CS 6351 DATA COMPRESSION

THIS LECTURE:
INTRODUCTION AND BACKGROUND

Instructor: Abdou Youssef

OBJECTIVES OF THIS LECTURE

By the end of this lecture, you will be able to:

- Describe what data compression is, including lossless and lossy compression
- Explain the motivation, goals, constraints and general strategies of data compression
- Define fundamental concepts related to compression, including measures for performance and for quality/fidelity
- State the foundational definitions of images, videos, audio signals, and related concepts relevant to compression
- Relate compression to data redundancy in its various forms, including spatial, temporal and human audio-visual redundancy
- Describe basic aspects of information theory, including the connection between information and uncertainty, measure of information content, and entropy, using probability

OUTLINE

- Definition, goal, and constraints of data compression
- Motivation for data compression
- Basic definitions in compression/coding
- Performance and quality metrics in compression
- Definitions related to images, video, and audio
- Overview of the general strategies for data compression (all based on redundancy reduction)
- Modeling and limitations of the human visual system
- Overview of information theory, with highlights relevant to compression

COMPRESSION
-- DEFINITION, GOAL, AND CONSTRAINTS IN BRIEF --

- **Definition**: What is Compression?
  - Compression is a process of deriving more compact (i.e., smaller) representations of data

- **Goal of Compression**
  - Significant reduction in the data size to reduce the storage/bandwidth requirements
- **Constraints on Compression**
  - Perfect or near-perfect reconstruction of original data

LOSSLESS AND LOSSY COMPRESSION

- **Two broad types of data compression**
  - **Lossless Compression**:
    - It is the kind of compression where the original data can be fully and 100% accurately reconstructed (recovered) from the compressed data
    - Needed for certain applications where full recovery of data is essential
      - Text compression
      - Database records compression (e.g., bank records, student records, tax records)
  - **Lossy Compression**:
    - It is the kind of compression where the original data can never be fully recovered from the compressed data with 100% fidelity
    - Needed in applications where the original data size is too large and needs to be compressed to much smaller sizes than lossless compression can achieve
      - Image, video and audio compression

GENERAL STRATEGIES FOR COMPRESSION

- We follow two broad strategies for data compression
  - Reducing the redundancy present in the data itself
  - Exploiting the characteristics (and limitations) of the human senses/faculties
    - Human vision
    - Human hearing
    - Actually all the 5 senses of humans have limitations, but digital data pertains only to the audio-visual senses (so far)

NEED/MOTIVATION FOR COMPRESSION

- Massive Amounts of Data are Involved in the Storage and Transmission of Text, Sound, Images, and Videos
- Applications
  - Medical imaging
  - Teleradiology
  - Space/Satellite imaging
  - Multimedia
  - Digital Video: entertainment, home use
  - Digital photography
  - Etc.

NEED/MOTIVATION FOR COMPRESSION
-- CONCRETE FIGURES: HOSPITALS AND NASA--

- A typical hospital used to generate **terabytes** of data per year (mostly images)
  - ~ 20 terabytes/year in 2011

- 1 megabyte(MB) = 2^{20} bytes = 1Million bytes
- 1 gigabyte (GB) = 2^{30} bytes = 1 Billion bytes
- 1 terabyte (BB) = 2^{40} bytes = 1 trillion bytes
- 1 petabyte (PB) = 2^{50} bytes = 1 quadrillion bytes
- 1 exabyte (EB) = 2^{60} bytes = 1 quintillion bytes

- Healthcare organizations have seen an explosive health data growth rate of 878% from 2016 to 2018 (see here)
  - Was about 1 petabyte/year in 2016
  - reached 8.41 petabytes/year on average in 2018
  - 2,314 exabytes in global healthcare in 2020 (Statista)
- NASA gathers gigabytes of data/second, rate growing exponentially

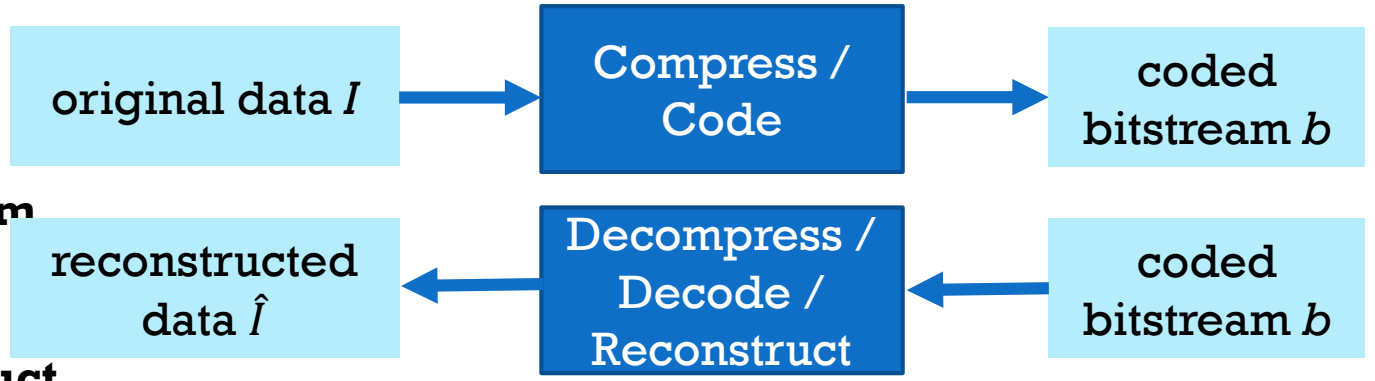
NEED/MOTIVATION FOR COMPRESSION
-- CONCRETE FIGURES: MOVIES --

- A 2-hour HD video = frame size × number of frames
  - A video is a sequence of images, called *frames,* played at a rate of 30 frames per second (fps)
  - Frame size (in HD)= 1920 width × 1080 height pixels =2,073,600 pixels
  - Number of frames= 30 fps × 60 seconds × 60 minutes × 2 hours=216,000 frames
  - Video size=2,073,600 × 216,000 = 448 Giga pixels, at 3 bytes/pixel, we get:
  - Video size=1.343 terabytes (TB) ≈ **15 Blu-ray DVD's**
- HD-video data rate per second = (1920×1080) ×30 fps × 24 bits/pixel= 1.5 Gb/s
  - With MPEG2 High (80Mb/s), need compression ratio of 18.66:1
  - With DVD MPEG2 (9.8Mb/s), need compression ratio of 152:1 (unless it is stored at 480 pixels in height)
- For Ultra High definition (4K/5K/6K/8K UHD), the amounts of data are much larger.
  - For example, in Super Hi-Vision specifications (Japan), the uncompressed video bit rate is 144Gb/s (100 times bigger than HD)

BASIC DEFINITIONS IN COMPRESSION/CODING

- **To compress:** to carry out the process of compression
- **To decompress:** to reconstruct the original data from the compressed representation
- Synonyms of "compress":
  - **code ≡ compress ≡ encode**
  - **(en)coding ≡ compression**
  - **(en)coder ≡ compression algorithm**
- Synonyms of "decompress":
  - **Decode ≡ decompress ≡ reconstruct**
  - **Decoding ≡ Decompression ≡ Reconstruction**
  - **Decoder ≡ decompression algorithm**

Note: "code" means different things in different contexts



Lossless compression: $\hat{I} = I$
Lossy compression: $\hat{I} \approx I$ but $\hat{I} \neq I$

- **Coded Bitstream**: The binary string representing the whole coded (compressed) data
- **Codeword**: A binary string representing either the whole coded bitstream or one coded data symbol (you will tell the difference from the context)

COMPRESSION PERFORMANCE METRICS
-- SIZE REDUCTION MEASURES --

- **Bit Rate** (or **bitrate or BR**): Average number of bits per original data element, after compression
  - In text compression: BR= avg. number of bits per character $= \frac{\#bits\ in\ the\ coded\ bitstream}{\#characters\ in\ the\ original\ text}$
  - In images/videos: BR= average number of bits per pixel $= \frac{\#bits\ in\ the\ coded\ bitstream}{\#pixels\ in\ the\ image\ or\ video}$
  - In audio: BR= avg. num. of bits per sound sample$= \frac{\#bits\ in\ the\ coded\ bitstream}{\#samples\ in\ the\ audio\ file}$
- **Compression Ratio** (CR) $= \frac{size\ of\ the\ uncompressed\ data\ (in\ bits)}{size\ of\ the\ coded\ bitstream\ (in\ bits)} = \frac{|I|}{|b|}$
- **Exercise**: A file has 1000 characters, and takes 8K bits before compression. After compression, its size became 4K bits. What is the bitrate? What the CR?

BASIC DEFINITIONS IN COMPRESSION/CODING
-- QUALITY MEASURES: SNR --

- Let I be an original signal (e.g., an image), and $\hat{I}$ be its lossily reconstructed counterpart
- **Signal-to-Noise Ratio** (**SNR**) in the case of lossy compression:
  - It is a measure of the quality of the reconstructed (decompressed/decoded) data
  - More accurately, it is the **fidelity** of the decoded data w.r.t. the original
  - Mathematically: $\textbf{SNR} = 10\log_{10}\left(\frac{\|I\|^2}{\|I-\hat{I}\|^2}\right) = 20\log_{10}\left(\frac{\|I\|}{\|I-\hat{I}\|}\right)$

    where for any vector/matrix/set of number $E = \{x_1, x_2, \ldots, x_N\}$,

    $$\|E\|^2 = x_1^2 + x_2^2 + \cdots + x_N^2$$

- The unit of SNR is "***decibel***" (or **dB** for short)
- So, if SNR = 23, we say the SNR is 23 dB

Question: Does it make sense to compute SNR for lossless compression? Why or why not?

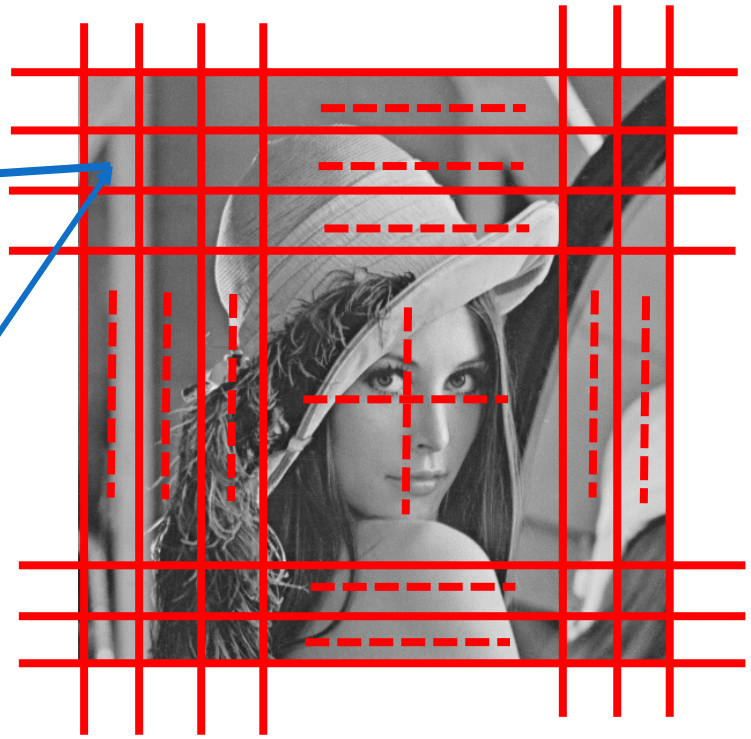
BASIC DEFINITIONS IN COMPRESSION/CODING
-- QUALITY MEASURES: MEAN-SQUARE ERROR --

- Mean-Square Error (MSE): $\text{MSE} = \frac{1}{N}\left\|I - \hat{I}\right\|^2$
- Relative Mean-Square Error (RMSE): $\text{RMSE} = \frac{\|I-\hat{I}\|^2}{\|I\|^2}$
- **Therefore, SNR** $= -10\log_{10}\text{RMSE}$
- **Observations:**
  - The smaller the RMSE (or the MSE), the higher the SNR
  - Therefore, the higher the SNR, the better the quality of the reconstructed data
- **Exercise**: Prove that if RMSE is decreased by a **factor** of 10, then SNR increases by 10 decibels.
- That justifies the multiplicative factor in the definition of the SNR.

BASIC IMAGE/VIDEO/SOUND DEFINITIONS (1/8)
-- IMAGES AND PIXELS --

- An **image** is a matrix of numbers, each number called a ***pixel*** (short for picture element)

$$\begin{bmatrix} 12 & 3 & 37 & 189 & 10 & \dots & 87 \\ 10 & 4 & 40 & 212 & 17 & \dots & 89 \\ & & & \vdots & & & \\ 15 & 7 & 59 & 241 & 25 & \dots & 94 \end{bmatrix}$$



Compute the average light intensity in that little rectangle

BASIC IMAGE/VIDEO/SOUND DEFINITIONS (2/8)
-- BINARY, GRAYSCALE, AND COLOR IMAGES --

- A **binary image** (or black-and-white B/W image) is an image where every pixel can have one of two values only (typically 0 and 1).
- A **grayscale image** is a non-color image, where the pixels are various shades of gray.
  - What we are calling here grayscale image is what we informally call black-and-white
  - In this course, B/W images are binary image, and grayscale image are grayscale images
  - Every pixel value is usually between 0 and 255 (so 8 bits long), but other range are used
- A **color image** is an image where every pixel can have a color.
- **Theorem** (Newton/Young): Every color is a combination of three colors (such as Red, Green and Blue), called the basic or primary colors.
- Therefore, in color images, every pixel is represented with three (numerical) components, such as (R,G,B), where R represents how much red there is in that pixel, G how much green, and B how much blue. Typically, each component is 1 byte long

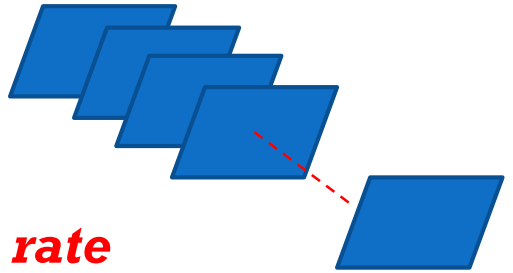
BASIC IMAGE/VIDEO/SOUND DEFINITIONS (3/8)
-- SPATIAL RESOLUTION --

- **Spatial resolution** of an image is:
  - The total number of pixels in the image (like you hear a megapixel image, or a 6-megapixel camera, ...); or
  - The number of pixels per row and per column
    - like when one says this image is an 1080x1920 image,
    - which means it has 1080 rows and 1920 columns,
    - which also means that every row has 1920 pixels and every column has 1080 pixels,
    - that is, the image has height 1080 pixels and width 1920 pixels; or
  - The number of pixels per inch; for binary images
    - like in fax machines, basic scanners, and old dot-matrix printers), it is called "dot per inch (dpi)".
- Note: For a fixed physical size image, the higher the spatial resolution, the more (and smaller) the pixels are, and thus the better the quality (and detail) of the image.

BASIC IMAGE/VIDEO/SOUND DEFINITIONS (4/8)
-- DENSITY RESOLUTION --

- The ***density resolution*** (or ***bit depth***) is the number of bits per pixel
- The higher the density resolution, the more colors (or shades of gray) can be represented, and thus the crisper or more detailed the image is
- But of course, the higher the density resolution, the more bits are needed to represent the uncompressed image
- **Question**: Does higher density resolution produce more redundancy, and thus more opportunity for deeper compression (i.e., higher compression ratios)?
  - Think about it
  - We'll see the question later in the semester

BASIC IMAGE/VIDEO/SOUND DEFINITIONS (5/8)
-- VIDEOS --



- A ***video*** is a sequence of images
- Every image in the sequence is called a ***frame***
- A video is captured/displayed at a certain rate, called the ***frame rate***
- A typical rate that does not show jerkiness is 30 frames per second (fps), in digital video and most games
- For higher definition (of motion), the rate can be higher, like 60 fps, 120 fps, 240 fps
  - More frames, better slow motion, but more storage
  - Question: Does higher fps produce more redundancy, and thus more opportunity for deeper compression (i.e., higher compression ratios)?
- Lower rates, likes 20 fps and even 15 fps, were used when communications and computers were slower.
- Lower than 15 fps rates would be quite jerky and unacceptable.

BASIC IMAGE/VIDEO/SOUND DEFINITIONS (6/8)
-- VIDEO RESOLUTION --

- Resolutions of various technologies
  - Standard definition (SD): Height = 576 or 480 pixels, Width = 768 or 640 pixels
  - High definition (HD): H = 1080 p, W = 1920 p
  - 4K ultra high definition (4K UHD): H = 2160 p, W =3840 p
  - 5K ultra high definition (5K UHD): H = 2880 p, W = 5120 p
  - 6K ultra high definition (6K UHD): H = 3456 p, W =6144 p
  - 8K ultra high definition (4K UHD): H = 4320 p, W = 7680 p
  - Note: UHD allows for frame rates of up to 120 fps, and the density resolution is 8 or 10 or 12 bits per color (i.e., 24 or 30 or 36 bits per pixel)
  - High-end gaming and 3D can allow for up to 240 fps, but that requires high end computers/GPUs and memory
- Note: You are not required to memorize those numbers

BASIC IMAGE/VIDEO/SOUND DEFINITIONS (7/8)
-- SOUND/AUDIO --

- A **sound/audio (digital) signal** is a sequence of values, called ***samples***, where every sample is the intensity of the recorded sound at the corresponding moment in time
- The ***sampling rate (SR)*** of a sound is the number of samples per second
- The CD quality sampling rate (SR) is 44.1K samples per second (or 44.1 KHz), usually at 16 bits per sample, though 24 bits per sample is now common
- In digital sound used for miniDV, digital TV, and DVD, the SR is 48 KHz
- In DVD-Audio and in Blu-ray audio tracks, the SR is 96 KHz or 192 KHz
- In the UHD Super Hi-Vision specifications:

SR: 48/96 kHz Bit length: 16/20/24 bit Number of channels: 24

BASIC IMAGE/VIDEO/SOUND DEFINITIONS (8/8)
-- SOUND/AUDIO: A/D AND D/A CONVERTERS --

- When the sampling rate is infinity, the signal becomes what is called ***analog signal***
- When a signal is captured as an analog signal, it can be converted to a digital signal by an ***analog-to-digital converter*** (also called A/D converter or digitizer)
- If a digital signal is fed into a ***digital-to-analog converter*** (also called D/A converter), the output is obviously an analog signal.
- ***Modulators*** are D/A converters
- ***Demodulators*** are A/D converters.
- So, a ***modem*** is both an A/D and D/A converter.

STRATEGIES FOR COMPRESSION
-- REDUNDANCY REDUCTION --

- We will exploit the following kinds of redundancy
  - Symbol-Level Representation Redundancy
  - Block-Level Representation Redundancy
  - Inter-Pixel Spatial Redundancy
  - Inter-Pixel Temporal Redundancy (in Video)
  - Audio-visual Redundancy: limitations of the human senses

REDUNDANCY REDUCTION
-- SYMBOL-LEVEL REPRESENTATION REDUNDANCY --

- Different symbols occur with different frequencies

A sample text:
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.

Letter	Count	Letter	Count	Letter	Count	Letter	Count	Letter	Count
a	16	g	2	m	2	s	8	y	3
b	4	h	13	n	9	t	22	z	0
c	3	i	10	o	6	u	5		
d	6	j	0	p	3	v	1		
e	28	k	0	q	1	w	3		
f	3	l	9	r	12	x	0		

- ***Fixed-length*** codes vs. ***variable-length*** codes
  - Fixed-length codes: all symbols have codewords of the same length, like in ascii
  - Variable-length codes: the codewords of different symbols can have different lengths
- Frequent symbols are better coded with short codewords
- Infrequent symbols are coded with long codewords

No compression can be achieved

Compression can be achieved

REDUNDANCY REDUCTION

-- BLOCK-LEVEL REPRESENTATION REDUNDANCY --

- Different blocks of data occur with varying frequencies

Example I: 1001001101001101101001101000001000100001101110 11

48 bits

If you code the individual symbols 0 and 1, then you can't do better than to code 1 with 1 and 0 with 0, thus achieving no compression, i.e., CR=1 and br=1

But, if you view the input as a sequence of 3-bit blocks, some blocks like (100 and 110) occur more frequently than other blocks:

I: 100 100 110 100 110 110 100 110 100 000 100 010 000 110 111 011

- Better then to code blocks than individual symbols

Coded bitstream: 1 1 01 1 01 01 1 01 1 001 1 000110 001 01 0000 00010

37 bits, thus, CR=48/37, and br=$\frac{37}{48} = 0.97 < 1$

x:y means y=codeword(x)

100: 1	011: 00010
110: 01	010: 000110
000: 001	101: 0001110
111: 0000	001: 0001111

- The block size can be fixed or variable
- The block-code size can be fixed or variable
- Frequent blocks are better coded with short codes
- Example techniques: Block-oriented Huffman, Run-Length Encoding (RLE), Arithmetic Coding, Lempil-Ziv (LZ), etc., which will be covered in the next couple of lectures

REDUNDANCY REDUCTION

-- INTER-PIXEL SPATIAL REDUNDANCY --

- Neighboring pixels tend to have similar values
- Neighboring pixels tend to exhibit high correlations
- Techniques: Decorrelation and/or processing in the frequency domain
- Spatial decorrelation converts correlations into symbol-redundancy or block-redundancy
- Frequency domain processing addresses visual redundancy (see below)
- We will exploit this kind of redundancy in lossy compression later in the semester

REDUNDANCY REDUCTION

INTER-PIXEL TEMPORAL REDUNDANCY (IN VIDEO & AUDIO)

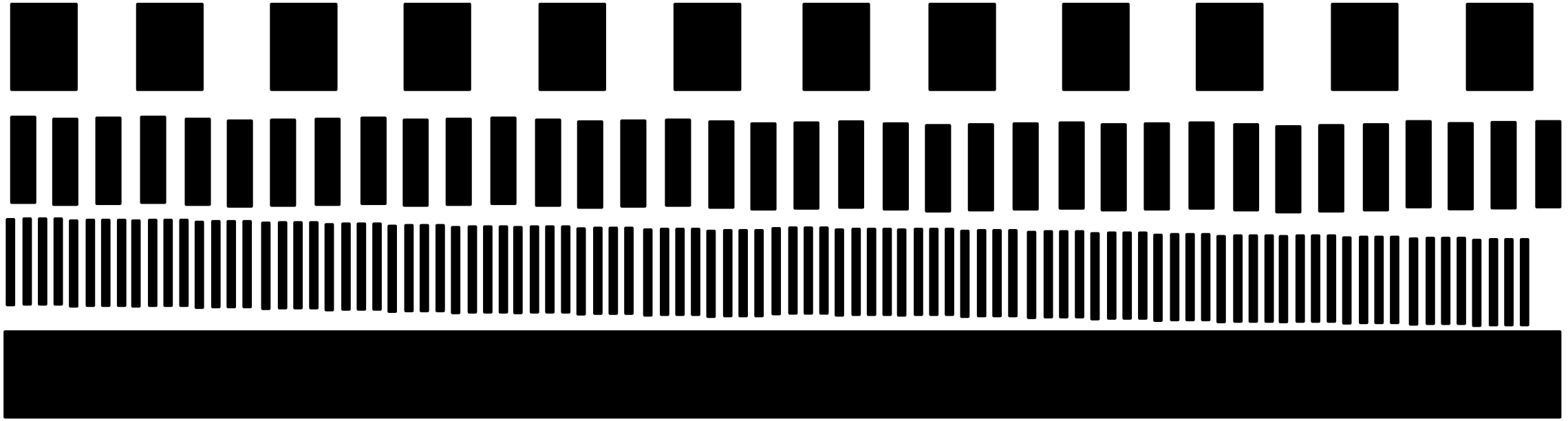
- Often, the majority of corresponding pixels in successive video-frames are identical over long spans of frames
  - Pixels of fixed backgrounds (e.g., walls) don't change from frame to frame
  - Pixels of moving objects move a little (or not at all) from frame to frame because the speed of movement of objects can be slower than the camera speed (i.e., than the frame rate)
  - Therefore, the higher the frame rate (fps), the more temporal redundancy
- Due to motion, blocks of pixels change in position but not in values between successive frames
- Thus, block-oriented motion-compensated redundancy reduction techniques are used for video compression (covered in later lectures)
- In audio, successive samples are similar/correlated. Why?

AUDIO-VISUAL REDUNDANCY
-- LIMITATION OF THE HUMAN SENSES --

- The human visual system (HVS) has certain limitations that make many image contents invisible
- Those contents, termed ***visually redundant***, are the target of removal in lossy compression
- That is, if you clutter too many details within a small spatial region, we stop being able to see contrasts and details
- This is illustrated and elaborated in the next several slides

LIMITATION OF THE HVS (1/7)

- Imagine a sequence of alternating black and white stripes of the same width
- Frequency: number of stripes per inch
- Increase the frequency and see what happens:

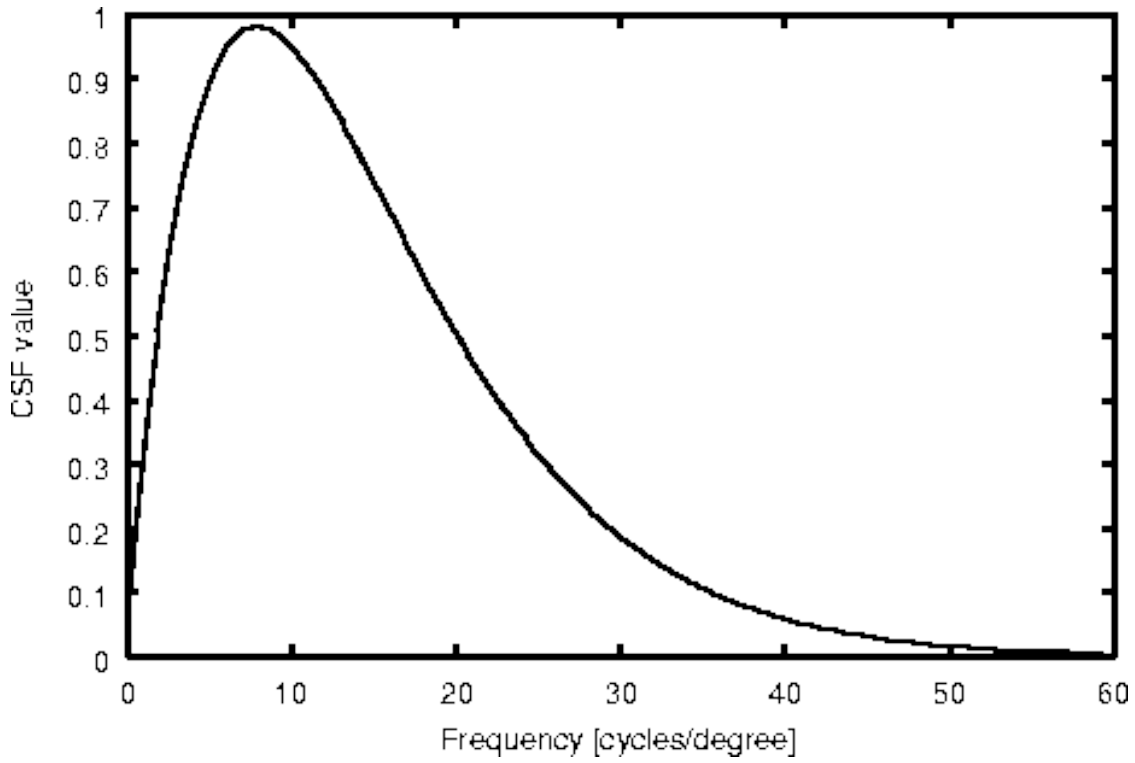


- As the frequency increases, we have more stripes/inch, so the stripes get thinner & thinner, which looks like the black stripes get closer to each other until we can't tell them apart

LIMITATION OF THE HVS (2/7)

- The graph below plots the contrast sensitivity function (CSF) as a function of frequency
  - Frequency = #stripes/degree
  - It can be converted into:

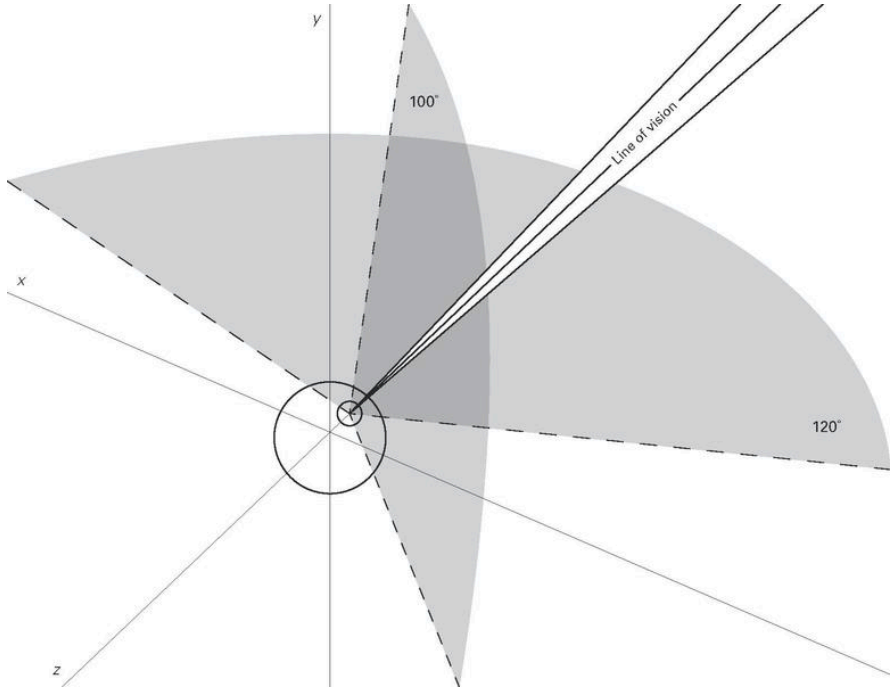
    Frequency = #stripes/inch



- It measures how well we can tell black stripes apart (assuming a white background), depending on the frequency of stripes, i.e., on how close the black stripes are to each other

LIMITATION OF THE HVS (3/7)
(SLIDES 3/7, 4/7 and 5/7 ARE OPTIONAL)

- This phenomenon has ben studied, formalized, and quantified
- It was found that the HVS can see only within a small range of spatial frequencies: 1-60 cycles/arc-degree
  - Vertical field of vision: around 150 degrees (but only 100 degrees of good vision)
  - Horizontal field of vision: around 210 degrees (but only 120 degrees of good vision)



LIMITATION OF THE HVS (4/7)

- To understand the frequencies in the plot better , we'll translate the frequencies from number of cycles per degree to number of stripes per inch.
- A full circle is 360 degrees
- At a viewing distance of R inches (i.e., a circle of radius R), the length (in inches) of one degree is:

Perimeter of a circle of radius R

$$\text{Length of } \mathbf{one\ degree} = \frac{2\pi R}{360} = \frac{2 \times 3.14 \times R}{360} = 0.0175\ R \text{ inches}$$

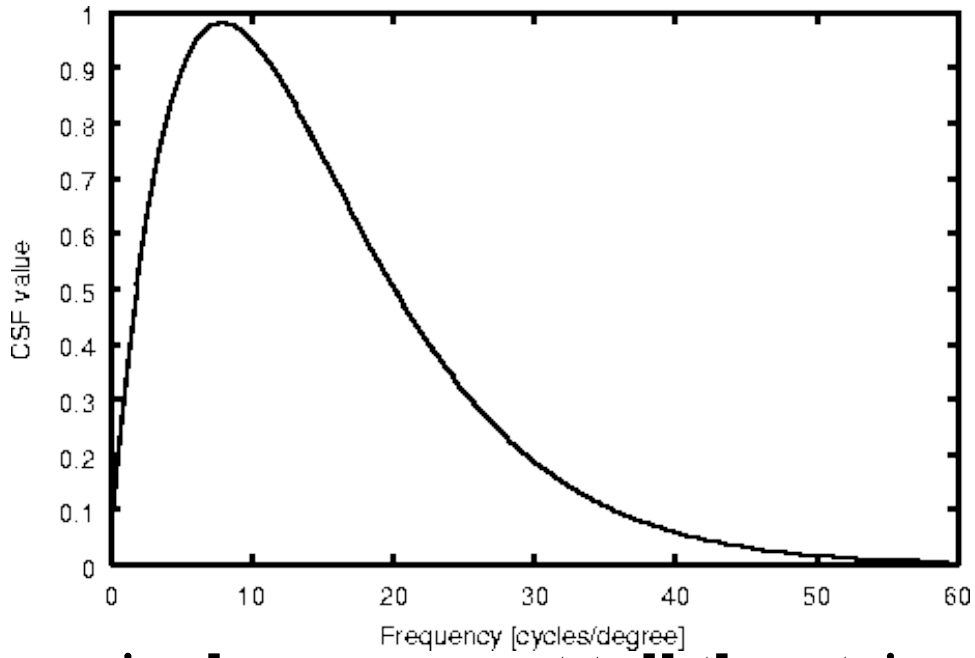
- 1 cycle/degree =1 stripe/degree= $\frac{360}{2\pi R}$ stripes per inch = $\frac{57.29}{R}$
- For example, at viewing distance R = 10 feet = 120 inches:
  - 1 cycle/degree = $\frac{57.29}{R} = \frac{57.29}{120} = 0.4775$ stripes per inch.

LIMITATION OF THE HVS (5/7)

- Therefore, at viewing distance of R=10 ft.=120 inches (i.e., almost 3 meters), here are the translations of a few frequencies:
  - 1 cycle per degree = 0.48 stripes per inch
  - 8 cycles per degree = 3.8 stripes per inch
  - 10 cycles per degree = 4.76 stripes per inch
  - 20 cycles per degree = 9.52 stripes per inch
  - 30 cycles per degree = 14.28 stripes per inch
  - 40 cycles per degree = 19.05 stripes per inch
  - 50 cycles per degree = 23.81 stripes per inch
  - 60 cycles per degree = 28.57 stripes per inch

LIMITATION OF THE HVS (6/7)

- The CSF graph shows that, on average, a person is most sensitive to contrast at 8 cycles/degree, i.e., when we have 3.8 stripes per inch at a viewing distance of 10 feet.



- It also shows that beyond 28 stripes per inch, we cannot tell the stripes apart (the stripes will appear fused into a continuous blob of gray/black), from a viewing distance of 10 feet.

LIMITATION OF THE HVS (7/7)

- Since we can't see "busy" visual data, we can eliminate it (or reduce) without visible effect, thus leading to compression
- That kind of compression is lossy, since we are actually throwing out information
- But since the removed information is not visible, the quality (or fidelity) of the reconstructed information will be quite high (humans won't notice/see the difference)
- How to locate and identify busy visual data, and how to reduce/remove it, is a very interesting subject for later lectures in this course

LIMITATIONS OF HUMAN HEARING

- The human audio system (ears) has similar characteristics and limitations
- Specifically, we can hear in a limited range of audio frequencies (up to 20KHz)
- So, any audio data coming from higher frequencies (i.e., higher pitches), is not audible to the human ear, and thus can be removed without effect (i.e., without being noticed by the human ear), leading to lossy but very good compression
- How to locate and identify busy (high-pitch) audio data, and how to reduce/remove it, is a very interesting subject for later lectures in this course

INFORMATION THEORY
-- BACKGROUND --

- Information Theory was one of the greatest scientific achievements of the 2oth century
- It deals with
  - Measuring information (in bits)
  - Coding information efficiently (e.g., lossless compression)
  - Error-correcting coding to tolerate error and noise in communication channels
- It was initially developed by Claude Shannon (the father of Information Theory)
- Our primary use of it is the notion of *entropy*

INFORMATION THEORY
-- DEFINITIONS --

- ***Discrete Memoryless Source S***: A data generator where the alphabet is finite and the symbols generated are independent of one another.
- Assume the alphabet is $\{a_1, a_2, \ldots, a_n\}$, and assume that we know the probability of each occurrence of each alphabet symbol
  - Let $p_i = \Pr[a_i]$
  - That is, in a sequence of symbols generated by that source, the character a_i occurs with probability p_i (for example, if p_i=0.15, then 15% of the symbols generated are a_i)
- ***Source with Memory***: A data generator where there is inter-symbol correlation
  - Each text file written by humans is a source with memory because it is in the nature of words that certain letters are more likely to occur after certain other letters
  - For example, if the current letter is 't', the next letter is more likely to be 'h' than 'x'
  - Similarly, after 'q' we expect letter 'u' with a very high probability
  - The probability distribution of the letters can in this case be quite complicated
- Sources of information: text files, email messages, radio/TV broadcasts, etc.

INFORMATION THEORY
-- MEASURE OF INFORMATION --

- Given a sequence of symbols (like a piece of text), the main question in this context is: **How much information is there in that sequence/piece of text?**
- But what is information? It is an interesting question, isn't it?
- For example, if you are now on top of Mount Everest and I tell you that "tomorrow will be very cold", does that have a lot of information?
  - No, since it is quite expected to be cold there, especially in Fall/Winter time
- But if I say to you now that "Tomorrow, there will be a solar eclipse"
  - That carries a lot of information because solar eclipses are rare events
- So, **information** can be viewed as ***uncertainty***: the more uncertain something is, the more information there is in a sentence about its occurrence

INFORMATION THEORY
-- MEASURE OF INFORMATION/UNCERTATINTY --

- Since information is ***uncertainty***, and uncertainty is quantified with probability, we should be able to measure information using probabilities
- One of Shannon's great contributions:
  - If a fact f occurs with a probability p, the amount of information that f carries is

$$\mathbf{-\log_2 p \text{ bits}}$$

  - We will not prove that, but instead we'll take it for granted and use it
- **Illustration and verification**: If our alphabet is $\{0, 1\}$, and 0 occurs with probability ½, and so does 1, how much information does "0" carry?
  - Answer: $-\log\left(\frac{1}{2}\right) = 1$ bit!
  - Same thing with "1": it carries 1 bit of information
- So, Shannon's measure of information is reasonable!

INFORMATION THEORY
-- ENTROPY: LOWER BOUND ON LOSSLESS BITRATE--

- Shannon was very interested to know the **minimum bitrate possible** for a given source of information, no matter which lossless coder is used. He found it, and called it entropy
- ***Entropy*** $\stackrel{\text{def}}{=}$ the minimum average number of bits/symbol possible
- Recall that bitrate (BR) of a coder is the average number of bits/symbol
- Therefore, **entropy = the minimum possible bitrate**
- That is, (the bitrate of any possible lossless coder) $\geq$ (the entropy of the data source)
  - No lossless compression scheme can beat the entropy
  - The best a lossless-coder designer can hope for is to achieve the entropy

INFORMATION THEORY
-- ENTROPY OF MEMORYLESS SOURCE--

- Recall that in a Discrete Memoryless Source S, the symbols are independent of one another, the alphabet is $\{a_1, a_2, \ldots, a_n\}$, and assume that the probabilities $p_i = \Pr[a_i]$ for all i are given/known/computable
- Entropy of such a source is the minimum average bits per symbol
- The bits needed for symbol a_i are at least the amount of information in a_i, that is, $-\log_2 p_i$ bits
- Thus, the minimum average number of bits, which must be weighted by the probabilities, is: $p_1(-\log p_1) + p_2(-\log p_2) + \cdots + p_n(-\log p_n)$
- Therefore, Entropy of S, denoted $H(S)$, is:

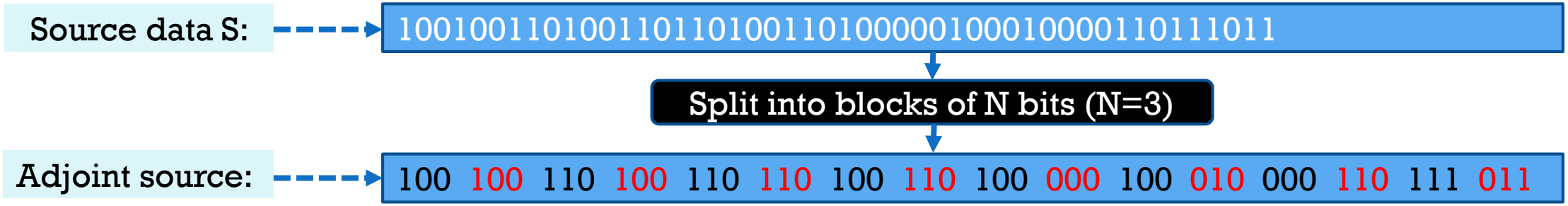
$$\boldsymbol{H(S) = -(p_1 \log p_1 + p_2 \log p_2 + \cdots + p_n \log p_n) = -\sum_{i=1}^{n} p_i \log p_i}$$

INFORMATION THEORY
-- ENTROPY OF A SOURCE WITH MEMORY --

- Recall that in a source S with memory, there is inter-symbol correlation
- Entropy $H(S)$ of source S is still the minimum average bits per symbol
- While it is still true that bits needed for symbol a_i by itself are at least $-\log_2 p_i$ bits, a sequence of symbols will need less than the sum of bits per symbol because there is correlation between those symbols
- Therefore, the previous mathematical expression of $H(S)$ in the memoryless source does not apply in a source with memory
- Computing the true entropy of a source with memory is indeed quite complicated and may be impossible in some cases
- But there is a way out: Entropy of *adjoint* sources

INFORMATION THEORY (ADJOINT SOURCES)

- Adjoint Source of Order N of a source with memory:



$$P(100) = \frac{6}{16}, P(110) = \frac{5}{16}, P(000) = \frac{2}{16}$$
$$P(010) = P(011) = P(111) = \frac{1}{16}$$

  - Split the source blocks of N symbols
  - Treat each block A as a macrosymbol, and compute its probability P_A
  - Treat the sequence of macrosymbols as a memoryless source. That is, treat the macrosymbol) as independent of one another
  - The entropy of this adjoint source is: $H_N(S) = -\sum_A P_A \log P_A$ ⟶ $H_3(S) = 2.18$

- **Theorem** (Shannon): $\frac{H_N(S)}{N} \to H(S)\ as\ N \to \infty$

As N gets larger, $\frac{H_N(S)}{N}$ gets smaller until $\approx H(S)$

- This implies that for any source S with memory, if we divide it into blocks of large enough size N and then block-code it without taking advantage of inter-block correlation, then we still approximate the performance of the best coder S.
- **Therefore, block coding is a good way to go when we have correlation**

EXERCISES

- **Exercise**: Suppose you have a large file of English text of N characters. How will you compute the probability of the letter 'h' in that file?
- **Exercise**: The conditional probability $\Pr[h/t]$, is defined to be the probability that the current letter (in the same text file as above) is 'h' given that the previous letter is 't'. How will you compute $\Pr[h/t]$ in that file?
- **Exercise**: Which of the following possibilities do you expect?

  (a) $\Pr[h/t] < \Pr[h]$, (b) $\Pr[h/t] = \Pr[h]$, (c) $\Pr[h/t] > \Pr[h]$
- **Exercise**: Which of the following possibilities do you expect?

  (a) $\Pr[x/t] < \Pr[x]$, (b) $\Pr[x/t] = \Pr[x]$, (c) $\Pr[x/t] > \Pr[x]$
- **Exercise**: How will you compute the probability of the string 'th' in that file?

NEXT LECTURE

- We will begin coverage of lossless compression
- We will cover popular lossless compression techniques
  - Huffman coding
  - Run-length encoding
  - Golumb coding
  - Arithmetic coding
- More lossless coding techniques will be covered in the lecture after that