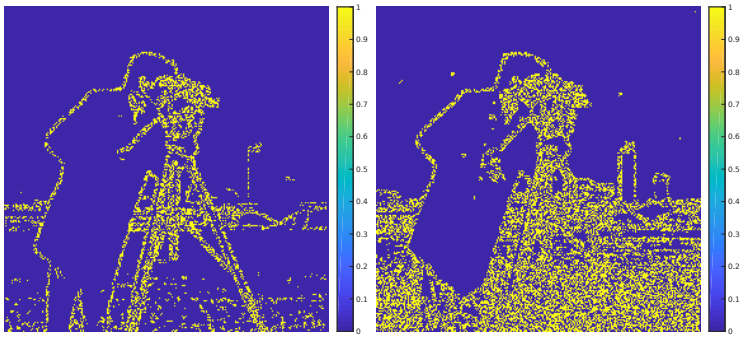
(a) Mensaje de 500 bits.

(b) Mensaje de 1000 bits.



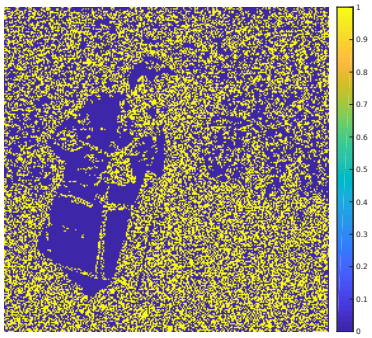
(c) Mensaje de 2000 bits.

(d) Mensaje de 5000 bits.



(e) Mensaje de 10000 bits.

(f) Mensaje de 3000 bits.



(g) Mensaje de 50000 bits.

Figura 5.2: Imagen diferencia modificando el umbral λ, con ventana de 3×3.

(a) Imagen original.

(b) Estegoimagen con ventana de 3×3.

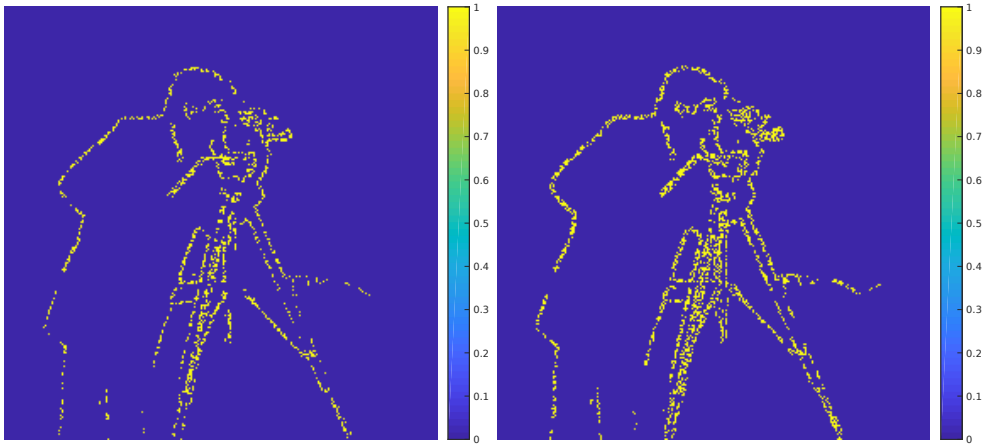
(c) Estegoimagen con ventana de 4×4.

(d) Estegoimagen con ventana de 8×8.

(e) Estegoimagen con ventana de 16×16.

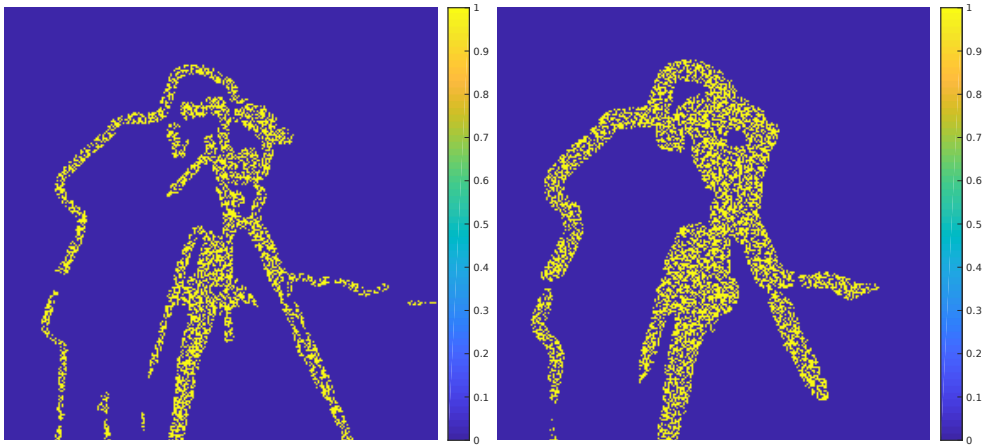
(f) Estegoimagen con ventana de 32×32.

Figura 5.3: Comparativa imagen original vs. estegoimágenes variación de ventana.



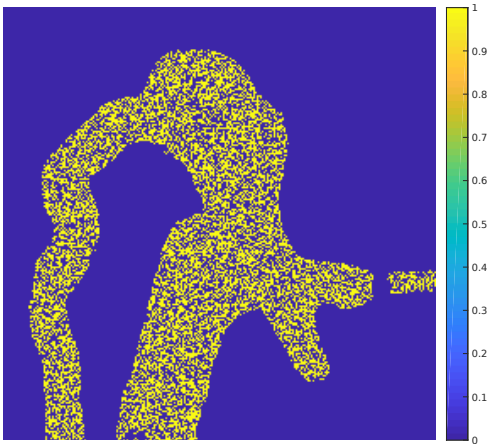
(a) Ventana de 3×3.

(b) Ventana de 4×4.



(c) Ventana de 8×8.

(d) Ventana de 16×16.



(e) Ventana de 32×32.

Figura 5.4: Imagen diferencia modificando la ventana, con $\lambda = 0,5$.

(a) Imagen original.

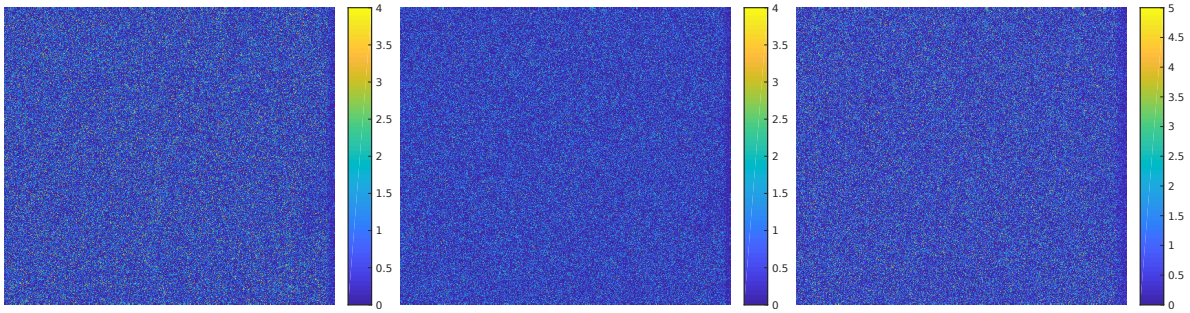
(b) Estegoimagen en espacio YCBCR.

(c) Estegoimagen 1 en espacio RGB.

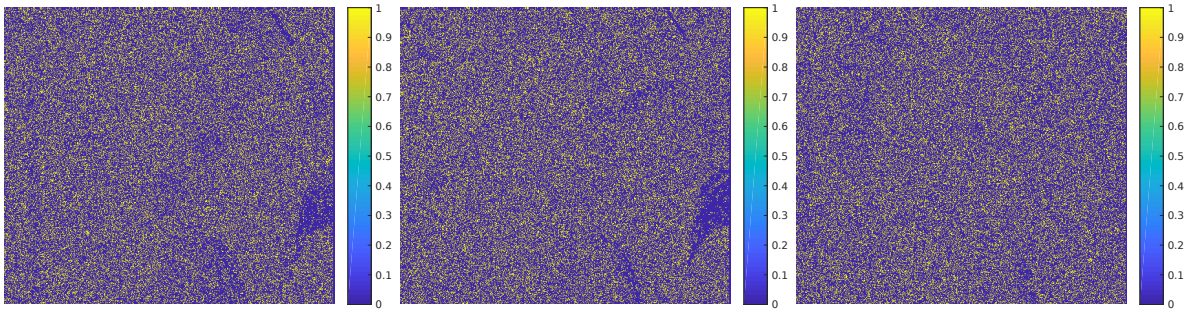
(d) Estegoimagen en crominancias.

(e) Estegoimagen 2 en espacio RGB.

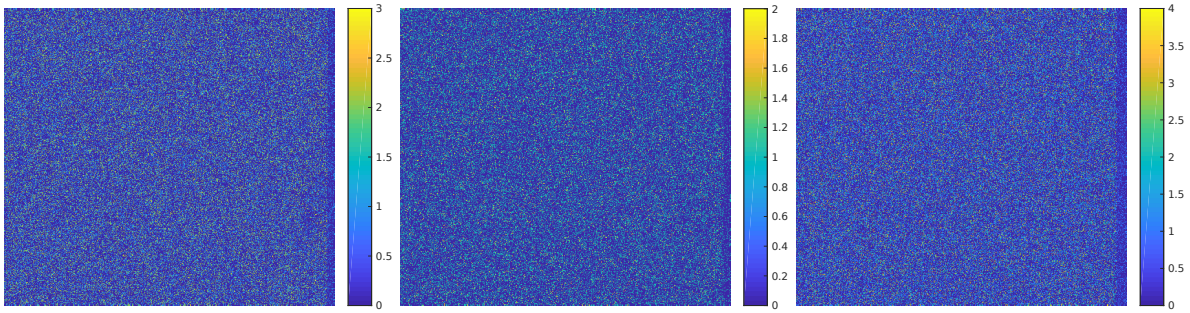
Figura 5.5: Comparativa imagen original vs. estegoimágenes color.



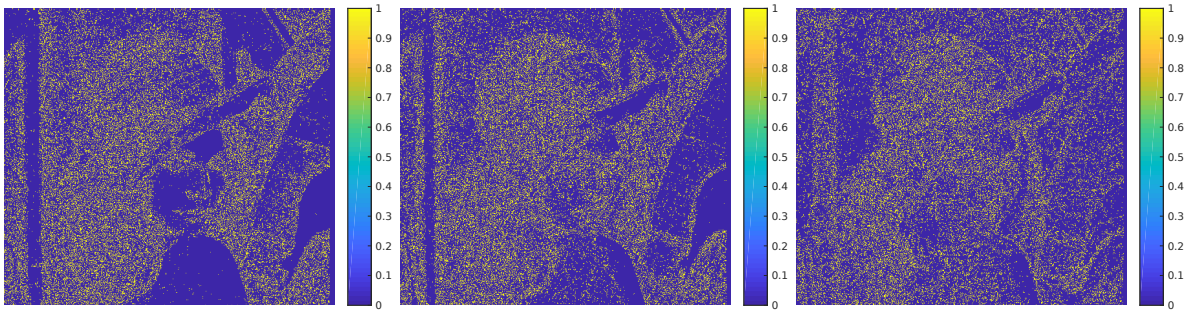
(a) Canal 1 YCBCR. (b) Canal 2 YCBCR. (c) Canal 3 YCBCR.



(d) Canal 1 RGB1. (e) Canal 2 RGB1. (f) Canal 3 RGB1.



(g) Canal 1 CBCR. (h) Canal 2 CBCR. (i) Canal 3 CBCR



(j) Canal 1 RGB2. (k) Canal 2 RGB2. (l) Canal 3 RGB2.

Figura 5.6: Imagenes diferencia color.

Esteganografiado Adaptativo en Zonas de Alta Frecuencia Espacial (cameraman)

Parámetros Esteganografiado					Evaluación		
Ventana	λ	Longitud Mensaje	Capacidad Imagen	% Cambio	% SSIM	% QILV	PSNR(dB)
3×3	0,718	500	500	0,76	100	100	48
3×3	0,655	1000	1064	1,53	100	100	45,26
3×3	0,560	2000	2079	3,05	100	100	42,21
3×3	0,365	5000	5001	7,63	100	100	38,29
3×3	0,174	10000	10035	15,26	99,99	99,99	35,19
3×3	0,0515	25000	25021	38,14	99,95	99,99	31,22
3×3	0,015	50000	50057	76,29	98,29	99,99	28,23
3×3	0,5	2936	2936	4,48	100	100	40,58
4×4	0,5	4581	4581	6,99	100	100	38,65
8×8	0,5	8448	8448	12,89	99,99	100	36,03
16×16	0,5	12290	12290	18,75	99,89	100	34,32
32×32	0,5	22220	22220	33,91	99,16	99,99	31,77

Esteganografiado Adaptativo en Zonas de Alta Frecuencia Espacial (lena)

Parámetros Modificación		Parámetros Esteganografiado				Evaluación		
Espacio	Canal	Ventana	λ	Longitud Mensaje	Capacidad Imagen	% SSIM	% QILV	PSNR(dB)
YCBCR	1	3×3	0,01838	250000	252873	98,11	99,99	23,45
YCBCR	2	3×3	0,048	250000	257499	99,13	100	25,91
YCBCR	3	3×3	0,053	250000	254761	97,99	99,99	22,1
% Cambio	95,37	**Total**		750000	765133	99,90	100	23,82
RGB	1	3×3	0,018	250000	250875	99,60	100	30,21
RGB	2	3×3	0,022	250000	250613	99,68	100	30,2
RGB	3	3×3	0,04	250000	250332	99,69	100	30,25
% Cambio	95,37	**Total**		750000	751820	99,98	100	30,22
YCBCR	1	-	-	0	0	98,63	99,99	24,91
YCBCR	2	3×3	0,048	250000	257499	99,56	100	28,92
YCBCR	3	3×3	0,053	250000	254761	98,41	99,99	23,15
% Cambio	63,58	**Total**		500000	512260	99,93	100	25,66
RGB	1	3×3	0,0399	166667	168295	99,81	100	31,99
RGB	2	3×3	0,0435	166667	167861	99,85	100	31,94
RGB	3	3×3	0,069	166667	167599	99,83	100	32,01
% Cambio	63,58	**Total**		500001	503755	99,99	100	31,98

Tabla 5.1: Datos de esteganografiado mediante modificación de LSB.

Como se puede observar en la tabla 5.1, este método ofrece una consistencia sólida con respecto a la modificación de la imagen. Si atendemos a los baremos, en especial QILV que compara la similitud basándose en varianzas locales, se podrá comprobar que incluso con tasas de cambio de píxeles superiores al 90 % la similitud estructural permanece prácitcamente intacta.

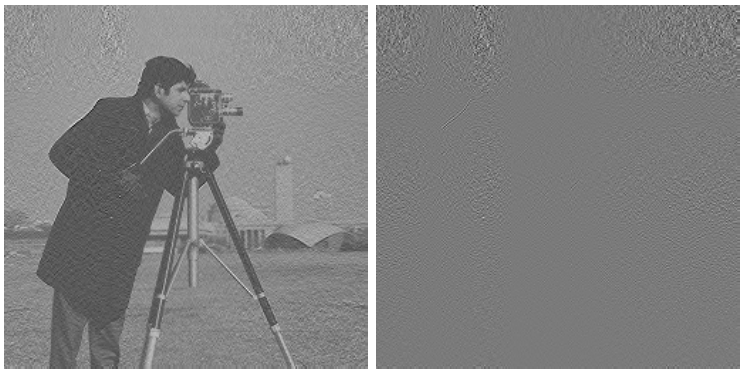
En otro orden de cosas, en este método, las imágenes no han sido modificadas utilizando un tipo `double`, sino que la modificación ha utilizado profundidades de bit de un byte. A la hora de calcular la varianza, se determinó que la aglomeración de coeficientes en torno al cero provocada por este momento central y la ausencia de *outliers* nos llevaron finalmente a utilizar la desviación típica.

5.2. Esteganografiado SVD

En esta sección se presentarán los resultados obtenidos por el esteganografiado SVD. En dicho esteganografiado, como se puede comprobar a partir de cierta longitud del mensaje, la información de imagen se pierde totalmente. Este aspecto implica un estudio previo de la imagen que, de no hacerse provocará lo visto en la imagen (e) de la figura 5.7. En este caso la recuperación de los datos no se ha hecho efectiva por excederse de las competencias de este trabajo, al desordenarse los datos incluidos por la naturaleza de dicha transformada. En cuanto a los datos presentados en la tabla 5.2, cabe destacar que el apartado «% Cambio» se refiere al porcentaje de elementos del vector diagonal ha sido modificado.

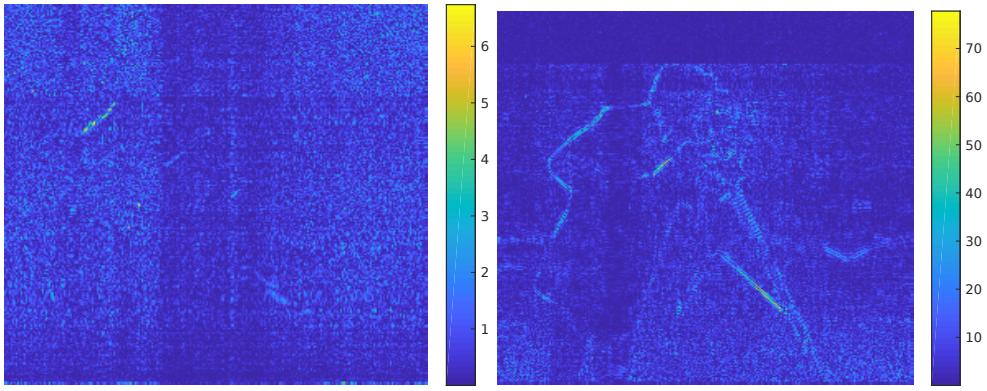
Al igual que en la sección anterior, las imágenes original y modificadas se pueden encontrar en 5.7 y las imágenes diferencia en 5.8. Los datos están volcados en la tabla 5.2.

(a) Imagen original. (b) Estegoimagen de 500 bits. (c) Estegoimagen de 1000 bits.

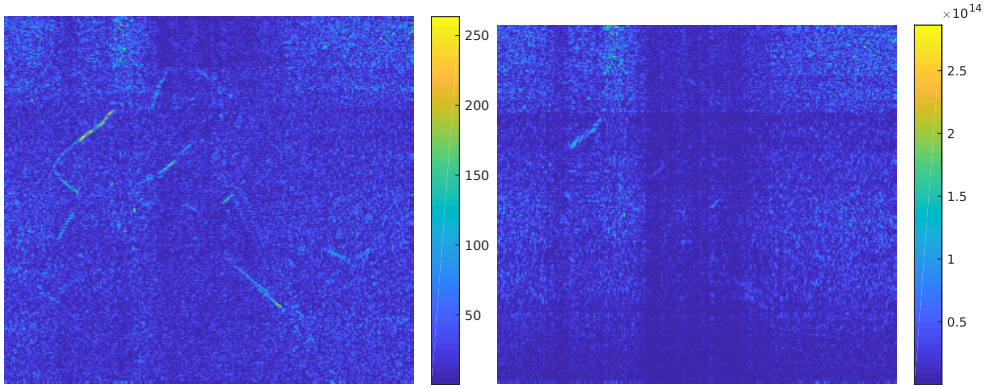


(d) Estegoimagen de 2000 bits. (e) Estegoimagen de 5000 bits.

Figura 5.7: Comparativa imagen original vs. estegoimágenes SVD.



(a) Mensaje de 500 bits. (b) Mensaje de 1000 bits.



(c) Mensaje de 2000 bits. (d) Mensaje de 5000 bits.

Figura 5.8: Imagenes diferencia SVD.

Esteganografiado SVD

Parámetros Esteganografiado				Evaluación		
Nº de huecos	Bits por hueco	Longitud Mensaje	% Cambio	% SSIM	% QILV	PSNR(dB)
100	5	500	39,06	94,04	99,99	26,68
100	10	1000	39,06	63,61	95,55	7,63
200	10	2000	78,13	31,14	25,58	-4,79
100	50	5000	39,06	0	0	-243,95

Tabla 5.2: Datos de esteganografiado basado en Descomposición en Valores Singulares.

A la luz de los datos que nos ofrece la tabla 5.2, podemos observar que este método no consigue el desempeño esperado. Todos los baremos ofreden datos desastrosos, salvo en el primer escenario, que apenas se incluyen datos.

La dificultad para embeber datos surge de la falta de espacio, dado que en este método, en detrimento de los otros dos, sólo contaremos con máx(N, M) siendo N, M las dimensiones de la imagen, dado que la matriz diagonal denominada D tendrá como máximo esos elementos. Como solución se ha optado por la inclusión de varios bits por hueco pero, dada la naturaleza de esta transformación, que reordenará en orden descendente los elementos, si incluimos demasiados la integridad de la imagen quedará completamente diluida, como se puede observar en la subfigura e de la figura 5.7.

Otro comentario al respecto de este método es que no es factible reducir la precisión de bit a un byte, dado que se perdería la información almacenada.

5.3. Esteganografiado DCT

En la presente sección comprobaremos el funcionamiento del esteganografiado basado en el algoritmo de compresión de JPG, en el que se podrán ver los resultados en las figuras 5.9 y 5.10, cuyos datos se han diseccionado en la tabla 5.3.

Cabe destacar que los datos de evaluación mostrados en la tabla están puestos en función de la transformación a imagen JPG de «cameraman», originalmente en formato TIFF. El apartado « % Cambio» se refiere al número de elementos cambiados, cuyo total se corresponde con el número de píxeles de la imagen.

Esteganografiado DCT

Parámetros Esteganografiado				Evaluación		
Nº de bloques	Bits por bloque	Longitud Mensaje	% Cambio	% SSIM	% QILV	PSNR(dB)
32	1	1024	1,56	99,13	100	45,49
32	5	5120	7,81	97,53	99,99	30,29
32	10	10240	15,63	96,36	99,95	24,04
32	35	35840	54,69	88,35	93,86	7,88

Tabla 5.3: Datos de esteganografiado basado en compresión JPG.

(a) Imagen original.

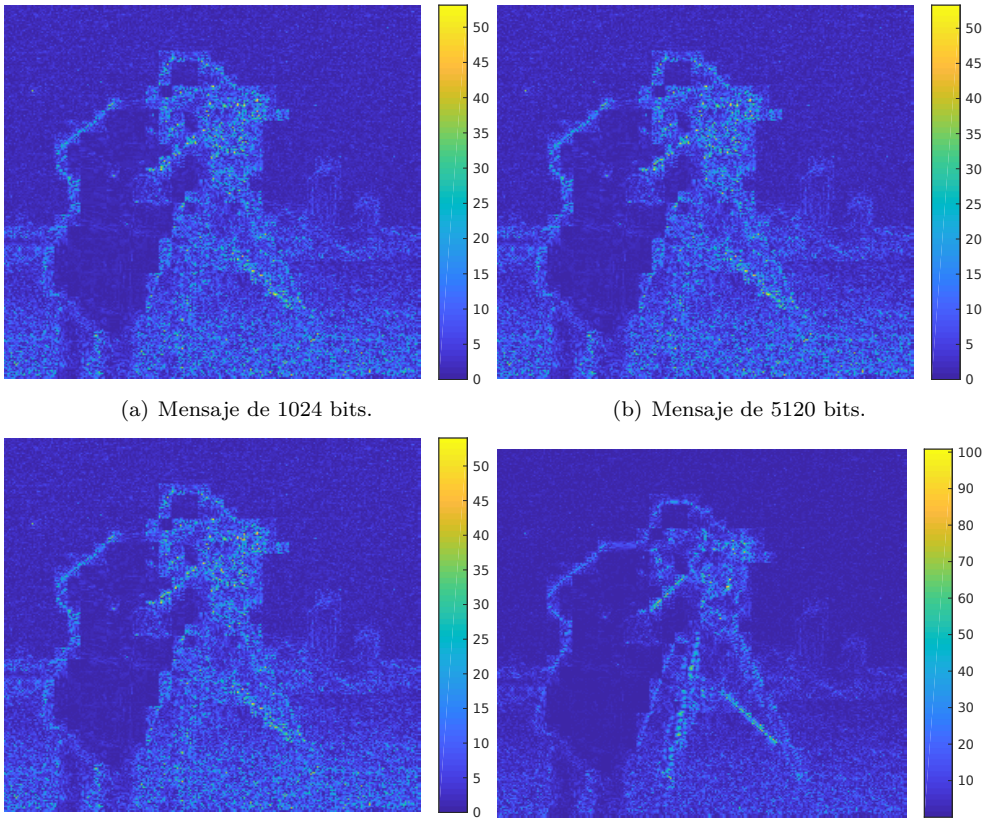
(b) Estegoimagen de 1024 bits.

(c) Estegoimagen de 5120 bits.

(d) Estegoimagen de 10240 bits.

(e) Estegoimagen de 35840 bits.

Figura 5.9: Comparativa imagen original vs. estegoimágenes DCT.



(a) Mensaje de 1024 bits.

(b) Mensaje de 5120 bits.

(c) Mensaje de 10240 bits.

(d) Mensaje de 35840 bits.

Figura 5.10: Imagenes diferencia DCT.

Los resultados del último método propuesto, reflejados en la tabla 5.3, ofrecen una tasa de similitud mediante QILV y SSIM nada desdeñable, dado que cambiando un 54,69 % de los coeficientes transformados, obtiene unas cotas cercanas al 90 % en el caso de SSIM y superiores en QILV.

Al igual que en el método basado en SVD, en éste tampoco hemos sido capaces de reducir la profundidad de bit a un byte, por lo que no podremos formatearlo como formato usual. El principal valor de este método de esteganografiado se encuentra en la tranformación utilizada, la DCT, presente en el formato JPEG, uno de los más extendidos por su alta tasa de compresión.

Dado que Internet está repleto de imágenes formateadas como JPEG, el esteganografiado de código ejecutable capaz de explotar una vulnerabilidad en el *parser* o intérprete de la imagen, podría comportar un método consistente con este fin, pero dada la imposibilidad de reducir la profundidad de bit, hará falta un perfeccionamiento.

5.4. Estegoanálisis

En cuanto a los resultados obtenidos del estegoanálisis, los resultados se pueden ver en la tabla 5.4. Como se puede observar en la tabla 5.4, el primer método propuesto tiene una robustez frente a ataques

Estegoanálisis: texto claro con mensaje desconocido

Límite inferior	Límite Superior	Paso	% Éxito	Tiempo por intento
0,7	0,8	0,0001	0,6	10,6 ms

Estegoanálisis: texto claro con mensaje conocido

Límite inferior	Límite Superior	Paso	% Éxito en binario	Éxito en ASCII	Tiempo por intento
0,7	0,8	0,0001	2,8	0,2	11,3 ms

Tabla 5.4: Datos de estegoanálisis.

de fuerza bruta de esta índole, habiendo simplificado en este caso la elección de los parámetros.

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

Una vez vistos los resultados y, teniendo en cuenta los objetivos propuestos en la introducción, podemos llegar a las siguientes conclusiones:

- El método con mejores resultados es el que se presentó en primer lugar, como se puede ver en las tablas del capítulo 5. Se puede ver que el volumen de datos apenas afecta a la calidad de la imagen aun ocupando más del 90 % como se puede ver en las pruebas hechas con la foto de *Lena*. Cabe destacar que es la única implementación que se puede llevar a cabo de forma práctica, dado que las siguientes no consiguieron pasar a profundidad de un byte, lo que no les permite ser formateados como ficheros normales.
- Los objetivos de esteganografiado se han llevado a cabo con excelente resultado, aunque no así los objetivos propuestos en materia de estegoanálisis, materia en la que nos hemos limitado a comprobar la robustez de nuestro método de esteganografiado. Queda pendiente entonces el desarrollo de un detector eficiente para un posterior trabajo.
- Para clausurar las conclusiones, cabe destacar que la consecución de ocupar más del 90 % de una imagen a todo color con un mínimo impacto supone la capacidad de introducir una cantidad ingente de bits, por lo que sería posible introducir el código de un *malware*, con lo que se confirma aquello dicho por [Bol17].

6.2. Líneas futuras

En nuestro primer método, esteganografiado adaptativo en zonas de alta frecuencia espacial, en las líneas futuras debemos incluir un algoritmo que calcule automáticamente ambos parámetros, umbral λ y tamaño de ventana, para una mejor adaptación a la longitud del mensaje. Otra mejora que habría que buscar es la capacidad de proteger el esteganografiado ante posibles compresiones con pérdidas.

En el método basado en la transformada del coseno, la prioridad sería encontrar la forma de reducir la profundidad a un byte, así como la capacidad de modificar imágenes cuyo número de filas y/o columanas no sea múltiplo de 8.

En cuanto al método basado en la descomposición en valores singulares, se podría buscar la creación de un algoritmo capaz de transformar la estegoimagen a profundidad de un byte, así como la mejora ya

adelantada en la discusión de los métodos, que sería la de regular la cantidad de bits por símbolo para conseguir hacer que la magnitud del estegovalor que sustituirá al valor original de la matriz diagonal, consiguiendo que el impacto visual, previsiblemente, mejore notablemente.

En cuanto al estegoanálisis, queda pendiente la creación de un detector capaz de detectar diferentes tipos de esteganografiado.

Bibliografía

[AF06] S. Aja-Fernandez, R. S. J. Estepar, C. Alberola-Lopez y C.-F. Westin. Image quality assessment based on local variance. En *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, págs. 4815–4818. IEEE, 2006.

[Bol17] Bollero, David. Si ves estas imágenes iguales, te colaron un virus, agosto de 2017.

[Cio18] R. Ciogranne. Determining JPEG Image Standard Quality Factor from the Quantization Tables, febrero de 2018.

[Jai89] A. K. Jain. *Fundamentals of digital image processing.* Englewood Cliffs, NJ: Prentice Hall,, 1989.

[Kra07] N. Krawetz. A Picture's Worth, Digital Image Analysis and Forensics, 2007.

[Mar00] Martin Oberzalek. Enigma - a very famous story of cryptology, marzo de 2000.

[MM97] A. L. Martín Marcos. Compresión de imágenes. norma JPEG. 1997.

[Rea17] Real Academia Española. Diccionario de la lengua española, 2017.

[Sta17] Stallings, William and Brown, Lawrie. *Computer Security: Principles and Practice.* Pearson, cuarta edición, agosto de 2017.

[Vij17] J. Vijayan. Steganography Use on the Rise Among Cyber Espionage, Cybercrime Groups, agosto de 2017.

[Vij18] J. Vijayan. Memes on Twitter Used to Communicate With Malware, diciembre de 2018.

[Wal92] G. K. Wallace. The JPEG Still Picture Compression Standard. IEEE Transactions on Consumer Electronics, Vol. 38, Num. 1, febrero de 1992.

[Wan04] Z. Wang *et al.* Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, Vol. 13, Num. 4, abril de 2004.

[Wes01] A. Westfeld. F5—a steganographic algorithm. En *International workshop on information hiding*, págs. 289–302. Springer, 2001.

[Zha05] X. Zhang y S. Wang. Steganography using multiple-base notational system and human vision sensitivity. *IEEE Signal Processing Letters*, volumen 12, nº 1, págs. 67–70, Jan de 2005.