image sensing.

noise: unwanted modification of signal during capture / transmission / processing / storage of signal.

1. photon shot noise: a random process, can't be optimised.
2. readout noise (convert electrons to voltage, using either CCD or CMOS.
3. other noise: thermal noise (high temp. electrons are ejected from silicon problem with long exposures) fixed pattern noise: defective pixels. astronomy

1. photo arrive the sensor ramdomly w/ Poisson distr.
   Possion distr. has mean = variance = λ.
   as light intensity ↑, due to CLT tends to Gaussian distr. w/ mean = var.
   Also, issue improves at well-lit areas: if $\lambda = 25$, 25 ± 5 is less optimal than $\lambda = 10000$, 10000 ± 100 is better.
   (scene dependent, nothing to do w/ sensor)

$$P(\text{signal} = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

2. read noise (modelled as Gaussian, scene independent, depend on quality of the sensor) good sensor has ↓ σ.

$$P(\text{signal} = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

~~dynamic~~ sensor dynamic range.

range of brightness a certain image sensor can measure:

max. capacity the well can detect. (# photons)

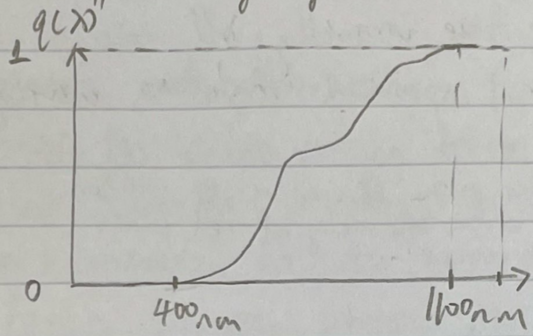
$$\text{Dynamic Range} = 20 \log\left(\frac{B_{max}}{B_{min}}\right) \quad (\text{dB})$$

min. detectable photon energy (# photons) photon flux $p(\lambda)$: just a freq. diagram, 1 photon @ 400nm, 2 photons @ 700 nm...

quantum efficiency $q(\lambda) = \frac{\text{\# electrons generated}}{\text{\# photon received}}$

(remember silicon chips receive photons, and electrons are ejected due to displacement)
so $q(\lambda)$ measure the "success detection" rate at light wavelength λ.

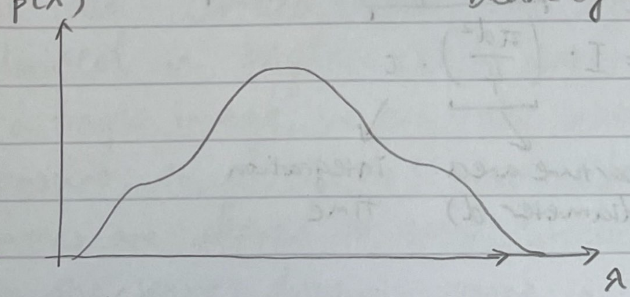
quantum efficiency of silicon:



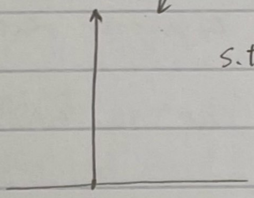
→ i.e. transparent, 100% efficiency when $\lambda \geq 1100$nm, useless when $\lambda \leq 400$nm.

electron flux $I = q(\lambda_i)p(\lambda_i)$ for monocromatic light w/ wavelength λ_i eg. 80% × 5 photons = 4 electrons (estimated)

spectral distribution $p(\lambda)$: the distribution of # photons in a beam of light.



$I = \int_0^\infty q(\lambda)p(\lambda)d\lambda$ → we know $q(\lambda)$ & $q(\lambda_i)$ due to property of the material (eg. silicon) to obtain $p(\lambda_i)$ (like, # photons at freq. λ_i), mathematically, we use Dirac func. $\delta(\lambda-\lambda_i)$



s.t. $\int_0^\infty \delta(\lambda-\lambda_i)\,d\lambda = 1$

λ_i ← eg. a filter for red/blue/green lights

then,

$$I = \int_0^\infty q(\lambda)\,p(\lambda)\,\delta(\lambda-\lambda_i)\,d\lambda \implies I = q(\lambda_i)\,p(\lambda_i)$$

3 colours: R, G, B can create virtually all colours a human eye can perceive. (even if spectral distr. is vastly different, human eye still perceive it as the same colour; due to

similar to the efficiency of silicon,

~~this is the efficiency of~~ red cones.

$$R = \int_{-\infty}^{\infty} h_r(\lambda)\,p(\lambda)\,d\lambda.$$

$$G = \int_{-\infty}^{\infty} h_g(\lambda)\,p(\lambda)\,d\lambda$$

$$B = \int_{-\infty}^{\infty} h_b(\lambda)\,p(\lambda)\,d\lambda$$

camera response func $f(\cdot)$

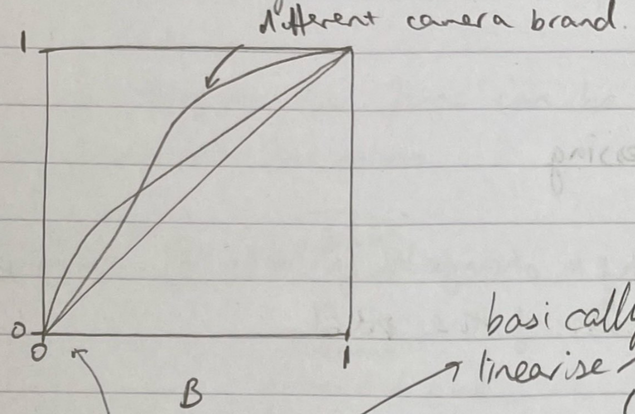
image brightness (scene) $B = \underset{\text{\# electrons}}{I} \cdot \overbrace{\underbrace{\left(\frac{\pi d^2}{4}\right)}_{\text{aparture area (diameter } d)} \cdot \underbrace{t}_{\text{integration time}}}^{e,\ \text{exposure}}$

measured brightness ~~M(B) = M(I·e)~~

$$M = f(B) = f(I \cdot e)$$

possibly non-linear

camera response func. are introduced to be non-linear. (aka. gamma curves) ie. brightness from what sensors pick up to image generation is adjusted, so that dimmer areas are discerned w/ greater resolution.



basically, we want to linearise the function. (brightness)

radiometric ~~cab~~ calibration: figuring (estimating) the camera response function. why? When we work on radiometric tasks such as scientific imaging, we want to correct the image into scene brightness, so it's important to know the gamma curve. We have to use .raw images for calibration though, because .jpeg image modify brightness (on top of brightness modified by CRF). and camera companies keep CRF confidential so you have to work out the function by myself.

<u>high dynamic range: multiple exposures.</u>

basic idea: camera (sensor) has a dynamic range, which is v. limited in brightness. To express more brightness information on a single image, we can take several images w/ different exposure, e_0, e_1, e_2, e_3. (given different exposure time, more/less photons are allowed to enter sensors).

Say the sensor's dynamic range is 255. by adding up 4 photos w/ different exposure, we're able to express a single image of dynamic range $255 \times 4 = 1020$, far surpass sensor's capability.

drawback: due to multiple shots taken, moving obj. are going to appear at different locations on that photo.

single-shot HDR imaging

some sensors are designed s.t. neighbouring pixels have different exposure time. This way it creates a high contrast image w/o motion artifact (resolution is sacrifised)

Image Processing

pixel processing: change brightness of every pixel, irrespect of the location of the pixel.

Linear Shift Invariant System (LSIS) (lenses are LSIS)

1. linear if $LSIS(f_1) = g_1$, $LSIS(f_2) = g_2$, $LSIS(\alpha f_1 + \beta f_2) = \alpha g_1 + \beta g_2$
2. S.I if $LSIS(f_1(x-a)) = LSIS(g_1(x-a))$

convolution ($\Rightarrow$ LSIS) of 2 func. $f(x) * h(x)$

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau) h(x-\tau) d\tau$$

$\hookrightarrow$ impulse response

$a*b = b*a$; $(a*b)*c = a*(b*c)$

2D convolution: $f[i,j] \rightarrow$ func $h[i,j]$ $\rightarrow$ output $g[i,j]$

$$g[i,j] = \sum_{m=1}^{M} \sum_{n=1}^{N} f[m,n] \underbrace{h[i-m, j-n]}_{\text{kernel}}$$

sliding kernel $h[-m,-n]$ to all $[i,j]$ pixel in img.

eg. Guassian kernel: size $K \times K$ where $K \approx 2\pi\sigma$ $\rightarrow$ std. div.

non-linear filter

bilateral filter: $$g[i,j] = \frac{1}{W_s} \sum_m \sum_n f[m,n] \underbrace{n_{\sigma_s}[i-m, i-n]}_{\text{spatial Guassian}} \underbrace{n_{\sigma_b}(f[m,n] - f[i,j])}_{\text{brightness Guassian}}$$

compare the brightness difference between pixel $[i,j]$ & $[m,n]$. if difference is small, give large weight so, for each pixel $[i,j]$, $g[i,j]$ are weighted s.t. only pixels w/ similar brightness (ie. large weight) are considered for Gaussian smoothing.

Fourier transformation: any periodic func can be expressed as an infinite sum of sinusoidal functions.

IFT:

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ux} du$$

$f(x)$: original signal

$e^{i2\pi ux}$: different sinusoidal func.

u: frequency

$e^{i\theta} = \cos\theta + i\sin\theta$

FT: $$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

FT good for transformation for signals from spatial domain ↔ freq. domain.

spatial domain: $g(x) = f(x) * h(x)$

freq. domain: $G(u) = F(u)H(u)$.

convolution in spatial domain ≡ multiplication in freq. domain. vice versa.

Why? → FT & IFT are faster to compute... compared to convolution. so, if we want high dim. $g(x) = f(x) * h(x)$, First, $f(x) \xrightarrow{FT} F(u)$, $h(x) \xrightarrow{FT} H(u)$, $G(u) = F(u)H(u)$ then $G(u) \xrightarrow{IFT} g(x)$.