

Ψηφιακή Επεξεργασία Εικόνας

ΑΝΑΦΟΡΑ PROJECT

Ιωαννίδης Χρήστος 2018030006
Σταθοπούλου Στέλλα 2018030208

ΜΕΡΟΣ Α:

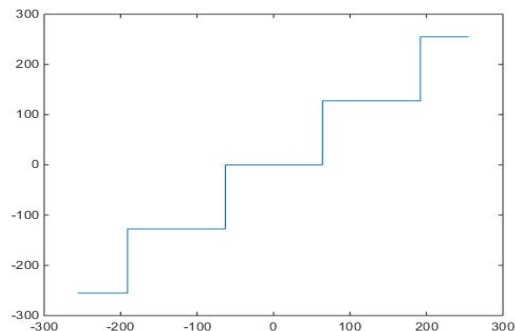
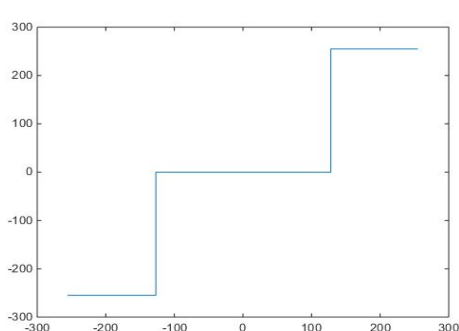
Συνάρτηση Uni_Scalar:

Η συνάρτηση αυτή κατασκευάστηκε σύμφωνα με τον ορισμό του Uniform Scalar Quantizer. Συγκεκριμένα, δέχεται σαν ορίσματα το εύρος τιμών του σήματος, τον αριθμό των bits μετά την κβάντιση (R), καθώς και τον αριθμό A. Τα στοιχεία αυτά είναι απαραίτητα για τον υπολογισμό των επιπέδων κβάντισης του σήματος, του διαστήματος που απέχουν μεταξύ τους και κατ επέκταση των τιμών που αντιστοιχούν σε κάθε επίπεδο. Ωστόσο, αξίζει να αναφερθεί ότι λήφθηκε υπ όψιν ο τύπος: $Q(x) = \Delta \cdot \text{sign}(x) \cdot \lceil |x|/\Delta + 1/2 \rceil$, που αναφέρεται στην απόλυτη τιμή του σήματος. Μ' αυτόν τον τρόπο, το σήμα πλέον ανήκει στο διάστημα $[0, A]$ και οι αρνητικές τιμές ανακτώνται με τη βοήθεια του προσήμου, εξοικονομώντας μνήμη.

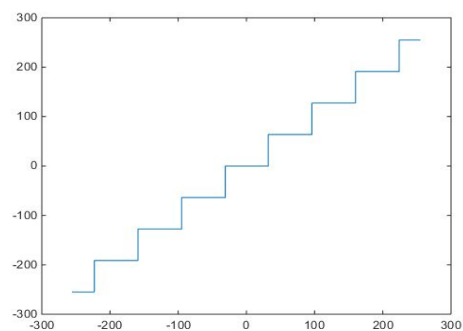
Σχεδιασμός Χαρακτηριστικών Συναρτήσεων του Κβαντιστή Ομοιόμορφης Κατανομής:

Κλήση της συνάρτησης uni_scalar(), έπειτα από τον προσδιορισμό του εύρους τιμών του σήματος και των άλλων δύο ορισμάτων για R από 0 έως 8 και σχεδίαση των χαρακτηριστικών με τη βοήθεια της συνάρτησης stairs(). Οι παράμετροι της συνάρτησης θα πρέπει να έχουν τις ίδιες διαστάσεις.

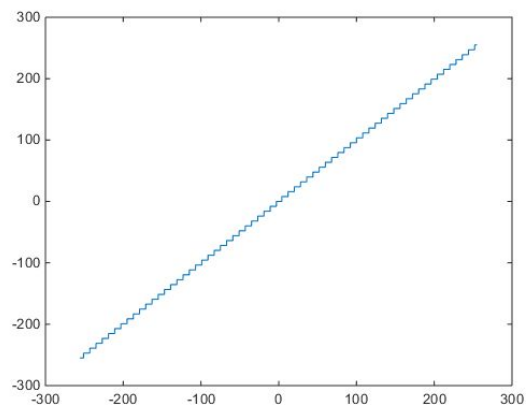
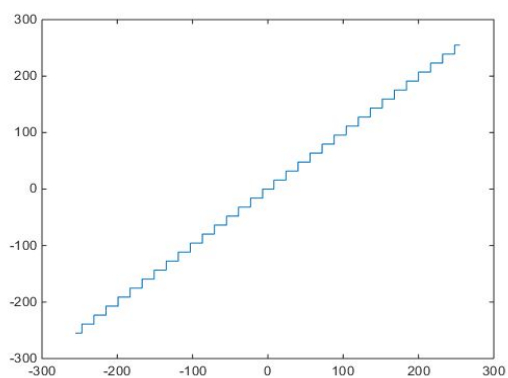
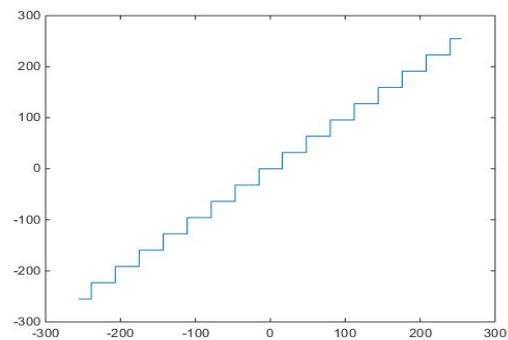
Κατά τη σχεδίαση είναι εμφανές ότι για μεγαλύτερες τιμές του R προκύπτουν περισσότερα επίπεδα κβάντισης λαμβάνοντας υπ όψιν κα το πρόσημο του σήματος. Παρατηρείται ακόμα ότι για R= 8 και R=9 η χαρακτηριστική καμπύλη αποκτά γραμμική μορφή.



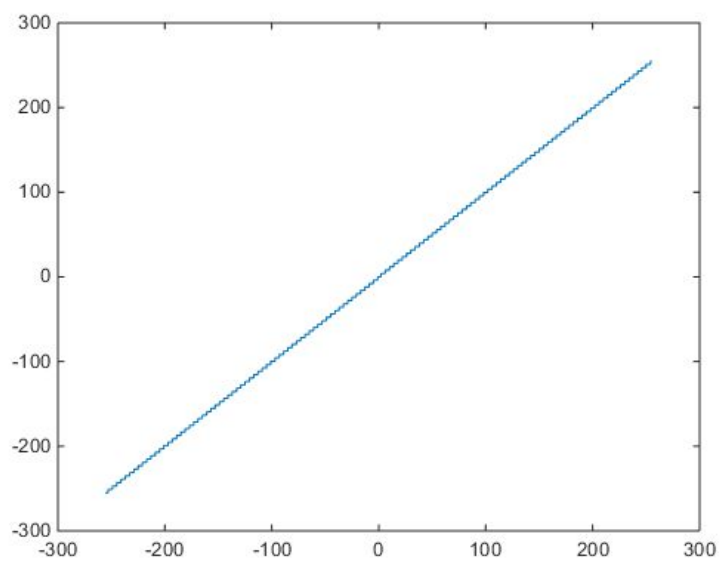
*R=0,R=1 αντίστοιχα



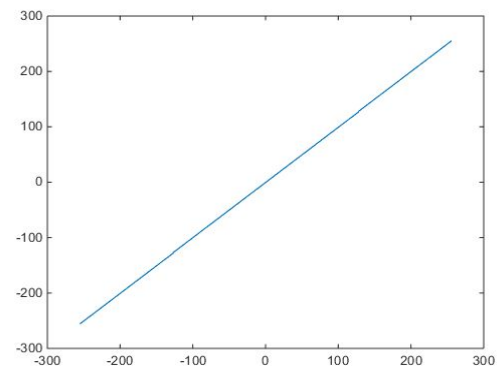
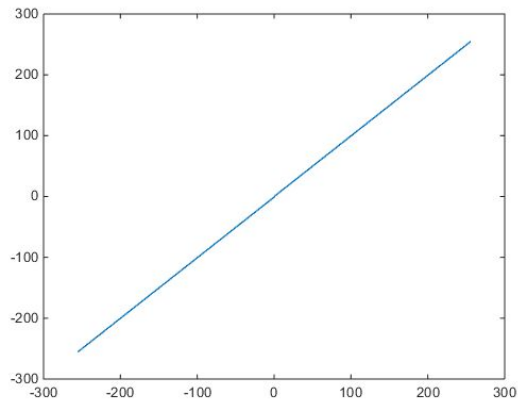
**R=2,3*



**Για R=4 ,R=5 αντίστοιχα*



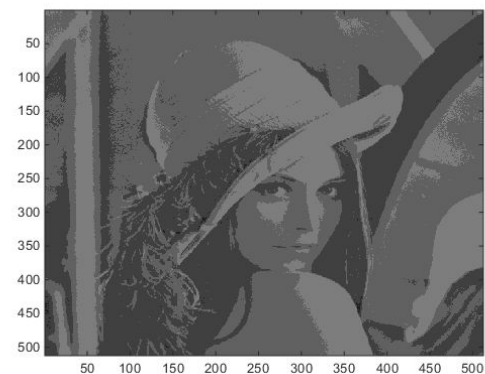
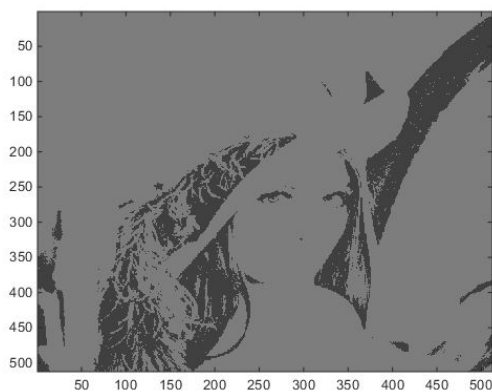
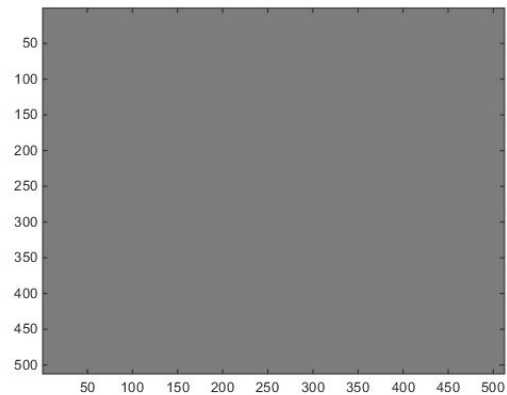
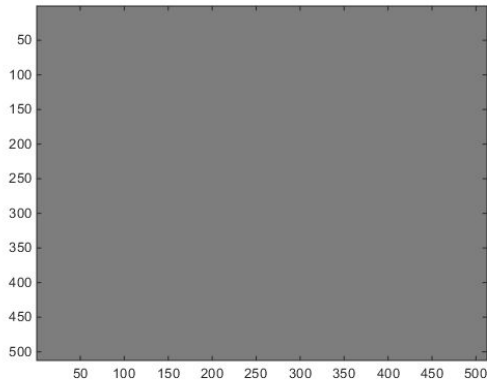
**Για R=6*



*Για $R=7$, $R=8$

Κβάντιση Ζητούμενης Εικόνας(8 bit) και Αναπαράσταση αποτελεσμάτων:

Η εικόνα “lena_gray_512.tif” που μας δίνεται αποτελείται από 8 bits και εμείς καλούμαστε να την κβαντίσουμε στα αντίστοιχα bits που αντιπροσωπεύει ο αριθμός R . Ωστόσο, η συνάρτηση που κατασκευάστηκε δέχεται σήματα μονής διάστασης. Επομένως πριν πραγματοποιηθεί η κβάντιση της εικόνας μέσω της `uni_scalar()` απαιτείται η μετατροπή της σε vector και έπειτα με τη βοήθεια της `reshape` επαναφέρεται στις διαστάσεις της αρχικής εικόνας(512 X 512). Τέλος, ακολουθεί η αναπαράσταση των εικόνων μέσω της `imagesc` στο διάστημα $[0 \text{ A}-1]$ και ο χρωματισμός τους σε γκρι χρώμα μέσω της `colormap()`.







Παρατηρείται ότι για μεγαλύτερο R παρουσιάζεται μεγαλύτερη γκάμα αποχρώσεων με αποτέλεσμα την ακριβέστερη αναπαράσταση της εικόνας ενώ για μικρότερο R παρόλο που χάνεται αρκετή πληροφορία, μας δίνεται η δυνατότητα να συμπίεσουμε σημαντικά το μέγεθος της εικόνας μιας και απαιτούνται λιγότερα bits για την αναπαράσταση των αποχρώσεων.

Μέτρηση Παραμόρφωσης Λόγω Κβάντισης:

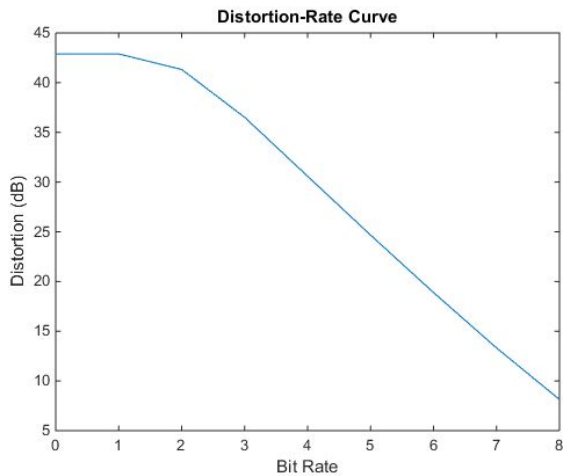
Η παραμόρφωση μετρήθηκε μέσω της συνάρτησης MSE (κατασκευάστηκε από προηγούμενα εργαστήρια), όπου μετράει την παραμόρφωση μεταξύ της αρχικής και της κβαντισμένης εικόνας. Η διαδικασία αυτή εκτελείται τόσες φορές όσες και το πλήθος τιμών του R.

Κατά τον υπολογισμό των παραμορφώσεων είναι αντιληπτό ότι η μικρότερη παραμόρφωση παρουσιάζεται για τη μεγαλύτερη τιμή του R. Αντίθετα, όταν το R λαμβάνει τη μικρότερη τιμή η παραμόρφωση γίνεται αισθητά μεγάλη.

dist_1= 1.9439e+04	,για R=0
dist_2=1.9439e+04	,για R=1
dist_3=1.3577e+04	,για R=2
dist_4=4.4814e+03	,για R=3
dist_5=1.1444e+03	,για R=4
dist_6=292.8126	,για R=5
dist_7=77.4146	,για R=6
dist_8=21.4833	,για R=7
dist_9=6.5071	,για R=8

Σχεδίαση Καμπύλης του Ρυθμού Παραμόρφωσης D(R):

Για τη σχεδίαση της συγκεκριμένης καμπύλης σε έναν πίνακα τοποθετούνται οι τιμές των bit και σε έναν άλλο πίνακα οι τιμές της παραμόρφωσης που υπολογίστηκαν προηγουμένως πολλαπλασιασμένες με το $10 \cdot \log_{10}$. Επομένως, η καμπύλη στον άξονα των x περιλαμβάνει τον δείκτη των bit και στον άξονα των y τον δείκτη των παραμορφώσεων που μετρήθηκαν βάσει της MSE σε dB. Η μορφή της καμπύλης οφείλεται στη σχέση των bits με τις παραμορφώσεις, η οποία σχολιάστηκε στο προηγούμενο ερώτημα. Ωστόσο, αξίζει να αναφερθεί ότι όταν το R λαμβάνει τις τιμές 0 και 1 η παραμόρφωση είναι η ίδια, το οποίο διαφαίνεται και στη γραφική.



ΜΕΡΟΣ Β:

Φόρτωση Αρχείου βίντεο:

Το αρχείο αυτό φορτώνεται μέσω της συνάρτησης VideoReader(filename) και περιλαμβάνεται σ' αυτή σαν όρισμα . Έτσι, δημιουργείται ένα πολυμεσικό αντικείμενο, το οποίο μπορεί να διαβάσει τα δεδομένα του βίντεο μέσα από αυτό.

Υπολογισμός του Αριθμού των Frames, του ρυθμού των frames, της ανάλυσης του κάθε frame και της συνολικής διάρκειας του βίντεο:

- Ο ρυθμός των Frames του video (frame_rate=30) , δηλαδή το πλήθος των frames που προκύπτουν ανά δευτερόλεπτο υπολογίζεται με την κλήση της υπάρχουσας μεθόδου FrameRate, η οποία αντιστοιχεί στο πολυμεσικό αντικείμενο που δημιουργήθηκε.
- Η συνολική διάρκεια του βίντεο(duration=4.7000) υπολογίζεται μέσω της κλήσης της μεθόδου Duration για το αντίστοιχο αντικείμενο, όπου υπολογίζεται το μέγεθος του αρχείου σε δευτερόλεπτα.
- Ο συνολικός Αριθμός Frames(number_of_frames=141) προκύπτει ως το γινόμενο του ρυθμού των frames και της διάρκειας του video, τα οποία φυσικά αντιστοιχούν στο αντικείμενο που δημιουργείται κατά την ανάγνωση του επιθυμητού αρχείου.
- Η ανάλυση του video αφορά το πλάτος και το ύψος του κάθε frame σε εικονοστοιχεία.
 - ❖ Πλάτος (Width=320): Κλήση της Width() για το αντίστοιχο αντικείμενο.
 - ❖ Ύψος (Height=240): Κλήση της Height() για το αντίστοιχο αντικείμενο.

Αναπαράσταση του Frame σε σχέση με τον ρυθμό των frames:

Μέσα σε ένα while loop ελέγχεται αν υπάρχει κάποιο frame διαθέσιμο για ανάγνωση από κάποιο αρχείο βίντεο και στη συνέχεια διαβάζεται το αμέσως επόμενο frame , που είναι διαθέσιμο και έπειτα διακόπτεται η εκτέλεση της προηγούμενης λειτουργίας για χρόνο (1/ ρυθμός των frames) πριν τη μετάβαση στο επόμενο frame. Για τη συγκεκριμένη

αναπαράσταση χρησιμοποιήθηκαν συνολικά οι συναρτήσεις `hasFrame()`, `readFrame()` and `pause()`.

Video at original framerate



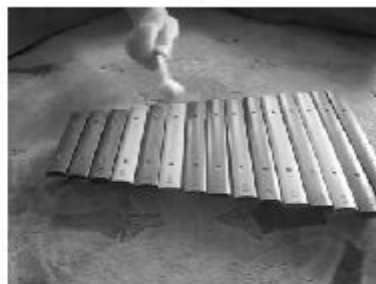
Εξαγωγή του 50ου frame του video και Μετατροπή της RGB εικόνας σε Γκρι Κλίμακα:

Μέσα στη `while` με τη βοήθεια ενός δείκτη ελέγχεται αν το τρέχον `frame` είναι το 50ο και στη συνέχεια εξάγεται, ενώ ακολουθεί η κατάλληλη ενημέρωση του δείκτη. Στη συνέχεια, μέσω της συνάρτησης `rgb2gray()` το `frame` αυτό μετατρέπεται σε γκρι κλίμακα. Τέλος με τη χρήση της `subplot()` μπορούν πλέον να αναπαρασταθούν οι δύο εικόνες σε κοινό παράθυρο.

frame 50



frame 50 grayscale

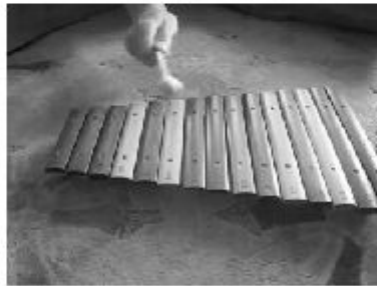


ΜΕΡΟΣ Γ:

Σε αυτό το μέρος καλούμαστε να δημιουργήσουμε ένα σύστημα συμπίεσης εικόνας χρησιμοποιώντας τον μετασχηματισμό Haar για την διαχωρισμό της πληροφορίας με βάση την συχνότητα και με την χρήση του κβαντιστή απο το μέρος 1 για την μείωση του εύρους τιμών των υψηλών συχνοτήτων οι οποίες έτσι και αλλιώς έχουν πολύ περιορισμένο εύρος τιμών με αποτέλεσμα να μπορούμε να πετύχουμε μεγάλη συμπίεση με μικρή επίπτωση στην σημαντική πληροφορία της εικόνας.

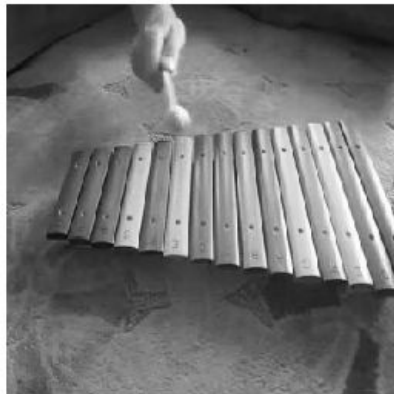
1. Η εικόνα που χρησιμοποιείται είσοδος είναι το frame 50 grayscale το οποίο εξάγαμε απο την προηγούμενη άσκηση.

frame 50 grayscale



2. Γίνεται resize της εικόνας σε 256x256.

resized frame 50



3. Στην συνέχεια χρησιμοποιούμε τον μετασχηματισμό Haar για να αποκτήσουμε 2 decomposition levels.
4. Κβαντίζουμε όλα τα subbands με quantization step $R=4$, εκτός απο το subband χαμηλής συχνότητας μιας και αυτό μεταφέρει την σημαντικότερη πληροφορία.

5. Υπολογίζουμε με την χρήση της συνάρτησης `entropy_calculate` την εντροπία του καθε subband μαζί με την συνολική εντροπία προκειμένου να υπολογίσουμε το compression ratio.

*Το compression ratio έχει υπολογιστεί με την χρήση του τύπου

$$(\text{original_image_entropy}/\text{quantized_image_entropy}) * 100$$

**Για τον διαχωρισμό τους, έχει δοθεί σε κάθε subband απο ένα όνομα που ξεκινάει με τα γράμματα (h,v,d) και a για το subband χαμηλής συχνότητας, ενώ ακολουθείται απο τον αριθμό που υποδηλώνει το decomposition level (1,2).

Τα αποτελέσματα είναι ως εξής:

h1	v1	d1	h2	v2	d2	Final Image
230%	124%	145%	26%	19%	22%	1152%

6. Επαναφέρουμε την αρχική εικόνα εφαρμόζοντας αντίστροφο μετασχηματισμό Haar και μετράμε την παραμόρφωση της εικόνας με την μέθοδο Peak Signal-to-Noise Ratio(PSNR) =87.3.
7. Χρησιμοποιούμε διαφορετικό βήμα κβάντισης για καθε layer με R=3 για το πρωτο layer και R=5 για το δεύτερο layer.
8. Υπολογίζουμε με την χρήση της συνάρτησης `entropy_calculate` την εντροπία του καθε subband μαζί με την συνολική εντροπία προκειμένου να υπολογίσουμε το compression ratio.

Τα αποτελέσματα είναι ως εξής:

h1	v1	d1	h2	v2	d2	Final Image
712%	212%	353%	16%	13%	14%	1147%

Παρατηρούμε οτι στο πρώτο layer παρατηρούμε αισθητά μεγαλύτερο compression ratio λόγω του μικρότερου βήματος κβάντισης ενώ στο δεύτερο layer ισχύει το αντίθετο. Το compression ratio της εικόνας είναι ελαφρώς μικρότερο.

9. Επαναφέρουμε την αρχική εικόνα εφαρμόζοντας αντίστροφο μετασχηματισμό Haar και μετράμε την παραμόρφωση της εικόνας με την μέθοδο Peak Signal-to-Noise Ratio(PSNR) =83.9. Η παραμόρφωση είναι μικρότερη απο την εικόνα του βήματος 6.

Τα τελικά αποτελέσματα:

original image



compresed image R=3,R=5



original image



compresed image R=4



Παρατηρείται ότι παρά το μεγάλο compression ratio οι τελικές εικόνες είναι φαινόνται σχεδόν ίδιες με τις αρχικές με γυμνό μάτι.