

ΓΑΒΡΙΗΛΙΔΗΣ ΣΟΦΟΚΛΗΣ-ΦΙΛΑΡΕΤΟΣ 2014030062
ΧΟΥΡΔΑΚΗΣ ΓΕΩΡΓΙΟΣ 2014030231

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΑΝΑΦΟΡΑ Project

Για το Part A:

1) Το σήμα εισόδου δημιουργούμε την $f(x)$:

```
fx = [];  
for i=-255:256  
    fx(i+256)=i;  
end
```

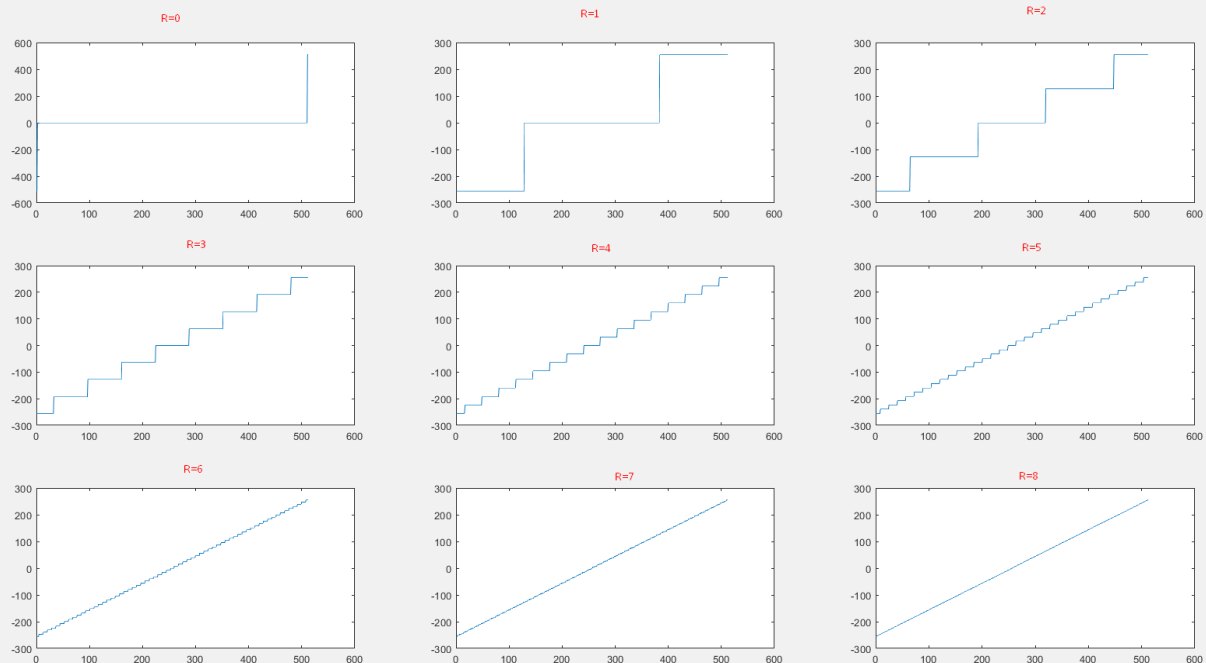
Έπειτα δημιουργούμε την συνάρτηση για την Uniform scalar quantizer

```
function usq = uni_scalar(R,A,X)  
    Level = 2^R;  
    Dlength = 2*A/Level;  
    fl = floor( (abs(X)/Dlength) + 0.5);  
    usq= Dlength*fl.*sign(X);  
end
```

2) Στην συνέχεια περνάμε το σήμα μας από την συνάρτηση που δημιουργήσαμε για R από 0 έως 8 και εμφανίζουμε τα αποτελέσματα

```
q0 = uni_scalar(0,255,fx); %l=1  
q1 = uni_scalar(1,255,fx); %l=2  
q2 = uni_scalar(2,255,fx); %l=4  
q3 = uni_scalar(3,255,fx); %l=8  
q4 = uni_scalar(4,255,fx); %l=16  
q5 = uni_scalar(5,255,fx); %l=32  
q6 = uni_scalar(6,255,fx); %l=64  
q7 = uni_scalar(7,255,fx); %l=128  
q8 = uni_scalar(8,255,fx); %l=256
```

```
figure  
subplot (3,3,1), plot (q0)  
subplot (3,3,2), plot (q1)  
subplot (3,3,3), plot (q2)  
subplot (3,3,4), plot (q3)  
subplot (3,3,5), plot (q4)  
subplot (3,3,6), plot (q5)  
subplot (3,3,7), plot (q6)  
subplot (3,3,8), plot (q7)  
subplot (3,3,9), plot (q8)
```



Στην matlab φαίνονται πολύ καλά και είναι αυτά που περιμέναμε... Εδώ χαλάνε λόγω του scaling

3)Μετά διαβάζουμε την εικόνα lena_gray_512.tif και καλούμε την συνάρτηση χρησιμοποιώντας την εικόνα.

```
i=imread("lena_gray_512.tif");

%R=0
usqlena0 = uni_scalar(0,255,i) ;
figure
imshow(usqlena0)

%R=1
usqlena1 = uni_scalar(1,255,i) ;
figure
imshow(usqlena1)

%R=2
usqlena2 = uni_scalar(2,255,i) ;
figure
imshow(usqlena2)

%R=3
usqlena3 = uni_scalar(3,255,i) ;
figure
imshow(usqlena3)

%R=4
usqlena4 = uni_scalar(4,255,i) ;
figure
imshow(usqlena4)

%R=5
usqlena5 = uni_scalar(5,255,i) ;
figure
imshow(usqlena5)

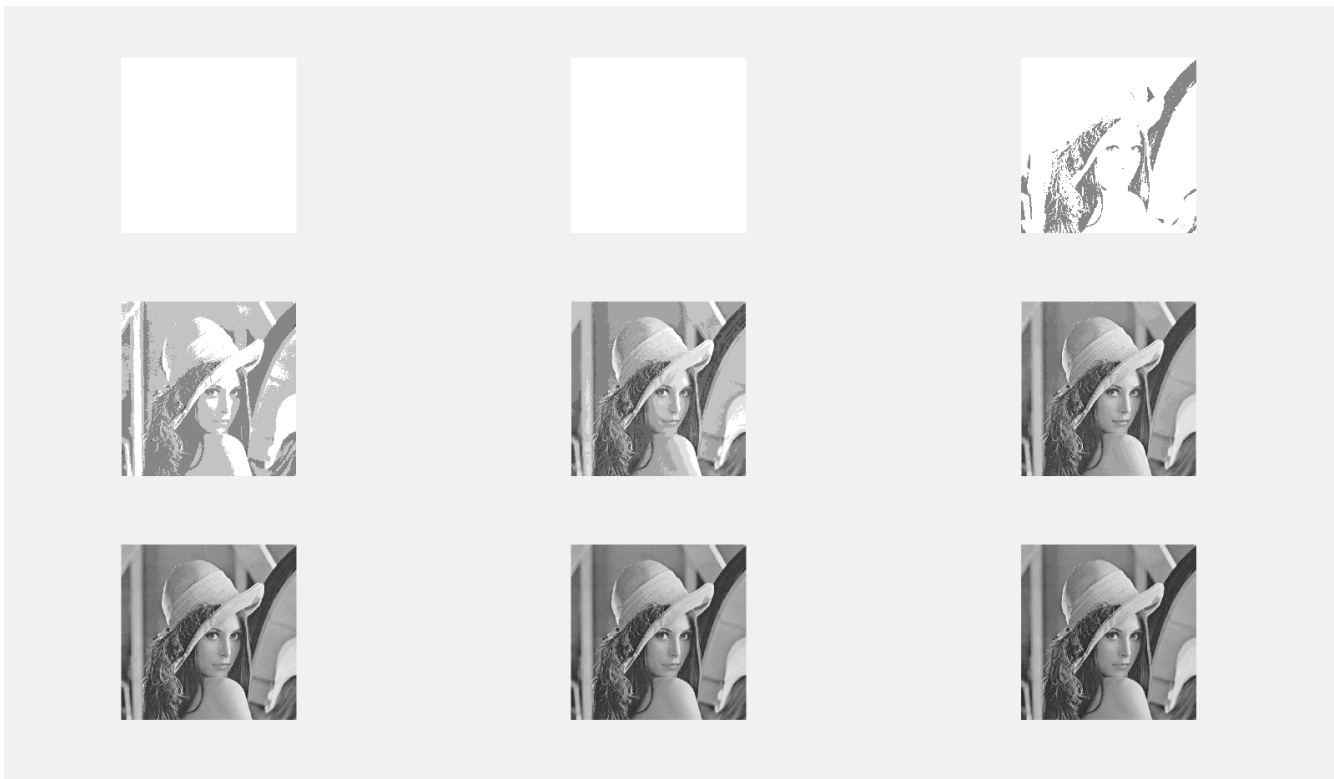
%R=6
usqlena6 = uni_scalar(6,255,i) ;
figure
imshow(usqlena6)

%R=7
usqlena7 = uni_scalar(7,255,i) ;
figure
imshow(usqlena7)

%R=8
usqlena8 = uni_scalar(8,255,i) ;
figure
imshow(usqlena8)

figure
subplot (3,3,1), imshow(usqlena0)
subplot (3,3,2), imshow (usqlena1)
subplot (3,3,3), imshow(usqlena2)
subplot (3,3,4), imshow(usqlena3)
subplot (3,3,5), imshow(usqlena4)
subplot (3,3,6), imshow (usqlena5)
```

```
subplot (3,3,7), imshow(usqlena6)
subplot (3,3,8), imshow(usqlena7)
subplot (3,3,9), imshow(usqlena8)
```



Ξεκινάει από $R=0$ έως το 8

Για το Part B:

Για το 1 και 2) Διαβάζουμε το βίντεο μας και εμφανίζουμε τις πληροφορίες του
`v = VideoReader('xylophone.mp4')`

VideoReader with properties:

General Properties:

```
Name: 'xylophone.mp4'
Path: 'C:\Users\chour\Desktop'
Duration: 4.7000
CurrentTime: 0.1333
NumFrames: 141
```

Video Properties:

```
Width: 320
Height: 240
FrameRate: 30
BitsPerPixel: 24
VideoFormat: 'RGB24'
```

Άρα τα numbers of frames είναι 141, το framerate είναι 30 , η ανάλυση του κάθε frame είναι 320 επί 240 και η διάρκεια του βίντεο είναι 4.7 seconds

3)Κάναμε ένα script το οποίο διαβάζει το κάθε frame και το εισάγει σε ένα πίνακα που περιέχει όλα τα frames:

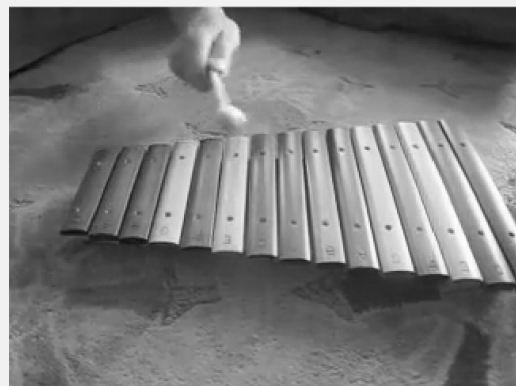
```
frames=[];  
while hasFrame(v)  
    frame = readFrame(v);  
    frames=[frames,frame];  
end  
figure  
imshow(frames)
```



Για λόγους εξοικονόμησης “χώρου” εκτυπώσαμε όλα τα frames σειριακά σε ένα plot και όχι ξεχωριστά

4)Κάναμε extract το frame 50 και το μετατρέψαμε σε ασπρόμαυρη εικόνα:

```
%B.4  
frame50 = read(v,50);  
%Grayscale  
frame50gray = rgb2gray(frame50);  
figure  
subplot (1,2,1), imshow(frame50)  
subplot (1,2,2), imshow(frame50gray)
```



Για το Part C:

1) Το frame50gray το είχα πάρει στο B4 ερώτημα

2)

```
frame50gray = imresize(frame50gray , [256 256]);
```

3) Για το 2 level decomposition χρησιμοποιούμε την συνάρτηση haart2 με όρισμα 2 που περιορίζει την συνάρτηση μόνο σε 2 επίπεδα

```
[a , h , v ,d] = haart2( frame50gray,2);
```

Για την επεξήγηση της εκχώρησης των τιμών από την συνάρτηση χρησιμοποιούμε την παρακάτω εικόνα

H7	H4	H1
H6	H5	
H3		H2

Η συνάρτηση δημιουργεί 3 cell array (h,v,d) που περιέχουν τα 6 subbands από τα 2 επίπεδα και το πίνακα a που περιέχει το τελευταίο subband του τελευταίου επιπέδου. Πιο αναλυτικά:

subband 7 → a

subband 6 → v{2}

subband 5 → d{2}

subband 4 → h{2}

subband 3 → v{1}

subband 2 → d{1}

subband 1 → h{1}

4)

```
a = uni_scalar(4,64,a);
```

```
h{2} = uni_scalar(4,64,h{2});
```

```
v{2} = uni_scalar(4,64,v{2});
```

```
d{2} = uni_scalar(4,64,d{2});
```

```
h{1} = uni_scalar(4,128,h{1});
```

```
v{1} = uni_scalar(4,128,v{1});
```

```
d{1} = uni_scalar(4,128,d{1});
```

Περνάμε το κάθε subband από την συνάρτησή μας Uniform scalar quantizer, για R=4 και τα εκχωρούμε το αποτέλεσμα ξανά στο subband για να το χρησιμοποιούμε για το reconstruct μετά το quantize

5) Για την εντροπία χρησιμοποιούμε την έτοιμη συνάρτηση της Matlab “entropy” διαφορετικά θα μπορούσαμε να κάνουμε imhist (“subband”) θα κρατούσαμε τις θετικές πιθανότητες από αυτό για να χρησιμοποιούμε το τύπο του αθροίσματος που δίνεται στην εκφώνηση

```
h1 = entropy(h{1})
h2 = entropy(d{1})
h3 = entropy(v{1})
h4 = entropy(h{2})
h5 = entropy(d{2})
h6 = entropy(v{2})
h7 = entropy(a)
```

6)Για το reconstruct της εικόνας με Inverse haar transform χρησιμοποιούμε την έτοιμη συνάρτηση της Matlab “ihaart2” στην οποία εισάγουμε τα quantized subbands και βρίσκουμε το psnr. Και υπολογίζουμε την ολική εντροπία

```
inverse2image = uint8(ihaart2(a,h,v,d));
htotal = entropy(inverse2image)
figure
imshow(inverse2image)
peaksnr1 = psnr(frame50gray , inverse2image)
```

7)Ομοίως με το c.4 κάνουμε quantize τα subbands αλλά με R=3 για το πρώτο και R=5 για το δεύτερο επίπεδο

```
[a1 , h1 , v1 ,d1] = haart2( frame50gray,2);

a1 = uni_scalar(5,64,a1);
h1{2} = uni_scalar(5,64,h1{2});
v1{2} = uni_scalar(5,64,v1{2});
d1{2} = uni_scalar(5,64,d1{2});
h1{1} = uni_scalar(3,128,h1{1});
v1{1} = uni_scalar(3,128,v1{1});
d1{1} = uni_scalar(3,128,d1{1});
```

8)Ομοίως με το c.5 βρίσκουμε τις εντροπίες του κάθε subband

```
e1 = entropy(h1{1})
e2 = entropy(d1{1})
e3 = entropy(v1{1})
e4 = entropy(h1{2})
e5 = entropy(d1{2})
e6 = entropy(v1{2})
e7 = entropy(a1)
```

9)Ομοίως με το c.6 κάνουμε το reconstruct με quantized subbands

```
inverse2image2 = uint8(ihaart2(a1,h1,v1,d1));
etotal = entropy(inverse2image2)
figure
imshow(inverse2image2)
peaksnr2 = psnr(frame50gray , inverse2image2)
```

10)Αυτά είναι τα αποτελέσματα για τα ερωτήματα 4,5,6

$$h1 = 0.3428$$

$$h2 = 0.1325$$

$$h3 = 0.5531$$

$$h4 = 0.9829$$

$$h5 = 0.6983$$

$$h6 = 0.9317$$

$$h7 = 0$$

$$h_{total} = 6.5759$$

$$peaksnr1 = 39.0941$$

Αυτά είναι τα αποτελέσματα για τα ερωτήματα 7,8,9

$$e1 = 0.1290$$

$$e2 = 0.0273$$

$$e3 = 0.3404$$

$$e4 = 0.9998$$

$$e5 = 0.8722$$

$$e6 = 0.9818$$

$$e7 = 0$$

$$e_{total} = 7.5663$$

$$peaksnr2 = 36.3529$$

Όσο μεγαλύτερη η εντροπία τόσο μεγαλύτερη η τυχαιότητα για τα textures της εικόνας . Όπως είναι λογικό με όσο μεγαλύτερο R έχουμε κάνει quantize τόσο μικρότερη η τυχαιότητα. Συνεπώς μεγαλύτερη εντροπία. Για αυτό βλέπουμε το δεύτερο επίπεδο να έχει μικρότερη εντροπία όταν έγινε με R=5 ενώ το 1ο που έγινε με R=3 να έχει μεγαλύτερη εντροπία.

Όσο μεγαλύτερο είναι το peak signal to noise ratio τόσο μικρότερος είναι ο θόρυβος και πιο πιστή είναι η εικόνα μας στην αρχική. Άρα η μέθοδος με R=4 είναι πιο αποδοτική

