Image Processing

Lecture 08: Image Compression – I

(Ch8 Image Compression and Watermarking)

Zhiguo Zhang

zhiguozhang@hit.edu.cn

Review of Last Lecture

- In the last lecture we learnt:
    - Full-color processing
        - Color transformations
        - Smoothing and sharpening
        - Color segmentation
        - Noise in color images

Contents of This Lecture

- Fundamentals

- Data Redundancies

- Image Compression Models

- Lossless Image Compression

    - Huffman Coding

Fundamentals

Motivation

- Much of the on-line information is graphical or pictorial, storage and communications requirements are immense.
- The spatial resolutions of today's imaging sensors and the standard of broadcast television are greatly developed.
    - 1991, Kodak, MC-200, 0.4M pixels, 1.7M Japan Yen (around 9500RMB).
    - 2024, Canon IXUS 285, 20-30M pixels, 2258RMB.
    - VGA resolution: 320×240, 640×480, 1280×720, 1920×1080 (1080P), 3840×2160 (4K), 7680×4320 (8K)…
- Methods of image compression are of practical and commercial interest.

Fundamentals

- Image compression addresses the problem of reducing the amount of data required to represent a digital image.
    - Remove the redundant data.
    - Statistically uncorrelated data set.
- The transformation is applied prior to storage or transmission of the image.
- Transformed images can be decompressed to reconstruct the original image or an approximation of it.

Applications

- Increased spatial resolutions
  - Image sensors
  - Broadcast television standards
  - Tele-video-conferencing
  - Remote sensing
  - Digital library, medical imaging, fax, ……
- An ever-expanding number of applications depend on the efficient processing, storage, and transmission of binary, gray-scale, and color images.

Classification

- Lossless compression
  - Also called information preserving compression or error-free compression.
  - Useful for legal and medical documents, remote sensing.

- Lossy compression
  - Provide higher levels of data reduction.
  - Useful in broadcast television, video conference, internet image transmission.
  - Some errors or loss can be tolerated.

Data and Information

- Distinguish the meaning of **data** and **information**:
    - Data: the means by which information is delivered.
    - Information: various amounts of data may be used to present the same amount of information.
- Example: Story
    - Story is information.
    - Word is data.

Data and Information

- Data Compression:
    - The process of reducing the amount of data required to represent a given quantity of information.
- Data Redundancy
    - If the two individuals use a different number of words to tell the same basic story.
    - At least one includes non-essential data. It is thus said to contain data redundancy.
    - Examples. 10^8: 100000000; HIT: Harbin Institute of Technology

Fundamentals

- **Data redundancy is the central issue in digital image compression.**

Data Redundancy

- **Data Redundancy:** not an abstract concept, but a mathematically quantifiable entity.
- If b and b' denote the number of information-carrying units in two data sets that represent the same information.
- The *relative data redundancy* R of the first data set can be defined as:

$$R = 1 - 1 / C$$

where *compression ratio* C is

$$C = b / b'$$

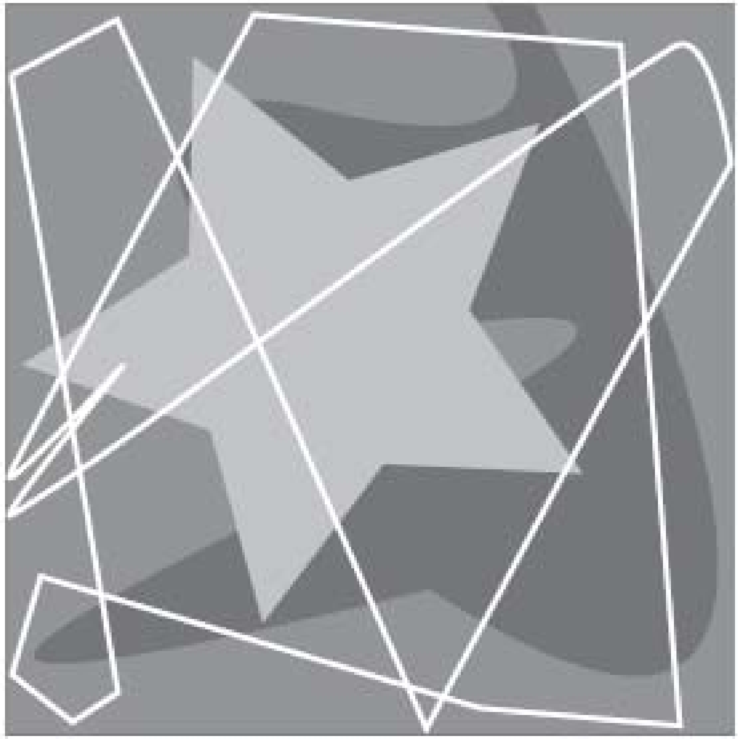
Data Redundancy

- C and R: $R = 1 - 1/C$, $C = b/b'$
  - When $b = b', C = 1$ and $R = 0$;
  - When $b' << b, C \to \infty$ and $R \to 1$;
  - When $b << b', C \to 0$ and $R \to -\infty$, not hoped situation.
  - Usually, C and R lie in the open intervals $[0, \infty)$ and $(-\infty, 1)$.
  - Example:
    - Compression ratio is 10.
    - Redundancy is 0.9.
    - It implies that 90% of the data in the first data set is redundant.

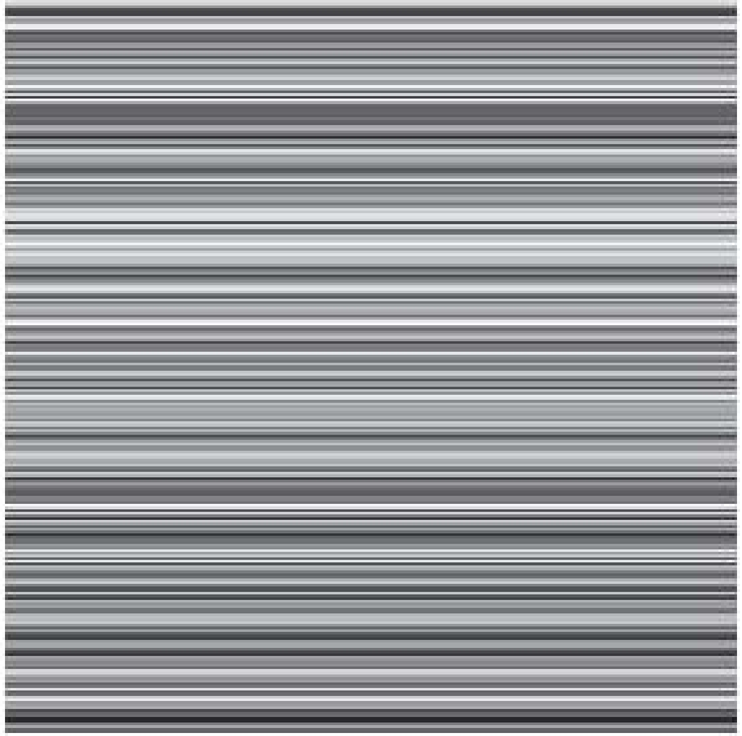
Data Redundancy

- In image processing, there are three basic data redundancies can be identified and exploited:
  - Coding Redundancy
  - Spatial and Temporal Redundancy
  - Irrelevant Information
- Data compression is achieved when one or more of these redundancies are reduced or eliminated.

Data Redundancy



coding redundancy



spatial redundancy

irrelevant information

Coding Redundancy

- A discrete random variable r_k in the interval [0, L – 1] represents the gray levels of an $M \times N$ image. Each r_k occurs with probability $p_r(r_k)$:

$$p_r(r_k) = \frac{n_k}{MN}, k = 0,1,2,..., L-1$$

where L is the number of intensity values, and n_k is the number of times that the *k*th intensity appears in the image.

Coding Redundancy

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

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

- The total number of bits required to code an $M \times N$ image is MNL_{avg}.

- For example, the gray levels of an image with a natural m-bit binary code. The constant m may be taken outside the summation, leaving only the sum of the $p_r(r_k)$ equals 1. Then: $L_{avg} = m$.

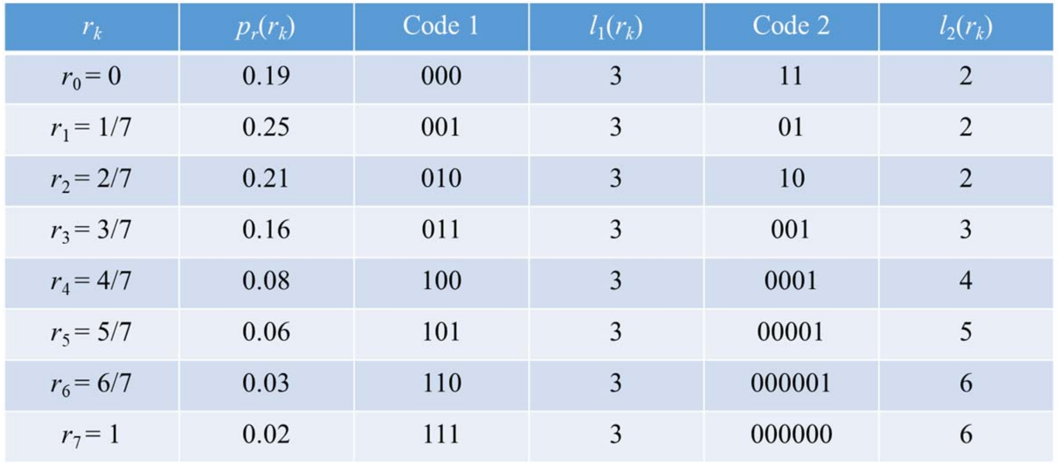
Coding Redundancy

- Example of coding for an 8-level image

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

Coding Redundancy

- For Code 2, the average number of bits required to code the image is reduced to:



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

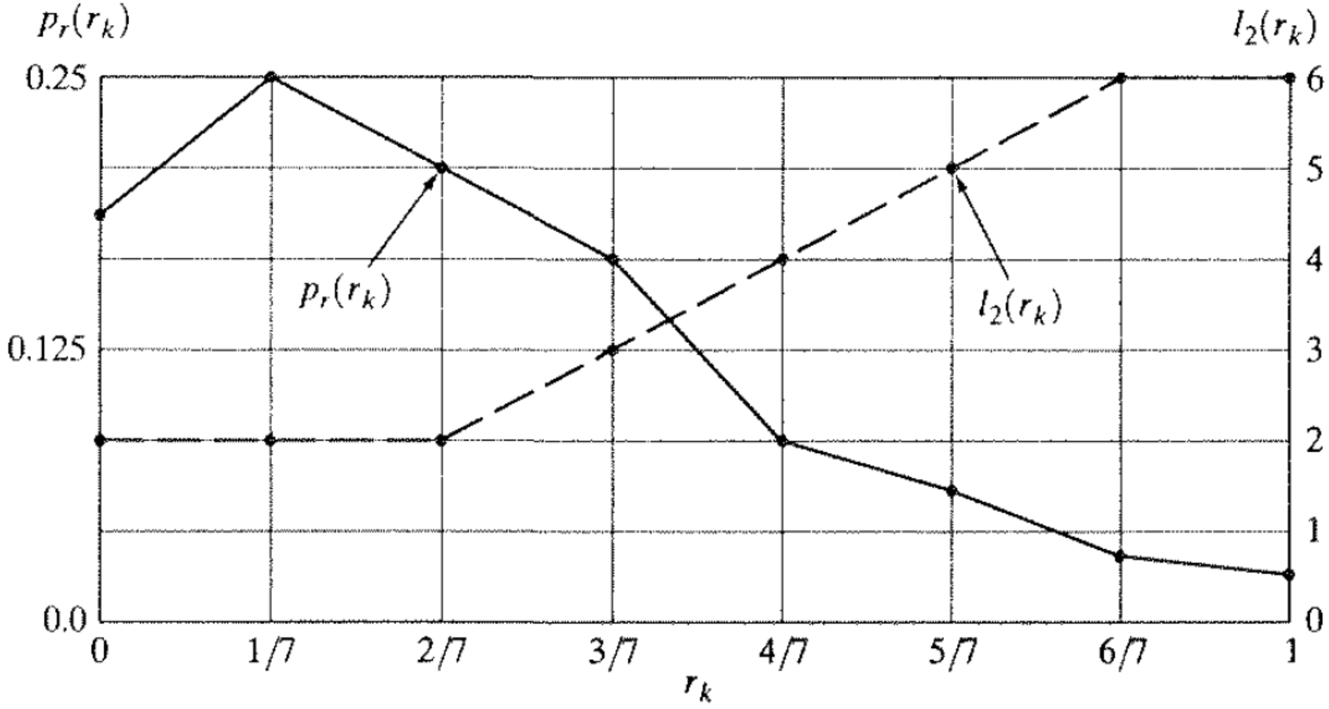
$$l_{avg} = \sum_{k=0}^{7} l(r_k)p_r(r_k)$$

$$= 2*0.19+2*0.25+2*0.21+3*0.16+4*0.08+5*0.06+6*0.03+7*0.02$$

$$= 2.7 bits$$

- The resulting compression ratio C is 3/2.7 or 1.11.
- Redundancy is R = 1−1/1.11 = 0.099.

Coding Redundancy



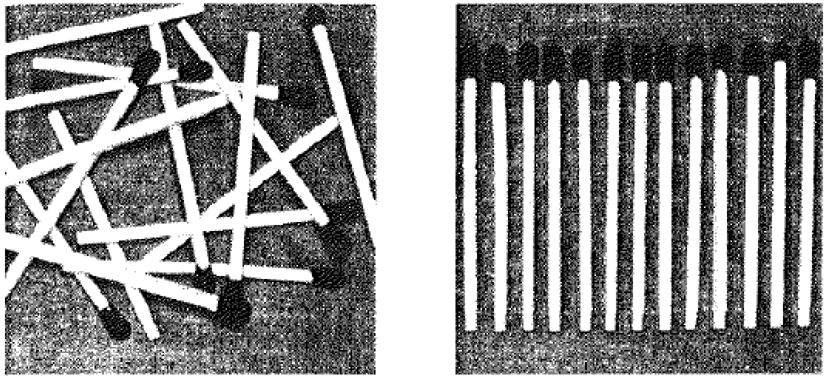
r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

Graphic representation of the fundamental basis of data compression through variable-length coding

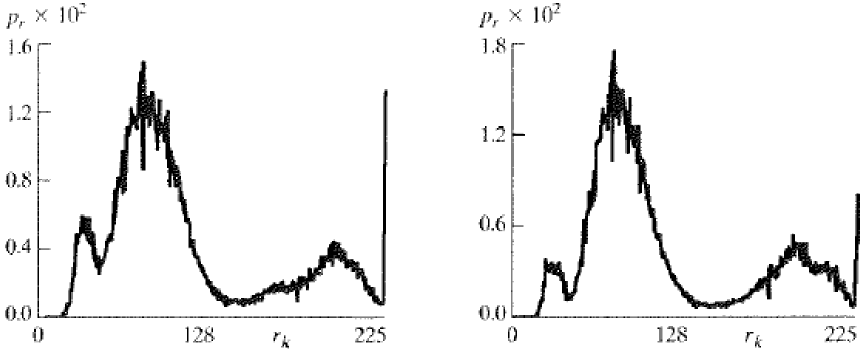
Coding Redundancy

- *Variable-Length Coding*: Assigning fewer bits to the more probable gray-levels.
- Coding redundancy exists,
    - when not taking full advantage of the probabilities of the events;
    - and it is almost always presented by using natural binary code.
- Underlying basis:
    - Certain gray levels are more probable than others.

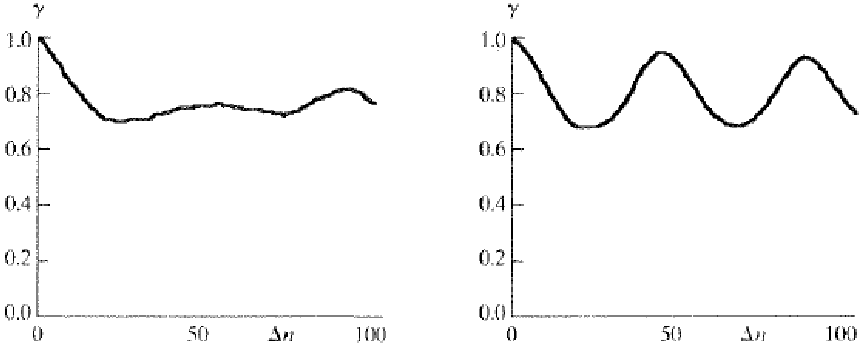
Spatial and Temporal Redundancy



Two images



Gray-level histograms



Normalized correlation coefficients along one line

Spatial and Temporal Redundancy

- The gray levels in these images are not equally probable, so variable-length coding can be used to reduce the coding redundancy.
- The coding process would not alter the level of correlation between the pixels within the images.
- The correlations come from the structural or geometric relationships between the objects in the image.
- These reflect another important data redundancy—spatial and temporal redundancy: one directly related to the interpixel correlations within an image.

Spatial and Temporal Redundancy

- The value of any given pixel can be predicted from the value of its neighbors.
- The information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel is redundant to an image.
- Other nomenclatures:
  - Interpixel redundancy
  - Geometric redundancy
  - Interframe redundancy

Spatial and Temporal Redundancy

Approaches to reduce spatial and temporal redundancy

- Transform into a more efficient (but usually nonvisual) format.
- For example, the difference between adjacent pixels can be used to represent an image.
- *Mapping*: transformations of the types that remove spatial and temporal redundancy.
- *Reversible Mappings*: the original image can be reconstructed from the transformed data.

Spatial and Temporal Redundancy

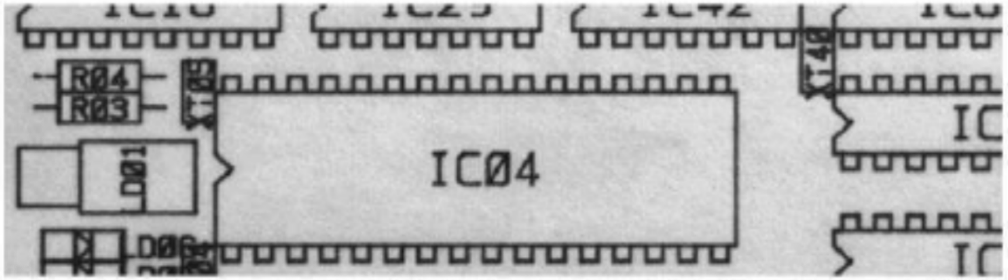
Run-length Coding

- Mapping the pixels along each scan line $f(x,0), f(x,1),\ldots, f(x, N–1)$ into a sequence of pairs $(g_1, w_1), (g_2, w_2), \ldots$
  - g_i denotes the i-th gray level encountered along the line,
  - w_i is the run length of the i-th run.
- “aabbbcddddd” can be represented as “a2b3c1d5”.
- 111110255555555778888888888888888, can be represented as: (1, 5)(0, 1)(2, 1)(5, 8)(7, 2)(8, 14).

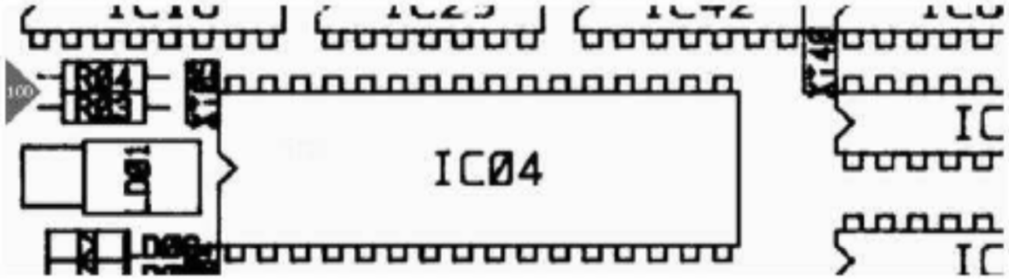
Spatial and Temporal Redundancy

- Illustration of run-length coding

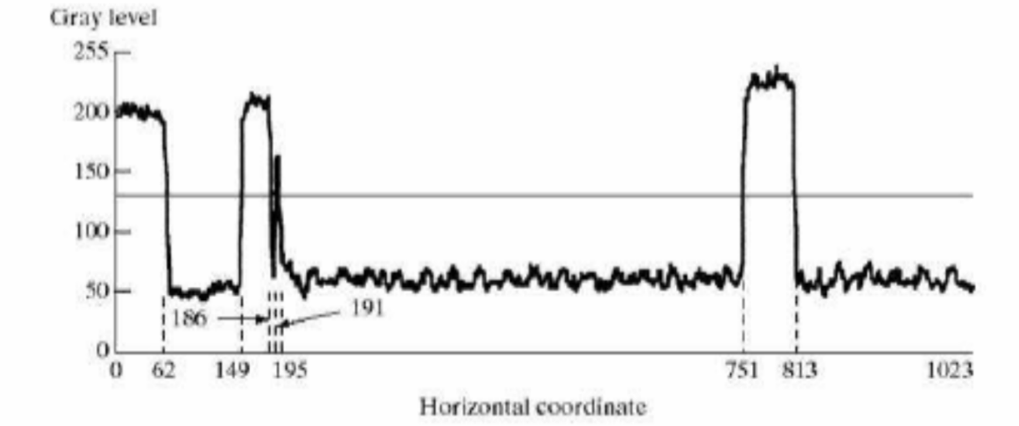
Original image



Binary image with line 100 marked



Line profile and binarization threshold



Run-length code (1, 63) (0, 87) (1, 37) (0, 5) (1, 4) (0, 556) (1, 62) (0, 210)

Irrelevant Information

- Eyes do not respond with equal sensitivity to all visual information.
- Certain information has less relative importance. This information is said to be **irrelevant redundant**.
- It can be eliminated without significantly impairing the quality of image perception.

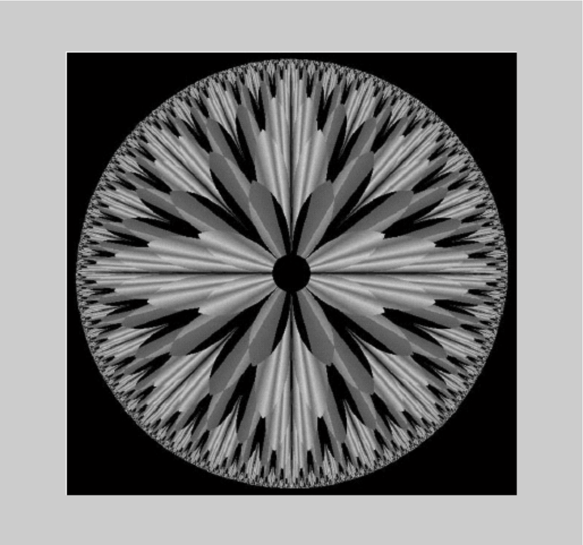
Irrelevant Information

- Human perception of the information in an image normally does not involve quantitative analysis of every pixel value in the image.
  - Find features such as edges or textual regions.
  - Mentally combines them into recognizable groupings.
  - The brain correlates these groupings with prior knowledge.
  - Then, complete the image interpretation process.

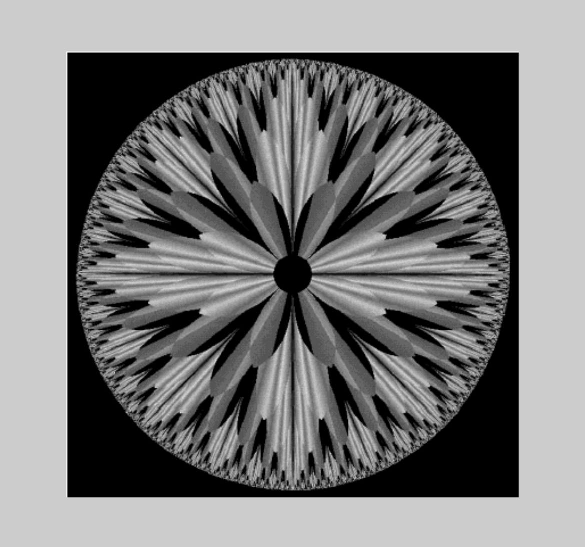
Irrelevant Information

- Lead to a loss of quantitative information (*quantization*).
  - Mapping a broad range of input values to a limited number of output values.
  - Irreversible Operation.

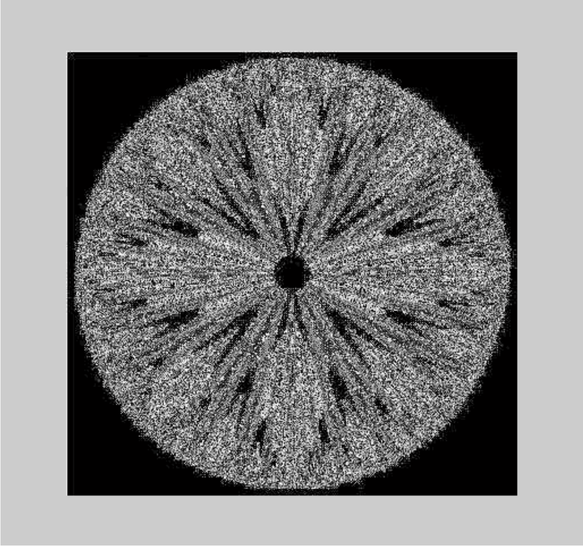
original



quantized

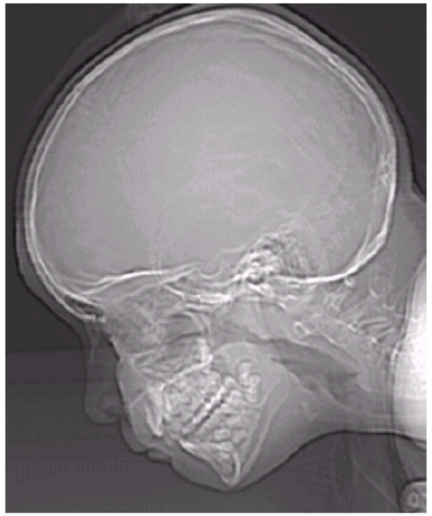


difference

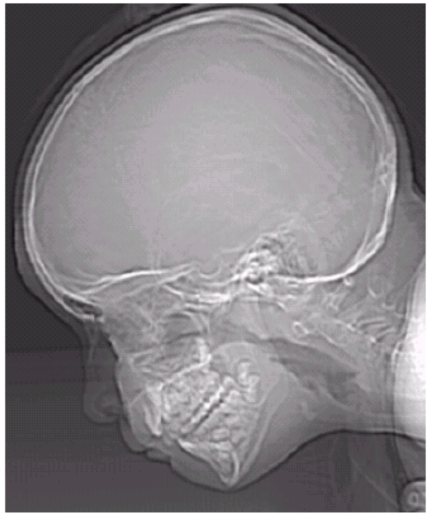


Irrelevant Information

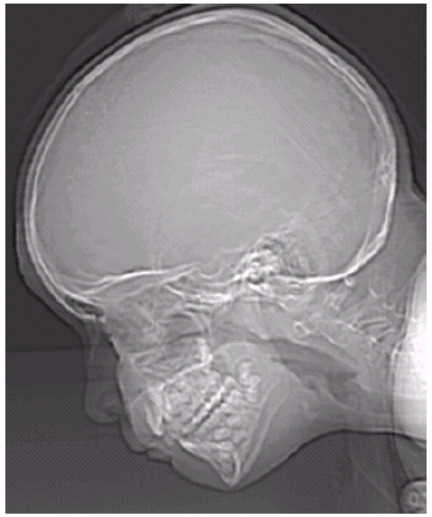
256 grey levels (8 bits per pixel)



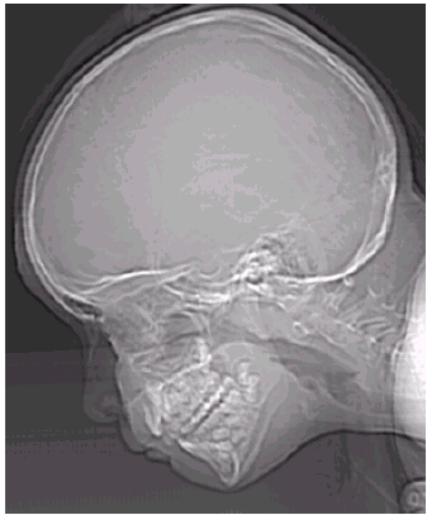
128 grey levels (7 bpp)

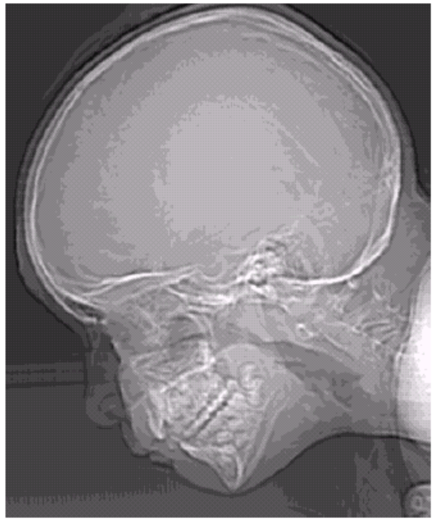


64 grey levels (6 bpp)

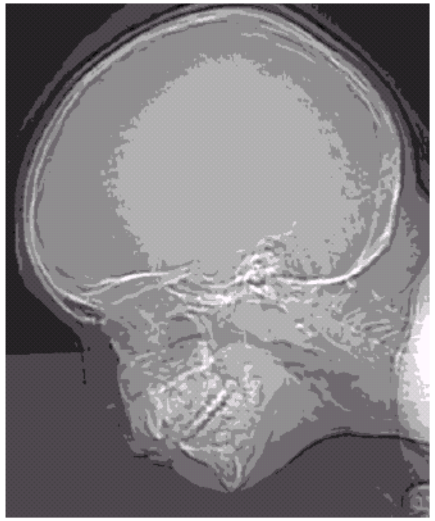


32 grey levels (5 bpp)

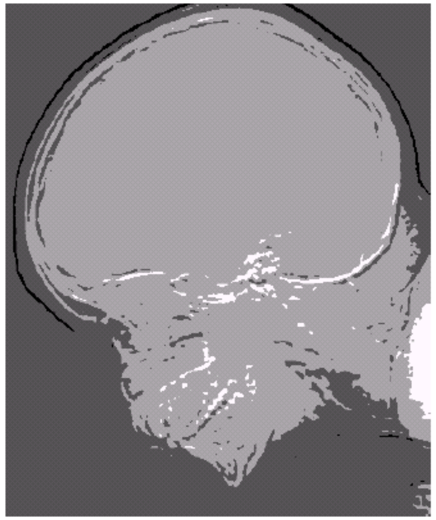




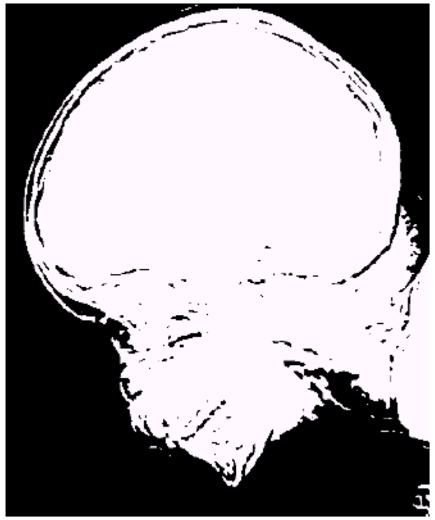
16 grey levels (4 bpp)



8 grey levels (3 bpp)



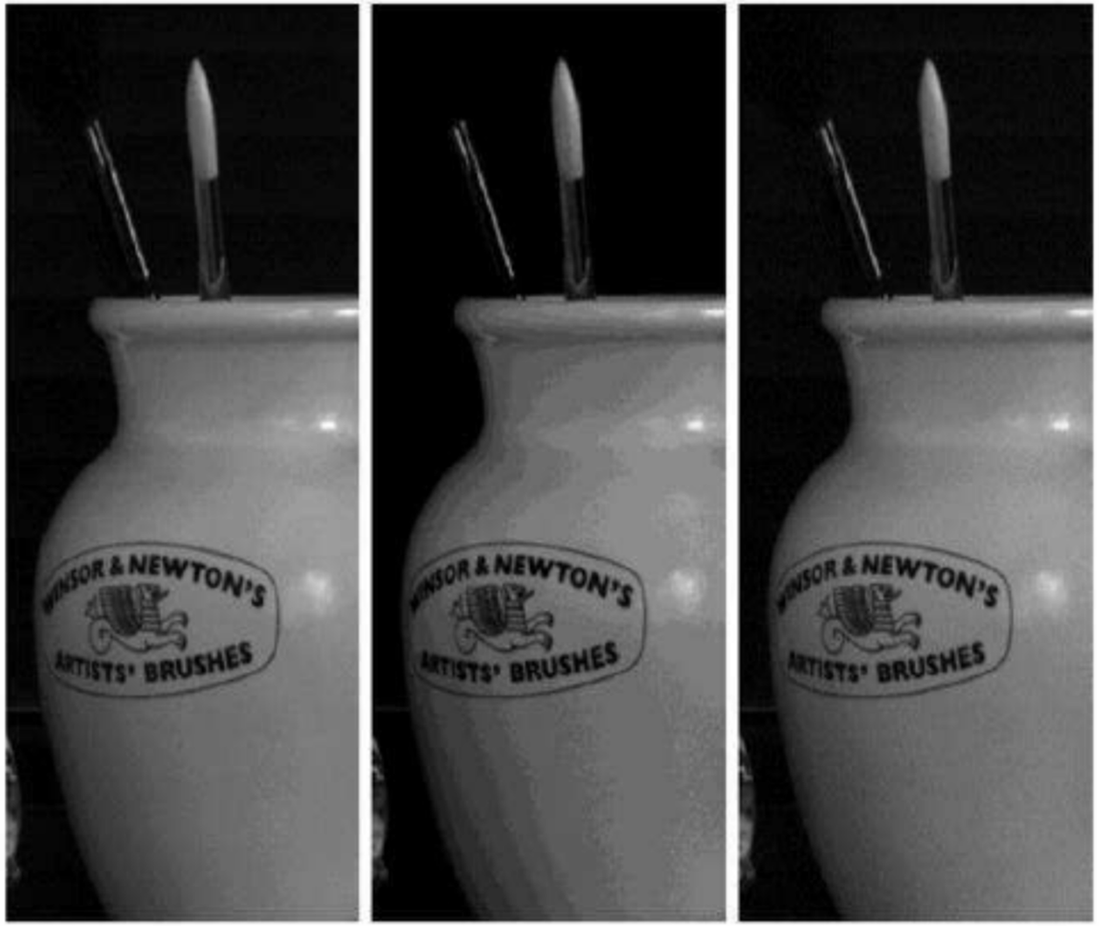
4 grey levels (2 bpp)



2 grey levels (1 bpp)

Irrelevant Information

- Example: Compression by quantization



a) Original image with 256 gray levels.

b) Uniform quantization to 16 gray levels.

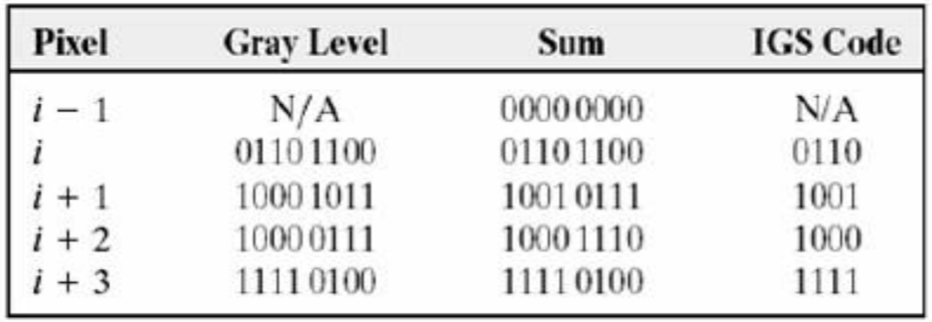
c) Improved Gray-Scale (IGS) quantization.

The compression are 2:1, but IGS is more complicated.

Irrelevant Information

Improved Gray-Scale (IGS) quantization

- A sum: initially set to zero.
- Add the four least significant bits of a previously generated sum with current 8-bit gray level.
- If the four most significant bits of the current value are 1111_2, 0000_2 is added instead.
- The four most significant bits of the resulting sum are used as the coded pixel value.



Pixel	Gray Level	Sum	IGS Code
$i-1$	N/A	0000 0000	N/A
i	0110 1100	0110 1100	0110
$i+1$	1000 1011	1001 0111	1001
$i+2$	1000 0111	1000 1110	1000
$i+3$	1111 0100	1111 0100	1111

Measuring Image Information

- How few bits are actually needed to represent the information in an image? That is, is there a minimum amount of data that is sufficient to describe an image without losing information?
- *Information theory* provides the mathematical framework to answer this and related questions.
- A random event E with probability $P(E)$ is said to contain

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

units of information. If $P(E) = 1$ (that is, the event always occurs), $I(E) = 0$ and no information is attributed to it.

Measuring Image Information

- The base of the logarithm in above equation determines the unit used to measure information. If the base m logarithm is used, the measurement is said to be in m-ary units.

- If the base 2 is selected, the unit of information is the *bit*.

- Note that if $P(E) = ½$, $I(E) = – \log_2 ½$ or 1 bit. That is, 1 bit is the amount of information conveyed when one of two possible equally likely events occurs. A simple example is flipping a coin and communicating the result.

Measuring Image Information

- Given a source of statistically independent random events from a discrete set of possible events $\{a_1, a_1, \ldots, a_J\}$, with associated probabilities $\{P(a_1), P(a_2), , \ldots, P(a_J)\}$, the average information per source output, called the *entropy* of the source, is

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

- a_j in this equation are called *source symbols*. Because they are statistically independent, the source itself is called a *zero-memory source*.

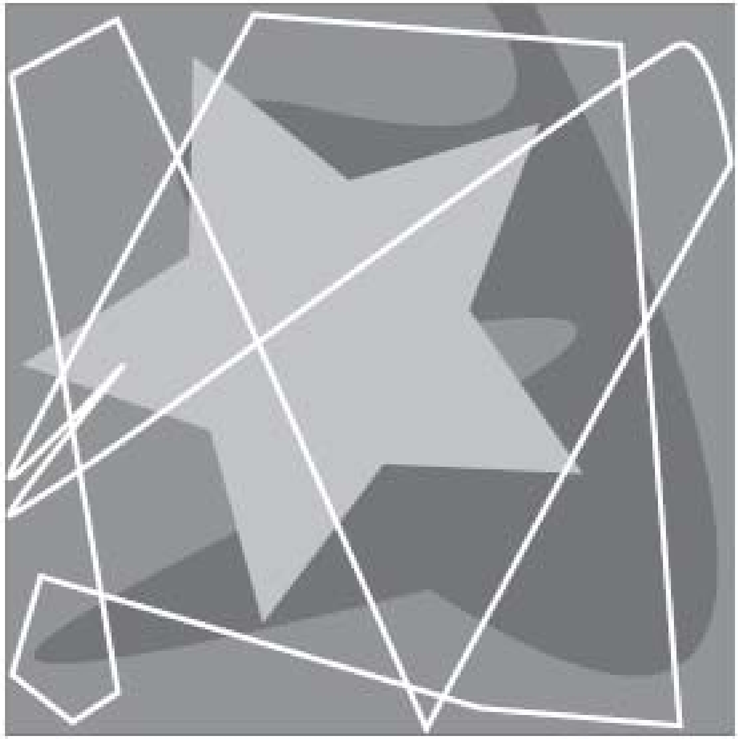
Measuring Image Information

- If an image is considered to be the output of an imaginary zero-memory "intensity source," we can use the histogram of the observed image to estimate the symbol probabilities of the source. Then, the intensity source's entropy becomes

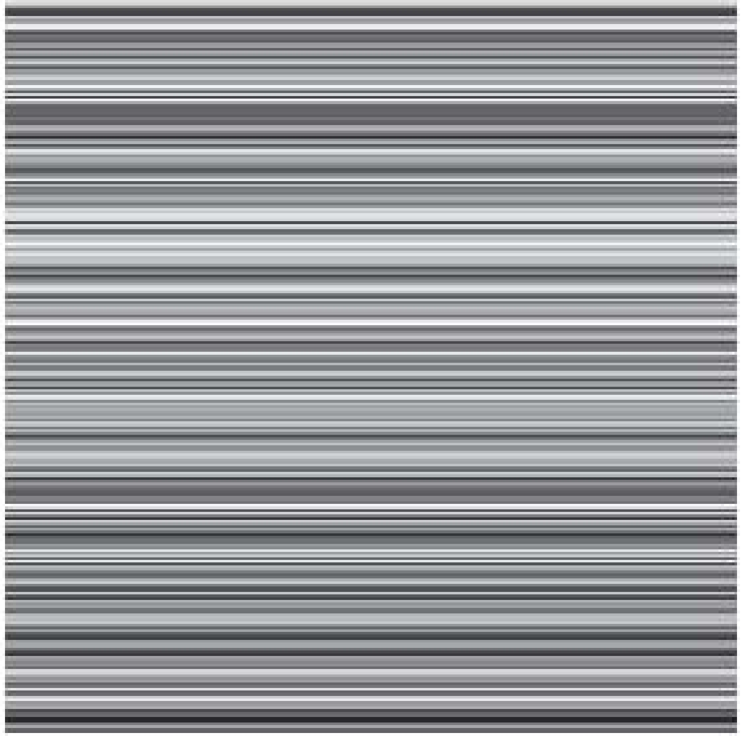
$$\tilde{H} = -\sum_{k=1}^{L-1} p_r(r_k) \log_2 p_r(r_k)$$

- Because the base 2 logarithm is used, above equation is the average information per intensity output of the imaginary intensity source in bits. It is not possible to code the *intensity values* of the imaginary source (and thus the sample image) with fewer than $\tilde{H}$ bits/ pixel.

Measuring Image Information



Entropy: 1.6614 bits/pixel



Entropy: 8 bits/pixel

Entropy: 1.566 bits/pixel

The amount of entropy, and thus information in an image, is far from intuitive.

Fidelity Criteria

- Compression may lead to loss information.
- Quantifying the nature and extent of information loss
    - Objective fidelity criteria
        - When the level of information loss can be expressed as a function of the input image, the compressed and output image.
        - Easy to operate (automatic).
        - Often requires the original copy as the reference.
    - Subjective fidelity criteria
        - Evaluated by human observers.
        - Do not require the original copy as a reference.
        - Most decompressed images ultimately are viewed by human.

Objective Fidelity Criteria

Root-Mean-Square (rms) Error:

- Let $f(x,y)$ represent an input image, and $\hat{f}(x,y)$ be an estimate or approximation of $f(x,y)$.

- For any value of x and y, the error $e(x, y)$ is given by:

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

- The root-mean-square error (e_{rms}) is

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

Objective Fidelity Criteria

Mean Square Signal-to-Noise Ratio:

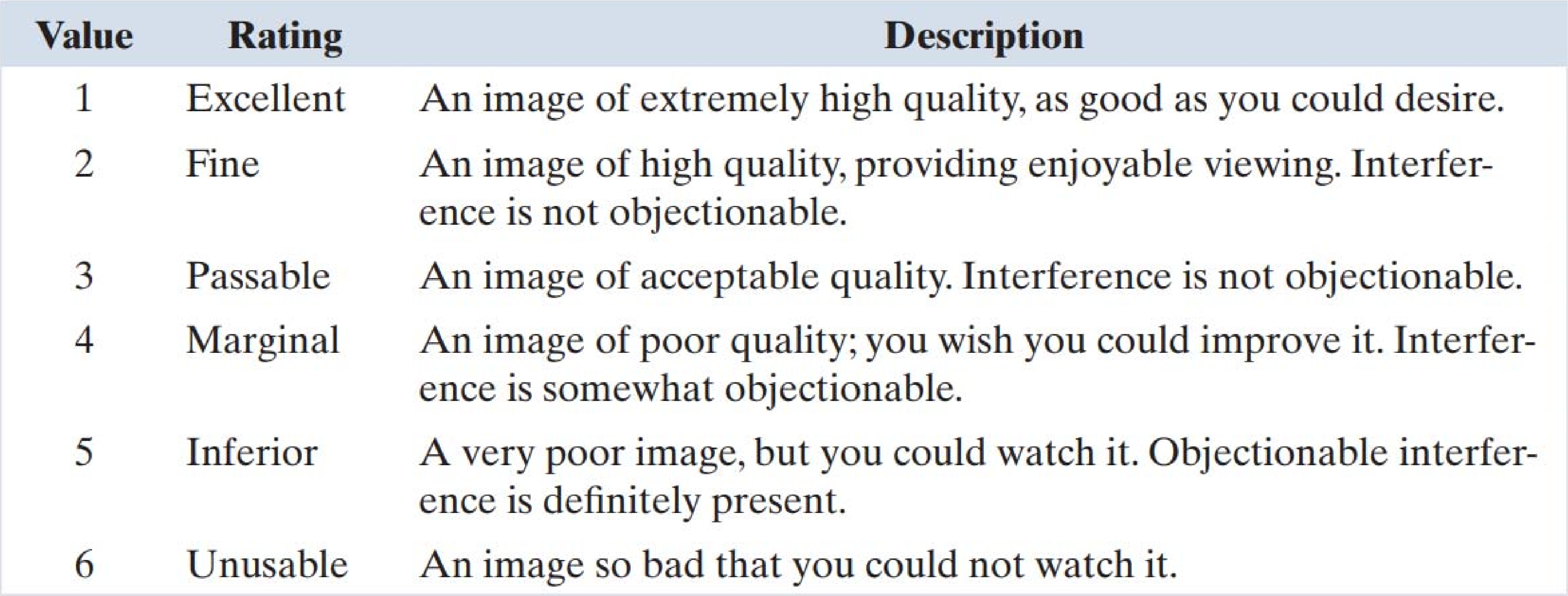
- The mean-square signal-to-noise ratios of the output image (SNR_{ms}) is:

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

Subjective Fidelity Criteria

- Most decompressed images are viewed by humans.
- Measuring image quality by the subjective evaluations is more appropriate.
- Example: evaluation or voting.

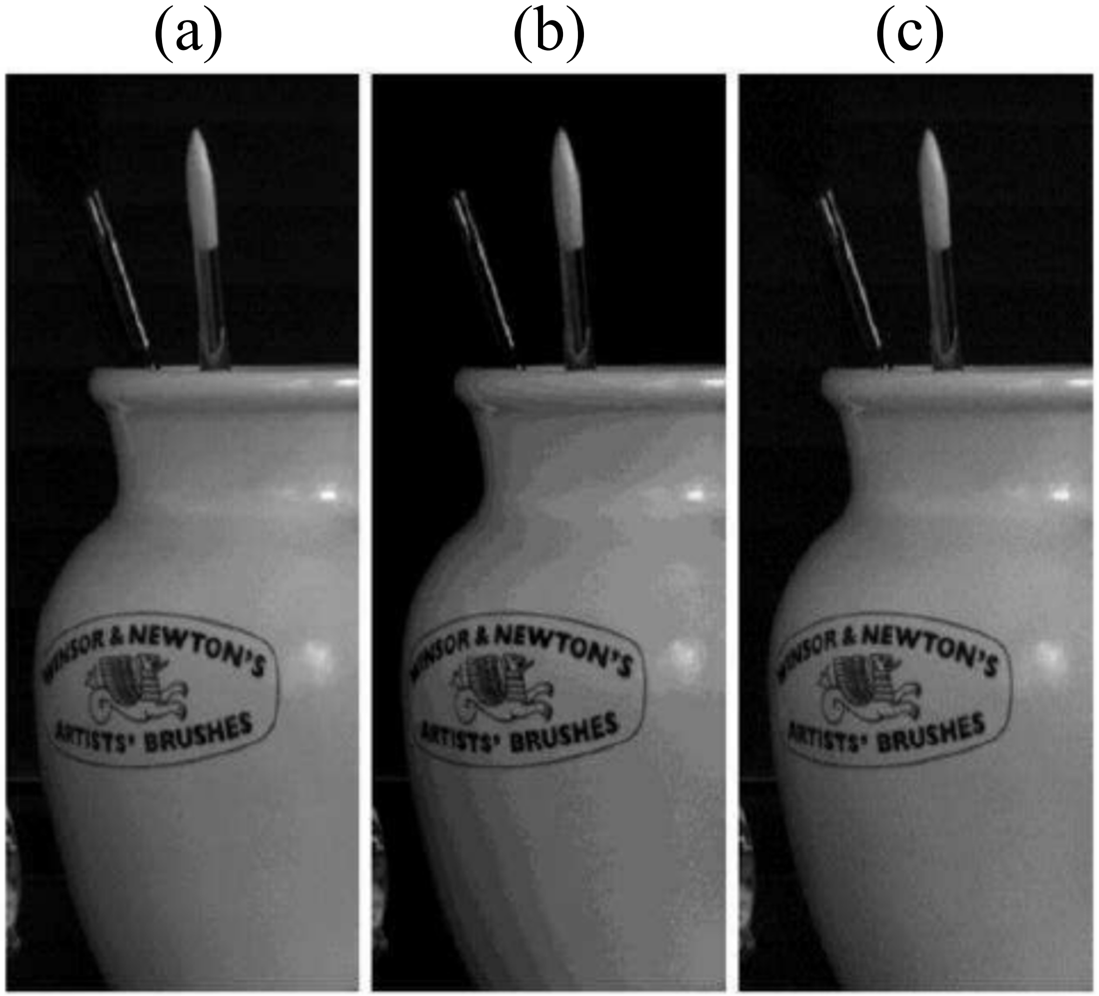
Subjective Fidelity Criteria



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.

Fidelity Criteria

- For example, the rms of (b) and (c) are 6.93 and 6.78.
- Based on objective fidelity, these values are quite similar.
- A subjective evaluation of the visual quality of the two coded images might: (b) marginal, and (c) passable.

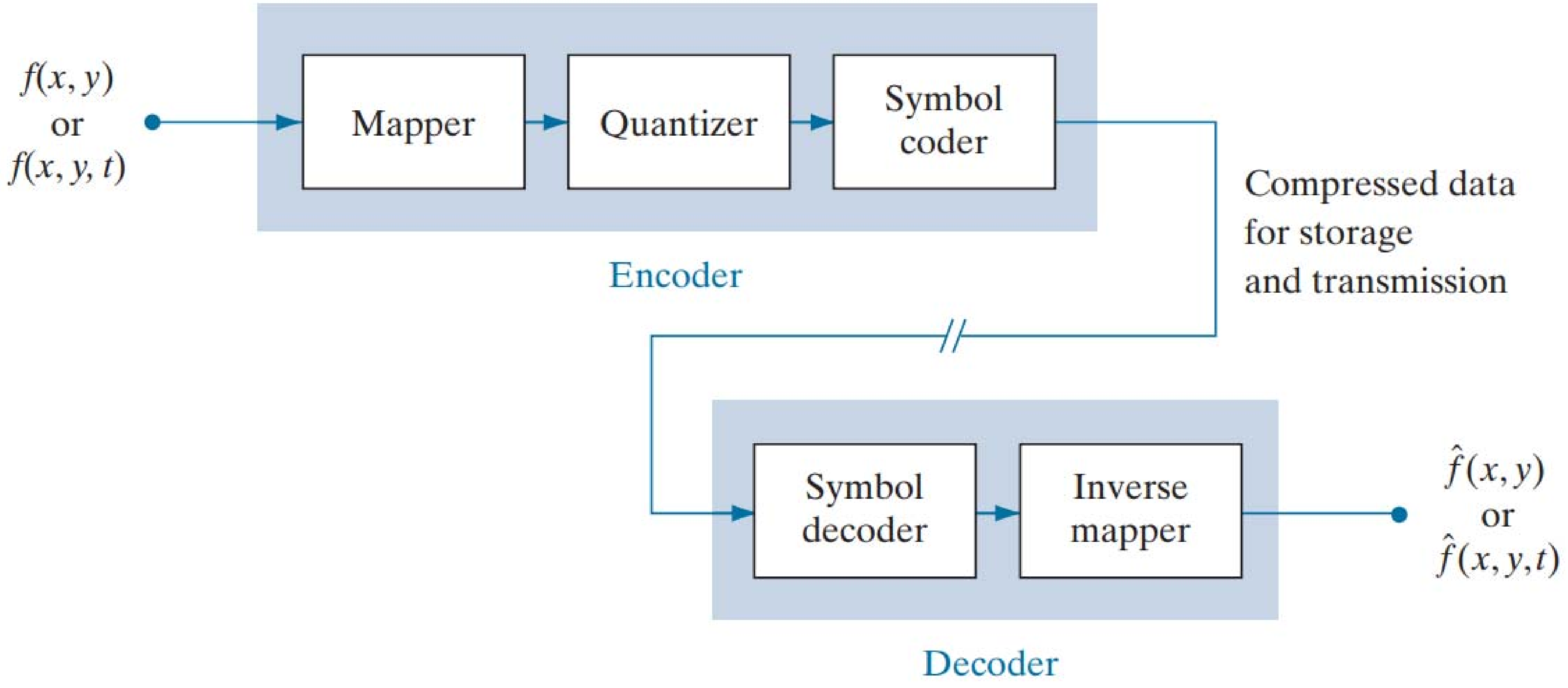


Compression Models

- A compression system consists of two distinct structural blocks: an *encoder* and a *decoder*.
- The encoder performs compression, and the decoder performs the complementary operation of decompression.
- A codec is a device or program that is capable of both encoding and decoding.

Compression Models

- Functional block diagram of a general image compression system.



Compression Models

- Each operation in the encoder is designed to reduce or eliminate one of the three redundancies.
  - Spatial and temporal redundancy (Mapper, reversible).
  - Irrelevant redundancy (Quantizer, irreversible).
  - Coding redundancy (Symbol Encoder, reversible).

Compression Models

Three steps for an encoder:

1. First, *mapper* transforms the input data into a format designed to reduce spatial-temporal redundancies in the input image (run-length coding). This operation generally is reversible.
2. Second, *quantizer* block reduces the accuracy of the mapper's output in accordance with some pre-established fidelity criterion. This stage reduces the irrelevant information of the input image. It is irreversible.
3. Third, *symbol encoder* creates a fixed or variable length code to represent the quantizer output. It can reduce coding redundancy, and it is reversible.

Compression Models

- The quantizer must be omitted when error-free compression is desired.
- Some compression techniques normally are modeled by merging blocks that are physically separate in above figure.
- The source decoder only contains two blocks: 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.

Image Formats and Compression Standards

- An *image file format* is a standard way to organize and store image data. It defines how the data is arranged and the type of compression (if any) that is used.
- An *image container* is similar to a file format, but handles multiple types of image data.
- Image *compression standards* define procedures for compressing and decompressing images—that is, for reducing the amount of data needed to represent an image.

Image Formats and Compression Standards

- Some popular image compression standards, file formats, and containers.
  - Internationally sanctioned entries are shown in blue; all others are in black.

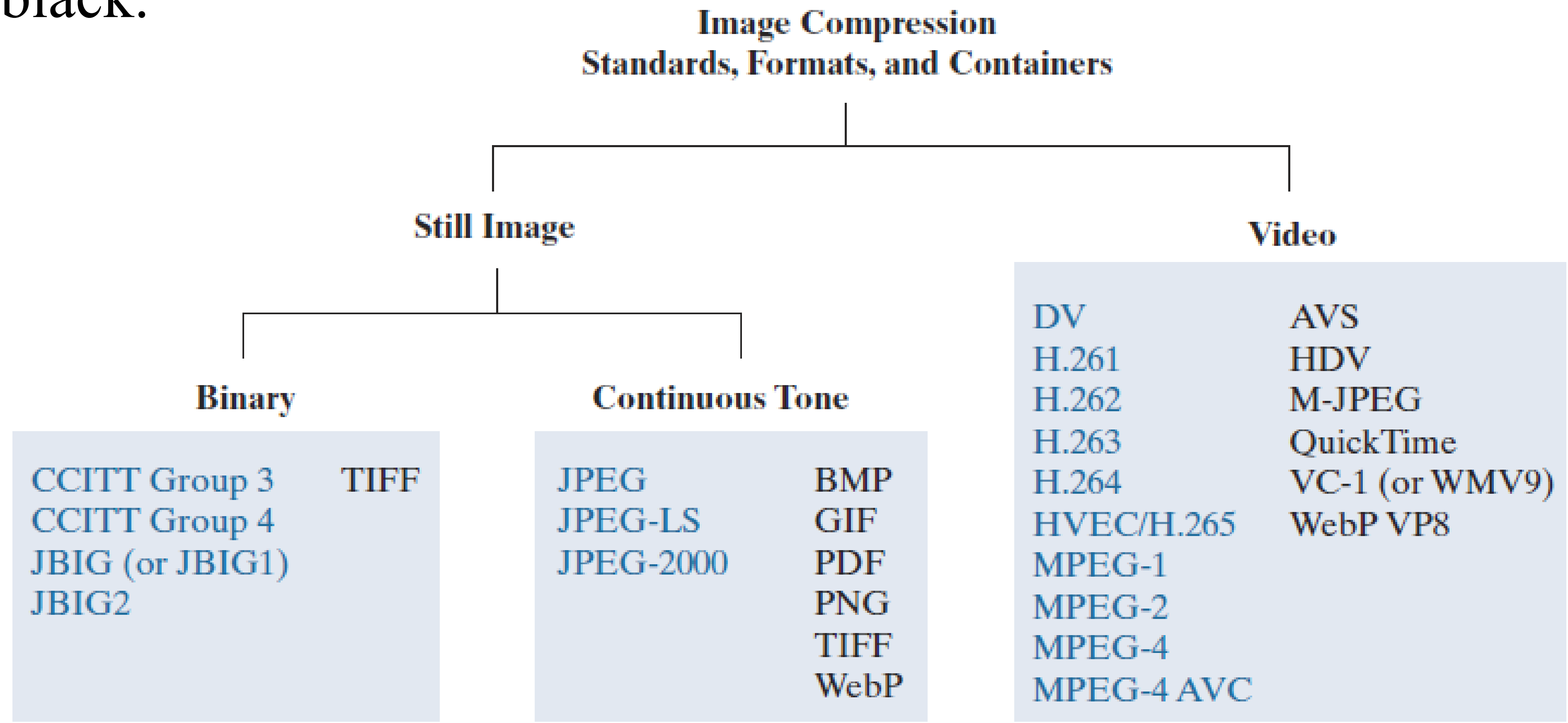


Image Compression
Standards, Formats, and Containers

Still Image

Binary

CCITT Group 3
CCITT Group 4
JBIG (or JBIG1)
JBIG2

TIFF

Continuous Tone

JPEG
JPEG-LS
JPEG-2000

BMP
GIF
PDF
PNG
TIFF
WebP

Video

DV
H.261
H.262
H.263
H.264
HVEC/H.265
MPEG-1
MPEG-2
MPEG-4
MPEG-4 AVC

AVS
HDV
M-JPEG
QuickTime
VC-1 (or WMV9)
WebP VP8

Lossless Image Compression

- Lossless Image Compression is also called Error-Free compression
- The need for error-free compression is motivated by the intended use or nature of the images.
- In some applications, it is the only acceptable means of data reduction.
  - Archival of medical or business documents, where lossy compression usually is prohibited for legal reasons.
  - Other is the processing of satellite images, where both the use and cost of collecting the data makes any loss undesirable.
  - Another is digital radiography, where the loss of information can compromise diagnostic accuracy.

Lossless Image Compression

- Two relatively independent operations
  - Eliminate coding redundancies.
  - Reduce spatial-temporal redundancies.
- They normally provide compression ratios of 2 to 10.
- Approaches:
  - Variable-length coding (coding redundancies)
    - Huffman coding
    - Arithmetic coding
  - LZW coding (spatial-temporal redundancies)

Variable-Length Coding

- Reducing coding redundancy: assign the shortest possible code words to the most probable gray levels.
    - Huffman Coding
    - Arithmetic Coding
- Remark: The source symbols may be either the gray levels of an image or the output of a gray-level mapping operation.

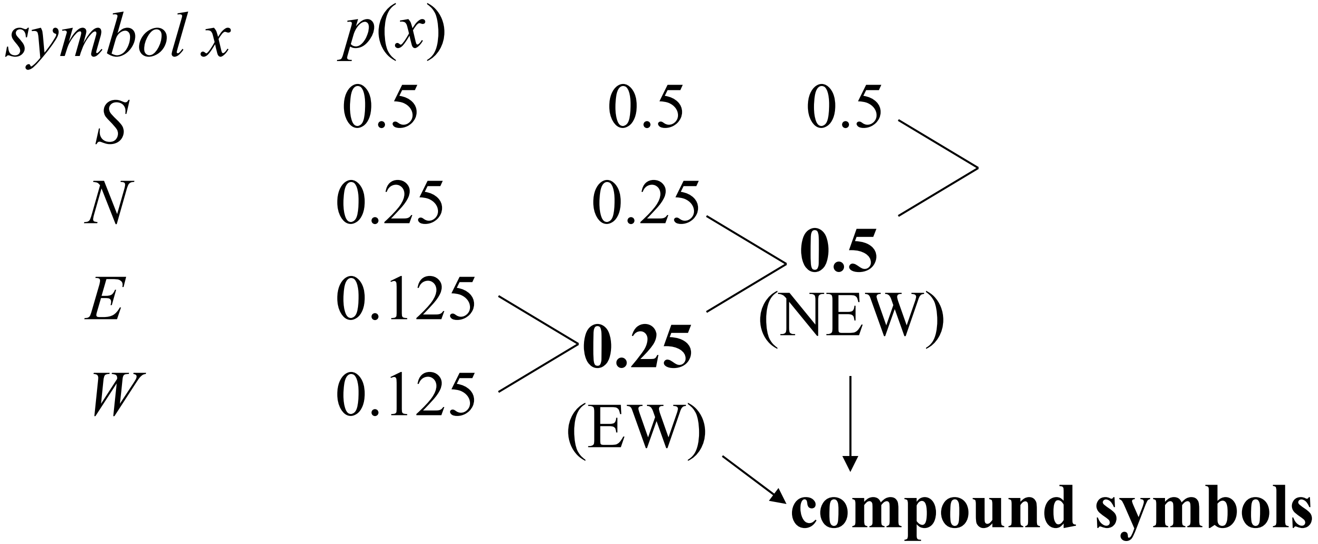
Huffman Coding

- Coding Procedures for an N-symbol source, two steps:
    - Source reduction
        - List all probabilities in a descending order.
        - Merge the two symbols with smallest probabilities into a new **compound symbol**.
        - Repeat the above two steps until a reduced source with two symbols is reached.
    - Codeword assignment
        - Start from the smallest source and work back to the original source.
        - Each merging point corresponds to a node in binary codeword tree.

Example I

For a string: “SENSSNSW”

Step 1: Source reduction



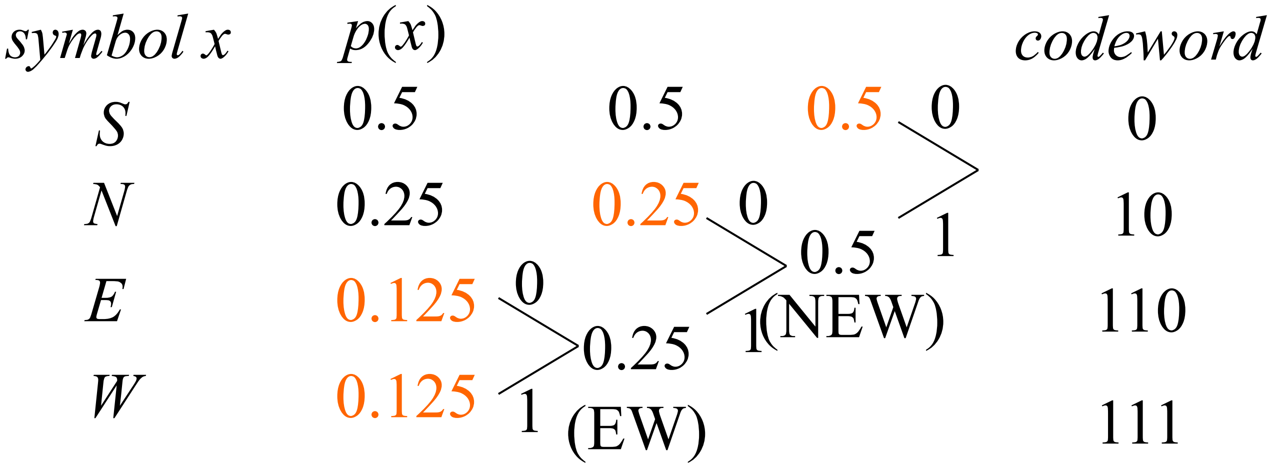
symbol x	*p*(*x*)		
S	0.5	0.5	0.5
N	0.25	0.25	**0.5** (NEW)
E	0.125	**0.25** (EW)	
W	0.125		

compound symbols

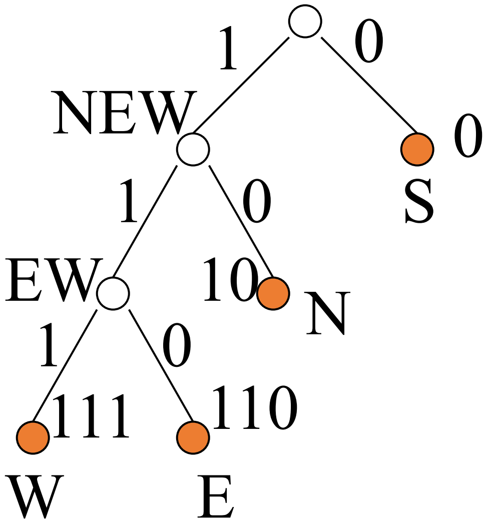
Example I

For a string: "SENSSNSW"

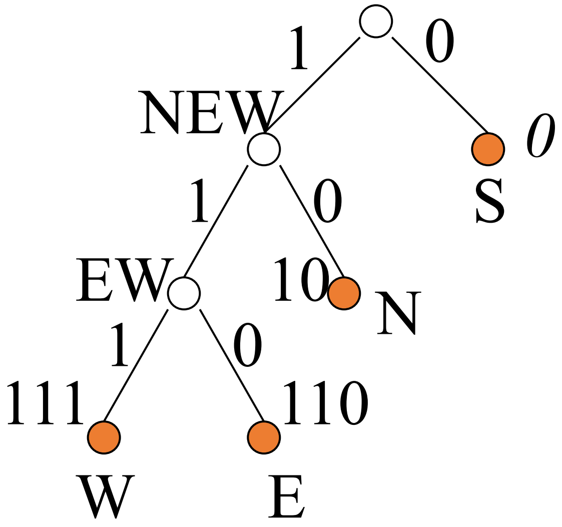
Step 2: Codeword assignment



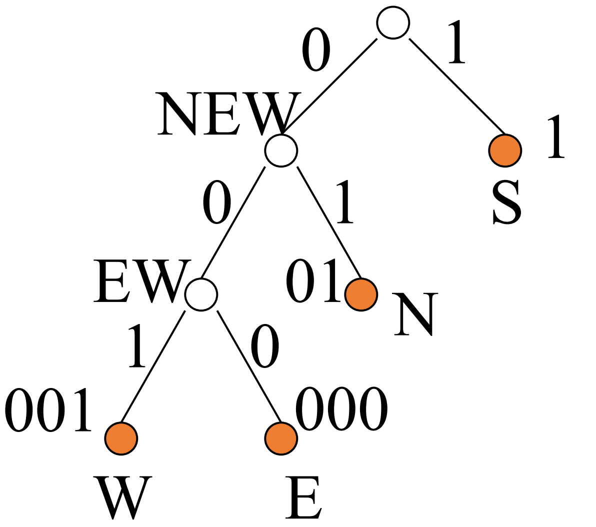
symbol x	*p*(*x*)			*codeword*
S	0.5	0.5	0.5 0	0
N	0.25	0.25 0	0.5 1 (NEW)	10
E	0.125 0	0.25 1 (EW)		110
W	0.125 1			111



Example I



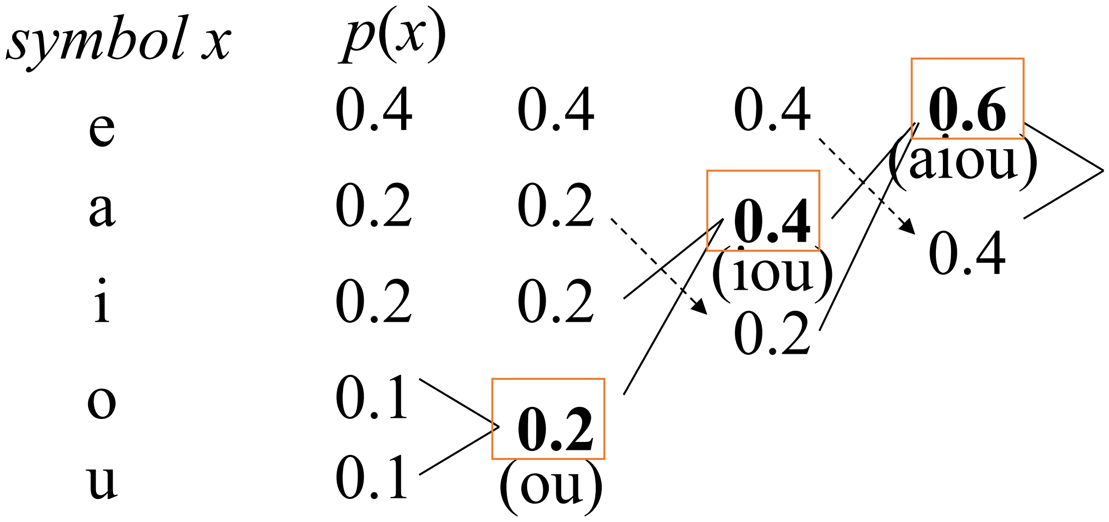
or



The codeword assignment is not unique. In fact, at each merging point (node), we can arbitrarily assign “0” and “1” to the two branches (average code length is the same).

Example II

Step 1: Source reduction

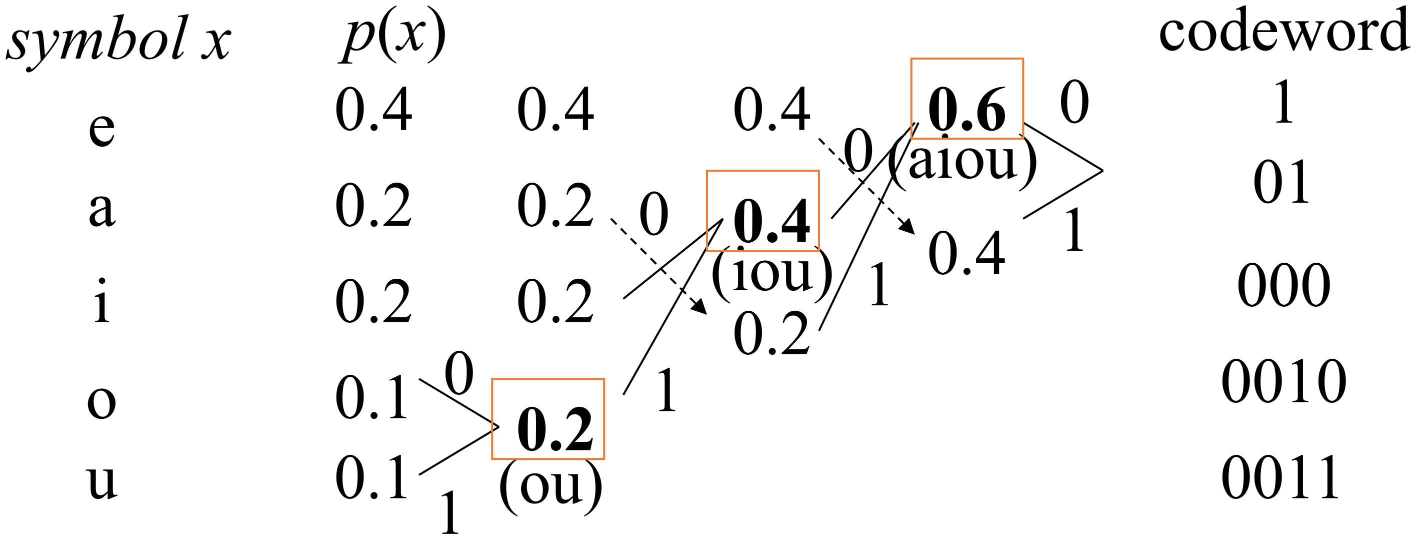


symbol x	$p(x)$			
e	0.4	0.4	0.4	**0.6** (aiou)
a	0.2	0.2	**0.4** (iou)	0.4
i	0.2	0.2	0.2	
o	0.1	**0.2** (ou)		
u	0.1			

□ **compound symbols**

Example II

Step 2: Codeword assignment



symbol x	*p*(*x*)	codeword
e	0.4	1
a	0.2	01
i	0.2	000
o	0.1	0010
u	0.1	0011

0.4 0.2 0.2 **0.2** (ou)

0.4 **0.4** (iou) 0.2

0.6 (aiou) 0.4

0 1 0 1 0 1 0 1

□ **compound symbols**

Example II

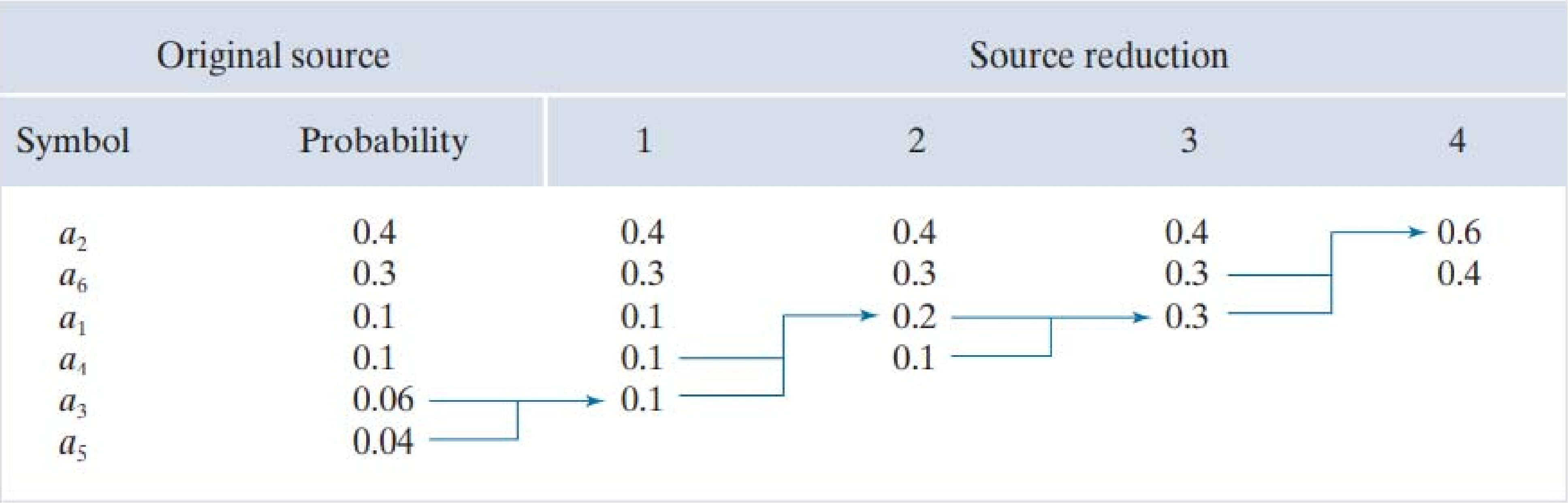
symbol x	$p(x)$	codeword	length
e	0.4	1	1
a	0.2	01	2
i	0.2	000	3
o	0.1	0010	4
u	0.1	0011	4

$$\bar{l} = \sum_{i=1}^{5} p_i l_i = 0.4\times1+0.2\times2+0.2\times3+0.1\times4+0.1\times4=2.2$$

If we use fixed-length codes, we have to spend three bits per sample, so the compression ratio is 3/2.2=1.364, and the code redundancy is 0.267.

Example III

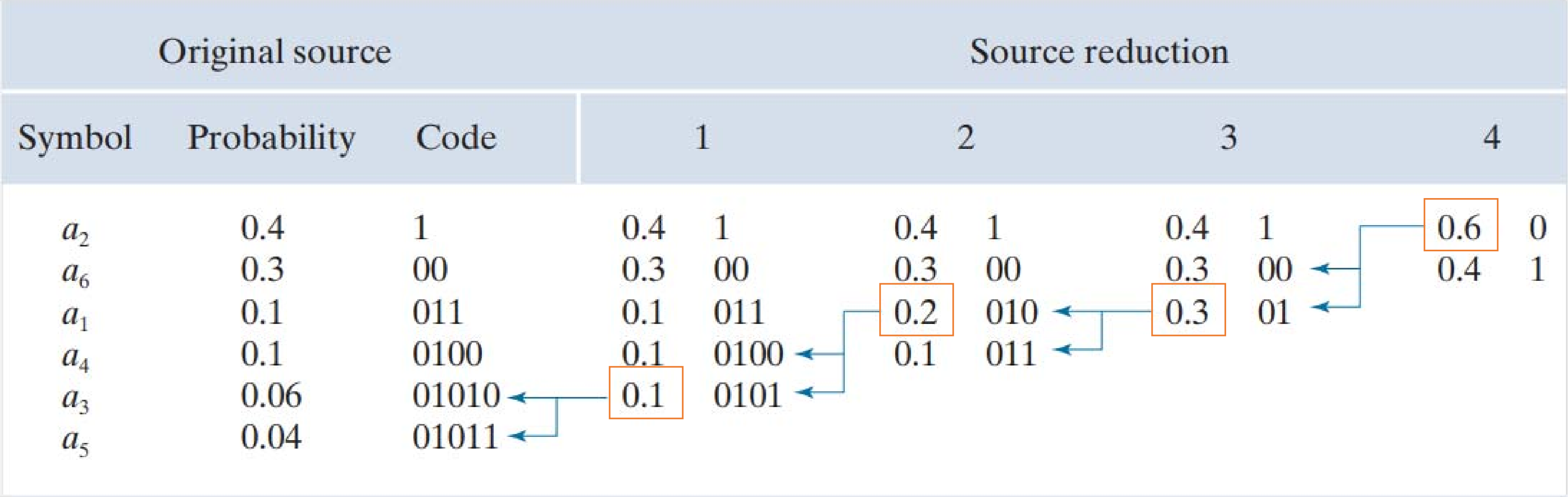
Step 1: Source reduction



Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Example III

Step 2: Codeword assignment

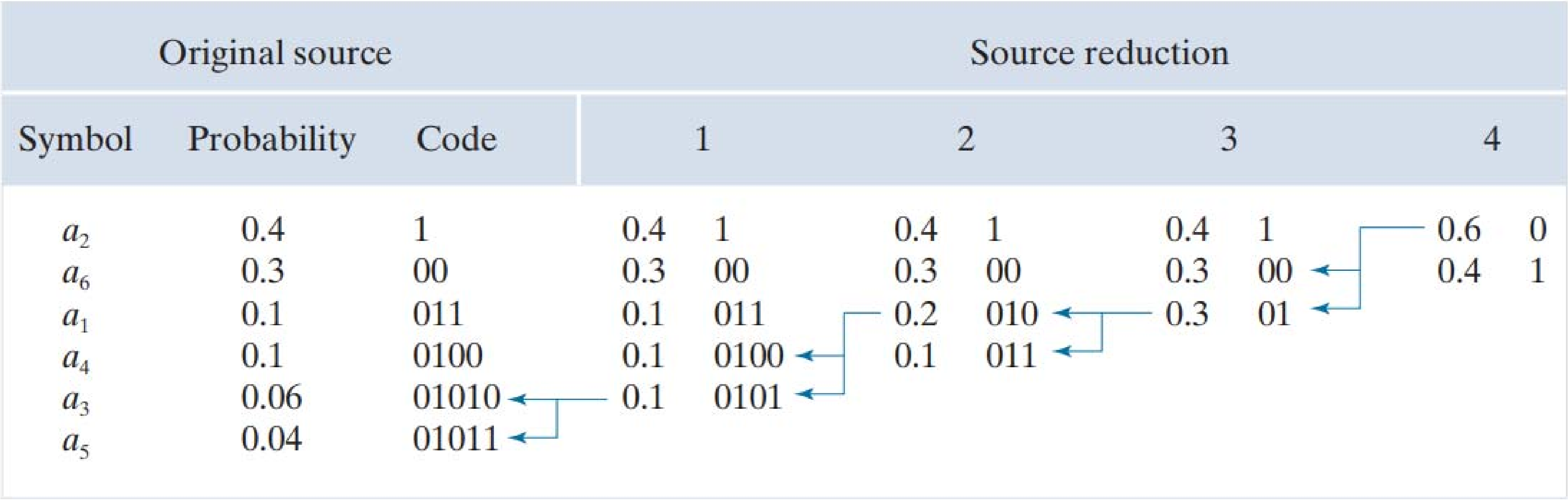


Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

The average length of the code is:

$$L_{avg} = 0.4\times1 + 0.3\times2 + 0.1\times3 + 0.1\times4 + 0.06\times5 + 0.04\times5$$
$$= 2.2 \text{ bits/symbol}$$

Huffman Coding



Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

- After the code has been created, coding and/or decoding is accomplished in a simple lookup table manner.
- Example: $\underline{01010}\ \underline{011}\ \underline{1}\ \underline{1}\ \underline{00}$; Answer: $a_3a_1a_2a_2a_6$

Summary

- In this lecture we have learnt:
  - Fundamentals
  - Data Redundancies
  - Image Compression Models
  - Lossless Image Compression
    - Huffman Coding

Optional Homework

Check the Textbook!

- **Chapter 8: Problems 8.1, 8.3, 8.6, 8.9**

- Homework answers will be provided at the end of each week.