



ΤΕΙ ΚΡΗΤΗΣ
ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

Αναγνώριση Αντικειμένου σε Εικόνα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΤΣΑΛΗ ΜΑΡΙΝΑ

2008

ΠΕΡΙΛΗΨΗ

Η φωτογραφία θεωρείτε ένα αρκετά διαδεδομένο μέσο επικοινωνίας για τον 20 ο αιώνα, καθώς και μια μορφή τέχνης συγγενική με την ζωγραφική. Με την εξέλιξη της τεχνολογίας και ειδικότερα της πληροφορικής, παρατηρούμε ότι η εικόνα είναι πλέον ένας σημαντικός τρόπος μετάδοσης πληροφορίας, μετά απο ανάλογη επεξεργασία. Η πιο απλή μορφή επεξεργασίας είναι η αποτύπωση της στον τύπο εικόνας που απαιτείται κάθε φορά, έτσι ώστε να γίνει ευκολότερη η περεταίρω επεξεργασία της στα ανάλογα προγράμματα χρήσης.

Ο ανθρώπινος εγκέφαλος καθημερινά γίνεται δέκτης εκατομμυρίων εικόνων από τις οποίες πολύ λίγες είναι ικανές να του τραβήξουν την προσοχή ώστε να τις επεξεργαστεί περισσότερο. Η εικόνα και γενικότερα η επεξεργασία της μπορεί να παρατηρηθεί σε πολλά επαγγέλματα και μπορεί να θεωρηθεί σαν εργαλείο σε πολλούς τομείς της ζωής μας. Έχοντας γνώση λοιπόν της ανάγκης για εκτενέστερη και λεπτομερέστατη επεξεργασία σε κάποιους τομείς, η διπλωματική εργασία στοχεύει στην μελέτη της δυνατότητα υλοποίησης κάποιου συστήματος, το οποίο κάθε γ χρόνο θα λαμβάνει νέες εικόνες από το περιβάλλον και θα ελέγχει αν στο χώρο αυτό υπάρχει κάποιο αντικείμενο-σκάφος, πάντοτε με βάσει αρχεία εικόνων που υπάρχουν αποθηκευμένα στην μνήμη του.

Πριν την ανάλυση του προγράμματος, είναι σημαντικό να επισημανθούν μερικές πληροφορίες που θα χρειαστούν, όσο αφορά στους τύπους των εικόνων που χρησιμοποιούνται καθώς και τα χαρακτηριστικά τους. Κατά αυτόν τον τρόπο, θα προηγηθεί μια εκτενής αναφορά του τρόπου λειτουργίας της εφαρμογής «Matlab» που χρησιμοποιείται για την υλοποίηση του κώδικα, καθώς και όλες οι απαιτούμενες αναλυτικές πληροφορίες για τις εντολές και τις δομές που είναι άκρως σημαντικές για να έχει αποτέλεσμα η εφαρμογή. Εν συνεχεία, θα παρατηρηθεί το σημείο που θα γίνει ανάπτυξη και παρουσίαση του κώδικα και των πειραμάτων που υλοποιούνται για την αναγνώριση αντικειμένων. Στο τέλος θα παρουσιαστούν τα συμπεράσματα αυτής της μελέτης μετά την υλοποίησή της, μαζί και με κάποιες ιδέες για βελτίωση του κώδικα.

SUMMARY

The photograph is considered a widespread means of communication for the 20 century, as well as a form of art related with the painting. With the development of technology and more specifically the information technology, it is observed that the picture is henceforth an important way of transmitting information. The simplest form of treatment is its imprinting in the exact type of picture we want each time, for further use by the appropriate programs.

The human brain becomes a receptor of millions of pictures daily, from which very few draw our attention. The picture, particularly its exploitation and usage can be observed in many professions and it can be considered as an important device in quite a few sectors in our life. Therefore, having in mind the need for more extensive and detailed exploitation of a picture in certain sectors, it is wise enough to study the possibility of constructing a system which would receive a number of new pictures every y year from the environment and scan them through for any objects-ships, based on the files and pictures stored in its system /memory.

Prior to analyzing the program, it is important to point out certain information regarding the types and characteristics of the pictures used. In this manner, an extensive report will primarily be given on the ways “Matlab” operates, while, at the same time, presenting - in an analytic manner - all the required information on the commands and the structures that are extremely important in order for the system to function. The point of study will then be identified, where all experiments will take place, and once the research has come to an end, all the conclusions will be presented, alongside with some ideas for further study.

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή.....	6
Εικόνες και χαρακτηριστικά τους	9
Οι τύποι των εικόνων	9
Τα BMP αρχεία	10
Τα JPEG ή JPG αρχεία	11
Τα TIF ή TIFF αρχεία.....	12
Τα GIF αρχεία.....	13
Το «εργαλείο» MATLAB	14
<i>Γενικές πληροφορίες για το Matlab</i>	14
Εντολές διαχείρισης του χώρου εργασίας.....	15
Οι εντολές quit και exit.....	15
Οι εντολές clear και clc.....	16
Η εντολή help	16
Οι εντολές who και whos	16
Είσοδος και έξοδος δεδομένων	17
Η εντολή disp.....	17
Η εντολή input.....	17
M-Files	18
Αρχεία συναρτήσεων	19
1. Επικεφαλίδα (header).....	19
2. Σχόλια (comments)	19
3. Εντολές (statements)	19
4. Υποσυναρτήσεις (subfunctions)	20
Προγραμματισμός στην Matlab.....	20
Τύποι τελεστές.....	21
Σχεσιακοί τελεστές.....	21
Λογικοί τελεστές	22
Δομές ελέγχου ροής	22
Βρόχοι for.....	22

Η εντολή if.....	23
Η εντολή switch	24
Άλλες χρήσιμες εντολές.....	25
Οι εντολές break, return και continue.....	25
Η συνάρτηση beep.....	26
Η λογική συνάρτηση find.....	26
Εισαγωγή και Εκτύπωση Δεδομένων	26
Εισαγωγή δεδομένων από τον χρήστη.....	26
Εκτύπωση δεδομένων στην οθόνη.....	26
Παράθυρο διαλόγου.....	27
Χαρακτήρες μετατροπής και απόδρασης	28
Χαρακτήρες μετατροπής (conversion characters)	28
Χαρακτήρες απόδρασης (escape characters).....	29
Εντολές επεξεργασίας εικόνας.....	29
Η εντολή image	29
Η εντολή imread	31
Η εντολή imshow	32
Η εντολή imshow	32
Αλγόριθμοι Αναγνώρισης Αντικειμένων	35
Είσοδος αντικειμένου σε στατικό περιβάλλον έρευνας.....	35
ΠΕΙΡΑΜΑ 1:.....	36
ΠΕΙΡΑΜΑ 2:.....	38
ΠΕΙΡΑΜΑ 3:.....	41
ΠΕΙΡΑΜΑ 4:.....	43
ΠΕΙΡΑΜΑ 5:.....	47
ΠΕΙΡΑΜΑ 6 :.....	48
Είσοδος αντικειμένου σε μεταβλητό περιβάλλον έρευνας	51
Προτάσεις Βελτίωσης	53

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Καθώς η ανθρωπότητα έχει περάσει το κατώφλι της τρίτης χιλιετίας, η εποχή της πληροφορικής και του ηλεκτρονικού υπολογιστή είναι ένα αδιαμφισβήτητο γεγονός. Η Πληροφορική και τα συναφή της παρακλάδια, έχουν διαπεράσει κάθε δραστηριότητα του ανθρώπου. Ποτέ προηγουμένως στην ανθρώπινη ιστορία οι εξελίξεις σε μια επιστήμη δεν είχαν τέτοια μεγάλη επιρροή και ποτέ προηγουμένως οι εξελίξεις δεν ήταν τόσο ραγδαίες. Οι παντός είδους υπολογιστές, τα εργαλεία και οι μηχανές που βοήθησαν και βοηθούν τον άνθρωπο να υπολογίζει και να επεξεργάζεται στοιχεία, αποτελούν το υπόβαθρο της Πληροφορικής. Η συνεχής ανάπτυξη της τεχνολογίας και η πρόοδος, η εξέλιξη της πληροφορικής και του ηλεκτρονικού υπολογιστή, ως είναι φυσικό, συνεπάγεται την αύξηση των αναγκών του ανθρώπου και την ανάγκη του να χρησιμοποιεί όλες εκείνες τις ανέσεις που του παρέχει αυτή η τεχνολογική εξέλιξη, όπως για παράδειγμα την επεξεργασία εικόνων μέσω ηλεκτρονικού υπολογιστή.

Πιο συγκεκριμένα, ο άνθρωπος σε καθημερινή βάση γίνεται δέκτης εκατομμύρια εικόνων, μερικών εκ των οποίων περνάνε από μπροστά του απαρατήρητες, ενώ άλλες πιο ουσιαστικές κατά την κρίση του εκάστοτε παρατηρητή, οδηγούν σε σημαντικά συμπεράσματα μετά από συγκεκριμένη επεξεργασία και ανάλυση. Στα πλαίσια της εικόνας μπορούν να συμπεριληφθούν απλές σκηνές της καθημερινότητας όπως για παράδειγμα ένα εικόνο-μήνυμα, ένα e-mail που θα δεχτεί κάποιος το οποίο θα περιέχει φωτογραφίες, μια εικόνα που βλέπει κάποιος από το παράθυρο του, ένα άρθρο στην εφημερίδα με φωτογραφίες, μια εικόνα στην τηλεόραση, μια φωτογραφία σε ένα κάδρο.

Πέρα αυτών των απλών καθημερινών εικόνων που επεξεργάζεται ο ανθρώπινος εγκέφαλος καθημερινά, υπάρχουν εργασίες που στηρίζονται αποκλειστικά και μόνο πάνω σε επεξεργασία και μελέτη φωτογραφιών και εικόνων. Μια τέτοια εργασία θα μπορούσε κάποιος να χαρακτηρίσει την δουλειά του φωτογράφου, ο οποίος καθημερινά δημιουργεί οπτικές εικόνες μέσω της καταγραφής και αποτύπωσης του φωτός, με χρήση κατάλληλων συσκευών. Η φωτογραφία, πέρα από την τεχνική της διάσταση, αναγνωρίζεται ως ένα από τα ευρύτερα διαδεδομένα μέσα επικοινωνίας του 20ου αιώνα καθώς και ως μία μορφή τέχνης συγγενικής με την ζωγραφική.

Θα μπορούσε κανείς να εντάξει σε αυτήν την κατηγορία τα ιατρικά επαγγέλματα, όπου καθημερινά έχουν να μελετήσουν πάρα πολλές εικόνες, όπως ακτινογραφίες, μαγνητικές και άλλου είδους εξετάσεις κατά τις οποίες απομονώνουν από τις «εικόνες» το σημείο εκείνο που τους ενδιαφέρει, όπως για παράδειγμα ένα ζωτικό όργανο, το οποίο το επεξεργάζονται, με σκοπό την εύρεση της διαφορετικότητας τού, έτσι ώστε να αντιληφθούν το πρόβλημα που υπάρχει.

Άλλες εφαρμογές της επεξεργασίας εικόνας είναι η γραφιστική, η οποία είναι μια μορφή οπτικής επικοινωνίας. Θεωρείτε μια διαδικασία στην οποία οι οπτικές πληροφορίες διαμορφώνονται σε ένα σχήμα, ώστε να μεταφερθεί ένα μήνυμα.

Ένας άλλος κλάδος που χρησιμοποιεί επεξεργασία εικόνων θα μπορούσε να χαρακτηριστεί το επάγγελμα που έχει να κάνει με τον τομέα της ασφάλειας. Πάνω σε αυτό το σύστημα παρακολούθησης θα μπορούσε να είναι ένα σύστημα κατάλληλα προσαρμοσμένο, το οποίο θα δέχεται τις εικόνες και θα τις επεξεργάζεται δίνοντας τελικά τα αποτελέσματα που του έχουν προσδιορίσει. Παράδειγμα αυτών των συστημάτων θα μπορούσαμε να χαρακτηρίσουμε την παρακολούθηση ενός μαγαζιού το βραδύ, μιας τράπεζα, ενός μεγάλου κυκλοφοριακού δρόμου, ενός λιμανιού, ενός αεροδρομίου.

Όλα αυτά θα μπορούσαν να αποτελούν περιοχές τις οποίες, μετά από εντολή, θα μπορούσε να έλεγχε το σύστημα 'Αναγνώριση Αντικειμένου σε εικόνα – με Matlab' για τυχόν κλέφτες, κάποιο όπλο, κάποιο δυστύχημα, κάποιο πλοίο ή κάποιο αεροπλάνο, έτσι ώστε να μπορέσει να ενημερώσει τον χρήστη για την ύπαρξη αυτού που αναζητάμε κάθε φορά και αυτός να δράσει ανάλογα.

Λαμβάνοντας λοιπόν υπόψη, κάποιες από αυτές τις εφαρμογές έγινε προσπάθεια για την υλοποίηση ενός συστήματος με το οποίο, χρησιμοποιώντας μια φωτογραφική μηχανή - η οποία αποτυπώνει φωτογραφίες κάθε γ χρόνο - θα μπορούσε ένα λιμάνι ή κάποιος προσωπικός χώρος στην παραλία να τεθεί υπό παρακολούθηση. Έτσι υποθέτοντας ότι ο χώρος που παρακολουθείται είναι μια θάλασσα η οποία τελειώνει εκεί που αρχίζει ο ουρανός - χωρίς αρχικά να υπάρχει κανένα άλλο αντικείμενο, με την παρακολούθηση, θα επισημανθεί στον χρήστη οποιοδήποτε αντικείμενο διέλθει στον παρακολουθούμενο χώρο.

Έχοντας υπόψη μας τα παραπάνω παραδείγματα για την χρήση και επεξεργασία εικόνας, αντιλαμβανόμαστε την ανάγκη να υπάρχουν αυτά τα μέσα επεξεργασίας. Αν δεν υπήρχαν θα ήταν πολύ δύσκολο να υλοποιήσουμε τέτοιου είδους εφαρμογές χωρίς την χρήση ηλεκτρονικών μέσων, αφού για να γίνει η υλοποίηση τους θα χρειαζόταν να υπάρχουν αρκετά άτομα καθ' όλη την περίοδο

μελέτης ώστε να ελέγχουν κάθε φορά το περιβάλλον παρακολούθησης, αυτό που χρειάζεται να κάνει τώρα ένας ηλεκτρονικός υπολογιστής μέσω της επεξεργασίας εικόνας.

Έχοντας κατανοήσει τις ανάγκες που υπάρχουν για αναγνώριση εικόνων , στα επόμενα κεφάλαια θα παρουσιάσουν μια μελέτη που αφορά στην αναγνώριση αντικειμένων σε εικόνα. Στο κεφάλαιο 2 αναλύονται μερικές σημαντικές πληροφορίες που είναι χρήσιμες όσον αφορά στους τύπους των εικόνων που χρησιμοποιήθηκαν καθώς και τα χαρακτηριστικά τους. Στο κεφάλαιο 3, γίνεται μια εκτενής αναφορά στις λειτουργίες της εφαρμογής Matlab που χρησιμοποιούνται για την υλοποίηση του κώδικα καθώς επίσης και αναλυτική παρουσίαση των σημαντικότερων πληροφοριών των εντολών και δομών που συνέδραμαν στην εφαρμογή. Εν συνεχεία, στο κεφάλαιο 4 γίνεται η ανάπτυξη και παρουσίαση του κώδικα και των πειραμάτων που υλοποιούνται για την αναγνώριση αντικειμένων. Τέλος στο κεφάλαιο 5, εφόσον η μελέτη της διπλωματικής αυτής εργασίας έχει λάβει χώρα, παρουσιάζονται τα συμπεράσματα και οι τρόποι βελτίωσης και κάποια σύντομη βιβλιογραφία η οποία βοήθησε για την μελέτη και υλοποίηση του συστήματος.

ΚΕΦΑΛΑΙΟ 2

Εικόνες και χαρακτηριστικά τους

Η ψηφιακή φωτογραφία αποτελεί ίσως την τελευταία σημαντική εξέλιξη σε ότι αφορά την τεχνική της φωτογραφίας. Σήμερα η ψηφιακή φωτογραφία βασίζεται πολύ στην χρήση ηλεκτρονικών μέσων. Σε κάθε φωτογραφία που αποτυπώνεται με την χρήση της κάμερας συντελείτε ένας πρωταρχικός έλεγχος – επεξεργασία για την σωστή αποτύπωση των χρωμάτων από την ίδια την συσκευή αποτύπωσης. Οι πληροφορίες των χρωμάτων μεταφέρονται στα ειδικά κυκλώματα της κάμερας με δυαδική μορφή τα οποία επεξεργάζονται και αποθηκεύουν την εικόνα σε μορφή αναγνώσιμη από άλλα μέσα. Η μορφή αυτή είναι μία σειρά δυαδικών αριθμών κατάλληλα οργανωμένων. Στην συνέχεια αυτή είναι έτοιμη να αναπαραχθεί όπου χρειάζεται, με την βοήθεια αποκωδικοποιητών της μορφής αποθήκευσης. Οι κυριότερες μορφές αποθήκευσης σε ψηφιακά μέσα είναι οι [jpeg](#), [gif](#), [tiff](#). Αυτή η ενότητα θα παρουσιάσει τα βασικά αυτών των μορφών αποθήκευσης καθώς και των εικόνων με την μορφή αποτύπωσης [bmp](#).

Οι τύποι των εικόνων

Αυτό το κομμάτι του βιβλίου αναφέρει μερικούς από τους πιο διαδεδομένους και περισσότερο χρησιμοποιούμενους τύπους εικόνων αναλύοντας τα βασικά τους στοιχεία, θεσπίζοντας με αυτόν τον τρόπο, μια πολύπλευρη άποψη για το θέμα των εικόνων.

Ο πίνακας που ακολουθεί περιλαμβάνει ένα μεγάλο αριθμό τύπων εικόνων που υπάρχουν και μπορεί να επεξεργαστεί το πρόγραμμα που θα χρησιμοποιηθεί για την υλοποίηση του κώδικα της μελέτης αυτής.

Format	Full Name	Variants
'bmp'	Windows Bitmap (BMP)	1-bit, 8-bit, and 24-bit uncompressed images
'gif'	Graphics Interchange Format (GIF)	8-bit images
'hdr'	Hierarchical Data Format (HDR)	8-bit raster image data sets, with or without associated colormap, 24-bit raster image data sets; uncompressed or with RLE or JPEG compression
'jpg' or 'jpeg'	Joint Photographic Experts Group (JPEG)	8-bit, 12-bit, and 16-bit Baseline JPEG images <div style="border: 1px solid black; padding: 2px; width: fit-content;">Note Indexed images are converted to RGB before writing out JPEG files, because the JPEG format does not support indexed images.</div>
'pnm'	Portable Bitmap (PBM)	Any 1-bit PBM image, ASCII (plain) or raw (binary) encoding
'pcx'	Windows Paintbrush (PCX)	8-bit images
'pgm'	Portable Graymap (PGM)	Any standard PGM image, ASCII (plain) encoded with arbitrary color depth, raw (binary) encoded with up to 16 bits per gray value
'png'	Portable Network Graphics (PNG)	1-bit, 2-bit, 4-bit, 8-bit, and 16-bit grayscale images; 8-bit and 16-bit grayscale images with alpha channels; 1-bit, 2-bit, 4-bit, and 8-bit indexed images; 24-bit and 48-bit truecolor images; 24-bit and 48-bit truecolor images with alpha channels
'ppm'	Portable Anymap (PNM)	Any of the PPM/PGM/PBM formats, chosen automatically
'ppm'	Portable Pixmap (PPM)	Any standard PPM image, ASCII (plain) encoded with arbitrary color depth; raw (binary) encoded with up to 16 bits per color component
'ras'	Sun Raster (RAS)	Any RAS image, including 1-bit bitmap, 8-bit indexed, 24-bit truecolor and 32-bit truecolor with alpha
'tif' or 'tiff'	Tagged Image File Format (TIFF)	Baseline TIFF images, including 1-bit, 8-bit, 16-bit, and 24-bit uncompressed images; 1-bit, 8-bit, 16-bit, and 24-bit images with packbits compression; 1-bit images with CCITT 1D, Group 3, and Group 4 compression; CIELAB, ICCLAB, and CMYK images
'xwd'	X Windows Dump (XWD)	8-bit ZPixmap

Η πρώτη στήλη περιέχει την μορφή αποθήκευσης κάθε τύπου εικόνας . Η επόμενη στήλη που ακολουθεί περιέχει το συνολικό όνομα αυτής της μορφής, πώς αναφέρονται ευρύτερα αυτά τα αρχεία. Και τέλος η τελευταία στήλη αναφέρει την χωρητικότητα κάθε αρχείου που μπορεί να καταλάβει, κατά την δημιουργία του. Έτσι για παράδειγμα το bmp είναι η επέκταση μιας φωτογραφίας τύπου Windows Bitmap (BMP) η οποία καταλαμβάνει σε χωρητικότητα 1-bit ή 8-bit ή 24-bit ασυμπίεστης εικόνας.

Τα BMP αρχεία

Τα αρχεία BMP (Bitmap) είναι ο πιο διαδεδομένος τρόπος αποθήκευσης εικόνων στον κόσμο των προσωπικών υπολογιστών (PC). Χρησιμοποιούνται από τα Microsoft-Windows για την αποθήκευση και χρήση εικόνων. Δεν είναι συμπιεσμένα ούτε έχουν απλή συμπίεση και αυτό τα κάνει εύκολα στον χειρισμό τους και επιταχύνει τις εργασίες ανάγνωσης και εγγραφής στον δίσκο. Μπορούν να αποθηκεύσουν εικόνες με 1 (μονόχρωμες), 4 (16 χρώματα), 8 (256 χρώματα) και 24 bit (16 εκατομμύρια χρώματα) ανά pixel.

Ένα αρχείο BMP περιέχει μια επικεφαλίδα που περιλαμβάνει πληροφορίες για το BMP αρχείο (file header), στην συνέχεια πληροφορίες για την αποθηκευμένη εικόνα (bitmap-info header), μετά μια παλέτα με τα χρώματα που χρησιμοποιούνται

στην εικόνα μας (η παλέτα είναι δυνατόν να μην υπάρχει) και τέλος την αποθηκευμένη εικόνα.

Η παλέτα είναι ένας πίνακας με το σύνολο των χρωμάτων που χρησιμοποιούνται στην εικόνα μας. Χρησιμοποιείται για να μειωθούν οι απαιτήσεις σε χώρο, για την αναπαράσταση εικόνων διατηρώντας όσο το δυνατόν την ακριβή απεικόνιση των χρωμάτων. Παλέτες έχουν συνήθως οι εικόνες με βάθος χρώματος 1bit, 4 bit και 8 bit. Για τις εικόνες αυτές η παλέτα περιέχει το πολύ 2, 16 και 256 χρώματα αντίστοιχα. Η εικόνα αποθηκεύεται με τη μορφή δεικτών στα στοιχεία της παλέτας.

Οι εικόνες με 24 bit/χρώμα αποθηκεύουν για κάθε χρώμα ξεχωριστά τις τιμές του κόκκινου, του πράσινου και του μπλε για κάθε pixel. Σε αυτές τις εικόνες αν υπάρχει παλέτα είναι για να αντικατασταθεί η παλέτα που χρησιμοποιείται από το σύστημα απεικόνισης εικόνων.

Τα JPEG ή JPG αρχεία

Το JPEG (Joint Photographic Experts Group) είναι μια τεχνική συμπίεσης με απώλειες για τις εικόνες χρώματος. Ο JPEG τύπος είναι μια ομάδα ISO/IEC εμπειρογνομόνων που δημιουργεί και διατηρεί τα πρότυπα για μια ακολουθία του αλγορίθμου συμπίεσης για τα αρχεία εικόνας υπολογιστών.

Τα πρότυπα JPEG βοηθούν τα αρχεία εικόνας για να συμπιεστούν σε περίπου πέντε τοις εκατό του αρχικού μεγέθους τους, αν και μερικές λεπτομέρειες χάνονται κατά τη διάρκεια της συμπίεσης. Τα αρχεία συνήθως που χρησιμοποιούνται είναι με τις εξής επεκτάσεις .jpg, .jpeg, .jfif, ή .jpe.

Η επέκταση JPEG διευκρινίζει μόνο πώς μια εικόνα μετασχηματίζεται σε μια ακολουθία από bytes.

Το JPEG είναι ένα από τα δημοφιλέστερα πρότυπα συμπίεσης εικόνας που χρησιμοποιούνται για την αποθήκευση και τη διαβίβαση των εικόνων στο διαδίκτυο. Ο κύριος λόγος πίσω από αυτήν την δημοτικότητα, είναι η εξαιρετικά αποτελεσματική συμπίεση που προσφέρεται από τα σχήματα αρχείων JPEG. Αυτή η συμπίεση επιτρέπει στους ανθρώπους να διαβιβάζουν γρήγορα (να αποστέλλουν ή να λαμβάνουν) τα αρχεία εικόνας μέσω του Διαδικτύου.

Το JPEG αποδίδει καλύτερα στις φωτογραφίες και τα έργα ζωγραφικής των ρεαλιστικών σκηνών με τις ομαλές παραλλαγές του χρώματος και του τόνου. Το JPEG δεν χρησιμοποιείται συχνά για τα διαγράμματα, τα σχέδια γραμμών, ή άλλες εικονικές γραφικές παραστάσεις. Αυτό είναι επειδή η μέθοδος συμπίεσης που χρησιμοποιείται στις εικόνες JPEG μπορεί να τις διαστρεβλώσει.

Το JPEG είναι μια μέθοδος συμπίεσης, με απώλειες, διενεργηθείσα να χρησιμοποιεί τον ιδιαίτερο μετασχηματισμό συνημίτονου, όπου κάποιο στοιχείο από την αρχική εικόνα χάνεται. Αν και το ποσό συμπίεσης εξαρτάται από την αρχική εικόνα, οι αναλογίες 10:1 ή 20:1 δεν προκαλούν χαρακτηριστικά την αξιοπρόσεχτη απώλεια στην αρχική εικόνα.

Στη μέθοδο συμπίεσης JPEG, η συμπίεση γίνεται με το διαχωρισμό της αρχικής εικόνας στους μικρούς φραγμούς εικονό-κυττάρου, οι οποίοι διχοτομούνται επανειλημμένως για να επιτύχουν το επιθυμητό ποσό συμπίεσης. Οι εικόνες JPEG μπορούν να δημιουργηθούν χρησιμοποιώντας το λογισμικό ή το υλικό. Το υλικό συμπίεσης JPEG παρέχει την ικανοποιητική ταχύτητα σε πραγματικό χρόνο, σε ώρα πτήσης συμπίεσης.

Όταν ένα αρχείο JPEG δημιουργείται ή μια εικόνα μετατρέπεται από ένα άλλο σχήμα σε JPEG, ο χρήστης πρέπει να διευκρινίσει την επιθυμητή ποιότητα της εικόνας. Εάν ο χρήστης επιλέγει να δημιουργήσει την υψηλότερη πιθανή ποιοτική εικόνα, το μέγεθος αρχείων θα είναι μεγάλο. Εάν ο χρήστης είναι πρόθυμος να θυσιάσει την ποιότητα εικόνας, το προκύπτον αρχείο JPEG μπορεί να είναι πολύ μικρότερο.

Τα TIF ή TIFF αρχεία

Τα αρχεία TIFF (Tagged Image File Format) είναι αρχεία που αναπτύχθηκαν από τις εταιρίες Aldus και Microsoft και χρησιμοποιούνται ευρέως για την ανταλλαγή εικόνων. Υποστηρίζουν εικόνες με τρία χρωματικά κανάλια και βάθος χρώματος έως και 48bits/pixel. Τα TIFF αρχεία αποτελούνται από μια επικεφαλίδα στην οποία ορίζονται οι διαστάσεις της εικόνας, τα χρωματικά κανάλια της (αν είναι RGB ή μόνο Gray) και τέλος από ένα κωδικό αριθμό που χαρακτηρίζει τη μορφή της συμπίεσης.

Τα GIF αρχεία

Η μορφή αυτών των αρχείων GIF (Graphics Interchange Format) αναπτύχθηκε από την CompuServe με στόχο την διευκόλυνση της ανταλλαγής εικόνων μέσω δικτύου. Το GIF χρησιμοποιεί συμπίεση χωρίς απώλεια πληροφορίας για τη μείωση του όγκου των ψηφιακών εικόνων, αλλά υποστηρίζει μόνο 256 χρώματα.

Ο συγκεκριμένος τύπος είναι ιδανικός για να λειτουργεί χωρίς απώλειες, για εικόνες βάθους 8bits και για εικόνες με πολλές ακμές και γωνίες όπως γραμμικά σχέδια. Γενικά η συμπίεση GIF είναι πιο αποτελεσματική σε εικόνες που έχουν μεγάλες περιοχές με ένα μόνο χρώμα. Αντίθετα δεν είναι κατάλληλος σαν τύπος για εικόνες με πολλά χρώματα γιατί κατά την συμπίεση χαλάει η ποιότητα της εικόνας.

ΚΕΦΑΛΑΙΟ 3

Το «εργαλείο» MATLAB

Σκεφτόμενοι την υλοποίηση του όλου πλάνου για επεξεργασία και αναγνώριση εικόνων και έχοντας μια μικρή γνώση όσο αφορά το «εργαλείο» Matlab θεωρήσαμε ότι θα μπορούσαμε να το χρησιμοποιήσουμε για να υλοποιήσουμε το κομμάτι της επεξεργασίας .

Επειδή γενικά είναι ένα εργαλείο που μπορεί να χρησιμοποιηθεί σε πολλά πράγματα και ως πρόγραμμα πέρα κάποιων ατόμων που ανήκουν στην φοιτητική κοινότητα δεν είναι και τόσο διαδεδομένο στον υπόλοιπο κόσμο, θεωρήσαμε αναγκαίο να προχωρήσουμε σε μια μικρή επανάληψη για όσους έχουν κάποιες γνώσεις και μια γρήγορη ενημέρωση για τους νέους χρήστες.

Γενικές πληροφορίες για το Matlab

Το MATLAB είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού και ένα αλληλεπιδραστικό περιβάλλον για την ανάλυση στοιχείων και την ανάπτυξη αλγορίθμων και εφαρμογών.

Είναι ένα από τα δημοφιλέστερα προγραμματιστικά πακέτα στην ακαδημαϊκή κοινότητα και μπορούμε να το συναντήσουμε σε πολλά εκπαιδευτικά ιδρύματα που ασχολούνται με τομείς εξειδικευμένους στην τεχνολογία.

Ένα από τα πιο σημαντικά εργαλεία κατά την χρήση του MATLAB είναι η πλούσια βοήθεια που σου παρέχει για την χρήση και την λειτουργία του, με ένα εύκολο και πολύ πρακτικό τρόπο καθώς και η εύρεση πολλών εγγράφων σε μορφή PDF από την ίδια την mathworks τα οποία καλύπτουν εκτενώς όλα τα toolboxes και τις λειτουργίες του. Αυτά μπορούν να βρεθούν στο site www.mathworks.co.uk. Και μέσω της εφαρμογής μπορούμε εύκολα πληκτρολογώντας την λέξη help και το όνομα της εντολής που μας ενδιαφέρει ή από το μενού βοήθειας της.

Όπως όλα τα προγράμματα έτσι και το MATLAB περιέχει τις γνωστές σε όλους γραμμές εργαλείων με τα αντίστοιχα μενού. Εκτός όμως αυτών των βασικών εργαλείων κάθε πρόγραμμα χρησιμοποιεί και δικά του εργαλεία. Η διπλωματική αυτή εργασία ωστόσο, θα επικεντρωθεί μόνο σε ένα μικρό μέρος, που αφορά στο πρόγραμμα μελέτης.

Εντολές διαχείρισης του χώρου εργασίας

Η παράγραφος αυτή παρουσιάζει τις χρήσιμες εντολές για τη διαχείριση του παραθύρου εργασίας και των ενεργών μεταβλητών που δημιουργήθηκαν για την μελέτη. Αυτές φαίνονται στον πίνακα που ακολουθεί.

Εντολή	Ερμηνεία
exit, quit	έξοδος από το πρόγραμμα
clear	διαγραφή ενεργών μεταβλητών
clc	καθαρισμός παραθύρου εργασίας
help	Βοήθεια
who, whos	κατάλογος ενεργών μεταβλητών εργασίας

Αξίζει να σημειωθούν οι εξής βασικές οδηγίες για το παράθυρο εντολών:

- Στην *MATLAB* υπάρχει διάκριση μεταξύ μικρών και κεφαλαίων γραμμάτων (οι μεταβλητές *A* και *a* είναι διαφορετικές μεταξύ τους).
- Όταν το όνομα μιας μεταβλητής αναγράφεται, η *MATLAB* τυπώνει στην οθόνη την τιμή της.
- Όταν το σύμβολο ';' στο τέλος μιας εντολής, το αποτέλεσμα της δεν τυπώνεται στην οθόνη.
- Μια ακολουθία εντολών της *MATLAB* μπορεί να γραφτεί σε μια γραμμή χωριζόμενη με κόμματα ή ερωτηματικά.
- Πατώντας τα πλήκτρα με τα πάνω και κάτω βέλη ([↑] και [↓]) μπορεί κανείς να διατρέξει όλες τις προηγούμενες εντολές. Επίσης μια προηγούμενη εντολή μπορεί να επαναληφθεί αν πληκτρολογηθούν τα πρώτα γράμματα και μετά πατηθεί το πλήκτρο με το πάνω βέλος [↑].

Οι εντολές quit και exit

Με τις εντολές **quit** και **exit** τερματίζεται η τρέχουσα εργασία. Πολλές φορές είναι καλό να αποθηκεύεται η εργασία που γίνεται ή/και τις τιμές των μεταβλητών που δημιουργούνται στην εκάστοτε εργασία.

Οι εντολές `clear` και `clc`

Η εντολή `clear var` διαγράφει τη μεταβλητή `var` του χώρου εργασίας. Η εντολή `clear` διαγράφει όλες τις ενεργές μεταβλητές. Το ίδιο κάνει και η εντολή `clear variables`. Αν θέλει κάποιος να διαγράψει μόνο τις μεταβλητές `var1`, `var2` και `var3` μπορεί να γράψει `clear var1 var2 var3` (οι `var1`, `var2` και `var3` χωρίζονται με κενά και όχι με κόμματα). Αν θέλει να διαγράψει όλες τις μεταβλητές που αρχίζουν από `Z` γράφει `clear Z*`.

Αν τέλος το όνομα της προς-διαγραφή μεταβλητής είναι αποθηκευμένο σε αλφαριθμητικό, για παράδειγμα τη `name`, μπορεί κανείς να χρησιμοποιήσει τη συναρτησιακή μορφή της εντολής `clear(name)`. Η εντολή `clc` καθαρίζει απλώς το παράθυρο εργασίας (δεν διαγράφονται οι μεταβλητές), ώστε να παρέχει την ευελιξία και την δυνατότητα ανάπτυξης του νέου κώδικα που παράγεται και να μπορεί κάποιος να τον βλέπει χωρίς να υπάρχουν υπολείμματα στην οθόνη άλλων εντολών.

Η εντολή `help`

Έχει ήδη χρησιμοποιηθεί η εντολή αυτή (βοήθεια). Η MATLAB είναι εφοδιασμένη με ένα εκτεταμένο on-line σύστημα βοήθειας. Η εντολή `help topic` παρέχει βοήθεια για το θέμα - `topic`. Κάποιος θα μπορούσε να αρχίσει με την εντολή `help help` η οποία εξηγεί πως λειτουργεί το σύστημα βοήθειας και αναφέρει επίσης κάποιες σχετικές εντολές. Γράφοντας απλά `help` μπορεί κανείς να λάβει έναν κατάλογο θεμάτων για τα οποία υπάρχει διαθέσιμη βοήθεια. Σ' αυτόν τον κατάλογο μπορεί κάποιος να βρει για παράδειγμα το θέμα «`elfun - elementary math functions`» (στοιχειώδεις μαθηματικές συναρτήσεις). Αν πληκτρολογήσει τώρα `help image` παίρνει τον κατάλογο των διαθέσιμων συναρτήσεων που αφορούν τις εικόνες.

Οι εντολές `who` και `whos`

Η εντολή `who` παρέχει απλώς ένα κατάλογο των ενεργών μεταβλητών (χωρίς άλλες πληροφορίες). Η εντολή `whos` μαζί με τον κατάλογο παρέχει πληροφορίες για όλες τις ενεργές μεταβλητές. Υπάρχει επίσης η δυνατότητα να χρησιμοποιηθούν οι δύο εντολές για μεμονωμένες μεταβλητές, όπως φαίνεται πιο κάτω:

`who var1 var2 var3`: κατάλογος των `var1`, `var2` και `var3`

who ab*: κατάλογος των μεταβλητών με όνομα που αρχίζει από *ab*

who *z: κατάλογος των μεταβλητών με όνομα που λήγει σε *z*.

who -file filename: κατάλογος των μεταβλητών που είναι αποθηκευμένες στο αρχείο *filename.mat*

Είσοδος και έξοδος δεδομένων

Η παράγραφος αυτή μελετά τρεις εντολές που έχουν να κάνουν με την απεικόνιση των μεταβλητών και των τιμών τους στην οθόνη, δηλαδή στο παράθυρο εντολών (Command window) της MATLAB. Αυτές φαίνονται στον πιο κάτω πίνακα:

Εντολή	Ερμηνεία
Disp	απεικόνιση μεταβλητών στην οθόνη
Input	υποβολέας για είσοδο δεδομένων

Η εντολή disp

Είναι ήδη γνωστό ότι αν θέλει κανείς να δει την τιμή μιας μεταβλητής *var* στην οθόνη αρκεί να γράψει το όνομα της. Εμφανίζεται τότε στην οθόνη σε μια γραμμή *var =* και μετά ακολουθεί η τιμή της *var* που μπορεί να είναι αριθμός ή πίνακας ή αλφαριθμητικό. Η εντολή `disp(var)` απεικονίζει την τιμή της μεταβλητής *var* (χωρίς το όνομα) στην οθόνη ενώ η εντολή `disp var` απεικονίζει το όνομα της μεταβλητής (χωρίς την τιμή). Επίσης η εντολή `disp('.....')` τυπώνει στην οθόνη το αλφαριθμητικό που περιέχεται μεταξύ των τόνων.

Η εντολή input

Η εντολή `input` έχει τη γενική μορφή **R=input('prompt')** όπου **prompt** ένα αλφαριθμητικό. Με την εντολή αυτή εμφανίζεται στην οθόνη η προτροπή `prompt` και το σύστημα αναμένει από τον χρήστη να εισαγάγει την τιμή της μεταβλητής *R* που μπορεί να είναι αριθμός ή αλφαριθμητικό ή διάνυσμα ή πίνακας ή ακόμα το αποτέλεσμα μιας ολόκληρης παράστασης σε γλώσσα MATLAB. Αν η *R* είναι διάνυσμα ή πίνακας τότε τα στοιχεία του εισάγονται κατά τα γνωστά μέσα σε αγκύλες. Αν η *R* είναι αλφαριθμητικό τότε η τιμή της πρέπει να δοθεί μέσα σε

τόνους ' '. Ένας τρόπος να αποφευχθούν οι τόνοι είναι με το να δηλωθεί μέσω της εντολής ότι η μεταβλητή είναι αλφαριθμητικό. Αυτό γίνεται ως εξής:

```
R=input('prompt', 's')
```

M-Files

Για να εκμεταλλευτεί κανείς πλήρως τις ικανότητες της MATLAB, πρέπει να μάθει πώς να δημιουργεί μεγάλες και συχνά πολύπλοκες ακολουθίες εντολών. Ο καλύτερος τρόπος για να επιτευχθεί αυτός ο στόχος είναι με τη χρήση αρχείων που καλούνται “m-files” αφού έχουν ως κατάληξη (extension) το .m. Τα m-files της MATLAB είναι τα αντίστοιχα των συναρτήσεων (functions) και των υπό-ρουτινών (subroutines) που συναντούνται σε άλλες γλώσσες προγραμματισμού όπως η C. Τα m-files που δημιουργεί ο χρήστης συμπληρώνουν τις συναρτήσεις βιβλιοθήκης της MATLAB που είναι επίσης m-files.

Τα m-files διακρίνονται σε

- **Αρχεία script ή αρχεία εντολών** (script m-files or command files) τα οποία δεν έχουν ορίσματα εισόδου και εξόδου αλλά εκτελούν μια ακολουθία εντολών σε μεταβλητές του χώρου εργασίας, και σε
- **Αρχεία συναρτήσεων** (function m-files) τα οποία περιλαμβάνουν μια γραμμή ορισμού συνάρτησης, δέχονται ορίσματα εισόδου και επιστρέφουν μεταβλητές εξόδου, και των οποίων οι εσωτερικές μεταβλητές είναι τοπικές (εκτός αν δηλωθούν ως ολικές με την εντολή global).

Τόσο τα αρχεία script όσο και τα αρχεία συναρτήσεων

- Δημιουργούνται ξεχωριστά με κάποιο συντάκτη (editor) όπως ο notepad ή ο wordpad ή ακόμα με το συντάκτη της MATLAB:

Για νέο αρχείο: File → New → M-file

Για υπάρχον αρχείο: File → Open

Μπορεί κάποιος, επίσης, να ανοίξει τον editor της MATLAB και με την εντολή

```
>> edit
```

- Πρέπει να βρίσκονται στον φάκελο εργασίας (working directory) ή στον φάκελο (directory) της MATLAB ή στον δρόμο αναζήτησης (research path) της MATLAB.
- Μπορούν να καλούν άλλα m-files ή ακόμα τον ίδιο τον εαυτό τους (αναδρομικά m-files).

Από τους διάφορους τύπους m αρχείων που χρησιμοποιεί η έρευνα αυτή, είναι τα αρχεία συναρτήσεων για αυτό και ακολουθεί εκτενέστερη ανάπτυξη αυτών.

Αρχεία συναρτήσεων

Τα αρχεία συναρτήσεων (function m-files) είναι τα αρχεία που θα χρησιμοποιούνται στο πείραμα της εργασίας αυτής, και περιέχουν μια ολοκληρωμένη ακολουθία εντολών της MATLAB με μεταβλητές εισόδου, input1, input2, με την οποία υπολογίζονται νέες μεταβλητές εξόδου output1, output2,

Η δομή τους είναι η εξής:

1. Επικεφαλίδα (header)

Μια γραμμή της μορφής:

```
function [output1, output2, ...] = filename (input1, input2, ....)
```

Το filename είναι το όνομα του function το οποίο αποθηκεύεται (υποχρεωτικά) στο m-file με όνομα filename.m. Είναι αξιοσημείωτο το γεγονός ότι οι μεταβλητές εισόδου είναι σε παρενθέσεις ενώ οι μεταβλητές εξόδου βρίσκονται σε αγκύλες. Οι τελευταίες δεν είναι απαραίτητες αν κάποιος έχει μόνο μια μεταβλητή εξόδου.

Για παράδειγμα αντί

```
function [out1] = parad(x, y)
```

μπορούμε να γράψουμε

```
function out1 = parad(x, y)
```

2. Σχόλια (comments)

Αυτά ξεκινούν αναγκαστικά με το σύμβολο % και είναι προαιρετικά. Τα σχόλια μετά την επικεφαλίδα αποτελούν και το κείμενο βοήθειας για το function. Με διαφορετικά λόγια, αυτά εμφανίζονται αν γράψουμε:

```
>> help filename
```

3. Εντολές (statements)

Ακολουθία εντολών της MATLAB με την οποία υπολογίζονται οι μεταβλητές εξόδου

Προκαταρκτικές εντολές
output1 =
output2 =
Κλπ

4. Υποσυναρτήσεις (subfunctions)

Αυτές είναι εσωτερικές συναρτήσεις functions που περιέχονται στο αρχείο filename.m. Αναγνωρίζονται μόνο από τα functions που περιέχονται στο αρχείο αυτό. Ορίζονται με τον ίδιο ακριβώς τρόπο, π.χ.

`function [out1, out2,] = subfname (x, y, z)`

Οι συναρτήσεις βιβλιοθήκης της MATLAB, όπως οι sin, max, exp και size, είναι function m-files.

Το όνομα μιας νέας function δεν πρέπει να συμπίπτει με όνομα συνάρτησης βιβλιοθήκης.

Γενικά για την ονομασία μιας m-συνάρτησης ακολουθούνται οι κανόνες ονοματολογίας που ισχύουν και για τις μεταβλητές:

- Το όνομα αρχίζει με γράμμα (του αγγλικού αλφαβήτου).
- Το όνομα περιέχει μόνο γράμματα, αριθμούς και υπό-παύλες (underscore).
- Δεν χρησιμοποιούνται ονόματα που έχουν δεσμευτεί από τη MATLAB (π.χ. συναρτήσεις βιβλιοθήκης και εργαλειοθηκών).
- Προτιμούνται μικρά ονόματα για πρακτικούς λόγους αν και δεν υπάρχει περιορισμός στο μήκος των ονομάτων.

Προγραμματισμός στην Matlab

Για να μπορέσουν να διατυπωθούν νέα, προχωρημένα προγράμματα (δηλ. m-files που θα παίζουν το ρόλο “νέων εντολών” στη MATLAB) πρέπει να

μελετηθούν καλά οι τέσσερις **δομές ελέγχου ροής** (flow control structures) που διαθέτει η MATLAB:

- **βρόχοι for** (for loops)
- **εντολή if** (if statement)
- **εντολή switch** (switch statement)

Η διπλωματική αυτή εργασία προτού προβεί στις δομές ελέγχου ροής θα αναφερθεί πρώτα στους λεγόμενους **σχεσιακούς** (relational operators) και **λογικούς τελεστές** (logical operators). Ένας από τους τύπους δεδομένων της MATLAB είναι ο **λογικός** (logical data type) με πιθανές τιμές τη λογική μονάδα 1 (αληθής, true) και το λογικό 0 (ψευδής, false). Οι λογικές μεταβλητές παράγονται από σχεσιακούς και λογικούς τελεστές ή συναρτήσεις και από τις συναρτήσεις **true** και **false**.

Τύποι τελεστές

Σχεσιακοί τελεστές

Οι **σχεσιακοί τελεστές** (relational operators) της MATLAB είναι οι πιο κάτω:

<	less than	μικρότερος
>	greater than	μεγαλύτερος
<=	less than or equal	μικρότερος ή ίσος
>=	greater than or equal	μεγαλύτερος ή ίσος
==	equal	ίσος
~=	not equal	άνισος

Παρατηρείται ότι σε αντίθεση με την εκχώρηση τιμής όπου χρησιμοποιείται το '=', για την ισότητα στις λογικές προτάσεις χρησιμοποιείται το '=='.

Μπορεί κάποιος επίσης, να συγκρίνει δύο πίνακες ίσων διαστάσεων ή τα στοιχεία ενός πίνακα με ένα αριθμό. Παράγεται τότε ένας πίνακας με λογικά στοιχεία:

- *στη σύγκριση πίνακα με πίνακα συγκρίνονται τα στοιχεία με την ίδια θέση, ενώ*
- *στη σύγκριση πίνακα με αριθμό συγκρίνεται κάθε στοιχείο του πίνακα με τον αριθμό αυτό.*

Λογικοί τελεστές

Οι **λογικοί τελεστές** (logical operators) φαίνονται στον πιο κάτω πίνακα μαζί με τις αντίστοιχες λογικές συναρτήσεις:

Συμβ.	Συνάρτηση	Περιγραφή
&	and	logical and λογικό και (κατά στοιχεία)
 	or	logical or λογικό ή (κατά στοιχεία)
~	not	logical not λογικό όχι

Η παράγραφος αυτή ασχολείται με αυτούς τους τρεις σημαντικότερους λογικούς τελεστές: **&**, **|** και **~**. Αν οι p και q είναι λογικές προτάσεις, ισχύουν τα εξής:

- Η $p \ \& \ q$ δίνει την τιμή 1 μόνο αν οι p και q είναι (και οι δύο) αληθείς.
- Η $p \ | \ q$ δίνει την τιμή 1 αν τουλάχιστο μια από τις p και q είναι αληθής.
- Η $\sim p$ δίνει την τιμή 1 μόνο αν η p είναι ψευδής και την τιμή 0 αν η p είναι αληθής.

Αντί των τελεστών **&**, **|** και **~** μπορεί κανείς να χρησιμοποιήσει και τις συναρτησιακές τους μορφές:

- Η **and(p,q)** είναι ισοδύναμη με την $p \ \& \ q$.
- Η **or(p,q)** είναι ισοδύναμη με την $p \ | \ q$.
- Η **not(p)** είναι ισοδύναμη με την $\sim p$.

Δομές ελέγχου ροής

Βρόχοι for

Οι βρόχοι for έχουν την εξής δομή:

```
for index = initial value (: step) : final value
    statements
end
```

Οι λέξεις “for” και “end” χρησιμοποιούνται στην αρχή και στο τέλος του βρόχου, ο μετρητής *index* παίρνει τις τιμές από *initial value* μέχρι *final value* με βήμα *step*, και οι εντολές (statements) εκτελούνται για όλες τις τιμές του μετρητή

index. Αν παραλειφθεί το βήμα, τότε η MATLAB χρησιμοποιεί το 1 σαν βήμα. Αξίζει να σημειωθεί επίσης ότι το βήμα μπορεί να είναι αρνητικό.

Παρατηρήσεις

Αν θέλει κανείς να γράψει ένα βρόχο σε μια γραμμή του παραθύρου εργασίας, χρησιμοποιεί οπωσδήποτε κόμμα ή ερωτηματικό μετά την εντολή `for` και μετά γράφει κατά τα γνωστά, τις επόμενες εντολές χωρίζοντάς τις επίσης με κόμματα ή ερωτηματικά. Στην τελευταία εντολή (`end`) δεν χρειάζονται ερωτηματικά.

Η εντολή *if*

Η εντολή `if` επιτρέπει τον έλεγχο εάν και εφόσον μια (ή περισσότερες) συνθήκες ισχύουν ώστε να εκτελεστεί σε κάθε περίπτωση η επιθυμητή ακολουθία εντολών και

πράξεων.

Η εντολή έχει την γενική μορφή:

```
if relation_1
    statement(s)
elseif relation_2
    statement(s)
else
    statement(s)
end
```

Οι συνθήκες ελέγχονται με τη χρήση σχεσιακών και λογικών τελεστών. Αξιοσημείωτο είναι επίσης το γεγονός ότι η εντολή `elseif` γράφεται σαν μια λέξη (δεν πρέπει να υπάρχει κενό μεταξύ του `else` και του `if`). Η απλούστερη μορφή της εντολής `if` είναι η πιο κάτω:

```
if relation
    statement(s)
end
```

Οι εντολές εκτελούνται μόνο αν ικανοποιείται η συνθήκη relation. Διαφορετικά δεν εκτελείται καμιά εντολή στο σημείο αυτό του προγράμματος και η ροή του τελευταίου συνεχίζει κανονικά. Αν θέλει κάποιος να γράψει στην ίδια γραμμή άλλες εντολές χρησιμοποιεί κόμμα ή ερωτηματικό για να χωρίσει την εντολή if από την επόμενη.

Παρακάτω παρουσιάζεται η περίπτωση κατά την οποία η “απόφαση” που χρειάζεται να παρθεί συμπεριλαμβάνει εναλλακτική περίπτωση, δηλ. ή το ένα θα συμβεί, ή το άλλο.

Η δομή είναι:

```
if relation 1
    statement(s)
else
    statement(s)
end.
```

Η εντολή switch

Η εντολή **switch-case** δίνει τη δυνατότητα επιλογής εκτέλεσης μιας ομάδα εντολών από άλλες πιθανές ομάδες.

Η γενική της δομή έχει ως εξής:

```
switch switch_expression
    case value_1
        statement(s)
    case value_2
        statement(s)
    case value_3
        statement(s)

    otherwise
        statement(s)
end
```

Σημειώνονται τα εξής γι’ αυτή την εντολή:

- Η πρώτη γραμμή περιέχει την λέξη κλειδί **switch**, ακολουθούμενη από το όνομα `switch_expression`, που θα δώσει κάποιος, το οποίο μπορεί να είναι βαθμωτή ποσότητα, αλφαριθμητικό, ή ακόμα και μαθηματική παράσταση με προκαθορισμένες μεταβλητές που μπορεί να πάρει μια τιμή.

- Μετά από το **switch**, ακολουθούν οι διάφορες εντολές **case**. Η κάθε μια έχει ένα όνομα (π.χ. `value_1`, `value_2` κλπ) το οποίο μπορεί να είναι βαθμωτή ποσότητα ή αλφαριθμητικό, και μετά ακολουθούν οι εντολές που θα εκτελεστούν αν βρεθούμε στη συγκεκριμένη περίπτωση.

- Μετά την τελευταία περίπτωση/εντολή **case**, ακολουθεί η προαιρετική περίπτωση/εντολή **otherwise** της οποίας οι εντολές θα εκτελεστούν αν καμιά από τις προηγούμενες περιπτώσεις δεν ισχύει.

- Σε αντίθεση με άλλες γλώσσες προγραμματισμού (όπως, π.χ. η C), στη MATLAB δεν χρειάζεται να διακοπεί η ροή της δομής μετά από κάθε **case**, μια και αυτό θα γίνει αυτόματα αφού μια από τις περιπτώσεις έχει επαληθευτεί.

Άλλες χρήσιμες εντολές

Οι εντολές *break*, *return* και *continue*

Οι βρόχοι `for` και `while` μπορούν να διακοπούν με την εντολή **break** η οποία μεταφέρει τον έλεγχο στην πρώτη εντολή μετά το τελικό `end` του βρόχου στον οποίο χρησιμοποιείται. Έτσι αν έχει κάποιος εγκλωβισμένους βρόχους ο έλεγχος περνά στον επόμενο (εξωτερικό βρόχο). Η εντολή `break` ορίζεται μόνο μέσα σε βρόχους `for`. Μια παρόμοια εντολή που μπορεί να χρησιμοποιηθεί εναλλακτικά είναι η εντολή **return**.

Γενικά η εντολή αυτή μεταφέρει τον έλεγχο στην καλούσα συνάρτηση (π.χ. στο καλόν `m-file`) ή στο πληκτρολόγιο. Χρησιμοποιείται επίσης για τερματισμό της εντολής **keyboard** που μεταφέρει τον έλεγχο από ένα `m-file` στο πληκτρολόγιο. Όπως μπορεί κανείς να παρατηρήσει στη συνέχεια με την εντολή `return`, μπορεί να διακοπεί όταν χρειάζεται η εκτέλεση των εντολών ενός `m-file`. Τέλος η εντολή **continue** μεταβιβάζει τον έλεγχο στην επόμενη επανάληψη ενός βρόχου `for` ή `while` χωρίς να εκτελεστούν οι εναπομένουσες εντολές του βρόχου.

Η συνάρτηση beep

Η συνάρτηση **beep** κάνει ένα προειδοποιητικό beep και χρησιμοποιείται συνήθως μαζί με μηνύματα σφάλματος.

Η λογική συνάρτηση find

Η πιο σημαντική λογική συνάρτηση είναι η **find**. Η συνάρτηση αυτή ερευνά ένα διάνυσμα ή ένα πίνακα και βρίσκει ποια στοιχεία του ικανοποιούν ένα δοσμένο κριτήριο.

Εισαγωγή και Εκτύπωση Δεδομένων

Εισαγωγή δεδομένων από τον χρήστη

Είναι ήδη γνωστό, ότι ένας απλός τρόπος για να εισαγάγει κάποιος την τιμή μιας νέας μεταβλητής είτε στο παράθυρο εντολών είτε μέσω κάποιου m-file είναι με την εντολή **input**.

Ακολουθεί παράδειγμα:

```
answer=input('ΓΡΑΨΤΕ yes ΓΙΑ ΝΑΙ ΚΑΙ no ΓΙΑ ΟΧΙ: \n', 's');  
ΓΡΑΨΤΕ yes ΓΙΑ ΝΑΙ ΚΑΙ no ΓΙΑ ΟΧΙ:  
Answer =  
yes
```

Εκτύπωση δεδομένων στην οθόνη

Ο πιο απλός τρόπος για να τυπωθούν τα αποτελέσματα της MATLAB στο παράθυρο εντολών είναι η πληκτρολόγηση του ονόματος της μεταβλητής ή ακόμα να μη χρήση ερωτηματικού στο τέλος της εντολής που την υπολογίζει. Η MATLAB γράφει τότε το όνομα, ένα '=' και μετά την τιμή της μεταβλητής. Η συνάρτηση **disp** παραλείπει το όνομα της μεταβλητής και το '=' και εκτυπώνει την τιμή της συνάρτησης σύμφωνα με την τρέχουσα format:

```
>> format long  
>> disp(1/eps), disp(pi)  
4.503599627370496e+015  
3.141592653589793
```

Ένα πλεονέκτημα της `disp` είναι το ότι επιτρέπει την εκτύπωση χρήσιμων ενημερωτικών αλφαριθμητικών. Για παράδειγμα

```
>> disp('Πείραμα 1'),  
Πείραμα 1
```

Κάτι που δεν μπορεί κάποιος να κάνει με τη `disp` είναι να τυπώσει διάφορες μεταβλητές σε μια γραμμή.

Παράθυρο διαλόγου

Η εντολή `msgbox(messege)` δημιουργεί ένα παράθυρο διαλόγου που εμφανίζεται αυτόματα.

Το μέγεθος του κειμένου προσαρμόζεται αυτόματα και θα έλεγε κανείς ότι καθορίζει και το μέγεθος του παραθύρου.

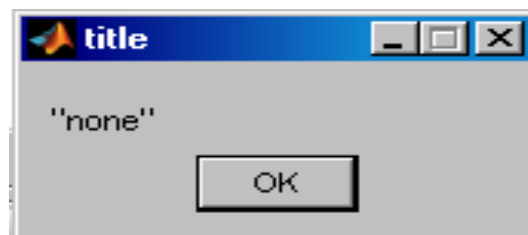
Το μήνυμα μπορεί να είναι ένα αλφαριθμητικό διάνυσμα ή ένας αλφαριθμητικός πίνακας.

Με την χρήση αυτής της αλλάζοντας τις παραμέτρους μπορεί κανείς να δηλώσει και τον τίτλο που θα ήθελε να εμφανίζει το παράθυρο.

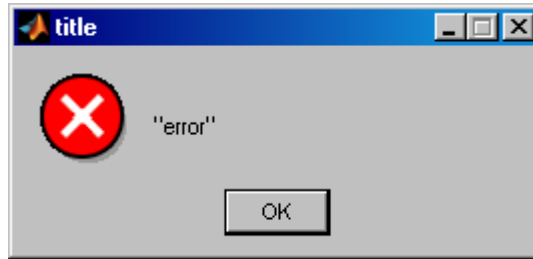
Msgbox(message, title)

Επίσης μπορεί κάποιος να εμφανίζει στο παράθυρο διαλόγου και κάποια εικονίδια έτσι ώστε να τραβάει πιο εύκολα την προσοχή του χρηστή. Οι παράμετροι που προσθέτονται στην εντολή για να εμφανίζει εικονίδια είναι:

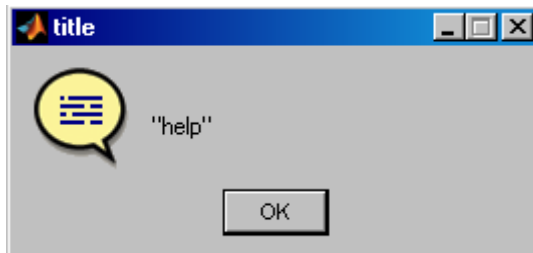
- **'none'** αν δεν θέλει κάποιος να εμφανίσει τίποτα ,



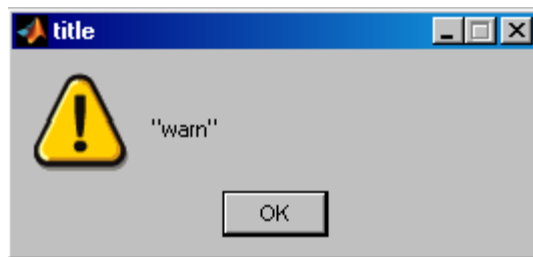
- **'error'** για να εμφανίσει ένα εικονίδιο λάθους,



- 'help' για να εμφανίσει ένα εικονίδιο με θαυμαστικό,



- 'warn' για εμφάνιση μηνύματος λάθους.



Msgbox('message', 'title', 'icon')

Σαν προεπιλογή είναι η παράμετρος 'none'.

Χαρακτήρες μετατροπής και απόδρασης

Χαρακτήρες μετατροπής (conversion characters)

Οι χαρακτήρες μετατροπής καθορίζουν το συμβολισμό της εκτύπωσης όπως φαίνεται στον πίνακα που ακολουθεί. Το κεφάλαιο αυτό επικεντρώνεται κυρίως στους πρώτους έξι χαρακτήρες.

Χαρακτήρας	Περιγραφή
%d	(Προσημασμένος) δεκαδικός συμβολισμός
%e	Εκθετικός συμβολισμός (με μικρό e, όπως στον 3.1415e+00)
%f	Συμβολισμός σταθερής υποδιαστολής
%g	Η πιο συμπαγής μορφή από τις %e και %f με παραλειψη αχρειαστων μηδενικών.
%c	Απλός χαρακτήρας
%s	Αλφαριθμητικό (ακολουθία χαρακτήρων)
%E	Εκθετικός συμβολισμός (με κεφαλαίο E, όπως στον 3.1415E+00)
%G	Όμοιο με την %g αλλά με τη χρήση κεφαλαίου E.
%i	(Προσημασμένος) δεκαδικός συμβολισμός
%o	(Μη προσημασμένος) οκταδικός συμβολισμός
%u	(Μη προσημασμένος) δεκαδικός συμβολισμός
%x	Δεκαεξαδικός συμβολισμός (με μικρά γράμματα x-f)
%X	Δεκαεξαδικός συμβολισμός (με κεφαλαία γράμματα A-F)

Χαρακτήρες απόδρασης (escape characters)

Οι χαρακτήρες απόδρασης καθορίζουν μη εκτυπωτικούς χαρακτήρες (nonprinting characters) σε μια προδιαγραφή μορφής:

Χαρακτήρας	Περιγραφή
\b	πίσω διάστημα (backspace)
\f	νέα τροφοδοσία (form feed)
\n	Νέα γραμμή
\r	Carriage return, παρόμοιο με τη νέα γραμμή
\t	Οριζόντια στοίχιση (Horizontal tab)
\\	\
' ' ή "	' (μονό εισαγωγικό)
%\%	(το σύμβολο) %

Ο χαρακτήρας \t είναι χρήσιμος στη δημιουργία πινάκων αφού εξασφαλίζει τη στοίχιση των μεταβλητών εξόδου.

Εντολές επεξεργασίας εικόνας

Η εντολή image

Η εντολή αυτή εμφανίζει το διάνυσμα που θα της δώσει κάποιος σαν εικόνα. Το κάθε στοιχείο του διανύσματος προσδιορίζει κάθε pixel της εικόνας. Το διάνυσμα μπορεί να είναι ένας πίνακας διαστάσεων MxN ή MxNx3 και μπορεί να έχει μεταβλητές τύπου double, uint8, uint16. Όταν το διάνυσμα είναι διδιάστατος πίνακας MxN, τα στοιχεία του χρησιμοποιούνται ως δείκτες του τρέχον πίνακα «Colormap» για να καθορίζει το χρώμα της εικόνας.

Η σύνταξη της εντολής είναι : **IMAGE(C)**

Για παράδειγμα αν ο πίνακας c είναι :

```
>> c=[2, 4, 6 ; 3, 6, 9 ; 1, 5, 0]
```

c =

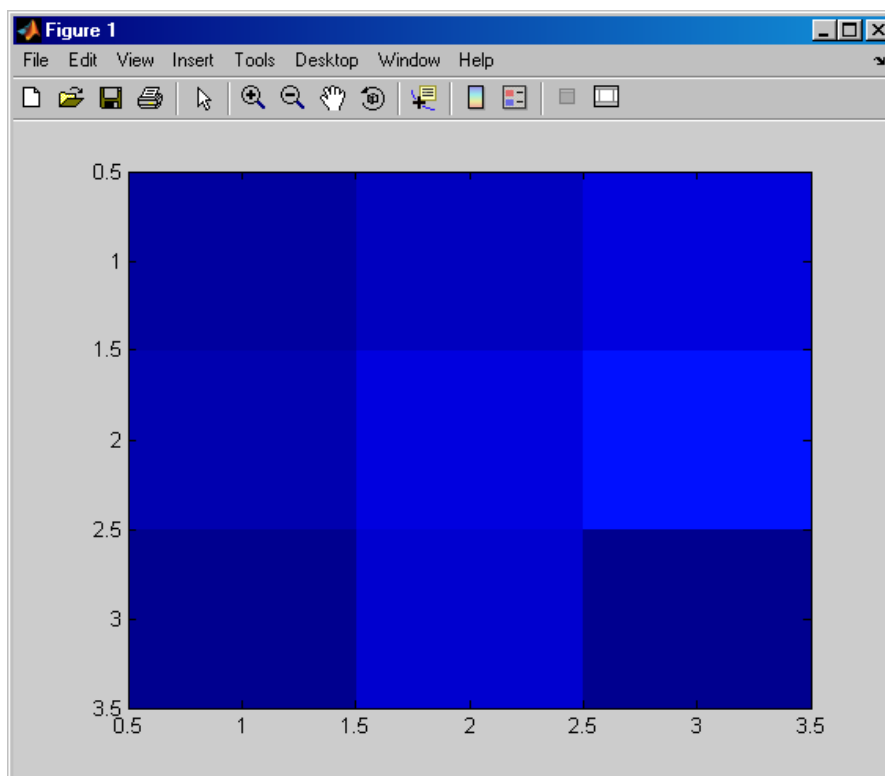
```
2 4 6
```

```
3 6 9
```

```
1 5 0
```

```
>> image(c)
```

Το αποτέλεσμα μετά την εκτέλεση της εντολής θα είναι :



Αλλάζοντας τα στοιχεία του πίνακα c παρατηρώ ότι αλλάζει και η απόχρωση των χρωμάτων:

```
>>c=[20, 45, 70 ; 35, 60 , 55; 10, 5, 25]
```

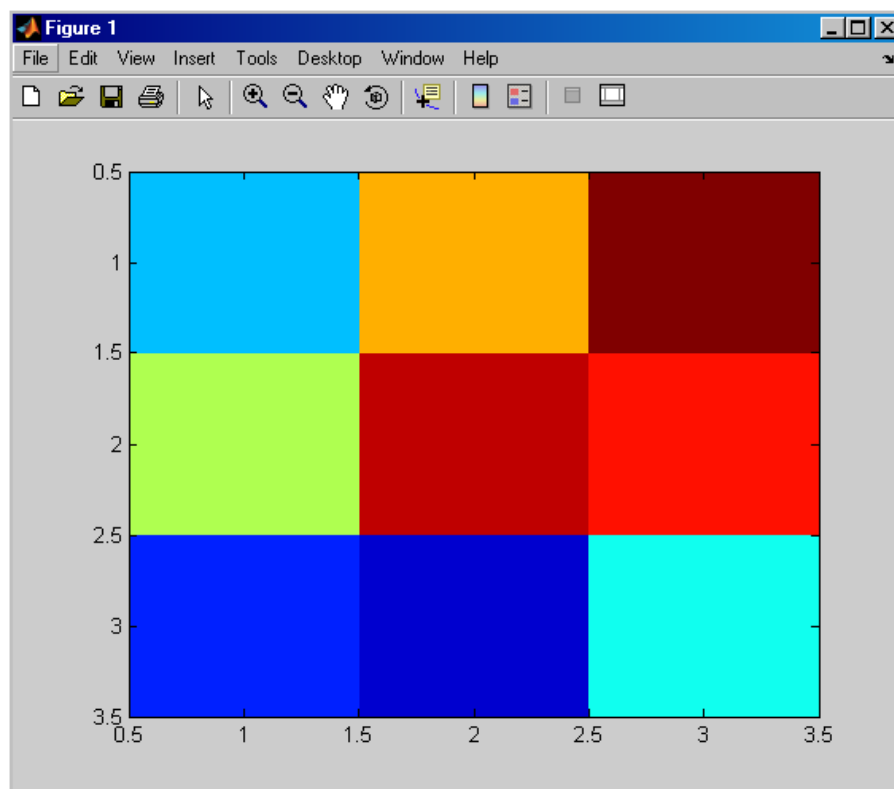
c =

```
20 45 70
```

```
35 60 55
```

```
10 5 25
```

Η εικόνα image(c) φαίνεται παρακάτω:



Η εντολή imread

Η εντολή `imread` βοηθάει στην 'μελέτη' μια εικόνας - κάποιας ορισμένης συμπίεσης και να την μεταφορά της στο Matlab σε μορφή πίνακα έτσι ώστε να μπορεί να τεθεί υπό επεξεργασία . Η εικόνα μπορεί να είναι είτε ασπρόμαυρη είτε έγχρωμη και προσδιορίζεται από το όνομα της ως `filename` .

Η εντολή `imread` συντάσσεται ως εξής :

>>A=imread(filename,fmt);

Το αρχείο της εικόνας που θέλει κάθε φορά κάποιος να διαβάσει, θα πρέπει να βρίσκεται στον τρέχον φάκελο που επεξεργάζεται αρχεία ή σε κάποιο φάκελο του μονοπατιού του Matlab για να μπορεί μετά από κάθε εκτέλεση εντολής να βρεθεί το αρχείο και να επεξεργαστεί. Με την λέξη *fmt* δηλώνεται η κατάληξη του αρχείου που θέλει κάποιος να διαβάσει.

Το αποτέλεσμα που θα λάβει κάποιος εκτελώντας αυτήν την εντολή αποθηκεύεται στην μεταβλητή *A*, η οποία είναι ένας πίνακας ο οποίος περιέχει τις πληροφορίες της φωτογραφίας.

Αν το αρχείο περιέχει μια ασπρόμαυρη φωτογραφία, τότε το *A* θα είναι ένας $M \times N$ πίνακας. Αν το αρχείο περιέχει μια φωτογραφία με χρώματα τότε το *A* θα είναι ένας πίνακας $M \times N \times 3$.

Εξαίρεση αποτελεί ο τύπος φωτογραφιών *tiff* όπου περιέχει άλλον τύπο χρωματικού κώδικα και το αποτέλεσμα της εντολής δηλαδή το *A* θα έχει σαν αποτέλεσμα έναν πίνακα $M \times N \times 4$

Η εντολή imshow

Η εντολή αυτή είναι από τις σημαντικότερες γιατί μας βοηθάει να εμφανίσουμε οποιαδήποτε εικόνα μας ενδιαφέρει.

Η σύνταξη της είναι η εξής :

A=IMSHOW(FILENAME)

Με αυτήν την εντολή εμφανίζεται η εικόνα που είναι αποθηκευμένη με το όνομα **FILENAME**. Το αρχείο πρέπει να περιέχει μια φωτογραφία η οποία μπορεί να διαβαστεί από αυτήν την εντολή. Η εντολή αυτή καλεί την εντολή *imread* για να διαβάσει την εικόνα από το αρχείο, όμως δεν αποθηκεύει τα δεδομένα στο περιβάλλον του MATLAB. Αν το αρχείο περιέχει πολλαπλές φωτογραφίες, τότε εμφανίζει μόνο την πρώτη. Το αρχείο που θα διαβάσει κάποιος θα πρέπει να βρίσκεται στο ίδιο μονοπάτι εργασίας όπου δουλεύει το MATLAB για να μπορεί να την επεξεργαστεί.

Η εντολή imfinfo

Με την εντολή αυτή μπορεί κάποιος να πληροφορηθεί για μια εικόνα .

Η σύνταξη της είναι:

INFO = IMFINFO(FILENAME,FMT)

Με το FILENAME καθορίζεται το όνομα της εικόνας, και με το FMT προσδιορίζεται ο τύπος της εικόνας. Το αρχείο για να το διαβάσει η εντολή πρέπει να είναι στον τρέχον φάκελο.

Για παράδειγμα :

```
>> info=imfinfo('images.jpg')
```

info =

```
Filename:      'images.jpg'  
FileModDate:   '09-Μαρ-2008 11:54:09'  
FileSize:      2868  
Format:        'jpg'  
FormatVersion: ''  
Width:         111  
Height:        111  
BitDepth:      24  
ColorType:     'truecolor'  
FormatSignature: ''  
NumberOfSamples: 3  
CodingMethod:  'Huffman'  
CodingProcess: 'Sequential'  
Comment:       {}
```

Εδώ παρουσιάζονται δύο παραδείγματα για την εντολή IMFINFO. Στο πρώτο παράδειγμα χρησιμοποιείται μια έγχρωμη φωτογραφία ενώ για το επόμενο μια

ασπρόμαυρη. Επομένως μπορεί κανείς να συμπεράνει τα χαρακτηριστικά της κάθε φωτογραφίας όπως τα παρουσιάζει η εντολή αυτή. Τα ονόματα της κάθε φωτογραφίας και ο τύπος της εμφανίζονται. Επίσης πληροφορίες παρέχονται για τις διαστάσεις κάθε εικόνας καθώς και σε ποιόν χρωματική κλίμακα ανήκουν. Τέλος η ημερομηνία δημιουργίας του κάθε αρχείου είναι επίσης προσβάσιμη.

```
>> INFO = IMFINFO(' paradeigma2a_blk.jpg')
```

```
INFO =
```

```
Filename:      'paradeigma2a_blk.jpg'  
FileModDate:   '25-Φεβ-2008 17:03:32'  
FileSize:      601463  
Format:        'jpg'  
FormatVersion: ''  
Width:         2592  
Height:        1944  
BitDepth:      8  
ColorType:     'grayscale'  
FormatSignature: ''  
NumberOfSamples: 1  
CodingMethod:  'Huffman'  
CodingProcess: 'Sequential'  
Comment:       {}
```

ΚΕΦΑΛΑΙΟ 4

Αλγόριθμοι Αναγνώρισης Αντικειμένων

Η προσπάθεια για υλοποίηση ενός προγράμματος που θα μπορούσε να ανταποκρίνεται στις ανάγκες της μελέτης της συγκεκριμένης εργασίας, είχε ως επακόλουθο τον πειραματισμό με πολλές εντολές και την υλοποίηση πολλών κομματιών κώδικα. Αυτό το κεφάλαιο παρουσιάζει κάποια από τα πειράματα που υλοποιήθηκαν μαζί με κάποιο κομμάτι του κώδικα τους που χρειάστηκε στην ενέργεια αυτή.

Το ιδανικό πρόγραμμα θα ήταν να μπορεί να γίνει αυτόματη ενημέρωση των νέων εικόνων που αποτυπώνονται από το περιβάλλον παρακολούθησης έτσι ώστε να μπορούν να επεξεργάζονται πιο άμεσα και να λαμβάνει ο ενδιαφερόμενος το αποτέλεσμα. εφόσον υπάρξει κάτι στο χώρο παρακολούθησης χωρίς κάποιου είδους θόρυβο αλλά να παίρναμε απ' ευθείας σε μια νέα εικόνα μετά το προειδοποιητικό μήνυμα και ήχο, το αντικείμενο (σκάφους) που καταφέραμε να εντοπίσει το σύστημα μας.

Η επίτευξη και υλοποίηση του παραπάνω στόχου, προϋποθέτει την επίτευξη μικρών βημάτων κάθε στιγμή για την ολοκλήρωση του γενικού συνόλου. Με λίγα λόγια, η υλοποίηση του κώδικα είναι το πρώτο και σημαντικότερο βήμα της εργασίας αυτής. Τα απαραίτητα εργαλεία για την επίτευξη αυτού του βήματος, είναι δυο ζεύγη φωτογραφιών, τις οποίες ο κώδικας θα επεξεργαστεί ώστε να εντοπίσει και να καταγράψει την παρουσία του πλοίου σε αυτές.

Με αυτήν την λογική λοιπόν, έχοντας ανά ζευγάρια τις εικόνες, η σύγκριση γίνεται πιο εύκολη. Στην μια εικόνα θα υπάρχει η παρουσία του αντικειμένου-πλοίου, ενώ στην άλλη όχι.

Είσοδος αντικειμένου σε στατικό περιβάλλον έρευνας

Για την υλοποίηση των παρακάτω πειραμάτων χρησιμοποιήσαμε μια στατική εικόνα θεωρώντας ότι το περιβάλλον παραμένει σταθερό, και την συγκρίναμε με κάποιες άλλες που ενδεχομένως υπάρχει αντικείμενο σε αυτές.

ΠΕΙΡΑΜΑ 1:

Για την υλοποίηση του πρώτου αυτού πειράματος χρησιμοποιήθηκαν δύο εικόνες σε ασπρόμαυρη κλίμακα, από τις οποίες στην πρώτη `deigma1a_blk` δεν υπάρχει κάποιος σκάφος, ενώ στην δεύτερη `deigma1b_blk` υπάρχει. Η πρώτη δοκιμασία αφορούσε εικόνες μορφής `.jpg`.

Η βασική ιδέα για αυτού του είδους τα πειράματα είναι η μετατροπή των στοιχείων των εικόνων σε στοιχεία πίνακα και η κατά στοιχείο σύγκριση τους για την ανεύρεση των διαφορών τους.

Παρακάτω είναι αναρτημένο το κομμάτι του κώδικα που δημιουργήθηκε και εκτελέστηκε για τον εντοπισμό σκάφους. Το πείραμα σε αυτήν την περίπτωση ονομάστηκε `paradeigma` και για να εκτελεστεί από το Matlab χρειάζεται μόνο να πληκτρολογήσουμε το όνομα του αρχείου στο `command` του Matlab.

```
function k=paradeigma

clear
clc
[x1,pa11]=imread('deigma1a_blk.jpg');
[x2,pa12]=imread('deigma1b_blk.jpg');
c=abs(double(x2)-double(x1));
[x,y]=find(c~=0);
new=255*ones(max(x)-min(x)+1,max(y)-min(y)+1);
xnew=x-min(x)+1;
ynew=y-min(y)+1;
for i=1:length(xnew)
    new(xnew(i),ynew(i))=x2(x(i),y(i))+1 ;
end
figure(1)
imshow(x1,pa11)
figure(2)
imshow(x2,pa12)

figure(3)
k=imshow(new,pa11);
y=pa12;
```

Μετά την εκτέλεση του προγράμματος, θεωρήθηκε σωστό να εμφανίζονται οι παρακάτω εικόνες για να μπορεί να παρατηρήσει κάποιος την λειτουργία του κώδικα μας. Παρατηρούμε λοιπόν, τις δύο εικόνες που συγκρίθηκαν και το αποτέλεσμα που πάρθηκε

Figure 1:

Εικόνα χωρίς αντικείμενο



Figure 2:

Εικόνα με αντικείμενο



Figure 3:



ΠΕΙΡΑΜΑ 2:

Ο κώδικας για το δεύτερο μας πείραμα είναι ο παρακάτω και το ζευγάρι των εικόνων που θα χρησιμοποιήσουμε είναι με κωδικοποίηση bmp :

```
function k=paradeigma2

clear
clc
[x1,pal1]=imread('deigma2a_blk.bmp');
[x2,pal2]=imread('deigma2b_blk.bmp');
c=abs(double(x2)-double(x1));
[x,y]=find(c~=0);
new=255*ones(max(x)-min(x)+1,max(y)-min(y)+1);
xnew=x-min(x)+1;
ynew=y-min(y)+1;
for i=1:length(xnew)
    new(xnew(i),ynew(i))=x2(x(i),y(i))+1 ;
end
figure(1)

imshow(x1,pal1)
title('Εικόνα χωρίς αντικείμενο')
figure(2)
imshow(x2,pal2)
title('Εικόνα με αντικείμενο')

figure(3)
k=imshow(new,pal1);
title('Αντικείμενο')
```

Για να τρέξουμε το πρόγραμμα αυτό πάμε στο command του Matlab γράφοντας την λέξη `paradeigma2`, όπου είναι η λέξη που έχουμε ονομάσει των κώδικα μας στο M-file, και παίρνουμε τα παρακάτω αποτελέσματα:

Figure 1:

Εικόνα χωρίς αντικείμενο



Figure 2:

Εικόνα με αντικείμενο



Figure 3:

Αντικείμενο



ΠΕΙΡΑΜΑ 3:

Ο κώδικας για το τρίτο μας πείραμα είναι ο παρακάτω και το ζευγάρι των εικόνων που θα χρησιμοποιήσουμε είναι με κωδικοποίηση gif :

```
function k=paradeigma3

clear
clc
[x1,pal1]=imread('deigma3a_blk.gif');
[x2,pal2]=imread('deigma3b_blk.gif');
c=abs(double(x2)-double(x1));
[x,y]=find(c~=0);
new=255*ones(max(x)-min(x)+1,max(y)-min(y)+1);
xnew=x-min(x)+1;
ynew=y-min(y)+1;
for i=1:length(xnew)
    new(xnew(i),ynew(i))=x2(x(i),y(i))+1 ;
end

figure(1)

imshow(x1,pal1)
title('Εικόνα χωρίς αντικείμενο')
figure(2)
imshow(x2,pal2)
title('Εικόνα με αντικείμενο')

figure(3)
k=imshow(new,pal1);
title('Αντικείμενο')
```

Για να τρέξουμε το πρόγραμμα αυτό πάμε στο command του Matlab γράφοντας την λέξη `paradeigma3`, όπου είναι η λέξη του έχουμε ονομάσει των κώδικα μας στο M-file, και παίρνουμε τα παρακάτω αποτελέσματα:

Figure 1:

Εικόνα χωρίς αντικείμενο



Figure 2:

Εικόνα με αντικείμενο



Figure 3:

Αντικείμενο



ΠΕΙΡΑΜΑ 4:

Σε αυτό το πείραμα δοκιμάσαμε κάτι καινούριο. Προσπαθήσαμε με τον νέο τροποποιημένο κώδικα που φτιάξαμε να κάνουμε αρχικά έναν έλεγχο αν υπάρχει αντικείμενο ή όχι στην εικόνα και μετά να μας δίνει τα αποτελέσματα της επεξεργασίας. Για να το κάνουμε αυτό, χρησιμοποιήσαμε την εντολή `msgbox` η οποία εμφανίζει το μήνυμα αν υπάρχει ή όχι αντικείμενο σε νέο παραθυράκι.

Εκδοχή με αντικείμενο:

```
function [k,pal1]=paradeigma

clear
clc
[x1,pal1]=imread('deigma1a_blk.jpg');
[x2,pal2]=imread('deigma1b_blk.jpg');

c=abs(double(x2)-double(x1));
if (c==0)
    beep
    msgbox('ΔΕΝ ΥΠΑΡΧΕΙ ΑΝΤΙΚΕΙΜΕΝΟ ΣΤΙΣ ΦΩΤΟΓΡΑΦΙΕΣ','ΣΥΝΓΚΡΙΝΣΗ
ΦΩΤΟΓΡΑΦΙΩΝ')

else
    beep
    msgbox('ΥΠΑΡΧΕΙ ΑΝΤΙΚΕΙΜΕΝΟ ΣΤΙΣ ΦΩΤΟΓΡΑΦΙΕΣ','ΣΥΝΓΚΡΙΝΣΗ
ΦΩΤΟΓΡΑΦΙΩΝ')
```

```

[x,y]=find(c~=0);
new=255*ones(max(x)-min(x)+1,max(y)-min(y)+1);
xnew=x-min(x)+1;
ynew=y-min(y)+1;

for i=1:length(xnew)
    new(xnew(i),ynew(i))=x2(x(i),y(i))+1 ;

end

disp('ΘΕΛΕΤΕ ΝΑ ΕΜΦΑΝΙΣΤΕΙ Η αρχική ΕΙΚΟΝΑ ΜΕ ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ?')
answe=input('ΓΡΑΨΤΕ yes ΓΙΑ ΝΑΙ ΚΑΙ no ΓΙΑ ΟΧΙ: \n', 's')
switch answe
    case 'yes'
        figure(2)
        imshow(x2,pal2)
        title('Εικόνα με αντικείμενο')
        figure (3)
        imshow(new,pal1);
        title('Αντικείμενο')
    case 'no'
        disp('το πειραμα τελειωσε το αντικειμενο που εμφανιστικε
ειναι το ακολουθο')
        figure (3)
        imshow(new,pal1);
        title('Αντικείμενο')
    otherwise

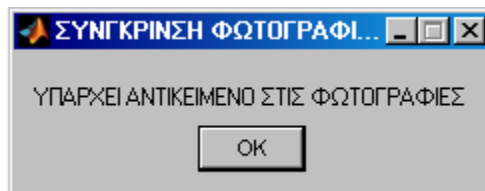
        disp('ΔΩΣΑΤΕ ΛΑΘΟΣ ΕΠΙΛΟΓΗ')

end

end
end

```

Για το παραπάνω πείραμα με τον έλεγχο που γίνεται φαίνεται ότι υπάρχει αντικείμενο, οπότε παίρνουμε το παρακάτω μήνυμα.



Αμέσως μετά στο command του matlab εμφανίζετε το μήνυμα
ΘΕΛΕΤΕ ΝΑ ΕΜΦΑΝΙΣΤΕΙ Η αρχική ΕΙΚΟΝΑ ΜΕ ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ?
ΓΡΑΨΤΕ yes ΓΙΑ ΝΑΙ ΚΑΙ no ΓΙΑ ΟΧΙ:

Για την περίπτωση που ο χρήστης πληκτρολογήσει `yes` θα εμφανιστεί η αρχική εικόνα που υπάρχει το αντικείμενο

Εικόνα με αντικείμενο



Και μαζί με αυτήν εμφανίζετε και η εικόνα με το αντικείμενο μόνο



Αν η επιλογή του χρήστη είναι *no*

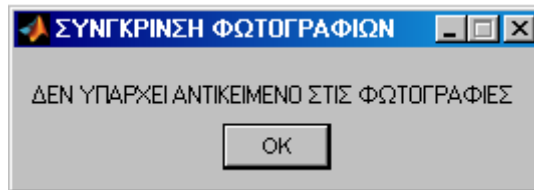
Το αποτέλεσμα είναι μόνο η παραπάνω εικόνα με το αντικείμενο, μετά τον έλεγχο.

Στο πείραμα αυτό βάλουμε και την εκδοχή ο χρήστης να μην πληκτρολογήσει κάτι ή να κάνει κάποιο λάθος, τότε το πείραμα τερματίζετε και βγάζει στον χρήστη το μήνυμα

«ΔΩΣΑΤΕ ΛΑΘΟΣ ΕΠΙΛΟΓΗ».

Εκδοχή χωρίς αντικείμενο:

Αν στον παραπάνω κώδικα αλλάξουμε το ζευγάρι των φωτογραφιών και βάλουμε δύο φωτογραφίες χωρίς κάποιο αντικείμενο, τότε τρέχοντας τον κώδικα αντί να μας βγάλει το προηγούμενο αποτέλεσμα, θα πάρουμε ένα μήνυμα της μορφής



Και σε αυτό το σημείο τερματίζεται ο κώδικας.

ΠΕΙΡΑΜΑ 5:

Παρακάτω παρατηρούμε τα αποτελέσματα από ένα ακόμα πείραμα το οποίο περιείχε αντικείμενο.

Η αρχική εικόνα που πείραμα είναι η παρακάτω, αλλάζοντας κάποιες μικρές παραμέτρους στο πείραμα μας πείραμα το παρακάτω αποτέλεσμα

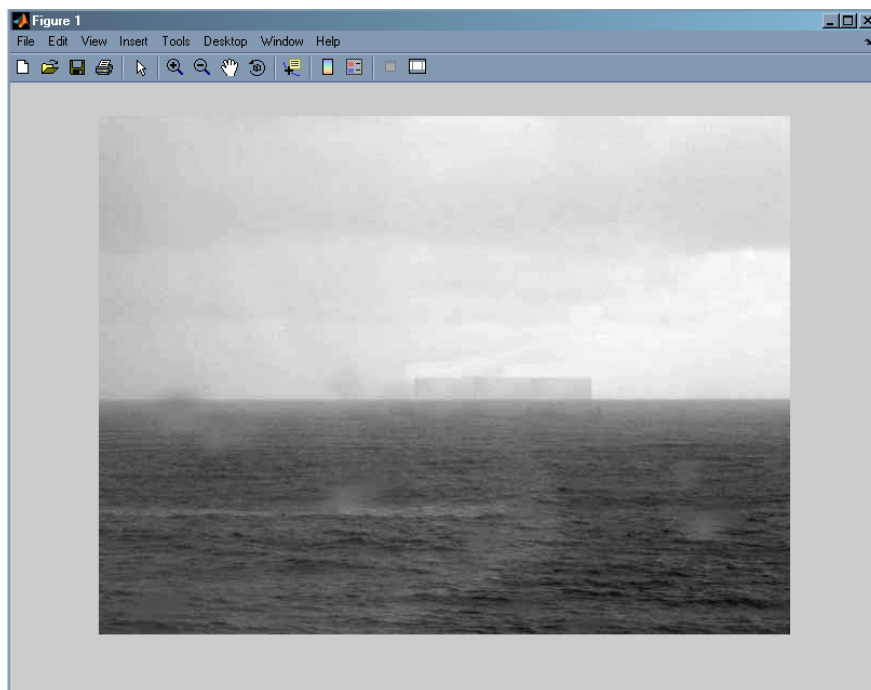


Figure 1: Αρχική εικόνα χωρίς αντικείμενο



Figure 2: Εικόνα με αντικείμενο

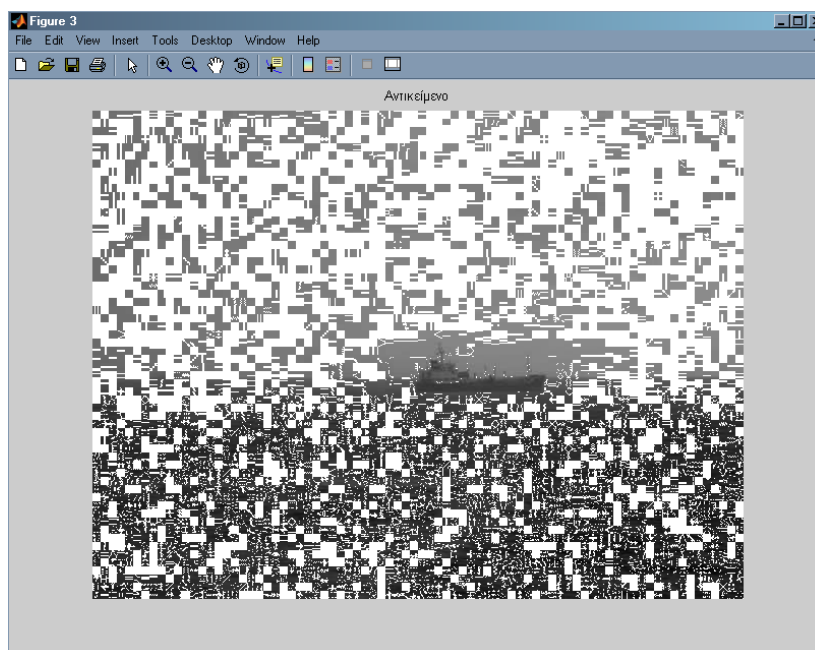


Figure 3 : Εικόνα με αντικείμενο

ΠΕΙΡΑΜΑ 6 :

Για τελευταίο πείραμα δοκιμάσαμε ένα παράδειγμα με μια μικρή διαφοροποίηση, δοκιμάσαμε να επεξεργαστούμε δύο εικόνες οι οποίες και οι δύο είχαν κάποιο αντικείμενο. Είναι δύο εικόνες όπου τραβήχτηκαν με πολύ μικρή

χρονική διαφορά, όταν είχε ήδη μπει ένα αντικείμενο στο χώρο παρακολούθησης και απλά εμείς το πήραμε σε διαφορετική φάση.

Το κομμάτι του κώδικα δεν αλλάζει, μόνο κάποιες πολύ μικρές διαφοροποιήσεις κάναμε, όσο αφορά τα ονόματα των εικόνων.



Αυτήν την εικόνα θα μπορούσαμε να την χαρακτηρίσουμε ως αρχική εικόνα, αμέσως μετά θα παρουσιαστούν η ακόλουθη εικόνα που θα επεξεργαστούμε και την τελική εικόνα που θα πάρουμε μετά την επεξεργασία.



Figure 2 : Εικόνα με αντικείμενο

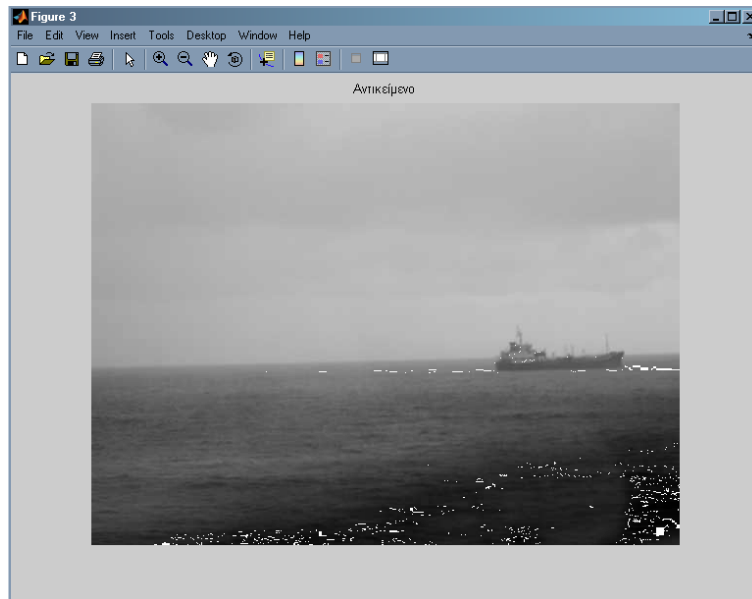


Figure 3 : Εικόνα με την διαφορά

Παρατηρούμε εδώ ότι στην τελική εικόνα που εμφανίζεται φαίνονται και κάποια κομμάτια ασπρο χρώμα, αυτό θα μπορούσαμε να το χαρακτηρίσουμε σαν θόρυβο ίσως από τον κυματισμό ή τον φωτισμό του περιβάλλοντος.

Τέλος στα παραπάνω πειράματα παρατηρούμε ότι η εικόνα που μας εμφανίζει το αντικείμενο εκτός του αντικείμενο φαίνετε και ένα μεγάλο μέρος της αρχικής εικόνας. Αυτό γίνεται γιατί αν θεωρήσουμε ότι έχουμε την φωτογραφική κάπου στερεωμενη και βγάζει επαναλαβόμενες εικόνες χωρίς να μετατοπιστεί καθόλου, ακόμα και όταν δεν υπάρχει αντικείμενο μέσα παρατηρούνται κάποιες πολύ μικρές αλλαγές στο περιβάλλον (σύννεφα, φως κλπ).

Είσοδος αντικειμένου σε μεταβλητό περιβάλλον έρευνας

Παραπάνω αναλύουμε τη μέθοδο με την οποία μπορούμε να αναγνωρίσουμε ένα αντικείμενο σε μια εικόνα όπου ο χώρος παρακολούθησης παραμένει σταθερός. Στο επόμενο κομμάτι θα δούμε μια εκδοχή με την οποία γίνεται αναγνώριση αντικειμένου στο χώρο παρακολούθησης στον οποίο υπάρχει κάποια μεταβολή, όπως για παράδειγμα κύμα, σύννεφα κ.α. παραλείποντας τυχόν ίχνη θορύβου.

Με το πείραμα μας μετατρέπουμε μια εικόνα σε πίνακα στοιχείων – pixels, όπου το περιεχόμενο τους στο σύνολο, αντιπροσωπεύει την κλίμακα χρωμάτων κάθε εικόνας. Υποθέτοντας ότι το αντικείμενο μας χαρακτηρίζεται από συγκεκριμένα χρώματα της κλίμακας του γκρι, ελέγχουμε την κάθε γραμμή του πίνακα και αν για συνεχόμενα “cell” pixels υπάρχουν αυτά τα χρώματα, εμφανίζεται ένα μήνυμα ότι βρέθηκε αντικείμενο.



Παρακάτω παρατηρούμε το κώδικα :

```
function [F,H,V,k]=boat (Y,X)
```

```
F=0;
```

```
H=0;
```

```
V=0;
```

```
cell=50;
```

```
c=imread('boat.jpg');
```

```
[x,y]=find(c~=0);
```

%Y, X εξωτερικές μεταβλητές διαστάσεις πίνακα

```
for i=1:length(X)
    k=0;
    for j=1:length(Y)

        if (c(i,j)<120 || c(i,j)>90)
            k=k+1;
        else
            k=0;
        end

        if k==cell
            F=1;
            H=i;
            V=j;
        end
    end
end
if F==1
    beep
    msgbox('ΒΡΕΘΗΚΕ ΑΝΤΙΚΕΙΜΕΝΟ', 'FOUND', 'help' )
end
end
```

ΚΕΦΑΛΑΙΟ 5

Προτάσεις Βελτίωσης

Στα προηγούμενα κεφάλαια παραθέσαμε δυο διαφορετικούς τρόπους υλοποίησης του πειράματός μας. Καταρχάς δοκιμάσαμε έναν τρόπο για την εύρεση αντικειμένου παίρνοντας ένα ζευγάρι εικόνων και αφού πρώτα αντιστοιχούσαμε κάθε σημείο της κάθε εικόνας σε στοιχείο πίνακα συγκρίναμε τις δύο εικόνες και σαν αποτέλεσμα πήραμε ένα μήνυμα αν υπάρχει αντικείμενο στην εικόνα ή όχι, καθώς θεωρήσαμε σημαντικό να δίνετε σε νέα εικόνα το αντικείμενο αυτό.

Δοκιμάσαμε ακόμα να μετατρέψουμε την εικόνα σε ένα πίνακα στοιχείων του οποίου ελέγξαμε τα στοιχεία του ανά γραμμή συγκρίνοντας τα με μία συγκεκριμένη περιοχή χρώματος, που θεωρήσαμε ότι είναι η χρωματική περιοχή που απεικονίζει το αντικείμενο.

Για να μπορέσουμε να θεωρήσουμε ότι η εφαρμογή που υλοποιήσαμε είναι αρκετά ικανοποιητική κατά την εφαρμογή της και μπορεί να ανταποκριθεί σε όλες τις περιπτώσεις, σε αυτό το κεφάλαιο θα δούμε τρόπους βελτίωσης.

Κατά την υλοποίηση δοκιμάσαμε παίρνοντας ζευγάρια, ζευγάρια τις φωτογραφίες και τις συγκρίναμε, βρίσκοντας αν υπάρχει κάποιο αντικείμενο ή όχι μέσα. Για να είναι ορθότερο το σύστημα μας, θα πρέπει να φτιαχτεί ένα αρχείο, το οποίο καταρχάς θα περιέχει εικόνες του χώρου παρακολούθησης με διάφορες πιθανές αλλαγές που μπορεί να παρουσιαστούν στο περιβάλλον από τον καιρό. Για παράδειγμα βροχή, ομίχλη, χιόνι, αέρα με διάφορες πιθανές αλλαγές στην φωτεινότητα ανάλογα σε ποια κατεύθυνση βρίσκετε ο ήλιος.

Πέρα αυτού του αρχείου, θα πρέπει να βρούμε έναν τρόπο να φορτώνουμε εικόνες απευθείας από τη κάμερα, για να γίνεται ποιο άμεσα η επεξεργασία και η εύρεση αντικειμένου. Καθώς και να δώσουμε έναν συγκεκριμένο χρόνο αποτύπωσης της νέας εικόνας που δεν θα πρέπει να είναι πολύ μεγάλος για να μην υπάρχει κίνδυνος να περάσει κάποιο άλλο αντικείμενο και να μην γίνει αντιληπτό αλλά και να μην είναι πολύ μικρός για να προλαβαίνει το σύστημα μας να ελέγχει την προηγούμενη εικόνα. Αυτό περισσότερο θα χρειαστεί αν πέρα από το προγραμματιστικό μέρος ασχοληθούμε και με την υλοποίηση του όλου «κυκλώματος» λειτουργίας.

Και τέλος το ιδανικότερο θα ήταν να υπάρχει ένα αρχείο με φωτογραφίες από όλες τις πλευρές κάθε σκάφους που υπάρχει στον κόσμο έτσι ώστε η αναγνώριση να ήταν ολοκληρωμένη μαζί με το μήνυμα που θα έδινε στον χρήστη ότι υπάρχει σκάφος στην περιοχή θα μπορούσε να του αναφέρει και το μοντέλο του.

Βιβλιογραφία

1. *The MathWorks*, <http://www.mathworks.com>
2. L.W. Fausett, *Applied Numerical Analysis using MATLAB*, Pearson-Prentice Hall, New Jersey, 2008.
3. D.J. Ingham and N.J. Ingham, *MATLAB Guide* (2nd Ed.), SIAM, 2005.
4. C.B. Moler, *Numerical Computing with MATLAB*, SIAM, 2004.
5. H. Moore, *MATLAB for Engineers*, Pearson-Prentice Hall, New Jersey, 2007.
6. Γ.Δ. Ακρίβης και Β.Α. Δουγαλής, *Εισαγωγή στην Αριθμητική Ανάλυση*, Πανεπιστημιακές Εκδόσεις Κρήτης, Ηράκλειο, 2004.
7. Ευάγγελος Β. Χατζίκος, *MATLAB 7 για μηχανικούς*, Εκδόσεις Τζιόλα, 2007.
8. Γιώργος Γεωργίου & Χρίστος Ξενοφώντος, *Εισαγωγή στη MATLAB*, Τμήμα Μαθηματικών και Στατιστικής Πανεπιστήμιο Κύπρου, 2007.
9. *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 2e, Steven C. Chapra, *Tufts University*.
10. *Digital Image Fundamentals in MATLAB*, Mohammad Nuruzzaman, *King Fahd University of Petroleum and Minerals, AuthorHouse*, 2005
11. *Digital Image Processing Using MATLAB*, Rafael C. Gonzalez, *University of Tennessee*, Richard E. Woods, *MedData Interactive*, Steven L. Eddins, *The MathWorks*, Prentice Hall, 2004
12. *ELEMENTARY MATHEMATICAL and COMPUTATIONAL TOOLS for ELECTRICAL and COMPUTER ENGINEERS USING MATLAB*, Jamal T. Manassah
13. Καλές πρακτικές για την ψηφιοποίηση, <http://digitization.hpclab.ceid.upatras.gr/index.php>