

ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

Εργασία 4

ΠΑΝΤΕΛΕΗΜΩΝ ΠΡΩΙΟΣ

ice18390023

9ο Εξάμηνο

ice18390023@uniwa.gr

Πέμπτη 16:00-19:00

Ομάδα χρηστών 18



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

Υπεύθυνοι καθηγητές

ΕΥΦΡΟΣΥΝΗ ΔΟΥΤΣΗ

Τμήμα Μηχανικών και Πληροφορικής Υπολογιστών
28 Φεβρουαρίου 2023

Περιεχόμενα

1	Uniform scalar quantizer	1
2	Haar transform	4

Κώδικες

Κατάλογος σχημάτων

1.1	Χαρακτηριστικές	1
1.2	Original cameraman image	2
1.3	Διαφορετικά επίπεδα κβάντισης	3
1.4	MSE για τα διαφορετικά επίπεδα κβάντισης	4

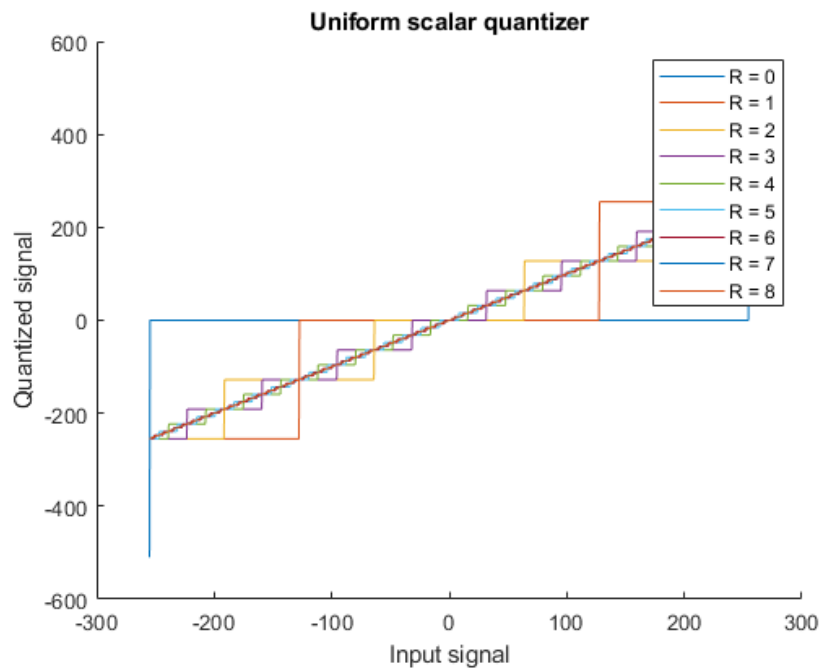
1 Uniform scalar quantizer

Με την χρήση της ακόλουθης συνάρτησης, επιτυγχάνουμε uniform scalar quantization.

```
1 function [quantized_signal] = uni_scalar(input_signal, R, A)
2 % input_signal: input signal to be quantized
3 % R: num of bits
4 % A: max amplitude
5
6 L = 2^R;
7 delta = 2*A/L;
8
9 quantized_signal = delta * sign(input_signal) * floor((abs(input_signal)/delta) + (1/2));
10
11 end
```

Έστω ένα γραμμικό σήμα $f(x)$, $x \in [-255, 255]$, τότε οι χαρακτηριστικές για κάθε R στο εύρος $[0,8]$ υπολογίζονται με τον αντίστοιχο κώδικα:

```
1 A = 255;
2 x = linspace(-A, A, 1000); % linear
3
4 figure;
5 hold on;
6 for R = 0:8
7     Q = uni_scalar(x, R, A);
8     plot(x, Q, 'DisplayName', ['R = ' num2str(R)]);
9 end
10 hold off;
11 xlabel('Input signal');
12 ylabel('Quantized signal');
13 title('Uniform scalar quantizer');
14 legend('show');
```



Σχήμα 1.1: Χαρακτηριστικές

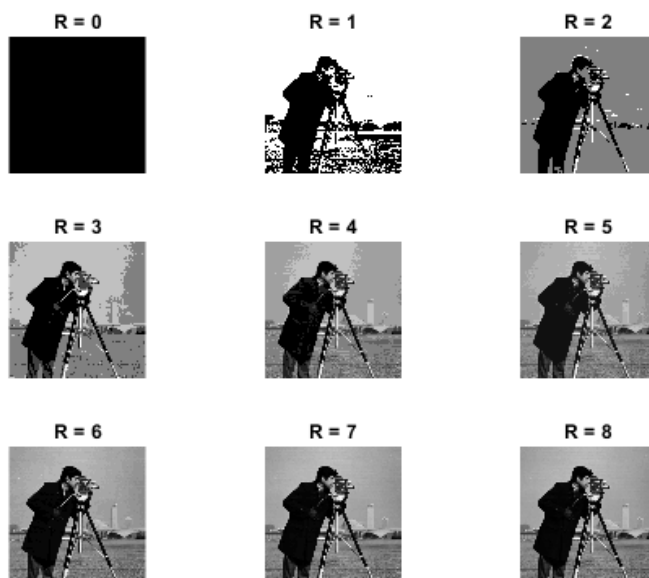
Ο κβαντισμός της εικόνας cameraman (εικ. 1.2) γίνεται με τον εξής κώδικα:

```
1 input_image = imread('cameraman.tif');
2 A = 255;
3
4 figure;
5 imshow(input_image, []);
6 title('Original image');
7
8 figure;
9 subplot(3, 3, 1);
10
11 for R = 0:8
12     quantized_image = uni_scalar(double(input_image), R, A);
13     subplot(3, 3, R+1);
14     imshow(quantized_image, []);
15     title(['R = ' num2str(R)]);
16 end
```

Original image



Σχήμα 1.2: Original cameraman image



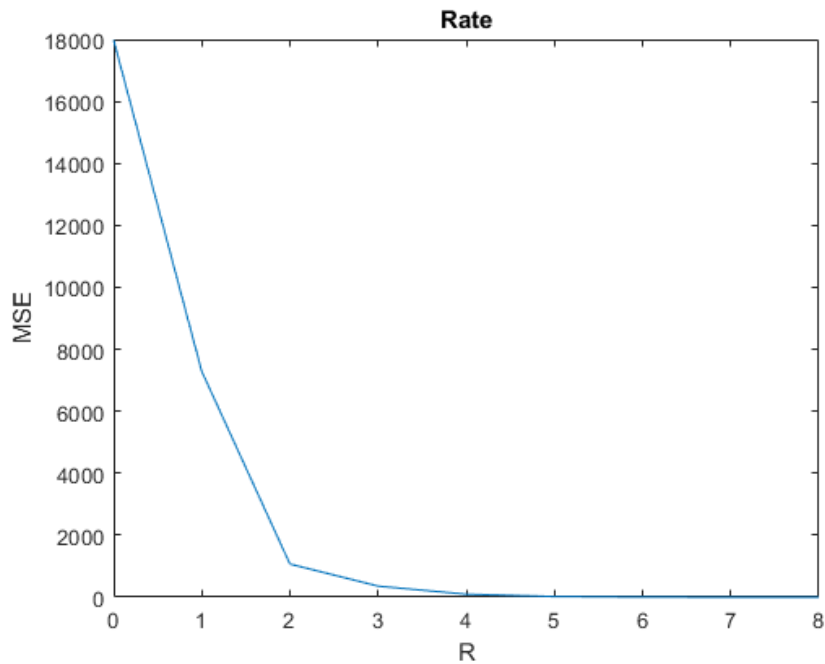
Σχήμα 1.3: Διαφορετικά επίπεδα κβάντισης

Με τον ακόλουθο κώδικα υπολογίζουμε το mse για κάθε επίπεδο κβάντισης:

```

1 input_image = double(imread('cameraman.tif'));
2 A = 255;
3
4 mse_values = zeros(1, 9);
5 for R = 0:8
6     quantized_image = uni_scalar(input_image, R, A);
7     mse_values(R+1) = mean((input_image(:)-quantized_image(:)).^2);
8 end
9
10 figure;
11 plot(0:8, mse_values);
12 xlabel('R');
13 ylabel('MSE');
14 title('Rate')

```



Σχήμα 1.4: MSE για τα διαφορετικά επίπεδα κβάντισης

Παρατηρούμε πως για $R=4$, έχουμε ικανοποιητικά αποτελέσματα με πάρα πολύ μικρό MSE και σχεδόν τον μισό αποθηκευτικό χώρο.

2 Haar transform

```

1 img = imread('cameraman.tif');
2
3 % apply Haar transform
4 [LL1, HL1, LH1, HH1] = dwt2(img, 'haar');
5 [LL2, HL2, LH2, HH2] = dwt2(cA1, 'haar');
6
7 % quantize the subbands
8 q_LL2 = uni_scalar(LL2, 4, 255);
9 q_HL2 = uni_scalar(HL2, 4, 255);
10 q_LH2 = uni_scalar(LH2, 4, 255);
11 q_HH2 = uni_scalar(HH2, 4, 255);
12 q_LL1 = uni_scalar(LL1, 2, 255);
13 q_HL1 = uni_scalar(HL1, 2, 255);
14 q_LH1 = uni_scalar(LH1, 2, 255);
15 q_HH1 = uni_scalar(HH1, 2, 255);
16
17 % entropy in each subband
18 entropy_LL2 = entropy(q_LL2);
19 entropy_HL2 = entropy(q_HL2);
20 entropy_LH2 = entropy(q_LH2);
21 entropy_HH2 = entropy(q_HH2);
22 entropy_LL1 = entropy(q_LL1);
23 entropy_HL1 = entropy(q_HL1);
24 entropy_LH1 = entropy(q_LH1);
25 entropy_HH1 = entropy(q_HH1);
26
27 % Total entropy

```

```
28 total_entropy = entropy_LL2 + entropy_HL2 + entropy_LH2 + entropy_HH2 + entropy_LL1 + entropy_HL1 +  
    entropy_LH1 + entropy_HH1;  
29 %total_entropy = entropy_LL2 + entropy_HL2 + entropy_LH2 + entropy_HH2 + entropy_HL1 + entropy_LH1 +  
    entropy_HH1  
30  
31 % inverse haar transform  
32  
33 recon_LL2 = idwt2(q_LL2, q_HL2, q_LH2, q_HH2, 'haar');  
34 recover_img = idwt2(recon_LL2, q_HL1, q_LH1, q_HH1, 'haar');  
35  
36 % psnr  
37  
38 original_img = im2double(img);  
39  
40 % Calculate the PSNR value  
41 psnr_value = psnr(original_img, recover_img);  
42  
43 % Display the PSNR value  
44 fprintf('PSNR value between img1 and img2: %.2f dB\n', psnr_value);
```