Representing Information Digitally
CST111—Introduction to Information Technology

Data and Computers
- Computers are multimedia devices, dealing with a vast array of information categories
- Computers store, present, and help us modify:
  - Numbers
  - Text
  - Audio
  - Images and graphics (in color)
  - Video

Analog vs. Digital (Page 1)
- Computers are finite!
  - So how do we represent an infinite world?
- The answer is that we represent enough of the world to satisfy our computational needs and our senses of sight and sound

Analog vs. Digital (Page 2)
- Analog is continuous data/information
  - Analogous to the actual information it represents
  - An infinite number of actual values exist between the markings on the measuring device

Analog vs. Digital (Page 3)
- Digital is discrete data/information
  - Involves taking many distinct samples of data, breaking the information up into separate elements

Analog vs. Digital (Page 4)
- Computers cannot work well with analog data, so the data must be digitized (breaking the data into pieces and representing those pieces separately)
- Why do we use binary to represent digitized data?

Electronic Signals (Page 1)
- Important facts about electronic signals:
  - An analog signal continually fluctuates in voltage up and down
  - A digital signal has only a “high” state or “low” state, which corresponds to the two binary digits 1 and 0

Electronic Signals (Page 2)
- All electronic signals (both analog and digital) degrade as they move down a line
  - Its voltage fluctuates due to environmental effects
- Periodically, a digital signal is reclocked to regain its original shape

A Binary System
- Two patterns make a binary system
  - Base 2 (digits are either 0 or 1)
- Any names can be given to the two patterns as long as they are consistent, i.e. present and absent

Binary System Examples
- Present  Absent
  True    False
  1      0
  On     Off
  Yes    No
  +      -
  Black  White
Bits and Bytes
- Bit—a binary digit (1 or 0) represented by a single memory circuit
  - “On” represents 1; “off” represents 0
- Byte—made up of 8 bits
  - The smallest “meaningful” storage unit in a computer (in most instances)

Why the Term “BYTE”
- Why is “byte” spelled with a “y”?
  - The engineers at IBM were looking for a word for a quantity of memory between a bit and a word (usually 32 bits)—the word “bite” seemed appropriate ...
  - But they changed the “i” to a “y”, to minimize typing errors—so that the changing of a single letter in each would not result in accidentally creating the other

Bits in Computer Memory
- Memory is arranged inside computers in a long continuous sequence of bits from the beginning of memory to its end
- An analogy is “Sidewalk Memory”
  - Each sidewalk square represents a memory slot (bit), and stones represent the presence or absence
  - If a stone is on the square, the value is 1, if not the value is 0

Combining Bit Patterns (Page 1)
- Since only two patterns exist, it is necessary to combine them into sequences to create enough symbols to encode necessary information
- Binary has 2 patterns
  - Arranging them into $n$-length sequences creates $2^n$ symbols

Combining Bit Patterns (Page 2)
- $n$ $2^n$ Symbols
  - 1 $2^1$ 2
  - 2 $2^2$ 4
  - 3 $2^3$ 8
  - 4 $2^4$ 16
  - 5 $2^5$ 32
  - 6 $2^6$ 64
  - 7 $2^7$ 128
  - 8 $2^8$ 256
  - 9 $2^9$ 512
  - 10 $2^{10}$ 1024

An 8-bit Byte
- Can represent...
  - A value (number)
  - A character
  - An instruction (or part of an instruction) — called a word

Digitizing Text
- Binary representation—1 and 0—used for encoding keyboard and other text characters
- What symbols should be included?
  - The list should be kept small enough to use the fewest number of bits, but large enough to not leave out critical characters

Assigning Symbols
In English there are:
- 26 uppercase as well as 26 lowercase letters
- 10 numeric digits
- 10 arithmetic characters
- 20 punctuation characters (including space)
- 3 non-printable characters (new line, tab, backspace)

For a total of 95 character symbols, a minimum of 7-bit sequences are needed
- \(2^6 = 64\)
- \(2^7 = 128\)

**ASCII**
- An early standard 7-bit code which handled all English characters is ASCII (American Standard Code for Information Interchange)
- It was primarily a coding pattern used for data communications between computers
- The complete specification can be found at [http://www.asciitable.com/](http://www.asciitable.com/)

**Extended ASCII: An 8-bit Code** (Page 1)
- By the mid-1960's, it became clear that 7-bit ASCII was not enough to represent text from languages other than English
- IBM extended ASCII to 8 bits \(2^8 = 256\) symbols

**Extended ASCII: An 8-bit Code** (Page 2)
- Called Extended ASCII (or sometimes ASCII-8), the first half is original ASCII with a 0 added at the beginning of each group of bits
- Handles most Western languages and several additional useful symbols
- The complete specification can be found at [http://www.idevelopment.info/data/Programming/ascii_table/PROGRAMMING_ascii_table.shtml](http://www.idevelopment.info/data/Programming/ascii_table/PROGRAMMING_ascii_table.shtml)

**Unicode** (Page 1)
- Several languages around the world have far more than 256 individual characters
- Unicode uses 16 bits (i.e. \(2^{16} = 65536\) characters)
  - First 7 bits (128 chars) are same as the original ASCII
  - Different locales have different character sets

**Unicode** (Page 2)
- Used in most modern day computer systems
- Complete Unicode specification can be found at [http://www.ssec.wisc.edu/~tomw/java/unicode.html](http://www.ssec.wisc.edu/~tomw/java/unicode.html)

**Text Compression**
- Assigning 16 bits to each character in a document uses too much file space
- Ways needed to store and transmit text efficiently
- Text compression techniques:
  - Keyword encoding
  - Run-length encoding
  - Huffman encoding

**Keyword Encoding** (Page 1)
- All frequently used words are replaced with a single character
- Format to specify the symbol:
  - Asterisk (*) followed by the symbol followed by a blank space and the word it replaces
- Examples:
  - *^ as
  - *~ the
  - *+ and

**Keyword Encoding** (Page 2)
Given the following paragraph:

We hold these truths to be self-evident, that all men are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness. That to secure these rights, Governments are instituted among Men, deriving their just powers from the consent of the governed. That whenever any Form of Government becomes destructive of these ends, it is the Right of the People to alter or to abolish it, and to institute new Government, laying its foundation on such principles and organizing its powers in such form, as to them shall seem most likely to effect their Safety and Happiness.

**Keyword Encoding** (Page 3)

The encoded paragraph is:

We hold # truths to be self-evident, $ all men are created equal, $ ~y are endowed by ~ir Creator with certain unalienable Rights, $ among # are Life, Liberty + ~ pursuit of Happiness. — $ to secure # rights, Governments are instituted among Men, deriving ~ir just powers from ~ consent of ~ governed, — $ whenever any Form of Government becomes destructive of # ends, it is ~ Right of ~ People to alter or to abolish it, + to institute new Government, laying its foundation on such principles + organizing its powers in such form, ^ to ~m shall seem most likely to effect ~ir Safety + Happiness.

**Keyword Encoding** (Page 4)

- What was the savings?
  - Original paragraph—656 characters
  - Encoded paragraph—596 characters
  - Characters saved—60 characters
  - Compression ratio—596/656 = 91%

**Run-Length Encoding** (Page 1)

- A single character may be repeated over and over again in a long sequence
- Replace a repeated sequence (nnnnnnnn) with:
  - A flag character
  - The repeated character
  - Number of repetitions
- Represented as *n8
  - * is the flag character
  - n is the repeated character
  - 8 is the number of times n is repeated

**Run-Length Encoding** (Page 2)

- Example 1:
  - Original text
    - bbbbbbbjjjklqqqqqq+++++
  - Encoded text
    - *b8jjkl*q6*+5 (Why is "l" not encoded? Or the "J"?)
  - The compression ratio is 15/25 or .6

**Run-Length Encoding** (Page 3)

- Example 2:
  - Original text
  - *x4*p4l*k7
  - Encoded text
    - xxxxppppllkkkkkk

- This type of repetition does not occur in English text; can you think of a situation where it might occur?
Huffman Encoding

- Why should the character “X” and “z” take up the same number of bits as “e” or “ ”?
- Huffman codes use variable-length bit strings to represent each character
- More frequently used letters have shorter strings to represent them

Huffman Encoding

Example 1:
- Original text: ballboard
- Encoded text: 1010001001001010110001111011
- The compression ratio is 28/73 or 39%

Example 2:
- Encode “doll”

Huffman Encoding

- In Huffman encoding no character’s bit string is the prefix of any other character’s bit string
- To decode:
  - Look for match left to right, bit by bit
  - Record letter immediately when a match is found
  - Continue at the next digit where you left off

Huffman Encoding

Example 1:
- Encoded text: 1011111001010
- Decoded text: ???
- The compression ratio is 13/32 or 41%

Example 2:
- Decode 1110100100

Huffman Encoding

- Technique for determining codes guarantees the prefix property of the codes
- Two types of codes:
  - General—based on use of letters in English (Spanish, ...)
  - Specialized—based on text itself or specific types of text

Representing Audio Information

- Sound is perceived when a series of air compressions vibrate a membrane in our ear which sends signals to our brain

Representing Audio Information

- Microphone converts sound pressure to voltage
  - The voltage in the signal varies in direct proportion to the sound wave
- The voltage signal is digitized by sampling:
  - Periodically measuring the voltage and recording its numeric value

Representing Audio Information

- How often should we sample?
  - A sampling rate of 44,100 times per second or greater is enough to create good sound reproduction
  - Some data is lost, but a reasonable sound is reproduced

CD Sound Reproduction

- CDs store audio information digitally
- On the surface of the CD are microscopic pits that represent binary digits
- A low intensity laser is pointed as the disc and the laser light reflects strongly if the surface is smooth and poorly if the surface is pitted

### Audio Formats
- Audio formats include:
  - WAV, AU, AIFF, VQF, and MP3
  - MP3 (MPEG-2, audio layer 3 file) is dominant
    - Analyzes frequency spread and discards information that cannot be heard by humans
    - Bit stream is compressed using a form of Huffman encoding to achieve additional compression

### Music Technology 101: Sampling Rate and Bit Depth Explained
- [http://www.youtube.com/watch?v=zC5KFnSUPNo](http://www.youtube.com/watch?v=zC5KFnSUPNo)

### Images and Graphics (Page 1)
- Color—the perception of the frequencies of light that reach the retinas of our eyes
- Retinas have three types of color photoreceptor cone cells that correspond to the colors of red, green, and blue

### Images and Graphics (Page 2)
- Color is expressed as an RGB (red-green-blue) value—three numbers that indicate the relative contribution of each of these three primary colors
- An RGB value of (255, 255, 0) maximizes the contribution of red and green, and minimizes the contribution of blue, resulting in a bright yellow

### Images and Graphics (Page 3)
- Color depth—the amount of data that is used to represent a color
- HiColor—a 16-bit color depth (five bits used for each number in an RGB value with the extra bit sometimes used to represent transparency)
- TrueColor—a 24-bit color depth: eight bits used for each number in an RGB value

### Indexed Color
- Web browsers (i.e. Internet Explorer) may support only a certain number of specific “safe” colors, creating a palette from which to choose

### Digitized Images and Graphics (Page 1)
- Digitizing a picture—representing it as a collection of individual dots called pixels
- Resolution—number of pixels used to represent a picture
- Raster Graphics—storage of data on a pixel-by-pixel basis
  - Bitmap (BMP), GIF, JPEG, and PNG are raster-graphics formats

### Digitized Images and Graphics (Page 2)
- Bitmap format—contains pixel color values of the image from left to right and from top to bottom
- GIF format (indexed color)—each image is made up of only 256 colors
- JPEG format—averages color hues over short distances
- PNG format—like GIF but achieves greater compression with wider range of color depths

### Vector Graphics (Page 1)
- Vector graphics is a format that describes an image in terms of lines and geometric shapes
- A vector graphic is a series of commands that describe a line’s direction, thickness, and color
- The file sizes tend to be smaller because not every pixel is described

### Vector Graphics (Page 2)
- The good side and the bad side...
  - Vector graphics can be resized mathematically and changes can be calculated dynamically as needed
• Vector graphics are *not* good for representing real-world images

**Representing Video** *(Page 1)*

- Video codec COmpressor/DECompressor (codec) methods are used to shrink the size of video to allow it to be played on a computer or over a network.
- Almost all video codecs use lossy compressions to minimize the huge amounts of data associated with video.

**Representing Video** *(Page 2)*

- Temporal compression (intraspace):
  - A technique based on the differences between consecutive frames.
  - If most of an image in two frames has not changed, why should space be wasted to duplicate all of the similar information?
- Spatial compression (interspace):
  - A technique based on removing redundant information within a frame.
  - This problem is essentially the same as that faced when compressing still images.

**Digital Video Technical Essentials**
- [http://vimeo.com/8752524](http://vimeo.com/8752524)

**Understanding Video Compression Part 1**
- [http://vimeo.com/8752360](http://vimeo.com/8752360)

**Understanding Video Compression Part 2**
- [http://vimeo.com/8752320](http://vimeo.com/8752320)