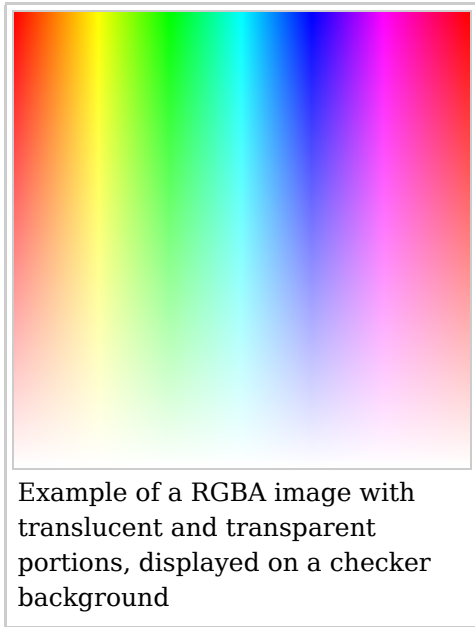# RGBA color space

From Wikipedia, the free encyclopedia

RGBA stands for **Red Green Blue Alpha**. While it is sometimes described as a color space, it is actually simply a use of the RGB color model, with extra information. The color is RGB, and may belong to any RGB color space, but an integral alpha value as invented by Catmull and Smith between 1971 and 1972 enables alpha compositing. The inventors named alpha after the Greek letter in the classic linear interpolation formula $\alpha A + (1-\alpha)B$.

The alpha channel is normally used as an opacity channel. If a pixel has a value of 0% in its alpha channel, it is fully transparent (and, thus, invisible), whereas a value of 100% in the alpha channel gives a fully opaque pixel (traditional digital images). Values between 0% and 100% make it possible for pixels to show through a background like a glass (translucency), an effect not possible with simple binary (transparent or opaque) transparency. It allows easy image compositing.

Example of a RGBA image with translucent and transparent portions, displayed on a checker background

PNG is an image format that uses RGBA.

Contents

ARGB

In computer graphics, pixels encoding the RGBA Color Space information must be stored in computer memory (or in files on disk), in well defined formats. In the most common format the intensity of each channel sample is defined by 8 bits, and are arranged in memory in such manner that a single 32-bit unsigned integer has the Alpha sample in the highest 8 bits, followed by the Red sample, Green sample and the Blue sample in the lowest 8 bits. This is often called "ARGB":

Sample Length:	8	8	8	8
Channel Membership:	Alpha	Red	Green	Blue
Bit Number:	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0

ARGB values are typically expressed using 8 hexadecimal digits, with each pair of the hexadecimal digits representing the values of the Alpha, Red, Green and Blue channel, respectively. For example `80FFFF00` represents 50.2% opaque (non-premultiplied) yellow. The `80` hex value, which is 128 in decimal, represents a 50.2% alpha value because 128 is approximately 50.2% of the maximum value of 255 (FF hex); to continue to decipher the `80FFFF00` value, the first `FF` represents the maximum value red can have; the second `FF` is like the previous but for green; the final `00` represents the minimum value blue can have (effectively – no blue). Consequently red + green yields yellow. In cases where the alpha is not used this can be shortened to 6 digits `RRGGBB`, this is why it was chosen to put the alpha in the top bits. Depending on the context a `0x` or a number sign (#)[1] is put before the hex digits.

A confusing aspect is that on a little-endian CPU (such as Intel or AMD processors) the byte for B is stored at the lowest address, with the bytes representing the colors are in the order B,G,R,A. On a big-endian machine the bytes are in the order A,R,G,B. This only applies if read bytewise; if read 32 bits at a time the result will be the same in both endians.

To indicate how it is to be interpreted this form of ARGB is often called ARGB32, because on both little-endian and big-endian each pixel can be read as 32 bits at a time and then interpreted as ARGB.

RGBA byte order

In some contexts, primarily OpenGL, the term "RGBA" actually means the colors are stored in memory such that R is at the lowest address, G after it, B after that, and A last. This is *not* the format described above. OpenGL describes the above format as "BGRA" on a little-endian machine and "ARGB" on a big-endian machine.

When there are more than 8 bits per channel (such as 16 bit or floating-point), it is very common that the channels are stored in R,G,B,A order, even if 8-bit data is stored as ARGB words. This can often lead to confusion when software is converted from 8 bits to higher resolution color.

Similar to the ARGB32 naming above, this encoding is also known as RGBA8888 or R8G8B8A8 because the format can be read on both little-endian and big-endian machine using four 8-bit reads giving R, G, B and A.

RGBA hexadecimal

In some software originating on big-endian machines such as Silicon Graphics, the term "RGBA" means color is specified similar to ARGB but with the alpha in the *bottom* 8 bits rather than the top. For example `808000FF` would be Red and Green:50.2%, Blue:0% and Alpha:100%, a brown. It should be noted that RGBA is relatively obscure compared to ARGB; e.g. Portable Arbitrary Map (PAM). Confusing the two can lead to serious color rendering errors.

Sample Length:	8	8	8	8
Channel Membership:	Red	Green	Blue	Alpha
Bit Number:	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0

The bytes are stored in memory on a little-endian machine in the order A,B,G,R.

References

1. ^ Microsoft MSDN XAML Color Structure reference (http://msdn.microsoft.com/en-us/library
   /system.windows.media.color(v=vs.95).aspx) (XAML/WPF/Silverlight), including `#aarrggbb` and `sc#`
   `scA,scR,scG,scB`

External links

- Alpha transparency (http://www.w3.org/TR/PNG-DataRep.html#DR.Alpha-channel) on
  W3C PNG specification
- RGBA Colors (http://www.css3.info/preview/rgba/) Preview page with implementation
  info on CSS3.info

Retrieved from "http://en.wikipedia.org/w/index.php?title=RGBA_color_space&
oldid=619540741"
Categories: Color space