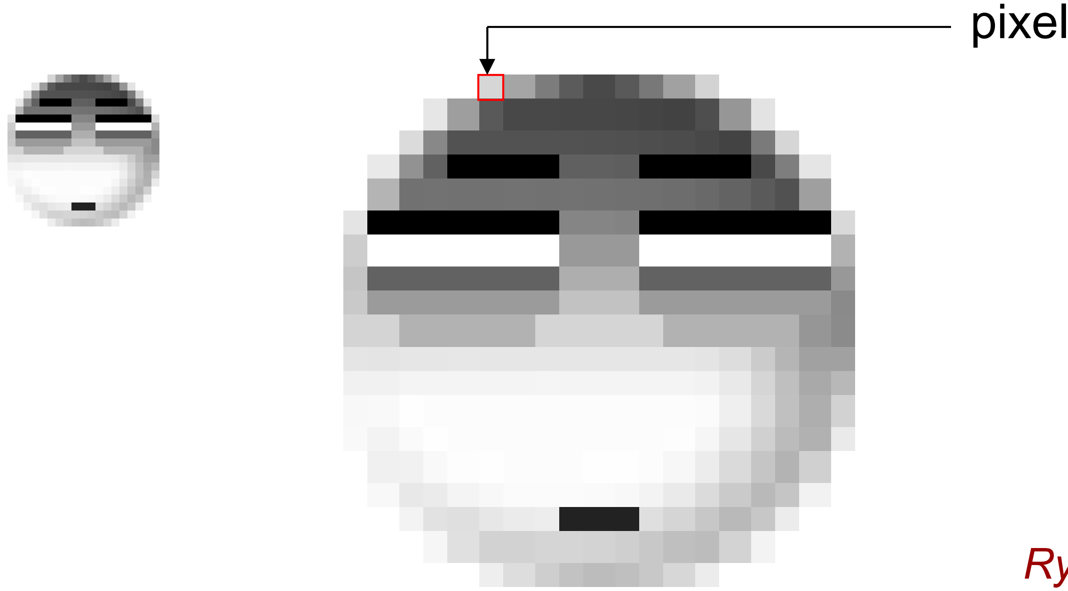
Image Representation

Rynson W.H. Lau

Pixels

- **Pixel**: comes from the words "*picture*" and "*element*". A pixel corresponds to the smallest unit in a digital picture.
- Image resolution refers to the number of pixels in a digital image. For example, HDTV has a resolution of 1920×1080 pixels.



Binary Images

- Each pixel is represented by a single bit (0 or 1).
- Such an image is also called a monochrome image since it contains only a single foreground color on a single background color (usually black/white).

original image

binary image

8-Bit Gray-Level Images

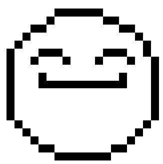
- Each pixel is represented by an 8-bit black and white value, i.e., between 0 and 255, where 0 corresponds to the smallest value (completely black) while 255 corresponds to the highest value (maximum brightness).

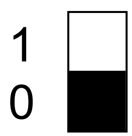
original image

8-bit gray-level image

Example Images

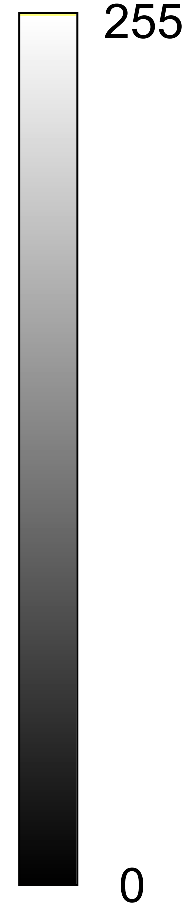
1-Bit Binary Image





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 Image

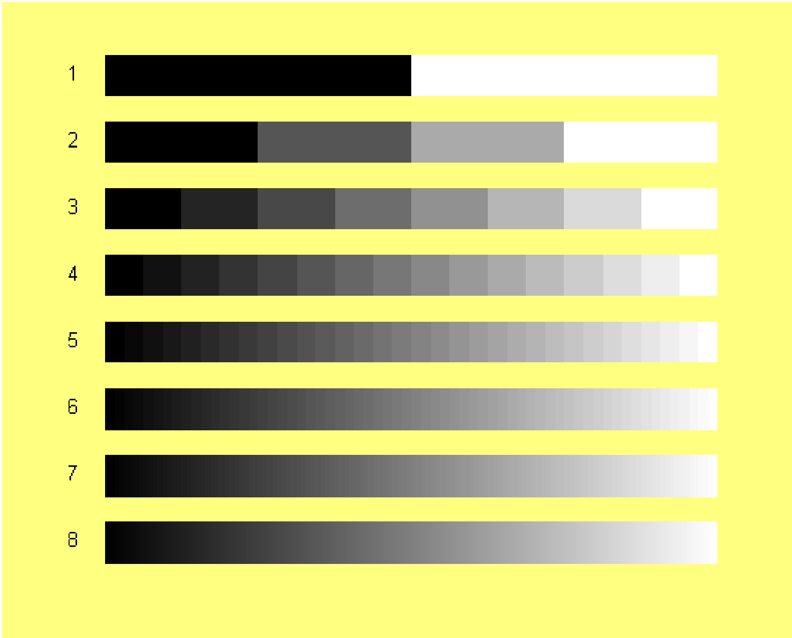


255	255	255	255	255	214	165	124	91	73	89	120	161	211	255	255	255	255	255
255	255	255	230	158	89	72	72	71	71	70	68	66	78	150	227	255	255	255
255	255	219	136	82	82	82	82	82	81	80	78	76	72	67	122	214	255	255
255	233	146	97	0	0	0	0	96	96	94	0	0	0	0	73	126	229	255
255	180	113	114	113	113	114	113	114	113	112	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

Number of Bits per Pixel

- As the number of bits per pixel increases by one, the number of gray values is doubled.
- As the number of different gray values increases, the boundary between any two adjacent values becomes less visible.

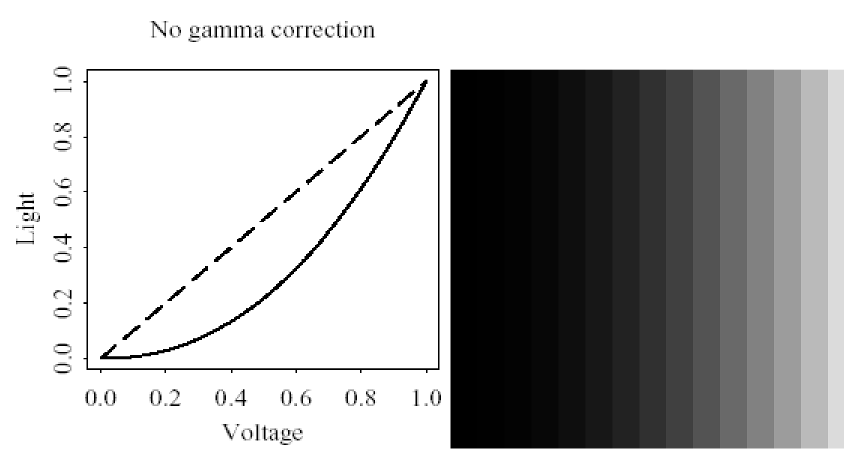
Bits per pixel (color depth)



Number of different gray values that can be specified by a fixed number of bits in a pixel of a digital image.

Gamma Correction

- To display an image, we typically need to convert each RGB value to a voltage to drive the display screen.
- However, the mapping between the driving voltage and the amount of light emitted is typically non-linear for most display devices, such as CRTs and LCDs. The light emitted is roughly proportional to the voltage *raised to a power*; and this power is called **gamma**, with symbol γ.
- Hence, if a pixel value in the red channel is R, the screen emits light proportional to R^{γ}. For LCD monitors, the gamma value is about 2.2.



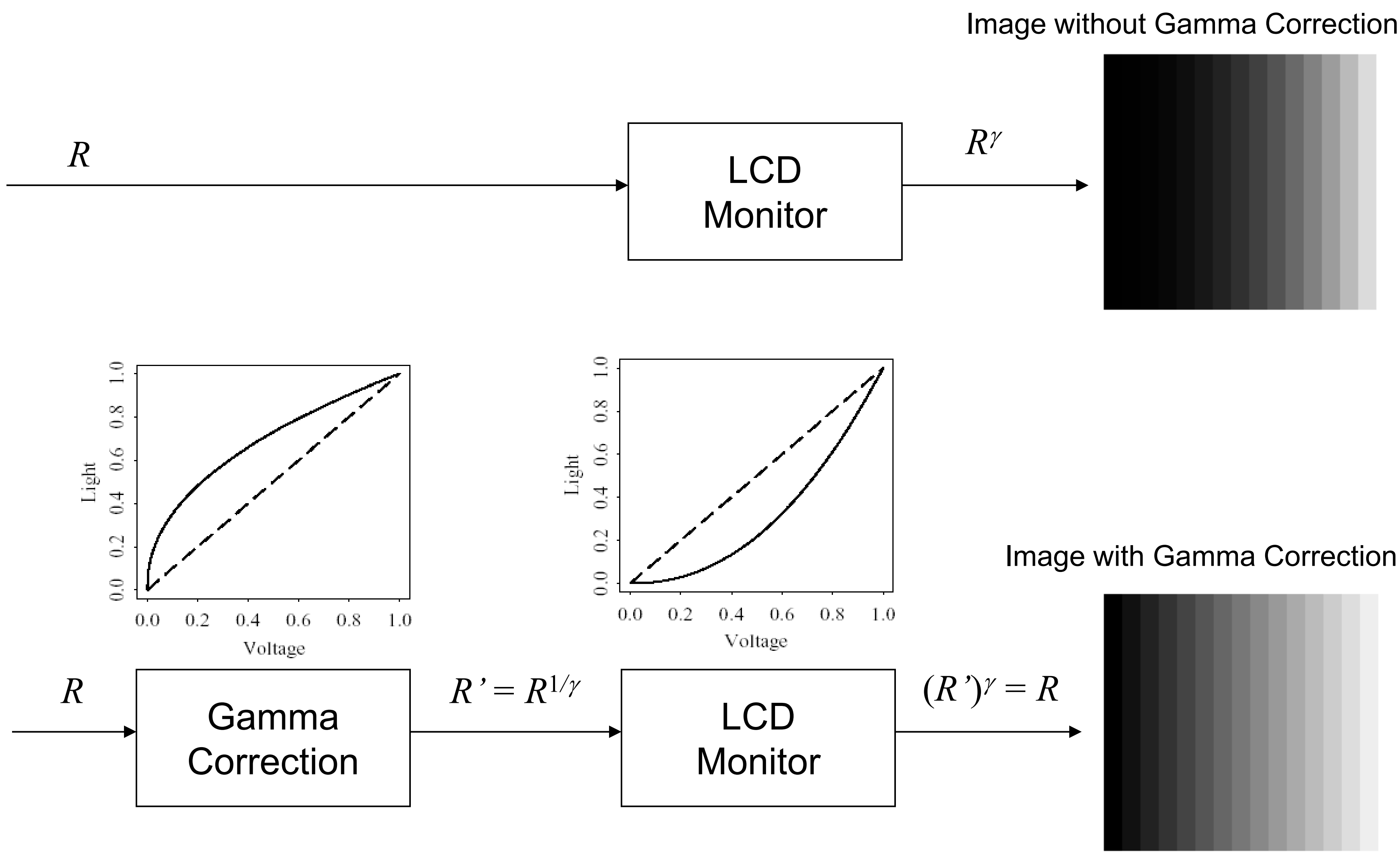
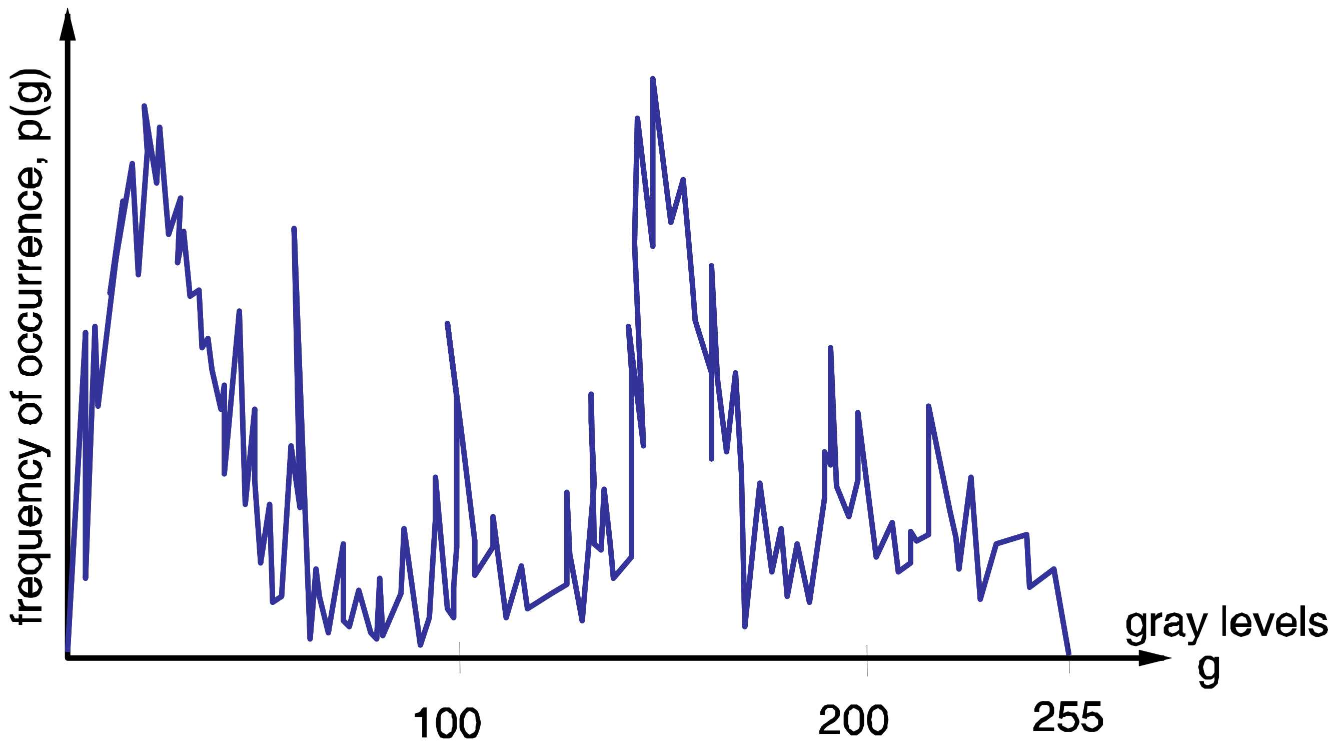


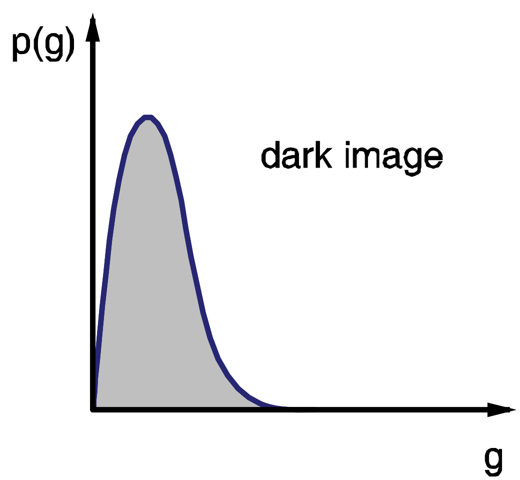
Image without Gamma Correction
R
LCD Monitor
Rγ
Light
Voltage
0.0 0.2 0.4 0.6 0.8 1.0
Image with Gamma Correction
R
Gamma Correction
R' = R^{1/γ}
LCD Monitor
(R')^γ = R

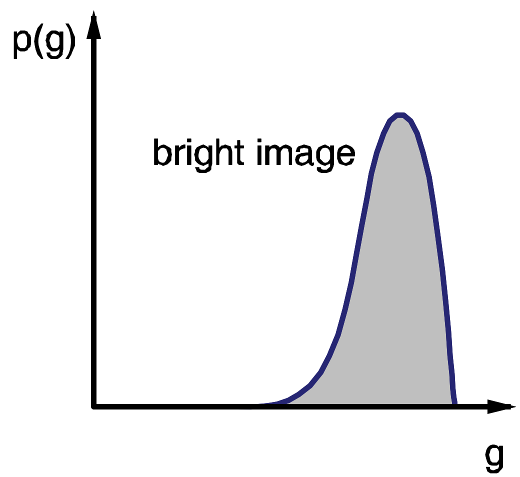
Histogram

- A histogram shows the number of occurrences, *p*(*g*), of each individual gray-level value, *g*, in an image.



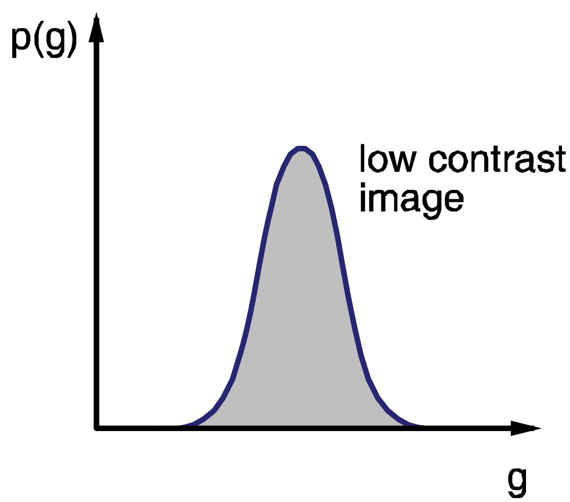
- The distribution of pixel values tells the quality of the image. For example:
  - If majority of the pixels have small gray values, the image will appear dark.
  - If majority of the pixels have large gray values, the image will appear bright.

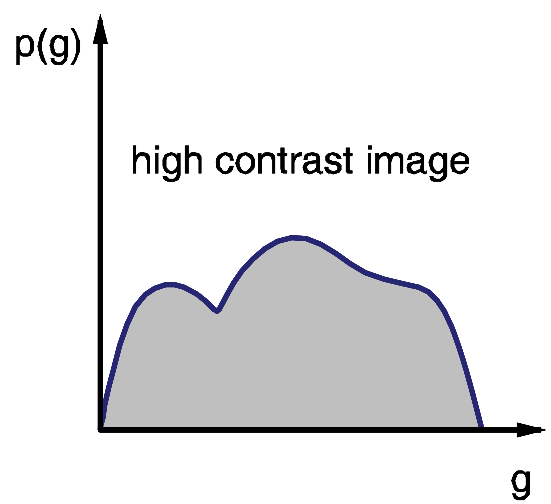




– If majority of pixel values are within a small range, the image will appear to have a low contrast. (The previous two images are also considered as low contrast images.)

– On the contrary, a high contrast image covers a wide range of gray values.





Brightness

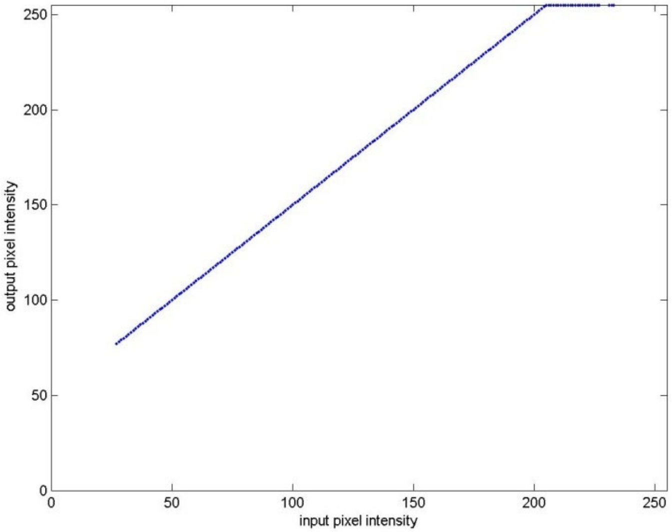
- Brightness of the image can be adjusted as follows:

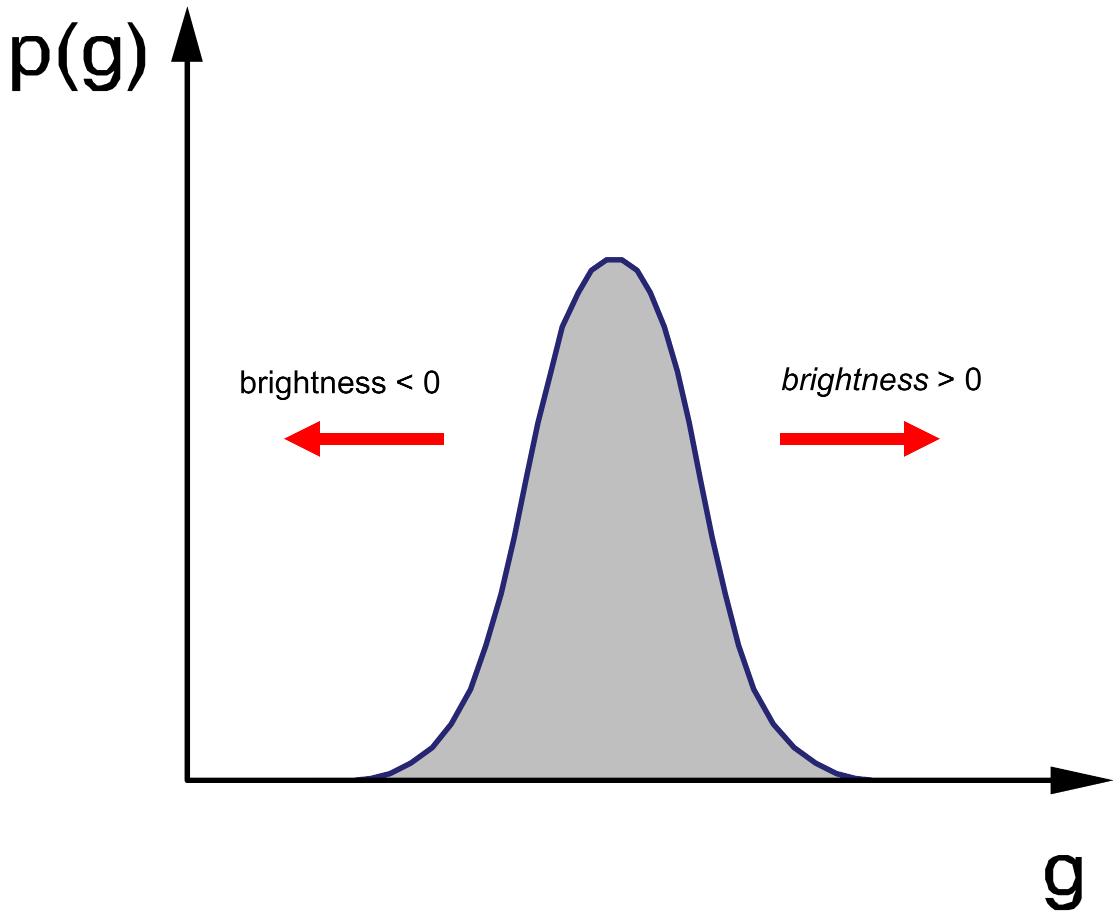
$$pixel'(x, y) = pixel(x, y) + brightness$$

- If the sum of the above equation exceeds the range of [0, 255], the output value is clipped to 0 or 255.
- This brightness adjustment is equivalent to shifting the histogram horizontally.

original image

image with increased brightness




p(g)
brightness < 0
brightness > 0
g

Contrast

- Contrast of an image can be adjusted as follows:

  $$pixel'(x, y) = contrast * (pixel(x, y) - g_{mean}) + g_{mean}$$

  where g_{mean} is the mean gray value of the original image.

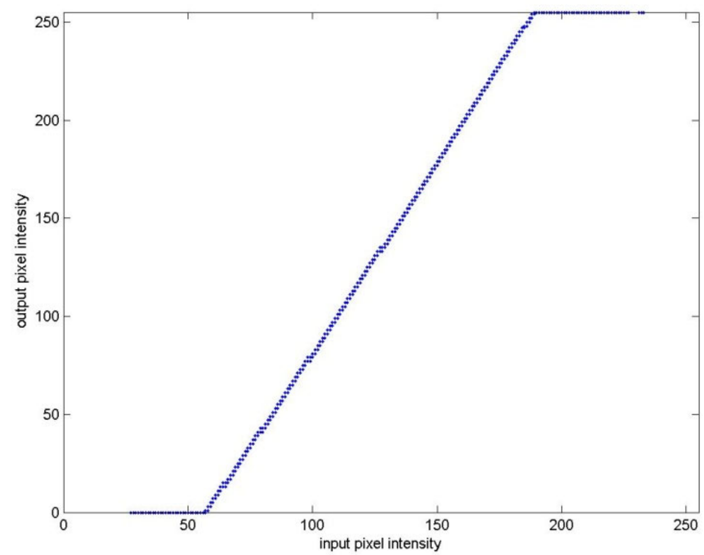
- If X and Y represent the horizontal and vertical resolutions of the image, g_{mean} can be computed as:

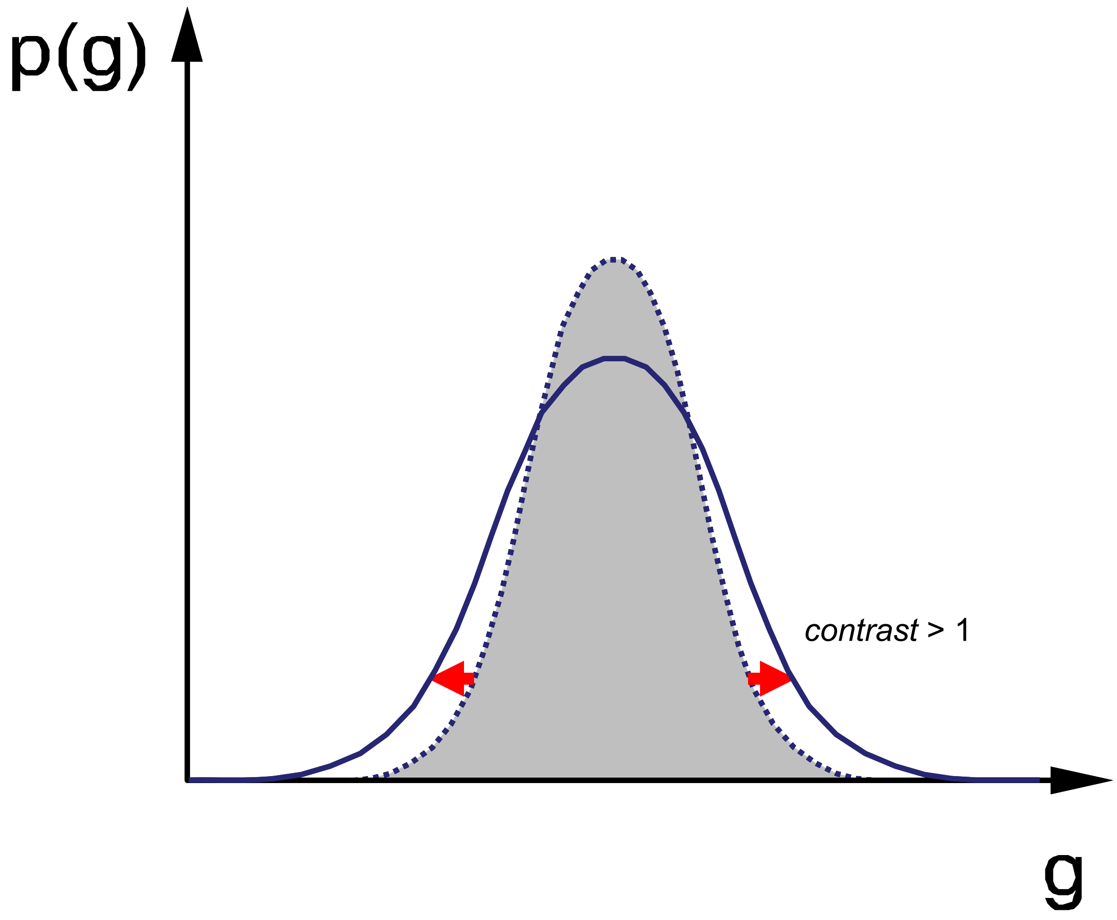
  $$g_{mean} = \frac{\sum_{y=0}^{Y} \sum_{x=0}^{X} pixel(x, y)}{X * Y}$$

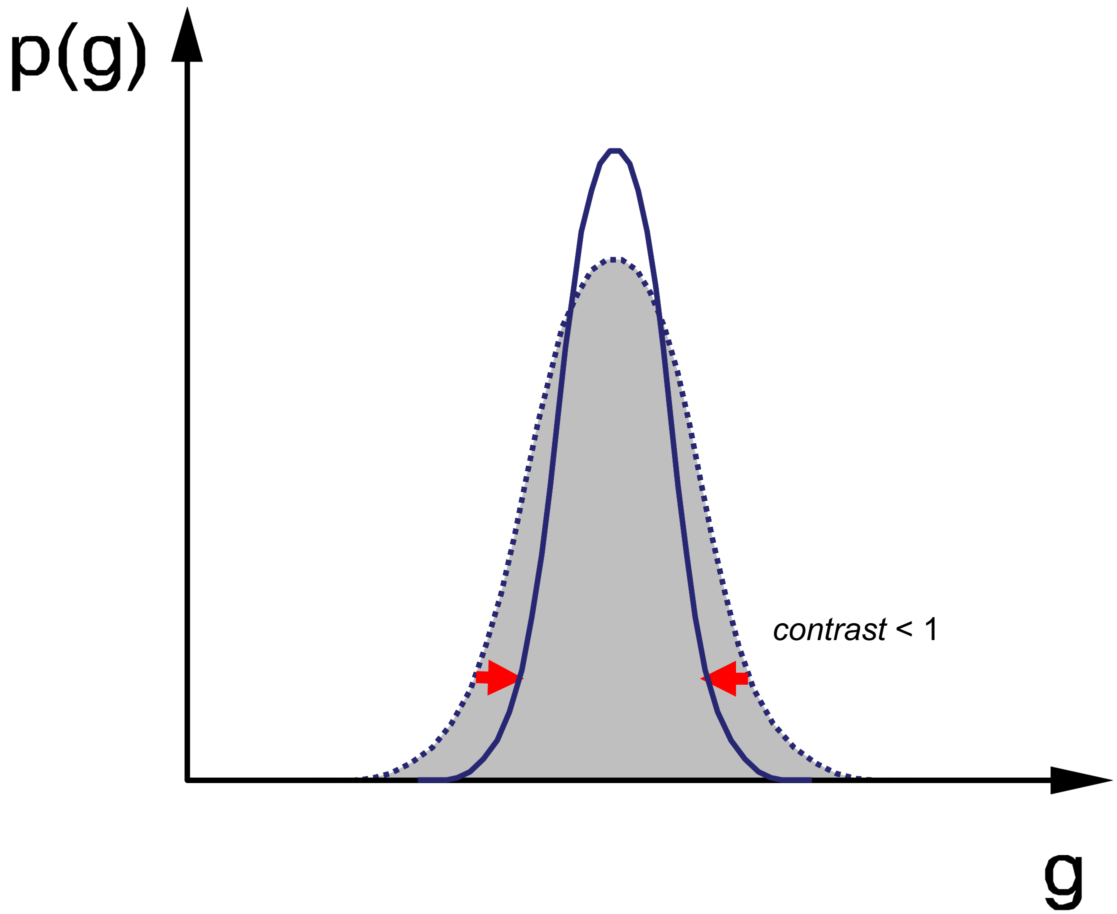
- Contrast correction is similar to scaling the histogram distribution. We reduce the contrast of an image by setting the *contrast* value (in the equation) to smaller than 1 and increase the contrast by setting *contrast* to larger than 1.
- Again, if an output value exceeds the range [0, 255], it is clipped to 0 or 255.


original image


image with increased contrast




p(g)
contrast > 1
g


p(g)
contrast < 1
g

Inversion (Negative)

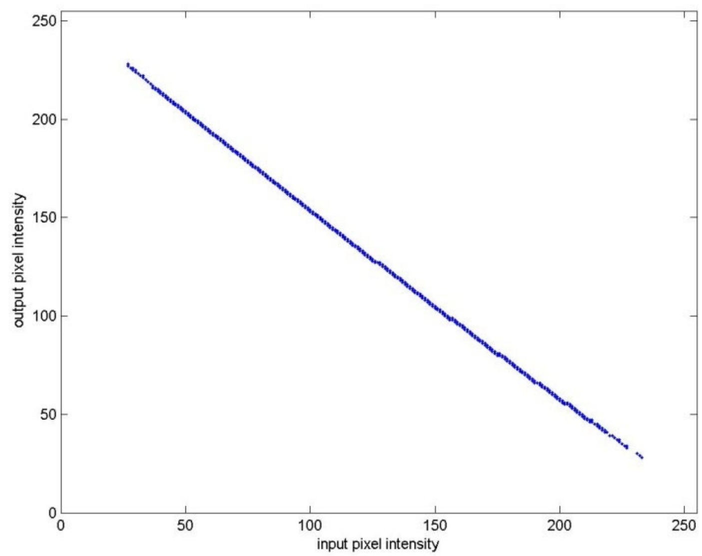
- Image inversion can be achieved as follows:

$$pixel'(x, y) = 255 - pixel(x, y)$$

- Films are negative, while photographs are positive.

Original image
(positive image)

negative Image

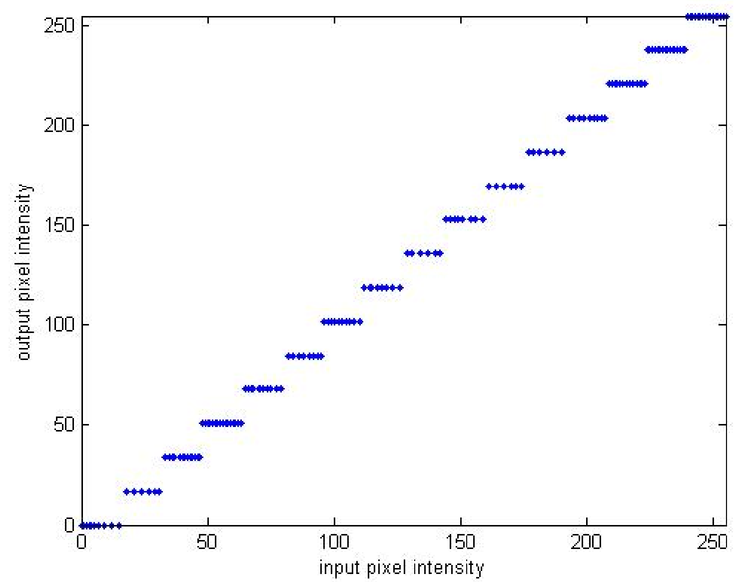


Quantization

➢ This is to quantize the gray values of an image into a smaller number of levels.

original image
(256 levels)

quantized Image
(16 levels)

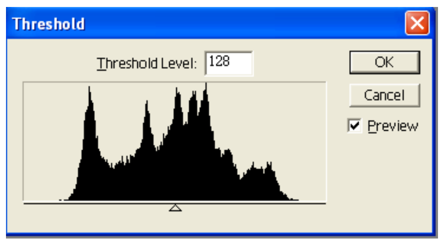


Thresholding (Binarization)

- This is to convert a gray-level image into a binary image:

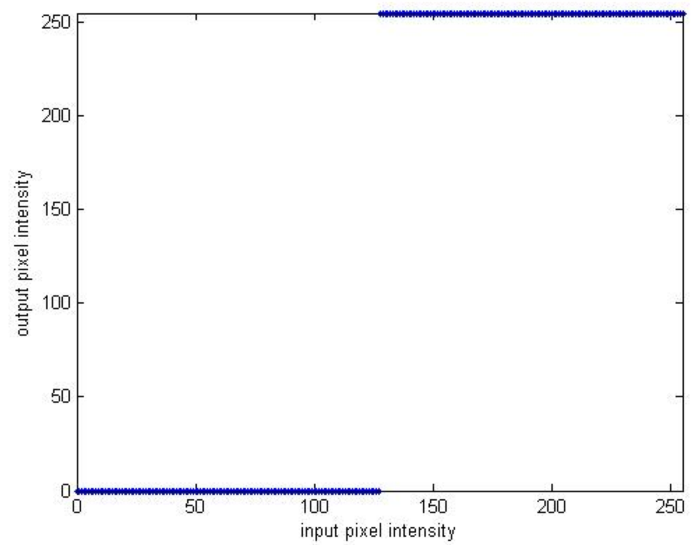
  pixel'(x, y) = 0 *if pixel(x, y) < threshold*

  pixel'(x, y) = 255 *if pixel(x, y) >= threshold*



original image

binary image
(threshold = 128)



Dithering

- In existing black-and-white printers, each pixel can only have 2 levels of intensity (1 bit), i.e., with or without ink.
- Color printers provide 15 levels (or 15 different colors), by combining four types of inks, cyan, magenta, yellow and black (i.e., C, M, Y, B, C+B, M+B, Y+B, C+Y, M+Y, C+M, C+M+Y, C+M+B, C+Y+B, M+Y+B, C+M+Y+B).
- ***Dithering*** is a technique to print more intensity (or color) levels by reducing the printing resolution, i.e., trading off spatial resolution for intensity resolution.
- The idea is to replace and print each image pixel with a dot pattern (multiple dots) such that the average intensity (or color) of the dot pattern approximates the original image pixel intensity.

- Consider printing an 8-bit gray-level image on a black-and-white laser printer. We may replace each image pixel with, say, a 2×2 dot pattern. A 2×2 matrix produces a total of 5 possible dot patterns:

image pixel value maps to range →	0 - 50	51 - 101	102 - 152	153 - 203	204 - 255
code to represent the range →	a	b	c	d	e
pixel pattern for printing →	$\begin{bmatrix} B & B \\ B & B \end{bmatrix}$	$\begin{bmatrix} W & B \\ B & B \end{bmatrix}$	$\begin{bmatrix} W & B \\ B & W \end{bmatrix}$	$\begin{bmatrix} W & W \\ B & W \end{bmatrix}$	$\begin{bmatrix} W & W \\ W & W \end{bmatrix}$

- Given an image pixel, we first map its value (between 0 and 255) into a new value (between a to e). We then map the new pixel value to one of the above 5 pixel patterns.

- An example:

$$\begin{bmatrix} 122 & 69 \\ 180 & 240 \end{bmatrix} \xrightarrow{\text{map each pixel value to a code}} \begin{bmatrix} c & b \\ d & e \end{bmatrix} \xrightarrow{\text{map each code to a pixel pattern}} \left[\begin{array}{cc:cc} W & B & W & B \\ B & W & B & B \\ \hdashline W & W & W & W \\ B & W & W & W \end{array}\right]$$

a 2×2 image

dithered binary image for printing

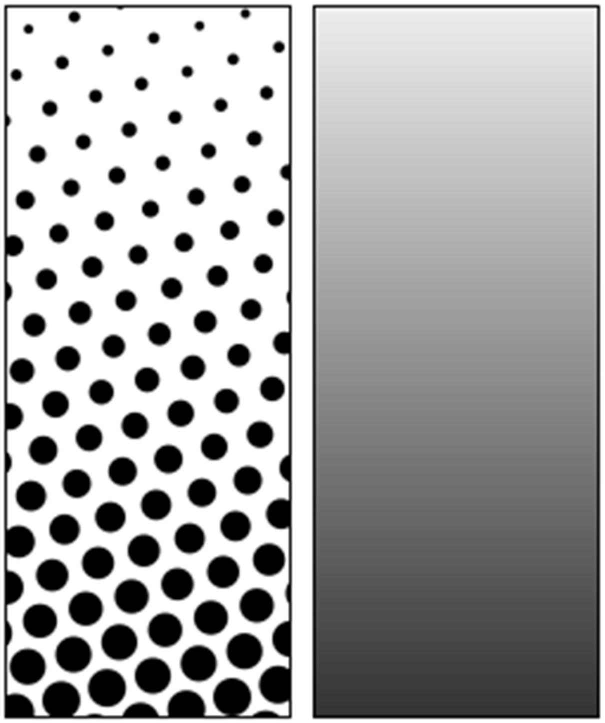
- A dithered example image:

original gray-level image

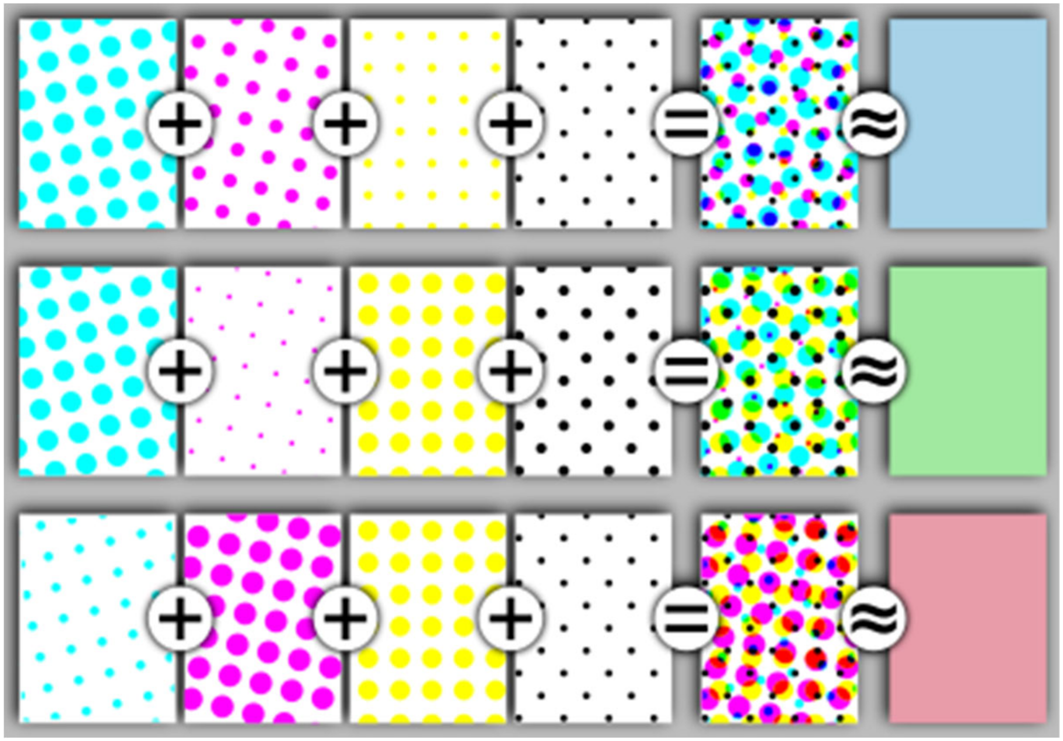
dithered binary image

magnification of a dithered region

- Printer dithering (also called halftoning) examples:



black-and-white dithering



Color dithering

Pixel Per Inch (PPI)

- PPI is a measure of the pixel density. It refers to the display or printing resolution of an input image.
- As suggested by Fotomax (www.fotomaxonline.com/service.asp?lang=en&page=175), the PPI requirement for digital photo printing is 300 PPI for best resolution, or 180 PPI for minimum resolution.

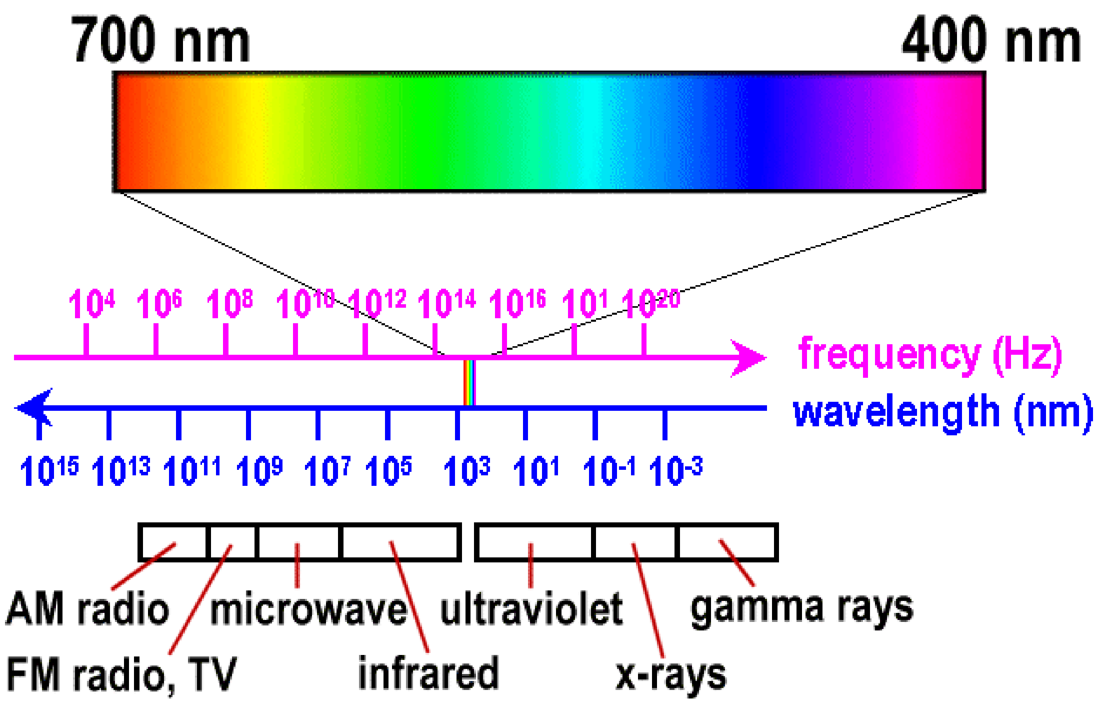
Product	Image Size (Inch)	The Best Resolution (Pixels)	Minimum Resolution (Pixels)
1.5"x2" ID Photo 8 pcs	1.5 x 2	450 x 600	270 x 360
2R	2.5 x 3.5	750 x 1050	450 x 630
3R	3.5 x 5	1050 x 1500	630 x 900
4D (4.5"x6")	4.5 x 6	1350 x 1800	810 x 1080
4R	4 x 6	1200 x 1800	720 x 1080
5R	5 x 7	1500 x 2100	900 x 1260
6R	6 x 8	1800 x 2400	1080 x 1440
8R	8 x 10	2400 x 3000	1440 x 1800
8F	8 x 12	2400 x 3600	1440 x 2160

Dot Per Inch (DPI)

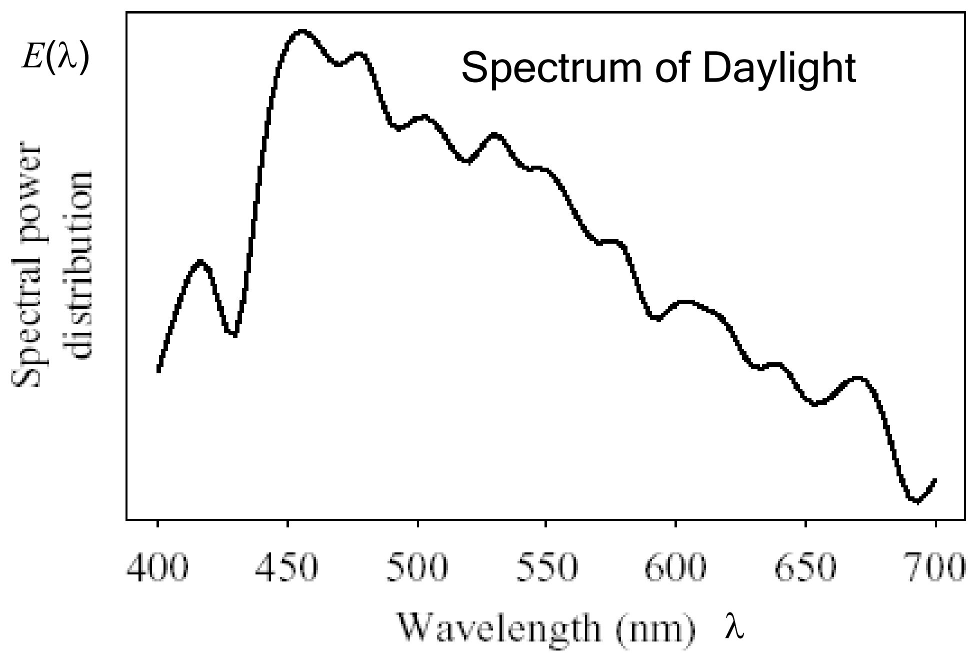
- DPI refers to the printer resolution. It controls the printing quality.
- It specifies the number of droplets of ink that can be printed on one inch of the paper.
- Typical printer DPIs include 360, 720, 1440, 2880, and 5760, but the differences among these settings may be small and not visible to naked eyes. Changing the DPI setting has no effect on the size of the print.
- To print an image, the PPI of the image file is mapped to the DPI of the printer – based on the dithering method that we have discussed.

Light and Spectra

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

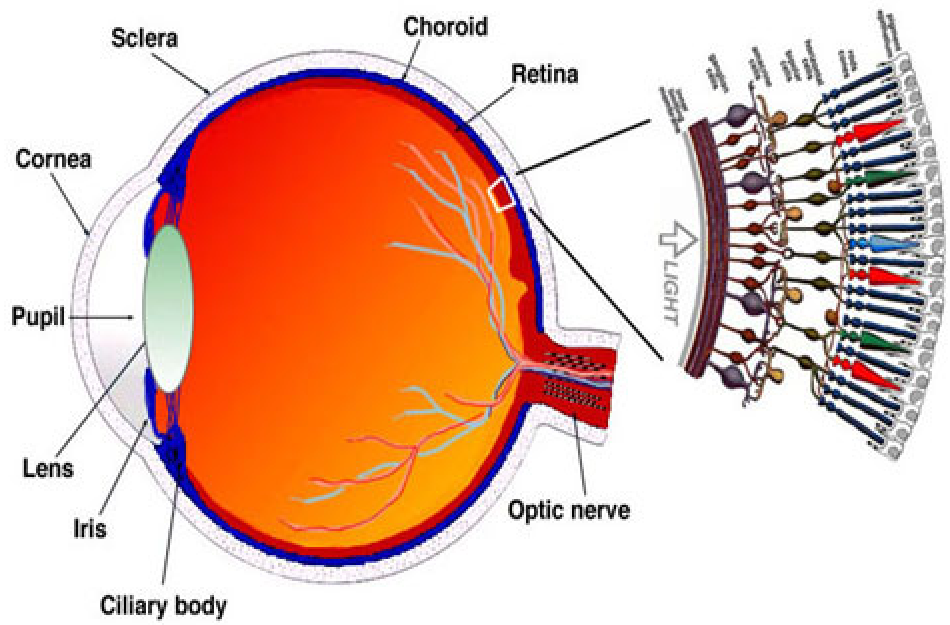


- Spectral Power Distribution (SPD), or *spectrum*, $E(\lambda)$, shows the relative amount of light energy at each wavelength λ.



Human Vision

- Retina consists of an array of rods and three kinds of cones
  - Rods are for sensing brightness (or luminance)
  - Cones are for sensing colors:
    1. L-cones: most sensitive to red light
    2. M-cones: most sensitive to green light
    3. S-cones: most sensitive to blue light

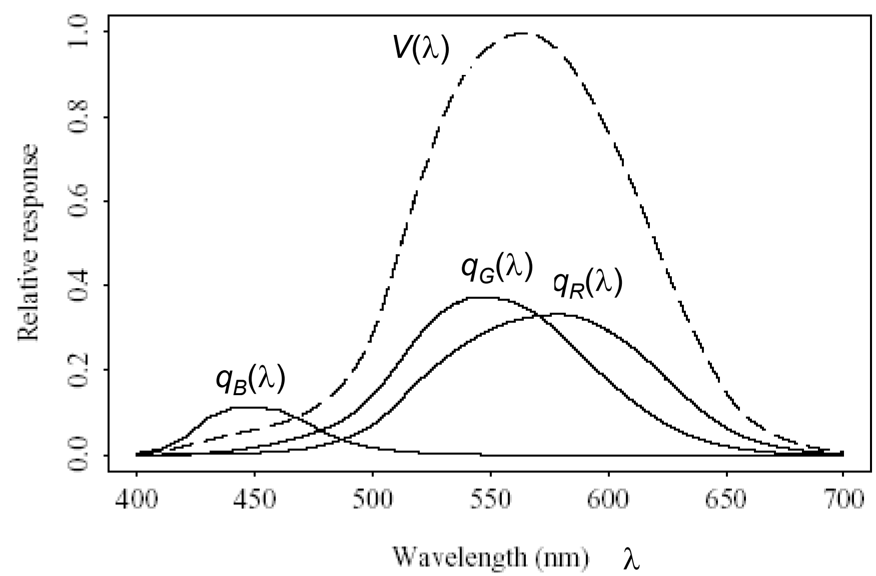


Spectral Sensitivity of the Eye

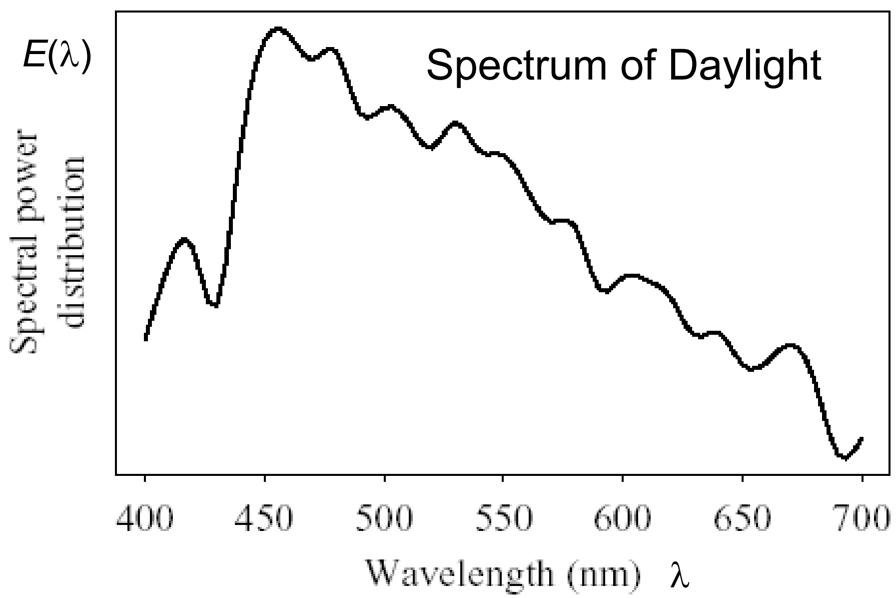
- The eye is most sensitive to light in the middle of the visible spectrum.
- The response of the eye to each color channel is proportional to the corresponding number of cones in the eye:
  - Red Receptor Sensitivity $q_R(\lambda)$
  - Green Receptor Sensitivity $q_G(\lambda)$
  - Blue Receptor Sensitivity $q_B(\lambda)$
- Luminous-efficiency function $V(\lambda)$:

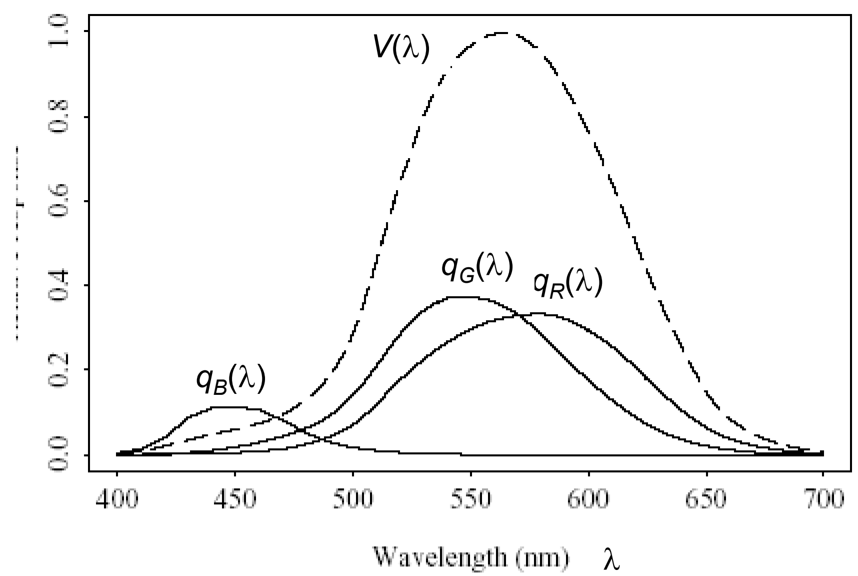
  overall sensitivity formed by the sum of the response curves for Red, Green and Blue.

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



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

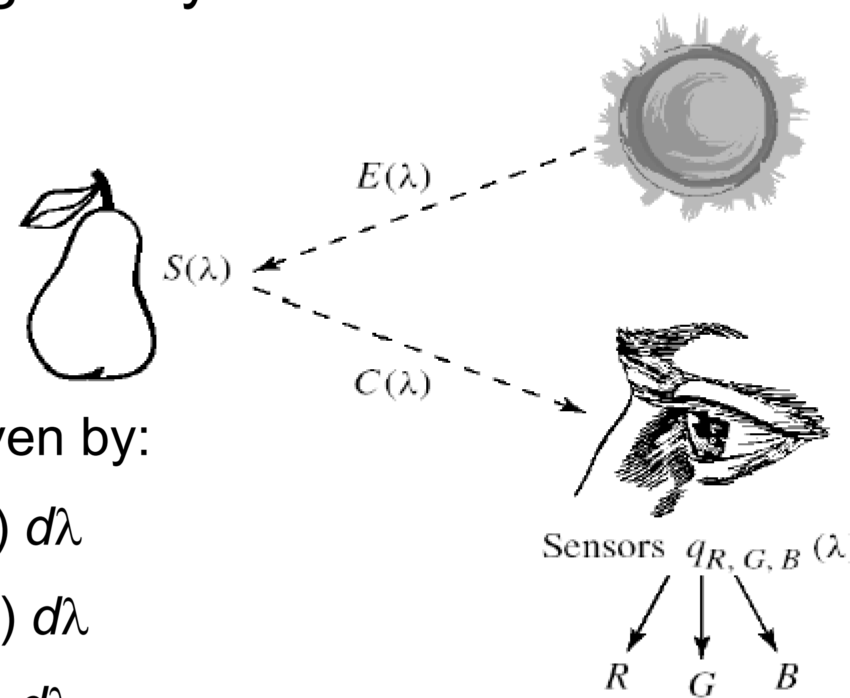




Source: http://escience.anu.edu.au/lecture/cg/Color/index.en.html

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,* $C(\lambda)$, entering the eye/sensor is defined by $C(\lambda) = E(\lambda)\, S(\lambda)$.



Total response on each channel is now given by:

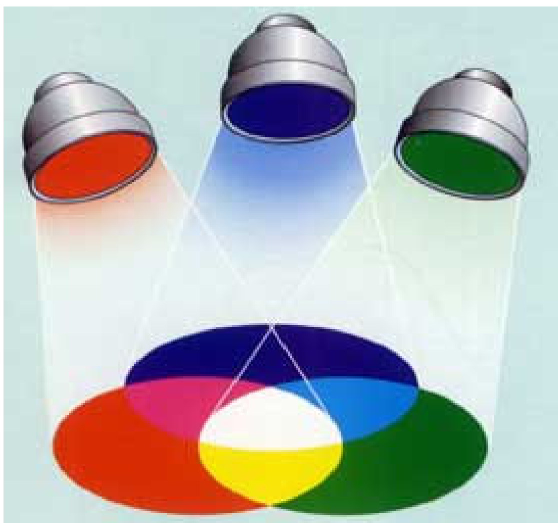
$$R = \int E(\lambda)\, S(\lambda)\, q_R(\lambda)\, d\lambda = \int C(\lambda)\, q_R(\lambda)\, d\lambda$$

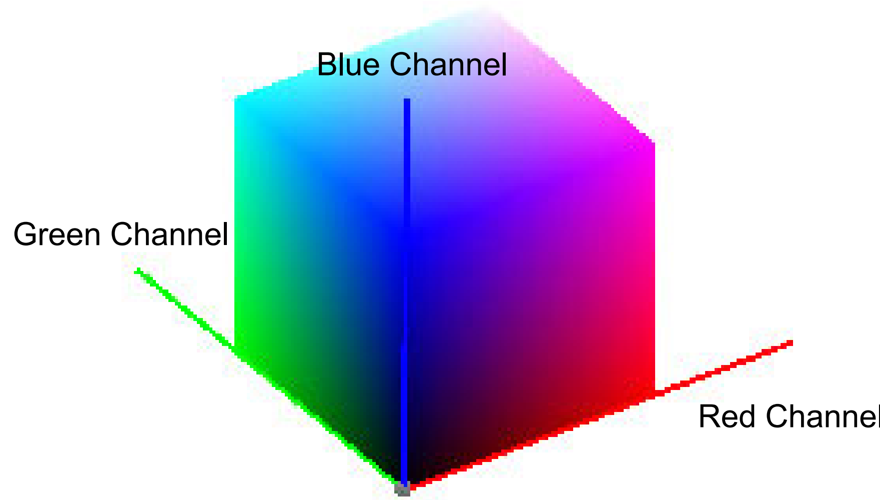
$$G = \int E(\lambda)\, S(\lambda)\, q_G(\lambda)\, d\lambda = \int C(\lambda)\, q_G(\lambda)\, d\lambda$$

$$B = \int E(\lambda)\, S(\lambda)\, q_B(\lambda)\, d\lambda = \int C(\lambda)\, q_B(\lambda)\, d\lambda$$

RGB Color Space

- Different combinations of the red, green, blue channel values result in different colors.





RGB Color Cube

24-Bit Color Images

- In a 24-bit color image, each pixel is represented by three bytes, usually in RGB format.
  - With 1 byte each for R, G, and B, this representation provides a total of 256×256×256 (or 16,777,216) possible colors.
  - However, this will consume a lot of memory. A single HD image of 1920×1080, without any compression, will require 5.93MB to store.
- Many 24-bit color images are actually stored as 32-bit images. The extra byte in each pixel is used to store an *alpha* value representing some special effect information (e.g., transparency).

- An example

24-Bit Color Image

R Channel

G Channel

B Channel

- Another 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

Color Models

- In the previous discussion, we assume that each image is represented by RGB channels.
- Such a representation is called the *RGB color model*.
- Another two popular models are the YUV and YIQ models.
- The YUV color model is used in PAL color TV broadcasting, while the YIQ is used in NTSC color TV broadcasting.
- (PAL was developed by the UK and NTSC by the US. PAL and NTSC are two popular analog video broadcasting standards, used before digital video broadcasting.)

The YUV Model

- In YUV, Y is the luminance signal and is computed as:

$$Y = 0.299R + 0.587G + 0.114B$$

- U and V are the chrominance signals and are computed as:

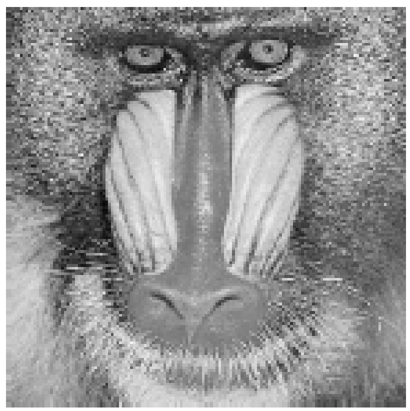
$$U = B - Y$$

$$V = R - Y$$

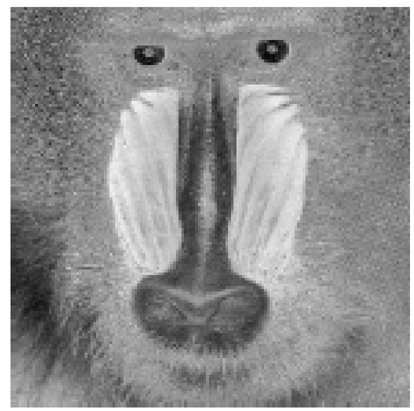
- As human eyes are more sensitive to the luminance signal, a high percentage of the bandwidth can then be allocated to carry the luminance signal in the PAL system.

- An example:

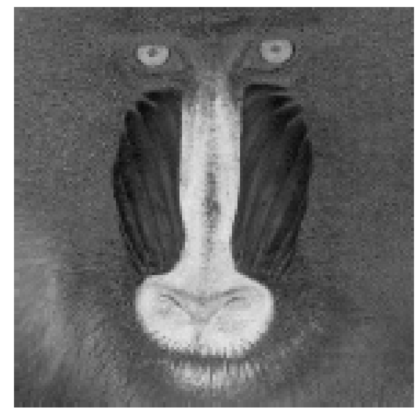
original color image



Y



U



V

The YIQ Model

- In YIQ, *Y* is again the luminance signal and computed in the same way as in YUV.
- *I* and *Q* are also the chrominance signals, but rotated by 33^o as:

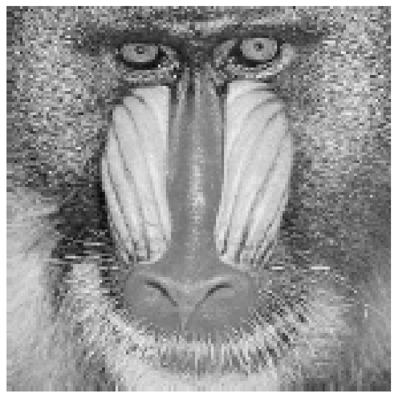
$$I = 0.877(R - Y)\cos 33^o - 0.492(B - Y)\sin 33^o$$
$$= 0.596R - 0.275G - 0.321B$$

$$Q = 0.877(R - Y)\sin 33^o - 0.492(B - Y)\cos 33^o$$
$$= 0.212R - 0.523G - 0.311B$$

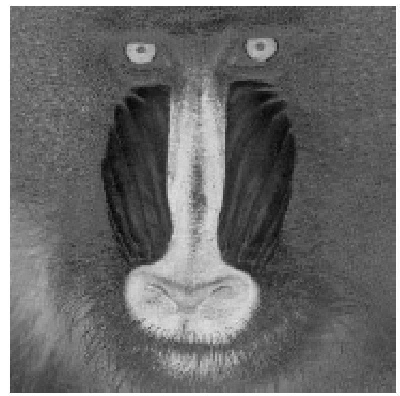
- Similar to PAL, a high percentage of the bandwidth can be allocated to carry the Y signal in the NTSC system.

- An example:

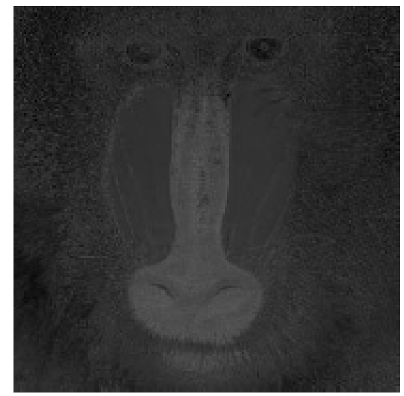
original color image



Y



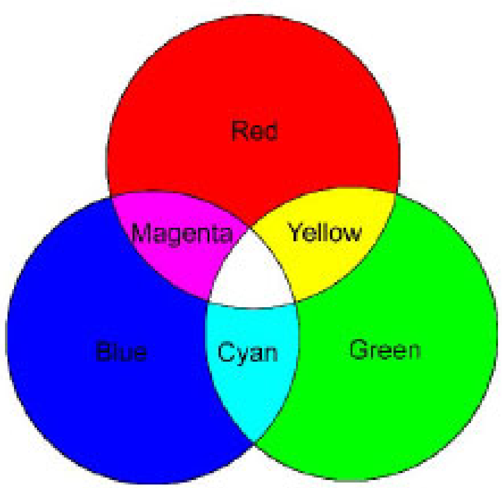
I



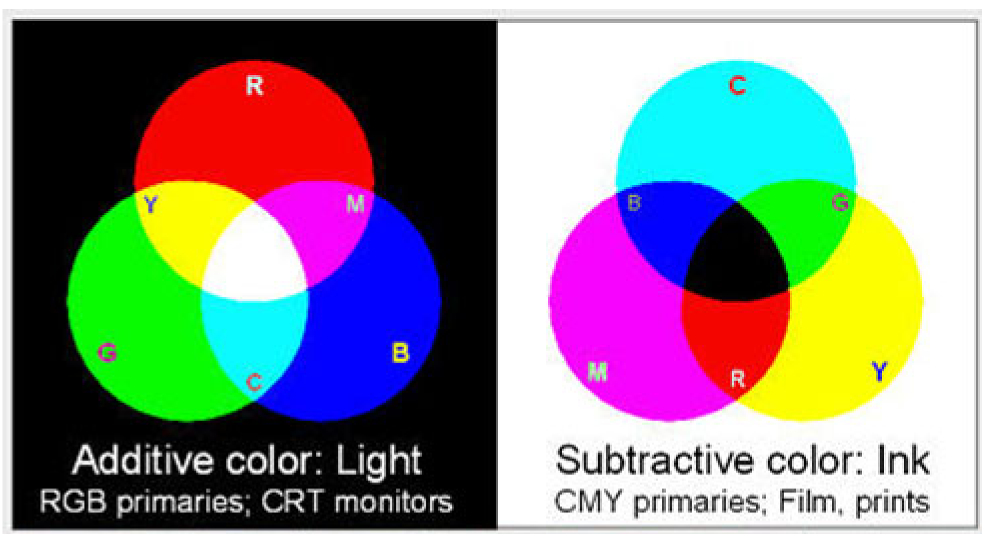
Q

Additive/Subtractive Color Models

- When two light beams hit the same point, their colors add together.
- When two nearby LCD points are turned on, their colors also add together.
- Hence, RGB (and so as YUV and YIQ) is referred to as an additive color model.



- However, in color printing, the opposite situation occurs.
- When depositing a drop of yellow ink, it subtracts blue color from the white paper. Hence, it reflects red and green colors. As a result, it appears yellow.
- So, instead of red/green/blue, we have -red/-green/-blue. These subtractive color primaries then become Cyan (C), Magenta (M) and Yellow (Y).



See:
http://www.youtube.com/watch?v=ygUchcpRNyk