

Imágenes digitales: introducción sobre el cómo, por qué y para qué...

Las primeras fotografías se obtenían marcando un material fotosensible con la poca luz que pasaba a través de un orificio en una caja oscura, ubicada frente a lo que se quería fotografiar. Hoy en día el mecanismo se complejizó mucho más, pero con la consecuente incorporación de mejoras en nitidez, resolución, velocidad de captura, color, etc.

¿Cómo se guarda una imagen en una computadora¹?

Primero hay que distinguir por lo menos 3 aspectos de una fotografía:

- 1) Resolución
- 2) Color
- 3) Profundidad de color (*bit depth*)

1. La primera se refiere a cuántos píxeles tiene la imagen, es decir, cuántas “unidades homogéneas de color”. Típicamente, una imagen tiene entre 0.5 y 10 megapíxeles, dependiendo de la capacidad de la cámara. Para mostrar estas imágenes de manera fiel, hay que tener una pantalla con la misma resolución. Se puede mostrar una fotografía tomada de 8.3 Mpx (4k UltraHighDefinition) en un monitor de 1024x768 (0.8 Mpx), pero indefectiblemente se pierde resolución en el proceso: la placa de video tiene que elegir una forma de combinar los píxeles de la imagen digital para mostrarlos en la pantalla de relativamente baja resolución.

2. La segunda se refiere esencialmente a si está en escala de grises o no. Esto determinará cuánta memoria del disco rígido ocupa cada píxel de la imagen. Si es en escala de grises (coloquialmente conocido como “blanco y negro”) basta con que la información del píxel sea el nivel de gris. Si hay 256 niveles de gris, puedo representar a cada píxel con un valor entre 0 (negro) y 255 (blanco). En el caso de imágenes a color, es fundamental el hecho de que cualquier color se puede representar con la unión de tres colores fundamentales, algo así como el Teorema Fundamental de Imágenes Digitales. Estos son la famosa trina de Telefé, RGB por Rojo, Verde y Azul. Nótese entonces que ahora un píxel debe contener tres números, que representan la cantidad de cada uno de los colores fundamentales².

3. Por último se encuentra la profundidad de color, la cual es fácil de ilustrar en el caso de escala de grises, respondiendo a la pregunta: ¿cuántos números puedo representar con n bits? Bueno, dado que cada uno puede tomar un valor que es o bien 0 o bien 1 (sistema binario), hay $2 \times 2 \times \dots \times 2$ (n veces) = 2^n números (¡de la misma forma que con el sistema decimal puedo representar 10^n números con n cifras!). La cantidad de bits que son asignados a un píxel es justamente la profundidad de color y determina cuántos valores intermedios hay entre blanco y negro. Para una imagen de 8-bit (expresión que significa cuántos bits son asignados a cada píxel), habrá $2^8=256$ tonos de grises posibles. En el caso de imágenes a color, la profundidad de color es típicamente 24-bit (=8 bit para cada color).

La información de cada imagen se ordena en matrices, donde cada elemento representa un píxel. Una imagen en blanco y negro tendrá una sola matriz y una imagen a color tres matrices. Con estos conceptos en mente, ya sabemos que en principio necesitamos, en una imagen de N -bit y M Mpx, $M \times N / 8$ MB de memoria en el disco.

¹ Esto implica necesariamente que la imagen es digital.

² Ojo. Hay dos formas de unir colores: una aditiva, que es la mencionada, y otra sustractiva, que consiste en combinar Cian, Magenta, Amarillo y Negro CMYK (este es el método de las impresoras).

Algunos conceptos extra

Alfa o transparencia: a veces las imágenes a color agregan una matriz que contiene la información de la transparencia de los píxeles, representada con valores de 0 a 1, transparente a opaco respectivamente. Esto se usa cuando se quieren superponer imágenes y es una propiedad que pueden guardar los archivos PNG. En este caso las imágenes a color pasan a ser de 32-bit (4 x 8-bit).

Próximamente... pero que pueden ir buscando por su cuenta

Intensidad lumínica de un píxel (cómo pasar de color a blanco y negro)

Hue (longitud de onda)

Saturación (algo así como contenido de gris)

Aliasing (imposibilidad de representar patrones de líneas con baja definición)

Compresión (algoritmo para que las imágenes pesen menos)

Formatos de imágenes (estrategias para optimizar la relación calidad/peso)

Contenidos:

Para los entusiastas de Python, acá un tutorial sobre el manejo de imágenes, donde van a poder aplicar algunos de los conceptos vistos hasta ahora:

<https://matplotlib.org/tutorials/introductory/images.html#sphx-glr-tutorials-introductory-images-py>

Y el infaltable ImageJ para analizar y editar imágenes. Es un software portátil liviano y muy completo:

<https://imagej.nih.gov/ij/download.html>