



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Qué hay dentro de un archivo de imagen en mapa de bits

Apellidos, nombre	Vicente Atienza Vanacloig (vatienza@disca.upv.es)
Departamento	Informática de Sistemas y Computadores (DISCA)
Centro	Escuela Técnica Superior de Ingeniería Informática

1 Resumen de las ideas clave

Este artículo trata sobre la codificación de archivos de imágenes en mapa de bits. Por mapa de bits nos referimos a las habituales imágenes formadas por una matriz de píxeles, que se guardan en archivos con formatos populares como GIF, BMP, JPG, etc. De una forma práctica y un punto de vista informático se explorarán las estrategias básicas que permiten que mediante la secuencia ordenada de bytes que constituye todo archivo digital se represente una imagen, posibilitando su almacenamiento e intercambio. En la redacción de este artículo se ha asumido que el lector posee conocimientos elementales sobre algunos términos informáticos básicos como *bit*, *byte*, *archivo*, *sistema de numeración binario* o *representación hexadecimal*.

2 Introducción

Las imágenes que capturan las cámaras digitales, las que se crean o editan mediante ordenadores o se intercambian a través de redes o soportes digitales deben codificarse mediante unos y ceros en forma de archivos digitales. La gran mayoría de estos archivos utilizan la estrategia de describir el contenido de la imagen en forma de una matriz de puntos de color elementales (los píxeles). Básicamente el archivo contendrá una secuencia de valores binarios que describen ordenadamente (típicamente por filas, de izquierda a derecha y de arriba abajo) el color de cada uno de estos píxeles. Este tipo de imágenes se conocen como **mapas de bits** (*bitmaps*) o imágenes *raster*.

La descripción más frecuente del color de cada píxel se realiza indicando la intensidad de cada uno de los tres colores primarios en el sistema RGB (rojo, verde y azul) cuya combinación aditiva en esas proporciones da como resultado el color buscado. Cada uno de los bytes que componen un archivo puede codificar un número entero, de 0 a 255. En un archivo de imagen, cada tres bytes consecutivos permiten describir el color de un píxel en RGB. El primer byte describe la intensidad de la componente roja del color (en una gama de 256 matices de rojo), el segundo la verde y el tercero la azul. Aunque a veces el orden es justamente el contrario.

CUESTIÓN 1. Demuestra que entiendes esta idea contestando a esta sencilla pregunta: ¿Cuántos bytes ocupará en un archivo el mapa de bits de una imagen de, digamos, 8 Megapíxeles?

La idea básica es sencilla, sin embargo sabemos que hay diferentes formatos de archivos para imagen (JPEG, GIF, TIFF,...) ¿Qué los diferencia? ¿Por qué una misma imagen guardada en diferentes formatos da como resultado diferentes tamaños de archivo? Por otro lado, ¿cómo es posible que el programa que abrió el archivo sepa la localización GPS donde se hizo esa foto o el modelo de cámara que se usó? ¿Hay un formato de archivo que se considere el mejor de todos?

3 Objetivos

Al finalizar este artículo, serás capaz de:

- Predecir de forma aproximada el tamaño que tendrá un archivo de imagen (no comprimido) a partir del tamaño en píxeles de la imagen que va a contener.
- Explicar por qué el tamaño de un archivo de imagen no se corresponde exactamente con el tamaño del mapa de bits y citar algunas de las informaciones adicionales almacenadas.
- Justificar la diferencia de tamaño observada en un archivo de imagen según el modo de color escogido para la codificación.
- Citar algunos formatos de imagen con posibilidad de almacenar imágenes indexadas.
- Interpretar la documentación explicativa de la estructura de un archivo de imagen.
- Explorar el contenido de un archivo mediante un editor hexadecimal.
- Identificar el formato de algunos archivos a través de su "número mágico".

4 ¿Qué hay dentro de un archivo de imagen?

De las consideraciones anteriores sin duda ya sospechas que ni todos los archivos codifican las imágenes de la misma manera, ni guardan exclusivamente los valores RGB de cada píxel, sino alguna información más.

Un ejercicio interesante sería indagar en el contenido de un archivo de imagen con una herramienta de inspección adecuada. Todos los archivos de ordenador son en esencia lo mismo: una sucesión ordenada de bytes que expresan valores de 0 a 255. Para entender el significado de cada uno de estos bytes hay que acudir a la documentación adecuada donde se describa el formato concreto de archivo bajo estudio. Esta información describirá las diferentes partes que componen el archivo y el significado detallado de cada una de ellas. En un archivo de imagen no es diferente.

Así que podríamos crear una imagen sencilla, elegir un formato no demasiado sofisticado, sin compresión, guardar el archivo y explorar cada uno de sus bytes con ayuda de una herramienta que lo permita. Estas herramientas son los editores de archivos o **editores hexadecimales**, que se llaman así porque suelen expresar el contenido de cada byte del archivo representándolo en hexadecimal. Se trata de una representación más compacta y manejable que el binario, aunque estrechamente relacionada con él. En hexadecimal, el valor de cada byte resulta en un "número" de 00 a FF, que corresponde a los valores decimales 0 y 255 respectivamente. Se trata de números de dos cifras, donde cada cifra puede tomar no diez, sino 16 valores, por ello hay que recurrir a las 6 primeras letras del alfabeto para expresar los valores de cifra superiores a 9.

¡Comencemos entonces! La Figura 1 muestra la minúscula imagen con la que vamos a trabajar. Se trata de un pequeño mapa de bits de 4 x 4 píxeles con diferentes colores.



Figura 1. Imagen ejemplo de 4 x 4 píxeles

Una vez guardada en formato BMP (durante el proceso, escogimos la opción de codificar el color con tres bytes RGB como se explicó antes), localizamos el archivo *img4x4-rgb.bmp* creado y lo abrimos con un editor hexadecimal.

La Figura 2 muestra el contenido de este archivo tal como lo presenta el programa editor. Este editor no conoce nada sobre las particularidades del formato de archivo que está explorando. Simplemente muestra la sucesión de valores en hexadecimal de los bytes que lo componen. Cada fila muestra el valor de 16 bytes consecutivos. Los encabezados en azul permiten numerar cada uno de estos bytes. No hay más que anteponer el número de la fila al de la columna y ya tenemos expresada (en hexadecimal) la posición del byte en el archivo. Vemos entonces que el programa muestra los bytes 00h al 65h (les ponemos detrás una h a partir de ahora a los números hexadecimales para no confundirlos con valores decimales). Esto supone un total de 66h bytes. Para obtener el equivalente decimal hay que hacer $6 \times 16 + 6 = 102$ bytes, que es, por tanto, el tamaño de nuestro archivo.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	42	4D	66	00	00	00	00	00	00	00	36	00	00	00	28	00	BZh.....6... (
1	00	00	04	00	00	00	04	00	00	00	01	00	18	00	00	00
2	00	00	32	00	00	00	12	0B	00	00	12	0B	00	00	00	00	..2.....
3	00	00	00	00	00	00	FF	00	FF	FF	00	FF	FF	00	FF	FFÿ.ÿÿ.ÿÿ.ÿÿ
4	00	FF	00	00	00	80	80	80	F0	F0	F0	FF	FF	FF	00	00	.ÿ...€€€ðððÿÿÿ..
5	FF	00	FF	FF	00	FF	00	FF	00	00	FF	FF	00	FF	FF	00	ÿ.ÿÿ.ÿ.ÿ..ÿÿ.ÿÿ.
6	FF	FF	00	FF	FF	00											ÿÿ.ÿÿ.

Figura 2. Edición hexadecimal del archivo *img4x4-rgb.bmp*

Frecuentemente los editores hexadecimales se usan para examinar archivos que contienen texto, en parte o en su totalidad, por lo que en la parte derecha verás para cada fila los 16 caracteres que corresponden si se entendiera que estos bytes estuvieran codificando texto en código ASCII. Pero sabemos que en nuestro caso esto no es así, lo que esperamos que codifiquen es el color de nuestros píxeles de forma numérica, así que en principio esta parte derecha no nos servirá de mucho.

De entrada, hay algo que no cuadra. Nuestro archivo tiene $4 \times 4 = 16$ píxeles y se ha escogido una variante de BMP de 3 bytes por píxel para expresar sus colores: $16 \times 3 = 48$ bytes. Éste debería ser el tamaño del archivo, sin embargo se observa que es bastante mayor: 102 bytes. Parecen confirmarse las sospechas de que un archivo de imagen guarda algo más que el valor de color de los píxeles.

Para estar en condiciones de interpretar el significado de cada byte es preciso recurrir a información de referencia acerca del formato BMP. Lo primero que nos van a decir [1] es que, efectivamente, el archivo se compone de varias partes y sólo una de ellas corresponde a los valores RGB del **mapa de bits**. Esta parte está marcada en azul en la Figura 3. Observa que este bloque, efectivamente, sí que tiene un tamaño de 48 bytes. Puedes intentar identificar cada uno de los píxeles de la imagen mediante su color. El orden en que aparecen las 3 componentes de color son primero B, luego G, por último R. El primer píxel guardado (bytes 36h, 37h y 38h) corresponde por tanto a los valores $RGB = FF00FFh$, es decir una mezcla de rojo y azul a su máxima intensidad. Esto corresponde al color magenta. Y sí, nuestra imagen contiene tonos magenta ¡en su fila inferior! Acabamos de descubrir una particularidad de los archivos BMP: normalmente el mapa de bits se guarda fila a fila, **de abajo arriba**. Puedes rastrear el resto de tripletas de color y buscar su correspondencia con los píxeles de la imagen. No encontrarás más sorpresas.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	42	4D	66	00	00	00	00	00	00	00	36	00	00	00	28	00	BMh.....6... (
1	00	00	04	00	00	00	04	00	00	00	01	00	18	00	00	00
2	00	00	32	00	00	00	12	0B	00	00	12	0B	00	00	00	00	..2.....
3	00	00	00	00	00	00	FF	00	FF	FF	00	FF	FF	00	FF	FFÿ.ÿÿ.ÿÿ.ÿÿ
4	00	FF	00	00	00	80	80	80	F0	F0	F0	FF	FF	FF	00	00	.ÿ...€€€ðððÿÿÿ..
5	FF	00	FF	FF	00	FF	00	FF	00	00	FF	FF	00	FF	FF	00	ÿ.ÿÿ.ÿ.ÿ.ÿÿ.ÿÿ.
6	FF	FF	00	FF	FF	00											ÿÿ.ÿÿ.

Figura 3. Interpretación del contenido del archivo *img4x4-rgb.bmp*

La documentación del formato BMP informa de que el archivo tiene otras partes además del mapa de bits propiamente dicho. Como la mayoría de los formatos de archivo, éste comienza con un bloque de **encabezado** de archivo. En un BMP se trata de un bloque de tamaño fijo: 14 bytes, que es el que en la Figura 3 está señalado en blanco. Uno de los propósitos de este encabezado es identificar este archivo como un BMP. Para ello, se utiliza una estrategia muy habitual: los primeros bytes (dos en este caso) contienen una **firma** o "**número mágico**". La firma para un BMP es un par de caracteres "BM" en código ASCII. Puedes identificarlos fácilmente en la parte derecha del listado (¡nos equivocamos al pensar que esta parte no nos iba a ser de utilidad!).

El encabezado tiene más funciones aparte de identificar el tipo de archivo. Frecuentemente en muchos formatos de archivo se puede distinguir una estructura contenedora de un contenido. El papel del **contenedor** puede asimilarse al de un sobre y el contenido sería el documento que éste encierra, que podría ser de diferentes tipos (una carta, una factura, etc.). El encabezado de un archivo suele ser parte de este contenedor e incorpora básicamente información para identificar y localizar en su interior el documento que contiene, que en ocasiones puede ser de tipos variados.

En el caso de un archivo BMP el contenido siempre es el mismo: una imagen *bitmap* de un tipo conocido como **DIB**. Pero un DIB tiene al menos dos partes: la **cabecera del DIB** (señalado en naranja en la Figura 3) y el **mapa de bits** del que ya habíamos hablado. El encabezado del archivo permite localizar estas partes: la cabecera del DIB no tiene problema, va a continuación del encabezado del archivo. Pero para determinar dónde comienza el mapa de bits, los 4 bytes 0Ah, 0Bh, 0Ch y 0Dh del encabezado (recuadrados en la figura) deben especificar la posición de inicio de éste dentro del archivo (00000036h en este caso). Finalmente, los bytes 02h, 03h, 04h y 05h informan del tamaño total del archivo BMP: 00000066h= 102 bytes. Cuando el archivo se lee y la imagen que contiene se carga en la memoria del ordenador, el encabezado del archivo se descarta.

La cabecera de la imagen DIB es una tabla que contiene toda la información necesaria para la correcta interpretación del mapa de bits. En la tabla que aparece a continuación seleccionamos los contenidos más relevantes del encabezado DIB de nuestro archivo (corresponden a los valores recuadrados en la zona naranja en la Figura 3):

Información	Posición (bytes)	Contenido	Interpretación
Tamaño tabla	0Eh-0Fh-10h-11h	00000028h	40 bytes
Ancho bitmap	12h-13h-14h-15h	00000004h	4 píxeles
Alto bitmap	16h-17h-18h-19h	00000004h	4 píxeles
Bits por píxel	1Ch	18h	24 bpp
Compresión	1Eh	00h	Sin compresión
Resolución horizontal	26h-27h-28h-29h	00000B12h	2836 pix/m
Resolución vertical	2Ah-2Bh-2Ch-2Dh	00000B12h	2836 pix/m
Colores paleta	2Eh	00h	Sin paleta

Figura 4. Principales informaciones contenidas en la cabecera DIB del archivo *img4x4-rgb.bmp*

La cabecera del DIB informa de que la imagen contiene un bitmap de 4 x 4 píxeles, donde el color de cada píxel se codifica mediante 24 bits (3 bytes), de forma directa (sin paleta de color) y no se ha aplicado ningún tipo de compresión. Se registra además la **resolución** vertical y horizontal de la imagen, lo que posibilita la determinación de su tamaño original.

CUESTIÓN 2. La resolución de la imagen está anotada de una forma poco habitual (en píxeles por metro). Determina cuál es la resolución de esta en imagen expresada en píxeles por pulgada (ppp). Considera 1 pulgada = 2,54 cm.

5 Más informaciones que puede contener un archivo de imagen

Por simplicidad, para el ejemplo anterior hemos escogido un formato relativamente sencillo, con una versión básica, pero podemos encontrar formatos más sofisticados, con mayores posibilidades. Los propios archivos BMP han ido ampliándose sucesivamente, con posibilidad de incluir imágenes con cabeceras más complejas, que permiten aportar nuevas informaciones a la imagen:

Uso de colores transparentes.

Se completa la descripción RGB de los colores con un cuarto canal de transparencia o **canal alfa**, que permite determinar el grado de transparencia de cada píxel.

Normalización del color

Se incluye información para permitir una descripción de los colores más fiel e independiente del dispositivo de visualización, de forma que se asegure que en cualquier dispositivo los colores se vean de idéntica manera. Para ello, en las versiones más simples de formato, pueden incluirse los valores de **gamma** para cada canal de color, (similares al ajuste de gamma, relacionado con el contraste, que puede hacerse en muchos monitores,). En el caso más sofisticado se incorpora un **perfil ICC de color**, un mecanismo que permitirá la traducción de los valores RGB almacenados a un espacio normalizado e independiente del dispositivo.

Otros formatos van más allá, añadiendo posibilidades adicionales:

Metadatos

Se trata de almacenar informaciones complementarias a la propia imagen. Por ejemplo en un archivo PNG existen campos para especificar informaciones tales como: **título, nombre del autor, descripción, copyright, hora de creación, software que se usó para su creación**, etc.

Otros formatos como JPEG permiten incluir metadatos en un formato normalizado denominado **EXIF** que muchas cámaras digitales utilizan. De esta forma es posible anotar en la imagen, además de sus características, los ajustes de cámara con los que se tomó la fotografía, como por ejemplo: **velocidad de disparo, diafragma, distancia focal del objetivo, si se usó flash o no, etc.** También incorpora otras informaciones como el **modelo de cámara** que se usó o la **localización GPS** de la toma si estaba disponible.

6 Archivos de imagen con paleta de color

Una estrategia frecuentemente utilizada para mantener reducido el tamaño de un archivo de imagen es codificar el color mediante una **paleta de colores**. El mapa de bits no contiene en este caso de forma directa los valores de color del píxel, sino que especifica un número (*índice*) de color. Es habitual que se manejen paletas de 256 colores, por lo que este número tomaría valores entre 0 y 255 y puede codificarse así mediante un único byte. Con este número o índice se acude a la paleta de colores, una estructura en forma de tabla que acompaña al mapa de bits y que especifica el color RGB que se asocia a cada uno de los 256 colores numerados. En ocasiones los programas de edición de imágenes se refieren a este método como **modo de color indexado**.

Como ejemplo, la Figura 5 representa el contenido de un archivo BMP para la misma imagen anterior, pero ahora codificada mediante una paleta de 256 colores. Esto lo señalan los bytes 1Ch y 2Eh de la cabecera del DIB, que indican que se están usando 8 bits por píxel en el mapa de bits (valor 08h) y que existe una paleta que contiene 10 colores (valor 0Ah). La paleta de colores en un BMP aparece a continuación de la cabecera del DIB (bloque en verde en la figura). Cada entrada contiene 4 bytes, los 3 primeros describen el color RGB, el cuarto, siempre a cero, no tiene uso. Los primera entrada de la paleta corresponde al color número 0, la siguiente al número 1 y así sucesivamente. Las 10 entradas de 4 bytes que forman la paleta (observa que nuestra imagen contenía sólo 10 colores diferentes) están recuadradas en la figura.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	42	4D	68	00	00	00	00	00	00	00	5E	00	00	00	28	00	Bmp..... ^... (.
1	00	00	04	00	00	00	04	00	00	00	01	00	08	00	00	00
2	00	00	12	00	00	00	12	0B	00	00	12	0B	00	00	0A	00
3	00	00	0A	00	00	00	FF	00	FF	00	FF	00	00	00	FF	FF ÿ.ÿ.ÿ...ÿÿ
4	00	00	00	FF	00	00	00	FF	FF	00	00	00	FF	00	FF	FF	...ÿ...ÿÿ...ÿ.ÿÿ
5	FF	00	F0	F0	F0	00	80	80	80	00	00	00	00	00	00	00	ÿ. ððð. €€€.
6	00	00	09	08	07	06	05	04	03	01	02	02	02	02	02	02

Figura 5. Interpretación del contenido del archivo *img4x4-paleta.bmp*

La siguiente figura, en la parte derecha, representa de manera más gráfica mediante una tabla cuál es el contenido de esta paleta (sólo sus 6 primeras entradas). Se detalla para cada índice de color (primera columna) la descripción RGB y el color que le corresponde (segunda columna).

2	2	2	2	
5	4	3	1	
0	0	0	0	

0	FF00FFh
1	0000FFh
2	00FFFFh
3	00FF00h
4	FFFF00h
5	FF0000h
...	...

Figura 6. Imagen del archivo *img4x4-paleta*. Izquierda: la imagen, con anotación de los índices de color usados para cada píxel. Derecha: paleta de color en forma de tabla.

A la derecha se ha incluido la imagen original, sobre la que se han anotado algunos de los índices de color asignados a cada píxel. Estos índices se han obtenido del mapa de bits (bloque azul) de la Figura 5. Ahora los valores de los

píxeles en el mapa de bits contienen referencias a entradas de la paleta de color en forma de índices codificados con un solo byte por píxel. Observa por ejemplo que los cuatro primeros píxeles del mapa de bits (4 primeros bytes del bloque azul de la Figura 5) hacen referencia al color de índice 0. La paleta nos dice que el color asignado al índice 0 es el magenta, que es, efectivamente, el color original de los 4 píxeles que forman la última fila de nuestra imagen.

Hay otros formatos de amplio uso que también pueden incluir imágenes con paleta de color, como PNG, TIFF o GIF. En este último, de hecho, es la única opción disponible.

CUESTIÓN 3. A partir de la interpretación del mapa de bits de la Figura 5 determina los índices de color que faltan en la imagen de la Figura 6, así como las entradas de la paleta por completar, con el color que les corresponde y su valor *RGB*.

7 Solución a las cuestiones

CUESTIÓN 1

El tamaño en bytes de un mapa de bits *RGB* que use 3 bytes por píxel, se obtiene fácilmente así: Tamaño (*bytes*)= número de píxeles x 3. Así que el de la imagen de 8 Mpíxeles ocupará $8M \times 3 = 24M$ bytes.

CUESTIÓN 2

Habrá que multiplicar el valor en píx/m por los metros que mide una pulgada: $2836 \times 0,0254 = 72$ ppp. Se trata de un valor de resolución bastante habitual, equivalente al que se considera estándar para la pantalla de un monitor.

CUESTIÓN 3

2	2	2	2
5	4	3	1
9	8	7	6
0	0	0	0

6	FFFFFFh
7	F0F0F0h
8	808080h
9	000000h

8 Cierre

A lo largo de este objeto de aprendizaje hemos revisado con detalle cuál es el contenido de un archivo de imagen en mapa de bits. Se ha considerado un ejemplo práctico, referido a una imagen que se guarda en formato BMP, primero en modo *RGB* y después en modo indexado para obtener una reducción en el tamaño del archivo. Hemos repasado además qué otras informaciones adicionales suele contener un archivo de imagen.

9 Bibliografía

[1] Digicamsoft, "Bitmap File Structure", Disponible en: <http://www.digicamsoft.com/bmp/bmp.html>

[2] Martin Reddy, "The Graphics File Format Page", Disponible en: <http://www.martinreddy.net/gfx/2d-hi.html>