10 – IMAGE COMPRESSION (A)

Image compression is the science of effectively coding digital images to reduce the number of bits required in representing an image. The purpose of doing so is to reduce the storage and transmission costs while maintaining good quality.

Consider the following storage and transmission requirements:

		Storage	Transmission @100 Mbps
Colour image	512×512 24 bits/pixel	768 KB	0.06 s
Video clip	640×480 24 bits/pixel 25 frames/s 1 minute	10.3 GB	1.8 mins
Video clip	1280×720 24 bits/pixel 25 frames/s 1 minute	30 GB	5.5 mins

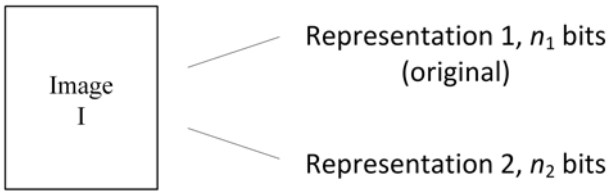
It is obvious that even with broadband connections, compression is needed to deliver multimedia data in a timely manner. (Note: HDTV is 1920×1080)

Applications requiring image compression:

- archival storage
- digital television (HDTV)
- DVD
- digital cameras
- video conferencing
- multimedia images/video

FUNDAMENTALS

The term image compression refers to the process of reducing the amount of data (the number of the bits) required to reqpresent an image. Let n_1 and n_2 denote the number of bits (or information-carrying units) in two representations of the same image.



Representation 2 achieves a compression ratio of

$$C_R = \frac{n_1}{n_2} \tag{1}$$

while representation 1 has a relative data redundancy

$$\begin{aligned} R_D &= (n_1 - n_2)/n_1 && (2)\\ &= 1 - (n_2/n_1) && (3)\\ &= 1 - \frac{1}{C_R} && (4) \end{aligned}$$

(a) $n_2 = n_1 \Rightarrow C_R = 1,\ R_D = 0$. The first representation contains no redundant data.

(b) $n_2 < n_1 \Rightarrow C_R > 1,\ 0 < R_D \leq 1$. There is compression and redundant data.

(c) $n_2 > n_1 \Rightarrow C_R < 1,\ R_D < 0$. This is the normally undesirable case of data expansion.

n_1	n_2	R_D	C_R
$10,000$	$10,000$	0	1
$10,000$	$5,000$	0.50	2
$10,000$	$20,000$	-1	0.5

In digital image compression, three basic data redundancies can be identified and exploited:

- coding redundancy
- interpixel redundancy
- psychovisual redundancy

Coding

A code is a system of symbols (letters, bits, etc.) used to represent a body of information or a set of events. Each piece of information or event is assigned a sequence of *code symbols* called a *code word*. The number of symbols in each code word is its *length*.

Example

Consider the binary coding of the decimal digits. The correspondence of binary sequences to decimal digits given in the table is an example of a code. The 10 decimal digits are called the *message symbols*, and the 10 binary sequences are the *code words*. Each code word is of length 4 bits.

Decimal digit	*Binary representation*
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Example - Morse Code

Code symbols: • (dot) — (dash)

Message (source) symbol	Code word	Length
a	• —	2
b	— • • •	4
c	— • —	3
d	— ••	3
e	•	1
f	•• — •	4
etc.		

A code word consists of a sequence of code symbols.

A message is a sequence of message symbols.

message	coded message
b e d	— • • • • — ••

A	.-	**M**	--	**Y**	-.--	**6**	-....
B	-...	**N**	-.	**Z**	--..	**7**	--...
C	-.-.	**O**	---	**Ä**	.-.-	**8**	---..
D	-..	**P**	.--.	**Ö**	---.	**9**	----.
E	.	**Q**	--.-	**Ü**	..--	.	.-.-.-
F	..-.	**R**	.-.	**Ch**	----	,	--..--
G	--.	**S**	...	**0**	-----	?	..--..
H	....	**T**	-	**1**	.----	!	··--·
I	..	**U**	..-	**2**	..---	:	---...
J	.---	**V**	...-	**3**	...--	“	.-..-.
K	-.-	**W**	.--	**4**	....-	‘	.----.
L	.-..	**X**	-..-	**5**	.....	=	-...-

Decoding, the inverse of coding, reconstructs the data. In this process, one needs to know both the specific rules according to which a data item is translated into a single code word (the "code book") and the organization of these data. In the case of a single digital image, for example, a knowledge of the number of rows and columns is required.

Decoding is not always straightforward. Consider this code

Message symbol	*Code word*
a_1	0
a_2	01
a_3	001
a_4	111

This binary sequence

$$111001$$

might have arisen from

$$a_4 a_3$$

or from

$$a_4 a_1 a_2$$

It is often a requirement that codes are *uniquely decodable*.

A natural (or straight) binary code is one in which each event or piece of information to be encoded (such as gray-level values) is assigned one of 2^m m-bit binary code words from an m-bit binary counting sequence. For the case $m = 2$, we have four 2-bit binary code words which we assign to four symbols:

Symbol	*Code word*
a_1	00
a_2	01
a_3	10
a_4	11

The images that we have considered so far have been coded and saved in the natural code, i.e., a gray level of value g is coded with its 8-bit binary equivalent. A gray level of 200, for example, is coded as 11001000.

Example

Letter	*Code 1*	*Code 2*	*Code 3*
I	0	000	01
J	1	001	00
K	10	010	10
L	11	011	1101
M	100	100	1100
N	101	101	11100
O	110	110	11101
P	111	111	11110

The table lists three codings of the letters I to P.

- Clearly, Code 1 is *not* uniquely decodable. It is not suitable for transmission of more than one code word, i.e., one letter. For example, the sequence 1000110 may be decoded in several ways, one of which is KIIJK.
- Code 2 is uniquely decodable. Any sequence with a number of bits equal to a multiple of 3 is allowed, and any such sequence, e.g., 010110100, will be unambiguously decipherable.
- Code 3, though it has variable code word lengths, is also uniquely decodable.

Coding Redundancy

Assume that a discrete variable r_k represents the gray levels of an image, and that each r_k occurs with probability $p_r(r_k)$:

$$p_r(r_k) = n_k/n \qquad k = 0, 1, 2, \ldots, L-1.$$

L is the number of gray levels, n_k is the number of times that the kth gray level appears in the image, and n is the total number of pixels in the image.

If the number of bits used to represent each value of r_k is $l(r_k)$, the average number of bits required to represent each pixel is

$$\bar{L} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k) \tag{5}$$

The total number of bits required to code an $M \times N$ image is $MN\bar{L}$.

Example

An 8-level image has the following gray-level distribution:

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(n_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1$	0.25	001	3	01	2
$r_2 = 2$	0.21	010	3	10	2
$r_3 = 3$	0.16	011	3	001	3
$r_4 = 4$	0.08	100	3	0001	4
$r_5 = 5$	0.06	101	3	00001	5
$r_6 = 6$	0.03	110	3	000001	6
$r_7 = 7$	0.02	111	3	000000	6

If a natural 3-bit binary code (Code 1) is used to represent the 8 possible gray levels,

$$\bar{L} = 3 \text{ bits}$$

.

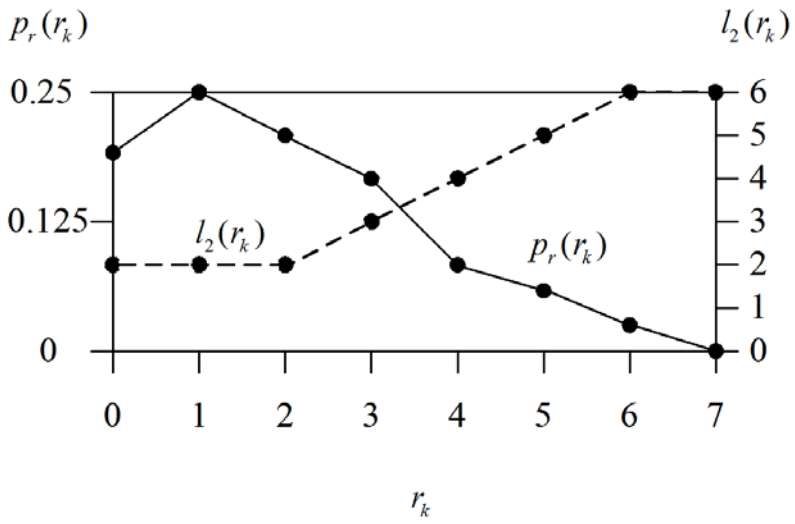
If Code 2 is used,

$$\begin{aligned} \bar{L} &= \sum_{k=0}^{7} l_2(r_k) p_r(r_k) \\ &= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + \\ &\quad 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02) \\ &= 2.7 \text{ bits} \end{aligned}$$

Comparing Code 1 and Code 2,

$$\begin{array}{lll} \text{Compression ratio} & C_R & = 3/2.7 = 1.11 \\ \text{Redundancy} & R_D & = 1 - \frac{1}{1.11} = 0.099 \end{array}$$

Here we show both the histogram of the image, $p_r(r_k)$, and $l_2(r_k)$. Since $l_2(r_k)$ increases as $p_r(r_k)$ decreases, the shortest codewords in Code 2 are assigned to the gray levels that occur most frequently.

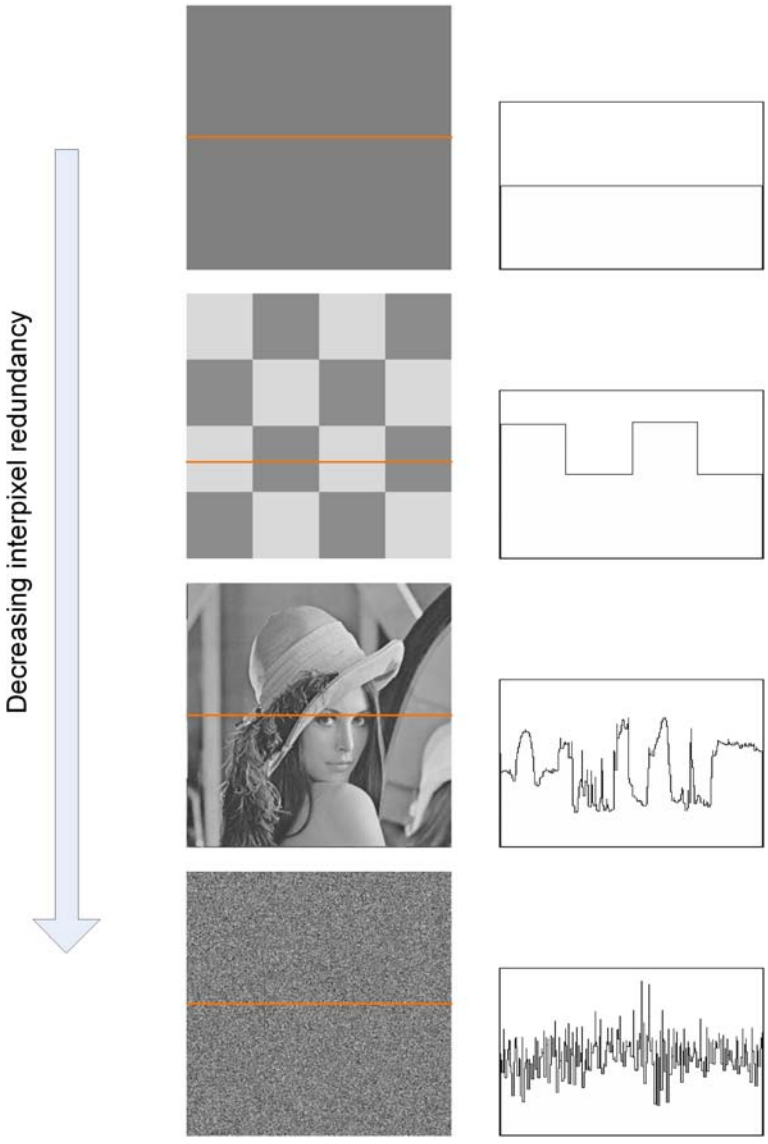


In the example, assigning fewer bits to the more probable gray levels than to the less probable ones achieves data compression. The process is known as *variable-length coding*.

If the coding is such that more code symbols than absolutely necessary are are used, we then have *coding redundancy*. In general, coding redundancy is present when the codes assigned to a set of events (such as gray-level values) have not been selected to take full advantage of the probabilities of the events. It is almost always present when an image's gray levels are represented with a straight or natural binary code.

Interpixel redundancy

This is also known as spatial redundancy, geometric redundancy, and interframe redundancy. This arises because the value of any given pixel can be reasonably predicted from the value of its neighbours. The information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel is redundant; it could have been estimated on the basis of the neighbour's values. For video images, this concept can be extended to include redundancy between frames of image data.



Psychovisual Redundancy

The eye does not respond with equal sensitivity to all information. Certain information simply has less relative importance than other information in normal visual processing. This information is said to be psychovisually redundant. It can be eliminated without significantly impairing the quality of image perception.

Since the elimination or reduction of psychovisually redundant data results in a loss of quantitative information, it is commonly referred to as *quantization*. This terminology is consistent with normal usage of the word, which generally means the mapping of a broad range of input values to a limited number of output values. Since it is an irreversible operation, quantization results in lossy data compression.

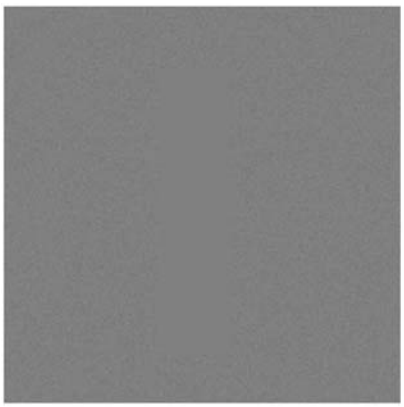
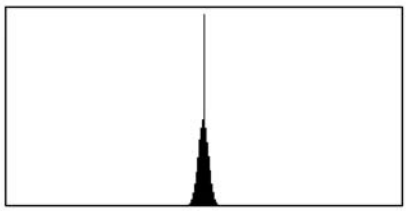
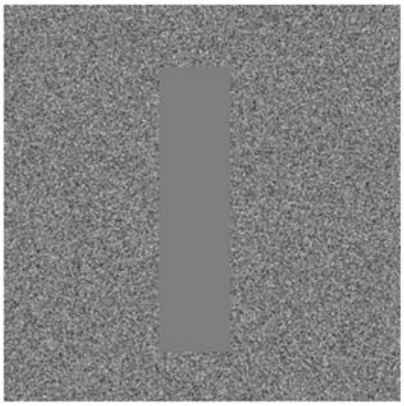


Image with psychovisual redundancy and its histogram





After contrast enhancement to highlight rectangle

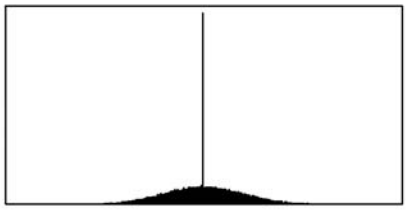
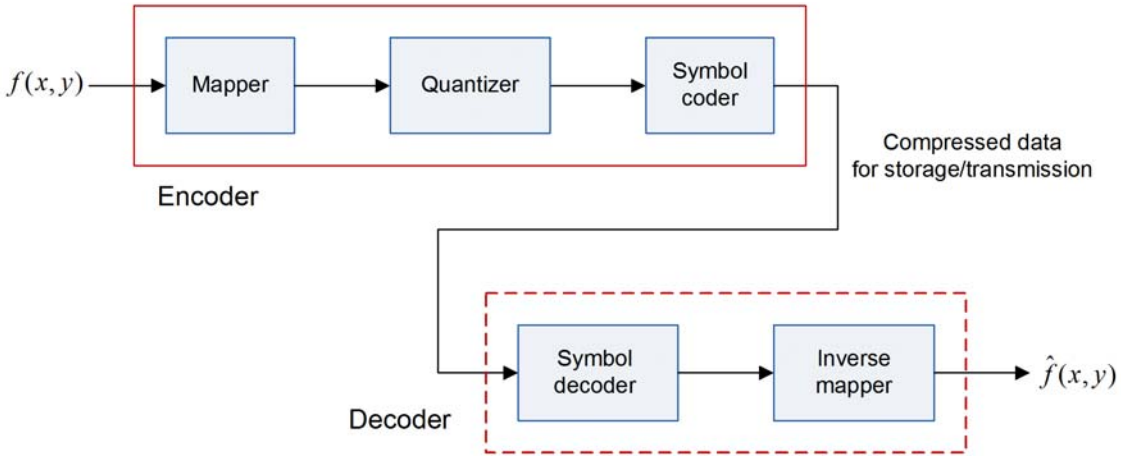


IMAGE COMPRESSION SYSTEMS



A compression system consists of an *encoder* and a *decoder*. The encoder creates a compressed representation of the input image $f(x, y)$. This representation is then stored or transmitted to another location. The decoder acts on this encoded representation to generate the reconstructed output (the decompressed image), $\hat{f}(x, y)$. A codec is a device or program that can perform both encoding and decoding.

$\hat{f}(x, y)$ may or may not be an exact replica of $f(x, y)$. If it is, the compression is called error free, lossless or information preserving. If not, it is called lossy and the reconstructed output image is distorted.

The source encoder reduces coding, interpixel, or psychovisual redundancies in the input image. The approach can be modelled by a series of three independent operations.

(a) *Mapper:* transforms the input data into a format designed to reduce spatial redundancy. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image. Examples of such operations are run-length coding and transform coding.

(b) *Quantizer:* reduces the accuracy of the mapper's output in accordance with some pre-established fidelity criterion. This stage reduces the psychovisual redundancies of the input image. The operation is irreversible and must be omitted when error-free compression is desired.

(c) *Symbol coder:* creates a fixed- or variable-length code to represent the quantizer output.

All three stages are not necessarily included in every compression system. In addition, some compression techniques normally are modelled by merging blocks that are physically separate in the figure. In predictive compression, the mapper and quantizer are often represented by a single block, which simultaneously performs both operations.

The source decoder contains only two components: a *symbol decoder* and an *inverse mapper*. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general source decoder model.

Fidelity Criteria

Compressed images may suffer from a loss of information. Two general classes of criteria are used to quantify the nature and extent of information loss: (1) objective fidelity criteria and (2) subjective fidelity criteria.

When the level of information loss can be expressed as a function of the original or input image, it is said to be based on an *objective fidelity criterion.*

Let $f(x, y)$ represent an input image and let $\hat{f}(x, y)$ denote an estimate or approximation of $f(x, y)$ that results from compressing and subsequently decompressing the input. For any value of x, y, the error $e(x, y)$ between $f(x, y)$ and $\hat{f}(x, y)$ can be defined as

$$e(x, y) = \hat{f}(x, y) - f(x, y) \tag{6}$$

so that the total error between the two image is

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)] \tag{7}$$

where the images are of size $M \times N$.

The *root-mean-square error*, e_{rms}, between $f(x, y)$ and $\hat{f}(x, y)$ is then the square root of the squared error averaged over the $M \times N$ array, or

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{1/2} \tag{8}$$

A closely related objective fidelity criterion is the mean-square signal-to-noise-ratio of the compressed-decompressed image given by

$$\text{SNR}_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2} \tag{9}$$

In dB, we have

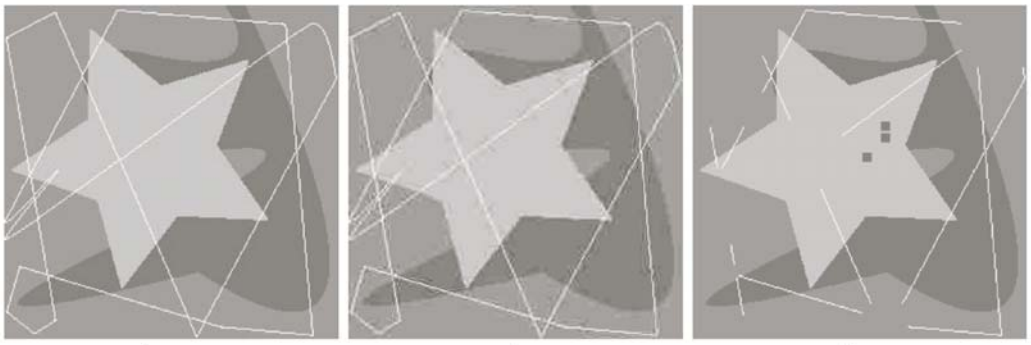
$$\text{SNR}_{ms} = 10 \log_{10} \left(\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2} \right) \tag{10}$$

The rms value of the signal-to-noise-ratio, denoted by SNR_{rms}, is obtained by taking the square root of SNR_{ms}.

Measuring image quality by the subjective evaluations of a human observer is often more appropriate since most decompressed images are ultimately viewed by human beings. This can be accomplished by showing a decompressed image to a viewers and averaging their evaluations. An example of a rating scale is shown in the table. The evaluations are said to be based on *subjective fidelity criteria*.

Value	*Rating*	*Description*
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Original image, I



I_1 I_2 I_3

e_{rms} = 5.17
Rating = excellent

e_{rms} = 15.67
Rating = passable

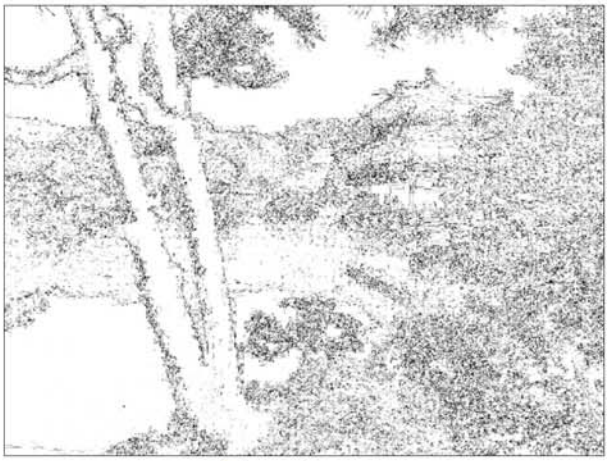
e_{rms} = 14.17
Rating = inferior

original

CR = 7.6, SNR = 33.7dB

Error image
(contrast-enhanced and inverted)

CR = 18.9, SNR = 25.4 dB



Error image
(contrast-enhanced and inverted)

BASIC CONCEPTS FROM INFORMATION THEORY

Information theory provides the mathematical framework needed to questions such as these:

- What is the smallest number of bits that is sufficient to code an image without loss of information?
- How efficient is one coding scheme compared to another?

Measuring Information

The fundamental premise of information theory is that the generation of information can be modelled as a probabilistic process that can be measured in a manner that agree with intuition.

A random event E that occurs with probability $P(E)$ is said to contain

$$I(E) = \log_r \frac{1}{P(E)} = -\log_r P(E) \tag{11}$$

units of information. Note that if $P(E) = 1$, (i.e., the event always occurs), $I(E) = 0$ and no information is attributed to it. Because no uncertainty is associated with the event, no information would be transferred by communicating that the event has occurred.

Consider an event E. If we are told that event E has occurred, then we have received $I(E)$ units of information, where

$$I(E) = \log_r \frac{1}{P(E)}$$

If base 2 logarithm is used ($r = 2$), the resulting unit of information is called a bit:

$$I(E) = \log_2 \frac{1}{P(E)} \text{ bits}$$

If base 10 logarithm is used, the unit of information is the Hartley:

$$I(E) = \log_{10} \frac{1}{P(E)} \text{ Hartleys}$$

We see that

$$1 \text{ Hartley} = 3.32 \text{ bits}$$

In general, if we use a logarithm to base r,

$$I(E) = \log_r \frac{1}{P(E)} \ r\text{-ary units}$$

Note that, if $P(E) = \frac{1}{2}$, $I(E) = -\log_2(\frac{1}{2})$, or 1 bit. That is, one bit is the amount of information we obtain when one of two possible equally likely alternatives is specified.

$P(E)$	$I(E)$
1	0 bit
1/2	1.0 bit
1/3	1.6 bits
1/4	2.0 bits
1/5	2.3 bits

Entropy

Assume that an information source generates a random sequence of symbols from a finite set of possible symbols. The set of source symbols $\{a_1, a_2, \ldots, a_J\}$ is referred to as the *source alphabet*. The elements of the set, denoted a_j, are called *symbols*.

The probability of the event that the source will produce symbol a_j is $P(a_j)$, and

$$\sum_{j=1}^{J} P(a_j) = 1 \tag{12}$$

The $J \times 1$ vector

$$\mathbf{z} = [P(a_1), P(a_2), \ldots, P(a_J)]^T$$

represents the source symbol probabilities.

Example

$$A = \{a_1, a_2, a_3, a_4\}, \quad \mathbf{z} = \begin{bmatrix} P(a_1) \\ P(a_2) \\ P(a_3) \\ P(a_4) \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{bmatrix}$$

Source A $\longrightarrow$	a_3	a_2	a_1	a_2	a_3	$\ldots$	
$I(a_i)$	1.74	2.32	3.32	2.32	1.74	$\ldots$	(bits)

If 100 symbols are generated, we would expect a_1 to appear 10 times, a_2 to appear 20 times, a_3 to appear 30 times, and a_4 to appear 40 times. The average information per source output (symbol) is then

$$\begin{aligned} I_{av} &= \frac{1}{100}\left[10 \times I(a_1) + 20 \times I(a_2) + 30 \times I(a_3) + 40 \times I(a_4)\right] \\ &= P(a_1) \times I(a_1) + P(a_2) \times I(a_2) + P(a_3) \times I(a_3) + P(a_4) \times I(a_4) \\ &= 1.85 \text{ bits} \end{aligned}$$

The average information per source output, denoted by $H(A)$, is called the *uncertainty* or *entropy* of the source, and is given by

$$H(A) = -\sum_{j=1}^{J} P(a_j) \log P(a_j) \tag{13}$$

As its magnitude increases, more uncertainty and thus more information is associated with the source. When the source symbols are equally probable, entropy is maximized, and the source provides the greatest possible average information per source symbol.

Example

Consider the source $A = \{a_1, a_2, a_3\}$ with $P(a_1) = \frac{1}{2}$, $P(a_2) = P(a_3) = \frac{1}{4}$. The entropy of the source is

$$\begin{aligned} H(A) &= \frac{1}{2}\log 2 + \frac{1}{4}\log 4 + \frac{1}{4}\log 4 \\ &= 1.5 \quad \text{bits} \end{aligned}$$

Consider the source $B = \{b_1, b_2, b_3\}$ with $P(b_1) = P(b_2) = P(b_3) = \frac{1}{3}$, then

$$\begin{aligned} H(B) &= \frac{1}{3}\log 3 + \frac{1}{3}\log 3 + \frac{1}{3}\log 3 \\ &= 1.58 \quad \text{bits} \end{aligned}$$

Example

Consider the English language. If the probabilities were equal for each letter, then

$$H = \log_2 26 = 4.70 \text{ bits}$$

i.e., the average information per letter would be 4.70 bits.

However, the relative frequencies of occurrence are unequal — e.g., e: 0.131, t: 0.105, z: 0.00077 — which leads to a reduction in average information to 4.15 bits per letter.

Example

In images, gray values correspond to symbols. So, for a 256-level image, we have

$a_0 =$ gray level 0, $\quad a_1 =$ gray level 1, $\quad \ldots, \quad a_{255} =$ gray level 255.

Consider the four 256×256 images, A, B, C, and D (and their respective histograms) below.

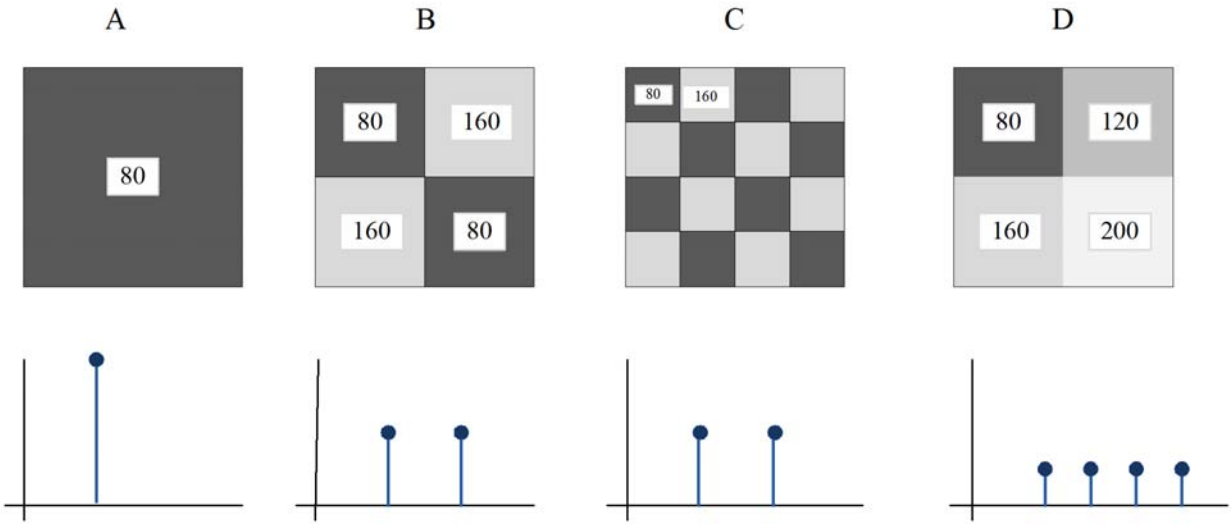


Image A:

$$H(A) = 1 \log(1) = 0 \text{ bit}$$

Image B:

$$\begin{aligned} H(B) &= \frac{1}{2}\log(2) + \frac{1}{2}\log(2) \\ &= 1.0 \text{ bit} \end{aligned}$$

Image C:

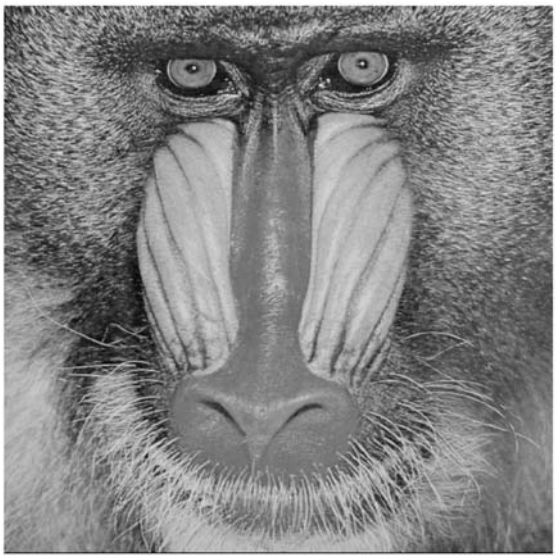
$$\begin{aligned} H(C) &= \frac{1}{2}\log(2) + \frac{1}{2}\log(2) \\ &= 1.0 \text{ bit} \end{aligned}$$

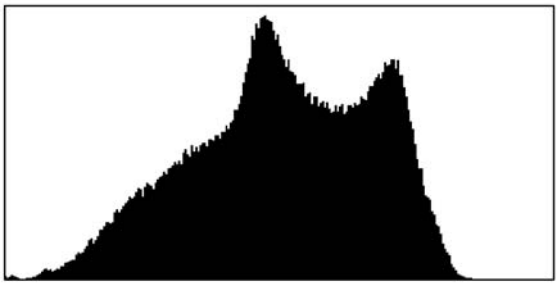
Image D:

$$\begin{aligned} H(D) &= \frac{1}{4}\log(4) + \frac{1}{4}\log(4) + \frac{1}{4}\log(4) + \frac{1}{4}\log(4) \\ &= 2.0 \text{ bits} \end{aligned}$$

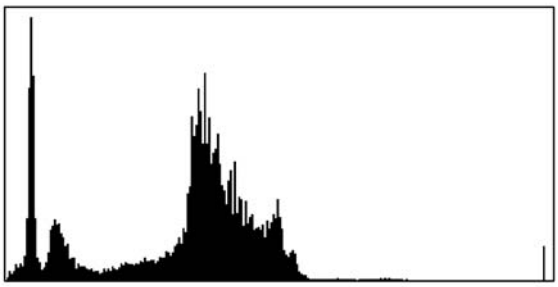
Examples

H = 8.0 bits





H = 7.358 bits



H = 6.667 bits

It can be proven that given an image I with entropy $H(I)$:

For any uniquely decodable code, C, $H(I)$ is a lower bound on the average word length $\bar{L}_C$:

$$\bar{L}_C \geq H(I) \tag{14}$$

Thus, $H(I)$ puts a limit on the achievable average word length. From this, it follow that the efficiency of a code C is given by

$$\eta = \frac{H(I)}{\bar{L}_C} \tag{15}$$

Examples

Source A :

Symbol	Prob.	Code I	Code II	Code III
a_1	$\frac{1}{2}$	00	0	0
a_2	$\frac{1}{4}$	01	10	1
a_3	$\frac{1}{4}$	10	11	10
	$\bar{L} =$	2.0	1.5	1.25
	$\eta =$	75%	100%	120%

- $H(A) = 1.5$ bits
- $\bar{L} \geq 1.5$
- Code III is not a uniquely decodable code.

Source B :

Symbol	Prob.	Code I	Code II
b_1	$\frac{1}{3}$	00	0
b_2	$\frac{1}{3}$	01	10
b_3	$\frac{1}{3}$	10	11
	$\bar{L} =$	2.0	1.67
	$\eta =$	79%	95%

- $H(B) = 1.58$ bits
- $\bar{L} \geq 1.58$
- Code II is the "best" code (shortest $\bar{L}$) that can be obtained by coding the symbols individually.