ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΜΕ ΧΡΗΣΗ ΣΥΓΧΡΟΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ

Φοιτητής : Γρυλλάκης Κωνσταντίνος

Επιβλέπων Καθηγητής : Δρ Κωνσταντάρας Αντώνιος

Εξεταστής Α' :

Εξεταστής Β' :

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

Περιεχόμενα

Abstract

Digital image processing is the use of computer algorithms to perfom image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analog image processing, it allows a much wider range of algorithms to be applied to the input data, and can avoid problems such as the build-up of noise and signal distortion during processing.

Ψηφιακή επεξεργασία εικόνας

Η ψηφιακή επεξεργασία εικόνας είναι η χρήση των αλγορίθμων από τους υπολογιστές για να επεξεργαστούν τις ψηφιακές εικόνες.Ως ένα υποπεδίο της ψηφιακής επεξεργασία σήματος, η ψηφιακή επεξεργασία εικόνας έχει πολλά πλεονεκτήματα σε σύγκριση με την αναλογική επεξεργασία εικόνας: επιτρέπει ένα πολύ μεγάλο φάσμα των αλγορίθμων που πρέπει να εφαρμόζονται για την εισαγωγή δεδομένων, και μπορούν να αποφευχθούν προβλήματα όπως η δημιουργία του θορύβου και σήματος στρέβλωση κατά την διάρκεια της μεταποίησης.

Ιστορία

Πολλές από τις τεχνικές της ψηφιακής επεξεργασίας εικόνας αναπτύχθηκαν στη δεκαετία του 1960 στα Jet Propulsion, MIT, Bell Labs, University of Maryland, και σε μερικά άλλα μερη, με εφαρμογή σε δορυφορικές εικόνες, τηλεγραφική φωτογραφία, ιατρική απεικόνηση, τηλεοπτικής τηλεφωνίας, η αναγνώρηση χαρακτήρων, και η φωτογραφική ενίσχυση. Όμως, το κόστος της επεξεργασίας ήταν αρκετά υψηλό με τον εξοπλισμό της εποχής. Στην δεκαετία του 1970, πολλαπλασιάζεται η ψηφιακή επεξεργασία εικόνας, όταν έκαναν την εμφανισεί τους

φθηνότεροι υπολογιστές και το αποκλειστικό υλικό έγινε πλέον διαθέσιμο. Εικόνες θα μπορούσαν τότε να υποβάλλονται σε επεξεργασία σε πραγματικό χρόνο, για ορισμένα ειδικά προβλήματα, όπως είναι τα τηλεοπτικά πρότυπα μετατροπής, όπως υπολογιστές γενικής χρήσης έγιναν πιο γρήγοροι, άρχισαν να αναλάβει το ρόλο του αποκλειστικού υλικού για όλους, αλλά το πιο εξειδικευμένο και υψηλές υπολογιστικές εργασίες.

Με το γρήγορο υπολογιστή και επεξεργαστές σήματος διαθέσιμες τη δεκαετία του 2000, ψηφιακή επεξεργασία εικόνας έχει καταστεί η πιο κοινή μορφή επεξεργασίας εικόνας, και χρησιμοποιείται γενικά, διότι δεν είναι μόνο η πλέον ευέλικτη μέθοδος, αλλά και η φθηνότερη.

Καθήκοντα

Η ψηφιακή επεξεργαία εικόνας επιτρέπει τη χρήση των πιο πολύπλοκων αλγορίθμων για επεξεργασία εικόνας, και ως εκ τούτου μπορεί να προσφέρει τόσο πιο εξελιγμένες επιδόσεις σε απλές εργασίες, καθώς και η εφαρμογή των μεθόδων που θα ήταν αδύνατον από αναλογικά μέσα.

Ειδικότερα, η ψηφιακή επεξεργασία εικόνας είναι η μοναδική πρακτική τεχνολογία για:

- Ταξινόμηση
- Εξαγωγή χαρακτηριστηκών
- Αναγνώρηση προτύπων
- Προβολή
- multi-κλίμακα ανάλυση σήματος

Ορισμένες τεχνικές που χρησιμοποιούνται στην ψηφιακή επεξεργασία εικόνας:

- Ανάλυση κύρια συστατικά
- Ανεξάρητη ανάλυση συνιστώσα
- Αυτό-οργάνωση χάρτες
- Νευρωνικά δίκτυα

Αναγνώρηση των δεδομένων εικόνας

Η λειτουργία imread διαβάζει μια εικόνα από οποιαδήποτε υποστηριζόμενη μορφή ενός αρχείου εικόνας,σε οποιαδήποτε από τα υποστηριζόμενα βάθη bit. Περισσότερες μορφές αρχείων εικόνας χρησιμοποιούν 8-bit για την αποθήκευση των pixel. Όταν αυτά διαβάζονται στη μνήμη, τα αποθηκεύει το Matlab ως κατηγορία unit-8. Για μορφές αρχείων που υποστηρίζει 16-bit δεδομένων, όπως PNG και TIFF, το Matlab αποθηκεύει τις εικόνες ως κατηγορία unit-16.

Παράδειγμα: RGB=imread('football.jpg');

Γραφή δεδομένων εικόνας

Η λειτουργία imwrite γράφει μια εικόνα σε ένα αρχείο γραφικών σε μια από τις υποστηριζόμενες μορφές. Η πιο βασική σύνταξη για την imwrite είναι : παίρνει την εικόνα, όνομα μεταβλητής και ένα όνομα αρχείου.Αν συμπεριλάβουμαι την επέκταση στο όνομα του αρχείου , τότε το MATLAB συνάγει την επιθυμητή μορφή αρχείου του.

Παράδειγμα: imwrite(X,map,'moon2.bmp');

Καθορίζοντας πρόσθετες μορφές συγκεκριμένων παραμέτρων

Όταν χρησιμοποιούμε την imwrite με κάποιες μορφές γραφικών, μπορούμε να καθορίσουμε πρόσθετες παραμέτρους, για παράδειγμα με png αρχεία να καθορίσουμε το βάθος bit ως πρόσθετη παράμετρο.Αυτό το παράδειγμα γράφει μια εικόνα A σε ένα αρχείο jpeg, χρησιμοποιώντας μία επιπλέον παράμετρο για να καθορίσει την παράμετρο της ποιότητας συμπίεσης.

Παράδειγμα: imwrite(RGB,'myfile.jpg','Quality',100);

Ανάγνωση και εγγραφής δυαδικών εικόνων σε μορφή 1-bit

Σε ορισμένες μορφές αρχείων μια δυαδική εικόνα μπορεί να αποθηκευτεί σε μορφή 1-bit.Αν η μορφή αρχείου υποστηρίζεται από το matlab τότε γράφει τις δυαδικές εικόνες ως εικόνες 1-bit από προεπιλογή.Όταν διαβάζουμε μια δυαδική εικόνα σε μορφή 1-bit, το matlab την απεικονίζει στον χώρο εργασίας ως μια λογική διάταξη.Το παρακάτω παράδειγμα διαβάζει μια δυαδική εικόνα και την αποθηκεύει ως αρχείο TIFF.Επειδή η μορφή TIFF υποστηρίζει εικόνες 1-bit,το αρχείο είναι γραμμένο στο δίσκο σε μορφή 1-bit.

Παράδειγμα: BW=imread('moon.jpg');
imwrite(BW,'tif');

Μετατρέποντας τις μορφές αρχείου γραφικών

Για να αλλάξουμε τη μορφή γραφικών μιας εικόνας, χρησιμοποιούμε την imread για την εισαγωγή της εικόνας στο χώρο εργασίας του matlab και στη συνέχεια χρησιμοποιούμε τη συνάρτηση imwrite για να εξάγουμε την εικόνα, προσδιορίζοντας την κατάλληλη μορφή αρχείου.

Αυτό το παράδειγμα χρησιμοποιεί τη συνάρτηση imread για να διαβάσει μια εικόνα σε μορφή bitmap(BMP) στο χώρο εργασίας, στη συνέχεια γράφει την εικόνα bitmap σε ένα αρχείο χρησιμοποιόντας μορφή Portable Network Graphics(PNG).

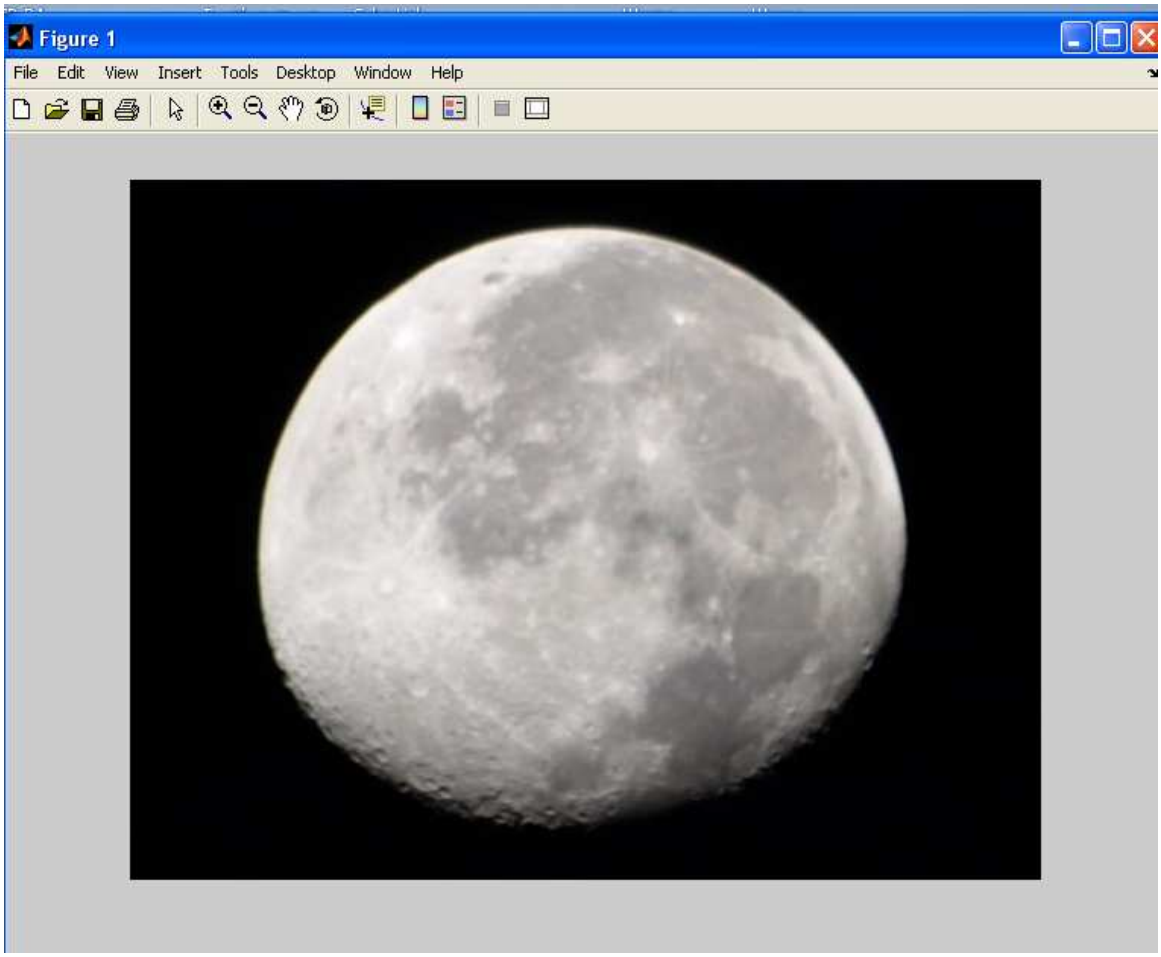
Παράδειγμα: bitmap=imread('moon.jpg','jpg');
Imwrite(bitmap,'moon2.png','png');

Χρησιμοποιώντας την imshow για να εμφανίσουμε τις εικόνες

Μπορούμε να χρησιμοποιήσουμε τη λειτουργία imshow για να εμφανίσουμε μια εικόνα που έχει είδη εισαχθεί στο χώρο εργασίας του MATLAB ή να εμφανίσουμε μια αποθηκευμένη εικόνα από ένα αρχείο γραφικών.Για παράδειγμα, αυτός ο κώδικας διαβάζει μια εικόνα στο χώρο εργασίας του MATLAB και δτη συνέχεια την εμφανίζει σε μορφή παραθήρου του MATLAB.

Παράδειγμα: moon=imread(‘fegari.tif’);

imshow(moon)



Προσδιορίζοντας την αρχική μεγέθυνση της εικόνας

Από προεπιλογή, η συνάρτηση imshow προσπαθεί να εμφανίσει την εικόνα στο σύνολό της,στο 100% της μεγένθυσης (ένα pixel οθόνης για κάθε pixel εικόνας). Πάντως, αν μια εικόνα είναι πολύ μεγάλη για να χωρέσει σε ένα παράθυρο σε ποσοστό 100% επί της οθόνης, η imshow μικραίνει την εικόνα ώστε να χωρέσει στην οθόνη και εμφανίζει ένα προειδοποιητικό μήνυμα στο χώρο εργασίας.

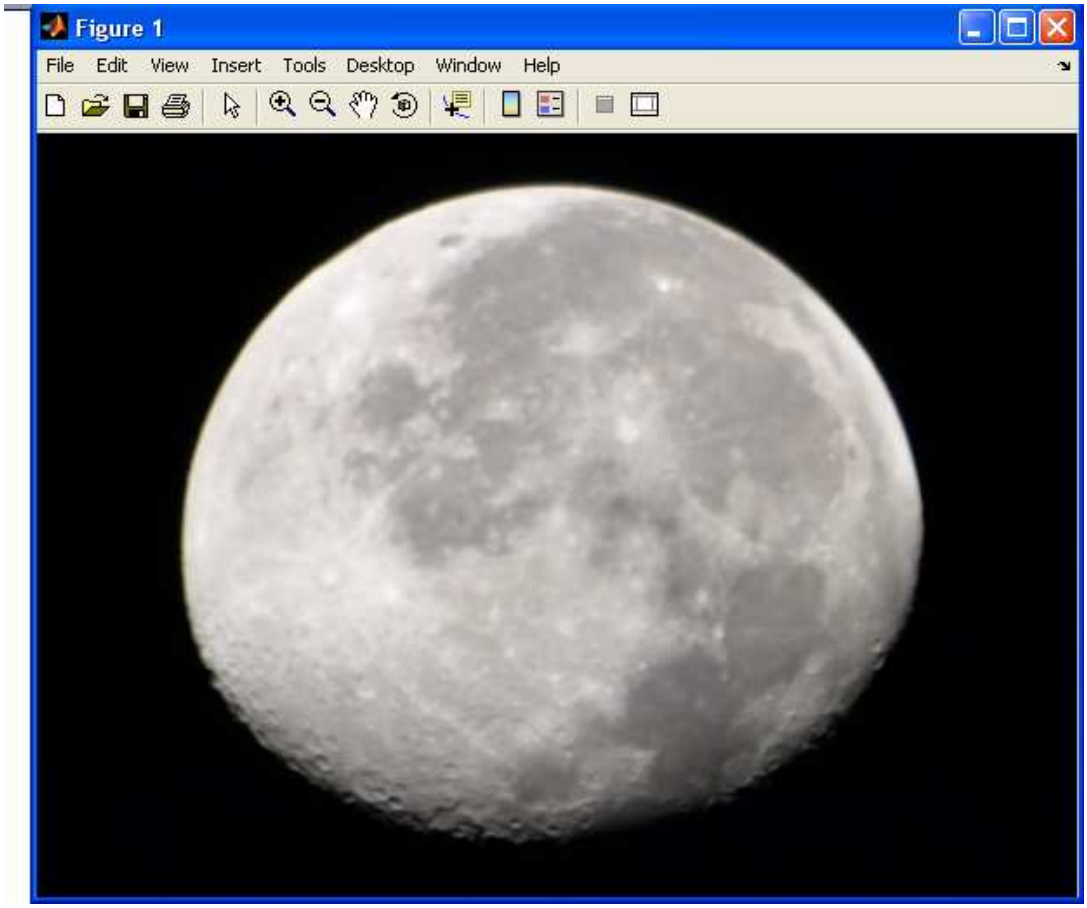
Για να παρακάμψουμε την αρχική προεπιλογή μεγέθυνσης της imshow, διευκρινίζουμε την αρχική παράμετρο μεγέθυνσης με την **'InitialMagnification'**. Για παράδειγμα, για να εμφανιστή η εικόνα σε μεγέθυνση 33% χρησιμοποιούμε τον παρακάτω κώδικα.

Παράδειγμα: moon=imread('fegari.jpg');
Imshow(moon,'InitialMagnification',33)

Ελέγχοντας την εμφάνιση του διαγράμματος

Η imshow από προεπιλογή εμφανίζει μια εικόνα σε ένα διάγραμμα με γκρι φόντο.Μπορούμε να αλλάξουμε αυτήν την προεπιλογή με την κατάργηση των συνόρων, χρησιμοποιόντας την παράμετρο **'Border'** της imshow, όπως φαίνεται στο παρακάτω παράδειγμα.

Παράδειγμα: imshow('fegari.jpg','Border','tight')



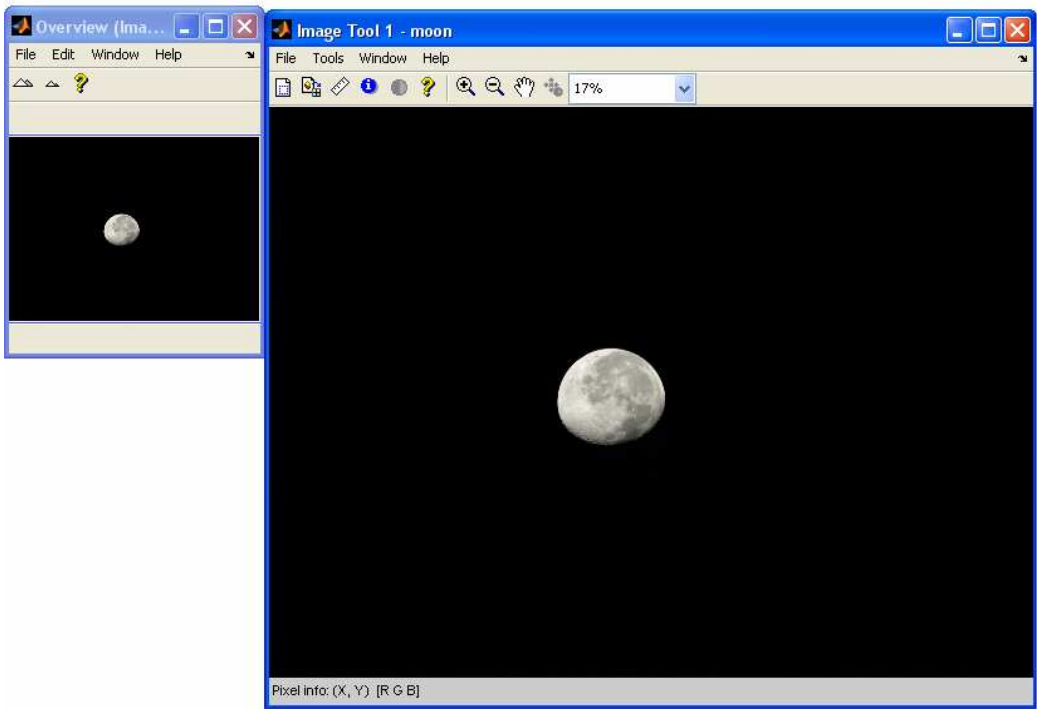
Χρησιμοποιόντας το Image Tool για να διερευνήσουμε τις εικόνες

Το Image Tool (εργαλείο εικόνας) είναι ένα εργαλείο απεικόνισης εικόνας το οποίο παρέχει πρόσβαση σε διάφορα άλλα συναφή εργαλεία όπως, το Pixel Region tool (εργαλείο περιοχής pixel), το Image Information tool (εργαλείο πληροφοριών εικόνας), και το Adjust Contrast tool (εργαλείο προσαρμογής αντίθεσεις).

Ανοίγοντας το Image Tool

Για να ξεκινήσει το Image tool (εργαλείο εικόνας), χρησιμοποιούμε την συνάρτηση imtool.

Παράδειγμα: moon=imread(‘fegari.jpg’);
imtool(moon)



Προσδιορίζοντας την αρχική μεγέθυνση εικόνας

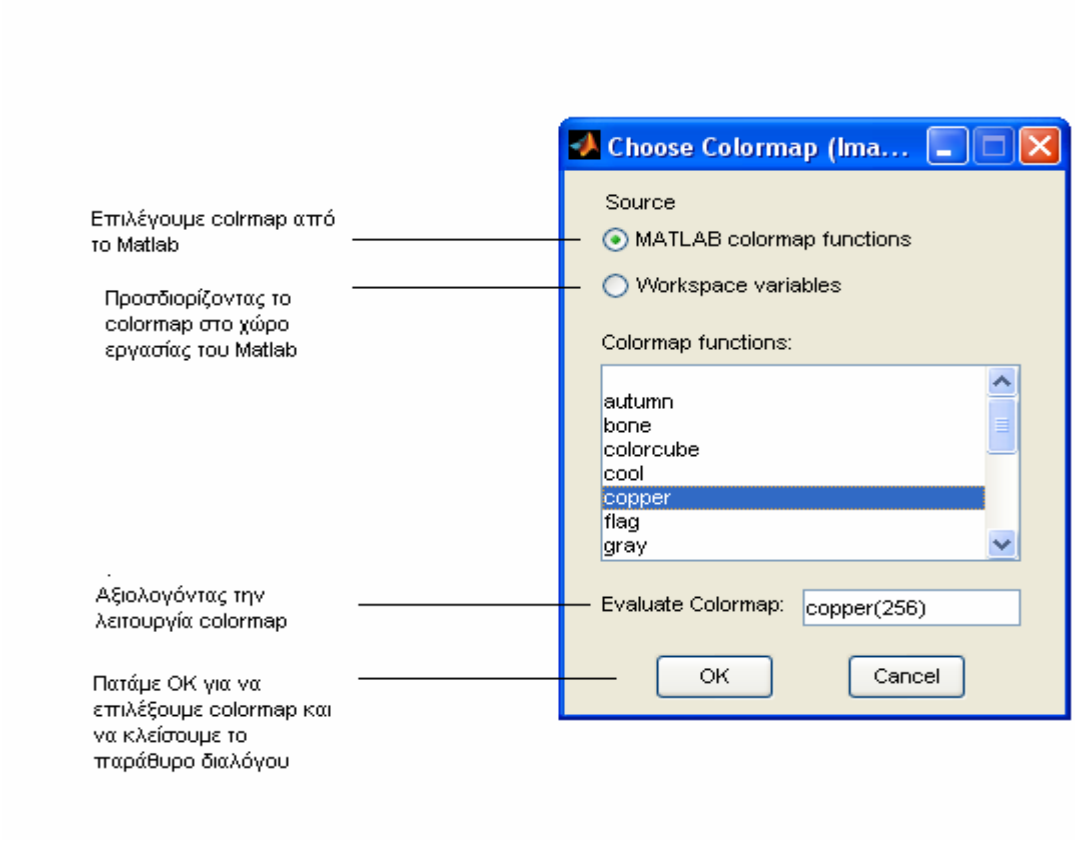
Όπως και με την imshow, έτσι και η συνάρτηση imtool προσπαθεί να εμφανίσει μια εικόνα στο σύνολό της σε μεγέθυνση 100%.

Παράδειγμα: pout=imread(‘pout.tif’);
imtool(pout,’InitialMgnification’,150);

Προσδιορίζοντας την colormap

Για να καθορίσουμε την παλέτα χρώματος,χρησιμοποιούμε για την εμφάνισή της την **Image tool** με μία εικόνα σε αποχρώσεις του γκρι,διαλέγουμε την επιλογή **Choose Colormap** από το μενού **Tools.** Αυτό ενεργοποιεί το εργαλείο **Choose Colormap**, όπως φαίνεται παρακάτω. Για να χρησιμοποιήσουμε το συγκεκριμένο εργαλείο μπορούμε να επιλέξουμε colormap από το matlab ή να επιλέξουμε μια μεταβλητή colormap από το χώρο εργασίας του matlab.

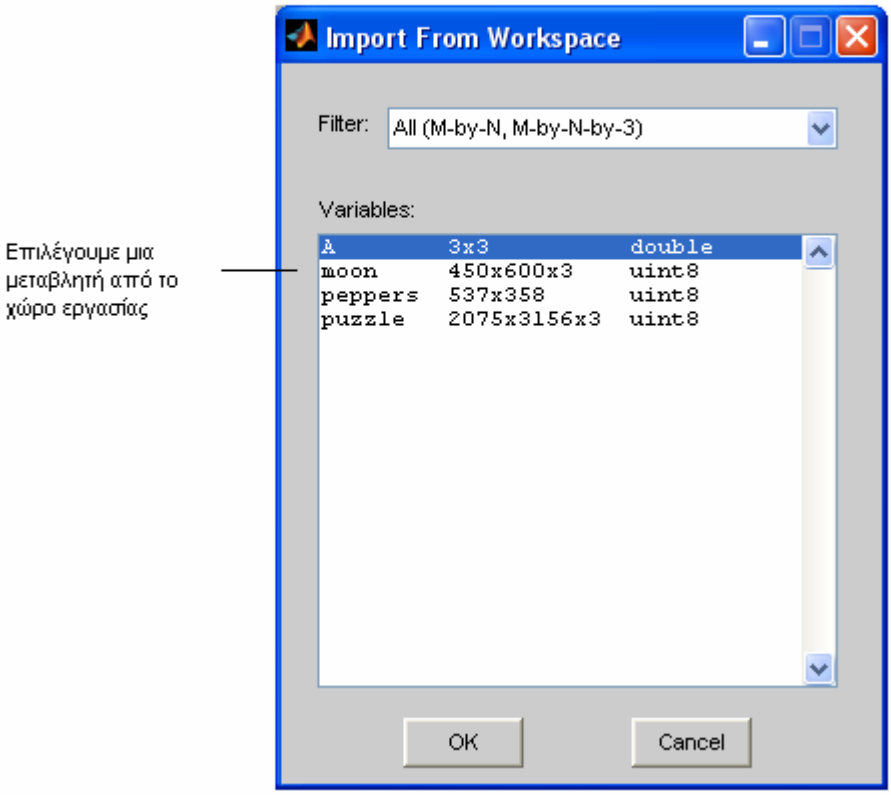
Επιλέγουμε Colormap Tool



Εισαγωγή δεδομένων εικόνας από το χώρο εργασίας

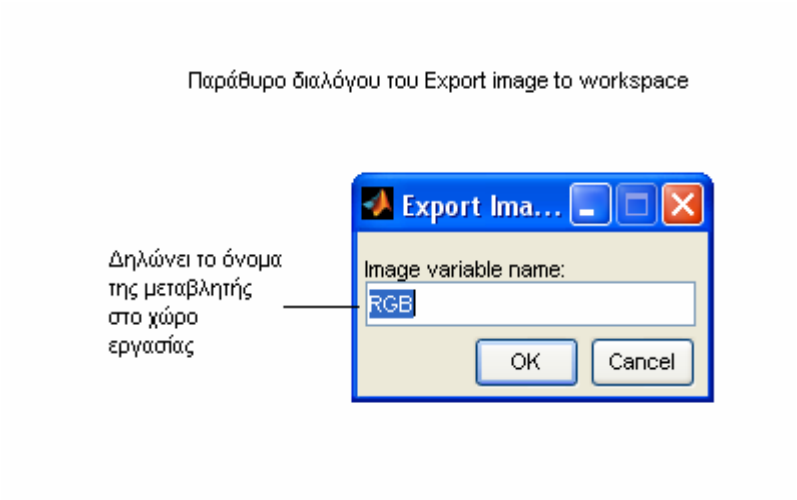
Μπορούμε να εισάγουμε δεδομένα εικόνας από το χώρο εργασίας του Matlab μέσα από το **Image Tool**, χρησιμοποιώντας την επιλογή **Import from Workspace** από το μενού **File** του Image tool. Στο παράθυρο διαλόγου, όπως φαίνεται παρακάτω, επιλέγουμε μια μεταβλητή από το χώρο εργασίας για να εμφανίσει την εικόνα.

Παράθυρο διαλόγου του Import from Workspace



Εξαγωγή δεδομένων εικόνας από το χώρο εργασίας

Για να εξάγουμε από την εμφανιζόμενη εικόνα στο image tool του χώρου εργασίας του Matlab, θα πρέπει να χρησιμοποιήσουμε την επιλογή **Export to Workspace** από το file μενού του image tool. Στο παράθυρο διαλόγου, όπως φαίνεται πιο κάτω, μπορούμε να καθορίσουμε το όνομα της μεταβλητής στο χώρο εργασίας. Από προεπιλογή το Image Tool συμπληρώνη το όνοματης μεταβλητής με **BW** για δυαδικές εικόνες, **RGB** για έγχρωμες εικόνες, και **I** για εικόνες σε αποχρώσεις του γκρι.



Χρησιμοποιώντας τη λειτουργία getimage για την εξαγωγή δεδομένων της εικόνας

Μπορούμε επίσης να χρησιμοποιήσουμε τη λειτουργία **getimage** για να φέρουμε δεδομένα της εικόνας από το image tool στο χώρο εργασίας του Matlab.Η συνάρτιση getimage ανακτά τα δεδομένα της εικόνας από την τρέχουσα λαβή γραφικών(Handle Graphics) του αντικειμένου της εικόνας. Επειδή από προεπιλογή το Image Tool δεν κάνει λαβές σε ορατά αντικείμενα, πρέπει να χρησιμοποιήσουμε από το toolbox την συνάρτηση **imgca** για να πάρουμε μια λαβή για τους άξονες της εικόνας που εμφανίζεται στο Image tool.

Παράδειγμα: moon=getimage(imgca)

Κλείνοντας το Image Tool

Για να κλείσουμε το παράθυρο του Image tool, χρησιμοποιούμε το πλήκτρο '**X**' στη γραμμή τίτλου του παραθύρου ή διαλέγουμε την επιλογή '**close**' από το file μενού του Image tool. Μπορούμε επίσης να χρησιμοποιήσουμε τη λειτουργία imtool για να κλείσουμε το image tool, και ακόμα να κλείσουμε κάθε συναφή εργαλεία που είναι ανοικτά.

Παράδειγμα: imtool close all

Εκτύπωση της εικόνας στο Image Tool

Για να εκτυπώσουμε την εικόνα που εμφανίζεται που εμφανίζεται στο Image tool, διαλέγουμε την επιλογή **Print to figure** από το μενού File. Το Image Tool δημιουργεί ένα καινούριο παράθυρο όπου εμφανίζει την εικόνα. Χρησιμοποιούμε την επιλογή **Print** από το μενού File του καινούριου παραθύρου για να εκτυπώσουμε την εικόνα.

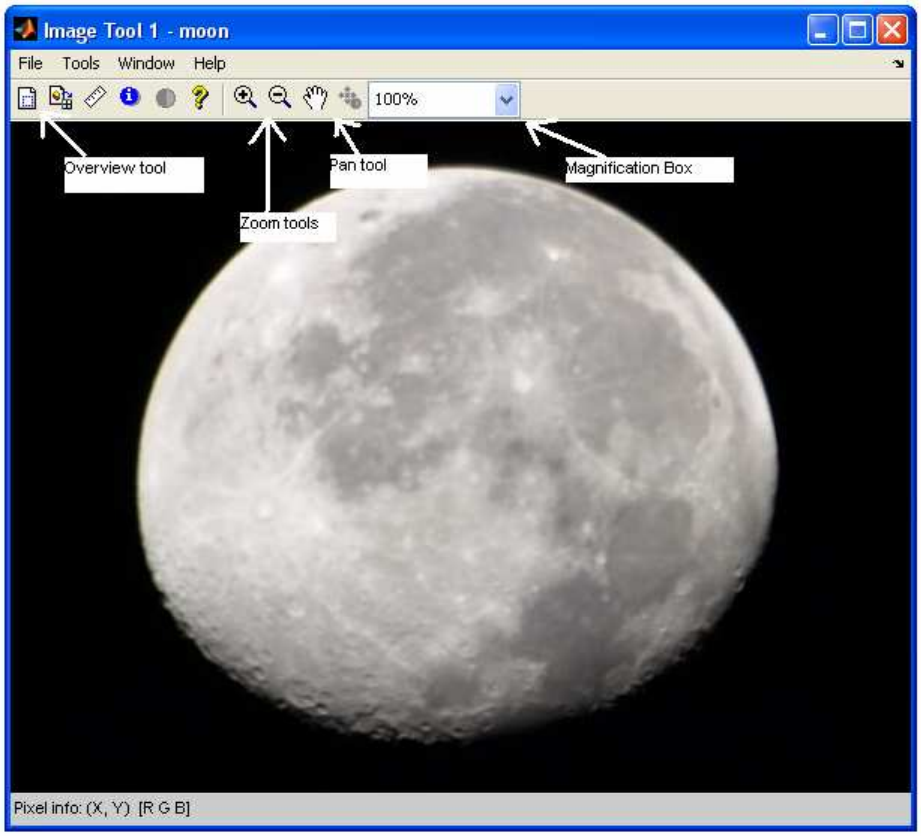
Χρησιμοποιόντας βοηθήματα πλοήγησης στο Image Tool

Overview tool – Παρέχει μια προβολή του συνόλου της εικόνας για να μας βοηθήσει να καταλάβουμε ποιο ποσοστό ή εικόνα εμφανίζεται στο Image Tool.

Pan tool – Μπορούμε να κάνουμε κλικ και να αρπάξει την εικόνα που εμφανίζετε και να την μετακινήσουμε μέσα στο εργαλείο εικόνας.

Zoom tools – Μας επιτρέπει να κάνουμε μεγέθυνση ή σμίκρυνση στην εικόνα.

Magnification Box – Μας επιτρέπει να καθορίσουμε το ακριβές μέγεθος της εικόνας.

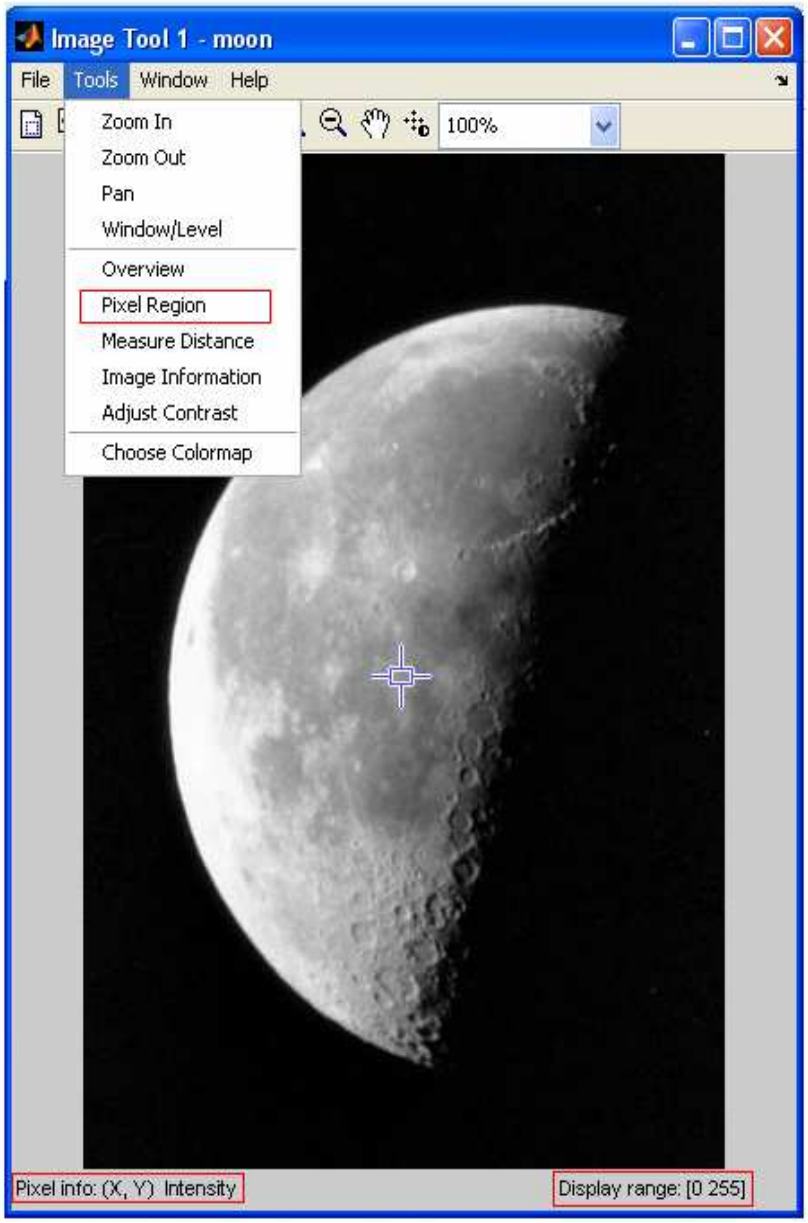


Βρίσκοντας πληροφορίες για τα pixels μιας εικόνας

Pixel Information tool – Απεικόνιση της θέσης και η αξία των pixels που δείχνει ο δρομέας στο παράθυρο Image Tool.

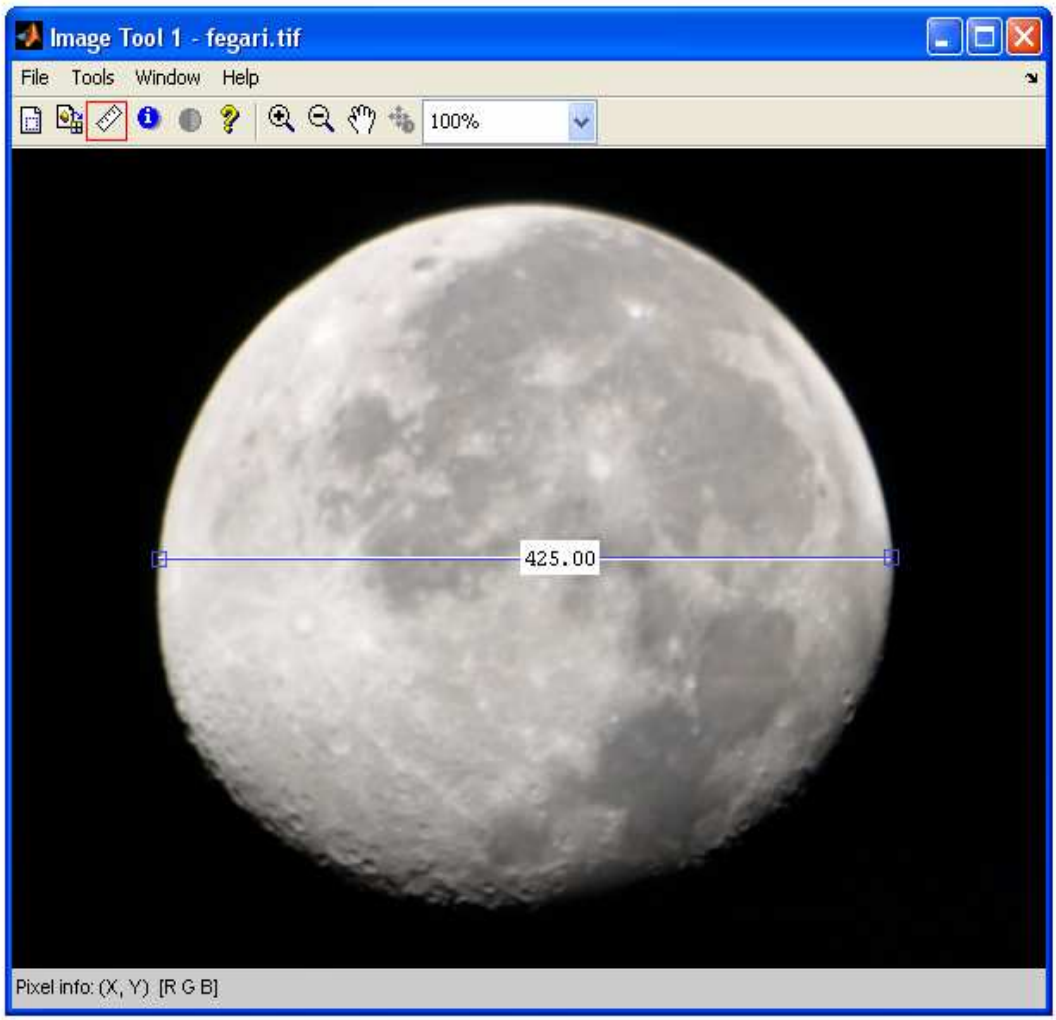
Display Range tool – Εμφανίζει το φάσμα απεικόνισεις της εικόνας στο παράθυρο Image tool.

Pixel Region tool – Εμφανίζει την αξία των pixels σε μια συγκεκριμένη περιοχή της εικόνας.



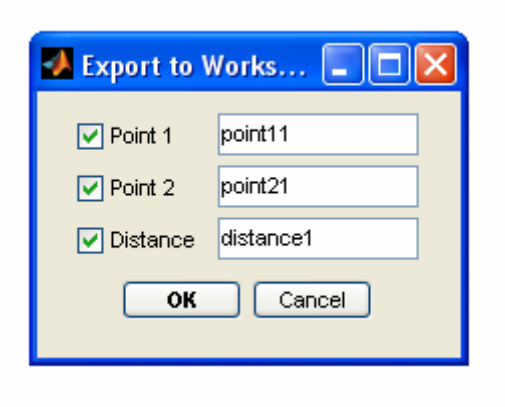
Μέτρηση χαρακτηριστικών σε μια εικόνα

Χρησιμοποιώντας το εργαλείο απόστασης



Εξαγωγή δεδομένων τελικό σημείο και απόσταση

Για να αποθηκεύσουμε την πληροφορία του τελικού σημείου και απόσταση, κάνουμε δεξί κλικ στην γραμμή της απόστασης και επιλέγουμε **export to wprkspace.** Στο παράθυρο διαλόγου μπορούμε να καθορίσουμε τα ονόματα των μεταβλητών που χρησιμοποιούνται πριν την αποθήκευση των πληροφοριών.



Προσαρμογή της εμφάνισης του Distance Tool

Χρησιμοποιώντας το μενού του Distance tool, μπορούμε να προσαρμόσουμε την εμφάνιση και τη συμπεριφορά, όπως :

Show Distance Label – Επιλέγουμε την εμφάνισει ή απόκρυψει τις ετικέτας που αναγράφει την απόσταση.

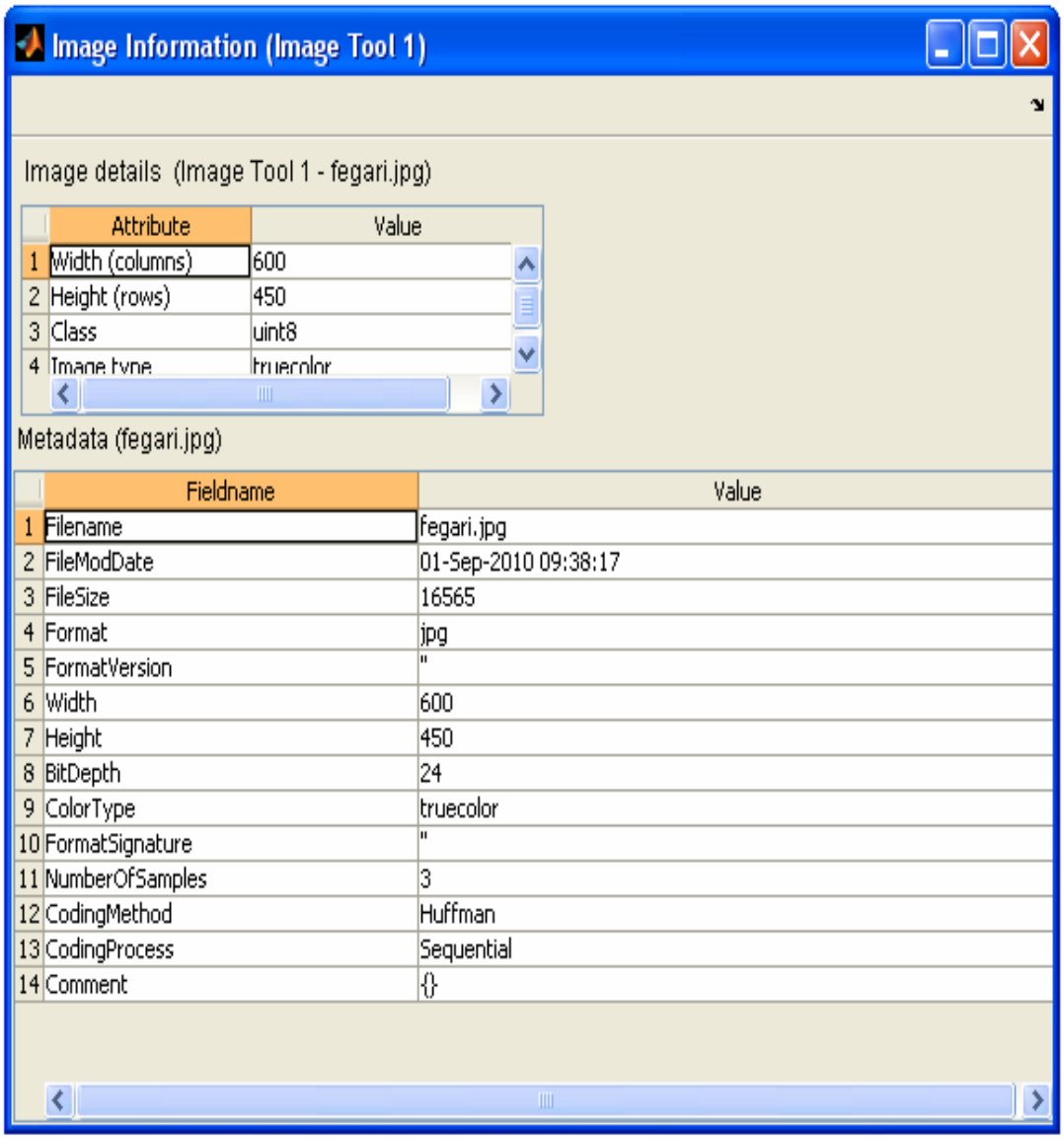
Set line color – Αλλάζουμε το χρώμα που χρησιμοποιήται για την εμφάνισει της γραμμής της απόστασης.

Constrain drag – Επιλέγουμε την κίνηση του εργαλείου για οριζόντια, κάθετη ή συνδιασμένη κίνηση.

Delete – Διαγράφουμε το αντικείμενο distance tool.

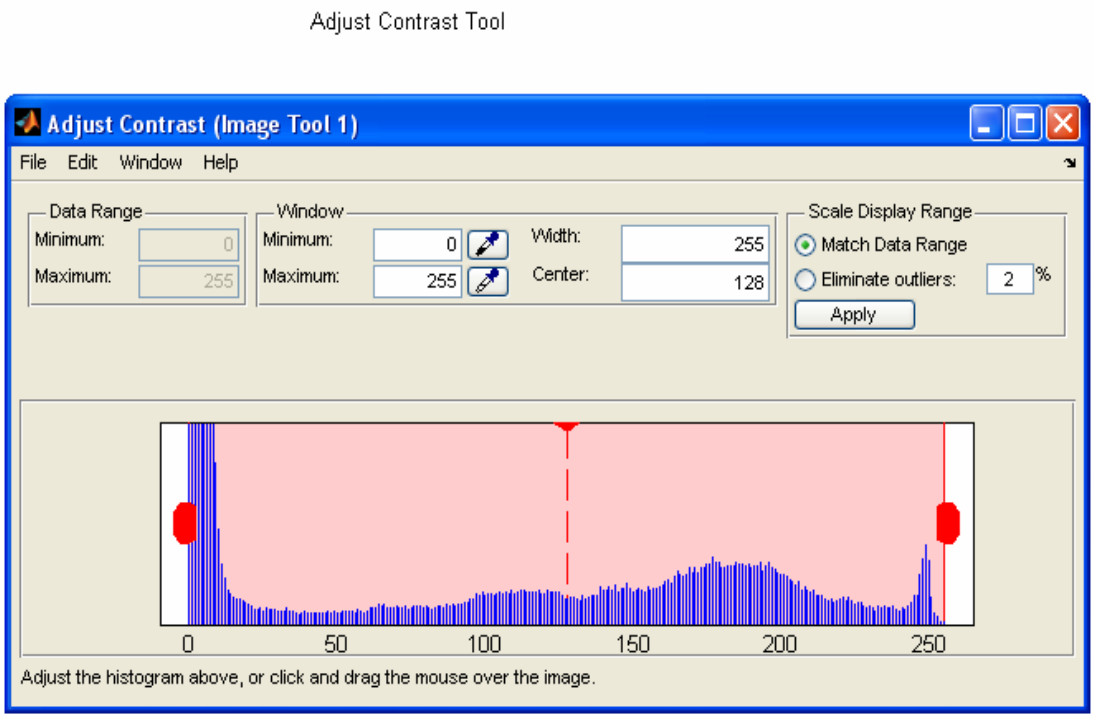
Παίρνοντας πληροφορίες σχετικά με την εικόνα

Για να ανοίξουμε το εργαλείο πληροφόρησης εικόνας κάνουμε κλικ στο κουμπί ‘i’ από την γραμμή εργαλείων ή διαλέγουμε την επιλογή **Image Information** από το μενού **Tools** του Image Tool.



Προσαρμογή της αντίθεσης και της φωτεινώτητας της εικόνας

Για να ρυθμίσουμε την αντίθεση και την φωτεινότητα της εικόνας που εμφανίζεται στο Image Tool, χρησιμοποιούμε το Adjust Contrast tool.

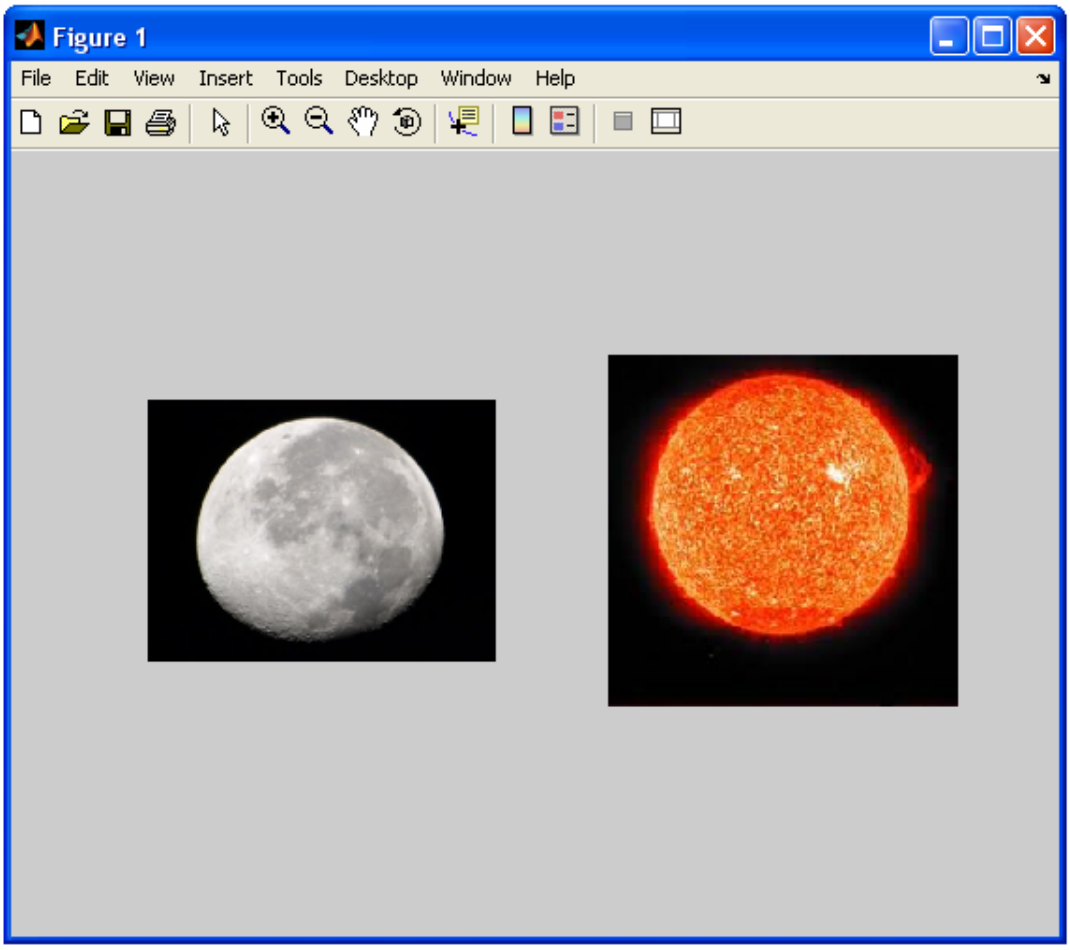


Προβολή πολλαπλών εικόνων

Βλέπουμε πολλαπλές εικόνες σε ίδιο σχήμα

Μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **imshow** ή τη συνάρτηση **subimage** μαζί με τη συνάρτηση **subplot** του Matlab για την εμφάνιση πολλών εικόνων σε ένα ενιαίο παράθυρο.

Παράδειγμα: [X1,map1]=imread('fegari.jpg');
[X2,map2]=imread('sun.jpg');
subplot(1,2,1),imshow(X1,map1)
subplot(1,2,2),imshow(X2,map2)



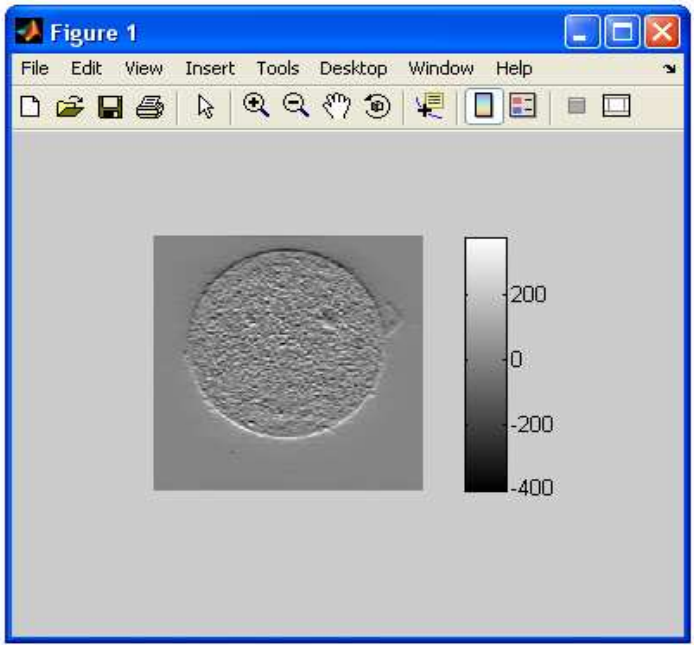
Ειδικές τεχνικές απεικόνισεις

Προσθήκη Colorbar

Για να προσθέσουμε ένα **colorbar** σε μια εικόνα που εμφανίζεται στο Image Tool, διαλέγουμε την επιλογή **Print to Figure** από το μενού **File** του Image Tool. Το Image Tool εμφανίζει την εικόνα σε ξεχωριστό παράθυρο, στο οποίο μπορούμε να προσθέσουμε ένα colorbar επιλέγοντας από το μενού **Insert** το **Colorbar.**

Παράδειγμα: RGB = imread(‘sun.jpg’);

```
I=rgb2gray(RGB);
h=[1 2 1; 0 0 0; -1 -2 -1];
I2=filter(h,I);
imshow(I2,’DisplayRange’,[]),colorbar
```

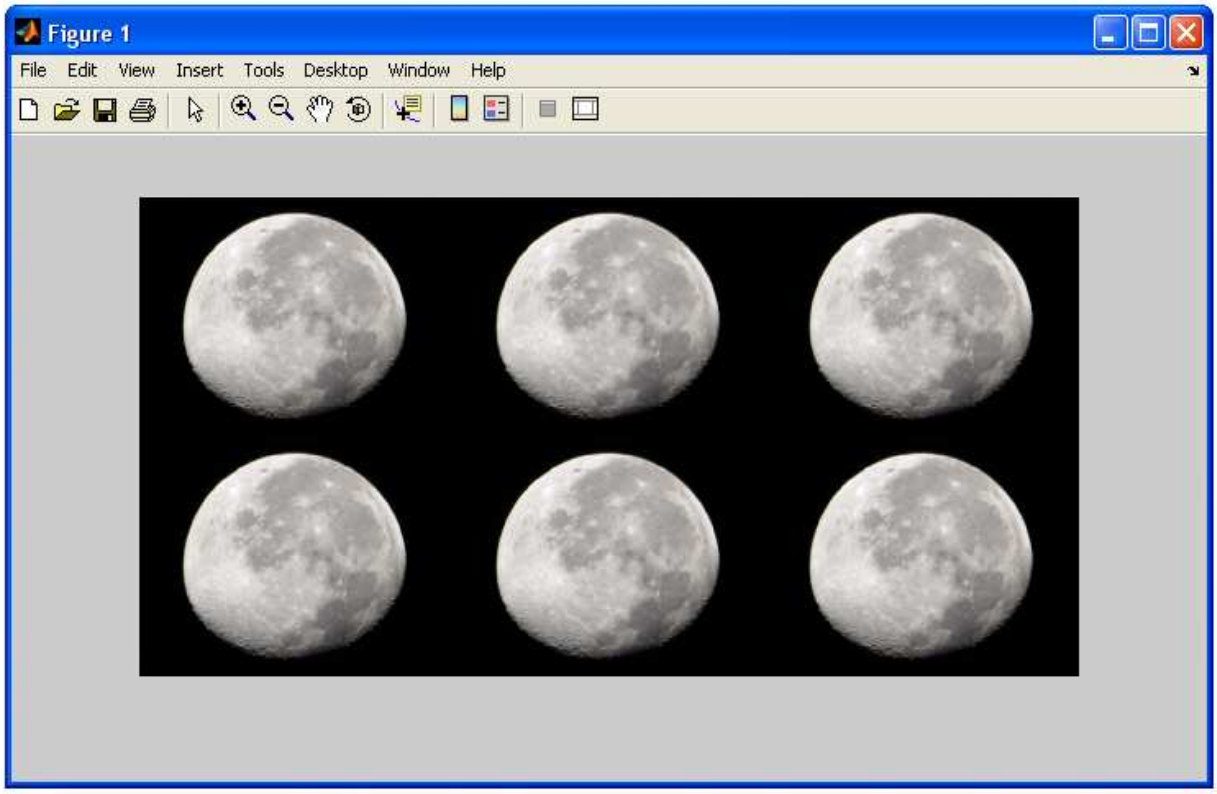


Βλέπουμε πολλαπλά καρέ της εικόνας σε μία καρτέλα

Για να δούμε κατευθείαν πολλά πλαίσια της εικόνας, θα χρεισημοποιήσουμε την συνάρτηση **montage.** Το montage εμφανίζει όλα τα πλαίσια της εικόνας, τα οργανώνει σε ένα ορθογώνιο πλέγμα. Το montage της εικόνας είναι ένα ξεχωριστό αντεικίμενο εικόνας.

Παράδειγμα: moon = imread('fegari.jpg');

```
moonArray=repmat(moon,[1 1 1 6]);
montage(moonArray);
```



Μετασχηματισμός χώρου

Αλλαγή μεγέθους μιας εικόνας

Χρησιμοποιώντας την συνάρτηση **imresize,** μπορούμε να καθορίσουμε το μέγεθος της εικόνας εξόδου με δύο τρόπους :
Καθορίζοντας το συντελεστή μεγέθυνσης που θα χρησιμοποιήσουμε στην εικόνα
Προσδιορίζοντας τις διαστάσης της εικόνας εξόδου.

Παράδειγμα: I=imread(‘moon.jpg’);
J=imresize(I,1.25);
imshow(I)
figure, imshow(J)

Προσδιορίζοντας το μέγεθος της εικόνας εξόδου

Μπορούμε να καθορίσουμε το μέγεθος της εικόνας εξόδου, πέρνοντας από ένα φορέα που να παρέχει τον αριθμό των γραμμών και στηλών στην εικόνα εξόδου.

Παράδειγμα: I=imread('moon.jpg');
J=imresize(I,[100 150]);
Figure, imshow(J)

Περιστροφή εικόνας

Η συνάρτιση **imrotate** δέχεται δύο κύριες μεταβλητές :

· Η εικόνα που θέλουμε να περιστρέψουμε

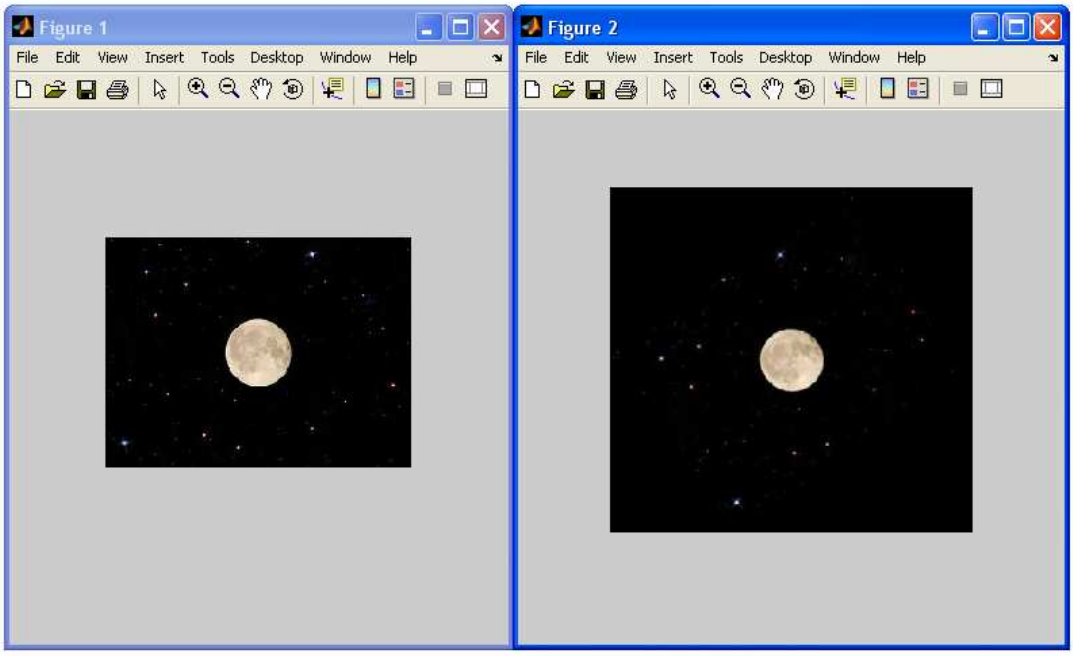
· Η γωνία περιστροφής

Παράδειγμα: J=imrotate(I,35)

Προσδιορίζοντας τη μέθοδο παρεμβολής

Από προεπιλογή, η **imrotate** χρησιμοποιεί την πλησιέστερη γειτονική παρεμβολή για να προσδιορίσει την αξία των pixels στην εικόνα, αλλά μπορούμε να καθορίσουμε και άλλες μεθόδους παρεμβολής.

Παράδειγμα: I=imread('moon.jpg');
J=imrotate(I,35,'bilinear');
Imshow(I)
figure, imshow(J)



Αρχική εικόνα

Περιστρεφόμενη εικόνα

Περικοπή μιας εικόνας

Για να εξάγουμε ένα ορθογώνιο τμήμα της εικόνας, χρησιμοποιούμε τη συνάρτηση **imcrop.** Η **imcrop** δέχεται δύο κύριες μεταβλητές :

· Την εικόνα που θα περικοπεί

· Της συντεταγμένες του ορθογωνίου που θα ορίσουμε για την περιοχή περικοπης.

Παράδειγμα: imshow moon.jpg

I=imcrop;

Imshow(I);

Επιλέγουμε σημεία ελέγχου

Για να προσδιορίσουμε τα σημεία ελέγχου κάνουμε μια διεργασία τεσσάρων σταδίων :

Για να ξεκινήσουμε το εργαλείο, προσδιορίζουμε την εικόνα εισροών και την εικόνα βάσης.

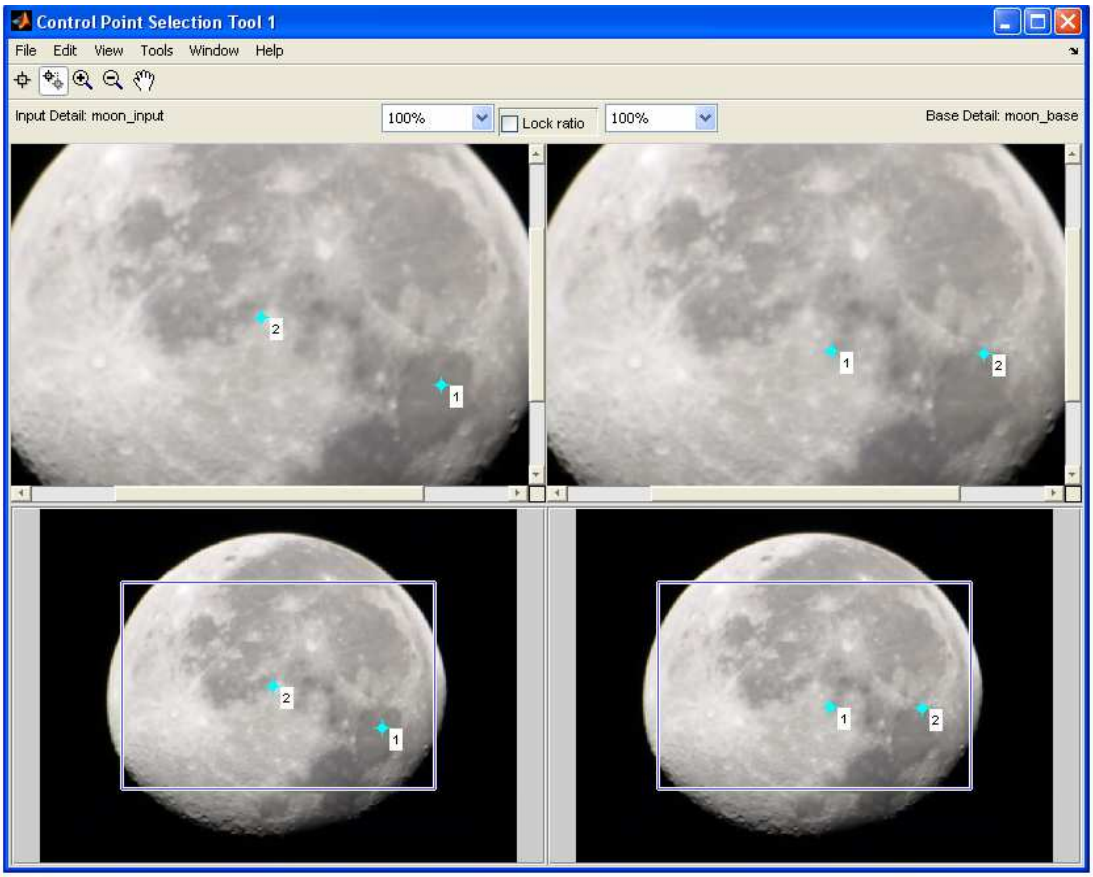
Χρησιμοποιούμε τη βοήθεια της πλοήγησης για να εξερευνήσουμε την εικόνα, ψάχνει για οπτικά στοιχεία που μπορεί να αναγνωρίσει και στις δύο εικόνες.

Η συνάρτηση **cpselect** μας παρέχει πολλούς τρόπους για να πλοηγηθούμε μέσα στην εικόνα, για να γυρίσουμε και να ζουμάρουμε περιοχές της εικόνας ώστε να τις προβάλουμε με περισσότερες λεπτομέρειες.

Προσδιορίζουμε και ελέγχουμε αν ταιριάζουν ζεύγη σημείων ανάμεσα στην εικόνα εισροής και την εικόνα βάσης.

Αποθηκεύουμε τα σημεία ελέγχου στο χώρο εργασίας του Matlab.

Παράδειγμα:
```
moon_base=imread('moon.jpg');
moon_input=moon_base;
cpselect(moon_input, moon_base);
```

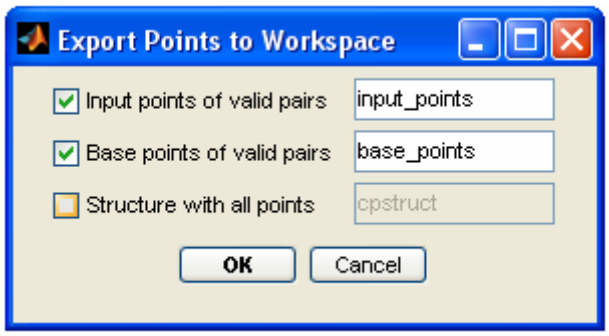


Εξαγωγή σημείων ελέγχου στο χώρο εργασίας

Για να αποθηκεύσουμε τα σημεία ελέγχου στο χώρο εργασίας του Matlab :

· Επιλέγουμε **File** στην γραμμή εργαλείων του Control Point Selection Tool.

· Μετά διαλέγουμε **Export Points to Workspace**, και εμφανίζεται το παρακάτω παράθυρο



Φιλτράρισμα και γραμμική σχεδίαση φίλτρων

Το φιλτράρισμα είναι μια τεχνική για την τροποποίηση ή την ενίσχυση μιας εικόνας. Για παράδειγμα μπορούμε να φιλτράρουμε μια εικόνα για να τονίσουμε ορισμένα χαρακτηριστηκά ή να αφαιρέσουμε άλλες λειτουργίες. Οι επεξεργασμένες εικόνες υλοποιούνται με εργασίες φιλτραρίσματος που περιλαμβάνουν την εξομάλυνση, το ακόνισμα και την ενίσχυση στα άκρα.

Φιλτράρισμα με χρήση της imfilter

Το φιλτράρισμα των εικόνων, με συσχέτιση ή με συνέλιξη, μπορεί να γίνει χρησιμοποιώντας την συνάρτιση imfilter. Αυτό το παράδειγμα φίλτρων χρησιμοποιεί μία εικόνα 5x5 που περιέχει ίσο βάρος. Ένα τέτοιο φίλτρο συχνά ονομάζεται μέσος όρος φίλτρων.

Παράδειγμα: I=imread(‘fegari.jpg’);

```
h=ones(5,5)/35;

I2=imfilter(I,h);

subplot(1,2,1), imshow(I), title(‘Αρχική Εικόνα’);

subplot(1,2,2), imshow(I2), title(‘Φιλτραρισμένη Εικόνα’);
```

Χρησιμοποιόντας **προκαθορισμένους τύπους φίλτρων**

Η συνάρτηση **fspecial** παράγει διάφορα είδη προκαθορισμένων φίλτρων, με την μορφή των πυρήνων συσχέτισης. Το παράδειγμα μας δείχνει την εφαρμογή μιας μάσκας φίλτρου **unsharp** σε ασπρόμαυρη εικόνα. Το φίλτρο συγκάλυψης unsharp έχει ως αποτέλεσμα να καθιστά τα άκρα και τις λεπτομέρειες στην εικόνα πιο οξύ.

Παράδειγμα:
```
I=imread('fegari.jpg');
h=fspecial('unsharp');
I2=imfilter(I,h);
subplot(1,2,1), imshow(I), title('Αρχική Εικόνα');
subplot(1,2,2), imshow(I2), title('Φιλτραρισμένη Εικόνα');
```

Μορφολογικές εργασίες

Μορφολογία είναι μία ευρεία σειρά λειτουργιών επεξεργασίας εικόνας που βασίζεται σε σχήματα. Η μορφολογική λειτουργία εφαρμόζει ένα διαρθρωτικό στοιχείο σε μία εικόνα εισόδου, δημιουργόντας μια εικόνα εξόδου του ίδιο μεγέθους. Οι πιο βασικές μορφολογικές εργασίες είναι η διαστολή και η διάβρωση. Σε μία μορφολογική εργασία, το κάθε pixel της εικόνας εξόδου βασίζεται σε μία σύγκριση των αντίστοιχων pixel της εικόνας εισόδου με τα γειτονικά τους. Επιλέγουμε το μέγεθος και το σχήμα της γειτονιάς, μπορούμε να δημιουργίσουμε μια μορφολογική εργασία που να είναι ευαίσθητη στα ειδικά σχήματα της εικόνας εισόδου.

Διαστολή και Διάβρωση

Η διαστολή και η διάβρωση είναι δύο βασικές μορφολογικές εργασίες. Η διαστολή προσθέτει pixels στα όρια των αντικειμένων σε μία εικόνα, ενώ η διάβρωση καταργεί pixels επί τον ορίων αντικειμένων. Ο αριθμός των pixels που προσθέτει ή αφαιρεί από τα αντικείμενα σε μία εικόνα εξαρτάται από το μέγεθος και το σχήμα του στοιχείου της διάρθωσης που χρησιμοποιήται για την επεξεργασία της εικόνας.

Διαβρώνοντας την εικόνα

Για να διαβρώσουμε μια εικόνα, χρησιμοποιούμε τη συνάρτηση **imerode.** Η συνάρτηση imerode δέχεται δύο κύριες παραμέτρους : Την εικόνα εισαγωγής προς επεξεργασία και το διαρθρωτικό στοιχείο, επιστρέφοντας με την συνάρτηση **strel,** μία δυαδική μήτρα που οριοθετεί τη γειτονιά του διαρθρωτικού στοιχείου.

Παράδειγμα: BW1=imread('fegari.jpg');
SE=strel('arbitrary',eye(13));
BW2=imerode(BW1,SE);
subplot(1,2,1), imshow(BW1), title('Αρχική Εικόνα');
subplot(1,2,2), imshow(BW2), title('Φιλτραρισμένη Εικόνα');

Συνδυάζοντας διαστολή και διάβρωση

Η διαστολή και η διάβρωση συχνά χρησιμοποιούνται σε συνδυασμό με την εφαρμογή της επεξεργασίας εικόνας. Για παράδειγμα, ο ορισμός του μορφολογικού ανοίγματος μιας εικόνας είναι η διάβρωση που ακολουθήται από διαστολή, χρησιμοποιώντας το ίδιο διαρθωτικό στοιχείο και για τις δύο λειτουργίες. Σχετική λειτουργία, είναι μορφολογικό κλείσιμο μιας εικόνας, είναι η αντίστροφη : αποτελείτε από τη διαστολή που ακολουθείται από διάβρωση με το ίδιο διαρθωτικό στοιχείο.

Μορφολογικό άνοιγμα

Μπορούμε να χρησιμοποιήσουμε το μορφολογικό άνοιγμα για την άρση των μικρών αντικειμένων από την εικόνα διατηρόντας παράλληλα το σχήμα και το μέγεθος των μεγαλύτερων αντικειμένων στην εικόνα.

Παράδειγμα:
```
BW1=imread('fegari.jpg');
SE=strel('rectangle',[40 30]);
BW2=imerode(BW1,SE);
subplot(1,2,1), imshow(BW1), title('Αρχική Εικόνα');
subplot(1,2,2), imshow(BW2), title('Φιλτραρισμένη Εικόνα');
```

Βασικές συναρτήσης διαστολής και διάβρωσης

Αυτή η ενότητα περιγράφει δύο κοινές λειτουργίες επεξεργασίας εικόνας που βασίζονται σε διαστολή και διάβρωση.

· Διάγραμμα σκελετού

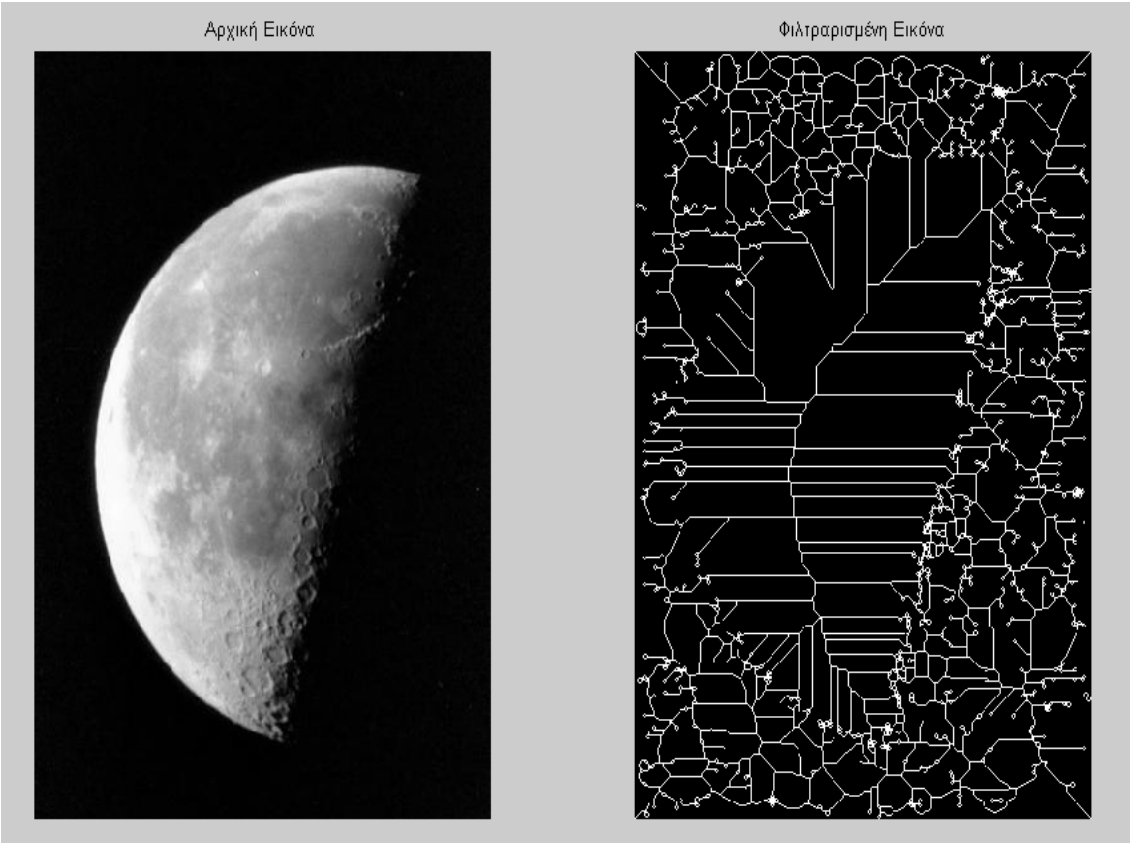
· Προσδιορισμός περιμέτρου

Διάγραμμα σκελετού

Για να μειώσουμε όλα τα αντικείμενα της εικόνας σε γραμμές, χωρίς να αλλάξουμε τη βασική δομή της εικόνας, χρησιμοποιόντας τη συνάρτηση **bwmorph**. Αυτή η διαδικασία είναι γνωστή ως διάγραμμα σκελετού.

Παράδειγμα: BW1=imread(‘moon2.jpg’);

```
BW2=bwmorph(BW1,’skel’,Inf);
subplot(1,2,1), imshow(BW1), title(‘Αρχική Εικόνα’);
subplot(1,2,2), imshow(BW2), title(‘Φιλτραρισμένη Εικόνα’);
```



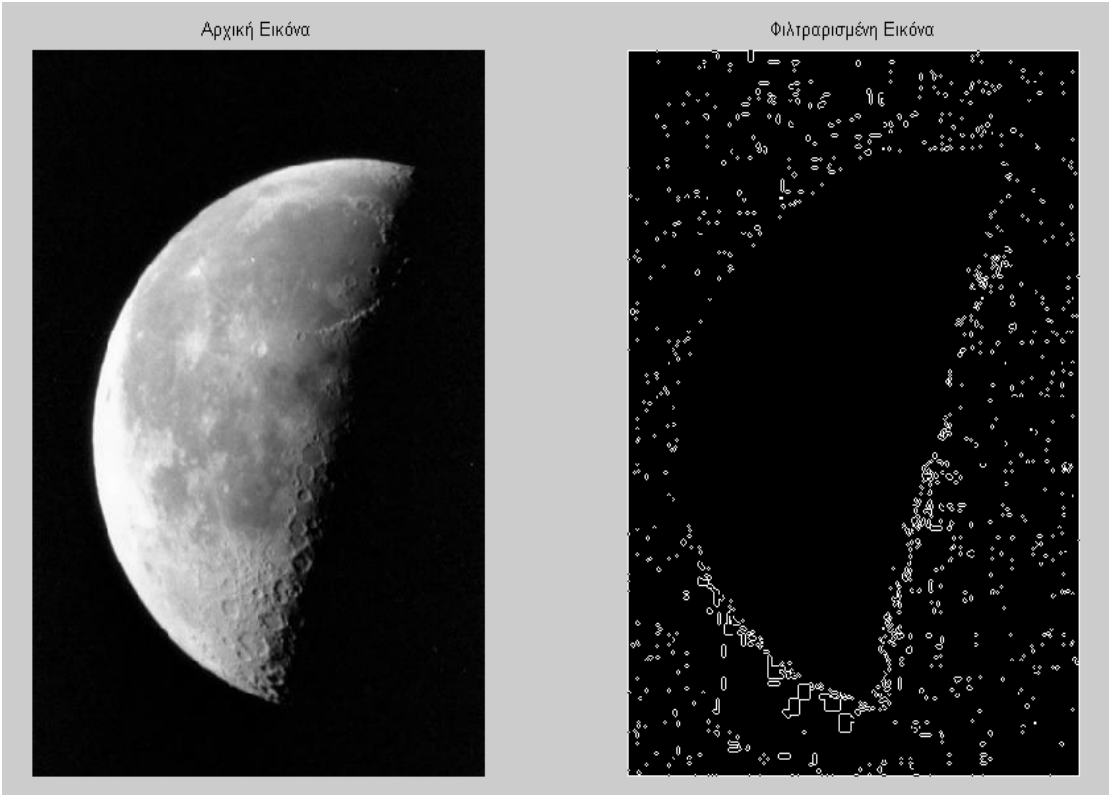
Προσδιορισμός περιμέτρου

Η συνάρτηση **bwperim** καθορίζει τα περιμετρικά pixel των αντικειμένων σε μία δυαδική εικόνα. Ένα pixel θεωρείτε περιμετρικό, εφόσον πληροί τα παραλάτω κριτίρια :

· Το pixel να είναι σε ισχύ.

· Ένα (ή περισσότερα) από τα pixel στη γειτονιά του να είναι μακριά.

Παράδειγμα: BW1=imread(‘moon2.jpg’);
BW2=bwperim(BW1);
subplot(1,2,1), imshow(BW1), title(‘Αρχική Εικόνα’);
subplot(1,2,2), imshow(BW2), title(‘Φιλτραρισμένη Εικόνα’);



Ανάλυση και ενίσχυση της εικόνας

Η παράγραφος αυτή περιγράφη τις συναρτήσης που υποστηρίζουν μια σειρά από στάνταρ λειτουργίες επεξεργασίας εικόνας για την ανάλυση και την ενίσχυση της εικόνας.

Βρίσκοντας πληροφορίες για την αξία των pixel

Αυτή η ενότητα περιγράφη πώς να πάρουμε πληροφορίες σχετικά με τις τιμές των δεδομένων που συνθέτουν μια εικόνα. Τα θέματα που καλύπτουν περιλαμβάνουν:

· Παίρνοντας πληροφορίες για τα pixel της εικόνας

· Παίρνοντας το προφίλ έντασης της εικόνας

· Εμφανίζοντας ένα περίγραμμα της γραφικής απεικόνισης των δεδομένων της εικόνας

· Δημιουργόντας ένα ιστόγραμμα εικόνας

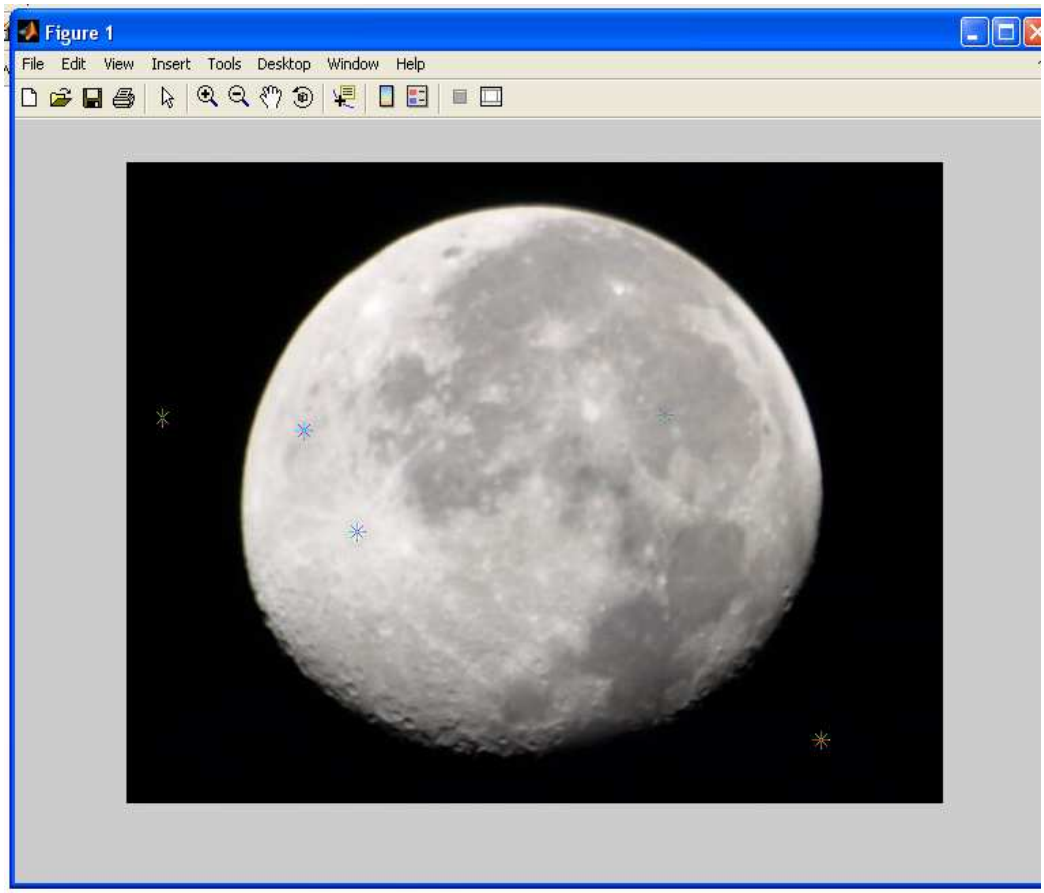
· Παίρνοντας συνοπτικά στατιστικά στοιχεία σχετικά με την εικόνα

Εμφανίζουμε μια εικόνα **imshow fegari.jpg**

Καλούμε την **impixel**. Όταν δεν εισάγουμε δεδομένα, η impixel συνδυάζει την εικόνα με τους τρέχουσες άξονες.

vals=impixel

Επιλέγουμε τα σημεία της εικόνας που θέλουμε να εξετάσουμε κάνοντας κλικ με το ποντίκη, η impixel τοποθετεί ένα αστέρι σε κάθε σημείο που επιλέγουμε.



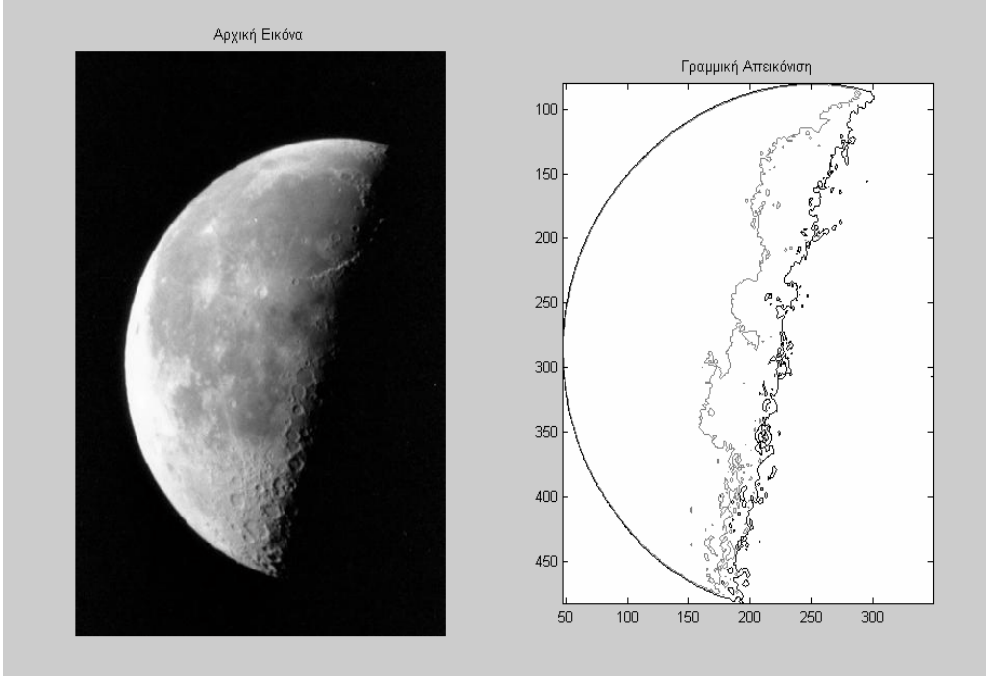
Όταν τελειώσουμε την επιλογή των σημείων, πατάμα **Enter**

```
vals =

  243   239   238

    1     1     3

  173   169   166

  205   201   198

    0     0     0
```

Εμφανίζοντας ένα περίγραμμα της γραφικής απεικόνισης των δεδομένων της εικόνας

Χρησιμοποιούμε την συνάρτηση **imcontour** για να εμφανίση ένα περίγραμμα από τη γραφική απεικόνιση δεδομένων μιας εικόνας σε αποχρώση του γκρι.

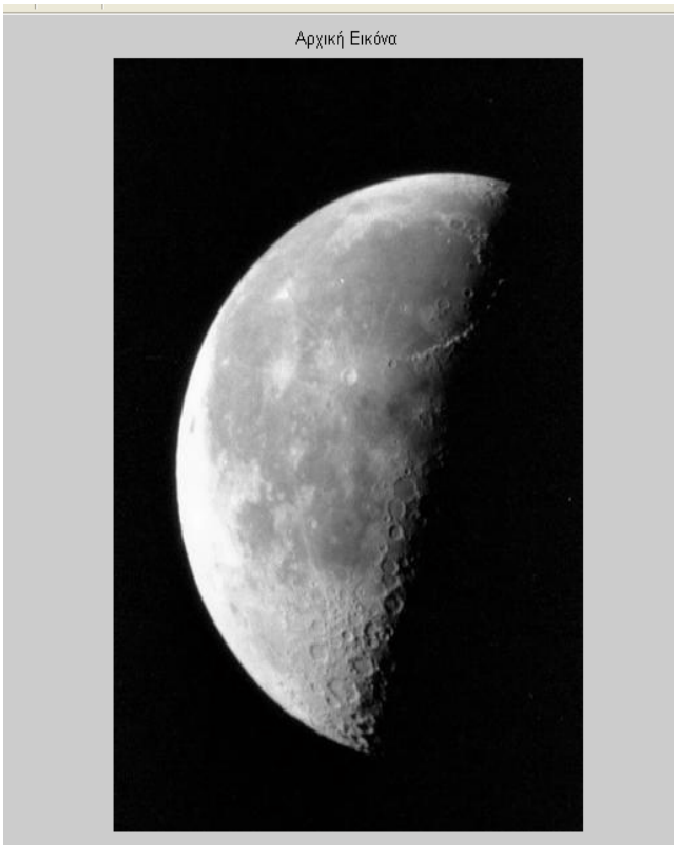
Παράδειγμα:
```
I=imread('moon2.jpg');
subplot(1,2,1), imshow(I), title('Αρχική Εικόνα');
subplot(1,2,2), imshow(I,3), title('Γραμμική Απεικόνιση');
```

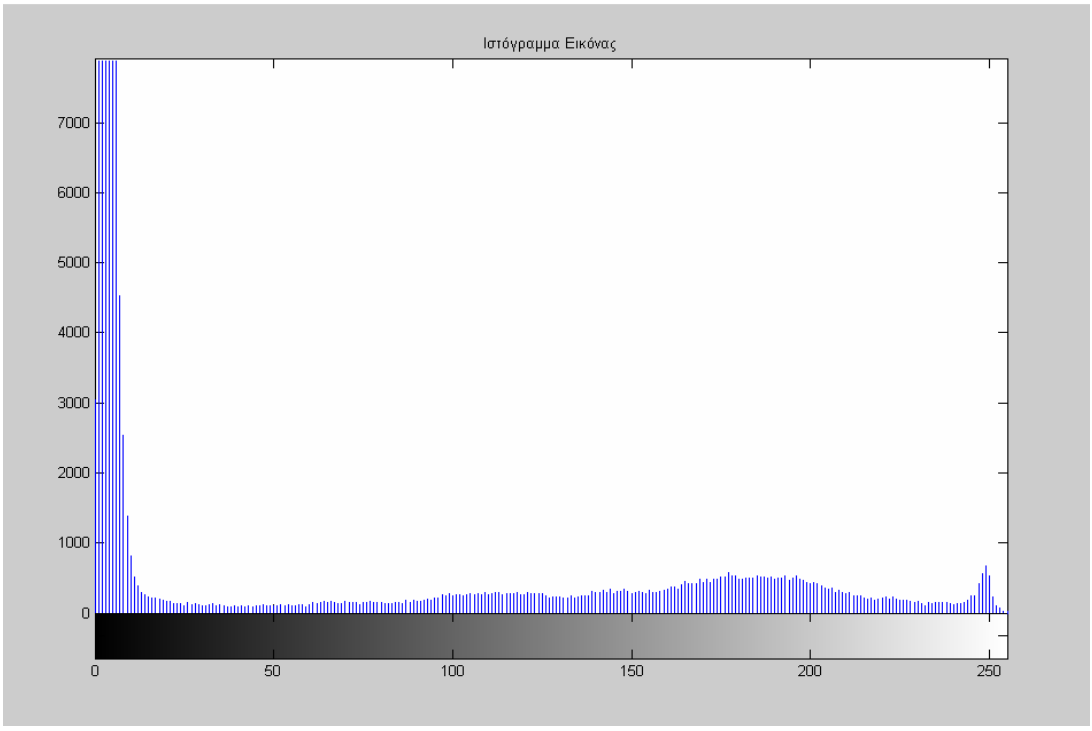


Δημιουργόντας ένα ιστόγραμμα εικόνας

Ένα ιστόγραμμα εικόνας είναι ένα διάγραμμαπου δείχνει την κατανομή των εντάσεων σε μία εικόνα αποχρώσεων γκρι. Για να δημιουργήσουμε ένα ιστόγραμμα εικόνας, χρησιμοποιούμε την συνάρτηση **imhist**. Αυτή η συνάρτηση δημιουργεί ένα ιστόγραμμα κάνοντας ‘ν’ ίσες αποστάσεις μεταξύ κάθε φάσμστος μετρήσεων. Τότε υπολογίζεται ο αριθμός των pixel σε κάθε περιοχή.

Παράδειγμα: I=imread(‘moon2.jpg’);
imshow(I), title(‘Αρχική Εικόνα’);
figure, imhist (I), title(‘Ιστόγραμμα Εικόνας’);


Αρχική Εικόνα


Ιστόγραμμα Εικόνας
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250

Αναλύοντας μία εικόνα

Η ενότητα αυτή περιγράφει τις τεχνικές ανάλυσης εικόνας που επιστρέφουν πληροφορίες σχετικά με τη δομή μιας εικόνας.

Ανίχνευση ακμών

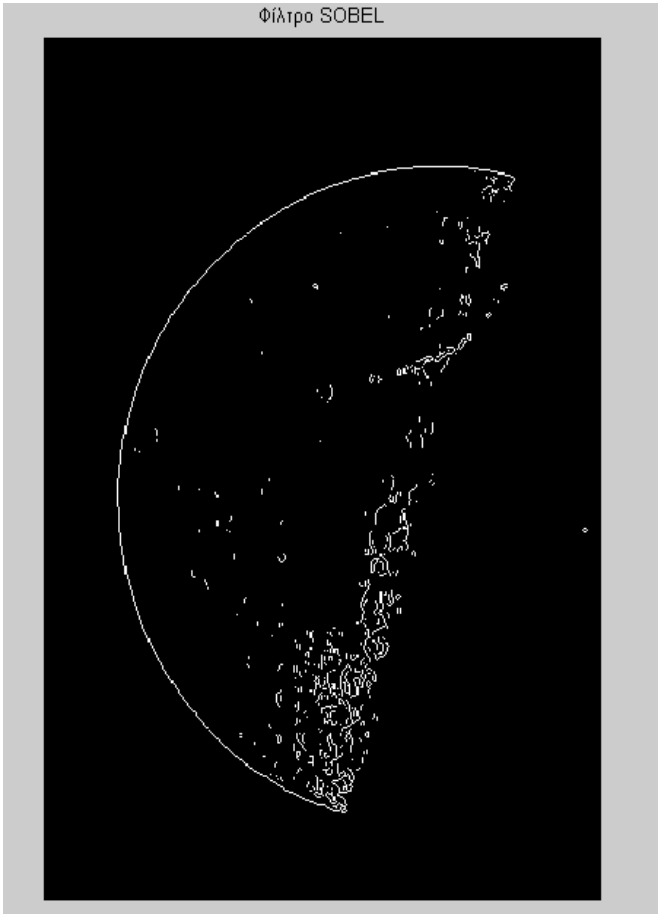
Σε μία εικόνα, η ακμή είναι η καμπύλη που ακολουθεί την πορεία της ταχείας αλλαγής της έντασης της εικόνας. Οι ακμές συχνά συνδέονται με τα όρια των αντικειμένων σε ένα μέρος της εικόνας.

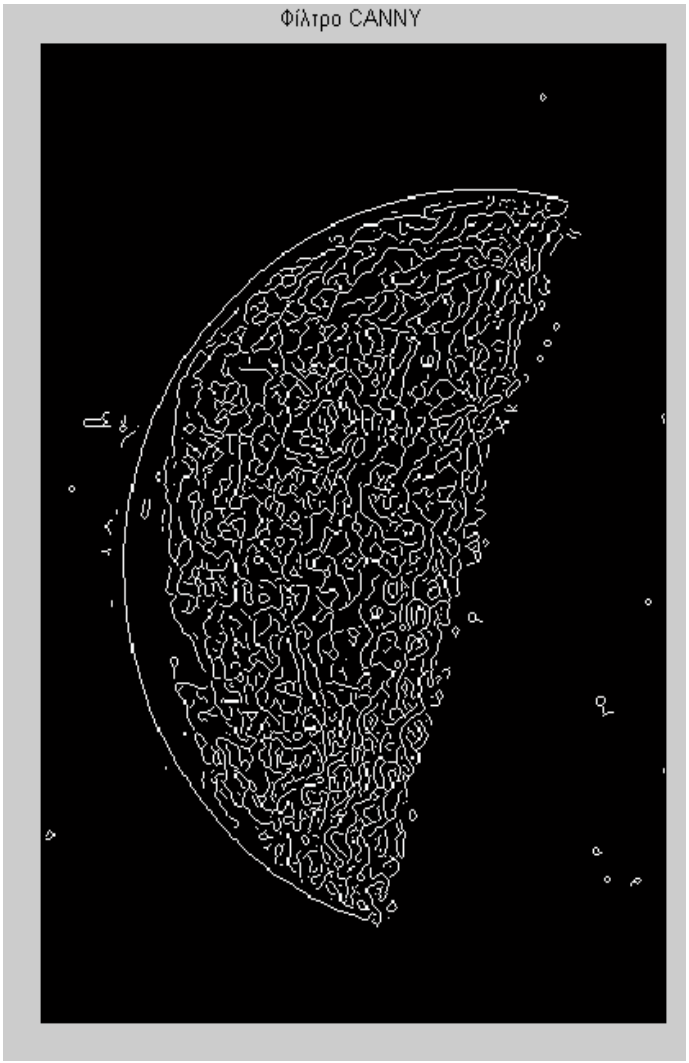
Η πιο ισχυρή μέθοδος ανίχνευσης ακμών που παρέχεται είναι η μέθοδος **Canny.** Η μέθοδος **Canny** διαφέρει από τις άλλες μεθόδους ανίχνευσης ακμων, στο ότι χρησιμοποιεί δύο διαφορετικά κατώτερα όρια (για εντοπισμό ισχυρών και ασθένων ακμών), και περιλαμβάνει τις αδύναμες ακμές στη παραγωγή μόνο εάν είναι συνδεδεμένες με ισχυρές ακμές.Αυτή η μέθοδος είναι συνεπώς λιγότερο πιθανό να ξεγελαστεί από το θόρυβο από τις άλλες μεθόδους, και το πιο πιθανό είναι να ανιχνεύση αλιθηνά αδύναμες ακμές.

Στο παρακάτω παράδειγμα χρησιμοποιούμε δύο συναρτήσεις ανίχνευσης ακμών για να της συγκρίνουμε, την ’**Sobel**’ και την **‘Canny’.** Θα εμφανίσουμε διαδοχικά την αρχική εικόνα και μετά φιλτραρισμένη με τις δύο συναρτήσης.

Παράδειγμα:
```
I=imread(‘moon2.jpg’);
imshow(I), title(‘Αρχική Εικόνα’);
BW1=edge(I,’sobel’);
BW2=edge(I,’canny’);
figure, imshow(BW1), title(‘Φίλτρο SOBEL’)
figure, imshow(BW2), title(‘Φίλτρο CANNY’)
```


Αρχική Εικόνα


Φίλτρο SOBEL



Εντοπισμός ορίων

Η εργαλειοθήκη του Matlab περιλαμβάνει δύο συναρτήσεις που μπορούμε να χρησιμοποιήσουμε για να βρούμε τα όρια των αντικειμένων σε δυαδικές εικόνες :

· **bwtraceboundary**

· **bwboundaries**

Η συνάρτηση **bwtraceboumdary** επιστρέφει την γραμμή και την στήλη των συντεταγμένων όλων των pixel στα όρια ενός αντικειμένου σε μία εικόνα. Πρέπει να καθορίσουμε τη θέση των ορίων pixel του αντικειμένου ως το σημείο εκκίνησης

Η συνάρτιση **bwboundaries** επιστρέφει τη γραμμή και τη στήλη των συντεταγμένων των ορίων pixel όλων των αντικειμένων σε μία εικόνα.

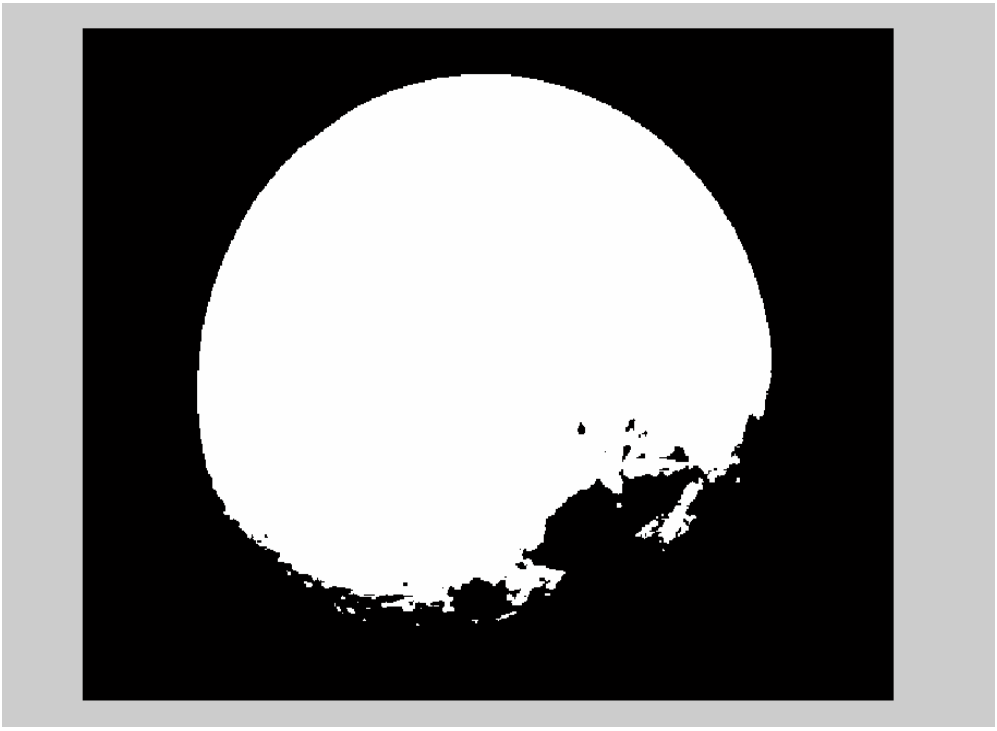
Παράδειγμα:

Διαβάζουμε την εικόνα και την εμφανίζουμε.

```
I=imread('fegari.jpg');
imshow(I)
```

Μετατρέπουμε την εικόνα σε δυαδική εικόνα.Οι συναρτήσεις **bwtraceboundary** και **bwboundaries** εργάζονται μόνο με δυαδικές εικόνες.

```
BW=im2bw(I);
figure, imshow(BW)
```



Καθορίζουμε τη γραμμή και τη στήλη συντεταγμένων ενός pixel στα όρια ενός αντικειμένου που θέλουμε να εντοπίσουμε. Η συνάρτηση **bwboundary** χρησιμοποιεί αυτό το σημείο ως θέση εκκίνησης για το όριο ανίχνευσης.

```
dim=size(BW)
col=round(dim(2)/2)-90
row=min(find(BW(:,col)))
```

```
dim =

          450   600


col =

                    210

row =

                     51
```

Καλώντας την συνάρτηση **bwtraceboundary** για να εντοπίσουμε το όριο από το συγκεκριμένοσημείο. Απαιτείται να δώσουμε κάποια δεδομένα, να καθορίσουμε μια δυαδική εικόνα, τις συντεταγμένες της γραμμής και στήλης του σημείου εκκίνησης, καθώς και την κατεύθυνση το πρώτο βήμα.

```
boundary=bwtraceboundary(BW,[row,col],'N');
```

Εμφανίζουμε την αρχική εικόνα και χρησιμοποιούμε της συντεταγμένες που επιστρέφονται από τη συνάρτηση **bwtraceboundary** για να σχεδιάσουμε τα όρια του αντικειμένου.

```
imshow(I)
hold on;
plot(boundary(:,2),boundary(:,1),'g','LineWidth',3);
```

Για να εντοπίσουμε τα όρια του συνόλου των αντικειμένων της εικόνας, χρησιμοποιούμε την συνάρτηση **bwboundaries.**

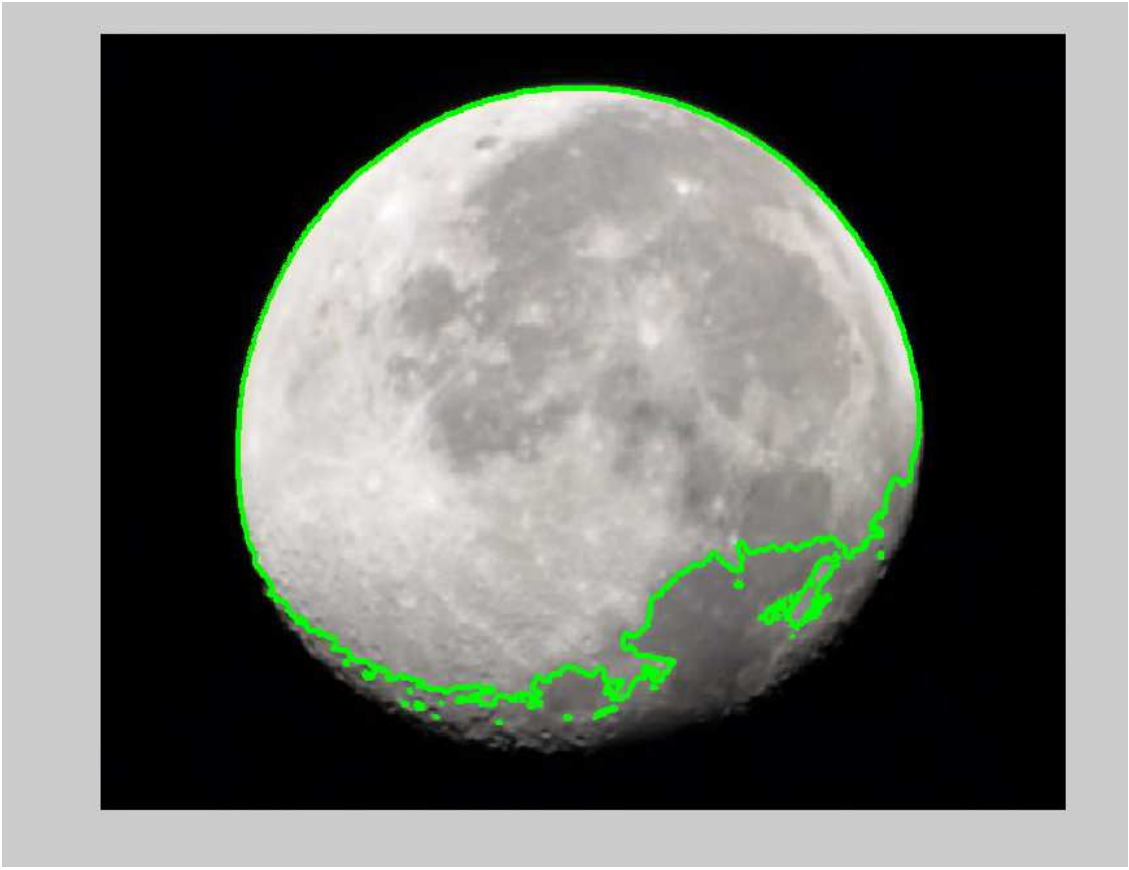
```
BW_filled=imfill(BW,'holes');
boundaries=bwboundaries(BW_filled)
```

```
boundaries =

  [1445x2 double]
  [   7x2 double]
  [  13x2 double
  [   3x2 double]
  [   7x2 double]
  [   5x2 double]
  [   2x2 double]
```

```
[  14x2 double]
[   2x2 double]
[   5x2 double]
[   5x2 double]
[   4x2 double]
[   2x2 double]
[   7x2 double]
[   9x2 double]
[  30x2 double]
[   9x2 double]
[   9x2 double]
[  47x2 double]
[   3x2 double]
[ 158x2 double]
[   2x2 double]
[   5x2 double]
[   4x2 double]
[   8x2 double]
```

Για την γραφική απεικόνιση όλων των ορίων των αντικειμένων της αρχικής εικόνας χρησιμοποιούμε τις συντεταγμένες που επιστρέφει η συνάρτηση **bwboundaries.**

```
for k=1:25
        b=boundaries{k};
        plot(b(:,2),b(:,1),'g','LineWidth',3);
end
```



Ανίχνευση γραμμών χρησιμοποιώντας το μετασχηματισμό Hough

Η Image Processing toolbox περιλαμβάνει συναρτήσεις που υποστηρίζουν των μετασχηματισμο Hough :

· hough

· houghpeaks

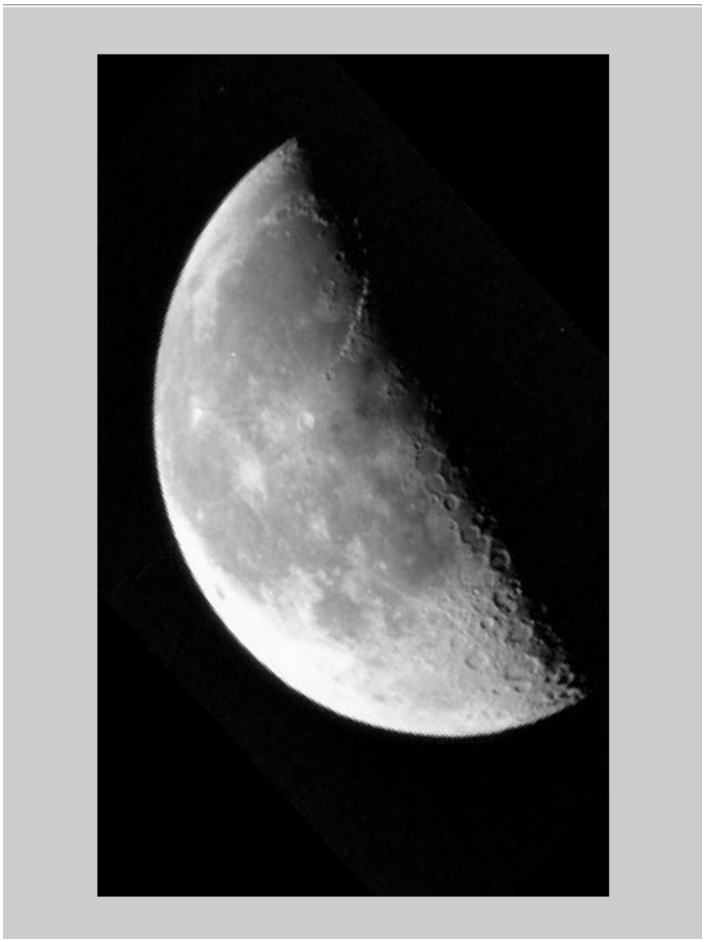
· houghlines

Διαβάζουμε και εμφανίζουμε την εικόνα

```
I=imread('moon2');

figure, imshow(I)
```

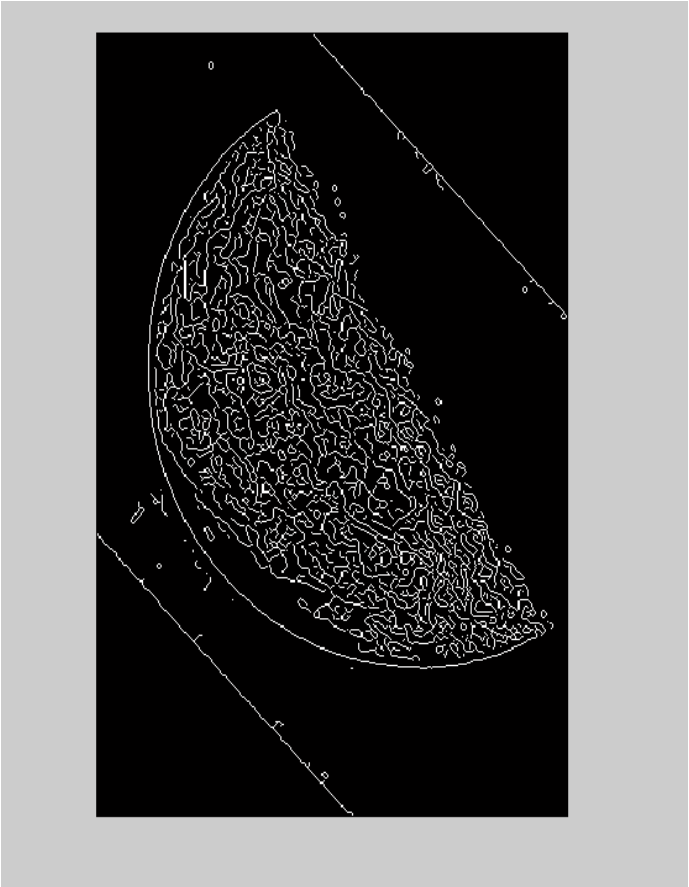
Περιστρέφουμε και κροπάρουμε την εικόνα.

```
rotI=imrotate(I,45,'crop');

figure, imshow(rotI)
```



Βρίσκουμε τις ακμές της εικόνας.

```
BW=edge(rotI,'canny');

figure, imshow(BW)
```

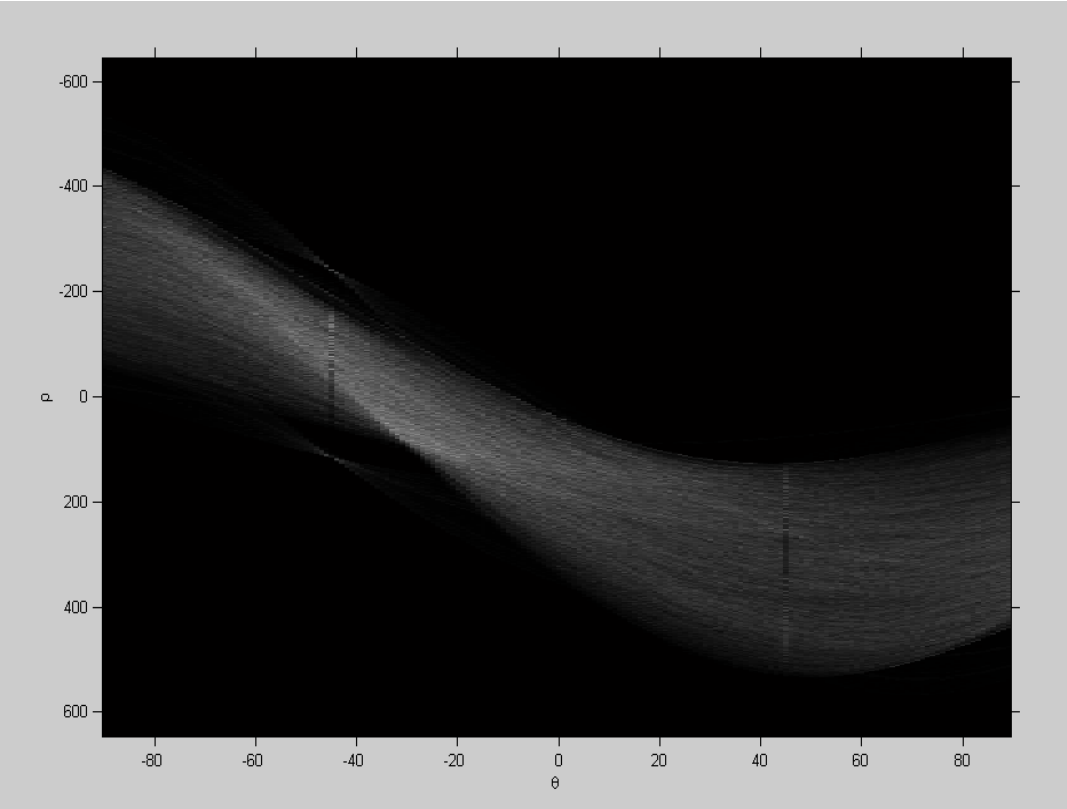


Υπολογίζουμε το μετασχηματισμό Hough της εικόνας χρησιμοποιώντας τη συνάρτηση **hough.**

```
[H,theta,rho]=hough(BW);
```

Εμφανίζουμε τον μετασχηματισμό.

```
imshow(H,[], ‘XData’,theta,’YData’,rho,…
        ’InitialMagnification’,’fit’);
xlabel(‘\theta’),ylabel(‘\rho’);
axis on, axis normal, hold on;
```

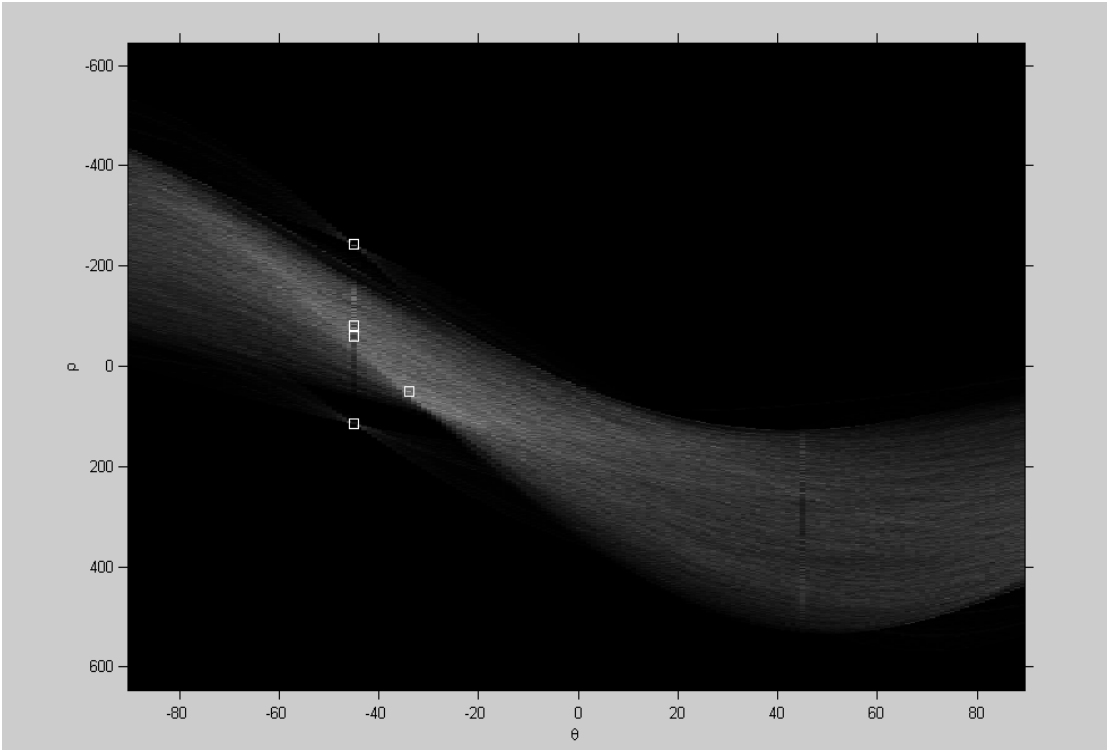


Για να βρούμε τις κορυφές του μετασχηματισμου Hough, χρησιμοποιούμε τη συνάρτιση **houghpeaks.**

```
P=houghpeaks(H,5,’threshold’,ceil(0.3*max(H(:))));
```

Γραφική απεικόνηση των κορυφών

```
x=theta(P(:,2));
y=rho(P(:,1));
plot(x,y,’s’,’color’,’white’);
```

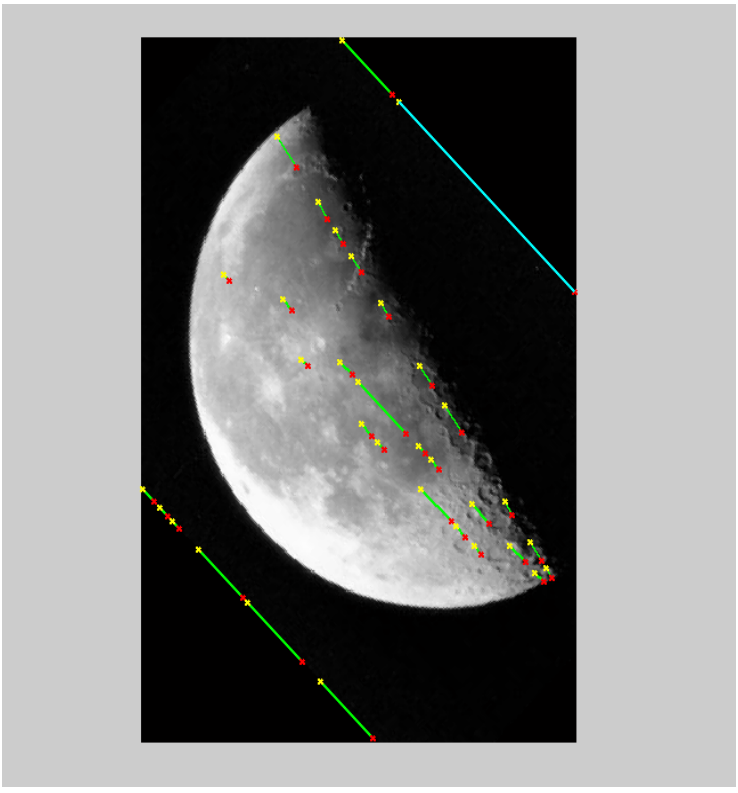


Για να βρούμε τις γραμμές της εικόνας.

```
lines=houghlines(BW,theta,rho,P,'FillGap',5,'MinLength',7);
```

Δημιουργούμε μία γραφική απεικόνιση που σκιαγραφεί τις γραμμές πάνω στην αρχική εικόνα.

```
figure, imshow(rotI), hold on
max_len=0;
for k=1:length(lines)
    xy=[lines(k).point1;lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2,'color','green');
%Γραφική απεικόνιση αρχή και τέλους γραμμών
    plot(xy(1,1),xy(1,2),'x','LineWidth',2,'color','yellow');
    plot(xy(2,1),xy(2,2),'x','LineWidth',2,'color','red');
%Καθορίζουμε τα σημεία τέλους του μεγαλύτερου τμήματος γραμμής
    len=norm(lines(k).point1-lines(k).point2);
    if (len>max_len)
        max_len=len;
        xy_long=xy;
    end
end
%Για να τονίσει τη μεγαλύτερη γραμμή
plot(xy_long(:,1),xy_long(:,2),'Linewidth',2,'Color','cyan');
```



Αναλύοντας την υφή μιας εικόνας

Το κουτί εργαλείων εμπεριέχει ένα σύνολο συναρτήσεων που μπορούν να χρησιμοποιηθούν για ανάλυση υφής. Η ανάλυση υφής χρησιμοποιήται σε μία ποικιλία εφερμογών, συμπεριλαμβανομένων κατανόησης από μακριά, αυτοματοποιημένη επιθεώρηση και ιατρική επιθεώρηση εικόνας. Η ανάλυση υφής μπορεί να χρησιμοποιηθεί για να βρεθούν τα όρια της υφής που αποτελείται τμηματοποίηση της. Η ανάλυση υφής μπορεί να είναι χρήσιμη όταν τα αντκείμενα σε μια εικόνα είναι πιο συγκεκριμένα από την υφή τους παρά από την συχνότητα, και οι παραδοσιακές τεχνικές περάσματος δεν μπορούν να χρησιμοποιηθούν αποτελεσματικά.

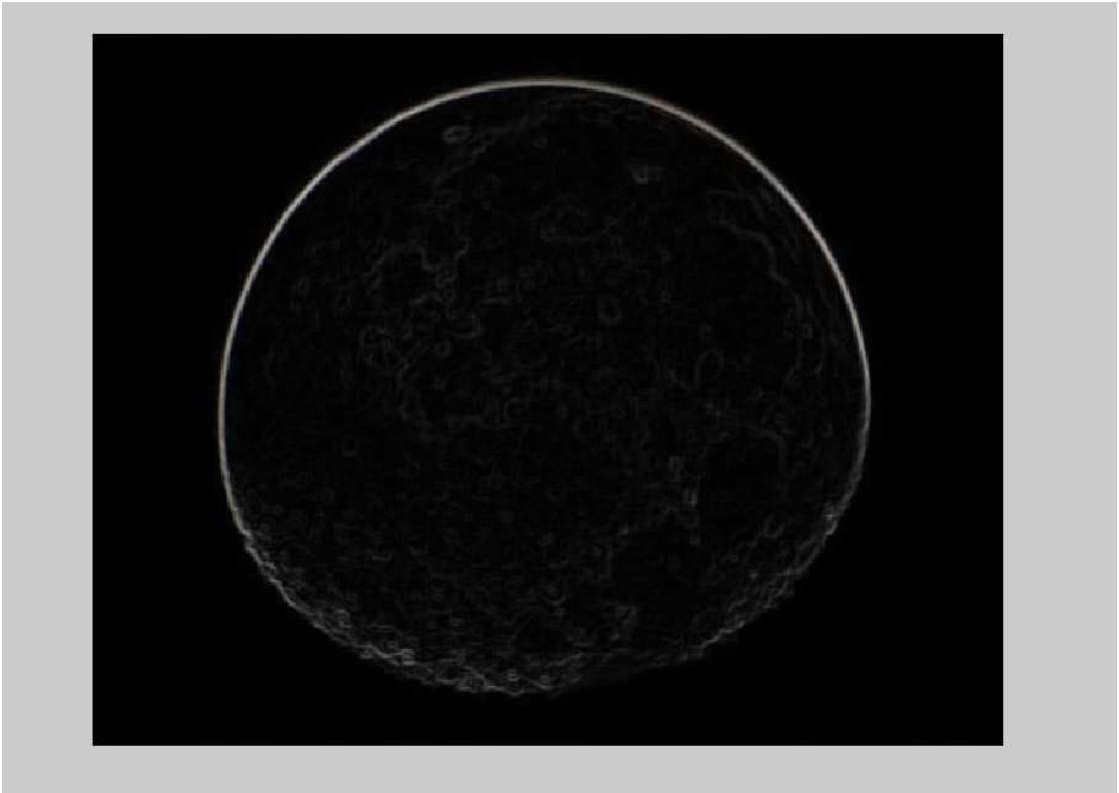
Λειτουργίες υφής

Διαβάζουμε και απεικονίζουμε την εικόνα

```
I=imread('fegari.jpg');
imshow(I)
```

Φιλτράρουμε την εικόνα με την συνάρτηση **rangefilt** και στη συνέχεια απεικονίζουμε τα αποτελέσματα.

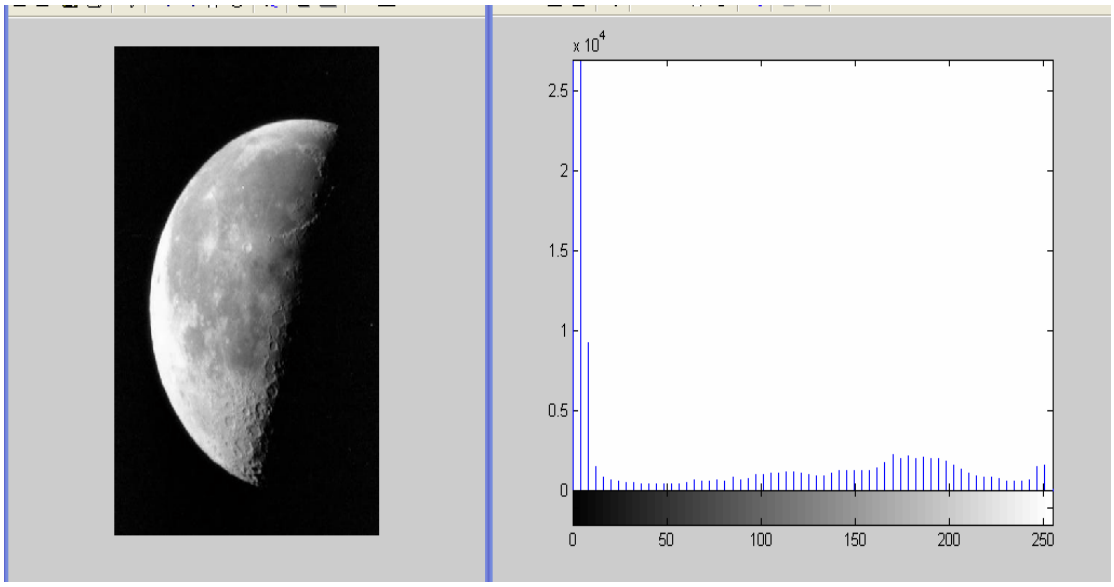
```
K=rangefilt(I);
figure, imshow(K)
```



Τροποποίηση συχνότητας

Η τροποποίηση συχνότητας είναι μια τεχνική ενίσχυσης εικόνας που χαρτογραφεί τις αξίες συχνότητας μιας εικόνας σε ένα νέο εύρος.

Παράδειγμα: I=imread('moon2.jpg');
imshow(I), title('Αρχική Εικόνα');
figure, imhist(I,64), title('Τροποποίηση Συχνότητας')



Τροποποιώντας τις αξίες συχνότητας σε συγκεκριμένο εύρος

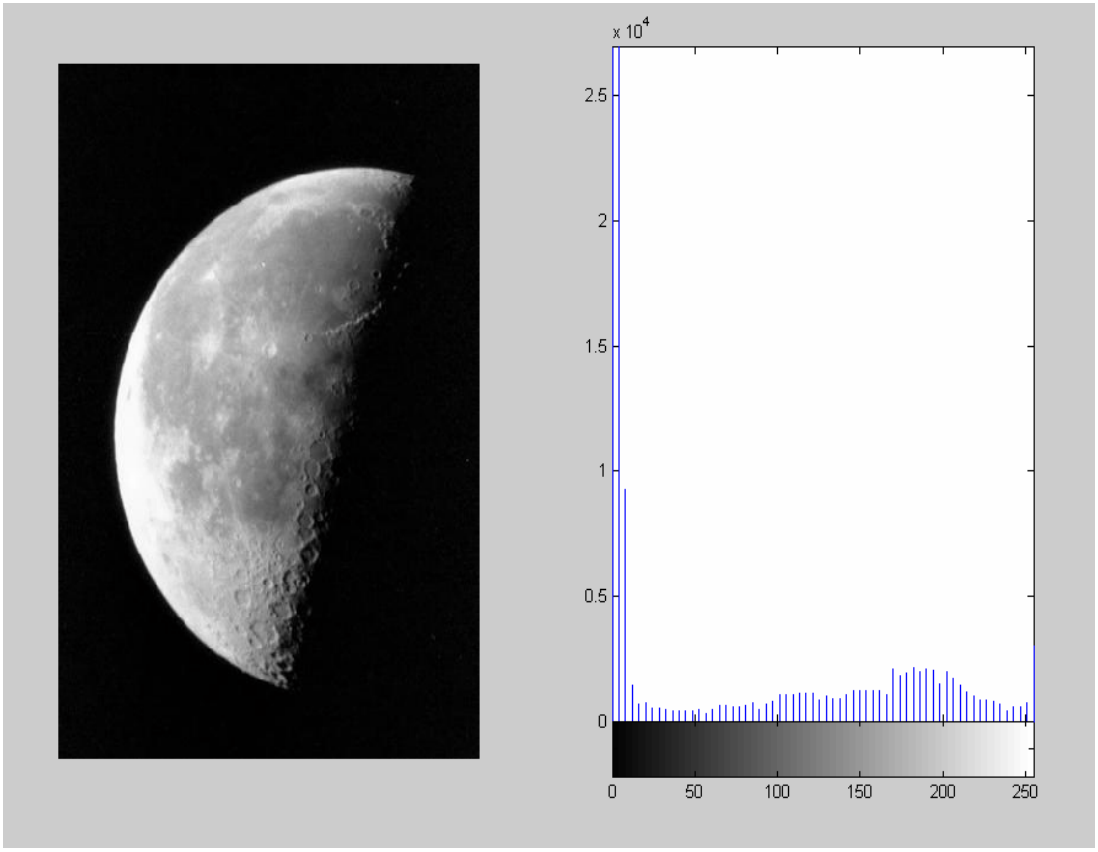
Μπορούμε να τροποποιήσουμε τις αξίες συχνότητας σε μια εικόνα χρησιμοποιώντας τη συνάρτηση **imadjust**, που καθορίζει το εύρος των αξιών συχνότητας στην εξερχόμενη εικόνα

Παράδειγμα: I=imread('moon2.jpg');

J=imadjust(I);

subplot(1,2,1),imshow(I)

subplot(1,2,2),imhist(J,64)

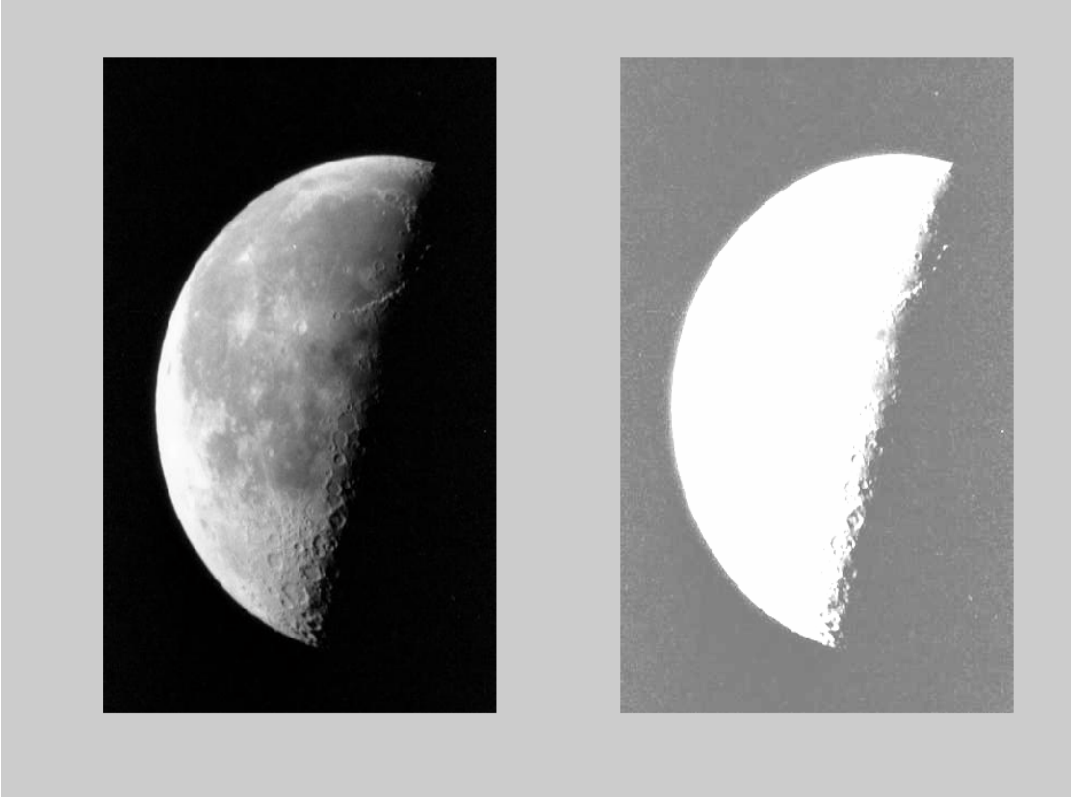


Καθορίζοντας τα όρια τροποποίησης

Μπορούμε να καθορίσουμε προαιρετικά το εύρος των εισερχόμενων και εξερχόμενων αξιών χρησιμοποιώντας τη συνάρτηση **imadjust**. Καθορίζουμε αυτά τα δύο εύροι σε δύο ποσότητες που περνάμε στο **imadjust** σαν δεδομένα. Ο πρώτος καθορίζει τις αξίες χαμηλής και υψηλής συχνότητας που θέλουμε να χαρτογραφήσουμε, και ο δεύτερος καθορίζει την κλίμακα κατά την οποία θέλουμε να χαρτογραφήσουμε.

Παράδειγμα: I=imread('moon2.jpg');

```
J=imadjust(I,[0 0.2],[0.5 1]);
subplot(1,2,1),imshow(I)
subplot(1,2,2),imshow(J)
```



Εξίσωση ιστογράμματος

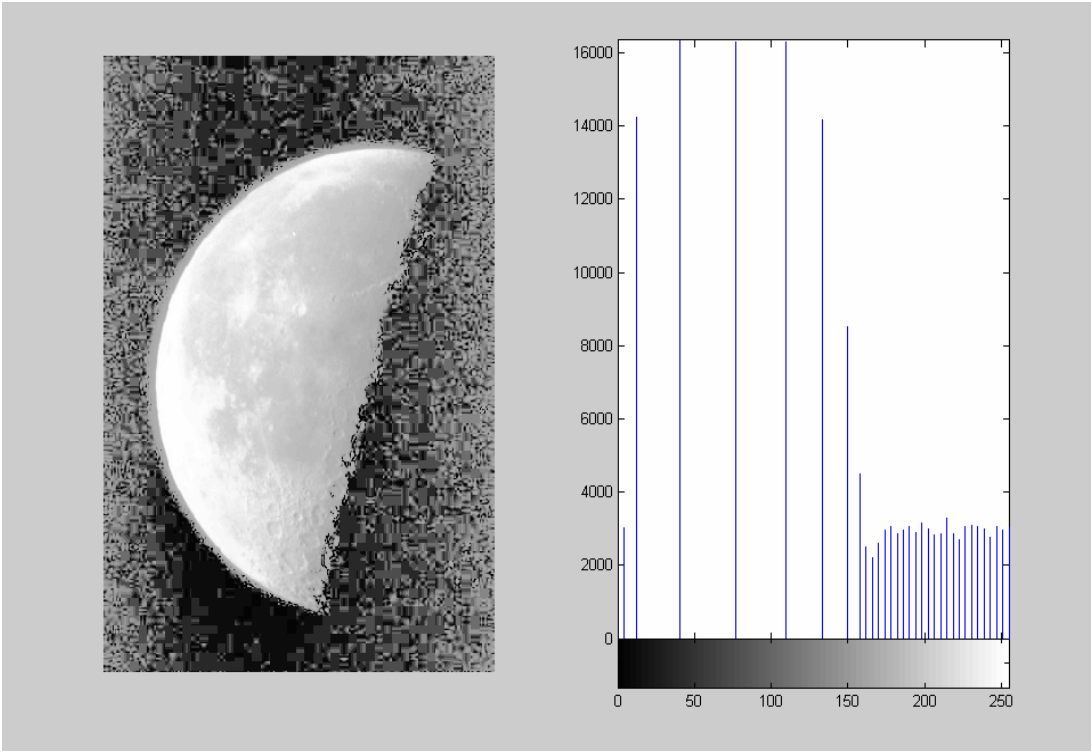
Η διαδικασία τροποποίησης αξιών συχνότητας μπορεί να γίνει αυτόματα με τη συνάρτηση **histeq**. Η **histeq** διεξάγει την εξίσωση ιστογράμματος, η οποία περιλαμβάνει μετατροπή αξιών συχνότητας έτσι ώστε το ιστόγραμμα της εξερχόμενης εικόνας να ταιρίαζει περίπου με ένα προκαθορισμένο ιστόγραμμα.

Παράδειγμα: I=imread('moon2.jpg');

J=histeq(I);

subplot(1,2,1),imshow(J)

subplot(1,2,2),imhist(J,64)



Αφαίρεση θορύβων

Οι ψηφιακές εικόνες έχουν τάση σε μία μεγάλη ποικιλία τύπων θορύβου. Ο θόρυβος είναι αποτέλεσμα λαθών στη διαδικασία απόκτησης εικόνας που οδηγεί σε ποσότητα pixel που δεν αντανακλούν τις σωστές συχνότητες της πραγματικής σκηνής. Υπάρχουν πολλοί τρόποι που μπορεί να εισαχθεί σε μία εικόνα, που εξαρτώνται από το πώς η εικόνα δημιουργείται.

Φιλτράρισμα του μέσου

Το φιλτράρισμα του μέσου είναι παρόμοιο με το να χρησιμοποιούμε ένα διάμεσο φίλτρο, στο οποίο κάθε εξερχόμενο pixel ρυθμίζεται σε ένα μέσο αξιών pixel στη γειτονιά των αντίστοιχων pixel.

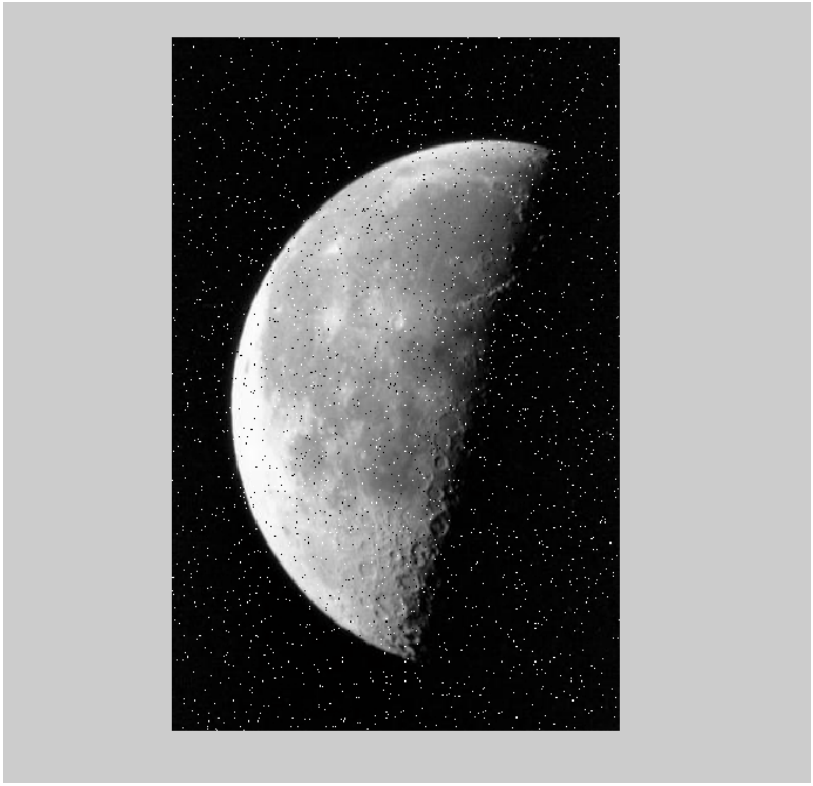
Παράδειγμα:

Διαβάζουμε και απεικονίζουμε την εικόνα

```
I=imread('moon2.jpg');
imshow(I)
```

Προσθέτουμε θόρυβο στην εικόνα.

```
J=imnoise(I,'salt & pepper',0.02);
figure,imshow(J)
```



Φιλτράρουμε την εικόνα, με ένα φίλτρο για να βγάλουμε τον θόρυβο και απεικονίζουμε παρακάτω τα αποτελέσματα.

```
K=filter2(fspecial('average',3),J)/200;
figure,imshow(K)
```

Θα χρησιμοποιήσουμε τώρα φίλτρο του μέσου για να φιλτράρουμε την εικόναμε τον θόρυβο και θα απεικονίσουμε παρακάτω τα αποτελέσματα.

Όπως θα παρατηρήσουμε το **medfilt2** κάνει καλύτερη δουλεία στην αφαίρεση θορύβου.

```
L=medfilt2(J,[3 3]);
subplot(1,2,1),imshow(K)
subplot(1,2,2),imshow(L)
```

Επεξεργασία βασισμένη σε ROI

Αυτό το κεφάλαιο περιγράφει πώς να ορίσουμε έναν τομέα ενδιαφέροντος (ROI) και να διεξάγουμε επεξεργασία στον τομέα ενδιαφλεροντος που ορίζουμε.

Ορίζοντας ένα τομέα ενδιαφέροντος (ROI)

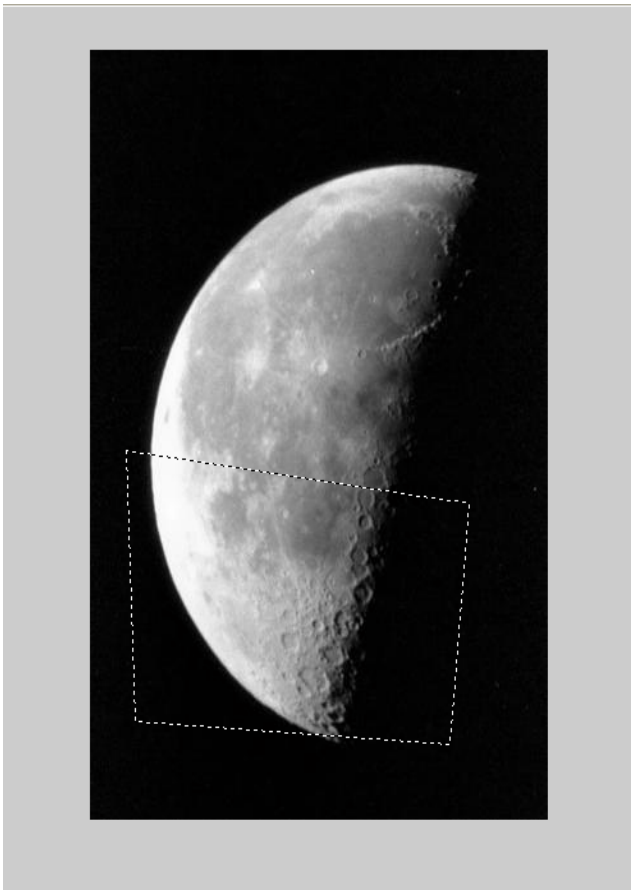
Ένας τομέας ενδιαφέροντος είναι ένα μέρος μιας εικόνας που θέλουμε να φιλτράρουμε ή να διεξάγουμε κάποια άλλη επεξεργασία σε αυτό. Ορίζουμε ένα ROI δημιουργόντας μία δυαδική μάσκα, η οποία είναι μία δυαδική εικόνα που είναι ίδιου μεγέθους με την εικόνα που θέλουμε να επεξεργαστούμε με pixel που ορίζουν τον ROI ρυθμισμένο στο 1 και όλα τα υπόλοιπα pixel ρυθμισμένα στο 0.

Επιλέγοντας διαδραστικά ένα πολυγωνικό ROI

Χρησιμοποιούμε την συνάρτηση **roipoly** για να καθορίσουμε διαδραστικά ένα πολύγωνο ROI σε μία εικόνα. Για να το πετύχουμε αυτό εμφανίζουμε την εικόνα χρησιμοποιώντας την **imshow** και μετά καλούμε την συνάρτηση **roipoly** χωρίς να δώσουμε εισερχόμενα δεδομένα. Όταν μετατοπίζουμε τον δείκτη πάνω στην εικόνα που εμφανίζεται πάνω στους τρέχοντες άξονες, ο δείκτης αλλάζει σε σταυρωτό σχήμα. Μπορούμε τότε να καθορίσουμε τις καθετότητες του πολυγώνου με το να κάνουμε κλικ σε διάφορα σημεία της εικόνας με το ποντίκη. Όταν έχουμε τελείωσει με την επιλογή καθετοτήτων, πατάμε **enter** το **roipoly** επιστρέφει μια δυαδική εικόνα ίδιου μεγέθους με την εισερχόμενη εικόνα, περιέχοντας 1 μέσα στο καθορισμένο πολύγωνο, και 0 σε όλα τα άλλα σημεία.

Παράδειγμα:

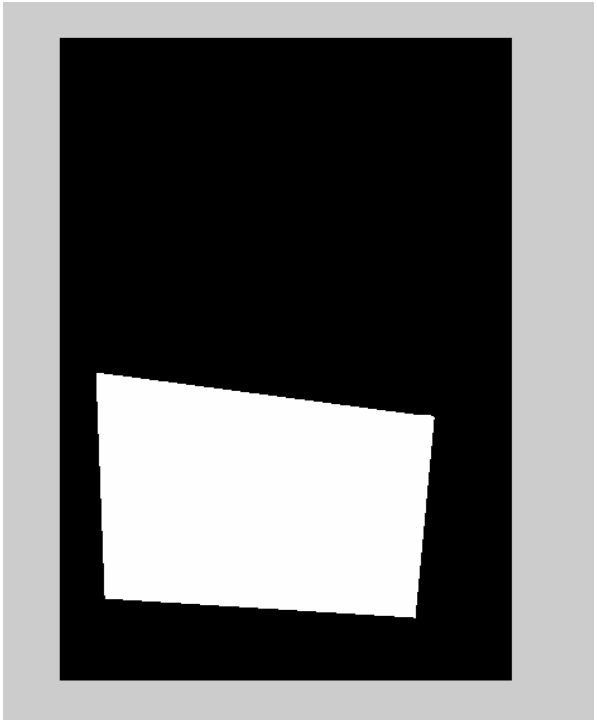
```
I=imread('moon2.jpg');
imshow(I)
BW=roipoly;
```



Μετά πατάμε Enter και γράφουμε το παρακάτω

```
imshow(BW)
```

και μας εμφανίζει την δυαδική εικόνα



Φιλτράροντας ένα ROI

Φιλτράρισμα ενός τομέα ενδιαφέροντος (ROI) είναι η διαδικασία εφαρμογής ενός φίλτρου σε ένα τμήμα μιας εικόνας, όπου μία δυαδική μάσκα ορίζει τον τομέα.

Φιλτράροντας έναν τομέα σε μία εικόνα

Το παρακάτω παράδειγμα χρησιμοποιεί μία μάσκα φιλτραρίσματος για να αυξήση την αντίθεση ενός συγκεκριμένου τομέα μίας εικόνας.

Παράδειγμα:

Διαβάζουμε την εικόνα

```
I=imread(‘moon2.jpg’);
```

Δημιουργούμε τη μάσκα

```
imshow(I);
BW=roipoly;
```

Δημιουργούμε το φίλτρο

```
h=fspecial(‘unsharp’);
```

Καλούμε τη συνάρτηση **roifilt2,** καθορίζοντας την εικόνα που πρόκειται να φιλτράρουμε, τη μάσκα, και το φίλτρο

```
I2=roifilt(h,I,BW);
subplot(1,2,1),imshow(I),title('Αρχική εικόνα')
subplot(1,2,2),imshow(I2),title('Φιλτραρισμένη εικόνα')
```

Ξεθαμπώνοντας με σταθεροποιημένο φίλτρο

Χρησιμοποιούμε τη συνάρτηση **deconvreg** για να ξεθαμπώσουμε μία εικόνα χρησιμοποιώντας ένα σταθεροποιήμένο φίλτρο. Ένα τέτοιο φίλτρο μπορεί να χρησιμοποιηθεί αποτελεσματικά όταν γνωρίζουμε περιορισμένες πληροφορίες για τον επιπρόσθετο θόρυβο.

Παράδειγμα:

Διαβάζουμε μία εικόνα

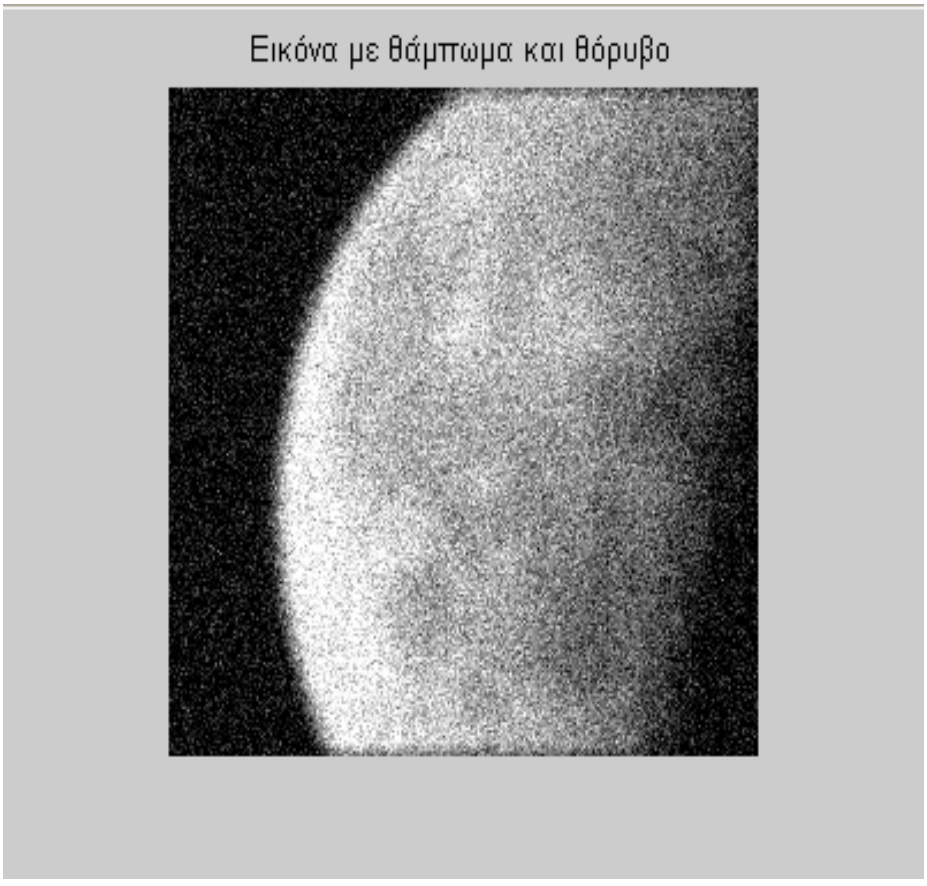
```
I=imread('moon2.jpg');
I=I(125+[1:256],1:256,:);
figure, imshow(I), title('Αρχική Εικόνα')
```

Δημιουργούμε το PSF.

```
PSF=fspecial(‘gausian’,11,5);
```

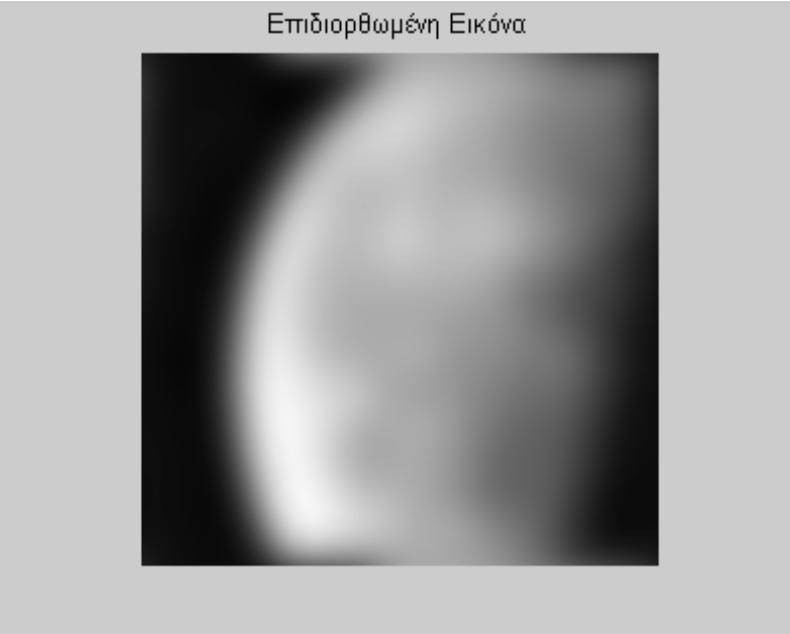
Δημιουργούμε μία εξομοιωμένη θαμπάδα στην εικόνα και προσθέτουμε θόρυβο.

```
Blurred=imfilter(I,PSF,’conv’);
V=.02;
BlurredNoisy=imnoise(Blurred,’gausian’,0,V);
figure, imshow(BlurresNoisy), title(‘Εικόνα με θάμπωμα και θόρυβο’)
```



Χρησιμοποιούμε τη συνάρτηση **deconvreg** για να ξεθαμπώσουμε την εικόνα, καθορίζοντας το PSF που χρησιμοποιήθηκε για να δημιουργήσουμε την θαμπάδακαι την δύναμη του θορύβου το NP.

```
NP=V*prod(size(I));
[reg1 LARGA]=deconvreg(BlurredNoisy,PSF,NP);
figure, imshow(reg1), title('Επιδιορθωμένη Εικόνα')
```



Ξεθαμπώνοντας με τον αλγόριθμο τυφλής αποκατάστασης

Η συνάρτηση **deconvblind**, ακριβώς όπως και η συνάρτηση **deconvlucy**, εφαρμόζει διάφορες προσαρμογές στον αυθεντικό αλγόριθμο μέγιστης πιθανότητας των Lucy-Richardson που αντιμετοπίζει πολύπλοκες διαδικασίες αποκατάστασης εικόνας.

Χρησιμοποιώντας τη συνάρτηση deconvblind για να ξεθαμπόσουμε μία εικόνα

Για να απεικονίσουμε τυφλή αποκατάσταση, το παρακάτω παράδειγμα δημιουργεί μία εξομοιωμένη θαμπή εικόνα και ύστερα χρησιμοποιεί την συνάρτηση **deconvblind** για να την ξεθαμπώση.

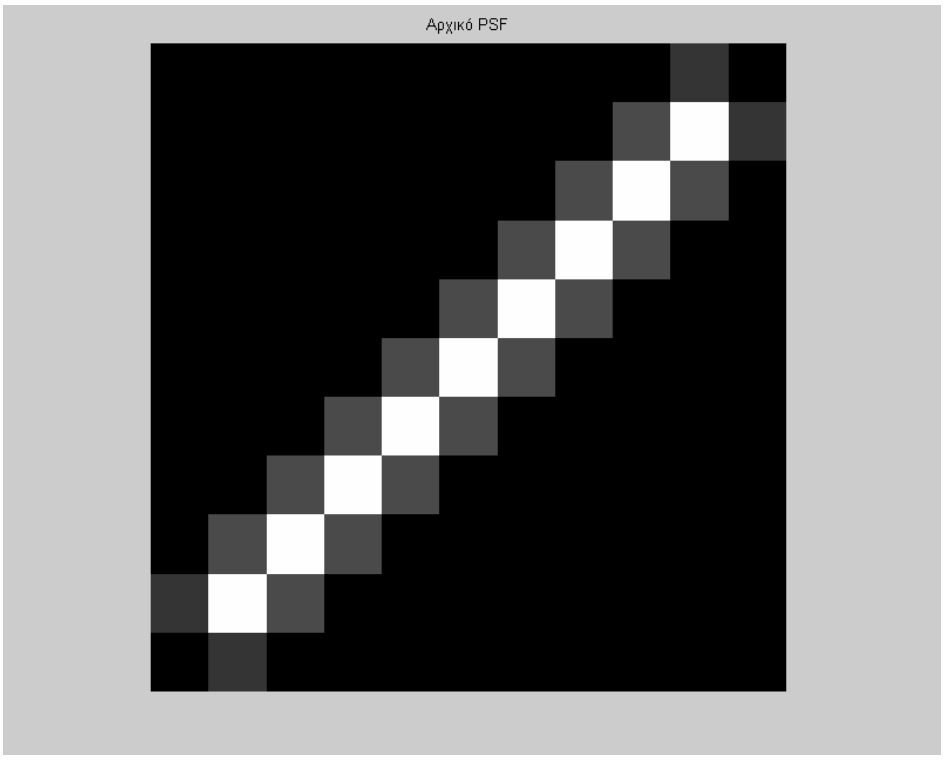
Παράδειγμα:

Διαβάζουμε μία εικόνα

```
I=imread('fegari.jpg');
figure, imshow(I), title('Αρχική Εικόνα')
```

Δημιουργούμε το PSF

```
PSF=fspecial('motion',13,45);
figure, imshow(PSF,[],'notruesize'),title('Αρχικό PSF')
```



Δημιουργούμε μία εξομοιωμένη θαμπάδα στην εικόνα

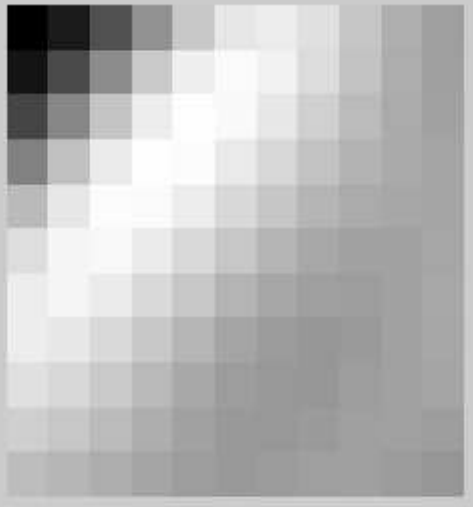
```
Blurred=imfilter(I,PSF,'circ','conv');
figure, imshow(Blurred) ,title('Θαμπωμένη Εικόνα')
```

Ξεθάμπωμα της εικόνας, κάνοντας μία αρχική πρόβλεψει του μεγέθους του PSF.

```
INITPSF=ones(size(PSF));
[J P]=deconvblind(Blurred,INITPSF,30);
subplot(2,1,1),imshow(J),title('Ξεθαμπωμένη Εικόνα')
subplot(2,1,2),imshow(P,[],'notruesize'),title('Απόδωσει PSF')
```

Ξεθαμπωμένη Εικόνα

Απόδωσει PSF



Διαδικασίες κύλισης γειτονιάς

Μία διεργασία κύλισης γειτονιάς είναι μία διεργασία που διεξάγει ένα pixel τη φορά, με την αξία οποιουδήποτε δωσμένου pixel στην εξερχόμενη εικόνα να καθορίζεται από την εφαρμογή ενός αλγορίθμου στις αξίες της αντίστοιχης γειτονίας εισερχομένων pixel. Μία γειτονιά pixel είναι κάποια μαζεμένα pixel, ορισμένα από τις σχετικές τοποθεσίες τους με αυτό το pixel, που ονομάζεται κεντρικό pixel. Η γειτονιά είναι ένα ορθογώνιο σχήμα και καθώς μετακινούμαστε από το ένα στοιχείο στο επόμενο σε μία μήτρα εικόνας, το γειτονικό σχήμα γλιστράει προς την ίδια κατεύθυνση.

Εφαρμόζοντας γραμμικό και μη γραμμικό φιλτράρισμα

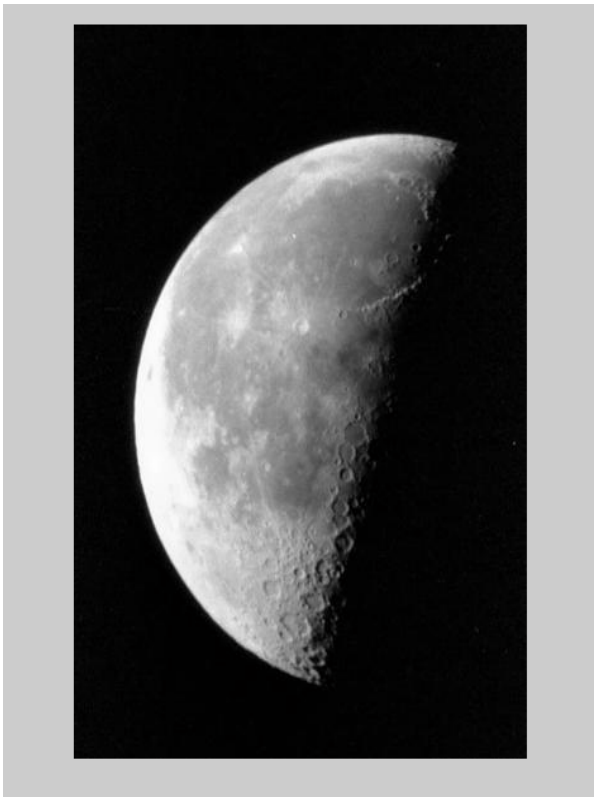
Μπορούμε να χρησιμοποιήσουμε διεργασίες κύλισης γειτονιάς για την εφαρμογή πολλών ειδών διεργασιών φιλτραρίσματος.

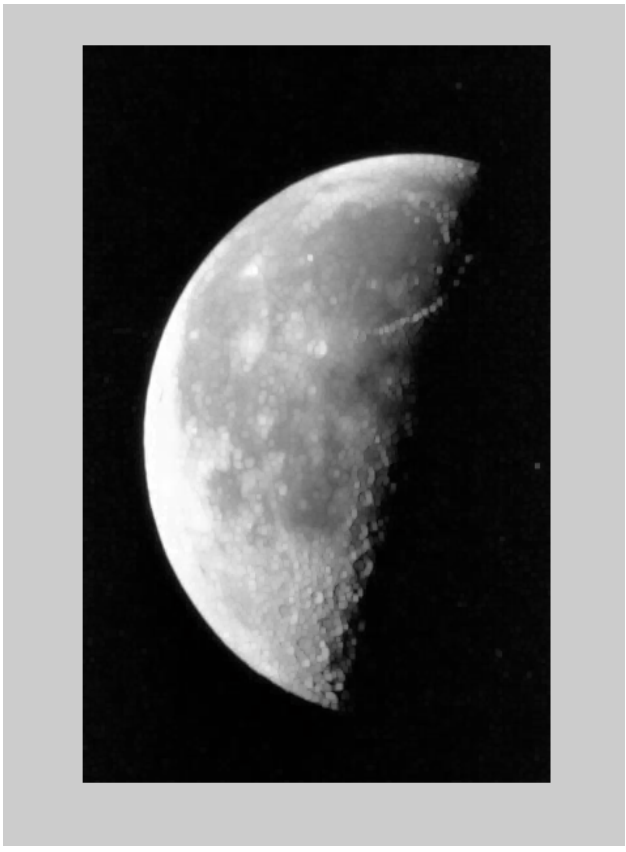
Μπορούμε να χρησιμοποιήσουμε τη συνάρτηση **nlfilter** για να εφαρμόσουμε μία ποικιλία διεργασιών κύλισης γειτονιάς. Η **nlfilter** παίρνει σαν εισερχόμενα δεδομένα μία εικόνα, ένα μέγεθος γειτονιας, και μία συνάρτηση, και επιστρέφει μία εικόνα ίδιου μεγέθους με την εισερχόμενη εικόνα. Η αξία κάθε pixel στην εξερχόμενη εικόνα καταχωρήται παίρνοντας την αντίστοιχη εισερχόμενη γειτονιά pixel στη συνάρτηση.

Παράδειγμα:

```
I=imread('moon2.jpg');
I2=nlfilter(I,[3 3],'std2');
imshow(I2)
```

```
I=imread('moon2.jpg');
f=@(x)max(x(:));
I2=nlfilter(I,[3 3],f);
figure, imshow(I)
figure, imshow(I2)
```





Σύνοψη

Σ'αυτήν την εργασία χρησιμοποιήσαμε το Matlab ώστε να δείξουμε πως μπορούμε να επεξεργαστούμε τις ψηφιακές εικόνες.

Αρχικά εισάγουμε την εικόνα στο Matlab ώστε να πάρουμε πληροφορίες σχετικά με τα pixels της εικόνας, την μέτρηση των χαρακτηριστικών, την ευρεση τελικού σημείου και άλλες πληροφορίες.

Ακόμα μπορούμε να επεξεργαστούμε την εικόνα που έχουμε εισαγάγει ώστε να πάρουμε συμπεράσματα γι'αυτήν ανάλογα με το τι θέλουμε να μάθουμε. Δηλαδή, προσαρμογή αντίθεσης φωτεινότητας, προβολή πολλαπλών εικόνων, πολλαπλά καρέ εικόνας, αλλαγή μεγέθους εικόνας, περικόπή εικόνας ώστε να μπορούμε να επεξεργαστούμε ένα μέρος της εικόνας και φιλτράρισμα εικόνας για τροποποίηση ή ενίσχυση. Επίσης στο Matlab μπορούμε να εντοπίσουμε και να διορθώσουμε μορφολογικές πράξεις όπως διαστολή και διάβρωση. Το Matlab είναι πολύ χρήσιμο ακόμα γιατί μας δίνει πληροφορίες για pixels, στατιστικά στοιχεία και το ιστόγραμμα. Μπορούμε ακόμα να αναλύσουμε την υφή της εικόνας ειδικά όταν πρόκειται για μορφολογικά-τοπογραφικά στοιχεία. Μία διεργασία τεχνικής ενίσχυσης είναι η τροποποίηση συχνότητας. Αυτές οι τεχνικές χρησιμεύουν για να αφαιρέσουμε θόρυβο από την εικόνα, να την ξεθαμπώσουμε και να κάνουμε διεργασίες κύλισης γειτονιάς των pixels.

Μπορούμε να βγάλουμε σαν συμπέρασμα ότι το Matlab είναι ένα πολύ χρήσιμο εργαλείο για την ψηφιακή ανάλυση και επεξεργασία εικόνας σύμφωνα με τα προαναφερθέντα, το οποίο μπορεί να χρησιμοποιήθει για πάρα πολλούς σκοπούς και σε όλους τους τομείς της επιστήμης,

Βιβλιογραφία

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

- "Ψηφιακή Επεξεργασία Εικόνας". Πήττας Ιωάννης. Ειδιωτική Έκδοση 2001

- "Image Processing the Fundamentals". Πέτρου Μαρία, Μποσδογιάννη Παναγιώτα. Wiley 1 edition (October 5, 1999). ISBN 0471998834

- "Image Processing and Analysis". R.Baldock, J.Graham. Oxford university press. ISBN 01909637008

- "The Image Processing Handbook". John C Russ. CRC Press 4th edition(Jule 26, 2002). ISBN 084931142x

- "Color Image Processing Methods and Aplication". R Lukas, K N Plataniotis. CRC/Taylor 8 Francis 2006. ISBN 084939774x

- "Digital Image Processing". Rafael C Gonzales, Richard E Woods. Pearson edicution.inc. ISBN 013168728x

- "Deblurring Images: Matrices, Spectra and Filtering". Hansen, Nagy O'Leary. Society for Industrial and Apllied Mathematics. ISBN 0848716187

- Image proceesing Toolbox από το Matlab