8.1.6 Image compression models (page 558)

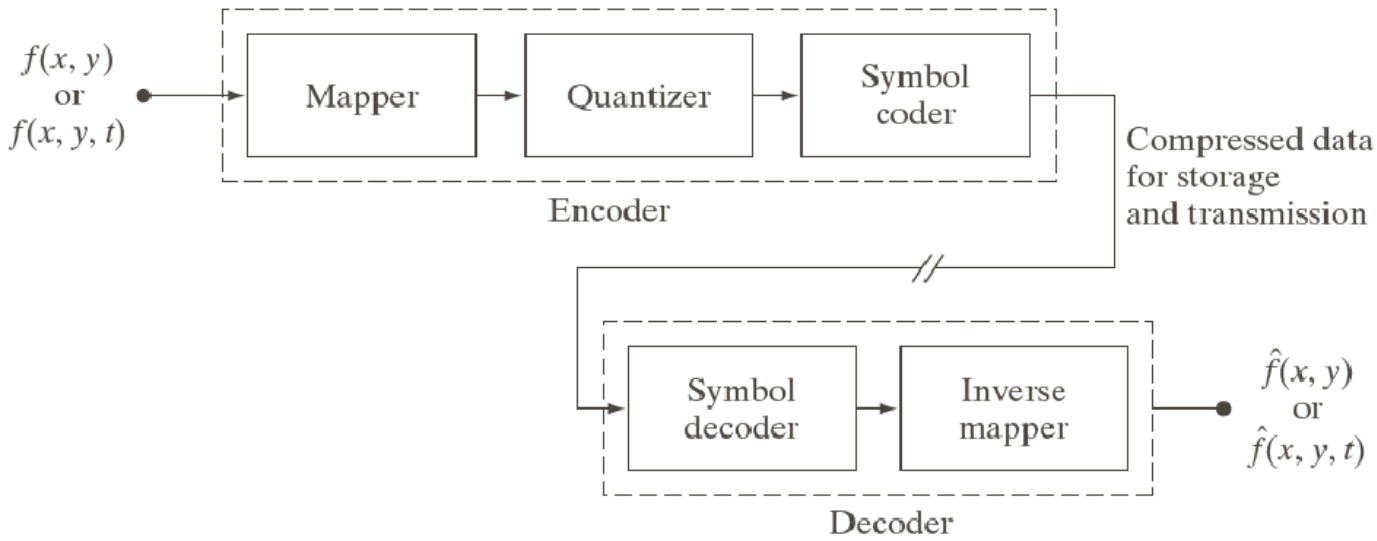

FIGURE 8.5 Functional block diagram of a general image compression system.

- **Mapper: Reduces spatial and temporal redundancy, e.g. run-length coding and calculation of DCT (reversible)**
- **Quantizer: Removes irrelevant information, e.g. reduction of number of grey scales and removal of high frequency content (irreversible)**
- **Symbol coder: Reduces coding redundancy, e.g. generates fixed- or variable-length code (reversible)**

8.1.7 Image formats, containers and compression standards (READ)

8.2 Some basic compression models

8.2.1 Huffman coding

- Most probable symbol is assigned the shortest code word
- Variable-length coding
- Used in, for example: CCITT, JBIG2, JPEG, MPEG-1,2,4, H.261, H.262, H.263, H.264

$$\underbrace{\text{Instantaneous}}$$
do not reference succeeding symbols

$$\underbrace{\text{uniquely decodable}}$$
symbols can be decoded in only one way

$$\underbrace{\text{block code}}$$
for each symbol: only one code

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				

FIGURE 8.7 Huffman source reductions.

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								

FIGURE 8.8 Huffman code assignment procedure.

Example

Decode: 010100111100

8.2.2 Golomb coding (READ)

8.2.3 Arithmetic coding • Not a block code • Entire sequence of symbols assigned a single arithmetic code

Example: Code $a_1\ a_2\ a_3\ a_3\ a_4$, where a_4 is EOB indicator

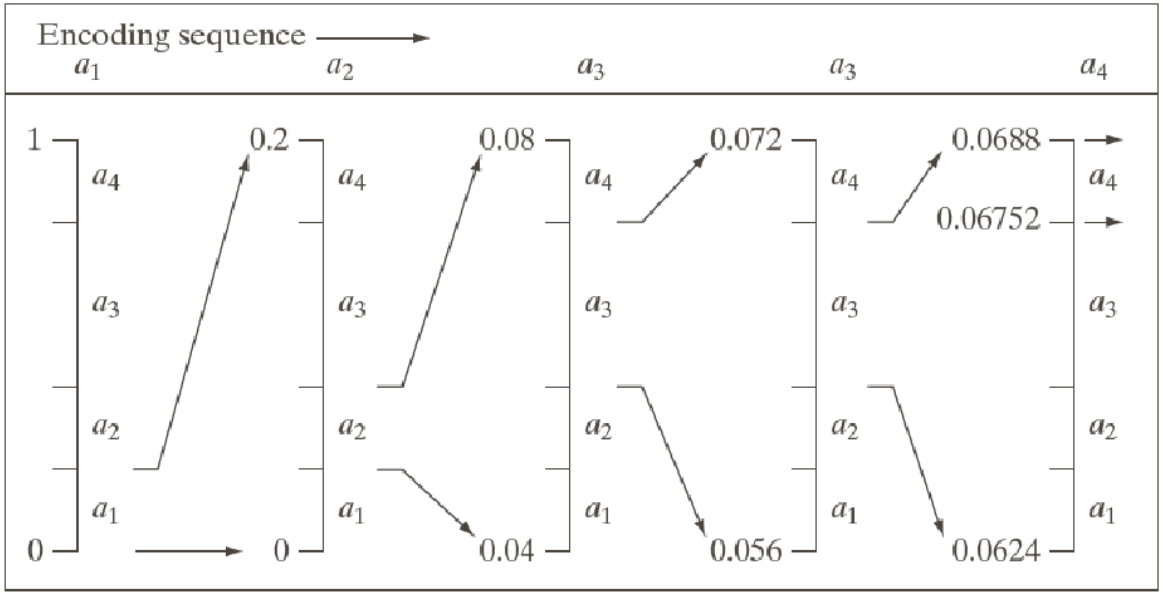

FIGURE 8.12 Arithmetic coding procedure.

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

TABLE 8.6 Arithmetic coding example.

8.2.4 LZW coding

- Also attack inter-pixel redundancies
- Assigns fixed-length code words to variable length sequences of source symbols
- Requires no a priori knowledge of probability of occurrence of symbols
- Integrated into GIF, TIFF and PDF file formats

8.2.5 and 8.2.7: Run-length and bit-plane coding 8.2.6 (READ)

Bit-plane coding

- Process the image's bit planes individually
- Decompose the image into a series of binary images
- Compress each binary image via a binary compression method

One-dimensional run-length coding

- Represent each row of the image or bit plane by a sequence of lengths that describes successive runs of black and white pixels

• **Standard compression approach in FAX coding (TIFF files):**

One-dimensional run-length coding: CCITT Group 3 standard
Two-dimensional run-length coding: CCITT Group 4 standard

• **Code each contiguous group of 0's or 1's encountered in a left-to-right scan of each row by its length**

(1) **Specify the value of first run of each row**
(2) **Assume that each row begins with a white run, whose run length may be zero**

• **Additional compression by variable length coding the run lengths themselves**

• **Black and white run lengths coded seperately tailored to their own statistics**

Two-dimensional run-length coding

• **Relative address coding (RAC)**

• **Not discussed**

CCITT Group 3 standard: 1D run-length coding (table A.1, p 933)

Run Length	White Code Word	Black Code Word	Run Length	White Code Word	Black Code Word
0	00110101	0000110111	32	00011011	000001101010
1	000111	010	33	00010010	000001101011
2	0111	11	34	00010011	000011010010
3	1000	10	35	00010100	000011010011
4	1011	011	36	00010101	000011010100
5	1100	0011	37	00010110	000011010101
6	1110	0010	38	00010111	000011010110
7	1111	00011	39	00101000	000011010111
8	10011	000101	40	00101001	000001101100
9	10100	000100	41	00101010	000001101101
10	00111	0000100	42	00101011	000011011010
11	01000	0000101	43	00101100	000011011011
12	001000	0000111	44	00101101	000001010100
13	000011	00000100	45	00000100	000001010101
14	110100	00000111	46	00000101	000001010110
15	110101	000011000	47	00001010	000001010111
16	101010	0000010111	48	00001011	000001100100
17	101011	0000011000	49	01010010	000001100101
18	0100111	0000001000	50	01010011	000001010010
19	0001100	00001100111	51	01010100	000001010011
20	0001000	00001101000	52	01010101	000000100100
21	0010111	00001101100	53	00100100	000000110111
22	0000011	00000110111	54	00100101	000000111000
23	0000100	00000101000	55	01011000	000000100111
24	0101000	00000010111	56	01011001	000000101000
25	0101011	00000011000	57	01011010	000001011000
26	0010011	000011001010	58	01011011	000001011001
27	0100100	000011001011	59	01001010	000000101011
28	0011000	000011001100	60	01001011	000000101100
29	00000010	000011001101	61	00110010	000001011010
30	00000011	000001101000	62	00110011	000001100110
31	00011010	000001101001	63	00110100	000001100111

CCITT terminating codes

CCITT Group 3 standard: 1D run-length coding... (table A.2, p 934)

CCITT makeup codes

Run Length	White Code Word	Black Code Word	Run Length	White Code Word	Black Code Word
64	11011	0000001111	960	011010100	0000001110011
128	10010	000011001000	1024	011010101	0000001110100
192	010111	000011001001	1088	011010110	0000001110101
256	0110111	000001011011	1152	011010111	0000001110110
320	00110110	000000110011	1216	011011000	0000001110111
384	00110111	000000110100	1280	011011001	0000001010010
448	01100100	000000110101	1344	011011010	0000001010011
512	01100101	0000001101100	1408	011011011	0000001010100
576	01101000	0000001101101	1472	010011000	0000001010101
640	01100111	0000001001010	1536	010011001	0000001011010
704	011001100	0000001001011	1600	010011010	0000001011011
768	011001101	0000001001100	1664	011000	0000001100100
832	011010010	0000001001101	1728	010011011	0000001100101
896	011010011	0000001110010			

Code Word		Code Word	
1792	00000001000	2240	000000010110
1856	00000001100	2304	000000010111
1920	00000001101	2368	000000011100
1984	000000010010	2432	000000011101
2048	000000010011	2496	000000011110
2112	000000010100	2560	000000011111
2176	000000010101		

- **If run length** < 64**: terminating code (Huffman type)**
- **If run length** ≥ 64**: makeup code** $+$ **terminating code**
- **EOL-code:** 000000000001
- **Note that (generally): white code** $<$ **black code**
- **Note that (generally): small run lengths more probable**