Compression of color images

G. Tochon

LRDE/EPITA

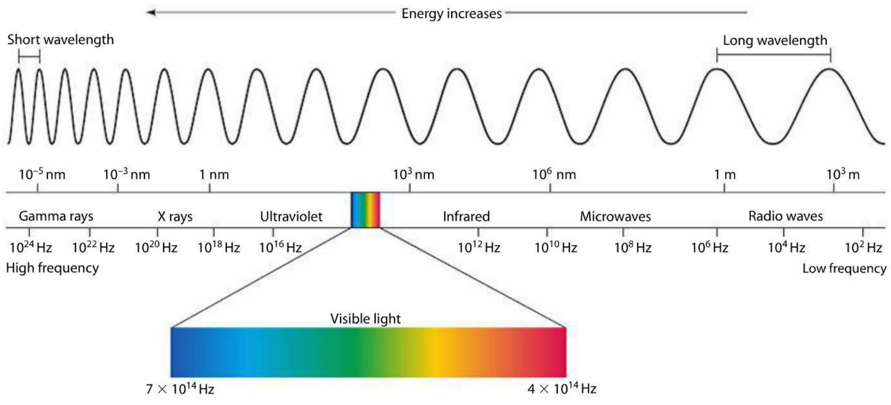
Apologies for the quite ugly looking document. The LaTeXcode was automatically generated from Matlab and I just made minor corrections by lack of time.

Contents

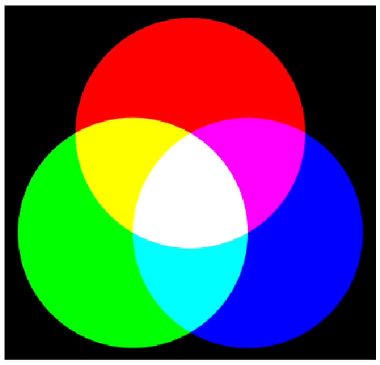
- Electromagnetic spectrum
- Structure of the (human) eye
- Structure of a digital camera sensor
- Going from grayscale to color images
- The RGB color space
- JPEG compression in the RGB color space
- JPEG compression in RGB space for all quality index values
- The main issue of the RGB color space
- Going from the RGB to the HSV color space
- Playing a little bit in the HSV color space (just for the lol)
- Going from RGB space into YUV space
- JPEG compression in the YUV space
- JPEG compression with 4:2:2 downsampling
- JPEG compression with 4:2:0 downsampling
- JPEG compression in YUV space for all quality index values

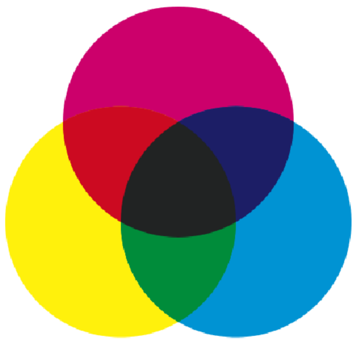
Electromagnetic spectrum

Colors and light, as our eyes perceive them, are only a tiny portion of the whole electromagnetic spectrum (wavelengths ranging from 400 nanometers - corresponding to the blue - to roughly 800 nanometers - corresponding to the red).



Any color accessible to the human eye can be decomposed as a combination of three primary colors ⇒ must define three colors that are primary to each other (one cannot be obtained as a combination of the two other). Given thos three primary colors, it is possible to recompose any color of the electromagnetic spectrum (for which the eye is sensible). In additive synthesis, the three primary colors are Red, Green and Blue. Mixing those three colors gives white. Mixing none gives black. The additive synthesis is the one used for digital screens (that emit light). In subtractive synthesis, the three primary colors are Yellow, Cyan and Magenta. In this strategy, colors are obtained by removing colors from the white light. Removing Yellow, Cyand and Magenta from white gives black. The subtractive synthesis is the one used for printing for example. The ink cartridges of any printer are composed of pigments that would block out the yellow, cyan and magenta colors of a white sheet of paper. Mixing those three primary colors allows to print any color on a white sheet of paper (which does not emit whit light, but reflect the ambiant one, contrarily to a screen that really emits light).


Additive synthesis


Subtractive synthesis

Structure of the (human) eye

Iris ⇔ shutter in a digital camera ⇒ modulates the amount of light that enter in the eye by the pupil (depends on the ambiant brightness)

Lens (cristallin) ⇔ exact same role as a classical optical lens: it refracts the light to be focused on the retina. It can change its curvature and thus impact the focal distance.

Retina ⇔ optical sensor. It is composed of photoreceptor cells of two kinds:

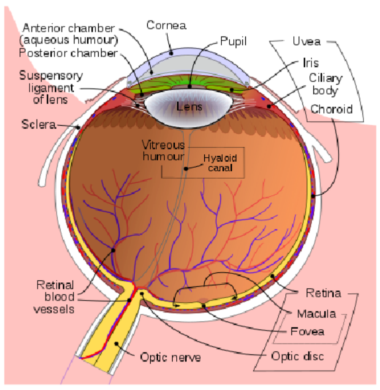
- rods (bâtonnets). Around 100 to 120 millions. They are sensible to luminosity (gray levels) and motion.
- cones (cônes). Around 5 millions. They are sensible to colors. Concentrated in the fovea area only.

Those photoreceptors convert the received light (photons) into electric impulses that go to the brain through the optical nerve.

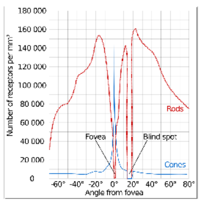
The image transmitted from the retina to the brain is:

- inverted.
- in grayscale almost everywhere, except in the fovea.
- blurred eveywhere, except in the macula area (which contains the fovea).

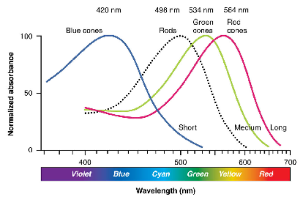
The brain does all the post-processing to recreate the original image.



Structure of the human eye



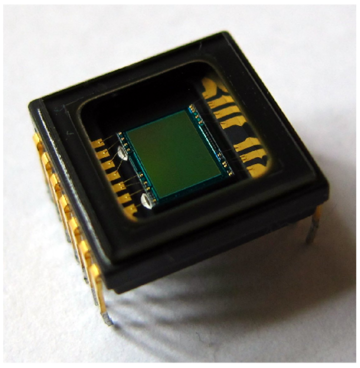
Sensitivity of eye photoreceptors



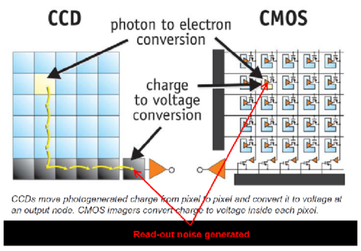
Sensitivity of eye photoreceptors

Structure of a digital camera sensor

Photodiodes in digital sensors play the same role as the eye photoreceptors: the convert the incident light into an electric signal. The conversion is done thanks to light-matter interaction (namely, the photoelectric effect). CCD and CMOS are the two big families of digital camera sensors, depending on how is done the conversion from charge to voltage.



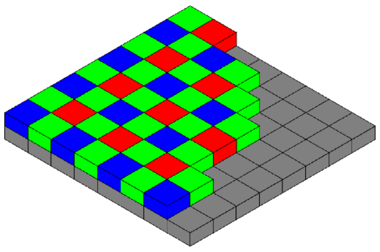
Imaging sensor

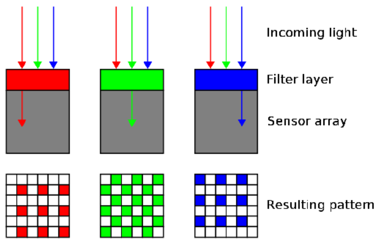

CCD
photon to electron conversion
CMOS
charge to voltage conversion
CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node. CMOS imagers convert charge to voltage inside each pixel.
Read-out noise generated

Difference between CCD and CMOS imaging technologies

Going from grayscale to color images

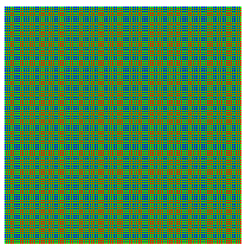
Photodiodes are sensible to the whole visible light. They sense grayscale images only, impossible to differentiate colors. ⇒ Use a Bayer filter to "re-create" the colors. One half of pixels are filtered to be green, the other half is split into red and blue. The reason is that the human eye is more sensible to the green color (more green cones than red/blue cones).

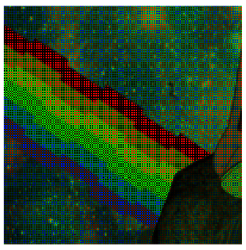

Bayer filter



Profile of Bayer filter

The image that is recorded "through" the Bayer filter is incomplete: only 1/4th of the pixels in the red and blue channels are known, and 1/2 of the green pixels as well. For each pixel site, one color is known, and the missing two must be interpolated. ⇒ in a pixel whose red value is known, the green and blue values must be interpolated from the 4 closest neighbors. The same idea goes for the green and blue values. Several interpolation strategies are possible, ranging from a simple spatial average to more complicated, optimization-absed techniques. This step is called demosaicing.


Bayer pattern

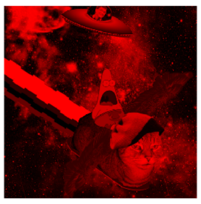

What is really sensed by the sensor

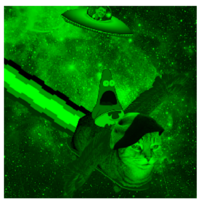
After demosaicing

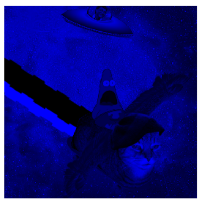
The RGB color space

A RGB color image can be decomposed as a combination of three ”base” color images, one for the Red channel (by nulling the all pixels in the Green and Blue channels), one for the Green channel and obviously one for the Blue channel.

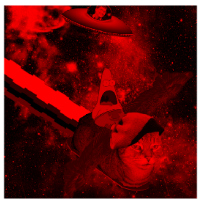
color image

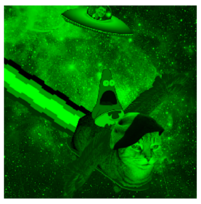

R channel


G channel

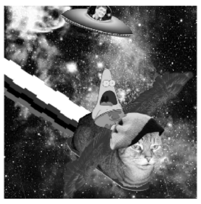

B channel

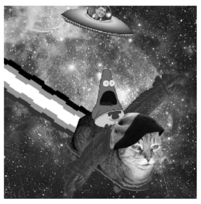
But another perspective is to view the RGB as three grayscale images stacked together. With this point of view, it becomes possible to do some color image processing by applying grayscale operations on each channel independently of the others. So it becomes possible to perform some color JPEG compression using the classical grayscale JPEG algorithm on each channel.

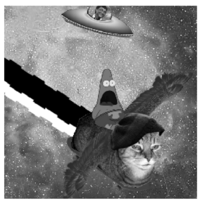

R channel


G channel

B channel


R channel (grayscale)


G channel (grayscale)


B channel (grayscale)

JPEG compression in the RGB color space

First attempt to compress color images → compressing each channel with JPEG algorithm independently of the others.

R channel

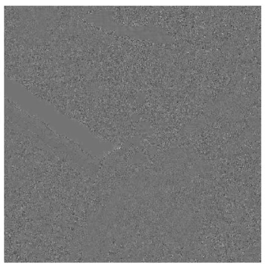
G channel

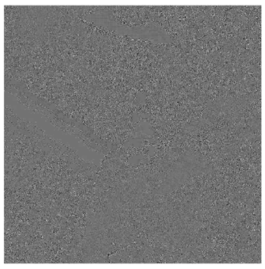
B channel

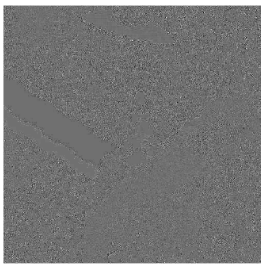
JPEG R channel
compression ratio: 2.8

JPEG G channel
compression ratio: 2.8

JPEG B channel
compression ratio: 2.87


R channel compression error
RMSE: 7.03
SNR: 24.3 dB

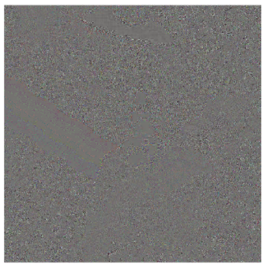

G channel compression error
RMSE: 7.05
SNR: 24.1 dB


B channel compression error
RMSE: 7.02
SNR: 25.4 dB

Compression ratios, RMSEs and SNRs are sensitively equivalent for each channel to the values obtained in the grayscale version, and this should not be a surprise. The engendered artifacts are the same (mosaic and ringing effects) as for the grayscale case. But dealing with colors yields another artifact: the false color artifact $\rightarrow$ modifying the pixel values on one channel independently of the other creates ”new” (hence, false) colors that were not present in the image before compression.

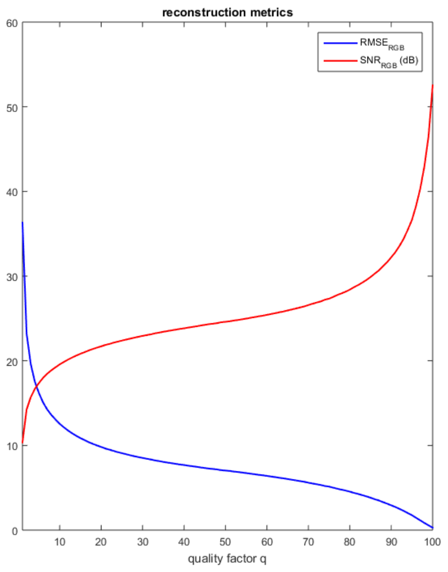
Before compression

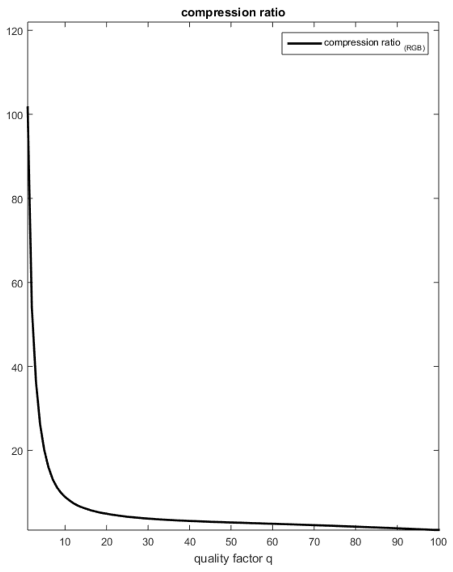
After JPEG decompression
compression ratio: 2.82


RGB compression error
mean RMSE: 7.03
mean SNR: 24.6 dB

JPEG compression in RGB space for all quality index values

The reconstruction metric curves for the RGB JPEG compression really look like those of the grayscale JPEG compression, which, again, is perfectly normal.

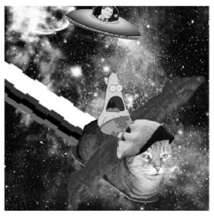




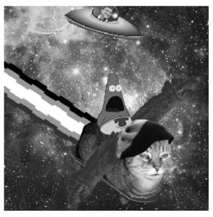
The main issue of the RGB color space

The RGB color space is actually not the most suited for color image compression. As a matter of fact, when looking at the three channels R, G and B independently, the image is perfectly recognizable in each one. It is because the colorimetric information is more or less equally spread over the three channels. Speaking in terms of data compression, this is redundancy, so there should be a way to circumvent this.

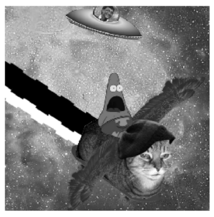
color image



R channel

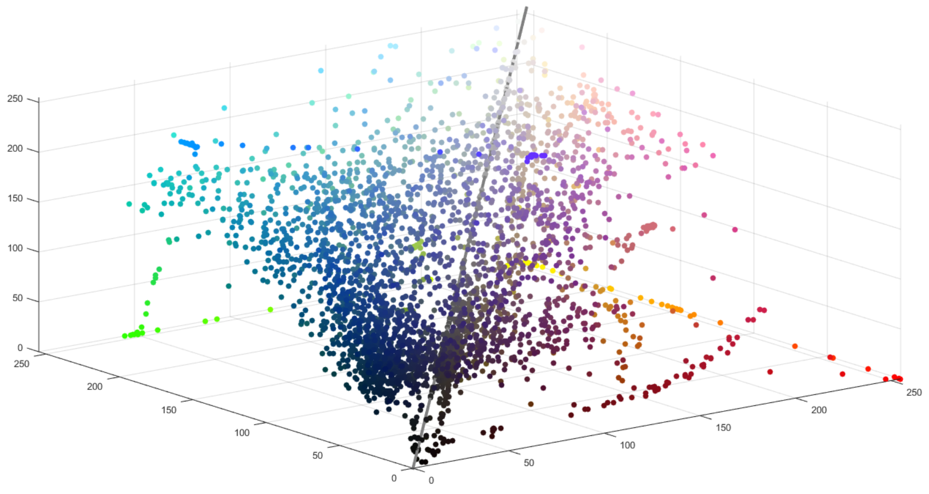


G channel



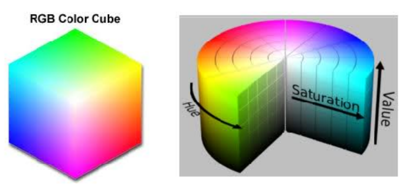
B channel

Thinking of a color pixel as a point in a 3-D space, it is possible to plot the image as a point cloud. Most of the points are distributed around the main diagonal which joins (0,0,0) and (255,255,255) (that is, all grayscale values, since a gray value is nothing more than a RGB triplet where the R, the G and the B component are equals). The main advantage of the RGB color space is that it closely reproduces what our eyes see in terms of stimuli (some red, some green and some blue).

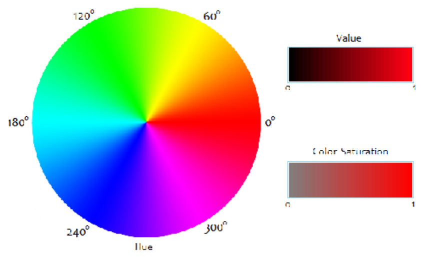


Going from the RGB to the HSV color space

Still thinking at an image as a data cloud in 3-D, we can transform the image by changing the coordinate system. A common transformation is the so called RGB to HSV transformation. The HSV, standing for Hue, Saturation Value, is a cylindrical-coordinate representation of the points lying within the RGB cube. The Hue, being the first channel of the HSV color representation, describes the color itself. It is encoded as the angle (in degrees) of the color in the chromatic circle (red = 0°, green = 120°, blue = 240°). The Saturation is the purity of the color. A saturation of 1 means perfect purity, and the lower the Saturation, the more dilated in white the color looks like. The Value is a kind of color brightness, the lower the Value, the darker the color looks like (up to the black for a Value of 1). The main advantage of the HSV color space is that it is close to the perception of color our brain has. For instance, it really makes sense to think of an image with pale colors as lacking saturation (low Saturation) or being too dark (low Value).



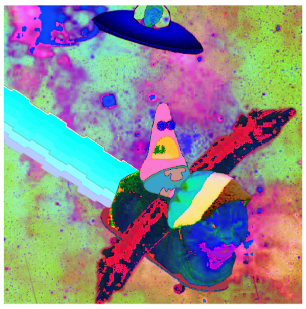
RGB and HSV color spaces



Interpretation of Hue, Saturation and Value components

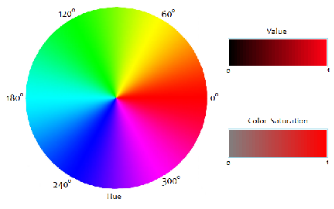
However, it makes no sense to try and visualize an HSV image just like a RGB one. Each pixel is still a triplet of values, but HSV "colors" in the RGB cube simply look weird.

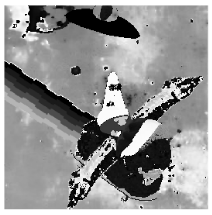
in RGB space

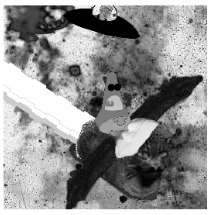

in HSV space

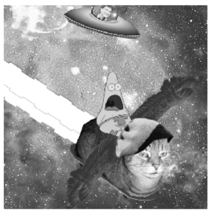
Looking at the H, S and V channels as grayscale images, we can see that the information has been kind of decorrelated (with respect to the RGB case) between the channels. But HSV is still not optimal in terms of compression, as it will further be seen.

in RGB spaces




Hue


Saturation

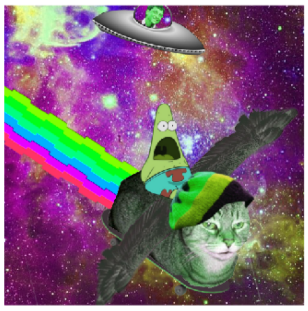

Value

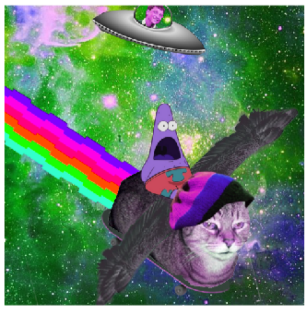
Playing a little bit in the HSV color space (just for the lol)

It is possible to do some funny stuff very easily when working in the HSV space (espacially in comparison of what would be necessary to achieve the same processing in the RGB space).

Hue is the color ⇒ shifting the Hue by a certain number of degrees amount to shift all colors in the colorimetric circle.

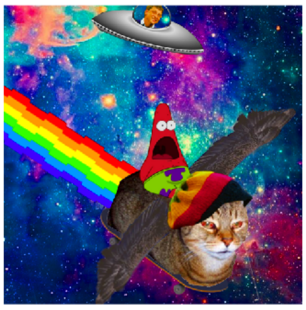
with normal Hue

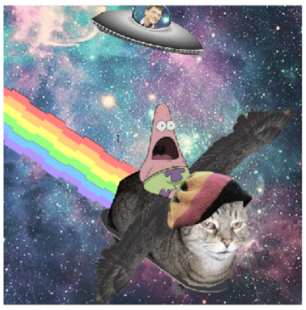

with Hue shifted from 90°


with Hue shifted from -90°

Saturation is how "punchy" is a color ⇒ doubling the Saturation will render all color really bright, while halving it will make the image look pale.

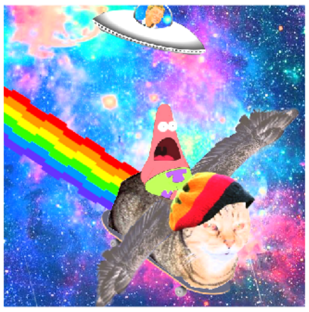
with normal saturation

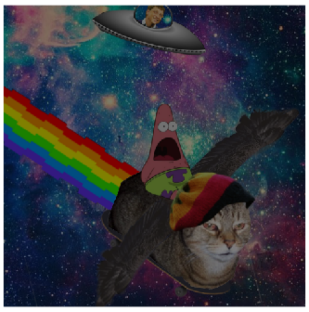

with doubled saturation


with halved saturation

The value is the color brightness, a doubling the value will make the picture look like it was overexposed, while halving it will obviously have an underexposition effect.

with normal saturation


with doubled value

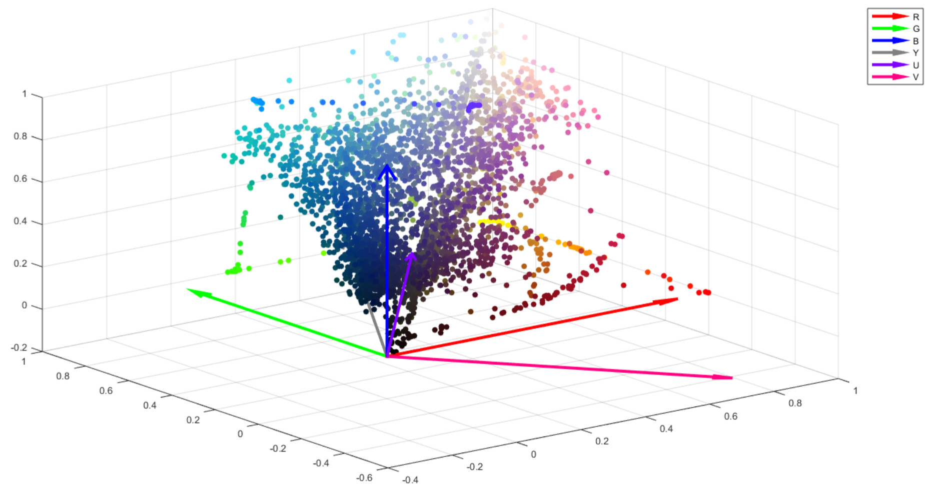

with halved value

Going from RGB space into YUV space

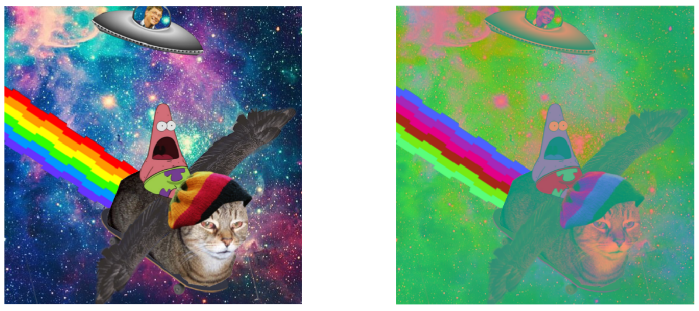
So the HSV space is nice for many color image processing, but still, we are lacking a good colorimetric space for image compression. Here comes the YUV. Orgininally, the YUV space was proposed by engineers to ease the transition between gray-color TV monitors and color TV monitors. As a matter of fact, the Y component is the Luminance of the image, and is nothing more than its grayscale version. The U and V components are called the Chrominance, and they represent the color information. The U channel is obtained as B−Y (scaled by some coefficient) and the V channel is R−Y (scaled by another coefficient). From a mathematical standpoint, it is possible to transform the RGB cube into the YUV space by a matrix multiplication (denoting a mere rotation of the R, G and B axes).

$$\mathrm{RGB2YUV} = \begin{bmatrix} 0.2990 & 0.5870 & 0.114 \\ -0.1471 & -0.2889 & 0.436 \\ 0.6150 & -0.5150 & -0.100 \end{bmatrix}$$

$$\mathrm{YUV2RGB} = (\mathrm{RGB2YUV})^{-1}$$

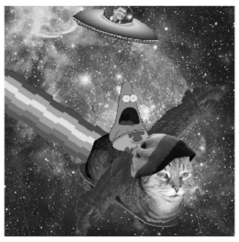


As for the HSV space, it is pointless to try and visualize a YUV image as if it was still in the RGB space (colors look as weird as in the HSV space).

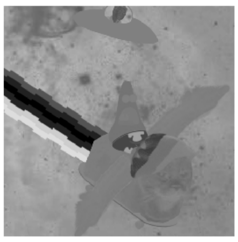


The major interest of the YUV space is that it totally decorrelates the brightness information (the Luminance) from the color information (the Chrominance). And by doing so, it turns out that the Chrominance channels do not contain ”that much” information with respect to the Luminance. From a data compression point of view, it means that it will be possible to severely compress the two Chrominance channels.

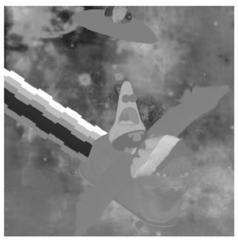
in RGB space



Luminance Y

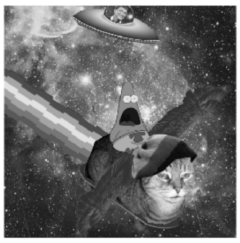


Chrominance U = (B-Y)/2.0321

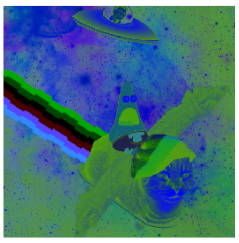


Chrominance V = (R-Y)/1.1398

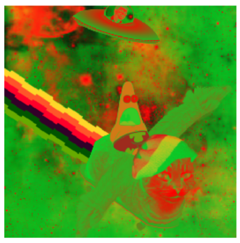
in RGB space



Luminance Y



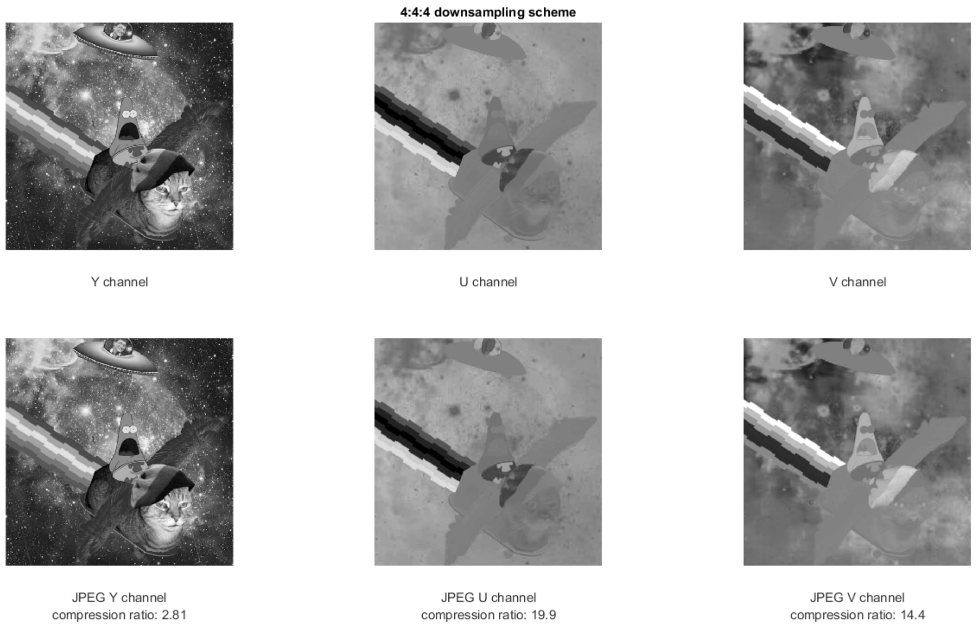
Chrominance U (in RGB color)

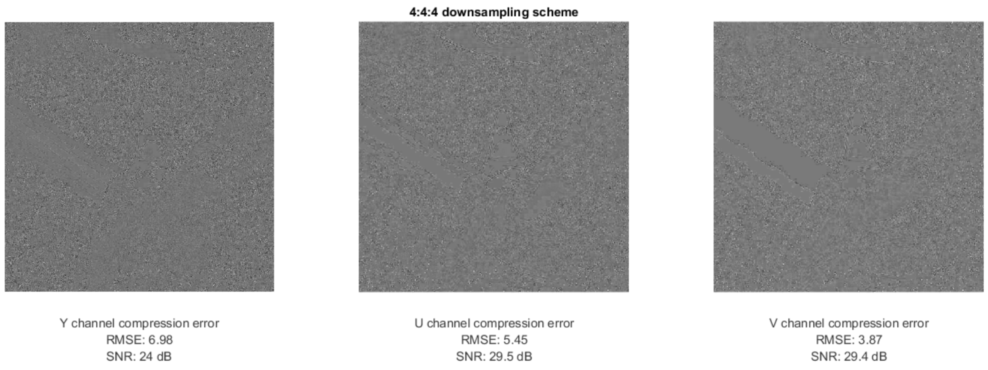


Chrominance V (in RGB color)

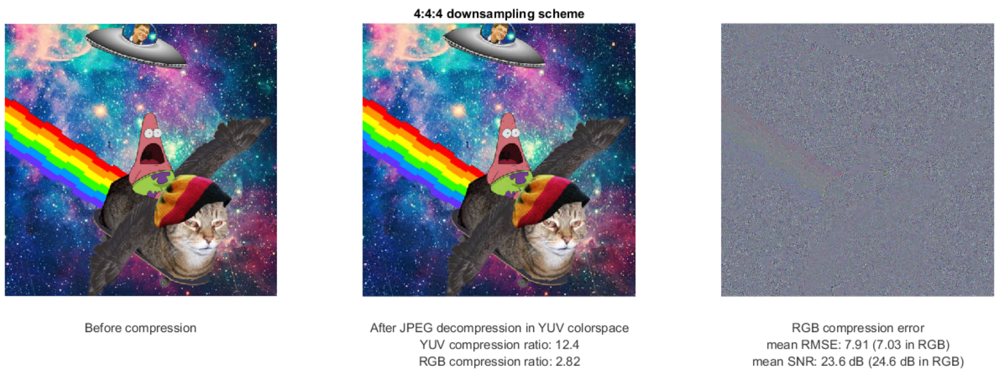
JPEG compression in the YUV space

Applying JPEG compression scheme on the Luminance channel Y is strictly equivalent to what was done up to now for grayscale images, so no wonder why the quality metrics and compression ratio are the same. For the Chrominance channels U and V however, as they appear mainly as piecewise-constant images (hence, with very few high frequencies), JPEG compression is super effective (note that the quantization matrix for the Chrominance is slightly different than for the Luminance). And indeed, the compression ratio for U and V channels is almost 10 times higher than for the Y channel $\Rightarrow$ #solid.



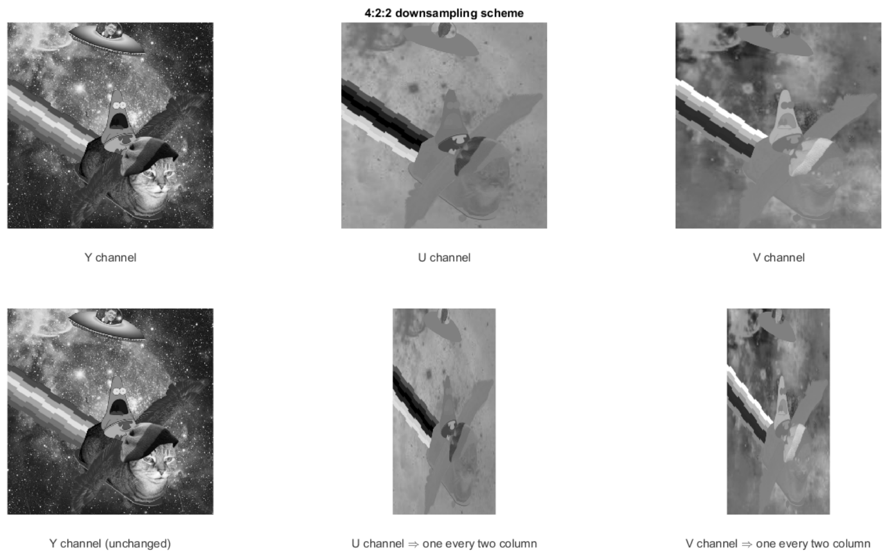

4:4:4 downsampling scheme
Y channel compression error
RMSE: 6.98
SNR: 24 dB
U channel compression error
RMSE: 5.45
SNR: 29.5 dB
V channel compression error
RMSE: 3.87
SNR: 29.4 dB

When evaluating the quality metrics in RGB space with and without compression in the YUV domain, one can see that the reconstruction is almost as good when performed in the YUV space, but the compression ratio is multiplied by over 4.

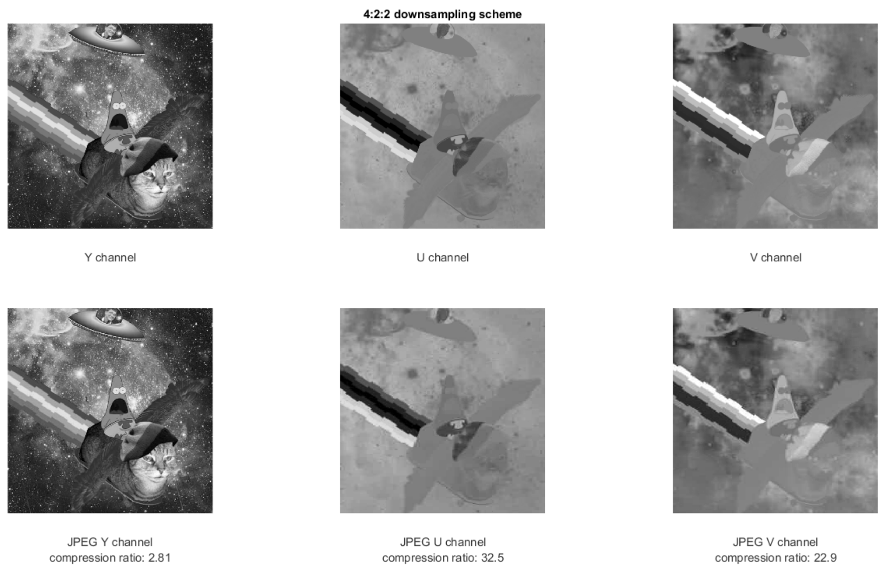

4:4:4 downsampling scheme
Before compression
After JPEG decompression in YUV colorspace
YUV compression ratio: 12.4
RGB compression ratio: 2.82
RGB compression error
mean RMSE: 7.91 (7.03 in RGB)
mean SNR: 23.6 dB (24.6 dB in RGB)

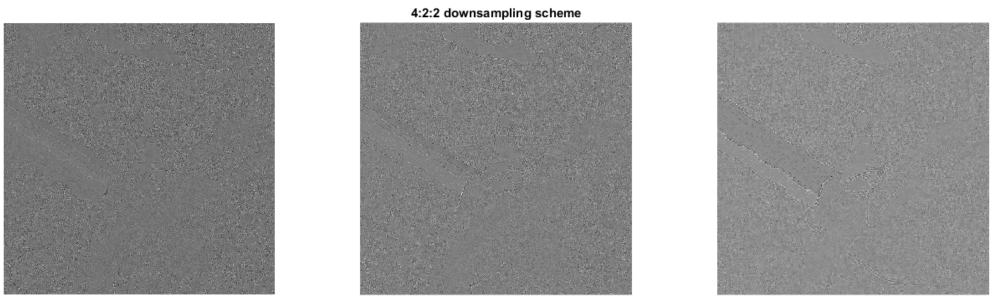
JPEG compression with 4:2:2 downsampling

The main advantage of the YUV space is that the compression can be even more brutal. As a matter of fact, the U and V channels do not contain much information. So an easy technique to save some space is to downsample the two U and V channels. In the 4:2:2 scheme, only one every two columns is retained, so half of the information is thrown away. The U and V images are then reduced by 2 ⇒ 2 pixels out of 4 are retained for U and V, while 4 out of 4 are kept for Y, hence the name 4:2:2 (and 4:4:4 when no downsampling is carried out).


4:2:2 downsampling scheme
Y channel
U channel
V channel
Y channel (unchanged)
U channel ⇒ one every two column
V channel ⇒ one every two column

During the decompression, the U and V channels must be resized to match the dimensions of the Y channel. And the simplest way to do so is to use a nearest-neighbor interpolation. It is very crude, but still works pretty well. And of course, the compression ratio gets twice better for the U and V channels (since half pixels are thrown away).





Y channel compression error
RMSE: 6.99
SNR: 24 dB

U channel compression error
RMSE: 7.72
SNR: 26.5 dB

V channel compression error
RMSE: 5.09
SNR: 27 dB

Downsampling by 2 the columns of the U and V channels turns out not to impact too much on the quality metrics (it decreases the SNR from 23.6 dB to 22.7 dB) while multipying the compression ratio by 3/2.

4:2:2 downsampling scheme

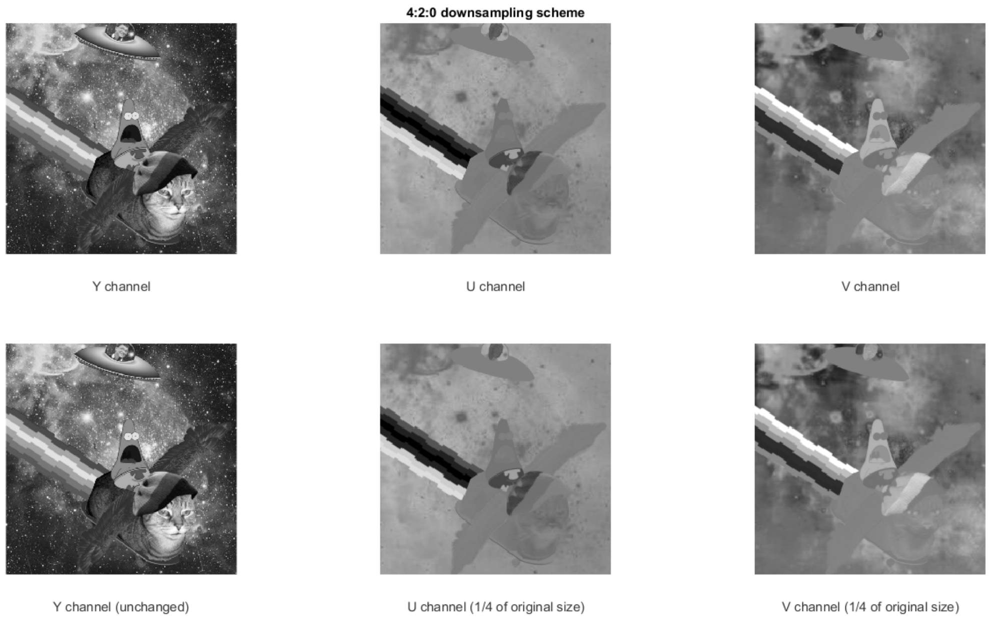
Before compression

After JPEG decompression in YUV colorspace
compression ratio: 19.4

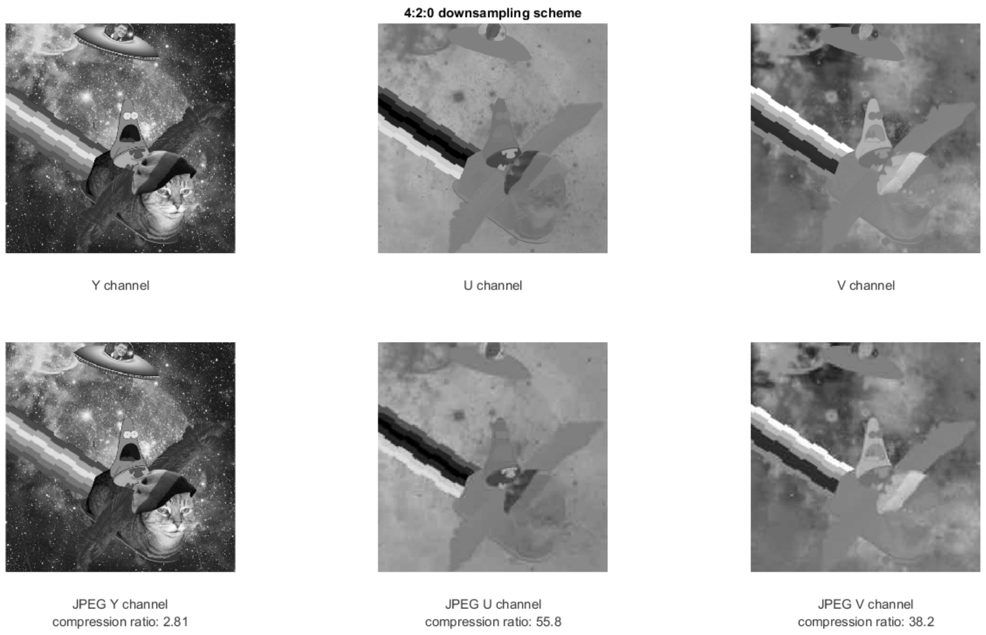
RGB compression error
mean RMSE: 8.76
mean SNR: 22.7 dB

JPEG compression with 4:2:0 downsampling

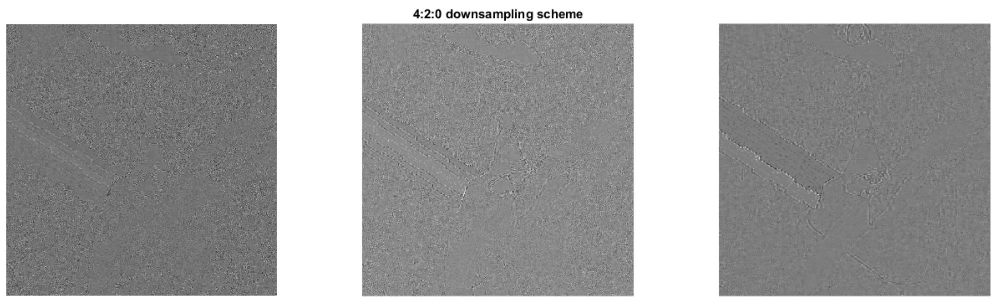
The 4:2:0 downsampling pushes to the extreme the idea of the 4:2:2 downsampling. This time, one every two lines and one every two columns is retained, which means that 3 pixels out of 4 are simply thrown away in the U and V channels during compression. And during decompression, a nearest-neighbor interpolation is conducted again.


4:2:0 downsampling scheme
Y channel
U channel
V channel
Y channel (unchanged)
U channel (1/4 of original size)
V channel (1/4 of original size)

Of course, the 4:2:0 downsampling scheme allows to reach very high compression ratios (up to 55.8 for the U channel in this example).



Nevertheless, the quality metrics remain more than acceptable, even when discarding 3/4 of the information and crudely upsampling back with nearest-neighbor interpolation.



Y channel compression error
RMSE: 7.01
SNR: 23.9 dB

U channel compression error
RMSE: 8.21
SNR: 25.9 dB

V channel compression error
RMSE: 7.52
SNR: 23.6 dB

In terms of reconstruction metrics computed in the RGB space, the 4:2:0 downsampling allows to multiply by 3 the compression factor (as opposed to the 4:4:4 case) while only reducing the SNR from 23.6 dB to 21.9 dB.

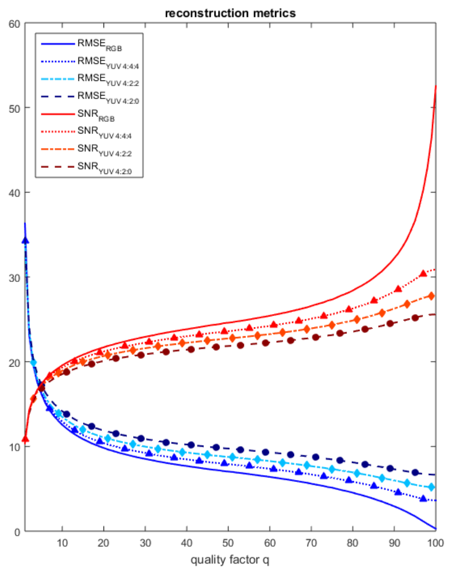
4:2:0 downsampling scheme

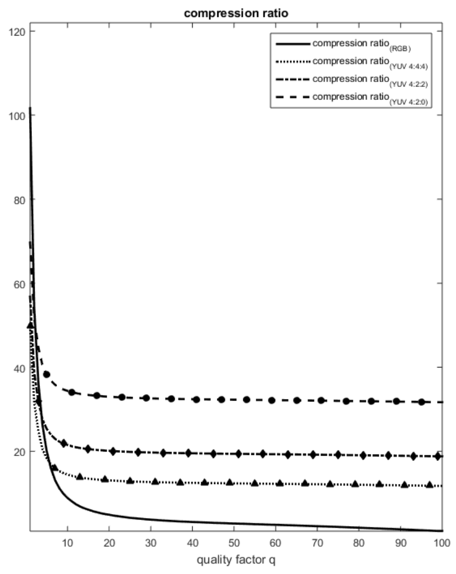
Before compression

After JPEG decompression in YUV colorspace
compression ratio: 32.3

RGB compression error
mean RMSE: 9.75
mean SNR: 21.9 dB

JPEG compression in YUV space for all quality index values





There is no real surprise when analyzing the quality metrics and compression ratio plots for the RGB and the three YUV/downsampling cases. Obviously, compressing the image in the YUV space yields better results than the RGB space in terms of compression ratio, but worse quality metrics. And the cruder the downsampling, the higher the compression ratio and the lower the quality metrics. But the amazing part is that the gain in compression ratio is totally worth the slight decrease in terms of image quality).