Fundamentals of Multimedia

Lecture 5
Lossless Data Compression
Variable Length Coding

Mahmoud El-Gayyar
elgayyar@ci.suez.edu.eg

Outcomes of Lecture 4

- *Data Compression*
  - Compression and compression ratio
- *Bound of lossless compression*
  - Data entropy
- *Fixed Length Coding*
  - Run Length Coding (RLC)
  - Dictionary-based Coding
    - *Lempel-Ziv-Welch (LZW) algorithm*

Outline

- *VariableLength Coding*
  - Shannon-Fano Algorithm
  - Huffman Coding Algorithm
- *Lossless Compression in JPEG images.*
  - Differential Coding
  - Lossless JPEG

Outline

- ***VariableLength Coding***
  - Shannon-Fano Algorithm
  - Huffman Coding Algorithm
- *Lossless Compression in JPEG images.*
  - Differential Coding
  - Lossless JPEG

Variable Length Coding (VLC)

- *The length of the codeword is variable depends on the probabilities of possible values in the alphabet (S)*
- *Symbols that occurs more frequently will have shorter codes.*

Shannon-Fano Algorithm

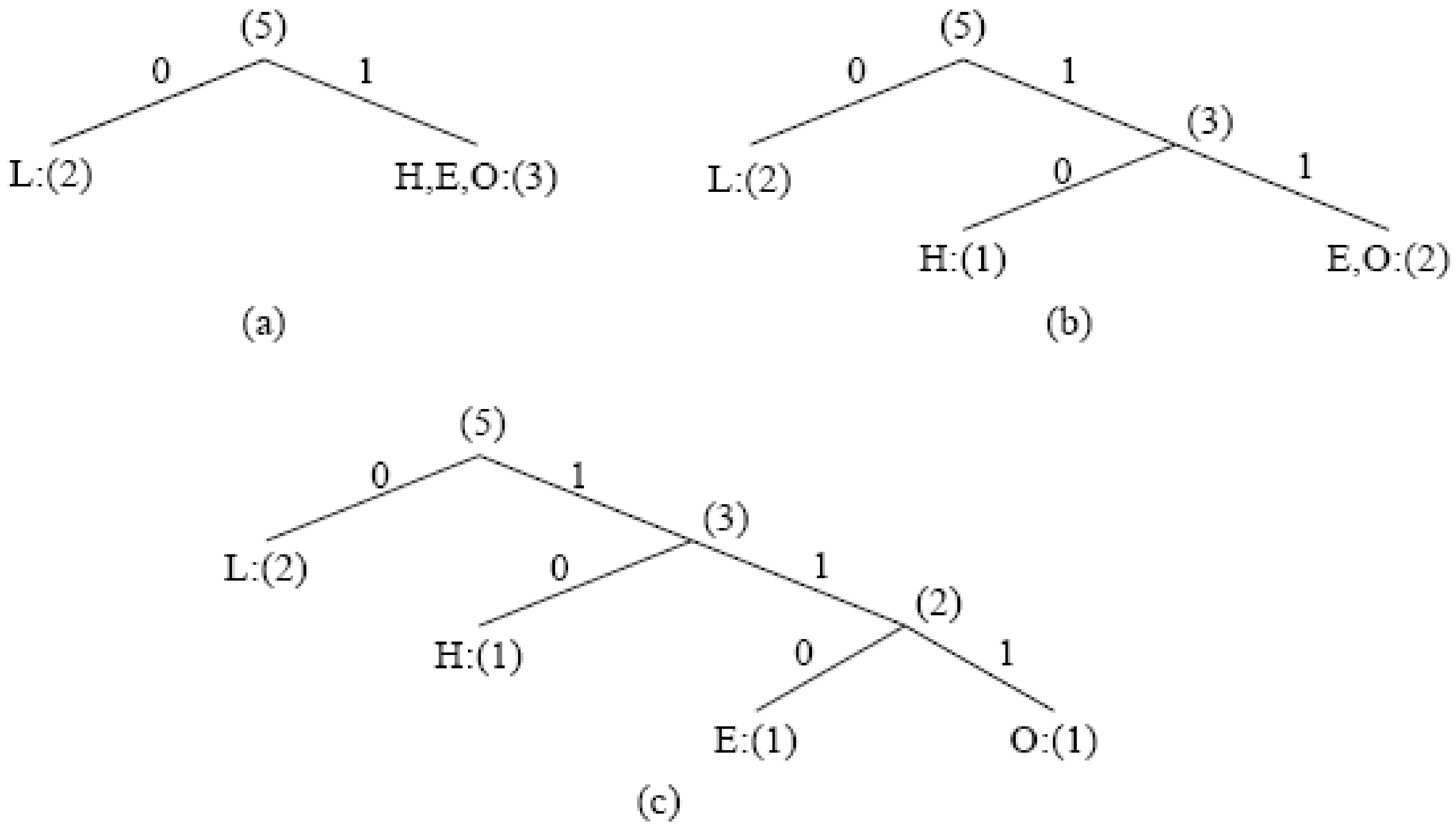
- *Example:*
  - Hello

Symbol	H	E	L	O
Count	1	1	2	1

- *Algorithm: Top-down approach*
  - Sort the symbols according to the frequency count of their occurrences.
  - Recursively divide the symbols into two parts, each with approximately the same number of counts, until all parts contain only one symbol.

Example: Shannon-Fano.

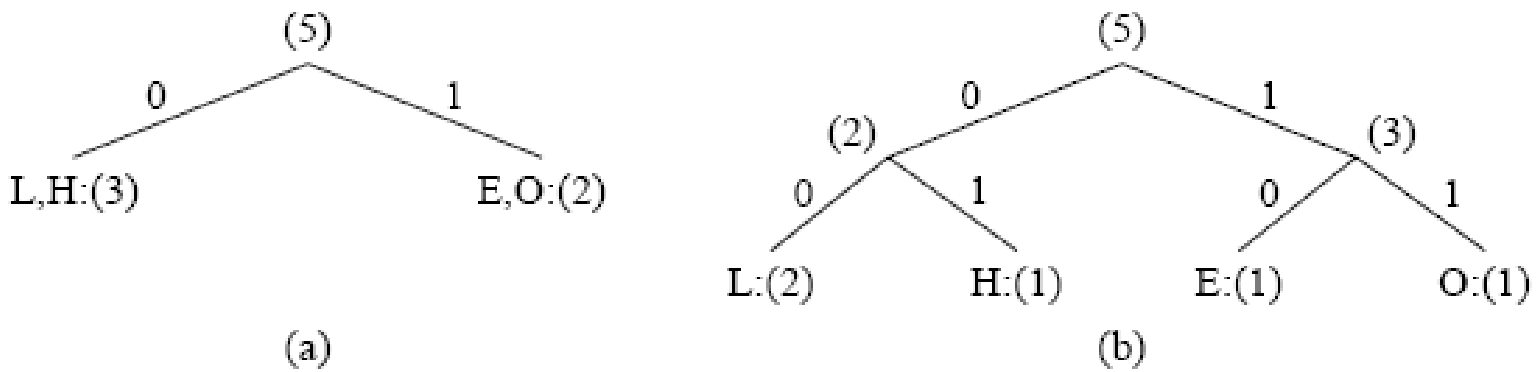
Symbol	L	H	E	O
Count	2	1	1	1



Example: Shannon-Fano.

Symbol	Count	$Log_2 1/p_i$	Code	# of bits
L	2	1.32	0	2
H	1	2.32	10	2
E	1	2.32	110	3
O	1	2.32	111	3
Total number of bits				10

Example: Another Coding- Shannon-Fano.



Symbol	Count	$Log_2 1/p_i$	Code	# of bits
L	2	1.32	00	4
H	1	2.32	01	2
E	1	2.32	10	2
O	1	2.32	11	2
Total number of bits				10

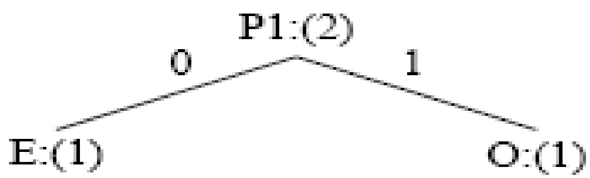
Huffman Coding Algorithm

- *Algorithm: Bottom-up approach*
  - ***(1) Initialization***: Put all symbols on a list sorted according to their frequency counts.
  - (2) Repeat until the list has only one symbol left:
    - *2.1 From the list pick two symbols with the lowest frequency counts. Form a Huffman sub-tree that has these two symbols as child nodes and create a parent node.*
    - *2.2 Assign the sum of the children's frequency counts to the parent and insert it into the list such that the order is maintained.*
    - *2.3 Delete the children from the list.*
  - 3. Assign a codeword for each leaf based on the path from the root.

Example: Huffman Coding

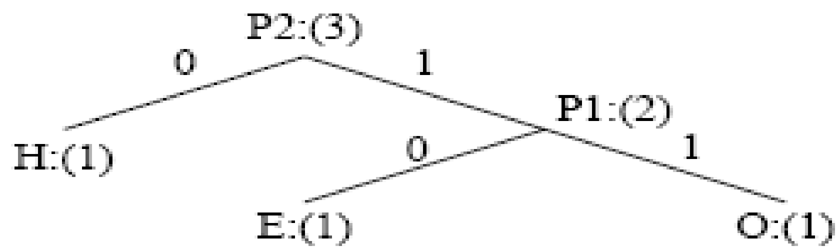
Symbol	L	H	E	O
Count	2	1	1	1

After iteration1: L P1 H



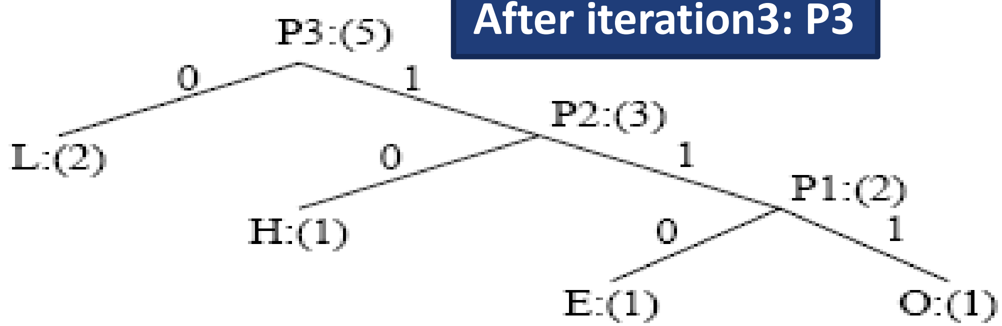
(a)

After iteration2: P2 L



(b)

After iteration3: P3



(c)

Symbol	L	H	E	O
Code	0	10	110	111

Properties Huffman Coding

- ***Unique Prefix Property:*** *No Huffman code is a prefix of any other Huffman code - prevents any ambiguity in decoding.*
  - Applied also to Shannon-Fano
- ***Optimality:*** *minimum redundancy code.*
  - The two least frequent symbols will have the same length for their Huffman codes, differing only at the last bit.
  - Symbols that occur more frequently will have shorter Huffman codes than symbols that occur less frequently.
  - The average code length for an information source S is strictly less than $\eta + 1$.

Shannon-Fano vs. Huffman Coding

- *Example: In a message, the codes and their frequencies are A(15), B(7), C(6), D(6), E(5). Encode this message with Shannon-fano and Huffman coding.*
- *Try yourself!*

- *Shannon-fano: 89 bits*
- *Huffman : 87 bits*

Fixed vs. Variable Length Coding

- *Data sequence*
  - {AAAAAAAAAAAAAAABBBBBBBCDCDCDCDCDCDEEEEE}
  - A text string containing 39 characters and their frequency counts as follows: A:(15), B:(7), C:(6), D:(6) and E:(5)
- *Compare with fix-length coding*

Probability	Huffman	Fix-length
A: 15/39	0	000
B: 7/39	100	001
C: 6/39	101	010
D: 6/39	110	011
E: 5/39	111	100
Total:	87 bits	117 bits

Outline

- ***VariableLength Coding***
  - Shannon-Fano Algorithm
  - Huffman Coding Algorithm
- ***Lossless Compression in JPEG images.***
  - Differential Coding
  - Lossless JPEG

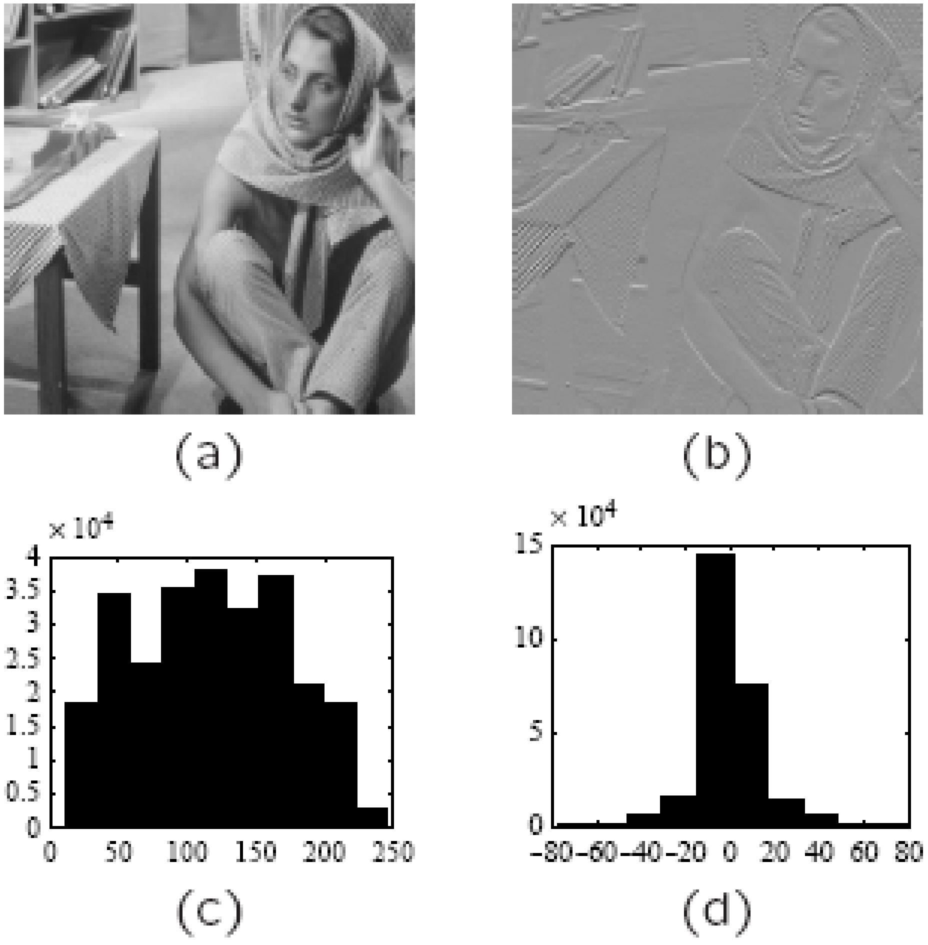
Differential Coding

- *Given an original image I(x, y), using a simple difference operator we can define a difference image d(x, y) as follows:*

$$d(x; y) = I(x; y) - I(x - 1; y)$$

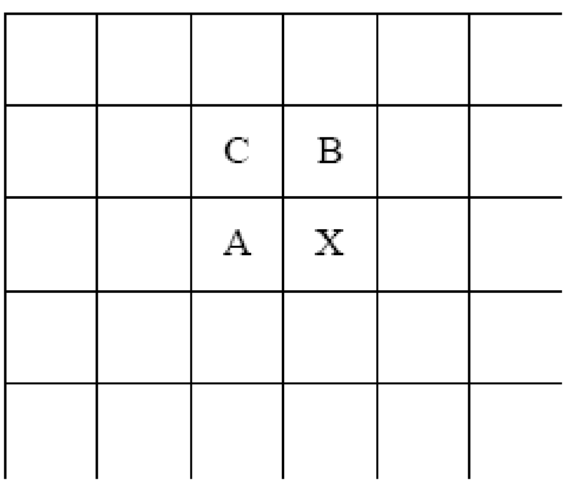
- *Due to spatial redundancy existed in normal images I, the difference image d will have a narrower histogram and hence a smaller entropy.*

Differential Coding



Lossless JPEG

- *A special case of the JPEG image compression.*
- *The Predictive method*
  - (1) Forming **a differential prediction**: A predictor combines the values of up to three neighboring pixels as the predicted value for the current pixel. The predictor can use any one of the seven schemes.



Predictor	Prediction
P1	A
P2	B
P3	C
P4	A + B - C
P5	A + (B - C) / 2
P6	B + (A - C) / 2
P7	(A + B) / 2

Lossless JPEG

- *The Predictive method*
  - **(2) Encoding:** The encoder compares the prediction with the actual pixel value at the position `X' and encodes the difference using one of the lossless compression techniques we have discussed, e.g., the Huffman coding scheme.
- *Since prediction must be based on previously encoded neighbors,*
  - first pixel in the image I(0, 0) → use its own value.
  - The pixels in the first row always use predictor P1, and those in the first column always use P2.

Other Lossless Compression

Compression Program	Compression Ratio			
	Lena	football	F-18	flowers
Lossless JPEG	1.45	1.54	2.29	1.26
Optimal lossless JPEG	1.49	1.67	2.71	1.33
compress (LZW)	0.86	1.24	2.21	0.87
gzip (LZ77)	1.08	1.36	3.10	1.05
gzip -9 (optimal LZ77)	1.08	1.36	3.13	1.05
pack (Huffman coding)	1.02	1.12	1.19	1.00

Lossless compression usually yields a relatively low compression ratio, which renders it impractical for most multimedia applications.

Summary

- *VariableLength Coding*
  - Shannon-Fano Algorithm
  - Huffman Coding Algorithm
- *Lossless Compression in JPEG images.*
  - Differential Coding
  - Lossless JPEG