



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**ΑΝΑΠΤΥΞΗ ΕΞΕΛΙΓΜΕΝΩΝ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΤΗΝ ΨΗΦΙΑΚΗ
ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΑΝΑΛΥΣΗ ΚΥΤΤΑΡΟΠΑΘΟΛΟΓΙΚΩΝ ΕΙΚΟΝΩΝ
ΜΕ ΧΡΗΣΗ ΠΛΑΤΟΦΟΡΜΑΣ MATLAB**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Α. Νάννος

Επιβλέπων : Δημήτριος Κουτσούρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**ΑΝΑΠΤΥΞΗ ΕΞΕΛΙΓΜΕΝΩΝ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΤΗΝ ΨΗΦΙΑΚΗ
ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΑΝΑΛΥΣΗ ΚΥΤΤΑΡΟΠΑΘΟΛΟΓΙΚΩΝ ΕΙΚΟΝΩΝ
ΜΕ ΧΡΗΣΗ ΠΛΑΤΟΦΟΡΜΑΣ MATLAB**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικόλαος Α. Νάννος

Επιβλέπων : Δημήτριος Κουτσούρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25^η Οκτωβρίου 2011 .

.....
Δ. Κουτσούρης
Καθηγητής Ε.Μ.Π.

.....
Κ. Νικήτα
Καθηγήτρια Ε.Μ.Π.

.....
Π. Τσανάκας
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2011

.....
Νικόλαος Α. Νάννος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Νικόλαος Νάννος, 2011

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΛΗΨΗ

Στην εποχή μας η ψηφιακή απεικόνιση κυτταροπαθολογικών εικόνων και η ψηφιακή τους επεξεργασία και ανάλυση με τη βοήθεια ηλεκτρονικού υπολογιστή γνωρίζει ραγδαία ανάπτυξη. Αυτό συμβαίνει κυρίως επειδή ο ηλεκτρονικός υπολογιστής αυτοματοποιεί τη διαδικασία αναγνώρισης κακοήθειας και έτσι απαιτείται λιγότερος χρόνος διάγνωσης της. Επιπλέον, τα αποτελέσματα μπορούν να αποθηκευτούν με σκοπό την περαιτέρω μελέτη τους.

Στην παρούσα λοιπόν διπλωματική εργασία αναπτύχθηκαν διάφοροι εξελιγμένοι αλγόριθμοι για την κατάτμηση διαφόρων κυτταροπαθολογικών εικόνων. Οι αλγόριθμοι αυτοί υλοποιήθηκαν σε πλατφόρμα MATLAB, που είναι ένα ολοκληρωμένο περιβάλλον προγραμματισμού το οποίο χρησιμοποιεί πίνακες (MAtrixLABoratory).

Συγκεκριμένα αναπτύχθηκαν απλοί αλγόριθμοι φιλτραρίσματος-ομαλοποίησης της αρχικής εικόνας με χρήση μορφολογικών τελεστών και φίλτρου ενδιάμεσης τιμής, ο αλγόριθμος κατωφλίωσης Otsu, ο αλγόριθμος ταξινόμησης K-μέσων με βάση το χρώμα. Ιδιαίτερη έμφαση δόθηκε στη κατάτμηση με χρήση μορφολογικών τελεστών και συγκεκριμένα στο μετασχηματισμό watershed στις διάφορες εκδοχές του. Τέλικά αναπτύχθηκε αλγόριθμος βασισμένος στο μετασχηματισμό της απόστασης και έγινε μέτρηση δύο βασικών χαρακτηριστικών της κυτταροπαθολογικής εικόνας: του εμβαδού και της κυκλικότητας κάθε κυττάρου.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

κυτταροπαθολογική εικόνα, κατάτμηση, μορφολογικοί τελεστές, κατωφλίωση μετασχηματισμός watershed, μετασχηματισμός απόστασης

ABSTRACT

Nowadays the digital representation of cytopathology images and the computer aided digital processing and analysis is growing very fast. That is happening because the computer makes the procedure automateted and the time needed to make cancer diagnosis is less. Also the results can be stored for further study.

In this thesis advanced algorithms about medical image segmentation were developed. These algorithms were developed in the environment of MATLAB, which is a programming environment using matrices(MATrix LABoratory).

Specifically, the developed algorithms were: simple algorithms for filtering and smoothing the initial image using morphological operators and median filter, the Otsu's method for thresholding, the color-based k-means algorithm. It was given special attention to morphological segmentation and specifically the watershed transform in different versions. Finally an algorithm based on distance transform was developed and there was extraction of two basic cyto image characteristics, the area and the roundness of each cell.

KEYWORDS

cyto pathological image, segmentation, morphological operators, thresholding watershed transform, distance transform

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ.....	9
ΚΕΦΑΛΑΙΟ 1	
ΚΥΤΤΑΡΟΠΑΘΟΛΟΓΙΑ.....	11
1.1 Το υγιές κύτταρο.....	11
1.2 το καρκινικό κύτταρο	12
ΚΕΦΑΛΑΙΟ 2	
ΨΗΦΙΑΚΗ ΑΠΕΙΚΟΝΙΣΗ ΚΥΤΤΑΡΙΚΩΝ ΕΙΚΟΝΩΝ.....	14
2.1 Προέλευση κυτταρικών εικόνων.....	14
2.2 Ανάλυση κυτταρικών εικόνων.....	14
2.3 Λήψη και ψηφιοποίηση κυτταρικών εικόνων.....	15
2.3.1 Διαδικασία λήψης και ψηφιοποίησης κυτταρικής εικόνας.....	15
2.3.2 Μικροσκόπια.....	15
ΚΕΦΑΛΑΙΟ 3	
ΣΤΟΙΧΕΙΑ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΩΝ ΜΕ MATLAB.....	20
3.1 Η ψηφιακή εικόνα.....	20
3.2 Επεξεργασία ψηφιακής εικόνας.....	21
3.3 Οι τύποι των εικόνων και η δομή τους στο Matlab.....	22
3.3.1 Ενδεικτικές εικόνες(Indexed images).....	23
3.3.2 Ασπρόμαυρες εικόνες(Grayscale/intensity images).....	23
3.3.3 Έγχρωμες εικόνες RGB.....	24
3.4 Χρήσιμες συναρτήσεις ανάλυσης και επεξεργασίας εικόνας στο Matlab.....	25
3.5 Βελτιστοποίηση εικόνας.....	26
3.5.1 Τεχνικές τοπικής επεξεργασίας ή επεξεργασίας σημείου.....	26
3.5.1.1 Αντιστοίχιση φωτεινότητας.....	26
3.5.1.2 Εξομάλυνση (smoothing) της εικόνας.....	29
3.5.1.3 Όξυνση-τονισμός των ακμών της εικόνας (edge sharpening).....	30
3.5.2 Φιλτράρισμα εικόνας για αφαίρεση θορύβου.....	33
3.5.2.1 Γραμμικά φίλτρα (Linear filtering).....	33
3.5.2.2 Φίλτρα ενδιάμεσης τιμής (Median filtering).....	34
3.5.2.3 Προσαρμοζόμενα φίλτρα (Adaptive filtering).....	34
3.6 Μορφολογική επεξεργασία εικόνας	34
3.6.1 Erosion(Διάβρωση) και Dilation(Διαστολή).....	35
3.6.2 Άνοιγμα(opening) και κλείσιμο(closing).....	37
3.6.3 Βασικές μορφολογικές συναρτήσεις στο Matlab.....	37
3.6.3.1 Ο μετασχηματισμός Hit-or-Miss.....	37
3.6.3.2 Η συνάρτηση δυαδικών εικόνων bwmorph.....	38
3.6.3.3 Ετικέτες σε συνδεδεμένα αντικείμενα δυαδικών εικόνων...	39
3.6.3.4 Οι μετασχηματισμοί top-hat και bot-hat.....	39
3.6.4 Μορφολογική ανακατασκευή εικόνας	40
3.6.4.1 Άνοιγμα με ανακατασκευή(opening by reconstruction).....	40
3.6.4.2 Γέμισμα οπών.....	41
3.6.4.3 Αφαίρεση αντικειμένων των συνόρων(border objects)....	41

ΚΕΦΑΛΑΙΟ 4	
ΚΑΤΑΤΜΗΣΗ ΕΙΚΟΝΑΣ.....	43
4.1 Κατωφλίωση.....	43
4.1.1 Απλή γενική κατωφλίωση(Basic Global Thresholding).....	44
4.1.1.1 Αυτόματη επιλογή κατωφλίου των Gonzales και Woods.....	44
4.1.1.2 Αυτόματη επιλογή κατωφλίου - Μη παραμετρική μέθοδος κατωφλίωσης OTSU.....	45
4.1.2 Προσαρμοστική ή Τοπική κατωφλίωση(Adaptive-Local Thresholding).....	48
4.2 Κατάτμηση με χρήση τεχνικών ταξινόμησης.....	49
4.2.1 Ο αλγόριθμος των K-μέσων(K-means).....	50
4.3 Κατάτμηση με χρήση μετασχηματισμού Watershed.....	54
4.3.1 Κατάτμηση με χρήση μετασχηματισμού Watershed χρησιμοποιώντας το μετασχηματισμό απόστασης.....	55
4.3.2 Κατάτμηση με χρήση μετασχηματισμού Watershed χρησιμοποιώντας Gradients.....	57
4.3.3 Κατάτμηση με χρήση Marker-controlled Watershed.....	58
ΚΕΦΑΛΑΙΟ 5	
ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕ ΧΡΗΣΗ MATLAB.....	61
5.1 Προεπεξεργασία εικόνας.....	61
5.2 Αλγόριθμος Κατωφλίωσης.....	65
5.3 Κατάτμηση βασισμένη στο χρώμα με χρήση ταξινομητή K-μέσων με εφαρμογή σε εικόνα καρκινικών κυττάρων στήθους.....	67
5.4 Αλγόριθμοι κατάτμησης με μορφολογικούς τελεστές.....	72
5.4.1 Αλγόριθμος κατάτμησης με χρήση φίλτρου ενδιάμεσης τιμής και μορφολογικούς τελεστές με εφαρμογή σε εικόνα λεμφοκυττάρων.....	72
5.4.2 Αλγόριθμος κατάτμησης με μορφολογικούς τελεστές, μέτρηση και ποσοτικοποίηση κυττάρων με εφαρμογή σε εικόνα εγκεφαλικών κυττάρων.....	75
5.5 Κατάτμηση με μετασχηματισμό Watershed.....	79
5.5.1 Κατάτμηση με watershed μετασχηματισμού απόστασης με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος.....	79
5.5.2 Κατάτμηση με watershed με χρήση Gradients, με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος.....	81
5.5.3 Κατάτμηση με marker-controlled watershed με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος.....	83
5.6 Κατάτμηση μόνο με μετασχηματισμό απόστασης.....	86
5.7 Μέτρηση χαρακτηριστικών των κυττάρων που ανιχνεύτηκαν.....	89
5.8 Συμπεράσματα και μελλοντική εργασία.....	94
ΠΑΡΑΡΤΗΜΑ.....	95
Α. Κώδικες Αλγορίθμων Κεφαλαίου 5 σε Matlab.....	95
Β. Ανάπτυξη Γραφικής Διεπαφάνειας Χρήστη (GUI).....	112
Γ. Κλινική εφαρμογή σε εικόνες που ελήφθησαν από το κυτταρολογικό εργαστήριο του Πανεπιστημιακού Γενικού Νοσοκομείου “ΑΤΤΙΚΟΝ”.....	124
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	129

ΕΙΣΑΓΩΓΗ

Σκοπός της εργασίας αυτής είναι η ανάπτυξη διαφόρων εξελιγμένων αλγορίθμων για την κατάτμηση κυτταροπαθολογικών εικόνων και η εξαγωγή διαφόρων χαρακτηριστικών των κυττάρων από αυτές για την αυτόματη διάγνωση κακοήθειας, με τη βοήθεια πλατφόρμας Matlab με το εργαλείο επεξεργασίας εικόνας (Image Processing Toolbox). Η αυτοματοποίηση της διαδικασίας έχει πολλά πλεονεκτήματα: ο υπολογιστής δίνει αντικειμενικά αποτελέσματα με αποτέλεσμα η κακοήθεια να ανιχνεύεται ευκολότερα, επίσης τα αποτελέσματα μπορούν εύκολα να αποθηκευτούν και να μελετηθούν μελλοντικά.

Αρχικά υλοποιήθηκαν αλγόριθμοι φιλτραρίσματος-ομαλοποίησης της εικόνας (προεπεξεργασία) με χρήση μορφολογικών τελεστών και φίλτρου ενδιάμεσης τιμής. Στη συνέχεια για την κατάτμηση υλοποιήθηκαν διάφοροι αλγόριθμοι: ο αλγόριθμος κατωφλίωσης Otsu, ο αλγόριθμος ταξινόμησης k-μέσων με βάση το χρώμα, αλγόριθμοι με χρήση μορφολογικών τελεστών, αλγόριθμοι με χρήση μετασχηματισμού watershed (μετασχηματισμού απόστασης, gradients και marker-controlled) καθώς και κατάτμηση μόνο με μετασχηματισμό απόστασης. Από τους αλγορίθμους αυτούς τελικά επιλέχθηκε εκείνος ο οποίος λειτούργησε καλύτερα στην κατάτμηση των εικόνων και ήταν πλήρως αυτοματοποιημένος (δηλαδή δεν χρειαζόταν καθόλου η επιλογή παραμέτρου από το χρήστη), ο αλγόριθμος αυτός ήταν του μετασχηματισμού της απόστασης. Τέλος, μετά από την ανάλυση της τμηματοποιημένης εικόνας μετρήθηκαν δύο βασικά χαρακτηριστικά των κυττάρων που ανιχνεύτηκαν, το μέγεθος (εμβαδόν σε pixel) και η κυκλικότητα.

Παρακάτω παρατίθεται μια σύντομη αναφορά στο περιεχόμενο των κεφαλαίων της εργασίας αυτής:

Στο **κεφάλαιο 1** παρουσιάζεται η δομή ενός υγιούς κυττάρου και ενός καρκινικού κυττάρου καθώς και οι διαφορές μεταξύ τους.

Στο **κεφάλαιο 2** γίνεται μια σύντομη περιγραφή για τον τρόπο λήψης και ψηφιοποίησης των κυτταρικών εικόνων, καθώς επίσης και μια σύντομη αναφορά για τον τρόπο λειτουργίας του μικροσκοπίου.

Στο **κεφάλαιο 3** παρουσιάζονται τα βασικότερα στοιχεία επεξεργασίας εικόνας με χρήση του Matlab. Γίνεται αναφορά στην ψηφιακή εικόνα και τον τρόπο επεξεργασίας της, στα είδη των εικόνων και τη δομή τους στο Matlab. Επίσης αναφέρονται οι βασικότεροι τρόποι βελτιστοποίησης εικόνας. Τέλος παρουσιάζεται η μορφολογική επεξεργασία εικόνας με τη βοήθεια του Matlab.

Στο **κεφάλαιο 4** γίνεται μια εκτενής αναφορά στις βασικότερες μεθόδους κατάτμησης εικόνων με αρκετά παραδείγματα αλγορίθμων σε Matlab.

Στο **κεφάλαιο 5** αναπτύσσονται και αναλύονται διάφοροι αλγόριθμοι κατάτμησης με εφαρμογή σε κυτταροπαθολογικές εικόνες, και υλοποιείται αλγόριθμος μέτρησης μεγέθους και κυκλικότητας των κυττάρων που ανιχνεύτηκαν.

Τέλος για την ευκολότερη χρήση των ανωτέρω αλγορίθμων του κεφαλαίου 5 δημιουργήθηκε μια Γραφική Διεπαφή Χρήστη(Graphics User Interface-GUI) της οποίας ο κώδικας παρατίθεται στο παράρτημα.

Στο παράρτημα επίσης παρατίθεται το αποτέλεσμα της επεξεργασίας και ανάλυσης εικόνων του εργαστηρίου Διαγνωστικής κυτταρολογίας του Πανεπιστημιακού Γενικού Νοσοκομείου “ΑΤΤΙΚΟΝ” , ως κλινική εφαρμογή της εργασίας αυτής.

ΚΕΦΑΛΑΙΟ 1

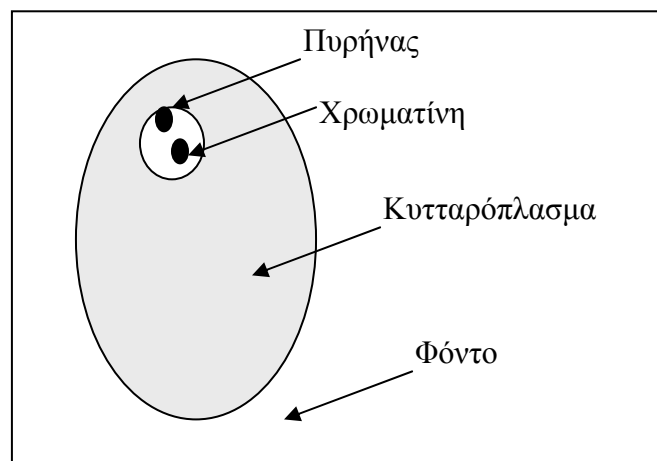
ΚΥΤΤΑΡΟΠΑΘΟΛΟΓΙΑ

1.1 Το υγιές κύτταρο

Η βασική μονάδα διαβίωσης του σώματος είναι το κύτταρο. Όλες οι δομές στο σώμα αποτελούνται από μια συλλογή από πολλά διαφορετικά κύτταρα, το καθένα εκτελεί σωστά την αποστολή του για τη στήριξη της συνολικής επιτυχίας ή της αποτυχίας του ανθρώπινου σώματος. Συνολικά, το σώμα μας περιέχει περίπου 75 τρισεκατομμύρια κύτταρα, το καθένα εργάζεται ανεξάρτητα, και αν αυτό γίνεται σωστά το συνολικό αποτέλεσμα είναι να έχουμε έναν υγιές οργανισμό.

Η ανθρώπινη ζωή ξεκινά με ένα μόνο κύτταρο. Τα υγιή κύτταρα έχουν τη δυνατότητα να διπλασιάζονται (κυτταρική διαίρεση) καθώς και την ικανότητα να εξειδικεύονται για να μπορέσουν να επιτελέσουν διαφορετικές λειτουργίες (διαφοροποίηση). Η κυτταρική διαίρεση εξασφαλίζει τη διατήρηση του οργανισμού όταν υπάρχει κυτταρική απώλεια. Από παρατήρηση γνωρίζουμε ότι όσο πιο διαφοροποιημένα είναι τα κύτταρα που αποτελούν έναν ιστό, τόσο σπανιότερα πολλαπλασιάζονται. Αυτό σημαίνει ότι ένας ιστός που αποτελείται από πολύ διαφοροποιημένα κύτταρα 'επιδιορθώνεται' δυσκολότερα μετά από τραυματισμό από ότι ένας ιστός του οποίου τα κύτταρα παρουσιάζουν μικρότερη διαφοροποίηση.

Ένα κύτταρο αποτελείται από ένα πυρήνα μέσα σε ένα κυτταρόπλασμα (Εικόνα 1.1).



Εικόνα 1.1 Υγιές Κύτταρο

Παρακάτω δίνονται ορισμένες πληροφορίες για τα δομικά χαρακτηριστικά ενός υγιούς κυττάρου:

- Τις περισσότερες φορές για τα δομικά στοιχεία του κυττάρου ισχύει η εξής σειρά:

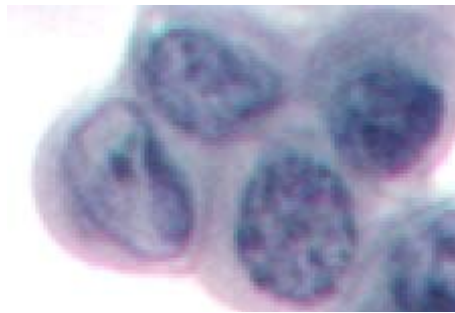
Συγκέντρωση χρωματίνης \subset Πυρήνας \subset Κυτταρόπλασμα \subset Φόντο

Δυστυχώς, πολλές διαφοροποιήσεις μπορούν να υπάρξουν όπως το φόντο μέσα στο κυτταρόπλασμα, ο πυρήνας εκτός του κυτταροπλάσματος ή συγκέντρωση χρωματίνης εκτός του πυρήνα.

- Ο πυρήνας είναι πιο σκούρος από το κυτταρόπλασμα· η αντίθεση μπορεί να είναι πολύ μικρή (ο πυρήνας και το κυτταρόπλασμα δεν είναι ομοιογενές).
- Εάν ένας πυρήνας είναι πολύ μικρός ($\approx 8\mu$), τότε δεν μπορεί να είναι καρκινικό το κύτταρο. Μπορεί να είναι ένα νεκρό κύτταρο ή συγκέντρωση χρωματίνης.

Τα περισσότερα υγιή κύτταρα είναι μικρά, και είναι σχεδόν κυκλικά και τα σύνορά τους είναι ομαλά. Άλλο ένα βασικό χαρακτηριστικό των υγιών κυττάρων είναι το ότι το κυτταρόπλασμα είναι πολύ μεγαλύτερο από τον πυρήνα. [15]

1.2 Το Καρκινικό Κύτταρο



Εικόνα 1.2 Καρκινικά Κύτταρα

Τα καρκινικά κύτταρα σε αντίθεση με τα υγιή δεν παρουσιάζουν διαφοροποίηση και έχουν ως βασική λειτουργία το πολλαπλασιασμό. Ενώ σε ένα φυσιολογικό ιστό ένας ακριβής ρυθμιστικός μηχανισμός καθορίζει τη συνεχή μετάβαση πολλαπλασιασμού και ανάπαυσης, ώστε να διατηρείται σταθερός ο αριθμός των κυττάρων, στο καρκινικό ιστό (όγκος) τα κύτταρα αναπαράγονται σχεδόν ανεξέλεγκτα, τα κύτταρα διαιρούνται πριν ολοκληρωθεί η διαίρεση των χρωμοσωμάτων ή κάποια χρωμοσώματα πολλαπλασιάζονται περισσότερο από μια φορά πριν από τη διαίρεση του κυττάρου.

Υπάρχει ένας μεγάλος αριθμός μικρών διαφορών μεταξύ ενός υγιούς κυττάρου και ενός καρκινικού κυττάρου. Τέσσερα όμως είναι τα βασικότερα [15] :

- Οι καρκινικοί πυρήνες έχουν τεράστια διάμετρο
- Ο λόγος πυρήνα/κυτταρόπλασμα είναι πολύ μεγαλύτερος από ένα υγιές κύτταρο(πλησιάζει τη μονάδα πολλές φορές)
- Το σχήμα του πυρήνα είναι ακανόνιστο, όπως φαίνεται στην εικόνα 1.2
- Τα σύνορα των καρκινικών πυρήνων είναι ανομοιογενή
- Τα καρκινικά κύτταρα είναι πιο σκούρα από τα υγιή.

ΚΕΦΑΛΑΙΟ 2

ΨΗΦΙΑΚΗ ΑΠΕΙΚΟΝΙΣΗ ΚΥΤΤΑΡΙΚΩΝ ΕΙΚΟΝΩΝ

2.1 Προέλευση κυτταρικών εικόνων

Οι κυτταρικές εικόνες που επεξεργαζόμαστε σε αυτή την εργασία προέρχονται από διάφορους ιστούς(στήθους, εγκέφαλου, πάγκρεας, τεστ παπ, θυρεοειδούς κλπ).

Η βασική διαδικασία λήψης εικόνας από κάποιο πιθανώς προσβεβλημένο από καρκίνο ιστό είναι η παρακάτω:

1. Αρχικά γίνεται απομάκρυνση ενός δείγματος κυττάρων από τον ιστό χρησιμοποιώντας βελόνα ή άλλη δειγματοληπτική συσκευή.
2. Στη συνέχεια τα κύτταρα μεταφέρονται σε γυάλινη επιφάνεια (slide) και στέλνονται στο κυτταρολογικό εργαστήριο, όπου προετοιμάζονται με ειδικές χρώσεις για την μικροσκοπική εξέταση.

2.2 Ανάλυση κυτταρικών εικόνων

Η αρχική μικροσκόπηση γίνεται από κυτταροτεχνολόγο (primary screener), ο οποίος έχει εκπαιδευτεί για να είναι ικανός να ανιχνεύει άτυπα κύτταρα ανάμεσα σε χιλιάδες φυσιολογικών κυττάρων στο επίχρισμα.

Όμως ο ανθρώπινος εγκέφαλος μπορεί να εξάγει συμπεράσματα σχετικά με τα ποιοτικά χαρακτηριστικά (πχ πολύ φωτεινό αντικείμενο) αλλά δε μπορεί να εξάγει ποσοτικές πληροφορίες(δε μπορεί να πει πόσο φωτεινό είναι το αντικείμενο). Αντιθέτως οι υπολογιστές μπορούν να δώσουν ποσοτικές πληροφορίες αλλά όχι ποιοτικές. Η απόλυτη φωτεινότητα (luminance) ενός αντικειμένου είναι ανεξάρτητη από τη φωτεινότητα των γύρω αντικειμένων. Το ανθρώπινο οπτικό σύστημα αντιλαμβάνεται τη σχετική φωτεινότητα (brightness) ενός αντικειμένου, δηλαδή τη φωτεινότητα του αντικειμένου σε σχέση με τη φωτεινότητα του φόντου του. Επιπλέον, η οπτική αντίληψη του ανθρώπινου ματιού είναι ευαίσθητη στις αντιθέσεις φωτεινότητας της εικόνας κι όχι στην καθ' αυτή φωτεινότητα. Το ανθρώπινο μάτι μπορεί να διακρίνει μόνο μερικές δεκάδες αποχρώσεις (επίπεδα) του γκριζου σε γκριζες εικόνες (gray-level images), ενώ είναι σε θέση να διακρίνει χιλιάδες χρώματα σε έγχρωμες εικόνες.

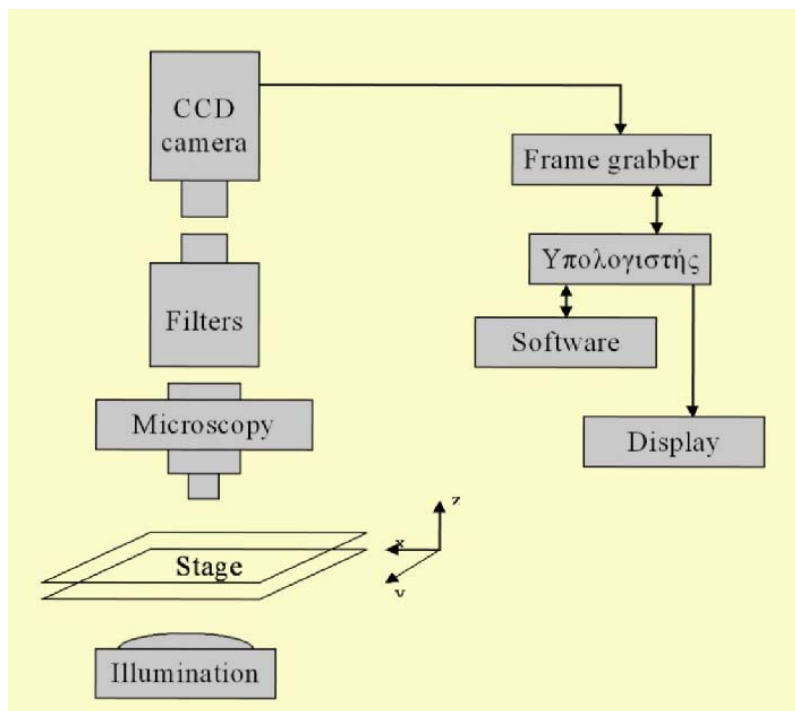
Στους υπολογιστές οι εικόνες απεικονίζονται σε ψηφιακή μορφή(ψηφιακές εικόνες). Οι ψηφιακές εικόνες προκύπτουν από κβαντισμό των εντάσεων των σημείων μιας εικόνας μετά από δειγματοληψία σε διακριτές τιμές. Μια εικόνα λοιπόν αποτελείται από διακριτά στοιχεία , τα pixels. Το χρώμα του κάθε pixel περιγράφεται από ένα διακριτό αριθμό. Στη γκριζα εικόνα η απόχρωση του γκρι για κάθε pixel δίνεται από έναν αριθμό από το 0 ως το 255 που αντιστοιχεί στο επίπεδο γκριζου του pixel. Στις έγχρωμες εικόνες , κάθε φασματικό κανάλι (συνήθως κοκκίνο,πράσινο, μπλε-RGB) παριστάνεται απο μια ξεχωριστή γκρι εικόνα.

2.3 Λήψη και ψηφιοποίηση κυτταρικών εικόνων

2.3.1 Διαδικασία λήψης και ψηφιοποίησης κυτταρικής εικόνας

Όπως αναφέρθηκε και προηγουμένως τα επιθηλιακά κύτταρα τοποθετούνται σε γυάλινη επιφάνεια (slide) και μετά από προετοιμασία με ειδικές χρώσεις εξετάζονται στο μικροσκόπιο και κατόπιν μέσω ψηφιακής φωτογραφικής μηχανής παίρνουμε φωτογραφία των υπο εξέταση κυττάρων.

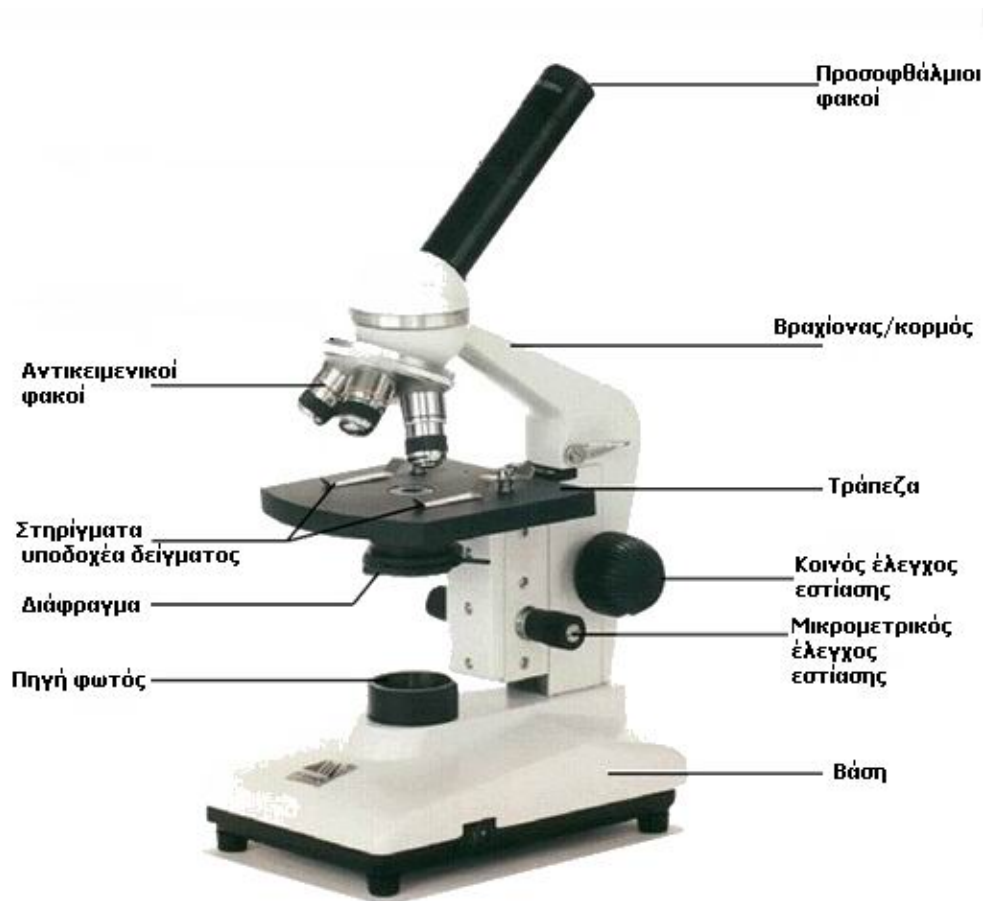
Η πειραματική διάταξη λήψης και ψηφιοποίησης των εικόνων φαίνεται στη παρακάτω εικόνα 2.1:



Εικόνα 2.1 Η πειραματική διάταξη που χρησιμοποιείται για τη λήψη και ψηφιοποίηση των εικόνων

2.3.2 Μικροσκόπια

Το πιο διαδεδομένο μικροσκόπιο σήμερα είναι το **σύνθετο μικροσκόπιο** το οποίο αποτελείται από ένα σταθερό τμήμα, τη βάση, πάνω στο οποίο είναι προσαρτημένες διατάξεις όπως το οπτικό τμήμα, η στήριξη των προς μελέτη παρασκευασμάτων (η αντικειμενοφόρος πλάκα), το σύνολο των μηχανισμών κίνησης για εστίαση και η συσκευή φωτισμού. Το φως διέρχεται από το σύστημα φωτισμού στο σύστημα φακών που επιτρέπει τον ομοιόμορφο φωτισμό του δείγματος. Η ένταση της δέσμης ρυθμίζεται κατάλληλα με τη χρήση διαφράγματος(Εικ. 2.2)



Εικόνα 2.2 Οπτικά και μηχανικά χαρακτηριστικά σύνθετου μικροσκοπίου

Τα βασικά χαρακτηριστικά ενός μικροσκοπίου είναι:

Τα βασικά χαρακτηριστικά ενός μικροσκοπίου είναι:

- Η διακριτική ικανότητα
- Το βάθος εστίασης
- Η διάμετρος πεδίου

και περιγράφονται παρακάτω.

Διακριτική ικανότητα

Ως όριο διακριτικής ικανότητας ορίζεται η ελάχιστη απόσταση μεταξύ δύο σημείων του παρατηρούμενου αντικειμένου που μπορούν να παρατηρηθούν ξεχωριστά.

Βάθος εστίασης

Ως βάθος εστίασης (Depth of Field) ορίζουμε το μικρό πάχος του παρασκευάσματος στο οποίο μπορούμε να εστιάσουμε με το μικροσκόπιο. Το βάθος εστίασης εξαρτάται από τον αντικειμενικό φακό που χρησιμοποιούμε και καθώς το N.A. αυξάνει, μειώνεται σημαντικά.

Διάμετρος πεδίου

Η διάμετρος του πεδίου είναι αντιστρόφως ανάλογη της ολικής μεγέθυνσης.

Στη κυτταρολογία χρησιμοποιούνται σήμερα κατά κύριο λόγο άλλα δύο είδη μικροσκοπίου, το **μικροσκόπιο αντίθετης φάσης** και το **μικροσκόπιο φθορισμού**.

Μικροσκόπιο αντίθετης φάσης (Εικ. 2.3(α))

Τα κύτταρα είναι διαφανή από τη φύση τους και, ως εκ τούτου, πρέπει να χρησιμοποιηθούν εξειδικευμένες τεχνικές για την παρατήρησή τους. Η αντίθεση φάσης είναι η λιγότερο τοξική μέθοδος η οποία δε δεν απαιτεί χρώση. Όταν το φως περνά μέσα από ένα διάφανο μέρος του δείγματος, αυτό μετατοπίζεται (διαθλώμενο (D) κύμα) σε σχέση με το ανεπηρέαστο φως (περιβάλλον (S) κύμα) και φτάνει περίπου το ένα τέταρτο του μήκους κύματος εκτός φάσης. Αυτή η διαφορά δεν είναι ανιχνεύσιμη από το ανθρώπινο μάτι. Ωστόσο, η αλλαγή στη φάση μπορεί να αυξηθεί στο $\frac{1}{2}$ του μήκους κύματος με ένα διαφανές πλακίδιο φάσης στο μικροσκόπιο προκαλώντας μια διαφορά στη φωτεινότητα (σωματιδιακό (P) κύμα). Λόγω της καταστρεπτικής παρέμβασης, το διαφανές αντικείμενο εμφανίζεται πιο σκούρο έναντι ενός πιο φωτεινού υπόβαθρου. Αυτό ονομάζεται σκοτεινή ή θετική αντίθεση φάσης (Εικ. 2.4(α)). Μια άλλη πιθανότητα είναι να επιβραδύνει το άμεσο φως έτσι ώστε το επηρεασμένο φως και το μη απορροφημένο να φτάσουν μαζί και να αναμειχθούν δημιουργικά. Αυτό έχει ως αποτέλεσμα το δείγμα να φαίνεται πιο φωτεινό από το φόντο. (Εικ. 2.4β)).

Η αυτοματοποιημένη ανάλυση των εικόνων με βιολογικές δομές (όπως είναι και οι κυτταρικές εικόνες) από το μικροσκόπιο αντίθεσης φάσης (Εικ. 2.3(α)) είναι αρκετά δύσκολη. Η σωστή ρύθμιση του μικροσκοπίου είναι απαραίτητη για να αξιοποιηθούν πλήρως οι δυνατότητές του και να εξασφαλιστεί η ομοιομορφία και φωτισμό χωρίς γυαλάδα (για παράδειγμα μέσω του φωτισμού kohler).

Ωστόσο το μικροσκόπιο αντίθεσης φάσης δημιουργεί φαινόμενα όπως σκίαση και φωτοστέφανο (Εικ. 2.5) που είναι αναπόφευκτα στα οπτικά συστήματα. Άλλο ένα φαινόμενο που παρουσιάζεται στο μικροσκόπιο αντίθεσης φάσης είναι ο ανομοιόμορφος φωτισμός. Εκδηλώνεται από τη κλίση στην ένταση φωτισμού που διέρχεται από τη μια πλευρά της εικόνας στην άλλη. Προκαλείται είτε από λάθος ρύθμιση του οπτικού συστήματος είτε από το μισοφέγγαρο του μέσου. [14], [22]

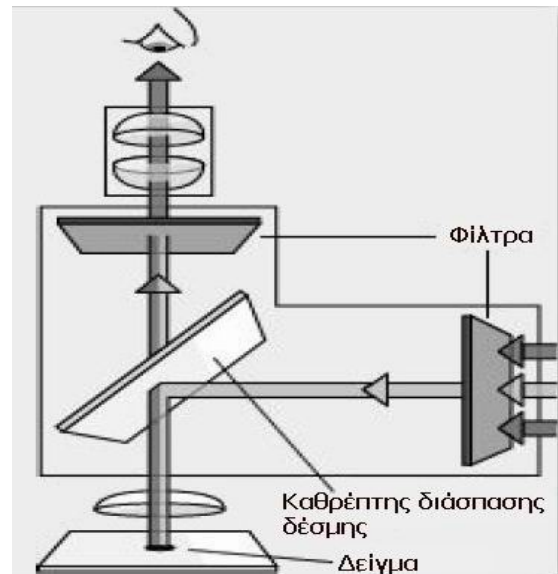
Μικροσκόπιο φθορισμού (Εικ. 2.3(β))

Όταν ένα συγκεκριμένο μέρος του κυτάρου πρέπει να παρατηρηθεί, μια διαφορετική τεχνική μικροσκοπίου πρέπει να χρησιμοποιηθεί. Το μικροσκόπιο φθορισμού χρησιμοποιείται για μελέτη δειγμάτων που μπορούν να υποχρεωθούν να φθορίσουν. Βασίζεται στο φαινόμενο κατά το οποίο ορισμένα υλικά εκπέμπουν ενέργεια ανιχνεύσιμη ως ορατό φως όταν ακτινοβοληθούν με φως συγκεκριμένου μήκους κύματος (Εικ. 2.3(β)). Πολλές διαφορετικές χρωστικές ουσίες μπορούν να χρησιμοποιηθούν για τη χρώση διαφορετικών δομών του κυτάρου. Από πλευράς κατασκευής το μικροσκόπιο φθορισμού είναι ένα κοινό μικροσκόπιο στο οποίο όμως το παρασκεύασμα μπορεί να φωτίζεται εκτός από το κλασικό τρόπο και με υπεριώδη ακτινοβολία. Οι φακοί των μικροσκοπίων αυτών είναι ειδικής κατασκευής, από γυαλί που δεν εμφανίζει "αυτοφθορισμό", και έχουν χαραγμένη πάνω τους τη λέξη Fluor ή Ultrafluor, μπορούν όμως να χρησιμοποιηθούν και για παρατήρηση σε διάταξη φωτεινού πεδίου.

Ανεπιθύμητα φαινόμενα στο μικροσκόπιο φθορισμού τα οποία συμβάλλουν στην τελική ποιότητα της εικόνας είναι ο ανομοιόμορφος φωτισμός και ο θόρυβος. Το πρώτο φαινόμενο σχηματίζεται από ένα μηνίσκο (μισοφέγγαρο), πχ κυρτότητα επιφάνειας μέσου. Το φως που διέρχεται από το μηνίσκο ανακλάται διαφορετικά με αποτέλεσμα η εικόνα να φωτίζεται με διαφορετικό τρόπο. Ο θόρυβος, γενικά τυχαία, μειώνει και αυξάνει τη φωτεινότητα της εικόνας σε κάθε pixel. Μπορεί να δημιουργείται λόγω της αστάθειας σε μια πηγή φωτός ή στον ίδιο τον ανιχνευτή εικόνας. [14], [22]

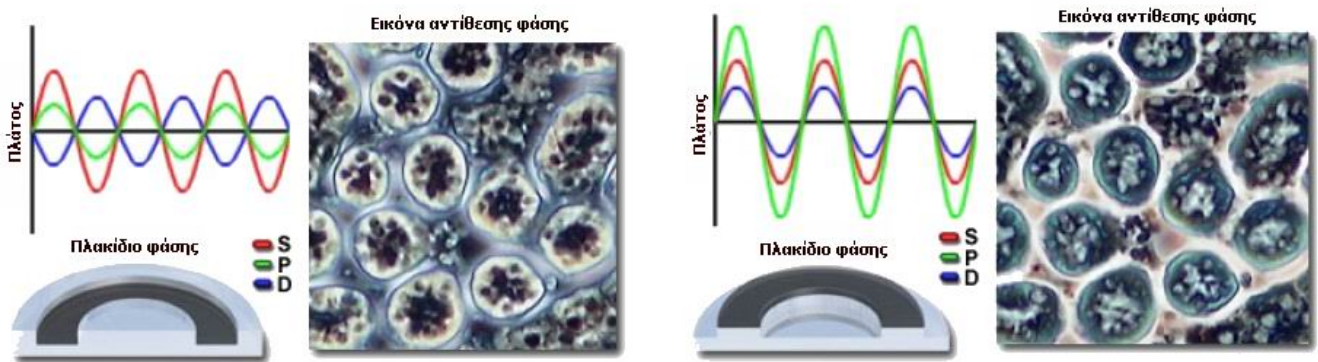


(α) Μικροσκόπιο αντίθεσης φάσης



(β) Μικροσκόπιο φθορισμού

Εικόνα 2.3 Βασικές αρχές λειτουργίας των μικροσκοπίων αντίθεσης φάσης και φθορισμού



(α) Θετική αντίθεση φάσης

(β) Αρνητική αντίθεση φάσης.

Εικόνα 2.4 Αρνητική και θετική αντίθεση φάσης ενός νεφρικού ιστού



Εικόνα 2.5 Φαινόμενο φωτιστέφανου στο μικροσκόπιο αντίθεσης φάσης

ΚΕΦΑΛΑΙΟ 3

ΣΤΟΙΧΕΙΑ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΩΝ ΜΕ MATLAB

3.1 Η ψηφιακή εικόνα

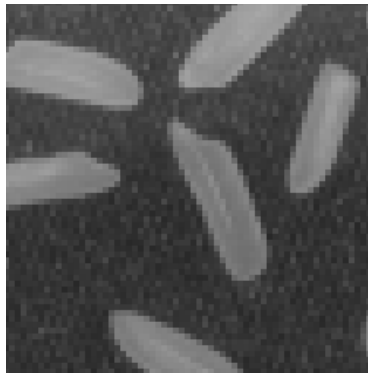
Η εικόνα σήμερα αποτελεί μία από τις σημαντικότερες πηγές πληροφορίας. Τη συναντούμε ως ακίνητη (φωτογραφία), ή κινούμενη (video), ασπρόμαυρη ή έγχρωμη. Η ψηφιακή εικόνα αποτελεί ό,τι πιο σύγχρονο, τόσο στο χώρο της ενημέρωσης (Internet), και της εκπαίδευσης (multimedia) όσο και στο χώρο του θεάματος της ψυχαγωγίας αλλά και της υγείας (ψηφιακή τηλεόραση, DVD κλπ.). [6]

Η εικόνα μπορεί να οριστεί ως μια δισδιάστατη συνάρτηση $f(x,y)$, όπου τα x,y είναι οι χωρικές συντεταγμένες (συντεταγμένες επιπέδου) και το πλάτος της f σε κάθε ζευγάρι συντεταγμένων x,y ονομάζεται ένταση της εικόνας στο σημείο αυτό, η f . Ο όρος επίπεδο του γκρι συχνά χρησιμοποιείται στις μονοχρωματικές εικόνες. Επίσης η εικόνα μπορεί να θεωρηθεί πως είναι η κατανομή της πληροφορίας στο επίπεδο (x,y) , έτσι η f περιγράφει μια επιφάνεια, η οποία έχει μεγάλη τιμή όπου η εικόνα είναι πιο λευκή ενώ για μικρές τιμές της f η εικόνα πλησιάζει το μαύρο. Στην ουσία η f περιγράφει την αμαύρωση της εικόνας σε κάθε θέση (x,y) .

Πρακτικά, κάθε εικόνα για να υποστεί ψηφιακή επεξεργασία θα πρέπει κατ' αρχήν να μετατραπεί σε ψηφιακή. Έτσι θα πρέπει να λάβουμε ισαπέχοντα δείγματα της συνάρτησης $f(x,y)$ στις θέσεις x και y . Η πυκνότητα με την οποία θα ληφθούν τα δείγματα καθορίζεται από το **θεώρημα δειγματοληψίας**:

Η απόσταση δύο διαδοχικών δειγμάτων στο επίπεδο (x,y) θα πρέπει να είναι μικρότερη από την ημιπερίοδο των ταχύτερων εναλλαγών της συνάρτησης $f(x,y)$. Με άλλα λόγια θα πρέπει να δειγματοληπτούμε αρκετά γρήγορα ώστε να προλαβαίνουμε τις γρήγορες εναλλαγές της αμαύρωσης της εικόνας. Στη συνέχεια τα δείγματά μας θα πρέπει να κβαντιστούν σε πεπερασμένο αριθμό σταθμών.

Συνήθως οι στάθμες που χρησιμοποιούνται είναι $255 (2^8 - 1)$. Στη στάθμη 255 αντιστοιχούμε το λευκό ενώ στη στάθμη 0 το μαύρο. Έτσι, ένας δυαδικός αριθμός των 8 bits (1 byte) επαρκεί για να περιγράψουμε την τιμή της αμαύρωσης ενός δείγματος της εικόνας που καλείται εικονοστοιχείο (picture element- pixel). Στην τελική της μορφή μια ασπρόμαυρη ψηφιακή εικόνα θα είναι όπως φαίνεται στην Εικόνα 3.1 [6]



Εικόνα 3.1: Η ψηφιακή εικόνα αποτελείται από μικρές κουκίδες τα εικονοστοιχεία (pixels). Τα λευκά εικονοστοιχεία αντιστοιχούν στο 255 ενώ τα μαύρα στο 0. Κάθε ένα εικονοστοιχείο είναι ένα δείγμα από τη συνάρτηση $f(x,y)$ που αντιστοιχεί στην αναλογική εικόνα.

Μια ψηφιακή εικόνα παριστάνεται μαθηματικά ως η κβαντισμένη σε πλάτος συνάρτηση :

$$f \rightarrow f_q(n_1, n_2)$$

όπου οι διακριτές χωρικές μεταβλητές n_1 και n_2 αντιστοιχούν στις συνεχείς χωρικές μεταβλητές x και y . Στη γενική περίπτωση, μη μέγιστη τιμή του n_1 είναι M και του n_2 είναι η N . Έτσι μπορούμε να θεωρήσουμε την ψηφιακή εικόνα ως μία μήτρα διαστάσεων $M \times N$ αριθμών που κάθε ένας από αυτούς εκπροσωπεί την τιμή του αντίστοιχου εικονοστοιχείου.

Σε ειδικού τύπου εικόνες, όπως το κείμενο ή το σχέδιο, αρκούν δύο μόνο στάθμες, οι 0 και 1, που αντιστοιχούν στο μαύρο και το λευκό. Στην περίπτωση αυτή ένα μόνο δυαδικό ψηφίο είναι αρκετό για να αναπαραστήσει την πληροφορία.

Σε περίπτωση που η εικόνα είναι **έγχρωμη** τότε σε κάθε θέση (n_1, n_2) η έχει τρεις τιμές (άνυσμα), οι οποίες αντιστοιχούν στις τιμές των χρωμάτων κόκκινο, πράσινο και μπλε (Red, Green and Blue –RGB). Όταν, τέλος, έχουμε διαδοχή εικόνων η συνάρτηση έχει μια επιπλέον μεταβλητή, το χρόνο t . Το σήμα είναι διακριτό και ως προς τον χρόνο ($t \rightarrow n_3$) [6]:

$$f \rightarrow f_q(n_1, n_2, n_3)$$

3.2 Επεξεργασία ψηφιακής εικόνας

Ψηφιακή Επεξεργασία Εικόνων (ΨΕΕ) είναι η εφαρμογή της ψηφιακής επεξεργασίας σημάτων πάνω σε συγκεκριμένα σήματα (εικόνες). Η επεξεργασία, μετάδοση και κατανόηση των εικόνων αποτελούν πεδία συνεχώς αναπτυσσόμενης έρευνας. Το μέγεθος μιας εικόνας απαιτεί τεράστια ταχύτητα υλοποίησης των αλγορίθμων για λειτουργία σε πραγματικό χρόνο. Η τεχνολογία ολοκληρωμένων κυκλωμάτων πολύ μεγάλης κλίμακας (VLSI), σε συνδυασμό με την ανάπτυξη αρχιτεκτονικών συνεχούς ροής (pipelining) με μεγάλο βαθμό παραλληλισμού, βοήθησε στη δυνατότητα υλοποίησης πολλών πολύπλοκων αλγορίθμων. Η ταυτόχρονη ελάττωση του κόστους των μνημών, επεξεργασιών, και γενικά της υπολογιστικής ισχύος, έχει κάνει οικονομικά βιώσιμη την ανάπτυξη συστημάτων επικοινωνίας και, επεξεργασίας εικόνων ακόμα και για οικιακή χρήση.

Οι αλγόριθμοι επεξεργασίας εικόνων εξυπηρετούν διάφορους σκοπούς σαν τους επόμενους:

α) την βελτίωση της ποιότητας των εικόνων, με χρήση κατάλληλων φίλτρων η την αποκατάσταση τους στην αρχική τους μορφή μετά από αλλοίωση τους λόγω επίδρασης θορύβου.

β) την κωδικοποίηση τους, έτσι ώστε η πληροφορία τους να μπορεί να περιγραφεί από μία σειρά όσο γίνεται μικρότερου αριθμού bit (συμπύεση δεδομένων) N με σκοπό την γρήγορη μετάδοση τους μέσω διαύλων περιορισμένης χωρητικότητας (bandwidth), ή την αποτελεσματική αποθήκευση τους σε περιορισμένο αποθηκευτικό χώρο με ικανοποιητική ποιότητα εικόνας.

γ) την μετατροπή φωτογραφιών σε εικόνες δύο μόνο αποχρώσεων (μαύρου-άσπρου), για εκτύπωση ή επίδειξη σε δυαδική μορφή.

δ) την τροποποίηση των εικόνων (π.χ. pixelate) εφαρμόζοντας έτσι επάνω τους «καλλιτεχνικές» φόρμες και απόψεις.

3.3 Οι τύποι των εικόνων και η δομή τους στο Matlab

Στην παρούσα Διπλωματική Εργασία η επεξεργασία των εικόνων βασίστηκε στην πλατφόρμα του MATLAB®. Η επεξεργασία και η ανάλυση των κυτταροπαθολογικών εικόνων βασίστηκε σε προγράμματα που υλοποιήθηκαν σε MATLAB®.

Η βασική δομή της πληροφορίας στο MATLAB® είναι ο πίνακας, ένα διατεταγμένο σύνολο πραγματικών ή φανταστικών αριθμών. Αυτό το αντικείμενο εξυπηρετεί την αναπαράσταση των εικόνων, οι οποίες είναι διατεταγμένα σύνολα χρωμάτων και εντάσεων του φωτός. Τα στοιχεία των πινάκων αυτών αποτελούνται αποκλειστικά και μόνο από πραγματικές τιμές αφού το MATLAB® δεν υποστηρίζει εικόνες πινάκων φανταστικών τιμών.

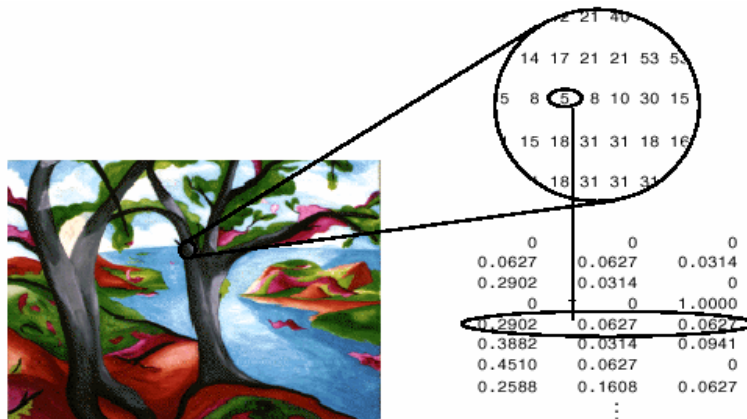
Το MATLAB® αποθηκεύει τις περισσότερες εικόνες σαν δισδιάστατους πίνακες, στους οποίους κάθε στοιχείο του πίνακα αναφέρεται σε ένα και μοναδικό pixel της εικόνας. Η λέξη pixel προέρχεται από τις λέξεις picture element (στοιχείο εικόνας) και συνήθως αναφέρεται σε μια κουκίδα (dot) της οθόνης του υπολογιστή. Αυτή η συμβατικότητα κάνει την επεξεργασία εικόνων με το MATLAB® όμοια με οποιαδήποτε άλλη εργασία σε πίνακες. [1]

Το MATLAB® υποστηρίζει 3 βασικούς τύπους εικόνας:

- **Ενδεικτικές εικόνες (indexed images)**
- **Ασπρόμαυρες εικόνες (grayscale /intensity images)**
- **Έγχρωμες εικόνες RGB**

3.3.1 Ενδεικτικές εικόνες (indexed images)

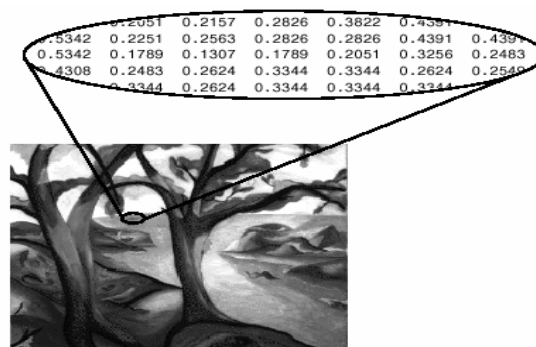
Μια indexed εικόνα αποτελείται από ένα πίνακα δεδομένων X , και ένα πίνακα χρωμάτων-παλέτα (color map), map. Ο map είναι ένας $m \times 3$ πίνακας κλάσης double ο οποίος περιέχει κινητής υποδιαστολής (floating-point) τιμές εύρους $[0,1]$ και υπάρχει αποθηκευμένος μέσα στο MATLAB®. Κάθε μια από τις γραμμές του map καθορίζει τα κόκκινα, πράσινα και μπλε συστατικά κάθε χρώματος αντίστοιχα. Μια ενδεικτική εικόνα χρησιμοποιεί “direct mapping” των τιμών του pixel σε color map τιμές. Το χρώμα κάθε pixel της εικόνας καθορίζεται χρησιμοποιώντας την ανταποκρινόμενη τιμή του X σαν ένδειξη στον map. Η τιμή 1 δείχνει την πρώτη γραμμή του map, η 2 την δεύτερη κ.ο.κ. [1]



Εικόνα 3.2 Παράδειγμα indexed εικόνας

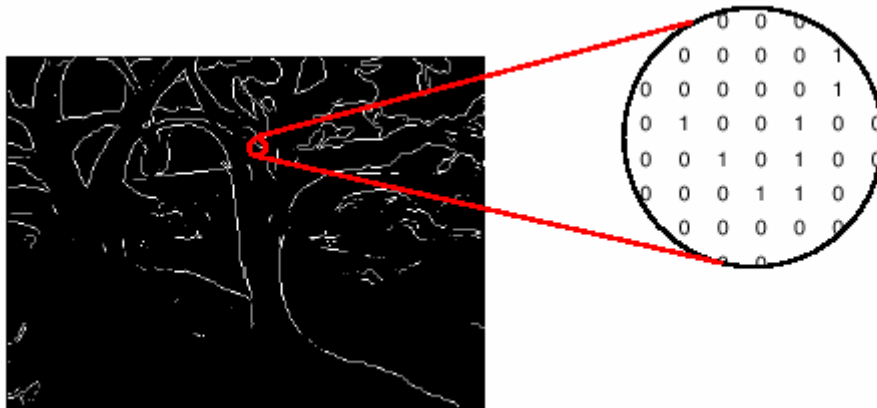
3.3.2 Ασπρόμαυρες εικόνες (grayscale /intensity images)

Μια ασπρόμαυρη εικόνα (εικόνα έντασης) είναι ένας πίνακας δεδομένων I του οποίου οι τιμές αναπαριστούν την ένταση του φωτός. Το MATLAB® αποθηκεύει μια εικόνα έντασης σε ένα απλό πίνακα, του οποίου κάθε στοιχείο αναφέρεται σε ένα και μοναδικό pixel. Ο πίνακας μπορεί να είναι κλάσης double, uint8 ή uint16. Τα στοιχεία του πίνακα αναπαριστούν διάφορες εντάσεις του φωτός (επίπεδα του γκρι) όπου η ένταση 0 αναπαριστά το μαύρο και η ένταση 1, 255 ή 65535 αναπαριστά το λευκό. [3]



Εικόνα 3.3 (α) Παράδειγμα ασπρόμαυρης εικόνας

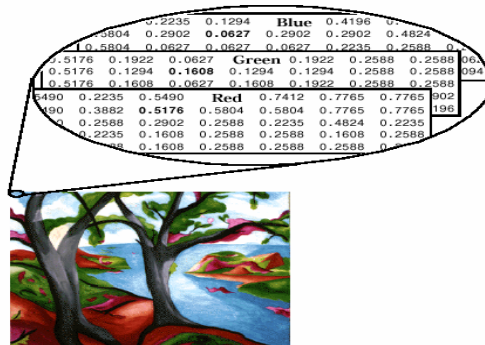
Ένα ειδικό είδος ασπρόμαυρης εικόνας η οποία όμως περιέχει μόνο μαύρο και άσπρο είναι η δυαδική. Σε μια **δυαδική εικόνα** κάθε pixel προέρχεται από μια από τις δυο διακριτές τιμές (0 ή 1). Βασικά αυτές οι δυο τιμές αναφέρονται σε on ή off. Μια δυαδική εικόνα αποθηκεύεται σε ένα διδιάστατο πίνακα μηδενικών (off pixels) και μονάδων (on pixels). [11]



Εικόνα 3.3 (β) Παράδειγμα δυαδικής εικόνας

3.3.3 Έγχρωμες εικόνες RGB

Μια **RGB εικόνα**, ορισμένες φορές αναφέρεται και σαν “truecolor” (πραγματικού χρώματος) εικόνα, αποθηκεύεται στο MATLAB® σαν ένα πίνακα δεδομένων $m \times n \times 3$ ο οποίος καθορίζει τα κόκκινα, πράσινα και μπλε χρωματιστά στοιχεία κάθε ανεξάρτητου pixel. Οι RGB εικόνες δεν χρησιμοποιούν χάρτη χρωμάτων (color map). Το χρώμα του κάθε pixel καθορίζεται από τον συνδυασμό των κόκκινων, πράσινων και μπλε εντάσεων. Ένας RGB MATLAB® πίνακας μπορεί να είναι κλάσης double, uint8 ή uint16. Σε ένα πίνακα κλάσης double κάθε χρώμα είναι μια τιμή στον πίνακα μεταξύ 0 και 1. Ένα pixel του οποίου το χρώμα έχει τιμή (0,0,0) παρουσιάζει το μαύρο ενώ ένα pixel το οποίο έχει τιμή μέσα στον πίνακα (1,1,1) παρουσιάζει το λευκό. Τα στοιχεία των τριών χρωμάτων για κάθε pixel αποθηκεύονται στις τρεις διαστάσεις του πίνακα. [3]



Εικόνα 3.4 Παράδειγμα έγχρωμης RGB εικόνας

3.4 Χρήσιμες συναρτήσεις ανάλυσης και επεξεργασίας εικόνας στο Matlab

1. Εισαγωγή εικόνας στο MATLAB

imread('filename')

2. Εμφάνιση εικόνας

imshow(f,G)

Όπου f η εικόνα και G ο αριθμός των επιπέδων της έντασης της εικόνας, αν το G παραληφθεί αυτόματα χρησιμοποιούνται 256 επίπεδα.

3. Αποθήκευση εικόνας σε αρχείο

imwrite(f,'filename')

4. Μετατροπή μιας έγχρωμης εικόνας σε ασπρόμαυρη

Η εντολή μετατροπής μιας εικόνας αποθηκευμένη σ'ένα πίνακα f σε ασπρόμαυρη είναι :

rgb2gray(f)

5. Ιστόγραμμα εικόνας

Το ιστόγραμμα μιας εικόνας είναι ένας πίνακας ο οποίος παρουσιάζει τις μεταβολές της έντασης σε μια ασπρόμαυρη ή ενδεικτική εικόνα.

Imhist(f)

Η εντολή **imhist** δημιουργεί ένα σχεδιάγραμμα το οποίο περιέχει η κάθετες γραμμές. Η κάθε γραμμή αναπαριστά ένα εύρος τιμών.

3.5 Βελτιστοποίηση εικόνας

Η Βελτίωση της ποιότητας της εικόνας (image enhancement) αποσκοπεί στο να δώσει στην εικόνα μια μορφή περισσότερο κατάλληλη για περαιτέρω επεξεργασία ή ανάλυση. Υπάρχουν δύο βασικοί τύποι μεθόδων βελτίωσης ποιότητας, που επεξεργάζονται την εικόνα είτε στο πεδίο του χώρου (spatial domain) ή της συχνότητας (frequency domain).

Στο πεδίο του χώρου, οι τεχνικές εκμεταλεύονται τις γεωμετρικές ιδιότητες της εικόνας, ενώ στο πεδίο της συχνότητας επεξεργάζονται το φάσμα της εικόνας. Οι σημαντικότερες τεχνικές βελτιστοποίησης αναπτύσσονται σύντομα στη συνέχεια. [2]

3.5.1 Τεχνικές τοπικής επεξεργασίας ή επεξεργασίας σημείου

Οι τεχνικές αυτές αναφέρονται στα επόμενα θέματα:

3.5.1.1 Αντιστοίχιση φωτεινότητας

i) Μετασχηματισμοί του γκρι επιπέδου των pixels-αύξηση και μείωση της κλίμακας του γκρι

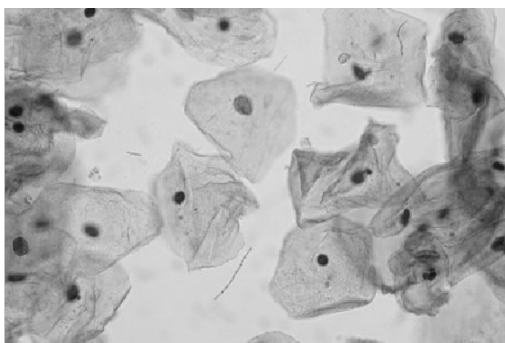
Η αλλαγή της κλίμακας των τιμών των pixel επιτυγχάνεται πολλαπλασιάζοντας με μια σταθερά, πρακτικά είναι η διαδικασία αλλαγής της αντίθεσης. Το MATLAB διαθέτει μια μόνο συνάρτηση που πετυχαίνει διάφορες επιδράσεις στην αντίθεση της εικόνας, τη συνάρτηση **imadjust**.

```
>>g=imadjust(f,[low_in, high_in],[low_out,high_out],gamma);
```

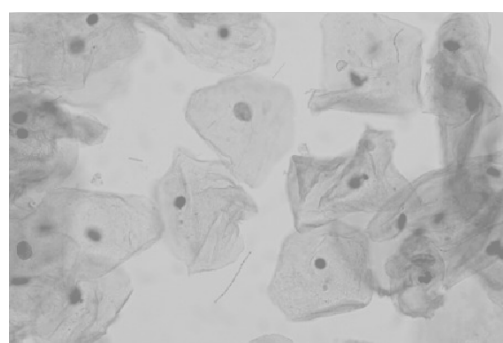
Η συνάρτηση **imadjust** αλλάζει τα όρια των τιμών έντασης της εικόνας σε νέες τιμές που προσδιορίζονται από τα `low_out`, `high_out` [9]

Παράδειγμα:

```
>>f=imread('image.jpg'); %RGB image
>>f=rgb2gray(f);
>>g=imadjust(f,stretchlim(f),[0.5 0.8]);
```



(α)



(β)

Εικόνα 3.5 (α) Αρχική γκρι εικόνα κυττάρων test pap(grayscale) **f**
(β) προσαρμοσμένη εικόνα **g1**

ii) Εξισορρόπηση ιστογράμματος(Histogram equalization)

Στα μαθηματικά, το ιστόγραμμα είναι η συνάρτηση $h(r_k)=n_k$, όπου n_k είναι ο αριθμός των εικονοστοιχείων που έχουν φωτεινότητα r_k .

Το ιστόγραμμα αποδίδεται με την κατανομή των αποχρώσεων του γκρι στην εικόνα και αντιστοιχεί σε ένα γράφημα το οποίο στον οριζόντιο άξονα έχει τις φωτεινότητες στο διάστημα $[0,L-1]$, με $L=2^q$, όπου q είναι τα bit της γκρι εικόνας(βάθος χρώματος). Έστω ότι η μεταβλητή r συμβολίζει τις διαφορετικές τιμές τόνου γκρι, και για $q=8$ παίρνει τιμές από 0 ως 255.

Διαιρώντας με το συνολικό πλήθος των εικονοστοιχείων παίρνουμε το κανονικοποιημένο ιστόγραμμα ή αλλιώς τη συνάρτηση πυκνότητας πιθανότητας (Probability Density Function-PDF).

Για $0 \leq k \leq 255$, η r_k συμβολίζει τις διαφορετικές τιμές τόνου γκρι(grey levels). Η πιθανότητα να εμφανιστεί η τιμή διαβάθμισης του γκρι r_k στην εικόνα είναι: [5], [9]

$$P_r(r_k) = \frac{n_k}{n} \quad (3.1)$$

όπου n είναι το συνολικό πλήθος εικονοστοιχείων της εικόνας.

Το γράφημα αυτής της συνάρτησης(3.1) παρέχει γενικές πληροφορίες για τη μορφή της εικόνας, όπως το δυναμικό εύρος και η αντίθεση της εικόνας (σκοτεινή, φωτεινή, υψηλής ή χαμηλής αντίθεσης). Η μορφή του ιστογράμματος της εικόνας παρέχει σημαντική πληροφορία για τη δυνατότητα ενίσχυσης αντίθεσης της εικόνας.

Η εμφάνιση της εικόνας είναι δυνατό να αλλάξει σημαντικά με την εφαρμογή κατάλ-ληλου μετασχηματισμού που μεταβάλλει τη συνάρτηση πυκνότητας πιθανότητας των χρωματικών πυκνοτήτων (ιστόγραμμα της εικόνας). Η πιο σημαντική τεχνική αυτής της κατηγορίας ονομάζεται εξισορρόπηση ή γραμμικοποίηση ιστογράμματος και αναφέρεται στην εφαρμογή μετασχηματισμού της μορφής: [5]

$$S_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j) \quad (3.2)$$

που παράγει επίπεδο s_k για κάθε τιμή εικονοστοιχείου r_k .

Το αποτέλεσμα του παραπάνω μετασχηματισμού είναι μια εικόνα με ομοιόμορφη κατανομή χρωματικών πυκνοτήτων. Ο μετασχηματισμός αυτό ισοδυναμεί με αύξηση του δυναμικού εύρους των εικονοστοιχείων και μπορεί να προκαλέσει σημαντική μεταβολή στην εμφάνιση της εικόνας.

Στο MATLAB η εξισορρόπηση ιστογράμματος επιτυγχάνεται με την εντολή **histeq**

<<g=histeq(f,nlev)

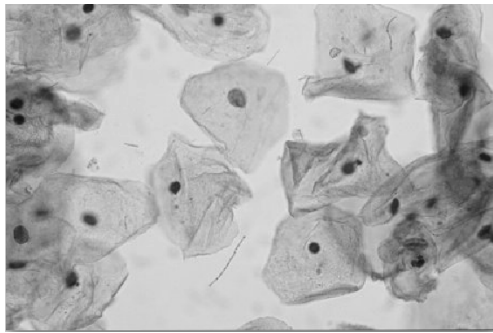
Όπου f είναι η αρχική εικόνα(γκρι) και $nlev$ είναι τα επίπεδα του γκρι για την τελική εικόνα εξόδου(αν αφήσουμε κενό το $nlev$ αυτό παίρνει την τιμή $nlev=64$)

Η εντολή που μας δίνει το ιστόγραμμα μιας εικόνας στο MATLAB είναι η **imhist**

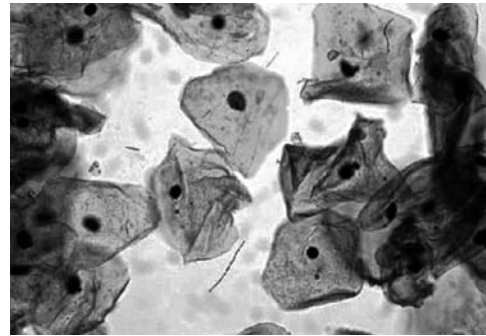
<<imhist(f,b)

Όπου f είναι η εικόνα, h το ιστόγραμμά της και b ο αριθμός που θα διαιρεθούν τα επίπεδα του γκρι, για παράδειγμα αν το $b=2$ τότε η κλίμακα των τόνων του γκρι θα χωριστεί σε 2 περιοχές: 0-127 και 128-255, και το ιστόγραμμα θα έχει 2 τιμές $h(1)$ που θα αντιστοιχεί στη περιοχή $[0,127]$ και $h(2)$ που θα αντιστοιχεί στη περιοχή $[128,255]$. Η προεπιλεγμένη τιμή αν μείνει κενό το πεδίο του b είναι $b=256$.

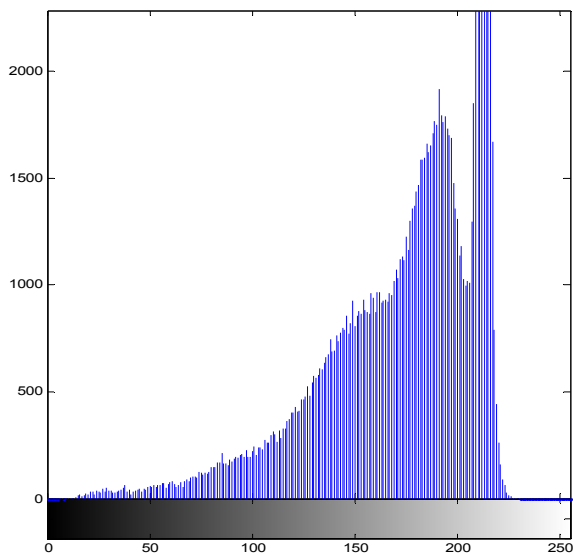
Στην εικόνα 3.6 φαίνεται η αρχική εικόνα με την τελική εικόνα μετά την εξισορρόπηση ιστογράμματος καθώς και τα αντίστοιχα ιστογράμμά τους.



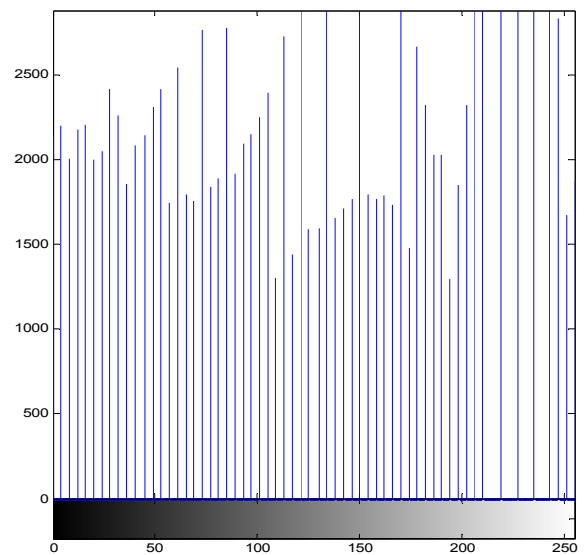
(α)



(β)



(γ)



(δ)

Εικόνα 3.5

(α) Αρχική γκρι εικόνα **f**

(β) Τελική εικόνα μετά την εξίσωση ιστογράμματος **g2**

(γ) Ιστόγραμμα αρχικής εικόνας **f**

(δ) Ιστόγραμμα τελικής εικόνας (nlev=256) **g2**

3.5.1.2 Εξομάλυνση (smoothing) της εικόνας

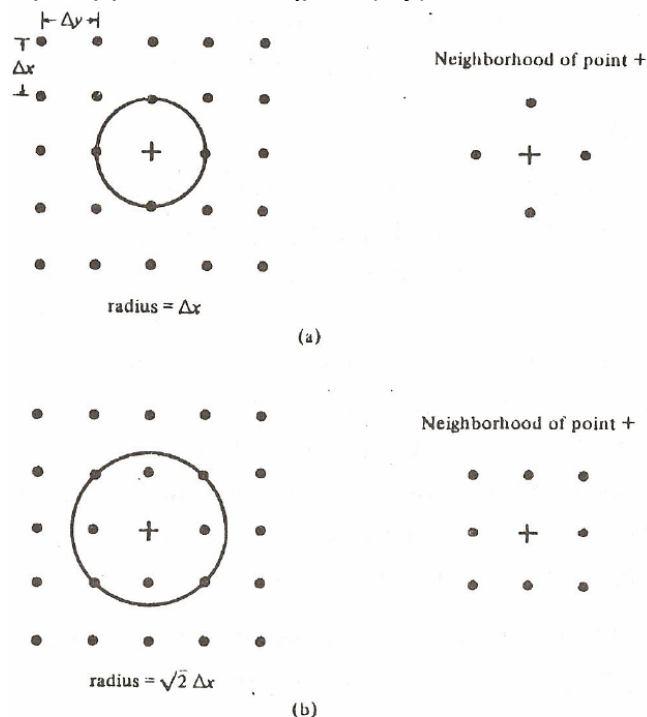
i) Φίλτρο μέσου όρου (mean or average)

Το φίλτρο αυτό επιδρά σε μια εικόνα $f(x,y)$ και δημιουργεί μια άλλη $g(x,y)$ με βάση τον τύπο:

$$g(x,y) = \frac{1}{M} \sum_{m,n \in S} f(m,n)$$

(3.3)

όπου S είναι μια περιοχή γύρω από το σημείο (x,y) , που περιλαμβάνει M σημεία, μεταξύ των οποίων ίσως και το ίδιο το (x,y) . Η εικόνα 3.7 δείχνει δύο τέτοιες περιοχές που δεν περιλαμβάνουν το σημείο (x,y) .



Εικόνα 3.7

Το φίλτρο μέσου όρου προκαλεί μεν εξομάλυνση του θορύβου, αλλά ταυτόχρονα λειαίνει (θολώνει, blur) και τις ακμές, τόσο περισσότερο, όσο μεγαλύτερη είναι η ακτίνα της περιοχής γύρω από το σημείο (x,y) . Μείωση του φαινομένου του blur μπορεί να γίνει αν χρησιμοποιηθεί μία διαδικασία κατωφλίωσης. Μία απλή τέτοια διαδικασία η οποία χρησιμοποιεί ένα κατώφλι T , αφού υπολογίσει τον μέσο όρο των σημείων σύμφωνα με τα προηγούμενα, περιλαμβάνει στην περιοχή S μόνο τα σημεία η διαφορά των οποίων από τον μέσο όρο είναι μικρότερη από T . [2]

Η υλοποίηση του φίλτρου μέσου όρου στο MATLAB γίνεται με την εντολή `fspecial`, με 2 τρόπους όπως φαίνεται παρακάτω:

`h1=fspecial('average',[r c])` : τετραγωνικό φίλτρο μέσου όρου

`h2=fspecial('disk',r)` : κυκλικό φίλτρο μέσου όρου (σε τετράγωνο $2r+1$)

Η εντολή που υλοποιεί το γραμμικό φιλτράρισμα στο MATLAB είναι η `imfilter`:
`g=imfilter(f,h,filtering_mode,boundary_options,size_options)`

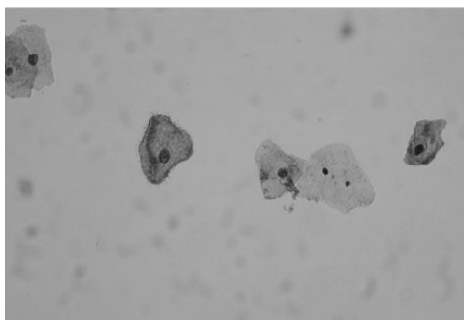
όπου:

f: είναι η αρχική εικόνα, h: το φίλτρο(μάσκα φίλτρου), filtering_mode: είναι για το αν το φιλτράρισμα θα γίνει με συνέλιξη(conv) ή με συσχέτιση(corr), boundary_options: έχει να κάνει με το γέμισμα στο σύνορο της εικόνας.

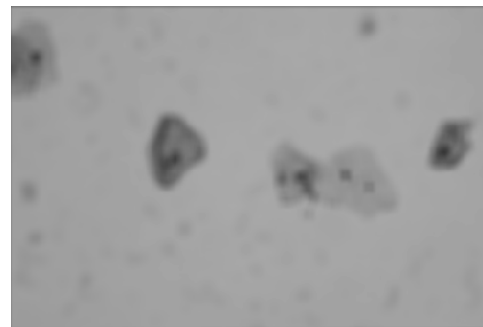
Παράδειγμα:

Εφαρμόζουμε το τετραγωνικό φίλτρο h1 στην αρχική κυτταρική εικόνα f και στη συνέχεια το κυκλικό με r=5, τα αποτελέσματα φαίνονται στην εικόνα 3.8

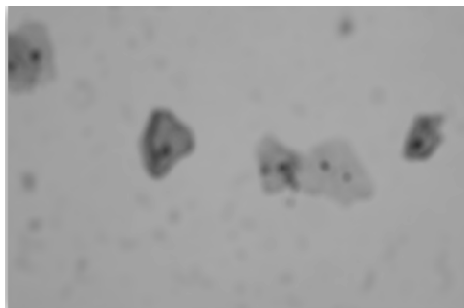
```
f=imread('image');
f=rgb2gray(f);
h1=fspecial('average',[10 10]);
g1=imfilter(f,h1,'replicate')
h2= fspecial('disk',5);
g2=imfilter(f,h2,'replicate')
```



(α)



(β)



(γ)

Εικόνα 3.8 (α) αρχική εικόνα f
(β) τελική φιλτραρισμένη εικόνα με τετραγωνικό φίλτρο μέσου όρου [10 10]
(γ) τελική φιλτραρισμένη εικόνα με κυκλικό φίλτρο μέσου όρου με r=5

3.5.1.3 Όξυνση-τονισμός των ακμών της εικόνας (edge sharpening)

Η άθροιση των τιμών των εικονοστοιχείων (φίλτρο μέσης τιμής) οδηγεί σε θόλωση των λεπτομερειών της εικόνας. Αφού η άθροιση είναι ανάλογη με τη διαδικασία της ολοκλήρωσης, η παραγωγή θα έχει αντίθετο αποτέλεσμα, ενισχύοντας τις λεπτομέρειες της εικόνας. Τα φίλτρα παραγωγής έχουν ως στόχο την έμφαση σε λεπτομέρειες της εικόνας ή τη βελτίωση λεπτομερειών που έχουν θολωθεί είτε κατά λάθος ή σαν συνέπεια συγκεκριμένων μεθόδων συλλογής των δεδομένων. Ο πιο

συνηθισμένος τύπος παραγωγίσης σε εφαρμογές επεξεργασίας εικόνας είναι το διάνυσμα της κλίσης.

Η κλίση μιας ψηφιακής εικόνας $f(x,y)$ ορίζεται από το διάνυσμα: [2], [5]

$$\nabla f(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Το μέτρο αυτού του διανύσματος είναι:

$$\|\nabla f\| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \approx G_x^2 + G_y^2 \approx |G_x| + |G_y|$$

και η κατεύθυνσή του ως προς τον άξονα x είναι:

$$a(x,y) = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

Το διάνυσμα κλίσης αποτελεί τη βάση για διάφορες προσεγγίσεις στην παραγωγή εικόνας. Αν θεωρήσουμε την περιοχή της εικόνας του εικ. 3.9, όπου z_1-z_9 δηλώνουν τις τιμές των χρωματικών πυκνοτήτων, οι μερικές παράγωγοι ως προς x και y στο σημείο z_5 μπορούν να προσεγγιστούν με διάφορους τρόπους

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	-2	-1
0	0	0
1	2	1

$$G_x = (z_7 + 2z_8 + z_9) -$$

-1	0	1
-2	0	2
-1	0	1

$$G_y = (z_3 + 2z_6 + z_9) -$$

Sobel

-1	-1	-1
0	0	0
1	1	1

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) -$$

-1	0	1
-1	0	1
-1	0	1

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) -$$

Prewitt

-1	0
0	1

$$G_x = (z_9 - z_5) -$$

0	-1
1	0

$$G_y = (z_8 - z_6) -$$

Roberts

Εικόνα 3.9 Περιοχή 3x3 μιας εικόνας και μάσκες για τον υπολογισμό παραγώγου στο σημείο z_5

Η εφαρμογή των παραπάνω τελεστών ισοδυναμεί με συνέλιξη της εικόνας με αντίστοιχες μάσκες εικ. 3.9.

Στο MATLAB η συνάρτηση που υλοποιεί διάφορους τελεστές παραγώγου είναι η `edge` η οποία συντάσσεται ως εξής:

$$[g, t] = \text{edge}(f, \text{'method'}, \text{'parameters'})$$

Το method μπορεί να είναι:

- sobel (parameters=T,dir)
- Prewitt(parameters=T,dir)
- Roberts(parameters=T,dir)
- Laplacian of Gaussian(LoG)(parameters=T,sigma)
- Zero crossings(parameters=T,H)
- Canny(parameters=T₀,sigma)

T: Είναι το κατώφλι(threshold) πάνω από το οποίο οι τιμές των τελεστών παραγώγου μπορούν να θεωρηθούν ακμές

Dir: Επιλέγει τη κατεύθυνση των ακμών που ανιχνεύονται

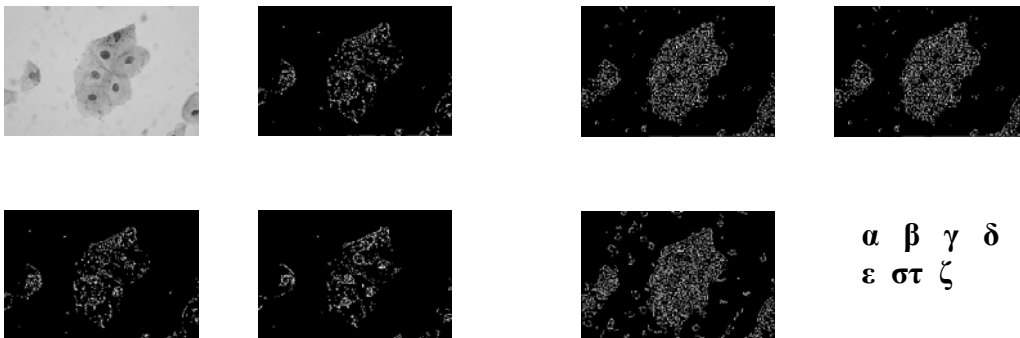
Sigma: είναι η σταθερή απόκλιση

H: συνάρτηση φίλτρου

T₀: T₀=[T1,T2] , T1<T2

Παράδειγμα:

```
clear;
close all;
f=imread('09.jpg');
f=rgb2gray(f);
g1=edge(f,'sobel');
g2=edge(f,'prewitt');
g3=edge(f,'roberts');
g4=edge(f,'log');
g5=edge(f,'zerocross');
g6=edge(f,'canny');
subplot(221);imshow(f);
subplot(222);imshow(g1);
subplot(223);imshow(g2);
subplot(224);imshow(g3);
figure;
subplot(221);imshow(g4);
subplot(222);imshow(g5);
subplot(223);imshow(g6);
```



Εικόνα 3.10 (α) Αρχική εικόνα

Ανίχνευση ακμών με μέθοδο: (β) sobel (γ) prewitt

(δ) Roberts (ε)Laplacian of Gaussian (στ)zero-crossings (ζ) canny

3.5.2 Φιλτράρισμα εικόνας για αφαίρεση θορύβου

Οι ψηφιακές εικόνες επηρεάζονται από διάφορους τύπους θορύβου, και ιδιαίτερα αυτές που προέρχονται από κάποια ιατρική συσκευή. Ο θόρυβος μπορεί να παρουσιαστεί σε μια εικόνα με διάφορους τρόπους, ανάλογα με τη κατασκευή της εικόνας. Για να αφαιρεθεί ο θόρυβος από μια εικόνα χρησιμοποιούμε διάφορα φίλτρα, ανάλογα με το είδος του θορύβου.

Φιλτράρισμα είναι μια τεχνική με την οποία τροποποιούμε ή προβάλλουμε μία εικόνα. Για παράδειγμα, μπορείς να φιλτράρεις μια εικόνα για να τονίσεις ορισμένα χαρακτηριστικά ή για να αφαιρέσεις άλλα. Το φιλτράρισμα είναι μια λειτουργία γειτονικών *pixel*, στην οποία η τιμή κάθε *pixel* της εικόνας που έχει περάσει από το φίλτρο καθορίζεται τοποθετώντας κάποιον αλγόριθμο στις τιμές των γειτονικών *pixel* της εικόνας πριν αυτή περάσει από το φίλτρο.

Τα φίλτρα που χρησιμοποιούνται περισσότερο για την αφαίρεση θορύβου είναι:

- (i) Τα γραμμικά φίλτρα (**Linear filtering**)
- (ii) Ενδιάμεσης τιμής (**median filtering**)
- (iii) Προσαρμοζόμενα (**Adaptive filtering**)

3.5.2.1 Γραμμικά φίλτρα (**Linear filtering**)

Με τον όρο γραμμικό φιλτράρισμα εννοούμε το φιλτράρισμα στο οποίο η τιμή κάθε εξερχόμενου *pixel* είναι ένας γραμμικός συνδυασμός των τιμών των εισερχομένων γειτονικών *pixel*. Για παράδειγμα, ένας αλγόριθμος που υπολογίζει ένα σταθμισμένο μέσο όρο από τα γειτονικά *pixel* είναι ένας τύπος από τη λειτουργία των γραμμικών φίλτρων.

Τα γραμμικά φίλτρα χρησιμοποιούνται για να αφαιρεθούν διάφοροι τύποι θορύβου. Συγκεκριμένα φίλτρα όπως τα φίλτρα Gaussian ή “averaging”, χρησιμοποιούνται γι’ αυτό το σκοπό. Για παράδειγμα, ένα “averaging filter” είναι χρήσιμο για να αφαιρεί κόκκους θορύβου από μια εικόνα. Κάθε *pixel* τοποθετείται στο μέσο των γειτονικών *pixels*, και οι τοπικές αποκλίσεις λόγω των κόκκων μειώνονται. [1]

- **Συνέλιξη (Convolution)**

Στο MATLAB[®], το γραμμικό φιλτράρισμα εικόνων υλοποιείται με δύο-διάστατη **Συνέλιξη**. Στη συνέλιξη, η τιμή ενός εξερχόμενου *pixel* υπολογίζεται από τον πολλαπλασιασμό στοιχείων από δύο πίνακες. Ο ένας από αυτούς τους πίνακες αναπαριστά την εικόνα, καθώς ο άλλος πίνακας είναι το φίλτρο. Η αναπαράσταση αυτού του φίλτρου είναι γνωστή ως “επεξεργαστής συνέλιξης”. Στο MATLAB[®] η **conv2** κάνει φιλτράρισμα εικόνας τοποθετώντας τον επεξεργαστή συνέλιξης στον πίνακα της εικόνας. [1]

- **Η συνάρτηση filter2**

Επιπρόσθετα το MATLAB[®] μας παρέχει την εντολή **filter2** για δυοδιάστατα γραμμικά φίλτρα. Η **filter2** δίνει τα ίδια αποτελέσματα με την **conv2** και διαφέρει στο γεγονός ότι παίρνει το υπολογιστικό μόριο ως εισερχόμενη παράσταση παρά ως επεξεργαστή συνέλιξης. (η **filter2** λειτουργεί εκτελώντας τον επεξεργαστή συνέλιξης από το υπολογιστικό μόριο και μετά καλεί την **conv2**). Η λειτουργία της εντολής **filter2** ονομάζεται «**correlation**». [1]

3.5.2.2 Φίλτρα ενδιάμεσης τιμής (Median filtering)

Στο Φιλτράρισμα με Median Filter (είναι παρόμοιο με ένα averaging filter) κάθε pixel παίρνει την μέση τιμή των γειτονικών pixel. Ωστόσο με median filtering η τιμή ενός εξερχόμενου pixel καθορίζεται από το median των γειτονικών pixel. Η εντολή **medfilt2** θέτει σε εφαρμογή το median filtering. Η εντολή **medfilt2** αφαιρεί το θόρυβο 'salt & pepper' πολύ καλά, χωρίς να θολώσει την εικόνα. [1]

3.5.2.3 Προσαρμοζόμενα φίλτρα (Adaptive filtering)

Η εντολή **wiener2** τροποποιεί τοπικές αντιθέσεις της εικόνας. Όταν οι αντιθέσεις είναι μεγάλες, η **wiener2** εκτελεί λίγη εξομάλυνση ενώ όταν είναι μικρές περισσότερη. Αυτή η προσέγγιση συχνά δίνει καλύτερα αποτελέσματα από τα γραμμικά φίλτρα. Τα προσαρμοστικά φίλτρα είναι πιο επιλεκτικά από ένα γραμμικό φίλτρο προστατεύοντας ακμές και άλλα υψηλής συχνότητας κομμάτια της εικόνας. Η εντολή **wiener2** χρησιμοποιεί προκαταρκτικούς υπολογισμούς και εφαρμόζει το φίλτρο στην εισερχόμενη εικόνα. Όμως χρειάζεται περισσότερο χρόνο για υπολογισμούς από το γραμμικό φιλτράρισμα. Η **wiener2** δουλεύει καλύτερα όταν ο θόρυβος είναι συνεχής και αυξανόμενος ("white") όπως ο θόρυβος τύπου Gaussian. [1]

3.6 Μορφολογική επεξεργασία εικόνας

Η θεωρία της μορφολογίας στα μαθηματικά μπορεί να αναπτυχθεί με πολλούς τρόπους. Εμείς θα ασχοληθούμε με τη μέθοδο εκείνη που χρησιμοποιεί λειτουργίες σε ένα σετ σημείων.

Οι μορφολογικοί τελεστές είναι μέθοδοι για την επεξεργασία δυαδικών ή grayscale εικόνων βασισμένοι στη γεωμετρία. Αυτοί οι τελεστές παίρνουν μια εικόνα ως είσοδο, εφαρμόζουν σε αυτήν ένα τελεστή, και επιστρέφουν την επεξεργασμένη εικόνα ίδιων διαστάσεων με την αρχική. Η τιμή κάθε pixel της εικόνας εξόδου βασίζεται στη σχέση του pixel εισόδου και του γειτονικού του. Ένα ουσιαστικό μέρος των μορφολογικών τελεστών είναι το δομικό στοιχείο (structure element). Αυτό μπορεί να είναι αυθαιρέτου σχήματος και μεγέθους (πχ παρακάτω στις εικόνες 3.11 και 3.12 αυτό είναι τετράγωνο 3x3). Οι μορφολογικοί τελεστές επεξεργάζονται μια εικόνα εφαρμόζοντας το δομικό στοιχείο (σαν μάσκα) στην εικόνα αυτή και χρησιμοποιώντας έναν set operator (τομή, ένωση, κλπ). Οι βασικοί μορφολογικοί τελεστές είναι οι: erosion, dilation, opening και closing. [3]

3.6.1 Erosion(Διάβρωση) και Dilation(Διαστολή)

Erosion (Διάβρωση): Η βασική επίδραση αυτού του τελεστή είναι να διαβρώνει τα σύνορα της περιοχής των pixels που βρίσκονται στο προσκήνιο (π.χ. λευκά pixels), έτσι μικραίνει τα αντικείμενα αφαιρώντας pixel στα σύνορα του αντικειμένου. [3]

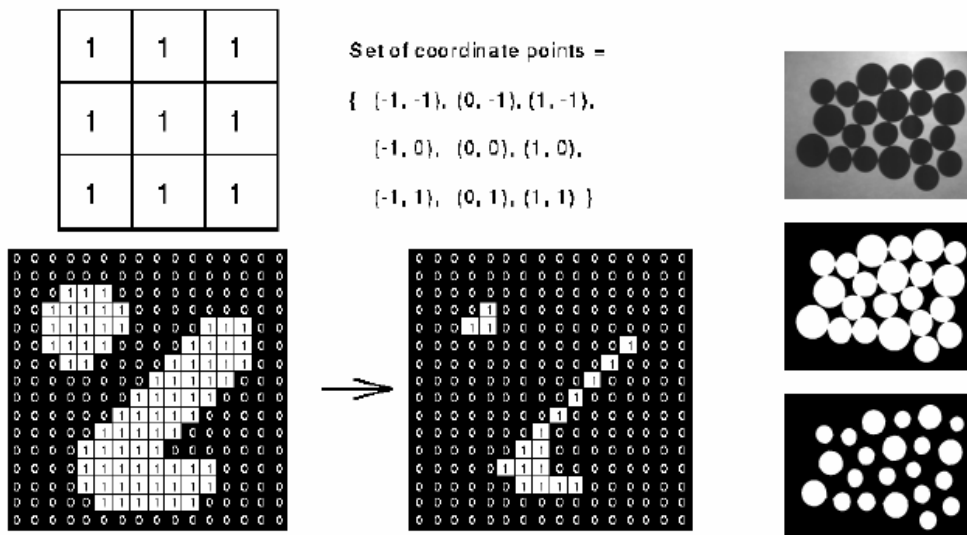


Η δυαδική Διάβρωση μια εικόνας $f(x,y)$ από ένα δομικό στοιχείο $s(x,y)$ ορίζεται από τη παρακάτω σχέση: [12]

$$f \ominus s = \begin{cases} 1 & \text{αν το } s \text{ χωράει στην } f \\ 0 & \text{διαφορετικά} \end{cases}$$

Η σχέση για τη διάβρωση γκρι εικόνας είναι: [12]

$$f \ominus s = \min_{j,k \in s} \{f(m-j, n-k) - s(j,k)\}$$



ΕΙΚΟΝΑ 3.11 Παραδείγματα λειτουργίας διάβρωσης

Dilation (Διαστολή): Η βασική επίδραση πάνω στην εικόνα είναι η διεύρυνση των συνόρων των περιοχών των pixels που βρίσκονται στο προσκήνιο (π.χ. λευκά pixels), έτσι μεγαλώνει τα αντικείμενα προσθέτοντας pixel στα σύνορα των αντικειμένων. [3]

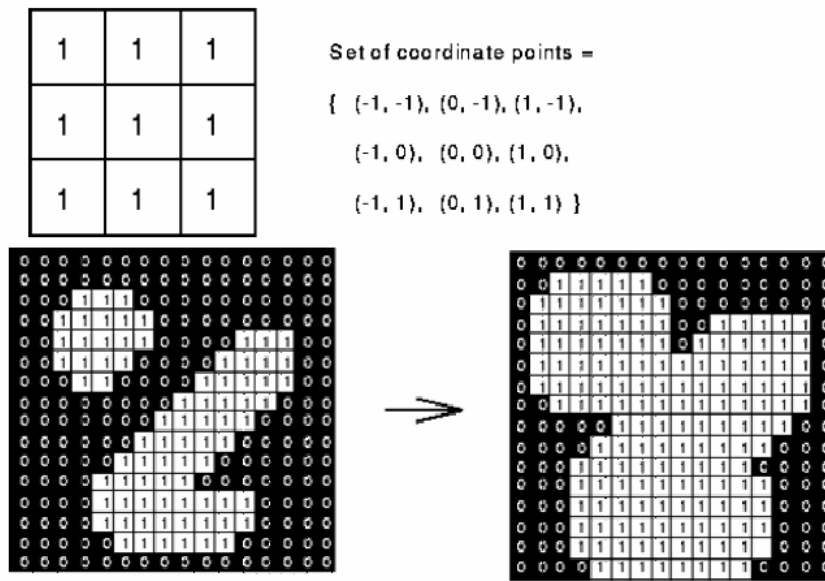


Η δυαδική Διαστολή μια εικόνας $f(x,y)$ από ένα δομικό στοιχείο $s(x,y)$ ορίζεται από τη παρακάτω σχέση: [12]

$$f \oplus s = \begin{cases} 1 & \text{αν το } s \text{ χτυπά στην } f \\ 0 & \text{διαφορετικά} \end{cases}$$

Η σχέση για τη διαστολή γκρι εικόνας είναι: [12]

$$f \oplus s = \max_{j,k \in S} \{f(m-j, n-k) - s(j, k)\}$$



ΕΙΚΟΝΑ 3.12 Παράδειγμα λειτουργίας διαστολής

Τα δομικά στοιχεία (se) διαφοροποιούνται ανάλογα με το αν η εικόνα που εφαρμόζονται είναι δυαδική (επίπεδο se) ή γκρι κλίμακας (μη επίπεδο se). [12]
 Στο MATLAB το δομικό στοιχείο δημιουργείται με την εντολή strel:

>>se=strel(shape,parameters)

Πίνακας 3.1

Shape	Parameters	Περιγραφή
diamond	R	Απόσταση από κέντρο στο μέγιστο άκρο του διαμαντιού
disk	R	Ακτίνα δίσκου
line	length, orientation	Μήκος στοιχείου και γωνία ως προς οριζόντιο άξονα(αντί-ωρολογιακά)
octagon	R	Μήκος από κέντρο σε μια κάθετη πλευρά, πρέπει να είναι πολ/σιο του 3
pair	offset	Πίνακας 2 στοιχείων που προσδιορίζει τη θέση του δεύτερου σημείου
periodicline	P, V	Δομικό στοιχείο 2*P+1 στοιχείων, το V είναι 2 στοιχείων πίνακας Το πρώτο στοιχείο είναι στην αρχή(1*V), το 2 στο -1*V, και τα υπόλοιπα στις θέσεις: 2*V, 2*V, ..., ..., P*V,-P*V
rectangle	MN	2 διαστάσεων πίνακας
square	W	Η διάσταση του τετραγώνου
arbitrary	NHOOD	Δημιουργεί αυθαίρετο δομικό στοιχείο se, NHOOD είναι πίνακας 0 και 1 που καθορίζουν το σχήμα

Τα μη-επίπεδα δομικά στοιχεία δημιουργούνται βάζοντας στην εντολή strel 2 πίνακες, ο πρώτος ορίζει τη γειτονιά(neighborhood), και ο δεύτερος ορίζει το ύψος σε κάθε σημείο της γειτονιάς.

Στο MATLAB, η διάβρωση και η διαστολή των δυαδικών και των γκρι εικόνων επιτυγχάνονται με τις εντολές:

>>dst=imerode(src,se);

>>dst=imdilate(src, se);

3.6.2 Άνοιγμα(opening) και κλείσιμο(closing)

Opening (Άνοιγμα): Η βασική επίδραση είναι μία erosion η οποία ακολουθείται από μία dilation. Το βασικό μειονέκτημα είναι ότι επιδρά πάνω σε όλες τις περιοχές των pixels σε προσκήνιο, αδιάκριτα. Υλοποιείται με μια διάβρωση της f απ'το s , ακολουθούμενη από μια διαστολή του αποτελέσματος από το s : [3]

$$f \circ s = (f \ominus s) \oplus s$$

Closing (Κλείσιμο): Η βασική επίδραση είναι μία dilation η οποία ακολουθείται από μία erosion. Υλοποιείται με μια διαστολή της f απ'το s , ακολουθούμενη από μια διάβρωση του αποτελέσματος από το s : [3]

$$f \cdot s = (f \oplus s) \ominus s$$

Αν χρησιμοποιήσουμε opening και closing μαζί σε μια grayscale εικόνα μπορούμε να επιτύχουμε ομαλοποίηση(smoothing) σε κάποιο βαθμό.

Στο MATLAB οι λειτουργίες opening και closing επιτυγχάνονται με τις παρακάτω εντολές: [9]

```
>>dst=imopen(src,se);
```

```
>>dst=imclose(src,se);
```

Σημείωση:

Οι `open(src)` και `close(src)` μπορούν να χρησιμοποιηθούν και υποθέτουν ένα 3×3 `se`)

3.6.3 Βασικές μορφολογικές συναρτήσεις στο Matlab

3.6.3.1 Ο μετασχηματισμός Hit-or-Miss

Συχνά είναι χρήσιμο να μπορούμε να αναγνωρίσουμε διάφορες τοπολογίες pixel, όπως απομονωμένα pixel στο φόντο. Ο μετασχηματισμός hit-or-miss είναι χρήσιμος σε τέτοιες εφαρμογές και υλοποιείται με χρήση διπλού δομικού στοιχείου $s=(s_1,s_2)$, και ορίζεται ως εξής: [9]

$$f \otimes s = (f \ominus s_1) \cap (f \ominus s_2)$$

Στο MATLAB υλοποιείται με τη συνάρτηση **bwhitmiss**:

```
>>C=bwhitmiss(f,s1,s2)
```

Παράδειγμα: Hit-or-miss στο Matlab

Έστω μια εικόνα f (δυαδική) που φαίνεται στην εικόνα 3.13α , ορίζουμε τα παρακάτω 2 δομικά στοιχεία s_1,s_2 :

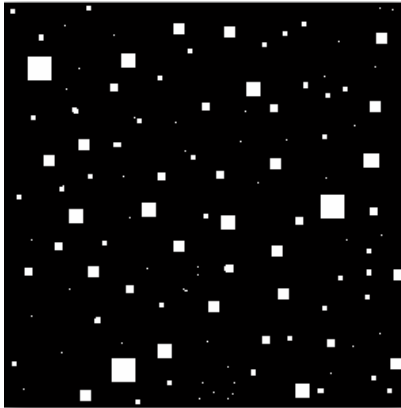
```
>>s1=strel([0 0 0; 0 1 1; 0 1 0]);
```

```
>>s2=strel([1 1 1; 1 0 0; 1 0 0]);
```

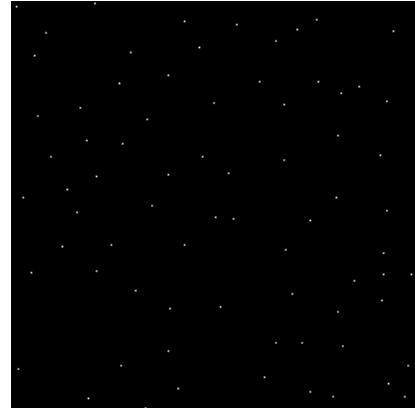
Μετά εφαρμόζουμε το μετασχηματισμό hit-or-miss

```
>>
```

Το αποτέλεσμα φαίνεται στην εικόνα 3.13 β



(α)



(β)

Εικόνα 3.13 Παράδειγμα χρήσης του μετασχηματισμού hit-or –miss

3.6.3.2 Η συνάρτηση δυαδικών εικόνων `bwmorph`

Η `bwmorph` χρησιμοποιεί μια ποικιλία από χρήσιμες λειτουργίες βασισμένες σε συνδυασμό διαστολών και διαβρώσεων. Συντάσσεται ως εξής:

$$g = \text{bwmorph}(f, \text{operation}, n)$$

όπου f είναι μια δυαδική εικόνα, operation είναι η λειτουργία που επιθυμούμε να κάνει (δες πίνακα 3.2), και το n είναι ένας θετικός ακέραιος που προσδιορίζει πόσες φορές θα εφαρμοστεί η λειτουργία. [9]

Πίνακας 3.2

operation	Περιγραφή
bothat	Λειτουργία “bottom-hat” με 3×3 se; <code>imbothat</code> για άλλα se
bridge	ενώνει pixel που χωρίζονται με ένα pixel κενό
clean	σβήνει μεμονωμένα pixel στο φόντο
close	πραγματοποιεί closing με se 3×3
diag	γεμίζει δίπλα από διαγώνια συνδεδεμένα pixel στο προσκήνιο
dilate	πραγματοποιεί dilation με se 3×3
erode	πραγματοποιεί erosion με se 3×3
fill	Γεμίζει τρύπες από μονά pixel; <code>imfill</code> για μεγαλύτερες τρύπες
hbreak	Αφαιρεί H-συνδεδεμένα pixels
majority	κάνε το pixel p pixel προσκήνιου αν τουλάχιστον 5 pixel στο $N_8(p)$ είναι pixel προσκήνιου, αλλιώς κάνε το p pixel φόντου
open	πραγματοποιεί opening με se 3×3
remove	Αφαιρεί τα εσωτερικά pixel (pixel προσκήνιου χωρίς γειτονικά φόντου)
shrink	Μικραίνει τα αντικείμενα χωρίς τρύπα σε ένα σημείο και τα αντικείμενα με τρύπα σε ένα δακτύλιο
skel	πραγματοποιεί skeletonize στην εικόνα
spur	αφαιρεί
thicken	“Παχαίνει” τα αντικείμενα χωρίς να τα συνδέει
thin	“Λεπταίνει” τα αντικείμενα χωρίς τρύπες σε μια γραμμή και με τρύπες σε δακτύλιο
tophat	Λειτουργία “top-hat” με 3×3 se; <code>imtophat</code> για άλλα se

3.6.3.3 Ετικέτες σε συνδεδεμένα αντικείμενα δυαδικών εικόνων

Η συνάρτηση που βάζει ετικέτες στα ξεχωριστά αντικείμενα μιας εικόνας είναι η **bwlabel**:

$$[L,num]=bwlabel(f,conn)$$

όπου f είναι η δυαδική εικόνα και $conn$ είναι ο τρόπος σύνδεσης (4 ή 8). Το L είναι ο πίνακας με τις ετικέτες και το num περιέχει τον αριθμό των αντικειμένων που αναγνωρίστηκαν. [9]

3.6.3.4 Οι μετασχηματισμοί top-hat και bot-hat

A) Μετασχηματισμός top-hat

Αν μια εικόνα στην οποία έχει εφαρμοσθεί opening την αφαιρέσουμε από την αρχική της αυτό ονομάζεται μετασχηματισμός top-hat. Η συνάρτηση `imtophat` επιτυγχάνει το ίδιο αποτέλεσμα σε ένα βήμα:

```
se=strel(disk,10);
```

```
fo=imopen(f,se);  $\longleftrightarrow$  f2=imtophat(f,se)
f2=imsubtract(f,fo);
```

B) Μετασχηματισμός bot-hat

Αν από μια εικόνα στην οποία έχει εφαρμοσθεί closing αφαιρέσουμε την αρχική της αυτό ονομάζεται μετασχηματισμός bot-hat. Η συνάρτηση `imbothat` επιτυγχάνει το ίδιο αποτέλεσμα σε ένα βήμα:

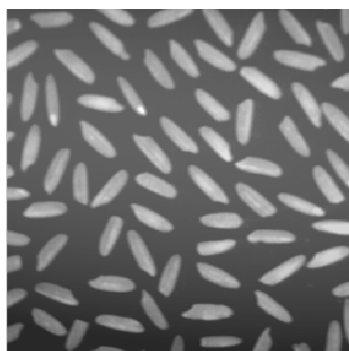
```
se=strel(disk,10);
```

```
fc=imclose(f,se);  $\longleftrightarrow$  f3=imbothat(f,se)
f3=imsubtract(fc,f);
```

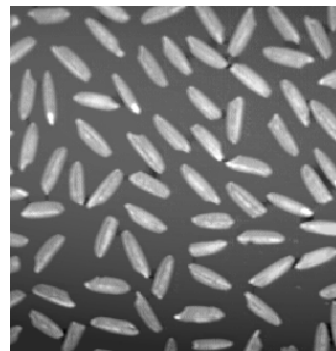
Οι δύο παραπάνω εντολές μπορούν να συνδυαστούν ώστε να επιτευχθεί βελτίωση στην αντίθεση της εικόνας (Εικ 3.14), όπως φαίνεται στις παρακάτω εντολές για παράδειγμα: [9]

```
>>se=strel('disk',3);
```

```
>>g=imsubtract(imadd(f,imtophat(f,se)), imbothat(f,se));
```



(α) Αρχική εικόνα



(β) Εικόνα με αυξημένη αντίθεση

Εικόνα 3.14 Αύξηση αντίθεσης με χρήση `imtophat` και `imbothat`

3.6.4 Μορφολογική ανακατασκευή εικόνας

Η ανακατασκευή είναι ένας μορφολογικός μετασχηματισμός που περιλαμβάνει δύο εικόνες και ένα δομικό στοιχείο. Η μια εικόνα είναι ο δείκτης, και είναι το αρχικό σημείο για το μετασχηματισμό. Η άλλη εικόνα είναι η μάσκα και περιορίζει το μετασχηματισμό. Το δομικό στοιχείο προσδιορίζει τη συνεκτικότητα. [9]

Αν g είναι ο δείκτης(marker) και f η μάσκα(mask), η ανακατασκευή της g από την f , $R_g(f)$, ορίζεται από την ακόλουθη διαδικασία:

1. Αρχικοποίησε την h_1 να είναι η εικόνα-δείκτης f
2. Δημιούργησε ένα δομικό στοιχείο: $s = \text{ones}(3)$
3. Επανάλαβε:

$$h_{k+1} = (h_k \oplus s) \cap g$$

Μέχρι $h_{k+1} = h_k$

Η εικόνα δείκτης f πρέπει να είναι υποσύνολο της g : $f \subseteq g$

Η συνάρτηση που υλοποιεί ανακατασκευή εικόνας στο MATLAB είναι η `imreconstruct`:

out=imreconstruct(marker, mask)

3.6.4.1 Άνοιγμα με ανακατασκευή(opening by reconstruction)

Στο μορφολογικό άνοιγμα(opening), η διάβρωση(erosion) τυπικά αφαιρεί μικρά αντικείμενα και η ακολουθούμενη διαστολή(dilation) τείνει να επαναφέρει το σχήμα των αντικειμένων που μένουν. Η ακρίβεια της επαναφοράς εξαρτάται από την ομοιότητα των σχημάτων και του δομικού στοιχείου. Η μέθοδος *άνοιγμα με ανακατασκευή(opening by reconstruction)* επαναφέρει ακριβώς το σχήμα των αντικειμένων που απομένουν μετά τη διάβρωση. Το *άνοιγμα με ανακατασκευή με χρήση του δομικού στοιχείου s* ορίζεται όπως φαίνεται παρακάτω: [9]

$$R_f(f \ominus s)$$

Παραδειγμα: Άνοιγμα με ανακατασκευή στο Matlab

```
>>f=imread('text.tif');
>>se=ones(40,1);
>>fe=imerode(f, se);
>>fo=imopen(f, se);
>>fobr=imreconstruct(fe,f);
```

Στην εικόνα 3.15 φαίνονται τα εξής:

3.15(β) η διάβρωση της αρχικής εικόνας κειμένου με κάθετη γραμμή ως δομικό στοιχείο (fe)

3.15(γ) η ανοιγμένη με κάθετη γραμμή ως δομικό στοιχείο (fo)

3.15(δ) η ανοιγμένη με ανακατασκευή με κάθετη γραμμή ως δομικό στοιχείο(fobr)

3.6.4.2 Γέμισμα οπών

Η μορφολογική ανακατασκευή έχει ένα ευρύ φάσμα πρακτικών εφαρμογών, η κάθε εφαρμογή καθορίζεται από την επιλογή των εικόνων δείκτη και μάσκας. Για παράδειγμα αν επιλέξουμε την εικόνα δείκτη f_m , να είναι 0 παντού εκτός από τα σύνορα της εικόνας όπου θα είναι $1-f$: [9]

$$f_m(x, y) = \begin{cases} 1-f(x, y) & \text{αν το (x,y) είναι στα σύνορα της f} \\ 0 & \text{διαφορετικά} \end{cases}$$

Τότε η $g=[R_{f^c}(f_m)]$ επιδρά στην f γεμίζοντας τις οπές της, όπως φαίνεται στην εικόνα 3.15(ε).

Η συνάρτηση **imfill** πραγματοποιεί την παραπάνω ενέργεια αυτόματα όταν χρησιμοποιηθεί η προαιρετική παράμετρος 'holes':

$$\mathbf{g=imfill(f,'holes')}$$

3.6.4.3 Αφαίρεση αντικειμένων των συνόρων(border objects)

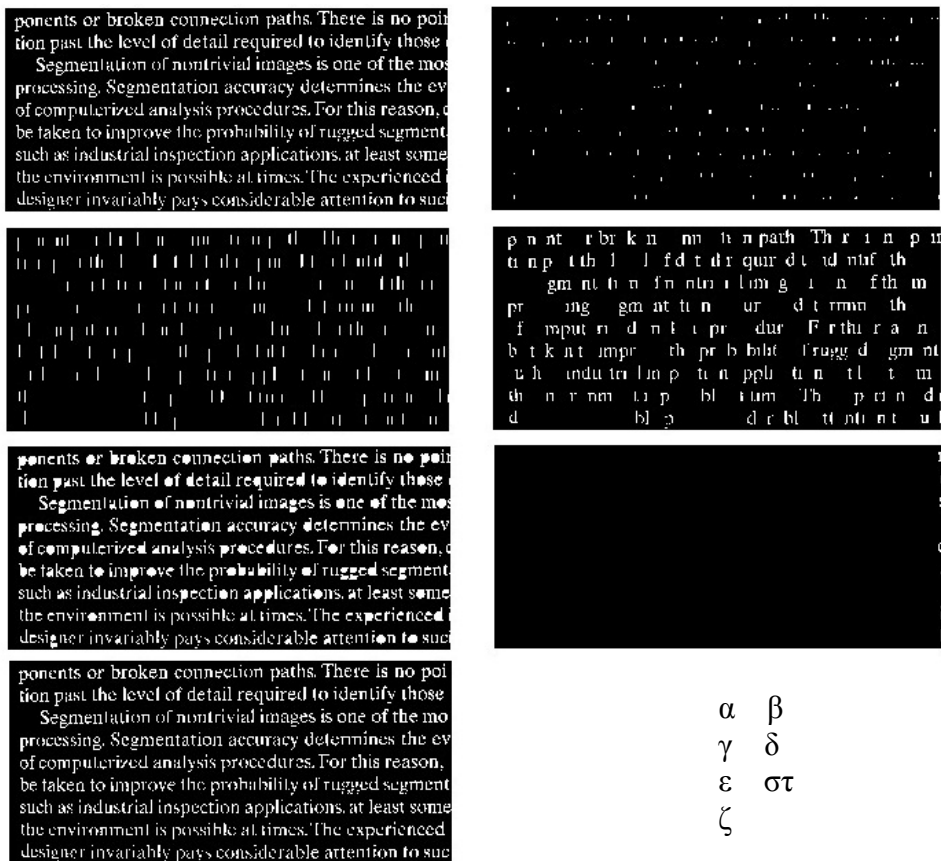
Ακόμη μια χρήσιμη εφαρμογή της ανακατασκευής εικόνας είναι η αφαίρεση των αντικειμένων που ακουμπούν στα σύνορα της εικόνας. Και πάλι η βασική ενέργεια είναι να επιλέξουμε το σωστό δείκτη και τη σωστή μάσκα για να επιτύχουμε το επιθυμητό αποτέλεσμα. Στην περίπτωση αυτή χρησιμοποιούμε την αρχική εικόνα σαν μάσκα και την f_m η οποία ορίζεται παρακάτω, σαν εικόνα-δείκτη: [9]

$$f_m(x, y) = \begin{cases} f(x, y) & \text{αν το (x,y) είναι στα σύνορα της f} \\ 0 & \text{διαφορετικά} \end{cases}$$

Η εικόνα 3.15(στ) δείχνει πως η ανακατασκευή, $R_f(f_m)$, περιέχει μόνο τα αντικείμενα που ακουμπούν στα σύνορα. Η διαφορά $f - R_f(f_m)$, που φαίνεται στην εικόνα 3.15(ζ), περιέχει μόνο τα αντικείμενα της αρχικής εικόνας που δεν ακουμπούν στα σύνορα. Η συνάρτηση **imclearborder** υλοποιεί την όλη διαδικασία αυτόματα:

$$\mathbf{g=imclearborder(f,conn)}$$

όπου f είναι η εικόνα εισόδου και g το αποτέλεσμα. Η τιμή του $conn$ είναι 4 ή 8(προεπιλεγμένο).



Εικόνα 3.15 Μορφολογική ανακατασκευή (α)αρχική εικόνα (β)διαβρωμένη με κάθετη γραμμή (γ) ανοιγμένη με κάθετη γραμμή (δ)ανοιγμένη με ανακατασκευή (ε)γέμισμα οπών (στ) χαρακτήρες που ακουμπούν στα σύνορα (ζ) χωρίς χαρακτήρες συνόρων

ΚΕΦΑΛΑΙΟ 4

ΚΑΤΑΤΜΗΣΗ ΕΙΚΟΝΑΣ

Εισαγωγή

Κατάτμηση ονομάζεται η διαδικασία κατά την οποία μια εικόνα χωρίζεται στα αντικείμενα ή μέρη από τα οποία αποτελείται και το φόντο της. Είναι το σημαντικό και δυσκολότερο κομμάτι στην επεξεργασία εικόνας και είναι καθοριστικό για την επιτυχία ή μη της τελικής ανάλυσης. Για το λόγο αυτό, ο αριθμός των μεθόδων κατάτμησης εικόνας που έχουν αναπτυχθεί είναι σχεδόν ο ίδιος με τον αριθμό των προβλημάτων κατάτμησης που υπάρχουν. Η αυτόματη κατάτμηση πυρήνων κυττάρων σε δισδιάστατες ή τρισδιάστατες εικόνες επιτρέπει τη μελέτη των ανεξάρτητων πυρήνων κυττάρων στο φυσικό τους χώρο (δηλαδή μέσα στον ιστό). Η αυτόματη κατάτμηση εικόνων είναι προτιμότερη από τη 'χειροκίνητη' χάραξη των ορίων κάθε πυρήνα με τη χρήση ποντικιού, αφού είναι πιο αντικειμενική. Ακόμη ένα πλεονέκτημα του αυτοματισμού είναι ότι αυξάνεται η ποσότητα των δεδομένων στα οποία μπορεί να εφαρμοστεί η κάθε μέθοδος, αφού ο ίδιος αλγόριθμος μπορεί να εφαρμοστεί σε όλες τις εικόνες, ανεξάρτητα από τα ιδιαίτερα χαρακτηριστικά τους. [8]

Στην αυτόματη κατάτμηση εικόνων κυττάρων συχνά εμφανίζονται προβλήματα, και οφείλονται κυρίως στα εξής:

α) Ανομοιομορφία έντασης φωτεινότητας του φόντου, εξαιτίας του πάχους του δείγματος και της ύπαρξης αντικειμένων σε επίπεδα κάτω από το επιφανειακό. Η ανομοιομορφία του φόντου καθιστά διαχωρισμό του φόντου από τα αντικείμενα πιο δύσκολη σε σχέση με το απόλυτα ομοιόμορφο φόντο.

β) Οι μεταβολές στη φωτεινότητα στο εσωτερικό των πυρήνων μπορεί να έχουν σαν αποτέλεσμα το διαχωρισμό ενός πυρήνα σε περισσότερα του ενός τμήματα κατά την κατάτμηση, αυτό έχει ως αποτέλεσμα την υπερ-κατάτμηση(over-segmentation).

γ) Πολλές φορές οι πυρήνες των κυττάρων είναι ενωμένοι(συσσωμάτωμα) και δυσκολεύει το διαχωρισμό τους σε ανεξάρτητους πυρήνες(συχνά ο διαχωρισμός τους είναι αδύνατος).

Στη συνέχεια αναλύονται μερικές από τις πιο βασικές μεθόδους κατάτμησης εικόνας.

4.1 Κατωφλίωση

Η κατωφλίωση είναι ένα χρήσιμο και απλό εργαλείο στην κατάτμηση γκρίζων εικόνων και βασίζεται στο γεγονός ότι τα αντικείμενα μίας εικόνας μπορούν να διακριθούν από το φόντο βάσει του χρώματός τους (επίπεδο του γκρίζου).

Χρησιμοποιώντας το ιστόγραμμα της εικόνας, γίνεται κατάλληλη επιλογή κάποιου ή κάποιων επιπέδων του γκρίζου, που θα χρησιμοποιηθεί ως κατώφλι ανάμεσα στις τιμές των pixels των αντικειμένων και του φόντου.

Η αρχική, γκρίζα, εικόνα μπορεί να μετατραπεί σε δυαδική, έτσι ώστε τα τμήματα της εικόνας που αντιστοιχούν σε αντικείμενα να εμφανίζονται στη δυαδική εικόνα άσπρα, ενώ τα υπόλοιπα τμήματα (φόντο) να εμφανίζονται στη δυαδική εικόνα

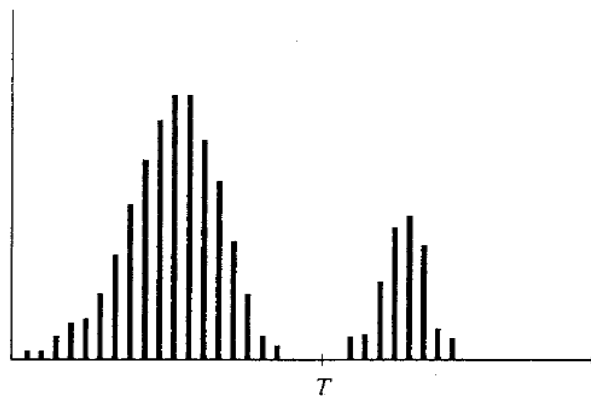
μαύρα (ή το αντίθετο). Μπορεί η μέθοδος αυτή να φαίνεται απλή, είναι όμως πολύ βασική στην επεξεργασία εικόνας.

4.1.1 Απλή γενική κατωφλίωση(Basic Global Thresholding)

Έστω η ιδανική περίπτωση, όπου στο ιστόγραμμα μίας γκριζας εικόνας $f(x,y)$ υπάρχει μία βαθιά και απότομη 'κοιλιάδα' ανάμεσα στις δύο κορυφές που αντιστοιχούν στα αντικείμενα (η μία) και το φόντο (η άλλη), και το επιθυμητό κατώφλι T επιλέγεται να είναι η μικρότερη τιμή της κοιλιάδας αυτής. Τότε κάθε σημείο (x,y) για το οποίο ισχύει $f(x,y) \geq T$ θα είναι σημείο αντικειμένου αλλιώς θα είναι σημείο του φόντου. Με άλλα λόγια, η κατωφλιωμένη εικόνα $g(x,y)$ ορίζεται ως εξής:

$$g(x,y) = \begin{cases} 1, & f(x,y) \geq T \\ 0, & f(x,y) < T \end{cases} \quad (4.1)$$

Ένας τρόπος επιλογής του κατωφλίου είναι με οπτική παρατήρηση του ιστογράμματος της εικόνας. Το ιστόγραμμα της εικόνας 4.1 έχει δύο ξεχωριστές περιοχές, έτσι είναι εύκολο να επιλεχθεί το κατώφλι T στη περιοχή που χωρίζει τις δύο αυτές περιοχές. [5], [11]



Εικόνα 4.1 Επιλογή κατωφλίου με οπτική παρατήρηση

Άλλη μέθοδος επιλογής κατωφλίου είναι η δοκιμαστική(trial and error), επιλέγουμε διαφορετικά κατώφλια μέχρι να πάρουμε το επιθυμητό αποτέλεσμα.

4.1.1.1 Αυτόματη επιλογή κατωφλίου των Gonzales και Woods

Για την αυτόματη επιλογή κατωφλίου οι Gonzales και Woods το 2002 περιέγραψαν την παρακάτω επαναληπτική διαδικασία: [8]

- (i) Επιλέγεται μια αρχική εκτίμηση για το κατώφλι T_0 (μια προτεινόμενη αρχική εκτίμηση είναι το μέσο μεταξύ της ελάχιστης και μέγιστης τιμής της έντασης της εικόνας)
- (ii) Τμηματοποιείται η εικόνα χρησιμοποιώντας το T . Έτσι θα προκύψουν δυο ομάδες pixel: η G_1 , αποτελούμενη από όλα τα pixel με τιμές έντασης $\geq T$, και η G_2 αποτελούμενη από τα pixel με τιμές $< T$
- (iii) Υπολογίζεται ο μέσος όρος των τιμών έντασης μ_1 και μ_2 για τα pixel κάθε περιοχής G_1 και G_2 .
- (iv) Υπολογίζεται ένα νέο κατώφλι: $T = \frac{1}{2}(\mu_1 + \mu_2)$
- (v) Επαναλαμβάνονται τα βήματα (ii) μέχρι (iv) μέχρι η διαφορά στο T μεταξύ δύο ακόλουθων τιμών να είναι μικρότερη από μια προεπιλεγμένη τιμή T_0 .

4.1.1.2 Αυτόματη επιλογή κατωφλίου - Μη παραμετρική μέθοδος κατωφλίωσης OTSU

Για την αυτόματη επιλογή κατωφλίου ο Nobuyuki Otsu το 1979 [13] πρότεινε μια μη παραμετρική μέθοδο κατωφλίωσης, η οποία είναι βασισμένη στο ιστογράμμα της εικόνας, η μέθοδος αυτή αναλύεται παρακάτω.

Έστω ότι τα pixels της εικόνας διακρίνονται σε L διακριτά επίπεδα γκριζου τόνου

$[1, 2, \dots, L]$, n_i το πλήθος των pixels που έχουν επίπεδο γκριζου τόνου i και $N=n_1+n_2+\dots+n_L$ το συνολικό πλήθος των pixels της εικόνας.

Για λόγους απλότητας το ιστογράμμα κανονικοποιείται και θεωρείται σαν μια κατανομή πιθανότητας:

$$p_i = \frac{n_i}{N} \text{ και } p_i \geq 0, \sum_{i=0}^{L-1} p_i = 1 \quad (4.2)$$

Υποθέτουμε ότι με μια τιμή κατωφλίου T δημιουργούνται δύο κλάσεις C_0 και C_1 (φόντο και αντικείμενα, αντίστοιχα). Η C_0 αναφέρεται στα pixels με επίπεδα $[1,2,\dots,T]$, και η C_1 αναφέρεται στα pixels με επίπεδα $[T+1,T+2,\dots,L]$. Συνεπώς οι αντίστοιχες πιθανότητες για τις κλάσεις και οι μέσες τιμές θα είναι:

$$\omega_0 = \Pr(C_0) = \sum_{i=0}^T p_i = \omega(T) \quad (4.3)$$

$$\omega_1 = \Pr(C_1) = \sum_{i=T+1}^L p_i = 1 - \omega(T) \quad (4.4)$$

$$\mu_0 = \sum_{i=0}^T i \Pr(i | C_0) = \sum_{i=0}^T i p_i / \omega_0 = \mu(T) / \omega(T) \quad (4.5)$$

$$\mu_1 = \sum_{i=T+1}^L i \Pr(i | C_1) = \sum_{i=T+1}^L i p_i / \omega_1 = \frac{\mu_\tau - \mu(T)}{1 - \omega(T)} \quad (4.6)$$

όπου

$$\omega(T) = \sum_{i=1}^T p_i \quad (4.7) \text{ και } \mu(T) = \sum_{i=1}^T i p_i \quad (4.8)$$

είναι οι μηδενικές και πρώτης τάξης, αντίστοιχα, αθροιστικές ροπές του ιστογράμματος μέχρι το επίπεδο T , και

$$\mu_\tau = \mu(L) = \sum_{i=1}^L i p_i \quad (4.9)$$

είναι η ολική μέση τιμή για την εικόνα. Για οποιοδήποτε T εύκολα βλέπουμε πως ισχύει:

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_\tau \text{ και } \omega_0 + \omega_1 = 1 \quad (4.10)$$

Οι διακυμάνσεις των κλάσεων είναι

$$\sigma_0^2 = \sum_{i=1}^T (i - \mu_0)^2 \Pr(i | C_0) = \sum_{i=1}^T (i - \mu_0)^2 p_i / \omega_0 \quad (4.11)$$

$$\sigma_1^2 = \sum_{i=T+1}^L (i - \mu_1)^2 \Pr(i | C_1) = \sum_{i=T+1}^L (i - \mu_1)^2 p_i / \omega_1 \quad (4.12)$$

Για την αξιολόγηση τα απόδοσης του διαχωρισμού με την επιλογή της τιμής κατωφλίου εισάγουμε τα εξής όρους

$$\lambda = \sigma_B^2 / \sigma_w^2, \quad \kappa = \sigma_T^2 / \sigma_w^2 \quad \text{και} \quad n = \sigma_B^2 / \sigma_T^2 \quad (4.13)$$

όπου

$$\sigma_w^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 \quad (4.14)$$

$$\sigma_B^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 = \omega_0 (\mu_0 - \mu_\tau)^2 + \omega_1 (\mu_1 - \mu_\tau)^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \quad (4.15)$$

(λόγω της 4.9)

και

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i \quad (4.16)$$

είναι η διακύμανση μεταξύ των κλάσεων, η διακύμανση σε κάθε κλάση και η ολική διακύμανση, αντίστοιχα. Επομένως το πρόβλημα μπορεί να αντιμετωπιστεί σαν ένα πρόβλημα βελτιστοποίησης για την επιλογή της τιμής T που μεγιστοποιεί κάποιον από τα όρους τα σχέσης (4.12). Επίσης επειδή πάντα ισχύει η παρακάτω σχέση:

$$\sigma_w^2 + \sigma_B^2 = \sigma_T^2 \quad (4.17)$$

οι τρεις όροι της (4.12) είναι ισοδύναμοι [$\kappa=1+\lambda$, $n=\lambda/(\lambda+1)$], άρα αρκεί να βρεθεί το T που μεγιστοποιεί οποιονδήποτε από τα λ , κ , n . Επειδή ο υπολογισμός του n είναι ευκολότερος, επιλέγουμε να λύσουμε αυτό. Είναι (συναρτήσει του Tα)

$$n(T) = \frac{\sigma_B^2(T)}{\sigma_T^2} \quad (4.18)$$

$$\Rightarrow \sigma_B^2(T) = \frac{[\mu_T \omega(T) - \mu(T)]^2}{\omega(T)[1 - \omega(T)]} \quad (4.19)$$

Τελικά η βέλτιστη τιμή κατωφλίου, T_{opt} , που μεγιστοποιεί το n , είναι το T που μεγιστοποιεί την ποσότητα

$$\sigma_B^2(T_{opt}) = \max_{1 \leq T \leq L} \sigma_B^2(T) \quad (4.20)$$

Παράδειγμα 4.1: Υπολογισμός καθολικού κατωφλίου στο Matlab

Στο παράδειγμα αυτό παρουσιάζεται η επαναληπτική μέθοδος εύρεσης κατωφλίου των Gonzales-woods καθώς και η μη παραμετρική μέθοδος του Otsu. Η εικόνα 4.2α απεικονίζει μια γκρι εικόνα κυττάρων από τεστ παπ.

α) Η επαναληπτική μέθοδος υλοποιείται ως εξής στο MATLAB

```
>>To=0.5
>>T=0.5*(double(min(f(:)))+double(max(f(:)))) ;
>>done=false;
>> while ~done
    g=f>=T;
    Tnext=0.5*(mean(f(g))+mean(f(~g)));
    done=abs(T-Tnext)<To;
    T=Tnext;
```

End

Το αποτέλεσμα είναι T= 159.8898

β) Μη παραμετρική μέθοδος ευρεσης κατωφλίου Otsu

Η μέθοδος ευρεσης κατωφλίου Otsu υλοποιείται αυτόματα στη κλιμακα 0-1 με τη συνάρτηση του MATLAB **graythresh**

```
>>T2=graythresh(f)
```

```
T2= 0.6275
```

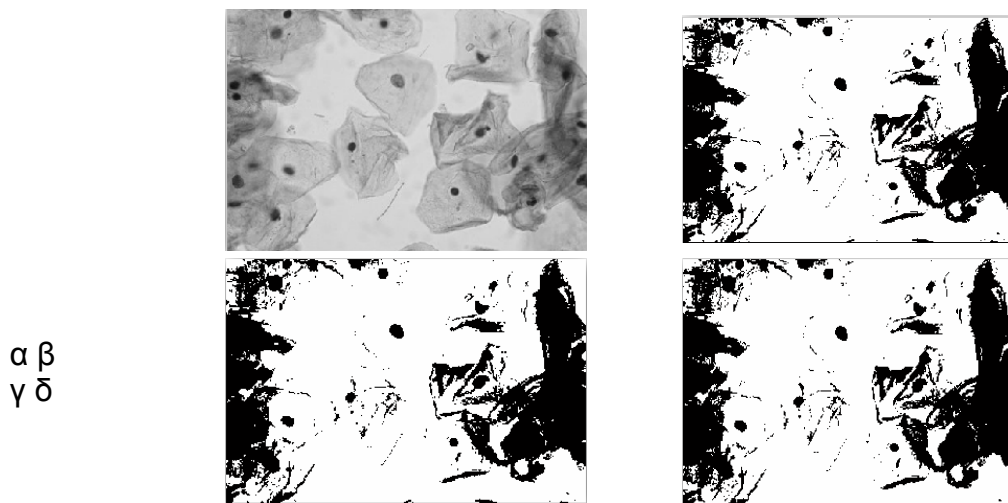
```
>>T2*255
```

```
Ans= 160
```

Μια δεύτερη συνάρτηση που υλοποιήσαμε με βάση τη θεωρία της μεθόδου otsu είναι η **otsu_thresh** και φαίνεται παρακάτω:

```
% euresh tou katwfliou thw eikonas apo to istogramma ths me thn texnikh
% του
% Otsu klimaka T: 0-255
function [Totsu]=otsu_thresh(f);
[h,g]=imhist(f);
x1=length(h);
N=(ones(1,x1))*h;
a=zeros(1,x1);
n=h;
p=n/N;
gp=g.*p;
mT=(ones(1,x1))*gp;
t=1;
while t<x1+1
    for k=1:t
        a(k)=1;
    end
    w(t)=a*p;
    m(t)=a*gp;
    if (w(t)>0) & (w(t)<1)
        sb(t)=( (mT*w(t)-m(t))^2) / (w(t)*(1-w(t)));
    else sb(t)=0;
    end
    t=t+1;
end
opt_sb=max(sb);
Totsu=find(sb==opt_sb);
if length(Totsu)>1
    Totsu=max(Totsu);
end
```

και δίνει T3=161



Εικόνα 4.2 (α)αρχική εικόνα (β)κατωφλίωση με Gonzales,
(γ)graythresh, (δ)otsu_thresh

Η κατωφλίωση υλοποιήθηκε με τη συνάρτηση **im2bw(f,T)** του MATLAB με το T στη περιοχή 0-1

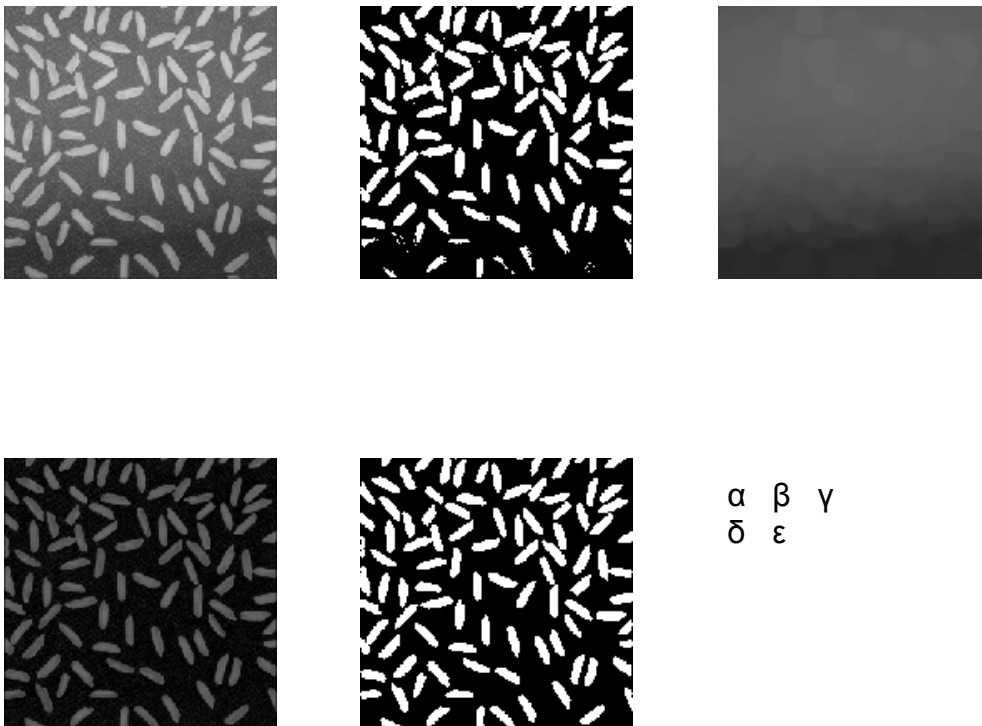
Μια παραλλαγή της παραπάνω διαδικασίας με χρήση απλού κατωφλίου, αποτελεί η χρήση δύο κατωφλίων T1 και T2 που ορίζουν ένα εύρος τιμών κατά τη διαδικασία κατωφλίωσης, η οποία εκφράζεται από τη σχέση:

$$g(x,y) = \begin{cases} 0, & f(x,y) < T_1 \\ 1, & T_1 \leq f(x,y) \leq T_2, \text{ διπλή κατωφλίωση} \\ 0, & f(x,y) > T_2 \end{cases} \quad (4.21)$$

Η επιτυχία της κατωφλίωσης εξαρτάται σημαντικά από την επιλογή της τιμής κατωφλίου.

4.1.2 Προσαρμοστική ή Τοπική κατωφλίωση(Adaptive-Local Thresholding)

Η μέθοδος της καθολικής κατωφλίωσης μπορεί να αποτύχει όταν η φωτεινότητα της εικόνας είναι ανομοιόμορφη(εικόνα 4.3 α,β). Τότε είναι κοινή πρακτική στις περιπτώσεις αυτές να προεπεξεργαστούμε την εικόνα για να αντισταθμίσουμε τα προβλήματα της φωτεινότητας και μετά να εφαρμόσουμε καθολική κατωφλίωση στην προεπεξεργασμένη εικόνα. [9]



Εικόνα 4.3 Μετ/σμος top-hat (α) αρχική εικόνα (β) Κατωφλιωμένη εικόνα (γ) Εικόνα μετά από open (δ) εικόνα με μετ/σμο top-hat (ε) κατωφλιωμένη top-hat εικόνα

Στην εικόνα 4.3 εφαρμόζουμε μετασχηματισμό top-hat και κατόπιν το αποτέλεσμα της κατωφλίωσης είναι σαφώς καλύτερο (εικ. 4.3 ε).

Αποδεικνύεται πως η ανωτέρω διαδικασία είναι ισοδύναμη με την κατωφλίωση της εικόνας $f(x,y)$ με μια τοπικά μεταβαλλόμενη συνάρτηση $T(x,y)$:

$$g(x,y) = \begin{cases} 1, & f(x,y) \geq T(x,y) \\ 0, & f(x,y) < T(x,y) \end{cases} \quad (4.22)$$

Όπου

$$T(x,y) = f_0(x,y) + T_0 \quad (4.23)$$

Η εικόνα $f_0(x,y)$ είναι το μορφολογικό άνοιγμα (opening) της f , και η σταθερά T_0 είναι το αποτέλεσμα της συνάρτησης $\text{graythresh}(\text{global thresholding})$ στην f_0 .

Γενικά η προσαρμοζόμενη κατωφλίωση (adaptive thresholding), χρησιμοποιεί περισσότερες της μιας τιμές κατωφλίων. Εφαρμόζεται όταν το επίπεδο της πυκνότητας του υποβάθρου δεν είναι σταθερό και το αντικείμενο παρουσιάζει μεταβολές μέσα στην εικόνα. Εξετάζει τις σχέσεις γειτονικών στοιχείων εικόνας και προσαρμόζει το κατώφλι σύμφωνα με τη στατιστική πυκνοτήτων σε διάφορες περιοχές.

4.2 Κατάτμηση με χρήση τεχνικών ταξινόμησης

Με τη διαδικασία ταξινόμησης επιτυγχάνεται η μείωση της ποσότητας των δεδομένων, ομαδοποιώντας δεδομένα με κοινά χαρακτηριστικά. Τέτοιες ομαδοποιήσεις διευκολύνουν τον ανθρώπινο εγκέφαλο να επεξεργαστεί την παρεχόμενη πληροφορία με τρόπο πιο αποτελεσματικό και ακριβή. Η ομαδοποίηση στο χώρο των χαρακτηριστικών των δεδομένων πραγματοποιείται αυτόματα με χρήση αλγορίθμων ταξινόμησης (classification). [5]

Οι αλγόριθμοι ταξινόμησης μπορούν να διακριθούν σε δυο βασικούς τύπους: τους *ιεραρχικούς* και τους *διαμεριστικούς*. Οι ιεραρχικοί αλγόριθμοι αρχικοποιούνται με τυχαίο προσδιορισμό των κατηγοριών (κλάσεων) και αναπτύσσονται διασπώντας μεγάλες μη ομογενείς κατηγορίες ή ενώνοντας μικρές κατηγορίες με μεγάλη ομοιότητα. Οι διαμεριστικοί αλγόριθμοι επιχειρούν να διαιρέσουν άμεσα τα δεδομένα σε ένα σύνολο διακριτών κατηγοριών, ελαχιστοποιώντας ένα μέτρο ανομοιότητας μεταξύ δεδομένων που ανήκουν στην ίδια κατηγορία και μεγιστοποιώντας το μέτρο αυτό για δεδομένα που ανήκουν σε διαφορετικές κατηγορίες. [5]

Θεωρούμε το πρόβλημα της ταξινόμησης ενός συνόλου διανυσμάτων:

$x_{ij} = (x_1, x_2, \dots, x_M)$, $i=1,2, \dots, N$ σε K κατηγορίες. Το κέντρο κάθε κατηγορίας, δηλ. ο αριθμητικός μέσος των διανυσμάτων που έχουν αποδοθεί σε αυτή, συμβολίζεται με m_k , $k=1,2, \dots, K$. Στην ανάλυση που ακολουθεί θεωρούμε ότι το διάνυσμα x_i

αποτελείται

από τη χρωματική τιμή ενός στοιχείου όγκου καθώς και τις χρωματικές τιμές των γειτονικών του στοιχείων, που περιλαμβάνονται σε μια περιοχή διαστάσεων $3 \times 3 \times 3$. Ανάλογα με τη φύση των δεδομένων μπορεί να απαιτηθεί η χρήση πιο εξεζητημένων διανυσμάτων χαρακτηριστικών που περιλαμβάνουν για παράδειγμα πληροφορία υφής (texture).

4.2.1 Ο αλγόριθμος των K-μέσων(K-means)

Ο αλγόριθμος των K-Μέσων, K-means, KM) εξελίσσεται ελαχιστοποιώντας την ακόλουθη ποσότητα:

$$E = \sum_{i=1}^N (\vec{x}_i - \vec{m}_k)^2 \quad (4.24)$$

Όπου \vec{m}_k είναι το κέντρο της κατηγορίας που είναι πιο κοντά στο διάνυσμα δεδομένων \vec{x}_i .

Η υλοποίηση του αλγορίθμου με τη μορφή ψευδοκώδικα παρουσιάζεται παρακάτω: [5]

t=0

Αρχικοποίηση των κέντρων των κατηγοριών, $k=1,2,\dots,K$ με τυχαία κέντρα $\vec{m}_k(0)$

Επανάληψη

{

Για τα κέντρα $\vec{m}_k(t)$ όλων των κατηγοριών

{

- Εντοπισμός των διανυσμάτων δεδομένων των οποίων η Ευκλείδεια απόσταση από το $\vec{m}_k(t)$ είναι ελάχιστη

- Το $\vec{m}_k(t+1)$ παίρνει τιμή ίση με το νέο κέντρο βάρους του \vec{x}_k ,

$$\vec{m}_k(t+1) = \frac{\vec{x}_k}{n_k}, \quad n_k: \text{το πλήθος διανυσμάτων στην } k\text{-κατηγορία}$$

}

t=t+1

}

Μέχρι $\left| \vec{m}_k(t-1) - \vec{m}_k(t) \right| \leq error$, για όλες τις κατηγορίες k.

Ο αλγόριθμος σε απλά βήματα είναι:

1. Επιλογή του αριθμού των ομάδων.
2. Τυχαία δημιουργία k ομάδων και ορισμός των κεντροειδών των ομάδων.
3. Μεταβίβαση του κάθε σημείου στο κεντροειδές της κοντινότερης ομάδας.
4. Υπολογισμός των νέων κεντροειδών των ομάδων.
5. Επανάληψη μέχρι να συγκλίνει ο αλγόριθμος σε κάποιο κριτήριο(αναφέρεται παρασπάνω στο ψευδοκώδικα).

Ο αλγόριθμος ξεκινά διαχωρίζοντας τα αρχικά σημεία σε k αρχικά σύνολα είτε τυχαία είτε χρησιμοποιώντας ευριστικά δεδομένα. Στη συνέχεια υπολογίζει το μεσαίο ή το κεντροειδές του κάθε συνόλου, υλοποιεί νέο διαχωρισμό ώστε το κάθε σημείο

να σχετίζεται με το κοντινότερο κεντροειδές. Έπειτα τα κεντροειδή ξαναυπολογίζονται για τις νέες ομάδες, ο αλγόριθμος επαναλαμβάνει τα δυο βήματα ωστόσο τα σημεία δεν μπορούν να αλλάξουν ομάδες (ή εναλλακτικά τα κεντροειδή παραμένουν αμετάβλητα).

Ο αλγόριθμος αυτός παραμένει διάσημος επειδή τείνει σε κάποιο όριο πολύ γρήγορα. Όσον αφορά την απόδοση ο αλγόριθμος δεν εγγυάται ότι θα αγγίξει το βέλτιστο. Η ποιότητα της τελική λύσης εξαρτάται πολύ από το αρχικό σύνολο ομάδων και μπορεί να είναι πολύ χαμηλότερη από το συνολικό βέλτιστο. Επίσης ένα άλλο μειονέκτημα του αλγόριθμου είναι ότι ο αριθμός των ομάδων πρέπει να οριστεί εξ αρχής.

Παράδειγμα 4.2: Εφαρμογή αλγορίθμου K-Μέσων(K-means) στο Matlab: [19]

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%COLOR BASED SEGMENTATION USING K-MEANS CLUSTERING%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;

%STEP 1: Διάβασμα εικόνας
he=imread('hestain.png');
subplot(1,2,1);imshow(he);title('Εικόνα H & E')

%STEP 2: Μετατροπή εικόνας RGB-->L*a*b*
cform=makecform('srgb2lab');
lab_he=applycform(he,cform);

%STEP 3: Ταξινόμηση χρωμάτων στο 'a*b*' χρωματικό μοντέλο χρησιμοποιώντας
%ταξινομητή K-μέσων
ab=double(lab_he(:,2:3));
nrows=size(ab,1);
ncols=size(ab,2);
ab=reshape(ab,nrows*ncols,2);
nColors=3;

%Επανάληψη ταξινόμησης 3 φορές για αποφυγή repeat τοπικών ελαχίστων
[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);

%STEP 4: Βάζουμε ετικέτες σε όλα τα pixels με τις πληροφορίες από την
%συνάρτηση kmeans
pixel_labels=reshape(cluster_idx,nrows,ncols);
subplot(1,2,2)
imshow(pixel_labels,[]),title('Εικόνα με ετικέτα από το cluster index');
```

%STEP 5: Εικόνες κατάτμησης της H&E με βάση το χρώμα.

```
segmented_images = cell(1,3);
rgb_label = repmat(pixel_labels,[1 1 3]);

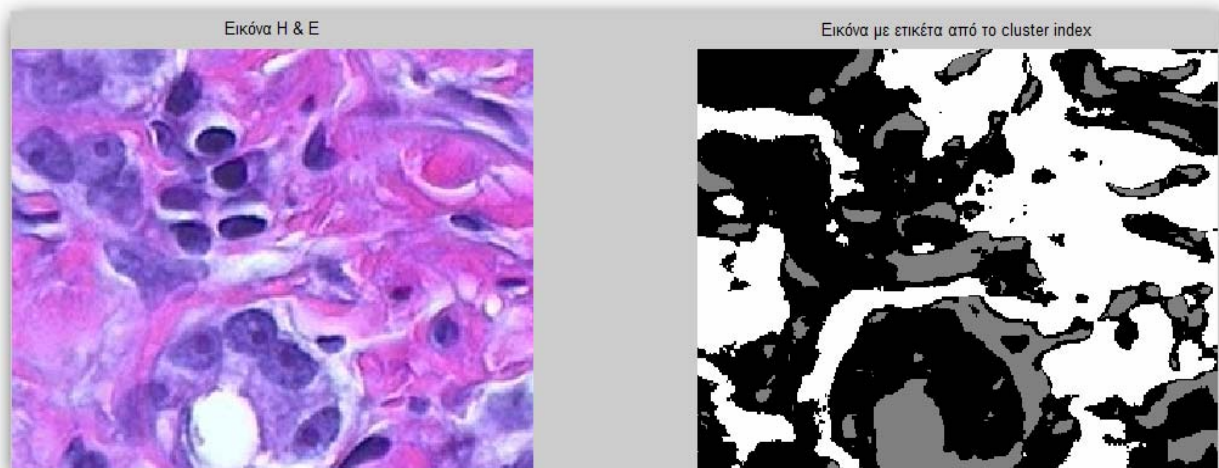
for k = 1:nColors
    color = he;
    color(rgb_label ~= k) = 0;
    segmented_images{k} = color;
end
figure;
subplot(2,2,1)
imshow(segmented_images{1}), title('Αντικείμενα στο cluster 1');
subplot(2,2,2)
imshow(segmented_images{2}), title('Αντικείμενα στο cluster 2');
subplot(2,2,3)
imshow(segmented_images{3}), title('Αντικείμενα στο cluster 3');
```

%STEP 6: Ξεχωρίζουμε τον πυρήνα σε διαφορετική εικόνα

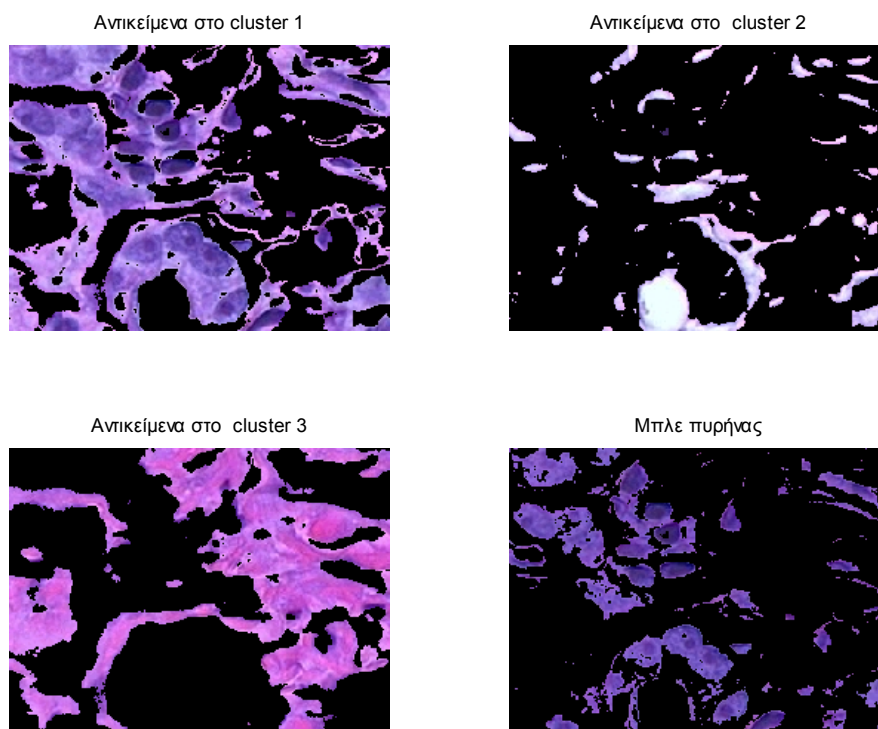
```
mean_cluster_val = zeros(3,1);
for k = 1:nColors
    mean_cluster_val(k) = mean(cluster_center(k));
end
[mean_cluster_val,idx] = sort(mean_cluster_val);
blue_cluster_num = idx(2);

L = lab_he(:,:,1);
blue_idx = find(pixel_labels == blue_cluster_num);
L_blue = L(blue_idx);
is_light_blue = im2bw(L_blue,graythresh(L_blue));

nuclei_labels = repmat(uint8(0),[nrows ncols]);
nuclei_labels(blue_idx(is_light_blue==false)) = 1;
nuclei_labels = repmat(nuclei_labels,[1 1 3]);
blue_nuclei = he;
blue_nuclei(nuclei_labels ~= 1) = 0;
subplot(2,2,4)
imshow(blue_nuclei), title('Μπλε πυρήνας');
```

ΑΠΟΤΕΛΕΣΜΑΤΑ:

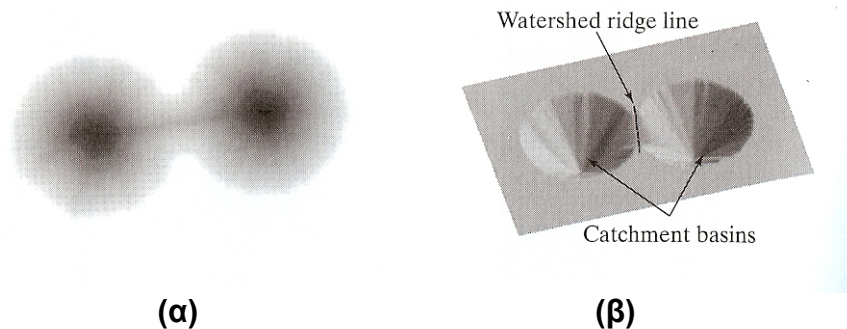
Εικόνα 4.4 Αρχική εικόνα κ εικόνα με ετικέτα απ'το cluster index



Εικόνα 4.5 Αποτέλεσμα clustering παραδείγματος color based k-means

4.3 Κατάτμηση με χρήση μετασχηματισμού Watershed

Γεωγραφικά, το watershed είναι η κορυφογραμμή που χωρίζει δύο περιοχές αποστράγγισης διαφορετικών ποταμών. Η κοιλάδα απορροής (catchment basin) είναι η περιοχή στην οποία συγκεντρώνεται το νερό. Ο μετασχηματισμός watershed εφαρμόζει τις παραπάνω ιδέες στην επεξεργασία γκριζων εικόνων με σκοπό να μπορέσει να λύσει μια ποικιλία προβλημάτων κατάτμησης. [9]



ΕΙΚΟΝΑ 4.6 (α) Γκριζα εικόνα μαύρων σταγόνων
(β) Η ίδια εικόνα μετασχηματισμένη σαν μια επιφάνεια

Ο αλγόριθμος watershed λοιπόν βασίζεται στην αναλογία των εναλλαγών του επιπέδου φωτεινότητας μίας εικόνας με τις εναλλαγές του υψόμετρου σε ένα τοπίο. Αν η εικόνα αντιστοιχιστεί με ένα τοπίο (Εικ. 4.6) και η ένταση της φωτεινότητας κάθε pixel αντιστοιχιστεί με υψόμετρο, στις περιοχές της εικόνας που έχουν μεγάλη φωτεινότητα αντιστοιχεί περιοχή του τοπίου που βρίσκεται σε μεγάλο υψόμετρο. Αν στο τοπίο αρχίσει να πέφτει νερό, ο πλημμυρισμός θα αρχίσει από τις περιοχές με χαμηλό υψόμετρο (κοιλάδες) και θα ανέβει προς τις περιοχές μεγάλου υψόμετρου (κορυφές). Όταν το νερό που έχει ξεκινήσει από διαφορετικές κοιλάδες φτάσει σε κάποια κορυφή που τις χωρίζει, δημιουργείται το όριο ανάμεσα στις δύο αυτές κοιλάδες. Σύμφωνα με την αντιστοίχιση που προτάθηκε παραπάνω, αν κανείς «πλημμυρήσει» την εικόνα, ξεκινώντας από τα σημεία χαμηλότερης έντασης (δηλαδή μικρότερου υψόμετρου), το νερό θα γεμίσει τις περιοχές της εικόνας, σταματώντας στα όρια τοπικού μεγίστου (που είναι οι ακμές των σχημάτων της εικόνας). Όταν το νερό που πλημμυρίζει δύο γειτονικές περιοχές φτάσει στο ίδιο μέγιστο, τότε «χτίζεται» το όριο των δύο περιοχών και με τον τρόπο αυτό πραγματοποιείται η κατάτμηση της εικόνας.

Πρέπει λοιπόν στις εικόνες κυττάρων οι περιοχές των πυρήνων να είναι πιο σκοτεινές από το φόντο της εικόνας, για μπορεί να χρησιμοποιηθεί ο αλγόριθμος watershed για την κατάτμηση τους, αν δεν συμβαίνει αυτό θα πρέπει να γίνει προετοιμασία της εικόνας σε αυτή τη μορφή πριν εφαρμοστεί ο αλγόριθμος watershed.

4.3.1 Κατάτμηση με χρήση μετασχηματισμού Watershed χρησιμοποιώντας το μετασχηματισμό απόστασης

Ο μετασχηματισμός της απόστασης μιας δυαδικής εικόνας είναι η απόσταση κάθε Pixel από το κοντινότερο μη μηδενικό pixel. Έτσι λοιπόν αν ο μετασχηματισμός της απόστασης εφαρμοστεί σε μια εικόνα με μαύρα αντικείμενα(πχ πυρήνες κυττάρων) και άσπρο φόντο προκύπτει μια γκριζα εικόνα στο οποίο κάθε pixel θα δείχνει την απόσταση του αντίστοιχου σημείου από το κοντινότερο σημείο του φόντου(μη μηδενικό pixel). Δηλαδή όσο πιο μακριά είναι ένα pixel της δυαδικής εικόνας από το φόντο τόσο πιο φωτεινό είναι το αντίστοιχο pixel της εικόνας του μετασχηματισμού απόστασης, δηλαδή τόσο μεγαλύτερη είναι η ένταση της φωτεινότητάς του.

Στην κατάτμηση εικόνων, αξιοποιείται το γεγονός ότι κάθε pixel στην εικόνα του μετασχηματισμού απόστασης είναι ενδεικτικό της απόστασης του αντίστοιχου pixel της αρχικής εικόνας από το φόντο. Αν λοιπόν υπολογιστεί η συμπληρωματική εικόνα της εικόνας του μετασχηματισμού απόστασης, σε μια εικόνα άσπρου φόντου, θα προκύψει μία εικόνα που θα έχει πολύ μικρή ένταση φωτεινότητας στις περιοχές των κυττάρων που είναι πιο μακριά από το φόντο (οι περιοχές αυτές θα είναι πολύ σκούρες), ενώ όσο τα pixel πλησιάζουν στο φόντο θα έχουν μεγαλύτερη ένταση της φωτεινότητας και οι περιοχές που αντιστοιχούν στο φόντο θα έχουν τη μέγιστη ένταση φωτεινότητας (θα είναι χρώματος λευκού). Αν λοιπόν εφαρμοστεί ο αλγόριθμος watershed στην ανάστροφη της εικόνας του μετασχηματισμού απόστασης, ο 'πλημμυρισμός' θα ξεκινήσει από τις περιοχές ελαχίστου (δηλαδή τις περιοχές που βρίσκονται πιο μακριά από το φόντο) και θα 'χτίσει' τα όρια στις περιοχές που βρίσκονται τα όρια των κυττάρων. [9]

Παράδειγμα 4.3: Κατάτμηση watershed με χρήση μετασχηματισμού απόστασης

Εισάγεται στο Matlab την εικόνα στην οποία θα γίνει η κατάτμηση και μετατρέπεται σε δυαδική με τις συναρτήσεις **im2bw** και **graythresh**:

```
fg=imread('image');
fbw=im2bw(fg,graythresh(fg));
```

Η εικόνα 4.7(α) δείχνει το αποτέλεσμα. Το επόμενο βήμα είναι να αναστραφεί η εικόνα και να υπολογιστεί ο μετασχηματισμός της απόστασης(αφού όπως αναφέρθηκε πρέπει να είναι άσπρο το φόντο κατά την εφαρμογή του μετασχηματισμού της απόστασης), με τη συνάρτηση **bwdist**:

```
fbwc=~fbw;
D=bwdist(fbwc);
```

Οι εικόνες 4.7(β), 4.7(γ) δείχνουν την συμπληρωματική εικόνα της (α) και τον μετασχηματισμό απόστασης σε αυτήν αντίστοιχα.

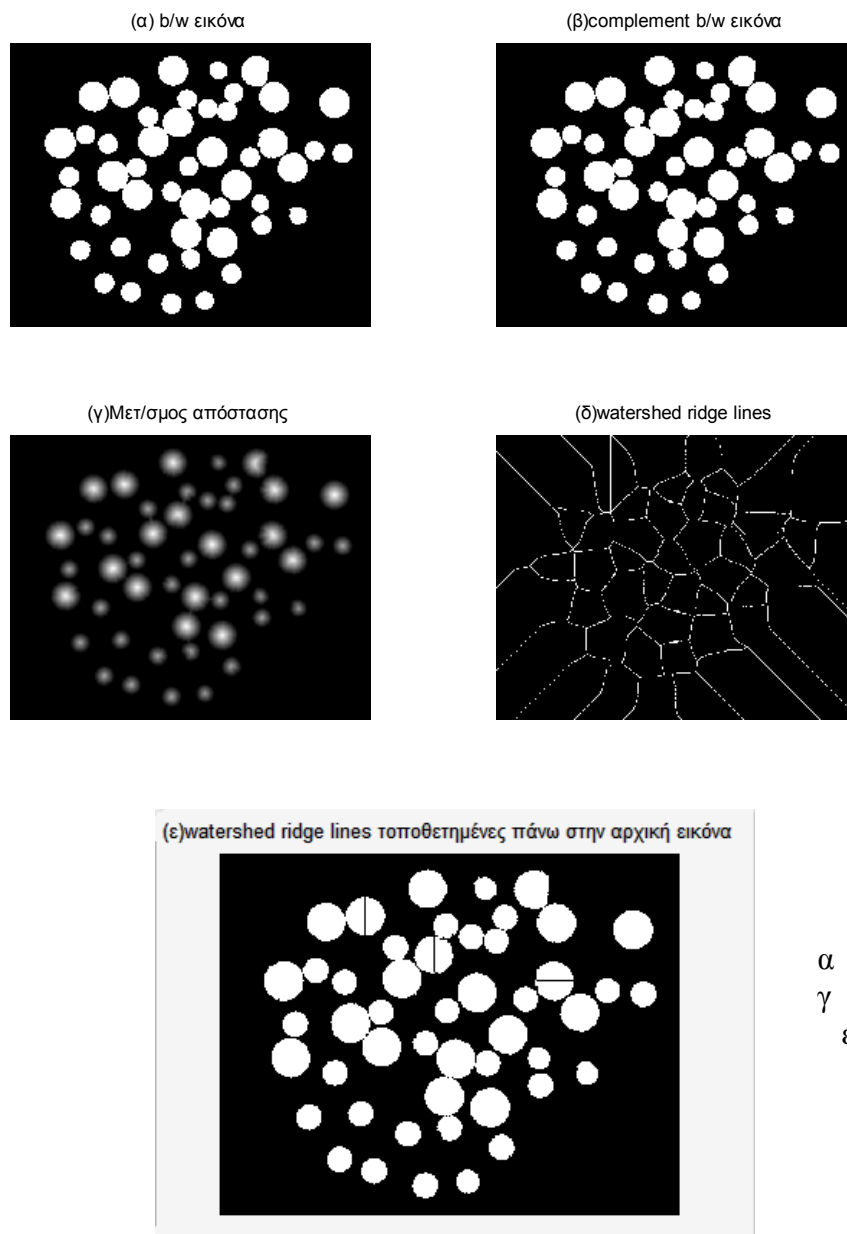
Μετά υπολογίζεται ο μετασχηματισμός watershed της αρνητικής εικόνας του μετασχηματισμού απόστασης, με τη συνάρτηση **watershed**:

```
L=watershed(-D);
w=L==0;
```

Τα μηδενικά pixel στο πίνακα L δείχνουν τα pixel κορυφογραμμής(ridge) ενώ τα pixel με τιμή 1 δείχνουν τις κοιλάδες(catchment basins). Το w δείχνει μόνο τα ridge pixel(Εικ. 4.7(δ)). Τελικά εφαρμόζεται η λογική πράξη AND της αρχικής εικόνας με το συμπλήρωμα του w ώστε να φανεί το αποτέλεσμα της κατάτμησης(Εικ 4.7(ε)):

$$f_{out}=g \& \sim w$$

Στην τελική εικόνα (Εικ.4.7(ε)) φαίνεται πως μερικά αντικείμενα διαχωρίστηκαν λάθος. Αυτό καλείται **υπερκατάτμηση(oversegmentation)**, και αποτελεί ένα σύνηθες πρόβλημα στις μεθόδους κατάτμησης με watershed.



ΕΙΚΟΝΑ 4.7 Κατάτμηση watershed με μετασχηματισμό απόστασης.

4.3.2 Κατάτμηση με χρήση μετασχηματισμού Watershed χρησιμοποιώντας Gradients

Συνήθως το πλάτος του gradient (βαθμωτή κλίση) χρησιμοποιείται σαν προεπεξεργασία μιας γκριζας εικόνας πριν εφαρμοστεί ο μετασχηματισμός watershed για τη κατάτμηση. Το πλάτος του gradient έχει υψηλές τιμές pixel (κοντά στο 1-λευκό) γύρω από τις ακμές των αντικειμένων, όπου υπάρχει απότομη εναλλαγή φωτεινότητας και πολύ χαμηλές (κοντά στο 0-μαύρο) οπουδήποτε αλλού. Αν στην εικόνα της gradient εφαρμοστεί ο αλγόριθμος watershed, το 'νερό' θα ξεκινήσει από τα χαμηλότερα τμήματα της εικόνας, δηλαδή τα τμήματα όπου η μεταβολή της φωτεινότητας στην αρχική εικόνα είναι μικρή, και θα φτάσει στα υψηλότερα, δηλαδή στα τμήματα όπου η μεταβολή της φωτεινότητας είναι έντονη (δηλαδή στις ακμές των αντικειμένων). Το επόμενο παράδειγμα αναδεικνύει αυτή την ιδέα. [9]

Παράδειγμα 4.4: Κατάτμηση γκριζας εικόνας με τη χρήση watershed και gradients

Η εικόνα 4.8(α) δείχνει μια εικόνα f , που περιέχει διάφορες μαύρες σταγόνες. Αρχικά υπολογίζεται το gradient της εικόνας χρησιμοποιώντας μεθόδους γραμμικού φιλτραρίσματος:

```
f=imread('image');
h=fspecial('sobel');
fd=double(f);
x=imfilter(fd,h,,'replicate');
y=imfilter(fd,h',,'replicate');
grad=sqrt(x.^2+y.^2);
```

⇔ [x,y]=gradient(fd);

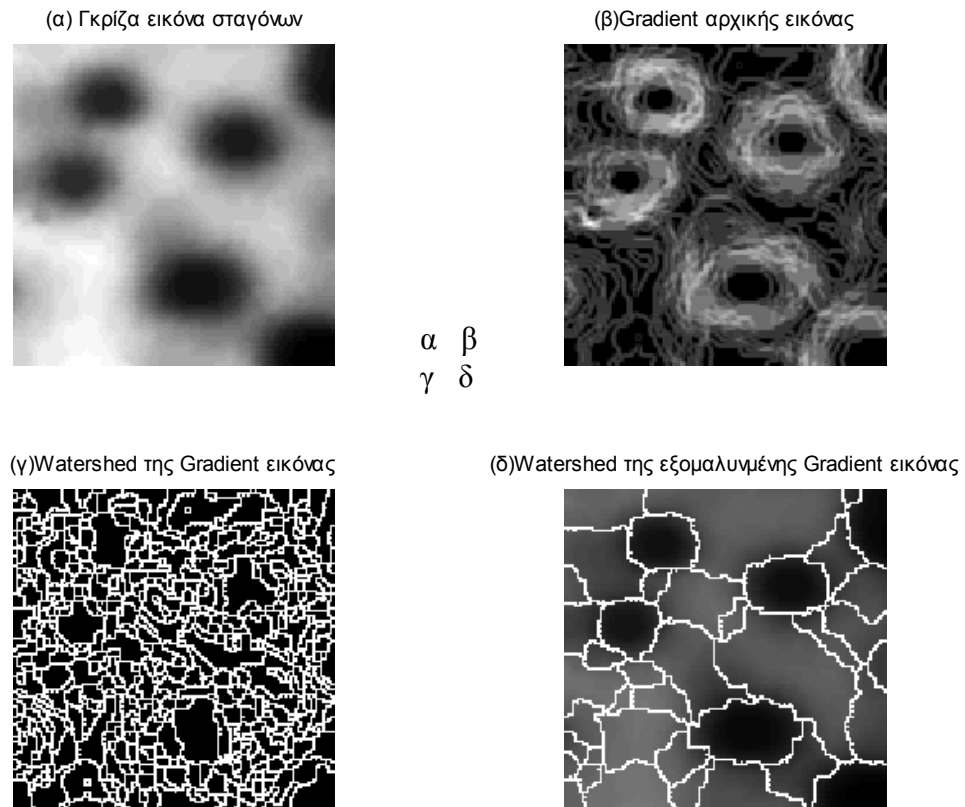
Η εικόνα 4.8(β) δείχνει την εικόνα του πλάτους του gradient, grad. Στη συνέχεια υπολογίζεται ο μετασχηματισμός watershed της gradient εικόνας και βρίσκονται οι watershed ridge lines:

```
L=watershed(grad);
wr=L==0;
```

Η εικόνα 4.8(γ) δείχνει πως η κατάτμηση δεν είναι καθόλου καλή, υπάρχουν πολλές ridge lines που δεν ανταποκρίνονται στα αντικείμενα που μας ενδιαφέρουν, δηλαδή έχουμε και εδώ υπερκατάτμηση. Μια λύση αυτού του προβλήματος είναι να γίνει η gradient εικόνα πιο ομαλή (smooth) και μετά να υπολογιστεί ο μετασχηματισμός watershed. Χρησιμοποιήθηκαν λοιπόν μορφολογικοί τελεστές opening-closing:

```
grad2=imclose(imopen(grad,ones(3,3)),ones(3,3));
L2=watershed(grad2);
wr2=L2==0;
f2=f;
f2(wr2)=255;
```

Οι δυο τελευταίες γραμμές του κώδικα τοποθετούν τις watershed ridge lines πάνω στην αρχική εικόνα (Εικ. 4.8(δ)). Παρόλο που επιτεύχθηκε βελτίωση σε σχέση με την εικόνα 4.8(γ) κάποιες ridge lines είναι άσχετες με τα αντικείμενα ενδιαφέροντος



ΕΙΚΟΝΑ 4.8 Κατάτμηση watershed με gradient.

4.3.3 Κατάτμηση με χρήση Marker-controlled Watershed

Η απευθείας εφαρμογή του μετασχηματισμού watershed στο gradient μιας εικόνας συνήθως οδηγεί σε υπερκατάτμηση (oversegmentation) λόγω του θορύβου και άλλων ανωμαλιών που εισάγει το gradient εξαιτίας της εναλλαγής της έντασης στα αντικείμενα (πχ πυρήνες κυττάρων) και του φόντου. Μια πρακτική λύση στο πρόβλημα είναι να περιοριστεί ο αριθμός των επιτρεπόμενων περιοχών με ένα ενδιάμεσο στάδιο προεπεξεργασίας σχεδιασμένο να δώσει περισσότερη πληροφορία στη διαδικασία της κατάτμησης.

Μια λύση στο πρόβλημα της υπερκατάτμησης είναι να αγνοηθούν κάποια ελάχιστα, ώστε ο «πλημμυρισμός» να ξεκινάει από συγκεκριμένα σημεία μόνο κι όχι από όλα. Αυτό επιτυγχάνεται με τον marker-controlled watershed μετασχηματισμό. Το marker (σημάδι) είναι ένα συνδεδεμένο αντικείμενο που ανήκει στην εικόνα. Χρειαζόμαστε εσωτερικά σημάδια μέσα στα αντικείμενα, όπου από εκεί θα ξεκινάει το 'νερό' του μετασχηματισμού watershed καθώς και εξωτερικά σημάδια όπου θα σταματάει. Προσοχή χρειάζεται στο να έχει κάθε αντικείμενο ενδιαφέροντος ένα εσωτερικό σημάδι, εάν έχει περισσότερα θα γίνει υπερκατάτμηση και αν δεν έχει καθόλου υποκατάτμηση. Επίσης τα εξωτερικά σημάδια πρέπει να βρίσκονται μεταξύ των σημαδεμένων, με εσωτερικούς markers, αντικειμένων. Για τον υπολογισμό των σημαδιών (markers) υπάρχουν διάφορες μέθοδοι όπως: γραμμικό φιλτράρισμα, μη γραμμικό φιλτράρισμα και μορφολογική επεξεργασία. [9]

Παράδειγμα 4.5: Κατάτμηση με Marker-controlled Watershed

Το παράδειγμα αυτό εφαρμόζει τον Marker-controlled watershed αλγόριθμο στην εικόνα 4.9(α). Αρχικά υπολογίζεται το gradient της 4.9(α) όπως στο παράδειγμα 4.4:

```
f=imread('image');
fd=double(f);

[x,y]=gradient(fd);
grad=sqrt(x.^2+y.^2);

L=watershed(grad);
wr=L==0;
```

Στην εικόνα 4.9(β) φαίνεται η υπερκατάτμηση που προκύπτει λόγω του μεγάλου αριθμού τοπικών ελαχίστων. Με τη συνάρτηση **imregionalmin(f)** υπολογίζεται η θέση των τοπικών ελαχίστων:

```
rm=imregionalmin(grad);
```

Η *rm* είναι μια δυαδική εικόνα που περιέχει τα τοπικά ελάχιστα της gradient εικόνας, στην εικόνα 4.9(γ) φαίνεται το γιατί ο μετασχηματισμός watershed έφτιαξε τόσες πολλές μικρές κοιλάδες(catchment basins).

Τα περισσότερα τοπικά ελάχιστα που φαίνονται στην εικόνα 4.9(γ) είναι πολύ 'ρηχά' και περιέχουν περιττή πληροφορία για τη κατάτμηση της εικόνας. Για να εξαλείψουμε αυτά τα περισευάμενα τοπικά ελάχιστα χρησιμοποιούμε τη συνάρτηση **imextendedmin(f,h)** που υπολογίζει τα 'χαμηλά σημεία' της εικόνας που βρίσκονται πιο βαθιά (από ένα κατώφλι *h*) σε σχέση με τα γύρω τους: [9]

```
h=2;
im=imextendedmin(f,h);
fim=f;
fim(im)=175;
```

Οι δύο τελευταίες γραμμές του κώδικα τοποθετούν τις περιοχές extended minima ('βαθιές περιοχές') με γκρι χρώμα μέσα στην αρχική εικόνα.

Στη συνέχεια πρέπει να βρεθούν οι εξωτερικοί markers, ή pixels που ανήκουν σίγουρα στο φόντο. Η προσέγγιση εδώ είναι να μαρκαριστεί το φόντο βρίσκοντας pixel τα οποία βρίσκονται μεταξύ των εσωτερικών markers. Το πρόβλημα αυτό λύνεται μέσω λύσης ενός άλλου προβλήματος watershed: συγκεκριμένα υπολογίζεται ο μετασχηματισμός watershed του μετασχηματισμού της απόστασης της εικόνας των εσωτερικών markers(*im*): [9]

```
Lim=watershed(bwdist(im));
em=Lim==0;
```

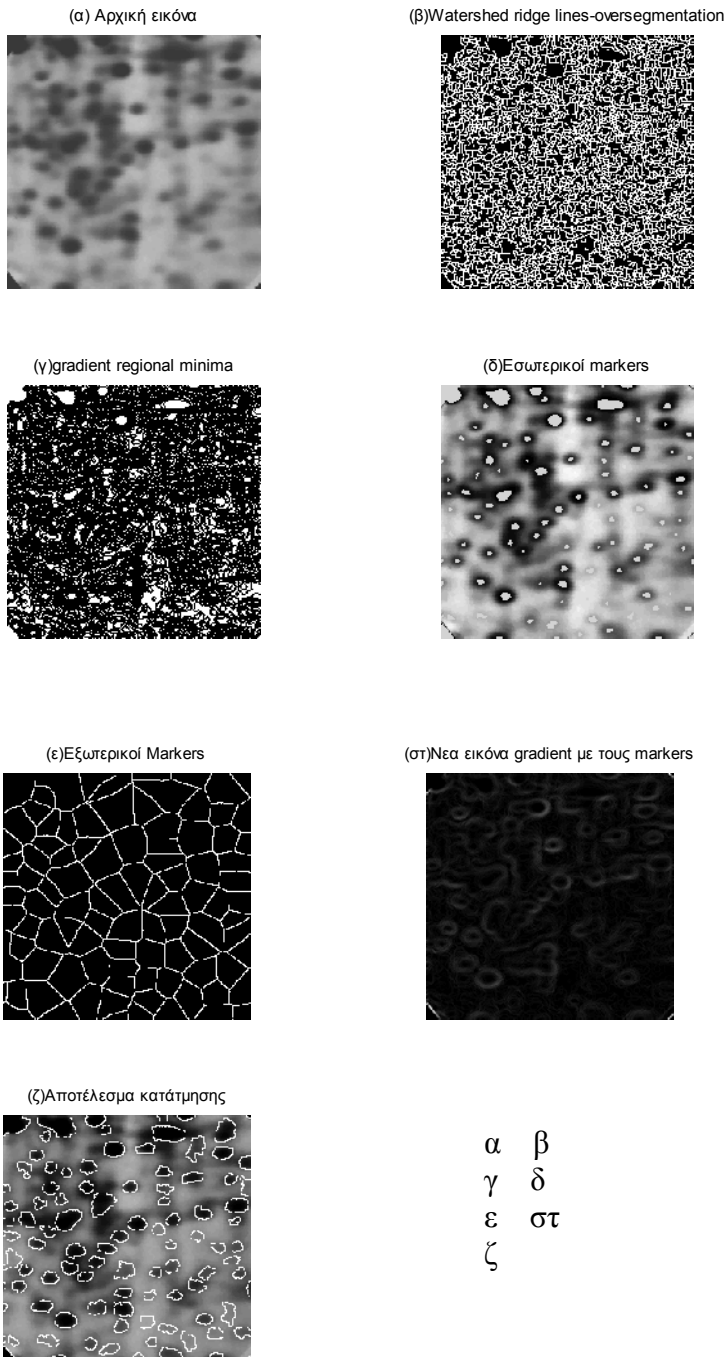
Η εικόνα 4.9(ε) δείχνει τις ridge lines στην δυαδική εικόνα *em*. Αφού οι ridge lines είναι μεταξύ των αντικειμένων που είναι σημαδεμένα με τους εσωτερικούς markers δε θα υπάρξει πρόβλημα.

Οι εσωτερικοί και οι εξωτερικοί markers χρησιμοποιούνται για να αλλάξουμε την εικόνα gradient με τη μέθοδο minima imposition η οποία τροποποιεί μια γκρι εικόνα ώστε τα τοπικά ελάχιστα να βρίσκονται μόνο στις σημαδεμένες περιοχές. Όλα τα υπόλοιπα pixel 'ανεβάζουν' τη τιμή τους για να εξαφανιστούν τα τοπικά ελάχιστα, αυτό επιτυγχάνεται με την εντολή **imimposemin(f,mask)**:

```
grad2=imimposemin(grad, im|em);
```

Η εικόνα 4.9(στ) δείχνει το αποτέλεσμα. Τέλος υπολογίζεται ο μετασχηματισμός watershed της τροποποιημένης εικόνας gradient. Το τελικό αποτέλεσμα φαίνεται στην εικόνα 4.9(ζ)

```
L2=watershed(grad2);
f2=f;
f2(L2==0)=255;
```



ΕΙΚΟΝΑ 4.9 Κατάτμηση marker-controlled watershed (με gradient)

ΚΕΦΑΛΑΙΟ 5

ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕ ΧΡΗΣΗ MATLAB¹

Εισαγωγή

Στο κεφάλαιο αυτό θα αναπτυχθούν αλγόριθμοι οποίοι ανιχνεύουν κυτταροπαθολογικά κύτταρα διαφόρων φωτογραφιών μικροσκοπίου(κύτταρα από ιστό μαστού, κύτταρα μελανωμάτων, λεμφοκυττάρων, εγκεφαλικά κύτταρα και κύτταρα από πάγκρεας). [18]

Σε κάθε είδος κυτταρικής εικόνας εφαρμόζεται και διαφορετική μέθοδος κατάτμησης. Στο τέλος του κεφαλαίου έχει επιλεγθεί η κυτταρική εικόνα του παγκρέατος για τη μέτρηση των χαρακτηριστικών των πυρήνων.

5.1 Προεπεξεργασία εικόνας

Στις περισσότερες εικόνες κυττάρων αν όχι σε όλες το φόντο δεν είναι ομοιόμορφο και πολλές φορές υπάρχουν διάφορα μαύρα σημάδια (ακαθαρσίες που μεγεθύνθηκαν στο μικροσκόπιο).

Η προ-επεξεργασία της εικόνας έχει σκοπό να μειώσει τις ανεπιθύμητες ατέλειες που έχουν εισαχθεί στην εικόνα από το σύστημα απεικόνισης. Κάποιοι μέθοδοι που χρησιμοποιούνται στην προ-επεξεργασία εικόνας είναι η μείωση των ανομοιομορφιών της εικόνας, η εξομάλυνση για την απαλοιφή του θορύβου και όξυνση των ακμών της εικόνας.

Ένας τρόπος να επιτευχθεί αυτό το φιλτράρισμα είναι η εφαρμογή στην εικόνα εναλλασσόμενων openings και closings, τα οποία βασίζονται στην ανακατασκευή (reconstruction). Τα φίλτρα ανακατασκευής ανήκουν στην τάξη των συνδεδεμένων τελεστών (connected operators), οι οποίοι έχουν την θεμελιώδη ιδιότητα της αλληλεπίδρασης με την εικόνα μέσω επίπεδων ζωνών. [4] Δεν απομακρύνουν συνιστώσες συχνότητας (όπως κάνουν τα γραμμικά φίλτρα) και δεν επηρεάζουν το σχήμα (όπως συμβαίνει με την εφαρμογή ενός απλού opening ή closing). Αυτό που ουσιαστικά κάνουν είναι να απομακρύνουν και να συγχωνεύουν επίπεδες ζώνες της εικόνας (δηλαδή ομοιόμορφες περιοχές).

Χωρίς την κατάλληλη προ-επεξεργασία της εικόνας τις περισσότερες φορές η κατάτμησή της είναι αδύνατη.

Παρακάτω αναφέρονται τέσσερις απλοί αλγόριθμοι προ-επεξεργασίας οι οποίοι εφαρμόζονται και στους αλγόριθμους κατάτμησης παρακάτω.

¹ Ο κώδικας MATLAB που χρησιμοποιήθηκε στο κεφάλαιο αυτό βρίσκεται στο παράρτημα που παρατίθεται στο τέλος της εργασίας.

1^{ος} Αλγόριθμος Προεπεξεργασίας(smooth1):

Η εικόνα μετατρέπεται από έγχρωμη(όλες οι κυτταρικές εικόνες είναι έγχρωμες λόγω του χρωματισμού τους από το κυτταρολόγο) σε γκρίζα, μετά με την εφαρμογή πολλαπλών opening και closing προκαλείται εξομάλυνση:

```
%ΟΜΑΛΟΠΟΙΗΣΗ ΚΑΙ ΑΦΑΙΡΕΣΗ ΘΟΡΥΒΟΥ ΜΕ RECONSTRUCTION
%ΣΕ GRAYSCALE ΕΙΚΟΝΑ
%=====
function [f] = smooth1(f)
f=rgb2gray(f);
for n=1:5
    se=strel('disk',n);
    marker=imerode(f,se);
    A=imreconstruct(marker,f);
    B=255-A;
    marker=imerode(B,se);
    f=255-imreconstruct(marker,B);
end;
```

2^{ος} Αλγόριθμος Προεπεξεργασίας(smooth2):

Χρησιμοποιήθηκε μόνο το 2ο φασματικό κανάλι της εικόνας (δηλαδή το πράσινο), συνεπώς δεν ήταν πια απαραίτητη η μετατροπή της σε γκρίζα, κι έπειτα ακολουθήθηκε η ίδια διαδικασία με τον πρώτο αλγόριθμο:

```
%ΕΞΟΜΑΛΥΝΣΗ ΤΗΣ ΕΙΚΟΝΑΣ ΚΑΙ ΑΠΑΛΟΙΦΗ ΘΟΡΥΒΟΥ ΜΕ RECONSTRUCTION
%ΣΕ ΕΓΧΡΩΜΗ ΕΙΚΟΝΑ ΣΤΟ 2ο ΧΡΩΜΑΤΙΚΟ ΚΑΝΑΛΙ

function [f] =smooth2(f)

f=f(:, :, 2);

for n=1:5
    se=strel('disk',n);
    marker=imerode(f,se);
    A=imreconstruct(marker,f);
    B=255-A;
    marker=imerode(B,se);
    f=255-imreconstruct(marker,B);
end;
```

3^{ος} Αλγόριθμος Προεπεξεργασίας(smooth3):

Αρχικά μετατράπηκε η εικόνα σε γκρίζα(grayscale) και στη συνέχεια με τη χρήση ενός φίλτρου ενδιάμεσης τιμής παραθύρου $W=[m \ n]$, η εικόνα έγινε πιο “ομαλή” , και η ένταση των pixels, ειδικά στο φόντο, έγινε ομοιόμορφη.

```
%ΕΞΟΜΑΛΥΝΣΗ ΤΗΣ ΕΙΚΟΝΑΣ ΚΑΙ ΑΠΑΛΟΙΦΗ ΘΟΡΥΒΟΥ ΜΕ ΦΙΛΤΡΟ ΕΝΔΙΑΜΕΣΗΣ
%ΤΙΜΗΣ
function [f] =smooth3(f,W)
f=rgb2gray(f);
f=medfilt2(f,W);
```

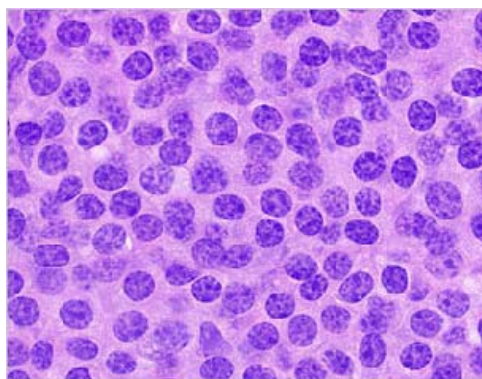
4^{ος} Αλγόριθμος Προεπεξεργασίας(smooth4):

Αρχικά η εικόνα μετατράπηκε σε γκριζα(gray scale) και στη συνέχεια χρησιμοποιήθηκε η συνάρτηση `imhmax(f,h)` η οποία υπολογίζει το μετασχηματισμό H-Maxima ο οποίος προσαρμόζει (συμπιέζει) όλες τις τιμές έντασης των pixel της εικόνας `f` που έχουν τιμή έντασης κάτω της τιμής `h`, είναι δηλαδή σαν φίλτρο που κρατάει τα μέγιστα της εικόνας που είναι μεγαλύτερα από το `h` (ή ισοδύναμα απορρίπτει τα ελάχιστα που είναι μικρότερα από το `h`). Μετά από διάφορες δοκιμές στις εικόνες που επεξεργαζόμαστε στην εργασία αυτή προκύπτει –χωρίς αυτό να είναι κανόνας-πως μια καλή τιμή για το `h` είναι το μισό του κατωφλίου που προκύπτει από τη μέθοδο Otsu(συνάρτηση `graythresh`):

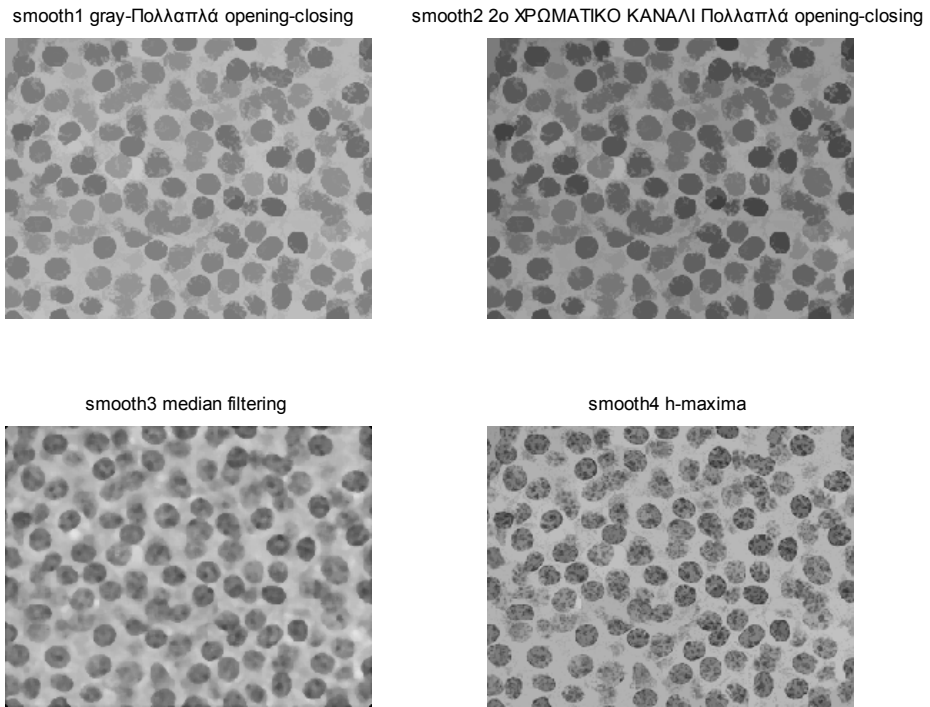
```
%ΕΞΟΜΑΛΥΝΣΗ ΤΗΣ ΕΙΚΟΝΑΣ ΚΑΙ ΑΠΑΛΟΙΦΗ ΘΟΡΥΒΟΥ ΜΕ ΤΗΝ
%ΣΥΝΑΡΤΗΣΗ imhmax και τη βοήθεια της μεθόδου Otsu
function [f] =smooth4(f)
    f=rgb2gray(f);
    T=255*graythresh(f);
    f=imhmax(f,T/2);
```

Παρακάτω δίνεται ένα παράδειγμα χρήσης του κάθε παραπάνω αλγορίθμου προεπεξεργασίας:

```
fc = imread(strcat(pname, fname));
subplot (221);
imshow(fc);title('Αρχική εικόνα');
figure;
fs1=smooth1(fc);
subplot(221);imshow(fs1);title('smooth1 gray-Πολλαπλά opening-closing ');
fs2=smooth2(fc);
subplot(222);imshow(fs2);title('smooth2 2ο ΧΡΩΜΑΤΙΚΟ ΚΑΝΑΛΙ Πολλαπλά
opening-closing');
fs3=smooth3(fc,[5 5]);
subplot(223);imshow(fs3);title('smooth3 median filtering');
fs4=smooth4(fc);
subplot(224);imshow(fs4);title('smooth4 h-maxima');
```



ΕΙΚΟΝΑ 5.1 Αρχική εικόνα πριν από το φιλτράρισμα



ΕΙΚΟΝΑ 5.2 Εικόνα μετά την προ-επεξεργασία

5.2 Αλγόριθμος Κατωφλίσωσης

Η πιο γνωστή και ακριβής μέθοδος αυτόματης επιλογής κατωφλίου σε γκριζες εικόνες είναι η μη παραμετρική μέθοδος Otsu την οποία είδαμε αναλυτικά στο κεφάλαιο 4. Τη μέθοδο αυτή υλοποιεί η συνάρτηση του MATLAB `graythresh(f)` η οποία μας επιστρέφει την τιμή του κατωφλίου στην περιοχή 0 ως 1. Μια ακόμη υλοποίηση του αλγορίθμου αυτού σε Matlab, σύμφωνα με τη θεωρία που αναπτύχθηκε στο προηγούμενο κεφάλαιο φαίνεται παρακάτω.

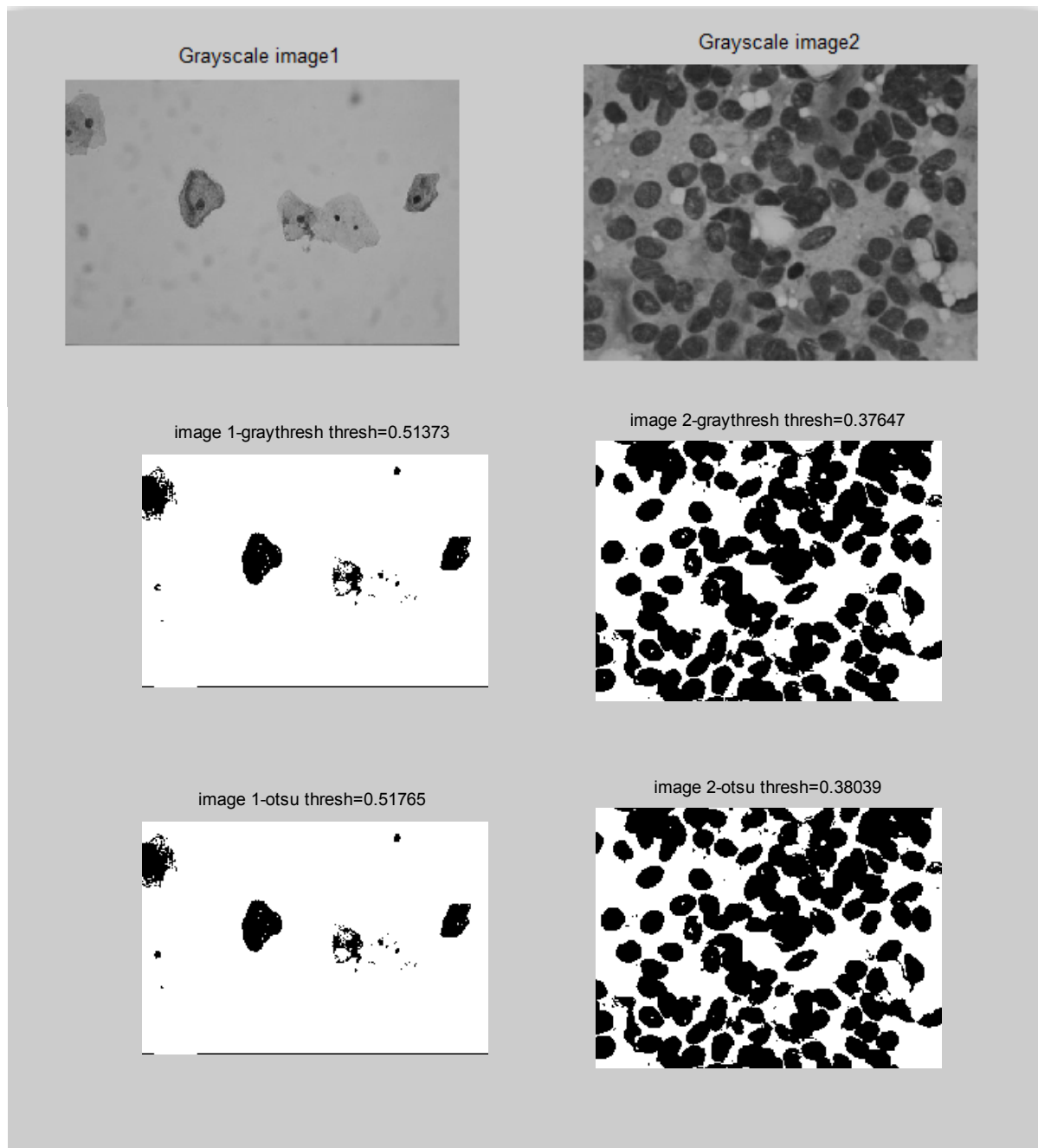
```
%ΕΥΡΕΣΗ ΤΟΥ ΚΑΤΩΦΛΙΟΥ ΤΗΣ ΕΙΚΟΝΑΣ ΑΠΟ ΤΟ ΙΣΤΟΓΡΑΜΜΑ ΜΕ ΤΗΝ
%ΤΕΧΝΙΚΗ ΤΟΥ Nobuyuki Otsu 1979
```

```
function [Thresh]=otsu(f);
[h,g]=imhist(f);

x1=length(h);
N=(ones(1,x1))*h;
a=zeros(1,x1);
n=h;
p=n/N;
gp=g.*p;
mT=(ones(1,x1))*gp;
t=1;
while t<x1+1
    for k=1:t
        a(k)=1;
    end
    w(t)=a*p;
    m(t)=a*gp;
    if (w(t)>0) & (w(t)<1)
        sb(t)=((mT*w(t)-m(t))^2)/(w(t)*(1-w(t)));
    else sb(t)=0;
    end
    t=t+1;
end

opt_sb=max(sb);
Thresh=find(sb==opt_sb);
if length(Thresh)>1
    Thresh=max(Thresh);
end
Thresh=Thresh/255
```

Παρακάτω στην εικόνα 5.3 βλέπουμε δύο διαφορετικές εικόνες στις οποίες εφαρμόσαμε κατωφλίωση με τη συνάρτηση ***im2bw(fg,Threshold)*** όπου το **threshold** είναι στη κλίμακα 0-1, εφαρμόσαμε τη συνάρτηση `graythresh` και τη συνάρτηση Otsu που αναφέραμε παραπάνω:

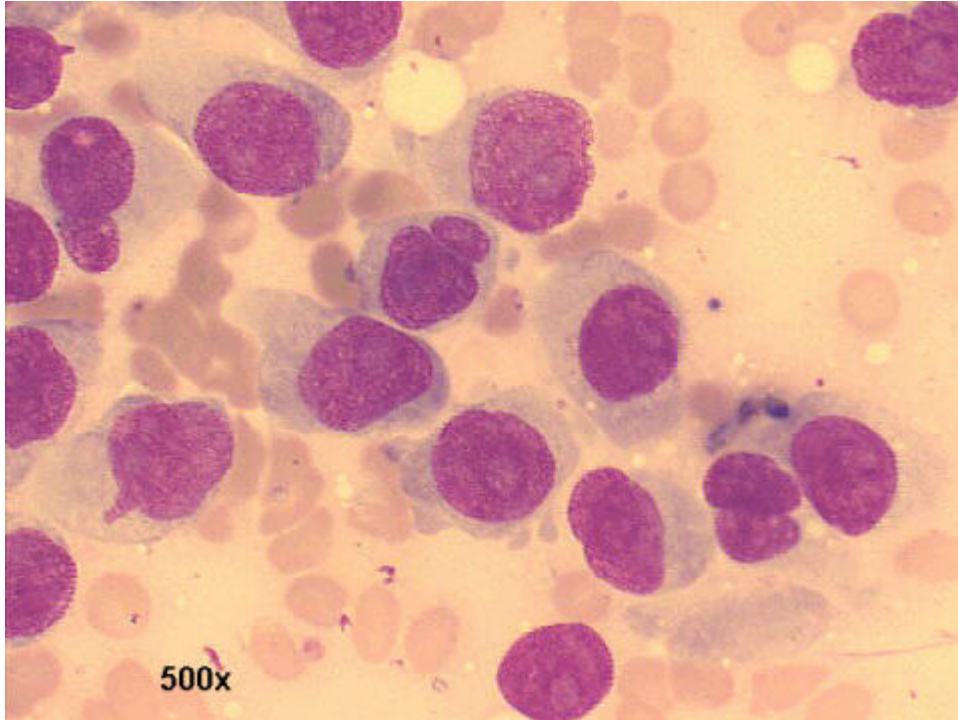


ΕΙΚΟΝΑ 5.3 Αρχικές εικόνες και οι εικόνες μετά την κατωφλίωση με την Otsu και τη graythresh του Matlab

Παρατηρούμε πως οι τιμές στο threshold που υπολογίζει η συνάρτηση Otsu που αναφέρθηκε πιο πάνω και η graythresh του Matlab έχουν πολύ μικρές διαφορές

5.3 Κατάτμηση βασισμένη στο χρώμα με χρήση ταξινομητή K-μέσων με εφαρμογή σε εικόνα καρκινικών κυττάρων στήθους

Η εικόνα που χρησιμοποιήθηκε για τον αλγόριθμο color k-means clustering (αναφέρθηκε στο κεφάλαιο 4) είναι μια εικόνα με κύτταρα από παρακέντηση στήθους η οποία έχει χρωματιστεί για καλύτερη ανίχνευση των κυττάρων.



ΕΙΚΟΝΑ 5.4 Πορογενές καρκίνωμα μαστού (x500)

Στόχος είναι να απομονώθούν τα κύτταρα (αντικείμενα) από το φόντο. Η μέθοδος color k-means clustering σε απλά βήματα έχει ως εξής: [19]

Βήμα1: Εισαγωγή εικόνας

Εισάγουμε την εικόνα σε πίνακα με χρήση του MATLAB
`f=imread('Breast ductal Carcinoma FNA.jpg');`

Βήμα2: Μετατροπή εικόνας από RGB σε L*a*b

Στη η εικόνα μετατρέπεται από το χρωματικό μοντέλο RGB στο χρωματικό μοντέλο L*a*b. Παρατηρώντας την εικόνα 5.4 αγνοώντας τις αλλαγές στη φωτεινότητα φαίνεται πως περιέχει 2 χρώματα ροζ και μωβ. Το χρωματικό μοντέλο L*a*b(γνωστό και ως CIELAB ή CIE L*a*b) βοηθά να ποσοτικοποιηθούν οι οπτικές διαφορές.

Στο matlab η μετατροπή γίνεται με τις συναρτήσεις *makecform* και *applycform*:
`cform=makecform('srgb2lab');`
`lab_f=applycform(f,cform);`

Βήμα3: Ταξινόμηση χρωμάτων στο **'*a*b*'** χώρο χρησιμοποιώντας ταξινόμηση k-μέσων

Η ταξινόμηση (ή ομαδοποίηση) είναι ένας τρόπος για να γίνει διαχωρισμός των αντικειμένων. Η ταξινόμηση k-μέσων χειρίζεται το κάθε αντικείμενο σα να έχει μια θέση στο χώρο. Βρίσκει τέτοια τμήματα ώστε τα αντικείμενα μέσα σε κάθε ομάδα να είναι τόσο κοντά το ένα με το άλλο όσο το δυνατόν, και τόσο μακριά από αντικείμενα άλλων ομάδων, όσο είναι δυνατόν. Η μέθοδος ταξινόμησης K-μέσων απαιτεί να προσδιορίσουμε τον αριθμό των κομματιών στα οποία θέλουμε να τμηματοποιηθεί η εικόνα και μια μέθοδο μέτρησης της απόστασης για να ποσοτικοποιηθεί το πόσο κοντά μεταξύ τους βρίσκονται δύο αντικείμενα.

Εφόσον η πληροφορία χρώματος βρίσκεται στο **'a*b*'** χώρο τα αντικείμενά μας είναι pixel με **'a*'** και **'b*'** τιμές. Χρησιμοποιούμε την συνάρτηση `kmeans` του Matlab για να ταξινομήσουμε τα αντικείμενα σε δύο τμήματα χρησιμοποιώντας την Ευκλείδεια μέτρηση απόστασης:

```
nColors=2;

ab=double(lab_f(:,:,2:3));
nrows=size(ab,1);
ncols=size(ab,2);
ab=reshape(ab,nrows*ncols,2);

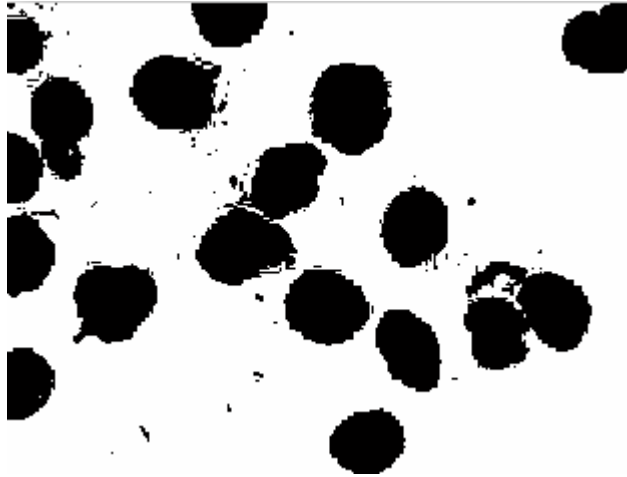
%repeat clustering 3 times to avoid local minima
[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);
```

Βήμα4: Τοποθέτηση ετικέτας σε κάθε pixel της εικόνας χρησιμοποιώντας τα αποτελέσματα της `kmeans`

Για κάθε αντικείμενο, η `kmeans` επιστρέφει ένα δείκτη που αντιστοιχεί σε ένα τμήμα. Το `cluster_center` που εξάγει η `kmeans` θα χρησιμοποιηθεί και πιο κάτω στον αλγόριθμό μας. Τοποθετούμε ετικέτες σε κάθε pixel της εικόνας με βάση το `cluster_index`:

```
pixel_labels=reshape(cluster_idx,nrows,ncols);
subplot(2,2,2)
imshow(pixel_labels,[]),title('Image labeled by cluster index');
```

Το αποτέλεσμα φαίνεται στην εικόνα 5.5 παρακάτω:



ΕΙΚΟΝΑ 5.5 Η εικόνα μετά την τοποθέτηση ετικετών

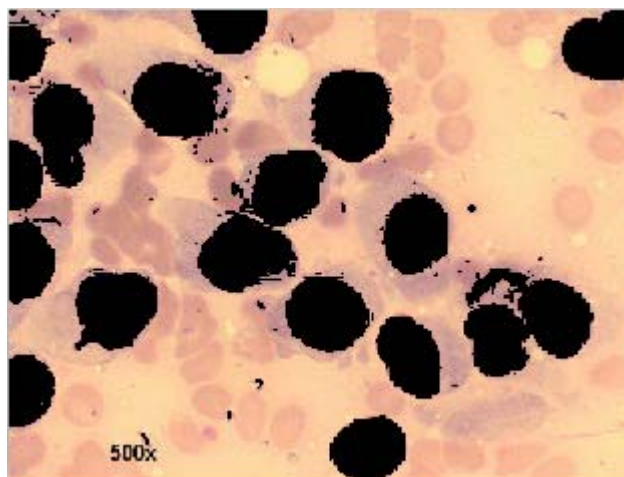
Βήμα5: Δημιουργία εικόνων που τμηματοποιούν την εικόνα με βάση το χρώμα

Με τη βοήθεια της μεταβλητής *pixel_labels*, γίνεται διαχωρισμός των αντικειμένων(κυττάρων) στην εικόνα 5.4 από το χρώμα, το οποίο θα έχει ως αποτέλεσμα τη δημιουργία 2 εικόνων.

```
segmented_images = cell(1,nColors);
rgb_label = repmat(pixel_labels,[1 1 3]);
```

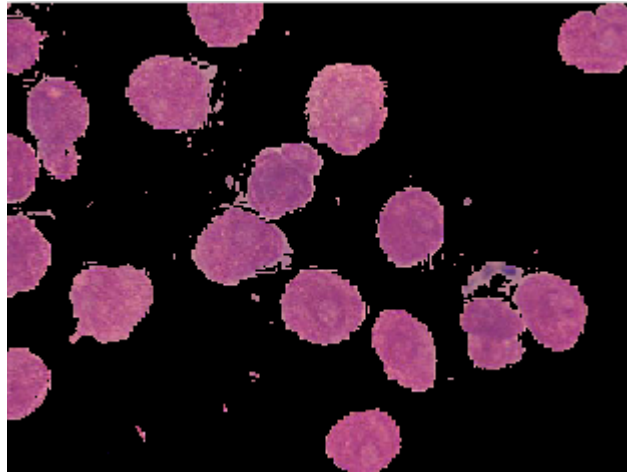
```
for k = 1:nColors
    color = f;
    color(rgb_label ~= k) = 0;
    segmented_images{k} = color;
end
```

```
imshow(segmented_images{1}), title('objects in cluster 1');
```



ΕΙΚΟΝΑ 5.6 Αντικείμενα στο cluster1

```
imshow(segmented_images{2}), title('objects in cluster 2');
```



ΕΙΚΟΝΑ 5.7 Αντικείμενα στο cluster2

Βήμα6: Κατάτμηση των κυττάρων σε ξεχωριστή εικόνα

Εδώ ανιχνεύεται η εικόνα που περιέχει τα μπλε αντικείμενα (κύτταρα). Στην εικόνα υπάρχουν μπλε σκούρα και μπλε ανοιχτά αντικείμενα, αυτά ξεχωρίζονται χρησιμοποιώντας το 'L*' επίπεδο στο 'L*a*b' χρωματικό μοντέλο. Τα κύτταρα είναι τα μπλε σκούρα.

Αφού βρεθεί το cluster με τα μπλε αντικείμενα, κατόπιν εξάγονται οι τιμές φωτεινότητας των pixel του και γίνεται κατωφλίωση με την *im2bw*.

Το *cluster_center* περιέχει την μέση τιμή των 'a*' και 'b*' για κάθε cluster. Το Μπλε cluster έχει τη δεύτερη μεγαλύτερη τιμή *cluster_center* (προσδιορίστηκε πειραματικά):

```
mean_cluster_val = zeros(nColors,1);
for k = 1:nColors
    mean_cluster_val(k) = mean(cluster_center(k));
end
[mean_cluster_val,idx] = sort(mean_cluster_val);

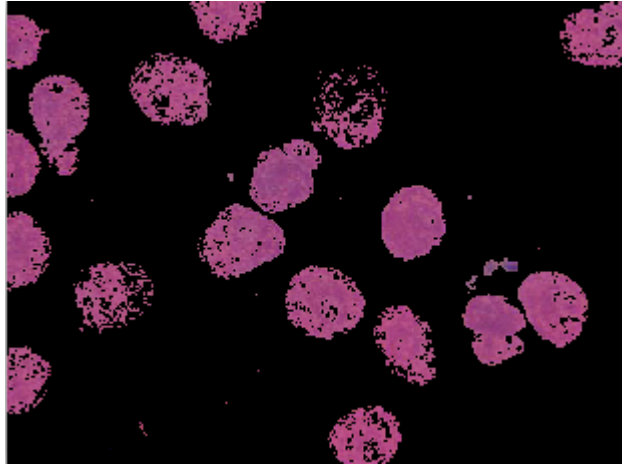
cell_cluster_num = idx(2);

L = lab_f(:, :, 1);
cell_idx = find(pixel_labels == cell_cluster_num);
L_cell = L(cell_idx);
is_light_cell = im2bw(L_cell,graythresh(L_cell));
```

Η μάσκα *is_light_cell* χρησιμοποιείται για να σημαδευτούν τα pixel που ανήκουν στα μωβ κύτταρα. Τέλος τα μωβ κύτταρα απεικονίζονται σε ξεχωριστή εικόνα.

```
cell_labels = repmat(uint8(0),[nrows ncols]);
cell_labels(cell_idx(is_light_cell==false)) = 1;
cell_labels = repmat(cell_labels,[1 1 3]);
```

```
cells = fc;  
cells(cell_labels ~= 1) = 0;  
figure;  
imshow(cells), title('violet cells');
```

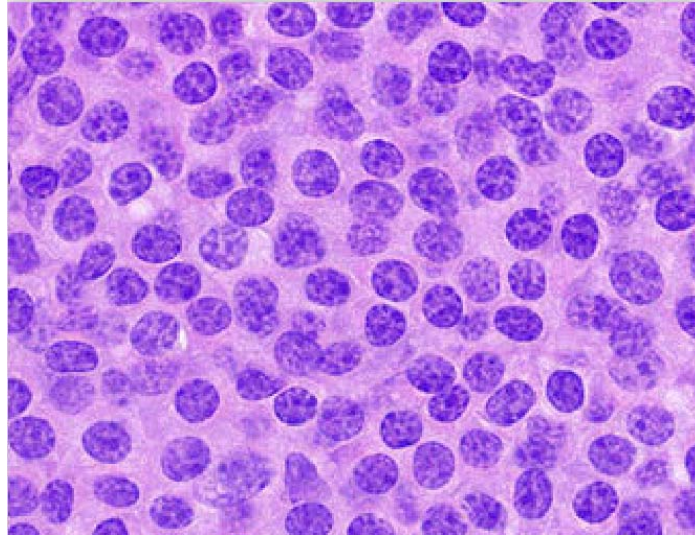


ΕΙΚΟΝΑ 5.8 Μωβ κύτταρα

5.4 Αλγόριθμοι κατάτμησης με μορφολογικούς τελεστές

5.4.1 Αλγόριθμος κατάτμησης με χρήση φίλτρου ενδιάμεσης τιμής και μορφολογικούς τελεστές με εφαρμογή σε εικόνα λεμφοκυττάρων

Η εικόνα που χρησιμοποιήθηκε σε αυτόν τον αλγόριθμο είναι μια εικόνα λεμφοκυττάρων.



ΕΙΚΟΝΑ 5.9 Αρχική εικόνα από λεμφοκύτταρα

Αρχικά εισάγεται η εικόνα στη μνήμη για να χρησιμοποιηθεί από το Matlab και μετατρέπεται σε grayscale (Εικ. 5.10β) :

```
fc=imread('lemphocytes.jpg');  
fg=rgb2gray(fc);
```

Στη συνέχεια για να γίνει η εικόνα πιο ομαλή εφαρμόζεται ένα φίλτρο ενδιάμεσης τιμής διαστάσεων 5x5 (Εικ. 5.10γ)²:

```
fm=medfilt2(fg, [5 5]);
```

Κατόπιν η εικόνα μετατρέπεται από grayscale σε ασπρόμαυρη, με βάση το κατώφλι που προκύπτει από τη μέθοδο Otsu (συνάρτηση `graythresh`) (Εικ.5.10δ):

```
T=graythresh(fm);  
fb=im2bw(fm, T);
```

Μετά ανιχνεύονται οι ακμές με τη συνάρτηση `edge` και τις μεθόδους:

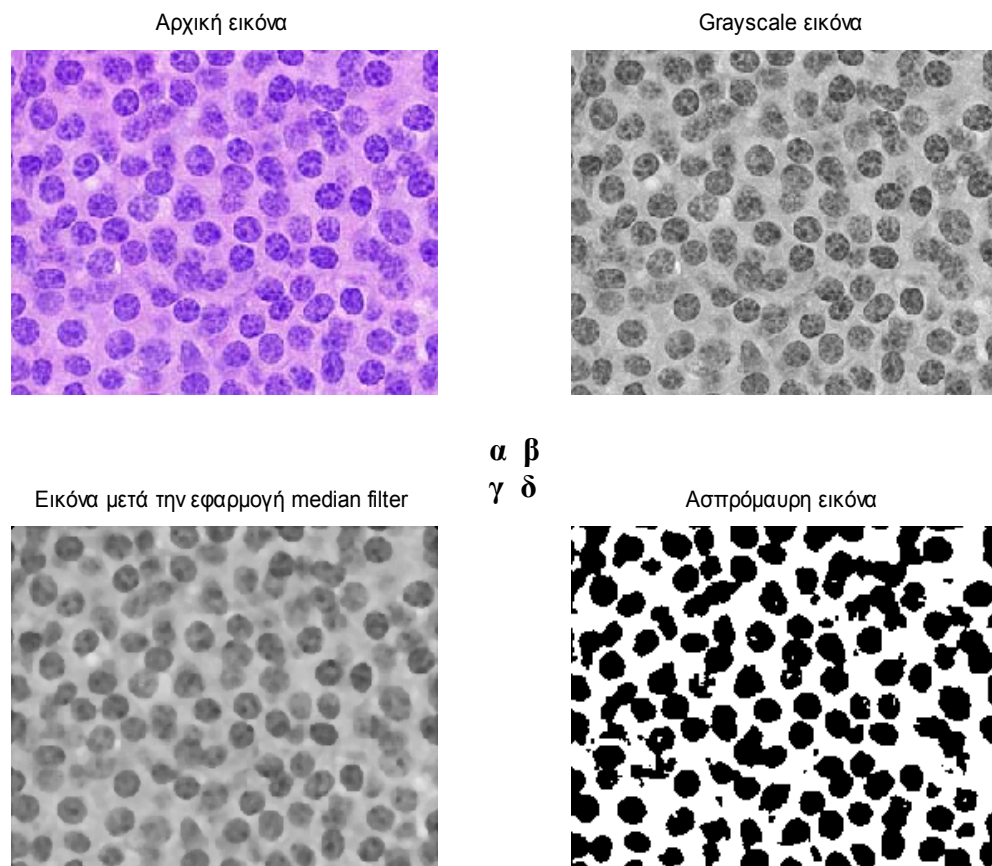
² Μπορεί αντί των παραπάνω εντολών να χρησιμοποιηθεί και η συνάρτηση προ-επεξεργασίας `smooth3(fc,[5 5]);`

Sobel (Εικ.5.11α), Prewitt(Εικ.5.11β), Roberts(Εικ.5.11γ), Laplacian of Gaussian(Εικ.5.11δ):

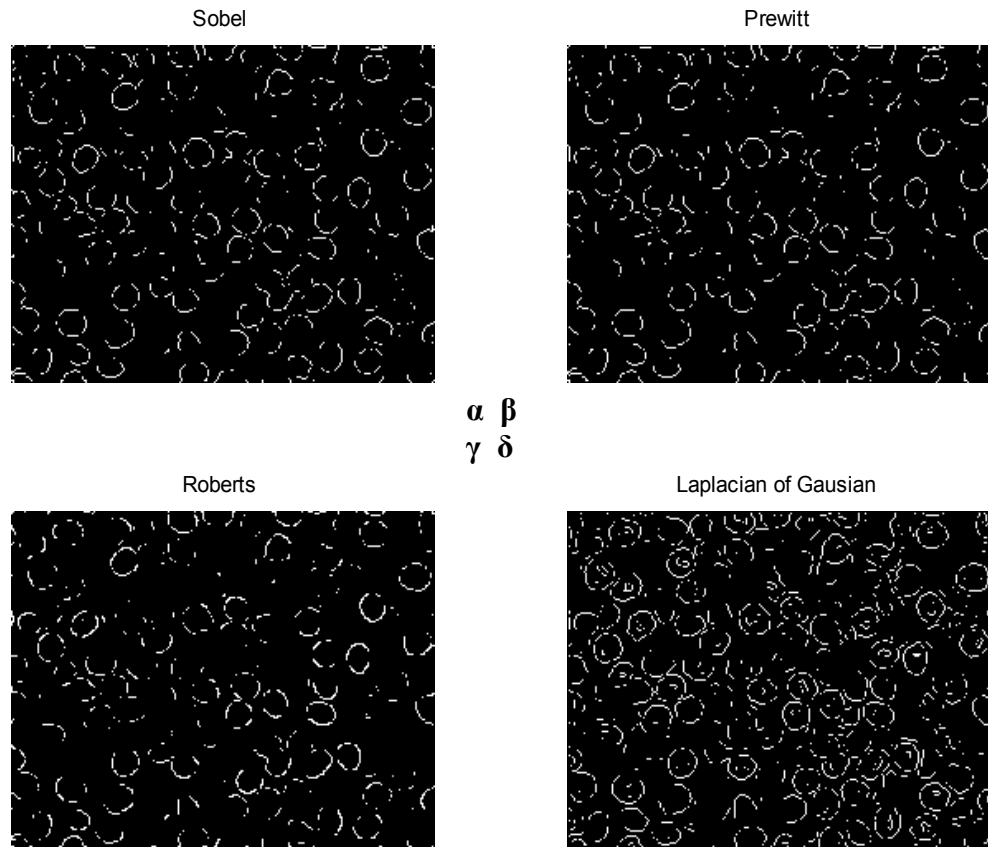
```
gS=edge(fm,'sobel');
gP=edge(fm,'prewitt');
gR=edge(fm,'Roberts');
gL=edge(fm,'log');
```

Τέλος εφαρμόζονται μορφολογικοί τελεστές στην ασπρόμαυρη εικόνα για να συγκριθούν τα αποτελέσματα από τη συνάρτηση edge. Αρχικά ορίζεται ως δομικό στοιχείο δίσκο διαμέτρου 1 και εφαρμόζεται στην αρχική ασπρόμαυρη εικόνα ένα dilation και αποθηκεύεται το αποτέλεσμα. Στη συνέχεια εφαρμόζεται πάλι στην αρχική ασπρόμαυρη εικόνα ένα erosion και αποθηκεύεται πάλι το αποτέλεσμα. Τέλος γίνεται συνένωση του αποτελέσματος της dilation με το αντίθετο της erosion, το αποτέλεσμα φαίνεται στην εικόνα 5.12: [7]

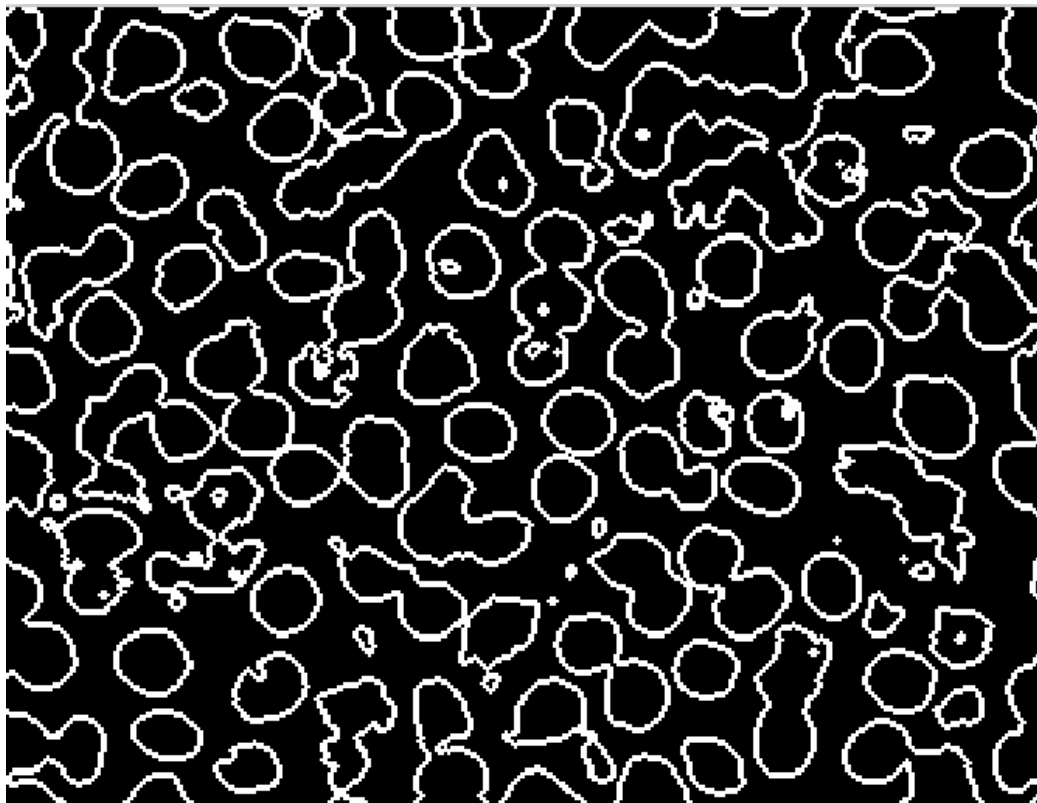
```
e=strel('disk',1);
f1=imdilate(fb,se);
f2=imerode(fb,se);
fo=f1 & ~f2;
```



ΕΙΚΟΝΑ 5.10



Εικόνα 5.11

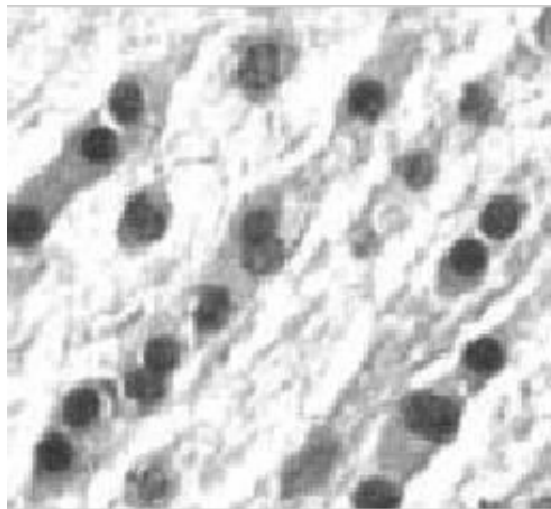


ΕΙΚΟΝΑ 5.12 Τελική εικόνα που προκύπτει με χρήση μορφολογικών τελεστών

5.4.2 Αλγόριθμος κατάτμησης με μορφολογικούς τελεστές, μέτρηση και ποσοτικοποίηση κυττάρων με εφαρμογή σε εικόνα εγκεφαλικών κυττάρων

Η αρχική εικόνα (μετά τη μετατροπή της σε grayscale για ταχύτερη επεξεργασία) φαίνεται στην εικόνα 5.13

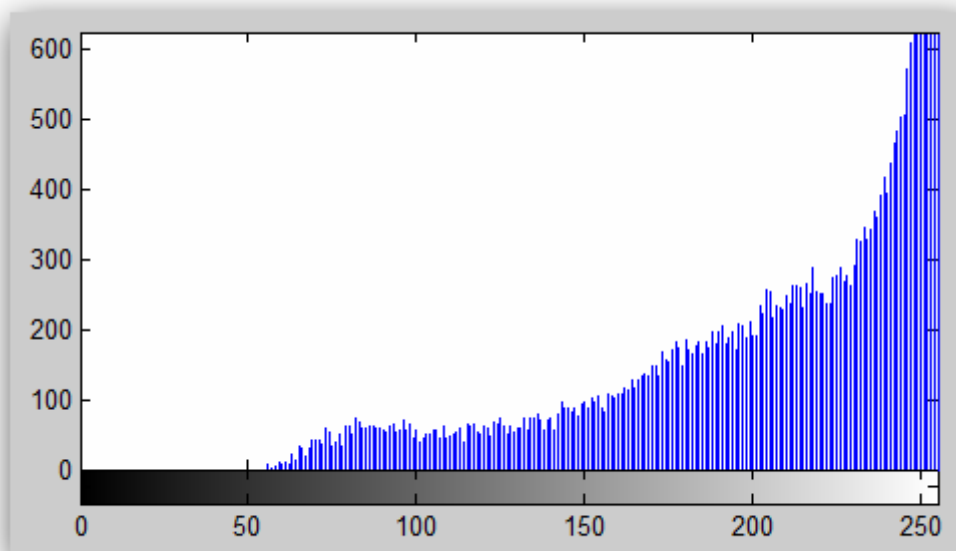
```
fc=imread('brain.cells');  
fg=rgb2gray(fc);
```



ΕΙΚΟΝΑ 5.13 Αρχική grayscale εικόνα εγκεφαλικών κυττάρων

Το ιστόγραμμα της αρχικής grayscale εικόνας δημιουργείται με την εντολή `imhist`:

```
imhist(fc);
```

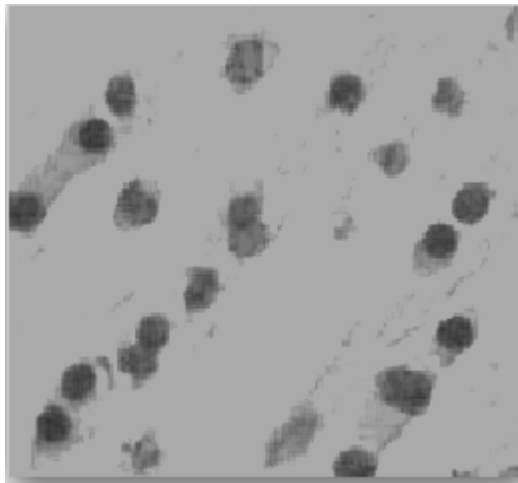


ΕΙΚΟΝΑ 5.14 Ιστόγραμμα της αρχικής grayscale εικόνας

Με τη βοήθεια του ιστογράμματος και της εντολής `imhmax` φιλτράρονται τα φωτεινά σημεία της εικόνας 5.13 ώστε να δημιουργηθεί ένα ομοιόμορφο φόντο:

```
fgf=imhmax(fg,85);
```

Το αποτέλεσμα είναι πως οι πυρήνες των κυττάρων τονίζονται όπως φαίνεται στην εικόνα 5.15



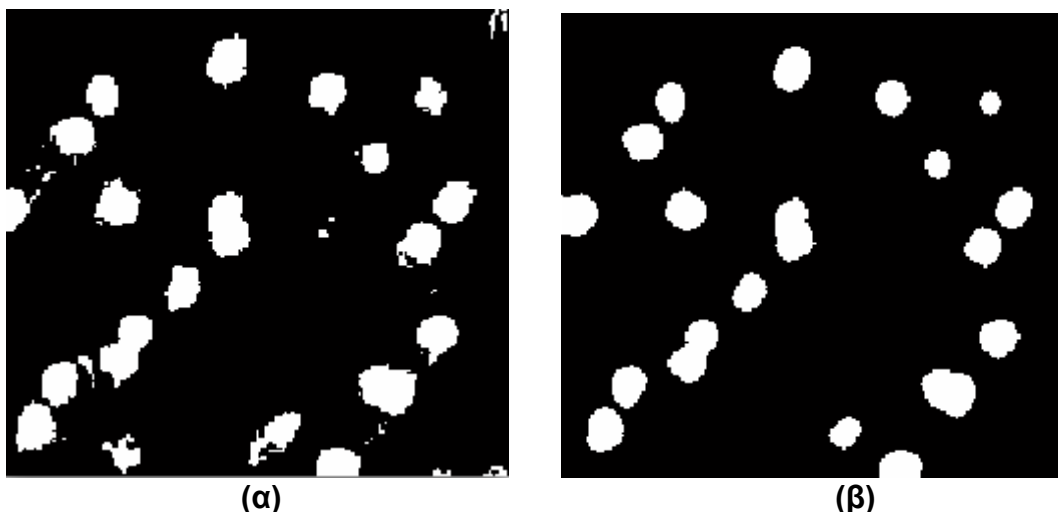
Εικόνα 5.15 Φιλτραρισμένη εικόνα με την `imhmax`

Στη συνέχεια η εικόνα μετατρέπεται σε δυαδική (με τη μέθοδο Otsu, συνάρτηση `graythresh`) και εφαρμόζεται `opening` με δομικό στοιχείο δίσκο διαμέτρου 4:

```
T=graythresh(fgf);  
fb=~im2bw(fgf,T);
```

```
se=strel('disk',4,0);  
fo=imopen(fb,se);
```

Τα αποτελέσματα φαίνονται στην εικόνα 5.12:



Εικόνα 5.16 (α) Δυαδική εικόνα σε μαύρο φόντο
(β) Δυαδική εικόνα μετά το `opening`

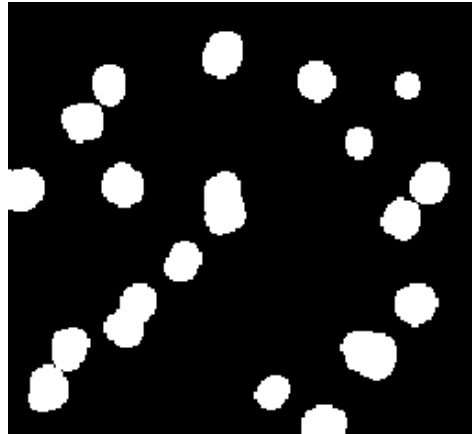
Μετά εφαρμόζεται `dilation` με δομικό στοιχείο γραμμή και στις 2 κατευθύνσεις (οριζόντια 0° και κάθετα 90°): [10]

```

se1=strel('line',2,0);
se2=strel('line',2,90);
f2=imdilate(f0,se1);
f2=imdilate(f2,se2);

```

Το αποτέλεσμα φαίνεται στην εικόνα 5.17



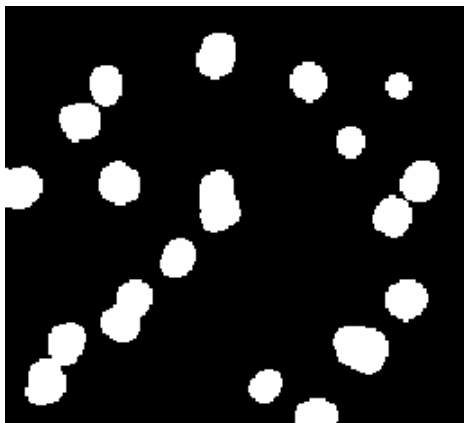
Εικόνα 5.17 Δυαδική εικόνα μετά το dilation γραμμής οριζόντια και κάθετα

Έπειτα χρησιμοποιείται η συνάρτηση **imfill** με τη παράμετρο 'holes' για να γεμίσουν οι πιθανές οπές μέσα στους πυρήνες των κυττάρων(στη προκειμένη περίπτωση δεν χρειάζεται αλλά δεν μας επηρεάζει αν χρησιμοποιήθει). Άλλη μια συνάρτηση που χρησιμοποιείται είναι η **imclearborder** η οποία αφαιρεί τα αντικείμενα που ακουμπούν στα σύνορα(στα άκρα) της εικόνας. Τα αποτελέσματα των δύο παραπάνω συναρτήσεων φαίνονται στην εικόνα 5.18:

```

f3=imfill(f2,'holes');
fcb=imclearborder(f3);

```



(α)

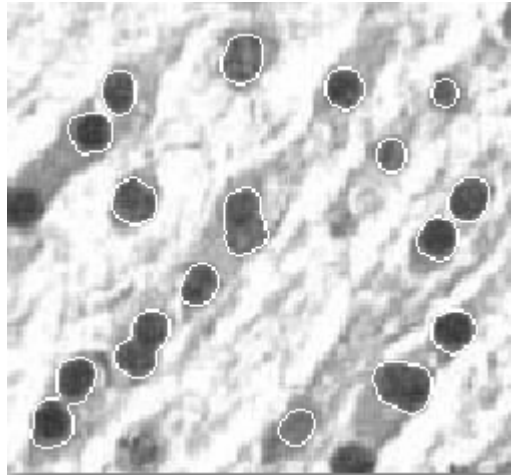


(β)

ΕΙΚΟΝΑ 5.18 (α) Εικόνα μετά το γέμισμα οπών
(β) Εικόνα μετά την αφαίρεση των πυρήνων που ακουμπούσαν στα άκρα

Στην τελική εικόνα υπολογίζεται η περίμετρος των πυρήνων που ανιχνεύθηκαν και η λογική αυτή εικόνα χρησιμοποιείται σαν μάσκα στην αρχική για φανεί η ανίχνευση.(Εικ 5.19):

```
fper=bwperim(fcb);
fmarked=fg;
fmarked(fper)=255;%Χρωματίζω την περίμετρο με άσπρο χρώμα
```



ΕΙΚΟΝΑ 5.19 Τελική εικόνα με κυκλωμένους τους πυρήνες που ανιχνεύτηκαν.

Τέλος χρησιμοποιώντας τη συνάρτηση **bwlabel** σημαδεύονται τα κύτταρα(‘με ετικέτες’) και καταμετρώνται. Η συνάρτηση **regionprops** βοηθά στη μέτρηση του εμβαδού των πυρήνων που ανιχνεύθηκαν:

```
[L,numcells]=bwlabel(fcb);
stats=regionprops(L,'Area','centroid');
cells=[stats.Area];
```

Η μεταβλητή numcells περιέχει την τιμή 15(αριθμός πυρήνων που ανιχνεύτηκαν)

Στη μεταβλητή cells περιέχονται τα εμβαδά(σε pixels) των πυρήνων:

```
>> cells
```

```
cells =
```

```
506 244 201 260 386 205 269 374 163 210 377 136 468 100
248
```

Παρατηρώντας τα αποτελέσματα φαίνεται πως η μέθοδος που ακολουθήθηκε δεν είναι ακριβής αφού ανιχνεύονται 2 πυρήνες σαν ένας.

5.5 Κατάτμηση με μετασχηματισμό Watershed

5.5.1 Κατάτμηση με watershed μετασχηματισμού απόστασης με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος

Αρχικά εισάγεται η εικόνα(Εικ.5.20(α)) στο Matlab και εφαρμόζεται ο αλγόριθμος προεπεξεργασίας `smooth2` που αναφέρθηκε παραπάνω(§5.1):

```
fc=imread('image');
fs=smooth2(fc);
```

Το αποτέλεσμα φαίνεται στην εικόνα 5.20(β)

Στη συνέχεια μετατρέπεται σε δυαδική με κατωφλίωση Otsu:

```
T=graythresh(fs);
fbw=im2bw(fs,T);
```

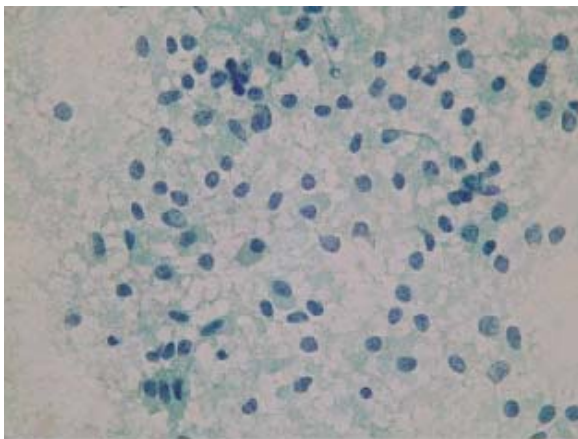
Στην εικόνα 5.20(γ) φαίνεται η δυαδική εικόνα που έχει προκύψει με άσπρο φόντο και μαύρους τους πυρήνες, έτσι η εφαρμογή του μετασχηματισμού της απόστασης σε αυτή την εικόνα δίνει τη πληροφορία για το πόσο κοντά στο φόντο βρίσκεται ο κάθε πυρήνας(μαύρο αντικείμενο), Εικ.5.20(δ)

```
D= bwdist(fbw);
```

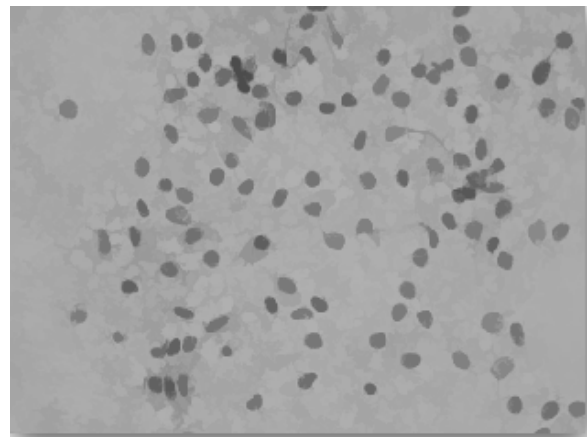
Στη συνέχεια αντιστρέφεται η εικόνα του μετασχηματισμού της απόστασης και εφαρμόζεται ο μετασχηματισμός `watershed`, έτσι ο 'πλημμυρισμός' θα ξεκινήσει από τις περιοχές ελάχιστης έντασης(δηλαδή από τις περιοχές μακριά από το φόντο) και θα 'βάλει' τα όριά του στις περιοχές που βρίσκονται τα όρια των πυρήνων: [9]

```
D=imcomplement(D);
L= watershed(D);
k=L==0;
fs=double(fs);
W=max(k,fs/255);
```

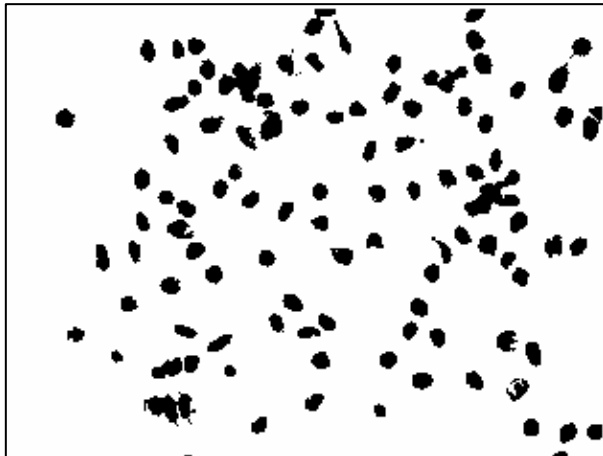
Το αποτέλεσμα φαίνεται στην τελική εικόνα 5.20(ε)



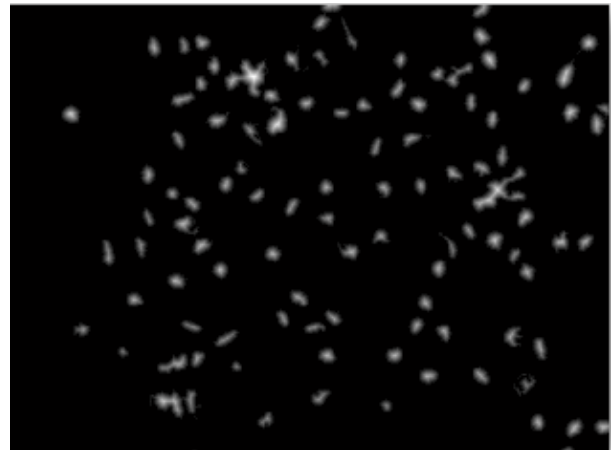
(α)



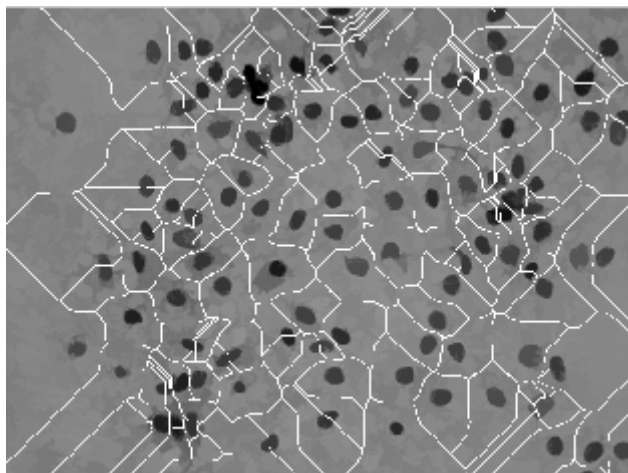
(β)



(γ)



(δ)



(ε)

ΕΙΚΟΝΑ 5.20

- (α) Αρχική κυτταροπαθολογική εικόνα παγκρέατος
- (β) Εξομαλυμένη εικόνα
- (γ) Δυαδική εικόνα μετά τη κατωφλίωση
- (δ) Μετασχηματισμός απόστασης της δυαδικής
- (ε) Κατάτμηση της εικόνας (ridge lines)

Παρατηρείται ότι η κατάτμηση της εικόνας είναι πολύ κακή, αφού αναγνωρίζονται ως πυρήνες περιοχές του υποβάθρου που περιέχουν τους πυρήνες.

5.5.2 Κατάτμηση με watershed με χρήση Gradients, με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος

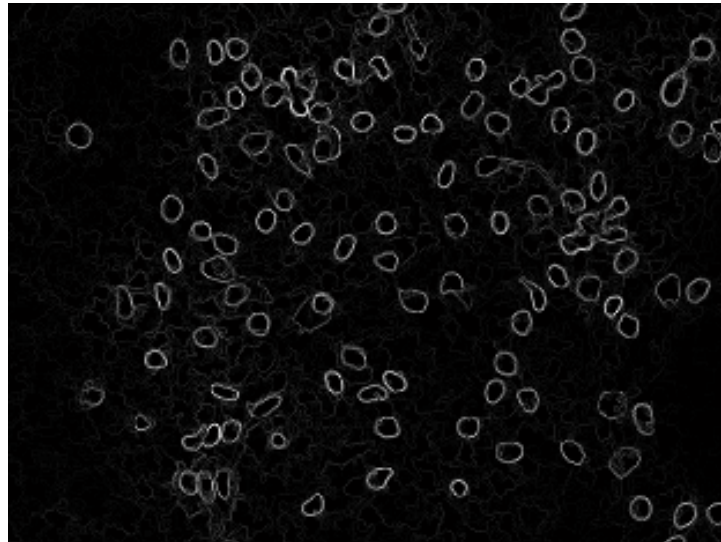
Η αρχική εικόνα είναι η ίδια με την 5.20(α). Η εικόνα εισάγεται στο matlab και ομαλοποιείται πάλι με τη συνάρτηση smooth2 :

```
fc=imread('image');
fs=smooth2(fc);
```

Στη συνέχεια γίνεται υπολογισμός του gradient της εικόνας και τον αποτέλεσμα φαίνεται στην εικόνα 5.21:

```
[x,y]=gradient(fs);
grad=sqrt(x.^2 + y.^2);
```

Στην εικόνα gradient κάθε pixel παίρνει τιμές πολύ κοντά στο μηδέν (δηλαδή αντιστοιχεί σε μαύρο) όταν γύρω από το αντίστοιχο pixel της αρχικής εικόνας δεν υπάρχει μεγάλη εναλλαγή στη φωτεινότητα, ενώ πλησιάζει στην τιμή ένα όσο αυξάνεται η εναλλαγή στη φωτεινότητα γύρω από το pixel αυτό (δηλαδή εκεί η εικόνα της gradient πλησιάζει το άσπρο).



ΕΙΚΟΝΑ 5.21 Η εικόνα της gradient της εικόνας που προέκυψε μετά την εφαρμογή

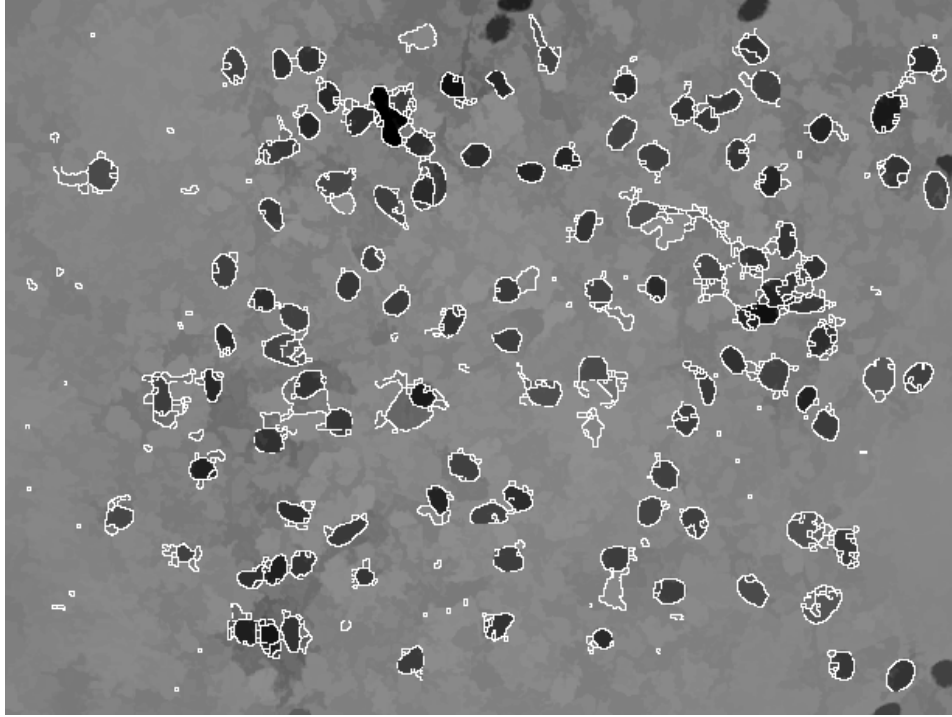
της smooth2. Οι περιοχές με ανοιχτό χρώμα είναι οι ακμές των αντικειμένων της αρχικής εικόνας.

Στη συνέχεια αφαιρούνται από την gradient εικόνα οι πυρήνες που ακουμπούν στα άκρα και έπειτα εφαρμόζεται ο μετασχηματισμός watershed στην τελική εικόνα. Ο 'πλημμυρισμός' θα ξεκινήσει από τα χαμηλότερα τμήματα της εικόνας, δηλαδή τα τμήματα όπου η μεταβολή της φωτεινότητας στην αρχική εικόνα είναι μικρή, και θα φτάσει στα υψηλότερα, δηλαδή στα τμήματα όπου η μεταβολή της φωτεινότητας είναι έντονη (δηλαδή στις ακμές των αντικειμένων). Έτσι, τα σύνορα των περιοχών θα συμπίπτουν με τα όρια των αντικειμένων: [9]

```
fc=imclearborder(grad);
L = watershed(fcb);
k=L==0;
w=max(k,fs/255);
```

Παρακάτω φαίνεται η υπέρθεση των watershed ridge lines στην αρχική εικόνα fs φαίνεται παρακάτω

Αρχική εικόνα με ridge lines του segmentation



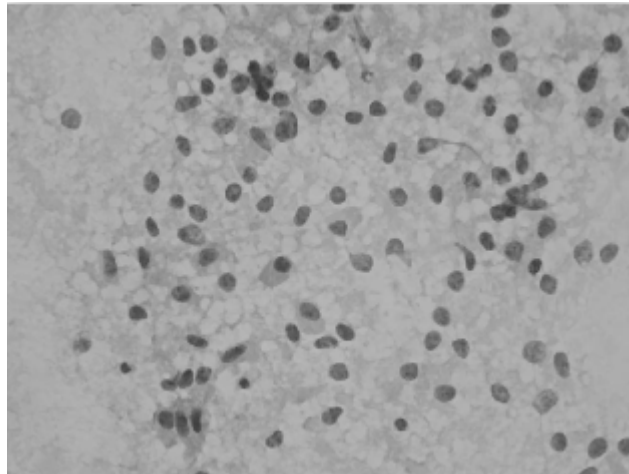
ΕΙΚΟΝΑ 5.22 Η τελική εικόνα που προκύπτει με υπέρθεση των watershed ridge lines στην αρχική εικόνα fs(oversegmentation)

Παρατηρείται εύκολα η υπερκατάτμηση(oversegmentation) των πυρήνων.

5.5.3 Κατάτμηση με **marker-controlled watershed** με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος

Αρχικά εισάγεται η εικόνα στο matlab και στη συνέχεια εξομαλύνεται με την συνάρτηση `smooth3`(§5.1), η εικόνα μετά την εξομάλυνση φαίνεται παρακάτω στην εικόνα 5.23:

```
fc=imread('image');
fs=smooth3(fc,[3 3]);
```



ΕΙΚΟΝΑ 5.23 Η εξομαλυμένη με median filter κυτταρική εικόνα

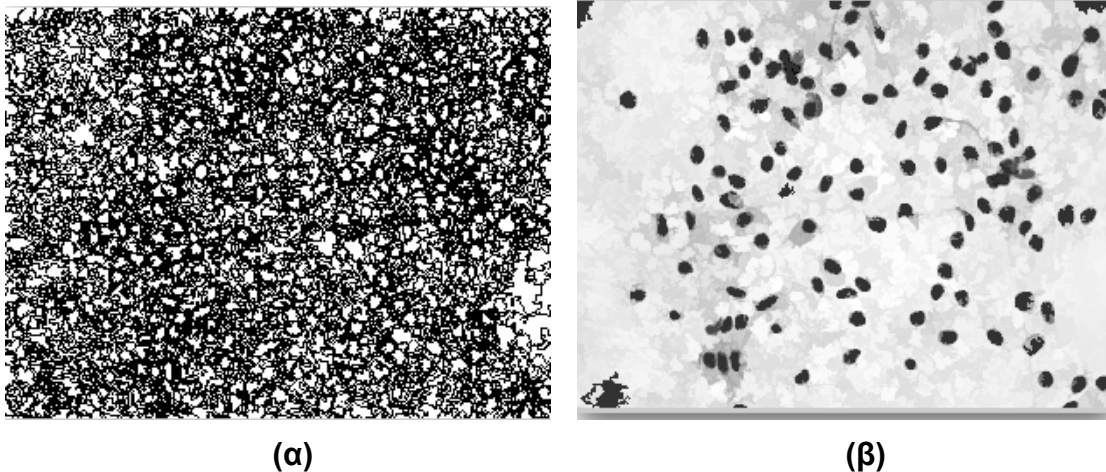
Όπως αναφέρθηκε και στο κεφάλαιο 4, αν ο μετασχηματισμός watershed εφαρμοστεί στη gradient εικόνα μιας εικόνας χωρίς να έχει γίνει προεργασία σε αυτήν θα υπάρξει υπερκατάτμηση λόγω εναλλαγής της έντασης της φωτεινότητας στο εσωτερικό των πυρήνων των κυττάρων, αλλά και του φόντου. Ένας τρόπος να αποφευχθεί αυτό είναι να αγνοηθούν κάποια ελάχιστα, ώστε ο 'πλημμυρισμός' να ξεκινάει από συγκεκριμένα σημεία μόνο κι όχι από όλα(εσωτερικά σημάδια) καθώς επίσης να σταματάει. Αυτό επιτυγχάνεται με χρήση της marker watershed κατάτμησης, στην οποία το 'νερό' ξεκινάει από συγκεκριμένα τοπικά ελάχιστα, αγνοώντας τα υπόλοιπα, που αποκαλούνται seeds (σπόροι) ή σημεία εκκίνησης.

Το πρόβλημα λοιπόν μεταφέρεται στην εύρεση αυτών των τοπικών ελαχίστων και απόρριψη κάποιων αυτόματα. Η συνάρτηση **imregionalmin(f)**, μας επιστρέφει μια δυαδική εικόνα με τις θέσεις των τοπικών ελαχίστων (Εικ. 5.24α) [9]

```
rm=imregionalmin(grad);
```

Η συνάρτηση **imextendedmin(f,h)** υπολογίζει μέσω του extended-minima μετασχηματισμού τα 'χαμηλά σημεία' της εικόνας που είναι βαθύτερα (από ένα κατώφλι h) σε σχέση με τα γύρω τους. Αυτά τα χαμηλά σημεία τα χρησιμοποιούνται σαν εσωτερικοί markers. (Εικ. 5.24β)

```
h=5;
im=imextendedmin(fs,h);
fim=fs;
fim(im)=100;
```



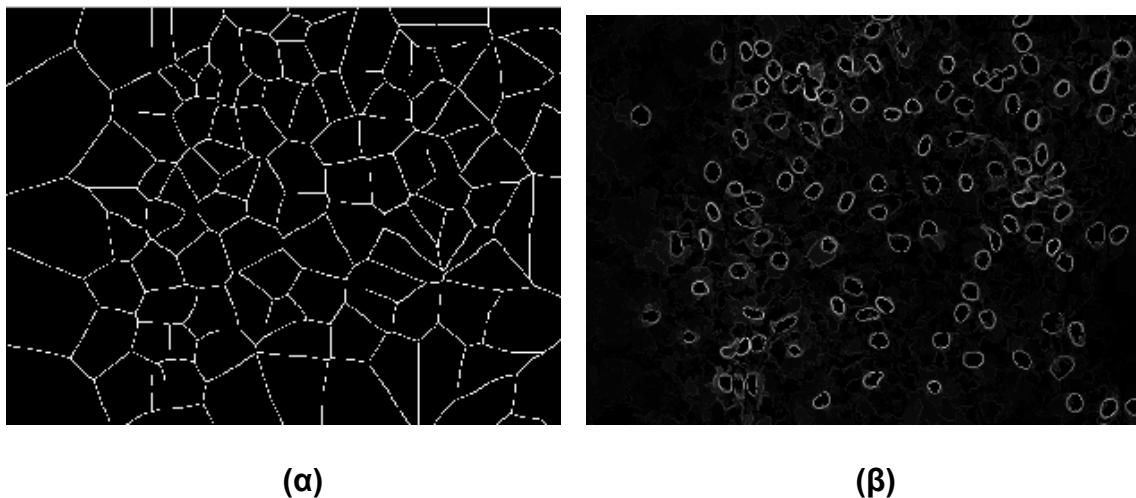
ΕΙΚΟΝΑ 5.24 (α) Εικόνα τοπικών ελαχίστων της εικόνας gradient
(β) Αρχική εικόνα με εσωτερικούς markers

Για εξωτερικοί markers θα χρησιμοποιηθούν pixels που ανήκουν σίγουρα στο φόντο. Η προσέγγιση εδώ είναι να μαρκαριστεί το φόντο βρίσκοντας pixels τα οποία βρίσκονται μεταξύ των εσωτερικών markers. Το πρόβλημα αυτό λύνεται μέσω λύσης της άλλου προβλήματος watershed: συγκεκριμένα υπολογίζεται ο μετασχηματισμός watershed του μετασχηματισμού της απόστασης της εικόνας των εσωτερικών markers(im): (Εικ. 5.25α) [9]

```
Lim=watershed(bwdist(im));  
em=Lim==0;
```

Οι εσωτερικοί και οι εξωτερικοί markers χρησιμοποιούνται για να αλλάξουμε την εικόνα gradient με τη μέθοδο minima imposition η οποία τροποποιεί μια γκρι εικόνα ώστε τα τοπικά ελάχιστα να βρίσκονται μόνο της σημαδεμένες περιοχές. Όλα τα υπόλοιπα pixels 'ανεβάζουν' τη τιμή της για να εξαφανιστούν τα τοπικά ελάχιστα, αυτό επιτυγχάνεται με την εντολή **imimposemin(f,mask)**: (Εικ. 5.25β) [9]

```
grad2=imimposemin(grad, im|em);
```

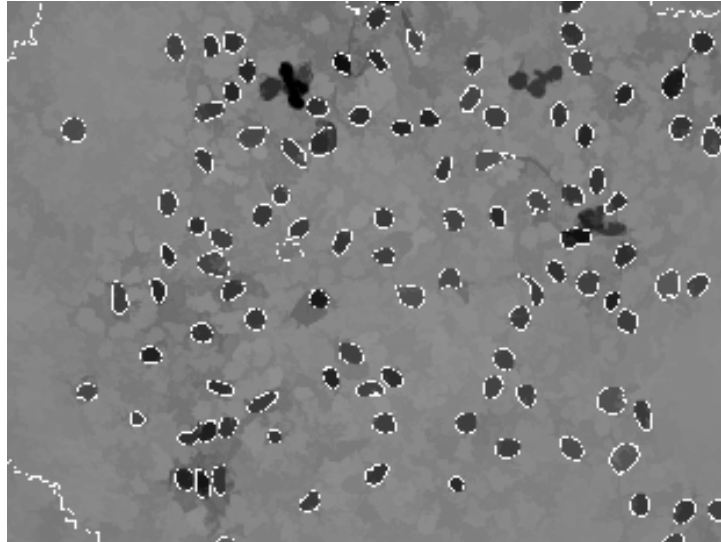


ΕΙΚΟΝΑ 5.25 (α) Εικόνα εξωτερικών markers
(β) Νέα εικόνα gradient με τους markers

Τώρα μπορεί να υπολογιστεί πλέον ο μετασχηματισμός watershed της τροποποιημένης εικόνας gradient:

```
L2=watershed(grad2);  
f2=f;  
f2(L2==0)=255;
```

Το αποτέλεσμα της παραπάνω κατάτμησης φαίνεται στην εικόνα 5.26:



ΕΙΚΟΝΑ 5.26 Τελική εικόνα κατάτμησης με marker-controlled watershed

Όπως φαίνεται στην παραπάνω εικόνα η κατάτμηση που επιτεύχθηκε είναι αρκετά καλή, αν και υπάρχουν ίχνη υποκατάτμησης. Παρόλα αυτά η μέθοδος αυτή δεν είναι αυτόματη αφού πρέπει ο χρήστης να επιλέγει την τιμή του h κάθε φορά.

5.6 Κατάτμηση μόνο με μετασχηματισμό απόστασης

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο ο μετασχηματισμός της απόστασης [9], [17] δεν είναι τίποτα άλλο παρά η απόσταση κάθε Pixel από το κοντινότερο μη μηδενικό pixel. Έτσι, αν έχουμε μια εικόνα μαύρου φόντου και λευκών αντικειμένων τότε ο πίνακας του μετασχηματισμού της απόστασης θα έχει την τιμή 1 MONO στα σύνορα των αντικειμένων με το φόντο (αφού μόνο αυτή η περιοχή έχει απόσταση 1 pixel). Χρησιμοποιώντας αυτήν την ιδιότητα, λοιπόν μπορούν να ανιχνευτούν εύκολα τα όρια των αντικειμένων (πυρήνες κυτταρικής εικόνας).

Η εικόνα που χρησιμοποιήθηκε και εδώ είναι η κυτταρική εικόνα του παγκρέατος που είδαμε παραπάνω (Εικ. 5.27α).

Αρχικά εισάγουμε την εικόνα στο Matlab και την εξομαλύνουμε με το συνάρτηση εξομάλυνσης `smooth2`: (Εικ.5.27β)

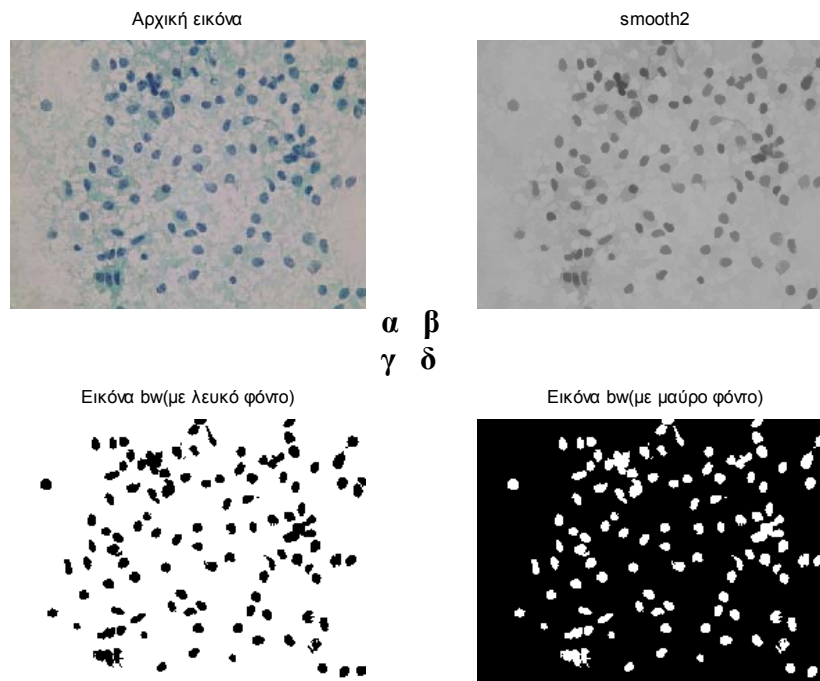
```
fc=imread('image');
fs=smooth2(fc);
```

Στη συνέχεια μετατρέπουμε την εικόνα σε δυαδική με τη `graythresh`:

```
%BHMA3: Μετατροπή σε b/w(ΛΕΥΚΟΥ ΦΟΝΤΟΥ)
T=graythresh(fs);
bw=im2bw(fs,T);
```

Η εικόνα που προκύπτει έχει λευκό φόντο (Εικ.5.27γ), όμως πρέπει η εικόνα να έχει μαύρο φόντο, ώστε ο μετασχηματισμός της απόστασης σε αυτήν να δώσει την τιμή 1 στα όρια των αντικειμένων (πυρήνες). Για το λόγο αυτό υπολογίζεται η συμπληρωματική της εικόνας αυτής: (Εικ. 5.27δ)

```
bw=imcomplement(bw);
```



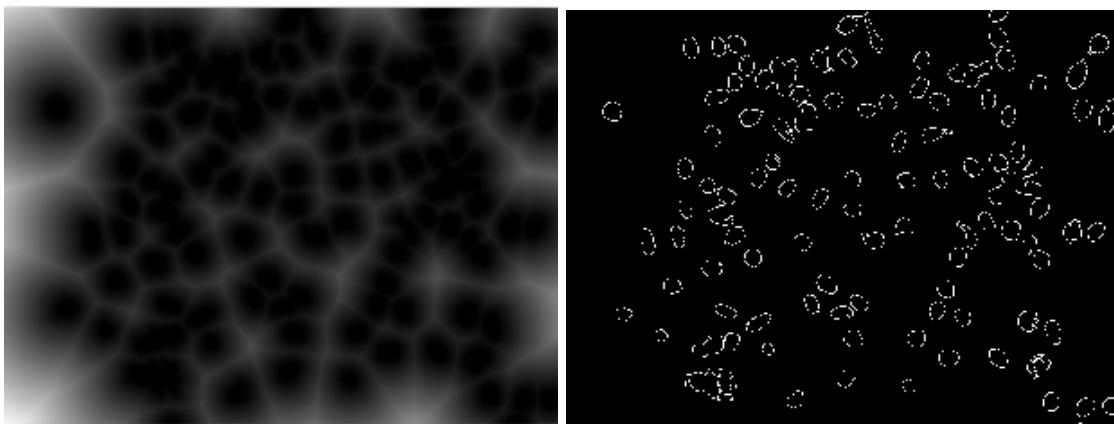
EIKONA 5.27

Έτσι τώρα που η *bw* έχει μαύρο φόντο εφαρμόζεται ο μετασχηματισμός της απόστασης σε αυτήν: (Εικ.5.28α)

```
DT= bwdist(bw);
```

Ανιχνεύονται στη συνέχεια τα όρια των πυρήνων της κυτταρικής εικόνας τα οποία βρίσκονται εκεί που τα ρίχει έχουν τιμή 1 όπως προαναφέρθηκε: (Εικ.5.28β)

```
nuclei=DT==1
```



(α)

(β)

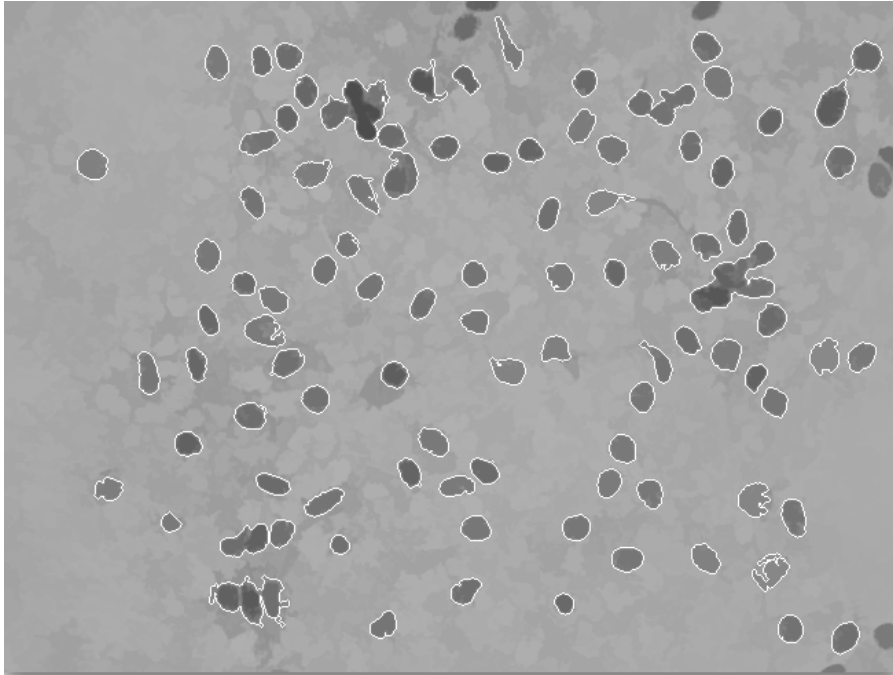
ΕΙΚΟΝΑ 5.28 (α) Εικόνα μετασχηματισμού απόστασης της *bw*
(β) Εικόνα που περιέχει τα όρια των πυρήνων

Τέλος αφαιρούνται τα όρια των πυρήνων που ακουμπούν στα άκρα της εικόνας και γίνεται υπέρθεση των τελικών ορίων στην αρχική εικόνα *fs*: (Εικ. 5.29)

```
nuclei=imclearborder(nuclei);
```

```
fs=double(fs);
```

```
w=max(nuclei,fs/255);
```



ΕΙΚΟΝΑ 5.29 Τελική εικόνα κατάτμησης

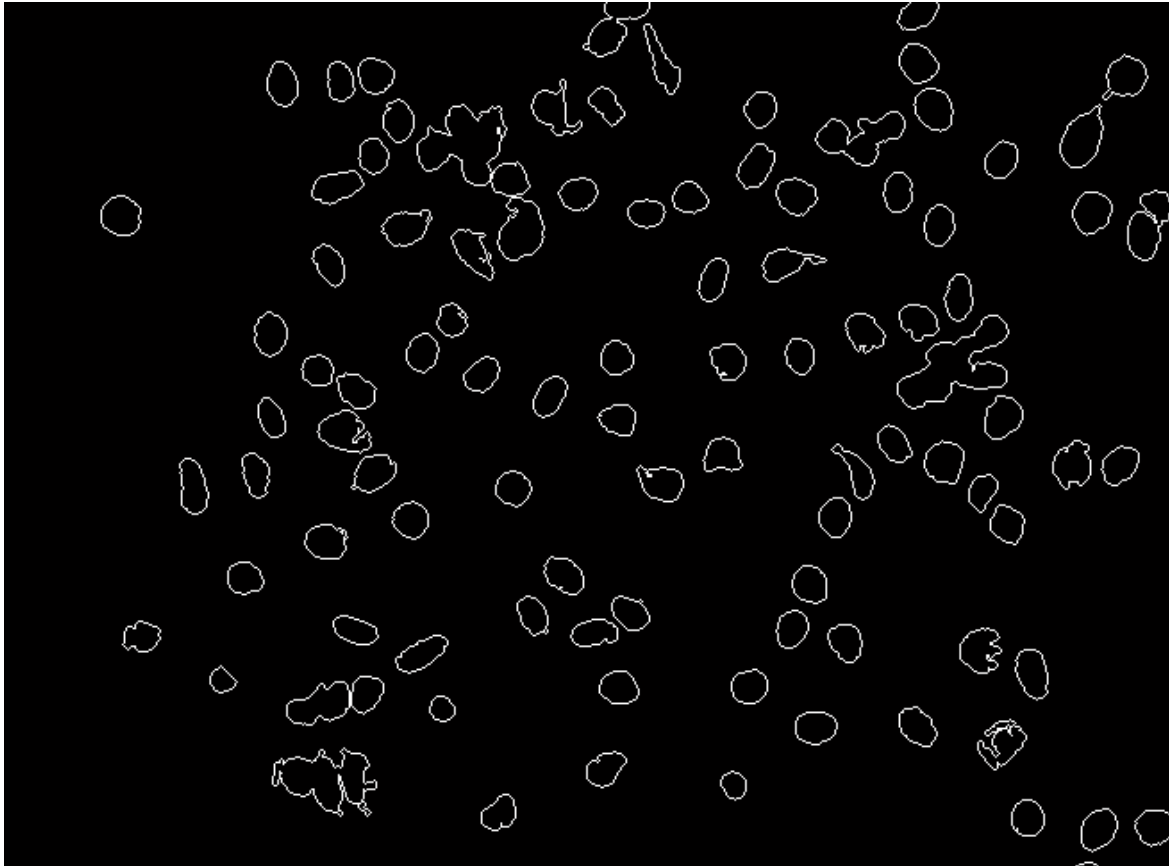
Όπως φαίνεται παραπάνω αυτή είναι η καλύτερη κατάτμηση που έχει επιτευχθεί, έχει αρκετά καλή υπολογιστική ταχύτητα και τέλος είναι απολύτως αυτόματη αφού ο χρήστης δεν χρειάζεται να κάνει τίποτα πέρα από την εισαγωγή της εικόνας.

Έτσι λοιπόν η μέθοδος αυτή θα προτιμηθεί στην συνέχεια και για να γίνει εξαγωγή πληροφορίας σχετικά με τους πυρήνες που ανιχνεύτηκαν.

5.7 Μέτρηση χαρακτηριστικών των κυττάρων που ανιχνεύτηκαν

Σε όλους τους προηγούμενους αλγορίθμους το θέμα ήταν η κατάτμηση της εικόνας, δηλαδή η ανίχνευση των πυρήνων στην τελευταία κυτταρική εικόνα του παγκρέατος. Η καλύτερη και απλούστερη μέθοδος ανίχνευσης των πυρήνων ήταν η εφαρμογή του μετασχηματισμού της απόστασης στη δυαδική της αρχικής εικόνας με μαύρο φόντο.

Αφού λοιπόν η εικόνα εισήχθη στο Matlab και εφαρμόστηκε όλη τη διαδικασία της παραγράφου 5.6 το αποτέλεσμα είναι η παρακάτω εικόνα(Εικ.5.30) με τα όρια των αντικειμένων(δηλ. των πυρήνων) που ανιχνεύτηκαν:



ΕΙΚΟΝΑ 5.30 Εικόνα με τα όρια των αντικειμένων που ανιχνεύτηκαν

Στη συνέχεια κατασκευάστηκε μια συνάρτηση η *Characteristics.m* η οποία υπολογίζει δυο βασικά χαρακτηριστικά από την εικόνα των ορίων των πυρήνων: το εμβαδό κάθε κυττάρου και τη κυκλικότητά του, επίσης κατατάσσει τα κύτταρα σε δύο κατηγορίες ανάλογα με το αν το εμβαδό είναι μικρότερο του μέσου όρου των εμβαδών όλων των κυττάρων ή όχι, και σε άλλες δύο ανάλογα με το αν η κυκλικότητά του είναι μεγαλύτερη του μέσου όρου των κυκλικοτήτων όλων των κυττάρων ή όχι.

Με τη συνάρτηση **imfill(f,'holes')** γεμίζει το εσωτερικό των ορίων των πυρήνων της εικόνας 5.30 και το αποτέλεσμα φαίνεται στην εικόνα 5.31



ΕΙΚΟΝΑ 5.31 Εικόνα με ‘γεμισμένα’ τα όρια των αντικειμένων που ανιχνεύτηκαν

Στη συνέχεια με τη συνάρτηση **bwboundaries(f,'noholes')** κατασκευάστηκε ο πίνακας L με τις ετικέτες των αντικειμένων (πυρήνων) χωρίς ετικέτες στις οπές . Επίσης η συνάρτηση αυτή επιστρέφει και έναν άλλο πίνακα B με τις συντεταγμένες των ορίων των αντικειμένων της f.

Μετά με την συνάρτηση **label2rgb** εμφανίζονται τα αντικείμενα χρωματισμένα διαφορετικά με τη βοήθεια του Label matrix.

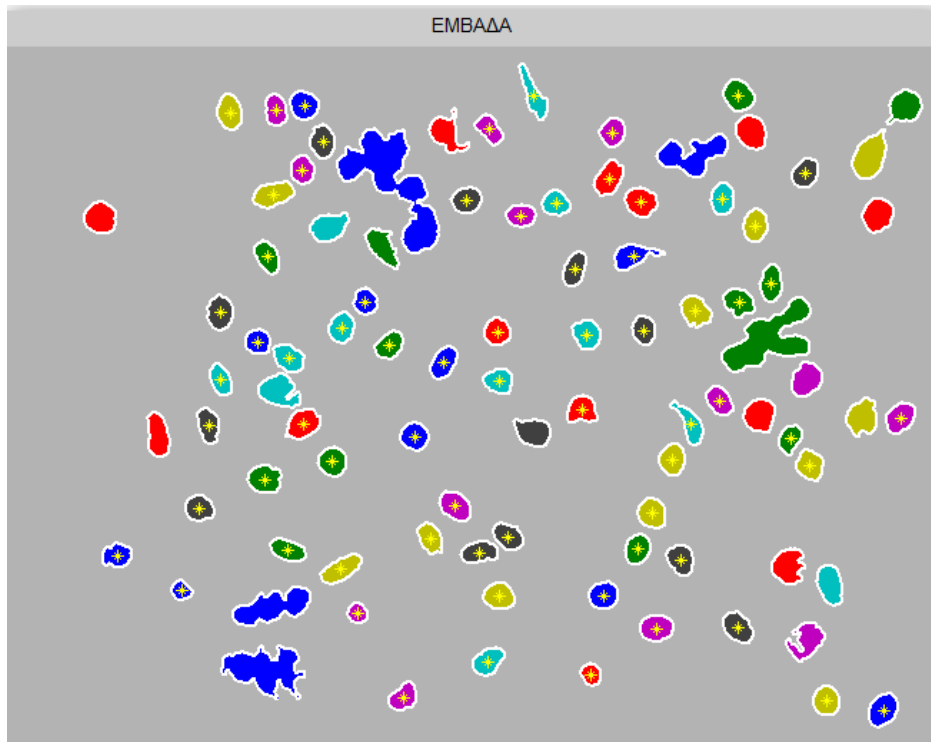
Η πιο βασική συνάρτηση που χρησιμοποιήθηκε για να εξαχθούν πληροφορίες για κάθε πυρήνα ήταν η **regionprops**. [21] Η συνάρτηση αυτή έδωσε το εμβαδόν κάθε κυττάρου.

Έπειτα με τη βοήθεια του πίνακα B που επέστρεψε η **bwboundaries** υπολογίστηκε την κυκλικότητα κάθε αντικειμένου που δίνεται από το τύπο: [20]

$$\text{Κυκλικότητα} = \frac{4 \cdot \pi \cdot \text{εμβαδόν}}{\text{περίμετρος}^2}$$

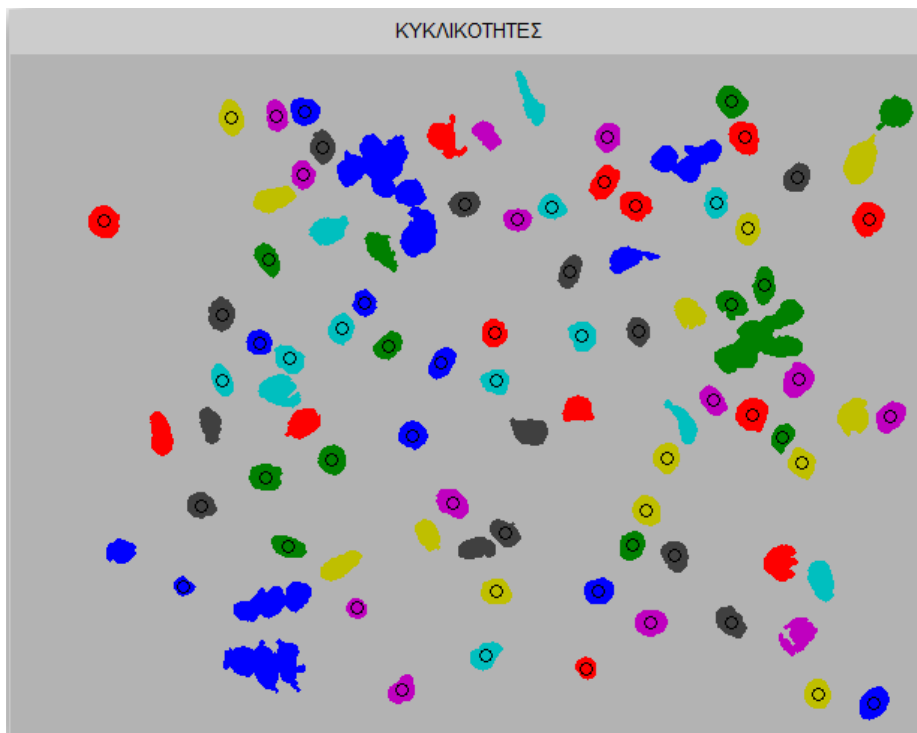
```
delta_sq = diff(boundary).^2;
perimeter = sum(sqrt(sum(delta_sq,2)));
area = stats(k).Area;
Roundness(k) = 4*pi*area/perimeter^2;
```

Κατόπιν σημαδεύονται οι πυρήνες των κυττάρων με εμβαδό μικρότερο του μέσου όρου έτσι ώστε να διαχωριστούν από τα συσσωματώματα(Εικ. 5.32)



ΕΙΚΟΝΑ 5.32 Εικόνα με επιλεγμένους τους πυρήνες(κίτρινος αστερίσκος*) με εμβαδόν μικρότερο του ΜΟ

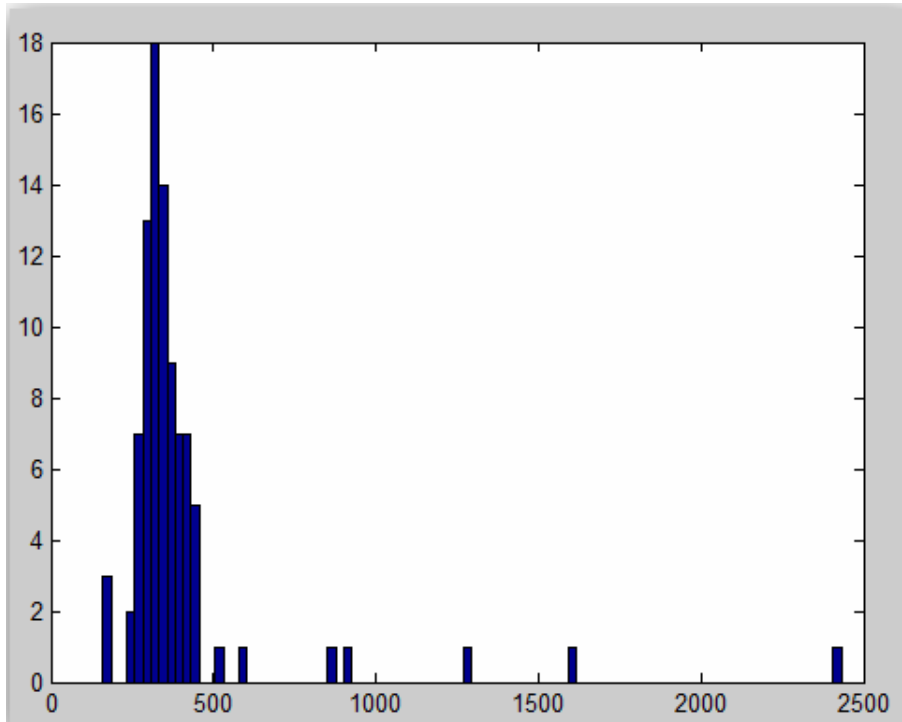
Επίσης σημαδεύονται οι πυρήνες με κυκλικότητα μεγαλύτερη του ΜΟ(επιλέχθηκε αυτή η συνθήκη γιατί οι περισσότεροι πυρήνες είναι σχεδόν κυκλικοί) (Εικ.5.33):



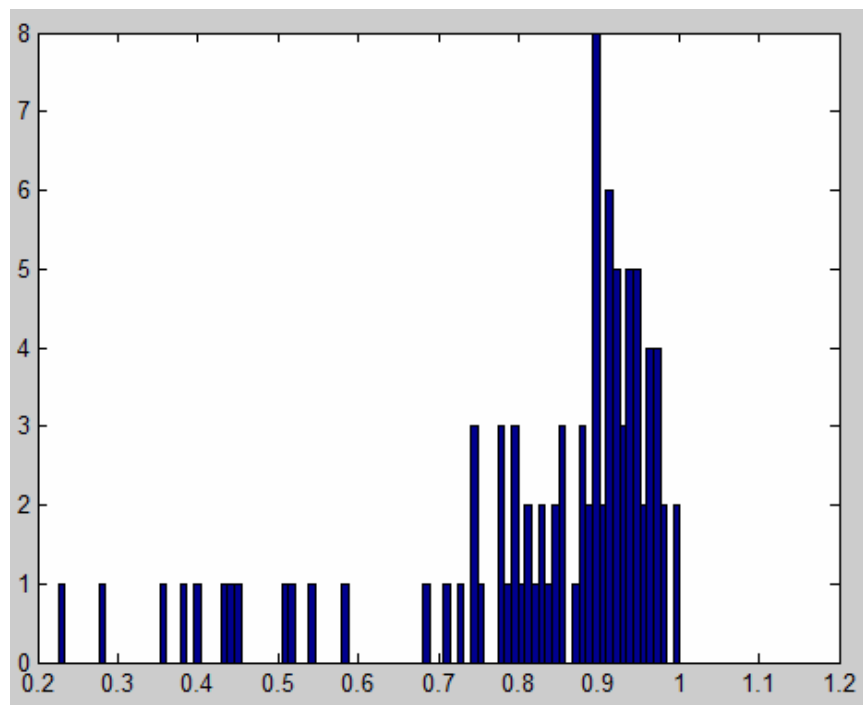
ΕΙΚΟΝΑ 5.33 Εικόνα με επιλεγμένους τους πυρήνες(μαύρος κύκλος) με κυκλικότητα μεγαλύτερη του ΜΟ

Τέλος υπολογίστηκαν κάποια στατιστικά στοιχεία σχετικά με το εμβαδόν και τη κυκλικότητα των κυττάρων.

Παρακάτω φαίνεται το ιστόγραμμα των εμβαδών (σε pixels) όλων των αντικειμένων που αναγνωρίστηκαν (Εικ. 5.34) καθώς και το ιστόγραμμα με τις κυκλικότητες όλων των αντικειμένων (Εικ. 5.35):



ΕΙΚΟΝΑ 5.34 Ιστόγραμμα των εμβαδών (σε pixels) όλων των αντικειμένων



ΕΙΚΟΝΑ 5.35 Ιστόγραμμα των κυκλικοτήτων όλων των αντικειμένων

Μερικές πληροφορίες εξήχθησαν για τους πυρήνες που ανιχνεύτηκαν στην κυτταρική εικόνα του παγκρέατος φαίνονται παρακάτω:

ΕΞΟΔΟΣ ΣΤΟ COMMAND WINDOW ΤΟΥ MATLAB:

ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ=92
ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ=36777 pixels

ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΑΤΟΜΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ
ΤΟ ΕΜΒΑΔΟ=70
ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΑΤΟΜΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ
ΤΟ ΕΜΒΑΔΟ=22227 pixels

ΣΥΝΟΛΙΚΟΣ ΣΥΣΣΩΜΑΤΩΜΑΤΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ ΤΟ
ΕΜΒΑΔΟ=22
ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΣΥΣΣΩΜΑΤΩΜΑΤΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ ΤΟ
ΕΜΒΑΔΟ=14550 pixels

ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ
ΤΟ ΕΜΒΑΔΟ=62
ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ
ΤΟ ΕΜΒΑΔΟ=19918 pixels

ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΜΗ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ
ΒΑΣΗ ΤΟ ΕΜΒΑΔΟ=30
ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΜΗ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ
ΒΑΣΗ ΤΟ ΕΜΒΑΔΟ=16859 pixels

5.8 Συμπεράσματα και μελλοντική εργασία

Στην παρούσα διπλωματική εργασία παρουσιάστηκαν διάφοροι αλγόριθμοι για την επεξεργασία κυτταροπαθολογικής εικόνας μικροσκοπίου με σκοπό την απομόνωση των καρκινικών πυρήνων. Η βασική λειτουργία των αλγορίθμων αυτών ήταν η κατάτμηση της εικόνας. Η κατάτμηση μιας κυτταροπαθολογικής εικόνας είναι το βασικότερο στάδιο της επεξεργασίας της για την εξαγωγή χρήσιμης πληροφορίας για την εικόνα, όπως είναι η πληροφορία του μεγέθους(εμβαδόν πυρήνα) και του σχήματος(κυκλικότητα).

Αναλυτικότερα όλοι οι αλγόριθμοι που αναπτύχθηκαν υλοποιούν προεπεξεργασία της εικόνας και στη συνέχεια κατάτμηση. Η επεξεργασία φαίνεται να είναι το απλό κομμάτι της κατάτμησης όμως αν αυτό το κομμάτι δεν υλοποιηθεί σωστά τότε η κατάτμηση θα αποτύχει. Μετά την επεξεργασία όπως αναφέρθηκε ακολουθεί η κατάτμηση, το δυσκολότερο κομμάτι της επεξεργασίας, εδώ υλοποιήθηκαν διάφοροι αλγόριθμοι αλλά δόθηκε ιδιαίτερη έμφαση στον αλγόριθμο watershed ο οποίος με τη χρήση markers έδωσε ικανοποιητικά αποτελέσματα. Τελικά όμως για την περαιτέρω ανάλυσή μας επιλέχθηκε ένας πιο απλός και αποτελεσματικός αλγόριθμος, ο μετασχηματισμός της απόστασης πάνω στην εικόνα ενδιαφέροντος αφού αυτή είχε προηγουμένως μετατραπεί σε ασπρόμαυρη με μαύρο φόντο, για να χρησιμοποιηθεί το χαρακτηριστικό του μετασχηματισμού της απόστασης που δίνει την τιμή 1 στα pixel των ορίων των αντικειμένων μιας μαύρου φόντου ασπρόμαυρης εικόνας(προκύπτει από τον ορισμό του μετασχηματισμού της απόστασης που είναι η απόσταση από μη μηδενικό pixel).

Τέλος αφού πραγματοποιήθηκε μια ικανοποιητική κατάτμηση έγινε εξαγωγή πληροφορίας των πυρήνων σχετικά με το μέγεθος και το σχήμα. Αυτά τα δύο βασικά χαρακτηριστικά των πυρήνων σε συνδυασμό με άλλα όπως η απόχρωση των πυρήνων, μπορούν μελλοντικά να αναλυθούν περισσότερο και να δώσουν περισσότερη πληροφορία ώστε να είναι ευκολότερο να εξαχθούν συμπεράσματα σχετικά με τη κακοήθειά τους ή όχι.

ΠΑΡΑΡΤΗΜΑ

Α. Κώδικες Αλγορίθμων κεφαλαίου 5 σε Matlab

5.1 Προεπεξεργασία Εικόνας

i) Συνολικός κώδικας εφαρμογής προεπεξεργασίας

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CELL PREPROCESSING %%
%% ©2010-11 Nikolaos Nannos %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;
%STEP 1: Read Image
[fname, pname] = uigetfile('*.jpg', 'Dialog Box');
fc = imread(strcat(pname, fname));
subplot (221);
imshow(fc); title('Αρχική εικόνα');

figure;
fs1=smooth1(fc);
subplot(221); imshow(fs1); title('smooth1 gray-Πολλαπλά opening-closing ');

fs2=smooth2(fc);
subplot(222); imshow(fs2); title('smooth2 2ο ΧΡΩΜΑΤΙΚΟ ΚΑΝΑΛΙ Πολλαπλά
opening-closing');

fs3=smooth3(fc, [5 5]);
subplot(223); imshow(fs3); title('smooth3 median filtering');

fs4=smooth4(fc);
subplot(224); imshow(fs4); title('smooth4 h-maxima');

```

ii) Συναρτήσεις προεπεξεργασίας

```

%=====
%ΟΜΑΛΟΠΟΙΗΣΗ ΚΑΙ ΑΦΑΙΡΕΣΗ ΘΟΡΥΒΟΥ ΜΕ RECONSTRUCTION%
%ΣΕ GRAYSCALE ΕΙΚΟΝΑ SMOOTH1 %
%=====
function [f] = smooth1(f)
f=rgb2gray(f);
for n=1:5
    se=strel('disk',n);
    marker=imerode(f,se);
    A=imreconstruct(marker,f);
    B=255-A;
    marker=imerode(B,se);
    f=255-imreconstruct(marker,B);
end;

```

```

%=====
%ΟΜΑΛΟΠΟΙΗΣΗ ΚΑΙ ΑΦΑΙΡΕΣΗ ΘΟΡΥΒΟΥ ΜΕ RECONSTRUCTION%
%ΣΤΟ 2ο ΧΡΩΜΑΤΙΚΟ ΚΑΝΑΛΙ SMOOTH2 %
%=====
function [f] =smooth2(f)
    f=f(:,:,2);
    for n=1:5
        se=strel('disk',n);
        marker=imerode(f,se);
        A=imreconstruct(marker,f);
        B=255-A;
        marker=imerode(B,se);
        f=255-imreconstruct(marker,B);
    end;

```

```

%=====
% ΕΞΟΜΑΛΥΝΣΗ ΤΗΣ ΕΙΚΟΝΑΣ ΚΑΙ ΑΠΑΛΟΙΦΗ ΘΟΡΥΒΟΥ ΜΕ %
%ΦΙΛΤΡΟ ΕΝΔΙΑΜΕΣΗΣ ΤΙΜΗΣ %
%=====
function [f] =smooth3(f,W)
    f=rgb2gray(f);
    f=medfilt2(f,W);

```

```

%=====
%ΕΞΟΜΑΛΥΝΣΗ ΤΗΣ ΕΙΚΟΝΑΣ ΚΑΙ ΑΠΑΛΟΙΦΗ ΘΟΡΥΒΟΥ ΜΕ ΤΗΝ%
%ΣΥΝΑΡΤΗΣΗ imhmax και τη βοήθεια της μεθόδου Otsu %
%=====
function [f] =smooth4(f)
    f=rgb2gray(f);
    T=255*graythresh(f);
    f=imhmax(f,T/2);

```


5.2 Αλγόριθμος Κατωφλίωσης

ι) Συνολικός κώδικας εφαρμογής κατωφλίωσης

```
close all;

%STEP 1: Read Image
f1=imread('pap1.jpg');
f2=imread('Myoepithelial adenoma .jpg');

fg1=rgb2gray(f1);
fg2=rgb2gray(f2);

Tfg1graythresh=graythresh(fg1);
Tfg2graythresh=graythresh(fg2);

Tfg1otsu=otsu(fg1);
Tfg2otsu=otsu(fg2);

%ΚΑΤΩΦΛΙΩΣΗ
fbwg1=im2bw(fg1,Tfg1graythresh);
fbwg2=im2bw(fg2,Tfg2graythresh);

fbwo1=im2bw(fg1,Tfg1otsu);
fbwo2=im2bw(fg2,Tfg2otsu);

subplot(221);imshow(fg1);title('Grayscale image1')
subplot(222);imshow(fg2);title('Grayscale image2')

figure;
subplot(221);imshow(fbwg1);
title(['image 1-graythresh thresh=',num2str(Tfg1graythresh)]);

subplot(222);imshow(fbwg2);
title(['image 2-graythresh thresh=',num2str(Tfg2graythresh)]);

subplot(223);imshow(fbwo1);
title(['image 1-otsu thresh=',num2str(Tfg1otsu)]);

subplot(224);imshow(fbwo2);
title(['image 2-otsu thresh=',num2str(Tfg2otsu)]);
```

ii) Συνάρτηση κατωφλίωσης Otsu

```
%ΕΥΡΕΣΗ ΤΟΥ ΚΑΤΩΦΛΙΟΥ ΤΗΣ ΕΙΚΟΝΑΣ ΑΠΟ ΤΟ ΙΣΤΟΓΡΑΜΜΑ ΜΕ ΤΗΝ
%ΤΕΧΝΙΚΗ ΤΟΥ Nobuyuki Otsu 1979
function [Thresh]=otsu(f);
[h,g]=imhist(f);

x1=length(h);
N=(ones(1,x1))*h;
a=zeros(1,x1);
n=h;
p=n/N;
gp=g.*p;
mT=(ones(1,x1))*gp;
t=1;
while t<x1+1
    for k=1:t
        a(k)=1;
    end
    w(t)=a*p;
    m(t)=a*gp;
    if (w(t)>0) & (w(t)<1)
        sb(t)=(mT*w(t)-m(t))^2/(w(t)*(1-w(t)));
    else sb(t)=0;
    end
    t=t+1;
end

opt_sb=max(sb);
Thresh=find(sb==opt_sb);
if length(Thresh)>1
    Thresh=max(Thresh);
end

Thresh=Thresh/255
```

5.3 Κατάτμηση βασισμένη στο χρώμα με χρήση ταξινομητή K-μέσων με εφαρμογή σε εικόνα καρκινικών κυττάρων στήθους

Συνολικός κώδικας

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%COLOR BASED SEGMENTATION USING K-MEANS CLUSTERING%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;

%STEP 1:  Διάβασμα εικόνας
[fname,pname] = uigetfile('*.jpg','Dialog Box');
fc = imread(strcat(pname, fname));
subplot(2,2,1);
imshow(fc)

%STEP 2:  Μετατροπή εικόνας RGB-->L*a*b*
cform=makecform('srgb2lab');
lab_fc=applycform(fc,cform);

%STEP 3:  Ταξινόμηση χρωμάτων στο 'a*b*' χρωματικό μοντέλο με χρήση K-
means clustering
ab=double(lab_fc(:,:,2:3));
nrows=size(ab,1);
ncols=size(ab,2);
ab=reshape(ab,nrows*ncols,2);
nColors=2;

%Επανάληψη του clustering 3 φορές για αποφυγή τ. ελαχίστων
[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);

%STEP 4:  Ετικετοποίηση κάθε pixel χρησιμοποιώντας το αποτέλεσμα της
kmeans
pixel_labels=reshape(cluster_idx,nrows,ncols);
subplot(2,2,2)
imshow(pixel_labels,[],title('Image labeled by cluster index'));

%STEP 5:  Δημιουργία εικόνων που τμηματοποιούν την εικόνα με βάση το χρώμα

segmented_images = cell(1,nColors);
rgb_label = repmat(pixel_labels,[1 1 3]);

for k = 1:nColors
    color = fc;
    color(rgb_label ~= k) = 0;
    segmented_images{k} = color;
end
figure;

```

```
subplot(2,2,1)
imshow(segmented_images{1}), title('objects in cluster 1');
subplot(2,2,2)
imshow(segmented_images{2}), title('objects in cluster 2');

%STEP 6: Τμηματοποίηση των κυτιάρων σε χωριστή εικόνα
mean_cluster_val = zeros(nColors,1);
for k = 1:nColors
    mean_cluster_val(k) = mean(cluster_center(k));
end
[mean_cluster_val,idx] = sort(mean_cluster_val);
cell_cluster_num = idx(2);

L = lab_fc(:,:,1);
cell_idx = find(pixel_labels == cell_cluster_num);
L_cell = L(cell_idx);
is_light_cell = im2bw(L_cell,graythresh(L_cell));

cell_labels = repmat(uint8(0),[nrows ncols]);
cell_labels(cell_idx(is_light_cell==false)) = 1;
cell_labels = repmat(cell_labels,[1 1 3]);
cells = fc;
cells(cell_labels ~= 1) = 0;
figure;
imshow(cells,[]), title('violet cells');
```

5.4 Αλγόριθμοι κατάτμησης με μορφολογικούς τελεστές

5.4.1 Αλγόριθμος κατάτμησης με χρήση φίλτρου ενδιάμεσης τιμής και μορφολογικούς τελεστές με εφαρμογή σε εικόνα λεμφοκυττάρων

Συνολικός κώδικας

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CELL SEGMENTATION WITH MEDIAN FILTER AND          %%
%% MATHEMATICAL MORPHOLOGY OPERATION                %%
%% ©2010-11 Nikolaos Nannos                         %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function []=simple_morphology1()
clear all;
close all;

%STEP 1:  Read Image
[fname,pname] = uigetfile('*.jpg','Dialog Box');
fc = imread(strcat(pname, fname));
subplot(2,2,1);
imshow(fc);title('Αρχική εικόνα');

%STEP 2:  Convert RGB image to Gray
fg=rgb2gray(fc);
subplot(2,2,2);
imshow(fg);title('Grayscale εικόνα');

%STEP 3:  Apply median filter
fm=medfilt2(fg, [5 5]);
subplot(2,2,3);
imshow(fm);title('Εικόνα μετά την εφαρμογή median filter');

%STEP 4:  Convert gray image to bw (threshold level is found by otsu's
method
%using graythresh)
T=graythresh(fm);
fb=im2bw(fm,T);
subplot(2,2,4);
imshow(fb);title('Ασπρόμαυρη εικόνα');

%STEP 5:  Edge detection with Sobels, Prewitts, Roberts, LoG
figure;
gS=edge(fm,'sobel');
gP=edge(fm,'prewitt');
gR=edge(fm,'Roberts');
gL=edge(fm,'log');
subplot(221);imshow(gS);title('Sobel');
subplot(222);imshow(gP);title('Prewitt');
subplot(223);imshow(gR);title('Roberts');
subplot(224);imshow(gL);title('Laplacian of Gaussian');

%STEP 6:  Morphological process
%se=strel([1 1 1;1 1 1;1 1 1]);
se=strel('disk',1);

f1=imdilate(fb,se);
f2=imerode(fb,se);
fo=f1 & ~f2;
figure;
imshow(fo);title('Μορφολογική Επεξεργασία')

```

5.4.2 Αλγόριθμος κατάτμησης με μορφολογικούς τελεστές, μέτρηση και ποσοτικοποίηση κυττάρων με εφαρμογή σε εικόνα εγκεφαλικών κυττάρων

Συνολικός κώδικας

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CELL SEGMENTATION %%
%% Counting and measuring Cells %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function []=simple_morphology2()
clear all;
close all;
%STEP 1: Read Image (Already gray)
[fname,pname] = uigetfile('*.jpg','Dialog Box');
fc = imread(strcat(pname, fname));
fg=rgb2gray(fc);
subplot(221);imshow(fg);title('Αρχική grayscale εικόνα');

%STEP 2: Φιλτράρισμα με imhmax
fgf=imhmax(fg,85);
subplot(222);imshow(fgf);title('Φιλτραρισμένη εικόνα');

%STEP 3: Μετατροπή σε binary και opening
T=graythresh(fgf);
fb=~im2bw(fgf,T);
subplot(223);imshow(fb);title('Διαδική εικόνα σε μαύρο φόντο');

se=strel('disk',4,0);
fo=imopen(fb,se);
subplot(224);imshow(fo);title('Διαδική εικόνα μετα το opening');

%STEP 4: Dilation με se γραμμής
se1=strel('line',2,0);
se2=strel('line',2,90);
f2=imdilate(fo,se1);
f2=imdilate(f2,se2);
figure;
subplot(221);imshow(f2);title('Εικόνα μετά το dilation γραμμής οριζόντια και κάθετα');

%STEP 5: Γέμισμα οπών με imfill
f3=imfill(f2,'holes');
subplot(222);imshow(f3);title('Γέμισμα οπών στα κύτταρα');

%STEP 6: Αφαίρεση κυττάρων που ακουμπούν στα σύνορα της εικόνας
fcb=imclearborder(f3);
subplot(223);imshow(fcb);title('Αφαίρεση κυττάρων που ακουμπούν στα σύνορα της εικόνας');

%STEP 7:
fper=bwperim(fcb);
fmarked=fg;
fmarked(fper)=255;%Χρωματίζω την περίμετρο με άσπρο χρώμα
subplot(224);imshow(fmarked);title('Ανιχνεύσιμα κύτταρα');

%STEP 8: Statistics
[L,numcells]=bwlabel(fcb);
stats=regionprops(L,'Area','centroid');
cells=[stats.Area];

```

5.5 Κατάτμηση με μετασχηματισμό Watershed

5.5.1 Κατάτμηση με watershed μετασχηματισμού απόστασης με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος

Συνολικός κώδικας

```

%=====
%ΚΑΤΑΤΜΗΣΗ ΜΕ ΕΦΑΡΜΟΓΗ ΤΟΥ WATERSHED ΣΤΟ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟ %
%ΑΠΟΣΤΑΣΗΣ ΤΗΣ ΕΙΚΟΝΑΣ. ΔΗΛΑΔΗ ΑΠΟ ΤΗΝ ΠΛΗΡΟΦΟΡΙΑ ΤΟΥ %
%ΣΧΗΜΑΤΟΣ ΤΩΝ ΠΥΡΗΝΩΝ-ΚΥΤΤΑΡΩΝ %
%=====
function []=watershed_distance()
clear all;
close all;

%STEP 1: Read Image
[fname, pname] = uigetfile('*.jpg', 'Dialog Box');
fc = imread(strcat(pname, fname));
subplot(221);
imshow(fc); title('Αρχική εικόνα');

%STEP 2: Ομαλοποίηση της εικόνας (περιέχει-αν χρειάζεται-τη μετατροπή σε
%Grayscale
fs=smooth2(fc);
subplot(222);
imshow(fs); title('Ομαλοποιημένη Grayscale εικόνα');

%Μετατροπή σε b/w με καταφλίωση Otsu
T=graythresh(fs);
fbw=im2bw(fs, T);
subplot(223);
imshow(fbw); title('Απρόμαυρη εικόνα');

%Υπολογισμός Μετ/σμου απόστασης
D= bwdist(fbw);
subplot(224);
imshow(D, []); title('Μετασχηματισμός Απόστασης');

D=imcomplement(D);
figure; subplot(221);
imshow(D, []); title('Συμπληρωματική του Μετασχηματισμού Απόστασης');

L= watershed(D);
subplot(222);
imshow(L, []); title('watershed segmentation using Distance transform');

k=L==0;
subplot(223);
imshow(k); title('watershed ridge lines του συμπληρωματικού μετ/σμου
απόστασης');

fs=double(fs);
W=max(k, fs/255);
subplot(224);
imshow(W, []); title('Αρχική εικόνα με ridge lines του segmentation');

```

5.5.2 Κατάτμηση με watershed με χρήση Gradients, με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος

Συνολικός κώδικας

```

%=====
%ΚΑΤΑΤΜΗΣΗ ΜΕ ΕΦΑΡΜΟΓΗ ΤΟΥ WATERSHED ΣΤΟ GRADIENT ΤΗΣ ΕΙΚΟΝΑΣ%
%ΔΗΛΑΔΗ ΤΗΝ ΠΛΗΡΟΦΟΡΙΑ ΤΩΝ ΑΚΜΩΝ ΤΩΝ ΠΥΡΗΝΩΝ-ΚΥΤΤΑΡΩΝ      %
%=====
function []=watershed_gradient()
clear all;
close all;

%STEP 1:  Read Image
[fname, pname] = uigetfile('*.jpg', 'Dialog Box');
fc = imread(strcat(pname, fname));
subplot(221);
imshow(fc);title('Αρχική εικόνα');

%STEP 2: Ομαλοποίηση της εικόνας (περιέχει-αν χρειάζεται-τη μετατροπή σε
%Grayscale
fs=smooth2(fc);
fs=double(fs);
subplot(222);
imshow(fs, []);title('Ομαλοποιημένη Grayscale εικόνα');

%Υπολογισμός του Gradient
[x, y]=gradient(fs);
grad=sqrt(x.^2 + y.^2);
subplot(223);
imshow(grad, []);title('To Gradient της grayscale εικόνας');

%Απομάκρυνση των πυρήνων που ακουμπούν στα άκρα της εικόνας
fcb=imclearborder(grad);
L = watershed(fcb);
subplot(224);
imshow(L, []);title('watershed of gradient');

k=L==0;
figure;subplot(221);
imshow(k); ('watershed ridge lines του gradient');
w=max(k, fs/255);
figure, imshow(w, []);title('Αρχική εικόνα με ridge lines του
segmentation');

```


5.5.3 Κατάτμηση με marker-controlled watershed με εφαρμογή σε κυτταροπαθολογική εικόνα παγκρέατος

Συνολικός κώδικας

```

=====
%(MARKER-CONTROLLED WATERSHED)
%ΕΦΑΡΜΟΓΗ WATERSHED ΣΕ ΣΥΓΚΕΚΡΙΜΕΝΕΣ ΠΕΡΙΟΧΕΣ
%ΤΗΣ ΕΙΚΟΝΑΣ (ΣΤΑ ΤΟΠΙΚΑ ΕΛΑΧΙΣΤΑ) ΩΣΤΕ Η ΠΛΗΜΜΥΡΑ
%ΝΑ ΑΡΧΙΖΕΙ ΑΠΟ ΤΟΥΣ ΠΥΡΗΝΕΣ ΚΑΙ ΝΑ ΣΤΑΜΑΤΑ ΣΕ
%ΣΗΜΕΙΑ ΑΝΑΜΕΣΑ ΣΤΟΥΣ ΠΥΡΗΝΕΣ
=====

function []=marker_watershed
clear all;
close all;

%STEP 1: Read Image
[fname, pname] = uigetfile('*.jpg', 'Dialog Box');
fc = imread(strcat(pname, fname));

%STEP 2: Ομαλοποίηση της εικόνας (περιέχει-αν χρειάζεται-τη μετατροπή σε
%Grayscale
fs=smooth2(fc);%
fs=double(fs);
subplot(221);
imshow(fs, []);title('(α) Αρχική Ομαλοποιημένη Grayscale εικόνα fs');
=====

%Υπολογισμός της gradient της εικόνας
[x,y]=gradient(fs);
grad=sqrt(x.^2+y.^2);

%Υπολογίζεται η περιοχή των τοπικών ελαχίστων της gradient εικόνας
rm=imregionalmin(grad);
subplot(222);imshow(rm);title('(β) Εικόνα τοπικών ελαχίστων της
gradient');

%Υπολογισμός 'χαμηλών σημείων' εικόνας, κάτω από την τιμή του h
h=5;
im=imextendedmin(fs,h)
fim=fs;
fim(im)=100;
subplot(223);imshow(fim, []);title('(γ) Εσωτερικοί Markers');

%Watershed στους internal markers για εύρεση των external markers
Lim=watershed(bwdist(im));
em=Lim==0;
subplot(224);imshow(em, []);title('(δ) Εξωτερικοί Markers');

%minima imposition των markers στην grad εικόνα
figure;
grad2=imimposemin(grad, im|em);
subplot(221);imshow(grad2, []);title('(ε)Νέα gradient με Markers');

%Υπολογισμός watershed τροποποιημένης gradient eik;onaw
L2=watershed(grad2);
f2=fs;
f2(L2==0)=255;
subplot(222);imshow(f2, []);title('(γ) Αποτέλεσμα κατάτμησης');

```

5.6 Κατάτμηση μόνο με μετασχηματισμό απόστασης

Συνολικός κώδικας

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CELL SEGMENTATION WITH Smoothing filters          %%
%% and DISTANCE TRANSFORM ONLY(In BLACK BACKGROUND)%%
%% ©2010-11 Nikolaos Nannos                          %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;

%BHMA1: Διάβασμα εικόνας
[fname, pname] = uigetfile('*.jpg', 'Dialog Box');
fc = imread(strcat(pname, fname));
subplot(221);
imshow(fc); title('Αρχική εικόνα');

%BHMA2: Εξομάλυνση εικόνας
fs=smooth2(fc);
subplot(222); imshow(fs); title('smooth2');
%=====

%BHMA3: Μετατροπή σε b/w(ΛΕΥΚΟΥ ΦΟΝΤΟΥ)
T=graythresh(fs);
bw=im2bw(fs, T);
subplot(223); imshow(bw); title('Εικόνα bw(με λευκό φόντο)');

%BHMA4: Εύρεση συμπληρωματικής της bw για να έχει ΜΑΥΡΟ ΦΟΝΤΟ
bw=imcomplement(bw);
subplot(224); imshow(bw); title('Εικόνα bw(με μαύρο φόντο)');

%BHMA5: Μετασχηματισμός Απόστασης
DT= bwdist(bw);
figure;
subplot(221); imshow(DT, []); title('Εικόνα μετ/σμου απόστασης');

%Σε ΜΑΥΡΟ ΦΟΝΤΟ μετά την εφαρμογή του μετ/σμου της απόστασης
%μόνο τα pixel της περιμέτρου γίνονται 1(απόσταση από μη μηδενικό
%pixel)
nuclei=DT==1;%Εύρεση ορίου πυρήνων κυτταρικής εικόνας
subplot(222); imshow(nuclei); title('Εικόνα ορίων των πυρήνων');

%Αφαίρεση των ορίων των πυρήνων που ακουμπούν στα όρια της εικόνας
nuclei=imclearborder(nuclei);
fs=double(fs);

%Υπέρθεση ορίων στην αρχική εικόνα fs
w=max(nuclei, fs/255);
figure;
imshow(w);

```

5.7 Μέτρηση χαρακτηριστικών των κυττάρων που ανιχνεύτηκαν

ι) Συνολικός κώδικας

```

%=====
%ΧΡΗΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΑΠΟΣΤΑΣΗΣ ΣΕ ΕΙΚΟΝΑ ΜΑΥΡΟΥ      %
%ΦΟΝΤΟΥ ΓΙΑ ΑΝΙΧΝΕΥΣΗ ΤΩΝ ΠΥΡΗΝΩΝ-ΚΑΤΑΤΜΗΣΗ           %
%ΚΑΙ ΕΞΑΓΩΓΗ ΠΛΗΡΟΦΟΡΙΑΣ ΣΧΕΤΙΚΑ ΜΕ ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ %
%ΤΩΝ ΠΥΡΗΝΩΝ                                             %
%©2011 Nannos Nikolaos                                   %
%=====

close all;
clear all;
clc;
%BHMA1: Διάβασμα εικόνας
[fname, pname] = uigetfile('*.jpg', 'Dialog Box');
fc = imread(strcat(pname, fname));
subplot(221);
imshow(fc); title('Αρχική εικόνα');

fs = smooth2(fc);

T=graythresh(fs);
[bw]=im2bw(fs, T);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%BHMA4: Εύρεση συμπληρωματικής της bw για να έχει ΜΑΥΡΟ ΦΟΝΤΟ
bw=imcomplement(bw);

%BHMA5: Μετασχηματισμός Απόστασης

DT= bwdist(bw);
subplot(222); imshow(DT, []); title('Εικόνα μετ/σμου απόστασης');

%Σε ΜΑΥΡΟ ΦΟΝΤΟ μετά την εφαρμογή του μετ/σμου της απόστασης
%μόνο τα pixel της περιμέτρου γίνονται 1 (απόσταση από μη μηδενικό
%pixel)
nuclei=DT==1;
%subplot(223);
figure; imshow(nuclei); title('Εικόνα ορίων των πυρήνων');

%Αφαίρεση πυρήνων που ακουμπούν στα όρια της εικόνας
nuclei=imclearborder(nuclei);
fs=double(fs);

%Υπέρθωση ορίων στην αρχική εικόνα fs
w=max(nuclei, fs/255);
figure, imshow(w);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[B, th1, th2, cells, alone, not_alone, round, not_round, R]=characteristics(nuclei
);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%=====
%STATISTICS %
%=====
%Συνολικός αριθμός κυττάρων και συνολικό εμβαδόν
figure;
num=length(cells);

area=sum(cells);
str1=sprintf('ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ=%d', num);
str2=sprintf('ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ=%d pixels\n', area);
disp(str1);
disp(str2);

hist(cells, num);
figure;
hist(R, num);

%ΚΑΤΑΤΑΞΗ ΜΕ ΒΑΣΗ ΤΟ ΕΜΒΑΔΟ
%=====
%Αριθμός ατομικών πυρήνων και εμβαδό όλων των ατομικών πυρήνων
num_alone=length(alone);
area_alone=sum(alone);
str1=sprintf('ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΑΤΟΜΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ
ΤΟ ΕΜΒΑΔΟ=%d', num_alone);
str2=sprintf('ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΑΤΟΜΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ ΤΟ
ΕΜΒΑΔΟ=%d pixels\n', area_alone);
disp(str1);
disp(str2);

%Αριθμός συσσωματωμάτων , και εμβαδό όλων των συσσωματωμάτων
num_not_alone=length(not_alone);
area_not_alone=sum(not_alone);

str1=sprintf('ΣΥΝΟΛΙΚΟΣ ΣΥΣΣΩΜΑΤΩΜΑΤΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ ΤΟ
ΕΜΒΑΔΟ=%d', num_not_alone);
str2=sprintf('ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΣΥΣΣΩΜΑΤΩΜΑΤΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ ΤΟ
ΕΜΒΑΔΟ=%d pixels\n', area_not_alone);
disp(str1);
disp(str2);

%ΚΑΤΑΤΑΞΗ ΜΕ ΒΑΣΗ ΤΗ ΚΥΚΛΙΚΟΤΗΤΑ
%=====
%Αριθμός κυκλικών πυρήνων και εμβαδό όλων των κυκλικών πυρήνων
num_round=length(round);
area_round=sum(round);

str1=sprintf('ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ
ΤΟ ΕΜΒΑΔΟ=%d', num_round);
str2=sprintf('ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ ΤΟ
ΕΜΒΑΔΟ=%d pixels\n', area_round);
disp(str1);
disp(str2);

```

```

%Αριθμός μη κυκλικών πυρήνων, και εμβαδό όλων των μη κυκλικών πυρήνων
num_not_round=length(not_round);
area_not_round=sum(not_round);

str1=sprintf('ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΜΗ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ
ΒΑΣΗ ΤΟ ΕΜΒΑΔΟ=%d',num_not_round);
str2=sprintf('ΣΥΝΟΛΙΚΟ ΕΜΒΑΔΟ ΜΗ ΚΥΚΛΙΚΩΝ ΠΥΡΗΝΩΝ ΠΟΥ ΑΝΙΧΝΕΥΤΗΚΑΝ ΜΕ ΒΑΣΗ
ΤΟ ΕΜΒΑΔΟ=%d pixels\n',area_not_round);
disp(str1);
disp(str2);

```

ii) Συνάρτηση υπολογισμού χαρακτηριστικών κυττάρου

```

%=====
%ΣΥΝΑΡΤΗΣΗ ΓΙΑ ΤΟΝ ΥΠΟΛΟΓΙΣΜΟ 2 ΒΑΣΙΚΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΤΩΝ ΠΥΡΗΝΩΝ:
%A) ΤΟΥ ΕΜΒΑΔΟΥ
%B) ΤΗΣ ΚΥΚΛΙΚΟΤΗΤΑΣ
%ΕΠΙΣΗΣ ΣΗΜΑΔΕΥΟΝΤΑΙ ΤΑ ΚΥΤΤΑΡΑ ΜΕ ΕΜΒΑΔΟ ΜΙΚΡΟΤΕΡΟ ΤΟΥ ΜΟ
%ΚΑΙ ΤΑ ΚΥΤΤΑΡΑ ΜΕ ΚΥΚΛΙΚΟΤΗΤΑ ΜΕΓΑΛΥΤΕΡΗ ΤΟΥ ΜΟ
%©2011 Nannos Nikolaos
%=====

function
[B,opt_area,opt_roundness,cells,alone,not_alone,round,not_round,Roundness]
=characteristics(final)

final = imfill(final,'holes');%Γέμισμα εσωτερικού πυρήνων
[B,L] = bwboundaries(final,'noholes');%Ανίχνευση ορίων αντικείμενου και
'ετικέτες' σε κάθε αντικείμενο

%Εμφάνιση του πίνακα L(label matrix) με διαφορετικά χρώματα στα
αντικείμενα(Πυρήνες)
figure;
imshow(label2rgb(L,'lines',[0.7 0.7 0.7],'shuffle'));

hold on

%Σχεδίαση περιγράμματος στους πυρήνες
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)
end

stats = regionprops(L,'Area','Centroid');

%Πίνακας εμβαδών
cells=[stats.Area];

```

```

%=====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ΣΗΜΑΔΕΜΑ ΕΒΜΑΔΩΝ ΜΙΚΡΟΤΕΡΩΝ ΤΟΥ ΜΕΣΟΥ ΟΡΟΥ%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Κριτήριο επιλογής σωστών πυρήνων και όχι συσσωματωμάτων (ΜΟ)
opt_area=mean(cells);

%Ανίχνευση μονοπύρηνων και σημάδεμα
%health: Πίνακας μονοπύρηνων
%cancer: Πίνακας συσσωματωμάτων
i1=1; i2=1;
for k = 1:length(B)
    if cells(k)<opt_area alone(i1)=cells(k);
        i1=i1+1;
        centroid = stats(k).Centroid;
        plot(centroid(1),centroid(2),'-r*');title('ΕΜΒΑΔΑ')
        %ind(i1)=k;
    else not_alone(i2)=cells(k);
        i2=i2+1;
    end
end
end
%=====

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ΕΥΡΕΣΗ ΚΥΚΛΙΚΟΤΗΤΑΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΚΑΙ ΣΗΜΑΔΕΜΕΜΑ%
%ΑΥΤΩΝ ΠΟΥ ΕΧΟΥΝ ΚΥΚΛΙΚΟΤΗΤΑ ΜΕΓΑΛΥΤΕΡΗ ΤΟΥ      %
%ΜΕΣΟΥ ΟΡΟΥ                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
imshow(label2rgb(L, 'lines', [.7 .7 .7], 'shuffle'));
hold on;
Roundness=ones(1,length(B));
for k = 1:length(B) %Για κάθε αντικείμενο (Πυρήνα)

    %Βρίσκονται οι συντεταγμένες κάθε ορίου αντικειμένου
    boundary = B{k};

    %Υπολογίζεται η περίμετρος του αντικειμένου
    delta_sq = diff(boundary).^2;
    perimeter = sum(sqrt(sum(delta_sq,2)));

    % Υπολογίζεται το εμβαδό του συγκεκριμένου αντικειμένου (πυρήνα) k
    area = stats(k).Area;

    % Υπολογίζεται το μέτρο της καμπυλότητας του πυρήνα k
    Roundness(k) = 4*pi*area/perimeter^2;

end

opt_roundness=mean(Roundness);
opt_roundness=0.8;

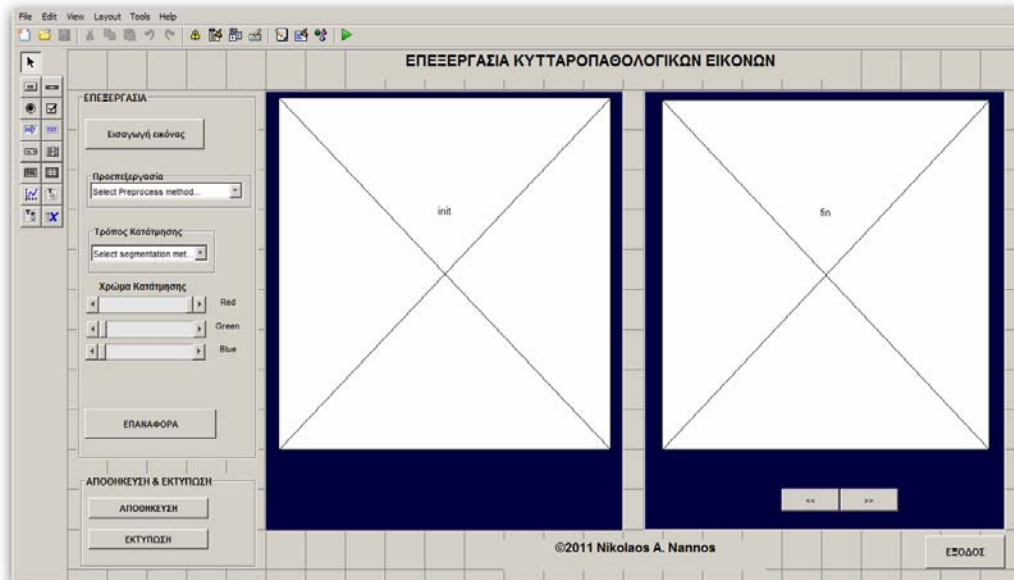
```

```
%Ανίχνευση μόνοπύρηνων και σημάδεμα
%health2: Πίνακας κυκλικών
%cancer2: Πίνακας μη κυκλικών
j1=1;j2=1;
for k = 1:length(B) %για καθε οριο
    if Roundness(k) > opt_roundness
        round(j1)=stats(k).Area;
        j1=j1+1;
        centroid = stats(k).Centroid;
        plot(centroid(1),centroid(2),'ko');title('ΚΥΚΛΙΚΟΤΗΤΕΣ');
    else not_round(j2)=stats(k).Area;
        j2=j2+1;
    end
end
```

Β. Ανάπτυξη Γραφικής Διεπιφάνειας χρήστη (GUI)

Η Γραφική διεπιφάνεια χρήστη(GUI) αναπτύχθηκε στο Matlab 7.12.0(R2011a) με τη βοήθεια του guide από τη γραμμή εντολών του Matlab, με σκοπό την καλύτερη και ευκολότερη παρουσίαση των αποτελεσμάτων της εργασίας αυτής.

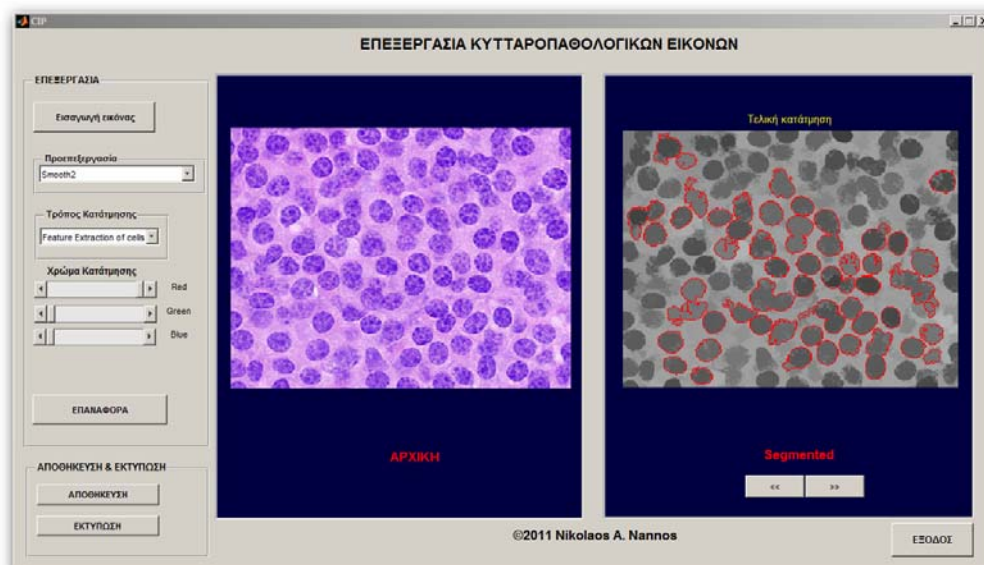
Αρχικά σχεδιάστηκαν όλα τα controls(buttons, sliders, Listboxes) και οι περιοχές (axes) όπου θα εμφανίζονται η αρχική και η τελική εικόνα. Το αποτέλεσμα φαίνεται παρακάτω:



ΕΙΚΟΝΑ Π.1

Στη συνέχεια έγινε ο προγραμματισμός του GUI, δηλαδή γράφηκε ο κώδικας για κάθε callback μέσα στο βασικό αρχείο που δημιουργήθηκε αυτόματα με το guide το CIP.m. Τέλος οι κώδικες του κεφαλαίου 5 της εργασίας μετατράπηκαν σε συναρτήσεις και αφαιρέθηκαν διάφορα στοιχεία εμφάνισης αποτελεσμάτων(imshow) αφού η αποεικόνιση γίνεται πλέον στο GUI.

Δίνεται παρακάτω το αποτέλεσμα επεξεργασίας μιας κυτταροπαθολογικής εικόνας



ΕΙΚΟΝΑ Π.2

Παρακάτω δίνεται ο κώδικας του GUI και ο κώδικας του κεφαλαίου 5 σε μορφή συναρτήσεων:

ΚΩΔΙΚΑΣ GUI (CIP.m)

```
function varargout = CIP (varargin)
% CIP M-file for CIP.fig
%   CIP, by itself, creates a new CIP or raises the existing
%   singleton*.
%
%   H = CIP returns the handle to a new CIP or the handle to
%   the existing singleton*.
%
%   CIP('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in CIP.M with the given input arguments.
%
%   CIP('Property','Value',...) creates a new CIP or raises the
%   existing singleton*. Starting from the left, property value pairs
are
%   applied to the GUI before CIP_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to CIP_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help CIP

% Last Modified by GUIDE v2.5 20-Oct-2011 15:05:47

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @CIP_OpeningFcn, ...
                  'gui_OutputFcn',  @CIP_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before CIP is made visible.
function CIP_OpeningFcn(hObject,~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to CIP (see VARARGIN)
```

```

% Choose default command line output for CIP
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CIP wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = CIP_OutputFcn(~, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as
cell array
% contents{get(hObject,'Value')} returns selected item from
popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Input.
function Input_Callback(hObject, eventdata, handles)
% hObject handle to Input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% cla(handles.init,'reset')
% cla(handles.fin,'reset')
% set(handles.init,'visible','off');
% set(handles.fin,'visible','off');

```

```

global fc fs;

cla(handles.fin,'reset')
set(handles.fin,'visible','off');

[fname,pname] = uigetfile(['*.jpg'], 'Διαλέξε μια RGB εικόνα');

fs=0;

if fname~=0
    fc = imread(strcat(pname, fname));

    %Αλλαγή περιεχομένου ετικετών στις εικόνες
    set(handles.initial,'string','ΑΡΧΙΚΗ');

        axes(handles.init);
        imshow(fc);

        axes(handles.fin);
end

% --- Executes on selection change in preprocess.
function preprocess_Callback(hObject, eventdata, handles)
% hObject    handle to preprocess (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns preprocess contents as
cell array
%         contents{get(hObject,'Value')} returns selected item from
preprocess

global fc seq ic;
global fs;
set(handles.final,'string','ΠΡΟΕΠΕΞΕΡΓΑΣΜΕΝΗ');

if ~isempty(fc)
switch get(handles.preprocess,'Value')
    case 1
        %Do nothing
        msgbox('No preprocess method was selected','Preprocess','warn');
    case 2
        fs=smooth1_gui(fc);
        seq={fs ['Όμαλοποίηση εικόνας με Reconstruction σε grayscale']};
        ic=1;
        imshow(seq{ic},[]);title(seq{ic+1},'color','y');
    case 3
        fs=smooth2_gui(fc);
        seq={fs ['Όμαλοποίηση εικόνας με Reconstruction στο 2ο κανάλι
χρώματος']};
        ic=1;
        imshow(seq{ic},[]);title(seq{ic+1},'color','y');
    case 4
        fs=smooth3_gui(fc,[5 5]);
        seq={fs ['Όμαλοποίηση εικόνας με median filter [5 5]']};
        ic=1;
        imshow(seq{ic},[]);title(seq{ic+1},'color','y');
end

```

```

    case 5
        fs=smooth4_gui(fc);
        seq={fs ['Όμαλοποίηση εικόνας με imhmax']};
        ic=1;
        imshow(seq{ic},[]);title(seq{ic+1},'color','y');
    end

else
    msgbox('Input a color image','Image input','warn');

end

% --- Executes during object creation, after setting all properties.
function preprocess_CreateFcn(hObject, eventdata, handles)
% hObject    handle to preprocess (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in Segmentation.
function Segmentation_Callback(hObject, eventdata, handles)
% hObject    handle to Segmentation (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns Segmentation contents as
cell array
%       contents{get(hObject,'Value')} returns selected item from
Segmentation

global fs;
global gs;
global fc;
global fout;
global ic seq;

set(handles.final,'string','Segmented');
if (~isempty(fs) & ic~=0) |
(get(handles.Segmentation,'value')==3) & (~isempty(fc))

switch get(handles.Segmentation,'Value')
    case 1
        %Do nothing
        msgbox('No segmentation method was selected','Preprocess','warn');

```

```

case 2
    %Simple Otsu Thresholding
    %=====
    fout=Otsu_thresholding_gui(fs);
    seq={fout ['Απλή κατωφλίωση Otsu']};
    ic=1;
    imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');

case 3
    %Color Based K-means Clustering
    %=====
    %Make Next, Previous buttons visible
    set(handles.previous, 'visible', 'on');
    set(handles.next, 'visible', 'on');

    a=Inputdlg('Give number of colors', 'nColors', 1, {'2'});
    nc=str2num(a{1});

    [p, s, cell]=color_kmeans_gui(fc, nc);
    %Αδειασμα μετρητή και δοχείο φωτωγραφιών για επόμενη επεξεργασία
    ic=0;
    seq={};

    seq{1}=p;
    seq{2}=['ΕΙΚΟΝΑ ΜΕ ΕΤΙΚΕΤΕΣ ΧΡΩΜΑΤΩΝ'];
    l=0;
    for k=3:2:(length(s)+nc+1)
        l=l+1;
        seq{k}=s{l};
        seq{k+1}=['ΑΝΤΙΚΕΙΜΕΝΑ ΣΤΟ CLUSTER ', num2str(l)];
    end
    seq{k+2}=cell;
    seq{k+3}=['ΕΙΚΟΝΑ ΚΥΤΤΑΡΩΝ ΠΟΥ ΑΝΙΧΝΕΥΘΗΚΑΝ'];

    % length(seq)
    ic=1;
    imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');

case 4
    %Morphology Segmentation 1
    %=====

    %Make Next, Previous buttons visible
    set(handles.previous, 'visible', 'on');
    set(handles.next, 'visible', 'on');
    %color=[0 0 1];

    color(1)=get(handles.RED, 'value');
    color(2)=get(handles.GREEN, 'value');
    color(3)=get(handles.BLUE, 'value');

    [gS, gP, gR, gL, fout]=morphology1_gui(fs, color);
    %Αδειασμα μετρητή και 'δοχείο' φωτωγραφιών για επόμενη επεξεργασία
    ic=0;

```

```

seq={};

seq={gS ['Sobel'] gP ['Prewitt'] gR ['Roberts'] gL ['LoG'] fout
['Τελική μορφολογική κατάτμηση 1']};
ic=1;
imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');
case 5
%Morphology Segmentation 2
%=====
set(handles.previous, 'visible', 'on');
set(handles.next, 'visible', 'on');

color(1)=get(handles.RED, 'value');
color(2)=get(handles.GREEN, 'value');
color(3)=get(handles.BLUE, 'value');

[fb, fo, f2, f3, fcb, fout]=morphology2_gui(fs, color)
%Αδειασμα μετρητή και δοχείο φωτωγραφιών για επόμενη επεξεργασία
ic=0;
seq={};

seq={fb, ['Διαδική εικόνα σε μαύρο φόντο'], fo, ['Διαδική εικόνα
μετά από opening'], f2, ['Εικόνα μετά από dilation γραμμής οριζόντια &
κάθεται'], f3, ['Γέμισμα οπών'], fcb, ['Αφαίρεση κυττάρων που ακουμπούν στα
άκρα'], fout, ['Τελική μορφολογική Κατάτμηση 2']};
ic=1;
imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');

case 6
%Watershed distance segmentation
%=====
set(handles.previous, 'visible', 'on');
set(handles.next, 'visible', 'on');

color(1)=get(handles.RED, 'value');
color(2)=get(handles.GREEN, 'value');
color(3)=get(handles.BLUE, 'value');

[fbw, D, D2, L, k, fout]=watershed_distance_gui(fs, color)
%Αδειασμα μετρητή και δοχείο φωτωγραφιών για επόμενη επεξεργασία
ic=0;
seq={};

seq={fbw, ['Ασπρόμαυρη εικόνα'], D, ['Μετασχηματισμός
Απόστασης'], D2, ['Συμπληρωματική του μετ/σμου Απόστασης'], L, ['Watershed
segmentation αντιστρ. μετ/σμου απόστασης'], k, ['Watershed ridge lines του
συμπλ. μετ/σμου απόστασης'], fout, ['Τελικό κατάτμηση watershed με μετ/σμο
απόστασης']};
ic=1;
imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');

case 7
%Watershed gradient segmentation
%=====
set(handles.previous, 'visible', 'on');
set(handles.next, 'visible', 'on');

color(1)=get(handles.RED, 'value');
color(2)=get(handles.GREEN, 'value');

```

```

color(3)=get(handles.BLUE,'value');

[grad,L,k,fout]=watershed_gradient_gui(fs,color)
%Αδειασμα μετρητή και δοχείο φωτωγραφιών για επόμενη επεξεργασία
ic=0;
seq={};

seq={grad,['H gradient της grayscale εικόνας'],L,['Watershed της
Gradient'],k,['Watershed ridge lines της Gradient'],fout,['Τελική
Κατάτμηση']};

ic=1;
imshow(seq{ic},[]);title(seq{ic+1},'color','y');

case 8
%Watershed marker-controlled
%=====
set(handles.previous,'visible','on');
set(handles.next,'visible','on');

color(1)=get(handles.RED,'value');
color(2)=get(handles.GREEN,'value');
color(3)=get(handles.BLUE,'value');

b=Inputdlg('Give the h of extended minima','extended
minima',1,{ '5' });
h=str2num(b{1});
[rm,fim,em,grad2,fout]=watershed_markers_gui(fs,h,color)
%Αδειασμα μετρητή και δοχείο φωτωγραφιών για επόμενη επεξεργασία
ic=0;
seq={};

seq={rm,['Εικόνα τοπικών ελαχίστων της Gradient'],fim,['Εσωτερικοί
Markers'],em,['Εξωτερικοί Markers'],grad2,['Νέα Gradient με
Markers'],fout,['Αποτέλεσμα κατάτμησης']};

ic=1;
imshow(seq{ic},[]);title(seq{ic+1},'color','y');

case 9
%Distance only segmentation
%=====
set(handles.previous,'visible','on');
set(handles.next,'visible','on');

color(1)=get(handles.RED,'value');
color(2)=get(handles.GREEN,'value');
color(3)=get(handles.BLUE,'value');

[bw,bw2,DT,nuclei,fout]=Distance_only_gui(fs,color);
%Αδειασμα μετρητή και δοχείο φωτωγραφιών για επόμενη επεξεργασία
ic=0;
seq={};

```

```

seq={bw, ['Εικόνα b/w σε λευκό φόντο'],bw2, ['Εικόνα b./w σε Μαύρο
φόντο'],DT, ['Εικόνα μετ/σμού απόστασης(στην b/w μαυρου
φόντου)'],nuclei, ['Εικόνα ορίων πυρήνων'],fout, ['Τελικό segmentation χωρίς
πυρήνες που ακουμπούν στα άκρα']};
ic=1;
imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');

case 10
%Feature extraction using Distance Transform(Η μέθοδος που
%λειτουργήσει καλύτερα)
%=====
set(handles.previous, 'visible', 'on');
set(handles.next, 'visible', 'on');

color(1)=get(handles.RED, 'value');
color(2)=get(handles.GREEN, 'value');
color(3)=get(handles.BLUE, 'value');

[DT,nuclei,w,Af,Rf,areahist,roundhist]=Final_characteristics_gui(fs,color)
;

%Aδειασμα μετρητή και δοχείο φωτογραφιών για επόμενη επεξεργασία
ic=0;
seq={};

seq={w, ['Τελική
κατάτμηση'],Af, ['EMBAΔΑ'],Rf, ['ΚΥΚΛΙΚΟΤΗΤΕΣ'],areahist, ['Ιστόγραμμα
Εμβαδών (pixels)'],roundhist, ['Ιστόγραμμα Κυκλικοτήτων']};

ic=1;
imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');

end

elseif isempty(fc)==1
msgbox('Input a color image and then preprocess','Image input
','warn');
else
msgbox('Preprocess first image','Image input','warn');
end

% --- Executes during object creation, after setting all properties.
function Segmentation_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Segmentation (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```



```
% --- Executes on button press in Reset.
function Reset_Callback(hObject, eventdata, handles)
% hObject    handle to Reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

cla(handles.init, 'reset')
cla(handles.fin, 'reset')
set(handles.init, 'visible', 'off');
set(handles.fin, 'visible', 'off');

set(handles.previous, 'visible', 'off');
set(handles.next, 'visible', 'off');

set(handles.initial, 'visible', 'off');
set(handles.final, 'visible', 'off');

clear all;

% --- Executes during object creation, after setting all properties.
function fin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate fin

% --- Executes on button press in next.
function next_Callback(hObject, eventdata, handles)
% hObject    handle to next (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global seq ic;

if ic<length(seq)-1
ic=ic+2;
imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');
else
    msgbox('End of sequence', 'warn', 'modal');
end
```

```

% --- Executes on button press in previous.
function previous_Callback(hObject, eventdata, handles)
% hObject    handle to previous (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global seq ic
if ic>=2
ic=ic-2;
imshow(seq{ic}, []);title(seq{ic+1}, 'color', 'y');
else
msgbox('Begin of sequence', 'warn', 'modal');
end

% --- Executes on slider movement.
function RED_Callback(hObject, eventdata, handles)
% hObject    handle to RED (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%        get(hObject, 'Min') and get(hObject, 'Max') to determine range of
slider

% --- Executes during object creation, after setting all properties.
function RED_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RED (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

% --- Executes on slider movement.
function GREEN_Callback(hObject, eventdata, handles)
% hObject    handle to GREEN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%        get(hObject, 'Min') and get(hObject, 'Max') to determine range of
slider

% --- Executes during object creation, after setting all properties.
function GREEN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GREEN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function BLUE_Callback(hObject, eventdata, handles)
% hObject      handle to BLUE (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of
slider

% --- Executes during object creation, after setting all properties.
function BLUE_CreateFcn(hObject, eventdata, handles)
% hObject      handle to BLUE (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in save.
function save_Callback(hObject, eventdata, handles)
% hObject      handle to save (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global seq ic;
pathname='';
if (~isempty(seq{ic}))
    [filename,pathname]=uiputfile([pathname '*.jpg'],'Save Image');
    if (filename (~0))
        imwrite(seq{ic},[pathname,filename],'jpg');
    end
end

% --- Executes on button press in print.
function print_Callback(hObject, eventdata, handles)
% hObject      handle to print (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global seq ic
figure('visible','off');
imshow(seq{ic},[]);
printpreview;

% --- Executes on button press in exit.
function exit_Callback(hObject, eventdata, handles)
% hObject      handle to exit (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close all;
clear all;

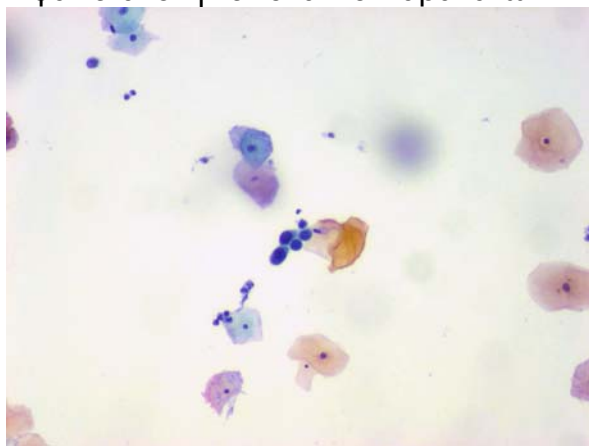
```

Γ. Κλινική εφαρμογή σε εικόνες που ελήφθησαν από το κυτταρολογικό εργαστήριο του Πανεπιστημιακού Γενικού Νοσοκομείου "ΑΤΤΙΚΟΝ"

Οι αλγόριθμοι που αναπτύχθηκαν στο κεφάλαιο 5 εφαρμόστηκαν σε 2 εικόνες του κυτταρολογικού εργαστηρίου του πανεπιστημιακού Νοσοκομείου "ΑΤΤΙΚΟΝ". [16]
Τα αποτελέσματα φαίνονται παρακάτω:

i) Εικόνα με κύτταρα της παραβασικής στοιβάδας με ευμεγεθείς υπερχρωματικούς πυρήνες (x10)

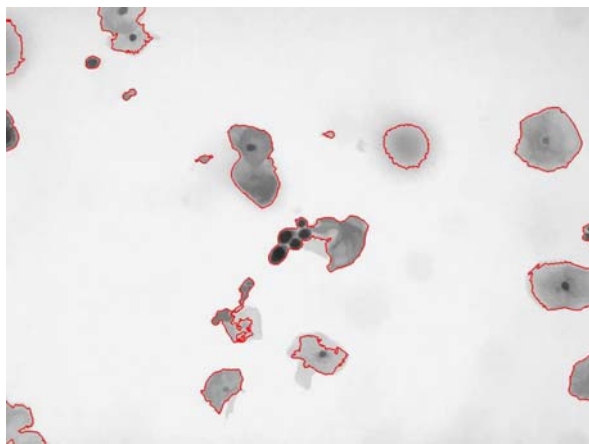
Το αποτέλεσμα της κατάτμησης με χρήση των αλγορίθμων του κεφαλαίου 5 φαίνεται στην εικόνα Π3 παρακάτω:



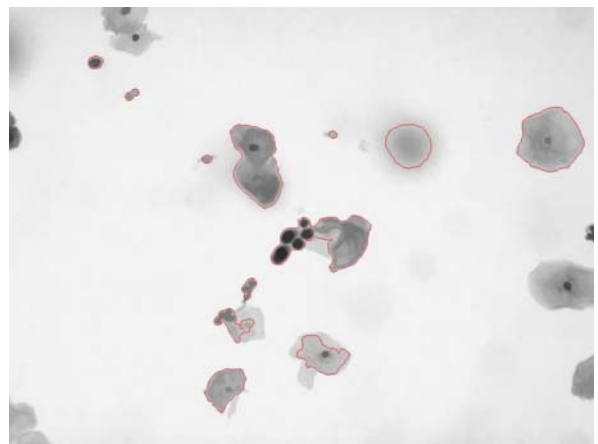
(α)



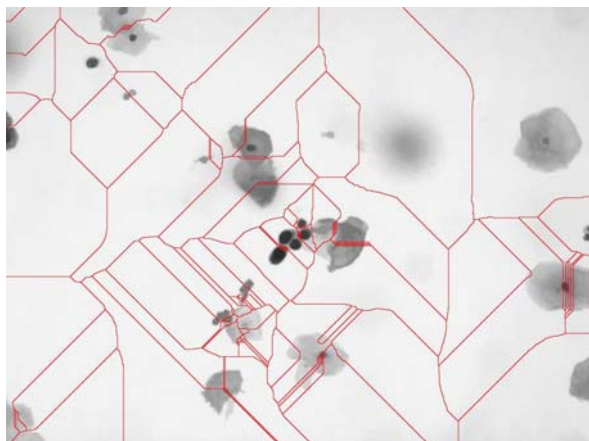
(β)



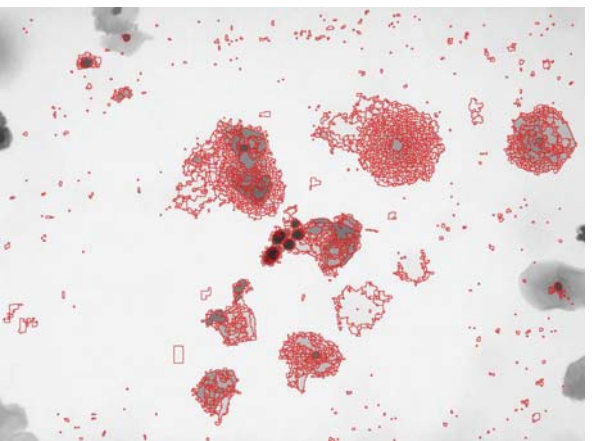
(γ)



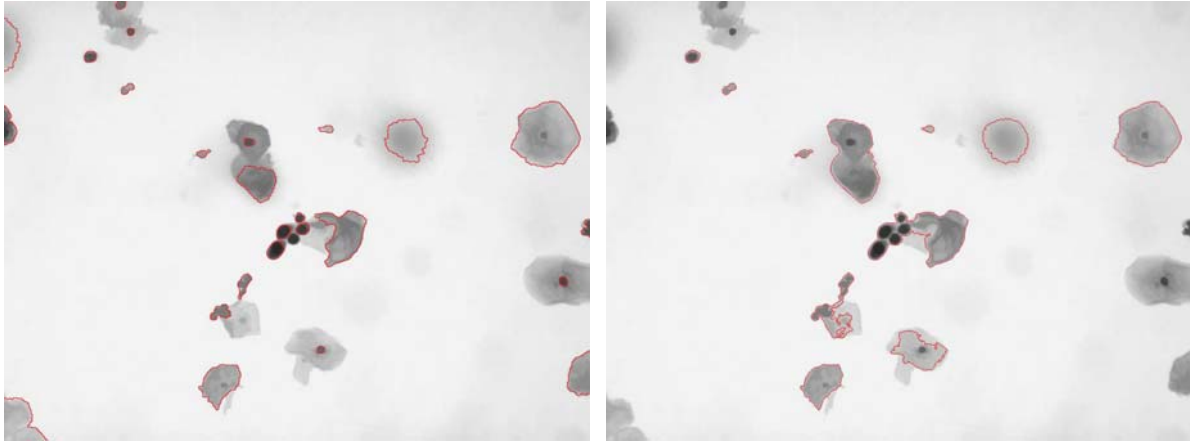
(δ)



(ε)



(στ)



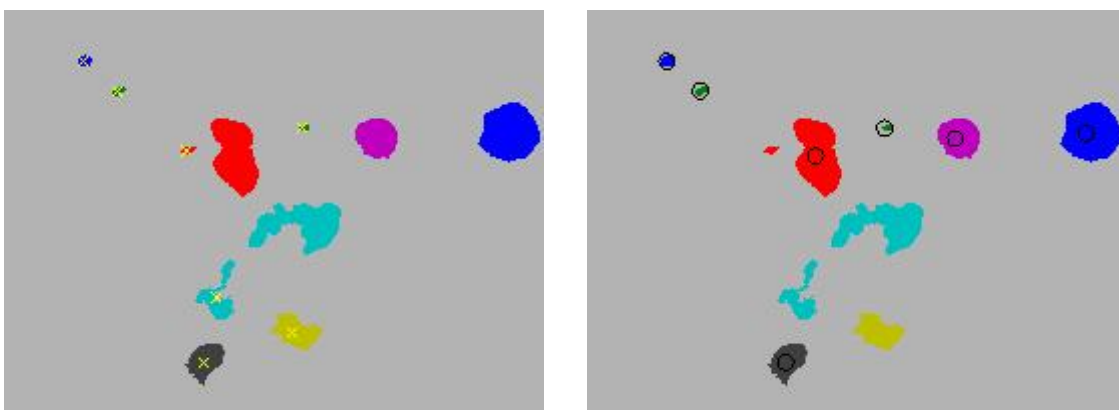
(ζ)

(η)

ΕΙΚΟΝΑ Π3 Εφαρμογή αλγορίθμων σε εικόνα από τεστ παπ

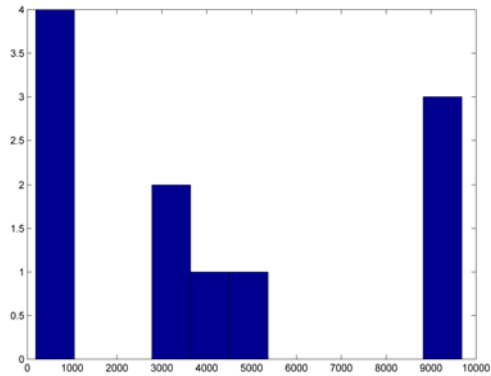
- (α) Αρχική κυτταρική εικόνα τεστ παπ
- (β) Απλή κατωφλίωση Otsu (§5.2)
- (γ) Κατάτμηση με φίλτρο ενδιάμεσης τιμής και μορφολογικούς τελεστές (§5.4.1)
- (δ) Κατάτμηση με μορφολογικούς τελεστές (§5.4.2)
- (ε) Κατάτμηση με watershed μετασχηματισμού απόστασης (§5.5.1)
- (στ) Κατάτμηση με watershed και gradients (§5.5.2)
- (ζ) Κατάτμηση με marker-controlled watershed με $h=25$ (§5.5.3)
- (η) Κατάτμηση μόνο με μετασχηματισμό απόστασης (§5.6)

Τα αποτελέσματα της ανάλυσης της κατάτμησης μετασχηματισμού απόστασης φαίνονται στις παρακάτω εικόνα Π4:

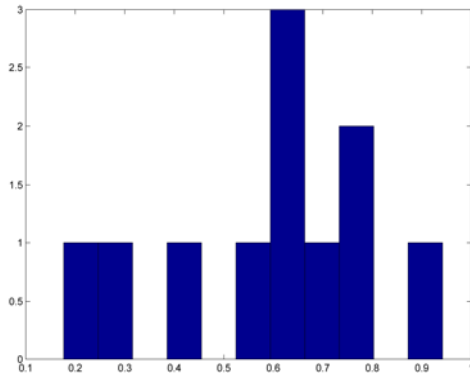


(α)

(β)



(γ)



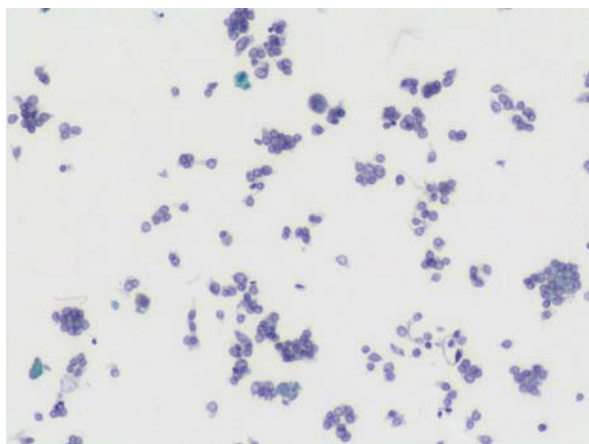
(δ)

ΕΙΚΟΝΑ Π4 Ανάλυση κατάτμησης μετασχηματισμού απόστασης (§5.6)

- (α) Εικόνα με επιλεγμένα τα κύτταρα με εμβαδόν μικρότερο του μέσου όρου
- (β) Εικόνα με επιλεγμένα τα κύτταρα με κυκλικότητα μεγαλύτερη του μέσου όρου
- (γ) Ιστόγραμμα εμβαδών κυττάρων που ανιχνεύτηκαν
- (δ) Ιστόγραμμα κυκλικοτήτων κυτταρων που ανιχνεύτηκαν

ii) Εικόνα κυτταροβριθές επιχρίσματος αποτελούμενου από μονοεπίπεδες ομάδες(x10)

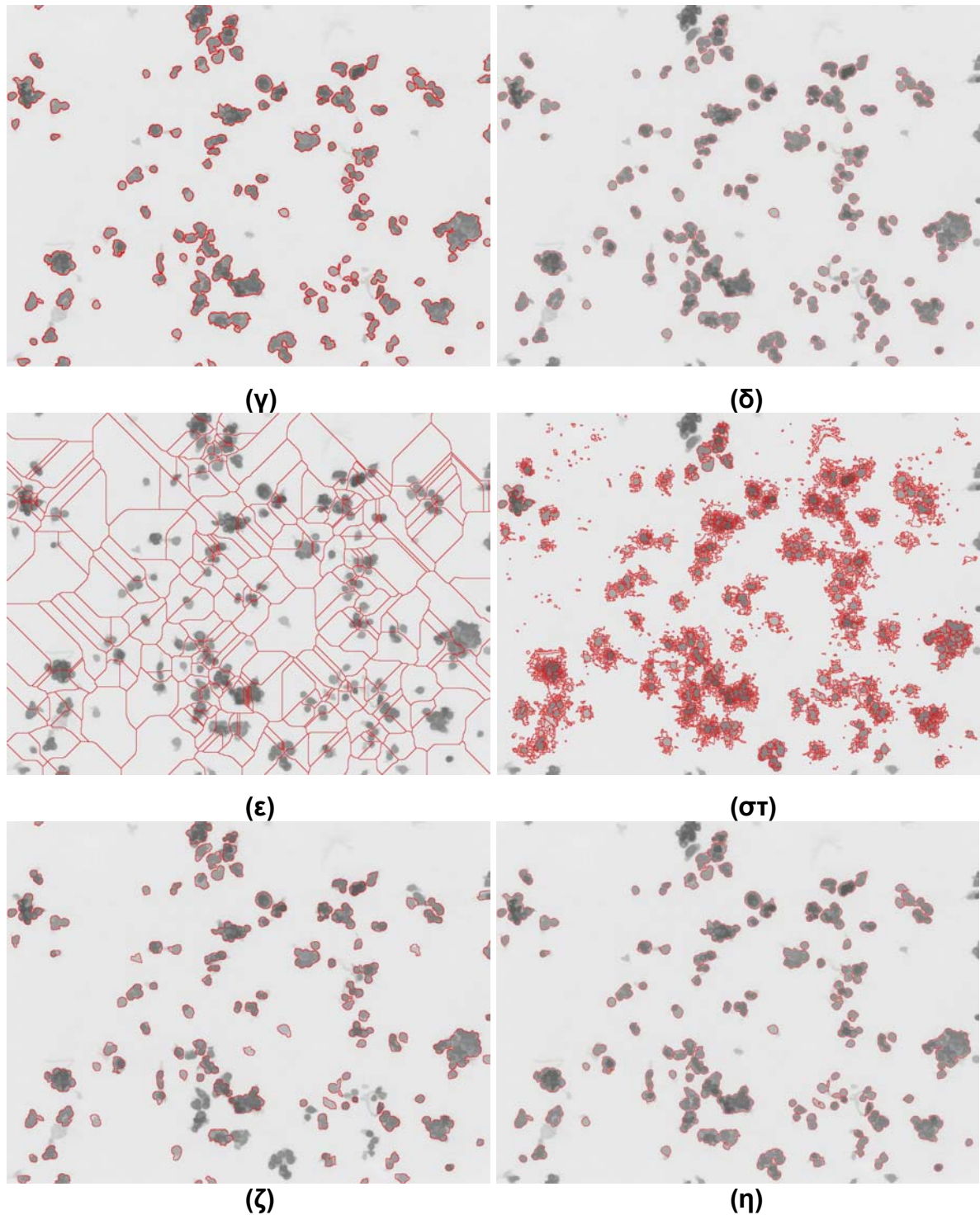
Το αποτέλεσμα της κατάτμησης με χρήση των αλγορίθμων του κεφαλαίου 5 φαίνεται στην εικόνα Π5 παρακάτω:



(α)



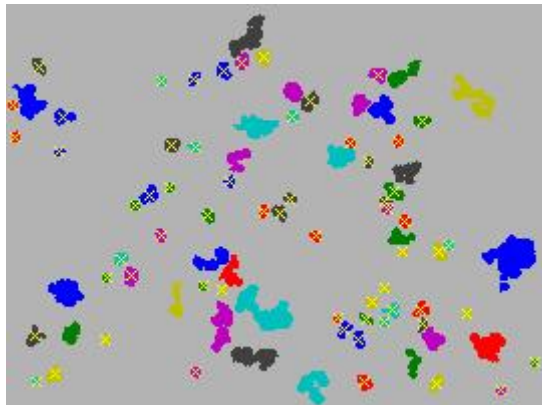
(β)



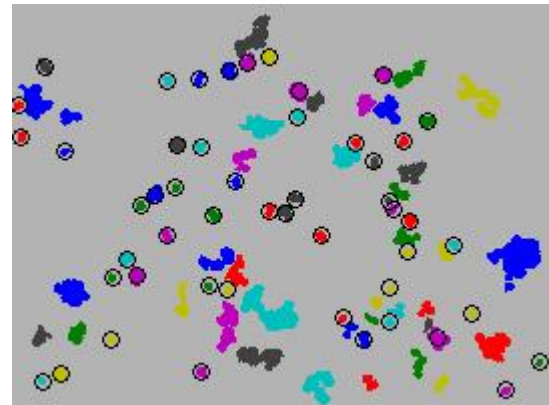
ΕΙΚΟΝΑ Π5 Εφαρμογή αλγορίθμων σε εικόνα θυρεοειδούς

- (α) Αρχική κυτταρική εικόνα θυρεοειδούς
- (β) Απλή κατωφλίωση Otsu (§5.2)
- (γ) Κατάτμηση με φίλτρο ενδιάμεσης τιμής και μορφολογικούς τελεστές (§5.4.1)
- (δ) Κατάτμηση με μορφολογικούς τελεστές (§5.4.2)
- (ε) Κατάτμηση με watershed μετασχηματισμού απόστασης (§5.5.1)
- (στ) Κατάτμηση με watershed και gradients (§5.5.2)
- (ζ) Κατάτμηση με marker-controlled watershed με $h=25$ (§5.5.3)
- (η) Κατάτμηση μόνο με μετασχηματισμό απόστασης (§5.6)

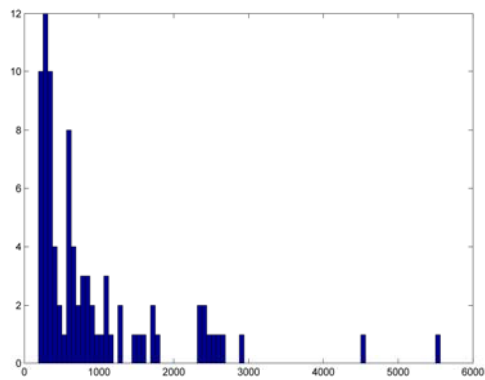
Τα αποτελέσματα της ανάλυσης της κατάτμησης μετασχηματισμού απόστασης φαίνονται στις παρακάτω εικόνα Π6:



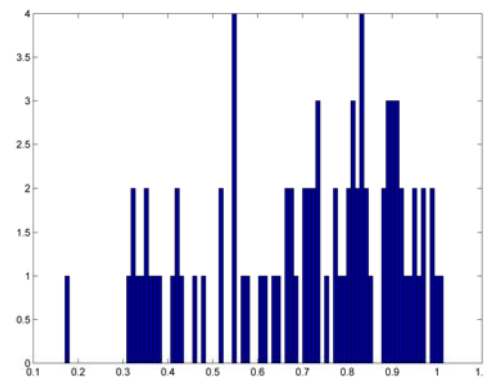
(α)



(β)



(γ)



(δ)

ΕΙΚΟΝΑ Π6 Ανάλυση κατάτμησης μετασχηματισμού απόστασης (§5.6)

- (α) Εικόνα με επιλεγμένα τα κύτταρα με εμβαδόν μικρότερο του μέσου όρου
- (β) Εικόνα με επιλεγμένα τα κύτταρα με κυκλικότητα μεγαλύτερη του μέσου όρου
- (γ) Ιστόγραμμα εμβαδών κυττάρων που ανιχνεύτηκαν
- (δ) Ιστόγραμμα κυκλικοτήτων κυτταρων που ανιχνεύτηκαν

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΛΛΗΝΟΓΛΩΣΣΗ

- [1] Αποστόλου Ν. (2005) *Μελέτη, σχεδίαση και ανάπτυξη εξελιγμένων αλγορίθμων για ψηφιακή ανάλυση και επεξεργασία ιατρικών εικόνων σε ενοποιημένη πλατφόρμα* Phd Thesis ΕΜΠ
- [2] Κόλλιας Σ. Δ. (2001) *Επεξεργασία, ανάλυση και τεχνολογία εικόνων Βίντεο*, Αθήνα: Σημειώσεις ΕΜΠ
- [3] Κουτσούρης Δ. *Επεξεργασία εικόνων DICOM με τη χρήση Matlab*, Αθήνα: Εργαστηριακή άσκηση ΕΜΠ
- [4] Μαραγκός Π.(2002) *Όραση Υπολογιστών*, Αθήνα: Σημειώσεις ΕΜΠ
- [5] Νικήτα Κ. *Στοιχεία επεξεργασίας ιατρικών εικόνων*, Αθήνα: Σημειώσεις
- [6] Παπασταματόπουλος Μ. (2005) *Μελέτη Σύγχρονων Τεχνικών Επεξεργασίας και Ανάλυσης Οφθαλμιατρικών εικόνων και εικόνων Video- Βρογχοσκοπίου Επεμβατικής Πνευμονολογίας με ιδιαίτερο κλινικό ενδιαφέρον*, Μεταπτυχιακή εργασία ΕΜΠ, Αθήνα(Πανεπιστήμιο Πατρών)

ΞΕΝΟΓΛΩΣΣΗ

- [7] Anoraganingrum D.(1999) "Cell segmentation with median filter and Mathematical Morphology Operation", Proceedings, International Conference on *Image Analysis and Processing*, pp. 1043-1046
- [8] Gonzalez R. C., Woods R. E. (2002) *Digital Image Processing*, Second edition, Prentice Hall, Upper Sandle River, NJ
- [9] Gonzalez R. C., Woods R. E., Eddins S. L.(2004), *Digital Image Processing Using MATLAB*, Prentice Hall, Upper Sandle River, NJ
- [10] Hernández L, Gothreaux P, Shih L. and G. Campbell (2006) "Toward Real-Time Biopsy Image Analysis and Cell Segmentation", in Proceedings. *IPCV*, pp.81-87.
- [11] McAndrew A. (2004), *An Introduction to Digital Image Processing with MATLAB*, Notes for SCM2511 Image Processing 1 Semester 1, Victoria University of Technology

- [12] Morris T., *Image Processing with MATLAB*, Supporting Material for COMP20072, Manchester
- [13] Otsu N. (1979), *A threshold Selection Method from Gray-Level Histograms*, IEEE Transactions for Systems, Man and Cybernetics, vol. SMC-9, No 1, January, pp. 62-66
- [14] Miroslaw L. (2006), *Multimodal image processing in Cytology*, PhD dissertation, Wroclaw University of Technology
- [15] Van Vliet N. (2003), *Image Segmentation Applied to Cytology*, Technical report, France

ΔΙΑΔΙΚΤΥΟ

- [16] Περιβάλλον Σύγχρονης Τηλεκπαίδευσης του Εργαστηρίου Διαγνωστικής Κυτταρολογίας του Πανεπιστημιακού Γενικού Νοσοκομείου “ΑΤΤΙΚΟΝ”
<http://cytodept.static.otenet.gr:8080/CytoTrainer/LoginServlet>
(Απαιτείται πρώτα εγγραφή στο σύστημα)
Ημερομηνία τελευταίας προσπέλασης: 14-10-2011
- [17] Fisher R., Perkins S., Walker A. and Wolfart E.(2003), *Distance Transform*
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/distance.htm>
Ημερομηνία τελευταίας προσπέλασης: 05-10-2011
- [18] Prolla J. C.- Atlas of cytopathology and cytology,
http://www.geocities.com/jcprolla/Cytopathology_Atlas.html
Ημερομηνία τελευταίας προσπέλασης: 15-09-2011
- [19] Mathworks, Examples and web seminars: *Color Based Segmentation Using K-means Clustering*
<http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexhistology.html>
Ημερομηνία τελευταίας προσπέλασης: 20-05-2011
- [20] Mathworks, Examples and web seminars: *Identifying Round Objects*

http://www.mathworks.com/products/demos/image/round_objects/ipexroundness.html

Ημερομηνία τελευταίας προσπέλασης: 05-10-2011

[21] Mathworks, documentation: Function *Regionprops*

<http://www.mathworks.com/help/toolbox/images/ref/regionprops.html>

Ημερομηνία τελευταίας προσπέλασης: 05-10-2011

[22] Microscopes wikipedia:

http://en.wikipedia.org/wiki/Microscope#Fluorescence_and_light_microscopy

Ημερομηνία τελευταίας προσπέλασης: 20-07-2011