Lecture 12:

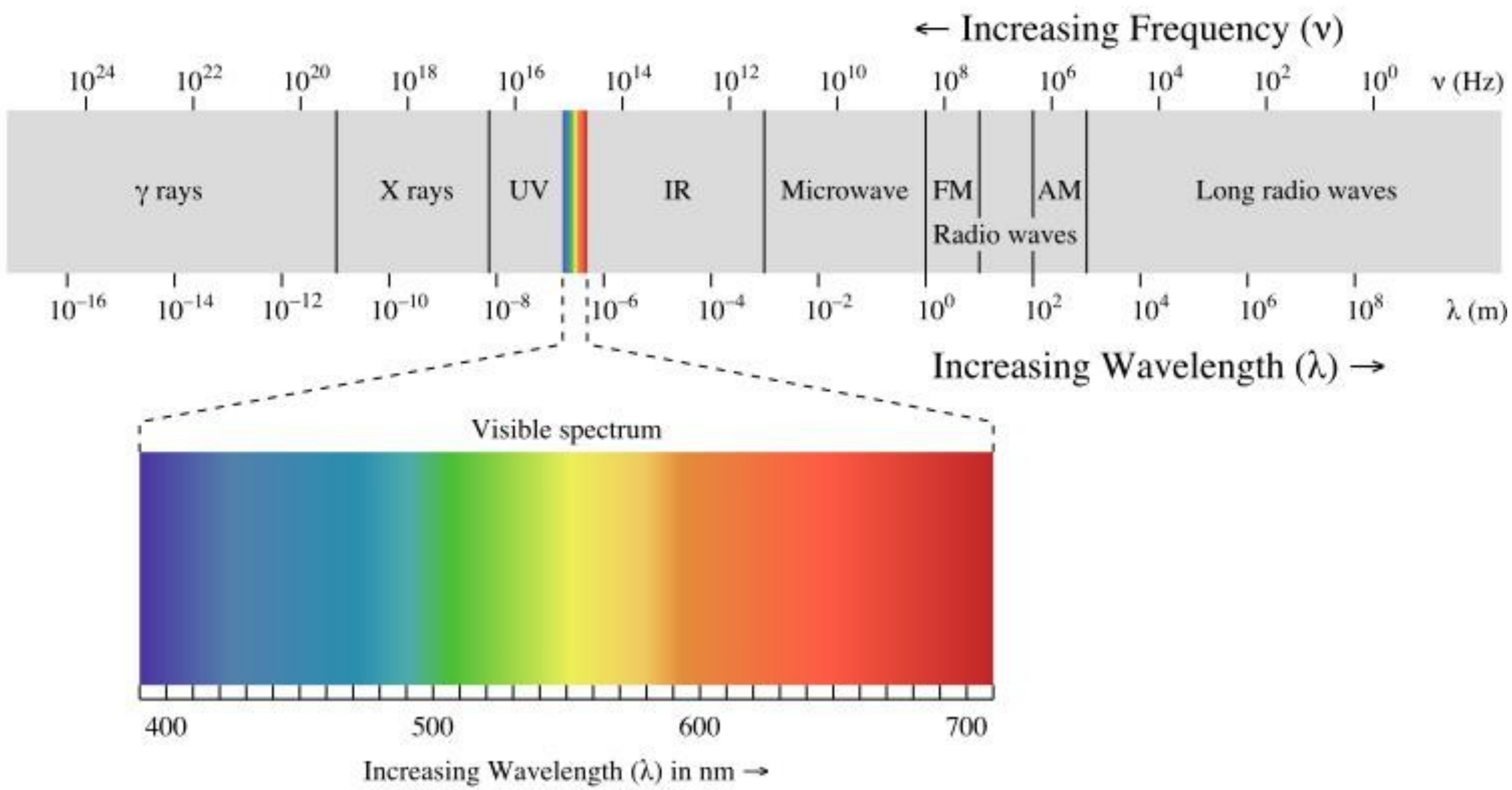
Color

Contents

1. Gamma Correction
2. Color Spaces
3. Transformations

Physics: No notion of "color"

- Light is simply a distribution of photons with different frequencies
- Specified as the "spectrum" of light
- No notion of "opposing color", "saturation", *etc*.

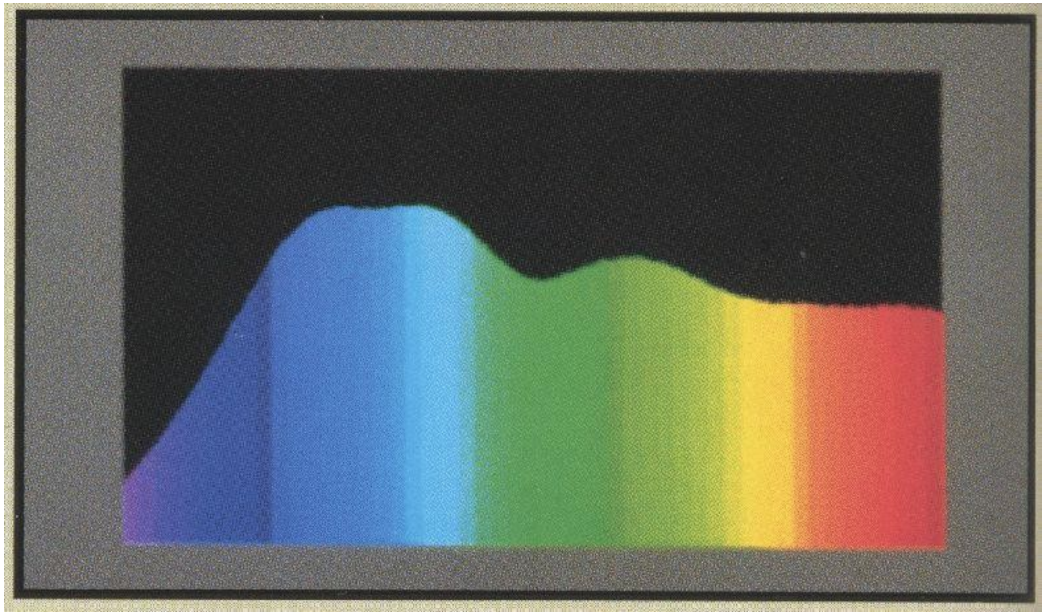


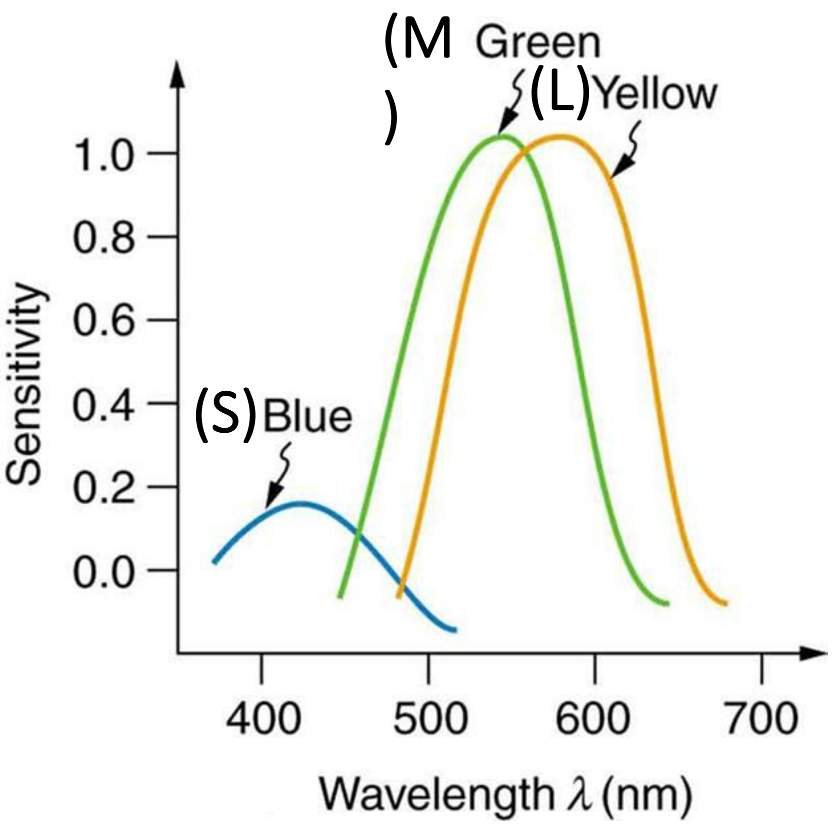
Human color perception

- Cones in retina: 3 different types
- Light spectrum is mapped to 3 different signal channels

Relative sensitivity of cones for different wavelengths

- Long (L, yellow / red), Medium (M, green), and Short (S, blue)



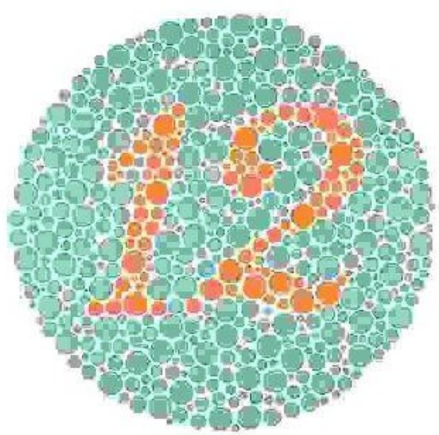


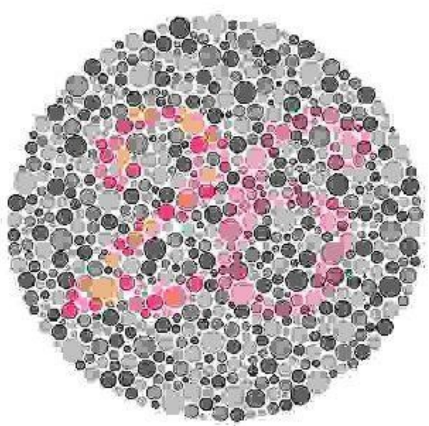
Di-chromaticity (dogs, cats)

- Yellow & blue-violet
- Green, orange, red indistinguishable

Tri-chromaticity (humans, monkeys)

- Red, green, blue
- Color-blindness (most often red-green)
  - Most often men





www.colorcube.com/illusions/clrblnd.html

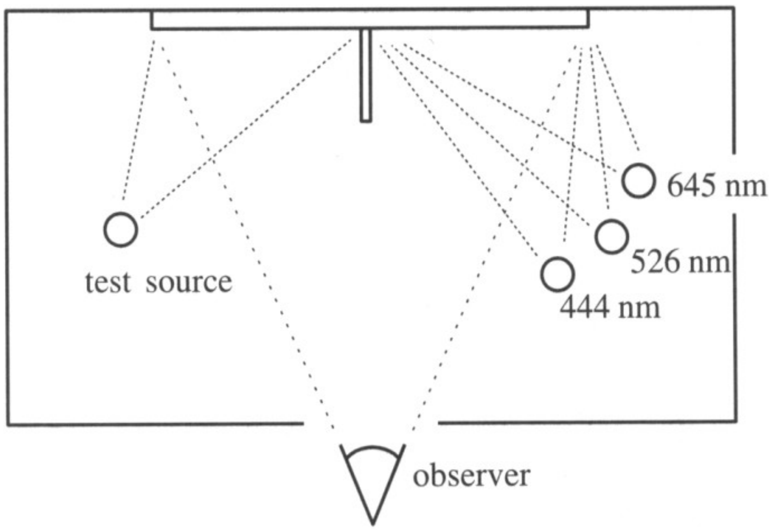
www.lam.mus.ca.us/cats/color/

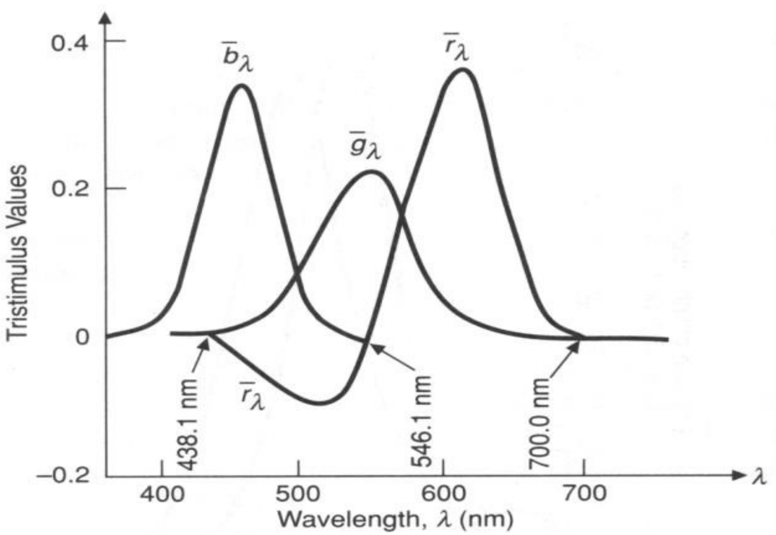
Observation

- Any color (left-hand side test source) can be matched using 3 linear independent reference primary colors (right-hand side)
- May require "negative" contribution of primary colors ⇔ positive contribution to test color
- "Matching curves" describe values for a certain set of primaries to match a mono-chromatic spectral test color of given intensity

Main results of key Color Matching Experiments

- Color perception forms a linear 3-D vector space
- Superposition holds





CIE color matching experiments

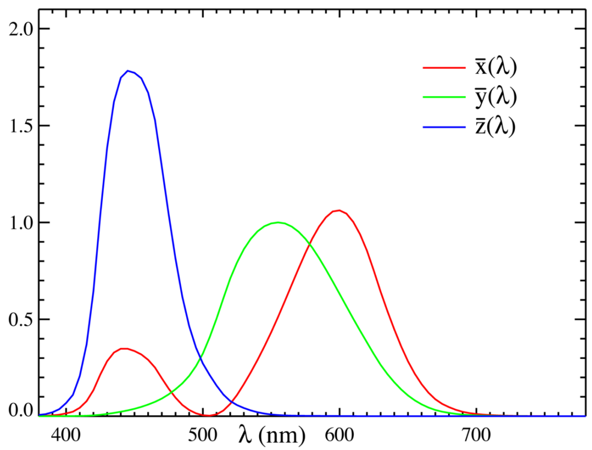
- First experiment [Guild and Wright, 1931]
  - Group of ~12 people with "normal" color vision (from London area)
  - 2-degree visual field (fovea only)
- Other experiment in 1964
  - Group of ~50 people (from different countries)
  - 10-degree visual field
  - More appropriate for larger field of view, but rarely used since similar

CIE-XYZ color space

- Transformation to a set of *virtual primaries*
  - Simple basis transform in 3D color space
- Goals:
  - Abstract from concrete primaries used in experiment
  - All matching functions should be positive
  - One primary should be roughly proportionally to light intensity (*luminosity function* $V(\lambda)$)

Standardized imaginary primaries CIE XYZ (1931)

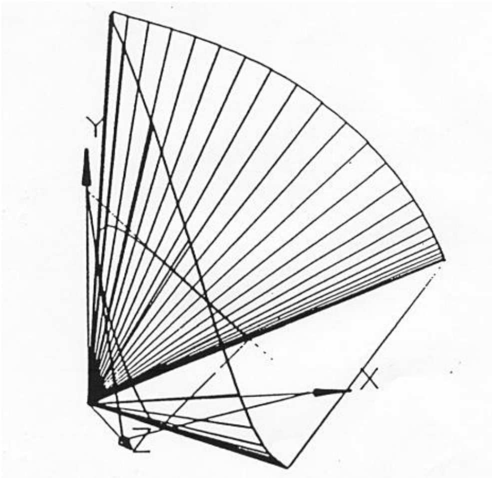
- Imaginary primaries “more saturated” than monochromatic lights
  - Could match all physically realizable color stimuli
- Defined via spectral matching for virtual CIE XYZ primaries
  - Virtual red (X), green (Y), blue (Z)
- Y is roughly equivalent to luminance
  - Shape similar to luminous efficiency function V(λ)
- Monochromatic spectral colors form a curve in 3D XYZ-space
  - Colors: combinations of monochromatic light ⇒ within the curve hull
  - Colors beyond visible limits typically ignored since not perceptible



$$X = K_m \int_{380}^{780} L(\lambda)\bar{x}(\lambda)d\lambda$$

$$Y = K_m \int_{380}^{780} L(\lambda)\bar{y}(\lambda)d\lambda$$

$$Z = K_m \int_{380}^{780} L(\lambda)\bar{z}(\lambda)d\lambda$$



CIE xy Chromaticity Diagram

Concentrate on color, not light intensity

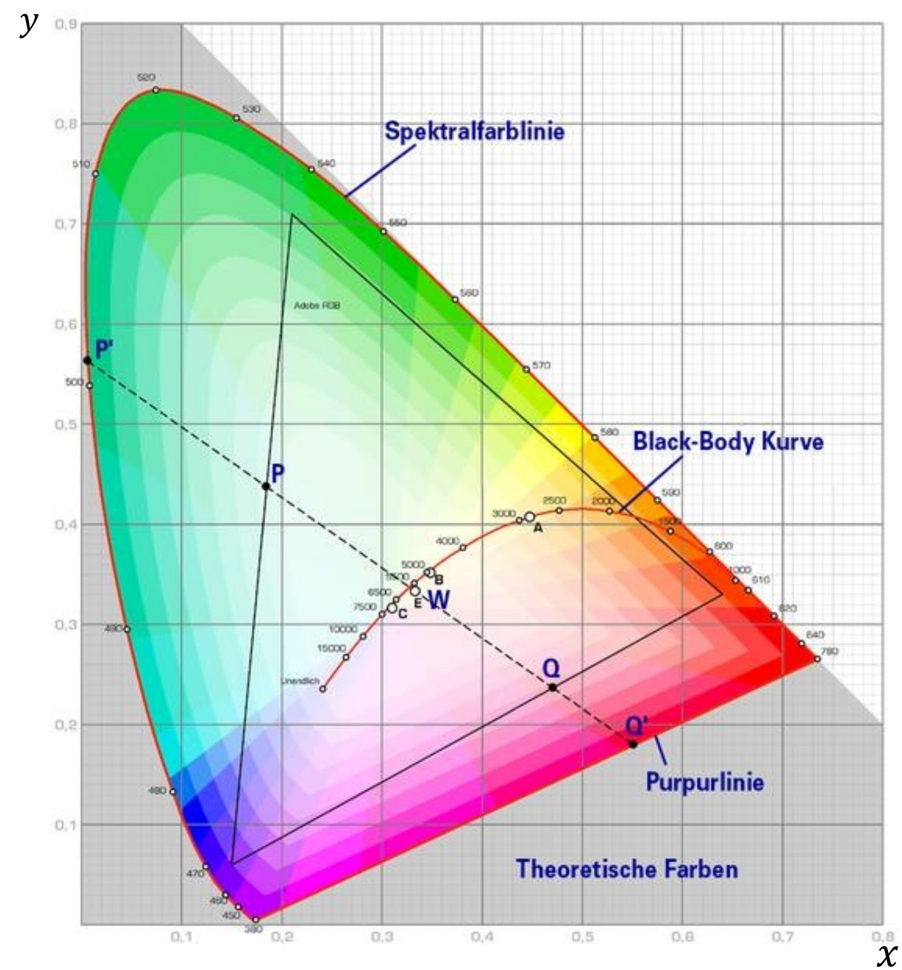
- Relative coordinates: projection on $X + Y + Z = 1$ plane (normalize)

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = 1 - x - y$$

- Chromaticity diagram
  - 2D plot over x and y
  - Points called “color locations”



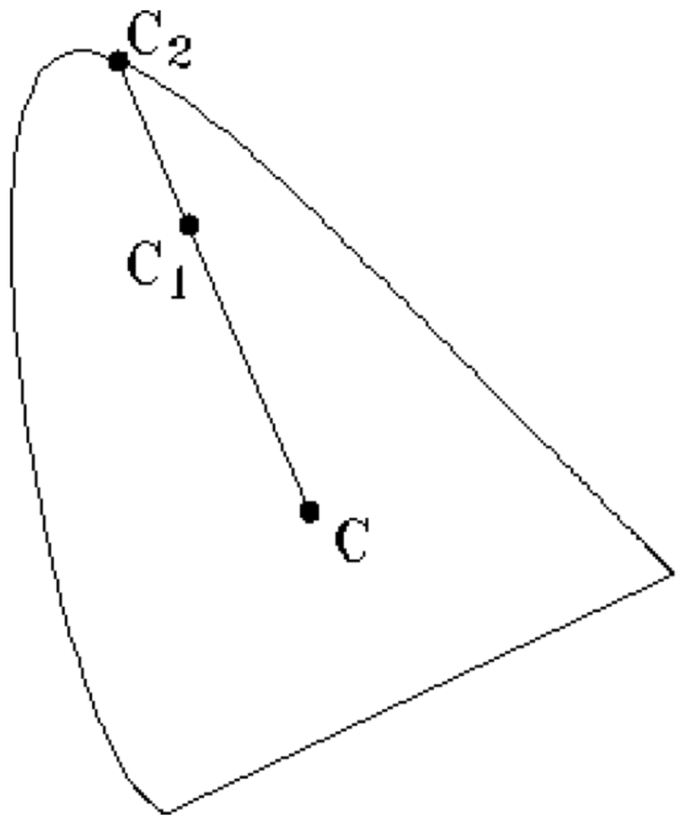
Locations of interest

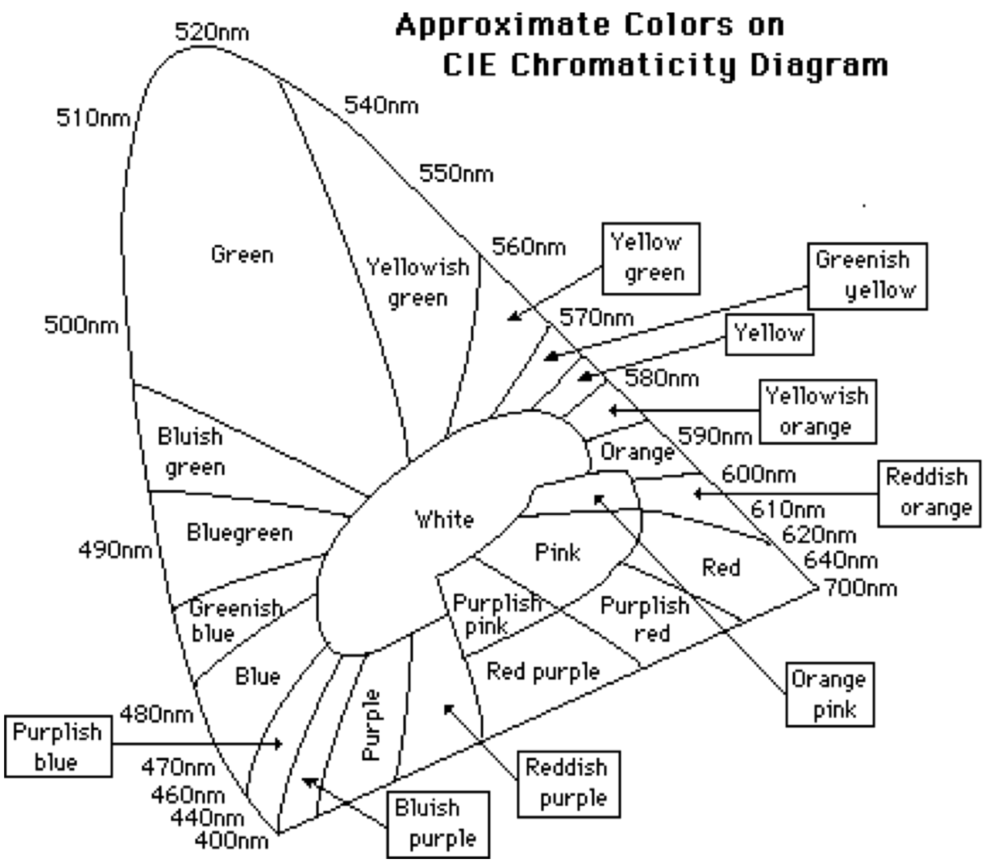
- Pure spectral colors (red line)
- Purple line: interpolate red & violet
- White point: ~(1/3, 1/3)
  - Device dependent / eye adaptation
- Black-body curve

Primaries of physical devices only allow subdomain

Specifying colors

- Saturation: relative distance between pure color and white point
- Complementary colors: on other side of white point

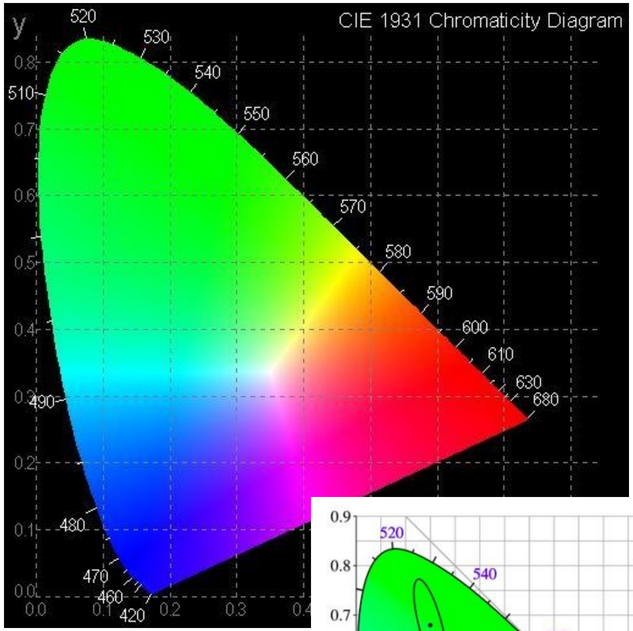




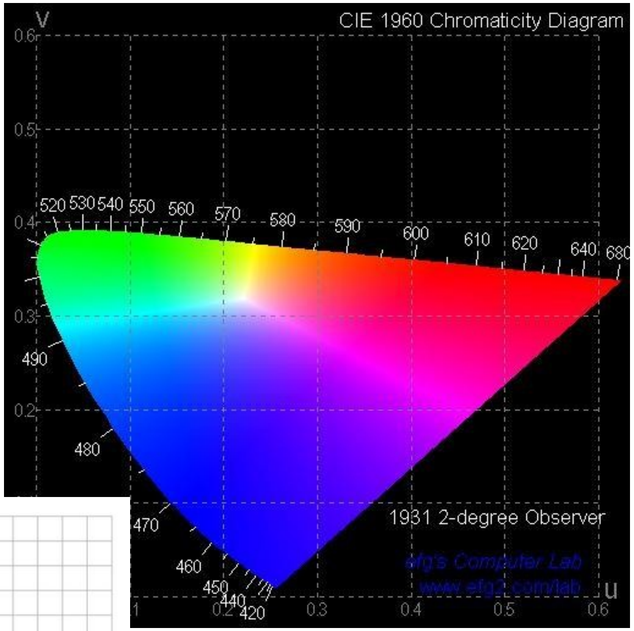
Distance threshold until perceptible color difference

- Very inhomogeneous ⇒ alternate transformations

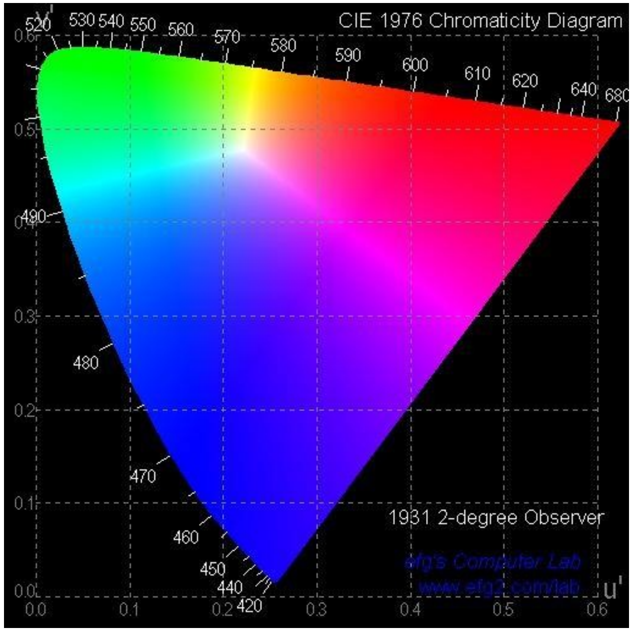
CIE-xy (1931)

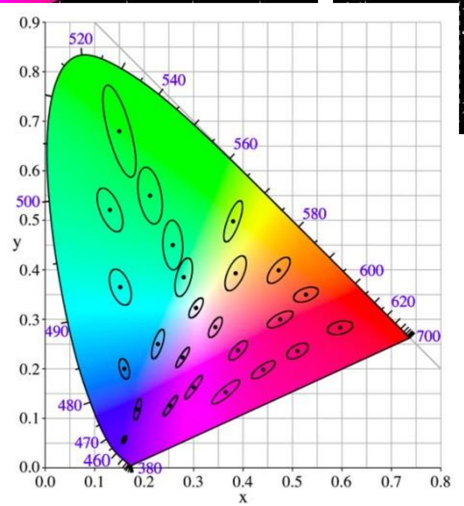


CIE-uv (1960)



CIE-u'v' (1976)





MacAdams ellipses: Same difference threshold

Gamut

- Set of representable colors

CIE XYZ gamut

- Device-independent

Device color gamut

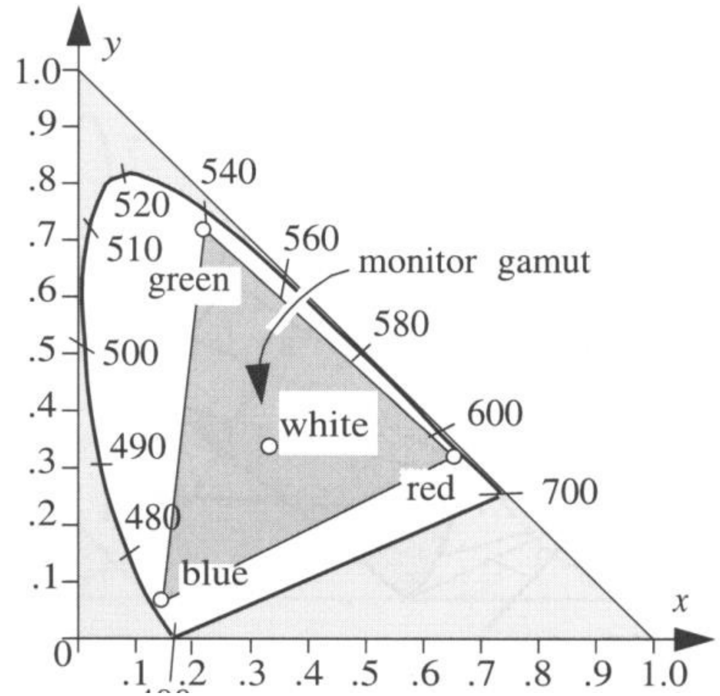
- Triangle inside color space defined by additive color blending

RGB colors

- Colors defined as linear combinations of primary colors of the device

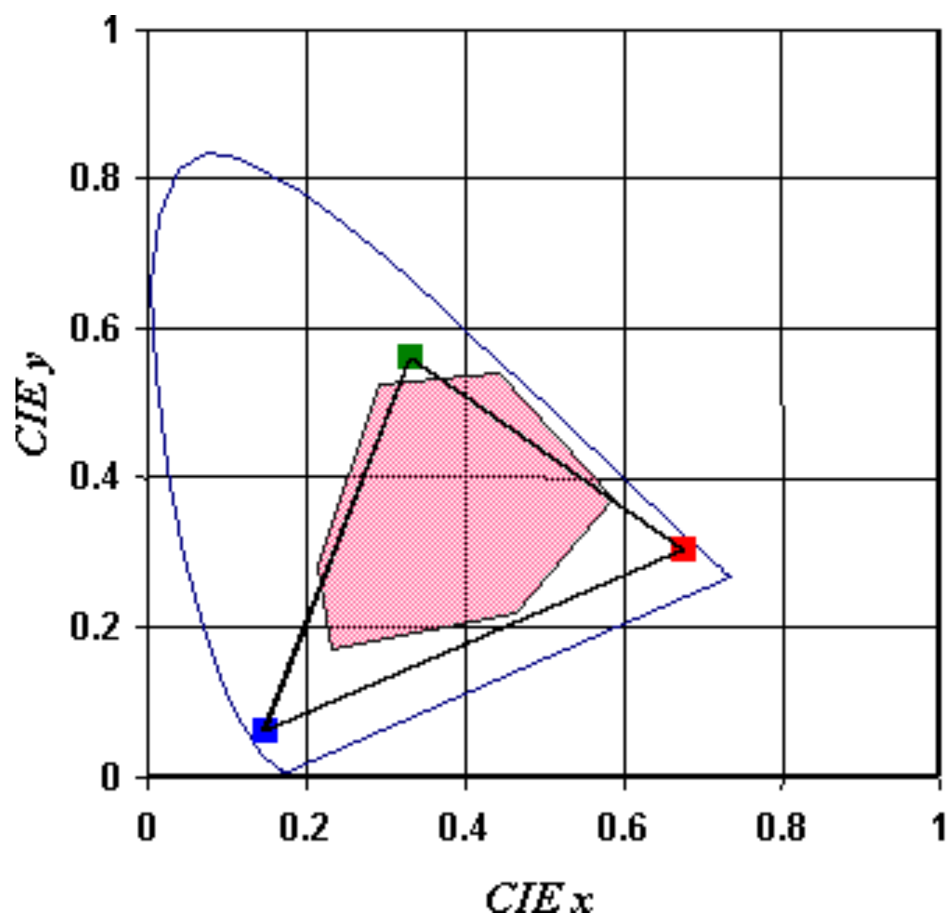
RGB space gamut

- Device (monitor/projector) dependent (!!!)
  - Choice of primaries used (lamps, LEDs)
  - Weighting of primaries (filters)
- White-point / temperature adjustment
  - Virtually moves colors within the gamut



Printer Color Gamut:

- Complex for printer due to subtractive color blending
- Complex interactions between printed color pts (mixing)
- Depends on printer colors and printer technique

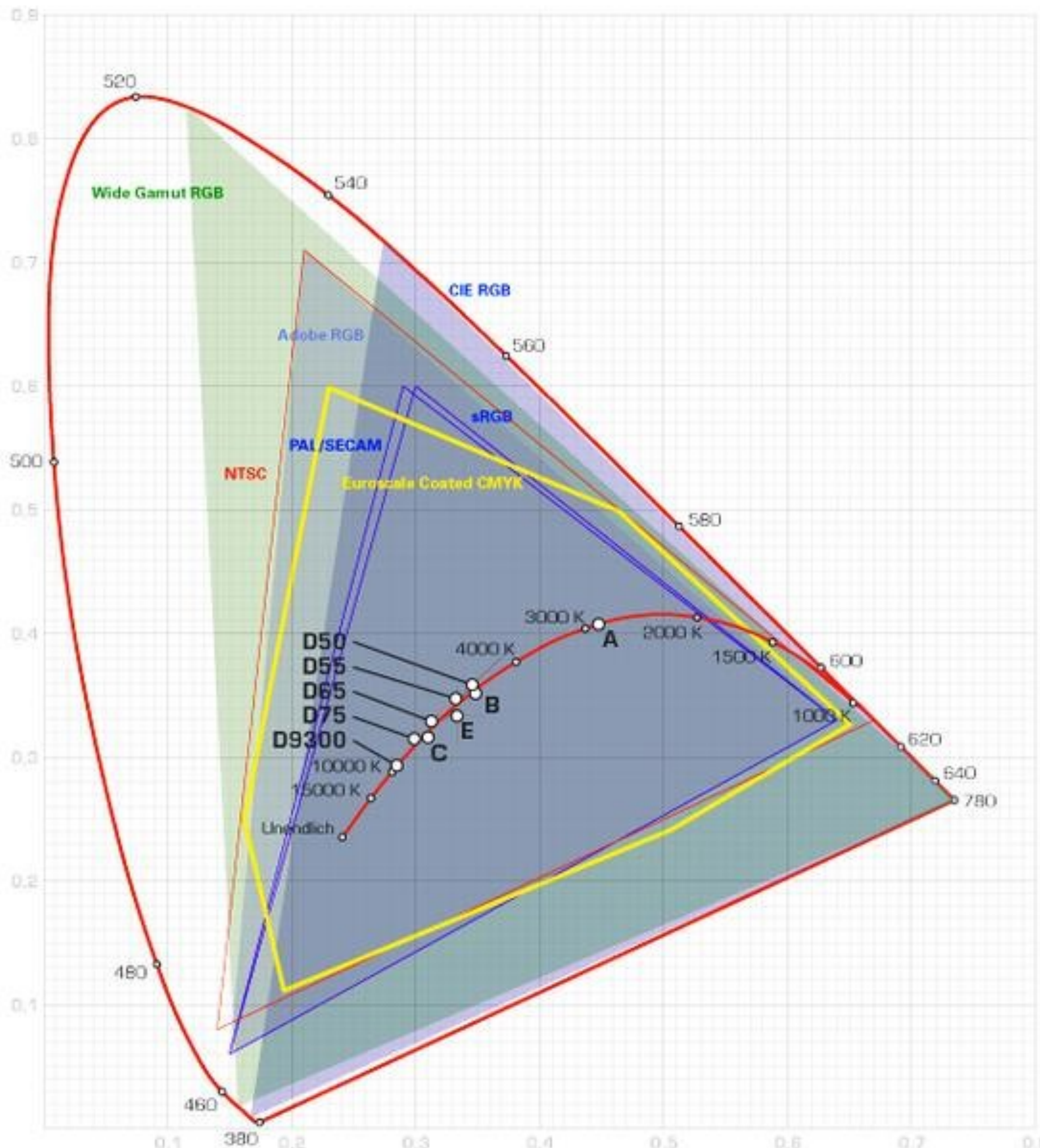


Gamut compression / mapping

- What to do if colors lay outside of the printable area?
  - Scaling, clamping, other non-linear mappings
- Each device should replace its out-of-gamut colors with the nearest approximate achievable colors
- Possible significant color distortions in a printed → scanned →displayed image

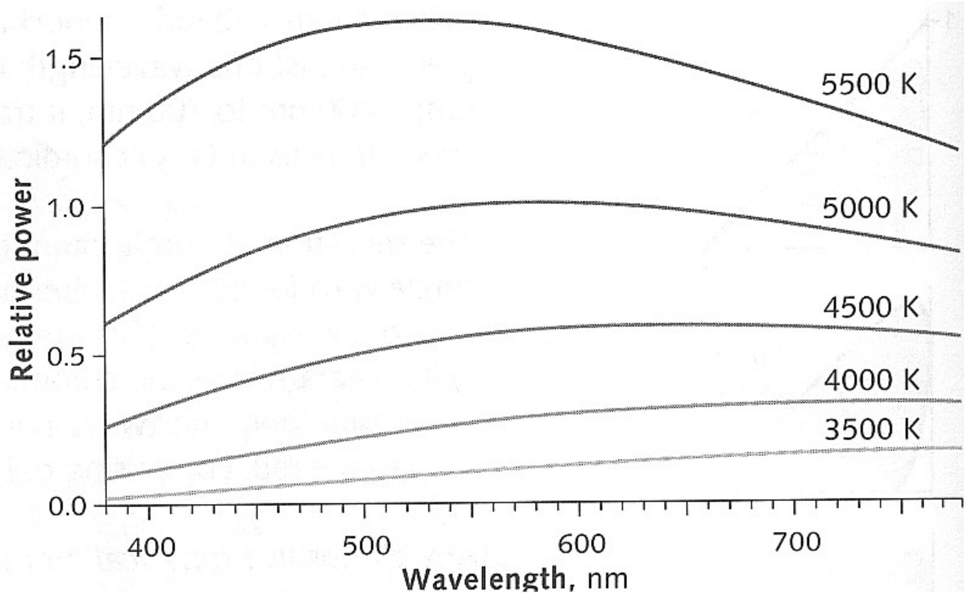
See color management later

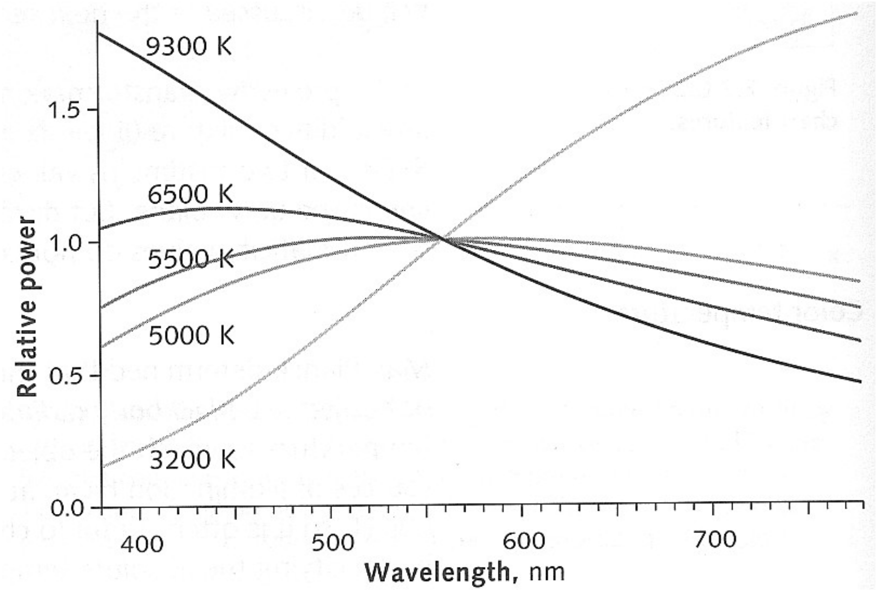
Different Color Gamuts



Theoretical light source: A black body radiator

- Perfect emitter: whole energy emitted by thermal excitation only
- Has a fixed frequency spectrum $\rho = \rho(\lambda, T)$ (Planck's law)
- Spectrum can be converted into CIE-xy color location
    - Energy shifts toward shorter wavelengths as the temperature of the black body increases
    - Normalizing the spectrum (at 550 nm)
- Allows for white point specification through temperatures





Properties of illuminant (light sources)

- Important in many applications
- Scenes look different under different (real or virtual) illumination

Set of standardized light sources

- Illuminant A – incandescent lighting conditions with a color temperature of about 2856°K
- Illuminant B – direct sunlight at about 4874°K
- Illuminant C – indirect sunlight at about 6774°K
- Illuminants D50 and D65 – different daylight conditions at color temperatures 5000°K and 6500°K, respectively

Practical use

- Spectral data of CIE standard illuminants available on the web
- Frequently used in the CG applications to compare against well-defined real-world lighting conditions

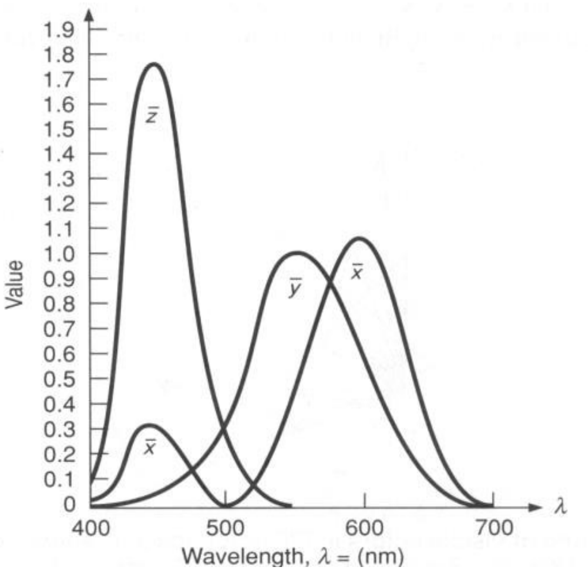
Additive color blending is a linear operation

- Can represent the operations as a matrix

Calculating primary components of a color

- Measure the spectral distribution (samples every 5-10 nm)
- Projecting from mD to 3D using sampled matching curves (loss of information)

$$\underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{3\times 1} = PL = \begin{bmatrix} x(\lambda) \\ y(\lambda) \\ z(\lambda) \end{bmatrix} L_e(\lambda) = \underbrace{\begin{bmatrix} [x_1, x_2, x_3, \dots, x_m] \\ [y_1, y_2, y_3, \dots, y_m] \\ [z_1, z_2, z_3, \dots, z_m] \end{bmatrix}}_{3\times m} \underbrace{\begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix}}_{m\times 1}$$



Transformation between color spaces:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Computing the transformation matrix M

- Given (*e.g.* from monitor manufacturer or measured)
  - Primary colors $(x_r, y_r), (x_g, y_g), (x_b, y_b)$
  - White point (x_w, y_w) for given color temperature ($R = G = B = 1$)
- Setting

$$z_r = 1 - x_r - y_r$$

$$C_r = X_r + Y_r + Z_r$$

$$x_r = \frac{X_r}{X_r + Y_r + Z_r} = \frac{X_r}{C_r} \rightarrow X_r = x_r C_r$$

  - Analogous for x_g, x_b
- R, G, B are factors modulating the primaries $(X_{rgb}, \mathrm{Y}_{rgb}, \mathrm{Z}_{rgb})$

$$M = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} = \begin{bmatrix} x_r C_r & x_g C_g & x_b C_b \\ y_r C_r & y_g C_g & y_b C_b \\ z_r C_r & z_g C_g & z_b C_b \end{bmatrix}$$

Computing the constants C_r, C_g, C_b

- Per definition the white point is given as $(R, G, B) = (1, 1, 1)$
  - $(X_w, Y_w, Z_w) = M \times (1,1,1)^T$

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} x_r C_r & x_g C_g & x_b C_b \\ y_r C_r & y_g C_g & y_b C_b \\ (1 - x_r - y_r) C_r & (1 - x_g - y_g) C_g & (1 - x_b - y_b) C_b \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

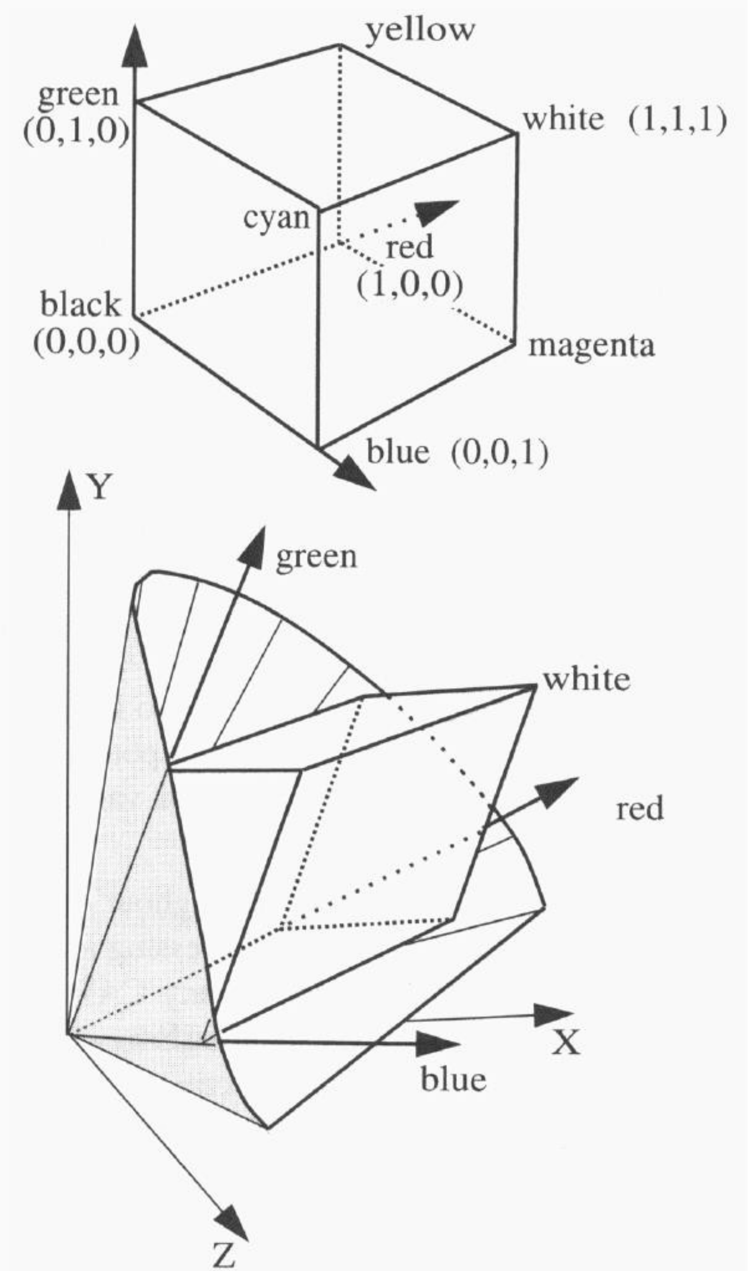
- (X_w, Y_w, Z_w) can be computed from (x_x, y_y)
  - Unspecified brightness
  - Use the normalization constant $Y_w = 1$

Can now compute conversion between color spaces of different devices by intermediate mapping to XYZ

RGB embedded in XYZ space

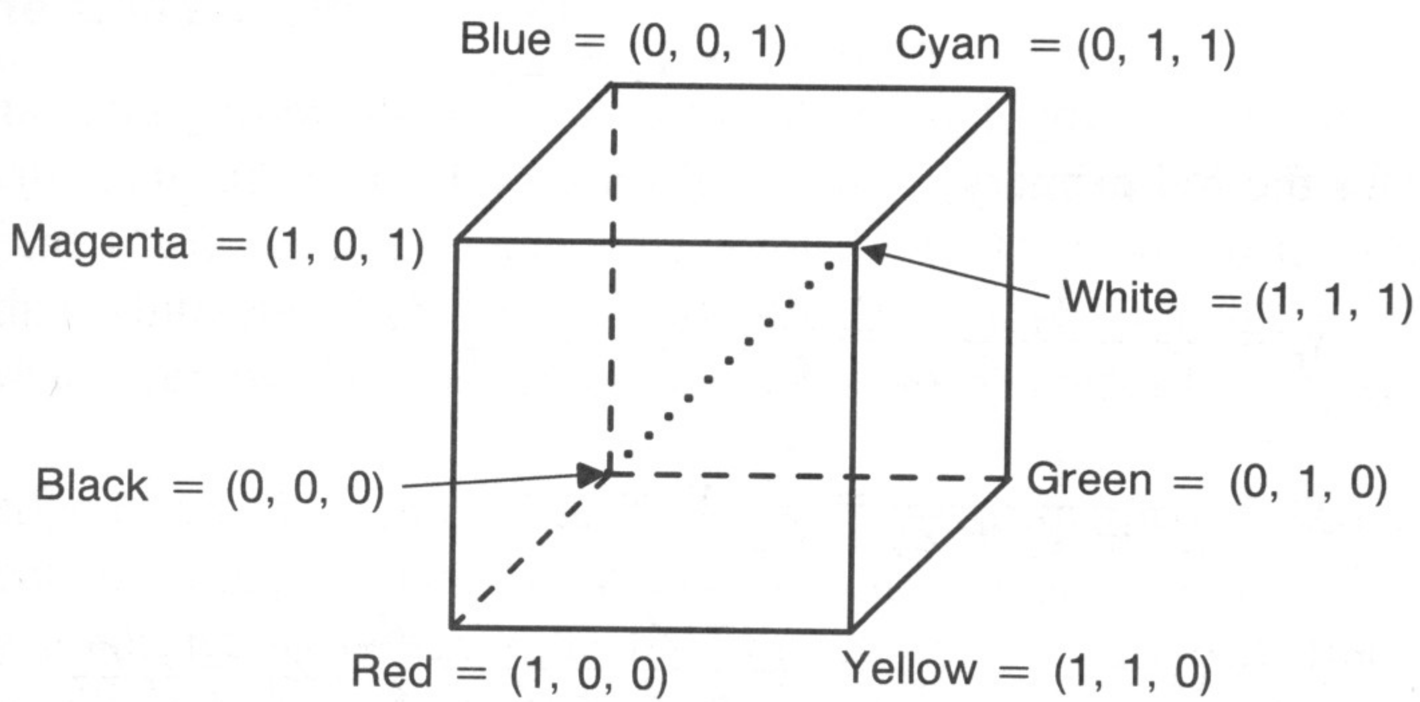
Basis change between RGB spaces

Possibly need to handle out-of-gamut colors



RGB:

- Simplest model for computer graphics
- Natural for additive devices (*e.g.* monitors)
- Device dependent (!!!)
    - **Most display applications do not correct for it!!!!**
- Many image formats don't allow primaries to be specified



Standardization of RGB

- RGB for standardized primaries and white point (and gamma)
- Specification of default CIE-XYZ values for monitors
  - Red: 0.6400, 0.3300
  - Green: 0.3000, 0.6000
  - Blue: 0.1500, 0.0600
  - White: 0.3127, 0.3290 (D65)
  - Gamma: 2.2
- Same values as HDTV and digital video (ITU-R 709)
- http://www.color.org

Utilization:

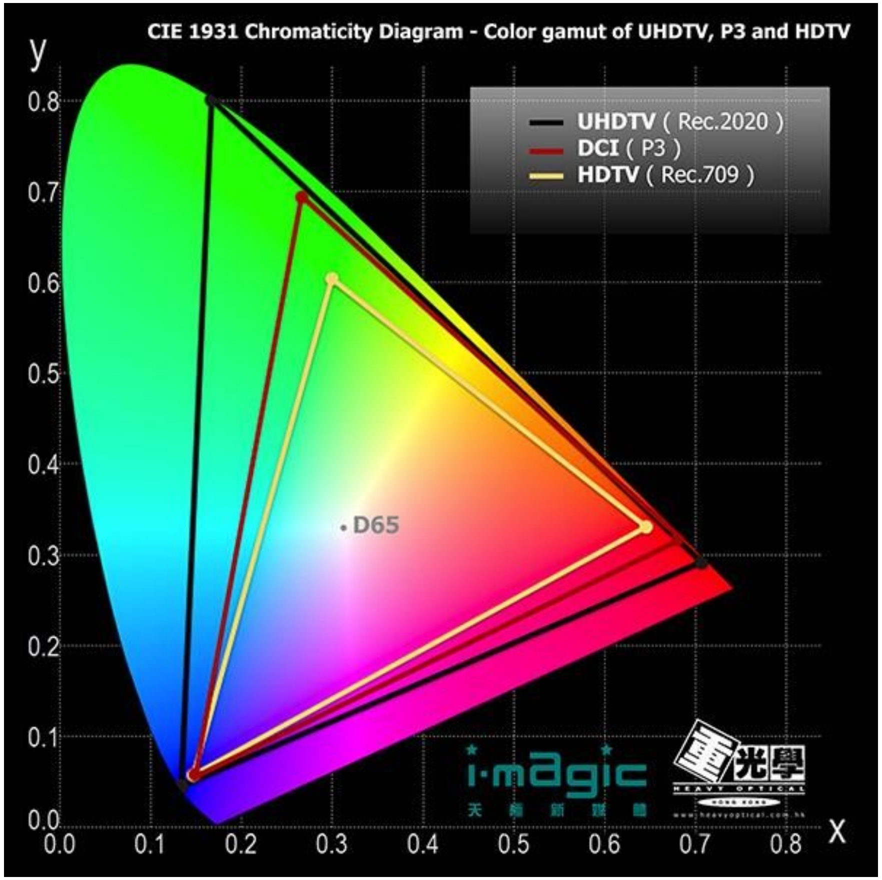
- sRGB is a standard replacement profile of International Color Consortium
- Assume all image data's without ICC profile implicitly lie in sRGB
  - Generating: ICC-Profile or writing sRGB
  - Reading/output: using ICC-Profile or assume sRGB

Standardization of 4K and 8K video format

- Resolution, frequency, digital representation
- Color gamut, gamma

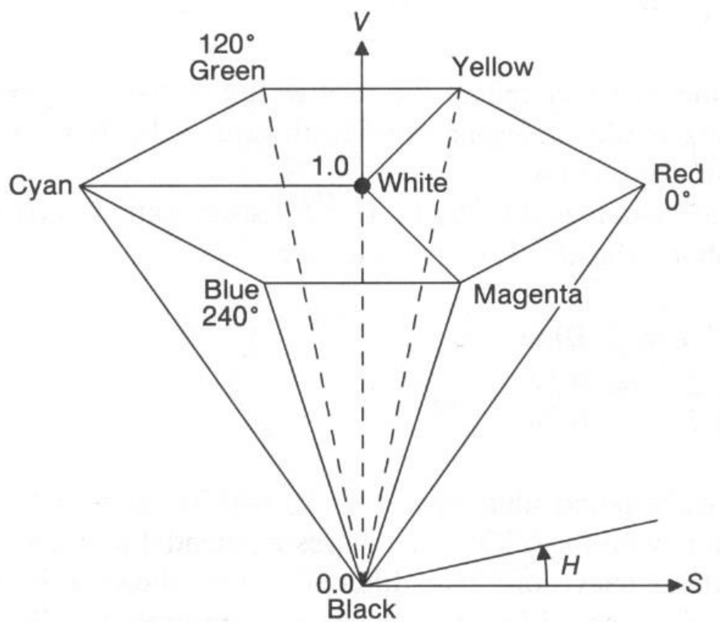
Specification of default CIE-xy values (Wide Gamut)

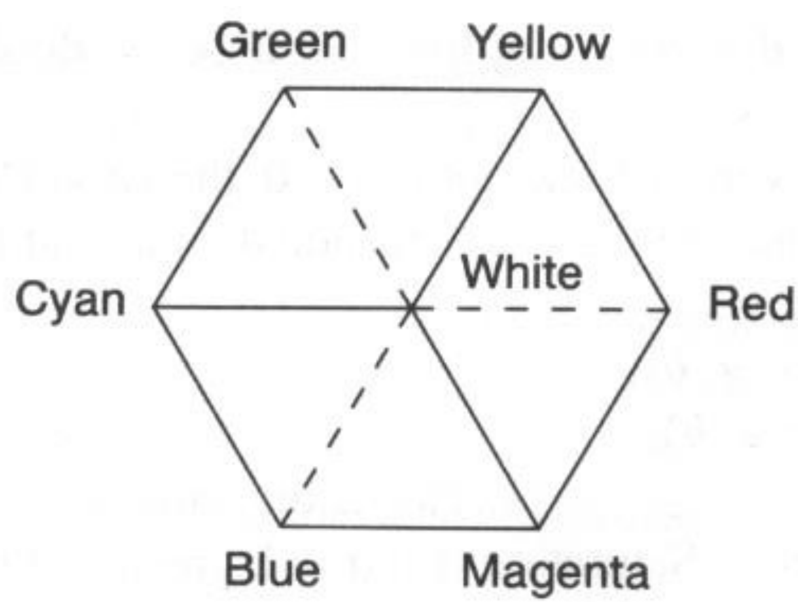
- Primaries are monochromatic!
- Red: 0.708, 0.292
- Green: 0.170, 0.797
- Blue: 0.131, 0.046
- White: 0.3127, 0.3290 (D65)
- Gamma depending on bit-depth



HSV / HSB (Hue, Saturation, Value / Brightness)

- Motivated from artistic use and intuitive color definition (vs. RGB)
  - H is equivalent to tone
  - S is equivalent to saturation (H undefined for S == 0)
  - V / B is equivalent to the gray value
- Pure tones for S == 1 and V == 1
- Intuitive model for color blending
- Builds on RGB



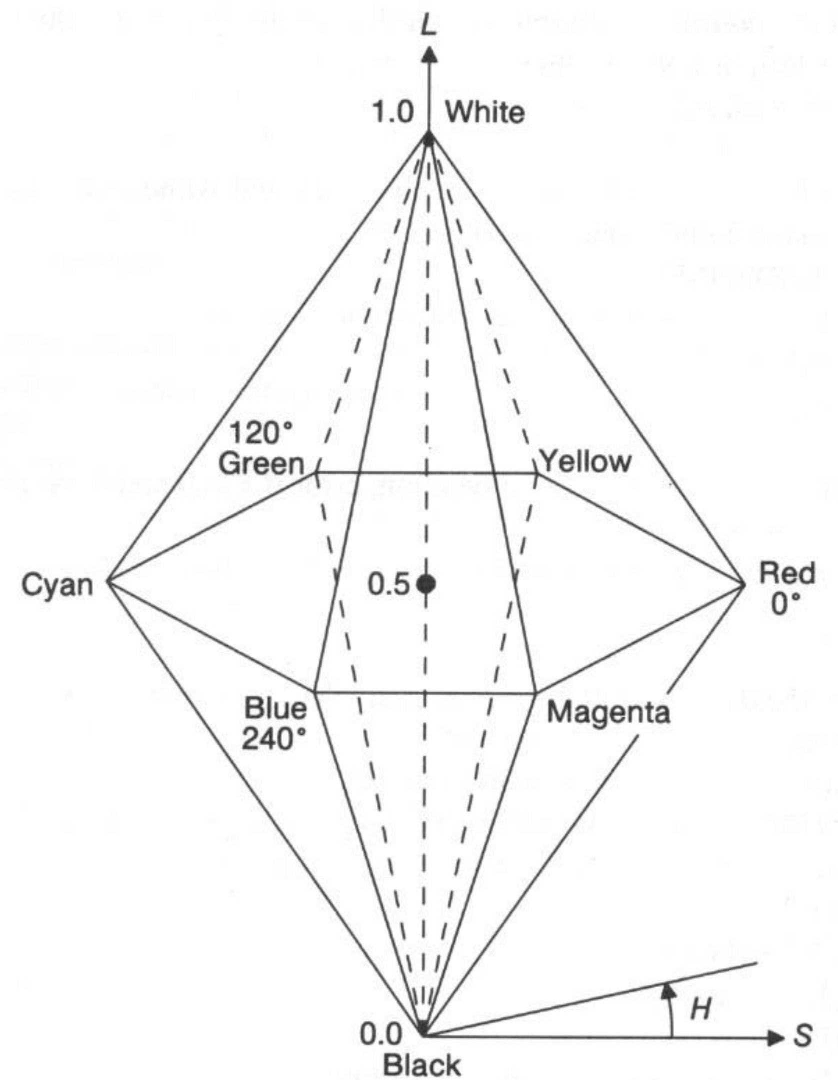


HLS (Hue, Lightness, Saturation)

- Similar to HSV / HSB
- Slightly less intuitive

Many other color models

- TekHVC
  - Developed by Tektronix
  - Perceptually uniform color space
- Video-processing
  - Y´, B-Y, R-Y
  - Y´IQ
  - Y´PrPb
  - Y´CrCb
- Non-linear color spaces



Interpolation (shading, anti-aliasing, blending)

- RGB: 0.5 red + 0.5 green = dark yellow:

  $0.5 * (1, 0, 0) + 0.5 * (0, 1, 0) = (0.5, 0.5, 0)$

- HSV: 0.5 red + 0.5 green = pure yellow:

  $0.5 * (0°, 1, 1) + 0.5 * (120°, 1, 1) = (60°, 1, 1)$

Interpretation

- Interpolation in RGB
  - Physical interpretation: linear mapping → interpolation in XYZ space
- Interpolation in HSV
  - Intuitive color interpretation: “yellow lies between red and green”

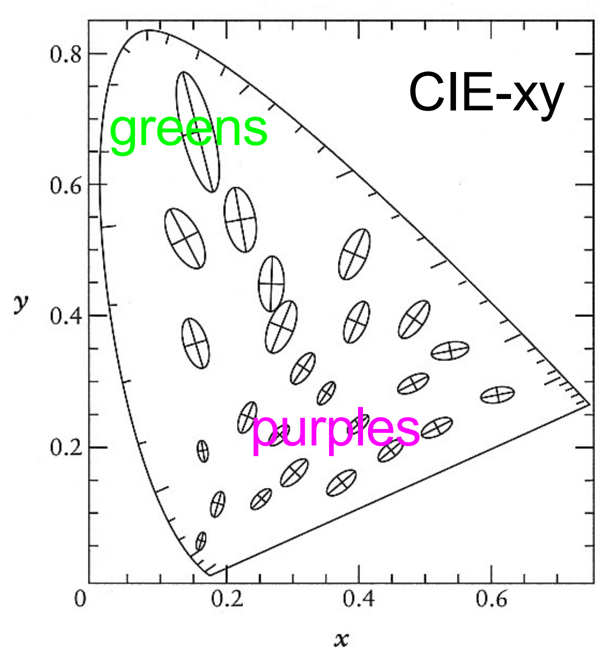
CIE-XYZ is perceptually non-uniform

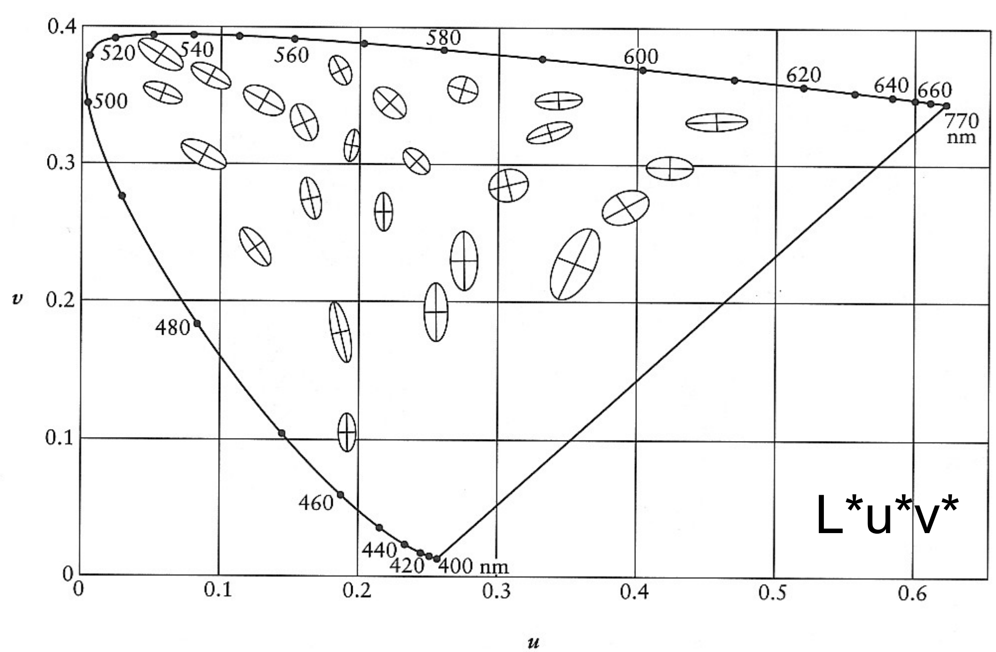
- Same perceived differences lead to very inhomogeneous differences of xy (purples tightly packed, greens stretched out)

L*u*v* / L*a*b* are device-independent color spaces

Computing difference between colors

- Transform colors to uniform color space (similarly to gamma)
- Measure color difference there





Transformation:

- Converting to XYZ (Y incidental luminance)
- Non-linear transformation on Y (Y_n is Y of the white point)

$$L^* = \begin{cases} {}^{Y}/_{Y_n} \geq 0.008856: & 116\left({}^{Y}/_{Y_n}\right)^{1/3} - 16 \\ {}^{Y}/_{Y_n} < 0.008856: & 903.3\left({}^{Y}/_{Y_n}\right) \end{cases}$$

$$L^* \in \{0, \ldots, 100\}$$

- Transformation of color differences

$$u' = {}^{4X}/_{X + 15Y + 3Z} \qquad a^* = 500L * \left[f\left({}^{X}/_{X_n}\right) - f\left({}^{Y}/_{Y_n}\right)\right]$$

$$v' = {}^{9Y}/_{X + 15Y + 3Z} \qquad b^* = 500L * \left[f\left({}^{Y}/_{Y_n}\right) - f\left({}^{Z}/_{Z_n}\right)\right]$$

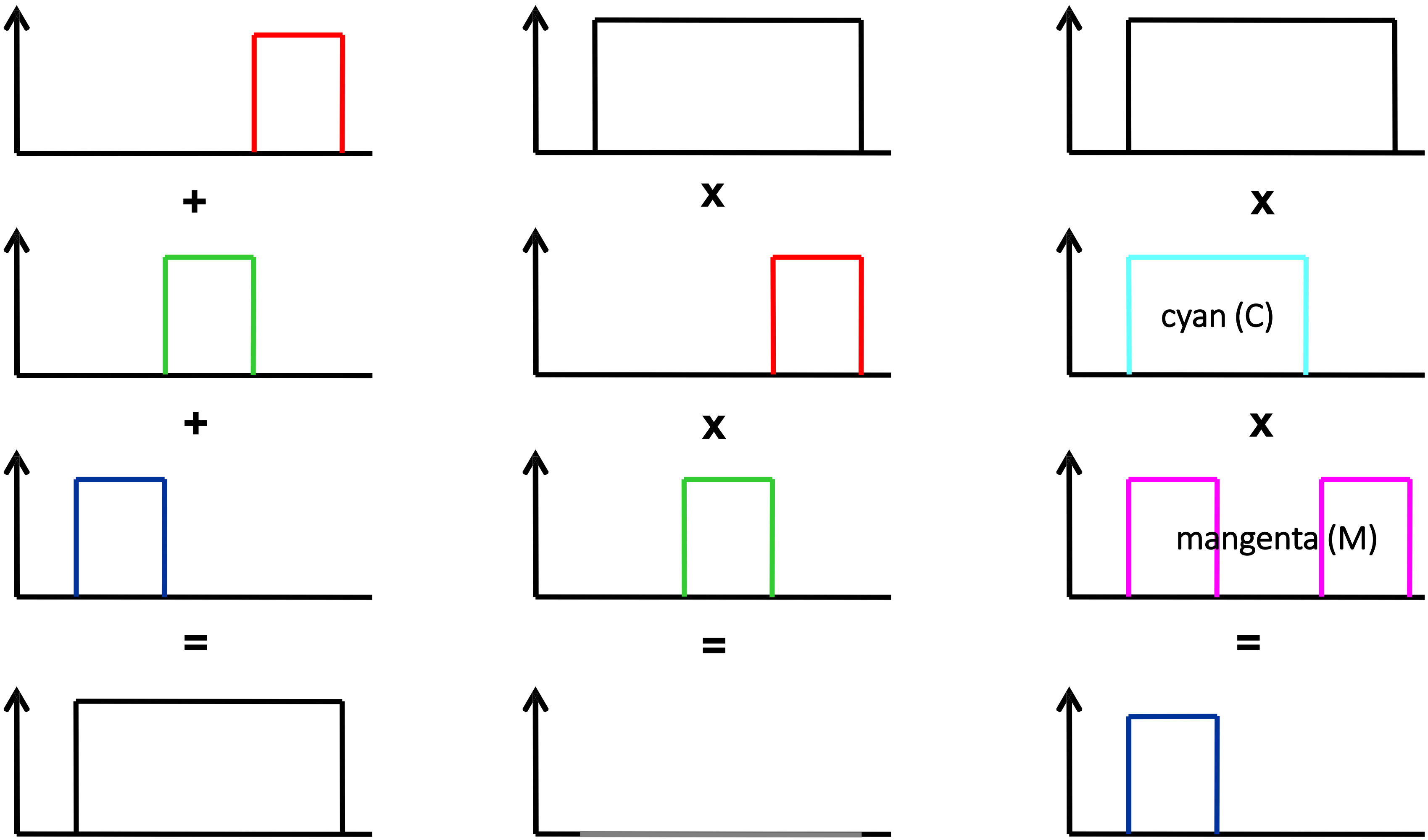
$$u^* = 13L * (u' - u'_n)$$

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

$$f(x) = \begin{cases} x \geq 0.008856 & x^{1/3} \\ x < 0.008856 & 7.787x + {}^{16}/_{116} \end{cases}$$

- Limited applicability to HDR

Corresponds to stacked color filters



Additive blending

Subtractive blending
Multiply by primaries: wrong !

Subtractive blending
Multiply by inverse primaries

Primarily used for printers

CMYK (Cyan, Magenta, Yellow, Black)

- In theory:
  - (C, M, Y) = 1 – (R, G, B)   // Hence "subtractive" color space
  - K = min(C, M, Y)   // Black (B already used for blue!)
  - (C, M, Y, K) = (C - K, M - K, Y - K, K)
- In practice: profoundly non-linear transformation
  - Other primary colors
  - Interaction of the color pigments among each other
  - Covering
  - Etc, *etc*...

Subtractive primary colors:

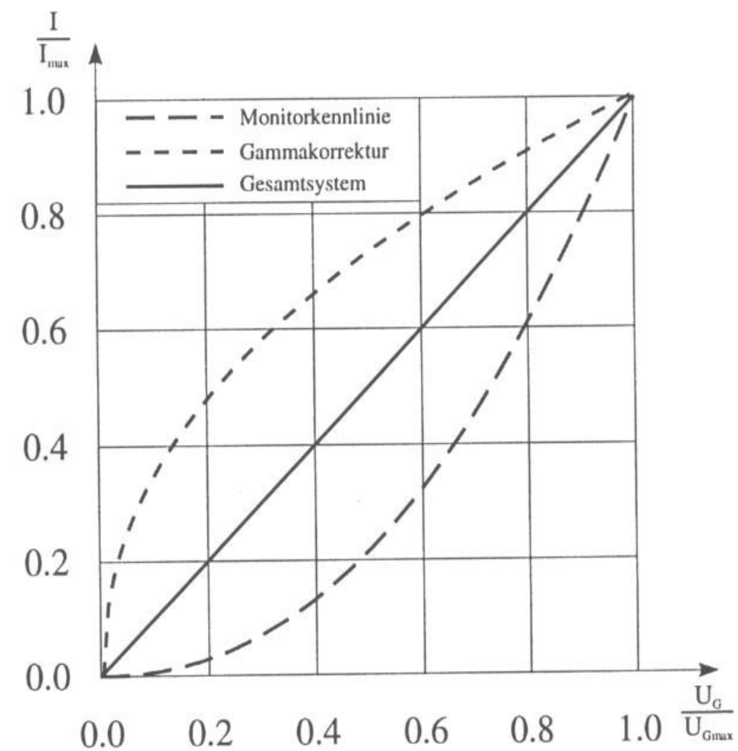
- Product of all primary colors must be black
- Any number of colors (CMY, CMYK, 6-color-print, *etc*...)
- It does not need to obtain (CMY) = 1 - (RGB)

Display-Gamma

- Intensity I of electron beam in CRT monitors is non-linear with respect to the applied voltage U
- Best described as power law: $L = U^{\gamma}$
- Gamma-Factor $\gamma = \sim 2.2$ due to physical reasons
- For compatibility also in other displays (LCD, OLED, *etc.*)

Gamma correction

- Pre-correct values with inverse to achieve linear curve overall
- Quantization loss if value represented with less than 12 bits
  - Hardly ever implemented this way in apps and HW



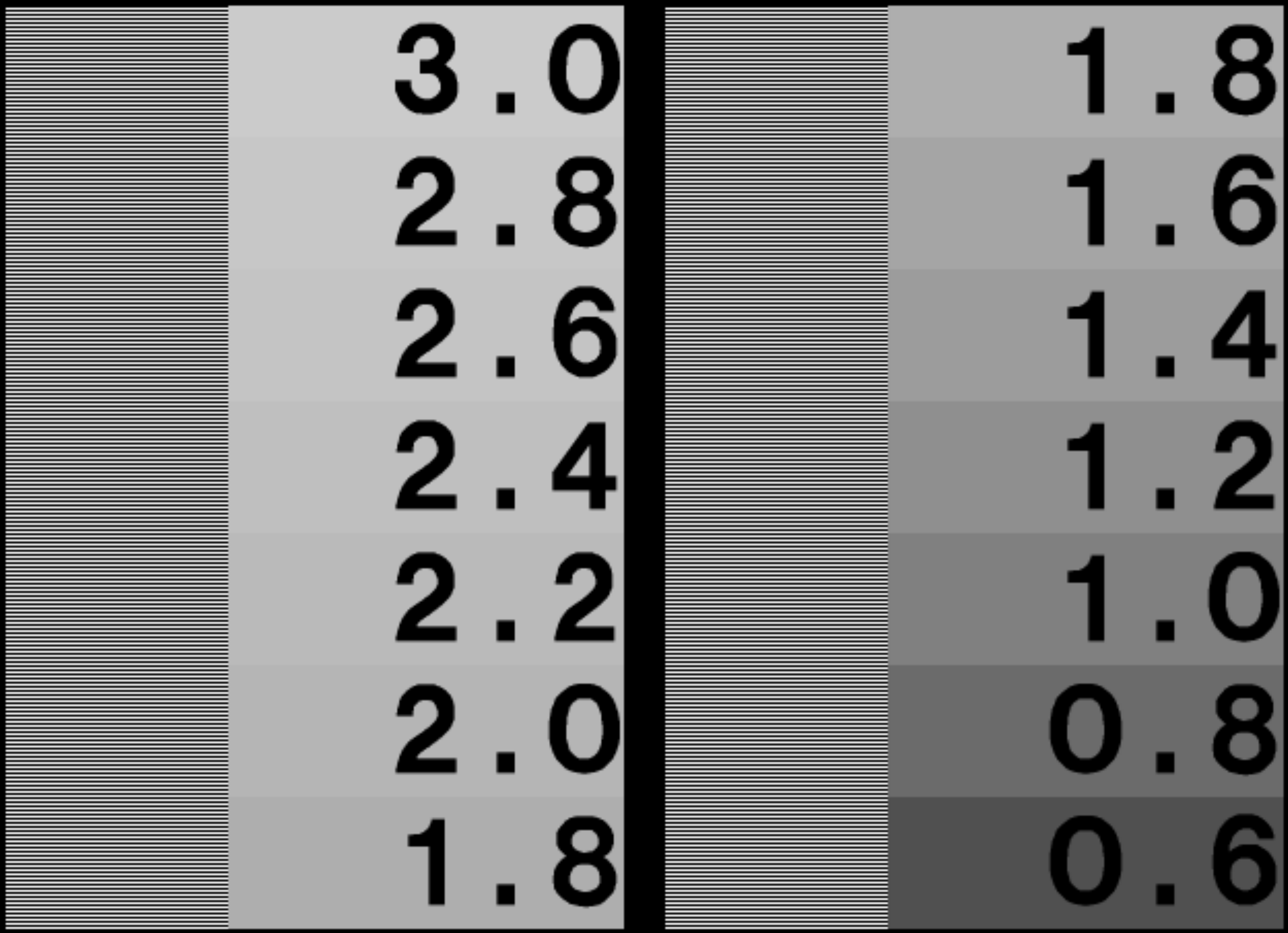
Gamma of monitor not always theoretical 2.2

Testing:

- 50% intensity should give 50% grey (half black-white)
- Match actual gray with true black / white average → γ

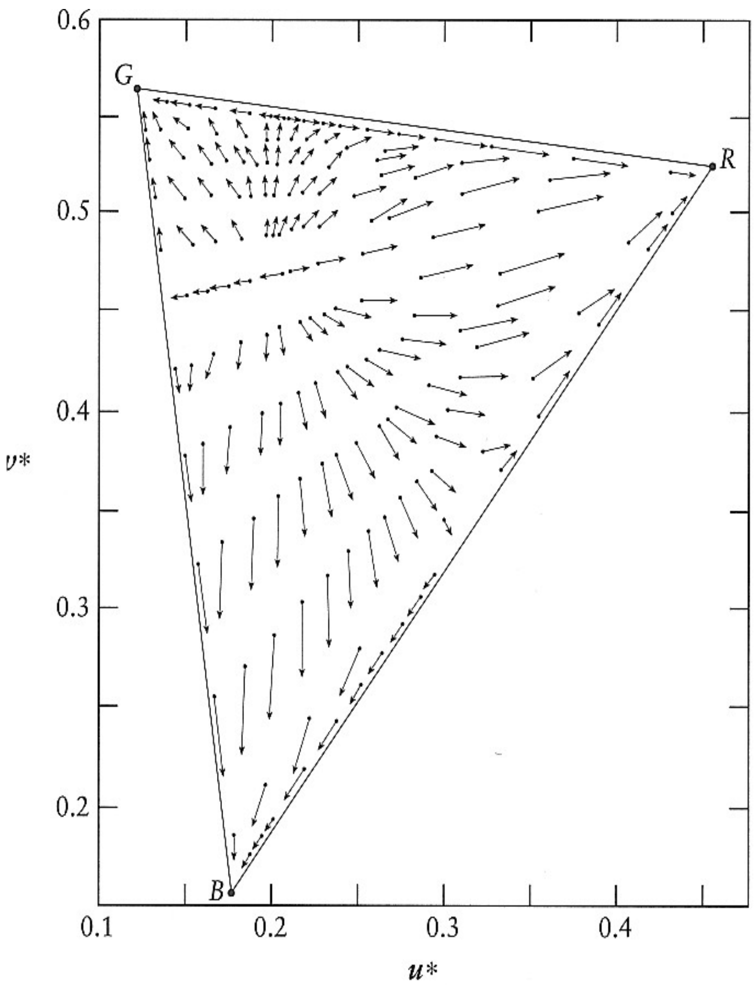
Usage of the gamma testing chart:

- Take some distance from your monitor, such that you can no longer see the horizontal lines.
- Where brightness of the lines area is equal to brightness of the number area, read your gamma setting.


3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.8
1.6
1.4
1.2
1.0
0.8
0.6

Problem:

- Non-linear operator: RGB components not uniformly scaled by a constant factor ⇒ strong color corruptions



Shifts in reproduced chromaticities resulting from uncompensated gamma of 1.273 (such a gamma is desirable to compensate the contrast lowering in the dim surround).

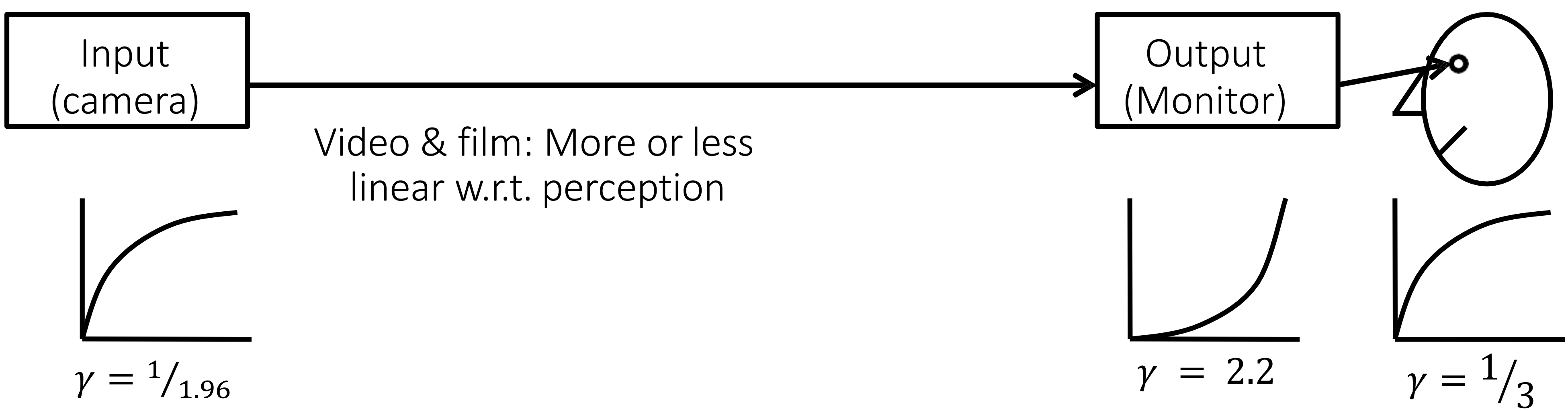
Camera-gamma

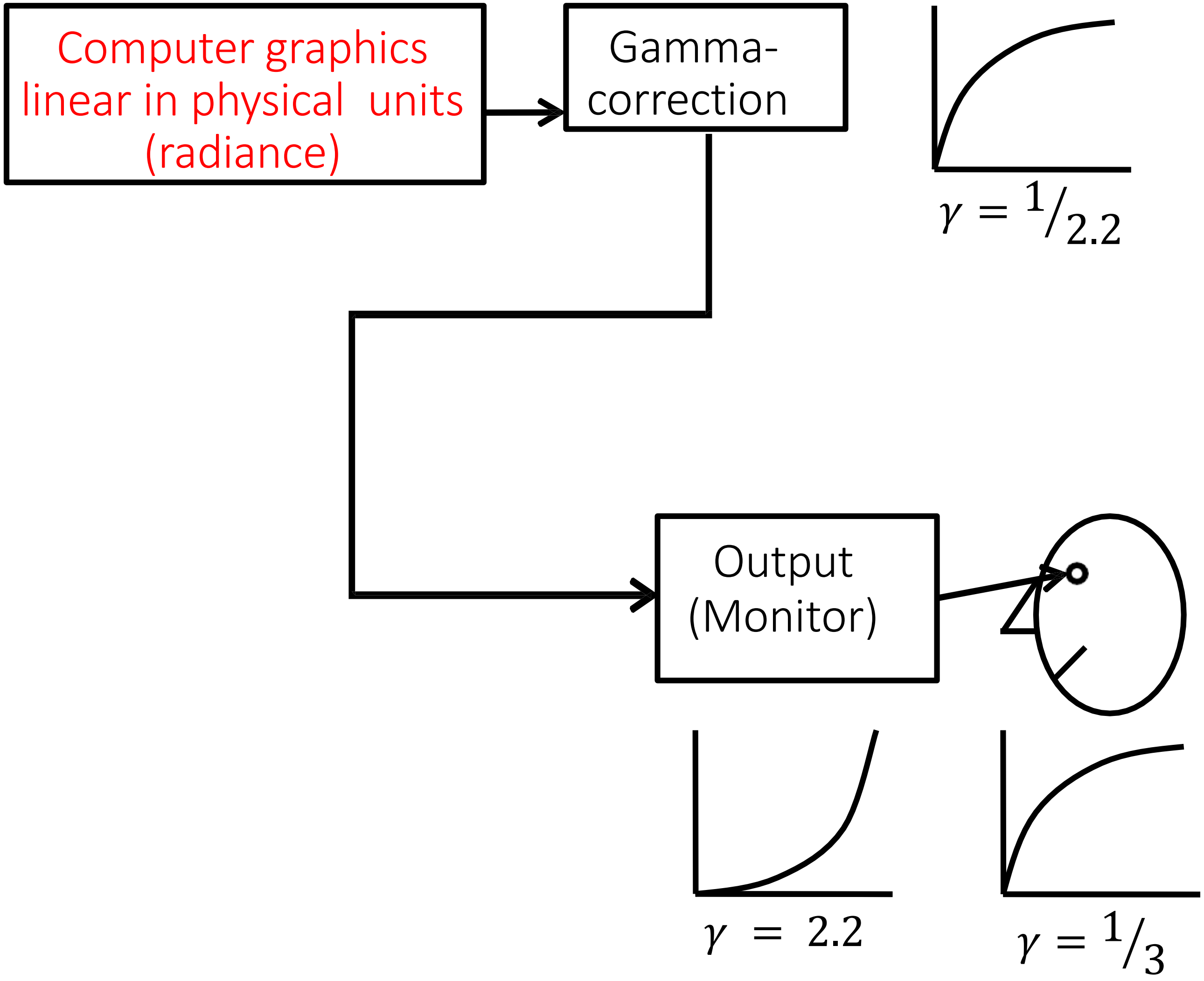
- Old cameras (electron tube) also had a gamma factor
- Essentially the inverse of the monitor gamma (due to physics)
  ⇒Display corrected the camera

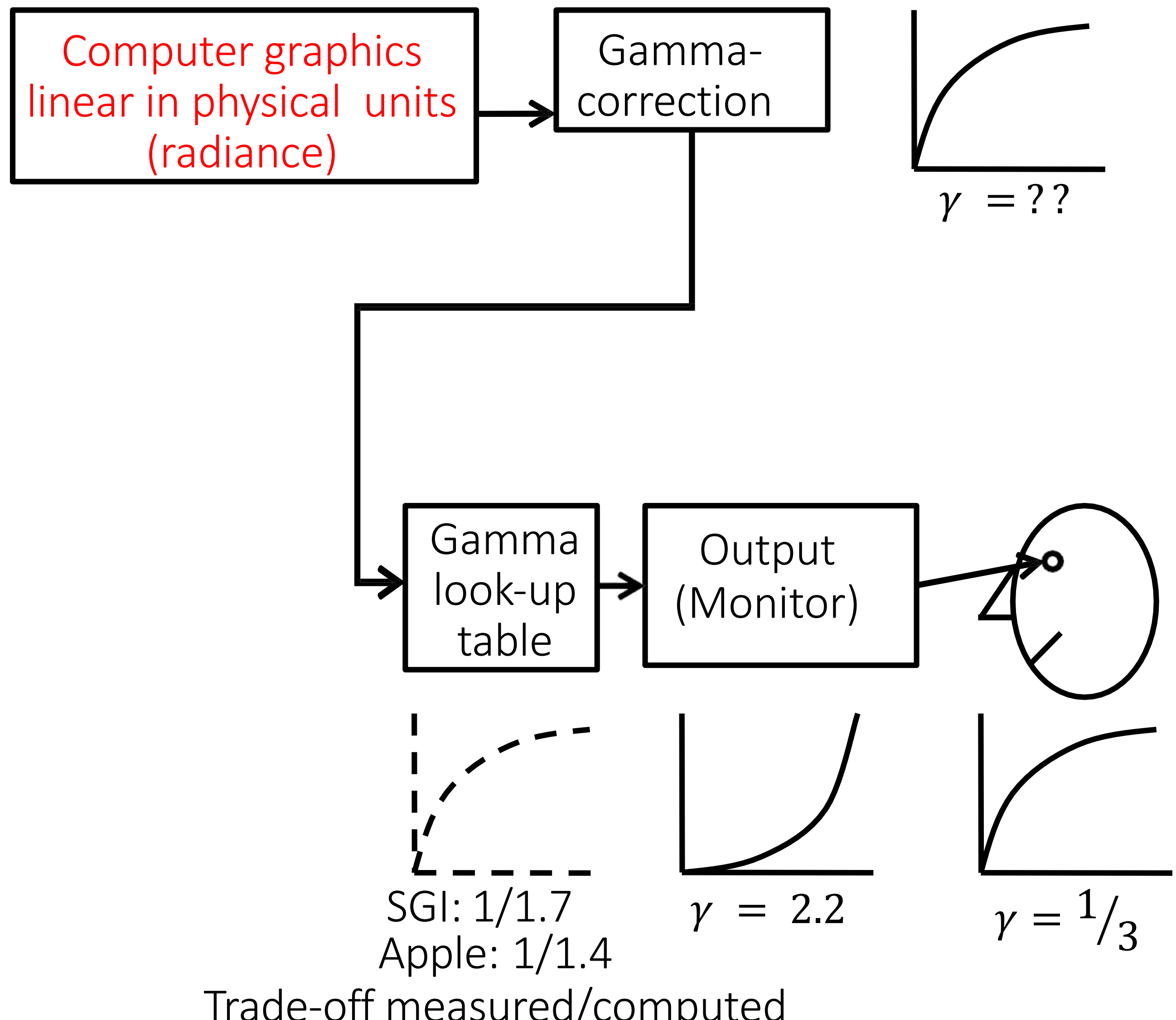
"Human-gamma"

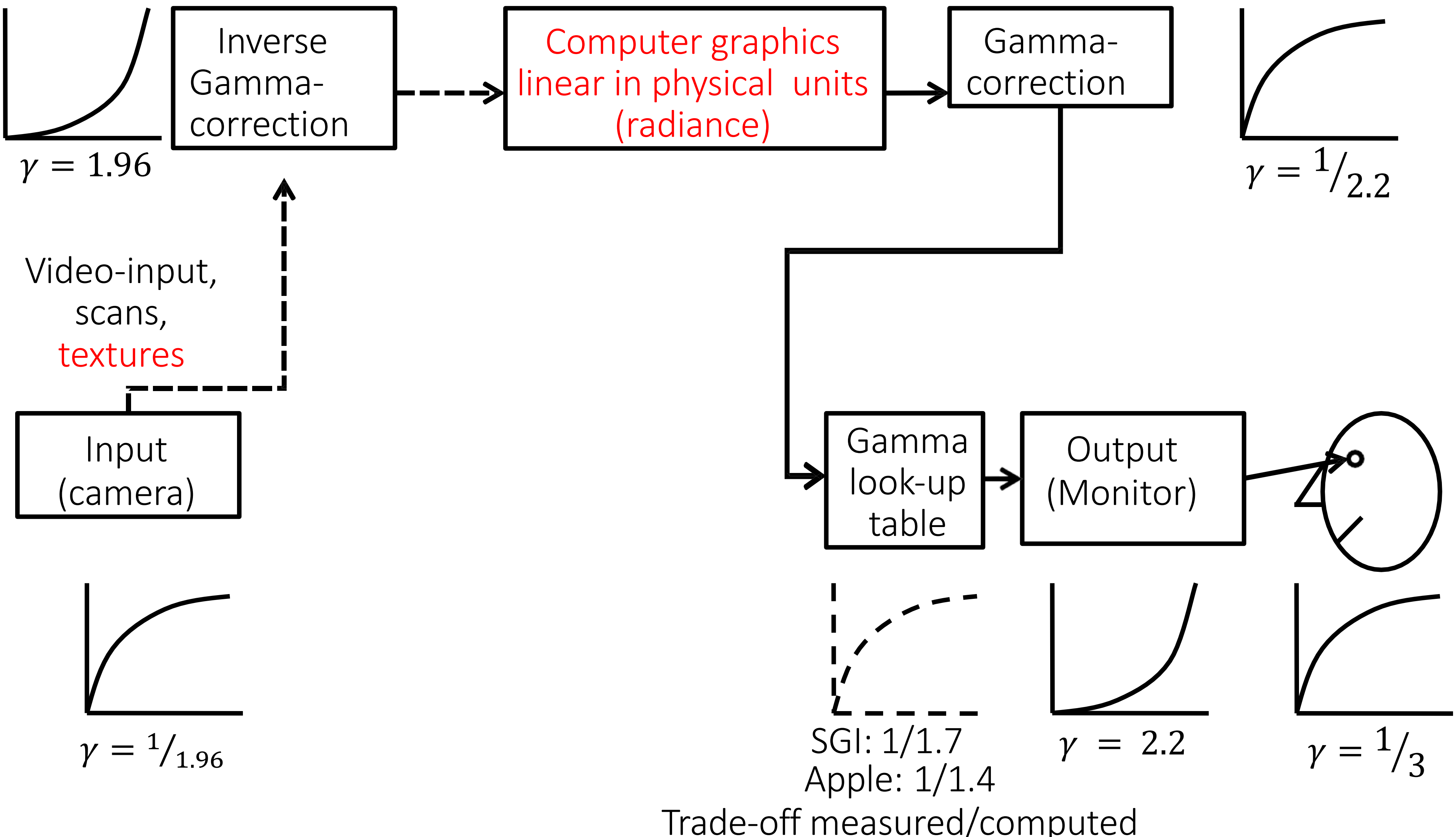
- Human brightness perception exhibits a log curve response
- Actually roughly follows a gamma curve with a value of 1/3
- Old cameras therefore encode light in a perceptually uniform way
  - Optimal for processing, compressing and transmitting values
- New cameras specifically generate the same output for compatibility reasons (!)

Color from Beginning to End




Computer graphics linear in physical units (radiance)
Gamma-correction
$\gamma = {}^{1}/_{2.2}$
Output (Monitor)
$\gamma = 2.2$
$\gamma = {}^{1}/_{3}$


Computer graphics linear in physical units (radiance)
Gamma-correction
$\gamma = ??$
Gamma look-up table
Output (Monitor)
SGI: 1/1.7
Apple: 1/1.4
Trade-off measured/computed
$\gamma = 2.2$
$\gamma = {}^{1}/_{3}$


Inverse Gamma-correction
Computer graphics linear in physical units (radiance)
Gamma-correction
$\gamma = 1.96$
$\gamma = {}^{1}/_{2.2}$
Video-input, scans, textures
Input (camera)
Gamma look-up table
Output (Monitor)
$\gamma = {}^{1}/_{1.96}$
SGI: 1/1.7
Apple: 1/1.4
Trade-off measured/computed
$\gamma = 2.2$
$\gamma = {}^{1}/_{3}$

Problems

- Color coordinate system often unknown
  - No support in image formats
  - Assume sRGB!
- Multiple color-space transformations
  - Loosing accuracy through quantization
    - Unless floats or many bits are used
- Gamma-correction depends on application
  - Non-linear:
    - Video-/image editing (but not all operations!)
  - Linear:
    - Image syntheses, interpolation, color blending, rendering, ...

International Color Consortium

- Standardized specification of color spaces
- Profile Connection Space (PCS) – intermediate, device-independent color space (CIELAB and CIEXYZ supported)
- ColorDevice #1 → PCS → ColorDevice #2

ICC profile

- A file with data describing the color characteristics of a device (such as a scanner, printer, monitor) or an image
- Simple matrices, transformation formulas (if necessary proprietary)
- Conversion tables

ICC library

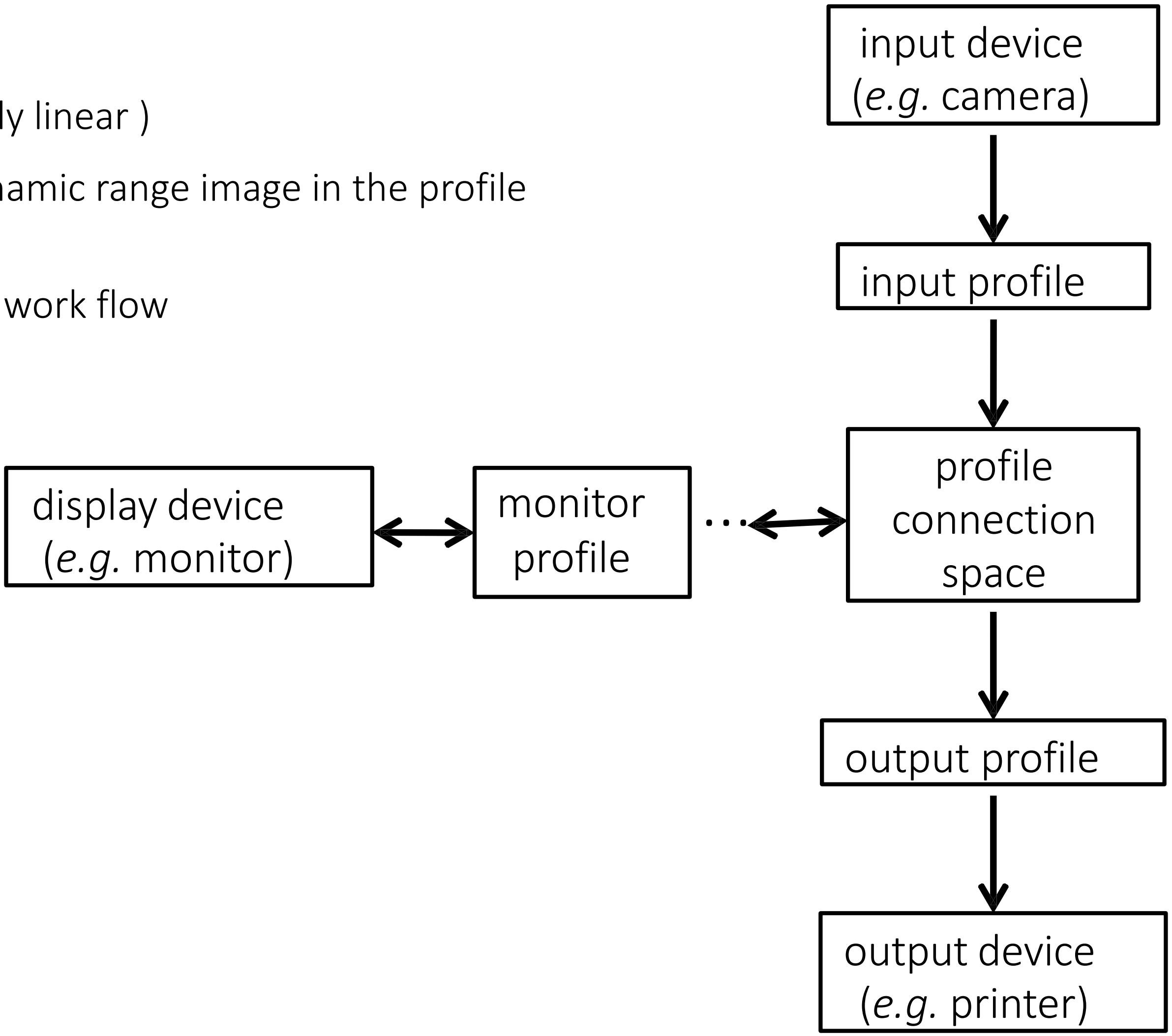
- Using profiles for color transformations
- Optimizes profile-sequences transformations, but no standard-API

Problems

- Inaccurate specifications, interoperability
- Profiles difficult to generate

ICC processing

- Typical profile connection spaces
    - CIELAB (perceptual linear)
    - CIEXYZ color space (physically linear)
- Can be used to create an high dynamic range image in the profile connection space
    - Allows for a color calibrated work flow



History

- Usually little user data, mostly data curated professionally
- Color issues with Web images due to different color displays
  - "Solved" by sRGB color space and better monitors (LCD/OLED)

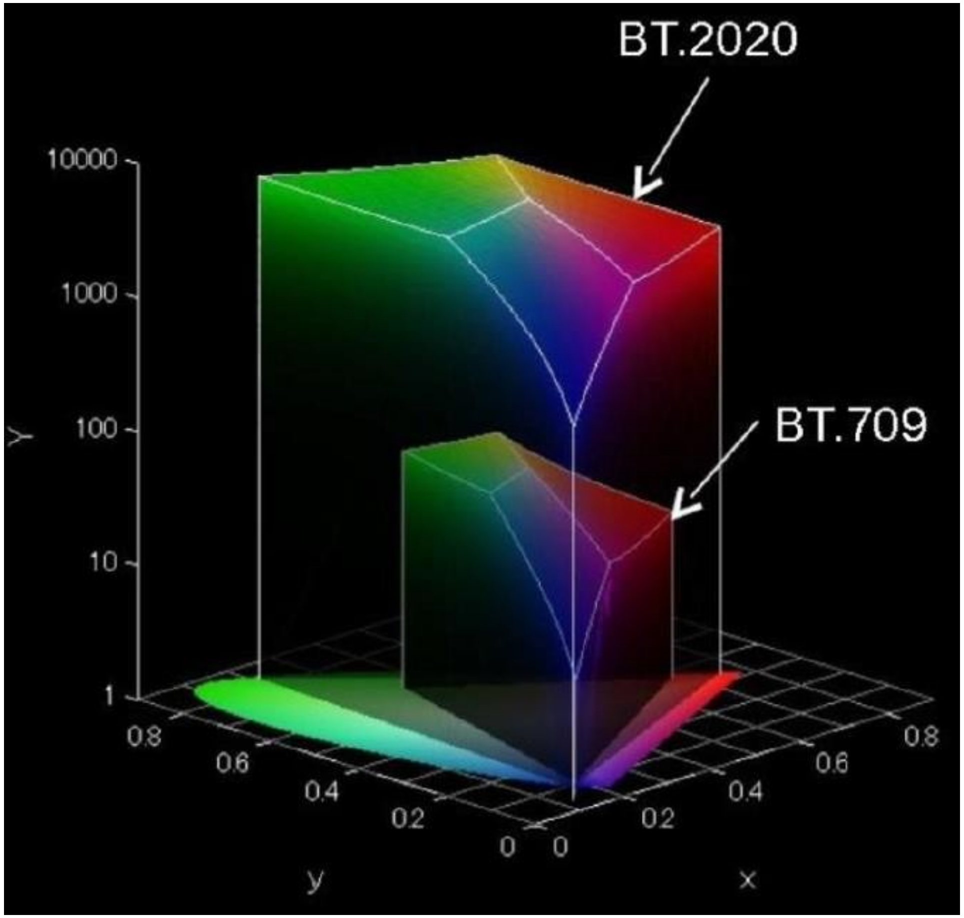
Big confusion: HDR Format (HDR10 vs. Dolby Vision)

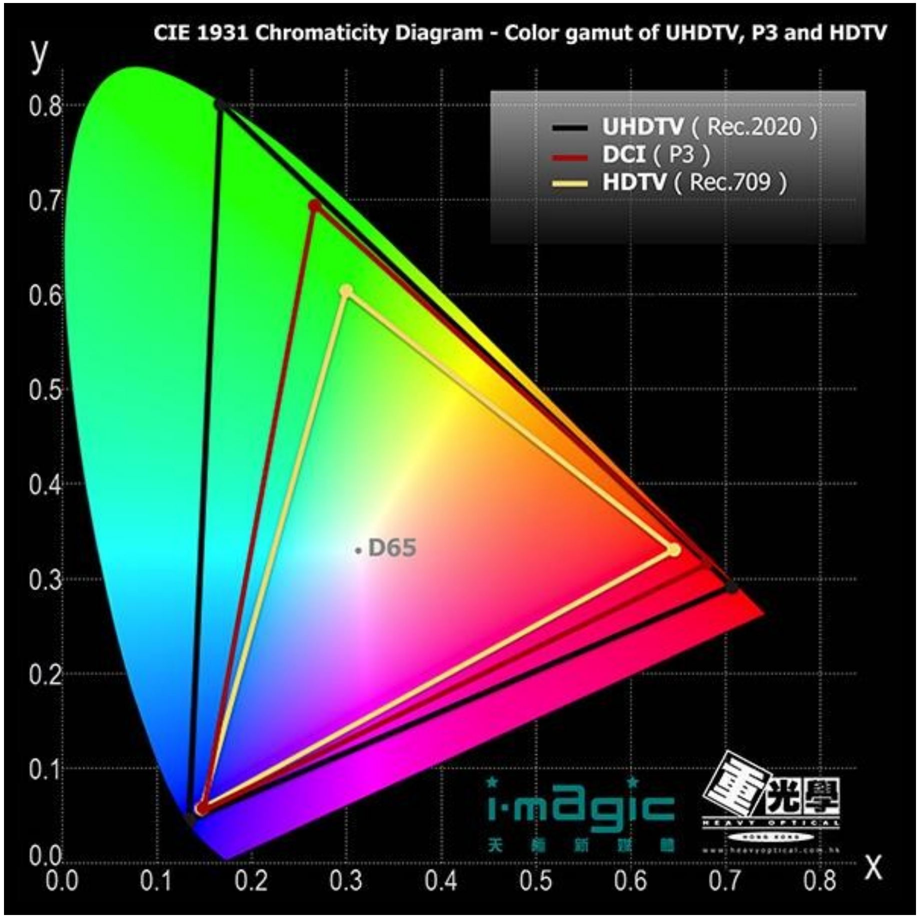
- Quantization (10 vs. 12 bit/sample)
- Color spaces (DCI-P3 vs. Rec. 2020)
- Maximum brightness (1 000 vs. 10 000 nits)
- Transfer functions (Perceptual Quantizer vs. Hybrid Log Gamma)
- Frame rate (!)
- Issue of "best" reconstruction filter during rendering
- Little support for still images (*e.g.* OpenEXR, JPEG-XR)
- Varying support in consumer displays, no cameras yet
- No good support for interactive applications (yet)

Need for tone and gamut mapping

- Because each display may be different

What's the expected behavior? What about reverse?





Assignment 3 (Programming part)

Submission deadline: Friday, 25. October 2019 9:45 (before the lecture)
Programming Assignments should be submitted via GitHub (please send your TA the link to you version (fork) of the repository). Every assignment sheets counts 100 points (theory and practice).

3.1 OBJ Scene loader (Points 30)
3.2 Implementation of a kd-tree acceleration structure (Points 70)
Goto: https://github.com/Jacobs-University/eyden-tracer-03 (for the task details)

Need Help? Ask TAs! and / or join CG Slack workspace!

- To join the **CG Slack workspace**, please write me at sergej.kosov@gmail.com a short (or empty) e-mail with subject: *"Join CG SLACK".*
  I will reply with *email invitation.*

Assistants

- Kechun Gu
  - Email: k.gu@jacobs-university.de
- Minji Kim
  - Email: m.kim@jacobs-university.de