

# **TEMA 1. Introducción a la teoría del color.**

## ***1. INTRODUCCIÓN***

En esta parte del temario realizaremos una introducción a las principales técnicas relacionadas con la teoría del color y su representación en un computador.

## ***2. CONCEPTOS BÁSICOS EN EL DISEÑO Y PRODUCCIÓN DE IMÁGENES POR COMPUTADOR.***

### **Introducción**

El concepto de imagen desde el punto de vista de la informática ha ido evolucionando conjuntamente con el desarrollo de la tecnología, especialmente en los últimos años. El aumento de las prestaciones en los ordenadores personales, ha hecho que aparezca cada año un nuevo conjunto de herramientas gráficas que pretende superar ampliamente a la anterior generación. Además, la importancia capital de Internet en el campo de las comunicaciones y la informática ha influido enormemente en todas las áreas de la informática profesional y doméstica y como no podía ser de otra forma, en el ámbito del desarrollo de imágenes ya que estas son una parte esencial de los contenidos en la red.

Hay que comprender en primer lugar, que lo que comúnmente conocemos como una imagen desde el punto de vista de la máquina (el ordenador) no es otra cosa que un conjunto de informaciones estructuradas. Además, esa información debe admitir una representación tal que se pueda traducir de forma sistemática a / desde los diferentes dispositivos gráficos (pantalla del monitor, impresora, cámara digital, escáner de sobremesa). Estas representaciones informáticas de las imágenes pueden ser guardadas en los dispositivos de almacenamiento del ordenador, tales como memorias, discos duros, disquetes, CDs, etc... En cierta forma podemos entender que cualquier programa capaz de manejar imágenes en un ordenador tiene que contener una parte que funciona como un traductor entre el concepto humano de imagen que se representa en la pantalla del monitor y sufre transformaciones provocadas por el diseñador y los formatos de representación que le permiten al programa de ordenador almacenar en su memoria o en otros dispositivos de almacenamiento la imagen.

En el marco señalado al principio de esta introducción, surgen un gran número de herramientas (programas de ordenador) que ayudan al diseñador a llevar su idea gráfica a una realización adecuada sobre los medios previstos para su presentación. Así que en parte por la propia evolución de los programas hacia la explotación eficiente de los cada vez más amplios recursos computacionales y de comunicaciones, y en parte por la propia diversidad de los medios donde son aplicables las imágenes existen un gran número de los denominados *formatos*, que es la palabra técnica para una representación informática, en este caso de una imagen.

Las características de los formatos de imagen están fuertemente ligadas a la propia naturaleza de las imágenes y a el proceso perceptivo. En primer lugar podemos entender una imagen como una percepción subjetiva de la realidad en donde se involucran aspectos fisiológicos como la captación de la luz y el color, o de más alto nivel en el proceso de abstracción como la detección de líneas, formas y volúmenes, y en un nivel mucho más alto la identificación de los objetos que forman parte de esa percepción que en definitiva es una imagen. Siguiendo un paralelismo bastante claro, en este módulo veremos representaciones de la imagen donde esta aparece como un mosaico regular de bloques sobre el espacio bidimensional. Cada bloque se denomina píxel y guarda la información del color de la región del espacio sobre la que esta situado. Este tipo de representaciones se suelen denominar “bitmap” o mapa de bits en castellano. Por otro lado, tenemos otro tipo de representación donde la imagen es vista como un conjunto de líneas y áreas delimitadas con diferentes colores. Estos elementos se almacenan en el formato usando parámetros que establecen las características del elemento: por ejemplo si en la imagen hay un círculo situado en la mitad de la imagen, la representación incluirá para esta forma los parámetros: tipo de forma (círculo), posición en la imagen, color y otras. Este tipo de representación se denomina vectorial y junto con los mapas de bits conforman las dos grandes clases de formatos tradicionalmente usados. En este punto hay que comentar que hoy en día existen formatos que admiten los dos conceptos.

El formato de representación de la imagen no sólo afecta a las características del almacenamiento de la misma sino que de forma esencial al modo en que podemos crear / manipular la imagen en el ordenador, por lo que es necesario aprender los

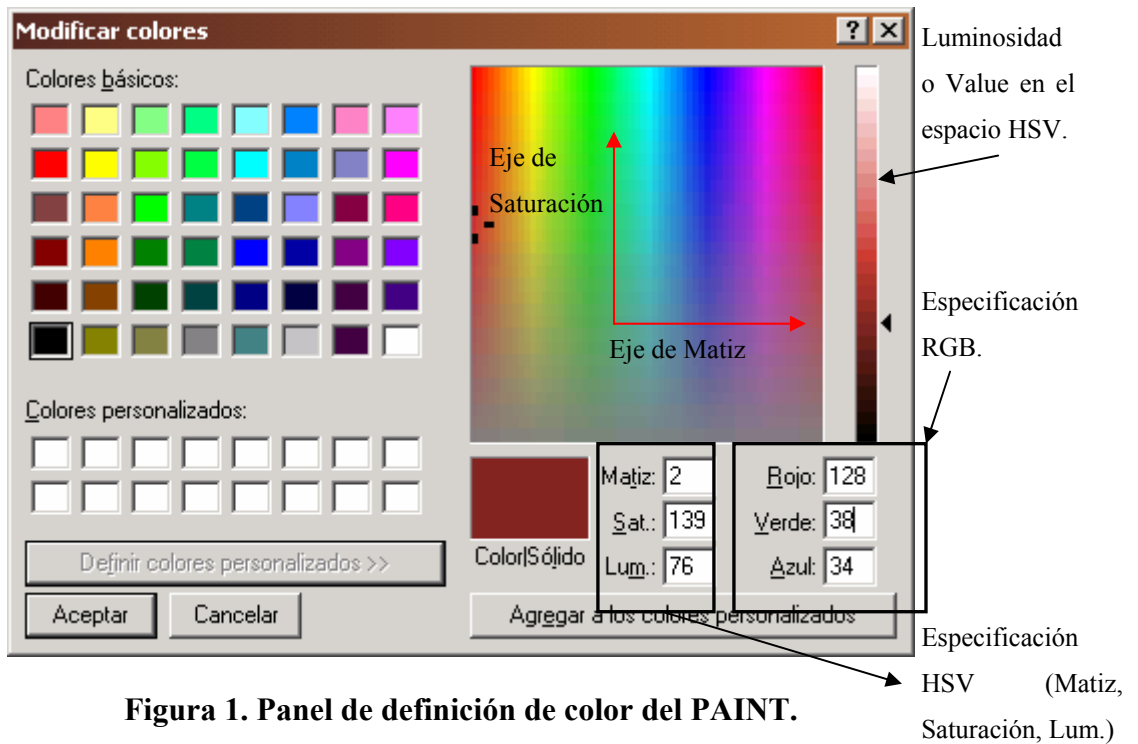
fundamentos más importantes de estas dos clases de formatos. A ello dedicaremos el resto de esta lección teórica.

## **El color**

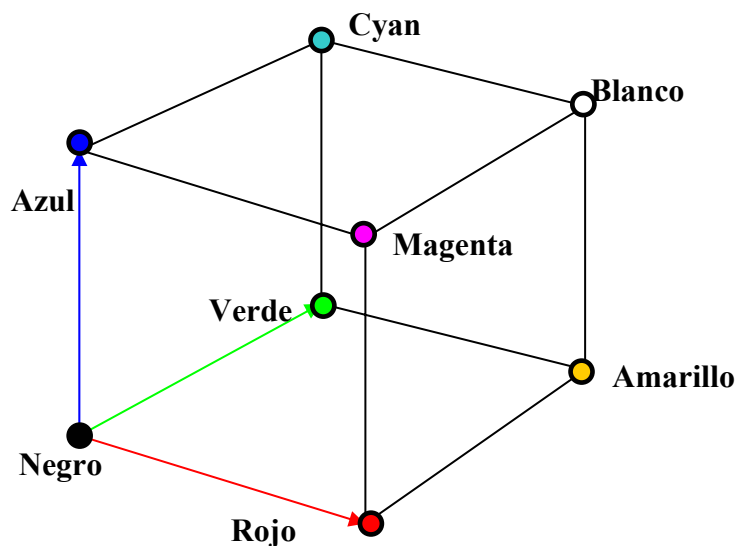
### El espacio de colores RGB

Los colores que observamos pueden construirse a partir de la mezcla de los colores denominados primarios rojo (red), verde (green) y azul(blue), cada uno con diferente intensidad. La intensidad del color primario establece una medida relativa de la cantidad de ese color producida en el dispositivo donde se visualiza la imagen. Por ejemplo, una intensidad de 0 en el canal primario rojo indica una cantidad 0 de color rojo. La intensidad máxima de color para cada canal depende del programa utilizado para generar el color y el dispositivo donde se va a representar. En el caso del programa GIMP esta intensidad máxima es 255 e indica la máxima cantidad de color primario que el dispositivo puede producir.

En teoría si la intensidad del color en los tres canales rojo, verde y azul es 0, el color producido es negro, mientras que si esta intensidad es 255 para los tres canales el color es blanco. En la práctica existen pequeñas diferencias entre los posibles dispositivos. El resto de colores se construyen usando diferentes intensidades para los canales RGB como se aprecia en la Figura 1. En el lado inferior derecho, vemos que se puede definir la cantidad de cada color Rojo, Verde o Azul para producir un nuevo color. En el mismo panel tenemos otras posibilidades para definir el color, como la definición en el espacio de colores Matiz, Saturación, Luminosidad que veremos más adelante.



Para entender el espacio de colores RGB es de especial utilidad lo que se denomina el cubo de colores. En el cubo de colores, cada eje es un color primario y cada punto dentro del cubo representa un color. Este cubo se ve en la figura, mostrando diferentes tripletas (R, G, B) y el color que producen.



**Rojo: (255, 0, 0) Verde: (0, 255, 0) Azul: (0, 0, 255)**  
**Negro: (0, 0, 0) Blanco: (255, 255, 255)**  
**Amarillo: (255,255,0) Cyan: (0,255,255) Magenta (255,0,255)**

El origen del cubo (0,0,0) representa la total ausencia de color (negro). La esquina opuesta representa la suma de las mayores intensidades de rojo, verde y azul, dando lugar al blanco. Las esquinas Rojo, Verde y Azul son los colores primarios y el resto de esquinas son los colores secundarios (Cyan, Amrillo y Magenta). La diagonal principal del cubo, la línea que va desde el negro al blanco, representa todos los puntos con igual cantidad de rojo, verde y azul que son los grises desde el más oscuro (negro) hasta el más claro (blanco). A este eje se le denomina eje neutro, ya que en los grises no predomina ningún matiz.

### El espacio de colores HSV.

En la Figura 1 vemos que también es posible obtener colores rellenando unos campos denominados matiz, saturación y luminosidad o en inglés hue, saturation y value. El matiz de un color es su localización dentro del espectro de colores: las palabras rojo o verde se refieren al matiz de dos colores. Por otra parte, la saturación representa la pureza del matiz respecto al blanco. Un color verde totalmente saturado lo podemos imaginar como una pintura verde que no ha sido mezclada con pintura blanca para rebajarla. Si se mezcla con pintura blanca el verde aparece más pastel y ya no está saturado. Finalmente un color tiene brillo o luminosidad y representa la cantidad de luz que el color refleja. Menos luz reflejado implica que percibimos el color de forma más oscura. Una medida del brillo, aunque no es la única, es el denominado Value. En la Figura 1, la barra situada más a la izquierda indica las diferentes luminosidades del color.

La relación entre el brillo y el espacio de colores RGB es la siguiente. Dos colores cuya suma de contenidos  $R + G + B$  es la misma tienen el mismo brillo. El color que más brillo tiene es el blanco  $R=255$   $G=255$  y  $B=255$ , no existiendo otro color con tanto brillo. Si nos movemos por el eje neutro del cubo de color RGB hacia el negro en planos perpendiculares tenemos los diferentes brillos en relación al color blanco. A la mitad del eje, tenemos el 50% de brillo en el plano perpendicular (la suma de las componentes  $R + G + B$  suman la mitad que para el blanco). El color con el menor brillo es el negro  $R + G + B=0$  y no hay otro color con menos brillo. Hay varias medidas para el brillo en GIMP se usan las siguientes:

Cantidad de luz:  $L = [\max(R,G,B) + \min(R,G,B)]/2$

Valor (value)  $V = \max(R,G,B)$

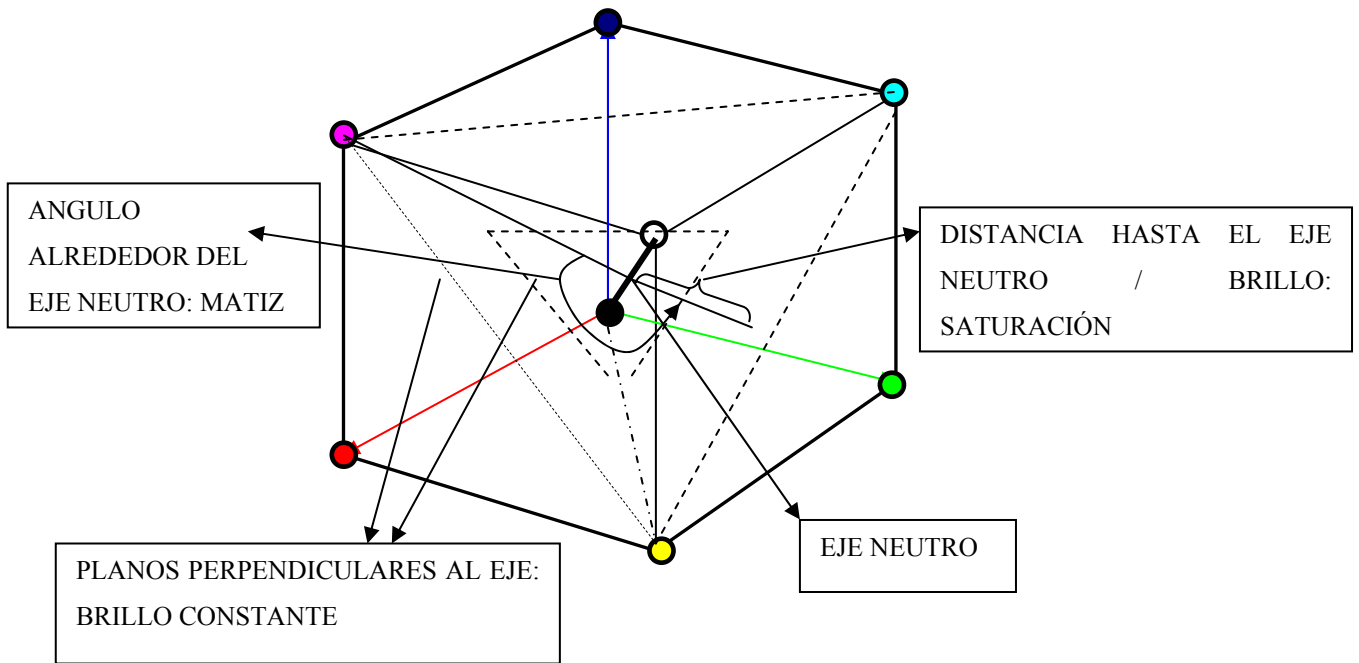
$$\text{Luminancia } Y = 0.30 * R + 0.59 * G + 0.11 * B$$

La luminancia es una medida del brillo que coincide bastante bien con la percepción humana del brillo.

La saturación también está relacionada con la representación del color en el espacio de colores RGB. El eje neutro en el cubo contiene tonos de grises y estos para nosotros se caracterizan por la ausencia de color (matiz). La cantidad de color de un punto en el cubo de colores RGB podemos medirla como la distancia desde el punto hasta el eje neutro. Cuanto más cercanos nos encontremos de este eje, menos cantidad de color apreciamos. La saturación es el cociente entre la cantidad de color y el brillo. De esta forma si nos movemos en un plano perpendicular al eje neutro (brillo constante), la saturación aumenta a medida que nos alejamos por el plano perpendicular de este eje.

El concepto de matiz está relacionado con lo que comúnmente entendemos por color. El matiz de un punto dentro del cubo de color RGB se define como la posición angular respecto del eje neutro. Todos los puntos que mantienen el mismo ángulo respecto del eje neutro tienen el mismo matiz. El matiz del punto Amarillo (255,255,0) es compartido por todos los puntos situados en el plano que forma un ángulo respecto al eje neutro igual al ángulo que le corresponde el punto (255,255,0).

Como resumen y posibles reglas a recordar, podemos observar que en el cubo de color RGB, los puntos Cyan, Magenta y Amarillo son más brillantes que los puntos Rojo, Verde y Azul, ya que los primeros se encuentran en un plano perpendicular al eje neutro más cercano al color blanco que los últimos. Lo mismo cabe decir de los puntos situados en la pirámide formada por los puntos Cyan, Magenta, Amarillo y Blanco y los puntos situados en la pirámide formada por los puntos Rojo, Verde, Azul y Negro. Además la cercanía del punto al eje neutro incrementa el aspecto pastel o acuso del color ya que disminuye la saturación.



La corrección gamma (parte de estos apuntes han sido extraídos de la recomendación del consorcio W3 sobre la especificación del formato PNG)

El proceso sufrido por una imagen desde que es capturada hasta que es mostrada en un dispositivo supone un conjunto de transformaciones que finalmente resultan en que la imagen representada en el dispositivo es diferente de la imagen original. Esto ocurre independientemente del tipo de imagen y formato.

Los cambios que se producen en la imagen se deben a la existencia de respuestas no-lineales en los diferentes elementos intermedios. Es decir, el dispositivo en cuestión, por ejemplo una pantalla, recibe una señal electrónica que debe ser convertida en intensidad de luz. Sin embargo, en contra de lo que sería deseable, la intensidad no es proporcional a la señal recibida, sino que sigue una función no lineal más complicada con una dependencia entrada salida del tipo:

$$\text{Salida} = \text{Entrada}^{\text{Gamma}}$$

Nota: esta ley es válida cuando tanto Entrada como Salida han sido normalizadas en el rango  $[0,1]$ , siendo 0 la intensidad / señal mínima y 1 la intensidad / señal máxima.

Se trata por lo tanto de una función no-lineal con un exponente de valor Gamma. Lo deseable sería que Gamma tomara el valor 1 con lo que tendríamos un efecto proporcional, sin embargo el valor real de Gamma suele estar por encima de 1.5 en las pantallas de ordenador (2.2 en un PC). Este razonamiento también se puede aplicar a los dispositivos de captura, en este caso la entrada es la intensidad luminosa y la salida es una señal electrónica.

Podemos plantearnos ahora la siguiente pregunta. ¿Cómo se combina este efecto en dispositivos reales que actuando sucesivamente estén en el camino del procesamiento de la imagen?. Se puede demostrar que en la mayoría de los casos es válida la suposición de que la composición de estos dispositivos puede modelarse con la misma función potencia donde el valor de gamma es el producto de las gammas asociadas a las diferentes componentes.

Si la gamma del sistema completo es 1.0 significa que hay una relación lineal entre la imagen de entrada y la imagen de salida. De esta manera, la relación de intensidad (cociente) entre dos puntos de la imagen original es la misma que entre los dos puntos correspondientes en la imagen de salida. Podríamos pensar que esta es la situación ideal, sin embargo esto no es tan sencillo.

El problema es que hay que considerar el entorno donde se va a visionar la imagen de salida. Este entorno afecta a la capacidad perceptiva del individuo, por lo que no siempre la representación fidedigna de la imagen original es deseable. Por ejemplo, si la imagen final va a ser proyectada en una habitación a oscuras, si usamos una representación fidedigna de la imagen original (con  $\gamma = 1$ ), tendremos la sensación de poco contraste y naturalidad en la imagen. En esa situación es más aconsejable un gamma superior, de 1.5. De hecho las películas para diapositivas tienen un gamma en torno a este valor. Si existe luz ambiente, pero esta luz es inferior a la de la imagen visionada (por ejemplo, ver la televisión en una habitación con luz natural al caer la tarde), el gamma apropiado puede estar en 1.25. Como vemos, la determinación del gamma adecuado requiere el conocimiento del tipo de entorno en el que la imagen será visualizada.

En la mayoría de los casos no dispondremos de esta información, por lo que por regla general se suele utilizar un gamma entre 1.0 y 1.25.



El parámetro gamma de un monitor suele ser de 2.5 (esto puede variar, en monitores modernos que incorporan tecnología para ajustar este parámetro). Para conseguir un gamma sobre 1.0 necesitamos un componente no lineal que añada al proceso que actúe de forma compensatoria. La pregunta es ¿dónde situar este compensador de gamma?

En los sistemas de difusión de video (una cámara captura la imagen y esta es enviada a los receptores) el sistema compensador suele estar en la propia cámara. Esta decisión se tomó para abaratar el coste de los receptores de televisión, colocando el circuito electrónico analógico encargado de la compensación en la cámara. El sistema original NTSC requería cámaras con un gamma de 0.45. En la actualidad se usa un gamma en las cámaras de 0.5, de forma que cuando la imagen se muestra en un monitor CRT con gamma 2.5, la gamma resultante está en 1.25.

Los programas de renderizado de imágenes por ordenador suelen producir imágenes con variaciones lineales en la intensidad. Si queremos que esta forma de la variación se mantenga en la representación en el monitor, hemos de compensar el gamma introducido por este último. La técnica utilizada normalmente es la aplicación de una tabla de búsqueda (o lookup table (LUT) en inglés), previa a la visualización en el monitor. Es decir, la intensidad almacenada es convertida a otra usando una tabla. Estas tablas de búsqueda pueden implementarse en el hardware del propio monitor o de la tarjeta gráfica y en realidad tratan de simular una ley de potencias, con un gamma aproximadamente de 0.4.

Obsérvese como en el caso de la difusión de video la corrección se realiza antes del almacenamiento de los datos, mientras que en el caso de la renderización la corrección se realiza a posteriori. El problema de realizar la corrección antes o después del almacenamiento de los datos tiene cierta importancia.

Debemos tener en cuenta que los datos se almacenan normalmente tratando de usar el mínimo número de bits, es decir hay una pérdida de precisión y dos niveles de intensidad inicialmente considerados diferentes al ser almacenados pasan a ser el mismo. Si esto sucede, la corrección gamma no funcionaría bien a posteriori ya que no se podría aumentar o disminuir la diferenciación entre niveles de intensidad. Esto obliga a utilizar más bits (al menos 12 por canal) en la codificación de imágenes a las que se pretende aplicar una corrección gamma a posteriori. En cambio, si la corrección se realiza antes de almacenar la imagen se pueden utilizar menos bits sin perder el efecto de la corrección.

Consideremos un proceso que incluye la captura (o cálculo) de la imagen, el almacenamiento en un fichero, la lectura del fichero y mostrar la imagen en un dispositivo. En estos procesos tenemos 5 fuentes de gamma. Usaremos las denominaciones típicas del formato PNG:

1. camera\_gamma: la gamma característica del sensor de imagen.
2. encoding\_gamma: la gamma de cualquier transformación realizada por el software que almacena la imagen en un fichero.
3. decoding\_gamma: la gamma de cualquier transformación realizada por el software que lee la imagen del fichero.
4. LUT\_gamma: la gamma que resulta de la aplicación de una corrección por tabla de búsqueda, si existiera.
5. CRT\_gamma: la gamma del monitor que normalmente es 2.5.

Además, tenemos la siguiente nomenclatura:

1. file\_gamma: la gamma de la imagen en el fichero, relativa a la escena original:  
$$\text{file\_gamma} = \text{camera\_gamma} * \text{encoding\_gamma}$$
2. display\_gamma: la gamma del sistema de visualización, que incluye la corrección LUT y el monitor CRT:  
$$\text{display\_gamma} = \text{LUT\_gamma} * \text{CRT\_gamma}$$
3. viewing\_gamma : la gamma resultante de todo el proceso y que normalmente debe estar entre 1.0 y 1.5.  
$$\text{viewing\_gamma} = \text{file\_gamma} * \text{decoding\_gamma} * \text{display\_gamma}$$

En los sistemas de video digital, la camera\_gamma está en 0.5, la CRT\_gamma en 2.5 y encoding\_gamma, decoding\_gamma y LUT\_gamma valen 1. De esta manera se consigue un viewing\_gamma de 1.25.

La mayoría de las estaciones de trabajo y monitores de PC no tiene una LUT para corregir previa a la visualización por lo que el display\_gamma es de 2.5. Por el contrario, los Macintosh suelen tener implementado en hardware una corrección LUT con gamma 0.72, dando un display\_gamma de 1.8. Algunos Macintosh tienen un panel de control para gamma que permiten obtener valores para display\_gamma de 1.0, 1.2, 1.4, 1.8 y 2.2.

## Los colores seguros del navegador.

Si bien los programas de diseño o la especificación HTML nos permiten usar una gran cantidad de colores (normalmente codificados con 24 bits de profundidad), aplicaciones como los navegadores web ofrecen limitaciones. La limitación software impuesta se basa normalmente en las características de la tarjeta de video. Esto quiere decir, que los navegadores no muestran toda la variedad de colores que puedan componer nuestro diseño, sino una fracción de ellos realizando las conversiones adecuadas.

En este sentido surgen dos problemas: ¿cómo podemos estar seguros de que nuestra página web será mostrada tal y como la hemos diseñado?, y ¿será mostrada del mismo modo en todos los navegadores?.

La clave está en la denominada **paleta segura**. La paleta segura es un conjunto de colores establecidos y compartidos por la mayoría de los navegadores. Esta paleta consta de 256 colores. Estos colores se codifican según sus componentes RGB, representándose su valor de intensidad. Los valores de intensidad utilizados en esta paleta sólo pueden ser algunos de los siguientes: 0, 51, 102, 153, 204 y 255 (00, 33, 66, 99, CC y FF)

## **Los formatos “mapa de bits” o raster-only formats.**

### La resolución.

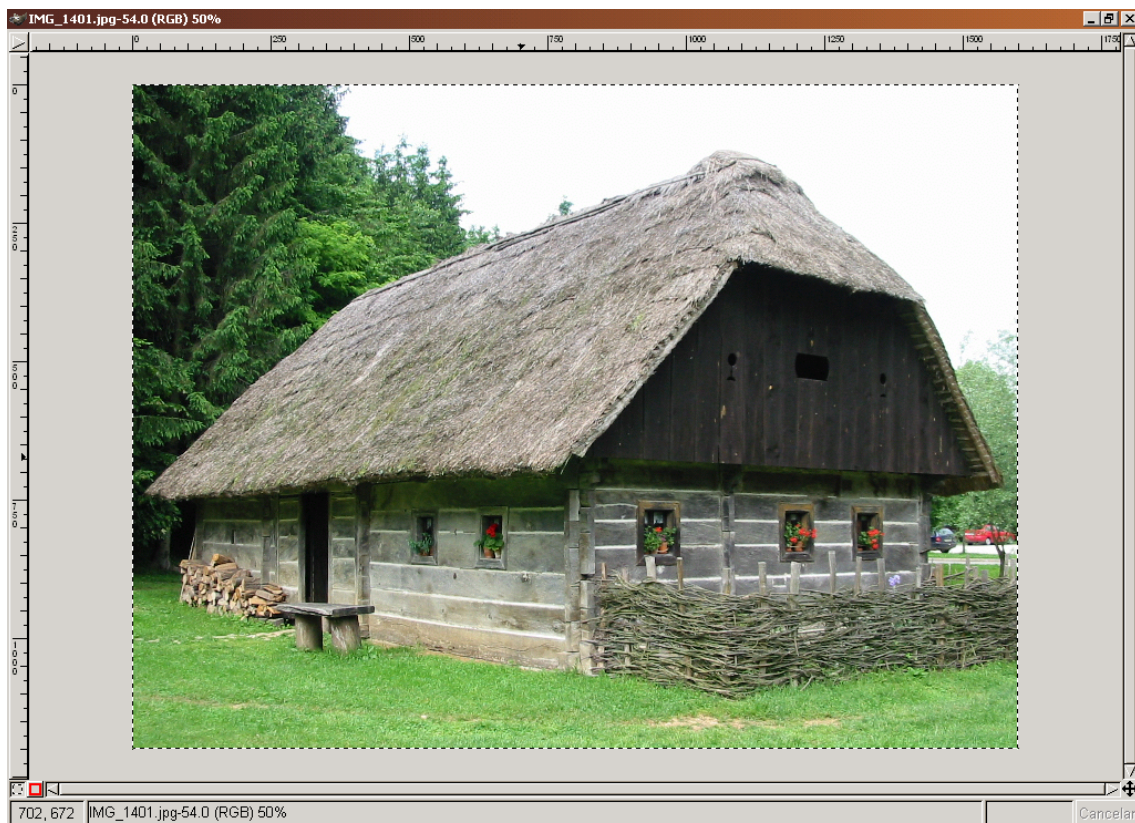
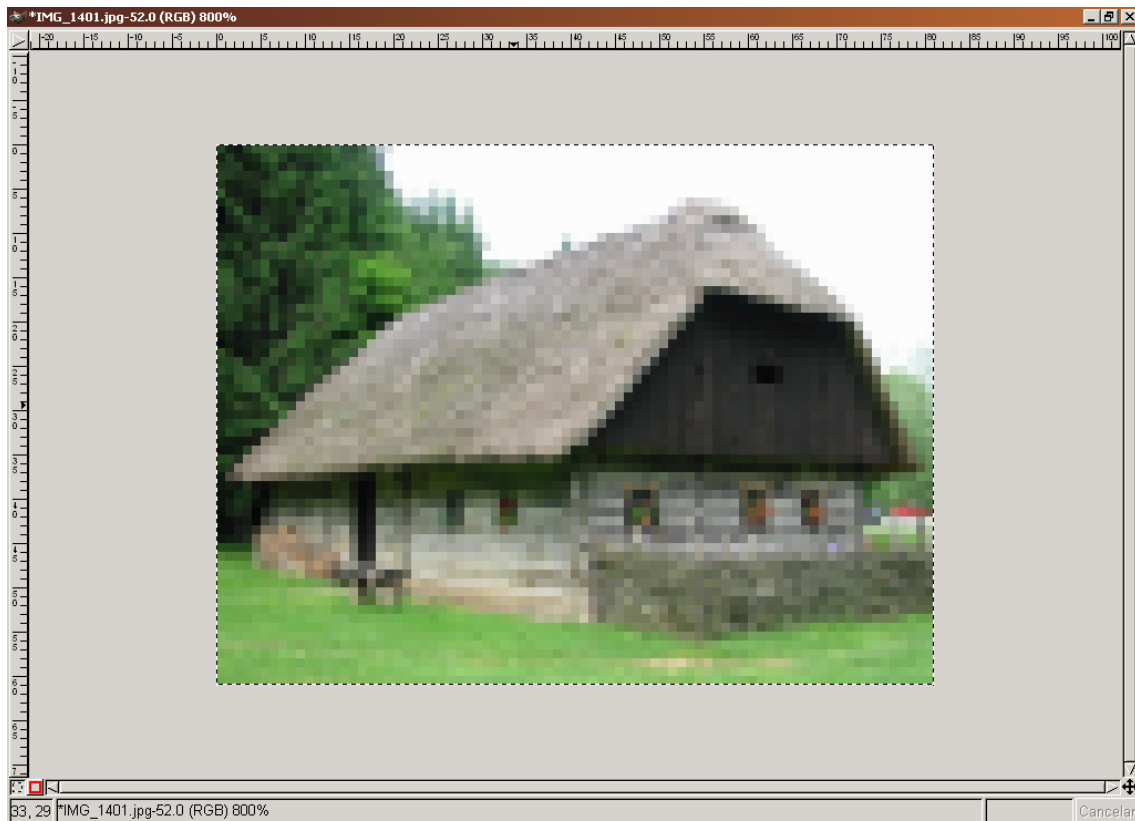
Las representaciones mapa de bits se basan en establecer una cuadrícula sobre la imagen, de forma que cada rectángulo de la cuadrícula contiene la información sobre el color predominante en esa región de la imagen. Cada rectángulo de esta retícula se denomina píxel. El número de píxeles por unidad de área o longitud constituye lo que se denomina resolución de la imagen. Por ejemplo, en un monitor la resolución se mide en píxeles o puntos por pulgada (ppi pixels per inch). La resolución del monitor depende de las características técnicas del mismo. La mayoría de monitores nuevos tienen una resolución de 96 píxeles por pulgada. La representación de mapa de bits de una imagen es traducida directamente al monitor de forma que cada píxel en el monitor corresponde con un píxel de la imagen. Esto tiene como consecuencia que si la resolución del monitor es mayor que la resolución de la imagen original, la imagen aparecerá en un

tamaño menor en el monitor (en una pulgada caben más píxeles en el monitor que en la imagen original). Un ejemplo: si se quiere mostrar una imagen de 1 pulgada por 1 pulgada con una resolución de 144 ppi sobre un monitor de 72 puntos por pulgada (72-dpi), aparecerá sobre el monitor con unas dimensiones de 2 pulgadas por 2 pulgadas, ya que para reunir 144 píxeles donde mostrar la imagen hacen falta 2 pulgadas por 2 pulgadas del monitor.

Con las impresoras pasa otro tanto. La mayoría de las impresoras láser de sobremesa tienen una resolución de 600 dpi (600 puntos por pulgada). Por el contrario las impresoras de chorro de tinta tienen una resolución de 300 a 600 dpi. Este tipo de información hay que considerarla a la hora de deducir el tamaño de la imagen y la calidad resultante cuando se quiere representar una imagen.

La representación de la imagen en un mapa de bits tendrá mayor calidad mientras mayor sea la resolución (para una misma área un mayor número de píxeles implica un tamaño menor del píxel, con lo que se representará con un mejor nivel de detalle la imagen). En la Figura 2 podemos apreciar la diferencia en el uso de dos resoluciones diferentes sobre una imagen. En la figura superior se puede apreciar el efecto de cuadrícula típico que surge cuando la resolución es demasiado baja .

El precio a pagar por una mayor resolución es la necesidad de una mayor capacidad de almacenamiento. Para tomar una decisión sobre la resolución con la que una imagen debe representarse hay que tener en muy en cuenta entre otros factores la capacidad del dispositivo donde se va a mostrar la imagen. Por ejemplo si la resolución y tamaño de nuestro monitor es tal que dispone de 600 píxeles de alto por 800 de ancho, y queremos representar al completo una imagen en él no podremos usar una imagen con 900 píxeles de alto por 1300 de ancho ya que no cabrá en la pantalla a menos que redimensionemos la imagen para adaptarla al dispositivo.



**Figura 2. Dos imágenes con diferente resolución. La superior tiene mucha menos resolución y ha sido ampliada a un tamaño similar a la original: obsérvese la pérdida de detalle y la peor calidad.**

### La representación del color.

Como se ha mencionada más arriba cada píxel de la imagen debe almacenar información acerca del color. Tal como se discutió, el color se puede representar por un conjunto de parámetros en diferentes espacios de colores. Por ejemplo en el espacio de color RGB necesitaremos tres parámetros para indicar las intensidades del rojo, el verde y el azul. Estos tres parámetros deben ser almacenados en cada píxel.

En este punto debemos profundizar un poco en la manera en la que los números son almacenados en un ordenador. El elemento básico es el bit. Un bit puede representar una elección entre dos opciones, por ejemplo blanco o negro. Si la imagen que tenemos que almacenar fuese en blanco y negro cada píxel podría contener un solo bit para indicar si el color asociado al píxel es blanco o negro. De esta forma, una imagen de 10 por 10 pulgadas con una resolución de 96 ppi tiene 9600 píxeles. Como cada píxel para una imagen en blanco y negro requiere 1 bit, harían falta en general 9600 bits.

Sin embargo hacen falta más de dos colores para representar con cierta calidad una imagen. Para ello usaremos más bits. Por ejemplo, con dos bits podemos representar cuatro. En general  $n$  bits permiten  $2^n$  opciones que para nosotros son posibles colores diferentes. El término profundidad de color hace referencia precisamente a esto. Una imagen representada con una profundidad de color (color depth) de 24 bits quiere decir que cada píxel representa un color de entre  $2^{24}$  (posibles). Este sería por ejemplo el caso de usar el espacio de colores RGB y guardar cada posible intensidad del rojo, el verde y el azul en 8 bits. Como son 3 intensidades necesitaremos:  $3 \times 8 = 24$  bits. El código 0 0 0 0 0 0 0 representa el 0 y el código 1 1 1 1 1 1 1 representa el 255 que es el mayor valor posible. La mayoría de los programas que usan una representación en el espacio RGB trabajan con esta forma de representar la intensidad entre los valores 0 y 255 con una representación interna de 8 bits para cada canal rojo, verde y azul y por lo tanto una profundidad de color por píxel de 24 bits.

Evidentemente el uso de más colores amplía los requerimientos de memoria. Para representar la misma imagen del ejemplo descrito para dos colores hacen falta 9600 píxeles  $\times$  24 bits = 28.125 Kbs, es decir 24 veces más como es lógico. Sin embargo como veremos más abajo la cantidad de memoria necesaria para almacenar la imagen

depende del formato en que ésta se almacene, por lo que estas medidas sólo son válidas si la imagen se almacena píxel a píxel con su representación de color asociada.

Los formatos de imagen del tipo mapa de bits.

A nadie se le escapa de lo dicho anteriormente que pueden existir mil formas de representar y ordenar esta información asociada a los píxeles. Atendiendo a las necesidades técnicas y a los recursos disponibles han ido apareciendo a lo largo de los últimos años una gran cantidad de estrategias para almacenar esta información. La lista es extensa y los detalles de cada uno de los formatos abundantes. Esto hace inviable pormenorizar todos y cada uno de ellos. En la tabla que se acompaña se muestra una lista de los más usuales con algunos datos relevantes. Además prestaremos especial atención a los más importantes Windows BitMap (WBMP), Graphic Interchange Format (GIF), Portable Network Graphics (PNG) , Tag Image File Format (TIFF) y el estándar JPEG.

BIFF	Formato para el programa XITE 3D
BMP	Formato bitmap de MS-Windows
BW	Formato para blanco y negro de SGI
CGM	Computer Graphics Metafile
DRAW	Formato de imagen vectorial orientado al objeto
DWG	Formato del programa AutoCAD
FAX	El estándar Facsimil
EPSF	Postscript encapsulado
FIG	Usado por la utilidad xfig.
FITS	Flexible Image Transport System
GIF	Graphics Interchange Format
GL	Formato para animación
HDF	Formato de datos jerárquico
ICC	Used for Kodak printer
IFF	Interchange Format Files
JPEG	JPEG File Interchange Format
MIF	Machine Independent Format
NAP	Formato orientado al objeto NAPLPS

netCDF	Network Common Data Form
PIX	Usado por los productos SGI Alias/Wavefronts
PCX	Formato usado por el programa PC Paintbrush
PNG	Portable Network Graphics
PBM+	Enhanced Portable Bitmap toolkit
RLE	Run Length Encoded Format (compression sin pérdidas se basa en la redundancia de las repeticiones).
RAS	Formato de la empresa SUN para imágenes mapa de bits.
RGB/RGBA	Formato de SGI para imágenes en color.
SLD/SLB	Formato de AutoDesk para presentaciones
SPRITE	Formato de la empresa Acorn para un Sistema Operativo para máquinas RISC
TGA	Targa File Format
TIFF	Tag Image File Format
VIFF	Del programa Khoros
X	The AVS Image Format
XBM	X BitMap Format
XPM	X PixMap Format
XWD	X Windows Dump Format

### ***Windows Bitmap***

Cada fichero contiene cuatro partes: un encabezado de fichero bitmap, un encabezado de información bitmap, una tabla de color y un conjunto de bytes que establece los bits incluidos en el bitmap.

El encabezado de fichero contiene información sobre las características del propio fichero, en concreto tipo, tamaño y distribución.

El encabezado de información bitmap contiene información sobre la codificación de la imagen en el archivo: dimensiones de la imagen, tipo de algoritmo de compresión y formato de color del bitmap.



La tabla de color es una característica muy importante dentro de este formato. Podemos imaginarnos esta tabla como un casillero donde cada casilla contiene 24 bits para representar un color. Con este casillero en lugar de utilizar 24 bits para hacer referencia al color asociado a un píxel podemos referenciar el número de casilla dentro del casillero. Supongamos que tenemos un bitmap con sólo un bit por píxel. En este caso sólo podemos hacer referencia a dos casillas, pero cada casilla puede tener un color de 24 bits. Esto supone un ahorro en muchos casos. Por ejemplo: en una imagen de 10 por 10 pulgadas con resolución 96 ppi, y un solo bit por píxel podemos codificar haciendo uso de la tabla de color dos colores de 24 bits en cada píxel con: 9600 bits en el mapa de bits y 48 bits más para los dos colores en la tabla de color es decir 9648 bits. Si codificáramos asociando 24 bits a cada píxel necesitaríamos 230400 bits.

El concepto de tabla de color que también se denomina paleta de colores se utiliza en muchos formatos y programas de dibujo. Otra gran ventaja es que podemos cambiar de una vez todos los colores de la imagen sin más que modificar los colores en la tabla de color.

Dentro de la cabecera de información sobre el bitmap se indica el número de bits dedicados a cada píxel y el número máximo de colores en el bitmap. Hay diferentes posibilidades: 1 bit por píxel o dos colores (monocromo), 4 bits por píxel o 16 colores, 8 bits por píxel o 256 colores o 24 bits por píxel. En este último caso no se hace uso de la tabla de color sino que se codifica directamente en formato RGB el color asociada a cada píxel.

Otra característica importante del formato Windows bitmap es que utiliza un tipo de codificación que se puede encuadrar en los formatos de compresión sin pérdidas. La compresión es un concepto que se aplica en informática en el sentido de evitar el uso de información redundante o poco importante para la aplicación con la finalidad de ahorrar espacio en memoria y/o tiempos de comunicación. Veamos un sencillo ejemplo. Supongamos la siguiente secuencia de números:

1 3 3 3 3 3 5 5 5 5 2 2 2 2 2 2 2 2 2 2 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4

A la hora de almacenar esta secuencia podemos optar por almacenar cada número por otra estrategia que reduzca las necesidades de almacenamiento. Podemos poner para cada número su valor y el número de veces que se repite. En el ejemplo propuesto, la secuencia se codifica como:

(1,0) (3,5) (5,5) (2,11) (3,4) (4,16).

Se aprecia que el número de elementos de la secuencia codificada es menor. Esta es la versión más primitiva de la técnica conocida en inglés como Run Length Encoding. En el caso de las imágenes esta técnica es beneficiosa cuando aparecen grandes zonas con el mismo color: en lugar de almacenar repetitivamente para cada píxel el mismo color, se cuentan las repeticiones sucesivas del color y se almacena este número junto a la codificación del color. La técnica se denomina compresión sin pérdidas porque se elimina información redundante pero se puede decodificar la secuencia de datos original exactamente a partir de su versión codificada. El formato Windows bitmap, codifica de esta forma aquellas imágenes con 4 bits por píxel y 8 bits por píxel (16 y 256 colores). Los resultados suelen ser mejores cuando la imagen proviene de dibujos para ordenador en lugar de imágenes reales, ya que en el caso de los dibujos la variedad de cambios en los colores es menor y suelen aparecer grandes superficies con el mismo color. Para las imágenes reales este tipo de codificación no es el mejor, por lo que las fotografías codificadas en Windows bitmap suelen ocupar una gran cantidad de memoria.

Estas son las características más relevantes del formato Windows bitmap, para una información más detallada se puede consultar: [www.dcs.ed.ac.uk/home/mxr/gfx/2d/BMP.txt](http://www.dcs.ed.ac.uk/home/mxr/gfx/2d/BMP.txt) (en inglés).

### ***Graphics Interchange Format (GIF)***

El formato de intercambio de gráficos o GIF está especialmente diseñado para la transmisión en línea de datos gráficos tipo mapa de bits entre ordenadores, siendo uno de los formatos gráficos más usados en internet. De hecho es un estándar MIME nativo para los navegadores. Fue desarrollado por la empresa CompuServe a finales de los años 80. El formato GIF utiliza el algoritmo de compresión LZW (Lempel-Ziv y Welch) de la empresa UNISYS.

Un conjunto de datos GIF, denominado GIF data stream puede contener uno o más imágenes con diferentes parámetros. Este hecho es aprovechado en el diseño de páginas web para incluir animaciones simples o secuencias de gráficos.

El decodificador del archivo GIF es un programa o parte de un programa que se encarga de leer paso a paso el GIF data stream e ir interpretándolo para producir los gráficos en el dispositivo de destino. A medida que va leyendo el GIF data stream puede ir encontrando diferentes gráficos e ir mostrándolos uno detrás de otro.

El formato GIF también se basa en el uso de tablas de color al igual que el Windows bitmap. De hecho el máximo número de colores que se puede emplear en un GIF es 256, por lo que el uso de color indexado es importante. Los archivos GIF pueden tener varias tablas de color locales (una para cada gráfico incluido en el archivo) y varias tablas de color globales usadas para aquellos gráficos que no dispongan de una tabla de color local. Las tablas de color también van usándose secuencialmente por parte del decodificador. Por ejemplo, el decodificador encuentra una tabla de color global y a continuación aparece un gráfico con una tabla de color local. El decodificador salva la tabla global y usa la tabla local para producir el gráfico. Continúa la decodificación y encuentra un gráfico sin una tabla de color local. Entonces, restaura la tabla global y con ella representa el gráfico. La lectura continúa y encuentra una tabla de color global nueva, que sustituye a la existente.

El formato GIF tiene además otras particularidades interesantes, sobre todo desde el punto de vista de la decodificación. Se pueden introducir campos denominados Delay Times (tiempos de retraso) que indican que el decodificador debe esperar un tiempo antes de continuar decodificación. Esto se usa para controlar el ritmo de una animación. Otra característica interesante es el uso del denominado Transparency Index. Este índice indica píxeles que el decodificador no debe modificar en el dispositivo de presentación del gráfico. De esta manera se consigue un efecto de transparencia entre el fichero procesado y el fondo sobre el que se está mostrando el gráfico. El transparency index actúa realmente como un cuarto canal adicional a los tres típicos R, G y B.

Es muy importante recordar que el formato GIF usa una variante del algoritmo de compresión sin pérdidas (ver explicación en Windows BITMAP) denominado LZW, mucho más eficiente que el mencionado RLE para el Windows Bitmap. En cualquier caso se trata también de un algoritmo sin pérdidas con lo que la imagen originalmente codificada se obtiene exactamente igual tras el proceso de decodificación.

Otro aspecto reseñable es la posibilidad de usar GIF entrelazado. Los GIF entrelazados los reconocemos cuando descargamos una página web, ya que la imagen aparece inicialmente con poca calidad y posteriormente se va perfilando. El entrelazado no afecta a la velocidad o al tamaño del GIF.

### ***Introducción al formato JPEG.***

Aunque más adelante se profundizará en este formato, dada su importancia describiremos aquí algunas de sus principales características.

JPEG es un algoritmo de compresión con pérdida de calidad, es decir parte de la información de la imagen es eliminada. JPEG se especializa en eliminar justo aquella información que es menos relevante en la percepción de la imagen con cierta calidad. Esto es una gran diferencia con el formato GIF donde no hay pérdida de información.

Otra característica es que se puede ajustar el nivel de compresión en JPEG, evidentemente a costa de aceptar una pérdida de calidad. Esta es una de las razones de su popularidad en la web: los diseñadores pueden diseñar optimizando para calidad en los gráficos o reducción del ancho de banda consumido.

JPEG trabaja siempre con 24 bits por píxel (8 bits por canal), por lo que todo lo relacionado con paleta de colores e indexación no se utiliza. Esta característica proviene del hecho de que fue diseñado para imágenes fotográficas.

Podemos encontrarnos con dos tipos de JPEG: el progresivo y el estándar. El primero permite incorporar el efecto de entrelazado.

### ***El formato PNG.***

Este formato es apoyado por el consorcio W3 debido a sus características especialmente adecuadas para la web. PNG utiliza compresión sin pérdidas.

A diferencia de GIF o JPEG, PNG puede almacenarse a muchas profundidades de bit diferentes utilizando diversos medios de almacenamiento. Por ejemplo, GIF sólo se puede almacenar a profundidades de bit de 8 bits o menores y JPEG de 24 bits. PNG puede almacenarse a 8, 24 o 32 bits.

Una característica importante es que PNG se creó de manera que se puede especificar en el formato las características de la plataforma donde fue creado, de manera que al menos en teoría, la imagen pueda ser recreada de forma fidedigna. Por ejemplo, se puede tratar de incorporar la misma corrección gamma. Sin embargo, en la práctica esto es dificultoso pues normalmente no se conoce el valor de gamma. Esta característica no ha sido desarrollada por parte de los fabricantes de navegadores.

El formato PNG dispone de un auténtico canal de transparencia alfa donde se pueden establecer 256 niveles de transparencia parcial. En PNG se utiliza por lo tanto un esquema de 4 canales RGBA (RGB + alfa).

Es interesante repasar la especificación de PNG más reciente en <http://www.w3.org/TR/PNG/>.

### ***El formato TIFF.***

TIFF es el acrónimo de Tagged Image File Format. Se trata de un formato muy popular para el intercambio de imágenes que no requieren la capacidad de ser secuenciadas (streamability). Esta capacidad significa que los elementos de la imagen pueden ser decodificados progresivamente, mostrándose sin necesidad de llegar a decodificar todo el fichero. Esta capacidad si bien no es necesaria en muchas aplicaciones, es importante en otros casos como por ejemplo la web. Por este motivo TIFF no se utiliza normalmente en páginas web.

Si hablamos del TIFF estándar (el formato a sufrido extensiones a lo largo del tiempo), tiene algunas limitaciones. Se trata de un formato de mapa de bits (no soporta gráficos vectoriales). No se pueden incluir información adicional como por ejemplo texto para anotar la imagen.

TIFF presenta varias características que aunque lo hacen más flexible, pueden dar problemas. Por ejemplo, el orden en el que se representan enteros de 32 bits puede variar: primero colocamos el byte más significativo (MSB) o primero colocamos el byte menos significativo (LSB). La primera forma es característica de Motorola, mientras que la segunda es de Intel. Pues bien, el TIFF especifica en la cabecera como debe ser este orden. Sin embargo, hay programas que usan TIFF que asumen un orden (dependiendo de la plataforma) haciendo caso omiso a la cabecera. En este caso el TIFF no podrá ser decodificado correctamente.

TIFF se usa mucho en el ámbito científico y en el intercambio de ficheros entre profesionales de la impresión, pues en su cabecera pueden incluirse parámetros de calibración para los colores, corrección gamma, etcétera. TIFF soporta un número de canales elevado (hasta 64K), lo que lo hace especialmente adecuado para almacenar imágenes de detectores que operan con múltiples bandas de frecuencia. Las imágenes pueden representarse en diversos esquemas de color: grayscale, pseudocolor, RGB, YCrCb, CMYK, CIE Lab, etcétera.

En cuanto a algoritmos de compresión el estándar también acepta una variedad de los mismos: sin compresión (datos en crudo), PackBits (RLE), LZW, CCITT Fax, JPEG sin pérdidas. El formato de los píxeles incluye enteros de 1 a 64 bits o números en punto flotante de 32 y 64 bits según el estándar IEEE.

Debemos recordar que existen extensiones de TIFF que incorporan otras características como por ejemplo la denominada JPEG en TIFF que permite incorporar JPEG con pérdidas dentro del formato. También debemos señalar el ZIP-TIFF que usa un algoritmo de compresión no propietario en lugar del LZW que es uno de los mejores algoritmos de compresión del TIFF estándar pero que adolece de estar bajo la patente de Unisys.

NOTA: Utilización de tramados

La reducción de colores diferentes en una imagen que tiene millones de colores implica necesariamente una pérdida en su calidad. Existen dos maneras de realizar este proceso: el tramado y el efecto banda.

El tramado es la colocación de píxeles de distintos colores dentro de una imagen que utiliza una paleta de 256 colores para simular un color que no existe en dicha paleta. Una imagen con tramado suele presentar ruido o crear la impresión de que está compuesta por píxeles dispersos.

Existen varios tipos de tramado. El tramado adaptado normalmente es el mejor de los métodos de tramado. En este caso se utiliza una paleta de 256 colores extraídos de los existentes en la imagen. En contraposición, tenemos el tramado de pantalla. En este caso la paleta está fijada y los colores de los píxeles se modifican para tratar de simular los colores permitidos por la paleta externa.

Si no se aplica una operación de tramado y se realiza la reducción de colores normalmente se produce el denominado efecto banda. El resultado son zonas de color sólido en las degradaciones.

La utilización de tramado aumenta el tamaño de los ficheros que utilizan formatos de compresión, pero es una alternativa a considerar si se quiere prescindir del efecto banda. La mayoría de los programas de diseño disponen de algoritmos de tramado.