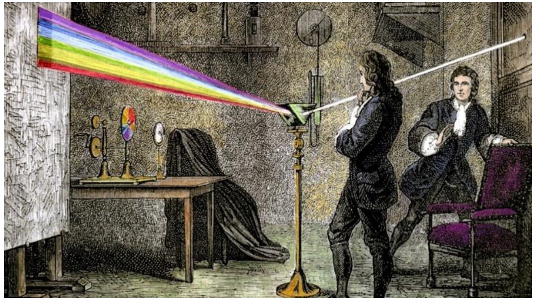
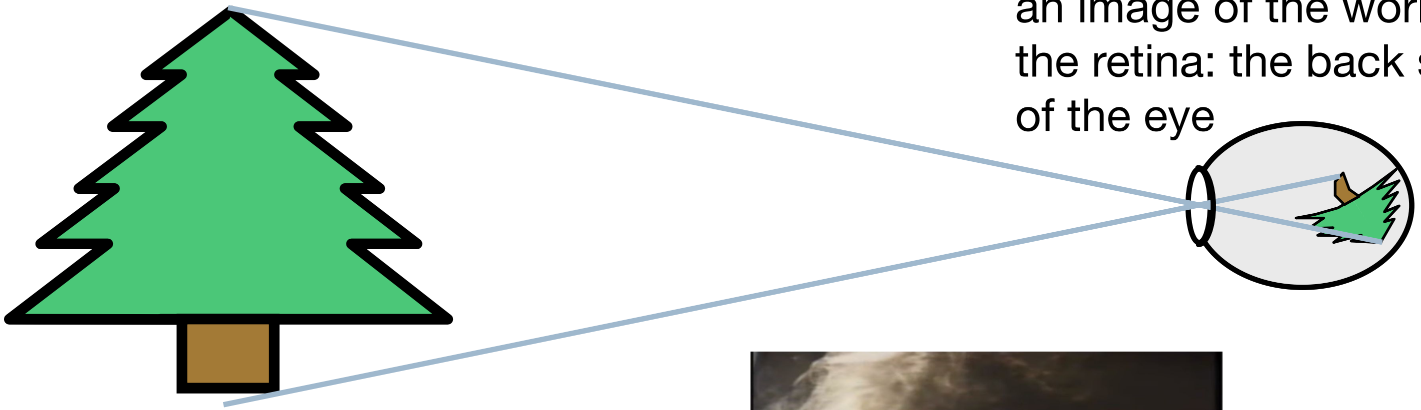
Introduction to Computer Graphics

- Background
- Rendering
- Graphics pipeline
- Rasterization
- Graphics hardware and OpenGL
- **Human vision and colour, tone mapping**



The workings of the human visual system

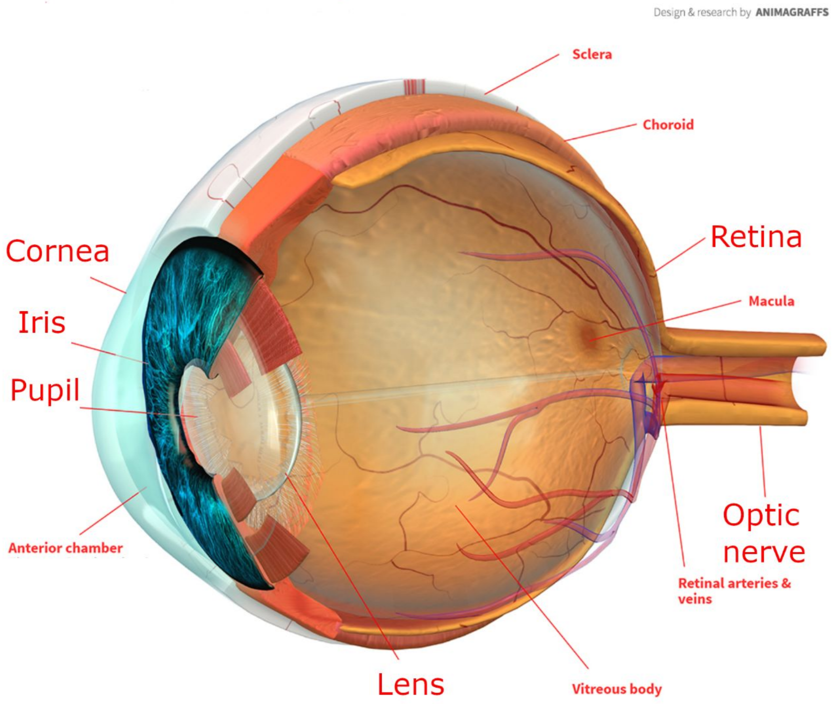
- to understand the requirements of displays (resolution, quantisation and colour) we need to know how the human eye works...



The lens of the eye forms an image of the world on the retina: the back surface of the eye

Inverted vision experiment

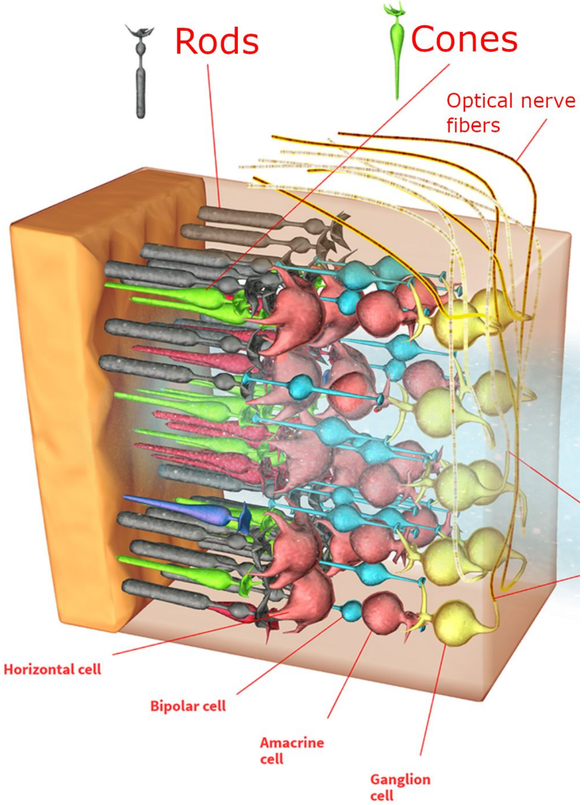
Structure of the human eye



See Animagraffs web page for an animated visualization
https://animagraffs.com/human-eye/

- the retina is an array of light detection cells
- the fovea is the high resolution area of the retina
- the optic nerve takes signals from the retina to the visual cortex in the brain
- cornea and lens focus the light on the retina
- pupil shrinks and expands to control the amount of light

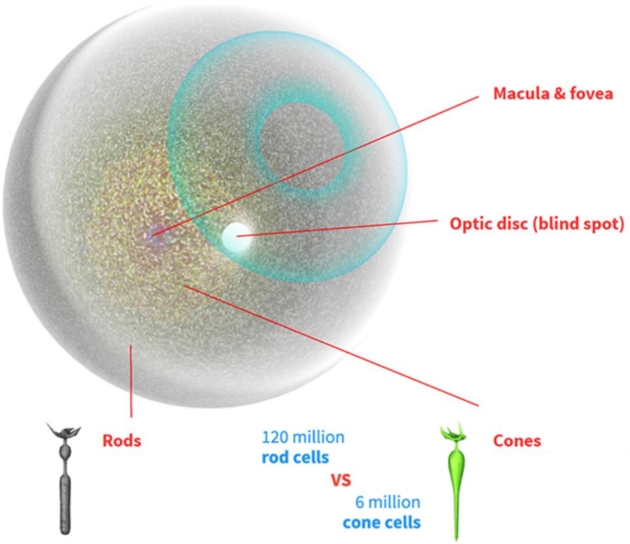
Retina, cones and rods

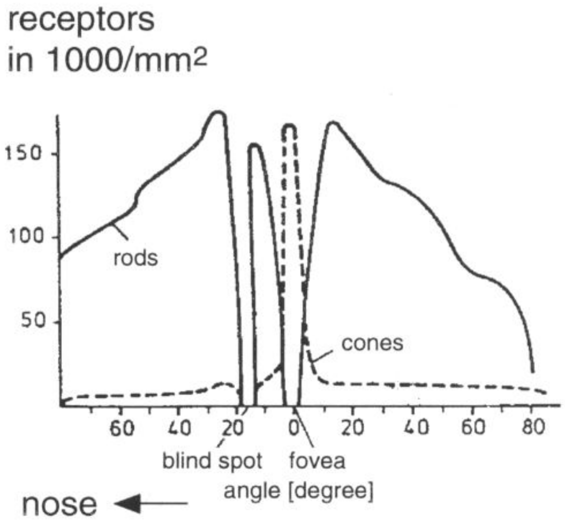


- 2 classes of photoreceptors
  - Cones are responsible for day-light vision and colour perception
    - Three types of cones: sensitive to short, medium and long wavelengths
  - Rods are responsible for night vison

Fovea, distribution of photoreceptors

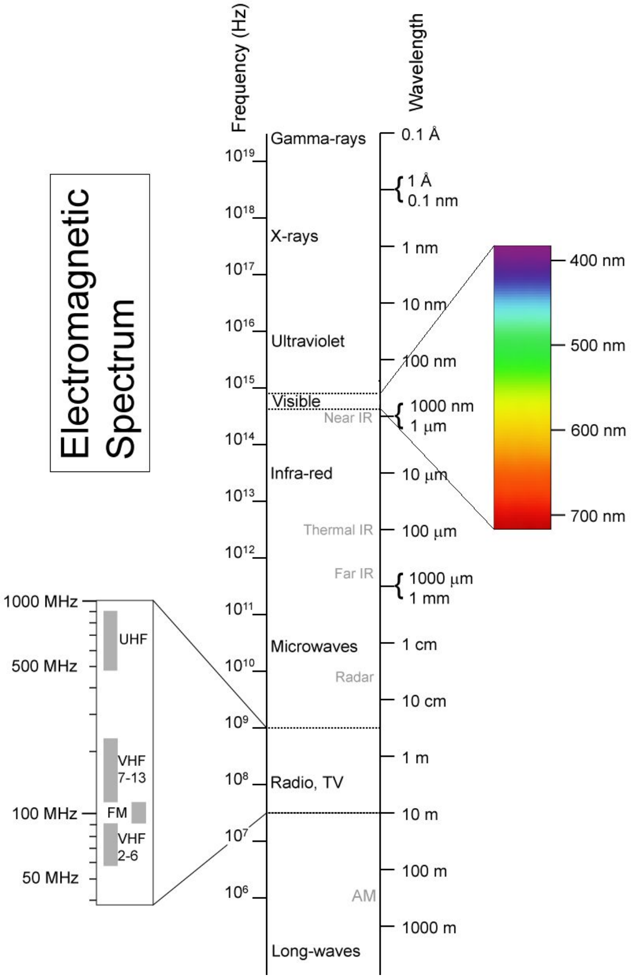
- the fovea is a densely packed region in the centre of the macula
  - contains the highest density of cones
  - provides the highest resolution vision





Electromagnetic spectrum

- Visible light
  - Electromagnetic waves of wavelength in the range 380nm to 730nm
  - Earth's atmosphere lets through a lot of light in this wavelength band
  - Higher in energy than thermal infrared, so heat does not interfere with vision

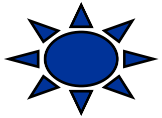


Colour

- There is no physical definition of colour – colour is the result of our perception

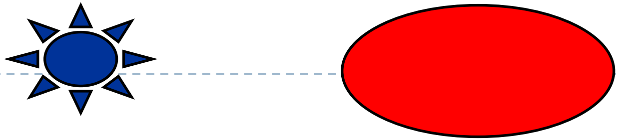
- For emissive displays / objects

colour = perception(spectral_emission)



- For reflective displays / objects

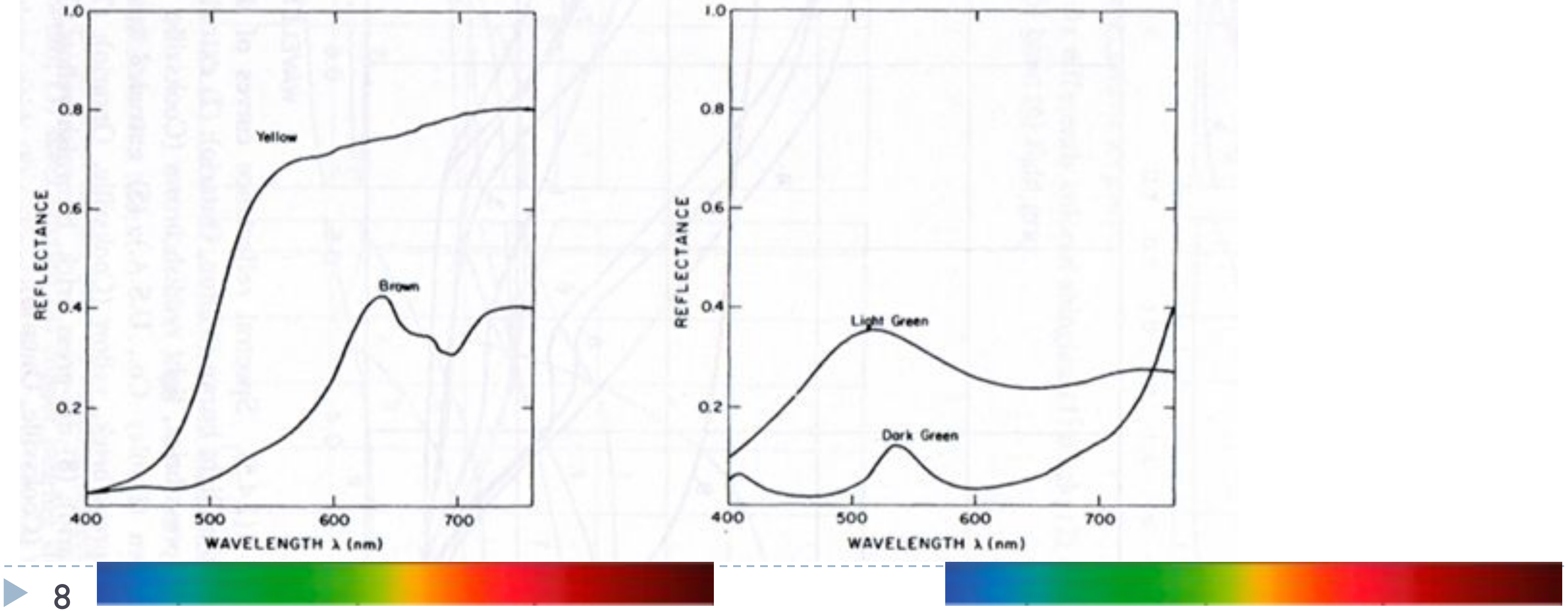
colour = perception(illumination × reflectance)



Reflectance

- Most of the light we see is reflected from objects
- These objects absorb a certain part of the light spectrum

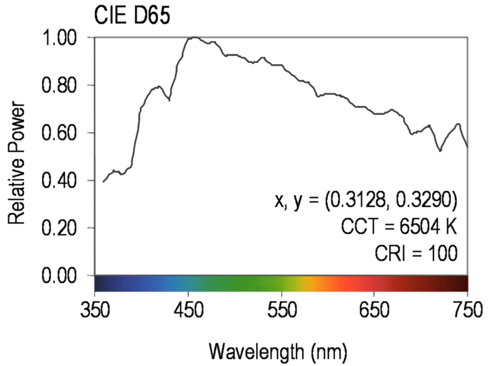
Spectral reflectance of ceramic tiles

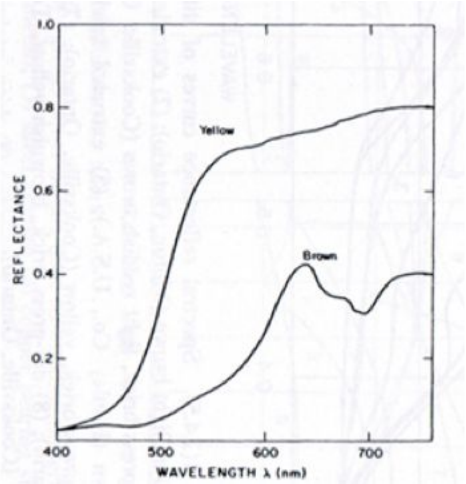


Reflected light

$$L(\lambda) = I(\lambda)R(\lambda)$$

- Reflected light = illumination × reflectance

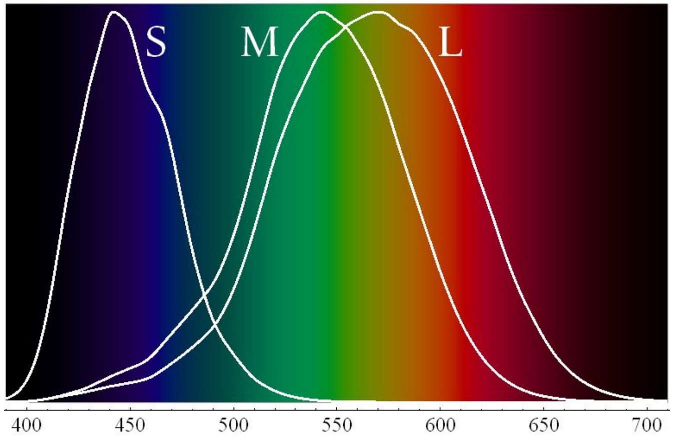




The same object may appear to have different colour under different illumination.

Colour vision

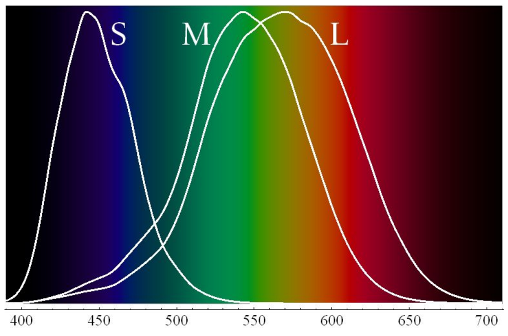
- Cones are the photreceptors responsible for color vision
  - Only daylight, we see no colors when there is not enough light
- Three types of cones
  - S – sensitive to short wavelengths
  - M – sensitive to medium wavelengths
  - L – sensitive to long wavelengths

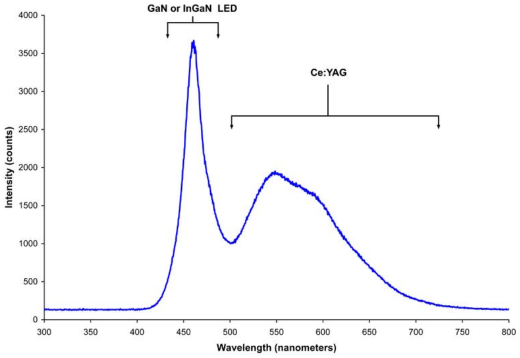


Sensitivity curves – probability that a photon of that wavelengths will be absorbed by a photoreceptor. S,M and L curves are normalized in this plot.

Perceived light

- cone response = sum(sensitivity × reflected light)





Although there is an infinite number of wavelengths, we have only three photoreceptor types to sense differences between light spectra

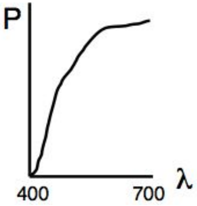
Formally

$$R_S = \int_{380}^{730} S_S(\lambda) \cdot L(\lambda) d\lambda$$

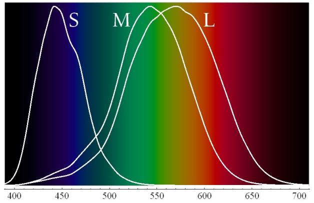
Index S for S-cones

Metamers

- Even if two light spectra are different, they may appear to have the same colour
- The light spectra that appear to have the same colour are called **metamers**
- Example:

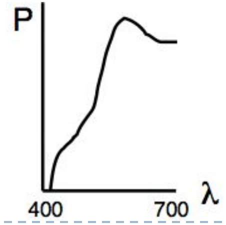


*

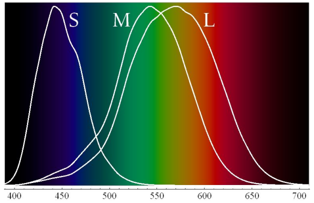


$= [L_1, M_1, S_1]$

||



*

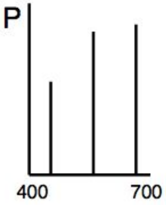


$= [L_2, M_2, S_2]$

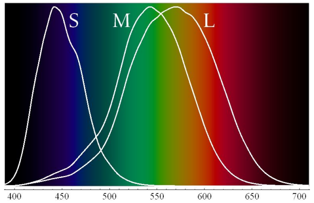
Practical application of metamerism

- Displays do not emit the same light spectra as real-world objects
- Yet, the colours on a display look almost identical

On the display

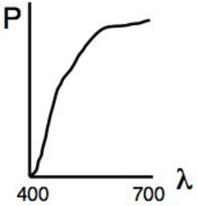


*

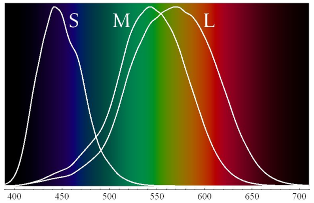


$= [L_1, M_1, S_1]$

=



*

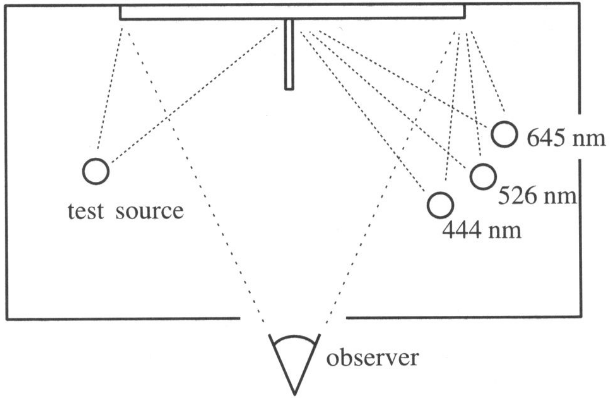


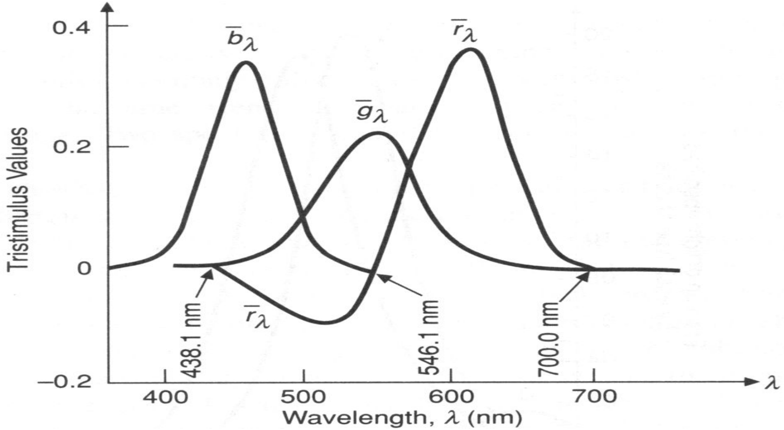
$= [L_2, M_2, S_2]$

In real world

Tristimulus Colour Representation

- Observation
  - Any colour can be matched using three linear independent reference colours
  - May require "negative" contribution to test colour
  - Matching curves describe the value for matching mono-chromatic spectral colours of equal intensity
    - With respect to a certain set of primary colours



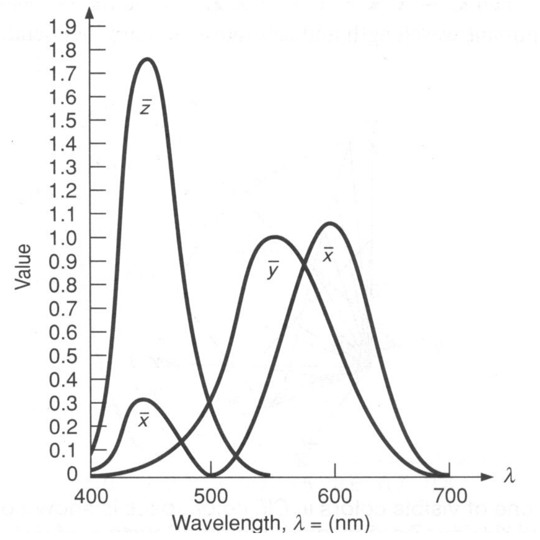


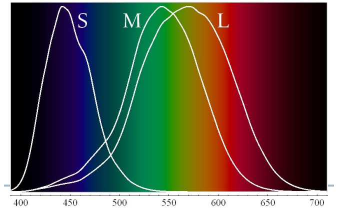
Standard Colour Space CIE-XYZ

- CIE Experiments [Guild and Wright, 1931]
  - Colour matching experiments
  - Group ~12 people with normal colour vision
  - 2 degree visual field (fovea only)
  - Basis for CIE XYZ 1931 colour matching functions
- CIE 2006 XYZ
  - Derived from LMS color matching functions by Stockman & Sharpe
  - S-cone response differs the most from CIE 1931
- CIE-XYZ Colour Space
  - Goals
    - Abstract from concrete primaries used in experiment
    - All matching functions are positive
    - Primary „Y" is roughly proportionally to light intensity (luminance)

Standard Colour Space CIE-XYZ

- Standardized imaginary primaries CIE XYZ (1931)
  - Could match all physically realizable colour stimuli
  - Y is roughly equivalent to luminance
    - Shape similar to luminous efficiency curve
  - Monochromatic spectral colours form a curve in 3D XYZ-space





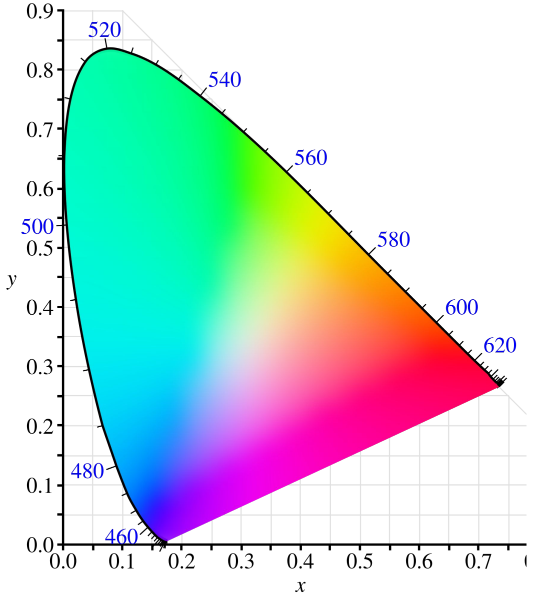
Cone sensitivity curves can be obtained by a linear transformation of CIE XYZ

CIE chromaticity diagram

- *chromaticity* values are defined in terms of x, y, z

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z} \quad \therefore \quad x+y+z=1$$

  - ignores luminance
  - can be plotted as a 2D function
- pure colours (single wavelength) lie along the outer curve
- all other colours are a mix of pure colours and hence lie inside the curve
- points outside the curve do not exist as colours



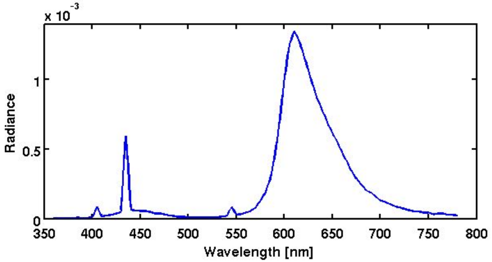
Luminance

- Luminance – measure of light weighted by the response of the achromatic mechanism. Units: cd/m^2 (ISO) or nit

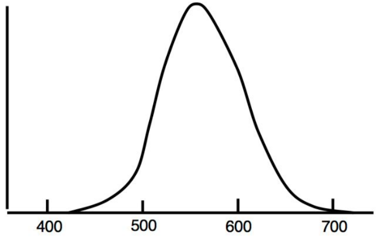
Luminance

$$L_V = \int_{350}^{700} kL(\lambda)V(\lambda)d\lambda \qquad k = \frac{1}{683.002}$$

Light spectrum (radiance)

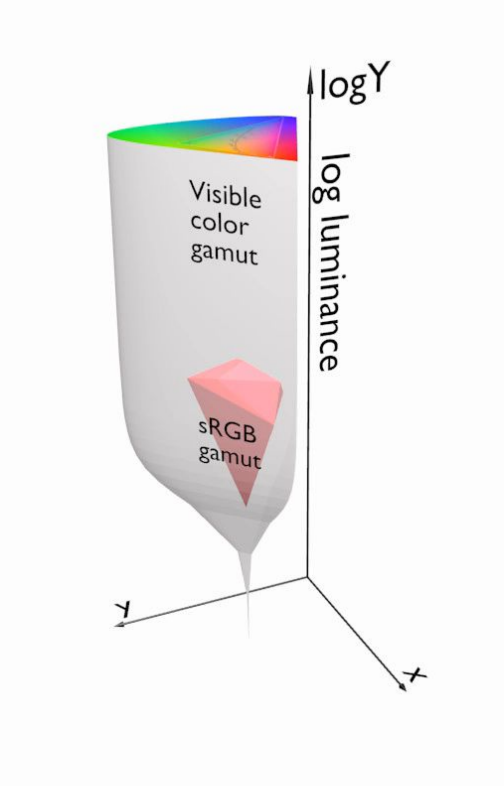


Luminous efficiency function (weighting)



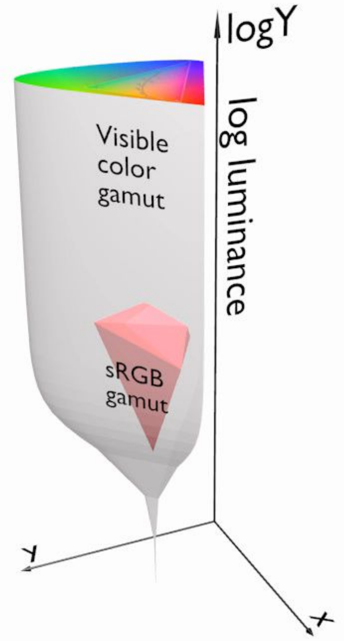
Visible vs. displayable colours

- All physically possible and visible colours form a solid in XYZ space
- A chromacity diagram is a slice taken from a 3D solid in XYZ space
- Each display device can reproduce a subspace of that space
- Colour Gamut – the solid in a colour space
  - Usually defined in XYZ to be device-independent

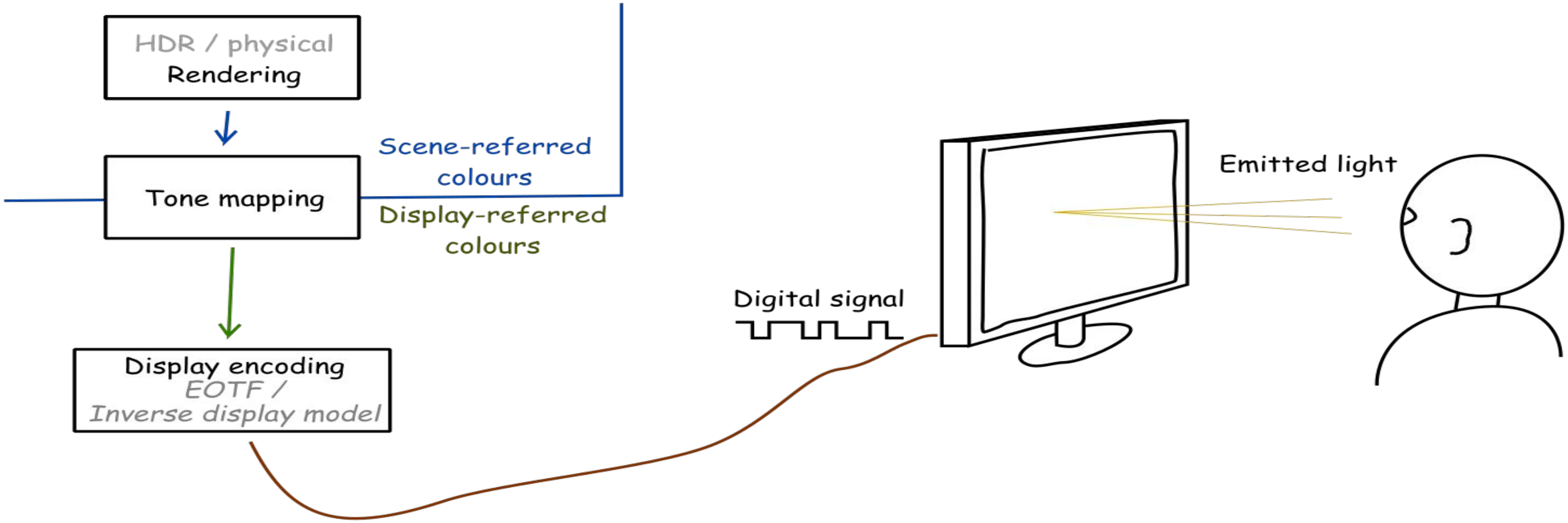


Standard vs. High Dynamic Range

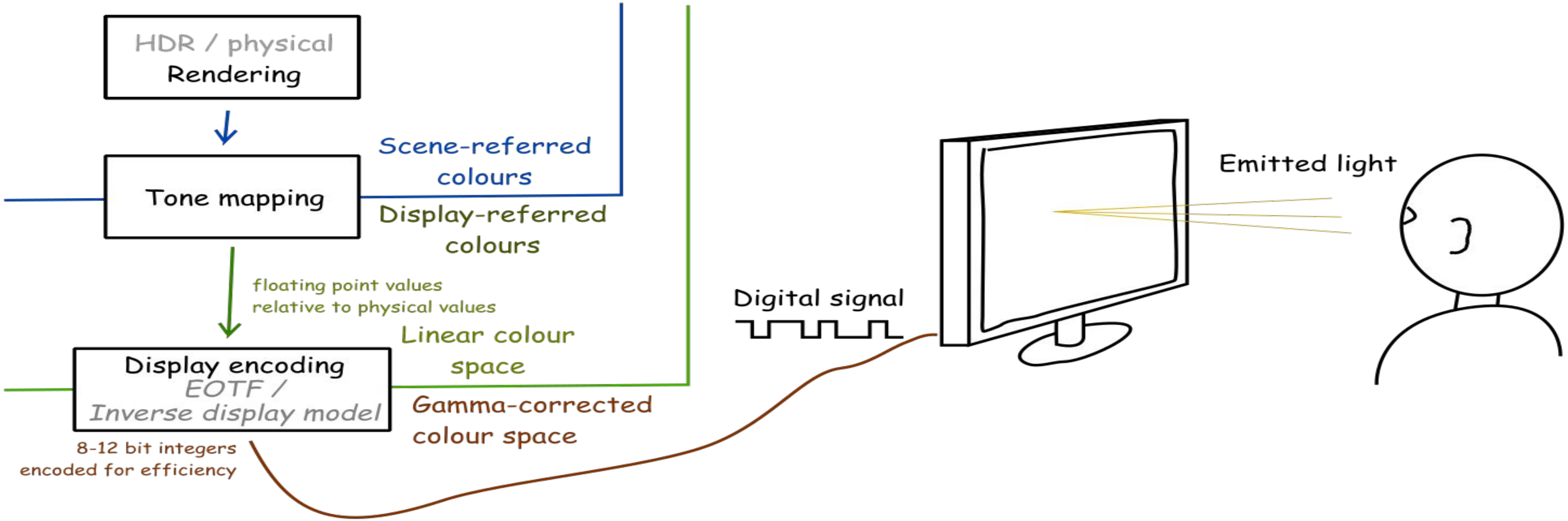
- **HDR** cameras/formats/displays attempt capture/represent/reproduce (almost) all visible colours
  - They represent scene colours and therefore we often call this representation *scene-referred*
- **SDR** cameras/formats/devices attempt to capture/represent/reproduce only colours of a standard sRGB colour gamut, mimicking the capabilities of CRTs monitors
  - They represent display colours and therefore we often call this representation *display-referred*



From rendering to display



From rendering to display



Display encoding for SDR: gamma correction

- Gamma correction is often used to encode luminance or tri-stimulus color values (RGB) in imaging systems (displays, printers, cameras, etc.)

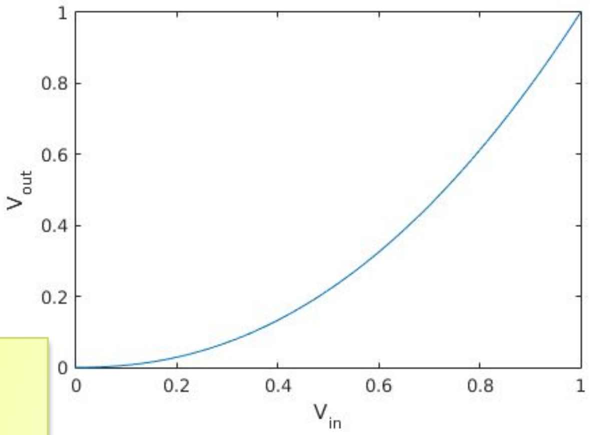
Gain

Gamma (usually =2.2)

$$\mathrm{V_{out}} = \mathrm{a} \cdot V_{in}^{\gamma}$$

(relative) Luminance
Physical signal

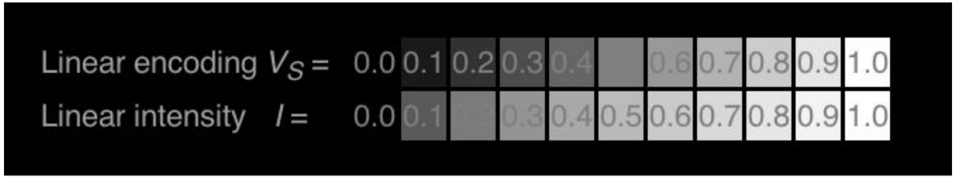
Luma
Digital signal (0-1)



Inverse: $\mathrm{V_{in}} = \left(\frac{1}{a} \cdot V_{out}\right)^{\frac{1}{\gamma}}$

Colour: the same equation applied to red, green and blue colour channels.

Why is gamma needed?



<- Pixel value (luma)
<- Luminance

- *Gamma-corrected* pixel values give a scale of brightness levels that is more perceptually uniform
- At least 12 bits (instead of 8) would be needed to encode each color channel without gamma correction
- And accidentally it was also the response of the CRT gun

Luma – gray-scale pixel value

- **Luma** - pixel brightness in *gamma corrected* units

$$L' = 0.2126R' + 0.7152G' + 0.0722B'$$

  - R', G' and B' are *gamma-corrected* colour values
  - Prime symbol denotes *gamma corrected*
  - Used in image/video coding

- Note that relative **luminance** if often approximated with

$$\begin{aligned} L &= 0.2126R + 0.7152G + 0.0722B \\ &= 0.2126(R')^{\gamma} + 0.7152(G')^{\gamma} + 0.0722(B')^{\gamma} \end{aligned}$$

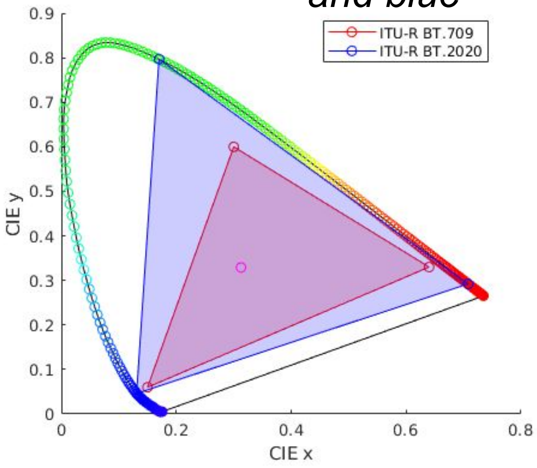
- $R, G,$ and B are *linear* colour values
- Luma and luminace are different quantities despite similar formulas

Standards for display encoding

Display type	Colour space	EOTF	Bit depth
Standard Dynamic Range	ITU-R 709	2.2 gamma / sRGB	8 to 10
High Dynamic Range	ITU-R 2020	ITU-R 2100 (PQ/HLG)	10 to 12

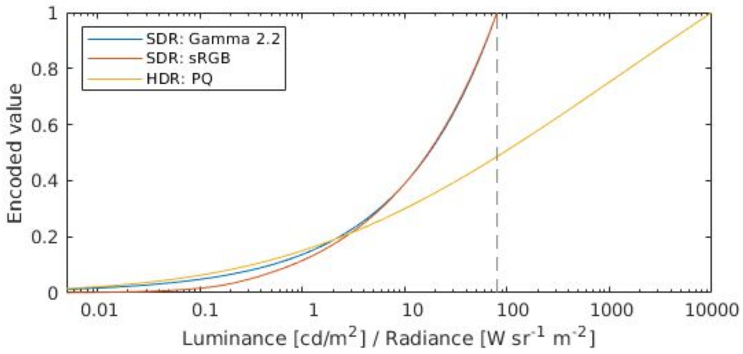
Colour space

What is the colour of "pure" red, green and blue



Electro-Optical Transfer Function

How to efficiently encode each primary colour



How to transform between RGB colour spaces?

- From ITU-R **709** RGB to ITU-R **2020** RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{R2020} = M_{XYZtoR2020} \cdot M_{R709toXYZ} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{R709}$$

- From ITU-R **2020** RGB to ITU-R **709** RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{R709} = M_{XYZtoR709} \cdot M_{R2020toXYZ} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{R2020}$$

- Where:

$$M_{R709toXYZ} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \text{ and } M_{XYZtoR709} = M_{R709toXYZ}^{-1}$$

$$M_{R2020toXYZ} = \begin{bmatrix} 0.6370 & 0.1446 & 0.1689 \\ 0.2627 & 0.6780 & 0.0593 \\ 0.0000 & 0.0281 & 1.0610 \end{bmatrix} \text{ and } M_{XYZtoR2020} = M_{R2020toXYZ}^{-1}$$

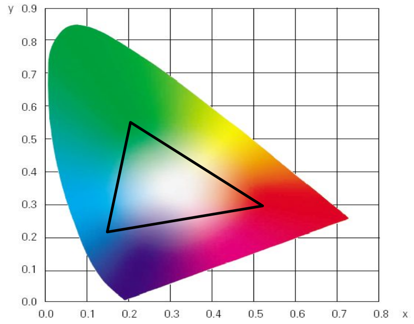
Representing colour

- We need a mechanism which allows us to represent colour in the computer by some set of numbers
  - A) preferably a small set of numbers which can be quantised to a fairly **small number of bits** each
    - Display-encoded RGB, sRGB
  - B) a set of numbers that are **easy to interpret**
    - Munsell's *artists'* scheme
    - HSV, HLS
  - C) a set of numbers in a 3D space so that the (Euclidean) distance in that space corresponds to approximately **perceptually uniform** colour differences
    - CIE Lab, CIE Luv

RGB spaces

- Most display devices that output light mix red, green and blue lights to make colour
  - televisions, CRT monitors, LCD screens

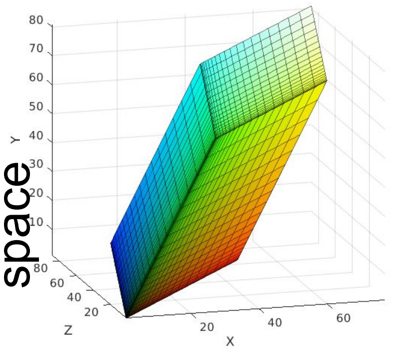
- RGB colour space
  - Can be **linear** (RGB) or **display-encoded** (R'G'B')
  - Can be **scene-referred** (HDR) or **display-referred** (SDR)
- There are multiple RGB colour spaces
  - ITU-R 709 (sRGB), ITU-R 2020, Adobe RGB, DCI-P3
    - Each using different primary colours
  - And different OETFs (gamma, PQ, etc.)
- Nominally, *RGB* space is a cube



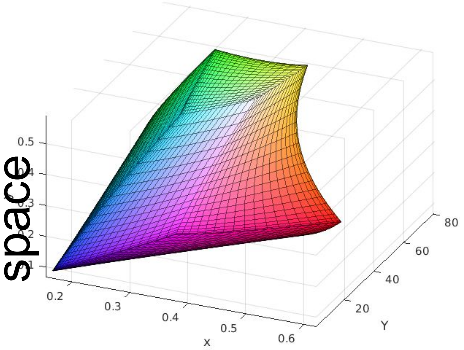
RGB in CIE *XYZ* space

- Linear RGB colour values can be transformed into CIE XYZ
  - by matrix multiplication
  - because it is a rigid transformation the colour gamut in CIE XYZ is a rotate and skewed cube
- Transformation into Yxy
  - is non-linear (non-rigid)
  - colour gamut is more complicated

RGB gamut in XYZ colour space



RGB gamut in Yxy colour space

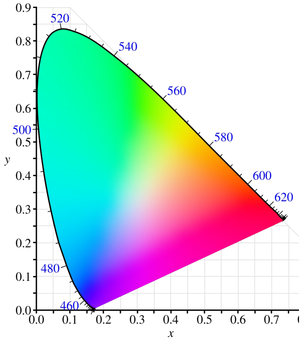


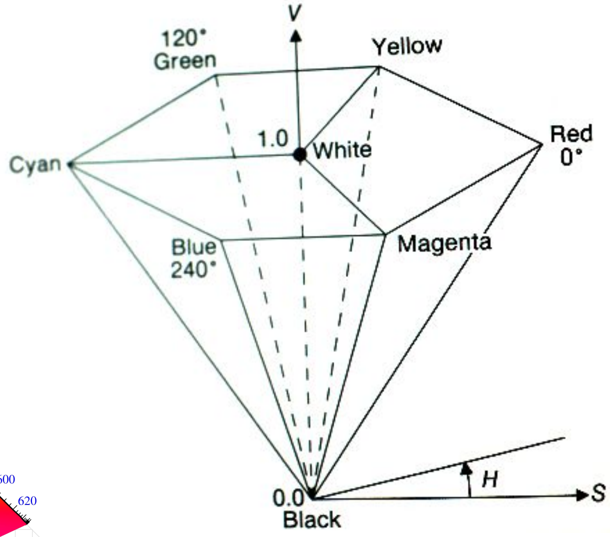
Colour spaces for user-interfaces

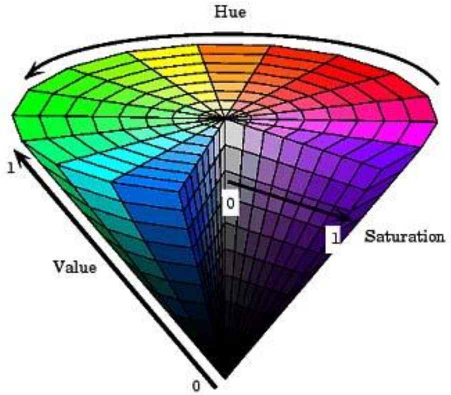
- *RGB* is are based on the physical devices which produce the coloured output
- *RGB* is difficult for humans to use for selecting colours
- HSV colour space is designed to be more intuitive:
  - hue — what is the principal colour?
  - value — how light or dark is it?
  - chroma — how vivid or dull is it?
- computer interface designers have developed basic transformations of *RGB* which resemble Munsell's human-friendly system

HSV: hue saturation value

- three axes
  - hue □ the dominant colour
  - value □ bright colours/dark colours
  - saturation □ vivid colours/dull colours



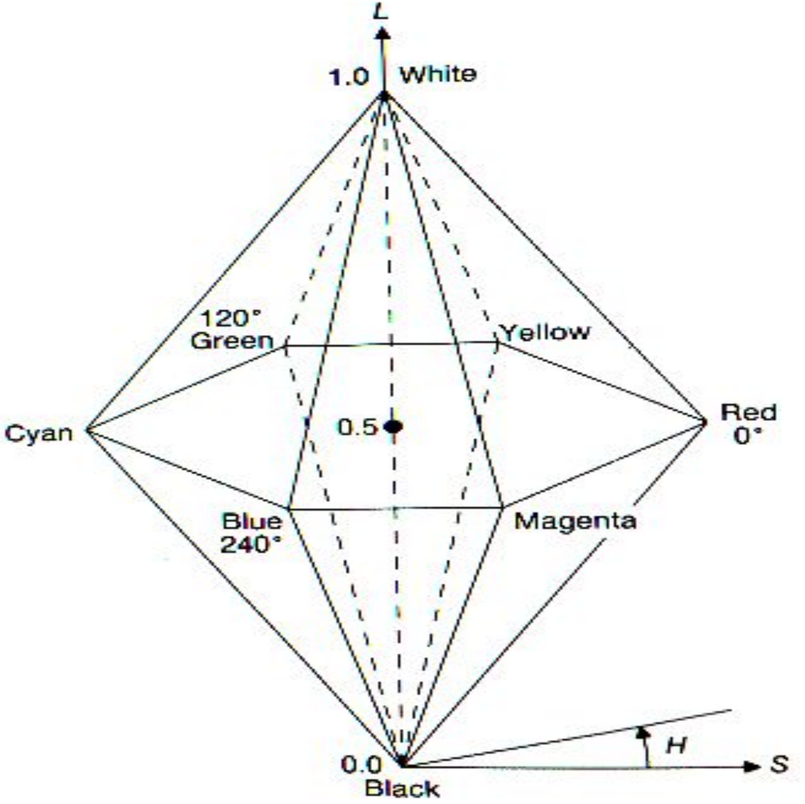




- designed by Alvy Ray Smith in 1978
- algorithm to convert *HSV* to *RGB* and back can be found in Foley et al., Figs 13.33 and 13.34

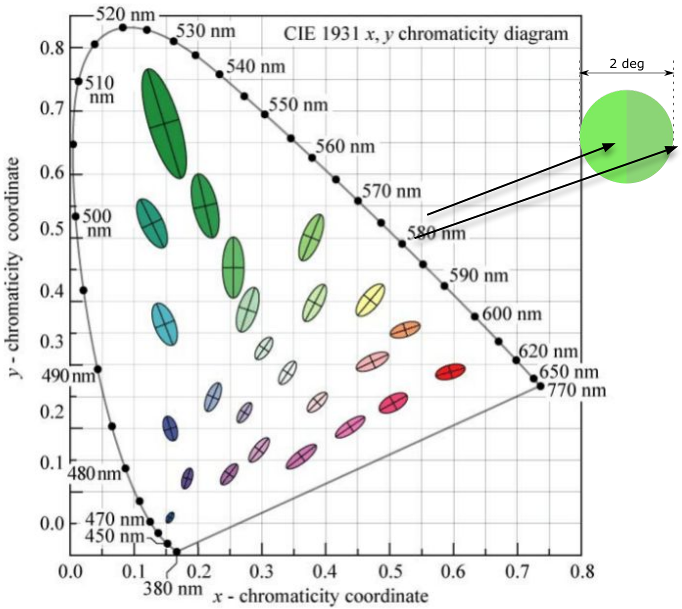
HLS: hue lightness saturation

- a simple variation of *HSV*
  - hue and saturation have same meaning
  - the term “lightness” replaces the term “value”
- designed to address the complaint that *HSV* has all pure colours having the same lightness/value as white
  - designed by Metrick in 1979
  - algorithm to convert *HLS* to *RGB* and back can be found in Foley et al., Figs 13.36 and 13.37

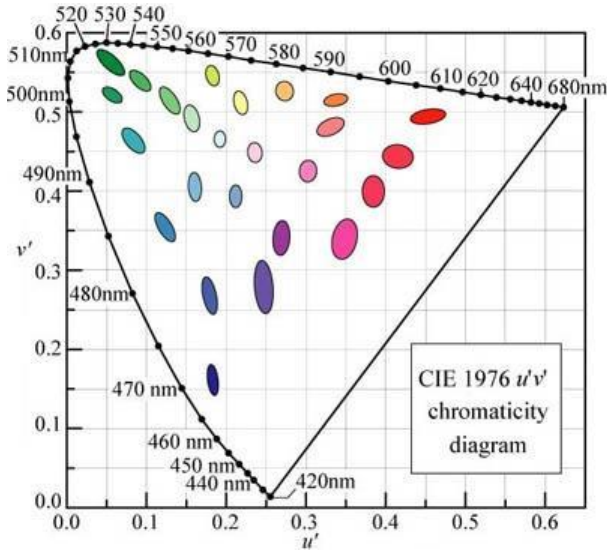


Perceptually uniformity

- MacAdam ellipses & visually indistinguishable colours



In CIE xy chromatic coordinates



In CIE u'v' chromatic coordinates

CIE $L^*u^*v^*$ and u'v'

- Approximately perceptually uniform
- u'v' chromacity

$$u' = \frac{4X}{X+15Y+3Z} = \frac{4x}{-2x+12y+3}$$
$$v' = \frac{9Y}{X+15Y+3Z} = \frac{9y}{-2x+12y+3}$$

- CIE LUV

Lightness

$$L^* = \begin{cases} \left(\frac{29}{3}\right)^3 Y/Y_n, & Y/Y_n \le \left(\frac{6}{29}\right)^3 \\ 116(Y/Y_n)^{1/3} - 16, & Y/Y_n > \left(\frac{6}{29}\right)^3 \end{cases}$$

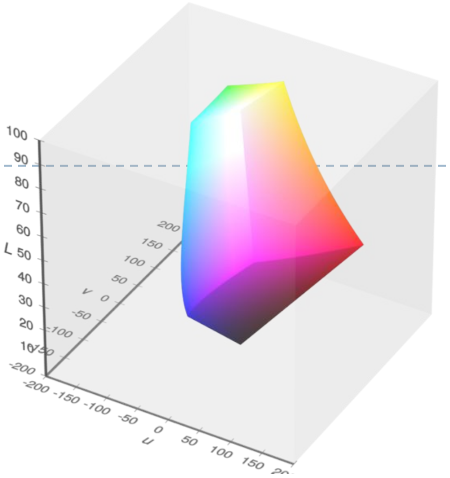
Chromacity coordinates

$$u^* = 13L^* \cdot (u' - u'_n)$$
$$v^* = 13L^* \cdot (v' - v'_n)$$

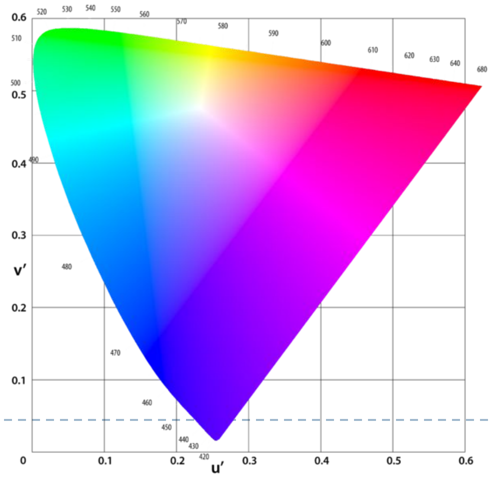
Colours less distinguishable when dark

- Hue and chroma

$$C^*_{uv} = \sqrt{(u^*)^2 + (v^*)^2}$$
$$h_{uv} = \operatorname{atan2}(v^*, u^*),$$



sRGB in CIE $L^*u^*v^*$



CIE L*a*b* colour space

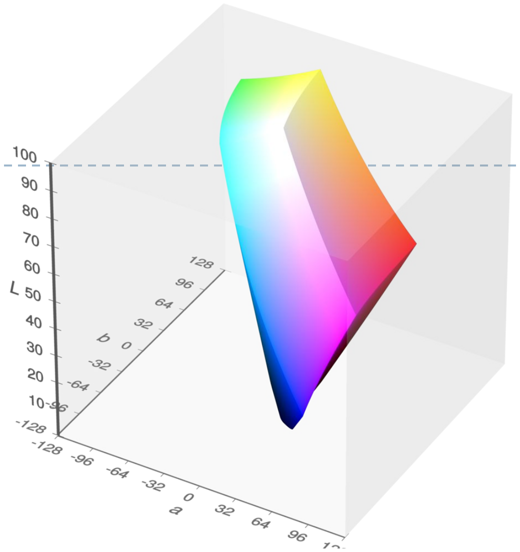
- Another approximately perceptually uniform colour space

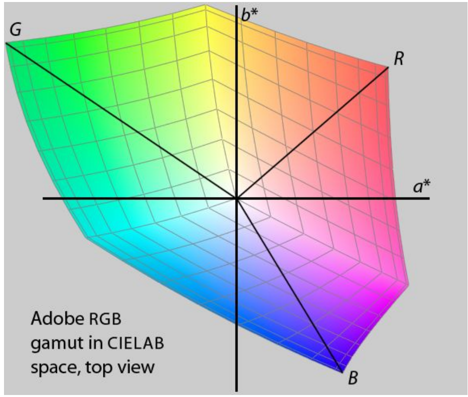
$$L^\star = 116 f\left(\frac{Y}{Y_n}\right) - 16$$
$$a^\star = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right)$$
$$b^\star = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right)$$

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{otherwise} \end{cases}$$
$$\delta = \frac{6}{29}$$

Trichromatic values of the white point, e.g.

$$X_n = 95.047,$$
$$Y_n = 100.000,$$
$$Z_n = 108.883$$





Adobe RGB gamut in CIELAB space, top view

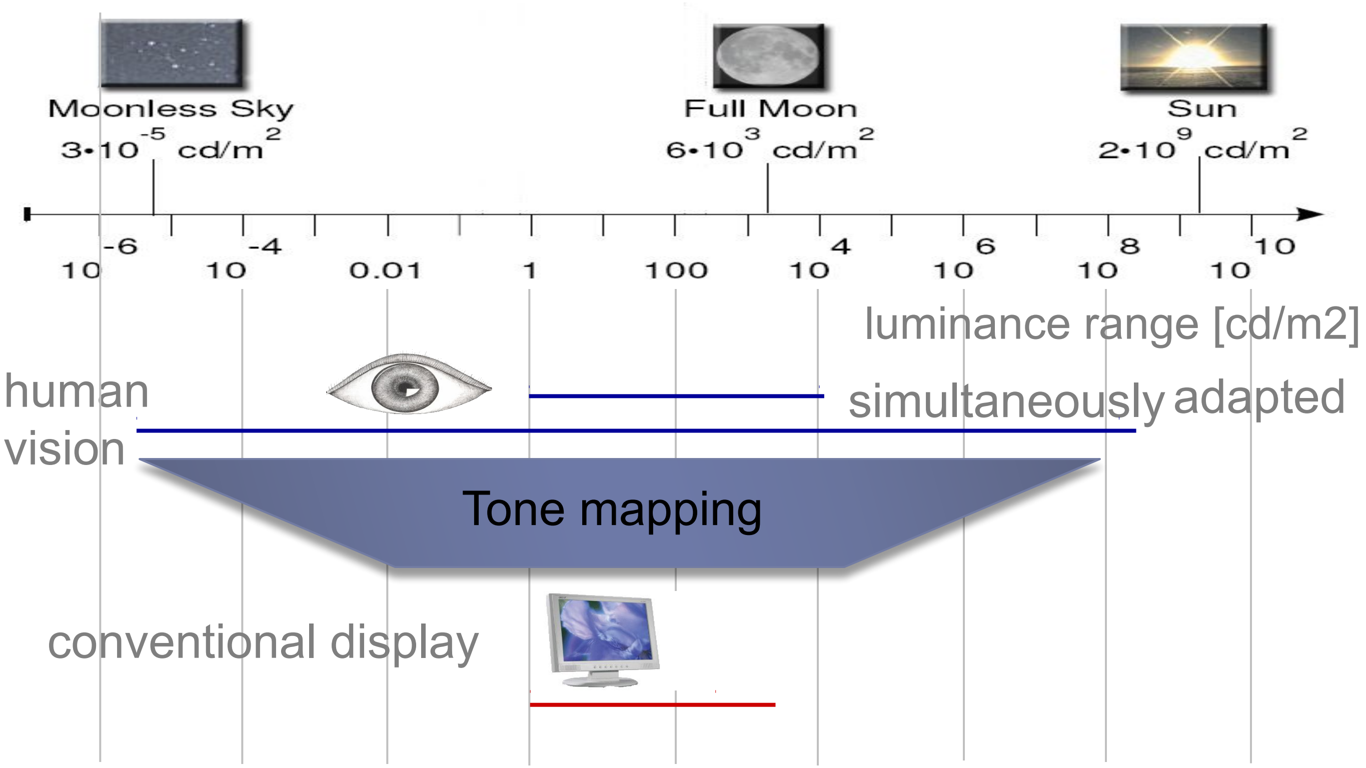
- Chroma and hue

$$C^\star = \sqrt{a^{\star\,2} + b^{\star\,2}}, \qquad h^\circ = \arctan\left(\frac{b^\star}{a^\star}\right)$$

Recap: Linear and display-encoded colour

- Linear colour spaces
  - Examples: CIE XYZ, LMS cone responses, linear RGB
  - Typically floating point numbers
  - Directly related to the measurements of light (radiance and luminance)
  - Perceptually non-uniform
  - Transformation between linear colour spaces can be expressed as a matrix multiplication
- Display-encoded and non-linear colour spaces
  - Examples: display-encoded (gamma-corrected, gamma-encoded) RGB, HVS, HLS, PQ-encoded RGB
  - Typically integers, 8-12 bits per colour channel
  - Intended for efficient encoding, easier interpretation of colour, perceptual uniformity

Tone-mapping problem



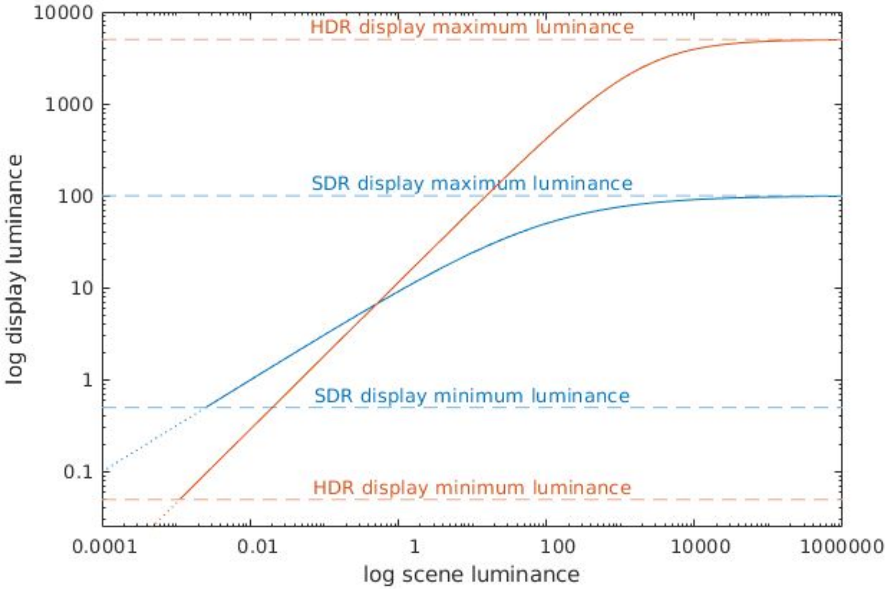
Why do we need tone mapping?

- To **reduce dynamic range**
- To **customize the look** (colour grading)
- To **simulate human vision** (for example night vision)
- To **simulate a camera** (for example motion blur)
- To **adapt** displayed images to a display and **viewing conditions**
- To make rendered images look **more realistic**
- To map from **scene- to display-referred** colours

- Different tone mapping operators achieve different combination of these goals

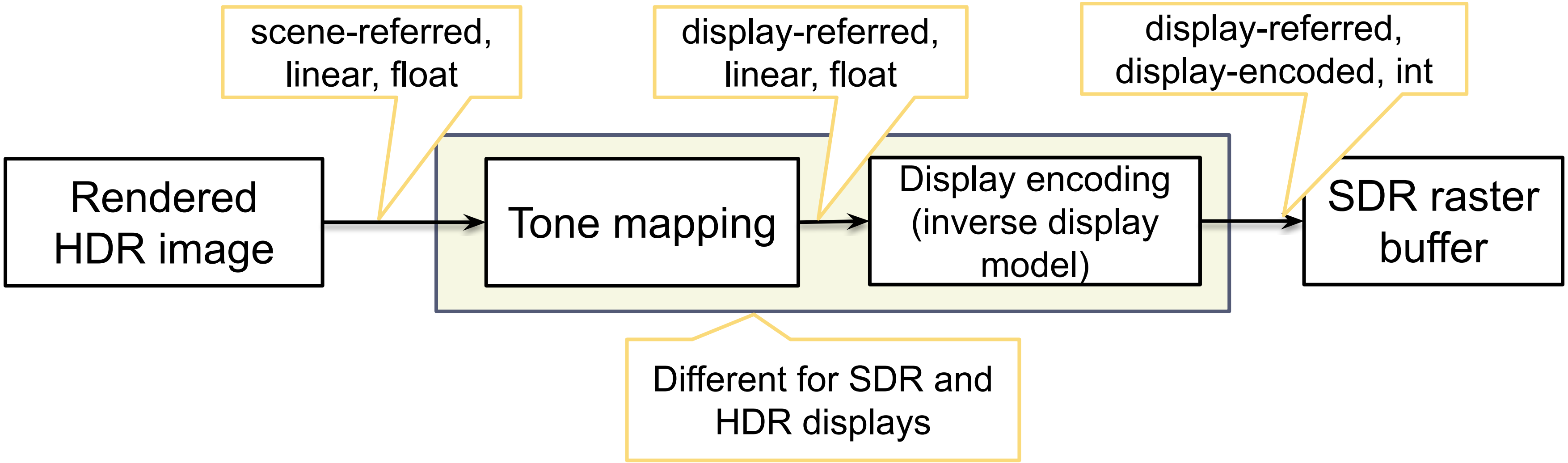
From scene- to display-referred colours

- The primary purpose of tone mapping is to transform an image from *scene-referred* to *display-referred* colours



Tone mapping and display encoding

- Tone mapping is often combined with display encoding



Basic tone-mapping and display coding

- The simplest form of tone-mapping is the exposure/brightness adjustment:

Display-referred relative red value [0;1]

$$R_d = \frac{R_s}{L_{white}}$$

Scene-referred

Scene-referred luminance of white

  - R for red, the same for green and blue

- The simplest form of display coding is the "gamma"

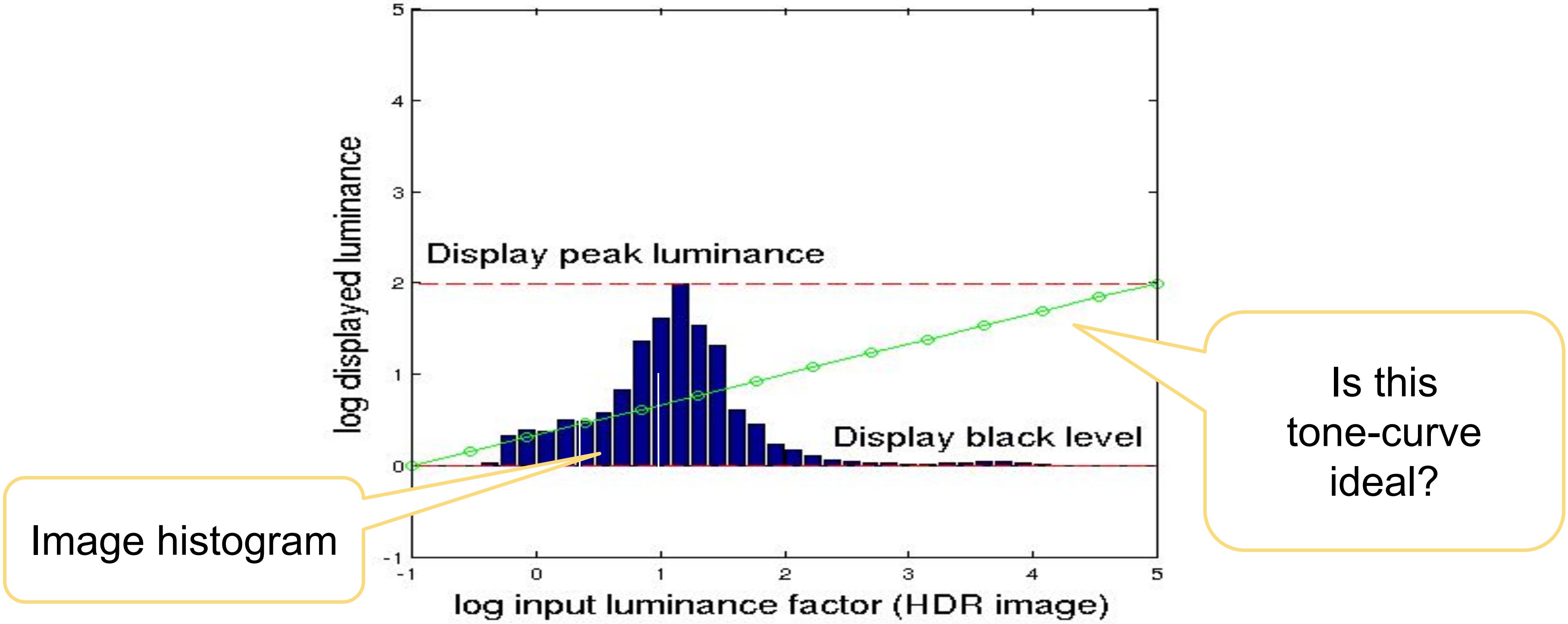
Prime (') denotes a gamma-corrected value

$$R' = (R_d)^{\frac{1}{\gamma}}$$

Typically γ=2.2

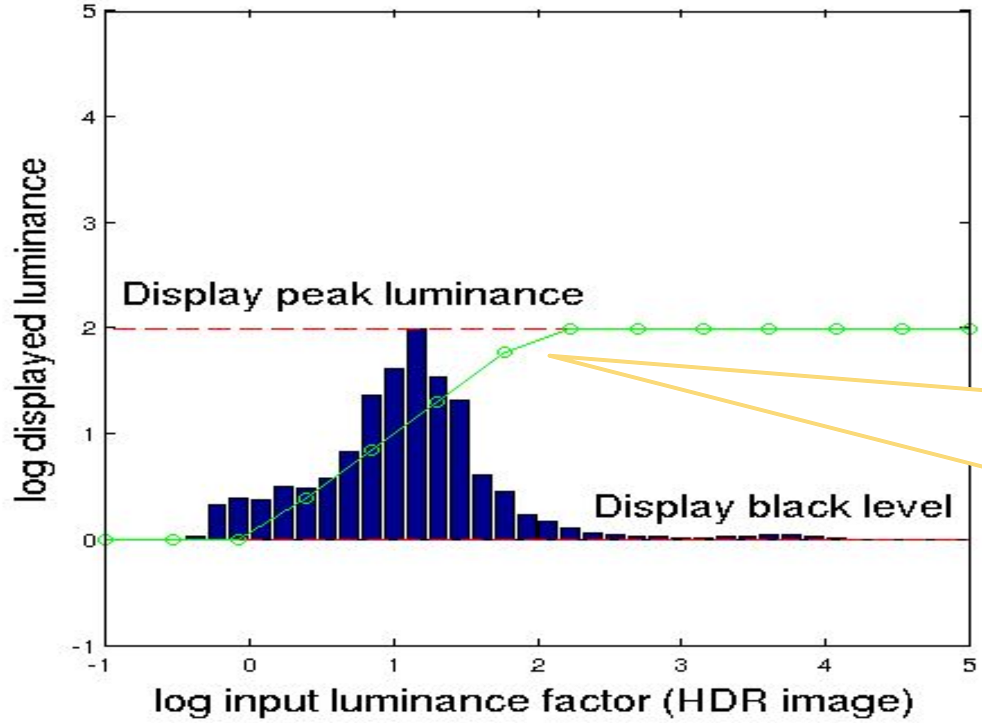
  - For SDR displays only

Tone-curve



Display peak luminance
Display black level
log displayed luminance
log input luminance factor (HDR image)
5
4
3
2
1
0
-1
-1
0
1
2
3
4
5
Image histogram
Is this tone-curve ideal?

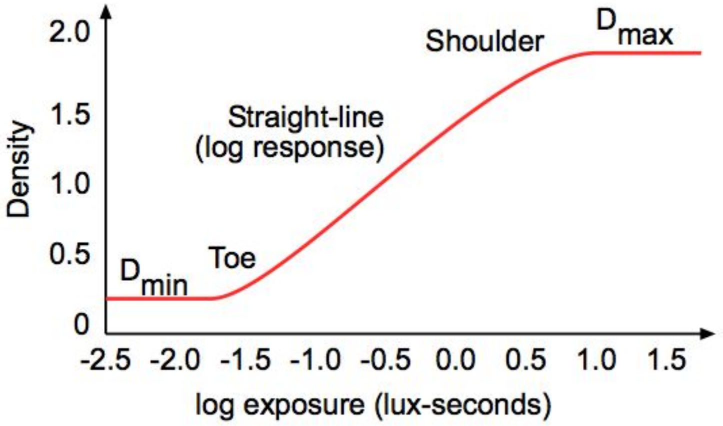
Tone-curve



Global tone-mapping is a compromise between clipping and contrast compression.

Sigmoidal tone-curves

- Very common in digital cameras
  - Mimic the response of analog film
  - Analog film has been engineered over many years to produce good tone-reproduction
- Fast to compute



Sigmoidal tone mapping

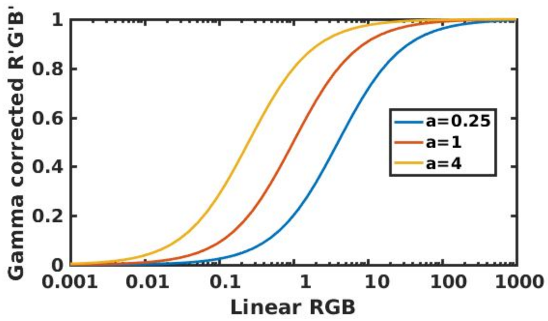
- Simple formula for a sigmoidal tone-curve:

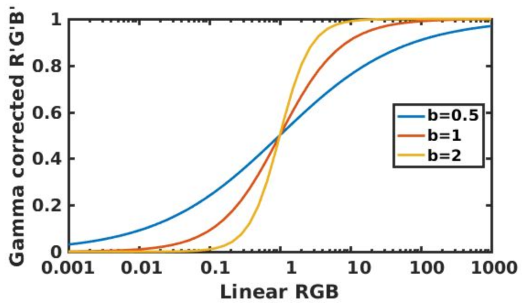
$$R'(x,y) = \frac{R(x,y)^b}{\left(\frac{L_m}{a}\right)^b + R(x,y)^b}$$

where L_m is the geometric mean (or mean of logarithms):

$$L_m = exp\left(\frac{1}{N}\sum_{(x,y)} \ln(L(x,y))\right)$$

and $L(x,y)$ is the luminance of the pixel (x,y).





Sigmoidal tone mapping example

a=0.25

a=1

a=4

b=0.5 b=1 b=2