Home / Wavelets / Digital Image Basics

Image Compression:
How Math Led to the JPEG2000 Standard

Digital Image Basics

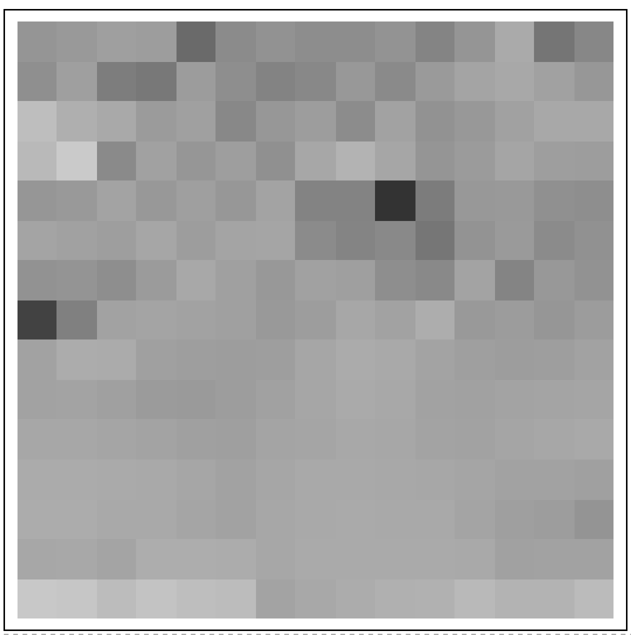
Grayscale Images

The image at right is a thumbnail version of a *digital grayscale image*. The image is a rectangular tiling of fundamental elements called *pixels*. A pixel (short for **pic**ture **el**ement) is a small block that represents the amount of gray intensity to be displayed for that particular portion of the image. For most images, pixel values are integers that range from 0 (black) to 255 (white). The 256 possible gray intensity values are shown below.

Thumbnail of a digital grayscale image. Fullsize version

The range of intensity values from 0 (black) to 255 (white).

Even if you view the full-size image, it is difficult to see the individual pixel intensities. This is the advantage of *high-resolution* images - more dots (or pixels) per inch (dpi) produces a finer image. Web applications often require images to have resolution 200dpi will printed matter such as books require 300dpi. The full-size image above is artificially enlarged - it is actually 2.016" x 3.024" or 508dpi. In order to get a better idea of pixel intensity values, we have taken the 15 x 15 pixel block that represents the bottom left-hand corner of the image and enlarged it. The images below show the enlarged block as well as their intensity values.



In This Section

- Wavelets and JPEG2000 Home
- **Digital Image Basics**
- Compression in a Nutshell
- Basic JPEG
  - The Discrete Cosine Tranformation
  - Quantization
  - Color Images
- Wavelet Transformations
  - The Haar Wavelet Transformation
  - General Wavelet Transformations
  - Wavelet Transformations used by JPEG2000
- JPEG2000
  - Features/Enhancements
  - Quantization
- References and Links

The lower left 15 x 15 pixel portion of the image.

150	154	160	157	106	140	147	142	141	147	132	150	171	117	136
144	159	125	121	157	143	132	136	153	138	155	164	169	162	152
190	175	169	155	161	136	152	158	141	162	147	153	161	168	169
185	203	139	161	151	159	145	167	179	167	150	155	165	159	158
151	153	163	152	160	152	164	131	131	51	124	152	154	145	143
164	162	158	167	157	164	166	139	132	138	119	148	154	139	146
147	148	143	155	169	160	152	161	159	143	138	163	132	152	146
66	129	163	165	163	161	154	157	167	162	174	153	156	151	156
162	173	172	161	158	158	159	167	171	169	164	159	158	159	162
163	164	161	155	155	158	161	167	171	168	162	162	163	164	166
167	167	165	163	160	160	164	166	169	168	164	163	165	167	170
172	171	170	170	166	163	166	170	169	168	167	165	163	163	160
173	172	170	169	166	163	167	169	170	170	170	165	160	157	148
167	168	165	173	173	172	167	170	170	171	171	169	162	163	162
200	198	189	196	191	188	163	168	172	177	177	186	180	180	188

Pixel intensity values of the lower left 15 x 15 pixel portion of the image.

The fundamental unit on a computer is a *bit*. A bit (binary unit) takes either the value 0 or 1. The *byte* (the fundamental unit of storage on a PC) is composed of 8 bits. Since each bit takes on one of two values and 8 bits make a byte, we can use the multiplication principle to realize that there are $2^8 = 256$ possible bytes. We represent these bytes in base 2. For example, the intensity 125 can be written as

$$\begin{aligned} 125 &= 64 + 32 + 16 + 8 + 4 + 1 \\ &= \mathbf{0} \times 128 + \mathbf{1} \times 64 + \mathbf{1} \times 32 + \mathbf{1} \times 16 + \mathbf{1} \times 8 + \mathbf{1} \times 4 + \mathbf{0} \times 2 + \mathbf{1} \times 1 \\ &= \mathbf{0} \times 2^7 + \mathbf{1} \times 2^6 + \mathbf{1} \times 2^5 + \mathbf{1} \times 2^4 + \mathbf{1} \times 2^3 + \mathbf{1} \times 2^2 + \mathbf{0} \times 2^1 + \mathbf{1} \times 2^0 \\ &= 01111101_2 \end{aligned}$$

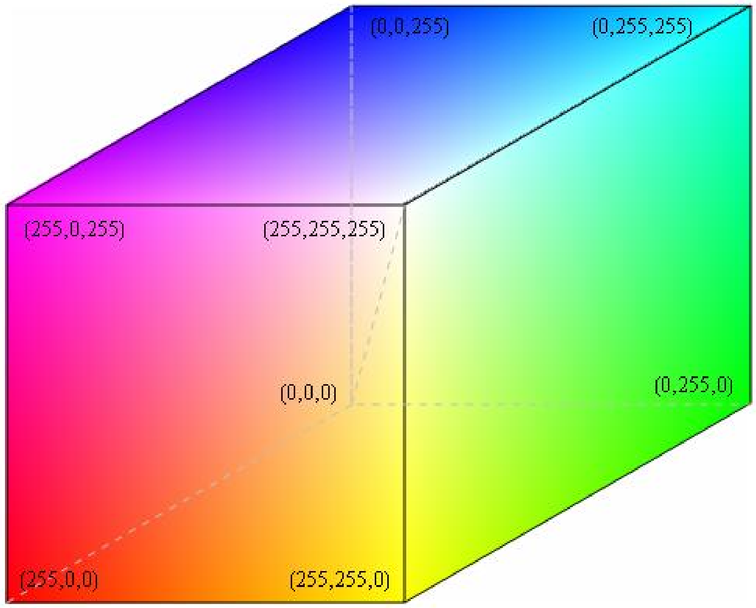
The American Standard Code for Information Exchange (ASCII) has assigned each of the 256 possible bytes to each keyboard character. Click here to see the ASCII chart for the first 128 bytes and here for the (extended) last 128 bytes. And yes, there are 256 characters on a standard keyboard! If you are using a PC, open Notepad, enable NUMLOCK, hold down the ALT key and type 125 on the numeric keypad - you will see the ASCII character for the right brace **}** appear on screen.

If an image of size M x N pixels is stored in raw format on a web server or digital camera, then aside from some header information, the file consists of M x N x 8 bits (zeros and ones) where the rows of the image are concatenated to form one long *bit stream*. For our example image, the bit stream has length 768 x 512 x 8 = 3,145,728 bits!

Color Images

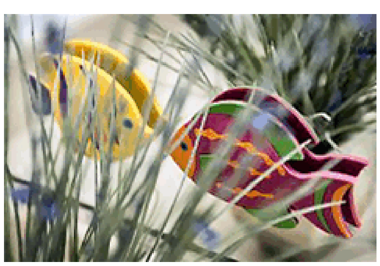
Color images require more storage space than grayscale images. Pixels in grayscale images need just one byte to indicate the intensity of gray needed to render the pixel on screen. It turns out that any color can be built using the correct combination of red, green, and blue. Thus, pixels in color images are represented by three values (r,g,b). The values indicate the intensity of red, green, and blue, respectively, needed to render the pixel on screen. The range of intensities is exactly the same as grayscale images - 0 means none of the color appears in the pixel and 255 indicates the highest level of the color is evident in the pixel. For example, the triple (128, 0, 128) would represent a medium purple while (255, 215, 0) represents gold. Click here for examples of other color triples.

The set of all triples form the *RGB Colorspace*. The space is shown in more detail in the two images that follow.

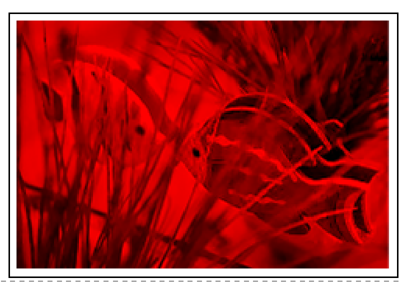


The RGB colorspace cube. Two-dimensional version

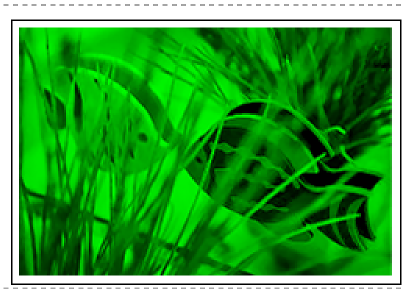
We can actually look at each of the red, green, and blue channels separately. Below are thumbnails of each. The dimensions of the original image are 768 x 512 pixels. Since each pixel requires 3 bytes of information, we can store the image to disk in raw format using 768 x 512 x 3 = 1,179,648 bytes. The bit stream would have length 1,179,648 x 8 = 9,437,184.

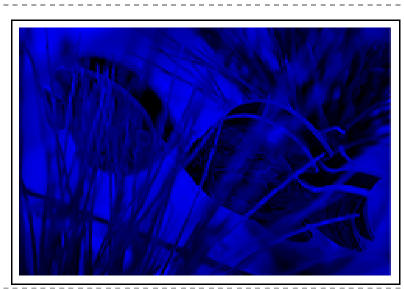


Thumbnail of a digital image. Fullsize version



Red channel. Fullsize version





Green channel. Fullsize version

Blue channel. Fullsize version

Color Space Conversion

Researchers have learned that for applications such as image compression, the RGB color space is not optimal. It turns out the human brain is more attuned to small changes in terms of *luminance* and *chrominance*. A luminance channel carries information regarding the brightness of a pixel. Chrominance is the difference between a color and a reference channel at the same brightness. Since there are three channels in an RGB colorspace, we will use one luminance and two chrominance channels to form a new space for image compression of color images. The most common of these spaces and the one used by JPEG2000 is the *YCbCr* space. The Y channel is luminance while Cb and Cr are chrominance channels.

The first step to convert a red, green, and blue triple (r,g,b) to YCbCr space is to divide each intensity by 255 so that the resulting triple has values in the interval [0,1]. Let's define $(r',g',b') = (r/255,g/255,b/255)$.

We obtain the luminance value y using the formula

$$y = .299r' + .587g' + .114b'$$

Note that if (r,g,b) = (0,0,0) (black), then y = 0. If (r,g,b) = (255,255,255) (white), then y = .299 + .587 + .114 = 1. We say the above formula for y is a *convex combination* of r', g', and b' since the multipliers .299, .587, and .114 are nonnegative and sum to one. Actually, this is exactly the formula suggested by the National Television System Committee (NTSC) for converting color feeds to black and white televisions sets.

For the chrominance channels, we measure the difference between two color channels and the reference channel y. In the YCbCr space, the colors we use are blue and red. The Cb channel is defined as

$$Cb = (b'-y)/1.772$$

and the Cr channel is given by

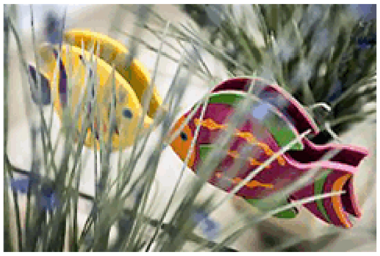
$$Cr = (r'-y)/1.402$$

The 1.772 and 1.402 appear in the denominators of Cb and Cr, respectively so that the resulting values lie in the interval [-1/2, 1/2].

For display purposes, the values y, Cb, and Cr are scaled by the formulas

$$Y = 219y + 16$$
$$C_b = 224Cb+128$$
$$C_r = 224Cr+128$$

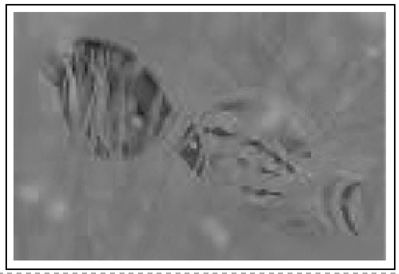
and then rounded to the nearest integer. The images below show the Y, C_b, and C_r channels for a color image.



Thumbnail of a digital image. Fullsize version

Y channel. Fullsize version

C_b channel. Fullsize version



C_r channel. Fullsize version

Images courtesy of Radka Tezaur.