



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

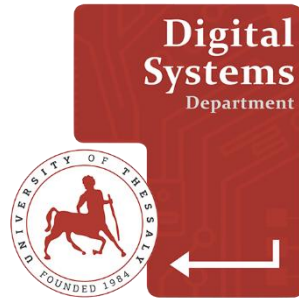
**Αποθετήριο Εργαστηριακών Α-
σκήσεων στην Επεξεργασία Εικό-
νας και Video για την Τριτοβάθ-
μια Εκπαίδευση**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΡΙΑΔΝΗ ΔΗΜΗΤΡΑ ΚΩΝΣΤΑΝΤΟΥΛΑ (312057)

Επιβλέπων: Απόστολος Ξενάκης, Επίκουρος καθηγητής

ΛΑΡΙΣΑ, 12/2024



UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

**Image and Video Processing Lab
Exercises Repository for Higher
Education Activities**

BACHELOR THESIS

ARIADNI DIMITRA KONSTANTOULA (3120157)

Supervisor: Apostolos Xenakis, Assistant Professor

LARISSA, 12/2024

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

ΑΡΙΑΔΝΗ ΔΗΜΗΤΡΑ ΚΩΝΣΤΑΝΤΟΥΛΑ

16/12/2024

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων	Απόστολος Ξενάκης Επίκουρος καθηγητής Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος 1	Ηλίας Σάββας Καθηγητής Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος 2	Κωνσταντίνος Χαϊκάλης Αναπληρωτής καθηγητής Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 21-03-2024

Περίληψη

Η παρούσα πτυχιακή εργασία επικεντρώνεται στη δημιουργία ενός αποθετηρίου εργαστηριακών ασκήσεων για την επεξεργασία εικόνας και βίντεο, σχεδιασμένου για χρήση στην τριτοβάθμια εκπαίδευση. Στόχος είναι η παροχή δομημένου εκπαιδευτικού υλικού που επιτρέπει στους φοιτητές να κατανοήσουν βασικές και προχωρημένες τεχνικές στην ψηφιακή επεξεργασία εικόνας μέσω του λογισμικού MATLAB.

Αρχικά, παρουσιάζονται οι βασικές αρχές της ψηφιακής εικόνας, συμπεριλαμβανομένων των τύπων εικόνων, του βάθους χρώματος και των μεθόδων δειγματοληψίας και κβαντοποίησης. Στη συνέχεια, αναλύονται θεμελιώδεις τεχνικές επεξεργασίας, όπως η εξομάλυνση, η ενίσχυση της αντίθεσης και η εφαρμογή φίλτρων. Ιδιαίτερη έμφαση δίνεται στην ανάλυση ιστογραμμάτων και στις μεθόδους συμπίεσης εικόνας, συμπεριλαμβανομένης της κωδικοποίησης Huffman.

Η εργασία περιλαμβάνει πρακτικές ασκήσεις που επιτρέπουν στους φοιτητές να εφαρμόσουν θεωρητικές γνώσεις, χρησιμοποιώντας εργαλεία του MATLAB για ανάλυση και μετατροπή εικόνων. Οι εργαστηριακές ασκήσεις είναι διαβαθμισμένης δυσκολίας και καλύπτουν από βασικά έως πιο προχωρημένα θέματα, ώστε να διευκολύνουν την εκμάθηση και την αυτοαξιολόγηση.

Τέλος, παρουσιάζονται συμπεράσματα και προτάσεις για περαιτέρω ανάπτυξη του αποθετηρίου, όπως η ενσωμάτωση τεχνολογιών μηχανικής μάθησης για βελτιωμένη ανάλυση εικόνας και η δημιουργία διαδραστικών πλατφορμών για την εκπαίδευση. Το αποθετήριο αυτό στοχεύει στη βελτίωση της εκπαιδευτικής διαδικασίας, παρέχοντας ένα πλήρες, πρακτικό και προσαρμόσιμο εργαλείο μάθησης για τους φοιτητές της επεξεργασίας εικόνας και βίντεο.

Abstract

This thesis focuses on the development of a repository of laboratory exercises for image and video processing, designed for use in higher education. The goal is to provide structured educational material that enables students to understand fundamental and advanced techniques in digital image processing using MATLAB.

Initially, the thesis presents the fundamental principles of digital images, including image types, color depth, and sampling and quantization methods. Then, essential processing techniques such as smoothing, contrast enhancement, and filtering are analyzed. Special emphasis is given to histogram analysis and image compression methods, including Huffman coding.

The study includes practical exercises that allow students to apply theoretical knowledge using MATLAB tools for image analysis and transformation. The laboratory exercises are structured progressively, covering both basic and advanced topics to facilitate learning and self-assessment.

Finally, conclusions and recommendations for further development of the repository are presented, such as integrating machine learning technologies for enhanced image analysis and creating interactive platforms for education. This repository aims to improve the educational process by providing a comprehensive, practical, and adaptable learning tool for students in image and video processing.

Ευχαριστίες

Εκφράζω την ειλικρινή μου ευγνωμοσύνη στην οικογένειά μου για την αμέριστη στήριξη, την υπομονή και την ενθάρρυνση καθ' όλη τη διάρκεια των σπουδών μου. Η σταθερή τους παρουσία και η πολύτιμη βοήθειά τους υπήρξαν καθοριστικοί παράγοντες στην ολοκλήρωση αυτής της προσπάθειας. Θερμές ευχαριστίες απευθύνω και στους φίλους μου, για τη διαρκή υποστήριξη, την κατανόηση και την ενθάρρυνση που μου παρείχαν σε κάθε στάδιο της διαδρομής μου. Η συμπαράστασή τους υπήρξε πολύτιμη, τόσο σε ακαδημαϊκό όσο και σε προσωπικό επίπεδο. Ιδιαίτερη μνεία αξίζει στον επιβλέποντα καθηγητή μου, κ. Απόστολο Ξενάκη, για την καθοδήγηση, τις επιστημονικές συμβουλές και τη διαρκή υποστήριξή του. Η συμβολή του ήταν καθοριστική στη διαμόρφωση και ολοκλήρωση της παρούσας πτυχιακής εργασίας.

Αριάδνη Δήμητρα Κωνσταντούλα

16/12/2024

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	VII
ABSTRACT	IX
ΕΥΧΑΡΙΣΤΙΕΣ.....	XI
ΠΕΡΙΕΧΟΜΕΝΑ.....	XIII
1 ΕΙΣΑΓΩΓΗ	1
2 ΟΡΙΣΜΟΣ	3
2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	5
3 ΜΕΓΕΘΗ ΚΑΙ ΔΙΑΣΤΑΣΕΙΣ ΕΙΚΟΝΑΣ	9
3.1 ΒΑΘΟΣ ΧΡΩΜΑΤΟΣ	10
3.1.1 <i>Binary image: Δυαδική εικόνα</i>	11
3.1.2 <i>Greyscale image: Μονοχρωματική με αποχρώσεις του γκρι</i>	12
3.1.3 <i>RGB image : Έγχρωμη εικόνα</i>	13
3.1.4 <i>Μετατροπή έγχρωμης εικόνας σε grayscale μέσω του MATLAB</i> 15	
3.1.5 <i>Μετατροπές τύπων εικόνων με τη χρήση του MATLAB</i>	15
4 ΣΥΜΠΙΕΣΗ ΕΙΚΟΝΑΣ	16
4.1 ΜΟΡΦΕΣ ΣΥΜΠΙΕΣΗΣ	17
4.2 ΚΩΔΙΚΟΠΟΙΗΣΗ ΕΝΤΡΟΠΙΑΣ.....	18
4.3 ΟΙ ΠΟΛΥΤΙΜΕΣ ΠΤΥΧΕΣ ΤΗΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΕΝΤΡΟΠΙΑΣ	19
4.4 ΚΩΔΙΚΟΠΟΙΗΣΗ ΠΗΓΗΣ	19
4.4.1 <i>Κωδικοποίηση μετασχηματισμού</i>	20
4.4.2 <i>Διαφορική ή προβλεπτική κωδικοποίηση</i>	20
4.4.3 <i>Διανυσματική κβαντοποίηση</i>	21
4.5 ΚΩΔΙΚΟΠΟΙΗΣΗ HUFFMAN.....	21
5 ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ.....	22

5.1	ΑΝΑΛΥΣΗ ΕΙΚΟΝΑΣ.....	22
5.1.1	Κατωφλίωση ή <i>Thresholding</i>	23
5.1.2	Αφαίρεση περιγράμματος	24
5.1.3	Διαμερισμός εικόνας.....	24
5.1.4	Σκελετοποίηση.....	25
5.1.5	Αραίωση.....	26
5.1.6	Κλάδεμα.....	26
5.2	ΑΝΑΓΝΩΣΗ ΕΙΚΟΝΑΣ	26
5.3	ΕΓΓΡΑΦΗ ΕΙΚΟΝΑΣ.....	28
5.4	ΑΠΕΙΚΟΝΙΣΗ ΕΙΚΟΝΑΣ	30
5.5	ΣΚΟΠΟΣ ΚΑΙ ΧΡΗΣΙΜΟΤΗΤΑ.....	34
5.6	ΠΛΗΡΟΦΟΡΙΕΣ ΕΙΚΟΝΑΣ	35
5.7	ΠΕΡΙΚΟΠΗ ΕΙΚΟΝΑΣ	36
5.8	ΠΕΡΙΣΤΡΟΦΗ ΕΙΚΟΝΑΣ.....	38
5.8.1	Πως λειτουργεί η περιστροφή.....	39
5.8.2	Διαδικασία δημιουργίας του πίνακα συντεταγμένων	39
5.8.3	Στρογγυλοποίηση συντεταγμένων.....	40
5.8.4	Προβλήματα και κενά στην εικόνα.....	40
5.8.5	Αντιμετώπιση κενών	41
5.8.6	Μετασχηματισμοί και συσχέτιση	41
5.8.7	Διατήρηση της ποιότητας.....	41
5.9	ΙΣΤΟΓΡΑΜΜΑ	44
5.9.1	Εφαρμογές ιστογράμματος.....	46
5.9.2	Διάγνωση ελλείψεων.....	46
5.9.3	Ορισμός ιστογράμματος.....	46
5.10	ΙΣΤΟΓΡΑΜΜΑ ΧΡΩΜΑΤΟΣ	49
5.11	ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ ΕΞΙΣΟΡΡΟΠΗΣΗΣ ΙΣΤΟΓΡΑΜΜΑΤΟΣ	50
6	ΕΠΕΞΕΡΓΑΣΙΑ ΒΙΝΤΕΟ.....	54
6.1	ΤOOLBOXES ΚΑΙ ΣΥΝΑΡΤΗΣΕΙΣ.....	54
6.2	ΕΝΤΟΠΙΣΜΟΣ ΚΙΝΗΣΗΣ ΚΑΙ ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΣΤΟ ΒΙΝΤΕΟ	58
6.3	ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΚΙΝΟΥΜΕΝΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ	59
6.3.1	Μέθοδοι ροής οπτικής.....	60
6.3.2	Μέθοδος διαφοράς φόντου	61

6.3.3	Μέθοδος διαφοράς γειτονικών καρέ.....	62
6.4	ΜΗΧΑΝΙΣΜΟΣ ΑΝΙΧΝΕΥΣΗΣ ΑΚΜΩΝ	63
6.5	ΑΝΙΧΝΕΥΣΗ ΚΙΝΗΣΗΣ	65
6.5.1	Μεθοδολογία	66
6.5.2	Σύστημα καταγραφής βίντεο	67
6.5.3	Αλγόριθμος εξαγωγής προσκηνίου	67
6.6	ΕΦΑΡΜΟΓΗ.....	68
6.6.1	Επεξεργασία βίντεο.....	70
6.6.2	Διαχωρισμός καρέ.....	70
6.6.3	Παρακολούθηση εικόνων.....	71
7	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	71
	ΒΙΒΛΙΟΓΡΑΦΙΑ	73
	ΠΑΡΑΡΤΗΜΑ Α.....	77

1 Εισαγωγή

Στην παρούσα εποχή, η αναπαράσταση της πληροφορίας δεν περιορίζεται μόνο στα πολυμέσα, αλλά εκφράζει το κύριο στοιχείο των σύγχρονων μέσων επικοινωνίας. Με βάση την έρευνα που αφορά τις διαδικασίες μάθησης, παρατηρείται ότι ο άνθρωπος απορροφά τις πληροφορίες κυρίως μέσω της οπτικής αντίληψης. Συνεπώς, η χρήση κατάλληλων εικόνων θεωρείται ένα ισχυρό εργαλείο πληροφόρησης.

Στον κλάδο των πολυμέσων, γίνεται χρήση διάφορων μορφών απεικόνισης για να αναπαριστήσουν την εικόνα ή το βίντεο που συνυπολογίζεται στις διάφορες εφαρμογές. Ομολογώ, η "στατική απεικόνιση" αναφέρεται σε μια εικόνα που δεν παρουσιάζει κίνηση, ενώ η "κινούμενη απεικόνιση" αναφέρεται σε μια εικόνα που παρουσιάζει κίνηση ή αλλαγή μέσω της τεχνικής του animation ή του βίντεο. Οι διάφορες μορφές απεικόνισης ενσωματώνονται στα πολυμέσα με σκοπό να επιτευχθούν διάφοροι στόχοι.

Μέσω της χρήσης εξειδικευμένων αισθητήρων, η εικόνα αποτυπώνει διάφορες μορφές, όπως ένα τοπίο, και τίθεται σε μια διαδικασία δειγματοληψίας. Για να είναι οποιαδήποτε πληροφορία συμβατή με ένα πληροφοριακό σύστημα, πρέπει πρώτα να υποβληθεί σε ψηφιοποίηση. Το μέσο που παρέχει τις οπτικές πληροφορίες συντίθεται από οπτικές ακτίνες, οι οποίες είναι αναλογικής φύσης. Επιπροσθέτως, άλλες μορφές δειγματοληψίας εμπεριέχουν τη χρήση οπτικών σαρωτών για την ψηφιοποίηση εκτυπωμένων φωτογραφιών ή τη χρήση δορυφόρων τηλεπισκόπησης για τη χαρτογράφηση τεράστιων περιοχών της επιφάνειας της Γης από μεγάλες αποστάσεις.

Μια εικόνα ψηφιακής μορφής ορίζεται ως μια σύνθεση εικονοστοιχείων οργανωμένη σε σχηματισμό μήτρας, που κωδικοποιεί μια οπτική απεικόνιση. Κατά συνέπεια, όλες οι ιδιότητες που σχετίζονται με τις ψηφιακές εικόνες συνδέονται με αυτή τη διάταξη και το μεμονωμένο εικονοστοιχείο. Αυτά τα γνωρίσματα περιέχουν χωρική ανάλυση το βάθος χρώματος και τον τύπο αποθήκευσης.

2 Ορισμός

Μια γνήσια εικόνα μπορεί να αλλάξει σε ψηφιακή αναπαράσταση με την ανάθεση αριθμητικών τιμών σε κάθε pixel. Αυτά τα pixel, τα οποία είναι μικρές περιοχές μέσα στην εικόνα, διαιρούνται και καταγράφονται από μια συσκευή απεικόνισης. Οι καταγε-γραμμένοι αριθμοί ή σύνολα αριθμών εκφράζουν συγκεκριμένες ιδιότητες κάθε pixel, όπως φωτεινότητα ή χρώμα. Για να διατηρηθεί η χωρική διάταξη της εικόνας, αυτοί οι αριθμοί οργανώνονται σε μια δομή που μοιάζει με πλέγμα, με γραμμές και στήλες που αντιστοιχούν στην κάθετη και οριζόντια θέση των εικονοστοιχείων. Επιπροσθέτως, η ψηφιακή απεικόνιση συγκροτεί τον αριθμητικό προσδιορισμό μιας δυαδικής εικόνας, καθώς οι τιμές των (M,N) συναρτώνται από το είδος της αυτής εικόνας. Στην πράξη, κάθε απεικόνιση απαιτείται να μετασχηματιστεί προς τη ψηφιακή μορφή προτού υπο-στεί κάποια μορφή ψηφιακής επεξεργασίας. Η διαδικασία αυτή επικεντρώνεται στη α-ναβάθμιση της ποιότητας της εικόνας, με στόχο την επίτευξη μείζονος σαφήνειας και ευκρίνειας, καθώς και στον αποτελεσματικό χειρισμό των δεδομένων που περιέχει, που είναι συχνά υποκείμενα σε απώλειες κατά τη μεταφορά τους από αισθητήρια και κατα-γραφικά μέσα. Έτσι, είναι αναγκαίο να ληφθούν εξίσου αποτελεσματικά δείγματα της συνάρτησης $I(M,N)$ σε κάθε σημείο M και N .

Η πυκνότητα των δειγμάτων καθορίζεται από τον νόμο της δειγματοληψίας. Η διάσταση μεταξύ δύο διαδοχικών δειγμάτων στον χώρο (M,N) πρέπει να είναι μικρότερη από το μισό του χρόνου των πιο γρήγορων αλλαγών της εικόνας. Κατά τα άλλα, θα πρέπει να προκρίνουμε δείγματα με επαρκή ταχύτητα για να επιλύσουμε τις ταχείες αλλαγές στη φωτεινότητα της εικόνας. Στη συνέχεια, τα δείγματα θα πρέπει να διακριθούν σε πεπερασμένο αριθμό σταθμών. Συνήθως, τα επίπεδα που χρησιμοποιούνται κυμαίνονται συνήθως στο $255 (2^8 - 1)$. Στο υψηλότερο επίπεδο, το 255, αντιστοιχεί το λευκό, ενώ στο χαμηλότερο, το 0, το μαύρο. Επομένως, ένας δυαδικός αριθμός με μήκος 8 bits (1 byte) αρκεί για να αναπαριστήσουμε την οξύτητα της αμαύρωσης ενός δείγματος της εικόνας, γνωστού και ως εικονοστοιχείο.

Οι ψηφιακές εικόνες φέρουν αρκετά χαρακτηριστικά. Ένα από αυτά είναι ο τύπος της εικόνας. Για παράδειγμα, μια εικόνα ασπρόμαυρης συγκαταλέγει μόνο την ένταση του φωτός που πέφτει στα pixels. Μια έγχρωμη εικόνα μπορεί να έχει τρία χρώματα, συνή-

θως RGB (Κόκκινο, Πράσινο, Μπλε) ή τέσσερα χρώματα, CMYK(Κυανό, Ματζέντα, Κίτρινο, μαύρο). Οι εικόνες RGB κατά κανόνα αξιοποιούνται σε οθόνες υπολογιστών και σαρωτές, ενώ οι εικόνες CMYK εφαρμόζονται σε εκτυπωτές έγχρωμων εκτυπώσεων. Υπάρχουν ομοίως μη-οπτικές εικόνες όπως οι υπέρηχοι ή η ακτινογραφία στις οποίες καταγράφεται η ένταση του ήχου ή των ακτίνων Χ. Στις εικόνες εύρους, καταγράφεται η διάσταση του pixel από τον παρατηρητή. Η ανάλυση εκφράζεται με τον αριθμό των pixels ανά ίντσα. Μια υψηλότερη ανάλυση δίνει μια πιο εμπεριστατωμένη εικόνα. Μια οθόνη υπολογιστή έχει συνήθως ανάλυση 100 ppi, ενώ ένας εκτυπωτής έχει ανάλυση που κυμαίνεται από 300 ppi έως περισσότερα από 1440 ppi. Γι' αυτό το λόγο κιόλας μια εικόνα φαίνεται πολύ καλύτερα στην εκτύπωση από ό,τι σε μια οθόνη.

Οι ψηφιακές εικόνες τείνουν να παράγουν σημαντικά και μεγάλα αρχεία και τακτικά δέχονται συμπίεση για να γίνουν μικρότερα. Η συμπίεση καταχράται το γεγονός ότι πολλά γειτονικά pixels στην εικόνα έχουν παρεμφερή χρώματα ή φωτεινότητα. Οι μέθοδοι συμπίεσης ποικίλλουν στην αποτελεσματικότητά τους και την ταχύτητά τους. Η μέθοδος GIF έχει καλή συμπίεση για εικόνες 8 bit, ενώ το JPEG είναι απώλειας ποιότητας, ήτοι προξενεί κάποια υποβάθμιση της εικόνας. Το προτέρημα του JPEG είναι η ταχύτητα, οπότε είναι κατάλληλο για κινούμενες εικόνες.

Ένα από τα προτερήματα των ψηφιακών εικόνων συγκριτικά των παραδοσιακών είναι η δυνατότητα να μεταδοθούν ηλεκτρονικά σχεδόν αμέσως και να μεταβληθούν εύκολα από ένα μέσο σε ένα άλλο, όπως από μια ιστοσελίδα σε μια οθόνη υπολογιστή ή έναν εκτυπωτή. Ένα μεγαλύτερο πλεονέκτημα είναι η δυνατότητα να τις τροποποιήσεις αντι-στοίχως με τις ανάγκες σου. Υπάρχουν πολλά συστήματα διαθέσιμα που παρέχουν στον χρήστη τη δυνατότητα να το κάνει αυτό, συμπεριλαμβανομένων του Photoshop, του Photopaint και του Gimp. Με ένα τέτοιο πρόγραμμα, ένας χρήστης μπορεί να αλλάξει τα χρώματα και τη φωτεινότητα μιας εικόνας, να διαγράψει ανεπιθύμητα αντιληπτά αντικείμενα, να μετακινήσει άλλα και να συνενώσει αντικείμενα από διάφορες εικόνες, μεταξύ πολλών άλλων λειτουργιών. Με αυτόν τον τρόπο, ένας χρήστης μπορεί να επεξεργαστεί φωτογραφίες οικογενειακών φωτογραφιών ή ακόμα και να δημιουργήσει νέες εικόνες. Άλλο λογισμικό, όπως τα επεξεργαστικά κειμένου και τα προγράμματα επεξεργασίας εκδόσεων, μπορεί εύκολα να εναρμονίσει ψηφιακές εικόνες με κείμενο για να παράγει βιβλία ή περιοδικά πολύ πιο εποικοδομητικά από τις παραδοσιακές διαδικασίες.

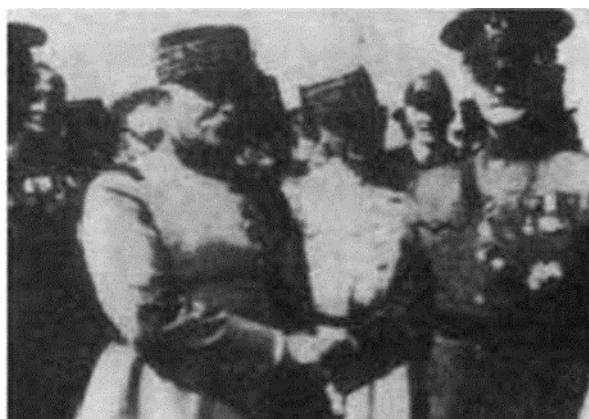
2.1 Ιστορική Αναδρομή

Μία από τις πρώιμες χρήσεις των ψηφιακών εικόνων προέκυψε στον τμήμα των εφημερίδων, όταν οι εικόνες αρχικά μετακινούνταν από το Λονδίνο στη Νέα Υόρκη διαμέσου υποθαλάσσιων αγωγών και καλωδίων. Το 1920, εξευρέθηκε το Bartlane σύστημα μετάδοσης εικόνας από τους Harry G. Bartholomew και Maynard D. McFarlane. Με το εν λόγω σύστημα, οι εικόνες μπορούσαν να εξαπλωθούν διαβαίνοντας τον Ατλαντικό Ωκεανό σε λιγότερο από τρεις ώρες, και αυτό με τη χρήση πέντε αποχρώσεων του γκρι. Είναι ενδιαφέρον να σημειωθεί ότι πριν την επινόηση αυτού του συστήματος, μια εικόνα απαιτούσε περισσότερο από μία εβδομάδα για να μεταδοθεί πέραν του Ατλαντικού. Το σύστημα Bartlane πρωτοεφαρμόστηκε για τη με-tάδοση μιας εικόνας μεταξύ του Λονδίνου και της Νέας Υόρκης το 1921. Κατά τη λήψη της εικόνας, χρησιμοποιήθηκε η τεχνική Half-toning για να γίνει άμεση εκτύπωσή της, χωρίς να καταχωρηθεί.



Εικόνα 1: Η εικόνα που εστάλη το 1921 μέσω του συστήματος Bartlane. [26]

Το 1929, διακρίθηκε η μετάδοση μιας εικόνας από το Λονδίνο προς τη Νέα Υόρκη, ακολουθούμενη από επείγουσα εκτύπωση χρησιμοποιώντας μέσα και τεχνικές φωτογραφικής αναπαραγωγής, αντί της τεχνικής Bartlane. Η εκτέλεση αυτών των τεχνικών έφερε σε αύξηση του αριθμού των αποχρώσεων του γκρι από 5 σε 15.



Εικόνα 2: Η εικόνα που διαβιβάστηκε το 1929 χρησιμοποιώντας φωτογραφικές τεχνικές αναπαραγωγής και 15 διαφορετικές αποχρώσεις του γκρι. [23]

Το 1964, για πρώτη φορά ψηφιακοί υπολογιστές καθιερώθηκαν στην κατεργασία εικόνων της σελήνης που ελήφθησαν από το διαστημόπλοιο Ranger 7 με μια φωτογραφική μηχανή. Ο σκοπός ήταν η κατάργηση των αλλοιώσεων που προέκυψαν λόγω της κίνησης του διαστημοπλοίου κατά τη λήψη των φωτογραφιών. Παράλληλα με τις εφαρμογές στον διαστημικό τομέα, οι τεχνικές ΕΕ ξεκίνησαν να εφαρμόζονται στην ιατρική εικόνα, στην παρακολούθηση απομακρυσμένων πόρων της Γης και στην αστρονομία κατά τα τέλη της δεκαετίας του 1960 και τις αρχές της δεκαετίας του 1970.



Εικόνα 3: Η πρώτη εικόνα του φεγγαριού που λήφθηκε από το αμερικανικό διαστημόπλοιο Ranger 7. [24]

Από τη δεκαετία του 1980 και μέχρι τις μέρες μας, η τεχνολογία των ψηφιακών εικόνων θα βιώσει αξιόλογη πρόοδο και θα εφαρμόζεται σε όλο και περισσότερους τομείς. Αυτό συμβαίνει καθότι τόσο οι υπολογιστές όσο και το αναγκαίο υλικό για την επεξεργασία εικόνας αρχίζουν να γίνονται πιο οικονομικά προσιτά και προσπελάσιμα εκείνη την περίοδο. Το 1994, εφαρμόζεται τεχνολογία ψηφιακών εικόνων για ιατρικές εφαρμογές στο Space Foundation στο Colorado. Το 2002, ο Raanan Fattal παρουσίασε το Gradient-Domain Image Processing, μια νέα μέθοδο επεξεργασίας ψηφιακών εικόνων, η οποία επαναπαύεται στις διαφορές μεταξύ των γειτονικών εικονοστοιχείων αντί για τις τιμές του κάθε εικονοστοιχείου.

3 Μεγέθη και διαστάσεις εικόνας

Η ψηφιακή αναπαράσταση, όπως περιγράφηκε, αντιπροσωπεύεται από ένα καθολικό σύνολο εικονοστοιχείων που οργανώνονται σε έναν πίνακα. Ένα από τα πρώτα και ουσιαστικότερα γνωρίσματα της εικόνας είναι οι διαστάσεις αυτού του πίνακα, οι οποίες δίνονται συνήθως ως πλάτος και ύψος σε pixel. Ο συνολικός αριθμός των pixel της εικόνας υπολογίζεται πολλαπλασιάζοντας τις αντίστοιχες διαστάσεις. Οι διαστάσεις μιας φωτογραφίας προβάλλουν το μέγεθος της εικόνας, το οποίο σχετίζεται με πολλές άλλες ιδιότητες και χαρακτηριστικά της. Κάποια από αυτά είναι τα εξής:

- Η μείζονα διάσταση της εικόνας εγκρίνει τη φιλοξενία περισσότερων λεπτομερειών. Για παράδειγμα, μια εικόνα με διαστάσεις 200x300 που παρασταίνει ένα πρόσωπο δεν δύναται να καταλογίσει ικανοποιητικώς τα χαρακτηριστικά, καθώς το κάθε εικονοστοιχείο είναι αρκετά μικρό. Αντιστρόφως, μια εικόνα με διαστάσεις 2592 x 1944 που απεικονίζει το ίδιο
- Οι διαστάσεις μιας εικόνας επιδρούν στο μέγεθος του αρχείου που χρησιμοποιείται για την αποθήκευσή της. Μια παράσταση με διαστάσεις 200 x 300 προϋποθέτει μικρότερο αποθηκευτικό χώρο σε σύγκριση με μια εικόνα 2000 x 3000, καθώς απαιτεί μόνο το 1/100 του μεγέθους της δεύτερης. Αυτή η ιδιότητα οδήγησε στην ανάγκη ανάπτυξης διαφορετικών μορφοποιήσεων αποθήκευσης, οι οποίες μπορούν να είναι μη συμπιεσμένες, συμπιεσμένες χωρίς απώλεια πληροφoρίας ή συμπιεσμένες με απώλεια πληροφoρίας, αντιστοίχως με την περίπτωση.
- Οι συμβατικές διαστάσεις ψηφιακών εικόνων παρασύρονται τόσο από τις προ-σβάσιμες τεχνολογίες απεικόνισης όσο και από τις απαιτήσεις των διαφόρων μέσων. Ανώτερης ανάλυσης εικόνες απαιτούν εκτυπωτές που μπορούν να ανα-παράγουν χιλιάδες εικονοστοιχεία σε 1 τ.εκ. χαρτιού, ενώ η ανάπτυξη πάνελ 4K και αντίστοιχων βιντεοκαμερών υποχρεώνει τη θεμελίωση και διαχείριση βίντεο υψηλής ανάλυσης 3840x2160. Πριν από 10 χρόνια, οι αντίστοιχες αναλύσεις βίντεο ήταν πολύ χαμηλότερες, όπως 360x288 για το πρότυπο PAL και 360x240 για το πρότυπο NTSC.

- Η κατεργασία των ψηφιακών εικόνων βασίζεται στην ερμηνεία των γνωρίσμα-των κάθε εικονοστοιχείου και τη σχέση τους με τα κοντινά τους pixel. Αυτό προξενεί αυξη-μένες απαιτήσεις σε υπολογιστική ισχύ για την πραγματοποίηση των εργασιών επεξερ-γασίας εικόνας. Ως απόρροια, διακρίνεται η αυξημένη α-νάγκη για πιο ισχυρά υπολογι-στικά συστήματα και αποδοτικότερους αλγορίθ-μους επεξεργασίας. Είναι ενδεικτικό το γεγονός ότι οι υπολογιστές που χρησι-μοποιούνται για εργασίες επεξεργασίας εικόνας και βίντεο συνήθως πληρούν τις υψηλότερες τεχνικές προδιαγραφές.

Όλα τα προαναφερθέντα στοιχεία τεκμηριώνουν τη σπουδαιότητα αυτού του χαρακτη-ριστικού και επεξηγούν γιατί οι διαστάσεις μιας εικόνας συγκροτούν το πρώτο και βασι-κό στοιχείο που συνοδεύει και περιγράφει μια ψηφιακή εικόνα. Είναι άμεσα συνδεδεμέ-νες με τη χωρική διάσταση και συμπληρώνονται από την πυκνότητα των εικονοστοιχεί-ων.

3.1 Βάθος χρώματος

Το βάθος χρώματος (μιας έγχρωμης εικόνας) ή "bits ανά pixel" είναι ο αριθμός των bits στους αριθμούς που απεικονίζουν τη λαμπρότητα ή το χρώμα. Περισσότερα bits εγκρί-νουν την καταγραφή περισσότερων χρωιών του γκρι ή περισσότερων χρωμάτων. Για πα-ράδειγμα, μια εικόνα RGB με 8 bits ανά χρώμα έχει συνολικά 24 bits ανά pixel ("α-ληθινό χρώμα"). Κάθε bit μπορεί να εκπροσωπεί δύο πιθανοφανή χρώματα, οπότε έ-χουμε συνολικά 16,777,216 πιθανά χρώματα. Μια τυπική εικόνα GIF σε μια ιστοσελίδα έχει 8 bits για όλα τα χρώματα συνολικά, δηλαδή 256 χρώματα. Ωστόσο, είναι μια πολύ μικρότερη εικόνα από μια με 24 bit, συνεπώς κατεβαίνει πιο γρήγορα. Μια εικόνα φαξ έχει μόνο ένα bit ή δύο "χρώματα", μαύρο και λευκό. Η εμφάνιση της εικόνας παρέχει περισσότερες λεπτομέρειες αναφορικά με το πώς οργανώνονται οι αριθμοί στο αρχείο εικόνας, συμπεριλαμβανομένης της μορφής συμπίεσης που χρησιμοποιείται, αν υφίστα-ται. Ενδιάμεσα στις πιο δημοφιλείς από τις δεκάδες διαθέσιμες μορφές είναι το TIFF, το GIF, το JPEG, το PNG και το Post-Script.

- **Εικόνες 1 bit.** Ένα μόνο bit ενδέχεται να παράξει δύο άξιες καταστάσεις για κάθε pixel: είτε 0 είτε 1. Αυτό επιτρέπει τη διάκριση ανάμεσα δύο μόνο κατα-στάσεων, οι οποίες ισοδυναμούν στο λευκό και το μαύρο χρώμα. Ως ακολού-θως, η ψηφιακή εικόνα κωδικοποιείται ως ασπρόμαυρη, δίχως να υπάρχουν κλίμακες του γκρι.
- **Εικόνες 8 bit.** Σε αυτή την περίπτωση, χρησιμοποιείται ένα byte (= 8 bit) για να αναπαραστήσει τη λαμπρότητα ενός pixel. Κάθε byte μπορεί να έχει 256 δια-φορετικές

τιμές, που ισοδυναμούν σε αλλιώτικες νότες του γκρι. Έτσι, μια ψηφιακή εικόνα κωδικοποιείται χρησιμοποιώντας 256 πιθανές τιμές, εγκρίνοντας την αναπαράσταση 256 αποχρώσεων του γκρι.

- **Εικόνες 16 bit.** Όταν αναφερόμαστε σε μια εικόνα με βάθος χρώματος 16 bit, συνήθως την αναφέρουμε ως Highcolor. Από τα 16 bit ανακύπτουν $2^{16} = 65536$ διαφορετικές αξίες, οι οποίες αντιστοιχούν σε διαφορετικά χρώματα. Κατά κανόνα, η πληροφορία αυτή κατανέμεται με τέσσερα bits για κάθε ένα από τα χρώματα κόκκινο και μπλε, και πέντε bits για το πράσινο. Αυτό συμβαίνει επειδή το ανθρώπινο μάτι είναι πιο ευαίσθητο στις παραλλαγές του πράσινου σε σχέση με τα άλλα χρώματα. Σε άλλες περιπτώσεις, το 16 bit ενδέχεται να χρησιμοποιηθεί για να αναπαραστήσει τυχόν καθαρότητα του χρώματος.
- **Εικόνες 24 bit.** Ομόγνωμα με τις προγενέστερες περιπτώσεις, υπάρχουν $2^{24} = 16.777.216$ διαφορετικές διακυμάνσεις χρωμάτων που μπορούν να παρατηρηθούν. Αυτή η κατηγορία εικόνων είναι γνωστή και ως «truecolor». Τα 24 bit χωρίζονται σε τρία byte, ένα για κάθε ένα από τα πρωταρχικά χρώματα (Κόκκινο, Πράσινο, Μπλε-RGB), το καθένα από τα οποία χρησιμοποιείται για τις 256 διαφορετικές αποχρώσεις του.
- **Εικόνες 48 bit.** Με την κωδικοποίηση σε 2^{48} περίπου 256 τρισεκατομμύρια χρώματα, αυτό το βάθος χρώματος ξεπερνά την ικανότητα αντίληψης του ανθρώπινου ματιού. Συναποτελεί προτιμώμενη ανάλυση σάρωσης από πολλούς σαρωτές, χρησιμοποιούμενη ως μέγιστη ανάλυση.
- Όπως συνεπάγεται εμφανώς, η ποιότητα του χρώματος συνδέεται άμεσα με το βάθος του χρώματος, δηλαδή την πιστότητα αναπαράστασης του ουσιαστικού χρώματος. Κάθε φορά που πληθύνεται το βάθος χρώματος στην κωδικοποίηση μιας εικόνας, χρησιμοποιούνται περισσότερα bits για κάθε εικονοστοιχείο με επακόλουθο να είναι πιο ακριβής η αναπαράσταση του χρώματος στην πραγματική του μορφή.

3.1.1 Binary image: Δυαδική εικόνα

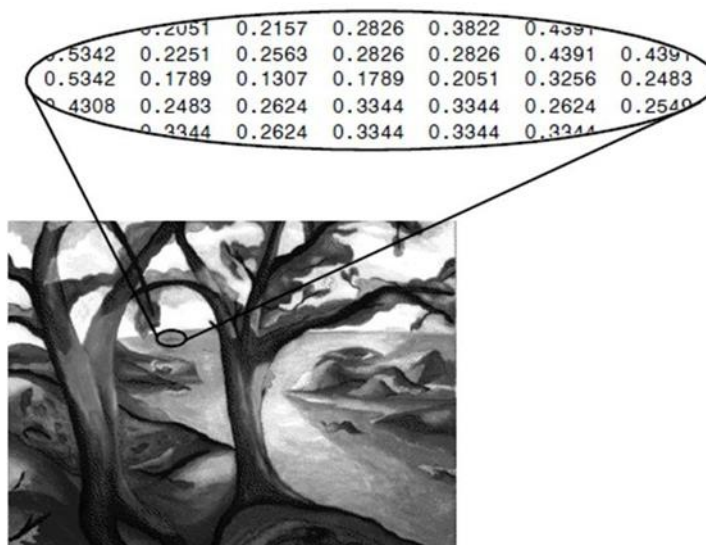
Ένα διαλεχτό γένος ασπρόμαυρης εικόνας, γνώριμη ως δυαδική, συνίσταται αποκλειστικά από τα χρώματα μαύρο και άσπρο. Σε μια τέτοια εικόνα, κάθε pixel δέχεται μία από τις δύο δυνατές τιμές (0 ή 1), αναπαριστώντας την κατάσταση on ή off. Οι δυαδικές εικόνες είναι συχνό φαινόμενο στην ψηφιακή επεξεργασία εικόνας και ανακύπτουν συνήθως από διάφορες μελέτες όπως η εφαρμογή κατωφλίου. Ένα παράδειγμα αυτού του είδους εικόνας παρουσιάζεται παρακάτω.



Εικόνα 4: Παράδειγμα δυαδικής εικόνας. [27]

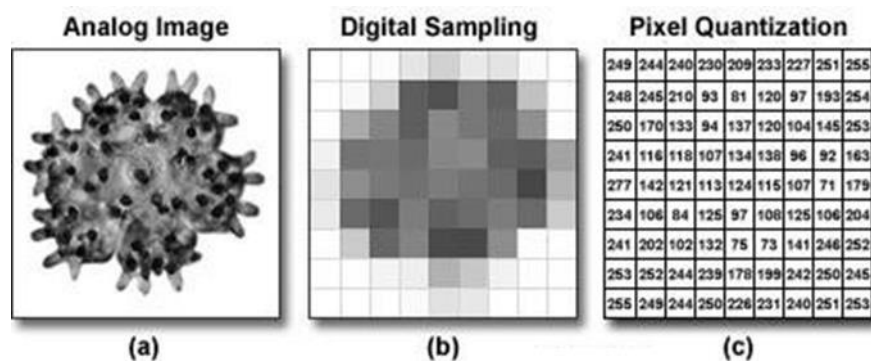
3.1.2 Greyscale image: Μονοχρωματική με αποχρώσεις του γκρι

Στην τεχνολογία φωτογραφίας και πληροφορικής, μια εικόνα σε κλίμακα του γκρι περιγράφει την οξύτητα του φωτός σε κάθε pixel με ένα μόνο δείγμα. Αυτές οι εικόνες, γνωστές και ως ασπρόμαυρες, εμπεριέχουν μόνο αποχρώσεις του γκρι, κατηγοριοποίηση από το πιο σκούρο μαύρο προς το πιο φωτεινό λευκό, αντιστοιχίζοντας την ανίσχυρη έως την ισχυρή ένταση του φωτός. Το MATLAB αποθηκεύει την εικόνα μονοχρωμίας σε έναν απλό πίνακα, όπου κάθε κελί ισοδυναμεί σε ένα μοναδικό pixel. Ο πίνακας μπορεί να είναι τύπου double, uint8 ή uint16. Τα στοιχεία του πίνακα εκφράζουν διάφορα επίπεδα φωτεινότητας (τα επίπεδα του γκρι), όπου το επίπεδο φωτεινότητας 0 αντιστοιχεί στο μαύρο χρώμα, ενώ τα επίπεδα φωτεινότητας 1, 255 ή 65535 αντιστοιχούν στο λευκό. Παρακάτω εμφανίζεται μια εικόνα ως παράδειγμα για τα παραπάνω.



Εικόνα 5: Παράδειγμα μιας γκρι εικόνας. [3]

Αν και είναι δυνατόν να αποθηκεύσουμε ψηφιακές εικόνες ως μονόχρωμες (grayscale), ακόμη και οι εικόνες που περικλείουν χρώμα περιέχουν πληροφορία σε μονόχρωμη κλίμακα. Αυτό οφείλεται στο γεγονός ότι κάθε εικονοστοιχείο μεταφέρει επίσης πληροφορία σχετικά με τη φωτεινότητά του, αυθυπόστατα από το χρώμα του. Η θεμελίωση μιας εικόνας με χρήση αισθητήρα ή άλλου εξοπλισμού λήψης εικόνας αναφέρεται στον προσδιορισμό της φωτεινότητας ή της έντασης του φωτός σε μια εικόνα ως μια συνάρτηση δύο διαστάσεων, $F(x, y)$, όπου (x, y) αναπαριστούν τις χωρικές συντεταγμένες. Κατά τη λήψη υπόψη λαμβάνεται μόνο η φωτεινότητα του φωτός.



Εικόνα 6: Η δημιουργία μια ψηφιακής grayscale εικόνας και η αναπαράσταση των εικονοστοιχείων με βάση την φωτεινότητα τους, α) Αναλογική εικόνα, β) Ψηφιακή Δειγματοληψία, γ) Κβαντισμός Εικονοστοιχείων. [3]

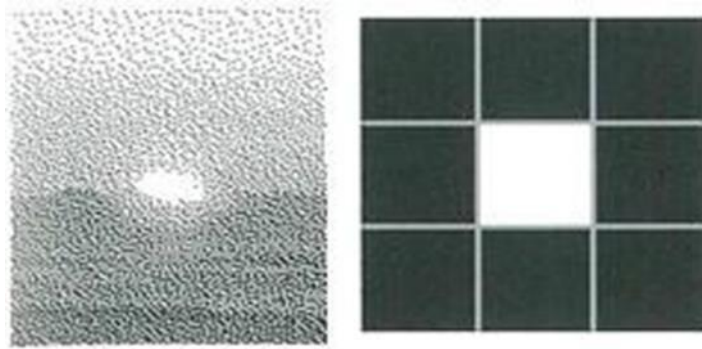
3.1.3 RGB image : Έγχρωμη εικόνα

Μια έγχρωμη εικόνα άπτεται σε μια ψηφιακή αναπαράσταση όπου κάθε pixel συμπεριλαμβάνει πληροφορίες αναλογικά με το χρώμα. Μια εικόνα RGB, ή γνωρίμη και ως εικόνα "πραγματικού χρώματος", αποθηκεύεται στο MATLAB σε μορφή ενός τρισδιάστατου πίνακα στοιχείων με διαστάσεις $m \times n \times 3$. Αυτός ο πίνακας προσδιορίζει τα επίπεδα του κόκκινου, του πράσινου και του μπλε για κάθε ανεξάρτητο pixel. Ένας πίνακας RGB μπορεί να υπάγεται σε μία από τις κλάσεις double, uint8 ή uint16. Σε έναν πίνακα κλάσης double, η τιμή κάθε χρώματος εκφράζεται ως μια τιμή μεταξύ 0 και 1. Ένα pixel με τιμή (0,0,0) περιγράφει το μαύρο χρώμα, ενώ ένα pixel με τιμή (1,1,1) αντιστοιχεί στο λευκό.



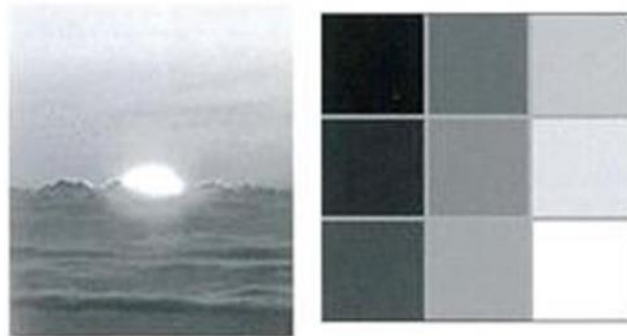
Εικόνα 7: Κατηγορίες εικόνων, binary, color, grayscale. [11]

Μια εικόνα που αξιοποιεί 1 bit βάθος χρώματος (0 ή 1) θα συμπεριλαμβάνει μόνο ασπρόμαυρα εικονοστοιχεία, δηλαδή pixels που είναι είτε λευκά είτε μαύρα. Μια τέτοια εικόνα είναι η ακόλουθη.



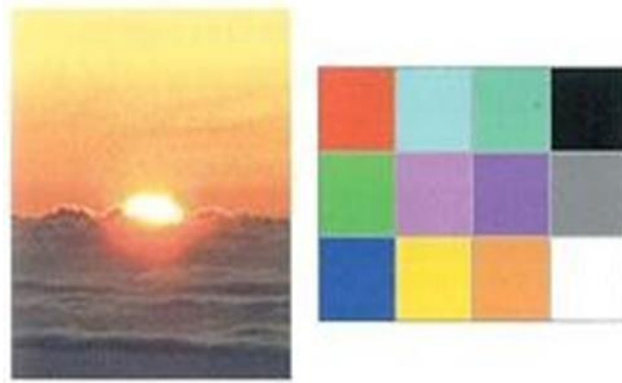
Εικόνα 8: Εικόνα που αξιοποιεί βάθος χρώματος 1 bit. [11]

Ομολόγως, μια εικόνα που χρησιμοποιεί 8 bit βάθος χρώματος (0 ή 1) θα περιέχει 256 διαβαθμίσεις του γκριζου. Μια τέτοια εικόνα εμφανίζεται στη συνέχεια.



Εικόνα 9: Εικόνα που αξιοποιεί βάθος χρώματος 8 bit. [11]

Για την κωδικοποίηση των χρωμάτων σε εικόνες με βάθος 24 bit, διατίθενται 8 bit για κάθε ένα από τα τρία πρωταρχικά χρώματα (κόκκινο, πράσινο και μπλε), συνολικά 24 bit.



Εικόνα 10: Εικόνα που αξιοποιεί βάθος χρώματος 24 bit. [11]

3.1.4 Μετατροπή έγχρωμης εικόνας σε grayscale μέσω του MATLAB

Για να τροποποιήσουμε μια έγχρωμη ή μια ασπρόμαυρη εικόνα σε grayscale, εκμεταλλευόμαστε το λογισμικό MATLAB προκειμένου να δημιουργήσουμε έναν πίνακα από εικονοστοιχεία grayscale. Αυτός ο πίνακας τυπώνει τα επίπεδα φωτεινότητας των εικονοστοιχείων που περιέχονται στην αρχική εικόνα. Κάθε εικονοστοιχείο αναπαρίσταται ως ακέραιος αριθμός φωτεινότητας που κυμαίνεται από το 0 έως το 255 (8 bit). Ο παρακάτω κώδικας MATLAB μπορεί να εφαρμοστεί για τη τροποποίηση οποιασδήποτε εικόνας ανεξαρτήτως ανάλυσης σε εικόνα grayscale.

```
clear

input_image = imread('tulips.jpg');
image = rgb2gray(input_image);
figure;
imshow(image)
;
```

3.1.5 Μετατροπές τύπων εικόνων με τη χρήση του MATLAB

Στο χώρο του MATLAB ενδημούν ενσωματωμένες λειτουργίες για τη τροποποίηση μεταξύ αλλιώτικων χρωματικών πρότυπων. Αυτές οι λειτουργίες είναι οι εξής:

- **gray2ind:** μεταβάλλει μια εικόνα από τον χώρο της γκρι κλίμακας σε μια εικόνα με χρωματικό δείκτη. Αναλυτικά, επινοεί έναν πίνακα ευρετηρίου και έναν πίνακα δεικτών που ισοδυναμεί σε κάθε pixel της εικόνας με ένα χρώμα από την παρεχόμενη παλέτα. Αυτό αφήνει συνήθως την αποθήκευση εικόνων με μικρότερη μνήμη από εκείνη που απαιτείται για εικόνες RGB.

- **im2bw:** αλλάζει μια εικόνα σε δυαδική εικόνα με βάση ένα κατώφλι. Κάθε pixel της εικόνας είτε θα γίνει λευκό (1) εάν η έντασή του είναι μεγαλύτερη από το κατώφλι, είτε μαύρο (0) εάν είναι μικρότερη ή ίση με αυτό. Αυτή η δυαδική εικόνα χρησιμοποιείται συνήθως για χρήσεις επεξεργασίας εικόνας όπως η ανίχνευση αντικειμένων ή η ανάλυση σχημάτων.
- **ind2gray:** μετατρέπει μια εικόνα χρωματικού δείκτη σε μια εικόνα τόνου του γκρι κάνοντας χρήση μια παλέτα χρωμάτων. Κάθε pixel της εικόνας χρωματικού δείκτη περιλαμβάνει έναν αριθμό που ισοδυναμεί σε ένα χρώμα στην παλέτα.
- **mat2gray:** μετατρέπει μια εικόνα από οποιαδήποτε μορφή (π.χ. uint8, uint16, double) σε μια εικόνα απόχρωσης του γκρι στην κλίμακα (0,1), όπου 0 αντιστοιχεί στο μαύρο και 1 αντιστοιχεί στο λευκό. Αυτή η μεταβολή είναι ωφέλιμη για την κανονικοποίηση των εικόνων πριν από την εκτέλεση κάποιων επεξεργασιών, καθώς επίσης και για την εμφάνιση ή αποθήκευση εικόνων σε μορφές που προϋποθέτουν τιμές στο διάστημα (0,1).
- **rgb2gray:** μετατρέπει μια RGB εικόνα σε μια εικόνα κλίμακας του γκρι. Αυτή η τροποποίηση συντελείται με τον τύπο απόχρωσης γκρι που παίρνει υπόψη τις ανθρώπινες αντιλήψεις των χρωμάτων. Κάθε pixel της εικόνας RGB λαμβάνεται ως ένας τριάδα τιμών (R, G, B).
- **rgb2ind:** μεταβάλλει μια εικόνα RGB σε μια εικόνα με διευρυμένη παλέτα χρωμάτων αξιοποιώντας τον αλγόριθμο της κβαντοποίησης. Αυτή η μετατροπή συγκαταλέγει την ανίχνευση μιας αρμόζουσας παλέτας χρωμάτων που εκπροσωπεί την εικόνα RGB και την αντιστοίχιση κάθε pixel της εικόνας σε ένα δείκτη στην παλέτα. Αυτό μετριάζει την πολυπλοκότητα της εικόνας ενώ διαφυλάσσει την ποιότητα της εικόνας όσο το δυνατόν καλύτερα. Το αποτέλεσμα είναι μια εικόνα με μικρότερο μέγεθος αρχείου και χρησιμοποιεί λιγότερη μνήμη για αποθήκευση, επιτρέποντας την εποικοδομητική αποθήκευση και μεταφορά της.

4 Συμπίεση εικόνας

Τα παρόντα μέσα αποθήκευσης, παρόλο που έχουν βελτιωθεί σπουδαία από τα προγενέστερα, συνεχίζουν να ξεπερνούν προκλήσεις στην αποθήκευση του υπερβολικά μεγάλου όγκου πληροφορίας που μπορεί να εμπεριέχει μορφές όπως τα βίντεο, ο ήχος, οι κι-

νούμενες εικόνες και τα στατικά γραφικά υψηλής ανάλυσης. Λαμβάνοντας υπόψη την ποικιλία των μεγεθών που μπορεί να έχει μια εικόνα, είτε είναι μεγάλη είτε μικρή, αξίζει να διερευνήσουμε τη διαδικασία συμπίεσης της. Παρότι οι σύγχρονοι αισθητήρες καταγραφής έχουν μεγάλους χώρους αποθήκευσης, εξακολουθεί να υπάρχει ανάγκη για επίπλεον υποτίμηση της πληροφορίας που αποθηκεύεται. Αυτό επιτυγχάνεται με τη συμπίεση της εικόνας, επιτρέποντας την αποθήκευση περισσότερων εικόνων στον καθορισμένο αποθηκευτικό χώρο. Επιπλέον, η μετακίνηση εικόνων μέσω οποιουδήποτε δικτύου, όπως το τηλεπικοινωνιακό ή το διαδίκτυο, μπορεί να συναποτελέσει πρόκληση λόγω του

μεγέθους των εικόνων. Για να διασφαλιστεί η γοργή μεταφορά και η αποφυγή συμφόρησης στο δίκτυο, είναι αναγκαίο να μεταφερθεί λιγότερο δεδομένο, διαφυλάσσοντας ταυτόχρονα την ποιότητα της εικόνας χωρίς αξιόλογες αλλοιώσεις.

Ειδικά για αποθηκευτικά μέσα όπως το CD-R και το DVD-R, η αποθήκευση πολυμεσικών περιεχομένων είναι κατά κανόνα καθορισμένη εξαιτίας του περιορισμένου χώρου αποθήκευσης. Για παράδειγμα, ένα CD-R μπορεί να εμπερικλείσει περίπου 700 MB δεδομένων, ποσότητα που επαρκεί για να αποθηκεύσει περίπου 75 λεπτά ασυμπίεστου ήχου υψηλής ποιότητας CD (320 Kbps-44100 Hz), αλλά μόνο περίπου 30 δευτερόλεπτα ασυμπίεστης ψηφιακής τηλεόρασης. Κατά τη ψηφιοποίηση μιας εικόνας με ανάλυση 2000x2000 σημείων, δημιουργούνται περίπου 4 εκατομμύρια εικονοστοιχεία (pixels). Αυτό ισοδυναμεί σε περίπου 10 MB αποθηκευτικού χώρου.

4.1 Μορφές συμπίεσης

Η απάντηση στο πρόβλημα του αποθηκευτικού χώρου που καταλαμβάνει η πληροφορία πηγάζει από τη συμπίεση. Επιδίωξη της συμπίεσης είναι να περιοριστεί το μέγεθος που απαιτεί η πληροφορία, αν και με αντάλλαγμα την ποιότητα και την πιστότητα του νοήματος της. Επιπροσθέτως, η διαδικασία της συμπίεσης και αποσυμπίεσης απαιτεί υπολογιστική ισχύ και χρόνο. Αυτό δηλώνει ότι υπάρχει ένα κόστος στην επιτέλεση αυτών των διαδικασιών, το οποίο μπορεί να είναι τόσο αξιόλογο που απαιτεί ακόμα και ειδικό υλικό για να μπορέσει να συντελεστεί σε πραγματικό χρόνο. Η πληροφορία που έχει συμπίεστεί σε μια καθορισμένη μορφή δεν είναι ωφέλιμη μέχρις ότου υποστεί τη διαδικασία της αποσυμπίεσης, ώστε να ανακτήσει το αρχικό της περιεχόμενο και τη σημα-

σιολογική του σημασία. Συνήθως, η προσοχή μας επικεντρώνεται στην ταχύτητα της αποσυμπίεσης και όχι στη συμπίεση. Σε πολλές εφαρμογές, η συμπίεση γίνεται μια φορά κατά τη διάρκεια της δημιουργίας χρησιμοποιώντας ειδικό υλικό και λογισμικό όπως είναι οι αλγόριθμοι, ενώ η αποσυμπίεση εκτελείται από τους καταληκτικούς χρήστες με τη χρήση γενικών υπολογιστών και εξειδικευμένου λογισμικού. Μπορούμε να αναγνωρίσουμε τους αλγορίθμους συμπίεσης σε δύο πρωταρχικούς τύπους:

- **Αλγόριθμοι Συμπίεσης Χωρίς Απώλειες (μη απωλεστικοί)**

Το ιδιαίτερο γνώρισμα αυτής της κατηγορίας αλγορίθμων είναι ότι η διαδικασία συμπίεσης δεν εκδηλώνει καμία διαστρέβλωση στο αρχικό αποτέλεσμα. Αυτό δηλώνει ότι μετά την αποσυμπίεση, η πληροφορία ανακτά ακριβώς την αρχική της μορφή. Συνήθως, αυτοί οι αλγόριθμοι χρησιμοποιούνται σε περιπτώσεις όπου κάθε μορφή απώλειας είναι απαγορευτική.

- **Αλγόριθμοι Συμπίεσης με Απώλειες (απωλεστικοί)**

Σε περιπτώσεις όπως ήχος, κινούμενη εικόνα, ψηφιακές φωτογραφίες κ.λπ., είναι δυνατό να επιτευχθεί βελτιωμένη συμπίεση μέσω μερικών υποχωρήσεων στην ποιότητα του συμπίεσμένου σήματος. Σε αυτές τις περιπτώσεις, είναι φανερό ότι το περιεχόμενο διαμένει ουσιαστικά αναλλοίωτο σε ό,τι αφορά την έννοια του, παρόλο που υπάρχει πτώση στην ποιότητά του. Ωστόσο, η ακολουθία bits που εκπροσωπεί το ψηφιακό σήμα προβάλλει αλλαγές. Στον τομέα του ήχου, επιστήμονες μετά από μακροχρόνιες έρευνες παρατήρησαν ότι η ακοή, όπως και η όραση, διαθέτει ορισμένες αδυναμίες. Αυτές οι αδυναμίες αξιοποιούνται από διάφορες μεθόδους απωλεστικής συμπίεσης προκειμένου να ευδοκμήσουν τους στόχους τους. Σε ό,τι αφορά τον ήχο, ο αλγόριθμος του mp3, το οποίο είναι το πιο δημοφιλές πρότυπο για τη διάδοση μουσικής μέσω Διαδικτύου, καταχράται την ανικανότητα της ανθρώπινης ακοής και τον τρόπο που ο εγκέφαλος εκλαμβάνει τους διάφορους ήχους. Κατά την επεξεργασία των ήχων, ο ανθρώπινος εγκέφαλος αδυνατεί να διακρίνει τη διαφορά μεταξύ δύο ηχητικών σημάτων που έχουν παραπλήσιο συχνοτικό περιεχόμενο, ενώ αντιλαμβάνεται μόνο τον ήχο με τη μεγαλύτερη ένταση.

4.2 Κωδικοποίηση εντροπίας

Στις περισσότερες εικόνες τα κοντινά τους εικονοστοιχεία είναι συσχετισμένα και έτσι περιλαμβάνουν περιττές πληροφορίες. Ο σκοπός μας είναι να εντοπίσουμε μια λιγότερο

συσχετισμένη αναπαράσταση της εικόνας, να πραγματοποιήσουμε μείωση της περιττότητας και μείωση της άσχετης πληροφορίας. Η ελάττωση της περιττότητας υπεξαιρεί τη διπλασιαστική πληροφορία από την πηγή του σήματος. Η μείωση της άσχετης πληροφορίας αποκλείει μέρη του σήματος που δεν θα προσέξει το Ανθρώπινο Οπτικό Σύστημα (HVS). Η κωδικοποίηση εντροπίας περισσότερο συμπιέζει τις κβαντοποιημένες τιμές με απώλειες, δίνοντας καλύτερη συμπίεση ολοκληρωτικά. Χρησιμοποιεί ένα μοντέλο για να ορίσει ακριβώς τις πιθανότητες για κάθε κβαντοποιημένη τιμή και παράγει έναν κατάλληλο κώδικα με βάση αυτές τις πιθανότητες, έτσι ώστε η παραγόμενη έξοδος του κώδικα θα είναι μικρότερη από την είσοδο. Οι τεχνικές κωδικοποίησης εντροπίας δεν παραλαμβάνουν υπόψη τους τη φύση των πληροφοριών που πρόκειται να συμπιεστούν. Έτσι, αυτές οι μέθοδοι αντιμετωπίζουν τα δεδομένα ως ένα απλό σύνολο δυαδικών ψηφίων.

4.3 Οι πολύτιμες πτυχές της κωδικοποίησης εντροπίας

Κάποιος θα μπορούσε να πιστέψει ότι η στάθμιση αρκεί για συμπίεση. Είναι αλήθεια ότι η στάθμιση συμπιέζει τα δεδομένα δηλωτικά. Μετά το βήμα της στάθμισης, πολλές από τις τιμές των εικονοστοιχείων απομακρύνονται ή αντικαθίστανται με άλλες αρμόζουσες τιμές. Πάντως, αυτές οι τιμές των εικονοστοιχείων συνεχίζουν να αναπαρίστανται είτε με 8 είτε με 16 bits. Έτσι, αποβλέπουμε στην ελαχιστοποίηση του αριθμού των bits που χρησιμοποιούνται μέσω της κωδικοποίησης εντροπίας. Ο μετασχηματισμός Karhunen-Loève ή οι Πρωτογενείς Συνιστώσες Κύριας (PCAs) συγκατατίθενται στην αναπαράσταση κάθε pixel στην ψηφιακή εικόνα με την ελάχιστη δυνατή αναπαράσταση bits, σύμφωνα με την πιθανότητά τους, και κατ' αυτόν τον τρόπο παρέχουν μια βελτιστοποιημένη αναπαράσταση χωρίς απώλειες χρησιμοποιώντας το ελάχιστο ποσό μνήμης.

4.4 Κωδικοποίηση πηγής

Οι μετασχηματισμοί που ανέχεται το αρχικό σήμα σε αυτήν την τεχνική κωδικοποίησης υπόκεινται στενά από τον τύπο του σήματος. Με άλλα λόγια, ο τρόπος με τον οποίο γίνεται η κωδικοποίηση εξαρτάται από το είδος της πληροφορίας που πρέπει να μεταδοθεί ή να αποθηκευτεί. Για παράδειγμα, ο λόγος εμπερικλείει συχνά διαστήματα σιωπής, τα οποία μπορούν να περιγραφούν με πιο ενεργητικό και εποικοδομητικό τρόπο. Αυτό σημαίνει ότι οι μετασχηματισμοί του σήματος εκμεταλλεύονται τα ιδιαίτερα ση-

μασιολογικά του χαρακτηριστικά για να μεταφέρουν την πληροφορία με πιο αποδοτικό τρόπο. Συνολικά, αυτές οι μέθοδοι μπορούν να προσφέρουν υψηλότερα ποσοστά συμπίεσης σε σύγκριση με την κωδικοποίηση εντροπίας. Ωστόσο, υπολείπονται στην επιμετάλλωση, καθώς το επίπεδο συμπίεσης που ευδοκιμεί είναι μεταβλητό ανάλογα με το περιεχόμενο που υποβάλλεται σε συμπίεση. Η κωδικοποίηση πηγής μπορεί να διαλέξει μεταξύ μεθόδων με απώλειες ή χωρίς απώλειες. Συνήθως, οι τεχνικές κωδικοποίησης πηγής διακρίνονται σε τρεις κύριες κατηγορίες. Αυτές οι κατηγορίες είναι οι εξής:

- **Κωδικοποίηση μετασχηματισμού**
- **Διαφορική ή προβλεπτική κωδικοποίηση**
- **Διανυσματική κβαντοποίηση**

4.4.1 Κωδικοποίηση μετασχηματισμού

Η μεθοδολογία της κωδικοποίησης μετασχηματισμού αξιοποιείται ευρέως για τη συμπίεση εικόνων. Κατά τη διαδικασία του μετασχηματισμού στην κωδικοποίηση, το σήμα υποβάλλεται σε μια μαθηματική τροποποίηση από τον αρχικό χρόνο ή χώρο σε ένα αφηρημένο πεδίο που είναι περισσότερο κατάλληλο για τη συμπίεση του. Αυτή η διαδικασία είναι αναστρέψιμη, που σημαίνει ότι υπάρχει η δυνατότητα εφαρμογής του αντίστροφου μετασχηματισμού, ο οποίος επανάγει το σήμα στην αρχική του μορφή. Ένα παράδειγμα παραπλήσιου μετασχηματισμού είναι ο μετασχηματισμός Fourier, ο οποίος μετατρέπει μια συνάρτηση $f(t)$ από το πεδίο του χρόνου σε μια νέα συνάρτηση $g(l)$ στο πεδίο των συχνοτήτων.

4.4.2 Διαφορική ή προβλεπτική κωδικοποίηση

Η διαφορική ή προβλεπτική κωδικοποίηση διασπάται σε δυο ερμηνείες.

- **Στην Απλή διαφορική παλμοκωδική διαμόρφωση (Differential Pulse Code Modulation – DPCM).** Αυτή η μορφή διαφορικής κωδικοποίησης είναι η πιο απλή ανάμεσα στις τρεις που υπάρχουν. Κάθε δείγμα προβλέπεται μόνο από την τιμή του προηγούμενου δείγματος. Ο μηχανισμός που χρησιμοποιείται για να καθορίσει τις τιμές διαμένει αμετάβλητος καθ' όλη τη διάρκεια της διαδικασίας κωδικοποίησης.
- **Στη Διαμόρφωση Δέλτα (Delta Modulation – PCM).** Η διαφορική κβαντοποίηση είναι μια διαδικασία τροποποίησης αναλογικού σήματος σε ψηφιακό και αντίστροφα, ευρέως χρησιμοποιούμενη στη μετάδοση φωνητικών πληροφοριών, όπου η ποιότητα δεν συγκροτεί κύριο παράγοντα. Στη διαδικασία της διαφορικής κβαντοποίησης, η απόκλιση

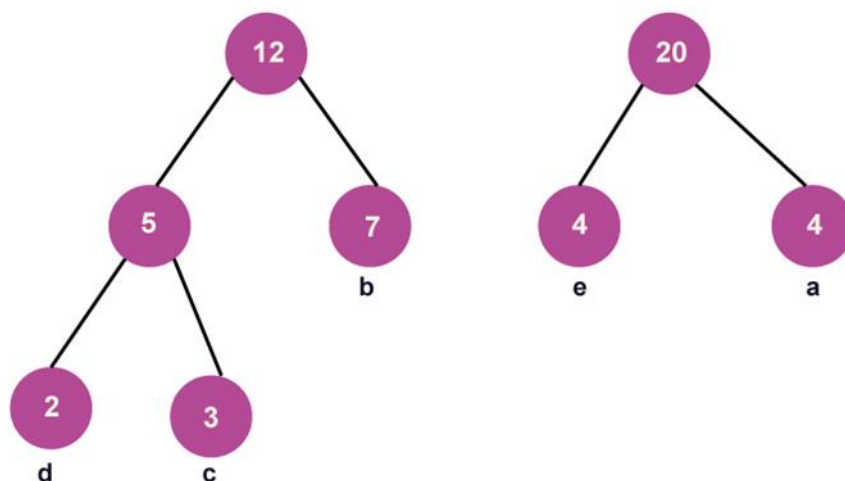
μεταξύ της προβλεπόμενης τιμής και της πραγματικής τιμής του δείγματος αναπαρίσταται με ένα μόνο bit. Αυτό εξυπνοποιεί ότι κάθε δείγμα μπορεί να είναι είτε λίγο μικρότερο είτε λίγο μεγαλύτερο από το προηγούμενό του, κατά ένα σταθερό κβάντο ή διάστημα δειγματοληψίας.

4.4.3 Διανυσματική κβαντοποίηση

Η διανυσματική κβαντοποίηση θεωρείται μια εκδοχή της μεθόδου αντικατάστασης προτύπων που πρόβαλε από τη διαδικασία κωδικοποίησης εντροπίας. Η ιδιαιτερότητα της εν λόγω τεχνικής είναι η ικανότητα απώλειας πληροφορίας. Στην ουσία, η κύρια αρχή της διανυσματικής κβαντοποίησης είναι η διαίρεση των δεδομένων σε διανύσματα.

4.5 Κωδικοποίηση Huffman

Η κωδικοποίηση Huffman εκπροσωπεί μια γενικευμένη προσέγγιση της στατιστικής κωδικοποίησης, όπου προσδιορίζεται η συχνότητα παρουσίασης κάθε χαρακτήρα σε μια συγκεκριμένη αλληλουχία στοιχείων. Με βάση τη συχνότητα εμφάνισης των χαρακτήρων, ο αλγόριθμος του Huffman υπολογίζει το ελάχιστο μήκος κωδικού που απαιτείται για κάθε χαρακτήρα και εφαρμόζει τη βέλτιστη ανάθεση κωδικών. Τα επακόλουθα αυτής της διαδικασίας αποθηκεύονται στο λεξικό. Η τεχνική του Huffman χρησιμοποιείται για τη συμπίεση τόσο κινούμενων όσο και στατικών εικόνων, καθώς και για αρχεία ήχου. Ανάλογα με την εφαρμογή, μπορεί να δημιουργηθεί ένα νέο λεξικό προκειμένου να επιτευχθεί η βέλτιστη συμπίεση. Στην περίπτωση της κινούμενης εικόνας, η θεμελίωση ενός λεξικού μπορεί να επαναλαμβάνεται για κάθε καρέ ή σειρά καρέ. Σε κάθε περίπτωση, η διαδικασία συμπίεσης πρέπει να διαφυλάσσει το λεξικό ώστε να επιτρέπεται η επαναφορά της αρχικής κατάστασης κατά την αποσυμπίεση.



5 Επεξεργασία εικόνας

5.1 Ανάλυση εικόνας

Η ερμηνεία μιας εικόνας αναφέρεται στον αριθμό των εικονοστοιχείων που συγκροτούν την εικόνα, υπολογισμένος σε μονάδες μήκους. Η ανάλυση μιας εικόνας αξιοποιείται για την κβαντικοποίηση των χαρακτηριστικών που διασκορπίζονται σε αυτήν, προκειμένου να προσδιοριστεί η ομοιογένεια ή ανομοιογένεια της διασποράς και να υπολογιστεί σε pixels per inch (ppi). Λεπτομερώς, αναφερόμαστε στην σημασία της χωρικής συχνότητας δειγματοληψίας, η οποία αφορά τον τρόπο με τον οποίο οι πληροφορίες σε μια εικόνα δείγματος αναπαρίστανται στον χώρο. Η συχνότητα δειγματοληψίας άπτεται στον αριθμό δειγμάτων που λαμβάνονται ανά μονάδα χρόνου από ένα αναλογικό σήμα, ενώ η ανάλυση αναφέρεται στον αριθμό δειγμάτων που λαμβάνονται ανά μονάδα μήκους από μια αναλογική εικόνα. Η απλή υψηλή ανάλυση δεν διαβεβαιώνει απαραίτητα μια ποιοτικότερη απεικόνιση ενός εικονογραφήματος. Επιπλέον, στην ανάλυση εικόνας, οι ακμές προβάλλουν κρίσιμο ρόλο καθώς συμπεριλαμβάνουν σημαντικές πληροφορίες σχετικά με τα όρια και τη δομή των αντικειμένων που απεικονίζονται. Ακολουθούν ορισμένα βήματα για την ανάλυση μιας εικόνας, τα οποία περιγράφονται παρακάτω.

1. Ανάλυση grayscale εικόνας

Κάνετε χρήση του ακόλουθο κώδικα:

```
I=imread('frog.tif');  
imshow(I)  
size(I)  
I  
max_val=max(I(:))  
min_val=min(I(:))  
mean_val=mean(I(:))
```

```

>> max_val=max(I(:))

max_val =

    254

>> min_val=min(I(:))

min_val =

     0

>> mean_val=mean(I(:))

mean_val =

    122.0409

```

Εικόνα 12: Αποτέλεσμα άσκησης

5.1.1 Κατωφλίωση ή Thresholding

Η διαδικασία της κατωφλίωσης τροποποιεί μια εικόνα από ασπρόμαυρη σε δυαδική μορφή, εγκρίνοντας την απλούστευση της εικόνας και την επεξεργασία της σε μετέπειτα στάδια. Η μετατροπή μιας εικόνας σε δυαδική γίνεται με την ορισμοθέτηση ενός επιπέδου κατωφλίου (threshold level), και στη συνέχεια, όλα τα εικονοστοιχεία που έχουν επίπεδο γκρι κάτω από αυτό το επίπεδο κατωφλίου αλλάζουν σε μαύρο και λαμβάνει την τιμή 0, ενώ τα υπόλοιπα αλλάζουν σε λευκό και παίρνει την τιμή 255. Το επίπεδο κατωφλίου μπορεί να προσδιοριστεί είτε μέσω μαθηματικών μεθοδολογιών είτε μελετώντας την εικόνα και το ιστόγραμμα της, όπου το ιστόγραμμα εκδηλώνει τη συχνότητα εμφάνισης κάθε επιπέδου γκρι στην εικόνα. Κατά την επιλογή του επιπέδου κατωφλίου, υπάρχει η πιθανότητα να ανακύψουν δύο μορφές σφαλμάτων, τα οποία είναι:

Τα pixel που επιθυμούμε να μην εντοπίζονται στην νέα μας εικόνα.

Στην νέα μας εικόνα ευρίσκονται pixel ενώ προβλεπόμενα δεν θα έπρεπε.

Προτού ληφθεί απόφαση για το επίπεδο κατωφλίου, είναι αξιόλογο να καθοριστεί ποιο είδος σφάλματος είναι πιο δεκτό ή προτιμητέο στο καθορισμένο πλαίσιο εφαρμογής. Συνήθως, προτιμάται η εμφάνιση του λάθους τύπου (β), δηλαδή να υπάρχουν pixel που κανονικά δεν θα έπρεπε να υπάρχουν και στη συνέχεια εφαρμόζεται η αυτόματη κατωφλίωση, όπου το επίπεδο κατωφλίου προσδιορίζεται με βάση μια εικόνα που αποτελείται από δύο διακριτά τμήματα. Τα δυο αυτά μέρη είναι τα εξής: το background και το αντικείμενο.

Τα επίπεδα γκρι του φόντου (background) διαθέτουν ένα διαφορετικό γκρι level σε σχέση με το αντικείμενο, και αυτά μπορούν να διακριθούν ως δύο κορυφές στο ιστόγραμμα. Το μέσο των δύο κορυφών μπορεί να οριστεί ως το επίπεδο κατωφλίου.

Τα επίπεδα γκρι του αντικείμενου συνοδεύουν μια στατιστική κατανομή.

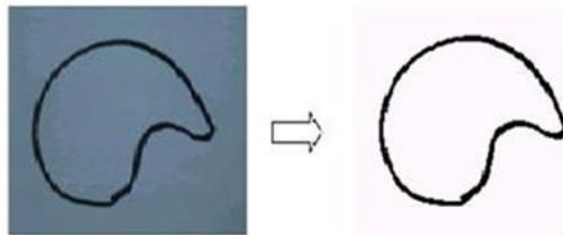
Οι εξισώσεις – μαθηματικές πράξεις πραγμάτωσης τους είναι:

$$\text{Για το γκρι επίπεδο: } T = \max \{ t(g) P - t(g) * (m(g) - m(G - 1)) / 2 \} - 1$$

$$\text{Για τον αριθμό των pixels με επίπεδο γκρι: } t(g) = \sum_{i=1}^g f(i)$$

$$\text{Για το μέσο γκρι επίπεδο: } m(g) = \frac{\sum_{i=0}^g f(i)}{t(g)}$$

Παρακάτω παρουσιάζεται η εικόνα που συνεπάγεται μετά τη διαδικασία του thresholding, τροποποιώντας μια ασπρόμαυρη εικόνα σε δυαδική, όπου καθορισμένα pixels είναι μαύρα και άλλα λευκά, ανάλογα με το κατώφλι που έχει οριστεί.



Εικόνα 13:Αποτέλεσμα Thresholding. [8]

5.1.2 Αφαίρεση περιγράμματος

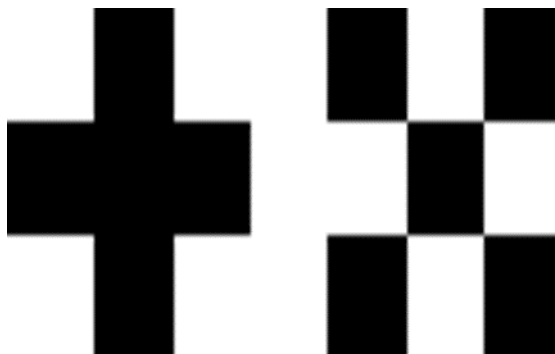
Σε αυτό το στάδιο, τα pixel που υπάγονται στο πλαίσιο της εικόνας τροποποιούνται ώστε να έχουν το ταυτόσημο χρώμα με το φόντο (background), συνεπώς εξελίσσονται σε λευκά. Αυτό το βήμα επιτυγχάνει την αποφυγή δυσκολιών όταν εφαρμόζονται αργότερα μάσκες, καθώς η πληροφορία στα περιγράμματα της εικόνας δεν χάνεται σημαντικά.

5.1.3 Διαμερισμός εικόνας

Σε αυτό το σημείο, συμβαίνει η διαίρεση της δυαδικής εικόνας σε διακριτά αντικείμενα, διαλέγοντας τα αντικείμενα με αλληιώτικα μεγέθη. Στη συνέχεια, επισημαίνουμε το μεγαλύτερο συνδετικό τμήμα στην εικόνα, που εικάζουμε ότι αντιπροσωπεύει το μονοπάτι, και τροποποιούμε το χρώμα των pixels που δεν ανήκουν σε αυτό το τμήμα, ώστε να αρμόζει με το χρώμα που έχει το background. Τα αντικείμενα που έχουν την ίδια ετικέτα

θεωρούνται συνδεδεμένα. Υπάρχουν δύο διαφορετικές προσεγγίσεις για τον ορισμό της σύνδεσης, οι οποίοι είναι:

- **Σύνδεση με τέσσερις γειτονικές θέσεις:** είναι μια σημασία που χρησιμοποιείται στην επεξεργασία εικόνας και άπτεται στον τρόπο με τον οποίο τα pixels σε μια δισδιάστατη εικόνα είναι συνδεδεμένα με τους γειτονικούς τους. Συγκεκριμένα, στην four connectedness, ένα pixel φαίνεται συνδεδεμένο με τα γειτονικά του pixel πάνω, κάτω, αριστερά και δεξιά του, αλλά όχι με τις γωνίες. Αυτό σημαίνει ότι ένα pixel έχει σύνδεση μόνο με τα pixel που εντοπίζονται άμεσα δίπλα του στις κατευθύνσεις πάνω, κάτω, αριστερά και δεξιά.
- **Σύνδεση με οκτώ γειτονικές θέσεις:** Αναλυτικά, στην eight connectedness, ένα pixel θεωρείται συνδεδεμένο με όλα τα pixel που εντοπίζονται απευθείας δίπλα του, συμπεριλαμβανομένων των pixel που βρίσκονται στις γωνίες του. Αυτό δηλώνει ότι ένα pixel έχει σύνδεση με όλα τα pixel που βρίσκονται απευθείας δίπλα του, σε κάθε δυνατή κατεύθυνση: πάνω, κάτω, αριστερά, δεξιά, καθώς και στις γωνίες.



Εικόνα 14:Δύο τρόποι σύνδεσης. [8]

5.1.4 Σκελετοποίηση

Σε αυτήν τη διαδικασία, επιχειρούμε να επισημάνουμε τον μέσο άξονα των γραμμών ή των αντικειμένων στην εικόνα, μετατοπίζοντας σταδιακά το εξωτερικό τους στρώμα. Αυτό επιτυγχάνεται με την εφαρμογή τακτικής και επαναλαμβανόμενης διαδικασίας εξάλειψης του εξωτερικού στρώματος, προκειμένου να εντοπιστεί ο πυρήνας ή ο κεντρικός άξονας των αντικειμένων. Όταν μια γραμμή είναι αρκετά παχιά σε μια εικόνα, συρρικνώνεται σε μια γραμμή πάχους ενός μόνο pixel. Ο τρόπος με τον οποίο τα pixels σχετίζονται για να διαπλάσουν το σκελετό υπόκειται από τον τύπο σύνδεσης που επιλέγεται για τη σκελετοποίηση της γραμμής. Κατά τη σκελετοποίηση μπορεί να ανακύψουν προβλήματα λόγω του θορύβου που είναι πιθανόν να εμφανίζεται στην εικόνα. Αυτό σημαί-

νει ότι ο θόρυβος μπορεί να επιδράσει στη σωστή ανίχνευση των γραμμών ή των αντικειμένων, καθιστώντας τη σκελετοποίηση πιο δύσκολη ή ακόμα και ανεφάρμοστη σε ορισμένες περιπτώσεις.

5.1.5 Αραίωση

Κατά τη διαδικασία αυτή, εφαρμόζονται μάσκες σε όλα τα pixel της επιφάνειας ενός αντικειμένου, με βλέψη την εξάλειψη του εξωτερικού layer. Εάν η μάσκα ταιριάζει με ένα pixel, τότε αυτό το pixel διαγράφεται και η διαδικασία επαναλαμβάνεται για όλα τα pixel μέχρι να εξεταστούν όλα τα επιμέρους pixel του αντικειμένου.

5.1.6 Κλάδεμα

Η διαδικασία αυτή είναι επαναληπτική και εφαρμόζεται για την απόσπαση των κλάδων ή των ακριανών σημείων ενός αντικειμένου, όπως ο σκελετός σε μια εικόνα. Κατά την επανάληψη, αφαιρείται το ακριανό pixel από κάθε κλάδο. Η διαδικασία παύει όταν δεν είναι δυνατή η επιπλέον μεταρρύθμιση της εικόνας.

5.2 Ανάγνωση εικόνας

Στο MATLAB, το περιβάλλον εργασίας παρέχει τη δυνατότητα επεξεργασίας εικόνων, και μία από τις βασικές λειτουργίες του είναι η ανάγνωση εικόνων από αρχεία αποθηκευμένα στον υπολογιστή. Η ανάγνωση αυτή πραγματοποιείται μέσω του χώρου πληκτρολόγησης εντολών, ο οποίος επιτρέπει στον χρήστη να εκτελεί εντολές σε πραγματικό χρόνο. Για την εκτέλεση της συγκεκριμένης λειτουργίας, το MATLAB περιλαμβάνει τη συνάρτηση `imread`. Η συνάρτηση `imread` είναι ένα εργαλείο που έχει σχεδιαστεί για να διευκολύνει την ανάγνωση αρχείων εικόνας από διάφορες γνωστές μορφές (formats), όπως BMP, JPEG, PCX, PNG και TIFF. Αυτές οι μορφές είναι ευρέως διαδεδομένες και χρησιμοποιούνται σε πολλές εφαρμογές, από τη φωτογραφία και τα γραφικά έως την επιστημονική ανάλυση εικόνας.

Κατά την ανάγνωση ενός αρχείου εικόνας, τα δεδομένα της εικόνας εισάγονται στη μνήμη του MATLAB σε συγκεκριμένες μορφές δεδομένων. Για τις περισσότερες μορφές εικόνας, όπως BMP, JPEG και PCX, η ανάγνωση γίνεται στη μορφή `uint8`, η οποία αντιπροσωπεύει ακέραιες τιμές 8-bit, δηλαδή τιμές από 0 έως 255. Αυτή η μορφή είναι ιδανική για την αναπαράσταση δεδομένων εικόνας που χρησιμοποιούν το φάσμα του

γκρι ή των χρωμάτων RGB. Ωστόσο, για συγκεκριμένες μορφές εικόνας, όπως PNG και TIFF, το MATLAB υποστηρίζει επίσης την ανάγνωση σε μορφή uint16. Η μορφή αυτή επιτρέπει μεγαλύτερο βάθος χρώματος, έως και 16 bit ανά κανάλι, κάτι που είναι χρήσιμο για επιστημονικές ή επαγγελματικές εφαρμογές που απαιτούν υψηλή ακρίβεια. Για να γίνει πιο κατανοητή η χρήση της συνάρτησης `imread`, μπορεί να δοθεί ένα παράδειγμα που δείχνει πώς πραγματοποιείται η ανάγνωση μιας εικόνας στο MATLAB. Η λειτουργία αυτή είναι απλή, αλλά εξαιρετικά ισχυρή, καθώς επιτρέπει στον χρήστη να φορτώσει την εικόνα στη μνήμη και να την επεξεργαστεί περαιτέρω με τις πλούσιες δυνατότητες επεξεργασίας που προσφέρει το MATLAB.

`x=imread('bird.tif');`

Η διαδικασία ανάγνωσης των δεδομένων φωτεινότητας μιας εικόνας στο MATLAB γίνεται με τρόπο που τα δεδομένα αυτά μεταφέρονται σε έναν πίνακα, ο οποίος ονομάζεται, για παράδειγμα, `x`. Ο πίνακας αυτός περιέχει τις τιμές που αντιστοιχούν στα pixel της εικόνας σε μορφή uint8. Η μορφή uint8 είναι ένα σύστημα αναπαράστασης που αποθηκεύει τιμές ακέραιων αριθμών από 0 έως 255, κάτι που είναι ιδανικό για την αποτύπωση επιπέδων φωτεινότητας ή χρωμάτων στην εικόνα. Η διαδικασία αυτή προϋποθέτει ότι η συνάρτηση που χρησιμοποιείται δέχεται δύο εισόδους. Το πρώτο όρισμα είναι το όνομα του αρχείου εικόνας, ενώ το δεύτερο είναι ο τύπος του αρχείου. Για παράδειγμα, ένα αρχείο μπορεί να ονομάζεται `εικόνα.jpg`, όπου το "εικόνα" είναι το όνομα του αρχείου και "jpg" ο τύπος του. Αυτά τα δύο δεδομένα είναι απαραίτητα για να κατανοήσει το MATLAB ποιο αρχείο να αναζητήσει και πώς να το διαχειριστεί.

Ωστόσο, η διαδικασία αυτή απαιτεί το αρχείο να βρίσκεται είτε στον τρέχοντα κατάλογο είτε σε έναν από τους καταλόγους που περιλαμβάνονται στη διαδρομή αναζήτησης του MATLAB. Αν το αρχείο δεν υπάρχει σε αυτούς τους καταλόγους, τότε πρέπει να καθορίσουμε τη διαδρομή του πλήρως. Η πλήρης διαδρομή είναι η ακριβής θέση του αρχείου μέσα στο σύστημα αρχείων του υπολογιστή μας. Για παράδειγμα, μια πλήρης διαδρομή μπορεί να είναι:

`C:\Users\Username\Pictures\εικόνα.jpg`.

Σε περίπτωση που το αρχείο δεν υπάρχει ούτε στον τρέχοντα κατάλογο ούτε στη διαδρομή αναζήτησης του MATLAB, ο χρήστης πρέπει να προσθέσει χειροκίνητα τη διαδρομή αυτή. Αυτό γίνεται με τη χρήση εντολών όπως η `addpath`, που επιτρέπει στον χρήστη να ενσωματώσει νέες τοποθεσίες στον χώρο αναζήτησης του MATLAB. Η διαδικασία ανάγνωσης ενός δυαδικού αρχείου στο MATLAB, όπως τα αρχεία με κατάληξη `.bin`,

απαιτεί τη χρήση κατάλληλων συναρτήσεων που είναι σχεδιασμένες για τη διαχείριση αυτού του τύπου δεδομένων. Τα δυαδικά αρχεία περιέχουν δεδομένα σε μορφή που δεν είναι άμεσα αναγνώσιμη από τον χρήστη, αλλά μπορούν να περιλαμβάνουν πληροφορίες όπως εικόνες, ήχους ή άλλες μη κειμενικές μορφές δεδομένων. Η ανάγνωση τέτοιων αρχείων στο MATLAB πραγματοποιείται μέσω ειδικών λειτουργιών που επιτρέπουν:

1. **Το άνοιγμα του αρχείου:** Το πρώτο βήμα είναι να ανοίξουμε το αρχείο σε κατάλληλη κατάσταση.
2. **Την ανάγνωση των δεδομένων:** Αφού το αρχείο ανοιχθεί, χρησιμοποιούνται συναρτήσεις που εξάγουν τα δεδομένα από το δυαδικό αρχείο και τα αποθηκεύουν σε μια κατάλληλη μορφή στο MATLAB, όπως πίνακες.

filename='bird.bin';

Η διαδικασία ανάγνωσης μιας εικόνας που έχει αποθηκευτεί σε ένα αρχείο MATLAB τύπου .mat πραγματοποιείται χρησιμοποιώντας τη συνάρτηση load. Τα αρχεία .mat είναι ειδικά αρχεία που χρησιμοποιεί το MATLAB για την αποθήκευση μεταβλητών και δεδομένων, όπως πίνακες, διανύσματα ή ακόμα και εικόνες. Αυτά τα αρχεία δημιουργούνται συνήθως με τη συνάρτηση save του MATLAB και επιτρέπουν την εύκολη αποθήκευση και φόρτωση δεδομένων χωρίς την ανάγκη επιπλέον μετατροπών. Όταν μια εικόνα αποθηκεύεται σε ένα αρχείο .mat, παραμένει στην ίδια μορφή που είχε στο MATLAB κατά τη στιγμή της αποθήκευσης (π.χ., ως πίνακας uint8, uint16, ή double). Για την ανάγνωσή της, αρκεί η χρήση της συνάρτησης load, η οποία φορτώνει όλες τις μεταβλητές που περιέχονται στο αρχείο στη μνήμη του MATLAB.

load bird;

5.3 Εγγραφή εικόνας

Στο MATLAB, εκτός από την ανάγνωση εικόνων, παρέχεται και η δυνατότητα εγγραφής εικόνων σε αρχεία χρησιμοποιώντας τη συνάρτηση imwrite. Αυτή η συνάρτηση επιτρέπει τη μετατροπή ενός πίνακα δεδομένων, ο οποίος αντιπροσωπεύει μια εικόνα, σε αρχείο εικόνας με διάφορες υποστηριζόμενες μορφές. Η δυνατότητα αυτή είναι χρήσιμη όταν θέλουμε να αποθηκεύσουμε εικόνες που έχουν δημιουργηθεί ή τροποποιηθεί στο MATLAB, για να τις χρησιμοποιήσουμε αργότερα σε άλλες εφαρμογές ή συστήματα. Η βασική χρήση της συνάρτησης imwrite περιλαμβάνει δύο κύρια ορίσματα:

- **Ο πίνακας της εικόνας:** Πρόκειται για τα δεδομένα που περιγράφουν την εικόνα, τα οποία έχουν τη μορφή πίνακα, όπως uint8, uint16, ή double, ανάλογα με τη μορφή και την ακρίβεια των δεδομένων.
- **Το όνομα και ο τύπος του αρχείου:** Καθορίζεται το όνομα του αρχείου εξόδου μαζί με την κατάληξή του η οποία προσδιορίζει τη μορφή της εικόνας που θα αποθηκευτεί.

imwrite(I,'bird.tif');

Η συνάρτηση imwrite του MATLAB παρέχει τη δυνατότητα να προσαρμόζουμε την εγγραφή εικόνας μέσω πρόσθετων παραμέτρων, οι οποίες εξαρτώνται από τον τύπο του αρχείου εικόνας που επιλέγουμε να αποθηκεύσουμε. Ορισμένες μορφές αρχείων εικόνας, όπως τα αρχεία JPEG, επιτρέπουν την προσαρμογή παραμέτρων που αφορούν την ποιότητα της εικόνας ή άλλες ιδιότητες, προκειμένου να ελεγχθεί ο τρόπος με τον οποίο θα αποθηκευτεί η εικόνα.

Για παράδειγμα, όταν αποθηκεύουμε την εικόνα σε μορφή JPEG, μπορούμε να προσδιορίσουμε παραμέτρους όπως η ποιότητα της εικόνας (που επηρεάζει το μέγεθος του αρχείου και την καθαρότητα της εικόνας). Ανάλογες παράμετροι υπάρχουν και για άλλους τύπους αρχείων, όπως τα PNG, TIFF, κ.λπ.

imwrite (I,'filename.jpg','quality', q);

Στο MATLAB, όταν δημιουργούμε εικόνες και τις αποθηκεύουμε σε μορφή JPEG, μπορούμε να καθορίσουμε την ποιότητά τους μέσω του συντελεστή ποιότητας (q), ο οποίος καθορίζει τη συμπίεση της εικόνας. Ο συντελεστής ποιότητας q κυμαίνεται από 0 έως 100 και επηρεάζει το μέγεθος του αρχείου και την ποιότητα της εικόνας. Όσο πιο κοντά στο 0 είναι ο συντελεστής ποιότητας, τόσο μεγαλύτερη είναι η συμπίεση και χειρότερη η ποιότητα. Αντίθετα, όσο πιο κοντά στο 100 είναι, τόσο καλύτερη είναι η ποιότητα και τόσο μικρότερη η συμπίεση. Για τις δυαδικές εικόνες (ή εικόνες 1-bit), η διαδικασία αποθήκευσης είναι διαφορετική, καθώς αυτές οι εικόνες αποθηκεύονται ως εικόνες 1-bit και το MATLAB τις διαχειρίζεται ως λογικές διατάξεις (δηλαδή πίνακες τύπου logical). Οι εικόνες αυτές χρησιμοποιούνται για αναπαράσταση εικόνας που περιέχει μόνο δύο τιμές, συνήθως μαύρο και άσπρο (0 και 1). Η μορφή TIFF υποστηρίζει τέτοιες εικόνες, και το MATLAB τις αποθηκεύει και τις διαχειρίζεται με τον κατάλληλο τρόπο.

BW=imread('bird.jpg');

imwrite(BW,'tif');

5.4 Απεικόνιση εικόνας

Στο MATLAB, εκτός από τη δυνατότητα επεξεργασίας και αποθήκευσης εικόνων, υπάρχει και η δυνατότητα απεικόνισής τους, δηλαδή της προβολής τους στην οθόνη ή στον χώρο εργασίας του MATLAB. Για την απεικόνιση εικόνας, χρησιμοποιούμε τη συνάρτηση `imshow`. Αυτή η συνάρτηση μας επιτρέπει να εμφανίσουμε εικόνες μέσα στο MATLAB και να τις παρατηρήσουμε ή να τις επεξεργαστούμε περαιτέρω.

`imshow(filename);`

Η συνάρτηση `imshow` στο MATLAB χρησιμοποιείται για την εμφάνιση εικόνας από ένα αρχείο, αλλά με έναν ιδιαίτερο τρόπο. Όταν καλούμε την `imshow(filename)`, η συνάρτηση διαβάζει την εικόνα από το αρχείο χρησιμοποιώντας την `imread` και την εμφανίζει στην οθόνη. Το σημαντικό είναι ότι η `imshow` δεν αποθηκεύει τα δεδομένα της εικόνας στο MATLAB workspace, αλλά μόνο εμφανίζει την εικόνα για την προσωρινή προβολή. Όταν το αρχείο γραφικών περιέχει πολλαπλές εικόνες, για παράδειγμα σε μορφή TIFF που υποστηρίζει πολλαπλές σελίδες, η `imshow` εμφανίζει την πρώτη εικόνα του αρχείου, αγνοώντας τις άλλες. Αν θέλουμε να δούμε κάποια άλλη εικόνα από το ίδιο αρχείο, μπορούμε να χρησιμοποιήσουμε την `imread` για να διαβάσουμε τη συγκεκριμένη εικόνα και στη συνέχεια να την εμφανίσουμε με `imshow`. Η `imshow` προσαρμόζεται ανάλογα με τον τύπο εικόνας που διαβάζει, και σε ορισμένες περιπτώσεις μπορεί να χρειαστεί να καθορίσουμε πρόσθετες παραμέτρους για να διασφαλίσουμε ότι η εικόνα εμφανίζεται σωστά. Αυτές οι παράμετροι διαφέρουν ανάλογα με το είδος της εικόνας (ασπρόμαυρη, έγχρωμη, 1-bit, κ.λπ.). Αν θέλουμε να εμφανίσουμε μια δυαδική εικόνα (1-bit εικόνα), η συνάρτηση `imshow` μπορεί να χρησιμοποιηθεί για να εμφανίσουμε την εικόνα με δυο μόνο χρώματα (π.χ., μαύρο και άσπρο).

`imshow('binary_image.tiff');`

Για έγχρωμες εικόνες (RGB), η `imshow` εμφανίζει την εικόνα με τα κανονικά χρωματικά κανάλια:

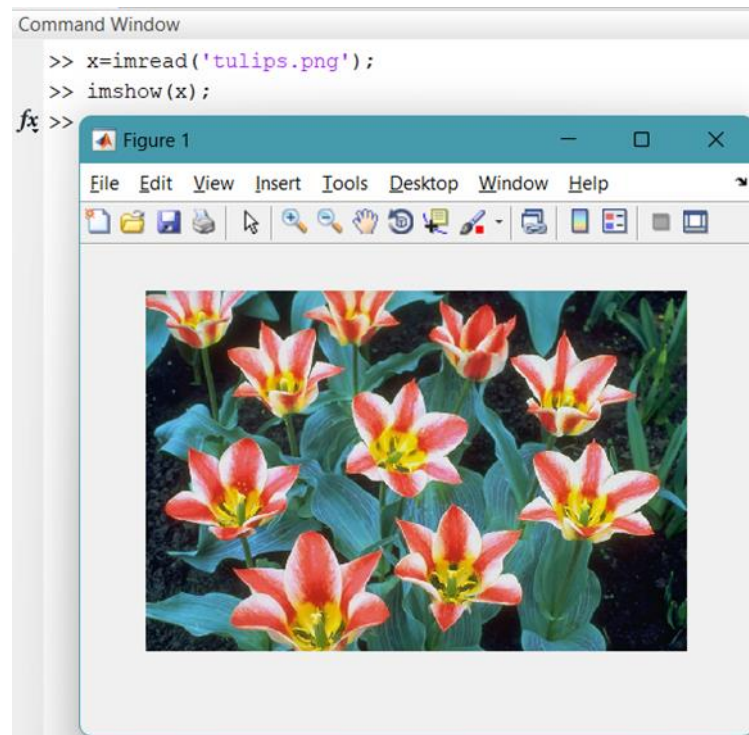
`imshow('color_image.jpg');`

Η συνάρτηση `imshow(filename)` στο MATLAB παρέχει έναν εύκολο τρόπο για την εμφάνιση εικόνας από αρχεία γραφικών χωρίς να αποθηκεύει τα δεδομένα στο workspace. Υποστηρίζει πολλούς τύπους εικόνες και προσαρμόζει την εμφάνιση ανάλογα με τον τύπο της εικόνας. Ωστόσο, εάν το αρχείο περιέχει πολλαπλές εικόνες, η `imshow` εμφανίζει μόνο την πρώτη εικόνα. Η απεικόνιση εικόνας φωτεινότητας (grayscale image) στο

MATLAB γίνεται με τη χρήση της συνάρτησης `imshow`, αλλά υπάρχουν κάποιες παραμέτρους και ιδιαίτερες περιπτώσεις που πρέπει να ληφθούν υπόψη όταν χειριζόμαστε εικόνες φωτεινότητας. Οι εικόνες φωτεινότητας είναι μονοχρωματικές εικόνες που περιέχουν μόνο πληροφορίες για την ένταση του φωτός (δηλαδή για τη φωτεινότητα κάθε pixel) και συνήθως αναπαρίστανται με κλίμακα του γκρι (από μαύρο έως λευκό).

`imshow(x);`

Παράδειγμα με εντολές που χρησιμοποιήθηκαν στο περιβάλλον του MATLAB:



Εικόνα 15:Αποτέλεσμα Matlab

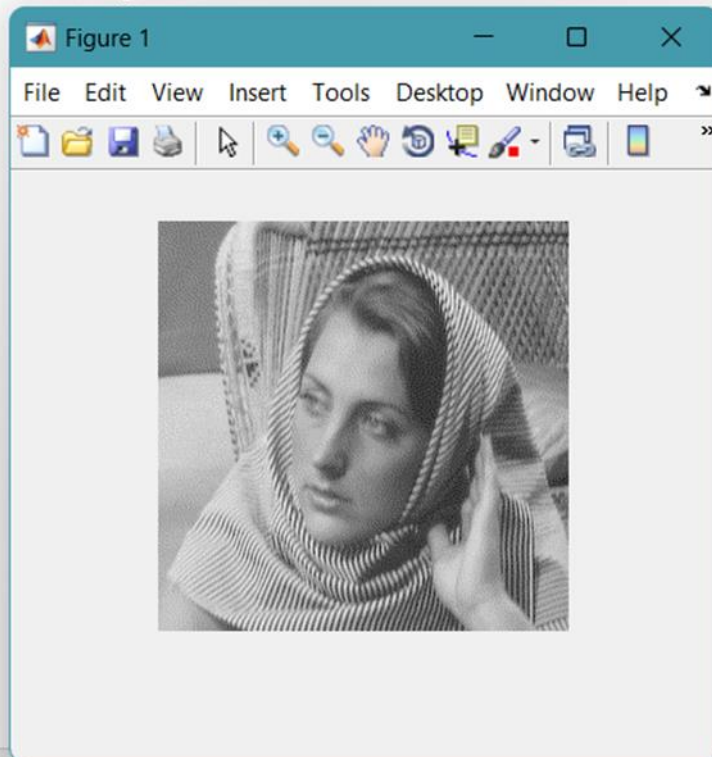
Για την παρουσίαση εικόνας χρωματικού δείκτη:

Η απεικόνιση εικόνας χρωματικού δείκτη (indexed image) στο MATLAB είναι μια διαδικασία που χρησιμοποιεί ένα σύστημα χρωμάτων (χρωματική παλέτα) για να αναπαραστήσει μια εικόνα. Αντί να αποθηκεύει τα δεδομένα της εικόνας με κάθε pixel να περιέχει μια τιμή χρώματος (όπως στις έγχρωμες εικόνες RGB), η εικόνα χρωματικού δείκτη αποθηκεύει μόνο τους δείκτες των χρωμάτων που αντλούνται από μια προκαθορισμένη παλέτα. Αυτή η μέθοδος είναι πιο αποδοτική από την αποθήκευση πλήρων χρωμάτων για κάθε pixel, ειδικά όταν οι εικόνες περιέχουν περιορισμένο αριθμό χρωμάτων. Η συνάρτηση `imshow` στο MATLAB υποστηρίζει την απεικόνιση εικόνας χρωματικού δείκτη όταν η εικόνα παρέχεται μαζί με την παλέτα χρωμάτων. Το MATLAB διαβάζει την εικόνα ως πίνακα δεικτών και χρησιμοποιεί την αντίστοιχη παλέτα για να απεικονί-

σει την εικόνα. Η απεικόνιση εικόνας χρωματικού δείκτη στο MATLAB γίνεται εύκολα με τη συνάρτηση `imshow` όταν παρέχεται μια εικόνα με τους αντίστοιχους δείκτες και μια παλέτα χρωμάτων. Η `imshow` χρησιμοποιεί την παλέτα για να εμφανίσει την εικόνα, όπου κάθε τιμή του πίνακα δείκτης αντιστοιχεί σε ένα χρώμα της παλέτας. Αυτή η μέθοδος είναι αποδοτική για εικόνες με περιορισμένο αριθμό χρωμάτων και επιτρέπει τη χρήση διαφόρων έτοιμων παλετών που παρέχει το MATLAB.

`imshow(X,map);`

```
>> load woman;  
imshow(X,map)  
>>
```

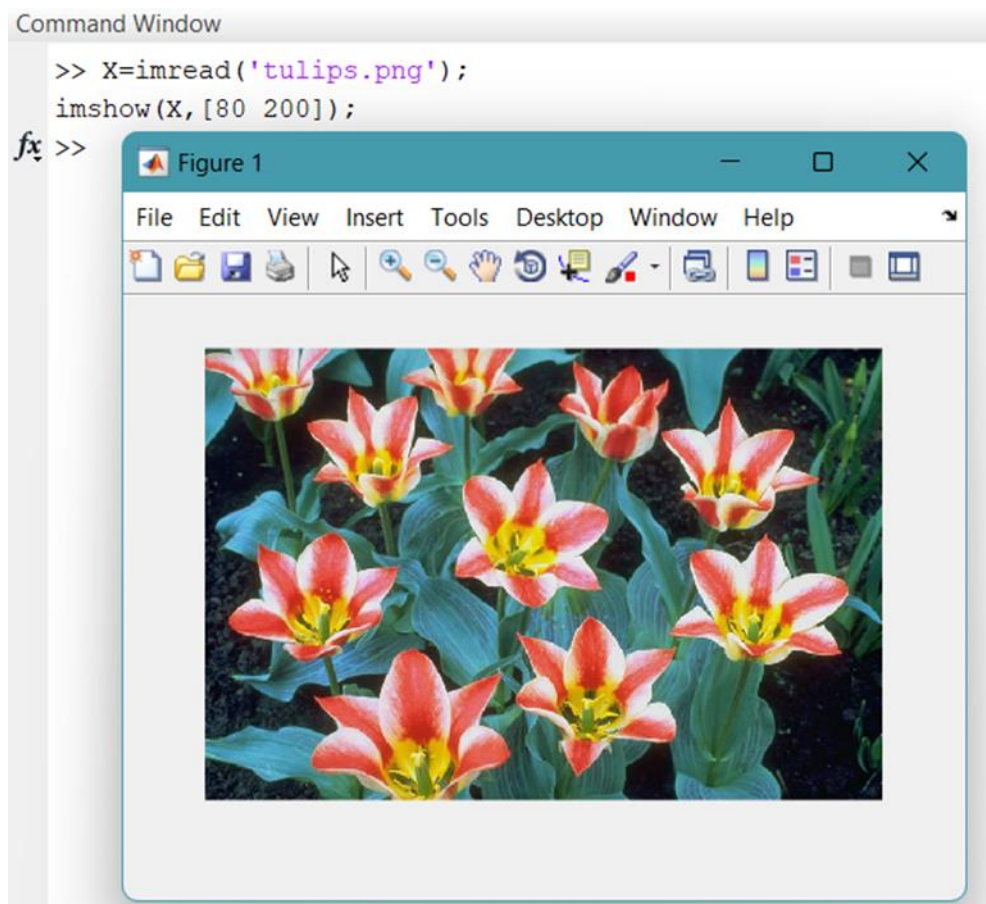


Εικόνα 16:Αποτέλεσμα Matlab

Η διαδικασία απεικόνισης τιμών που είναι μικρότερες ή ίσες με μία καθορισμένη τιμή, που ονομάζεται `low`, ως 0 και εκείνων που είναι μεγαλύτερες ή ίσες με μία άλλη καθορισμένη τιμή, που ονομάζεται `high`, ως 1, γίνεται με κλιμάκωση όλων των ενδιάμεσων τιμών να κυμαίνονται στο διάστημα $[0,1]$. Δηλαδή, οι τιμές που βρίσκονται μεταξύ του `low` και του `high` μετατρέπονται γραμμικά στο συγκεκριμένο διάστημα, ενώ οι τιμές που είναι κάτω από το `low` καταγράφονται ως 0 και εκείνες που ξεπερνούν το `high` ως 1. Αυτή η μέθοδος χρησιμοποιείται συχνά σε τεχνικές κανονικοποίησης ή μετατροπής εικόνας, όπου το ζητούμενο είναι να αναπαρασταθούν οι τιμές σε ένα ομοιογενές διάστημα, π.χ., το διάστημα $[0, 1]$ ή $[0, 255]$.

- **Ορισμός των Ορίων (low και high):** Η διαδικασία ξεκινά καθορίζοντας δύο οριακές τιμές: τη low και την high. Αυτές οι τιμές καθορίζουν το εύρος των δεδομένων που θέλουμε να κανονικοποιήσουμε. Οι τιμές που είναι μικρότερες από το low ανατίθενται στην τιμή 0, και εκείνες που είναι μεγαλύτερες ή ίσες με το high ανατίθενται στην τιμή 1.
- **Κλιμάκωση Ενδιάμεσων Τιμών:** Οι τιμές που είναι μεταξύ του low και του high κλιμακώνονται γραμμικά στο διάστημα [0, 1]. Αυτό σημαίνει ότι οι τιμές που είναι κοντά στο low θα μετατραπούν σε τιμές κοντά στο 0, ενώ εκείνες που είναι κοντά στο high θα μετατραπούν σε τιμές κοντά στο 1. Οι υπόλοιπες τιμές θα ακολουθήσουν αυτή τη γραμμική αναλογία.
- **Απλοποίηση και Χρήση:** Αυτή η τεχνική είναι χρήσιμη όταν χρειάζεται να αναπαραστήσουμε δεδομένα σε ένα συγκεκριμένο εύρος για λόγους απεικόνισης ή επεξεργασίας. Για παράδειγμα, εάν μια εικόνα έχει τιμές φωτεινότητας από 0 έως 255 και θέλουμε να τις αναπαραστήσουμε σε κλίμακα [0, 1], μπορούμε να χρησιμοποιήσουμε αυτή τη μέθοδο κλιμάκωσης.

`imshow(x,[low high]);`



Εικόνα 17: Απεικόνιση εικόνας φωτεινότητας

Στο συγκεκριμένο παράδειγμα, αναφερόμαστε σε μια διαδικασία κλιμάκωσης των τιμών ενός πίνακα X που περιέχει δεδομένα, όπως είναι οι τιμές φωτεινότητας των pixels μιας εικόνας. Ο πίνακας αυτός έχει ελάχιστη τιμή 63 και μέγιστη τιμή 241, και σκοπός μας είναι να επανακαθορίσουμε τις τιμές των pixels σε ένα νέο εύρος τιμών, για λόγους καλύτερης απεικόνισης ή ανάλυσης. Στο παράδειγμα αυτό, οι ελάχιστες και μέγιστες τιμές του πίνακα X είναι 63 και 241, αντίστοιχα. Αυτές οι τιμές καθορίζουν το αρχικό εύρος των δεδομένων. Οι τιμές του πίνακα πρέπει να μετατραπούν ώστε να απεικονίζονται με βάση καθορισμένα όρια. Ο πίνακας X θα κλιμακωθεί ώστε να έχει ως όρια τις τιμές 80 και 200. Αυτό σημαίνει ότι:

- Όλες οι τιμές που είναι μικρότερες ή ίσες με 80 θα αντιστοιχούν σε 0, δηλαδή θα απεικονίζονται ως μαύρο χρώμα.
- Όλες οι τιμές που είναι μεγαλύτερες ή ίσες με 200 θα αντιστοιχούν σε 255, δηλαδή θα απεικονίζονται ως άσπρο χρώμα.
- Οι ενδιάμεσες τιμές μεταξύ του 80 και του 200 θα κλιμακωθούν γραμμικά στο διάστημα $[0, 255]$, ανάλογα με την τιμή του κάθε pixel.

5.5 Σκοπός και χρησιμότητα

Η διαδικασία αυτή είναι χρήσιμη σε περιπτώσεις επεξεργασίας εικόνας ή ανάλυσης δεδομένων, όπου επιθυμούμε να κλιμακώσουμε τις τιμές των δεδομένων ώστε να βελτιώσουμε την αναγνωσιμότητα ή να ανταποκριθούν σε ένα προκαθορισμένο εύρος τιμών. Αυτή η τεχνική μπορεί να χρησιμοποιηθεί:

- Για την ενίσχυση της αντίθεσης εικόνας, όταν θέλουμε να αναδείξουμε περιοχές χαμηλής ή υψηλής φωτεινότητας.
- Στην ανάλυση δεδομένων, όταν απαιτείται να απομονώσουμε ή να απεικονίσουμε συγκεκριμένα εύρη τιμών.
- Στη μηχανική μάθηση, όταν απαιτείται η κανονικοποίηση δεδομένων εισόδου για την εκπαίδευση μοντέλων.

5.6 Πληροφορίες εικόνας

Το MATLAB προσφέρει τη δυνατότητα να εξετάσουμε λεπτομερείς πληροφορίες σχετικά με μια εικόνα, χρησιμοποιώντας τη συνάρτηση `imfinfo`. Αυτή η συνάρτηση παρέχει μια πληθώρα δεδομένων που σχετίζονται με τα χαρακτηριστικά της εικόνας, όπως ο τύπος αρχείου, οι διαστάσεις, το βάθος χρώματος και άλλα. Η βασική σύνταξη της συνάρτησης είναι η εξής:

`imfinfo filename`

Όπου το `filename` αντιπροσωπεύει το όνομα του αρχείου της εικόνας που θέλουμε να αναλύσουμε. Η συνάρτηση επιστρέφει μια δομή που περιέχει όλες τις σχετικές πληροφορίες της εικόνας, καθιστώντας την ιδιαίτερα χρήσιμη για εφαρμογές που απαιτούν επεξεργασία εικόνας ή έλεγχο των ιδιοτήτων των αρχείων. Για παράδειγμα, ας υποθέσουμε ότι έχουμε μια εικόνα με όνομα `tulips.jpg` στο MATLAB. Μπορούμε να χρησιμοποιήσουμε τη συνάρτηση `imfinfo` για να εξετάσουμε τα χαρακτηριστικά της εικόνας. Ακολουθεί ένα παράδειγμα κώδικα που δείχνει πώς μπορούμε να το επιτύχουμε:

```
Command Window
>> x=imread('tulips.jpg');
>> imfinfo tulips.jpg

ans =

    Filename: 'C:\Users\user\OneDrive\Υπολογιστής\Lab_2\tulips.jpg'
  FileModDate: '29-May-2024 17:09:37'
    FileSize: 86296
      Format: 'jpg'
FormatVersion: ''
      Width: 768
      Height: 512
    BitDepth: 24
    ColorType: 'truecolor'
FormatSignature: ''
NumberOfSamples: 3
   CodingMethod: 'Huffman'
   CodingProcess: 'Sequential'
        Comment: {}
```

Εικόνα 18:Αποτέλεσμα Matlab

Η εικόνα `tulips.jpg` είναι μία έγχρωμη φωτογραφία αποθηκευμένη σε μορφή JPEG (`jpg`), η οποία αποτελεί μία από τις πιο δημοφιλείς μορφές αρχείων εικόνας λόγω της ικανότητάς της να συνδυάζει καλή ποιότητα και μικρό μέγεθος αρχείου μέσω συμπίεσης. Οι διαστάσεις της εικόνας είναι 768 pixels σε πλάτος και 512 pixels σε ύψος, καθιστώντας την

ορθογώνια και ιδανική για ψηφιακή προβολή, είτε σε οθόνες είτε σε εφαρμογές που απαιτούν εικόνες μεσαίας ανάλυσης. Το βάθος χρώματος της εικόνας είναι 24-bit, το οποίο σημαίνει ότι κάθε pixel της εικόνας αποτελείται από 3 κανάλια χρώμα-τος (Red, Green, Blue - RGB), με το καθένα να διαθέτει 8 bits πληροφορίας. Αυτό επι-τρέπει την αναπαράσταση έως και 16,777,216 χρωμάτων (Truecolor), εξασφαλίζοντας πλούσιες και ρεαλιστικές χρωματικές αποχρώσεις. Το χαρακτηριστικό αυτό καθιστά την εικόνα κατάλληλη για εφαρμογές που απαιτούν υψηλή πιστότητα χρωμάτων, όπως ψηφιακή φωτογραφία ή επεξεργασία εικόνας. Το αρχείο έχει μέγεθος 86,296 bytes (~86 KB), γεγονός που οφείλεται στη χρήση της συμπίεσης Huffman, μιας τεχνικής συμπίεσης δεδομένων χωρίς απώλειες που χρησιμοποιείται συχνά σε JPEG αρχεία για να μειώσει το μέγεθος, διατηρώντας παράλληλα την οπτική ποιότητα της εικόνας. Επιπλέον, η διαδικασία κωδικοποίησης είναι Sequential, κάτι που σημαίνει ότι τα δεδομένα της εικόνας αποθηκεύονται και διαβάζονται σειριακά. Αυτή η μέθοδος είναι πιο αποδοτική για πολλές εφαρμογές και συμβάλλει στη γρήγορη απόδοση της εικόνας όταν φορτώνεται ή προβάλλεται. Συνολικά, η εικόνα tulips.jpg αποτελεί ένα χαρακτηριστικό παράδειγμα αρχείου JPEG με καλή ισορροπία ανάμεσα σε ποιότητα και αποδοτική αποθήκευση. Τα χαρακτηριστικά της την καθιστούν ιδανική για προβολή σε ψηφιακές πλατφόρμες και για χρήση σε εργασίες που περιλαμβάνουν επεξεργασία ή ανάλυση εικόνας, χωρίς να απαιτεί υπερβολικά μεγάλο αποθηκευτικό χώρο.

5.7 Περικοπή εικόνας

Το MATLAB παρέχει τη δυνατότητα περικοπής μιας εικόνας, επιτρέποντας τη δημιουργία τμημάτων της αρχικής εικόνας που μας ενδιαφέρουν περισσότερο. Αυτό μας δίνει τη δυνατότητα να εστιάσουμε σε συγκεκριμένες περιοχές της εικόνας για λεπτομερή παρατήρηση ή να προσομοιώσουμε ένα είδος μεγέθυνσης, zoom, σε ένα συγκεκριμένο σημείο. Επιπλέον, μπορούμε να απεικονίσουμε μόνο ένα επιλεγμένο τμήμα της εικόνας, αγνοώντας τα υπόλοιπα. Η περικοπή εικόνας γίνεται μέσω ενός διαδραστικού εργαλείου που παρέχεται από το MATLAB. Το εργαλείο αυτό εμφανίζεται πάνω από την εικόνα που έχει φορτωθεί σε ένα figure και ονομάζεται «εικόνα στόχου» [Χάτζικος, 2010]. Το εργαλείο περικοπής είναι ένα ορθογώνιο πλαίσιο, το οποίο μπορεί να μετακινηθεί, να αλλάξει μέγεθος και να προσαρμοστεί μέσω του ποντικιού. Ο χρήστης μπορεί να χρησι-

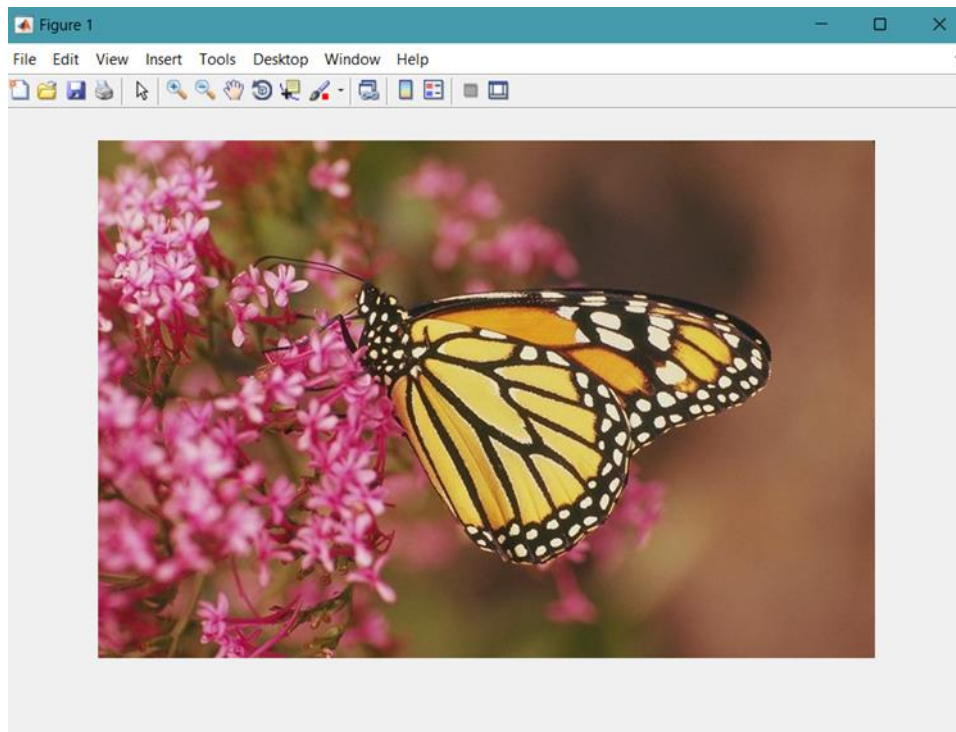
μπουήσει το πλαίσιο αυτό για να καθορίσει την περιοχή της εικόνας που θέλει να διατηρήσει.

Η διαδικασία ολοκληρώνεται με δύο τρόπους: είτε με διπλό κλικ μέσα στην περιοχή του ορθογωνίου, είτε με δεξί κλικ και επιλογή της εντολής Crop Image από το μενού που εμφανίζεται. Μετά την εκτέλεση της περικοπής, εμφανίζεται η επιλεγμένη περιοχή της εικόνας. Η σύνταξη της εντολής στο MATLAB είναι απλή και μπορεί να εφαρμοστεί σε οποιαδήποτε εικόνα, κάνοντάς την ιδανική για εφαρμογές που απαιτούν εστίαση ή προσαρμογή στην ανάλυση συγκεκριμένων τμημάτων της εικόνας.

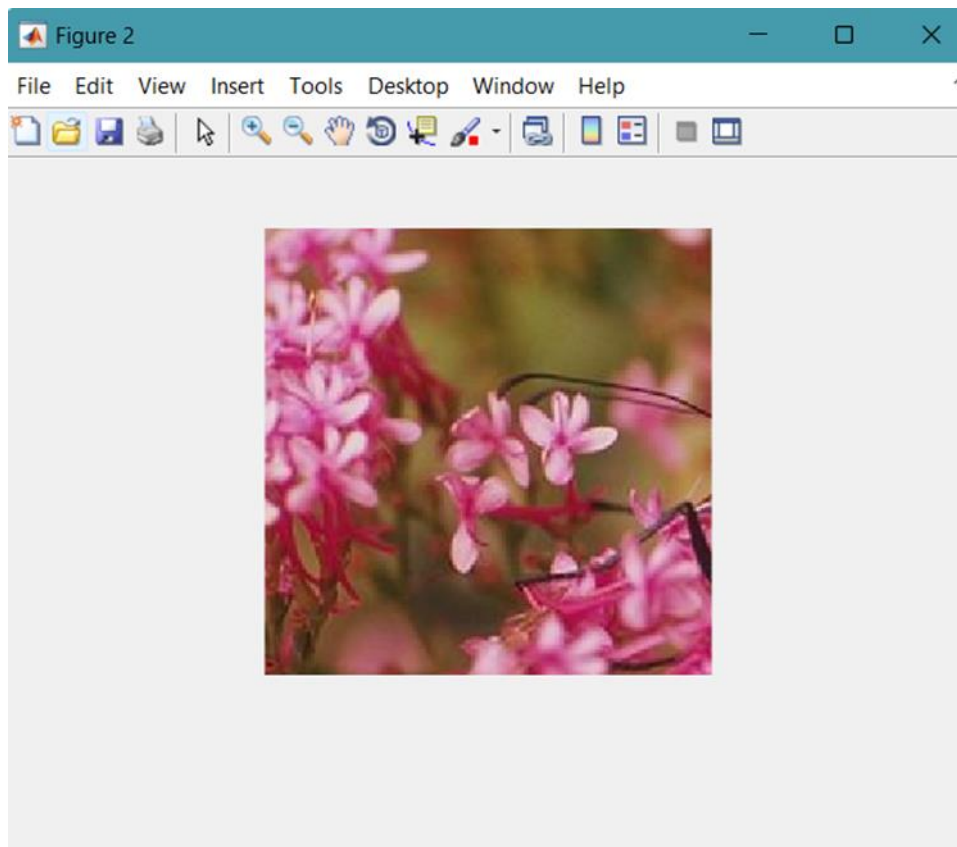
Αυτό το εργαλείο είναι ιδιαίτερα χρήσιμο για εργασίες ανάλυσης εικόνων, επεξεργασίας και οπτικοποίησης, καθώς προσφέρει έναν εύκολο και διαδραστικό τρόπο να επισημά-
νουμε και να απομονώσουμε τα πιο σημαντικά τμήματα μιας εικόνας.

Y=imcrop(x);

```
x = imread('monarch.jpg');  
figure(1);imshow(x);  
rect = [50, 50, 200, 200];  
y = imcrop(x, rect);  
figure(2);imshow(y);
```



Εικόνα 19: Πριν την περικοπή της



Εικόνα 20: Μετά την περικοπή της

5.8 Περιστροφή εικόνας

Το MATLAB μας προσφέρει τη δυνατότητα να πραγματοποιήσουμε περιστροφή μιας εικόνας, επιτρέποντάς μας να επεξεργαστούμε και να αλλάξουμε τη γωνία προβολής της. Για να διευκολυνθεί η διαδικασία και να γίνει πιο κατανοητή, θα χρησιμοποιήσουμε μια εικόνα στα επίπεδα του γκρι (grayscale), καθώς είναι πιο απλή στη διαχείριση σε σχέση με μια έγχρωμη εικόνα.

Η περιστροφή της εικόνας βασίζεται σε έναν πίνακα περιστροφής, rotation matrix, ο οποίος ορίζεται ως εξής:

$$M = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Εικόνα 21: Τύπος περιστροφής εικόνας

Όπου:

- θ είναι η γωνία περιστροφής σε μοίρες ή ακτίνια

- Οι συναρτήσεις $\cos$ και $\sin$ καθορίζουν τη γεωμετρική σχέση για την περιστροφή κάθε σημείου της εικόνας.

5.8.1 Πως λειτουργεί η περιστροφή

Η διαδικασία περιστροφής επηρεάζει τις συντεταγμένες κάθε pixel της εικόνας. Ο παραπάνω πίνακας περιστροφής εφαρμόζεται για να υπολογιστούν οι νέες θέσεις των pixel μετά την περιστροφή. Τα νέα σημεία προκύπτουν από τη μετασχηματιστική σχέση:

$$\begin{bmatrix} x_{\text{new}} \\ y_{\text{new}} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x_{\text{old}} \\ y_{\text{old}} \end{bmatrix}$$

Εικόνα 22: Διαδικασία περιστροφής εικόνας

Όταν εφαρμόζουμε μια περιστροφή σε μια εικόνα με τη χρήση ενός πίνακα περιστροφής, πρέπει να εξετάσουμε πώς υπολογίζονται οι νέες συντεταγμένες κάθε pixel. Η διαδικασία αυτή απαιτεί μια προσεκτική διαχείριση των συντεταγμένων και της παραμόρφωσης της εικόνας, καθώς κάθε σημείο (pixel) μετατοπίζεται και το νέο του σημείο υπολογίζεται με βάση τον πίνακα περιστροφής.

5.8.2 Διαδικασία δημιουργίας του πίνακα συντεταγμένων

- Ορισμός Συντεταγμένων Pixel: Στο αρχικό βήμα, δημιουργούμε έναν πίνακα που περιέχει τις συντεταγμένες (X, Y) όλων των pixel της εικόνας. Αυτές οι συντεταγμένες αναφέρονται στις θέσεις των pixel στον άξονα X (στήλες) και Y (γραμμές). Οι αριθμοί των γραμμών και στηλών είναι αριθμητικά αναγνωρίσιμα στο πλέγμα της εικόνας, αλλά οι συντεταγμένες (X, Y) εκφράζουν τη θέση κάθε pixel στο χώρο της εικόνας.
- Εφαρμογή του Πίνακα Περιστροφής: Όταν περιστρέφουμε την εικόνα κατά μια γωνία θ , οι νέες συντεταγμένες των pixel υπολογίζονται από τον πίνακα περιστροφής:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Εικόνα 23: Πίνακας περιστροφής

όπου (x, y) είναι οι αρχικές συντεταγμένες και (x', y') οι νέες.

Στο script αυτό, θα υλοποιήσετε τον παρακάτω κώδικα για να μετατρέψετε την εικόνα σε μια νέα διάταξη μέσω της στροφής της

```

X = [0 1 0 0 0; 0 0 1 1 1; 0 0 0 0 0; 0 0 0 1 0; 0 0 0
0 1];
Y = shiftright(X);
disp(Y);
function [Y] = shiftright(X)
for i = 1:size(X,1)
    for j = 1:size(X,2)
        Y(i,size(X,1) + 1 - j) = X(j,i);
    end
end
end
end

```

Αρχικός πίνακας

0	1	0	0	0
0	0	1	1	1
0	0	0	0	0
0	0	0	1	0
0	0	0	0	1

Περιστραμμένος πίνακας

0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	1	0	1	0
1	0	0	1	0

Παρόμοιες ασκήσεις για εξάσκηση θα βρείτε στο **Παράρτημα Α** και πιο συγκεκριμένα στο **Lab_3. [σελ.118]**

5.8.3 Στρογγυλοποίηση συντεταγμένων

Κατά την εφαρμογή της περιστροφής, οι νέες συντεταγμένες (x' , y') δεν είναι πάντα ακέραιες τιμές. Συχνά, οι υπολογισμένες τιμές των pixel είναι δεκαδικές, και επομένως πρέπει να στρογγυλοποιήσουμε τα αποτελέσματα για να τις μετατρέψουμε σε ακέραιους αριθμούς. Αυτή η διαδικασία μπορεί να οδηγήσει σε περιπτώσεις όπου δύο γειτονικά pixels, μετά την περιστροφή, καταλήγουν να έχουν τις ίδιες συντεταγμένες.

5.8.4 Προβλήματα και κενά στην εικόνα

Η στρογγυλοποίηση των συντεταγμένων προκαλεί αρκετά προβλήματα, ειδικά όταν δυο κοντινά ή γειτονικά pixels μετασχηματίζονται στη ίδια θέση μετά την περιστροφή. Αυτή η κατάσταση οδηγεί στην απώλεια των αρχικών λεπτομερειών της εικόνας, καθώς τα pixels «συγχωνεύονται» και αντιπροσωπεύονται από τις ίδιες συντεταγμένες στην περιστραμμένη εικόνα. Επιπλέον, κατά την περιστροφή, μπορεί να προκύψουν κενά ή περιοχές που δεν περιέχουν κανένα pixel, ειδικά στις γωνίες της εικόνας. Αυτά τα κενά προκύπτουν επειδή δεν υπάρχει αντιστοιχία σε μια ή περισσότερες από τις νέες θέσεις των

pixel, αφού η περιστροφή μπορεί να προκαλέσει τη δημιουργία περιοχών που δεν καλύπτονται από την αρχική εικόνα.

5.8.5 Αντιμετώπιση κενών

Για να αντιμετωπίσουμε τα κενά και τις συγχωνευμένες συντεταγμένες, μπορούμε να χρησιμοποιήσουμε τεχνικές όπως:

- **Παρεμβολή:** Χρησιμοποιώντας τεχνικές παρεμβολής μπορούμε να υπολογίσουμε τις τιμές των pixel στις περιοχές που προκύπτουν μετά την περιστροφή και να γεμίσουμε τα κενά.
- **Γέμισμα Κενών:** Εναλλακτικά, μπορούμε να γεμίσουμε τα κενά με μια καθορισμένη τιμή (όπως μαύρο ή λευκό χρώμα) για να εξασφαλίσουμε ότι η εικόνα είναι πλήρης, χωρίς να προκαλούνται σημαντικές παραμορφώσεις.

5.8.6 Μετασχηματισμοί και συσχέτιση

Οι μετασχηματισμοί με συσχέτιση εκφράζουν τη διαδικασία εφαρμογής του πίνακα περιστροφής στις συντεταγμένες των pixel και συνήθως περιλαμβάνουν τη χρήση συσχέτισης ή μετατόπισης των pixels για να ταιριάζουν στις νέες θέσεις. Οι εξισώσεις μετασχηματισμού συνήθως περιλαμβάνουν:

- **Αναστροφή Συντεταγμένων:** Αντί να υπολογίσουμε τις νέες θέσεις των pixel, υπολογίζουμε τις αρχικές θέσεις για κάθε νέο pixel.
- **Αναδιάταξη με Παρεμβολή:** Με τον υπολογισμό της θέσης κάθε pixel στο νέο σύστημα, χρησιμοποιούμε τις μεθόδους παρεμβολής για να εξασφαλίσουμε ότι οι τιμές των νέων pixel υπολογίζονται με βάση τις γειτονικές τιμές.

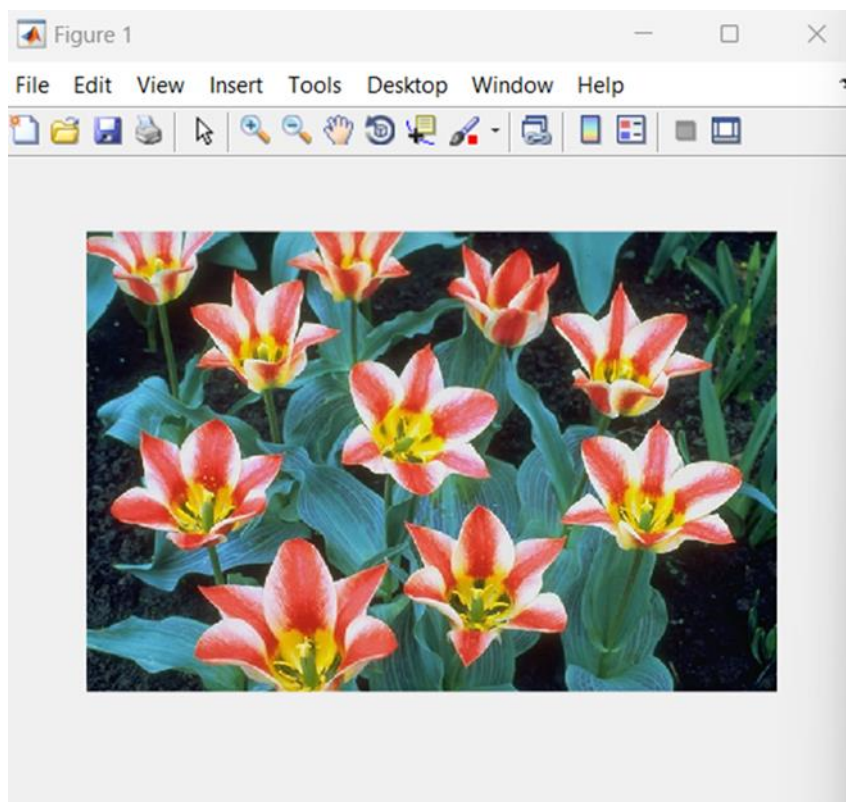
5.8.7 Διατήρηση της ποιότητας

Η διατήρηση της ποιότητας μιας εικόνας κατά την περιστροφή είναι ένα σημαντικό ζήτημα στην επεξεργασία εικόνας. Η περιστροφή, ειδικά όταν δεν είναι πολλαπλάσιο των 90°, μπορεί να προκαλέσει αλλαγές στη διάταξη των pixel, εισάγοντας κενά (μαύρες ή λευκές περιοχές) στις γωνίες της εικόνας ή παραμορφώσεις στο περιεχόμενο. Αυτό συμβαίνει επειδή οι συντεταγμένες των pixel της εικόνας πρέπει να μετασχηματιστούν με βάση τον πίνακα περιστροφής, και σε πολλές περιπτώσεις τα νέα pixel δεν ταιριάζουν ακριβώς στο αρχικό πλέγμα. Για την αντιμετώπιση αυτών των προβλημάτων, το

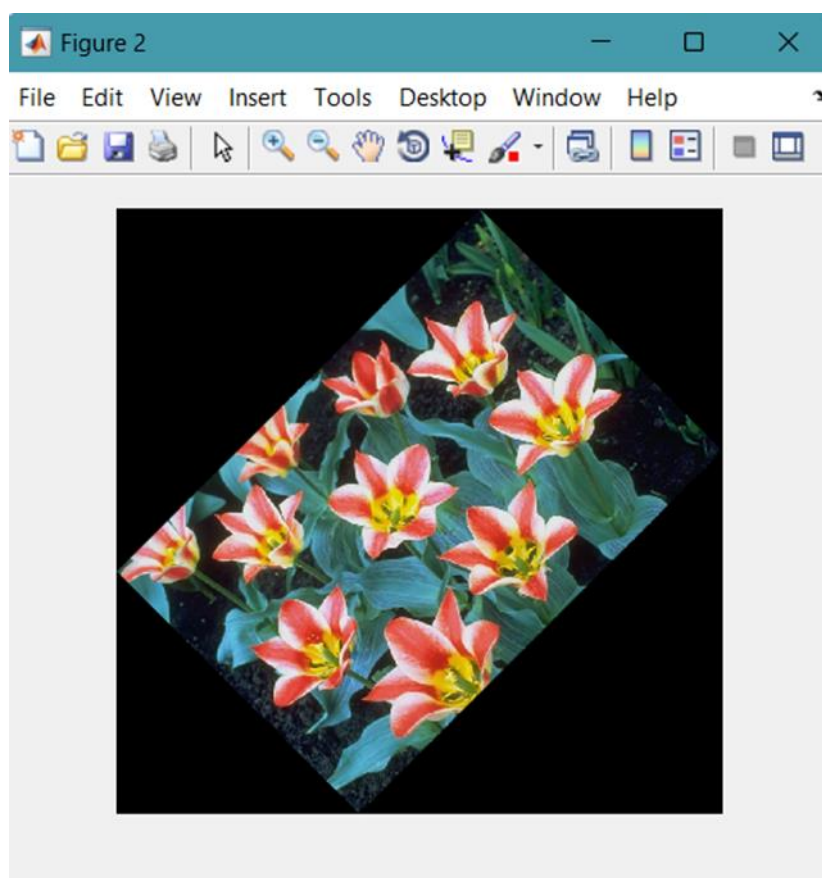
MATLAB διαθέτει ενσωματωμένες λειτουργίες που εξασφαλίζουν την ομαλή εκτέλεση της περιστροφής, χωρίς να υποβαθμίζεται η ποιότητα της εικόνας. Συγκεκριμένα:

- **Αναπροσαρμογή Μεγέθους Εικόνας:** Όταν μια εικόνα περιστρέφεται, το MATLAB επεκτείνει αυτόματα τις διαστάσεις της για να εξασφαλίσει ότι ολόκληρη η περιστραμμένη εικόνα χωράει στο νέο πλαίσιο. Αυτό σημαίνει ότι η εικόνα που προκύπτει μπορεί να έχει μεγαλύτερο μέγεθος από την αρχική, καθώς το MATLAB δημιουργεί ένα νέο ορθογώνιο πλαίσιο που περιέχει όλες τις νέες συντεταγμένες των pixel. Αυτή η προσέγγιση αποτρέπει την απώλεια δεδομένων ή την αποκοπή τμημάτων της εικόνας.
- **Γέμισμα Κενών Περιοχών:** Μετά την περιστροφή, στις γωνίες της νέας εικόνας ενδέχεται να εμφανιστούν κενά, τα οποία γεμίζονται από το MATLAB με προεπιλεγμένα χρώματα (συνήθως μαύρο). Ωστόσο, ο χρήστης μπορεί να προσαρμόσει αυτά τα κενά σύμφωνα με τις απαιτήσεις του, επιλέγοντας διαφορετική τιμή γεμίσματος.
- **Παρεμβολή:** Κατά την περιστροφή, οι νέες θέσεις των pixel δεν αντιστοιχούν πάντα σε ακέραιες τιμές συντεταγμένων. Το MATLAB χρησιμοποιεί μεθόδους παρεμβολής, για να υπολογίσει τις τιμές των νέων pixel από τις γειτονικές τους.
- **Διατήρηση της ανάλυσης:** Η περιστροφή μπορεί να επηρεάσει την ευκρίνεια της εικόνας, ιδιαίτερα αν περιέχει λεπτομέρειες όπως γραμμές ή σχήματα. Το MATLAB, μέσω των προηγμένων τεχνικών παρεμβολής και προσαρμογής του πλέγματος των pixel, διασφαλίζει ότι η ανάλυση και η λεπτομέρεια της εικόνας παραμένουν όσο το δυνατόν πιο κοντά στην αρχική.
- **Προσαρμογή τύπου εξόδου:** Αν ο χρήστης επιλέξει τη ρύθμιση crop, το MATLAB περιορίζει τη νέα εικόνα στο μέγεθος της αρχικής, κόβοντας τυχόν έξ-τρα περιοχές. Αυτό είναι χρήσιμο για συγκεκριμένες εφαρμογές όπου απαιτείται σταθερό μέγεθος εικόνας.

```
k = imread('tulips.jpg');  
n = imrotate(k,45,'bilinear');  
figure(1);imshow(k)  
figure(2);imshow(n)
```



Εικόνα 24:Πριν την περιστροφή της



5.9 Ιστόγραμμα

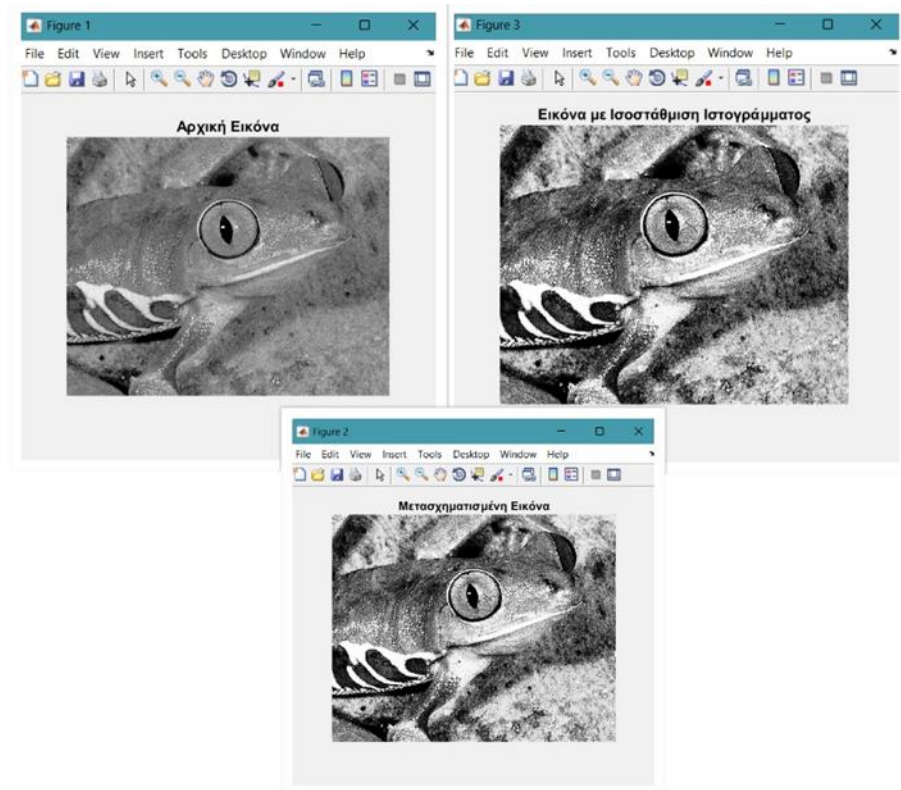
Μια εξαιρετικά σημαντική μέθοδος στην ανάλυση και επεξεργασία ψηφιακών εικόνων είναι η αντίληψη της φωτεινότητας μιας εικόνας ως συνάρτησης $f(i,j)$. Στο πλαίσιο αυτό, ο οριζόντιος άξονας i υποδηλώνει τα επίπεδα έντασης φωτός (σε ένα σύστημα 8-bit, αυτά κυμαίνονται από 0 έως 255), ενώ ο κατακόρυφος άξονας j αντιπροσωπεύει τον αριθμό των εικονοστοιχείων (pixels) που ανταποκρίνονται σε κάθε επίπεδο έντασης. Αυτή η φωτεινότητα μπορεί να θεωρηθεί ως μια στοχαστική μεταβλητή, η οποία περιγράφεται από μια συνάρτηση κατανομής πιθανότητας ή πυκνότητας πιθανότητας. Η δομή αυτής της συνάρτησης περιλαμβάνει κρίσιμες πληροφορίες για τα χαρακτηριστικά της εικόνας. Ωστόσο, δεδομένου ότι η ακριβής συνάρτηση πυκνότητας πιθανότητας σπανίως είναι άμεσα διαθέσιμη, χρειάζεται να υπολογιστεί ή να εκτιμηθεί από την ίδια την εικόνα. Αυτή η εκτίμηση πραγματοποιείται μέσω του ιστογράμματος, το οποίο αντιπροσωπεύει την εμπειρική κατανομή πιθανότητας. Το ιστόγραμμα μιας εικόνας περιέχει ουσιώδεις πληροφορίες για τη δομή και το περιεχόμενό της. Συνοψίζει τη διασπορά των επιπέδων φωτεινότητας ή χρωμάτων σε ένα σύνολο pixels, αποτυπώνοντας τη συχνότητα εμφάνισης κάθε επιπέδου. Με άλλα λόγια, το ιστόγραμμα παρέχει μια οπτική αναπαράσταση της κατανομής της έντασης ή της φωτεινότητας μιας εικόνας, περιγράφοντας πώς κατανέμονται τα χρώματα ή οι τόνοι φωτός σε αυτή.

Χρησιμοποιώντας το ιστόγραμμα, μπορούμε να αποκτήσουμε πολύτιμες γνώσεις για την κατανομή των χαρακτηριστικών της εικόνας, όπως η αντίθεση, η φωτεινότητα ή οι τυχόν ανισορροπίες στη χρωματική της παλέτα.

Το ιστόγραμμα μιας εικόνας αποτελεί ένα πολύτιμο εργαλείο για την ανάλυση, την αξιολόγηση, τη βελτιστοποίηση, την τροποποίηση και την εξαγωγή χαρακτηριστικών της εικόνας. Παρέχει σημαντικές πληροφορίες για το χρώμα, όπως η φωτεινότητα, η αντίθεση και η χρωματική ισορροπία, καθιστώντας το χρήσιμο για την ανάλυση και την αξιολόγηση της ποιότητας της εικόνας. Η γραφική αναπαράσταση που προσφέρει το ιστόγραμμα αποτυπώνει τη συχνότητα εμφάνισης των διαφορετικών τιμών φωτεινότητας στην εικόνα. Δείχνει πόσα pixels αντιστοιχούν σε κάθε επίπεδο έντασης, χωρίς όμως να παρέχει πληροφορίες για τη γεωγραφική κατανομή αυτών των τιμών μέσα στην εικόνα. Για παράδειγμα, μπορεί να αποκαλύψει αν μια εικόνα περιέχει σκοτεινές περιοχές, υ-

περβολικά φωτεινές ή έχει έλλειψη αντίθεσης, χωρίς να καθορίζει τη θέση αυτών των περιοχών.

```
I = imread('frog.png');
I = I(:,:,1);
horizontia_diastash = size(I,2);
katakoryfh_diastash = size(I,1);
histogram = zeros(1,256);
I_equal = zeros(size(I), 'uint8');
for level = 0:255
    I_level = (I == level);
    histogram(level+1) = sum(I_level(:));
end
cum_hist = cumsum(histogram);
norm_cum_hist = floor(cum_hist / (horizontia_diastash * katakoryfh_diastash) * 255);
for i = 1:size(I,1)
    for j = 1:size(I,2)
        I_equal(i,j) = norm_cum_hist(I(i,j)+1);
    end
end
figure, imshow(I);
title('Αρχική Εικόνα');
figure, imshow(I_equal);
title('Μετασχηματισμένη Εικόνα');
hist = imhist(I);
J = histeq(I);
figure, imshow(J);
title('Εικόνα με Ισοστάθμιση Ιστογράμματος');
```



Εικόνα 26: Αποτέλεσμα Matlab

5.9.1 Εφαρμογές ιστογράμματος

- Για τη διόρθωση φωτεινότητας και αντίθεσης, βελτιώνοντας την οπτική ποιότητα.
- Για την αναγνώριση ανισορροπιών στη χρωματική παλέτα της εικόνας.
- Για την εξαγωγή χαρακτηριστικών που σχετίζονται με τη δομή και το περιεχόμενο της εικόνας.

5.9.2 Διάγνωση ελλείψεων

Το ιστόγραμμα είναι ιδιαίτερα χρήσιμο για την ανίχνευση ανεπαρειών στην ποιότητα της εικόνας. Παρουσιάζει άμεσα αν η εικόνα έχει χαμηλή αντίθεση, αν κυριαρχούν υπερβολικά σκοτεινές ή φωτεινές περιοχές ή αν η φωτεινότητα της εικόνας είναι ισορροπημένη.

5.9.3 Ορισμός ιστογράμματος

Για μια εικόνα με L επίπεδα φωτεινότητας που κυμαίνονται στην περιοχή $[0, 255]$, το ιστόγραμμα ορίζεται με την παρακάτω διακριτή συνάρτηση

Όπου:

- rk: Το επίπεδο φωτεινότητας k.
- nk: Ο αριθμός των pixels που έχουν την τιμή φωτεινότητας rk.

Στο MATLAB, για να υπολογίσουμε το ιστόγραμμα μιας εικόνας, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση:

`h=imhist(I,b)`

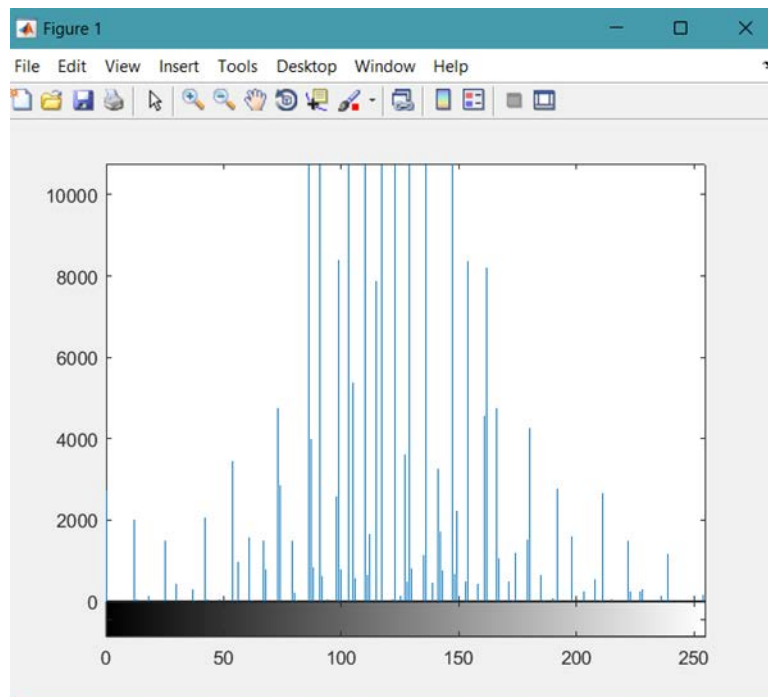
Η χρήση του `imhist` είναι ουσιώδης για τη διερεύνηση της ποιότητας της εικόνας και την προετοιμασία της για περαιτέρω επεξεργασία, όπως διόρθωση φωτεινότητας ή βελτίωση της αντίθεσης. Για την ανάλυση εικόνων που χρησιμοποιούν χρωματικό δείκτη, το MATLAB παρέχει τη συνάρτηση:

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

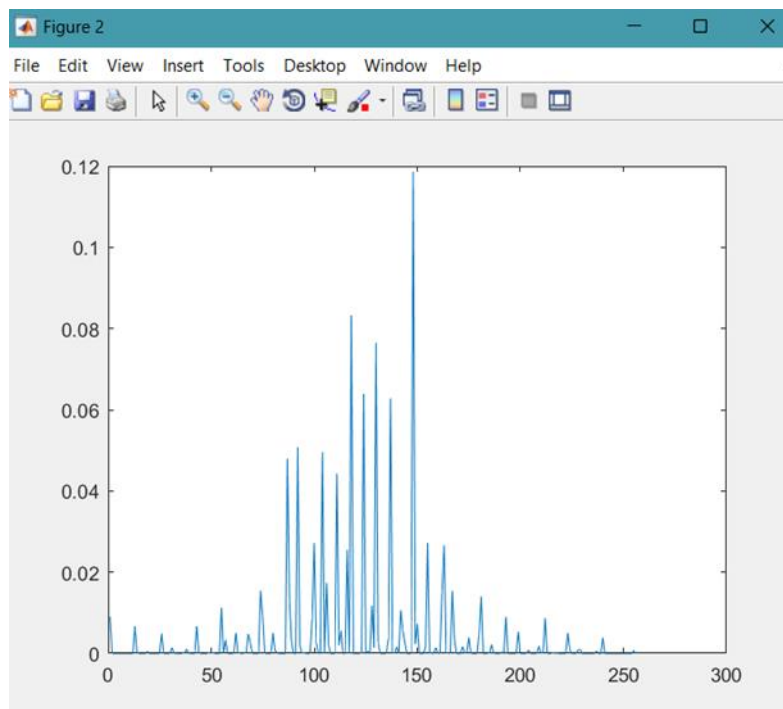
Αυτή η συνάρτηση υπολογίζει το ιστόγραμμα της εικόνας `X` χρησιμοποιώντας τον χρωματικό χάρτη `map` που τη συνοδεύει. Αντί για επίπεδα φωτεινότητας, όπως στις εικόνες σε γκριζες αποχρώσεις, εδώ το ιστόγραμμα αναλύει την κατανομή των χρωμάτων σύμφωνα με τον καθορισμένο χρωματικό χάρτη. Η συνάρτηση `numel(I)` στο MATLAB χρησιμοποιείται για τον υπολογισμό του συνολικού αριθμού των pixels μιας εικόνας εισόδου `I`. Αυτός ο αριθμός είναι σημαντικός για πολλές εφαρμογές επεξεργασίας εικόνας, ιδιαίτερα όταν απαιτείται η κανονικοποίηση δεδομένων ή η ανάλυση της πυκνότητας κατανομής φωτεινότητας. Η χρήση των συναρτήσεων `numel`, `imhist`, και `plot` διευκολύνει τη βαθιά κατανόηση της κατανομής φωτεινότητας ή χρωμάτων μιας εικόνας, επιτρέποντας την ανάλυση, τη σύγκριση, και την εφαρμογή βελτιστοποιήσεων, όπως η εξίσωση ιστογράμματος ή η βελτίωση της αντίθεσης.

```
x=imread('frog.png');
h=imhist(x);
h(100:105)
ans =
    8407
    789
     0
     0
   15287
     0
n=size(x,1)*size(x,2)
n =
   309258
p=h/n;
p(100:105) '
ans =
    0.0272    0.0026     0     0    0.0494     0
figure(1);imhist(x)
figure(2);plot(p);
```

Εικόνα 27:Κώδικας Matlab



Εικόνα 28:Αρχικό ιστόγραμμα



Εικόνα 29:Τελικό ιστόγραμμα

Μια εικόνα με ισορροπημένη αντίθεση και φωτεινότητα χαρακτηρίζεται από ένα ιστόγραμμα που κατανέμεται ομαλά σε όλο το φάσμα των τιμών έντασης, από 0 έως 255. Στις περιπτώσεις αυτές, οι ακραίες τιμές είναι όσο το δυνατόν πιο κοντά στα όρια του

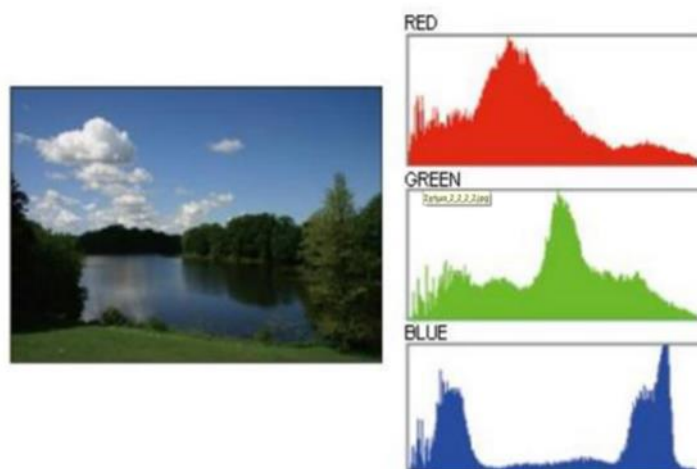
φάσματος, ενώ τα pixels κατανέμονται ομοιόμορφα στις ενδιάμεσες τιμές. Αντίθετα, μια εικόνα με χαμηλή φωτεινότητα εμφανίζει συσσώρευση pixels στις χαμηλές τιμές έντασης (κοντά στο 0), ενώ μια πολύ φωτεινή εικόνα δείχνει συγκέντρωση pixels προς τις υψηλές τιμές (κοντά στο 255). Εικόνες με μειωμένη αντίθεση παρουσιάζουν ιστόγραμμα που είναι συμπιεσμένο σε μια στενή περιοχή του φάσματος, δημιουργώντας συστάδες. Η θέση αυτής της συστάδας υποδεικνύει αν η εικόνα είναι πολύ σκοτεινή, υπερβολικά φωτεινή ή μονοτονικά γκρίζα. Επιπλέον, ένα ιστόγραμμα με μεγάλες τιμές διασποράς που περιλαμβάνει ένα τοπικό μέγιστο μπορεί να αποτελεί ένδειξη παρουσίας παρασκηνίου στην εικόνα.

Για περισσότερη εξάσκηση σε παρόμοιες ασκήσεις ανατρέξτε στο **Παράρτημα Α** και πιο συγκεκριμένα στο **Lab_4. [σελ. 138]**

5.10 Ιστόγραμμα χρώματος

Το χρωματικό ιστόγραμμα, που χρησιμοποιείται σε εικόνες RGB, παρέχει πληροφορίες για την κατανομή των χρωμάτων. Ωστόσο, η χρήση του απαιτεί προσοχή, καθώς:

- Εικόνες με πολύ διαφορετικά αντικείμενα μπορεί να έχουν παρόμοια χρωματικά ιστόγραμμα, γεγονός που περιορίζει την αξία του για συγκρίσεις.
- Εικόνες με το ίδιο αντικείμενο αλλά διαφορετικά χρώματα εμφανίζουν εντελώς διαφορετικά ιστόγραμμα, παρά την ομοιότητα του περιεχομένου τους.
- Το χρωματικό ιστόγραμμα αγνοεί πλήρως το σχήμα και είναι ευαίσθητο σε μεταβολές φωτεινότητας, καθιστώντας το λιγότερο αξιόπιστο για ορισμένες εφαρμογές.



Εικόνα 30: Χρωματικό ιστόγραμμα. [8]

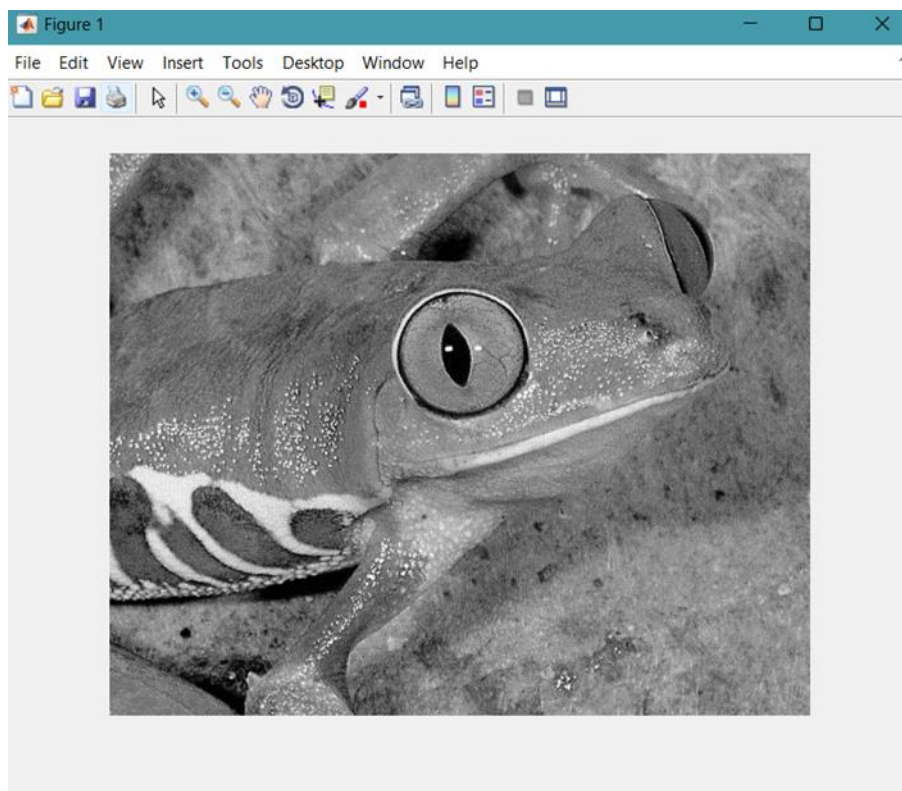
Τέλος, το MATLAB προσφέρει τη δυνατότητα εφαρμογής της τεχνικής εξισορρόπησης ιστογράμματος σε μια εικόνα, μια διαδικασία που ενισχύει την αντίθεση της εικόνας βελτιώνοντας την οπτική της ποιότητα. Η τεχνική αυτή βασίζεται στη μετατροπή των επιπέδων φωτεινότητας της εικόνας έτσι ώστε να κατανέμονται ομοιόμορφα σε όλο το φάσμα των τιμών φωτεινότητας, συνήθως από 0 έως 255 για εικόνες 8-bit.

5.11 Πλεονεκτήματα της εξισορρόπησης ιστογράμματος

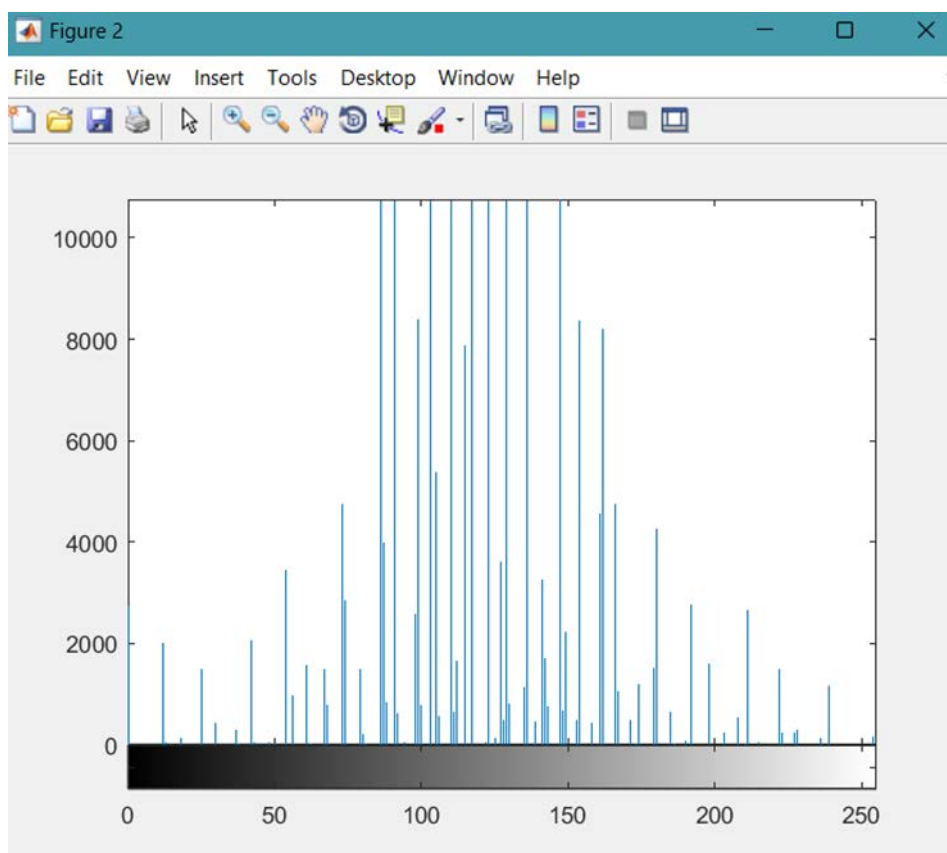
- **Βελτίωση Αντίθεσης:** Η προκύπτουσα εικόνα παρουσιάζει αυξημένη αντίθεση, με αποτέλεσμα να είναι πιο ευκρινής και ευδιάκριτη.
- **Εξίσου Κατανεμημένα Pixels:** Τα επίπεδα φωτεινότητας αναδιατάσσονται ώστε να αξιοποιείται πλήρως το δυναμικό εύρος της εικόνας.
- **Καταλληλότητα για Υποβαθμισμένες Εικόνες:** Εικόνες με χαμηλή αντίθεση, όπως αυτές που είναι πολύ σκοτεινές ή φωτεινές, βελτιώνονται σημαντικά.

```
x=imread('frog.png');  
figure (1);imshow(x)  
figure(2);imhist(x)  
y=im2double(x);  
w=imadjust(y,[min(y(:)),max(y(:))],[0 1]);  
figure(3);imhist(w)  
figure(4);imshow(w)
```

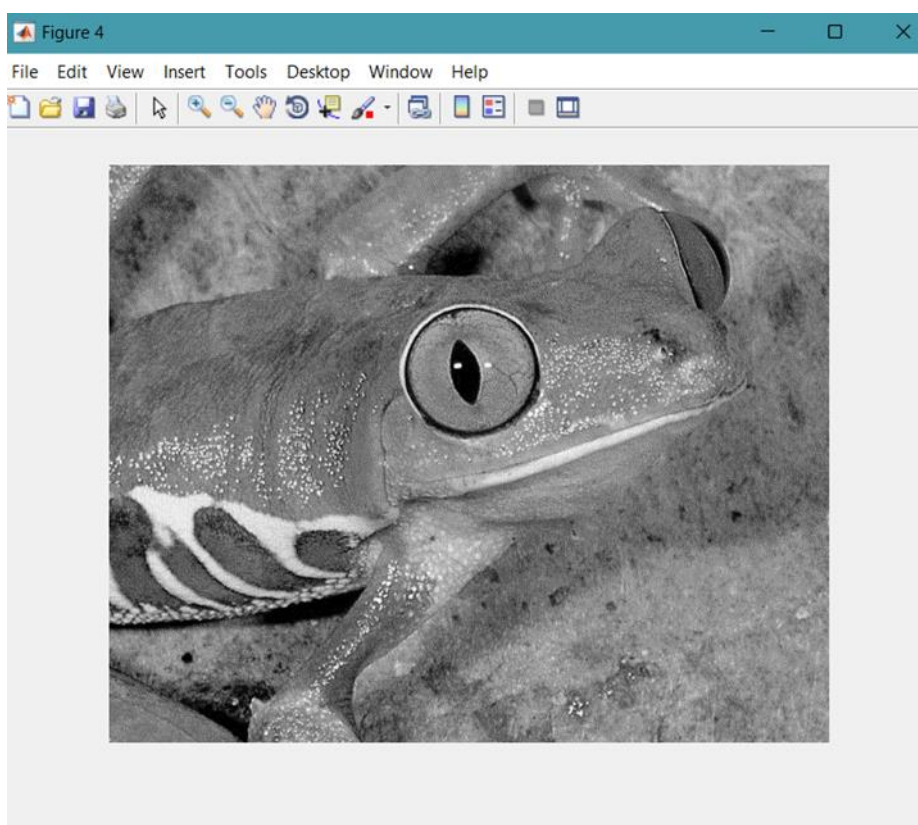
Εικόνα 31:Κώδικας Matlab



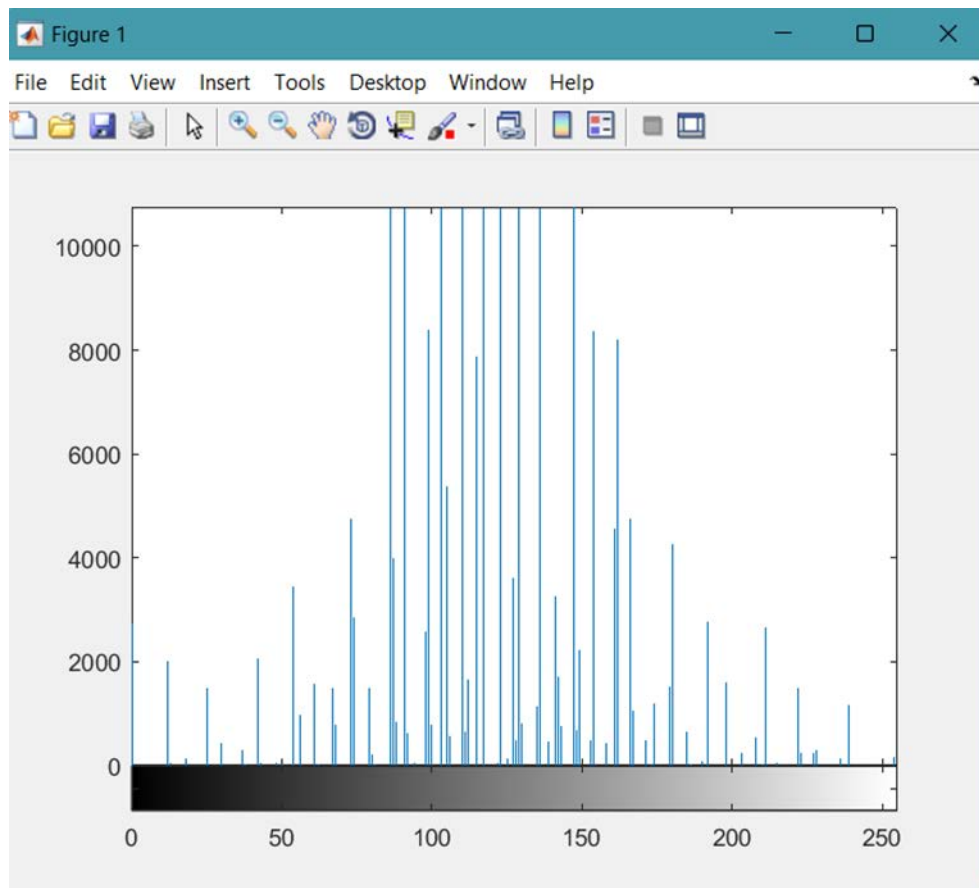
Εικόνα 32: Απεικόνιση εικόνας χρωματικού δείκτη



Εικόνα 33: Ιστόγραμμα εικόνας χρωματικού δείκτη



Εικόνα 34: Εικόνα μετά την ισοστάθμιση



Εικόνα 35: Ιστόγραμμα ισοσταθμισμένο

Στην αρχική εικόνα του παραδείγματος, μπορούμε να παρατηρήσουμε την αρχική κατανομή των pixels, η οποία συνήθως έχει περιορισμένη αντίθεση και φωτεινότητα. Στην επόμενη εικόνα, το ιστόγραμμα της αρχικής εικόνας αποκαλύπτει ότι οι τιμές των pixels είναι συγκεντρωμένες κυρίως γύρω από το κέντρο του φάσματος, δηλαδή στο μέσο της κλίμακας φωτεινότητας. Αυτό υποδηλώνει ότι η εικόνα έχει περιορισμένη αντίθεση, με την πλειονότητα των εικονοστοιχείων να έχουν παρόμοιες τιμές φωτεινότητας. Στην επόμενη εικόνα, η οποία δείχνει την εικόνα μετά την επεξεργασία, η εικόνα παρουσιάζει αυξημένη αντίθεση. Αυτό επιτυγχάνεται μέσω της εξισορρόπησης ιστογράμματος, η οποία κατανέμει τις τιμές των pixels σε όλο το εύρος της φωτεινότητας. Η επεξεργασμένη εικόνα εμφανίζεται πιο ευκρινής, με τις λεπτομέρειες να γίνονται πιο ευδιάκριτες, ιδιαίτερα σε σκοτεινές ή φωτεινές περιοχές που προηγουμένως ήταν λιγότερο ξεκάθαρες. Τέλος, στην τελευταία εικόνα, το ιστόγραμμα της εξισορροπημένης εικόνας δείχνει μια ομοιόμορφη κατανομή των τιμών των pixels σε όλο το εύρος του ιστογράμματος. Οι τιμές φωτεινότητας είναι τώρα κατανομημένες σχεδόν εξίσου σε όλο το φάσμα, από το 0 έως

το 255, κάτι που σημαίνει ότι η εικόνα έχει αποκτήσει ομοιόμορφη αντίθεση και φωτεινότητα. Αυτό έχει ως αποτέλεσμα την ενίσχυση των λεπτομερειών και τη βελτίωση της οπτικής ποιότητας της εικόνας.

6 Επεξεργασία Βίντεο

6.1 Toolboxes και Συναρτήσεις

Στο MATLAB, οι λειτουργίες επεξεργασίας βίντεο κατατάσσονται σε δύο βασικές κατηγορίες: τις ενσωματωμένες συναρτήσεις του MATLAB και τις συναρτήσεις της εργαλειοθήκης Computer Vision System Toolbox (CVST). Οι βασικές συναρτήσεις (BM) περιλαμβάνουν εργαλεία για αρχικές μαθηματικές πράξεις σε πίνακες και γραφικές απεικονίσεις. Αντίθετα, η CVST προσφέρει προηγμένους αλγορίθμους και εφαρμογές για την ανάπτυξη και προσομοίωση συστημάτων όρασης υπολογιστή, καλύπτοντας εξειδικευμένες διεργασίες όπως ανάλυση αντικειμένων, ανίχνευση και παρακολούθηση κίνησης.

Ορισμένες λειτουργίες, όπως η εισαγωγή, εξαγωγή και αναπαραγωγή βίντεο, υπάρχουν και στις δύο κατηγορίες. Ανάλογα με τις απαιτήσεις μιας εφαρμογής, μπορεί να χρησιμοποιούνται συναρτήσεις τόσο από το βασικό MATLAB όσο και από την CVST. Σε αυτό το βιβλίο, όπου κρίνεται αναγκαίο, διευκρινίζεται η προέλευση κάθε συνάρτησης μέσω παραδειγμάτων. Οι δυνατότητες του MATLAB στην επεξεργασία βίντεο αναλύονται μέσα από συγκεκριμένες περιπτώσεις χρήσης στο παρόν κεφάλαιο. Το περιεχόμενο βασίζεται στην έκδοση MATLAB 2018b, ωστόσο οι περισσότερες συναρτήσεις είναι συμβατές με εκδόσεις από το 2015 και μεταγενέστερες, χωρίς σημαντικές αλλαγές. Το MATLAB υποστηρίζει διάφορες μορφές βίντεο, όπως MPEG-1, Windows Media Video, MPEG-4 με κωδικοποίηση H.264, Apple QuickTime Movie, 3GPP και Ogg Theora. Τα περισσότερα αρχεία βίντεο που χρησιμοποιούνται στα παραδείγματα είναι ήδη ενσωματωμένα στο MATLAB και δεν απαιτούνται από τον χρήστη.

Σε αυτό το κεφάλαιο, οι βασικές συναρτήσεις (BM) που χρησιμοποιούνται ταξινομούνται σε πέντε διαφορετικές κατηγορίες: Θεμελιώδεις Λειτουργίες της Γλώσσας, Μαθηματικά, Γραφικά, Εισαγωγή και Ανάλυση Δεδομένων, καθώς και Σενάρια και Συναρ-

τήσεις Προγραμματισμού. Παρακάτω παρουσιάζεται μια λίστα με τις συναρτήσεις αυτές, μαζί με τη δομή τους και μια σύντομη περιγραφή για καθεμία.

Θεμελιώδεις Λειτουργίες της Γλώσσας

- **clc:** Καθαρισμός του παραθύρου εντολών

Γραφικά

- **figure:** Δημιουργία παραθύρου για γραφήματα
- **frame2im:** Μετατροπή καρέ βίντεο σε δεδομένα εικόνας
- **getFrame:** Λήψη στιγμιότυπου από γραφικά ή αξονικό παράθυρο ως καρέ βίντεο
- **im2frame:** Μετατροπή εικόνας σε καρέ βίντεο
- **plot:** Σχεδίαση διδιάστατης γραφικής παράστασης
- **subplot:** Δημιουργία πολλαπλών γραφημάτων σε ένα παράθυρο
- **title:** Προσθήκη τίτλου σε γράφημα
- **xlabel, ylabel:** Ορισμός ετικετών για τους άξονες X και Y

Εισαγωγή και Ανάλυση Δεδομένων

- **clear:** Αφαίρεση μεταβλητών από τη μνήμη του MATLAB
- **hasFrame:** Έλεγχος αν υπάρχει διαθέσιμο καρέ για ανάγνωση
- **mmfileinfo:** Ανάκτηση πληροφοριών για αρχείο πολυμέσων
- **readFrame:** Ανάγνωση καρέ από αρχείο βίντεο
- **VideoReader:** Εισαγωγή αρχείων βίντεο στο MATLAB
- **VideoWriter:** Δημιουργία και αποθήκευση αρχείων βίντεο

Σενάρια και Συναρτήσεις Προγραμματισμού

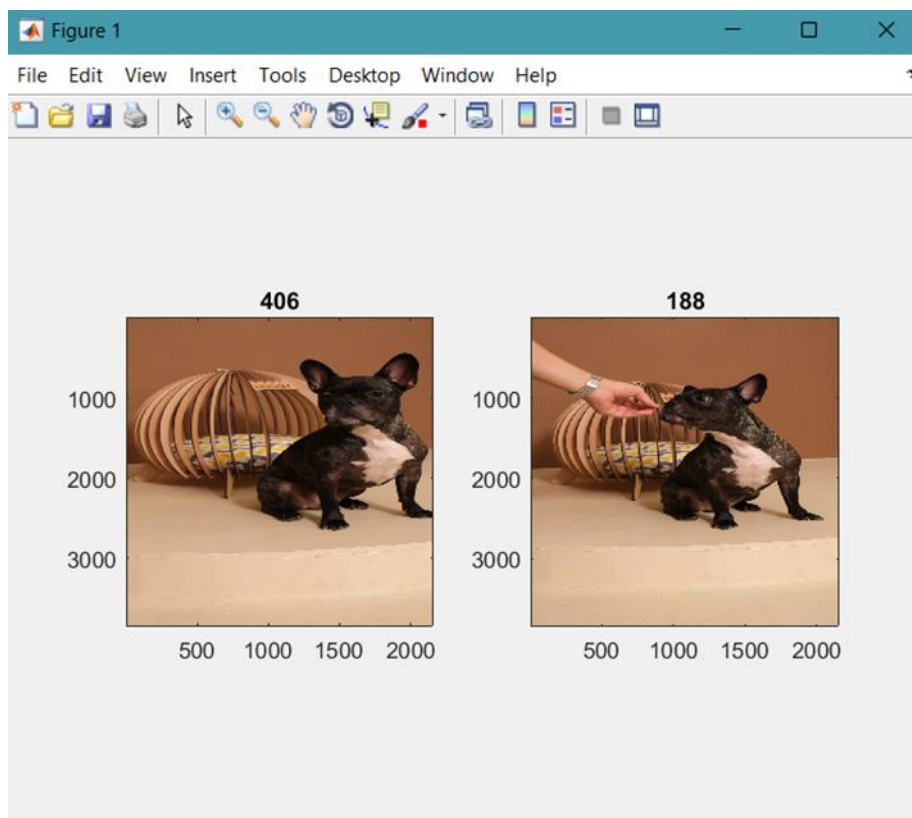
- **for ... end:** Εκτέλεση ενός συνόλου εντολών για προκαθορισμένο αριθμό επαναλήψεων
- **load:** Φόρτωση δεδομένων από αρχείο στο περιβάλλον εργασίας
- **pause:** Διακοπή της εκτέλεσης του MATLAB προσωρινά
- **while ... end:** Εκτέλεση εντολών όσο ισχύει μια συνθήκη

Προχωρημένες Τεχνικές Ανάπτυξης

- **release:** Αποδέσμευση πόρων από αντικείμενα συστήματος
- **step:** Εκτέλεση αλγορίθμου σε αντικείμενο συστήματος (System Object)

Παράδειγμα προς κατανόηση: Γράψε ένα πρόγραμμα που να διαβάζει και να παίζει ένα βίντεο.

```
fv = 'dog.avi';  
fa = 'RockDrums-44p1-stereo-11secs.mp3';  
imshow(fv);  
ia = mmfileinfo(fa); ia.Audio  
iv = mmfileinfo(fv); iv.Video  
vr = VideoReader(fv);  
d = vr.Duration  
h = vr.Height
```



Εικόνα 36: Αποτέλεσμα Matlab

Μια εναλλακτική προσέγγιση για την αναπαραγωγή βίντεο είναι η χρήση της μεθόδου `readFrame`, η οποία επιτρέπει την ανάγνωση κάθε καρέ ξεχωριστά και την προβολή του μέσω βρόχου. Κάθε καρέ που επιστρέφεται (`vf`) αποθηκεύεται ως τρισδιάστατος πίνακας στην περίπτωση έγχρωμης εικόνας. Η διαδικασία επαναλαμβάνεται ανάλογα με το συνολικό πλήθος των καρέ που πρόκειται να εμφανιστούν. Ο συνολικός αριθμός των καρέ ενός βίντεο μπορεί να προσδιοριστεί μέσω της ιδιότητας `NumberOfFrames` του αντικειμένου βίντεο. Η αναπαραγωγή πραγματοποιείται προβάλλοντας κάθε καρέ ως εικόνα μέσα σε ένα παράθυρο γραφικών (`figure window`). Η καθυστέρηση μεταξύ των

διαδοχικών εικόνων καθορίζεται από τον ρυθμό καρέ (frame rate) και ισούται με το αντίστροφο της τιμής του. Για την εμφάνιση του αριθμού καρέ που προβάλλεται κάθε στιγμή, μπορεί να χρησιμοποιηθεί ένας μετρητής, ο οποίος θα εμφανίζει την τιμή στον τίτλο του παραθύρου. Αυτή η μέθοδος είναι χρήσιμη όταν η αναπαραγωγή του βίντεο πρέπει να ξεκινήσει από συγκεκριμένο χρονικό σημείο αντί από την αρχή. Αυτό επιτυγχάνεται ρυθμίζοντας την ιδιότητα CurrentTime κατά τη δημιουργία του αντικειμένου βίντεο. Όταν η αναπαραγωγή δεν ξεκινά από την αρχή, το πλήθος των καρέ που θα προβληθούν δεν είναι γνωστό εκ των προτέρων. Σε αυτή την περίπτωση, ο βρόχος while είναι πιο κατάλληλος από τον for, καθώς επιτρέπει τον έλεγχο διαθεσιμότητας νέων καρέ μέσω της μεθόδου hasFrame.

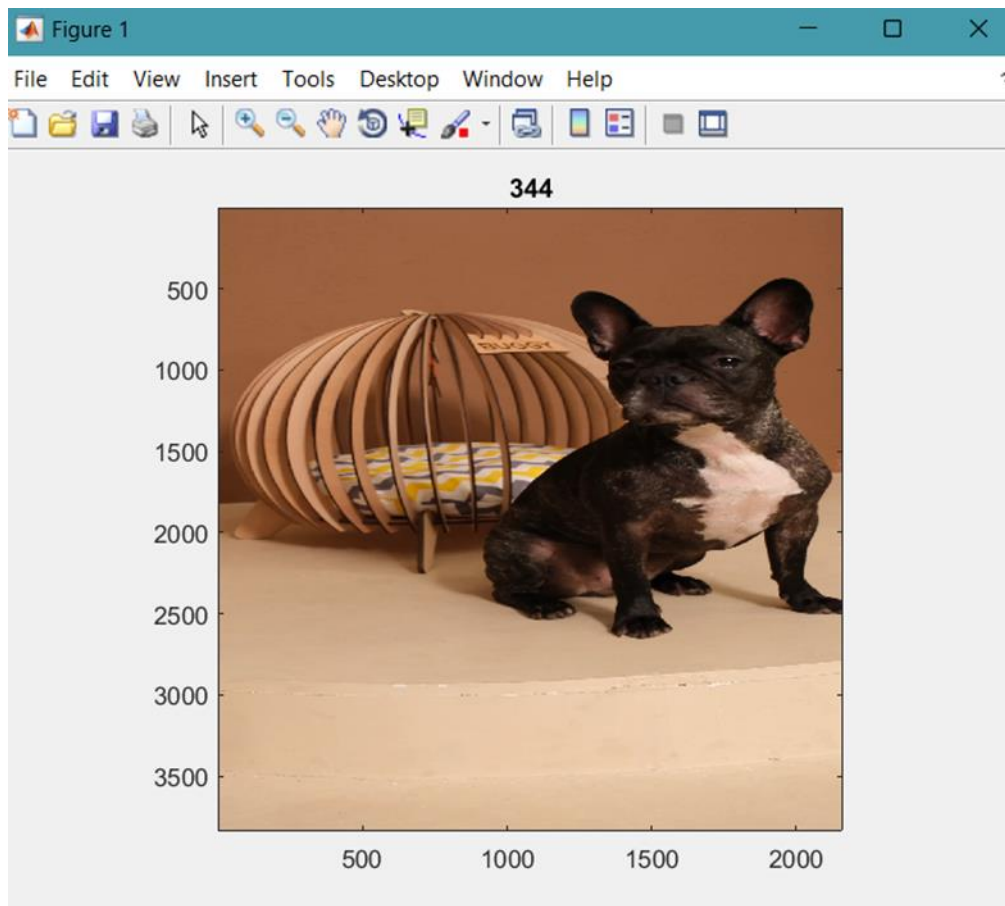
Το ακόλουθο παράδειγμα δείχνει δύο τρόπους προβολής ενός βίντεο:

1. Ολόκληρο το βίντεο με όλα τα καρέ του.
2. Το βίντεο με εκκίνηση από χρονικό σημείο που απέχει 2,5 δευτερόλεπτα από την αρχή, εμφανίζοντας παράλληλα το πλήθος των καρέ που έχουν αναπαραχθεί.

Παράδειγμα προς επίλυση: Δημιουργήστε ένα πρόγραμμα που αναπαράγει ένα βίντεο δύο φορές.

```
vr = VideoReader('dog.mp4');
w = vr.Width;
h = vr.Height;
r = 2;
m = struct('cdata', zeros(h,w,3, 'uint8'), 'color-map', []);
k = 1;
while hasFrame(vr)
    m(k).cdata = readFrame(vr);
    k = k+1;
end
hf = figure;
x = 150; y = 200;
set(hf, 'position', [x y w h]);
movie(hf, m, r, vr.FrameRate);
```

Για περισσότερη εξάσκηση ανατρέξτε στο **Παράρτημα Α και πιο συγκεκριμένα στο Lab_6. [σελ.165]**



Εικόνα 37: Αποτέλεσμα Matlab

6.2 Εντοπισμός κίνησης και ανίχνευση αντικειμένων στο βίντεο

Το παρόν άρθρο παρουσιάζει μια μέθοδο για την ανίχνευση και παρακολούθηση αντικειμένων σε μια ζώνη επιτήρησης. Εξετάζεται η τυπική προσέγγιση που βασίζεται στην ανάλυση εικονοστοιχείων. Η κάμερα (webcam) χρησιμοποιείται ως αισθητήρας για την ανίχνευση και παρακολούθηση αντικειμένων μέσα στην επιτηρούμενη περιοχή. Η ανίχνευση ακμών, που αποτελεί μέθοδο τμηματοποίησης εικόνας, χρησιμοποιείται για την εξαγωγή ακριβών πληροφοριών σχετικά με τις πραγματικές ακμές ενός βίντεο σε πραγματικό χρόνο. Ο αλγόριθμος διαχωρισμού φόντου διασφαλίζει την ξεκάθαρη διαφοροποίηση μεταξύ του προσκηνίου και του φόντου.

Η προ επεξεργασία του βίντεο περιλαμβάνει διαδικασίες όπως ο διαχωρισμός καρέ, η κατωφλίωση, οι δυαδικές λειτουργίες, η εξίσωση ιστογράμματος και η ανίχνευση ακμών, ιδιαίτερα σε βίντεο κυκλοφορίας, με στόχο την παρακολούθηση και αναγνώριση πολλαπλών αντικειμένων. Επιπλέον, ένας βηματικός κινητήρας μπορεί να αξιοποιηθεί

για την περιστροφή της κάμερας σε οποιαδήποτε κατεύθυνση, επιτρέποντας την αποτελεσματική παρακολούθηση και αναγνώριση αντικειμένων στην επιτηρούμενη περιοχή. Ο μετασχηματισμός Contourlet εφαρμόζεται για την εξαγωγή χαρακτηριστικών, διευκολύνοντας την αναγνώριση αντικειμένων σε μια περιοχή επιτήρησης. Παράλληλα, η αντιστοίχιση προτύπων παίζει καθοριστικό ρόλο στην αναγνώριση διαφορετικών αντικειμένων μέσα σε ένα βίντεο.

Η παρακολούθηση και αναγνώριση κινούμενων αντικειμένων, ως μια καινοτόμος τεχνολογία που βασίζεται στην επεξεργασία εικόνας και βίντεο, αποτελεί έναν από τους βασικούς τομείς έρευνας σε εφαρμογές όπως η υπολογιστική όραση, η ρομποτική και η επεξεργασία δεδομένων βίντεο. Αυτή η τεχνολογία έχει ευρεία εφαρμογή σε τομείς όπως η επεξεργασία βίντεο, η συμπίεση βίντεο, η αναγνώριση στόχων, η έξυπνη παρακολούθηση, η ανάκτηση βίντεο, η αλληλεπίδραση ανθρώπου-υπολογιστή και η βιοϊατρική, μεταξύ άλλων. Η παρακολούθηση κινούμενων αντικειμένων περιλαμβάνει τον εντοπισμό της παρουσίας τους σε ένα βίντεο, τον βασικό καθορισμό της θέσης του στόχου και την αναγνώρισή του. Η ακρίβεια στην ανίχνευση των αντικειμένων επηρεάζει σημαντικά τα επόμενα στάδια παρακολούθησης και αναγνώρισης. Η εφαρμογή του MATLAB στην επεξεργασία εικόνας και βίντεο μπορεί να βελτιώσει σε μεγάλο βαθμό την αποδοτικότητα της παρακολούθησης των κινούμενων αντικειμένων. Αυτό το άρθρο αναλύει την εφαρμογή ενός αλγορίθμου ανίχνευσης κινούμενων αντικειμένων με χρήση MATLAB, με παράδειγμα τον αλγόριθμο διαφοράς καρέ. Ο αλγόριθμος χρησιμοποιείται για την ανίχνευση κινούμενων αντικειμένων σε βίντεο και περιλαμβάνει διαδικασίες όπως η απόκτηση βίντεο, η αλλαγή τύπου βίντεο, η μετατροπή σε επίπεδα γκρι, το φιλτράρισμα βίντεο, η εξαγωγή χαρακτηριστικών και ο διαχωρισμός των καρέ.

6.3 Παρακολούθηση κινούμενων αντικειμένων

Η ακολουθία βίντεο αποτελείται από μια σειρά από καρέ που διατηρούν μια συγκεκριμένη χρονική συνέχεια. Η ανίχνευση των κινούμενων αντικειμένων σε βίντεο πραγματοποιείται με την εξαγωγή διαδοχικών ακολουθιών καρέ από την αρχική ακολουθία, ακολουθώντας έναν καθορισμένο κύκλο ή χρονική περίοδο. Αυτή η διαδικασία επιτρέπει την ανίχνευση της κίνησης των αντικειμένων από το ένα καρέ στο επόμενο. Όταν εξετάζουμε την ανίχνευση κινούμενων αντικειμένων σε στατικές εικόνες, η παρακολούθηση των αντικειμένων βασίζεται κυρίως στα χαρακτηριστικά της κίνησης τους μέσα στο βίντεο. Αυτό σημαίνει ότι χρησιμοποιούμε τη χρονική συνέχεια των καρέ για να εντοπί-

σουμε την κίνηση. Η χρονική συνέχεια είναι η διαφορά που παρατηρείται μεταξύ ενός κινούμενου αντικειμένου σε ένα καρέ και ενός αντικειμένου που βρίσκεται σε κατάσταση ακινησίας σε ένα άλλο καρέ. Η αλληλουχία αυτών των καρέ μας βοηθά να εντοπίσουμε την κίνηση και να παρακολουθήσουμε τη θέση του αντικειμένου σε κάθε χρονική στιγμή.

Οι μέθοδοι που χρησιμοποιούνται συνήθως για την ανίχνευση κινούμενων αντικειμένων μέσω ανάλυσης ακολουθιών βίντεο περιγράφονται συνοπτικά παρακάτω. Η ανάλυση αυτή στηρίζεται στην αρχή ότι δύο ή περισσότερα καρέ που αποκτώνται σε διαφορετικούς χρόνους περιέχουν πληροφορίες σχετικά με την κίνηση που συμβαίνει μεταξύ του συστήματος εικόνας (όπως η κάμερα) και της σκηνής. Όταν η κάμερα καταγράφει την ίδια σκηνή σε διαδοχικά καρέ, τα καρέ αυτά εμφανίζουν διαφορές που οφείλονται στην κίνηση των αντικειμένων μέσα στη σκηνή ή στην κίνηση της ίδιας της κάμερας. Έτσι, μέσω της ανάλυσης αυτών των καρέ, μπορούμε να εξάγουμε σημαντικές πληροφορίες για τις κινήσεις που συμβαίνουν. Οι τεχνικές που χρησιμοποιούνται για την ανίχνευση κινούμενων αντικειμένων συνήθως αναλύουν την αλλαγή μεταξύ των καρέ με διάφορους τρόπους, προκειμένου να κατανοήσουμε την κίνηση που υπάρχει μέσα στη σκηνή.

Οι κύριες μέθοδοι ανάλυσης ακολουθιών βίντεο είναι οι εξής:

- **Μέθοδος Ροής Οπτικής**
- **Μέθοδος Διαφοράς Φόντου**
- **Μέθοδος Διαφοράς Γειτονικών Καρέ**

Αυτές οι μέθοδοι παρέχουν διάφορους τρόπους ανίχνευσης και παρακολούθησης της κίνησης, επιτρέποντας την ανίχνευση κινούμενων αντικειμένων σε δυναμικές σκηνές.

6.3.1 Μέθοδοι ροής οπτικής

Η μέθοδος οπτικής ροής, που χρησιμοποιείται συχνά για την ανίχνευση κίνησης σε ακολουθίες βίντεο, εκτιμά την φαινομενική κίνηση των αντικειμένων αναλύοντας τις αλλαγές στην ένταση των εικονοστοιχείων μεταξύ διαδοχικών καρέ. Βασίζεται στην υπόθεση ότι η φωτεινότητα των εικονοστοιχείων παραμένει σταθερή με την πάροδο του χρόνου, επιτρέποντας τον υπολογισμό των διανυσμάτων κίνησης που περιγράφουν την κίνηση των εικονοστοιχείων στην εικόνα. Ωστόσο, αυτή η μέθοδος αντιμετωπίζει αρκετές προκλήσεις.

Ένα βασικό πρόβλημα είναι η ευαισθησία της σε θόρυβο, καθώς ακόμη και μικρές διαταραχές στην εικόνα μπορούν να επηρεάσουν σημαντικά την ακρίβεια της εκτίμησης

κίνησης. Επιπλέον, η μέθοδος απαιτεί σύνθετους υπολογισμούς, ιδιαίτερα όταν υπάρχουν μεγάλες μετατοπίσεις, γεγονός που καθιστά δύσκολη την επίτευξη απόδοσης σε πραγματικό χρόνο. Στην πράξη, ο υπολογισμός των διανυσμάτων κίνησης μπορεί να είναι υπολογιστικά δαπανηρός και χρονοβόρος, περιορίζοντας τη χρησιμότητά του σε περιβάλλοντα που απαιτούν γρήγορους χρόνους επεξεργασίας.

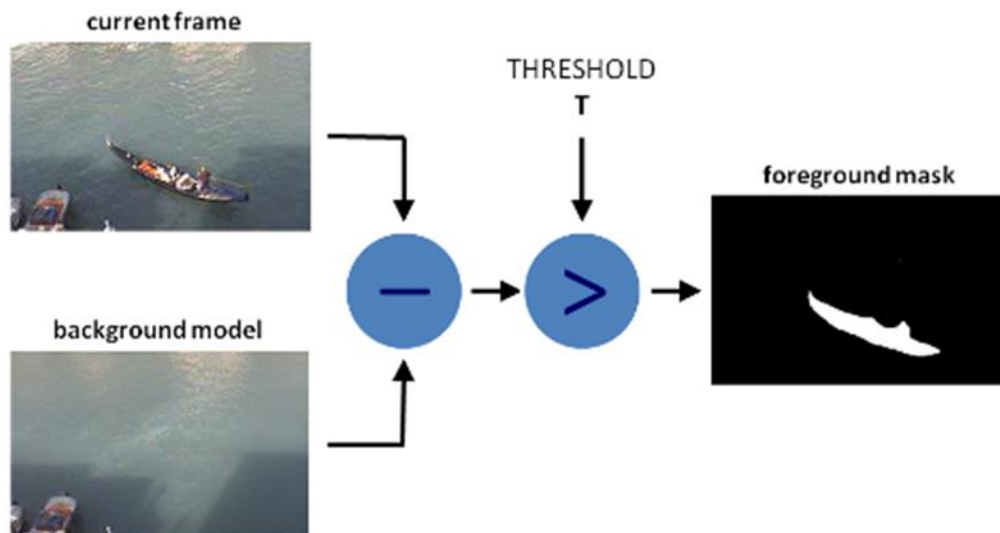
Επιπλέον, η οπτική ροή είναι συχνά λιγότερο αποτελεσματική σε σενάρια που περιλαμβάνουν αποκρύψεις, δραστικές αλλαγές στον φωτισμό ή μεγάλες κινήσεις, καθώς αυτοί οι παράγοντες μπορεί να διαταράξουν τη συνέχεια που υποθέτει η μέθοδος. Παρά τις προόδους που έχουν γίνει για τη βελτίωση της ακρίβειας και της αποδοτικότητας των αλγορίθμων οπτικής ροής, όπως μέσω ανάλυσης πολλαπλών επιλύσεων ή τεχνικών βελτίωσης, η μέθοδος εξακολουθεί να αντιμετωπίζει προβλήματα με τη διατήρηση της απόδοσης σε δυναμικά ή πολύπλοκα βίντεο.

6.3.2 Μέθοδος διαφοράς φόντου

Η μέθοδος διαφοράς φόντου χρησιμοποιείται για την ανίχνευση της περιοχής κίνησης μέσω της σύγκρισης του τρέχοντος καρέ με το μοντέλο του φόντου. Στην προσέγγιση αυτή, η εικόνα χωρίζεται σε προσκήνιο και φόντο. Ο αλγόριθμος δημιουργεί ένα μοντέλο φόντου και στη συνέχεια συγκρίνει κάθε εικονοστοιχείο του τρέχοντος καρέ με αυτό του φόντου, pixel προς pixel. Τα εικονοστοιχεία που συμφωνούν με το μοντέλο του φόντου κατατάσσονται ως φόντο, ενώ τα υπόλοιπα χαρακτηρίζονται ως προσκήνιο.

Αυτή η μέθοδος έχει χαμηλή υπολογιστική πολυπλοκότητα και χρησιμοποιείται συχνά σε εφαρμογές παρακολούθησης κινούμενων αντικειμένων σε σταθερά περιβάλλοντα. Ωστόσο, παρουσιάζει αρκετές προκλήσεις. Με την πάροδο του χρόνου, το μοντέλο του φόντου καθίσταται ευαίσθητο σε εξωτερικές μεταβολές, όπως αλλαγές στον φωτισμό ή στις συνθήκες του περιβάλλοντος. Αυτό μπορεί να οδηγήσει σε σφάλματα στην ανίχνευση, με την εμφάνιση ψευδών αντικειμένων ή την αδυναμία σωστής ανίχνευσης της κίνησης. Επομένως, απαιτείται ανανέωση του μοντέλου φόντου για να παραμείνει αξιόπιστο σε πραγματικούς χρόνους.

Η μέθοδος αυτή δυσκολεύεται να διαχειριστεί περιπτώσεις με σημαντικές μεταβολές του φόντου, όπως μεγάλες διαφοροποιήσεις στο φωτισμό ή κινήσεις της κάμερας. Επίσης, δεν είναι κατάλληλη για σενάρια που περιλαμβάνουν τακτικές αλλαγές στη σκηνή, καθώς οι αλλαγές αυτές ενδέχεται να προκαλέσουν σφάλματα στην ανίχνευση.



Εικόνα 38: Παράδειγμα εικόνας διαφοράς φόντου. [29]

6.3.3 Μέθοδος διαφοράς γειτονικών καρέ

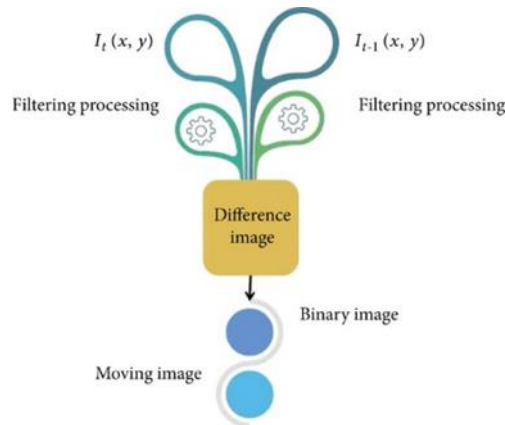
Η μέθοδος διαφοράς γειτονικών καρέ είναι μια απλή και αποτελεσματική τεχνική για την ανίχνευση της κίνησης, στην οποία τα κινούμενα αντικείμενα εξάγονται με βάση τις διαφορές μεταξύ δύο ή τριών συνεχόμενων καρέ από ένα βίντεο. Αυτή η μέθοδος θεωρείται πολύ άμεση, καθώς εντοπίζει γρήγορα τις περιοχές που παρουσιάζουν αλλαγές από το ένα καρέ στο άλλο. Είναι ιδιαίτερα χρήσιμη για την ανίχνευση των αντικειμένων που κινούνται μέσα στο βίντεο, καθώς επικεντρώνεται στις σχετικές κινήσεις των εικονοστοιχείων (pixels) από το ένα καρέ στο επόμενο.

Η βασική δύναμη αυτής της μεθόδου είναι η ικανότητά της να ανιχνεύει γρήγορα και σταθερά τις περιοχές αλλαγής, καθώς το χρονικό διάστημα μεταξύ των καρέ είναι πολύ μικρό. Αυτό σημαίνει ότι οι αλλαγές στο φωτισμό και άλλες παρεμβολές που μπορεί να επηρεάσουν την εικόνα έχουν περιορισμένη επίδραση στην ανίχνευση, καθιστώντας τη μέθοδο ιδιαίτερα αποτελεσματική και αξιόπιστη σε τέτοιες συνθήκες.

Επιπλέον, η μέθοδος της διαφοράς καρέ προσαρμόζεται καλά σε περιβάλλοντα με έντονες διακυμάνσεις ή ταχύτατες αλλαγές, καθώς μπορεί εύκολα να εντοπίσει εκείνα τα pixels που προκαλούν σαφείς αλλαγές στην εικόνα όταν το αντικείμενο κινείται. Αυτό την καθιστά χρήσιμη σε καταστάσεις όπου η κίνηση είναι ευδιάκριτη και έντονη.

Ωστόσο, η μέθοδος αυτή παρουσιάζει αδυναμίες όταν οι αλλαγές στα pixels είναι πολύ μικρές ή σχεδόν αμελητέες. Σε τέτοιες περιπτώσεις, η μέθοδος δυσκολεύεται να ανιχνεύσει την κίνηση, καθώς τα χαρακτηριστικά της δεν μπορούν να διακρίνουν εύκολα τις ελάχιστες διαφορές μεταξύ των καρέ. Επομένως, η μέθοδος είναι πιο αποτελεσματική

κή σε περιβάλλοντα με απλό φόντο και περιορισμένη περιβαλλοντική παραμόρφωση, καθώς και σε καταστάσεις όπου οι κινήσεις είναι ξεκάθαρες και εμφανείς.



Εικόνα 39: Μέθοδος διαφοράς γειτονικών καρέ. [30]

6.4 Μηχανισμός ανίχνευσης ακμών

Η ανίχνευση ακμών είναι μια κρίσιμη τεχνική στην επεξεργασία εικόνας και βίντεο που εστιάζει στην αναγνώριση περιοχών της εικόνας όπου παρατηρούνται απότομες αλλαγές στην ένταση των pixel. Αυτές οι περιοχές, που αντιστοιχούν συνήθως σε περιγράμματα αντικειμένων, είναι καθοριστικές για την ανάλυση εικόνας, καθώς επιτρέπουν την απομόνωση και αναγνώριση σημαντικών χαρακτηριστικών.

Για την απόκτηση εικόνας μέσω MATLAB, πρώτα πρέπει να δημιουργηθεί ένα αντικείμενο εισόδου βίντεο, το οποίο αναπαριστά τη σύνδεση μεταξύ του MATLAB και της συσκευής απόκτησης εικόνας. Η ανάλυση βίντεο συνήθως απαιτεί την εφαρμογή αλγορίθμων επεξεργασίας βίντεο για την προετοιμασία της εικόνας για περαιτέρω ανάλυση. Για τη συγκεκριμένη εργασία, το βίντεο αναλύεται κατά την προβολή του. Για την ανάλυση έγχρωμων καρέ, είναι απαραίτητο να γίνει πρώτα η τμηματοποίηση. Η ανίχνευση ακμών είναι μία από τις πιο κοινές μεθόδους τμηματοποίησης εικόνας στην ανίχνευση αντικειμένων. Δεδομένου ότι οι ακμές περιέχουν κάποιες από τις πιο χρήσιμες πληροφορίες στην εικόνα, μπορούν να χρησιμοποιηθούν για την εξαγωγή των περιγραμμάτων κάθε διαφορετικού αντικειμένου σε μια εικόνα. Σε εικόνα σε κλίμακα του γκρι, μια ακμή μπορεί να οριστεί ως η τοπική ασυνέχεια στις τιμές έντασης (τιμές γκριζου) που υπερβαίνει ένα καθορισμένο κατώφλι. Για έγχρωμα καρέ, μπορούν να εφαρμοστούν παρόμοιες μέθοδοι ανίχνευσης ακμών στο συστατικό έντασης της εικόνας. Έχουν αναπτυχθεί πολλοί αλγόριθμοι ανίχνευσης ακμών, οι οποίοι περιλαμβάνουν τους Sobel, Prewitt,

Roberts, Laplacian of a Gaussian, Zero Crossing και Canny. Ο αλγόριθμος Canny χρησιμοποιείται συχνά, καθώς είναι λιγότερο ευαίσθητος στους θορύβους σε σύγκριση με άλλες μεθόδους ανίχνευσης ακμών. Ο αλγόριθμος αυτός μπορεί να συνοψιστεί ως εξής:

- **Μέθοδος ανίχνευσης ακμών:** Η ανίχνευση ακμών είναι κρίσιμη για την ανάλυση εικόνας και βίντεο. Χρησιμοποιείται για να εντοπιστούν οι γραμμές, οι άκρες και οι αλλαγές στην ένταση του φωτός που αντιστοιχούν σε όρια μεταξύ διαφορετικών αντικειμένων ή περιοχών μέσα σε μια εικόνα.
- **Τμηματοποίηση εικόνας:** Η ανίχνευση ακμών είναι επίσης σημαντική για την τμηματοποίηση της εικόνας. Με την τμηματοποίηση, η εικόνα χωρίζεται σε διάφορα τμήματα ή περιοχές με βάση τα χαρακτηριστικά τους
- **Προβλήματα και Περιορισμοί:** Η ανίχνευση ακμών συχνά αντιμετωπίζει προβλήματα θορύβου. Θορυβώδεις περιοχές στην εικόνα μπορεί να προκαλέσουν ψευδείς ακμές και να μειώσουν την ακρίβεια του αλγορίθμου.
- **Οφέλη του Canny αλγορίθμου:** Ο Canny αλγόριθμος είναι ιδιαίτερα δημοφιλής στην ανίχνευση ακμών λόγω της ικανότητάς του να μειώνει τον θόρυβο και να εντοπίζει με ακρίβεια τις ακμές σε εικόνες. Το χαρακτηριστικό του ότι είναι λιγότερο ευαίσθητος στους θορύβους τον καθιστά πιο αποδοτικό σε πραγματικές συνθήκες, όπου οι εικόνες συχνά περιέχουν θόρυβο.

Για το λόγο αυτό, αλγόριθμοι όπως ο Canny θεωρούνται συχνά προτιμότεροι, καθώς είναι πιο ανθεκτικοί στον θόρυβο, διατηρώντας παράλληλα την ακρίβεια στην ανίχνευση ακμών. Ωστόσο, ακόμη και με αυτές τις εξελιγμένες μεθόδους, η ανίχνευση ακμών μπορεί να δυσκολεύεται σε δυναμικά ή σύνθετα περιβάλλοντα, όπου οι συνθήκες φωτισμού αλλάζουν γρήγορα ή όπου τα αντικείμενα επικαλύπτονται. Για το λόγο αυτό, συχνά χρησιμοποιούνται επιπλέον τεχνικές προ επεξεργασίας, όπως η μείωση θορύβου ή η εξομάλυνση, προκειμένου να ενισχυθεί η ανθεκτικότητα της ανίχνευσης ακμών. Συνολικά, η ανίχνευση ακμών είναι ένα θεμελιώδες βήμα σε πολλές εφαρμογές επεξεργασίας εικόνας και βίντεο, με πολλές τεχνικές και αλγόριθμους να έχουν αναπτυχθεί για την αύξηση της ακρίβειας και της αποτελεσματικότητας της διαδικασίας. Ωστόσο, πρέπει να ληφθούν υπόψη οι περιορισμοί της, όπως η ευαισθησία στους θορύβους και η ανάγκη για αναβάθμιση των μεθόδων όταν η ανάλυση απαιτεί υψηλή ακρίβεια.

Η υλοποίηση του ανιχνευτή ακμών Canny στο MATLAB γίνεται μέσω της συνάρτησης `edge`. Η απόδοση αυτού του ανιχνευτή ελέγχεται κυρίως από δύο παραμέτρους: το πλάτος της μάσκας Gaussian που χρησιμοποιείται στη φάση επεξεργασίας της εικόνας και

τις τιμές των κατωφλίων υστέρησης. Όσο μεγαλύτερο είναι το πλάτος της μάσκας Gaussian, τόσο λιγότερη είναι η ευαισθησία του ανιχνευτή στον θόρυβο της εικόνας. Ωστόσο, αυτό μπορεί να προκαλέσει μεγαλύτερη απόσταση μεταξύ των πραγματικών ακμών της εικόνας και των ακμών που εντοπίζει ο αλγόριθμος. Το πλάτος της μάσκας Gaussian σχετίζεται με την τυπική απόκλιση που χρησιμοποιείται στην εξίσωση του φίλτρου Gaussian. Η προεπιλεγμένη τιμή αυτής της τυπικής απόκλισης στη συνάρτηση edge είναι 1.

Αναλυτικά, ο ανιχνευτής Canny χρησιμοποιεί τη μάσκα Gaussian για να "στρογγυλέψει" την εικόνα και να μειώσει τον θόρυβο, προετοιμάζοντας την για την ανίχνευση ακμών. Η υστέρηση βοηθά να προσδιοριστούν ποια ακμές είναι πιο αξιόπιστες με βάση τις παραμέτρους κατωφλίου, δίνοντας έτσι μια πιο ακριβή ανίχνευση. Αυτές οι παράμετροι πρέπει να προσαρμοστούν με βάση τις ανάγκες της εκάστοτε εικόνας και του περιβάλλοντος ανάλυσης.

Το MATLAB αποθηκεύει εικόνες ως δύο ή τρεις διαστάσεων πίνακες. Στην περίπτωση των εικόνων RGB, ο πίνακας έχει τρεις διαστάσεις, με κάθε στοιχείο να αντιστοιχεί σε ένα pixel της εικόνας. Αυτή η δομή πίνακα επιτρέπει την άμεση διαχείριση και επεξεργασία εικόνας μέσω του MATLAB, κάτι που είναι ιδιαίτερα χρήσιμο για την ανάλυση και επεξεργασία βίντεο. Ένα από τα σημαντικά πλεονεκτήματα του MATLAB είναι ότι παρέχει την εργαλειοθήκη απόκτησης βίντεο, η οποία επιτρέπει τη σύνδεση με συσκευές λήψης βίντεο, όπως κάμερες. Η εργαλειοθήκη αυτή συνδυάζεται εύκολα με την εργαλειοθήκη επεξεργασίας βίντεο του MATLAB, δίνοντας στους χρήστες τη δυνατότητα να αποκτούν, να επεξεργάζονται και να αναλύουν βίντεο με αποτελεσματικότητα και ευχέρεια. Αυτός ο συνδυασμός εργαλειοθηκών διευκολύνει την επεξεργασία βίντεο και την ανάλυση εικόνας, επιτρέποντας στους χρήστες να επικεντρωθούν στη διαδικασία ανίχνευσης και επεξεργασίας δεδομένων βίντεο χωρίς να χρειάζεται να ανησυχούν για τον τρόπο αλληλεπίδρασης των εργαλείων μεταξύ τους. Οι δυνατότητες αυτές ενισχύουν την αποδοτικότητα και την ακρίβεια στις εφαρμογές βίντεο και εικόνας του MATLAB, καθιστώντας το μια ισχυρή πλατφόρμα για επαγγελματίες και ερευνητές.

6.5 Ανίχνευση κίνησης

Η ανίχνευση κίνησης σε διαδοχικές εικόνες αφορά την αναγνώριση των κινούμενων αντικειμένων σε μια σκηνή, κάτι που είναι θεμελιώδες για συστήματα παρακολούθησης και επιτήρησης. Σε συστήματα βίντεο, η ανίχνευση κίνησης αναφέρεται στην ικανότητα

του συστήματος να εντοπίζει κάθε μεταβολή στην εικόνα που υποδηλώνει κίνηση, επιτρέποντας την καταγραφή σημαντικών γεγονότων. Αυτή η διαδικασία συνήθως βασίζεται σε αλγορίθμους λογισμικού που παρακολουθούν συνεχώς το περιβάλλον και, όταν ανιχνεύσουν κίνηση, στέλνουν ένα σήμα στην κάμερα για να ξεκινήσει την καταγραφή. Αυτή η διαδικασία είναι επίσης γνωστή ως ανίχνευση δραστηριότητας.

Συχνά, οι σύγχρονοι αλγόριθμοι ανίχνευσης κίνησης είναι ικανοί να αναλύσουν και τον τύπο της κίνησης. Αυτό μπορεί να βοηθήσει να κατανοηθεί αν η κίνηση είναι αναμενόμενη και αν ανταποκρίνεται σε καθορισμένα πρότυπα. Για παράδειγμα, σε συστήματα επιτήρησης εξωτερικών χώρων, οποιαδήποτε μικρή κίνηση που ξεπερνά ένα όριο ανοχής μπορεί να καταγραφεί ως «κίνημα», εξασφαλίζοντας ότι δεν διαφεύγουν σημαντικά γεγονότα. Έτσι, ακόμη και μικρές και φαινομενικά αδιάφορες κινήσεις μπορούν να εντοπιστούν και να καταγραφούν για μελλοντική ανάλυση.

6.5.1 Μεθοδολογία

Η αξιόπιστη παρακολούθηση των κινούμενων ανθρώπων είναι απαραίτητη για την εκτίμηση κίνησης, την επιτήρηση μέσω βίντεο και τις διασυνδέσεις ανθρώπου-υπολογιστή. Το παρόν έργο προτείνει μια νέα προσέγγιση για την παρακολούθηση κίνησης ανθρώπων, η οποία συνδυάζει τεχνικές βασισμένες σε εικόνα και μοντέλο. Στην περίπτωση του μονοκαναλικού βίντεο, η επεξεργασία γίνεται τόσο σε επίπεδο pixel όσο και σε επίπεδο αντικειμένου. Στο επίπεδο pixel, συγκρίνεται η τιμή κάθε pixel και δημιουργείται ένα νέο αντικείμενο, το οποίο ονομάζεται "προσκήνιο", όταν η νέα εικόνα συγκρίνεται με την τυπική εικόνα φόντου.

Η υπολογιστική όραση είναι ο τομέας που ασχολείται με τη διασύνδεση των υπολογιστών με το περιβάλλον τους μέσω οπτικών μέσων. Ένα κλασικό πρόβλημα σε αυτόν τον τομέα είναι η αναγνώριση αντικειμένων, που αφορά την ταυτοποίηση ενός γνωστού αντικειμένου σε μια εικόνα. Αν και οι άνθρωποι μπορούμε να αναγνωρίζουμε νέα αντικείμενα αμέσως και εύκολα, αυτό αποδεικνύεται δύσκολο και υπολογιστικά απαιτητικό για τους υπολογιστές. Για την επιτυχημένη αναγνώριση αντικειμένων, απαιτείται μια πιο συγκεκριμένη προσέγγιση στο πρόβλημα, κάτι που καθιστά τη διαδικασία ακόμη πιο περίπλοκη.

Αναλύοντας το, η αναγνώριση αντικειμένων μέσω υπολογιστικής όρασης εμπλέκει πολύπλοκες διαδικασίες, που περιλαμβάνουν την ικανότητα των υπολογιστών να αναγνωρίζουν και να κατανοούν εικόνες με τρόπους που είναι φυσικοί για τον άνθρωπο. Οι τε-

χνικές παρακολούθησης κίνησης και ανάλυσης αντικειμένων είναι καθοριστικές για την ανάπτυξη συστημάτων παρακολούθησης και ασφαλείας, καθώς και για εφαρμογές στον τομέα της ρομποτικής και άλλων τεχνολογικών και βιομηχανικών τομέων.

6.5.2 Σύστημα καταγραφής βίντεο

Αυτή η ενότητα περιλαμβάνει μια κάμερα που καταγράφει ένα καρέ ως την αναφορά ή το υπόβαθρο για την παρακολούθηση κίνησης ή εξαγωγή προσκηνίου. Το καρέ αυτό χρησιμεύει ως σημείο αναφοράς για να συγκριθούν μετέπειτα καρέ κατά την επεξεργασία βίντεο για να εντοπιστούν τυχόν κινήσεις. Αρχικά, το σύστημα καταγράφει μια εικόνα του τυπικού φόντου και στη συνέχεια καταγράφει βίντεο με τον αριθμό καρέ που καθορίζεται από τον χρήστη. Το βίντεο επεξεργάζεται με σκοπό την ανίχνευση κίνησης συγκρίνοντας τα καρέ του βίντεο με το στιγμιότυπο του φόντου για να προσδιοριστεί αν υπάρχουν αλλαγές.

Αναλυτικά, το σύστημα καταγραφής βίντεο χρησιμοποιεί την κάμερα για να δημιουργήσει μια στατική αναφορά του φόντου, η οποία αποτελεί τη βάση για την ανίχνευση τυχόν κίνησης στην ακολουθία των καρέ του βίντεο. Αυτή η διαδικασία είναι θεμελιώδης για την παρακολούθηση και ανάλυση κίνησης, καθώς επιτρέπει τον εντοπισμό αντικειμένων ή δραστηριοτήτων που ξεχωρίζουν από το περιβάλλον τους. Ο αριθμός των καρέ και η χρονική διάρκεια της καταγραφής καθορίζονται από τις ανάγκες του συστήματος, ενώ η διαδικασία επεξεργασίας συμβαδίζει με την ανάγκη για γρήγορη ανίχνευση και απόκριση σε πραγματικό χρόνο.

6.5.3 Αλγόριθμος εξαγωγής προσκηνίου

Ο αλγόριθμος εξαγωγής προσκηνίου έχει ως στόχο να αναγνωρίσει τα αντικείμενα στο προσκηνίο και να αφαιρέσει τα στοιχεία του φόντου από την εικόνα. Για να γίνει αυτό, πρώτα πρέπει να δημιουργηθεί μια "συνθετική" εικόνα φόντου, η οποία θα χρησιμεύσει ως βάση αναφοράς για την εκτέλεση της διαδικασίας διαχωρισμού εικόνας. Οι εικόνες που περιλαμβάνονται στην επεξεργασία είναι συνήθως συνεχόμενα καρέ από ένα βίντεο, και αυτές απαιτούν προετοιμασία για να εντοπιστούν οι αλλαγές μεταξύ φόντου και προσκηνίου.

Ο αλγόριθμος περιλαμβάνει τα παρακάτω στάδια:

- **Δημιουργία Μοντέλου Φόντου**
- **Ανίχνευση Κίνησης ή Αντικειμένων**

- **Αφαίρεση του Φόντου**
- **Σύγκριση Εικόνας**

Δημιουργία Μοντέλου Φόντου: Η εικόνα του φόντου που θεωρείται ως πρότυπο ή το πρώτο καρέ του βίντεο χρησιμοποιείται ως η βασική εικόνα αναφοράς για τη σύγκριση με τα υπόλοιπα καρέ του βίντεο. Σκοπός αυτής της σύγκρισης είναι να ανιχνευθούν τυχόν κινήσεις, είτε αυτές αφορούν αντικείμενα, είτε άλλες μεταβολές στην εικόνα.

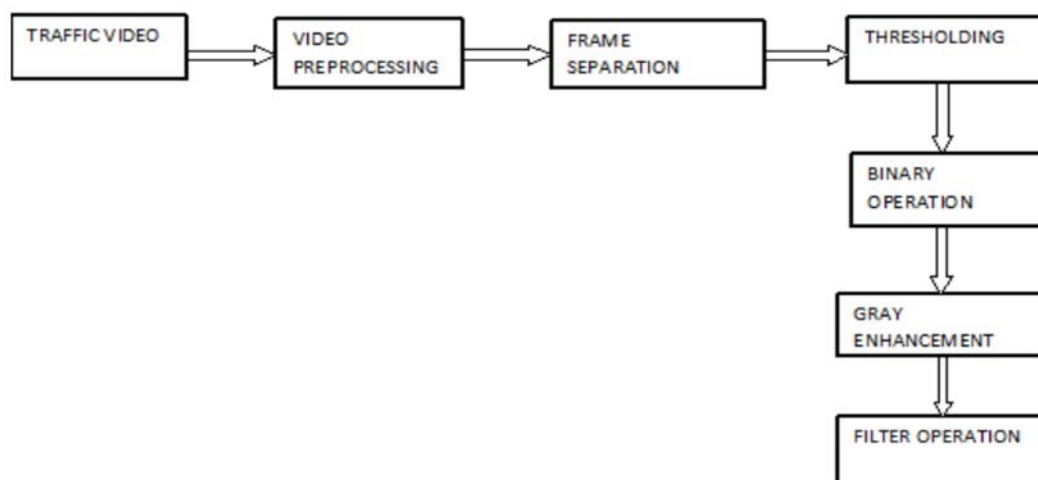
Αφαίρεση του Φόντου: Εδώ πραγματοποιείται η αφαίρεση του φόντου από το προβαλλόμενο για να εντοπιστούν οποιεσδήποτε επαρκείς αλλαγές στις τιμές των pixel. Εάν η αλλαγή στις τιμές των pixel είναι αρκετά μεγάλη ώστε να καταγραφεί ως κίνηση, τότε αυτή η διαφορά μεταξύ των δύο καρέ καταγράφεται ως κίνηση στην εικόνα.

Σύγκριση εικόνας: Η διαφορά εικόνας που κατασκευάζεται όταν υπολογίζεται η διαφορά μεταξύ της αναφοράς εικόνας και των τρεχόντων καρέ του βίντεο, πρώτα μετατρέπεται σε δυαδική μορφή και στη συνέχεια ο συνολικός αριθμός των "1" θα προσδιορίσει αν η αλλαγή είναι επαρκής για να δηλωθεί ως κίνηση. Εάν η αλλαγή είναι πολύ μικρή, δεν θεωρείται σημαντική κίνηση, αλλά αν η αλλαγή είναι αρκετά μεγάλη, τότε μπορεί να δηλωθεί ως ανιχνευμένη κίνηση.

Ανίχνευση Κίνησης ή Αντικειμένων: Εδώ, όπως μπορεί να καταλάβει κανείς, η εικόνα δείχνει αν υπάρχει σημαντική κίνηση ή όχι. Εάν αρχικά ανιχνευθεί κάποια κίνηση και στη συνέχεια, όταν η διαφορά μεταξύ των καρέ είναι ίδια για αρκετά διαδοχικά καρέ, αυτό υποδεικνύει ότι το κινούμενο αντικείμενο έχει σταματήσει. Σε αυτή την περίπτωση, το συγκεκριμένο καρέ θεωρείται ως το νέο καρέ αναφοράς για την ανίχνευση οποιασδήποτε κίνησης.

6.6 Εφαρμογή

Η παρακολούθηση και αναγνώριση αντικειμένων είναι ένας πολύ ενδιαφέρον τομέας, ο οποίος χρησιμοποιείται κυρίως στη ρομποτική και τις στρατιωτικές εφαρμογές. Μια αποδοτική μετασχηματισμένη μέθοδος εφαρμόζεται για να βοηθήσει στην παρακολούθηση και αναγνώριση των αντικειμένων στην περιοχή παρακολούθησης. Το βίντεο αποκτάται σε μορφή Audio Video Interleave (AVI) και υποβάλλεται σε επεξεργασία. Αυτή η διαδικασία είναι κρίσιμη για εφαρμογές όπου απαιτείται η ανίχνευση και παρακολούθηση αντικειμένων σε πραγματικό χρόνο, με την τεχνολογία AVI να προσφέρει έναν ευέλικτο και ευρέως υποστηριζόμενο τρόπο για τη διαχείριση και ανάλυση βίντεο.



Εικόνα 40: Η Εικόνα 35 δείχνει το διάγραμμα ροής για την παρακολούθηση αντικειμένων και τις βασικές λειτουργίες που εμπλέκονται κατά την παρακολούθηση ενός αντικειμένου.

Η παρακολούθηση και αναγνώριση αντικειμένων είναι ένας τομέας που έχει μεγάλη σημασία, κυρίως στις εφαρμογές ρομποτικής και στρατιωτικών συστημάτων. Για την παρακολούθηση και αναγνώριση αντικειμένων σε επιτήρηση, χρησιμοποιείται ένας αποτελεσματικός μετασχηματισμός που βοηθά στον εντοπισμό και την παρακολούθηση του αντικειμένου στην περιοχή επιτήρησης. Στη διαδικασία αυτή, η μορφή βίντεο Audio Video Interleave (AVI) χρησιμοποιείται για την επεξεργασία και παρακολούθηση.

Για την παρακολούθηση σε πραγματικό χρόνο, πρέπει να δημιουργηθεί επικοινωνία μεταξύ του MATLAB και της κάμερας (webcam) ώστε να ληφθούν τα καρέ του βίντεο. Αρχικά, το πλαίσιο αναφοράς πρέπει να αποκτηθεί και να συγκριθεί με το τρέχον καρέ για να προσδιοριστεί η αλλαγή, δηλαδή η κίνηση του αντικειμένου. Ένα σημαντικό στοιχείο για την ακριβή παρακολούθηση του αντικειμένου είναι το κεντροειδές του, δηλαδή το σημείο που αντιστοιχεί στο κέντρο του αντικειμένου.

Ο μετασχηματισμός Contourlet είναι μία τεχνική που εφαρμόζεται στην ανάλυση εικόνας και προσφέρει γρήγορη υλοποίηση με βάση την αποσύνθεση πυραμίδας Laplacian, ακολουθούμενη από την εφαρμογή κατευθυντικών φίλτρων σε κάθε υποζώνη περασμένης ζώνης. Η χρήση αυτών των φίλτρων επιτρέπει την αποτελεσματική επεξεργασία εικόνων με ήπιες αλλαγές και καθαρά περιγράμματα. Το αποτέλεσμα αυτής της διαδικασίας είναι ένα σύστημα ανάλυσης πολλαπλών επιπέδων με κατευθυντική δομή, το οποίο βοηθά στην ακριβή αναγνώριση των περιγραμμάτων και στην παρακολούθηση των α-

ντικειμένων. Ο μετασχηματισμός Contourlet, που βασίζεται σε φίλτρα διπλής δομής, επιτρέπει την επεξεργασία εικόνας με τον πιο αποδοτικό τρόπο, ξεπερνώντας τα προβλήματα που σχετίζονται με άλλες μεθόδους, όπως οι μετασχηματισμοί wavelet και curvelet.

6.6.1 Επεξεργασία βίντεο

Μετά την απόκτηση του βίντεο, είναι απαραίτητο να γίνει κάποια επεξεργασία προκειμένου να βελτιωθούν τα χαρακτηριστικά της εικόνας και να προετοιμαστεί για περαιτέρω ανάλυση. Ο στόχος της προ επεξεργασίας εικόνας είναι να βελτιώσει την ποιότητα της εικόνας εξαλείφοντας τον θόρυβο και επισημαίνοντας τις σημαντικές δομές της εικόνας, έτσι ώστε να γίνει πιο ευανάγνωστη και να προσφέρει καλύτερη ανάλυση. Αυτό βοηθά στην ανάδειξη των βασικών χαρακτηριστικών και των λεπτομερειών που ενδέχεται να επηρεάζουν την ανάλυση.

Η διαδικασία διαίρεσης του βίντεο, γνωστή ως video segmentation, στοχεύει στο να χωρίσει το βίντεο σε τμήματα ή shots. Αυτά τα τμήματα αντιπροσωπεύουν μια συνεχιζόμενη σειρά από καρέ που προέρχονται από την ίδια κάμερα. Αυτή η διαίρεση επιτρέπει στο σύστημα να κατανοήσει το περιεχόμενο του βίντεο πιο εύκολα, κάνοντας τις επόμενες φάσεις ανάλυσης πιο αποτελεσματικές.

Εν ολίγοις, η προ επεξεργασία του βίντεο είναι κρίσιμη για να επιτευχθεί μια πιο ακριβής και αποδοτική ανάλυση, ειδικά σε περιβάλλοντα όπου η ακρίβεια και η αναγνώριση αντικειμένων ή συμβάντων είναι θεμελιώδους σημασίας.

6.6.2 Διαχωρισμός καρέ

Η διαδικασία του διαχωρισμού των καρέ από την ροή βίντεο (frame separation) είναι ένα απαραίτητο βήμα για την εκτέλεση διαφόρων επεξεργασιών, όπως η κατωφλική ανάλυση (thresholding) και οι δυαδικές λειτουργίες (binary operations). Όταν διαχωριστούν τα καρέ, είναι δυνατό να εφαρμοστούν οι κατάλληλοι αλγόριθμοι ή επεξεργασίες σε κάθε καρέ ξεχωριστά, προκειμένου να αναλυθεί με μεγαλύτερη ακρίβεια το περιεχόμενο του βίντεο.

Μετά την ολοκλήρωση των επεξεργασιών σε κάθε καρέ, τα καρέ πρέπει να επανενωθούν για να δημιουργήσουν μια ενιαία ροή βίντεο. Αυτή η διαδικασία είναι σημαντική για να αναλυθεί το τελικό αποτέλεσμα του βίντεο μετά την εκτέλεση όλων των απαιτούμενων επεξεργασιών. Στην ουσία, ο διαχωρισμός και η συνένωση των καρέ επιτρέπει

την εφαρμογή πολύπλοκων αλγορίθμων επεξεργασίας εικόνας σε κάθε μεμονωμένο καρέ, ενώ ταυτόχρονα διατηρείται η συνοχή του βίντεο στο σύνολό του.

Σε γενικές γραμμές, αυτή η διαδικασία είναι κρίσιμη για την επίτευξη αποτελεσματικών αναλύσεων βίντεο, ειδικά όταν απαιτείται να εντοπιστούν σημαντικά χαρακτηριστικά ή να αναλυθούν κινήσεις σε μια σειρά καρέ.

6.6.3 Παρακολούθηση εικόνων

Στην παρακολούθηση εικόνων από την εθνική οδό, χρησιμοποιούμε την καθορισμένη εικόνα της οδικής σκηνής ως περιοχή επεξεργασίας της εικόνας, η οποία εντοπίζει την περιοχή του δρόμου. Επομένως, η εικόνα του φόντου παραμένει απλή. Για να ανταποκριθεί στις απαιτήσεις της πραγματικής απόδοσης σε πραγματικό χρόνο και να μειώσει την ανάγκη για υπολογιστική προσπάθεια στις αλγορίθμους, το παρόν έργο υιοθετεί έναν στατιστικό αλγόριθμο για την κατασκευή του φόντου, βασισμένο στον μέσο όρο της ακολουθίας, ο οποίος προσφέρει υψηλή ταχύτητα, λιγότερη υπολογιστική προσπάθεια και απλότητα στην εφαρμογή του.

Αναλύοντας τη διαδικασία, ο στόχος είναι να εντοπιστεί η περιοχή του δρόμου με την απλή χρήση εικόνας φόντου, γεγονός που επιτρέπει την ταχύτερη και αποτελεσματικότερη ανάλυση του κινούμενου αντικειμένου χωρίς να απαιτείται υπερβολικός υπολογιστικός φόρτος. Η τεχνική αυτή, η οποία βασίζεται σε έναν στατιστικό μοντέλο φόντου, χρησιμοποιεί έναν αλγόριθμο υπολογισμού του μέσου όρου των ακολουθιών για να καθορίσει το φόντο, εξασφαλίζοντας ότι οι υπολογισμοί γίνονται γρήγορα και με λιγότερη υπολογιστική προσπάθεια.

Αυτός ο αλγόριθμος επιτρέπει τη βελτιστοποίηση της διαδικασίας παρακολούθησης και την αναγνώριση του κινούμενου αντικειμένου σε πραγματικό χρόνο, γεγονός που είναι κρίσιμο για εφαρμογές παρακολούθησης σε πραγματικές συνθήκες, όπως οι δρόμοι ή άλλες εξωτερικές περιοχές με συνεχείς και γρήγορες αλλαγές.

7 Συμπεράσματα

Η παρούσα πτυχιακή εργασία είχε ως στόχο τη δημιουργία ενός αποθετηρίου εργαστηριακών ασκήσεων για την επεξεργασία εικόνας και βίντεο, προκειμένου να υποστηρίξει τη διδασκαλία και να βοηθήσει τους φοιτητές να κατανοήσουν καλύτερα τις σχετικές

έννοιες. Μέσα από τη χρήση του MATLAB, σχεδιάστηκαν ασκήσεις που ξεκινούν από βασικές έννοιες, όπως η εξομάλυνση εικόνας και τα ιστογράμματα, και προχωρούν σε πιο σύνθετες τεχνικές, όπως η συμπίεση εικόνας με τη μέθοδο Huffman.

Κατά τη διάρκεια της εργασίας, αντιμετωπίστηκαν προκλήσεις, όπως η επιλογή των κατάλληλων ασκήσεων και η διαμόρφωσή τους ώστε να είναι κατανοητές και αποτελεσματικές για τους φοιτητές. Η χρήση πραγματικών δεδομένων από αποθετήρια, όπως αυτό του University of Waterloo, συνέβαλε στην ενίσχυση της πρακτικής διάστασης των ασκήσεων. Εάν ξεκινούσα ξανά την πτυχιακή, θα ενσωμάτωνα περισσότερα διαδραστικά στοιχεία, όπως MATLAB Live Scripts, και μεγαλύτερη ποικιλία δεδομένων, ώστε οι φοιτητές να έχουν περισσότερες δυνατότητες πειραματισμού.

Η εργασία αυτή μου προσέφερε πολύτιμες γνώσεις στην επεξεργασία εικόνας, καθώς και εμπειρία στη δημιουργία εκπαιδευτικού υλικού. Μελλοντικά, το αποθετήριο μπορεί να επεκταθεί με ασκήσεις που αφορούν σύγχρονες τεχνικές, όπως η χρήση τεχνητής νοημοσύνης στην ανάλυση εικόνας και η real-time επεξεργασία βίντεο. Επιπλέον, θα μπορούσε να εξελιχθεί σε μια διαδικτυακή εκπαιδευτική πλατφόρμα, όπου οι φοιτητές θα μπορούν να πειραματίζονται και να λαμβάνουν ανατροφοδότηση αυτόματα.

Βιβλιογραφία

ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Αντζουλάτος. 2017. “ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ MATLAB/SIMULINK.” 2017.
- [2] Καρέτσας. 2020. “ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ.”
- [3] Μιχαλίδης. 2021. “ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΙΜΩΝ ΤΜΗΜΑ ΦΥΣΙΚΗΣ ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΚΑΙ ΜΕΙΩΣΗ ΘΟΡΥΒΟΥ : ΥΛΟΠΟΙΗΣΗ ΣΕ VHDL ΚΑΙ MATLAB.”
- [4] Μπέκιος, and Μπαλέση. 2018. “" ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΑΥΤΟΜΑΤΗΣ ΠΡΟΣΑΡΜΟΓΗΣ ΠΟΙΟΤΗΤΑΣ ΠΟΛΥΜΕΣΩΝ ".”
- [5] Μπεληγιάννης, Αρ. 2006. “Α ΤΕΙΜΕΣΟΛ ΟΓΓΙΟ Υ.”
- [6] Ντάλλης, Παύλος. 2021. “ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΑΝΑΛΥΣΗ ΙΑΤΡΙΚΗΣ ΕΙΚΟΝΑΣ ΔΕΡΜΑΤΟΣΚΟΠΙΟΥ.”
- [7] Παπαδάκος , and Έλληνας. 2017. “ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ. ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε. ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ Επεξεργασία εικόνας και αναγνώριση προτύπων με το Raspberry Pi.”
- [8] Ζιώγος, Εισηγήτρια Δρ, Βελώνη, and Καθηγήτρια Αθηνά. 2017. “Επεξεργασία εικόνας με τη χρήση του MATLAB.”
- [9] Παναγιωτίδης. 2006. “Μ. Παραγωγή Πολυμεσικού Υλικού Και Ηλεκτρονικών Δραστηριοτήτων Για Γλωσσική εκπαίδευση. .” 2006.
- [10] Φλουρή. 2003. “ Σκέψεις για την αναζήτηση ενός πλαισίου επιμόρφωσης και διαβίου μάθησης των εκπαιδευτικών στην κοινωνία της γνώσης. .” Ατραπός. 2003.
- [11] Τάντη. 2017. “ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ

ΕΛΛΑΔΟΣ ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΜΕΣΩΝ ΜΑΖΙΚΗΣ ΕΝΗΜΕΡΩΣΗΣ ΠΤΥΧΙΑΚΗ
ΕΡΓΑΣΙΑ ΑΝΑΠΤΥΞΗ ΔΙΑΔΡΑΣΤΙΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΔΙΑΧΕΙΡΗΣΗ
ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΜΕΣΩ ΔΙΑΔΙΚΤΥΟΥ ΜΕ
ΠΡΟΣΑΝΑΤΟΛΙΣΜΟ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ.”

ΞΕΝΟΓΛΩΣΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- [12] Song, Myung-Sin. 2010. “Entropy Encoding in Wavelet Image Compression.”
- [13] ebooks. ebooks.edu.gr. διαδραστικά βιβλία. [Ηλεκτρονικό]
2017 www.ebooks.edu.gr.
- [14] esa. European space agency- esa eduspace. esa.int. [Ηλεκτρονικό] 2017
www.esa.int/special/eduspace
- [15] Bigge, and Morris L. 1990. “Θεωρίες μάθησης για εκπαιδευτικούς. .” Πατάκη.
1990.
- [16] Gonzalez, Rafael, and Richard Eugene Woods. 2008. “ Ψηφιακή Επεξεργασία Ει-
κόνας.” Τζιόλας. 2008.
- [17] Song, S. H., Marco Antonelli, Tony W. K. Fung, Brandon D. Armstrong, Amy
Chong, Albert Lo, and Bertram E. Shi. 2019. “Developing and Assessing MATLAB
Exercises for Active Concept Learning.” IEEE Transactions on Education 62 (1):
2–10. <https://doi.org/10.1109/te.2018.2811406>.
- [18] Sabbouh, Marwan, Jeff Higginson, Salim Semy, and Danny Gagne. 2007. “Script-
ing Language .” 2007.
- [19] Venkatesan, R, and A Ganesh. 2024. “Real Time Implementation on Moving Object
Tracking and Recognition Using Matlab.”
- [20] Hayes, and Monson. 2012. “ Schaums Outline of Digital Processing. .” 2012.
- [21] Jayaraman. 2009. “ Digital Image Processing. .” 2009.
- [22] S. Bistarelli, "Semirings for Soft Constraint Solving and Programming", Springer-
Verlag, LNCS-2962, pp. 1–20, 2004.
- [23] <https://deixto.com> - Σύστημα αποδελτίωσης ιστού DEiXTo. [25/12/2021]

- [24] Google. 2025a. ‘1929 Picture Bartlane.’ 2025. <https://www.google.com/search?q=1929+picture+Bartlane&tbm=isch>.
- [25] Google. 2025b. “ ‘Ranger 7 First Picture.’ ” 2025. <https://www.google.com/search?q=Ranger+7+first+picture>.
- [26] “Grey Scale Image with 0 and 1 - Google Search.” 2024. Google.com. 2024. [https://www.google.com/search?q=grey+scale+image+with+0+and+1](https://www.google.com/search?q=grey+scale+image+with+0+and+1&tbm=isch&chips=q:grey+scale+image+with+0+and+1).
- [27] Google. 2025c. “1921 Picture Bartlane.” 2025. <https://www.google.com/search?q=1921picture+Bartlane&tbm=isch>.
- [28] Wikipedia. 2020. “Binary Image.” Wikipedia. December 27, 2020. https://en.wikipedia.org/wiki/Binary_image.
- [29] “OpenCV: How to Use Background Subtraction Methods.” n.d. Docs.opencv.org. https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html
- [30] ResearchGate. 2025. “ ‘Flowchart of Background Difference Method.’ .” 2025. https://www.researchgate.net/figure/Flowchart-of-background-difference-method_fig3_362845384
- [31] “Algorithm of Huffman Code - Tpoint Tech.” 2025. Www.tpointtech.com. 2025. <https://www.tpointtech.com/huffman-coding-algorithm>

Παράρτημα Α

Εργαστήριο 1 – Lab 1

Διανύσματα, Πίνακες και Βασικά μονοδιάστατα (1D) σήματα

Σκοπός εργαστηρίου:

Η εξοικείωση των φοιτητών με αναπαράσταση και πράξεις ανάμεσα στους πίνακες και τα διανύσματα στο περιβάλλον του MATLAB, καθώς και παραγωγή και αναπαράσταση βασικών σημάτων σε 1D. Τέτοια μπορεί να είναι τα ακόλουθα: κρουστική απόκριση, βηματική συνάρτηση, ημιτονοειδή αλλά και εκθετικά σήματα.

Προσέξτε ότι:

Βεβαιωθείτε ότι έχετε δημιουργήσει έναν φάκελο στον υπολογιστή σας με την ονομασία "Επεξεργασία Εικόνας και Βίντεο", που αντιστοιχεί στο μάθημα. Μέσα σε αυτόν τον κύριο φάκελο, δημιουργήστε έναν υποφάκελο με το ονοματεπώνυμό σας. Αυτός ο υποφάκελος θα χρησιμεύει για την οργάνωση και αποθήκευση όλων των αρχείων .m του MATLAB που θα δημιουργήσετε κατά τη διάρκεια του μαθήματος. Η σωστή οργάνωση των αρχείων σας θα διευκολύνει την πρόσβαση και τη διαχείριση των εργασιών σας.

1. Μαθησιακοί Στόχοι

Με την ολοκλήρωση αυτού του εργαστηρίου, ο φοιτητής θα μπορεί να:

- **Κατανοήσει** τη βασική δομή δεδομένων στο MATLAB (πίνακες, διανύσματα, μήτρες) και τις εντολές δημιουργίας τους (π.χ. zeros, ones, eye, randi).

- **Εφαρμόσει** αριθμητικές πράξεις και βασικές λειτουργίες σε πίνακες και διανύσματα, κατανοώντας τη διαφορά μεταξύ στοιχειοβάθμιας και γραμμικής πράξης (π.χ. `.*` vs. `*`, `.^` vs. `^`).
- **Αναπτύξει** δεξιότητες στην πρόσβαση και επεξεργασία στοιχείων πινάκων μέσω indexing και χρήση βρόχων for.
- **Δημιουργήσει** και **διαχειριστεί** γραφικές παραστάσεις (plot, stem) για την αναπαράσταση βασικών συναρτήσεων και σημάτων.
- **Σχεδιάσει** scripts και συναρτήσεις στο MATLAB για την αυτοματοποίηση διαδικασιών, με έμφαση σε δομές ελέγχου (if-else) και user inputs.
- **Εξερευνήσει** τις βασικές έννοιες του συστήματος επεξεργασίας σήματος (π.χ. Δέλτα συνάρτηση, κρουστική απόκριση) και να **εφαρμόσει** τις σχετικές πρακτικές ασκήσεις.

2. Πώς το μαθαίνουν:

- **Θεωρητική Παρουσίαση & Παραδείγματα:** Στην εισαγωγή κάθε ενότητας, παρέχονται θεωρητικές επεξηγήσεις με παραδείγματα που δείχνουν τον τρόπο λειτουργίας κάθε εντολής και δομής δεδομένων.
- **Πρακτική Εφαρμογή:** Οι φοιτητές **υλοποιούν** ασκήσεις στο MATLAB, όπου βήμα-βήμα **αναπτύσσουν** τον κώδικα, **πειραματίζονται** με διάφορες παραμέτρους και **παρατηρούν** τα αποτελέσματα στην οθόνη.
- **Συγκριτική Ανάλυση:** Μέσω ερωτήσεων κατανόησης και συγκρίσεων (π.χ. τι δείχνει η εντολή length έναντι της size), **αναλύουν** τις διαφορές και **κατανοούν** την εφαρμογή κάθε εντολής.
- **Ανάπτυξη Δικτυωμένων Projects:** Οι ασκήσεις **ενθαρρύνουν** την **ανάπτυξη** μικρών projects που **συνδυάζουν** διάφορες τεχνικές (π.χ. δημιουργία πίνακα, επεξεργασία μεταβλητών, δημιουργία γραφικών παραστάσεων) ώστε να δουν πώς **αλληλεπιδρούν** όλα τα στοιχεία σε μια ολοκληρωμένη εφαρμογή.

Τι είναι οι πίνακες στο Matlab;

Πίνακες: Στο MATLAB, οι πίνακες είναι η βασική δομή δεδομένων. Χρησιμοποιούνται για να αποθηκεύουν και να επεξεργάζονται αριθμητικά δεδομένα. Ένας πίνακας μπορεί να είναι:

- **Διανύσματα (1D):** Μια γραμμή ή στήλη (π.χ., `[1 2 3]` ή `[1; 2; 3]`).
- **Μήτρες (2D):** Δύο διαστάσεις (γραμμές $\times$ στήλες).
- **Πολυδιάστατοι πίνακες (3D+):** Για πιο σύνθετα δεδομένα, όπως έγχρωμες εικόνες.

Χρήσεις Πινάκων:

- Αποθήκευση αριθμητικών δεδομένων.

- Επεξεργασία εικόνας.
- Μαθηματικές πράξεις, όπως πρόσθεση, πολλαπλασιασμός.

Βασικές εντολές:

dir: Προβάλλουμε όλα τα στοιχεία που υπάρχουν στον τρέχοντα φάκελο.

help: η οποία μπορεί επίσης να εμπερικλείει το όνομα μιας καθορισμένης οδηγίας ή ενός τελεστή, για τα οποία θέλουμε να παραλάβουμε πληροφορίες ή κάποια υπόδειξη.

help ops: Με την οποία μπορούμε να λάβουμε βοήθεια για όλους τους διαθέσιμους τελεστές.

edit: Η οποία χρησιμοποιείται για να δημιουργήσουμε ένα νέο script αρχείο. Μπορούμε επίσης να προσθέσουμε το όνομα του αρχείου, το οποίο πρέπει να τελειώνει με την κατάληξη .m.

cd ‘διαδρομή φακέλου’: Την οποία εισάγουμε στο command line.

clear all: Με αυτήν διαγράφονται όλες οι μεταβλητές.

clc: Με αυτήν εκκαθαρίζεται η οθόνη.

Άσκηση για εξάσκηση:

Εκτελέστε τις παραπάνω εντολές για να τις κατανοήσετε καλύτερα.

Εκτέλεση βασικών συναρτήσεων, ανάθεση τιμών και μεταβλητές

$x = 7;$

$y = 7 * x;$

$z = 5 * x + 6 * y;$

Η χρήση του ελληνικού ερωτηματικού στο τέλος των προτάσεων έχει ως συνέπεια το αποτέλεσμα να μην εμφανίζεται, αλλά να αποθηκεύεται στη μνήμη. Το σύμβολο % χρησιμοποιείται για να προσθέτουμε σχόλια σε μία γραμμή.

Extra (+): Ορίστε μερικές μεταβλητές, δώστε τους τιμές και δημιουργήστε εκφράσεις. Μετά, διαγράψτε όλες τις μεταβλητές.

Εργαστηριακό μέρος 1

Αριθμητικές Πράξεις, Γραφικές Παραστάσεις, Πίνακες και Διανύσματα.

3. Δημιουργία Πινάκων

- Δημιουργήστε έναν πίνακα 3×3 με τις εξής τιμές:
 - Πρώτη γραμμή: 1, 2, 3
 - Δεύτερη γραμμή: 4, 5, 6
 - Τρίτη γραμμή: 7, 8, 9

Υπόδειξη: Χρησιμοποιήστε την εντολή: [... ; ...].

- Δημιουργήστε έναν μηδενικό πίνακα 3×3 .

Υπόδειξη: Χρησιμοποιήστε την εντολή **zeros**.

- Δημιουργήστε έναν μοναδιαίο πίνακα 3×3 .

Υπόδειξη: Χρησιμοποιήστε την εντολή **eye**.

- Δημιουργήστε έναν πίνακα 4×5 με τυχαίους ακέραιους αριθμούς από 1 έως 10.

Υπόδειξη: Χρησιμοποιήστε την εντολή **randi**.

Η διαχείριση πινάκων είναι ένα από τα βασικά σημεία στα οποία το MATLAB ξεχωρίζει σε σύγκριση με άλλες ερευνητικές πλατφόρμες. Η δήλωση των πινάκων, όταν είναι γνωστά τα στοιχεία τους, μπορεί να γίνει με τον παρακάτω τρόπο:

```
A = [2 2 4 3; %τυχαίος πίνακας 4x4  
7 9 8 6;  
4 5 3 2;  
9 8 7 6];
```

ΑΠ: Πίνακας 4x4

```
B= [3 1 1 0; %τυχαίος πίνακας 4x4  
    2 6 7 0;  
    7 4 3 9;  
    0 2 7 8];
```

ΑΠ: Πίνακας 4x4

4. Αναφορά σε συγκεκριμένο στοιχείο

- Προσπελάστε το στοιχείο της 2ης γραμμής και 1ης στήλης και αποθηκεύστε το σε μια μεταβλητή A21.

Υπόδειξη: Χρησιμοποιήστε τη σύνταξη A(γραμμή, στήλη).

Για να προσπελάσετε ένα στοιχείο του πίνακα A που βρίσκεται στη δεύτερη γραμμή και πρώτη στήλη, μπορείτε να το κάνετε ως εξής:

```
A21= A(2,1);
```

ΑΠ: 7

- Αλλάξτε την τιμή του στοιχείου της 3ης γραμμής και 2ης στήλης σε 10.

Υπόδειξη: Χρησιμοποιήστε την ακόλουθη εντολή για να αλλάξετε την τιμή: **A(3,2) = 10;**

5. Χρήση Βρόχων για Πρόσβαση σε Όλα τα Στοιχεία

Indexing: Η πρόσβαση σε συγκεκριμένα στοιχεία του πίνακα γίνεται με τη σύνταξη **A(i,j)**, όπου το **i** αντιστοιχεί στη γραμμή και το **j** στη στήλη.

Βρόχοι: Ο διπλός βρόχος **for** επιτρέπει την πρόσβαση και την επεξεργασία όλων των στοιχείων ενός πίνακα.

- Γράψτε έναν διπλό φωλιασμένο βρόχο **for** για να προσπελάσετε όλα τα στοιχεία του πίνακα A και να τα εμφανίσετε στη γραμμή εντολών.

Υπόδειξη: Χρησιμοποιήστε την εντολή **size** για να καθορίσετε τις διαστάσεις του πίνακα.

Όλα τα στοιχεία ενός πίνακα μπορούν να προσπελαστούν χρησιμοποιώντας έναν διπλό φωλιασμένο βρόχο for, όπως φαίνεται παρακάτω:

```
for i=1:size(A,1)
    for j=1:size(A,2)
        A(i,j);
    end
end
```

ΑΠ: i=4 και j=4

- Προσπελάστε τη 2η στήλη του πίνακα A και αποθηκεύστε την σε μια μεταβλητή A_deuteri_stili.

Υπόδειξη: Χρησιμοποιήστε τη σύνταξη **A(:, στήλη)**.

- Άλλαξε το στοιχείο της 3ης γραμμής και της 2ης στήλης σε 10:

Υπόδειξη: Χρησιμοποίησε την εντολή **A(3,2) = 10;**

Το σύμβολο «:» επιτρέπει την πρόσβαση σε όλα τα στοιχεία ενός πίνακα κατά μήκος μιας συγκεκριμένης διάστασης.

```
A_deuteri_stili= A(:,2);
```

ΑΠ: [2;9;5;8]

6. Βασικές Πράξεις Πινάκων

- Πρόσθεση πινάκων

B = [3 1 1 0;

2 6 7 0;

7 4 3 9;

0 2 7 8];

C = A + B; % Πρόσθεση αντίστοιχων στοιχείων

- Εκτελέστε την πρόσθεση δύο πινάκων A και B και αποθηκεύστε το αποτέλεσμα σε έναν νέο πίνακα C.

1. Άσκηση για Εξάσκηση:

x=ones(1,5);

y=zeros(1,5);

a= [3 2 1];

b= [6 5 4];

c= [3 2 1; 6 5 4];

ΑΠ: x= [1 1 1 1 1] , y=[0 0 0 0 0] , a=[3 2 1] , b=[6 5 4] , C=

$\begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \end{bmatrix}$

- Προβάλετε ό,τι έχετε δημιουργήσει.

2. Άσκηση προς επίλυση

Δίνεται ο πίνακας:

A = [2 4 6 8;

1 3 5 7;

9 8 7 6;

4 3 2 1];

- Δημιουργήστε έναν νέο πίνακα B 4×4, όπου κάθε στοιχείο του είναι το διπλάσιο του αντίστοιχου στοιχείου του πίνακα A.

Υπόδειξη: Χρησιμοποιήστε πράξεις πίνακα.

- Αλλάξτε την τιμή του στοιχείου στη 2η γραμμή και 3η στήλη του πίνακα B σε 20.
- Υπολογίστε το άθροισμα όλων των στοιχείων του πίνακα B και αποθηκεύστε το σε μια μεταβλητή `sum_B`.

Υπόδειξη: Χρησιμοποιήστε την εντολή `sum` δύο φορές.

- Εξαγάγετε την 3η γραμμή του πίνακα B και αποθηκεύστε την σε μια μεταβλητή `row3_B`.
- Γράψτε έναν βρόχο `for` για να εμφανίσετε όλα τα στοιχεία του πίνακα B που είναι μεγαλύτερα από 10.

Ερωτήσεις κατανόησης

- Τι εμφανίζουν οι εντολές **`length(x)`** και **`length(y)`**; Αυτές οι εντολές δείχνουν το μήκος των διανυσμάτων. **ΑΠ= 5 και 5**
- Ποιο μέγεθος του πίνακα δίνεται από την εντολή **`size(c)`**; **ΑΠ= 2 3**
- Άλλες έτοιμες συναρτήσεις για τη δημιουργία πινάκων και διανυσμάτων είναι:
- **Zeros:** Δημιουργεί έναν πίνακα ή ένα διάνυσμα γεμάτο μηδενικά.
- **Ones:** Δημιουργεί έναν πίνακα ή ένα διάνυσμα γεμάτο μονάδες.
- **Eye:** Δημιουργεί έναν πίνακα με τη διαγώνιο γεμάτη με μονάδες και τα υπόλοιπα στοιχεία μηδενικά.
- **Rand:** Δημιουργεί έναν πίνακα ή ένα διάνυσμα με τυχαίες τιμές από την ομοιόμορφη κατανομή [0,1].
- **Randn:** Δημιουργεί έναν πίνακα ή ένα διάνυσμα με τυχαίες τιμές από την κανονική κατανομή με μέση τιμή 0 και διασπορά 1.
- **Randi:** χρησιμοποιείται για τη δημιουργία πινάκων με τυχαίες ακέραιες τιμές σε ένα καθορισμένο εύρος.

3. Άσκηση κατανόησης

- Παρακαλώ εκτελέστε τις εντολές που ακολουθούν σε αυτήν τη σειρά.

`Ones(3)`, `zeros(3)`, `magic(3)`, `rand(3)`, `rand(3,3)`, `randi(3,3)`.

- Τι παρατηρείτε; Ποιοι πίνακες έχουν γίνει και πόσοι είναι;

$$\text{Ones}(3) \quad \mathbf{A\Pi} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Zeros}(3) \quad \mathbf{A\Pi} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{Magic}(3) \quad \mathbf{A\Pi} = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

$$\text{Rand}(3) \quad \mathbf{A\Pi} = \begin{bmatrix} 0.8147 & 0.9134 & 0.2785 \\ 0.9058 & 0.6324 & 0.5469 \\ 0.1270 & 0.0975 & 0.9575 \end{bmatrix}$$

$$\text{Rand}(3,3) \quad \mathbf{A\Pi} = \begin{bmatrix} 0.9649 & 0.9572 & 0.1419 \\ 0.1576 & 0.4854 & 0.4218 \\ 0.9706 & 0.8003 & 0.9157 \end{bmatrix}$$

$$\text{Randi}(3,3) \quad \mathbf{A\Pi} = \begin{bmatrix} 3 & 1 & 3 \\ 3 & 3 & 3 \\ 2 & 3 & 3 \end{bmatrix}$$

Η εντολή **randi(3, 2, 2, 1)** παράγει έναν τρισδιάστατο πίνακα διαστάσεων $2 \times 2 \times 1$. Στην ουσία, δημιουργείται ένας δισδιάστατος πίνακας 2×2 με τυχαίους ακέραιους από το διάστημα $[1,3][1,3][1,3]$.

➤ Τι παράγει η εντολή **c=randi(3,2,2,1);**

$$\mathbf{A\Pi} = \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix}$$

Παρατηρήσεις:

➤ Οι πίνακες που δημιουργούνται από τις εντολές **rand** και **randi** διαφέρουν κάθε φορά που εκτελούνται, διότι βασίζονται σε τυχαίους αριθμούς.

- Ο πίνακας magic έχει μαθηματική ιδιαιτερότητα, καθώς οι τιμές του ικανοποιούν τις συνθήκες μαγικού τετραγώνου.

Θυμάμαι ότι:

- Το στοιχείο που βρίσκεται στην πρώτη γραμμή και τη δεύτερη στήλη του πίνακα **c** αναπαρίσταται ως: **c(1,2)**. Στο MATLAB, η αρίθμηση των γραμμών και των στηλών των πινάκων ξεκινά από το 1, σε αντίθεση με άλλες γλώσσες προγραμματισμού που ξεκινούν από το 0.
- **c(2,:)**: Η ανάκτηση μιας ολόκληρης γραμμής ενός πίνακα, όπως η δεύτερη γραμμή, επιτυγχάνεται με τη χρήση της ακόλουθης εντολής.
- **c(:,3)**: Αντίστοιχα, η ανάκτηση της δεύτερης στήλης γίνεται με αυτήν την εντολή.
- Ας υποθέσουμε ότι έχουμε δημιουργήσει τον πίνακα **c = randi(100,5,5,1)**, ο οποίος είναι ένας πίνακας 5x5. Για να εξάγουμε έναν υποπίνακα 2x2 από τον **c**, μπορούμε να χρησιμοποιήσουμε την ακόλουθη διαδικασία:
d=c(1:2,2:3)
- Αν υπάρχει, ο αντίστροφος ενός πίνακα **A** μπορεί να υπολογιστεί χρησιμοποιώντας την έτοιμη συνάρτηση **inv(A)**.
- Ο ανάστροφος ενός πίνακα **A** ορίζεται ως ένας πίνακας που έχει τις στήλες του **A** ως γραμμές και το αντίστροφό του ως στήλες.
- Ο πολλαπλασιασμός πινάκων πραγματοποιείται με τη χρήση του τελεστή ***** στη γλώσσα προγραμματισμού MATLAB, όπως και στα μαθηματικά, δηλαδή **ανά γραμμή και στήλη**.

ΠΡΟΣΟΧΗ

- **A*B**: αν ορίζεται
- **A*B'**: όταν θέλουμε να υπολογίσουμε το εσωτερικό γινόμενο
- **A.*B**: Όταν θέλουμε να πολλαπλασιάσουμε στοιχείο προς στοιχείο
- Το ίδιο ισχύει και όταν θέλουμε να υψώσουμε σε δύναμη .^

Εντολές γραφικών:

- **plot**: Δημιουργία γραμμικής γραφικής παράστασης.
- **stem**: Χρήση για διακριτές τιμές (π.χ. για την απεικόνιση της Δέλτα συνάρτησης).
- **xlabel, ylabel, title, legend**: Εντολές για την επισήμανση των αξόνων, τίτλων και επεξηγήσεων στο γράφημα.
- **grid on**: Ενεργοποίηση του πλέγματος για καλύτερη ανάγνωση της γραφικής παράστασης.
- **hold on/off**: Διαχείριση της εμφάνισης πολλαπλών γραμμών σε ένα γράφημα.

1. Γραφική Παράσταση

Υποθέστε ότι έχουμε το διάστημα $[-3,3]$ και το έχουμε διαχωρίσει σε ίσα τμήματα με βήμα 0.7. Τα σημεία αποθηκεύονται στο διάνυσμα x . Χρησιμοποιούμε την ακόλουθη εντολή :

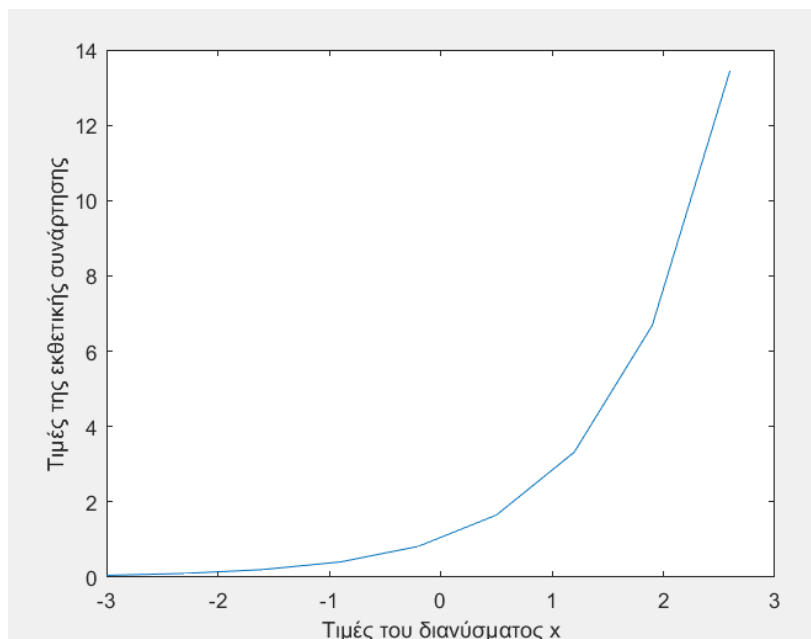
$x = [-3:0.7:3];$

Προχωρώντας, θα υπολογίσουμε την εκθετική συνάρτηση $y = e^x$ για κάθε σημείο στο διάνυσμα x και θα αποθηκεύσουμε τις αντίστοιχες τιμές στο διάνυσμα y .

$y = \exp(x);$

$\text{plot}(x,y); \text{xlabel}(\text{'Τιμές του διανύσματος } x\text{'}); \text{ylabel}(\text{'Τιμές της εκθετικής συνάρτησης'})$

Η εντολή `plot` δημιουργεί τη γραφική παράσταση συνδέοντας τα σημεία που υπολογίζονται από τις τιμές του x και του y . Στην προκειμένη περίπτωση, η εκθετική συνάρτηση e^x αυξάνεται εκθετικά και παρουσιάζει ραγδαία άνοδο στις θετικές τιμές του x .



2. Άσκηση πρακτικής:

- Δημιουργήστε ένα διάνυσμα x που περιέχει τιμές από το -5 έως το 5 με βήμα 0.5 .
- Υπολογίστε τις τιμές της συνάρτησης:
 - a) $y1 = x^2$
 - b) $y2 = \sin(x)$

- Δημιουργήστε σε μια ενιαία γραφική παράσταση (με την εντολή plot) τις συναρτήσεις y_1 και y_2 :
 - a) Χρησιμοποιήστε διαφορετικά χρώματα για κάθε καμπύλη.
 - b) Προσθέστε τίτλο, υπόμνημα (legend) και ετικέτες στους άξονες.

Απάντηση:

1. Χρήση της εντολής: $x = -5:0.5:5;$
2. Χρήση της εντολής: $y_1 = x.^2;$
 $y_2 = \sin(x);$
3. Χρήση της εντολής: **plot(x, y1, 'r-', 'LineWidth', 2);**
hold on;
plot(x, y2, 'b--', 'LineWidth', 2);
hold off;
4. Χρήση της εντολής: **title('Γραφική Παράσταση των $y_1 = x^2$ και $y_2 = \sin(x)$ ');**
xlabel('Τιμές του διανύσματος x');
ylabel('Τιμές των συναρτήσεων');
legend('y_1 = x^2', 'y_2 = sin(x)');
grid on;

3. Άσκηση πρακτικής:

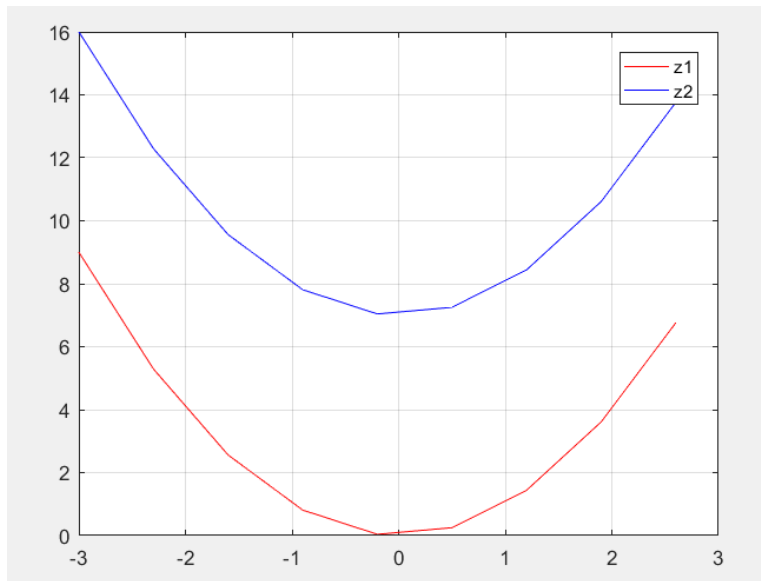
- Ζητείται η δημιουργία γραφικής αναπαράστασης των:

$$Z_1 = x^2$$

$$Z_2 = x^2 + 7$$

- Με τις τιμές x να είναι στο διάστημα $[-7, 7]$ σε και τις δύο περιπτώσεις, δημιουργήστε ένα κοινό γράφημα που παρουσιάζει τα δύο σύνολα δεδομένων.

Χρήσιμη εντολή: **plot(x,z1,'-r',x,z2,'-b');grid on;legend('z1','z2')**



4. Άσκηση πρακτικής

- **Δημιουργία** δύο **ημιτονοειδών** σημάτων με διαφορετικές **παραμέτρους** (π.χ. συχνότητες, πλάτη) και **απεικόνιση** τους στο ίδιο γράφημα.
- **Επεξήγηση:** Δοκίμασε να πειραματιστείς με την αλλαγή των τιμών για να δεις πώς επηρεάζουν την εμφάνιση των σημάτων.

5. 3) Δημιουργία script – συναρτήσεις

Το πρώτο script ζητά από τον χρήστη να εισάγει δύο αριθμούς και συγκρίνει τις τιμές τους. Αν ο πρώτος αριθμός είναι μεγαλύτερος, το πρόγραμμα εκτυπώνει σχετικό μήνυμα. Αν ο δεύτερος είναι μεγαλύτερος, εκτυπώνει άλλο μήνυμα. Αν οι αριθμοί είναι ίσοι, εμφανίζεται μήνυμα ισότητας.

Επεξήγηση όρων

- **Scripts:** Αρχεία με επέκταση .m που περιέχουν εντολές MATLAB και εκτελούνται ως σύνολο.
- **Συναρτήσεις:** Επιτρέπουν την επαναχρησιμοποίηση κώδικα. Ορίζονται με τη λέξη-κλειδί **function** και έχουν καθορισμένες εισόδους και εξόδους.
- **Δομές Ελέγχου:** Οι εντολές **if**, **else**, **elseif** επιτρέπουν τη λήψη αποφάσεων με βάση συνθήκες.

Γράψε ένα script που ζητά από τον χρήστη δύο αριθμούς και εμφανίζει μήνυμα σύγκρισης (ο πρώτος μεγαλύτερος, ο δεύτερος μεγαλύτερος ή ίσοι).

- Για να συνθέσετε ένα script, ένα αρχείο με εκτελέσιμο κώδικα, επινοείτε ένα νέο αρχείο χρησιμοποιώντας την εντολή **'edit'** και αποθηκεύετε το με κατάληξη **.m**.
- Πληκτρολογήστε τον ακόλουθο κώδικα και αποθηκεύστε το script ως **myscript_1.m**.

```

a = input('Δώσε μου τον πρώτο αριθμό');
b = input('Δώσε μου τον δεύτερο αριθμό');
if (a>b)
    sprintf('Ο %d είναι μεγαλύτερος από τον %d',a,b)
else if (b>a)
    sprintf('Ο %d είναι μεγαλύτερος από τον %d',b,a)
else
    sprintf('Ο %d είναι ίσος με τον %d',a,b)
end
end

```

Η συνάρτηση **replace** δέχεται έναν **πίνακα A** ως **είσοδο** και επιστρέφει έναν **πίνακα Y**, όπου:

- Τα θετικά στοιχεία του πίνακα A παραμένουν ως έχουν.
 - Τα αρνητικά στοιχεία του πίνακα A αντικαθίστανται από τις απόλυτες τιμές τους.
- Στη συνέχεια, πληκτρολογήστε και τον επόμενο κώδικα και αποθηκεύστε το script ως **myscript_2.m**.

```

function Y = replace(A)
[Z,N] = size(A);
for i = 1:Z
    for j = 1:N
        if A(i,j) > 0
            Y(i,j) = A(i,j);
        else
            Y(i,j) = -A(i,j);
        end
    end
end

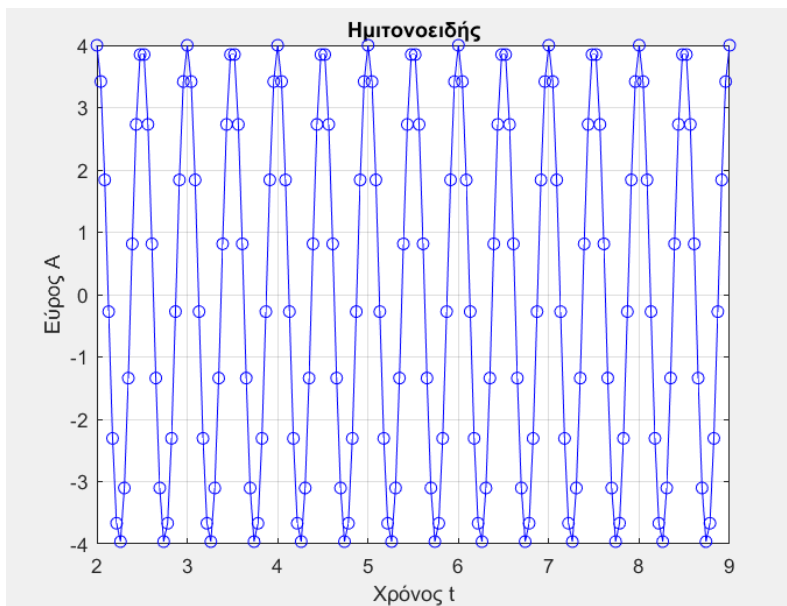
```

```
end  
end
```

6. Άσκηση εξάσκησης:

- Πληκτρολογήστε τον επόμενο κώδικα και πατήστε αποθήκευση ως **myscript_3.m**
- Ποια είναι η γραφική παράσταση που δημιουργείται;

```
fs=23; %συχνότητα δειγματοληψίας  
Ts=1/fs; %περίοδος δειγματοληψίας  
tmin=2; %ελάχιστη τιμή του άξονα  
tmax=9; %μέγιστη τιμή του άξονα  
t=tmin:Ts:tmax; %δειγματοληψία άξονα χρόνου κάθε Ts  
f=2; %ημιτονοειδής συχνότητα  
A=4; %πλάτος ημιτόνου  
phi=0; %φάση ημιτόνου  
%υπολογισμός ημιτόνου  
x=A*cos(2*pi*f*t+phi);  
%plot ημιτόνου  
figure;  
plot(t,x,'bo-');xlabel('Χρόνος t');ylabel('Εύρος A');title('Ημιτονοειδής')  
grid on;
```



7. Άσκηση πρακτικής:

Ας υποθέσουμε ότι θέλουμε να σχεδιάσουμε γραφικά και ένα δεύτερο ημιτονοειδές σήμα. Λαμβάνοντας υπόψη τα παρακάτω, παρακαλούμε να απεικονίσετε και τα δύο σήματα σε μία γραφική παράσταση:

$f_s = 8;$

$T_s = 1/f_s;$

$t_{\min} = 2;$

$t_{\max} = 10;$

$t = t_{\min}:T_s:t_{\max};$

$f_1 = 2;$

$A_1 = 7;$

$\phi_1 = 0;$

$f_2 = 3;$

$A_2 = 9;$

$\phi_2 = 0;$

$x_1 = A_1 \cdot \cos(2 \cdot \pi \cdot f_1 \cdot t + \phi_1);$

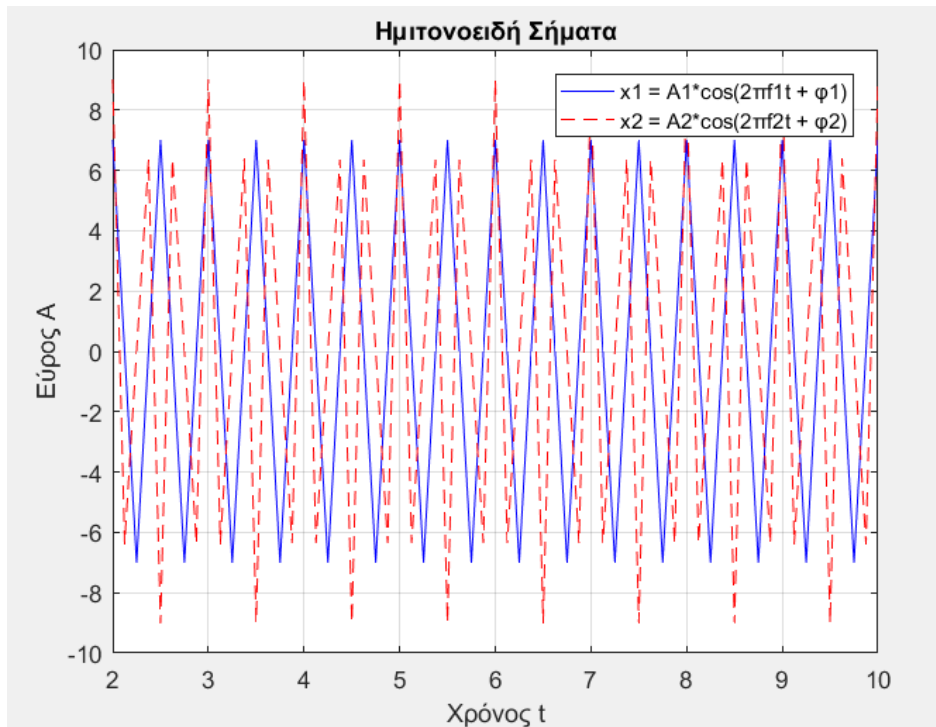
$x_2 = A_2 \cdot \cos(2 \cdot \pi \cdot f_2 \cdot t + \phi_2);$

`figure;`

```

plot(t,x1,'b-',t,x2,'r--');xlabel('Χρόνος t');ylabel('Εύρος A');title('Ημιτονοειδή Σή-
ματα');
legend('x1=A1*cos(2πf1t + φ1)','x2=A2*cos(2πf2t + φ2)');
grid on;

```



1. Μέρος 2ο : Βασικά σήματα

• Δέλτα συνάρτηση

Η **Δέλτα συνάρτηση** (ή αλλιώς Impulse Function) είναι μια θεμελιώδης έννοια στα συστήματα επεξεργασίας σήματος. Στη διακριτή μορφή της, ορίζεται ως:

- $\delta[n]=1$, αν $n=0$
- $\delta[n]=0$, αν $n \neq 0$

Η Δέλτα συνάρτηση χρησιμοποιείται για την **ανάλυση** συστημάτων και την **κατανόηση** της **απόκρισής** τους σε βασικά σήματα.

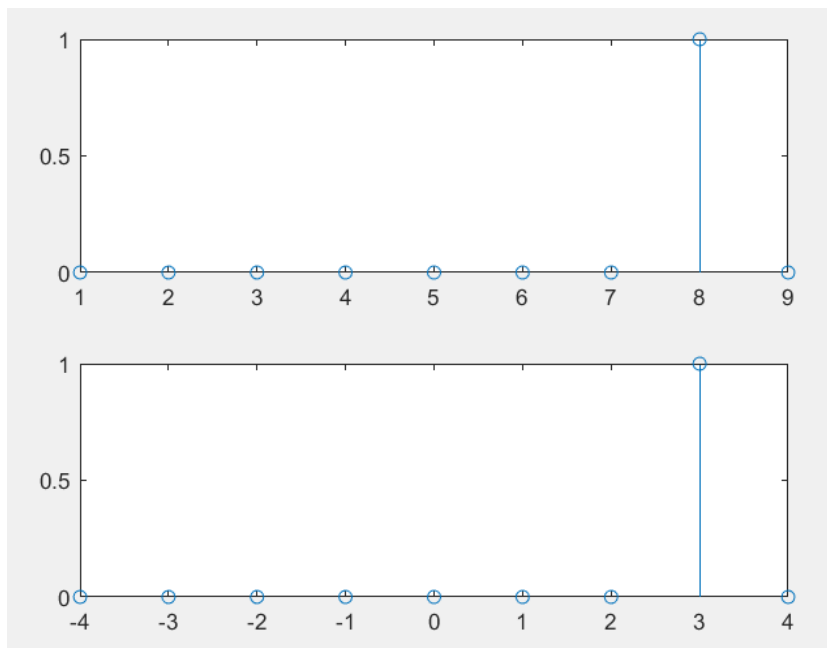
- **Χρήση της εντολής stem:**

Η εντολή **stem** απεικονίζει **διακριτά** σημεία, κάτι που είναι ιδανικό για την **απεικόνιση** της δέλτα συνάρτησης.

- **Γράψτε** κώδικα **MATLAB** για να δημιουργήσετε το διάνυσμα **Imp** και να θέσετε την τιμή **1** στην **8η** θέση.
- **Δημιουργήστε** έναν άξονα χρόνου **n=-4:4n**
- **Χρησιμοποιήστε** την εντολή **stem** για να σχεδιάσετε **δύο** γραφικές παραστάσεις:
 - a) Στην πρώτη, σχεδιάστε το σήμα χωρίς άξονα χρόνου.
 - b) Στη δεύτερη, σχεδιάστε το σήμα σε σχέση με τον άξονα χρόνου.
- **Εκτελέστε** το script και **απαντήστε** στις παρακάτω ερωτήσεις:
 - c) Σε ποια θέση εμφανίζεται η μοναδική μηδενική τιμή στη Δέλτα συνάρτηση;
 - d) Ποια είναι η διαφορά ανάμεσα στις δύο παραστάσεις;

```
Imp = zeros(1,9);  
Imp(8) = 1;  
subplot(2,1,1);  
stem(Imp)  
n= -4:4;  
subplot(2,1,2);stem(n,Imp);  
figure;
```

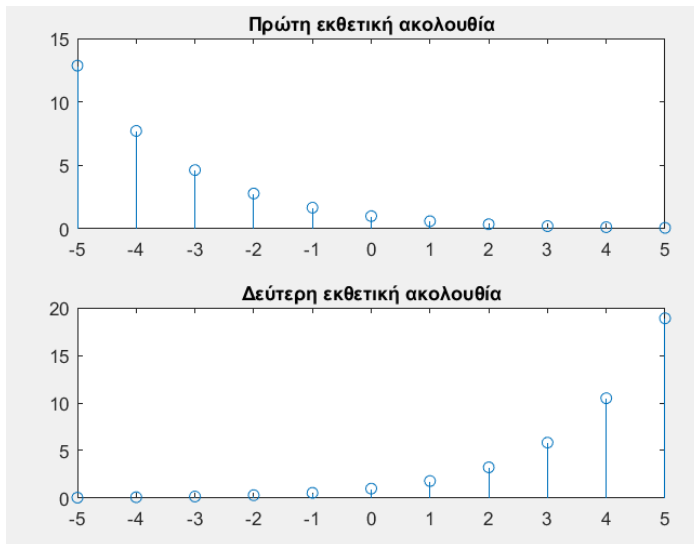
➤ Τι συμβολίζει το αποτέλεσμα του κώδικα;



2. Άσκηση πρακτικής:

Χρησιμοποιείτε τον ακόλουθο κώδικα και παρουσιάστε αυτό που θα σας εμφανίσει.

```
n=-5:5;
a1=0.6;
a2=1.8;
figure;
subplot(2,1,1);stem(n,a1.^n);title('Πρώτη εκθετική ακολουθία');
subplot(2,1,2);stem(n,a2.^n);title('Δεύτερη εκθετική ακολουθία');
```



3. Ασκήσεις προς επίλυση

Άσκηση 1 Να δημιουργηθεί ένα νέο script του οποίου το όνομα θα είναι **mylab1.1.m**, στο οποίο:

Δημιουργήστε έναν πίνακα A με διαστάσεις 3x3, όπου όλα τα στοιχεία της πρώτης σειράς είναι 2, όλα τα στοιχεία της δεύτερης σειράς είναι 4 και όλα τα στοιχεία της τρίτης σειράς είναι 6.

Έπειτα,

1. Ποιο είναι το **μέγεθος** του πίνακα;
2. Ποιος είναι ο **ανάστροφος** του πίνακα και στη συνέχεια να τον παρουσιάσετε.
3. Απομονώστε τη **1η** και την **3η** γραμμή του πίνακα σε δύο νέους πίνακες και εμφανίστε τους ως A και C αντίστοιχα.
4. Βρείτε το **εσωτερικό γινόμενο** των A και C.
5. Και τέλος, υπολογίστε τον πίνακα K χρησιμοποιώντας τον τύπο $K=4A+5Z$, όπου ο πίνακας Z αποτελείται αποκλειστικά από το στοιχείο 1 σε όλες του τις θέσεις.

Άσκηση 2 Να δημιουργηθεί ένα νέο script του οποίου το όνομα θα είναι **mylab1.2.m**, στο οποίο:

1. Δημιουργήστε έναν πίνακα A διαστάσεων 5x5 χρησιμοποιώντας τη συνάρτηση magic.

2. Ποια τα συνολικά αθροίσματα ανά γραμμή, στήλη, της κύριας διαγώνιου, του κάτω και του άνω τριγωνικού πίνακα;
3. Ποιο το **max** και ποιο το **min** του πίνακα;
4. Να γίνει η ταξινόμηση ανά γραμμή σε **φθίνουσα** σειρά και τα αποτελέσματα να διατηρηθούν στον πίνακα A.
5. Αντίστοιχα, να γίνει η ταξινόμηση ανά στήλη σε **αύξουσα** σειρά και τα αποτελέσματα να διατηρηθούν στον πίνακα C.

Άσκηση 3 Να δημιουργηθεί ένα νέο script του οποίου το όνομα θα είναι **mylab1.3.m**, στο οποίο:

- Δημιουργήστε έναν πίνακα A **5×55** με τυχαίες ακέραιες τιμές από **-10 έως 10**.
- Αντικαταστήστε όλα τα αρνητικά στοιχεία του πίνακα με την τιμή **0**.
- Υπολογίστε το άθροισμα των θετικών στοιχείων του πίνακα χρησιμοποιώντας την εντολή **sum**.
- Δημιουργήστε έναν άξονα **x=-3:0.1:3**.
- 4. Σχεδιάστε τις συναρτήσεις **y1=x²** και **y2=sin(x)** στην ίδια γραφική παράσταση με διαφορετικά χρώματα και στυλ γραμμής.
- Εμφανίστε τίτλο στους άξονες.

Άσκηση 4 Να δημιουργηθεί ένα νέο script του οποίου το όνομα θα είναι **mylab1.4.m**, στο οποίο:

- Δημιουργήστε μια συνάρτηση **replace_negatives.m** που δέχεται έναν πίνακα A ως είσοδο και επιστρέφει έναν πίνακα B όπου όλα τα αρνητικά στοιχεία αντικαθίστανται με τις απόλυτες τιμές τους.
- Δημιουργήστε έναν πίνακα **C=randi([-5,5],6,6)** με τυχαίες τιμές από **-5 έως 5**.
- Εφαρμόστε τη συνάρτηση **replace_negatives** στον πίνακα C.
- Σχεδιάστε τρεις διαφορετικές συναρτήσεις **y1=e^x**, **y2=sin(x)** και **y3=cos(x)** σε ένα μόνο γράφημα με διαφορετικά χρώματα και στυλ γραμμής.
- Προσθέστε τίτλους στους άξονες.

5. Συμπεράσματα και Προτάσεις

- **Ανασκόπηση:** Κατά την ολοκλήρωση του εργαστηρίου θα έχεις **αποκτήσει** σημαντικές δεξιότητες στην **επεξεργασία πινάκων**, στη **γραφική αναπαράσταση δεδομένων** και στη **δημιουργία αυτοματοποιημένων scripts**.
- **Εφαρμογές:** Οι τεχνικές που έχουν διδαχθεί είναι θεμελιώδεις για την επιστημονική υπολογιστική **ανάλυση**, την **επεξεργασία σήματος** και την **ανάλυση εικόνας**.

- **Περαιτέρω Διάβασμα:** Συνιστάται η μελέτη επιπλέον **συναρτήσεων** του **MATLAB** (π.χ. polyfit, fft, filter) για την **επεξεργασία** πιο σύνθετων προβλημάτων.

Εργαστήριο 2 – Lab 2

Διαχείριση εικόνων – Χρωματικοί χώροι

Σκοπός εργαστηρίου:

Οι φοιτητές να εξοικειωθούν με τις βασικές έννοιες της επεξεργασίας εικόνας, όπως η αναπαράσταση ψηφιακών εικόνων και τα χρωματικά συστήματα. Επίσης, να μάθουν τις απαραίτητες εντολές στο MATLAB ώστε μπορούν να επεξεργαστούν και να διαχειριστούν εικόνες με αποτελεσματικότητα.

Προσέξτε ότι:

Βεβαιωθείτε ότι έχετε δημιουργήσει έναν φάκελο στον υπολογιστή σας με την ονομασία "Επεξεργασία Εικόνας και Βίντεο", που αντιστοιχεί στο μάθημα. Μέσα σε αυτόν τον κύριο φάκελο, δημιουργήστε έναν υποφάκελο με το ονοματεπώνυμό σας. Αυτός ο υποφάκελος θα χρησιμεύει για την οργάνωση και αποθήκευση όλων των αρχείων .m του MATLAB που θα δημιουργήσετε κατά τη διάρκεια του μαθήματος. Η σωστή οργάνωση των αρχείων σας θα διευκολύνει την πρόσβαση και τη διαχείριση των εργασιών σας.

1. Μαθησιακοί Στόχοι

Με την ολοκλήρωση αυτού του εργαστηρίου ο φοιτητής θα είναι σε θέση να:

- **Κατανοήσει** τους βασικούς χρωματικούς χώρους (**RGB, Grayscale, HSV**, κλπ.) και πώς αυτοί αναπαρίστανται στο MATLAB.
- **Εφαρμόσει** εντολές για την **ανάγνωση, εμφάνιση** και **αποθήκευση** εικόνων (imread, imshow, imwrite, imfinfo).
- **Απομονώσει** και **αναλύσει** τις χρωματικές συνιστώσες (**R, G, B**) και να **κατανοήσει** τη συμβολή τους στην τελική εμφάνιση μιας εικόνας.
- **Εκτελέσει** μετατροπές μεταξύ διαφορετικών χρωματικών χώρων (π.χ. μετατροπή έγχρωμης εικόνας σε grayscale) και να **συγκρίνει** τις μεθόδους (μέσος όρος vs. συνταγή βαρύτητας).
- **Υπολογίσει** στατιστικά χαρακτηριστικά εικόνων (max, min, mean) και να τα **συσχετίσει** με την ποιότητα και τα χαρακτηριστικά των εικόνων.
- **Εφαρμόσει** τεχνικές διαχείρισης εικόνων και προχωρημένες λειτουργίες, όπως η αλλαγή μεγέθους, ο υπολογισμός του μεγέθους σε bits/bytes και η ανάλυση της συμπίεσης.

2. Πώς το μαθαίνουν:

Μέσω των παρακάτω εργαστηριακών ασκήσεων, ο φοιτητής δεν θα εξασκηθεί μόνο στις βασικές εντολές, αλλά θα υλοποιήσει επίσης:

- **Αναπτύξει** scripts που υλοποιούν βασικές και προχωρημένες λειτουργίες επεξεργασίας εικόνας.
- **Πειραματιστεί** με διαφορετικούς χρωματικούς χώρους και τεχνικές μετατροπής, **συγκρίνοντας** τα αποτελέσματα.
- **Αναλύσει** στατιστικά χαρακτηριστικά εικόνων για να **κατανοήσει** τη σχέση μεταξύ βάθους χρώματος, ποιότητας και συμπίεσης.

Ποιοι είναι οι βασικοί τύποι εικόνων;

- **Οι binary (δυαδικοί):** οι εικόνες αποθηκεύονται στο MATLAB ως πίνακες με διαστάσεις $N \times M$, με τιμές που εκφράζονται σε κλίμακα από 0 έως 1, όπου οι αριθμοί αντιστοιχούν στη φωτεινότητα ή το επίπεδο γκρι της εικόνας.
- **Οι έγχρωμες εικόνες:** οι εικόνες μπορεί να αποθηκεύονται στο MATLAB χρησιμοποιώντας το χρωματικό σύστημα RGB και να αναπαρίστανται ως πίνακες $M \times N \times 3$, όπου οι αριθμοί αντιπροσωπεύουν τις τιμές του κόκκινου (R), του πράσινου (G) και του μπλε (B) χρώματος. Οι τιμές αυτές μπορεί να είναι πραγματικοί αριθμοί σε κλίμακα από 0 έως 1, ή από 0 έως 255, ή από 0 έως 65535, ανάλογα με τις απαιτήσεις της εφαρμογής.
- **Φωτεινότητας:** Οι εικόνες αποθηκεύονται ως πίνακες $N \times M$ με πραγματικούς αριθμούς στην κλίμακα [0,1], [0,255] ή [0,65535], ανάλογα με τις απαιτήσεις της εφαρμογής.

Ανάγνωση εικόνων

Κάποιες χρήσιμες εντολές που θα χρειαστούμε από το περιβάλλον του Matlab είναι οι ακόλουθες:

Διαδικασίες Εισαγωγής και Εξαγωγής Εικόνων

- **Imread:** Ανάγνωση εικόνας από αρχείο και εισαγωγή στο χώρο εργασίας.
- **Imwrite:** Εγγραφή της εικόνας από το workspace σε αρχείο εικόνας, όπως ένα .jpg.
- **Imfinfo:** Προσφέρει πληροφορίες σχετικά με ένα αρχείο εικόνας.

Εντολές Οπτικοποίησης

- **Imshow:** Παρουσιάζει την εικόνα μας.
- **Subimage:** Παρουσιάζει πολλές και διάφορες εικόνες μέσα σε ένα σχήμα.

Εντολές Μετατροπής

- **gray2ind:** Τροποποίηση από επίπεδα γκρι ή δυαδικές εικόνες σε εικόνες δεικτών.
- **ind2gray:** Μετατροπή εικόνας δεικτών σε εικόνα αντίστοιχων επιπέδων γκρι.
- **mat2gray:** Αλλαγή ενός πίνακα σε εικόνα επιπέδου γκρι.
- **rgb2gray:** Τροποποίηση έγχρωμης εικόνας σε γκρι.
- **ind2rgb:** Τροποποίηση μιας εικόνας δεικτών σε έγχρωμη.

Ως όνομα αρχείου εικόνας μπορούμε να χρησιμοποιήσουμε τους παρακάτω τύπους:

- Τύπος εικόνας JPEG (.jpg ή .jpeg)
- Τύπος εικόνας Portable Network Graphics (.png)
- Τύπος εικόνας Bitmap (.bmp)
- Τύπος εικόνας Tagged Image File Format (.tif ή .tiff)
- Τύπος εικόνας Graphics Interchange Format (.gif)

Αποθηκεύουμε τις διαστάσεις μιας εικόνας με τον τύπο: **[M, N] = size (I);**

Προσέχω: Εάν έχουμε φορτώσει μια έγχρωμη εικόνα, τότε η τιμή του N θα είναι 3 φορές τη διάσταση της εικόνας λόγω των καναλιών RGB. Μπορούμε να εξάγουμε τις διαστάσεις – συνιστώσες R, G και B με τον εξής τρόπο:

- $I(:, :, 1)$ για την R
- $I(:, :, 2)$ για την B
- $I(:, :, 3)$ για την G

Άσκηση προς επίλυση

Στόχος άσκησης: Να κατανοήσετε πώς αναλύονται τα στατιστικά χαρακτηριστικά μιας εικόνας σε MATLAB και πώς διαφοροποιούνται αυτά μεταξύ grayscale και έγχρωμων εικόνων.

Από το αποθετήριο εικόνων: <https://links.uwaterloo.ca/Repository.html> κάνετε λήψη την εικόνα **frog** από το grayscale set 2.

1. Ανάλυση grayscale εικόνας

Κάνετε χρήση τον ακόλουθο κώδικα:

```
I=imread('frog.tif');
imshow(I)
size(I)
I
max_val=max(I(:))
min_val=min(I(:))
mean_val=mean(I(:))
```

```
>> max_val=max(I(:))

max_val =

    254

>> min_val=min(I(:))

min_val =

     0

>> mean_val=mean(I(:))

mean_val =

    122.0409
```

- Τι σημαίνουν τα παραπάνω: **max_val**, **min_val** και **mean_val**;
- Ποιες τιμές έχουν;

2. Ανάλυση έγχρωμης εικόνας

Αυτή τη φορά κάνετε λήψη από το παραπάνω link τις έγχρωμες τουλίπες και χρησιμοποιείτε ξανά τον ίδιο κώδικα. **Τι παρατηρείτε;**

- Τι παρατηρείτε για τις τιμές **max_val**, **min_val** και **mean_val** της έγχρωμης εικόνας;

3. Σύγκριση και Επεξήγηση

- Συγκρίνετε τις τιμές μεταξύ της grayscale και της έγχρωμης εικόνας.
- Πώς ερμηνεύετε τις διαφορές;
- Ποιες είναι οι πιθανές αιτίες για τις διαφορές στη μέση τιμή (mean_val);

4. Εμφανίστε τα στατιστικά ανά κανάλι χρώματος (R, G, B) για την εικόνα "tulips". Χρησιμοποιήστε:

```
R = tulips(:,:,1);
G = tulips(:,:,2);
B = tulips(:,:,3);
mean_R = mean(R(:));
mean_G = mean(G(:));
mean_B = mean(B(:));
```

- Ποιο κανάλι πιστεύετε ότι συνεισφέρει περισσότερο στη μέση τιμή της έγχρωμης εικόνας;

Καταγραφή εικόνων

Στόχος άσκησης: Να εξοικειωθείτε με την καταγραφή και επεξεργασία των επιμέρους συνιστωσών (Red, Green, Blue) μιας έγχρωμης εικόνας. Θα μάθετε πώς να απομονώνετε τις συνιστώσες RGB από μια έγχρωμη εικόνα και να τις εμφανίζετε ξεχωριστά.

Στο περιβάλλον του Matlab υπάρχουν κάποιες συγκεκριμένες εντολές για την καταγραφή εικόνων. Αυτές είναι οι ακόλουθες:

1. Φόρτωση και Καταγραφή Συνιστωσών RGB:

- κατεβάστε την εικόνα **"tulips.tif"**.
- Φορτώστε την εικόνα χρησιμοποιώντας την εντολή **imread** και απομονώστε τις τρεις συνιστώσες της (Red, Green, Blue) χρησιμοποιώντας την αρίθμηση των καναλιών (1 για Red, 2 για Green, 3 για Blue).
- Αποθηκεύστε κάθε συνιστώσα σε ξεχωριστό αρχείο χρησιμοποιώντας την εντολή **imwrite**.
- Χρησιμοποιήστε την εντολή **subplot** για να εμφανίσετε όλες τις συνιστώσες (Red, Green, Blue) και την αρχική εικόνα σε ένα μόνο **figure**.
- Κάθε συνιστώσα πρέπει να εμφανίζεται σε διαφορετικό τμήμα του **figure** με τίτλο.

```
I=imread('tulips.tif');  
imwrite(I(:,:,1),'tulipsRED.tif');  
Ired=imread('tulipsRED.tif');  
imshow(Ired),title('RED Συνιστώσα');
```



- Επαναλάβετε τον κώδικα και για τις άλλες δυο συνιστώσες **GREEN** και **BLUE**. Τι παρατηρείτε;
- Τι παρατηρείτε όταν εμφανίζετε κάθε συνιστώσα ξεχωριστά;
- Ποιες περιοχές της εικόνας είναι εμφανείς σε κάθε συνιστώσα;
- Ποια συνιστώσα κυριαρχεί στην εμφάνιση της εικόνας;
- Με τη χρήση της εντολής **subplot** εμφανίστε σε ένα συνολικό figure τα **RGB**.

```
I = imread('tulips.tif');
subplot(1, 4, 1);imshow(I(:,:,1));title('Red Συνιστώσα');
subplot(1, 4, 2);imshow(I(:,:,2));title('Green Συνιστώσα');
subplot(1, 4, 3);imshow(I(:,:,3));title('Blue Συνιστώσα');
subplot(1, 4, 4);imshow(I);title('Αρχική Εικόνα'); figure;
```

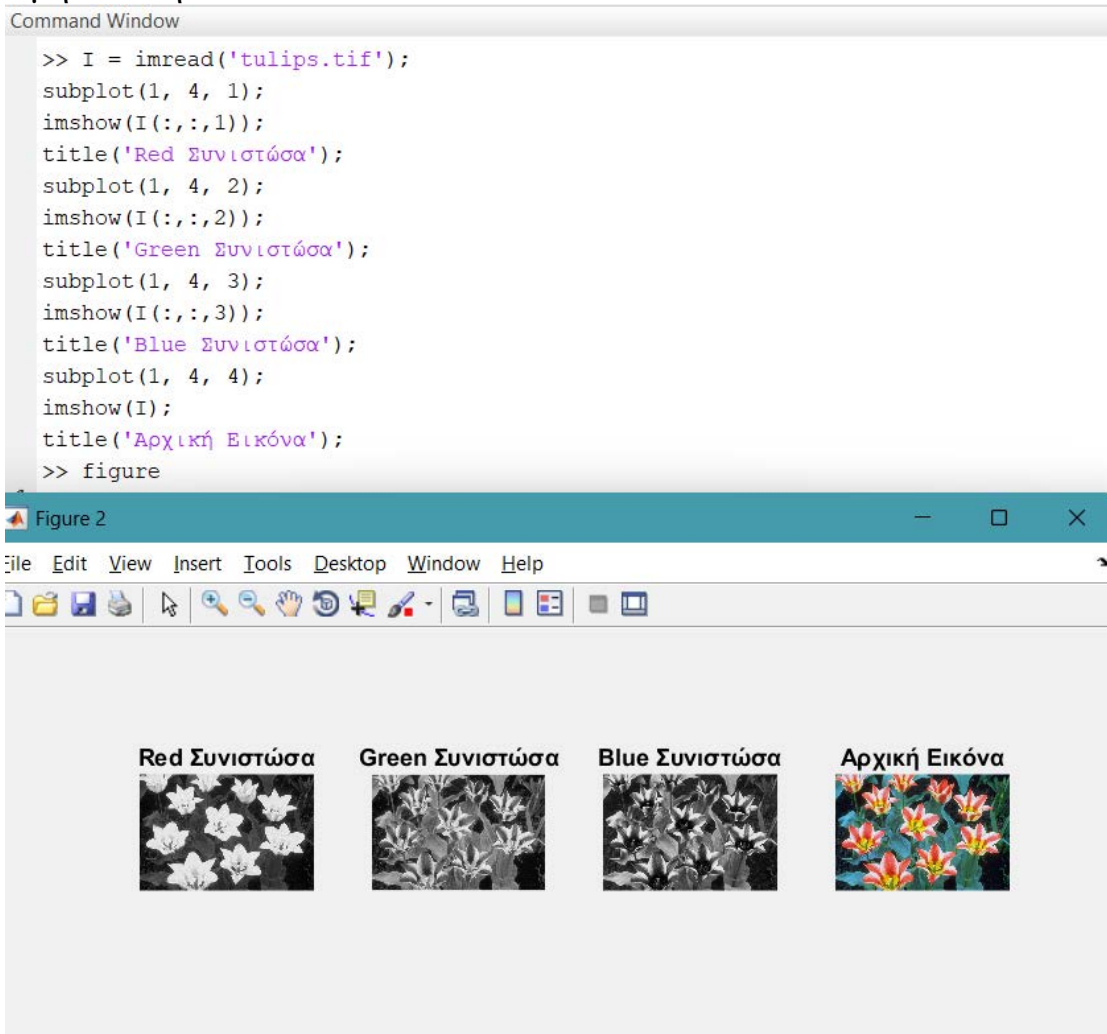
Επιπλέον ερωτήσεις

- Ποιες είναι οι διαφορές που παρατηρείτε στις συνιστώσες (**Red, Green, Blue**);
- Δοκιμάστε να απομονώσετε μία συνιστώσα και να την εμφανίσετε σε συνδυασμό με τις υπόλοιπες δύο με σκοπό να ανακατασκευάσετε την εικόνα.

- Εάν εμφανίσετε μόνο τη **Red** συνιστώσα, τι βλέπετε; Είναι η εικόνα πλήρης ή εμφανίζεται μόνο το κόκκινο μέρος της;

```
I_red_only = I;  
I_red_only(:, :, 2) = 0;  
I_red_only(:, :, 3) = 0;
```

- Χρήσιμη εντολή:



Υπολογισμός πλήθους bits (μεγέθους) εικόνων

Στόχος Άσκησης: Να κατανοήσετε τον υπολογισμό του μεγέθους (bits, bytes) μιας εικόνας, τη σχέση του με το βάθος χρώματος (bit depth) και την επίδραση της συμπίεσης στο μέγεθος του αρχείου.

1. Κατανόηση Θεωρίας

Μία εικόνα με διαστάσεις $M \times N$, όπου M είναι ο αριθμός των γραμμών και N ο αριθμός των στηλών, συντίθεται καθολικά από $M \times N$ pixels. Κάθε ένα από αυτά τα pixels αποφέρει πληροφορία σχετικά με το χρωματισμό ή την απόχρωση που παρουσιάζει. Ο αριθμός των αποχρώσεων που δύναται να αναπαραστήσει ένα pixel συναρτάται από το βάθος bit m , το οποίο είναι επίσης γνωστό ως βάθος χρώματος. Το βάθος bit διατυπώνει την χρωματική πληροφορία που μπορεί να αποθηκεύσει κάθε pixel. Συγκεκριμένα, αν το βάθος bit είναι m , τότε ο αριθμός των δυνατών αποχρώσεων που μπορεί να αναπαραστήσει ένα pixel είναι $G = 2^m$. Παραδείγματος χάρη:

- Αν $m = 1$, τότε το pixel μπορεί να έχει **2 αποχρώσεις**, μπορεί να είναι είτε μαύρο είτε άσπρο.
- Αν $m = 8$, τότε το pixel μπορεί να έχει **256 αποχρώσεις**, μπορεί να αναπαραστήσει 256 διαφορετικά επίπεδα του γκρι σε μία grayscale εικόνα.
- Αν $m = 24$, τότε το pixel μπορεί να έχει 16.777.216 αποχρώσεις, μία πλήρη παλέτα χρωμάτων σε μία έγχρωμη εικόνα.

2. Εκτέλεση Κώδικα και Υπολογισμοί

Χρησιμοποιείστε τον παρακάτω κώδικα και υπολογίστε τα ζητούμενα. Σε περίπτωση που γίνει λήψη εικόνας τύπου **TIFF**, μετατρέψτε την σε **JPG**. Επιλέξτε την εικόνα που αναπαριστά τις τουλίπες (**tulips**).

```
a=imfinfo('tulips.jpg');  
imbits=a.Width*a.Height*a.BitDepth  
imbytes=imbits/8  
filesize=a.FileSize  
imcompress=imbytes/filesize
```

- Ποιο είναι το συνολικό πλήθος **bits** (imbits) που απαιτείται για την αποθήκευση της εικόνας χωρίς συμπίεση;
- Ποιο είναι το μέγεθος του συμπιεσμένου αρχείου (**filesize**);
- Ποιο είναι το ποσοστό συμπίεσης της εικόνας (**imcompress**);

3. Σύγκριση Βαθών Bit

Μετατρέψτε την εικόνα σε grayscale (8 bits/pixel) χρησιμοποιώντας την εντολή:

```
gray_img = rgb2gray(imread('tulips.jpg'));  
imwrite(gray_img, 'tulips_gray.jpg');  
b = imfinfo('tulips_gray.jpg');  
gray_imbits = b.Width * b.Height * b.BitDepth;  
gray_imbytes = gray_imbits / 8;  
gray_filesize = b.FileSize;  
gray_imcompress = gray_imbytes / gray_filesize;
```

- Υπολογίστε το **gray_imcompress** για την **grayscale** εικόνα και συγκρίνετε τα αποτελέσματα με την έγχρωμη εικόνα.
- Πώς αλλάζει το μέγεθος και ο λόγος συμπίεσης όταν μειώνεται το βάθος **bit**;
- Τι ρόλο παίζει η ποιότητα συμπίεσης και πώς επηρεάζει την ακρίβεια της εικόνας;

Command Window

```
>> a=imfinfo('tulips.jpg');  
>> imbits=a.Width*a.Height*a.BitDepth  
  
imbits =  
  
    9437184  
  
>> imbytes=imbits/8  
  
imbytes =  
  
    1179648  
  
>> filesize=a.FileSize  
  
filesize =  
  
    86296  
  
>> imcompress=imbytes/filesize  
  
imcompress =  
  
    13.6698
```

Θυμάμαι ότι:

1. **Imbits**= Το σύνολο των bit της εικόνας
2. **Imcompress**= Ο συντελεστής συμπίεσης που συμβολίζεται με (%)
3. **FileSize**= Το σύνολο των byte ε-

Ασκήσεις κατανόησης

1. Υπολογισμός Μεγέθους Εικόνας

Μια εικόνα έχει διαστάσεις 800 x 600 pixels και βάθος bit 24.

- Υπολογίστε το συνολικό πλήθος **bits** (imbits) και **bytes** (imbytes) που απαιτούνται για την αποθήκευση της εικόνας χωρίς συμπίεση.

2. Ανάλυση Βάθους Bit

Δίνεται μια grayscale εικόνα με βάθος bit 8 και διαστάσεις 1024 x 768 pixels.

- Πόσες διαφορετικές **αποχρώσεις** μπορεί να αναπαραστήσει η εικόνα;
- Πόσος είναι ο απαιτούμενος **χώρος αποθήκευσης** (σε bytes) χωρίς συμπίεση;

3. Σύγκριση Συμπίεσης

Μια έγχρωμη εικόνα έχει συνολικό μέγεθος σε bytes 3.145.728 όταν αποθηκεύεται χωρίς συμπίεση και συμπιεσμένο μέγεθος 512.000 bytes.

- Υπολογίστε τον **λόγο συμπίεσης** της εικόνας.
- Είναι υψηλός ή χαμηλός ο λόγος αυτός για μία **JPEG** εικόνα; Γιατί;

Μετατροπή μεγέθους μιας εικόνας

Στόχος Άσκησης: Να κατανοήσετε τη διαδικασία αλλαγής μεγέθους μιας εικόνας σε MATLAB, να αναγνωρίσετε τις διαφορές στην ποιότητα απεικόνισης καθώς μειώνεται η ανάλυση και να κατανοήσετε τη χρήση κλιμάκων ανάλυσης για εικόνες.

Κάνετε λήψη από το αποθετήριο εικόνων εκείνη με το όνομα **monarch**. Αν δεν βρίσκεται στην μορφή **jpg**, απλώς μετατρέψτε την. Πληκτρολογήστε στο command window την εντολή **edit myscript_1** και στη συνέχεια αποθηκεύστε το. Χρησιμοποιείστε τον ακόλουθο κώδικα:

```
I = imread('monarch.jpg');  
im=rgb2gray(imread('monarch.jpg'));  
im1=imresize(im, [1024 1024]);  
im2=imresize(im1, [1024 1024]/2);  
im3=imresize(im1, [1024 1024]/4);  
im4=imresize(im1, [1024 1024]/8);  
im5=imresize(im1, [1024 1024]/16);  
  
im6=imresize(im1, [1024 1024]/32);  
figure;  
subplot(2,3,1);imshow(im1);title('1024x1024');  
subplot(2,3,2);imshow(im2);title('512x512');  
subplot(2,3,3);imshow(im3);title('256x256');  
subplot(2,3,4);imshow(im4);title('128x128');  
subplot(2,3,5);imshow(im5);title('64x64');  
subplot(2,3,6);imshow(im6);title('32x32');
```

figure;

1. Παρατηρήσεις Ποιότητας

- Τι αλλαγές παρατηρείτε στην ποιότητα της εικόνας καθώς μειώνεται η ανάλυση (π.χ., από 1024x1024 σε 32x32);

2. Υπολογισμοί

Υπολογίστε πόσα pixels περιέχει κάθε εικόνα στις εξής αναλύσεις:

- 1024x1024
- 512x512
- 256x256
- 128x128
- 64x64
- 32x32

3. Ποιο είναι το μέγεθος του αρχείου (σε bytes) για κάθε ανάλυση; Χρησιμοποιήστε την εντολή:

```
info = imfinfo('filename.jpg');  
filesize = info.FileSize;
```

- Πώς μεταβάλλεται το μέγεθος του αρχείου καθώς μειώνεται η ανάλυση;

Ασκήσεις κατανόησης

1. Υπολογισμός Pixels Μίας Εικόνας

Στόχος άσκησης: Να υπολογίσετε τον αριθμό των pixels μιας εικόνας για διαφορετικές αναλύσεις.

Πληκτρολογήστε τον παρακάτω κώδικα στο MATLAB:

```
resolutions = [1024, 512, 256, 128, 64, 32];  
disp('Ανάλυση | Αριθμός Pixels');  
disp('-----');  
for r = resolutions  
    num_pixels = r * r;
```

```
fprintf('%4dx%4d | %12d\n', r, r, num_pixels);  
end
```

- Πόσα pixels έχει η εικόνα στις αναλύσεις 1024x1024 και 32x32;
- Ποια είναι η διαφορά στον αριθμό των pixels ανάμεσα στις αναλύσεις 512x512 και 256x256.

2. Ποσοστιαία Μείωση Pixels

Στόχος άσκησης: Να υπολογίσετε το ποσοστό μείωσης των pixels όταν μειώνεται η ανάλυση.

Πληκτρολογήστε τον παρακάτω κώδικα στο MATLAB:

```
original_size = 1024 * 1024;  
resolutions = [1024, 512, 256, 128, 64, 32];  
disp('Ανάλυση | Μείωση Pixels (%)');  
disp('-----');  
for r = resolutions  
    num_pixels = r * r;  
    reduction_percentage = (1 - (num_pixels / original_size)) * 100;  
    fprintf('%4dx%4d | %17.2f %%\n', r, r, reduction_percentage);  
end
```

- Ποια είναι η ποσοστιαία μείωση pixels όταν περνάμε από ανάλυση **1024x1024** σε **512x512**;
- Τι παρατηρείτε για το ποσοστό μείωσης όσο μειώνεται η ανάλυση;

Χρωματικοί χώροι

1. Μετατροπή Έγχρωμης Εικόνας σε Γκρί (Grayscale)

Στόχος άσκησης: Η άσκηση στοχεύει στην κατανόηση του τρόπου μετατροπής μιας έγχρωμης εικόνας σε γκρί, χρησιμοποιώντας την μέση τιμή ή τον αλγόριθμο μετατροπής μέσω των καναλιών RGB. Επίσης, η άσκηση ενσωματώνει την έννοια της απόχρωσης σε μια έγχρωμη εικόνα.

2. Υπολογισμός Απόχρωσης

Η απόχρωση αναφέρεται στον τύπο του χρώματος που προκύπτει από την αναλογία των τριών καναλιών RGB. Για να υπολογίσουμε την απόχρωση της εικόνας, μπορούμε να χρησιμοποιήσουμε τον μέσο όρο των τιμών των τριών καναλιών (R, G, B) για κάθε pixel.

Για κάθε pixel, ο υπολογισμός της απόχρωσης είναι απλός:

$$\text{Απόχρωση} = \frac{R+G+B}{3}$$

```
R = [50, 41, 37; 21, 8, 23; 14, 42, 29];  
G = [160, 132, 166; 177, 189, 164; 144, 123, 189];  
B = [250, 250, 250; 250, 250, 250; 250, 250, 250];  
hue = (R + G + B) / 3;  
disp('Απόχρωση για κάθε pixel:');  
disp(hue);
```

- Ποια είναι η απόχρωση για κάθε pixel, χρησιμοποιώντας τον υπολογισμό του μέσου όρου των τιμών **RGB**;

Ας θεωρήσουμε ένα τμήμα μιας έγχρωμης εικόνας $I(x,y)$ στο χρωματικό σύστημα RGB, όπου οι τιμές φωτεινότητας για τα τρία κανάλια προέρχονται από τους ακόλουθους πίνακες:

$$R = \begin{bmatrix} 50 & 41 & 37 \\ 21 & 8 & 23 \\ 14 & 42 & 29 \end{bmatrix} \quad G = \begin{bmatrix} 160 & 132 & 166 \\ 177 & 189 & 164 \\ 144 & 123 & 189 \end{bmatrix} \quad B = \begin{bmatrix} 250 & 250 & 250 \\ 250 & 250 & 250 \\ 250 & 250 & 250 \end{bmatrix}$$

- Ποια θα είναι η απόχρωση αυτού του τμήματος της εικόνας;
- Εφαρμόστε τον εξής αλγόριθμο για να μετατρέψετε την έγχρωμη εικόνα σε εικόνα σε επίπεδα του γκρι (grayscale):

Θυμάμαι ότι:

Ο παρακάτω αλγόριθμος χρησιμοποιείται από τη συνάρτηση `rgb2gray` του MATLAB για τη μετατροπή έγχρωμων εικόνων σε εικόνες επιπέδων του γκρι. Ο αλγόριθμος βασίζεται στη μετατροπή των τιμών χρώματος RGB σε τιμές φωτεινότητας, σύμφωνα με το χρωματικό σύστημα YUV. Ειδικότερα, υπολογίζει το κανάλι φωτεινότητας Y, το οποίο συνδυάζει τα κανάλια R (κόκκινο), G (πράσινο) και B (μπλε) με τα εξής βάρη:

$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

```
R = [50 41 37; 21 8 23; 14 42 29];  
G = [160 132 166; 177 189 164; 144 123 189];  
B = [250 250 250; 250 250 250; 250 250 250];  
I = cat(3, R, G, B);  
imshow(uint8(I));
```

3. Άσκηση Κατανόησης: Υπολογισμός και Σύγκριση Τιμών Γκρι

Στόχος άσκησης: Η άσκηση αυτή στοχεύει στην κατανόηση της μετατροπής μιας εικόνας RGB σε γκρι και στη σύγκριση των αποτελεσμάτων με διαφορετικές μεθόδους.

Δίνεται το παρακάτω τμήμα εικόνας με τιμές RGB:

```
R = [100, 150, 200];  
G = [50, 100, 150];  
B = [200, 250, 100];
```

- **Μετατροπή σε Γκρι (Grayscale):** Χρησιμοποιήστε τη μέθοδο του μέσου όρου για να μετατρέψετε κάθε pixel σε γκρι:

$$\text{Grayscale} = \frac{R+G+B}{3}$$

Ή κάνε χρήση του τύπου: $Y = 0.2989 * R + 0.5870 * G + 0.1140 * B$

```
R = [100, 150, 200];  
G = [50, 100, 150];  
B = [200, 250, 100];  
grayscale_avg = (R + G + B) / 3;
```

```

grayscale_weighted = 0.2989 * R + 0.5870 * G + 0.1140 *
B;
disp('Γκρίζες Τιμές (Μέσος Όρος) : ');
disp(grayscale_avg);
disp('Γκρίζες Τιμές (Συνταγή Βαρύτητας) : ');
disp(grayscale_weighted);

```

- Συγκρίνετε τα αποτελέσματα της μετατροπής με μέσο όρο και με τη συνταγή βαρύτητας. Ποιες διαφορές παρατηρείτε;
- Ποια μέθοδος προτιμάτε για να μετατρέψετε την εικόνα σε γκρί, και γιατί;

1. Μετατροπή Έγχρωμης Εικόνας σε Γκρί (Grayscale) με Ανάλυση Συνιστωσών

Στόχος άσκησης: Στην άσκηση αυτή, καλείστε να μετατρέψετε μια έγχρωμη εικόνα σε επίπεδα του γκρι (grayscale), εφαρμόζοντας αναλυτικά τον τύπο μετατροπής για κάθε συνιστώσα RGB ξεχωριστά και να υπολογίσετε τις αντίστοιχες τιμές γκρί.

Η εικόνα έχει κυρίως **μπλε απόχρωση**, με μια **ελαφριά πράσινη συνιστώσα** και **ελάχιστο κόκκινο**. Η μετατροπή σε γκρί μπορεί να γίνει αναλυτικά για κάθε συνιστώσα RGB, χρησιμοποιώντας τον τύπο μετατροπής:

$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

Αυτός ο τύπος εφαρμόζεται σε κάθε pixel της εικόνας, και κάθε συνιστώσα **RGB** συμβάλλει διαφορετικά στην **τελική** τιμή γκρί λόγω των διαφορετικών βαρών που δίνονται στα κανάλια.

Χρησιμοποίησε τον παρακάτω κώδικα και απάντησε στις εξής ερωτήσεις:

```

R = [100, 120, 140; 110, 130, 140; 90, 100, 150];
G = [50, 60, 70; 80, 90, 100; 40, 50, 60];
B = [200, 220, 250; 210, 230, 240; 190, 180, 170];
I_gray = 0.2989 * R + 0.5870 * G + 0.1140 * B;
disp('Τελικές τιμές εικόνας σε επίπεδα του γκρι
(Grayscale): ');

```

`disp(I_gray);`

- Ποιες είναι οι τελικές τιμές για την εικόνα σε επίπεδα του γκρι για το πρώτο pixel (100, 50, 200) χρησιμοποιώντας τον τύπο μετατροπής;
- Πώς επηρεάζει η μεγάλη τιμή του μπλε την τελική εικόνα σε επίπεδα του γκρι σε σχέση με το κόκκινο και το πράσινο;

1. Η εικόνα θα έχει κυρίως μπλε απόχρωση με μια ελαφριά πράσινη συνιστώσα. Το κόκκινο είναι ελάχιστα παρόν, οπότε δεν επηρεάζει σημαντικά την τελική χρωματική εντύπωση.



2. Για να μετατρέψεις την έγχρωμη εικόνα σε επίπεδα του γκρι εφαρμόζοντας τον τύπο αναλυτικά για κάθε μια συνιστώσα ξεχωριστά. Οπότε θα έχουμε τα εξής:

$$\begin{aligned} Y(1,1) &= 0.2989 * 50 + 0.5870 * 160 + 0.1140 * 250 = 11.956 + 105.66 + 28.5 = 137.365 \\ Y(1,2) &= 0.2989 * 41 + 0.5870 * 132 + 0.1140 * 250 = 15.543 + 97.442 + 28.5 = 118.3639 \\ Y(1,3) &= 0.2989 * 37 + 0.5870 * 166 + 0.1140 * 250 = 23.912 + 103.90 + 28.5 = 137.1453 \\ Y(2,1) &= 0.2989 * 21 + 0.5870 * 177 + 0.1140 * 250 = 16.141 + 108.01 + 28.5 = 138.5559 \\ Y(2,2) &= 0.2989 * 8 + 0.5870 * 189 + 0.1140 * 250 = 9.8637 + 106.83 + 28.5 = 141.5742 \\ Y(2,3) &= 0.2989 * 23 + 0.5870 * 164 + 0.1140 * 250 = 25.407 + 95.681 + 28.5 = 132.1647 \\ Y(3,1) &= 0.2989 * 14 + 0.5870 * 144 + 0.1140 * 250 = 9.5648 + 105.66 + 28.5 = 116.7336 \\ Y(3,2) &= 0.2989 * 42 + 0.5870 * 123 + 0.1140 * 250 = 17.934 + 102.72 + 28.5 = 113.1988 \\ Y(3,3) &= 0.2989 * 29 + 0.5870 * 189 + 0.1140 * 250 = 20.923 + 99.790 + 28.5 = 148.7791 \end{aligned}$$

Συνεπώς, η τελική εικόνα σε επίπεδα του γκρι θα έχει τις εξής τιμές:

$$I(\text{gray}) = \begin{bmatrix} 137 & 118 & 137 \\ 139 & 142 & 132 \\ 117 & 113 & 149 \end{bmatrix}$$

4. Άσκηση κατανόησης: Υπολογισμός Γκρι Αξίας για Διάφορα Pixels

Στόχος άσκησης: Η άσκηση αυτή έχει στόχο την εξοικείωση με τη διαδικασία μετατροπής των τιμών RGB σε επίπεδα του γκρι, και τη συγκέντρωση της κατανόησης για τον τρόπο που κάθε χρωματική συνιστώσα επηρεάζει την τελική εικόνα.

Δίνονται οι τιμές **RGB** για διάφορα pixels της εικόνας:

Pixel 1: R=120, G=80, B=200

Pixel 2: R=100, G=150, B=50

Pixel 3: R=60, G=120, B=180

Χρησιμοποιώντας τον τύπο μετατροπής σε γκρι:

- Υπολογίστε τις τιμές γκρι για κάθε ένα από τα παραπάνω pixels.
- Ποιο από τα παραπάνω **pixels** θα έχει τη **φωτεινότερη** τιμή γκρι και ποιο το **σκοτεινότερο**;
- Πώς επηρεάζουν οι τιμές **RGB** (ειδικά η πράσινη και η μπλε συνιστώσα) την τελική φωτεινότητα της εικόνας;

5. Ασκήσεις προς επίλυση

Άσκηση 1: Επεξεργασία και Μετατροπή Έγχρωμης Εικόνας σε Γκρι με Ανάλυση Χρωματικών Συνιστωσών

Στην άσκηση αυτή θα εργαστείτε με μια εικόνα RGB και θα υπολογίσετε τις τιμές του γκρι χρησιμοποιώντας δύο διαφορετικές μεθόδους μετατροπής. Επίσης, θα υπολογίσετε την απόχρωση για κάθε pixel της εικόνας.

- Δεδομένα RGB για 5 Pixels:
R=[180, 50, 130, 90, 200]
G=[100, 200, 170, 130, 70]
B=[220, 60, 120, 170, 250]
- Χρησιμοποιώντας τις παρακάτω μεθόδους, μετατρέψτε την εικόνα σε επίπεδα του γκρι:
 - **Μέθοδος 1:** $\text{Grayscale} = \frac{R+G+B}{3}$

➤ **Μέθοδος 2:** $Y = 0.2989 * R + 0.5870 * G + 0.1140 * B$

- Υπολογίστε την απόχρωση για κάθε pixel της εικόνας χρησιμοποιώντας τον τύπο:

➤ $\text{Απόχρωση} = \frac{R+G+B}{3}$

- Συγκρίνετε τα αποτελέσματα των δύο μεθόδων και εξηγήστε τις διαφορές στις τιμές γκρί. Ποια μέθοδος φαίνεται να αναπαριστά καλύτερα τη φωτεινότητα;

Ερωτήσεις:

- a) Ποιες είναι οι τελικές τιμές **γκρί** για κάθε **pixel** χρησιμοποιώντας και τις **δύο** μεθόδους;
- b) Ποιες αποχρώσεις **επικρατούν** για κάθε pixel και πώς η αξία της απόχρωσης συνδέεται με τις τιμές **RGB**;
- c) Ποιες είναι οι **διαφορές** μεταξύ των δύο μεθόδων μετατροπής σε γκρί και ποια θεωρείτε ότι είναι πιο αντιπροσωπευτική για την αναπαράσταση της **φωτεινότητας**;

Άσκηση 2: Ανάλυση Μεγέθους Εικόνας και Συμπίεση

Αυτή η άσκηση αναφέρεται στον υπολογισμό του μεγέθους μιας εικόνας και την ανάλυση του αποτελέσματος της συμπίεσης της εικόνας.

- **Δεδομένα Εικόνας:** Η εικόνα έχει διαστάσεις 1024 x 1024 pixels και βάθος χρώματος 24 bits (έγχρωμη εικόνα RGB). Χρησιμοποιήστε τις παρακάτω πληροφορίες για να υπολογίσετε το μέγεθος της εικόνας σε bits, bytes και megabytes.

Χρησιμοποιήστε τους εξής τύπους:

➤ **Bits:** $\text{bits} = M \times N \times \text{bit depth}$

➤ **Bytes:** $\text{bytes} = \frac{\text{bits}}{8}$

➤ **Megabytes:** $\text{MB} = \frac{\text{bytes}}{1024^2}$

- Αν η εικόνα **συμπιεστεί κατά 50%**, υπολογίστε το νέο μέγεθος σε **bytes** και **megabytes**.
- Επαναλάβετε την άσκηση για μία εικόνα με βάθος χρώματος **8 bits** (grayscale εικόνα). Υπολογίστε το **μέγεθος** της εικόνας σε **bytes** και **megabytes** πριν και μετά τη συμπίεση.

Ερωτήσεις:

- a) Ποιο είναι το **μέγεθος** της εικόνας σε **bits**, **bytes** και **megabytes** για την εικόνα RGB (1024 x 1024 pixels με βάθος χρώματος 24 bits);
- b) Αν η εικόνα **συμπίεστεί κατά 50%**, ποιο θα είναι το μέγεθος της εικόνας σε **bytes** και **megabytes**;
- c) **Συγκρίνετε** τα μεγέθη της εικόνας πριν και μετά τη συμπίεση και αναλύστε τα **πλεονεκτήματα** της συμπίεσης εικόνας σε σχέση με το αρχικό μέγεθος.

3. Συμπεράσματα & Προτάσεις για Εμβάθυνση

- Ο φοιτητής **εξοικειώνεται** με την **ανάγνωση**, **ανάλυση** και **επεξεργασία** εικόνων και video, εστιάζοντας στους χρωματικούς χώρους και στα χαρακτηριστικά τους.
- **Εφαρμόζονται** μετατροπές μεταξύ διαφορετικών χρωματικών συστημάτων και τεχνικές φιλτραρίσματος για την ανίχνευση ακμών και την απομόνωση σημαντικών στοιχείων της εικόνας.
- Μέσω projects που ενσωματώνουν επεξεργασία video και τεχνικές computer vision, **κατανοούνται** εφαρμογές για την παρακολούθηση κινούμενων αντικειμένων.

Εργαστήριο 3 – Lab 3

Γεωμετρικοί Μετασχηματισμοί – Γεωμετρικές πράξεις

Σκοπός: Οι φοιτητές να εξοικειωθούν με τις βασικές έννοιες της επεξεργασίας εικόνας, όπως οι γεωμετρικές επιτελέσεις μεταξύ των εικόνων. Επίσης, να μάθουν τις απαραίτητες εντολές στο MATLAB ώστε μπορούν να επεξεργαστούν και να διαχειριστούν εικόνες με αποτελεσματικότητα.

Προσέξτε ότι:

Βεβαιωθείτε ότι έχετε δημιουργήσει έναν φάκελο στον υπολογιστή σας με την ονομασία "Επεξεργασία Εικόνας και Βίντεο", που αντιστοιχεί στο μάθημα. Μέσα σε αυτόν τον κύριο φάκελο, δημιουργήστε έναν υποφάκελο με το ονοματεπώνυμό σας. Αυτός ο υποφάκελος θα χρησιμεύει για την οργάνωση και αποθήκευση όλων των αρχείων .m του MATLAB που θα δημιουργήσετε κατά τη διάρκεια του μαθήματος. Η σωστή οργάνωση των αρχείων σας θα διευκολύνει την πρόσβαση και τη διαχείριση των εργασιών σας.

1. Μαθησιακοί Στόχοι

Με την ολοκλήρωση αυτού του εργαστηρίου ο φοιτητής θα είναι σε θέση να:

- **Κατανοήσει** τις βασικές έννοιες των γεωμετρικών μετασχηματισμών και των γεωμετρικών πράξεων στην επεξεργασία εικόνας, καθώς και τη σχέση τους με τους πίνακες στο MATLAB.
- **Εφαρμόσει** σημειακές λειτουργίες και μετασχηματισμούς (όπως περιστροφή, μετατόπιση, κλίμακα, παραμόρφωση) σε εικόνες, **κατανοώντας** τη διαφορά μεταξύ έγχρωμων και grayscale εικόνων.
- **Αναπτύξει** συναρτήσεις και scripts που **υλοποιούν** μετασχηματισμούς (π.χ. στροφή, αλλαγή διάταξης) και να δει πώς οι πίνακες μπορούν να **χρησιμοποιηθούν** για την αναπαράσταση εικόνων.
- **Συνθέσει** πολλαπλούς γεωμετρικούς μετασχηματισμούς (π.χ. σύνθεση περιστροφής, μετατόπισης, κλίμακας και παραμόρφωσης) για την **αποκατάσταση** ή αλλαγή της εμφάνισης των εικόνων.
- **Επεκτείνει** τις βασικές γνώσεις με εφαρμογές σε πιο προχωρημένες τεχνικές όπως ο affine μετασχηματισμός, ώστε να **κατανοήσει** και να **αποκαταστήσει** γεωμετρικές παραμορφώσεις.

2. Πώς το Μαθαίνουν

- **Θεωρητική Παρουσίαση & Παραδείγματα:** Στην εισαγωγή κάθε ενότητας, **παρέχονται** θεωρητικές επεξηγήσεις με παραδείγματα που δείχνουν την αναπαράσταση των εικόνων ως πίνακες και τις βασικές εντολές για γεωμετρικούς μετασχηματισμούς.
- **Πρακτική Εφαρμογή:** Μέσω μιας σειράς εργαστηριακών ασκήσεων, ο φοιτητής **υλοποιεί** κώδικα στο MATLAB, **πειραματίζεται** με μετασχηματισμούς, **αλλάζει** παραμέτρους και **παρατηρεί** τις επιπτώσεις στις εικόνες.
- **Συγκριτική Ανάλυση & Ερωτήσεις Κατανόησης:** Με τη βοήθεια ερωτήσεων, **αναλύει** τις διαφορές μεταξύ των αποτελεσμάτων πριν και μετά τους μετασχηματισμούς, **εμβαθύνοντας** στην κατανόηση των γεωμετρικών αλλαγών.
- **Ανάπτυξη Projects:** Ο φοιτητής θα **υλοποιήσει** ολοκληρωμένα projects που **συνδυάζουν** διάφορους μετασχηματισμούς, προκειμένου να **αναδείξει** την ικανότητά του να **συνθέτει** σύνθετους μετασχηματισμούς και να **επιλύει** προβλήματα πραγματικού κόσμου (π.χ. διόρθωση παραμορφώσεων).

3. Βασικές Έννοιες και Εντολές

- **Εικόνες ως Πίνακες:** Οι εικόνες αναπαρίστανται στο MATLAB ως πίνακες. Για παράδειγμα, μια grayscale εικόνα είναι ένας πίνακας $N \times M$, ενώ μια έγχρωμη εικόνα (RGB) έχει διαστάσεις $M \times N \times 3$.
- **Σημειακές Λειτουργίες:** Εφαρμογή λειτουργιών σε κάθε pixel ξεχωριστά (π.χ. πολλαπλασιασμός με σταθερό αριθμό για αύξηση της φωτεινότητας).
- **Γεωμετρικοί Μετασχηματισμοί:** Οι βασικοί μετασχηματισμοί περιλαμβάνουν:
- **Περιστροφή:** Μετατόπιση και αλλαγή γωνίας (χρησιμοποίηση συναρτήσεων όπως `imwarp`).
- **Μετατόπιση:** Μετακίνηση εικόνας κατά ορισμένα pixels.
- **Κλίμακα (Scaling):** Αλλαγή μεγέθους της εικόνας.
- **Παραμόρφωση (Affine):** Σύνθετοι μετασχηματισμοί που συνδυάζουν περιστροφή, κλίμακα και μετατόπιση.
- **Συναρτήσεις MATLAB:** Χρήση εντολών όπως `imread`, `imshow`, `imwrite`, `imresize`, `imwarp` και `fspecial` για τη δημιουργία φίλτρων.

1. Εισαγωγή στη Δημιουργία Τυχαίων Εικόνων και Σημειακές Λειτουργίες

Στόχος άσκησης: Οι φοιτητές θα κατανοήσουν τη χρήση της εντολής **randi** για τη δημιουργία τυχαίων εικόνων, θα μάθουν την έννοια των σημειακών λειτουργιών και πώς επηρεάζουν την εικόνα, και θα αναγνωρίσουν τις διαφορές μεταξύ έγχρωμων και γκρί-κλίμακας εικόνας, χρησιμοποιώντας τη συνάρτηση **rgb2gray**.

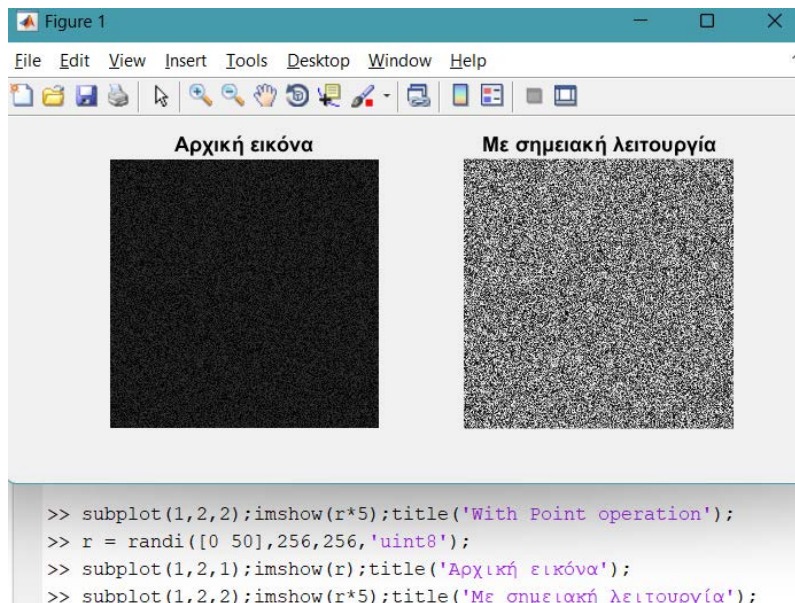
Μια από τις εντολές που θα χρησιμοποιήσουμε είναι η ακόλουθη:

```
r = randi([0 255],512,512,'uint8');
```

Άσκηση για εξάσκηση

- Εκτέλεση του παρακάτω κώδικα για δημιουργία τυχαίας εικόνας και παρατήρηση των αποτελεσμάτων.

```
r = randi([0 50],256,256,'uint8');  
subplot(1,2,1);imshow(r);title('Αρχική εικόνα');  
subplot(1,2,2);imshow(r*5);title('Με σημειακή λειτουργία');
```



- Τι συμβαίνει όταν εκτελείτε τον κώδικα;
- Πώς επηρεάζει η λειτουργία `randi([0 50],256,256,'uint8')` την εικόνα;
- Ποιες είναι οι **διαφορές** ανάμεσα στην **αρχική** εικόνα και στην **επεξεργασμένη** με την σημειακή λειτουργία `r*5`; Ποιο είναι το **αποτέλεσμα** της συνάρτησης αυτής;

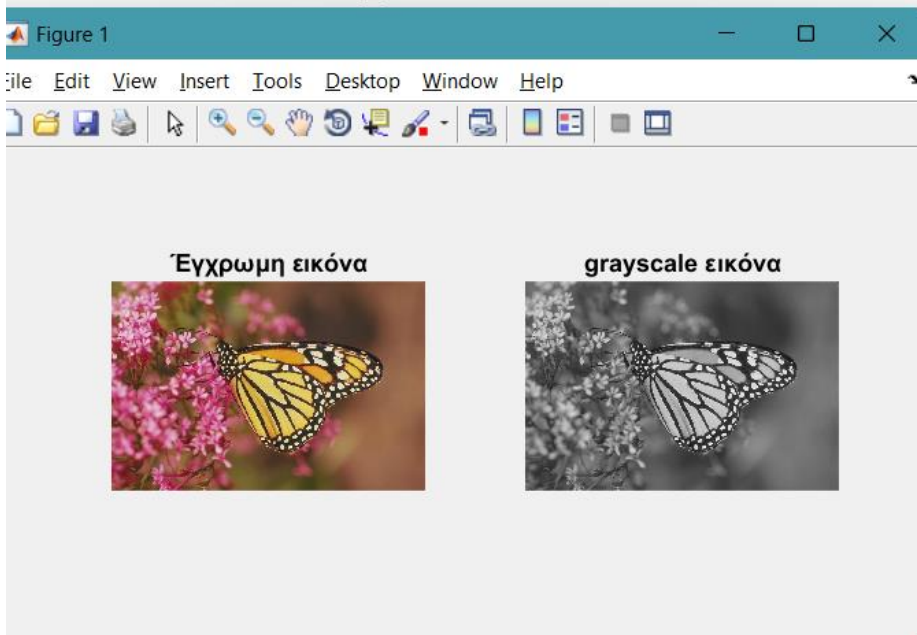
- **Χρήση έγχρωμης εικόνας monarch.jpg**

Κάνετε χρήση την έγχρωμη εικόνα ονόματος monarch. Θα τη βρείτε στο ακόλουθο link <https://links.uwaterloo.ca/Repository.html>. Τι συμπεράσματα βγάζετε από τον παρακάτω κώδικα;

```
I_color = imread('monarch.jpg');
I_gray = rgb2gray(I_color);
subplot(1,2,1);imshow(I_color);title('Έγχρωμη εικόνα');
subplot(1,2,2);imshow(I_gray);title('grayscale εικόνα');
```

- Ποιες **διαφορές** παρατηρείτε μεταξύ της **έγχρωμης** εικόνας και της **grayscale** εικόνας;
- Τι συμβαίνει όταν χρησιμοποιούμε τη συνάρτηση `rgb2gray`;
- Ποια είναι η **χρησιμότητα** της μετατροπής μιας έγχρωμης εικόνας σε γκρι-κλίμακα;

```
>> I_color = imread('monarch.jpg');
>> I_gray = rgb2gray(I_color);
>> subplot(1,2,1);imshow(I_color);title('Έγχρωμη εικόνα');
>> subplot(1,2,2);imshow(I_gray);title('grayscale εικόνα');
```



2. Άσκηση κατανόησης: Δημιουργία Τυχαίας Εικόνας

Στόχος άσκησης: Κατανόηση της δημιουργίας τυχαίων εικόνων με την εντολή `randi` και εφαρμογή σημειακής λειτουργίας.

- Δημιουργήστε μια **τυχαία** εικόνα μεγέθους **128x128 pixels**, όπου οι τιμές των pixels θα κυμαίνονται από **50** έως **150**, χρησιμοποιώντας την εντολή **`randi`**.
- Εφαρμόστε την σημειακή λειτουργία πολλαπλασιάζοντας την εικόνα με **3**.
- Εμφανίστε την αρχική και την επεξεργασμένη εικόνα χρησιμοποιώντας την εντολή **`subplot`**.

```
r = randi([50 150], 128, 128, 'uint8');
subplot(1,2,1); imshow(r); title('Αρχική Εικόνα');
subplot(1,2,2); imshow(r*3); title('Με Σημειακή Λει-
τουργία');
```

- Τι παρατηρείτε στην εικόνα **πριν** και **μετά** την εφαρμογή της σημειακής λειτουργίας;
- Τι συμβαίνει όταν πολλαπλασιάζετε την εικόνα με **3**;

3. Άσκηση κατανόησης: Μετατροπή Έγχρωμης Εικόνας σε Γκρί-Κλίμακα

Στόχος άσκησης: Κατανόηση της χρήσης της συνάρτησης `rgb2gray` και αναγνώριση των διαφορών μεταξύ έγχρωμης και γκρί-κλίμακας εικόνας.

- Κατεβάστε την εικόνα 'monarch.jpg' από το link: <https://links.uwaterloo.ca/Repository.html>
- Φορτώστε την εικόνα στην MATLAB και μετατρέψτε την σε γκρί-κλίμακα χρησιμοποιώντας τη συνάρτηση **rgb2gray**.
- Εμφανίστε την **έγχρωμη** και την **grayscale** εικόνα χρησιμοποιώντας **subplot**.

```
I_color = imread('monarch.jpg');  
I_gray = rgb2gray(I_color);  
subplot(1,2,1); imshow(I_color); title('Έγχρωμη Εικό-  
να');  
subplot(1,2,2); imshow(I_gray); title('Grayscale Εικό-  
να');
```

- Ποιες **διαφορές** παρατηρείτε μεταξύ της **έγχρωμης** και της **grayscale** εικόνας;
- Πώς επηρεάζει η συνάρτηση **rgb2gray** την εικόνα;

4. Επεξεργασία Εικόνας με Πίνακες στην MATLAB

Στο MATLAB, οι εικόνες είναι πίνακες όπου κάθε στοιχείο αντιστοιχεί σε ένα pixel. Οι σημειακές λειτουργίες εφαρμόζονται σε κάθε pixel ξεχωριστά, επιτρέποντας την τροποποίηση της έντασης ή των χρωμάτων.

- **Σημειακές Λειτουργίες**

Οι σημειακές λειτουργίες **εφαρμόζονται** σε κάθε pixel **ξεχωριστά**. Με τη χρήση αυτών των λειτουργιών, μπορούμε να τροποποιήσουμε την **ένταση** ή τις **χρωματικές τιμές** των pixels. Για παράδειγμα, πολλαπλασιάζοντας κάθε pixel με έναν αριθμό αυξάνουμε την έντασή του.

- **Στροφή Πίνακα**

Η **στροφή** ενός πίνακα αναστρέφει τη **διάταξη** του, μετατοπίζοντας τις τιμές του κατά **90** μοίρες ή με άλλες γωνίες. Στην επεξεργασία εικόνας, αυτό χρησιμοποιείται για να περιστρέψουμε ή να αλλάξουμε την κατεύθυνση της εικόνας.

Στόχος άσκησης: Η δημιουργία και επεξεργασία ενός πίνακα **5x5** για αναπαράσταση εικόνας, η υλοποίηση μιας συνάρτησης για τη στροφή της εικόνας και η κατανόηση του τρόπου με τον οποίο οι πίνακες χρησιμοποιούνται για την αποτύπωση εικόνας στο MATLAB. Οι φοιτητές θα μάθουν πώς να εφαρμόζουν λειτουργίες επεξεργασίας, όπως η στροφή των δεδομένων, και θα κατανοήσουν τον ρόλο των πινάκων στην επεξεργασία εικόνας.

5. Άσκηση για εξάσκηση

- Δημιουργήστε έναν πίνακα 5x5 που να αναπαριστά την παρακάτω εικόνα, όπου τα 1 αναπαριστούν το "άσπρο" και τα 0 το "μαύρο":

```
A=zeros(5,5);
```

```
A(1,2)=1; A(2,3:5)=1; A(4,4)=1; A(5,5)=1;
```

```
0  1  0  0  0
0  0  1  1  1
0  0  0  0  0
0  0  0  1  0
0  0  0  0  1
```

- Δημιουργία νέου script:** Δημιουργήστε ένα νέο script το οποίο θα ονομάσετε **lab_3_script.m**. Στο script αυτό, θα υλοποιήσετε τον παρακάτω κώδικα για να μετατρέψετε την εικόνα σε μια νέα διάταξη μέσω της στροφής της:

➤ Η συνάρτηση **shiftright(X)** παίρνει έναν πίνακα **X** και επιστρέφει έναν νέο πίνακα **Y** που είναι μια στροφή της εικόνας. Στη συγκεκριμένη περίπτωση, κάθε **στήλη** του πίνακα **X** γίνεται **γραμμή** του πίνακα **Y** και αντίστροφα.

```
X = [0 1 0 0 0; 0 0 1 1 1; 0 0 0 0 0; 0 0 0 1 0; 0 0 0
0 1];
Y = shiftright(X);
disp(Y);
function [Y] = shiftright(X)
for i = 1:size(X,1)
```

```

for j = 1:size(X,2)
    Y(i,size(X,1) + 1 - j) = X(j,i);
end
end
end

```

Αρχικός πίνα-

Στριμμένος πίνα-

0	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0	1	0
0	0	0	0	1	1	0	0	1	0

- Τι παρατηρείτε στην εικόνα **πριν** και **μετά** την εφαρμογή της στροφής;
- Ποια είναι η βασική λειτουργία που εκτελεί η συνάρτηση **shiftright(X)**;
- Ποιες άλλες λειτουργίες επεξεργασίας εικόνας θα μπορούσατε να εφαρμόσετε σε τέτοιες εικόνες χρησιμοποιώντας **παρόμοιες** τεχνικές;

6. Άσκηση κατανόησης: Δημιουργία Πίνακα Εικόνας

- Δημιουργήστε έναν πίνακα 3x3, με 1 στα κεντρικά και τα γωνιακά στοιχεία, και 0 παντού αλλού.

```
A = [1 0 1; 0 1 0; 1 0 1];
```

```
disp(A);
```

- Ποια είναι η **διάταξη** του πίνακα A και πώς αντιστοιχεί στην εικόνα;

7. Άσκηση κατανόησης: Στροφή Πίνακα 3x3

- Δημιουργήστε έναν πίνακα 3x3 και εφαρμόστε μια συνάρτηση για να τον περιστρέψετε κατά 90 μοίρες δεξιόστροφα.

```
X = [1 0 0; 0 1 0; 0 0 1];
```

```
Y = shiftright(X);
```

```
disp(Y);
```

- Τι παρατηρείτε στην εικόνα μετά την **περιστροφή** του πίνακα κατά **90** μοίρες; Ποιες είναι οι θέσεις των **1s** μετά την **περιστροφή**;

8. Άσκηση κατανόησης: Αντίστροφη Στροφή Πίνακα

- Δημιουργήστε έναν πίνακα 4x4 και εφαρμόστε αντίστροφη στροφή (180 μοίρες) της εικόνας.

```
X = [1 0 0 1; 0 1 1 0; 0 1 1 0; 1 0 0 1];
Y = rot90(X, 2);
disp(Y);
```

- Ποιες είναι οι **αλλαγές** στην εικόνα όταν εφαρμόζεται στροφή **180** μοιρών στον πίνακα;

1. Γεωμετρική Επεξεργασία Εικόνας με Affine Μετασχηματισμό

Στόχος άσκησης: Εφαρμογή του Affine μετασχηματισμού σε μια εικόνα για την αποκατάσταση γεωμετρικών παραμορφώσεων, επεξεργασία εικόνας με χρήση φίλτρων για την αποκατάσταση χαρακτηριστικών φωτεινότητας και εξοικείωση με τη χρήση πινάκων και μετασχηματισμών στην MATLAB.

- Αποθηκεύστε την εικόνα με όνομα `frog` από το παρακάτω link:
<https://links.uwaterloo.ca/Repository.html>
- Φορτώστε την εικόνα στο MATLAB:

```
I = imread('frog.jpg');
imshow(I);
title('Αρχική Εικόνα');
```
- Για να εφαρμόσετε μια στροφή στην εικόνα χρησιμοποιώντας την ίδια συνάρτηση που είδαμε στην προηγούμενη εργασία, χρησιμοποιήστε την παρακάτω συνάρτηση **affineTransform** για την εφαρμογή ενός Affine μετασχηματισμού.

```
tform = affine2d([1 0 0; 0 1 0; 0.1 0.1 1]);
I_transformed = imwarp(I, tform);
subplot(1,2,1);
imshow(I);
title('Αρχική Εικόνα');
subplot(1,2,2);
imshow(I_transformed);
title('Εικόνα μετά τον Affine Μετασχηματισμό');
```
- Αν θέλουμε να βελτιώσουμε ή να αποκαταστήσουμε τα χαρακτηριστικά **φωτεινότητας** της εικόνας, μπορούμε να χρησιμοποιήσουμε φίλτρα όπως το **φίλτρο άμβλυνσης** ή φίλτρα για αύξηση της ευκρίνειας. Εδώ, θα χρησιμοποιήσουμε ένα απλό φίλτρο άμβλυνσης για την εικόνα.

```
h = fspecial('average', [5 5]);
```

```

I_blurred = imfilter(I, h);
figure;
subplot(1,2,1);
imshow(I);
title('Αρχική Εικόνα');
subplot(1,2,2);
imshow(I_blurred);
title('Εικόνα με Φίλτρο Αμβλυνσης');

```

- Πώς επηρεάζει ο **Affine** μετασχηματισμός την εμφάνιση της εικόνας; Παρατηρήσατε κάποια γεωμετρική **παραμόρφωση**;
- Ποιες είναι οι αλλαγές στην εικόνα **μετά** την εφαρμογή του φίλτρου αμβλυνσης; Πώς επηρεάζει τη **φωτεινότητα** και την **ευκρίνεια**;
- Τι άλλο μπορείτε να προσθέσετε στον **Affine** μετασχηματισμό για να αποκαταστήσετε άλλες **παραμορφώσεις** (π.χ. κλίμακα ή περιστροφή);



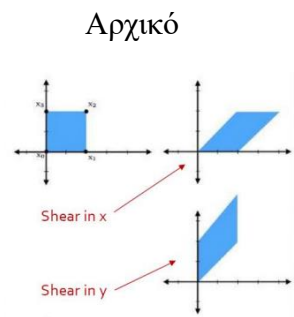
Εκτός από την επεξεργασία της εικόνας με φίλτρα για τη αναβάθμιση ή αποκατάσταση των χαρακτηριστικών φωτεινότητας, εξίσου δηλωτική είναι και η αποκατάσταση των γεωμετρικών χαρακτηριστικών της. Μια εικόνα δύναται να είναι παραποιημένη από τη στιγμή της θεμελίωσης της. Σκεφτείτε ένα έγγραφο που έχει σκαναριστεί υπό μια μικρή γωνία ή μια φωτογραφία όπου τα αντικείμενα είναι παραμορφωμένα λόγω προοπτικής. Ο πιο αναγνωρισμένος μετασχηματισμός για την αποκατάσταση των γεωμετρικών παραμορφώσεων της εικόνας είναι ο μετασχηματισμός

Affine. Αυτός ο μετασχηματισμός αφήνει την μεταρρύθμιση των συντεταγμένων της αρχικής εικόνας μέσω της χρήσης πινάκων. Οι πίνακες που θα φανούν χρήσιμοι είναι οι εξής:

Ταυτότητα (Identity)	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$
Κατοπτρισμός (mirroring) με βάση τον οριζόντιο άξονα	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$
Κατοπτρισμός (mirroring) με βάση τον κατακόρυφο άξονα	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$

Κλιμάκωση (Scaling)	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$	S_x : Πολλαπλασιάζει το μέγεθος στην οριζόντια διάσταση S_y : Πολλαπλασιάζει το μέγεθος στην κατακόρυφη διάσταση
Περιστροφή (Rotation)	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$	θ : περιστροφή σε μοίρες
Μετατόπιση (Translation)	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$	t_x : απόσταση στον οριζόντιο άξονα σε pixels t_y : απόσταση στον κατακόρυφο άξονα σε pixels

Παραμόρφωση (Shear) στον κατακόρυφο άξονα	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$	Sh_x : Εκφράζει το μέγεθος της παραμόρφωσης στην οριζόντια διάσταση
Παραμόρφωση (Shear) στον οριζόντιο άξονα	$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$	Sh_y : Εκφράζει το μέγεθος της παραμόρφωσης στην κατακόρυφη διάσταση



2. Εφαρμογή Πολλαπλών Μετασχηματισμών σε Εικόνα

Τι γίνεται όταν χρειάζεται να εφαρμοστούν πολλαπλοί μετασχηματισμοί; Σε αυτήν την περίπτωση, μπορούμε να πολλαπλασιάσουμε τους πίνακες των επιθυμητών γεωμετρικών μετασχηματισμών.

Στόχος άσκησης: Η άσκηση στοχεύει στην κατανόηση και εφαρμογή του τύπου $\mathbf{p}' = (\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S}) \cdot \mathbf{p}$ για τη σύνθεση πολλαπλών γεωμετρικών μετασχηματισμών, όπως είναι η περιστροφή, η κλίση και η μετατόπιση, και στην εφαρμογή τους σε μια εικόνα. Οι φοιτητές θα εξοικειωθούν με την έννοια του συνδυασμού αυτών των μετασχηματισμών μέσω της χρήσης πινάκων και θα μάθουν πώς να εφαρμόζουν σύνθετους γεωμετρικούς μετασχηματισμούς στην MATLAB, χρησιμοποιώντας τον πολλαπλασιασμό των αντίστοιχων πινάκων μετασχηματισμού.

Όταν εφαρμόζουμε πολλαπλούς γεωμετρικούς μετασχηματισμούς σε μια εικόνα, μπορούμε να τους συνδυάσουμε χρησιμοποιώντας τον τύπο: $\mathbf{p}' = (\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S}) \cdot \mathbf{p}$

Όπου,

- $\mathbf{p}$ είναι το σημείο της εικόνας πριν τον μετασχηματισμό.
- $\mathbf{T}, \mathbf{R}, \mathbf{S}$ είναι οι πίνακες μετασχηματισμού για τη μετατόπιση, περιστροφή και κλίση αντίστοιχα.
- $\mathbf{p}'$ είναι το σημείο της εικόνας μετά τον εφαρμοσμένο συνδυασμένο μετασχηματισμό.

$$\begin{aligned} \mathbf{p}' = (\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S}) \cdot \mathbf{p} &= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}' & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{S}' & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}'\mathbf{S}' & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{aligned}$$

- Αποθηκεύστε την εικόνα με όνομα frog από το παρακάτω link:
<https://links.uwaterloo.ca/Repository.html>
`I = imread('frog.jpg');`
`imshow(I);`

```
title('Αρχική Εικόνα');
```

- Ορίζουμε τρεις διαφορετικούς μετασχηματισμούς:
 - **Μετασχηματισμός Κλίσης:** $S = \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$;
 - **Μετασχηματισμός Περιστροφής:** $R = \begin{bmatrix} \cos(45) & -\sin(45) & 0 \\ \sin(45) & \cos(45) & 0 \\ 0 & 0 & 1 \end{bmatrix}$;
 - **Μετασχηματισμός Μετατόπισης:** $T = \begin{bmatrix} 1 & 0 & 50 \\ 0 & 1 & 30 \\ 0 & 0 & 1 \end{bmatrix}$;
- Χρησιμοποιούμε την **imwarp** για να εφαρμόσουμε τον συνδυασμένο μετασχηματισμό στην εικόνα:

```
I_transformed = imwarp(I, affine2d(M));  
subplot(1,2,1);  
imshow(I);  
title('Αρχική Εικόνα');  
subplot(1,2,2);  
imshow(I_transformed);  
title('Εικόνα με Πολλαπλούς Μετασχηματισμούς');
```

 - Ποιες είναι οι **αλλαγές** στην εικόνα **μετά** την εφαρμογή των **τριών** μετασχηματισμών (κλίση, περιστροφή, και μετατόπιση);
 - Ποιες **διαφορές** παρατηρείτε όταν εφαρμόζετε τους μετασχηματισμούς;

3. Άσκηση κατανόησης: Εφαρμογή Σύνθετου Μετασχηματισμού με Περιστροφή και Μετατόπιση

Στόχος άσκησης: Να κατανοήσετε τη σύνθεση μετασχηματισμών και να παρατηρήσετε την επίδραση της περιστροφής και της μετατόπισης σε μια εικόνα.

- Φορτώστε την εικόνα **frog.jpg**
- Εφαρμόστε πρώτα έναν **μετασχηματισμό περιστροφής** (rotation) κατά 60 μοίρες.
- Στη συνέχεια, εφαρμόστε έναν **μετασχηματισμό μετατόπισης** (translation) κατά **100 pixels** προς τα **δεξιά** και **50 pixels** προς τα **κάτω**.
- Σχηματίστε τον **τελικό** πίνακα μετασχηματισμού $M=T \cdot R$ και εφαρμόστε τον στην εικόνα χρησιμοποιώντας τη συνάρτηση **imwarp**.

```
I = imread('frog.jpg');  
imshow(I);  
title('Αρχική Εικόνα');
```

```

R = [cosd(60) -sind(60) 0; sind(60) cosd(60) 0; 0 0 1];
I_transformed = imwarp(I, affine2d(M));
subplot(1,2,1);
imshow(I);
title('Αρχική Εικόνα');
subplot(1,2,2);
imshow(I_transformed);
title('Εικόνα μετά από Περιστροφή και Μετατόπιση');

```

- Πώς η περιστροφή και η μετατόπιση επηρεάζουν τη θέση και την εμφάνιση των αντικειμένων στην εικόνα;
- Τι παρατηρείτε όταν αλλάζετε την κατεύθυνση της μετατόπισης ή τη γωνία περιστροφής;

4. Άσκηση κατανόησης: Σύνθεση Κλίσης, Περιστροφής και Μετατόπισης

Στόχος άσκησης: Να εξετάσετε την επίδραση της εφαρμογής τριών μετασχηματισμών (κλίση, περιστροφή και μετατόπιση) σε μία εικόνα.

- Φορτώστε την εικόνα **monarch.jpg**
- Εφαρμόστε έναν μετασχηματισμό **κλίσης** στην εικόνα, κατά **0.3** στην x κατεύθυνση.
- Εφαρμόστε έναν μετασχηματισμό **περιστροφής** κατά **45 μοίρες**.
- Τέλος, εφαρμόστε έναν **μετασχηματισμό μετατόπισης** κατά 50 pixels προς τα αριστερά και **30** pixels προς τα **πάνω**.
- Σχηματίστε τον **τελικό** πίνακα μετασχηματισμού $M=T \cdot R \cdot S$ και εφαρμόστε τον στην εικόνα με τη συνάρτηση **imwarp**.

```

I = imread('monarch.jpg');
imshow(I);
title('Αρχική Εικόνα');
S = [1 0.3 0; 0 1 0; 0 0 1];
R = [cosd(45) -sind(45) 0; sind(45) cosd(45) 0; 0 0 1];
M = T * R * S;
I_transformed = imwarp(I, affine2d(M));

```

```
subplot(1,2,1);
imshow(I);
title('Αρχική Εικόνα');
subplot(1,2,2);
imshow(I_transformed);
title('Εικόνα με Κλίση, Περιστροφή και Μετατόπιση');
```

- Ποιες είναι οι **διαφορές** στην εικόνα όταν εφαρμόζετε **έναν** μόνο μετασχηματισμό σε σχέση με την εφαρμογή όλων των μετασχηματισμών μαζί;
- Πώς επηρεάζει η **κλίση** την εμφάνιση της εικόνας σε σχέση με την **περιστροφή** και τη **μετατόπιση**;

5. Άσκηση εξάσκησης

Παρακαλώ γράψτε κώδικα στην MATLAB για τη δημιουργία των πινάκων Affine για, μετατόπιση κατοπτρισμό, κλιμάκωση, περιστροφή και παραμόρφωση.

- Περιστροφή **47** μοιρών

$$R = \begin{bmatrix} \cos(47^\circ) & \sin(47^\circ) & 0 \\ -\sin(47^\circ) & \cos(47^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Σμίκρυνση κατά **50%** και προς τις δυο διευθύνσεις x,y

$$S_c = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Μετατόπιση κατακόρυφα κατά **-300** pixel και οριζόντια κατά **400** pixel

$$T = \begin{bmatrix} 1 & 0 & -300 \\ 0 & 1 & 400 \\ 0 & 0 & 1 \end{bmatrix}$$

- Παραμόρφωση κατά 0,7

$$Sh = \begin{bmatrix} 1 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ο τελικός πίνακας έπειτα από τους μετασχηματισμούς είναι ο εξής: **Trasform=T*R*Sc*Sh**

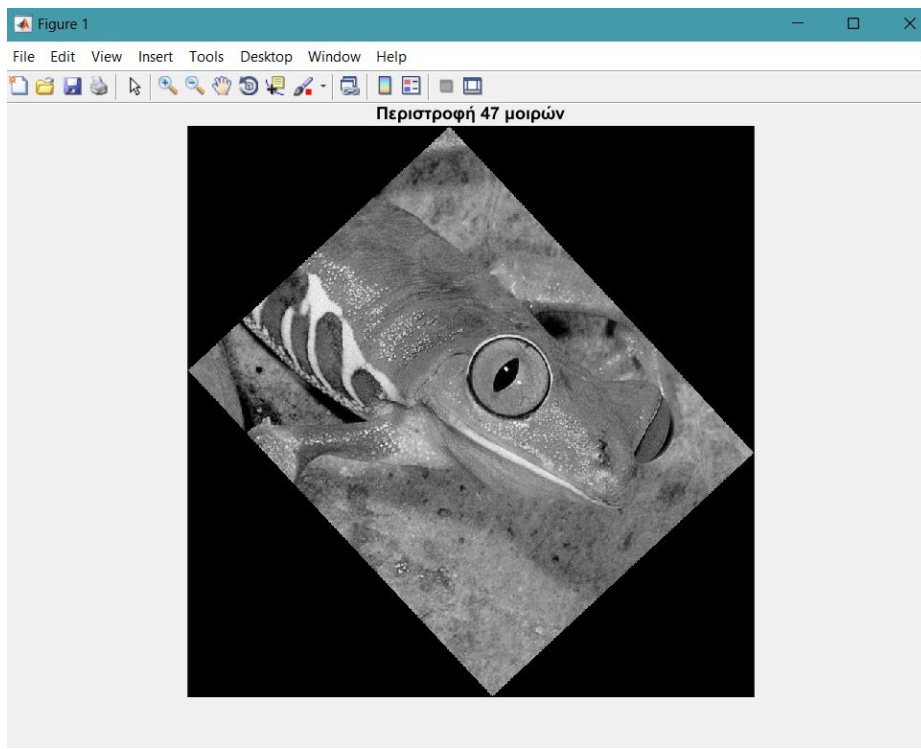
Trasform =

```
-0.2718    0.0371 -300.0000
-0.2455   -0.2977  400.0000
      0           0    1.0000
```

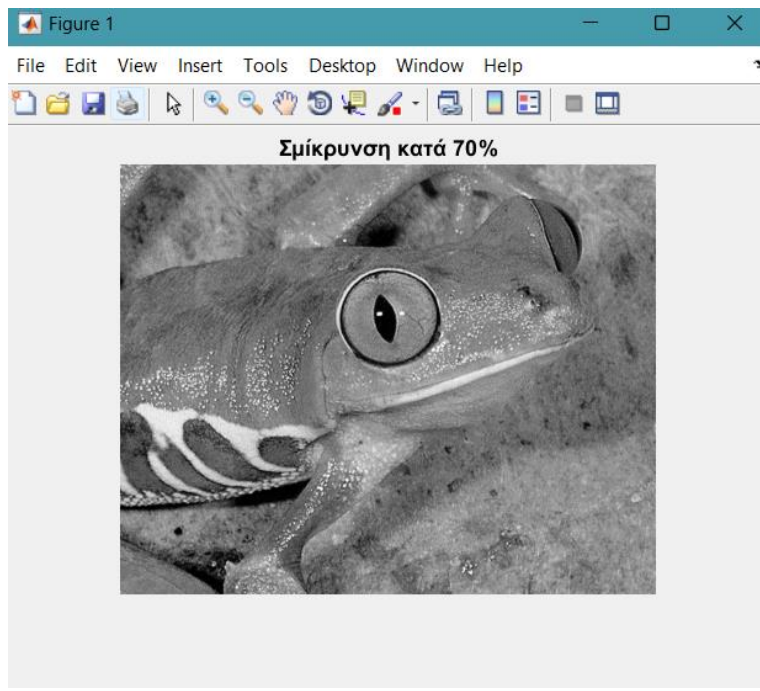
6. Άσκηση εξάσκησης

Φορτώστε στο MATLAB την εικόνα με την ονομασία **'frog.tif'** στο workspace και προβάλλετε την σε ένα καινούργιο παράθυρο. Εφαρμόστε στη σειρά τις παρακάτω λειτουργίες χρησιμοποιώντας τους αντίστοιχους πίνακες μετασχηματισμού:

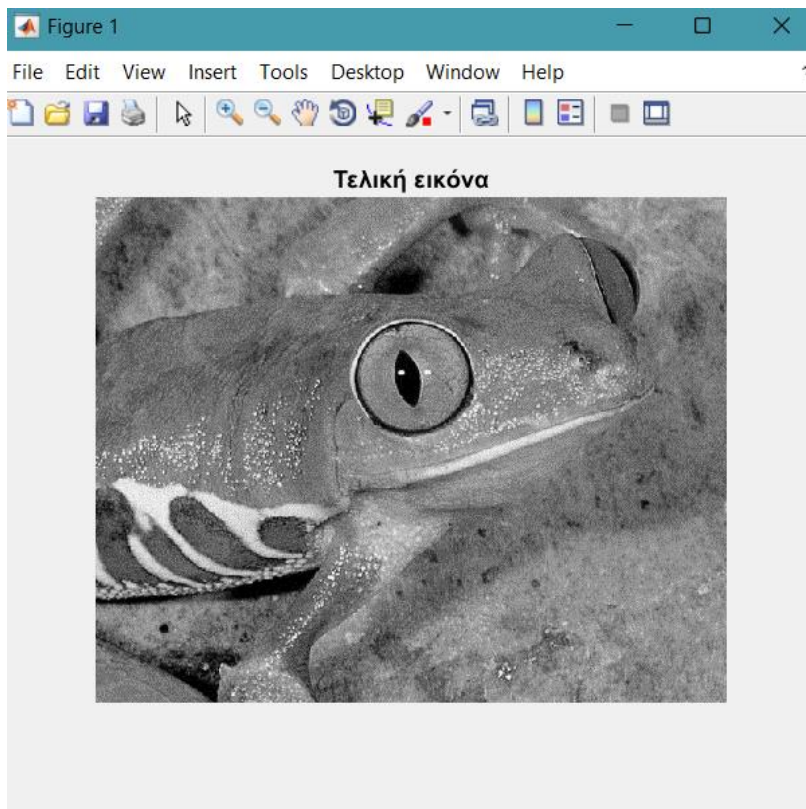
- Περιστροφή 47 μοιρών
- **Σμίκρυνση** κατά **70%** και προς τις δυο διευθύνσεις x,y
- Μετατόπιση **κατακόρυφα** κατά **-300 pixel** και οριζόντια κατά **400 pixel**
- Παραμόρφωση κατά **0,7**
- Όλα τα παραπάνω μαζί
`I = imread('frog.tif');`
`angle_rad = deg2rad(47);`
`rotation_matrix = [cos(angle_rad) sin(angle_rad) 0; -`
`sin(angle_rad) cos(angle_rad) 0; 0 0 1];`
`I_rotated = imwarp(I, affine2d(rotation_matrix));`
`figure, imshow(I_rotated), title('Περιστροφή 47 μοι-`
`ρών');`



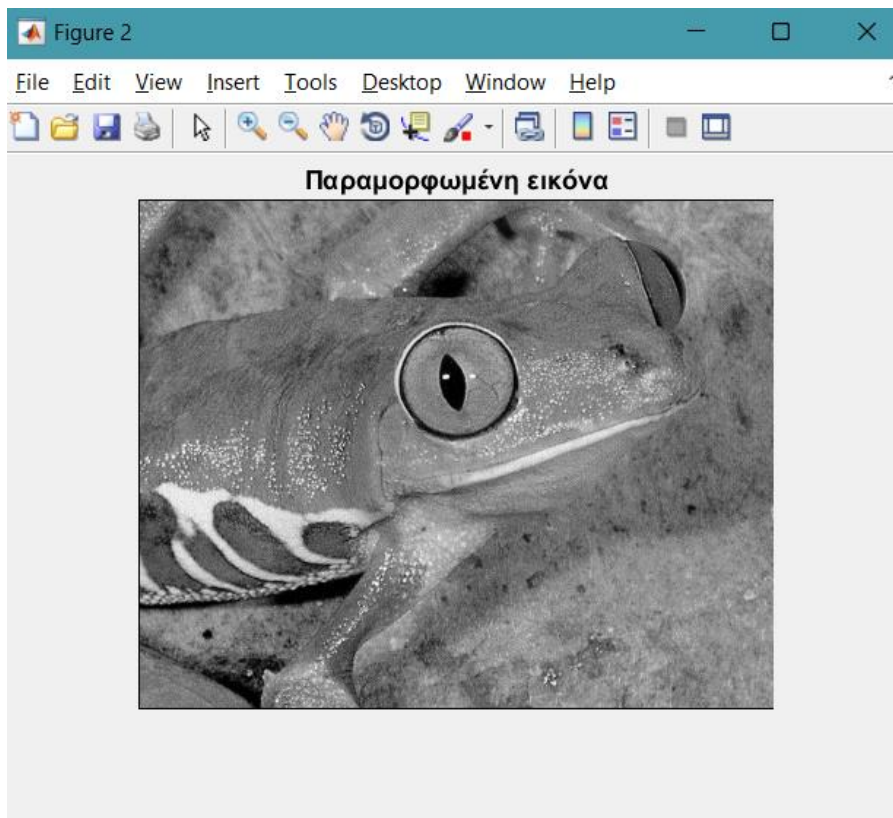
```
scaleX = 0.7;  
scaleY = 0.7;  
scaling_matrix = [scaleX 0 0; 0 scaleY 0; 0 0 1];  
I_scaled = imresize(I, [round(size(I,1)*scaleY)  
round(size(I,2)*scaleX)]);  
figure, imshow(I_scaled), title('Σμίκρυνση κατά  
70%');
```



```
I=imread('frog.tif');  
T = [1 0 0; 0 1 0; 400 -300 1];  
tform = affine2d(T);  
outputImage = imwarp(I, tform);  
figure;  
imshow(outputImage);title('Τελική εικόνα');
```



```
I= imread('frog.tif');  
S = [0.7 0 0; 0 0.7 0; 0 0 1];  
tform = affine2d(S);  
outputImage = imwarp(I, tform);  
figure;  
imshow(outputImage);  
title('Παραμορφωμένη εικόνα');
```



7. Άσκηση προς επίλυση: Συνδυασμός Κλίσης, Περιστροφής και Μετατόπισης

- Φορτώστε την εικόνα **frog.tif** και πραγματοποιήστε τους εξής μετασχηματισμούς:
 - **Περιστροφή** της εικόνας κατά **60** μοίρες.
 - **Σμίκρυνση** κατά **70%** στις διαστάσεις x και y.
 - **Παραμόρφωση** κατά **0.5** στην κατεύθυνση x.
 - **Μετατόπιση** της εικόνας κατά **100 pixels κατακόρυφα** και **150 pixels οριζόντια**.
 - Προβάλλετε την εικόνα **μετά** από κάθε μετασχηματισμό.
 - Στη συνέχεια, συνθέστε όλους τους παραπάνω μετασχηματισμούς σε έναν και εφαρμόστε τον στην εικόνα.

8. Άσκηση προς επίλυση: Σύνθεση Πολύπλοκων Μετασχηματισμών

- Φορτώστε την εικόνα **frog.tif** και πραγματοποιήστε τους εξής μετασχηματισμούς:
 - **Περιστροφή** της εικόνας κατά **45** μοίρες γύρω από το κέντρο της εικόνας.
 - **Σμίκρυνση** της εικόνας κατά **50%** στις διαστάσεις x και y.
 - **Παραμόρφωση** με παράγοντα **0.4** στην κατεύθυνση x και **0.6** στην κατεύθυνση y.
 - **Μετατόπιση** της εικόνας κατά **300 pixels κατακόρυφα** και **-250 pixels οριζόντια**.

Στη συνέχεια, εφαρμόστε τα εξής:

- a) **Αλλαγή κλίμακας** με συντελεστή **2** στην κατεύθυνση x και **1.5** στην κατεύθυνση y .
- b) **Περιστροφή** κατά **-90** μοίρες γύρω από το κέντρο της εικόνας.
 - Συνθέστε όλους αυτούς τους μετασχηματισμούς σε έναν ενιαίο πίνακα μετασχηματισμού και εφαρμόστε τον στην εικόνα.
 - Προβάλλετε την εικόνα μετά από κάθε μετασχηματισμό, καθώς και την τελική εικόνα **μετά** την σύνθεση.

4. Συμπεράσματα και Προτάσεις για Εμβάθυνση

- Ο φοιτητής έμαθε να αναπαριστά εικόνες ως πίνακες και να εφαρμόζει σημειακές λειτουργίες για την τροποποίηση της έντασης ή του χρώματος.
- Μέσα από τις ασκήσεις, εξοικειώθηκε με βασικές και σύνθετες γεωμετρικές πράξεις (περιστροφή, μετατόπιση, κλίμακα, παραμόρφωση) και έμαθε να συνθέτει πολλαπλούς μετασχηματισμούς.
- Η πρακτική εφαρμογή με τη χρήση συναρτήσεων όπως `imwarp` και `affine2d` επιτρέπει την κατανόηση της γεωμετρικής επεξεργασίας και της αποκατάστασης παραμορφώσεων.

Εργαστήριο 4 – Lab 4

Εξομάλυνση Εικόνας - Ιστόγραμμα - Θόρυβος - Φίλτρα

Σκοπός:

Σε αυτό το εργαστήριο, οι φοιτητές θα εμβαθύνουν σε βασικές τεχνικές της επεξεργασίας εικόνας, όπως η εξομάλυνση εικόνας, η ανάλυση ιστογράμματος, η αφαίρεση θορύβου και η εφαρμογή φίλτρων για την ενίσχυση ή τη βελτίωση μιας εικόνας. Οι τεχνικές αυτές αποτελούν βασικά εργαλεία στην επεξεργασία εικόνας και έχουν εφαρμογή σε τομείς όπως η ιατρική απεικόνιση, η αναγνώριση προσώπων και η βελτίωση ποιότητας εικόνας σε φωτογραφίες και βίντεο.

Προσέξτε ότι:

Βεβαιωθείτε ότι έχετε δημιουργήσει έναν φάκελο στον υπολογιστή σας με την ονομασία "Επεξεργασία Εικόνας και Βίντεο", που αντιστοιχεί στο μάθημα. Μέσα σε αυτόν τον κύριο φάκελο, δημιουργήστε έναν υποφάκελο με το ονοματεπώνυμό σας. Αυτός ο υποφάκελος θα χρησιμεύει για την οργάνωση και αποθήκευση όλων των αρχείων .m του MATLAB που θα δημιουργήσετε κατά τη διάρκεια του μαθήματος. Η σωστή οργάνωση των αρχείων σας θα διευκολύνει την πρόσβαση και τη διαχείριση των εργασιών σας.

1. Μαθησιακοί στόχοι

Με την ολοκλήρωση αυτού του εργαστηρίου ο φοιτητής θα είναι σε θέση να:

- **Κατανοήσει** τη λειτουργία και τη σημασία του ιστογράμματος στην ανάλυση εικόνας.
- **Εφαρμόσει** τεχνικές ισοστάθμισης ιστογράμματος για τη βελτίωση της ποιότητας εικόνων.
- **Αντιμετωπίσει** θόρυβο σε εικόνες **χρησιμοποιώντας** διαφορετικά φίλτρα, όπως το Gaussian, το median και το Wiener.
- **Σχεδιάσει** και να **εφαρμόσει** δικά του φίλτρα για τη βελτίωση της ποιότητας εικόνων και την αφαίρεση θορύβου.
- **Συγκρίνει** τις αποδόσεις διαφορετικών φίλτρων και να **αναλύει** ποια τεχνική είναι καταλληλότερη για κάθε περίπτωση.

2. Πώς το Μαθαίνουν

- **Πραγματοποιώντας** πειράματα σε πραγματικές εικόνες, οι φοιτητές θα μπορούν να **κατανοήσουν** πώς το ιστόγραμμα επηρεάζει τη φωτεινότητα και την αντίθεση μιας εικόνας.
- Μέσω της σύγκρισης διαφορετικών τεχνικών εξομάλυνσης, θα δουν ποια φίλτρα είναι πιο αποτελεσματικά ανάλογα με τον τύπο θορύβου.
- **Αναπτύσσοντας** δικά τους φίλτρα και αλγορίθμους, θα **αποκτήσουν** πρακτικές δεξιότητες στη διαχείριση και επεξεργασία εικόνας.
- **Εκτελώντας** αναλύσεις πριν και μετά την εφαρμογή τεχνικών, θα **βελτιώσουν** την ικανότητά τους να αξιολογούν αποτελέσματα επεξεργασίας εικόνας.

3. Ιστόγραμμα και Ισοστάθμιση Ιστογράμματος

Το ιστόγραμμα μιας εικόνας δείχνει τη συχνότητα εμφάνισης των επιπέδων γκρι σε αυτήν. Η ισοστάθμιση ιστογράμματος (histogram equalization) είναι μια τεχνική που επιτρέπει τη βελτίωση της αντίθεσης μιας εικόνας, κάνοντας την κατανομή των τιμών φωτεινότητας πιο ομοιόμορφη.

Εντολές MATLAB που χρησιμοποιούνται:

- **imhist():** Υπολογίζει και εμφανίζει το ιστόγραμμα μιας εικόνας.

- **histeq()**: Εφαρμόζει ισοστάθμιση ιστογράμματος για τη βελτίωση της αντίθεσης.

Θυμάμαι ότι:

Μία από τις πιο χρήσιμες εντολές αυτού του κεφαλαίου είναι:

```
r = randi([0 255],512,512,'uint8');
```

Τι κάνει αυτός ο κώδικας; Κατασκευάζουμε έναν πίνακα με στοιχεία που κυμαίνονται από 0 έως 255 και είναι ισομερώς διανεμημένα.

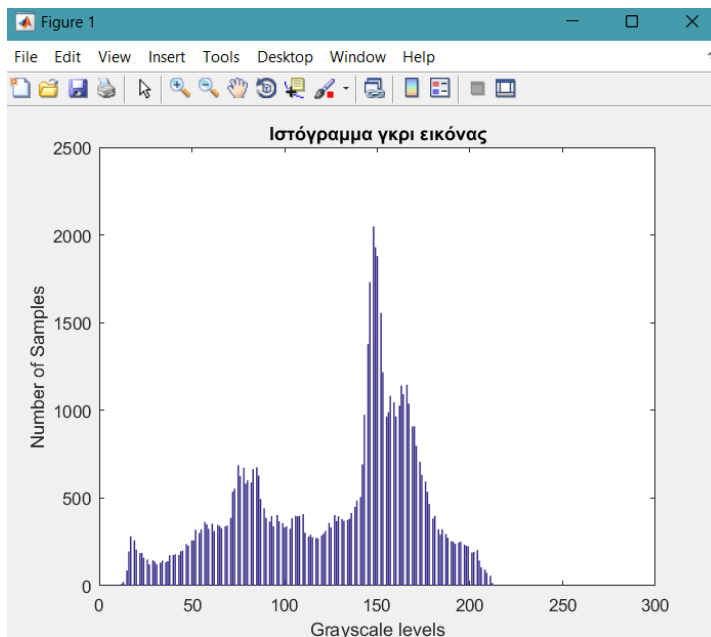
Άσκηση 1: Υπολογισμός και Ισοστάθμιση Ιστογράμματος

Αξιοποίησε τον ακόλουθο κώδικα:

```
Igray = imread ('bird.tif');  
h1 = imhist(Igray);  
bar(h1)  
bar(h1);xlabel('Grayscale levels');ylabel('Number of  
Samples');title('Ιστόγραμμα γκρι εικόνας')
```

Ερωτήσεις Κατανόησης:

- Τι διαφορές παρατηρείτε μεταξύ του αρχικού και του εξισωμένου ιστογράμματος;
- Σε ποιες περιπτώσεις είναι χρήσιμη η ισοστάθμιση ιστογράμματος;

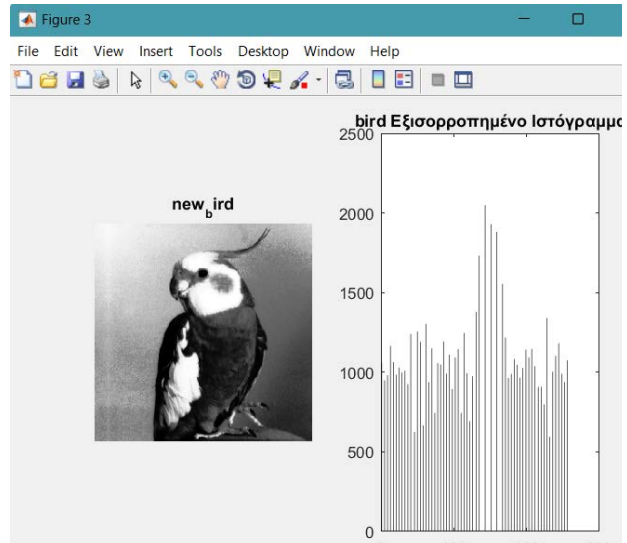
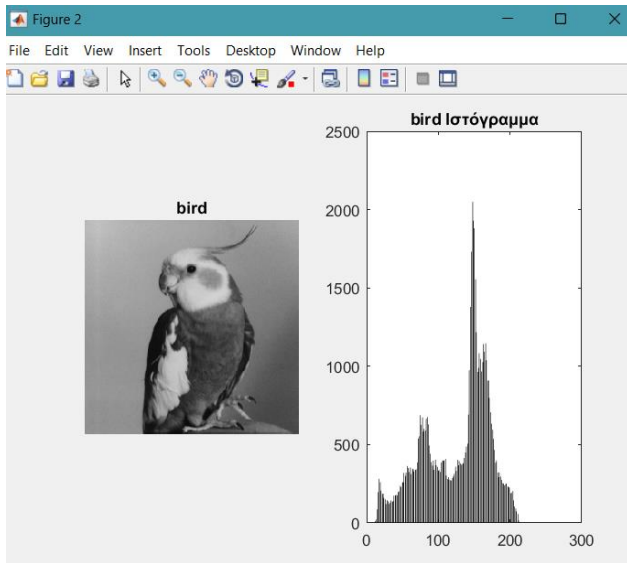


Εάν το ιστόγραμμα της εικόνας δεν συμμορφώνεται με τα κριτήριά μας, μπορούμε να αξιοποιήσουμε την εντολή `histeq` (εξίσωση ιστογράμματος), που αποτελεί μια αυτόματη διαδικασία, η οποία μετασχηματίζει το ιστόγραμμα σε ένα πιο ισοκατανομημένο. Η εντολή `histeq` παίρνει ως είσοδο έναν πίνακα (που αναπαριστά μια εικόνα στο Matlab) και δημιουργεί έναν νέο πίνακα, ο οποίος αντιπροσωπεύει την τροποποιημένη εικόνα, με τα επίπεδα του γκρι να έχουν αναδιανεμηθεί ώστε να επιτευχθεί πιο ομοιόμορφη κατανομή.

Χρησιμοποιήστε ακόμη και αυτόν τον κώδικα:

```
Igray = imread('bird.tif');
h1 = imhist(Igray);
figure
subplot(1,2,1);imshow(Igray);title('bird');sub-
plot(1,2,2);bar(h1);title('bird Ιστόγραμμα');
figure
new_Igray=histeq(Igray);
h2 = imhist(new_Igray);
subplot(1,2,1);                                imshow(new_Igray);ti-
tle('new_bird');subplot(1,2,2);bar(h2);title('bird Ε-
ξισορροπημένο Ιστόγραμμα');
```

Τι εμφανίζεται;



Άσκηση 2: Αφαίρεση Θορύβου με Φίλτρα

- Φορτώστε την εικόνα “frog.png” και προσθέστε θόρυβο.
- Αφαιρέστε τον θόρυβο χρησιμοποιώντας διαφορετικά φίλτρα και συγκρίνετε τα αποτελέσματα.

Υπόδειξη: Χρησιμοποιήστε τον τύπο `I = imread('frog.png');`

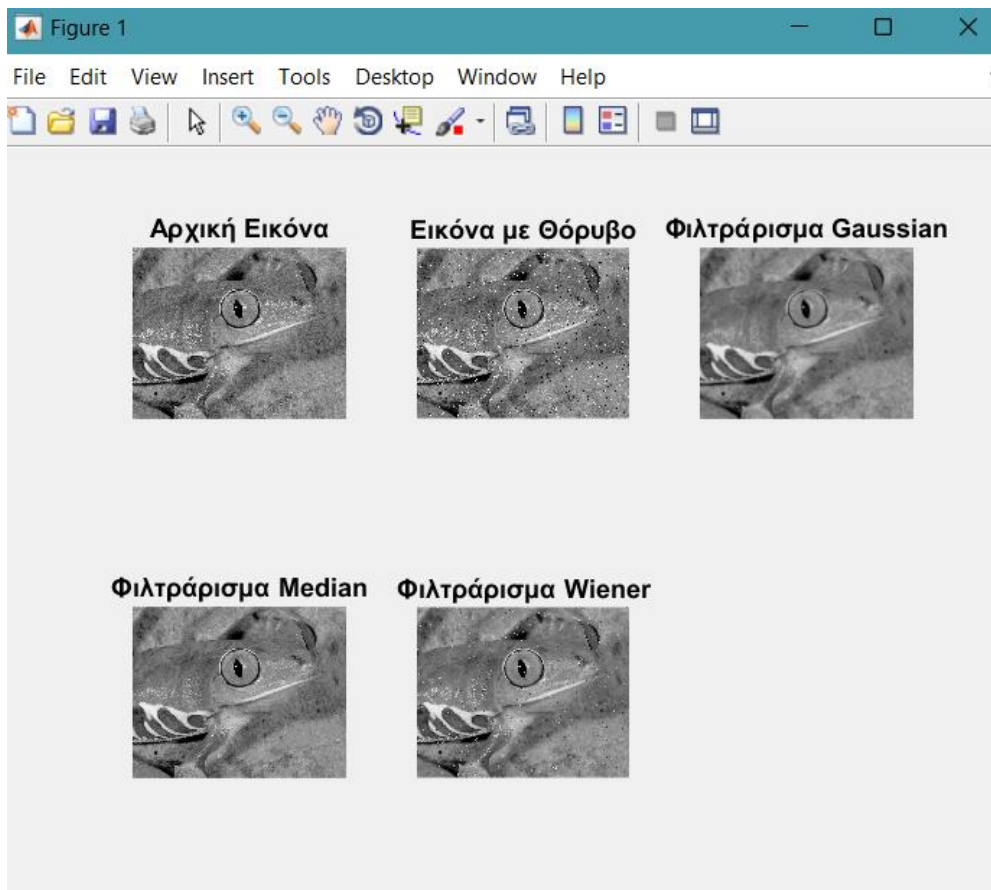
```
I_noisy = imnoise(I, 'salt & pepper', 0.02);
```

```
I_gauss = imgaussfilt(I_noisy, 2);
```

```
I_median = medfilt2(I_noisy, [3 3]);
```

```
I_wiener = wiener2(I_noisy, [5 5]);
```

Το αποτέλεσμα θα πρέπει να είναι αυτό!!!



Ερωτήσεις κατανόησης

- Ποιο φίλτρο παρήγαγε το καλύτερο αποτέλεσμα;
- Πώς επιλέγουμε ποιο φίλτρο είναι κατάλληλο σε κάθε περίπτωση;

Άσκηση εξάσκησης

Φορτώστε την εικόνα «frog.png» στο workspace της MATLAB και επισημάνετε τις αλλοιώσεις της. Υπολογίστε το ιστόγραμμα της εικόνας και προσαρμόστε τη διαδικασία ισοστάθμισης του ιστογράμματος.

- **Υπολογίστε** το συσσωρευμένο ιστόγραμμα.
- **Υπολογίστε** το ρυθμισμένο ιστόγραμμα διαιρώντας με το συνολικό αριθμό των στοιχείων της εικόνας.
- **Πολλαπλασιάστε** το κανονικοποιημένο ιστόγραμμα με τον συνολικό αριθμό των βαθμίδων γκρι.

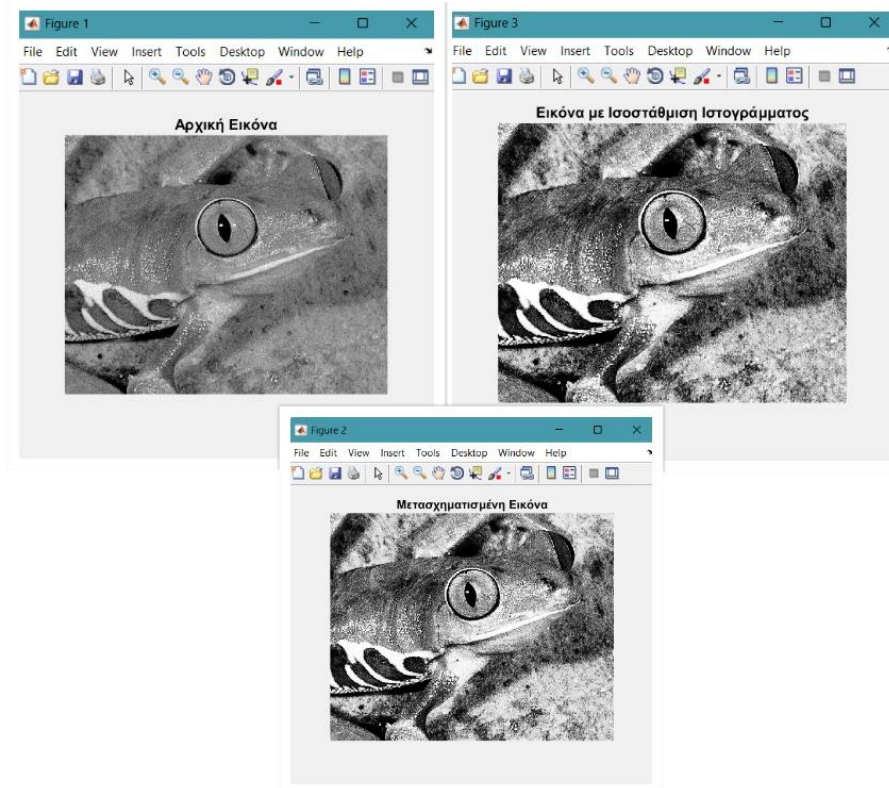
- **Αντικαταστήστε** την τιμή κάθε εικονοστοιχείου της εικόνας με τη νέα τιμή φωτεινότητας που προκύπτει από τον υπολογισμό.

Δείτε πρώτα την αρχική εικόνα και μετά την μετασχηματισμένη εικόνα. Επαληθεύστε τα αποτελέσματα χρησιμοποιώντας τις ενσωματωμένες συναρτήσεις της MATLAB για το ιστόγραμμα (imhist()) και την εξίσωση του ιστογράμματος (histeq()).

Λύση

```
I = imread('frog.png');
I = I(:,:,1);
horizontia_diastash = size(I,2);
katakoryfh_diastash = size(I,1);
histogram = zeros(1,256);
I_equal = zeros(size(I), 'uint8');
for level = 0:255
    I_level = (I == level);
    histogram(level+1) = sum(I_level(:));
end
cum_hist = cumsum(histogram);
norm_cum_hist = floor(cum_hist / (horizontia_diastash *
katakoryfh_diastash) * 255);
for i = 1:size(I,1)
    for j = 1:size(I,2)
        I_equal(i,j) = norm_cum_hist(I(i,j)+1);
    end
end
figure, imshow(I);
title('Αρχική Εικόνα');
figure, imshow(I_equal);
title('Μετασχηματισμένη Εικόνα');
hist = imhist(I);
J = histeq(I);
```

```
figure, imshow(J);
title('Εικόνα με Ισοστάθμιση Ιστογράμματος');
```



Άσκηση 3: Ανάγνωση της Εικόνας & Υπολογισμός Ιστογράμματος

Συντάξτε ένα νέο **script** που να φορτώνει την εικόνα του ‘frog’ από τον φάκελο datasets στο περιβάλλον εργασίας του MATLAB το οποίο να **υπολογίζει** και να **προβάλλει** το ιστόγραμμα της εικόνας. Μελετήστε το ιστόγραμμα και επιλέξτε μια τιμή ως κατώφλι για τα επίπεδα του γκρι. **Διατηρήστε** μόνο τα εικονοστοιχεία της εικόνας που υπερβαίνουν αυτή την τιμή κατωφλίου και **δημιουργήστε** μια δυαδική εικόνα με λευκά εικονοστοιχεία στο φόντο και μαύρα στην εικόνα που απεικονίζει τον βάτραχο.

Βοήθεια*

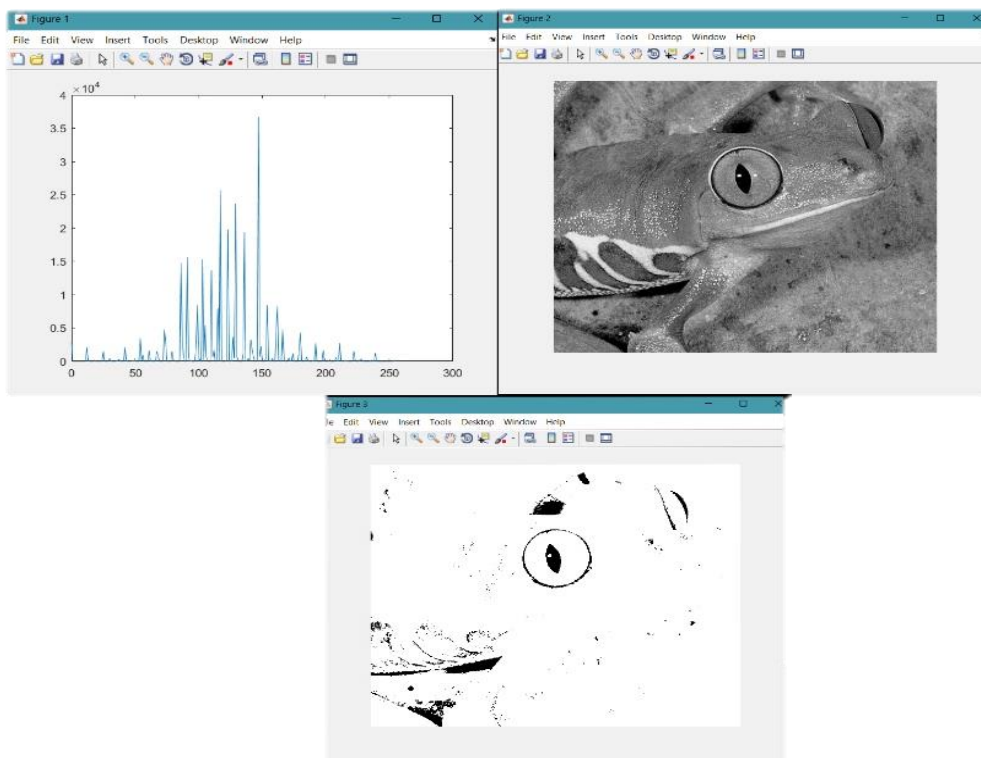
- Για να **καθορίσετε** την τιμή κατωφλίου, **εντοπίστε** το τοπικό **ελάχιστο** του ιστογράμματος που βρίσκεται ανάμεσα στις δύο κορυφές χαμηλής και υψηλής φωτεινότητας.

- Για την απόκτηση του ιστογράμματος, χρησιμοποιήστε την εντολή **imhist()**. Δημιουργήστε επίσης ένα διάνυσμα $x = [0, 1, 2, \dots, 255]$ για τον άξονα x του ιστογράμματος.
- Για να προβάλετε το ιστόγραμμα, χρησιμοποιήστε την εντολή **plot(x, histogram)**.
- Για να διατηρήσετε μόνο τα εικονοστοιχεία που ξεπερνούν μια συγκεκριμένη τιμή, χρησιμοποιήστε τη συνθήκη $I_bin = (I > threshold)$.

```

I = imread('frog.tif');
hist = imhist(I,256);
x = 0:1:255;
figure, plot(x,hist);
threshold = 50;
I_segmented =(I > 50);
figure, imshow(I)
figure, imshow(I_segmented)

```



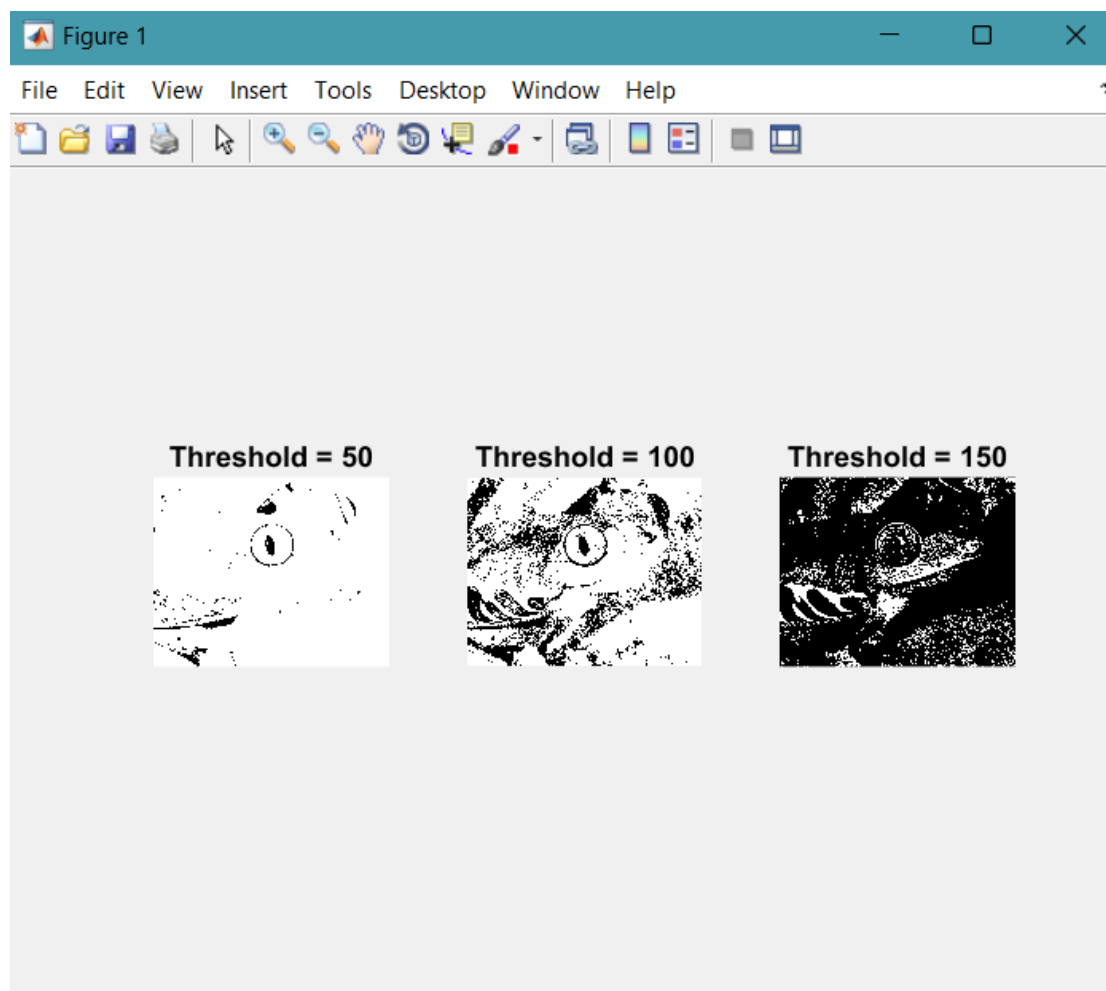
Ερωτήσεις Κατανόησης:

- Τι αλλαγές παρατηρείτε μεταξύ της αρχικής και της φιλτραρισμένης εικόνας;
- Πώς επηρεάζει το μέγεθος του παραθύρου το αποτέλεσμα του φιλτραρίσματος;

4. Άσκηση εξάσκησης

- Δοκιμάστε διαφορετικές τιμές κατωφλίου (threshold = 50, 100, 150) και παρατηρήστε τις αλλαγές.
- Εφαρμόστε thresholding με βάση το τοπικό ελάχιστο του ιστογράμματος.
- Σύγκριση των αποτελεσμάτων για διαφορετικές τιμές κατωφλίου.

Το αποτέλεσμα θα πρέπει να είναι αυτό!!!!



5. Ισοστάθμιση Ιστογράμματος

Η **ισοστάθμιση ιστογράμματος** είναι μια τεχνική **βελτίωσης** της αντίθεσης μιας εικόνας. Εφαρμόζεται κυρίως σε εικόνες όπου η κατανομή των επιπέδων του γκρι δεν είναι ομοιόμορφη, με

αποτέλεσμα να εμφανίζονται πολύ σκοτεινές ή πολύ φωτεινές περιοχές. Με την ισοστάθμιση ιστογράμματος, αναδιανέμουμε τα επίπεδα φωτεινότητας ώστε να είναι πιο ομοιόμορφα κατανεμημένα, αυξάνοντας έτσι την αντίθεση και καθιστώντας περισσότερες λεπτομέρειες ορατές.

Για να εφαρμόσουμε την ισοστάθμιση, ακολουθούμε τα εξής βήματα:

- **Υπολογισμός Ιστογράμματος**

Το **ιστόγραμμα** μιας εικόνας είναι ένας πίνακας που δείχνει πόσα **pixels** περιέχουν κάθε τιμή φωτεινότητας (**0-255**).

Για κάθε pixel της εικόνας, **μετράμε** πόσες φορές **εμφανίζεται** η συγκεκριμένη τιμή γκρι. Δημιουργούμε έναν πίνακα **H(i)** με **256 θέσεις**, όπου η κάθε θέση δείχνει τη **συχνότητα εμφάνισης** της αντίστοιχης φωτεινότητας.

6. Σύγκριση Ισοστάθμισης Ιστογράμματος με το `histeq()` του MATLAB

Χρησιμοποιήστε την ενσωματωμένη συνάρτηση **histeq()** του MATLAB και **συγκρίνετε** τα αποτελέσματα με τη χειροκίνητη υλοποίηση.

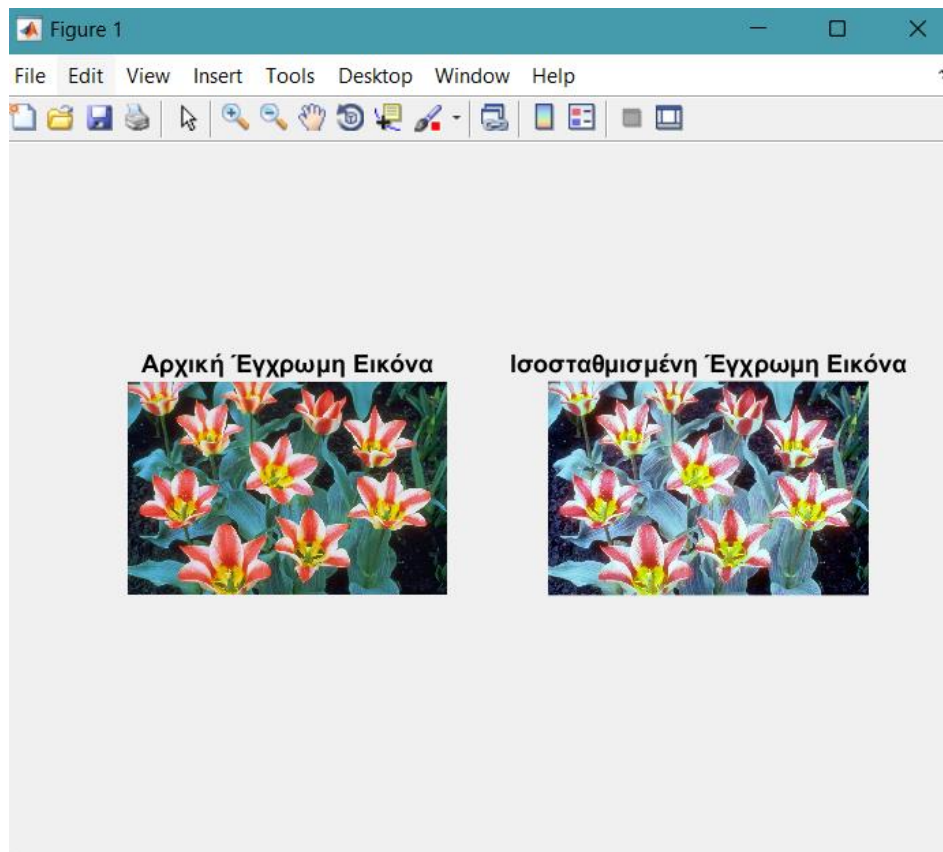
```
I_histeq = histeq(I_gray);  
figure;  
subplot(1,2,1);          imshow(I_equalized);          ti-  
tle('Χειροκίνητη Ισοστάθμιση');  
subplot(1,2,2); imshow(I_histeq); title('Ισοστάθμιση με  
histeq());
```

Ερωτήσεις κατανόησης:

- Οι δύο μέθοδοι δίνουν το ίδιο αποτέλεσμα;
- Ποια είναι η διαφορά μεταξύ τους;

7. Ισοστάθμιση Ιστογράμματος σε Έγχρωμες Εικόνες

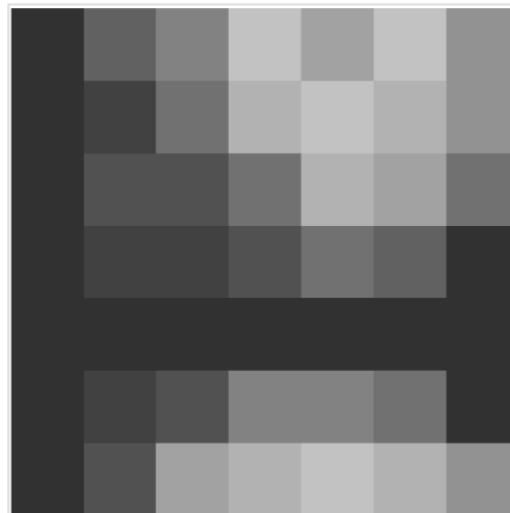
Εφαρμόστε την ισοστάθμιση ιστογράμματος ξεχωριστά στα κανάλια R, G, B μιας έγχρωμης εικόνας “tulips.tif” και συγκρίνετε τα αποτελέσματα. Το αποτέλεσμα θα πρέπει να είναι αυτό!!!



Άσκηση 4

Στην εικόνα I που ακολουθεί, εκτελέστε τη διαδικασία ισοστάθμισης του ιστογράμματος, υπολογίζοντας αναλυτικά τις τιμές των εξισωμένων επιπέδων του γκρι. Τέλος, επαληθεύστε τα αποτελέσματα χρησιμοποιώντας κώδικα MATLAB.

3	6	8	12	10	12	9
3	4	7	11	12	11	9
3	5	5	7	11	10	7
3	4	4	5	7	6	3
3	3	3	3	3	3	3
3	4	5	8	8	7	3
3	5	10	11	12	11	9



Βάσει της θεωρίας, τα βήματα που πρέπει να ακολουθήσουμε είναι τα εξής:

1. Αρχικά, πρέπει να υπολογίσουμε το ιστόγραμμα της εικόνας, καταγράφοντας την συχνότητα εμφάνισης κάθε επιπέδου του γκρι.
2. Μετά, υπολογίζουμε το σωρευτικό άθροισμα των συχνοτήτων εμφάνισης. Για κάθε επίπεδο φωτεινότητας, αθροίζουμε την εμφάνιση του επιπέδου αυτού και όλων των προγενέστερων επιπέδων.
3. Διαιρούμε με τον συνολικό αριθμό των εικονοστοιχείων.
4. Στο τέλος, πολλαπλασιάζουμε αυτήν την τιμή με το μέγιστο επίπεδο φωτεινότητας.

Διαθέσιμα επίπεδο του γκρι	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Συχνότητα Εμφάνισης επιπέδων	0	0	0	15	4	5	2	5	3	3	3	5	4	0	0	0
Σωρευμένο άθροισμα	0	0	0	15	19	24	26	31	34	37	40	45	49	49	49	49
Κανονικοποίηση δια των αριθμό των pixels (=49)	0/49	0/49	0/49	15/49	19/49	24/49	26/49	31/49	34/49	37/49	40/49	45/49	49/49	49/49	49/49	49/49
=	0	0	0	0.3061	0.3878	0.4898	0.5306	0.6327	0.6939	0.7551	0.8163	0.9184	1	1	1	1
Πολ/σμος με το μέγιστο διαθέσιμο επίπεδο γκρι (=15)	0	0	0	4.59	5.82	7.35	7.96	9.49	10.41	11.33	12.25	13.78	15	15	15	15
Τελική τιμή επιπέδου	0	0	0	5	6	7	8	9	10	11	12	14	15	15	15	15

$$I = [3 \ 6 \ 8 \ 12 \ 10 \ 12 \ 9;$$

```

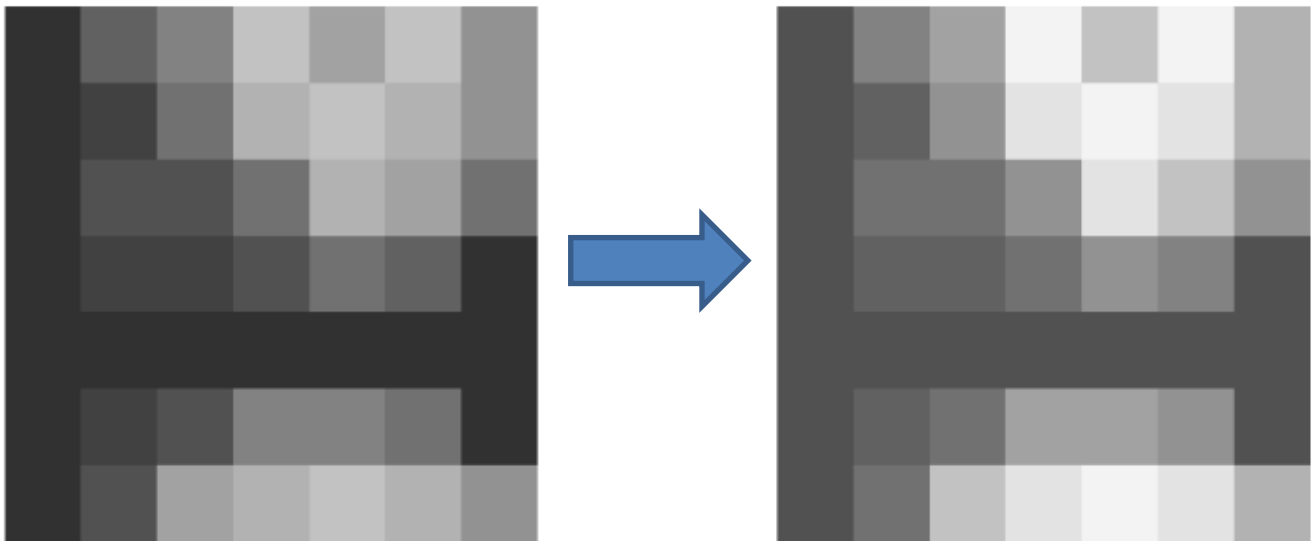
3 4 7 11 12 11 9;
3 5 5 7 11 10 7;
3 4 4 5 7 6 3;
3 3 3 3 3 3 3;
3 4 5 8 8 7 3;
3 5 10 11 12 11 9];
figure;
imshow(uint8(16 * I)); title('Original Image
(scaled)');
I_new_normalized = histeq(I, 16);
I_new_16levels = round(15 * I_new_normalized);
disp('Final Image after Histogram Equalization
(I_new_16levels):');
disp(I_new_16levels);
figure;
subplot(1, 2, 1);imshow(uint8(16 * I));title('Original
Image (scaled)');
subplot(1, 2, 2);imshow(uint8(I_new_16levels));ti-
tle('Equalized Image (16 levels)');
subplot(1,2,1);imshow(uint8(16*I));title('Initial Im-
age');subplot(1,2,2);imshow(uint8(16*I_new_16lev-
els));title('Final Image')

```

3	6	8	12	10	12	9
3	4	7	11	12	11	9
3	5	5	7	11	10	7
3	4	4	5	7	6	3
3	3	3	3	3	3	3
3	4	5	8	8	7	3
3	5	10	11	12	11	9



5	8	10	15	12	15	11
5	6	9	14	15	14	11
5	7	7	9	14	12	9
5	6	6	7	9	8	5
5	5	5	5	5	5	5
5	6	7	10	10	9	5
5	7	12	14	15	14	11



Φίλτρα

Η εφαρμογή φίλτρων στην επεξεργασία εικόνας αποτελεί κρίσιμη μέθοδο για την προ επεξεργασία ή την ενίσχυση της εικόνας. Ανάλογα με τον τύπο και τις χαρακτηριστικές δυνατότητες του φίλτρου, οι τεχνικές που χρησιμοποιούνται μπορεί να αναδείξουν ακμές ή στοιχεία της εικόνας, να εξομαλύνουν συγκεκριμένες περιοχές ή να συνδυάσουν και τις τρεις αυτές λειτουργίες ταυτόχρονα. Η διαδικασία φιλτραρίσματος περιλαμβάνει τη χρήση ενός «κυλιόμενου» παραθύρου που μετακινείται σε όλη την επιφάνεια της εικόνας. Συνήθως, η τιμή της φωτεινότητας του κεντρικού pixel αντικαθίσταται με μια τιμή που υπολογίζεται σύμφωνα με τον τύπο του φίλτρου που εφαρμόζεται. Το μέγεθος του παραθύρου είναι μια κρίσιμη παράμετρος που μπορεί να προσαρμόζεται ανάλογα με τις ανάγκες της εφαρμογής ή της ανάλυσης που πραγματοποιείται. Για παράδειγμα,

μεγαλύτερα παράθυρα μπορούν να προσφέρουν πιο ομαλές αλλαγές στην εικόνα αλλά με λιγότερες λεπτομέρειες, ενώ μικρότερα παράθυρα ενδέχεται να διατηρούν περισσότερες λεπτομέρειες αλλά με λιγότερη εξομάλυνση. Δύο ευρέως χρησιμοποιούμενα φίλτρα είναι το φίλτρο του κινούμενου μέσου, που υπολογίζει τον μέσο όρο των τιμών των pixel εντός του παραθύρου, και το φίλτρο του μέσου, που χρησιμοποιεί μια στατική μέθοδο για τον υπολογισμό των τιμών. Οι παρακάτω εντολές είναι χρήσιμες για την εφαρμογή και τη διαχείριση αυτών των φίλτρων, διευκολύνοντας την επεξεργασία της εικόνας με διάφορους τρόπους:

Φίλτρα	Εντολή Matlab
Gaussian	Imgaussfilt
Wiener	Wiener2
Median filter	Medfilt2
Standard deviation	Stdfilt
Range filter	Rangefilt
Entropy filter	entropyfilt

1. Φίλτρο Μέσης Τιμής (Mean Filter)

Το φίλτρο **μέσης τιμής** (Mean Filter) είναι ένα γραμμικό φίλτρο εξομάλυνσης που χρησιμοποιείται στην επεξεργασία εικόνας για τη μείωση του θορύβου και την εξομάλυνση των λεπτομερειών. Εφαρμόζεται σε κάθε pixel μιας εικόνας, αντικαθιστώντας την αρχική τιμή του με τον μέσο όρο των τιμών των γειτονικών pixels μέσα σε ένα προκαθορισμένο παράθυρο.

- **Επιλέγουμε** ένα τετράγωνο παράθυρο (**kernel**) μεγέθους $N \times N$ (π.χ., **3x3**, **5x5**, **7x7**).
- **Υπολογίζουμε** τη μέση τιμή των τιμών φωτεινότητας των **pixels** μέσα στο παράθυρο.
- **Αντικαθιστούμε** την τιμή του κεντρικού pixel με αυτή τη μέση τιμή.
- **Επαναλαμβάνουμε** τη διαδικασία για κάθε pixel της εικόνας.

Το φίλτρο αυτό βοηθάει στην εξομάλυνση απότομων αλλαγών φωτεινότητας, κάνοντας την εικόνα πιο ομαλή και λιγότερο θορυβώδη.

Άσκηση 1

Φορτώστε μια εικόνα από τον φάκελο `datasets` και εφαρμόστε σε αυτήν ένα φίλτρο μέσης τιμής (mean filter). Καθορίστε το μέγεθος του φίλτρου ως παράμετρο, επιλέγοντας το κατάλληλο μέγεθος του παραθύρου για την επεξεργασία. Αφού ολοκληρωθεί η εφαρμογή του φίλτρου, εμφανίστε και τις δύο εικόνες: την αρχική και την επεξεργασμένη. Αυτό θα σας επιτρέψει να συγκρίνετε τις διαφορές μεταξύ της εικόνας πριν και μετά την εφαρμογή του φίλτρου.

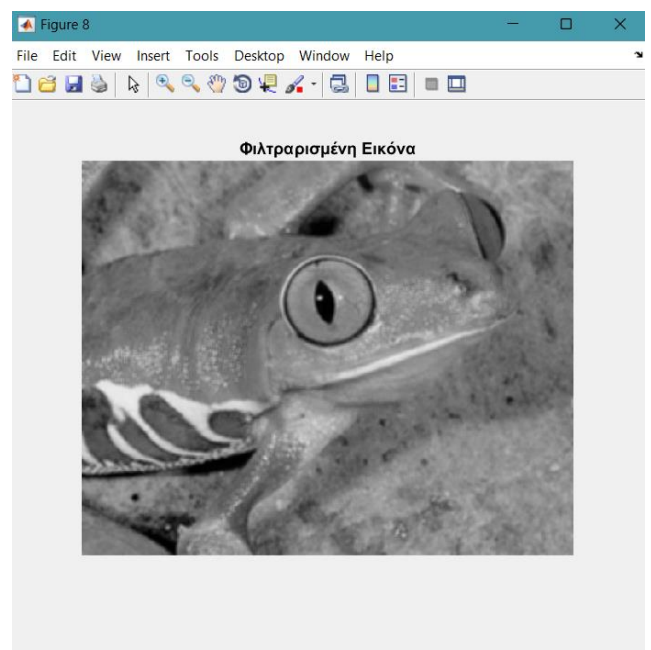
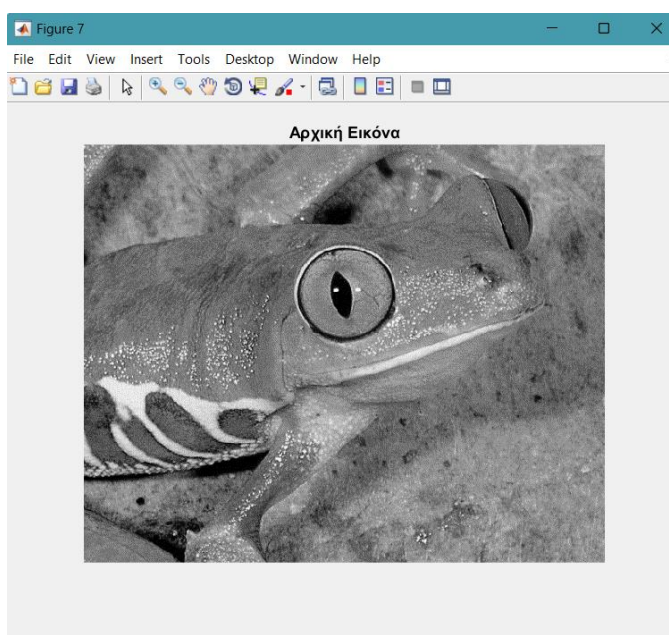
Υπόδειξη: Εφαρμόστε τα βήματα που περιγράφονται στην υπόδειξη για να καθοδηγηθείτε στη σύνταξη του κώδικα. Τα βήματα αυτά περιλαμβάνουν τη διαδικασία φόρτωσης της εικόνας, την εφαρμογή του φίλτρου μέσης τιμής, και την εμφάνιση των αποτελεσμάτων, ώστε να διασφαλίσετε ότι καλύπτονται όλα τα απαραίτητα βήματα για την ολοκλήρωση της εργασίας σας.

1. Φορτώστε την εικόνα από το αρχείο χρησιμοποιώντας την εντολή `imread()`. Στη συνάρτηση `imread()`.
2. Δημιουργήστε μια εικόνα γεμάτη με μηδενικές τιμές, η οποία να έχει το ίδιο μέγεθος με την αρχική εικόνα, για να χρησιμοποιηθεί ως η τελική φιλτραρισμένη εικόνα.
3. Καθορίστε το μέγεθος του παραθύρου και αποθηκεύστε το σε μια μεταβλητή.
4. Υπολογίστε, με βάση το μέγεθος του φίλτρου, την απόσταση από το κεντρικό pixel προς κάθε πλευρά.
5. Δημιουργήστε ένα μηδενικό παράθυρο με διαστάσεις ίσες με αυτές του παραθύρου που θα χρησιμοποιήσετε για την σάρωση της εικόνας.
6. Υλοποιήστε δύο επαναληπτικούς βρόχους `for` τοποθετημένους το ένα μέσα στο άλλο, ώστε να διατρέχουν όλες τις γραμμές και στήλες της εικόνας.
7. Για κάθε παράθυρο που εξετάζεται μέσα στον δεύτερο επαναληπτικό βρόχο, υπολογίστε τη μέση τιμή φωτεινότητας του παραθύρου.
8. Τοποθετήστε τη μέση τιμή που υπολογίσατε στην κεντρική θέση του παραθύρου στην τελική φιλτραρισμένη εικόνα.

9. Ολοκληρώστε τους δύο βρόχους.
10. Προβάλετε την αρχική εικόνα και την τελική εικόνα σε ξεχωριστά παράθυρα χρησιμοποιώντας τις εντολές `figure` και `imshow`.

Ποιες είναι οι διαφορές που παρατηρείτε μεταξύ της φιλτραρισμένης εικόνας και της αρχικής;

```
I = imread('frog.png');
filterSize = 5;
h = fspecial('average', filterSize);
filteredImg = imfilter(I, h, 'replicate');
figure;
imshow(I);
title('Αρχική Εικόνα');
figure;
imshow(filteredImg);
title('Φιλτραρισμένη Εικόνα');
disp('Εκτέλεση φιλτραρίσματος ολοκληρώθηκε.');
```



Παρατηρούμε τα εξής:

1. Μείωση Θορύβου
2. Εξομάλυνση Λεπτομερειών
3. Αναγνώριση Ακμών
4. Ομοιομορφία Χρωμάτων
5. Σαφήνεια Εικόνας

Άσκηση 2

Ας υποθέσουμε ότι ο πίνακας αντιπροσωπεύει μια εικόνα $I = \begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 3 \\ 4 & 4 & 1 \end{bmatrix}$

Χρησιμοποιήστε ένα φίλτρο μέσου, εφαρμόζοντας την προσέγγιση zero padding για να διαχειριστείτε τα άκρα της εικόνας. Στη συνέχεια, εφαρμόστε φίλτρο διάμεσου, χρησιμοποιώντας την προσέγγιση symmetric padding για τη διαχείριση των ορίων της εικόνας.

Εφαρμόστε τον παρακάτω κώδικα και εξηγήστε τη λειτουργία του, γράφοντάς τον με διαφορετική διατύπωση.

```
I=[2 3 4; 1 2 3; 4 4 1];  
mean_mask=fspecial('average',[3 3])  
I_mean=round(imfilter(I,mean_mask))  
I_median=medfilt2(I,'symmetric')
```

mean_mask =			I_mean =			I_median =		
0.1111	0.1111	0.1111	1	2	1	2	3	3
0.1111	0.1111	0.1111	2	3	2	2	3	3
0.1111	0.1111	0.1111	1	2	1	4	3	2

1. Άσκηση προς επίλυση

Δημιουργήστε ένα φίλτρο μέσης τιμής (median filter) και εφαρμόστε το στην εικόνα tulips.png.. Μπορείτε να την λύσετε σύμφωνα με τα βήματα της άσκησης 1.

```
I = imread('tulips.png');
filtered = uint8(zeros(size(I,1),size(I,2)));
megethos_parathyrou = 3;
offset = megethos_parathyrou/2 - 0.5;
parathyro = zeros(megethos_parathyrou,megethos_parathyrou);
for i=1+offset:size(I,1)-offset
    for j=1+offset:size(I,2)-offset
        parathyro(:, :)=I(i-offset:i+offset,j-offset:j+offset);
        line_parathyro = parathyro(:);
        median_parathyroy = median(line_parathyro);
        filtered(i,j) = median_parathyroy;
    end
end
figure, imshow(I)
figure, imshow(filtered)
```

- Παρατηρήστε τα αποτελέσματα και αναλύστε τι παρατηρείτε.

2. Ασκήσεις προς επίλυση: Εξομάλυνση Εικόνας

Σε αυτήν την άσκηση θα εργαστείτε με μια εικόνα που περιέχει θόρυβο και θα εφαρμόσετε διάφορα φίλτρα εξομάλυνσης για να μειώσετε την αιχμηρότητα της εικόνας.

- **Φορτώστε** την εικόνα **"tulips.jpg"**.

- Εφαρμόστε το μέσο φίλτρο (**mean filter**) με μέγεθος παραθύρου **3x3**.
- Εφαρμόστε το φίλτρο Gaussian με μέγεθος παραθύρου **5x5** και τυπική απόκλιση **2**.
 - Ποια είναι η επίδραση του φίλτρου μέσου όρου στην εικόνα; Ποιες περιοχές φαίνονται πιο "θολές";
 - Ποιες είναι οι διαφορές μεταξύ του μέσου φίλτρου και του Gaussian φίλτρου όσον αφορά την εξομάλυνση;

Συμπεράσματα και Προτάσεις για Εμβάθυνση

- Οι τεχνικές ιστογράμματος μπορούν να χρησιμοποιηθούν για τη βελτίωση της ποιότητας εικόνας.
- Τα φίλτρα εικόνας είναι απαραίτητα για την αφαίρεση θορύβου και τη βελτίωση λεπτομερειών.
- Οι διαφορετικές τεχνικές επεξεργασίας εικόνας πρέπει να επιλέγονται ανάλογα με την εκάστοτε ανάγκη.

Εργαστήριο 5 – Lab 5

Συμπίεση Εικόνας - Εντροπία

Σκοπός:

Οι φοιτητές να εξοικειωθούν με τις βασικές έννοιες της επεξεργασίας εικόνας, όπως η αναπαράσταση ψηφιακών εικόνων και τα χρωματικά συστήματα. Επίσης, να μάθουν τις απαραίτητες εντολές στο MATLAB ώστε μπορούν να επεξεργαστούν και να διαχειριστούν εικόνες με αποτελεσματικότητα.

Προσέξτε ότι:

Βεβαιωθείτε ότι έχετε δημιουργήσει έναν φάκελο στον υπολογιστή σας με την ονομασία "Επεξεργασία Εικόνας και Βίντεο", που αντιστοιχεί στο μάθημα. Μέσα σε αυτόν τον κύριο φάκελο, δημιουργήστε έναν υποφάκελο με το ονοματεπώνυμό σας. Αυτός ο υποφάκελος θα χρησιμεύει για την οργάνωση και αποθήκευση όλων των αρχείων .m του MATLAB που θα δημιουργήσετε κατά τη διάρκεια του μαθήματος. Η σωστή οργάνωση των αρχείων σας θα διευκολύνει την πρόσβαση και τη διαχείριση των εργασιών σας.

1. Μαθησιακοί Στόχοι

Οι μαθησιακοί στόχοι αυτής της ενότητας είναι να βοηθήσουν τους φοιτητές να κατανοήσουν τις έννοιες της εντροπίας, της συμπίεσης εικόνας και της τεχνικής Huffman. Στο τέλος της ενότητας, οι φοιτητές θα είναι σε θέση να:

1. **Υπολογίζουν** την εντροπία μιας εικόνας, **κατανοώντας** τον ρόλο της στην ανάλυση και συμπίεση εικόνας.
2. **Εφαρμόζουν** τη μέθοδο συμπίεσης Huffman σε μια εικόνα και να **κατανοούν** τις αρχές πίσω από αυτήν τη μέθοδο.
3. **Συγκρίνουν** τις τιμές εντροπίας πριν και μετά τη συμπίεση, **κατανοώντας** πώς η συμπίεση επηρεάζει την πληροφορία και την ποιότητα της εικόνας.
4. **Αναλύουν** την αποδοτικότητα της συμπίεσης και να **εξηγούν** τις αλλαγές στις τιμές εντροπίας με βάση τις τεχνικές συμπίεσης που χρησιμοποιούνται.

2. Πώς το Μαθαίνουν

5. **Θεωρητική Παρουσίαση & Παραδείγματα:** Οι φοιτητές μαθαίνουν τις βασικές έννοιες, όπως εντροπία και συμπίεση εικόνας, μέσω θεωρητικών επεξηγήσεων και παραδειγμάτων που δείχνουν πώς αναπαρίστανται οι εικόνες ως πίνακες και πώς εκτελούνται γεωμετρικοί μετασχηματισμοί στο MATLAB.
6. **Πρακτική Εφαρμογή:** Μέσω ασκήσεων εργαστηρίου, οι φοιτητές υλοποιούν κώδικα στο MATLAB, πειραματίζονται με μετασχηματισμούς και παρατηρούν την επίδραση στις εικόνες.
7. **Συγκριτική Ανάλυση & Ερωτήσεις Κατανόησης:** Αντιπαραβάλλουν τα αποτελέσματα πριν και μετά από μετασχηματισμούς, αναλύοντας τις αλλαγές στην ποιότητα της εικόνας και την εντροπία.
8. **Ανάπτυξη Projects:** Εφαρμόζουν τις γνώσεις τους σε ολοκληρωμένα projects, συνδυάζοντας μετασχηματισμούς και τεχνικές συμπίεσης για την επίλυση πραγματικών προβλημάτων εικόνας.

3. Εντροπία Εικόνας (Image Entropy)

Η εντροπία είναι ένας δείκτης που μετράει την αβεβαιότητα ή τη πληροφορία σε μια εικόνα. Υψηλή εντροπία σημαίνει ότι η εικόνα περιέχει περισσότερη πληροφορία (περισσότερες διαφοροποιήσεις στους τόνους), ενώ χαμηλή εντροπία υποδεικνύει λιγότερη ποικιλία (όπως μια εικόνα με μια ενιαία χρωματική περιοχή).

Ο τύπος για την υπολογισμό της εντροπίας μιας εικόνας είναι:

$$H = - \sum_{i=1}^n p_i \log_2(p_i)$$

4. Συμπίεση Εικόνας και Τεχνική Huffman

Η συμπίεση εικόνας έχει ως στόχο την μείωση του μεγέθους των δεδομένων που απαιτούνται για την αποθήκευση ή μετάδοση της εικόνας, ενώ διατηρεί την ποιότητά της όσο το δυνατόν περισσότερο. Η τεχνική Huffman είναι μια μέθοδος ασύμμετρης συμπίεσης, η οποία εκμεταλλεύεται την κατανομή των pixel της εικόνας και αντιστοιχεί σε μικρότερες ακολουθίες bits στις πιο συχνές τιμές pixel.

Ο αλγόριθμος Huffman δημιουργεί ένα δυαδικό δέντρο βασισμένο στις συχνότητες εμφάνισης των τιμών pixel και κατασκευάζει το πιο σύντομο δυαδικό κώδικα για τις πιο συχνές τιμές.

Θυμάμαι ότι:

Μία από τις πιο χρήσιμες εντολές αυτού του κεφαλαίου είναι:

```
r = randi([0 255],512,512,'uint8');
```

Τι κάνει αυτός ο κώδικας; Κατασκευάζουμε έναν πίνακα με στοιχεία που κυμαίνονται από 0 έως 255 και είναι ισομερώς διανεμημένα.

Άσκηση 1

Δίνεται μια γκριζα εικόνα bird.tif. Η εργασία σας είναι να υπολογίσετε την εντροπία της εικόνας και στη συνέχεια να εφαρμόσετε συμπίεση της εικόνας χρησιμοποιώντας την τεχνική συμπίεσης Huffman. Συγκρίνετε την αρχική εντροπία με την εντροπία της συμπιεσμένης εικόνας.

- Φορτώστε την εικόνα και υπολογίστε την εντροπία της.
- Εφαρμόστε τη συμπίεση Huffman στην εικόνα.
- Υπολογίστε την εντροπία της συμπιεσμένης εικόνας.
- Συγκρίνετε τις δύο τιμές εντροπίας και εξηγήστε τα αποτελέσματα.

```
I = imread('bird.tif');
initial_entropy = entropy(I);
fprintf('Αρχική Εντροπία: %.2f\n', initial_entropy);
compressed_entropy = entropy(I);
fprintf('Εντροπία μετά τη συμπίεση: %.2f\n', compressed_entropy);
if compressed_entropy < initial_entropy
    disp('Η εντροπία μειώθηκε μετά τη συμπίεση.');
```

```
else
```

```
    disp('Η εντροπία δεν άλλαξε σημαντικά μετά τη συμπίεση.');
```

```
end
```

Αρχική Εντροπία: 6.77

Εντροπία μετά τη συμπίεση: 6.77

Η εντροπία δεν άλλαξε σημαντικά μετά τη συμπίεση.

Ερώτηση κατανόησης

- Ποια είναι η διαφορά στην εντροπία πριν και μετά τη συμπίεση; Τι μας λέει αυτό για την αποδοτικότητα της μεθόδου Huffman;

Άσκηση 2

Από τη βάση δεδομένων εικόνων που βρίσκεται στον σύνδεσμο: <https://links.uwaterloo.ca/Repository.html>, κατεβάστε 2 εικόνες σε αποχρώσεις του γκρι με διαστάσεις 512x512 και 2 έγχρωμες εικόνες.

- Χρησιμοποιήστε την κατάλληλη εντολή για να υπολογίσετε την εντροπία κάθε grayscale εικόνας.
- Χρησιμοποιήστε την κατάλληλη εντολή για να υπολογίσετε την εντροπία κάθε έγχρωμης εικόνας αντίστοιχα.
- Για κάθε έγχρωμη εικόνα, υπολογίστε την εντροπία των καναλιών R, G και B χωριστά. Στη συνέχεια, συγκρίνετε αυτές τις τιμές με τις εντροπίες που υπολογίσατε για τις grayscale εικόνες στο δεύτερο ερώτημα και δείτε αν υπάρχουν διαφορές.
- Δημιουργήστε μια grayscale εικόνα τύπου uint8 με διαστάσεις 512 x 512, στην οποία οι τιμές των pixel κυμαίνονται ομοιόμορφα από 0 έως 255. Προβάλτε την εικόνα και υπολογίστε την εντροπία της. Στη συνέχεια, συγκρίνετε την εντροπία με τις εντροπίες που υπολογίσατε στο πρώτο ερώτημα και καταγράψτε τις όποιες διαφορές παρατηρήσετε.

```
I = imread('zelda.tif');  
if size(I, 3) == 3  
    I = rgb2gray(I);  
end  
image_entropy = entropy(I);  
fprintf('Η εντροπία της εικόνας zelda.tif είναι: %.4f\n', image_entropy);
```

Η εντροπία της εικόνας zelda.tif είναι: 7.2668

```
I = imread('boat (1).tif');  
if size(I, 3) == 3  
    I = rgb2gray(I);  
end  
image_entropy = entropy(I);
```

```
fprintf('Η εντροπία της εικόνας boat.tif είναι: %.4f\n', image_entropy);
```

Η εντροπία της εικόνας boat.tif είναι: 7.1238

```
I = imread('serrano.tif');  
gray_image = rgb2gray(I);  
entropy_gray = entropy(gray_image);  
fprintf('Η εντροπία της έγχρωμης εικόνας (μετατρεπόμενη σε grayscale) είναι: %.4f\n', entropy_gray);
```

Η εντροπία της έγχρωμης εικόνας (μετατρεπόμενη σε grayscale) είναι: 6.9796

```
I = imread('frymire.tif');  
gray_image = rgb2gray(I);  
entropy_gray = entropy(gray_image);  
fprintf('Η εντροπία της έγχρωμης εικόνας (μετατρεπόμενη σε grayscale) είναι: %.4f\n', entropy_gray);
```

Η εντροπία της έγχρωμης εικόνας (μετατρεπόμενη σε grayscale) είναι: 5.7383

```
I= imread('serrano.tif');  
R_channel = I(:, :, 1);  
G_channel = I(:, :, 2);  
B_channel = I(:, :, 3);  
entropy_R = entropy(R_channel);  
entropy_G = entropy(G_channel);  
entropy_B = entropy(B_channel);  
fprintf('Η εντροπία του καναλιού R είναι: %.4f\n', entropy_R);  
fprintf('Η εντροπία του καναλιού G είναι: %.4f\n', entropy_G);  
fprintf('Η εντροπία του καναλιού B είναι: %.4f\n', entropy_B);
```

Η εντροπία του καναλιού R είναι: 5.8806

Η εντροπία του καναλιού G είναι: 6.0238

Η εντροπία του καναλιού B είναι: 5.0938

```

I= imread('frymire.tif');
R_channel = I(:,:,1);
G_channel = I(:,:,2);
B_channel = I(:,:,3);
entropy_R = entropy(R_channel);
entropy_G = entropy(G_channel);
entropy_B = entropy(B_channel);
fprintf('H εντροπία του καναλιού R είναι: %.4f\n',
entropy_R);
fprintf('H εντροπία του καναλιού G είναι: %.4f\n',
entropy_G);
fprintf('H εντροπία του καναλιού B είναι: %.4f\n',
entropy_B);

```

```

H εντροπία του καναλιού R είναι: 4.4556
H εντροπία του καναλιού G είναι: 4.6566
H εντροπία του καναλιού B είναι: 4.5954

```

```

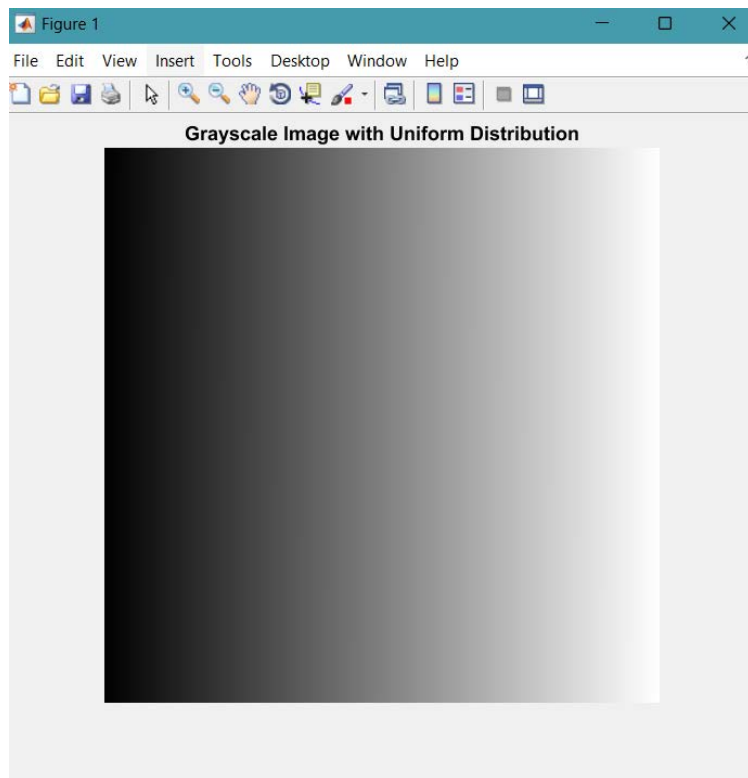
image_size = [512, 512];
num_levels = 256;
values = linspace(0, 255, numel(zeros(image_size)));
image = uint8(reshape(values, image_size));
imshow(image);
title('Grayscale Image with Uniform Distribution');
entropy_value = entropy(image);
fprintf('H εντροπία της εικόνας είναι: %.4f\n', en-
tropy_value);

```

```

H εντροπία της εικόνας είναι: 7.9983

```



1. Υπολογισμός Εντροπίας και Ανάλυση RGB

Δίνεται η εικόνα **peppers.png**, η οποία είναι έγχρωμη και έχει διαστάσεις **512x512 pixels**.

- **Φορτώστε** την εικόνα **peppers.png** και **υπολογίστε** την εντροπία της εικόνας.
- **Εμφανίστε** την εντροπία και **αναφέρετε** το αποτέλεσμα.
- **Υπολογίστε** την εντροπία για τα τρία κανάλια (Red, Green, Blue) της εικόνας ξεχωριστά.
- **Συγκρίνετε** τις εντροπίες των καναλιών και **εξηγήστε** αν υπάρχουν διαφορές μεταξύ τους. Τι πληροφορίες δίνει κάθε κανάλι όσον αφορά την εντροπία;

2. Εντροπία Εικόνας και Συμπίεση

Δίνεται η γκριζα εικόνα **"bird.tif"**. Στην εργασία σας, καλείστε να υπολογίσετε την εντροπία της εικόνας, να εφαρμόσετε τη μέθοδο συμπίεσης **Huffman** και να συγκρίνετε την αρχική εντροπία με την εντροπία της συμπιεσμένης εικόνας.

- **Φορτώστε** την εικόνα **"bird.tif"**.
- **Υπολογίστε** την αρχική εντροπία της εικόνας.
- **Εφαρμόστε** τη συμπίεση της εικόνας με τη μέθοδο **Huffman**.
- **Υπολογίστε** την εντροπία της συμπιεσμένης εικόνας.
- **Συγκρίνετε** τις δύο τιμές εντροπίας και **εξηγήστε** τα αποτελέσματα. Ποια είναι η σχέση μεταξύ της **εντροπίας** και της **συμπίεσης**; Η εντροπία **μειώθηκε** μετά τη συμπίεση;

3. Συγκριτική Ανάλυση Καναλιών και Συμπίεσης

- **Συγκρίνετε** την εντροπία του καναλιού R με τα κανάλια G και B. Ποιες πληροφορίες **μεταφέρει** το κάθε κανάλι και πώς σχετίζεται με την εντροπία;
- Σε ποιο κανάλι **παρατηρείτε** την **υψηλότερη** εντροπία και γιατί;
- Πώς η εντροπία **επηρεάζει** τη συμπίεση μιας εικόνας; Αν η εικόνα έχει υψηλή εντροπία, ποιο θα είναι το αποτέλεσμα στην τεχνική Huffman;
- Ποιες είναι οι **συνέπειες** της συμπίεσης στη διατήρηση της πληροφορίας της εικόνας; Ποιες εικόνες θα έχουν **καλύτερη** ποιότητα μετά την συμπίεση και ποιες **χειρότερη**, σύμφωνα με την εντροπία;

4. Συμπεράσματα και Προτάσεις για Εμβάθυνση

- Ο φοιτητής έμαθε τη σημασία και τη διαδικασία υπολογισμού της εντροπίας μιας εικόνας.
- Η εφαρμογή της συμπίεσης Huffman στη βελτιστοποίηση εικόνας έδωσε στον φοιτητή την ικανότητα να κατανοήσει και να εφαρμόσει αλγορίθμους συμπίεσης χωρίς απώλειες. Ο φοιτητής εξοικειώθηκε με την κωδικοποίηση Huffman και τη χρήση της για την αποδοτικότερη αποθήκευση εικόνας, χωρίς να χάνει σημαντική πληροφορία, σε αντίθεση με τη συμπίεση με απώλειες.
- Ο φοιτητής έμαθε να αναλύει την επίδραση των διαφόρων καναλιών χρώματος (R, G, B) στην εντροπία και να συγκρίνει την εντροπία εικόνας και καναλιών χωριστά.
- Η κατανόηση του θορύβου και της επίδρασής του στην εντροπία και στην ποιότητα της εικόνας προσέφερε στον φοιτητή μια σημαντική δεξιότητα στην ανάλυση και τη διαχείριση εικόνας με θόρυβο.

Εργαστήριο 6 – Lab 6

Επεξεργασία Βίντεο - Φίλτρα

Σκοπός:

Οι φοιτητές να εξοικειωθούν με τις βασικές έννοιες της επεξεργασίας βίντεο, όπως η αναπαράσταση ψηφιακών εικόνων καρέ και τα χρωματικά συστήματα. Επίσης, να μάθουν τις απαραίτητες εντολές στο MATLAB ώστε μπορούν να επεξεργαστούν και να διαχειριστούν βίντεο με αποτελεσματικότητα.

Προσέξτε ότι:

Βεβαιωθείτε ότι έχετε δημιουργήσει έναν φάκελο στον υπολογιστή σας με την ονομασία "Επεξεργασία Εικόνας και Βίντεο", που αντιστοιχεί στο μάθημα. Μέσα σε αυτόν τον κύριο φά-

κελο, δημιουργήστε έναν υποφάκελο με το ονοματεπώνυμό σας. Αυτός ο υποφάκελος θα χρησιμεύει για την οργάνωση και αποθήκευση όλων των αρχείων .m του MATLAB που θα δημιουργήσετε κατά τη διάρκεια του μαθήματος. Η σωστή οργάνωση των αρχείων σας θα διευκολύνει την πρόσβαση και τη διαχείριση των εργασιών σας.

1. Μαθησιακοί Στόχοι

Με την ολοκλήρωση του εργαστηρίου, οι φοιτητές θα πρέπει να είναι σε θέση να:

- **Κατανοούν** τη βασική αναπαράσταση του βίντεο και τα χρωματικά συστήματα.
- **Εφαρμόζουν** βασικές εντολές του MATLAB για φόρτωση, επεξεργασία και αποθήκευση βίντεο.
- **Μετατρέπουν** βίντεο σε κλίμακα του γκρι και να **εφαρμόζουν** συμπίεση με διαφορετικές μεθόδους.
- **Αναλύουν** και **εφαρμόζουν** φίλτρα επεξεργασίας εικόνας και βίντεο, όπως Gaussian Blur, Unsharp Masking, Edge Detection και Contrast Stretching.
- **Συγκρίνουν** και **αξιολογούν** τα αποτελέσματα των διαφορετικών φίλτρων.

2. Πως το μαθαίνουν

- **Διαδραστική εκμάθηση:** Οι φοιτητές θα πειραματιστούν με τις λειτουργίες του MATLAB χρησιμοποιώντας πραγματικά βίντεο και παρατηρώντας τα αποτελέσματα.
- **Σύγκριση διαφορετικών τεχνικών:** Θα εφαρμοστούν διαφορετικά φίλτρα για να κατανοήσουν τη χρήση και τη σημασία τους.
- **Συνεργατική μάθηση:** Οι φοιτητές μπορούν να δουλέψουν σε ομάδες για να επιλύσουν τις ασκήσεις και να συζητήσουν τις λύσεις τους.
- **Αξιολόγηση και σχολιασμός:** Οι φοιτητές θα καταγράψουν τα αποτελέσματά τους και θα αξιολογήσουν ποια μέθοδος είναι πιο αποτελεσματική.

Αναπαράσταση Βίντεο: Ένα βίντεο αποτελείται από διαδοχικές εικόνες (καρέ) που προβάλλονται με συγκεκριμένη ταχύτητα (frame rate), συνήθως 24-30 καρέ ανά δευτερόλεπτο. Τα βίντεο αποθηκεύονται σε διάφορες μορφές (MP4, AVI, MOV), με διαφορετικές μεθόδους συμπίεσης.

Χρωματικά Συστήματα: Τα περισσότερα βίντεο χρησιμοποιούν το σύστημα RGB, όπου κάθε pixel αποτελείται από τρία χρωματικά κανάλια (κόκκινο, πράσινο, μπλε). Η μετατροπή σε γκρι κλίμακα υπολογίζεται ως ένας σταθμισμένος μέσος όρος των καναλιών RGB.

3. Φίλτρα και η χρήση τους:

- **Gaussian Blur:** Χρησιμοποιείται για θόλωση και μείωση θορύβου.

- **Unsharp Masking:** Βελτιώνει την ευκρίνεια μιας εικόνας.
- **Edge Detection:** Ανιχνεύει και τονίζει τις άκρες των αντικειμένων.
- **Contrast Stretching:** Αυξάνει τη διαφορά μεταξύ φωτεινών και σκοτεινών περιοχών.

Κατεβάστε το βίντεο που θα βρείτε στο ακόλουθο link: <https://www.pexels.com/el-gr/video/856787/>

4. Άσκηση 1: Εισαγωγή στην Επεξεργασία Βίντεο

Στόχος: Κατανόηση των βασικών λειτουργιών του MATLAB για τη διαχείριση βίντεο.

- **Φορτώστε** ένα βίντεο στο MATLAB, κατά προτίμηση το “fish.mp4”.
- **Προβάλετε** τα καρέ με `imshow`.
- **Ανακτήστε** πληροφορίες για το βίντεο (διάρκεια, καρέ ανά δευτερόλεπτο).

Το αποτέλεσμα θα πρέπει να είναι αυτό!!!

Διάρκεια: 5.53 δευτερόλεπτα

Καρέ ανά δευτερόλεπτο: 29.97

Συνολικά καρέ: 165

5. Άσκηση 2: Μετατροπή βίντεο σε κλίμακα του γκρι και εξαγωγή του ως νέο αρχείο

Φορτώστε στο Matlab ένα βίντεο fish.mp4 και στη συνέχεια ακολουθήστε τα ακόλουθα βοηθητικά βήματα για την επίλυση της άσκησης.

Θυμάμαι ότι:

- **Φόρτωση του αρχείου βίντεο:** Εφαρμόστε τη συνάρτηση VideoReader του MATLAB για να εισάγετε το αρχείο βίντεο που επιθυμείτε, όπως για παράδειγμα το αρχείο video.mp4, ώστε να μπορέσετε να επεξεργαστείτε τα καρέ του βίντεο στον κώδικά σας.
- **Μετατροπή σε γκρι κλίμακα:** Χρησιμοποιήστε τη συνάρτηση rgb2gray για να μετατρέψετε κάθε καρέ του βίντεο σε εικόνα μόνο με αποχρώσεις του γκρι.
- **Δημιουργία νέου αρχείου βίντεο:** Εφαρμόστε τη συνάρτηση VideoWriter για να δημιουργήσετε ένα νέο αρχείο βίντεο αποθηκεύοντας το επεξεργασμένο βίντεο. Επιλέξτε τη μορφή MPEG-4 για την αποθήκευση και καθορίστε το όνομα του αρχείου, όπως για παράδειγμα output_gray_video.mp4, έτσι ώστε να διατηρηθεί η ποιότητα του βίντεο και να είναι έτοιμο για αναπαραγωγή ή περαιτέρω επεξεργασία.
- **Εξαγωγή του βίντεο:** Αποθηκεύστε το επεξεργασμένο βίντεο διατηρώντας τον ίδιο ρυθμό καρέ όπως είχε το αρχικό βίντεο. Αυτό εξασφαλίζει ότι το νέο βίντεο θα έχει την ίδια ταχύτητα αναπαραγωγής με το πρωτότυπο, παρέχοντας ομαλή και ακριβή αναπαραγωγή του περιεχομένου.

Ερωτήσεις προς επίλυση

1. Τι επιπτώσεις έχει η μετατροπή σε γκρι κλίμακα στα δεδομένα του βίντεο;

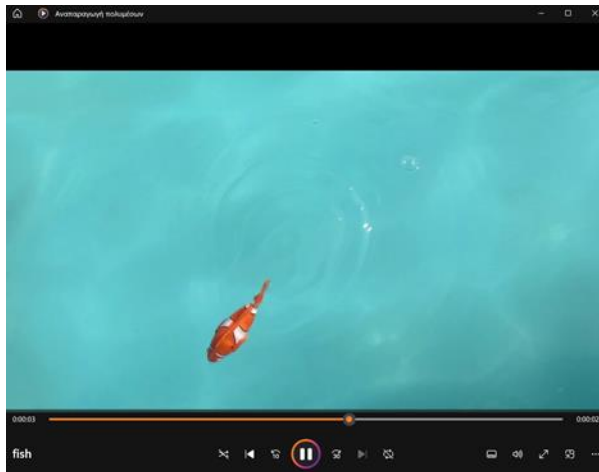
2. Ποια είναι η διαφορά στην εμφάνιση του βίντεο;
3. Πώς αλλάζει η ποιότητα του βίντεο μετά τη μετατροπή;
4. Πώς επηρεάζεται το μέγεθος του αρχείου;
5. Τροποποίησε τον κώδικα ώστε να αποθηκεύει το νέο βίντεο με διαφορετικές μεθόδους συμπίεσης.
6. Τροποποίησε τον κώδικα ώστε να εμφανίζει μόνο ένα κανάλι χρώματος, μόνο το κόκκινο κανάλι και ξεχωριστά μόνο με το μπλε κανάλι.

Λύση

Χρησιμοποιήστε τη συνάρτηση VideoReader του MATLAB για να εισάγετε το αρχείο βίντεο που επιθυμείτε, όπως το fish.mp4. Η συνάρτηση αυτή επιτρέπει την ανάγνωση του βίντεο και την πρόσβαση στα καρέ του.

```
vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
duration = vidObj.Duration;
numFrames = floor(duration * frameRate);
disp(['Αριθμός καρέ: ', num2str(numFrames)]);
outputVideo = VideoWriter('output_gray_video.mp4',
'MPEG-4');
outputVideo.FrameRate = frameRate;
open(outputVideo);
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    grayFrame = rgb2gray(frame);
    writeVideo(outputVideo, grayFrame);
end
close(outputVideo);

close(outputVideo);
Αριθμός καρέ: 165
```



Η μετατροπή σε γκριζα κλίμακα μειώνει την ποσότητα πληροφορίας που αποθηκεύεται σε κάθε καρέ του βίντεο. Το βίντεο χάνει τις πληροφορίες χρώματος και κρατά μόνο την ένταση φωτός. Αυτό μπορεί να μειώσει το μέγεθος του αρχείου και την επεξεργαστική απαιτητικότητα, αλλά μπορεί επίσης να επηρεάσει την οπτική ποιότητα και την ικανότητα αναγνώρισης στοιχείων βασισμένων σε χρώμα. Το βίντεο σε κλίμακα του γκρι εμφανίζει μόνο αποχρώσεις του γκρι, χωρίς χρώμα. Η έλλειψη χρώματος μπορεί να κάνει δύσκολη την αναγνώριση ή την ανάλυση αντικειμένων που βασίζονται σε χρωματικές πληροφορίες. Το βίντεο μπορεί να φαίνεται λιγότερο ζωντανό και πιο απλοϊκό.

Συμπίεση σε MPEG-4

```
outputVideoMPEG4 = VideoWriter('out-  
put_mpeg4_video.mp4', 'MPEG-4');  
outputVideoMPEG4.FrameRate = frameRate;  
open(outputVideoMPEG4);  
while hasFrame(vidObj)  
    frame = readFrame(vidObj);  
    grayFrame = rgb2gray(frame);  
    writeVideo(outputVideoMPEG4, grayFrame);  
end
```



Συμπίεση σε JPEG AVI

```
vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
outputVideoMJPEG = VideoWriter('out-
put_mjpeg_video.avi', 'Motion JPEG AVI');
outputVideoMJPEG.FrameRate = frameRate;
open(outputVideoMJPEG);
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    grayFrame = rgb2gray(frame);
    writeVideo(outputVideoMJPEG, grayFrame);
end
close(outputVideoMJPEG);
```



Συμπίεση Uncompressed AVI

```
vidObj = VideoReader('fish.mp4');
```

```

frameRate = vidObj.FrameRate;
outputVideoUncompressed = VideoWriter('output_uncom-
pressed_video.avi', 'Uncompressed AVI');
outputVideoUncompressed.FrameRate = frameRate;
open(outputVideoUncompressed);
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    grayFrame = rgb2gray(frame);
    writeVideo(outputVideoUncompressed, grayFrame);
end
close(outputVideoUncompressed);

```



Μπλε κανάλι

```

vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
outputVideoBlue = VideoWriter('blue_channel_out-
put.avi', 'Uncompressed AVI');
outputVideoBlue.FrameRate = frameRate;
open(outputVideoBlue);
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    blueChannel = frame(:, :, 3);
    blueFrame = cat(3, zeros(size(blueChannel)), ze-
ros(size(blueChannel)), blueChannel);
    writeVideo(outputVideoBlue, blueFrame);
end
close(outputVideoBlue);

```



Κόκκινο κανάλι

```
vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
outputVideoRed = VideoWriter('red_channel_output.avi',
'Uncompressed AVI');
outputVideoRed.FrameRate = frameRate;
open(outputVideoRed);
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    redChannel = frame(:,:,1);
    redFrame = cat(3, redChannel, zeros(size(redChannel)), zeros(size(redChannel)));
    writeVideo(outputVideoRed, redFrame);
end
close(outputVideoRed);
```



6. Άσκηση 3: Μετατροπή σε Γκρι Κλίμακα και Ανάλυση Μεγέ- θους

Στόχος: Κατανόηση της μείωσης πληροφορίας στη γκρι κλίμακα.

- **Μετατρέψτε** ένα βίντεο, κατά προτίμηση το “**fish.mp4**” σε γκρι κλίμακα.
- **Συγκρίνετε** το μέγεθος των αρχείων πριν και μετά τη μετατροπή.
- **Σχολιάστε** την επίδραση στη συμπίεση.

Το αποτέλεσμα θα πρέπει να είναι αυτό!!!

Μέγεθος αρχικού βίντεο: 3.10 MBΜέγεθος γκρι βίντεο: 4.14 MBΜείωση μεγέθους: -33.57%>>

7. Άσκηση 3: Εφαρμογή και Αξιολόγηση Σύνθετων Φίλτρων Επεξεργασίας Βίντεο με MATLAB

Φορτώστε στο Matlab ένα βίντεο fish.mp4 και στη συνέχεια ακολουθήστε τα ακόλουθα βοη-
θητικά βήματα για την επίλυση της άσκησης.

- **Φόρτωση του αρχείου βίντεο:** Εφαρμόστε τη συνάρτηση VideoReader του MATLAB για να εισάγετε το αρχείο βίντεο που επιθυμείτε, όπως για παράδειγμα το αρχείο video.mp4, ώστε να μπορέσετε να επεξεργαστείτε τα καρέ του βίντεο στον κώδικά σας.
- **Δημιουργία Νέου Βίντεο:** Για να δημιουργήσετε ένα νέο αρχείο βίντεο, χρησιμοποιήστε τη συνάρτηση VideoWriter του MATLAB. Ορίστε το όνομα του νέου αρχείου βίντεο, όπως processed_video.mp4, και διασφαλίστε ότι ο ρυθμός καρέ του νέου βίντεο είναι ίδιος με αυτόν του αρχικού βίντεο. Αφού ρυθμίσετε το όνομα και τις παραμέτρους, ανοίξτε το αρχείο για να προχωρήσετε στην εγγραφή των επεξεργασμένων καρέ.
- **Εφαρμογή Φίλτρων:** Για να επεξεργαστείτε κάθε καρέ του βίντεο, εφαρμόστε τα παρακάτω φίλτρα σε καθένα από αυτά:

Φίλτρα	Συνάρτηση στο Matlab
Φίλτρο Θολώματος (Gaussian Blur)	imgaussfilt
Φίλτρο Ευκρίνειας (Unsharp Masking)	imsharpen
Φίλτρο Ανίχνευσης Άκρων (Edge Detection)	edge
Φίλτρο Αντίθεσης (Contrast Stretching)	imadjust

Ερωτήσεις προς επίλυση

- 1) Πώς επηρεάζουν οι διαφορετικές παράμετροι την ποιότητα του βίντεο;

- 2) Πώς μπορεί η επεξεργασία να βελτιώσει την ορατότητα λεπτομερειών;
- 3) Εφαρμόστε το φίλτρο θολώματος (Gaussian Blur) με διάφορες τιμές του παραμέτρου σ (π.χ., 1, 2, 3). Δημιουργήστε μια σύγκριση των επεξεργασμένων βίντεο και αναφέρετε πώς οι διαφορετικές τιμές σ επηρεάζουν την οπτική ποιότητα του βίντεο.
- 4) Εφαρμόστε το φίλτρο ευκρίνειας (Unsharp Masking) σε καρέ του βίντεο και καταγράψτε τα αποτελέσματα.
- 5) Εφαρμόστε τα φίλτρα με διαφορετικές παραμέτρους και συγκρίνετε τα αποτελέσματα.
- 6) Δημιουργήστε ένα σύνθετο φίλτρο που συνδυάζει θόλωση, ευκρίνεια, ανίχνευση άκρων και ενίσχυση αντίθεσης. Εφαρμόστε αυτό το σύνθετο φίλτρο στα καρέ του βίντεο και αξιολογήστε το τελικό αποτέλεσμα.

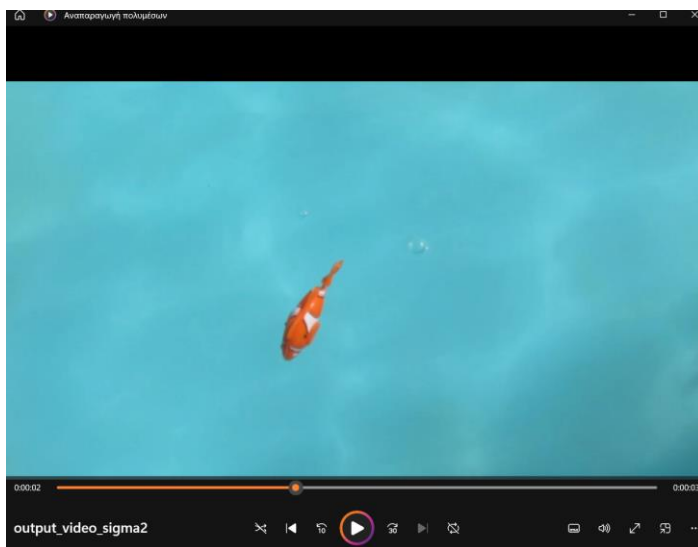
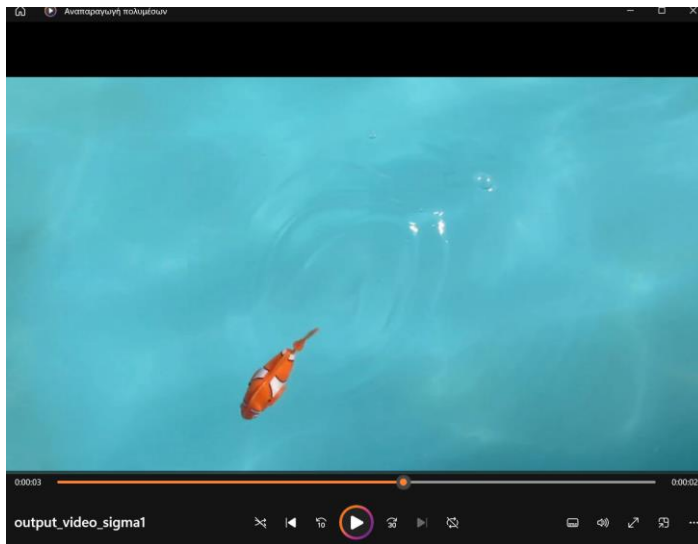
Λύση

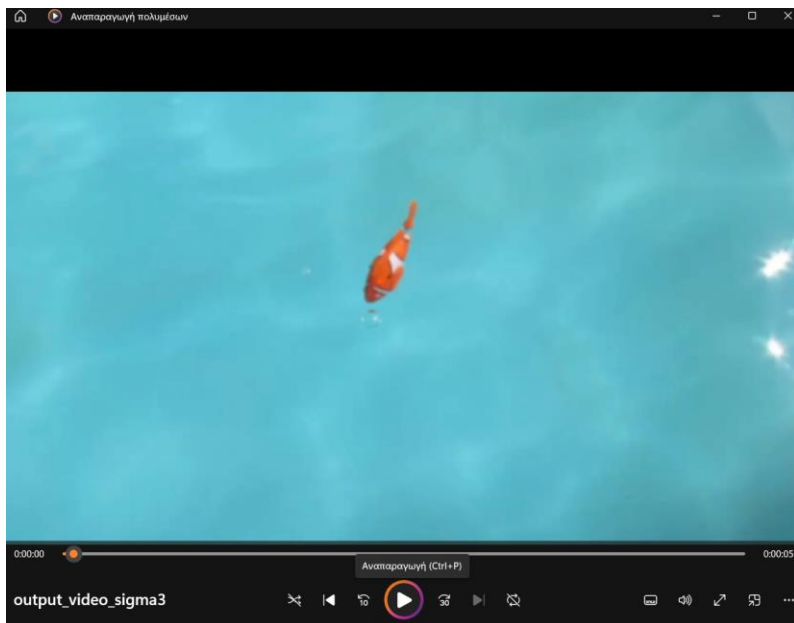
```

vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
outputVideoSigma1 = VideoWriter('out-
put_video_sigma1.mp4', 'MPEG-4');
outputVideoSigma2 = VideoWriter('out-
put_video_sigma2.mp4', 'MPEG-4');
outputVideoSigma3 = VideoWriter('out-
put_video_sigma3.mp4', 'MPEG-4');
outputVideoSigma1.FrameRate = frameRate;
outputVideoSigma2.FrameRate = frameRate;
outputVideoSigma3.FrameRate = frameRate;
open(outputVideoSigma1);
open(outputVideoSigma2);
open(outputVideoSigma3);
sigmaValues = [1, 2, 3];
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    for i = 1:length(sigmaValues)
        sigma = sigmaValues(i);
        blurredFrame = imgaussfilt(frame, sigma);
        Switch sigma
            case 1
                writeVideo(outputVideoSigma1,
blurredFrame);
            case 2
                writeVideo(outputVideoSigma2,
blurredFrame);
            case 3
                writeVideo(outputVideoSigma3,
blurredFrame);

```

```
        end
    end
end
close(outputVideoSigma1);
close(outputVideoSigma2);
close(outputVideoSigma3);
```





Για $\sigma = 1$: Η εικόνα παρουσιάζει μια ελαφριά θόλωση. Οι λεπτομέρειες παραμένουν σχετικά ευκρινείς, ενώ η συνολική εικόνα εμφανίζεται πιο ομαλή και μαλακή.

Για $\sigma = 2$: Η θόλωση είναι μέτρια, με αποτέλεσμα οι λεπτομέρειες να αρχίζουν να εξαφανίζονται και οι άκρες να μην είναι τόσο ευκρινείς.

Για $\sigma = 3$: Η θόλωση είναι έντονη, καθιστώντας τις λεπτομέρειες δύσκολα αναγνωρίσιμες και κάνοντάς την εικόνα να φαίνεται πιο θολή και ασαφής.

```
vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
outputVideo = VideoWriter('output_unsharp_mask-
ing.mp4', 'MPEG-4');
outputVideo.FrameRate = frameRate;
open (outputVideo);
sigma = 1.0;
amount = 1.5;
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    grayFrame = rgb2gray(frame);
    blurredFrame = imgaussfilt(grayFrame, sigma);
    sharpenedFrame = imadjust(grayFrame - blurredFrame,
    [], [], amount) + grayFrame;
    writeVideo(outputVideo, uint8(sharpenedFrame));
end
close(outputVideo);
```



```

vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
outputVideoSigma1 = VideoWriter('output_sigma1.mp4',
'MPEG-4');
outputVideoSigma2 = VideoWriter('output_sigma2.mp4',
'MPEG-4');
outputVideoSigma3 = VideoWriter('output_sigma3.mp4',
'MPEG-4');
outputVideoSigma1.FrameRate = frameRate;
outputVideoSigma2.FrameRate = frameRate;
outputVideoSigma3.FrameRate = frameRate;
open(outputVideoSigma1);
open(outputVideoSigma2);
open(outputVideoSigma3);
sigma_values = [1.0, 2.0, 3.0];
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    grayFrame = rgb2gray(frame);
    blurredFrame1 = imgaussfilt(grayFrame, sigma_val-
ues(1));
    blurredFrame2 = imgaussfilt(grayFrame, sigma_val-
ues(2));
    blurredFrame3 = imgaussfilt(grayFrame, sigma_val-
ues(3));
    writeVideo(outputVideoSigma1,
uint8(blurredFrame1));

```

```
        writeVideo(outputVideoSigma2,  
uint8(blurredFrame2));  
        writeVideo(outputVideoSigma3,  
uint8(blurredFrame3));  
end  
close(outputVideoSigma1);  
close(outputVideoSigma2);  
close(outputVideoSigma3);
```





Η αύξηση της τιμής του σ ενισχύει την επίδραση του φίλτρου θολώματος, κάνοντάς την πιο έντονη. Αυτό έχει ως αποτέλεσμα την μεγαλύτερη απώλεια λεπτομερειών στην εικόνα και την επακόλουθη δημιουργία μιας πιο ομαλής και θολής εμφάνισης. Όσο υψηλότερη είναι η τιμή του σ , τόσο πιο ευρύ είναι το φάσμα της θόλωσης που εφαρμόζεται, με αποτέλεσμα τις πιο αμβλείες άκρες και λιγότερη καθαρότητα στις λεπτομέρειες.

```
vidObj = VideoReader('fish.mp4');
frameRate = vidObj.FrameRate;
outputVideo = VideoWriter('processed_video.mp4', 'MPEG-4');
outputVideo.FrameRate = frameRate;
open(outputVideo);
sigma = 2.0;
amount = 1.5;
radius = 1.0;
edgeMethod = 'sobel';
contrastFactor = 1.5;
while hasFrame(vidObj)
    frame = readFrame(vidObj);
    grayFrame = rgb2gray(frame);
    blurredFrame = imgaussfilt(grayFrame, sigma);
    sharpFrame = imsharpen(blurredFrame, 'Amount', amount, 'Radius', radius);
    edges = edge(sharpFrame, edgeMethod);
    contrastFrame = imadjust(sharpFrame, [], [], contrastFactor);
    processedFrame = imadd(contrastFrame, uint8(edges) * 255);
```

```
writeVideo(outputVideo, processedFrame);  
end  
close(outputVideo);
```



Ασκήσεις προς επίλυση

8. Άσκηση 1: Εφαρμογή Φίλτρων και Οπτική Αξιολόγηση

Στόχος: Κατανόηση των διαφορετικών φίλτρων και της επίδρασής τους.

1. **Εφαρμόστε** Gaussian Blur σε διαφορετικές τιμές sigma.
2. **Δοκιμάστε** το φίλτρο Unsharp Masking και **συγκρίνετε** την ευκρίνεια.
3. **Εφαρμόστε** ανίχνευση ακμών και **σχολιάστε** το αποτέλεσμα.

9. Άσκηση 2: Συνδυασμός Φίλτρων σε Σύνθετη Επεξεργασία

Στόχος: Κατανόηση του τρόπου συνδυασμού πολλών φίλτρων για καλύτερη επεξεργασία.

1. **Μετατρέψτε** το βίντεο κατά προτίμηση το “fish.mp4” σε γκρι κλίμακα.
2. **Θολώστε** την εικόνα με Gaussian Blur.
3. **Εφαρμόστε** Unsharp Masking.
4. **Ανιχνεύστε** άκρα.
5. **Εφαρμόστε** Contrast Stretching.
6. **Συγκρίνετε** τα αποτελέσματα με το αρχικό βίντεο.

10. Άσκηση 3: Ανίχνευση Κίνησης και Παρακολούθηση Αντικειμένων

Γράψτε ένα script στο MATLAB για την ανίχνευση κίνησης σε ένα βίντεο και την παρακολούθηση των κινούμενων αντικειμένων. Το script θα πρέπει να επισημαίνει τα κινούμενα αντικείμενα με ένα ορθογώνιο πλαίσιο και να εμφανίζει την τροχιά του κεντροειδούς τους.

Οδηγίες:

1. **Φορτώστε** ένα προ-ηχογραφημένο βίντεο με κινούμενα αντικείμενα.
2. **Εκτελέστε** διαφορά καρέ για την ανίχνευση κίνησης.
3. **Εφαρμόστε** φίλτρο Gaussian για τη μείωση του θορύβου στην ανίχνευση κίνησης.
4. **Εντοπίστε** τα κινούμενα αντικείμενα και **σχεδιάστε** ορθογώνια πλαίσια γύρω τους.

11. Άσκηση 4: Φιλτράρισμα Βάση Χρώματος

Αναπτύξτε μια συνάρτηση στο MATLAB που φιλτράρει όλα τα χρώματα σε ένα βίντεο εκτός από ένα συγκεκριμένο εύρος χρωμάτων. Μετατρέψτε τις υπόλοιπες περιοχές του βίντεο σε ασπρόμαυρο.

Οδηγίες:

1. **Φορτώστε** ένα αρχείο βίντεο.
2. **Μετατρέψτε** κάθε καρέ στον χρωματικό χώρο HSV.
3. **Εφαρμόστε** μια μάσκα για την απομόνωση του συγκεκριμένου εύρους χρωμάτων.
4. **Μετατρέψτε** τις περιοχές του καρέ που δεν καλύπτονται από τη μάσκα σε ασπρόμαυρο.
5. **Εμφανίστε** και **αποθηκεύστε** το επεξεργασμένο βίντεο.

12. Συμπεράσματα και Προτάσεις για Εμβάθυνση

Συμπεράσματα: Η επεξεργασία βίντεο με χρήση MATLAB προσφέρει σημαντικά εργαλεία για την ανάλυση και βελτίωση της ποιότητας των βίντεο. Οι φοιτητές, μέσα από τις ασκήσεις, κατανόησαν τη σημασία της προ επεξεργασίας εικόνας, καθώς και των διαφόρων φίλτρων που επηρεάζουν τη δομή και την ποιότητα των καρέ. Η μείωση της πληροφορίας μέσω της γκρι κλίμακας συνέβαλε στη συμπίεση του αρχείου, ενώ η χρήση φίλτρων ενίσχυσε την οπτική ποιότητα.

Προτάσεις για Εμβάθυνση:

6. **Περαιτέρω πειραματισμός:** Οι φοιτητές μπορούν να δοκιμάσουν άλλες τεχνικές επεξεργασίας εικόνας, όπως μορφολογικές λειτουργίες ή adaptive filtering.
7. **Σύγκριση διαφορετικών αλγορίθμων:** Ανάλυση της απόδοσης διαφορετικών μεθόδων συμπίεσης και των επιπτώσεών τους στην ποιότητα του βίντεο.

8. **Ανάπτυξη αυτόματων διαδικασιών:** Δημιουργία script που εφαρμόζουν διαφορετικά φίλτρα ανάλογα με το περιεχόμενο του βίντεο.
9. **Εφαρμογή μηχανικής μάθησης:** Διερεύνηση τεχνικών ενίσχυσης εικόνας μέσω αλγορίθμων τεχνητής νοημοσύνης για τη βελτίωση χαμηλής ποιότητας βίντεο.
10. **Εφαρμογές στην πραγματική ζωή:** Εξέταση του πώς η επεξεργασία βίντεο μπορεί να βελτιώσει εφαρμογές όπως η αναγνώριση αντικειμένων, η παρακολούθηση κυκλοφορίας και η βελτίωση παλαιών ταινιών.

