

ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ ΚΑΙ
ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ

ΚΩΔΙΚΟΠΟΙΗΣΗ
ΑΠΟΚΩΔΙΚΟΠΟΙΗΣΗ
JPEG



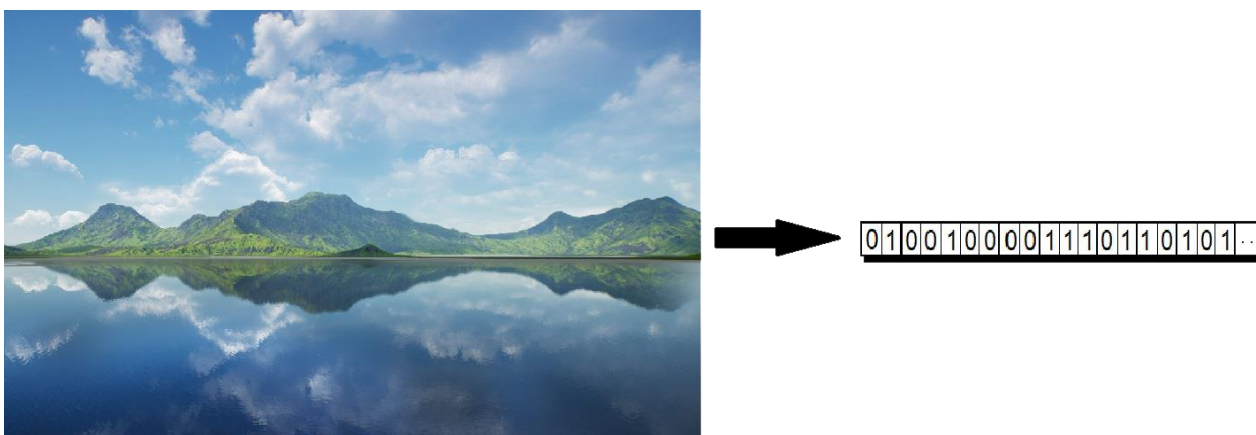
ΟΝΟΜΑ: ΚΩΝΣΤΑΝΤΙΝΟΣ
ΕΠΙΘΕΤΟ: ΛΕΤΡΟΣ
ΣΧΟΛΗ: ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ: ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧ.
ΥΠΟΛΟΓΙΣΤΩΝ
ΑΕΜ: 8851
ΕΤΟΣ: 2019-2020

Περιεχόμενα

1. Εισαγωγή	3
2. Τμήμα A – JPEG Library.....	4
2.1 Προεπεξεργασία.....	4
2.2 Μετασχηματισμός DCT.....	7
2.3 Κβαντισμός	8
2.4 Zig Zag Scanning και Run Length Encoding	9
2.5 Huffman Encoding.....	11
2.6 Συνάρτηση Επίδειξης - Demo 1.....	14
3. Τμήμα B – JPEG Integration	16
3.1 JPEG Encoder/Decoder	16
3.2 Καταγραφή Αποτελεσμάτων	17
3.3 Συνάρτηση Επίδειξης - Demo 2.....	33
4. Επιπρόσθετα Αρχεία MATLAB	34

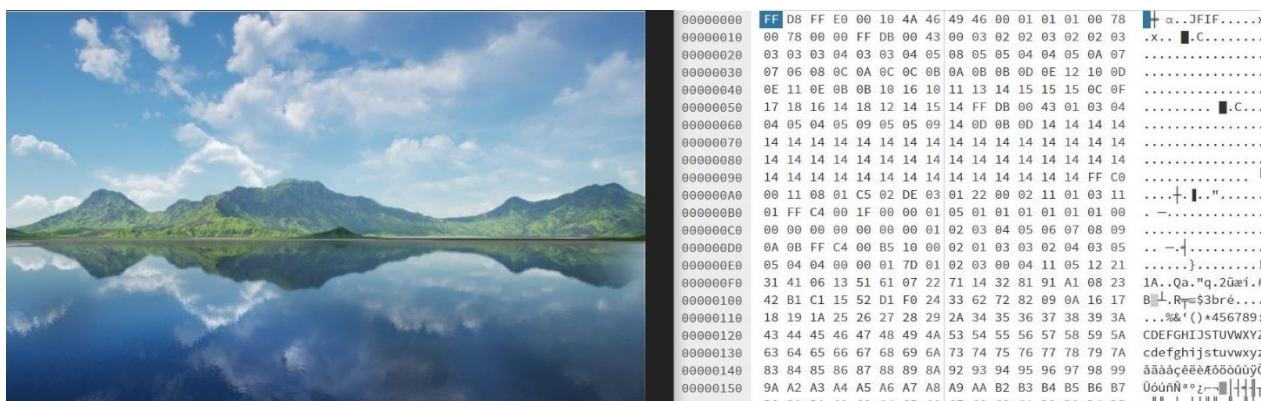
1. Εισαγωγή

Σκοπός της εργασίας αυτής είναι η δημιουργία ενός κωδικοποιητή-αποκωδικοποιητή ακίνητων εικόνων με απώλειες (Lossy Compression) με βάση το πρότυπο JPEG (ISO/IEC 109181:1994). Συγκεκριμένα, η υλοποίηση που ακολουθεί επιδιώκει να προσεγγίσει την εκδοχή baseline sequential DCTbased.



Σχήμα 1.1: Κωδικοποίηση Εικόνας – Αναπαράσταση μέσω ενός Stream από bits.

Η παραπάνω διαδικασία διακρίνεται σε τρία τμήματα, εκ των οποίων στην παρούσα εργασία υλοποιούνται τα δύο πρώτα. Παραλείπεται το τμήμα που αφορά την τελική κατασκευή του bit-Stream κωδικοποίησης, με τη συμμετοχή των διάφορων headers της εικόνας, ώστε να είναι αποκωδικοποιήσιμα από οποιονδήποτε αποκωδικοποιητή.



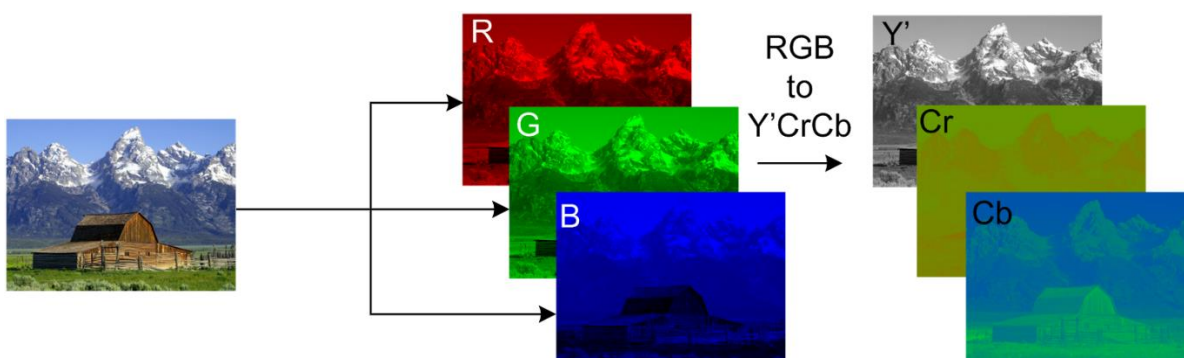
Σχήμα 1.2: Πραγματικό παράδειγμα bit-Stream Εικόνας. Κάθε ζεύγος αριθμών στο δεκαεξαδικό σύστημα αναπαριστά ένα Byte.

2. Τμήμα A – JPEG Library

Σκοπός του τμήματος αυτού είναι η επεξεργασία της προς συμπίεσης εικόνας προκειμένου να επιτευχθεί η κωδικοποίηση-αποκωδικοποίηση. Συγκεκριμένα, στο στάδιο αυτό γίνεται η δημιουργία όλων των συναρτήσεων που θα συμμετάσχουν στην υλοποίηση του προτύπου. Για κάθε συνάρτηση που αφορά τον κωδικοποιητή, υλοποιείται και η αντίστροφή της, η οποία θα βρίσκεται στο αντίστοιχο τμήμα του αποκωδικοποιητή.

2.1 Προεπεξεργασία

Συγκεκριμένα, η διαδικασία κωδικοποίησης ξεκινάει με τη μετατροπή της εικόνας από το σύστημα χρωματικών συνιστωσών RGB (Red – Green – Blue) στο σύστημα YCbCr (Luminance – Chrominance).



Σχήμα 2.1: Μετατροπή Εικόνας από το σύστημα RGB στο σύστημα YCbCr.

Η εικόνα RGB έχει τη μορφή τρισδιάστατου πίνακα μεγέθους $R \times C \times 3$, όπου R , C η οριζόντια και κατακόρυφη διάσταση ενώ η τρίτη διάσταση λαμβάνει τις τιμές 1 (Red), 2 (Green), 3 (Blue) καθώς περιέχει τις τρεις χρωματικές συνιστώσες της εικόνας.

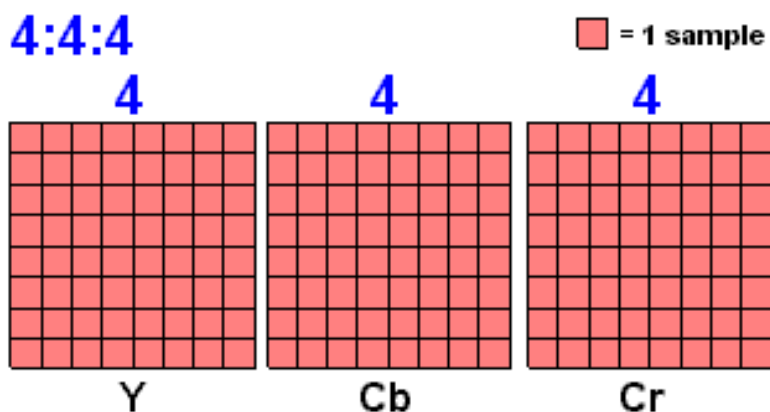
Για να επιτευχθεί η παραπάνω μετατροπή, για κάθε εικονοστοιχείο (pixel), της αρχικής εικόνας RGB πραγματοποιείται η παρακάτω πράξη:

$$\begin{bmatrix} Y_{ij} \\ Cb_{ij} \\ Cr_{ij} \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \cdot \begin{bmatrix} R_{ij} \\ G_{ij} \\ B_{ij} \end{bmatrix}$$

όπου οι δείκτες i, j υποδεικνύουν τη θέση του εκάστοτε pixel της εικόνας. Έτσι τελικά κατασκευάζεται ένας τρισδιάστατος πίνακας με τις τρεις συνιστώσες Y , Cb και Cr .

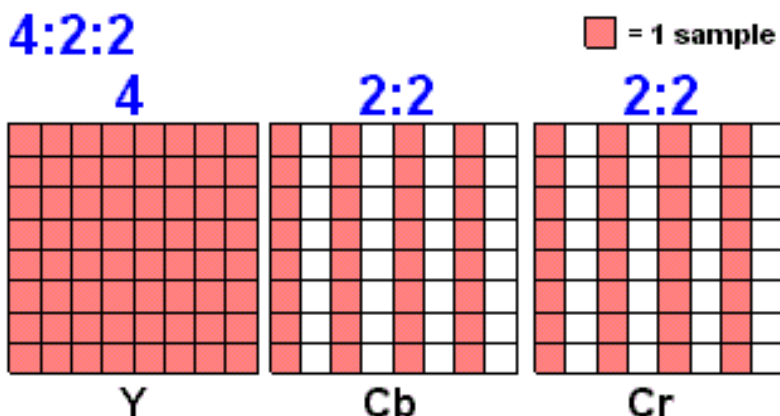
Στη συνέχεια ακολουθείται ένα στάδιο υποδειγματοληψίας χρώματος προκειμένου να μειωθεί ο όγκος της εικόνας. Συγκεκριμένα, η διαδικασία αυτή διακρίνεται σε 3 κατηγορίες.

Η πρώτη κατηγορία ονομάζεται υποδειγματοληψία χρώματος **4:4:4**, κατά την οποία λαμβάνεται ολόκληρη η εικόνα που σχηματίστηκε **χωρίς κάποια δειγματοληψία**.



Σχήμα 2.2: Υποδειγματοληψία Χρώματος 4:4:4.

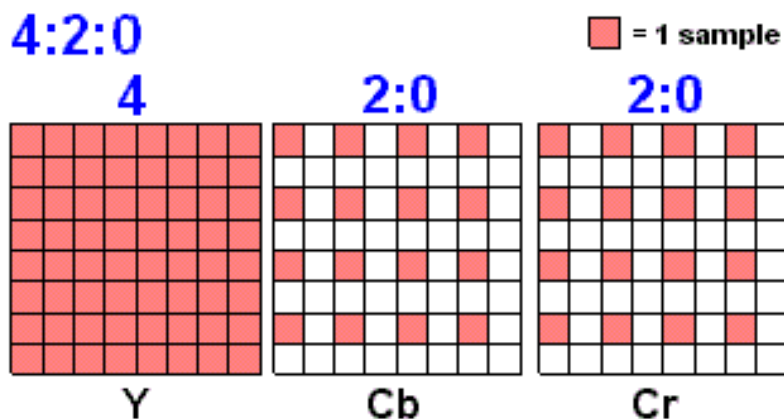
Η δεύτερη κατηγορία ονομάζεται υποδειγματοληψία χρώματος **4:2:2**. Στην κατηγορία αυτή διαγράφονται οι στήλες των χρωματικών συνιστωσών Cb, Cr που βρίσκονται σε άρτιες θέσεις στον πίνακα YCbCr. Είναι εμφανές ότι η κατηγορία αυτή **εισάγει απώλειες** καθώς δεν υπάρχει τρόπος πλήρους ανακατασκευής της εικόνας μετά από τη διαδικασία αυτή. Το αποτέλεσμα που προκύπτει είναι τρεις πίνακες διαστάσεων $R \times C$, $R \times \frac{C}{2}$, $R \times \frac{C}{2}$, για τις συνιστώσες Y, Cb, Cr αντίστοιχα.



Σχήμα 2.3: Υποδειγματοληψία Χρώματος 4:2:2.

Τέλος, η τρίτη κατηγορία που υλοποιείται ονομάζεται υποδειγματοληψία χρώματος **4:2:0**. Στην κατηγορία αυτή διαγράφονται τόσο οι στήλες όσο

και οι γραμμές των χρωματικών συνιστωσών Cb, Cr που βρίσκονται σε άρτιες θέσεις στον πίνακα YCbCr. Είναι εμφανές ότι η κατηγορία αυτή **εισάγει ακόμα μεγαλύτερες απώλειες** καθώς και πάλι δεν υπάρχει τρόπος πλήρους ανακατασκευής της εικόνας μετά από τη διαδικασία αυτή. Το αποτέλεσμα που προκύπτει είναι τρεις πίνακες διαστάσεων $R \times C$, $\frac{R}{2} \times \frac{C}{2}$, $\frac{R}{2} \times \frac{C}{2}$, για τις συνιστώσες Y, Cb, Cr αντίστοιχα.



Σχήμα 2.4: Υποδειγματοληψία Χρώματος 4:2:0.

Η διαδικασία αποκωδικοποίησης προκύπτει με αντίστροφη σειρά των παραπάνω βημάτων.

Αρχικά γίνεται μετατροπή από τους τρεις πίνακες Y, Cb, Cr με μειωμένες διαστάσεις σε ένα τρισδιάστατο πίνακα διάστασης $R \times C \times 3$. Για να επιτευχθεί αυτό γίνεται παρεμβολή προκειμένου να συμπληρωθούν οι θέσεις των δειγμάτων που χάθηκαν κατά την υποδειγματοληψία. Για τη διαδικασία της παρεμβολής στις διάφορες κατηγορίες υποδειγματοληψίας γίνεται χρήση του γινομένου Kronecker.

Σύμφωνα με τον ορισμό, αν ο **A** είναι ένας $m \times n$ πίνακας και ο **B** είναι ένας $p \times q$ πίνακας, τότε το γινόμενο Kronecker $\mathbf{A} \otimes \mathbf{B}$ είναι ο μπλοκ πίνακας $pm \times qn$:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

Αρχικά, για την κατηγορία υποδειγματοληψίας χρώματος 4:4:4 δεν υπάρχει ανάγκη για παρεμβολή. Για την κατηγορία 4:2:2 εφαρμόζεται το γινόμενο Kronecker ανάμεσα στους χρωματικούς πίνακες Cb, Cr με το

διάνυσμα $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ ενώ για την κατηγορία 4:2:0 εφαρμόζεται το γινόμενο Kronecker ανάμεσα στους χρωματικούς πίνακες Cb, Cr με τον πίνακα $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

Τέλος, η μετατροπή από εικόνα YCbCr σε RGB, γίνεται αντίστροφα και συγκεκριμένα μέσω του παρακάτω μετασχηματισμού:

$$\begin{bmatrix} R_{ij} \\ G_{ij} \\ B_{ij} \end{bmatrix} = \begin{bmatrix} 0 \\ -128 \\ -128 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.344136 & -0.714136 \\ 1 & 1.772 & 0 \end{bmatrix} \cdot \begin{bmatrix} Y_{ij} \\ Cb_{ij} \\ Cr_{ij} \end{bmatrix}$$

** Σημείωση:* Σε περίπτωση που το πλήθος γραμμών και στηλών του τρισδιάστατου πίνακα RGB δεν είναι ακέραιο πολλαπλάσιο του 16, αφαιρείται από το τέλος των εικόνων το απαραίτητο πλήθος γραμμών και στηλών ώστε να πληρείται η παραπάνω συνθήκη.

2.2 Μετασχηματισμός DCT

Στη συνέχεια οι πίνακες Y, Cb και Cr που προέκυψαν από την προηγούμενη διαδικασία διαχωρίζονται σε μπλοκ μεγέθους 8×8 στα οποία εφαρμόζεται στη συνέχεια ο μετασχηματισμός DCT2 (Discrete Cosine Transform 2D). Έτσι προκύπτει ένα σύνολο από μπλοκ που περιέχουν τους συντελεστές DCT. Το πρώτο στοιχείο του κάθε μπλοκ ονομάζεται DC όρος ενώ τα υπόλοιπα στοιχεία AC όροι.

Ο μετασχηματισμός DCT2 φαίνεται παρακάτω:

$$S_{pq} = \frac{a_p a_q}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \left[\frac{\pi(2m+1)p}{2M} \right] \cos \left[\frac{\pi(2n+1)q}{2N} \right]$$

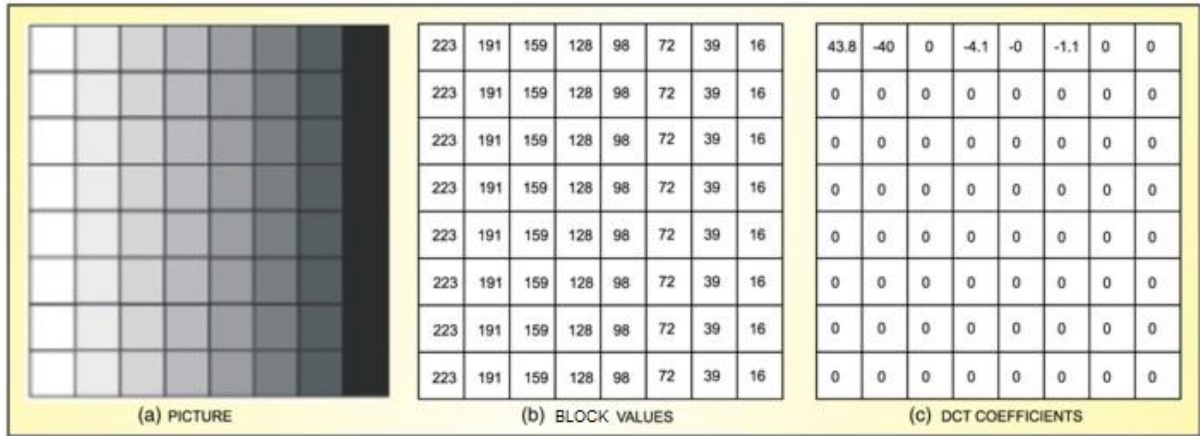
Όπου $0 \leq p \leq M-1$ και $0 \leq q \leq N-1$, S_{pq} ο συντελεστής που βρίσκεται στη θέση (p, q) του μετασχηματισμένου μπλοκ, A_{mn} το στοιχείο που βρίσκεται στη θέση (m, n) του κβαντισμένου μπλοκ και a_p, a_q συντελεστές ίσοι με 1 αν $p = 0$ ή $q = 0$ αντίστοιχα και ίσοι με $\sqrt{2}$, διαφορετικά.

Ο αντίστροφος μετασχηματισμός είναι:

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \frac{a_p a_q}{\sqrt{MN}} S_{pq} \cos \left[\frac{\pi(2m+1)p}{2M} \right] \cos \left[\frac{\pi(2n+1)q}{2N} \right]$$

Όπου $0 \leq m \leq M-1$ και $0 \leq n \leq N-1$, A_{mn} ο συντελεστής που βρίσκεται στη θέση (m,n) του αρχικού μπλοκ, S_{pq} το στοιχείο που βρίσκεται στη θέση (m,n) του μετασχηματισμένου μπλοκ και a_p, a_q συντελεστές ίσοι με 1 αν $p=0$ ή $q=0$ αντίστοιχα και ίσοι με $\sqrt{2}$, διαφορετικά.

Στη συγκεκριμένη περίπτωση των μπλοκ 8×8 , έχουμε $M = N = 8$.



Σχήμα 2.5: Μετασχηματισμός DCT σε Μπλοκ Ασπρόμαυρης Εικόνας.

Επιπλέον, σύμφωνα με το πρότυπο γίνεται αφαίρεση του όρου 128 στα στοιχεία του μπλοκ πριν το μετασχηματισμό και αντίστροφα πρόσθεση αυτού μετά τον αντίστροφο μετασχηματισμό.

2.3 Κβαντισμός

Στο σημείο αυτό γίνεται κβαντισμός των στοιχείων κάθε μπλοκ μέσω του παρακάτω τύπου:

$$qBlock_{ij} = \text{round} \left(\frac{dctBlock_{ij}}{qScale \cdot qTable_{ij}} \right)$$

όπου οι δείκτες i, j υποδεικνύουν τη θέση του εκάστοτε δείγματος στο συγκεκριμένο μπλοκ, $qScale$ σταθερά ρύθμισης του κβαντιστή και $qTable$ ένας πίνακας μεγέθους 8×8 που ονομάζεται πίνακας κβαντισμού και διαφέρει για τα μπλοκ της χρωματικής συνιστώσας Y από τις υπόλοιπες (Cb ,

Cr). Οι πίνακες qTable που χρησιμοποιούνται στην παρούσα υλοποίηση είναι οι προτεινόμενοι πίνακες όπως ορίζονται από το πρότυπο.

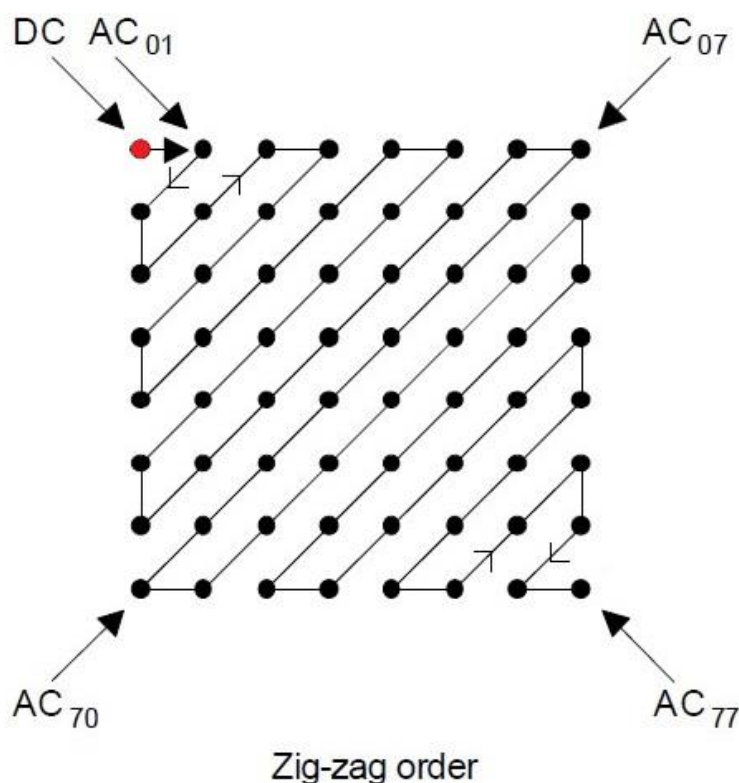
Η διαδικασία αποκβαντισμού των στοιχείων κάθε μπλοκ στον αποκωδικοποιητή γίνεται αντίστροφα,

$$dctBlock_{ij} = qScale \cdot qTable_{ij} \cdot qBlock_{ij}$$

Είναι φανερό πως η διαδικασία αυτή, όπως και εκείνη της υποδειγματοληψίας χρώματος, **εισάγει απώλειες** στη ποιότητα της εικόνας, καθώς δεν υπάρχει τρόπος ανάκτησης της πληροφορίας που χάθηκε λόγω της στρογγυλοποίησης.

2.4 Zig Zag Scanning και Run Length Encoding

Στο επόμενο στάδιο κάθε μπλοκ αναδιατάσσεται στη μορφή διανύσματος, ώστε να γίνει η απαραίτητη προετοιμασία για κωδικοποίηση. Η παραπάνω αναδιάταξη γίνεται μέσω μιας σάρωσης zig-zag όπως φαίνεται στο παρακάτω σχήμα.

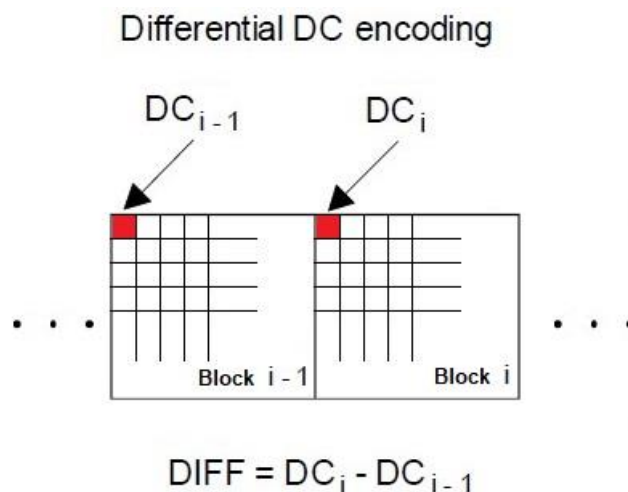


Σχήμα 2.6: Μετατροπή Μπλοκ σε Διάνυσμα 64 Στοιχείων. Με AC_{ij} συμβολίζεται ο συντελεστής AC στη θέση του μπλοκ (i, j).

Ο λόγος που συμβαίνει αυτό αφορά ένα βασικό χαρακτηριστικό των συντελεστών DCT. Οι συντελεστές DCT που προηγούνται, σε κάθε διάσταση (οριζόντια και κατακόρυφη), στο μετασχηματισμένο μπλοκ αντικατοπτρίζουν τις μικρότερες συχνότητες που εμφανίζονται στο αντίστοιχο μπλοκ εικόνας ενώ οι συντελεστές που έπονται αντικατοπτρίζουν τις μεγαλύτερες συχνότητες. Με τη σάρωση αυτή έχει παρατηρηθεί ότι το νέο διάνυσμα συχνά συμπληρώνεται με μεγάλες ακολουθίες μηδενικών στοιχείων οι οποίες θα κωδικοποιηθούν στη συνέχεια με κατάλληλο τρόπο για περεταίρω συμπίεση. Για τον ίδιο λόγο επίσης, αντί για την κωδικοποίηση του συντελεστή DC κάθε μπλοκ, γίνεται κωδικοποίηση της διαφοράς του εκάστοτε DC συντελεστή από το DC συντελεστή του προηγούμενου μπλοκ DCpred (DPCM). Ως αρχική τιμή του DCpred τίθεται η τιμή μηδέν για κάθε συνιστώσα Y, Cb και Cr.

Με το πέρας της παραπάνω διαδικασίας έχει δημιουργηθεί ένα διάνυσμα μήκους 64 στοιχείων, με πρώτο στοιχείο τη διαφορά κβαντίσμεων DC-DCpred και υπόλοιπα στοιχεία τους κβαντισμένους συντελεστές AC. Το διάνυσμα αυτό θα αναφέρεται στη συνέχεια ως διάνυσμα zig zag. Για την κωδικοποίηση των τελευταίων 63 στοιχείων ακολουθείται μια παραλλαγή της κωδικοποίησης Run Length Encoding (RLE). Συγκεκριμένα, η παραλλαγή αυτή κωδικοποιεί το παραπάνω διάνυσμα σε ένα πίνακα διάστασης $Z \times 2$, ο οποίος στη συνέχεια θα αναφέρεται ως πίνακας rle.

Η πρώτη γραμμή του πίνακα αυτού συμπληρώνεται με ένα μηδενικό στοιχείο ακολουθούμενο από τη DC διαφορά που αναφέρθηκε προηγουμένως.



Σχήμα 2.7: Εξαγωγή της DC διαφοράς ανάμεσα σε δύο διαδοχικά μπλοκ.

Για τις υπόλοιπες γραμμές η διαδικασία συμπλήρωσης είναι η εξής:

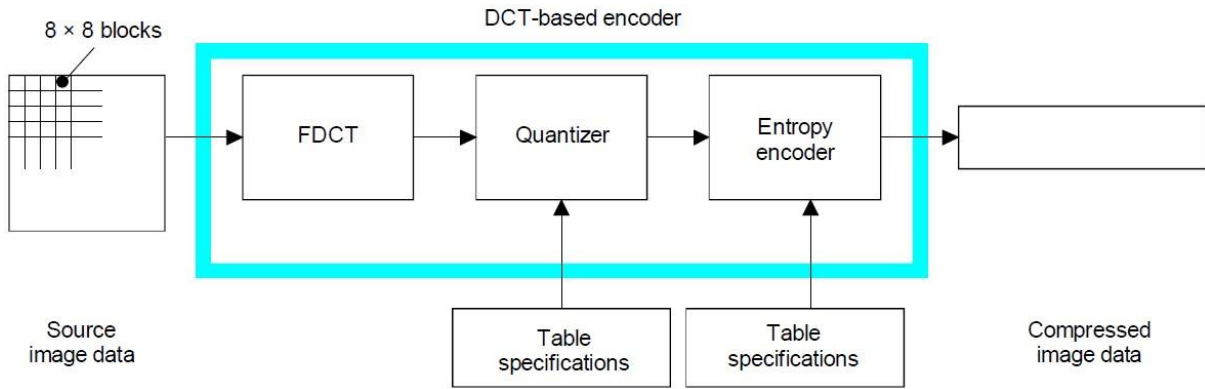
Γίνεται καταμέτρηση των συνεχόμενων μηδενικών στοιχείων του διανύσματος zig zag μέχρι την εμφάνιση ενός μη μηδενικού στοιχείου. Το

πλήθος που υπολογίστηκε εισάγεται στην πρώτη στήλη του πίνακα ενώ το μη μηδενικό στοιχείο εισάγεται στη δεύτερη στήλη. Η διαδικασία αυτή επαναλαμβάνεται μέχρι το τέλος του διανύσματος zig zag. Στην περίπτωση που η ακολουθία μηδενικών στοιχείων φτάνει σε πλήθος τον αριθμό 16, η γραμμή του πίνακα rle συμπληρώνεται από το διάνυσμα [15 0] και η καταμέτρηση των μηδενικών μηδενίζεται. Η περίπτωση αυτή θα αναφέρεται στη συνέχεια ως ZRL (Zero Run Length). Τέλος, μετά την καταμέτρηση του τελευταίου μη μηδενικού συντελεστή DCT ο πίνακας rle σταματάει να συμπληρώνεται ενώ προστίθεται ως τελευταία γραμμή το διάνυσμα [0 0] που υποδηλώνει το τέλος του μπλοκ EOB (End Of Block). (εκτός αν το τελευταίο στοιχείο του block είναι μη μηδενικό)

Για την αποκωδικοποίηση του πίνακα rle και τη μετατροπή του πίσω σε κβαντισμένα μπλοκ ακολουθείται η αντίστροφη διαδικασία μετατρέποντας αρχικά τον πίνακα rle σε ένα διάνυσμα 64 θέσεων, υπολογίζοντας το DC όρο με βάση τον προηγούμενο και τέλος, ανακατασκευάζοντας το κβαντισμένο μπλοκ με zig zag σειρά.

2.5 Huffman Encoding

Τέλος, ο πίνακας rle κωδικοποιείται και αυτός με τη σειρά του με τη μορφή ενός bit-Stream, βάση της κωδικοποίησης Huffman. Για την παραγωγή του κώδικα Huffman χρησιμοποιούνται προκαθορισμένοι πίνακες που ορίζει το πρότυπο και οι οποίοι διαφέρουν αφενός ανάμεσα στις DC διαφορές με τους AC συντελεστές και αφετέρου ανάμεσα στα μπλοκ τύπου Y με τα μπλοκ τύπου Cb, Cr. Οι πίνακες αυτοί εισάγονται στο περιβάλλον MATLAB με τη μορφή δύο πινάκων cell για τις DC διαφορές και τους AC συντελεστές αντίστοιχα, καθένας από τους οποίους περιέχει τους τρεις πίνακες κωδικοποίησης, έναν για τη φωτεινότητα και δύο (ίδιοι μεταξύ τους) για τις χρωματικότητες. Οι πίνακες αυτοί περιέχουν τις δυαδικές κωδικές λέξεις σε μορφή String οι οποίες κατά την κωδικοποίηση μετατρέπονται σε διάνυσμα γραμμής double με τιμές 0 και 1.

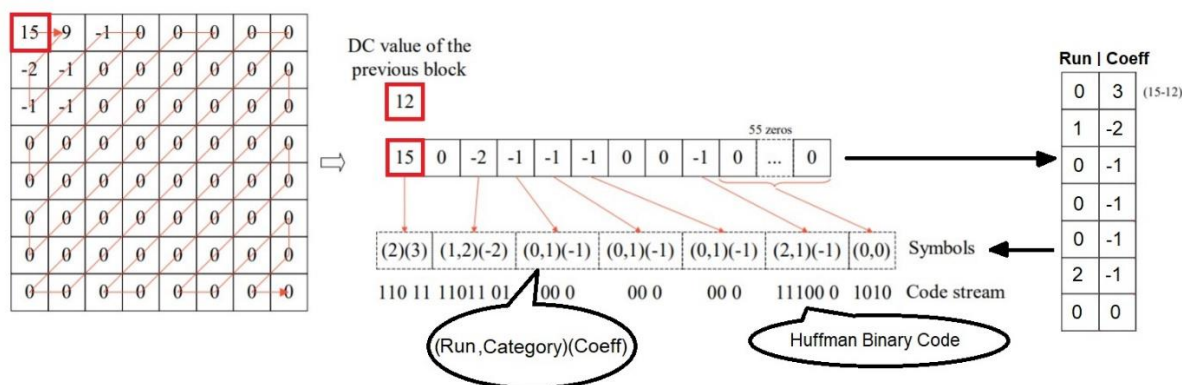


Σχήμα 2.8: Διαδικασία Κωδικοποίησης Μπλοκ Εικόνας.

Η διαδικασία κωδικοποίησης διαφέρει για τις DC διαφορές και τους AC συντελεστές. Για ένα πίνακα *rle* που αντιστοιχεί σε ένα κβαντισμένο μπλοκ η διαδικασία είναι η εξής:

Για τη DC διαφορά αρχικά εντοπίζεται η κατηγορία στην οποία ανήκει. Η κατηγορία αυτή προκύπτει ως το πλήθος των bit του αντίστοιχου δυαδικού αριθμού. Εξάιρεση αποτελεί ο αριθμός μηδέν ο οποίος ανήκει στην κατηγορία 0. Για τους αρνητικούς αριθμούς χρησιμοποιείται η μορφή αναπαράστασης συμπληρώματος του 1 (One's Complement) προτού γίνει ο εντοπισμός της κατηγορίας τους. Η κατηγορία που εντοπίστηκε κωδικοποιείται με βάση τον κατάλληλο πίνακα κωδικοποίησης Huffman και προστίθεται στο bit-Stream ακολουθούμενη από τη δυαδική αναπαράσταση της κβαντισμένης διαφοράς DC.

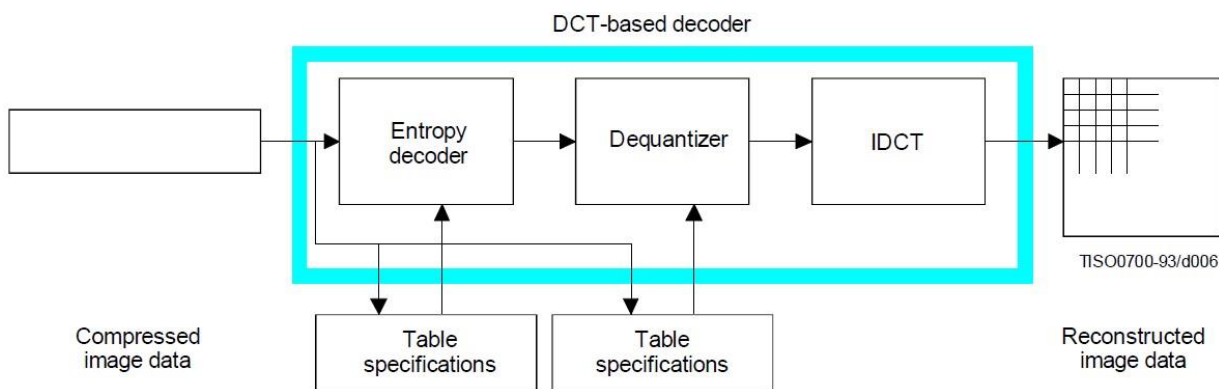
Για τους συντελεστές AC αρχικά εντοπίζεται με τον ίδιο τρόπο η κατηγορία στην οποία ανήκουν. Με βάση αυτή καθώς και το μέγεθος των μηδενικών ακολουθιών που βρίσκεται στην πρώτη στήλη του πίνακα *rle* αναζητείται στον αντίστοιχο πρότυπο πίνακα η δυαδική λέξη που κωδικοποιεί το ζεύγος αυτό. Η λέξη αυτή προστίθεται στο bit-Stream ακολουθούμενη και πάλι από τη δυαδική αναπαράσταση του συντελεστή προς κωδικοποίηση (δεύτερη στήλη του πίνακα *rle*).



Σχήμα 2.9: Μετατροπή Κβαντισμένου Μπλοκ σε bit-Stream.

Σημειώνεται ότι για την παραπάνω διαδικασία χρησιμοποιείται η **ολική μεταβλητή blockType**, για την αναγνώριση των μπλοκ ως προς τον τύπο τους (φωτεινότητα - χρωματικότητα).

Η αντίθετη διαδικασία ακολουθείται κατά την αποκωδικοποίηση.



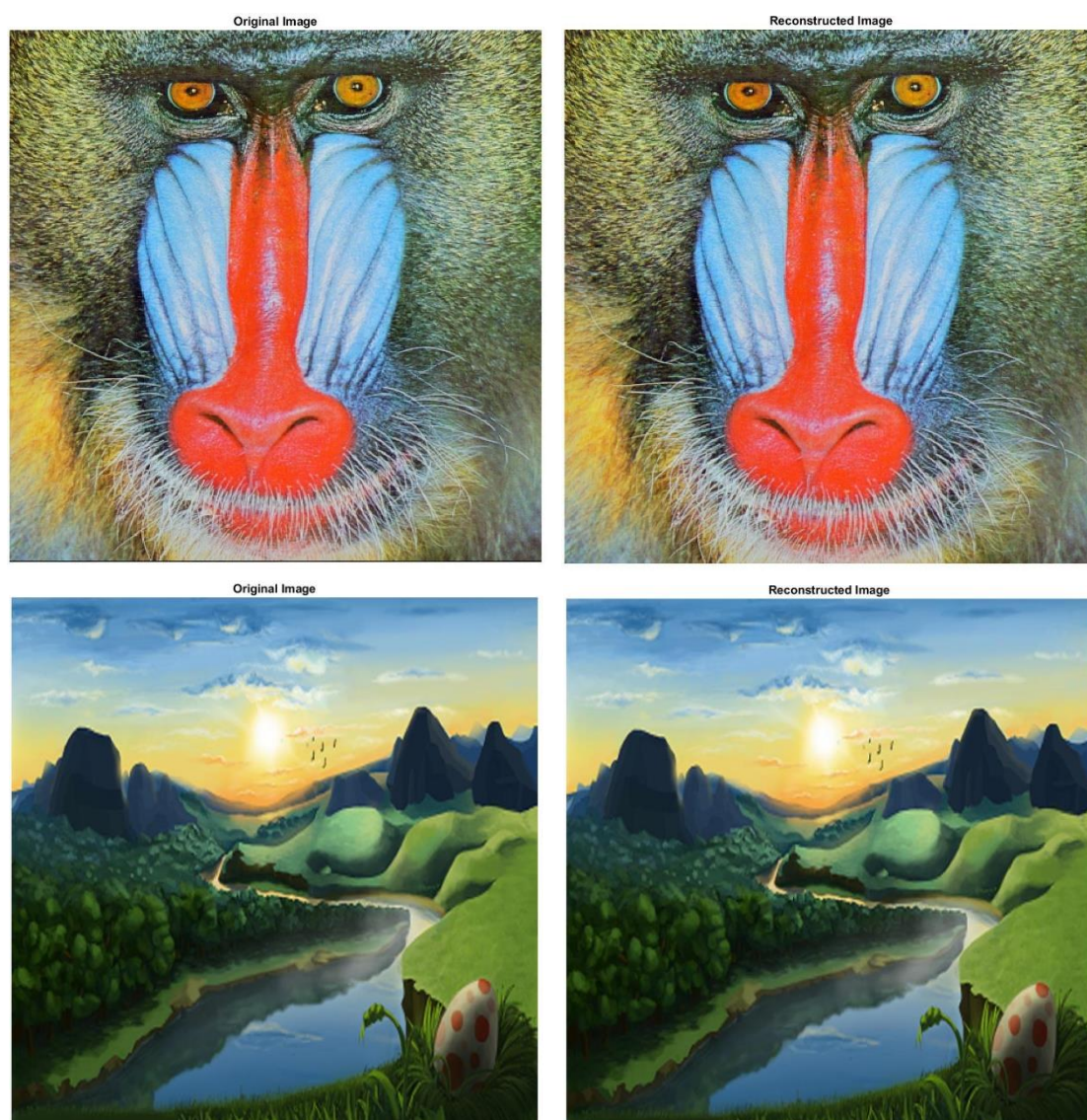
Σχήμα 2.10: Διαδικασία Αποκωδικοποίησης Μπλοκ Εικόνας.

Συγκεκριμένα, χρησιμοποιούνται δύο δείκτες (start και stop) οι οποίοι αρχικά ξεκινούν και οι δύο από την πρώτη θέση του bit-Stream. Ο δείκτης stop ξεκινάει να αυξάνει σταδιακά ελέγχοντας κάθε φορά αν υπάρχει μοναδική ακολουθία bits ανάμεσα στους δείκτες start και stop. Σε περίπτωση που αυτό συμβαίνει, τότε από το δείκτη του στοιχείου που βρέθηκε στον πρότυπο πίνακα εξάγεται η κατηγορία και ο συντελεστής του πίνακα rle. Μόλις αυτό συμβεί ο δείκτης start μετατοπίζεται ώστε να συμπίπτει με το δείκτη stop. Η διαδικασία επαναλαμβάνεται για όλο το bit-Stream.

2.6 Συνάρτηση Επίδειξης - Demo 1

Με την ολοκλήρωση των παραπάνω συναρτήσεων ακολουθεί μια συνάρτηση επίδειξης της λειτουργίας τους. Στο πρώτο ερώτημα ζητείται η μετατροπή δύο εικόνων από RGB σε YCbCr με υποδειγματοληψία χρώματος 4:2:2 και 4:4:4 αντίστοιχα και η ανακατασκευή τους μέσω της αντίστροφης συνάρτησης. Στο δεύτερο ερώτημα ζητείται η μετατροπή τους σε YCbCr με ίδια υποδειγματοληψία χρώματος, έπειτα η εφαρμογή του DCT στα διάφορα μπλοκ, ο κβαντισμός αυτών ($qScale = 0.6$ και 5 αντίστοιχα) και τέλος η ανακατασκευή τους μέσω της αντίστροφης διαδικασίας.

Τα αποτελέσματα για το πρώτο ερώτημα φαίνονται στη συνέχεια:



Σχήμα 2.11: Παραδείγματα Μετατροπής εικόνας RGB σε YCbCr και επαναφορά της σε RGB.

Και για το δεύτερο ερώτημα:



Σχήμα 2.12: Παραδείγματα Μετατροπής εικόνας RGB σε YCbCr, εφαρμογής μετασχηματισμού DCT, κβαντισμού καθώς και της αντίστροφης διαδικασίας.

Δηλαδή η ανακατασκευή και στις δύο περιπτώσεις είναι σχεδόν άρτια παρόλες τις απώλειες που υπάρχουν λόγω της υποδειγματοληψίας χρώματος και του κβαντισμού.

3. Τμήμα B – JPEG Integration

Το επόμενο τμήμα έχει ως στόχο την ενσωμάτωση συναρτήσεων που υλοποιήθηκαν στο προηγούμενο τμήμα σε μια γενικότερη συνάρτηση και την εξαγωγή ποσοτικών και ποιοτικών συμπερασμάτων για τη συμπίεση που επιτεύχθηκε.

3.1 JPEG Encoder/Decoder

Συγκεκριμένα, δημιουργείται ένας πίνακας τύπου cell, κάθε στοιχείο του οποίου είναι μια δομή τύπου struct.

Το πρώτο struct του cell αυτού περιέχει όλους τους πίνακες που αφορούν τα μπλοκ της εικόνας, όπως οι πίνακες κβαντισμού qTableL, qTableC για τη φωτεινότητα Y και τις χρωματικότητες Cb, Cr αντίστοιχα (για την ακρίβεια περιέχει το γινόμενο qScale · qTableL και qScale · qTableC αντίστοιχα) και τους πίνακες για την κωδικοποίηση Huffman, DCL, ACL, DCC και ACC για τη φωτεινότητα και τις χρωματικότητες αντίστοιχα, στην περίπτωση κωδικοποίησης DC διαφορών ή AC συντελεστών.

Τα υπόλοιπα cells του πίνακα περιέχουν για κάθε μπλοκ τα χαρακτηριστικά blkType με τον τύπο του εκάστοτε μπλοκ (Y, Cb, Cr), τη θέση του στην εικόνα μέσω των indHor, indVer δεικτών και το bit-Stream που δημιουργήθηκε από τη συνάρτηση huffmanEncode της προηγούμενης βαθμίδας.

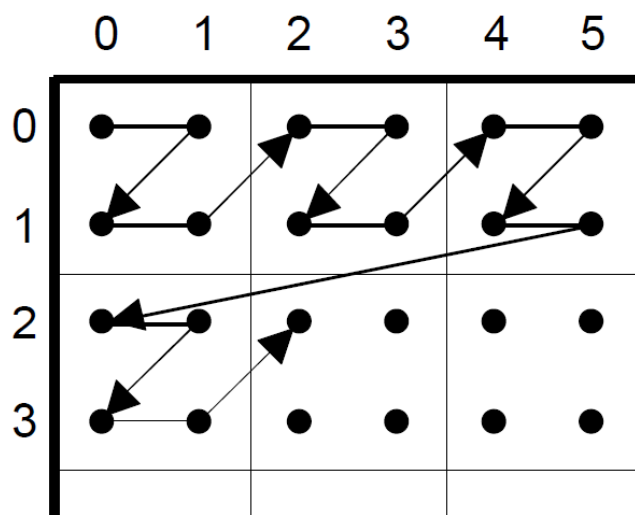
<u>struct</u> with fields:	<u>struct</u> with fields:
qTableL: [8×8 double]	blkType: "Y"
qTableC: [8×8 double]	indHor: 1
DCL: [12×1 string]	indVer: 1
DCC: [12×1 string]	huffStream: [1×338 double]
ACL: [162×1 string]	
ACC: [162×1 string]	

Σχήμα 3.1: Δομές που περιέχονται στον πίνακα τύπου Cell. Η δομή στα αριστερά βρίσκεται στο πρώτο κελί του πίνακα Cell ενώ στα δεξιά φαίνεται η μορφή της δομής στα υπόλοιπα κελιά.

Η παραπάνω διαδικασία ξεκινάει με τη μετατροπή της εικόνας RGB σε YCbCr για κάποια υποδειγματοληψία χρώματος από αυτές που αναφέρθηκαν προηγουμένως. Έπειτα μέσω μιας τριπλά εμφολευμένης επαναληπτικής διαδικασίας για κάθε μπλοκ 8x8 και κάθε συνιστώσα (Y ή Cb ή Cr) ακολουθούν οι διαδικασίες μετασχηματισμού DCT, κβαντισμού

του μπλοκ για τη ζητούμενη τιμή $qScale$, μετατροπής σε πίνακα Run Length Encoding όπως περιεγράφηκε και τέλος παραγωγής του ζητούμενου bit-Stream μέσω της `huffmanEncode` συνάρτησης.

Σημειώνεται ότι στην περίπτωση υποδειγματοληψίας χρώματος 4:2:0, η σειρά με την οποία κωδικοποιούνται τα μπλοκ Y δεν είναι η συμβατική, αλλά αυτή που φαίνεται στο παρακάτω σχήμα.



Σχήμα 3.2: Σειρά κωδικοποίησης μπλοκ Y, στην περίπτωση υποδειγματοληψίας 4:2:0.

Για τη διαδικασία της αποκωδικοποίησης ακολουθείται η παραπάνω διαδικασία με την αντίστροφη σειρά χρησιμοποιώντας τις αντίστροφες συναρτήσεις που υλοποιήθηκαν.

Σημειώνεται ότι για την εύρεση της κατηγορίας υποδειγματοληψίας χρώματος κατά την αποκωδικοποίηση γίνεται σύγκριση μεγεθών των πινάκων Y, Cb, Cr μετά τον από-κβαντισμό. Επιπλέον, ο απο-κβαντισμός γίνεται πάντοτε με τιμή $qScale=1$ μιας και η σταθερά αυτή έχει ενσωματωθεί στον πίνακα κβαντισμού `qTable` που λαμβάνεται από το πρώτο struct του cell, `JPEGen`.

3.2 Καταγραφή Αποτελεσμάτων

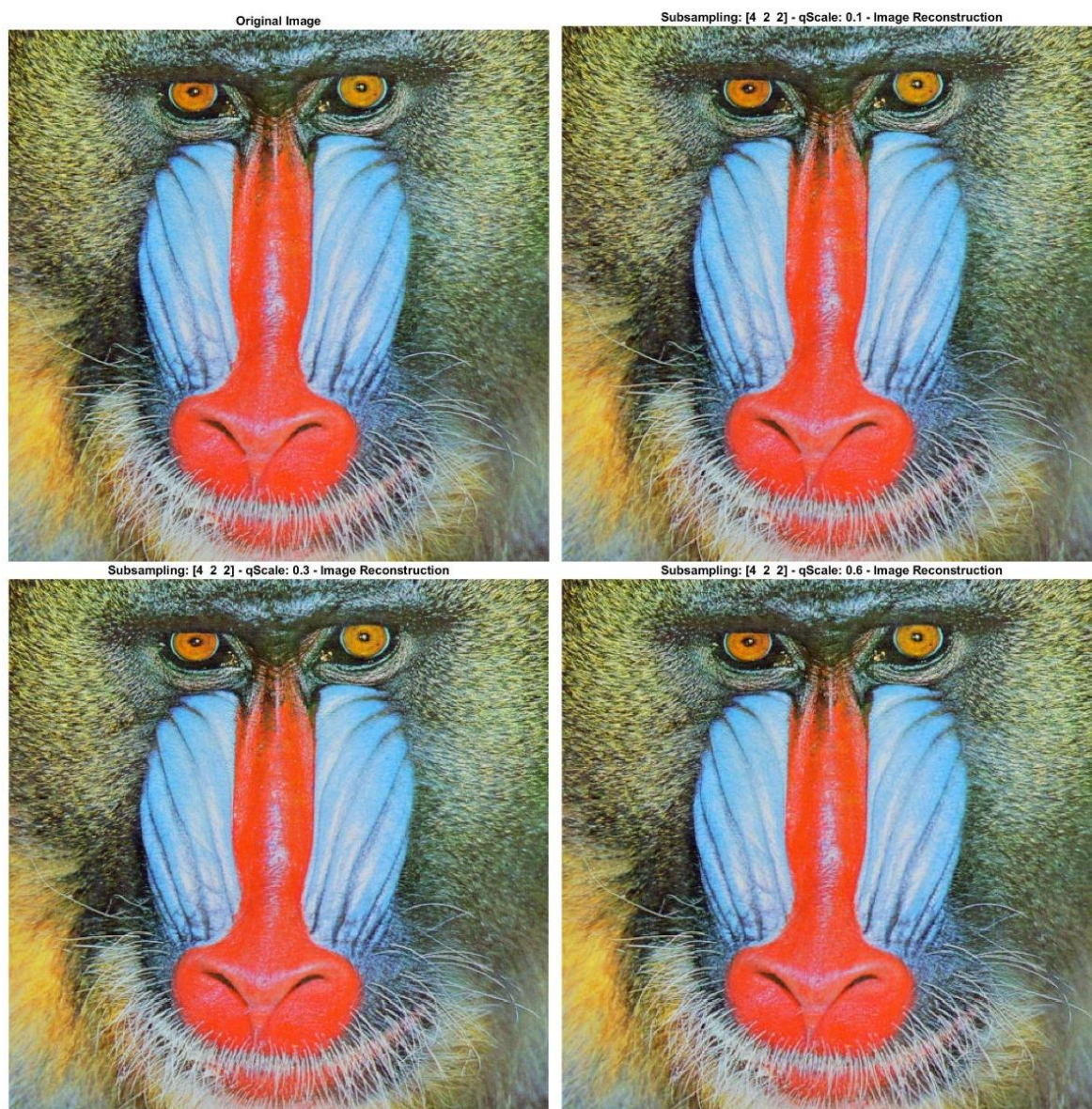
Στη συνέχεια, χρησιμοποιείται η παραπάνω συνάρτηση για την κωδικοποίηση των δοσμένων εικόνων που παρουσιάστηκαν στο Demo1 και την εξαγωγή χρήσιμων συμπερασμάτων για τη λειτουργικότητα του κωδικοποιητή. Συγκεκριμένα, η καταγραφή των αποτελεσμάτων διακρίνεται σε δύο τμήματα.

Το πρώτο τμήμα περιλαμβάνει την κωδικοποίηση των δύο εικόνων που χρησιμοποιήθηκαν προηγουμένως για διαφορετικές τιμές της παραμέτρου $qScale$ κατά τον κβαντισμό. Η υποδειγματοληψία χρώματος για κάθε εικόνα γίνεται με τις ίδιες παραμέτρους που χρησιμοποιήθηκαν στο Demo1. Οι τιμές $qScale$ που εξετάζονται είναι:

$$qScale = [0.1 \quad 0.3 \quad 0.6 \quad 1 \quad 2 \quad 5 \quad 10]$$

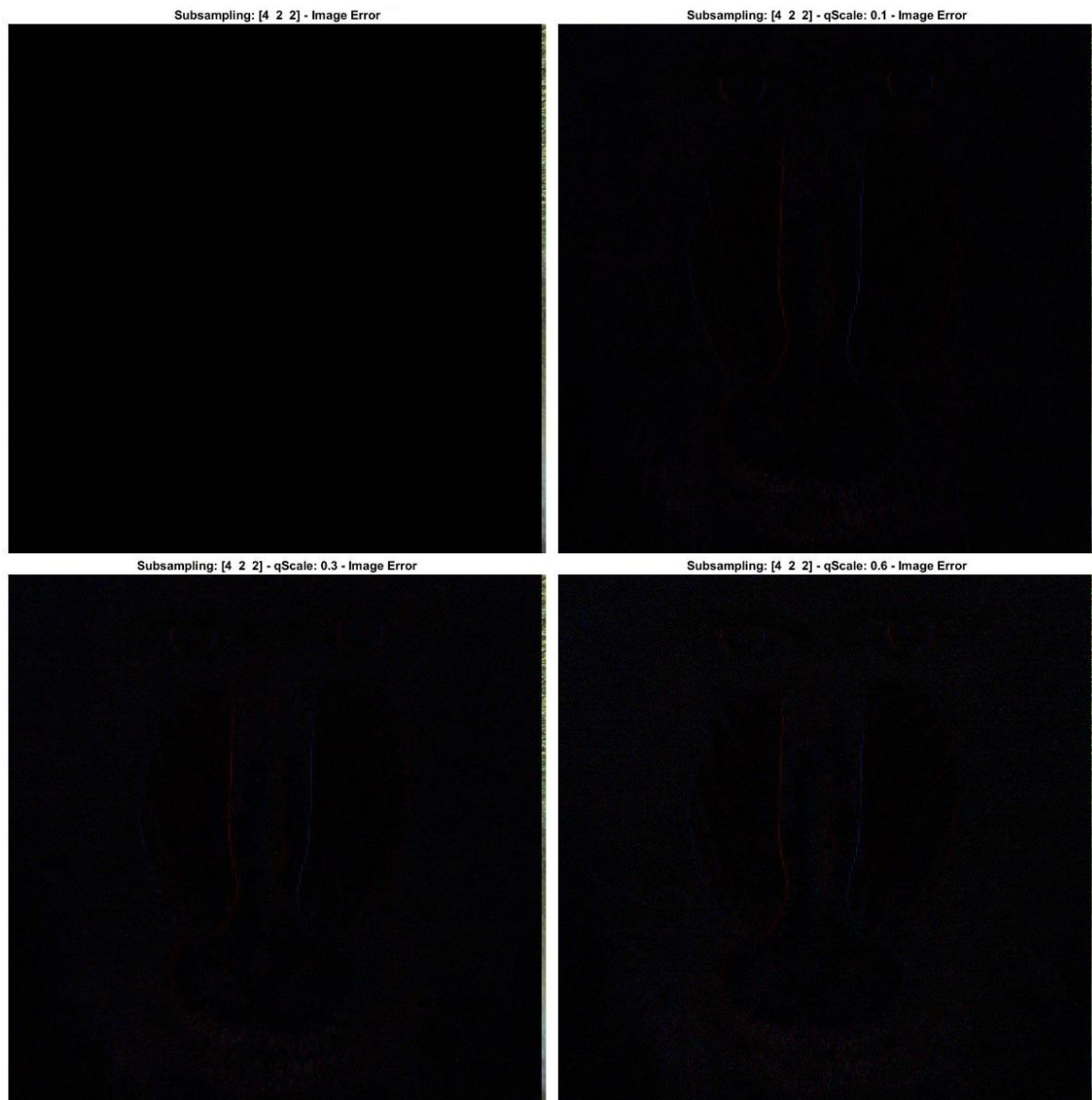
Επίσης εξάγονται οι τιμές του Μέσου Τετραγωνικού Σφάλματος (Mean Square Error - MSE), το πλήθος των bits του bit-Stream κάθε εικόνας και ο λόγος συμπίεσης σε κάθε περίπτωση.

Τα παραπάνω αποτελέσματα φαίνονται στη συνέχεια:



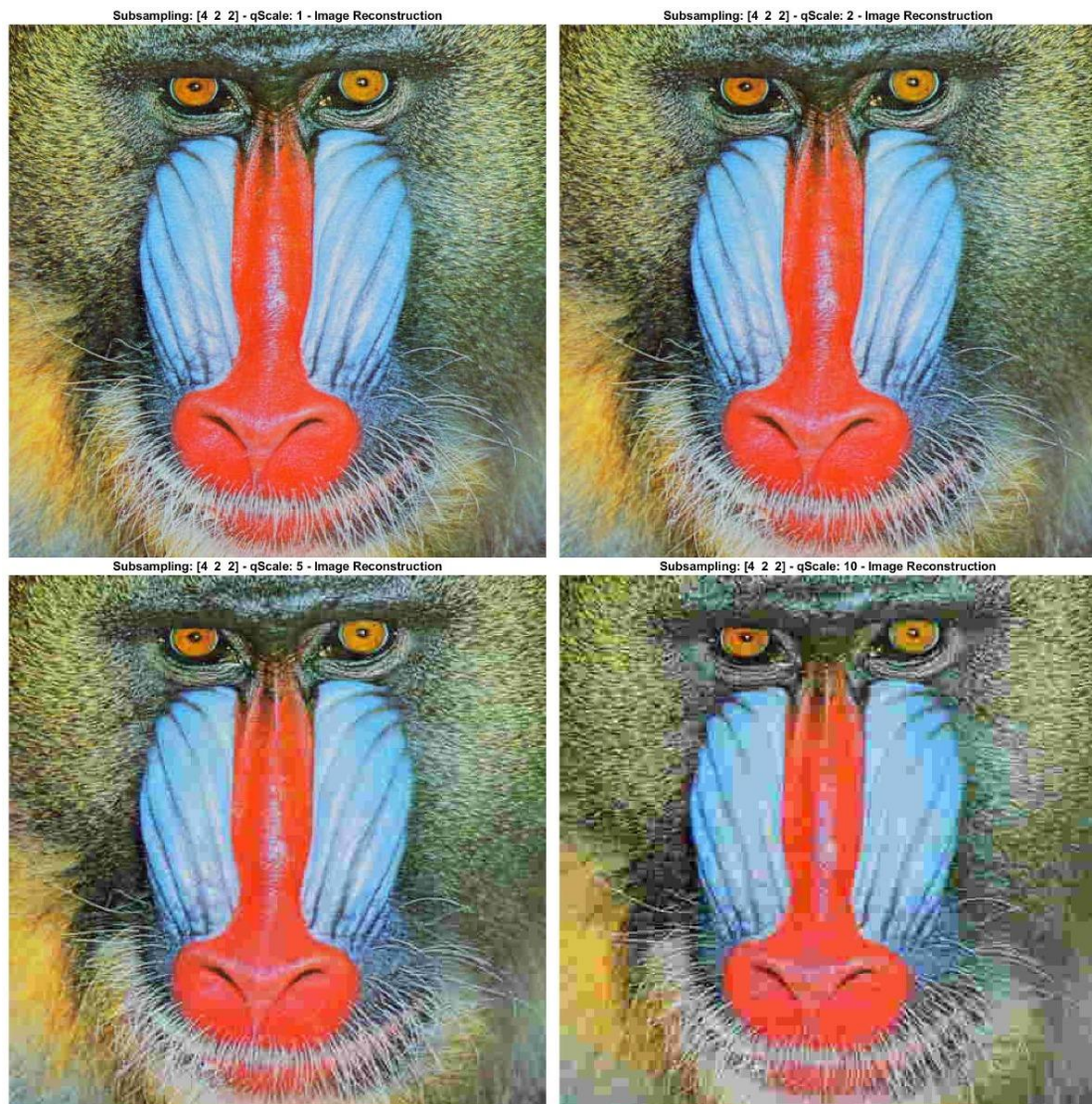
Σχήμα 3.3: Αποτελέσματα - Εικόνα 1 για τιμές $qScale$: 0.1 , 0.3 , 0.6.

Όπου αρχικά φαίνεται η πρότυπη καθώς και οι εικόνες για τις πρώτες τρεις από τις ζητούμενες επιλογές παραμέτρων $qScale$ που εξετάζονται. Ακολουθούν οι αντίστοιχες εικόνες σφαλμάτων:



Σχήμα 3.4: Σφάλμα - Εικόνα 1 για τιμές $qScale$: 0.1 , 0.3 , 0.6.

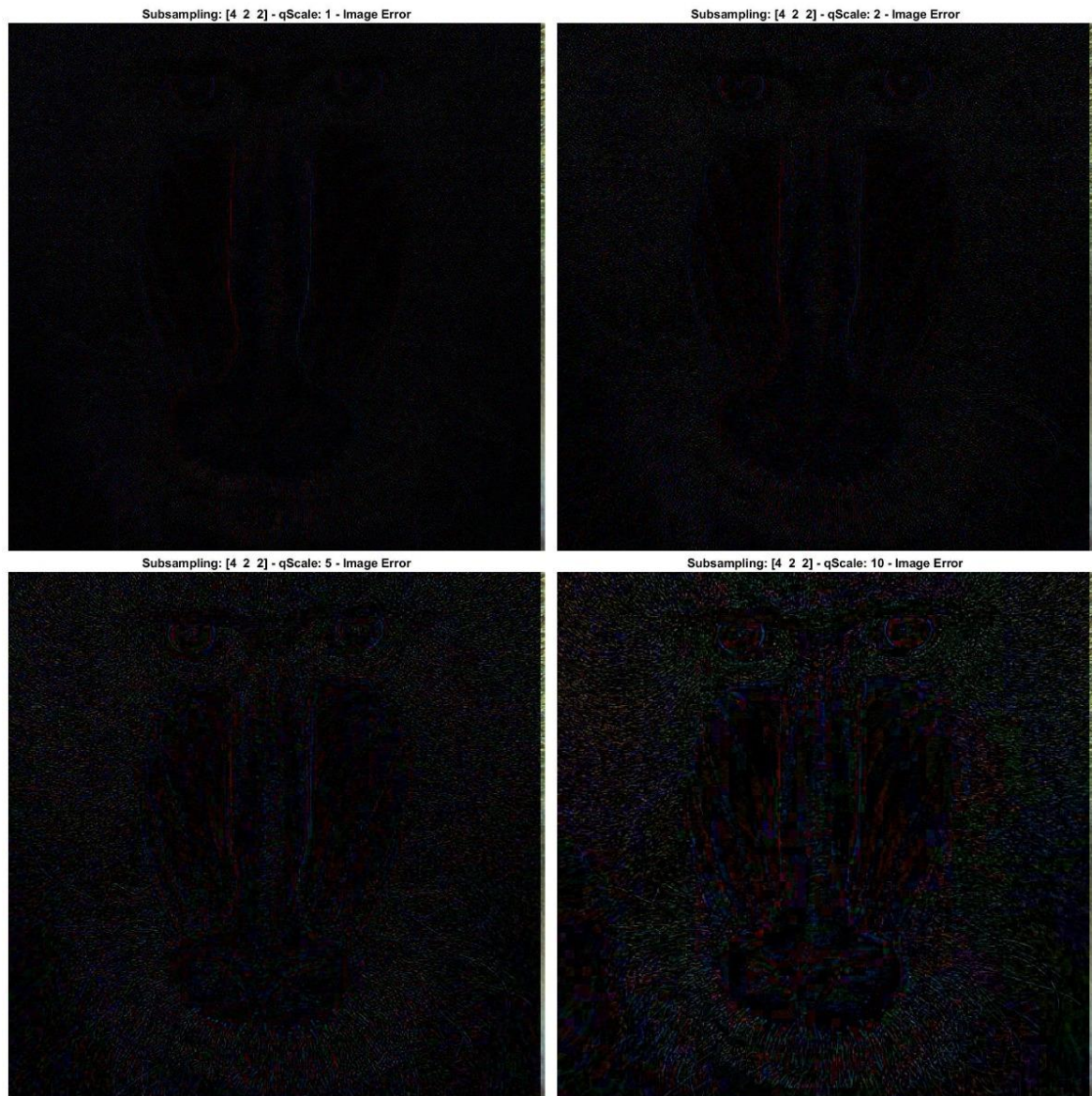
Από τις οποίες φαίνεται ότι το σφάλμα είναι από μηδενικό (πρώτη εικόνα) έως πολύ μικρό (υπόλοιπες εικόνες). Στη συνέχεια ακολουθούν και οι υπόλοιπες εικόνες για τιμές $qScale$.



Σχήμα 3.5: Αποτελέσματα - Εικόνα 1 για τιμές qScale: 1, 2, 5 και 10.

Βλέπουμε ότι η ποιότητα των εικόνων αυτών είναι χαμηλότερη από τις προηγούμενες. Αυτό επιπλέον επιβεβαιώνεται και από τις αντίστοιχες εικόνες σφαλμάτων οι οποίες περιέχουν περισσότερη πληροφορία σε σχέση με προηγούμενως.

Οι εικόνες αυτές φαίνονται στη συνέχεια:



Σχήμα 3.6: Σφάλμα - Εικόνα 1 για τιμές $qScale$: 1 , 2 , 5 και 10.

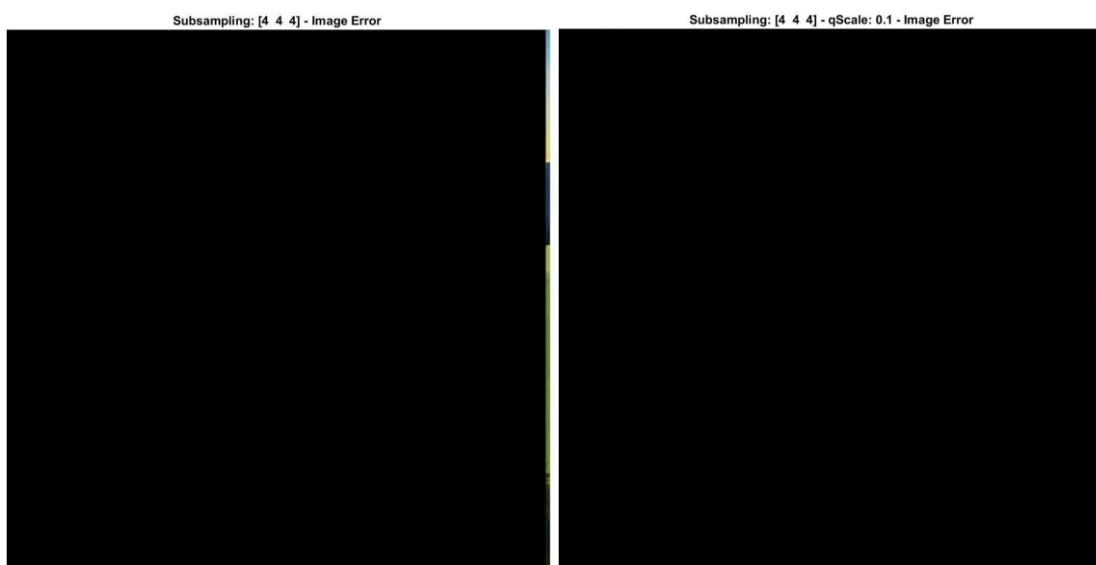
Παρατηρείται, λοιπόν, ότι καθώς αυξάνεται η τιμή της παραμέτρου $qScale$, αυξάνονται και οι απώλειες-σφάλμα. Ωστόσο, το μέγεθος του bit-Stream μειώνεται (δηλαδή αυξάνεται ο λόγος συμπίεσης).

Όμοια για τη δεύτερη εικόνα ακολουθούν τα αντίστοιχα πειραματικά αποτελέσματα:

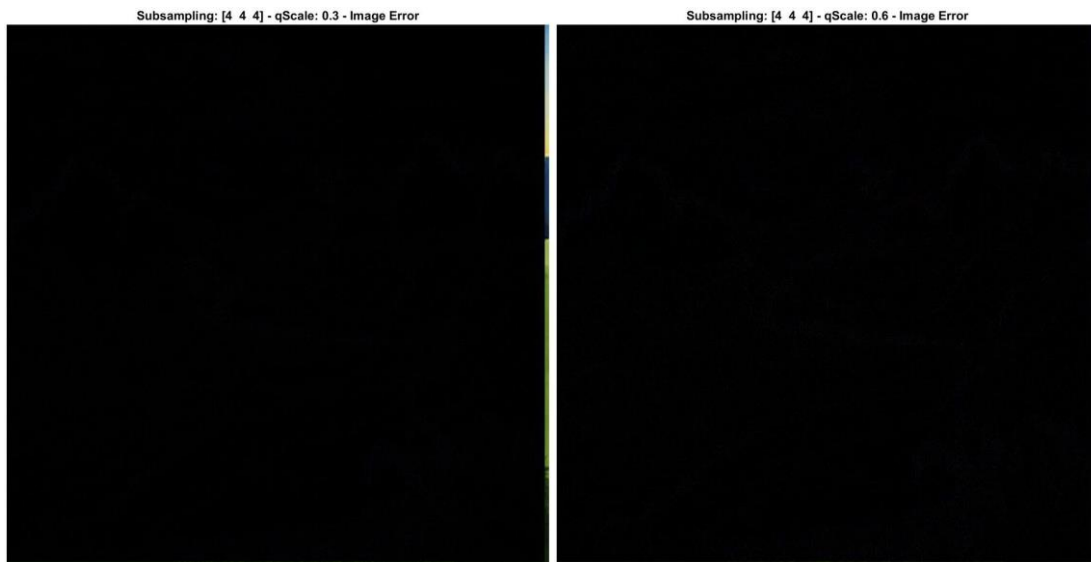
Κωδικοποίηση – Αποκωδικοποίηση JPEG



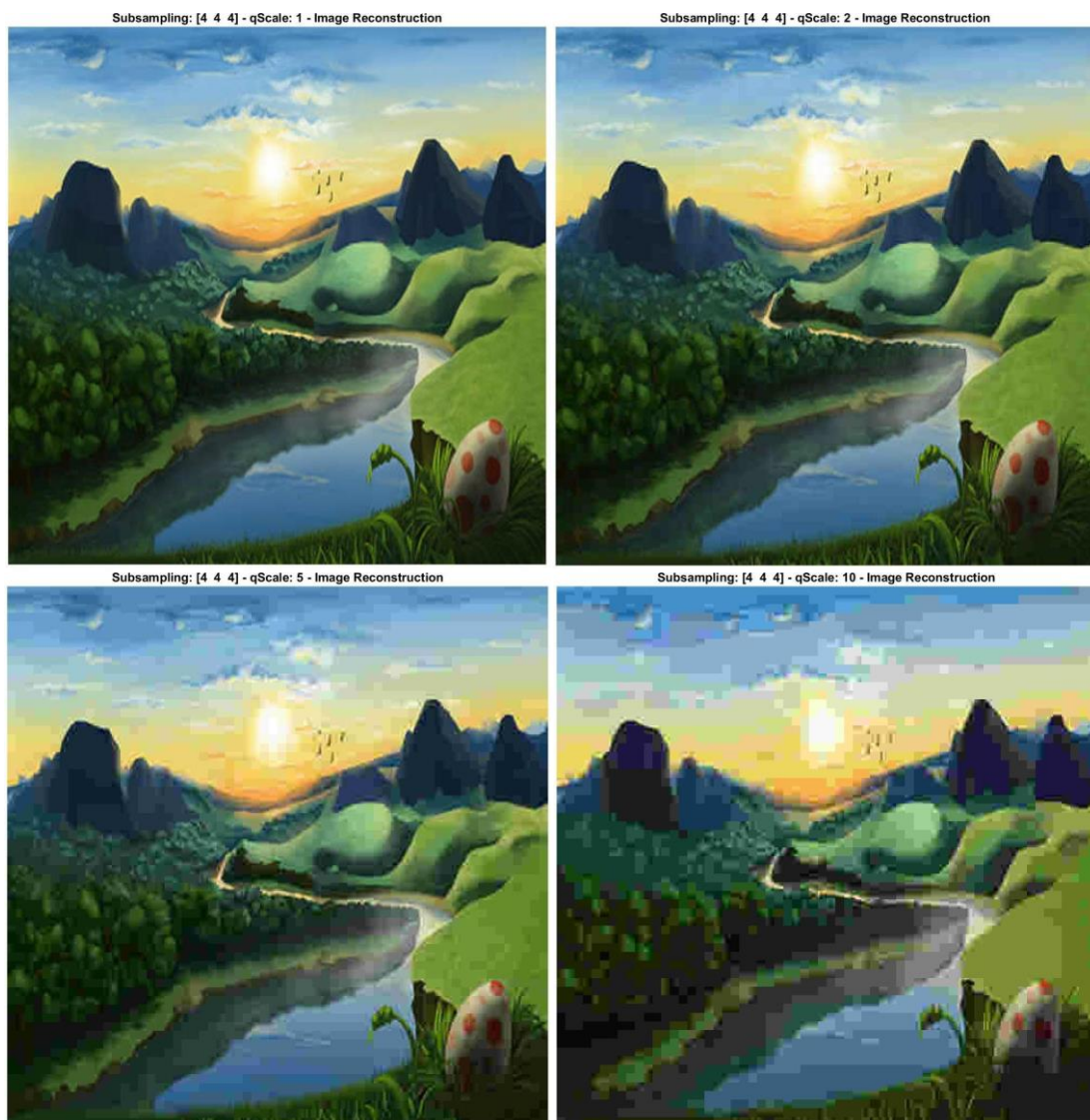
Σχήμα 3.7: Αποτελέσματα - Εικόνα 2 για τιμές $qScale$: 0.1 , 0.3 , 0.6.



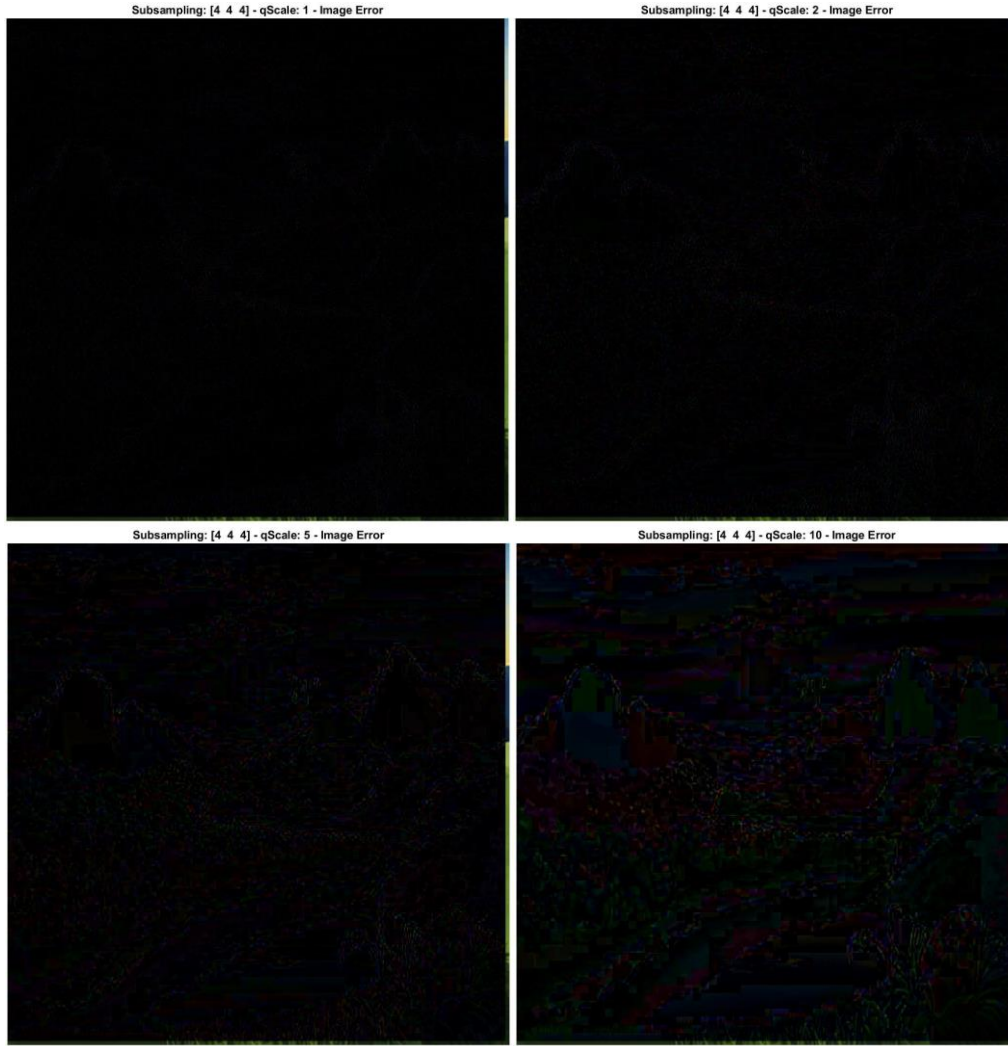
Κωδικοποίηση – Αποκωδικοποίηση JPEG



Σχήμα 3.8: Σφάλμα - Εικόνα 2 για τιμές $qScale$: 0.1 , 0.3 , 0.6.



Σχήμα 3.9: Αποτελέσματα - Εικόνα 2 για τιμές $qScale$: 1 , 2 , 5 και 10.



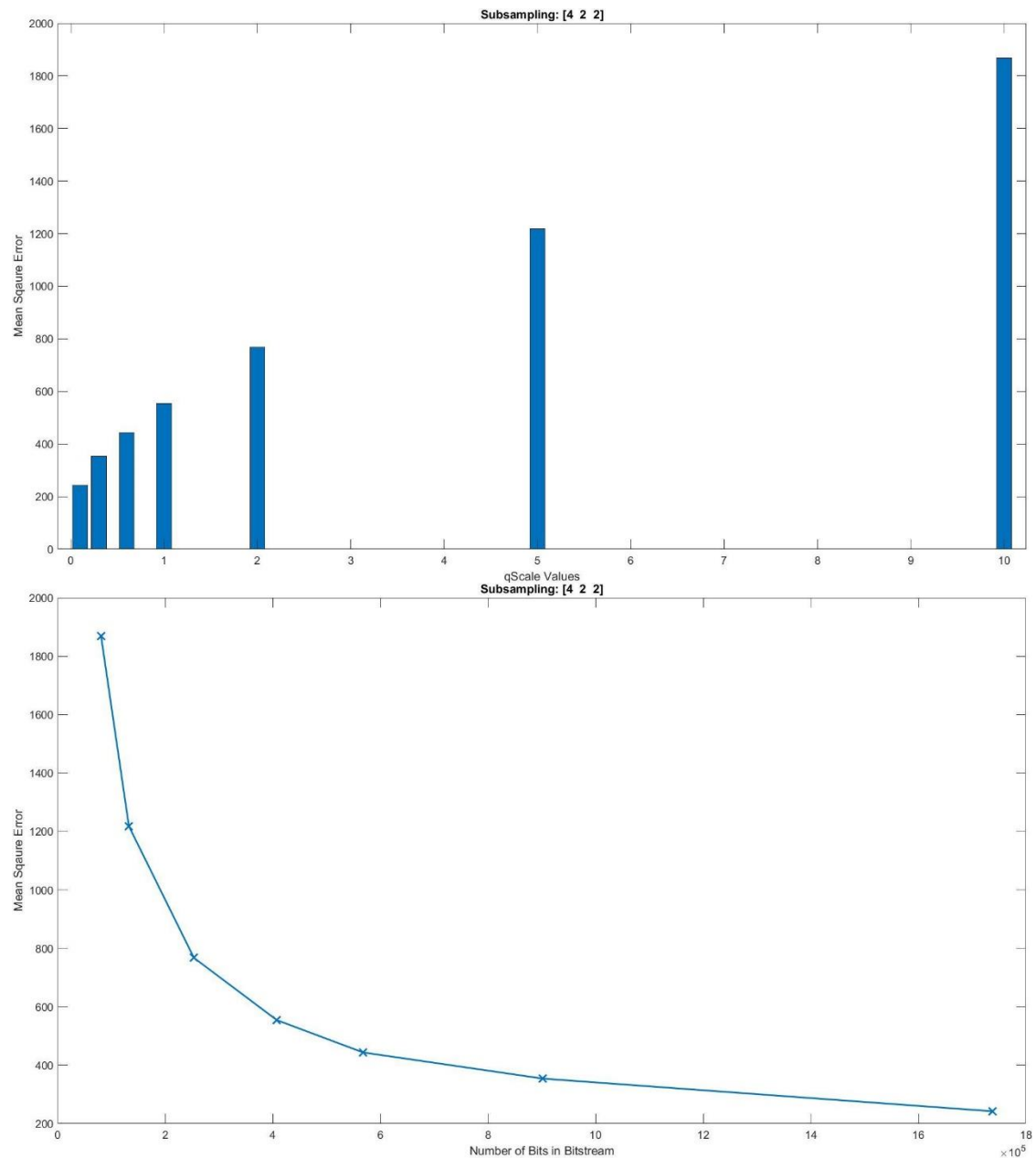
Σχήμα 3.10: Σφάλμα - Εικόνα 2 για τιμές qScale: 1 , 2 , 5 και 10.

Μπορεί, επομένως, να εξαχθεί το συμπέρασμα ότι το τίμημα της υψηλής συμπίεσης είναι η μείωση της ποιότητας του προς συμπίεση αρχείου αλλά και το αντίστροφο. Μικρές απώλειες, έχουν υψηλότερο κόστος σε bits, δηλαδή μικρότερο λόγο συμπίεσης.

Στη συνέχεια ακολουθούν οι γραφικές παραστάσεις του μέσου τετραγωνικού σφάλματος συναρτήσει των διάφορων παραμέτρων qScale καθώς και του αριθμού bits για κάθε περίπτωση qScale κάθε εικόνας, οι οποίες επαληθεύουν το παραπάνω συμπέρασμα. Πρέπει να σημειωθεί ότι το συνολικό MSE λαμβάνεται ως το άθροισμα των MSE κάθε συνιστώσας:

$$MSE = \frac{1}{numel(R)} \cdot \sum_{i=1}^{numel(R)} (R_i - \hat{R}_i)^2 + \frac{1}{numel(G)} \cdot \sum_{i=1}^{numel(G)} (G_i - \hat{G}_i)^2 + \frac{1}{numel(B)} \cdot \sum_{i=1}^{numel(B)} (B_i - \hat{B}_i)^2$$

Κωδικοποίηση – Αποκωδικοποίηση JPEG

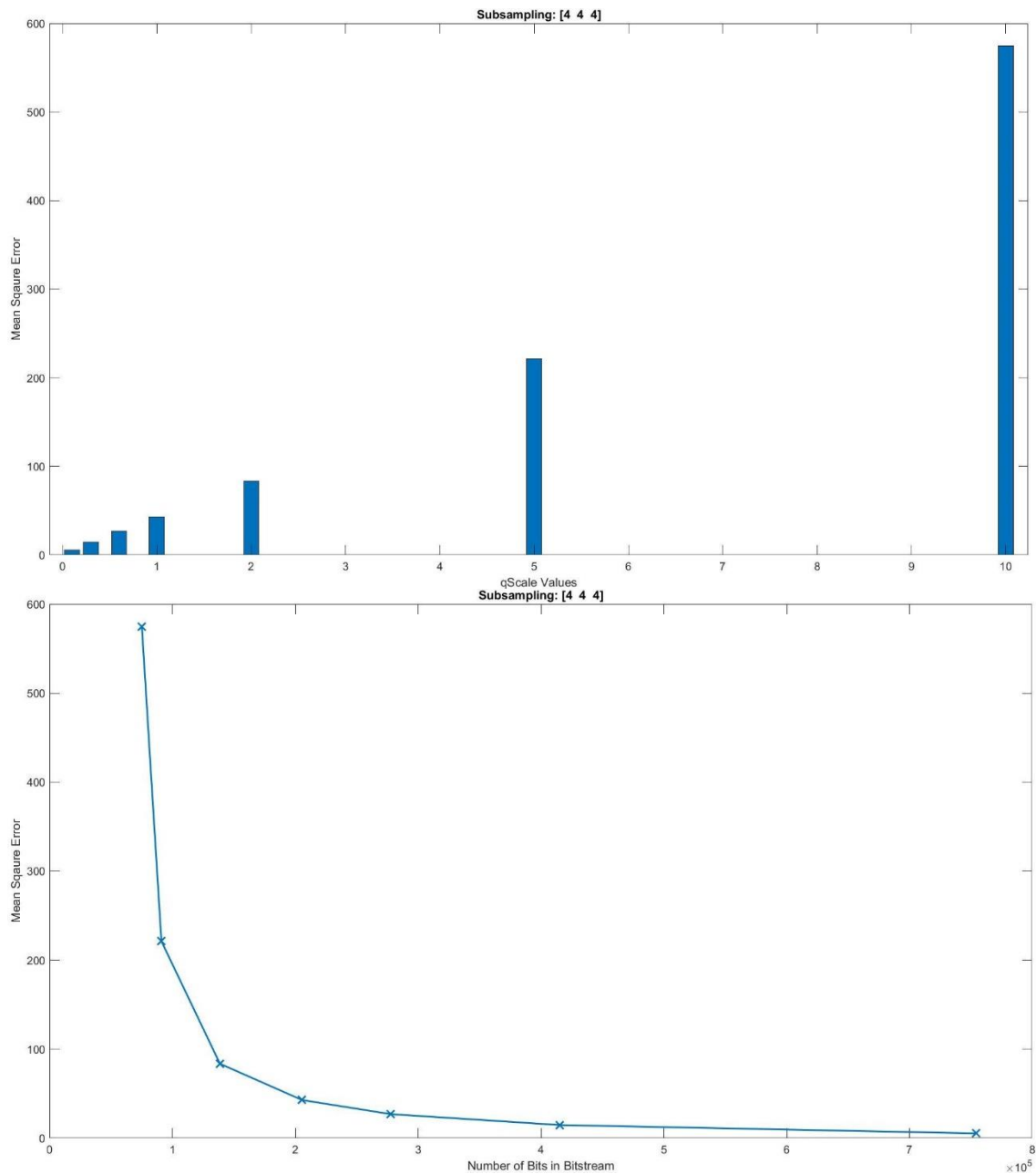


Σχήμα 3.11: Μέσο Τετραγωνικό σφάλμα συναρτήσσει της παραμέτρου $qScale$ και του αριθμού Bits στο παραγόμενο bitstream για την Εικόνα 1.

qScale	Compression Ratio	Mean Square Error	Number of bits in bitsream
0.1	3.397563	242.174261	1737829
0.3	6.546541	354.027779	901909
0.6	10.402881	443.619370	567572
1.0	14.497967	554.340604	407256
2.0	23.317671	767.466978	253215
5.0	44.598414	1217.922200	132390
10.0	73.209061	1868.459775	80651

Πίνακας 3.1: Αποτελέσματα - Εικόνα 1.

Κωδικοποίηση – Αποκωδικοποίηση JPEG



Σχήμα 3.12: Μέσο Τετραγωνικό σφάλμα συναρτήσει της παραμέτρου $qScale$ και του αριθμού Bits στο παραγόμενο bitstream για την Εικόνα 2.

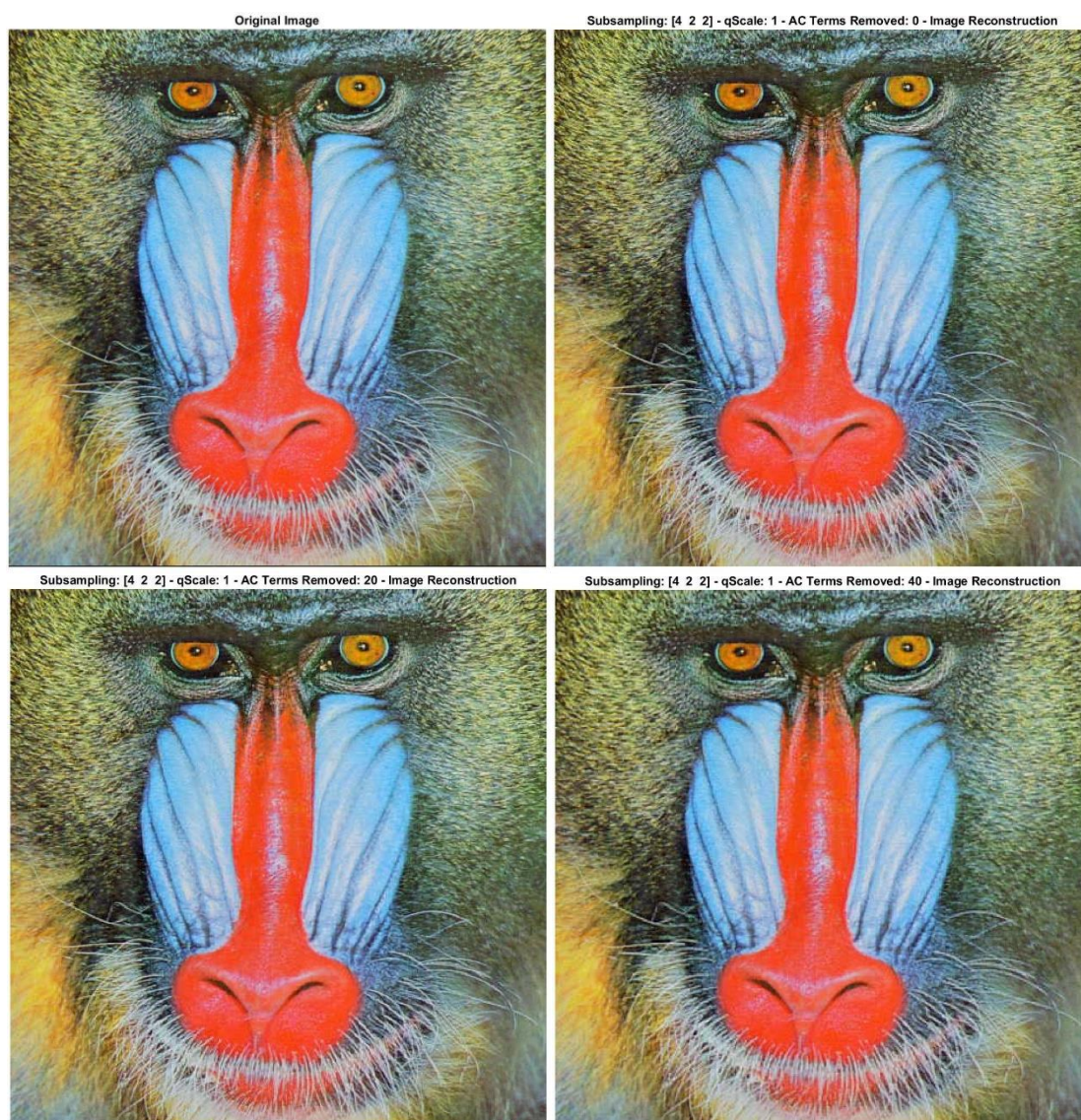
qScale	Compression Ratio	Mean Square Error	Number of bits in bitsream
0.1	7.829585	5.370911	754112
0.3	14.223284	14.780592	415121
0.6	21.266560	26.995220	277637
1.0	28.720056	42.847335	205584
2.0	42.569766	83.598225	138699
5.0	64.837740	221.221880	91064
10.0	78.596222	574.877455	75123

Πίνακας 3.2: Αποτελέσματα - Εικόνα 2.

Το δεύτερο τμήμα περιλαμβάνει την κωδικοποίηση των δύο εικόνων με διαφορετικούς πίνακες κβαντισμού qTable. Συγκεκριμένα, οι πίνακες κβαντισμού τροποποιούνται κατάλληλα ώστε να μηδενιστούν οι 20, 40, 50, 60 και 63 πλέον υψίσυχνι όροι των dctBlock. Η τιμή των παραμέτρων subimg για την υποδειγματοληψία χρώματος, λαμβάνεται όπως στο Demo1 για κάθε εικόνα αντίστοιχα ενώ η παράμετρος qScale λαμβάνεται ίση με 1.

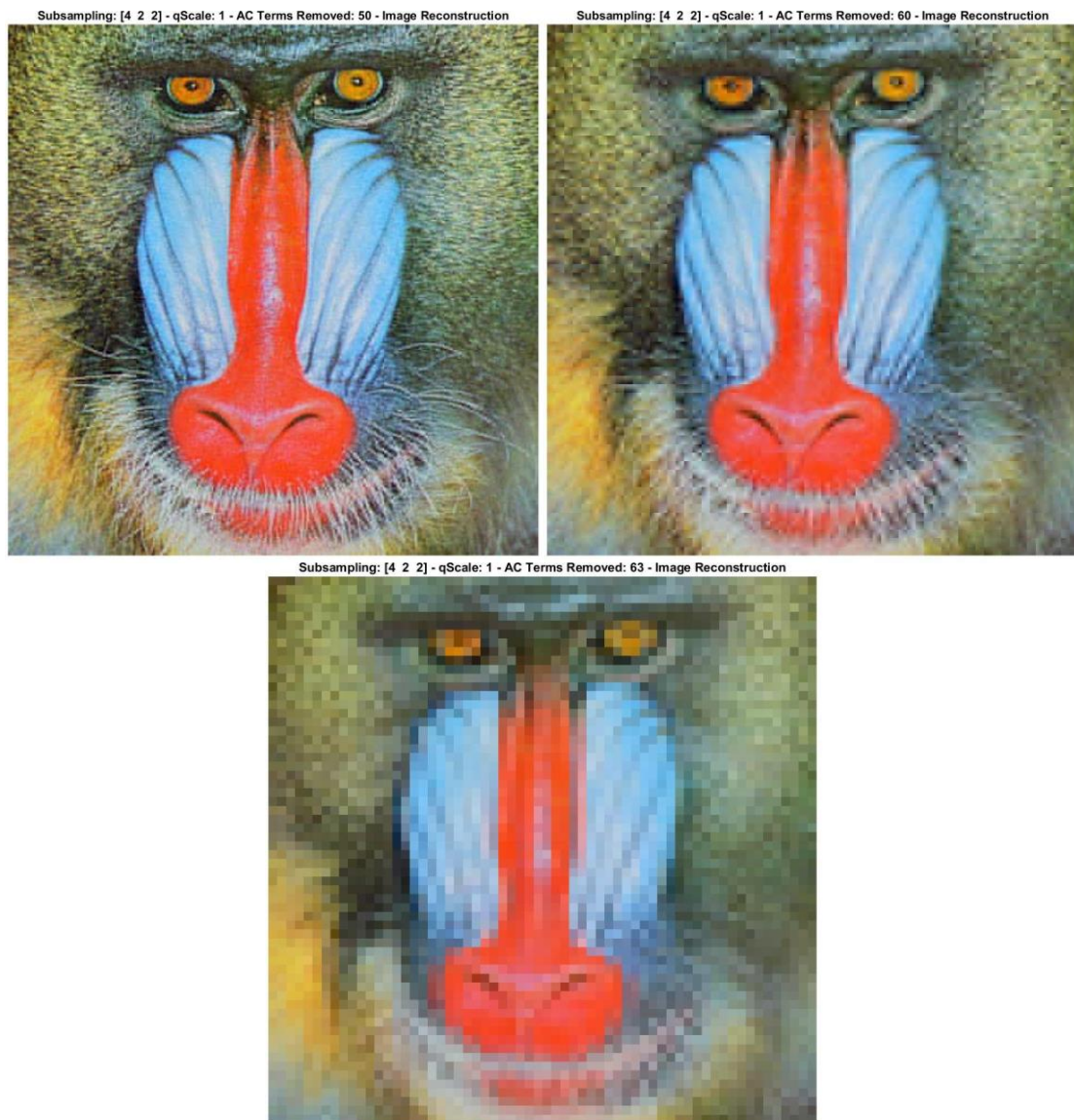
Επίσης για λόγους μεγαλύτερης πληρότητας εξάγονται και πάλι οι τιμές του Μέσου Τετραγωνικού Σφάλματος κάθε εικόνας, το πλήθος των bits του bit-Stream και ο λόγος συμπίεσης σε κάθε περίπτωση.

Τα παραπάνω αποτελέσματα φαίνονται στη συνέχεια:



Σχήμα 3.13: Αποτελέσματα - Εικόνα 1 για 0, 20, 40 μηδενισμένους υψίσυχνους όρους.

Κωδικοποίηση – Αποκωδικοποίηση JPEG



Σχήμα 3.14: Αποτελέσματα - Εικόνα 1 για 50, 60, 63 μηδενισμένους υψίσυχνους όρους.

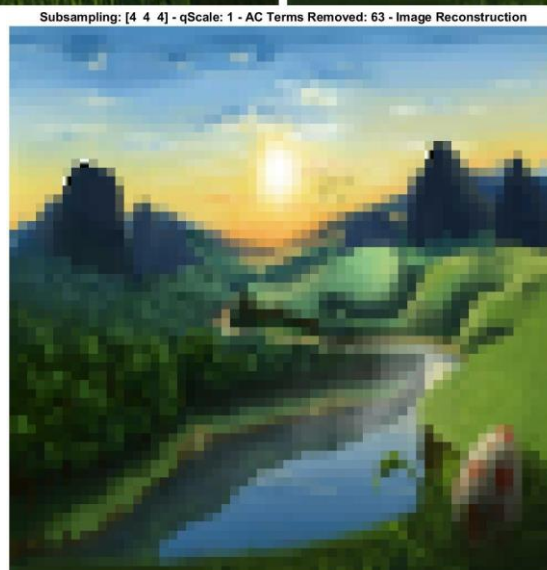
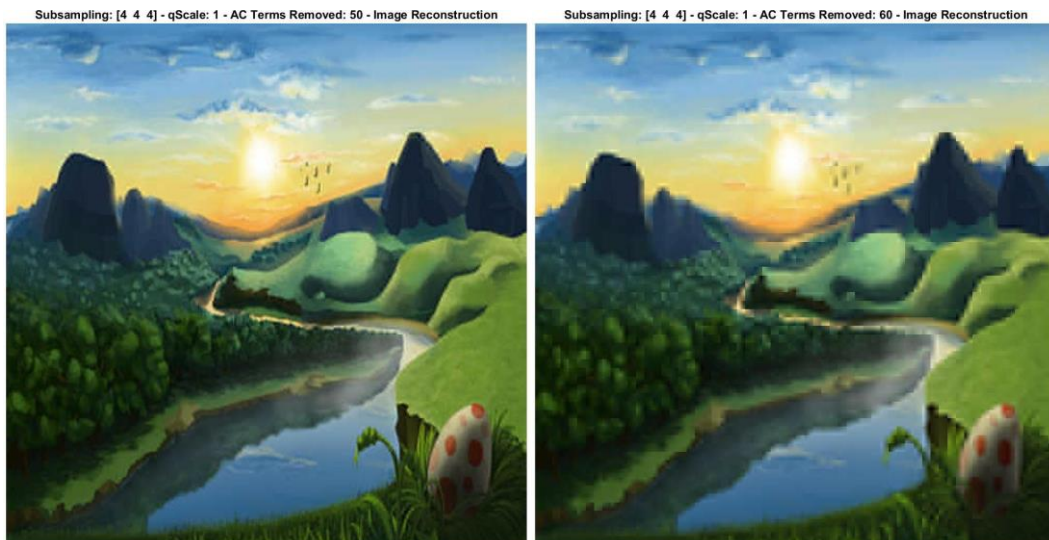
Για τη δεύτερη εικόνα αντίστοιχα:



Κωδικοποίηση – Αποκωδικοποίηση JPEG



Σχήμα 3.15: Αποτελέσματα - Εικόνα 2 για 0, 20, 40 μηδενισμένους υψίσυχνους όρους.



Σχήμα 3.16: Αποτελέσματα - Εικόνα 2 για 50, 60, 63 μηδενισμένους υψίσυχνους όρους.

Παρατηρείται, αντίστοιχα, ότι καθώς αυξάνεται το πλήθος των μηδενισμένων υψίσυχων όρων στον πίνακα κβαντισμού, αυξάνονται και οι απώλειες-σφάλμα. Ωστόσο, το μέγεθος του bit-Stream μειώνεται (δηλαδή ο αυξάνεται ο λόγος συμπίεσης). Επομένως, επαληθεύεται και πάλι το γεγονός ότι η μείωση της ποιότητας αποφέρει μεγαλύτερη εξοικονόμηση χώρου (μικρότερο bit-Stream).

Για το μηδενισμό των υψίσυχων όρων αντικαθίστανται οι τελευταίοι (σε προς τα πίσω Zig-Zag σειρά) AC συντελεστές των πινάκων κβαντισμού με την τιμή 1000, όπως φαίνεται στο παρακάτω σχήμα.

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

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

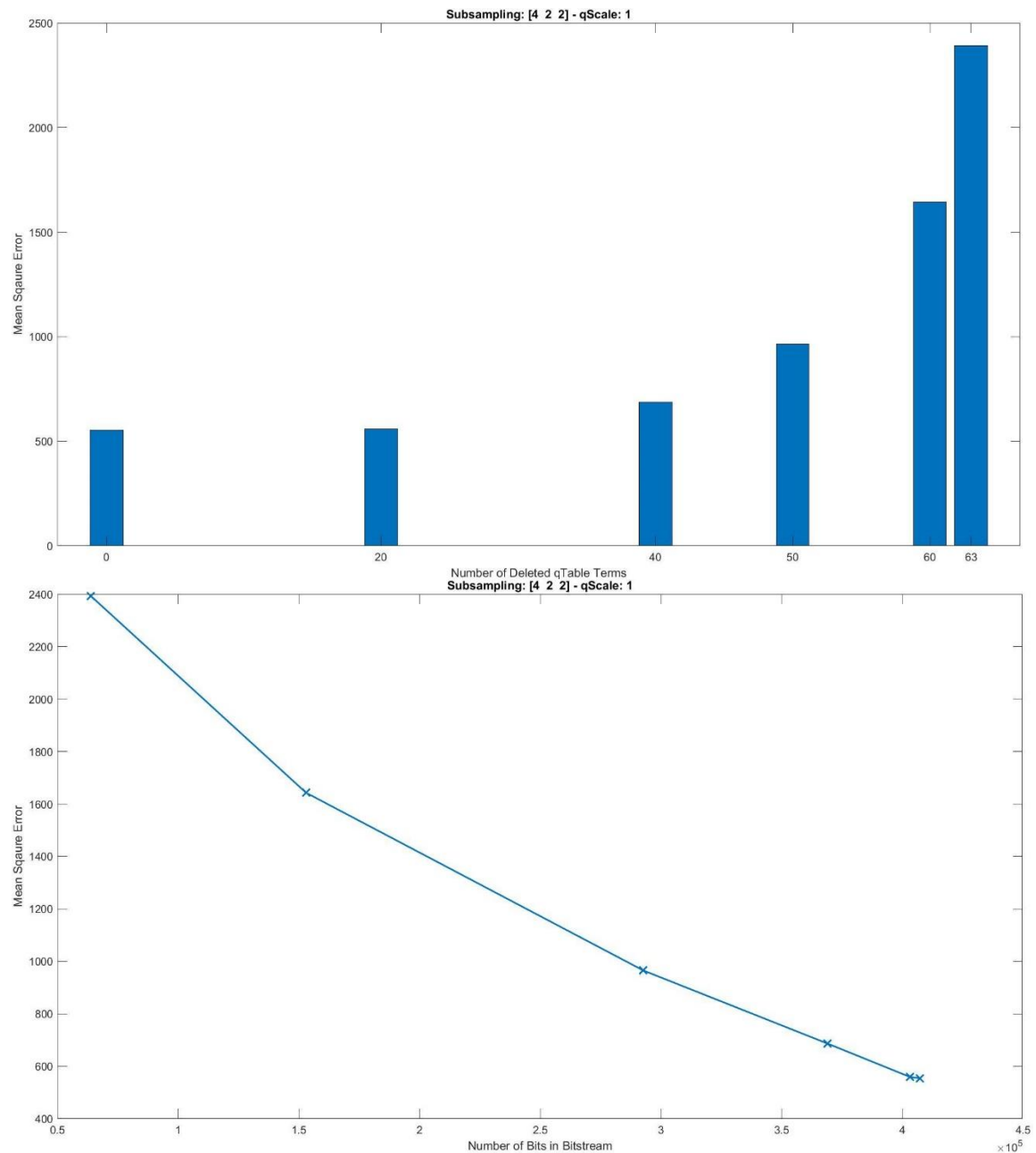
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	1000
49	64	78	87	103	121	1000	1000
72	92	95	98	112	1000	1000	1000

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Σχήμα 3.17: Παράδειγμα μηδενισμού των τελευταίων έξι υψίσυχων όρων αντικαθιστώντας με zig-zag σειρά τους τελευταίους έξι συντελεστές του πίνακα κβαντισμού με την τιμή 1000.

Στη συνέχεια ακολουθούν οι γραφικές παραστάσεις του μέσου τετραγωνικού σφάλματος συναρτήσεως των του πλήθους των τελευταίων μηδενισμένων υψίσυχων συντελεστών AC καθώς και του αριθμού bits για κάθε περίπτωση qScale κάθε εικόνας, οι οποίες επαληθεύουν το παραπάνω συμπέρασμα.

Κωδικοποίηση – Αποκωδικοποίηση JPEG

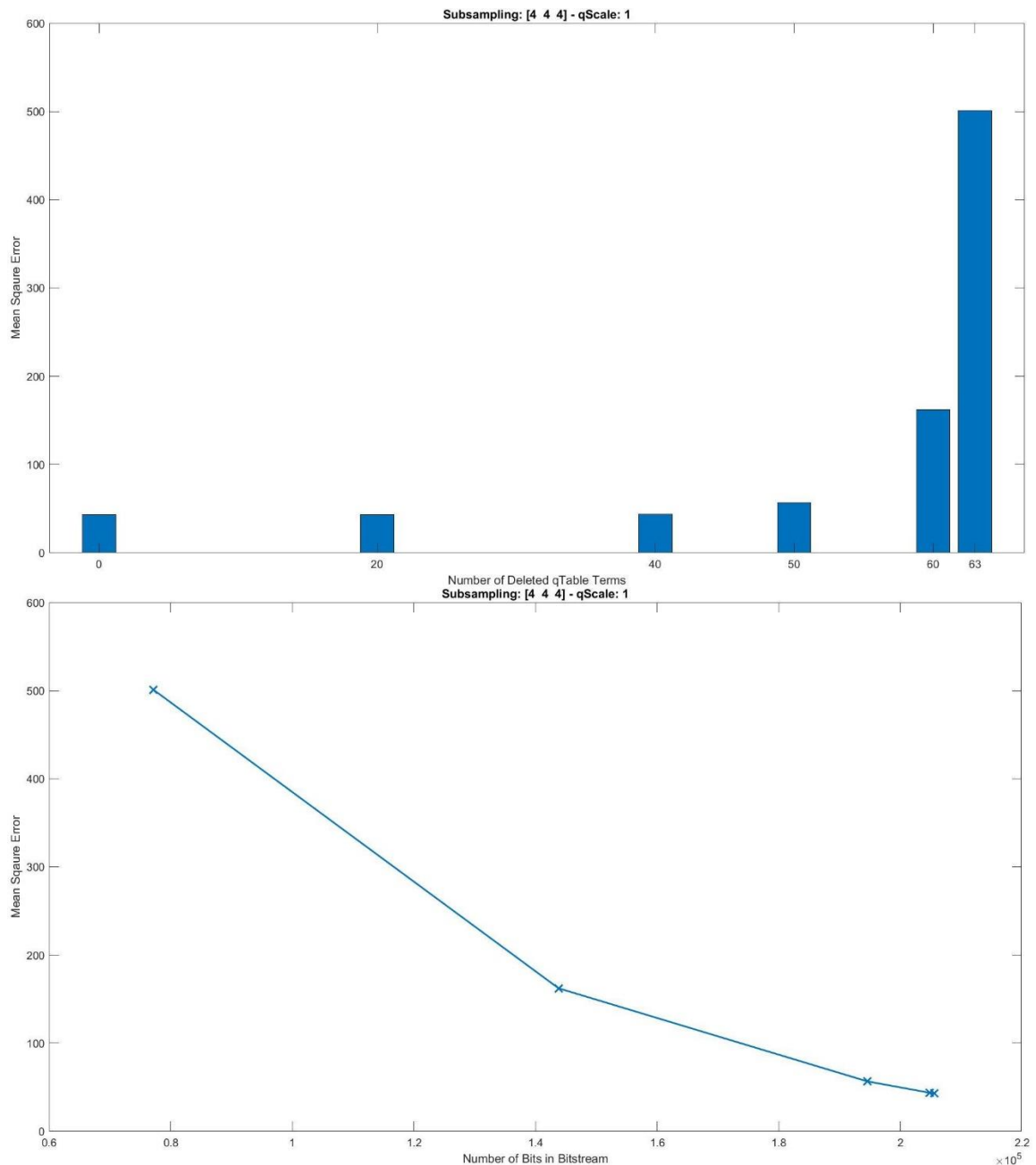


Σχήμα 3.18: Μέσο Τετραγωνικό σφάλμα συναρτήσει των τελευταίων μηδενισμένων υψίσυχνων συντελεστών AC και του αριθμού Bits στο παραγόμενο bitstream για την Εικόνα 1.

AC Terms Removed	Compression Ratio	Mean Square Error	Number of bits in bitsream
0	14.497967	554.340604	407256
20	14.643991	559.184492	403195
40	16.002949	686.863732	368956
50	20.166348	965.068626	292784
60	38.576626	1642.984062	153056
63	92.358460	2393.077450	63929

Πίνακας 3.3: Αποτελέσματα 2 - Εικόνα 1.

Κωδικοποίηση – Αποκωδικοποίηση JPEG



Σχήμα 3.19: Μέσο Τετραγωνικό σφάλμα συναρτήσει των τελευταίων μηδενισμένων υψίσυχων συντελεστών AC και του αριθμού Bits στο παραγόμενο bitstream για την Εικόνα 2.

AC Terms Removed	Compression Ratio	Mean Square Error	Number of bits in bitsream
0	28.720056	42.847335	205584
20	28.720056	42.847335	205584
40	28.829015	43.767877	204807
50	30.344873	56.662022	194576
60	41.058266	162.298143	143805
63	76.515357	501.179244	77166

Πίνακας 3.4: Αποτελέσματα 2 - Εικόνα 2.

3.3 Συνάρτηση Επίδειξης - Demo 2

Έπειτα, ακολουθεί ακόμα μία συνάρτηση επίδειξης που αφορά την αποτελεσματικότητα του αλγορίθμου-κώδικα. Στη συνάρτηση αυτή υπολογίζονται για τις δύο εικόνες που χρησιμοποιήθηκαν στο Demo1 τα εξής μεγέθη:

- α) Η εντροπία της εικόνας RGB.
- β) Η εντροπία των κβαντισμένων συντελεστών DCT.
- γ) Η εντροπία χρησιμοποιώντας τους πίνακες Run Length Encoding.

Για τον πρώτο υπολογισμό, γίνεται η καταμέτρηση της εντροπίας για κάθε χρωματική συνιστώσα ξεχωριστά η οποία πολλαπλασιάζεται στη συνέχεια με το πλήθος δειγμάτων της εικόνας και τέλος προστίθενται οι τιμές που υπολογίστηκαν.

Στη συνέχεια, για το δεύτερο υπολογισμό, γίνεται συνένωση των κβαντισμένων μπλοκ DCT κάθε συνιστώσας σε ένα πίνακα, υπολογίζεται η εντροπία για κάθε μία συνιστώσα ξεχωριστά αφού πολλαπλασιαστεί με το πλήθος των δειγμάτων της και στο τέλος προστίθενται οι τιμές που υπολογίστηκαν, ώστε ο υπολογισμός να είναι συγκρίσιμος με τον προηγούμενο.

Τέλος, για τον τρίτο υπολογισμό συνενώνονται οι πίνακες RLE κάθε συνιστώσας, πολλαπλασιάζονται με τον αριθμό των στοιχείων αυτών και με όμοιο τρόπο προστίθενται οι τιμές που προκύπτουν για να εξαχθεί η συνολική εντροπία. Σημειώνεται ότι κατά τον υπολογισμό της εντροπίας λαμβάνεται κάθε ζεύγος (Run , Coefficient) ως κωδικό σύμβολο και όχι κάθε στοιχείο του πίνακα ξεχωριστά.

Τα αποτελέσματα της διαδικασίας αυτής φαίνονται στη συνέχεια με τη μορφή πίνακα:

Εντροπία	RGB Image	Quantized DCT Blocks	Run Length Encoded Blocks
Εικόνα 1	5.7290e+06	7.2585e+05	5.5903e+05
Εικόνα 2	5.7226e+06	1.4877e+05	5.0423e+04

Παρατηρούμε ότι η εντροπία της εικόνας RGB είναι η μέγιστη σε σχέση με τις άλλες δύο. Στη συνέχεια ακολουθεί η εντροπία των κβαντισμένων

συντελεστών DCT (DCT Blocks) και τέλος η εντροπία των μηκών διαδρομής RLE. Μπορεί, λοιπόν, να εξαχθεί το συμπέρασμα ότι καθώς προχωρούν τα διάφορα στάδια της κωδικοποίησης προς τη δημιουργία του τελικού bit-Stream, η εντροπία μειώνεται προκειμένου να επιτευχθεί συμπίεση.

4. Επιπρόσθετα Αρχεία MATLAB

Εκτός από τα ζητούμενα αρχεία MATLAB, παραδίδονται επιπλέον τα εξής αρχεία:

Tables.m: Το αρχείο αυτό περιέχει τους πρότυπους πίνακες ACC, ACL, DCC, DCL που χρησιμοποιούνται κατά την κωδικοποίηση Huffman, τους πίνακες κβαντισμού qTableL, qTableC και τον πίνακα zig-zag που χρησιμοποιείται κατά την κωδικοποίηση.

distortedqTables.m: Το αρχείο αυτό περιέχει τους αλλοιωμένους πίνακες κβαντισμού, που χρησιμοποιούνται ώστε να μηδενιστούν οι τελευταίοι υψίσυχντοι AC όροι όπως ζητείται.

results.m: Το αρχείο αυτό περιέχει τον κώδικα που αφορά την εκτίμηση του μέσου τετραγωνικού σφάλματος σε σύγκριση με τα διάφορα μεγέθη bitstream για διαφορετικές τιμές της παραμέτρου qScale όπως ζητείται.

resultsB.m: Το αρχείο αυτό περιέχει τον κώδικα που αφορά την εκτίμηση του μέσου τετραγωνικού σφάλματος σε σύγκριση με τα διάφορα μεγέθη bitstream όταν έχουν μηδενιστεί οι τελευταίοι υψίσυχντοι AC όροι όπως ζητείται.

*** Σημείωση για τον κώδικα:** Κατά τη διαδικασία κωδικοποίησης (JPEGencode) ενσωματώνονται στο πρώτο struct του ζητούμενου cell, οι πίνακες qTableL και qTableC αφού πολλαπλασιαστούν πρώτα με την τιμή qScale. Επιπλέον η μεταβλητή blockType ορίζεται ως ολική μεταβλητή (global variable) προκειμένου να χρησιμοποιηθεί από τη συνάρτηση κωδικοποίησης Huffman (huffEnc) χωρίς να της δοθεί ως όρισμα.