



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Τμήμα Ηλεκτρονικής

Επεξεργασία Φωνής και Εικόνας

Πτυχιακή Εργασία : Χρήστος Νικολόπουλος
Όνομα Διδάσκοντος Καθηγητή : Δρ. Αντώνιος Κωνσταντάρας

Περιεχόμενα

| | |
|---|----|
| Περίληψη..... | 4 |
| Ψηφιακή Επεξεργασία Εικόνας..... | 5 |
| Ιστορικό | 5 |
| Διεργασίες..... | 5 |
| Ανάγνωση αρχείων εικόνας..... | 6 |
| Εγγραφή αρχείων εικόνας..... | 6 |
| Εξειδίκευση επιπρόσθετων παραμέτρων ειδικής μορφής..... | 6 |
| Ανάγνωση και εγγραφή δυαδικών εικόνων σε διαμόρφωση 1-Bit..... | 6 |
| Μετατροπή αρχείων γραφικής μορφής | 6 |
| Χρήση του imshow για απεικόνιση εικόνων..... | 7 |
| Προσδιορισμός αρχικής μεγέθυνσης εικόνας..... | 8 |
| Έλεγχος εμφάνισης σχήματος..... | 8 |
| Γραμμή εργαλείων διερεύνησης εικόνων | 9 |
| Άνοιγμα γραμμής εργαλείων εικόνας..... | 9 |
| Προσδιορισμός αρχικής μεγέθυνσης της εικόνας..... | 9 |
| Προσδιορισμός του χάρτη χρωμάτων..... | 9 |
| Εισαγωγή δεδομένων εικόνας από το workspace..... | 10 |
| Εξαγωγή δεδομένων εικόνας από το workspace..... | 11 |
| Χρήση του getimage για την εξαγωγή δεδομένων εικόνας..... | 11 |
| Κλείσιμο της γραμμής εργαλείων εικόνας..... | 11 |
| Εκτύπωση της εικόνας από τη γραμμή εργαλείων εικόνας..... | 11 |
| Χρήση του βοηθήματος πλοήγησης της γραμμής εργαλείων εικόνας..... | 11 |
| Εξαγωγή πληροφοριών σχετικά με τα pixels εικόνας..... | 12 |
| Μέτρηση στοιχείων εικόνας..... | 13 |
| Εξαγωγή δεδομένων των παραμέτρων και απόστασης..... | 14 |
| Παραμετροποίηση της εμφάνισης του εργαλείου απόστασης..... | 14 |
| Εξαγωγή πληροφοριών εικόνας..... | 15 |
| Ρύθμιση αντίθεσης και τη φωτεινότητας εικόνας..... | 15 |
| Προβολή πολλαπλών εικόνων..... | 16 |
| Απεικόνιση πολλαπλών εικόνων στο ίδιο σχήμα..... | 16 |
| Τεχνικές ειδικών απεικονίσεων..... | 17 |
| Πρόσθεση μπάρας χρωμάτων..... | 17 |
| Απεικόνιση πολλαπλών πλαισίων εικόνας ταυτόχρονα..... | 18 |
| Χωρική μετατροπή..... | 19 |
| Αλλαγή μεγέθους εικόνας..... | 19 |
| Προσδιορισμός μεγέθους της εικόνας εξόδου..... | 19 |
| Περιστροφή μιας εικόνας | 20 |
| Προσδιορισμός της μεθόδου παρεμβολής..... | 20 |
| Cropping μιας εικόνας..... | 21 |
| Διαλογή σημείων ελέγχου..... | 21 |
| Εξαγωγή σημείων ελέγχου στο workspace..... | 22 |
| Γραμμικό φιλτράρισμα και φίλτρο σχεδίου..... | 23 |
| Φιλτράρισμα με χρήση του imfilter..... | 23 |
| Χρήση προκαθορισμένων τύπων φίλτρου..... | 23 |
| Μορφολογικές λειτουργίες..... | 24 |
| Διαστολή και διάβρωση..... | 24 |
| Διάβρωση εικόνας..... | 24 |
| Συνδυασμός διαστολής και διάβρωσης..... | 25 |
| Μορφολογικό άνοιγμα..... | 25 |
| Διαστολή και διάβρωση-Βασικές Συναρτήσεις..... | 26 |

| | |
|---|----|
| Σκελετοποίηση..... | 26 |
| Περιμετρικός προσδιορισμός..... | 26 |
| Ανάλυση και Ενίσχυση Εικόνας..... | 27 |
| Εξαγωγή αξιών και στατιστικών στοιχείων pixel..... | 27 |
| Παρουσίαση δεδομένων χωροπεριγράμματος εικόνας..... | 28 |
| Δημιουργία ιστογράμματος εικόνας..... | 30 |
| Ανάλυση εικόνας..... | 31 |
| Ανίχνευση άκρων..... | 31 |
| Εντοπισμός ορίων..... | 34 |
| Ανίχνευση γραμμών με χρήση μετατροπής Hough..... | 37 |
| Ανάλυση υψής εικόνας..... | 43 |
| Λειτουργίες υψής..... | 43 |
| Προσαρμογή έντασης..... | 45 |
| Προσαρμογή τιμών έντασης σε ορισμένο πεδίο..... | 46 |
| Εξειδίκευση των ορίων προσαρμογής..... | 46 |
| Εξίσωση ιστογράμματος..... | 47 |
| Απαλοιφή θορύβου..... | 48 |
| Φίλτρο μέσου..... | 48 |
| Επεξεργασία ROI..... | 51 |
| Επιλογή πεδίου ενδιαφέροντος (ROI)..... | 51 |
| Επιλογή πολυγωνικού πεδίου ενδιαφέροντος (ROI) διαδραστικά..... | 51 |
| Φιλτράρισμα ROI..... | 53 |
| Φιλτράρισμα πεδίου εικόνας..... | 53 |
| Καθαρισμός εικόνας με κανονικό φίλτρο..... | 54 |
| Καθαρισμός εικόνας με αλγόριθμο τυφλής αποσυνέλιξης..... | 56 |
| Χρήση λειτουργίας αποσυνέλιξης για το καθαρισμό εικόνας..... | 56 |
| Συρόμενες γειτονικές εφαρμογές..... | 60 |
| Εφαρμογή γραμμικού και μη γραμμικού φιλτραρίσματος..... | 60 |

Περίληψη

Στην επιστήμη του Ηλεκτρονικού Μηχανικού και των υπολογιστών, η επεξεργασία εικόνας είναι οποιαδήποτε μορφή επεξεργασίας σήματος για την οποία η είσοδος είναι μια εικόνα, όπως ένα πλαίσιο φωτογραφίας ή βίντεο. Η έξοδος από την επεξεργασία εικόνας μπορεί να είναι είτε μια εικόνα ή ένα σύνολο σχετικών χαρακτηριστικών ή παραμέτρων της εικόνας. Οι περισσότερες τεχνικές επεξεργασίας εικόνας συμπεριφέρονται στην εικόνα ως δισδιάστατο σήμα και κάνουν εφαρμογή των προτύπων σήματος επεξεργασίας.

Παρακάτω χρησιμοποιούμε 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 εύρεση 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, αλλαγή μεγέθους εικόνας, περικοπή εικόνας ώστε να μπορούμε να επεξεργαστούμε ένα μέρος της εικόνας και φιλτράρισμα εικόνας για τροποποίηση ή ενίσχυση. Επιπλέον 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 ξεθολώσουμε και να κάνουμε διεργασίες κύλισης γειτονιάς των pixels. Εκμεταλλευόμενοι τις ικανότητες του Matlab που περιγράψαμε παραπάνω σε αυτήν την εργασία έχουμε υλοποιήσει κάποιες εφαρμογές τμηματοποίησης εικόνας με βάση τα χαρακτηριστικά τους. Εν κατακλείδι μπορούμε να συμπεράνουμε ότι το Matlab είναι πολύ χρήσιμο εργαλείο για την ψηφιακή ανάλυση και επεξεργασία εικόνας σύμφωνα με τα προαναφερθέντα, κι επίσης μπορεί να χρησιμοποιηθεί για πάρα πολλούς σκοπούς και σε άλλους τομείς της επιστήμης.

Abstract

In electrical engineering and computer science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. We use Matlab to show how to edit digital images. For start we import the image in Matlab to get information about the pixels of the image, the measurement of features, finding endpoint and other information. Still we are able to process the image that we introduce to take conclusions depending on what we want to learn. That is, adjusting contrast brightness, viewing multiple images, multiple image frames, image resizing, cropping the image so we can work out a part of image filtering and image modification or enhancement. In addition Matlab can identify and correct morphological operations like dilation and erosion. Matlab is also useful because it gives us information about pixels, statistics and histogram. We can also analyze the texture of the image especially for morphological-topographical data. A process of technical assistance is to change frequency. These techniques are used to remove noise from image to clear and make processes running neighborhood of pixels. Exploiting the capabilities of Matlab described earlier in this work we have implemented some applications, image segmentation based on their characteristics. In conclusion we can deduce that Matlab is a very useful tool for analysis and digital image processing as described above, and also can be used for many purposes and in other areas of science.

Ψηφιακή επεξεργασία Εικόνας

Η ψηφιακή επεξεργασία εικόνας είναι η χρήση των υπολογιστικών αλγορίθμων για την εκτέλεση επεξεργασίας εικόνων σε ψηφιακές εικόνες. Ως υποκατηγορία της επεξεργασίας ψηφιακού σήματος, η ψηφιακή επεξεργασία εικόνας έχει πολλά πλεονεκτήματα σε σχέση με την αναλογική επεξεργασία εικόνας: επιτρέπει τη χρήση ενός ευρύτερου φάσματος αλγορίθμων στα δεδομένα εισόδου και μπορούν να αποφευχθούν προβλήματα όπως η δημιουργία θορύβου και η αλλοίωση του σήματος κατά την επεξεργασία.

Ιστορικό

Πολλές από τις τεχνικές ψηφιακής επεξεργασίας εικόνας, ή ψηφιακής επεξεργασίας φωτογραφίας όπως συχνά αποκαλείται, αναπτύχθηκαν το 1960 στα εργαστήρια Jet Propulsion, MIT, Bell Labs, Πανεπιστήμιο Maryland και σε αρκετά άλλα μέρη, με εφαρμογή στις δορυφορικές εικόνες, στη μετατροπή τηλεγραφικών φωτογραφιών, ιατρικών φωτογραφιών, βιντεοκλήσης, αναγνώριση χαρακτήρων και την ενίσχυση φωτογραφίας. Ωστόσο το κόστος της επεξεργασίας ήταν αρκετά υψηλό λόγω του υπολογιστικού εξοπλισμού εκείνης της εποχής. Το 1970, η ψηφιακή επεξεργασία εικόνας αναπτύχθηκε περισσότερο, όταν εμφανίστηκαν φθηνότεροι υπολογιστές και εξειδικευμένα εξαρτήματα. Τότε μπορούσε να γίνει η επεξεργασία εικόνων σε πραγματικό χρόνο, σε κάποια εξειδικευμένα προβλήματα όπως στη μετατροπή του τηλεοπτικού σήματος. Καθώς οι ευρείες χρήσης υπολογιστές γίνονταν πιο γρήγοροι, άρχισαν να παίρνουν το ρόλο ειδικού εξοπλισμού για πολλές περιπτώσεις εκτός από τις πιο εξειδικευμένες και τις υψηλές υπολογιστικά εφαρμογές. Με την εμφάνιση γρήγορων υπολογιστών και επεξεργαστών σημάτων το 2000, η ψηφιακή επεξεργασία εικόνας έγινε ένας από τους πιο κοινούς τρόπους επεξεργασίας και χρησιμοποιείται ευρέως, όχι μόνο ως ο πιο πολύπλευρος τρόπος επεξεργασίας, αλλά και ως ο πιο φθηνός.

Διεργασίες

Η ψηφιακή επεξεργασία εικόνας επιτρέπει την χρήση πολλών και περίπλοκων αλγορίθμων στην επεξεργασία, και για αυτό το λόγο μπορεί να προσφέρει την πιο εξελιγμένη λειτουργία σε απλές εφαρμογές και την εφαρμογή μεθόδων, οι οποίες θα ήταν αδύνατες με τα αναλογικά μέσα. Συγκεκριμένα, η ψηφιακή επεξεργασία εικόνας είναι η μόνη πρακτική τεχνολογία για τα εξής :

- Ταξινόμηση
- Εξαγωγή χαρακτηριστικών
- Αναγνώριση σχεδίων
- Προβολή
- Ανάλυση σήματος πολλαπλής κλίμακας

Ορισμένες τεχνικές που έχουν εφαρμογή στην ψηφιακή επεξεργασία εικόνας, περιλαμβάνουν:

- Ανάλυση αρχικών στοιχείων
- Ανάλυση ανεξάρτητων στοιχείων
- Χάρτες αυτο-οργάνωσης
- Κρυφά μοντέλα Markov
- Νευρωνικά δίκτυα

1[^] Azriel Rosenfeld, *Picture Processing by Computer*, New York: Academic Press, 1969

Ανάγνωση αρχείων εικόνας

Το function `imread` κάνει ανάγνωση εικόνας από οποιαδήποτε υποστηριζόμενη μορφή αρχείου εικόνας, σε οποιοδήποτε υποστηριζόμενο βάθος bit. Οι περισσότερες μορφές αρχείων εικόνας κάνουν χρήση 8 bit για την αποθήκευση pixel. Όταν αυτά διαβάζονται από την μνήμη, το MATLAB τα αποθηκεύει ως class unit 8. Για μορφές αρχείων τα οποία υποστηρίζουν 16-bit δεδομένα, PNG και TIFF, το MATLAB αποθηκεύει τις εικόνες σαν class unit 16.

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

Εγγραφή αρχείων εικόνας

Το function `imwrite` εγγράφει μια εικόνα, σε ένα αρχείο γραφικών, σε μια από τις υποστηριζόμενες μορφές. Η πιο βασική σύνταξη του `imwrite` παίρνει το μεταβλητό όνομα εικόνας και το όνομα του αρχείου. Εάν συμπεριληφθεί η προέκταση στο όνομα του αρχείου, το MATLAB συνάγει από αυτό την επιθυμητή μορφή αρχείου.

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

Εξειδίκευση επιπρόσθετων παραμέτρων ειδικής μορφής

Κατά τη χρήση του `imwrite` σε ορισμένες μορφές γραφικών, μπορεί να γίνει εξειδίκευση επιπρόσθετων παραμέτρων. Για παράδειγμα, με τα PNG αρχεία, μπορεί να γίνει εξειδίκευση του βάθους bit ως επιπρόσθετη παράμετρος. Αυτό το παράδειγμα εγγράφει μια εικόνα A σε ένα αρχείο JPEG, χρησιμοποιώντας επιπρόσθετη παράμετρο την εξειδίκευση της ποιότητας συμπίεσης.

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

Ανάγνωση και εγγραφή δυαδικών εικόνων σε μορφή 1-bit

Σε ορισμένους τύπους αρχείων, μια δυαδική εικόνα μπορεί να αποθηκευτεί με μορφή 1-bit. Εάν η μορφή αρχείου το υποστηρίζει, το MATLAB εγγράφει δυαδικές εικόνες σε 1-bit εξ'ορισμού. Όταν γίνεται ανάγνωση μιας δυαδικής εικόνας σε μορφή 1-bit, το MATLAB το αναπαριστά στο φύλλο εργασίας σαν μια λογική ακολουθία. Αυτό το παράδειγμα διαβάζει δυαδική εικόνα και την γράφει ως αρχείο TIFF. Επειδή η μορφή αρχείων TIFF υποστηρίζει 1-bit εικόνες, το αρχείο γράφεται στο δίσκο ως μορφή 1-bit.

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

Μετατροπή αρχείων γραφικής μορφής

Για την αλλαγή της γραφικής μορφής μιας εικόνας, γίνεται χρήση του `imread` για την εισαγωγή της εικόνας στο φύλλο εργασίας του MATLAB και όταν χρησιμοποιείται το function `imwrite` γίνεται η εξαγωγή της εικόνας, προσδιορίζοντας το κατάλληλο τύπο αρχείου.

Για την απεικόνιση, του συγκεκριμένου παραδείγματος το `imread` διαβάζει μια εικόνα σε μορφή `bitmap(BMP)` στο φύλλο εργασίας. Το παράδειγμα τότε εγγράφει τη `bitmap` εικόνα σε ένα αρχείο χρησιμοποιώντας PNG (Portable Network Graphics) μορφή.

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

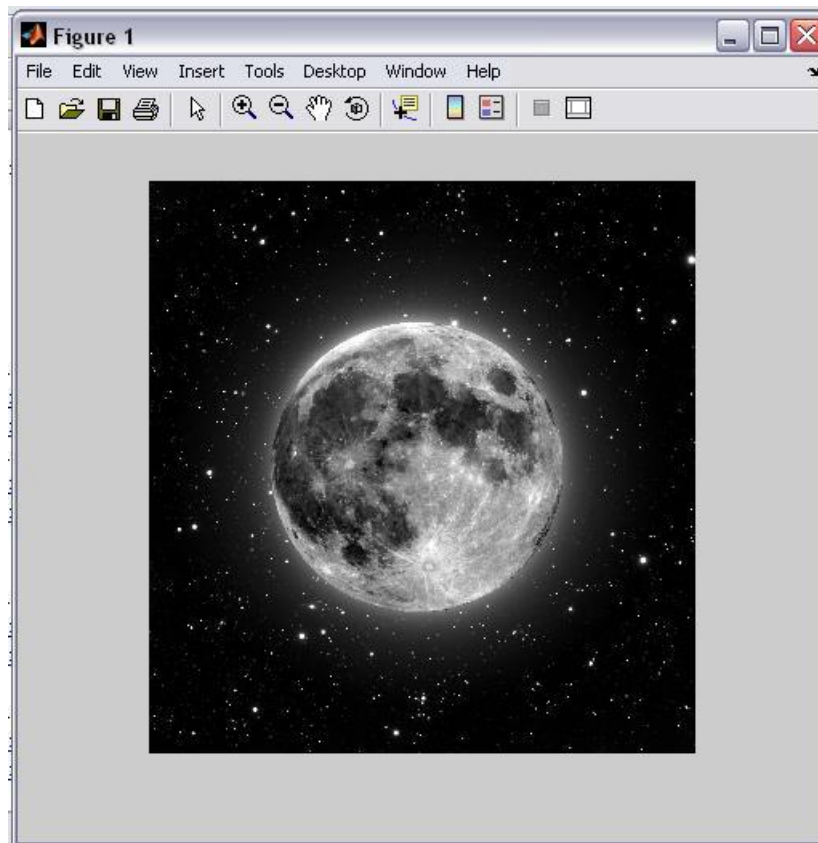
Χρήση του imshow για την εμφάνιση εικόνων

Μπορεί να γίνει χρήση του imshow function για την εμφάνιση μιας εικόνας η οποία έχει ήδη εισαχθεί στο φύλλο εργασίας του MATLAB ή την εμφάνιση μιας εικόνας η οποία έχει αποθηκευτεί σε ένα αρχείο γραφικών. Για παράδειγμα, αυτός ο κώδικας διαβάζει μια εικόνα μέσα στο φύλλο εργασίας του MATLAB και τότε την εμφανίζει στο σχηματικό παράθυρο του MATLAB.

Παράδειγμα:

```
moon = imread('moon.tif');
```

```
imshow(moon);
```



Προσδιορισμός αρχικής μεγέθυνσης εικόνας

Εξ'ορισμού το `imshow` επιχειρεί να εμφανίσει την εικόνα στο 100 % της μεγέθυνσής της (ένα pixel οθόνης για κάθε pixel εικόνας). Ωστόσο, εάν μια εικόνα είναι πολύ μεγάλη για να χωρέσει σε ένα παράθυρο σχηματικού στην οθόνη στο 100% της μεγέθυνσης, το `imshow` κλιμακώνει την εικόνα για να χωρέσει στην οθόνη και εμφανίζει μήνυμα προειδοποίησης.

Για να ξεπεραστεί η αρχική λειτουργία μεγέθυνσης εικόνας, για κάποια ειδικότερη αιτία, πρέπει να προσδιοριστεί στο `imshow` η παράμετρος της αρχικής μεγέθυνσης (`InitialMagnification`). Για παράδειγμα, για την εμφάνιση μιας εικόνας στο 50% μεγέθυνσης, να γίνει χρήση του παρακάτω κώδικα.

Παράδειγμα: `moon = imread('moon.jpg');`
`imshow(moon, 'InitialMagnification', 50)`

Έλεγχος εμφάνισης σχήματος

Εξ'ορισμού, όταν το `imshow` εμφανίζει μια εικόνα μέσα σε ένα παράθυρο (`figure`), περιβάλλει την εικόνα σε γκρι πλαίσιο. Αυτή η ρύθμιση μπορεί να αλλαχθεί και να αφαιρεθεί το πλαίσιο χρησιμοποιώντας την παράμετρο πλαισίου στο `imshow`, όπως παρουσιάζεται στο ακόλουθο παράδειγμα.

Παράδειγμα:

`imshow('moon.jpg','Border','tight')`



ω

Γραμμή εργαλείων διερεύνησης εικόνας

Η γραμμή εργαλείων εικόνας είναι ένα εργαλείο εμφάνισης εικόνων το οποίο παρέχει πρόσβαση σε αρκετά άλλα εργαλεία, όπως είναι το εργαλείο περιφέρειας pixel, το εργαλείο πληροφορίας εικόνων και το εργαλείο προσαρμογής αντίθεσης.

Άνοιγμα γραμμής εργαλείων εικόνας

Για να ξεκινήσει το image tool, πρέπει να χρησιμοποιηθεί το function του imtool

Παράδειγμα:

```
moon = imread('moon.jpg');
```

```
imtool(moon)
```

Προσδιορισμός αρχικής μεγέθυνσης εικόνας

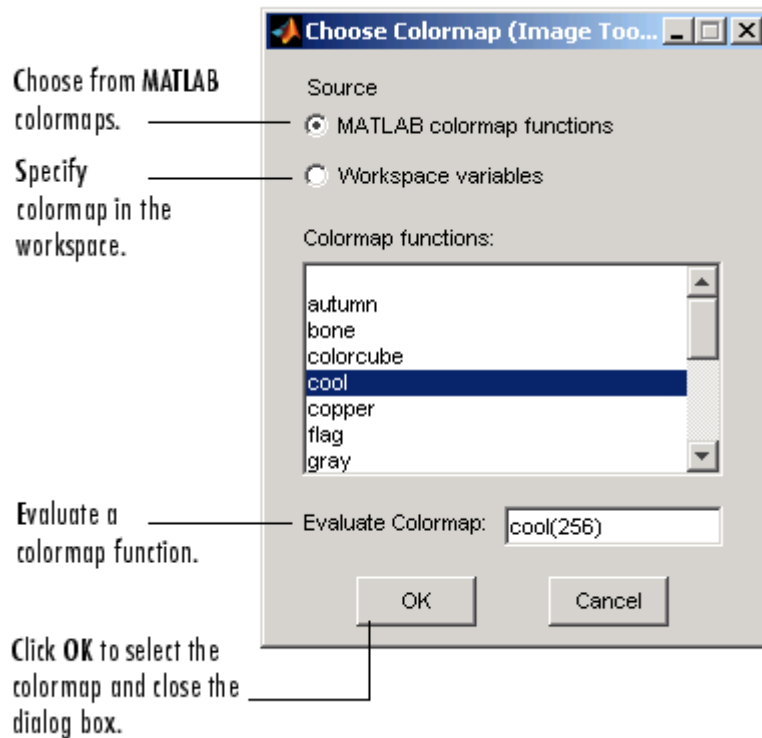
Όπως στο imshow, το function imtool προσπαθεί να απεικονίσει την εικόνα στο 100% μεγέθυνσης (ένα pixel οθόνης για κάθε pixel εικόνας)

Παράδειγμα: rout = imread('rout.tif');
imtool(rout, 'InitialMagnification', 150)

Προσδιορισμός του χάρτη χρωμάτων

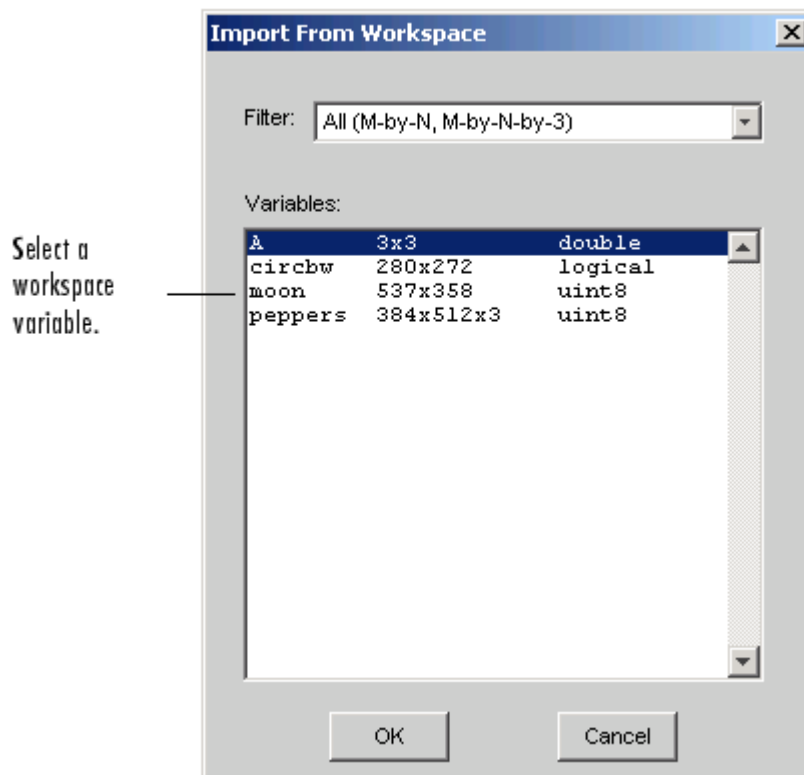
Για να προσδιοριστεί ο χάρτης χρωμάτων στην απεικόνιση μιας εικόνας ή μιας εικόνας grayscale στα εργαλεία εικόνας, πρέπει να γίνει χρήση της επιλογής choose colormap στο μενού tools. Αυτό ενεργοποιεί το εργαλείο choose colormap. Χρησιμοποιώντας αυτό το εργαλείο μπορεί να γίνει επιλογή ενός colormap του MATLAB ή να επιλεγεί μια μεταβλητή colormap από το φύλλο εργασίας του MATLAB.

Choose Colormap Tool



Εισαγωγή δεδομένων εικόνας από το workspace

Για να γίνει εισαγωγή δεδομένων εικόνας από το φύλλο εργασίας του MATLAB στο image tool, πρέπει να χρησιμοποιηθεί η επιλογή Import from Workspace από το File menu του Image Tool. Μέσα στο κουτί διαλόγου, όπως παρακάτω, πρέπει να γίνει επιλογή της μεταβλητής του φύλλου εργασίας την οποία θέλουμε να εισάγουμε μέσα στο φύλλο εργασίας.



Εξαγωγή δεδομένων από το workspace

Για να εξάγουμε την εικόνα την οποία απεικονίζουμε στο image tool του φύλλου εργασίας του MATLAB, μπορεί να χρησιμοποιηθεί η επιλογή Export to Workspace στο μενού Image Tool File



Χρήση του getimage για την εξαγωγή δεδομένων εικόνας

Μπορεί επίσης να χρησιμοποιηθεί το getimage function για να πάρουμε δεδομένα εικόνας από το Image tool στο φύλλο εργασίας του MATLAB. Το getimage function ανακτά δεδομένα εικόνας (Δεδομένα CD) από το Current Handle Graphics image object. Επειδή, εξ'ορισμού, το Image Tool δεν χειρίζεται τα οπτικά αντικείμενα, πρέπει να χρησιμοποιηθεί το toolbox function imgca για να χειριστεί τους άξονες της εικόνας που απεικονίζεται στο Image Tool.

Παράδειγμα: moon=getimage(imgca);

Κλείσιμο της γραμμής εργαλείων εικόνας

Για να κλείσει το παράθυρο Image Tool, πρέπει να γίνει χρήση του κουμπιού Close στη μπάρα του παραθύρου ή να επιλεγεί η επιλογή close από το μενού File του Image Tool. Μπορεί επίσης να γίνει χρήση της λειτουργίας imtool για να επανεμφανιστεί το χερούλι στο Image Tool και να χρησιμοποιηθεί το χερούλι για να κλείσει. Όταν κλείσει το Image Tool, κλείνουν ταυτόχρονα και όλες οι σχετικές γραμμές εργαλείων που είναι ανοιχτές.

Παράδειγμα:

```
imtool close
```

```
imtool close all
```

Εκτύπωση της εικόνας από τη γραμμή εργαλείων

Για να εκτυπωθεί η εικόνα που απεικονίζεται στο Image Tool, πρέπει να γίνει χρήση της επιλογής Print to Figure από το μενού File. Το Image Tool ανοίγει ένα άλλο παράθυρο και εμφανίζει την εικόνα. Χρησιμοποιείστε την επιλογή Print από το μενού File για να εκτυπωθεί η εικόνα.

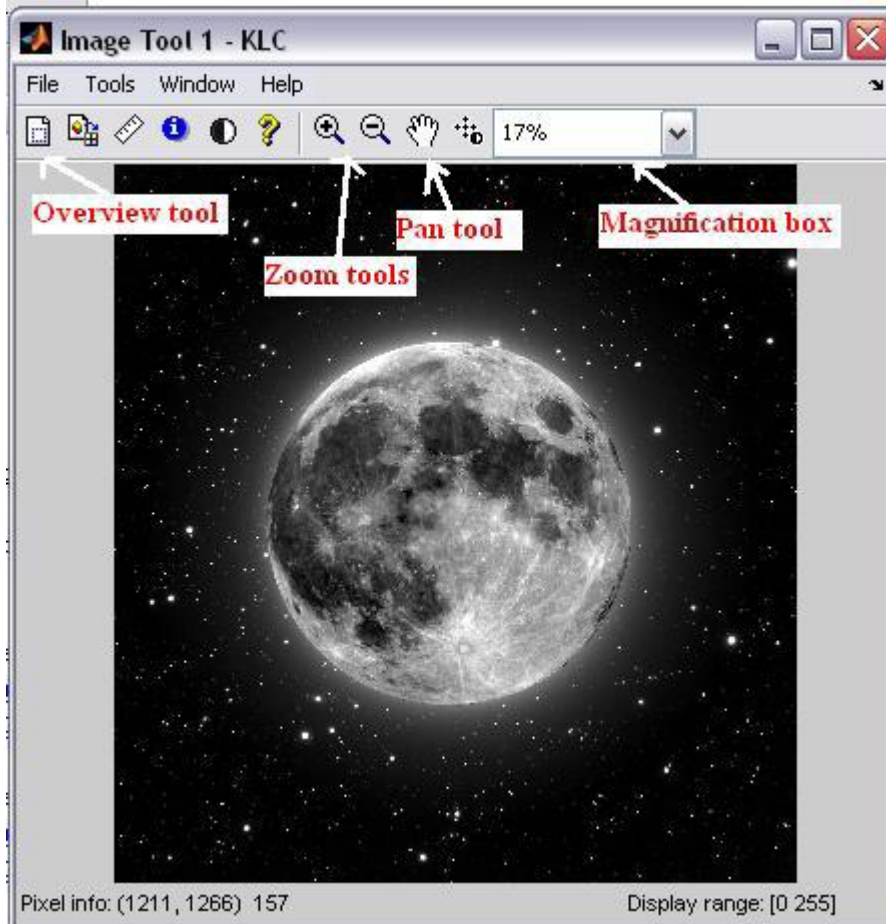
Χρήση του βοηθήματος πλοήγησης της γραμμής εργαλείων εικόνας

Εργαλείο Overview - Εμφανίζει ολόκληρη την εικόνα για να μας βοηθήσει να κατανοήσουμε πιο μέρος της εικόνας απεικονίζεται εκείνη τη στιγμή στο Image Tool.

Εργαλείο Pan - Επιτρέπει να κάνουμε Click and grab της εικόνας που απεικονίζεται και να την μεταφέρουμε στο Image Tool.

Εργαλεία Zoom - Μας επιτρέπει να κάνουμε μεγέθυνση ή σμίκρυνση της εικόνας.

Κουτί Magnification - Μας επιτρέπει να ορίσουμε την ακριβή μεγέθυνση της εικόνας.

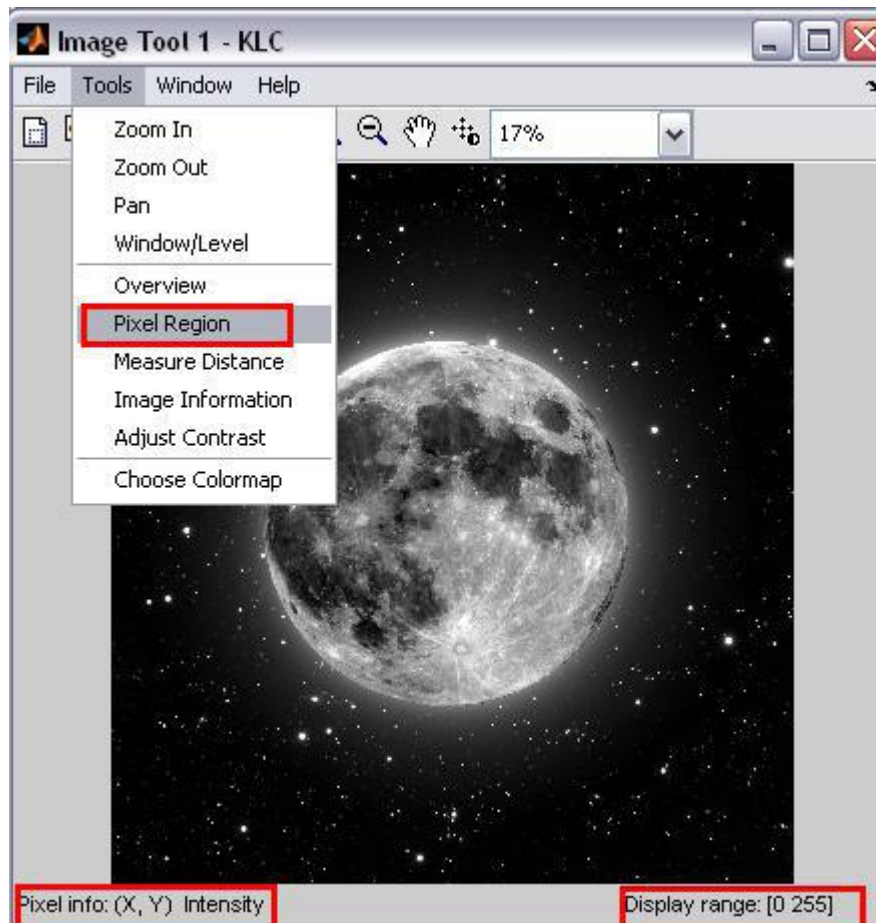


Εξαγωγή πληροφοριών σχετικά με τα Pixel εικόνας

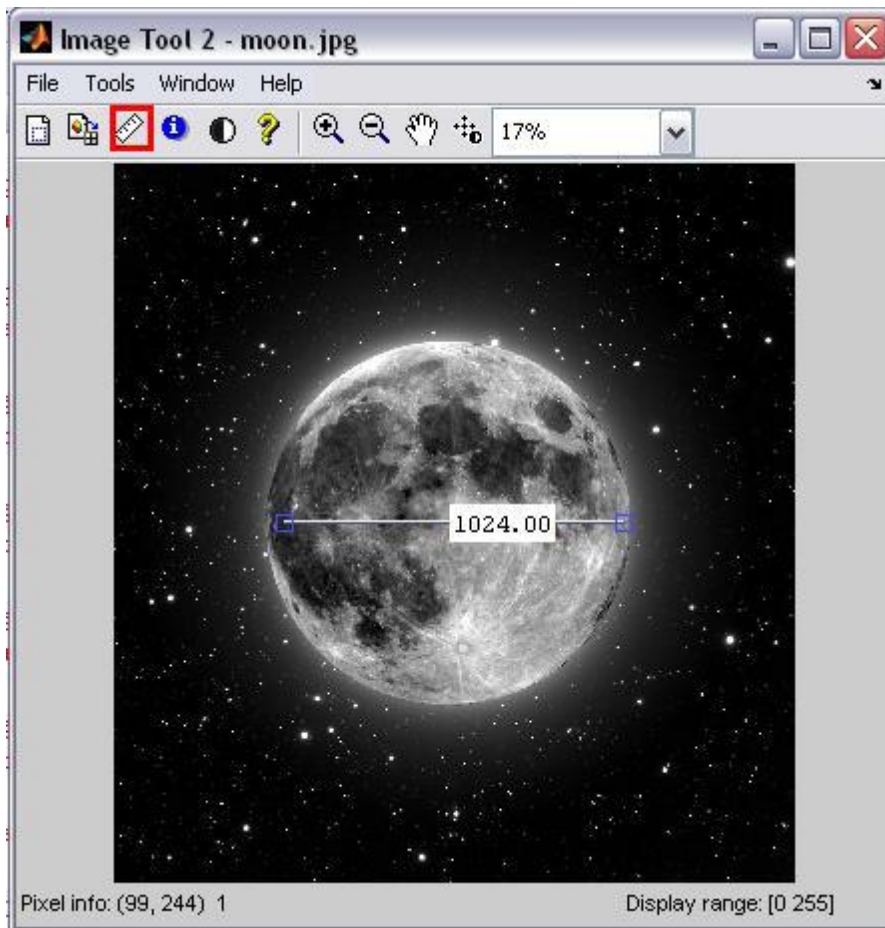
Εργαλείο Pixel Information- Εμφανίζει την τοποθεσία και την τιμή του pixel κάτω από τον κέρσορα στο παράθυρο του Image Tool.

Εργαλείο Display Range- Εμφανίζει το πεδίο απεικόνισης της εικόνας στο παράθυρο του Image Tool.

Εργαλείο Pixel Region- Εμφανίζει μια πολύ κοντινή προβολή των pixels σε μια συγκεκριμένη περιοχή εικόνας.

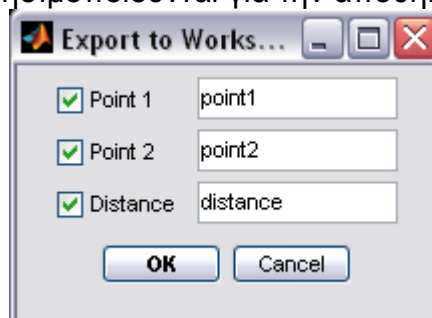


Μέτρηση στοιχείων εικόνας
Χρησιμοποιώντας το Distance Tool



Εξαγωγή δεδομένων των παραμέτρων και απόστασης

Για να αποθηκεύσουμε τις τοποθεσίες παραμέτρων και τις πληροφορίες απόστασης, πρέπει να κάνουμε δεξί click στο εργαλείο Distance και να επιλέξουμε την επιλογή Copy pixel info από το μενού περιεχομένων. Το εργαλείο Distance ανοίγει το κουτί διαλόγου Export to Workspace. Μπορούμε να χρησιμοποιήσουμε αυτό το κουτί διαλόγου, προκειμένου να προσδιορίσουμε τα ονόματα των μεταβλητών που χρησιμοποιούνται για την αποθήκευση των πληροφοριών αυτών.



Παραμετροποίηση εμφάνισης του εργαλείου απόστασης

Χρησιμοποιώντας το μενού περιεχομένων του εργαλείου Distance, μπορούμε να διαμορφώσουμε πολλές πτυχές της εμφάνισης και συμπεριφοράς αυτού, συμπεριλαμβανομένων:
Εναλλαγή ετικέτας του εργαλείου Distance από on σε off χρησιμοποιώντας την επιλογή Show

Distance Label.

Αλλαγή του χρώματος που χρησιμοποιείται για την απεικόνιση της γραμμής του εργαλείου Distance κάνοντας χρήση της επιλογής Set line color.

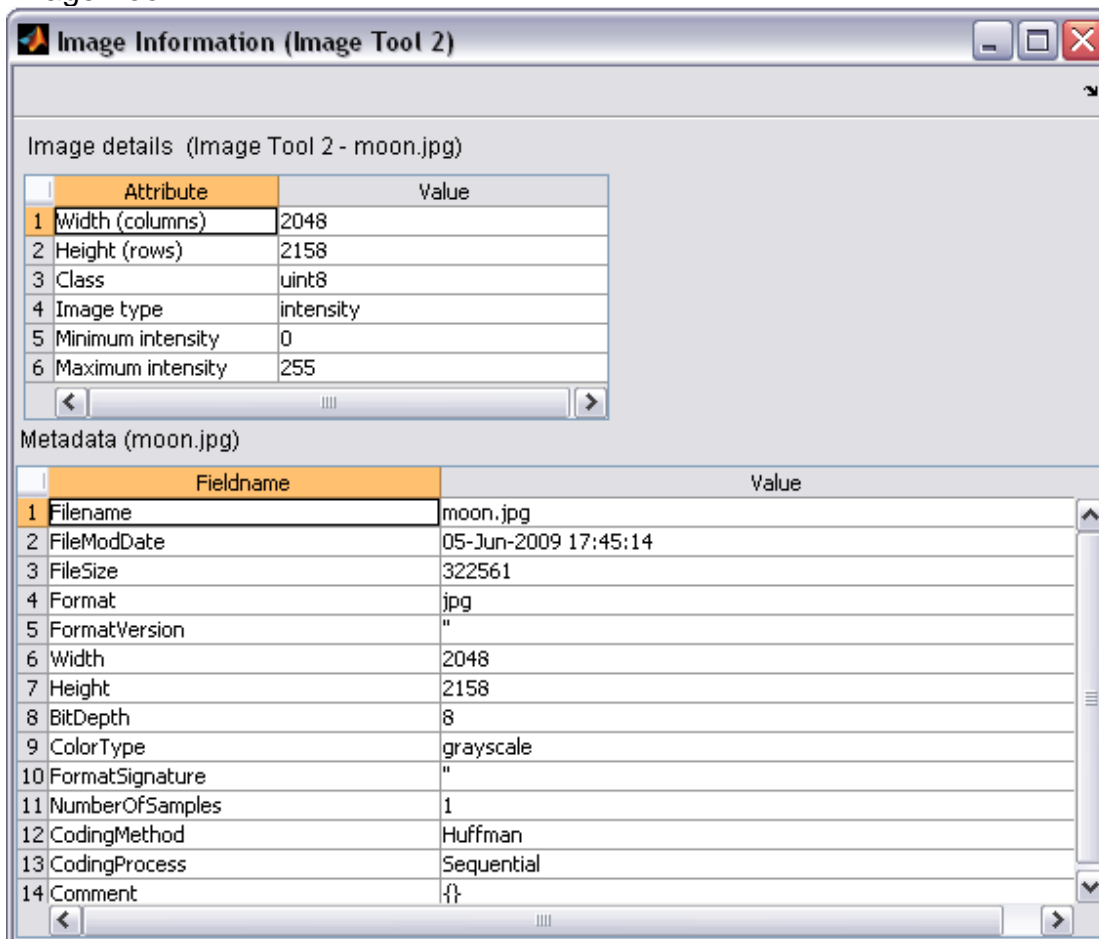
Περιορισμός της κίνησης του εργαλείου είτε οριζόντια είτε κατακόρυφα χρησιμοποιώντας την επιλογή Constrain drag.

Διαγραφή του αντικειμένου του εργαλείου Distance χρησιμοποιώντας την επιλογή Delete.

Κάνοντας δεξί click στο εργαλείο Distance, για να αποκτήσουμε πρόσβαση σε αυτό το μενού περιεχομένων.

Εξαγωγή πληροφοριών εικόνας

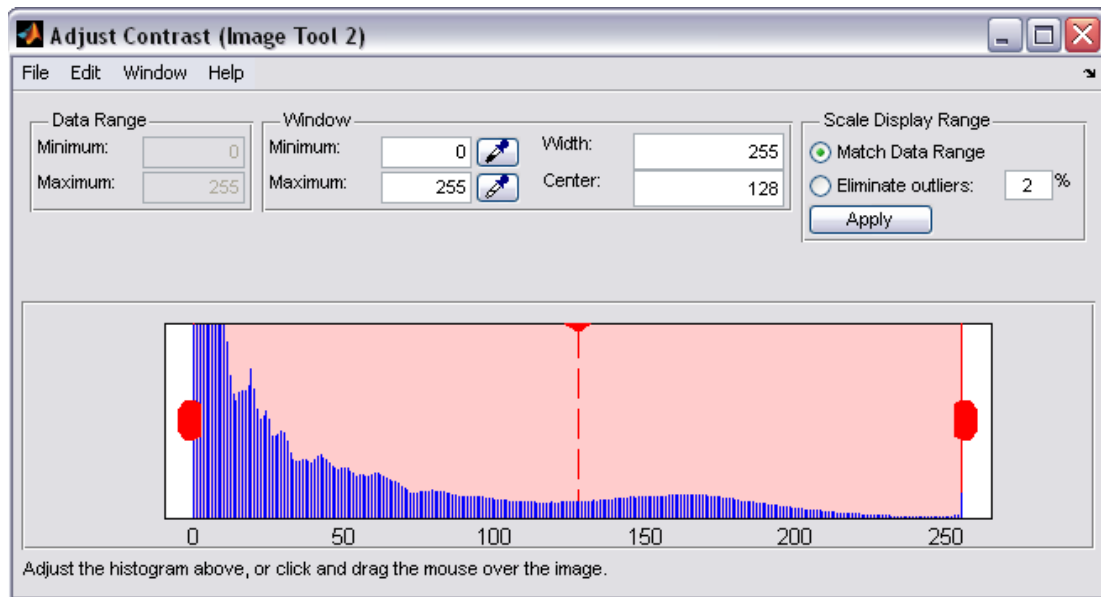
Εκκινήστε το εργαλείο Image Information κάνοντας click στο κουμπί Image Information στην γραμμή εργαλείων του Image Tool ή επιλέγοντας την επιλογή Image Information από το μενού Tools στο Image Tool.



Ρύθμιση αντίθεσης και φωτεινότητας εικόνας

Για να ρυθμίσουμε την αντίθεση και τη φωτεινότητα μιας εικόνας που απεικονίζεται στο Image Tool, χρησιμοποιούμε το εργαλείο Adjust Contrast.

Adjust Contrast Tool



Προβολή πολλαπλών εικόνων

Απεικόνιση πολλαπλών εικόνων στο ίδιο πλαίσιο

Μπορούμε να χρησιμοποιήσουμε την λειτουργία `imshow` σε συνδυασμό με τη λειτουργία `subplot` του MATLAB ή τη λειτουργία `subimage` του MATLAB, για να εμφανίσουμε πολλαπλές εικόνες σε ένα και μόνο παράθυρο πλαισίου.

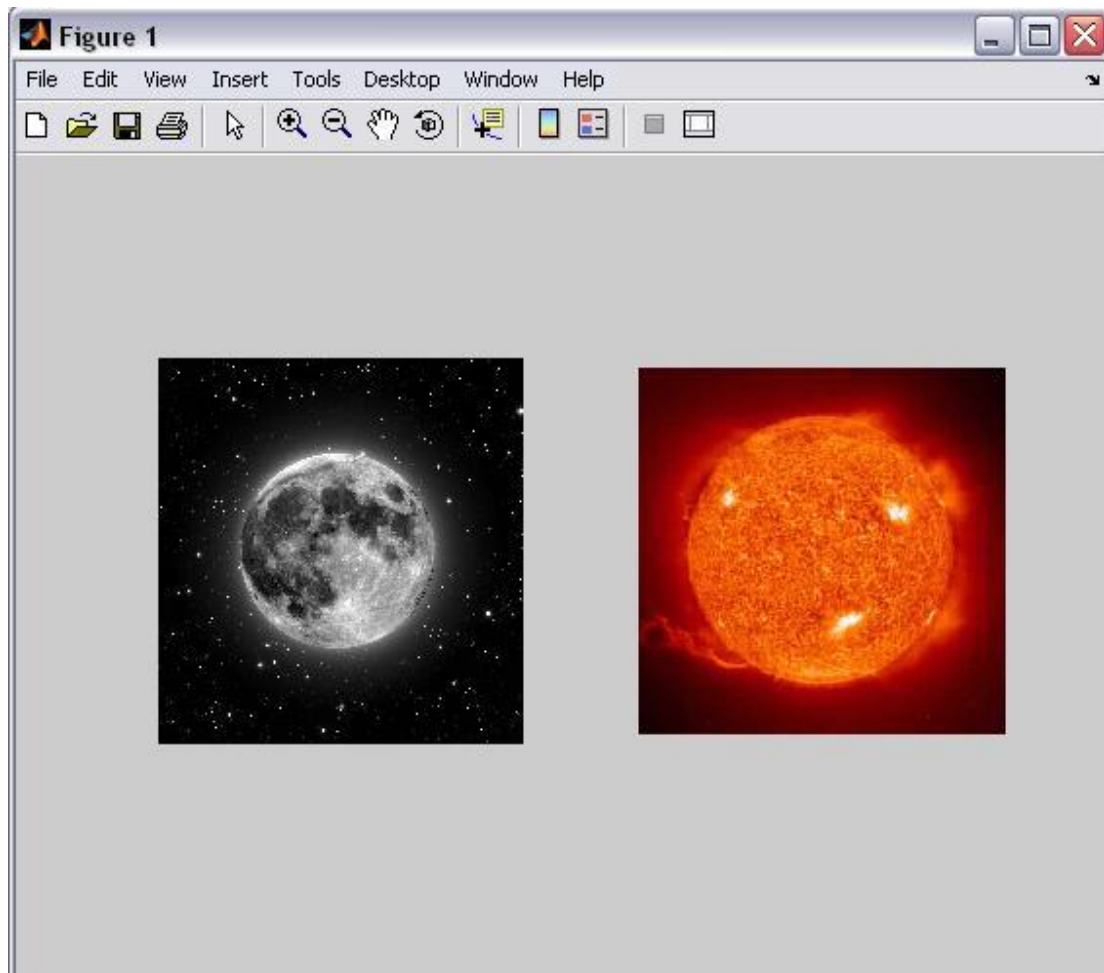
Παράδειγμα:

```
[X1,map1]=imread('moon.jpg');
```

```
[X2,map2]=imread('sun.jpg');
```

```
subplot(1,2,1), imshow(X1,map1)
```

```
subplot(1,2,2), imshow(X2,map2)
```

Τεχνικές ειδικών απεικονίσεων

Πρόσθεση μπάρας χρωμάτων

Αν θέλουμε να προσθέσουμε μια μπάρα χρωμάτων σε μια εικόνα που απεικονίζεται στο Image Tool, επιλέγουμε την επιλογή Print to Figure από το μενού File του Image Tool. Το Image Tool εμφανίζει την εικόνα σε ένα χωριστό παράθυρο πλαισίου, στο οποίο μπορούμε να προσθέσουμε μια μπάρα χρωμάτων.

Παράδειγμα:

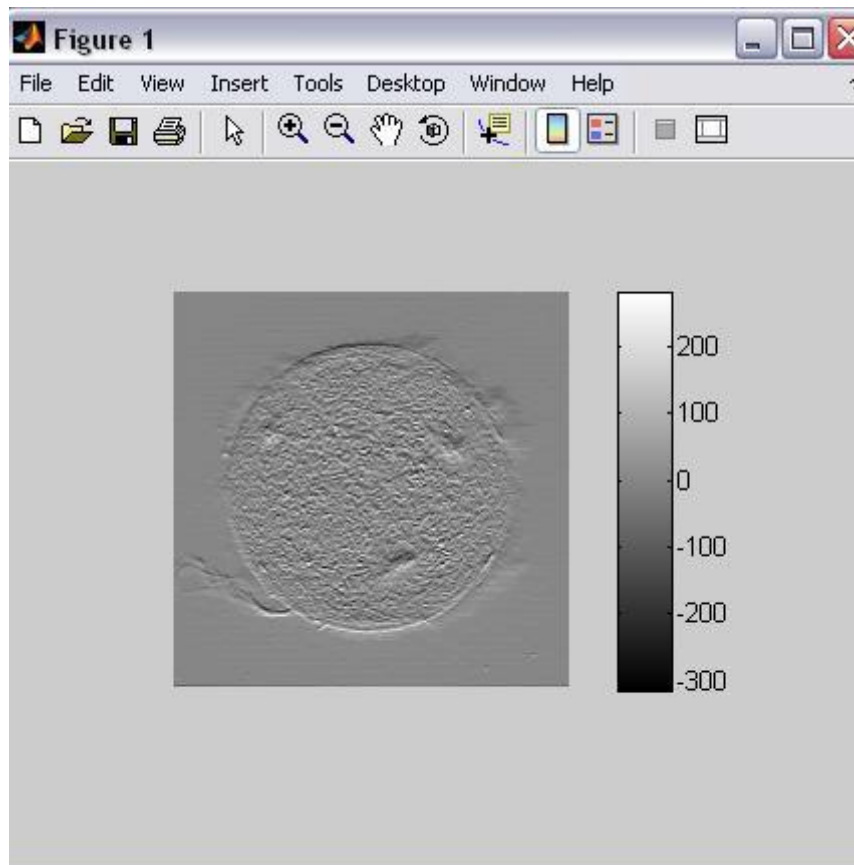
```
RGB = imread('sun.jpg');
```

```
I = rgb2gray(RGB);
```

```
h = [1 2 1; 0 0 0; -1 -2 -1];
```

```
I2 = filter2(h,I);
```

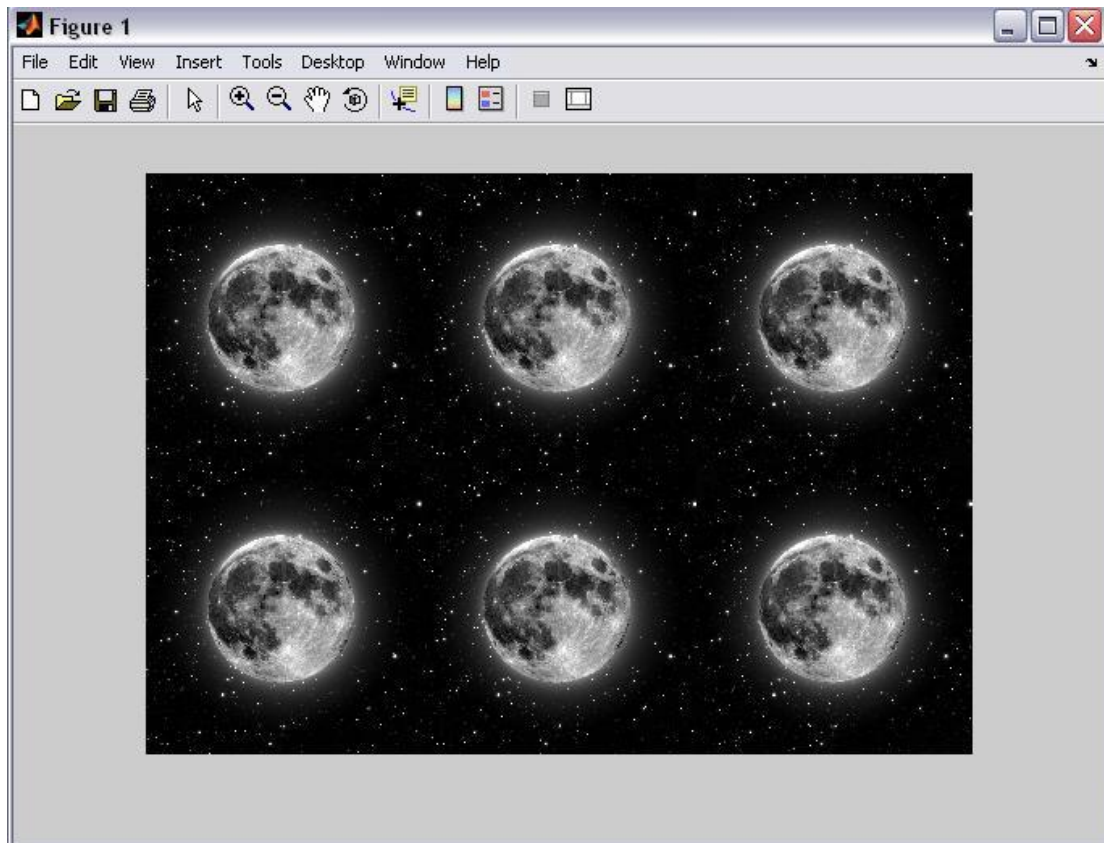
```
imshow(I2,'DisplayRange',[0,1]), colorbar
```



Εμφάνιση Πολλαπλών πλαισίων Εικόνας Ταυτόχρονα

Για να εμφανίσουμε πολλαπλά πλαίσια σε μια ακολουθία με μια κίνηση, χρησιμοποιούμε την λειτουργία `montage`. Το `montage` εμφανίζει όλα τα πλαίσια εικόνων, τακτοποιώντας τα σε ένα ορθογώνιο πλέγμα. Το μοντάζ των εικόνων αποτελεί αντικείμενο μιας μόνο εικόνας. Τα πλαίσια εικόνων μπορούν να είναι σε κλίμακα αποχρώσεων του γκρι, σε πίνακα ή εικόνες με πραγματικό χρώμα. Αν προσδιορίσουμε εικόνες σε πίνακα, πρέπει όλες να χρησιμοποιούν τον ίδιο χάρτη χρωμάτων.

Παράδειγμα: `moon = imread('moon.jpg');`
`moonArray = repmat(moon, [1 1 1 6]);`
`montage(moonArray);`



Χωρική Μετατροπή

Αλλαγή μεγέθους Εικόνας

Με τη χρήση του `imresize`, μπορούμε να προσδιορίσουμε το μέγεθος της εικόνας εξόδου με δύο τρόπους:

Προσδιορίζοντας τον παράγοντα μεγέθυνσης που χρησιμοποιείται στην εικόνα

Προσδιορίζοντας τις διαστάσεις της εικόνας εξόδου

Παράδειγμα:

```
I = imread('moon.jpg');
```

```
J = imresize(I,1.25);
```

```
imshow(I)
```

```
figure, imshow(J)
```

Προσδιορισμός Μεγέθους Εικόνας Εξόδου

Μπορούμε να προσδιορίσουμε το μέγεθος της εικόνας εξόδου εισάγοντας ένα διάνυσμα, το οποίο περιέχει τον αριθμό των γραμμών και στηλών της εικόνας εξόδου.

Παράδειγμα:

```
I = imread('moon.jpg');
```

```
J = imresize(I,[100 150]);
```

```
imshow(I)
```

figure, imshow(J)

Περιστροφή Εικόνας

Η λειτουργία `imrotate` έχει δύο βασικές δυνατότητες:

- Την περιστροφή εικόνας
- Τη γωνία περιστροφής

Παράδειγμα:

```
J = imrotate(I,35);
```

Προσδιορισμός Μεθόδου Παρεμβολής

Εξ ορισμού, το `imrotate` χρησιμοποιεί την πλησιέστερη γειτονική παρεμβολή για να καθορίσει την τιμή των pixel στην εικόνα εξόδου, ωστόσο μπορεί κανείς να ορίσει και άλλες μεθόδους παρεμβολής.

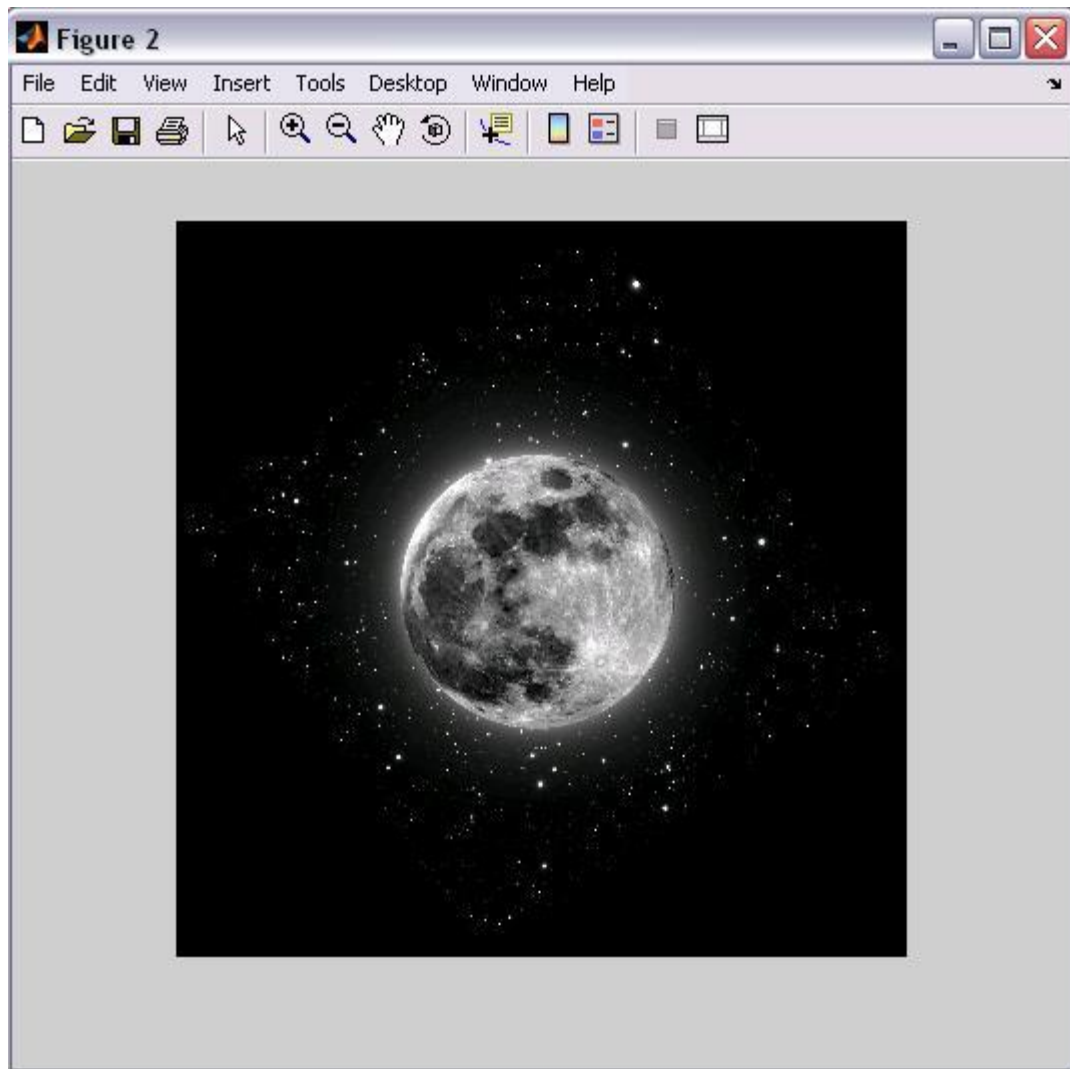
Παράδειγμα:

```
I = imread('moon.jpg');
```

```
J = imrotate(I,35,'bilinear');
```

```
imshow(I)
```

```
figure, imshow(J) 27
```



Περικοπή Εικόνας

Για να εξαγάγουμε ένα ορθογώνιο τμήμα μιας εικόνας, χρησιμοποιούμε τη λειτουργία `imcrop`, η οποία έχει δύο βασικές δυνατότητες:

- Την περικοπή της εικόνας
- Τις συντεταγμένες ενός ορθογωνίου, το οποίο ορίζει την περιοχή περικοπής

Παράδειγμα:

```
imshow moon.jpg
```

```
I = imcrop;  
imshow(I); 28
```

Επιλογή Σημείων Ελέγχου

Η επιλογή των σημείων ελέγχου είναι μια διαδικασία με τέσσερα στάδια:
Εκκινήστε το εργαλείο, προσδιορίζοντας την εικόνα εισόδου και την εικόνα βάσης.

Χρησιμοποιείτε τα βοηθήματα πλοήγησης για να εξερευνήσετε την εικόνα, αναζητώντας οπτικά στοιχεία, τα οποία μπορείτε να αναγνωρίσετε και στις δύο εικόνες.

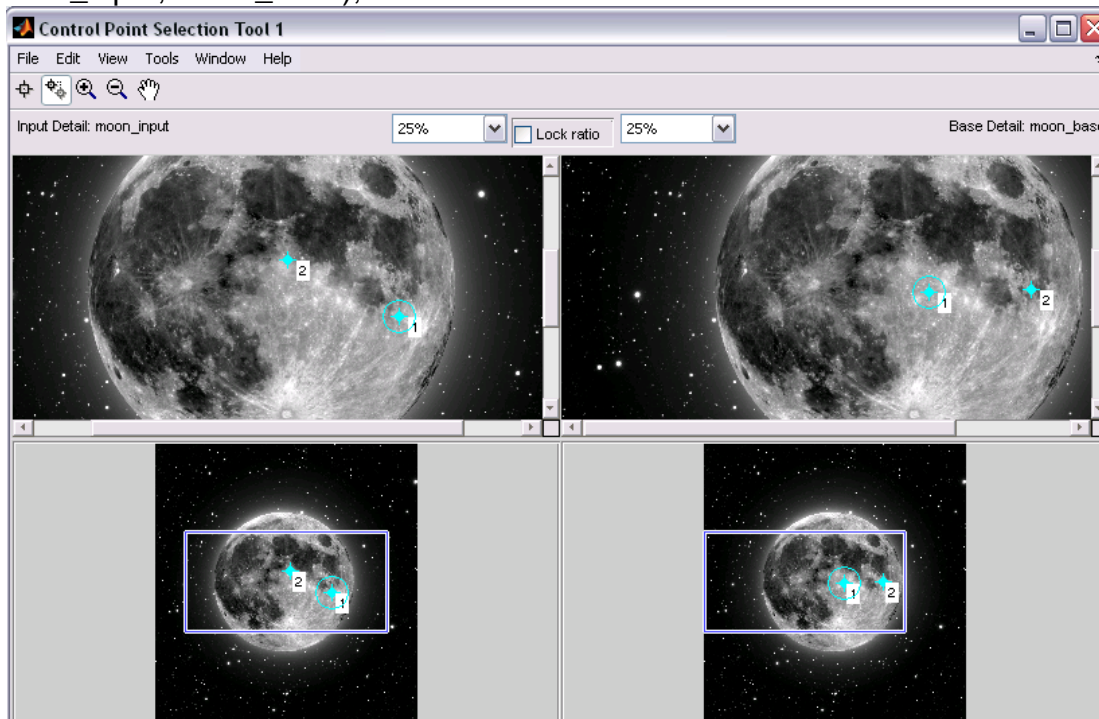
Το `cpselect` μας παρέχει πολλούς τρόπους πλοήγησης σε μια εικόνα, ραπνίng και μεγεθύνοντας, προκειμένου να μπορούμε να δούμε περιοχές της εικόνας με μεγαλύτερη λεπτομέρεια.

Προσδιορίζουμε τα ζευγάρια που ταιριάζουν στα σημεία ελέγχου στην εικόνα εισόδου και στην εικόνα βάσης.

Αποθηκεύουμε τα σημεία ελέγχου στο φύλλο εργασίας του MATLAB.

Παράδειγμα:

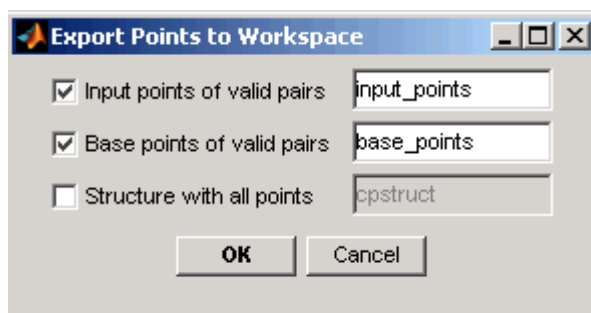
```
moon_base = imread('moon.jpg');  
moon_input = moon_base;  
cpselect(moon_input, moon_base); 29
```



Εξαγωγή Σημείων Ελέγχου στο workspace

Για να αποθηκεύσουμε τα σημεία ελέγχου στο φύλλο εργασίας του MATLAB,

- Επιλέγουμε το File στη μπάρα μενού των Εργαλείων Control Point Selection.
- Επιλέγουμε την Εξαγωγή των Σημείων στην επιλογή Φύλλο Εργασίας. Το Εργαλείο Control Point Selection εμφανίζει το κουτί διαλόγου.



Γραμμικό Φιλτράρισμα και Σχεδιασμός Φίλτρου

Το Φιλτράρισμα είναι μια τεχνική για την τροποποίηση ή την ενίσχυση μιας εικόνας. Για παράδειγμα, μπορούμε να φιλτράρουμε μια εικόνα για να τονίσουμε ορισμένα χαρακτηριστικά της ή για να αφαιρέσουμε ορισμένα άλλα. Οι λειτουργίες της επεξεργασίας εικόνας που παρέχονται με το φιλτράρισμα περιλαμβάνουν τη λείανση, ενίσχυση του προφίλ και ενίσχυση των άκρων.

Φιλτράρισμα με χρήση του `imfilter`

Το φιλτράρισμα των εικόνων είτε με συσχέτιση είτε με συνέλιξη, μπορεί να γίνει με τη χρήση της λειτουργίας `imfilter`. Αυτό το παράδειγμα φιλτράρει μια εικόνα με ένα 5 by 5 φίλτρο που περιέχει ίσα βάρη. Ένα τέτοιο φίλτρο το ονομάζουμε συχνά φίλτρο ισορροπίας.

Παράδειγμα:

```
I = imread('moon.jpg');  
  
h = ones(5,5) / 35;  
  
I2 = imfilter(I,h);  
  
imshow(I), title('Original Image');  
  
figure, imshow(I2), title('Filtered Image')
```



Χρήση Προκαθορισμένων Τύπων Φίλτρου

Η λειτουργία `fspecial` παράγει αρκετά είδη προκαθορισμένων φίλτρων στην μορφή συσχετιζόμενων πυρήνων. Το παράδειγμά μας παρουσιάζει την εφαρμογή ενός φίλτρου μάσκας `unsharp` σε μια εικόνα γκρι κλίμακας. Το φίλτρο μάσκας `unsharp` έχει την ικανότητα να κάνει τα άκρα και τις πολύ μικρές λεπτομέρειες της εικόνας πιο ζωντανές.

Παράδειγμα

```
I = imread('moon.jpg');  
  
h = fspecial('unsharp');  
  
I2 = imfilter(I,h);  
  
imshow(I), title('Original Image')  
  
figure, imshow(I2), title('Filtered Image')
```



Μορφολογικές Λειτουργίες

Η μορφολογία είναι ένας ευρύς τομέας των λειτουργιών επεξεργασίας εικόνας, ο οποίος επεξεργάζεται τις εικόνες που έχουν σαν βάση σχήματα. Οι μορφολογικές λειτουργίες εφαρμόζουν ένα δομικό στοιχείο σε μια εικόνα εισόδου, δημιουργώντας μια εικόνα εξόδου ιδίων διαστάσεων. Οι πιο βασικές μορφολογικές λειτουργίες είναι η διαστολή και η διάβρωση. Σε μια μορφολογική λειτουργία, η αξία καθενός pixel στην εικόνα εξόδου βασίζεται στην σύγκριση του αντίστοιχου pixel με την εικόνα εισόδου. Επιλέγοντας το μέγεθος και το σχήμα του πεδίου, μπορεί κανείς να δημιουργήσει μια μορφολογική λειτουργία ευαίσθητη σε ορισμένα σχήματα στην εικόνα εισόδου.

Διαστολή και Διάβρωση

Η διαστολή και η διάβρωση αποτελούν δύο θεμελιώδεις μορφολογικές λειτουργίες. Η διαστολή προσθέτει pixels στα όρια των αντικειμένων μιας εικόνας, ενώ η διάβρωση αφαιρεί pixels από τα όρια των εικόνων. Ο αριθμός των pixels που προστίθενται ή αφαιρούνται από τα αντικείμενα σε μια εικόνα εξαρτάται από το μέγεθος και το σχήμα του δομικού στοιχείου που χρησιμοποιείται για την επεξεργασία της εικόνας.

Διάβρωση Εικόνας

Για να διαβρώσουμε μια εικόνα, χρησιμοποιούμε τη λειτουργία `imerode`. Η λειτουργία `imerode` έχει δύο βασικές δυνατότητες: Την επεξεργασία της εικόνας εισόδου (ασπρόμαυρη, δυαδική ή δυαδική εικόνα με πακέτα), ένα αντικείμενο δομικού στοιχείου, το οποίο επιστρέφεται από τη λειτουργία `strel` ή μια δυαδική μήτρα που προσδιορίζει το πεδίο ενός δομικού στοιχείου.

Παράδειγμα:

```
BW1 = imread('moon.jpg');
SE = strel('arbitrary',eye(13));
BW2 = imerode(BW1,SE);
>> imshow(BW1),title('Original Image')
figure, imshow(BW2),title('Filtered Image') 34
```




Συνδυασμός Διαστολής και Διάβρωσης

Η Διαστολή και η διάβρωση χρησιμοποιούνται συχνά σε συνδυασμό με την εφαρμογή επεξεργασίας εικόνας. Για παράδειγμα, ο ορισμός του μορφολογικού ανοίγματος μιας εικόνας είναι μια διάβρωση, που ακολουθείται από μια διαστολή, χρησιμοποιώντας το ίδιο στοιχείο διάθρωσης και για τις δύο λειτουργίες. Η σχετική λειτουργία, το μορφολογικό δηλαδή κλείσιμο μιας εικόνας, είναι το αντίστροφο: συνίσταται η διαστολή που ακολουθείται από διάβρωση με το ίδιο διαρθρωτικό στοιχείο.

Μορφολογικό Άνοιγμα

Μπορείτε να χρησιμοποιήσετε το μορφολογικό άνοιγμα για την άρση των μικρών αντικειμένων από μια εικόνα διατηρώντας παράλληλα το σχήμα και το μέγεθος των μεγαλύτερων αντικειμένων στην εικόνα.

Παράδειγμα:

```
BW1 = imread('moon.jpg');
```

```
SE = strel('rectangle',[40 30]);
```

```
BW2 = imerode(BW1,SE);
```

```
imshow(BW1), title ('Original image')
```

```
figure, imshow (BW2) , title ('Filtered image')
```



Διαστολή και διάβρωση – Βασικές Συναρτήσεις

Το τμήμα αυτό περιγράφει δύο πράξεις κοινής επεξεργασίας εικόνας που βασίζονται σε διαστολή και διάβρωση:

Skeletonization

Περιμετρικός Προσδιορισμός

Σκελετοποίηση

Για τη μείωση των γραμμών όλων των αντικειμένων μιας εικόνας, χωρίς αλλαγή στη βασική δομή της εικόνας, χρησιμοποιήστε τη `bwmorph` λειτουργία. Η διαδικασία αυτή είναι γνωστή ως `skeletonization`.

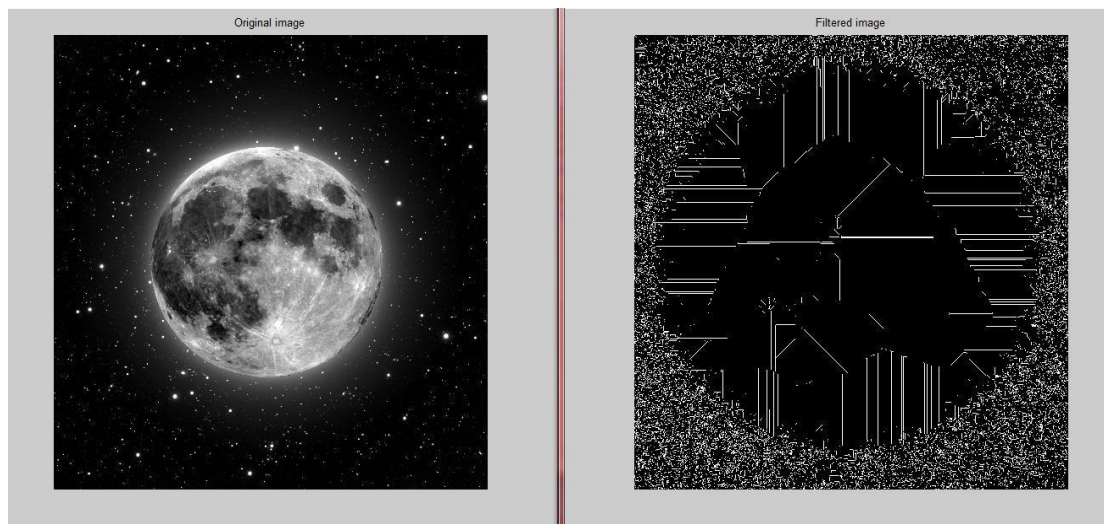
Παράδειγμα:

```
BW1 = imread('moon.jpg');
```

```
BW2 = bwmorph(BW1,'skel',Inf);
```

```
imshow(BW1), title ('Original image')
```

```
figure, imshow(BW2), title ('Filtered image')
```



Περιμετρικός Προσδιορισμός

Η λειτουργία `bwperim` καθορίζει τη περίμετρο των pixels των αντικειμένων σε μια δυαδική εικόνα. Ένα pixel θεωρείται pixel περιμέτρου, εφόσον πληροί τα εξής κριτήρια:

Το pixel είναι ενεργοποιημένο.

Ένα (ή περισσότερα) από τα κοντινά pixels είναι off.

Για παράδειγμα, αυτός ο κωδικός βρίσκει τα περιμετρικά pixels σε μια

δυναμική εικόνα ενός κυκλώματος.

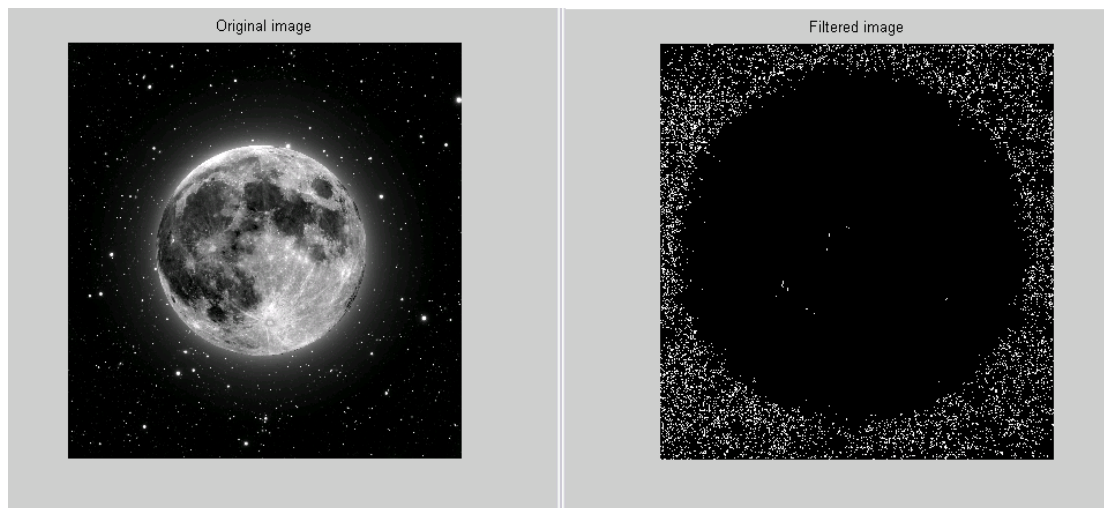
Παράδειγμα:

```
BW1 = imread('moon.jpg');
```

```
BW2 = bwperim(BW1);
```

```
imshow(BW1), title ('Original image')
```

```
figure, imshow(BW2), title ('Filtered image') 37
```



Ανάλυση και Ενίσχυση εικόνας

Το κεφάλαιο αυτό περιγράφει τις λειτουργίες που υποστηρίζουν μια σειρά από σπάντα λειτουργίες επεξεργασίας εικόνας για την ανάλυση και ενίσχυση εικόνων.

Εξαγωγή αξιών και στατιστικών στοιχείων pixel

Αυτή η ενότητα περιγράφει πώς να εξαχθούν πληροφορίες σχετικά με τις τιμές των δεδομένων που συνθέτουν μια εικόνα. Θέματα που καλύπτονται περιλαμβάνουν:

Συγκέντρωση πληροφοριών για Pixels Εικόνας

Προφίλ Έντασης μιας εικόνας

Χρησιμοποίηση ενός Οικοπέδου Contour για δεδομένα εικόνας

Δημιουργία ενός Ιστογράμματος εικόνας

Συγκέντρωση Στατιστικής Περίληψης σχετικά με μια εικόνα

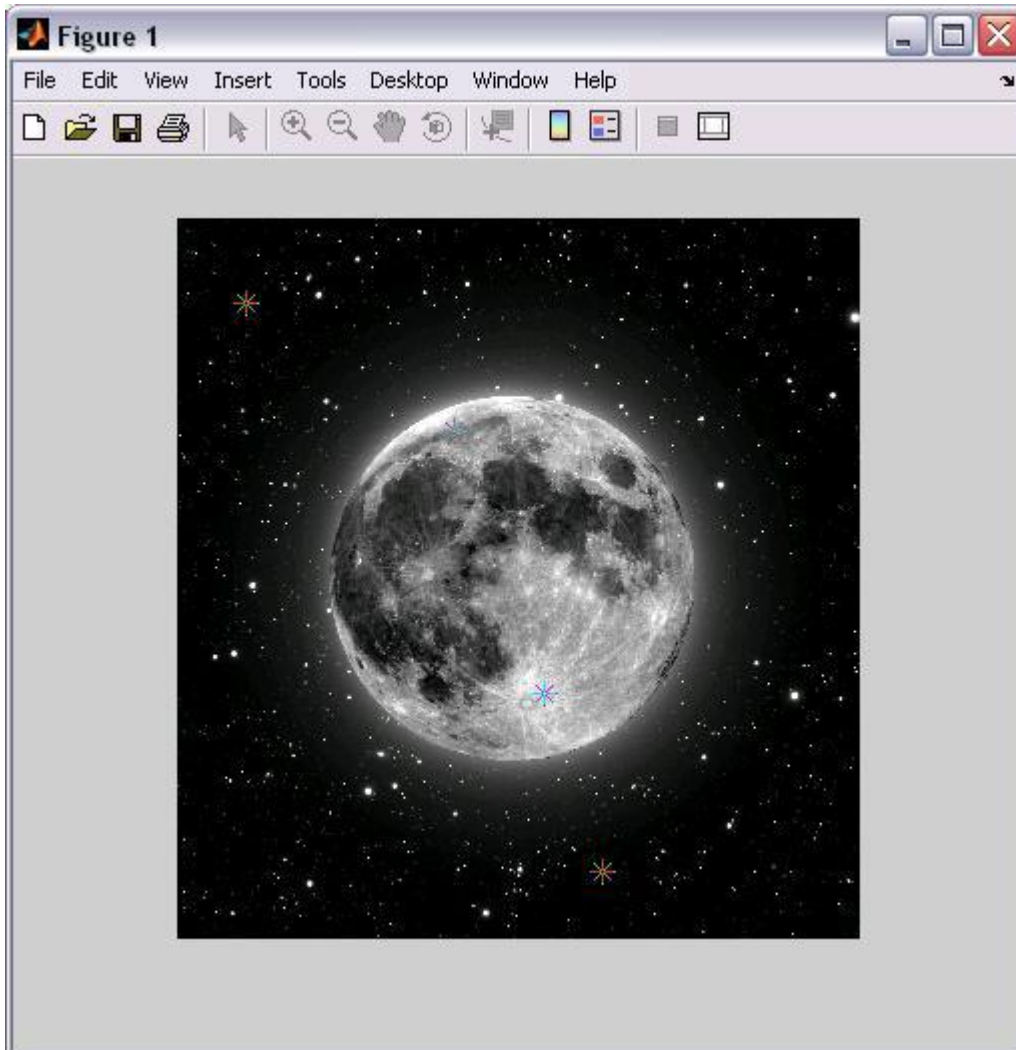
Υπολογιστικές ρυθμίσεις για Περιφέρειες μιας Εικόνας

Εμφάνιση μιας εικόνας imshow moon.jpg

Καλέστε το `imshow`. Όταν καλείται χωρίς ρυθμίσεις επί των εισροών, το `imshow` συντάσσεται με την εικόνα κατά τους τρέχοντες άξονες.

```
vals = imshow
```

Επιλέξτε τα σημεία που θέλετε να εξεταστεί η εικόνα
Κάνοντας κλικ με το ποντίκι. Το `imshow` τοποθετεί ένα αστέρι σε κάθε σημείο
επιλογής.



Όταν τελειώσετε την επιλογή των σημείων, πατήστε Enter.

```
vals = 39  
173 173 173  
216 216 216  
1 1 1  
1 1 1  
44 44 44
```

Εμφάνιση δεδομένων χωροπεριγράμματος εικόνας

Χρησιμοποιήστε τη λειτουργία `imshow` για την εμφάνιση χωροπεριγράμματος από τα δεδομένα σε εικόνα γκρι κλίμακας.

Παράδειγμα:

Αυτή η εντολή διαβάζει μια γκρι εικόνα και την εμφανίζει

```
I = imread('moon.jpg');
```

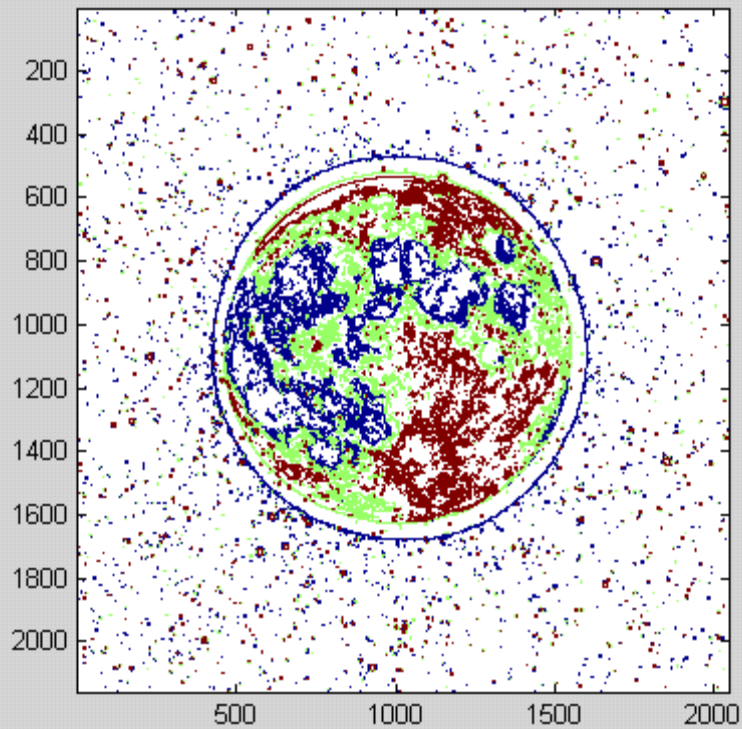
```
imshow(I)
```

original



Εμφανίζεται ένας χώρος μιας εικόνας με περίγραμμα του γκρι.

figure, imcontour(1,3)



Δημιουργία Ιστογράμματος εικόνας

Το ιστόγραμμα μιας εικόνας είναι ένα γράφημα που δείχνει την κατανομή της έντασης σε ένα δείκτη ή τις διαβαθμίσεις του γκρι στην εικόνα. Για τη δημιουργία ενός ιστογράμματος μιας εικόνας, χρησιμοποιήστε την `imhist` λειτουργία. Αυτή η λειτουργία δημιουργεί ένα ιστόγραμμα με `n` κλάδους, όπου το καθένα παριστάνει μια περιοχή τιμών των δεδομένων. Αυτό τότε υπολογίζει τον αριθμό των pixels σε κάθε περιοχή.

Παράδειγμα:

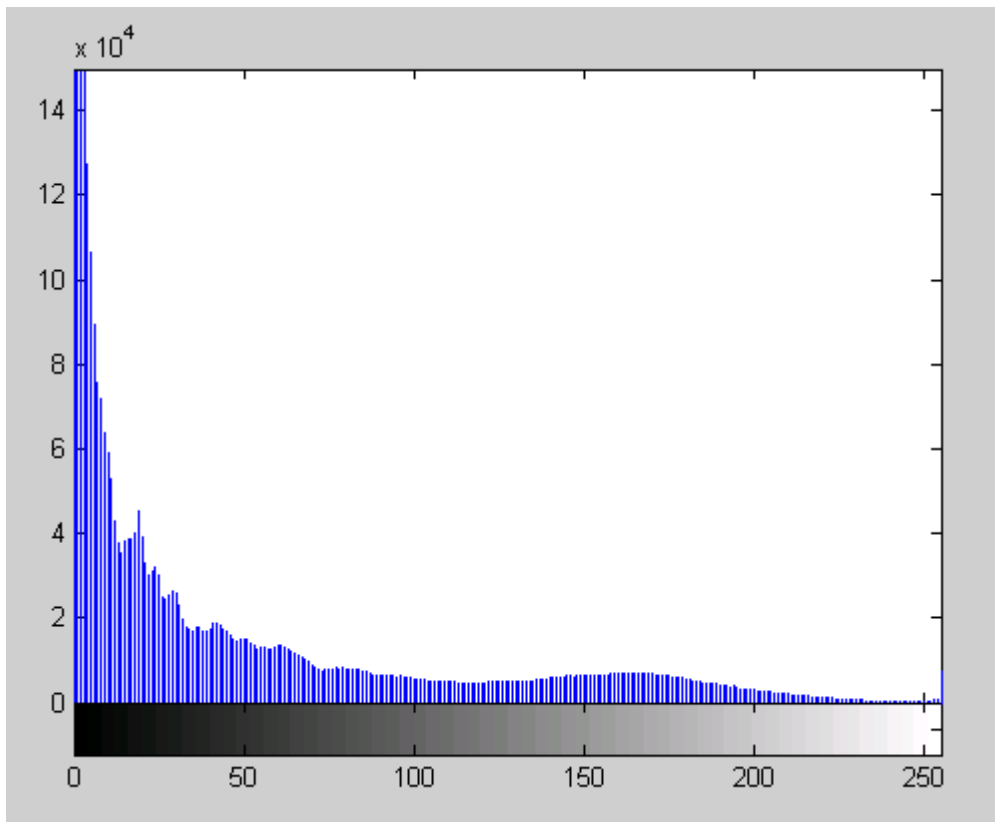
Read image and display it.

```
I = imread('moon.jpg');
```

```
imshow(I)
```



Display histogram of image.
`figure, imhist(I)`



Ανάλυση εικόνας

Αυτή η ενότητα περιγράφει τις τεχνικές ανάλυσης εικόνας που επιστρέφουν πληροφορίες σχετικά με τη δομή μιας εικόνας.

Ανίχνευση άκρων

Σε μια εικόνα, μια ακμή είναι μια καμπύλη που ακολουθεί μια πορεία ταχείας μεταβολής της έντασης της εικόνας. Τα άκρα είναι συχνά συνδεδεμένα με τα όρια των αντικειμένων σε μια σκηνή.

Η ανίχνευση άκρων χρησιμοποιείται για να προσδιορίσει τα άκρα σε μια εικόνα.

Η πιο δυναμική μέθοδος ανίχνευσης άκρων που παρέχει είναι η μέθοδος Canny. Η μέθοδος Canny διαφέρει από τις άλλες μεθόδους ανίχνευσης άκρης στο ότι χρησιμοποιεί δύο διαφορετικά κατώφλια (για τον εντοπισμό ισχυρών και ασθενών άκρων), και περιλαμβάνει τις αδύναμες άκρες στην έξοδο μόνο εφόσον είναι συνδεδεμένες σε ισχυρές άκρες. Η μέθοδος αυτή είναι λιγότερο πιθανό, από τις υπόλοιπες που μπορούν, να ξεγελαστούν από το θόρυβο, και πιο πιθανό να ανιχνεύσουν τα αληθινά αδύναμα άκρα.

Παράδειγμα:

Read image and display it.

```
I = imread('moon.jpg');
```

```
imshow(I)
```



Εφαρμόστε τους ανιχνευτές άκρων Sobel και Canny για την εικόνα και αποικονήστε τους.

```
BW1 = edge(I,'sobel');
```

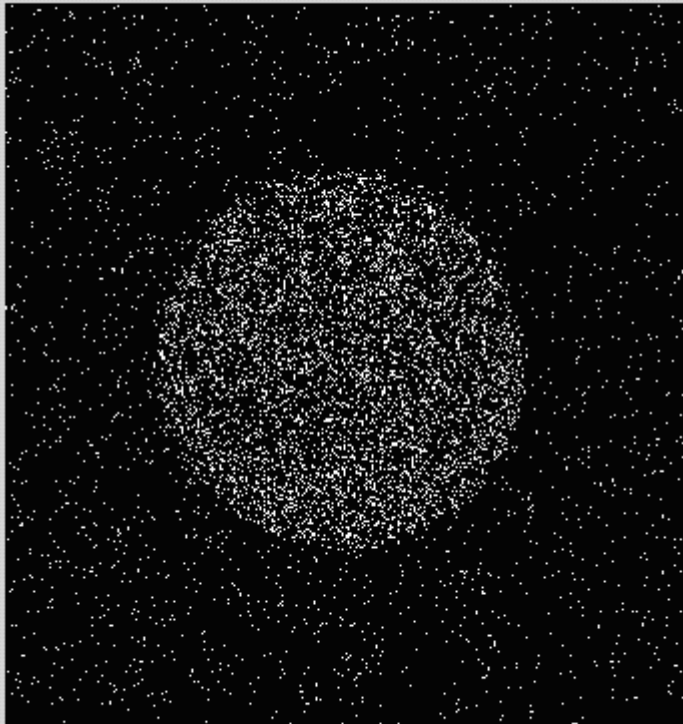
```
BW2 = edge(I,'canny');
```

```
imshow(BW1)
```

```
figure, imshow(BW2)
```




SOBEL FILTER



CANNY FILTER

Εντοπισμός ορίων

Η εργαλειοθήκη περιλαμβάνει δύο λειτουργίες που μπορούν να χρησιμοποιηθούν για να βρεθούν τα όρια των αντικειμένων σε μια δυαδική εικόνα:

`bwtraceboundary`

`bwboundaries`

Η λειτουργία `bwtraceboundary` επιστρέφει τις συντεταγμένες της γραμμής και της στήλης όλων των pixels στα σύνορα ενός αντικειμένου σε μια εικόνα. Πρέπει να καθοριστεί η θέση των συνόρων του pixel στο αντικείμενο ως το σημείο εκκίνησης για την παρακολούθηση.

Η `bwboundaries` συνάρτηση επιστρέφει τις συντεταγμένες της γραμμής και της στήλης των συνόρων των pixels όλων των αντικειμένων σε μια εικόνα.

Παράδειγμα:

Read image and display it.

```
I = imread('moon.jpg');
```

```
imshow(I)
```



Μετατρέψτε την εικόνα σε μια δυαδική εικόνα. Οι `bwtraceboundary` και `bwboundaries` λειτουργούν μόνο με δυαδικές εικόνες.

```
BW = im2bw (I)?
```

imshow (BW)



Καλέστε την `bwtraceboundary` να ανιχνεύσει τα όρια καθορισμένου σημείου. Ως απαιτούμενες συνιστώσες, πρέπει να καθορισθεί μια δυαδική εικόνα, οι συντεταγμένες γραμμής και στήλης του σημείου εκκίνησης, καθώς και η κατεύθυνση του πρώτου βήματος.

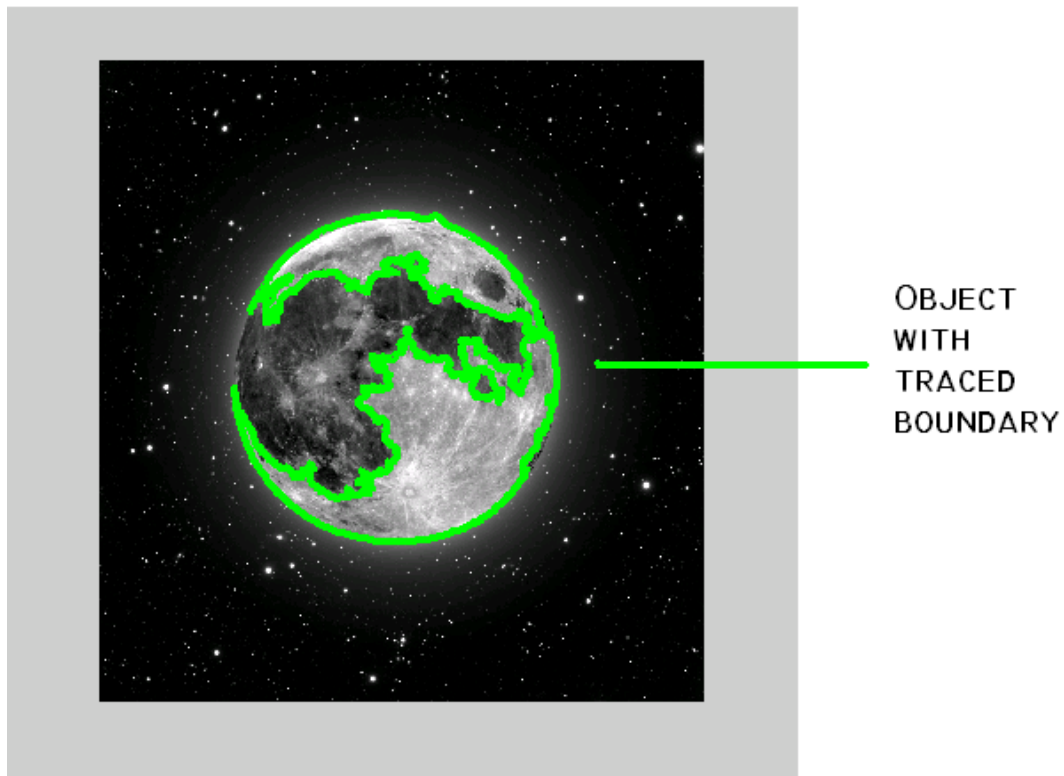
```
boundary = bwtraceboundary(BW,[row, col],'N');
```

Εμφάνιση της αρχικής κλίμακας του γκρι μιας εικόνας και χρησιμοποίηση των συντεταγμένων που επιστρέφονται από `bwtraceboundary` για τη σχεδίαση των συνόρων της εικόνας.

```
imshow(I)
```

```
hold on;
```

```
plot(boundary(:,2),boundary(:,1),'g','LineWidth',3);
```

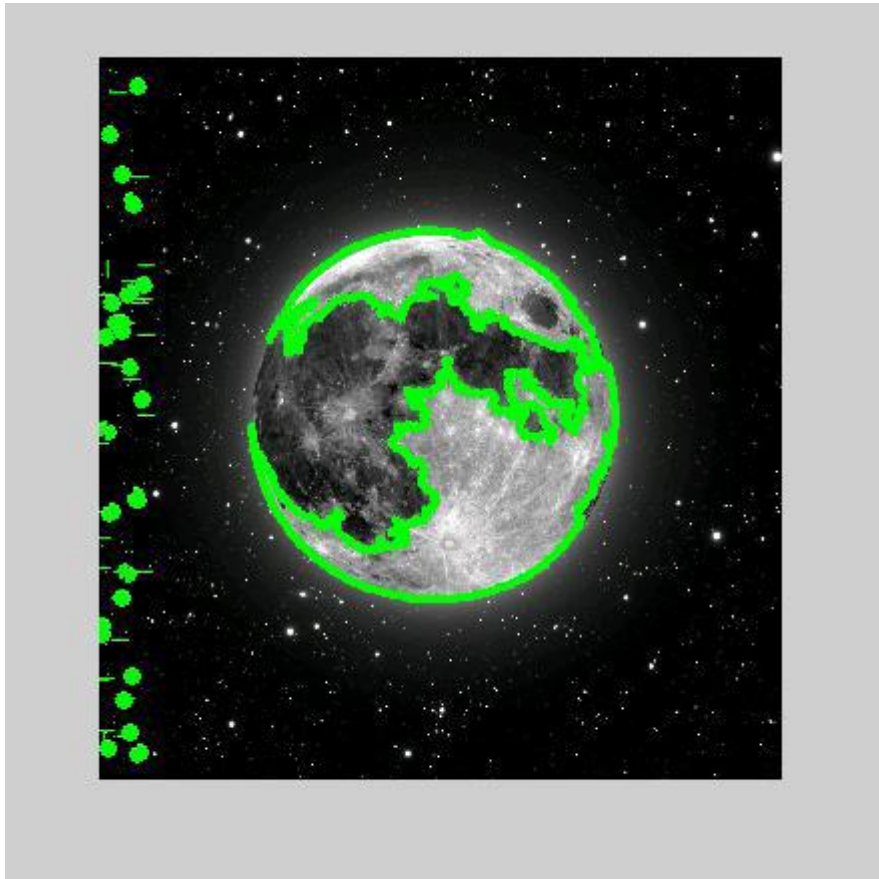


Για να εντοπιστούν τα όρια του συνόλου των σημείων στην εικόνα, χρησιμοποιήστε την λειτουργία `bwboundaries`.

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

Σχεδίαση των συνόρων όλων των σημείων για την αρχική κλίμακα του γκρι της εικόνας χρησιμοποιώντας τις συντεταγμένες που επιστρέφονται από `bwboundaries`.

```
for k=1:100  
b = boundaries{k};  
plot(b(:,2),b(:,1),'g','LineWidth',3);  
end
```



Ανίχνευση γραμμών με χρήση του μετασχηματισμού Hough

Το Image Processing Toolbox περιλαμβάνει λειτουργίες που υποστηρίζουν το μετασχηματισμό Hough.

Hough

houghpeaks

houghlines

Διαβάστε μια εικόνα στο χώρο εργασίας του MATLAB.

```
I = imread('moon.jpg');
```



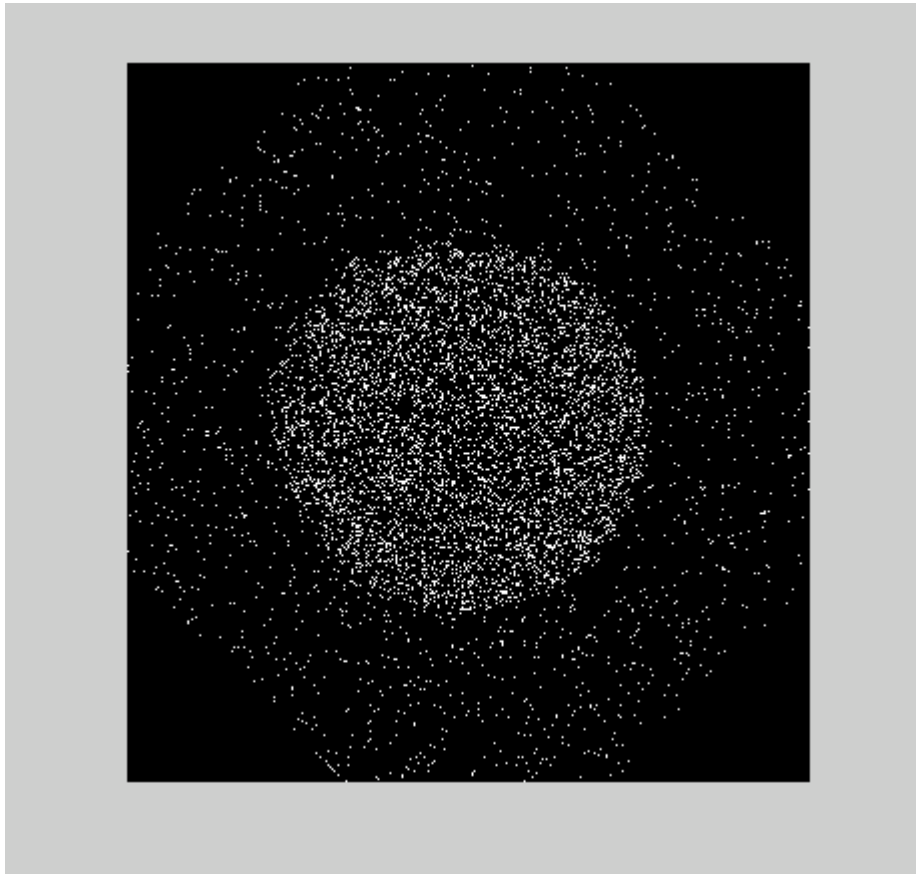
Περιστροφή και περικοπή της εικόνας.

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



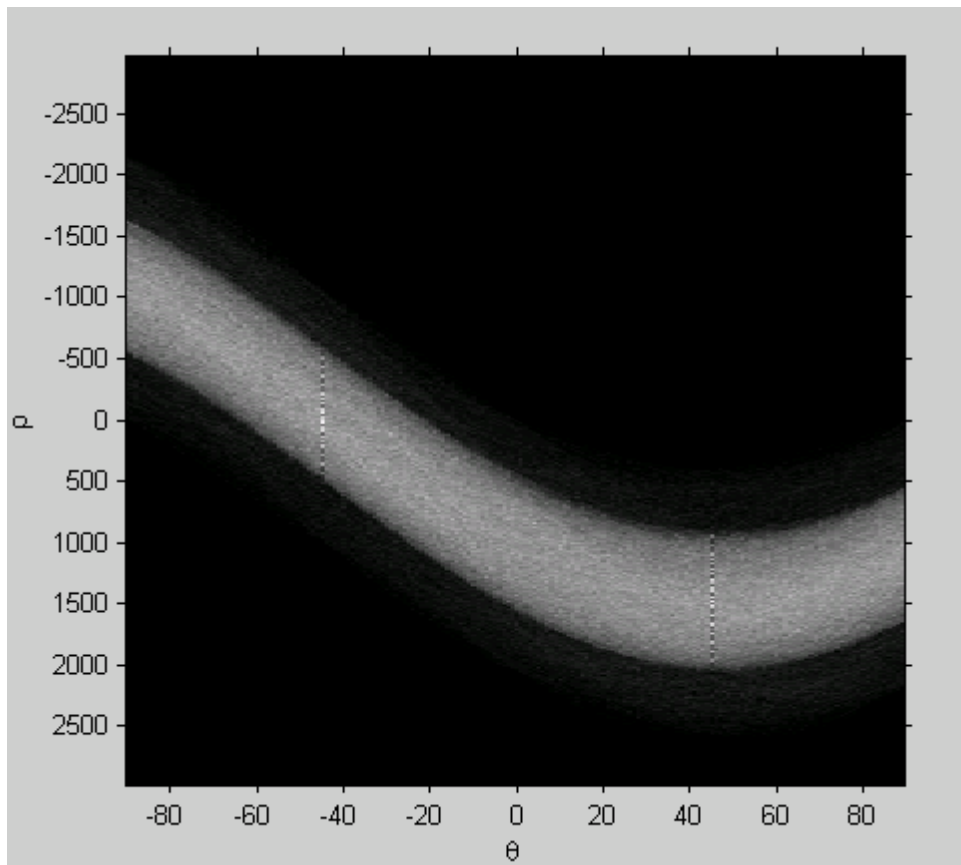
Βρείτε τα άκρα της εικόνας.

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



Υπολογίζουμε το μετασχηματισμό Hough της εικόνας χρησιμοποιώντας τη λειτουργία Hough.
`[H,theta,rho] = hough(BW);`

Εμφανίζοντας το μετασχηματισμό.
`imshow(H,[],'XData',theta,'YData',rho,...
'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;`



Βρίσκουμε τις κορυφές του μετασχηματισμού της μήτρας Hough, H, χρησιμοποιώντας τη λειτουργία `houghpeaks`.

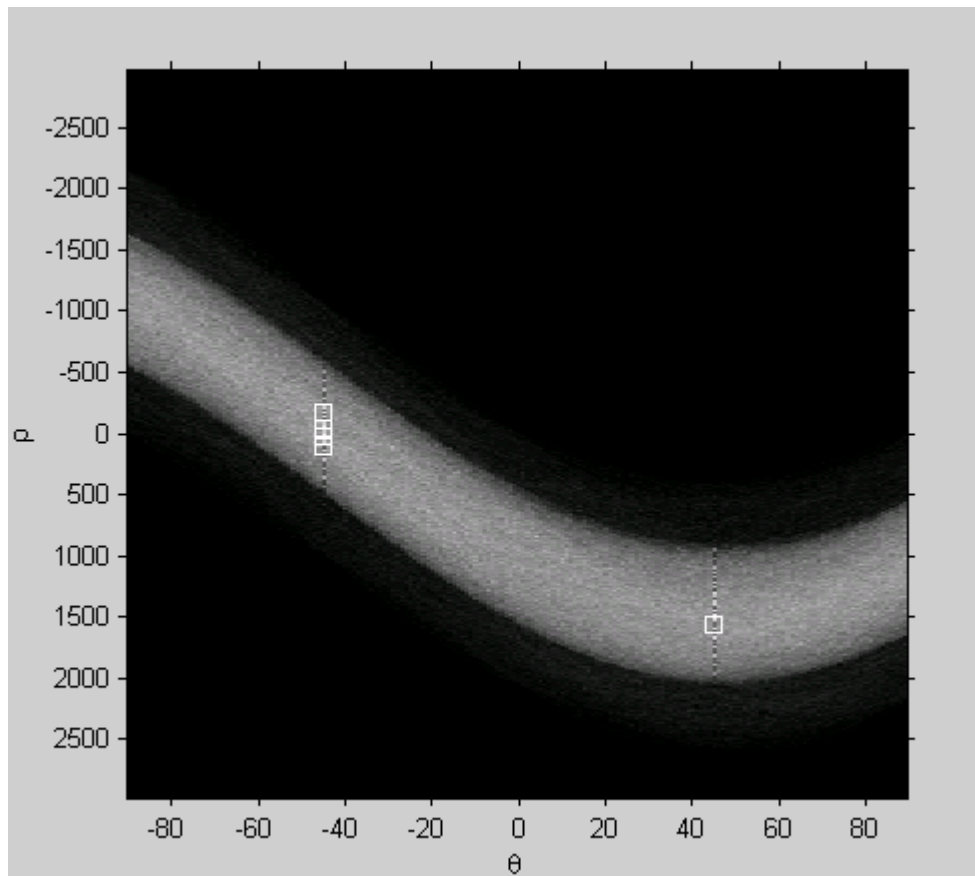
```
P = houghpeaks(H,5,'threshold',ceil(0.3*max(H(:))));
```

Plot the peaks.

```
x = theta(P(:,2));
```

```
y = rho(P(:,1));
```

```
plot(x,y,'s','color','white');
```



Βρείτε τις γραμμές της εικόνας.

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

Δημιουργήστε ένα σχέδιο που επιβάλλει τις γραμμές της αρχικής εικόνας.

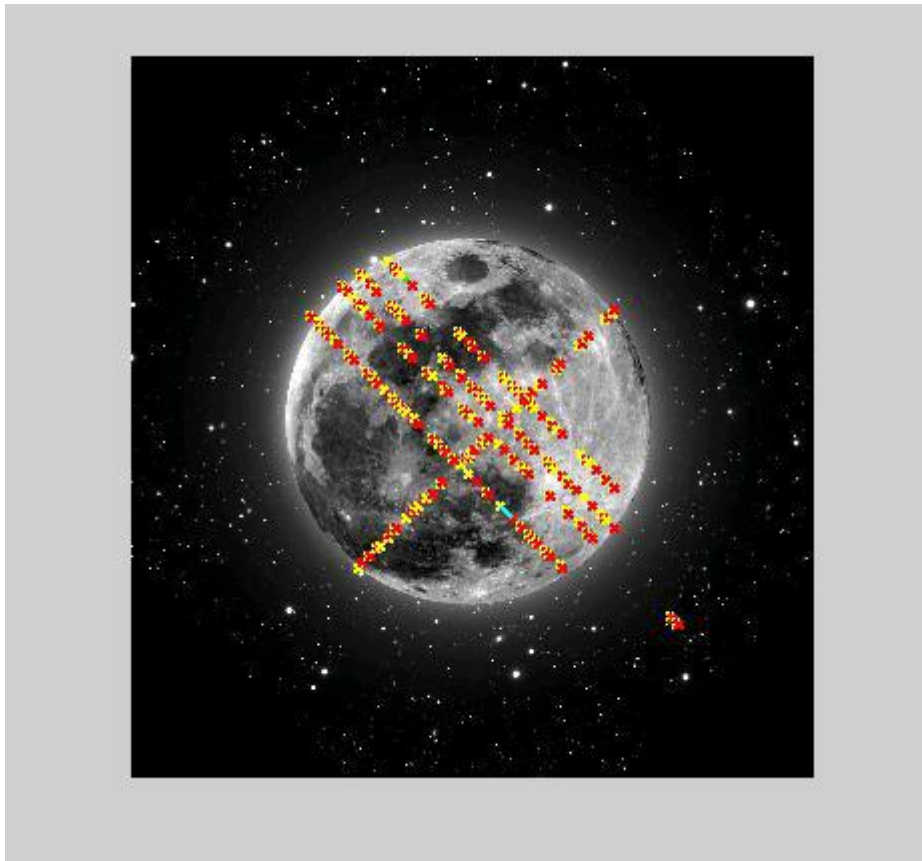
```
figure, imshow(rotI), hold on
max_len = 0;
for k = 1:length(lines)
xy = [lines(k).point1; lines(k).point2];
plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
% Plot beginnings and ends of lines
plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
% Determine the endpoints of the longest line segment
len = norm(lines(k).point1 - lines(k).point2);
if ( len > max_len)
max_len = len;
xy_long = xy;
end
end
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','cyan');
```

```
len = norm(lines(k).point1 - lines(k).point2);
if ( len > max_len)
max_len = len;
```

```

xy_long = xy;
end
end
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','cyan');

```



Ανάλυση υφής εικόνας

Η εργαλειοθήκη υποστηρίζει ένα σύνολο λειτουργιών που μπορεί να χρησιμοποιηθεί για την ανάλυση της υφής. Η ανάλυση υφής χρησιμοποιείται σε μια ποικιλία εφαρμογών, συμπεριλαμβανομένων της τηλεπισκόπησης, του αυτοματοποιημένου έλεγχου, και της επεξεργασίας ιατρικής εικόνας. Η ανάλυση της υφής μπορεί να χρησιμοποιηθεί για να βρεθεί η υφή των συνόρων, που ονομάζεται υφή κατάτμησης. Η "ανάλυση υφής μπορεί να είναι χρήσιμη όταν τα αντικείμενα σε μια εικόνα χαρακτηρίζονται περισσότερο από την υφή τους, παρά από την ένταση, και οι παραδοσιακές τεχνικές κατωφλίου δεν μπορούν να χρησιμοποιηθούν αποτελεσματικά.

Λειτουργίες Υφής

Read in the image and display it.

```
I = imread('moon.jpg');
```

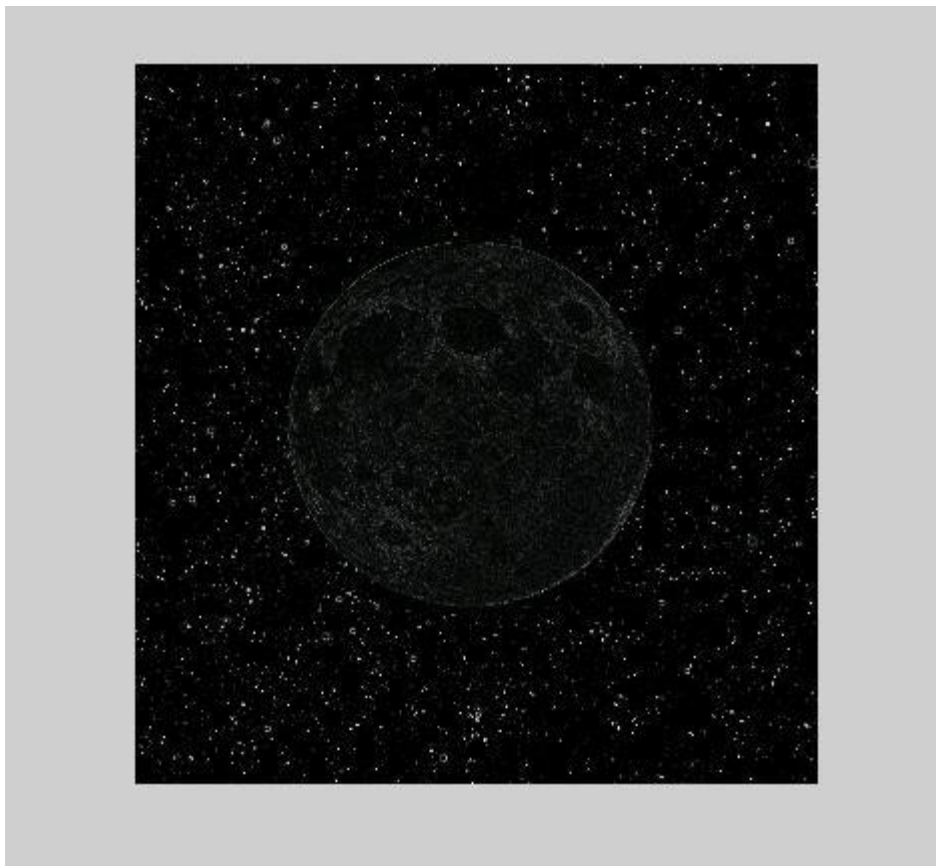
```
imshow(I)
```



Φιλτράρισμα της εικόνας με τη λειτουργία `rangefilt` και εμφάνιση των αποτελεσμάτων.

```
K = rangefilt(I);
```

```
figure, imshow(K)
```



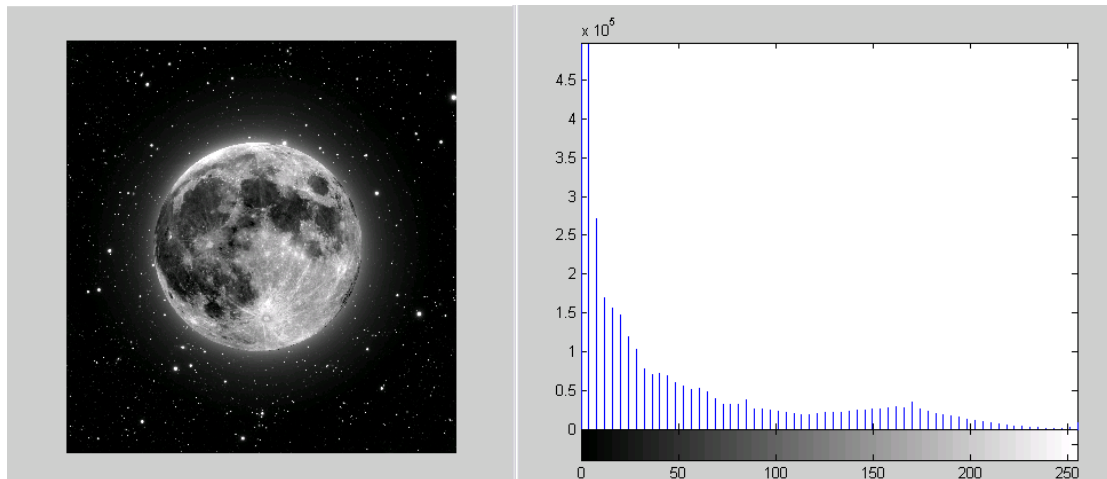
Προσαρμογή Έντασης

Προσαρμογή έντασης είναι μια τεχνική βελτίωσης της εικόνας που χαρτογραφεί τις τιμές της έντασης μιας εικόνας για τις τιμές της έντασης σε μια νέα σειρά. Παράδειγμα:

```
I = imread('moon.jpg');
```

```
imshow(I)
```

```
figure, imhist(I,64)
```



Προσαρμογή τιμών Έντασης σε ορισμένο πεδίο

Μπορεί να ρυθμιστεί η τιμή της έντασης σε μια εικόνα χρησιμοποιώντας τη λειτουργία `imadjust`, όπου μπορεί να καθοριστεί το εύρος των τιμών της έντασης της εικόνας εξόδου.

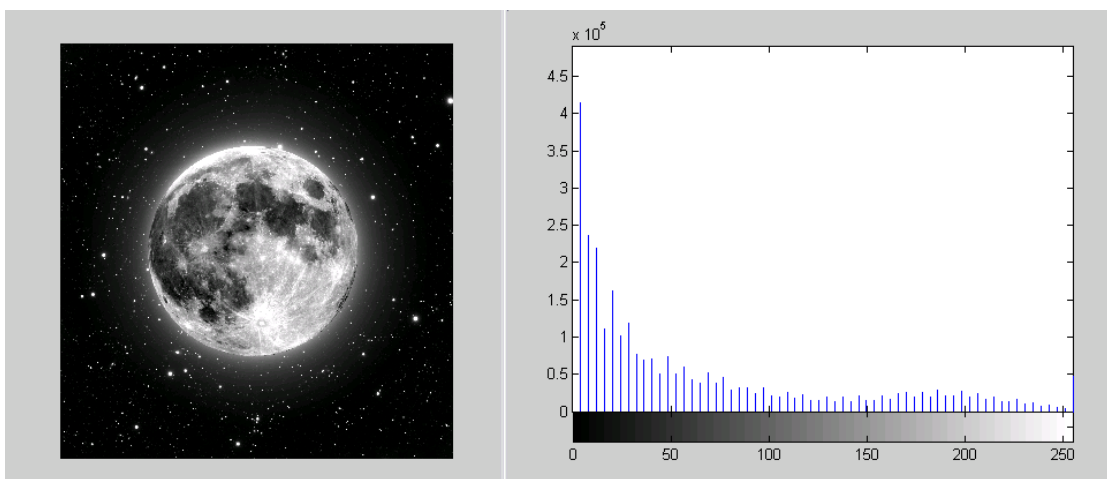
Παράδειγμα:

```
I = imread('moon.jpg');
```

```
J = imadjust(I);
```

```
imshow(J)
```

```
figure, imhist(J,64)
```



Εξειδίκευση των ορίων προσαρμογής

Μπορείτε να ορίσετε προαιρετικά το εύρος των τιμών των εισροών και τις τιμές της τελικής παραγωγής που χρησιμοποιούν `imadjust`. Μπορείτε να καθορίσετε αυτές τις σειρές σε δύο φορείς που θα περάσετε στο `imadjust` ως δεδομένα. Ο πρώτος φορέας προσδιορίζει τις χαμηλές και υψηλές τιμές έντασης που θέλετε να καθορίσετε. Το δεύτερο διάνυσμα καθορίζει την κλίμακα στην οποία θέλετε να αντιστοιχίσετε αυτά.

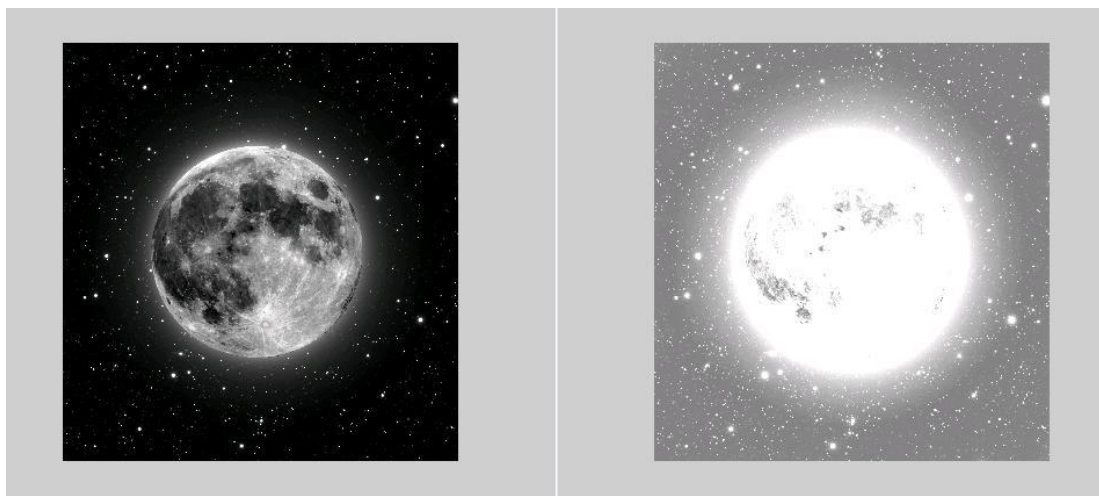
Παράδειγμα:

```
I = imread('moon.jpg');
```

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

```
imshow(I)
```

```
figure, imshow(J)
```



Εξίσωση Ιστογράμματος

Η διαδικασία προσαρμογής τιμών έντασης μπορεί να γίνει αυτόματα από τη λειτουργία histeq. Το histeq εκτελεί εξίσωση ιστογράμματος, η οποία συνεπάγεται στη μετατροπή των μέτρων έντασης έτσι ώστε το ιστόγραμμα της εικόνας εξόδου να ταιριάζει σε ένα συγκεκριμένο ιστόγραμμα.

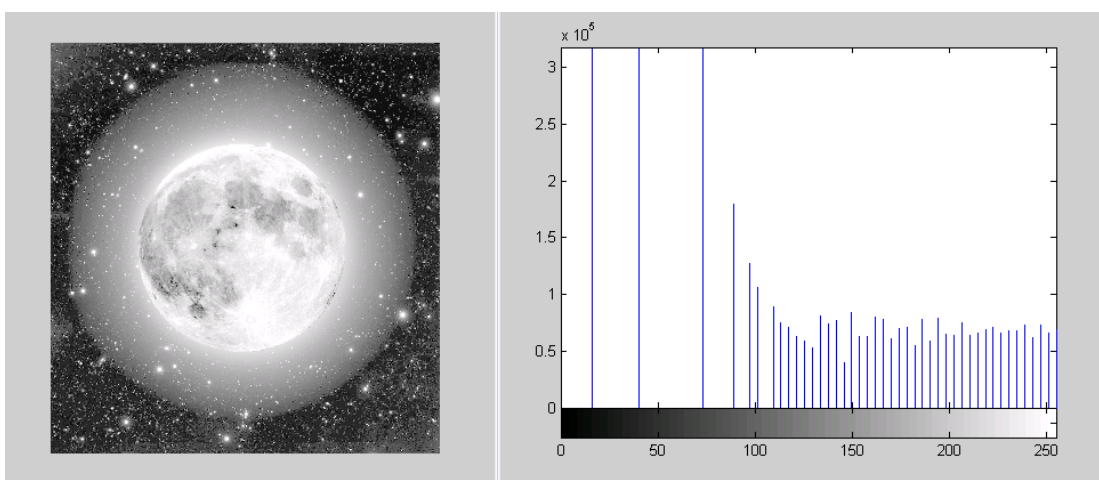
Παράδειγμα:

```
I = imread('moon.jpg');
```

```
J = histeq(I);
```

```
imshow(J)
```

```
figure, imhist(J,64)
```



Απαλοιφή θορύβου

Οι ψηφιακές εικόνες είναι επιρρεπείς σε μια μεγάλη ποικιλία τύπων θορύβου.

Ο θόρυβος είναι το αποτέλεσμα λαθών στην διαδικασία απόκτησης εικόνας η οποία έχει ως αποτέλεσμα οι τιμές των pixel να εμφανίζουν ψευδή ένταση της πραγματικής σκηνής. Υπάρχουν αρκετοί τρόποι με τους οποίους ο θόρυβος μπορεί να εισαχθεί σε μια εικόνα, ανάλογα με τον τρόπο της δημιουργίας της.

Φίλτρο μέσου

Διάμεσο φιλτράρισμα είναι παρόμοιο με τη χρήση του φίλτρου μέσου όρου, σε αυτό, κάθε pixel εξόδου ορίζεται ένας μέσος όρος τιμών pixel στη γειτονιά του αντίστοιχου κυττάρου εικόνας εισαγωγής.

Παράδειγμα:

Διάβαση και απεικόνιση μιας εικόνας.

```
I = imread('moon.jpg');
```

```
imshow(I)
```



Προσθέστε θόρυβο

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

```
figure, imshow(J)
```




Φιλτράρετε την εικόνα με το θόρυβο και εξάγετε τα αποτελέσματα.

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

```
figure, imshow(K)
```



Τώρα χρησιμοποιείτε ένα διάμεσο φίλτρο για την εικόνα με το θόρυβο και δείτε τα αποτελέσματα. Παρατηρήστε ότι η λειτουργία `medfilt2` κάνει καλύτερη δουλειά στην άρση του θορύβου, με λιγότερη σύγχυση των άκρων.

```
L = medfilt2(J,[3 3]);
```

```
figure, imshow(K)
```

```
figure, imshow(L)
```



Επεξεργασία ROI

Αυτό το κεφάλαιο περιγράφει πώς να καθοριστεί μια περιοχή ενδιαφέροντος (ROI) και εκτελεί την επεξεργασία ROI που καθορίζουμε.

Επιλογή πεδίου ενδιαφέροντος (ROI)

Μια περιοχή ενδιαφέροντος (ROI) είναι ένα τμήμα μιας εικόνας που θέλουμε να φιλτράρουμε ή να εκτελέσουμε κάποια άλλη ενέργεια. Μπορεί να καθοριστεί μια ROI με τη δημιουργία μιας δυαδικής μάσκας, η οποία είναι μια δυαδική εικόνα που έχει το ίδιο μέγεθος με την εικόνα που θέλουμε να επεξεργαστούμε με εικονοστοιχεία που καθορίζουν τις επιλογές ROI σε 1 και όλα τα άλλα pixels σε 0.

Επιλογή πολυγωνικού πεδίου ενδιαφέροντος ROI διαδραστικά

Χρησιμοποιήστε τη λειτουργία `roipoly` για να τον ορισμό ενός πολυγωνικού ROI διαδραστικά σε μια συγκεκριμένη εικόνα. Για να το κάνετε αυτό, εμφανίστε μια εικόνα, χρησιμοποιώντας το `imshow` και στη συνέχεια καλέστε την `roipoly` χωρίς ρυθμίσεις εισροών. Όταν μετακινείτε τον κέρσορα πάνω από την εικόνα που εμφανίζεται στους τρέχοντες άξονες, ο δρομέας αλλάζει στο `crosshairs` σχήμα, μπορείτε να καθορίσετε έπειτα τις κορυφές του πολυγώνου κάνοντας κλικ σε σημεία της εικόνας με το ποντίκι.

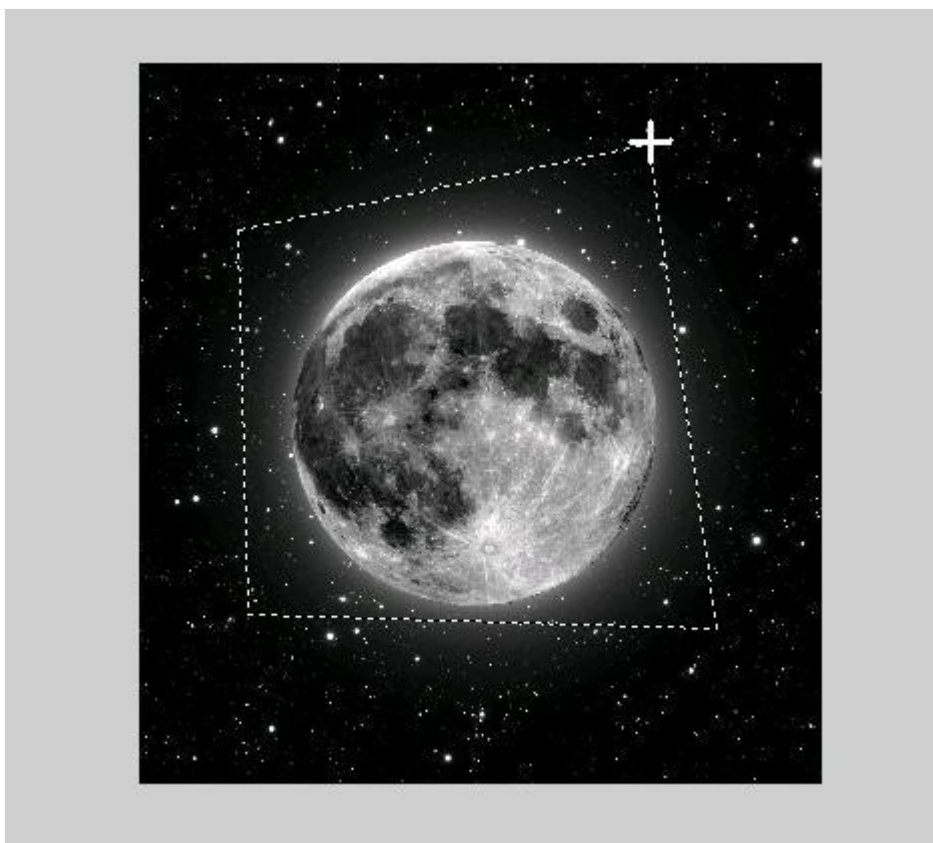
Όταν τελειώσετε με την επιλογή των κορυφών, πιάστε `Return`. Η `roipoly` επιστρέφει μια δυαδική εικόνα του ίδιου μεγέθους με την εικόνα εισαγωγής, που περιέχει την τιμή 1 μέσα στο συγκεκριμένο πολύγωνο, και 0 για οπουδήποτε αλλού.

Παράδειγμα:

```
I = imread('moon.jpg');
```

```
imshow(I)
```

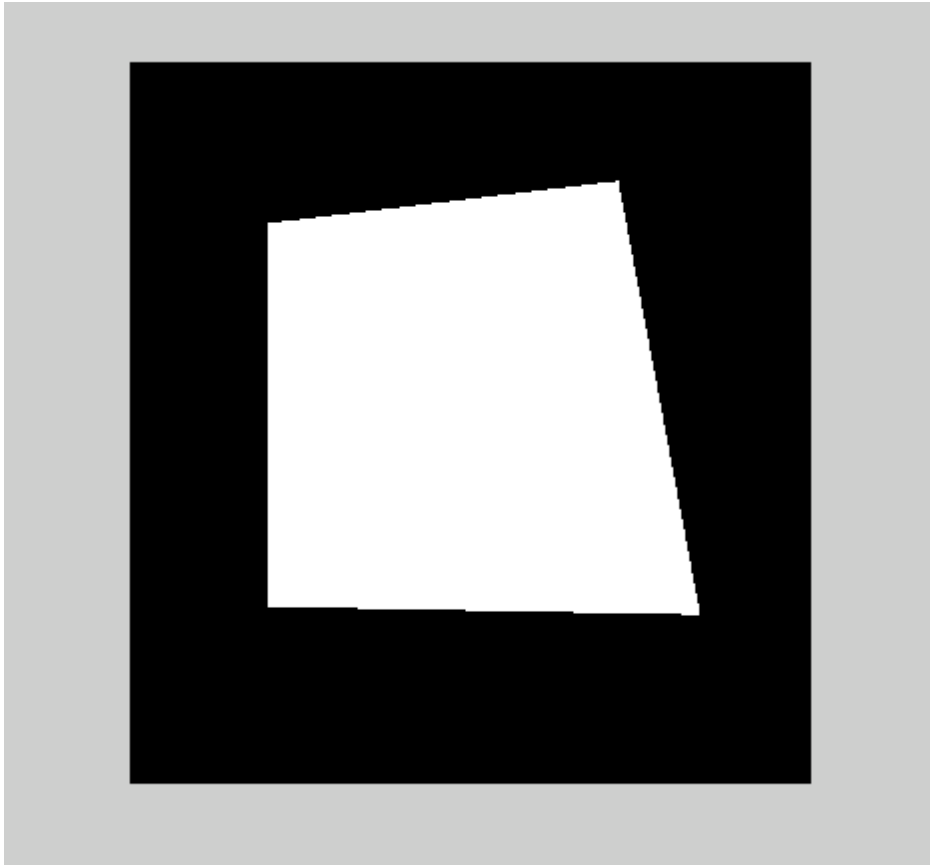
```
BW = roipoly;
```



Μετά την επιλογή πατήστε Enter και γράψτε

`imshow (BW)`

και θα δείτε κάτι παρόμοιο



Φιλτράρισμα ROI

Φιλτράροντας μια περιοχή ενδιαφέροντος (ROI) είναι η διαδικασία της εφαρμογής ενός φίλτρου σε μια περιοχή εικόνας, όπου μια δυαδική μάσκα καθορίζει την περιοχή. Για παράδειγμα, μπορείτε να εφαρμόσετε ένα φίλτρο έντασης προσαρμογής σε ορισμένες περιφέρειες της εικόνας.

Φιλτράρισμα πεδίου εικόνας

Το παράδειγμα αυτό χρησιμοποιεί μάσκα φιλτραρίσματος για αύξηση της αντίθεσης από μια συγκεκριμένη περιοχή εικόνας
Διαβάστε στην εικόνα.

```
I = imread('moon.jpg');
```

Δημιουργήστε τη μάσκα.

```
imshow(I)
```

```
BW = roioly;
```

Δημιουργήστε το φίλτρο.

```
h = fspecial('unsharp');
```

Καλέστε το **roifilt2**, προσδιορίζοντας την εικόνα που πρέπει να φιλτράρετε, τη μάσκα, και το φίλτρο.

```
I2 = roifilt2(h,I,BW);
```

```
imshow(I)
```

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



Καθαρισμός εικόνας με κανονικό φίλτρο

Χρησιμοποιήστε τη λειτουργία `deconvreg` για την αποθολώση εικόνας με ένα κανονικό φίλτρο. Ένα κανονικό φίλτρο μπορεί να είναι αποτελεσματικό όταν υπάρχουν περιορισμένες πληροφορίες για τον προσθετικό θόρυβο.

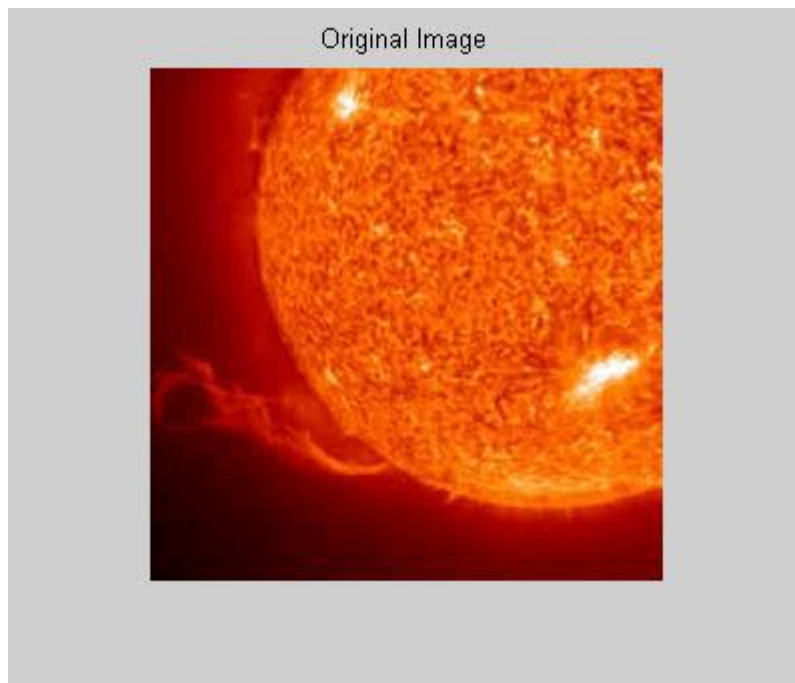
Παράδειγμα:

Διαβάστε μια εικόνα στο χώρο εργασίας του MATLAB.

```
I = imread('sun.jpg');
```

```
I = I(125+[1:256],1:256,:);
```

```
figure; imshow(I); title('Original Image');
```



Δημιουργήστε το PSF.

```
PSF = fspecial('gaussian',11,5);
```

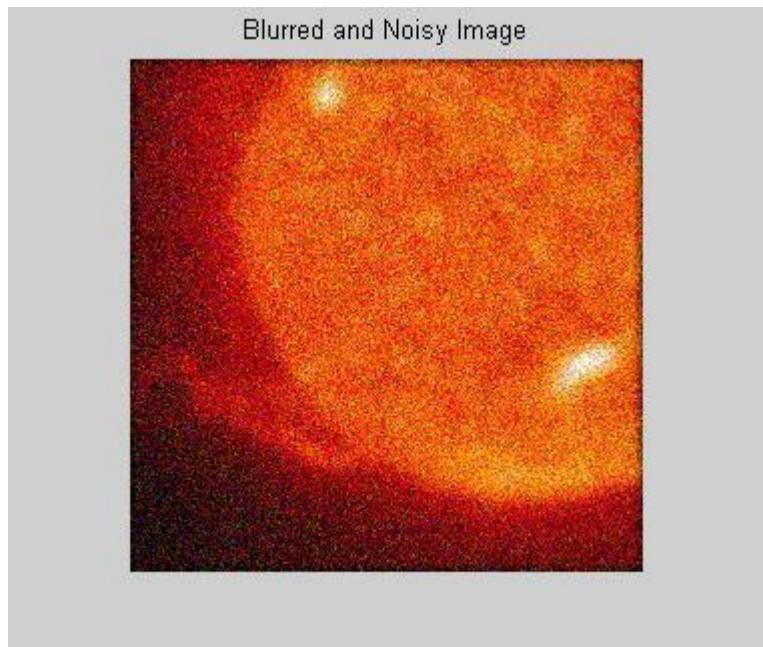
Δημιουργήστε μια εικονική θαμπάδα στην εικόνα και προσθέστε θόρυβο.

```
Blurred = imfilter(I,PSF,'conv');
```

```
V = .02;
```

```
BlurredNoisy = imnoise(Blurred,'gaussian',0,V);
```

```
figure;imshow(BlurredNoisy);title('Blurred and Noisy Image');
```

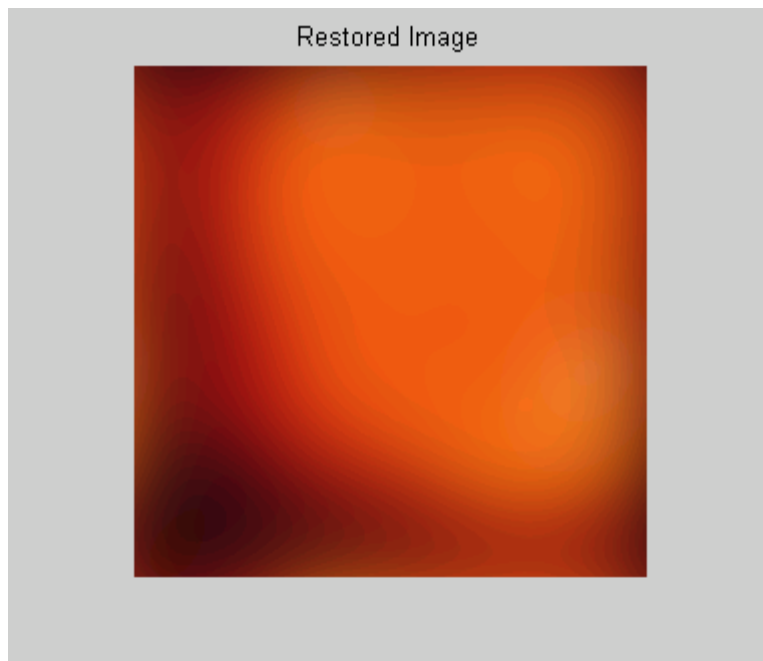


Χρησιμοποιήστε το `deconvreg` να αποθολώσετε την εικόνα, διευκρινίζοντας τις PSF για να δημιουργηθεί η θαμπάδα και την ισχύ του θορύβου, NP.

```
NP = V*prod(size(I));
```

```
[reg1 LAGRA] = deconvreg(BlurredNoisy,PSF,NP);
```

```
figure,imshow(reg1),title('Restored Image');
```



Καθαρισμός εικόνας με αλγόριθμο τυφλής αποσυνέλιξης

Η λειτουργία `deconvblind`, όπως και η λειτουργία `deconvlucy`, υλοποιεί διάφορες προσαρμογές στον αρχικό αλγόριθμο μέγιστης πιθανοφάνειας Lucy-Richardson, η διεύθυνση είναι περίπλοκη εργασία για την αποκατάσταση της εικόνας.

Χρήση λειτουργίας αποσυνέλιξης για το καθαρισμό εικόνας

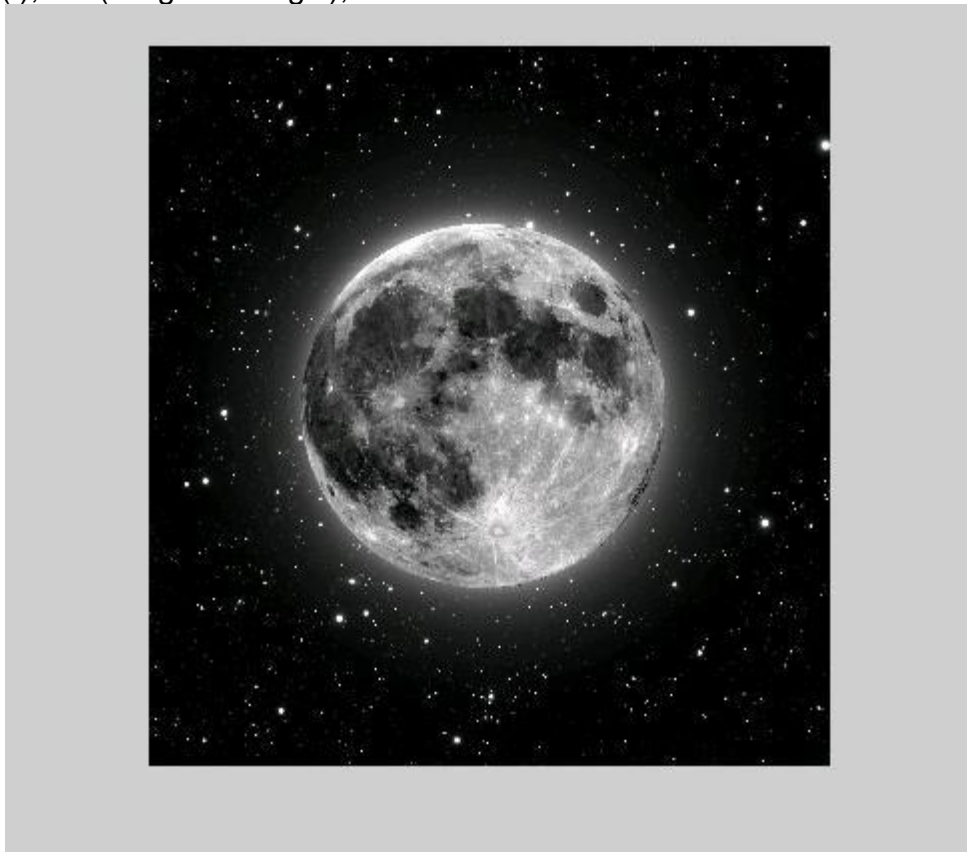
Για να φανεί η τυφλή αποσυνέλιξη, το παράδειγμα κάνει προσομοίωση θολής εικόνας και στη συνέχεια χρησιμοποιεί την λειτουργία `deconvblind` για την αποθόλωση αυτής.

Παράδειγμα:

Διαβάστε μια εικόνα στο χώρο εργασίας του MATLAB.

```
I = imread('moon1.jpg');
```

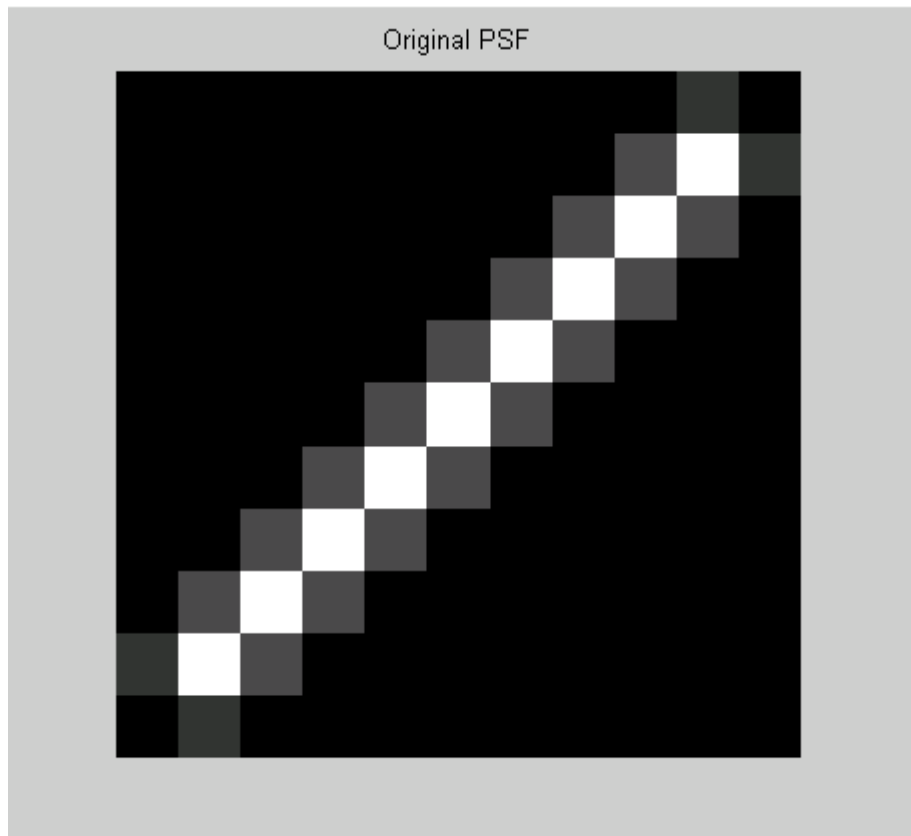
```
figure; imshow(I); title('Original Image');
```



Δημιουργήστε το PSF.

```
PSF = fspecial('motion',13,45);
```

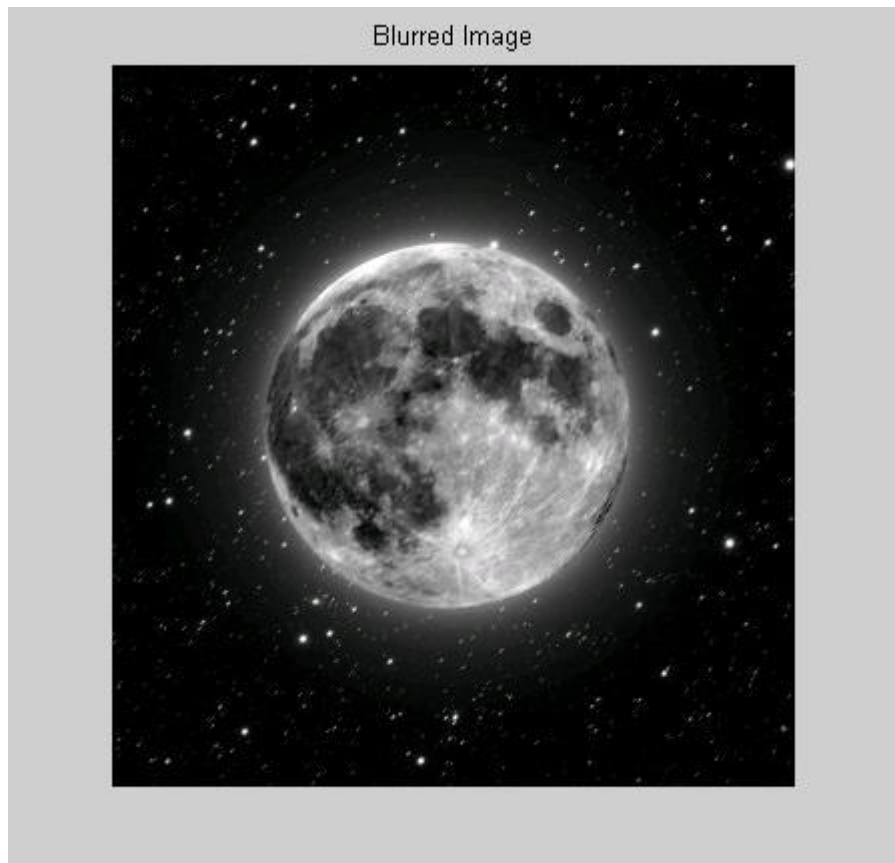
```
figure; imshow(PSF,[],'notruesize'); title('Original PSF');
```

Δημιουργήστε μια προσομοίωση θαμπάδας στην εικόνα.

```
Blurred = imfilter(I,PSF,'circ','conv');
```

```
figure; imshow(Blurred); title('Blurred Image');
```



Αποθολώστε την εικόνα, καθιστώντας την αρχική υπόθεση για το μέγεθος των PSF.

```
INITPSF = ones(size(PSF));
```

```
[J P]= deconvblind(Blurred,INITPSF,30);
```

```
figure; imshow(J); title('Restored Image');
```

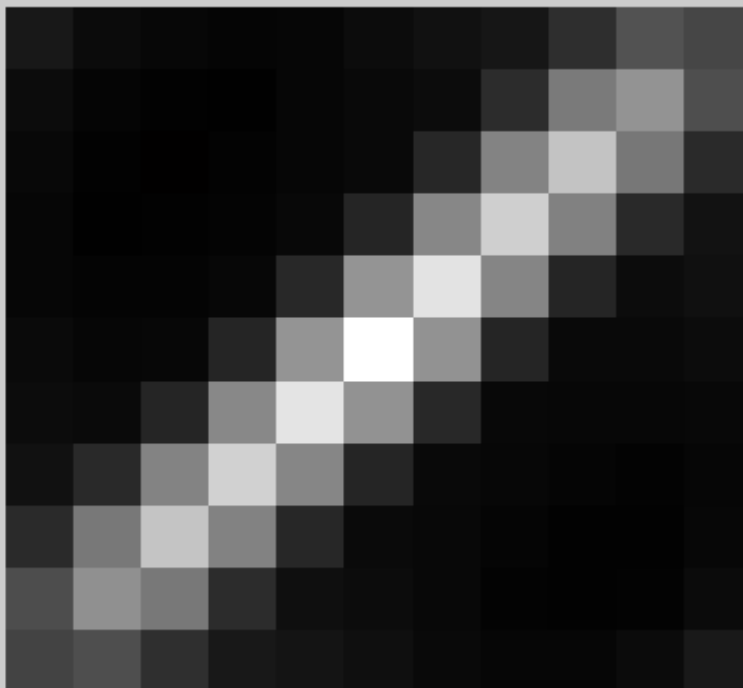
```
figure; imshow(P,[],'notruesize');
```

```
title('Restored PSF');
```

Restored Image



Restored PSF



Συρόμενες γειτονικές εφαρμογές

Μια συρόμενη εφαρμογή γειτονιάς είναι μια λειτουργία που εκτελείται για ένα pixel κάθε φορά, με την αξία οποιουδήποτε pixel της εικόνας εξόδου που καθορίζεται από τη εφαρμογή ενός αλγορίθμου στις τιμές κάθε ανταποκρινόμενου pixel εισόδου στη γειτονιά. Η γειτονιά pixel είναι κάποιο σύνολο pixels, που ορίζεται από τις θέσεις του, η οποία ονομάζεται κεντρικό. Η γειτονιά είναι ένα ορθογώνιο μπλοκ, και κατά τη μετακίνηση από το ένα στοιχείο στο επόμενο σε μια μητρική εικόνα, το μπλοκ γειτονιά ολισθαίνει στην ίδια κατεύθυνση.

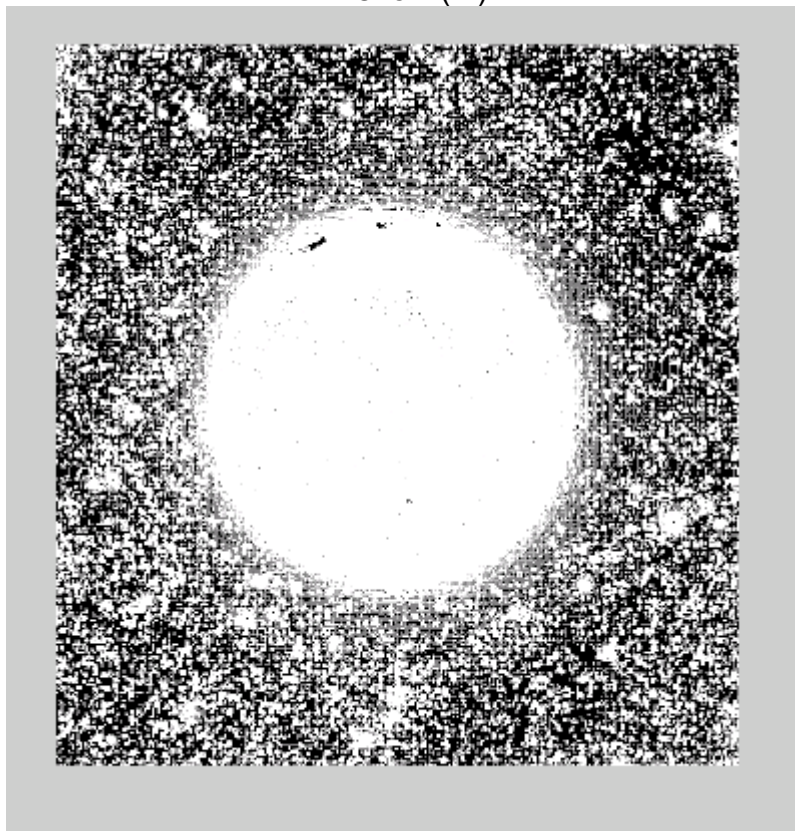
Εφαρμογή γραμμικού και μη-γραμμικού Φιλτραρίσματος

Μπορείτε να χρησιμοποιήσετε τις εργασίες ολίσθησης γειτονιάς για την εφαρμογή φιλτραρίσματος. Μπορείτε να χρησιμοποιήσετε τη λειτουργία `nfilter` για την εφαρμογή μιας ποικιλίας εργασιών ολίσθησης γειτονιάς. Το `nfilter` λαμβάνει τις ρυθμίσεις της εισόδου μιας εικόνας, ένα μέγεθος γειτονιάς, και μια συνάρτηση που επιστρέφει μια βαθμίδα. Επιστρέφοντας μια εικόνα του ίδιου μέγεθους με την εικόνα εισόδου. Η τιμή του κάθε pixel της εικόνας εξόδου υπολογίζεται με το πέρασμα του κάθε ανταποκρινόμενου pixel εισόδου στη γειτονιά και εν συνεχεία στη συνάρτηση.

```
I = imread («moon.jpg »)?
```

```
I2 = nfilter (I, [3 3], «std2 »)?
```

```
imshow (I2)
```



```
I = imread('moon.jpg');
```

```
f = @(x) max(x(:));
```

```
I2 = nfilter(I,[3 3],f);
```

```
imshow(I);
```

```
figure, imshow(I2);
```

