Exercices Image & Video Technology

Léo Moulin

January 2018

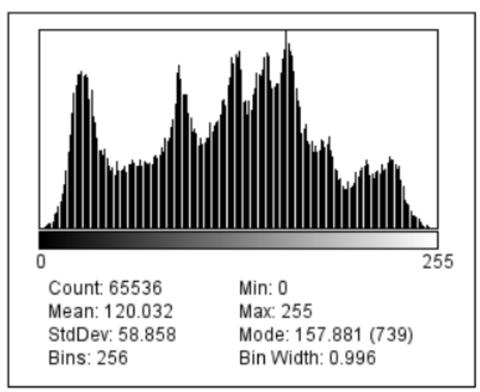
Session 1

Get Started

The original raw image of Lena contains 256x256 pixels, with a mean value of 120.032 and a standard deviation of 58.858.

The window/level allows to adjust the pixel of the image. It allows to control the lower and upper limits of the display range. In the histogram, the line (controlled by the window/level) defines the mapping between the original values and the adjusted ones. The window/level that encloses the full range of pixels is thus the case where the line is going from the bottom corner left to the upper corner right.

(a)

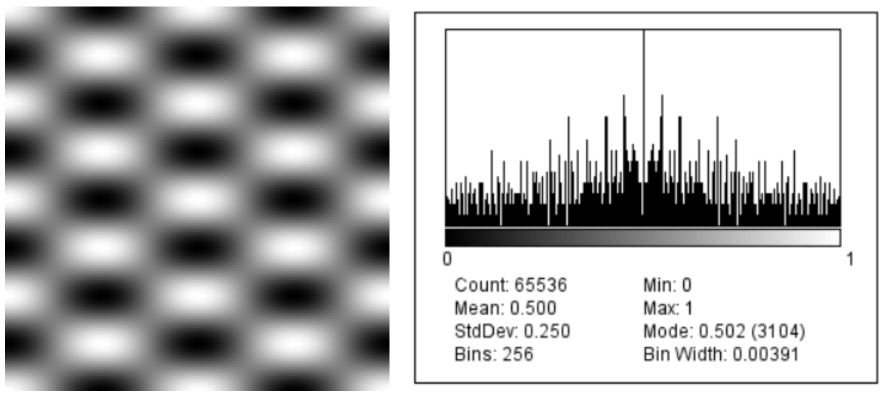


(b)

Figure 1: (a) Original 256x256 grayscale image (b) Histogram of the original image

Create and Store RAW 32pp Grayscale Images

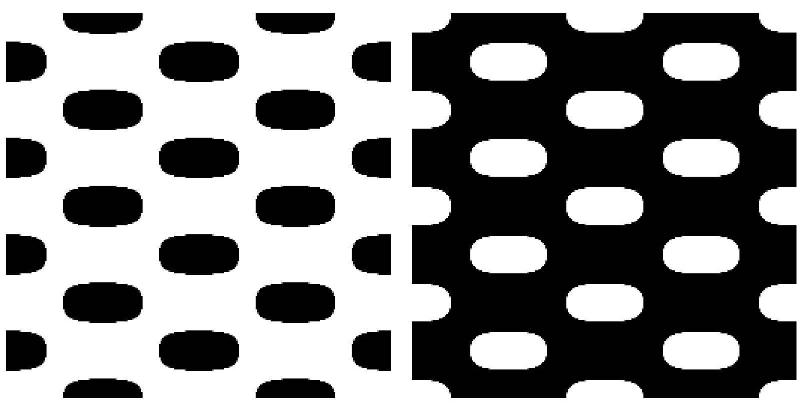
The **cosines image** is created with $I(x, y) = 1/2 + 1/2cos(x\pi/32)cos(y\pi/64)$. As each pixel is encoded as 32 bits values, the size of the image should be of 256x256x4 bytes = 262144 bytes. As 1 kilobyte = 1024 bytes, the size of the image is 256KB, which can be verified in the properties of the obtained image.



(a) (b)

Figure 2: (a) Cosines image 256x256 (b) Histogram of the cosines image

In Figure 3, as the winodw/level line is vertical, all the pixels at the left of the line are mapped to the minimum value (white), while all the values at the right of the line are mapped to the maximum value (black).



(a) (b)

Figure 3: Examples of adjustement with window/level (a) Minimum size of window and level = 90 (b) Minimum size of window and level = 150

Generate Uniform and Gaussian-distributed Random Images

The MSE has been calculated as :

$$MSE = 1/(W * H) \sum_{i=1}^{W} \sum_{j=1}^{H} (image1[i][j] - image2[i][j])^2 \quad (1)$$

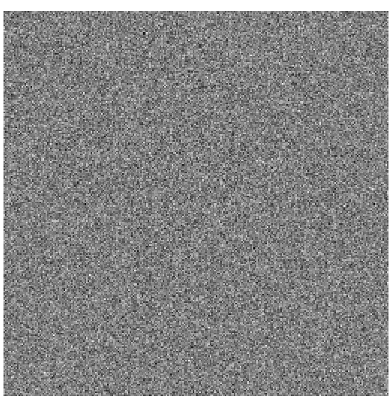
The PSNR is calculated as

$$PSNR = 10 log_{10}(\frac{Max * Max}{MSE}) \quad (2)$$

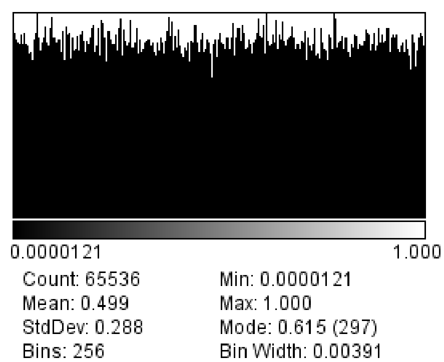
where the maximum value of the pixels is 1.
Next, the MSE and the PSNR between the original image and the noise is computed for various standard deviatioon of the noise.

Standard deviation	MSE	PSNR [dB]
0.1	0.0099672	20.0143
0.2	0.0398688	13.9937
0.3	0.0897048	10.4718
0.5	0.24918	6.03487

The MSE between the uniformly random image and the mean image (of value 0.5) is 0.083227 and the corresponding PSNR is 10.7974. The value of the noise variance that matches the MSE of the uniform random image is 0.288, which in fact the standard deviation of the uniform random image.



(a)



(b)

Figure 4: (a) Uniformly Random 256x256 image (b) Histogram of the random image

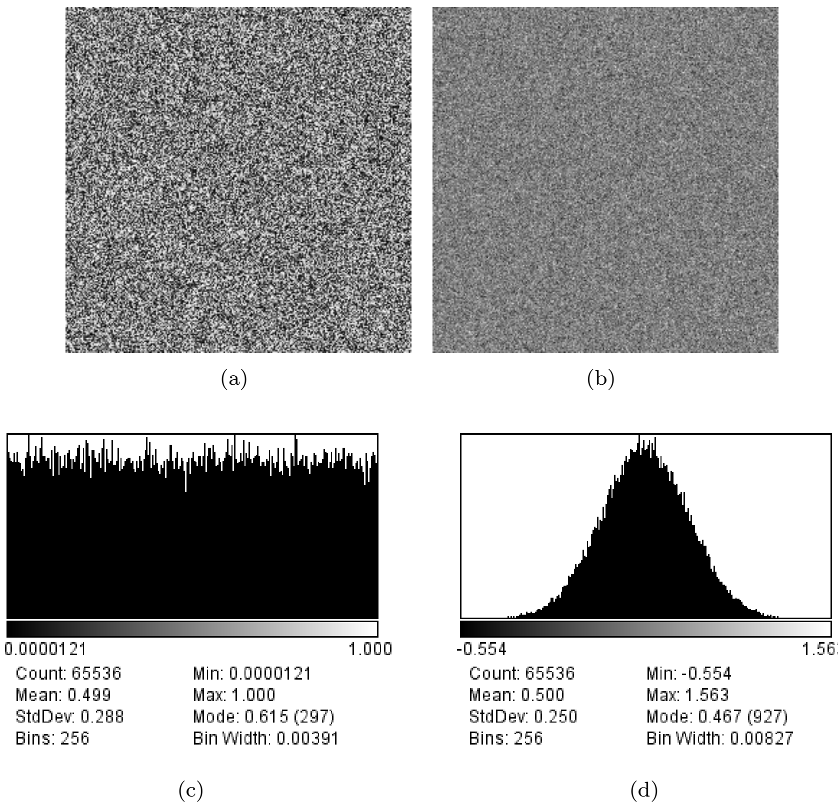


Figure 5: Comparison of configurations with the same PSNR (a) Uniformly Random 256x256 image (b) Gaussian noise of variance 0.25 (c) Histogram of the random (d) Histogram of the gaussian

Session 2

Additive Gaussian Noise

In Figure 6, the impact of the noise for various noise variances on the original image is compared qualitatively. The values of the PSNR are given below.

Standard deviation of the noise	PSNR [dB]
0.01	40.0143
0.05	26.0349
0.1	20.0145
0.3	10.4718

Figure 6: Comparison of the effect of the additive Gaussian noise on the original image. (a) Standard deviation = 0.01 (b) Standard deviation = 0.05 (c) Standard deviation = 0.1 and (d) Standard deviation = 0.3

Blur and Sharpen With 3x3 Kernel Convolution

The blur kernel has been computed as follows :

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma^2}} \tag{3}$$

The impact of the convolution of the 3x3 blur kernel on the original image is pictured in Figure 7 for different values of the variance of the Gaussian. The difference between $\sigma = 1$ and the others is visible, while the difference between $\sigma = 0.1$, $\sigma = 0.3$ and $\sigma = 0.5$ is very difficult to see. This can be explained by the size of the kernel. As the size is 3x3, the convolution only operates on 3x3 regions, thus the average values are more in adequation with the original value. If the kernel had a larger size, the blur could have been better observed.

For same PSNR, the blurred image looks better than the noisy image at naked eyes.
The values of the PSNR between the original image and the blurred image for

Figure 7: Comparison of the effect of the blur on the original image. (a) Sigma = 0.1 (b) Sigma = 0.3 (c) Sigma = 0.5 and (d) Sigma = 1.

different σ is depicted below.

Sigma of the blur	PSNR [dB]
0.1	28.036
0.3	28.033
0.5	27.758
1	26.688

Lastly, the blurred image is sharpened with the unsharp masking method. The final PSNR between the original image and the blurred (variance = 0.5) then sharpened image is increased to $PSNR = 35.7175dB$.

Figure 8: Blurred image + unsharp masking

Image Capture artifacts

In Figure 9, we first apply the blur operation to the noisy image. In the second image, we add the gaussian noise to the blurred image. The results show that the PSNR is larger for the case noise + blur. As the blur is used to cancel out the noise, it must be apply after the noise.

Operations	PSNR [dB]
Blur + noise	20.9215
Noise + blur	23.7484

(a) (b)

Figure 9: Comparison of the order of the operations (a) Blur + noise (b) Noise + blur

Session 3 & 4

Discrete cosine transforms (DCT)

For the Discrete Cosine Transform, most of the signal information tends to be concentrated in a few low-frequency components of the DCT. The gain of the DC coefficient is given by the coefficient (0,0), the lowest in frequency of the DCT.
The Figure 12 shows if the DCT coefficients are higher or lower (in absolute value) that the threshold (th = 10). It shows that the DCT coefficients are concentrated in the low frequencies.

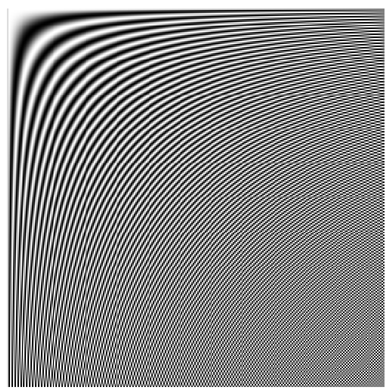
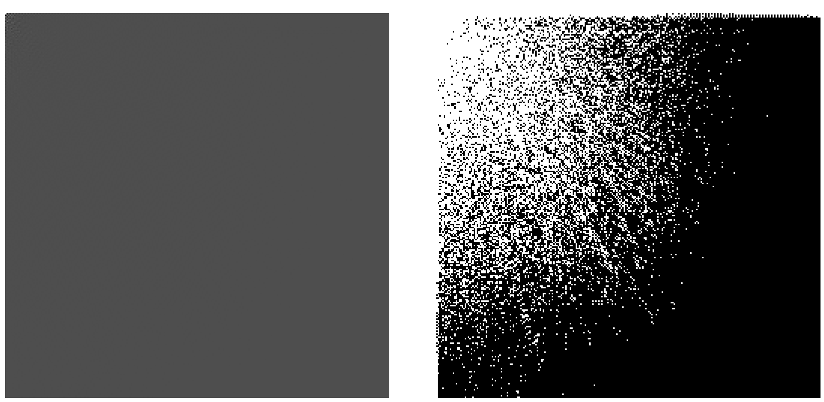


Figure 10: 2D Basis vectors



(a) (b)

Figure 11: (a) DCT coefficents (b) DCT coefficients with threshold = 10

Next, the 2 dictionnary of the DCT coefficients and the dictionnary of the IDCT coefficients are displayed in Figure 12.

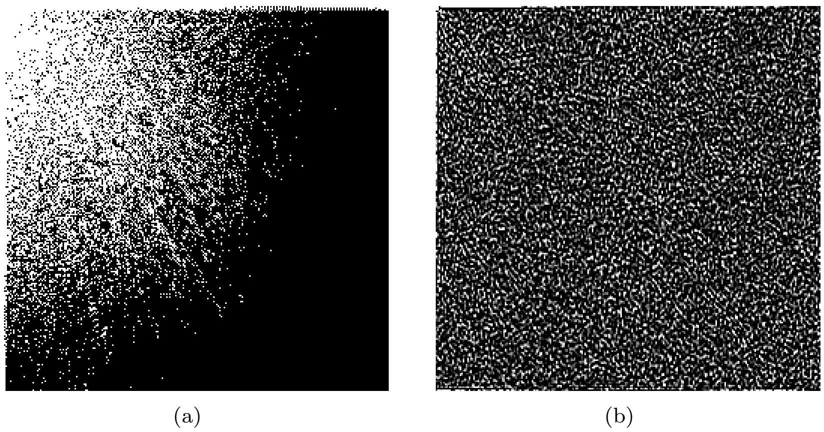


Figure 12: DCT and IDCT dictionnaries (window/level for visualization) (a) DCT (b) IDCT

Next, the PSNR between the original image and the DCT+IDCT one is computed for different value of the threshold. The values (in absolute value) below the threshold will be canceled out. Higher is the threshold, higher is the number of coefficients of the DCT canceled out. The PSNR decreases with the threshold. The results are shown in Figure 13 and 14.

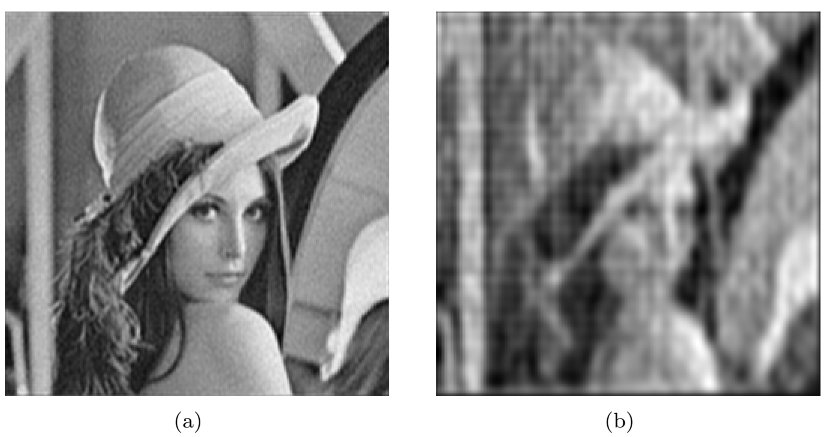


Figure 13: Reconstructed image from DCT (a) threshold = 0.1 (b) threshold = 1

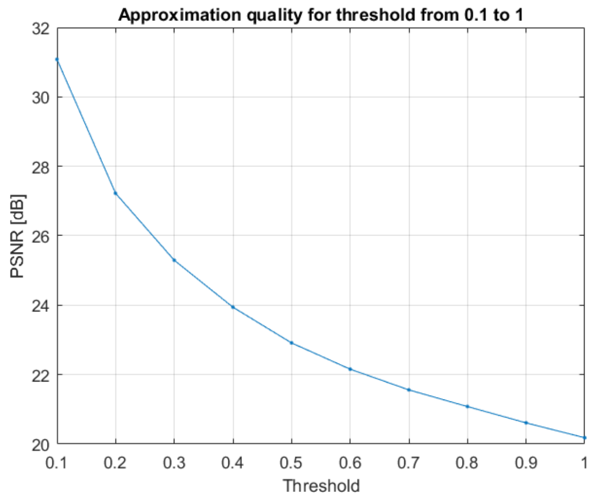


Figure 14: PSNR for threshold from 0.1 to 1

Lossy JPEG image approximation

The quantization matrix (Figure 15 and 16) is designed to provide more resolution to lower frequencies components over higher frequency components. In addition, it transforms as many components to 0, which can be encoded with greatest efficiency.
The quantization procedure is done so :

$$z_{kl} = round(\frac{y_{kl}}{q_{kl}}) \tag{4}$$

where y_{kl} is a 8x8 block of the normalized image, q_{kl} is the quantization matrix, z_{kl} is the 8x8 block quantized. $k, l = \{0, .., 7\}$

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Figure 15: Quantization matrix

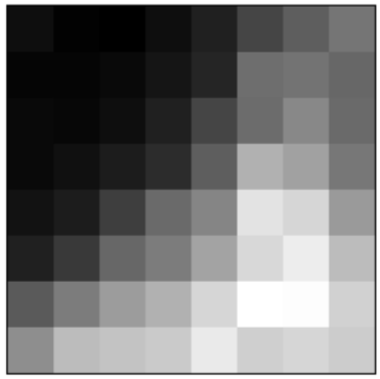
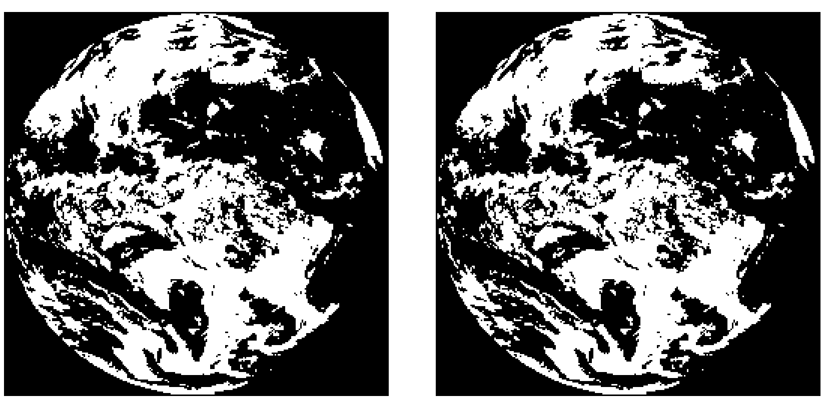


Figure 16: Quantization matrix coefficients

Session 5 & 6

Bilevel Images and Run-length encoding (RLE)

The RLE scheme is implemented in this section. The RLE scheme consists of transforming a bilevel image into an array of integers, each representing the run lengths of the occurence of a same value. The first bit represents the first pixel color.
The PSNR between the original bilevel image and the encoded + decoded image is equal infinity, which verifies that the compression is indeed lossless.



(a) (b)

Figure 17: (a) Original bilevel image (b) Encoded + Decoded image

Discrete probability density functions (PDF)

In this section, an array P with the number of occurences for each run lengths is created. The longest run length N is simply obtained by taking the maximum value of the RLE encoded array. The number of runs M is obtained by taking the number of elements of the RLE encoded array.
With the values of N and M, we can create the probability density function by adding, for each run length, its occurence in the new array P. N will be the maximum value of P, as it is the longest run length. In the case of the *earth binary* file, $N = 374$ and $M = 5375$.

The entropy is calculated as $H = -\sum_{i=1}^{N} p(i)log_2(p(i)) = 4.5878bit/run$. The minimum message length in bits can be chosen equal to 1 bit. In the minimum case (entropy coding), the message length in bits is the entropy H times the number of run lengths M. For the *earth binary* file, we obtain $H * M = 24659.424bits$. We need 24660 bits in the entropy case.

Exp-Golomb variable-length code (VLC)

In this section, the *golomb()* and *inversegolomb()* are implemented.

Lossless bilevel image compression(Project)

The first step of the project consists of loading the bitlevel image. Then the image is transformed with a RLE scheme. Then, the RLE code is encoded with a Exp-golomb coding and saved in a txt file. This binary file contains $31KB = 31256bits$, meaning that we achieved a compression ratio of :

$$CR = \frac{256 * 256bits}{31256bits} = 2.09 \tag{5}$$

Huffmann encoding was also tried to increase the compression ratio, without

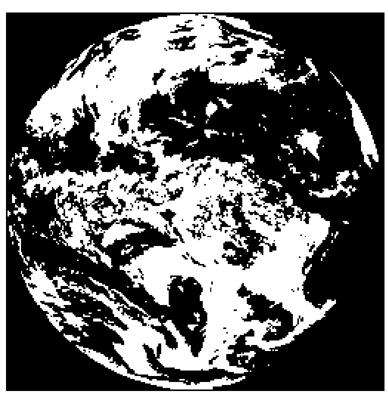


Figure 18: Bitlevel input image of size 256KB

success.