University of Tennessee

Subject: Digital Image Processing (ECE 472)

Project 6

Author: Haoling Jia

Lecturer: Dr. Hairong Qi

11/20/2011

100+25

Project 6 Image Compression

Task 1: Huffman Coding

Derive Huffman code using paper and pencil. For each of the examples given, answer the following questions:

a) (12/9) What's the entropy of the original data source (use 1st order estimate)?

$H(Ex1) = -[0.15*\log_2 0.15+0.11*\log_2 0.11+0.19*\log_2 0.19+0.15*\log_2 0.15+ 0.08*\log_2 0.08+0.02*\log_2 0.02+0.2*\log_2 0.2+0.1*\log_2 0.1] = 2.83$ (bits/pixel)

$H(Ex2) = -[0.32*\log_2 0.32+0.16*\log_2 0.16+0.09*\log_2 0.09+0.05*\log_2 0.05+ 0.14*\log_2 0.14+0.15*\log_2 0.15+0.01*\log_2 0.01+0.08*\log_2 0.08] = 2.64$ (bits/pixel)

$H(Ex3) = -[0.125*\log_2 0.125+0.123*\log_2 0.123+0.123*\log_2 0.125+0.125*\log_2 0.125+ 0.125*\log_2 0.125+0.125*\log_2 0.125+0.125*\log_2 0.125+ 0.125*\log_2 0.125]=3$(bits/pixel)

b) (3/1) What's the average coding length if using fixed-length coding?

To represent the eight values in the table, the length of coding should be at least 3. And the average coding length is also 3 for the fixed-length of coding.

c) (20/20) What's the average coding length of Huffman coding? (Need to show details of Huffman coding)

The Huffman coding of example 1, 2, 3 is shown in table 1, table2, table3, respectively.

Table 1 Huffman coding of Ex1

Huffman coding for example 1

Pixel value	Probability	Step1	Step2	Step3	Step4	Step5	Step6
R6	0.2(00)	0.2(00)	0.2(00)	0.2(00)	0.3(10)	0.39(0)	0.39(0)
R2	0.19(01)	0.19(01)	0.19(01)	0.19(01)	0.31(11)	0.3(10)	0.61(1)
R0	0.15(100)	0.15(100)	0.15(100)	0.15(100)	0.2(00)	0.31(11)	
R3	0.15(110)	0.15(110)	0.15(110)	0.15(110)	0.19(01)		
R1	0.11(111)	0.11(111)	0.20(110)	0.31(11)			
R7	0.1(1100)	0.1(1100)	0.11(111)				
R4	0.08(11110)	0.1(1111)					
R5	0.02(11111)						

The average coding length of example 1 of Huffman coding is:

$L_{avg}=2*0.2+2*0.19+3*0.15+3*0.15+3*0.11+4*0.1+5*0.08+5*0.02= 2.91$ (bits/pixel)

Table 2 Huffman coding of Ex2

Huffman coding for example 2

Pixel value	Probability	Step1	Step2	Step3	Step4	Step5	Step6
R0	0.32(11)	0.32(11)	0.32(11)	0.32(11)	0.32(11)	0.39(0)	0.39(0)
R1	0.16(00)	0.16(00)	0.16(00)	0.16(00)	0.29(10)	0.29(10)	0.61(1)
R5	0.15(100)	0.15(100)	0.15(100)	0.23(01)	0.16(00)	0.32(11)	
R4	0.14(101)	0.14(101)	0.14(101)	0.15(100)	0.23(01)		
R2	0.09(011)	0.09(011)	0.14(010)	0.14(101)			
R7	0.08(0100)	0.08(0100)	0.09(011)				
R3	0.05(01010)	0.06(0101)					
R6	0.01(01011)						

The average coding length of example 2 of Huffman coding is:

L_{avg}=2*0.32+2*0.16+3*0.15+3*0.14+3*0.09+4*0.08+5*0.05+5*0.01

= 2.72 (bits/pixel)

Table 3 Huffman coding of Ex3

Huffman coding for example 3

Pixel value	Probability	Step1	Step2	Step3	Step4	Step5	Step6
R0	0.125(000)	0.125(000)	0.125(000)	0.25(01)	0.25(00)	0.5(1)	0.5(0)
R1	0.125(001)	0.125(001)	0.125(001)	0.25(10)	0.25(01)	0.25(00)	0.5(1)
R5	0.125(010)	0.125(010)	0.25(10)	0.25(11)	0.25(10)	0.25(01)	
R4	0.125(011)	0.125(011)	0.25(11)	0.125(000)	0.25(11)		
R2	0.125(100)	0.25(11)	0.125(010)	0.125(001)			
R7	0.125(101)	0.125(100)	0.125(011)				
R3	0.125(110)	0.125(101)					
R6	0.125(111)						

The average coding length of example 3 of Huffman coding is:

L_{avg}=3*0.125*8= 3 (bits/pixel)

d) (5/5) Comment on relationship between probability combination, entropy, amount of information, and uncertainty.

The larger the probability combination (for example, 50%*50% > 100%*0%), the higher the entropy, the higher the uncertainty, and the smaller the amount of information that we could get.

(60/40) Task 2: Integrated Approach

The following figure shows an 8x8 image. Fixed-length coding (00, 01, 10, 11) is used for its four brightness levels. Answer the following questions:

a) (5/5) Identify at least two kinds of redundancy existed in this image. Briefly

explain the redundancy.

In this image, coding redundancy exists, since it did not take full advantages of probabilities of each gray level.

Spatial redundancy also exists, because the gray level each pixel is similar to the values of neighboring pixels'.

b) (10/5) Derive the probability of appearance (that forms the histogram) for each intensity level. Calculate the entropy of this image.

Pixel No. of 00 = 16

Pixel No. of 01 = 32

Pixel No. of 10 = 8

Pixel No. of 11 = 8

Probability

P(00) = 16/64 = 0.25

P(01) = 32/64 = 0.5

P(10) = 8/64 = 0.125

P(10) = 8/64 = 0.125

$H = 0.25*\log_2 0.25+0.5*\log_2 0.5+0.125*\log_2 0.125+0.125*\log_2 0.125=1.75(\text{bits/pixel})$

c) (10/5) Derive the Huffman code.

R0, R1, R2, and R3 are used to denote 00, 01, 10, and 11, respectively.

Table 4 Huffman coding of task2

Huffman coding for task 2

Pixel value	Probability	Step1	Step2
R0	0.5(0)	0.5(0)	0.5(0)
R1	0.25(10)	0.25(10)	0.5(1)
R5	0.125(110)	0.25(11)	
R4	0.125(111)		

d) (5/0) Calculate the average length of the original code and that of the derived Huffman code.

$L_{original} = 2;$

$L_{Huffman} = 0.5*1+0.25*2+0.125*3+0.125*3 = 1.75$

e) (5/0) Calculate the ratio of image size (in bits) between using the fixed length coding (size1) and Huffman coding (size2). Calculate the relative coding redundancy.

the ratio of image size (in bits) between using the fixed length coding (size1) and Huffman coding (size2):

$C_R = \text{size1/size2} = (2*64)/(1.75*64) = 1.14$

relative coding redundancy:

$R_D = 1 - 1/C_R = 0.125$

f) (25/25) Choose one of the following problems and complete the whole compression process (i.e., complete all the three steps listed).

i) Use bit-plane coding + run-length coding + Huffman coding to compress the

image. Calculate the final image size (in bits) (size3).

Initial coding is in table 5

Table 5 Original code for task2

01	01	01	01	01	01	01	01
10	00	11	00	00	11	00	01
10	00	01	00	01	00	00	01
10	00	01	10	11	01	11	01
10	00	01	11	10	01	00	01
10	00	01	01	01	01	11	01
00	11	00	11	00	10	00	01
01	01	01	01	01	01	01	01

1) When applying XOR binary, pixel values will be transfered as follows: From 00 to 00, 01 to 01, 10 to 11, 11 to 10, as in table 6.

Table 6 XOR binary code for task2

01	01	01	01	01	01	01	01
11	00	10	00	00	10	00	01
11	00	01	00	01	00	00	01
11	00	01	11	10	01	10	01
11	00	01	10	11	01	00	01
11	00	01	01	01	01	10	01
00	10	00	10	00	11	00	01
01	01	01	01	01	01	01	01

2) Bitplane coding is in table 7 a), and b).

Table 7 a) Bit plane 1

0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0
1	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0
1	0	0	1	1	0	0	0
1	0	0	0	0	0	1	0
0	1	0	1	0	1	0	0
0	0	0	0	0	0	0	0

Table 7 b) Bit plane 0

1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1
1	0	1	0	1	0	0	1
1	0	1	1	0	1	0	1
1	0	1	0	1	1	0	1
1	0	1	1	1	1	0	1
0	0	0	0	0	1	0	1
1	1	1	1	1	1	1	1

3) Run-length coding is in table 8, starts with 1 in each row.

Table 8 Run-length coding for bit plane 1 and 0

Bit plane 1	Bit plane 0
0 8	8
1 1 1 2 1 2	1 6 1
1 7	1 1 1 1 1 2 1
1 2 2 1 1 1	1 1 2 1 1 1 1
1 2 2 3	1 1 1 1 2 1 1
1 5 1 1	1 1 4 1 1
0 1 1 1 1 1 1 2	0 5 1 1 1
0 8	8

4) Probability distribution is shown in table 9.

Table 9 Probability distribution for bit plane 1 and bit plane 0

Bit plane 1			Bit plane 0		
Pixel value	No.	Probability	Pixel value	No.	Probability
0	3	0.09	0	1	0.03
1	19	0.56	1	27	0.75
2	7	0.20	2	3	0.08
3	1	0.03	3	0	0
4	0	0	4	1	0.03
5	1	0.03	5	1	0.03
6	0	0	6	1	0.03
7	1	0.03	7	0	0
8	2	0.06	8	2	0.05

Table 10 Probability distribution for a) bit plane 1, and b) bit plane 0

a) Huffman coding for bit plane 1

Pixel value	No.	Step1	Step2	Step3	Step4	Step5	Step6
1	19	0.56(1)	0.56(1)	0.56(1)	0.56(1)	0.56(1)	0.44(0)
2	7	0.20(01)	0.20(01)	0.20(01)	0.20(01)	0.24(00)	0.56(1)
0	3	0.09(001)	0.09(001)	0.09(001)	0.15(000)	0.20(01)	
8	2	0.06(0001)	0.06(0001)	0.09(0000)	0.09(001)		
3	1	0.03(00001)	0.06(00000)	0.06(0001)			
5	1	0.03(000000)	0.03(00001)				
7	1	0.03(000001)					

b) Huffman coding for bit plane 0

Pixel value	No.	Step1	Step2	Step3	Step4	Step5	Step6
1	27	0.75(1)	0.75(1)	0.75(1)	0.75(1)	0.75(1)	0.25(0)
2	3	0.08(000)	0.08(000)	0.08(000)	0.12(01)	0.13(00)	0.75(1)
8	2	0.05(001)	0.05(001)	0.05(001)	0.08(000)	0.12(01)	
0	1	0.03(0100)	0.06(011)	0.06(010)	0.05(001)		
4	1	0.03(0101)	0.03(0100)	0.06(011)			
5	1	0.03(0110)	0.03(0101)				
6	1	0.03(0111)					

The size of each bit-plane is as follows:

Size1 = 1*19 + 2*7 + 3*3 + 4*2 + 5*1 + 6*1 + 6*1 = 67(bits)

Size0 = 1*27 + 3*3 + 3*2 + 4*1 + 4*1 + 4*1 + 4*1 = 58(bits)

Therefore, the final image size = 67 + 58 = 125 (bits)

(+25/25) Task 3: DM Lossy Predictive Coding

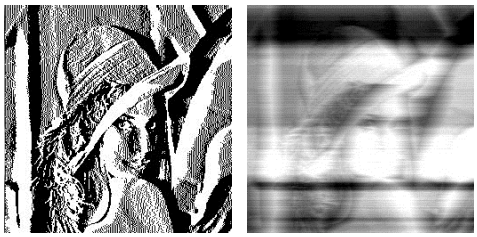
(15) Task 3.1: Implement DM lossy predictive coder and decoder.

The coder and decoder codes are in the cpp files.

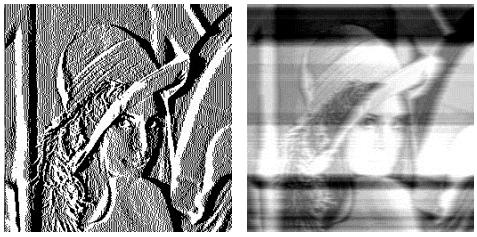
(5) Task 3.2: Show decompression results by using different delta values.

Results are shown in Fig. 1.

original

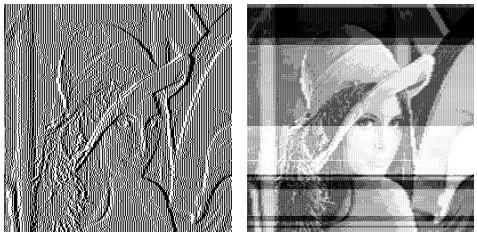


Delta=5

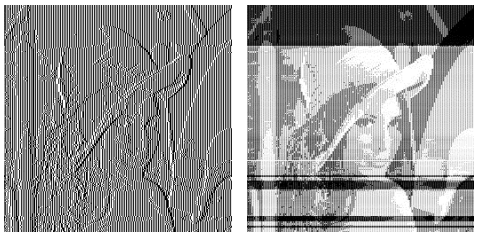


Delta=10

Delta=20



Delta=40



Delta=80

Figure 1 Code (left column) and decode (right column) effects on a black-white image

(5) Task 3.3: Comment on the result.

As shown in Fig.1, the left column shows the decoding results, while the right column shows the decoding results. When used different deltas, the coding and decoding effects are also quite different. With the increase of delta from 5 to 80, the coding images become more blurred, and the decoding images become better until delta=20, and then blurred more.

Why? When delta is too small, it can not represent the input's largest changes, then a distoration of slope overload occurs. However, when delta is too large, it can not represent the smallest changes, then the granular noise appears.

Source code

As shown in the .ps files.