# 

# Πανεπιστήμιο Αιγαίου

## Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

### *Πολυμέσα*

# Τεχνική Αναφορά Άσκησης 2

Τσότρας Στέφανος 321/2013189

Κοντογιάννης Νίκος 321/2015094

8/12/2019

1. Για αυτό το ερώτημα για τη κατασκευή ενός απλού κωδικοποιητή μήκους διαδρομής (run-length encoder), υλοποιήθηκαν δύο encoders (στη συνέχεια και decoders).

Ο ένας κάνει απλή κωδικοποίηση των τιμών μίας grayscale εικόνας, δηλαδή σώζει τη τιμή της μεταβλητής της εικόνας και στην επόμενη θέση το πόσες φορές συναντάται η ίδια τιμή μέχρι να βρούμε διαφορετική, τότε στην επόμενη θέση σώζεται η νέα τιμή και στη συνέχεια το πόσες φορές συναντάται κ.τ.λ.

Ο άλλος κωδικοποιητής πρώτα μετατρέπει την εικόνα σε δυαδική ροή και έπειτα εφαρμόζει την ίδια τεχνική, μόνο που τώρα δε χρειάζεται να αποθηκεύσουμε την μεταβλητή που επαναλαμβάνεται, μόνο την πρώτη, γιατί ξέρουμε ότι οι εναλλασόμενες τιμές είναι 0 ή 1.

Οι λόγοι συμπίεσης για κάθε εικόνα είναι :

Εικόνα Normal Encoding Binary Encoding

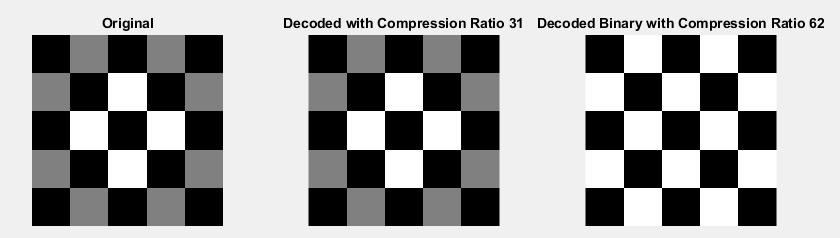
Chess image 31:1 62:1

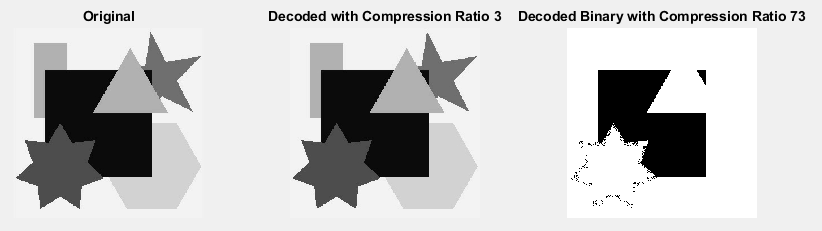
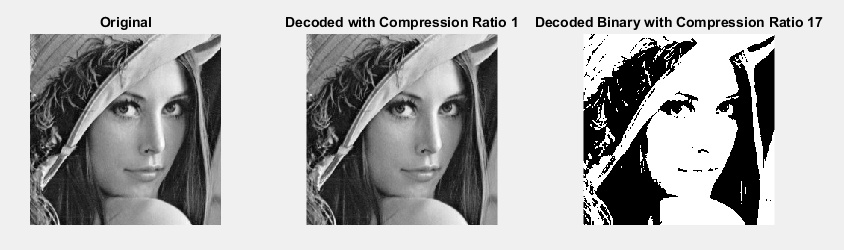
Geometrical 3:1 73:1

Lenna 1:1 17:1

1. Για την αποκωδικοποίηση των εικόνων υλοποιήθηκαν και οι αντίστοιχοι decoders.

Ο normal decoder παίρνει τη κωδικοποιημένη εικόνα και για όσες φορές λέει η κωδικοποίηση τοποθετεί την τιμή σε καινούργιο πίνακα (με μέγεθος της αρχικής εικόνας), μέχρι ο μετρητής να γίνει ίσος με το μέγεθος των στηλών της αρχικής εικόνας (μήκος στήλης και γραμμής έχουν αποθηκευτεί ως οι πρώτες δύο εγγραφές στην κωδικοποιημένη εικόνα) και αντίστοιχα για τον binary decoder.

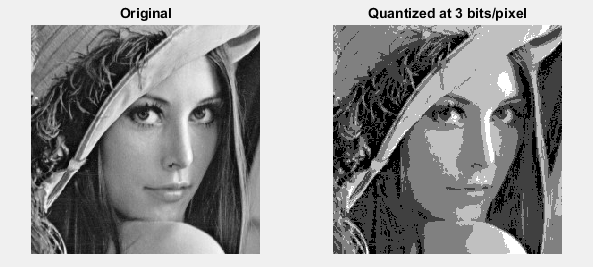




Όπως βλέπουμε απτά αποτελέσματα των αποκωδικοποιημένων εικόνων η ‘’normal’’ κωδικοποίηση έχει σχετικά μικρό λόγο συμπίεσης αλλά κρατάει την εικόνα αυτούσια, ενώ η binary κωδικοποίηση έχει πολύ μεγάλο λόγο συμπίεσης αλλά ουσιαστικά γυρνάει την εικόνα ασπρόμαυρη και οι λεπτομέρειες χάνονται όπως και κάποια σχέδια που δεν περνάνε το threshold στην μετατροπή σε binary οπότε παίρνουν τιμή 0.

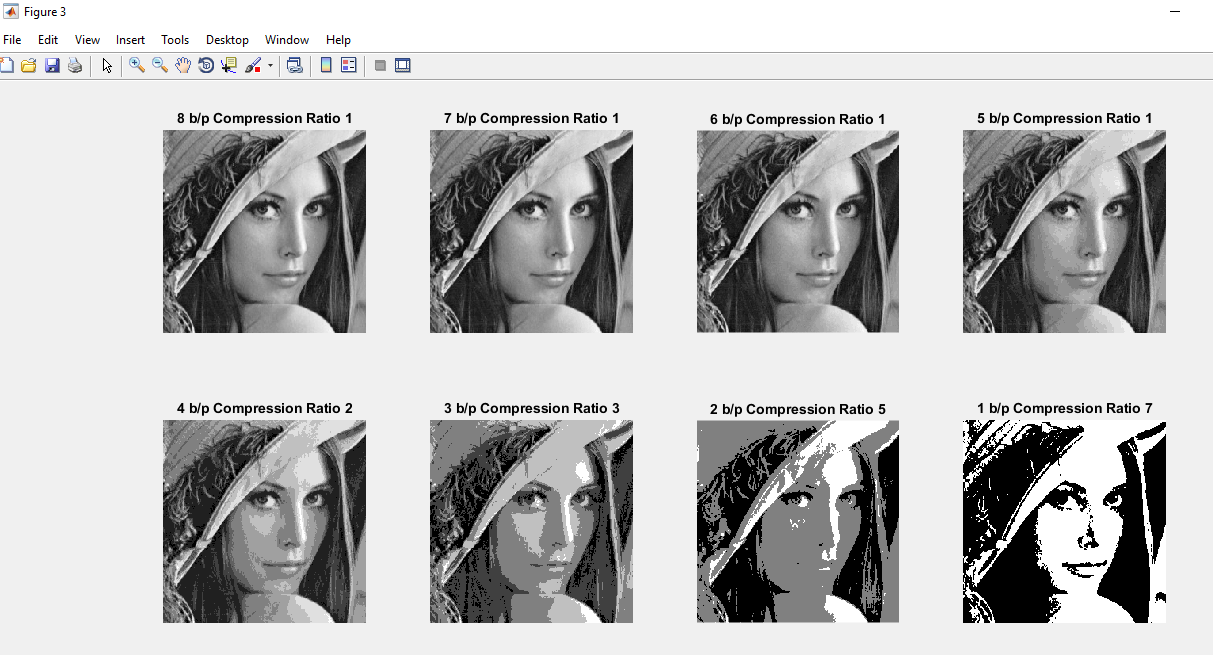
1. Υλοποιήθηκε συνάρτηση imquant η οποία η οποία κβαντίζει μία εικόνα στο δοθέν bit/δείγμα.

Αυτό πετυχαίνετε διαιρώντας τις τιμές της εικόνας με 2^bit όπου bit, το πόσο bit/δείγμα θέλουμε να κατεβάσουμε την εικόνα και στη συνέχεια στρογγυλοποιούμε τα αποτελέσματα.



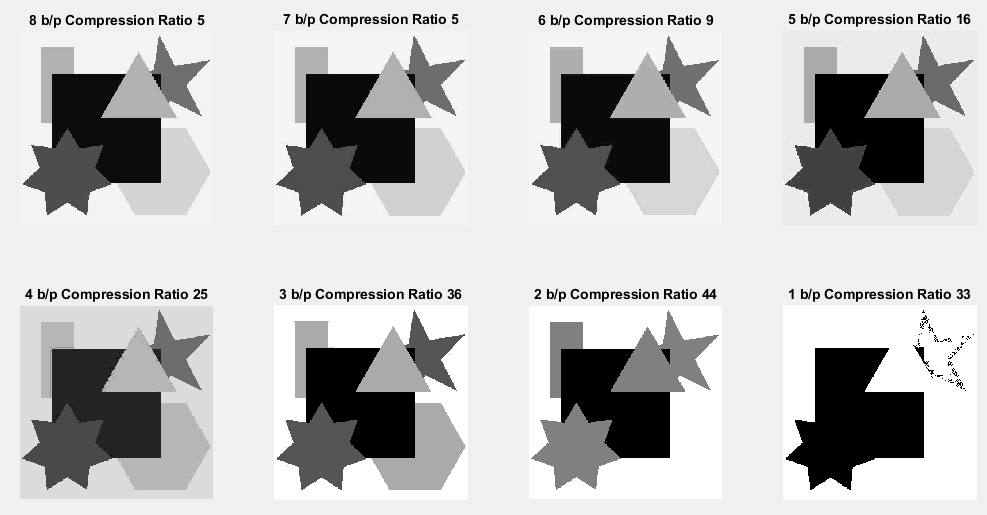
1. Υλοποιήθηκε συνάρτηση multQuantMo η οποία κβαντίζει μία εικόνα σε όλες τις διαφορετικές τιμές του πλήθους των bits/pixel, 8-1, υπολογίζοντας για κάθε τιμή τον λόγο συμπίεσης της κωδικοποίησης τους με run-length encoding.

Επίσης, η συνάρτηση αποθηκεύει κάθε στάδιο κβάντισης ως εικόνα με τη getframe, έτσι ώστε να χρησιμοποιηθούν αργότερα σαν είσοδο στην movie.



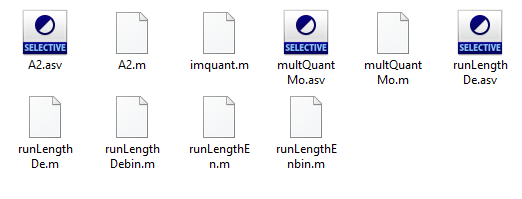
Τέλος, η συνάρτηση εμφανίζει το παραπάνω figure απεικονίζοντας την εικόνα στα διάφορα στάδια κβάντισης και τον λόγο συμπίεσης με την runLengthEn.

Παρατηρείται ότι όσο μικραίνουν τα διαθέσιμα bit per pixel ο λόγος συμπίεσης ανεβαίνει, λογικό εφόσον οι τιμές των pixel περιορίζονται και έτσι έχουμε παραπάνω ίδια pixel.



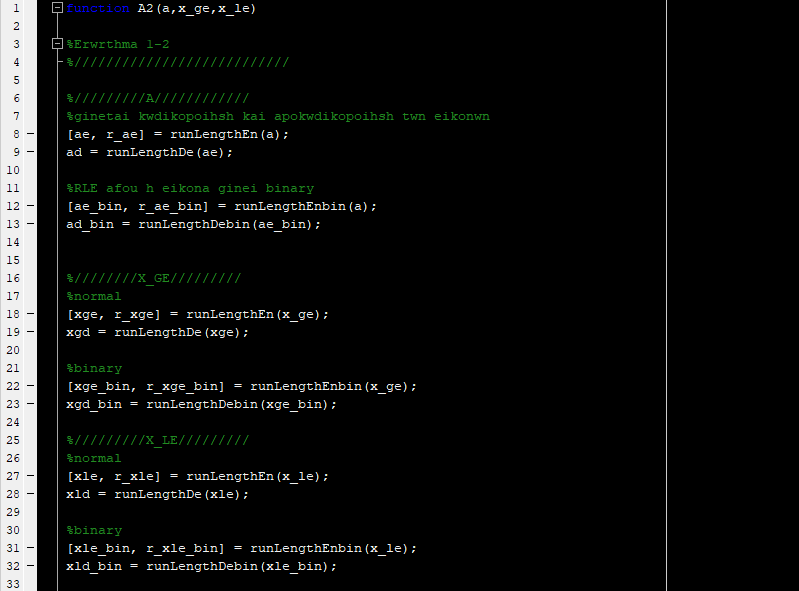
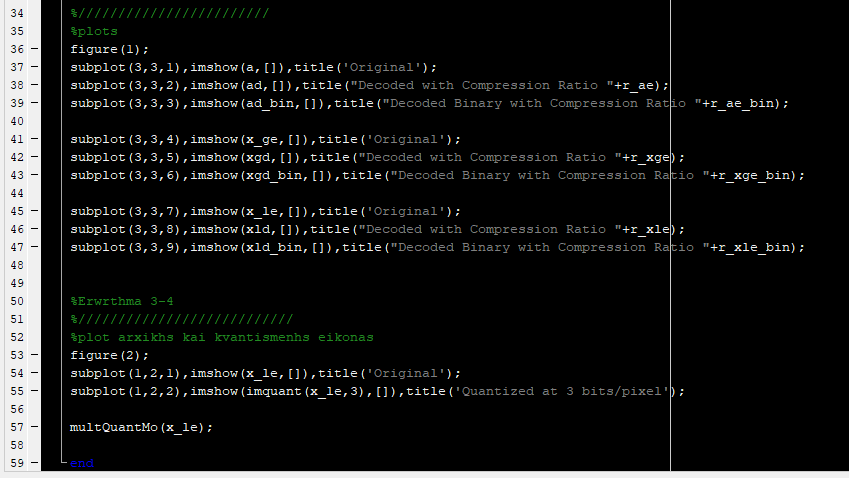
Στο παραπάνω σχήμα βλέπουμε μία ενδιαφέρον φαινόμενο, συγκεκριμένα στη τιμή 1b/p κάποιες τιμές στο πάνω δεξιά αστέρι περνάνε το threshold και παίρνουν την τιμή 1 κάνοντας την συμπίεση λιγότερο αποτελεσματική ακόμα και απτις 2 πιο μεγάλες τιμές κβάντισης .

**Παράρτημα**



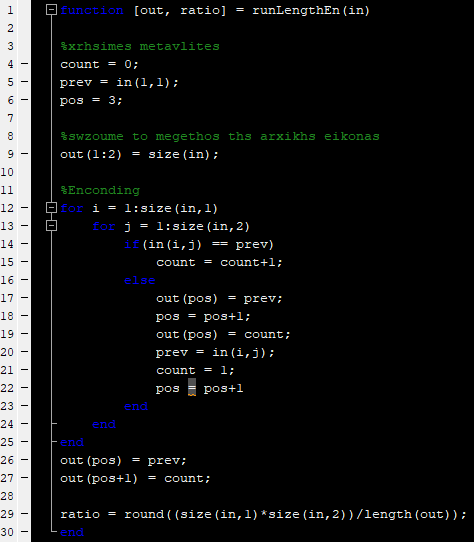
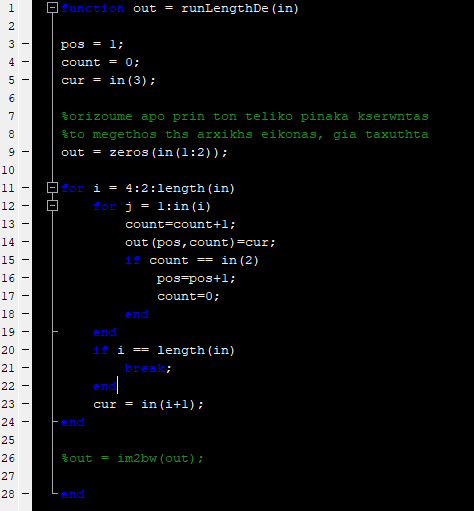
**Κώδικας**

Η βασική συνάρτηση που απαντάει στα ερωτήματα και καλεί τις υπόλοιπες είναι η A2.

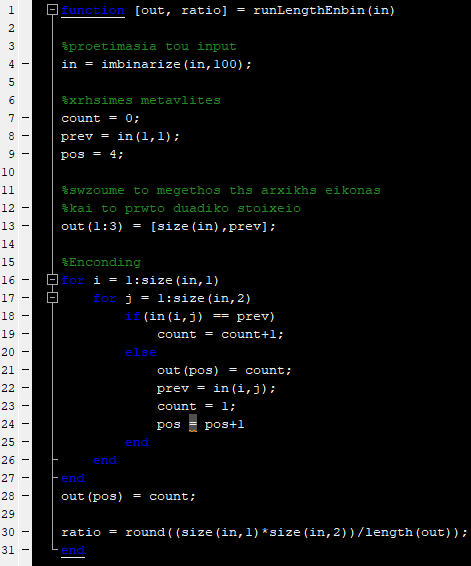
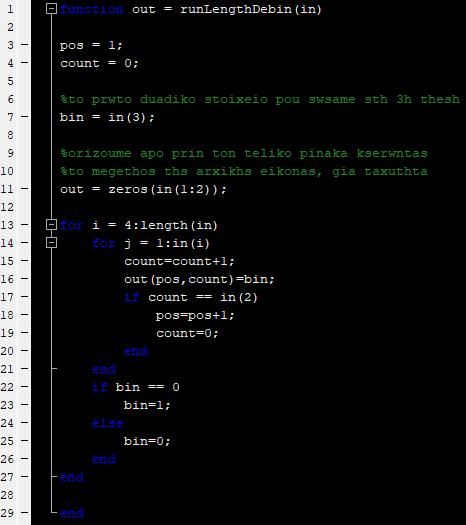


Οι υπόλοιπες συναρτήσεις που υλοποιήθηκαν είναι οι runLengthEn, runLengthDe, runLengthEnbin, runLengthDebin, imquant, multQuantMo.

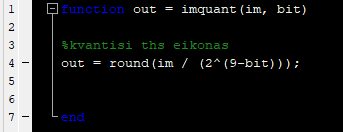
runLengthEn runLengthDe



runLengthEnbin runLengthDebin



* Imquant



* multQuantMo

