

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης  
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Ηλεκτρονικής και Υπολογιστών



Εργασία στο Μάθημα Συστήματα Πολυμέσων και Εικονική  
Πραγματικότητα

**Κωδικοποίηση/Αποκωδικοποίηση ακίνητης εικόνας  
κατά το πρότυπο JPEG**

Όνομα: Στέφανος Γανωτάκης

AEM: 7664

Email: sganotak@auth.gr

ΙΑΝΟΥΑΡΙΟΣ 2020

## Περιεχόμενα

1	Εισαγωγή .....	3
1.1	Αντικείμενο και Δομή Εργασίας.....	3
2	1 <sup>ο</sup> Παραδοτέο: JPEG Library (Demo 1) .....	4
2.1	Α' Μέρος.....	4
2.2	Β' Μέρος.....	6
3	2 <sup>ο</sup> Παραδοτέο: JPEG Intergration (Demo 2) .....	9
3.1	Πείραμα 1: Mean Square Error .....	10
3.2	Πείραμα 2:Τροποποίηση πινάκων κβαντισμού.....	13
3.3	Υπολογισμός Εντροπίας .....	18
4	3 <sup>ο</sup> Παραδοτέο (JPEG Syntax) .....	19
4.1	Κωδικοποίηση .....	19
4.2	Αποκωδικοποίηση.....	19
4.3	Πείραμα 3:Λόγος Συμπίεσης.....	20

# 1 Εισαγωγή

Το κείμενο αυτό αποτελεί την αναφορά για την εργασία στα Συστήματα Πολυμέσων του ακαδημαϊκού έτους 2019-2020. Παρακάτω εμπεριέχονται συνοπτικές επεξηγήσεις πάνω στην λειτουργία του παραδοτέου κώδικα και σχόλια πάνω στα αποτελέσματα. Η εργασία υλοποιήθηκε πλήρως, και σε κάθε μία από τις παρακάτω ενότητες, αναλύεται ξεχωριστά η κάθε βαθμίδα του συστήματος και τα αποτελέσματά της. Τέλος είναι σημαντικό να αναφερθεί ότι εκτός από τις εικόνες που δίνονται για πειραματισμό η λειτουργία του κώδικα επαληθεύτηκε και σε ένα πλήθος από δοκιμαστικές εικόνες που διαθέτει η βιβλιοθήκη του Matlab και πληρούν τις προϋποθέσεις της εκφώνησης.

## 1.1 Αντικείμενο και Δομή Εργασίας

Στην εργασία αυτή αναπτύχθηκε ένας κωδικοποιητής/αποκωδικοποιητής ακίνητης εικόνας κατά το πρότυπο **JPEG**. Πιο συγκεκριμένα υλοποιήθηκε η εκδοχή **baseline sequential DCT-based**. Ο παραδοτέος κώδικας έχει δομηθεί σε τρεις ενότητες, οι οποίες επεξηγούνται συνοπτικά παρακάτω:

- Το πρώτο παραδοτέο υλοποιεί μια απλή (από)κωδικοποίηση δυο εικόνων μέχρι και το στάδιο του κβαντισμού.
- Το δεύτερο παραδοτέο προσομοιώνει την πλήρη διαδικασία κωδικοποίησης μιας ακίνητης εικόνας και την αντίστροφη διαδικασία της. Επιπλέον πραγματοποιούνται διάφορα πειράματα και εξάγονται μετρικές σχετικές με την ποιότητα των ανακατασκευασμένων εικόνων
- Το τρίτο παραδοτέο αφορά τη ενσωμάτωση του JPEG syntax στον αλγόριθμο κωδικοποίησης που αναπτύχθηκε στο δεύτερο παραδοτέο, έτσι ώστε τα κωδικοποιημένα αρχεία που παράγονται να είναι συμβατά με το πρότυπο JFIF και κατ' επέκτασιν να μπορούν να διαβαστούν από οποιοδήποτε πρόγραμμα επεξεργασίας εικόνας.

Πρέπει να σημειωθεί ότι οι συναρτήσεις που υλοποιήθηκαν είναι αρκετά περισσότερες σε πλήθος από αυτές που περιγράφονται στην εκφώνηση της εργασίας, διευκολύνοντας έτσι την ανάπτυξη και την επαλήθευση του κώδικα. Ο κώδικας και η λειτουργία κάθε προγράμματος θα αναλυθεί με μεγαλύτερη λεπτομέρεια στις ενότητες που ακολουθούν αλλά και εντός των προγραμμάτων υπό την μορφή σχολίων.

## 2 1<sup>ο</sup> Παραδοτέο: JPEG Library (Demo 1)

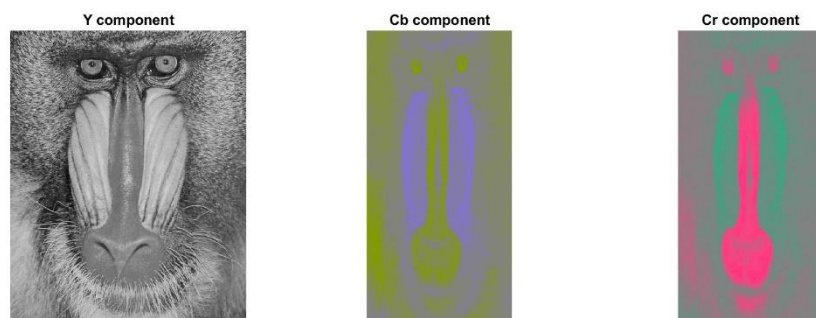
### 2.1 Α' Μέρος

Το πρώτο μέρος του 1<sup>ου</sup> παραδοτέου υλοποιεί την μετατροπή δυο εικόνων RGB σε YCbCr και έπειτα πίσω σε RGB. Οι συναρτήσεις που χρησιμοποιήθηκαν για αυτό το κομμάτι της εργασίας είναι οι:

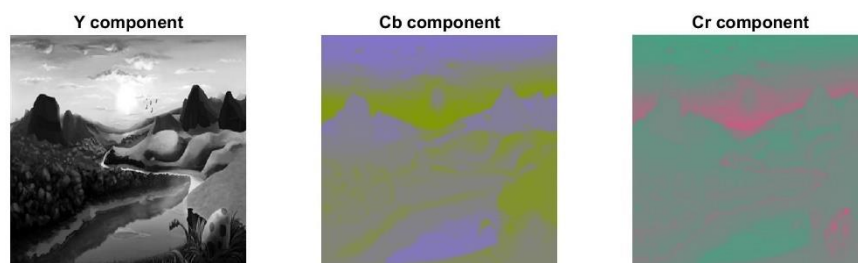
- **convert2ycbcr**
- **convert2rgb**

Οι οποίες υλοποιήθηκαν όπως ακριβώς ορίζονται στην εκφώνηση.

. Για την εικόνα 1 χρησιμοποιήθηκε υποδειγματοληψία 4:2:2, δηλαδή ο αριθμός των δειγμάτων για τις κατακόρυφες συνιστώσες των στοιχείων χρωματικότητας μειώθηκε στο μισό του αρχικού ενώ για την εικόνα 2 4:4:4, δεν υπήρξε δηλαδή καμία υποδειγματοληψία. Καλώντας την **convert2ycbcr** και για τις δυο εικόνες, προκύπτουν τα ακόλουθα αποτελέσματα, που είναι οι συνιστώσες φωτεινότητας και χρωματικότητας για την κάθε εικόνα

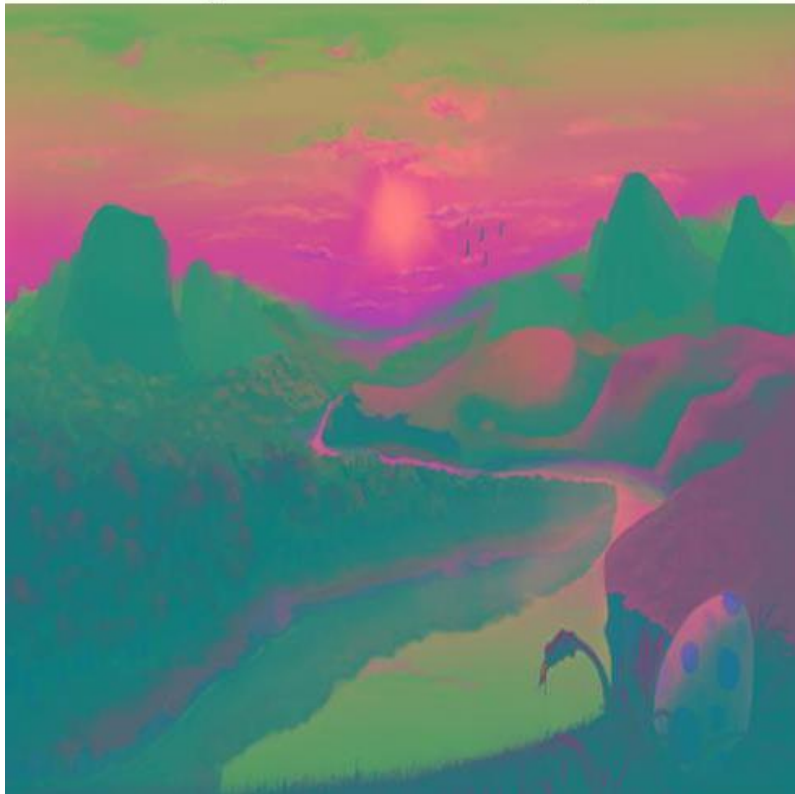


Εικόνα 1: YCbCr components πρώτης εικόνας



Εικόνα 2: YCbCr components 2ης εικόνας

## image 2:YCbCr Color Space



Εικόνα 3: YCbCr αναπαράσταση 2ης εικόνας

Για την αντίστροφη διαδικασία καλείται η **convert2rgb** ώστε οι 2 εικόνες να επανέλθουν στο spatial domain. Για την υπερδειγματοληψία των μορφών 4:2:2 και 4:2:0 αναπτύχθηκε η συνάρτηση **upsample\_nn**, η οποία επαναφέρει τις συνιστώσες χρωματικότητας στο αρχικό τους μέγεθος μέσω της μεθόδου του κοντινότερου γείτονα. Με το πέρας της μετατροπής, παράγονται οι ακόλουθες εικόνες, οι οποίες είναι και ίδιες με τις αρχικές. Συνεπώς επαληθεύεται η ορθή λειτουργία των δυο συναρτήσεων που αναπτύχθηκαν

Image 1

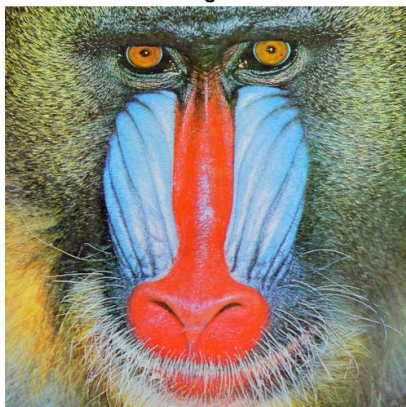
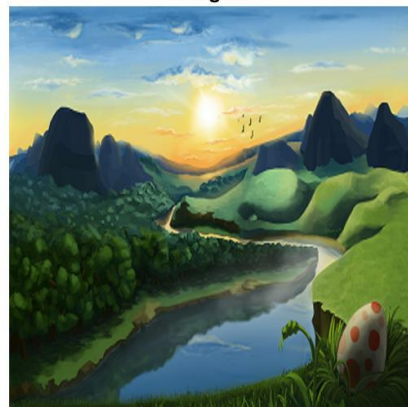


Image 2



## 2.2 Β' Μέρος

Στο δεύτερο μέρος του 1<sup>ου</sup> παραδοτέου επαναλαμβάνεται η ίδια διαδικασία με αυτή του πρώτου μέρους με την προσθήκη ωστόσο δυο παραπάνω βημάτων: του DCT μετασχηματισμού, του κβαντισμού και των αντίστροφων διαδικασιών τους. Οι συναρτήσεις που υλοποιήθηκαν για αυτό το βήμα είναι οι:

- **blockDCT,**
- **quantizeJPEG**

Για τη δημιουργία των κβαντισμένων DCT συντελεστών και οι αντίστροφες τους:

- **iblockDCT**
- **dequantizeJPEG**

Και οι τέσσερις συναρτήσεις υλοποιήθηκαν όπως ακριβώς ορίζονται στην εκφώνηση. Για την πρώτη εικόνα χρησιμοποιήθηκε κλίμακα κβαντισμού 0.6 ενώ για την δεύτερη 5. Στις επόμενες εικόνες παρουσιάζονται τα συνολικά αποτελέσματα της κατάτμησης σε blocks, του μετασχηματισμού DCT και του κβαντισμού. Τα δύο τελευταία απεικονίζονται μόνο για την εικόνα 2, καθώς η απουσία υποδειγματοληψίας επιτρέπει τη σύνθεση των επιμέρους συνιστωσών φωτεινότητας και χρωματικότητας σε μια ενιαία εικόνα καθιστώντας τα αποτελέσματα πιο εμφανή.

Image 1 Block Segmentation

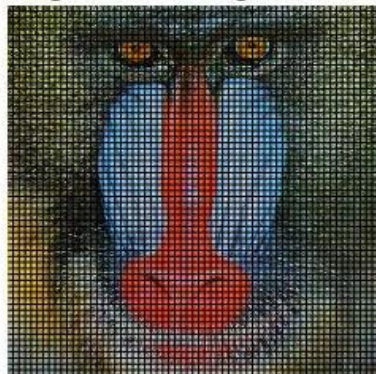


Image 2 Block Segmentation



Εικόνα 4: Κατάτμηση εικόνων σε 8x8 blocks



## Image 2:Block DCT Transformation



Εικόνα 5: Μετασχηματισμός DCT εικόνας 2

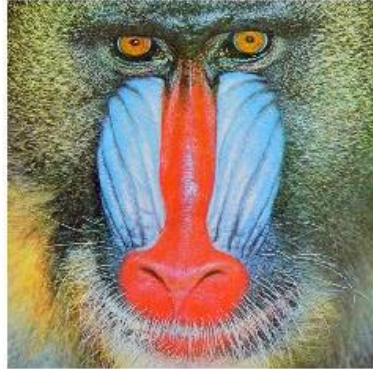
## Image 2:Quantized DCT Blocks



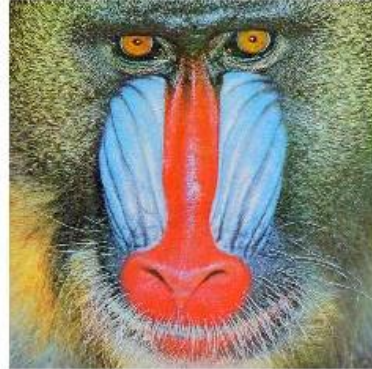
Εικόνα 6:Κβαντισμος DCT συντελεστών εικόνας 2

Τα αποτελέσματα που προκύπτουν μετά την ανακατασκευή των εικόνων είναι τα παρακάτω

**Original Image 1**



**Reconstructed Image 1**



*Εικόνα 7: Σύγκριση αρχικής και ανακατασκευασμένης εικόνας 1*

**Original Image 2**



**Reconstructed Image 2**



*Εικόνα 8: Σύγκριση αρχικής και ανακατασκευασμένης εικόνας 2*

Όπως γίνεται αντιληπτό, η ποιότητα της ανακατασκευασμένης εικόνας 2 υπέφερε εξαιτίας της μεγάλης τιμής που δόθηκε στην κλίμακα κβαντισμού, που προκάλεσε και τον μηδενισμό των περισσότερων υψηλών συχνοτήτων των DCT blocks . Όσον αφορά την εικόνα 1 φαίνεται ότι δεν υπήρχε αισθητή μείωση της ποιότητας.



### 3 2<sup>ο</sup> Παραδοτέο: JPEG Intergration (Demo 2)

Για την εκπόνηση του δεύτερου παραδοτέου υλοποιήθηκαν οι συναρτήσεις **JPEGencode** και **JPEGdecode**. Η λειτουργία των δυο συναρτήσεων περιγράφεται αναλυτικά παρακάτω

- **JPEGencode**: Η **JPEGencode** λαμβάνει ως όρισμα την αρχική εικόνα, τον τύπο υποδειγματοληψίας και την κλίμακα κβαντισμού. Η αρχική RGB εικόνα μέσω της **rgb2ycbcr** μετατρέπεται σε YCbCr και υποδειγματοληπτείται ανάλογα με τον τύπο δειγματοληψίας που επιλέχθηκε, έπειτα γίνεται κατάρτηση της εικόνας σε 8x8 blocks και πραγματοποιείται μετασχηματισμός DCT σε κάθε block, μέσω της συνάρτησης **blockDCT**. Στη συνέχεια κάθε **DCTblock** κβαντίζεται μέσω της **quantizeJPEG**, δημιουργώντας έτσι τον κβαντισμένο πίνακα των DCT συντελεστών για τα στοιχεία της φωτεινότητας και χρωματικότητας της εικόνας. Κάθε πίνακας υπάγεται σε zig-zag scanning και στην συνέχεια σε Run Length Encoding μέσω της συνάρτησης **runLength**, δίνοντας ένα πίνακα Nx2 για κάθε block ενός στοιχείου, όπου N ο αριθμός των συμβόλων που εντόπισε ο αλγόριθμος. Η πρώτη γραμμή αυτού του πίνακα απεικονίζει τα σύμβολα που προέκυψαν από την κωδικοποίηση, ενώ η δεύτερη το μήκος διαδρομής του εκάστοτε συμβόλου. Οι πίνακες που προέκυψαν από την RLE κωδικοποιούνται περαιτέρω κατά Huffman, με την βοήθεια της συνάρτησης **huffEnc** και των έτοιμων πινάκων που διατίθενται στο πρότυπο, δίνοντας μας μια ακολουθία από bits για κάθε πίνακα RLE. Κάθε δυαδική ακολουθία που παράγει η κωδικοποίηση Huffman αποθηκεύεται σε έναν cell πίνακα από structs όπως ζητήθηκε από την εκφώνηση.
- **JPEGDecode**: Η **JPEGdecode** λαμβάνει ως όρισμα των cell πίνακα που δημιουργεί η **JPEGencode** και ξεκινά την διαδικασία της αποκωδικοποίησης. Αρχικά γίνεται scan του κάθε στοιχείου του cell πίνακα και δημιουργούνται οι πίνακες με τις κωδικοποιημένες ακολουθίες Huffman για τα στοιχεία Y,Cb και Cr. Κάθε κελί των πινάκων αυτών υπάγεται σε αποκωδικοποίηση Huffman μέσω της **huffDec** δίνοντας καινούργιους πίνακες, τα κελιά των οποίων περιέχουν τις κωδικοποιημένες RLE ακολουθίες των κβαντισμένων DCT συντελεστών των στοιχείων. Έπειτα χρησιμοποιείται η **irunLength** για κάθε block η οποία μέσω inverse zig zag scanning και Run Length Decoding κατασκευάζει τα αρχικά 8x8 block των κβαντισμένων DCT συντελεστών. Για κάθε block πραγματοποιείται αποκβαντισμός και αντίστροφος μετασχηματισμός DCT μέσω των **dequantizeJPEG** και **iblockDCT**. Στη συνέχεια αφού αναπαραχθούν οι αρχικοί πίνακες των συνιστωσών Y,Cb και Cr γίνεται επιστροφή στο spatial domain με τη χρήση της **convert2rgb**. Έτσι ολοκληρώνεται η ανακατασκευή της κωδικοποιημένης εικόνας.

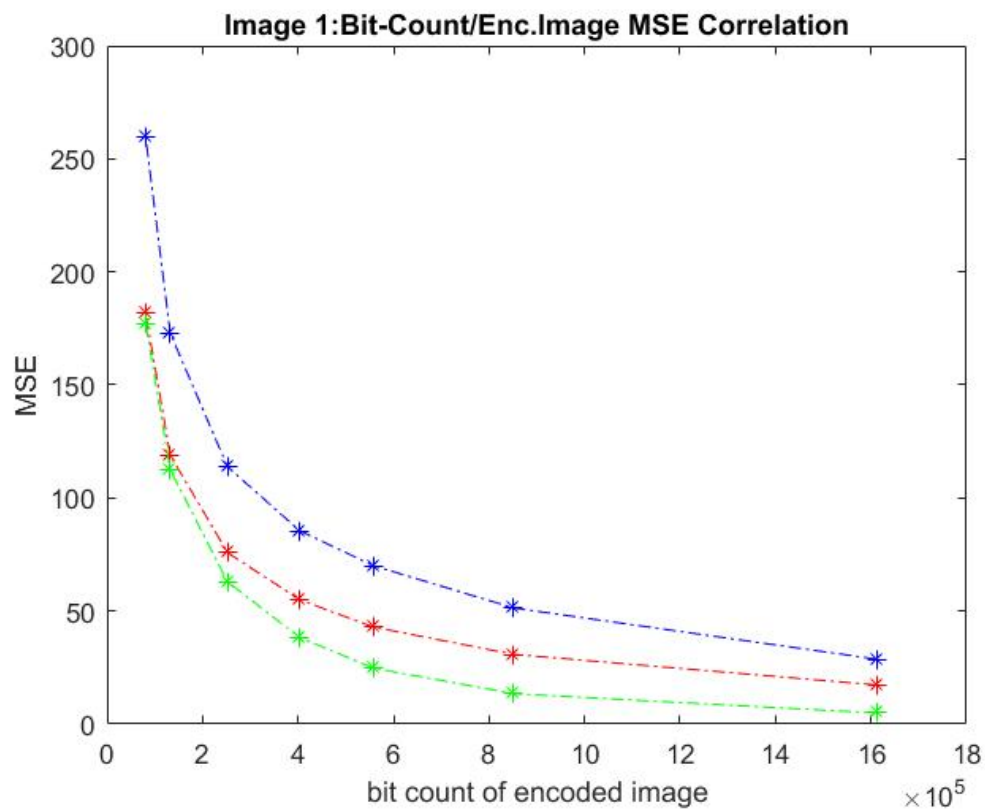
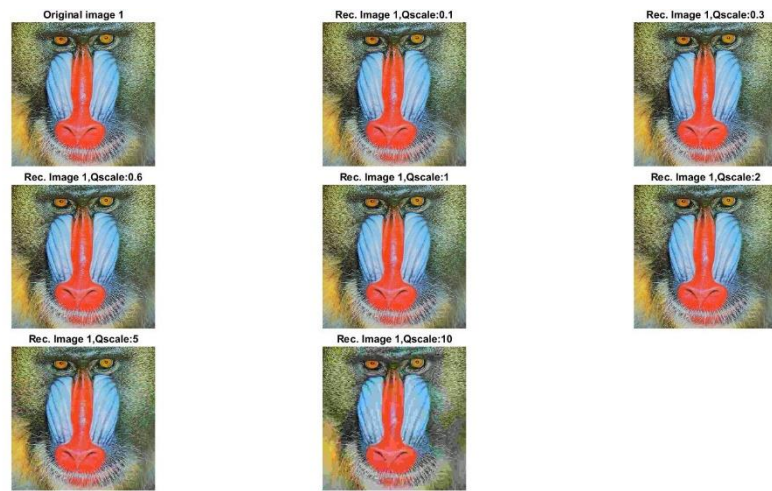
Εκτός από τις συναρτήσεις που ζητήθηκαν από την εκφώνηση, υλοποιήθηκαν και οι ακόλουθες συναρτήσεις:

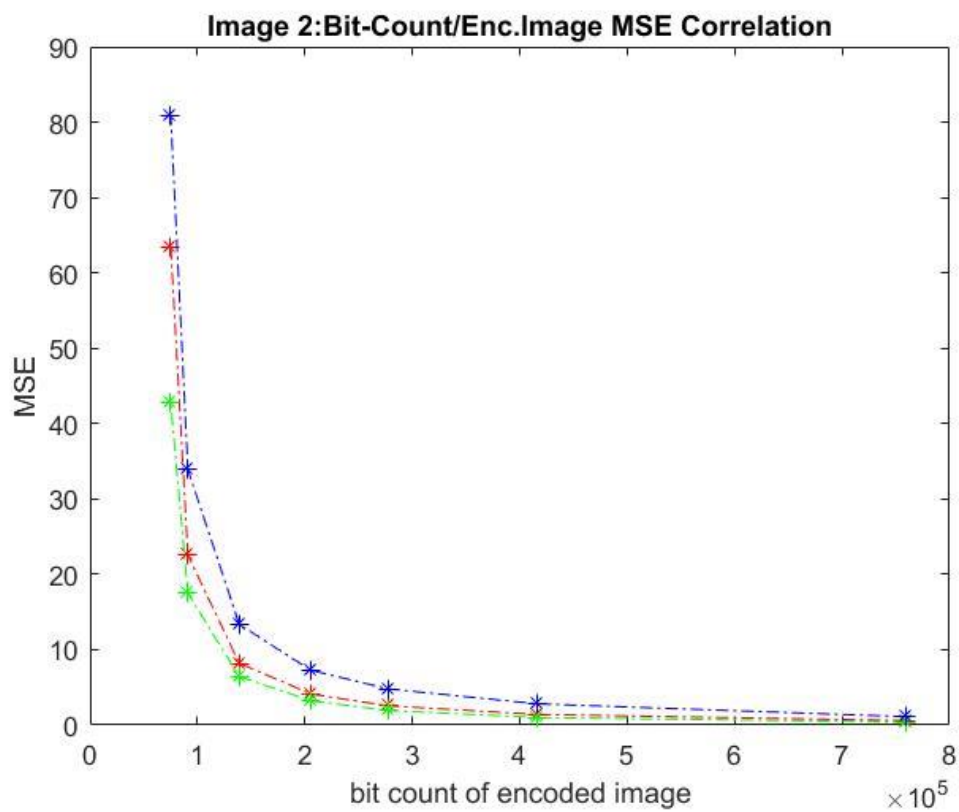
Συνάρτηση	Σύνοψη
<b>out=zigzag(in)</b>	Πραγματοποιεί zig zag scanning του πίνακα που δίνεται ως όρισμα
<b>out=invzigzag(in,num_rows,num_cols)</b>	Αντίστροφη διαδικασία της παραπάνω.
<b>mse = MSEcalc( ogimg,recimg )</b>	Υπολογισμός μέσου τετραγωνικού σφάλματος μεταξύ αρχικής και ανακατασκευασμένης εικόνας
<b>entropycell = entropycalc( img,subimg,qScale )</b>	Υπολογισμός εντροπίας

- Για τις συναρτήσεις **huffEnc** και **huffDec** προστέθηκαν δύο επιπλέον ορίσματα, τα AC και DC που αντιπροσωπεύουν τους πίνακες Huffman που χρησιμοποιούνται για την κωδικοποίηση των AC και DC συντελεστών για τα στοιχεία φωτεινότητας και χρωματικότητας. Οι πίνακες αυτοί είναι αποθηκευμένοι στο αρχείο **HuffmanTables.mat** και φορτώνονται ως global μεταβλητές στο πρόγραμμα πριν την έναρξη της κωδικοποίησης.
- **Παρατήρηση:** Είναι σημαντικό να αναφερθεί ότι εφόσον δεν ζητήθηκε από την εκφώνηση, η σειρά της (από)κωδικοποίησης που πραγματοποιούν οι **JPEGencode/JPEGdecode** σε αυτό το στάδιο της εργασίας δεν επηρεάζεται από τον τύπο της υποδειγματοληψίας, αφού στη παρούσα υλοποίηση πρώτα κωδικοποιούνται όλα τα Y blocks, στη συνέχεια τα Cb blocks και τέλος τα Cr blocks. Ωστόσο για την εκπόνηση του 3<sup>ου</sup> παραδοτέου έχει αναπτυχθεί ανάλογη συνάρτηση που προσομοιώνει την ορθή σειρά (από)κωδικοποίησης σύμφωνα με το πρότυπο.

### 3.1 Πείραμα 1: Mean Square Error

Για το πρώτο πείραμα κατασκευάστηκε ένα διάνυσμα με διάφορες τιμές κλίμακας κβαντισμού, πιο συγκεκριμένα τις (0.1,0.3,0.6,1,2,5,10). Έπειτα για κάθε εικόνα που δίνεται για πειραματισμό παράχθηκε το κωδικοποιημένο Huffman bitstream μέσω της συνάρτησης **JPEGencode** για όλες τις τιμές του διανύσματος με τις κλίμακες κβαντισμού. Ο συνολικός αριθμός των bits κάθε κωδικοποιημένης ακολουθίας μετρήθηκε και αποθηκεύτηκε σε δομή δεδομένων. Η ανακατασκευή των εικόνων πραγματοποιήθηκε με την χρήση της **JPEGdecode**. Τέλος με την συνάρτηση **MSEcalc** υπολογίστηκε το μέσο τετραγωνικό σφάλμα (MSE) για κάθε χρωματική συνιστώσα μεταξύ της αρχικής και της ανακατασκευασμένης εικόνας. Τα αποτελέσματα του παραπάνω πειράματος παρουσιάζονται στις εικόνες και τις γραφικές παραστάσεις που ακολουθούν





Τέλος, οι μέσες τιμές του τετραγωνικού σφάλματος και για τις 3 χρωματικές συνιστώσες για κάθε τιμή τις κλίμακας κβαντισμού συνοψίζονται στον ακόλουθο πίνακα.



Κλίμακα Κβαντισμού	Εικόνα 1	Εικόνα 2
	MSE	MSE
<b>0.1</b>	16.9898	0.7008
<b>0.3</b>	31.8547	1.7450
<b>0.6</b>	45.9320	3.0933
<b>1</b>	59.5531	4.8638
<b>2</b>	84.1690	9.3387
<b>5</b>	134.7342	24.6285
<b>10</b>	206.3000	62.3710

Όπως διαπιστώνεται έπειτα από την μελέτη των παραπάνω πινάκων και διαγραμμάτων, υπάρχει μεγάλη διαφορά στην τάξη μεγέθους του MSE μεταξύ των δυο εικόνων. Μάλιστα, προκαλεί εντύπωση το γεγονός ότι το MSE για κλίμακα κβαντισμού 1 στην εικόνα 1 είναι περίπου ίσο με το MSE για κλίμακα κβαντισμού 10 στην εικόνα 2. Η διαφορά αυτή στην ποιότητα της ανακατασκευής των δυο εικόνων θα μπορούσε να αποδοθεί στο γεγονός ότι η εικόνα 1 εμφανίζει μεγαλύτερη «παλέτα χρωμάτων» και κατ' επέκταση μικρότερη συσχέτιση μεταξύ των pixels. Δηλαδή, κατά τον μετασχηματισμό DCT, η πληροφορία που αποθηκεύεται στους AC όρους της εικόνας 1 είναι αρκετά σημαντικότερη από την αντίστοιχη της εικόνας 2. Έτσι με τον μηδενισμό όλο και περισσότερων υψηλόσυχων συντελεστών που επιφέρει η αύξηση της κλίμακας κβαντισμού η πληροφορία που χάνεται στην περίπτωση της εικόνας 1 είναι αρκετά μεγαλύτερη από αυτήν της εικόνας 2.

Εξετάζοντας τις γραφικές παραστάσεις του MSE για το κάθε χρώμα ξεχωριστά, φαίνεται ότι στην εικόνα 2, το πράσινο που είναι και το κυρίαρχο χρώμα παρουσιάζει αρκετά μικρότερο σφάλμα σε σχέση με τα άλλα 2. Απεναντίας στην περίπτωση της εικόνας 1, το μπλε φαίνεται να έχει το μεγαλύτερο σφάλμα.

### 3.2 Πείραμα 2: Τροποποίηση πινάκων κβαντισμού

Για αυτό το πείραμα δημιουργήθηκε το script **quantmodification.m**, στο οποίο ορίζεται ένα διάνυσμα με τον ζητούμενο αριθμό των AC συντελεστών που είναι επιθυμητό να μηδενιστούν. Στην συνέχεια αφού πραγματοποιηθεί ο μετασχηματισμός DCT για κάθε στοιχείο της εικόνας, δημιουργούνται οι ανάλογοι πίνακες κβαντισμού φωτεινότητας και χρωματικότητας, όπου τα η AC στοιχεία του εκάστοτε πίνακα είναι ίσα με την τιμή 1024, την μέγιστη δηλαδή τιμή που μπορεί να λάβει μια AC συνιστώσα για 8 bit pixels. Τα υπόλοιπα 63-η είναι ίσα με την μονάδα. Έτσι εξασφαλίζεται ότι για κάθε block, κατά τον κβαντισμό θα υπάρχουν τουλάχιστον η μηδενικοί AC συντελεστές. Με την εκτέλεση του script προκύπτουν τα παρακάτω αποτελέσματα

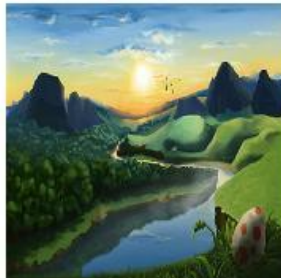
**Original Image 1**



**Reconstructed Image 1:20 Zero AC Coeff**



**Original Image 2**

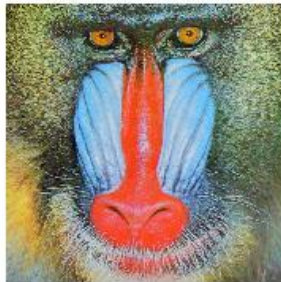


**Reconstructed Image 2:20 Zero AC Coeff**

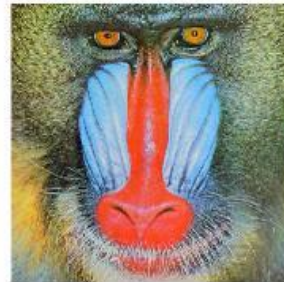


*Εικόνα 9: 20 μηδενικοί AC συντελεστές*

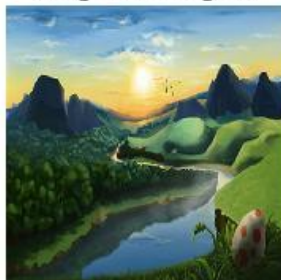
**Original Image 1**



**Reconstructed Image 1:40 Zero AC Coeff**



**Original Image 2**

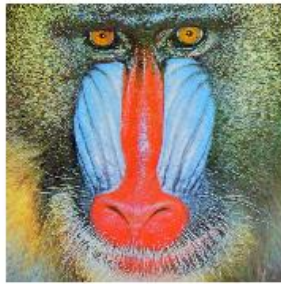


**Reconstructed Image 2:40 Zero AC Coeff**

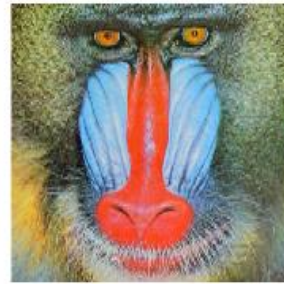


*Εικόνα 10: 40 μηδενικοί AC συντελεστές*

**Original Image 1**



**Reconstructed Image 1:50 Zero AC Coeffs**



**Original Image 2**

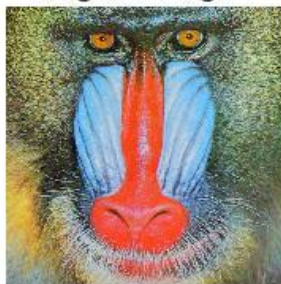


**Reconstructed Image 2:50 Zero AC Coeffs**

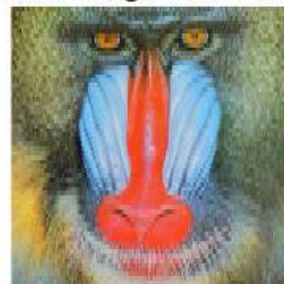


*Εικόνα 11:50 μηδενικοί AC συντελεστές*

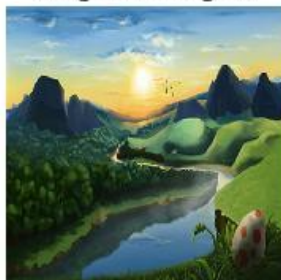
**Original Image 1**



**Reconstructed Image 1:60 Zero AC Coeffs**



**Original Image 2**

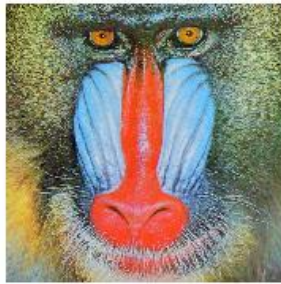


**Reconstructed Image 2:60 Zero AC Coeffs**

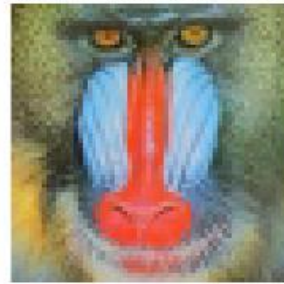


*Εικόνα 12:60 μηδενικοί AC συντελεστές*

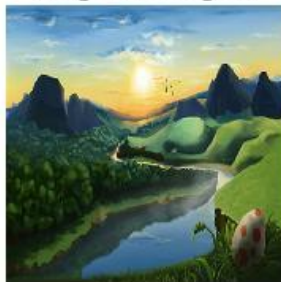
**Original Image 1**



**Reconstructed Image 1:63 Zero AC Coeff**



**Original Image 2**



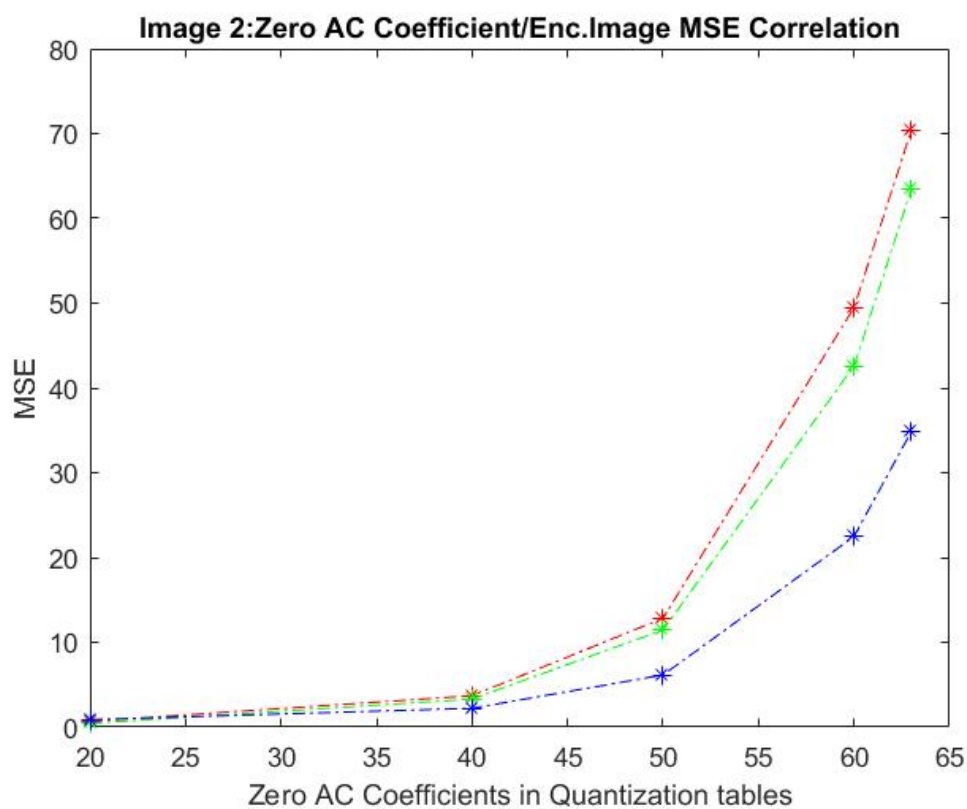
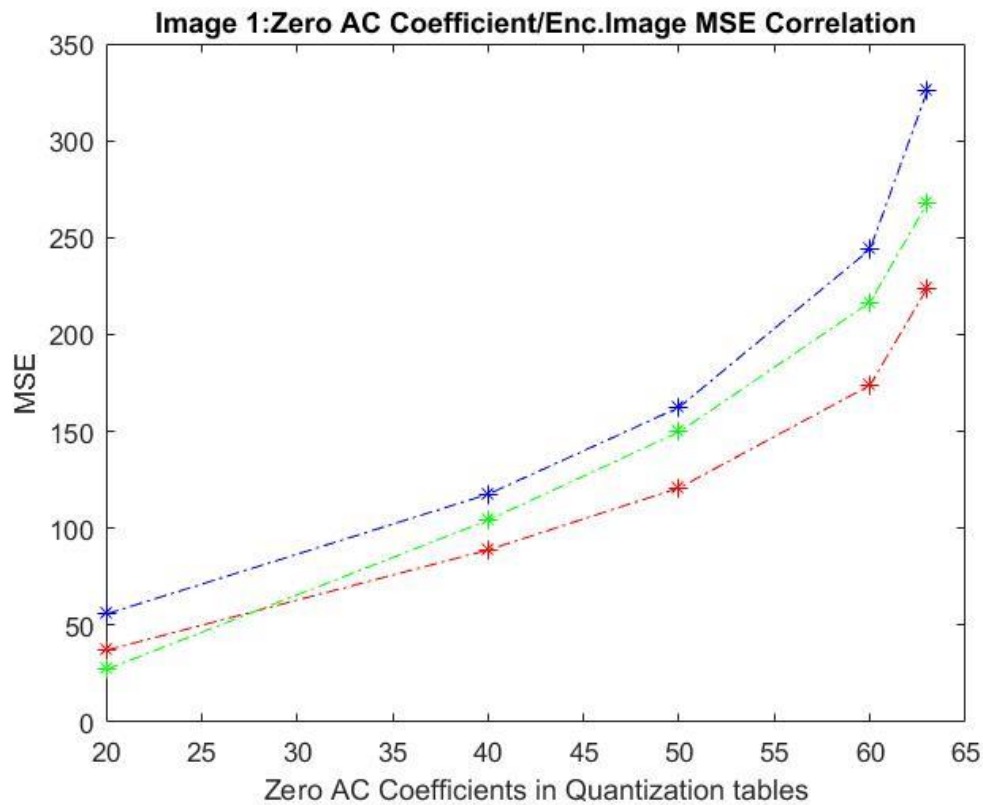
**Reconstructed Image 2:63 Zero AC Coeff**



*Εικόνα 13:63 AC μηδενικοί συντελεστές*

Παρόλο που στην αναφορά ίσως να μην είναι εύκολα διακριτό λόγω του μεγέθους των γραφημάτων, για 50 και άνω μηδενικούς AC συντελεστές υπάρχει σημαντική πτώση στην ποιότητα των ανακατασκευασμένων εικόνων . Αυτό συμβαίνει επειδή κατά τον μετασχηματισμό DCT η μεγαλύτερη ποσότητα της πληροφορίας συλλέγεται στο άνω αριστερά κομμάτι των blocks. Συνεπώς όταν μηδενιστούν αυτοί οι όροι κατά τον κβαντισμό, η ποιότητα ανακατασκευής μειώνεται δραματικά. Το πόρισμα αυτό επιβεβαιώνεται και από τα διαγράμματα για το MSE για κάθε χρωματική συνιστώσα σε συνάρτηση με τον αριθμό των μηδενικών AC συντελεστών που παρατίθενται στην συνέχεια.





Όπως βλέπουμε το MSE αυξάνεται σχεδόν εκθετικά όσο μηδενίζονται περισσότεροι AC συντελεστές. Επιπλέον, οι τιμές του MSE για την εικόνα 1 είναι πολύ μεγαλύτερες από αυτές της εικόνας 2. Η διαφορά αυτή οφείλεται πιθανόν στην ύπαρξη

υποδειγματοληψίας αλλά και στην μεγαλύτερη πυκνότητα πληροφορίας που υπάρχει εξαρχής στην εικόνα 1 και χάνεται κατά τον κβαντισμό

### 3.3 Υπολογισμός Εντροπίας

Με τη χρήση της συνάρτησης **entropycalc.m** υπολογίζεται η εντροπία της κάθε εικόνας στο spatial domain, των κβαντισμένων συντελεστών DCT και τέλος των RLE ακολουθιών. Τα αποτελέσματα για κάθε περίπτωση παρουσιάζονται στους παρακάτω πίνακες.

Spatial Domain	Εικόνα 1	Εικόνα 2
Εντροπία R	7.6878	7.6287
Εντροπία G	7.4655	7.6773
Εντροπία B	7.7440	7.6096
Μέση Εντροπία	7.6324	7.6385

Q.DCT blocks	Εικόνα 1	Εικόνα 2
Εντροπία Y	1.8771	0.2359
Εντροπία Cb	0.5492	0.1307
Εντροπία Cr	0.5187	0.1211
Μέση Εντροπία	0.9817	0.1626

RLE sequences	Εικόνα 1	Εικόνα 2
Εντροπία Y	2.9722	0.8815
Εντροπία Cb	2.0081	0.3353
Εντροπία Cr	1.9178	0.2120
Μέση Εντροπία	2.2828	0.4763

Όπως βλέπουμε, κατά τον κβαντισμό υπάρχει σημαντική μείωση της εντροπίας και για τις δύο εικόνες. Ωστόσο όταν κωδικοποιούνται τα κβαντισμένα DCT blocks κάθε εικόνας κατά RLE υπάρχει αύξηση της εντροπίας. Αυτό είναι αναμενόμενο, δεδομένου ότι ο RLE είναι ένας lossless αλγόριθμος και με την κωδικοποίηση οι διαστάσεις των block μειώνονται χωρίς να υπάρχει μείωση της πληροφορίας, αυξάνοντας έτσι τις πιθανότητες εμφάνισης των συμβόλων.

## 4 3<sup>ο</sup> Παραδοτέο (JPEG Syntax)

Για την εκπόνηση του 3<sup>ου</sup> παραδοτέου, υλοποιήθηκαν οι συναρτήσεις **JPEGencodeStream** και **JPEGdecodeStream**. Επιπρόσθετα αναπτύχθηκε το script **demo3.m** για τον υπολογισμό του λόγου συμπίεσης των εικόνων

### 4.1 Κωδικοποίηση

Για το στάδιο της κωδικοποίησης, τηρήθηκε η ίδια μεθοδολογία με την **JPEGencode**, ωστόσο για την δημιουργία του header file και κατ' επέκτασιν του τελικού κωδικοποιημένου bitstream αναπτύχθηκαν οι επιπλέον συναρτήσεις:

Συνάρτηση	Σύνοψη
<b>header=</b> <b>createHeader(row,col,subimg,qTableL,qTableC)</b>	Δημιουργία header σύμφωνα με το πρότυπο JFIF
<b>bytes =</b> <b>interleave(encstream,subimg,row,col )</b>	Δημιουργία της τελικής ακολουθίας των συμβόλων της εικόνας ανάλογα με τον τύπο της υποδειγματοληψίας
<b>encodeDHTheader()</b>	Κωδικοποίηση header για τους πίνακες huffman

- **Παρατήρηση:** Στην παρούσα εργασία θεωρήθηκε περιττό να αναπτυχθεί αλγόριθμος για την κωδικοποίηση των πινάκων Huffman σε ακολουθίες από bytes. Για την σύνθεση της ακολουθίας DHT χρησιμοποιήθηκαν οι έτοιμοι πίνακες της ενότητας K.3.3 του προτύπου.

### 4.2 Αποκωδικοποίηση

Η αποκωδικοποίηση ξεκινάει με το σκανάρισμα του header και την εξαγωγή των απαραίτητων μεταδεδομένων για την ανακατασκευή της εικόνας. Αφού ολοκληρωθεί η προαναφερθείσα διαδικασία, ξεκινάει το στάδιο της αποκωδικοποίησης του bitstream, το οποίο περιλαμβάνει όλη την πληροφορία για την εικόνα. Αρχικά, γίνεται μετατροπή του bitstream σε δυαδική μορφή και έπειτα καθορίζεται η σειρά αποκωδικοποίησης ανάλογα με τον τύπο της υποδειγματοληψίας. Τέλος δημιουργείται μια δομή δεδομένων όμοια με το JPEGenc του προηγούμενου παραδοτέου και δίνεται ως όρισμα στην **JPEGdecode**, μέσω της οποίας ανακατασκευάζεται η εικόνα

Συνάρτηση	Σύνοψη
<b>[encodedstream,rows,cols,subimg,qTableL,qTableC, DC_L,DC_C,AC_L,AC_C]=decodeHeader(JPEGencStream)</b>	Αποκωδικοποίηση header
<b>[ DC_L,DC_C,AC_L,AC_C ] = decodeDHTheader( dht )</b>	Αποκωδικοποίηση header για τους πίνακες Huffman
<b>[streambuffer,huffStream]=bufferStream(huffStream,DC,AC )</b>	Δημιουργία buffer με την ακολουθία Huffman για κάθε block της εικόνας

### 4.3 Πείραμα 3:Λόγος Συμπίεσης

Το αρχείο **demo3.m** χρησιμοποιεί το ίδιο εύρος τιμών που ορίστηκε για τις κλίμακες κβαντισμού στο 2<sup>ο</sup> παραδοτέο και παράγει 7 δυαδικά αρχεία για κάθε μια από τις δυο εικόνες που δίνονται για πειραματισμό, αφού αυτές κωδικοποιηθούν απ' την **JPEGencodeStream**. Οι λόγοι συμπίεσης κάθε εικόνας για όλες τις κλίμακες κβαντισμού παρουσιάζονται συνολικά στον παρακάτω πίνακα

Αρχικό μέγεθος εικόνας 1 (Kb)			673	
Αρχικό μέγεθος εικόνας 2 (Kb)			511	
Κλίμακα Κβαντισμού	Εικόνα 1		Εικόνα 2	
	Μέγεθος (Kb)	Λόγος Συμπίεσης	Μέγεθος (Kb)	Λόγος Συμπίεσης
<b>0.1</b>	199	3.3952	94	5.4614
<b>0.3</b>	105	6.4068	52	9.925
<b>0.6</b>	69	9.7649	35	14.7828
<b>1</b>	51	13.4367	26	19.8565
<b>2</b>	32	21.3572	18	29.0928
<b>5</b>	17	40.1058	12	43.5883
<b>10</b>	11	64.2831	10	52.2954