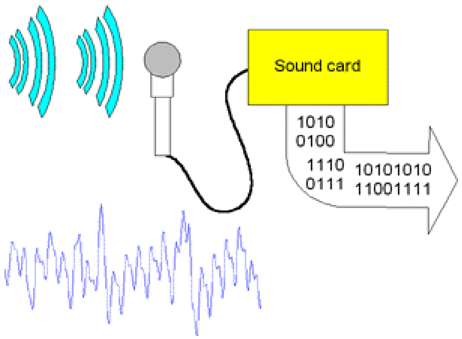
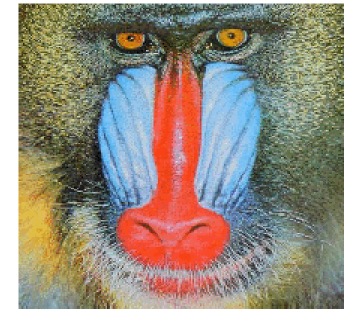
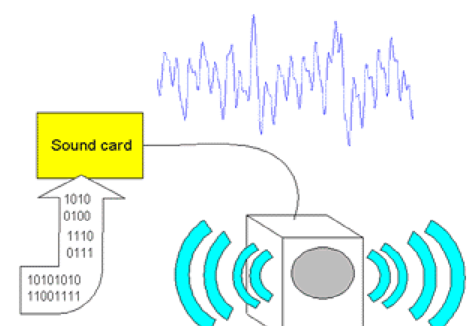




Color Science and Image Representation

IT6902; Semester B, 2004/2005; Leung

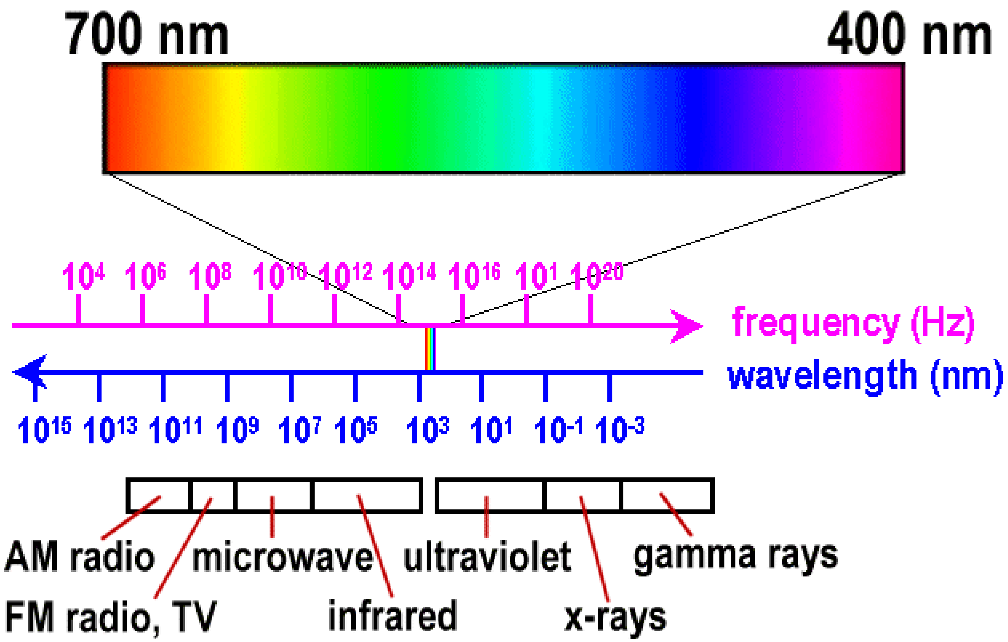


Color Science

- Introduction
  - Light and Spectra
  - Human Vision
  - Spectral Sensitivity of the Eye
  - Image Formation
- Color Representation and Specification
  - Color-Matching Functions
  - CIE Chromaticity Diagram
  - Out-of-Gamut Colors
  - Color Monitor Specification
  - Subtractive Color: CMY Color Model
- Perception of Color
  - Simultaneous Contrast
  - Mach Bands

Light and Spectra

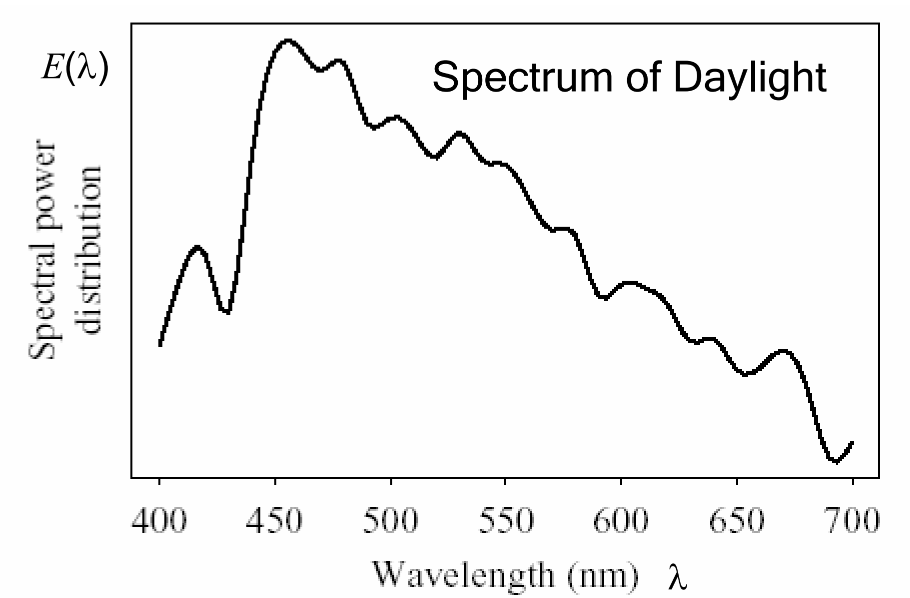
- Light is an electromagnetic (EM) wave
- Visible light ranges from 400nm to 700nm



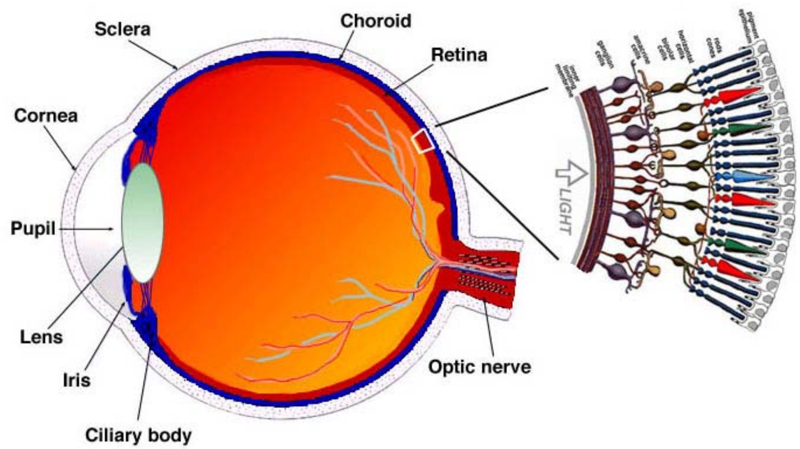
Source: http://escience.anu.edu.au/lecture/cg/Color/visibleSpectrum.en.html

Light and Spectra (2)

- Spectral Power Distribution (SPD), or *spectrum*, $E(\lambda)$, shows the relative amount of light energy at each wavelength λ



Human Vision



. A drawing of a section through the human eye with a schematic enlargement of the retina.

Source: http://webvision.med.utah.edu/sretina.html

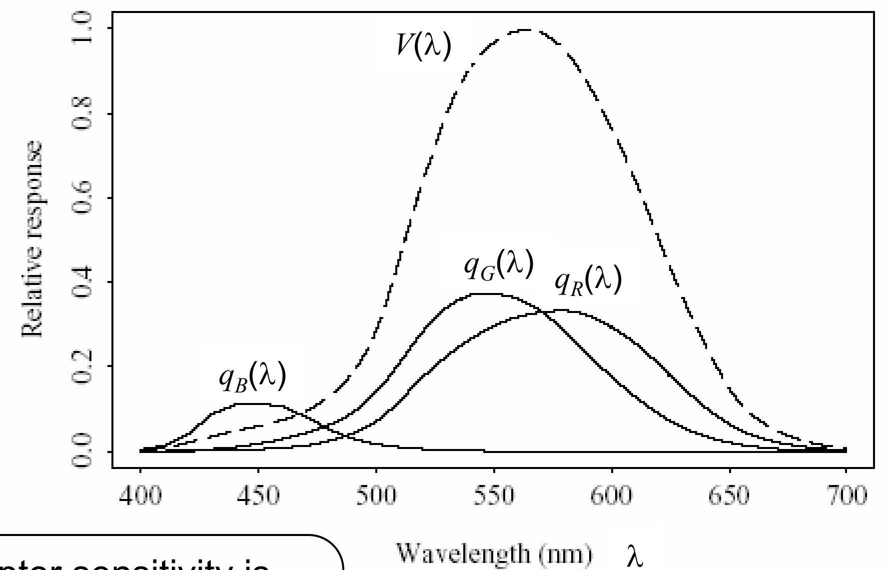
- Retina consists of an array of rods and three kinds of cones
  - Rods are for night vision
  - Cones are for color vision

Three kinds of cones:

1. L-cone: most sensitive to red light
2. M-cone: most sensitive to green light
3. S-cone: most sensitive to blue light

Spectral Sensitivity of the Eye

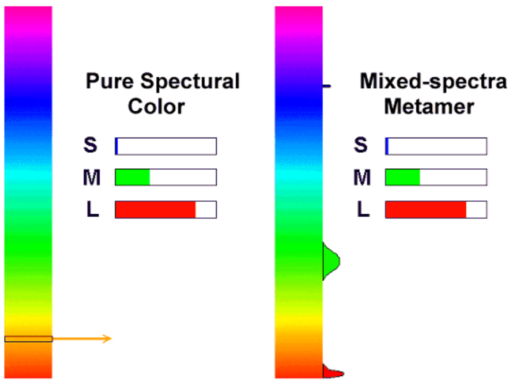
- The eye is most sensitive to light in the middle of the visible spectrum.
- The response in each color channel in the eye is proportional to the number of neurons ring.
  - Red Receptor Sensitivity $q_R(\lambda)$
  - Green Receptor Sensitivity $q_G(\lambda)$
  - Blue Receptor Sensitivity $q_B(\lambda)$
- Luminous-efficiency function $V(\lambda)$: overall sensitivity that is formed by the sum of the response curves for Red, Green and Blue



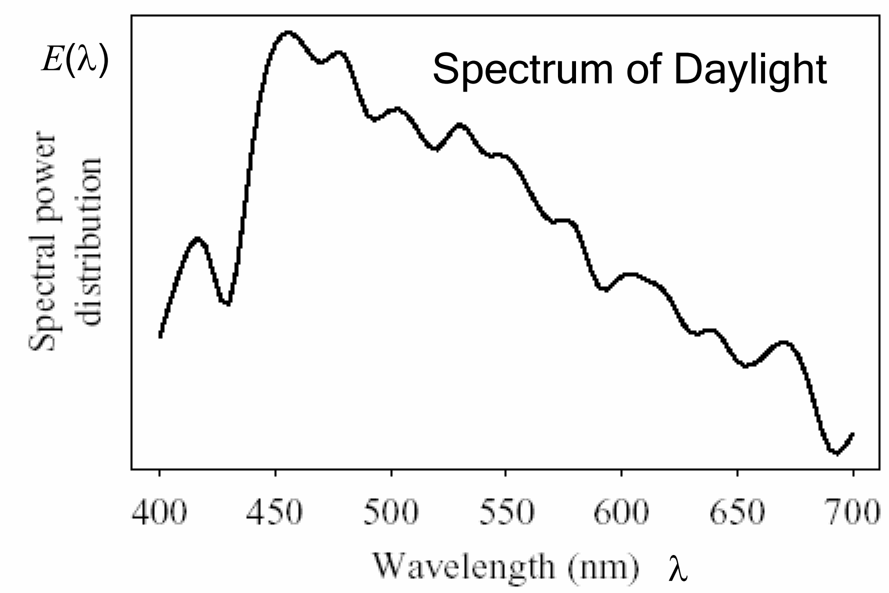
The Blue receptor sensitivity is not shown to scale because it is much smaller than the curves for Red or Green.

Spectral Sensitivity of the Eye (2)

- Total response on each channel is given by:
  - $R = \int E(\lambda)\, q_R(\lambda)\, d\lambda$
  - $G = \int E(\lambda)\, q_G(\lambda)\, d\lambda$
  - $B = \int E(\lambda)\, q_B(\lambda)\, d\lambda$



Source: http://escience.anu.edu.au/lecture/cg/Color/colorPerception.en.html



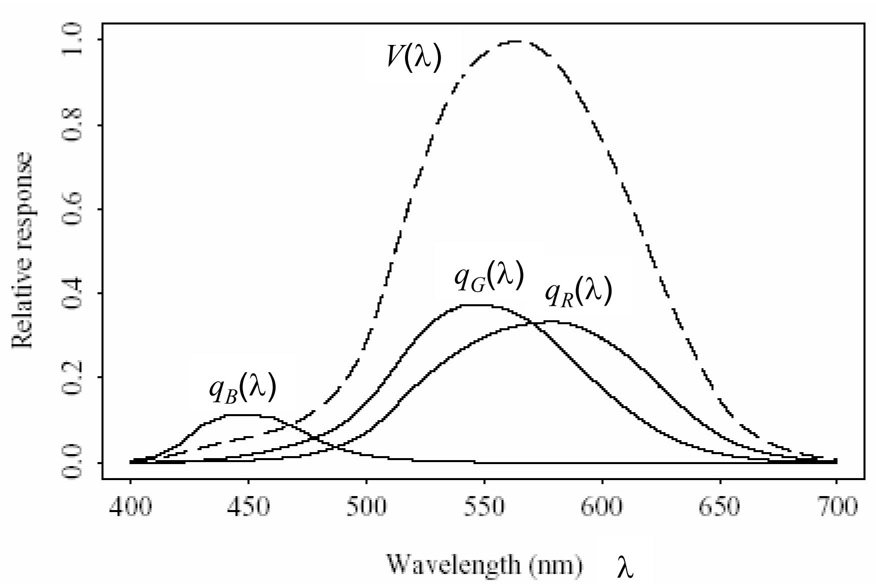
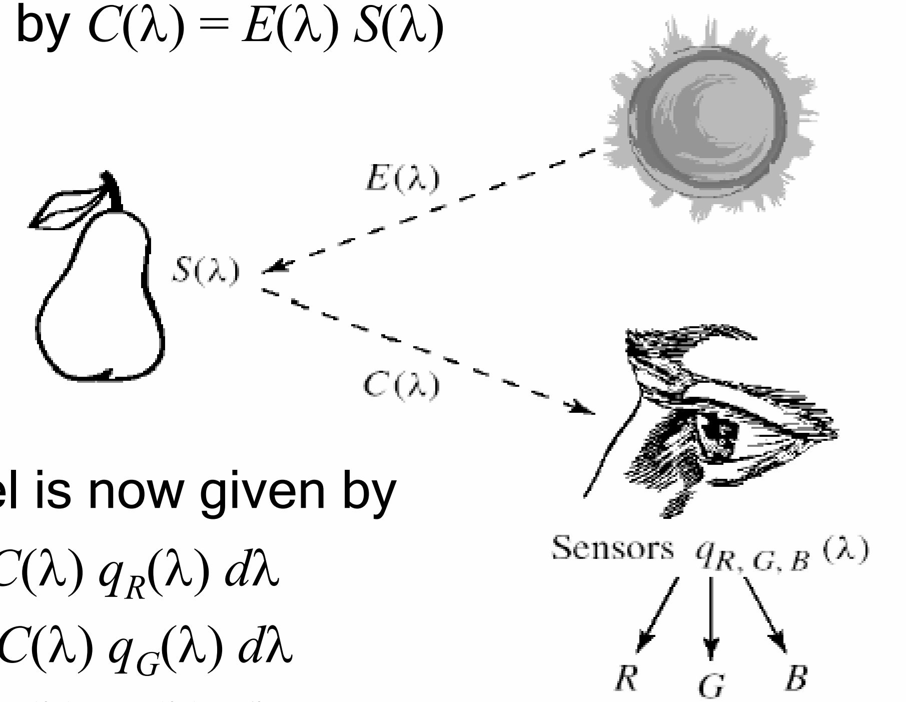


Image Formation

- Light from the illuminant (light source) with SPD $E(\lambda)$ impinges on a surface, with surface spectral reflectance function $S(\lambda)$, is reflected, and then is filtered by the eye's cone functions $q(\lambda)$.
- The *color signal* $C(\lambda)$, is defined by $C(\lambda) = E(\lambda)\, S(\lambda)$

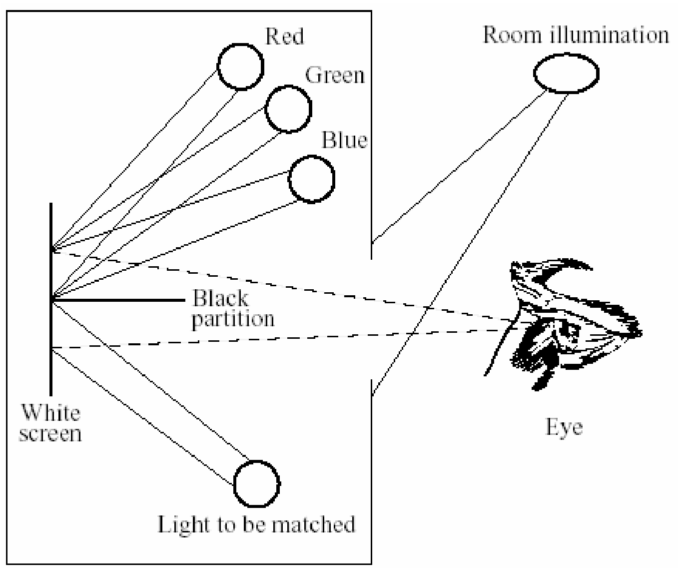


Total response on each channel is now given by

$$R = \int E(\lambda)\, S(\lambda)\, q_R(\lambda)\, d\lambda = \int C(\lambda)\, q_R(\lambda)\, d\lambda$$
$$G = \int E(\lambda)\, S(\lambda)\, q_G(\lambda)\, d\lambda = \int C(\lambda)\, q_G(\lambda)\, d\lambda$$
$$B = \int E(\lambda)\, S(\lambda)\, q_B(\lambda)\, d\lambda = \int C(\lambda)\, q_B(\lambda)\, d\lambda$$

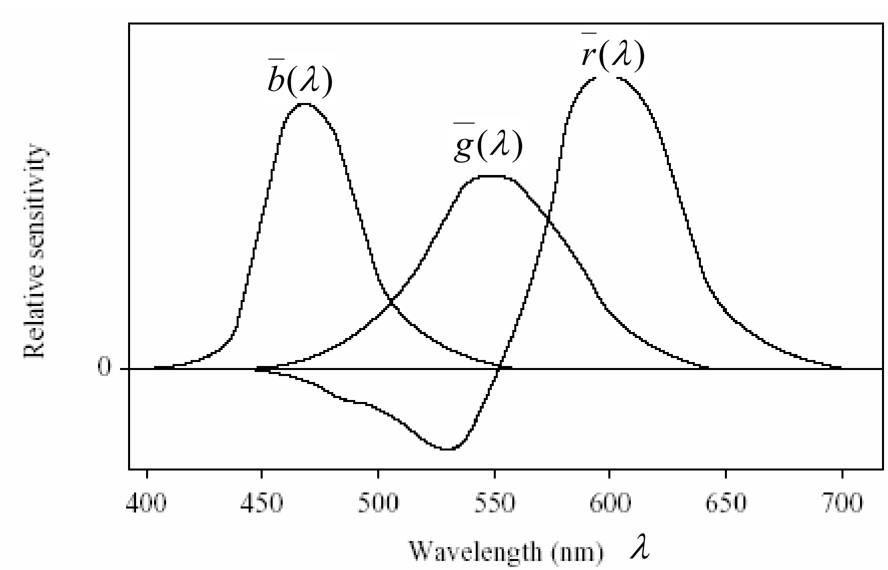
Color-Matching Functions

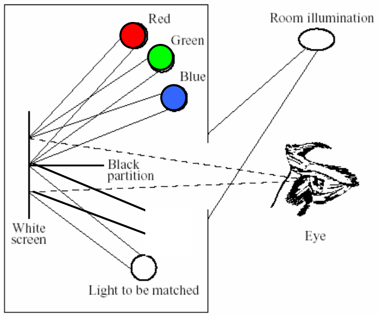
- The particular set of three basic lights R,G,B used in this experiment is called the set of color primaries.
- Color-matching experiment: To match a given color, a subject is asked to separately adjust the brightness of the three primaries using a set of controls until the resulting spot of light matches most closely with the desired color.
- Colorimeter: A device for carrying out the color-matching experiment



Color-Matching Functions (2)

- The amounts of R, G, and B the subject selects to match each single-wavelength light forms the *color-matching curves*.
- Color primaries: peaks at 440nm, 545nm and 580nm





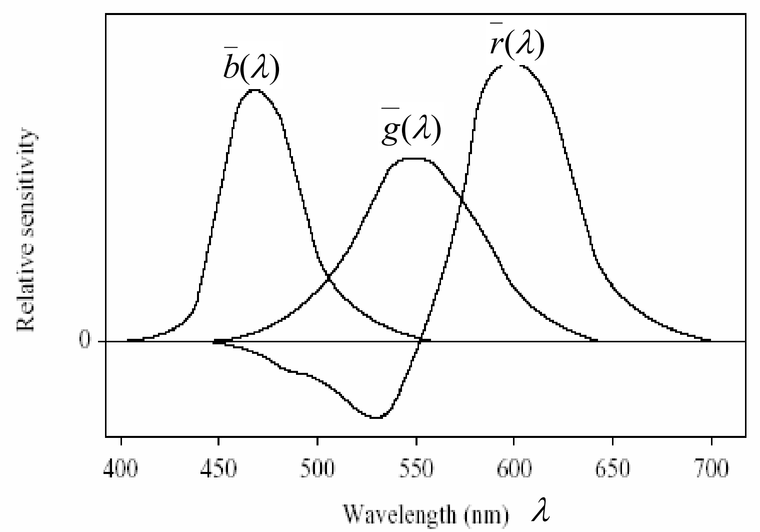
CIE RGB color-matching functions $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$

Commission Internationale de L'Eclairage

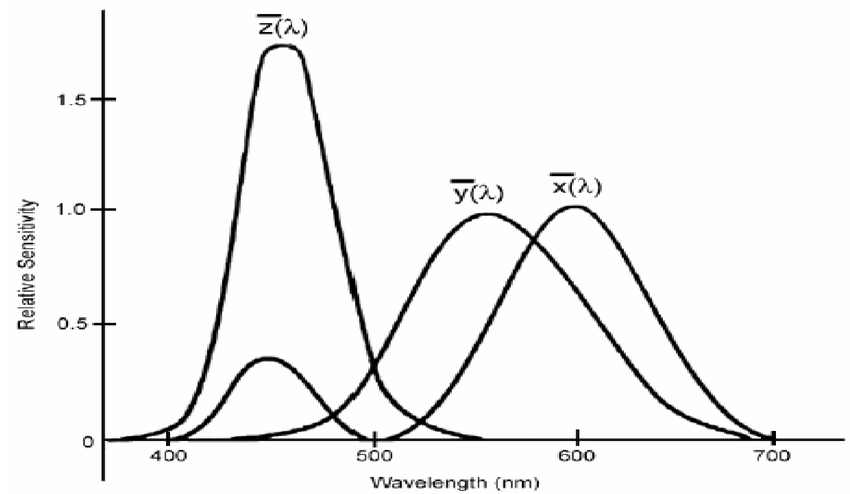
The negative part of the curve Indicates that some color cannot be reproduced by a linear combination of the primaries. For such color, one or more primary lights has to be shifted from one side to the other

CIE Chromaticity Diagram

- Since the $\bar{r}(\lambda)$ color-matching curve has a negative lobe, a set of fictitious primaries were devised that lead to color-matching functions with only positives values.
- They are a 3×3 matrix away from $\bar{r}, \bar{g}, \bar{b}$ curves, and are denoted $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$



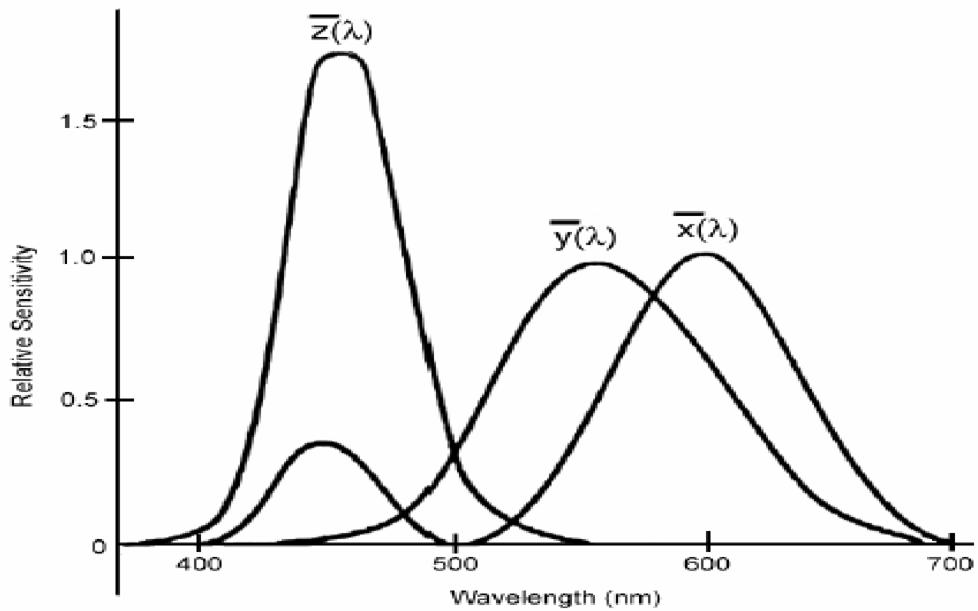
CIE RGB color-matching functions



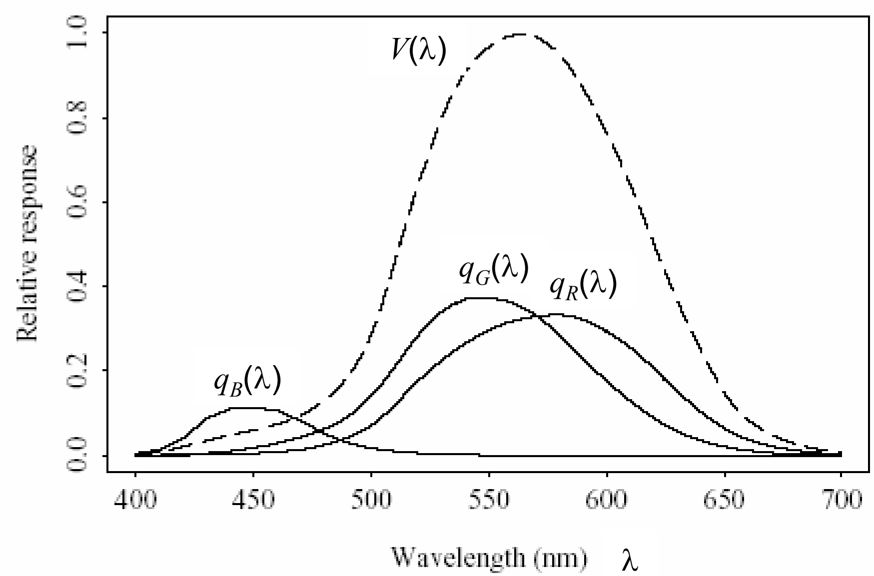
CIE standard color-matching functions

CIE Chromaticity Diagram (2)

- The matrix is chosen such that the middle standard color-matching function $\bar{y}(\lambda)$ exactly equals the luminous-efficiency curve $V(\lambda)$



CIE standard color-matching functions



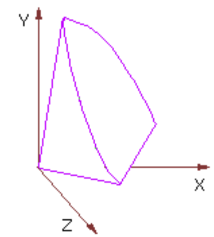
CIE Chromaticity Diagram (3)

- For a general SPD $E(\lambda)$, the essential "colorimetric" information required to characterize a color is the set of *tristimulus values* X,Y,Z (Y == luminance):

$$X = \int E(\lambda)\bar{x}(\lambda)d\lambda$$

$$Y = \int E(\lambda)\bar{y}(\lambda)d\lambda$$

$$Z = \int E(\lambda)\bar{z}(\lambda)d\lambda$$

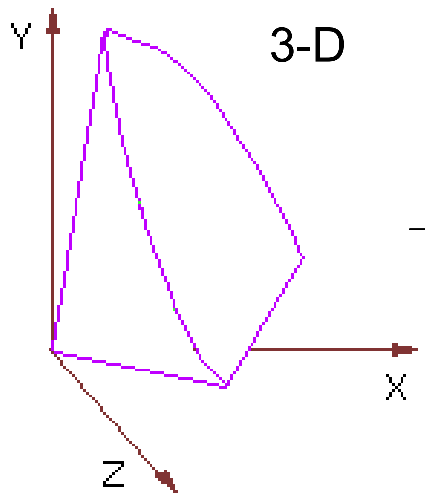


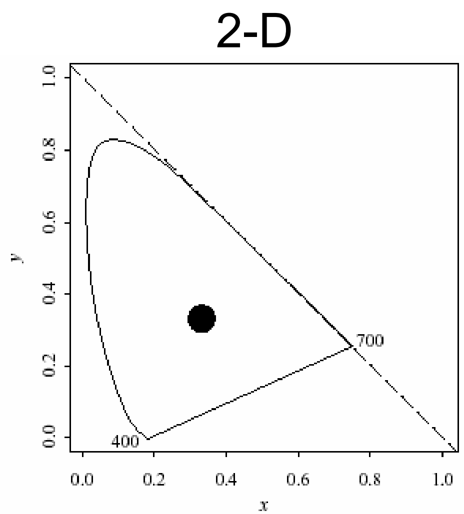
- 3-D data is difficult to visualize, so the CIE devised a 2-D diagram based on the tristimulus values (X,Y,Z)
- We divide the triple by the sum $X+Y+Z$ to make the chromaticity:
  - $x = X/(X+Y+Z)$
  - $y = Y/(X+Y+Z)$
  - $z = Z/(X+Y+Z)$
- This effectively means that one value out of the set (x,y,z) is redundant since we have $x + y + z = (X+Y+Z)/(X+Y+Z) = 1$

  so that $z = 1 - x - y$

CIE Chromaticity Diagram (4)

- Effectively, we are projecting each tristimulus vector (X,Y,Z) onto the plane connecting points (1,0,0), (0,1,0), and (0,0,1).



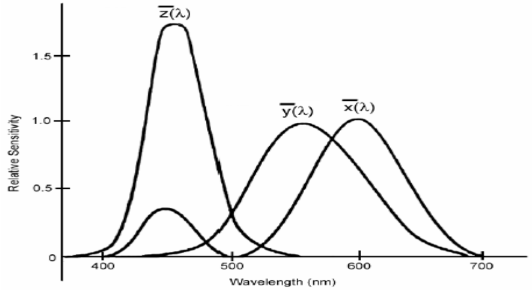


Source: http://www.dgp.toronto.edu/~karan/courses/csc418/fall_2002/notes/colour.html

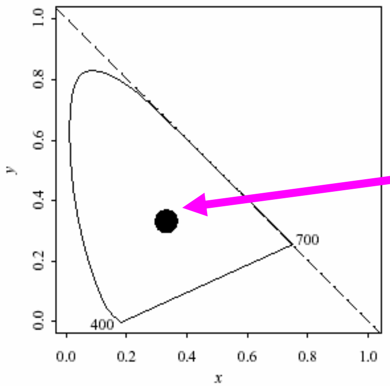
- Since $x,y \leq 1$ and $x + y \leq 1$, all possible chromaticity values lie below the dashed diagonal line

CIE Chromaticity Diagram (5)

- The color matching curves each add up to the same value → the area under each curve is the same for each of $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$

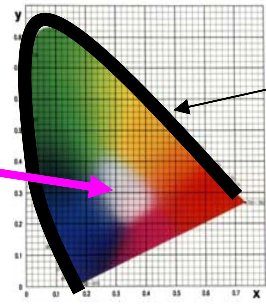


- For an $E(\lambda) = 1$ for all λ, an "equi-energy white light", chromaticity values are (1/3,1/3).



White point

- unsaturated



the locus of points for monochromatic light

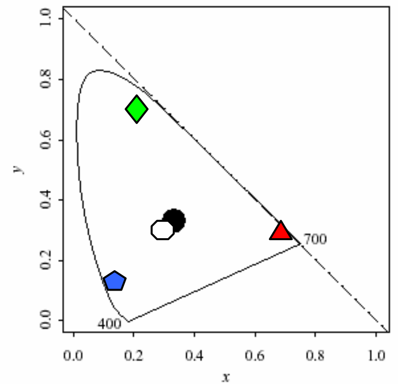
- "Pure" colors
- Most saturated

Source: http://escience.anu.edu.au/lecture/cg/Color/CIEChromaticityDiagram.en.html

Color Monitor Specifications

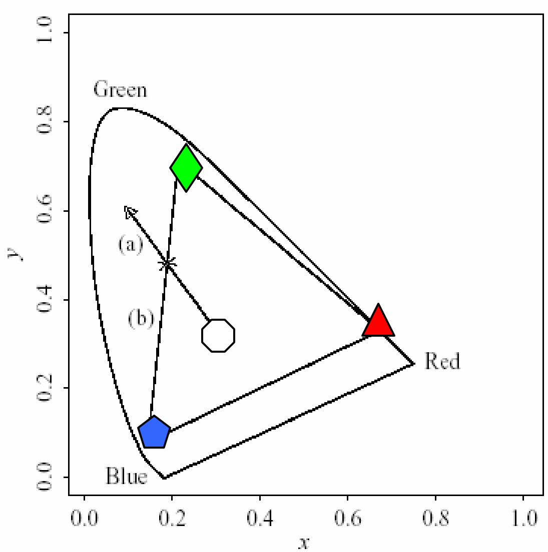
- Color monitors are specified in part by the white point chromaticity that is desired if the *RGB* electron guns are all activated at their highest value (1.0, if we normalize to [0,1]).
- There are several monitor specifications in current use:

System	Red		Green		Blue		White Point	
	x_r	y_r	x_g	y_g	x_b	y_b	x_W	y_W
NTSC	0.67	0.33	0.21	0.71	0.14	0.08	0.3101	0.3162
SMPTE	0.630	0.340	0.310	0.595	0.155	0.070	0.3127	0.3291
EBU	0.64	0.33	0.29	0.60	0.15	0.06	0.3127	0.3291



Out-of-Gamut Colors

- The triangular gamut for the NTSC system is shown on the CIE diagram. A monitor can display only the colors inside a triangular gamut.



System	Red		Green		Blue		White Point	
	x_r	y_r	x_g	y_g	x_b	y_b	x_W	y_W
NTSC	0.67	0.33	0.21	0.71	0.14	0.08	0.3101	0.3162
SMPTE	0.630	0.340	0.310	0.595	0.155	0.070	0.3127	0.3291
EBU	0.64	0.33	0.29	0.60	0.15	0.06	0.3127	0.3291

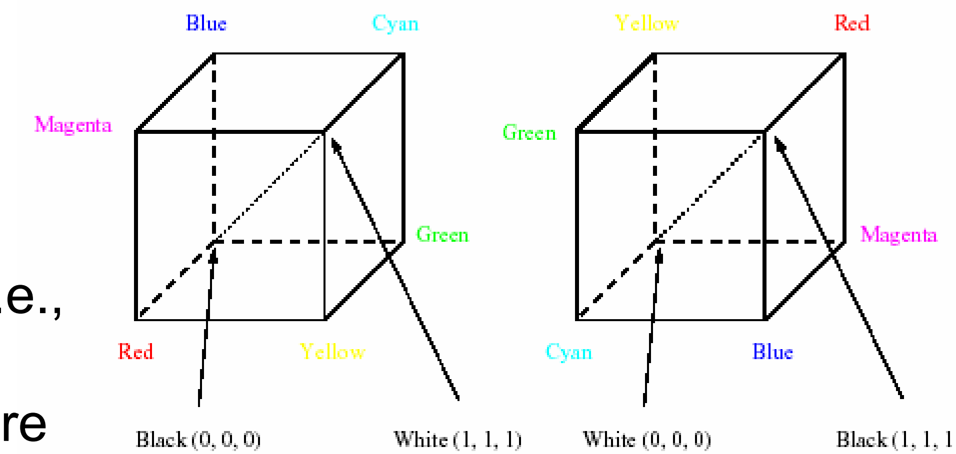
$$\begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- What do we do if any of the *RGB* numbers is *negative*? That color, visible to humans, is out-of-gamut for our display. We could simply use the closest in-gamut color available

Subtractive Color: CMY Color Model

- So far, we have been dealing only with additive color. Namely, when two light beams impinge on a target, their colors add; when two phosphors on a CRT screen are turned on, their colors add.
- But for ink deposited on paper, the opposite situation holds: yellow ink subtracts blue from white illumination, but reflects red and green; it appears yellow.

Instead of red, green, and blue primaries, we need primaries that amount to -red, -green and –blue, i.e., we need to subtract R, G or B. These subtractive color primaries are Cyan (C), Magenta (M) and Yellow (Y) inks.



The RGB Cube

The CMY Cube

Transformation from RGB to CMY

- Simplest model we can invent to specify what ink density to lay down on paper, to make a certain desired RGB color:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Then the inverse transform is:

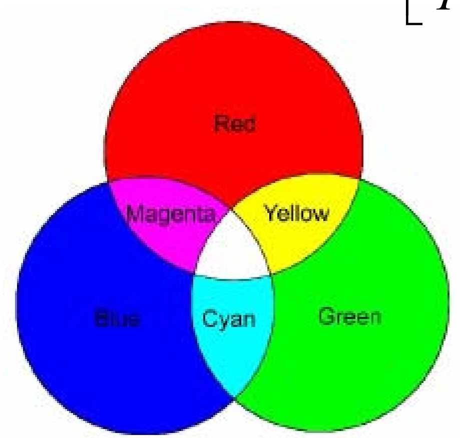
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

Undercolor Removal: CMYK System

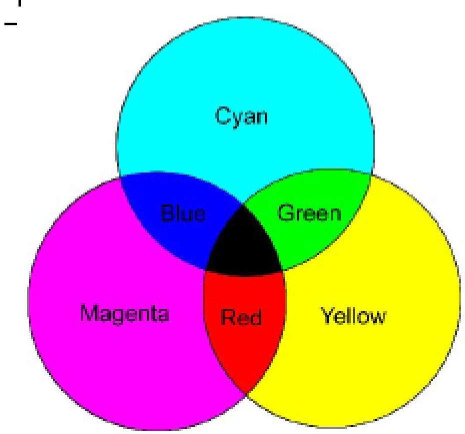
- Undercolor removal: Sharper and cheaper printer colors: calculate that part of the CMY mix that would be black, remove it from the color proportions, and add it back as real black.

$$K = \min\{C, M, Y\}$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} \Rightarrow \begin{bmatrix} C - K \\ M - K \\ Y - K \end{bmatrix}$$



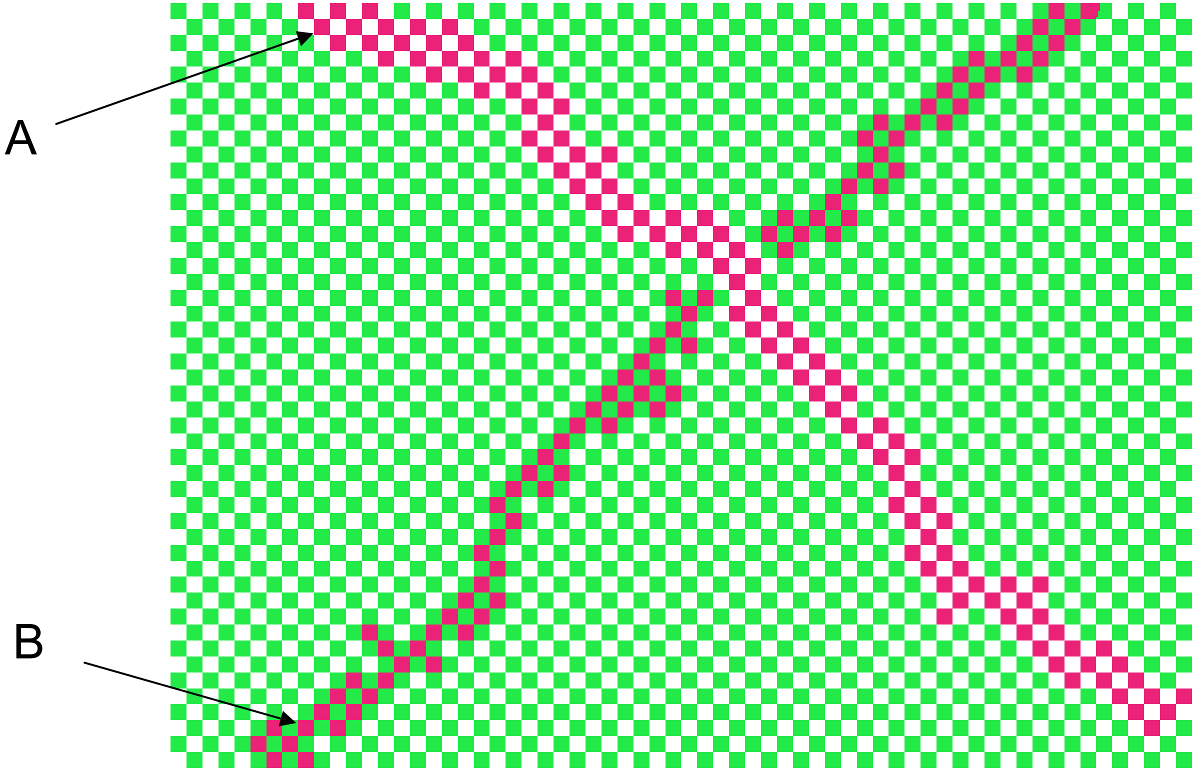
RGB is used to specify additive color



CMY is used to specify subtractive color

Illusion 1

- Which pink pixels (A or B) have larger luminance?



http://www.eyetricks.com/0204.htm

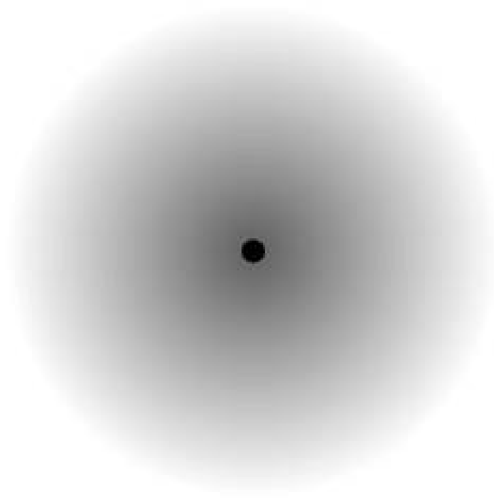
Simultaneous Contrast

- Our perception is sensitive to luminance contrast rather than the absolute luminance values
- Weber's law: if the luminance f_o of an object is just noticeably different from the luminance f_s of its surround, then their ratio is

$$\frac{|f_s - f_o|}{f_o} = \text{constant}$$

Illusion 2

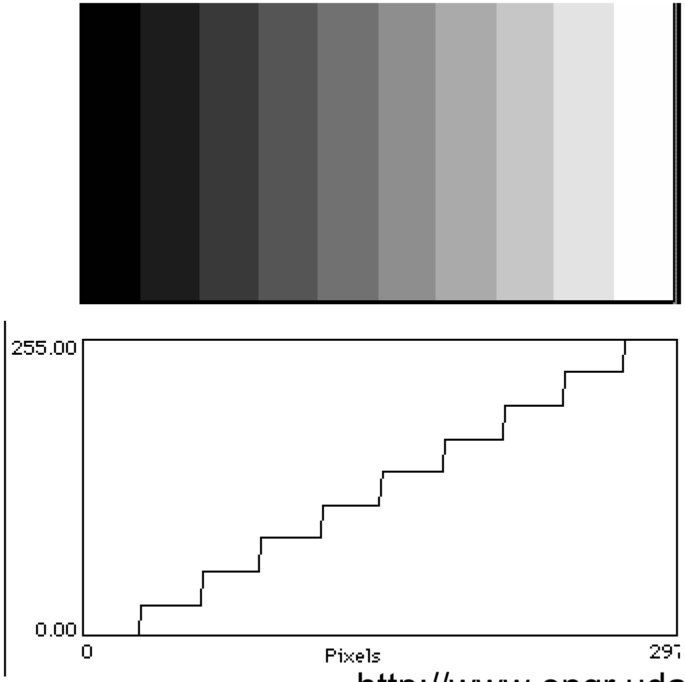
- Stare at the dot in the middle, do you see the outer edges of the circle fade away?

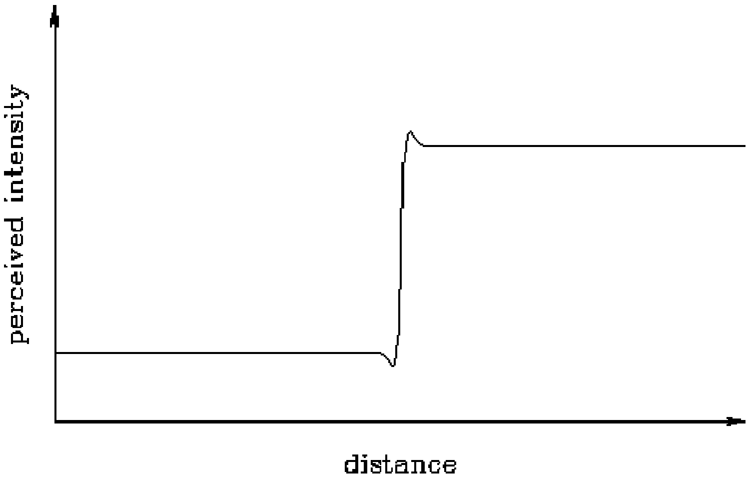


http://www.eyetricks.com/1002.htm

Mach Bands

- Spatial interaction of luminance from an object and its surround creates a phenomenon called the Mach band effect
- This effect shows that brightness is not a monotonic function of luminance





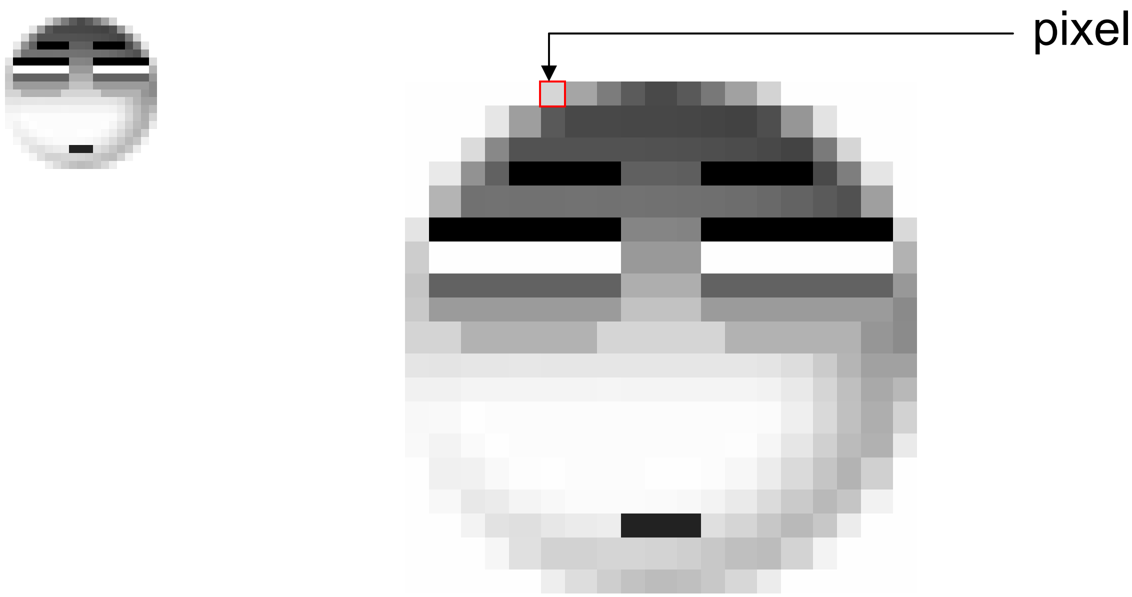
http://www.engr.udayton.edu/faculty/jloomis/ece563/notes/color/GrayScale/grays.html

Image Representation

- Pixel
- 1-Bit Binary Images and 8-Bit Gray-Level Images
- Image Manipulation
  - Changing the Image Resolution
  - Changing the Brightness
  - Changing the Contrast
  - Histogram Equalization
  - Dithering
- 24-Bit Color Images and 8-Bit Color Images
  - Color Look Up Tables (LUTs)
  - Median-Cut Algorithm

Pixel

- Pixel: contraction of the words “picture” and “element”. A pixel corresponds to the smallest detail in a picture that one wants to preserve.



1-Bit Binary Images

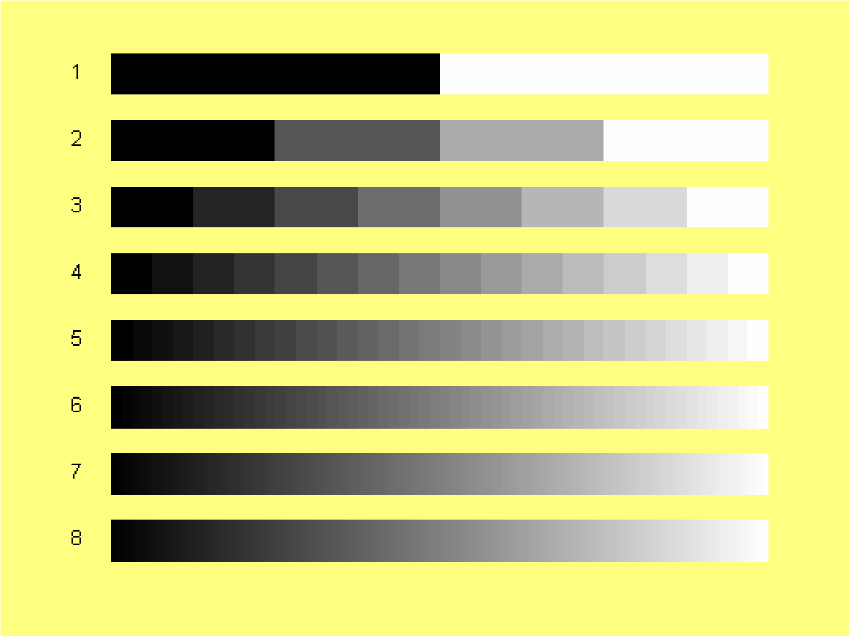
- Each pixel is stored as a single bit (0 or 1), so referred to as binary image.
- Such an image is also called a 1-bit monochrome image since it contains no color.

8-Bit Gray-Level Images

- Each pixel has a gray-value between 0 and 255. Each pixel is represented by a single byte (8 bits); e.g., a dark pixel might have a value of 10, and a bright one might be 230.
- Bitmap: The two-dimensional array of pixel values that represents the graphics/image data.
- Image resolution refers to the number of pixels in a digital image, say 64×64 (higher resolution always yields better quality).

Bits per Pixel

Bits per pixel

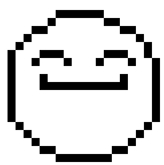


- If number of bits increase by one, number of gray values will double
- As the number of different gray levels increases, it becomes harder for us to visually distinguish the boundaries between the different levels

Bands of gray values that can be specified by a fixed number of bits in the binary representation of the pixels

Example Images

1-Bit Binary Images



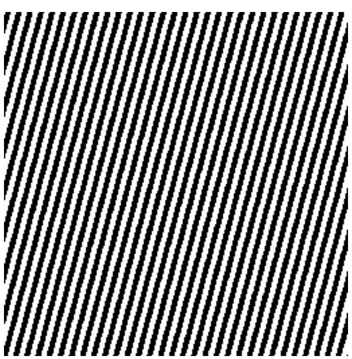
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	1
1	1	1	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	0	0	0	1	1	1	1	1	0	0	0	1	1	0	1
0	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1

8-Bit Gray-Level

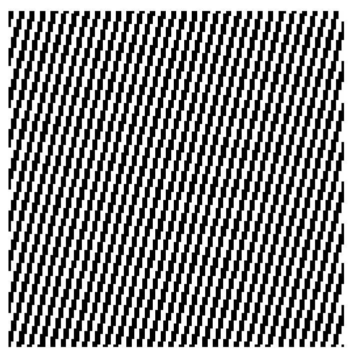
255	255	255	255	255	214	165	124	91	73	89	120	161	211	255	255	255	255	255
255	255	255	230	158	89	72	72	71	71	70	68	66	78	150	227	255	255	255
255	255	219	136	82	82	82	82	82	81	80	78	76	72	67	122	214	255	255
255	233	146	97	0	0	0	0	96	96	94	0	0	0	0	73	126	229	255
255	180	113	114	113	113	114	113	114	113	112	111	109	105	99	90	81	159	255
228	0	0	0	0	0	0	0	133	133	131	0	0	0	0	0	0	0	217
205	255	255	255	255	255	255	255	153	153	153	255	255	255	255	255	255	255	178
197	98	98	98	98	98	98	98	174	174	174	98	98	98	98	98	98	98	152
201	155	155	155	155	155	155	155	194	194	194	155	155	155	155	155	155	155	138
212	212	178	178	178	178	178	213	213	213	213	213	178	178	178	178	178	150	139
229	228	230	230	231	230	230	230	230	230	230	230	230	228	221	203	181	161	161
241	241	244	244	244	244	244	245	244	244	244	244	244	242	233	213	191	169	184
248	249	254	252	252	252	252	252	252	252	252	252	252	251	239	217	192	174	210
249	243	250	254	252	252	252	252	252	252	252	252	253	246	230	208	187	177	234
255	240	241	249	253	254	252	252	252	254	254	252	247	236	218	197	179	208	255
255	248	232	235	244	248	251	251	251	250	248	243	234	219	202	185	197	242	255
255	255	243	226	223	231	235	237	34	34	34	223	213	199	185	199	236	255	255
255	255	255	246	224	210	210	213	213	211	207	200	191	188	211	243	255	255	255
255	255	255	255	255	238	221	206	194	188	191	200	215	236	255	255	255	255	255

Changing the Image Resolution

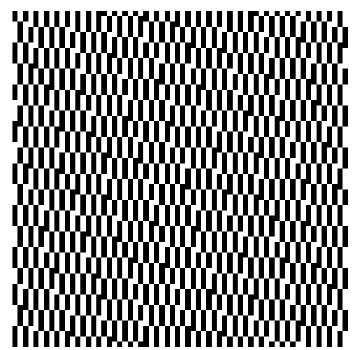
- The pixel value is a sample of light intensity. Sampling can be considered as the process of taking samples of light intensity to form the pixels.
- The resolution of an image can be made smaller by downsampling, e.g., keeping one pixel for every 2×2 block of pixels. The sampling rate is descreased after downsampling.
- Sampling artifacts: distortions arising in a sampled image when the sampling rate is too small to capture the finest detail in the input image. This resulting effect is called spatial aliasing.



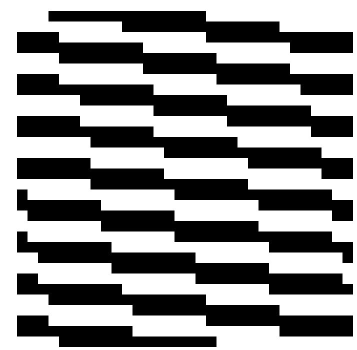
256×256



128×128



64×64



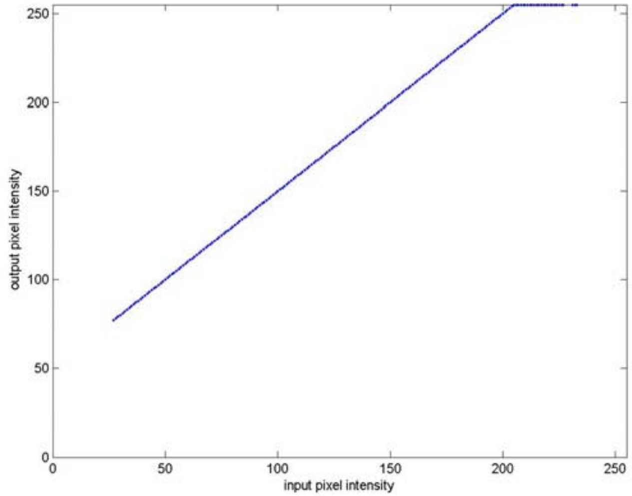
32×32

Changing the Brightness

- Brightness can be changed by adding or subtracting a constant value to all pixel values
- When sum or difference is not a possible pixel value, the nearest possible value is used

Original image

Image with increased brightness

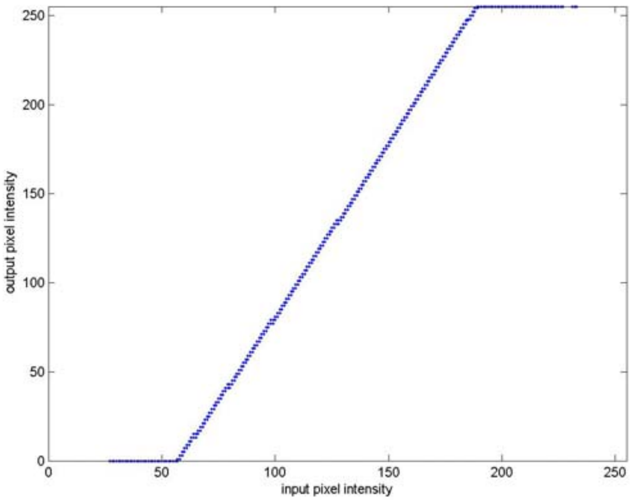


Changing the Contrast

- Add an offset and then multiple with a constant value to all pixel values
- When result is not a possible pixel value, the nearest possible value is used

Original image

Image with increased contrast

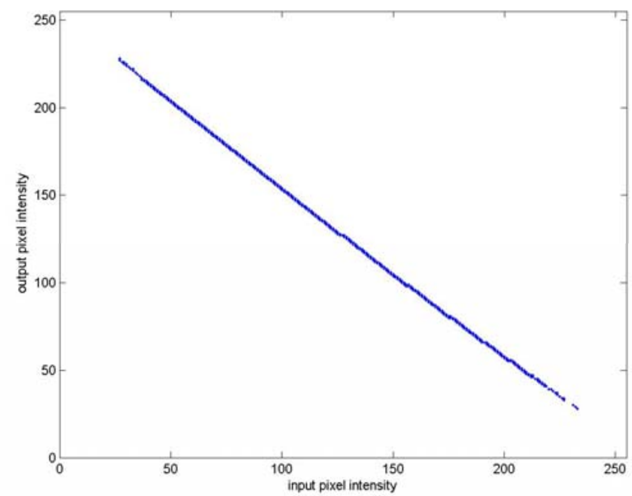


Negative Image

- Subtract a constant value from all pixel values
- Film: negative; Photograph: positive

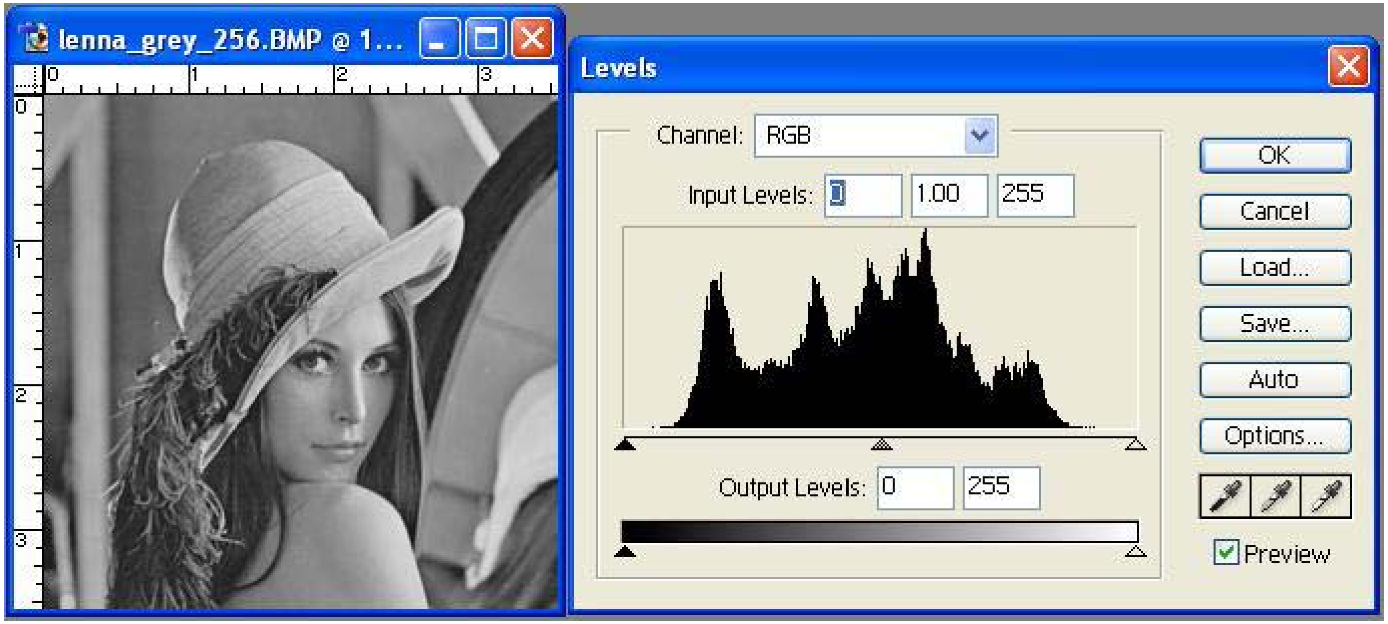
Original image

Negative Image



Histogram

- A function showing, for each intensity level, the number of pixels in the image that have that gray level. The x-axis is the intensity level and the y-axis is the frequency of occurrence (number of pixels)



Histogram Equalization

- Objective: transform an input image to an output image with equally many pixels at every gray level (a flat histogram)
- After histogram equalization, actual histogram will usually take on a rather ragged appearance due to the finite number of available gray levels. Some gray levels will be unoccupied and others highly populated

Before histogram equalization

After histogram equalization

Dithering

- When an image is printed, the basic strategy of dithering is used, which trades intensity resolution for spatial resolution to provide ability to print multi-level images on 2-level (1-bit) printers.
- Dithering is used to calculate patterns of dots such that values from 0 to 255 correspond to patterns that are more and more filled at darker pixel values, for printing on a 1-bit printer.
- The main strategy is to replace a pixel value by a larger pattern, say 2×2 or 4×4 binary matrix, such that the resulting black and white dot pattern approximates shading.

Dithering (2)

- For example, if we use a 2×2 dither matrix $\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$
- we can first re-map image values in 0..255 into the new range 0..4 by (integer) dividing by 256/5. Then, e.g., if the pixel value is 0 we print nothing, in a 2×2 area of printer output. But if the pixel value is 4 we print all four dots.
- The rule is: If the intensity is > the dither matrix entry then print an on dot at that entry location. Replace each pixel by an $n \times n$ matrix of dots.

Example:

Dither matrix

$$\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 122 & 69 \\ 180 & 240 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \longrightarrow \left[\begin{array}{cc:cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hdashline 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{array}\right]$$

2×2 Gray-Level Image

Integer divide by 256/5

Dithered Binary Image

Dithering (3)

- The dither matrix can be 4×4 instead of 2×2

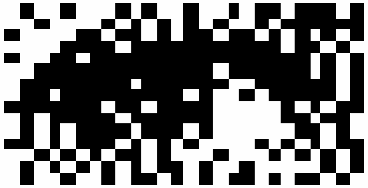
$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

Q: How to derive the 8×8 dither matrix ?

- Example images

Original Gray-Level Image

Dithered Binary Image



Detail of Dithered version

24-Bit Color Images

- In a color 24-bit image, each pixel is represented by three bytes, usually representing RGB.
  - This format supports 256×256×256 possible combined colors, or a total of 16,777,216 possible colors.
  - However such flexibility does result in a storage penalty: A 640×480 24-bit color image would require 900KB (640×480×24/8/1024) of storage without any compression.
- Many 24-bit color images are actually stored as 32-bit images, with the extra byte of data for each pixel used to store an *alpha* value representing special effect information (e.g., transparency).

24-Bit Color Images (2)

- Example

24-Bit Color Image

R Channel

G Channel

B Channel

24-Bit Color Images (3)

- Example

24-Bit Color Image

255	255	255	255	255	113	106	94	84	76	69	69	78	97	255	255	255	255	255
255	255	255	114	108	96	77	54	36	31	34	44	65	82	89	96	255	255	255
255	255	110	100	84	64	36	14	5	4	9	23	42	47	44	45	47	255	255
90	245	84	68	48	26	9	2	0	0	2	188	15	11	8	8	11	9	255
75	66	46	26	12	4	1	0	0	0	197	226	5	1	1	1	2	2	245
197	34	13	3	1	0	0	0	0	197	236	245	197	1	1	1	1	0	0
236	197	3	0	0	0	1	4	197	236	255	255	236	197	3	1	1	0	0
255	236	197	188	188	188	188	197	236	255	255	255	255	236	197	188	1	0	0
255	255	245	245	245	245	245	245	255	255	255	255	255	255	245	236	197	0	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	236	0	0
0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	255	245	2	142
253	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	245	8	183
249	253	255	255	255	255	255	255	255	255	255	255	255	255	255	255	64	172	199
247	250	255	255	255	255	0	255	0	255	255	255	255	255	255	251	209	178	232
255	242	249	254	255	255	255	0	255	255	255	255	255	255	255	247	203	206	255
255	246	227	237	251	255	255	255	255	255	255	255	255	255	249	223	202	242	255
255	255	237	212	224	246	252	255	255	255	255	255	254	247	221	204	235	255	255
255	255	255	242	215	212	230	243	247	249	249	243	230	211	213	241	255	255	255
255	255	255	255	255	234	218	209	205	202	205	209	217	234	255	255	255	255	255

R Channel

255	255	255	255	255	42	38	32	25	17	8	5	12	29	255	255	255	255	255
255	255	255	44	37	28	20	11	7	5	2	3	8	19	30	34	255	255	255
255	255	40	32	22	11	4	1	0	0	1	1	3	7	9	12	13	255	255
17	239	22	14	7	2	1	0	0	0	0	147	1	1	1	2	2	2	255
7	9	6	3	1	0	0	0	0	0	158	178	0	0	0	0	0	0	245
189	1	1	0	0	0	0	0	0	159	190	194	154	0	0	0	0	0	0
215	162	0	0	0	0	1	1	161	191	205	203	185	151	0	0	0	0	0
221	193	162	155	155	155	154	161	192	206	204	203	200	181	146	134	0	0	0
212	207	199	201	202	202	201	200	207	206	204	203	200	196	182	168	132	0	0
201	200	201	203	207	209	209	208	207	206	204	202	197	192	184	179	155	0	0
0	0	0	0	0	0	208	208	207	0	0	0	0	0	0	165	154	0	92
167	126	116	139	175	199	204	207	206	205	200	188	163	132	118	128	136	0	132
138	61	41	81	146	186	202	204	203	201	194	172	125	67	44	75	0	89	165
134	35	10	54	129	177	0	199	0	196	188	161	103	32	7	46	77	98	215
177	76	46	82	138	174	188	0	190	188	182	159	116	61	41	64	82	156	252
219	170	110	113	149	170	177	180	178	177	171	159	136	109	93	91	129	226	255
245	235	202	148	135	153	161	163	163	160	158	152	143	130	110	138	213	253	255
253	252	251	230	176	135	135	142	144	144	143	137	129	127	170	229	253	255	255
255	255	255	255	255	219	180	151	128	116	129	151	180	220	255	255	255	255	255

G Channel

255	255	255	255	255	0	0	0	0	0	0	0	0	0	255	255	255	255	255
255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255
255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255
0	211	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	245
153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66
86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123
117	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	196
177	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111	252
219	160	54	0	0	0	0	0	0	0	0	0	0	0	0	0	71	213	255
245	235	184	74	0	0	0	0	0	0	0	0	0	0	0	74	195	253	255
253	252	251	217	120	23	0	0	0	0	0	0	0	23	118	217	253	255	255
255	255	255	255	255	198	130	72	28	3	28	72	130	198	255	255	255	255	255

B Channel

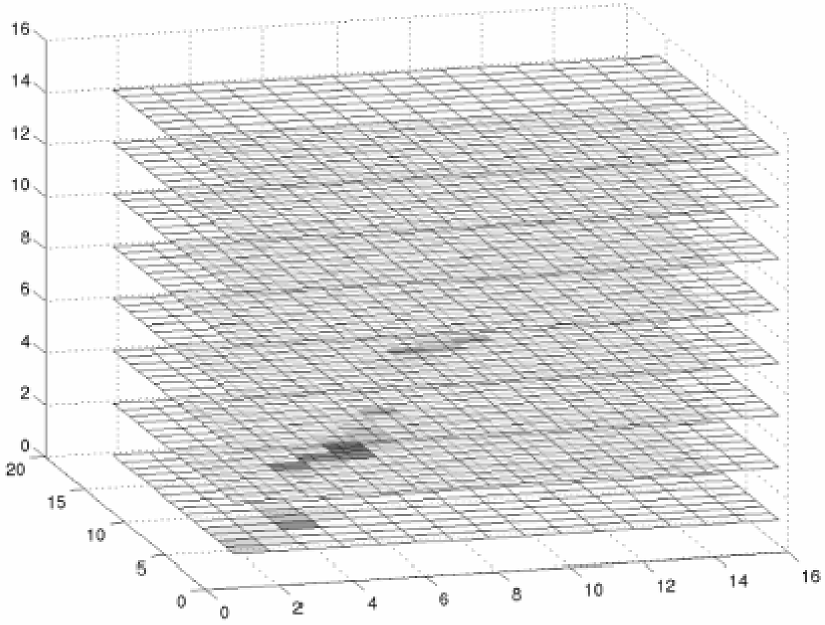
8-Bit Color Images

- Many systems can make use of 8 bits of color information (the so-called “256 colors”) in producing a screen image.
- Such image files use the concept of a lookup table to store color information.
- Basically, the image stores not color, but instead just a set of bytes, each of which is actually an index into a table with 3-byte values that specify the color for a pixel with that lookup table index.

8-Bit Color Images (2)

- Example

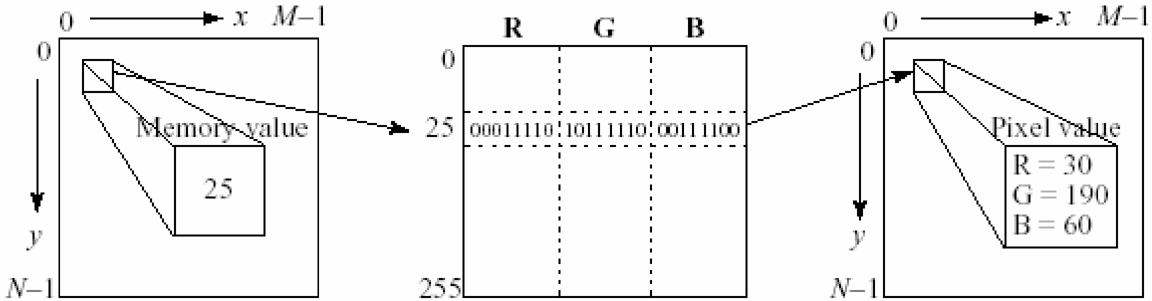
8-Bit Color Image



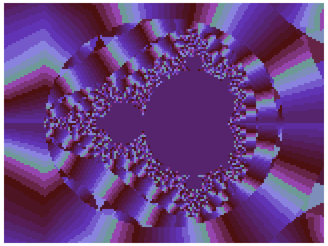
3-D Histogram

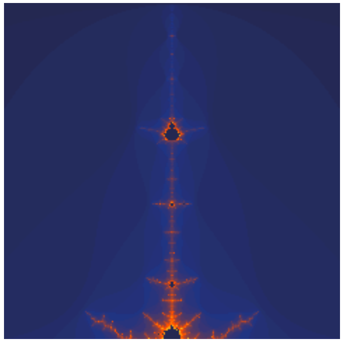
Color Lookup Table

- The idea used in 8-bit color images is to store only the index, or code value, for each pixel. Then, e.g., if a pixel stores the value 25, the meaning is to go to row 25 in a color look-up table (LUT).



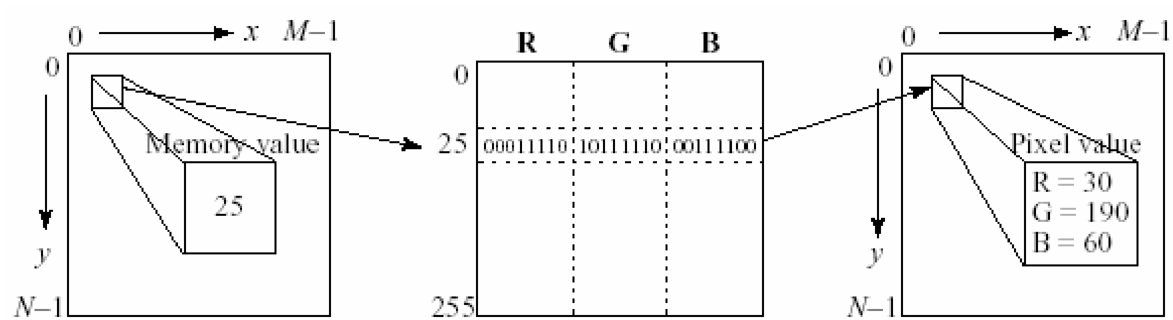
- A very simple animation process is possible via simply changing the color table: this is called color cycling or palette animation.





How to Devise a Color Lookup Table

- Approach 1: The most straightforward way to make 8-bit look-up color out of 24-bit color would be to divide the RGB cube into equal rectangular blocks.
- The centers of each of the resulting rectangular blocks would serve as the entries in the color LUT.
- Since humans are more sensitive to R and G than to B, we could shrink the R range and G range 0..255 into the 3-bit range 0..7 and shrink the B range down to the 2-bit range 0..3, thus making up a total of 8 bits.

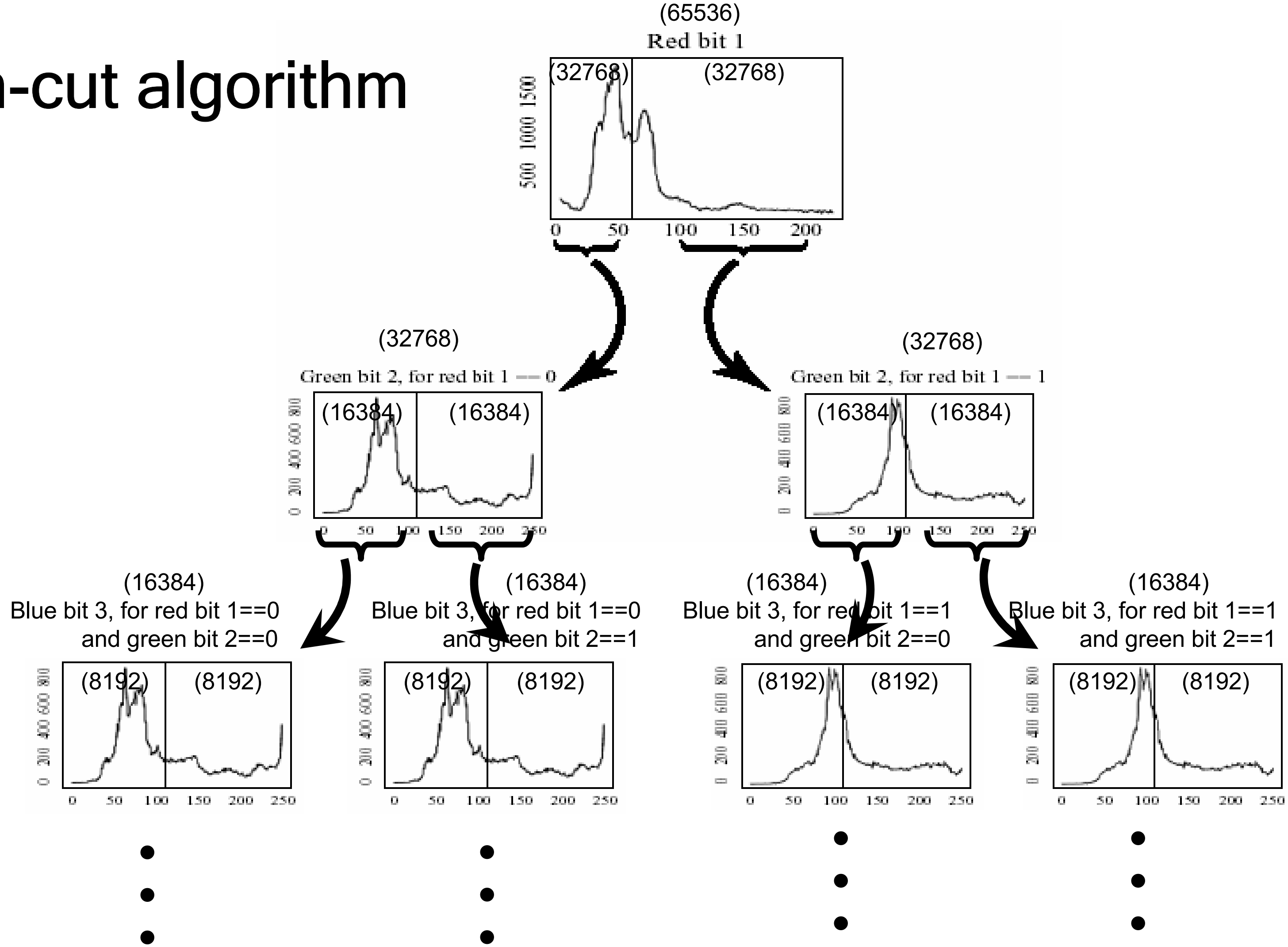


How to Devise a Color Lookup Table (2)

- Approach 2: Median-Cut Algorithm: A simple alternate solution that does a better job for this color reduction problem.
- The idea is to sort the R byte values and find their median; then values smaller than the median are labeled with a “0” bit and values larger than the median are labeled with a “1” bit.
- This type of scheme will indeed concentrate bits where they most need to differentiate between high populations of close colors.
- One can most easily visualize finding the median by using a histogram showing counts at position 0..255.

How to Devise a Color Lookup Table (3)

- Median-cut algorithm



Summary

Color Science

- Introduction
  - Light and Spectra
  - Human Vision
  - Spectral Sensitivity of the Eye
  - Image Formation
- Color Representation and Specification
  - Color-Matching Functions
  - CIE Chromaticity Diagram
  - Out-of-Gamut Colors
  - Color Monitor Specification
  - Subtractive Color: CMY Color Model
- Perception of Color
  - Simultaneous Contrast
  - Mach Bands

Image Representation

- Pixel
- 1-Bit Binary Images and 8-Bit Gray-Level Images
- Image Manipulation
  - Changing the Image Resolution
  - Changing the Brightness
  - Changing the Contrast
  - Histogram Equalization
  - Dithering
- 24-Bit Color Images and 8-Bit Color Images
  - Color Look Up Tables (LUTs)
  - Median-Cut Algorithm

References

1. Ze-Nian Li and Mark S. Drew : Fundamentals of Multimedia, (Prentice Hall, 2004, ISBN: 0130618721)
2. Kenneth R. Casleman : Digital Image Processing, (Prentice Hall, 1996, ISBN: 0132114674)
3. Geoffrey C. Orsak, Sally L. Wood, Scott C. Douglas, David C. Munson, John R. Treichler, Ravindra A. Athale, Mark W. Yoder, Mark A. Yoder: Engineering Our Digital Future: The Infinity Project, (Prentice Hall, 2003, ISBN: 0130354821)
4. Anil K. Jain : Fundamentals of Digital Image Processing, (Prentice Hall, 1989, ISBN: 0133361659)