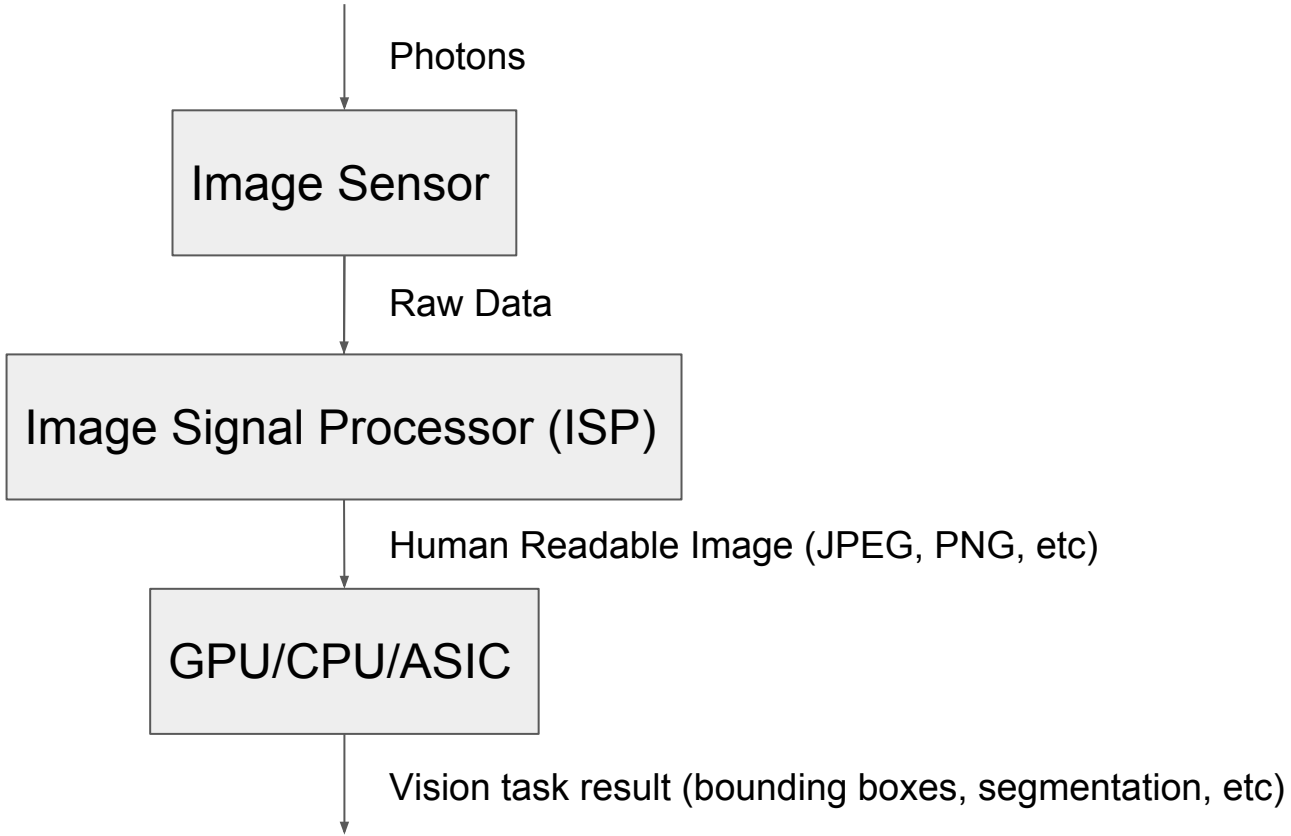
Photons to Pixels

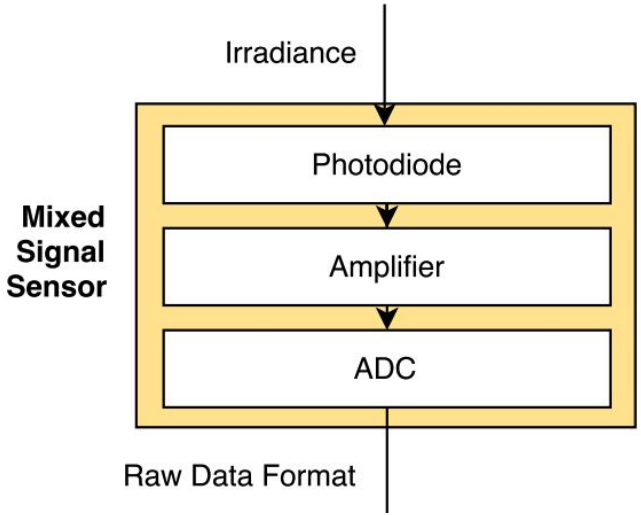
The Imaging Pipeline

Mark Buckler

The Vision Pipeline

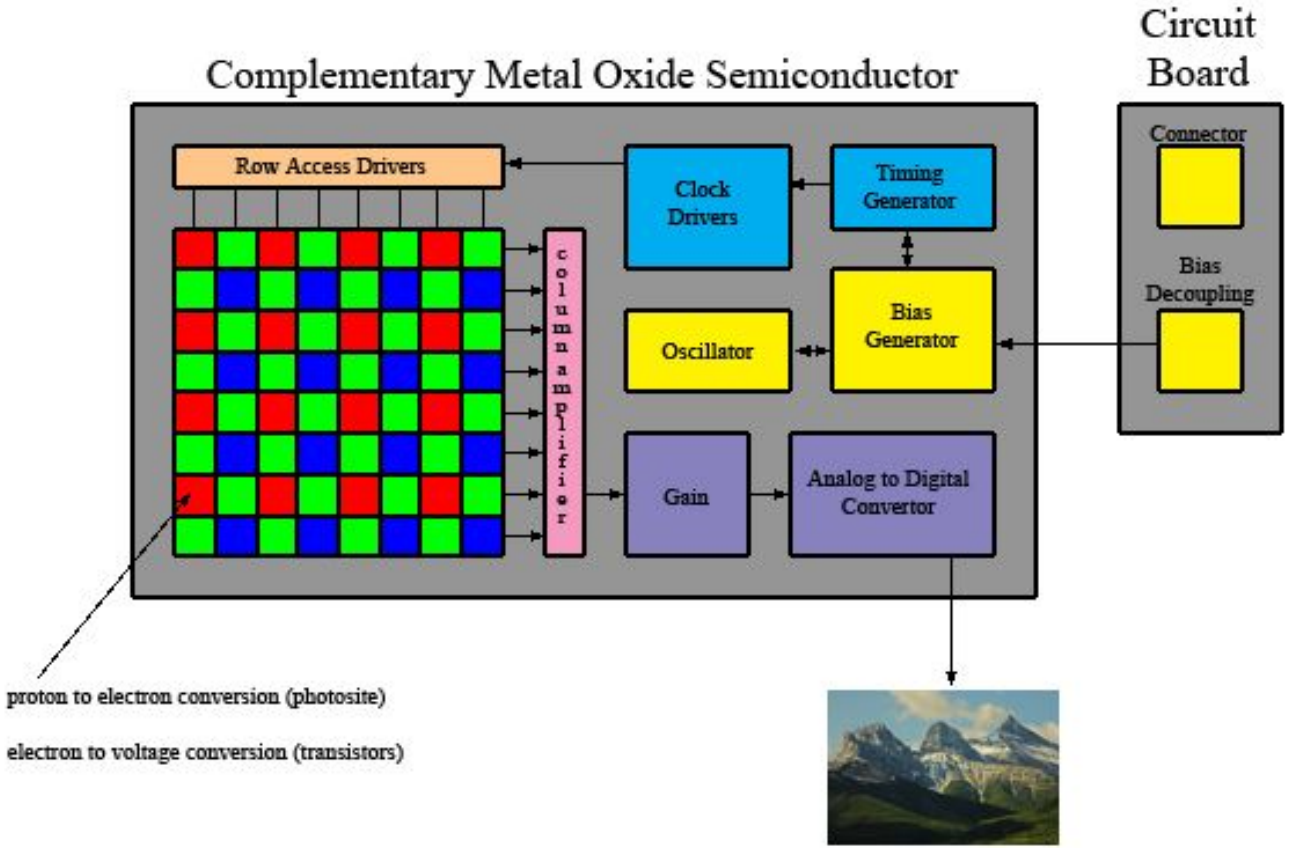


The Image Sensor (excluding optics)



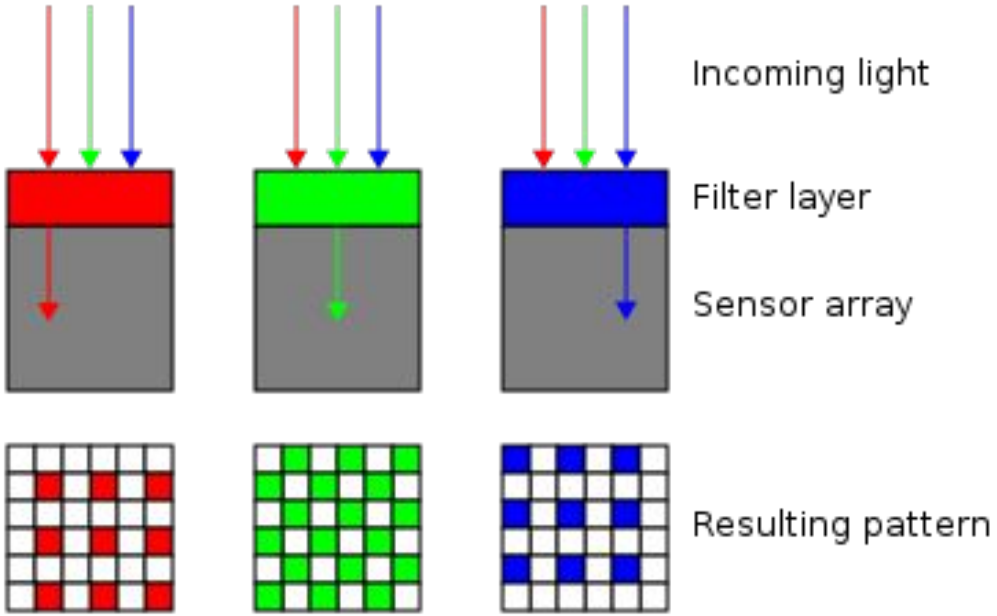
https://arxiv.org/pdf/1705.04352.pdf

The Image Sensor



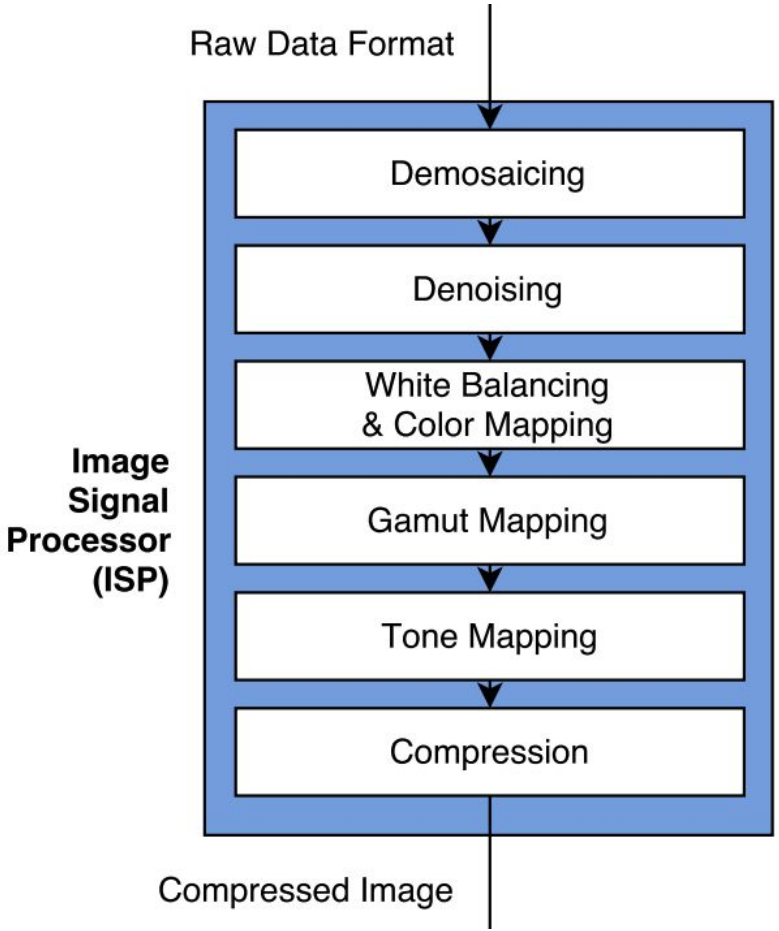
https://www.globalspec.com/learnmore/video_imaging_equipment/video_cameras_accessories/cmos_cameras

The Bayer Pattern



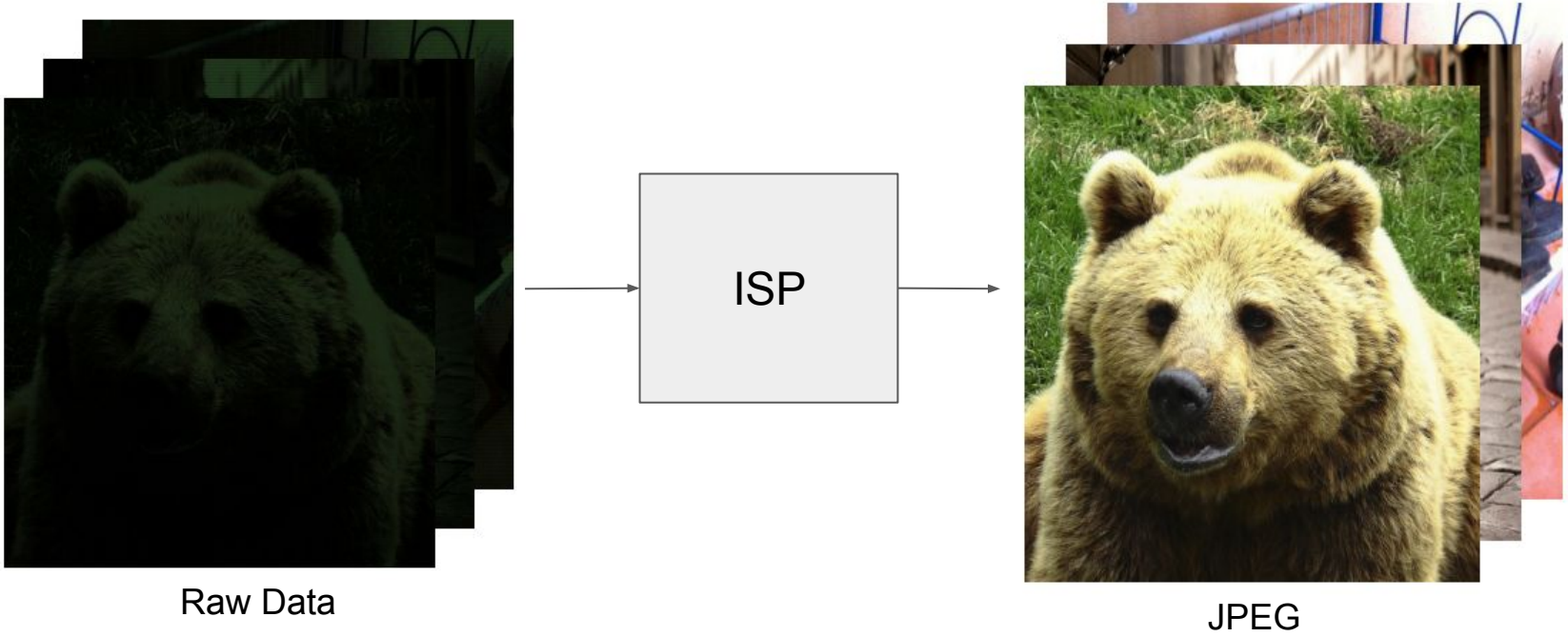
https://en.wikipedia.org/wiki/Image_sensor

The ISP



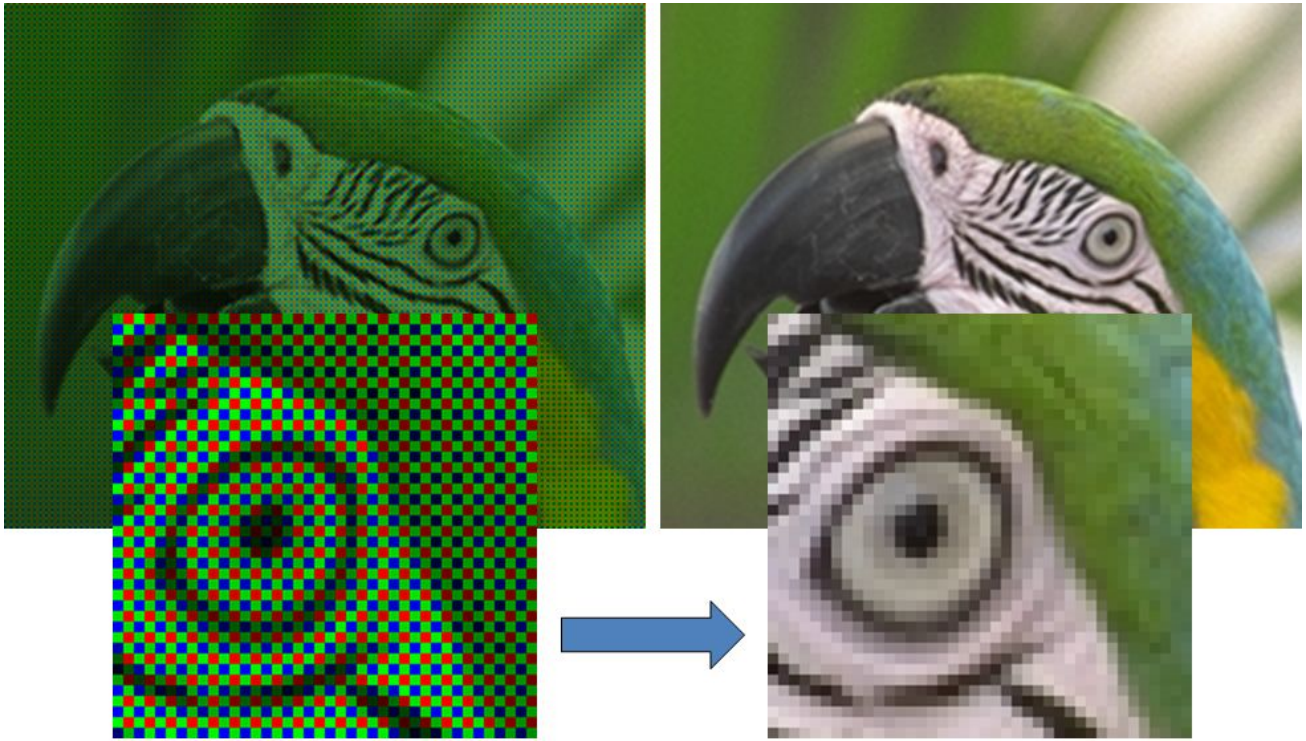
https://arxiv.org/pdf/1705.04352.pdf

The ISP: Visual effect



https://arxiv.org/pdf/1705.04352.pdf

The ISP: Demosaicing



http://www.ok.sc.e.titech.ac.jp/~mtanaka/research.html

The ISP: Denoising

- The most computationally expensive step in the ISP by far
- Especially important for low light (shot noise)
- Must balance need for smoothing aberrations with need for sharp edges

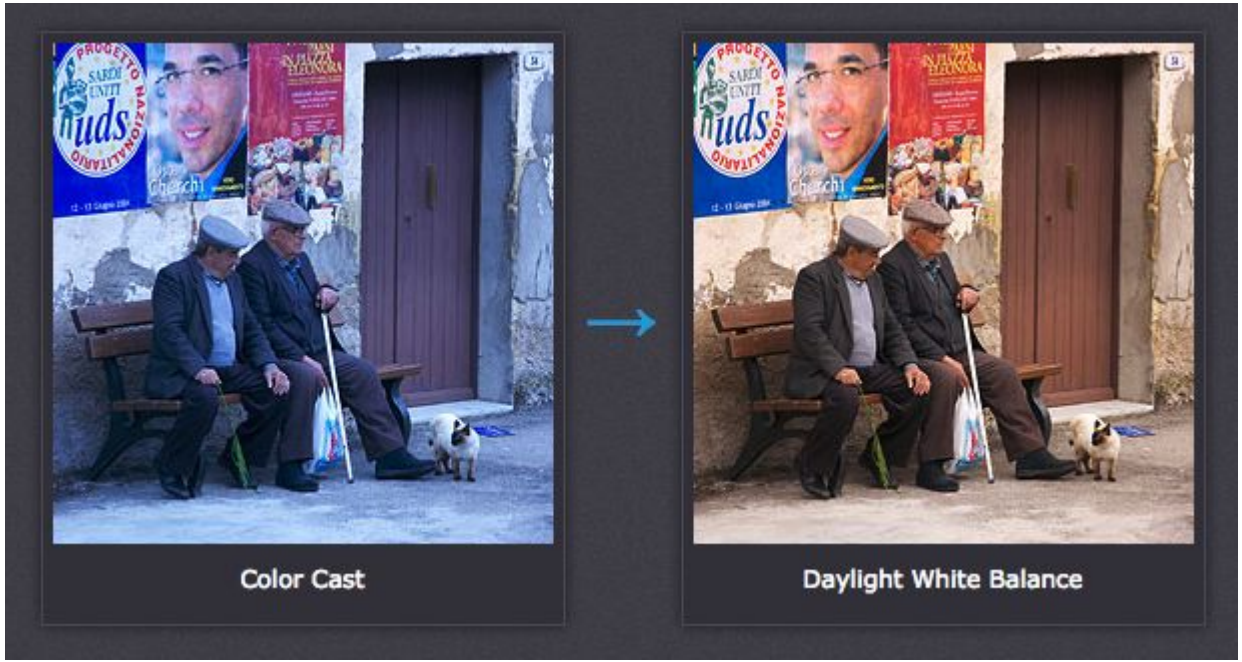
Original

Noisy image

Denoised image

https://en.wikipedia.org/wiki/Total_variation_denoising

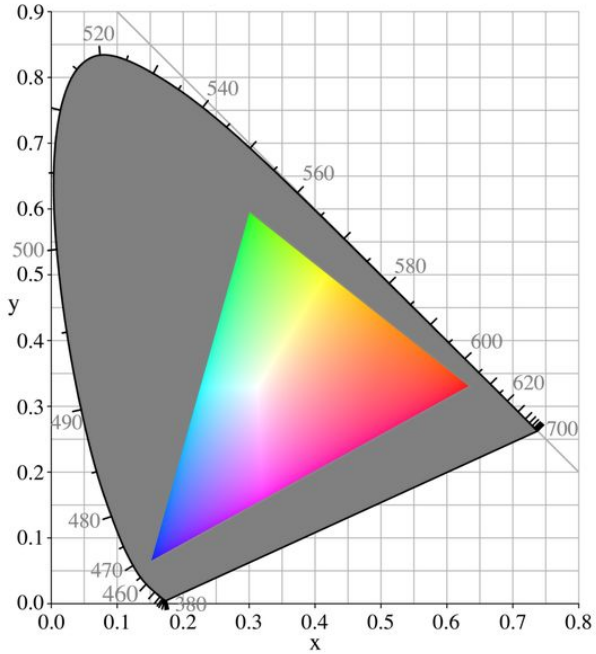
The ISP: Color Mapping & White Balancing



https://www.cambridgeincolour.com/tutorials/white-balance.htm

The ISP: Gamut Mapping

- Mapping between color systems with more or less representable range



Showing a CIE 1931 chromaticity diagram of:

- Human eye gamut (grey + colored)
- Computer monitor gamut (colored)

Each image sensor has its own gamut

https://en.wikipedia.org/wiki/Gamut

The ISP: Tone Mapping (global/gamma compression)

- PDF of natural light is log-normal.

  Terrible for linear quantization!
- Tone mapping computes log,

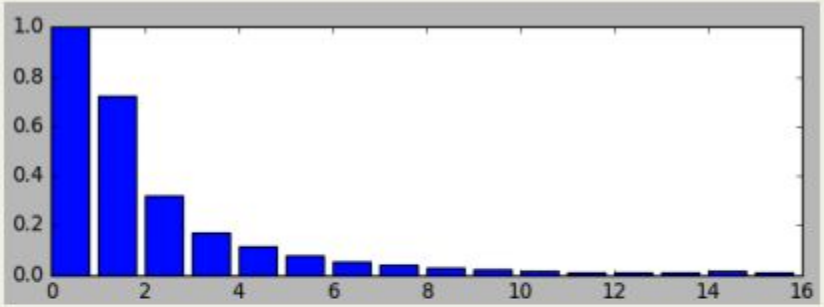
  resulting in normal distribution

$$V_{\text{out}} = AV_{\text{in}}^{\gamma}$$

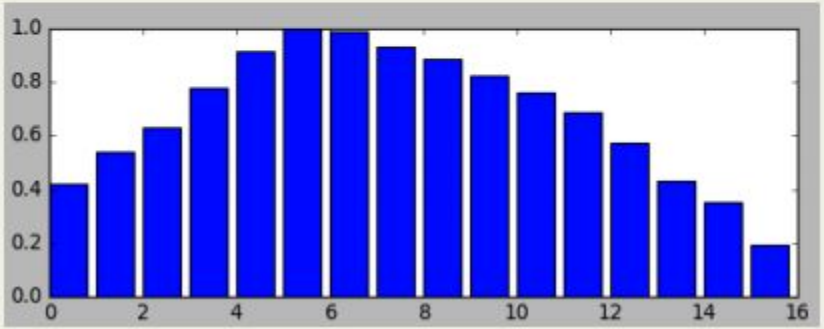
https://en.wikipedia.org/wiki/Gamma_correction

The ISP: Tone Mapping (global/gamma compression)

Raw data
(lognormal distribution)

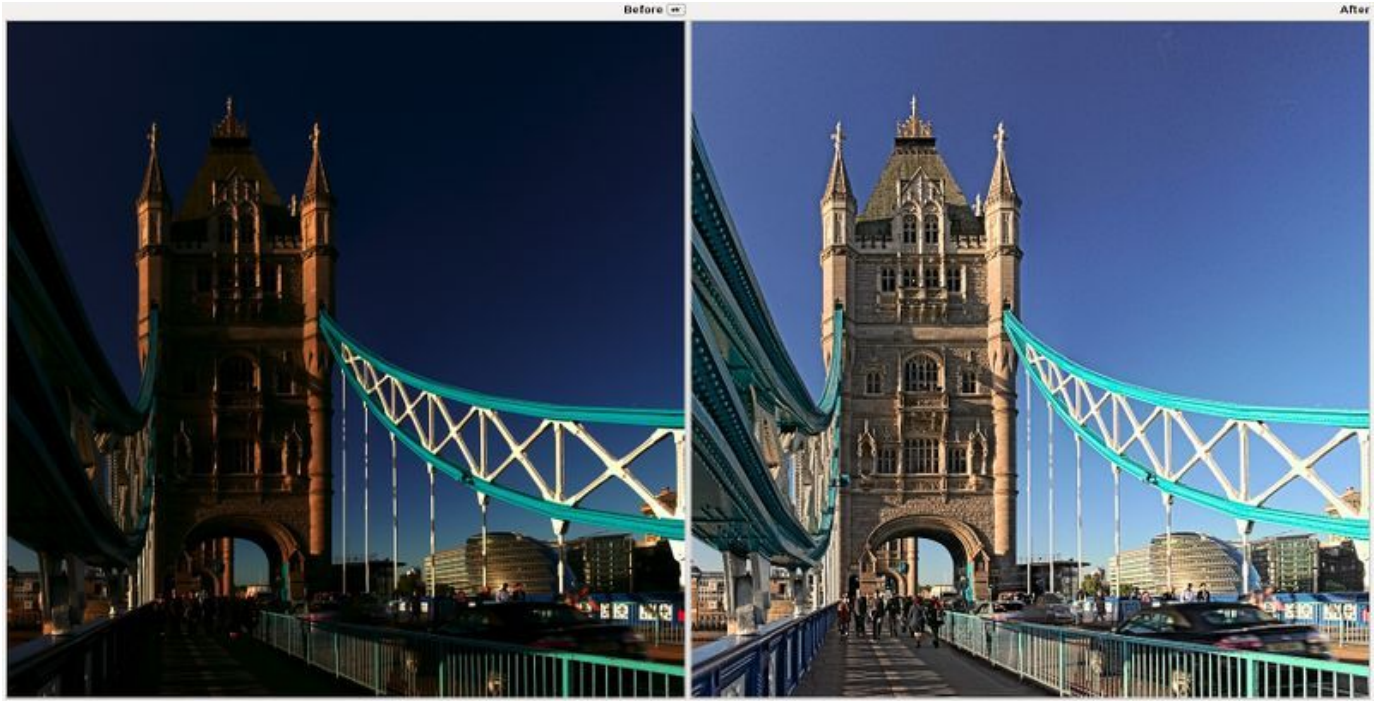


Tone mapped raw
data
(normal distribution)



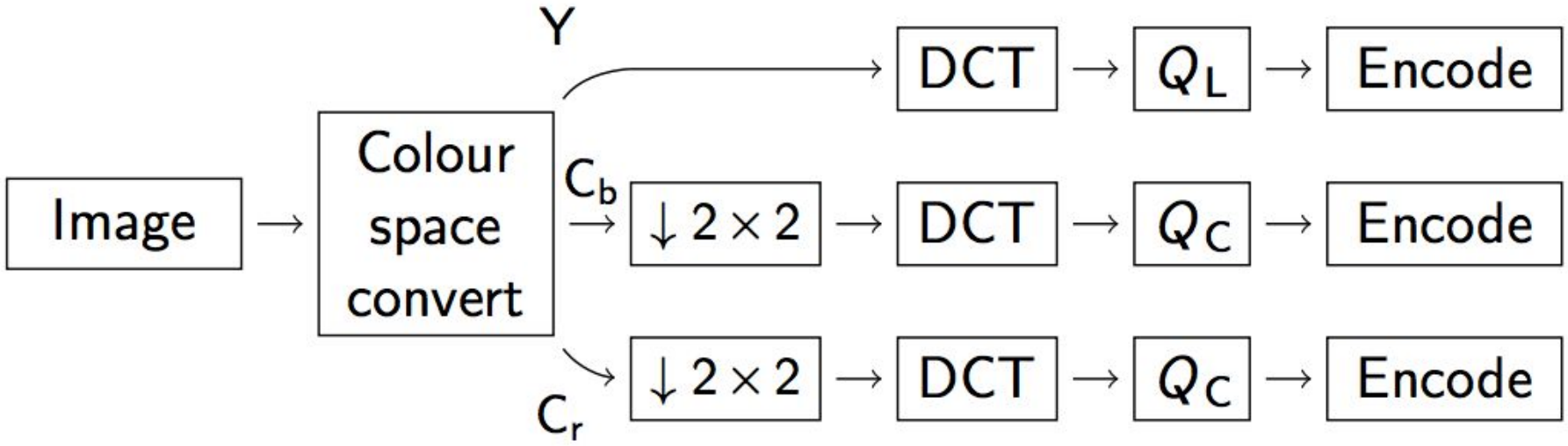
http://www.markbuckler.com/pdf/iccv-2017-slides.pdf

The ISP: Tone Mapping (local/HDR)



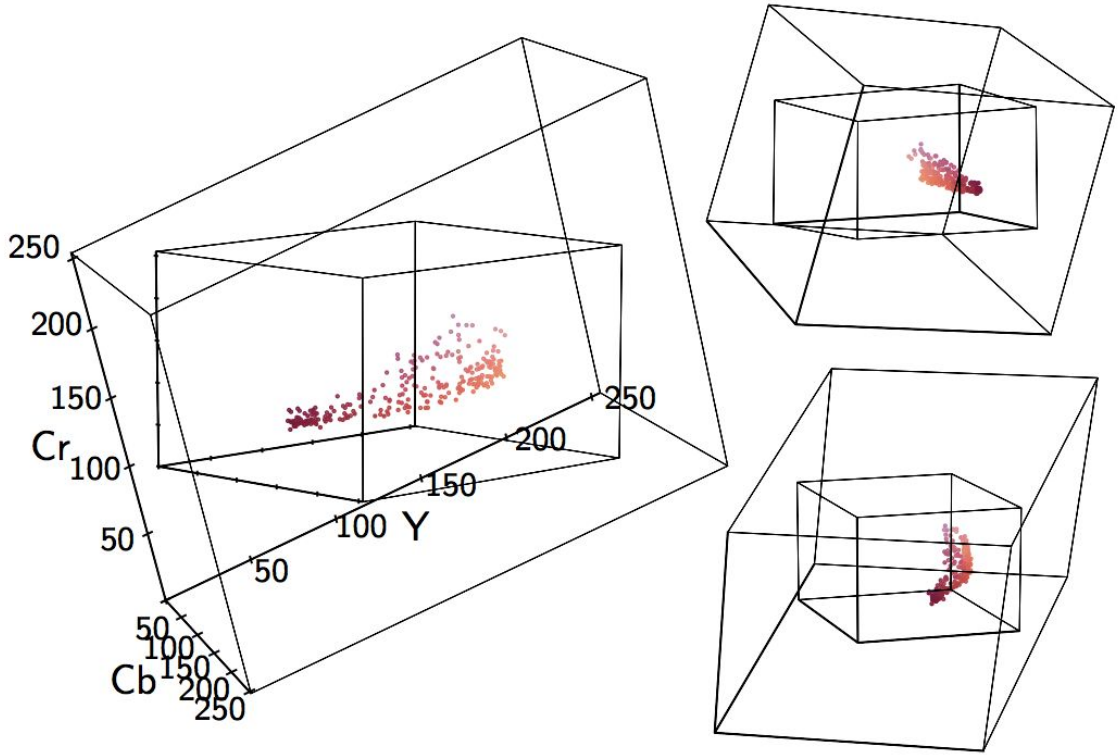
https://rawpedia.rawtherapee.com/File:Rt407-ba-tonemapping-hdr-cropped.jpg

The ISP: Compression (JPEG)



https://www.cl.cam.ac.uk/teaching/1011/R08/jpeg/acs10-jpeg.pdf

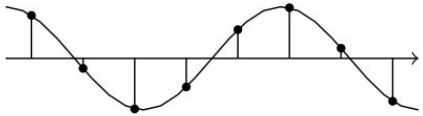
JPEG: Color space conversion

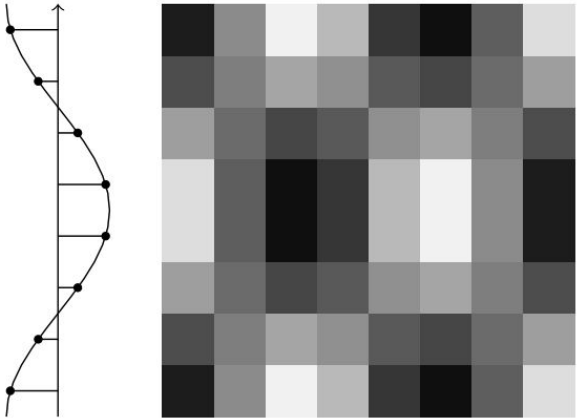


https://www.cl.cam.ac.uk/teaching/1011/R08/jpeg/acs10-jpeg.pdf

JPEG: The Discrete Cosine Transform

The 2-D DCT is a linear, separable transform which represents a block of sample values as the weighting factors of sampled cosine functions at various frequencies.

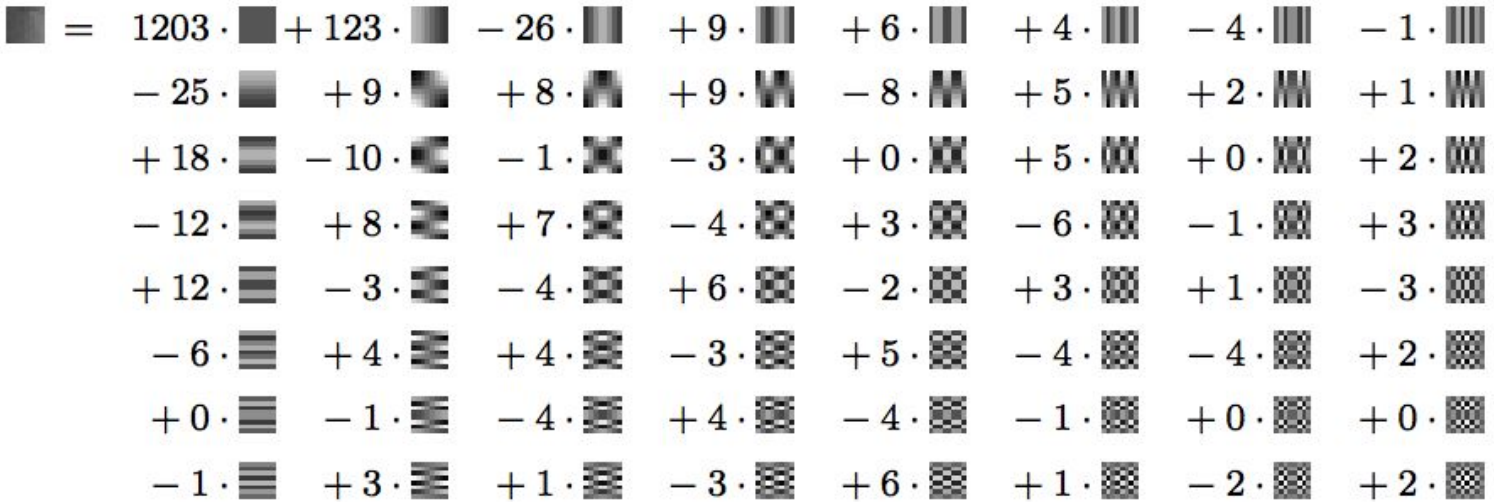




https://www.cl.cam.ac.uk/teaching/1011/R08/jpeg/acs10-jpeg.pdf

JPEG: The Discrete Cosine Transform

The transform represents an 8×8 matrix of samples as a weighted sum of the DCT basis vectors:



https://www.cl.cam.ac.uk/teaching/1011/R08/jpeg/acs10-jpeg.pdf

JPEG: Quantize

$$\begin{bmatrix} 313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\ -38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\ -20 & -17 & 10 & 33 & 21 & -6 & -16 & -9 \\ -10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\ -6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\ 2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\ 4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\ 3 & 1 & 0 & -4 & -2 & -1 & 3 & 1 \end{bmatrix} \div \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} = \begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

8 x 8 DCT Terms

Quantization table (Matrix)

Result

http://www.robertstocker.co.uk/jpeg/jpeg_new_10.htm

JPEG: Quantize

$$\begin{bmatrix} 313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\ -38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\ -20 & -17 & 10 & 33 & 21 & -6 & -16 & -9 \\ -10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\ -6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\ 2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\ 4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\ 3 & 1 & 0 & -4 & -2 & -1 & 3 & 1 \end{bmatrix} \div \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} = \begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

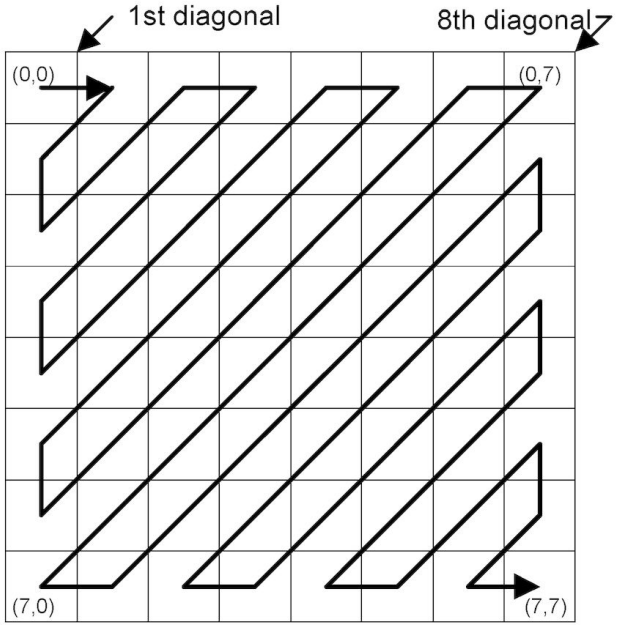
8 x 8 DCT Terms | Quantization table (Matrix) | Result

Notice the blurring around the characters in this image?
That blurring is directly because of this quantization step

http://www.robertstocker.co.uk/jpeg/jpeg_new_10.htm

JPEG: Encode

$$\begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Zig-zag run length encoding

http://www.robertstocker.co.uk/jpeg/jpeg_new_10.htm

The End!