**EE475 Compression**

**Fall 2012**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SUB-SAMPLING** | **PCM** | **DPCM** | **VQ** | **DCT** | **JPEG** | **Total** |
|  |  |  |  |  |  | **/200** |

1. **Try to give concise one or two sentence answers, explanations**
2. **I suggest you copy and past images from the VC Demo program to enhance your answers**

**IMAGE COMPRESSION SOFTWARE**

In the following, study the image processing function experimenting with various parameters, report your experience briefly in a few sentences. Copy and paste the images and graphs to accompany your commentaries.

**SUBSAMPLING**

* Inspect the spectrum of the image Build512B.bmp. The DC component is in the center. Give explanations for the observed line structures, that is identify the image structures to which the spectral line structures correspond to. Note: *SS Button 🡪 Spectrum 🡪 Apply*

As expected, the DC component is the strongest component. Other than that, there are a few lines going through the center, which are dominant. These few lines are caused mainly by the orientation of the building in the image. For instance, the horizontal lines are caused by the ground contours of the building, whereas the vertical lines can be caused by the pillars, windows, etc.

* Subsample by a factor of 2, once without anti-aliasing filter and then with 17-tap aliasing filter. Can you pinpoint the aliasing events in the subsampled image without anti-aliasing filtering? Does anti-aliasing filter improve it? In turn, does the anti-aliasing filter cause any distortions? Explain the difference in the spectra of the subsampled images with and without anti-aliasing filters. .

Note: *SS 🡪 Factor 🡪 2 , check Apply Subsampling box, check Blow-up Subsampled Image box*

*SS 🡪 Filter 🡪 Check or uncheck Apply anti-alias filter, Set the number of taps to 17.*

The downsampling of the image by a factor of 2 caused aliasing. It can be observed that the spectrum is repeated at the corners and also edges of the image. The anti-aliasing filter caused distortion where the repetitions of the spectrum overlap.

**PCM**

* Study the images Lena256B, Clown256B, Odie256B. At which rate do the artifacts become objectionable? At this rate how many gray values are available to represent the image?

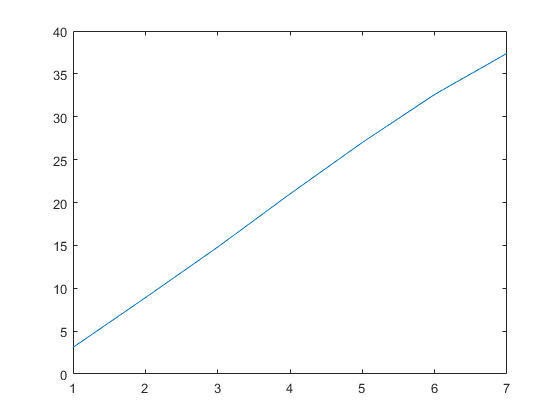
Note: *PCM button 🡪 Select bitrate 🡪 Apply. Do for bit rates 7 to 1.*

|  |  |  |
| --- | --- | --- |
|  | Bit-rate at which artifacts become objectionable | Gray values at this rate |
| Lena256B | 4 | 0-15 |
| Clown256B | 3 | 0-7 |
| Odie256B | 2 | 0-3 |

For Lena256B, Fill in the table.

|  |  |
| --- | --- |
| Bit rate | SNR (dB) |
| 1 | 3.1 |
| 2 | 8.9 |
| 3 | 14.8 |
| 4 | 21 |
| 5 | 27 |
| 6 | 32.6 |
| 7 | 37.4 |

* Draw an SNR-versus-bit-rate plot for Lena256B. Explain the reason why you are getting such a slope for this curve.



Increasing bitrate corresponds to increasing quality, and hence increasing SNR in the image. However, the curve slowly starts to converge after some point, since after some value, increasing bitrate has no meaning, i.e. it will have no contribution to the quality of the image.

* Explain the behavior of Odie under PCM, which is quite different than that of Lena. What distortions occur in Odie? What distortions occur in Lena?

The behavior of Odie is quite different than Lena, since it requires less bits for representation. The distortion in Odie occurs with the colors; even though the features are clearly visible, the gray level values are distorted due to insufficient bits. However, for Lena; there are many details, and the details require the usage of more bits to be recognizable. So, in the case of Lena, the distortion occurs as a loss of details, and also color.

**DPCM**

* Select the Lena256B image and the 1-D predictor. Carry out compression at bit rates 6 to 1 bpp and obtain the SNR – Observe the gain over PCM.

Note: *DPCM button 🡪 Model 🡪 Select the first prediction model. Select Bit Rate from 1 to 6.*

For Lena256B, Fill in the table.

|  |  |  |
| --- | --- | --- |
| Bit rate | DPCM SNR (dB) | PCM SNR (dB)  Copy from previous table |
| 1 | 7.4 | 3.1 |
| 2 | 13.2 | 8.9 |
| 3 | 18.5 | 14.8 |
| 4 | 24.3 | 21 |
| 5 | 31.2 | 27 |
| 6 | 36.5 | 32.6 |
| 7 |  |  |

* Compare visually the images with their PCM version and state at what rate DPCM achieves a performance equal to that of PCM?

The quality of 4-bit PCM can be achieved with 3-bit DPCM.

* Use four types of prediction region for DPCM at bit rate level 5 and comment on the performance differences. Sketch the prediction context.

Visually, there is no difference, however the SNR’s increase slightly. The SNR values for the 4 types of prediction are 31.2, 32.5, 32,7 35.0, respectively.

O X , O O X, O O O X, O O O O X.

* Observe the correlation matrix. What is the theoretical gain over PCM in terms of bit rate reduction at the same quality? Does the experimental result match the theory? Recall the prediction gain for one-tap predictor:  where r(1) is the normalized correlation coefficient.

Experimental results don’t match the theory. For example for OX model, the prediction gain is 8.0 and r(1) is 0.936. But looking at the above values we can’t see such high value of gain.

For OX model prediction gain is 8, for OOX model it is 18.2, for OOOX model it is 20.6 and for OOOOX model it is 23.1.

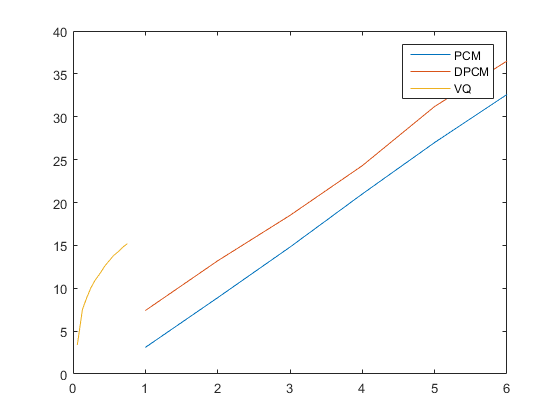
**VQ**

Load the predesigned codebook Standard\_4x4\_min1-max12.cbk. This codebook has been designed on 4 other images other than Lena.

Note: *VQ button 🡪 Codebook 🡪 Load codebook 🡪 Select Standard\_4x4\_min1-max12.cbk 🡪 Open*

*Bits 🡪 Select*

* Make an SNR-bit rate plot on the same curve as PCM/DPCM. Note: Be careful to first calculate the VQ bit rate (bits/pixel) before you plot.



**Start this curve from 0.**

* What type distortion do you observe .at low bit rates? Does it appear to be additive white Gaussian noise?

“ The distortion is not due to additive white Gaussian noise, since it does not look to be random. The distortion is due to constructing the image with fewer pixels, as fewer pixels mean less details and loss of some information. Because of the codebook, the correlation between the pixels are lost.

* Can you use codebooks obtained by Lena-type images on cartoons and maps? If yes, how? If not, why not?

Probably not, since cartoons do not have much detail, they consist of a few distinctive sections, whereas Lena-type images have many details and objects and hence, they have different neighborhood properties, therefore they will result in codebooks that are not suitable for cartoons.

**JPEG**

* Use the Lena256B image and the standard luminance normalization matrix. Fill the SNR-bit rate tables, one for each entropy coding choice. At the same time, write down the corresponding QF: Quality Factors. How much additional SNR does entropy-coding give?

Note: *JPEG button 🡪 Huffman tab 🡪 Select FLC, Standard VLC, Optimal VLC in turn*

*JPEG button 🡪 Bit rate tab 🡪 Bitrate 🡪 Select bit rate as 0.5, 0.8, 1.0, 1.5, 2.0, 2.5, 3.0 in turn 🡪 Apply*

Write down the Encoded Bit Rate in bpp, Optimized Quality Factor and the corresponding SNR. (FLC: Fixed Length Coding, VLC: Variable Length Coding)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bit rate | FLC | | | Standard VLC | | |
|  | Encoded Bit Rate | Quality factor | SNR (dB) | Encoded Bit Rate | Quality factor | SNR |
| 0.5 | 0.7 | 2 | 7.4 | 0.5 | 17 | 15.7 |
| 0.8 | 0.8 | 4 | 10.4 | 0.8 | 38 | 18.2 |
| 1.0 | 1.0 | 6 | 12 | 1.0 | 55 | 19.6 |
| 1.5 | 1.5 | 13 | 14.9 | 1.5 | 78 | 22.4 |
| 2.0 | 2.0 | 23 | 16.6 | 2.0 | 87 | 24.8 |
| 2.5 | 2.5 | 34 | 17.8 | 2.4 | 91 | 26.7 |
| 3.0 | 3.0 | 48 | 19.0 | 3.0 | 94 | 28.8 |

The SNR values for VLC coding are approximately 8dB higher than the SNR values for FLC.

* Compare the resulting SNRs from a “flat normalization matrix” against that of the “standard luminance normalization” and comment.

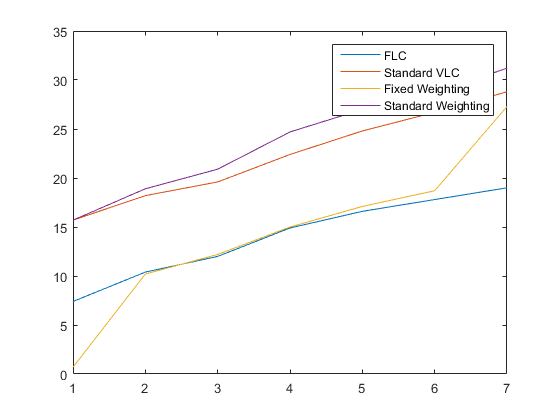
Note: *Repeat above selecting: Quant 🡪 Choose Flat*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bit rate | Fixed weighting | | | Standard weighting (psychovisual) | | |
|  | Encoded Bit Rate (b | Quality factor | SNR (dB) | Encoded Bit Rate | Quality factor | SNR |
| 0.5 | 0.6 | 2 | 0.7 | 0.5 | 41 | 15.7 |
| 0.8 | 0.8 | 14 | 10.2 | 0.8 | 66 | 18.9 |
| 1.0 | 1.0 | 21 | 12.2 | 1.0 | 76 | 20.9 |
| 1.5 | 1.5 | 36 | 15 | 1.5 | 87 | 24.7 |
| 2.0 | 2.0 | 53 | 17.1 | 2.0 | 91 | 27.0 |
| 2.5 | 2.5 | 65 | 18.7 | 2.4 | 93 | 28.7 |
| 3.0 | 3.0 | 72 | 27.3 | 2.9 | 95 | 31.2 |

The SNR values of fixed weighting are lower than the SNR values of standard rating.

Flat normalization matrix does not provide a significant enhancement in terms of SNR values, but the quality factor increases especially at low bit rates.

Draw the four bit rate-SNR curves on the same plot:



* Observe the images with flat versus standard normalization matrix at 0.5 bpp. Their SNRs are equal, yet one looks better than the other one. Explain this dilemma.

Because standard normalization is designed according to Human Visual System. Compared to flat basis; “psychovisual” basis looks better to human eye.

**DCT**

Select the Lena256B image. Study the block sizes of 2x2, 4x4 and 8x8. Explain what you observe in the “DCT: Original Coefficients” window. Explain what you observe in the “DCT: Coded Coefficients” window. Select PCM compression for all DCT coefficients. Pick c = 0.75 as the exponent power of the generalized Gaussian distribution.

Note: *DCT button 🡪 Size tab 🡪 Check encoding of DCT coefficients 🡪 Set transform size as 2x2, 4x4 and 8x8 in turn and press apply*

*Coefs tab 🡪 Select PCM for First and Others, Select C value as 0.75*

*Bitrate 🡪 Check the entropy coding. Select each of the bit rates in turn.*

We know that the low frequency components constitute the base of an image, and the high frequency components (the edges which give the detail) add upon them to refine the image, thereby giving a detailed image. Therefore we expect that the upper left part of the transform(low frequency components) has the most important information. We can clarify our assumption looking at the dct.

For different block sizes we get different dcts as expected. Because the image is broken into different number of blocks.

In contrast bottom right has both high frequency components in each direction.

In coded coefficients we get rid of the high frequency components which have small values. They are not coded because they are below the threshold value. Doing this we lose some information about the image but the magnitude is so low that we can reconstruct the original image perfectly by inverse dct. In coded coefficients we don’t have the bottom right part of dct. But it all depends on our compression ratio. The greater the compression value is, the greater distortion we will induce in image.

Increasing bit rate we get a better reconstruction step by step. Look at the below examples.





Size : 4x4 bitrate : 2.0

Size : 4x4 bitrate : 0.5