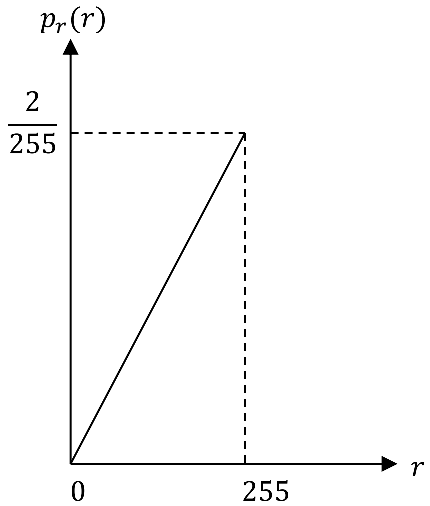
Image Compression Sample Exam Problems

Imperial College UG4, MSc COMSP, MSc AML

1. Consider an image with intensity $f(x,y)$ that can be modeled as a sample obtained from the probability density function $p_r(r)$ sketched below:



(i) Suppose that **three** reconstruction levels are assigned to quantize the intensity $f(x,y)$. Determine these reconstruction levels using a uniform quantizer.
(ii) Determine the codeword to be assigned to each of the three reconstruction levels using Huffman coding. For your codeword assignment, determine the average number of bits required to represent r.
(iii) Determine the entropy, the redundancy and the coding efficiency of the Huffman code for this example. Comment on the efficiency of Huffman code for this particular set of symbols.

Solution

(i)

The range of intensities is divided in three intervals, namely, $\left[0, \frac{255}{3}\right], \left[\frac{255}{3}, 2 \cdot \frac{255}{3}\right], \left[2 \cdot \frac{255}{3}, 1\right]$ or $[0, 85], [85,\ 170], [170, 255]$.
The three reconstruction levels are:

$s_1 = \frac{1}{2}85 \cong 43$, $s_2 = 43 + 85 = 128$, $s_3 = 128 + 85 = 213$.

(ii)

The equation which describes the pdf function is $p_r(r) = \frac{2}{255 \cdot 255} r$. Based on this, we can find the probabilities of the 3 reconstruction levels.

The probability of s_1 is $\int_0^{\frac{255}{3}} \frac{2}{255 \cdot 255} w dw = \frac{2}{255 \cdot 255} \frac{w^2}{2} \Big|_0^{\frac{255}{3}} = \frac{2}{255 \cdot 255} \frac{255^2}{2 \cdot 3^2} = \frac{1}{9}$

The probability of s_2 is $\int_{\frac{255}{3}}^{\frac{2 \cdot 255}{3}} \frac{2}{255 \cdot 255} w dw = \frac{2}{255 \cdot 255} \frac{w^2}{2} \Big|_{\frac{255}{3}}^{\frac{2 \cdot 255}{3}} = \frac{2}{255 \cdot 255} \left(\frac{4 \cdot 255^2}{2 \cdot 3^2} - \frac{255^2}{2 \cdot 3^2} \right) = \frac{1}{3}$

The probability of s_3 is $1 - \frac{1}{9} - \frac{1}{3} = 1 - \frac{1}{9} - \frac{3}{9} = \frac{5}{9}$

Huffman coding of the above set of symbols is easy to calculate. You can work it out. It is presented in the Table below.

Symbol	**Probability**	**Code**
$s_1 = 43$	1/9	11
$s_2 = 128$	3/9	10
$s_3 = 213$	5/9	0

The average number of bits per symbol is $l_{avg} = \frac{1}{9} \cdot 2 + \frac{3}{9} \cdot 2 + \frac{5}{9} \cdot 1 = \frac{13}{9} = 1.444$ bits/symbol.

(iii)

The set of symbols (alphabet) consists of the three reconstruction levels. The entropy of the set is: $H(s) = \sum_{i=1}^{3} p_i I(s_i) = -\sum_{i=1}^{3} p_i \log_2(p_i) = 1.35164$ bits/symbol.

The redundancy is $l_{avg} - H(s) = 1.444 - 1.35164 = 0.09236$ or $\frac{l_{avg} - H(s)}{H(s)}\% = 6.8\%$ of entropy. Huffman code exhibits a low redundancy for the specific alphabet and therefore, it is efficient enough.

2. Consider a grey level image $f(x, y)$ with grey levels ranging from 0 to 255. Assume that the image $f(x, y)$ has medium contrast. Furthermore, assume that the image $f(x, y)$ contains large areas of slowly varying intensity. Consider now the image:

$$g(x, y) = f(x, y) - 0.5f(x - 1, y) - 0.5f(x, y - 1).$$

(i) Sketch a possible histogram of the image $f(x, y)$.

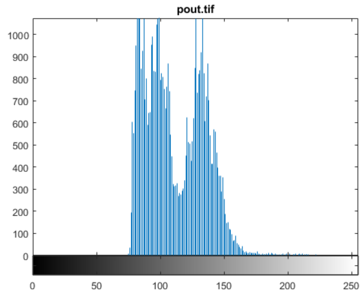
(ii) Discuss the characteristics of the histogram of the image $g(x, y)$.

(iii) Explain which of the two images is more amenable to compression using Huffman code.

Solution

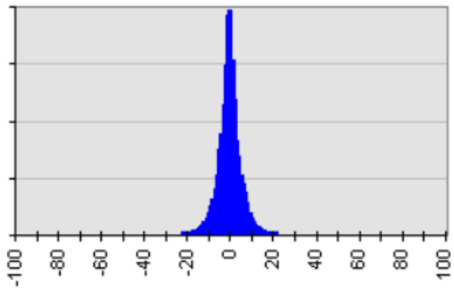
(i)

A possible histogram of a real-life medium contrast image is shown below.



(ii)

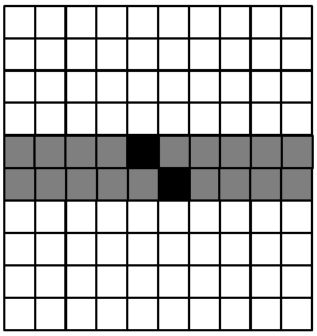
Due to the fact that natural images contain large areas of slowly varying intensity, we may assume that the image $g(x,y)$ contains a large number of small values and furthermore, it has both positive and negative values which have equal probabilities. Therefore, a possible histogram for $g(x,y)$ is shown below.



(iii)

Huffman coding $g(x,y)$ instead of $f(x,y)$ is more meaningful since the pdf of $g(x,y)$ is less uniform.

3. The following figure shows a 10×10 image with 3 different grey levels (black, a shade of grey, white).



(i) Derive the probability of appearance for each intensity (grey) level. Calculate the entropy of this image. Derive the Huffman code.
(ii) Calculate the length of the fixed length code and that of the derived Huffman code.
(iii) Calculate the ratio of image size (in bits) between using the fixed length code and Huffman code. Calculate the relative coding redundancy.
(iv) Derive the extended-by-two Huffman code.
(v) Calculate the ratio of image size (in bits) between using the fixed length code and the extended Huffman code. Calculate the relative coding redundancy. Comment on the efficiency of the extended Huffman code for this image.

Solution

(i)

The probability of appearance of each intensity is shown in Table below:

Symbol (Intensity)	Probability	Codeword
white s_1	0.8	0
black s_2	0.02	11
grey s_3	0.18	10

The entropy of the source is $H(s) = \sum_{i=1}^{3} p_i I(s_i) = -\sum_{i=1}^{3} p_i \log_2(p_i) = 0.816$ bits/symbol.

(ii)

Since we have 3 symbols, a fixed-length code would require $l_{\text{fixed}} = 2$ bits per symbol. With the use of the above Huffman code the average number of bits per symbol is $l_{\text{avg}} = \sum_{i=1}^{3} p_i l_i = 0.8 + 0.02 \cdot 2 + 0.18 \cdot 2 = 1.2$ bits/symbol.

(iii)

The ratio of image size (in bits) between using the fixed length code and the Huffman code is $\frac{l_{\text{fixed}}}{l_{\text{avg}}} = \frac{2}{1.2} = 1.67$. The coding redundancy is $l_{\text{avg}} - H(s) = 1.2 - 0.816 = 0.384$ bits per symbol.

(iv)

The extended-by-two Huffman code is shown in the Table below.

Symbol	Probability	Codeword
$s_1 s_1$	0.64	0
$s_1 s_2$	0.016	10101
$s_1 s_3$	0.144	11
$s_2 s_1$	0.016	101000
$s_2 s_2$	0.0004	10100101
$s_2 s_3$	0.0036	1010011
$s_3 s_1$	0.144	100
$s_3 s_2$	0.0036	10100100
$s_3 s_3$	0.0324	1011

(v)

With the use of the above Huffman code the average number of bits per symbol is $l_{\text{avg}} = \sum_{i=1}^{9} p_i l_i = 1.7228$ bits/**new** symbol or 0.8614 bits/**original** symbol. The ratio of image size (in bits) between using the fixed length code and the Huffman code is $\frac{l_{\text{fixed}}}{l_{\text{avg}}} = \frac{2}{0.8614} = 2.321$. The coding redundancy is $l_{\text{avg}} - H(s) = 0.8614 - 0.816 = 0.0454$ bits per symbol. Obviously, the extended Huffman code is more efficient compared to the standard Huffman code.

4. (i) Give the definition of a Discrete Memoryless Source (DMS).
   (ii) Consider a set of symbols generated from a DMS. Give the minimum number of bits per symbol that we can achieve if we use Huffman coding for the binary representation of the symbols. Explain what type of probabilities the symbols must possess in order to achieve the minimum number of bits per symbol using Huffman coding.
   (iii) Provide a scenario where Huffman coding would not reduce the number of bits per symbol from that achieved using fixed number of bits per symbol.

Solution

(i)

The discrete memoryless source (DMS) is defines as a symbol source which has the property that its output at a certain time does not depend on its output at any earlier time.

(ii)

The minimum number of bits per symbol we can achieve is the entropy of the source. In order to achieve this, the probabilities of the symbols must be negative powers of 2.

(iii)

This happens when the probabilities of the symbols are equal or have small differences.

5. The following figure shows a list of 7 symbols and their probabilities. It is assumed that these symbols are generated by a Discrete Memoryless Source (DMS).

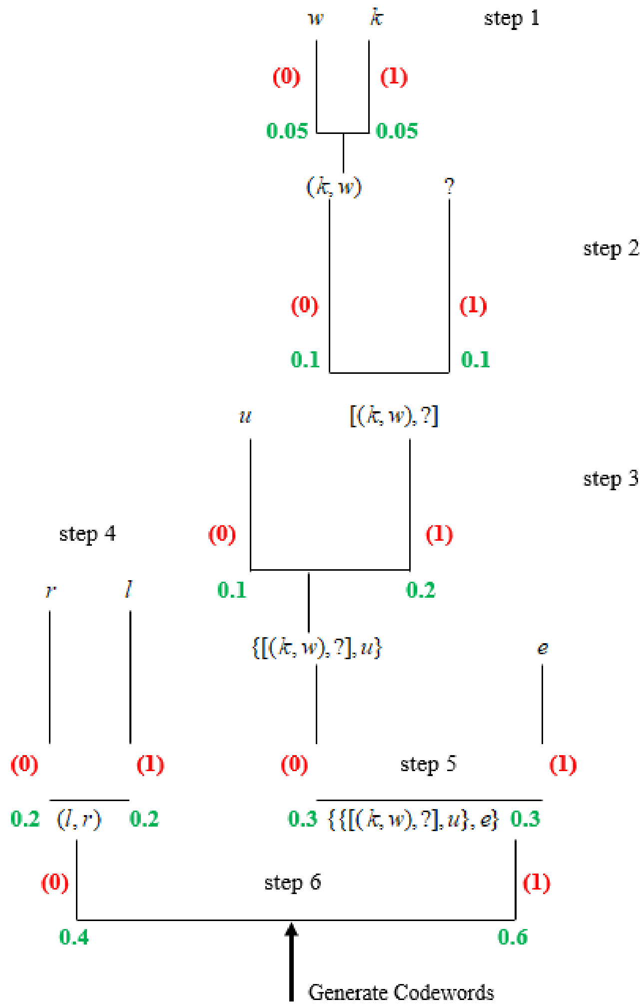
Symbol	**Probability**
k	0.05
l	0.2
u	0.1
w	0.05
e	0.3
r	0.2
?	0.1

   (i) Derive a Huffman code taking into consideration that in a particular transmission system, the probability of a 1 being transmitted as 0 is zero and the probability of a 0 being transmitted as a 1 is 0.05.
   (ii) Calculate the compression ratio.
   (iii) In the particular transmission system described in (i) above, find the probability of a codeword equal to 100 being transmitted wrongly.

Solution

(i)

The method to follows is that, while merging branches, 1 is preferred to 0 since it has zero probability of being wrongly transmitted and therefore, we assign 1s to branches that correspond to symbols with higher probabilities.



Symbol	**Probability**	**Codeword**
k	0.05	10101
l	0.2	01
u	0.1	100
w	0.05	10100
e	0.3	11
r	0.2	00
?	0.1	1011

(ii)

The average codeword length is calculated as:

$$l_{\text{avg}} = \sum_{i=1}^{7} p_i l_i = 0.05 \cdot 5 + 0.2 \cdot 2 + 0.1 \cdot 3 + 0.05 \cdot 5 + 0.3 \cdot 2 + 0.2 \cdot 2 + 0.1 \cdot 4 = 0.25 + 0.4 + 0.3 + 0.25 + 0.6 + 0.4 + 0.4 = 2.6 \text{ bits per symbol}$$

Since our alphabet has seven symbols, a fixed-length coder would require at least three bits per symbol. In this example, we have reduced the representation from three bits per symbol to 2.6 bits per symbol; thus, the corresponding compression ratio can be stated as $\frac{3}{2.6} = 1.15$.

(ii)

The probability of 100 being transmitted correctly is the collective probability of both zeros being transmitted correctly. The ones are not of any consideration since they are always transmitted correctly. The probability of a zero being transmitted correctly is 0.95. The probability of two consecutive zeros being transmitted correctly is $0.95 \cdot 0.95 = 0.9025$. Therefore, the probability of 100 being transmitted wrongly is $1 - 0.9025 = 0.0975$.

6. (i) Name three reasons why it might be beneficial to compress files.
   (ii) Consider the 8×8 image $f(x, y), x, y = 0, \ldots, 7$ shown in the figure below. The top left corner is the point $(x, y) = (0,0)$. Explain how **differential coding** can be used to compress this image if the prediction formula is $f(x, y) - f(x + 1, y)$ for $x < 7$ and 0 for $x = 7$.

x

0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3
5	5	5	5	5	5	5	5
7	7	7	7	7	7	7	7

y

Solution

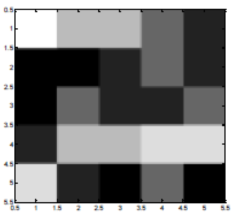
(i)

1. To save memory space.
2. To decrease transmission time when transferring files over networks.
3. To make some programs work faster (e.g., by decreasing disk access time).

(ii)

After differential coding using the given formula, the resulting image has only 3 values, namely, $-1,1,0$. The value of 0 dominates, and therefore, it is must more efficient to use differential coding since the histogram of the new image is more skewed.

7. The following figure shows a 5×5 image with 6 different grey levels with values shown on the right figure.



180	160	160	140	120
110	110	120	140	120
110	140	120	120	140
120	160	160	170	170
170	120	110	140	110

(i) Derive the probability of appearance (that forms the histogram) for each intensity (grey) level. Calculate the entropy of this image.
(ii) Derive the Huffman code.
(iii) Calculate the length of the fixed length code and that of the derived Huffman code.
(iv) Calculate the compression ratio and the relative coding redundancy.

Solution

(i)

$$p(110) = \frac{5}{25} = 0.2, p(120) = \frac{7}{25} = 0.28, p(140) = \frac{5}{25} = 0.2, p(160) = \frac{4}{25} = 0.16$$
$$p(170) = \frac{3}{25} = 0.12, p(180) = \frac{1}{25} = 0.04$$
$H(s) = \sum_{i=1}^{6} p_i I(s_i) = -\sum_{i=1}^{6} p_i \log_2(p_i) = 2.4188$ bits/symbol

(ii)

Symbol	Prob	Code	Prob	Code	Prob	Code	Prob	Code	Prob	Code
120	0.28	**01**	0.28	01	0.32	00	0.40	1	0.60	0
110	0.2	**10**	0.2	10	0.28	01	**0.32**	00	0.40	1
140	0.2	**11**	0.2	11	**0.2**	10	**0.28**	01		
160	0.16	**000**	**0.16**	000	**0.2**	11				
170	**0.12**	**0010**	**0.16**	001						
180	**0.04**	**0011**								

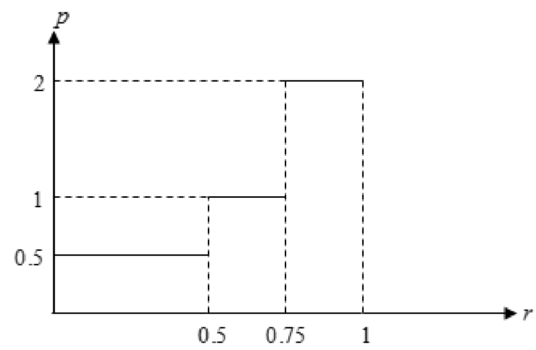
(iii)

We have 6 symbols and therefore, the fixed length representation requires 3 bits/symbol.
Using the above Huffman code yields:

$$l_{\text{avg}} = \sum_{i=1}^{6} p_i l_i = 2.48 \text{ bits per symbol}$$

(iv)
Based on the above measurements we have the following metrics:
Compression ratio $\frac{l_{\text{fixed}}}{l_{\text{avg}}} = \frac{3}{2.48}$
Redundancy $l_{\text{avg}} - H(s) = 2.48 - 2.4188 = 0.0612$ bits per symbol

8. Consider an image with intensity $f(x, y)$ that can be modeled as a sample obtained from the probability density function sketched below:



(i) Suppose that four reconstruction levels are assigned to quantize the intensity $f(x, y)$. Determine these reconstruction levels using a uniform quantizer.

(ii) Explain briefly why uniform quantization of an image may not be optimal in terms of the mean squared error.

(iii) Determine the codeword to be assigned to each of the four reconstruction levels using Huffman coding. Specify what the reconstruction level is for each codeword. For your codeword assignment, determine the average number of bits required to represent r.

(iv) Determine the entropy, the redundancy and the coding efficiency of the Huffman code for this example.

Solution

(i)

The four reconstruction levels are:
$r_1 = \frac{1}{8}$ with probability $p(r_1) = \frac{1}{4} \cdot 0.5 = \frac{1}{8}$
$r_2 = \frac{3}{8}$ with probability $p(r_2) = \frac{1}{4} \cdot 0.5 = \frac{1}{8}$
$r_3 = \frac{5}{8}$ with probability $p(r_3) = \frac{1}{4} \cdot 1 = \frac{1}{4}$
$r_4 = \frac{7}{8}$ with probability $p(r_4) = \frac{1}{4} \cdot 2 = \frac{1}{2}$

(ii)

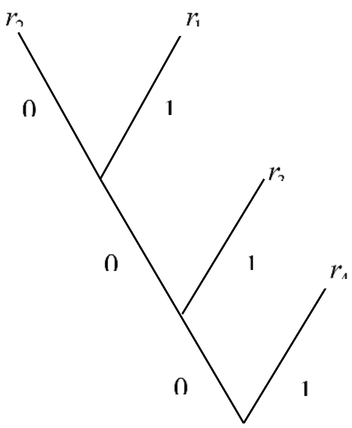
Uniform quantization is not optimal because it does not exploit the characteristics of the pdf of the alphabet to be quantized. A good quantizer assigns more reconstruction levels withing ranges of signal values that occur more frequently.

(iii)

The Huffman code is found below.

Step 1		Step 2		Step 3	
r_4	$1/2$	r_4	$1/2$	r_4	$1/2$
r_3	$1/4$	r_3	$1/4$	$\{r_3, \{r_2, r_1\}\}$	$1/2$
r_2	$1/8$	$\{r_2, r_1\}$	$1/4$		
r_1	$1/8$				

Symbol	***Codeword***
r_1	001
r_2	000
r_3	01
r_4	1



The average number of bits to represent $f(x,y)$ is:
$l_{\text{avg}} = \sum_{i=1}^{4} p_i l_i = \frac{1}{2} + \frac{1}{4} \cdot 2 + 2 \cdot \frac{1}{8} \cdot 3 = \frac{7}{4} = 1.75$ bits/symbol

(iv)

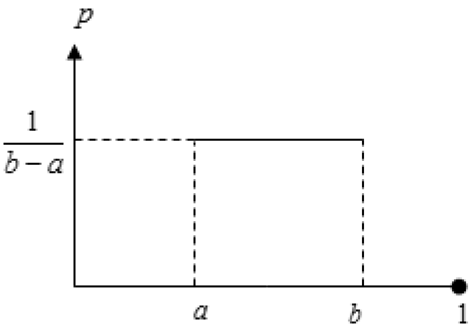
The entropy of the source is:

$$H(s) = \sum_{i=1}^{4} p_i I(s_i) = -\sum_{i=1}^{4} p_i \log_2(p_i) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) - 2 \cdot \frac{1}{8}\log_2\left(\frac{1}{8}\right)$$

$= \frac{1}{2} + \frac{1}{2} + 2 \cdot \frac{3}{8} = \frac{7}{4} = 1.75$ bits/symbol

Based on the above measurements we have the following metrics:
Coding efficiency $\frac{H(s)}{l_{\text{avg}}} = 1$
Redundancy $l_{\text{avg}} - H(s) = 0$ bits per symbol

9. Consider an image with intensity $f(x,y)$ that can be modeled as a sample obtained from the probability density function sketched below:

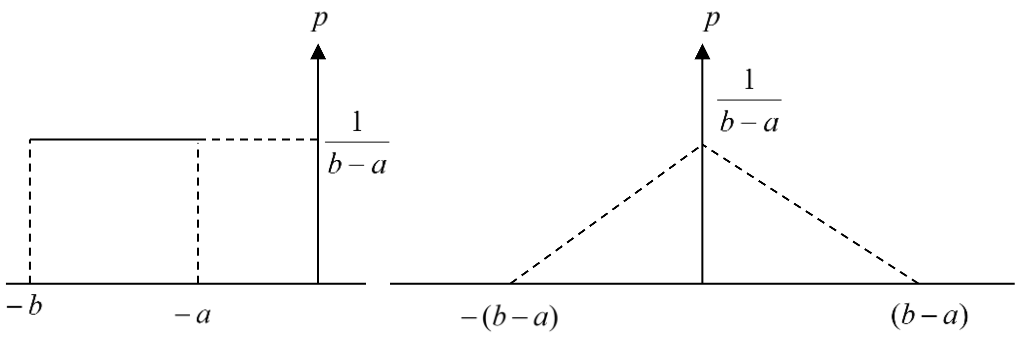


From $f(x,y)$ create an image

$$g(x,y) = f(x,y) - f(x-1,y-1).$$

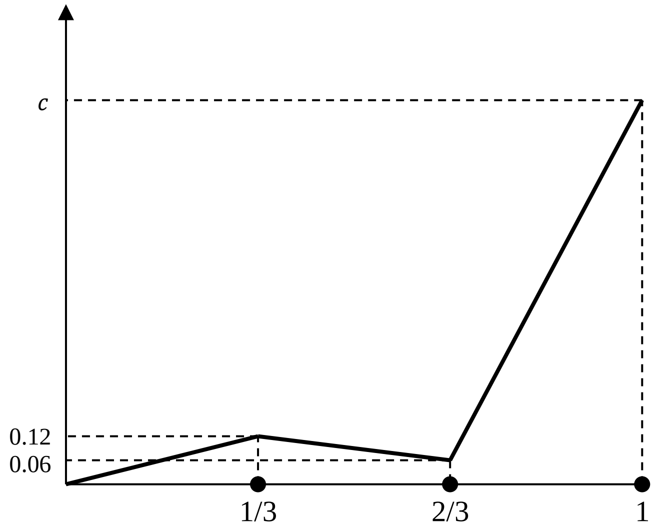
Explain why applying symbol encoding to $g(x,y)$ is more efficient than applying symbol encoding directly to the original image $f(x,y)$. Use the property that adding uncorrelated images convolves their histograms.

Solution



The pdf of $-f(x-1,y-1)$ is shown on the left and the pdf of $g(x,y)$ is shown on the right (convolution of the pdfs of $f(x,y)$ and $g(x,y)$). The pdf of $g(x,y)$ is more skewed and therefore, more appropriate for symbol encoding.

10. Consider an image with intensity $f(x, y)$ that can be modeled as a sample obtained from the probability density function sketched in the figure below.



Figure

(i) Determine the constant c shown in the figure above.
(ii) Suppose that three reconstruction levels are assigned to quantize the intensity $f(x, y)$. Determine these reconstruction levels using a uniform quantizer.
(iii) Determine the codeword to be assigned to each of the three reconstruction levels (symbols) using Huffman coding. For your codeword assignment, determine the average number of bits required to represent the image intensity.
(iv) Determine the entropy, the redundancy and the coding efficiency of the Huffman code for this example. Comment on the efficiency of Huffman code for this particular set of symbols.
(v) In the above set of symbols apply the extended by two Huffman coding. Explain the motivation for using extended Huffman code in the given set of symbols. Determine the redundancy and the coding efficiency of the extended by two Huffman code for this example. Comment on its efficiency.

Solution

(i)

The area under the curve shown above must be 1.
This consists of the following three areas:

1. The triangle on the left with area $\frac{0.12}{6} = 0.02$.
2. The trapezium in the middle with area
$$\frac{1}{2}(0.12 + 0.06) \cdot \frac{1}{3} = \frac{0.18}{6} = 0.03$$
3. The trapezium on the right with area
$$\frac{1}{2}(c + 0.06) \cdot \frac{1}{3} = \frac{c + 0.06}{6}$$
Therefore, $\frac{0.12+0.18+c+0.06}{6} = \frac{0.36+c}{6} = 1 \Rightarrow c = 5.64$
The trapezium in the right has area $\frac{5.64+0.06}{6} = \frac{5.7}{6} = 0.95$

(ii)

The three reconstruction levels are $r_1 = \frac{1}{6}, r_2 = \frac{3}{6}, r_3 = \frac{5}{6}$.

(iii)

The probabilities of the 3 reconstruction levels are:
$p\left(\frac{1}{6}\right) = 0.02$, $p\left(\frac{3}{6}\right) = 0.03$, $p\left(\frac{5}{6}\right) = 0.95$

Intensity	**Probability**	**Codeword**
r_1	0.02	00
r_2	0.03	01
r_3	0.95	1

The average number of bits to represent $f(x, y)$ is:
$l_{\text{avg}} = \sum_{i=1}^{3} p_i l_i = 0.95 + 0.03 \cdot 2 + 0.02 \cdot 2 = 1.05$ bits/symbol

(iv)

The entropy of the source is:
$H(s) = \sum_{i=1}^{3} p_i I(s_i) = -\sum_{i=1}^{3} p_i \log_2(p_i) = 0.3349$ bits/symbol

Based on the above measurements we have the following metrics:
Coding efficiency $\frac{H(s)}{l_{\text{avg}}} = 0.319$
Redundancy $l_{\text{avg}} - H(s) = 1.05 - 0.335 = 0.715$ bits per symbol
$\frac{l_{\text{avg}} - H(s)}{H(s)}$%=213% of entropy

(v)

Huffman code exhibits a very high redundancy for the specific alphabet and therefore, is not efficient enough.