**Ψηφιακή́ Επεξεργασία και Ανάλυση Εικόνας**

Ακαδημαϊκό́ Έτος 2023-2024

**Εργαστηριακή́ Άσκηση - Μέρος Α**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

**Ονοματεπώνυμο:** Φωτάκης Ανδρέας **ΑΜ:** 1084674

**Ερώτηση 1.1**

Περιγραφή σχήματος

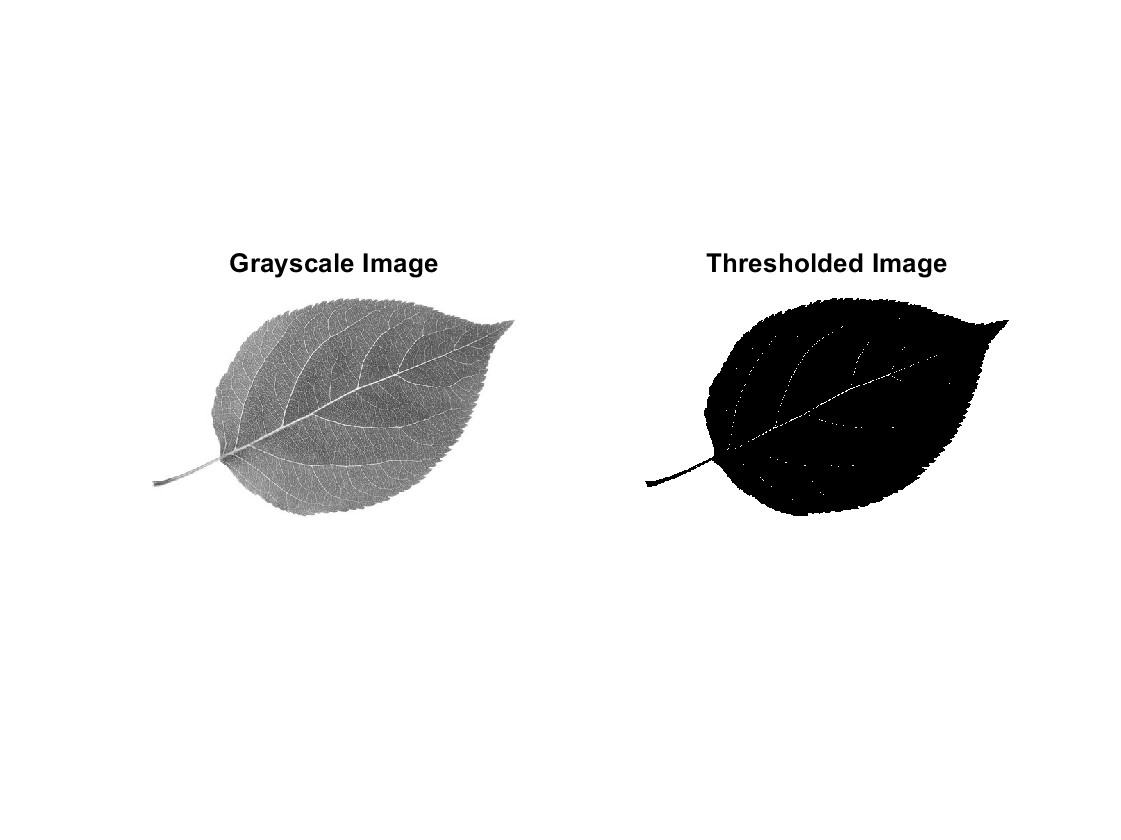
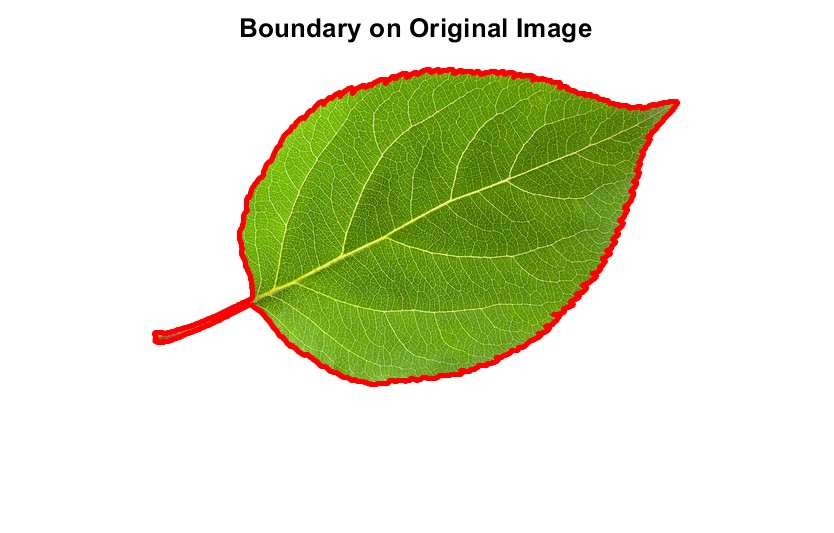
Βήμα 1

Figure 1: Αριστερά η αρχική εικόνα έχει μετατραπεί σε grayscale. Δεξιά η εικόνα έχει κατωφλιωθεί με τιμή κατωφλίου 𝜏 = 220.



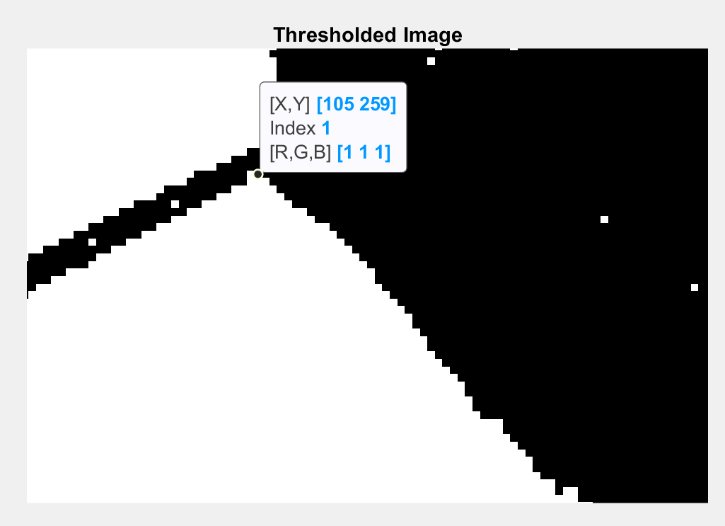
Βήμα 2

Figure 2: Η αρχική εικόνα με το περίγραμμα του σχήματος του φύλλου να απεικονίζεται με κόκκινο.

Για την εξαγωγή του περιγράμματος χρησιμοποιήθηκε η συνάρτηση της Matlab:

[B = bwtraceboundary(BW,P,fstep)](https://www.mathworks.com/help/images/ref/bwtraceboundary.html#d126e53398).

Η συνάρτηση αυτή απαιτεί για την εύρεση του περιγράμματος να δοθεί ένα σημείο ακριβώς δίπλα από ένα pixel στο περίγραμμα της εικόνας (P).

Για να βρούμε αυτό το σημείο μεγεθύνουμε στην κατωφλωμένη εικόνα και επιλέγουμε να δούμε τις συντεταγμένες του.

Βήμα 3

Οι περιγραφείς Fourier αναπαρίστανται ως μια ακολουθία συντεταγμένων (𝑥κ, 𝑦κ ) όπου

κ = 0,1, ... ,𝐾 – 1 και μιγαδικά ως:

Υπολογίζονται λοιπόν από τα σημεία του περιγράμματος ως εξής:

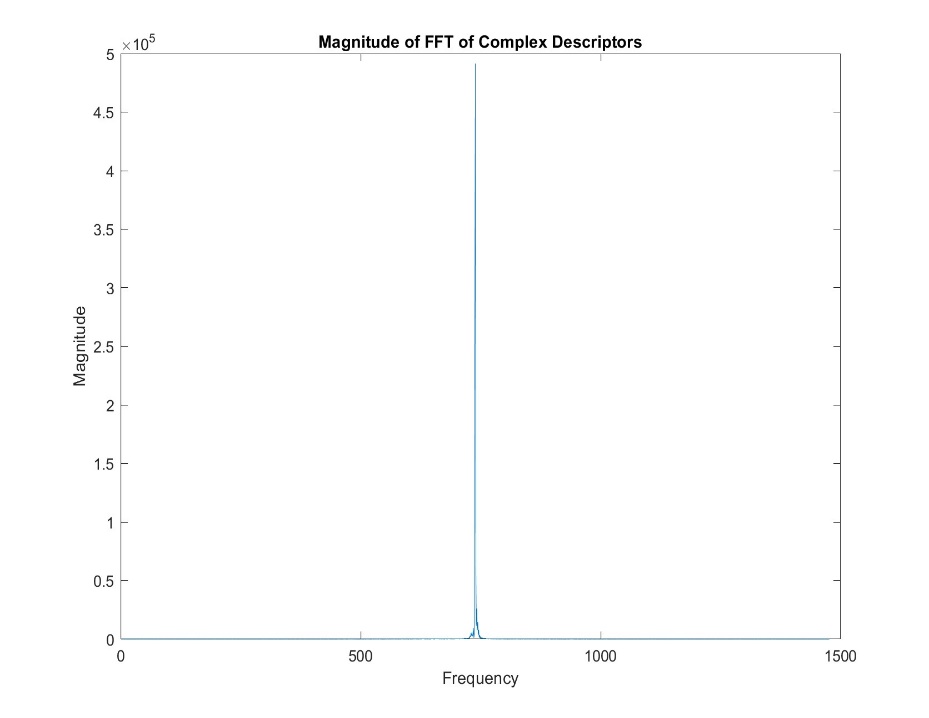
complex = boundaries(:, 2) + 1i \* boundaries(:, 1);

Μετατρέπουμε έτσι την δυσδιάστατη αναπαράσταση των σημείων του περιγράμματος 

σε μια μονοδιάστατη μιγαδική ακολουθία:

Στην συνέχεια υπολογίζουμε τον FFT των descriptors για να αποτυπώσουμε την χωρική πληροφορία στο πεδίο της συχνότητας:

complex\_descriptors = fftshift(fft(complex));

Η συνάρτηση fftshift() μετατοπίζει τη συνιστώσα μηδενικής συχνότητας στο κέντρο του πίνακα εξόδου, διασφαλίζοντας ότι η πληροφορία χαμηλής συχνότητας βρίσκεται στο κέντρο.

Παρατηρούμε ότι το συχνοτικό περιεχόμενο κατανέμεται στις χαμιλές συχνότητες, όπως αναμέναμε.

Βήμα 4

Οι προσεγγίσεις περιγράμματος μπορούν να προκύψουν με τη διαγραφή συντελεστών υψηλής συχνότητας.

Επειδή με τον fftshift() οι χαμηλές συχνότητες μετατοπίζονται στο κέντρο του πίνακα εξόδου και όχι γύρω από το 0 θα πρέπει να υπολογίσουμε το νέο αυτό κέντρο για να επιλέξουμε σωστά τις χαμηλές συχνότητες.

center\_index = numel(complex\_descriptors) / 2;

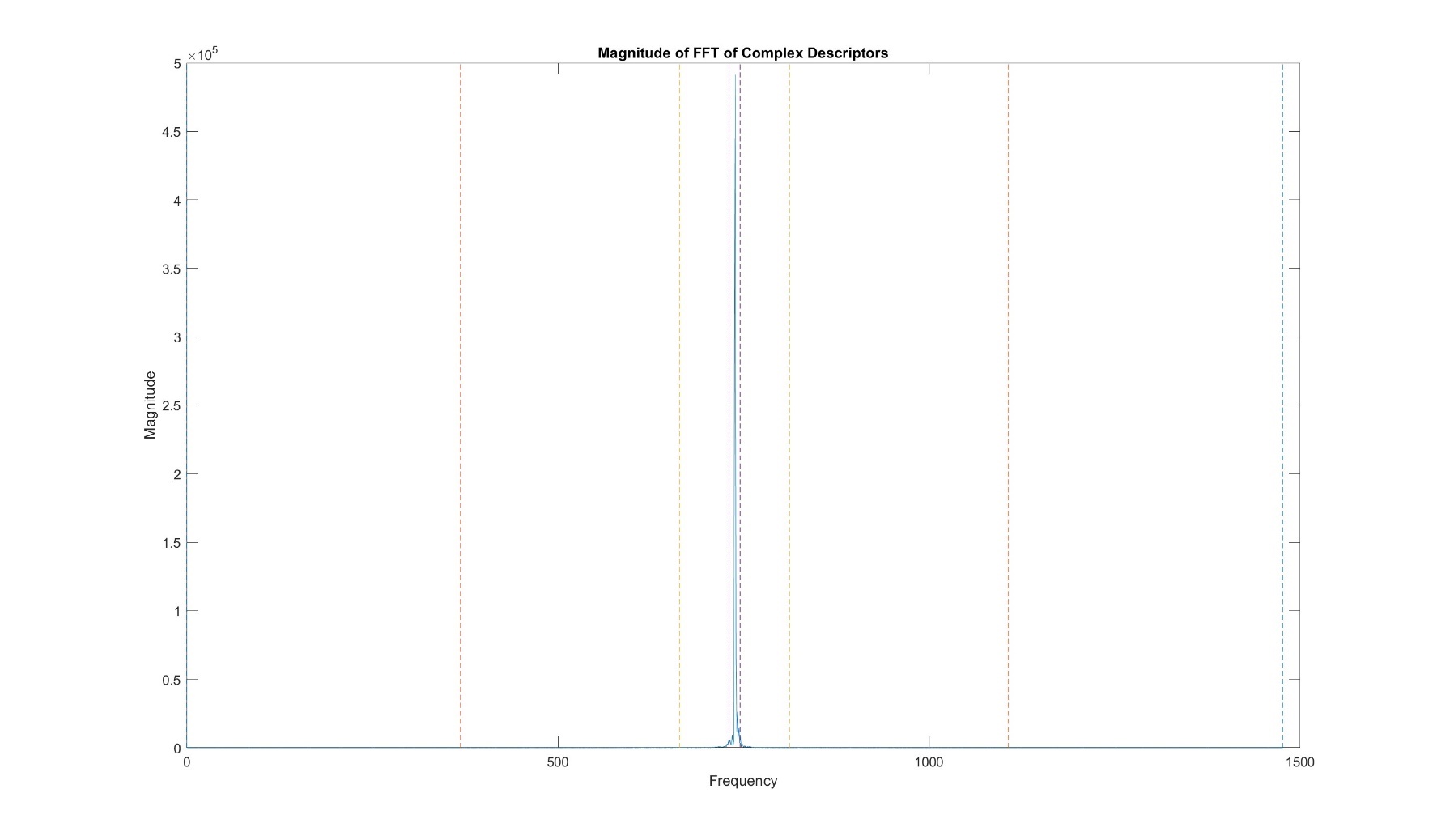
Για διαφορετικά λοιπόν ποσοστά σημαντικών συντελεστών που θα κρατήσουμε, υπολογίζουμε το πλήθος αυτών των συντελεστών, βάση του πλήθους των descriptors.

x = percentages(i) / 100;

num\_coeffs = round(x \* length(complex\_descriptors));

Για την ανακατασκευή, επιλέγουμε από τις χαμηλές συχνότητες οι μισές να είναι αριστερά του κέντρου και οι άλλες μισές να είναι δεξιά του.

reconstructed\_descriptors = complex\_descriptors(center\_index - num\_coeffs/2 + 1 : center\_index + num\_coeffs/2);



Στην συνέχεια επιστρέφουμε το σήμα στο πεδίο του χώρου:

reconstructed\_contour = ifft(ifftshift(reconstructed\_descriptors));

Οι ανακατασκευές που προκύπτουν:

|  |  |
| --- | --- |
|  |  |
|  |  |

A graph of a graph

Description automatically generated

Παρατηρούμε ότι για 1% των συντελεστών το αποτέλεσμα της ανακατασκευής δεν είναι καλό καθώς το κατώφλι που θέσαμε για τις συχνότητες κόβει συχνότητες που είναι σημαντικές.

Για το 10% των συντελεστών η ανακατασκευή κρατάει το βασικό σχήμα του φύλλου αλλά χάνει την λεπτομέρεια των φυλλωμάτων.

Για το 50% και το 100% η ανακατασκευή του περιγράμματος είναι πολύ κοντά στην αρχική.

A green leaf with red border

Description automatically generatedΒήμα 5

Με αντίστοιχο τρόπο με προηγουμένως χρησιμοποιώντας την συνάρτηση της Matlab:

[B = bwtraceboundary(BW,P,fstep)](https://www.mathworks.com/help/images/ref/bwtraceboundary.html#d126e53398).

|  |  |
| --- | --- |
|  |  |
| A green leaf with red border  Description automatically generated |  |

**Ερώτηση 1.2**

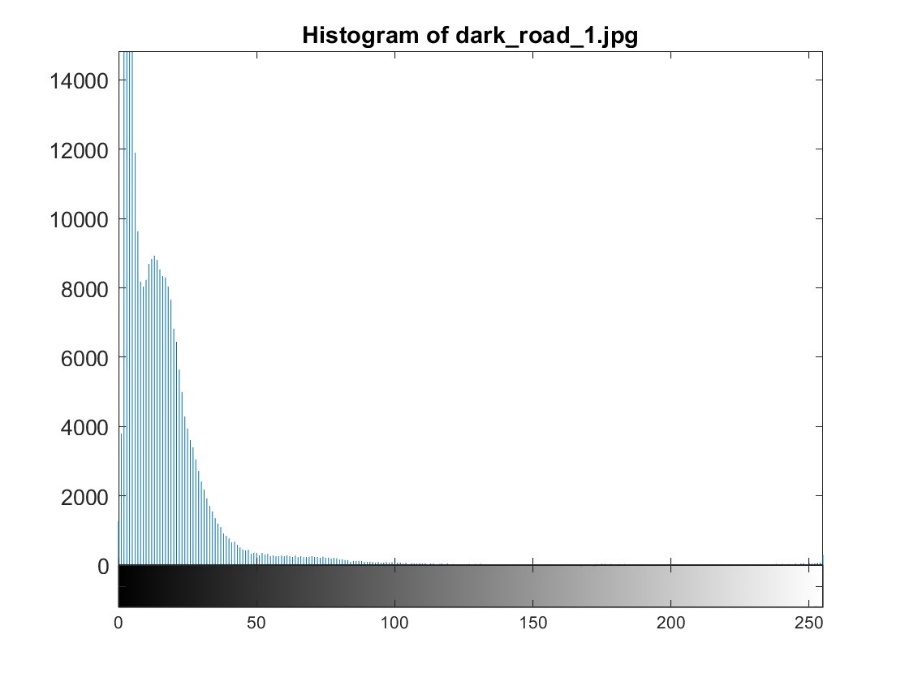
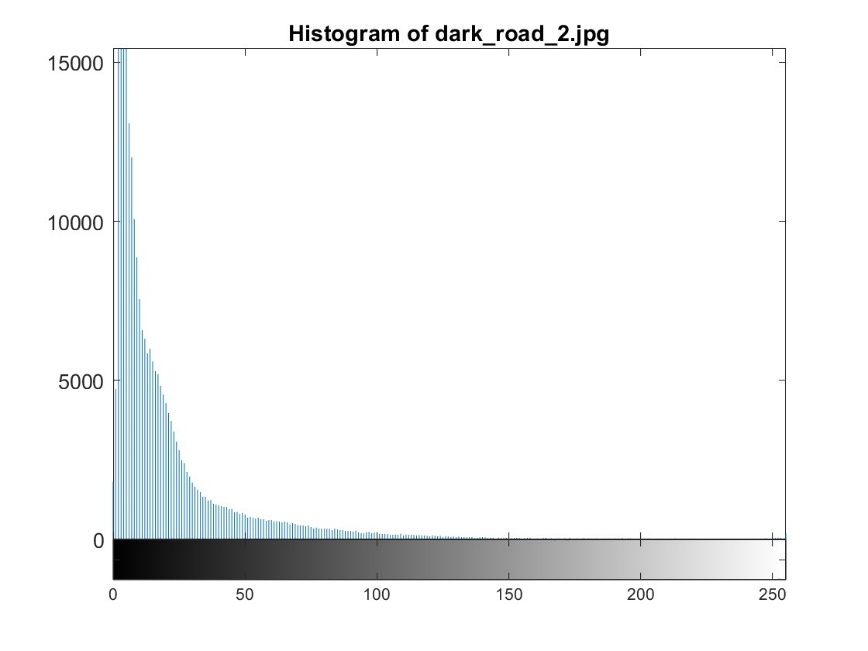
Συμπίεση Εικόνας

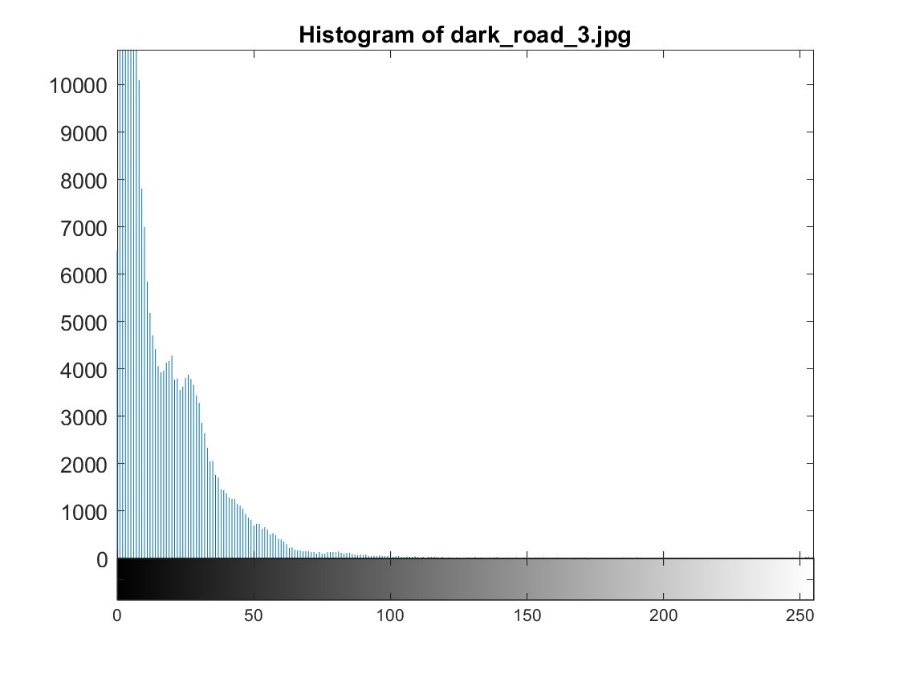
**Ερώτηση 1.3**

Φιλτράρισμα θορύβου

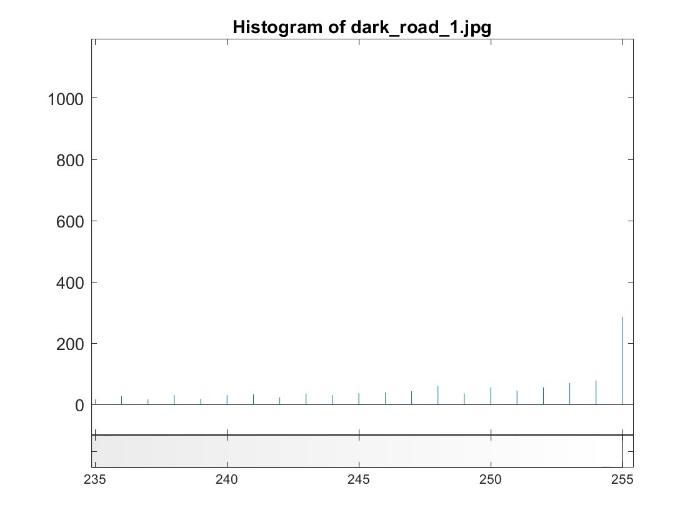
**Ερώτηση 1.4**

Βελτίωση Εικόνας – Εξίσωση Ιστογράμματος

Βήμα 1

Παρατηρούμε ότι και στα τρία ιστογράμματα οι τιμές κατανέμονται σε χαμηλές εντάσεις όπως αναμένουμε αφού η εικόνα είναι σκοτεινή. 

Εάν κάνουμε ζουμ στην εικόνα θα παρατηρήσουμε ότι το πλάτος των υψηλών εντάσεων είναι πολύ μικρό.

Επομένως η εικόνα είναι πιο σκοτεινή από

Βήμα 2

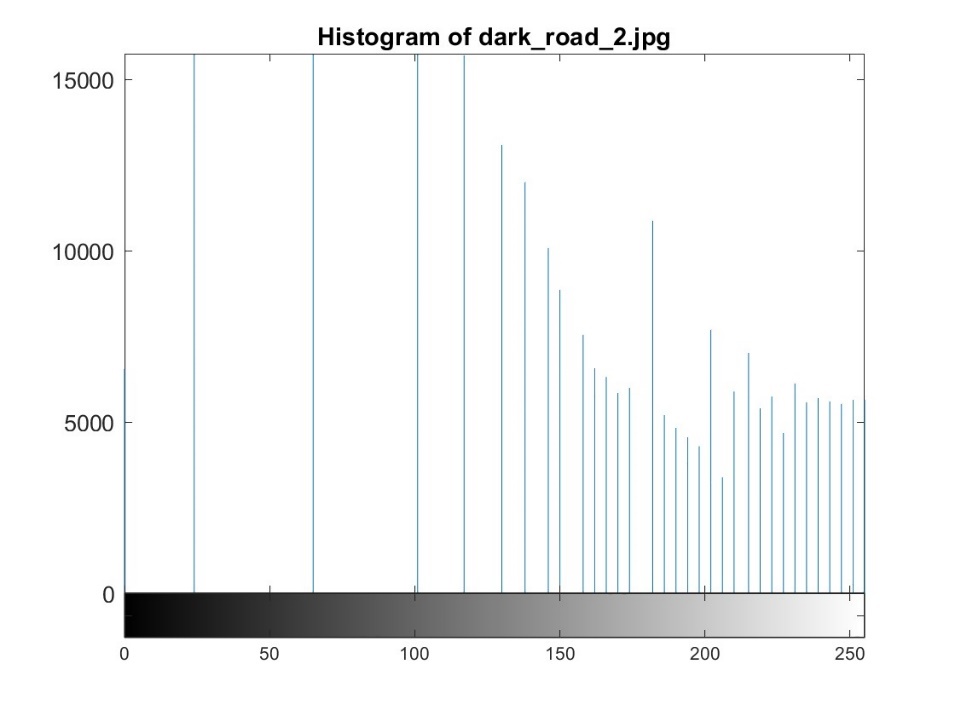
Εφαρμόζουμε ολική́ εξίσωση ιστογράμματος στις αρχικές εικόνες με την χρήση της συνάρτησης της MATLAB:

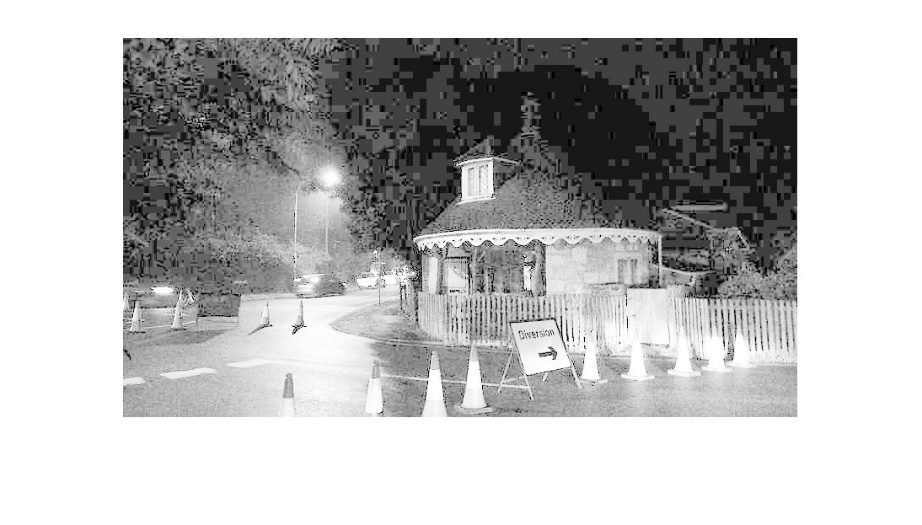
[J = histeq(Image)](https://www.mathworks.com/help/images/ref/histeq.html#d126e131711)

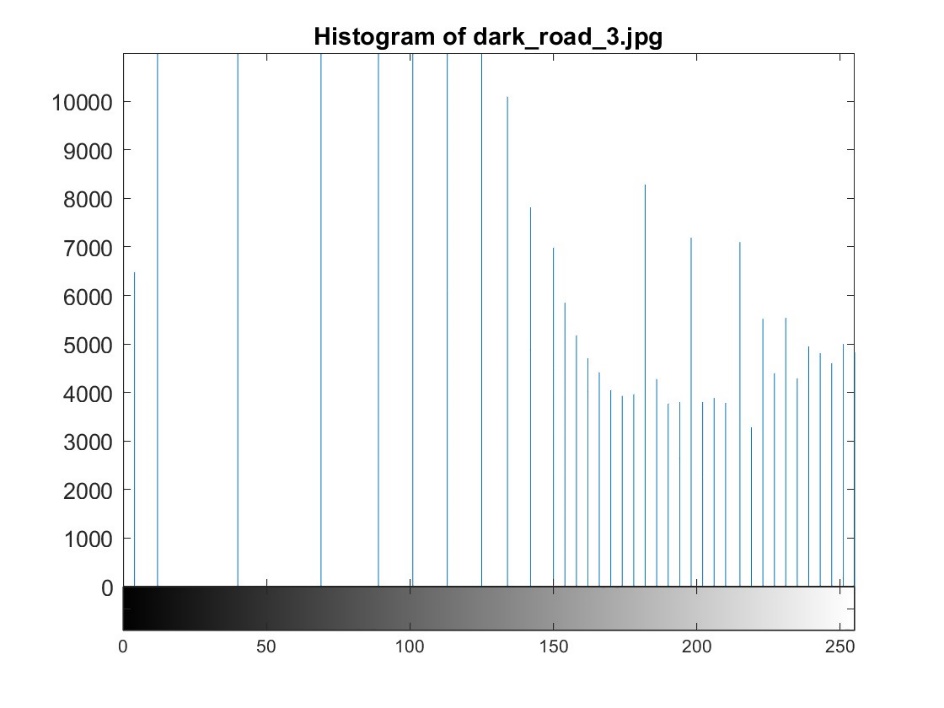
A graph of a graph showing dark road

Description automatically generated with medium confidence









Επειδή η τεχνική αυτή στοχεύει στην κατανομή των τιμών έντασης σε όλο το εύρος, μπορεί μερικές φορές να οδηγήσει σε ορισμένες περιοχές της εικόνας να γίνουν πολύ φωτεινές και επομένως σε απώλεια λεπτομερειών ή ακόμη και σε αποκοπή σε ακραίες περιπτώσεις.

Παρατηρούμε και στις τρις εικόνες ότι η εικόνα έχει αλλοιωθεί δραματικά από αυτό το φαινόμενο.

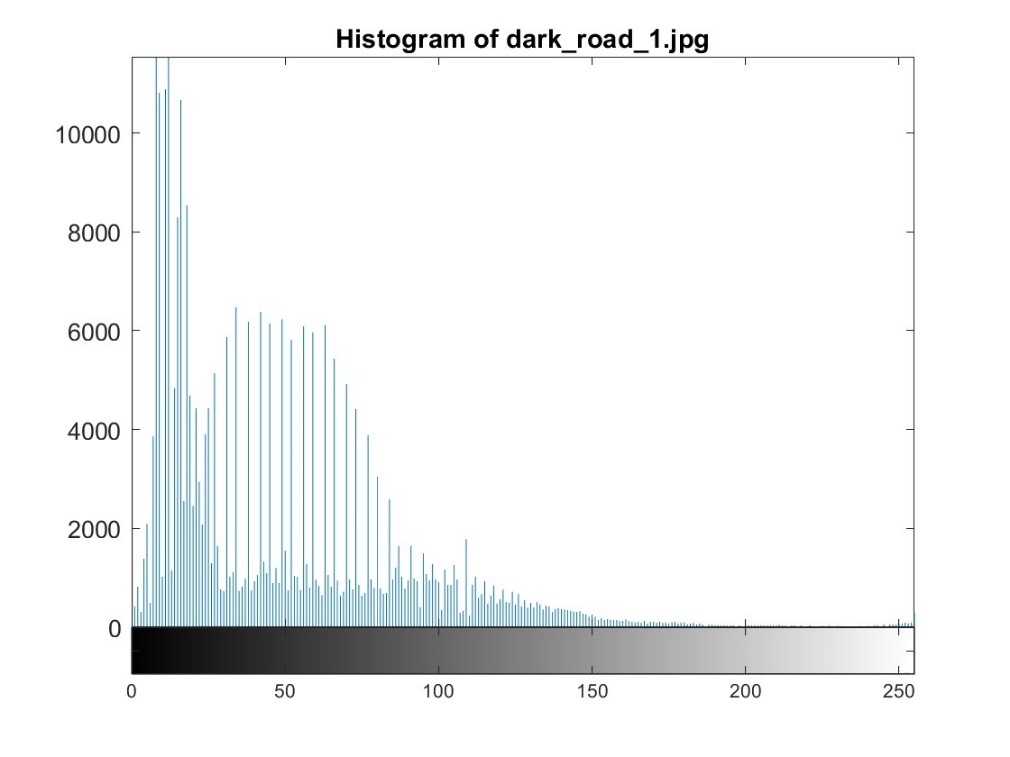
Με την ανακατανομή των τιμών έντασης, η ισοστάθμιση ιστογράμματος μπορεί να αναδείξει τις κρυμμένες λεπτομέρειες της εικόνας. Σε μια σκοτεινή εικόνα, μπορεί να υπάρχουν λεπτά χαρακτηριστικά ή υφές που είναι δύσκολο να διακριθούν λόγω της έλλειψης αντίθεσης. Η εξίσωση ιστογράμματος μπορεί να βοηθήσει στην αποκάλυψη αυτών των λεπτομερειών αυξάνοντας την αντίθεση μεταξύ γειτονικών εικονοστοιχείων. Εάν συγκρίνουμε τις εικόνες με τις αρχικές θα παρατηρήσουμε λεπτομέρειες στην εικόνα που δεν ήταν ορατές.

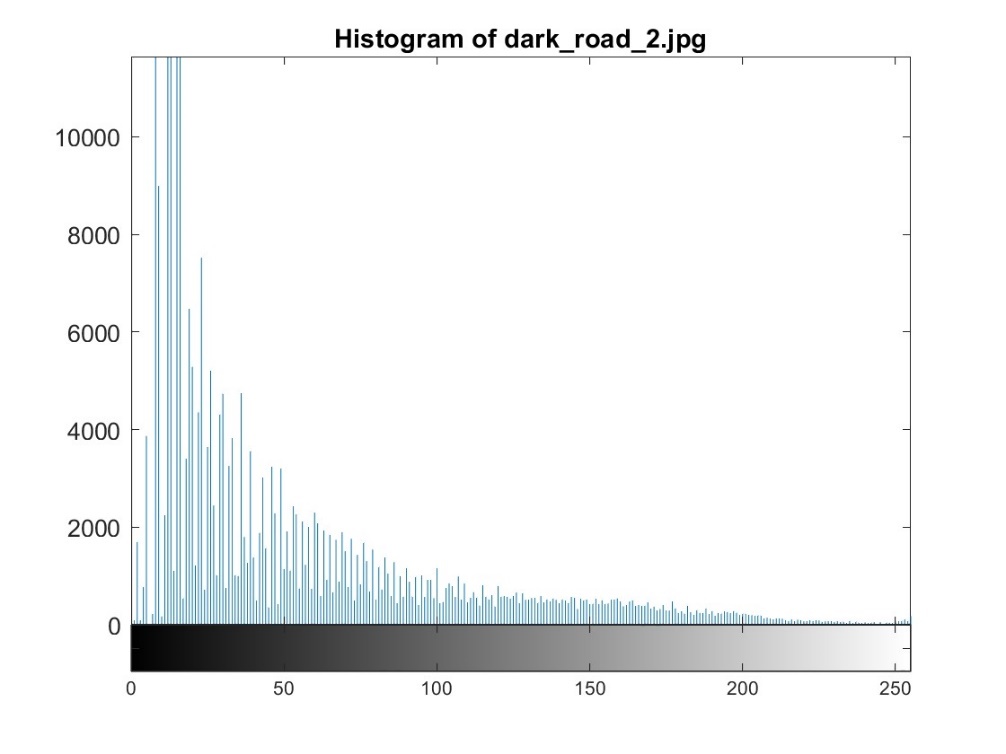
Βήμα 3

Εφαρμόζουμε ολική́ εξίσωση ιστογράμματος στις αρχικές εικόνες με την χρήση της συνάρτησης της MATLAB:

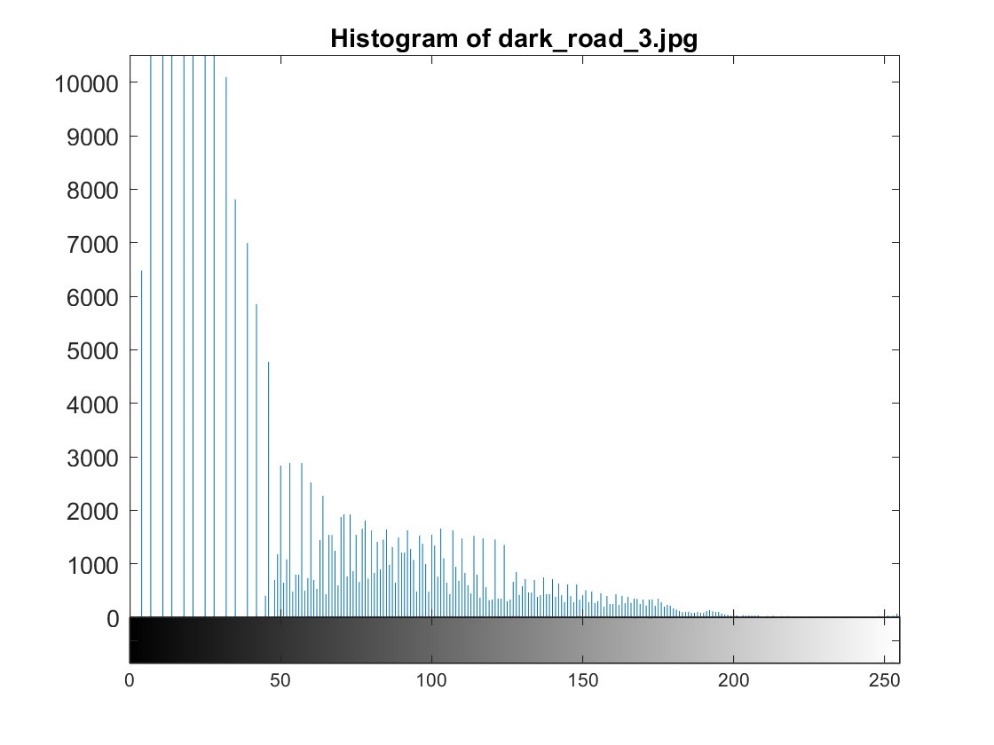
[J = adapthisteq(I,Name,Value)](https://www.mathworks.com/help/images/ref/adapthisteq.html#d126e9246)

Μετά από δοκιμή διάφορων τιμών για το παράθυρο, τελικά χρησιμοποιήθηκε παράθυρο [2 2], λόγο ικανοποιητικών αποτελεσμάτων χωρίς ιδιαίτερη προσθήκη θορύβου.









**Ερώτηση 1.5**

Αποκατάσταση Εικόνας (Wiener deconvolution)

**Ερώτηση 1.6**

Ανίχνευση Ακμών

Βήμα 1

Ορισμός μασκών Sobel:

sobel\_x = [-1 -2 -1; 0 0 0; 1 2 1];

sobel\_y = [-1 0 1; -2 0 2; -1 0 1];

Εφαρμογή των μασκών πάνω στην εικόν:

gradient\_x = conv2(double(image), sobel\_x, 'same');

gradient\_y = conv2(double(image), sobel\_y, 'same');

A dark hallway with columns

Description automatically generated

A dark hallway with many doors

Description automatically generated

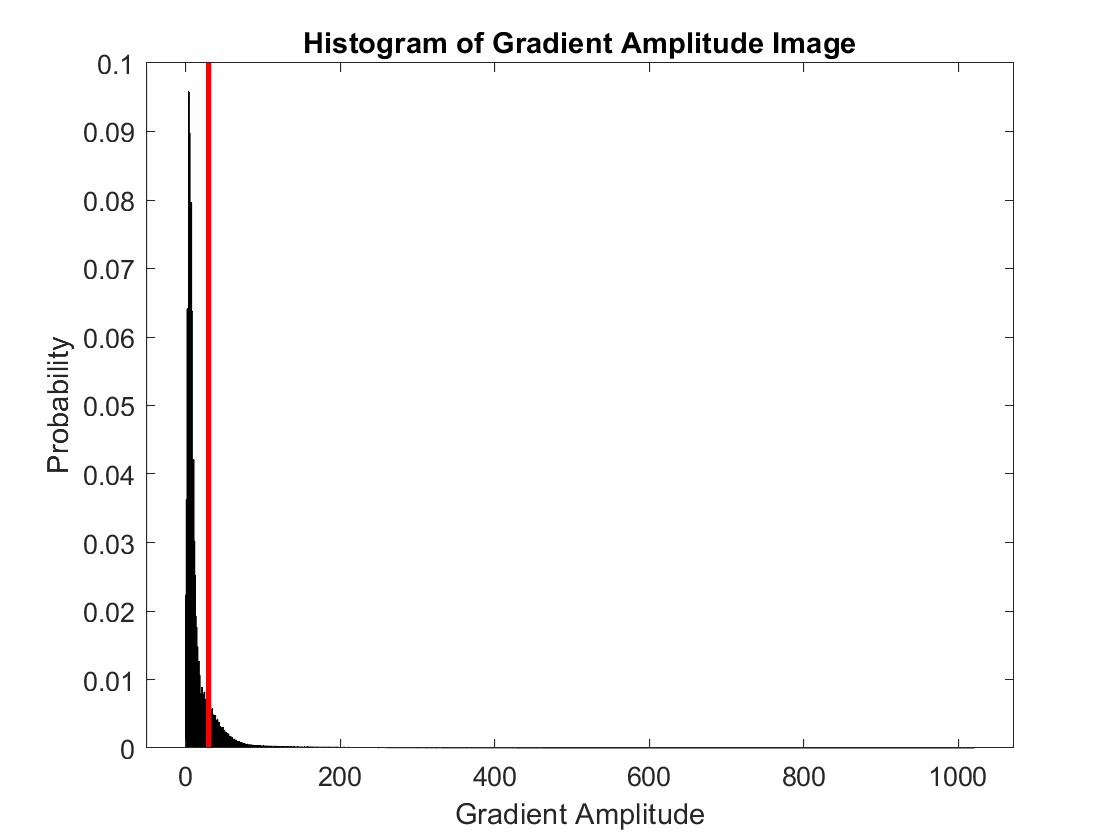
Βήμα 2

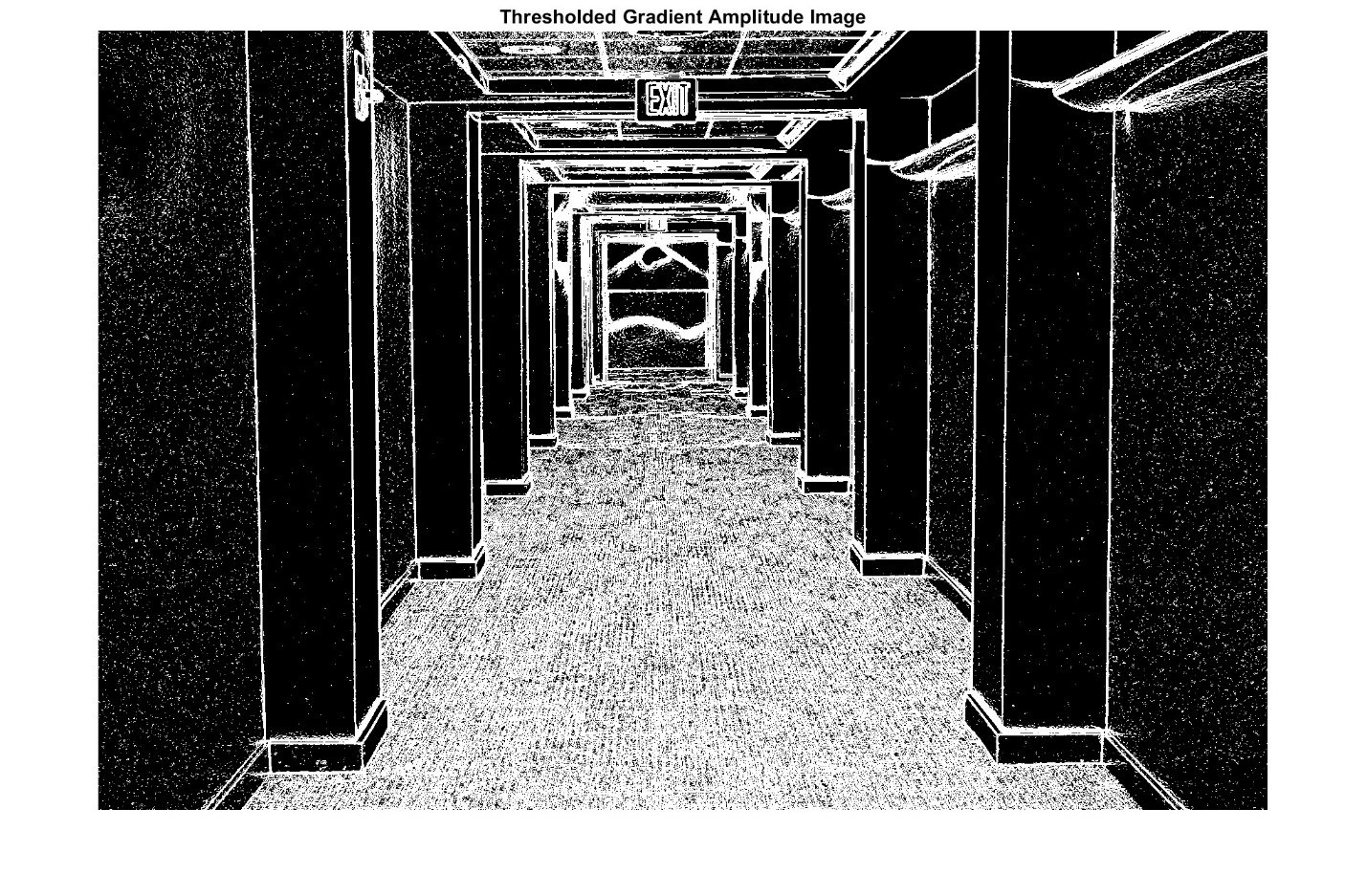
Επιλογή μιας τιμής κατωφλίου:

threshold = 30;

Κατωφλίωση gradient εικόνας:

thresholded\_image = gradient\_amplitude\_gamma > threshold;





Βήμα 3

**Ερώτηση 1.7**

Κατάτμηση Εικόνας

**Ερώτημα 1**

Βήμα a

Υπολογισμός ιστογράμματος:

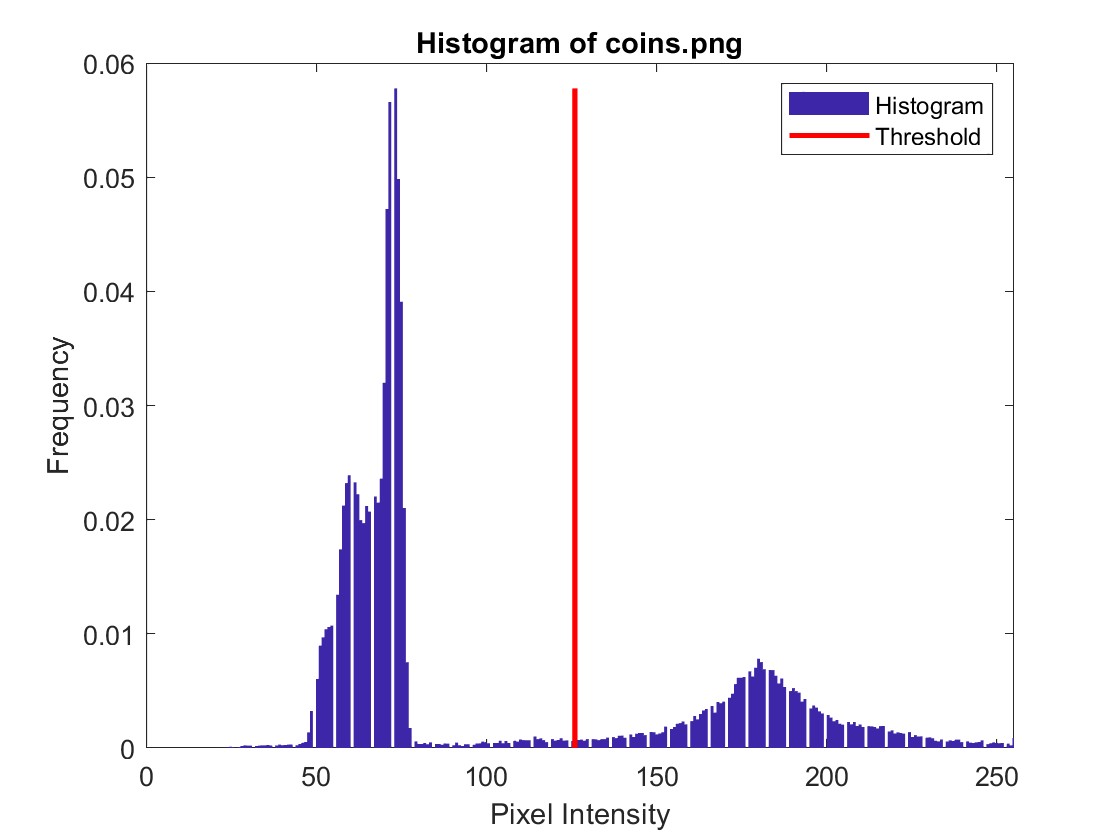
[counts, bins] = imhist(coins, 300);

Υπολογισμός βέλτιστου κατωφλίου με την μέθοδο Otsu χρησιμοποιώντας την συνάρτηση της MATLAB:

[T = graythresh(I)](https://www.mathworks.com/help/images/ref/graythresh.html#d126e131124)

counts = counts / sum(counts);

otsuThreshold = graythresh(coins);



A group of white circles

Description automatically generated

Βήμα b

A close-up of several coins

Description automatically generated

Βήμα c

Για την αποτύπωση των κερμάτων με διαφορετικό χρώμα σε λευκό φόντο πρώτα αποτυπώθηκαν τα κέρματα σε λευκό φόντο με την χρήση μάσκας.

% Create a white background

whiteBackground = ones(size(coins), 'uint8') \* 255;

% Invert binary thresholded image to get background mask

backgroundMask = ~binaryImage;

maskedWhiteBackground = bsxfun(@times, whiteBackground, uint8(backgroundMask)); % Apply inverted mask to white background



Εφαρμογή της μάσκας στην αρχική εικόνα.

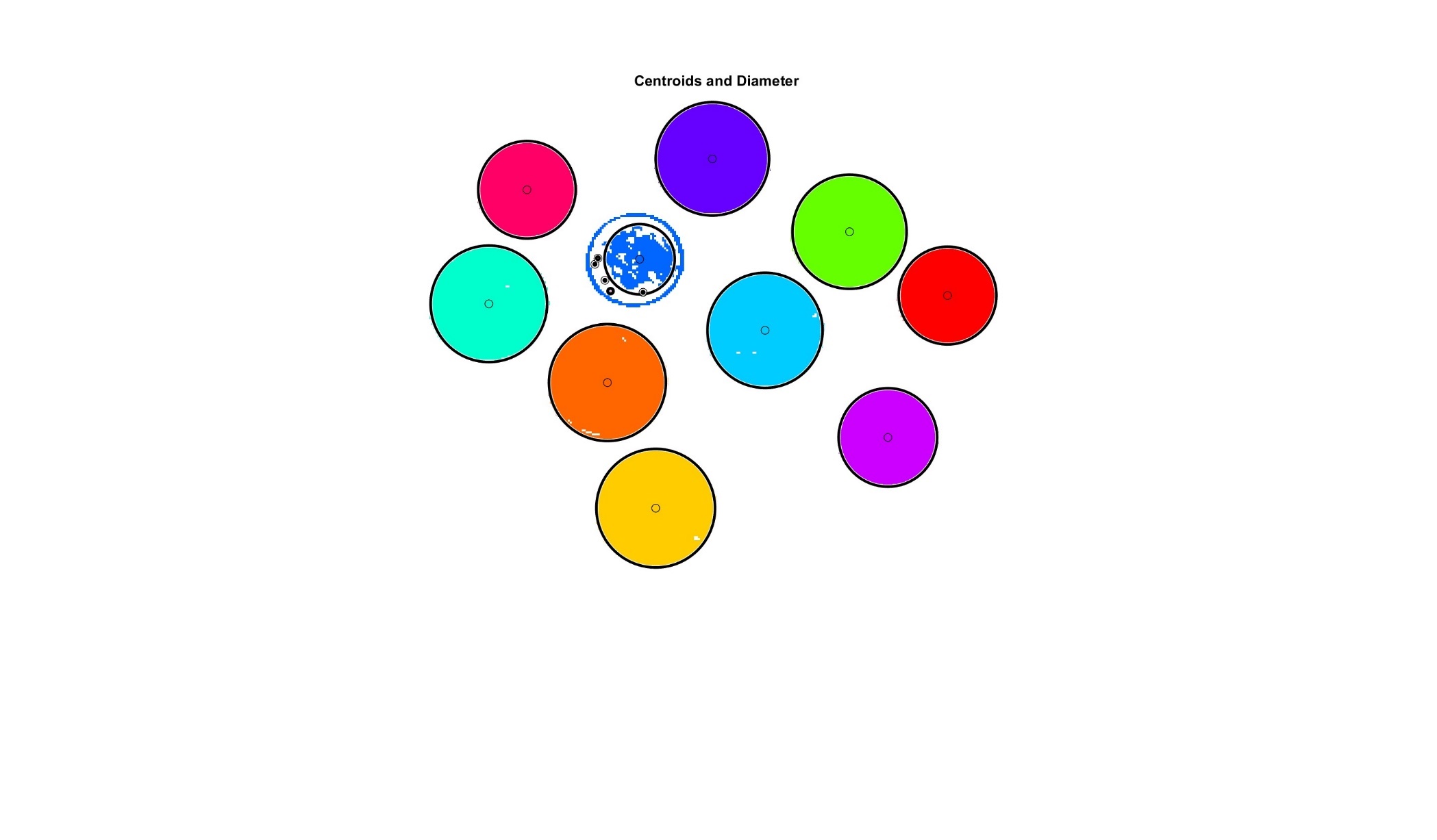
Στην συνέχεια χρησιμοποιείται η συνάρτηση bwlabel που επισημαίνει κάθε συνδεδεμένη συνιστώσα με έναν μοναδικό ακέραιο αριθμό.

[L = bwlabel(BW)](https://www.mathworks.com/help/images/ref/bwlabel.html#d126e46524)

Η επισημασμένη εικόνα μετατρέπεται σε εικόνα RGB όπου σε κάθε συνδεδεμένο στοιχείο (νόμισμα) ανατίθεται διαφορετικό χρώμα. Για το σκοπό αυτό χρησιμοποιείται η συνάρτηση label2rgb.

[RGB = label2rgb(L,cmap,zerocolor,order)](https://www.mathworks.com/help/images/ref/label2rgb.html#d126e223811)

Πάνω σε αυτή την εικόνα εφαρμόζεται η μάσκα που δημιουργήσαμε προηγουμένως.

Βήμα d

Χρησιμοποιώντας τη συνάρτηση regionprops, υπολογίζονται οι ιδιότητες των επισημασμένων περιοχών (νομίσματα). Συγκεκριμένα, υπολογίζονται το κεντροειδές και η ισοδύναμη διάμετρος κάθε νομίσματος.

Για την αναπαράσταση δημιουργείται μια άλλη εικόνα όπου το κεντροειδές κάθε νομίσματος επισημαίνεται με μια μαύρη κουκκίδα και σχεδιάζεται ένας κύκλος γύρω από κάθε νόμισμα για να αναπαραστήσει τη διάμετρό του.

Αυτό γίνεται με τις εντολές:

center = props(i).Centroid;

viscircles(center, diameter/2, 'EdgeColor', 'k');

**Ερώτημα 2**

Βήμα a

Μετασχηματισμός της αρχικής εικόνας Ι μεγέθους [𝑚, 𝑛, 3] σε ένα μητρώο 𝛸 διάστασης 𝑚𝑛 × 3 με την χρήση της reshape: [B = reshape(A,sz)](https://www.mathworks.com/help/matlab/ref/reshape.html#d126e1452532)

% Read the image

I = imread('football.jpg');

% Convert the image to double

I\_double = im2double(I);

% Reshape the image to [m\*n, 3]

[m, n, ~] = size(I\_double);

X = reshape(I\_double, m\*n, 3);

Βήμα b

Εφαρμογή του αλγορίθμου k-means στο μητρώο 𝛸 που υπολογίσαμε στο βήμα a, για διάφορες τιμές του κ με χρήση της συνάρτησης της MATLAB:

[idx = kmeans(X,k)](https://www.mathworks.com/help/stats/kmeans.html#d126e664699)

Επιστρέφει τα cluster indices IDX και τα κεντροειδή.

Βήμα c

Για κάθε cluster, δημιουργεί μια δυαδική μάσκα χρησιμοποιώντας τους δείκτες των cluster. Στη συνέχεια, εφαρμόζεται αυτή η μάσκα στην αρχική εικόνα και αποθηκεύεται η τμηματοποιημένη περιοχή στο segmented\_images.

% Construct images for each region

for j = 1:k

% Create a mask for pixels belonging to region j

mask = reshape(IDX == j, [m, n]);

% Apply the mask to the original image

segmented\_images(:, :, :, j) = repmat(mask, [1, 1, 3]) .\* I\_double;

end

A close-up of a football

Description automatically generated

