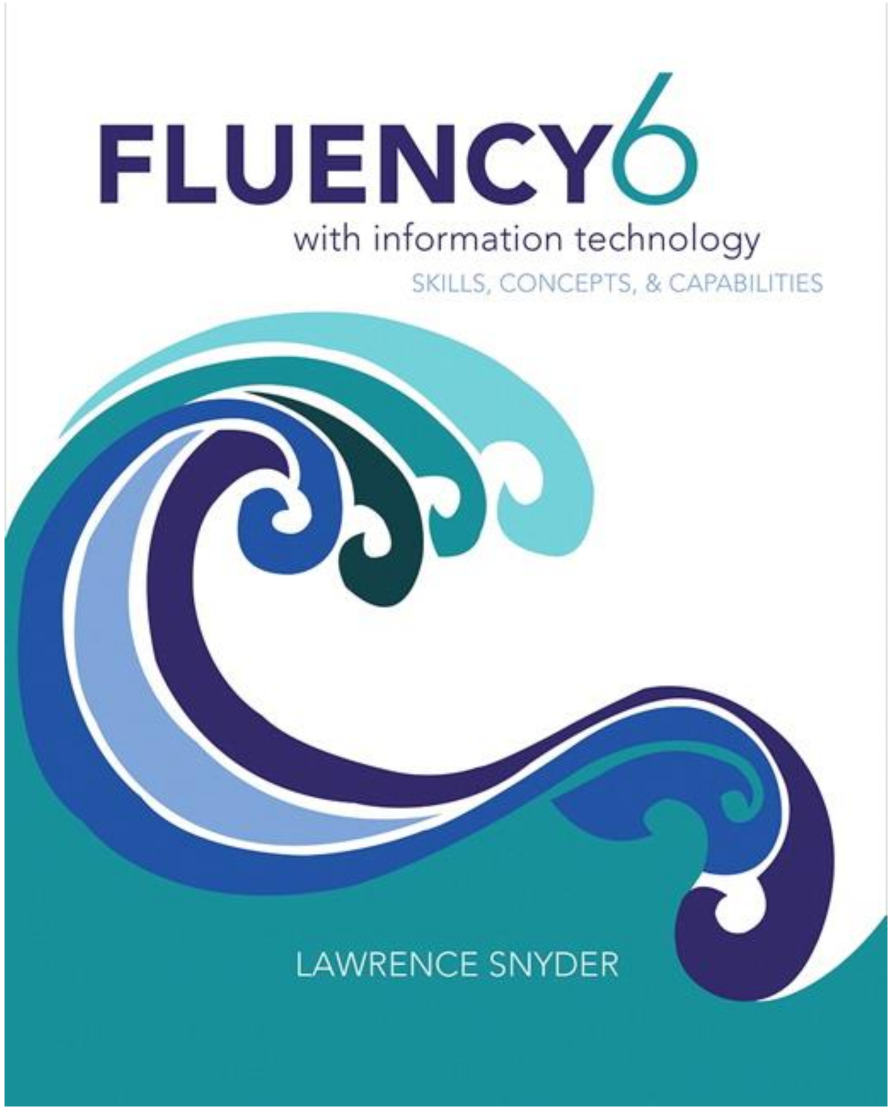


Chapter 8

Representing Multimedia Digitally

Table of Contents

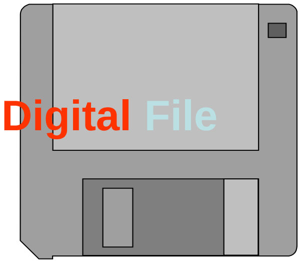
Learning Objectives

- Explain how RGB color is represented in bytes
- Explain the difference between “bits” and “binary numbers”
- Change an RGB color by binary addition
- Discuss concepts related to digitizing sound waves
- Explain data compression and its lossless and lossy variants
- Explain the meaning of the Bias-Free Universal Medium Principle

Digital : 통일된 정보표현방법

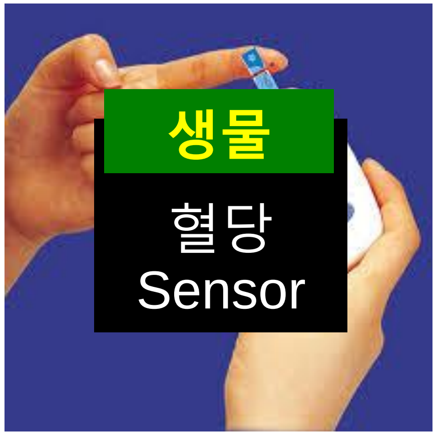
보조자료

**모든 정보는 digital로 표현될 수 있다



- 수
- 글씨(Text)
- 소리(Audio)
- 화면(Image)
- 동화상(Video)

무게 온도 수질 당뇨



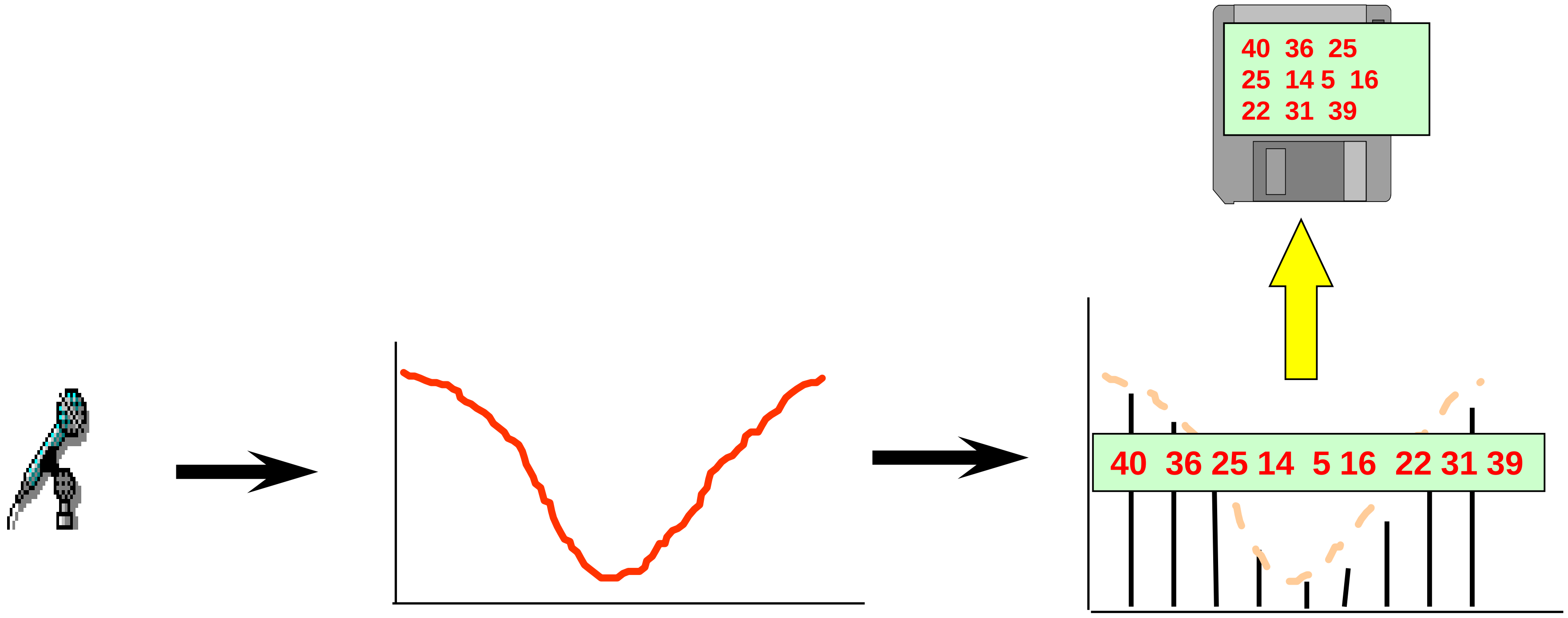
디지털

60 Kg	38 도	6 PH	110 mg/dl

보조자료

모든 Multimedia 정보는 숫자 (digit)로 표현가능!!

Audio Data



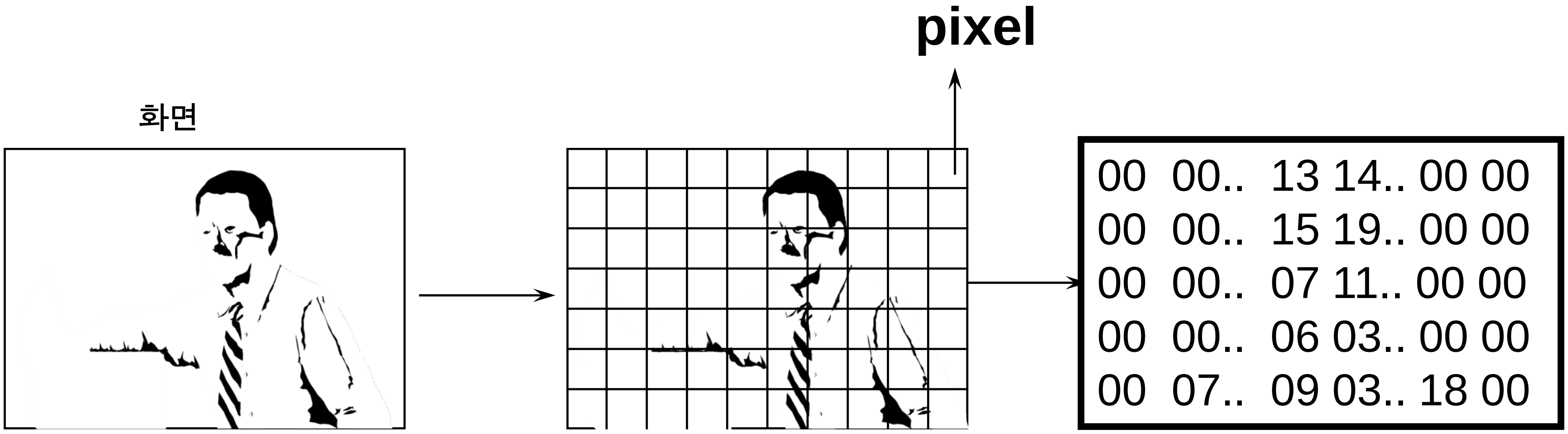
Analog 신호 **Digital 신호**

컴퓨터용 마이크 ==> A/D 변환
컴퓨터용 스피커 ==> D/A 변환

보조자료

모든 Multimedia 정보는 숫자 (digit)로 표현가능!!

Image Data



```
00  00..  13 14.. 00 00
00  00..  15 19.. 00 00
00  00..  07 11.. 00 00
00  00..  06 03.. 00 00
00  07..  09 03.. 18 00
```

각 **pixel** 당 **Black의 density를 나타내는 숫자**

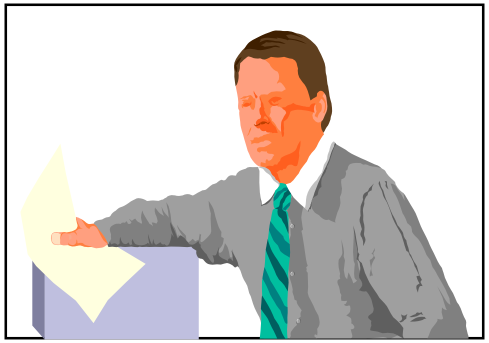
빛의 세기 $\longrightarrow$ **전기 신호 (analog)** $\longrightarrow$ **수치화 (digital)**

보조자료

모든 Multimedia 정보는 숫자 (digit)로 표현가능!!

Color Image Data

화면: **RGB Monitor (빛)**



→

12 4 34 57 99 12
……

56 7 13 44 66 23
12 4 34 57 99 12
……

12 8 12 33 99 12
56 7 13 44 66 23
12 4 34 57 99 12
……

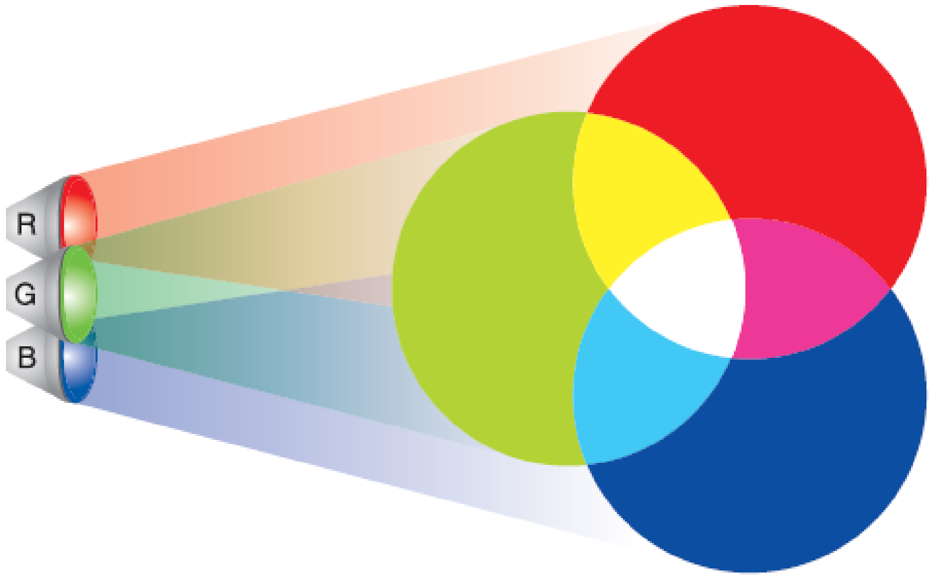
** 동영상(Video) --- 초당 25 - 30 개의 정지화상(image)을 교체

Digitizing Data

- Digitizing is more than letters, numbers, and metadata (chapter 7)
- Digitizing includes other forms of digitized information, known as multimedia (photos, audio, and video) with the same digital encoding principle
- Color on a Computer Display:
  - Pixels are small points of colored light arranged in a grid
  - Each pixel is formed from 3 colored lights: red, green, and blue
    - known as RGB (always in that order)
- Turning on one light at a time, the computer display turns red, green, or blue
- Turning off all of them makes black
- Turning on all of them makes white
- All other colors are made by using different amounts or intensities of the 3 lights

Yellow = R + G?

- Combining red light and green light makes yellow light ! (not orange light?)
- There is a difference between colored light and colored paint
- The paint reflects some colors and absorbs others
  - When white light strikes paint, some light is absorbed (we can't see it) and some light is reflected (we see it)
- In the case of a pixel, the light shines directly at our eyes
  - Nothing is absorbed & Nothing is reflected
  - Just see pure colored light



LCD Display Technology

- At left in the close-up of an LCD is an arrow pointer with two enlargements of it
- From a distance, the pixels appear white
- Close up, the pixels are red, green, and blue colored lights

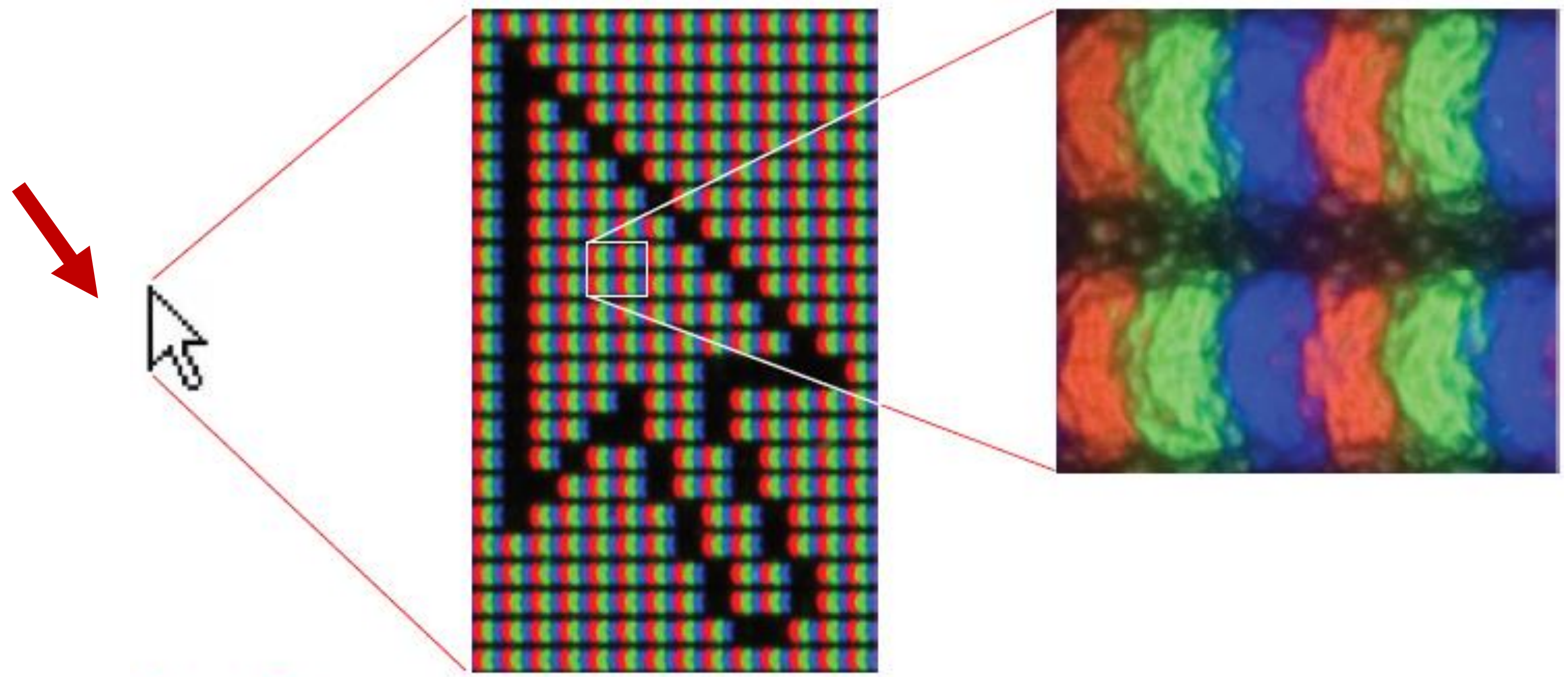


Figure 8.2 A close-up image of an LCD (liquid crystal display) pixel grid; at left is a standard arrow pointer. The enlargement in the middle shows the RGB-triple lights that make pixels. Most are at full intensity creating the white background, and some are turned off (zero intensity) to create the arrow outline. A 2 × 2 white pixel region is further enlarged.

Black and White Colors

- The intensity of RGB light is usually given by a binary number stored in a byte
- Representing the color of a single pixel requires 3 bytes (1 byte for each color)
  - Smallest intensity is 0000 0000 ($\rightarrow$ 0)
  - Largest intensity is 1111 1111 ($\rightarrow$ 255)
- The range of intensity values is 0 through 255 (8 bits $\rightarrow$ 2^8)for each color
- Black is the absence of light:
  - $\underbrace{\text{0000 0000}}_{\boxed{R}}$ $\underbrace{\text{0000 0000}}_{\boxed{G}}$ $\underbrace{\text{0000 0000}}_{\boxed{B}}$ RGB bit assignment for black
- White is the full intensity of each color:
  - $\underbrace{\text{1111 1111}}_{\boxed{R}}$ $\underbrace{\text{1111 1111}}_{\boxed{G}}$ $\underbrace{\text{1111 1111}}_{\boxed{B}}$ RGB bit assignment for white

Color Intensities

- Consider blue (0000 0000   0000 0000   1111 1111)

  R   G   B

- The 8 bits specifying its intensity have position values
- If we want the sub pixel to at half intensity: each bit contributes half as much power as the bit to its left

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0

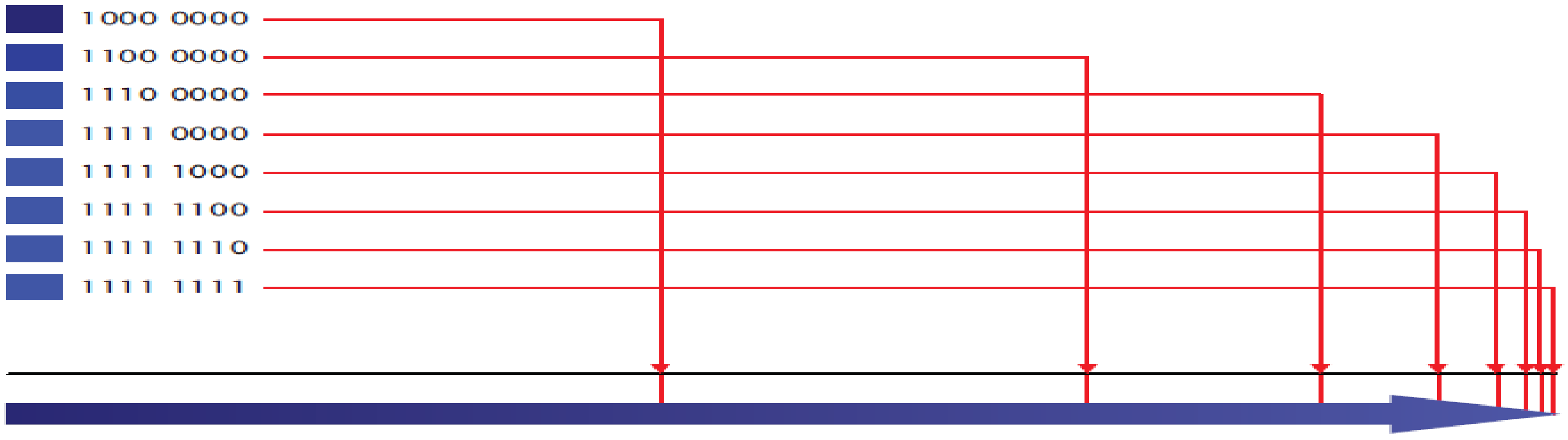


Figure 8.3 Progressively increasing the intensity for a blue subpixel; each bit contributes half as much power as the bit to its left.

Decimal to Binary

- The 10 bit table for a decimal number 0 ~1024 to a binary number

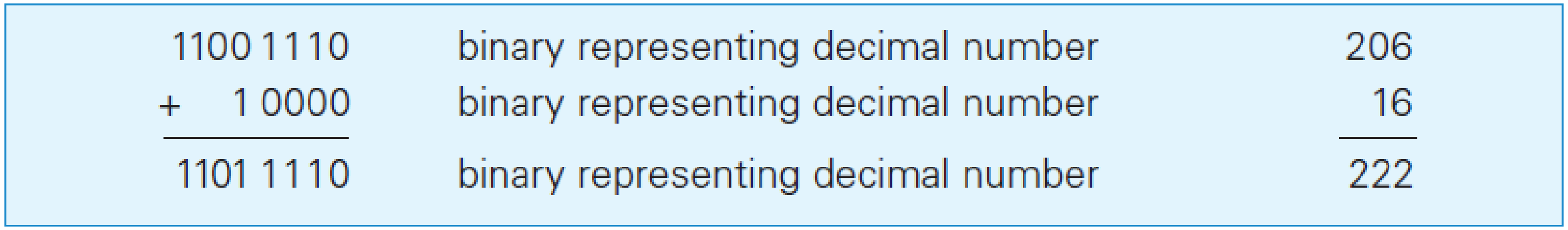
Number being converted	356									
Place value	512	256	128	64	32	16	8	4	2	1
Subtract										
Binary Number										

- Let's call the number being converted "nbc" and the place value "pv" respectively
- <u>If nbc ≥ pv</u>, enter 1 into the binary number row & put the value of nbc– pc into the subtract row, <u>otherwise</u> enter 0 into the binary number row & move to the next slot

Number being converted	356	365	109	109	45	13	13	5	1	1
Place value	512	256	128	64	32	16	8	4	2	1
Subtract		109		45	13		5	1		0
Binary Number	0	1	0	1	1	0	1	1	0	1

Lighten Up

- Changing Colors by Addition
  - –To make a lighter color of gray, we change the common value to be closer to white

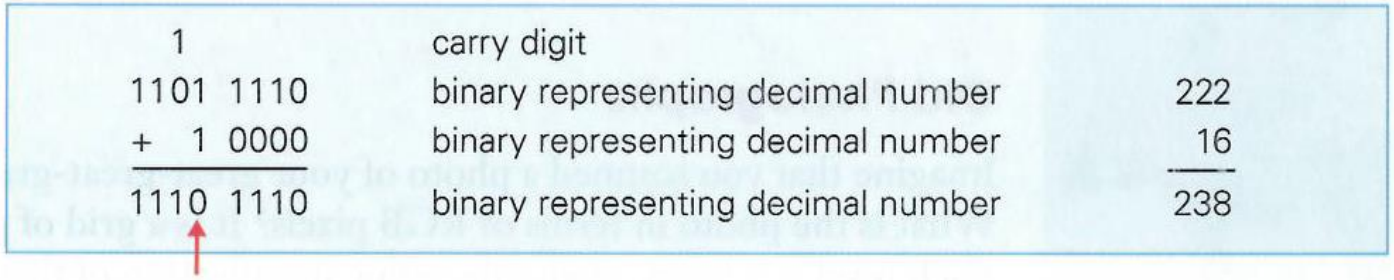


1100 1110	binary representing decimal number	206
+ 1 0000	binary representing decimal number	16
1101 1110	binary representing decimal number	222

Figure 8.4 Adding 16 to a binary value to lighten RGB intensities.

Lighter Still (좀더 밝게)…

- Make the color lighter still by another 16 units of intensity for each RGB byte
- The 16’s position is already filled with a 1: 1101 1110
- “Carry” to the next higher place



1	carry digit	
1101 1110	binary representing decimal number	222
+ 1 0000	binary representing decimal number	16
1110 1110	binary representing decimal number	238

Figure 8.5 Adding 16 to the binary representation of the intensity 222, requiring a carry.

Binary Addition

- Same as decimal addition but with only 2 digits
- Work from right to left, adding digits in each place position, writing the sum below
- Like decimal addition, there are two cases:
  - Add the two numbers in a place and the result is expressed as a single digit
  - Add two numbers in a place and the result requires carrying to the next higher place

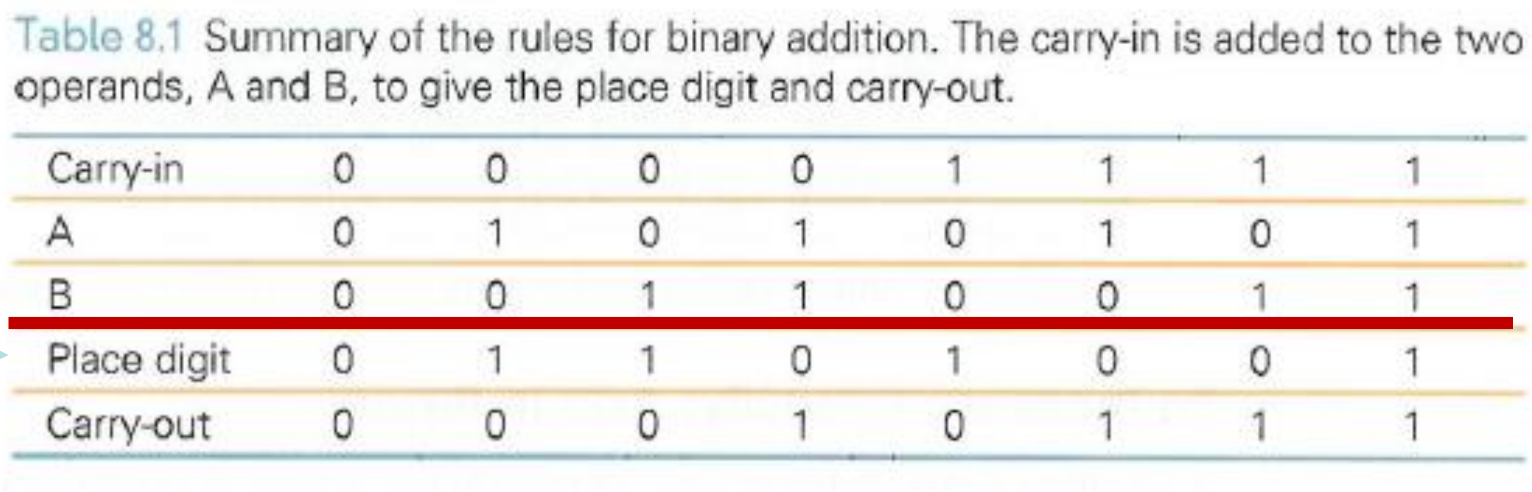


Table 8.1 Summary of the rules for binary addition. The carry-in is added to the two operands, A and B, to give the place digit and carry-out.

Carry-in	0	0	0	0	1	1	1	1
A	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1
Place digit	0	1	1	0	1	0	0	1
Carry-out	0	0	0	1	0	1	1	1

Computing on Representations

- When digital information is changed through computation, it is known as computing on representations
- For example: changing the brightness and contrast of a photo
- Brightness refers to how close to white the pixels are
- Contrast is the size of difference between the darkest and lightest portions of the image

Levels Graph

- Photo manipulation SW often gives the values of the pixels in a Levels graph
- 0 percent is called the black point, or 00 00 00 (Hexa)
- 100 percent is the white point, or FF FF FF (Hexa)
- The midpoint is called the gamma point and it is the midpoint in the pixel range

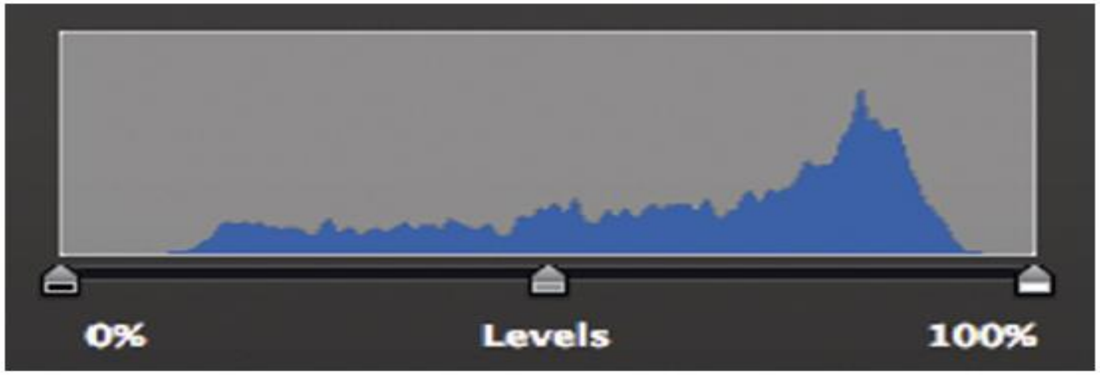
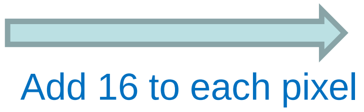


Figure 8.6 The Levels graph for the GGGM photo; the horizontal axis is the 256 pixel values, and the vertical axis is the number of pixels in the image with that value.

Brightness

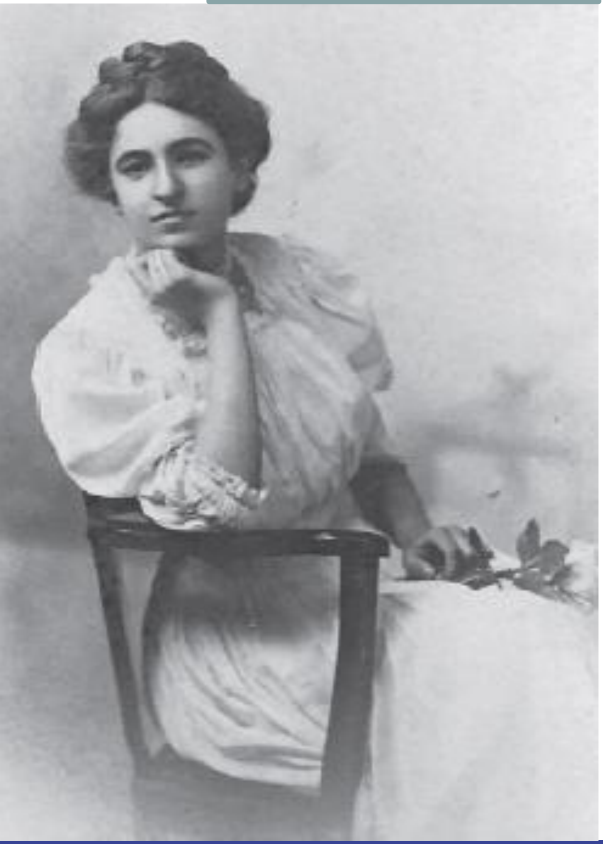
- We want all the pixels to be closer to intense white, but to keep their relative relationships ➔ Ex. Add 16 to each pixel
- A pixel which is RGB (197, 197, 197) becomes RGB (213, 213, 213)



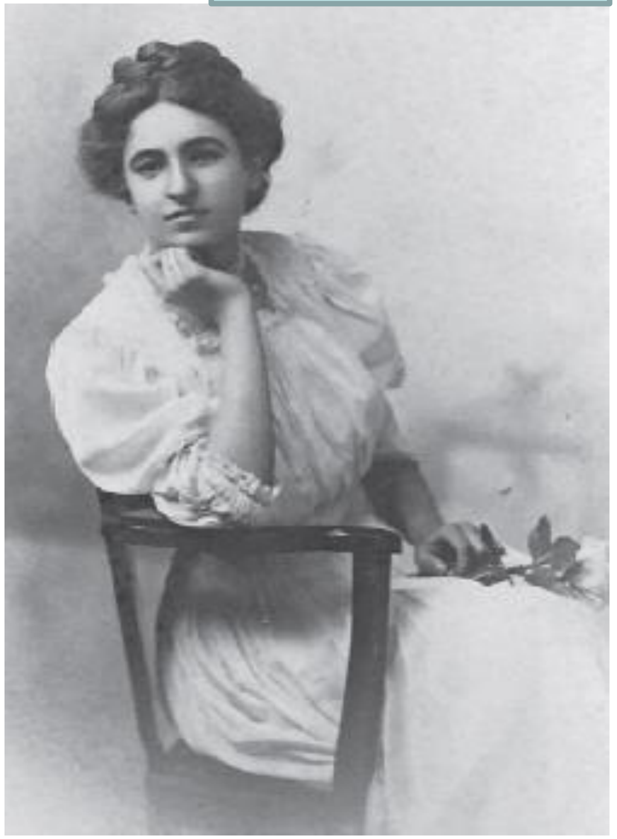
Contrast

- Goal is not to shift the Levels graph right, but rather to "stretch the Level graph out" toward the right

Original Photo



More Contrasted Photo



Add a smaller amount for dark pixels

Add a larger amount for light pixels

New Levels Graph

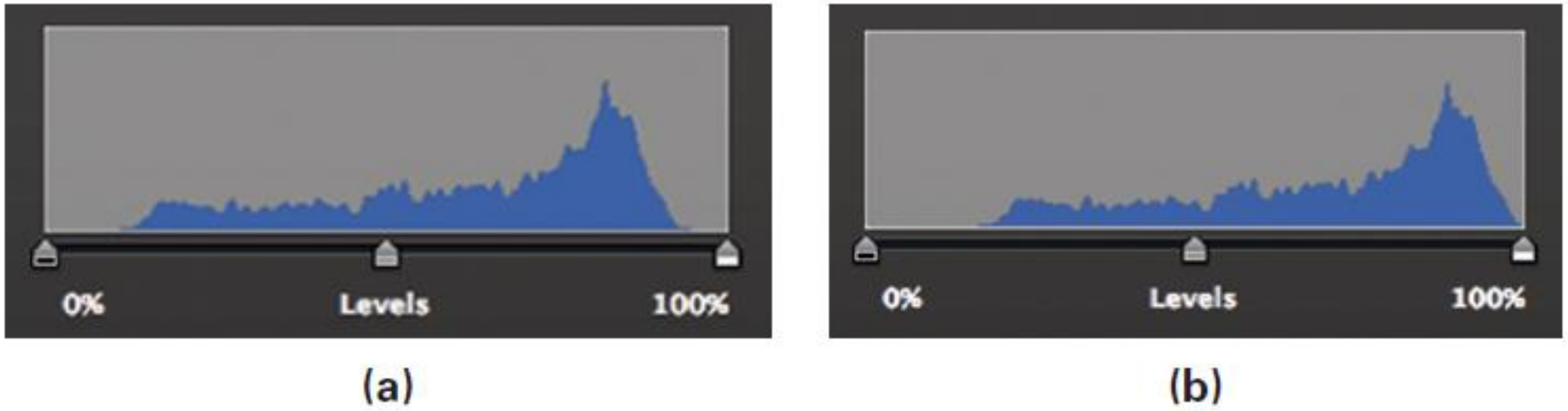


Figure 8.8 Levels graph for GGGM photo; (a) original and (b) after brighten-ing by adding 16; notice that the graph is simply shifted right by 16.

for each P_0 $\Longrightarrow$ creare new P_n

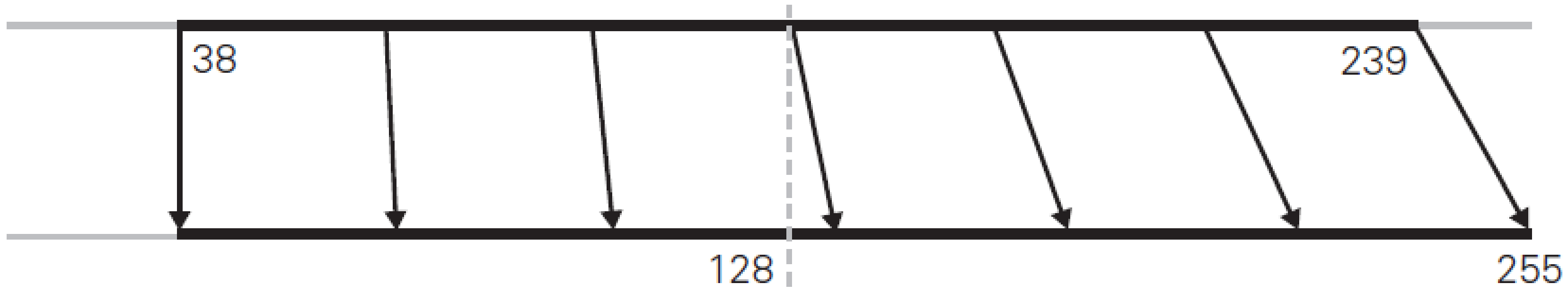


Figure 8.9 "Stretching" the pixel values right from a range [38–239] to [38–255].

New Levels Math

- For every original pixel Po, subtract the amount of the lower end of the range: Po – 38
  - That tells how much to increase each pixel position; smaller (darker) numbers get lightened less than larger (lighter) numbers
- Then we multiply by the size of the new interval divided by the size of the old interval $\frac{(255-38)}{(239-38)} = \frac{217}{210}$ -

  $\rightarrow$ 1.08
- Add the low end of the original range back in again to return each pixel to its new position along the second line
- The equation for the value in each pixel position of the new image:

  Pn = (Po – 38)*1.08 + 38. Round the answer to a whole number
- Try it yourself!
  - For original pixel 239, did you get 255?
    For original pixel 157, did you get 167?

Adding Color [1/3]

- Whenever the 3 bytes (1byte for one color of RGB) differ in value, there is color ➔ 2^8 X 2^8 X 2^8 colors!
- "highlights" are the lightest 25 percent of the pixels, and "shadows" are the darkest 25 percent of the pixels
- Must count the pixels to know those values:
  - There are 600 × 800 = 480,000 pixels in the image
- Pick the lowest pixel value and go up to the next level and keep adding until you have approximately ¼ of the total pixels (in this case 120,000)
- Pick the highest pixel value and go down to the next level, adding until you have the top ¼ of the total pixels

Adding Color [2/3]

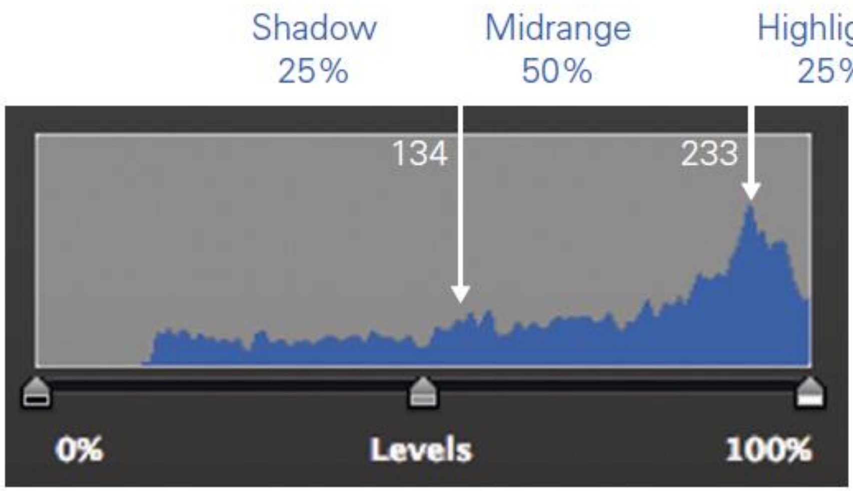
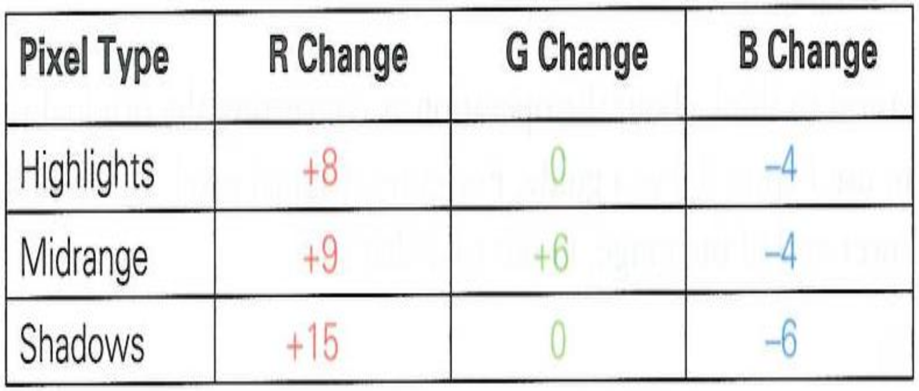


Figure 8.10 The Levels graph for the image in Figure 8.7(c), and boundaries between the highlights, midrange, and shadows.

480,000 Pixels

Highlight 25%	[255–234]	121,339
Midrange 50%	[233–135]	239,540
Shadows 25%	[134–38]	119,121



Pixel Type	R Change	G Change	B Change
Highlights	+8	0	–4
Midrange	+9	+6	–4
Shadows	+15	0	–6

Adding Color [3/3]

- Simple algorithm to colorize an image:

- For each pixel, get the red sub pixel (anyone will work) and check its range and classify them into 3 groups

- Using the color modifications given above for that portion of the image, adjust the color of each sub pixel

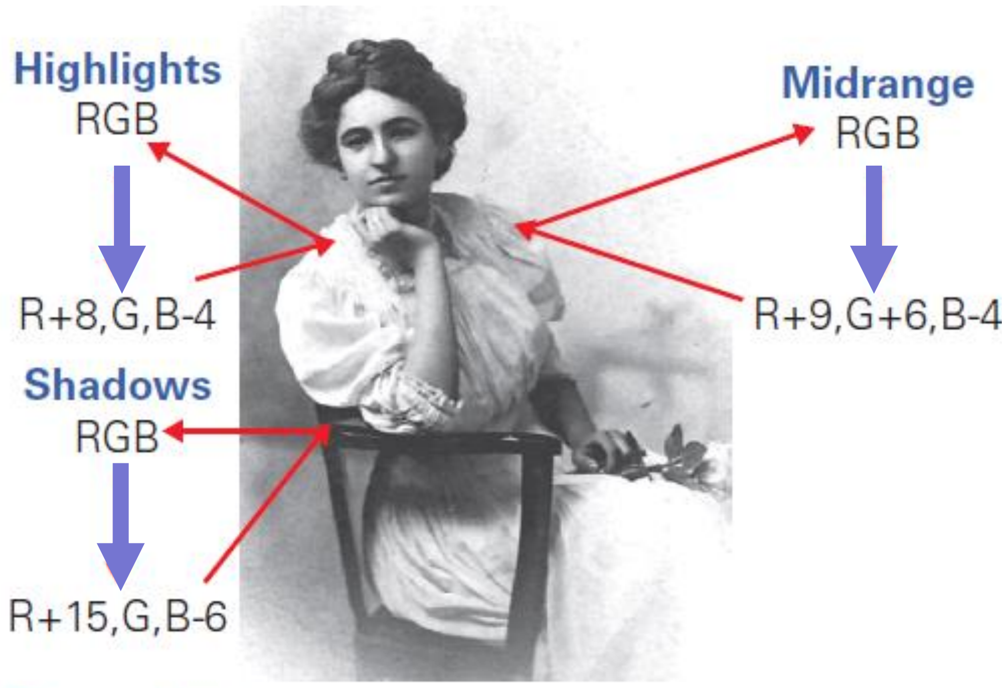
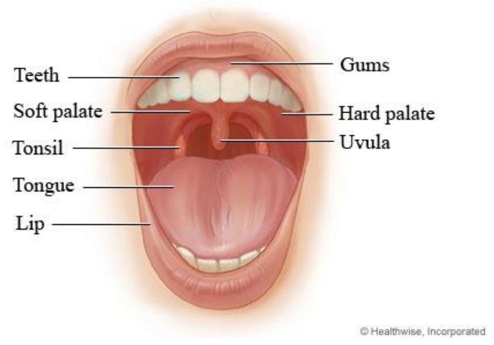


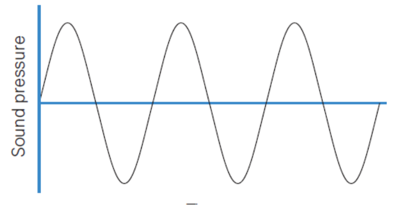
Figure 8.11 The gray is changed to sepia by checking each pixel, deciding if it's a highlight, midrange or shadow, and replacing the pixel with a corresponding revised RGB assignment.

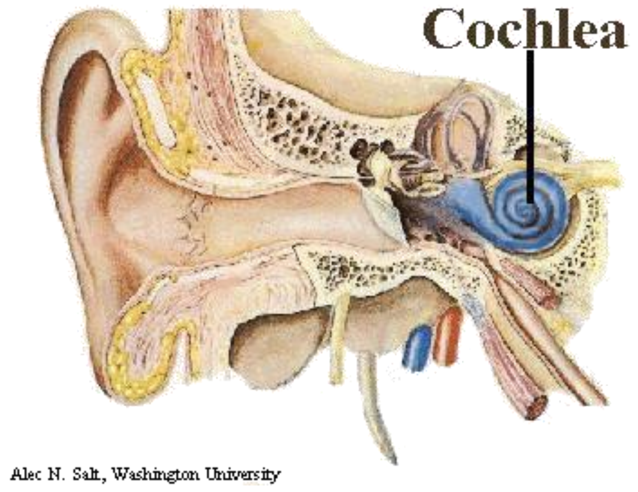
Sepia: 오징어 먹물색, 암갈색

Digitizing Sound [1/2]

- An object creates sound by vibrating in a medium (such as air)
- Vibrations push the air causing pressure waves to emanate from the object, which in turn vibrate our eardrums
- Vibrations are then transmitted by 3 tiny bones to the fine hairs of our cochlea, stimulating nerves that allow us to sense the waves and "hear" them as sound







- Emanate(발산하다, 방사하다), Eardrum(고막), Cochlea(달팽이관)

Digitizing Sound [2/2]

- The force, or intensity of the push, determines the volume
- The frequency is the number of waves per second
- The pitch is the frequency of the pushes

continuous (analog) representation of the wave

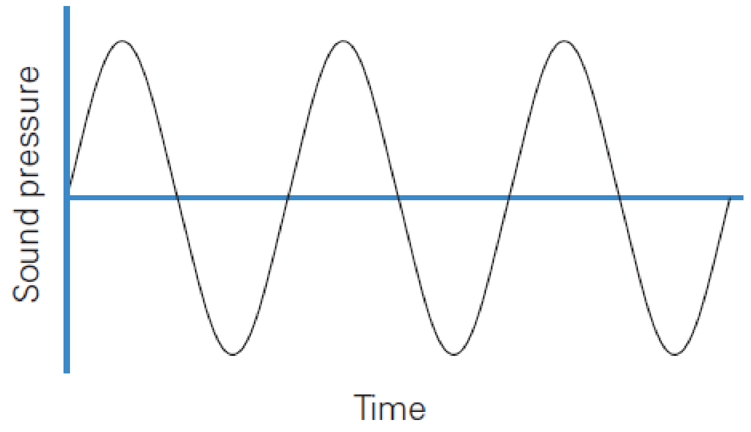


Figure 8.12 Sound wave. The horizontal axis is time; the vertical axis is sound pressure.

- To digitize you must convert sound to bits
- For a sound wave, use a binary number to record the amount that the wave is above or below the 0 line at a given point on our graph

Analog to Digital

- At what point do you measure?

–There are infinitely many points along the line, too many to record every position of the wave

- Sample or take measurements at regular intervals
- Number of samples in a second is called the sampling rate
- The faster the rate, the more accurately the wave is recorded

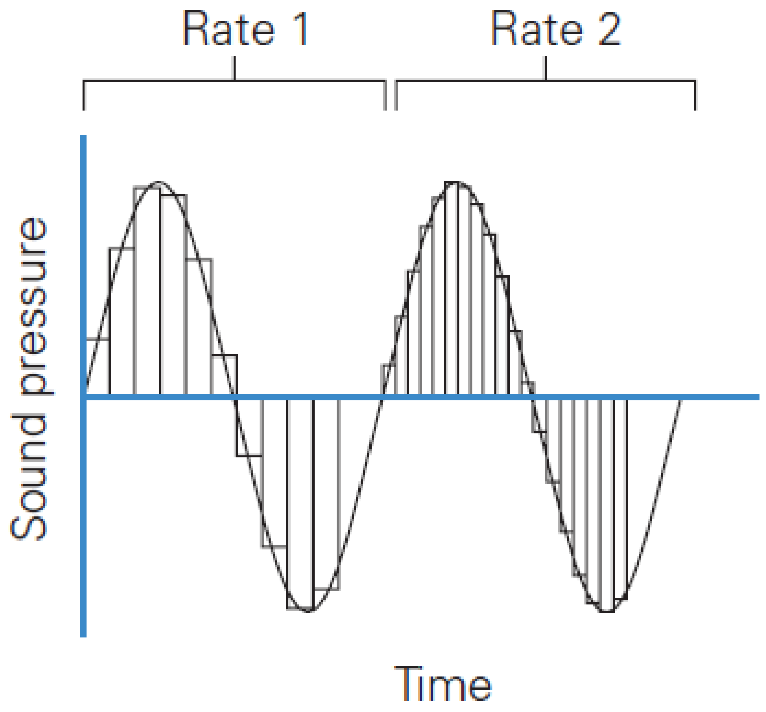


Figure 8.13 Two sampling rates; the rate on the right is twice as fast as that on the left.

Nyquist Rule for Sampling

- If the sampling were too slow, sound waves could “fit between” the samples and you would miss important segments of the sound
- The Nyquist rule says that a sampling rate must be at least twice as fast as the fastest frequency
- Because humans can hear sound up to roughly 20,000 Hz, a 40,000 Hz sampling rate fulfills the Nyquist rule for digital audio recording
- For technical reasons a somewhat faster-than-two-times sampling rate was chosen for digital audio (44,100 Hz)

보조자료

인간의 가청주파수

- 가청주파수 = Audio Frequency Band
- Hz: 1초당 발생하는 cycle의 개수
- 인간의 가청주파수: 20 — 20000 Hz
  - 즉 1초당 wave가 20개에서 20000개정도 생기는 audio를 인간은 들을수 있다
- Digital audio에서는 초당 Sampling을 40000개 정도 하면 인간의 귀에 Analog audio와 가장 유사하게 들린다!

Digitizing Process

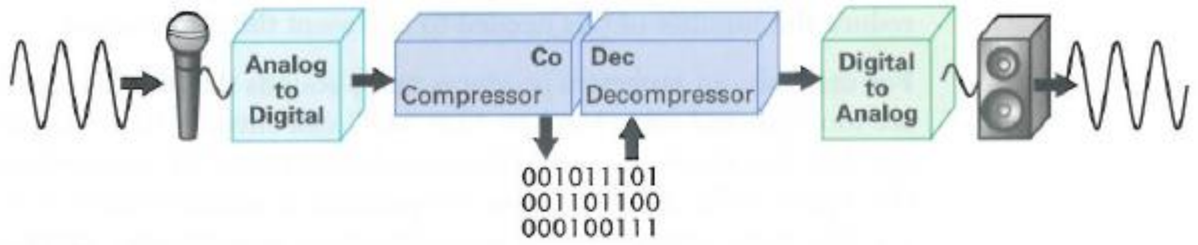


Figure 8.14 Schematic for analog-to-digital and digital-to-analog conversion.

- The digitizing process works as follows:
  - Sound is picked up by a microphone (transducer)
  - Signal (electrical wave) is fed into an analog-to-digital converter (ADC), which takes the continuous wave and samples it at regular intervals, outputting for each sample binary numbers to be written to memory
  - The process is reversed to play the sound: The numbers are read from memory into a digital-to-analog converter (DAC)
  - Electrical wave (signal) created by interpolation between the digital values (filling in or smoothly moving from one value to another)
  - The electrical signal is then input to a speaker which converts it into a sound wave

How Many Bits per Sample?

- To make the samples perfectly accurate, you need an unlimited number of bits for each sample
  - We can only get an approximate measurement
- Bits must represent both positive and negative values
  - –Wave has both positive and negative sound pressure

- If another bit is used, the sample would be twice as accurate
- More bits yields a more accurate digitization
- Audio digital representation uses 16 bits

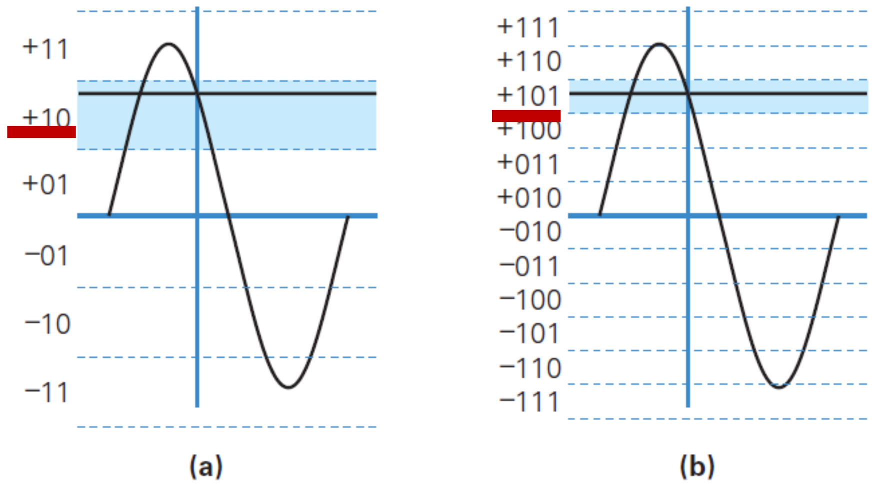


Figure 8.15 (a) Three-bit precision for samples requires that the indicated reading is approximated as +10. (b) Adding another bit makes the sample twice as accurate.

Advantages of Digital Sound

- A key advantage of digital information is the ability to compute on the representation
- One computation of value is to compress the digital audio or reduce the number of bits needed
  - What about sounds that the human ear can't hear because they are either too high or too low?
  - MP3 is really a form of computing on the representation which allows for compression (with a ratio of more than 10:1)
- Another key advantage of digital representations is that digital can be reproduced exactly

Digital Images and Video

- An image is a long sequence of RGB pixels
- The picture is 2 dimensional, but think of the pixels stretched out one row after another in memory
- Example:
  - Suppose a 8 (inch) × 10 (inch) image picture has 300 pixels/inch
  - That's 80 square inches: If each square inch requires 300 × 300 = 90,000 pixels, the image needs 7,200,000 pixels = 7.2 million pixels = 7.2 megapixels
  - At 3 bytes per pixel, it takes 21.6 MB (3 * 7.2 million) of memory to store one 8 (inch) × 10 (inch) color image
  - Sending a picture across a standard 56 Kbyte/sec phone connection would take at least (21.6MB) / 56KB = 3,085 seconds (or more than 51 minutes)

Image Compression

- Typical monitor has fewer than 100 ppi (pixels per inch)
  - In the previous example, one inch accommodates 300 pixels.
  - Storing the picture at 100 ppi monitor is a factor of nine (1/9) savings immediately
  - Even so, sending the picture still requires more than 5.5 mins using the 56 Kbyte/sec Phone connection
- Compression means to change the representation in order to use fewer bits to store or transmit information
  - Use run-length encoding to specify how long the first sequence (run) of 0's is, then how long the next sequence of 1's is, then how long the next sequence of 0's is, then …
  - "011011011011011" ➔ "5"011

–참고: FAX (Efficient application of run-length encoding)

- a sequences of 0's and 1's that encode where the page is white (0) or black (1)

Compression

- Run-length encoding is "lossless" compression scheme
  - The original representation of 0's and 1's can be perfectly reconstructed from the compressed version
- The opposite of lossless compression is "lossy" compression
  - The original representation cannot be exactly reconstructed from the compressed form
- MP-3 is probably the most famous compression scheme
  - MP3 is lossy because the high notes cannot be recovered
- JPG (or JPEG) is a lossy compression for images
  - Exploits the same kinds of "human perception" characteristics that MP-3 does, only for light and color

JPEG Compression

- Humans are quite sensitive to small changes in brightness (luminance 밝기의 차이)
    - Brightness levels must be preserved between uncompressed and compressed versions
- People are not sensitive to small differences in color (chrominance 색의 질 차이)
- JPEG (Joint Photographic Experts Group) is capable of a 10:1 compression without detectable loss of clarity simply by keeping the regions small

Image를 작은 rectangle로 나누어 한rectangle을 한가지 대표 색깔로 표현. ➔ Pixel별로 표현할때보다 적은 bit수로 가능

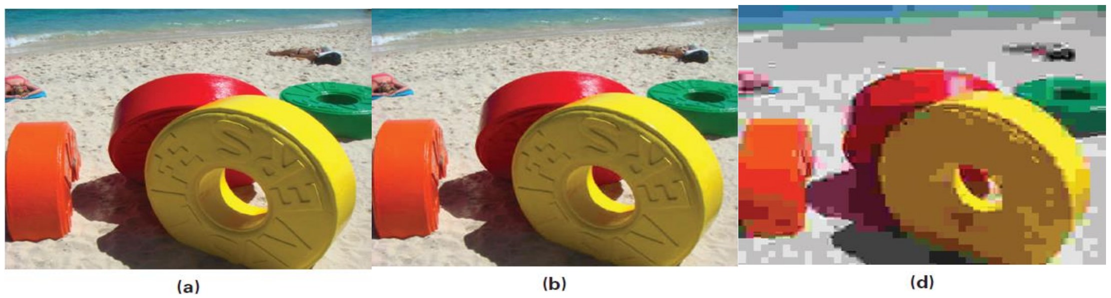


Figure 8.16 Life Saver Sculpture at the Beach (400 × 300 pixels); (a) original 202 KB, (b) 10:1 compression (20 KB), (d) 25:1 ratio (8 KB).

- The benefit of greater than 10:1 compression is smaller files
    –Eventually the picture begins to “pixelate” or get “jaggies”

– Pixelate: 픽셀이 나타나다, Jaggy = Jagged: 톱니자국이 나는

MPEG Compression

- MPEG is the same idea applied to motion pictures
- It seems like an easy task
  - Each image/frame is not seen for long
  - Couldn’t we use even greater levels of single-image compression?
  - It takes many “stills” to make a movie
- In MPEG compression, JPEG-type compression is applied to each frame
- “Interframe coherency” is used
  - Two consecutive video images are usually very similar, MPEG compression only has to record and transmit the “differences” between frames
  - Resulting in huge amounts of compression

Optical Character Recognition (OCR)

- Very sophisticated technology that enables a computer to "read" printed characters
- OCR's business applications include:
  - U.S. Postal Service processing up to 45,000 pieces of mail per hour (2% error rate)
  - In banking, the magnetic numbers at the bottom of checks have been read by computers since the 1950s

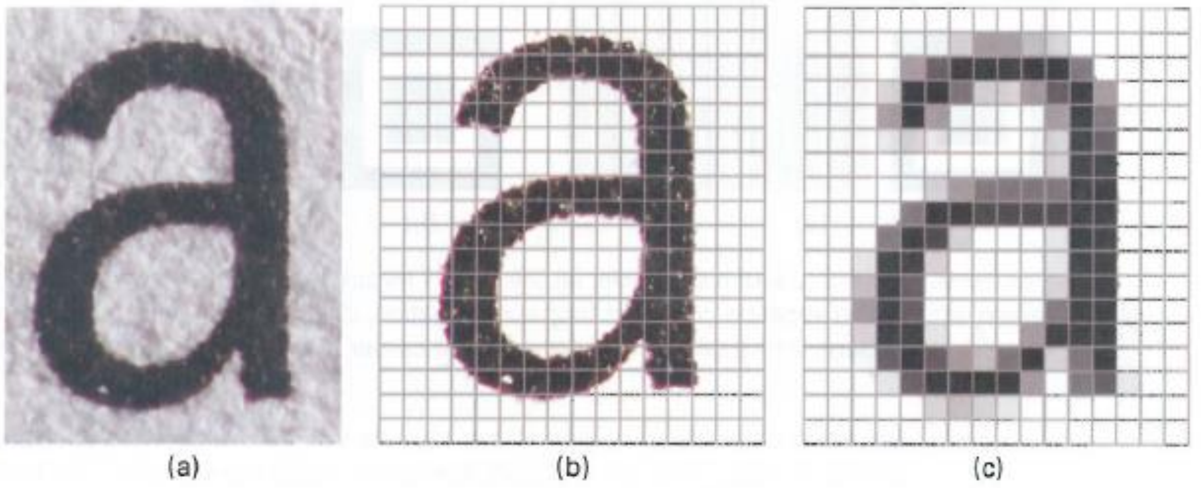


Figure 8.18 Lowercase, sanserif a: (a) shows the analog letter as it appears on the printed page; (b) the letter overlaid by a grid showing where the image will be sampled; and (c) shows the letter in its digital form as it might appear after being captured by a scanner or camera. Each pixel is an estimate of how dark the corresponding area is.

sanserif font: 산세리프 서체

PEARSON

The familiar QR Code is a two-dimensional, optically readable encoding that is more versatile than bar codes. Developed in 1994 by Denso Wave, a subsidiary of Toyota, to track parts in factories, the QR Code was ingeniously designed to be read quickly and in different orientations. QR Codes can handle Japanese Kanji characters, and they allow four different levels of

Capacity

Data Type	*Max Content**
Numeric only	7,089 characters
Alphanumeric	4,296 characters
Binary (8-bit)	2,953 bytes
Kanji/Kana	1,817 characters

*Largest size, smallest error correction

Error Correction Options

EC Level	*% Restored**
L	7%
M	15%
Q	25%
H	30%

*Amount of loss that can be recovered

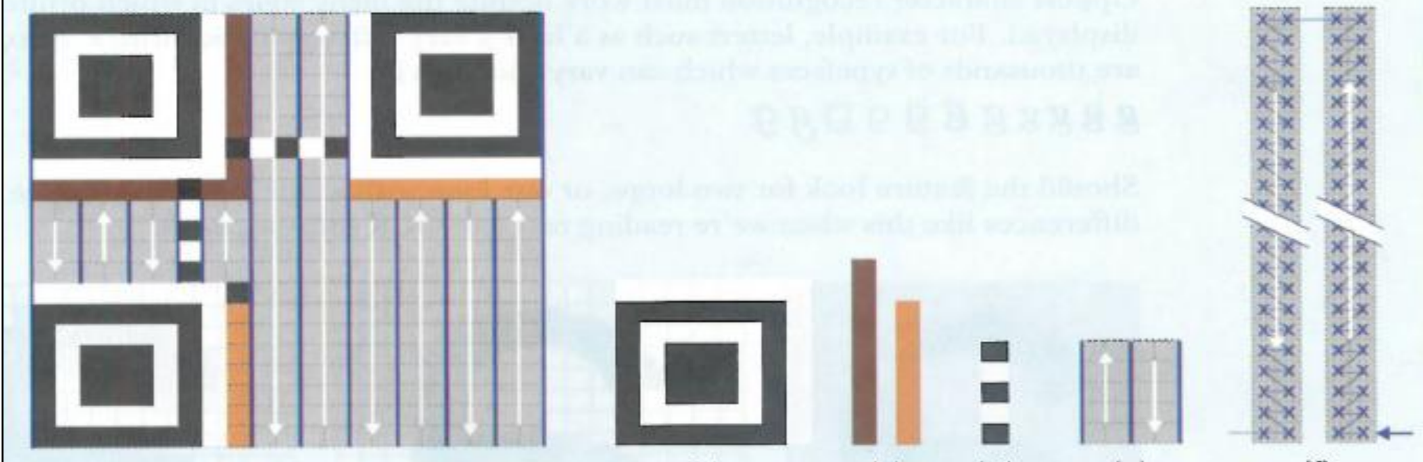


Figure 8.17 Structure of a QR Code: (a) the overall layout, (b) a position symbol with its 1-unit "quiet" region, (c) the two separate copies of format information, (d) timing symbols, (e) data area, and (f) double-bit assignment pattern, which begins in the lower right corner of the layout shown in (a) and continues.

Multimedia Challenges

- Despite of the ingenuity of audio and video compression!
  - Multimedia software applications loses frames, pixelates images, garbles words

- Latency : The time it takes for information to be delivered
  - The system must operate fast enough and precisely enough to appear natural
  - Long latencies just make us wait, but long latency can ruin the effect!
  - Causes: slow server, network congestion
  - There is an absolute limit to information delivery — the speed of light
- Bandwidth: How much information is transmitted per unit time
  - Higher bandwidth usually means lower latency

- Virtual Reality is challenged by both latency and bandwidth limitations
  - Creating a synthetic world and delivering it to our senses is a difficult technical problem

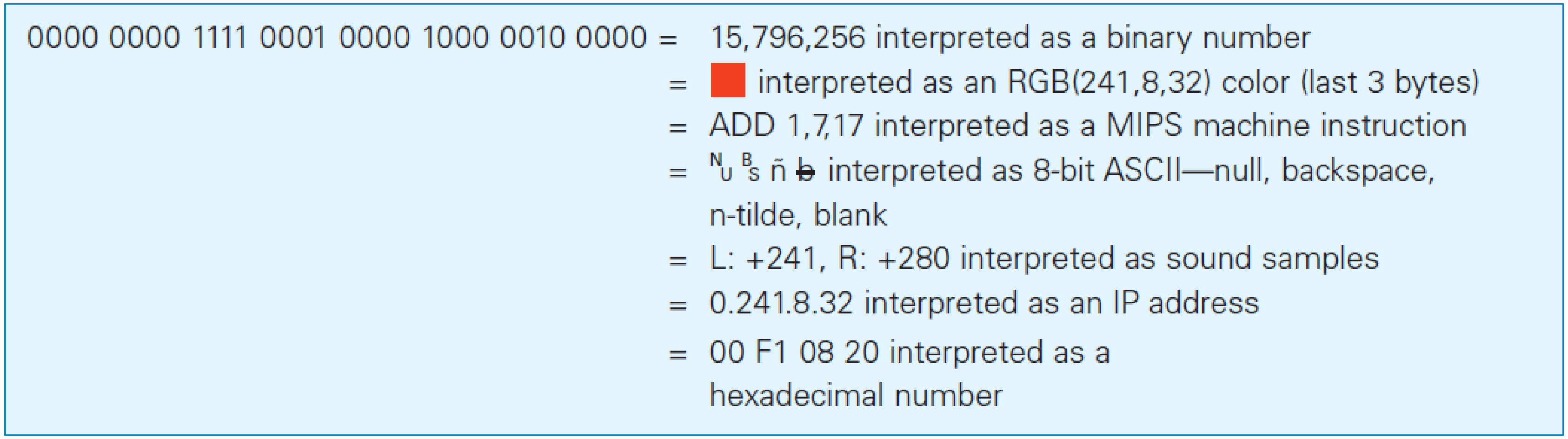
Bits are Bias-Free Universal Medium

- Bits are bias-free ➔ no inherent meaning
- Given a bit sequence: 0000 0000 1111 0001 0000 1000 0010 0000
  - There is no way to know what information it represents
- The meaning of the bits comes entirely from the interpretation placed on them by users or by the computer

- Bits can represent all discrete information
- 4 bytes (32 bits) can represent many kinds of information

Bits are Not Necessarily Binary Numbers!

- Computers represent information as bits
- Bits can be interpreted as binary numbers
- Bits do not always represent binary numbers
  - ASCII characters
  - RGB colors
  - Or an unlimited list of other things



0000 0000 1111 0001 0000 1000 0010 0000 = 15,796,256 interpreted as a binary number
= 🟥 interpreted as an RGB(241,8,32) color (last 3 bytes)
= ADD 1,7,17 interpreted as a MIPS machine instruction
= $^{N}{}_{U}$ $^{B}{}_{S}$ ñ ƀ interpreted as 8-bit ASCII—null, backspace, n-tilde, blank
= L: +241, R: +280 interpreted as sound samples
= 0.241.8.32 interpreted as an IP address
= 00 F1 08 20 interpreted as a hexadecimal number

Figure 8.18 Illustration of the principle that "bits are bits." The same 4 bytes shown can be interpreted differently.

Summary [1/2]

- Binary representation and binary arithmetic are the same as they are for decimal numbers, but they are limited to 2 digits
- The decimal equivalent of binary numbers is determined by adding their powers of 2 corresponding to 1's
- With RGB color, each intensity is a 1-byte numeric quantity represented as a binary number
- We can use arithmetic on the intensities to "compute on the representation," for example, making gray lighter and colorizing a black-and-white picture from the 19th century

Summary [2/2]

- When digitizing sound, sampling rate and measurement precision determine how accurate the digital form is; uncompressed audio requires more than 80 million bits (10 Mbyte) per minute
- Compression makes large files manageable:
  - MP3 for audio, JPEG for still pictures, and MPEG for video
  - These compact representations work because they remove unnecessary information
- Optical character recognition technology improves our world
- The Bias-Free Universal Medium Principle embodies the magic of computers through universal bit representations and unbiased encoding