COLOR IMAGES, COLOR SPACES AND COLOR IMAGE PROCESSING

Ole-Johan Skrede

08.03.2017

INF2310 - Digital Image Processing

Department of Informatics
The Faculty of Mathematics and Natural Sciences
University of Oslo

After original slides by Fritz Albregtsen

TODAY'S LECTURE

- Color, color vision and color detection
- Color spaces and color models
- Transitions between color spaces
- Color image display
- Look up tables for colors
- Color image printing
- Pseudocolors and fake colors
- Color image processing
- Sections in Gonzales & Woods:
    - 6.1 *Color Funcdamentals*
    - 6.2 *Color Models*
    - 6.3 *Pseudocolor Image Processing*
    - 6.4 *Basics of Full-Color Image Processing*
    - 6.5.5 *Histogram Processing*
    - 6.6 *Smoothing and Sharpening*
    - 6.7 *Image Segmentation Based on Color*

MOTIVATION

- We can differentiate between thousands of colors
- Colors make it easy to distinguish objects
  - Visually
  - And digitally
- We need to:
  - Know what color space to use for different tasks
  - Transit between color spaces
  - Store color images rationally and compactly
  - Know techniques for color image printing

THE COLOR OF THE LIGHT FROM THE SUN

SPECTRAL EXITANCE

The light from the sun can be modeled with the *spectral exitance of a black surface* (the radiant exitance of a surface per unit wavelength)

$$M(\lambda) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{\exp\left\{\frac{hc}{\lambda kT}\right\} - 1}.$$

where

- $h \approx 6.626\,070\,04 \times 10^{-34}\ \mathrm{m^2\,kg\,s^{-1}}$ is the *Planck constant*.
- $c = 299\,792\,458\ \mathrm{m\,s^{-1}}$ is the speed of light.
- λ [m] is the radiation wavelength.
- $k \approx 1.380\,648\,52 \times 10^{-23}\ \mathrm{m^2\,kg\,s^{-2}\,K^{-1}}$ is the *Boltzmann constant*.
- T [K] is the surface temperature of the radiating body.

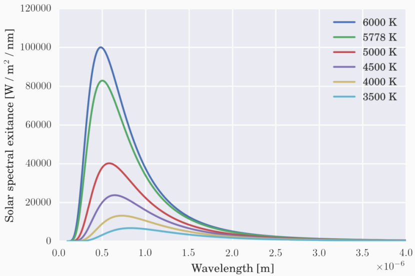


Figure 1: Spectral exitance of a black body surface for different temperatures.

SPECTRAL IRRADIANCE

The distance from the earth to the sun is about $d \approx 1.496 \times 10^{11}$ m, and the radius of the sun is about $r \approx 6.957 \times 10^{8}$ m. The radiation we measure at the top of the earth's atmosphere from a black body sphere is then given by the spectral irradiance

$$E_0(\lambda) = M(\lambda) \left(\frac{r}{d}\right)^2 .$$

The wavelength at the peak is given by the *Wien displacement law*

$$\lambda_{max} \approx \frac{2.897\,772\,9 \times 10^{-3}\,\text{m K}}{T} \approx 501.51\,\text{nm}$$

which is green ($RGB \approx (0, 255, 135)$).

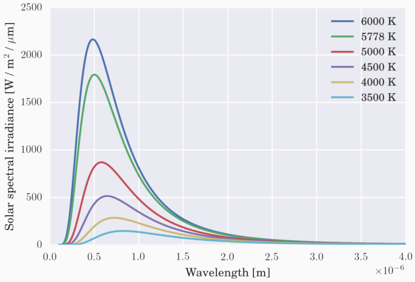


Figure 2: Spectral irradiance of a black body surface for different temperatures.

SPECTRUM OF SOLAR RADIATION

- The spectrum of the Sun's solar radiation is close to that of a black body.
- The Sun emits EM radiation across most of the EM spectrum.
- The stronges output is in the range for visible light (from about 380 nm to about 780nm.

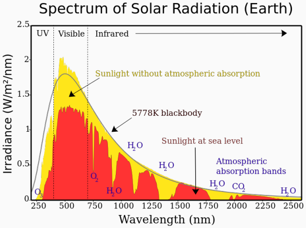


Figure 3: Solar irradiance spectrum above atmosphere and at surface. Extreme UV and X-rays are produced (at left of wavelength range shown) but comprise very small amounts of the Sun's total output power. (Source: By Nick84 - http://commons.wikimedia.org/wiki/File:Solar_spectrum_ita.svg, CC BY-SA 3.0,

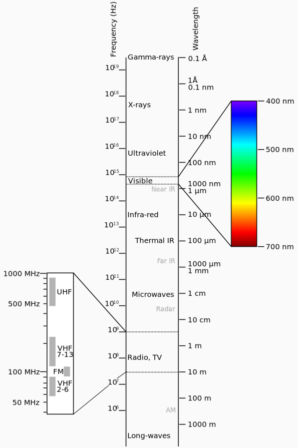


Figure 4: The electromagnetic spectrum (Source: By Victor Blacus - SVG version of File:Electromagnetic-Spectrum.png, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=22428451`)

LIGHT PROPAGATION BEHAVIOUR

- **Absorption:** Light stops at the object, and do not reflect or refract, making it appear dark.
- **Reflection** (on a smooth surface): Light bounces off the surface of a material at an angle equal to the incident angle (as in a mirror).
- **Scatter** (reflection on rough surface): Light bounces off in many directions.
- **Transmission:** Light travels through the object. Example: Glass.
- **Refraction:** Changes in wave speed when light transmits through a different material causes distortions from the original path.
- **Diffraction:** Bending of light around corners of an obstacle or an aperture.

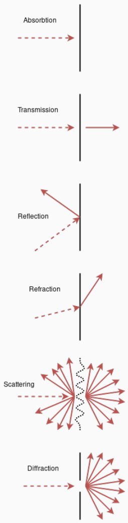
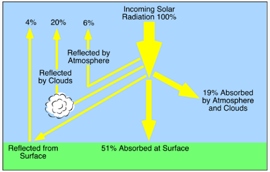
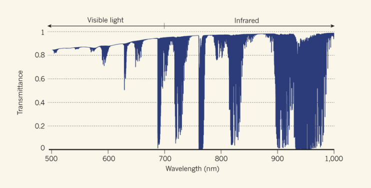


Figure 5: Different light path options

- Not all radiation from the sun that reaches our planet reaches the earth's surface.
- A lot of radiation (especially in the UV and IR wavelength ranges) are absorbed by molecules such as H_2O and CO_2.



(a) Solar radiation partitions



(b) Radiation transmittance

Figure 6: Sunlight and the earth.

- Dispersion is the phenomenom where the phase speed of the light is dependent on the frequency.
- This is a property of the propagation media, and they are called *dispersive media*

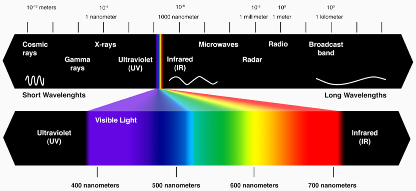


Figure 7: Colors and wavelengths

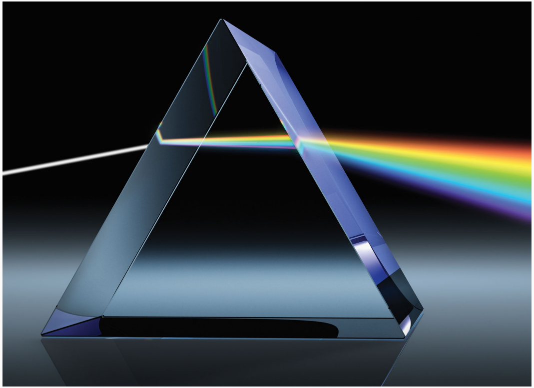


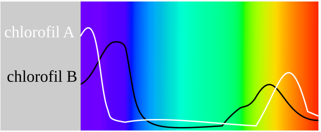
Figure 8: Illustration of dispersion. The different colors in white light propagating through a dispersive prism is revealed since different wavelengths are refracted at different angles.

- *Achromatic light* is light without color, grayscale etc. Characterized by its intensity.
- *Chromatic light* is the colors, and span a narrow band of the electromagnetic spectrum
- Thre basic quantities determines chromatic light:
    - *Radiance:* Total amount of energy from the light source. Measured in Watts [W]
    - *Luminance:* The amount of energy an observer percieves. Measured in lumens [lm]
    - *Brightness:* Analogue to the achromatic intensity. Subjective and difficult to measure.

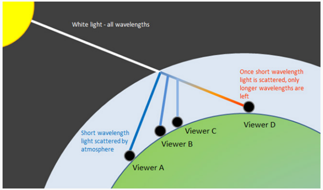
THE COLOR OF AN OBJECT

- The color we percieve in an object is determined by the nature of the light reflected by the object.
- Therefore, the color of an object is determined by
  - The light hitting the object.
  - How much light is reflected and absorbed.
- Or, in other words
  - The spectral distribution of the incident light.
  - The spectral distribution of the reflected light.
- The reflection properties are determined by
  - Chemical pigments
  - Physical surface structures
- Together, this determines what wavelengths are reflected, absorbed or transmitted.
- How we actually percieve color is complicated. Two complementary theories: *Trichromacy* and *Color opponency*.

- Vegetation
  - Green plants are green because of a pigment called *chlorophyll* which absorbs red and blue wavelengths.
  - As chlorophylls degrade in the autumn, hidden pigments of yellow *xanthophylls* and orange *beta-carotene* are revealed.
- The sky
  - When the sun is high in the sky: the blue part (shorter wavelengths) are scattered from air particles more than red light. (Technically, the sky is violet, but we perceive it as blue. More on that in a second.)
  - At sunset: More atmosphere to propagate throught, thus most of the blue light is scattered, revealing the red light.



(a) Absorbtion spectrum of chlorophyll



(b) Sky colors

TRICHROMATIC THEORY (YOUNG, HELMHOLTZ)

- The retina is responsible for light detection in the eye
- Two types of detectors: rods and cones
- Rods detects achromatic light
- Cones detects chromatic light, and can be devided into three principal sensing receptors (*cone opsins*) that has peak sensitivity at different parts of the spectrum
    - S (short): Around blue ($\sim$ 445 nm, (2 %, but most sensitive)
    - M (middle): Around green ($\sim$ 535 nm, (33 %)
    - L (long): Around red ($\sim$ 575 nm, (65 %)

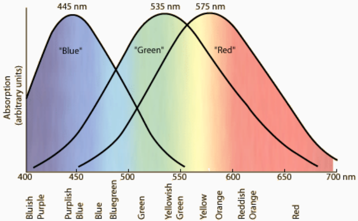


Figure 10: Absorbtion of light by S ("blue"), M ("green") and L ("red") cones.

COLOR OPPONENCY THEORY (HERING)

- Three classes of opponent channels:
  - Red-Green: Either red or green (no such thing as greenish-red). Senses the difference (S + L) - M
  - Yellow-Blue: Either yellow or blue (no such thing as bluish-yellow). Senses the difference (M + L) - S
  - White-Black: Luminesance level, a spectrum of graylevels.
- The response from S-, M-, and L-cones are shown in the figure to the right
- This can explain some aspects of color blindness, and color afterimages

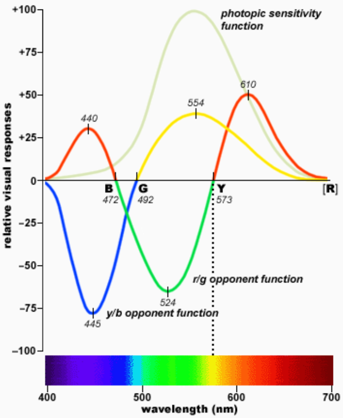
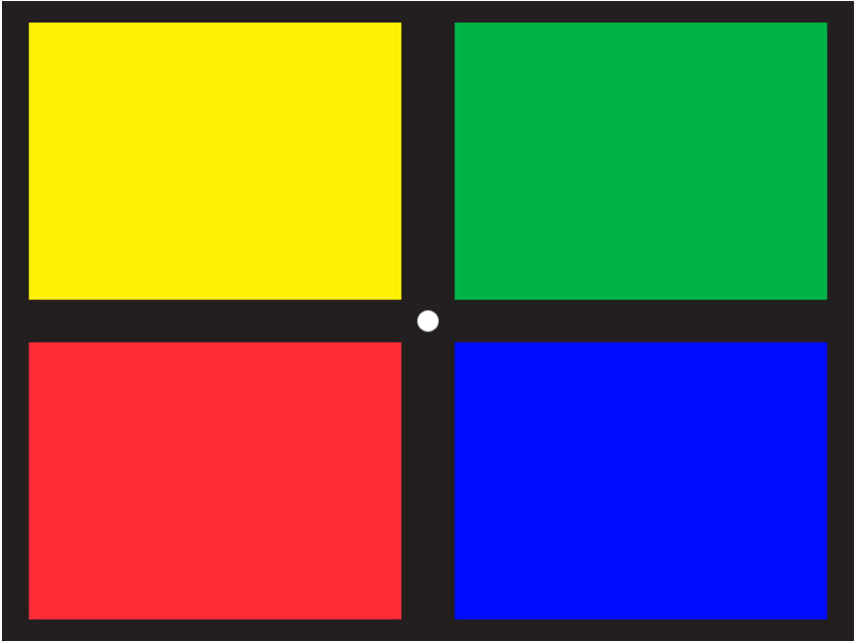


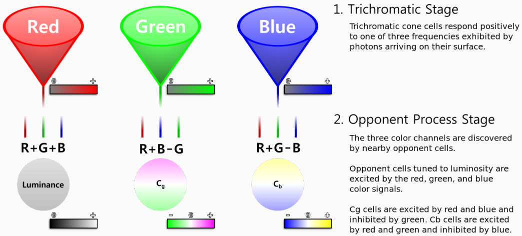
Figure 11: Opponent functions in spectral hues chromatic response functions for the CIE 1964 standard observer. (Source: https://www.handprint.com/HP/WCL/color6.html)



STAGE (OR ZONE) THEORY

The brain combines information from each type of receptor to give rice to different color perception. Modern theory of visual perception incorporates trichromatic theory and opponence theory, in two stages. In summary, some percieved colours can be explained as

- Blue light stimulates S more than green or red light, but M and L more weakly
- Blue-green light stimulates M moore than L, and S more strongly
- Green light stimulates M more than S and L
- Green-Yellow light stimulates both L and M equally strong, but S weakly
- Red light stimulates L much more than M, and S hardly at all



COLOR DESCRIPTION

SOME WORDS AND THEIR MEANINGS

All colors can be fully described by their *hue*, *saturation* and some form of *intensity, or brightness value*.

- Hue: The dominant wavelength of the color.
- Saturation: How saturated the color is with white light, from a pure color (no white) to fully saturated (white).
- Intensity: How bright the color is, some notion of darkness.
- Hue and saturation together determines the color, and we call these two *chromaticity*.

- The chromaticity determines both the dominating wavelength and the saturation of the color.
- Chromaticity and brightness together fully describes a color.
- For instance, different graylevels have the same chromaticity, but different intensity.

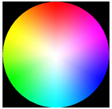


Figure 13: Chromaticity modeled as a unit polar coordinate chart (ρ, θ) where fully saturated colors are at the boundary ($\rho = 1$), and the angle θ represent the wavelength, and thereby the hue.

ADDITIVE COLOR MIXING

- Mixing light of two or more colors
- Additive primaries: Red, Green, Blue
- Additive secondaries: Yellow, Cyan, Magenta
- Utilized in TV and Computer monitors
- Difference between a mix of red and green (yellow) and yellow light (wavelength at about 580 nm), but we detect no difference.

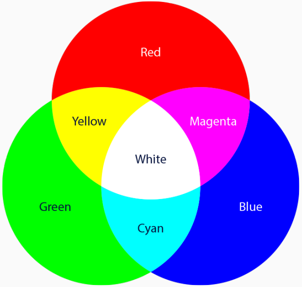


Figure 14: Additive mixing

SUBTRACTIVE COLOR MIXING

- Mixing filters or pigments that absorbs and reflect different colors.
- If you begin with white light, the result color you see is a result of a subtractive color process where different wavelengths have been absorbed.
- Subtractive primaries: Yellow, Cyan, Magenta
- Subtractive secondaries: Red, Green, Blue
- Mixing of paint is subtractive (although, you probably used a RYB color model in kindergarden).

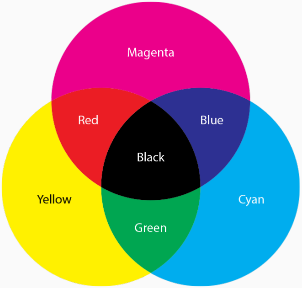


Figure 15: Subtractive mixing

HUMAN COLOR VISION

- We can differentiate between about 100 different hues
- We can differentiate between about 6000 different hue and intensity combinations.
- For each of those, we can differentiate about 60 different saturation levels.
- In total we differentiate about 360 000 colors.

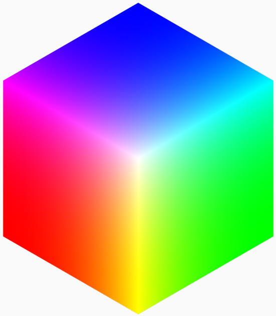


Figure 16: RGB cube

COLOR SPACES

COLOR MODELS AND COLOR SPACES

- A color model is an abstract mathematical model describing how colors can be represented by tuples of numbers (e.g. RGB or CMYK)
- A color space is a specific organization of colors. Can have different color spaces for a specific color model
- E.g. sRGB and Adobe RGB are color spaces of the RGB color model.
- Color spaces and models are sometimes (incorrectly) used indiscriminately.

Systems covered today.

- CIE standard color model
  - CIEXYZ
- RGB(A) color model
  - sRGB, Adobe RGB
  - HSV, HSL and HSI
- CMY(K) color model
- YUV color model
  - YUV, YIQ, YC_bC_r

What is not covered is e.g. the *LAB* color model (see e.g. Hunter Lab and CIEL*a*b*).

CIE STANDARD OBSERVER

- CIE (*Commission Internationale de l'Eclairage*) defined a model of standard tristimulus values $\{X, Y, Z\}$.
- $\{X, Y, Z\}$ are hypothetical, but $X \sim R$, $Y \sim G$ and $Z \sim B$
- A quantitive measurement of colors, where any wavelength can be matched perceptually by positive combinations of X, Y, and Z.
- Defined such that for a hypothetical light source with temperature 5400 K, $X = Y = Z = 1$
- In the same way, white light from indirect daylight (D65, 6500 K), $X = 0.950456$, $Y = 1$, $Z = 1.088754$.

$$X = \int_{380}^{780} L_{e,\Omega,\lambda}(\lambda)\bar{x}(\lambda)\,\mathrm{d}\lambda,$$

$$Y = \int_{380}^{780} L_{e,\Omega,\lambda}(\lambda)\bar{y}(\lambda)\,\mathrm{d}\lambda,$$

$$Z = \int_{380}^{780} L_{e,\Omega,\lambda}(\lambda)\bar{z}(\lambda)\,\mathrm{d}\lambda,$$

$L_{e,\Omega,\lambda}$ is the spectral radiance.

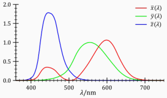


Figure 17: The CIE 1931 standard observer color matching functions [1].

[1] By User:Acdx - Own work, GFDL, https://commons.wikimedia.org/w/index.php?curid=6233111

- From the tristimulus values, we can construct a set of chromaticity coordinates $\{x, y, z\}$.

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

- Note that $z = 1 - x - y$
- This is often used to derive different color spaces.
- The chromaticity diagram on the left represent the *human gamut* (colors visible to us).
- A gamut is a certain complete subset of colors.

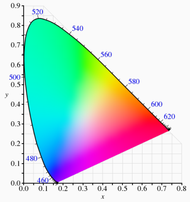


Figure 18: The CIE 1931 xy color space chromaticity diagram [1].

[1] By BenRG - File:CIExy1931.svg, Public Domain, https://commons.wikimedia.org/w/index.php?curid=7889658

CIE HUMAN GAMUT

- The curved edge of the gamut called the *spectral locus*, and is fully saturated monochromatic colors (each point representing a wavelength).
- The straight line on the lower part is called the *line of purples*, and have no monochromatic color equivalent. Purple and magenta do not exist in the rainbow (but violet do).
- The line segment between two points contain all colors that can be created with different mixtures of the two colors at those points.
- Therefore a color space gamut is convex. E.g. a triangle if it is built of three primary colors.

Example gamut of the CIE 1931 RGB primaries, where

Red: 700 nm

Green: 546.1 nm

Blue: 435.8 nm

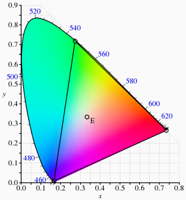
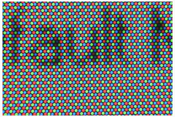


Figure 19: The CIE 1931 RGB [1].

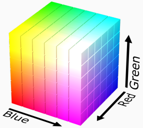
[1] By BenRG - Own work, inspired by File:CIExy1931.png, Public Domain, https://commons.wikimedia.org/w/index.php?curid=7889718

- An additive color model where the primaries red, green and blue, are added together to form a wide variety of colors.
- Device dependent, meaning that the color is dependent on the device that detect or output the color.
- Typical input devices are video cameras, digital cameras and image scanners.
- Typical output devices are TV, computer and mobile displays.

- Organized in a unit cube with coordinates (r, g, b), where $(0, 0, 0)$ is black and $(1, 1, 1)$ is white.
- A typical 24 bit display has 256 values for each channel.



(a) CRT monitor [1]



(b) RGB color cube [2]

[1] By Gona.eu - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=6223756
[2] By SharkD - Own work, GFDL, https://commons.wikimedia.org/w/index.php?curid=3375025

RGBA COLOR MODEL

- As the RGB color model with and added α-channel.
- Usually used as an opacity channel. 0% is fully transparent, and 100% is fully opaque.
- Allows image composition, as background is visible trough foreground if the foreground is transparent.

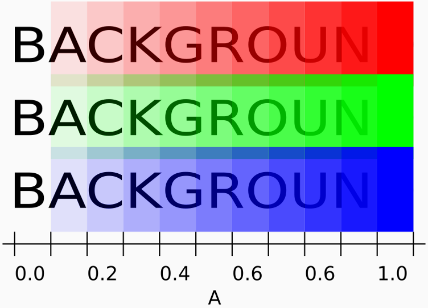


Figure 21: Red, green and blue colors with different transparency (alpha). Note that when overlapping, blue is overlapping green which is overlapping red.

SRGB COLOR SPACE

- Created in 1996 by HP and Microsoft to approximate the color gamut of most common computer monitors
- Most commonly used RGB color space, standard color space for images on the internet.
- Relatively small gamut, covers about 35% of all visible colors.

Chromaticity	Red	Green	Blue	White
x	0.6400	0.3000	0.1500	0.3127
y	0.3300	0.6000	0.0600	0.3290
Y	0.2126	0.7152	0.0722	1.0000

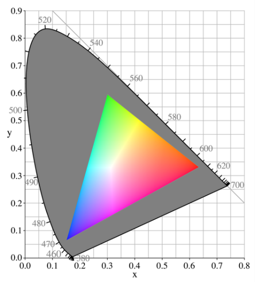


Figure 22

The CIE XYZ tristimulus values must be normalized to the D65 white point ($X = 0.9505$, $Y = 1.0000$, $Z = 1.0890$). Then compute a linear transform

$$\begin{bmatrix} R_{\text{lin}} \\ G_{\text{lin}} \\ B_{\text{lin}} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Now, these values must be *gamma-corrected* before we get the final sRGB values

$$C_{\text{srgb}} = \begin{cases} 12.92 C_{\text{linear}}, & C_{\text{linear}} \leq 0.0031308 \\ (1+a) C_{\text{linear}}^{1/2.4} - a, & C_{\text{linear}} > 0.0031308 \end{cases}$$

where $a = 0.055$, for $C \in \{R, G, B\}$.

ADOBE RGB COLOR SPACE

- Created in 1998 by Adobe to view most colors available to CMYK printers.
- Larger gamut than the sRGB, covers about 50% of all visible colors.
- Richer in the green-cyan region.

Chromaticity	Red	Green	Blue	White
x	0.6400	0.2100	0.1500	0.3127
y	0.3300	0.7100	0.0600	0.3290

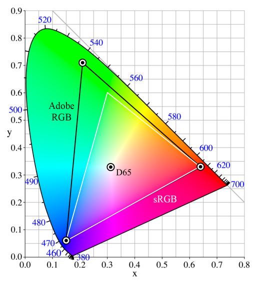


Figure 23: [1]

[1] By Mbearnstein37 - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=29994804

- **C**yan, **M**agenta, **Y**ellow, **K**ey (black)
- CMYK (in $[0, 1]$) to RGB (in $[0, 1]$):

$$R = (1 - C)(1 - K)$$
$$G = (1 - M)(1 - K)$$
$$B = (1 - Y)(1 - K)$$

- RGB (in $[0, 1]$) to CMYK (in $[0, 1]$):

$$K = 1 - \max\{R, G, B\}$$
$$C = (1 - R - K)/(1 - K)$$
$$M = (1 - G - K)(1 - K)$$
$$Y = (1 - B - K)(1 - K)$$

- The CMYK-model is subtractive
- RGB is common in display monitors, CMYK is common in printers.

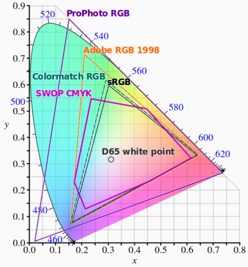


Figure 24: [1] Collection of gamuts

[1]By BenRG and cmglee - http://commons.wikimedia.org/wiki/File:CIE1931xy_blank.svg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=32158329

DEVICE DEPENDENCE

- RGB colors on a screen is dependent on the properties of the screen. That is, the same image can look different on two different screens.
- This is also the case for CMYK: the same image printed on two different printers can look completely different. The color is dependent on the printer, the printer ink, the paper etc.
- There is not allways overlap between CMYK and RGB colors. The monitor can display some colors that the printer cannot print, and vice versa.
- We say that RGB and CMYK are *device dependent* color models.
- CIEXYZ is an example of a model that is *device independent*.
- The number of stable, ”recognizable” colors on a monitor is actually quite small.

HUE, SATURATION, INTENSITY (HSI)

- Describe colors by their hue, saturation and intensity.
- This can be veiwed in a double cone (fig on the right).
    - Hue: An angle from red (H = 0).
    - Saturation: Distance from center axis.
    - Intensity: Vertical axis.
- RGB is useful for color generation. HSI more useful for color description and color image processing.

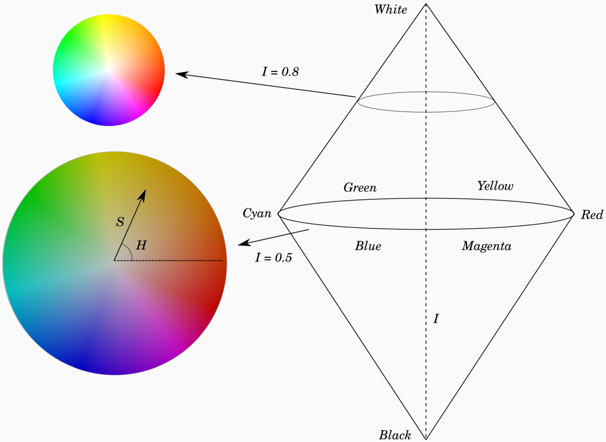


Figure 25: HSI double cone

For $R, G, B \in [0, 1]$

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases}$$

where[1]

$$\theta = \arccos\left\{\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-G)(R-B)}}\right\}.$$

Saturation is given by

$$S = 1 - \frac{3\min\{R, G, B\}}{R+G+B}.$$

Intensity is given by

$$I = \frac{1}{3}(R+G+B).$$

Notice that H is not defined for $R = G = B$, and S is not defined when $I = 0$.

[1]Remember to convert from radians to degrees. This also apply in the next slides.

HSI TO RGB

$0 < H \leq 120$:

$$R = I\left(1 + \frac{S\cos H}{\cos(60 - H)}\right)$$
$$G = 3I - (R + B)$$
$$B = I(1 - S)$$

$120 < H \leq 240$:

$$H = H - 120$$
$$R = I(1 - S)$$
$$G = I\left(1 + \frac{S\cos H}{\cos(60 - H)}\right)$$
$$B = 3I - (R + B)$$

$240 < H \leq 360$:

$$H = H - 240$$
$$R = 3I - (R + B)$$
$$G = I(1 - S)$$
$$B = I\left(1 + \frac{S\cos H}{\cos(60 - H)}\right)$$

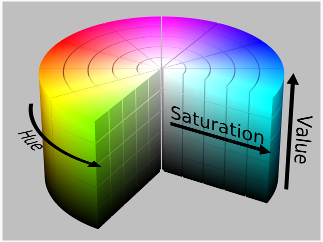
HSV (HUE, SATURATION, VALUE AND HSL (HUE, SATURATION, LIGHTNESS)

- HSV and HSL are alternatives to HSI.
- The hue the same in all three representations.
- Intensity, Value and Lightness are different

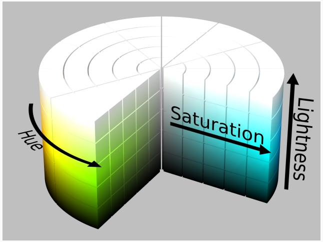
$$V = \max\{R, G, B\}$$
$$L = \frac{1}{2}(\max\{R, G, B\} + \min\{R, G, B\})$$

- Saturation is also different

$$S_{HSV} = \frac{\max\{R, G, B\} - \min\{R, G, B\}}{V}$$
$$S_{HSL} = \frac{\max\{R, G, B\} - \min\{R, G, B\}}{1 - |2L - 1|}$$



(a) HSV cylinder



(b) HSL cylinder

COMPARISON OF SOME COLORS

Note that all values are in range $[0, 1]$ except hue, which is in range $[0, 360]$

	RGB	CMY	HSI
Red	$(1, 0, 0)$	$(0, 1, 1)$	$(0, 1, 1/3)$
Yellow	$(1, 1, 0)$	$(0, 0, 1)$	$(60, 1, 2/3)$
Green	$(0, 1, 0)$	$(1, 0, 1)$	$(120, 1, 1/3)$
Blue	$(0, 0, 1)$	$(1, 1, 0)$	$(240, 1, 1/3)$
White	$(1, 1, 1)$	$(0, 0, 0)$	$(0, 0, 1)$
Gray	$(1/2, 1/2, 1/2)$	$(1/2, 1/2, 1/2)$	$(0, 0, 1/2)$
Black	$(0, 0, 0)$	$(1, 1, 1)$	$(0, 0, 0)$

LUMINANCE AND CHROMINANCE COLOR MODEL

- Several kinds: YUV, YIQ, YCbCr, YPbPr, etc.
- For YUV:
  - Y is luminance (luma).
  - U and V are chrominance.
- These kind of models are common in TV and video encoding, partly for historical reasons (compatibility with black and white TV).

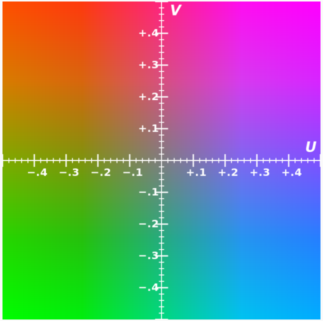


Figure 27: YUV at $Y = 0.5$.

YIQ

- NTSC is the standard for TV and video in North America and Japan, they use the system YIQ.
- Y describes luminance, I and Q describes chrominance information
- RGB to YIQ

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.522 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- YIQ to RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.623 \\ 1 & -0.272 & -0.648 \\ 1 & -1.105 & 0.705 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

$R, G, B, Y \in [0, 1]$, $I \in [-0.5957, 0.5957]$ and $Q \in [-0.5226, 0.5226]$.

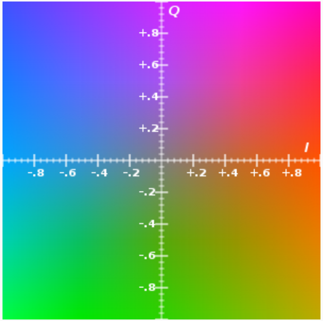


Figure 28: YIQ at $Y = 0.5$. Note that I and Q are scaled to $[-1, 1]$.

YCBCR

- This is used in digital TV and video.
- Y is luminance (luma)
- Cb is blue minus luma (B - Y)
- Br is red minus luma (R - Y)
- YCbCr is digital, RGB can both be analog or digital.
- MPEG-compression (in DVD, digital TV and video) is coded in YCbCr.
- Digital video cameras (MiniDV, DV, Digital Betacam etc) provides a YCbCr signal over a digital link like FireWire or SDI.
- The analog dual of YCbCr is YPbPr.

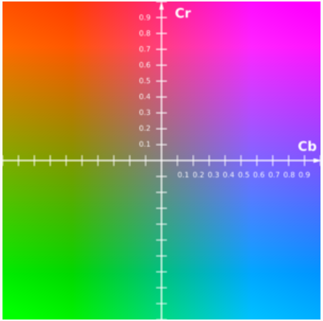


Figure 29: YCbCr at $Y = 0.5$.

COLOR SPACES, SUMMARY

- A color model is a standardized way to specify colors.
- Specidies a coordinate system where each point is a color.
- Different color models serves different purposes

RGB: color monitor display
CMYK: color printing
HSI, HSV, HSL: human color perception
YIQ, YCbCr, YUV: color compression in video and TV coding
CIEXYZ, CIELAB: device independent standard

DIGITAL COLOR IMAGES

True color uses all colors in the color space.

- Used in applications that contain many colors with subtle differences. E.g. digital photography or photorealistic rendering.
- Two main ways to organize true color: Component ordering and packed ordering.

Indexed color uses only a subset of colors.

- Application dependence on which subset to use.
- Reduces memory and computation cost.
- Used when subtle color differences is not vital.

- Each image index consist of multiple channels, one for each color component.
- For RGB, one red channel, one green channel and one blue channel.
- If each channel is discretized into 8 bits (256 intensity values for red, green and blue), we get $(2^8)^3 = 2^{24} = 16\,777\,216$ different colors

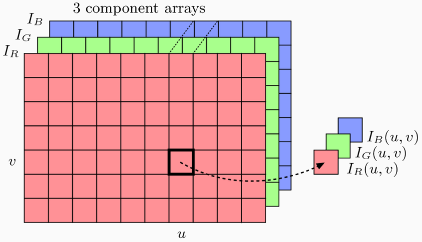
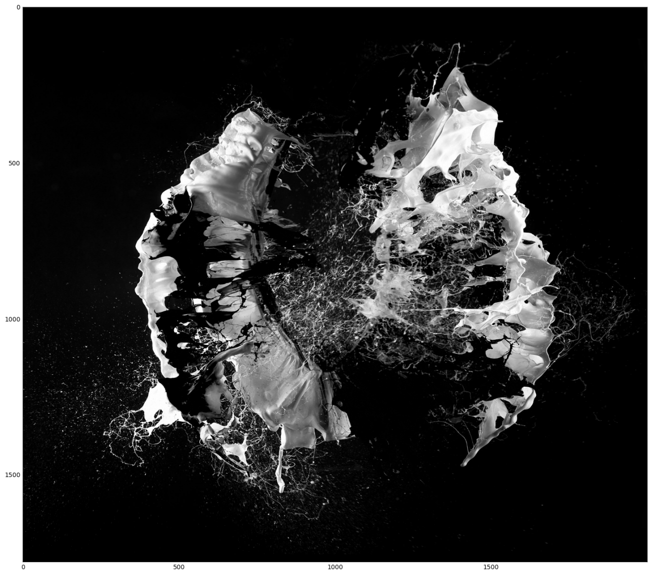


Figure 30: True color, component ordering[1].

[1]Wilhelm Burger, Mark J. Burge, *Principles of Digital Image Processing: Fundamental Techniques*, 2010

EXAMPLE

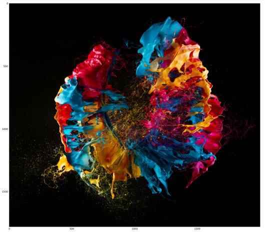
RED CHANNEL



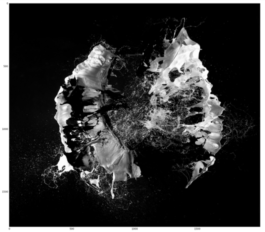
GREEN CHANNEL

BLUE CHANNEL

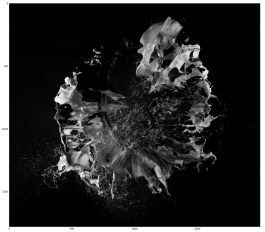
FULL EXAMPLE SUMMARY



(a) All



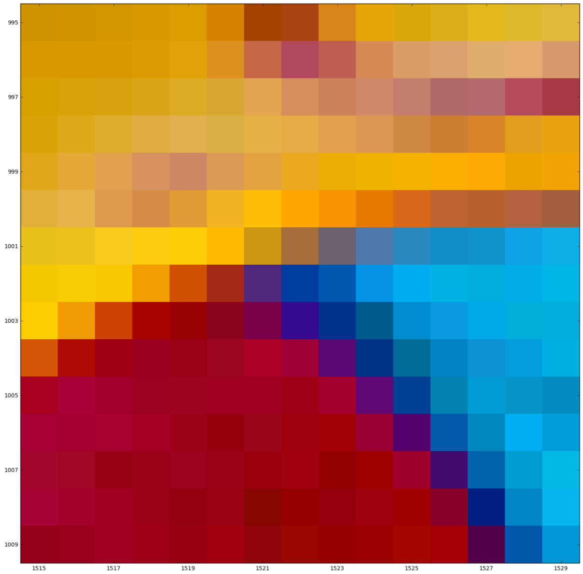
(b) Red



(c) Green

(d) Blue

EXAMPLE, DETAIL



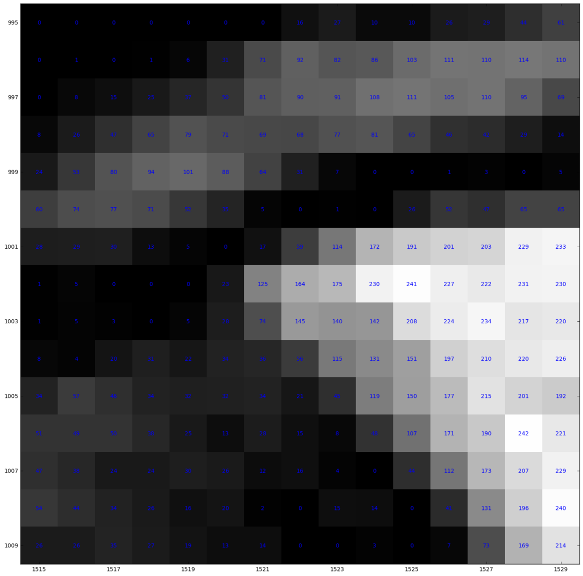
RED CHANNEL

	1515		1517		1519		1521		1523		1525		1527		1529
995	208	208	212	216	220	215	164	169	218	231	217	220	229	222	226
	216	217	217	220	227	221	198	178	190	214	220	218	224	231	215
997	215	217	215	217	222	215	227	215	206	209	195	176	181	182	168
	217	222	224	225	226	219	229	231	227	220	207	205	216	226	234
999	224	228	227	216	205	219	227	235	237	240	245	251	255	238	243
	225	233	222	214	224	241	255	255	252	231	217	190	182	181	163
1001	232	237	249	253	255	254	205	166	108	80	41	17	17	14	13
	244	248	251	243	210	164	79	0	0	4	0	0	0	0	1
1003	255	241	205	167	155	140	123	53	0	0	0	10	1	1	3
	214	175	159	154	156	157	173	160	89	0	0	0	11	5	0
1005	171	170	163	158	158	159	161	160	164	98	0	4	0	8	5
	167	165	170	166	157	150	155	158	163	152	84	6	0	0	0
1007	166	164	153	153	156	156	155	162	148	160	160	66	0	0	2
	168	163	162	158	148	154	136	148	150	158	162	136	3	0	9
1009	151	154	162	159	152	162	145	155	153	155	166	166	82	0	1

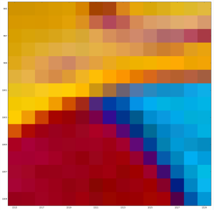
GREEN CHANNEL

	1515		1517		1519		1521		1523		1525		1527		1529
995	147	148	150	154	158	133	67	68	133	166	167	174	186	185	185
	153	154	152	155	162	146	103	72	94	138	157	162	173	172	154
997	160	164	163	166	172	168	167	144	131	138	126	104	106	75	58
	164	171	175	174	176	176	178	172	161	152	137	128	132	159	164
999	169	169	162	146	135	154	161	169	174	179	180	175	171	165	165
	178	180	155	139	155	181	191	168	148	122	104	99	96	96	94
1001	195	194	203	204	206	187	151	110	97	120	138	143	146	163	176
	201	205	201	158	81	41	39	61	88	146	174	180	175	174	181
1003	206	156	65	4	2	4	0	11	52	90	141	154	169	174	174
	84	10	0	1	0	7	0	0	7	50	107	132	146	157	177
1005	0	0	0	0	4	0	2	1	0	9	65	129	155	148	139
	0	0	0	0	1	0	6	0	2	0	1	91	135	175	158
1007	7	7	1	1	2	0	0	0	4	1	0	11	101	155	188
	4	2	0	0	3	3	7	0	0	0	0	1	31	134	182
1009	0	0	3	0	0	0	5	7	0	0	7	1	0	86	151

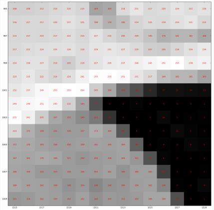
BLUE CHANNEL



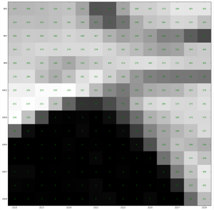
DETAIL EXAMPLE SUMMARY



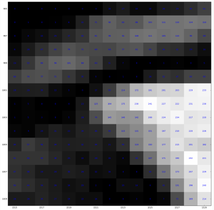
(a) All



(b) Red



(c) Green



(d) Blue

TRUE COLOR: PACKED ORDERING

- Color components are packed together at the same image element.

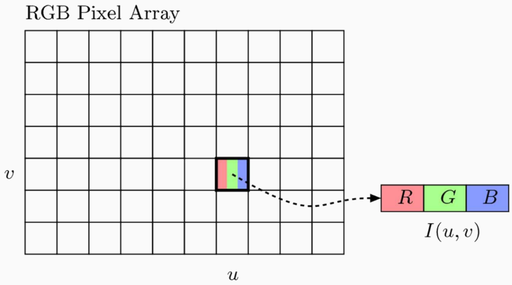


Figure 33: True color, packed ordering[1].

[1]Wilhelm Burger, Mark J. Burge, *Principles of Digital Image Processing: Fundamental Techniques*, 2010

- Endianness: Sequential order used to numerically interpret a range of bytes in computer memory as a larger, composed word value.
    - Little endian: Bytes are stored by increasing significance, with the least significant byte (LSB) [1] stored first. Common in microprocessors (e.g. Intel x86 processors).
    - Big endian: Bytes are stored by decreasing significance, with the most significant byte (MSB)[2] stored first. Common in data networking (e.g. in the Internet Protocol suite).
- A 32 bit word $ARGB$ with decreasing significance $A \rightarrow R \rightarrow G \rightarrow B$ would be stored as $ARGB$ in a big-endian system, but as $BGRA$ in a little-endian system.

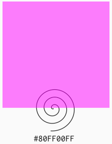
[1]The byte containing the least significant (”the rightmost”) bit.
[2]The byte containing the most significant (”the leftmost”) bit.

EXAMPLE: ENDIANNESS CONFUSION

A 32 bit ARGB value `0x80FF00FF` would be interpreted differently by big-endian and little-endian systems if not handeled properly.

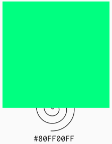
Big-endian interpretation: `0x80FF00FF`.

	Hex	Decimal (8-bit)
Alpha	80	128 (50.2 %)
Red	FF	255
Green	00	0
Blue	FF	255



Little-endian interpretation: `0xFF00FF80`.

	Hex	Decimal (8-bit)
Alpha	FF	255 (100 %)
Red	00	0
Green	FF	255
Blue	80	128



INDEXED COLOR

- An image contains indices of a look up table (LUT) of colors (palette) in stead of color intensity values.
- Permits only a limited number of colors.
- Often are these images stored in indexed GIF or PNG formats.
- Use color quantization to transform optimally from a true color image to indexed color image.

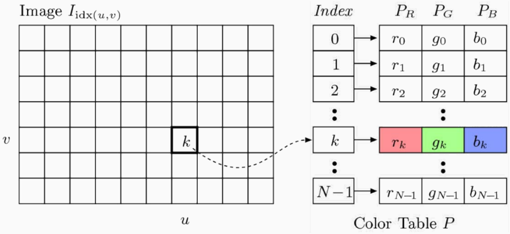


Figure 34: Indexed color ordering[1].

[1] Wilhelm Burger, Mark J. Burge, *Principles of Digital Image Processing: Fundamental Techniques*, 2010

COLOR IMAGES AND LOOK UP TABLES

- As we have seen, in true color, an RGB pixel can be stored in 24 bits (8 for each color).
- To reduce this size, we could e.g. assign 3 bits to each color.
- This would result in only 512 different possible colors, and a total of 9 bits ber pixel.
- A region with many nuances of a color would not loog good.
- It is certain that all 512 colors exist in the image.
- Alternatively, one could use 8 bits and a LUT.
- Each row in the table represent a 24 bit RGB color.
- The table consist of the 256 colors that best represent the image.

COLOR QUANTIZATION

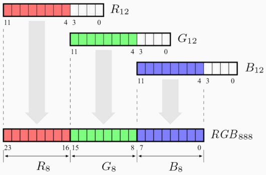
- Purpose: reduce size of color images.
- E.g. 24 bit TIFF to 8 bit TIFF.
- Replace the true color with a best match from a smaller subset.
- Some different quantization algorithms:
    - Uniform quantization
    - Median-cut algorithm

Convert each component c of the original RGB value independently and uniformly to the new value $\hat{c}$

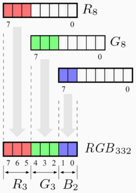
$$\hat{c} = \text{floor}\left\{c\frac{N}{\hat{N}}\right\}$$

Here $c \in \{r, g, b\}$ and N is the number of intensity values in the respective representation.

Example: 3×12 bit ($N = 2^{12} = 4096$) true color image to 3×8 bit ($\hat{N} = 2^8 = 256$) value:



Example: 3×8 bit ($N = 256$) to a $3:3:2$ packed 8 bit: 3 bits for red ($\hat{N} = 8$), 3 bits for green ($\hat{N} = 8$), and 2 bits for blue ($\hat{N} = 4$).



This transforms a 24 bit true color image to 8 bit indexed color image.

1. Find the box in the RGB space that cover all colors present in the image.
2. Sort the colors in the box along the longest RGB dimension of the box. This is done by computing the color histogram.
3. Split the box in two at the median in the sorted list.
4. Repeat step 2 and 3 for all boxes (including the new ones that you create). Repeat until you have 256 boxes.
5. For each box, compute the mean RGB value in that box, and let this value represent the value of the box.
6. Map each 3×8 bit RGB value in the original image to the index of the box which value is the closest in the RGB space.

ENDIANNESS AND LUT

A LUT is also prone to confusion in endianness. For instance, a LUT with 16 bit values and a $5:6:5$ bit packed RGB ordering, (from MSB to LSB) `1000010000010000` would be interpreted differently. Let "|" represent a byte delimeter, then we would have a:

Big endian interpretation

`10000100 | 00010000`

Component	Binary	Decimal
Red	`10000`	$16\ (\sim 50\%)$
Blue	`100000`	$32\ (\sim 50\%)$
Green	`10000`	$16\ (\sim 50\%)$

Little endian interpretation

`00010000 | 10000100`

Component	Binary	Decimal
Red	`00010`	$2\ (\sim 6.25\%)$
Blue	`000100`	$4\ (\sim 6.25\%)$
Green	`00100`	$4\ (\sim 12.5\%)$

PSEUDO COLORS

- Pseudo-color images can be graylevel images where each graylevel is assigned an RGB value according to some LUT.
- Is often used to emphasize small graylevel differences (e.g. in medical imaging).
- Also often used in graphical display of data.
- If the color-LUT is mapped back to graylevels, the intensity should(!) be correct.
- Colormaps in plotting libraries are often pseudo-color LUT.

- Often, we want the colormap to be *perceptually uniform*: Equal steps in the data are percieved as equal steps in the color space.
- The human brain percieves changes in lightness as changes in data better than changes in hue.
- Up until recently, Matlab (until 2014) and Python's Matplotlib (until 2.0 release) used `jet` as the default colormap for data display[1].
- `jet` is a rainbow colormap that is not perceptually uniform.

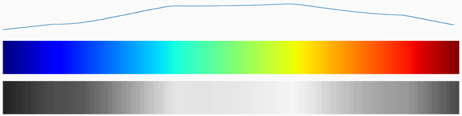
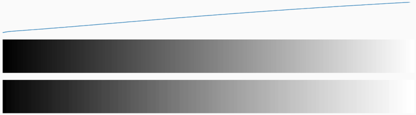


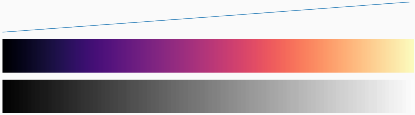
Figure 35: A gradient of `jet`, and the lightness component of the `CAM02-UCS` colorspace.

[1] See a more detailed wrap-up here: `https://bids.github.io/colormap/`

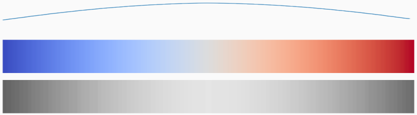
COMPARISON OF SOME COLORMAPS



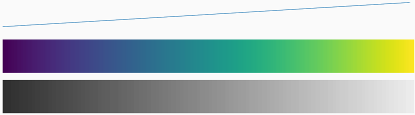
(a) `gray` (sequential)



(b) `magma` (sequential perceptual uniform)



(c) `coolwarm` (diverging)



(d) `viridis` (sequential perceptual uniform) (new matplotlib default)

- We can produce rasterized data based on observations, simulations, computations etc.
- For instance population density or precipitation (nedbør) data projected on a world map.
- The use of a LUT gives a graphical display that is *not* created by imaging.

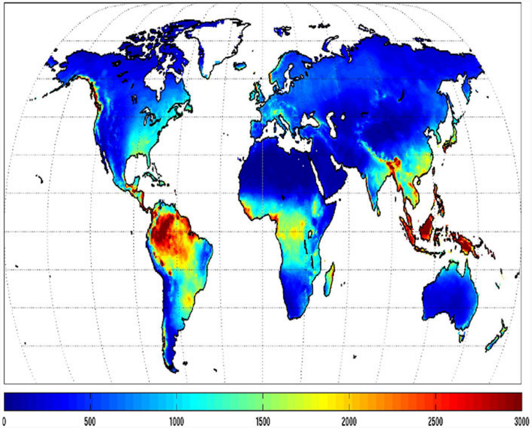


Figure 37: Example of global mean annual (mm/year) percipitation (notice the colormap). (Source: `https://data.giss.nasa.gov/impacts/agmipcf/`)

Using imaging from outside the visible spectrum and mapping it to RGB.
Example: NOAA (*National Oceanic and Atmospheric Administration*) satelites equipped with a AVHRR (*Advanced Very High Resolution Radiometer*), an instrument sensing in the visual and infrared part of the EM spectrum[1].

Band	Band width	Applications
1 (visible)	0.58 µm - 0.68 µm	Clouds and land surfaces cartography (day)
2 (near IR)	0.725 µm - 1.00 µm	Clouds and land surfaces cartography (day)
3A (near IR)	1.580 µm - 1.64 µm	Snow and ice detection
3B (IR)	3.550 µm - 3.93 µm	Clouds and sea surface temperature mapping (night)
4 (IR)	10.30 µm - 11.30 µm	Clouds and sea surface temperature mapping (night)
5 (IR)	11.50 µm - 12.50 µm	Sea surface temperature

[1] `http://eoedu.belspo.be/en/satellites/noaa.htm`

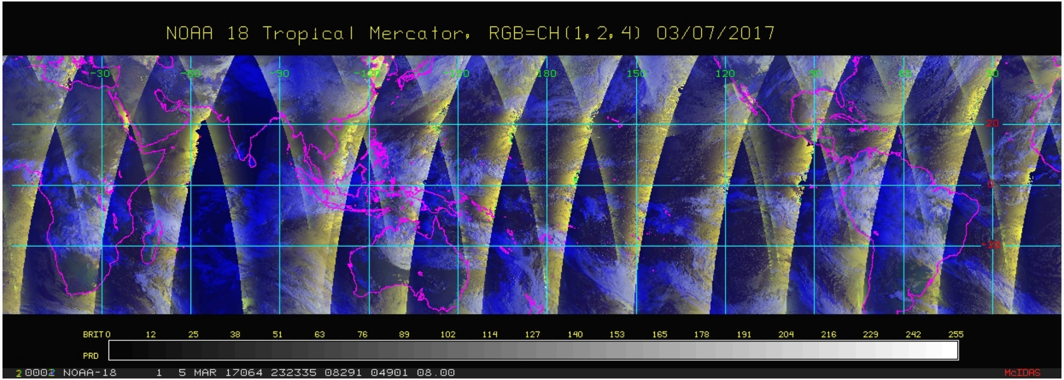


Figure 38: Composite mapped mosaics, band 1, 2, 4 [1]

[1] http://www.ssd.noaa.gov/POES/COMP/

QUANTIZATION AND PRINTING

GRAYSCALE PRINTING

- Printers print gralevels in binary (black or nothing).
- We can remediate this problem by introducing a finer mesh.
- That is, the printer uses *halftones*.
- This works since the we percieve a mean of close intensity values.
- The challenge is to create patterns of binary pixels that represent a gray level.
- Several methods exist:
    - Global threshold (be smarter!)
    - *Patterning*
    - Ordered dithering
    - Error diffusion

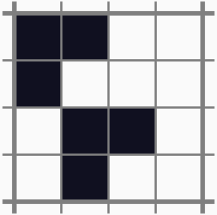


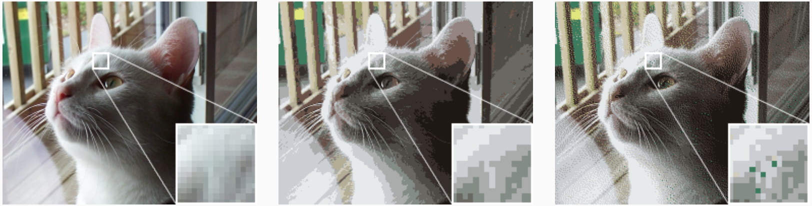
Figure 39: One pixel partitioned into 4×4 uniform sub-pixels.

(a) Original

(b) Result

DITHERING

- Dithering is a method used to create an illusion of color.
- Often used to remedy color quantization.
- There exist several dithering algorithms, we will cover
  - Thresholding (technically a dithering method)
  - Ordered dithering
  - Error diffusion dithering



(a) Original (b) 16 colors, no dithering (c) 16 colors, with dithering

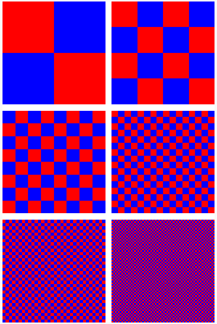


Figure 42

ORDERED DITHERING

- We threshold the values in an image using a dither matrix.
- A dither matrix D_n
  - Has shape $2^n \times 2^n$
  - Partitions the graylevel scale $[0, 255]$ into $(2^n)^2$ equidistant values.
- Example:

$$D_2 = \begin{bmatrix} 0 & 128 & 32 & 160 \\ 192 & 64 & 224 & 96 \\ 48 & 176 & 16 & 144 \\ 240 & 112 & 208 & 80 \end{bmatrix}$$

- Simulate the subpixel partition by scaling the image up with a factor 2^n.
- Use the dither matrix as a mask and ”place it on the original pixels”.
- Threshold the ”subpixel value” against the corresponding element in the dither matrix.
- That is, let $I \in [0, 255]^{h\times w}$ be the original image, and $I_n \in \{0, 255\}^{2^n h\times 2^n w}$ be the new, dithered image, then

$$I_n[i, j] = \begin{cases} 255, & \text{if } I[\lfloor \frac{i}{2^n} \rfloor, \lfloor \frac{j}{2^n} \rfloor] > D_n[i \mod 2^n] \\ 0, & \text{if } I[\lfloor \frac{i}{2^n} \rfloor, \lfloor \frac{j}{2^n} \rfloor] \leq D_n[i \mod 2^n] \end{cases}$$

for $i \in [0, 2^n h - 1]$ and $j \in [0, 2^n w - 1]$.

EXAMPLE OF ORDERED DITHERING USING D_2

(a) Original

(b) Result

- Used to reduce quantization levels in an image.
- Distributes the quantization error in made in one pixel to nearby pixels.
- Example: Floyd-Steinberg dithering
- Each traversed pixel is assigned the value that is closest in the palette.
- The residual is then weighted and added to the nearby pixels as

$$\begin{bmatrix} p & c & \frac{3}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{bmatrix}$$

where p signifies the previously visited pixel, and c the current one.

```
# Floyd–Steinberg on an uint8 image im
palette = [0, 255]
m, n = im.shape
new_im = im.copy()

for i in range(m):
  for j in range(n):
    old_val = new_im[i, j]
    new_val = nearest_val(old_val, palette)
    new_im[i, j] = new_val
    quant_error = old_val – new_val
    # Boundary treatement ignored for
     readability
    new_im[i+1, j–1] += int(quant_error*3/16)
    new_im[i+1, j] += int(quant_error*5/16)
    new_im[i+1, j+1] += int(quant_error*1/16)
    new_im[i, j+1] += int(quant_error*7/16)

return new_im.astype(np.uint8)
```

ERROR DIFFUSION EXAMPLE

(a) Original

(b) Result

- A CMYK color model is used.
- Halftone patterns are used at certain angles (different for each color) to create color patterns.
- We percieve the result in such a way that no sharp color transitions are seen.

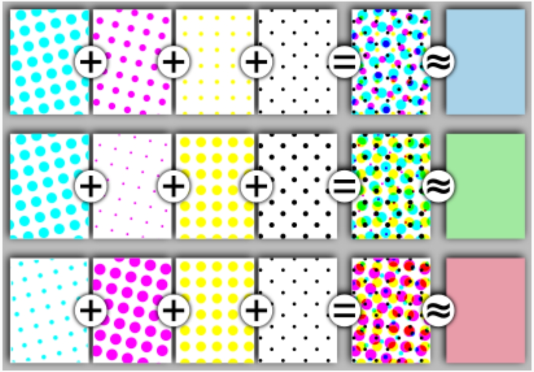


Figure 45: Three examples of modern color halftoning.[1]

- An image with 3 color channels has a 3D cube as histogram
- With one bin for each color value, a histogram gets of an 8-bit RGB image gets $(2^8)^3 = 16777216$ bins.
- A 1024×1024 image can maximally fill $1/16$ of this cube $(2^{2 \cdot 10}/2^{24} = 2^{-4} = 1/16$.
- In other words: the cube is mostly empty.
- For this reason, it is most common to project the histogram to 1D or 2D.
    - 1D: Projection to the R-, G-, or B-axis.
    - 2D: Projection to the RG-, RB-, or GB-plane.

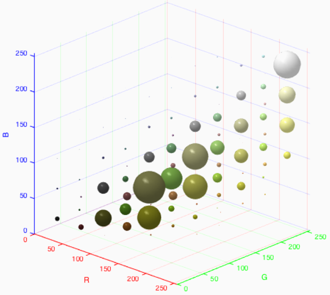
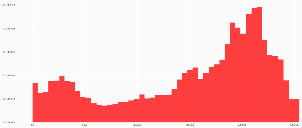


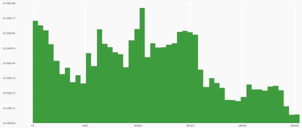
Figure 46: 3D histogram example

COMPONENT COLOR HISTOGRAM

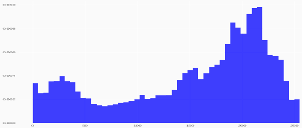
Figure 47: Lena



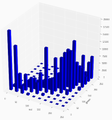
(a) Red



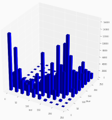
(b) Green



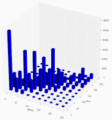
(c) Blue



(a) Red and green



(b) Red and blue



(c) Green and blue

HISTOGRAM EQUALIZATION IN RGB

- Histogram equalization on each RGB component, independently.
- This often produce a bad result.
- It is better to do it in HSI:
    1. Transform the image from RGB to HSI.
    2. Do histogram equalization on the *I*-component.
    3. Transform the new HSI image back to RGB.

- The H and S channel are not changed.
- But since the I channel is, color perception can be altered.
- To remediate this, you can adjust the saturation S before you transform back to RGB.

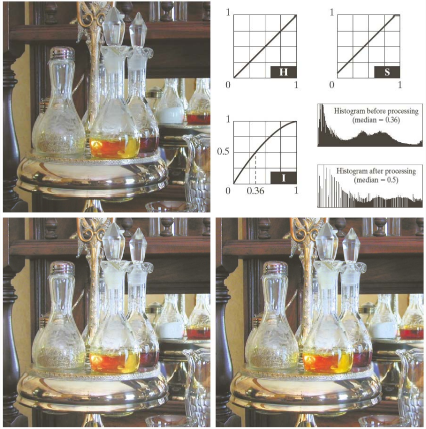
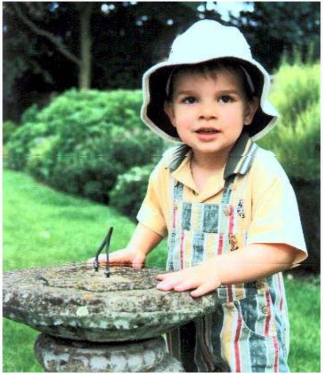


Figure 50: Histogram equalization followed by saturation adjustment in HSI.

(a) Original

(b) RGB

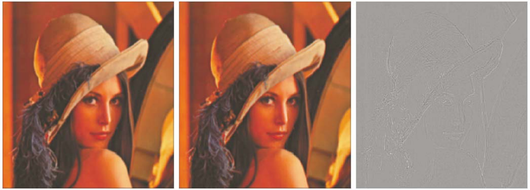


(c) HSI

- RGB-filtering blurs colors.
- Filtering of the I component in HSI produces a smooth image, without color adjustment.

Figure 52: Lena and RGB components.

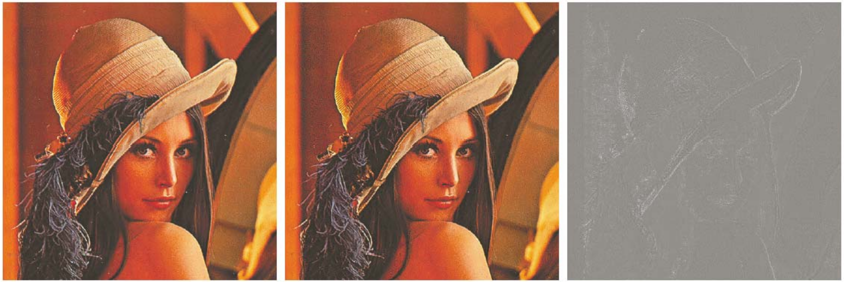
(a) HSI components



(b) Left: RGB lowpass. Middle: Filtering I component and converting back to HSI. Right: Difference between results.

LAPLACE FILTERING

- We can make a graylevel image appear sharper by adding a scaled laplace of the same image (previous lecture).
- RGB:
  - Compute the laplace of each RGB component, and add it to the respective component.
  - The color of each pixel is then influenced by the color of neighbouring pixels.
- HSI:
  - Compute the laplace of the intensity channel, and add it to the intensity channel.
  - The color is preserved, but the intensity near edges is changed.



COLOR IMAGE THRESHOLDING

- Suppose that we have observed the same scene in different wavelengths.
- We can then threshold based on:
  - 2D histogram
  - 3D histogram
  - Higher order histograms
- Simple method:
  1. Decide thresholds for each channel independently.
  2. Combine the segmented channels into one image.
- For RGB, this corresponds to partition the RGB space into a boxes.

A bit more sophisticated method.

1. Choose an arbitrary point in the multidimensional color-space as reference, e.g. (R_0, G_0, B_0) in RGB space.
2. Let f_R, f_G, f_B be the different color components of an RGB image.
3. Compute the distance based on the reference point

$$d(x,y) = \sqrt{(f_R[x,y] - R_0])^2 + (f_G[x,y] - G_0])^2 + (f_B[x,y] - B_0])^2}$$

4. Then, compute the final segmentation g as

$$g[x,y] = \begin{cases} 1, & \text{if } d(x,y) \leq d_{max} \\ 0, & \text{if } d(x,y) > d_{max} \end{cases}$$

   for some threshold d_{max}.
5. This is then a sphere with radius d_{max} around the reference point.

We can do the same, but with an ellipse in RGB space in stead of a sphere.

1. Choose an arbitrary point in the multidimensional color-space as reference, e.g. (R_0, G_0, B_0) in RGB space.
2. Choose distance thresholds d_R, d_G, d_B.
3. Compute the distance based on the reference point

$$d(x,y) = \sqrt{\frac{(f_R[x,y]-R_0])^2}{d_R^2} + \frac{(f_G[x,y]-G_0])^2}{d_G^2} + \frac{(f_B[x,y]-B_0])^2}{d_B^2}}$$

4. Then, compute the final segmentation g as

$$g[x,y] = \begin{cases} 1, & \text{if } d(x,y) \leq 1 \\ 0, & \text{if } d(x,y) > 1 \end{cases}$$

for some threshold d_{max}.

THRESHOLDING HSI

- Transform to to HSI.
- Suppose that we want to segment parts in an image
  - with a certain hue
  - and above some saturation threshold.
- Create a mask by segmenting the saturation image.
- Multiply the hue component image with this mask.
- Chose a hue interval corresponding to the desired color (remember that hue is circular).

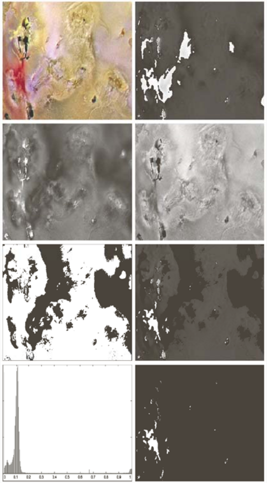


Figure 55: Image segmentation in HSI space (row, colum): (1, 1): Original. (1, 2): Hue. (2, 1): Saturation. (2, 2): Intensity. (3, 1): Binary saturation mask. (3, 2): Product of (1, 2) and (3, 1). (4, 1): Histogram. (4, 2) Segmentation of red components in (1, 1).

$$F = \sqrt{\frac{1}{2}(g_{xx} + g_{yy}) + (g_{xx} - g_{yy})\cos(2\theta) + 2g_{xx}\sin(2\theta)}$$

where

$$g_{xx} = \left(\frac{\partial f_R}{\partial x}\right)^2 + \left(\frac{\partial f_G}{\partial x}\right)^2 + \left(\frac{\partial f_B}{\partial x}\right)^2,$$

$$g_{yy} = \left(\frac{\partial f_R}{\partial y}\right)^2 + \left(\frac{\partial f_G}{\partial y}\right)^2 + \left(\frac{\partial f_B}{\partial y}\right)^2,$$

$$g_{xy} = \frac{\partial f_R}{\partial x}\frac{\partial f_R}{\partial y} + \frac{\partial f_R}{\partial x}\frac{\partial f_R}{\partial y} + \frac{\partial f_R}{\partial x}\frac{\partial f_R}{\partial y},$$

and

$$\theta = \frac{1}{2}\arctan\left(\frac{2g_{xy}}{g_{xx} - g_{yy}}\right).$$

Figure 56: RGB edge detection. (1, 1): Original. (1, 2): Gradient in RGB color vector space (F). (2, 1): Gradients computed on per RGB component, and then added. (2, 2): Difference between (1, 2) and (2, 1).

- Add gaussian noise on each RGB component ($\mu = 0$, $\sigma^2 = 800$).
- The noise is not that visible in the RGB image.
- Convert the noisy image to HSI.
- The hue and saturation channels are very noisy.
- The intensity channel is less noisy than the RGB channels.

(a) Noisy RGB components

(b) Noisy HSI components

Questions?