

UNIVERSIDAD DE MURCIA

COMPRESIÓN MULTIMEDIA

PCEO

Compresión de Imágenes Basada en DCT: Estudio Experimental

Autor:

Jose Antonio Lorencio Abril
(joseantonio.lorencioa@um.es)

Profesor:

Jesús Damián Jiménez Re

Convocatoria Enero 2021/22

Índice

1. Resumen	4
2. Introducción	5
3. Metodología	7
3.1. Imágenes seleccionadas	7
3.1.1. ejemploMulticolor	7
3.1.2. windows	8
3.1.3. leon	8
3.1.4. captura	9
3.1.5. bmp_24	9
3.1.6. dots	10
3.1.7. blackbuck	10
3.1.8. greenland_grid_velo	11
3.1.9. snail	12
3.1.10. lena	12
3.2. Obtención de datos	13
3.3. Comparación cuantitativa	13
3.4. Comparación cualitativa	13
3.5. Software utilizado	13
3.6. Características técnicas del hardware	13
4. Resultados experimentales	14
4.1. Estudio Cualitativo	14
4.1.1. Conclusión de la comparación cualitativa	27
4.2. Estudio Cuantitativo	28
4.2.1. Resultados medios	38
4.2.2. ¿Qué sucede con las imágenes png?	39

5. Conclusión	40
A. Codificación Huffman	42

1. Resumen

En el presente trabajo se analizará el funcionamiento de dos compresores desarrollados en la asignatura de Compresión Multimedia.

Tenemos dos pares compresor/descompresor con los que comprimiremos y descomprimiremos un conjunto de imágenes seleccionadas. A partir de este proceso obtendremos los siguientes datos relativos al proceso estudiado:

- **Ratio de Compresión (RC):** mide cuánto se ha comprimido la imagen. Compara, por tanto, el tamaño final del fichero con el tamaño original:

$$RC = \frac{T_{Orig} - T_{Final}}{T_{Orig}}$$

Por lo que este es un valor que buscamos maximizar.

- **Error cuadrático medio (MSE):** es una forma de establecer una distancia entre imágenes¹. Se supone que dos imágenes cercanas en esta distancia deben ser muy parecidas, aunque muchas veces observamos que una imagen con mayor MSE que otra, se parece más a la primera. No obstante, es de evidente utilidad práctica, por lo que es común su uso. Este valor querremos minimizarlo.
- **Ratio de Señal a Ruido (SNR):** es otra medida de semejanza entre imágenes, pero esta no se basa en la distancia, sino que compara la cantidad de señal codificada por cada unidad de error obtenida. Este valor buscamos maximizarlo, pues la cantidad de señal por cada unidad de error querremos que sea lo mayor posible.

Usaremos, por tanto, estos valores para realizar el estudio sobre los dos compresores, aplicados sobre las diferentes imágenes.

¹En realidad, el MSE es un concepto más general de distancia, no solo para imágenes. Su nombre viene de que se usa el cuadrado de la diferencia entre los valores buscados y los obtenidos en el proceso que estamos modelando. Para una discusión detallada sobre el MSE puede consultar [5]

2. Introducción

Desde la prehistoria, la imagen ha sido importante para el ser humano. Se estima que hace aproximadamente 900.000 años que hubo gentes muy parecidas a nosotros en Atapuerca ([1]), donde dejaron su legado a través de imágenes dibujadas sobre las paredes de las cuevas. Desde entonces, no ha habido ningún período histórico en que la imagen no fuese un punto central en las civilizaciones humanas. No obstante, ha sido siempre un bien escaso, hasta el siglo pasado.

En el siglo XX la producción mundial de imagen aumentó en varios factores de magnitud, y la razón es la evidente: el desarrollo de la computación, que implicó la invención de la imagen digital, una novedosa forma de hacer y mantener imágenes: usando una cadena ordenada de bits. Una industria entera creció en torno a este simple concepto, y el aumento de la calidad y la cantidad de las imágenes hicieron que la cadena de bits necesitada fuese desorbitada².

Se idearon entonces diferentes formas de codificar la información, intentando transmitir la máxima cantidad posible de esta, con la mínima cantidad posible de espacio. Intuitivamente, a modo de ejemplo, podríamos pensar que queremos transmitir la máxima cantidad posible de información acerca de un número, con la cantidad mínima de caracteres utilizados. En este sentido, la numeración en base 10 es mejor que la codificación binaria, por ejemplo, el número 8 necesita únicamente un carácter para ser transmitido en decimal, mientras que necesita cuatro en binario ($8_{10} = 1000_2$). Llegamos así hasta la actualidad, donde las técnicas de codificación de la información son muy avanzadas y con impresionantes resultados.

Los codificadores utilizados en este trabajo se basan en la codificación Huffman³. Esta técnica utiliza un alfabeto de longitud variable, que básicamente asigna a los términos codificados las *letras* (elementos del alfabeto) más cortos en orden de cantidad de apariciones. Además, nuestros compresores se valen de unas tablas que representan los alfabetos, disponiendo de dos tipos:

- El **compresor por defecto** se llama así porque utiliza unas tablas Huffman estándar, que se aplican sin variación a cualquier imagen que queramos codificar.
- Por otro lado, el **compresor customizado** se vale de unas tablas customizadas, que varían en función de las características de la imagen codificada.

Además, en el proceso se aplica la **transformada discreta del coseno**⁴, que consiste en transformar la imagen a un espacio discreto que reduce la dimensionalidad. Además, el resultado de este proceso se cuantiza, aplicando un filtro que elimina ciertos coeficientes y que depende del valor del factor de calidad pasado como parámetro. A mayor valor de este factor, mayores deben ser los coeficientes para pasar el filtro. Esta eliminación debe tratar de maximizar la cantidad de coeficientes descartados, maximizando también el parecido de la imagen reconstruida a la original.

Como se deduce del párrafo anterior, la codificación que usamos es una **codificación con pérdidas**: su aplicación entraña pérdida de información debido al proceso de cuantización/descuantización. En este tipo de codificaciones, el resultado obtenido difiere del original, por lo que es de extrema importancia estudiar si la diferencia obtenida en el proceso hace que los resultados sean inservibles. De nada sirve poder comprimir mucho un archivo, si no somos capaces de obtener de vuelta el archivo original (o algo razonablemente similar).

Así, utilizaremos para este estudio ambos compresores sobre diferentes imágenes y comprobaremos los resultados

²Esto sucedió no solo en el campo de la imagen, sino en cualquier campo que requiriese la codificación de información: el texto, la imagen, el vídeo, y los ficheros informáticos en general.

³Para una explicación detallada, ver A.

⁴Para una explicación en profundidad, puede consultarse [2]

obtenidos, en busca de una configuración, o al menos un conjunto de configuraciones, que produzca resultados aceptables de forma generalizada.

3. Metodología

Lo primero es seleccionar las imágenes a analizar. Mis pretensiones eran tener gran diversidad tanto de tipos de imágenes como de sus tamaños.

Una nota importante es que todas las imágenes las tengo tanto en PNG como en BMP, pues me parecía interesante analizar si el cambio de formato implicaba alguna diferencia con respecto a nuestros compresores. Algunas de las imágenes son, entonces, BMP transformados a PNG, o viceversa. Esto no debería de suponer ningún problema (al contrario que sucedería con JPG), pues ambos formatos son sin pérdidas, y puede interpretarse el cambio de uno a otro simplemente como una traducción del "lenguaje" de codificación utilizado en cada caso.

3.1. Imágenes seleccionadas

3.1.1. ejemploMulticolor

La primera imagen que he seleccionado fue una de las que cree con Paint para la entrega parcial. La imagen es la siguiente:

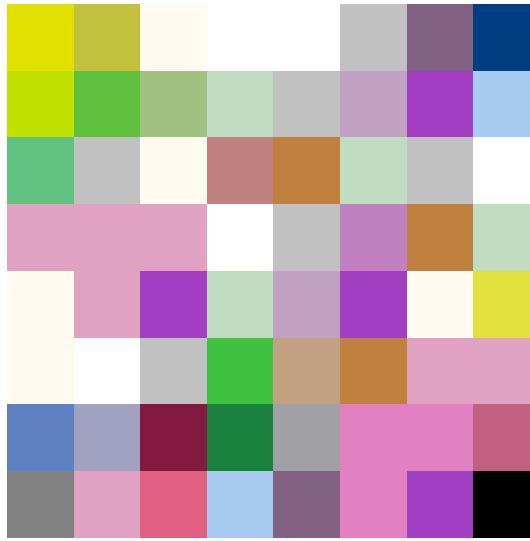


Figura 1: ejemploMulticolor

Respecto a las características técnicas de la imagen, es una imagen de 8x8 píxeles, que ocupa 1.11KB y fue creada por mí con Paint, y guardada en formato bmp de 24bits. La imagen en PNG, por su parte, ocupa 260B, bastante menos que la BMP (como es esperable, pues PNG comprime imágenes BMP sin pérdidas).

Esta imagen presenta distintos colores dispuestos de forma aleatoria. Es esperable que los resultados obtenidos al comprimir/descomprimir esta imagen sean muy malos: para mantener buena calidad, comprimiremos muy poco, pues la baja correlación de los píxeles de la imagen obliga a necesitar utilizar (prácticamente) todos los coeficientes de la transformada. Si lo que queremos es alcanzar una buena cota de compresión, la calidad se verá muy mermada. Por otro lado, el SNR

La imagen puede verse en [/source/Images/Original/ejemploMulticolor.\(bmp o png\)](/source/Images/Original/ejemploMulticolor.(bmp o png)) .

3.1.2. windows

Esta imagen consiste en una captura de pantalla del fondo de windows:

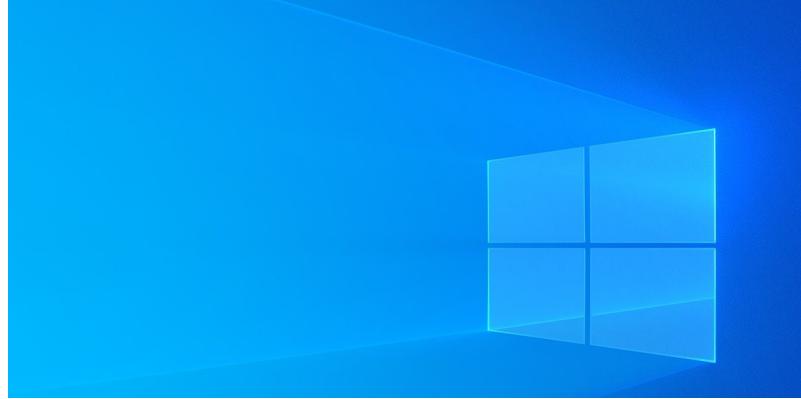


Figura 2: windows

Esta imagen es de 1212x603, una imagen bastante grande, que ocupa 2.09MB en su versión BMP y 440KB en formato PNG.

El gran tamaño de la imagen, en conjunción con la baja variabilidad que se puede observar a simple vista en los colores de la imagen, son indicadores de que podemos esperar altas cotas de compresión y buenos valores del MSE y del SNR.

3.1.3. leon

Esta es otra captura de pantalla de una imagen de google (la forma más sencilla que encontré de coger una imagen concreta en formato png o bmp fue mediante capturas de pantalla, de cualquier otra forma no podía estar seguro de si una imagen era realmente nativa del formato en que estaba subida a internet).



Figura 3: leon

Es de 775x436, por lo que se podría considerar de tamaño medio, y ocupa 991KB (bmp) y 733KB (png). Observamos que en esta ocasión png no es capaz de comprimir tanto como con la anterior. Esto tiene mucho

sentido, ya que esta imagen presenta varias zonas diferenciadas, y una mayor variabilidad. Así, es esperable que la compresión sin pérdida de información no pueda ser muy grande.

En nuestro sistema, de compresión con pérdidas, esperaremos, razonando de forma similar, que el MSE crezca rápidamente. No obstante, es probable que incluso en una imagen con tantos detalles como esta, y tal como se vio en la entrega parcial, un MSE alto no implique unas grandes diferencias perceptuales respecto a la imagen original.

3.1.4. captura

Esta imagen es una captura del menú de Matlab. Me parecía interesante añadirla porque ofrece un fondo constante, sobre el que encontramos diferentes elementos que requieren de gran nivel de detalle, como palabras y símbolos.

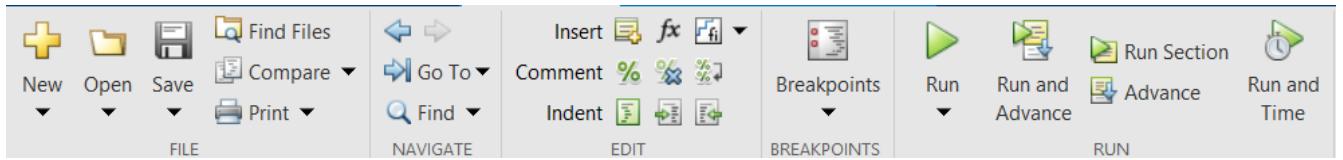


Figura 4: captura

Es de 992x113 y ocupa 328KB (bmp) y 35.6KB (png).

Como la imagen es bastante plana, no es descabellado pensar que obtendremos buenos resultados de compresión. No obstante, las letras y símbolos son un factor importante que pueden dificultar la tarea. Es difícil hacer un pronóstico en este caso.

3.1.5. bmp_24

Esta y las siguientes imágenes han sido obtenidas de [4]. Esta corresponde a la imagen con el mismo nombre que se encuentra en esa página. Como vemos, proporcionan tanto el png como el bmp, por lo que es idóneo con la línea del trabajo buscado.

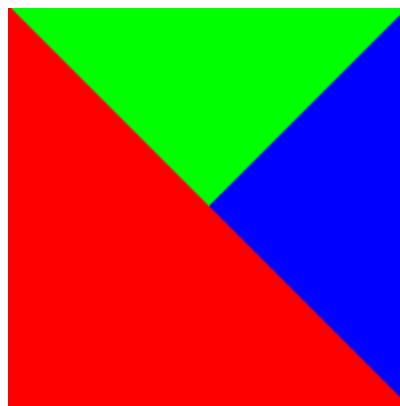


Figura 5: bmp_24

Esta imagen de 200x200 (117KB el bmp y 1.38KB el png) puede parecer muy sencilla, pero las diagonales presentan cambios bruscos de color, lo que puede resultar en algunas complicaciones.

3.1.6. dots

Imagen obtenida del mismo directorio que la anterior.

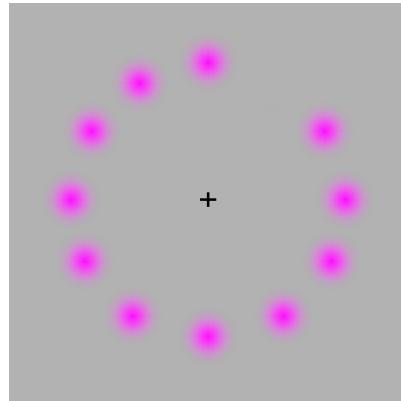


Figura 6: dots

Tiene un tamaño de 467x467 píxeles y ocupa 214KB (bmp) y 5.29KB (png).

Esta imagen presenta únicamente dos colores, y los cambios de uno a otro se hacen de forma muy suavizada. Por tanto, podemos esperar buenos resultados de compresión con baja pérdida de calidad.

3.1.7. blackbuck



Figura 7: blackbuck

La imagen de esta gacela tiene un tamaño de 512x512 y ocupa 768KB (bmp) y 159KB (png) en memoria.

La imagen original ya presenta pequeñas deformidades y es de baja calidad, por lo que es probable que en este caso podamos obtener aparentemente malos resultados teóricos, pero que estudiados cualitativamente no impliquen grandes discrepancias visuales.

3.1.8. greenland_grid_velo

La siguiente imagen me parece muy interesante:

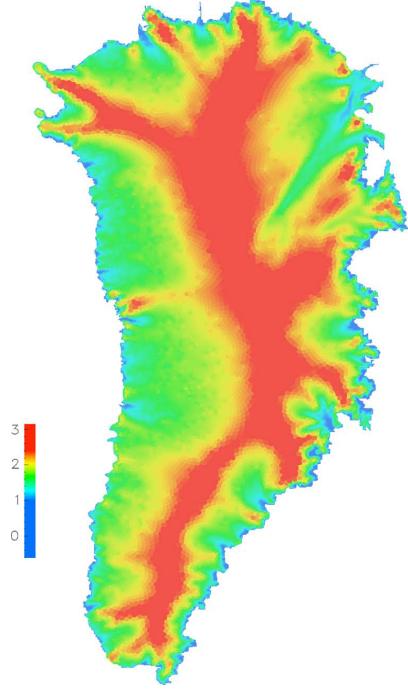


Figura 8: greenland_grid_velo

Es de 762x1309 píxeles (bastante grande) y ocupa 2.85MB (bmp) y 634KB (png).

Me parece de gran interés esta imagen, pues presenta una amplia gama cromática, y aunque la mayoría de cambios de tono se hacen de forma gradual, es evidente que se efectúan algunos cambios bruscos. Además, la leyenda de la izquierda, presenta también números, de forma que hay muchos efectos que pueden resultar en malos resultados. Por otro lado, el gran tamaño de la imagen sabemos que es una ventaja, pues localmente los píxeles serán muy parecidos. Yo diría que en este caso los resultados visuales serán muy buenos, y los teóricos serán también buenos, hasta cierto momento en que el MSE será demasiado alto (y el SNR será demasiado bajo).

3.1.9. snail

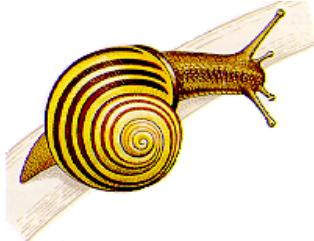


Figura 9: snail

Esta pequeña imagen de 256x256 píxeles ocupa 192KB en formato bmp y 32.7KB en formato png.

Hay pocos colores y la mayoría de la imagen es blanca. Esto puede hacernos pensar que los resultados obtenidos serán buenos en cuanto a compresión, ya que habrá varios bloques totalmente blancos que ocuparán muy poco. No obstante, el MSE no está tan claro que vaya a ser bueno, ni tampoco los resultados cualitativos.

3.1.10. lena

Por último, incluimos la imagen de Lena, utilizada clásicamente en estudios de compresión de imágenes y que es, de hecho, un estándar de facto en la industria⁵.



Figura 10: lena

Esta famosa imagen es de 512x512 y ocupa 257KB (bmp) y 311KB (png). Nótese este extraño fenómeno... ocupa más la versión comprimida con png. Y es que esta imagen es conocida por ser muy compleja de comprimir.

Así, es probable que obtengamos unos resultados malos al comprimir con pérdidas esta imagen, que a simple vista podría parecer sencilla, ya que es en blanco y negro y no se aprecian extrañezas significativas. Veremos qué sucede.

⁵Para más información ver [6]

3.2. Obtención de datos

Para cada una de estas imágenes he aplicado el compresor y el descompresor desarrollado en la asignatura. Tanto el que usa las tablas por defecto de codificación, como el que calcula y utiliza las tablas customizadas.

Además, este proceso se ha realizado en cada foto para ambos formatos (bmp y png), y para valores del parámetro de calidad 1, 25, 50, 100, 250, 500 y 1000. He elegido estos valores para tener un amplio rango de resultados, que además fuese desde factores que implican muy poca pérdida hasta factores muy grandes que implicarán una gran disminución de la calidad de imagen, pero grandes tasas de compresión.

3.3. Comparación cuantitativa

Para cada uno de estos valores del factor de calidad y cada imagen disponemos, entonces, de los valores del ratio de compresión (RC), del error cuadrático medio (MSE) y del ratio señal/ruido (SNR).

Estos valores son los que analizaremos en cada caso, comparando los obtenidos mediante la compresión por defecto y la customizada para cada foto. Además, se compararán los resultados de algunas fotos entre sí, los que parezcan más significativos.

Este estudio cuantitativo es evidentemente importante, ya que mediante mediciones y hechos cuantitativos es la única forma en la que podemos valorar de forma generalizada cómo funciona nuestro compresor. Sabemos, claro está, que no solo esto es importante, pues también buscamos que las imágenes finales se parezcan visualmente a las originales, lo que no siempre sucede aunque estas se encuentren 'cerca' de manera cuantitativa.

3.4. Comparación cualitativa

Para este último respecto realizaremos un análisis cualitativo de las imágenes producidas con distintos factores de calidad. Los factores elegidos serán 50, 100 y 250, que pueden considerarse valores que producirán imágenes muy similares a la original (50), algo similares a la original (100) y que entrañan una pérdida significativa de información (250), de forma generalizada.

3.5. Software utilizado

Para el desarrollo de los compresores/descompresores, y para la toma de datos he utilizado *Matlab R2020a*, instalado sobre Windows 10 Home.

Para el análisis de los datos obtenidos he exportado estos a formato excel y he utilizado el software de *Google Sheets* para la obtención de las gráficas comparativas.

3.6. Características técnicas del hardware

En este tipo de estudios es importante considerar el hardware sobre el que trabajamos, pues los resultados son altamente dependientes del mismo. En mi caso, el trabajo ha sido realizado sobre un procesador *Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz*, con 20,0 GB de memoria RAM instalada.

4. Resultados experimentales

Vamos a comenzar realizando la comparación cualitativa, ya que esta se encuentra sujeta a la subjetividad del experimentador (en este caso yo), y su realización tras el análisis cuantitativo entrañaría problemas de sesgo de confirmación, los cuales encuentro menos probables si el proceso se realiza al contrario.

4.1. Estudio Cualitativo

Para la comparación cualitativa, vamos a considerar, para cada imagen seleccionada, las imágenes generadas por el proceso de compresión/descompresión correspondientes a los valores del factor de calidad 50,100 y 250. Esto lo haremos para ambos codificadores.

ejemploMulticolor

En la figura 11 observamos la imagen original junto con los resultados obtenidos. Esta imagen es difícil de analizar, pues es originalmente muy pequeña y al expandirla es extraña. No obstante, si nos fijamos, observamos que con factor de calidad 50 no se aprecian grandes diferencias, algunos colores están más claros y oscuros que en la original, pero nada parece fuera de lugar. Observamos el mismo suceso para ambas codificaciones, default y custom. Este fenómeno será recurrente a lo largo del trabajo: se observará que los resultados visuales obtenidos por ambos compresores son prácticamente idénticos.

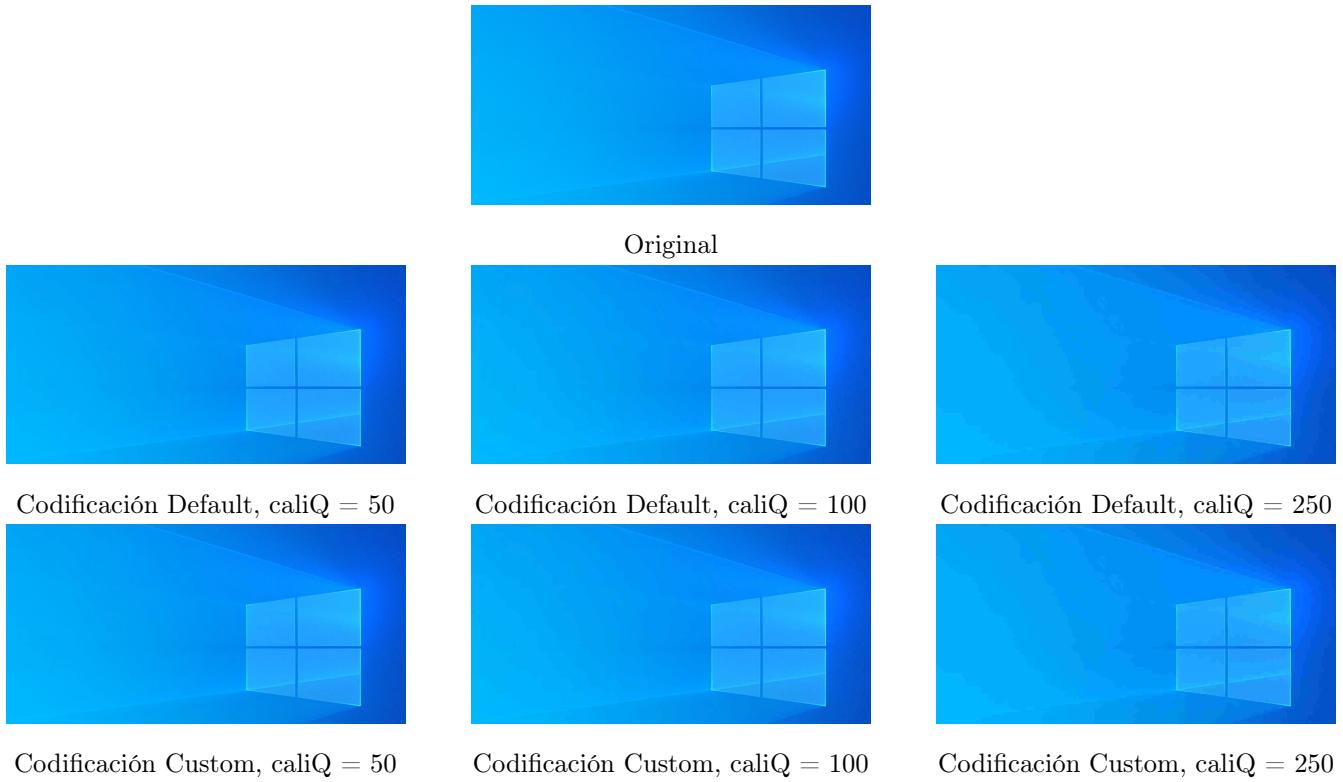
Pasando al factor de calidad 100, se ven ya diferencias notables con la original, aunque si esto fuese un pequeño bloque que forma parte de una imagen mucho mayor, no es descabellado pensar que la diferencia continuase siendo imperceptible.

Esto no sucede ya cuando pasamos al factor 250, con el que la imagen solo mantiene algunas tonalidades por zonas respecto a la original, pero que ciertamente es una imagen completamente distinta.

Comentar, además, que en este y todos los siguientes casos, no existe diferencia visual entre los resultados obtenidos con las imágenes bmp y las png. Este hecho no lo repetiremos.

En este caso, por lo tanto, un factor de calidad de 50 parece obtener buenos resultados visuales.

Figura 11: Aplicación a *ejemplomulticolor.bmp*

windowsFigura 12: Aplicación a *windows.bmp*

En la figura 12 se observa que para los factores de calidad 50 y 100 la diferencia es imperceptible. Para el factor 250 ya si se visualizan ondulaciones que no existen en la imagen original, pero sigue siendo razonablemente parecida al resultado idóneo. Esto ya lo comentábamos al seleccionar la imagen: esta es muy sencilla y es esperable que puedan usarse mayores factores de calidad sin que suponga una fuerte pérdida de calidad.

De nuevo, observamos la igualdad práctica entre los resultados con el codificador por defecto y el customizado.

Para este caso, entonces, parece sensato elegir un valor de 100 para el factor de calidad como el idóneo. El razonamiento es muy simple, si nos encontramos ante varios resultados que a la vista parecen iguales, debemos elegir aquel que más comprima el archivo. Como hemos comentado anteriormente, el factor de calidad es un valor que tratamos de maximizar, pues hace que la compresión sea mayor. La conclusión comentada es entonces la lógica, si nos basamos únicamente en la percepción visual.

leon

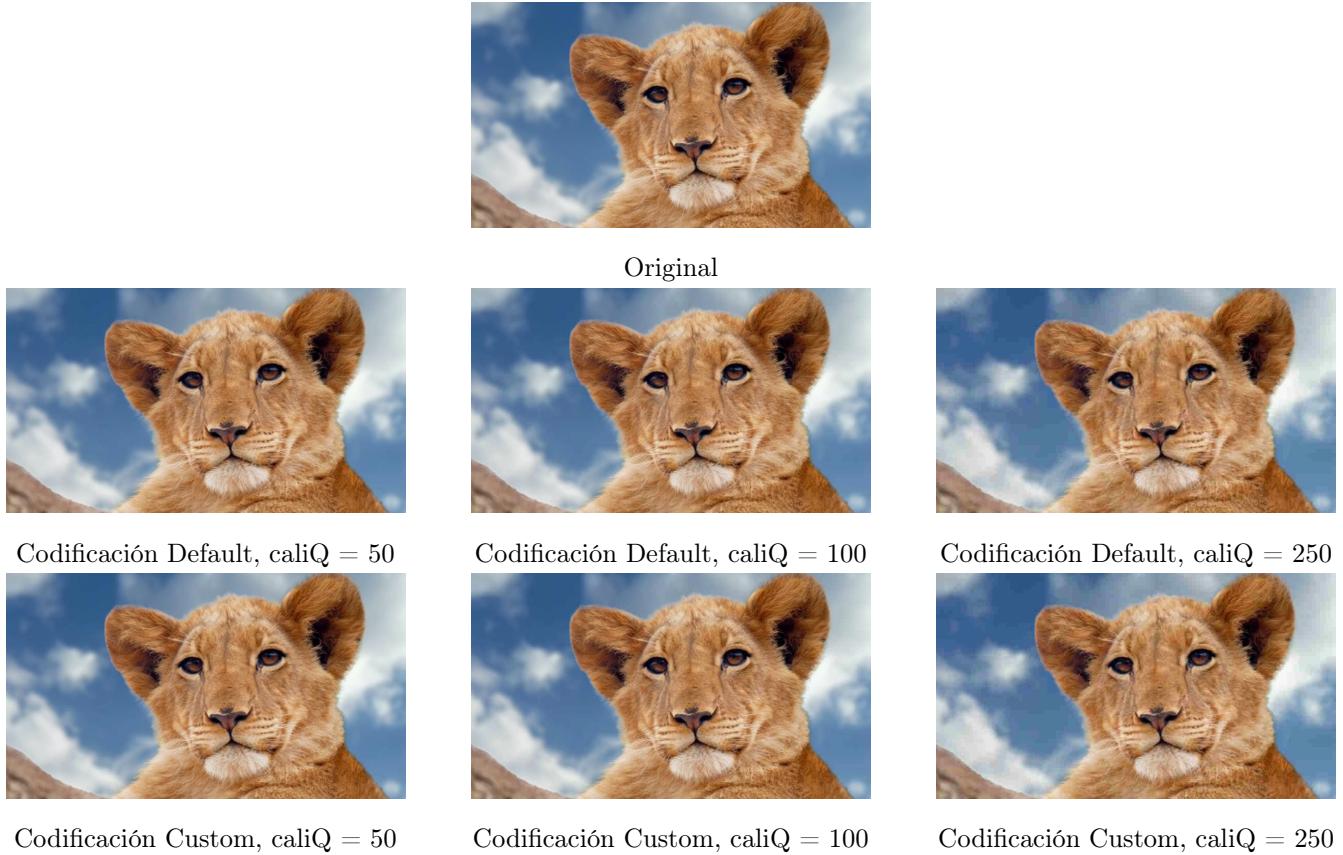


Figura 13: Aplicación a *leon.bmp*

En esta ocasión las imágenes recuperadas respetan muchísimo a la original. Lo único que se observa es un efecto bruma que casi no se nota para el factor 100, pero que para el factor 250 se torna evidente.

Como viene sucediendo, los resultados default y customizados son idiferenciables. Este aspecto no se volverá a comentar, a no ser que sea relevante.

De nuevo, elegir el factor 100 parece ser una buena decisión, pues si bien con 250 se sigue interpretando bien la imagen, se pierde mucha calidad, seguramente demasiada.

captura

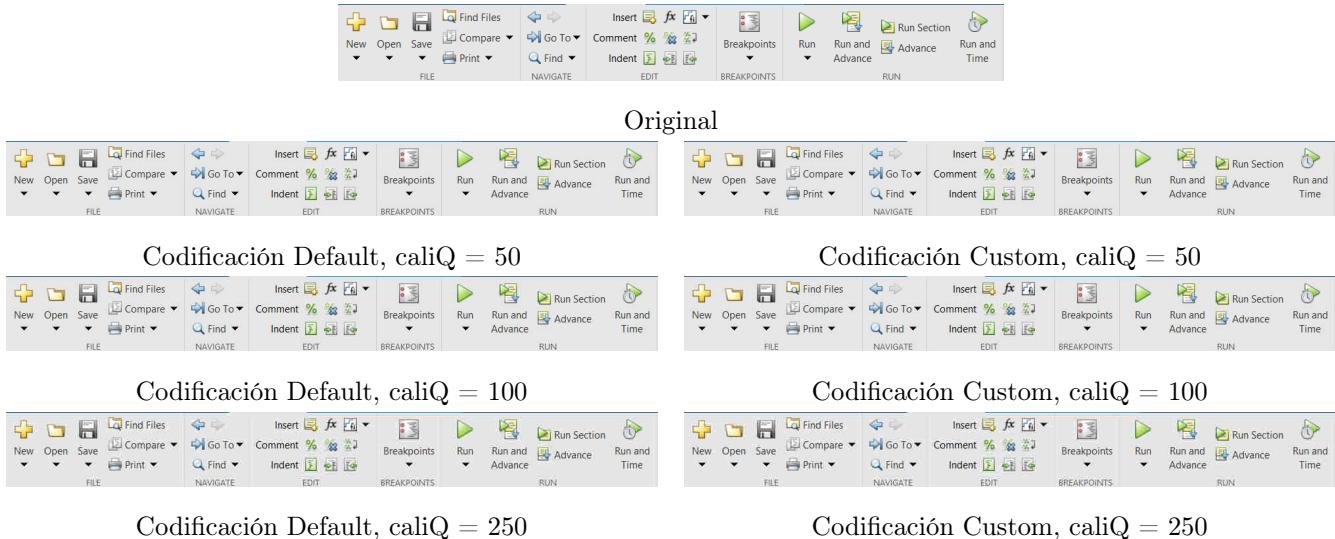


Figura 14: Aplicación a *captura.bmp*

Esta imagen es muy interesante. Como observamos⁶, casi todos los detalles son captados correctamente incluso utilizando el factor 250. No obstante, este parece el prototípico caso en el que basta que haya algo de pérdida de información para que el proceso no pueda considerarse correcto. Aquí, la palabra *FILE*, abajo a la izquierda pierde mucha definición, hasta tal punto que es posible que personas con dificultades visuales no puedan leerlo. Así, aunque en general capta bien los detalles de la imagen, este hecho aparentemente insignificante, junto con que los iconos aparecen algo difuminados, mearía decantarme por utilizar el factor 100.

⁶ Esta imagen recomiendo visualizarla en el fichero original, ya que en la presente memoria no se capta bien lo que comento. Puede encontrarla en *source/Images/EstudioCualitativo/dft*

bmp_24

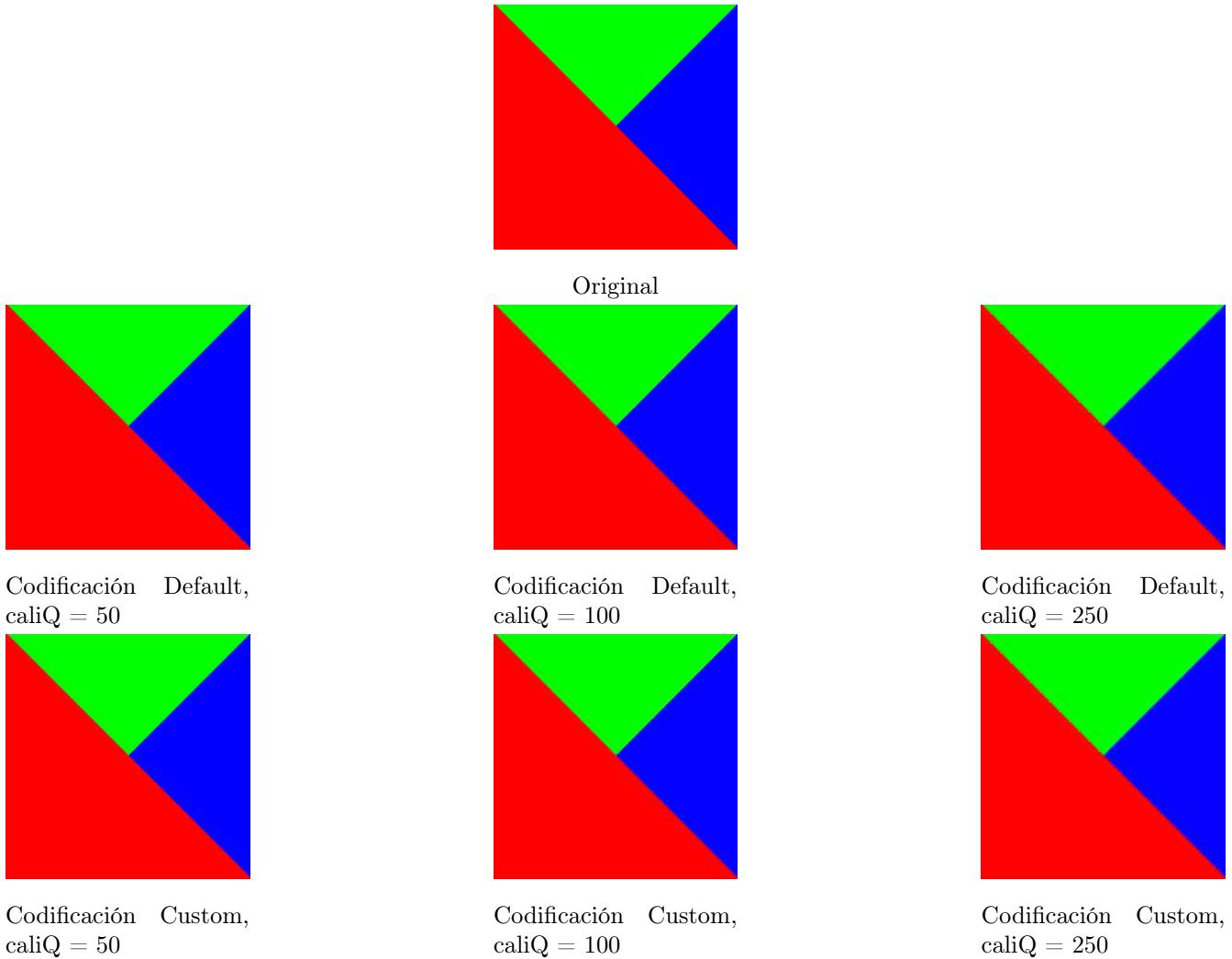
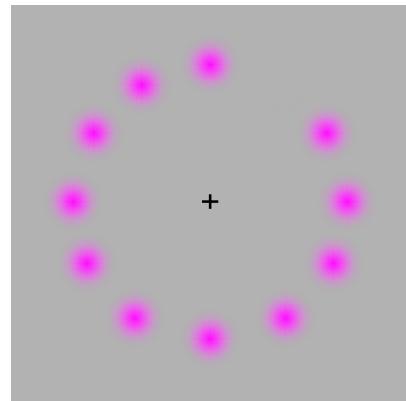


Figura 15: Aplicación a *bmp_24.bmp*

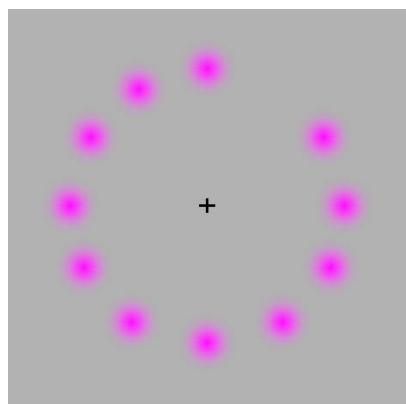
En este caso todos los factores producen un resultado muy parecido al original. Con factor 250 los bordes de las diagonales se ven algo difuminados, pero parece una diferencia insignificante. En esta ocasión elegiría el factor 250.

Tal y como comentaba en la introducción, los únicos problemas que se observan en esta imagen son los encontrados en las diagonales. No obstante, creía que serían mayores que los que observamos.

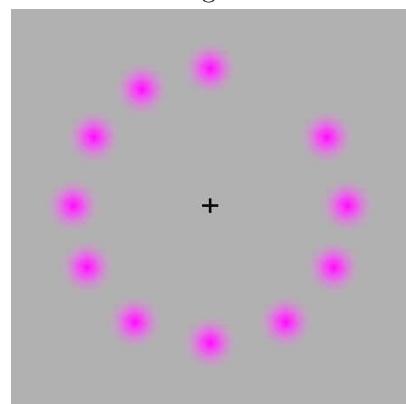
dots



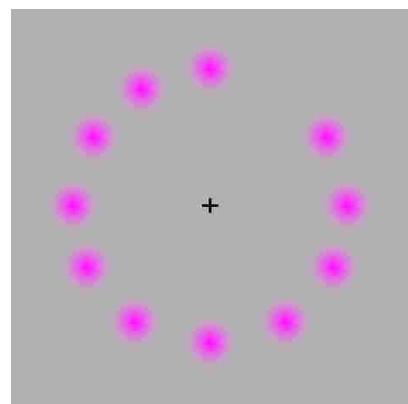
Original



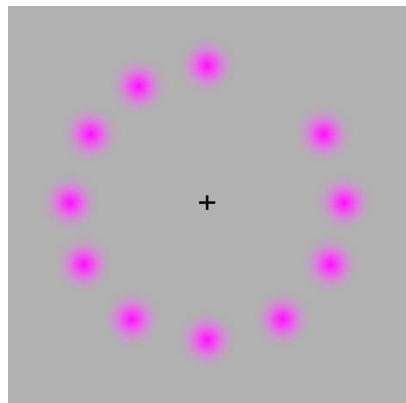
Codificación Default, caliQ = 50



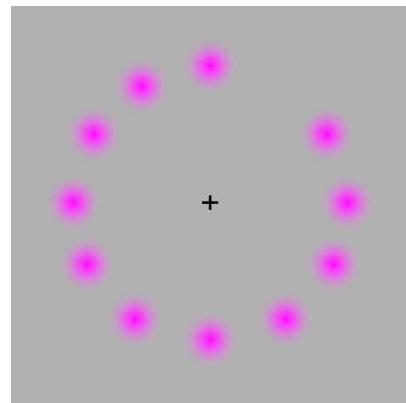
Codificación Default, caliQ = 100



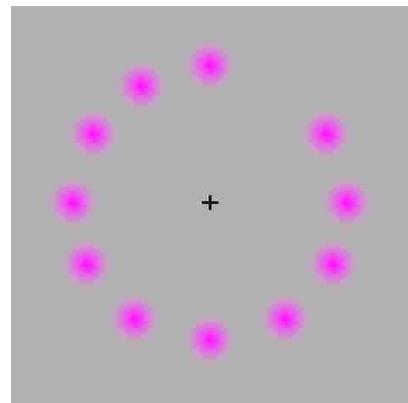
Codificación Default, caliQ = 250



Codificación Custom, caliQ = 50



Codificación Custom, caliQ = 100



Codificación Custom, caliQ = 250

Figura 16: Aplicación a *dots.bmp*

Como dije en la introducción, de esta imagen esperaba buenos resultados. Efectivamente, vemos que en los tres casos el resultado es considerablemente similar al original. No obstante, para el factor 250 se pixela mucho la difuminación de los puntos violetas. El resultado con factor 100 no presenta este fenómeno y por tanto elegimos

este como factor óptimo entre los considerados.

blackbuck

Original



Codificación Default, caliQ = 50



Codificación Default, caliQ = 100



Codificación Default, caliQ = 250



Codificación Custom, caliQ = 50



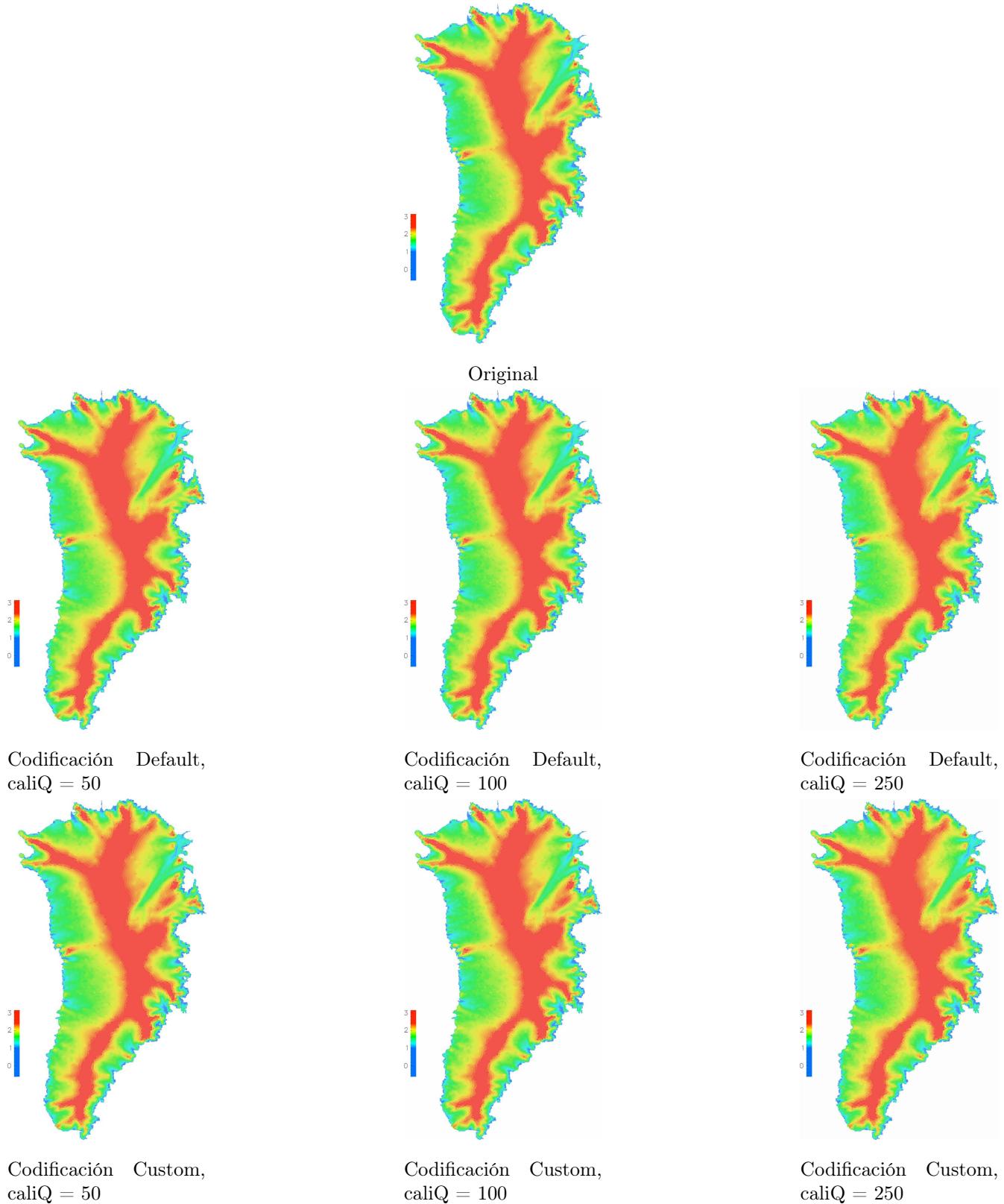
Codificación Custom, caliQ = 100



Codificación Custom, caliQ = 250

Figura 17: Aplicación a *blackbuck.bmp*

Para esta imagen los resultados son excelentes. Gracias a la gran zona constante y a la baja diversidad cromática que presenta, así como los suaves cambios entre las zonas diferenciadas de la imagen ayudan a que podamos aumentar mucho el factor de calidad sin perjudicar el resultado cualitativo. Entonces, elegimos para codificar esta imagen el factor 250.

greenland_grid_veloFigura 18: Aplicación a *greenland_grid_velo.bmp*

Estos resultados se ajustan a las predicciones que efectuamos en la introducción: son muy buenos. Como expliqué antes, creo que esto se debe al gran tamaño de la imagen y a los suaves cambios que se dan en la imagen. Me sorprende bastante que la leyenda se conserve tan bien, pues esperaba que el compresor encontrase dificultades con esa parte. Nada más lejos de la realidad, con el factor 250 los resultados son casi idénticos a la imagen original. Elegimos así este como factor de codificación.

snail

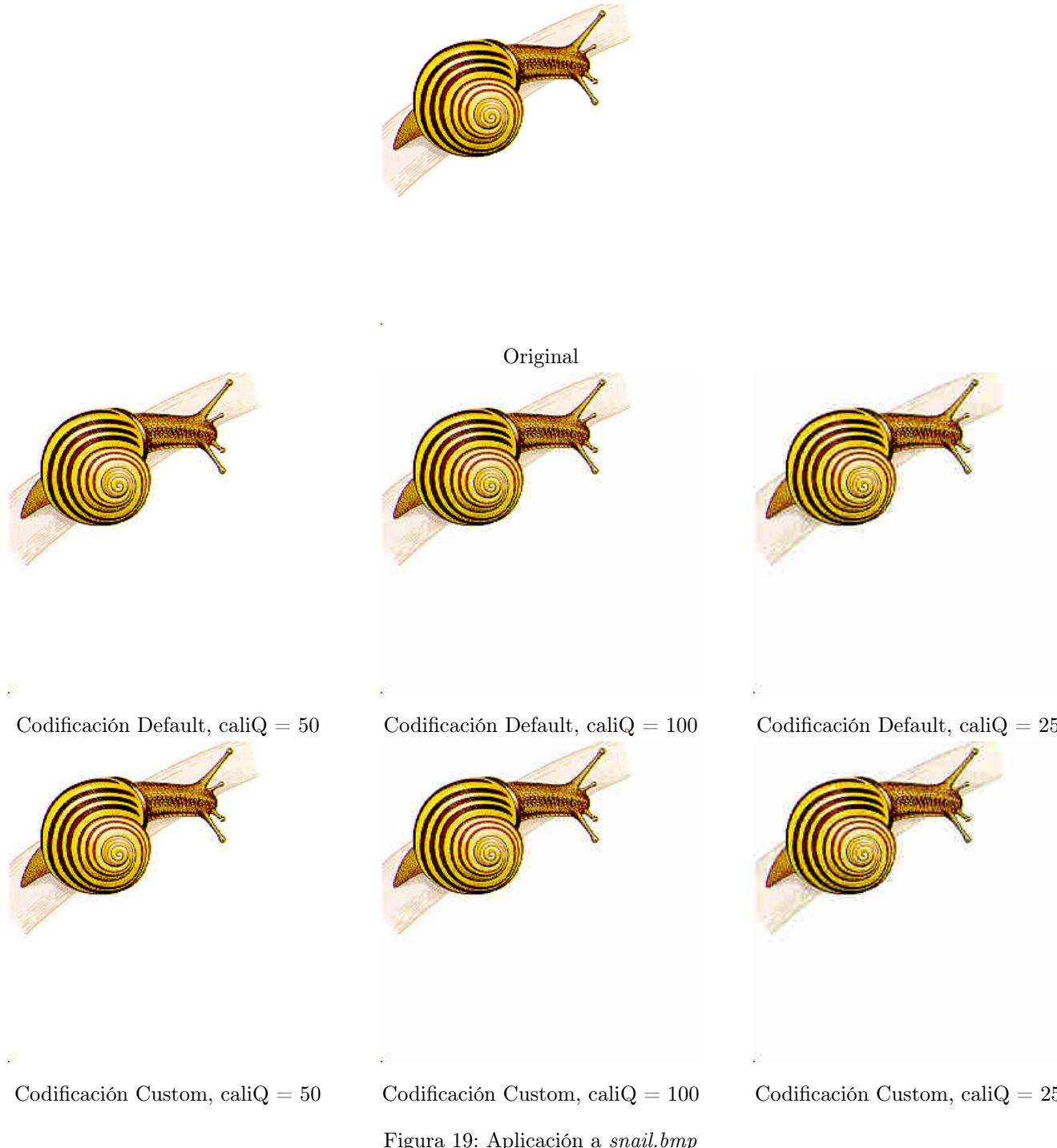


Figura 19: Aplicación a *snail.bmp*

Para la imagen del caracol vemos que los resultados son bastante aceptables. Para el factor 100 la imagen reconstruida es casi igual que la original. Para 250 ya se observan difuminaciones en las bordes y el resultado no parece ser tan bueno como podríamos esperar. Haremos entonces la codificación con factor 100.

lena

Llegamos a la imagen más famosa entre los investigadores del procesamiento de imágenes: la imagen de Lena, que vemos en la figura 19.



Figura 20: Aplicación a *lena.bmp*

Los resultados visuales son sorprendentemente buenos. Con factor 100 no se puede diferenciar apenas la imagen original de la reconstruida, solo se nota una ligera modificación en las sombras de la imagen, casi imperceptible. Y con factor 250 sí que podemos ver como se pixela la imagen y pierde demasiada calidad. En este caso, de nuevo, elegiremos el factor 100.

4.1.1. Conclusión de la comparación cualitativa

Los resultados cualitativos nos han proporcionado mucha información, que nos permite corroborar algunas de las hipótesis que manejábamos al principio:

- Las imágenes de gran tamaño pueden comprimirse mejor que las de pequeño tamaño
- Las mayores discrepancias entre las imágenes reconstruidas y las originales se dan en bordes, sombras, cambios de textura y, en general, cambios bruscos de color o tonalidad
- Aumentar el factor de calidad hace que el resultado pierda parecido con la imagen original

Además, generalmente hemos obtenido que el factor idóneo es 100, por lo que es razonable pensar que este presentará buenos resultados en el estudio cualitativo. Los valores mayores se espera que proporcionen resultados muy pobres, que no compensarán el aumento en la tasa de compresión consecuente.

4.2. Estudio Cuantitativo

Hasta ahora hemos analizado los resultados visuales, que nos han permitido elegir, mediante criterios perceptuales (y por tanto, subjetivos) un valor candidato a ser un buen factor de calidad: 100.

En el estudio cuantitativo trataremos de analizar esta misma cuestión, pero con criterios objetivos, ya sea basados en la distancia (usando el MSE) o en el ruido (SNR). Como ya hemos comentado, hemos usado los valores 1,25,50,100,250,500 y 1000 para el factor de calidad, produciendo, por tanto, para cada imagen 7 ternas (RC,MSE,SNR).

ejemploMulticolor

Obtenemos la siguiente gráfica entre el RC y el MSE:

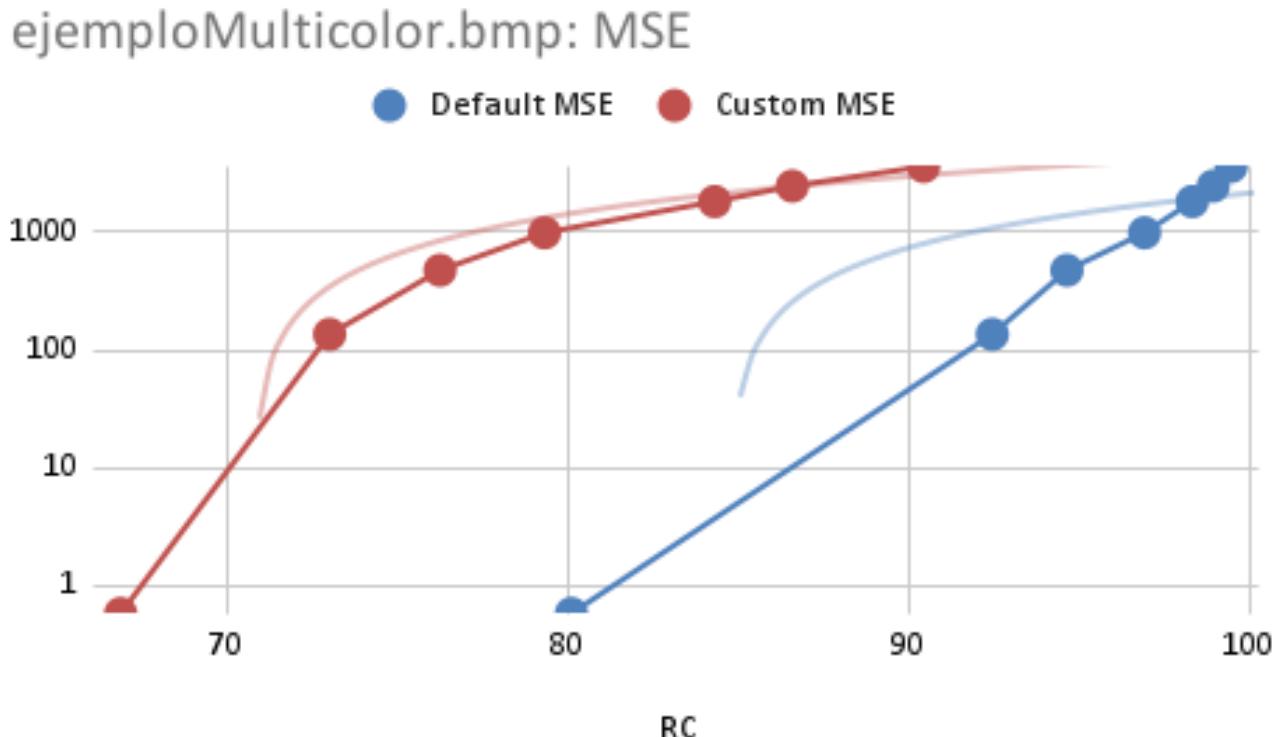


Figura 21: ejemploMulticolor: RC vs log(MSE)

La gráfica muestra los pares (RC,MSE), para cada valor del factor de calidad, ordenados de izquierda a derecha. Es decir, el primer punto rojo se genera con el compresor custom para caliQ=1, el segundo con caliQ=25, y así. De igual forma con los puntos azules para el compresor por defecto.

Además, la escala del MSE es logarítmica. Esto es útil para el análisis pues los valores del MSE se disparan rápidamente, por lo que una escala normal dificultaría mucho la interpretación de las gráficas.

Por otro lado, en un color translúcido podemos ver las líneas de tendencia, que son crecientes como cabría esperar.

Primero, notemos que para un mismo factor de calidad, los puntos de ambos compresores se encuentran prácticamente alineados: tienen el mismo MSE. Esto concuerda con nuestra experiencia perceptual previa, en la que no encontrábamos diferencias entre las imágenes reconstruidas con uno u otro compresor. Las diferencias están, por lo tanto, en el grado de compresión alcanzada para un mismo nivel del MSE.

El caso del *ejemploMulticolor* es muy curioso, ya que será el único en el que encontraremos dos gráficas tan dispares y totalmente separadas. Como puede observarse, el compresor por defecto obtiene mejores ratios de compresión para cualquier valor de caliQ, obteniendo los mismos valores del MSE. Por tanto, en este caso es indudable que se preferable utilizar el compresor por defecto. Por otro lado, respecto al factor de calidad, vemos que para un valor de 100 (el cuarto punto), el MSE se sitúa ya en 1000. Esto es demasiado, y probablemente deberíamos elegir un valor menor, en consonancia con lo que vimos en el estudio cualitativo.

Respecto al SNR, observamos la figura 22.

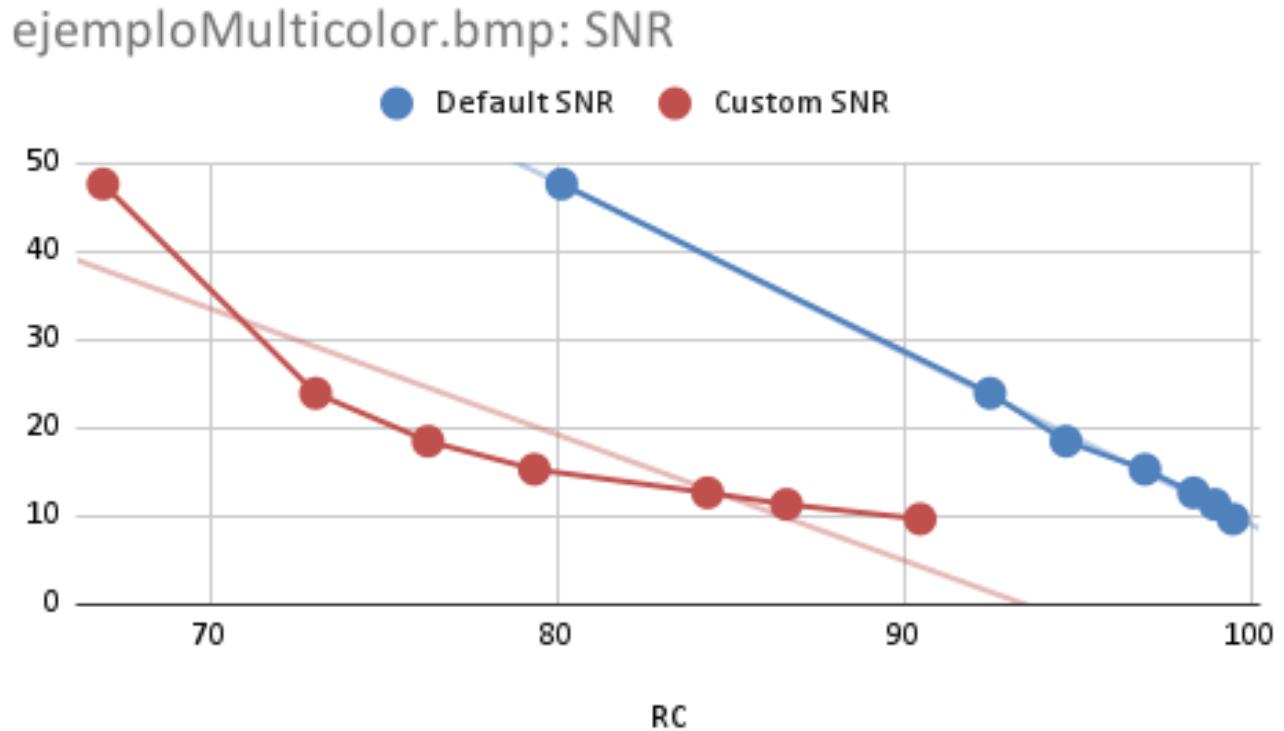


Figura 22: ejemploMulticolor: RC vs SNR

En esta ocasión, las gráficas tienen una tendencia decreciente, como esperaríamos (recordemos que a mayor SNR, menos ruido estaremos introduciendo en la imagen final respecto a la original).

Observamos, además, que tal y como sucede con el MSE, los puntos generados con el mismo factor de calidad se sitúan a la misma altura, por lo que el análisis es sencillo: para un mismo factor de calidad, será preferible aquel compresor que consiga que el punto esté más a la derecha (tendrá mejor RC). Así, obtenemos la misma conclusión

que antes: para esta imagen es preferible el compresor por defecto.

Por otro lado, basándonos en lo que ya sabemos, podemos hacer una estimación sobre qué valores del SNR comienzan a ser deficientes. Como sabemos que el factor de calidad 100 genera una mala aproximación en este caso, podemos estimar que un SNR de alrededor de 20 es posible que implique malos resultados.

windows

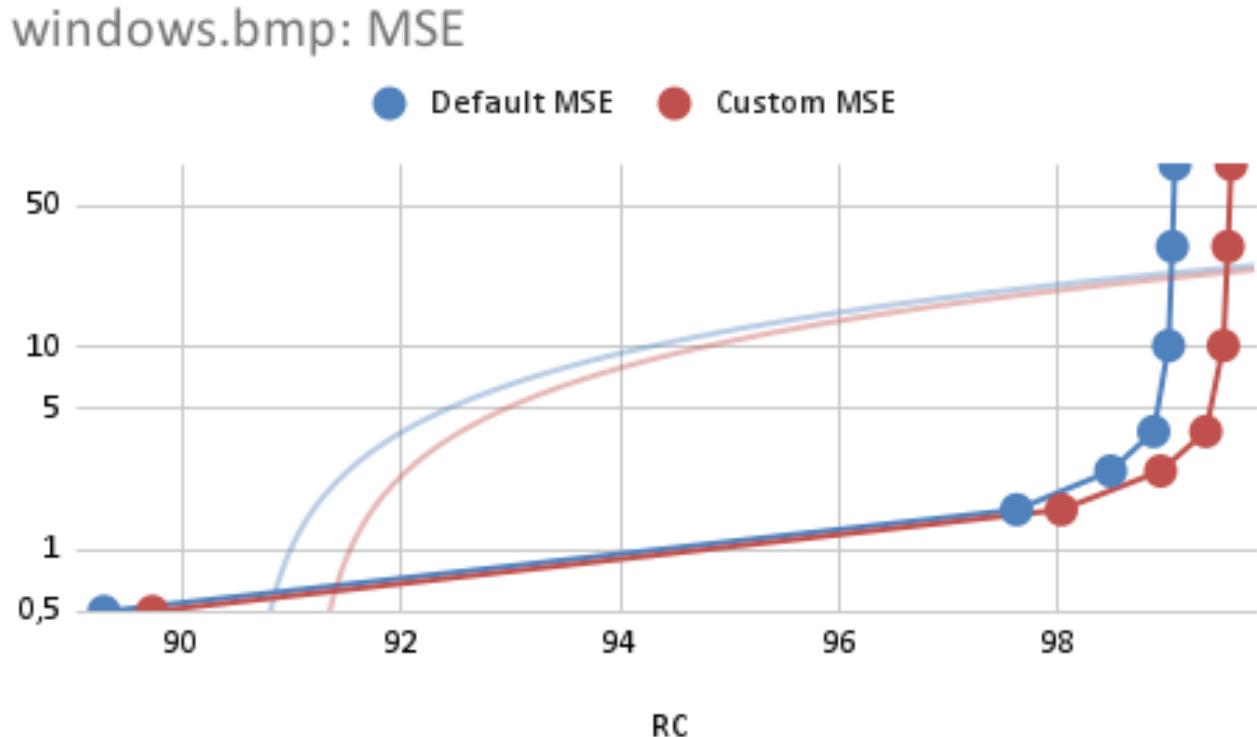


Figura 23: windows: RC vs log(MSE)

Esta gráfica es más paradigmática y se parece más a lo que nos encontraremos en general que la anterior.

Observamos de nuevo que los puntos aparecen alineados, este hecho se dará en todos los casos y no es necesario volver a notarlo, al igual que el crecimiento de la gráfica (o el decrecimiento en el caso del SNR, que veremos a continuación).

Vemos como el MSE es muy pequeño en todos los casos. Compárese con la imagen anterior, por ejemplo, en la que se alcanzaba rápidamente un MSE de 1000. En esta, en cambio, alcanzamos ratios de compresión cercanos al 100 %, con un MSE de alrededor de 50.

Una observación que puede sernos útil es como a partir de caliQ=100 (el cuarto punto), la gráfica es casi vertical. Eso significa que aumenta mucho el MSE pero muy poco el RC al aumentar el valor de caliQ. Por tanto, cuando llegamos a estas zonas verticales no tiene sentido seguir aumentando el factor de calidad, ya que empeoraremos considerablemente la calidad de la imagen sin obtener una ganancia significativa en términos de compresión.

Además, ahora es el compresor customizado el que se sitúa todo el rato a la derecha, por lo que este es el que deberíamos preferir para codificar esta imagen. El factor de calidad de 100 parece además la mejor elección entre las posibles: el último antes del tramo vertical de la gráfica. Este valor nos da el excelente valor de algo mas de 99 % de compresión, por lo que realmente es difícil pedir más.

Pasamos al SNR:

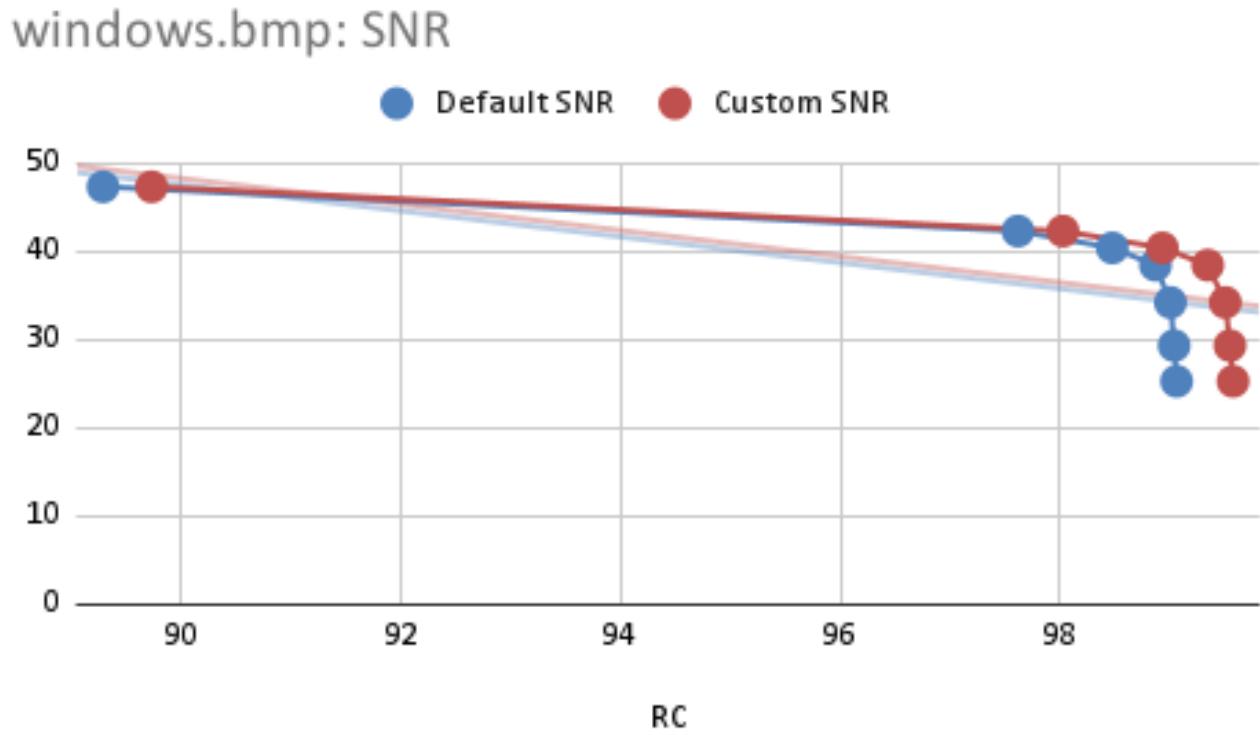


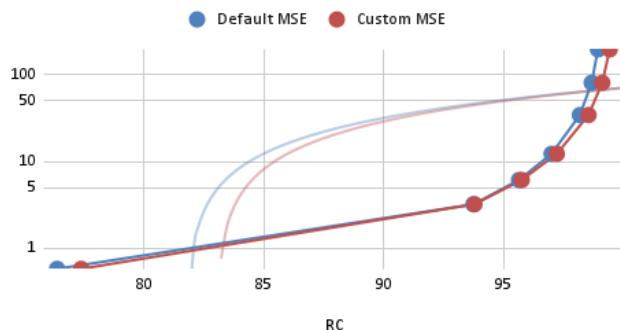
Figura 24: windows: RC vs SNR

Observamos prácticamente los mismos fenómenos que en la gráfica con MSE: el compresor customizado siempre a la derecha y un tramo final vertical (aunque obviamente este es decreciente). Las conclusiones obtenidas son las mismas y, en general, parece que el MSE y el SNR son ambas medidas muy similares en cuanto a la información que nos proporcionan. De ser esto así, diría que en términos de eficiencia deberíamos utilizar el SNR, ya que requiere ligeramente de menos cómputo (aunque debería ser el SNR antes de tomar el logaritmo en base 10). En las siguientes gráficas comprobaremos si siempre arrojan la misma información o si, por el contrario, en ocasiones se complementan o incluso se contradicen.

leon

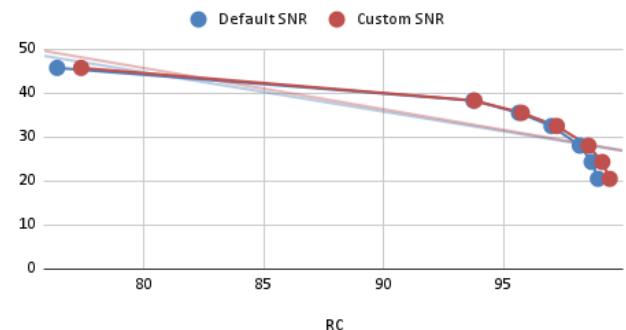
En la figura 25 encontramos las dos gráficas referentes a la imagen *leon.bmp*.

leon.bmp: MSE



RC vs log(MSE)

leon.bmp: SNR



RC vs SNR

Figura 25: Gráficas de leon.bmp

Vemos dos gráficas muy similares a las de la imagen anterior, y que nos dan conclusiones análogas.

No obstante, parece significativo que el cambio en los valores del MSE es mayor que el cambio en los del SNR, que mantiene un rango similar. De ser esto así y en caso de seguir proporcionando las mismas conclusiones, sería mejor utilizar una medida como el SNR, ya que es más sencillo comparar valores que se encuentran dentro de un pequeño rango, que compararlos en uno mayor. Por ejemplo, en este caso y el anterior, para caliQ=100, obtenemos:

caliQ=100	windows	leon
MSE	3,861250677	12,27295156
SNR	38,47262791	32,53528044

Y se ve cómo los valores del SNR son mucho más cercanos, por lo que parece razonable pensar que es más informativo transmitir este valor que el MSE, si este hecho se mantiene de forma general.

captura

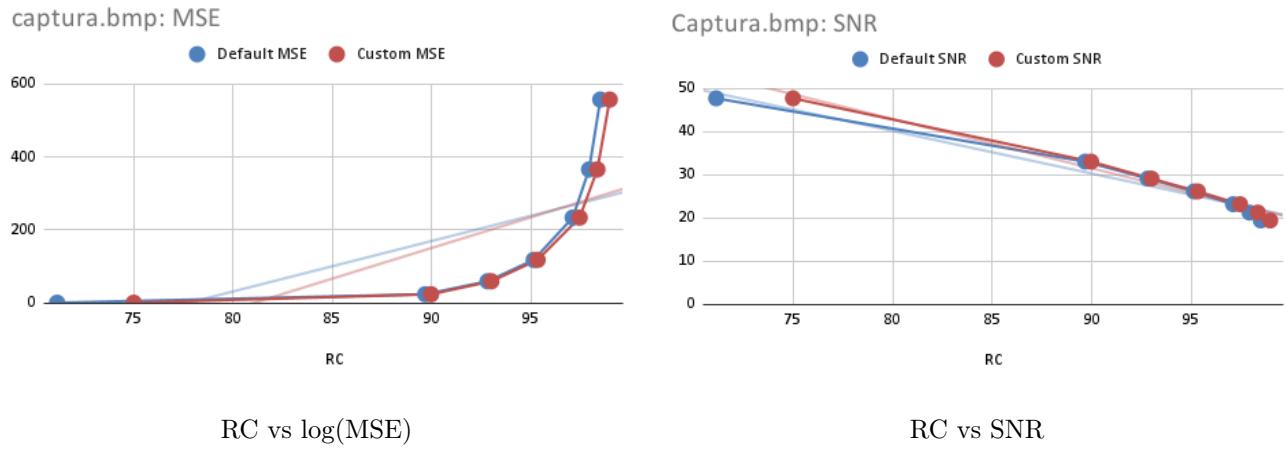


Figura 26: Gráficas de captura.bmp

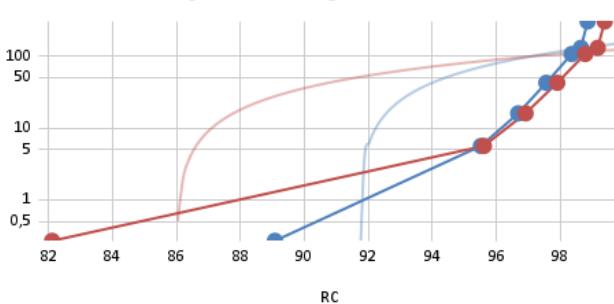
Esta vez vemos que ambos compresores se comportan de forma muy parecida, aunque el customizado ligeramente mejor en todos los casos. Además, seguimos viendo gran variabilidad en el rango del MSE, frente a un rango prácticamente idéntico del SNR.

Para esta imagen, el análisis cualitativo nos hizo concluir que usar un valor de 100 para el factor de calidad era correcto. Por tanto, parece que, viendo la información disponible hasta el momento, tomar como criterio de calidad que el SNR se encuentre alrededor de 30dB parece algo muy razonable. De esta forma, seleccionando el valor caliQ=100, obtendríamos una tasa de compresión del 95 %, la cual es considerablemente alta.

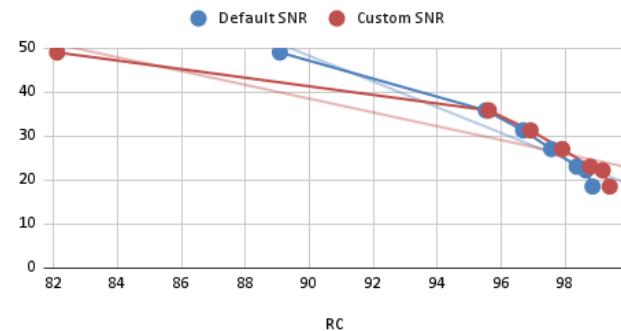
De hecho, en todos los casos vistos hasta ahora un valor del SNR de 30 nos da resultados satisfactorios. Un criterio similar usando el MSE no parece posible de obtener.

bmp_24

bmp_24.bmp: MSE



bmp_24.bmp: SNR



RC vs log(MSE)

RC vs SNR

Figura 27: Gráficas de bmp_24.bmp

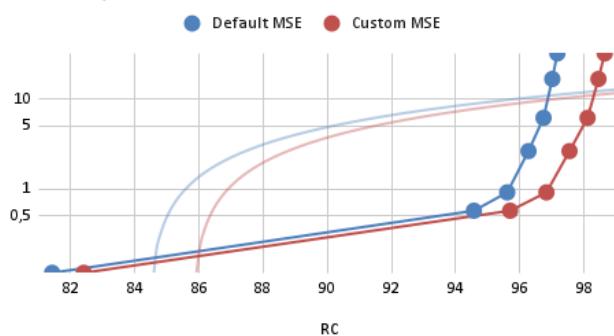
Esta vez el compresor por defecto funciona mejor para el factor de calidad 1, pero para los demás lo adelante el customizado, aunque no con mucha diferencia.

Si siguiésemos el criterio establecido hasta ahora de obtener un SNR cercano a 30, seleccionaríamos un factor de calidad de 50 o 100. Recordemos que en análisis cualitativo seleccionamos 250 como factor de calidad, y es que esta imagen es muy sencilla y los errores que se introducen no se traducen directamente en una disminución de la interpretabilidad de la misma. Esto deja patente que, si bien podemos establecer criterios generales de valoración de la calidad de los resultados, es obvio que un estudio pormenorizado de cada caso es siempre más preciso. Pero esto, claro está, no es practicable a gran escala, por lo que los criterios generales son necesarios.

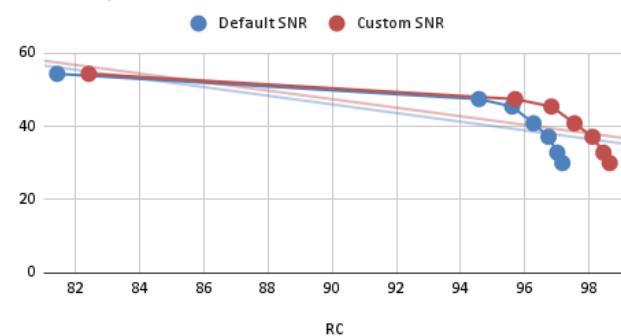
De nuevo, elegimos el compresor customizado con caliQ=100 obteniendo una compresión del 98 %.

dots

dots.bmp: MSE



dots.bmp: SNR



RC vs log(MSE)

RC vs SNR

Figura 28: Gráficas de dots.bmp

Volvemos a obtener las gráficas prototípicas, que nos llevan a la misma conclusión que siempre que aparecen: el compresor customizado funciona mejor para esta imagen.

Por otro lado, observamos muy bajos valores del MSE, y valores mayores del SNR de los que estamos acostumbrados.

Recordemos que para esta imagen obtuvimos muy buenos resultados con todos los factores de calidad utilizados en el estudio cualitativo, aunque concluimos que el valor de 250 implicaba una pixelación excesiva y decidimos tomar 100 como valor óptimo. Según nuestro criterio del SNR, esta vez elegiríamos caliQ=500 o incluso 1000. Pero esta imagen hay que mirarla con lupa, notemos que es casi entera constante, es decir, toda la parte gris contribuye al SNR de forma positiva, pues ahí no se producirá error.

Es decir, el SNR no es muy informativo para imágenes que presentan un alto porcentaje de zonas idénticas. Si vemos el resultado con caliQ=1000, en la figura 29, observamos que es muy pobre.

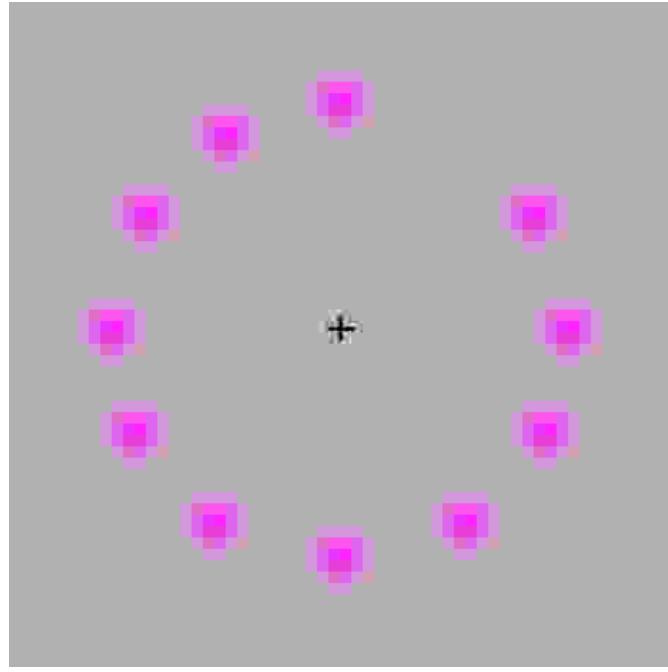
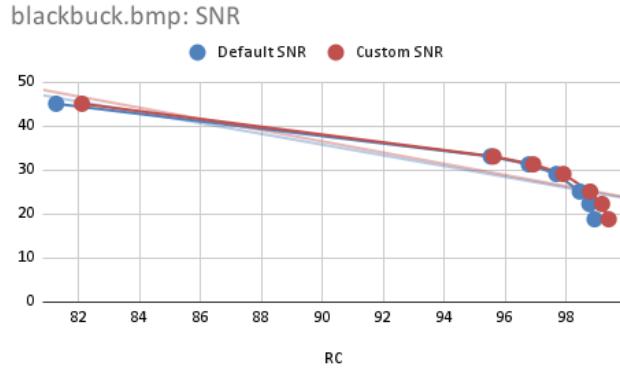
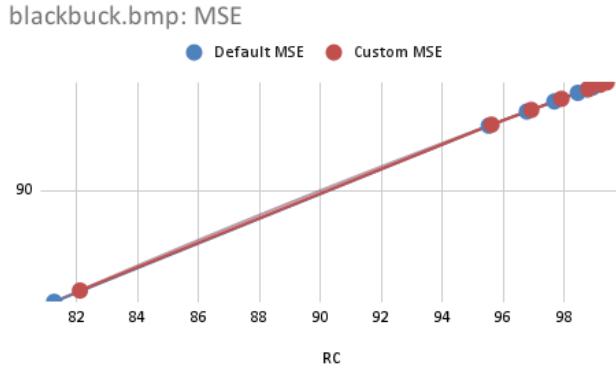


Figura 29: Reconstrucción de dots.bmp con caliQ=1000

Por tanto, de alguna forma vemos que sí que se complementan el MSE y el SNR. El criterio que utilizamos del SNR será más fiable cuanto mayor sea el MSE. Esto parece un buen criterio de calidad, ya que como hemos visto, el MSE por sí solo no es capaz de mantener un rango de valores estable, por lo que su utilización como criterio estándar de calidad es compleja. Por otro lado, el SNR mantiene un rango mucho menor de valores, de forma que facilita la comparación entre distintas aplicaciones del compresor. No obstante, este es altamente dependiente de la morfología de la imagen y estará sesgado positivamente hacia imágenes con muchas zonas constantes. Como el MSE sí que es útil para medir este último hecho, puede servirnos para establecer un factor de corrección sobre el SNR (esto escapa del presente estudio, pero parece algo posible de realizar).

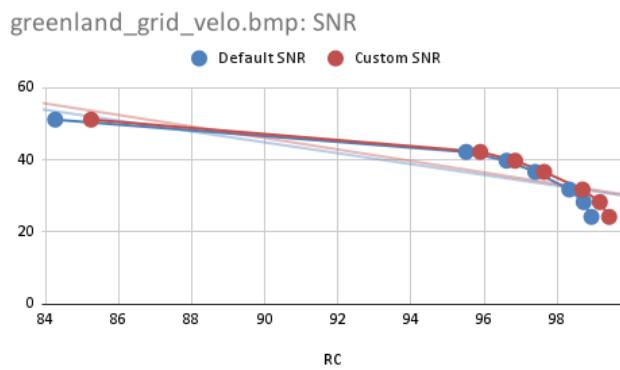
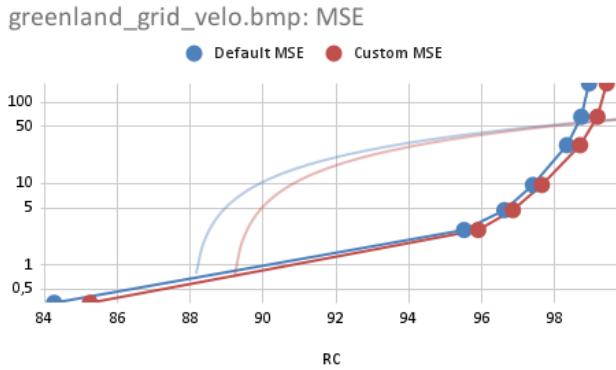
blackbuck

RC vs log(MSE)

RC vs SNR

Figura 30: Gráficas de blackbuck.bmp

Para la imagen del ciervo ambos compresores nos proporcionan resultados casi idénticos, aunque ligeramente mejores para el customizado, como viene siendo normal. Además, sobre SNR=30 se coloca el punto correspondiente a caliQ=100, aunque esta imagen también presenta una gran parte constante, por lo que probablemente podamos apurar más y tomar caliQ=250, en concordancia con la elección que tomamos en la comparación cualitativa, y obteniendo una tasa de compresión de 99 %, que es excelente.

grenland_grid_velo

RC vs log(MSE)

RC vs SNR

Figura 31: Gráficas de greenland_grid_velo.bmp

Obtenemos otra vez las gráficas prototípicas y las conclusiones que ya hemos comentado anteriormente.

Un valor de 100 o de 250 para el factor de calidad parecen buenas elecciones según los criterios que venimos siguiendo. Estos nos dan tasas de compresión del 97 % y el 99 %, respectivamente. Por tanto, elegimos 250 como valor

óptimo del factor de calidad. Si revisamos el estudio cualitativo, estos resultados son coherentes con las conclusiones que obtuvimos en su momento.

snail

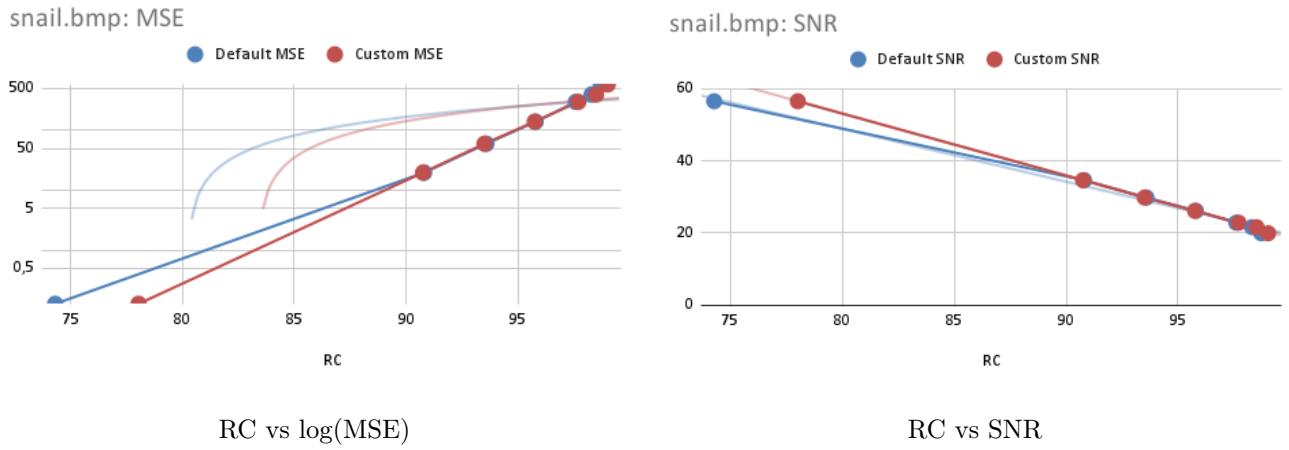


Figura 32: Gráficas de snail.bmp

Aquí los resultados de ambos compresores son idénticos, exceptuando el valor 1 del factor de calidad (pero este es irrelevante prácticamente). Además, observamos que el RC es menor que en las imágenes anteriores.

Alrededor de los 30dB encontramos los puntos generados a partir de los factores 50 y 100 (RC de 94 % y 96 % respectivamente). Por tanto, una buena elección parece ser elegir el factor 100 (mayor compresión y similar calidad esperada) y el compresor por defecto, que necesita de menos cálculo al usar tablas predefinidas.

lena

Veamos las gráficas obtenidas para la última de las imágenes estudiadas:

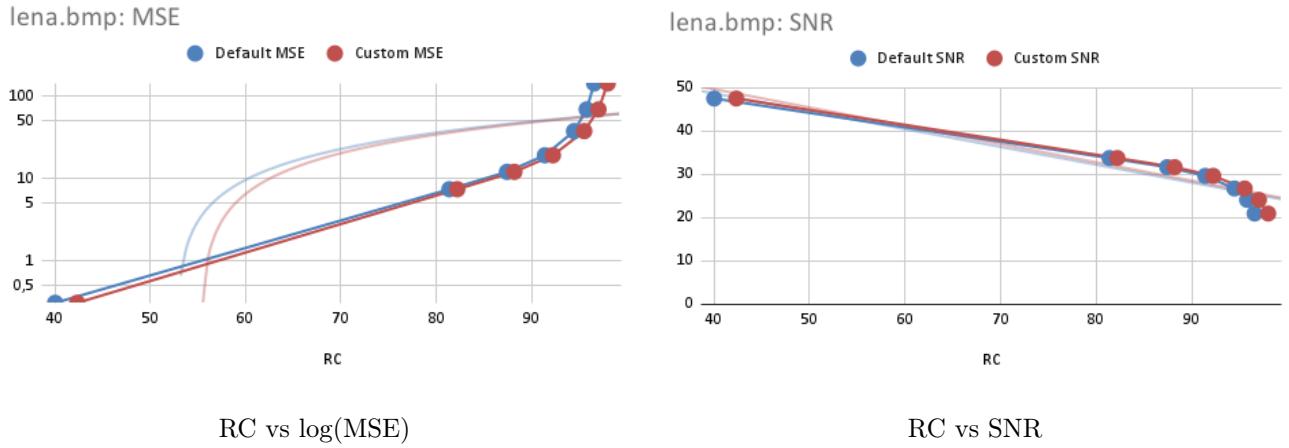


Figura 33: Gráficas de lena.bmp

De nuevo, las gráficas prototípicas. El compresor customizado funciona mejor y nuestro criterio nos dice que utilicemos el valor de caliQ=100, con una tasa de compresión de un 92 %. Nótese las bajas tasas de compresión obtenidas para esta imagen.

4.2.1. Resultados medios

Puede ser interesante ver qué sucede con los resultados promedio. Para ello, tomamos el promedio de todos los valores obtenidos, tanto de RC, como de MSE y SNR, diferenciando, claro está, el caso del compresor por defecto del customizado. Obtenemos las siguientes gráficas:

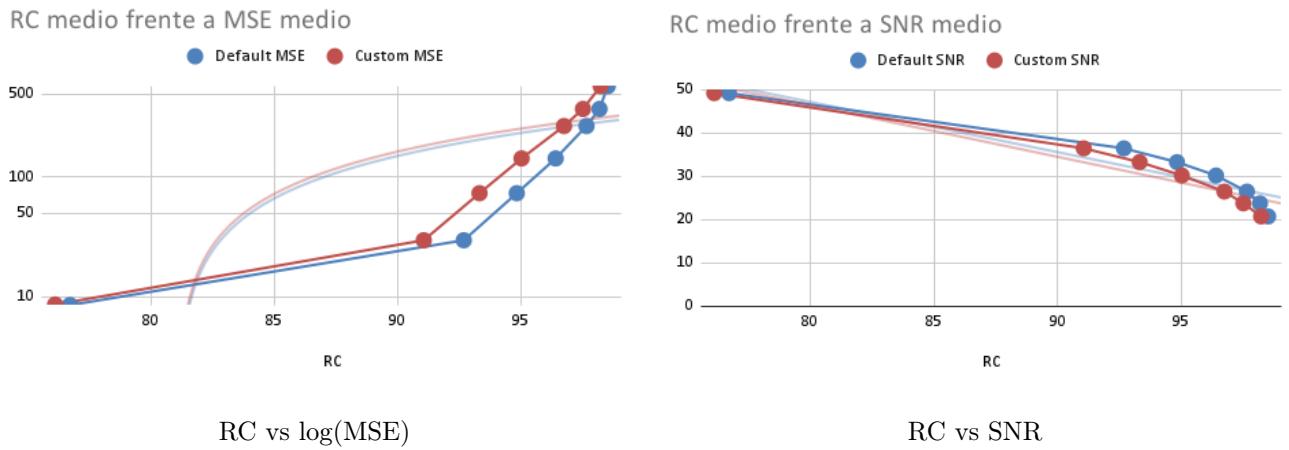


Figura 34: Gráficas de los resultados medios

Vemos que las medias parecen sugerir que el compresor por defecto funciona mejor. No obstante, hemos determinado que el compresor customizado funcionaba mejor en 8 de los 10 casos estudiados (el 80 %) y al menos tan bien como el compresor por defecto en 9 de los 10 casos. Esta aparente discrepancia se debe a la gran discrepancia que se encuentra en el caso del *ejemploMulticolor*, para el que el compresor por defecto funcionaba mejor, y daba

unos resultados altísimos para el MSE y bastante bajos dentro del rango en el que se mueve el SNR. Por tanto, creo que es conveniente construir unas nuevas gráficas de medias obviando los datos de esta imagen de comportamiento inusual.

Las nuevas gráficas son las siguientes:

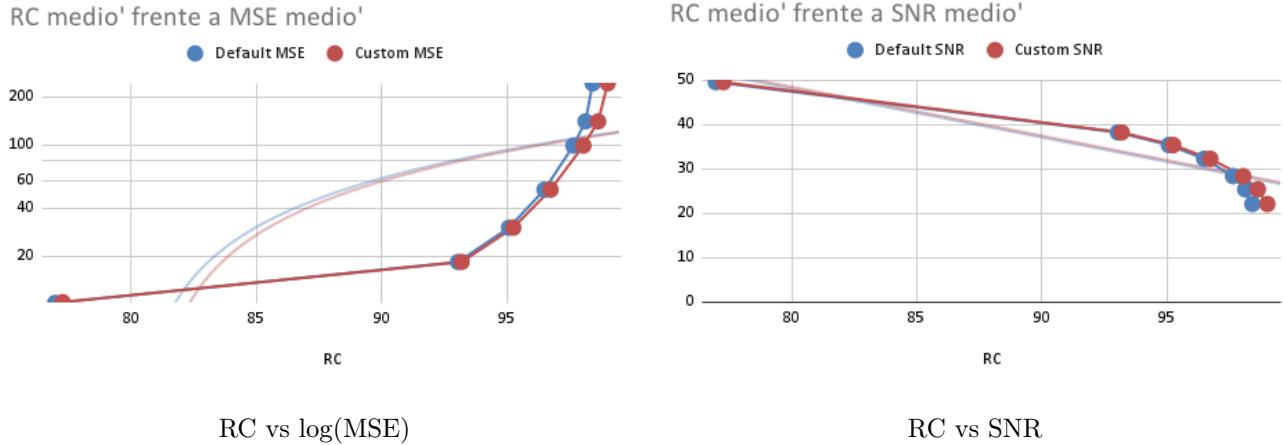


Figura 35: Gráficas de los resultados medios (obviando *ejemploMulticolor*)

Y estos resultados parecen mucho más coherentes con la comparativa realizada: el compresor customizado es tendencialmente mejor que el por defecto.

Además, hemos elegido en la mayoría de casos un valor para el factor de calidad de 100, que es el que se sitúa en media sobre los 30dB, por lo que no íbamos muy desencaminados con nuestro criterio de verificación de calidad.

Tomando siempre este factor de calidad, obtendríamos, por tanto, una compresión media de alrededor del 97%, y no parece que la pérdida de la calidad que se da sea excesiva.

De este modo, si hubiésemos de elegir una única configuración estándar que diese resultados aceptables en todos los casos y los mejores resultados en la mayoría de casos, elegiríamos el compresor customizado con un factor de calidad de 100.

4.2.2. ¿Qué sucede con las imágenes png?

Los resultados obtenidos para las imágenes png, como puede consultarse en el excel */source/Data/Datos definitivos.xls*, son prácticamente idénticos a los obtenidos con las imágenes en formato bmp. Lo único que cambia es el ratio de compresión, que suele ser menor para las imágenes png, lo cual es evidente si lo pensamos, ya que nuestro compresor actúa sobre la imagen, obviando el formato subyacente, por lo que cualquier imagen que codifiquemos ocupará lo mismo independientemente del formato en que se encuentre. Como png es un formato comprimido sin pérdidas, una imagen en este formato ocupa menos que la misma imagen en bmp (sin pérdidas pero sin comprimir). Por lo tanto, era esperable obtener peores tasas de compresión, y los mismos resultados en cuanto a MSE y SNR.

En algunas ocasiones incluso se obtiene una tasa negativa de compresión (ocupa más la imagen comprimida que la original en formato png). Esto se debe a que para ciertas imágenes será más eficiente la codificación utilizada por png que la que usa nuestro compresor.

5. Conclusión

Como hemos visto, las cuestiones relacionadas con el establecimiento de valores por defecto en términos de compresión es muy complejo, pues siempre podemos encontrar casos que produzcan malos resultados para una configuración dada. No obstante, sí que hemos sido capaces de encontrar una configuración que ofrece resultados razonables en todos los casos estudiados, y muy buenos en la mayoría de ellos.

Este estudio podría proseguirse tratando de establecer de qué forma podríamos construir un factor de corrección para la medida de la calidad de la imagen mediante SNR utilizando los valores del MSE, como comentaba en el estudio cuantitativo. También podrían estudiarse otras variables que tienen indudable importancia en el proceso de compresión, como son el tiempo de cómputo o si puede determinarse antes de la compresión una aproximación del valor óptimo del factor de calidad (por ejemplo midiendo la entropía de la imagen antes de comprimirla), lo cual no es descabellado.

Como se puede observar, respecto a las mejoras propuestas, se ha realizado la comparación de 10 imágenes, que además se ha complementado con la comparación con las mismas imágenes en dos formatos: bmp y png.

Además, se ha añadido al estudio la medición del SNR, lo cual creo que ha sido acertado y fructífero.

Referencias

- [1] Fundación Atapuerca. *Los yacimientos de la Sierra de Atapuerca*. URL: <https://www.atapuerca.org/es/atapuerca/Yacimientos-de-Atapuerca>.
- [2] Vladimir Britanak. *Discrete cosine and sine transforms : general properties, fast algorithms and integer approximations*. Amsterdam Boston London: Academic, 2007. ISBN: 9780123736246.
- [3] Luis Daniel Hernández. *Compresores estadísticos: Método de Huffman*. Apuntes de la asignatura Compresión Multimedia, Curso 21/22, UMU, DIIC.
- [4] University of South Carolina. *BMP Files*. Último acceso: 2/1/2022. URL: <https://people.math.sc.edu/Burkardt/data/bmp/bmp.html>.
- [5] Dennis O. Wackerly. *Mathematical Statistics with Applications*. CENGAGE LEARNING, nov. de 2007. 912 págs. ISBN: 0495385085. URL: https://www.ebook.de/de/product/6702600/dennis_o_wackerly_mathematical_statistics_with_applications.html.
- [6] wikipedia. *Lena*. URL: <https://es.wikipedia.org/wiki/Lenna>.

Apéndice A Codificación Huffman

En los apuntes de la asignatura ([3]) hay una explicación profunda y ejemplificada. Aquí voy a resumirla.

El método de Huffman busca conseguir un **código óptimo**⁷, que básicamente es un código cuya longitud media de palabra es mínima entre los códigos que codifican cierta información.

Para este propósito, el método de Huffman consta del siguiente proceso:

- Se ordenan los símbolos a codificar por probabilidad de aparición decreciente.
- Se construye un nuevo conjunto de símbolos, consistente en el anterior, pero agrupando los dos términos menos probables. La probabilidad de este símbolo es la suma de los dos símbolos juntados.
- Se repiten los dos primeros pasos hasta conseguir un conjunto con únicamente dos símbolos.
- Se asignan las palabras 0 y 1 a los dos símbolos del conjunto reducido obtenido.
- Se vuelve hacia atrás en el proceso, asignando las palabras 1 y 0 de forma acumulativa hasta llegar al conjunto de símbolos original.

De esta forma se obtiene un árbol binario (no balanceado) denominado **árbol de Huffman** y el código resultante es un **código de Huffman**.

⁷Para definiciones precisas, consultar los apuntes mencionados.