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

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ MATLAB/SIMULINK

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ:
ΑΝΤΖΟΥΛΑΤΟΣ ΑΛΕΞΑΝΔΡΟΣ
Α.Μ.: 2010007
ΕΠΙΒΛΕΠΩΝ:
ΒΑΣΙΛΕΙΟΣ ΜΠΟΖΑΝΤΖΗΣ

ΣΠΑΡΤΗ, ΑΠΡΙΛΙΟΣ 2017

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

"Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάση επιστημονικής παράφρασης.

Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων.

Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δε μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας."

Όνομα και Επώνυμο Συγγραφέα (Με Κεφαλαία): ΑΛΕΞΑΝΔΡΟΣ ΑΝΤΖΟΥΛΑΤΟΣ

Υπογραφή (Ολογράφως, χωρίς μονογραφή):

Ημερομηνία (Ημέρα – Μήνας – Έτος): ΔΕΥΤΕΡΑ 23 ΑΠΡΙΛΙΟΥ 2018

Περιεχόμενα

ΠΕΡΙΛΗΨΗ

Σε αυτήν την πτυχιακή δημιουργείται μία σειρά ασκήσεων για το αντικείμενο «Ψηφιακή Επεξεργασία Εικόνας». Αρχικά παρουσιάζονται στο Κεφάλαιο 1 εισαγωγικές έννοιες και βασικές συναρτήσεις εισόδου/εξόδου του πακέτου MATLAB. Στη συνέχεια στο Κεφάλαιο 2 παρουσιάζονται γεωμετρικοί μετασχηματισμοί, όπως αλλαγή μεγέθους, περιστροφή, περικοπή και κύρτωση εικόνας. Στο Κεφάλαιο 3 παρουσιάζεται η έννοια του ιστογράμματος εικόνας, οι μέθοδοι μέτρησης αντίθεσης (contrast) εικόνας και η βελτίωση της αντίθεσης εικόνας. Στο Κεφάλαιο 4 παρουσιάζεται το φιλτράρισμα εικόνας με φίλτρο μέσης τιμής (Mean-value filter), φίλτρο μεσαίας τιμής (Median filter) και Γκαουσιανό φίλτρο ομαλοποίησης. Στην συνέχεια στο Κεφάλαιο 5 παρουσιάζεται η μετάδοσης εικόνας με δισδιάστατο Μετασχηματισμό Συνημιτόνου (Discrete Cosine Transform) και στη συνέχεια κβάντιση μέσα από το ψηφιακό συμμετρικό κανάλι (Binary Symmetric Channel). Στο Κεφάλαιο 6 παρουσιάζονται τα συμπεράσματα των κεφαλαίων

Λέξεις-Κλειδιά: Επεξεργασία Εικόνας, Ιστόγραμμα, Φιλτράρισμα Εικόνας, Θόρυβος, Μετασχηματισμός Συνημιτόνου, Μετάδοση Εικόνας, Ψηφιακό Συμμετρικό Κανάλι.

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

1.1 Ψηφιακή Εικόνα

Η μετάβαση από τον αναλογικό στο ψηφιακό κόσμο συνεπάγεται τη μετατροπή των αναλογικών σημάτων σε ψηφιακά. Έτσι, το αναλογικό σήμα μιας εικόνας μεταφέρεται στον ψηφιακό κόσμο με τη μορφή διακεκριμένου σήματος, που έχει τη μορφή ψηφιακών πινάκων.

Μια ψηφιακή εικόνα μπορεί να είναι:

- δυαδική (binary image).

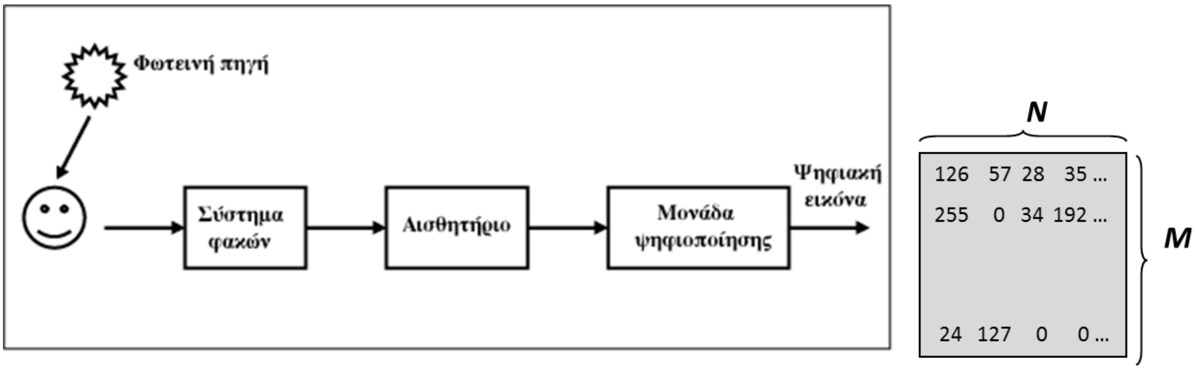
μονοχρωματική με αποχρώσεις του γκρι (gray level ή gray scale image).

- έγχρωμη (color image).

Σχήμα 1.1 (α) color image (β) gray scale (γ) binary image

1.2 Τυπικό σύστημα παραγωγής ψηφιακής εικόνας

Το αισθητήριο μετατρέπει το φως σε ηλεκτρικό ρεύμα στους άξονες (x,y), το οποίο δειγματοληπτείται, κβαντίζεται και ψηφιοποιείται μέσω ενός μετατροπέα A/D. Η ψηφιακή εικόνα (gray scale), η οποία παράγεται, συμβολίζεται με ένα δισδιάστατο πίνακα M × N pixels.



Σχήμα 1.2 Μπλοκ διάγραμμα σύλληψης εικόνας και μορφή ψηφιακής εικόνας.

Μια ψηφιακή εικόνα αποχρώσεων του γκρι, διαστάσεων M x N , παριστάνεται από έναν δισδιάστατο πίνακα ακεραίων αριθμών:

$$I(i,j) \text{ με } i = 1 \ldots M \text{ και } j = 1 \ldots N \qquad (1.1)$$

όπου $0 \le I(i,j) \le L - 1$. Το L ισούται συνήθως με μια δύναμη του 2, δηλαδή $L = 2^k$, με συνηθέστερη τιμή το $k = 8$, που αντιστοιχεί σε 256 αποχρώσεις του γκρι. Η τιμή $I(i,j)$ αντιπροσωπεύει τη φωτεινότητα του εικονοστοιχείου (pixel) (i,j).

Η απλούστερη μορφή μιας εικόνας είναι η δυαδική μορφή. Μια δυαδική εικόνα έχει μόνο δύο στάθμες φωτεινότητας, που συνήθως είναι το μαύρο και το άσπρο (L=2 και k=1). Το μαύρο αντιστοιχεί στην τιμή 0 και το άσπρο στην τιμή 1. Μια δυαδική εικόνα καταλαμβάνει μικρότερη μνήμη και η επεξεργασία της απαιτεί μικρότερο υπολογιστικό κόστος. Σε δυαδική μορφή μπορούν να απεικονισθούν σημαντικές πληροφορίες όπως είναι το εμβαδόν και η θέση των αντικειμένων, η μορφή αντικειμένων κ.α

Σημειώνεται ότι πάρα πολλές εφαρμογές της ΨΕΕ, όπως η οπτική αναγνώριση χαρακτήρων (OCR: Optical Character Recognition), η αναγνώριση υπογραφής (signature recognition), η αναγνώριση αποτυπωμάτων (fingerprint recognition), γίνονται συνήθως με τη χρήση των δυαδικών εικόνων.

Οι έγχρωμες ψηφιακές εικόνες αποτελούν το μέσο για την απεικόνιση του πραγματικού κόσμου. Μια έγχρωμη ψηφιακή εικόνα αποτελείται από τρεις gray scale εικόνες. Δηλαδή το χρώμα του κάθε εικονοστοιχείου έχει τρεις συνιστώσες που αντιστοιχούν στις αποχρώσεις των αντίστοιχων εικονοστοιχείων των τριών εικόνων. Μια ψηφιακή έγχρωμη εικόνα διαστάσεων M × N μπορεί να παρασταθεί ως:

$$I_c(i,j) \text{ με } i = 1 \ldots M \text{ και } j = 1 \ldots N \qquad (1.2)$$

όπου $0 \le I_c(i,j) \le L - 1$, για κάθε $c = 1, 2, 3$. Έτσι το χρώμα του κάθε pixel (i,j) προκύπτει από το συνδυασμό τριών χρωματικών συνιστωσών:

$$Color(i,j) = [\, I_1(i,j) ,\, I_2(i,j) ,\, I_3(i,j) \,] \qquad (1.3)$$

1.3. Το χρωματικό σύστημα RGB

Για παράδειγμα, στο χρωματικό σύστημα RGB, το κάθε χρώμα συντίθεται από τα χρώματα κόκκινο (Red), πράσινο (Green) και μπλε (Blue). Στο Σχήμα 1.3

παρουσιάζεται μια RGB εικόνα και η διάσπασή της στα τρία επιμέρους χρώματα.

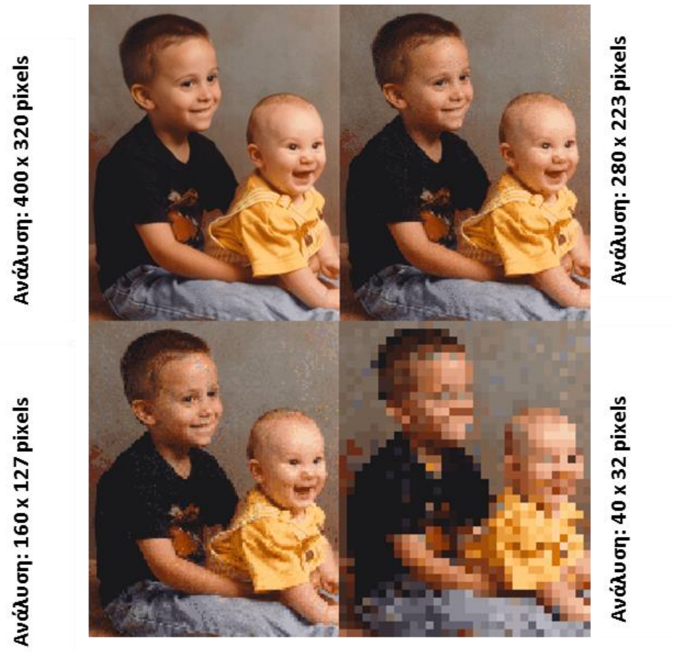
Σχήμα 1.3 Η αρχική έγχρωμη (RGB) εικόνα και οι τρεις συνιστώσες της red, green, blue.

Το k αποδίδει το πλήθος των αποχρώσεων και καλείται βάθος χρώματος (color depth). Μεγαλύτερο βάθος σημαίνει περισσότερες διαθέσιμες αποχρώσεις. Ο αριθμός των bits μιας εικόνας είναι b=M x N x k. Προφανώς μια έγχρωμη εικόνα απαιτεί τριπλάσιο αριθμό bits από μια gray scale εικόνα.

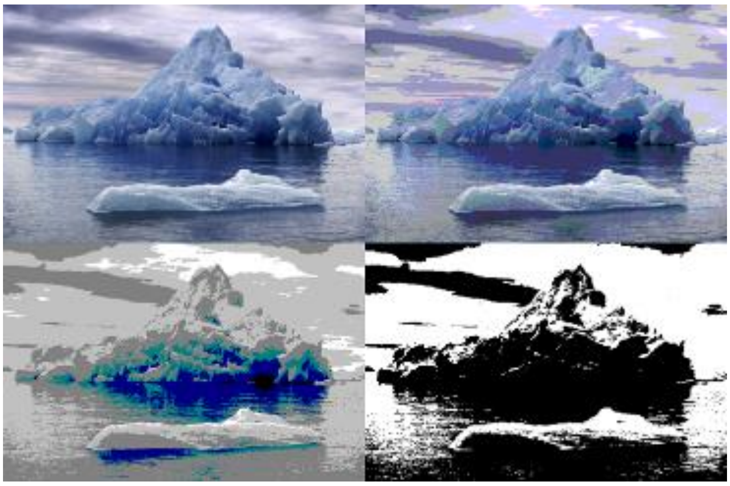
1.4 Ευκρίνεια εικόνας

Η ευκρίνεια μιας εικόνας καθορίζει το πόσο καλά μπορούμε να βλέπουμε τις λεπτομέρειές της. Ισούται με το πλήθος των pixels ανά μονάδα επιφανείας, και συνήθως μετριέται σε pixels / in^2 ή διαφορετικά σε d.p.i (dots per inch). Είναι φανερό ότι η ευκρίνεια εξαρτάται τόσο από τις διαστάσεις, όσο και από το πλήθος των αποχρώσεων της κάθε εικόνας. Αν για παράδειγμα κρατήσουμε σταθερό το βάθος χρώματος και μεταβάλλουμε (μειώνουμε) τις διαστάσεις μιας εικόνας, τότε η εικόνα θα εμφανίσει το φαινόμενο του σκακιού (checkboard). Δηλαδή η εικόνα θα κατατεμαχιστεί σε ομοιόμορφες χρωματικά περιοχές, όπως στο Σχήμα 1.4.

Αν από την άλλη πλευρά κρατήσουμε σταθερές τις διαστάσεις και μειώσουμε το βάθος χρώματος, τότε θα εμφανιστούν πάλι οπτικά ομοιόμορφες χρωματικές περιοχές, που όμως η ευκρίνειά τους θα καθορίζεται από το k και τη συγγένεια των τοπικών αποχρώσεων. Αυτό φαίνεται στο Σχήμα 1.5.



Σχήμα 1.4 Μεταβολή της ευκρίνειας κρατώντας σταθερό το k και μεταβάλλοντας τις διαστάσεις της εικόνας.



Σχήμα 1.5 Μεταβολή της ευκρίνειας κρατώντας σταθερές τις διαστάσεις της εικόνας και μεταβάλλοντας το k.

Εικόνα 1η : k=24 bit (16 εκατομμύρια χρώματα)

Εικόνα 2η : k=8 bit (256 χρώματα)

Εικόνα 3η : k=4 bit (16 χρώματα)

Εικόνα 4η : k=1 bit (2χρώματα)

1.5. Αντίθεση (contrast) εικόνας

Η μορφή της ψηφιακής εικόνας, μετά τη δειγματοληψία και την κβάντιση, είναι ένας πίνακας (matrix) πραγματικών αριθμών.

Αντίθεση ή contrast σε μια εικόνα είναι η κλίμακα από τη σκοτεινότερη στη φωτεινότερη περιοχή της και ορίζεται ως αντίθεση Michelson ή αντίθεση RMScotrast. Οι ορισμοί και περισσότερες πληροφορίες θα δοθούν στο αντίστοιχο κεφάλαιο.

1.6. Ποσοτικό κριτήριο ποιότητας ψηφιοποιημένης εικόνας

‘Εστω μία εικόνα MxN η οποία ψηφιοποιείται και κάθε εικονοστοιχείο της I(m,n) κβαντίζεται (στρογγυλοποιείται) από I(m,n) σε I’(m,n) κατά την ψηφιοποίηση της εικόνας. Το τετραγωνικό σφάλμα θα είναι τότε

$[I(m,n)-I'(m,n)]^2$

Το μέσο τετραγωνικό σφάλμα από την κβάντιση μίας MxN εικόνας θα είναι τότε

$$\sigma_e{}^2 = \frac{1}{MxN}\sum_{m=1}^{M}\sum_{n=1}^{N}(I(m,n) - I'(m,n))^2$$

Η ενέργεια της αρχικής εικόνας δίνεται από τον τύπο

$$\sigma^2 = \frac{1}{MxN}\sum_{m=1}^{M}\sum_{n=1}^{N} I^2(m,n)$$

Άρα η ποιότητα της εικόνας μετά την ψηφιοποίηση δίνεται από τον τύπο

$(\text{Signal to Noise Ratio})_{dB} = 10\log(\sigma^2/\sigma_e{}^2)$

Στην επεξεργασία εικόνας χρησιμοποιούμε και τον όρο Peak-Signal-to-Noise-Ratio (PSNR), όπου υποθέτουμε μέγιστη ενέργεια για κάθε εικονοστοιχείο, I(m,n) = 255, άρα:

$$\sigma^2 = \frac{1}{MxN}\sum_{m=1}^{M}\sum_{n=1}^{N} I^2(m,n) = 255 * 255$$

$(\text{PSNR})_{dB} = 10\log(255 * 255/\sigma_e{}^2)$

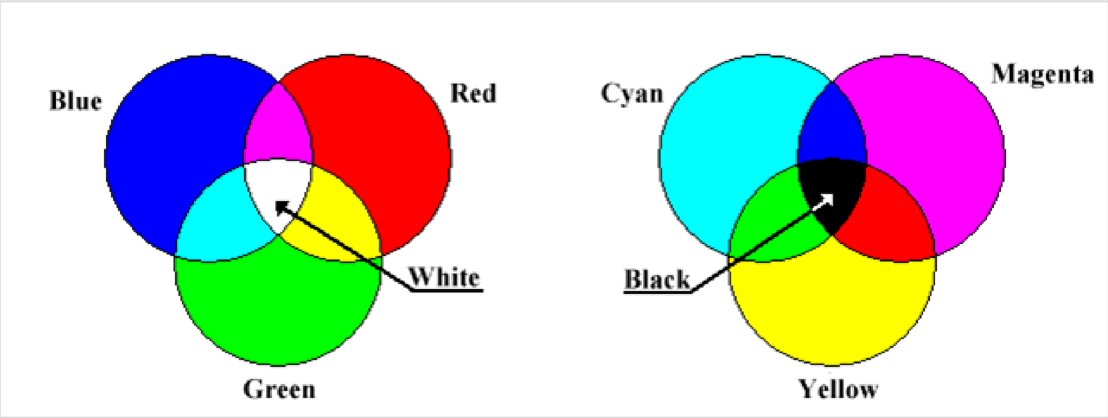
1.7 Χρωματικά συστήματα

Ο χαρακτηρισμός του φωτός είναι βασικός στην κατανόηση του χρώματος. Αν το φως είναι άχρωμο, τότε το μόνο χαρακτηριστικό του γνώρισμα είναι η ένταση (intensity). Ένα παράδειγμα αχρωματικής όψης είναι αυτή της ασπρόμαυρης τηλεόρασης όπου έχουμε απουσία χρώματος με εναλλαγή μόνο της ποσότητας φωτός. Με αυτόν τον τρόπο δημιουργούνται διαφορετικά επίπεδα του γκρι. Συνεπώς, ο όρος επίπεδο του γκρι αναφέρεται στη βαθμωτή μέτρηση της έντασης που διακυμαίνεται από το μαύρο στο γκρίζο και καταλήγει στο λευκό.

Οι τρεις βασικές ποσότητες που χρησιμοποιούνται για να περιγράψουν την ποιότητα της πηγής του χρωματικού φωτός είναι η ακτινοβολία (**radiance**) , η φωτεινότητα (**luminance**) και η λαμπρότητα (**brightness**).

Η ακτινοβολία είναι το τελικό ποσό ενέργειας που εκπέμπεται από την πηγή του φωτός, και μετριέται συνήθως σε Watt.

Τα βασικά χρώματα μπορούν να αναμιχθούν ώστε να παραχθούν τα δευτερεύοντα χρώματα (secondary colors). Αυτά είναι τα: ματζέντα (κόκκινο και μπλε), κυανό (πράσινο και μπλε), κίτρινο (κόκκινο και πράσινο). Ο συνδυασμός των τριών βασικών χρωμάτων ή ενός δευτερεύοντος με το αντίθετο βασικό, παράγουν το λευκό χρώμα αν αναμιχθούν με συγκεκριμένες εντάσεις φωτός. Τα αποτελέσματα αυτά φαίνονται στο παρακάτω Σχήμα



Σχήμα 1.6 Συνδυασμοί χρωμάτων

Στην πτυχιακή αυτή χρησιμοποιούμε το μοντέλο RGB. Το μοντέλο αυτό απλοποιεί αρκετά την σχεδίαση γραφικών συστημάτων, αλλά δημιουργεί δυσκολίες στην ανάπτυξη αλγορίθμων λόγω της συσχέτισης των χρωματικών συνιστωσών του. Για παράδειγμα η εξισορρόπηση ιστογράμματος λειτουργεί μόνο επί της φωτεινότητας μίας εικόνας και πραγματοποιείται πιο εύκολα στο μοντέλο HSI. Έτσι πολλές φορές είναι η αναγκαία η μετατροπή ενός χρωματικού μοντέλου σε ένα άλλο. Χρησιμοποιείται συνήθως σε έγχρωμες οθόνες και κάμερες. Η μετατροπή RGB σε κλίμακα γκρι είναι ο μέσος όρος των τριών χρωματικών συνιστωσών:

Gray=0.333R + 0.333G + 0.333B

1.8. Αναπαράσταση ψηφιακής πληροφορίας εικόνας στο MATLAB

Οι τύποι εικόνων που μπορεί να αναπαραστήσει το MATLAB είναι:

(α) Εικόνες φωτεινότητας (Intensity images) όπου κάθε εικονοστοιχείο είναι μεταξύ 0 και 255 (για τύπο δεδομένων uint8) ή 0 και 65535 (uint16)

(β) Δυαδικές εικόνες στις οποίες κάθε εικονοστοιχείο έχει τιμή 0 ή 1

(γ) Εικόνες έγχρωμες τύπου RGB, με διαστάσεις M(σειρές)xN(στήλες)x3, όπου το 3 σημαίνει ότι ο κάθε πίνακας MxN έχει τιμές για κάθε ένα από τα χρώματα (R,G,B). Οι τιμές μπορεί να είναι double, uint8 ή uint16.

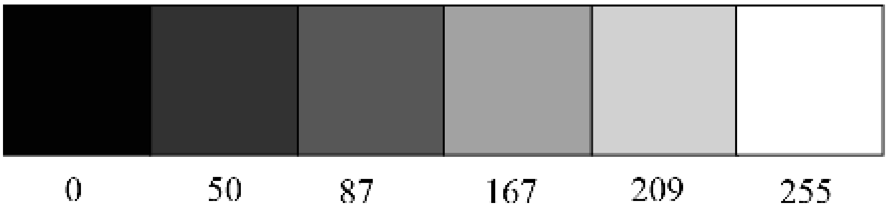
(δ) Εικόνες με χρωματικό δείκτη, όπου ο πίνακας με τιμές φωτεινότητας 0…(2^{m-1}-1) της κατηγορίας (α) συνοδεύεται από τον χρωματικό πίνακα διαστάσεων 2^{m-1}x3, ο οποίος δείχνει για κάθε τιμή φωτεινότητας τις τιμές των R,G,B (τις τιμές των χρωμάτων).

1.9. Βασικές συναρτήσεις εισόδου/εξόδου στο MATLAB

1.9.1 Η συνάρτηση imread()

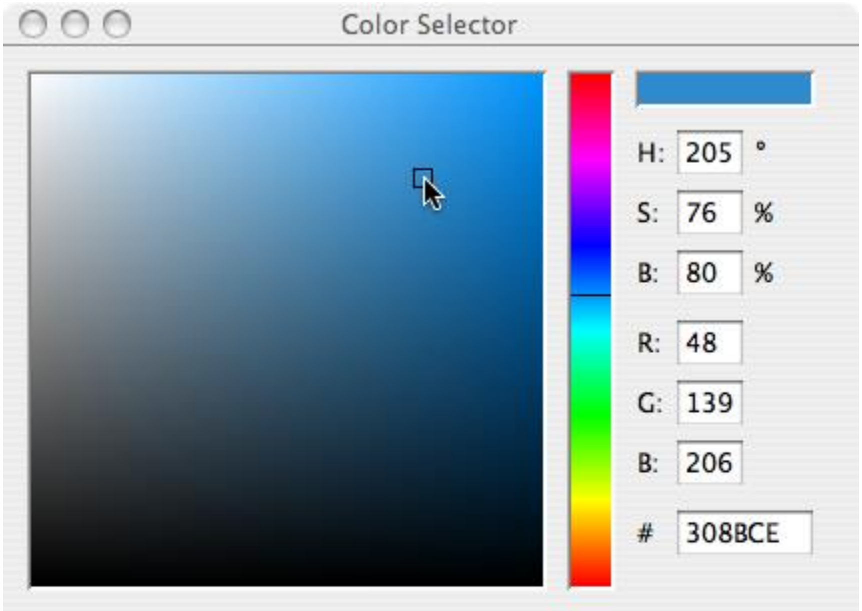
Η συνάρτηση imread() διαβάζει την εικόνα και την αποθηκεύει σε έναν πίνακα g.

Αν πρόκειται για ασπρόμαυρη εικόνα, τότε ο πίνακας g θα έχει 2 διαστάσεις: ο αριθμός των σειρών και στηλών του θα ισούται με τον αριθμό σειρών και στηλών εικονοστοιχείων της εικόνας. Άρα ο πίνακας θα έχει τόσα στοιχεία όσα και τα εικονοστοιχεία της εικόνας. Η τιμή κάθε στοιχείου θα είναι 0…255 σύμφωνα με το παρακάτω:



Σχήμα 1.7 Αντιστοίχιση τιμών ασπρόμαυρου εικονοστοιχείου

Αν πρόκειται για έγχρωμη εικόνα ο πίνακας g θα είναι τριδιάστατος. Οι δύο πρώτες διαστάσεις (σειρά,στήλη) πάλι θα δείχνουν την θέση του εικονοστοιχείου στην εικόνα, Η τρίτη διάσταση θα μας δείχνει την απόχρωση του εικονοστοιχείου στο RGB.



Σχήμα 1.8 Αποχρώσεις έγχρωμου εικονοστοιχείου

Στο ανωτέρω σχήμα, η απόχρωση του επιλεγμένου εικονοστοιχείου θα δίνεται από το RGB [48 139 206]. Άρα ο πίνακας g θα έχει διαστάσεις ΣειρέςxΣτήλεςx3.

1.9.2. Η συνάρτηση imwrite()

Η συνάρτηση imwrite() εγγράφει το περιεχόμενο πίνακα ή πινάκων σε ένα αρχείο εικόνας. Πιο συγκεκριμένα:

- imwrite(A,filename): εγγράφει τα δεδομένα εικόνας A στο αρχείο filename

- imwrite(A,map,filename): εγγραφή για την περίπτωση (δ) της 1.8, όπου χρειάζονται τα δεδομένα του πίνακα φωτεινότητας A και ο χρωματικός πίνακας map

1.9.3. Η συνάρτηση imshow()

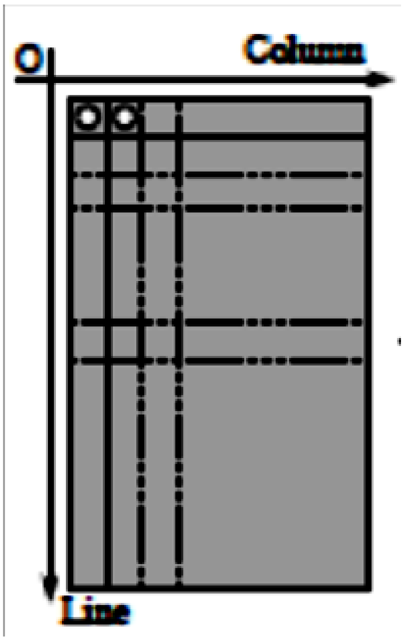
Με την συνάρτηση αυτή παρουσιάζεται μία εικόνα με βάση τα δεδομένα πινάκων ή ενός αρχείου. Πιο συγκεκριμένα:

- imshow(I): παρουσιάζει την εικόνα με βάση τα δεδομένα ενός πίνακα I (εικόνας greyscale, RGB ή δυαδικής)
- imshow(X,map):παρουσιάζει την εικόνα για την περίπτωση (δ) της 1.8, όπου χρειάζονται τα δεδομένα του πίνακα φωτεινότητας A και ο χρωματικός πίνακας map
- imshow(X ή filename, Name,Value): τυπώνει μία εικόνα με βάση τα δεδομένα πινάκων ή ενός αρχείου με κάποια επιλογή, π.χ. imshow(g,'Border','tight') παρουσιάζεται η εικόνα χωρίς περιθώρια (επιλογή Border tight).

ΚΕΦ 2. ΓΕΩΜΕΤΡΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ

2.1 Εισαγωγικά

Σε αυτό το κεφάλαιο θα παρουσιαστούν ασκήσεις που αλλάζουν τα γεωμετρικά χαρακτηριστικά μίας εικόνας. Οι απλοί γεωμετρικοί μετασχηματισμοί, όπως οι μετατοπίσεις και οι παραμορφώσεις είναι προβληματικοί λόγω της ακεραίας φύσης των αριθμών και της θέσης των εικονοστοιχείων σε μια εικόνα. Η αρχή των αξόνων θεωρείται ότι είναι στην πάνω δεξιά άκρη της εικόνας



Σχήμα 2.1 Η εικόνα στον δισδιάστατο χώρο

2.2 Αλλαγή μεγέθους εικόνας

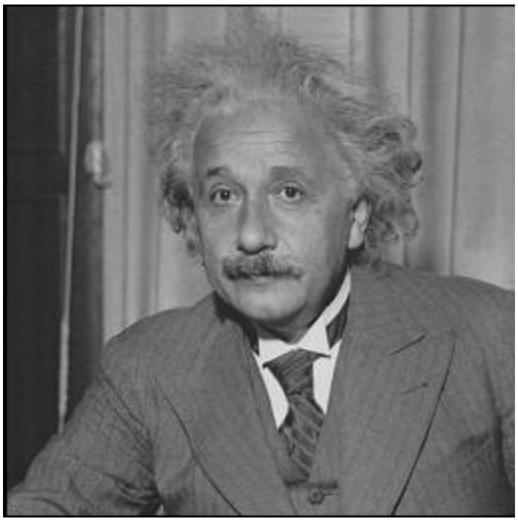
Χρησιμοποιείται η συνάρτηση imresize() για την αλλαγή του μεγέθους της εικόνας. Με το παρακάτω πρόγραμμα αλλάζει το μέγεθος της εικόνας.

```
function allagi_megethous(image,upsos,platos)

% image: eikona
% upsos,platos: kainourgies diastaseis
%'Border','tight': emfanisi eikonwn xwris perithwrio
% figure: krataei tin eikona pou edeixe sti proigoumeni grammi

g = imread(image);
```

```
n = imresize(g,[upsos platos]);
imshow(g,'Border','tight');
figure;
imshow(n,'Border','tight');
```

Οι είσοδοι της συνάρτησης είναι το αρχείο της εικόνας (image), και το επιθυμητό ύψος (upsos) και πλάτος (platos) μετά τον μετασχηματισμό.

Παράδειγμα 1



Σχήμα 2.2. Ανάγνωση ασπρόμαυρης εικόνας

```
g=imread('einstein.jpg');
>> size(g)
ans =
  256   256
```

Παράδειγμα 2

Σχήμα 2.3. Ανάγνωση έγχρωμης εικόνας

```
>> g=imread('lena.bmp');
>> size(g)
ans =
   512   512     3
```

Στην συνέχεια με την συνάρτηση imresize(g,[upsos platos]) αλλάζει ο πίνακας g στο επιθυμητό ύψος και πλάτος.

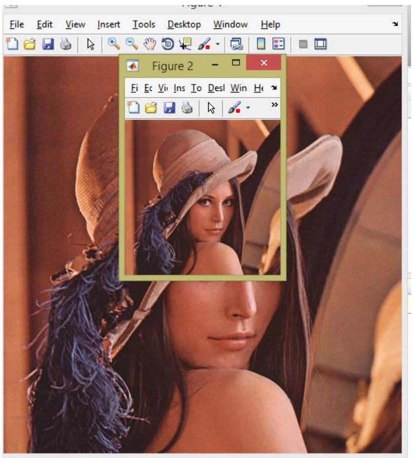
Παράδειγμα 3

allagi_megethous('einstein.jpg',90,90)

Σχήμα 2.4. Αλλαγή μεγέθους ασπρόμαυρης εικόνας

Παράδειγμα 4

allagi_megethous('lena.bmp',200,200)



Σχήμα 2.5. Αλλαγή μεγέθους έγχρωμης εικόνας

2.3. Περιστροφή εικόνας

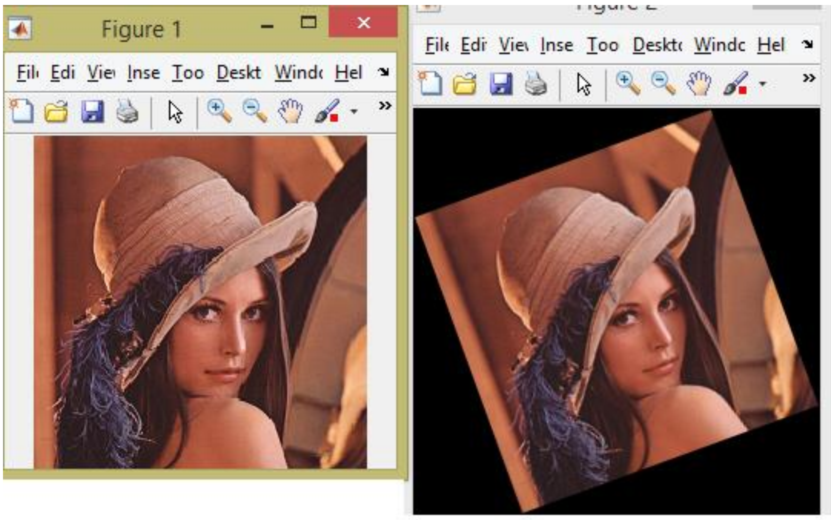
Η περιστροφή της εικόνας γίνεται με το παρακάτω πρόγραμμα

```
function peristrofi_eikonas(image,gonia,algorithmos)
% image: eikona
% gonia: gonia peristrofis
% algorithmos: algorithmos peristrofis, px 'bilinear'
%'Border','tight': emfanisi eikonwn xwris perithwrio
% figure: krataei tin eikona pou edeixe sti proigoumeni grammi
g = imread(image);
n = imrotate(g,gonia,algorithmos);
imshow(g,'Border','tight');
figure;
imshow(n,'Border','tight');
```

Η συνάρτηση που χρησιμοποιούμε είναι η imrotate(A,angle,method), και σε αυτήν A είναι η εικόνα, angle η γωνία και method η μέθοδος. Η μέθοδος είναι:
'nearest' για παρεμβολή κοντινότερου γείτονα (Nearest-neighbor interpolation)
'bilinear' για διγραμμική παρεμβολή (Bilinear interpolation)

Παράδειγμα 5

peristrofi_eikonas('lena_mikri.png',20,'bilinear')



Σχήμα 2.6. Περιστροφή έγχρωμης εικόνας

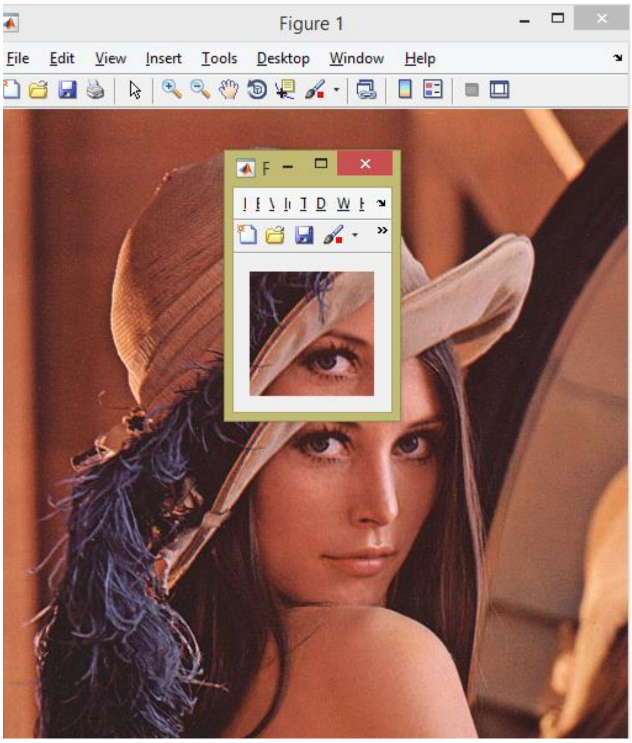
2.4. Περικοπή εικόνας

Με το πρόγραμμα αυτό μπορεί να περικοπεί ένα τμήμα της εικόνας. Αρχικά επιλέγεται ένα σημείο αναφοράς του τμήματος που θέλουμε να κόψουμε, οι συντεταγμένες είναι xmin,ymin. Αυτό το σημείο θα είναι η αρχή των αξόνων πάνω αριστερά στην αρχική εικόνα. Στην συνέχεια επιλέγουμε το πλάτος και το ύψος του κομματιού που θέλουμε να κόψουμε και είναι οι μεταβλητές platos και upsos.

```
function perikopi_eikonas(image,xmin,ymin,platos,upsos)
% image: eikona
% xmin ymin: suntetagmenes simeiou anaforas tou kommatiou pou kovoume
% (H arxi twn axonwn einai panw aristera stin arxiki eikona)
% platos, upsos: diastaseis tou kommatiou pou kovoume
%'Border','tight': emfanisi eikonwn xwris perithwrio
% figure: krataei tin eikona pou edeixe sti proigoumeni grammi
g = imread(image);
n = imcrop(g,[xmin ymin platos upsos]);
imshow(g,'Border','tight');
figure;
imshow(n,'Border','tight');
```

Παράδειγμα 6

>> perikopi_eikonas('lena.bmp',200,200,100,100)



Σχήμα 2.7 Περικοπή τμήματος εικόνας

2.5 Κύρτωση εικόνας

Η συνάρτηση κύρτωση εικόνας δέχεται ως εισόδους την εικόνα και 2 πίνακες U,X.:

```
function kurtosi_eikonas(image,U,X)

% typos metasximatismou:  'projective'   (metasximatismos provolis)
% U,X: Gia 'projective' einai pinakes 4x2 kai periexoun suntetagmenes
korifwn 2 tetrapleurwn.
% U: arxikes suntetagmenes eikonas
% X: telikes suntetagmenes kurwmenis eikonas
% imread: sunartisi anagnwsis eikonas kai apothikeusis se pinaka I

I=imread(image);

udata = [0 1];

vdata = [0 1];

tform = maketform('projective',U,X);

[B,xdata,ydata] = imtransform(I, tform, 'bicubic','udata',
udata,'vdata', vdata, 'size', size(I),'fill', 128);


imshow(I,'XData',udata,'YData',vdata);
axis on;
figure;

imshow(B,'XData',xdata,'YData',ydata);
axis on;
```

Για την περίπτωση του τύπου μετασχηματισμού προβολής (projective) που μελετάται εδώ, οι πίνακας U,X είναι διαστάσεων 4x2 και περιέχουν τις συντεταγμένες αντίστοιχα του τετραπλεύρου της αρχικής και της κυρτωμένης εικόνας.

Η εικόνα διαβάζεται με την συνάρτηση I=imread(image); και αποθηκεύεται στον πίνακα I.

Στην συνέχεια ορίζονται οι βοηθητικές μεταβλητές udata=[0 1] και vdata=[0 1]. Στην συνέχεια ορίζεται η δομή tform με την βοήθεια της συνάρτησης maketform('projective',U,X), στην οποία καθορίζεται ο τύπος του μετασχηματισμού κύρτωσης (‘projective’). Ο μετασχηματισμός που θέλουμε να υλοποιήσουμε είναι τώρα αποθηκευμένος στην δομή tform (μπορούμε να το φανταστούμε ως ένα σύστημα που προκαλεί κύρτωση εικόνας και έχει την ονομασία tform). Μία δομή στο MATLAB δημιουργείται με διάφορες συναρτήσεις και είναι ακριβώς όπως οι δομές δεδομένων σε όλες τις γλώσσες προγραμματισμού.

Τέλος με την συνάρτηση [B,xdata,ydata] = imtransform(I, tform, 'bicubic','udata', udata,'vdata', vdata, 'size', size(I),'fill', 128); δημιουργείται ο πίνακας της κυρτωμένης εικόνας X, καθώς και τα νέα διαστήματα συντεταγμένων xdata και tdata που χρειάζονται για να απεικονιστούν και οι δύο εικόνες με τις συναρτήσεις:

imshow(I,'XData',udata,'YData',vdata);

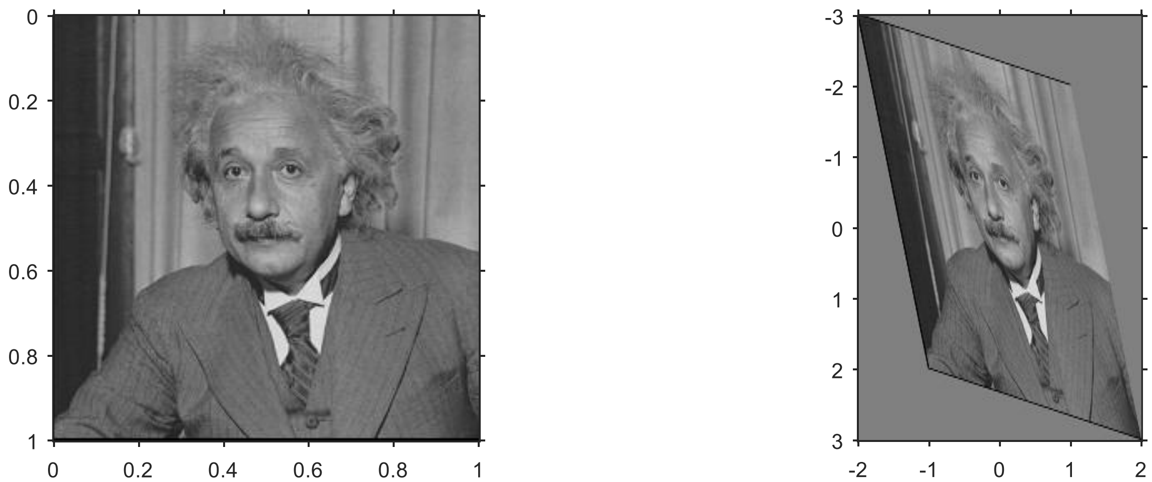
imshow(B,'XData',xdata,'YData',ydata);

Παράδειγμα με κύρτωση ασπρόμαυρης εικόνας

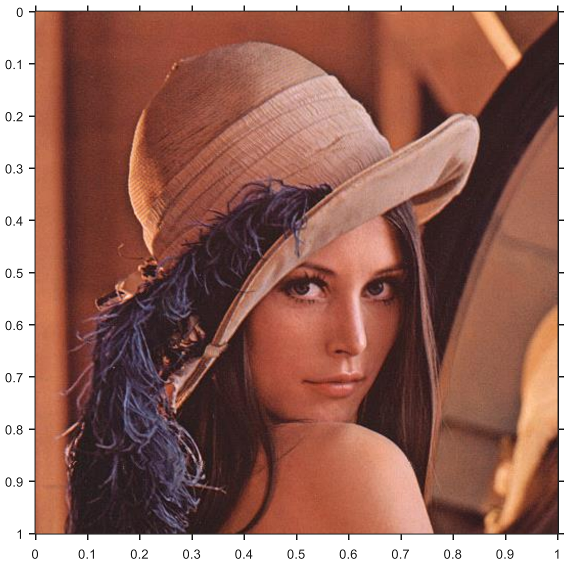
U=[0 0; 1 0; 1 1; 0 1];

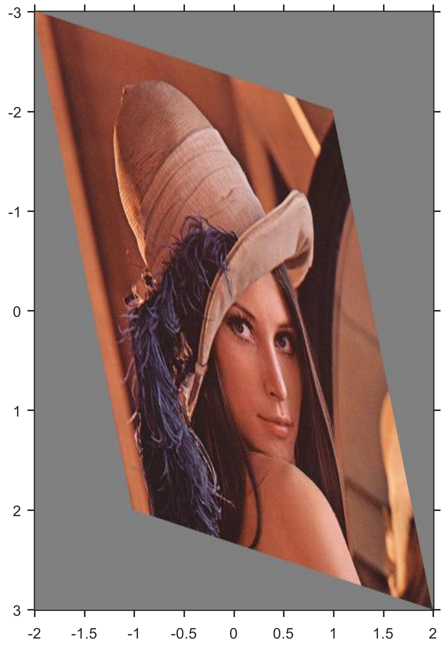
X=[-2 -3; 1 -2; 2 3; -1 2];

kurtosi_eikonas('einstein.jpg',U,X);



Σχήμα 2.8 Κύρτωση ασπόμαυρης εικόνας





Σχήμα 2.9 Κύρτωση έγχρωμης εικόνας

ΚΕΦΑΛΑΙΟ 3. ΒΕΛΤΙΩΣΗ ΑΝΤΙΘΕΣΗΣ

3.1. Ιστογράμματα

Το ιστόγραμμα μιας ψηφιακής εικόνας (gray scale για χάριν απλότητας) με L επίπεδα φωτεινότητας στην περιοχή [0 255] ορίζεται σαν την ακόλουθη διακριτή συνάρτηση:

$h(r_k) = n_k$ (3.1)

Όπου r_k είναι το k επίπεδο φωτεινότητας στο διάστημα [0 255] και n_k είναι ο αριθμός των pixels στην εικόνα που έχουν επίπεδο φωτεινότητας r_k.

Πολλές φορές χρησιμοποιείται το κανονικοποιημένο ιστόγραμμα (normalized histogram) το οποίο ευρίσκεται διαιρώντας τη σχέση (3.1) με το συνολικό αριθμό των εικονοστοιχείων στην εικόνα:

$p(r_k) = \frac{h(r_k)}{n}$ (3.2)

όπου n είναι ο συνολικός αριθμός των εικονοστοιχείων της εικόνας και k = 1,2,…,L. Το $p(r_k)$ αντιπροσωπεύει την πιθανότητα εμφάνισης του επιπέδου φωτεινότητας r_k. Η συνάρτηση του Matlab για την εύρεση του ιστογράμματος είναι:

h = imhist (I,b)

Όπου I είναι η εικόνα εισόδου και b είναι ο αριθμός των βημάτων (bins), ο οποίος είναι ίσος με 256 εάν δεν ορίζεται. Το κανονικοποιημένο ιστόγραμμα δημιουργείται με τη σχέση:

p = imhist (I,b) / numel (I)

Η συνάρτηση numel (I) υπολογίζει τον αριθμό των εικονοστοιχείων της εικόνας εισόδου. Η απεικόνιση του ιστογράμματος στην οθόνη γίνεται μόνο με τη συνάρτηση imhist (I,b).

3.2 Μέτρηση αντίθεσης εικόνας με την μέθοδο Michelson

Αν υποθέσουμε ότι ο πίνακας των εικονοστοιχείων μίας εικόνας είναι I ,
τότε η αντίθεση (contrast) κατά Michelson ορίζεται ως εξής:

$C = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$ (3.3)

3.3 Μέτρηση αντίθεσης εικόνας κατά RMScontrast

Αν υποθέσουμε ότι ο πίνακας των εικονοστοιχείων μίας εικόνας είναι I ,

τότε η αντίθεση (contrast) κατά RMScontrast ορίζεται ως εξής:

$$C = \sqrt{\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{N}(\ I_{ij} - \bar{I}\ )^2} \qquad (3.4)$$

όπου $\bar{I}$ είναι η μέση τιμή όλων των στοιχείων του πίνακα I().

Υπολογίζουμε επομένως τις αποκλίσεις $(\ I_{ij} - \bar{I}\ )^2$ όλων των στοιχείων, τις αθροίζουμε, και υπολογίζουμε την τετραγωνική ρίζα του πηλίκου του αθροίσματος διά τις διαστάσεις του πίνακα (τυπική απόκλιση).

3.4 Βελτίωση αντίθεσης

Η βελτίωση της αντίθεσης γίνεται με την μέθοδο της εξισορρόπησης ιστογράμματος. Η μέθοδος αυτή γίνεται με την συνάρτηση equalised = histeq(g);

Η συνάρτηση αυτή δέχεται ως είσοδο τον πίνακα g και επιστρέφει τον πίνακα equalised της εικόνας με εξισορροπημένο ιστόγραμμα. Το πρόγραμμα που γράφηκε για την βελτίωση της αντίθεσης είναι το ακόλουθο:

```
function
[g,equalised,Michelson_arxikis,Michelson_exisoroppimenis,RMScontrast_
arxiki,RMScontrast_exisoropimeni]=istogrammata(image);
% image: eikona

g = imread(image);
imshow(g,'Border','tight');
figure;
imhist(g);
figure;
% Auxisi tis antithesis me exisoropisi tou istogrammatos
equalised = histeq(g);
imshow(equalised,'Border','tight');
figure;
imhist(equalised);


% Ypologismos antithesis Michelson arxikis eikonas
Michelson_arxikis = (max(g(:))-min(g(:)))/(max(g(:))+min(g(:)));
% Ypologismos antithesis Michelson exisoroppimenis eikonas
Michelson_exisoroppimenis = (max(equalised(:))-
min(equalised(:)))/(max(equalised(:))+min(equalised(:)));

% Ypologismos antithesis RMScontrast gia arxiki eikona
g_mesi=mean(g(:));
tetragwna=(g-g_mesi).^2;
```

```
RMScontrast_arxiki = sqrt(sum(tetragwna(:))/(size(g,1)*size(g,2)));

% Ypologismos antithesis RMScontrast gia exisoropimeni eikona
equalised_mesi=mean(equalised(:));
tetragwna=(equalised-equalised_mesi).^2;
RMScontrast_exisoropimeni =
sqrt(sum(tetragwna(:))/(size(equalised,1)*size(equalised,2)));
```

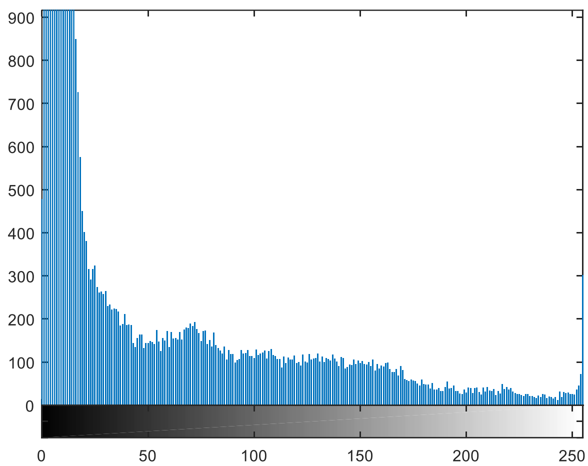
3.5. Αποτελέσματα

Θα χρησιμοποιήσουμε την τεχνική αυτή για εξισορρόπηση ασπρόμαυρης εικόνας. Η εικόνα εισόδου, η βελτιωμένη εικόνα εξόδου και το ιστόγραμμα φαίνονται παρακάτω.

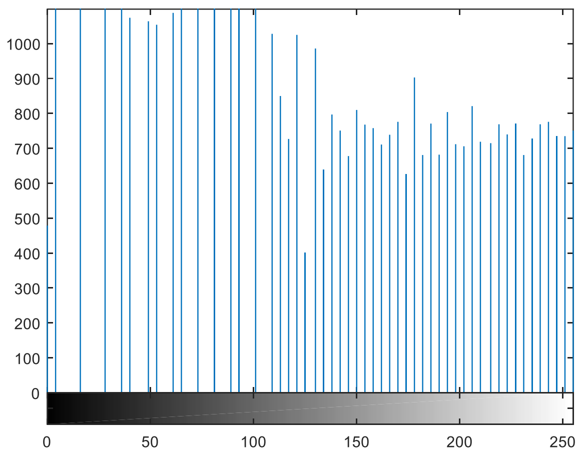
(α)

(β)

Σχήμα 3.1. Αρχική (α) και εξισορροπημένη εικόνα



(α)



(β)

Σχήμα 3.2. Ιστόγραμμα αρχικής (α) και εξισορροπημένης (β) εικόνας

Από τα αποτελέσματα της εκτέλεσης θα δούμε ότι στο εξισορροπημένο ιστόγραμμα οι τιμές δεν συγκεντρώνονται γύρω από κάποιο συγκεκριμένο σημείο αλλά διασπείρονται σε όλο το φάσμα τιμών [0,255]. Επομένως στην εξισορροπημένη εικόνα η αντίθεση της εικόνας είναι ισοσκελισμένη.

Σε αυτό το παράδειγμα υπολογίστηκε:

RMScontrast_arxiki = 9.0292

RMScontrast_exisoropimeni =10.8046

ΚΕΦΑΛΑΙΟ 4. ΦΙΛΤΡΑΡΙΣΜΑ ΕΙΚΟΝΑΣ

4.1 Εισαγωγικά

Ένα από τα πιο συνηθισμένα προβλήματα στην ψηφιακή επεξεργασία εικόνας είναι η βελτίωση της ποιότητας της εικόνας με την αφαίρεση του θορύβου από αυτήν. Η εφαρμογή ενός φίλτρου αφαίρεσης του θορύβου είναι από τις βασικές τεχνικές επεξεργασίας εικόνας. Στην πτυχιακή αυτή θα εξεταστούν το φίλτρο μέσης τιμής, το φίλτρο μεσαίας τιμής, φιλτράρισμα Gauss και φιλτράρισμα με χρήση του διακριτού μετασχηματισμού Fourier DFT.

4.2 Θόρυβος

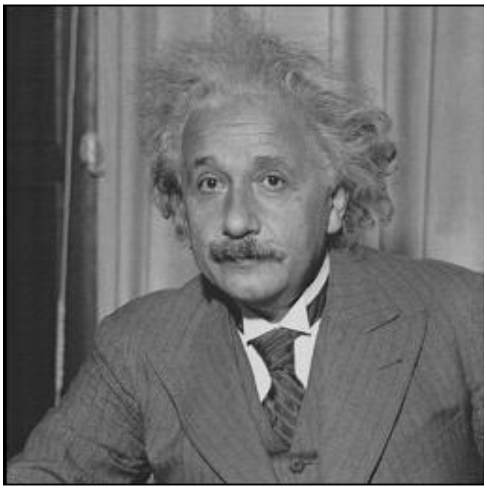
Στην πτυχιακή αυτή θα ασχοληθούμε με 2 είδη θορύβου εικόνας, τον θόρυβο 'Gauss' και τον θόρυβο 'Salt & Pepper'. Παρακάτω παραθέτουμε παράδειγμα ασπρόμαυρης και έγχρωμης εικόνας και το αποτέλεσμα των 2 θορύβων σε κάθε μία. Η αλλοιωμένη με θόρυβο εικόνα προκύπτει με την εφαρμογή της συνάρτησης imnoise():

- Η συνάρτηση image_w_noise=imnoise(I,'salt & pepper',0.03); παράγει από την αρχική εικόνα I την αλλοιωμένη με θόρυβο 'salt & pepper' εικόνα image_w_noise. Η παράμετρος d=0.03 είναι η φασματική πυκνότητα θορύβου
- Η συνάρτηση image_w_noise=imnoise(I,'gaussian',0,0.02); παράγει από την αρχική εικόνα I την αλλοιωμένη με θόρυβο 'gaussian' εικόνα image_w_noise. Η παράμετρος μ=0 είναι η μέση τιμή και η παράμετρος $σ^2$=0.02 είναι η διασπορά του θορύβου.

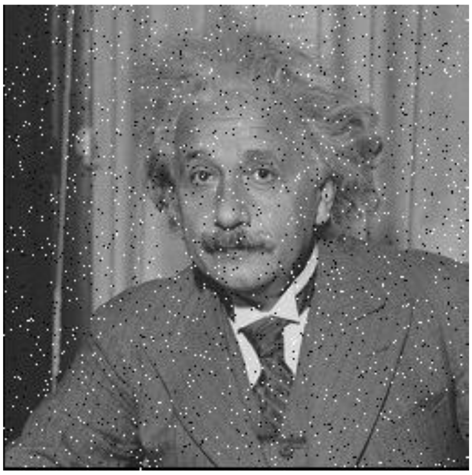
Παρακάτω παραθέτουμε παραδείγματα της επίδρασης του θορύβου σε ασπρόμαυρες και έγχρωμες εικόνες.

<u>Παραδείγματα Lena και Einstein με την επίδραση δύο ειδών θορύβου</u>

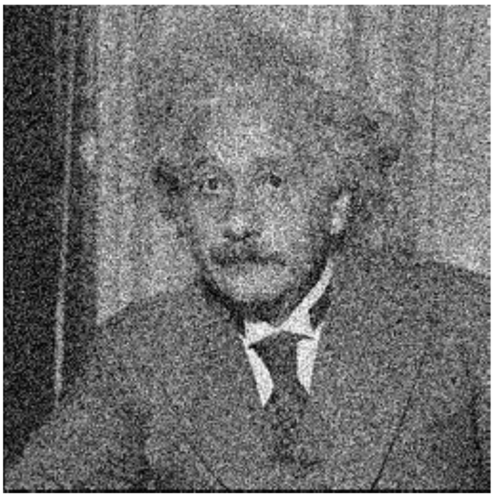
(1) Ασπρόμαυρη εικόνα Einstein.jpg



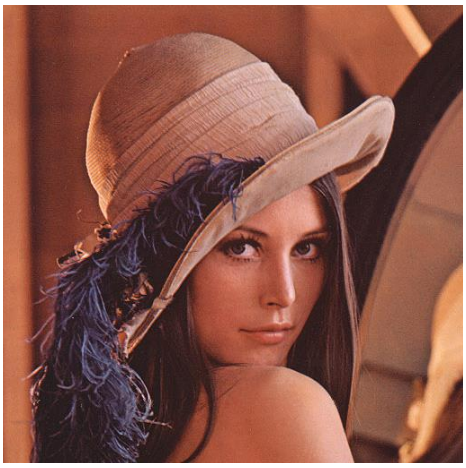
```
>> I=imread('einstein.jpg');
>> image_w_noise=imnoise(I,'salt & pepper',0.03);
>> imshow(image_w_noise,'Border','tight')
```



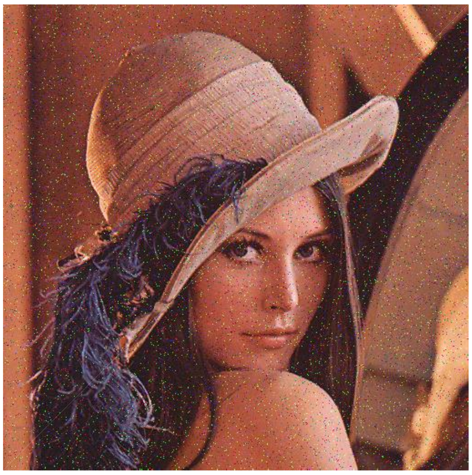
```
>> I=imread('einstein.jpg');
>> image_w_noise=imnoise(I,'gaussian',0,0.02);
>> imshow(image_w_noise,'Border','tight')
```



(2) Έγχρωμη εικόνα Einstein.jpg



```
>> I=imread('lena.bmp');
>> image_w_noise=imnoise(I,'salt & pepper',0.03);
>> imshow(image_w_noise,'Border','tight')
```



```
>> I=imread('lena.bmp');
>> image_w_noise=imnoise(I,'gaussian',0,0.02);
>> imshow(image_w_noise,'Border','tight')
```

4.3 Φίλτρο μέσης τιμής (Mean-value filter)

Το φίλτρο μέσης τιμής, είναι μια απλή τεχνική εξάλειψης και εξομάλυνσης του θορύβου από ψηφιακές εικόνες. Σύμφωνα με την τεχνική αυτή από την αρχική εικόνα παράγεται μία νέα εικόνα ιδίων διαστάσεων όπου κάθε εικονοστοιχείο της έχει φωτεινότητα τη μέση τιμή των τιμών φωτεινότητας μιας γειτονιάς του αντίστοιχου εικονοστοιχείου της αρχικής εικόνας.

Το φίλτρο μέσης τιμής είναι ουσιαστικά ένα βαθυπερατό φίλτρο. Το αποτέλεσμα είναι πολύ ικανοποιητικό όσον αφορά την απαλλαγή από τον θόρυβο αλλά ταυτόχρονα δημιουργείται μια θαμπάδα στην εικόνα, όπως θα διαπιστώσουμε από τα παραδείγματα που θα επιδειχθούν.

Αν υποθέσουμε ότι η αρχική εικόνα είναι ο πίνακας f() (2 διαστάσεων για ασπρόμαυρη ή 3 διαστάσεων για έγχρωμη), τότε η φιλτραρισμένη εικόνα είναι ο πίνακας g. Στις συντεταγμένες (x,y) αντιστοιχεί το εικονοστοιχείο f(x,y) της αρχικής εικόνας και το εικονοστοιχείο της φιλτραρισμένης εικονας:

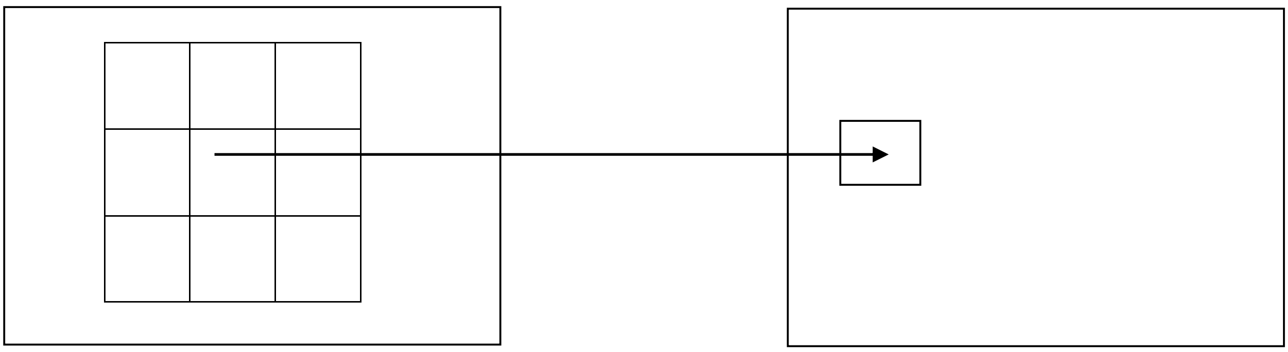
$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x-i, y-j) * \omega(i,j) \quad (4.1)$$

Η μάσκα είναι ο τετραγωνικός πίνακας 3x3 ω που είναι της μορφής:

$$\omega = \begin{matrix} \omega(-1,-1) & \omega(-1,0) & \omega(-1,1) \\ \omega(0,-1) & \omega(0,0) & \omega(0,1) \\ \omega(1,-1) & \omega(1,0) & \omega(1,1) \end{matrix} \quad (4.2)$$

Όταν χρησιμοποιείται γραμμικό φίλτρο μέσου όρου NxN, ισχύει $\omega_{i,j} = \frac{1}{N*N}$, άρα στην συγκεκριμένη περίπτωση $\omega_{i,j} = \frac{1}{9}$, i,j=-1,0,1.

Με αυτόν τον τρόπο το εικονοστοιχείο g(x,y) είναι ο μέσος όρος (για αυτό και λέγεται φίλτρο μέσου όρου) των στοιχείων της αρχικής εικόνας που βρίσκονται σε απόσταση μέχρι 1 εικονοστοιχείο γύρω από το f(x,y):



Αρχική Εικόνα f(x,y) Φιλτραρισμένη Εικόνα g(x,y)

Στο παραπάνω σχήμα ο υπολογισμός του μέσου όρου g(x,y) έγινε από ένα τετράγωνο 9 στοιχείων, στο κέντρο του οποίου είναι το f(x,y). Θα μπορούσαμε να πάρουμε ένα τετράγωνο 15 στοιχείων, όποτε και $\omega_{i,j} = \frac{1}{15}$, i,j=-2,-1,0,1,2 κόκ. Για να υπολογιστούν οι τιμές των στοιχείων που βρίσκονται στα άκρα της νέας εικόνας η μάσκα τοποθετείται έτσι ώστε κάποια στοιχεία της βρίσκονται έξω από τα όρια της αρχικής εικόνας. Σε αυτή την περίπτωση θεωρείται αυθαίρετα ότι τα στοιχεία της αρχικής εικόνας που δεν υπάρχουν έχουν κάποια συγκεκριμένη τιμή (padding). Συνήθως η τιμή αυτή είναι το μηδέν (zero-padding) ή η κοντινότερη τιμή της αρχικής εικόνας. Η υλοποίηση του φίλτρου μπορεί να γίνει με διάφορους τρόπους στο MATLAB:

(1) I_rec=filter2(f, image_w_noise)
Φιλτράρει τα δεδομένα στον πίνακα image_w_noise με τη 2-διάστατη μάσκα του πίνακα f. Τα δεδομένα των πινάκων εισόδου και εξόδου είναι πάντα double (πραγματικοί αριθμοί διπλής ακρίβειας). Ο υπολογισμός των ακραίων στοιχείων γίνεται με zero-padding

(2) I_rec=imfilter(image_w_noise,f,X)
Τα δεδομένα εισόδου και εξόδου δεν είναι αναγκαστικά double (μπορούν να είναι και single, logical, char, cell, struct, int8, uint8, int16, uint16, int32, uint32, int64, uint64). Οι πίνακες και εισόδου και εξόδου (όπως προφανώς και η μάσκα f) μπορούν να είναι και άνω των 2 διαστάσεων. Αν X>0, τα στοιχεία εκτός εικόνας παίρνουν την τιμή X, αν X=0 τα στοιχεία εκτός εικόνας παίρνουν την τιμή 0 (zero-padding), αν X=’replicate’ τότε οι τιμές του πίνακα εισόδου που βρίσκονται εκτός των ορίων του, έχουν τιμή ίση με την κοντινότερη τιμή του πίνακα.

4.3.1. Παρουσίαση του αλγορίθμου

Παρουσιάζεται ο αλγόριθμος φιλτραρίσματος με φίλτρο μέσης τιμής:

```
function Mean_filter(image)

thorivos=input('Dose eidos thoruvou, 1: Salt and Pepper, 2: Gaussian Noise      ');
I=imread(image);
```

```
switch thorivos
    case 1
            spnoise = input('Dose puknotita thoruvou     ');
            image_w_noise=imnoise(I,'salt & pepper',spnoise);
    case 2
            gaussiannoise=input('Dose diaspora thorivou     ');
            image_w_noise=imnoise(I,'gaussian',gaussiannoise);
end

filtro=input('Dose eidos filtrou, 1:2-D me zeropadding (mono gia
aspromauri), 2:Poludiastato filtro    ');

switch filtro
    case 1
            N=input('Dose diastasi toy tetragonou     ');
            f=ones(N,N)/(N*N);
            I_rec=filter2(f,image_w_noise);
            imshow(I,'Border','tight');
            figure;
            imshow(image_w_noise,'Border','tight');
            figure;
            imshow(uint8(I_rec) ,'Border','tight');
    case 2  % Kataskeui filtrou mesis timis
             X=input('Dose tupo padding, 0:zero, X:X-padding,
            ''replicate'':replicate     ');
            f = fspecial('average');
            I_rec = imfilter(image_w_noise,f,X);
            imshow(I,'Border','tight');
            figure;
            imshow(image_w_noise,'Border','tight');
            figure;
            imshow(I_rec,'Border','tight');
End
```

Υπενθυμίζεται ότι:

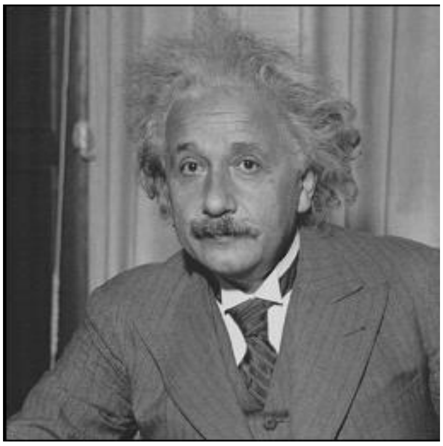
-Η συνάρτηση x=input('Keimeno'); τυπώνει το κειμένο, διαβάζει την τιμή που δίνει ο χρήστης και την καταχωρεί στην μεταβλητή x

- Η εντολή f=ones(N,N)/(N*N); κατασκευάζει τον πίνακα ω της σχέσης (4.2)

-Η εντολή f = fspecial('average'); κατασκευάζει την πολυδιάστατη μάσκα f

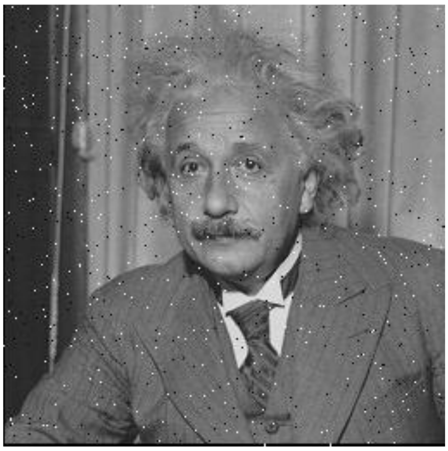
4.3.2. Αποτελέσματα για ασπρόμαυρη εικόνα

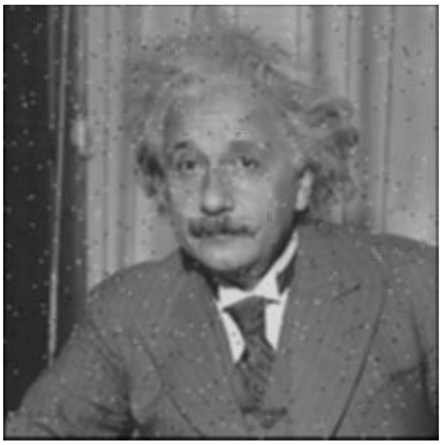
Θα χρησιμοποιήσουμε τον αλγόριθμο για ασπρόμαυρη εικόνα, μάσκα δισδιάστατη με zeropadding. Θα μεταβάλλεται το είδος και η ένταση του θορύβου καθώς και το μέγεθος του τετραγώνου της μάσκας.



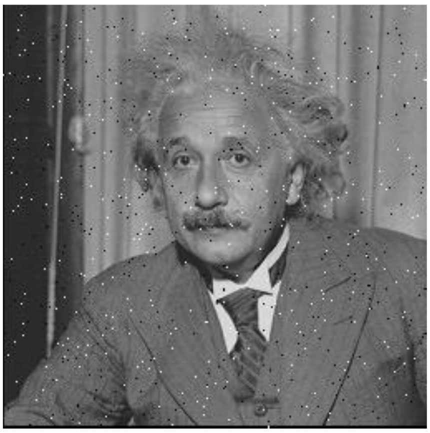
Αρχική εικόνα

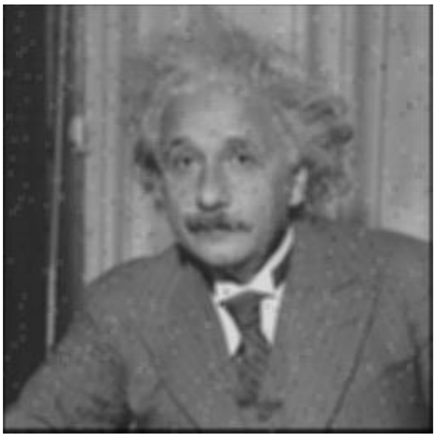
(α) Θόρυβος Salt&Pepper



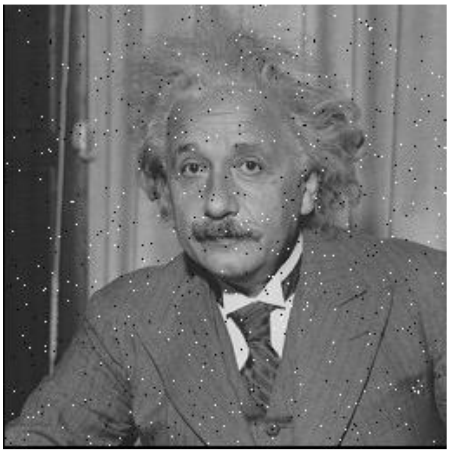
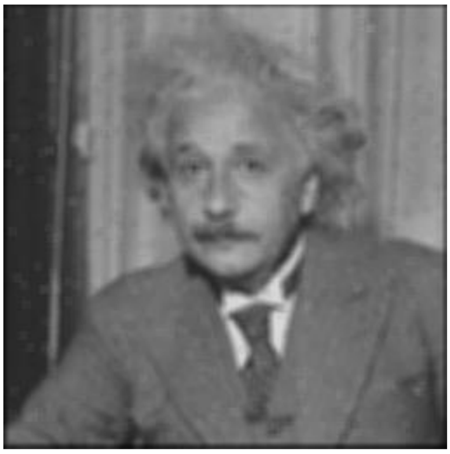


Θόρυβος με φασματική πυκνότητα θορύβου 0.01 και N=3

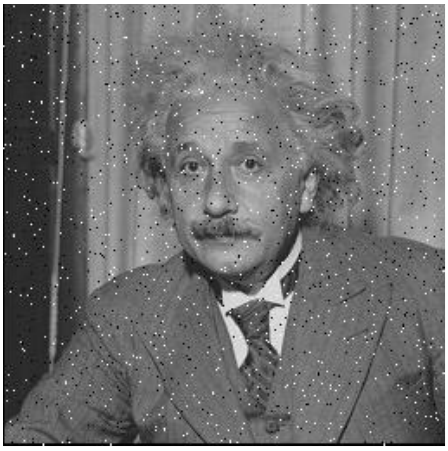
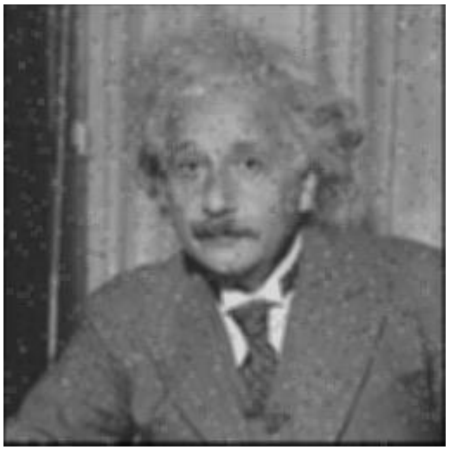




Θόρυβος με φασματική πυκνότητα θορύβου 0.01 και N=4

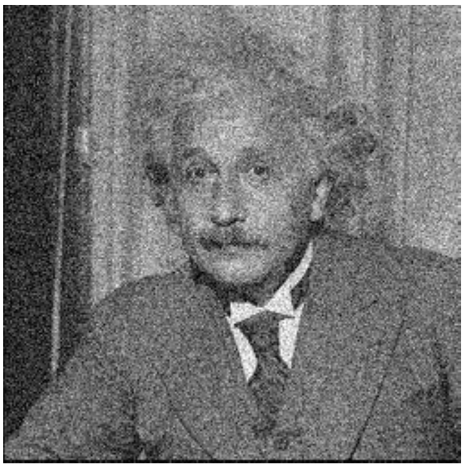
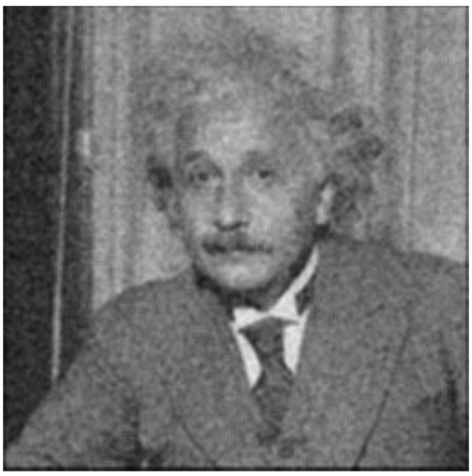



Θόρυβος με φασματική πυκνότητα θορύβου 0.01 και N=5

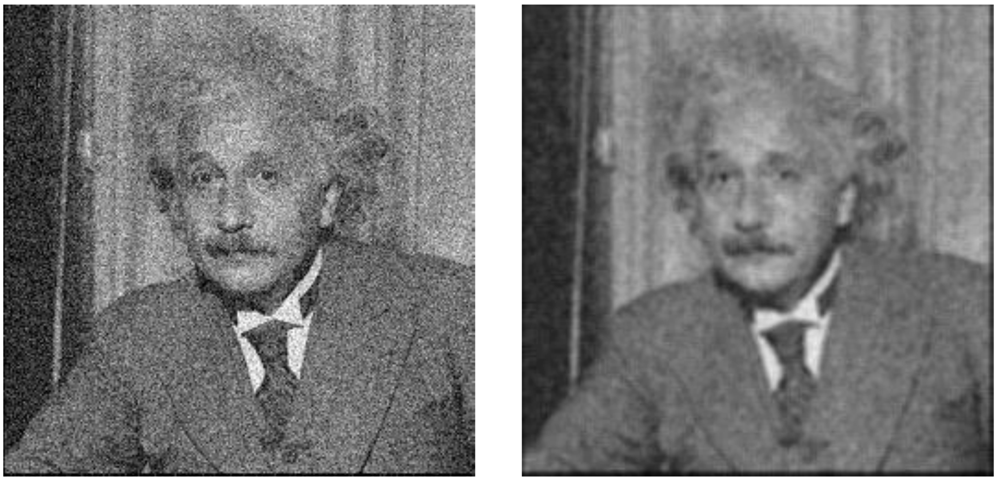



Θόρυβος με φασματική πυκνότητα θορύβου 0.02 και N=4

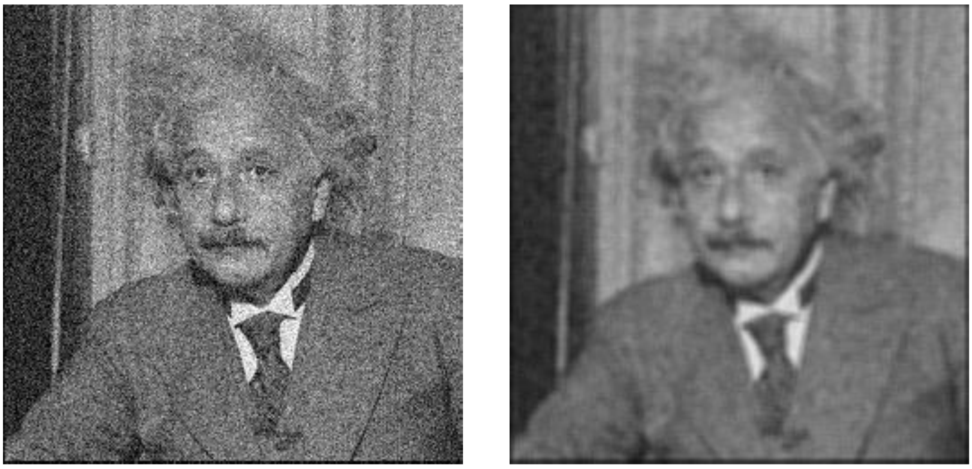
(β) Θόρυβος Gauss

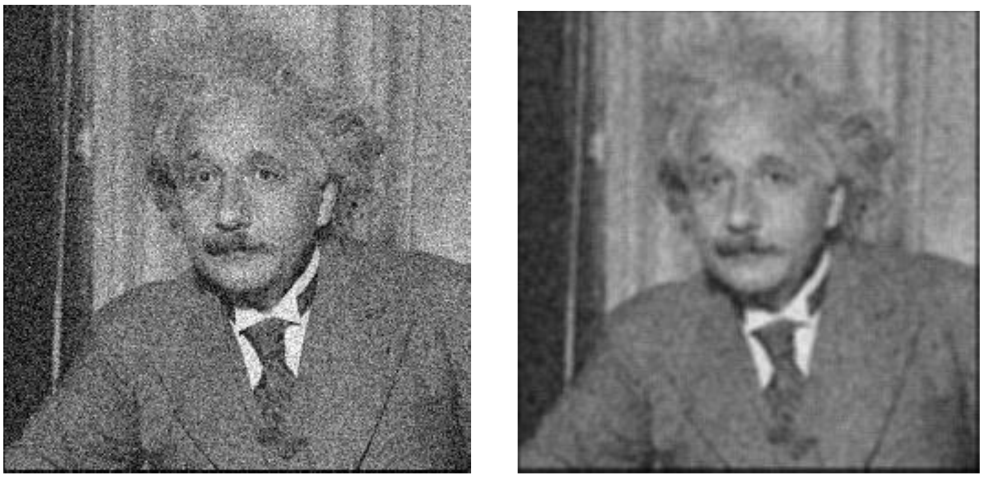
Θόρυβος με διασπορά 0.01 και N=3



Θόρυβος με διασπορά 0.01 και N=4



Θόρυβος με διασπορά 0.01 και N=5



Θόρυβος με διασπορά 0.02 και N=4

4.3.3. Αποτελέσματα για έγχρωμη εικόνα

Θα χρησιμοποιήσουμε τον αλγόριθμο για έγχρωμη εικόνα. Θα μεταβάλλεται το είδος και η ένταση του θορύβου καθώς και το padding.

(α) Θόρυβος Salt&Pepper

Θόρυβος με φασματική πυκνότητα θορύβου 0.01 και zeropadding (X=0)

Θόρυβος με φασματική πυκνότητα θορύβου 0.01 και replicate padding

Θόρυβος με φασματική πυκνότητα θορύβου 0.02 και zeropadding (X=0)

Θόρυβος με φασματική πυκνότητα θορύβου 0.02 και replicate padding

(β) Θόρυβος Gauss

Θόρυβος με διασπορά 0.01 και zeropadding (X=0)

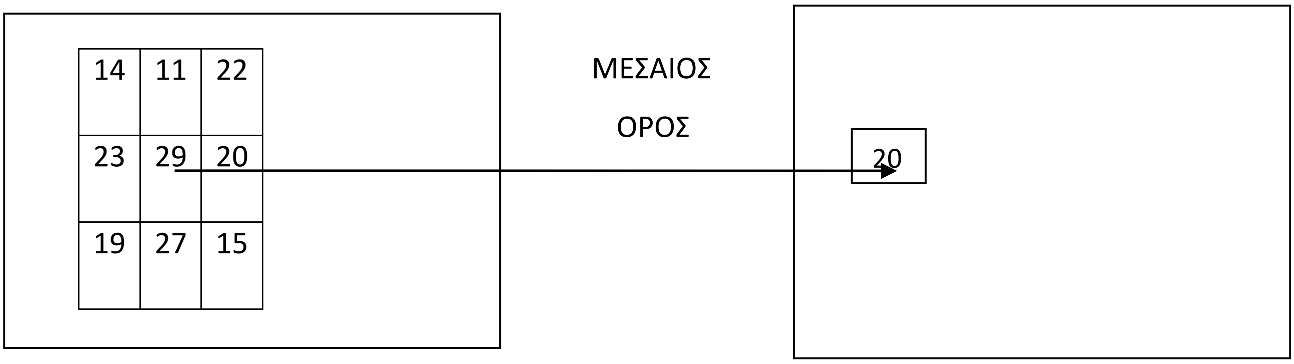
Θόρυβος με διασπορά 0.01 και replicate padding

Θόρυβος με διασπορά 0.02 και zeropadding (X=0)

Θόρυβος με διασπορά 0.02 και replicate padding

4.4 Φίλτρο μεσαίου (Median filter)

Μία άλλη μέθοδος εξομάλυνσης και μείωσης του θορύβου είναι το φίλτρο της ενδιάμεσης ή μεσαίας τιμής. Σύμφωνα με την τεχνική αυτή οι τιμές των εικονοστοιχείων μιας γειτονιάς ταξινομούνται και η τιμή κάθε εικονοστοιχείου της νέας εικόνας είναι η μεσαία (διάμεση ή ενδιάμεση) από τις τιμές των εικονοστοιχείων της γειτονιάς του αντίστοιχου εικονοστοιχείου της αρχικής.



Αρχική Εικόνα f(x,y) Φιλτραρισμένη Εικόνα g(x,y)

Ο ενδιάμεσος της γειτονιάς του 29 βρίσκεται αν ταξινομηθούν τα εικονοστοιχεία:
11 14 15 19 20 22 23 27 29

και είναι ο 20.

4.4.1. Αλγόριθμος φίλτρου μεσαίας τιμής

Παρουσιάζεται ο αλγόριθμος φίλτρου μεσαίας τιμής:

```
function Median_filter(image)

% To filtro einai to filtro endiamesis timis (median filter)
% Lamvanontai upopsin gia kathe pixel ena mxn tmima tis eikonas gurw
apo to pixel

m=input('Dwse mikos parathurou: ');
n=input('Dwse platos parathurou: ');
noise_type=input('Tupos thoruvou: 1 gia Gaussian, 2 gia Salt&Pepper:
');
A=imread(image);



switch noise_type
    case 1
      gaussiannoise=input('Dwse diaspora: ');
      Ig=imnoise(A,'gaussian',gaussiannoise);
      I_rec_g = medfilt2(Ig,[m n]);
      imshow(A,'Border','tight');
```

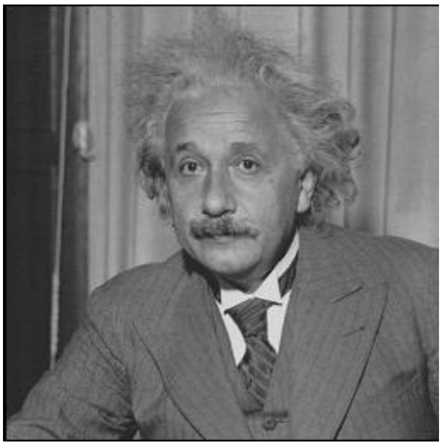
```
    figure;
    imshow(Ig,'Border','tight');
    figure;
    imshow(I_rec_g,'Border','tight');
  case 2
     spnoise=input('Dwse fasmatiki puknotita thoryvou: ');
     Isp=imnoise(A,'salt & pepper',spnoise);
     I_rec_sp = medfilt2(Isp,[m n]);
     imshow(A,'Border','tight');
     figure;
     imshow(Isp,'Border','tight');
     figure;
     imshow(I_rec_sp,'Border','tight');
end
```

Με την εντολή A=imread(image); η εικόνα αποθηκεύεται σε ένα πίνακα A. Στην συνέχεια δημιουργείται η αλλοιωμένη εικόνα Ig= imnoise(A,'gaussian',gaussiannoise); κατόπιν εφαρμογής Gaussian θορύβου με διασπορά gaussiannoise. Επίσης δημιουργείται η αλλοιωμένη εικόνα Isp=imnoise(A,'salt & pepper',spnoise); κατόπιν εφαρμογής θορύβου salt&pepper φασματικής πυκνότητας θορύβου spnoise.

Η πρώτη φιλτράρεται με την εντολή I_rec_g = medfilt2(Ig,[m n]); και η δεύτερη με την εντολή I_rec_sp = medfilt2(Isp,[m n]);. Και στις δύο περιπτώσεις το φιλτράρισμα γίνεται για ένα παράθυρο mxn γύρω από κάθε εικονοστοιχείο.

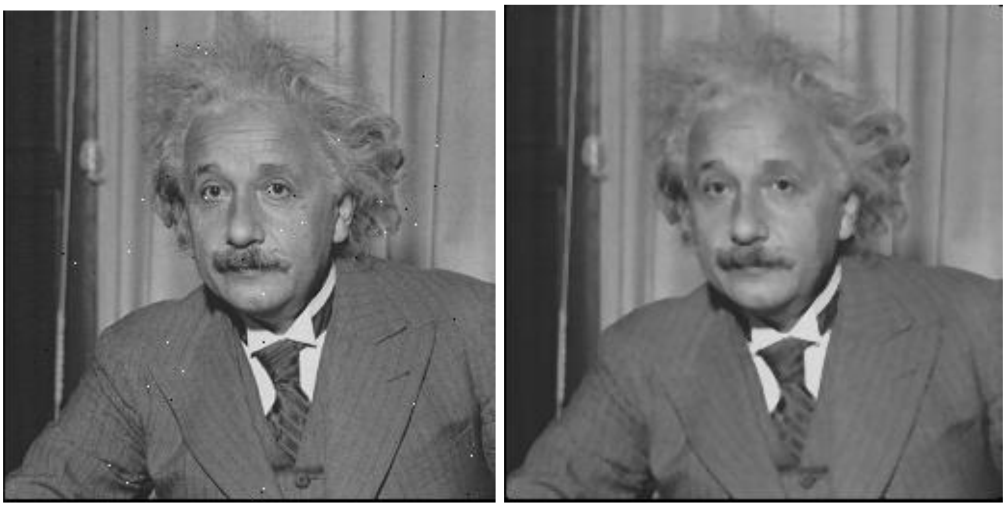
4.4.2. Αποτελέσματα για ασπρόμαυρη εικόνα

Θα χρησιμοποιήσουμε τον αλγόριθμο για ασπρόμαυρη εικόνα. Θα μεταβάλλεται το είδος και η ένταση του θορύβου καθώς και το μέγεθος του τετραγώνου της μάσκας.

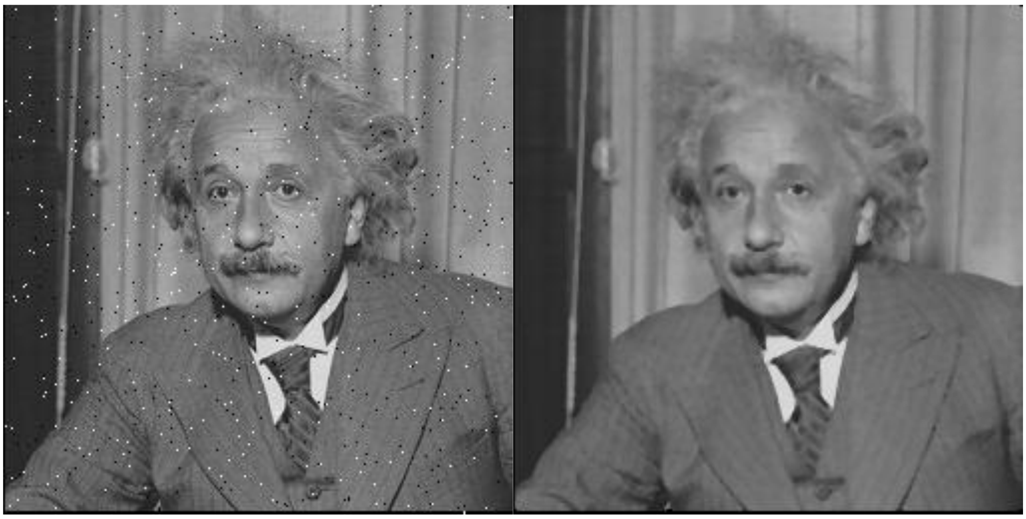


Αρχική εικόνα

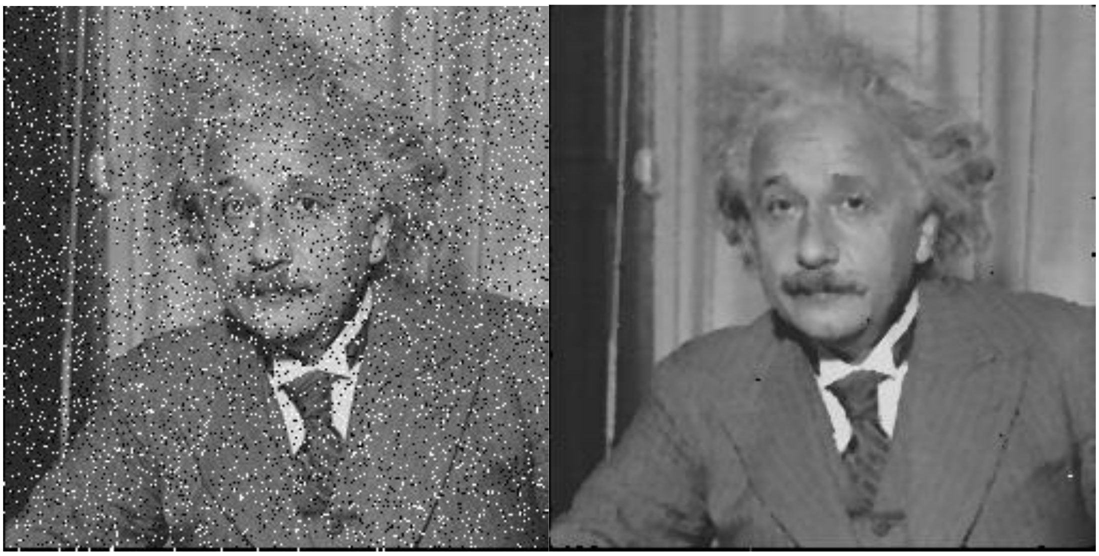
(α) Θόρυβος Salt&Pepper



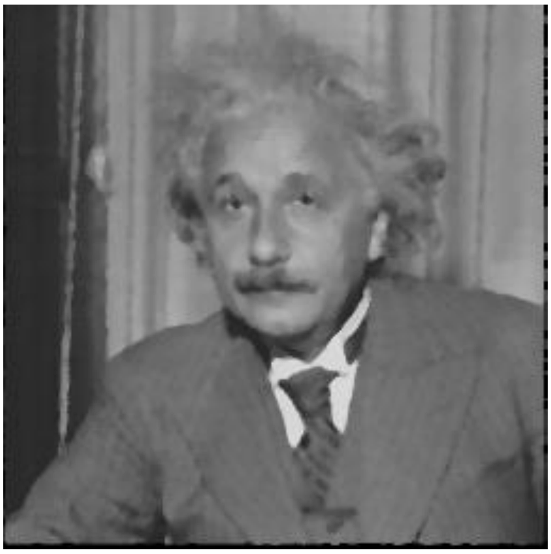
Φασματική πυκνότητα θορύβου 0.001 και παράθυρο 3x3



Φασματική πυκνότητα θορύβου 0.01 και παράθυρο 3x3



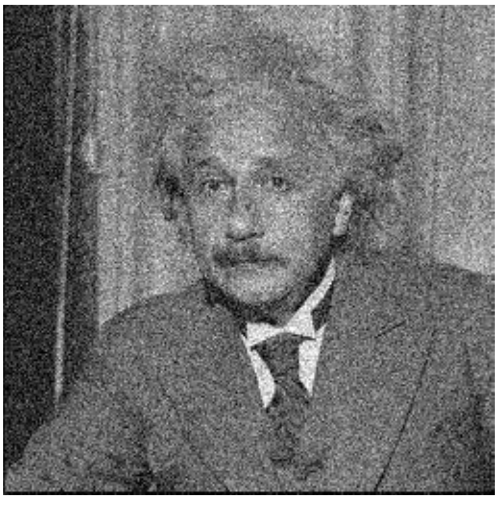
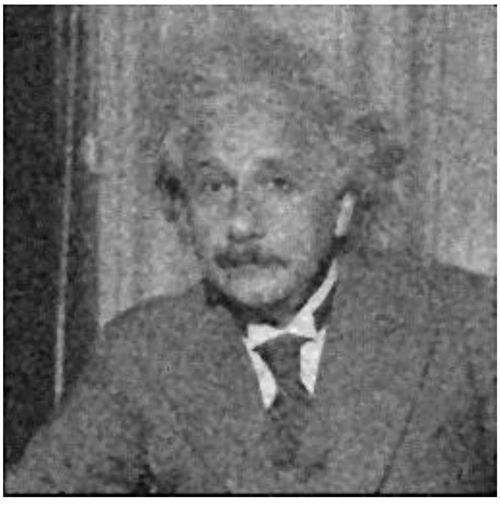
Φασματική πυκνότητα θορύβου 0.1 και παράθυρο 3x3

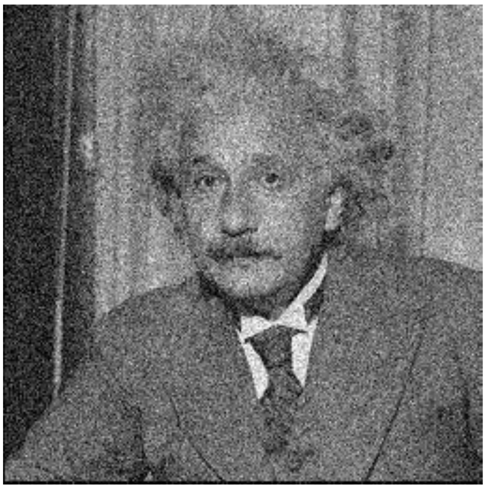
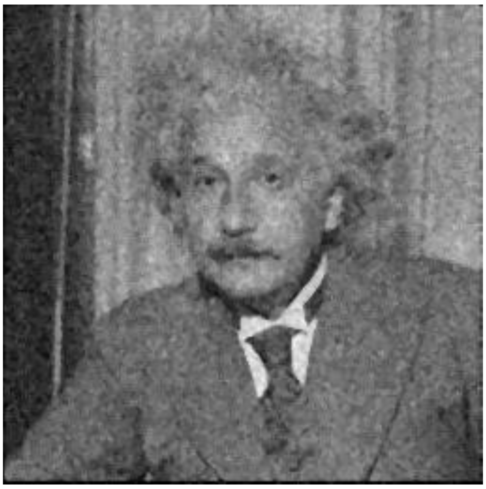
Φασματική πυκνότητα θορύβου 0.1 και παράθυρο 4x4

Παρατηρούμε πολύ καλά αποτελέσματα του φίλτρου για μεγάλο φάσμα τιμών της φασματικής πυκνότητας θορύβου. Παρατηρούμε επίσης ότι για μεγαλύτερα παράθυρα, διορθώνονται περισσότερα λάθη αλλά η εικόνα θα έχει χάσει ως προς την αντίθεση χρωματισμού και την ευκρίνειά της.

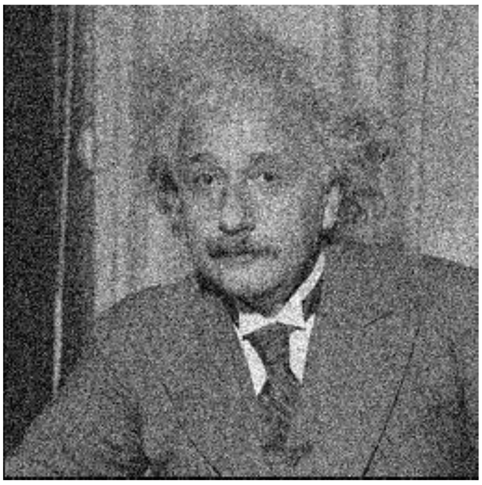
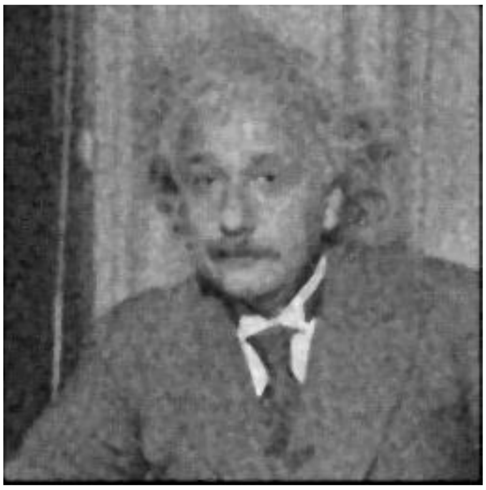
(β) Θόρυβος Gauss

Διασπορά θορύβου 0.0001 και παράθυρο 3x3

Διασπορά θορύβου 0.001 και παράθυρο 3x3

Διασπορά θορύβου 0.0001 και παράθυρο 4x4

Παρατηρούμε μικρά αποτελέσματα του φίλτρου για μεγάλο φάσμα τιμών της διασποράς θορύβου Gauss. Παρατηρούμε επίσης ότι για μεγαλύτερα παράθυρα, διορθώνονται περισσότερα λάθη αλλά η εικόνα θα έχει χάσει ως προς την αντίθεση χρωματισμού και την ευκρίνειά της.

4.4.3. Αποτελέσματα για έγχρωμη εικόνα

Θα χρησιμοποιήσουμε τον αλγόριθμο για έγχρωμη εικόνα. Θα μεταβάλλεται το είδος και η ένταση του θορύβου καθώς και το μέγεθος του τετραγώνου της μάσκας.

Παρατηρούμε πολύ καλά αποτελέσματα του φίλτρου για μεγάλο φάσμα τιμών της φασματικής πυκνότητας θορύβου. Παρατηρούμε επίσης ότι για μεγαλύτερα παράθυρα, διορθώνονται περισσότερα λάθη αλλά η εικόνα θα έχει χάσει ως προς την αντίθεση χρωματισμού και την ευκρίνειά της.

Παρατηρούμε μικρά αποτελέσματα του φίλτρου για μεγάλο φάσμα τιμών της διασποράς θορύβου Gauss. Παρατηρούμε επίσης ότι για μεγαλύτερα παράθυρα, διορθώνονται περισσότερα λάθη αλλά η εικόνα θα έχει χάσει ως προς την αντίθεση χρωματισμού και την ευκρίνειά της.

4.5. Το Γκαουσιανό φίλτρο ομαλοποίησης

Το Γκαουσιανό χαμηλοπερατό φίλτρο ορίζεται στις δύο διαστάσεις ως $h(u,v) = (1/(2*\pi*\sigma^2)) * e\hat{}(- (u^2 + v^2) / \sigma^2)$, όπου σ είναι η τυπική απόκλιση του φίλτρου. Δηλαδή το μεγαλύτερο βάρος έγκειται στο κέντρο και υπάρχει ομαλή εξασθένηση όσο απομακρυνόμαστε από το κέντρο. Συνήθως οι υψηλές συχνότητες στο φάσμα μιας εικόνας συνεισφέρουν στο διαχωρισμό των αντικειμένων δηλαδή στις λεπτομέρειες ενώ οι χαμηλές αντιπροσωπεύουν μεγάλες περιοχές αντικειμένων. Εφαρμόζοντας το χαμηλοπερατό φίλτρο οι υψηλές συχνότητες χάνονται και ο διαχωρισμός των αντικειμένων, οι ακμές, γίνονται περισσότερο δυσδιάκριτες. Με την εφαρμογή του φίλτρου οι περιοχές της εικόνας που διαχωρίζουν τα αντικείμενα, δηλαδή οι ακμές της θα έρθουν σε πιο κοντινές τιμές μετά την εφαρμογή του φίλτρου, το οποίο τείνει να εξαλείψει τις σημαντικές διαφορές μεταξύ γειτονικών εικοστοιχείων. Έτσι οι ακμές των αντικειμένων γίνονται δυσδιάκριτες και θολές.

4.5.1. Παρουσίαση του αλγορίθμου του Γκαουσιανού φίλτρου

Ο αλγόριθμος είναι ο εξής:

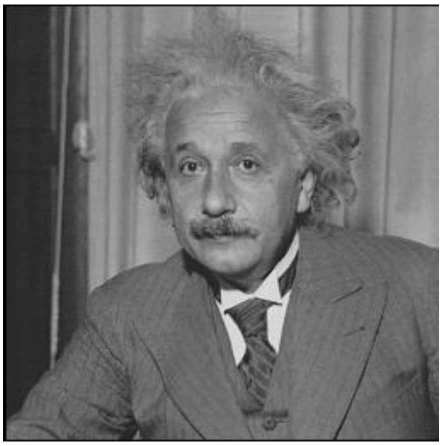
```
function Gaussian_filtering(image)

% Xamiloperato filtrarisma eikonwn gia afairesi upsisuxnwn armonikwn,
thoruvou klp
noise_type=input('Tupos thoruvou: 1 gia Gaussian, 2 gia Salt&Pepper:
');
A=imread(image);

switch noise_type
    case 1
      gaussiannoise=input('Dwse diaspora: ');
      Ig=imnoise(A,'gaussian',gaussiannoise);
      I_rec_g = imgaussfilt(Ig, 2);
      imshow(A,'Border','tight');
      figure;
      imshow(Ig,'Border','tight');
      figure;
      imshow(I_rec_g,'Border','tight');
    case 2
       spnoise=input('Dwse fasmatiki puknotita thoryvou: ');
       Isp=imnoise(A,'salt & pepper',spnoise);
       I_rec_sp = imgaussfilt(Isp, 2);
```

```
        imshow(A,'Border','tight');
        figure;
        imshow(Isp,'Border','tight');
        figure;
        imshow(I_rec_sp,'Border','tight');
end
```

Η εντολή I_rec_g imgaussfilt(Ig, 2); φιλτράρει μία εικόνα αλλοιωμένη κατά Gauss και η εντολή I_rec_sp = imgaussfilt(Isp, 2); φιλτράρει μία εικόνα αλλοιωμένη κατά Salt&Pepper.

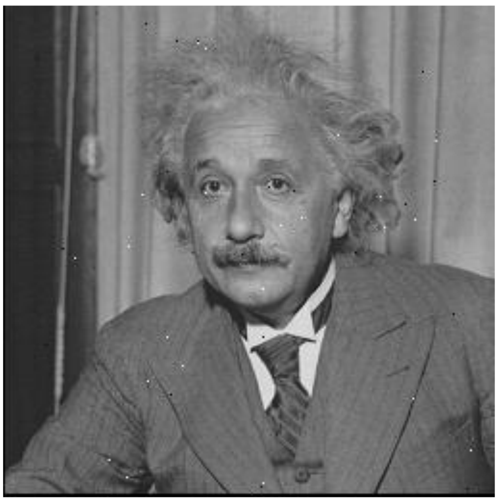
4.5.2. Αποτελέσματα για ασπρόμαυρη εικόνα

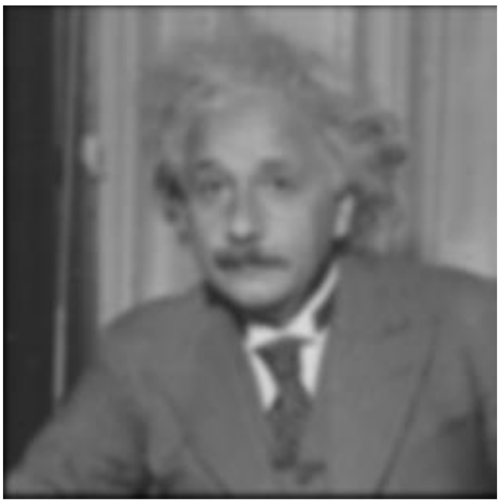
Θα χρησιμοποιήσουμε τον αλγόριθμο για ασπρόμαυρη εικόνα. Θα μεταβάλλεται το είδος και η ένταση του θορύβου καθώς και το μέγεθος του τετραγώνου της μάσκας.



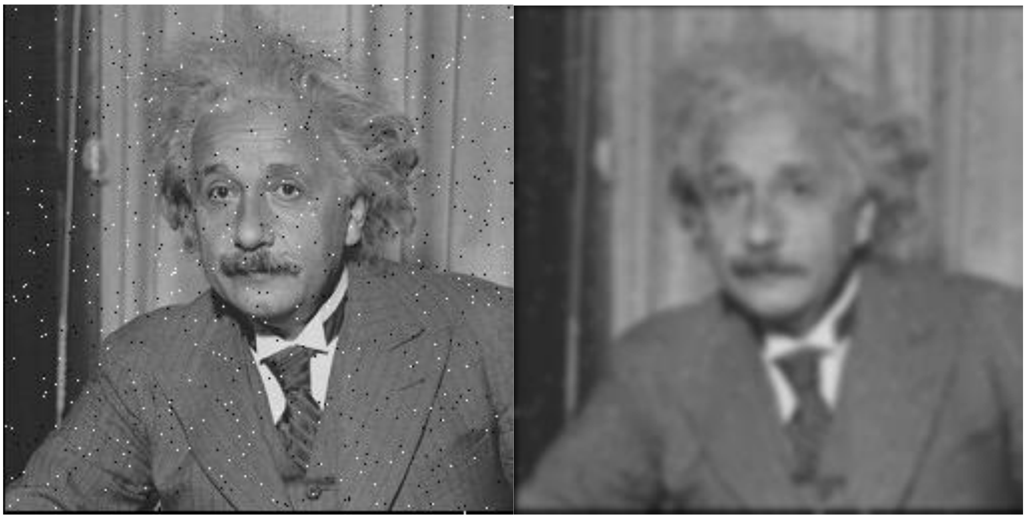
Αρχική εικόνα

(α) Θόρυβος Salt&Pepper





Φασματική πυκνότητα θορύβου 0.001

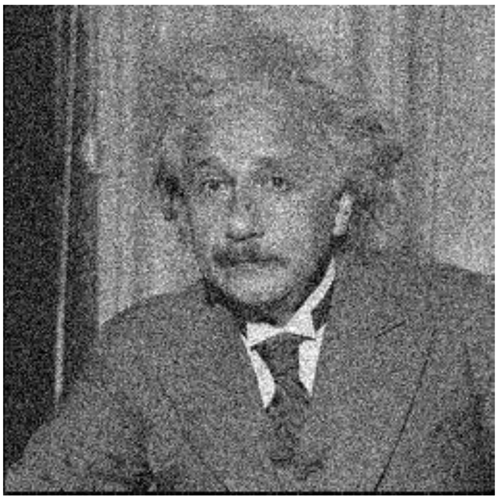
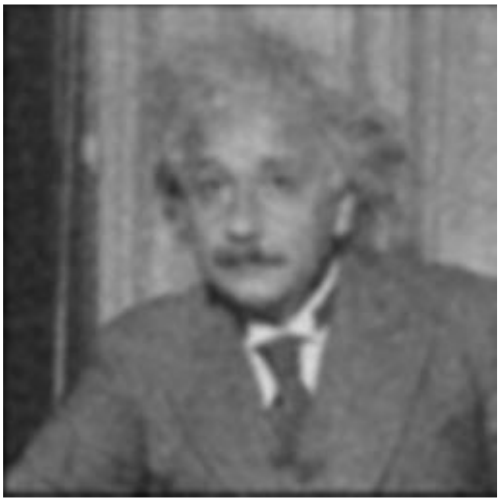


Φασματική πυκνότητα θορύβου 0.01

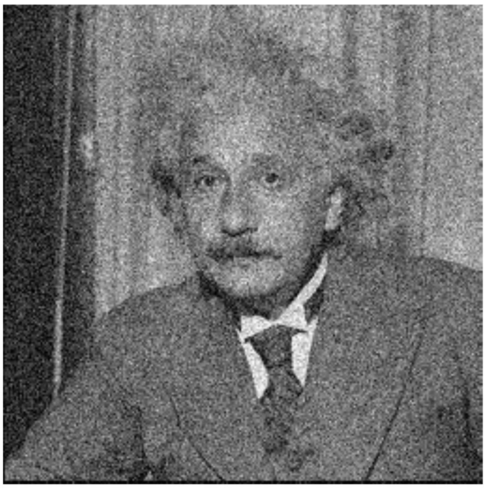
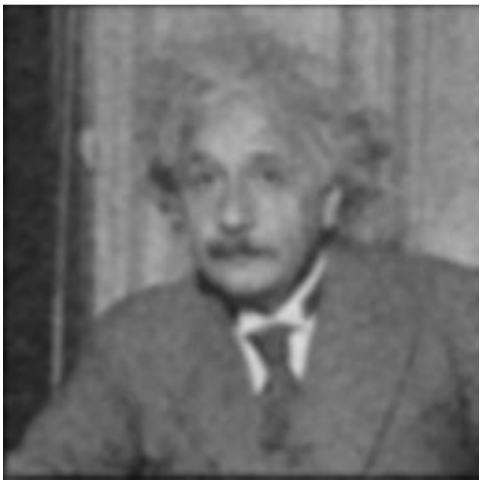
Φασματική πυκνότητα θορύβου 0.1

Παρατηρούμε πολύ καλά αποτελέσματα του φίλτρου για χαμηλότερες τιμές της φασματικής πυκνότητας θορύβου.

(β) Θόρυβος Gauss

Διασπορά θορύβου 0.0001

Διασπορά θορύβου 0.001

Παρατηρούμε μικρά αποτελέσματα του φίλτρου για μεγάλο φάσμα τιμών της διασποράς θορύβου Gauss.

4.5.3. Αποτελέσματα για έγχρωμη εικόνα

Θα χρησιμοποιήσουμε τον αλγόριθμο για έγχρωμη εικόνα. Θα μεταβάλλεται το είδος και η ένταση του θορύβου καθώς και το μέγεθος του τετραγώνου της μάσκας.

Παρατηρούμε πολύ καλά αποτελέσματα του φίλτρου για χαμηλότερες τιμές της φασματικής πυκνότητας θορύβου. Παρατηρούμε μικρά αποτελέσματα του φίλτρου για μεγάλο φάσμα τιμών της διασποράς θορύβου Gauss.

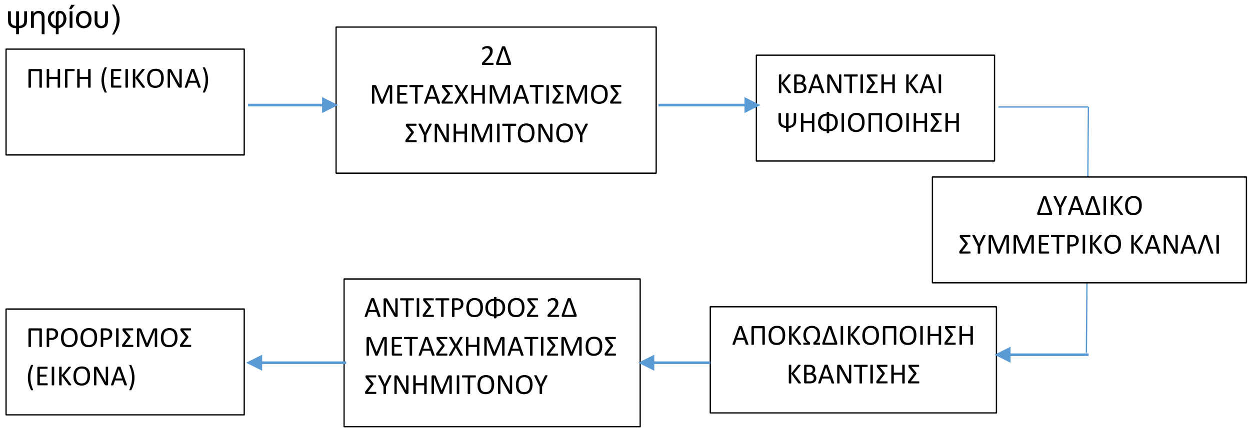
ΚΕΦΑΛΑΙΟ 5. ΜΕΤΑΔΟΣΗ ΕΙΚΟΝΑΣ

5.1 Εισαγωγικά

Σε αυτό το κεφάλαιο θα μελετήσουμε σύστημα μετάδοσης εικόνας, στο οποίο εφαρμόζουμε κωδικοποίηση εικόνας βασισμένο σε μετασχηματισμό. Ο Μετασχηματισμός που θα χρησιμοποιήσουμε είναι ο ΜΣ Διακριτού Μετασχηματισμού Συνημιτόνου (Discrete Cosine Transform, DCT).

Στην συνέχεια η (συμπιεσμένη) πληροφορία κβαντίζεται και ψηφιοποιείται. Τέλος μεταδίδεται μέσα από ένα Δυαδικό Συμμετρικό Κανάλι (Binary Symmetric Channel,BSC).

Η μετάδοση εικόνας εφαρμόζεται για διάφορες τιμές των σχεδιαστικών παραμέτρων (επιλεγόμενος ΜΣ, διάστημα κβάντισης, πιθανότητα αλλαγής ψηφίου)



Σχήμα 5.1. Σύστημα Μετάδοσης Εικόνας

5.2. Κωδικοποίηση πηγής με τον Δισδιάτατο (2Δ) διακριτό ΜΣ Συνημιτόνου

Οι τεχνικές κωδικοποίησης πηγής αυτές είναι τεχνικές με πρόβλεψη και εφαρμόζονται στα εικονοστοιχεία (pixels) της εικόνας, επομένως είναι χωρικές τεχνικές (spatial domain techniques).

Για διακριτά σήματα 2 διαστάσεων $x[n_1,n_2]$, όπως είναι οι εικόνες (διακριτά επειδή έχουν διακριτό αριθμό εικονοστοιχείων κατά μήκος και κατά πλάτος), με διαστάσεις N_1, N_2 τα διανύσματα βάσης για $n_1=0,\ldots, N_1-1$, $n_2=0,\ldots, N_2-1$, $k_1=0,\ldots, N_1-1$, $k_2=0,\ldots, N_1-1$ είναι τα $\mathbf{g}_{k1k2}$ και οι διαστάσεις τους $g_{k1k2}(n_1,n_2)$ δίνονται από την σχέση:

$$g_{k1k2}\,(n_1,n_2) = g_{k1}(n_1)\; g_{k2}(n_2) \qquad (5.1)$$

Τα διανύσματα $\mathbf{g}_k$ είναι ορθοκανονικά, δηλαδή: (α) το εσωτερικό γινόμενο τους ανά δύο είναι πάντα 0 (είναι ανά δύο ορθογώνια) και (β) το εσωτερικό γινόμενο του κάθε διανύσματος με τον εαυτό του (δηλαδλη το τετράγωνο του μέτρου του διανύσματος) είναι πάντα 1. Οι διαστάσεις τους $g_k(n)$ δίνονται από την σχέση:

$g_0(n) = \frac{1}{\sqrt{N}}$ και $g_k(n) = \sqrt{\frac{2}{N}}\cos(\frac{\pi}{2N}k(2n+1))$ (5.2)

Για παράδειγμα:

$$g_{00}(n_1, n_2) = \frac{1}{\sqrt{N_1 N_2}},$$

$$g_{0k_2}(n_1, n_2) = \frac{1}{\sqrt{N_1}}\sqrt{\frac{2}{N_2}} \cdot \cos(\frac{\pi k_2(2n_2+1)}{2N_2}),$$

$$g_{k_1k_2}(n_1, n_2) = \sqrt{\frac{2}{N_1}} \cdot \cos(\frac{\pi k_1(2n_1+1)}{2N_1}) \cdot \sqrt{\frac{2}{N_2}} \cdot \cos(\frac{\pi k_2(2n_2+1)}{2N_2})$$

Οι προβολές ενός διανύσματος (εικόνας) $x[n_1,n_2]$ στα ορθοκανονικά διανύσματα 2 διαστάσεων της σχέσης (5.1) είναι ουσιαστικά ο δισδιάστατος (2Δ) διακριτός ΜΣ Συνημιτόνου (2D-DCT):

$$c_{00} = \frac{1}{\sqrt{N_1 N_2}}\sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} x[n_1, n_2]$$

$$c_{0k_2} = \sqrt{\frac{2}{N_1 N_2}}\sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} x[n_1, n_2]\cos(\frac{\pi k_2(2n_2+1)}{2N_2})$$

$$c_{k_10} = \sqrt{\frac{2}{N_1 N_2}}\sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} x[n_1, n_2]\cos(\frac{\pi k_1(2n_1+1)}{2N_1})$$

$$c_{k_1k_2} = \sqrt{\frac{2}{N_1 N_2}}\sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} x[n_1, n_2]\cos(\frac{\pi k_1(2n_1+1)}{2N_1})\cos(\frac{\pi k_2(2n_2+1)}{2N_2})$$

(Σχέσεις 5.3.α-5.3.δ, $k_1 \neq 0$, $k_2 \neq 0$)

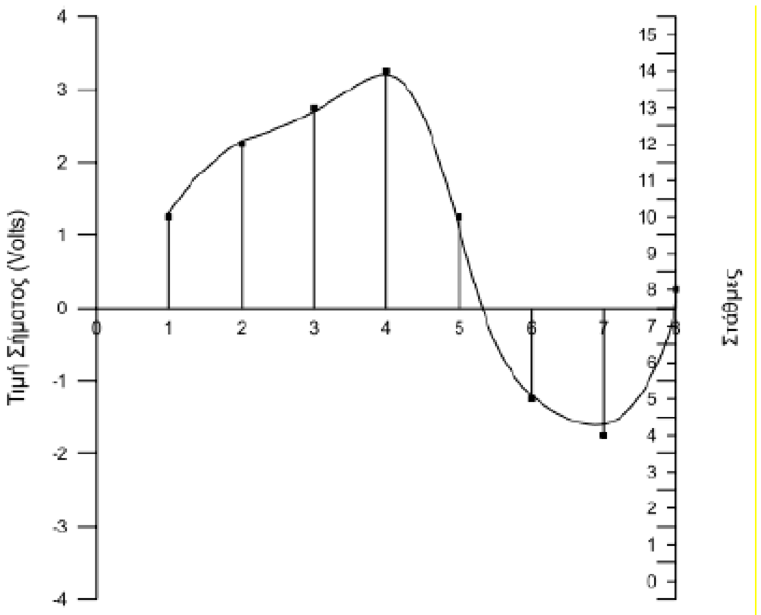
Με τον Μετασχηματισμό αυτό το σήμα $x[n_1,n_2]$ γράφεται ως γραμμικός συνδυασμός των ορθοκανονικών διανυσμάτων (σχέσεις 5.1-5.2) με συντελεστές που δίνονται από τις σχέσεις (5.3.α-5.3.δ):

$$x[n_1, n_2] = \sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} \mathrm{c}_{k_1k_2}\mathrm{g}_{k_1k_2}(\mathrm{n1}, \mathrm{n2})$$

5.3 Κβάντιση

Ένα σήμα το οποίο παράγεται από τον φυσικό κόσμο ή την ανθρώπινη δραστηριότητα είναι συνήθως συνεχούς χρόνου και παίρνει τιμές σε ένα συνεχές πεδίο τιμών πλάτους. Έτσι και οι τιμές που προκύπτουν από τον 2Δ ΜΣ DCT είναι συνεχούς πλάτους. Αυτό δυσκολεύει με πολλούς τρόπους την επεξεργασία των ψηφιακών σημάτων (αποθήκευση, μετάδοση). Γι' αυτό το αρχικό σήμα μπορεί να προσεγγιστεί ικανοποιητικά από ένα σήμα το οποίο αποτελείται από διακριτές προκαθορισμένες στάθμες (επίπεδα πλάτους) ορισμένα έτσι ώστε να ελαχιστοποιούνται οι διαφορές με το αρχικό σήμα. Η διαδικασία αυτή λέγεται κβάντιση (quantization).

Στην διαδικασία της κβάντισης των δειγμάτων του δειγματοληπτημένου σήματος, επιλέγεται αρχικά η περιοχή μέσα στην οποία παίρνει τιμές το πλάτος των δειγμάτων έτσι ώστε να μπορεί μέσα σε αυτό το εύρος να περιέχεται το σήμα (σχήμα 5.2). Στην συνέχεια επιλέγεται το πλήθος των επιτρεπόμενων τιμών πλάτους και η επιλεγμένη περιοχή χωρίζεται σε ισάριθμα διαστήματα (την απόσταση μεταξύ των προκαθορισμένων τιμών πλάτους). Τέλος, γίνεται η στρογγυλοποίηση/προσέγγιση των τιμών των δειγμάτων στην πλησιέστερη επιτρεπτή στάθμη, όπως δίνεται στον πίνακα του σχήματος 5.3. Έτσι οι τιμές που προκύπτουν για τα πλάτη των δειγμάτων, ανήκουν σε ένα πεπερασμένο σύνολο τιμών. Είναι προφανές ότι κατά την κβάντιση υπεισέρχεται ένα σφάλμα (θόρυβος κβάντισης), το οποίο οφείλεται στην στρογγυλοποίηση των τιμών των δειγμάτων και το οποίο προκαλεί αλλοίωση του σήματος. Όσο περισσότερες είναι οι επιτρεπόμενες στάθμες, τόσο μικραίνει ο θόρυβος κβάντισης, με αποτέλεσμα τη μείωση της αλλοίωσης του σήματος.



Σχήμα 5.2 Κβάντιση

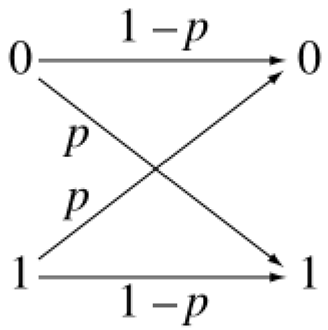
Τιμή Δείγματος (V)	1,3	2,3	2,7	3,2	1,1	-1,2	-1,6	0,1
Πλησιέστερη στάθμη (V)	1,25	2,25	2,75	3,25	1,25	-1,25	-1,75	0,25
Αριθμός στάθμης	10	12	13	14	10	5	4	8
Δυαδικός Κώδικας	1010	1100	1101	1110	1010	0101	0100	1000

Σχήμα 5.3 Διακριτοποίηση και ψηφιοποίηση

Σε αυτό το σενάριο μετάδοσης επιλέγουμε μεταβλητό διάστημα κβάντισης και κανόνα κβάντισης την στρογγυλοποίηση στην πλησιέστερη επιτρεπτή στάθμη.

5.4 Δυαδικό Συμμετρικό Κανάλι (Binary Symmetric Channel - BSC)

Ένα δυαδικό συμμετρικό κανάλι (ΔΣΚ) διασταυρωμένο με την πιθανότητα p είναι μία δυαδική είσοδος, το δυαδικό κανάλι εξόδου το οποίο αντιστρέφει την είσοδο με πιθανότητα p:



Σχήμα 5.4

Η πιθάνοτητα να έχουν γίνει k λάθη κατά την μετάδοση μίας δυαδικής λέξης μήκους n (k<n) θα είναι λοιπόν:

$p(j|i)=p^k(1-p)^{n-k}$ (5.4)

Η ανωτέρω σχέση δίνει την πιθανότητα να ληφθεί η δυαδική λέξη j όταν έχει σταλεί η δυαδική λέξη i. Οι δύο αυτές δυαδικές λέξεις i,j μήκους n διαφέρουν σε k θέσεις (k λάθη κάτα την μετάδοση).

5.5. Αλγόριθμος Υλοποίησης – Επεξήγηση

Ο αλγόριθμος επεξήγησης είναι ο εξής:

```
function  SC_rate=dct_paradeigma(image)

% Efamogi metadosis eikonas me xrisi tou DCT
% Xrisimopoioyme Kvantisi me kanona kvantisis tin stroguloopoiisi
```

```
A=imread(image);
J=dct2(A);
J(abs(J) < 10) = 0;
diastima = input('Dose diastima kvantisis: ');
J_quantized = diastima*round(J/diastima);
mikos_lexis = ceil(log2((max(J_quantized(:))-
min(J_quantized(:)))/diastima));
% Ta stoixeia tou J_quantized psifiopoiountai kai topothetountai ana
stili
J_tr = dec2bin(J_quantized+abs(min(J_quantized(:))),mikos_lexis);
for i=1:size(J_tr,1)
  for j=1:size(J_tr,2)
      J_Tx(i,j)=str2num(J_tr(i,j));
  end
end

crossover_p = input('Dose pithanotita allagis bit: ');
J_Rx = bsc(J_Tx,crossover_p);
J_received =int2str(J_Rx);
J_reconstr = bin2dec(J_received)-abs(min(J_quantized(:)));

for j=1:size(J_quantized,2)
  J_reconstruct(1:size(J_quantized,1),j)=J_reconstr((j-
1)*size(J_quantized,1)+1:j*size(J_quantized,1));
end

I_rec= idct2(J_reconstruct);
imshow(A,'Border','tight');
figure;
imshow(I_rec,[0 255],'Border','tight');

%Ypologismos toy apaitoumenou source coding rate (bits/source symbol)
transmitted = size(J,1)*size(J,2) - size(find(J==0),1);
transmitted_bits = transmitted*mikos_lexis + size(find(J==0),1)*1;
SC_rate = transmitted_bits/(size(J,1)*size(J,2));
```

Επεξήγηση του αλγορίθμου

(α) Αρχικά η εικόνα διαβάζεται με την συνάρτηση imread(image) και αποθηκεύεται στον πίνακα A.

(β) Στην συνέχεια η συνάρτηση J=dct2(A) υπολογίζει τον 2Δ ΜΣ DCT της εικόνας και αποθυκεύει τους συντελεστές του ΜΣ στον πίνακα J.

(γ) Με την επόμενη εντολή `J(abs(J) < 10) = 0;` μηδενίζονται οι συντελεστές με απόλυτη τιμή κάτω από 10, καθώς η μετάδοσή τους δεν επηρεάζει την ποιότητα της εικόνας.

(δ) Στην συνέχεια ο χρήστης εισάγει στον αλγόριθμο το διάστημα κβάντισης (diastima) με την εντολή `diastima = input('Dose diastima kvantisis: ');`

(ε) Στην συνέχεια με την εντολή `J_quantized = diastima*round(J/diastima);` υπολογίζονται οι διακριτές (κβαντισμένες) τιμές όλων των τιμών του πίνακα J. Ο κανόνας κβάντισης, όπως ειπώθηκε, είναι η στρογγυλοποίηση προς την πλησιέστερη διακριτή τιμή.

(στ) Με την εντολή `mikos_lexis = ceil(log2((max(J_quantized(:))-min(J_quantized(:)))/diastima));` υπολογίζεται το μήκος της ψηφιακής λέξης που απαιτείται για την αναπαράσταση των διακριτών τιμών. Αρχικά διαιρούμε το εύρος των διακριτών τιμών `max(J_quantized(:))-min(J_quantized(:))` με το διάστημα κβάντισης diastima, βρίσκοντας έτσι το πλήθος των διακριτών τιμών. Μετά παίρνουμε τον λογάριθμο με βάση το 2 και στρογγυλοποιούμε προς τα πάνω για να βρούμε το μήκος της ψηφιακής λέξης.

(ζ) Στην συνέχεια με την εντολή που ακολουθεί:

```
J_tr = dec2bin(J_quantized+abs(min(J_quantized(:))),mikos_lexis);
```

μετατρέπονται όλες οι διακριτές τιμές που βρίσκονται αποθηκευμένες στον πίνακα J_quantized σε ψηφιακές λέξεις αποθηκευμένες στον πίνακα J_tr. Αρχικά στις τιμές του πίνακα J_quantized προστίθεται το απόλυτο του μεγαλύτερου αρνητικού του πίνακα J_quantized για να εξασφαλιστεί ότι όλες οι τιμές του πίνακα J_quantized θα είναι θετικές ή 0. Έτσι δημιουργείται ο πίνακας `J_quantized+abs(min(J_quantized(:))`. Στην συνέχεια με τη συνάρτηση dec2bin(J_quantized+abs(min(J_quantized(:)),mikos_lexis) όλες οι διακριτές τιμές του πίνακα J_quantized+abs(min(J_quantized(:)) μετατρέπονται σε ψηφιακές λέξεις με σταθερό μήκος, το οποίο είναι το mikos_lexis που υπολογίσαμε παραπάνω. Οι ψηφιακές λέξεις αυτές αποθηκεύονται, η μία κάτω από την άλλη, στον πίνακα J_tr. Τέλος θα πρέπει να τονιστεί ότι το MATLAB αντιλαμβάνεται τις ψηφιακές αυτές λέξεις, προκειμένου να τις αναπαραστήσει, ως συμβολοσειρές (string).

(η) Στην συνέχεια ακολουθεί ο διπλός βρόχος:

```
for i=1:size(J_tr,1)
  for j=1:size(J_tr,2)
      J_Tx(i,j)=str2num(J_tr(i,j));
  end
end
```

Με τον διπλό αυτό βρόχο, κάθε ψηφιακή λέξη αλλάζει μορφή και από συμβολοσειρά γίνεται πίνακας μήκους mikos_lexis και περιεχόμενα 0 ή 1. Ο νέος πίνακας, με την ονομασία J_Tx θα είναι αυτός που θα εισέλθει στο κανάλι.

(θ) Με την εντολή `crossover_p = input('Dose pithanotita allagis bit: ');` ο χρήστης δίνει στον αλγόριθμο την πιθανότητα αλλαγής ψηφίου crossover_p

(ι)Με την εντολή `J_Rx = bsc(J_Tx,crossover_p);` περνάει η πληροφορία από το κανάλι και αποθηκεύεται στον πίνακα J_Rx

(ια) Με την εντολή `J_received =int2str(J_Rx);` οι ψηφιακές λέξεις που λάβαμε από το κανάλι J_Rx αλλάζουν μορφή και κάθε ψηφιακή λέξη μετατρέπεται από πίνακας μήκους mikos_lexis σε συμβολοσειρά (string)

(ιβ) Με την εντολή `J_reconstr = bin2dec(J_received)-abs(min(J_quantized(:)));` οι ψηφιακές λέξεις του πίνακα J_received μετατρέπονται σε διακριτές πραγματικές τιμές και αφαιρείται από όλες τις διακριτές τιμές η ποσότητα `abs(min(J_quantized(:)))` που είχε προστεθεί σε όλες αυτές στο βήμα (ζ) (ιγ) Με τον βρόχο

```
for j=1:size(J_quantized,2)
  J_reconstruct(1:size(J_quantized,1),j)=J_reconstr((j-
  1)*size(J_quantized,1)+1:j*size(J_quantized,1));
end
```

αναδιατάσσονται οι διακριτές τιμές του πίνακα J_reconstr. Πιο συγκεκριμένα, στον πίνακα J_reconstr οι διακριτές τιμές είναι η μία κάτω από την άλλη. Στον πίνακα J_reconstruct οι τιμές τοποθετούνται με την σειρά που εμφανίστηκαν στον πίνακα J_quantized (ή στον πίνακα J)

(ιδ) Με την εντολή `I_rec= idct2(J_reconstruct);` υπολογίζουμε τον αντίστροφο 2Δ Μετασχηματισμό Συνημιτόνου και αποθηκεύουμε τις τιμές (της ανακατασκευασμένης) εικόνας στον πίνακα I_rec

(ιε) Με τις εντολές

```
imshow(A,'Border','tight');
figure;
imshow(I_rec,[0 255],'Border','tight');
```

προβάλλουμε την αρχική εικόνα (πηγή) και την ανακατασκευασμένη εικόνα (προορισμός).

(ιστ) Με την εντολή `transmitted = size(J,1)*size(J,2) - size(find(J==0),1);` υπολογίζουμε το πλήθος των στοιχείων-συμβόλων του πίνακα J τα οποία δεν είναι μηδέν και τα οποία θα μεταδοθούν (αφαιρούμε από το συνολικό πλήθος των στοιχείων του J το πλήθος των στοιχείων που είναι μηδέν).

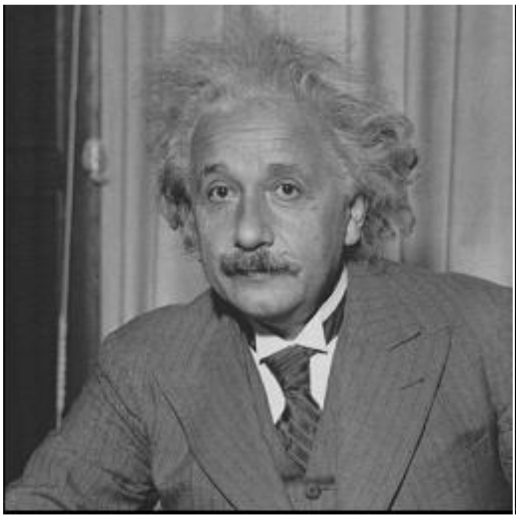
(ιζ) Με την εντολή `transmitted_bits = transmitted*mikos_lexis + size(find(J==0),1)*1;` υπολογίζουμε το πλήθος των ψηφίων (bits) που θα μεταδοθούν και τα οποία είναι: (1) ψηφιακές λέξεις μήκους mikos_lexis και πλήθους transmitted και (2) σημαίες (flags) του 1 bit για κάθε μη μεταδιδόμενο σύμβολο (μηδενικό του πίνακα J)

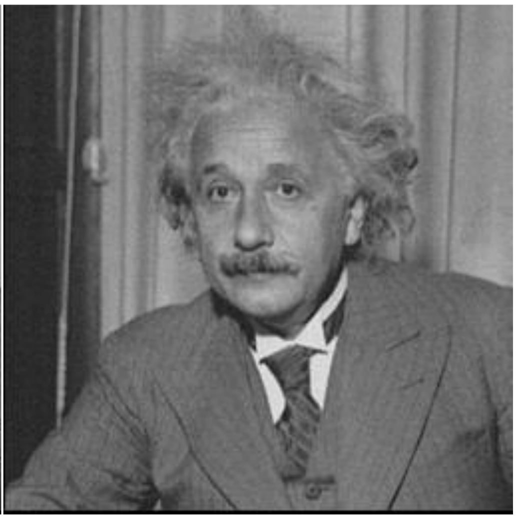
(ιη) Τέλος με την εντολή `SC_rate = transmitted_bits/(size(J,1)*size(J,2));` υπολογίζεται ο ρυθμός κωδικοποίησης πηγής (συμπίεσης) σε αριθμό bits ανά σύμβολο πηγής, που είναι το πλήθος των συνολικά μεταδιδόμενων ψηφίων (bits) προς το πληθός των συνολικών συμβόλων του πίνακα J (ή της εικόνας A).

5.6 Αποτελέσματα του αλγορίθμου

Ο αλγόριθμος δοκιμάστηκε για την ασπρόμαυρη είκονα einstein.jpg.

(i) Βήμα κβάντισης 1 και πιθανότητα αλλαγής ψηφίου 0 (αθόρυβο κανάλι)

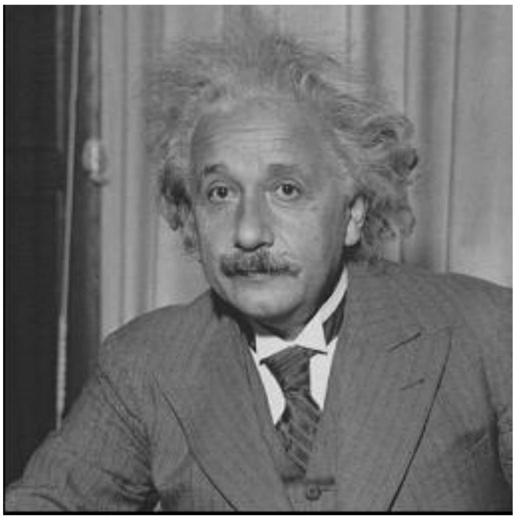


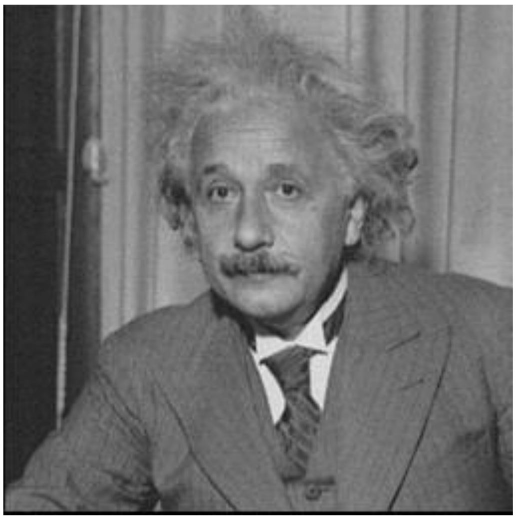


Αρχική Εικόνα   Ανακατασκευασμένη εικόνα

Η ποιότητα της ανακατασκευασμένης εικόνας είναι πολύ καλή και δύσκολα διακρίνεται διαφορά ποιότητας από την αρχική. Ο ρυθμός κωδικοποίησης πηγής υπολογίστηκε από τον αλγόριθμο σε 4.27 bits/symbol, που είναι πολύ μικρότερος από τα 8 bits/symbol που θα χρειάζονταν για να αναπαραστήσουν κάθε symbol (κάθε symbol είναι μία διακριτή τιμή από 0 έως 255 που αντιστοιχεί στις 256 αποχρώσεις του γκρι και αντιστοιχεί σε ένα εικονοστοιχείο – pixel)

(ii) Βήμα κβάντισης 2 και πιθανότητα αλλαγής ψηφίου 0 (αθόρυβο κανάλι)

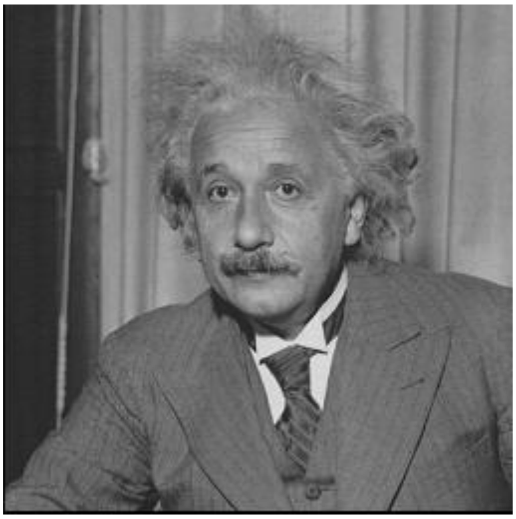
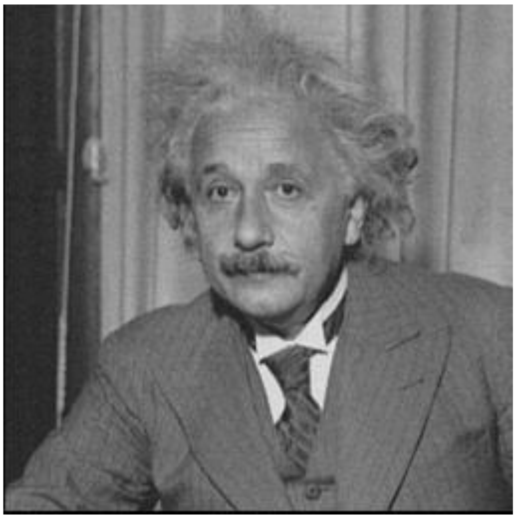




Αρχική Εικόνα   Ανακατασκευασμένη εικόνα

Η ποιότητα της ανακατασκευασμένης εικόνας είναι ακόμη πολύ καλή και δύσκολα διακρίνεται διαφορά ποιότητας από την αρχική. Ο ρυθμός κωδικοποίησης πηγής υπολογίστηκε από τον αλγόριθμο σε 4.03 bits/symbol.

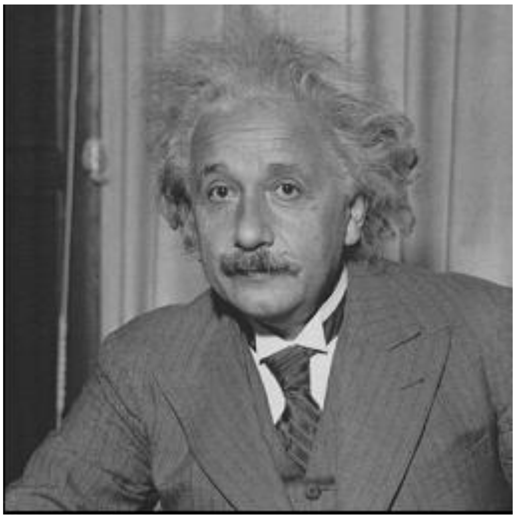
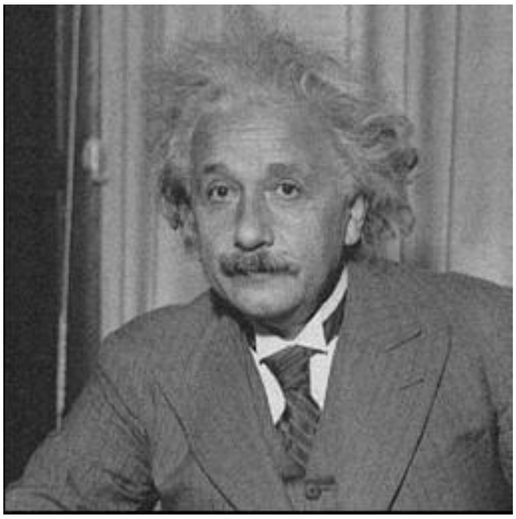
(iii) Βήμα κβάντισης 5 και πιθανότητα αλλαγής ψηφίου 0 (αθόρυβο κανάλι)

Αρχική Εικόνα   Ανακατασκευασμένη εικόνα

Η ποιότητα της ανακατασκευασμένης εικόνας είναι ακόμη πολύ καλή και δύσκολα διακρίνεται διαφορά ποιότητας από την αρχική. Ο ρυθμός κωδικοποίησης πηγής υπολογίστηκε από τον αλγόριθμο σε 3.80 bits/symbol.

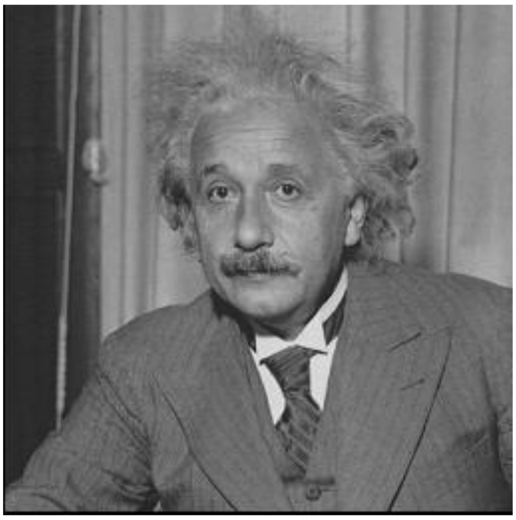
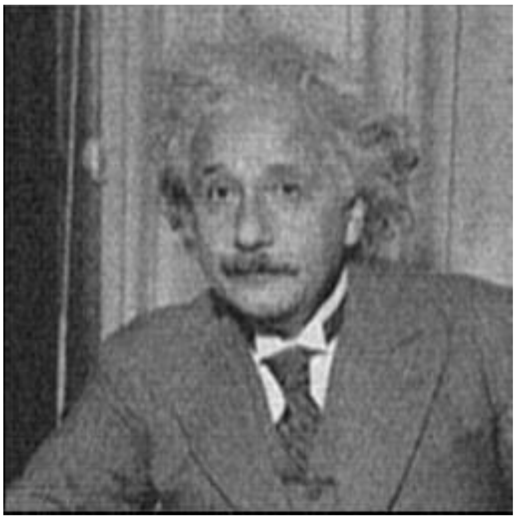
(iv) Βήμα κβάντισης 20 και πιθανότητα αλλαγής ψηφίου 0 (αθόρυβο κανάλι)

Αρχική Εικόνα   Ανακατασκευασμένη εικόνα

Η ποιότητα της ανακασκευασμένης εικόνας είναι σχετικά καλή και σίγουρα η επίδραση της κβάντισης έχει αρχίσει να γίνεται αισθητή. Ο ρυθμός κωδικοποίησης πηγής έχει μειωθεί σε 3.33 bits/symbol!

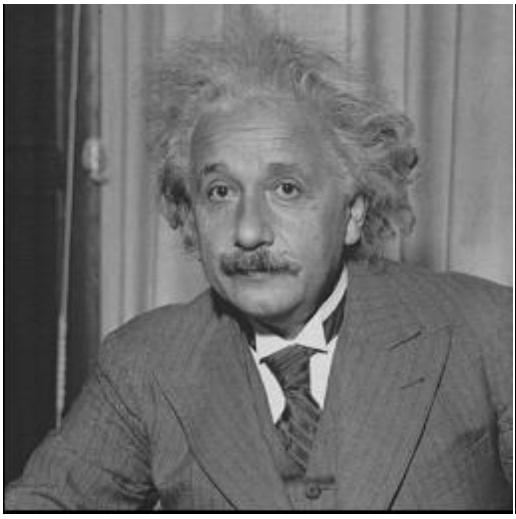
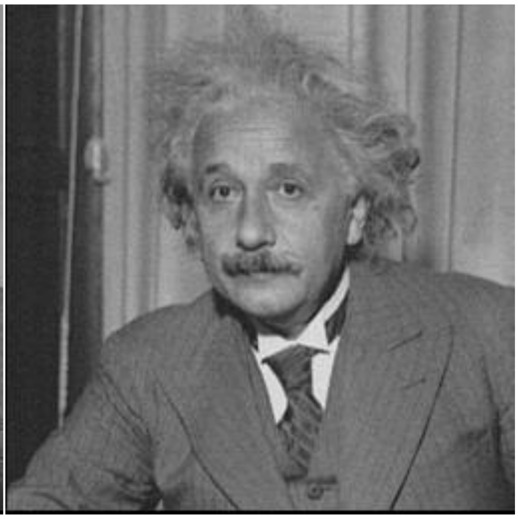
(v) Βήμα κβάντισης 50 και πιθανότητα αλλαγής ψηφίου 0 (αθόρυβο κανάλι)

Αρχική Εικόνα Ανακατασκευασμένη εικόνα

Η ποιότητα της ανακασκευασμένης εικόνας είναι μέτρια έως κακή, ωστόσο ακόμη μπορούμε να καταλάβουμε ποιος είναι ο εικονιζόμενος! Ο ρυθμός κωδικοποίησης πηγής έχει μειωθεί σε 3.10 bits/symbol!

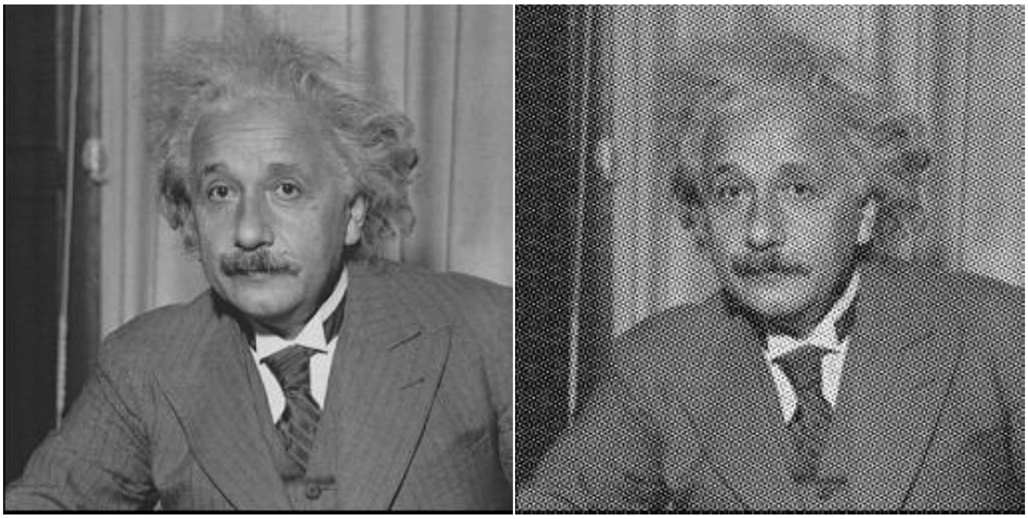
(vi) Βήμα κβάντισης 1 και πιθανότητα αλλαγής ψηφίου 1e-6 (ελάχιστα θορυβώδες κανάλι)

Αρχική Εικόνα Ανακατασκευασμένη εικόνα

Παρατηρούμε ότι με πιθάνοτητα αλλαγής 10^{-6} δεν επηρεάζεται η ποιότητα της ανακατασκευασμένης εικόνας

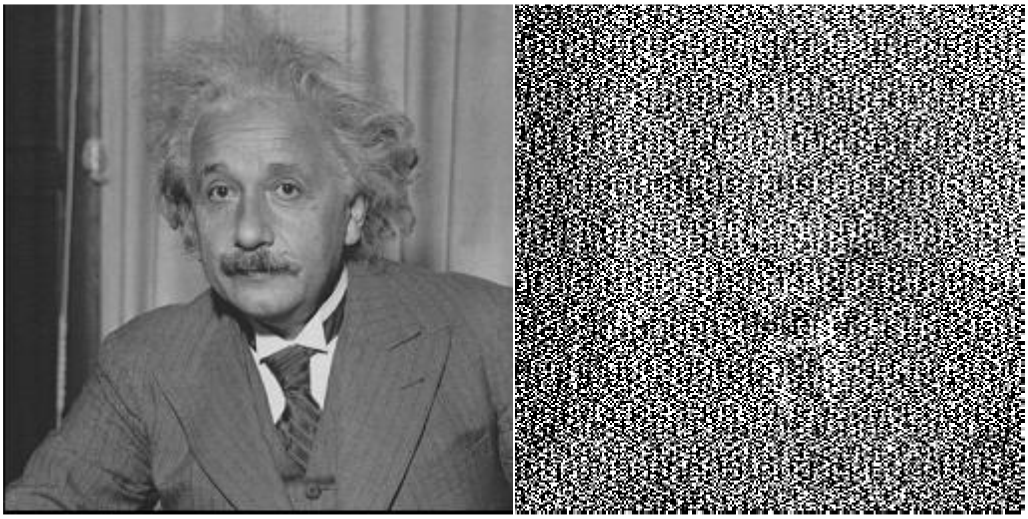
(vii) Βήμα κβάντισης 1 και πιθανότητα αλλαγής ψηφίου 1e-5 (ελάχιστα θορυβώδες κανάλι)



Αρχική Εικόνα Ανακατασκευασμένη εικόνα

Παρατηρούμε ότι με πιθάνοτητα αλλαγής 10^{-5} επηρεάζεται η ποιότητα της ανακατασκευασμένης εικόνας

(viii) Βήμα κβάντισης 1 και πιθανότητα αλλαγής ψηφίου 1e-5 (ελάχιστα θορυβώδες κανάλι)



Παρατηρούμε ότι με πιθάνοτητα αλλαγής 10^{-4} η ανακατασκευασμένη εικόνα δεν διακρίνεται καν! Επομένως για πιθανότητες αλλαγής ψηφίου μεγαλύτερες από 10^{-6} είναι επιτακτική η ανάγκη χρήσης ενός σχήματος κωδικοποίησης καναλιού.

ΚΕΦΑΛΑΙΟ 6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Από τις εργαστηριακές ασκήσεις που παρουσιάστηκαν προκύπτουν τα εξής συμπεράσματα:

(1) Είναι εύχρηστοι οι γεωμετρικοί μετασχηματισμοί (αλλαγή μεγέθους, περικοπή, περιστροφή και κύρτωση εικόνας) με χρήση του MATLAB

(2) Από τα αποτελέσματα της βελτίωσης της αντίθεσης εικόνας βλέπουμε ότι στο εξισορροπημένο ιστόγραμμα οι τιμές δεν συγκεντρώνονται γύρω από κάποιο συγκεκριμένο σημείο αλλά διασπείρονται σε όλο το φάσμα τιμών [0,255]. Επομένως στην εξισορροπημένη εικόνα η αντίθεση της εικόνας είναι ισοσκελισμένη. Ποσοτικά αυτή η εξισορρόπηση φαίνεται από τη μέτρηση κατά RMScontrast

(3) Το φιλτράρισμα μέσης τιμής (Mean-Value filtering) αποδίδει καλύτερα για την περίπτωση του Salt&Pepper θορύβου. Επίσης αποδίδει καλύτερα όταν το τετράγωνο μεγαλώνει (μεγάλες τιμές του N) αλλά αυτό δημιουργεί θολούρα στην εικόνα. Δεν παρατηρήθηκε κάποια διαφορά όταν μεταβάλλονταν το padding μεταξύ zero-padding και replicate-padding

(4) Το φιλτράρισμα μεσαίου (Median-Value filtering) έχει καλύτερη απόδοση από το φιλτράρισμα μέσης τιμής, ειδικά για την περίπτωση του θορύβου Salt&Pepper τα αποτελέσματα είναι εντυπωσιακά! Όσο αυξάνει το μέγεθος του παραθύρου, τόσο αντιμετωπίζεται πιο αποτελεσματικά ο θόρυβος, εισάγεται όμως στην εικόνα μία σχετικά θολούρα

(5) Το Γκαουσιανό φίλτρο ομαλοποίησης παρουσιάζει πολύ καλύτερα αποτελέσματα για την περίπτωση του θορύβου Gauss, ακόμη και όταν αυτός είναι αυξημένος. Εισάγεται όμως πάλι μία σχετική θολούρα

(6) Στο παράδειγμα της μετάδοσης παρατηρούμε ότι η αναπαραγόμενη εικόνα στο δέκτη έχει πάρα πολύ καλή ποιότητα, ακόμη και όταν η κβάντιση δεν είναι λεπτομερειακή. Παρατηρούμε επίσης μικρή ανθεκτικότητα στο θόρυβο του καναλιού, κάτι που κάνει την ανάγκη για κωδικοποίηση καναλιού επιτακτική. Παρατηρούμε επίσης μεγάλη εξοικονόμηση σε μεταδιδόμενο αριθμό ψηφίων ανά σύμβολο (bits per

source symbol) σε σχέση με τη περίπτωση που δεν υπάρχει καθόλου κωδικοποίηση πηγής.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Addison – Wesley Publishing Company, Inc, “An Introduction to Computer Graphics Concepts (Εισαγωγή στα Computer Graphics)”, Εκδόσεις Anubis (1993).

[2] Αθανάσιος Στυλιάδης, “Γραφικά με Ηλεκτρονικό Υπολογιστή (Computer Graphics)”, Εκδόσεις Ζήτη, (1999).

[3] Li J., “A Wavelet Aproach to Edge Detection”, Thesis for the Master of Science in Mathematics, Sam Houston State University, Huntsville, Texas (2003).

[4] Κωνσταντίνος Παπαρρίζος, “Matlab 6.5”, Εκδόσεις Ζυγός, Θεσσαλονίκη (2004).

[5]Ευάγγελος Χατζίκος, “MATLAB 6 για μηχανικούς”, Εκδόσεις Τζιόλα (2003).

[6] Κόλλιας Σ. Δ. (2001) Επεξεργασία, ανάλυση και τεχνολογία εικόνων και Βίντεο, Σημειώσεις ΕΜΠ, Αθήνα 2001

[7] Anoraganingrum D.(1999) “Cell segmentation with median filter and Mathematical Morphology Operation”, Proceedings, International Conference on Image Analysis and Processing, pp. 1043-1046

[8] Gonzalez R. C., Woods R. E., Eddins S. L.(2004), Digital Image Processing Using MATLAB, Prentice Hall, Upper Sandle River, NJ

[9] "Digital Image Processing Using Matlab, 2e". Gonzales, Woods and Edding. Publishing: 2nd edition 2009. ISBN 0982085400