Scalar and Vector Quantization

-JIGAR GADA

USC ID: 8979-1025-58
jgada@usc.edu
Graduate Student, EE Dept.
University of Southern California

Problem 2. Scalar and Vector Quantization

Quantization is basically a mapping from many-to-few which makes the process non-reversible. Hence Compression techniques involving quantization are lossy. Depending on the type of input, the quantization can be scalar (scalar input) or vector (vector input).

2.1. Scalar quantization

2.1.1 Motivation

Scalar quantization in one of the simplest techniques for quantization. Generally in particular genre of images, the probability distribution is not uniform. Assigning equal number of bits to each the pixel can prove to be redundant. In such cases, we can reduce the number of bits used for representing the pixels at the cost of some loss in the data. This loss is not much noticeable and can be used for multimedia applications where quality of the images is not much important.

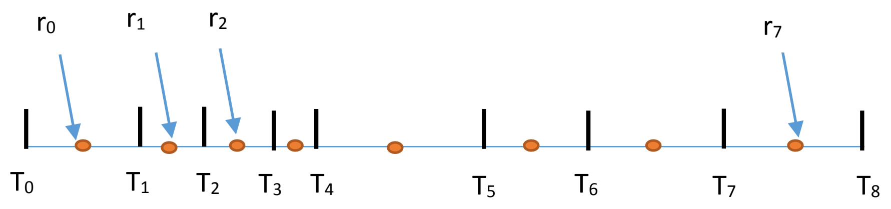
2.1.2 Approach

In this report, we have used Iterative Lloyd Max quantizer design which is non-uniform quantizer. The idea behind this design is simple.

- We divide the range into intervals (t_k, t_{k+1}) where the number of intervals depend on the number of bits used for quantization. For 'n' bit quantizer, the number of intervals will be 2^n.
- Each of the interval has a reconstruction value (r_k) i.e. If the value in the original data falls in an interval [t_k, t_{k+1}], then we represent its value with r_k.

We use the Lloyd max approach to design the intervals and its quantization value which is explained in the next section.

2.1.2.1. Iterative Lloyd Max Quantizer design



For n bit quantizer, L = 2^n

1. Guess initial set of representative levels,
   r_q, q = 0, 1… L-1
2. Calculate decision threshold

$$tq = \frac{r_q + q_{q-1}}{2}, \quad q = 0,1,\dots L-1$$

3. Calculate new representative values

$$r_q = \frac{\sum_{n=t_q}^{t_{q-1}} n\, P_n[n]}{\sum_{n=t_q}^{t_{q-1}} P[n]}, q = 0,1,\dots L-1$$

4. Update the average distortion

$$\frac{MSE_{j-1} - MSE_j}{MSE_j} < \varepsilon$$

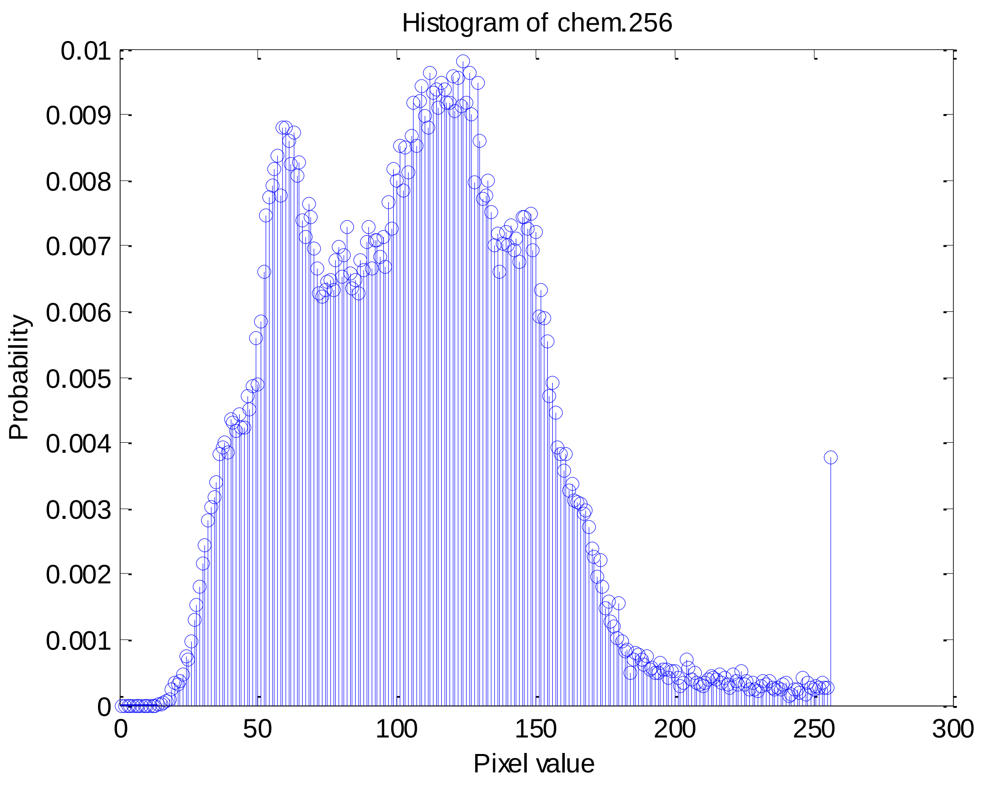
5. Repeat steps 2,3 and 4 until average distortion less than Ɛ (< 0.001)

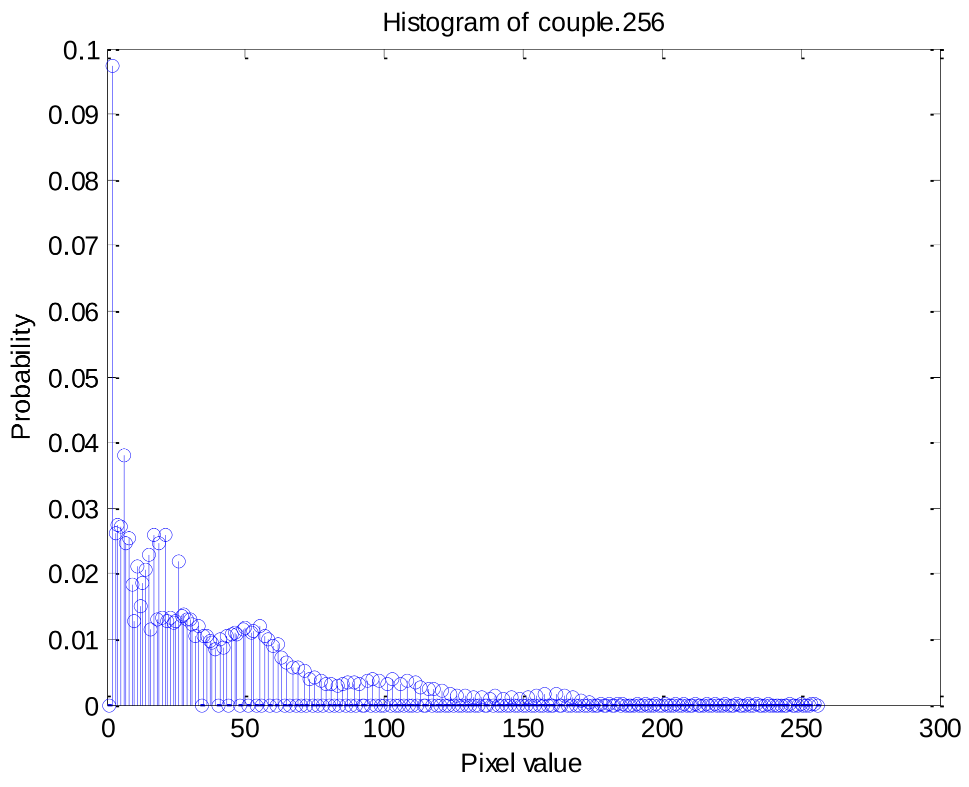
2.1.3. Problem 2, Part A

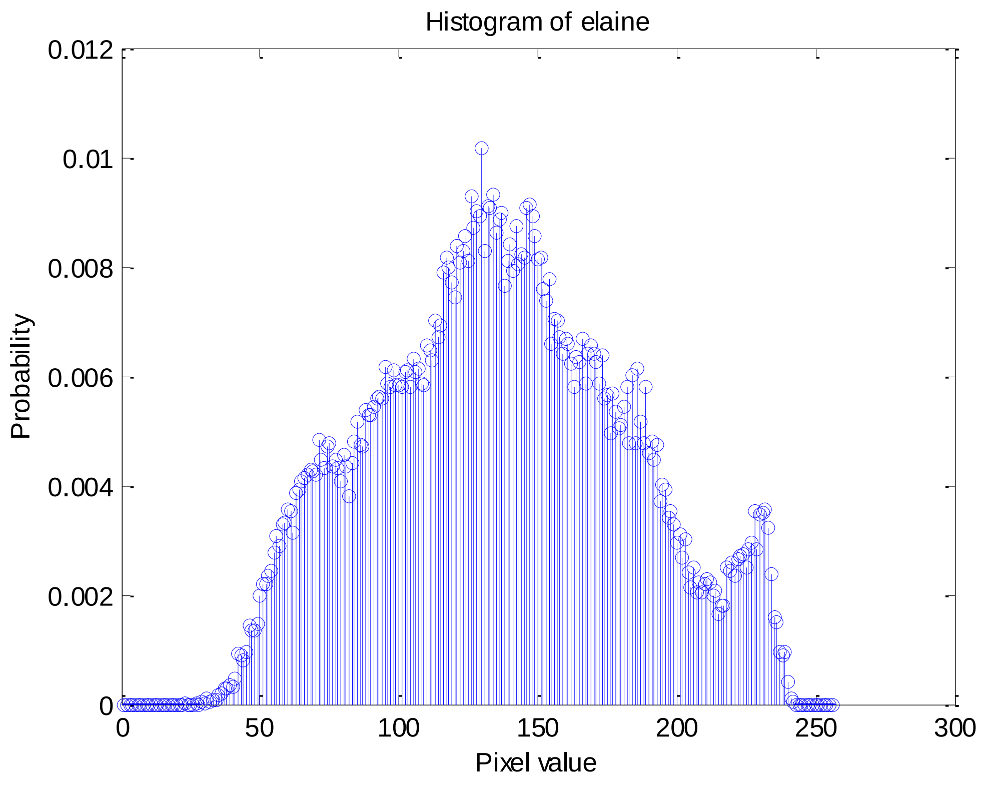
Following files have been provided for quantization. All files have a resolution of 256x256 with 8 bits/pixel.

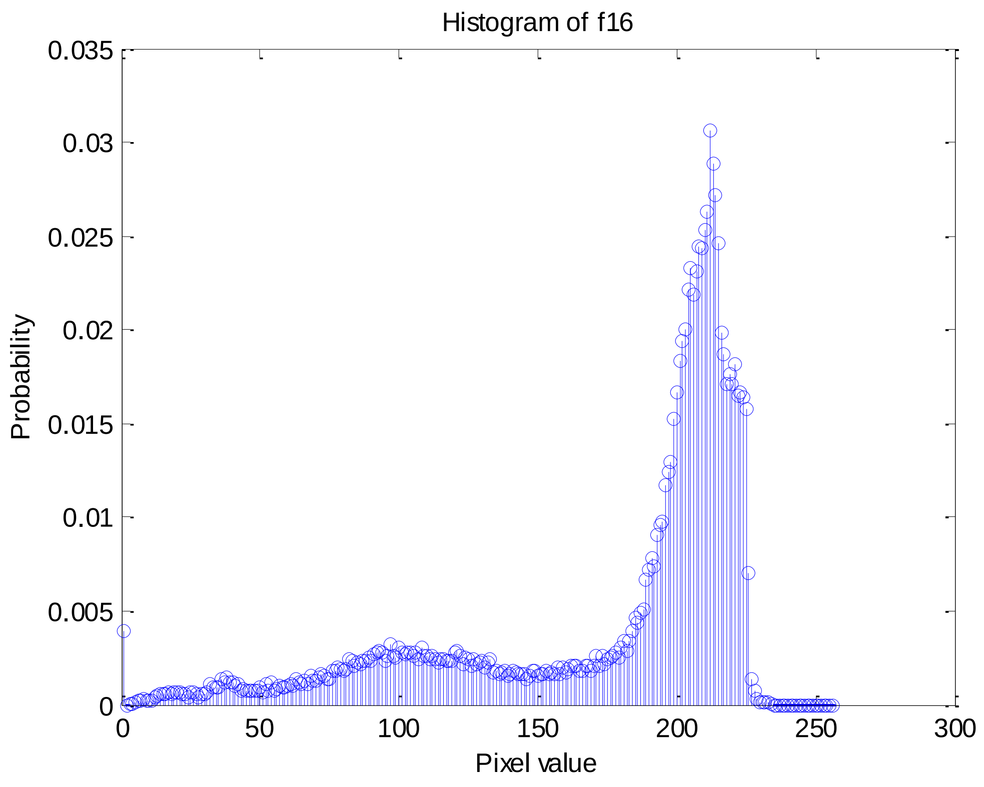
- chem.256
- couple.256
- elaine.256
- f16.256
- house.256
- moon.256

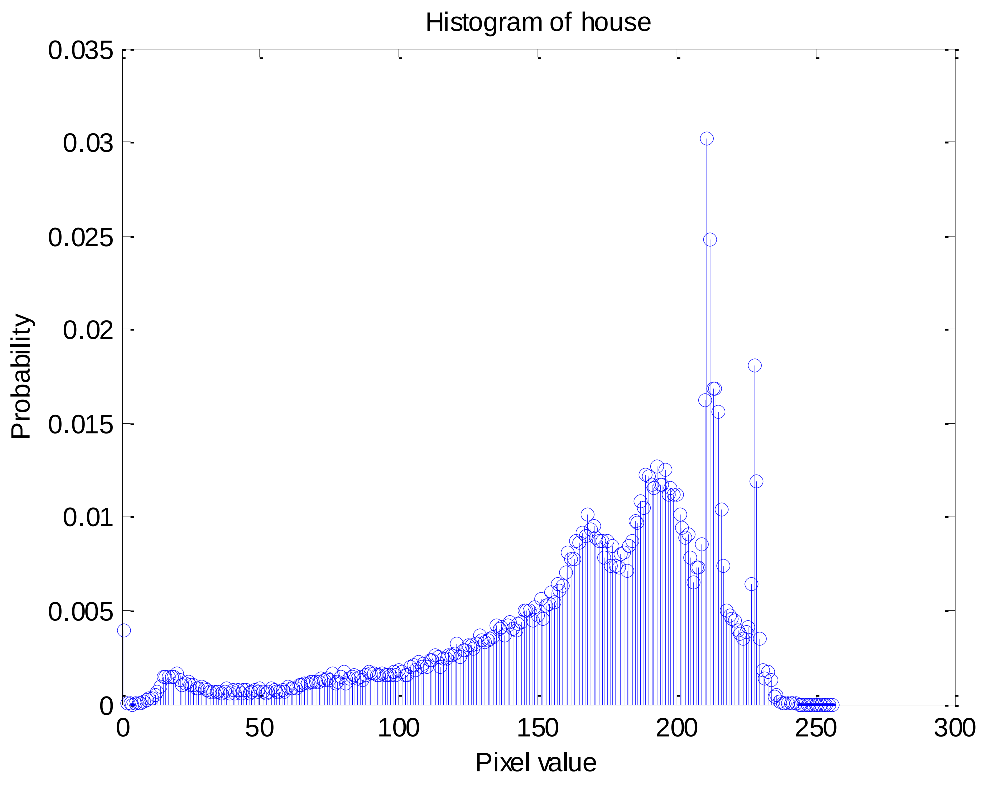
2.1.3.1. Histogram of each of the image file

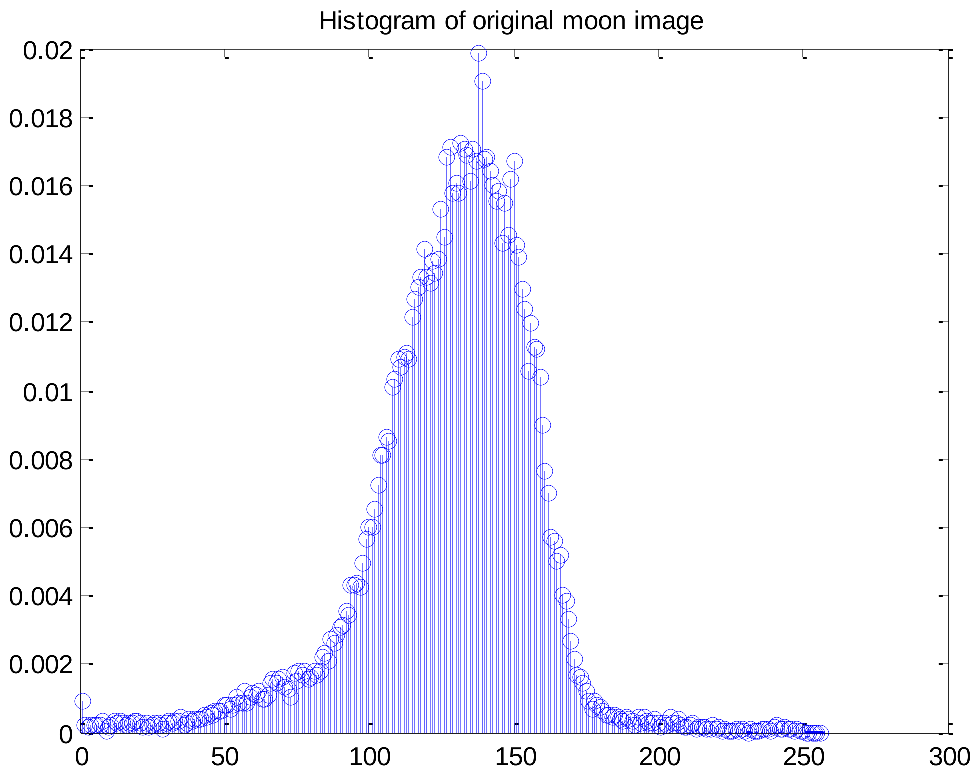



Histogram of couple.256
Probability
Pixel value
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
50
100
150
200
250
300


Histogram of elaine
Probability
Pixel value
0.012
0.01
0.008
0.006
0.004
0.002
0
0
50
100
150
200
250
300


Histogram of f16
Probability
Pixel value
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300


Histogram of house
Probability
Pixel value
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300



2.1.3.2 Designing a 3-bit and 5-bit scalar quantizer

The approach discussed in section 2.1.2.1. is used for training the quantizer.

A. Following 3 image files have been used for training:

- chem.356
- house.256
- moon.256

B. Initial Quantizer

For initial stage, uniform quantization has been chosen. The range 0-255 is divided uniformly into L intervals. Values of ‘r’ are center of each of the interval, r0 = (0 + 255/ (2*L)), r1 = (255/L + 255/ (2*L)) and so on.

E.g. For a 3-bit (L = 8) quantizer, the initial values of ‘r’ are:

r0	15.9375
r1	47.8125
r2	79.6875
r3	111.5625
r4	143.4375
r5	175.3125
r6	207.1875
r7	239.0625

C. Compact code

In some cases if the initial quantization steps are not chosen wisely, some codeword(s) may end up having no samples mapped to it resulting in empty quantization cells. This will result in an error in step 3 of Iterative Lloyd Max Quantizer design resulting in the value of P[n] = 0 for that cell and compiler will throw divide by zero error during runtime.

Such an error is avoided by assigning a value of 1 to the sum of P[n] if the value of P[n] = 0 for some level. Assigning such a small value to a particular interval will not harm the quantizer procedure. The algorithm will eventually iterate and assign larger intervals to the cells where the probability values are not dense.

D. Average distortion

Average distortion given by step 4 of quantizer design is updated at each step. The stopping criterion for the algorithm is when the average distortion is less than 0.001.

E. Peak Signal to Noise Ratio (PSNR)

Three images (2.1.3) have been used for training the quantizer.

For a 3-bit quantizer, it takes 15 iterations for the algorithm to converge and for 5-bit quantizer, it takes 3 iterations.

3 bit quantizer training	
Iteration no.	**PSNR**
1	29.0196
2	29.2545
3	29.4437
4	29.5817
5	29.6817
6	29.7799
7	29.8731
8	30.0033
9	30.1680
10	30.2732
11	30.3075
12	30.3438
13	30.3525
14	30.3593
15	30.3593

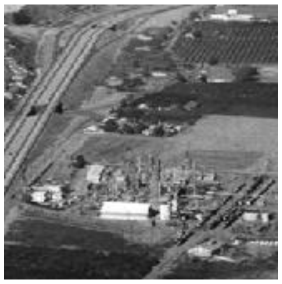
5 bit quantizer training	
Iteration no.	**PSNR**
1	41.0476
2	41.0570
3	41.0570

2.1.3.3. Using the designed Quantizer

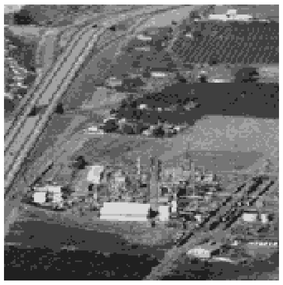
All the 6 images (train and test) are quantized using the above trained quantizer using both 3-bit and 5-bit quantizer.

The images below indicate the quality of 3-bit and 5-bit quantized image.

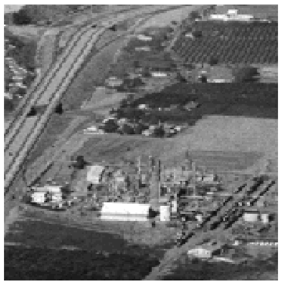
Original Image



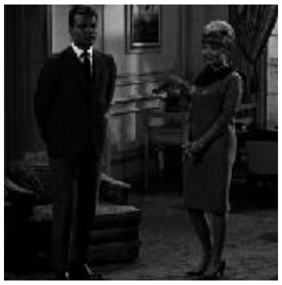
3 bit Quantized Image



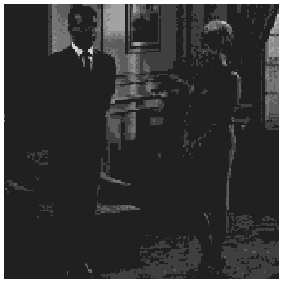
5 bit Quantized Image



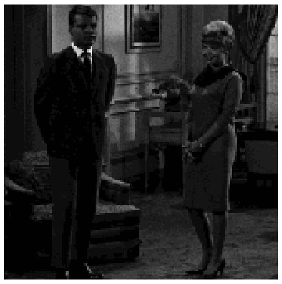
Original Image



3 bit Quantized Image

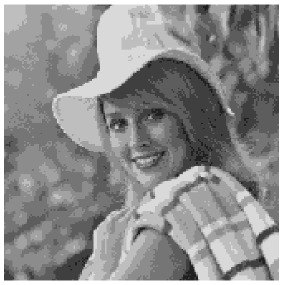


5 bit Quantized Image

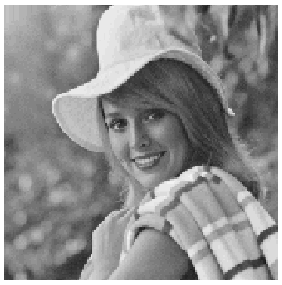


Original Image

3 bit Quantized Image

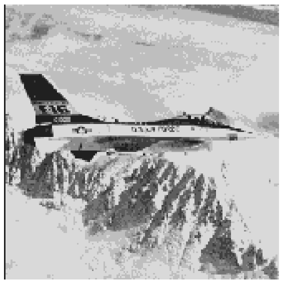


5 bit Quantized Image

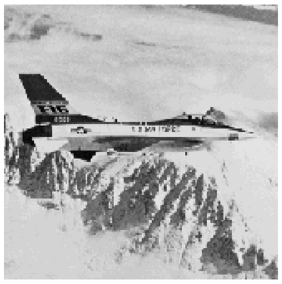


Original Image

3 bit Quantized Image

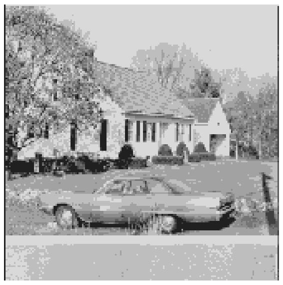


5 bit Quantized Image

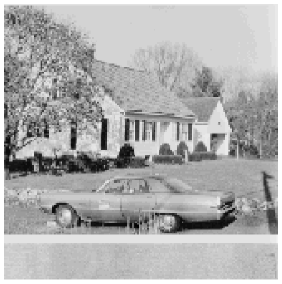


Original Image

3 bit Quantized Image

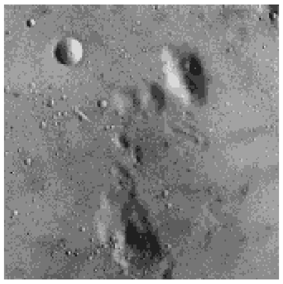


5 bit Quantized Image

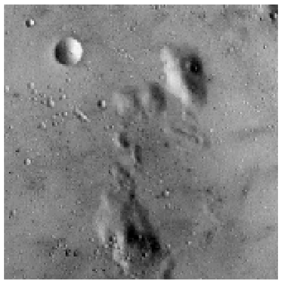


Original Image

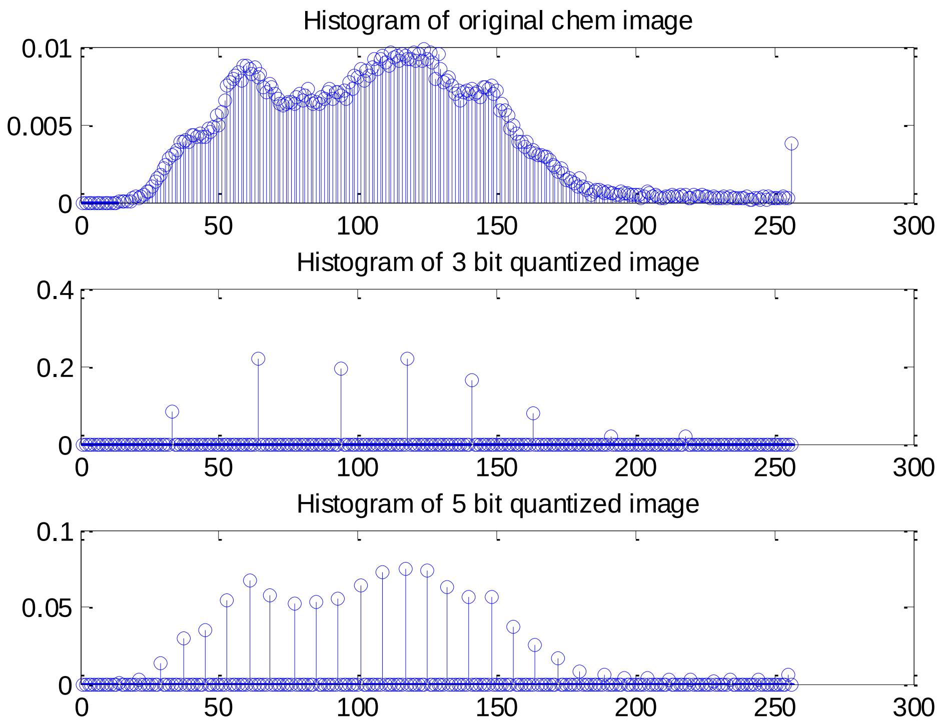
3 bit Quantized Image

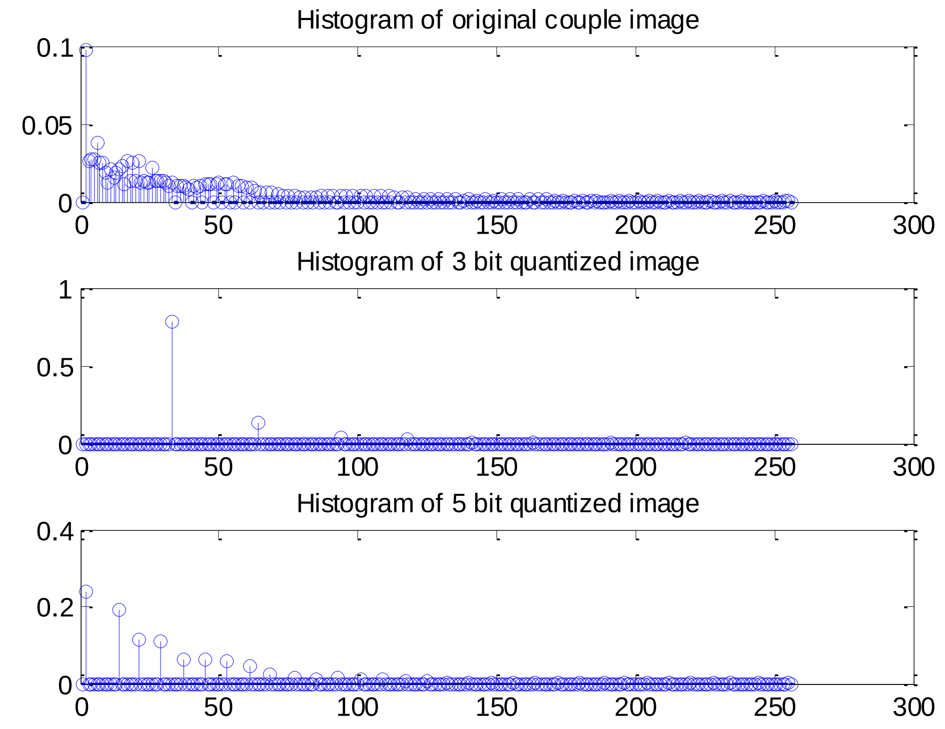


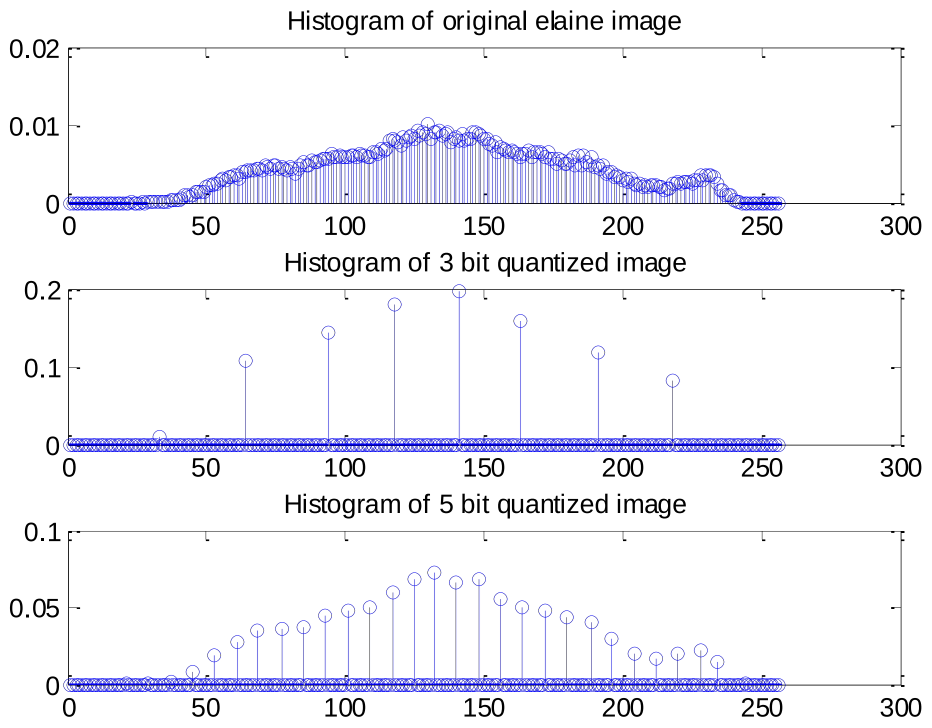
5 bit Quantized Image

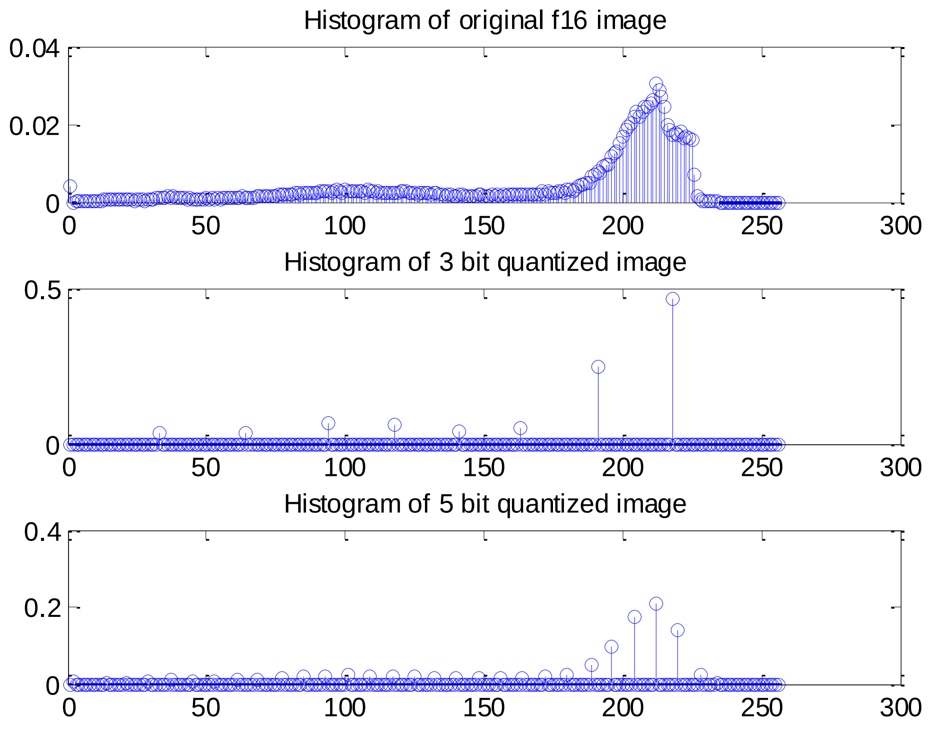


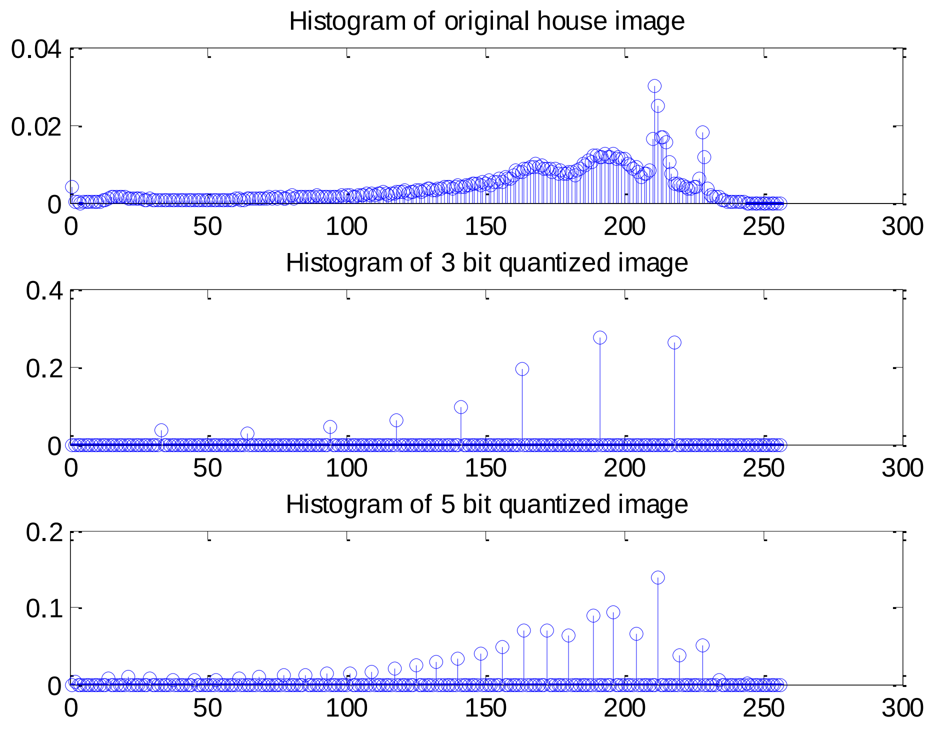
2.1.3.4. Comparison of histogram of the images before and after quantization

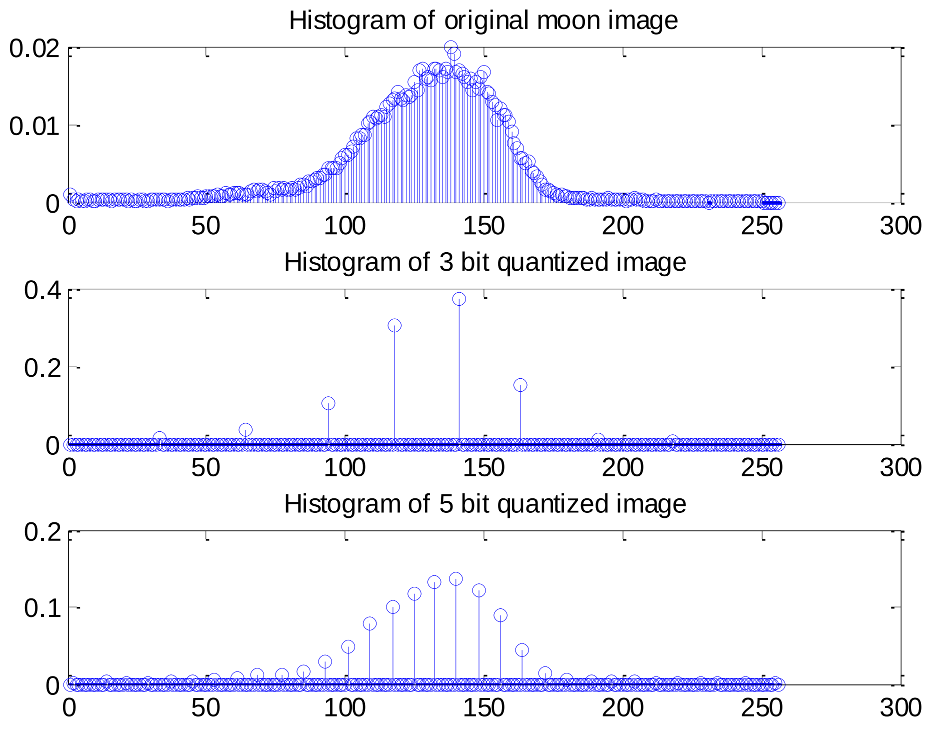





Histogram of original elaine image
0.02
0.01
0
0
50
100
150
200
250
300
Histogram of 3 bit quantized image
0.2
0.1
0
0
50
100
150
200
250
300
Histogram of 5 bit quantized image
0.1
0.05
0
0
50
100
150
200
250
300


Histogram of original f16 image
0.04
0.02
0
0
50
100
150
200
250
300
Histogram of 3 bit quantized image
0.5
0
0
50
100
150
200
250
300
Histogram of 5 bit quantized image
0.4
0.2
0
0
50
100
150
200
250
300





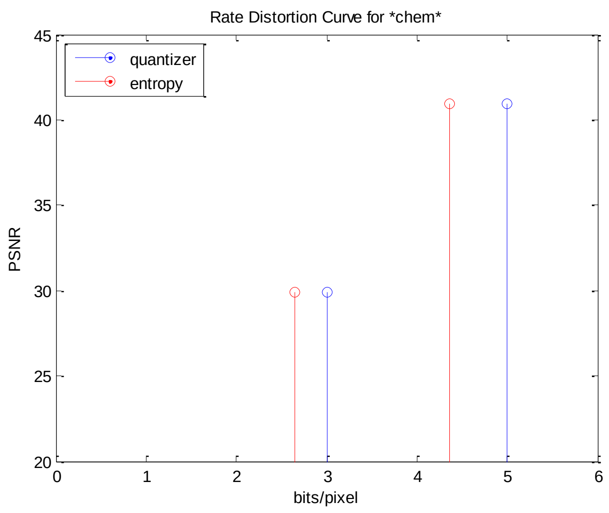
As we can see clearly from the above histogram images, an indication of good quantization is 'more levels in more probable areas'. I.e. the intervals are smaller in densely populated area.

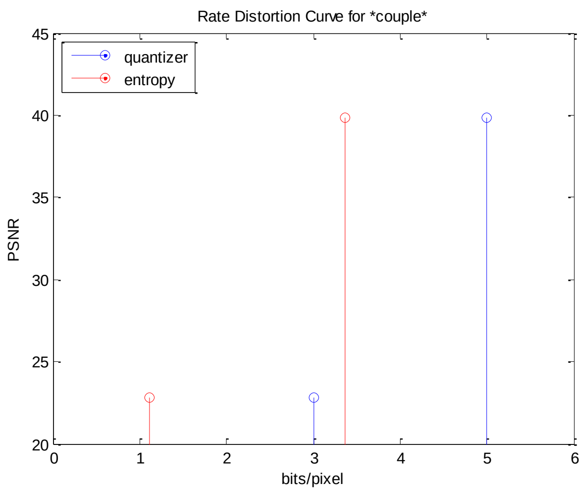
This idea makes the MSE smaller as the difference between the reconstructed value and the original value is smaller for most of the data.

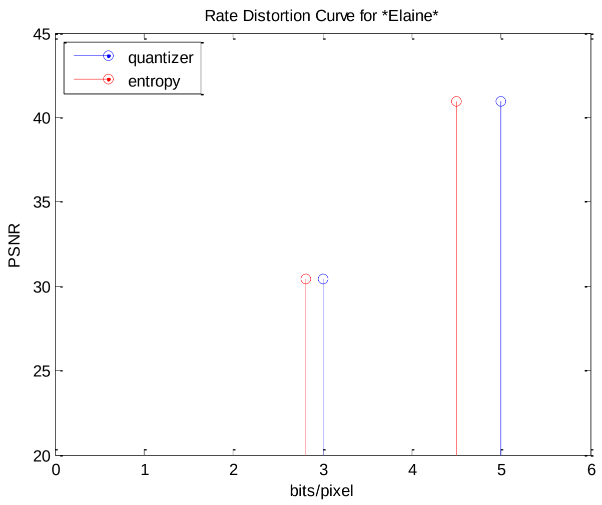
2.1.3.5. And 2.1.3.6 Rate Distortion pairs and theoretically optimal entropy code

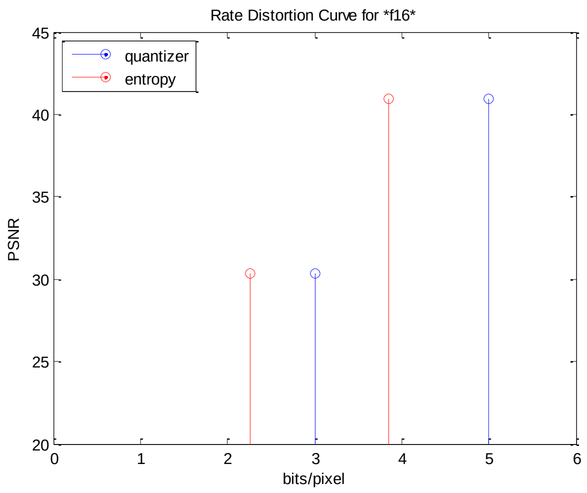
PSNR for each of the test and train images are computed using the above trained quantizer and optimal entropy code is computed for the quantized images.

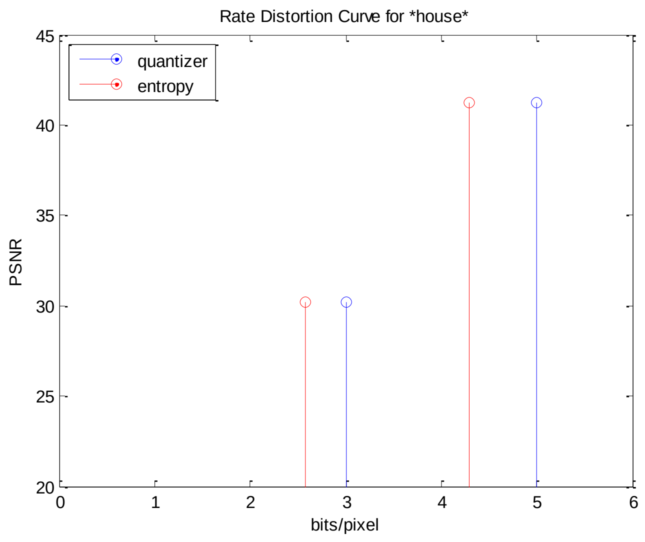
Image	**3-bit**		**5-bit**	
	PSNR	**Entropy**	**PSNR**	**Entropy**
chem	29.8752	2.650	40.9673	4.358
Couple	22.8272	1.118	39.8290	3.368
Elaine	30.4322	2.810	40.9658	4.500
F16	30.3214	2.263	40.9533	3.856
House	30.2183	2.577	41.2253	4.288
Moon	31.0697	2.176	40.9832	3.721

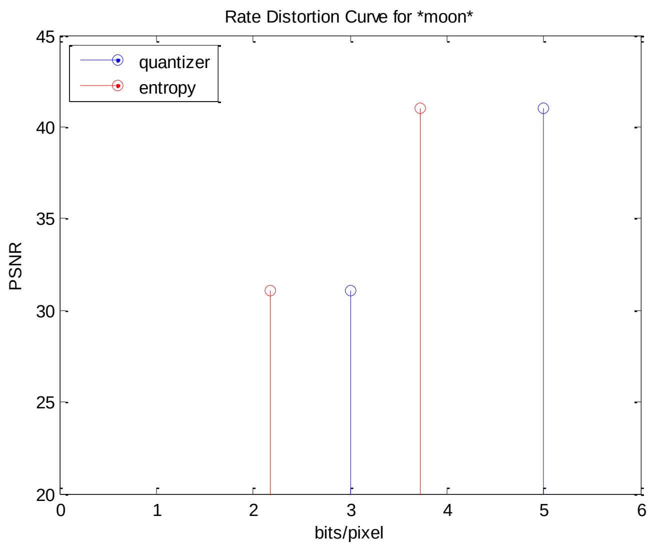












2.1.3.7 Discussion

- Lloyd-Max Scalar Quantizer designs a non-linear quantizer which adapts to the local statistics of the training files to assign the intervals and reconstruction values. So this type of design is highly dependent of the type of training data. For best results, design a quantizer for each genre of image.
- Choosing initial values for quantization in important as the training will only improve on those models. Initial values have to be chosen such that the probability of all intervals are almost the same. This type of design will give good results.
- The iterative algorithm is designed such that the PSNR is improved at each iteration and average distortion for the training data reduces at each step. Finally the algorithm converges when PSNR is almost unchanged for successive iterations.
- From the type of the training files used, the histogram is more concentrated in the center so the intervals are small towards the center. However *couple.256* file which is not used for training has its histogram concentrated in the start. Hence the PSNR for couple.256 is low compared to other images.
- For images which are not uniformly distributed (couple, f16, moon), the optimal entropy code is quite less.
- Naturally, the quality of the image improves as the number of bits used for quantization are increased.
- Summing up, the Lloyd Max scalar quantizer works best if we use a different quantizer for each genre of image.

2.2. Vector Quantization (VQ)

2.2.1. Motivation

Vector Quantization is a well-known technique for lossy data compression, pattern recognition and density estimation. Unlike Scalar quantization, VQ groups source data into vectors. So instead of looking at one value, if we look at the data from a higher dimension, it gives a better fit to the quantizer structure.

In this report, we analyze two methods used for vector quantization, Standard vector quantization and Tree structured vector quantization.

2.2.2. Approach

A vector quantizer Q of dimension K and size N is a mapping

$Q: R^k \rightarrow C$

Where: $C = \{y_1, y_2 \ldots y_n\}$ is a codebook of size N

And $y_1 \ldots y_N \in R^k$

The N-size codebook is generated by partitioning R^k into N disjoint regions. Thus if a particular codeword falls in a particular region R_i it is encoded with its corresponding codeword from the codebook table.

The whole process of encoding and decoding can be explained from the following figure.

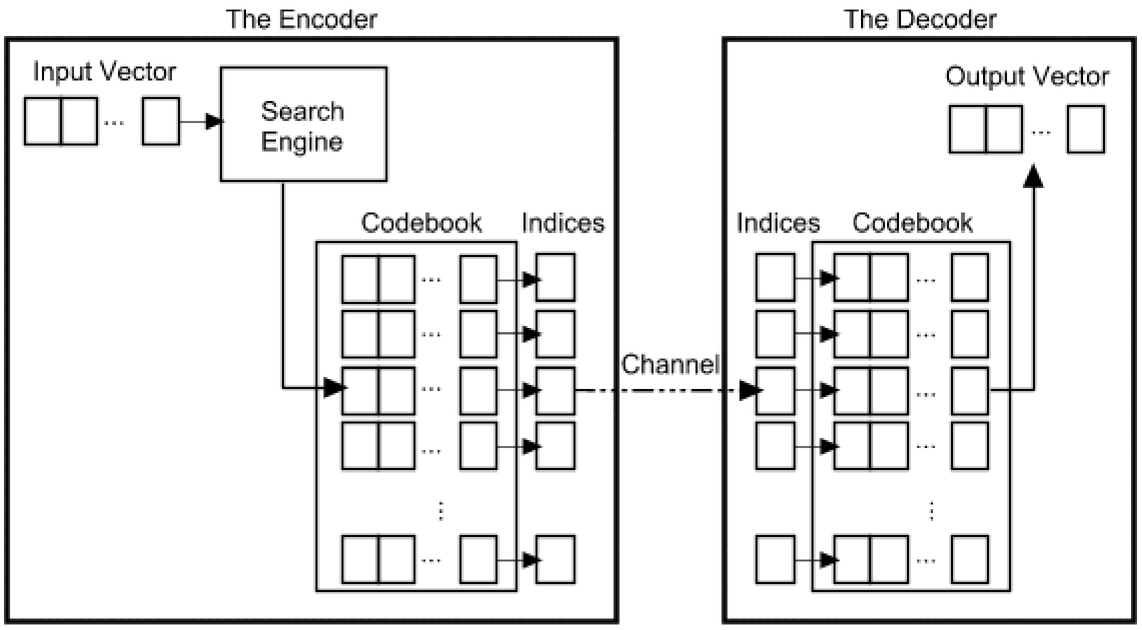


Image Ref: http://www.mqasem.net/vectorquantization/vq.html

The main problem involved in the vector Quantization is to create the codebook. There are multiple methods to create the codebook. We analyze the Standard Vector Quantization and Tree Structured Vector Quantization (TSVQ) method. The source code for SVQ and TSVQ has been provided. We use the source code to quantize the images and observe its quality.

A. Image Blocking

The source code provided for TSVQ and SVQ scan input as continuous streams of size N (2x2, 4x4, 8x8 …). We need to arrange the data as blocks of size N before feeding it to the source code.

E.g. if the image data had following pixel values

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

The file stores the data as continuous stream 1,2,3,4,5,6,7,8…16

Reordering of data (for block of size N = 4 (2x2)) is done as follows:

First read the positions (1,1),(1,2),(2,1),(2,2) and store the data, then go along the columns and then the rows.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1,2,5,6 is read

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

3,4,7,8 read

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

9,10,13,14 read

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

11,12,15,16 read

The modified file will store data as 1,2,5,6,3,4,7,8,9,10,13,14,11,12,15,16.

The source code will now read the data as 1,2,5,6 (First block) 3,4,7,8 (Second block) and so on which is the format of vectors for VQ.

B. Standard VQ

Training

Generalized Lloyd algorithm is used for creating the codebook which involves following steps:

1. Find the centroid of all the vectors.
2. Split the centroid into two.
3. Create clusters around new centroid based on Euclidian distance.
4. Computing new centroids for the new clusters.
5. Repeat clustering and finding new centroids until convergence reached.

This figure explains the clustering process involved in SVQ

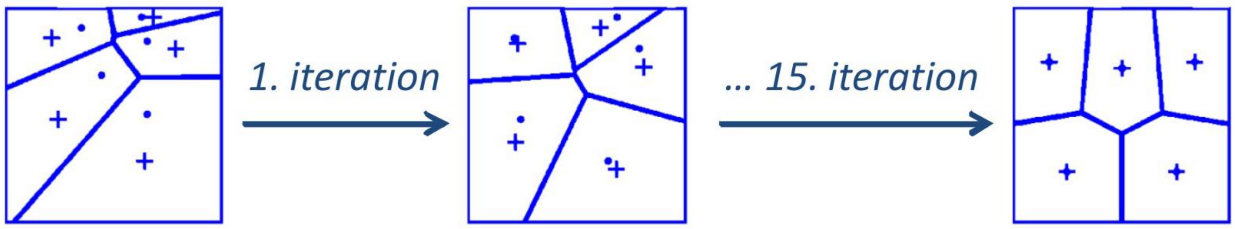


Image Ref: http://www.nt.tuwien.ac.at/uploads/media/LloydQuantizationThesis-Mayer.pdf

Training set for SVQ contains 3 images

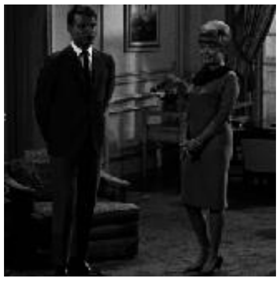
- chem.256
- house.256
- moon.256

Testing set consists contains 3 images

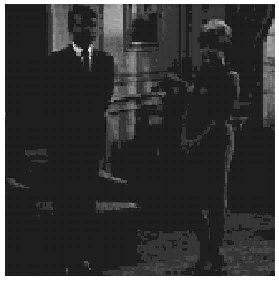
- f16.256
- couple.256
- elaine.256

Images have been tested using SVQ for block sizes of 1(2x2), 2(4x4) and 3(8x8) and codebook size of 16,32,64 and their quantized images have been compared.

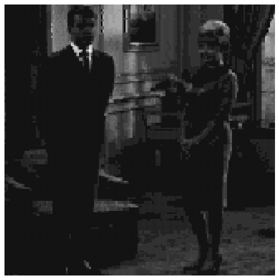
Original Image



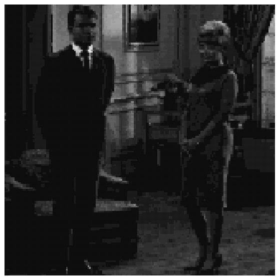
codebook size = 16, blockSize = 1



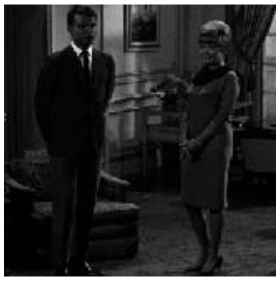
codebook size = 32, blockSize = 1



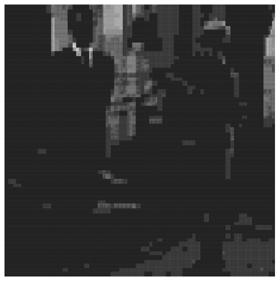
codebook size = 64, blockSize = 1



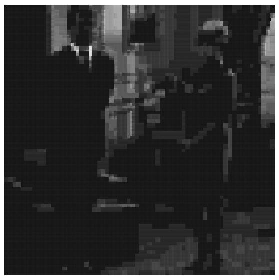
Original Image



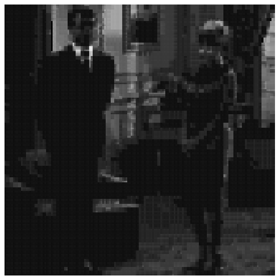
codebook size = 16, blockSize = 2



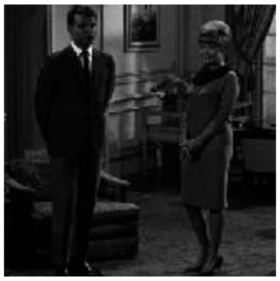
codebook size = 32, blockSize = 2



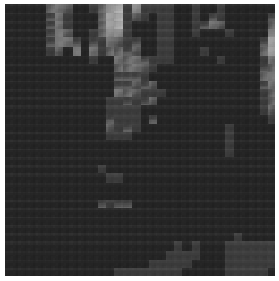
codebook size = 64, blockSize = 2



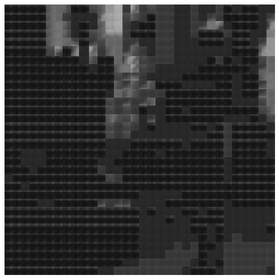
Original Image



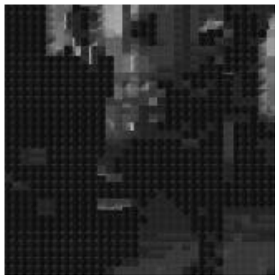
codebook size = 16, blockSize = 3



codebook size = 32, blockSize = 3

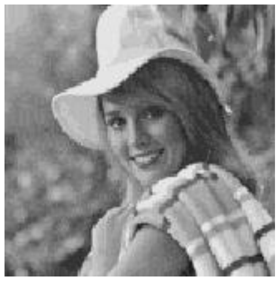


codebook size = 64, blockSize = 3

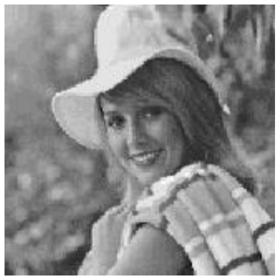


Original Image

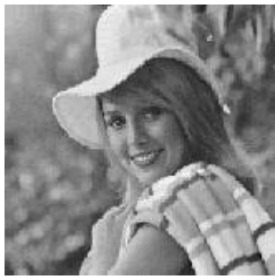
codebook size = 16, blockSize = 1



codebook size = 32, blockSize = 1

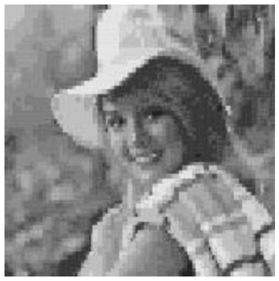


codebook size = 64, blockSize = 1

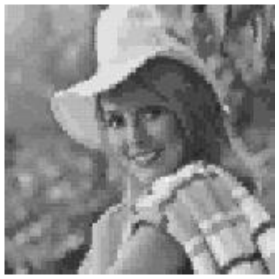


Original Image

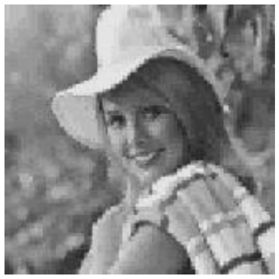
codebook size = 16, blockSize = 2



codebook size = 32, blockSize = 2

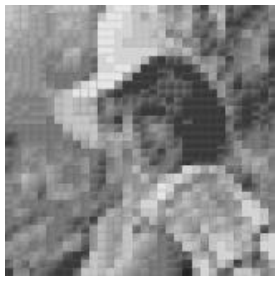


codebook size = 64, blockSize = 2

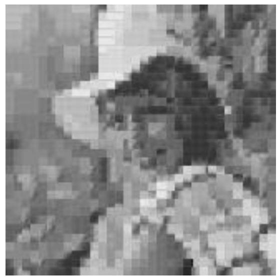


Original Image

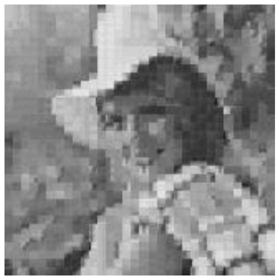
codebook size = 16, blockSize = 3



codebook size = 32, blockSize = 3

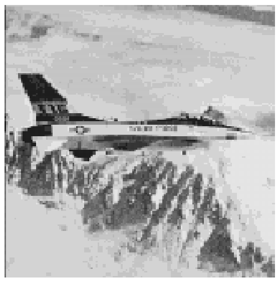


codebook size = 64, blockSize = 3

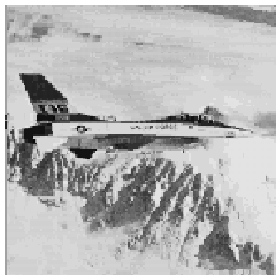


Original Image

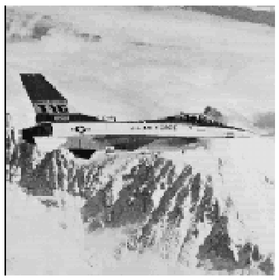
codebook size = 16, blockSize = 1



codebook size = 32, blockSize = 1

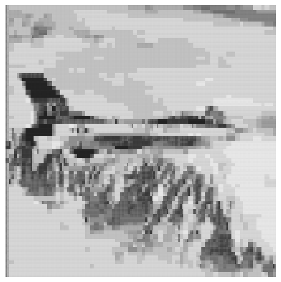


codebook size = 64, blockSize = 1

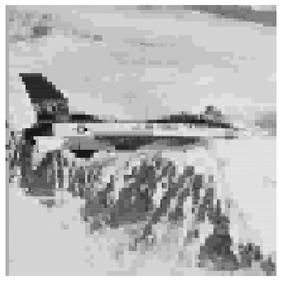


Original Image

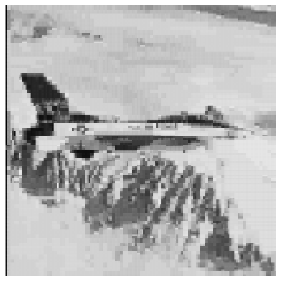
codebook size = 16, blockSize = 2



codebook size = 32, blockSize = 2

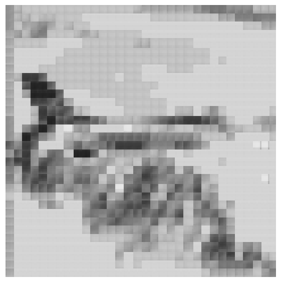


codebook size = 64, blockSize = 2

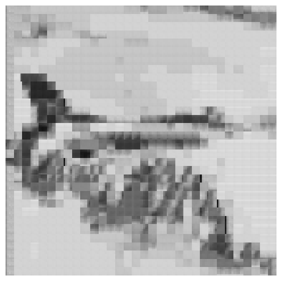


Original Image

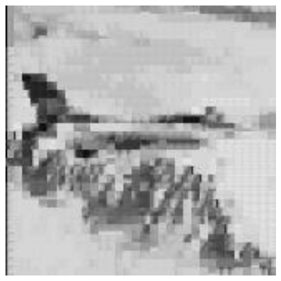
codebook size = 16, blockSize = 3



codebook size = 32, blockSize = 3



codebook size = 64, blockSize = 3



RESULTS for SVQ

EE (Empirical Entropy)
AD (Average Distortion)

Image		**Code book size = 16**		
		1 (2x2)	**2 (4x4)**	**3 (8x8)**
elaine	**EE**	3.86	3.62	3.84
	AD	291.22	2894.68	24841.68
couple	**EE**	1.43	1.22	1.15
	AD	1295.59	8056.05	42860.43
f16	**EE**	3.11	3.03	2.99
	AD	1079.89	7533.66	57186.10

Image		**Code book size = 32**		
		1 (2x2)	**2 (4x4)**	**3 (8x8)**
elaine	**EE**	4.35	4.23	4.52
	AD	219.73	2246.71	20311.65
couple	**EE**	2.12	1.85	1.93
	AD	510.99	4094.30	35078.45
f16	**EE**	3.91	3.66	3.82
	AD	538.11	5820.54	43020.60

Image		**Code book size = 64**		
		1 (2x2)	**2 (4x4)**	**3 (8x8)**
elaine	**EE**	5.14	5.17	5.36
	AD	158.82	1821.20	18415.60
couple	**EE**	2.83	2.60	2.27
	AD	294.76	2578.48	22523.22
f16	**EE**	4.65	4.60	4.67
	AD	395.78	4842.09	39365.43

Discussion

- Increase in the codebook size improves the quality of image but also increases the bits/pixel.
- Increase in size of the block tends to degrade the quality of the image (Average distortion is very high for higher block sizes as observed from the results and image). Block size of 2x2 is optimal (It has the least average distortion).
- Since a vector (block of data) is replaced with a codeword, higher block sizes tend to have pixelated appearance.
- Empirical Entropy remains more or less same for different block size. However EE increases with increase in the codebook size.

- Vector quantization also like scalar quantization is dependent on the training data, so a single codebook for all images can give bad results for images quite different from those in the training set.
- At each step of training, SVQ compares the distance of all of the training vectors to find new clusters. Hence the training process is slow.

C. Tree Structured Vector Quantization (TSVQ)

Training

Training process for the TSVQ is same as SVQ except that once the region is partitioned, only the samples in that partition will be checked for creating the new centroid.

Stopping criterion is the number of levels (bits/pixel)

The following figure illustrates the procedure.

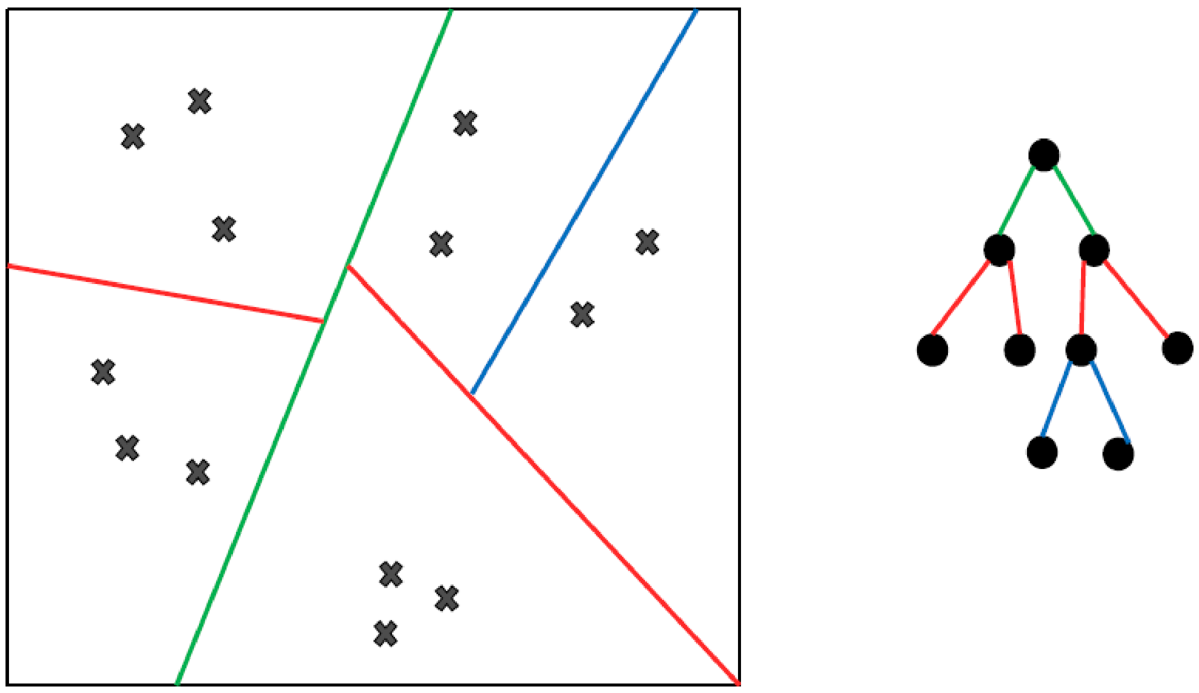


Image Ref: Discussion Slides.

Training set for SVQ contains 3 images

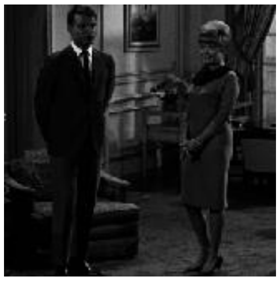
- chem.256
- house.256
- moon.256

Testing set consists contains 3 images

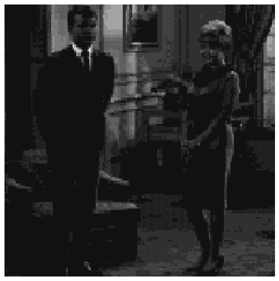
- f16.256
- couple.256
- elaine.256

Images have been tested using TSVQ for block sizes of 1(2x2), 2(4x4) and 3(8x8) and bitrate of 4.5 and 6 bits/vector and their quantized images have been compared.

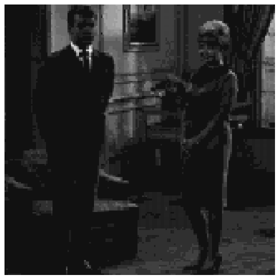
Original Image



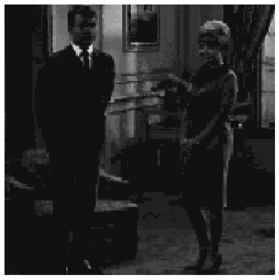
bits/Vector = 4, blockSize = 1



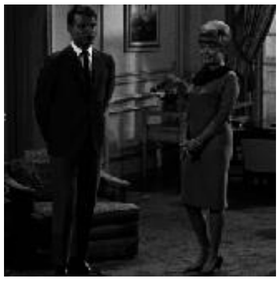
bits/Vector = 5, blockSize = 1



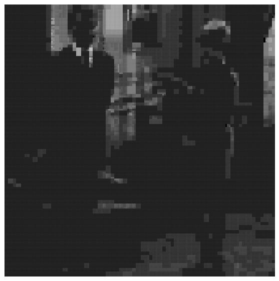
bits/Vector = 6, blockSize = 1



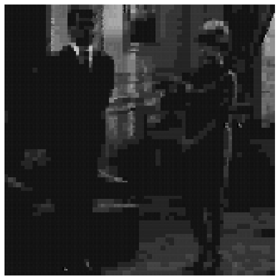
Original Image



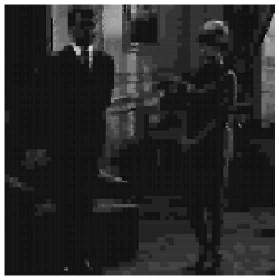
bits/Vector = 4, blockSize = 2



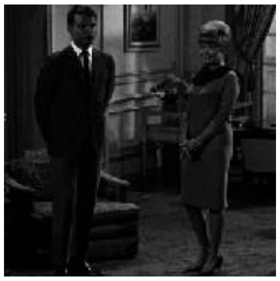
bits/Vector = 5, blockSize = 2



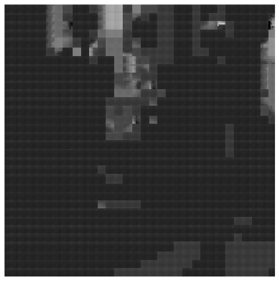
bits/Vector = 6, blockSize = 2



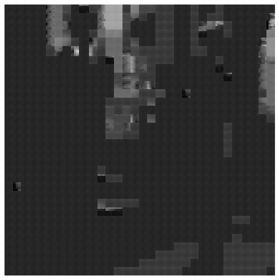
Original Image



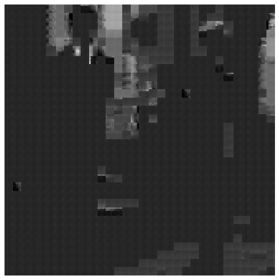
bits/Vector = 4, blockSize = 3



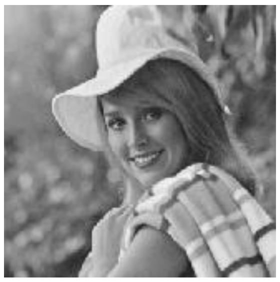
bits/Vector = 5, blockSize = 3



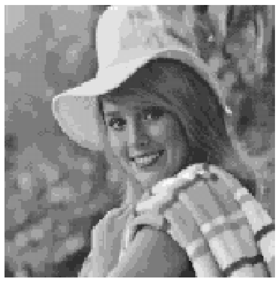
bits/Vector = 6, blockSize = 3



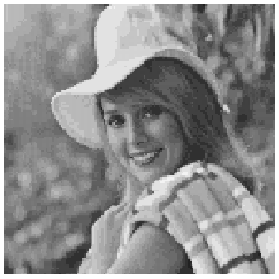
Original Image



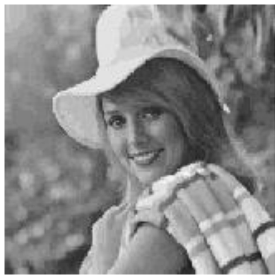
bits/Vector = 4, blockSize = 1



bits/Vector = 5, blockSize = 1

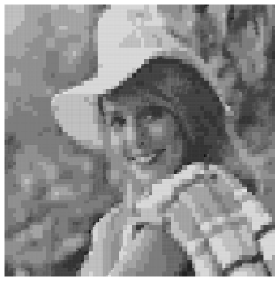


bits/Vector = 6, blockSize = 1

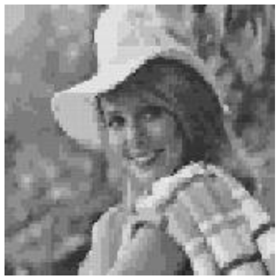


Original Image

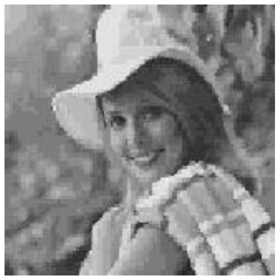
bits/Vector = 4, blockSize = 2



bits/Vector = 5, blockSize = 2

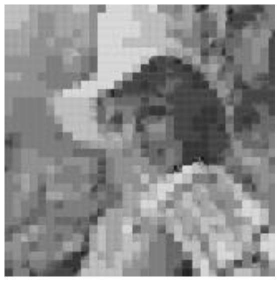


bits/Vector = 6, blockSize = 2

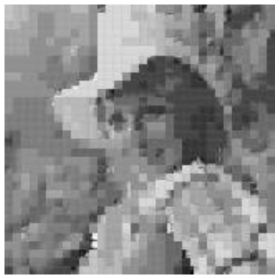


Original Image

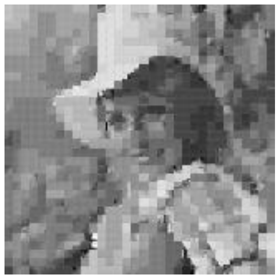
bits/Vector = 4, blockSize = 3



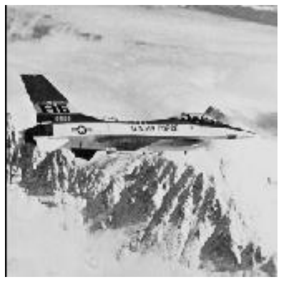
bits/Vector = 5, blockSize = 3



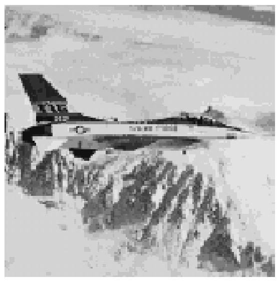
bits/Vector = 6, blockSize = 3



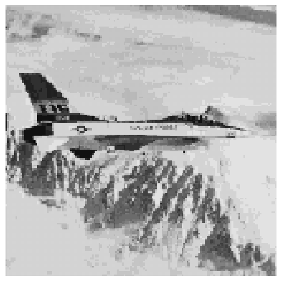
Original Image



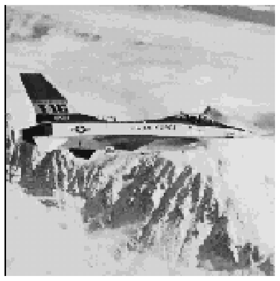
bits/Vector = 4, blockSize = 1



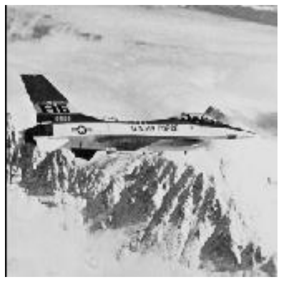
bits/Vector = 5, blockSize = 1



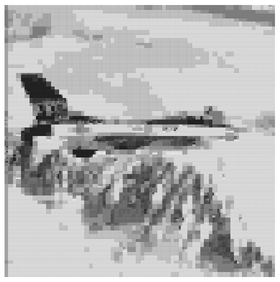
bits/Vector = 6, blockSize = 1



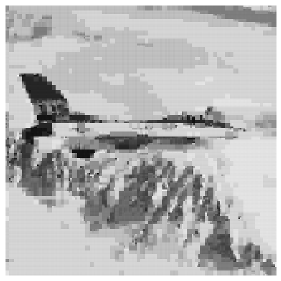
Original Image



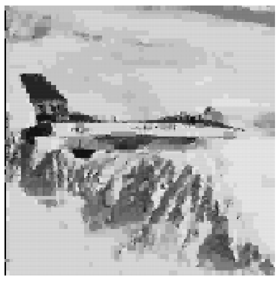
bits/Vector = 4, blockSize = 2



bits/Vector = 5, blockSize = 2

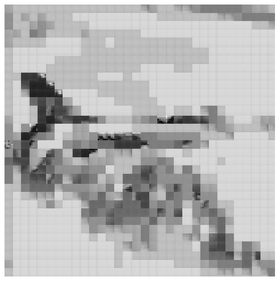


bits/Vector = 6, blockSize = 2

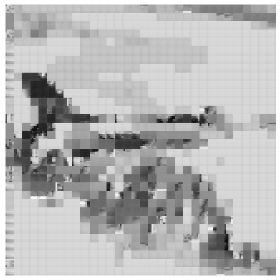


Original Image

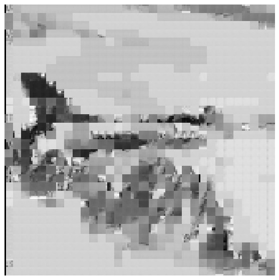
bits/Vector = 4, blockSize = 3



bits/Vector = 5, blockSize = 3



bits/Vector = 6, blockSize = 3



RESULTS for TSVQ

EE (Empirical Entropy)
AD (Average Distortion)

Image		**Bitrate = 4 bits/vector**		
		1 (2x2)	**2 (4x4)**	**3 (8x8)**
elaine	**EE**	3.47	3.52	3.53
	AD	320.86	3033.64	25744.39
couple	**EE**	2.12	1.52	1.33
	AD	467.02	7053.94	41136.26
f16	**EE**	3.17	2.89	2.80
	AD	792.03	8879.61	57038.21

Image		**Bitrate = 5 bits/vector**		
		1 (2x2)	**2 (4x4)**	**3 (8x8)**
elaine	**EE**	4.17	4.01	4.10
	AD	229.07	2466.86	23565.29
couple	**EE**	2.19	2.15	1.46
	AD	448.43	3044.46	40487.76
f16	**EE**	3.64	3.70	3.49
	AD	551.80	6019.73	49179.70

Image		Bitrate = 6 bits/vector		
		1 (2x2)	**2 (4x4)**	**3 (8x8)**
elaine	**EE**	4.41	4.63	4.64
	AD	180.80	2063.08	21376.4
couple	**EE**	2.51	2.19	1.61
	AD	393.96	2955.48	39823.10
f16	**EE**	4.03	4.22	4.22
	AD	342.72	5049.86	42868.71

Discussion

- Increase in the bitrate improves the quality of image but also increases the bits/pixel.
- Increase in size of the block tends to degrade the quality of the image (Average distortion is very high for higher block sizes as observed from the results and image). Block size of 2x2 is optimal (It has the least average distortion).
- Since a vector (block of data) is replaced with a codeword, higher block sizes tend to have pixelated appearance.
- Empirical Entropy remains more or less same for different block size. However EE increases with increase in the codebook size.
- Vector quantization also like scalar quantization is dependent on the training data, so a single codebook for all images can give bad results for images quite different from those in the training set.
- At each step of training, TSVQ compares the distance of the training vectors in that cluster only to find new clusters. Hence the training process is relatively fast.

SVQ	TSVQ
Training involves all the samples at each iteration	Training involves samples in a particular cluster at any iteration
Training process is slow	Training process is fast
Average distortion is less	Average distortion is high
Quality of the image is better	Quality of the image slightly worse than SVQ