

What is a Digital Image Channel and the Alpha Channel?

[The Beginner's Guide to Lighting for Film is available right now! Click this to learn more.](#)

Disclosure: Links in this post may be to our affiliates; sales through affiliate links may benefit this site. Please help support wolfcrow and buy from [B&H](#) and [Amazon](#). It won't cost you anything extra.

Think of this box as a pixel:



Let's say this pixel is part of a 2-bit image. Two measly bits. This pixel can hold only four values. (An 8-bit pixel can hold 256 values, and a 10-bit pixel can hold 1024 values).

Values? What values? *Values of what?*

Glad you asked. Three times. The correct answer is: 'Values of anything at all.' However, that's not going to be of much help, so let me break it down.

What is a Digital Image Channel?

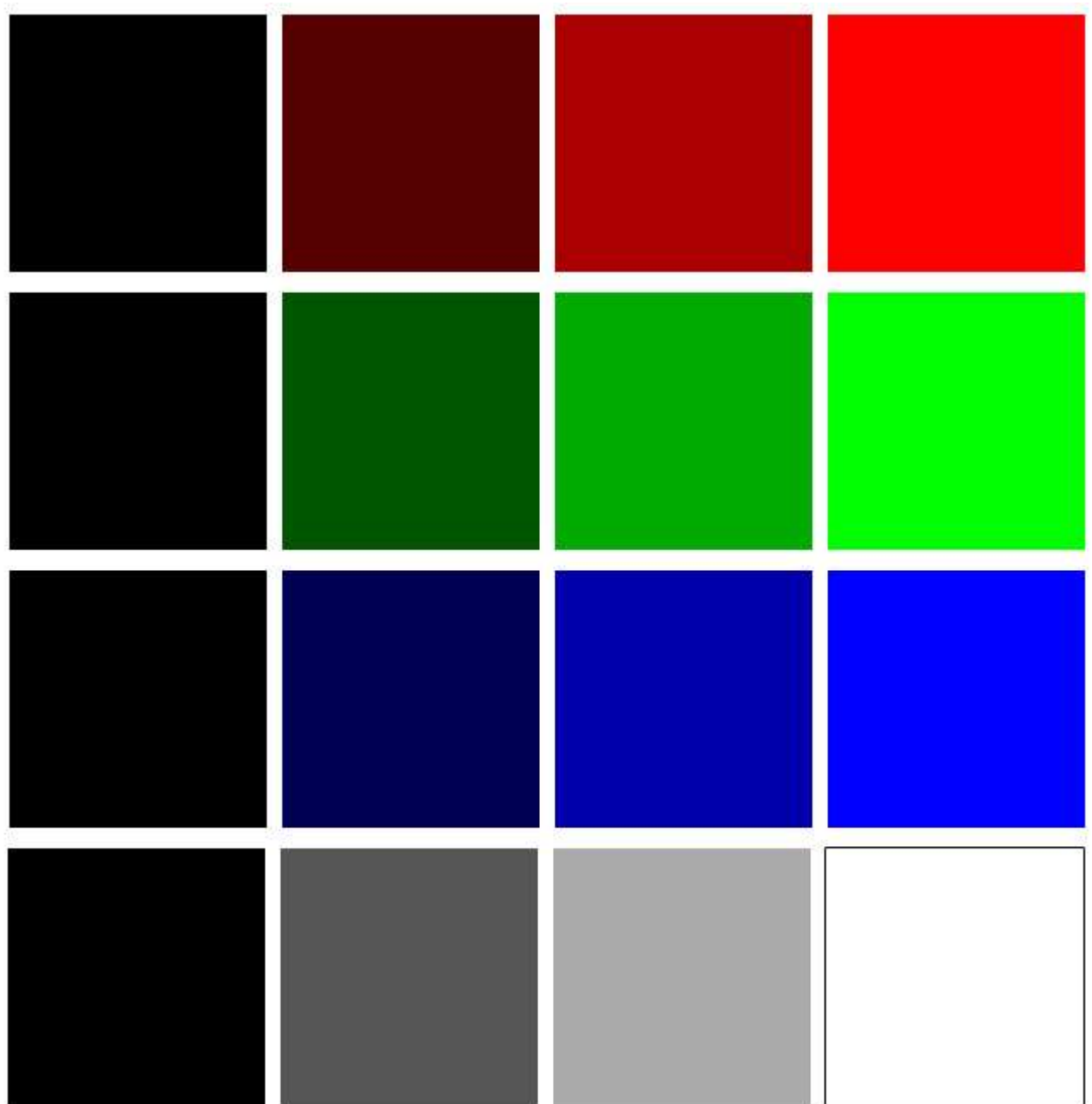
If you've read [Driving Miss Digital](#), [Color Bit Depth and Color Model](#), and [Understanding Luminance and Chrominance](#), you'll know that digital video has two fundamental ways of breaking down colors:

- The RGB model
- The Y'CbCr model

The common factor among both these models is that they come in threes – Three Colors Red, Green and Blue; or three 'components' Luma (Y'), and two Chromas (Cb and Cr).

A pixel needs all three values to make a color. In this respect, we call each color a Channel. Think of a channel as a pipe (or a channel of water). Pipe R transports red water, pipe G transports green water and pipe B transports blue water.

Let's get back to our 2-bit pixel. If I break down my 2-bit pixel into RGB, I'll have four values of each color channel. This is what it looks like (forget the fourth row for now):

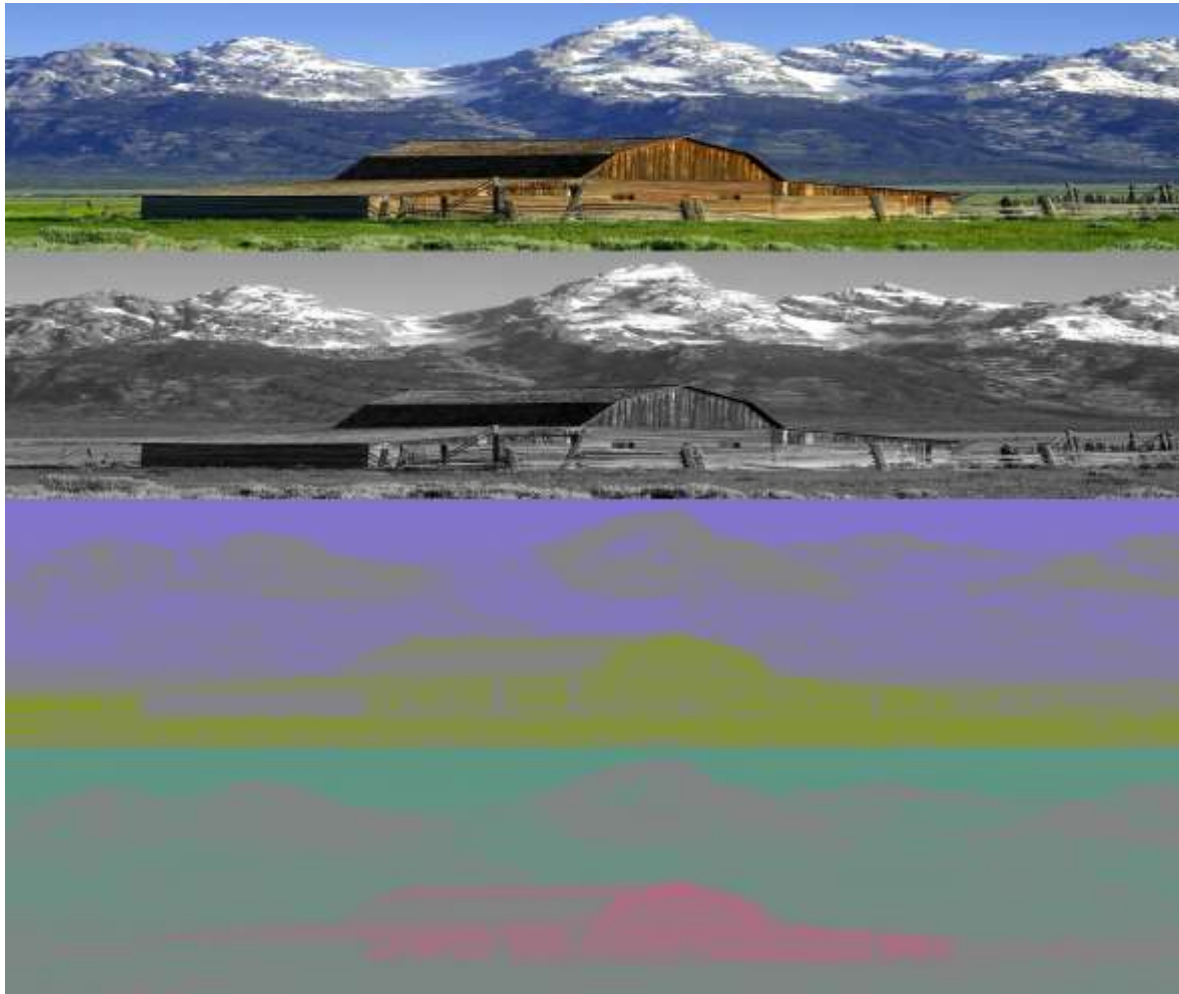


Similarly, this is what a Y'CbCr image looks like when broken down into its three components:

So, what is a channel? A channel is a set of values of any part of an image. No rules, really. If you had a group shot of your favorite Hollywood stars, you could write an algorithm to pick out your favorite star and put that information into a channel. Anything goes.

However, the most common usage of a channel is when it is divided into either RGB or Y'CbCr. In the above RGB breakdown, the red channel can have four possible values, the

green channel can have four possible values, and the blue channel can have four possible values.



Let's say you wanted to separate yellow in the image, and you knew how to do it with some crazy math. You can do it. You can have all the three colors – R, G and B, *and* a Yellow channel. A simple computer program can display the image as it is with just the RGB values, but the extra Yellow channel is there if you only want to tweak yellow. Hey, guess what? We're not limited to colors.

If you want to pass an image to your friend, along with the latest stock price of the company you are working for, you can have three channels R, G, B, and a fourth channel which contained the latest price. But, hold on –

Traditionally, it is considered bad form to pass metadata through channels. You only pass information regarding the pixel. If the data has nothing to do with the pixel, it doesn't go into the channel.

So, other than colors, what else can a channel hold that is of any use to anyone?

Alpha Channel

The fourth-most widely used channel is the Alpha Channel. Look at the fourth row in the above RGB image. It's a group of four grey boxes. But make no mistake, this is not a Luma

channel (It could be, but that's not what we're using it for here) nor a grayscale channel (Again, it could be, but that's not what we're using it for here).

The Alpha Channel is the channel that contains transparency information.

Let's make a full RGB 2-bit pixel by combining the second red value (R channel), third green value (G channel) and fourth blue value (B channel). This is what it'll look like:



Now let's make this pixel disappear. Since we are limiting ourselves to a 2-bit image, we can have only four variations of transparency. The first one is zero transparency, while the last one is fully transparent. We use the grayscale values to tell the program how much transparency we want in that pixel. This is what it'll look like:



The black gray box tells the computer to pass all the RGB values as is, while the white box blocks them all. The other two only blocks them in varying degrees, like a faucet controlling the flow of water.

Wait a minute! Doesn't white pass info, while black blocks it? You're right! I showed you the 'wrong' (It's not wrong, just unconventional) way to tell you that mathematically, you can do it either way. However, the convention is to use **black to block data and white to pass data** – the reverse of the above image. This is the result:



Did I confuse you? Don't worry. Just remember it like this: Black blocks. That's it.

Don't be fooled into assuming only four channels are necessary. There are many instances (especially in the visual effects industry) where many more channels are useful. Each area of an image is broken down for specific work, and a VFX person who specializes in one area can make changes without disturbing the other parts.

The reason why OpenEXR is used in the VFX industry is because it supports many channels, unlike JPEG, which only supports three channels RGB (no transparency).