

XAMU – Xarxes Multimèdia - EPSEVG-UPC

Sessió Pràctica Unitat 4, Primavera 2024

Estudi Previ

Compressió d'Imatge

Ernest Anguera Aixalà
Nawal Bouallala Safyoun
Mariona Farré Tapias
Iván García Rodríguez

Índex:

Exercici 1: Bitmaps.....	3
Apartat a.....	3
Apartat b.....	3
Apartat c.....	3
Exercici 2: Característiques d'imatge.....	4
Apartat a.....	4
Apartat b.....	5
Apartat c.....	5
Exercici 3: Còdecs moderns: WebP, JPEG-XR.....	6
Exercici 4: Característiques de LViewPro.....	7
Apartat a.....	7
Apartat b.....	11
Exercici 5: Característiques de XnView MP.....	12
Exercici 6: FFmpeg.....	17
Exercici 7: Perspectiva de gènere.....	18
Apartat a.....	18
Apartat b.....	18
Apartat c.....	19
Apartat d.....	19
Apartat e.....	19

Exercici 1: Bitmaps

Apartat a

Podem calcular l'espai que ocupa l'imatge amb la fórmula:

$$Mida = pixels_{horitzontal} * pixels_{vertical} * profunditat\ de\ color$$

En aquest cas, com que la profunditat de color és de 256 nivells de grisos, necessitem 8 bits per a representar els 256 nivells. Sabem que 8 bits són un byte, per tant, la profunditat de color és d'1 byte.

$$Mida = 1742 * 2299 * 1 = 4004858\ bytes = 3.81933\ Mbytes$$

Per tant, la imatge original ocuparà 3.82 Mbytes

Apartat b

Podem calcular l'espai que ocupa l'imatge amb la fórmula:

$$Mida = pixels_{horitzontal} * pixels_{vertical} * profunditat\ de\ color$$

En aquest cas, la profunditat serà 0.75 bits/píxel, com s'indica a la taula que es troba als apunts del tema 4 de teoria:

Bits/píxel a la salida	Calidad subjetiva
0,15	Reconocible
0,25	Útil
0,75	Excelente
1,5	Indistinguible

$$Mida = 1742 * 2299 * 0.75/8 = 375455.4375\ bytes = 0.358\ Mbytes$$

Per tant, la imatge original ocuparà 0.358 Mbytes

Apartat c

En el cas d'una imatge BMP, cada píxel es representa amb 24 bits (8 bits per a cada color primari: vermell, verd i blau). Però en el cas 4:2:2, cada píxel es representa amb 16 bits (8 bits per al canal Y, 4 bits per al canal U i 4 bits per al canal V).

Lavors:

$$Mida_{bitmap} = pixels_{horitzontal} * pixels_{vertical} * profunditat\ de\ color$$

$$Mida_{bitmap} = 1742 * 2299 * 16/8 = 8008716\ bytes = 7.638\ Mbytes$$

$$Mida_{jpeg} = pixels_{horitzontal} * pixels_{vertical} * (4:2:2) * "qualitat"$$

$$Mida_{jpeg} = 1742 * 2299 * (1 + .5 + .5) * 0.75 = 6007287\ bits/8\ bit/bytes = 750910.875\ bytes = 0.7161\ Mbytes$$

Exercici 2: Característiques d'imatge

Visualitzeu els fitxers collserola.tif (és la imatge a què es fa referència a l'estudi previ 1), desert.bmp i koala.bmp. Per a cadascuna comenteu, breument:

Apartat a

collserola.tif:

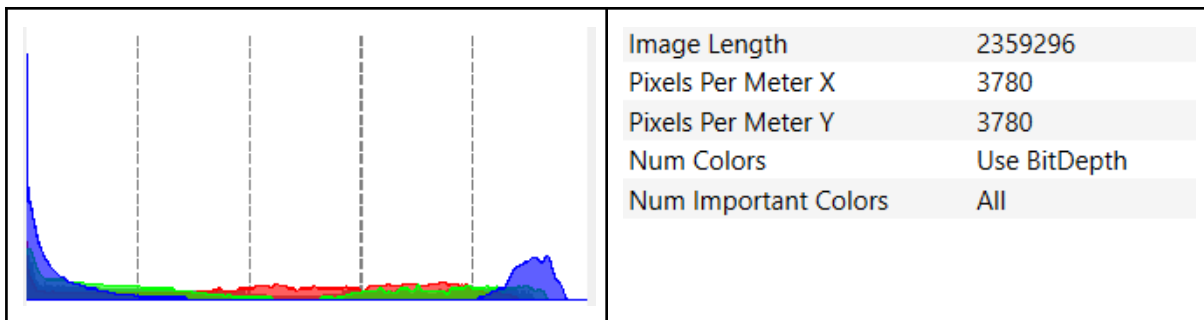
8 bits per tant 256 nivells = 256 colors diferents per la imatge



Com que els colors son grisos, és a dir no hi ha components RGB, la freqüència es nula. Per tant, tampoc hi ha pixels per metre en el paràmetre X i Y.

desert.bmp:

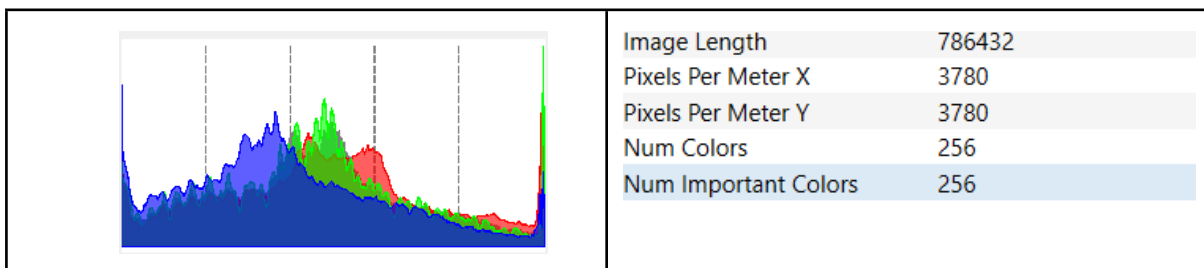
24 bits per tant 16777216 nivells = 16777216 colors diferents per la imatge



Com que aquí sí que hi ha els colors RGB, podem veure les diferents frequencies. Per exemple el color R predomina al centre, mentre que el B predomina als laterals. Els píxels per paràmetre X i Y son 3780

koala.bmp:

24 bits per tant 16777216 nivells = 16777216 colors diferents per la imatge



En aquest cas predomina el component B, pero en el centre predomina el component B i G. També cal resaltar que hi ha el mateix nombre de píxels tant pel paràmetre X com el paràmetre Y:3780

Apartat b

Collserola

$1742 \times 2299 \times 8\text{bits} = 32038864 \text{ bits} = 4004858 \text{ bytes}$

4,0 MB (4.007.884 bytes) a Linux

dessert

$1024 \times 768 \times 24\text{bits} = 18874368 \text{ bits} = 2359296 \text{ bytes}$

2,4 MB (2.359.350 bytes) a Linux

koala

$1024 \times 768 \times 24\text{bits} = 18874368 \text{ bits} = 2359296 \text{ bytes}$

2,4 MB (2.359.350 bytes) a Linux

Com podem veure hi ha una petita diferència, això és degut a que molts sistemes operatius consideren que 1Mbytes son 1000000 bytes en comptes de 1048576 bytes. Per això hi ha aquesta petita diferència.

Apartat c

Comparar la informació que aporta MediaInfo:

Collserola

$1742 \times 2299 \times 8\text{bits} = 32038864 \text{ bits} = 4004858 \text{ bytes} = 4004858 / 1048576 = 3.819 \text{ Mbytes}$

3,82 MB a MediaInfo

desert

$1024 \times 768 \times 24\text{bits} = 18874368 \text{ bits} = 2359296 \text{ bytes} = 2359296 / 1048576 = 2.25 \text{ Mbytes}$

2,25 MB a MediaInfo

koala

$1024 \times 768 \times 24\text{bits} = 18874368 \text{ bits} = 2359296 \text{ bytes} = 2359296 / 1048576 = 2.25 \text{ Mbytes}$

2,25 MB a MediaInfo

Com podem veure no hi ha diferències, ja que obtenim els mateixos valors.

Exercici 3: Còdecs moderns: WebP, JPEG-XR

Mireu les entrades (en anglès, preferentment) de la Wikipedia per WebP5 i JPEG-XR6 . Per a cada cas, desvcriviu breument l'algoritme de compressió: quina transformada utilitza, quin submostreig de crominància, quin codificador entròpic, etc. Compareu cadascun amb JPEG.

WebP utilitza un algorisme de compressió que es basa en la codificació intra-frame, aprofitant la similitud entre píxels veïns. Utilitza transformacions de blocs amb vuit bits de profunditat de color i submostreig de croma en una proporció de 4:2:0. La longitud màxima de les imatges a WebP és de 16.384 píxels a qualsevol costat. Es basa en la predicció de blocs, on cada bloc prediu els valors dels blocs veïns, compriment les dades mal predites mitjançant una transformada discreta del cosinus o una transformada Walsh-Hadamard, seguit de codificació aritmètica.

D'altra banda, JPEG XR utilitza principalment transformacions de bloc de 4x4, encara que també té opcions per a 2x2 i 8x8 per a casos especials. Utilitza un còdec de bloc per a la compressió sense pèrdua, semblant a JPEG , però admet profunditats de color de fins a 32 bits. La codificació de l'entropia implica un esquema de predicció de coeficients DC i CA, reordenació de coeficients i codificació de Huffman adaptativa.El codificador que utilitza inclou YUV_444, YUV_422, YUV_420, i Y_only monocrom.

En comparació, WebP utilitza un mètode de codificació aritmètic, més efectiu que la codificació Huffman emprada a JPEG. La diferència principal en termes de capacitat de compressió és que WebP ocupa menys d'espai que JPEG en la mateixa qualitat. A més, WebP suporta transparència alfa, mentre que JPEG XR no ho fa.

	WebP	JPEG
Extensió	.webp	.jpeg .jpg
Submostreig de crominància	1:2 (YCbCr 4:2:0)	(YCbCr 4:2:0)
Codificador entròpic	Si	Si
Compressió amb pèrdues	Si	Si
Compressió sense pèrdues	Si	Si

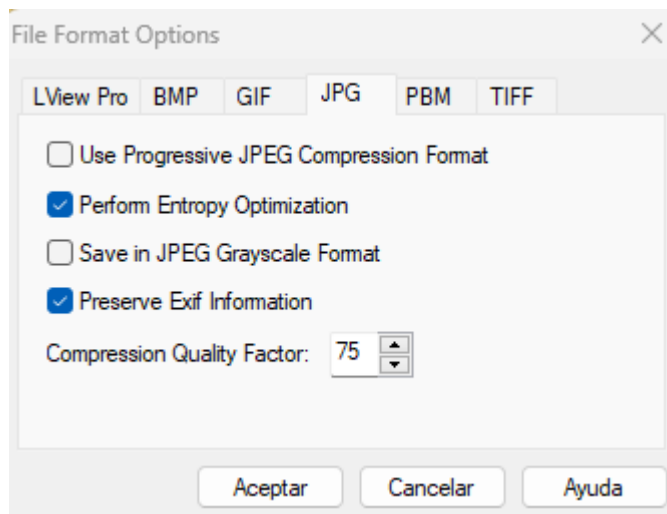
Exercici 4: Característiques de LViewPro

Instal·leu LViewPro. Obriu una imatge, feu “Save as”, i comenteu

Apartat a

Totes les imatges són guardades a partir de la mateixa imatge (birds.bmp dels documents de la unitat 4)

- JPG



Ens permet activar la compressió progressiva que augmenta la qualitat de la imatge a cada passada.

Per defecte està activada l'opció de optimització de l'entropia que busca reduir la redundància i augmentar la uniformitat en la distribució de les dades per millorar l'efectivitat de la compressió. És a dir, es calculen les taules de Huffman òptimes per aquell fitxer en concret, en comptes d'utilitzar les taules per defecte.

També ens permet activar l'opció d'escala de grisos que guarda la imatge amb tons grisos per reduir mida del fitxer.

Per defecte ve l'opció de guardar la informació EXIF que ens guarda informació de la imatge com el model de la càmera que l'ha fet, data i hora...

A més ens deixa escollir el factor de qualitat de la compressió.

Com sabem JPG comprimeix amb pèrdues, això li permet reduir molt la mida del fitxer sacrificant qualitat:

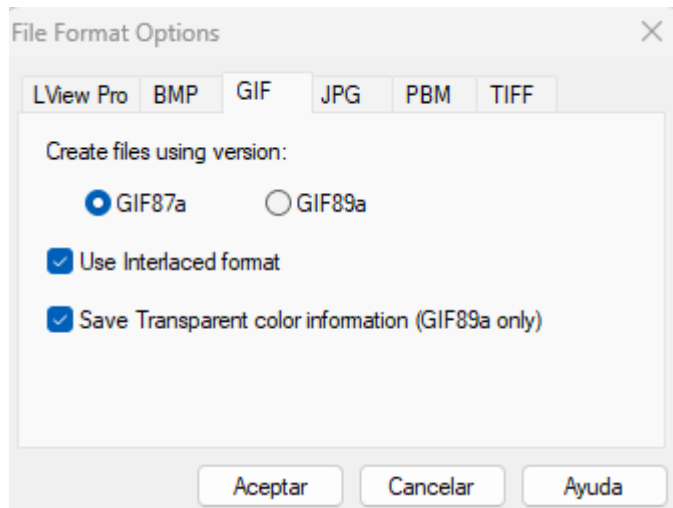
.bmp:

Tamaño:	1,12 MB (1.179.702 bytes)
Tamaño en disco:	1,12 MB (1.183.744 bytes)

.jpg:

Tamaño:	39,2 KB (40.145 bytes)
Tamaño en disco:	40,0 KB (40.960 bytes)

- GIF



Ens deixa escollir quina versió de GIF volem utilitzar, les diferències més importants són que a la versió 89a es va afegir la possibilitat d'animació mitjançant diverses imatges i la capacitat de designar un color dins la paleta de 256 colors (que és present a les 2 versions) com a transparent, que és el que fa l'opció "save transparent color information".

Per defecte utilitza entrellaçat que divideix la imatge en línies senars i parells, les transmet de manera alternativa i s'entrellacen per formar un quadre complet.

Sabem que GIF comprimenix mitjançant LWZ utilitzant diccionaris i sense pèrdues, però el que fa que disminueixi molt la mida és el fet que només pot guardar 256 colors:

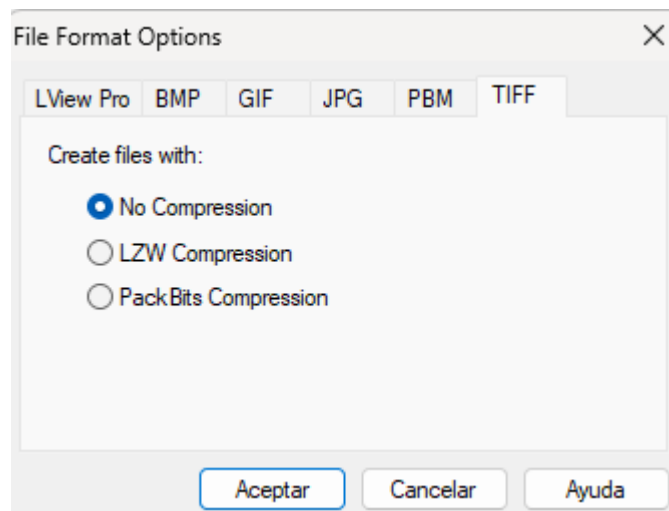
.bmp:

Tamaño:	1,12 MB (1.179.702 bytes)
Tamaño en disco:	1,12 MB (1.183.744 bytes)

.gif:

Tamaño:	222 KB (228.209 bytes)
Tamaño en disco:	224 KB (229.376 bytes)

- TIFF



Tenim l'opció de no comprimir i com es pot veure la mida del fitxer augmenta molt poc.

.bmp:

Tamaño:	1,12 MB (1.179.702 bytes)
Tamaño en disco:	1,12 MB (1.183.744 bytes)

.tif(No Compression):

Tamaño:	1,12 MB (1.183.884 bytes)
Tamaño en disco:	1,13 MB (1.187.840 bytes)

També tenim l'opció de comprimir amb LZW, que com hem vist és un mètode de compressió sense pèrdues, tot i que augmenta la mida del fitxer degut a que s'afegeixen metadades.

.bmp:

Tamaño:	1,12 MB (1.179.702 bytes)
Tamaño en disco:	1,12 MB (1.183.744 bytes)

.tif(LZW):

Tamaño:	1,18 MB (1.239.114 bytes)
Tamaño en disco:	1,18 MB (1.241.088 bytes)

L'última opció és comprimir amb PackBits, que com LZW és un mètode de compressió sense pèrdues.

.bmp:

Tamaño:	1,12 MB (1.179.702 bytes)
Tamaño en disco:	1,12 MB (1.183.744 bytes)

.tif(PackBits):

Tamaño:	1,13 MB (1.188.342 bytes)
Tamaño en disco:	1,13 MB (1.191.936 bytes)

- PNG

No ens donen opcions per guardar la imatge com a .png.

La combinación de una compresión sin pérdidas eficiente, el filtrado avanzado y la flexibilidad en la gestión de la profundidad de color hace que PNG sea extremadamente eficaz para reducir el tamaño de los archivos de imágenes mientras mantiene su calidad intacta.

La combinació d'una compressió sense pèrdues eficients, el filtrat avançat i la flexibilitat en la gestió de la profunditat de color fa que PNG sigui eficaç per reduir el tamany dels fitxers mantenint la qualitat.

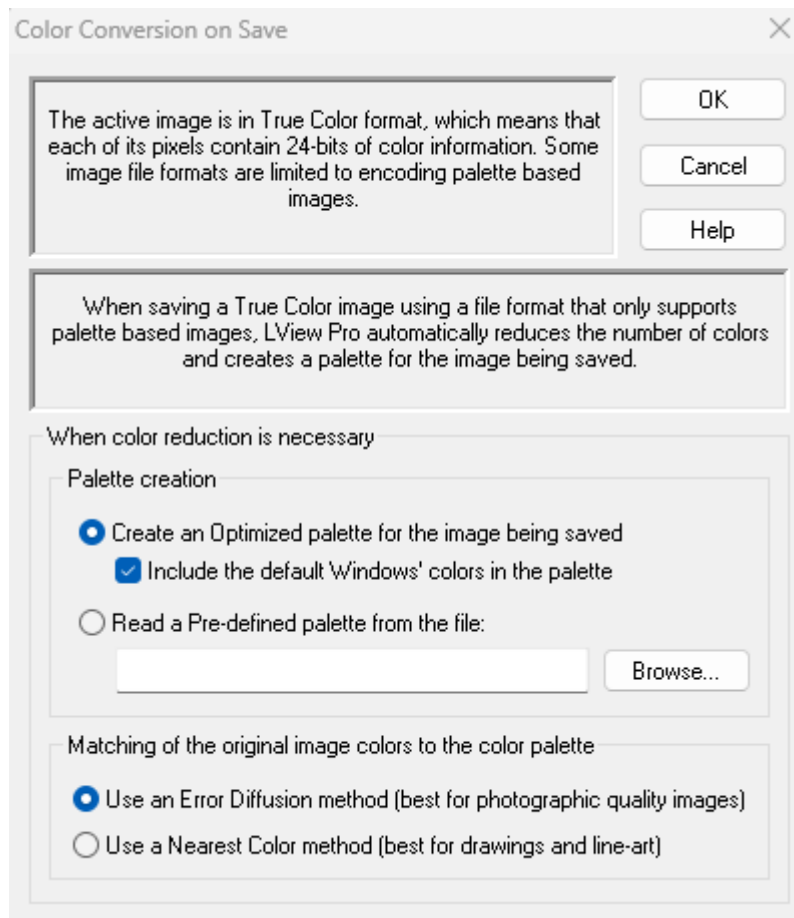
.bmp:

Tamaño:	1,12 MB (1.179.702 bytes)
Tamaño en disco:	1,12 MB (1.183.744 bytes)

.png:

Tamaño:	616 KB (630.793 bytes)
Tamaño en disco:	620 KB (634.880 bytes)

Apartat b



Com hem vist a teoria ens informa que cada pixel conté 24 bits. També ens diu que si guardem una imatge amb “True Color” (24 bits/pixel) a un tipus de fitxer que només accepta imatges basades amb una paleta de colors l'aplicació reduirà el nombre de colors de la imatge original i crearà una paleta de colors adient.

Ens permet canviar algunes opcions per quan aquest sigui el cas.

Ens dona l'opció de que l'aplicació crea la paleta o que llegeixi una que tinguem nosaltres guardada.

Per tal de que coincideixin els colors de la imatge a la nova paleta tenim 2 mètodes:

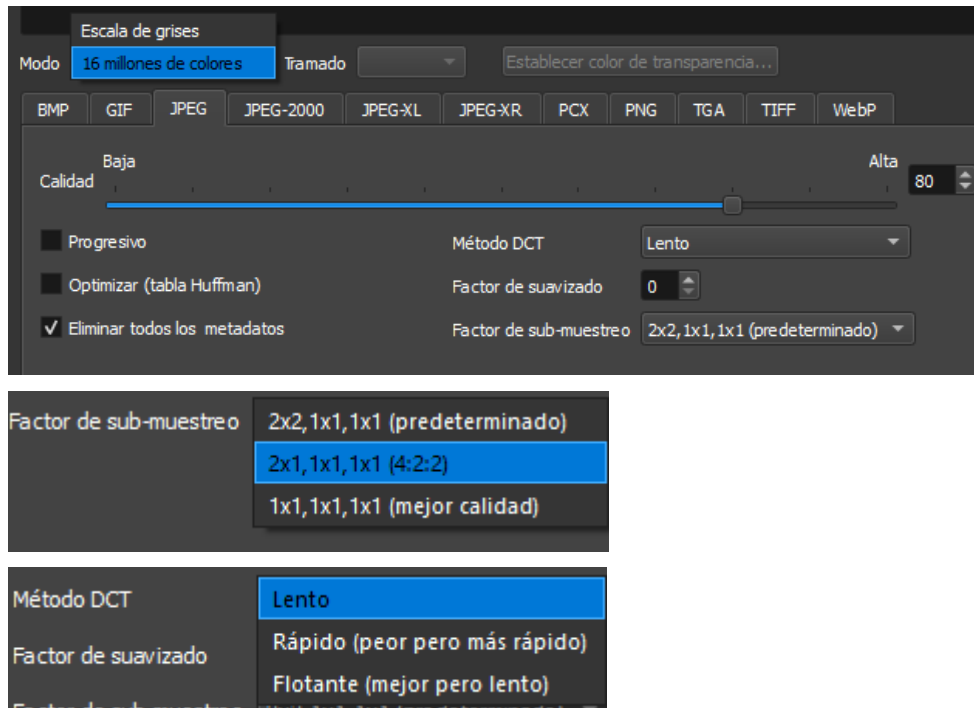
Error Diffusion method: La idea darrera d'aquest mètode és distribuir l'error introduït al reduir la paleta de colors als pixels veïns. Això es fa de manera que l'error no generi distorsions notòries en la imatge.

Nearest Color method: Aquest mètode assigna a cada color de la imatge original el color més proper dins de la paleta que tenim. És un mètode més simple i ràpid.

Exercici 5: Característiques de XnView MP

Instal·leu XnView MP. Obriu una imatge, feu “Export”, i comenteu els paràmetres per JPEG, GIF, PNG, TIFF, BMP, i JPEG 2000. Relacioneu-los amb el que heu après a classe sobre aquests formats, i compareu amb els que heu vist a LViewPro.

- JPEG



Com podem veure podem escollir si volem la imatge en escala de grisos o amb 16 milions de colors.

Ens deixa també establir una qualitat del 0 al 100. Activar o no l'opció de progressiu on la imatge es guardarà i carregarà de manera progressiva amb diverses passades. També ens deixa activar l'opció d'optimitzar amb una taula Huffman, com hem dit a l'exercici anterior això faria reduir molt la mida del fitxer degut a que hi hauran pèrdues.

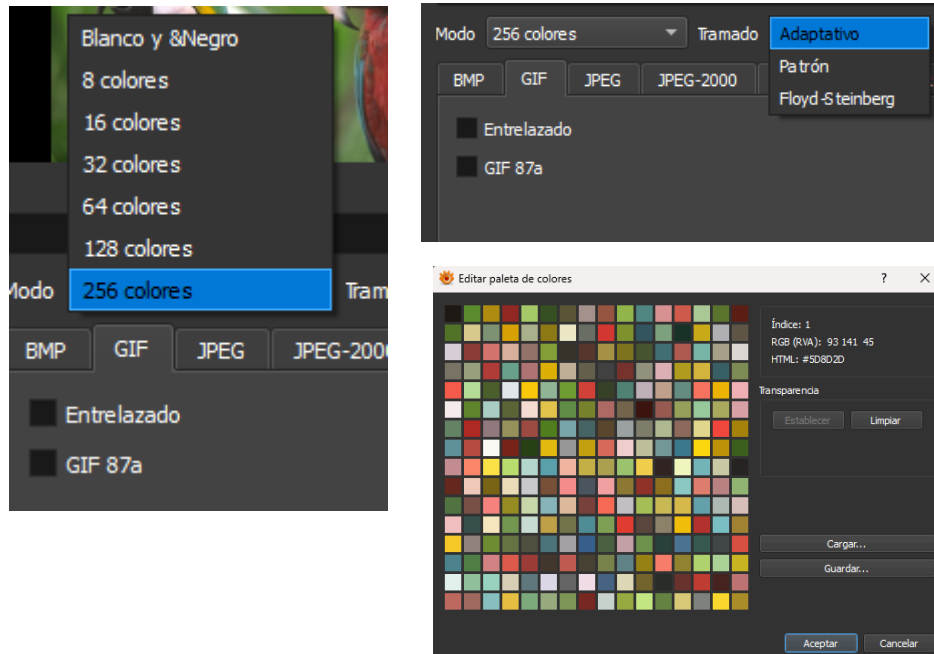
Ve per defecte l'opció d'eliminar metadades que hem explicat a l'exercici anterior, degut a que volem que la imatge resultat pesi el mínim possible.

Tenim opcions respecte al del mètode DCT que tradueix dels pixels a freqüències per després ser codificades mitjançant Huffman (en aquest cas).

També tenim un factor de “suavizado” que va del 0 al 10 i que redueix el factor de bloking, és a dir la creació de blocs després de comprimir la imatge.

Per últim tenim el factor de “sub-mostreo” que s'utilitza per reduir la quantitat de dades necessàries per emmagatzemar informació de color.

- GIF



Com podem veure ens deixa escollir més nivells de la paleta de colors que amb BMP, tot i que com sabem GIF està limitat a 256 colors.

Per corregir l'error al reduir els colors a una paleta de 256 colors podem escollir fer-ho de diferents formes:

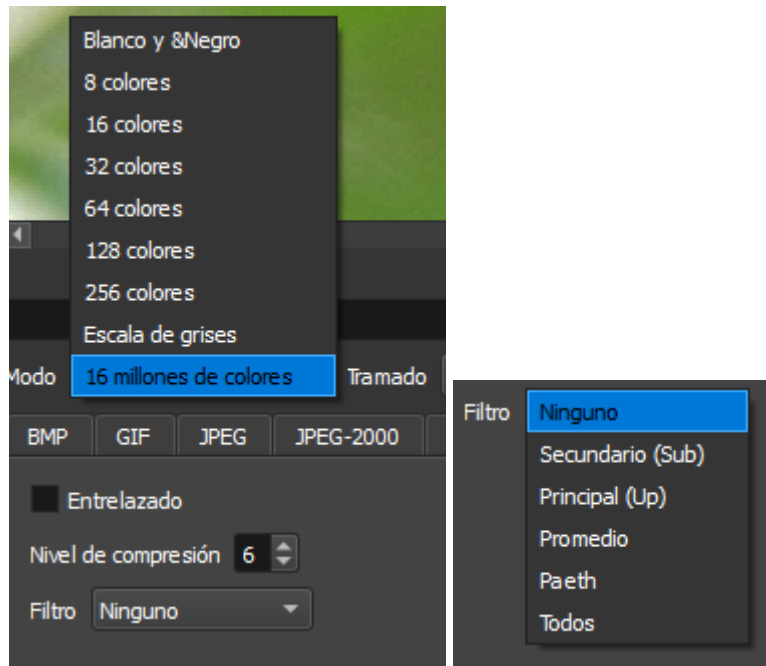
Amb tramat adaptatiu, que distribueix l'error en píxels veïns.

Amb tramat patró, que crea patrons que simulen diferents tons o colors.

Amb tramat Floyd-Steinberg, assigna a cada color un color de la paleta i distribueix l'error als píxels veïns.

A més ens deixa escollir quin dels colors serà el color transparent com es veu a la tercera foto.

- PNG



Ens permet escollir la paleta de colors com els altres formats.

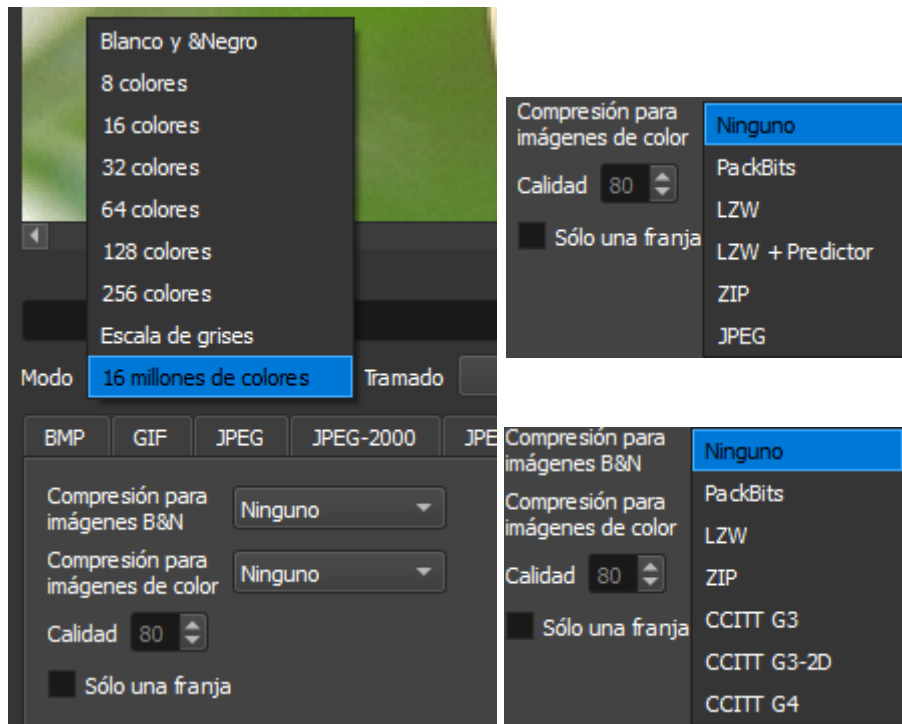
També apareix el tramat com amb GIF però no ens deixa escollir quin tipus.

Apareix un nivell de compressió que va del 0-9, que és una relació entre la compressió i la velocitat. Com més compressió més lent i a l'inrevés.

A més tenim l'opció de filtres que s'utilitzen per reduir diferències entre píxels perquè la compressió sigui més efectiva:

- Sub: Aquest filtre calcula la diferència entre el valor actual del píxel i el valor del píxel anterior a la mateixa fila.
- Up: El filtre Up calcula la diferència entre el valor del píxel actual i el valor del píxel a la mateixa posició a la fila anterior.
- Average: Aquest filtre calcula el promig entre el píxel anterior a la mateixa fila i el píxel a la fila anterior a la mateixa posició.
- Paeth: Preveu el valor del píxel actual a partir de tres píxels: l'anterior a la mateixa fila, el superior a la mateixa columna i l'anterior i superior diagonalment.

- TIFF



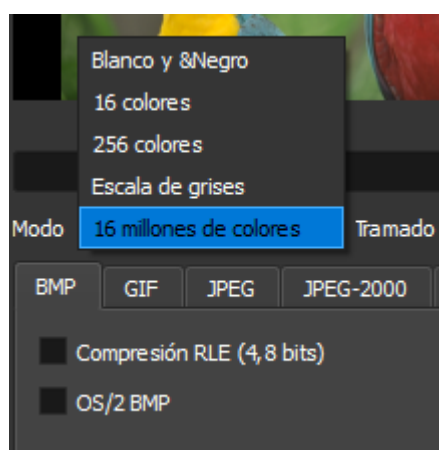
Veiem que podem escollir la paleta de colors.

A part podem escollir una qualitat de la imatge.

Per últim tenim mètodes de compressió per escollir diferenciats per color i per blanc i negre.

Podem veure el mètode LZW, que com PackBits són mètodes de compressió sense pèrdues.

- BMP

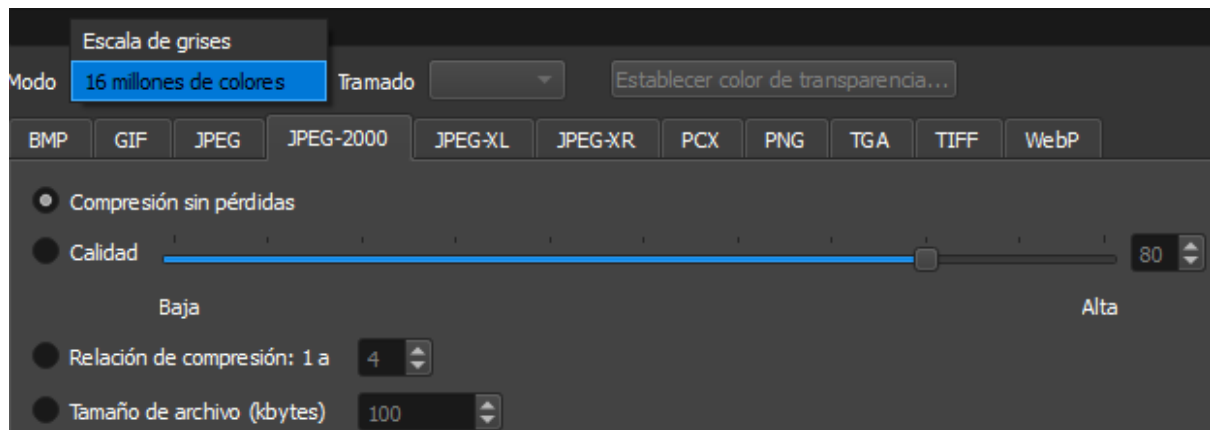


Com podem veure ens permet escollir la paleta de colors amb la que guardarem la imatge de manera diferent a LViewPro.

A més ens deixa activar la compressió RLE, que com sabem serveix per comprimir imatges sobretot amb grans àrees de color uniforme.

OS/2 BMP és un format d'imatge variant de BMP.

- JPEG-2000



A part de poder escollir entre 16 milions de colors i escala de grisos podem activar o desactivar la compressió sense pèrdues.
Podem establir una qualitat, una relació de compressió i podem establir una mida del fitxer resultant.

Exercici 6: FFmpeg

Paràmetres generals:

- -i: Especifica el fitxer d'entrada (BMP o TIFF).

Opcions per a cada format:

GIF:

Exemple:

Unset

```
ffmpeg -i input.bmp output.gif
```

PNG:

-

Exemple:

Unset

```
ffmpeg -i input.tiff output.png
```

JPEG:

- q:v: Controla la qualitat de la imatge (valors entre 1 i 100, 100 és la màxima qualitat).
- -crf: Controla la taxa de bits constant (valors entre 0 i 51, 0 és la màxima taxa de bits).

Exemple:

Unset

```
ffmpeg -i input.bmp -q:v 90 -o output.jpg
```

Control de la qualitat a JPEG:

- **-q:v`: Aquest paràmetre defineix la qualitat de la imatge JPEG en una escala de 1 a 100, on 1 és la pitjor qualitat i 100 la millor. Un valor més alt genera una imatge de major qualitat amb un tamany de fitxer més gran.

Exemple:

Unset

```
ffmpeg -i lena.tif -q:v 31 lena31.jpg
```

Aquesta comanda codifica l'imatge lena.tif a JPEG amb una qualitat de 31, resultant en una imatge amb menor pèrdua de detalls però un tamany de fitxer més petit que amb una qualitat més alta.

Exercici 7: Perspectiva de gènere

Apartat a

Als anys 70, els camps de processament de senyals i compressió d'imatges, com molts altres camps tècnics, eren majoritàriament masculins, igualment que els percentatges exactes d'homes no puguin ser fàcilment disponibles, està ben documentat que els camps de la ciència, tecnologia, enginyeria i matemàtiques, històricament han estat i continuen estant dominats per homes. Aquesta divisió es deu en gran part a les barreres socials, culturals i institucionals que impedièn que les dones tinguessin l'oportunitat d'accedir a aquestes àrees d'estudi. Aquests obstacles estaven enllaçats amb les expectatives socials que promovien que les dones seguissin el rol tradicional de casar-se, quedar-se a casa i en la majoria de casos ni tan sols tenir l'opció d'estudiar. I quan les dones tenien accés a l'educació superior, sovint triaven professions considerades més "femenines", com l'ensenyament o la infermeria, mentre que els camps tècnics i científics eren predominantment masculins.

Durant els anys 70, els rols de les dones en la societat també estaven evolucionant, la dècada va estar marcada per moviments significatius per la igualtat de gènere. Igualment, encara estaven poc representades en professions tècniques d'alt salari i d'estatus elevat, i sovint eren davant dels mitjans de comunicació i publicitat de maneres que reforçaven els rols de gènere tradicionals o les objectivaven. Aquesta representació no només limitava la percepció del que les dones podien aspirar a ser professionalment sinó que també influïa en com la societat en general veia les capacitats i els rols apropiats per a les dones.

L'elecció d'una imatge eròtica d'una dona de Playboy com a imatge de prova estàndard en la recerca de processament d'imatges pot ser vista com influenciada per diversos factors. La predominança d'homes en el camp podria haver portat a la falta de consciència o preocupació sobre les implicacions de la model al utilitzar la imatge. A més les tendències de l'època, que sovint objectivaven les dones, probablement van contribuir a que la elecció fos inquestionada entre els que seleccionaven la imatge. Igualment que també es pot justificar amb les qualitats tècniques de la imatge del seu alt contrast, nitidesa i les diferents textures, sigues triada des d'un punt de vista tècnic.

Apartat b

Durant els anys 70, era poc probable l'ús d'una imatge eròtica d'un model masculí com a estàndard en la investigació, donada la predominança masculina en els camps tècnics i les normes culturals de l'època que no contribuïen en la sexualització dels homes en contextos professionals. En canvi, al 2020, tot i una major obertura i igualtat de gènere, l'ús de qualsevol imatge eròtica, ja sigui masculina o femenina, en contextos acadèmics seria una opció conflictiva per raons d'objectivació i sensibilitat social, sent la tendència actual fotografies més neutres i respectuoses en la selecció d'imatges per a la investigació.

Apartat c

El críticisme públic sobre l'ús de la imatge "Lena" en la comunitat científica i tecnològica va començar a guanyar prominència a finals dels anys 90. Un exemple clau va ser l'escrit de 1999 de la matemàtica Dianne P. O'Leary, on criticava l'ús d'imatges suggerents, com la de "Lena", en les classes sobre processament d'imatges. O'Leary argumentava que aquestes imatges transmetien el missatge que els continguts estaven orientats només cap als homes, cosa que reflectia i potencialment reforçava la predominança masculina en camps com la informàtica. Aquesta crítica va destacar la necessitat de reconsiderar les pràctiques d'ensenyament i la selecció de materials en àmbits acadèmics i professionals per fomentar un entorn més inclusiu i respectuós.

Apartat d

Diverses organitzacions científiques i professionals han establert polítiques clares sobre l'ús de la imatge "Lena":

- Journal of Modern Optics (2017): Aquesta revista va publicar un editorial suggerint alternatives a "Lena", com "Pirate", "Cameraman" i "Peppers", que ofereixen característiques tècniques similars per provar tècniques de processament d'imatges sense la càrrega ètica.
- Nature Nanotechnology (2018): Aquesta revista va decidir no considerar articles que utilitzin la imatge de "Lena", mostrant un compromís amb l'actualització dels recursos utilitzats en la recerca científica i amb l'ètica.
- SPIE - Optical Engineering (2018): SPIE va anunciar que desaconsellava fortament l'ús de "Lena", acceptant nous treballs que la continguin només si hi ha una justificació científica convincent. També van destacar l'obsolescència tecnològica de la imatge a l'actualitat.
- IEEE (2024): L'IEEE va anunciar que a partir de l'1 d'abril de 2024, ja no permetrà l'ús de la imatge "Lena" en les seves publicacions, marcant un canvi significatiu cap a estàndards més ètics en recerca i publicació.

Aquestes polítiques representen un canvi cap a majors consideracions ètiques en la selecció d'imatges estàndard de prova i reconeixen que els estàndards tecnològics han d'evolucionar per mantenir-se rellevants i efectius en el testatge de les tecnologies modernes.

Apartat e

La matemàtica belga és Ingrid Daubechies. És una de les figures clau en el desenvolupament de la transformada wavelet, un concepte fonamental en el processament de senyals i la compressió d'imatges. La seva contribució més notable en aquest camp és el desenvolupament de la "wavelet de Daubechies", que ha tingut aplicacions extenses en diversos àmbits tecnològics.

En relació amb els còdecs d'imatge moderns, la feina de Daubechies ha influenciat directament el desenvolupament del format JPEG 2000. Aquest format de compressió d'imatges utilitza la tecnologia de wavelet per millorar la qualitat de les imatges comprimides

en comparació amb els estàndards anteriors, com el JPEG original, que utilitzava la transformada discreta de cosinus (DCT). JPEG 2000 permet una compressió més eficaç sense pèrdua de qualitat i amb millor gestió de la compressió amb pèrdua, essent particularment útil en aplicacions on la qualitat d'imatge és crítica.

https://ca.wikipedia.org/wiki/Ingrid_Daubechies