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](#_Toc30509213)

[2. Τμήμα Α – JPEG Library 4](#_Toc30509214)

[2.1 Προεπεξεργασία 4](#_Toc30509215)

[2.2 Μετασχηματισμός DCT 7](#_Toc30509216)

[2.3 Κβαντισμός 8](#_Toc30509217)

[2.4 Zig Zag Scanning και Run Length Encoding 9](#_Toc30509218)

[2.5 Huffman Encoding 11](#_Toc30509219)

[2.6 Συνάρτηση Επίδειξης - Demo 1 13](#_Toc30509220)

[3. Τμήμα Β – JPEG Integration 14](#_Toc30509221)

[3.1 JPEG Encoder/Decoder 14](#_Toc30509222)

[3.2 Καταγραφή Αποτελεσμάτων 16](#_Toc30509223)

[3.3 Συνάρτηση Επίδειξης - Demo 2 17](#_Toc30509224)

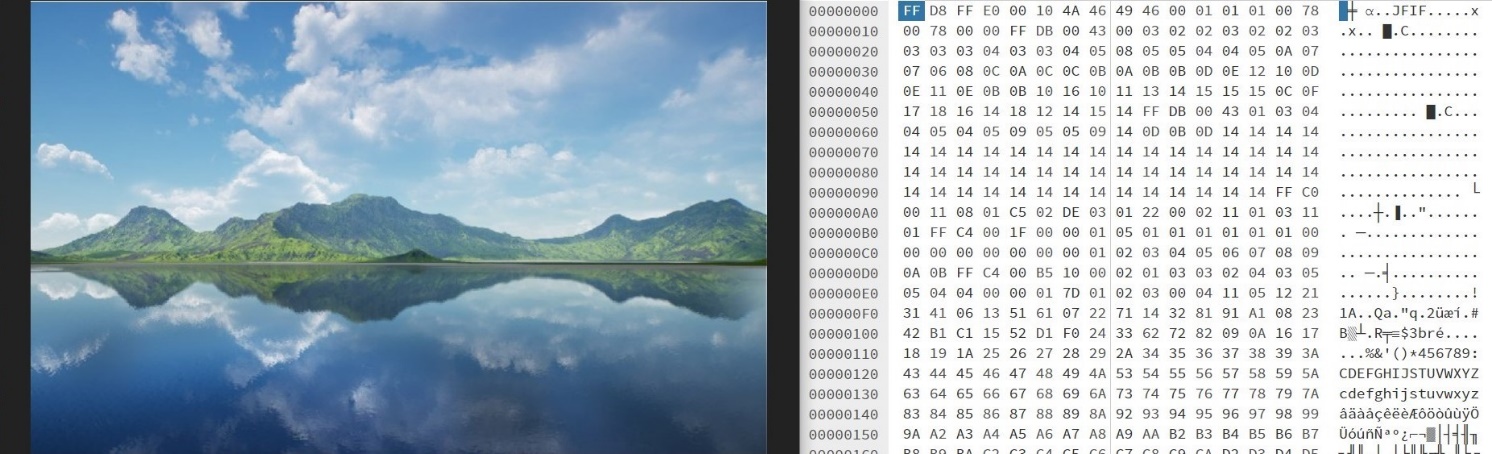
# 1. Εισαγωγή

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.



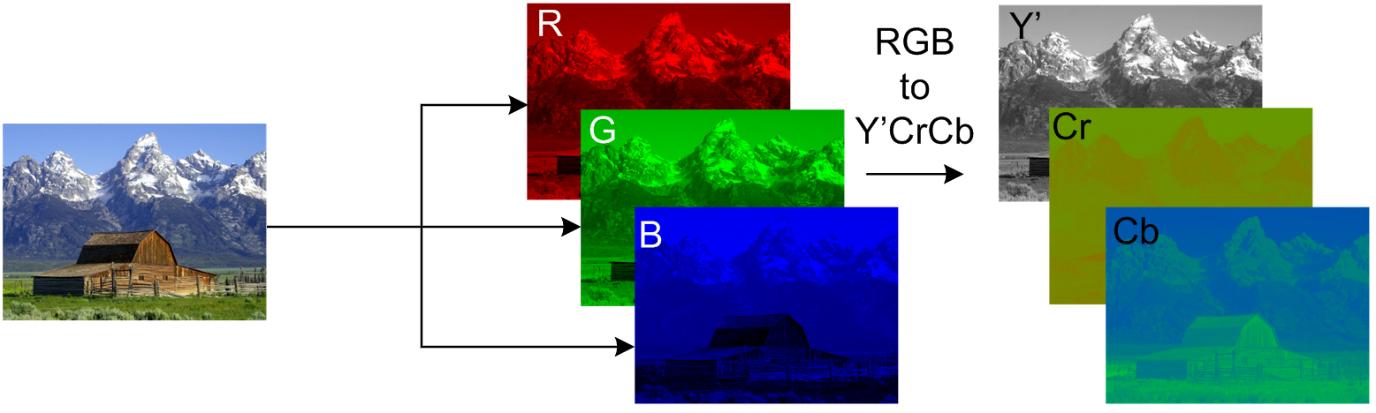
*Σχήμα 1.2: Πραγματικό παράδειγμα bit-Stream Εικόνας. Κάθε ζεύγος αριθμών στο δεκαεξαδικό σύστημα αναπαριστά ένα Byte.*

# 2. Τμήμα Α – 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.*

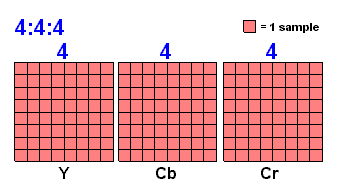
Η εικόνα RGB έχει τη μορφή τρισδιάστατου πίνακα μεγέθους , όπου R, C η οριζόντια και κατακόρυφη διάσταση ενώ η τρίτη διάσταση λαμβάνει τις τιμές 1 (Red), 2 (Green), 3 (Blue) καθώς περιέχει τις τρεις χρωματικές συνιστώσες της εικόνας.

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

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

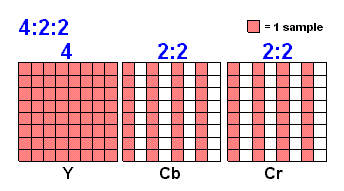
Στη συνέχεια ακολουθείται ένα στάδιο υποδειγματοληψίας χρώματος προκειμένου να μειωθεί ο όγκος της εικόνας. Συγκεκριμένα, η διαδικασία αυτή διακρίνεται σε 3 κατηγορίες.

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

**

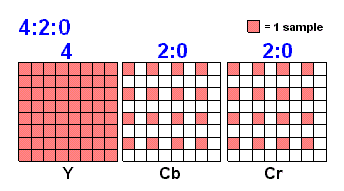
*Σχήμα 2.2: Υποδειγματοληψία Χρώματος 4:4:4.*

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 , , , 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 , , , για τις συνιστώσες Y, Cb, Cr αντίστοιχα.

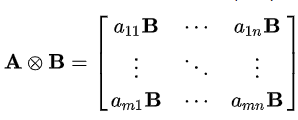
**

*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 . 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 o **Α** είναι ένας πίνακας και ο **Β** είναι ένας πίνακας, τότε το γινόμενο Kronecker **A** ⊗ **B** είναι ο μπλοκ πίνακας :



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 ενώ για την κατηγορία 4:2:0 εφαρμόζεται το γινόμενο Kronecker ανάμεσα στους χρωματικούς πίνακες Cb, Cr με τον πίνακα .

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

***\* 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 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.

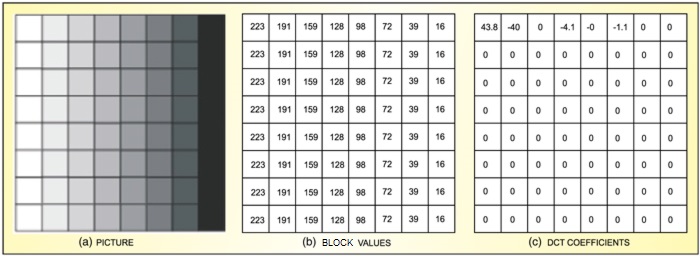
Ο μετασχηματισμός DCT2 φαίνεται παρακάτω:

Όπου και , ο συντελεστής που βρίσκεται στη θέση του μετασχηματισμένου μπλοκ, το στοιχείο που βρίσκεται στη θέση του κβαντισμένου μπλοκ και συντελεστές ίσοι με 1 αν ή αντίστοιχα και ίσοι με , διαφορετικά.

Ο αντίστροφός μετασχηματισμός είναι:

Όπου και , ο συντελεστής που βρίσκεται στη θέση του αρχικού μπλοκ, το στοιχείο που βρίσκεται στη θέση του μετασχηματισμένου μπλοκ και συντελεστές ίσοι με αν ή αντίστοιχα και ίσοι με , διαφορετικά.

Στη συγκεκριμένη περίπτωση των μπλοκ , έχουμε .



*Σχήμα 2.5: Μετασχηματισμός DCT σε Μπλοκ Ασπρόμαυρης Εικόνας.*

## 2.3 Κβαντισμός

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

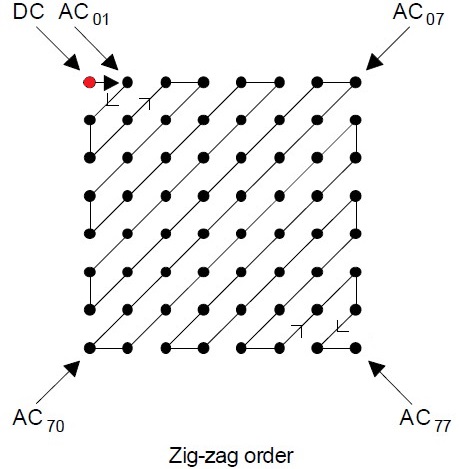
όπου οι δείκτες υποδεικνύουν τη θέση του εκάστοτε δείγματος στο συγκεκριμένο μπλοκ, qScale σταθερά ρύθμισης του κβαντιστή και qTable ένας πίνακας μεγέθους που ονομάζεται πίνακας κβαντισμού και διαφέρει για τα μπλοκ της χρωματικής συνιστώσας Y από τις υπόλοιπες (Cb, Cr). Οι πίνακες qTable που χρησιμοποιούνται στην παρούσα υλοποίηση είναι οι προτεινόμενοι πίνακες όπως ορίζονται από το πρότυπο.

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

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.

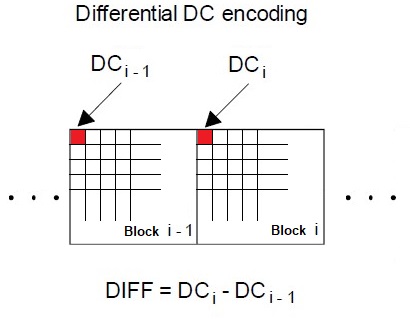


*Σχήμα 2.6: Μετατροπή Μπλοκ σε Διάνυσμα 64 Στοιχείων. Με ACij συμβολίζεται ο συντελεστής AC στη θέση του μπλοκ (i , j).*

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 , 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.*

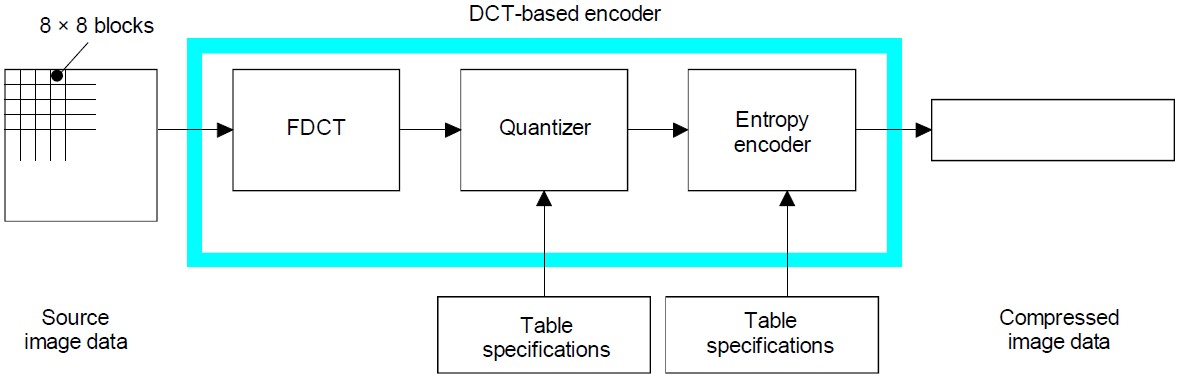
Για τις υπόλοιπες γραμμές η διαδικασία συμπλήρωσης είναι η εξής:

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 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 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.

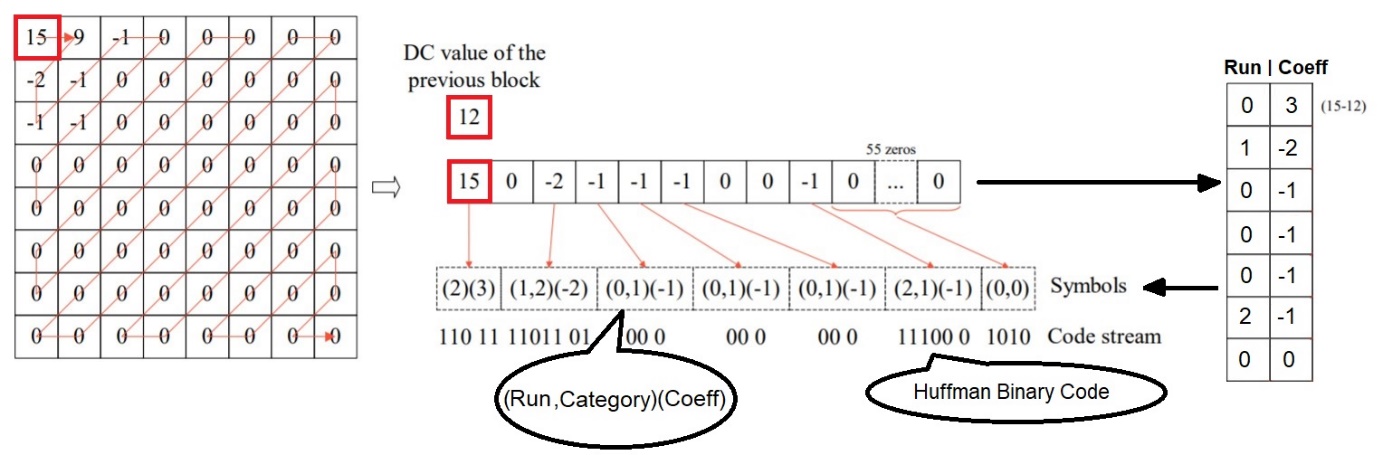


*Σχήμα 2.8: Διαδικασία Κωδικοποίησης Μπλοκ Εικόνας.*

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. Εξαίρεση 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.

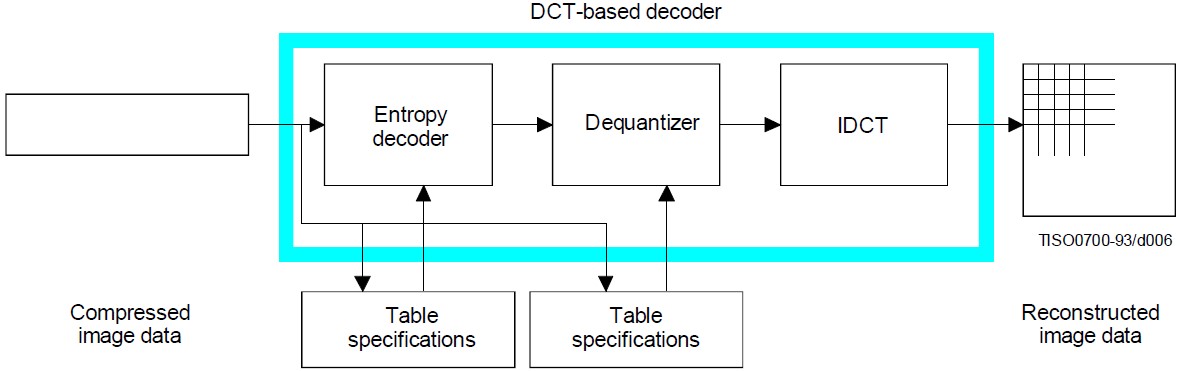
Για τους συντελεστές AC αρχικά εντοπίζεται με τον ίδιο τρόπο η κατηγορία στην οποία ανήκουν. Με βάση αυτή καθώς και το μέγεθος των μηδενικών ακολουθιών που βρίσκεται στην πρώτη στήλη του πίνακα rle αναζητείται στον αντίστοιχο πρότυπο πίνακα η δυαδική λέξη που κωδικοποιεί το ζεύγος αυτό. Η λέξη αυτή προστίθεται 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 επαναφοράς 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 Β – JPEG Integration

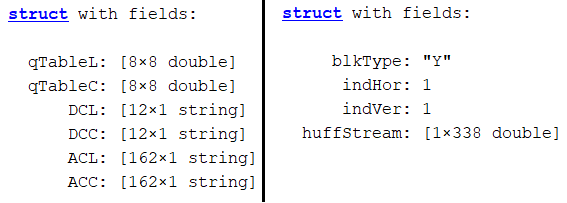
Το επόμενο τμήμα έχει ως στόχο την ενσωμάτωση συναρτήσεων που υλοποιήθηκαν στο προηγούμενο τμήμα σε μια γενικότερη συνάρτηση και την εξαγωγή ποσοτικών και ποιοτικών συμπερασμάτων για τη συμπίεση που επιτεύχθηκε.

## 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 και 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.

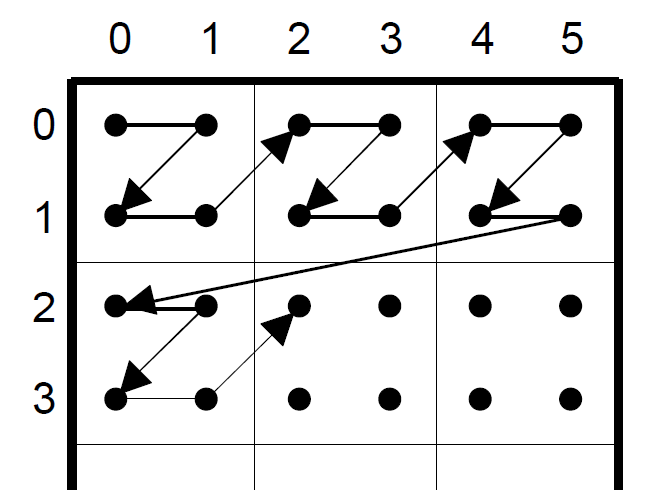
Τα υπόλοιπα cells του πίνακα περιέχουν για κάθε μπλοκ τα χαρακτηριστικά blkType με τον τύπο του εκάστοτε μπλοκ (Y, Cb, Cr), τη θέση του στην εικόνα μέσω των indHor, indVer δεικτών και το bit-Stream που δημιουργήθηκε από τη συνάρτηση huffmanEncode της προηγούμενης βαθμίδας.



*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, JPEGenc.

## 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:

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:

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

Τα παραπάνω αποτελέσματα παρουσιάζονται με τη μορφή πίνακα στη συνέχεια:

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 υψίσυχνοι όροι των dctBlock. Η τιμή της παραμέτρου qScale λαμβάνεται ίση με 1, ενώ για την υποδειγματοληψία χρώματος χρησιμοποιούνται οι ίδιες παράμετροι όπως στο Demo1.

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:

Παρατηρείται, αντίστοιχα, ότι καθώς αυξάνεται το πλήθος των μηδενισμένων υψίσυχνων 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 υψίσυχνων όρων αντικαθίστανται οι τελευταίοι (σε Zig-Zag σειρά) AC συντελεστές των κβαντισμένων DCT Block με την τιμή 1000.

Τα παραπάνω αποτελέσματα παρουσιάζονται με τη μορφή πίνακα στη συνέχεια:

## 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:

1. Η εντροπία της εικόνας RGB.
2. Η εντροπία των κβαντισμένων συντελεστών DCT.
3. Η εντροπία χρησιμοποιώντας τους πίνακες Run Length Encoding.

Για τον πρώτο υπολογισμό, γίνεται η καταμέτρηση της εντροπίας για κάθε χρωματική συνιστώσα ξεχωριστά και στο τέλος προστίθενται οι τιμές που υπολογίστηκαν.

Στη συνέχεια, για το δεύτερο υπολογισμό γίνεται συνένωση των κβαντισμένων μπλοκ DCT κάθε συνιστώσας, υπολογίζεται η εντροπία για κάθε μία ξεχωριστά 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 και με όμοιο τρόπο προστίθενται οι τιμές που προκύπτουν για να εξαχθεί η συνολική εντροπία.

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

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 Block) 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.