



Τ.Ε.Ι. ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Συμπίεση/αποσυμπίεση εικόνας με χρήση του Διακριτού Μετασχηματισμού Συνημιτόνου
(Discrete Cosine Transform)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
του
ΚΙΡΚΙΡΗ ΘΕΟΔΩΡΟΥ

Επιβλέπων : Δρ. Βασίλειος Μποζαντζής
Ηλεκτρολόγος Μηχανικός ΕΜΠ
Διδάκτωρ του University of Sussex, UK



Τ.Ε.Ι. ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Συμπίεση / αποσυμπίεση εικόνας με χρήση του Διακριτού Μετασχηματισμού Συνημιτόνου
(Discrete Cosine Transform)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
του
ΚΙΡΚΙΡΗ ΘΕΟΔΩΡΟΥ

Επιβλέπων: Δρ. Μποζαντζής Βασίλειος
Ηλεκτρολόγος Μηχανικός ΕΜΠ
Διδάκτωρ του University of Sussex, UK

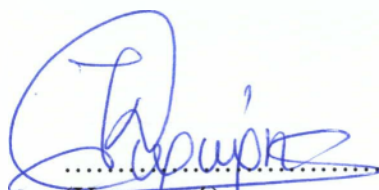
Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....
(Υπογραφή)

.....
(Υπογραφή)

.....
(Υπογραφή)

Σπάρτη, 2014



(Υπογραφή)
ΚΙΡΚΙΡΗΣ ΘΕΟΔΩΡΟΣ

© 2014 – All rights reserved

Περίληψη

Η παρούσα πτυχιακή διαπραγματεύεται τη χρήση του διακριτού μετασχηματισμού συνημιτόνου (DCT) στη συμπίεση εικόνας. Σκοπός της είναι να εισάγει την έννοια της συμπίεσης εικόνας, να αναλύσει με απλό τρόπο τη λογική του μετασχηματισμού και στη συνέχεια να παρουσιάσει τη συμβολή του στη συμπίεση εικόνας μέσα από παραδείγματα στο MATLAB.

Λέξεις Κλειδιά:

«pixel, εικόνα, συμπίεση εικόνας, κωδικοποίηση εικόνας, χρωματικό βάθος, διακριτός μετασχηματισμός συνημιτόνου, DCT»

Abstract

This thesis describes the use of the discrete cosine transform in image compression. Its main purpose is to introduce the concept of image compression, to analyze the logic of the transformation in a simple way and then present its contribution to image compression through examples in MATLAB.

Keywords: “pixel, image, image compression, image coding, bit depth, discrete cosine transform, DCT”

Περιεχόμενα

1 Εισαγωγή.....	1
1.1 Τοποθέτηση του προβλήματος	2
1.2 Στόχος της πτυχιακής.....	8
1.3 Οργάνωση Κειμένου.....	8
2 Συμπίεση εικόνας	9
2.1 Ορισμός πλεοναζόντων δεδομένων	10
2.1.1 Πλεονασμός κωδικοποίησης	11
2.1.2 Πλεονασμός μεταξύ διαφορετικών pixels.....	11
2.1.3 Πλεονασμός που οφείλεται στην ανθρώπινη αντίληψη.....	12
2.2 Συμπίεση με απώλειες	12
2.2.1 Κωδικοποίηση με χρήση μετασχηματισμών.....	12
3 Το πρότυπο JPEG – Περιγραφή του DCT.....	15
3.1 Εισαγωγή	15
3.1.1 Η γέννηση του προτύπου JPEG.....	15
3.1.2 Υπόβαθρο: Απαιτήσεις και διαδικασία επιλογής	15
3.1.3 Αρχιτεκτονική του προτύπου JPEG	16
3.2 Διακριτός μετασχηματισμός συνημιτόνου – DCT.....	17
3.2.1 Εισαγωγή.....	17
3.2.2 Ορισμός.....	18
3.3 Βήματα επεξεργασίας για την 8x8 DCT κωδικοποίηση.....	20
3.3.1 8x8 FDCT-IDCT.....	21
3.3.2 Κβάντιση	23
3.3.3 DC Κωδικοποίηση & Ακολουθία Zig-Zag	24
3.3.4 Κωδικοποίηση εντροπίας	25
3.3.5 Ποιότητα συμπίεσης και εικόνων.....	26
4 Επεξεργασία εικόνας και εφαρμογή του DCT - Παραδείγματα με το MATLAB.....	29
4.1 Εισαγωγή στην επεξεργασία εικόνας.....	29
4.2 Είδη ψηφιακών εικόνων	30
4.3 Επεξεργασία εικόνας στο MATLAB.....	32
4.3.1 Εισαγωγή εικόνας στο MATLAB.....	33
4.3.2 Εφαρμογή του DCT με κλήση της συνάρτησης 'dct2'.....	38
4.3.3 Εφαρμογή του DCT με χρήση πίνακα.....	40
5 Επίλογος	59
6 Παράρτημα – κώδικας MATLAB	63
6.1 Εφαρμογή του DCT με κλήση της συνάρτησης 'dct2'.....	63
6.2 Εφαρμογή του DCT με χρήση πίνακα συντελεστών	64
6.2.1 Βασικό script	64
6.2.2 Αυτοματοποιημένο script	65
Βιβλιογραφικές Αναφορές.....	77

Εικόνες

Εικόνα 1.1 – Τα pixels μίας εικόνας (μεγέθυνση).....	2
Εικόνα 1.2 – Εικόνα με βάθος χρώματος 2 bit	4
Εικόνα 1.3 – Εικόνα με βάθος χρώματος 8 bits μονόχρωμο	4
Εικόνα 1.4 – Εικόνα με βάθος χρώματος 8 bits έγχρωμο	5
Εικόνα 1.5 – Εικόνα με βάθος χρώματος 16 bits (High color)	5
Εικόνα 1.6 – Εικόνα με βάθος χρώματος 24 bits (True color)	6
Εικόνα 1.7 – Εικόνα με βάθος χρώματος 32bit (HDR Toning).....	6
Εικόνα 3.1 – Οι συχνότητες των 64 συναρτήσεων βάσης ενός πίνακα $A_{8 \times 8}$	19
Εικόνα 3.2 – Εφαρμογή του DCT δύο διαστάσεων σε ασπρόμαυρη εικόνα.....	27
Εικόνα 4.1 – Πίνακας απεικόνισης στοιχείων $M \times N$	30
Εικόνα 4.2 – Διακριτές τιμές φωτεινότητας των pixels μιας ασπρόμαυρης εικόνας	31
Εικόνα 4.3 – Διακριτές τιμές φωτεινότητας των pixels μιας grayscale εικόνας.....	31
Εικόνα 4.4 – Διακριτές τιμές φωτεινότητας των pixels μιας έγχρωμης εικόνας	32
Εικόνα 4.5 – Παράθυρο τρέχοντα φακέλου	33
Εικόνα 4.6 – Τρισδιάστατος πίνακας $512 \times 512 \times 3$	33
Εικόνα 4.7 – Εικόνα που προκύπτει από τον πίνακα A	34
Εικόνα 4.8 – Οι δισδιάστατοι πίνακες B και C	34
Εικόνα 4.9 – Εικόνα που προκύπτει από τον πίνακα B.....	35
Εικόνα 4.10 – Εικόνα που προκύπτει από τον πίνακα C.....	35
Εικόνα 4.11 – Απεικόνιση τιμών στο επίπεδο.....	36
Εικόνα 4.12 – Απεικόνιση τιμών στο χώρο	36
Εικόνα 4.13 – Αποτελέσματα τιμών RGB για το pixel στη θέση $(x,y)=(265,493)$	37
Εικόνα 4.14 – Γραφική απεικόνιση της δύναμης συντελεστών DCT	38
Εικόνα 4.15 – Συσχέτιση συντελεστών DCT	38
Εικόνα 4.16 – Γραφική απεικόνιση της δύναμης συντελεστών DCT μετά από επιλεκτικό μηδενισμό	39
Εικόνα 4.17 – Συσχέτιση συντελεστών DCT μετά από επιλεκτικό μηδενισμό.....	39
Εικόνα 4.18 – Ανασύνθεση εικόνας με το 100% των συντελεστών DCT	39
Εικόνα 4.19 – Ανασύνθεση εικόνας με το 10% των συντελεστών DCT	39
Εικόνα 4.20 – Πίνακας συντελεστών DCT πριν την απόρριψη	41
Εικόνα 4.21 – Πίνακας συντελεστών DCT μετά την απόρριψη	41
Εικόνα 4.22 – Πηγαία εικόνα πριν το μετασχηματισμό	42
Εικόνα 4.23 – Πηγαία εικόνα μετά από μετασχηματισμό και απόρριψη συντελεστών DCT	42
Εικόνα 4.24 – Τιμές έντασης φωτεινότητας πηγαίας εικόνας πριν το μετασχηματισμό (3D).	42
Εικόνα 4.25 – Τιμές έντασης φωτεινότητας μετά το μετασχηματισμό και την απόρριψη συντελεστών DCT (3D).....	42
Εικόνα 4.26 – Τιμές έντασης φωτεινότητας πηγαίας εικόνας πριν το μετασχηματισμό (2D).	43
Εικόνα 4.27 – Τιμές έντασης φωτεινότητας μετά το μετασχηματισμό και την απόρριψη συντελεστών DCT (2D).....	43
Εικόνα 4.28 – Τιμές έντασης φωτεινότητας πηγαίας εικόνας πριν το μετασχηματισμό (3D, λεπτομέρεια)	43
Εικόνα 4.29 – Τιμές έντασης φωτεινότητας μετά το μετασχηματισμό και την απόρριψη συντελεστών DCT (3D, λεπτομέρεια).....	43
Εικόνα 4.30 – Πηγαία εικόνα χωρισμένη σε 8×8 blocks και συντελεστές DCT πριν την απόρριψη.....	44

Εικόνα 4.31 - Πηγαία εικόνα χωρισμένη σε 8x8 blocks και συντελεστές DCT μετά την απόρριψη.....	44
Εικόνα 4.32 - Συντελεστές DCT πηγαίας εικόνας (λεπτομέρεια)	44
Εικόνα 4.33 - Συντελεστές DCT μετά την απόρριψη (λεπτομέρεια)	44
Εικόνα 4.34 - Αποτελέσματα για διατήρηση 3/64 συντελεστών	45
Εικόνα 4.35 - Αποτελέσματα για διατήρηση 6/64 συντελεστών.....	46
Εικόνα 4.36 - Αποτελέσματα για διατήρηση 10/64 συντελεστών.....	47
Εικόνα 4.37 - Αποτελέσματα για διατήρηση 15/64 συντελεστών.....	48
Εικόνα 4.38 - Αποτελέσματα για διατήρηση 21/64 συντελεστών.....	49
Εικόνα 4.39 - Αποτελέσματα για διατήρηση 28/64 συντελεστών.....	50
Εικόνα 4.40 - Αποτελέσματα για διατήρηση 36/64 συντελεστών.....	51
Εικόνα 4.41 - Αποτελέσματα για διατήρηση 43/64 συντελεστών.....	52
Εικόνα 4.42 - Αποτελέσματα για διατήρηση 49/64 συντελεστών.....	53
Εικόνα 4.43 - Αποτελέσματα για διατήρηση 54/64 συντελεστών.....	54
Εικόνα 4.44 - Αποτελέσματα για διατήρηση 58/64 συντελεστών.....	55
Εικόνα 4.45 - Αποτελέσματα για διατήρηση 61/64 συντελεστών.....	56
Εικόνα 4.46 - Αποτελέσματα για διατήρηση 63/64 συντελεστών.....	57
Εικόνα 4.47 - Αποτελέσματα για διατήρηση 64/64 συντελεστών.....	58
Εικόνα 5.1 - Παράληψη DC συνιστώσας.....	61

Σχήματα

Σχήμα 2.1 - Βασικό σύστημα συμπίεσης εικόνας	13
Σχήμα 3.1 - Βήματα επεξεργασίας DCT κωδικοποιητή	20
Σχήμα 3.2 - Βήματα επεξεργασίας DCT αποκωδικοποιητή	20
Σχήμα 3.3 - Διαφορική DC κωδικοποίηση	25
Σχήμα 3.4 - Zig-Zag ακολουθία	25

Πίνακες

Πίνακας 1.1 – Κυριότερες τιμές χρωματικού βάθους	3
Πίνακας 1.2 – Απαιτούμενος αποθηκευτικός χώρος για ασυμπίεστη εικόνα 8MP	7
Πίνακας 2.1 – Βασικές μέθοδοι απωλεστικής / μη απωλεστικής συμπίεσης	9
Πίνακας 3.1 - Επίπεδα συμπίεσης και ποιότητα εικόνας	26

Εξισώσεις

Εξίσωση 3.1 – DCT δύο διαστάσεων, πίνακα $AM \times N$	18
Εξίσωση 3.2 – Γενική εξίσωση FDCT για πίνακες $AN \times N$	21
Εξίσωση 3.3 - Γενική εξίσωση IDCT για πίνακες $AN \times N$	21
Εξίσωση 3.4 - Εξίσωση FDCT για πίνακες $A8 \times 8$	21
Εξίσωση 3.5 - Εξίσωση IDCT για πίνακες $A8 \times 8$	21

ΕΠΙΣΤΗΜΟΝΙΚΗ ΕΡΕΥΝΑ

1990

1

Εισαγωγή

Μάτι ή οφθαλμός: Το αισθητήριο όργανο της όρασης των ζωντανών οργανισμών. Μέσω αυτού λαμβάνονται τα οπτικά ερεθίσματα που στέλνονται στον εγκέφαλο ο οποίος τα μεταφράζει σε εικόνες.



Εικόνα¹: Η αποτύπωση (απομνημόνευση) της εμφάνισης ενός ή περισσότερων αντικειμένων που μπορεί να γίνει σε διάφορα μέσα όπως το χαρτί, η μνήμη και η οθόνη ενός υπολογιστή ή μιας φωτογραφικής μηχανής, το ανθρώπινο μυαλό κλπ. Μία στατική εικόνα περιέχει συνήθως πληροφορία δύο ή τριών διαστάσεων ενώ η «κινούμενη» εικόνα (βίντεο) έχει την έννοια της απεικόνισης ενός θέματος, στο χρόνο. Ψηφιακή εικόνα είναι η αναπαράσταση της εικόνας όπως αυτή αναφέρθηκε νωρίτερα,

αλλά σε ένα υπολογιστικό σύστημα με χρήση της δυαδικής λογικής.

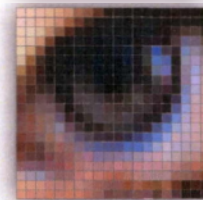
Η εικόνα είναι αδιαμφισβήτητα μία από τις σημαντικότερες πηγές πληροφορίας από την προϊστορική κιόλας εποχή όπου ο άνθρωπος χρησιμοποιεί εικόνες για να αποτυπώσει και να περιγράψει πρόσωπα και καταστάσεις. Ακόμη και στη σύγχρονη εποχή της πληροφορίας, τόσο η μετάδοση όσο και η αποθήκευση εικόνων είναι κεντρικής σημασίας σε διάφορες εφαρμογές.

¹ Επάνω: Lena Söderberg, γεννημένη 31 Μαρτίου 1951 στη Σουηδία, εμφανίστηκε στο ως Playmate με το όνομα Lenna Sjöobloom. Η φωτογραφία τραβήχτηκε από τον Dwight Hooker και δημοσιεύτηκε στο περιοδικό Playboy, τεύχος Νοεμβρίου 1972. Έκτοτε αποτελεί την πηγαία εικόνα σε μεγάλη πλειοψηφία μελετών και δημοσιεύσεων, σχετικών με την επεξεργασία ψηφιακής εικόνας.

Η μετάδοση ψηφιακής εικόνας βρίσκει εφαρμογή (μεταξύ άλλων) στην τηλεόραση, την τηλεδιάσκεψη, τη video-τηλεφωνία, την τηλεομοιοτυπία, τις δορυφορικές επικοινωνίες, τις επικοινωνίες υπολογιστών αλλά και στις στρατιωτικές επικοινωνίες σε αεροπλάνα, radar και sonar κλπ. Αντίστοιχα η αποθήκευση εφαρμόζεται σε πολυμέσα, ρομποτική, υπολογιστική όραση και σε αποθήκευση εκπαιδευτικών και επιχειρησιακών δεδομένων, κινουμένων εικόνων, δορυφορικών εικόνων, γεωφυσικών εικόνων, και ιατρικών εικόνων κλπ.

1.1 Τοποθέτηση του προβλήματος

Μία αρχαία κινέζικη παροιμία λέει «μία εικόνα αξίζει όσο χίλιες λέξεις», θέλοντας να δώσει έμφαση στην τεράστια ποσότητα πληροφορίας που περιέχεται σε μία και μόνο εικόνα. Τα βασικά δομικά στοιχεία από τα οποία αποτελείται μία εικόνα στην ψηφιακή της μορφή ονομάζονται pixels (Εικόνα 1.1) και αναπαριστώνται με τη μορφή δισδιάστατου πίνακα. Ο αριθμός των pixels για κάθε εικόνα είναι σταθερός, συνεπώς όσο μεγαλύτερο είναι το πλήθος τους, τόσο καλύτερη είναι η ποιότητα και η ευκρίνεια της εικόνας.



Εικόνα 1.1– Τα pixels μίας εικόνας (μεγέθυνση)

Κάθε pixel αντιστοιχίζεται με ένα πολύ συγκεκριμένο σημείο της εικόνας και η τιμή του, η οποία αντιπροσωπεύει την ένταση φωτεινότητας του χρώματος στο συγκεκριμένο σημείο, αποτελείται από μία ή περισσότερες συνιστώσες ανάλογα με το χρωματικό βάθος. Χρωματικό βάθος (ή βάθος χρώματος) καλείται το πλήθος των bits που χρησιμοποιούνται για την αναπαράσταση του χρώματος κάθε pixel και εκφράζεται με δυνάμεις του 2 (

Πίνακας 1.1).

Bits	Αριθμός χρωμάτων	Περιγραφή	Παράδειγμα
1-bit monochrome	$2^1 = 2$	Ασπρόμαυρη εικόνα, χωρίς διαβαθμίσεις του γκριζου. Κάθε pixel μπορεί να πάρει μόνο 2 τιμές, 0 ή 1 (Binary)	Εικόνα 1.2
8-bit grayscale	$2^8 = 256$	Πρόκειται για εικόνα σε αποχρώσεις του γκριζου	Εικόνα 1.3
8-bit color	$2^8 = 256$	Έγχρωμη εικόνα, με περιορισμένο αριθμό χρωμάτων	Εικόνα 1.4
16-bit High Color	$2^{16} = 65.536$	Έγχρωμη εικόνα με αρκετά καλή ποιότητα χρώματος. Το βάθος χρώματος αναφέρεται και ως High Color	Εικόνα 1.5
24-bit True Color	$2^{24} = 16.777.216$	Έγχρωμη εικόνα με άριστη ποιότητα χρώματος. Το βάθος χρώματος αναφέρεται και ως True Color	Εικόνα 1.6
30-bit 36-bit 48-bit Deep Color	$2^{30} = 1.073.741.824$ $2^{36} = 68.719.476.736$ $2^{48} = 281.474.976.710.656$	Έγχρωμη εικόνα με περισσότερα από 8bits / χρώμα. Χρησιμοποιείται σε ειδικές εφαρμογές για μεγαλύτερο εύρος έντασης χρώματος (π.χ. high dynamic range imaging)	

Πίνακας 1.1 – Κυριότερες τιμές χρωματικού βάθους



Εικόνα 1.2 – Εικόνα με βάθος χρώματος 2 bit



Εικόνα 1.3 – Εικόνα με βάθος χρώματος 8 bits μονόχρωμο



Εικόνα 1.4 - Εικόνα με βάθος χρώματος 8 bits έγχρωμο



Εικόνα 1.5 - Εικόνα με βάθος χρώματος 16 bits (High color)



Εικόνα 1.6 – Εικόνα με βάθος χρώματος 24 bits (True color)



Εικόνα 1.7 – Εικόνα με βάθος χρώματος 32bit (HDR Toning)

Όσο αυξάνονται η ανάλυση και το βάθος χρώματος μιας ψηφιακής εικόνας, τόσο αυξάνεται και ο χώρος που απαιτείται για την αποθήκευση της. Στον παρακάτω πίνακα (

Πίνακας 1.2) αναγράφεται ο αποθηκευτικός χώρος που απαιτείται για μία ασυμπίεστη εικόνα ανάλυσης 8MP ($= 8 \cdot 10^6$ pixels), με διαφορετικό βάθος χρώματος κάθε φορά. Ενδεικτικά αναφέρεται ότι μία φωτογραφική μηχανή με αποθηκευτικό χώρο 1GB, μπορεί να αποθηκεύσει 40 περίπου ασυμπίεστες φωτογραφίες με 24-bit βάθος χρώματος.

Βάθος χρώματος	Απαιτούμενος αποθηκευτικός χώρος
1-bit	0,95 MB
8-bit	7,63 MB
16-bit	15,26 MB
24-bit	22,89 MB

Πίνακας 1.2 – Απαιτούμενος αποθηκευτικός χώρος για ασυμπίεστη εικόνα 8MP

Δεδομένου του τεράστιου όγκου δεδομένων, του περιορισμένου χώρου αποθήκευσης καθώς και των πεπερασμένων ταχυτήτων μεταφοράς των δεδομένων, είναι προφανές ότι σε εφαρμογές που απαιτούν την αποθήκευση ή τη μεταφορά εικόνων, η συμπίεση τους αποτελεί απαραίτητη προϋπόθεση.

Προκειμένου να καταστεί δυνατή η χρήση της ψηφιακής εικόνας σε ένα ευρύ πεδίο εφαρμογών, έπρεπε να βρεθούν τεχνικές συμπίεσης ώστε να μειωθεί ο απαιτούμενος αποθηκευτικός χώρος, χωρίς ωστόσο να υπάρχει σημαντική απώλεια πληροφορίας. Η αναζήτηση αυτή οδήγησε στην ανάπτυξη ενός κλάδου επεξεργασίας της εικόνας που επικεντρώνεται στη συμπίεση ψηφιακής εικόνας.

Συμπίεση καλείται η διαδικασία ελάττωσης των δεδομένων που απαιτούνται για την αναπαράσταση μιας μορφής πληροφορίας. Ο όρος «συμπίεση εικόνας» είναι συνώνυμος στη διεθνή βιβλιογραφία με την «κωδικοποίηση εικόνας», καθώς οι κωδικοποιητές εικόνας συμπεριλαμβάνουν επιμέρους βαθμίδες συμπίεσης. Με βάση τις επιμέρους βαθμίδες συμπίεσης, οι τεχνικές συμπίεσης εικόνας μπορούν να ταξινομηθούν στις τεχνικές κωδικοποίησης εντάσεων των pixels, στις τεχνικές κωδικοποίησης με χρήση μετασχηματισμών και στις υβριδικές τεχνικές που αποτελούν συνδυασμό των παραπάνω.

1.2 Στόχος της πτυχιακής

Ο βασικός στόχος της παρούσας πτυχιακής εργασίας είναι να περιγράψει το διακριτό μετασχηματισμό συνημιτόνου και να παρουσιάσει τα αποτελέσματα της εφαρμογής του στη συμπίεση εικόνας, μέσα από παραδείγματα στο MATLAB.

1.3 Οργάνωση Κειμένου

Στο Κεφάλαιο 1 έγινε μία εισαγωγή στην ψηφιακή εικόνα, τις εφαρμογές της, καθώς και στο πως γίνεται η διαχείριση της από τα σύγχρονα ψηφιακά υπολογιστικά συστήματα. Στα κεφάλαια που ακολουθούν γίνεται μια εισαγωγή στη συμπίεση εικόνας (Κεφάλαιο 2) και του διακριτού μετασχηματισμού συνημιτόνου (Κεφάλαιο 4), ενώ ακολουθούν παραδείγματα στο MATLAB.

Πιο συγκεκριμένα, στο Κεφάλαιο 2 αναλύεται η συμπίεση της εικόνας, ο πλεονασμός των δεδομένων καθώς και οι τεχνικές απώλεστικής συμπίεσης, ενώ στο Κεφάλαιο 3 γίνεται αναφορά στο πρότυπο JPEG και την αρχιτεκτονική του, καθώς και στη σχέση του με το διακριτό μετασχηματισμό συνημιτόνου. Στο Κεφάλαιο 4 γίνεται εισαγωγή στην επεξεργασία εικόνας στο MATLAB και περιγράφεται αναλυτικά ο τρόπος εφαρμογής του DCT μέσα από παραδείγματα, ενώ παρατίθενται και αναλυτικά αποτελέσματα.

Στο Κεφάλαιο 5 αναλύονται τα συμπεράσματα που προέκυψαν από την ανάλυση εικόνας στο Κεφάλαιο 4, ενώ στο Κεφάλαιο 6 παρατίθεται ο κώδικας MATLAB που χρησιμοποιήθηκε κατά την εκπόνηση της πτυχιακής εργασίας. Τέλος, στο Κεφάλαιο 7 υπάρχουν οι βιβλιογραφικές αναφορές.

2

Συμπίεση εικόνας

Αντίστοιχα με τη συμπίεση δεδομένων, συμπίεση εικόνας καλείται η διαδικασία ελάττωσης των δεδομένων που απαιτούνται για την αναπαράσταση μίας ψηφιακής εικόνας. Ανάλογα με τη μορφή συμπίεσης, διακρίνουμε δύο βασικές κατηγορίες: α) απωλεστική συμπίεση και β) μη απωλεστική συμπίεση. Η απωλεστική (lossy) συμπίεση οδηγεί σε απώλεια πληροφορίας προκειμένου να επιτευχθεί το επιθυμητό αποτέλεσμα, ενώ αντίθετα η μη απωλεστική (lossless) διατηρεί την ακεραιότητα των δεδομένων.

Η επιλογή της μεθόδου που θα ακολουθηθεί κάθε φορά, εξαρτάται από την εφαρμογή για την οποία αυτή προορίζεται. Εφαρμογές όπως η αποθήκευση εγγράφων επιχειρήσεων ή ιατρικών δεδομένων (π.χ. ακτινογραφίες) απαιτούν συμπίεση χωρίς απώλειες. Υπάρχουν ωστόσο και εφαρμογές (π.χ. η αποθήκευση οικιακών ψηφιακών φωτογραφιών) στις οποίες οι απώλειες είναι μέχρι ένα σημείο ανεκτές, επομένως μπορεί να εφαρμοστεί μια μέθοδος που αλλοιώνει σε κάποιο βαθμό την αρχική εικόνα προκειμένου να επιτευχθεί μεγαλύτερος βαθμός συμπίεσης. Στον παρακάτω πίνακα, αναφέρονται μερικές βασικές μέθοδοι απωλεστικής / μη απωλεστικής συμπίεσης (Πίνακας 2.1).

Είδη μη απωλεστικής συμπίεσης	Είδη απωλεστικής συμπίεσης
Run-length encoding (π.χ. BMP, TGA, TIFF)	Μείωση του χρωματικού χώρου (color space) στα χρώματα που εμφανίζονται περισσότερο στην εικόνα.
DPCM	Υποδειγματοληψία χρωματικότητας
LZW (π.χ. GIF)	Μέσω μετασχηματισμού (π.χ. Fourier, DCT, wavelet)
Deflation (π.χ. PNG)	Fractal compression

Πίνακας 2.1 – Βασικές μέθοδοι απωλεστικής / μη απωλεστικής συμπίεσης

Από μαθηματική άποψη, η συμπίεση μιας εικόνας είναι ο μετασχηματισμός ενός δισδιάστατου πίνακα από pixels σε ένα σύνολο στατιστικά ασυσχέτιστων δεδομένων. Εφόσον υπάρχει κάποιος βαθμός συσχέτισης στα δεδομένα αυτά, σημαίνει ότι θεωρητικά είναι δυνατή η συμπίεση της εικόνας σε μεγαλύτερο βαθμό. Ο μετασχηματισμός αυτός εφαρμόζεται πριν την αποθήκευση ή τη μετάδοση της εικόνας. Για να την αποσυμπιέσουμε, να ανακτήσουμε δηλαδή είτε την αρχική εικόνα είτε μια προσέγγιση αυτής, εφαρμόζουμε στο σύνολο των δεδομένων που αποθηκεύτηκαν ή μεταδόθηκαν τον αντίστροφο του μετασχηματισμού που εφαρμόσαμε κατά τη συμπίεση.

Ένας αλγόριθμος συμπίεσης κρίνεται ως βέλτιστος αν μπορεί να οδηγήσει σε μεγάλη συμπίεση δεδομένων, χωρίς να εισάγει μεγάλη παραμόρφωση, ενώ παράλληλα έχει μικρή υπολογιστική πολυπλοκότητα. Οι απαιτήσεις των αλγορίθμων συμπίεσης σε υπολογισμούς είναι ιδιαίτερα υψηλή, καθώς αφενός το πλήθος των δεδομένων που επεξεργάζονται είναι ιδιαίτερα μεγάλο, αφετέρου η φιλοσοφία των τεχνικών συμπίεσης απαιτεί μεγάλο αριθμό υπολογισμών.

Η συμπίεση μιας εικόνας κρίνεται απαραίτητη για την εξοικονόμηση αποθηκευτικού χώρου και εύρους ζώνης κατά τη μεταφορά και αποθήκευση δεδομένων. Το πρόβλημα της συμπίεσης συνίσταται στη μείωση των δεδομένων που απαιτούνται για την αναπαράσταση της εικόνας αυτής και μπορεί να επιτευχθεί όταν μια εικόνα περιέχει πλεονάζοντα δεδομένα (δεδομένα δηλαδή που δεν είναι απαραίτητα για την αναπαράσταση της). Υπάρχουν διάφορα είδη πλεοναζόντων δεδομένων που μπορούν να αφαιρεθούν προκειμένου να καταστεί αυτό δυνατό.

2.1 Ορισμός πλεοναζόντων δεδομένων

Αρχικά είναι σημαντικό να διακρίνουμε τη διαφορά μεταξύ πληροφορίας (information) και δεδομένων (data). Στην περίπτωση μιας ψηφιακής εικόνας, **πληροφορία** είναι η τιμή έντασης της φωτεινότητας καθενός από τα pixels που την αποτελούν. Από την άλλη μεριά, **δεδομένα** είναι το μέσο αναπαράστασης αυτής της πληροφορίας. Η ίδια πληροφορία μπορεί να αναπαρασταθεί με πολλούς διαφορετικούς τρόπους, δηλαδή με πολλά διαφορετικά σύνολα δεδομένων. Η συμπίεση μιας εικόνας λοιπόν έγκειται στην αναζήτηση ενός συνόλου δεδομένων, που μπορούν να αναπαραστήσουν επακριβώς την ίδια την πληροφορία (ή και με ορισμένη αλλοίωση όπου αυτό είναι αποδεκτό), με το μικρότερο δυνατό μέγεθος. Η παραπάνω προσέγγιση μπορεί να εκφραστεί ποσοτικά μέσα από μαθηματικούς τύπους.

Υποθέτουμε ότι ένα σύνολο δεδομένων που περιέχει n_1 μονάδες (για παράδειγμα bits) αναπαριστά μια ορισμένη πληροφορία (μια ψηφιακή εικόνα στην περίπτωση μας). Αν η ίδια πληροφορία μπορεί να αναπαρασταθεί από ένα σύνολο δεδομένων που περιέχει n_2 μονάδες, τότε ο λόγος συμπίεσης C_R ορίζεται ως:

$$C_R = \frac{n_1}{n_2}$$

και ο σχετικός πλεονασμός δεδομένων ως

$$R_D = 1 - \frac{1}{C_R}$$

Για παράδειγμα, αν $\frac{n_1}{n_2} = 2$, τότε ο λόγος συμπίεσης είναι 2 και ο σχετικός πλεονασμός δεδομένων είναι 0,5 που σημαίνει ότι τα μισά από τα δεδομένα του πρώτου συνόλου είναι πλεονάζοντα, κάτι που όντως ισχύει, αφού η ίδια πληροφορία μπορεί να αναπαρασταθεί από το δεύτερο σύνολο με ακριβώς τις μισές μονάδες. Στις ψηφιακές εικόνες εντοπίζονται τρία διαφορετικά είδη πλεονασμού:

- πλεονασμός κωδικοποίησης
- πλεονασμός μεταξύ διαφορετικών pixels
- πλεονασμός που αφορά τον τρόπο που αντιλαμβάνεται το ανθρώπινο μάτι μια εικόνα.

Οι αλγόριθμοι συμπίεσης εικόνας προσπαθούν να μειώσουν ή ακόμα και να εξαλείψουν ένα ή περισσότερα από αυτά τα είδη πλεοναζόντων δεδομένων.

2.1.1 Πλεονασμός κωδικοποίησης

Ο κώδικας που επιλέγεται για την αναπαράσταση μιας πληροφορίας παίζει σημαντικό ρόλο στον συνολικό όγκο των δεδομένων που θα προκύψει για την αποθήκευση αυτής της πληροφορίας. Αν σε μια ψηφιακή εικόνα που περιέχει n pixels και L διαφορετικά επίπεδα φωτεινότητας, η πιθανότητα εμφάνισης του k -οστού επιπέδου φωτεινότητας είναι p_k και το μήκος της κωδικής λέξης που χρησιμοποιείται για το συμβολισμό αυτού του επιπέδου φωτεινότητας είναι l_k , τότε το μέσο μήκος για την αναπαράσταση κάθε pixel είναι:

$$L_{\text{average}} = \sum_{k=0}^{L-1} l_k * p_k$$

Από την παραπάνω σχέση προκύπτει ότι αν αναθέσουμε σε επίπεδα φωτεινότητας που εμφανίζονται συχνά (δηλαδή μεγάλο p_k), κωδικές λέξεις μικρού μήκους και σε επίπεδα φωτεινότητας που εμφανίζονται σπάνια (δηλαδή μικρό p_k), κωδικές λέξεις μεγάλου μήκους, μπορούμε να πετύχουμε ένα μικρότερο μέσο μήκος λέξης για την αναπαράσταση κάθε pixel σε σχέση με τον κοινό δυαδικό κώδικα.

2.1.2 Πλεονασμός μεταξύ διαφορετικών pixels

Αυτό το είδος πλεονασμού εμφανίζεται εξαιτίας της αυτοσυσχέτισης που υπάρχει μεταξύ διαφορετικών pixels σε μία εικόνα. Όσο μεγαλύτερη είναι η

αυτοσυσχέτιση, τόσο μεγαλύτερος είναι και ο πλεονασμός που υπάρχει μέσα στην εικόνα, αν κάθε pixel έχει κωδικοποιηθεί ανεξάρτητα από τα υπόλοιπα. Αυτό συμβαίνει γιατί η πληροφορία που φέρει κάθε μεμονωμένο pixel είναι μικρή, μιας και μπορεί να προβλεφθεί από τα γειτονικά του χρησιμοποιώντας λιγότερα δεδομένα από αυτά που θα χρειάζονταν, αν θεωρούσαμε το pixel αυτό απομονωμένο από την υπόλοιπη εικόνα.

2.1.3 Πλεονασμός που οφείλεται στην ανθρώπινη αντίληψη

Η αφαίρεση αυτού του είδους πλεονασμού, οδηγεί σε απώλεια πληροφορίας, σε αντίθεση με τις δύο προηγούμενες περιπτώσεις, όπου δεν υπάρχει απώλεια. Επειδή το ανθρώπινο μάτι δεν αντιλαμβάνεται όλα τα στοιχεία μιας εικόνας με την ίδια σημασία, μπορεί να αφαιρεθεί ένα μέρος της πληροφορίας χωρίς σημαντικές επιπτώσεις στο τελικό οπτικό αποτέλεσμα που θα προκύψει.

Πιο συγκεκριμένα, ο ανθρώπινος εγκέφαλος δεν επεξεργάζεται τα pixels μιας εικόνας ένα προς ένα, αλλά εντοπίζει συνήθως χαρακτηριστικές περιοχές της εικόνας, όπως οι ακμές και οι υφές που παρουσιάζει, και στη συνέχεια συνδυάζει την πληροφορία αυτή με προηγούμενη γνώση για να παράγει το τελικό αποτέλεσμα.

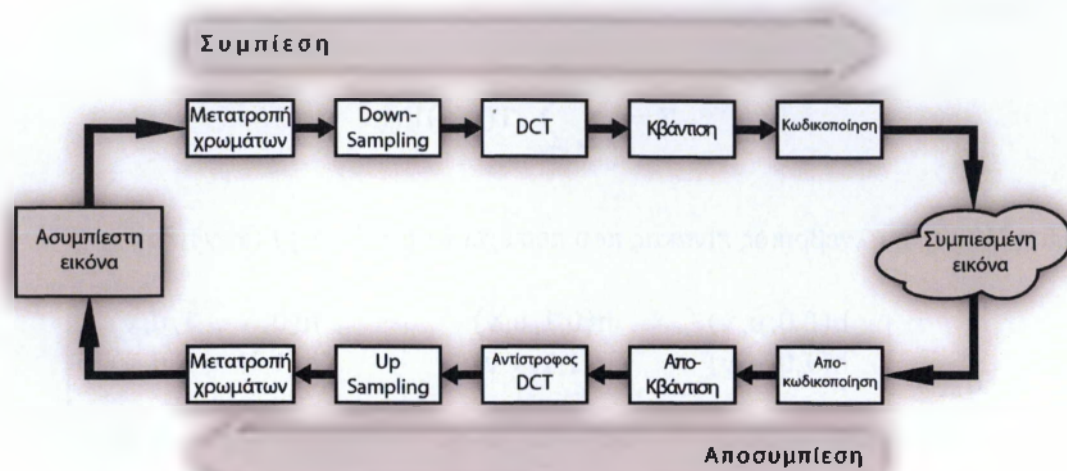
Ενώ δηλαδή στον πλεονασμό κωδικοποίησης και πλεονασμό μεταξύ διαφορετικών pixels μιλάμε για πλεονάζοντα δεδομένα τα οποία δεν είναι απαραίτητα για την ακριβή αναπαράσταση μιας συγκεκριμένης πληροφορίας, στην περίπτωση του πλεονασμού που οφείλεται στην ανθρώπινη αντίληψη, αναφερόμαστε σε πλεονάζουσα πληροφορία που μπορεί να αφαιρεθεί γιατί δεν είναι ιδιαίτερα σημαντική για το οπτικό αποτέλεσμα που αντιλαμβάνεται ο άνθρωπος.

2.2 Συμπίεση με απώλειες

2.2.1 Κωδικοποίηση με χρήση μετασχηματισμών

Στην παράγραφο αυτή θα αναφερθούμε σε μεθόδους που επιδρούν σε διάφορους μετασχηματισμούς εικόνας. Οι μετασχηματισμοί αυτοί πρέπει να είναι γραμμικοί και αντιστρεπτοί, όπως π.χ. ο διακριτός μετασχηματισμός Fourier (Discrete Fourier Transform – DFT) ή ο διακριτός μετασχηματισμός συνημιτόνου (Discrete Cosine Transform – DCT).

Στο Σχήμα 2.1 φαίνεται η γενική μέθοδος που ακολουθείται. Μία εικόνα μεγέθους $N \times N$ διαιρείται σε μικρότερες εικόνες $n \times n$, σε κάθε μία από τις οποίες εφαρμόζεται ο μετασχηματισμός. Αρκετοί από τους συντελεστές που προκύπτουν έχουν μικρή σημασία για την αναπαράσταση της εικόνας και συνεπώς μπορούν να κβαντιστούν ή ακόμα και να αφαιρεθούν τελείως χωρίς σημαντική επίπτωση στην εικόνα που θα προκύψει από την αποσυμπίεση. Στη συνέχεια οι συντελεστές κωδικοποιούνται.



Σχήμα 2.1 - Βασικό σύστημα συμπίεσης εικόνας

Ο κώδικας μπορεί να είναι κάποιος κώδικας μεταβλητού μήκους. Στον αποκωδικοποιητή ακολουθείται η αντίστροφη διαδικασία: αποκωδικοποίηση συντελεστών, αντίστροφος μετασχηματισμός, ένωση των επιμέρους πχη εικόνων για τον σχηματισμό της $N \times N$ εικόνας. Η μόνη διαδικασία που είναι μη αντιστρεπτή (και συνεπώς δεν υπάρχει αντίστροφη διαδικασία στον αποκωδικοποιητή), είναι η κβάντιση. Εκεί εξάλλου υφίσταται και η απώλεια πληροφορίας κατά τη συμπίεση της εικόνας. Τα βήματα που ακολουθούνται μπορούν είτε να προσαρμόζονται ειδικά για κάθε εικόνα είτε όχι, οπότε μιλάμε για προσαρμοστική ή μη προσαρμοστική κωδικοποίηση με χρήση μετασχηματισμών.

Ο μετασχηματισμός που θα χρησιμοποιηθεί μπορεί να είναι οποιοσδήποτε δισδιάστατος διακριτός μετασχηματισμός. Η επιλογή του σχετίζεται καθαρά με την εφαρμογή (δηλαδή το επιτρεπτό σφάλμα) και τους διαθέσιμους υπολογιστικούς πόρους. Η συμπίεση, κατά συνέπεια και η απώλεια πληροφορίας, δεν πραγματοποιείται κατά τη διαδικασία μετασχηματισμού αλλά κατά τη διαδικασία της κβάντισης. Οι μετασχηματισμοί άλλωστε είναι αντιστρεπτοί.

Θεωρούμε μία $N \times N$ εικόνα $f(x, y)$, η οποία στην πραγματικότητα αποτελεί μία από τις εικόνες που προκύπτουν από κατακερματισμό της αρχικής σε ένα πλήθος μικρότερων εικόνων, και έναν μετασχηματισμό αυτής $T(u, v)$ ο οποίος εκφράζεται με τη γενική σχέση:

$$T(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x, y, u, v), \quad u, v = 0, 1, 2, \dots, N-1$$

Ο αντίστροφος μετασχηματισμός θα ορίζεται από μία σχέση της μορφής:

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u, v) h(x, y, u, v), \quad x, y = 0, 1, 2, \dots, N-1$$

Η παραπάνω σχέση μπορεί να γραφεί γενικά ως:

$$F = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u, v) H_{uv}$$

όπου F είναι ο αλγεβρικός πίνακας που περιέχει τα pixels της $f(x, y)$ και

$$H_{uv} = \begin{bmatrix} h(0,0, u, v) & h(0,1, u, v) & \dots & h(0, N-1, u, v) \\ h(1,0, u, v) & h(1,1, u, v) & \dots & h(1, N-1, u, v) \\ \vdots & \vdots & \ddots & \vdots \\ h(N-1,0, u, v) & h(N-1,1, u, v) & \dots & h(N-1, N-1, u, v) \end{bmatrix}$$

Στη συνέχεια ορίζουμε μία συνάρτηση μάσκας για τους συντελεστές του μετασχηματισμού:

$$m(u, v) = \begin{cases} 0, & \text{αν } T(u, v) \text{ ικανοποιεί το κριτήριο αποκοπής} \\ 1, & \text{σε οποιαδήποτε άλλη περίπτωση} \end{cases}$$

Ο πίνακας F μπορεί τότε να προσεγγιστεί από τη σχέση:

$$\hat{F} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u, v) m(u, v) H_{uv}$$

Το μέσο τετραγωνικό σφάλμα που προκύπτει από την προσέγγιση αυτή είναι:

$$\begin{aligned} e_{ms} &= E\{\|F - \hat{F}\|^2\} = E\left\{\left\|\sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u, v) H_{uv} [1 - m(u, v)]\right\|^2\right\} = \\ &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sigma_{T(u,v)}^2 [1 - m(u, v)] \end{aligned}$$

Όπου $\sigma_{T(u,v)}^2$ είναι η διασπορά του συντελεστή $T(u, v)$ του μετασχηματισμού.

Συνεπώς, το μέσο τετραγωνικό σφάλμα είναι ίσο με το άθροισμα των διασπορών όλων των συντελεστών που αποκόπηκαν (όταν ένας συντελεστής διατηρείται, τότε $m(u, v) = 1$, επομένως $1 - m(u, v) = 0$ και η διασπορά του δεν περιλαμβάνεται στο άθροισμα). Οι μετασχηματισμοί που καταφέρνουν να μεταφέρουν περισσότερη πληροφορία σε μικρότερο πλήθος συντελεστών, χαρακτηρίζονται από μικρότερο σφάλμα συμπίεσης. Ο μετασχηματισμός που συνήθως χρησιμοποιείται είναι ο διακριτός μετασχηματισμός συνημιτόνου (DCT), λόγω του μικρού σφάλματος που τον χαρακτηρίζει.

3

Το πρότυπο JPEG – Περιγραφή του DCT

3.1 Εισαγωγή

3.1.1 Η γέννηση του προτύπου JPEG

Από τις αρχές του 1980, η ψηφιακή τεχνολογία παρουσίαζε αλματώδη ανάπτυξη από όλες τις πλευρές. Οι συσκευές απόκτησης εικόνων, αποθήκευσης δεδομένων, εκτύπωσης και εμφάνισης εικόνων με δομική βάση το bit, έφεραν στην επιφάνεια ένα μεγάλο αριθμό από εφαρμογές της ψηφιακής εικόνας, οι οποίες όμως ήταν πολύ δαπανηρές και εξειδικευμένες λόγω του μεγάλου κόστους τους. Η ανάγκη ωστόσο για επέκτασή τους σε γενικής χρήσεως υπολογιστικά συστήματα όπως το κείμενο και τα γεωμετρικά γραφικά ήταν εμφανής. Το κυριότερο εμπόδιο για πολλές εφαρμογές ήταν ο όγκος των δεδομένων που απαιτείται για την απευθείας αναπαράσταση μιας ψηφιακής εικόνας και η μόνη λύση ήταν η συμπίεση της. Από μόνη της όμως η τεχνολογία συμπίεσης δεν είναι αρκετή. Είναι υψίστης σημασίας ένα πρότυπο να εξασφαλίζει διαλειτουργικότητα σε διάφορα περιβάλλοντα, συσκευές και εφαρμογές. Το κενό αυτό ήρθε να καλύψει το JPEG.

3.1.2 Υπόβαθρο: Απαιτήσεις και διαδικασία επιλογής

Ο στόχος του JPEG ήταν να αναπτυχθεί μια μέθοδος για συμπίεση εικόνας συνεχών τόνων που πληροί τις ακόλουθες απαιτήσεις:

- Να είναι η τελευταία λέξη της τεχνολογίας όσον αφορά το ρυθμό συμπίεσης και την πιστότητα της εικόνας σε ένα ευρύ φάσμα ποιοτικών εκτιμήσεων εικόνας, και ειδικότερα η οπτική πιστότητα της εικόνας να χαρακτηρίζεται από πολύ καλή έως άριστη. Επίσης, ο κωδικοποιητής πρέπει να είναι παραμετροποιήσιμος ώστε η εφαρμογή (ή ο χρήστης) να μπορεί να ορίσει την επιθυμητή αναλογία συμπίεσης/ποιότητας.

- Να εφαρμόζεται σχεδόν σε οποιοδήποτε είδος ψηφιακής πηγαίας εικόνας συνεχών τόνων (δηλ. για τους περισσότερους πρακτικούς σκοπούς, να μην περιορίζεται σε εικόνες ορισμένων διαστάσεων, χρώματος, κλπ.) και να μην περιορίζεται σε κατηγορίες εικόνων με περιορισμούς στο περιεχόμενο των σκηνών, όπως η πολυπλοκότητα, το εύρος χρωμάτων, ή οι στατιστικές ιδιότητες.
- Να έχει μικρή υπολογιστική πολυπλοκότητα ώστε να είναι εφικτές εφαρμογές λογισμικού με λογική απόδοση σε ένα μεγάλο εύρος επεξεργαστών, καθώς επίσης και υλοποιήσεις υλικού με βιώσιμο κόστος για εφαρμογές που απαιτούν την υψηλή απόδοση.
- Να έχει τις ακόλουθες καταστάσεις λειτουργίας:
 - Ακολουθιακή κωδικοποίηση: κάθε τμήμα της εικόνας κωδικοποιείται με μια απλή σάρωση από αριστερά προς τα δεξιά και από πάνω προς τα κάτω.
 - Προοδευτική κωδικοποίηση: η εικόνα κωδικοποιείται με πολλαπλές σαρώσεις για εφαρμογές στις οποίες ο χρόνος μετάδοσης είναι μεγάλος, και ο θεατής προτιμά να παρακολουθήσει την εικόνα να εμφανίζεται με πολλαπλά περάσματα (από μια θολή εικόνα στην τέλεια εικόνα).
 - Χωρίς απώλειες κωδικοποίηση: η εικόνα κωδικοποιείται έτσι ώστε να είναι εγγυημένη η ακριβής αποκατάσταση κάθε τιμής δείγματος της πηγαίας εικόνας (ακόμα κι αν το αποτέλεσμα είναι χαμηλή συμπίεση).
 - Ιεραρχική κωδικοποίηση: η εικόνα κωδικοποιείται σε πολλαπλές αναλύσεις έτσι ώστε οι χαμηλής-ανάλυσης εκδόσεις να μπορούν να ανακτηθούν χωρίς να πρέπει να αποσυμπίεστεί η εικόνα σε πλήρη ανάλυση.

Τον Ιούνιο του 1987, το JPEG διεξήγαγε μια διαδικασία επιλογής ανάμεσα σε 12 προτάσεις για μεθόδους συμπίεσης (βασισμένη στην τυφλή αξιολόγηση της υποκειμενικής ποιότητας μιας εικόνας), και μείωσε τις προτεινόμενες μεθόδους σε τρεις. Τρεις άτυπες ομάδες εργασίας συστήθηκαν για να βελτιώσουν αυτές τις μεθόδους, και τον Ιανουάριο του 1988 με μια δεύτερη και αυστηρότερη διαδικασία επιλογής, κατέληξε ότι η πρόταση ADCT με βάση τον 8x8 DCT είχε παράγει την καλύτερη ποιότητα εικόνας.

Την χρονική στιγμή της επιλογής της, η DCT μέθοδος ήταν μόνο μερικώς καθορισμένη για μερικές από τις καταστάσεις λειτουργίας. Από το 1988 μέχρι το 1990, το JPEG ανέλαβε το αρκετά μεγάλο και σοβαρό έργο του ορισμού, της τεκμηρίωσης, της εξομοίωσης, της δοκιμής, της επικύρωσης, και απλά της συμφωνίας σε ένα μεγάλο αριθμό από λεπτομέρειες απαραίτητες για την πραγματική διαλειτουργικότητα και καθολικότητα.

3.1.3 Αρχιτεκτονική του προτύπου JPEG

Το προτεινόμενο πρότυπο περιέχει τις τέσσερις «καταστάσεις λειτουργίας» που προσδιορίστηκαν νωρίτερα (παρ. 3.1.2). Για κάθε κατάσταση, ορίζονται ένας ή

περισσότεροι ευδιάκριτοι codecs². Οι codecs που περιλαμβάνονται σε μια κατάσταση λειτουργίας διαφέρουν ανάλογα με την ακρίβεια των δειγμάτων της πηγαίας εικόνας που μπορούν να χειριστούν ή την μέθοδο κωδικοποίησης εντροπίας που χρησιμοποιούν. Αν και ο όρος codec χρησιμοποιείται συχνά στην βιβλιογραφία, δεν υπάρχει καμία απαίτηση ότι οι υλοποιήσεις πρέπει να περιλαμβάνουν και κωδικοποιητή και αποκωδικοποιητή. Πολλές εφαρμογές έχουν συστήματα ή συσκευές που απαιτούν μόνο τον ένα ή τον άλλο.

Οι τέσσερις καταστάσεις λειτουργίας και οι διάφοροι codecs τους, έχουν προκύψει από το στόχο του JPEG να είναι οικουμενικό και από την ποικιλομορφία των τύπων εικόνας στις εφαρμογές. Τα πολλά κομμάτια του JPEG μπορεί να δώσουν την εντύπωση μιας ανεπιθύμητης πολυπλοκότητας, αλλά πρέπει στην πραγματικότητα να θεωρηθούν ως ένα περιεκτικό «κουτί εργαλείων» που μπορεί να εκτείνεται σε ένα ευρύ φάσμα εφαρμογών εικόνας συνεχών τόνων.

Ο βασικός ακολουθιακός codec είναι εγγενώς μια πλούσια και περίπλοκη μέθοδος συμπίεσης που είναι ικανοποιητική για πολλές εφαρμογές. Εφαρμόζοντας κατάλληλα αυτήν την ελάχιστη μονάδα του JPEG, παρέχεται στη βιομηχανία μια σημαντική αρχική δυνατότητα για ανταλλαγή εικόνων.

3.2 Διακριτός μετασχηματισμός συνημιτόνου – DCT

3.2.1 Εισαγωγή

Ο διακριτός μετασχηματισμός συνημιτόνου, εκφράζει μία πεπερασμένη ακολουθία σημείων ως άθροισμα συνημιτονοειδών συναρτήσεων που ταλαντώνονται σε διαφορετικές συχνότητες. Τέτοιου είδους μετασχηματισμοί είναι σημαντικοί για μία πληθώρα επιστημονικών αλλά και μηχανικών εφαρμογών όπως α) η απωλεστική συμπίεση ήχου (MP3) και εικόνας (JPEG) όπου παραλείπονται οι συνιστώσες των υψηλών συχνοτήτων, β) οι φασματικές μέθοδοι για την αριθμητική επίλυση των μερικών διαφορικών εξισώσεων κλπ.

Η χρήση των συνημιτονοειδών συναρτήσεων, αντί των ημιτονοειδών, είναι υψίστης σημασίας για τη συμπίεση, καθώς απαιτούνται λιγότεροι υπολογισμοί για την προσέγγιση ενός τυπικού σήματος, ενώ στις διαφορικές εξισώσεις τα συνημίτονα εκφράζουν μία συγκεκριμένη επιλογή οριακών συνθηκών.

Η πιο κοινή παραλλαγή του διακριτού μετασχηματισμού συνημιτόνου είναι η DCT-II και καλείται απλώς DCT, ενώ η DCT-III είναι η αντιστροφή του DCT-II και καλείται αντίστροφος DCT ή IDCT. Είναι ο δεύτερος σε χρήση μετασχηματισμός, μετά το διακριτό μετασχηματισμό Fourier και η ευρεία χρήση του οφείλεται στην ιδιότητα του να συμπιέζει δεδομένα, η οποία υπερέχει σε σχέση με άλλους μετασχηματισμούς και προσεγγίζει εκείνη του βέλτιστου μετασχηματισμού

² Προέρχεται από τα αρχικά των λέξεων Coder/Decoder (κωδικοποιητής/αποκωδικοποιητής)

Karhunen – Loeve³, (KLT). Για το λόγο αυτό, οι πιο σημαντικές εφαρμογές του είναι η συμπίεση σημάτων ήχου και εικόνας.

Η εικόνα μπορεί να θεωρηθεί ως ένα δισδιάστατο σήμα, και οι μετασχηματισμοί που χρησιμοποιούνται καθώς και η όλη επεξεργασία μεταφέρονται στις δύο διαστάσεις. Για το λόγο αυτό, ο διακριτός μετασχηματισμός συνημιτόνου όταν χρησιμοποιείται στα συστήματα συμπίεσης ψηφιακών εικόνων παίρνει τη μορφή του δισδιάστατου (2D) μετασχηματισμού.

Όπως στην περίπτωση του μονοδιάστατου, ο δισδιάστατος είναι επίσης ασυμπτωτικά ισοδύναμος με το βέλτιστο μετασχηματισμό Karhunen – Loeve και είναι ο επόμενος καλύτερος στην ιεραρχία των μετασχηματισμών όσον αφορά στη στατιστική αποσυσχέτιση των δισδιάστατων σημάτων.

Σε μία ψηφιακή εικόνα υπάρχουν περιοχές με έντονες αλλαγές στην ένταση της φωτεινότητας, υπάρχουν όμως και περιοχές στις οποίες παρατηρούνται ελάχιστες αλλαγές, ή και καμία. Ο διακριτός μετασχηματισμός συνημιτόνου αναπαριστά μία εικόνα σαν ένα άθροισμα συνημιτονοειδών συναρτήσεων διαφορετικού πλάτους και συχνότητας η κάθε μία και έχει μία πολύ σημαντική ιδιότητα: το μεγαλύτερο μέρος της σημαντικής πληροφορίας βρίσκεται συγκεντρωμένο σε μερικούς από τους συντελεστές του, ενώ στους υπόλοιπους εμπεριέχεται ελάχιστη έως μηδαμινή πληροφορία και επομένως αυτοί μπορούν να παραληφθούν. Αυτή ακριβώς η ιδιότητά του είναι και ο λόγος για τον οποίο χρησιμοποιείται στη συμπίεση εικόνας, και πιο συγκεκριμένα είναι ο πυρήνας του προτύπου JPEG.

3.2.2 Ορισμός

Ο μετασχηματισμός συνημιτόνου δύο διαστάσεων ενός πίνακα $A_{M \times N}$ ορίζεται ως εξής:

$$B_{pq} = a_p a_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad 0 \leq p \leq M-1, \quad 0 \leq q \leq N-1$$

$$a_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad a_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases}$$

Εξίσωση 3.1 – DCT δύο διαστάσεων, πίνακα $A_{M \times N}$

Οι τιμές B_{pq} ονομάζονται συντελεστές DCT για τον πίνακα A_{mn} .

Ο DCT είναι αντιστρεπτός μετασχηματισμός και ο αντίστροφος δίνεται από τη σχέση:

³ Ο μετασχηματισμός Karhunen – Loeve δημιουργεί υποεικόνες με ασυσχέτιστα στοιχεία και για το λόγο αυτό επιτυγχάνει την καλύτερη συμπίεση από όλες τις άλλες κωδικοποιήσεις μετασχηματισμού.

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} a_p a_q B_{pq} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad 0 \leq m \leq M-1, \quad 0 \leq n \leq N-1$$

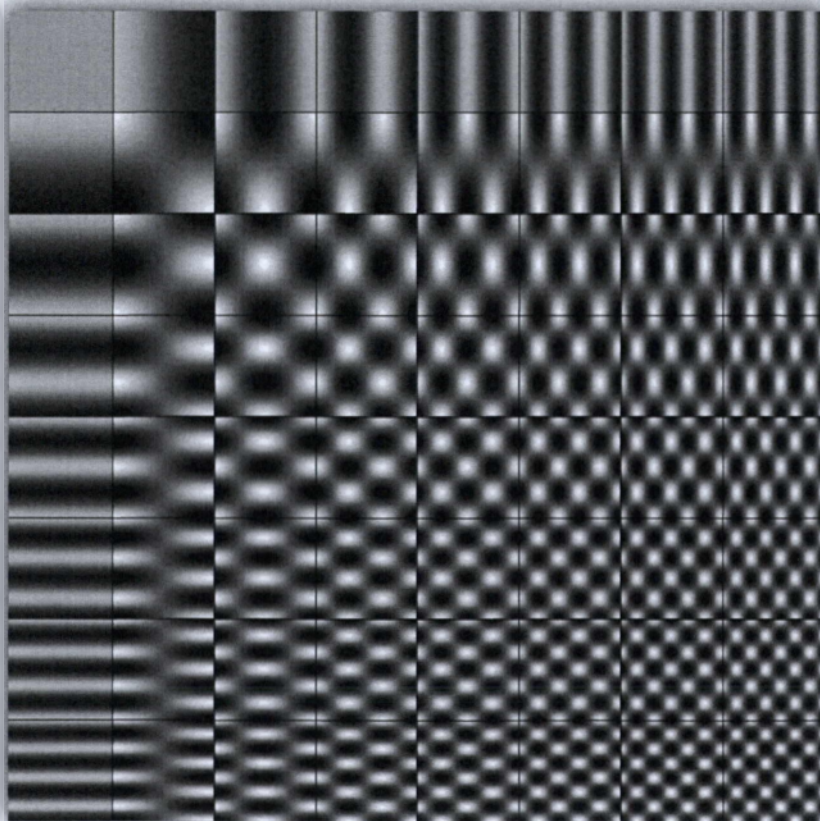
$$a_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad a_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases}$$

Από τη σχέση του αντίστροφου μετασχηματισμού συνημιτόνου, προκύπτει ότι οποιοσδήποτε πίνακας $A_{M \times N}$ μπορεί να γραφτεί ως άθροισμα $M \times N$ εξισώσεων της μορφής:

$$a_p a_q \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad 0 \leq p \leq M-1, \quad 0 \leq q \leq N-1$$

Οι συναρτήσεις αυτές ονομάζονται συναρτήσεις βάσης του DCT ενώ οι συντελεστές B_{pq} μπορούν να θεωρηθούν ως βάρη, τα οποία εφαρμόζονται σε κάθε συνάρτηση βάσης.

Συγκεκριμένα για πίνακες 8×8 , οι 64 συναρτήσεις βάσης απεικονίζονται παρακάτω, στην Εικόνα 3.1.



Εικόνα 3.1 – Οι συχνότητες των 64 συναρτήσεων βάσης ενός πίνακα $A_{8 \times 8}$

Οι οριζόντιες συχνότητες αυξάνονται από τα αριστερά προς τα δεξιά και οι κατακόρυφες αυξάνονται από πάνω προς τα κάτω. Η σταθερή συνάρτηση βάσης επάνω αριστερά συχνά ονομάζεται DC συνάρτηση βάσης, ενώ ο αντίστοιχος συντελεστής B_{00} ονομάζεται DC συνιστώσα.

3.3 Βήματα επεξεργασίας για την 8x8 DCT κωδικοποίηση

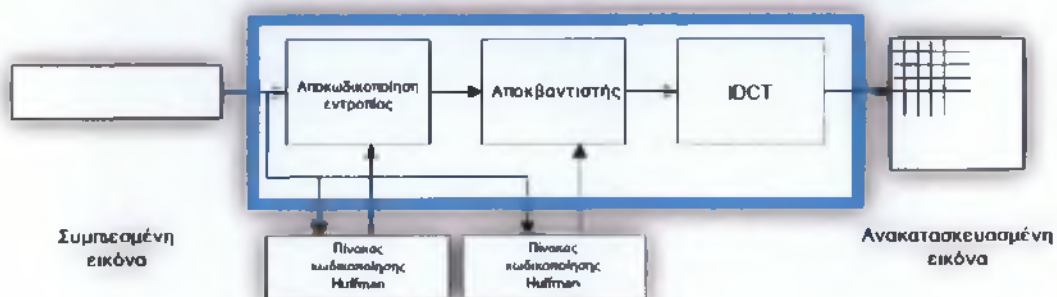
Παρακάτω παρουσιάζονται τα βασικά βήματα επεξεργασίας που αποτελούν την βάση των DCT μεθόδων λειτουργίας και εξηγείται η ειδική περίπτωση συμπίεσης grayscale εικόνας.

Η εικόνα χωρίζεται σε 8x8 τμήματα δειγμάτων (blocks), κάθε ένα από τα οποία εισάγεται στον κωδικοποιητή. Στη συνέχεια υπολογίζεται ξεχωριστά για κάθε τμήμα ο DCT δύο διαστάσεων, γίνεται η κβάντιση των συντελεστών του, η κωδικοποίηση και τέλος η μετάδοση (Σχήμα 3.1).



Σχήμα 3.1 - Βήματα επεξεργασίας DCT κωδικοποιητή

Κατά την αντίστροφη διαδικασία (Σχήμα 3.2), ο αποκωδικοποιητής αποκωδικοποιεί τους συντελεστές, υπολογίζει τον αντίστροφο DCT δύο διαστάσεων για κάθε τμήμα και στο τέλος ανασυνθέτει τα τμήματα για να σχηματίσει την αρχική εικόνα. Στις περισσότερες εικόνες, αρκετοί από τους DCT συντελεστές έχουν τιμή κοντά στο μηδέν και επομένως μπορούν να παραληφθούν, χωρίς να επηρεάζεται σημαντικά η ποιότητα της ανακατασκευασμένης εικόνας.



Σχήμα 3.2 - Βήματα επεξεργασίας DCT αποκωδικοποιητή

3.3.1 8x8 FDCT-IDCT

Στην εισαγωγή στον κωδικοποιητή, τα δείγματα της πηγαίας εικόνας είναι ομαδοποιημένα σε 8x8 τμήματα που μετατρέπονται από μη προσημασμένους ακεραίους αριθμούς με εύρος $[0, 2^P - 1]$ σε προσημασμένους ακεραίους αριθμούς με εύρος $[-2^P, 2^{P-1} - 1]$, και εισάγονται στον ευθύ μετασχηματισμό (FDCT).

Στην έξοδο από τον αποκωδικοποιητή, ο αντίστροφος μετασχηματισμός (IDCT) επιστρέφει 8x8 τμήματα δειγμάτων για να ανακατασκευάσει την εικόνα. Οι ακόλουθες εξισώσεις προκύπτουν από την Εξίσωση 3.1 και είναι εξιδανικευμένοι μαθηματικοί ορισμοί του NxN FDCT (Εξίσωση 3.2) και NxN IDCT (Εξίσωση 3.3):

$$F(u, v) = \frac{1}{\sqrt{2N}} C(u)C(v) \left[\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

Εξίσωση 3.2 – Γενική εξίσωση FDCT για πίνακες A_{NxN}

$$f(x, y) = \frac{1}{\sqrt{2N}} \left[\sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

Εξίσωση 3.3 – Γενική εξίσωση IDCT για πίνακες A_{NxN}

Οι παραπάνω εξισώσεις μπορούν να γραφούν πιο συγκεκριμένα για τον 8x8 FDCT και 8x8 IDCT μιας εικόνας, όπως φαίνεται στην Εξίσωση 3.4 και Εξίσωση 3.5 αντίστοιχα:

$$F(u, v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

Εξίσωση 3.4 - Εξίσωση FDCT για πίνακες A_{8x8}

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

Εξίσωση 3.5 - Εξίσωση IDCT για πίνακες A_{8x8}

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & u, v = 0 \\ 1, & u, v \neq 0 \end{cases}$$

Πιο απλά, ο διακριτός μετασχηματισμός συνημιτόνου είναι ένα γινόμενο

$$C = T * A * T'$$

όπου A είναι ένα 8×8 block της αρχικής εικόνας, T ο πίνακας μετασχηματισμού και T' ο ανάστροφος του πίνακα T . Ο πίνακας μετασχηματισμού για τον 8×8 DCT είναι ο παρακάτω:

$$T = \frac{1}{2} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \cos \frac{\pi}{16} & \cos \frac{3\pi}{16} & \cos \frac{5\pi}{16} & \cos \frac{7\pi}{16} & \cos \frac{9\pi}{16} & \cos \frac{11\pi}{16} & \cos \frac{13\pi}{16} & \cos \frac{15\pi}{16} \\ \cos \frac{2\pi}{16} & \cos \frac{6\pi}{16} & \cos \frac{10\pi}{16} & \cos \frac{14\pi}{16} & \cos \frac{18\pi}{16} & \cos \frac{22\pi}{16} & \cos \frac{26\pi}{16} & \cos \frac{30\pi}{16} \\ \cos \frac{3\pi}{16} & \cos \frac{9\pi}{16} & \cos \frac{15\pi}{16} & \cos \frac{21\pi}{16} & \cos \frac{27\pi}{16} & \cos \frac{33\pi}{16} & \cos \frac{39\pi}{16} & \cos \frac{45\pi}{16} \\ \cos \frac{4\pi}{16} & \cos \frac{12\pi}{16} & \cos \frac{20\pi}{16} & \cos \frac{28\pi}{16} & \cos \frac{36\pi}{16} & \cos \frac{44\pi}{16} & \cos \frac{52\pi}{16} & \cos \frac{60\pi}{16} \\ \cos \frac{5\pi}{16} & \cos \frac{15\pi}{16} & \cos \frac{25\pi}{16} & \cos \frac{35\pi}{16} & \cos \frac{45\pi}{16} & \cos \frac{55\pi}{16} & \cos \frac{65\pi}{16} & \cos \frac{75\pi}{16} \\ \cos \frac{6\pi}{16} & \cos \frac{18\pi}{16} & \cos \frac{30\pi}{16} & \cos \frac{42\pi}{16} & \cos \frac{54\pi}{16} & \cos \frac{66\pi}{16} & \cos \frac{78\pi}{16} & \cos \frac{90\pi}{16} \\ \cos \frac{7\pi}{16} & \cos \frac{21\pi}{16} & \cos \frac{35\pi}{16} & \cos \frac{49\pi}{16} & \cos \frac{63\pi}{16} & \cos \frac{77\pi}{16} & \cos \frac{91\pi}{16} & \cos \frac{105\pi}{16} \end{bmatrix}$$

Ο DCT σχετίζεται με τον διακριτό μετασχηματισμό Fourier (DFT). Μερικά αυθόρμητα συμπεράσματα για τη DCT συμπίεση, μπορούν να ληφθούν αν θεωρήσουμε τον FDCT ως έναν αναλυτή αρμονικών και τον IDCT ως έναν συνθέτη αρμονικών. Κάθε 8×8 block της πηγαίας εικόνας, είναι ουσιαστικά ένα διακριτό σήμα 64 σημείων το οποίο είναι συνάρτηση των δύο χωρικών διαστάσεων x και y . Ο FDCT παίρνει ένα τέτοιο σήμα ως είσοδο του και το αποσυνθέτει σε 64 ορθογώνια σήματα βάσης, κάθε ένα από τα οποία περιέχει μία από τις 64 μοναδικές δισδιάστατες (2D) χωρικές συχνότητες που περιλαμβάνει το φάσμα του σήματος εισαγωγής. Η έξοδος του FDCT είναι το σύνολο των πλατών των 64 σημάτων βάσης (ή αλλιώς συντελεστών DCT) των οποίων οι τιμές καθορίζονται μοναδικά από το διακριτό 64 σημείων σήμα στην είσοδο.

Οι τιμές των συντελεστών DCT μπορούν έτσι να θεωρηθούν ως σχετικό ποσό 2D χωρικών συχνοτήτων που περιλαμβάνονται στο σήμα εισαγωγής 64 σημείων. Ο συντελεστής μηδενικής συχνότητας και στις 2 διαστάσεις καλείται DC συντελεστής ενώ οι υπόλοιποι 63 συντελεστές ονομάζονται AC συντελεστές. Επειδή οι τιμές των δειγμάτων τυπικά αλλάζουν αργά από σημείο σε σημείο κατά μήκος της εικόνας, το βήμα επεξεργασίας FDCT θέτει τα θεμέλια για την επίτευξη της συμπίεσης πληροφορίας με τη συγκέντρωση του περισσότερου σήματος στις χαμηλότερες χωρικές συχνότητες. Έτσι για ένα τυπικό 8×8 block από μία τυπική πηγαία εικόνα, οι περισσότερες από τις χωρικές συχνότητες έχουν πλάτος μηδέν ή κοντά στο μηδέν, και δε χρειάζεται να κωδικοποιηθούν.

Στην αποκωδικοποίηση, ο IDCT αντιστρέφει αυτό το βήμα επεξεργασίας. Παίρνει τους 64 συντελεστές DCT (που σε εκείνο το σημείο έχουν κβαντιστεί) και ανακατασκευάζει ένα σήμα εξόδου 64 σημείων, αθροίζοντας τα βήματα βάσης. Από μαθηματική άποψη, ο DCT είναι μία 1:1 χαρτογράφηση για διανύσματα 64 σημείων μεταξύ του πεδίου της συχνότητας και της εικόνας. Εάν ο FDCT και ο IDCT

μπορούσαν να υπολογιστούν με τέλεια ακρίβεια και εάν οι συντελεστές DCT δεν ήταν κβαντισμένοι όπως στην ακόλουθη περιγραφή, το αρχικό σήμα 64 σημείων θα μπορούσε να ανακτηθεί πλήρως. Θεωρητικά ο DCT δεν εισάγει καμία απώλεια στα δείγματα της πηγαίας εικόνας αλλά τα μετασχηματίζει μόνο σε ένα πεδίο, στο οποίο μπορούν να κωδικοποιηθούν αποτελεσματικότερα.

Ακόμη και αν ληφθεί υπόψη η πεπερασμένη ακρίβεια των εισόδων DCT και των αποτελεσμάτων, ανεξάρτητα σχεδιασμένες εφαρμογές του ίδιου αλγορίθμου FDCT ή IDCT διαφέρουν στην ακρίβεια με την οποία αναπαρίστανται οι όροι συνημιτόνου ή τα ενδιάμεσα αποτελέσματα. Διαφέρουν επίσης και στον τρόπο που αυτές αθροίζουν και στρογγυλοποιούν κλασματικές τιμές, και τελικά παράγουν ελαφρώς διαφορετικά αποτελέσματα με την ίδια είσοδο.

Για να διατηρηθεί η δυνατότητα για καινοτομία και προσαρμογή μέσα στις υλοποιήσεις, έχει επιλεγεί να μη διευκρινιστεί ένας μοναδικός αλγόριθμος για τους FDCT και IDCT στα προτεινόμενα πρότυπα του JPEG. Αυτό κάνει δυσκολότερη την επιβεβαίωση της συμβατότητας με το πρότυπο, επειδή δύο συμβατοί κωδικοποιητές/αποκωδικοποιητές δε θα παράγουν (γενικά) τα ίδια αποτελέσματα για δεδομένη ίδια είσοδο. Το πρότυπο JPEG αντιμετωπίζει αυτό το ζήτημα με τη διενέργεια μίας δοκιμής ακριβείας, ως τμήμα των δοκιμών συμβατότητας για όλους τους DCT κωδικοποιητές και αποκωδικοποιητές. Αυτό εξασφαλίζει ότι δε θα υπάρξουν υλοποιήσεις που θα υποβίβαζαν την ποιότητα της εικόνας.

Για κάθε DCT κατάσταση λειτουργίας, το JPEG διευκρινίζει χωριστούς κωδικοποιητές για εικόνες με 8 bit και με 12 bit ανά συνιστώσα. Οι κωδικοποιητές 12 bit, που πρέπει να καλύψουν ορισμένους τύπους ιατρικών και άλλων εικόνων, απαιτούν μεγαλύτερους υπολογιστικούς πόρους για να επιταχύνουν την απαιτούμενη FDCT ή IDCT ακρίβεια. Εικόνες με άλλη ακρίβεια δείγματος μπορούν συνήθως να καλυφθούν από κωδικοποιητές είτε των 8 bit είτε των 12 bit, αλλά αυτό πρέπει να γίνεται εκτός του προτύπου JPEG. Για παράδειγμα, θα ήταν ευθύνη της εκάστοτε εφαρμογής να αποφασίσει πως θα εισάγει ένα δείγμα 6 bit στον κωδικοποιητή 8 bit, πως θα το ανοίξει στην έξοδο του αποκωδικοποιητή και πως θα κωδικοποιήσει οποιεσδήποτε απαραίτητες σχετικές πληροφορίες.

3.3.2 Κβάντιση

Μετά από την έξοδο από τον FDCT, κάθε ένας από τους 64 DCT συντελεστές κβαντίζονται ομοιόμορφα από κοινού με έναν πίνακα κβάντισης 64-στοιχείων (Quantization Table), ο οποίος πρέπει να είναι ορισμένος από την εφαρμογή (ή το χρήστη) σαν είσοδος στον κωδικοποιητή. Κάθε στοιχείο του μπορεί να έχει οποιαδήποτε ακέραια τιμή από 0 έως 255, γεγονός που ορίζει το μέγεθος βήματος του κβαντιστή για τον αντίστοιχο DCT συντελεστή του. Ο σκοπός της κβάντισης είναι να επιτύχει την περαιτέρω συμπίεση αναπαριστώντας τους DCT συντελεστές με ακρίβεια όχι μεγαλύτερη από αυτή που είναι απαραίτητη για να επιτευχθεί η επιθυμητή ποιότητα εικόνας. Με άλλα λόγια, ο στόχος αυτού του βήματος επεξεργασίας είναι να απορριφθούν οι πληροφορίες που δεν είναι οπτικά σημαντικές.

Η κβάντιση είναι μια χαρτογράφηση πολλών σε ένα, και επομένως αξιωματικά περιλαμβάνει απώλειες. Είναι μάλιστα η κύρια πηγή απωλειών στους DCT κωδικοποιητές.

Η κβάντιση ορίζεται ως το πηλίκο κάθε συντελεστή DCT με το αντίστοιχο μέγεθος βήματος του κβαντιστή, ακολουθούμενο από την στρογγυλοποίηση στον κοντινότερο ακέραιο αριθμό:

$$F^Q(u, v) = \text{Integer Round} \left(\frac{F(u, v)}{Q(u, v)} \right)$$

Αυτή η τιμή στην έξοδο κανονικοποιείται με το μέγεθος του βήματος του κβαντιστή. Η αποκβάντιση είναι η αντίστροφη λειτουργία, η οποία σε αυτή την περίπτωση σημαίνει απλά ότι η κανονικοποίηση αναιρείται πολλαπλασιάζοντας με το μέγεθος του βήματος, επιστρέφοντας το αποτέλεσμα σε μια αναπαράσταση κατάλληλη για την εισαγωγή στον IDCT:

$$F^Q(u, v) = F^Q(u, v) * Q(u, v)$$

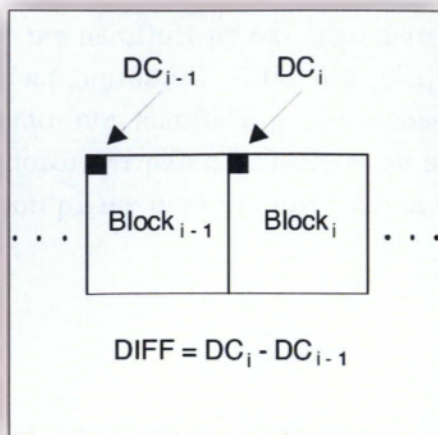
Όταν ο στόχος είναι να συμπεστεί η εικόνα όσο το δυνατόν περισσότερο χωρίς ορατά σημάδια, κάθε μέγεθος βήματος (ιδανικά) πρέπει να επιλεγεί ως το αντιληπτικά κατώτατο όριο (ή αλλιώς η «ακριβώς παρατηρίσιμη διαφορά») για την οπτική συμβολή της αντίστοιχης βάσης συνημιτόνου. Αυτά τα κατώφλια είναι επίσης συνάρτηση των χαρακτηριστικών της πηγαίας εικόνας, των χαρακτηριστικών της οθόνης και της απόστασης παρατήρησης. Για εφαρμογές στις οποίες αυτές οι μεταβλητές μπορούν να είναι εύλογα και καλά καθορισμένες, μπορούν να διενεργηθούν ψυχο-οπτικά (psycho-optical) πειράματα για να καθοριστούν τα καλύτερα κατώφλια. Πειράματα που περιγράφονται στην διεθνή βιβλιογραφία έχουν οδηγήσει σε ένα σύνολο πινάκων κβάντισης που περιλαμβάνονται στο πρότυπο ISO, όχι όμως σαν απαίτηση.

3.3.3 DC Κωδικοποίηση & Ακολουθία Zig-Zag

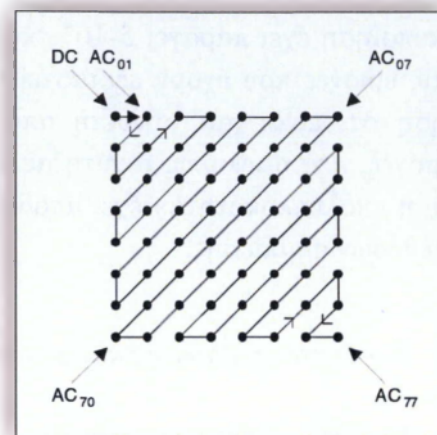
Μετά από την κβάντιση, ο DC συντελεστής αντιμετωπίζεται χωριστά από τους 63 συντελεστές AC. Ο DC συντελεστής είναι ένα μέτρο της μέσης τιμής των 64 δειγμάτων της υποεικόνας. Επειδή υπάρχει συνήθως ισχυρή συσχέτιση μεταξύ των DC συντελεστών των γειτονικών 8x8 blocks, ο κβαντισμένος DC συντελεστής κωδικοποιείται ως η διαφορά του από τον DC όρο του προηγούμενου block στην σειρά κωδικοποίησης (που καθορίζεται στην συνέχεια), όπως παρουσιάζεται στο Σχήμα 3.3.

Τέλος, όλοι οι κβαντισμένοι συντελεστές διατάσσονται σε μια zig-zag ακολουθία, όπως παρουσιάζεται στο Σχήμα 3.4. Αυτή η διάταξη βοηθά στην υλοποίηση της κωδικοποίησης εντροπίας τοποθετώντας τούς χαμηλής συχνότητας

συντελεστές (που είναι πιθανότατα διαφορετικοί από το μηδέν) πριν από τους συντελεστές υψηλής συχνότητας.



Σχήμα 3.3 - Διαφορική DC κωδικοποίηση



Σχήμα 3.4 - Zig-Zag ακολουθία

3.3.4 Κωδικοποίηση εντροπίας

Το τελικό βήμα επεξεργασίας του DCT κωδικοποιητή είναι η κωδικοποίηση εντροπίας. Αυτό το βήμα επιτυγχάνει πρόσθετη συμπίεση χωρίς απώλειες με την κωδικοποίηση των κβαντισμένων DCT συντελεστών, συμπαγώς βασισμένο στα στατιστικά τους χαρακτηριστικά. Το πρότυπο JPEG ορίζει δύο μεθόδους κωδικοποίησης εντροπίας: την κωδικοποίηση Huffman και την αριθμητική κωδικοποίηση. Ο βασικός codec χρησιμοποιεί την κωδικοποίηση Huffman, αλλά ορίζονται codecs και με τις δύο μεθόδους για όλες τις καταστάσεις λειτουργίας.

Είναι χρήσιμο να θεωρηθεί η κωδικοποίηση εντροπίας ως διαδικασία 2 βημάτων. Το πρώτο βήμα μετατρέπει την zig-zag ακολουθία κβαντισμένων συντελεστών σε μια ενδιάμεση ακολουθία συμβόλων, ενώ το δεύτερο μετατρέπει τα σύμβολα σε μια ακολουθία στοιχείων στην οποία τα σύμβολα δεν έχουν πλέον εξωτερικά προσδιορίσιμα όρια. Η μορφή και ο καθορισμός των ενδιάμεσων συμβόλων εξαρτάται και από την DCT κατάσταση λειτουργίας και από την μέθοδο κωδικοποίησης εντροπίας.

Η κωδικοποίηση Huffman απαιτεί ένα ή περισσότερα σύνολα πινάκων κωδικοποίησης Huffman να ορίζονται από την εφαρμογή. Οι ίδιοι πίνακες που χρησιμοποιούνται για να συμπιεστεί μια εικόνα απαιτούνται και για να αποσυμπιεστεί. Οι πίνακες Huffman μπορούν να προκαθοριστούν και να χρησιμοποιούνται μέσα σε μια εφαρμογή ως προεπιλεγμένοι, ή να υπολογίζονται ειδικά για κάθε δεδομένη εικόνα σε ένα αρχικό πέρασμα συλλογής στατιστικών στοιχείων πριν από τη συμπίεση. Τέτοιες επιλογές είναι καθαρά θέμα των εφαρμογών που χρησιμοποιούν το JPEG καθώς το ίδιο το πρότυπο JPEG δεν ορίζει συγκεκριμένους πίνακες Huffman.

Αντιθέτως, η συγκεκριμένη αριθμητική μέθοδος κωδικοποίησης που ορίζεται στο πρότυπο JPEG, δεν απαιτεί την εισαγωγή κάποιου πίνακα επειδή είναι σε θέση να

προσαρμόσκει στα στατιστικά χαρακτηριστικά της εικόνας ενώ κωδικοποιεί την ίδια την εικόνα. Ωστόσο μπορούν να χρησιμοποιηθούν και εξωτερικοί πίνακες για ελαφρώς καλύτερη απόδοση, χωρίς ωστόσο αυτό να είναι απαραίτητο. Η αριθμητική κωδικοποίηση έχει παράγει 5-10% καλύτερη συμπίεση από τη Huffman για πολλές από τις εικόνες που έχουν εξεταστεί από τα μέλη του JPEG. Εντούτοις, υπάρχει η αίσθηση ότι είναι πιο σύνθετη από την κωδικοποίηση Huffman για ορισμένες εφαρμογές, π.χ. η υψηλής ταχύτητας hardware υλοποιήσεις. Σε όλη την ιστορία του JPEG, η «πολυπλοκότητα» έχει αποδειχθεί το πιο αόριστο κριτήριο για τη σύγκριση των μεθόδων συμπίεσης.

3.3.5 Ποιότητα συμπίεσης και εικόνων

Για τις έγχρωμες εικόνες με μέτρια σύνθετες σκηνές, όλες οι DCT καταστάσεις λειτουργίας παράγουν τυπικά τα ακόλουθα επίπεδα ποιότητας εικόνας για το αναφερόμενο εύρος συμπίεσης. Αυτά τα επίπεδα είναι μόνο ένας οδηγός, καθώς η ποιότητα και η συμπίεση μπορούν να ποικίλουν σημαντικά ανάλογα με τα χαρακτηριστικά της πηγαίας εικόνας και το περιεχόμενο των σκηνών. Οι μονάδες bits/pixel αναφέρονται στον συνολικό αριθμό bits στη συμπίεσμένη εικόνα, συμπεριλαμβανομένων των τμημάτων χρωμικότητας (chrominance), που διαιρούνται με τον αριθμό των δειγμάτων στο τμήμα φωτεινότητας.

bits/pixel	Ποιότητα	Ικανοποίηση
0,25 - 0,50	μέτρια έως καλή	για κάποιες εφαρμογές
0.50 - 0.75	καλή έως πολύ καλή ποιότητα	για πολλές εφαρμογές
0,75 - 1,50	άριστη ποιότητα	για τις περισσότερες εφαρμογές
1,50 - 2,00	συνήθως δεν ξεχωρίζει από την αρχική	για απαιτητικές εφαρμογές

Πίνακας 3.1 - Επίπεδα συμπίεσης και ποιότητα εικόνας

Στην Εικόνα 3.2 φαίνεται η εφαρμογή του DCT δύο διαστάσεων σε μία ασπρόμαυρη εικόνα, για τμήματα διαστάσεων 8x8. Σε κάθε τμήμα διατηρούνται μόνο οι πρώτοι 10 από τους 64 συντελεστές DCT ενώ οι υπόλοιποι μηδενίζονται. Στη συνέχεια γίνεται ανακατασκευή της εικόνας με εφαρμογή του αντίστροφου DCT για κάθε τμήμα. Παρά το γεγονός ότι η ποιότητα στην ανακατασκευασμένη εικόνα είναι αρκετά υποβαθμισμένη, παραμένει αναγνωρίσιμη παρόλο που παραλείφθηκε σχεδόν το 85% των συντελεστών DCT.



(πριν την εφαρμογή DCT)



(μετά την εφαρμογή DCT)

Εικόνα 3.2 – Εφαρμογή του DCT δύο διαστάσεων σε ασπρόμαυρη εικόνα



4

Επεξεργασία εικόνας και εφαρμογή του DCT - Παραδείγματα με το MATLAB

4.1 Εισαγωγή στην επεξεργασία εικόνας

Μία εικόνα μπορεί να οριστεί ως μία συνάρτηση δύο διαστάσεων $f(x, y)$ όπου x και y είναι οι συντεταγμένες της, και η τιμή της f για κάθε ζεύγος συντεταγμένων (x, y) καλείται ως φωτεινότητα της εικόνας στο σημείο αυτό. Μία αναλογική εικόνα έχει συνεχή πεδία τιμών x, y και συνεχείς τιμές φωτεινότητας. Για την ψηφιοποίηση μίας αναλογικής εικόνας, με άμεσο σκοπό την επεξεργασία της με τη βοήθεια ενός ηλεκτρονικού υπολογιστή ή απλώς την αποθήκευση της σε ψηφιακή μορφή, απαιτείται αρχικά η μετατροπή των συντεταγμένων σε διακριτές τιμές (δειγματοληψία – sampling) και των συνεχών τιμών της φωτεινότητας σε διακριτές (κβάντιση – quantization). Με τον τρόπο αυτό προκύπτει μία ψηφιακή εικόνα.

Για παράδειγμα, μία 8-bit grayscale εικόνα διαστάσεων $M \times N$ pixels, θα απεικονίζεται στο MATLAB με έναν δισδιάστατο πίνακα $M \times N$ στοιχείων (Εικόνα 4.1). Λόγω της κβάντισης των επιπέδων του γκρι σε L επίπεδα, κάθε στοιχείο $f(x, y)$ θα ορίζεται στο διάστημα $0 \leq f(x, y) \leq L - 1$. Η τιμή 0 αντιστοιχεί στο μαύρο χρώμα, ενώ η τιμή $L - 1$ στο άσπρο χρώμα. Τυπικά απαιτούνται $M \times N \times k$ bits για την αποθήκευση της δεδομένης εικόνας και επειδή $k = 8$, κάθε στοιχείο $f(x, y)$ θα ορίζεται στο διάστημα $0 \leq f(x, y) \leq 255$.



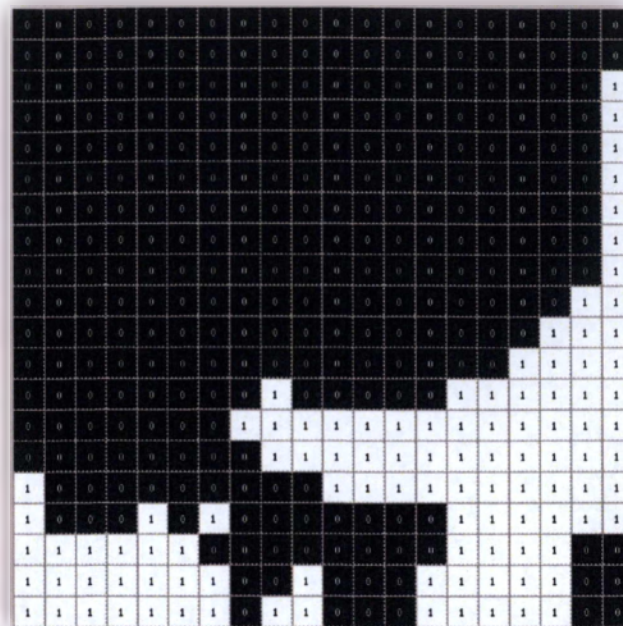
Εικόνα 4.1 – Πίνακας απεικόνισης στοιχείων $M \times N$

Η επεξεργασία μιας εικόνας έχει την έννοια της αλλαγής των χαρακτηριστικών της, είτε για να βελτιώσουμε την πληροφορία της ώστε να φαίνεται καλύτερη στο ανθρώπινο μάτι, είτε για να καταστεί καταλληλότερη για την επεξεργασία της από ηλεκτρονικό υπολογιστή. Οι άνθρωποι προτιμούν τις εικόνες ευκρινείς, καθαρές και λεπτομερείς, σε αντίθεση με τους υπολογιστές που προτιμούν τις εικόνες απλές και απέριτες.

4.2 Είδη ψηφιακών εικόνων

Όπως αναφέρθηκε νωρίτερα, το δομικό στοιχείο μίας ψηφιακής εικόνας είναι το pixel. Ανάλογα με το βάθος χρώματος της εικόνας, κάθε pixel μπορεί να αποτελείται από μία ή περισσότερες συνιστώσες οι οποίες αντιπροσωπεύουν την τιμή έντασης της φωτεινότητας του χρώματος του.

Στις ασπρόμαυρες εικόνες (2-bit binary), κάθε pixel μπορεί να πάρει διακριτές τιμές 0 ή 1 (Εικόνα 4.2). Έτσι χρειάζεται μόνο 1bit/pixel για την αποθήκευση της πληροφορίας.



Εικόνα 4.2 - Διακριτές τιμές φωτεινότητας των pixels μιας ασπρόμαυρης εικόνας

Στις εικόνες με διαβαθμίσεις του γκρι (8-bit grayscale), κάθε εικονοστοιχείο παίρνει μία τιμή η οποία κυμαίνεται από 0 (μαύρο) μέχρι 255 (άσπρο) (Εικόνα 4.3) ενώ στις έγχρωμες (true color, RGB) κάθε pixel δημιουργείται με σύνθεση τριών βασικών χρωμάτων: κόκκινο, πράσινο, μπλε (Red, Green, Blue) και έχει μία τιμή η οποία περιγράφεται με βάση το επιμέρους ποσοστό φωτεινότητας του κάθε χρώματος που εμπεριέχεται σε αυτό (Εικόνα 4.4). Κάθε ένα από τα βασικά χρώματα μπορεί να πάρει διακριτές τιμές από 0-255 και κατά συνέπεια μία έγχρωμη εικόνα μπορεί να αναπαρασταθεί από $256^3 = 16.777.216$ πιθανά χρώματα.

42	32	27	22	33	25	28	26	28	23	33	38	47	73
38	34	36	26	22	22	26	26	28	30	67	51	99	117
27	33	30	27	24	26	29	24	33	40	56	82	139	168
23	26	31	32	26	27	28	43	48	55	87	127	169	205
28	46	44	26	19	14	19	77	77	50	76	119	184	217
45	68	68	30	16	14	17	38	81	51	33	58	170	220
41	86	85	64	28	20	22	53	109	65	48	51	163	220
25	84	91	81	58	51	52	81	102	62	26	77	187	223
53	41	74	91	75	74	82	73	59	42	51	126	210	226
79	42	44	69	76	80	69	64	39	50	110	182	222	227
109	80	46	40	37	44	41	49	66	103	166	212	224	223
119	111	89	71	49	51	62	80	127	164	200	218	219	219
125	134	119	105	96	97	115	138	170	198	212	214	218	221
133	144	145	134	134	141	150	163	194	201	207	211	214	219

Εικόνα 4.3 – Διακριτές τιμές φωτεινότητας των pixels μιας grayscale εικόνας

R: 37 G: 17 B: 13	R: 46 G: 33 B: 28	R: 46 G: 25 B: 24	R: 45 G: 26 B: 28	R: 27 G: 16 B: 20	R: 31 G: 26 B: 22	R: 41 G: 23 B: 22	R: 65 G: 60 B: 70	R: 62 G: 40 B: 54	R: 76 G: 45 B: 49
R: 44 G: 22 B: 14	R: 54 G: 45 B: 26	R: 62 G: 37 B: 31	R: 35 G: 22 B: 19	R: 21 G: 16 B: 26	R: 17 G: 13 B: 8	R: 23 G: 20 B: 3	R: 74 G: 77 B: 83	R: 81 G: 73 B: 91	R: 68 G: 42 B: 48
R: 55 G: 41 B: 35	R: 79 G: 65 B: 56	R: 80 G: 61 B: 70	R: 41 G: 25 B: 24	R: 21 G: 14 B: 13	R: 15 G: 13 B: 14	R: 18 G: 19 B: 4	R: 43 G: 37 B: 26	R: 88 G: 78 B: 79	R: 66 G: 42 B: 59
R: 47 G: 40 B: 28	R:100 G: 79 B: 88	R: 91 G: 80 B: 91	R: 76 G: 58 B: 62	R: 35 G: 23 B: 34	R: 20 G: 19 B: 24	R: 24 G: 22 B: 13	R: 64 G: 50 B: 38	R:106 G:106 B:135	R: 80 G: 57 B: 71
R: 44 G: 33 B: 19	R: 91 G: 80 B: 83	R: 96 G: 88 B: 94	R: 93 G: 73 B: 89	R: 74 G: 51 B: 56	R: 69 G: 44 B: 40	R: 61 G: 48 B: 48	R: 88 G: 79 B: 75	R:102 G:106 B: 91	R: 83 G: 50 B: 71
R: 68 G: 47 B: 42	R: 55 G: 36 B: 34	R: 80 G: 70 B: 76	R:100 G: 84 B:101	R: 91 G: 68 B: 66	R: 89 G: 69 B: 61	R: 88 G: 80 B: 79	R: 88 G: 67 B: 68	R: 81 G: 49 B: 51	R: 64 G: 31 B: 42
R: 96 G: 70 B: 83	R: 48 G: 41 B: 34	R: 48 G: 43 B: 38	R: 79 G: 64 B: 56	R: 87 G: 72 B: 65	R: 91 G: 75 B: 81	R: 88 G: 61 B: 59	R: 72 G: 58 B: 70	R: 47 G: 33 B: 46	R: 59 G: 45 B: 51
R:124 G:101 B:114	R: 91 G: 74 B: 81	R: 54 G: 44 B: 37	R: 50 G: 40 B: 14	R: 44 G: 35 B: 28	R: 52 G: 42 B: 34	R: 51 G: 36 B: 40	R: 50 G: 45 B: 65	R: 70 G: 61 B: 81	R:104 G: 96 B:140
R:142 G:111 B:101	R:123 G:102 B:122	R: 99 G: 80 B:105	R: 83 G: 64 B: 79	R: 64 G: 40 B: 57	R: 66 G: 42 B: 57	R: 70 G: 55 B: 81	R: 91 G: 71 B: 97	R:123 G:120 B:173	R:142 G:164 B:222
R:162 G:109 B:110	R:154 G:122 B:139	R:135 G:110 B:125	R:126 G: 91 B:123	R:110 G: 83 B:123	R:108 G: 84 B:131	R:114 G:107 B:155	R:130 G:136 B:173	R:150 G:172 B:209	R:183 G:199 B:231

Εικόνα 4.4 – Διακριτές τιμές φωτεινότητας των pixels μιας έγχρωμης εικόνας

4.3 Επεξεργασία εικόνας στο MATLAB

Προκειμένου να επεξεργαστούμε μία εικόνα στο MATLAB θα χρειαστεί πρώτα να τη φορτώσουμε. Υπάρχει δυνατότητα φόρτωσης και έγχρωμων και ασπρόμαυρων εικόνων. Αν η εικόνα είναι έγχρωμη αλλά το χρώμα δεν είναι σημαντικό, μπορούμε να την μετατρέψουμε σε grayscale προκειμένου να μειωθεί ο όγκος της πληροφορίας προς επεξεργασία.

Για παράδειγμα το χρώμα δε μπορεί να είναι σημαντικό σε μία εικόνα όταν προσπαθούμε να αντιμετωπίσουμε ένα συγκεκριμένο αντικείμενο που έχει καλή αντίθεση με το περιβάλλον του. Στο παρακάτω παράδειγμα περιγράφεται η διαδικασία φόρτωσης διαφορετικών εικόνων στο MATLAB.

Οι εντολές που θα χρησιμοποιηθούν είναι οι εξής:

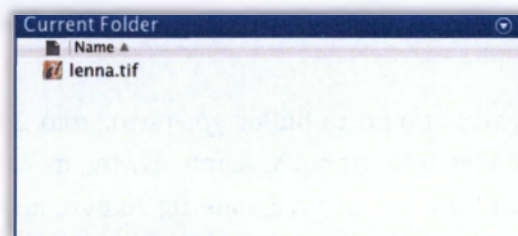
- imread:** Χρησιμοποιείται για να διαβάσει ένα αρχείο εικόνας σε μία συγκεκριμένη μορφή. Γενικά η εντολή imread διαβάζει τις τιμές των εικονοστοιχείων της εικόνας και επιστρέφει έναν πίνακα με όλες αυτές τις τιμές. Επομένως έχοντας διαβάσει την εικόνα και έχοντας τη σε έναν πίνακα τιμών στο Matlab, μπορεί να εφαρμοστεί οποιαδήποτε λειτουργία.
- imshow:** Ανασυνθέτει την εικόνα από έναν πίνακα τιμών και την εμφανίζει.
- rgb2grey:** Αλλάζει μία εικόνα από έγχρωμη σε ασπρόμαυρη.
- im2bw:** Αλλάζει μία εικόνα από έγχρωμη ή ασπρόμαυρη σε μονόχρωμη.

4.3.1 Εισαγωγή εικόνας στο MATLAB

Η βασική δομή δεδομένων του MATLAB είναι οι πίνακες, ένα διατεταγμένο σύνολο πραγματικών ή σύνθετων στοιχείων. Ένας πίνακας αποτελεί τη καταλληλότερη δομή δεδομένων για την αναπαράσταση μίας εικόνας ως διατεταγμένου συνόλου πραγματικών τιμών χρώματος ή έντασης φωτεινότητας χρώματος.

Στην περίπτωση των έγχρωμων εικόνων, απαιτείται ένας τρισδιάστατος πίνακας x,y,z όπου ο άξονας x αντιπροσωπεύει την ένταση του κόκκινου χρώματος, ο y του πράσινου και ο z του μπλε. Αντίθετα, στην περίπτωση ασπρόμαυρων εικόνων (είτε grayscale είτε binary), απαιτείται ένας δισδιάστατος πίνακας, κάθε στοιχείο του οποίου αντιπροσωπεύει την ένταση φωτεινότητας του λευκού.

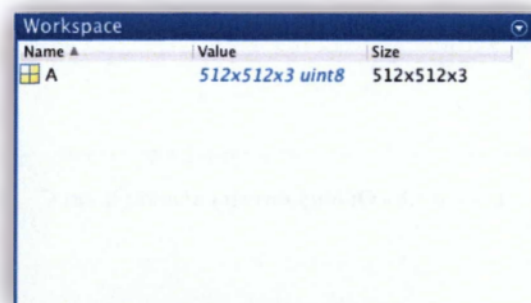
Έστω ότι στον τρέχοντα φάκελο εργασίας του MATLAB, υπάρχει το παρακάτω αρχείο εικόνας, ανάλυσης 512x512 pixels (Εικόνα 4.5). Πληκτρολογούμε:



Εικόνα 4.5 - Παράθυρο τρέχοντα φακέλου

```
A=imread('lenna.tif');  
figure(1);imshow(A)
```

Ο παραπάνω κώδικας διαβάζει την αρχική εικόνα βάθους χρώματος 24 bits, αποθηκεύει τις τιμές φωτεινότητας σε έναν τρισδιάστατο πίνακα A (Εικόνα 4.6) και στη συνέχεια ανακατασκευάζει την εικόνα που προκύπτει από τις τιμές που έχουν αποθηκευτεί στον πίνακα αυτόν (Εικόνα 4.7).



Εικόνα 4.6 - Τρισδιάστατος πίνακας 512x512x3

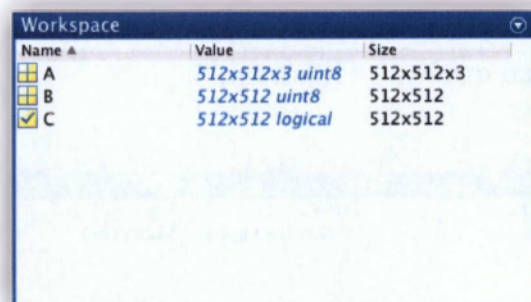


Εικόνα 4.7 – Εικόνα που προκύπτει από τον πίνακα A

Στη συνέχεια μετατρέπουμε το βάθος χρώματος από 24-bit σε 8-bit grayscale και 2-bits (binary), αποθηκεύουμε τις τιμές φωτεινότητας σε δισδιάστατους πίνακες B και C αντίστοιχα (Εικόνα 4.8), και εμφανίζουμε τις εικόνες που προκύπτουν από τους πίνακες αυτούς (Εικόνα 4.9 και Εικόνα 4.10 αντίστοιχα). Πληκτρολογούμε:

```
B=rgb2gray(A);  
figure(2);imshow(B);
```

```
C=im2bw(A);  
figure(3);imshow(C);
```



Name	Value	Size
A	512x512x3 uint8	512x512x3
B	512x512 uint8	512x512
C	512x512 logical	512x512

Εικόνα 4.8 - Οι δισδιάστατοι πίνακες B και C



Εικόνα 4.9 - Εικόνα που προκύπτει
από τον πίνακα B



Εικόνα 4.10 - Εικόνα που προκύπτει
από τον πίνακα C

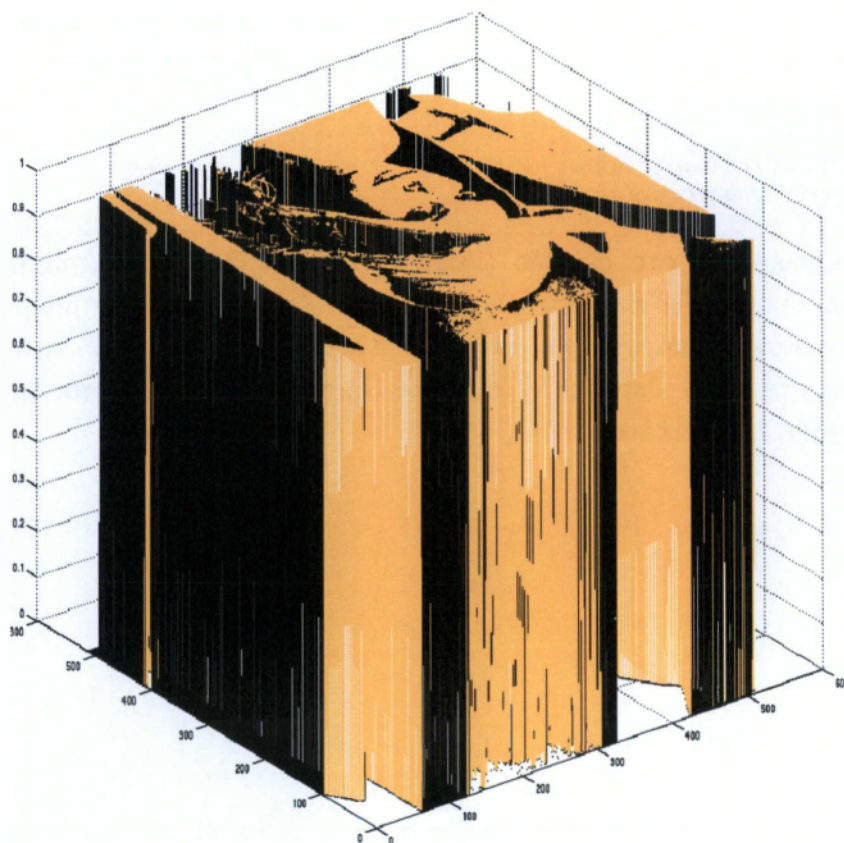
Προκειμένου να κατανοήσουμε τον τρόπο με τον οποίο αναπαρίσταται μία εικόνα με βάση τις τιμές φωτεινότητας κάθε pixel, αρκεί να τη θεωρήσουμε ως ένα σύνολο καρτεσιανών συντεταγμένων, όπου οι άξονες x, y αντιπροσωπεύουν τη θέση κάθε pixel, ενώ ο άξονας z την ένταση της φωτεινότητας κάθε χρώματος για το ίδιο pixel (αυτήν ακριβώς τη μορφή έχουν και οι πίνακες A, B, C που περιέχουν τα δεδομένα του παραδείγματος). Ο κώδικας που ακολουθεί εμφανίζει μία γραφική απεικόνιση των τιμών της έντασης στο επίπεδο και στο χώρο. Προκειμένου να απλουστεύσουμε τη διαδικασία, θα χρησιμοποιήσουμε τον πίνακα C που περιέχει τις τιμές φωτεινότητας για βάθος χρώματος 2-bit, αφού πρώτα μετατρέψουμε τις τιμές του από δυαδικές, σε πραγματικές διπλής ακρίβειας. Πληκτρολογούμε:

```
D=double(C);
imagesc(D), colormap(copper(256));
mesh(D), colormap(copper(256));
```

Η απεικόνιση στο επίπεδο (Εικόνα 4.11) απλώς αναπαράγει την ίδια την εικόνα όπου οι περιοχές που δεν έχουν καθόλου φωτεινότητα είναι μαύρες, ενώ στις φωτεινές περιοχές υπάρχει χρωματισμός. Ωστόσο βοηθά στην κατανόηση της απεικόνισης στο χώρο, όπου φαίνεται πλέον ότι οι σκοτεινές περιοχές παίρνουν τιμή 0 ενώ οι φωτεινές 1 (Εικόνα 4.12).



Εικόνα 4.11 – Απεικόνιση τιμών στο επίπεδο

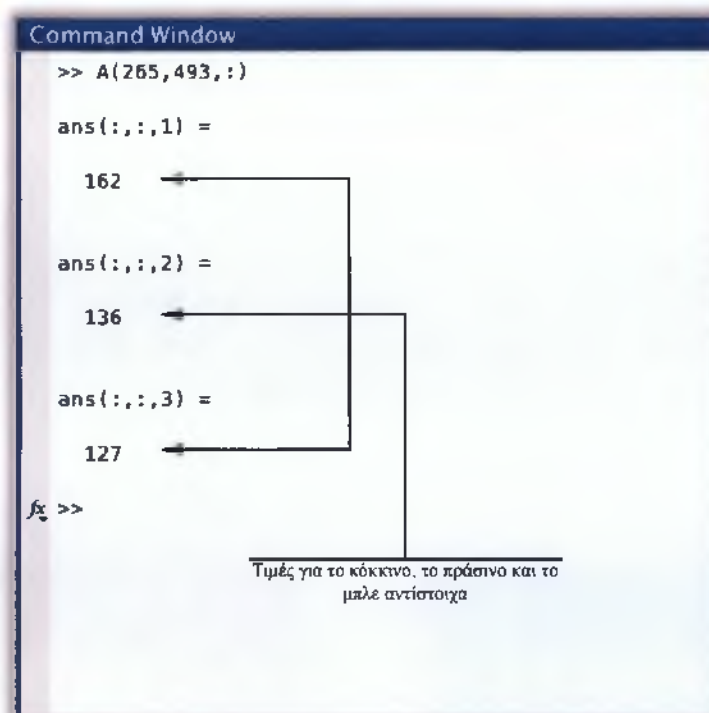


Εικόνα 4.12 – Απεικόνιση τιμών στο χώρο

Πληκτρολογώντας τον παρακάτω κώδικα, μπορούμε να αντλήσουμε πληροφορίες και για μεμονωμένα pixels σε μία εικόνα:

```
A(265,493,:)
```

Το αποτέλεσμα που εμφανίζεται αφορά το pixel που βρίσκεται στη θέση $(x,y) = (265,493)$ και αναφέρεται στις τιμές έντασης της φωτεινότητας για κάθε ένα από τα χρώματα κόκκινο-πράσινο-μπλε, που βρίσκονται αποθηκευμένες στον πίνακα A (Εικόνα 4.13). Αντίστοιχα μπορούμε να αντλήσουμε πληροφορίες για την ένταση φωτεινότητας στο συγκεκριμένο (ή διαφορετικό) pixel, από τους πίνακες B και C.



Εικόνα 4.13 - Αποτελέσματα τιμών RGB για το pixel στη θέση $(x,y)=(265,493)$

Στη συνέχεια θα εφαρμόσουμε το διακριτό μετασχηματισμό συνημιτόνου. Όπως αναφέρθηκε νωρίτερα, ο διακριτός μετασχηματισμός συνημιτόνου αναπαριστά μία εικόνα ως ένα άθροισμα συνημιτονοειδών διαφορετικού πλάτους και συχνότητας. Στο MATLAB υπάρχουν δύο μέθοδοι για την εφαρμογή του. Η πρώτη είναι καλώντας τη συνάρτηση 'dct2', η οποία χρησιμοποιεί έναν αλγόριθμο που βασίζεται στο γρήγορο μετασχηματισμό Fourier για την εφαρμογή σε εικόνες υψηλής ανάλυσης. Η δεύτερη μέθοδος χρησιμοποιεί έναν πίνακα μετασχηματισμού που προκύπτει από τη συνάρτηση 'dctmtx' και είναι πιο αποδοτική για τετραγωνικές εικόνες χαμηλής ανάλυσης.

Θα εξετάσουμε την εφαρμογή του DCT και με τις δύο μεθόδους, ωστόσο στην πρώτη μέθοδο θα γίνει απλή αναφορά, ενώ η δεύτερη θα χρησιμοποιηθεί για τη μελέτη διαφορετικών ποσοστών συμπίεσης και τη διεξαγωγή αποτελεσμάτων.

4.3.2 Εφαρμογή του DCT με κλήση της συνάρτησης 'dct2'

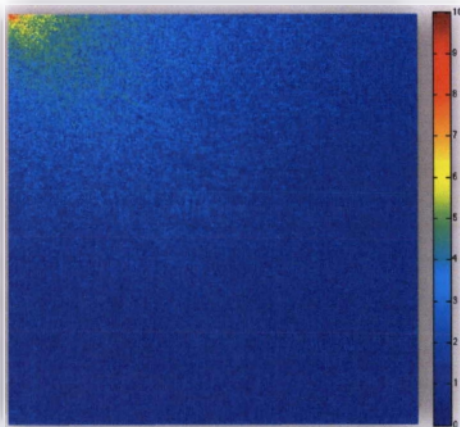
Έχοντας κρατήσει τους πίνακες από το προηγούμενο παράδειγμα, θα εφαρμόσουμε το DCT καλώντας τη συνάρτηση 'dct2'. Η συνάρτηση δέχεται σαν είσοδο έναν πίνακα δύο διαστάσεων, το μετασχηματίζει εφαρμόζοντας το διακριτό μετασχηματισμό συνημιτόνου και σαν έξοδο δίνει έναν πίνακα δύο διαστάσεων, ο οποίος περιέχει τους συντελεστές του μετασχηματισμού. Πληκτρολογούμε:

```
DCT = dct2(B);
figure, imshow(log(abs(DCT)), [0,10]), colormap(jet(256)), colorbar;
```

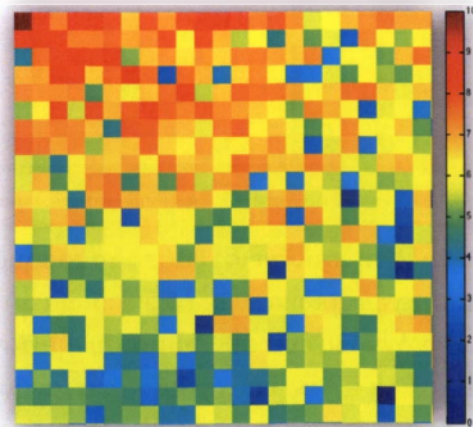
Ο παραπάνω κώδικας μετασχηματίζει τον πίνακα B και αποθηκεύει τους συντελεστές σε έναν νέο δισδιάστατο πίνακα, με όνομα DCT. Στη συνέχεια απεικονίζει γραφικά τους συντελεστές αυτούς, υπολογίζοντας για κάθε έναν το λογάριθμο της απόλυτης τιμής του, προκειμένου να φανεί η δύναμη της συμβολής του (Εικόνα 4.14).

Προκειμένου να κατανοήσουμε ακόμη καλύτερα το ρόλο των συντελεστών, καθώς επίσης τη συσχέτισή τους αλλά και το ρόλο τους στη συμπίεση και την ανακατασκευή της εικόνας, θα μηδενίσουμε μερικούς από αυτούς (συγκεκριμένα όσους έχουν απόλυτη τιμή μικρότερη από 60) και στη συνέχεια θα εμφανίσουμε τη γραφική απεικόνισή τους, υπολογίζοντας πάλι για κάθε έναν το λογάριθμο της απόλυτης τιμής του (Εικόνα 4.16). Πληκτρολογούμε:

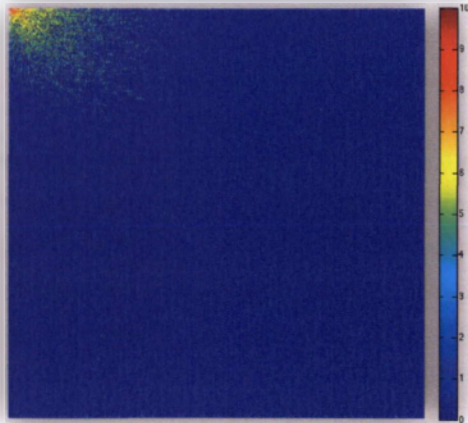
```
DCT(abs(DCT)<60) = 0;
figure, imshow(log(abs(DCT)), [0,10]), colormap(jet(256)), colorbar;
```



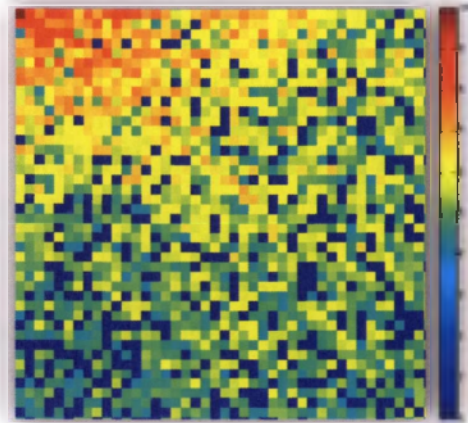
Εικόνα 4.14 - Γραφική απεικόνιση της δύναμης συντελεστών DCT



Εικόνα 4.15 - Συσχέτιση συντελεστών DCT



Εικόνα 4.16 - Γραφική απεικόνιση της δύναμης συντελεστών DCT μετά από επιλεκτικό μηδενισμό



Εικόνα 4.17 – Συσχέτιση συντελεστών DCT μετά από επιλεκτικό μηδενισμό

Παρατηρούμε ότι οι συντελεστές που βρίσκονται επάνω αριστερά έχουν σαφώς μεγαλύτερη δύναμη και παρουσιάζουν μεγάλη συσχέτιση μεταξύ τους (Εικόνα 4.15), ενώ όσο απομακρυνόμαστε η τιμή των συντελεστών μειώνεται, όπως επίσης μειώνεται και η συσχέτιση τους με τους αρχικούς. Μηδενίζοντας κάποιους από αυτούς και συγκεκριμένα όπως προκύπτει από την Εικόνα 4.16, περίπου το 90%, κρατάμε την πληροφορία μόνο από το 10% του αρχικού συνόλου των συντελεστών. Προφανώς η εικόνα που θα προκύψει από την ανασύνθεση των εναπομεινάντων συντελεστών θα είναι υποδεέστερη της αρχικής, ωστόσο θα είναι αναγνωρίσιμη όπως προκύπτει από την Εικόνα 4.18 και την Εικόνα 4.19 αντίστοιχα.



Εικόνα 4.18 - Ανασύνθεση εικόνας με το 100% των συντελεστών DCT



Εικόνα 4.19 - Ανασύνθεση εικόνας με το 10% των συντελεστών DCT

4.3.3 Εφαρμογή του DCT με χρήση πίνακα

Στο παράδειγμα που ακολουθεί εξετάζουμε αναλυτικά την εφαρμογή του διακριτού μετασχηματισμού συνημιτόνου με χρήση πίνακα συντελεστών.

Πιο αναλυτικά, ο κώδικας που ακολουθεί ορίζει το διακριτό μετασχηματισμό συνημιτόνου για 8x8 blocks, χωρίζει την εικόνα σε 8x8 blocks και τα μετασχηματίζει με βάση τον παραπάνω πίνακα, απορρίπτοντας κάθε φορά ένα σύνολο συντελεστών. Στη συνέχεια ανασυνθέτει την εικόνα εφαρμόζοντας τον αντίστροφο μετασχηματισμό σε κάθε 8x8 block, εμφανίζει το αρχικό και το τελικό αποτέλεσμα, καθώς επίσης και γραφικές απεικονίσεις της έντασης φωτεινότητας/pixel αλλά και των συντελεστών DCT.

Επειδή χρησιμοποιούνται νέα δεδομένα και μεταβλητές, πληκτρολογούμε τον παρακάτω κώδικα προκειμένου να σβήσουμε από το MATLAB τα δεδομένα του προηγούμενου παραδείγματος.

```
clear all;
close all;
clc;
```

Θεωρώντας ότι χρησιμοποιούμε την ίδια εικόνα με το προηγούμενο παράδειγμα, πληκτρολογούμε τον παρακάτω κώδικα:

```
A = imread('lenna.tif');
I = rgb2gray(A);
I = im2double(I);
T = dctmtx(8);
dct = @(block_struct) T * block_struct.data * T';
B = blockproc(I, [8 8], dct);
mask = [1 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0];
B2 = blockproc(B, [8 8], @(block_struct) mask .* block_struct.data);
invdct = @(block_struct) T' * block_struct.data * T;
I2 = blockproc(B2, [8 8], invdct);
```

Ο παραπάνω κώδικας, γραμμή προς γραμμή εκτελεί τα παρακάτω βήματα:

- διαβάζει την αρχική εικόνα βάθους χρώματος 24-bit και αποθηκεύει τις τιμές σε έναν τρισδιάστατο πίνακα A,
- μετατρέπει την εικόνα σε 8-bit grayscale και αποθηκεύει τις τιμές σε ένα διςδιάστατο πίνακα I,
- αλλάζει τον τύπο μεταβλητής σε πραγματικό διπλής ακρίβειας,
- ορίζει τον $T_{8 \times 8}$ πίνακα μετασχηματισμού DCT,
- ορίζει τη συνάρτηση μετασχηματισμού dct,

- χωρίζει την εικόνα σε 8x8 blocks και τη μετασχηματίζει, αποθηκεύοντας τους συντελεστές σε ένα δισδιάστατο πίνακα B (Εικόνα 4.20),
- ορίζει μία μάσκα για το σύνολο των συντελεστών DCT που απορρίπτονται (αρχικά, για κάθε 8x8 block, απορρίπτονται οι 63 από τους 64 συντελεστές, και παραμένει μόνο ο 1^{ος}) (Εικόνα 4.21),
- εφαρμόζει τη μάσκα στον μετασχηματισμένο πίνακα και αποθηκεύει τις νέες τιμές σε έναν νέο πίνακα B2,
- ορίζει τη συνάρτηση για τον αντίστροφο μετασχηματισμό idct,
- ανασυνθέτει την εικόνα από τον πίνακα B2 που προέκυψε μετά την απόρριψη συντελεστών,

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	5.2809	0.0215	0.0162	-6.0874e...	4.9020e...	-8.5938e...	-0.0263	0.0100	5.2108	0.0908	0.0123	0.0195	-0.0333	0.04
2	0.0356	-0.0078	0.0042	-0.0240	0.0079	0.0164	-0.0170	0.0126	0.0532	-0.0090	0.0098	0.0262	-0.0166	0.01
3	-0.0233	2.3787e...	-0.0100	0.0076	-0.0016	-0.0031	0.0080	-0.0070	-0.0200	0.0087	-0.0067	-0.0141	0.0044	-0.00
4	0.0116	0.0066	0.0112	0.0057	-0.0051	-0.0064	0.0012	0.0035	-0.0080	-0.0072	0.0027	0.0029	0.0072	-0.00
5	-0.0064	-0.0090	-0.0042	-0.0072	0.0093	0.0049	-0.0062	-0.0020	0.0137	0.0044	9.4347e...	0.0015	-0.0088	0.00
6	0.0072	0.0066	-0.0074	-0.0010	-0.0099	0.0049	0.0063	0.0031	-9.0072e...	-0.0012	-0.0032	6.3100e...	0.0018	0.00
7	-0.0089	-0.0026	0.0149	0.0093	0.0076	-0.0129	-0.0037	-0.0042	-0.0150	-9.5628e...	0.0035	-0.0041	0.0056	-0.00
8	0.0066	2.7318e...	-0.0118	-0.0089	-0.0039	0.0110	0.0013	0.0011	0.0151	0.0012	-0.0022	0.0041	-0.0062	0.00
9	5.1275	0.0200	-0.0233	0.0029	-0.0069	0.0091	1.0208e...	0.0030	5.1279	0.0268	0.0080	0.0069	0.0289	0.03
10	-0.0112	-0.0198	-0.0182	-0.0089	0.0174	-0.0010	0.0028	-0.0063	0.0170	0.0138	0.0040	0.0091	-0.0029	0.03
11	0.0130	-0.0200	0.0070	-0.0036	-0.0152	0.0012	0.0206	-0.0058	0.0033	0.0058	-0.0113	0.0086	0.0071	0.00
12	0.0206	-0.0236	0.0082	2.0097e...	0.0023	-0.0120	0.0035	0.0128	6.3481e...	0.0035	0.0158	-0.0013	0.0212	5.0133e...
13	0.0059	0.0083	-0.0055	0.0071	-9.8039e...	-7.3271e...	-0.0030	0.0013	0.0113	0.0134	0.0113	-0.0160	-0.0015	1.1122e...
14	-0.0016	0.0046	-0.0170	0.0173	-0.0038	-0.0057	0.0076	-0.0068	0.0038	-0.0035	-4.5852e...	-0.0021	-2.2074e...	-0.00
15	0.0084	0.0020	-0.0137	0.0111	0.0042	-0.0025	-0.0041	-0.0014	0.0066	0.0072	0.0058	-0.0147	0.0082	-0.01
16	0.0088	-0.0064	0.0221	-1.2610e...	-0.0081	0.0070	-0.0089	0.0018	-0.0065	0.0061	-0.0056	-6.0012e...	-0.0114	-0.00
17	5.1020	0.0169	-0.0041	0.0071	3.3307e...	-5.1592e...	-0.0047	-0.0030	5.1456	-0.0158	0.0027	0.0108	0.0054	-0.00
18	0.0029	0.0074	-0.0295	0.0129	0.0078	-0.0084	-0.0016	0.0076	-0.0157	0.0047	0.0060	1.1190e...	-0.0156	-0.01
19	-0.0054	0.0142	0.0165	-0.0087	4.9288e...	-0.0051	9.5410e...	0.0068	-0.0118	0.0046	-0.0127	0.0041	-0.0042	-0.00
20	-0.0017	-0.0015	-0.0044	-0.0017	0.0097	0.0121	-0.0206	-0.0015	-0.0088	-0.0048	-0.0078	0.0031	0.0184	-0.00
21	1.0788e...	-0.0222	-0.0098	0.0076	-0.0078	0.0122	0.0019	-0.0125	0.0113	0.0027	0.0032	-0.0194	-0.0015	-0.01

Εικόνα 4.20 – Πίνακας συντελεστών DCT πριν την απόρριψη

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	5.2809	0	0	0	0	0	0	0	5.2108	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	5.1275	0	0	0	0	0	0	0	5.1279	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	5.1020	0	0	0	0	0	0	0	5.1456	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Εικόνα 4.21 - Πίνακας συντελεστών DCT μετά την απόρριψη

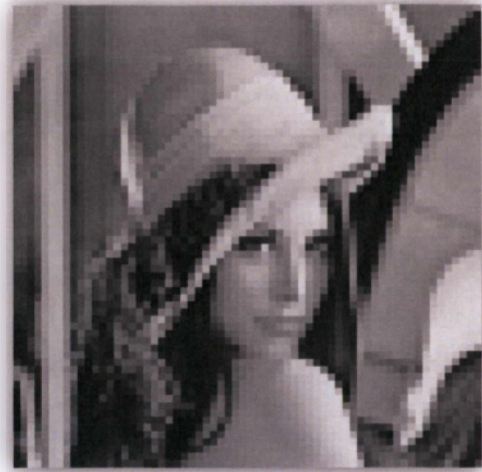
Έχοντας ολοκληρώσει τη διαδικασία εισαγωγής της εικόνας στο MATLAB και της εφαρμογής του ευθύ και αντίστροφου μετασχηματισμού με απόρριψη των 63 συντελεστών από το σύνολο των 64, πληκτρολογούμε τον παρακάτω κώδικα για να

εμφανίσουμε την εικόνα πριν (Εικόνα 4.22) και μετά το μετασχηματισμό (Εικόνα 4.23).

```
figure, imshow(I)
figure, imshow(I2)
```



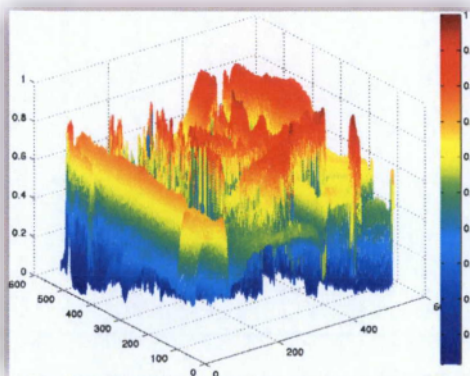
Εικόνα 4.22 – Πηγαία εικόνα
πριν το μετασχηματισμό



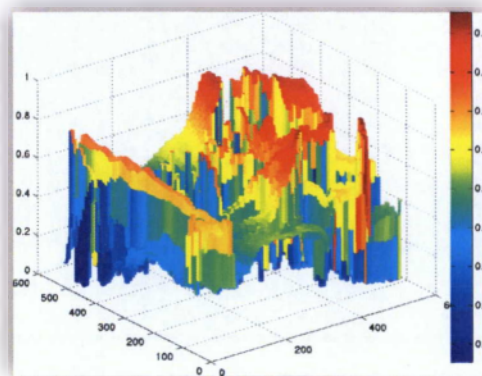
Εικόνα 4.23 – Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT

Ο παρακάτω κώδικας εμφανίζει τις τιμές έντασης φωτεινότητας για κάθε pixel. Παρατηρούμε ότι πριν την απόρριψη συντελεστών, για κάθε 8x8 block υπάρχουν 64 διαφορετικές τιμές όπως φαίνεται στην Εικόνα 4.24 και την Εικόνα 4.26 (λεπτομέρεια: Εικόνα 4.28), ενώ μετά την απόρριψη υπάρχει μόνο μία (αυτή που αντιστοιχεί στον 1^ο συντελεστή) όπως φαίνεται στην Εικόνα 4.25 και την Εικόνα 4.27 (λεπτομέρεια: Εικόνα 4.29).

```
figure(1), mesh(I), colormap(jet(256)), colorbar
figure(2), mesh(I2), colormap(jet(256)), colorbar
```



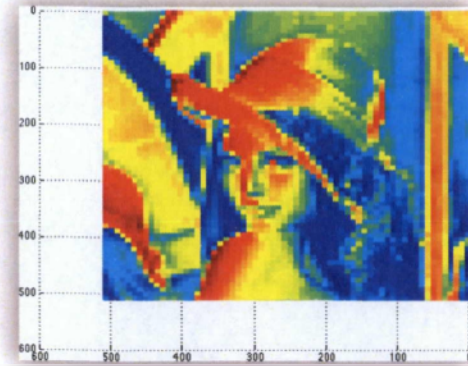
Εικόνα 4.24 – Τιμές έντασης φωτεινότητας
πηγαίας εικόνας πριν το μετασχηματισμό (3D)



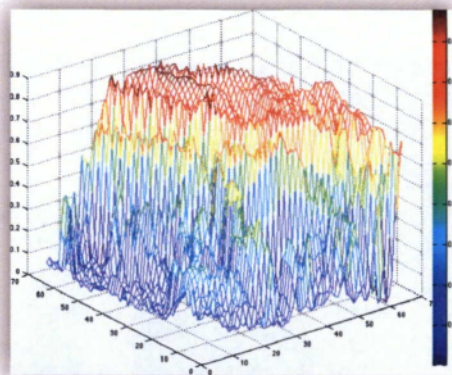
Εικόνα 4.25 - Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την απόρριψη
συντελεστών DCT (3D)



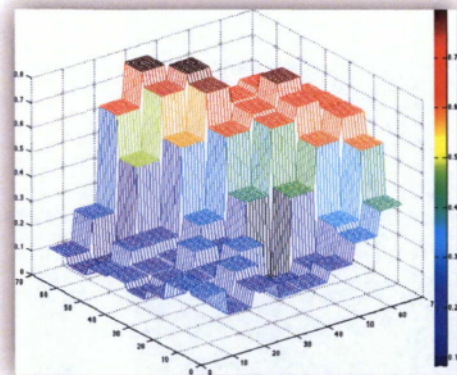
Εικόνα 4.26 - Τιμές έντασης φωτεινότητας
πηγαίας εικόνας πριν το μετασχηματισμό (2D)



Εικόνα 4.27 - Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την απόρριψη
συντελεστών DCT (2D)



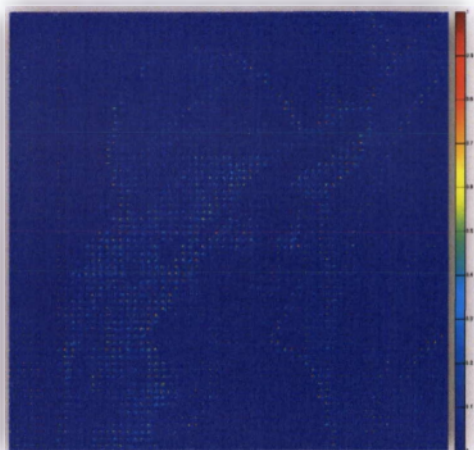
Εικόνα 4.28 - Τιμές έντασης φωτεινότητας
πηγαίας εικόνας πριν το μετασχηματισμό
(3D, λεπτομέρεια)



Εικόνα 4.29 - Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την απόρριψη
συντελεστών DCT (3D, λεπτομέρεια)

Ο παρακάτω κώδικας απεικονίζει γραφικά τους πίνακες B και B2 που περιέχουν τους συντελεστές DCT. Ο πίνακας B περιέχει όλους τους συντελεστές για κάθε 8x8 block (), ενώ στον πίνακα B2 υπάρχει μόνο ο 1^{ος} από κάθε 8x8 block ().

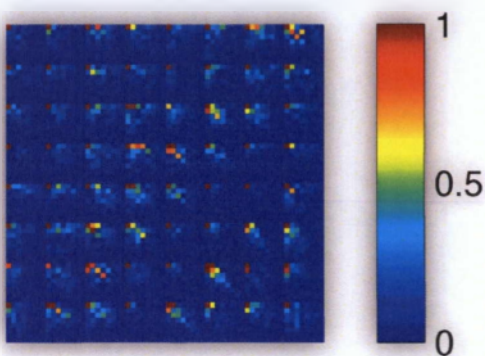
```
figure, imshow(abs(B),[]), colormap(jet(256)), colorbar;
figure, imshow(abs(B2),[]), colormap(jet(256)), colorbar;
```



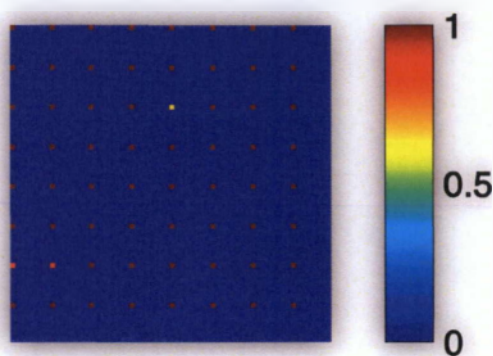
Εικόνα 4.30 – Πηγαία εικόνα χωρισμένη σε 8x8 blocks και συντελεστές DCT πριν την απόρριψη



Εικόνα 4.31 - Πηγαία εικόνα χωρισμένη σε 8x8 blocks και συντελεστές DCT μετά την απόρριψη



Εικόνα 4.32 - Συντελεστές DCT πηγαίας εικόνας (λεπτομέρεια)



Εικόνα 4.33 - Συντελεστές DCT μετά την απόρριψη (λεπτομέρεια)

Με παρόμοιο τρόπο, διατηρώντας κάθε φορά ολοένα και περισσότερους συντελεστές (αλλάζοντας δηλαδή τη μάσκα από 0 σε 1), παρατηρούμε ότι η ποιότητα της ανακατασκευασμένης εικόνας βελτιώνεται σταδιακά όπως προκύπτει από τις εικόνες στις σελίδες που ακολουθούν.

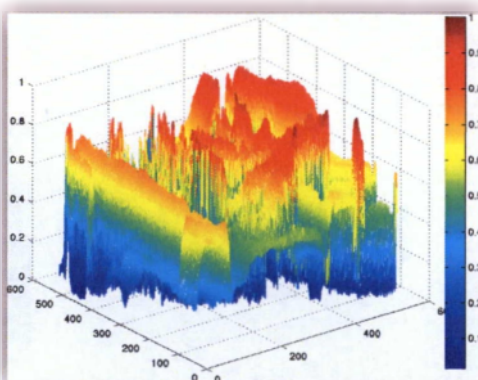
Διατήρηση 3 συντελεστών:



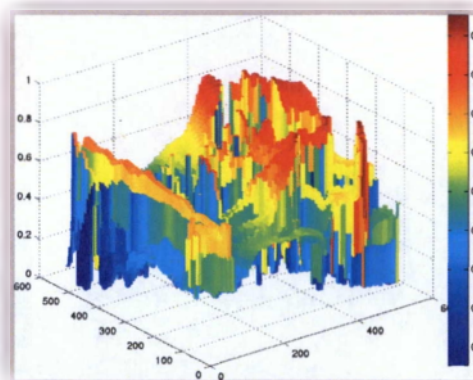
Πηγαία εικόνα
πριν το μετασχηματισμό



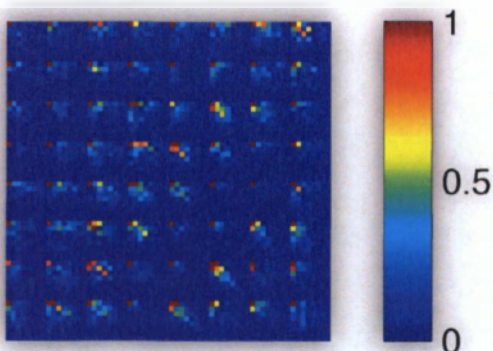
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



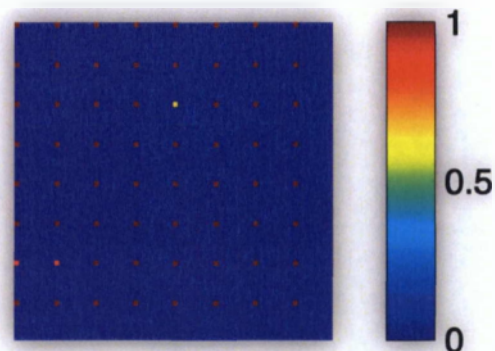
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό DCT (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.34 – Αποτελέσματα για διατήρηση 3/64 συντελεστών

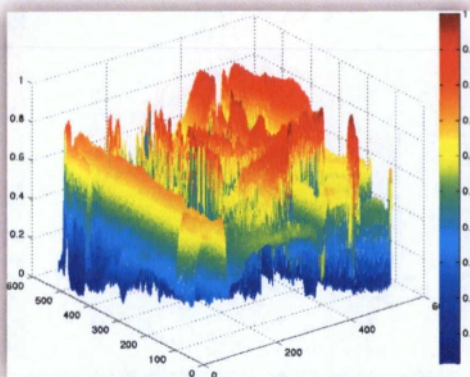
Διατήρηση 6 συντελεστών:



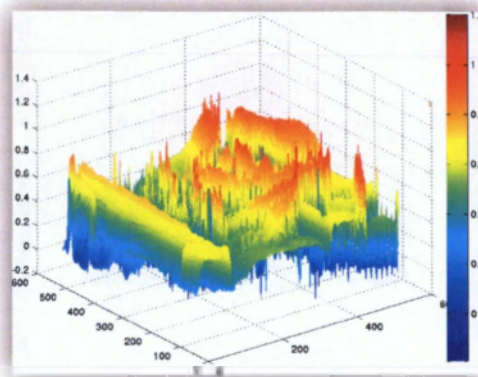
Πηγαία εικόνα
πριν το μετασχηματισμό



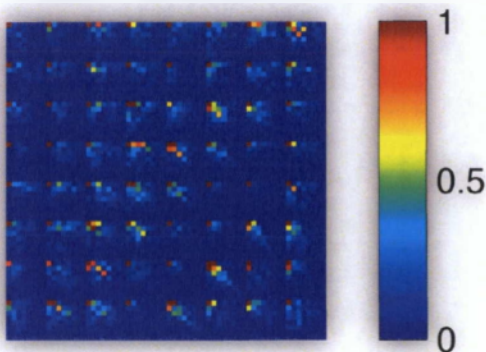
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



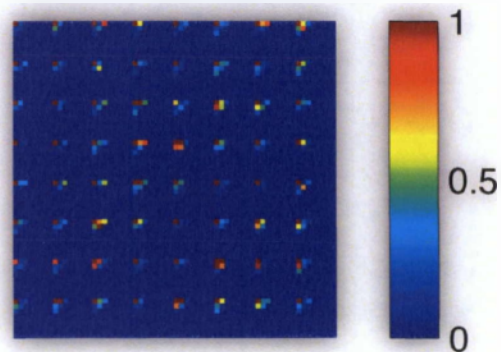
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό DCT (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.35 - Αποτελέσματα για διατήρηση 6/64 συντελεστών

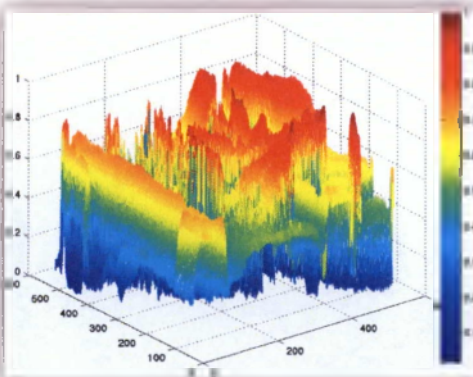
Διατήρηση 10 συντελεστών:



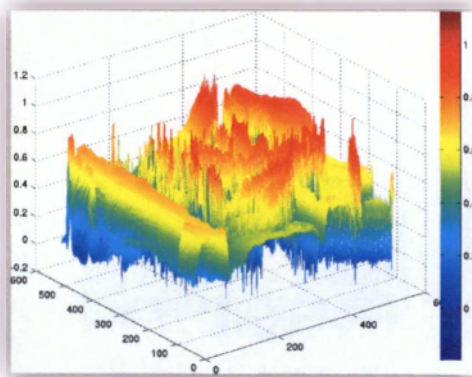
Πηγαία εικόνα
πριν το μετασχηματισμό



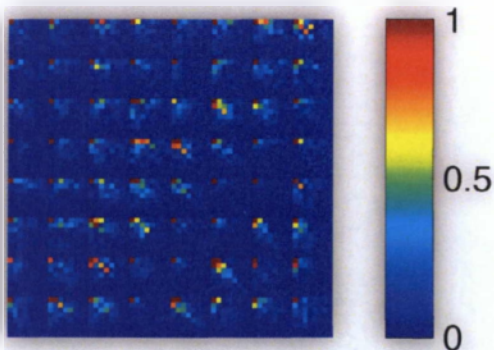
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



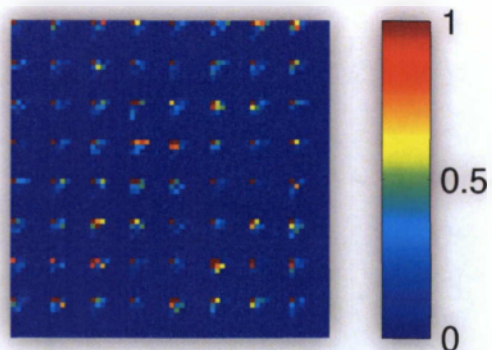
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό DCT (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.36 - Αποτελέσματα για διατήρηση 10/64 συντελεστών

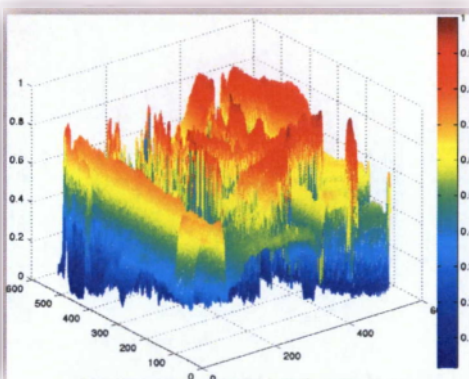
Διατήρηση 15 συντελεστών:



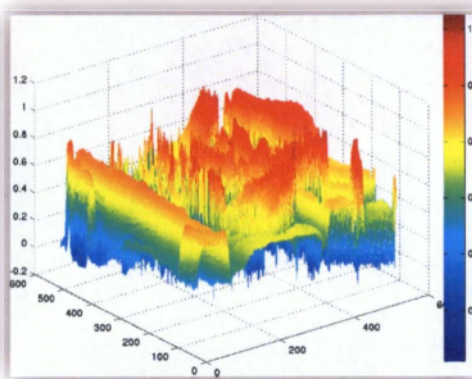
Πηγαία εικόνα
πριν το μετασχηματισμό



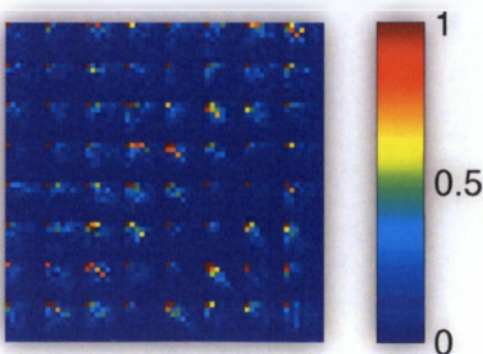
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



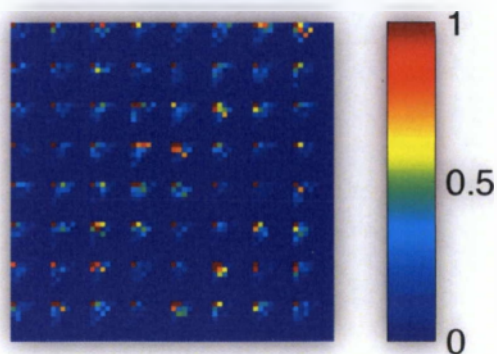
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό DCT (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.37 - Αποτελέσματα για διατήρηση 15/64 συντελεστών

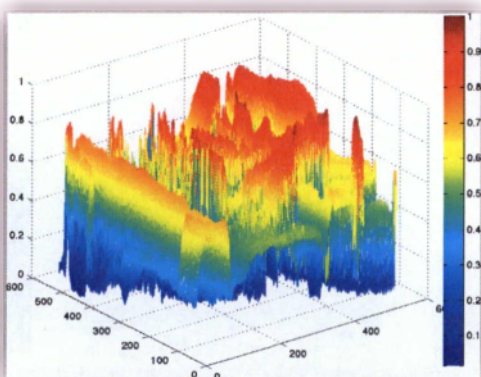
Διατήρηση 21 συντελεστών:



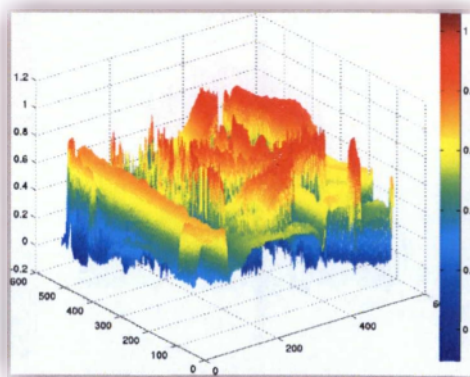
Πηγαία εικόνα
πριν το μετασχηματισμό



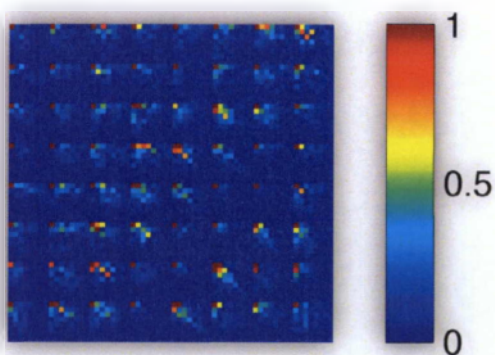
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



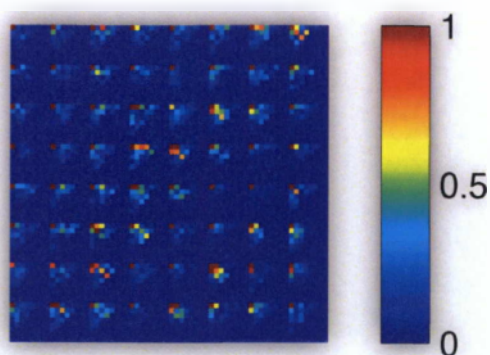
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.38 - Αποτελέσματα για διατήρηση 21/64 συντελεστών

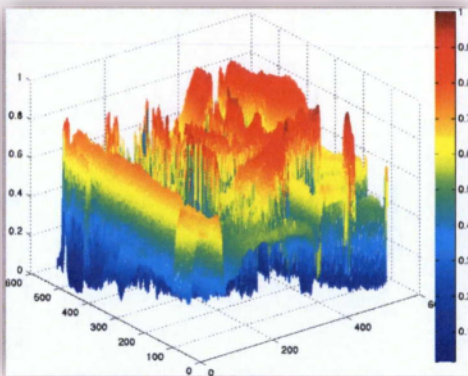
Διατήρηση 28 συντελεστών:



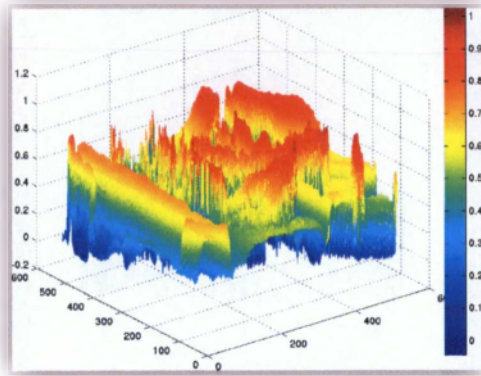
Πηγαία εικόνα
πριν το μετασχηματισμό



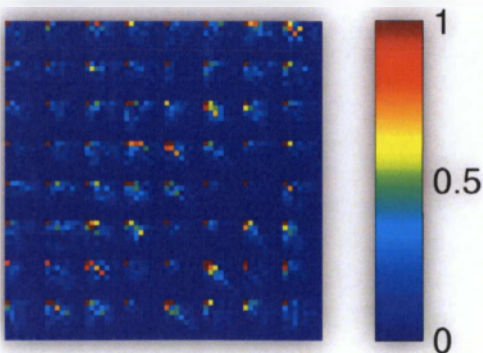
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



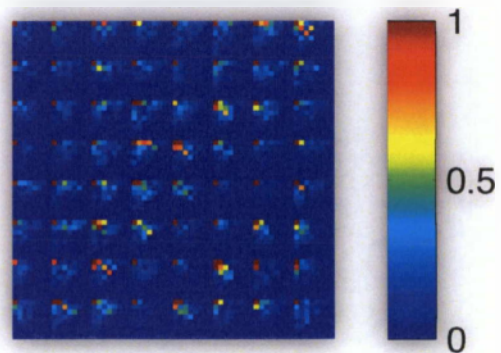
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.39 - Αποτελέσματα για διατήρηση 28/64 συντελεστών

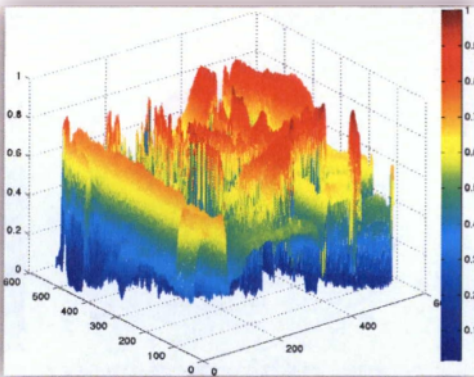
Διατήρηση 36 συντελεστών:



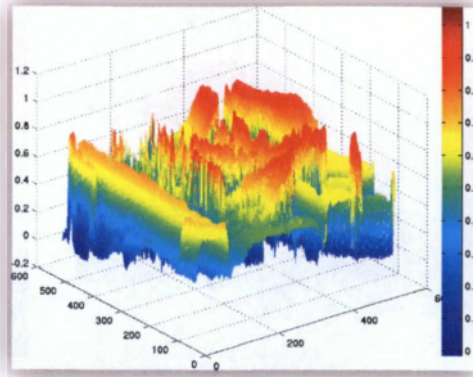
Πηγαία εικόνα
πριν το μετασχηματισμό



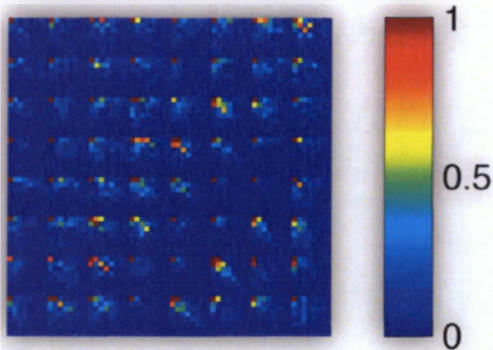
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



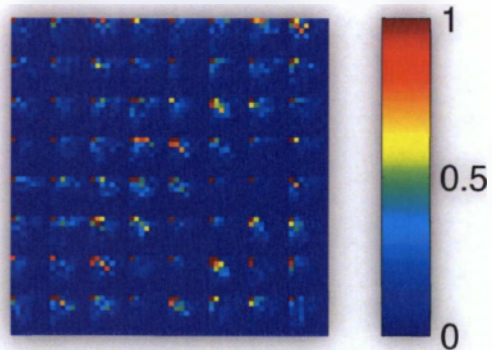
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.40 - Αποτελέσματα για διατήρηση 36/64 συντελεστών

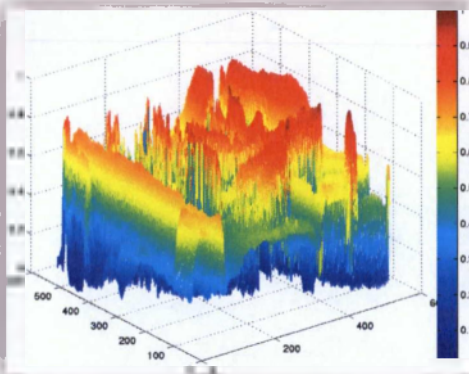
Διατήρηση 43 συντελεστών:



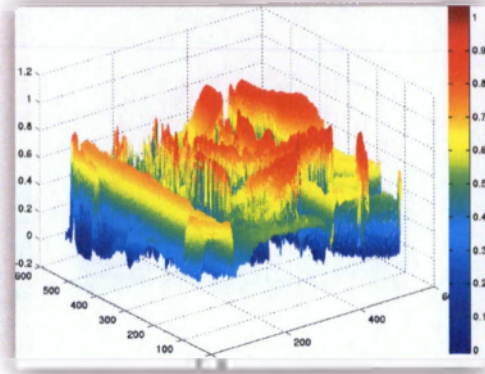
Πηγαία εικόνα
πριν το μετασχηματισμό



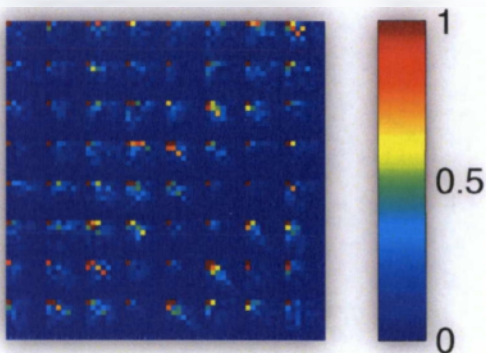
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



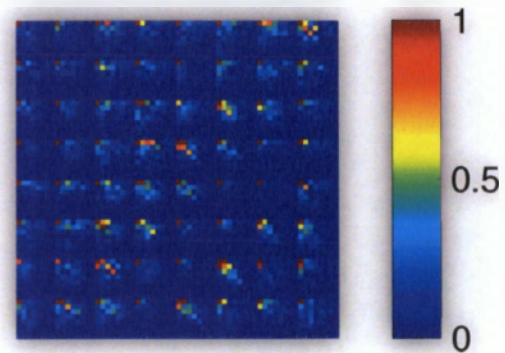
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.41 - Αποτελέσματα για διατήρηση 43/64 συντελεστών

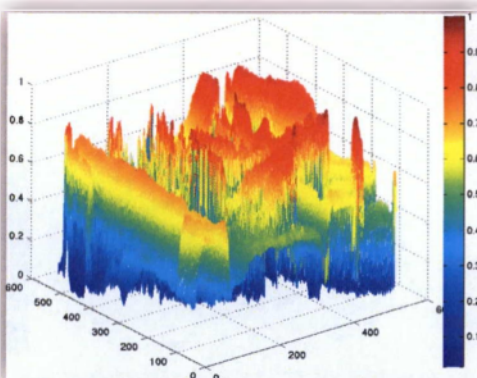
Διατήρηση 49 συντελεστών:



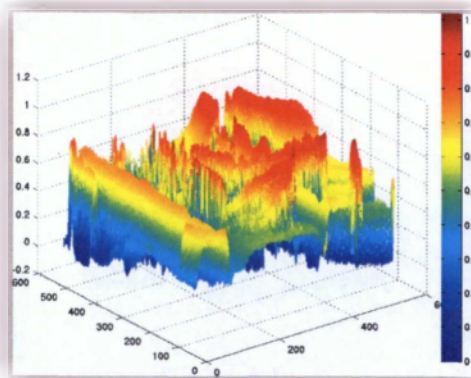
Πηγαία εικόνα
πριν το μετασχηματισμό



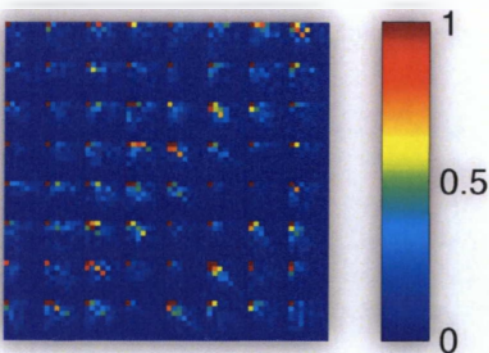
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



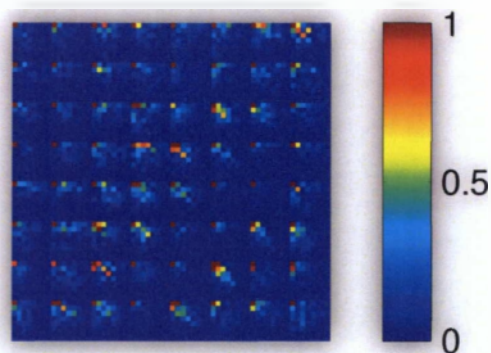
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.42 - Αποτελέσματα για διατήρηση 49/64 συντελεστών

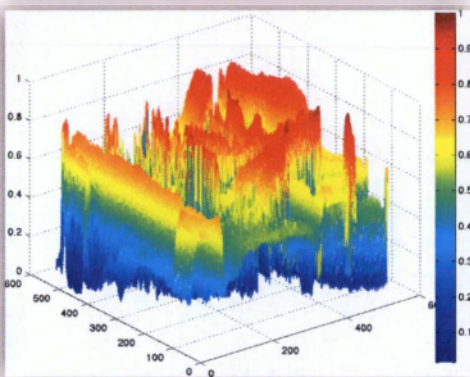
Διατήρηση 54 συντελεστών:



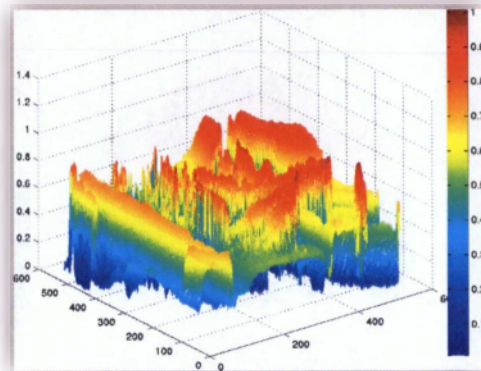
Πηγαία εικόνα
πριν το μετασχηματισμό



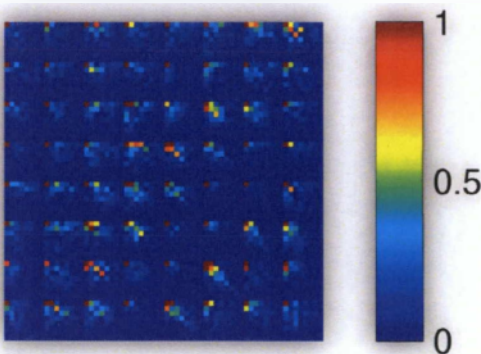
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



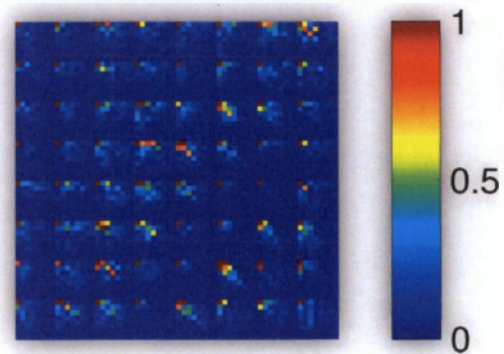
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.43 - Αποτελέσματα για διατήρηση 54/64 συντελεστών

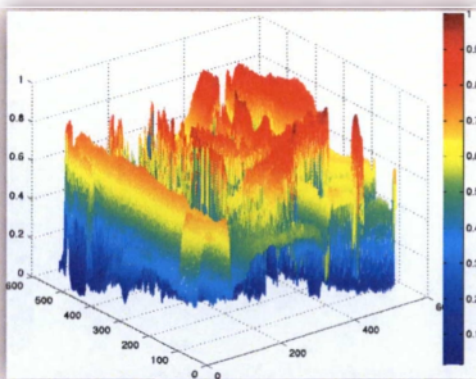
Διατήρηση 58 συντελεστών:



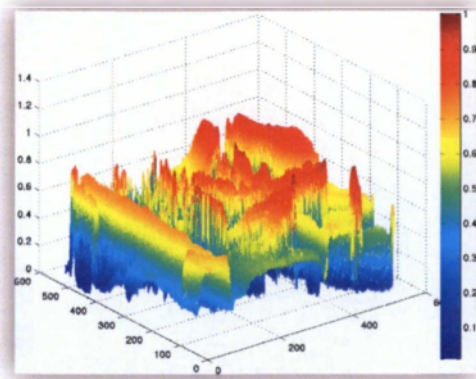
Πηγαία εικόνα
πριν το μετασχηματισμό



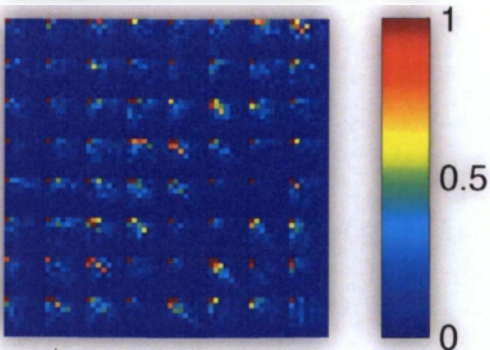
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



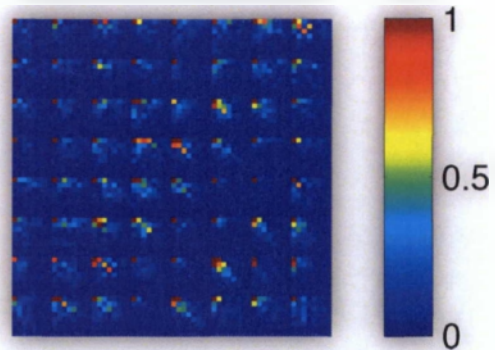
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.44 - Αποτελέσματα για διατήρηση 58/64 συντελεστών

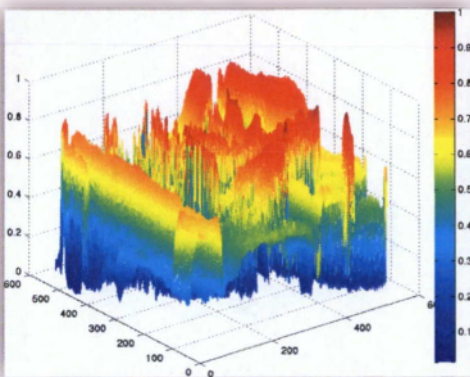
Διατήρηση 61 συντελεστών:



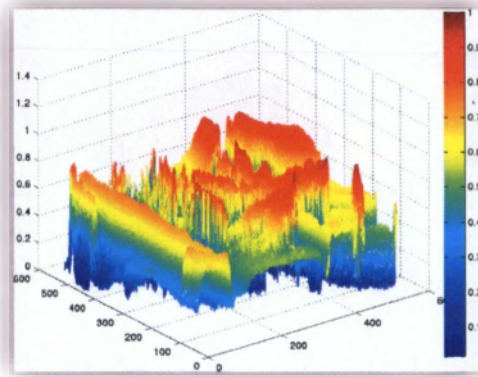
Πηγαία εικόνα
πριν το μετασχηματισμό



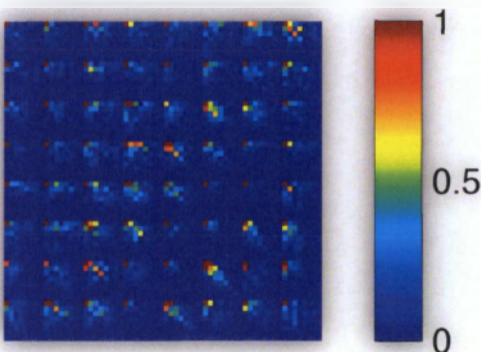
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



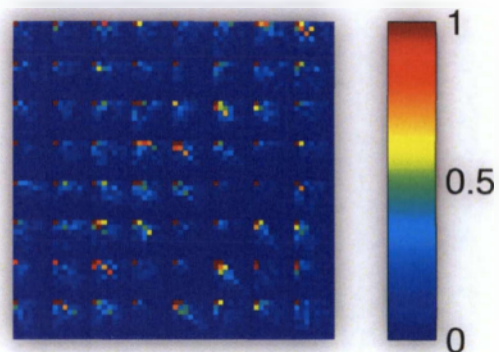
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.45 - Αποτελέσματα για διατήρηση 61/64 συντελεστών

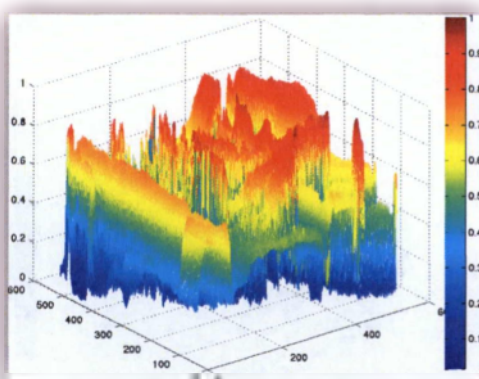
Διατήρηση 63 συντελεστών:



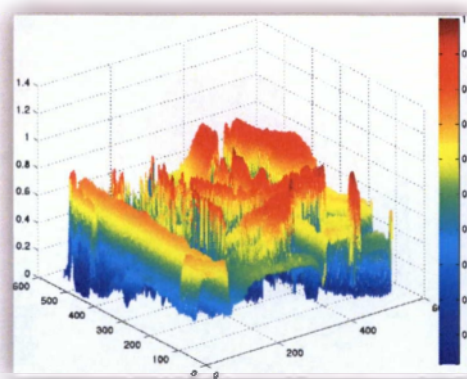
Πηγαία εικόνα
πριν το μετασχηματισμό



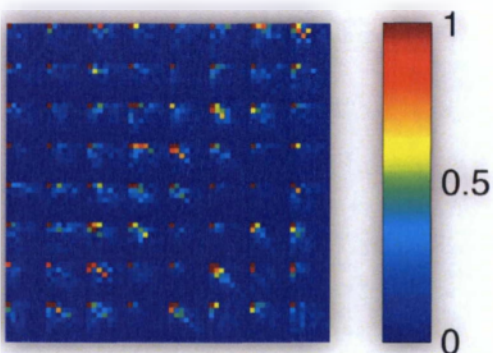
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



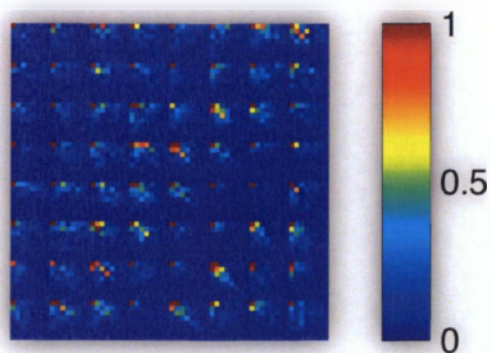
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.46 - Αποτελέσματα για διατήρηση 63/64 συντελεστών

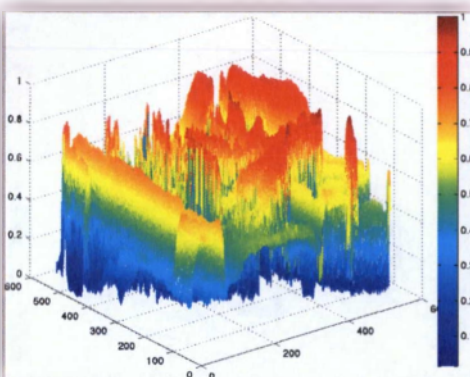
Διατήρηση 64 συντελεστών:



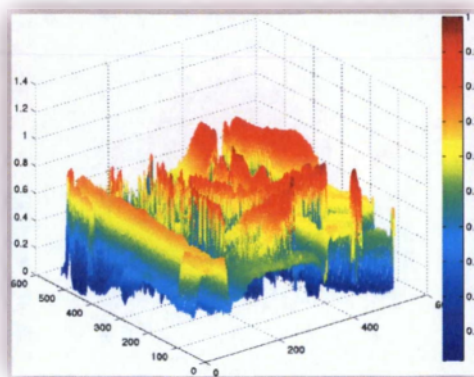
Πηγαία εικόνα
πριν το μετασχηματισμό



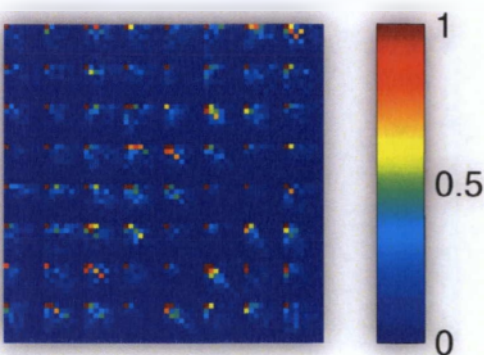
Πηγαία εικόνα
μετά από μετασχηματισμό
και απόρριψη συντελεστών DCT



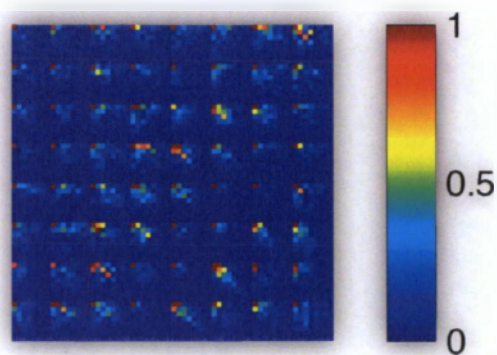
Τιμές έντασης φωτεινότητας πηγαίας εικόνας
πριν το μετασχηματισμό (3D)



Τιμές έντασης φωτεινότητας
μετά το μετασχηματισμό και την
απόρριψη συντελεστών DCT (3D)



Συντελεστές DCT
πηγαίας εικόνας (λεπτομέρεια)



Συντελεστές DCT
μετά την απόρριψη (λεπτομέρεια)

Εικόνα 4.47 - Αποτελέσματα για διατήρηση 64/64 συντελεστών

5

Επίλογος

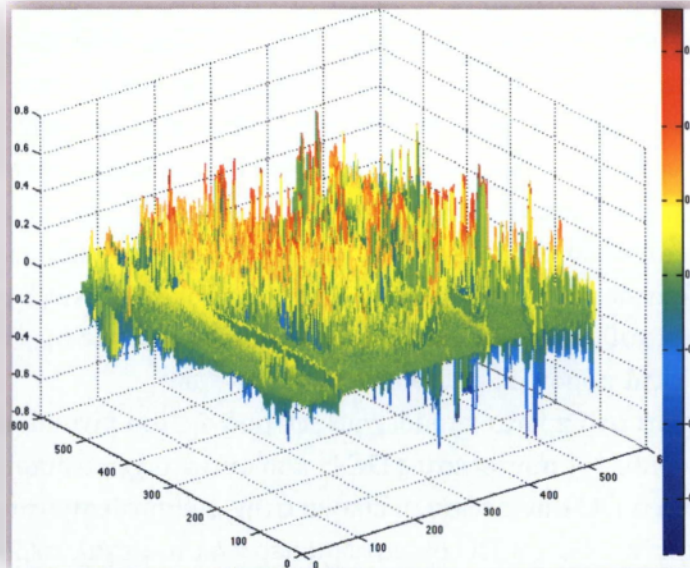
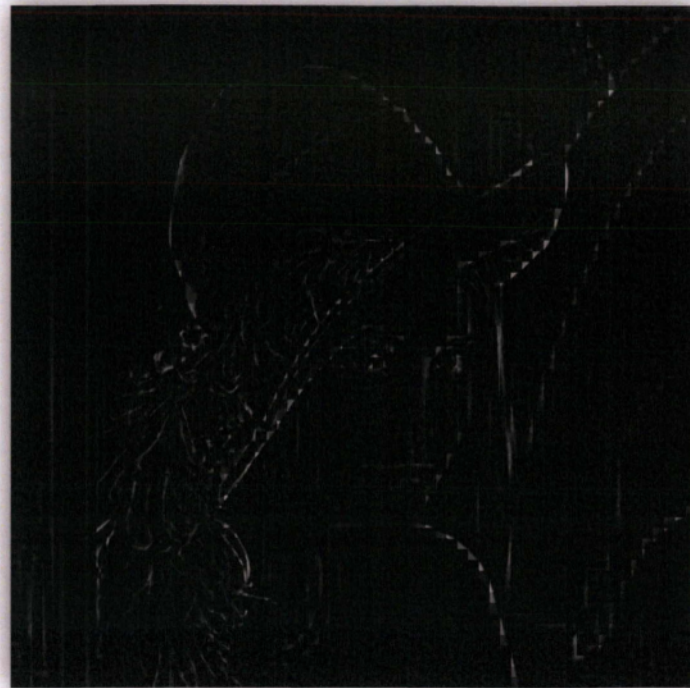
Συνοψίζοντας, ο διακριτός μετασχηματισμός συνημιτόνου έχει πολλά χαρακτηριστικά που τον καθιστούν ένα αποτελεσματικό εργαλείο για τη συμπίεση ψηφιακών εικόνων. Είναι ορθομοναδιαίος, που σημαίνει ότι ικανοποιεί τη συνθήκη διατήρησης της ενέργειας, υπάρχουν πολύ αποτελεσματικοί και γρήγοροι αλγόριθμοι για τον υπολογισμό του, ενώ οι πίνακες που δίνει σαν αποτέλεσμα, αποτελούνται σε μεγάλο ποσοστό από τιμές κοντά στο μηδέν.

Το κυριότερο μειονέκτημα του είναι ότι ενώ τα 8x8 blocks στην είσοδο του αποτελούνται από ακέραιους αριθμούς, η τιμές τις εξόδου είναι πραγματικοί αριθμοί. Για το λόγο αυτό απαιτείται ένας κβαντιστής στην έξοδο του μετασχηματιστή προκειμένου να παράγει μία έξοδο που θα αποτελείται από ακέραιους αριθμούς. Όλοι οι αριθμοί που είναι κοντά στο μηδέν, μηδενίζονται, ενώ άλλοι συρρικνώνονται ώστε να έχουν τιμές πιο κοντά στο μηδέν. Τελικά όλες οι κβαντισμένες τιμές στρογγυλοποιούνται στον πλησιέστερο ακέραιο και στη συνέχεια κωδικοποιούνται. Η απόρριψη συντελεστών DCT, η κβάντιση και η κωδικοποίηση κατά το πρότυπο JPEG είναι αυτά που το καθιστούν ως απώλεστικό πρότυπο συμπίεσης, το οποίο ωστόσο χρησιμοποιείται κατά κόρον σε μία πληθώρα εφαρμογών.

Παρά το γεγονός ότι στο πρώτο παράδειγμα εφαρμογής του πίνακα μετασχηματισμού απορρίφθηκαν οι 63/64 συντελεστές DCT, που αντιστοιχεί περίπου στο 98%, και διατηρήθηκε μόνο η DC συνιστώσα, η εικόνα στην ανασύνθεση ήταν αναγνωρίσιμη.

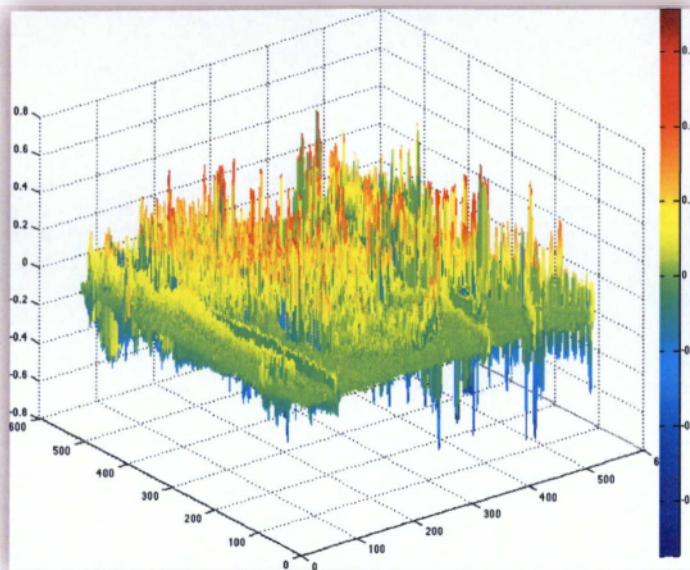
Αυτό συμβαίνει διότι η DC συνιστώσα περιέχει το μεγαλύτερο ποσοστό πληροφορίας. Χωρίς αυτήν, το αποτέλεσμα στην έξοδο ακόμη και μετά από

διατήρηση όλων των υπολοίπων συντελεστών θα ήταν όπως στην



Εικόνα 5.1.

Ωστόσο, όπως προκύπτει και από την εφαρμογή στο MATLAB και την εξαγωγή των αποτελεσμάτων που παρουσιάστηκαν στο Κεφάλαιο 4, παρ. 4.3.3, με διατήρηση της DC συνιστώσας και σταδιακή διατήρηση επιπλέον συντελεστών, η ποιότητα της ανακατασκευασμένης εικόνας βελτιωνόταν δραματικά για διατήρηση 3, 6, και 10 συντελεστών.



Εικόνα 5.1 – Παράληψη DC συνιστώσας

Με διατήρηση 15, 21 και 28 συντελεστών, η ποιότητα της εικόνας βελτιώθηκε ακόμη περισσότερο, ωστόσο όχι σε τόσο μεγάλο βαθμό, ενώ στη συνέχεια, για διατήρηση 36, 43 συντελεστών, οι βελτιώσεις στο τελικό οπτικό αποτέλεσμα ήταν ανεπαίσθητες. Με την διατήρηση 43 και 49 συντελεστών, οι βελτιώσεις ήταν σχεδόν μηδαμινές, ενώ με διατήρηση 54, 58, 61, 63 και 64 συντελεστών δεν υπήρξε καμία απολύτως αλλαγή στην ανακατασκευασμένη εικόνα.

Από τα παραπάνω συμπεραίνουμε ότι ο διακριτός μετασχηματισμός συνημιτόνου για κάθε 8x8 block, συγκεντρώνει το μεγαλύτερο μέρος της

πληροφορίας στη DC συνιστώσα και τις κοντινές σε αυτήν AC συνιστώσες. Οι υπόλοιποι συντελεστές έχουν δευτερεύοντα ρόλο και ουσιαστικά η συμβολή τους στην ανακατασκευή της εικόνας (από κάποιο αριθμό συντελεστών και πάνω) είναι μηδαμινή. Είναι αυτοί οι συντελεστές που τελικά μπορούν να παραληφθούν χωρίς αυτό να έχει σοβαρή επίπτωση στο τελικό οπτικό αποτέλεσμα, είναι όμως αρκετό για ένα αρκετά μεγάλο ποσοστό συμπίεσης των δεδομένων, το οποίο στη συνέχεια μεγαλώνει ακόμη περισσότερο μετά την κβάντιση και την κωδικοποίηση.

6

Παράρτημα – κώδικας MATLAB

Παρακάτω παρατίθεται ο κώδικας MATLAB που χρησιμοποιήθηκε κατά την εκπόνηση της παρούσας πτυχιακής εργασίας.

6.1 Εφαρμογή του DCT με κλήση της συνάρτησης 'dct2'

```
clc;
clear all;
close all;

%anagnosi eikonas
A=imread('lenna.tif');
%emfanisi eikonas color 24bit
figure, imshow(A);

%metatropi se greyscale 8bit
B=rgb2gray(A);
%emfanisi eikonas greyscale 8bit
figure, imshow(B);

%metatropi se binary 2bit
C=im2bw(A);
%emfanisi eikonas binary 2bit
figure, imshow(C);

%metatropi pinaka C apo binary se double
D=double(C);
%provoli pixel sto epipedo
imagesc(D), colormap(copper(256));
%provoli pixel sto xoro
mesh(D), colormap(copper(256));

%emfanisi timon foteinotitas RGB se kapoio pixel
A(265,493,:)

%efarmogi tou dct2 (FFT based)
DCT = dct2(B);
```



```

%grafiki apeikonisi syntelestwn
figure, imshow(log(abs(DCT)),[0,10]), colormap(jet(256)), colorbar;
%midenismos syntelestwn me apolyti timi < 60
DCT(abs(DCT)<60) = 0;
%apeikonisi neon sinistoseon
figure, imshow(log(abs(DCT)),[0,10]), colormap(jet(256)), colorbar;

%anakataskevi eikonas me xrisi tou antistrofou DCT
IDCT = idct2(DCT);
%emfanisi arxikis eikonas
figure, imshow(B);
%emfanis eikonas meta tin efarmogi dct kai midenismo sinteleston
figure, imshow(IDCT,[0 255]);

```

6.2 Εφαρμογή του DCT με χρήση πίνακα συντελεστών

6.2.1 Βασικό script

```

clc;
clear all;
close all;

%anagnosi eikonas
A = imread('lenna.tif');

%metatropi se greyscale 8bit kai allagi typou metavlitis
I = rgb2gray(A);
I = im2double(I);

%dimiourgia 8x8 pinaka sintelestwn metasximatismou
T = dctmtx(8);

%orismos sinartisis metasximatismou DCT gia block 8x8
dct = @(block_struct) T * block_struct.data * T';

%diaxorismos eikonas se block 8x8 kai metasximatismos
B = blockproc(I,[8 8],dct);

%maska sinteleston block 8x8
mask = [
1   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
];
%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

```



```

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(1), imshow(I)
figure(2), imshow(I2)
imwrite(I, '01 IMAGE BEFORE DCT.tif');
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(3), mesh(I), colormap(jet(256)), colorbar
figure(4), mesh(I2), colormap(jet(256)), colorbar
saveas(3, '03 FOTEINOTHTA ORIGINAL.tif');
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston prin kai meta tin aporripsi
figure(5), imshow(abs(B),[]), colormap(jet(256)), colorbar;
figure(6), imshow(abs(B2),[]), colormap(jet(256)), colorbar;
saveas(5, '05 ORIGINAL COEFFICIENTS.tif');
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%03
coef = '03';
mask = [
1 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

```



```

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[]), colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%06
coef = '06';
mask = [
1   1   1   0   0   0   0   0
1   1   0   0   0   0   0   0
1   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
0   0   0   0   0   0   0   0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[]), colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%10
coef = '10';
mask = [
1   1   1   1   0   0   0   0
1   1   1   0   0   0   0   0
1   1   0   0   0   0   0   0
1   0   0   0   0   0   0   0

```

```

0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow(abs(B2),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%15
coef = '15';
mask = [
1  1  1  1  1  0  0  0
1  1  1  1  0  0  0  0
1  1  1  0  0  0  0  0
1  1  0  0  0  0  0  0
1  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel

```

```

figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[]), colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%21
coef = '21';
mask = [
1 1 1 1 1 1 0 0
1 1 1 1 1 0 0 0
1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
1 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[]), colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%28
coef = '28';
mask = [
1 1 1 1 1 1 1 0
1 1 1 1 1 1 0 0
1 1 1 1 1 0 0 0
1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
1 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
];

```

```

mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%36
coef = '36';
mask = [
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0
1 1 1 1 1 1 0 0
1 1 1 1 1 0 0 0
1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
1 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[], colormap(jet(256)), colorbar;

```



```

saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%43
coef = '43';
mask = [
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0
1 1 1 1 1 1 0 0
1 1 1 1 1 0 0 0
1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
1 1 0 0 0 0 0 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow(abs(B2),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%49
coef = '49';
mask = [
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0
1 1 1 1 1 1 0 0
1 1 1 1 1 0 0 0
1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

```

```

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%54
coef = '54';
mask = [
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0
1 1 1 1 1 1 0 0
1 1 1 1 1 0 0 0
1 1 1 1 0 0 0 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

```

```

%58
coef = '58';
mask = [
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   0
1   1   1   1   1   1   0   0
1   1   1   1   1   0   0   0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow(abs(B2),[]), colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%61
coef = '61';
mask = [
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   0
1   1   1   1   1   1   0   0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2

```

```

I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%63
coef = '63';
mask = [
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

%64
coef = '64';
mask = [
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

```



```

1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1
];
mkdir(coef);
cd(coef);

%efarmogi maskas aporripsis sinteleston DCT
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);

%orismos sinartisis IDCT gia block 8x8
invdct = @(block_struct) T' * block_struct.data * T;

%anakataskevi eikonas me efarmogi IDCT ston B2
I2 = blockproc(B2,[8 8],invdct);

%emfanisi eikonas prin kai meta ton DCT
figure(2), imshow(I2)
imwrite(I2, '02 IMAGE AFTER DCT.tif');

%emfanisi timon entasis foteinotitas/pixel
figure(4), mesh(I2),colormap(jet(256)), colorbar
saveas(4, '04 FOTEINOTHTA DCT.tif');

%grafiki apeikonisi DCT synteleston meta tin aporripsi
figure(6), imshow((abs(B2)),[], colormap(jet(256)), colorbar;
saveas(6, '06 DCT COEFFICIENTS.tif');
close all;
cd ../

```


Βιβλιογραφικές Αναφορές

- Δήμου, Α. (2004). *Κωδικοποίηση εικόνων βάθους με το πρότυπο JPEG 2000*. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Θεσσαλονίκη.
- Δρε, Χ. Ν. (1995). *Νέοι αλγόριθμοι και αρχιτεκτονικές VLSI για συμπίεση εικόνων με χρήση μετασχηματισμών DCT, WAVELET και διανυσματικής κβάντισης*. Πανεπιστήμιο Πατρών, Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πάτρα.
- Καραθανάσης, Χ. (1993). *Αλγόριθμοι και αρχιτεκτονικές VLSI για συμπίεση ψηφιακών εικόνων*. Πανεπιστήμιο Πατρών, Πάτρα.
- Κολοβού, Α. (2012). *Εισαγωγή στην επεξεργασία εικόνας - Παραδείγματα με χρήση MATLAB*. Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Αθήνα.
- Λιωνής, Ι. Γ., & Ρέκκας, Σ. Μ. (2006). *Αλγόριθμοι συμπίεσης δεδομένων ολοκληρωτικής φωτογραφίας*. Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Αθήνα.
- Ράμμος, Δ. (2003). *Μέθοδοι αξιολόγησης και βελτίωσης εικόνας με μπλοκ μετασχηματισμό*. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Θεσσαλονίκη.
- Ροζάκης, Γ. Κ. (2010). *Εύρεση γεωμετρικών χαρακτηριστικών ερυθρών αιμοσφαιρίων από εικόνες σκεδασμένου φωτός από φωτοεκπέμπουσες διόδους (LEDs)*. Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πάτρα.
- Τατσάκη, Α. (1995). *Ανάπτυξη αποδοτικών αλγορίθμων DCT, DST, διανυσματικής κβάντισης πλέγματος και αρχιτεκτονικών VLSI για συμπίεση εικόνων*. Πανεπιστήμιο Πατρών, Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πάτρα.
- Mathworks.com, (2014). *Discrete Cosine Transform - MATLAB & Simulink*. [online] Available at: <http://www.mathworks.com/help/images/discrete-cosine-transform.html> [Accessed 16 Nov. 2014].
- Mathworks.com, (2014). *2-D discrete cosine transform - MATLAB dct2*. [online] Available at: <http://www.mathworks.com/help/images/ref/dct2.html> [Accessed 16 Nov. 2014].

- Mathworks.com, (2014). *Working with Images in MATLAB Graphics - MATLAB & Simulink*. [online] Available at: http://www.mathworks.com/help/matlab/creating_plots/working-with-images-in-matlab-graphics.html [Accessed 20 Nov. 2014].
- MATLAB?, H. (2013). *Image Processing Fundamentals, Basics of MATLAB and Embedded System Practicals on LPC2148....: How to apply DCT to Color Image & Grayscale Image in MATLAB?*. [online] Imagelpcmatlab.blogspot.gr. Available at: <http://imagelpcmatlab.blogspot.gr/2013/12/how-to-apply-dct-to-color-image.html> [Accessed 16 Nov. 2014].
- Whydomath.org, (2014). *Why Do Math?*. [online] Available at: <http://www.whymath.org/node/wavlets/dct.html> [Accessed 16 Nov. 2014].
- Whydomath.org, (2014). *Why Do Math?*. [online] Available at: <http://www.whymath.org/node/wavlets/basicjpg.html> [Accessed 16 Nov. 2014].
- Wikipedia, (2014). *JPEG*. [online] Available at: <http://en.wikipedia.org/wiki/JPEG> [Accessed 16 Nov. 2014].
- Wikipedia, (2014). *Discrete cosine transform*. [online] Available at: http://en.wikipedia.org/wiki/Discrete_cosine_transform [Accessed 16 Nov. 2014].