[1] Arithmetic Coding of the Residual Image. In HW-1, we use the following filter to predict the pixel values,

|  |  |  |
| --- | --- | --- |
| 1/3 | 1/3 | 0 |
| 1/3 | 0 | 0 |
| 0 | 0 | 0 |

f1=

And verified that the predicted image residual indeed follows a GD like distribution, as shown in the Fig 1. b) and c) below (matlab code intraPrediction.m attached)

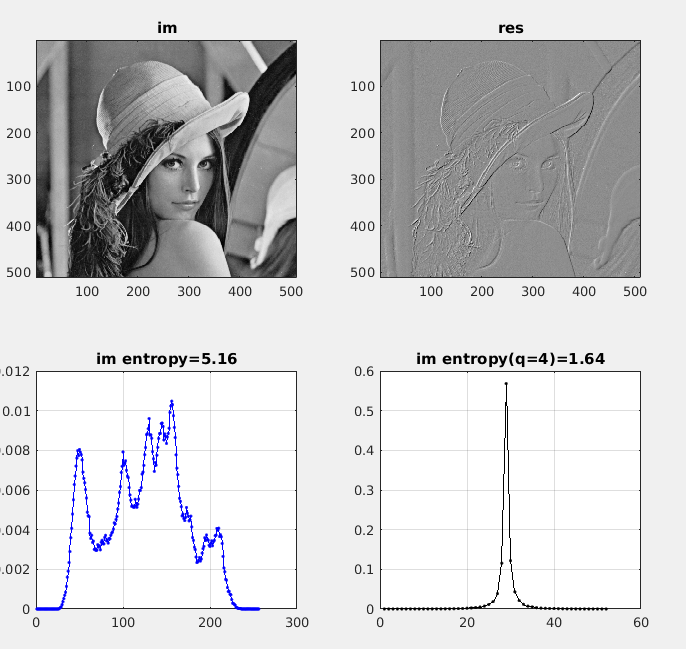


Figure 1.

[30 pts] a) for predicted error image, as shown in Fig.1d, with quantizaer q=1, design a ExpGolomb codes that can binarize the residual image representation, and then apply context adaptive binary arithmetic coding, i.e, the prob of symbols 1 and 0 are changing over the encoding process, and generate bit stream. What is the average (per pixel) bit rate of such coding scheme ?

The ExpGolomb Code book:

|  |  |  |  |
| --- | --- | --- | --- |
| **Category Code** | **Category Levels** | **Fixed Code Len** | **Fixed Level Code** |
| 0 | {0} |  | Nil |
| 10 | {-1, +1} | 1 | {0, 1} |
| 110 | {-3, -2, +2, +3} | 2 | {00 01 10 11} |
| 1110 | {-7 -6 -5 -4 4 5 6 7} | 3 | {000 001 010 011 100 101 110 111} |
| 11110 | {-15 -14, …, -9, -8, 8, 9, …. 15} | 4 | {0000 0001 … 1110 1111} |
| 111110 | {-31 -14 -16, 16 17,…, 30 31} | 5 | {00000 00001 … 11110 11111} |

The matlab code for intra prediction and quantization:

|  |
| --- |
|  |

[30pts] b) Now introducing quantization in intraPrediction() with q=[2, 4, 8], use the Golomb + AC in 1.a) to compute the bits per pixel, and also reconstruct the image from residual and compute the PSNR of the reconstruction quality. Note that the 1st row and 1st column are given directly, the residual images is of size (N-1) x (N-1) due to filtering.

[40pts] 2. For the bi-level image given in Fig. 2, design a context aware binary arithmetic coding scheme, i.e,

|  |  |  |
| --- | --- | --- |
| A | B | 0 |
| C | X | 0 |
| 0 | 0 | 0 |

The context consists of neighboring pixel locations {a, b, c} [hint, we have 8 context], and generate bitstream. What is the average bit rate in this case ?

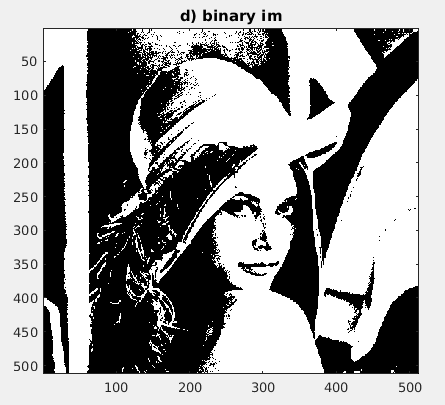


Figure 2. Binary Image

|  |
| --- |
|  |