Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πτυχιακή εργασία

Τίτλος

Ευφυή συστήματα διαχείρισης μεταφορών/συγκοινωνιών με βάση την εικόνα

Τσουρουπάκης Εμμανουήλ(ΑΜ : 568)

Επιβλέπων καθηγητής : Τριανταφυλλίδης Γεώργιος

Abstract

In this work we used Matlab to show how we can work with digital images. First import the image in Matlab to get information on the pixels of the image, measuring the characteristics, finding endpoint and other information.

Also we can process the image we get introduced to conclusions about it depending on what we learn. That is, adjusting brightness contrast, view multiple images, multiple image frames, image resizing, cropping the image so we can work out a part of the image and filtering image modification or enhancement.

In addition in Matlab we can identify and correct morphological operations like dilation and erosion. The Matlab is very useful because it gives us more information about pixels, statistics and histogram. We can also analyze the texture of the image especially for topographic-morphological data. A process of technical assistance is to change frequency. These techniques are used to remove noise from the image to become clear and make processes running neighborhood of pixels.

By exploiting the capabilities of Matlab described earlier in this work we have implemented some applications, image segmentation based on characteristics. More on these applications involve the identification of objects in an image based on image characteristics and objects. It is recognition of objects in drab background, <<darkness>> image, object recognition by size, based on the curvature, based on the help of transformation Watershed, through video.

It has also been implemented and an application GUI (Graphical User Interface), which has the possibility through a video to identify the cars there and make a calculation of the movement of cars. All these applications are explained in more detail in the following chapters.

In conclusion we can draw the conclusion that as Matlab is a very useful tool for digital image processing and analysis set out above, which can be used for many purposes and in all fields of science.

Σύνοψη

Σ΄ αυτήν την εργασία χρησιμοποιήσαμε το Matlab ώστε να δείξουμε πως μπορούμε να επεξεργαστούμε τις ψηφιακές εικόνες. Αρχικά εισάγουμε την εικόνα στο Matlab ώστε να πάρουμε πληροφορίες σχετικά με τα pixels της εικόνας, την μέτρηση των χαρακτηριστικών, την εύρεση τελικού σημείου και άλλες πληροφορίες.

Ακόμα μπορούμε να επεξεργαστούμε την εικόνα που έχουμε εισαγάγει ώστε να πάρουμε συμπεράσματα γι' αυτήν ανάλογα με το τι θέλουμε να μάθουμε. Δηλαδή, προσαρμογή αντίθεσης φωτεινότητας, προβολή πολλαπλών εικόνων, πολλαπλά καρέ εικόνας, αλλαγή μεγέθους εικόνας, περικοπή εικόνας ώστε να μπορούμε να επεξεργαστούμε ένα μέρος της εικόνας και φιλτράρισμα εικόνας για τροποποίηση ή ενίσχυση.

Επιπλέον στο Matlab μπορούμε να εντοπίσουμε και να διορθώσουμε μορφολογικές πράξεις όπως διαστολή και διάβρωση. Το matlab είναι πολύ χρήσιμο ακόμα γιατί μας δίνει πληροφορίες για pixels, στατιστικά στοιχεία και το ιστόγραμμα. Μπορούμε ακόμα να αναλύσουμε την υφή της εικόνας ειδικά όταν πρόκειται για μορφολογικά-τοπογραφικά στοιχεία. Μία διεργασία τεχνικής ενίσχυσης είναι η τροποποίηση συχνότητας. Αυτές οι τεχνικές χρησιμεύουν για να αφαιρέσουμε θόρυβο από την εικόνα, να την ξεθολώσουμε και να κάνουμε διεργασίες κύλισης γειτονιάς των pixels.

Εκμεταλλευόμενοι τις ικανότητες του Matlab που περιγράψαμε παραπάνω σε αυτήν την εργασία έχουμε υλοποιήσει κάποιες εφαρμογές τμηματοποίησης εικόνας με βάση τα χαρακτηριστικά τους. Αναλυτικότερα στις εφαρμογές αυτές γίνεται αναγνώριση των αντικειμένων σε μια εικόνα με βάση τα χαρακτηριστικά της εικόνας και των αντικειμένων. Συγκεκριμένα γίνεται αναγνώριση αντικειμένων σε μονότονο background, <<σκοτεινιάζοντας>> την εικόνα, αναγνώριση αντικειμένων με βάση το μέγεθος τους, με βάση την κυρτότητα τους, με βάση την βοήθεια του μετασχηματισμού Watershed, μέσα από βίντεο.

Επίσης έχει υλοποιηθεί και μια εφαρμογή GUI(Graphical User Interface), η οποία έχει την δυνατότητα μέσα από ένα βίντεο να εντοπίζει τα αυτοκίνητα που υπάρχουν και να κάνει έναν υπολογισμό της κίνησης των αυτοκινήτων. Όλες αυτές οι εφαρμογές επεξηγούνται εκτενέστερα στα παρακάτω κεφάλαια.

Εν κατακλείδι μπορούμε να συμπεράνουμε ότι το Matlab είναι πολύ χρήσιμο εργαλείο για την ψηφιακή ανάλυση και επεξεργασία εικόνας σύμφωνα με τα προαναφερθέντα, κι επίσης μπορεί να χρησιμοποιηθεί για πάρα πολλούς σκοπούς και σε όλους τους τομείς της επιστήμης.

Πίνακας Περιεχομένων

Κεφάλαιο 1

Εισαγωγή

Όπως προαναφέραμε στην σύνοψη σε αυτήν την εργασία έχουμε υλοποιήσει κάποιες εφαρμογές σε Matlab χωριζοντάς τες σε δύο κατηγορίες. Η πρώτη κατηγορία αναφέρεται στην ψηφιακή επεξεργασία στατικών εικόνων και η δεύτερη κατηγορία αναφέρεται στα συστήματα παρακολούθησης κυκλοφορίας. Για να γίνει πιο κατανοητή η χρησιμότητα ύπαρξης τέτοιων συστημάτων πρέπει να αναφερθούν κάποια στοιχεία, όπως η εξέλιξη τους με το πέρασμα των χρόνων, το τι πρέπει να κάνουμε για να στηθούν τέτοια συστήματα, ποιες οι δυνατότητες που θα μας παρέχουν κ.α.

1.1 Ψηφιακή επεξεργασία στατικών εικόνων

Τα τελευταία χρόνια και κυρίως μετά την ανάπτυξη της αγοράς των ψηφιακών μηχανών ακούγεται όλο και περισσότερο η έννοια της ψηφιακής επεξεργασίας εικόνας και φωτογραφίας. Ως ψηφιακή επεξεργασία εικόνας ορίζεται η χρήση (μέσω του υπολογιστή) διαφόρων αλγορίθμων με σκοπό την επεξεργασία ψηφιακών εικόνων και φωτογραφιών.

Η ψηφιακή επεξεργασία εικόνας (ΨΕΕ) αποτελεί έναν ευρύ επιστημονικό κλάδο που αναπτύχθηκε με την ραγδαία εξέλιξη των υπολογιστών. Ο όρος εικόνα χρησιμοποιείται ευρύτερα από την απλή απεικόνιση ενός σκηνικού έως την αποτύπωση κάθε είδους πληροφοριών.

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

Μερικά ερωτήματα που γεννιούνται είναι το ποιες είναι οι διεργασίες που εκτελούνται κατά την ψηφιακή επεξεργασία εικόνας, που χρησιμεύει η ψηφιακή επεξεργασία εικόνας, ποιες εφαρμογές έχει στην ζωή μας και πόσο μπορεί να μας βοηθήσει σε διάφορους τομείς της επιστήμης.

Μπορούμε λοιπόν να πούμε ότι οι διεργασίες μιας ψηφιακής επεξεργασίας εικόνας είναι:

- Καταγράφει εικόνες(μία προς μία ή και ακολουθία).
- Επεξεργάζεται εικόνες με ψηφιακό τρόπο.
- Εξάγει νέες εικόνες

Όσον αφορά την χρησιμότητα της ψηφιακής επεξεργασίας εικόνας μπορούμε να πούμε ότι μας βοηθάει για την:

- Ευκολότερη αποθήκευση και μετάδοση εικόνων σε ψηφιακή μηχανή, στο PC
- Μετάδοση εικόνων από δορυφόρους στη Γη
- Γίνεται καλή συμπίεση, πιο αποτελεσματική
- Χρήσιμες εικόνες διαθέσιμες προς μετάδοση
- Βελτίωση/αποκατάσταση ποιότητας

Μερικές εφαρμογές που έχει η ψηφιακή επεξεργασία εικόνας στην ζωή μας είναι:

- Ιατρικές εικόνες
- Δορυφορικές εικόνες, surveillance
- Ασφάλεια(κάμερες στους δρόμους)
- Έλεγχος για όγκο σε ακτινογραφία
- Ανάλυση καιρικών φαινομένων
- κίνηση σε αεροδρόμια,δρόμους
- Βελτίωση ποιότητας και αλλαγή εμφάνισης εικόνας γίνεται τώρα από τον καθένα

- Αυτόματη αναγνώριση χαρακτήρων(τράπεζες, αρχαιολογία, ασφάλεια, ταχ.κώδικες)
- Αυτόματη αναγνώριση δακτυλικών αποτυπωμάτων
- Αυτόματη αναγνώριση προσώπων, εκφράσεων

1.2 Συστήματα παρακολούθησης κυκλοφορίας

Η κατακόρυφη αύξηση των κυκλοφοριακών φόρτων που λαμβάνει χώρα τα τελευταία χρόνια σε όλα τα οδικά δίκτυα παγκοσμίως έχει επιβάλλει νέες απαιτήσεις στο χώρο της διαχείρισης των εν λειτουργία οδικών έργων. Η φιλοσοφία της διεύρυνσης της οδικής υποδομής ως λύση στο πρόβλημα έδειξε γρήγορα τη δυσκαμψία της, οπότε ως γόνιμη στρατηγική αντιμετώπισης έχει χριστεί πλέον η φιλοσοφία της διαχείρισης της κυκλοφορίας. Για να πραγματοποιηθεί, όμως, η διαχείριση αυτή, ασφαλώς απαιτείται πρώτα η απόκτηση των σχετικών δεδομένων της κυκλοφορίας, μέσα από την ίδια την οδική υποδομή. Παράλληλα, οι απαιτήσεις ασφάλειας που προβάλλουν οι σύγχρονοι αυτοκινητόδρομοι επιβάλλουν ταχύτερη ανίχνευση και αντιμετώπιση των πάσης φύσεως προβληματικών καταστάσεων στην κυκλοφορία, που φτάνουν ως την άμεση επιτήρηση σε πραγματικό χρόνο.

Οι συνθήκες αυτές έχουν οδηγήσει στην ανάπτυξη ενός ακόμη πεδίου εξοπλισμού των οδών, αυτό του εξοπλισμού παρακολούθησης της κυκλοφορίας. Μέχρι τώρα στο πεδίο αυτό περιλαμβάνονταν μόνο οι ανιχνευτές για τους σκοπούς της φωτεινής σηματοδότησης, καθώς και κάμερες κλειστού τηλεοπτικού κυκλώματος σε επικίνδυνα σημεία, όπως σήραγγες ή γέφυρες. Η σύγχρονη παρακολούθηση αξιοποιεί τόσο τις υπάρχουσες, όσο και νέες τεχνολογίες για τους σκοπούς της, δηλαδή τη μέτρηση πάσης φύσεως κυκλοφοριακών δεδομένων και την επιτήρηση της κυκλοφορίας.

Τύποι συστημάτων παρακολούθησης της κυκλοφορίας

Ο εξοπλισμός των συστημάτων παρακολούθησης της κυκλοφορίας μπορεί να κατηγοριοποιηθεί στους παρακάτω τύπους:

- **Μαγνητικοί ανιχνευτές:** Είναι παγκοσμίως ο ευρύτερα χρησιμοποιούμενος τύπος, με κύρια εφαρμογή τους σηματοδοτημένους κόμβους. Χρησιμοποιούνται για μέτρηση κυκλοφοριακών δεδομένων.
- **Αισθητήρες ανίχνευσης έξω από το οδόστρωμα:** Είναι συσκευές που τοποθετούνται επάνω από το οδόστρωμα, εφαρμόζοντας διάφορες τεχνολογίες όπως μικροκύματα, λέιζερ ή υπέρυθρες ακτινοβολίες. Χρησιμοποιούνται και αυτοί για μέτρηση κυκλοφοριακών δεδομένων.
- **Κλειστά αναλογικά κυκλώματα τηλεόρασης:** Η γνωστή πρακτική όπου αναλογικές κάμερες αποστέλλουν την εικόνα σε οθόνες του κέντρου διαχείρισης, για άμεση παρακολούθηση από το προσωπικό.
- **Ψηφιακή επεξεργασία εικόνας:** Η πιο σύγχρονη τεχνική, όπου ψηφιακές κάμερες αποστέλλουν την εικόνα σε υπολογιστικά συστήματα τόσο για εξαγωγή κυκλοφοριακών δεδομένων, όσο και για διαπίστωση ειδικών συμβάντων, σε κάθε περίπτωση μετά από ηλεκτρονική επεξεργασία της εικόνας.

Εν συνεχεία ας παρουσιάσουμε συνοπτικά τους παραπάνω τύπους παρακολούθησης κυκλοφορίας

Μαγνητικοί ανιχνευτές

Οι σπουδαιότεροι εκπρόσωποι αυτής της κατηγορίας είναι οι ανιχνευτές βρόχου. Η εφαρμογή των ανιχνευτών αυτών ήταν πολύ συνηθισμένη σε σηματοδοτημένους κόμβους για τις ανάγκες της πρόνοιας, αλλά πλέον έχει επεκταθεί και στο πεδίο της παρακολούθησης της κυκλοφορίας, και ειδικότερα για την απόκτηση κυκλοφοριακών δεδομένων.

Οι εν λόγω ανιχνευτές είναι συσκευές που αποτελούνται από ένα βρόχο καλωδίου, τοποθετημένου κάτω από την οδική επιφάνεια, ο οποίος διαρρέεται από ηλεκτρικό ρεύμα. Ένα

όχημα που διέρχεται επάνω από το καλώδιο λειτουργεί ως πυρήνας στο πηνίο, μεταβάλλοντας τη χωρητικότητά του, και με αυτό τον τρόπο πιστοποιεί την διέλευση του.

Εικόνα 1. Ζεύγη μαγνητικών ανιχνευτών βρόχου

Στοιχεία που είναι δυνατόν να μετρηθούν με τη βοήθεια των ανιχνευτών βρόχου είναι ο κυκλοφοριακός φόρτος, η κατανομή κατά λωρίδα, η πυκνότητα των οχημάτων και οι χρονικοί διαχωρισμοί. Επίσης, όπως φαίνεται στην Εικόνα 1, τοποθετώντας τους βρόχους σε ζεύγη κατά μήκος της λωρίδας είναι δυνατή και η μέτρηση της ταχύτητας, όπως και του μήκους κάθε οχήματος, επιτρέποντας την κατηγοριοποίηση της κυκλοφορίας κατά τύπο οχημάτων.

Επιπλέον, προχωρημένες τεχνικές διαχείρισης των εν λόγω συστημάτων ανίχνευσης χρησιμοποιούν την προαναφερθείσα ικανότητα μέτρησης του μήκους για ταυτοποίηση της διέλευσης του ίδιου οχήματος σε επόμενα σημεία, επιτρέποντας έτσι την εξαγωγή συμπερασμάτων για τους χρόνους διαδρομής και τις μέσες ταχύτητες κίνησης στην οδό. Το ποσοστό σφάλματος στην αναγνώριση των οχημάτων με αυτή τη μέθοδο δεν ξεπερνάει το 5%. Επίσης, η ταυτοποίηση της διέλευσης ενός οχήματος μπορεί να πραγματοποιηθεί και με τη βοήθεια του μαγνητικού <<αποτυπώματος>> που αφήνει κάθε ξεχωριστό όχημα κατά τη διέλευσή του από κάθε βρόχο. Με τη διασπορά σημείων ανίχνευσης σε ένα οδικό δίκτυο ή στις εισόδους και εξόδους αυτοκινητοδρόμων είναι δυνατή η εξαγωγή δεδομένων σχετικά με την προέλευση και προορισμό των μετακινήσεων.

Τα πλεονεκτήματα της χρήσης μαγνητικών ανιχνευτών βρόχου ως συστήματα παρακολούθησης είναι η απλότητα και το χαμηλό κόστος εγκατάστασης, όπως και η δεδομένη εμπειρία από την πολύχρονη χρήση τους. Επίσης, δίνουν αξιόπιστα αποτελέσματα, ενώ η λειτουργία τους δεν επηρεάζεται από τις καιρικές συνθήκες. Μειονέκτημά τους είναι η ανάγκη επέμβασης επάνω στην οδό, τόσο για την εγκατάσταση, όσο και για τη συντήρησή τους.

Αισθητήρες ανίχνευσης έξω από το οδόστρωμα

Ως εναλλακτική λύση στους μαγνητικούς ανιχνευτές βρόχου στο πεδίο της απόκτησης κυκλοφοριακών δεδομένων, έχουν επινοηθεί και χρησιμοποιούνται διάφοροι τύποι συσκευών ανίχνευσης, τοποθετούμενοι έξω από το οδόστρωμα. Το χαρακτηριστικό όλων αυτών των συσκευών είναι η παρακολούθηση επάνω ή δίπλα από την οδό, με τοποθέτηση σε δικές τους ή υπάρχουσες διατάξεις στήριξης. Κατά συνέπεια, δεν απαιτείται η παρενόχληση της κυκλοφορίας κατά την εγκατάσταση, λειτουργία και συντήρηση των διατάξεων αυτών.

Μερικές από τις διατάξεις που μπορούν να συναντηθούν είναι:

- **Ανιχνευτές μικροκυμάτων:** Κατά τη λειτουργία τους εκπέμπουν μικροκύματα και μετρούν τη μεταβολή στη συχνότητα του ανακλώμενου επάνω στην επιφάνεια παρακολούθησης σήματος. Η μεταβολή αυτή είναι ανάλογη με την ταχύτητα των οχημάτων και μπορεί να υπολογιστεί με μεγάλη ακρίβεια. Ωστόσο, η μέτρηση είναι δυνατή μόνο σε περίπτωση ελαφράς κυκλοφορίας. Σε πυκνή, αργή ή στάσιμη κυκλοφορία η λειτουργία είναι προβληματική έως αδύνατη.
- **Ανιχνευτές υπερήχων:** Υπολογίζουν την απόσταση από τα οχήματα, με τη βοήθεια υπερήχων που ανακλώνται επάνω σε αυτά και στο οδόστρωμα. Η λειτουργία τους παρεμποδίζεται από παράσιτα πηγών θορύβου.
- **Ενεργητικοί ανιχνευτές υπερύθρων:** Εκπέμπουν μία σειρά από αόρατες υπέρυθρες ακτίνες και αναλύουν τις αντανακλάσεις από τις λείες επιφάνειες των οχημάτων. Μπορούν να μετρήσουν κυκλοφοριακούς φόρτους και ταχύτητες οχημάτων.
- **Παθητικοί ανιχνευτές υπερύθρων:** Υπολογίζουν τις μεταβολές στη θερμική ακτινοβολία που προκαλούνται σε ένα συγκεκριμένο πεδίο λήψης.

Εικόνα 2. Παθητικός ανιχνευτής υπερύθρων

Οι ανιχνευτές αυτοί μπορούν να λειτουργήσουν υπό οποιεσδήποτε συνθήκες κυκλοφορίας, απαιτούν πολύ λίγη ενέργεια και το κόστος τους είναι εξαιρετικά ανταγωνιστικό, ωστόσο πάσχουν στην ακρίβεια υπολογισμού της ταχύτητας.

- **Ανιχνευτές λέιζερ:** Χρησιμοποιούν ακτινοβολία τύπου λέιζερ για να υπολογίσουν την απόσταση από τα οχήματα, ενώ είναι ικανοί να προσδιορίσουν και το περίγραμμά τους. Υπολογίζουν με μεγάλη ακρίβεια φόρτους, ταχύτητα και ταξινόμηση κατά τύπο οχήματος, αλλά το κόστος τους καθιστά απαγορευτική τη μαζική τους χρήση.

Κάθε τύπος από τους προαναφερθέντες παρουσιάζει διάφορα πλεονεκτήματα και μειονεκτήματα, καθώς και διάφορες δυνατότητες ανίχνευσης συγκεκριμένων στοιχείων κυκλοφορίας, ενώ σημαντική παράμετρος στη λειτουργία τους είναι και οι περιβαλλοντικές συνθήκες. Για τη διεύρυνση των δυνατοτήτων ανίχνευσης μπορούν να συνδυαστούν επιμέρους τεχνολογίες, όπως παθητικοί ανιχνευτές υπερύθρων μαζί με υπέρηχους ή μικροκύματα. Πάντως, από τους προαναφερθέντες τύπους ο μόνος που διαθέτει την ικανότητα αναγνώρισης οχημάτων, οπότε μπορεί και να εφαρμοστεί για ταυτοποίηση, σε αναλογία με τους ανιχνευτές βρόχου, είναι οι ανιχνευτές με λέιζερ, το κόστος των οποίων, όμως, είναι πολύ υψηλό. Κατά συνέπεια, μπορεί να εξαχθεί το συμπέρασμα ότι οι ανιχνευτές βρόχου μάλλον θα παραμείνουν για αρκετό καιρό ακόμη οι δημοφιλέστεροι, με τους ανιχνευτές εκτός οδοστρώματος να αποτελούν απλώς εναλλακτική λύση όπου δεν είναι δυνατή η εφαρμογή τους.

Κλειστά αναλογικά κυκλώματα τηλεόρασης

Η χρήση κλειστών κυκλωμάτων τηλεόρασης αποτελεί την πιο συνηθισμένη μέθοδο επιτήρησης σε οποιοδήποτε πεδίο, οπότε και η επιτήρηση της κυκλοφορίας δεν θα μπορούσε να αποτελεί εξαίρεση στον κανόνα. Μία σειρά από αναλογικές κάμερες, που εφαρμόζονται κατά μήκος μίας οδού ή σε συγκεκριμένα σημεία ενός οδικού δικτύου, μεταφέρει την εικόνα σε οθόνες στο κέντρο διαχείρισης, όπου το προσωπικό μπορεί να παρακολουθήσει άμεσα τη διεξαγωγή της κυκλοφορίας, ενώ είναι δυνατή και η εγγραφή σε βίντεο.

Ασφαλώς η μέθοδος αυτή παρακολούθησης, τουλάχιστον άμεσα, μπορεί να εφαρμοστεί μόνο για επίβλεψη, και όχι για απόκτηση κυκλοφοριακών δεδομένων. Κατά την επίβλεψη της κυκλοφορίας με τη μέθοδο κλειστού κυκλώματος παρακολούθησης, το προσωπικό του κέντρου διαχείρισης είναι σε θέση να διαπιστώσει διάφορες προβληματικές καταστάσεις όπως ατυχήματα, συμφορήσεις, παράνομα ή προβληματικά κινούμενα οχήματα, δυσμενείς καιρικές συνθήκες, και να αντιδράσει άμεσα, γνωρίζοντας και το πραγματικό μέγεθος του προβλήματος. Σαν απλά παραδείγματα, σε περίπτωση κυκλοφοριακής συμφόρησης σε κάποιο σημείο αυτοκινητοδρόμου το προσωπικό μπορεί να κανονίσει εύκολα και άμεσα την απεικόνιση ενός προειδοποιητικού μηνύματος σε πινακίδα μεταβλητών μηνυμάτων, ή να κινήσει άμεσα τις απαραίτητες διαδικασίες σε περίπτωση ατυχήματος. Σημαντικός, επίσης, είναι ο ρόλος της επίβλεψης και στην περίπτωση των σηράγγων, όπου ένα ατύχημα ή μία δυσλειτουργία του μηχανολογικού εξοπλισμού μπορεί να αποβούν εξαιρετικά επικίνδυνα.

Ένα από τα σημαντικότερα στοιχεία στην απόδοση του όλου συστήματος επίβλεψης είναι η κατάλληλη τοποθέτηση των καμερών. Αυτή εξαρτάται τόσο από τις απαιτήσεις της επίβλεψης, όσο και από τις δυνατότητες της ίδιας της συσκευής που επιλέγεται. Υπάρχουν συσκευές με δυνατότητες περιστροφής κατά τον οριζόντιο και κατακόρυφο άξονα, καθώς και μεγέθυνσης και εστίασης διαφόρων βαθμών, άμεσα χειριζόμενες σε πραγματικό χρόνο από το προσωπικό του κέντρου. Αυτές οι διατάξεις, βέβαια, απαιτούν πιο σύνθετη υποδομή επικοινωνίας, αλλά επιτρέπουν την κάλυψη ευρύτερης και μεγαλύτερου μήκους περιοχής, μέχρι και 800 m βάθους.

Σοβαρό ελάττωμα της μεθόδου επίβλεψης με κλειστό κύκλωμα τηλεόρασης αποτελεί η δυσκολία λήψης στο σκοτάδι και υπό δυσμενείς συνθήκες ορατότητας, όπως βροχή, χιόνι, ομίχλη, σκόνη, καπνός. Επειδή, δε, είναι αυτές οι περιπτώσεις που καθίσταται ίσως περισσότερο χρήσιμη η επιτήρηση, αρκετοί κατασκευαστές έχουν προχωρήσει στη δημιουργία συσκευών λήψης με υπέρυθρες ακτίνες, που διαθέτουν βελτιωμένες σχετικές ικανότητες, όπως φαίνεται στις Εικόνες 3 έως 5.

Εικόνα 3. Άποψη από συνήθεις συνθήκες ορατότητας

Εικόνα 4. Νυχτερινή άποψη με κάμερα βραχέων υπερύθρων

Εικόνα 5. Νυχτερινή άποψη με κάμερα μακρών υπερύθρων

Παρακολούθηση με ψηφιακή επεξεργασία εικόνας

Η εφαρμογή συστημάτων ψηφιακής επεξεργασίας εικόνας αποτελεί την πιο σύγχρονη, αποτελεσματική και συγχρόνως ολοκληρωμένη μέθοδο παρακολούθησης της κυκλοφορίας. Κατά τη μέθοδο αυτή, όπως και στην περίπτωση των κλειστών κυκλωμάτων τηλεόρασης, λαμβάνεται εικόνα από την οδό με τη βοήθεια κάμερας και αναπαράγεται σε οθόνες του κέντρου διαχείρισης. Πλην όμως, η εικόνα αυτή εισάγεται και σε κατάλληλη ηλεκτρονική υπολογιστική μονάδα, όπου και υφίσταται επεξεργασία για τη λήψη όλων των επιθυμητών στοιχείων, παρακάμπτοντας τον ανθρώπινο παράγοντα. Η παρακολούθηση της κυκλοφορίας με ψηφιακή επεξεργασία εικόνας μπορεί να χρησιμοποιηθεί τόσο για εξαγωγή κυκλοφοριακών δεδομένων, όσο και για επιτήρηση της κυκλοφορίας.

Η αρχή λειτουργίας του συστήματος φαίνεται στην Εικόνα 6. Κατά την εγκατάσταση του συστήματος, αφού οριστικοποιηθεί η θέση της κάμερας και ξεκινήσει η λήψη της εικόνας, ο χειριστής ορίζει στην οθόνη γραμμές και περιοχές ανίχνευσης, ανάλογα με τα στοιχεία που είναι επιθυμητό να λαμβάνονται. Μόλις κάποιο όχημα πατήσει κάποια γραμμή ή εισέλθει σε κάποια περιοχή, ανιχνεύεται. Στη συνέχεια, μία σειρά από αλγορίθμους αναλαμβάνει να επεξεργαστεί περαιτέρω την εικόνα και να εξάγει όλα τα επιθυμητά στοιχεία, απεικονίζοντας τα στην οθόνη, αλλά και αποθηκεύοντας τα για δημιουργία διαχρονικών δεδομένων.

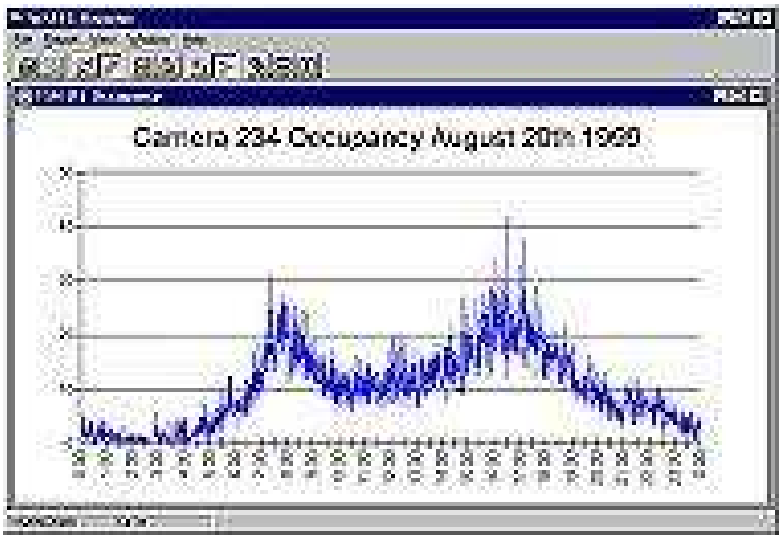
Εικόνα 6. Εικόνα λειτουργίας συστήματος ψηφιακής επεξεργασίας εικόνας. Διακρίνονται οι γραμμές ανίχνευσης

Συνοπτικά, οι δυνατότητες που παρέχονται από τη χρήση ενός σχετικού συστήματος μπορεί να είναι:

- Μέτρηση κυκλοφοριακών φόρτων
- Μέτρηση ταχύτητας οχημάτων και μέσης ταχύτητας κίνησης
- Κατανομή κυκλοφορίας κατά λωρίδα
- Χωρικοί και χρονικοί διαχωρισμοί
- Πυκνότητα κυκλοφορίας
- Εκτίμηση μήκους οχημάτων και αντίστοιχη ταξινόμηση φόρτων
- Αναγνώριση παρουσίας οχημάτων σε εισόδους κόμβων
- Μέτρηση μήκους ουράς αναμονής σε εισόδους κόμβων
- Αναγνώριση συμφορήσεως
- Αναγνώριση προβληματικής κίνησης οχημάτων, όπως υψηλή ή χαμηλή ταχύτητα και απότομη μεταβολή της, στάση, αντίθετη κίνηση
- Ανίχνευση αντικειμένων στο οδόστρωμα
- Ανίχνευση καπνού ή ομίχλης
- Δυνατότητα παρακολούθησης οχήματος βάσει διαστάσεων και χρώματος

Βέβαια, μία εγκατάσταση ψηφιακής επεξεργασίας εικόνας δεν είναι απαραίτητο να περιλαμβάνει όλες τις προαναφερθείσες δυνατότητες ταυτόχρονα, παρά μόνο όσες χρειάζονται σε κάθε συγκεκριμένη περίπτωση. Συνήθως το διαθέσιμο λογισμικό διατίθεται σε τρεις διαφορετικές δυνατότητες, **για μέτρηση κυκλοφοριακών δεδομένων**, για **ανίχνευση περιστατικών (επιτήρηση)** και για **διαχείριση σηματοδοτημένων κόμβων**.

Κατά τη **μέτρηση κυκλοφοριακών δεδομένων** μπορούν να μετρηθούν διάφορα στοιχεία όπως ο κυκλοφοριακός φόρτος, οι ταχύτητες των οχημάτων, η κατανομή κατά λωρίδα, οι χωρικοί και χρονικοί διαχωρισμοί, η πυκνότητα κυκλοφορίας. Επίσης, υπάρχει η δυνατότητα μέτρησης του μήκους κάθε οχήματος, με αντίστοιχη κατανομή της κυκλοφορίας κατά κατηγορία. Τα στοιχεία μήκους σε συνδυασμό με το χρώμα, επιτρέπουν και την ταυτοποίηση της διέλευσης ενός συγκεκριμένου οχήματος από επόμενα παρακολουθούμενα σημεία, για την εξαγωγή μέσων ταχυτήτων και χρόνων διαδρομής, όπως και δεδομένων προέλευσης και προορισμού, σε αναλογία με τα όσα εκτέθηκαν σχετικά με τους μαγνητικούς βρόχους παραπάνω. Το πλεονέκτημα είναι ότι με τη βοήθεια του λογισμικού όλα αυτά τα στοιχεία μπορούν να παρουσιαστούν σε διάφορες μορφές, καθώς και να αποθηκευτούν και να χρησιμοποιηθούν για εξαγωγή διαχρονικών στοιχείων, εντελώς αυτόματα.



Εικόνα 7. Παρουσίαση στοιχείων κυκλοφορίας

Η **ανίχνευση διαφόρων ειδών περιστατικών** πραγματοποιείται και αυτή αυτόματα, με τη δυνατότητα παρατήρησης συμφόρησης, σταματημένων ή κινούμενων αντίθετα οχημάτων, αντικειμένων στο οδόστρωμα.

Εικόνα 8. Ανίχνευση περιστατικού

Με την ανίχνευση κάποιου περιστατικού ενημερώνεται το προσωπικό του κέντρου, το οποίο έχει βέβαια την δυνατότητα να επιβεβαιωθεί και ιδίοις όμμασι από την οθόνη του, διαπιστώνοντας το είδος και την έκτασή του και αποφασίζοντας για τις ενέργειες στις οποίες θα προβεί. Απλώς δεν απαιτείται η συνεχής επαγρύπνησή του, όπως συμβαίνει στην περίπτωση των κλειστών κυκλωμάτων τηλεόρασης. Ιδιαίτερη εφαρμογή της δυνατότητας ανίχνευσης περιστατικών μπορεί να πραγματοποιηθεί σε επικίνδυνα σημεία, όπως οι σήραγγες, Εικόνα 9.

Εικόνα 9. Ανίχνευση σταματημένου οχήματος σε σήραγγα

Τέλος, κατά τη δυνατότητα της **διαχείρισης σηματοδοτημένων κόμβων** το σύστημα επεξεργασίας έχει τη δυνατότητα ανίχνευσης οχημάτων που αναμένουν ή προσεγγίζουν στην παρακολουθούμενη πρόσβαση, τη μέτρησή τους, καθώς και την εκτίμηση του μήκους ενδεχόμενης ουράς. Τα στοιχεία αυτά χρησιμοποιούνται στη διαχείριση της υπάρχουσας σηματοδότησης.

Εικόνα 10. Ανίχνευση αναμενόντων οχημάτων σε σηματοδότη

Βασικοί παράγοντες στην αποτελεσματικότητα και αξιοπιστία της ηλεκτρονικής παρακολούθησης είναι το είδος και η θέση των συσκευών λήψης εικόνας που χρησιμοποιούνται. Καθώς η ποιότητα της γραφικής επεξεργασίας εξαρτάται από την αντίστοιχη της εικόνας, οι κάμερες που χρησιμοποιούνται στα συστήματα ψηφιακής επεξεργασίας απαιτείται να είναι μεγαλύτερης ευκρίνειας σε σχέση με τις κοινές κάμερες των κλειστών κυκλωμάτων. Έγχρωμες κάμερες είναι δυνατόν να χρησιμοποιηθούν, αλλά είναι εν γένει λιγότερο ευαίσθητες σε σχέση με τις ασπρόμαυρες.

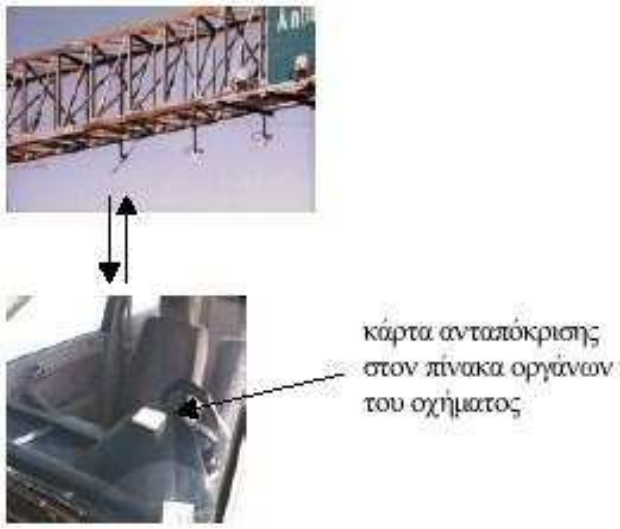
Εικόνα 11. Τύποι συσκευών λήψης εικόνας για εφαρμογές ψηφιακής επεξεργασίας

Η βέλτιστη θέση κάθε κάμερας εξαρτάται από το είδος της εφαρμογής, καθώς και από τις περιβαλλοντικές συνθήκες. Γενικά οι κάμερες θα πρέπει να είναι τοποθετημένες όσο το δυνατό υψηλότερα, και στο κέντρο της ζώνης ανίχνευσης. Εάν αυτό δεν είναι δυνατό, προτιμάται η τοποθέτηση κοντά στην εσωτερική λωρίδα, καθώς με αυτό τον τρόπο αποφεύγεται το γεγονός βραδέα και ογκώδη οχήματα να κλείνουν το οπτικό πεδίο στις παρακείμενες λωρίδες. Η απευθείας πρόσπτωση του ηλιακού φωτός επάνω στο φακό θα πρέπει να αποφεύγεται. Το οπτικό πεδίο εξαρτάται από το ύψος τοποθέτησης, όπως και από το εύρος ανοίγματος του φακού. Ενδεικτικά, για την ανίχνευση σταματημένων οχημάτων η ζώνη ανίχνευσης συνήθως περιορίζεται σε 350 m σε ανοιχτές οδούς και σε 15 φορές το ύψος της κάμερας μέσα σε σήραγγες. Επίσης, μία κάμερα με σχεδόν κατακόρυφη τοποθέτηση (κοιτάζοντας προς τα κάτω) παρέχει σαφέστερη διάκριση των διαδοχικών οχημάτων, μειώνοντας τα σφάλματα αναγνώρισης καθότι φαίνονται τα διάκενα μεταξύ τους, αλλά το εξαιρετικά περιορισμένο πεδίο λήψης την καθιστά ακατάλληλη για επιτήρηση.

Τέλος, αναφέρεται ότι, όπως και στην περίπτωση των κλειστών κυκλωμάτων, έτσι και εδώ βασική αδυναμία της μεθόδου είναι η δυσκολία ανίχνευσης σε συνθήκες χαμηλού φωτισμού ή περιορισμένης ορατότητας. Ομοίως, αντιμετώπιση του προβλήματος επιτυγχάνεται με χρήση τεχνολογιών όπως οι υπέρυθρες ακτινοβολίες.

Παρακολούθηση με τη βοήθεια καρτών ανταπόκρισης οχημάτων

Πέρα από τις προαναφερθείσες, μία ακόμη μέθοδος που εφαρμόζεται σε αρκετές περιοχές του κόσμου και χρίζει αναφοράς είναι και η παρακολούθηση με τη βοήθεια καρτών ανταπόκρισης στο εσωτερικό των οχημάτων. Η μέθοδος αυτή είναι ευρύτερα γνωστή με τη μορφή της ηλεκτρονικής χρέωσης διοδίων. Οι οδηγοί που επιθυμούν την αποφυγή της ταλαιπωρίας στάσης σε σταθμούς διοδίων εφοδιάζονται με μία ειδική κάρτα, την οποία τοποθετούν στο εσωτερικό του οχήματος. Σε κάθε σταθμό διοδίων υπάρχει ειδικός αναγνώστης, που αναγνωρίζει το συγκεκριμένο όχημα βάσει της κάρτας και χρεώνει το σχετικό αντίτιμο στον οδηγό.



Εικόνα 12. Λειτουργία συστήματος παρακολούθησης με κάρτες ανταπόκρισης οχημάτων

Η δυνατότητα αυτή αναγνώρισης κάθε συγκεκριμένου οχήματος μπορεί να χρησιμοποιηθεί ταυτόχρονα και για την ταυτοποίηση της διέλευσης του οχήματος και από επόμενους σταθμούς ανάγνωσης, παρέχοντας κυκλοφοριακά δεδομένα. Βέβαια, η μέθοδος αυτή προφανώς συνεργάζεται μόνο με τα εφοδιασμένα με κάρτα οχήματα, άρα τα αξιόπιστα δεδομένα που μπορεί να παρέχει είναι πολύ φτωχά, περιορισμένα μόνο σε μέσες ταχύτητες και χρόνους διαδρομής. Ωστόσο, η μέθοδος αυτή είναι και αντίστοιχα απλή, φθηνή και εύκολης εφαρμογής, καθώς απαιτεί απλώς την τοποθέτηση αναγνωστών σε οποιοδήποτε πρόσφορο σημείο (μαζί με την υποδομή επικοινωνίας, βέβαια) και αξιοποιεί την ήδη υπάρχουσα υποδομή χρηστών-κατόχων κάρτας.

Κεφάλαιο 2

Πειραματικό μέρος

Αφού αναλύσαμε παραπάνω τους δύο τύπους εφαρμογών που υπάρχουν σε αυτήν την εργασία(ψηφιακή επεξεργασία στατικών εικόνων και συστήματα παρακολούθησης κυκλοφορίας), είμαστε έτοιμοι να προχωρήσουμε στα επόμενα κεφάλαια αναλύοντας εκτενέστερα τους τρόπους υλοποίησης αυτών των εφαρμογών.

Από αυτό το κεφάλαιο μέχρι και το κεφάλαιο 8 ασχολούμαστε με το πειραματικό μέρος της εργασίας αυτής. Πιο συγκεκριμένα στα κεφάλαια 2 έως 6 ασχολούμαστε με την ψηφιακή επεξεργασία στατικών εικόνων, στο κεφάλαιο 7 ασχολούμαστε με την ψηφιακή επεξεργασία εικόνας μέσα από βίντεο, ενώ στο κεφάλαιο 8 δημιουργούμε ένα GUI το οποίο μπορούμε να το κατατάξουμε σαν ένα σύστημα παρακολούθησης κυκλοφορίας.

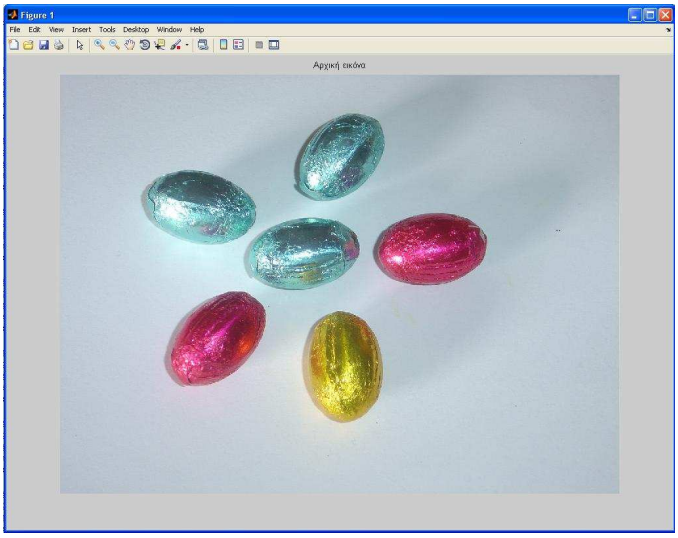
Αναγνώριση αντικειμένων σε μονότονο background

Ένα αντικείμενο σε μια εικόνα είναι εύκολο να αναγνωριστεί εάν το background της εικόνας είναι μονότονο Με την χρησιμοποίηση των ακμών της εικόνας και μερικών βασικών μορφολογικών εργαλείων είναι εύκολο να γίνει.

Περιεχόμενα

Βήμα 1: Διάβασμα εικόνας

i=imread('chocolate.jpg');
figure,imshow(i),title('Αρχική εικόνα');

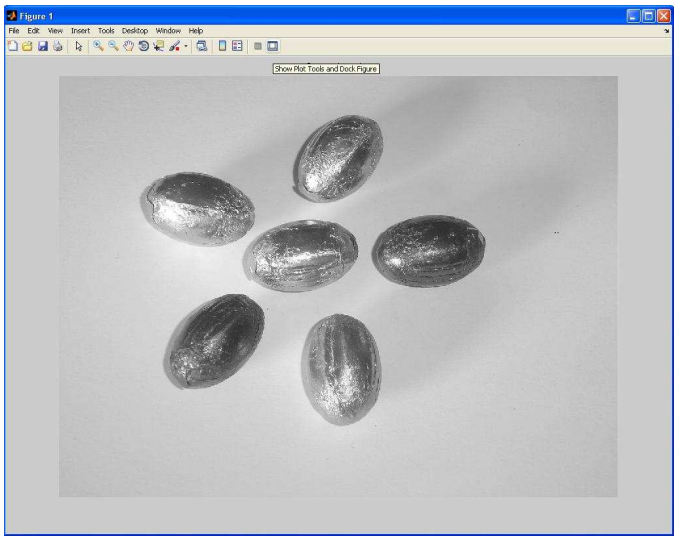


Σχήμα 1

Όπως βλέπουμε και στο σχήμα 1 με την συνάρτηση ***imread*** το πρόγραμμα διαβάζει την εικόνα <<chocolate.jpg>> και με την εντολή ***imshow*** μας την απεικονίζει στην οθόνη.

Βήμα 2: Μετατροπή εικόνας από RGB σε GRAYSCALE

igray=rgb2gray(i);
figure,imshow(igray),title('Grayscale εικόνα');



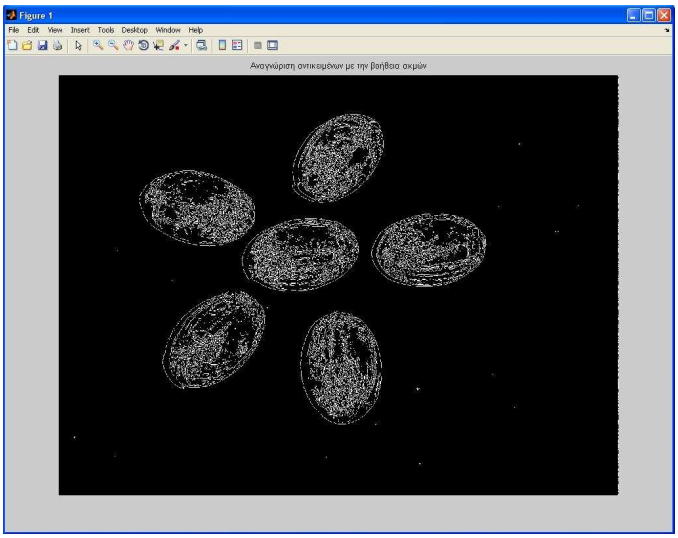
Σχήμα 2

Επειδή είναι πιο εύκολη η επεξεργασία μιας εικόνας με όσο το δυνατόν λιγότερα επίπεδα φωτεινότητας, γι'αυτό μετατρέπουμε την εικόνα σε grayscale(255 επίπεδα γκρι). Όπως φαίνεται στο σχήμα 2 για την μετατροπή της εικόνας μας βοηθάει η συνάρτηση ***rgb2gray***.

Βήμα 3: Αναγνώριση αντικειμένων με την βοήθεια ακμών

Επειδή τα αντικείμενα στην εικόνα μας έχουν διαφορετικό contrast από το background μπορούμε να αναδείξουμε τις λεπτομέρειες τους με την βοήθεια των ακμών.

[value threshold]=edge(igray,'sobel');
timi=.5;
binary=edge(igray,'sobel',threshold*timi);
figure, imshow(binary),title('Αναγνώριση αντικειμένων με την βοήθεια ακμών');



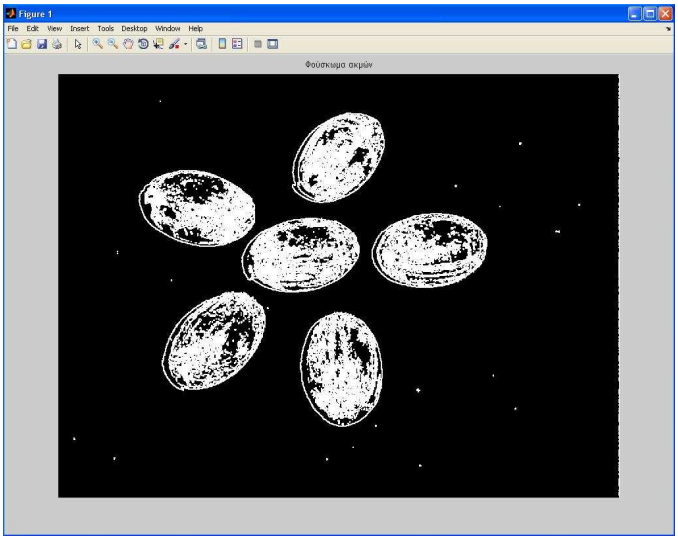
Σχήμα 3

Χρησιμοποιώντας την συνάρτηση ***edge*** και με ένα κατώφλι(threshold) της τάξεως του 0.5 δημιουργούμε μια δυαδική εικόνα όπου απεικονίζονται οι ακμές στο άσπρο χρώμα και το background στο μαύρο. Τα pixels που είχαν τιμή μεγαλύτερη του 0.5 πήραν τιμή 1(άσπρο), ενώ τα υπόλοιπα πήραν τιμή 0(μαύρο).

Βήμα 4: Διόγκωση ακμών

Για να γίνει σωστά η αναγνώριση των αντικειμένων θα πρέπει να <<φουσκώσουμε >> τις ακμές με αποτέλεσμα να πάρουμε όλο το εμβαδόν του αντικειμένου.

```
se90 = strel('line', 3, 90);
se0 = strel('line', 3, 0);
binarydil=imdilate(binary,[se90 se0]);
figure, imshow(binarydil), title('Φούσκωμα ακμών');
```



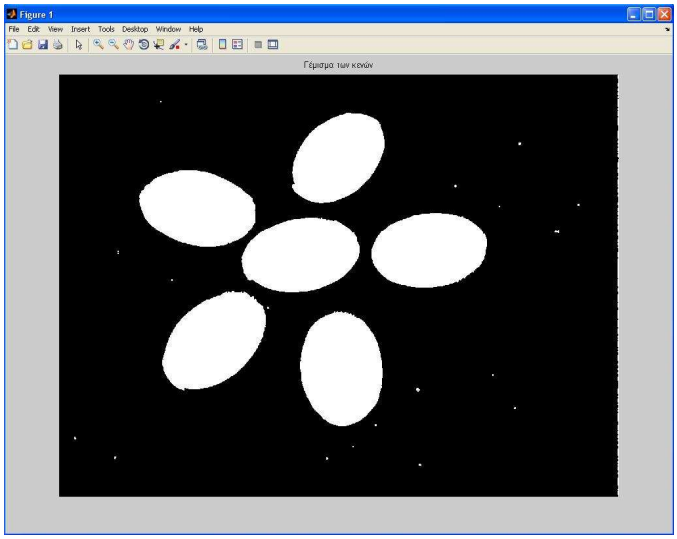
Σχήμα 4

Με την μορφολογική συνάρτηση ***strel*** παίρνουμε τις διαστάσεις των αντικειμένων και με την συνάρτηση ***imdilate*** πετυχαίνουμε την διόγκωση των ακμών όπως φαίνεται και στο σχήμα 4.

Βήμα 5: Γέμισμα των κενών

Αφού διογκώσαμε τις ακμές στο προηγούμενο βήμα αυτό που απομένει για να πάρουμε όλο το εμβαδόν των αντικειμένων είναι να γεμίσουμε τα μαύρα διαστήματα μέσα στα αντικείμενα.

binaryfill = imfill(binarydil,'holes');
figure, imshow(binaryfill),title('Γέμισμα των κενών');



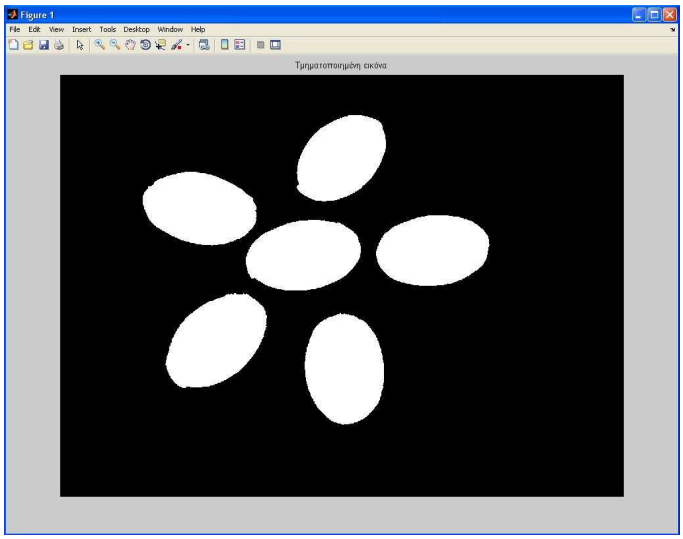
Σχήμα 5

Όπως φαίνεται στο σχήμα 5 η συνάρτηση ***imfill*** με το όρισμα holes μας βοηθάει να πετύχουμε τον σκοπό μας.

Βήμα 6: Ολοκλήρωση της αναγνώρισης των αντικειμένων

Στην εικόνα που φαίνεται στο σχήμα 5 παρατηρούμε ότι έχουν μείνει κάποια άσπρα σημεία έξω από την περιοχή του ενδιαφέροντος μας. Για να ολοκληρώσουμε την αναγνώριση των αντικειμένων πρέπει αυτά τα άσπρα σημεία να απαλειφθούν από την εικόνα μας.

elementd=strel('diamond',2);
binaryfinal=imerode(binarynoboard,elementd);
binaryfinal=imerode(binaryfinal,elementd);
figure,imshow(binaryfinal),title('Τμηματοποιημένη εικόνα');

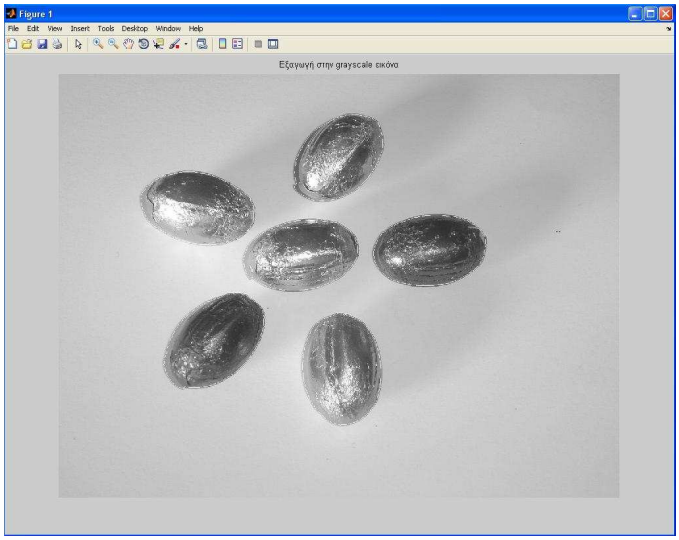


Σχήμα 6

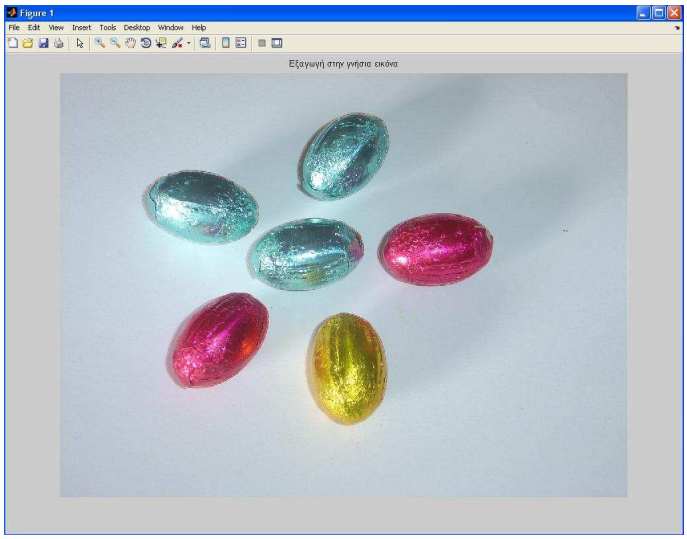
Η συνάρτηση ***imerode*** και με την βοήθεια της ***strel*** μας δίνουν το αποτέλεσμα που βλέπουμε στο σχήμα 6. Η συνάρτηση ***imerode*** εξαλείφει τις άσπρες περιοχές οι οποίες έχουν σχήμα ***diamond*** με μέγεθος μικρότερου του 2, όπως φαίνεται και στις εντολές παραπάνω.

Βήμα 7: Απεικόνιση του αποτελέσματος

binaryoutline=bwperim(binaryfinal);



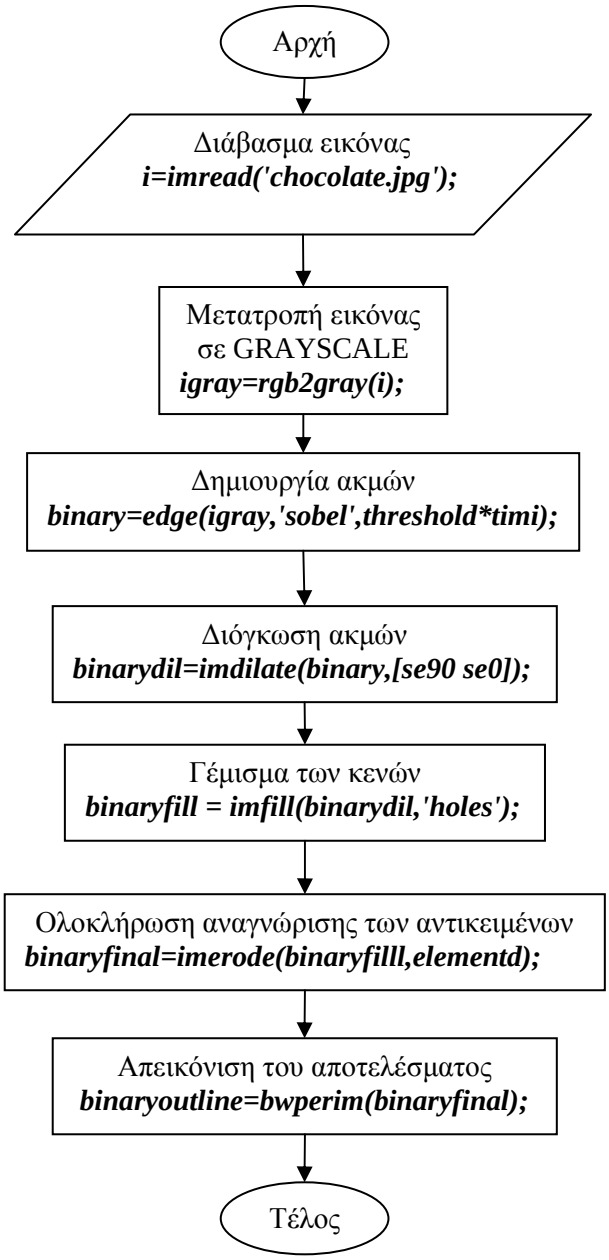
Σχήμα 7



Σχήμα 8

Η συνάρτηση ***bwperim*** δημιουργεί ένα περίγραμμα γύρω από τα αντικείμενα που έχουν αναγνωριστεί όπως φαίνεται στα σχήματα 7 και 8.

Στο παρακάτω διάγραμμα ροής απεικονίζονται τα βήματα που εκτελέστηκαν για την αναγνώριση αντικειμένων σε μονότονο background



Κεφάλαιο 3

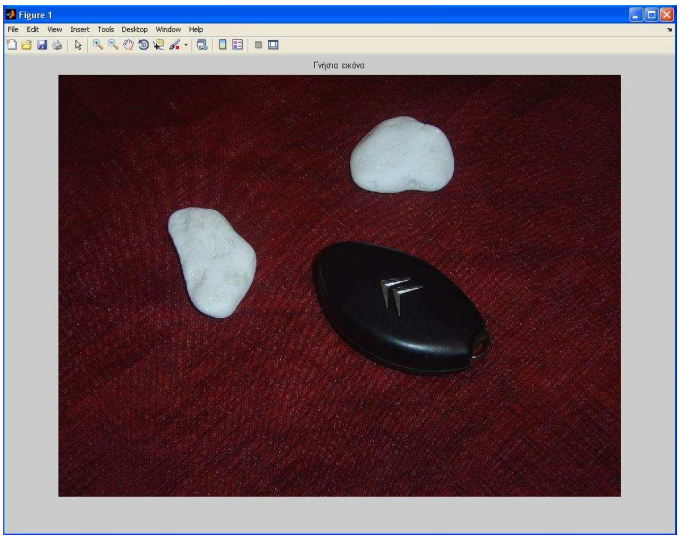
Αναγνώριση αντικειμένων <<σκοτεινιάζοντας>> την εικόνα

Για να <<σκοτεινιάσουμε>> την εικόνα χρησιμοποιούμε την συνάρτηση ***imextendedmax***. Η συνάρτηση αυτή μας γυρνάει μια δυαδική εικόνα όπου οι περιοχές που έχουν τιμή μικρότερη από το threshold(κατώφλι) που έχουμε ορίσει παίρνουν την τιμή 0(μαύρο) και οι υπόλοιπες παίρνουν την τιμή 1(άσπρο). Στα παρακάτω βήματα περιγράφεται αναλυτικότερα ο τρόπος με τον οποίο χρησιμοποιούμε αυτήν την συνάρτηση.

Περιεχόμενα

Βήμα 1: Διάβασμα εικόνας

i=imread('dark.jpg');
figure,imshow(i),title('Γνήσια εικόνα');

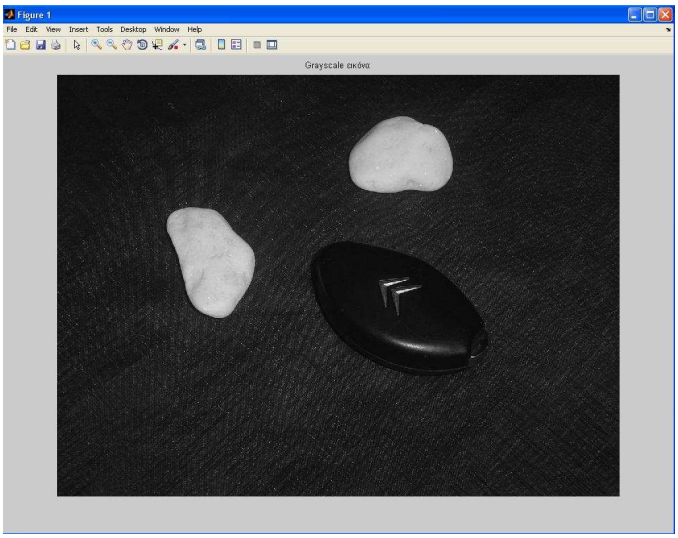


Σχήμα 1

Με την συνάρτηση ***imread*** το πρόγραμμα διαβάζει την εικόνα και με την εντολή imshow μας την απεικονίζει σε ένα παράθυρο(figure).

Βήμα 2: Μετατροπή της εικόνας από RGB σε GRAYSCALE

I=rgb2gray(i);
figure,imshow(I),title('Grayscale εικόνα');

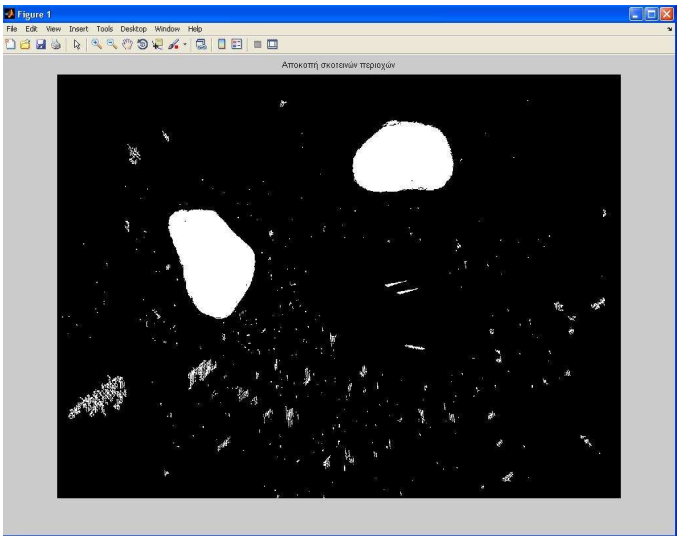


Σχήμα 2

Επειδή είναι πιο εύκολη η επεξεργασία μιας εικόνας με όσο το δυνατόν λιγότερα επίπεδα φωτεινότητας, γι'αυτό μετατρέπουμε την εικόνα σε grayscale(255 επίπεδα γκρι). Όπως φαίνεται στο σχήμα 2 για την μετατροπή της εικόνας μας βοηθάει η συνάρτηση ***rgb2gray***.

Βήμα 3: <<Σκοτείνιασμα>> εικόνας

darkValue=100;
noDarkSpot=imextendedmax(darkSpot,darkValue);
figure,imshow(noDarkSpot),title('Αποκοπή σκοτεινών περιοχών');



Σχήμα 3

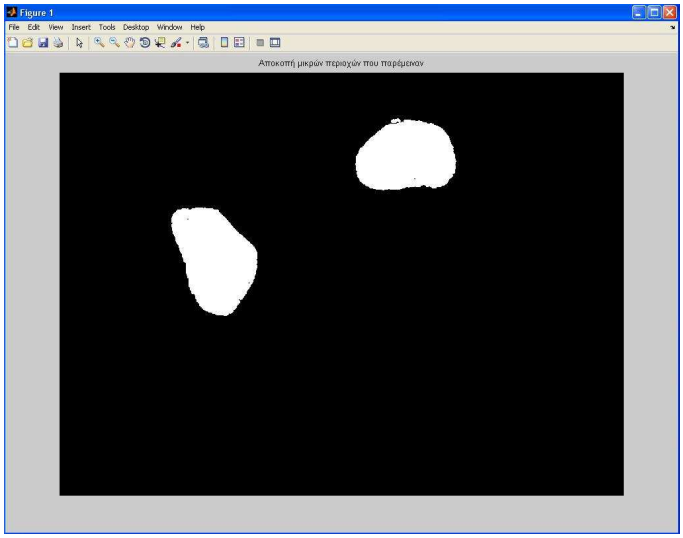
Στο σχήμα 3 φαίνεται η δυαδική εικόνα που μας γυρνάει η συνάρτηση ***imextendedmax*** Έχουμε ορίσει ένα threshold(κατώφλι) ***darkValue=100***. Όσα pixels έχουν τιμή μικρότερη του 100 γίνονται 0(μαύρο) και όσα έχουν τιμή μεγαλύτερη του 100 γίνονται 1(άσπρο).

Παρατήρηση: Το τρίτο αντικείμενο που θέλουμε να αναγνωρίσουμε είναι το κλειδί όπως φαίνεται στο σχήμα 2. Επειδή όμως το κλειδί είναι μαύρο σημαίνει ότι τα pixels του έχουν τιμές μικρότερες από το 100 δηλαδή μικρότερες από το κατώφλι μας. Αυτό έχει σαν αποτέλεσμα όταν κάνουμε εισαγωγή της συνάρτησης ***imextendedmax*** τα pixels που αναπαριστούν το κλειδί να γίνονται όλα 0 και έτσι το κλειδί να χάνεται.

Το πόρισμα που βγάζουμε αν ακολουθήσουμε αυτόν τον αλγόριθμο είναι ότι αντικείμενα που έχουν σκοτεινό ή μαύρο χρώμα να μην μπορούν να αναγνωριστούν Ας συνεχίσουμε την διαδικασία στα παρακάτω βήματα με τα άλλα δύο αντικείμενα που δεν χάθηκαν με την εισαγωγή αυτού του αλγορίθμου.

Βήμα 4: Χρησιμοποίηση της μορφολογικής συνάρτησης imopen

```
sedisk=strel('disk',4);
noSmallPixels=imopen(noDarkSpot,sedisk);
figure,imshow(noSmallPixels),title('Αποκοπή μικρών περιοχών που παρέμειναν');
```



Σχήμα 4

Όπως παρατηρήσαμε στο σχήμα 3 ναι μεν αναγνωρίστηκαν τα αντικείμενα αλλά παρέμειναν και κάποια άλλα pixels τα οποία δεν ανήκουν στην περιοχή του ενδιαφέροντος μας. Παρέμειναν γιατί είχαν τιμή μεγαλύτερη του 100 αλλά εμείς πρέπει να τα αφαιρέσουμε για να μείνουν στην εικόνα μόνο τα pixels των αντικειμένων που θέλουμε να αναγνωρίσουμε.

Για να το επιτύχουμε αυτό χρησιμοποιούμε την μορφολογική συνάρτηση ***imopen*** όπου μηδενίζει την τιμή των pixels που δεν πληρούν την συνθήκη strel('disk',4). Η συνθήκη αυτή υποδεικνύει τα pixels που έχουν μορφή δίσκου. Ο αριθμός μας δείχνει το μέγεθος του δίσκου που θέλουμε να κρατήσουμε από την εικόνα μας. Στο συγκεκριμένο παράδειγμα ο αριθμός 4 είναι αρκετός για να απαλειφθούν τα pixels που δεν θέλουμε και να πάρουμε το αποτέλεσμα του σχήματος 4.

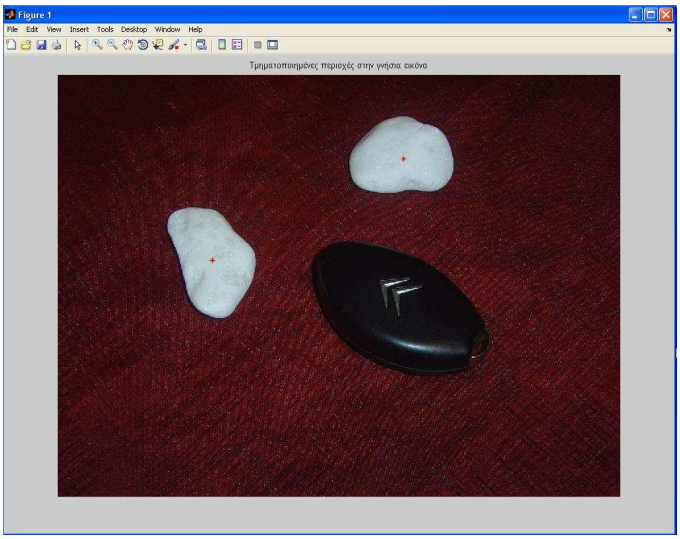
Βήμα 5: Υπολογισμός του κέντρου των αντικειμένων που παρέμειναν

s = regionprops(noSmallPixels,darkSpot, {'Centroid'});

Για να μπορέσουμε να αναδείξουμε το τελικό αποτέλεσμα στο παρακάτω βήμα πρέπει πρώτα να υπολογίσουμε σε ποια θέση βρίσκονται τα αντικείμενα που φαίνονται στο σχήμα 4. Για να το επιτύχουμε αυτό χρησιμοποιούμε την συνάρτηση ***regionprops***. Βάζοντας σαν όρισμα στην συνάρτηση το Centroid παίρνουμε τις συντεταγμένες του κέντρου των αντικειμένων από το σχήμα 4.Έτσι στο βήμα 6 μπορούμε να αναδείξουμε το τελικό αποτέλεσμα.

Βήμα 6: Απεικόνιση του αποτελέσματος

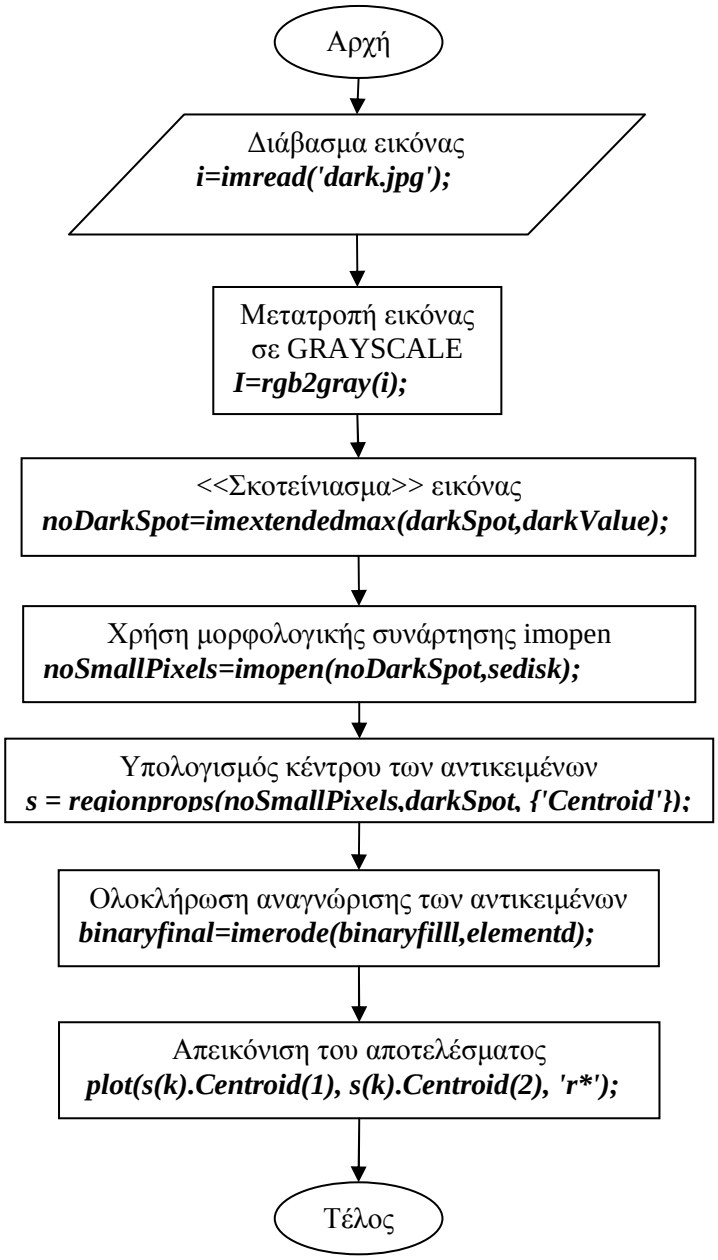
```
imshow(i)
title('Τμηματοποιημένες περιοχές στην γνήσια εικόνα');
hold on
numObj = numel(s);
for k = 1 : numObj
   plot(s(k).Centroid(1), s(k).Centroid(2), 'r*');
end
hold off
```



Σχήμα 5

Αφού υπολογίσαμε το κέντρο των αντικειμένων στο βήμα 5 με την βοήθεια της plot τοποθετούμε ένα κόκκινο αστερίσκο και παίρνουμε το αποτέλεσμα του σχήματος 5

Στο παρακάτω διάγραμμα ροής απεικονίζονται τα βήματα που εκτελέστηκαν για την αναγνώριση αντικειμένων <<σκοτεινιάζοντας>> την εικόνα.



Κεφάλαιο 4

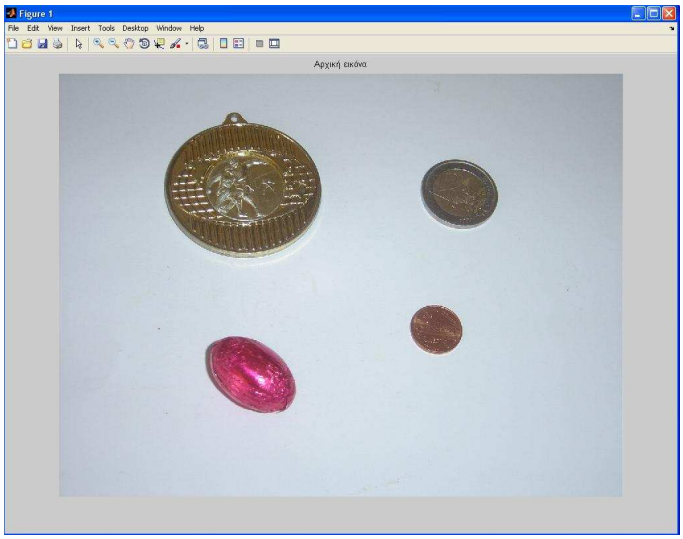
Αναγνώριση αντικειμένων με βάση το μέγεθος τους

Στόχος μας είναι να γίνει η αναγνώριση των αντικειμένων με βάση το μέγεθος τους. Έτσι θα μπορούμε εύκολα να αναγνωρίσουμε το μεγαλύτερο ή το μικρότερο αντικείμενο σε μια εικόνα. Στο παρακάτω παράδειγμα θα αναγνωρίσουμε το μικρότερο αντικείμενο από την εικόνα.

Περιεχόμενα

Βήμα 1: Διάβασμα εικόνας

l=imread('synthetic.JPG');
figure,imshow(l),title('Αρχική εικόνα');

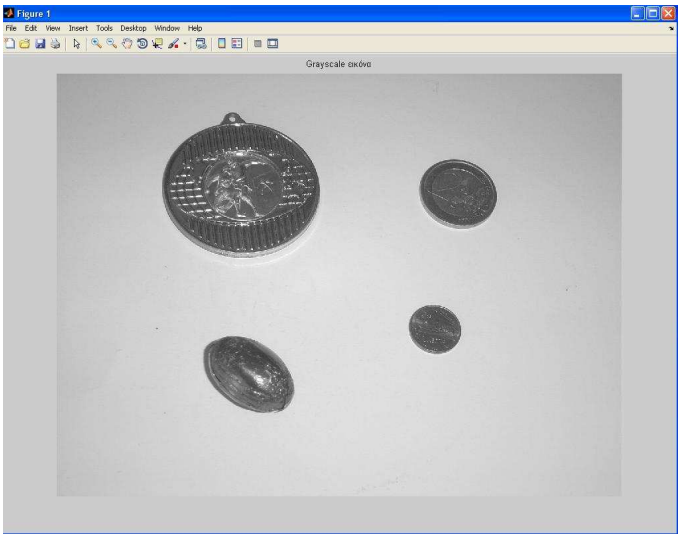


Σχήμα 1

Με την συνάρτηση ***imread*** το πρόγραμμα διαβάζει την εικόνα και με την εντολή imshow μας την απεικονίζει σε ένα παράθυρο(figure).

Βήμα 2: Μετατροπή της εικόνας από RGB σε GRAYSCALE

I=rgb2gray(l);
figure,imshow(I),title('Grayscale εικόνα');



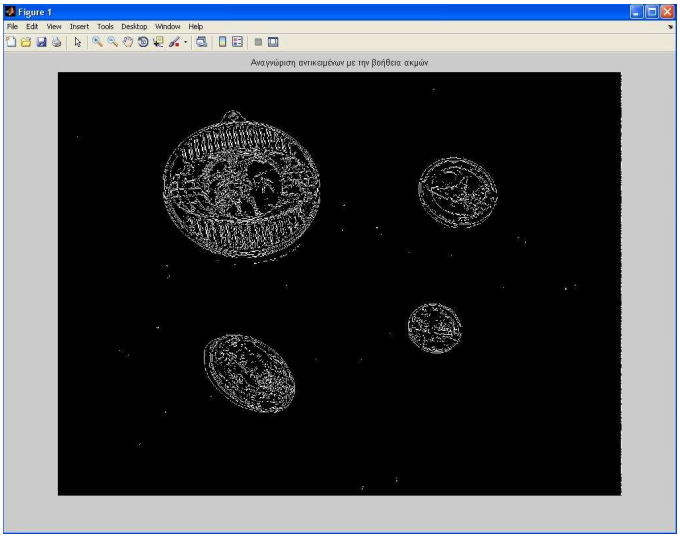
Σχήμα 2

Επειδή είναι πιο εύκολη η επεξεργασία μιας εικόνας με όσο το δυνατόν λιγότερα επίπεδα φωτεινότητας, γι'αυτό μετατρέπουμε την εικόνα σε grayscale(255 επίπεδα γκρι). Όπως φαίνεται στο σχήμα 2 για την μετατροπή της εικόνας μας βοηθάει η συνάρτηση ***rgb2gray***.
.

Βήμα 3: Αναγνώριση αντικειμένων με την βοήθεια ακμών

Επειδή τα αντικείμενα στην εικόνα μας έχουν διαφορετικό contrast από το background μπορούμε να αναδείξουμε τις λεπτομέρειες τους με την βοήθεια των ακμών.

[value threshold]=edge(I,'sobel');
timi=.5;
binary=edge(I,'sobel',threshold*timi);
figure, imshow(binary),title('Αναγνώριση αντικειμένων με την βοήθεια ακμών');



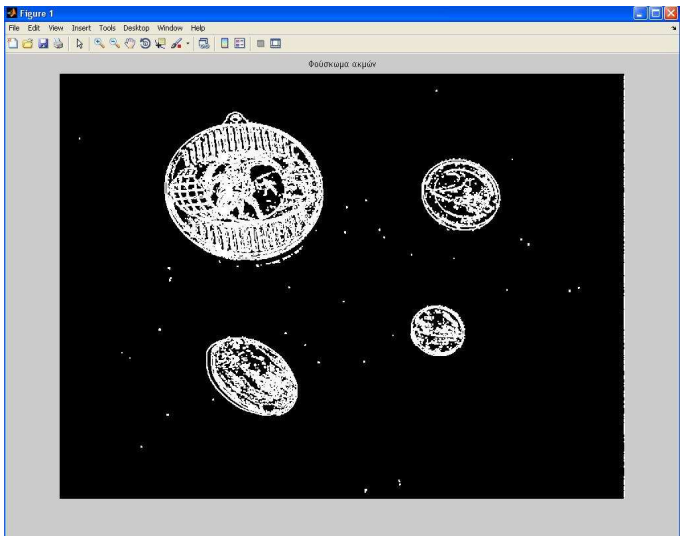
Σχήμα 3

Χρησιμοποιώντας την συνάρτηση ***edge*** και με ένα κατώφλι(threshold) της τάξεως του 0.5 δημιουργούμε μια δυαδική εικόνα όπου απεικονίζονται οι ακμές στο άσπρο χρώμα και το background στο μαύρο. Τα pixels που είχαν τιμή μεγαλύτερη του 0.5 πήραν τιμή 1(άσπρο), ενώ τα υπόλοιπα πήραν τιμή 0(μαύρο).

Βήμα 4: Διόγκωση ακμών

Για να γίνει σωστά η αναγνώριση των αντικειμένων θα πρέπει να <<φουσκώσουμε >> τις ακμές με αποτέλεσμα να πάρουμε όλο το εμβαδόν του αντικειμένου.

```
se90 = strel('line', 3, 90);
se0 = strel('line', 3, 0);
binarydil=imdilate(binary,[se90 se0]);
figure, imshow(binarydil), title('Φούσκωμα ακμών');
```



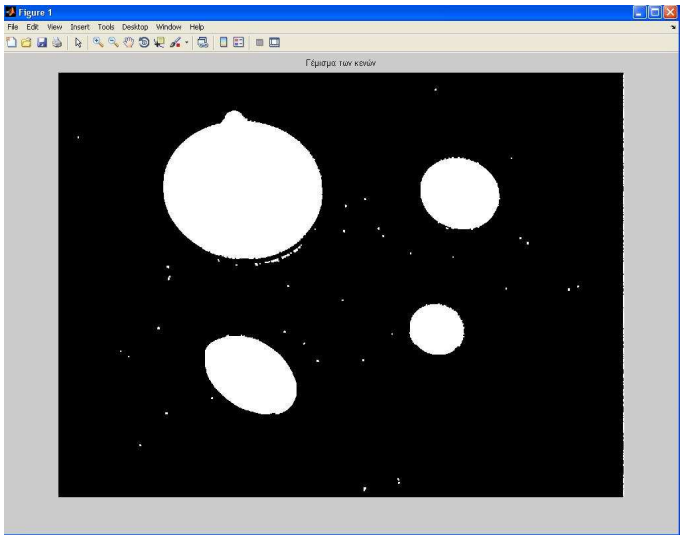
Σχήμα 4

Με την μορφολογική συνάρτηση ***strel*** παίρνουμε τις διαστάσεις των αντικειμένων και με την συνάρτηση ***imdilate*** πετυχαίνουμε την διόγκωση των ακμών όπως φαίνεται και στο σχήμα 4.

Βήμα 5: Γέμισμα των κενών

Αφού διογκώσαμε τις ακμές στο προηγούμενο βήμα αυτό που απομένει για να πάρουμε όλο το σχήμα των αντικειμένων είναι να γεμίσουμε τα μαύρα διαστήματα μέσα στα αντικείμενα.

```
binaryfill = imfill(binarydil,'holes');
figure, imshow(binaryfill),title('Γέμισμα των κενών');
```



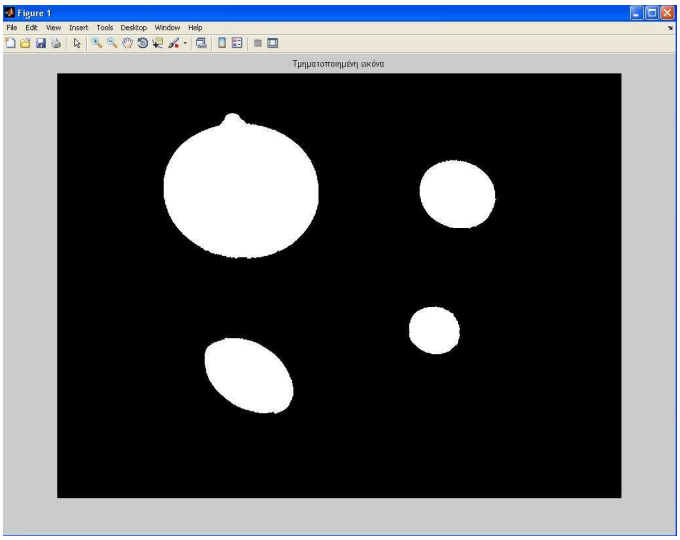
Σχήμα 5

Όπως φαίνεται στο σχήμα 5 η συνάρτηση ***imfill*** με το όρισμα holes μας βοηθάει να πετύχουμε τον σκοπό μας.

Βήμα 6: Ολοκλήρωση της αναγνώρισης των αντικειμένων

Στην εικόνα που φαίνεται στο σχήμα 5 παρατηρούμε ότι έχουν μείνει κάποια άσπρα σημεία έξω από την περιοχή του ενδιαφέροντος μας. Για να ολοκληρώσουμε την αναγνώριση των αντικειμένων πρέπει αυτά τα άσπρα σημεία να απαλειφθούν από την εικόνα μας.

```
elementd=strel('diamond',2);
BW=imerode(binarynoboard,elementd);
BW=imerode(BW,elementd);
figure,imshow(BW),title('Τμηματοποιημένη εικόνα');
```

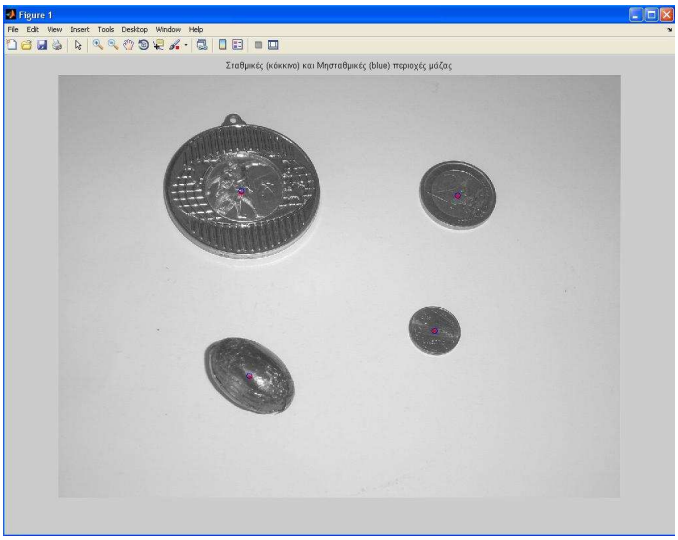


Σχήμα 6

Η συνάρτηση ***imerode*** και με την βοήθεια της ***strel*** μας δίνουν το αποτέλεσμα που βλέπουμε στο σχήμα 6.

Βήμα 7: Σύγκριση σταθμικής και μη σταθμικής μάζας των αντικειμένων

```
s=regionprops(BW,I,{‘Centroid’,WeightedCentroid’});
imshow(I)
title('Σταθμικές (κόκκινο) και Μησταθμικές (blue) περιοχές μάζας');
hold on
numObj = numel(s);
for k = 1 : numObj
   plot(s(k).WeightedCentroid(1), s(k).WeightedCentroid(2), 'r*');
   plot(s(k).Centroid(1), s(k).Centroid(2), 'bo');
end
hold off
```

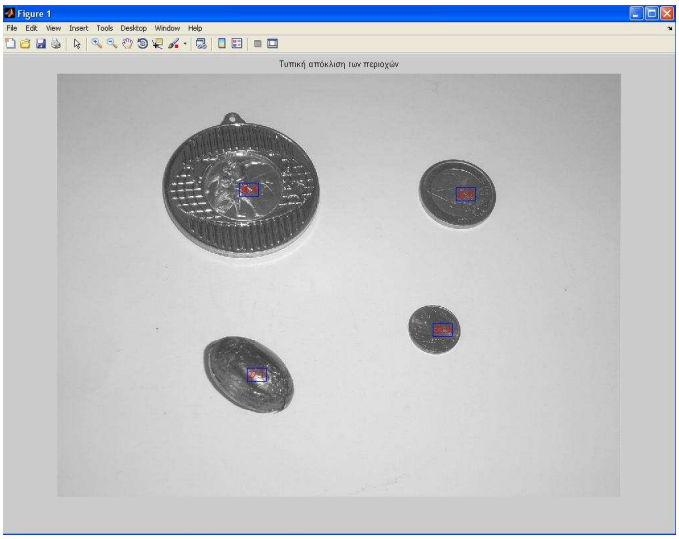


Σχήμα 7

Με την βοήθεια της συνάρτησης ***regionprops*** υπολογίζουμε την σταθμική και την μη σταθμική μάζα των αντικειμένων του σχήματος 6 που αναγνωρίσαμε βάζοντας τα ορίσματα Centroid και WeightedCentroid αντίστοιχα. Με την βοήθεια της plot συγκρίνουμε τις δύο μάζες βάζοντας ένα κόκκινο αστερίσκο για την σταθμική και ένα μπλε κύκλο για την μη σταθμική μάζα. Το αποτέλεσμα φαίνεται στο σχήμα 7

Βήμα 8: Υπολογισμός τυπικής απόκλισης των αντικειμένων

```
s = regionprops(BW, I, {'Centroid','PixelValues','BoundingBox'});
imshow(I);
title('Τυπική απόκλιση των περιοχών');
hold on
for k = 1 : numObj
   s(k).StandardDeviation = std(double(s(k).PixelValues));
   text(s(k).Centroid(1),s(k).Centroid(2), ...
      sprintf('%2.1f', s(k).StandardDeviation), ...
      'EdgeColor','b','Color','r');
end
hold off
```

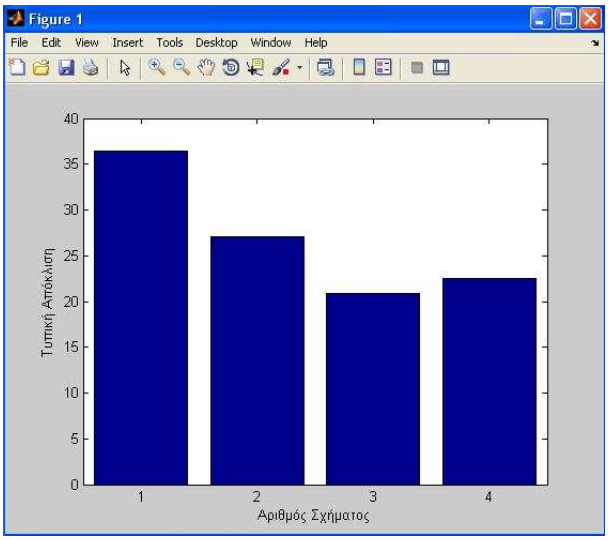


Σχήμα 8

Η εντολή s(k).StandardDeviation = std(double(s(k).PixelValues)); στον παραπάνω κώδικα μας υπολογίζει την τυπική απόκλιση των τεσσάρων αντικειμένων που έχουν αναγνωρίσει.. Επίσης τα ορίσματα στην ***regionprops*** θα πρέπει να είναι Centroid, PixelValues και BoundingBox. Το αποτέλεσμα φαίνεται στο σχήμα 8.

Βήμα 9: Αναπαράσταση τυπικής απόκλισης με την βοήθεια αξόνων

```
figure
bar(1:numObj,[s.StandardDeviation]);
xlabel('Αριθμός Σχήματος');
ylabel('Τυπική Απόκλιση');
```

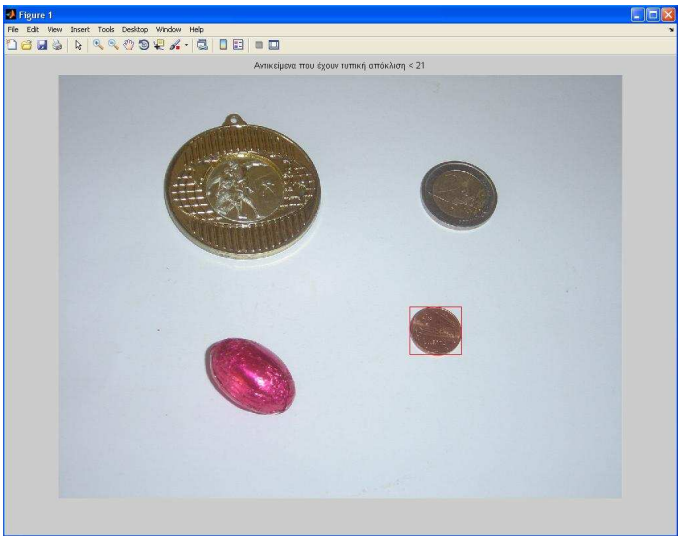


Σχήμα 9

Στην ουσία αναπαριστούμε την τιμές που υπολογίσαμε από το βήμα 8 σε άξονες.

Βήμα 10: Εντοπισμός του μικρότερου σχήματος της εικόνας

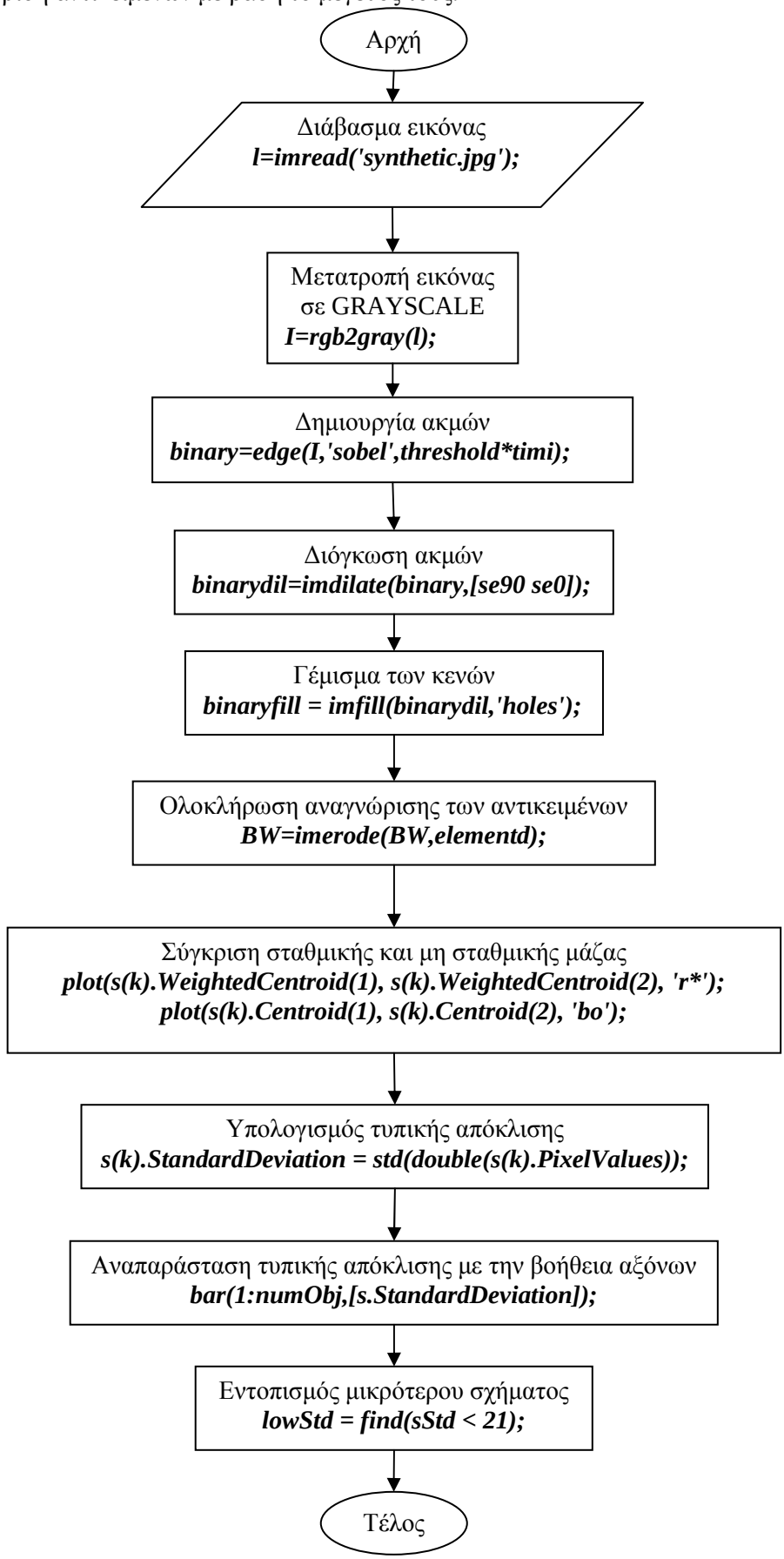
```
sStd = [s.StandardDeviation];
lowStd = find(sStd < 21);

imshow(l);
title('Αντικείμενα που έχουν τυπική απόκλιση < 21');
hold on;
for k = 1 : length(lowStd)
   rectangle('Position', s(lowStd(k)).BoundingBox, ...
      'EdgeColor','r');
end
hold off;
```



Σχήμα 10

Στο σχήμα 10 απεικονίζεται ο εντοπισμός του μικρότερου αντικειμένου. Σε αυτό μας βοήθησε η τυπική απόκλιση βάζοντας μια συνθήκη να μας εντοπίσει τα αντικείμενα που έχουν τυπική απόκλιση μικρότερη του 21.(lowStd = find(sStd < 21);).

Στο παρακάτω διάγραμμα ροής απεικονίζονται τα βήματα που εκτελέστηκαν για την αναγνώριση αντικειμένων με βάση το μέγεθος τους.



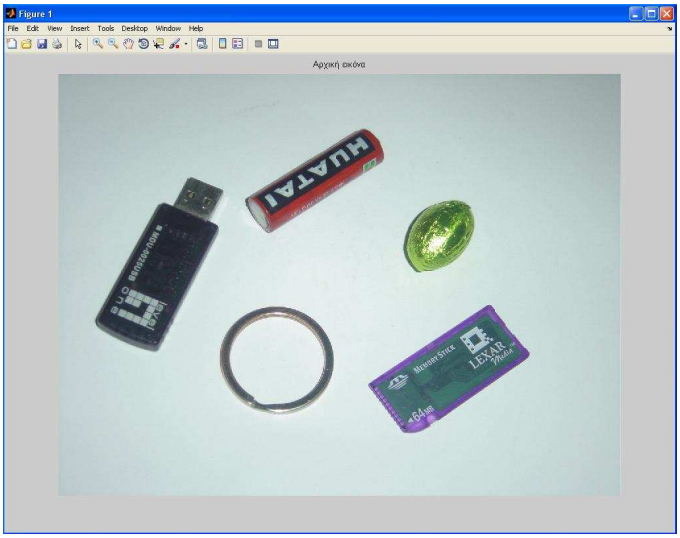
Κεφάλαιο 5

Αναγνώριση αντικειμένων με βάση την κυρτότητα τους

Περιεχόμενα

Βήμα 1: Διάβασμα εικόνας

l=imread('round2.JPG');
figure,imshow(l),title('Αρχική εικόνα');

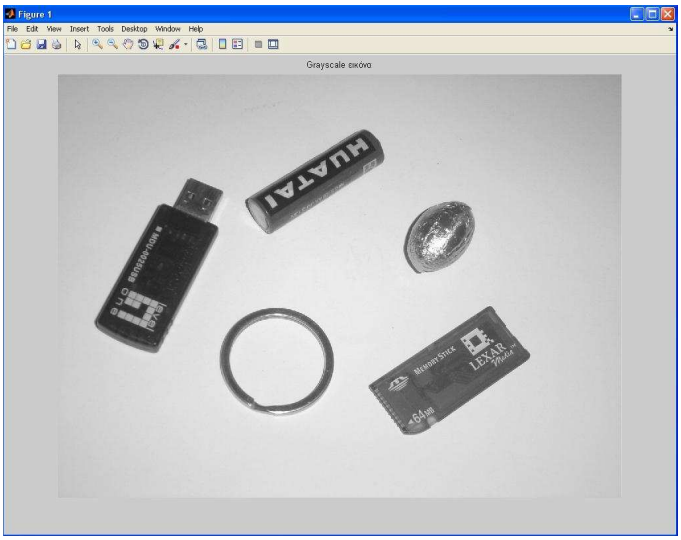


Σχήμα 1

Με την συνάρτηση ***imread*** το πρόγραμμα διαβάζει την εικόνα και με την εντολή imshow μας την απεικονίζει σε ένα παράθυρο(figure).

Βήμα 2: Μετατροπή της εικόνας από RGB σε GRAYSCALE

I=rgb2gray(I);
figure,imshow(I),title('Grayscale εικόνα');



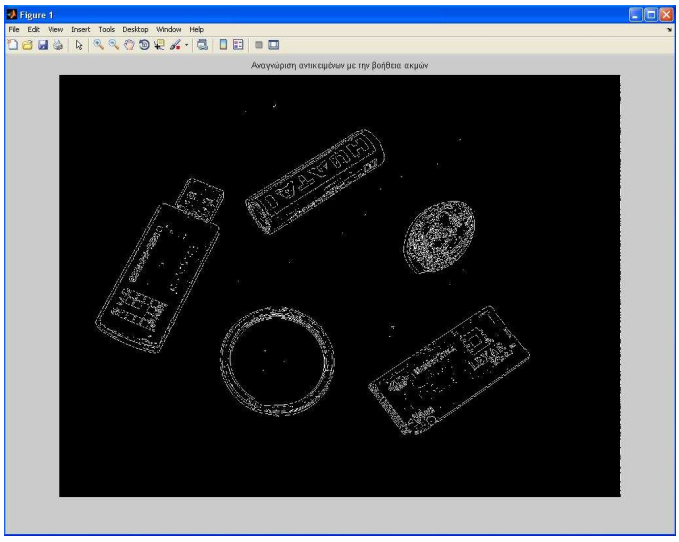
Σχήμα 2

Επειδή είναι πιο εύκολη η επεξεργασία μιας εικόνας με όσο το δυνατόν λιγότερα επίπεδα φωτεινότητας, γι'αυτό μετατρέπουμε την εικόνα σε grayscale(255 επίπεδα γκρι). Όπως φαίνεται στο σχήμα 2 για την μετατροπή της εικόνας μας βοηθάει η συνάρτηση ***rgb2gray***.

Βήμα 3: Αναγνώριση αντικειμένων με την βοήθεια ακμών

Επειδή τα αντικείμενα στην εικόνα μας έχουν διαφορετικό contrast από το background μπορούμε να αναδείξουμε τις λεπτομέρειες τους με την βοήθεια των ακμών.

[value threshold]=edge(I,'sobel');
timi=.5;
binary=edge(I,'sobel',threshold*timi);
figure, imshow(binary),title('Αναγνώριση αντικειμένων με την βοήθεια ακμών');



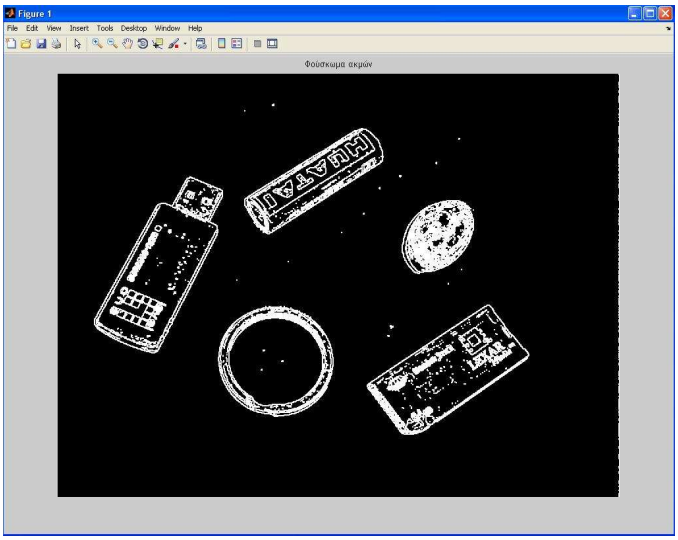
Σχήμα 3

Χρησιμοποιώντας την συνάρτηση ***edge*** και με ένα κατώφλι(threshold) της τάξεως του 0.5 δημιουργούμε μια δυαδική εικόνα όπου απεικονίζονται οι ακμές στο άσπρο χρώμα και το background στο μαύρο. Τα pixels που είχαν τιμή μεγαλύτερη του 0.5 πήραν τιμή 1(άσπρο), ενώ τα υπόλοιπα πήραν τιμή 0(μαύρο).

Βήμα 4: Διόγκωση ακμών

Για να γίνει σωστά η αναγνώριση των αντικειμένων θα πρέπει να <<φουσκώσουμε >> τις ακμές με αποτέλεσμα να πάρουμε όλο το εμβαδόν του αντικειμένου.

```
se90 = strel('line', 3, 90);
se0 = strel('line', 3, 0);
binarydil=imdilate(binary,[se90 se0]);
figure, imshow(binarydil), title('Φούσκωμα ακμών');
```



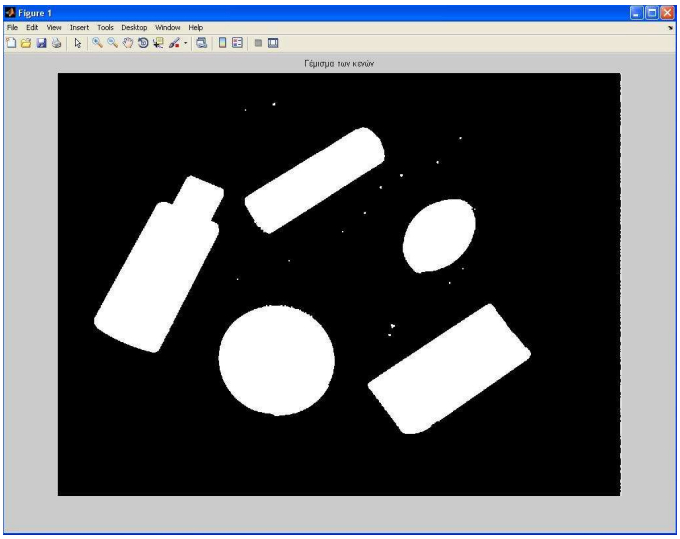
Σχήμα 4

Με την μορφολογική συνάρτηση ***strel*** παίρνουμε τις διαστάσεις των αντικειμένων και με την συνάρτηση ***imdilate*** πετυχαίνουμε την διόγκωση των ακμών όπως φαίνεται και στο σχήμα 4.

Βήμα 5: Γέμισμα των κενών

Αφού διογκώσαμε τις ακμές στο προηγούμενο βήμα αυτό που απομένει για να πάρουμε όλο το σχήμα των αντικειμένων είναι να γεμίσουμε τα μαύρα διαστήματα μέσα στα αντικείμενα.

```
binaryfill = imfill(binarydil,'holes');
figure, imshow(binaryfill),title('Γέμισμα των κενών');
```



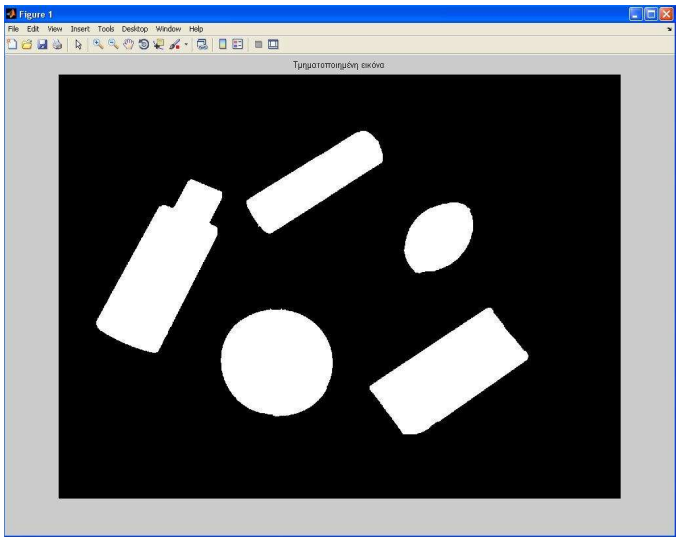
Σχήμα 5

Όπως φαίνεται στο σχήμα 5 η συνάρτηση ***imfill*** με το όρισμα holes μας βοηθάει να πετύχουμε τον σκοπό μας.

Βήμα 6: Ολοκλήρωση της αναγνώρισης των αντικειμένων

Στην εικόνα που φαίνεται στο σχήμα 5 παρατηρούμε ότι έχουν μείνει κάποια άσπρα σημεία έξω από την περιοχή του ενδιαφέροντος μας. Για να ολοκληρώσουμε την αναγνώριση των αντικειμένων πρέπει αυτά τα άσπρα σημεία να απαλειφθούν από την εικόνα μας.

```
elementd=strel('diamond',2);
bw=imerode(binarynoboard,elementd);
bw=imerode(bw,elementd);
figure,imshow(bw),title('Τμηματοποιημένη εικόνα');
```

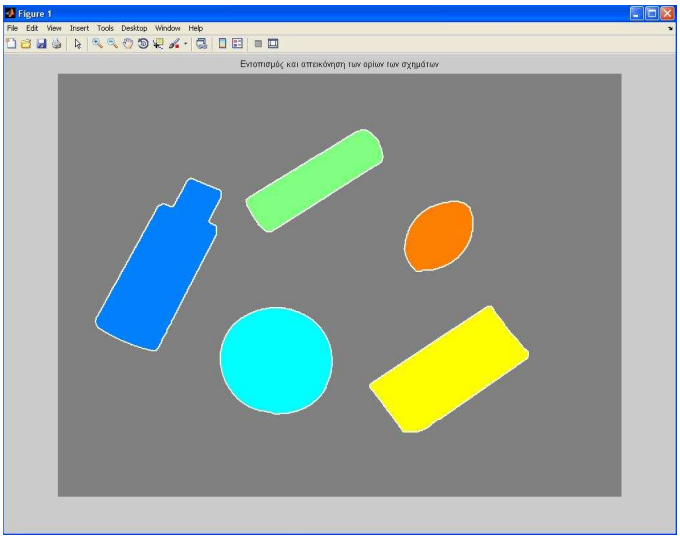


Σχήμα 6

Η συνάρτηση ***imerode*** και με την βοήθεια της ***strel*** μας δίνουν το αποτέλεσμα που βλέπουμε στο σχήμα 6.

Βήμα 7: Εντοπισμός των ορίων των αντικειμένων

```
[B,L] = bwboundaries(bw,'noholes');
% Απεικόνηση του label matrix και σχεδίαση των ορίων των σχημάτων
imshow(label2rgb(L, @jet, [.5 .5 .5]))
hold on
for k = 1:length(B)
  boundary = B{k};
  plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)
end
title('Εντοπισμός και απεικόνιση των ορίων των σχημάτων');
```



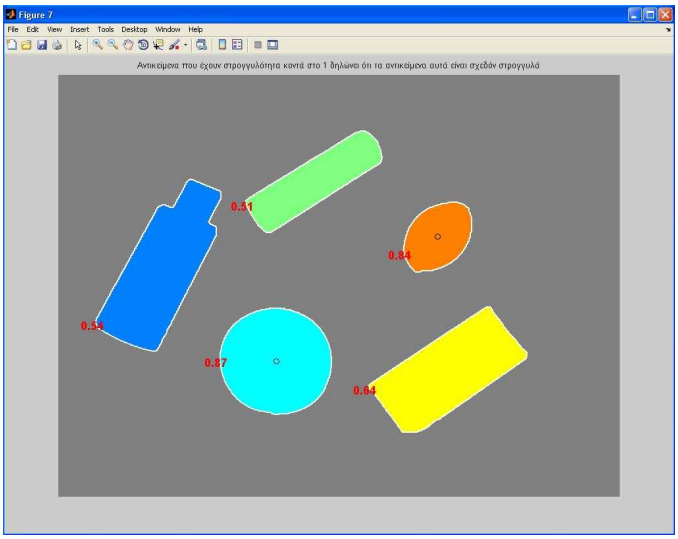
Σχήμα 7

Η συνάρτηση ***bwboundaries*** αναγνωρίζει τα όρια των αντικειμένων του σχήματος 6 και με την συνάρτηση ***label2rgb*** τοποθετείται χρώμα σε κάθε ένα από τα αντικείμενα.

Βήμα 8: Προσδιορισμός κυρτότητας για κάθε αντικείμενο

```
%% ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΠΟΥ ΕΙΝΑΙ ΣΤΡΟΓΓΥΛΑ
stats = regionprops(L,'Area','Centroid');
threshold = 0.80;
% Ένα loop για τα αντικείμενα
for k = 1:length(B)

% Λαμβάνουμε τις (X,Y) συντεταγμένες των αντικειμένων στο αντίστοιχο label 'k'
  boundary = B{k};

  % Υπολογισμός της περιμέτρου του αντικειμένου
  delta_sq = diff(boundary).^2;
  perimeter = sum(sqrt(sum(delta_sq,2)));

% Λαμβάνουμε τον υπολογισμό του area του αντικειμένου στο αντίστοιχο label 'k'
  area = stats(k).Area;

  % Υπολογίζουμε την στρογγυλότητα μα βάση τον τύπο
  metric = 4*pi*area/perimeter^2;

  % Απεικόνηση του αποτελέσματος για την στρογγυλότητα
```

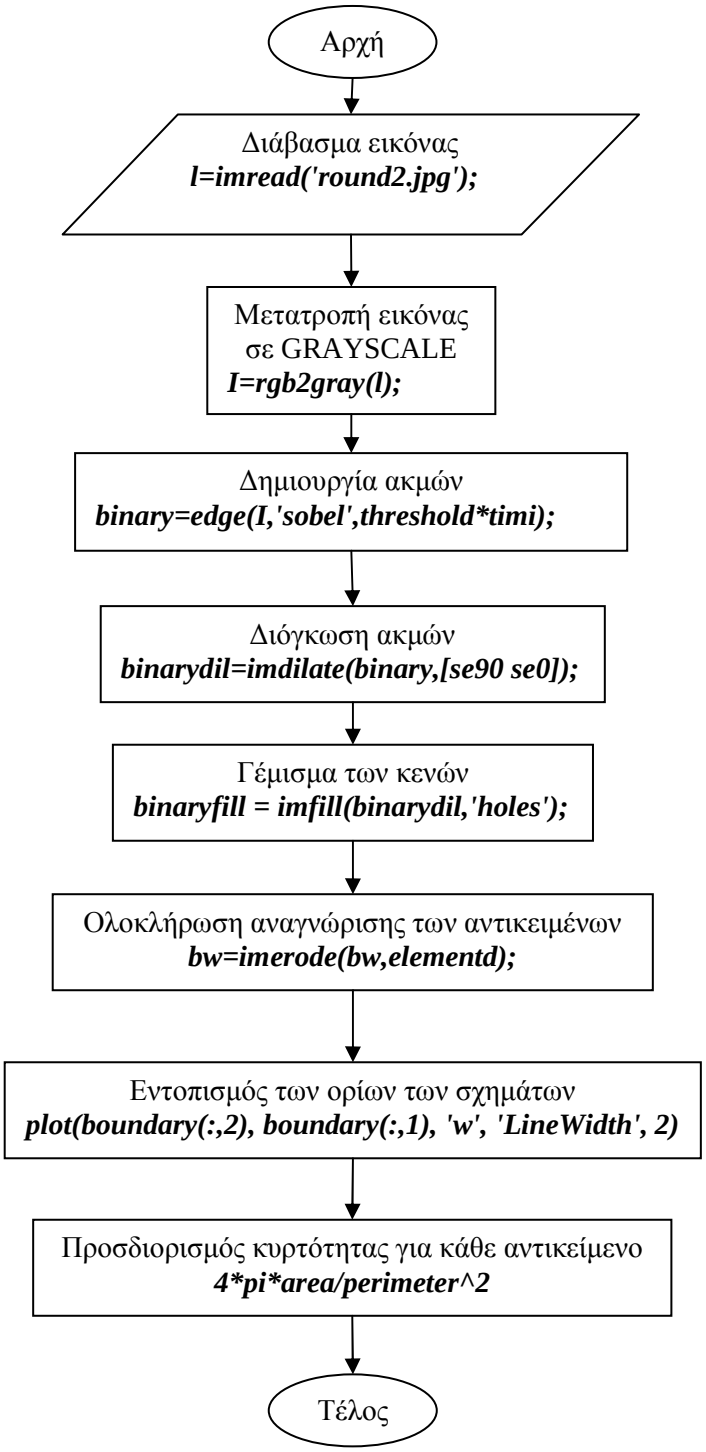
```
  metric_string = sprintf('%2.2f',metric);

% Μαρκάρισμα των αντικειμένων που πληρούν την παρακάτω συνθήκη με ένα
  % μαύρο κύκλο (Το threshold στο παράδειγμα εδώ είναι 0.80)
  if metric > threshold
    centroid = stats(k).Centroid;
    plot(centroid(1),centroid(2),'ko');
  end

text(boundary(1,2)-35,boundary(1,1)+13,metric_string,'Color','r',...
      'FontSize',14,'FontWeight','bold');

end
title(['Αντικείμενα που έχουν στρογγυλότητα κοντά στο 1',...
      ' δηλώνει ότι τα αντικείμενα αυτά είναι σχεδόν στρογγυλά']);
```



Σχήμα 8

Η συνάρτηση ***regionprops*** παίρνει τις συντεταγμένες των αντικειμένων από την εικόνα του σχήματος 7. Στην συνέχεια με την εντολή ***metric = 4*pi*area/perimeter^2*** υπολογίζουμε την κυρτότητα των αντικειμένων. Οι εντολές που απεικονίζονται παραπάνω δείχνουν αναλυτικά τα βήματα για το αποτέλεσμα του σχήματος 8.

Στο παρακάτω διάγραμμα ροής απεικονίζονται τα βήματα που εκτελέστηκαν για την αναγνώριση αντικειμένων με βάση την κυρτότητα τους.



Κεφάλαιο 6

Αναγνώριση αντικειμένων με την βοήθεια του μετασχηματισμού Watershed

Ο μετασχηματισμός watershed εντοπίζει τα περιγράμματα των αντικειμένων που είναι πιο ανυψωμένα στην εικόνα(στο foreground) και τα παρουσιάζει σαν μια επιφάνεια όπου τα φωτεινά pixel έχουν υψηλές τιμές και τα σκοτεινά pixel χαμηλές.

Η κατάτμηση μιας εικόνας χρησιμοποιώντας τον μετασχηματισμό watershed δουλεύει καλύτερα εάν αναγνωρίσουμε τα μπροστινά αντικείμενα και τις περιοχές που δηλώνουν το background σε μια εικόνα.

Ο μετασχηματισμός watershed ακολουθεί τα παρακάτω βασικά βήματα:

1. Υπολογίζει μια συνάρτηση κατάτμησης: Είναι μια εικόνα όπου οι σκοτεινές περιοχές είναι τα αντικείμενα που προσπαθούμε να αναγνωρίσουμε.

2 Υπολογισμός σημείων στο foreground: Είναι συνδεδεμένα pixel που ανήκουν στο εσωτερικό των αντικειμένων.

3. Υπολογισμός σημείων στο background: Είναι συνδεδεμένα pixel που δεν είναι κομμάτι κανενός αντικειμένου.

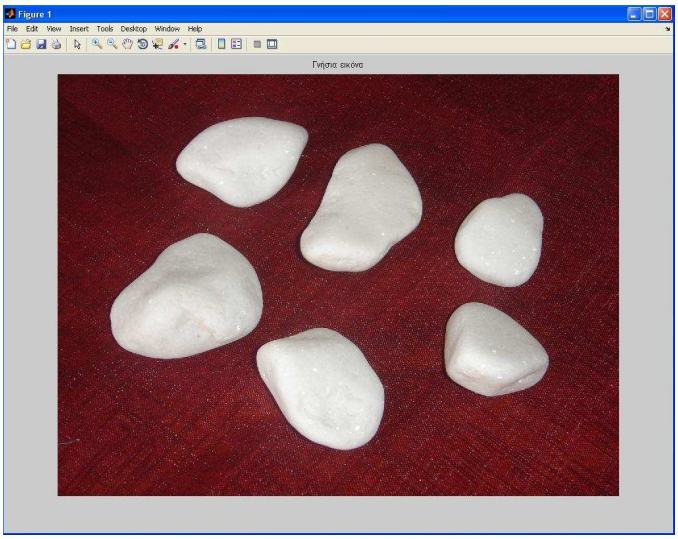
4. Μετατρέπουμε την συνάρτηση κατάτμησης έτσι ώστε να έχουμε τα ελάχιστα σημεία των pixel που ανήκουν στο background και στο foreground.

5. Εφαρμόζουμε τον μετασχηματισμό watershed στην συνάρτηση κατάτμησης που δημιουργήσαμε.

Περιεχόμενα

Βήμα 1: Διάβασμα εικόνας

```
rgb = imread('watersheed.jpg');
figure,imshow(rgb),title('Γνήσια εικόνα');
```

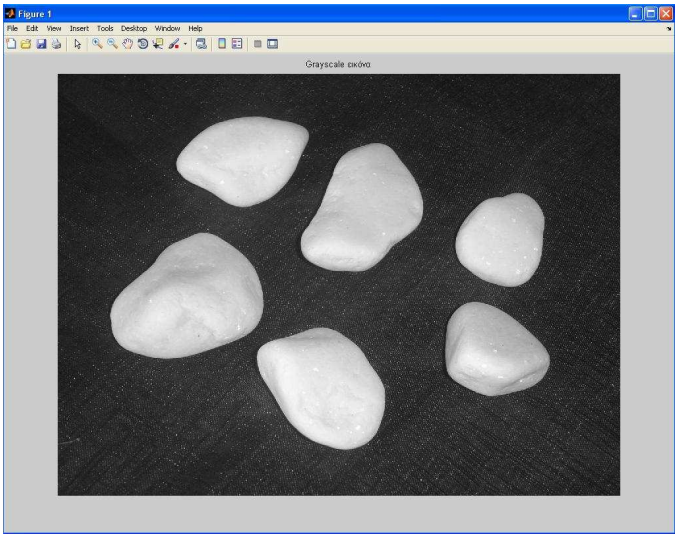


Σχήμα1

Με την συνάρτηση ***imread*** το πρόγραμμα διαβάζει την εικόνα και με την εντολή ***imshow*** μας την απεικονίζει σε ένα παράθυρο(figure).

Βήμα 2: Μετατροπή της εικόνας από RGB σε GRAYSCALE

```
I=rgb2gray(rgb);
figure,imshow(I),title('Grayscale εικόνα');
```

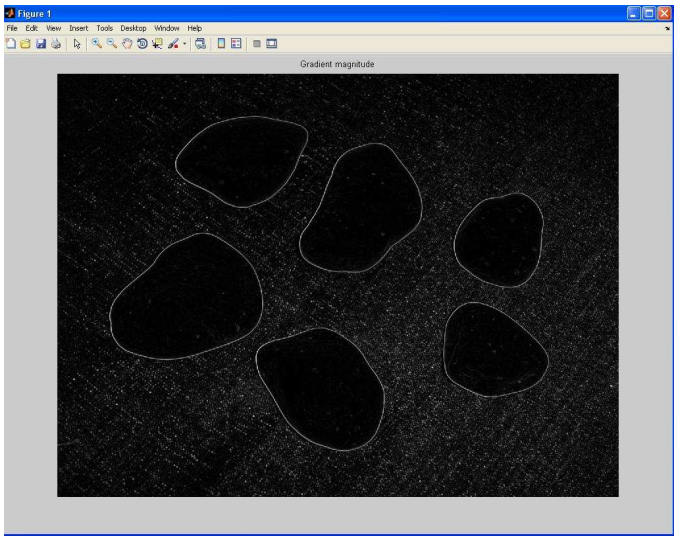


Σχήμα 2

Επειδή είναι πιο εύκολη η επεξεργασία μιας εικόνας με όσο το δυνατόν λιγότερα επίπεδα φωτεινότητας, γι'αυτό μετατρέπουμε την εικόνα σε grayscale(255 επίπεδα γκρι). Όπως φαίνεται στο σχήμα 2 για την μετατροπή της εικόνας μας βοηθάει η συνάρτηση ***rgb2gray***.

Βήμα 3: Χρήση του Gradient Magnitude σαν συνάρτηση κατάτμησης

```
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(I), hy, 'replicate');
Ix = imfilter(double(I), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);
figure, imshow(gradmag,[]), title('Gradient magnitude')
```



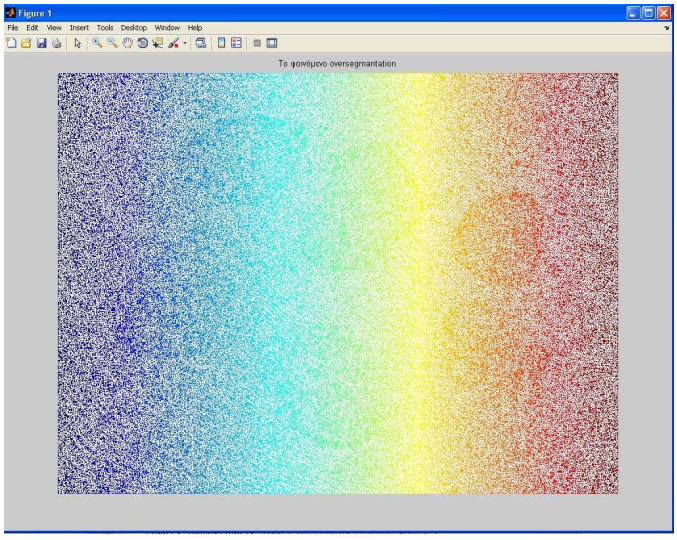
Σχήμα 3

Χρησιμοποιώντας την μάσκα του Sobel για τις ακμές, την συνάρτηση ***imfilter*** και μια αριθμητική πράξη υπολογίζουμε το gradient magnitude(κλιμακωτή μεταβολή) όπως φαίνεται στις εντολές και την εικόνα του σχήματος 3.

Το gradient magnitude είναι μεγαλύτερο στο περίγραμμα των αντικειμένων και μικρότερο στο εσωτερικό τους ,όπως βλέπουμε και στο σχήμα.

Παρατήρηση: Τοποθέτηση του watershed μετασχηματισμού *πάνω* στο gradient magnitude

L = watershed(gradmag);
Lrgb = label2rgb(L);
figure, imshow(Lrgb), title('Το φαινόμενο oversegmantation')



Σχήμα 4

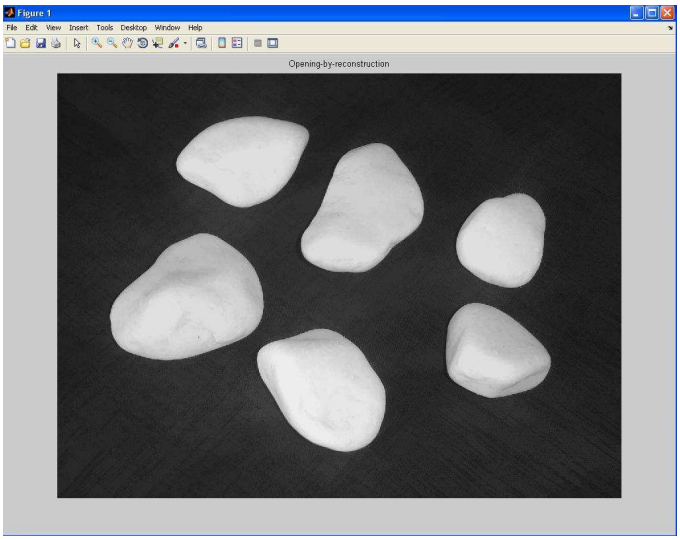
Χωρίς κάποια προεπεξεργασία δεν μπορούμε να τοποθετήσουμε τον watershed μετασχηματισμό πάνω στο gradient magnitude γιατί τότε έχουμε το φαινόμενο oversegmentation όπως φαίνεται στο σχήμα 4

Βήμα 4: Υπολογισμός σημείων στο foreground

Για να γίνει ο υπολογισμός σημείων στο foreground χρειάζεται να χρησιμοποιήσουμε μια μορφολογική τεχνική που λέγετε Opening-Closing By Reconstruction. Η τεχνική αυτή θα δημιουργήσει κάποιες επιφάνειες πάνω στα αντικείμενα που προσπαθούμε να αναγνωρίσουμε, οι οποίες ονομάζονται regional maxima.Στα παρακάτω 4 βήματα θα δείξουμε πως υλοποιείται αυτή η τεχνική και πως τοποθετούνται οι regional maxima πάνω στα αντικείμενα.

Βήμα 1: Opening By Reconstruction

se = strel('disk',20);
Ie = imerode(I, se);
Iobr = imreconstruct(Ie, I);
figure, imshow(Iobr), title('Opening-by-reconstruction')

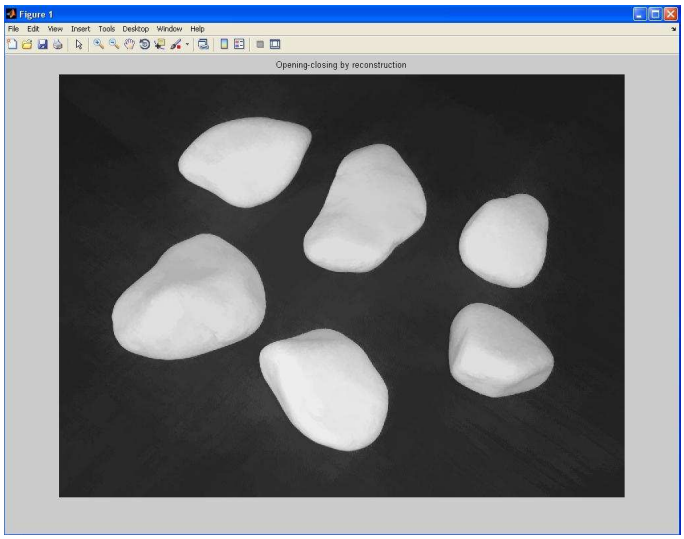


Σχήμα 5

Η τεχνική Opening By Reconstruction είναι μια διάβρωση που προήλθε από μια μορφολογική ανακατασκευή της εικόνας. Η συνάρτηση ***imreconstruct*** είναι αυτή που μας βοηθάει να υλοποιήσουμε αυτήν την τεχνική όπως φαίνεται στο σχήμα 5.

Βήμα 2: Opening-Closing By Reconstruction

Iobrd = imdilate(Iobr, se);
Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr));
Iobrcbr = imcomplement(Iobrcbr);
figure, imshow(Iobrcbr), title('Opening-closing by reconstruction')

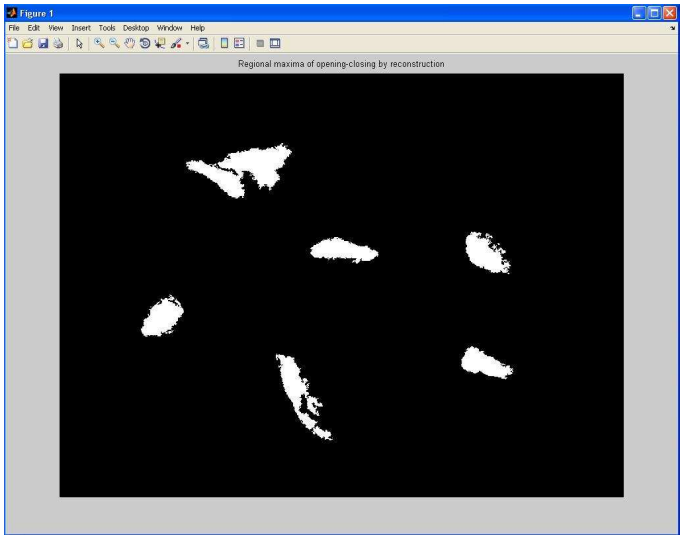


Σχήμα 6

Η συνάρτηση ***imcomplement*** είναι αυτή μου μας υλοποιεί την Opening-Closing By Reconstruction τεχνική όπως φαίνεται και στο σχήμα 6

Βήμα 3: Δημιουργία επιφανειών regional maxima για τον εντοπισμό των αντικειμένων

fgm = imregionalmax(Iobrcbr);
figure, imshow(fgm), title('Regional maxima of opening-closing by reconstruction')

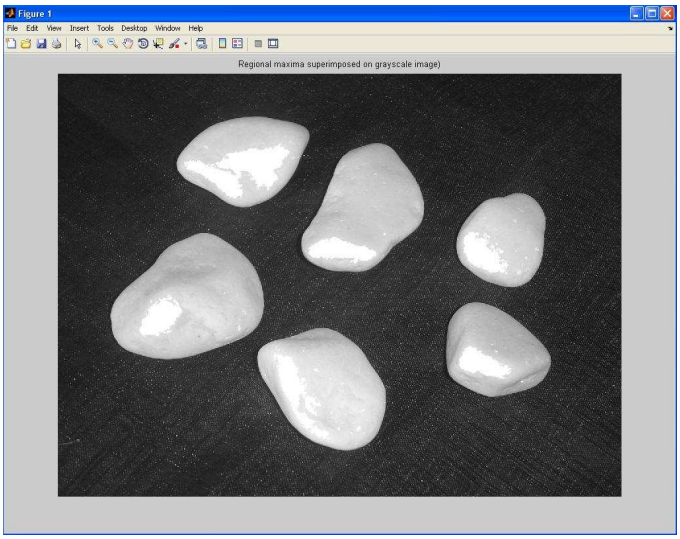


Σχήμα 7

Με την βοήθεια της συνάρτησης ***imregionalmax*** και τοποθετώντας της την τεχνική Opening-Closing By Reconstruction παίρνουμε τις regional maxima επιφάνειες

Βήμα 4: Τοποθέτηση των regional maxima επιφανειών στην αρχική μας εικόνα

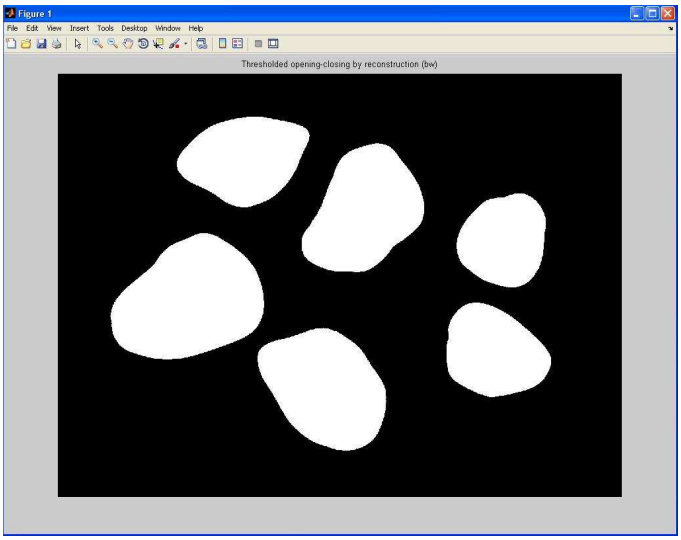
I2 = I;
I2(fgm) = 255;
figure, imshow(I2), title('Regional maxima superimposed on grayscale image)')



Σχήμα 8

Βήμα 5: Υπολογισμός σημείων στο background

bw = im2bw(Iobrcbr, graythresh(Iobrcbr));
figure, imshow(bw), title('Thresholded opening-closing by reconstruction (bw)')



Σχήμα 9

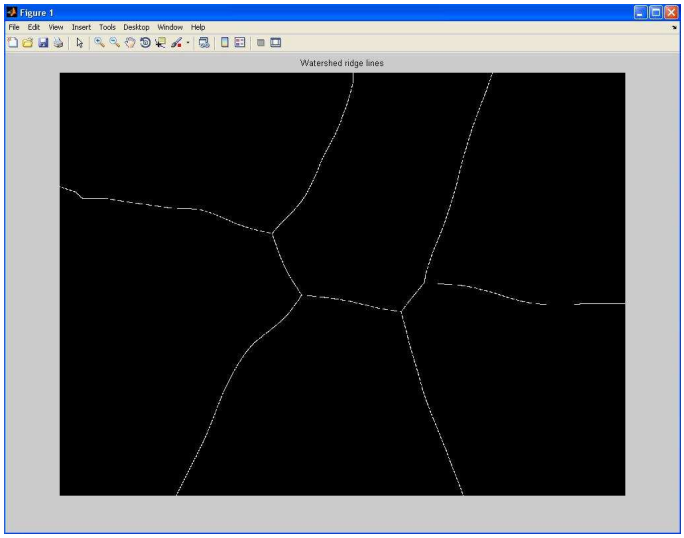
Στην συνάρτηση ***im2bw*** τοποθετούμε την εικόνα που δημιουργήσαμε στο σχήμα 7 με αποτέλεσμα να παίρνουμε το αποτέλεσμα που φαίνεται παραπάνω. Η συνάρτηση ***im2bw*** λειτουργεί με την λογική ότι τα μαύρα pixel ανήκουν στο background.

Βήμα 6: Μετατροπή της συνάρτησης κατάτμησης έτσι ώστε να έχουμε τα ελάχιστα σημεία των pixel που ανήκουν στο background και στο foreground.

Επειδή όμως δεν θέλουμε τα μαύρα pixels να είναι τόσο κοντά στα όρια των αντικειμένων μας που προσπαθούμε να αναγνωρίσουμε, υπολογίζουμε την σκελετική περιοχή του foreground όπου βρίσκονται τα αντικείμενα.

Για να γίνει αυτό υπολογίζουμε τον μετασχηματισμό απόστασης(distance transform) με την συνάρτηση ***bwdist*** και εν συνεχεία εισάγουμε το αποτέλεσμα αυτό στον μετασχηματισμό watershed**.** Το παρακάτω σχήμα εξηγεί καλύτερα τον τρόπο με τον οποίο υπολογίζεται η σκελετική περιοχή του foreground**.**

D = bwdist(bw);
DL = watershed(D);
bgm = DL == 0;
figure, imshow(bgm), title('Watershed ridge lines')



Σχήμα 10

Βήμα 7: Εφαρμογή του μετασχηματισμού watershed στην συνάρτηση κατάτμησης

Η συνάρτηση ***imimposemin*** χρησιμοποιείται για να τροποποιήσει το gradient magnitude(σχήμα 3), έτσι ώστε οι επιφάνειες regional maxima να τοποθετηθούν στις σωστές θέσεις στο foreground όπου βρίσκονται τα αντικείμενα που θέλουμε να αναγνωρίσουμε.

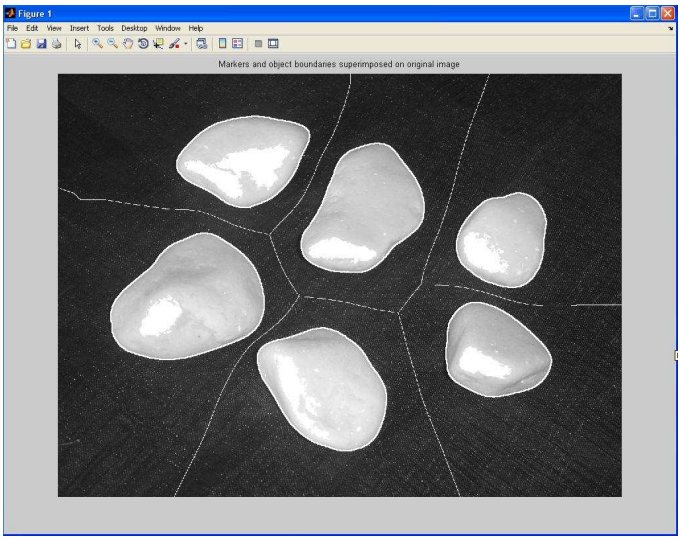
gradmag2 = imimposemin(gradmag, bgm | fgm);

Τώρα πλέον είμαστε έτοιμοι να τοποθετήσουμε τον μετασχηματισμό watershed στο gradient magnitude.

L = watershed(gradmag2);

Βήμα 8: Απεικόνιση του αποτελέσματος

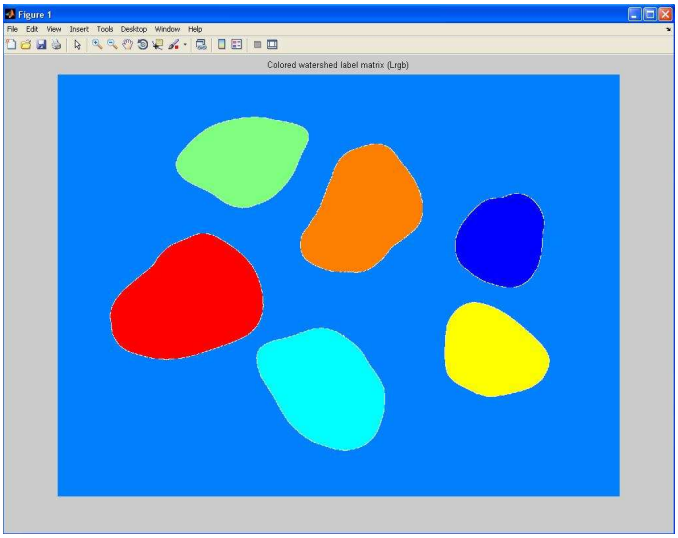
```
I4 = I;
I4(imdilate(L == 0, ones(3, 3)) | bgm | fgm) = 255;
figure, imshow(I4)
title('Markers and object boundaries superimposed on original image')
```



Σχήμα 11

Μία τεχνική απεικόνισης είναι να εμφανιστούν το foreground, το background, και το gradient magnitude μαζί στην αρχική μας εικόνα. Η συνάρτηση ***imdilate*** μας βοηθάει να υλοποιήσουμε την εικόνα του σχήματος 11.

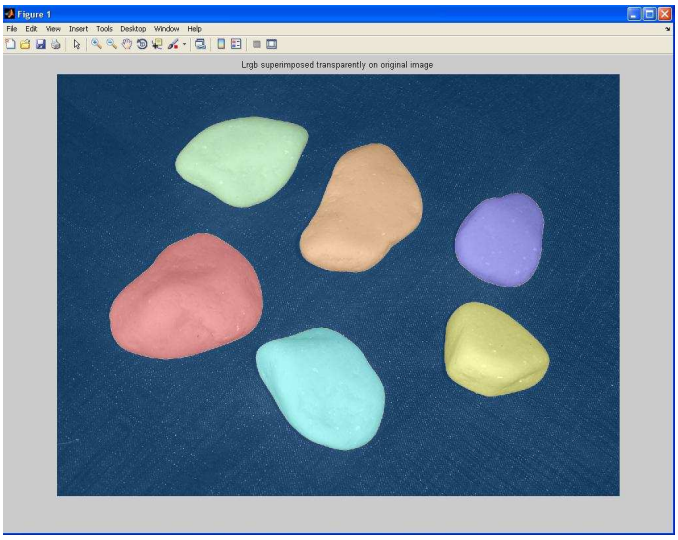
```
Lrgb = label2rgb(L, 'jet', 'w', 'shuffle');
figure, imshow(Lrgb)
title('Colored watershed label matrix (Lrgb)')
```



Σχήμα 12

Στο σχήμα 12 φαίνεται άλλη μία τεχνική απεικόνισης χρησιμοποιώντας το label matrix. Για να χρησιμοποιήσουμε το label matrix κάνουμε χρήση της συνάρτησης ***label2rgb***.

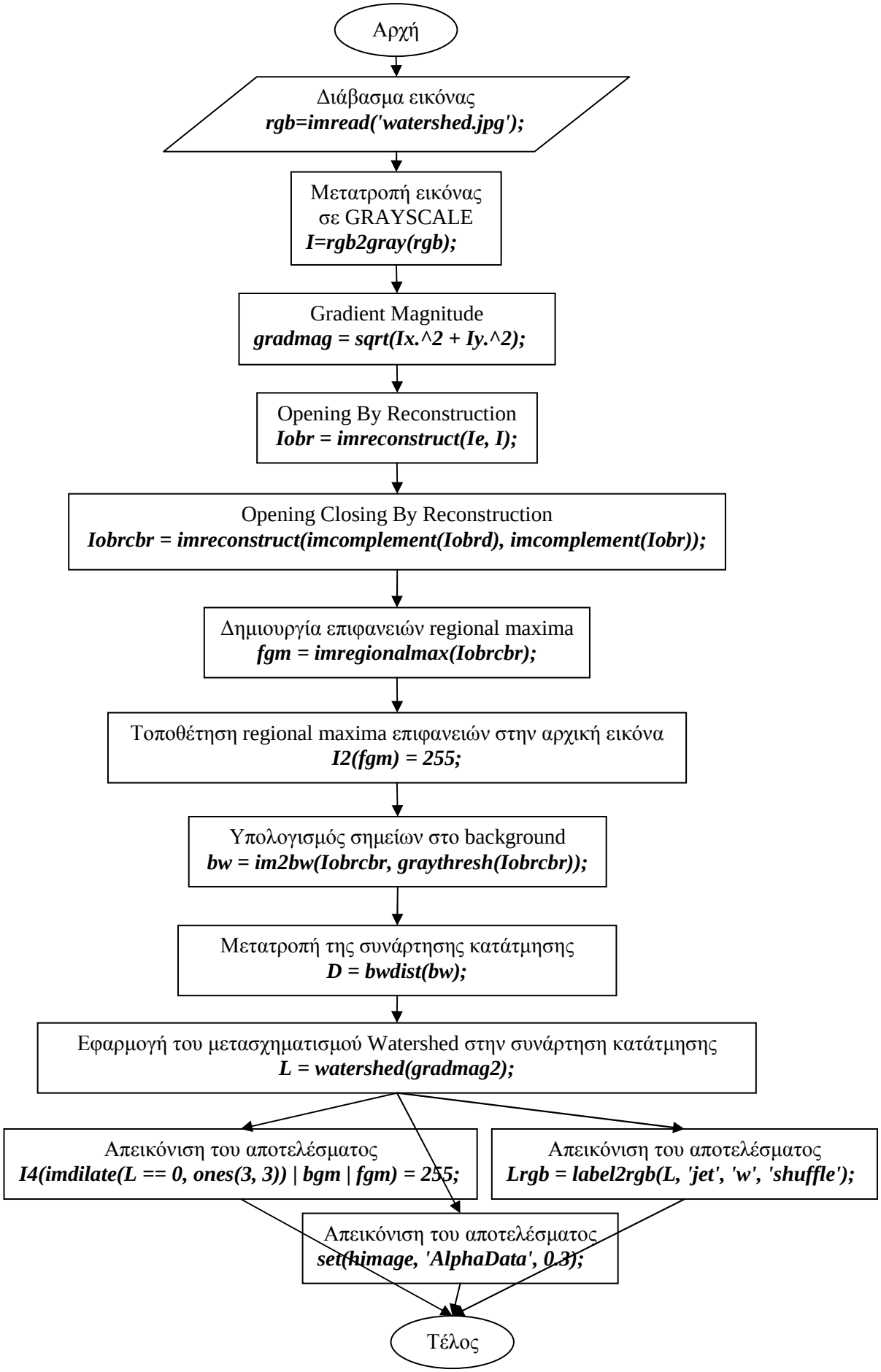
figure, imshow(I), hold on
himage = imshow(Lrgb);
set(himage, 'AlphaData', 0.3);
title('Lrgb superimposed transparently on original image')



Σχήμα 13

Τέλος μπορούμε να τοποθετήσουμε το label matrix που δημιουργήσαμε στο σχήμα 12 στην αρχική μας εικόνα. Έτσι έχουμε το αποτέλεσμα του σχήματος 13.

Στο παρακάτω διάγραμμα ροής απεικονίζονται τα βήματα που εκτελέστηκαν για την αναγνώριση αντικειμένων με την βοήθεια του μετασχηματισμού Watershed.


Αρχή
Διάβασμα εικόνας
rgb=imread('watershed.jpg');
Μετατροπή εικόνας σε GRAYSCALE
I=rgb2gray(rgb);
Gradient Magnitude
gradmag = sqrt(Ix.^2 + Iy.^2);
Opening By Reconstruction
Iobr = imreconstruct(Ie, I);
Opening Closing By Reconstruction
Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr));
Δημιουργία επιφανειών regional maxima
fgm = imregionalmax(Iobrcbr);
Τοποθέτηση regional maxima επιφανειών στην αρχική εικόνα
I2(fgm) = 255;
Υπολογισμός σημείων στο background
bw = im2bw(Iobrcbr, graythresh(Iobrcbr));
Μετατροπή της συνάρτησης κατάτμησης
D = bwdist(bw);
Εφαρμογή του μετασχηματισμού Watershed στην συνάρτηση κατάτμησης
L = watershed(gradmag2);
Απεικόνιση του αποτελέσματος
I4(imdilate(L == 0, ones(3, 3)) | bgm | fgm) = 255;
Απεικόνιση του αποτελέσματος
Lrgb = label2rgb(L, 'jet', 'w', 'shuffle');
Απεικόνιση του αποτελέσματος
set(himage, 'AlphaData', 0.3);
Τέλος

Κεφάλαιο 7

Αναγνώριση αντικειμένων μέσα από βίντεο

Στόχος μας είναι να γίνει η αναγνώριση αντικειμένων μέσα από ένα βίντεο και όχι από μια στατική εικόνα. Για να το επιτύχουμε αυτό θα πάρουμε ένα frame από το βίντεο θα υλοποιήσουμε τον αλγόριθμο για να αναγνωρίσουμε τα αντικείμενα σε αυτό το frame και εν συνεχεία θα εφαρμόσουμε τον αλγόριθμο σε ολόκληρο το βίντεο.

Περιεχόμενα

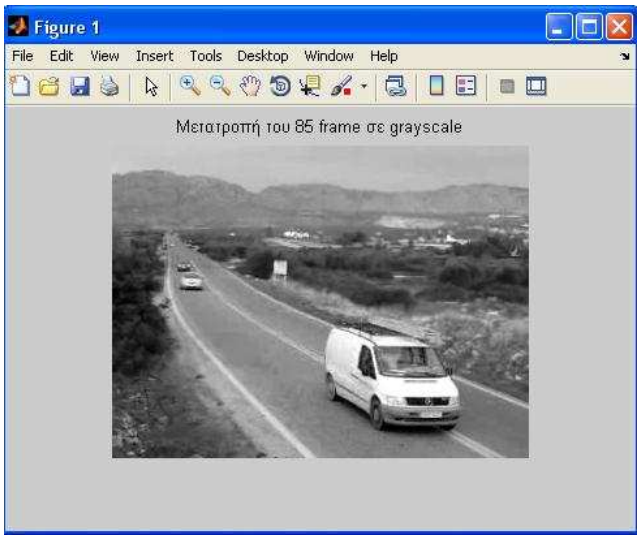
Βήμα 1: Διάβασμα βίντεο

video = mmreader('road1.avi')

Με την συνάρτηση ***mmreader*** διαβάζουμε το βίντεο συλλέγοντας επιπλέον πληροφορίες όπως η διάρκεια του σε δευτερόλεπτα, τα συνολικά frames που έχει κ.α.

Βήμα 2: Επιλογή frame και μετατροπή του σε GRAYSCALE εικόνα

```
l=read(video,85);
thisFrame = rgb2gray(l);
figure,imshow(thisFrame),title('Μετατροπή του 85 frame σε grayscale');
```

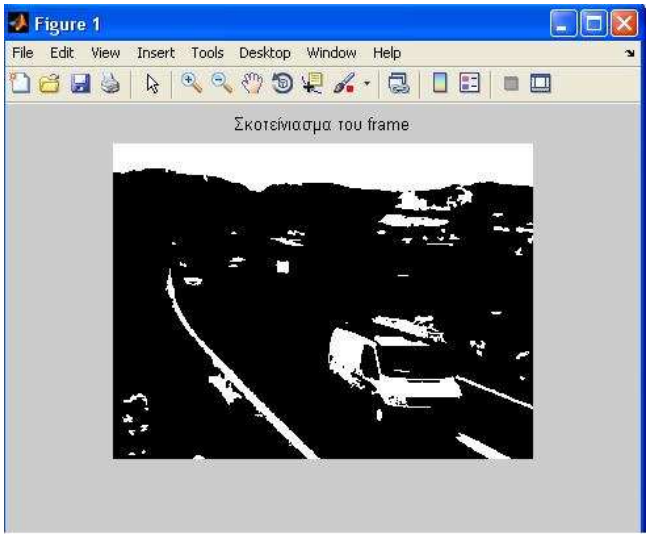


Σχήμα 1

Σε αυτό το βήμα επιλέξαμε το 85 frame από το βίντεο και το μετατρέψαμε σε μια GRAYSCALE εικόνα.

Βήμα 3: <<Σκοτείνιασμα>> της εικόνας

darkValue = 50;
noDarkFrame = imextendedmax(thisFrame,darkValue);
figure, imshow(noDarkFrame),title('Σκοτείνιασμα του frame');

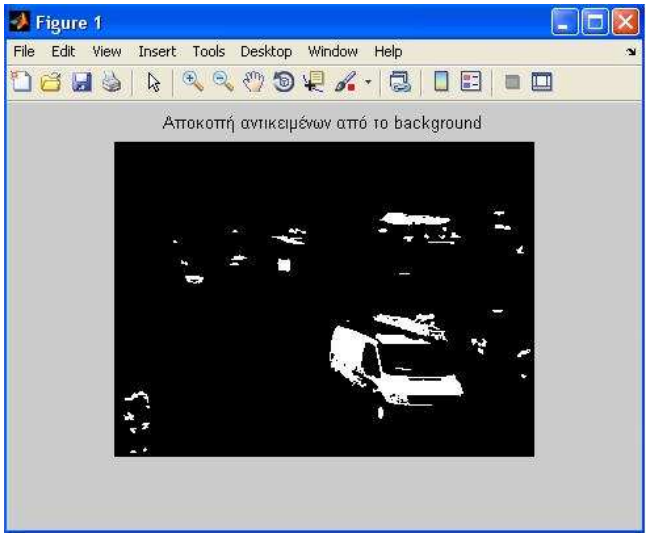


Σχήμα 2

Σε αυτό το βήμα ορίζουμε ένα threshold(κατώφλι)=50. Η συνάρτηση ***imextendedmax*** μας επιστρέφει μια δυαδική εικόνα όπου τα pixels που έχουν τιμή μικρότερη του 50 γίνονται 0(μαύρο) και τα υπόλοιπα γίνονται 1(άσπρο). Έτσι παίρνουμε το αποτέλεσμα που φαίνεται στο σχήμα 2.

Βήμα 4: Αποκοπή περιοχών που ακουμπούν στα άκρα της εικόνας

cutBoard=imclearborder(noDarkFrame,26);
figure,imshow(cutBoard),title('Αποκοπή αντικειμένων από το background');



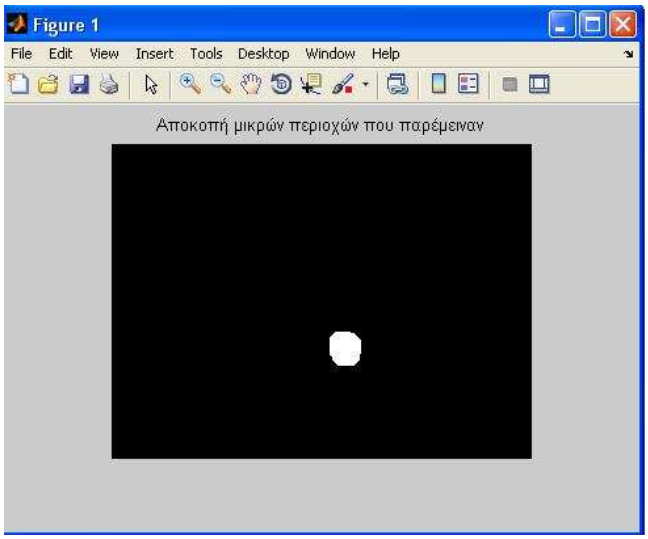
Σχήμα 3

Επειδή η λογική που θα ακολουθήσουμε είναι η άσπρη περιοχή που μένει στην εικόνα να είναι το αμάξι που θέλουμε να αναγνωρίσουμε πρέπει να απαλείψουμε τον ουρανό που

φαίνεται στο σχήμα 2. Αυτό το επιτυγχάνουμε με την συνάρτηση ***imclearborder*** και το αποτέλεσμα φαίνεται στο σχήμα 3.

Βήμα 5: Χρήση μορφολογικής συνάρτησης imopen

```
sedisk = strel('disk',10);
noSmallStructures = imopen(cutBoard, sedisk);
figure,imshow(noSmallStructures),title('Αποκοπή μικρών περιοχών που παρέμειναν');
```



Σχήμα 4

Η συνάρτηση ***imopen*** παίρνει τις άσπρες περιοχές που φαίνονται στο σχήμα 3 και εξαλείφει αυτές που δεν έχουν μορφή δίσκου μεγέθους 10(***sedisk = strel('disk',10);***). Το 10 σημαίνει να κρατήσει όλες τις περιοχές μορφής δίσκου που έχουν εμβαδό 10 και τις υπόλοιπες να τις κάνει ίσες με 0(μαύρο). Έτσι το αποτέλεσμα φαίνεται στο σχήμα 4.

Βήμα 6: Εισαγωγή αλγορίθμου στο βίντεο

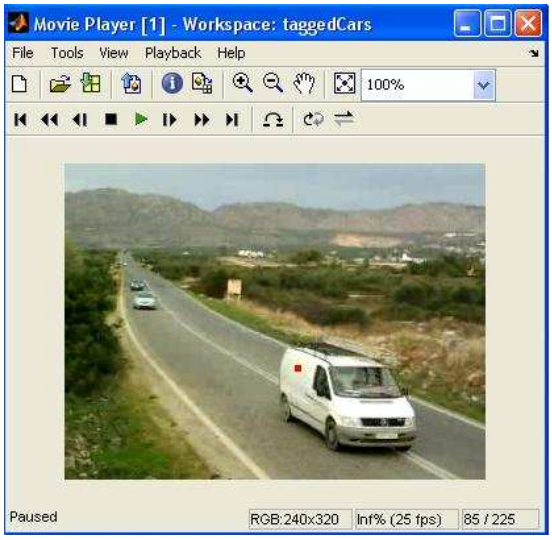
```
nframes = get(video, 'NumberOfFrames');
I = read(video, 1);
taggedCars = zeros([size(I,1) size(I,2) 3 nframes], class(I));

for k = 1 : nframes
   singleFrame = read(video, k);

   % Μετατροπή σε grayscale για την μορφολογική επεξεργασία.
   I = rgb2gray(singleFrame);

   % Σκοτείνιασμα των frames.
   noDarkFrames = imextendedmax(I, darkValue);

   % Αποκοπή περιοχών που ακουμπούν στο backround
   cutBoard = imclearborder(noDarkFrames,26);

   % Αποκοπή γραμμών και άλλων αντικειμένων που δεν έχουν μορφή δίσκου.
   noSmallStructures = imopen(cutBoard, sedisk);
```

```
    % Παίρνουμε την περιοχή και το κέντρο των αντικειμένων που έμειναν
    % στο frame. Τα αντικείμενα με την μεγαλύτερη περιοχή είναι
    % τα αυτοκίνητα. Δημιουργούμε ένα αντίγραφο του αρχικού μας frame και
    % στην ουσία μαρκάρουμε το αυτοκίνητο αλλάζοντας το χρώμα του
    % κεντρικού pixel σε κόκκινο.
    taggedCars(:,:,:,k) = singleFrame;

    stats = regionprops(noSmallStructures, {'Centroid','Area'});
    if ~isempty([stats.Area])
        areaArray = [stats.Area];
        [junk,idx] = max(areaArray);
        c = stats(idx).Centroid;
        c = floor(fliplr(c));
        width = 2;
        row = c(1)-width:c(1)+width;
        col = c(2)-width:c(2)+width;
        taggedCars(row,col,1,k) = 255;
        taggedCars(row,col,2,k) = 0;
        taggedCars(row,col,3,k) = 0;
    end
end
```

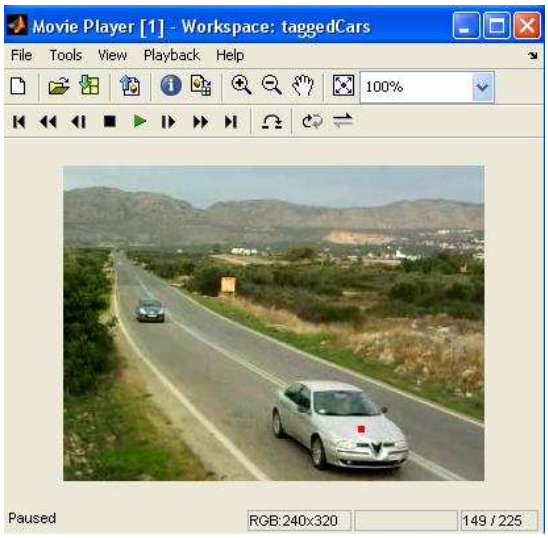
Τα βήματα 1 έως 5 τα θεωρούμε σαν ένα αλγόριθμο για τον εντοπισμό των αντικειμένων που θέλουμε. Επόμενο βήμα είναι να εισάγουμε το αλγόριθμο αυτό σε όλα τα frames του βίντεο έτσι ώστε να εντοπίσουμε όλα τα αντικείμενα του βίντεο

Βήμα 7: Απεικόνιση του αποτελέσματος

```
frameRate = get(video,'FrameRate');
implay(taggedCars,frameRate);
```



Σχήμα 5

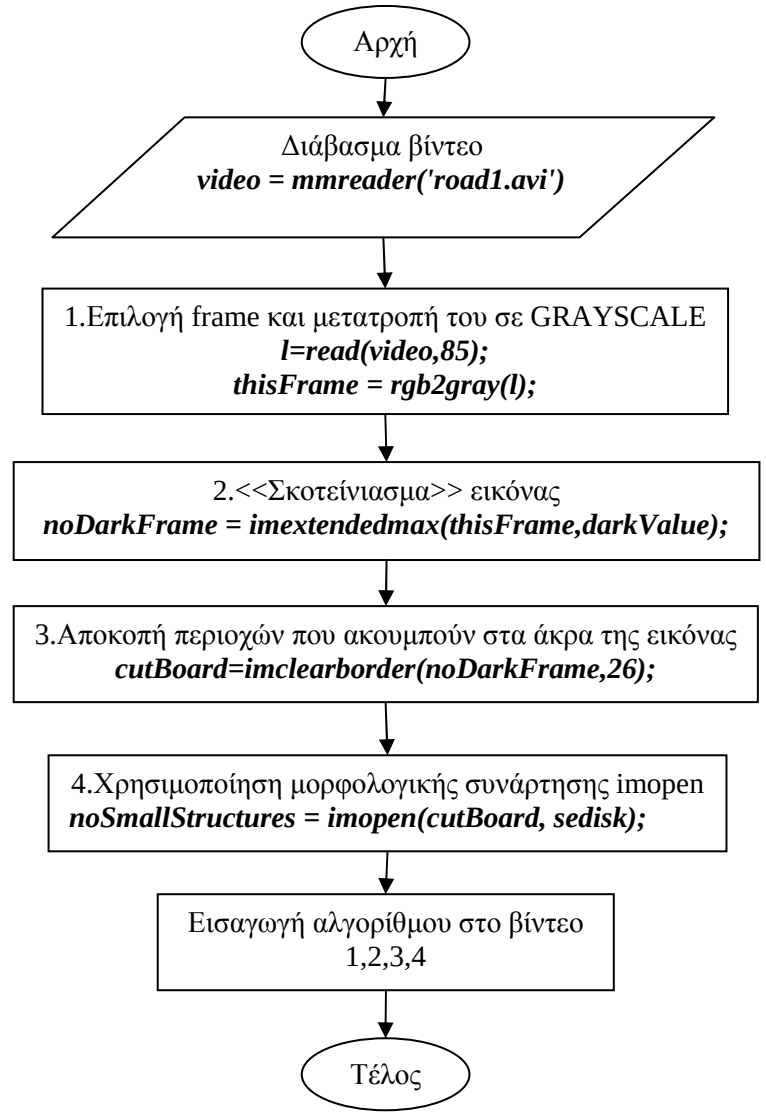


Σχήμα 6

Αφού εκτελέσουμε τον κώδικα του βήματος 6 το μόνο που μένει είναι να δούμε το αποτέλεσμα. Με την συνάρτηση ***implay*** μπορούμε να παρακολουθήσουμε το βίντεο να παίζει. Όπως είπαμε παραπάνω οι άσπρες περιοχές που θα παραμείνουν μετά από την εισαγωγή του αλγορίθμου σε κάθε frame θα είναι το αμάξι που θέλουμε να εντοπίσουμε.

Για να φανεί καλύτερα στο μάτι ο εντοπισμός των αμαξιών αυτό που κάνουμε είναι να αλλάζουμε το κεντρικό pixel της άσπρης περιοχής σε κόκκινο και έτσι έχουμε το αποτέλεσμα του σχήματος 5 και 6. Τα σχήματα 5 και 6 είναι 2 φωτογραφίες κατά την διάρκεια αναπαραγωγής του βίντεο

Στο παρακάτω διάγραμμα ροής απεικονίζονται τα βήματα που εκτελέστηκαν για την αναγνώριση αντικειμένων μέσα από βίντεο.



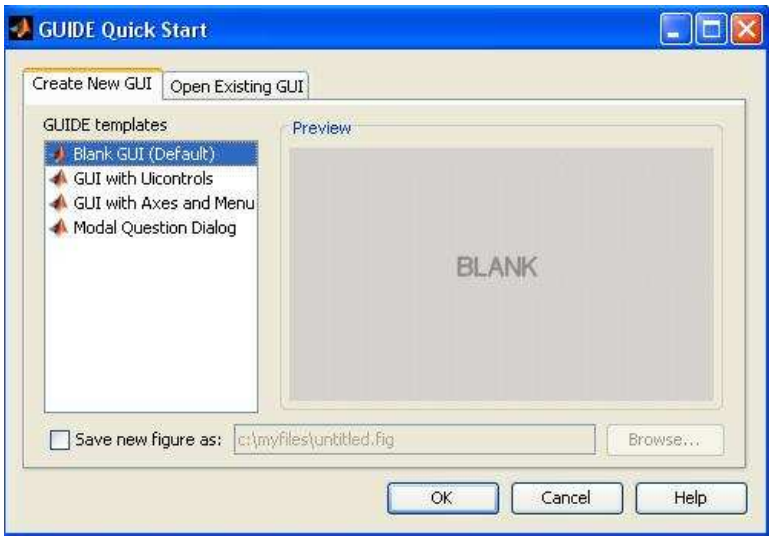
Κεφάλαιο 8

Υπολογισμός κίνησης αυτοκινήτων μέσω εφαρμογής GUI(GRAPHICAL USER INTERFACE)

Στόχος μας είναι να δημιουργήσουμε ένα γραφικό περιβάλλον με το οποίο ο χρήστης θα έχει την δυνατότητα να δει αναλυτικά τα βήματα υλοποίησης ενός αλγορίθμου για τον υπολογισμό της κίνησης αυτοκινήτων από ένα βίντεο χωρίς να έχει απαραίτητες γνώσεις κώδικα και του Matlab. Για την υλοποίηση αυτής της εφαρμογής χρησιμοποίησα την έκδοση Matlab R2010b.

Περιεχόμενα

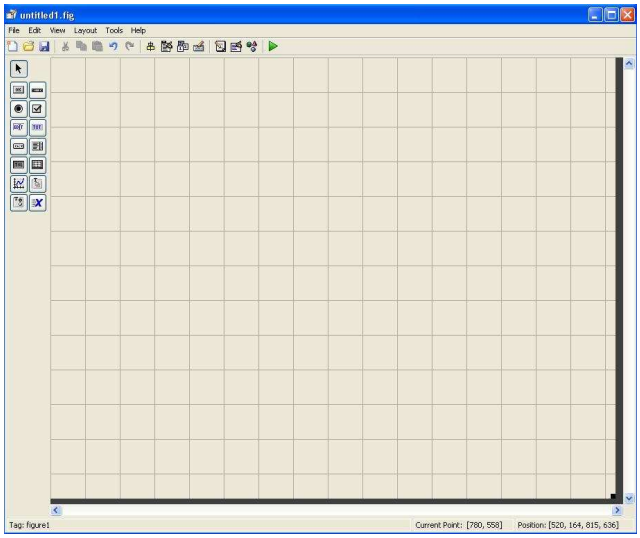
Βήμα 1: Δημιουργία GUI μέσω Matlab



Σχήμα 1

Στην γραμμή εργαλείων του Matlab κάνουμε κλικ στο κουμπί GUIDE και μας εμφανίζεται το παράθυρο του σχήματος 1. Εναλλακτικά μπορούμε να πάμε στο παράθυρο Command Window του Matlab και να πληκτρολογήσουμε την λέξη GUIDE. Για να συνεχίσουμε επιλέγουμε το Blank GUI και πατάμε OK.

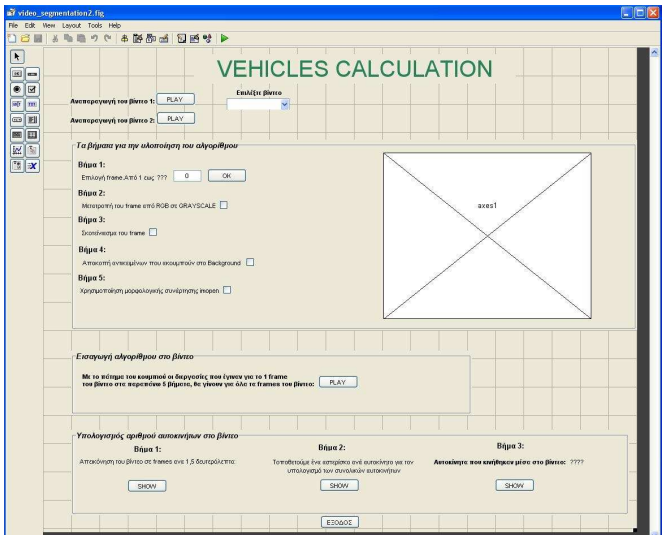
Βήμα 2: Σχεδιασμός του GUI



Σχήμα 2

Αφού επιλέξαμε το Blank GUI και πατήσαμε OK από το προηγούμενο βήμα στην οθόνη μας εμφανίζεται το παράθυρο του σχήματος 2. Στην αριστερή πλευρά του παραθύρου μας υπάρχει μια εργαλειοθήκη όπου μπορούμε με drag & drop να εισάγουμε διάφορα κουμπιά για τον σχεδιασμό της εφαρμογής μας.

Βήμα 3: Απεικόνιση του GUI



Σχήμα 3

Στο σχήμα 3 παρατηρούμε το τελικό αποτέλεσμα που δημιουργήσαμε με το drag & drop της εργαλειοθήκης. Σε κάθε αντικείμενο που τοποθετούμε μπορούμε με διπλό κλικ να πειράξουμε κάποιες ιδιότητες τους. Μερικές από αυτές είναι να αλλάξουμε το title, το χρώμα, την γραμματοσειρά τους κ.α. Επίσης υπάρχει και η ιδιότητα tag στην οποία δίνουμε κάποιο όνομα με το οποίο αναφερόμαστε στο αντικείμενο όταν εισάγουμε κάποιο κώδικα. Τα αντικείμενα που τοποθετήσαμε στην εφαρμογή είναι τα εξής:

1. Static Text: Είναι οι λεζάντες με τις οποίες δίνουμε οδηγίες στην εφαρμογή
2. Pushbutton: Είναι τα κουμπιά με τα οποία εκτελούνται κάποιες διεργασίες
3. Pop-up Menu: Είναι το αντικείμενο με το οποίο επιλέγουμε το βίντεο που θα γίνει η επεξεργασία

4. Edit Text: Είναι το αντικείμενο στο οποίο εισάγουμε το frame που θέλουμε να γίνει η επεξεργασία
5. Checkbox: Είναι τα κουμπιά στα οποία βάζουμε ένα tick για να εκτελεστεί κάποια διεργασία
6. Axes: Είναι ένα σύστημα αξόνων στο οποίο θα εμφανίζονται τα αποτελέσματα των διεργασιών που θα εκτελούνται κάθε φορά

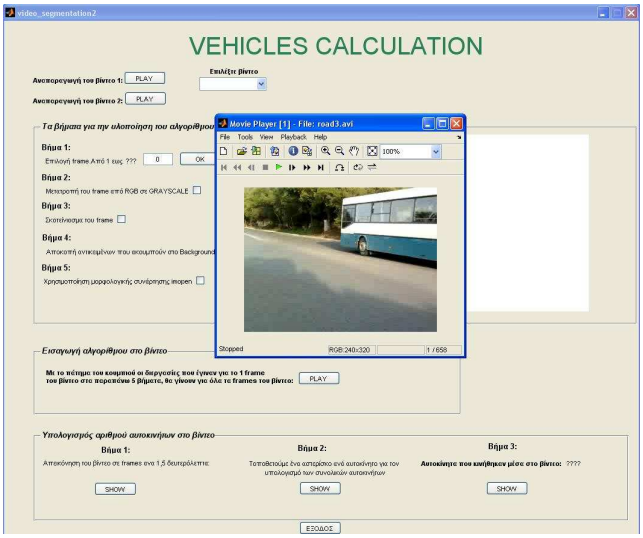
Αφού διαμορφώσουμε την εφαρμογή μας έτσι όπως θέλουμε τότε την αποθηκεύουμε δίνοντας της ένα όνομα. Στο συγκεκριμένο παράδειγμα το όνομα που δόθηκε είναι video_segmentation. Με την αποθήκευση της εφαρμογής δημιουργούνται αυτόματα δύο αρχεία. Ένα αρχείο που έχει επέκταση fig και είναι το αρχείο που φαίνεται στο σχήμα 3 και ένα αρχείο που έχει επέκταση m στο οποίο αναγράφεται μόνο ο κώδικας που δημιουργήθηκε από την εφαρμογή.

Στο m file που δημιουργήθηκε υπάρχουν και κάποιες συναρτήσεις οι οποίες αντιστοιχούν στο κάθε αντικείμενο που δημιουργήσαμε και έχουν όνομα ίδιο με το όνομα που τοποθετήσαμε στην ιδιότητα tag κάθε αντικειμένου. Επομένως μπορούμε να εισάγουμε κώδικα σε αυτές τις συναρτήσεις για να κάνουν κάποιες διεργασίες όταν τις επιλέξουμε από την εφαρμογή π.χ. πατώντας ένα κουμπί.

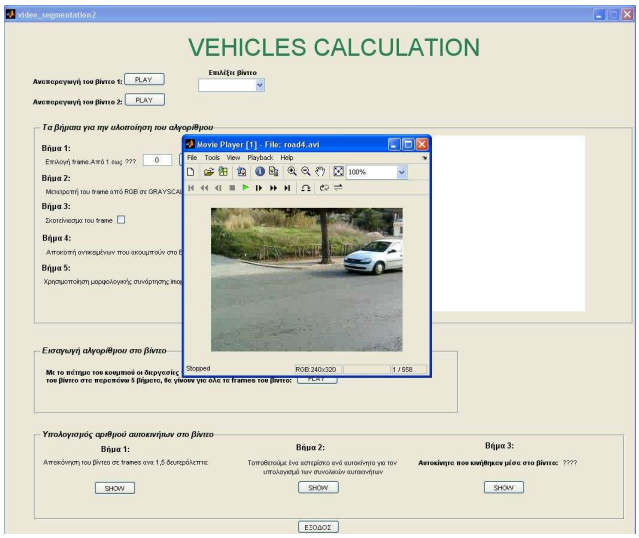
Βήμα 4: Εισαγωγή κώδικα σε κάθε αντικείμενο της εφαρμογής

1.Pushbuttons Αναπαραγωγή του βίντεο 1 και 2

```
function play_pushbutton_Callback(hObject, eventdata, handles)
% Διάβασμα βίντεο και αναπαραγωγή του με την implay
video = mmreader('road3.avi');
implay('road3.avi');

function play2_pushbutton_Callback(hObject, eventdata, handles)
% Διάβασμα βίντεο και αναπαραγωγή του με την implay
video = mmreader('road4.avi');
implay('road4.avi');
```



Σχήμα 4



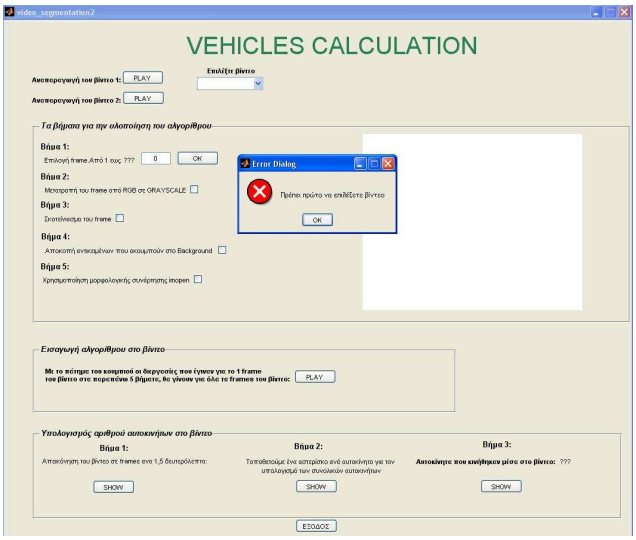
Σχήμα 5

Πατώντας τα κουμπιά αναπαραγωγή βίντεο της εφαρμογής τότε μας παρουσιάζονται τα βίντεο όπως φαίνεται στα σχήματα 4 και 5. Η συνάρτηση play_pushbutton αναφέρεται για το πρώτο κουμπί της εφαρμογής ενώ η συνάρτηση play2_pushbutton για το δεύτερο κουμπί της εφαρμογής. Η εντολή ***mmreader*** διαβάζει το βίντεο και η εντολή ***implay*** κάνει αναπαραγωγή του βίντεο που διαβάστηκε.

2.Pop-up Menu: Επιλέξτε βίντεο

Στο σημείο που επιλέγουμε το βίντεο έχουμε τοποθετήσει ένα Pop-up Menu το οποίο μας δίνει την δυνατότητα να επιλέξουμε ποιο βίντεο θέλουμε. Έχουμε 3 περιπτώσεις που μπορούμε να επιλέξουμε από το Pop-up Menu. $1^{η}$ περίπτωση να επιλέξουμε το βίντεο 1, $2^{η}$ περίπτωση να επιλέξουμε το βίντεο 2 και $3^{η}$ περίπτωση να μην επιλέξουμε κανένα από τα δύο.
Για την περίπτωση που δεν επιλέγουμε κανένα από τα δύο βίντεο έχουμε:

```
case 1
    errordlg('Πρέπει να επιλέξετε ένα από τα 2 βίντεο');
    minima = '???';
    set(handles.frame_input,'String','0');
    %Ξετσεκάρει το gray_checkbox
    minVal = get(handles.gray_checkbox,'Min');
    set(handles.gray_checkbox,'Value',minVal);
    %Ξετσεκάρει το dark_checkbox
    minVal = get(handles.dark_checkbox,'Min');
    set(handles.dark_checkbox,'Value',minVal);
    %Ξετσεκάρει το background_checkbox
    minVal = get(handles.background_checkbox,'Min');
    set(handles.background_checkbox,'Value',minVal);
    %Ξετσεκάρει το imopen_checkbox
    minVal = get(handles.imopen_checkbox,'Min');
    set(handles.imopen_checkbox,'Value',minVal);
    %Refresh axes
    imshow('white.jpg','Parent',handles.axes1);
    set(handles.choice_text,'String',minima);
    set(handles.traffic_text,'String',minima);
```

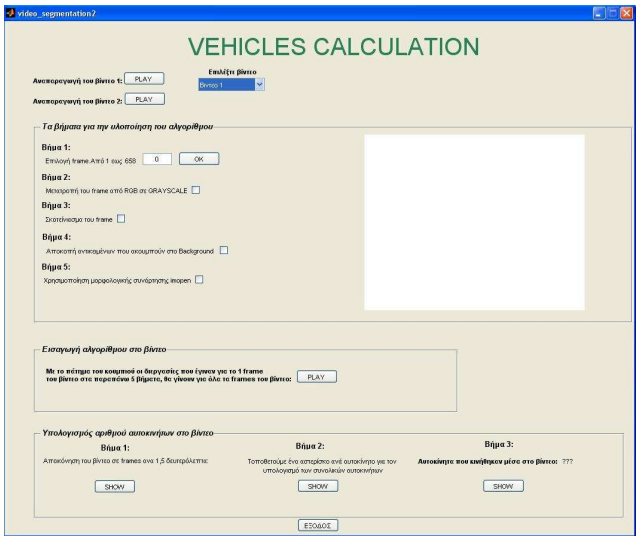


Σχήμα 6

Στην περίπτωση που δεν επιλέξουμε κανένα από τα 2 βίντεο τότε μας εμφανίζεται ένα error dialog που μας παροτρύνει να επιλέξουμε κάποιο βίντεο. Ακόμη ξεμαρκάρει όλα τα checkboxes που έχουμε παρακάτω και μηδενίζει το σύστημα αξόνων axis βάζοντας ένα άσπρο φόντο. Τέλος στο static text που έχουμε για να μας δείχνει των αριθμό των frames του βίντεο τοποθετεί το string ??? γιατί δεν είναι επιλεγμένο κανένα βίντεο Οι εντολές παραπάνω δείχνουν τον τρόπο με τον οποίο γίνονται όλα αυτά και το σχήμα 6 απεικονίζει το αποτέλεσμα.

Για την περίπτωση που διαλέγουμε το βίντεο 1 έχουμε:

```
case 2
    video = mmreader('road3.avi');
    nframes = get(video, 'NumberOfFrames');
    set(handles.frame_input,'String','0');
    minima = '???';
    %Ξετσεκάρει το gray_checkbox
    minVal = get(handles.gray_checkbox,'Min');
    set(handles.gray_checkbox,'Value',minVal);
    %Ξετσεκάρει το dark_checkbox
    minVal = get(handles.dark_checkbox,'Min');
    set(handles.dark_checkbox,'Value',minVal);
    %Ξετσεκάρει το background_checkbox
    minVal = get(handles.background_checkbox,'Min');
    set(handles.background_checkbox,'Value',minVal);
    %Ξετσεκάρει το imopen_checkbox
    minVal = get(handles.imopen_checkbox,'Min');
    set(handles.imopen_checkbox,'Value',minVal);
    %Refresh axes
    imshow('white.jpg','Parent',handles.axes1);
    set(handles.choice_text,'String',nframes);
    set(handles.traffic_text,'String',minima);
```

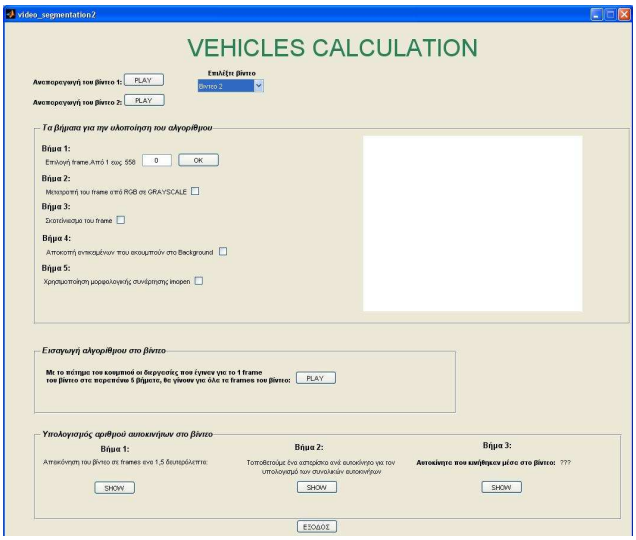


Σχήμα 7

Στην περίπτωση αυτή διαβάζεται το road3.avi, ξεμαρκάρονται όλα τα checkboxes και μηδενίζεται το σύστημα αξόνων axis βάζοντας ένα άσπρο φόντο. Επίσης στο static text που μας δείχνει τον αριθμό των frames τοποθετείται ο αριθμός των frames του βίντεο 1. Οι εντολές δείχνουν τον τρόπο με τον οποίο γίνονται όλα αυτά και στο σχήμα 7 φαίνεται το αποτέλεσμα.

Για την περίπτωση που διαλέγουμε το βίντεο 2 έχουμε:

```
case 3
    video = mmreader('road4.avi');
    nframes = get(video, 'NumberOfFrames');
    minima = '???';
    set(handles.frame_input,'String','0');
    %Ξετσεκάρει το gray_checkbox
    minVal = get(handles.gray_checkbox,'Min');
    set(handles.gray_checkbox,'Value',minVal);
    %Ξετσεκάρει το dark_checkbox
    minVal = get(handles.dark_checkbox,'Min');
    set(handles.dark_checkbox,'Value',minVal);
    %Ξετσεκάρει το background_checkbox
    minVal = get(handles.background_checkbox,'Min');
    set(handles.background_checkbox,'Value',minVal);
    %Ξετσεκάρει το imopen_checkbox
    minVal = get(handles.imopen_checkbox,'Min');
    set(handles.imopen_checkbox,'Value',minVal);
    %Refresh axes
    imshow('white.jpg','Parent',handles.axes1);
    set(handles.choice_text,'String',nframes);
    set(handles.traffic_text,'String',minima);
```



Σχήμα 7

Στην περίπτωση αυτή τότε διαβάζεται το road4.avi ,ξεμαρκάρονται όλα τα checkboxes και μηδενίζεται το σύστημα αξόνων axis βάζοντας ένα άσπρο φόντο. Επίσης στο static text τοποθετείται ο αριθμός των frames του βίντεο 3.

3.Edit Text: Επιλογή frame

Στο σημείο αυτό πρέπει να τοποθετήσουμε έναν αριθμό στο edit text όπου είναι το frame που επιλέγουμε για να γίνει η επεξεργασία. Εδώ έχουμε τρεις περιπτώσεις, η περίπτωση ο αριθμός που εισάγαμε να αφορά το βίντεο 1, η περίπτωση ο αριθμός που εισάγαμε να αντιστοιχεί στο βίντεο 2 και η περίπτωση να μην έχει επιλεγεί κανένα βίντεο.

Για την πρώτη περίπτωση έχουμε:

```
if choice_num == nframes1
   %Παίρνει την τιμή από το πεδίο και την αποθηκεύει στο input σαν string
   input = str2num(get(hObject,'String'));
   %Ξετσεκάρει το gray_checkbox
   minVal = get(handles.gray_checkbox,'Min');
   set(handles.gray_checkbox,'Value',minVal);
   %Ξετσεκάρει το dark_checkbox
   minVal = get(handles.dark_checkbox,'Min');
   set(handles.dark_checkbox,'Value',minVal);
   %Ξετσεκάρει το background_checkbox
   minVal = get(handles.background_checkbox,'Min');
   set(handles.background_checkbox,'Value',minVal);
   %Ξετσεκάρει το imopen_checkbox
   minVal = get(handles.imopen_checkbox,'Min');
   set(handles.imopen_checkbox,'Value',minVal);
   video = mmreader('road3.avi');
   nframes = get(video, 'NumberOfFrames');
   %Τσεκάρει αν το frame_input παίρνει σωστές τιμές
   if (isempty(input))
    set(hObject,'String','0')
    errordlg('Πρέπει να βάλετε μόνο νούμερα στο πεδίο');
   end
   if  input>nframes
      set(hObject,'String','0')
      errordlg('Οι τιμές στο πεδίο πρέπει να είναι μεταξύ του 1 και 658');
```

```
   end
   if  input<1
      set(hObject,'String','0')
      errordlg('Οι τιμές στο πεδίο πρέπει να είναι μεταξύ του 1 και 658');
   end
end
```

Στην περίπτωση αυτή ξεμαρκάρονται όλα τα checkboxes δημιουργείται ένας έλεγχος όπου σιγουρευόμαστε ότι η τιμή που μπήκε στο πεδίο είναι μέσα στα όρια των frames του βίντεο και ότι δεν είναι γράμματα. Εάν εντοπιστεί μία τέτοια περίπτωση τότε εμφανίζεται ένα error dialog που υποδεικνύει το σφάλμα που εντοπίστηκε. Οι εντολές παραπάνω δείχνουν τον τρόπο με τον οποίο γίνονται όλα αυτά.
Για την δεύτερη περίπτωση έχουμε:

```
if choice_num == nframes2
   %Παίρνει την τιμή από το πεδίο και την αποθηκεύει στο input σαν string
   input = str2num(get(hObject,'String'));
   %Ξετσεκάρει το gray_checkbox
   minVal = get(handles.gray_checkbox,'Min');
   set(handles.gray_checkbox,'Value',minVal);
   %Ξετσεκάρει το dark_checkbox
   minVal = get(handles.dark_checkbox,'Min');
   set(handles.dark_checkbox,'Value',minVal);
   %Ξετσεκάρει το background_checkbox
   minVal = get(handles.background_checkbox,'Min');
   set(handles.background_checkbox,'Value',minVal);
   %Ξετσεκάρει το imopen_checkbox
   minVal = get(handles.imopen_checkbox,'Min');
   set(handles.imopen_checkbox,'Value',minVal);
   video = mmreader('road4.avi');
   nframes = get(video, 'NumberOfFrames');
   %Τσεκάρει αν το frame_input παίρνει σωστές τιμές
   if (isempty(input))
    set(hObject,'String','0')
    errordlg('Πρέπει να βάλετε μόνο νούμερα στο πεδίο');
   end
   if  input>nframes
      set(hObject,'String','0')
      errordlg('Οι τιμές στο πεδίο πρέπει να είναι μεταξύ του 1 και 558');
   end
   if  input<1
      set(hObject,'String','0')
      errordlg('Οι τιμές στο πεδίο πρέπει να είναι μεταξύ του 1 και 558');
   end
end
```

Στην περίπτωση αυτή ξεμαρκάρονται όλα τα checkboxes δημιουργείται ένας έλεγχος όπου σιγουρευόμαστε ότι η τιμή που μπήκε στο πεδίο είναι μέσα στα όρια των frames του βίντεο και ότι δεν είναι γράμματα. Εάν εντοπιστεί μία τέτοια περίπτωση τότε εμφανίζεται ένα error dialog που υποδεικνύει το σφάλμα που εντοπίστηκε. Οι εντολές παραπάνω δείχνουν τον τρόπο με τον οποίο γίνονται όλα αυτά.
Για την Τρίτη περίπτωση έχουμε:

```
if choice == '???'
   set(hObject,'String','0')
```

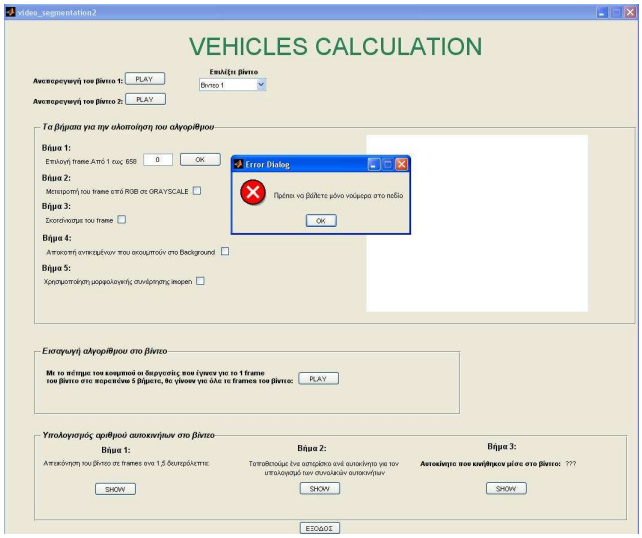
```
   errordlg('Πρέπει πρώτα να επιλέξετε βίντεο');
end
```

Στην περίπτωση αυτή απλώς εμφανίζεται ένα error dialog το οποίο μας παροτρύνει να επιλέξουμε ένα από τα δύο βίντεο όπως φαίνεται και στις εντολές παραπάνω.

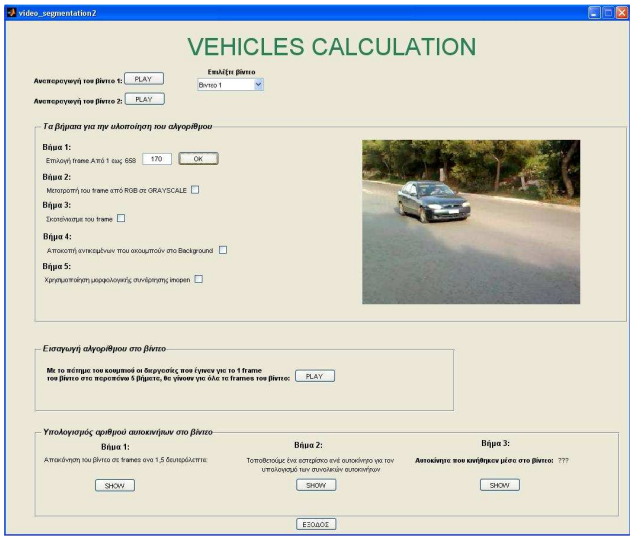
4. Pushbutton: Επιλογή frame

Στο σημείο αυτό αφού τοποθετήσουμε τον αριθμό στο edit text πρέπει να πατήσουμε το κουμπί για να ενεργοποιηθούν όλες οι ενέργειες που περιγράψαμε παραπάνω.

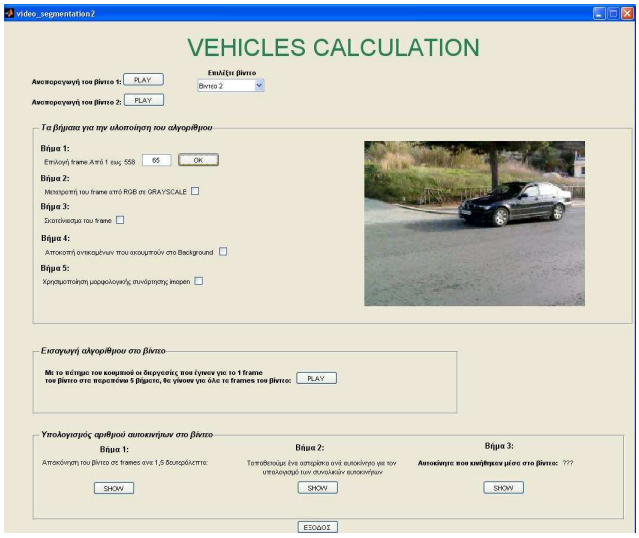
```
%Παίρνουμε την τιμή από το frame_input και την εισάγουμε στην μεταβλητή a
a = get(handles.frame_input,'String');
%Μετατρέπουμε την τιμή a από string σε num και την τοποθετούμε στην a_num
a_num = str2num(a);
%Διάβασμα του βίντεο
video1 = mmreader('road3.avi');
nframes1 = get(video1, 'NumberOfFrames');
choice = get(handles.choice_text,'String');
choice_num = str2num(choice);
if choice_num == nframes1
   video = mmreader('road3.avi');
else
   video = mmreader('road4.avi');
end
%Με την read παίρνουμε το frame που τοποθετείτε στην selected_frame
selected_frame=read(video,a_num);
%Παρουσίαση του αποτελέσματος στους άξονες
imshow(selected_frame);
```



Σχήμα 8



Σχήμα 9



Σχήμα 10

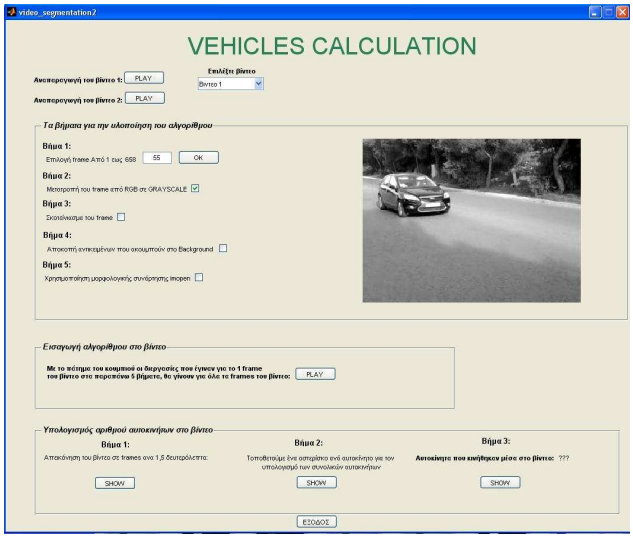
Με το πάτημα του κουμπιού εκτελούνται οι διεργασίες που περιγράψαμε παραπάνω στο edit text. Εάν ο αριθμός που τοποθετήθηκε στο edit text ικανοποιεί όλες τις συνθήκες ελέγχου τότε το πρόγραμμα εκτελεί τις παραπάνω εντολές και τοποθετεί στο σύστημα αξόνων axis το frame που πληκτρολογήσαμε. Στο σχήμα 8 απεικονίζεται η περίπτωση στην οποία αντί για αριθμό βάλαμε γράμματα με αποτέλεσμα να εμφανιστεί ένα error dialog. Στα σχήματα 9 και 10 απεικονίζονται οι περιπτώσεις όπου το πρόγραμμα εμφάνισε το frame που ζητήσαμε στο σύστημα αξόνων axis για το βίντεο 1 και βίντεο 2 αντίστοιχα.

5. Checkbox: Μετατροπή εικόνας από RGB σε GRAYSCALE

Σε αυτό το σημείο αφού επιλέξαμε το frame αρχίζει η δημιουργία του αλγορίθμου μας για τον εντοπισμό των αυτοκινήτων μέσα στο βίντεο. Επειδή είναι πιο εύκολη η επεξεργασία μιας εικόνας με όσο το δυνατόν λιγότερα επίπεδα φωτεινότητας, γι'αυτό μετατρέπουμε την εικόνα σε grayscale(255 επίπεδα γκρι). Αυτό είναι το πρώτο βήμα του αλγορίθμου μας και η μετατροπή της εικόνας θα γίνει απλά επιλέγοντας το ανάλογο checkbox.

```
%Παίρνουμε την τιμή από το frame_input και την εισάγουμε στην μεταβλητή a
   a = get(handles.frame_input,'String');
   %Μετατρέπουμε την τιμή a από string σε num και την τοποθετούμε στην a_num
```

```
a_num = str2num(a);
%Έλεγχος αν έχει επιλεγεί frame
if a_num==0
   errordlg('Πρέπει πρώτα να επιλέξετε ένα frame');
   %Ξετσεκάρει το gray_checkbox
   minVal = get(handles.gray_checkbox,'Min');
   set(handles.gray_checkbox,'Value',minVal);
else
%Διάβασμα του βίντεο
video1 = mmreader('road3.avi');
nframes1 = get(video1, 'NumberOfFrames');
choice = get(handles.choice_text,'String');
choice_num = str2num(choice);
if choice_num == nframes1
   video = mmreader('road3.avi');
else
   video = mmreader('road4.avi');
end
%Με την read παίρνουμε το frame που τοποθετείτε στην selected_frame
selected_frame=read(video,a_num);
%Μετατροπή σε Grayscale εικόνα
graySelected_frame = rgb2gray(selected_frame);
%Παρουσίαση του αποτελέσματος στους άξονες
imshow(graySelected_frame);
```



Σχήμα 11

Αφού επιλέξουμε το checkbox για την μετατροπή της εικόνας οι διαδικασίες είναι οι ακόλουθες:

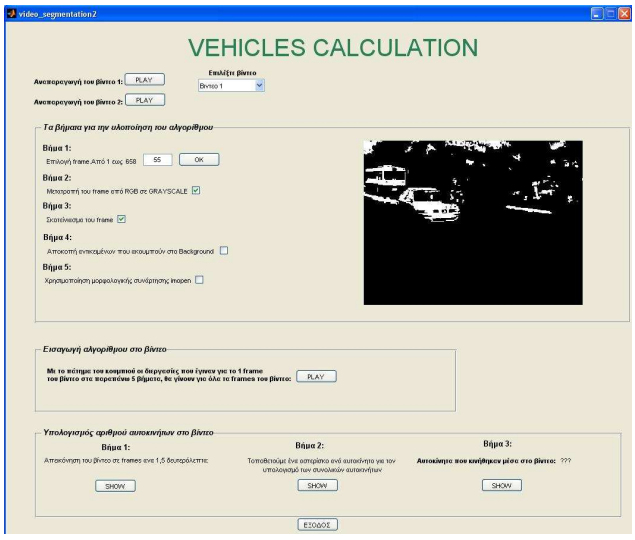
- Γίνεται έλεγχος εάν έχει επιλεγεί κάποιο frame και αν όχι τότε εμφανίζεται ένα error dialog.
- Εάν έχει επιλεγεί κάποιο frame γίνεται έλεγχος από ποιο βίντεο προέρχεται.
- Ξεμαρκάρει τα υπόλοιπα checkboxes εκτός του εαυτού του.
- Μετατρέπει την εικόνα από RGB σε GRAYSCALE.

Οι εντολές παραπάνω δείχνουν τον τρόπο υλοποίησης των ενεργειών αυτών και το σχήμα 11 απεικονίζει το αποτέλεσμα.

6. Checkbox: Σκοτείνιασμα του frame

Αφού υλοποιήθηκε το πρώτο βήμα του αλγορίθμου μας είμαστε έτοιμοι να πάμε στο δεύτερο που είναι το <<σκοτείνιασμα>> του frame. Αυτό γίνεται απλά επιλέγοντας το αντίστοιχο checkbox της εφαρμογής.

```
%Έλεγχος αν το gray_checkbox είναι τσεκαρισμένο
    if(get(handles.gray_checkbox,'Value') == get(handles.gray_checkbox,'Min'))
       errordlg('Πρέπει πρώτα να μετατραπεί το frame σε grayscale');
       %Ξετσεκάρει το checkbox
        minVal = get(handles.dark_checkbox,'Min');
        set(handles.dark_checkbox,'Value',minVal);
    else
       %Παίρνουμε την τιμή από το frame_input και την εισάγουμε στην μεταβλητή a
       a = get(handles.frame_input,'String');
       %Μετατρέπουμε την τιμή a από string σε num και την τοποθετούμε στην a_num
       a_num = str2num(a);
       %Διάβασμα του βίντεο
       video1 = mmreader('road3.avi');
       nframes1 = get(video1, 'NumberOfFrames');
       choice = get(handles.choice_text,'String');
       choice_num = str2num(choice);
       if choice_num == nframes1
          video = mmreader('road3.avi');
       else
          video = mmreader('road4.avi');
       end
       %Με την read παίρνουμε το frame που τοποθετείτε στην selected_frame
       selected_frame=read(video,a_num);
       %Μετατροπή σε Grayscale εικόνα
       graySelected_frame = rgb2gray(selected_frame);
       %Threshold
       darkValue = 50;
       %Σκοτείνιασμα frame με την χρήση της imextendedmax
       darkFrame = imextendedmax(graySelected_frame,darkValue);
       %Παρουσίαση του αποτελέσματος στους άξονες
       imshow(darkFrame);
```



Σχήμα 12

Οι ενέργειες που γίνονται επιλέγοντας το checkbox είναι οι εξής:

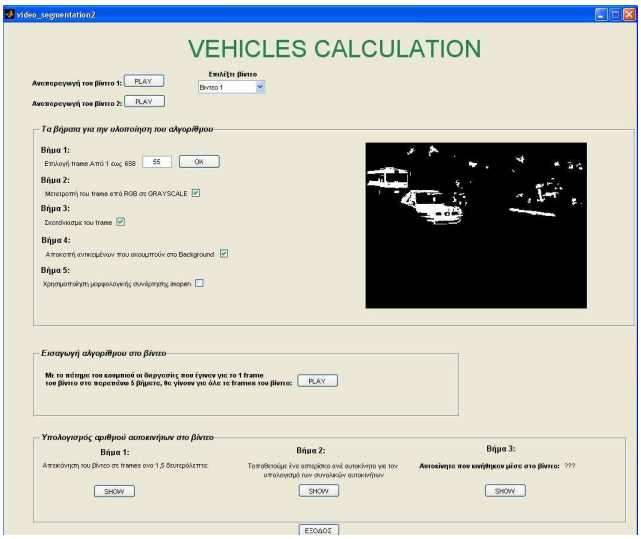
- Γίνεται έλεγχος αν το παραπάνω checkbox είναι τσεκαρισμένο για να μπορέσει να συνεχιστεί η διαδικασία
- Ξεμαρκάρει τα checkboxes που βρίσκονται κάτω από αυτό.
- Ελέγχει το frame που έχει επιλεγεί από ποιο frame προέρχεται.
- <<Σκοτεινιάζει>> την εικόνα.

Το <<σκοτείνιασμα>> της εικόνας το επιτυγχάνουμε με την χρήση της συνάρτησης ***imextendedmax***. Ορίζουμε ένα κατώφλι(threshold) στην περίπτωση μας είναι 50. Η συνάρτηση ***imextendedmax*** μας γυρνάει μια δυαδική εικόνα όπου pixels με τιμή μεγαλύτερη του 50 παίρνουν την τμή 1(άσπρο) και pixels με τιμή μικρότερη του 50 παίρνουν την τιμή 0(μαύρο). Στο σχήμα 12 απεικονίζεται το αποτέλεσμα αυτών των διαδικασιών.

7. Checkbox: Αποκοπή αντικειμένων που ακουμπούν στο background

Το τρίτο βήμα της υλοποίησης του αλγορίθμου είναι η αποκοπή των άσπρων pixels που ακουμπούν στα όρια της εικόνας με αποτέλεσμα τα μόνα pixels που θα μείνουν στην εικόνα να είναι το αυτοκίνητο που θέλουμε να αναγνωρίσουμε. Αυτήν την διαδικασία κάνει και το συγκεκριμένο checkbox.

```
%Έλεγχος αν το dark_checkbox είναι τσεκαρισμένο
    if(get(handles.dark_checkbox,'Value') == get(handles.dark_checkbox,'Min'))
       errordlg('Πρέπει πρώτα να σκοτεινιάσουμε το frame');
       %Ξετσεκάρει το checkbox
        minVal = get(handles.background_checkbox,'Min');
        set(handles.background_checkbox,'Value',minVal);
    else
       %Παίρνουμε την τιμή από το frame_input και την εισάγουμε στην μεταβλητή a
       a = get(handles.frame_input,'String');
       %Μετατρέπουμε την τιμή a από string σε num και την τοποθετούμε στην a_num
       a_num = str2num(a);
       %Διάβασμα του βίντεο
       video1 = mmreader('road3.avi');
       nframes1 = get(video1, 'NumberOfFrames');
       choice = get(handles.choice_text,'String');
       choice_num = str2num(choice);
       if choice_num == nframes1
          video = mmreader('road3.avi');
       else
          video = mmreader('road4.avi');
       end
       %Με την read παίρνουμε το frame που τοποθετείτε στην selected_frame
       selected_frame=read(video,a_num);
       %Μετατροπή σε Grayscale εικόνα
       graySelected_frame = rgb2gray(selected_frame);
       %Threshold
       darkValue = 50;
       %Σκοτείνιασμα frame με την χρήση της imextendedmax
       darkFrame = imextendedmax(graySelected_frame,darkValue);
       %Αποκοπή αντικειμένων από το background
       cutBoard=imclearborder(darkFrame,26);
       %Παρουσίαση του αποτελέσματος στους άξονες
       imshow(cutBoard);
```



Σχήμα 13

Οι ενέργειες που γίνονται κλικάροντας το checkbox είναι οι εξής:

- Γίνεται έλεγχος αν το παραπάνω checkbox είναι κλικαρισμένο για να μπορέσει να συνεχίσει η διαδικασία
- Ξετσεκάρει το checkbox που βρίσκεται κάτω από αυτό
- Ελέγχει το frame που έχει επιλεγεί από ποιο βίντεο προέρχεται
- Εξαλείφει τα pixels που ακουμπούν πάνω στα όρια της εικόνας

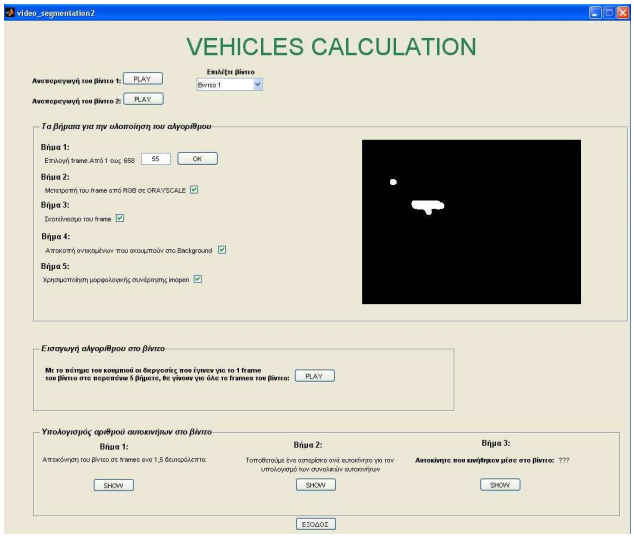
Η αποκοπή των pixels γίνεται με την βοήθεια της συνάρτησης ***imclearborder***. Στο σχήμα 13 απεικονίζεται το αποτέλεσμα αυτών των ενεργειών.

8. Checkbox: Χρησιμοποίηση μορφολογικής συνάρτησης imopen

Το τελευταίο βήμα για την ολοκλήρωση του αλγορίθμου για την αναγνώριση των αυτοκινήτων μέσα από το βίντεο είναι η χρήση της μορφολογικής συνάρτησης ***imopen***. Όπως παρατηρήσαμε στο σχήμα 12 ναι μεν αποκόπηκαν τα pixels από τα όρια της εικόνας αλλά παρέμειναν και κάποια άλλα pixels τα οποία δεν ανήκουν στην περιοχή του ενδιαφέροντος μας. Παρέμειναν γιατί είχαν τιμή μεγαλύτερη του 50(threshold) αλλά εμείς πρέπει να τα αφαιρέσουμε για να μείνουν στην εικόνα μόνο τα pixels των αντικειμένων που θέλουμε να αναγνωρίσουμε(αυτοκίνητα) Για να το επιτύχουμε αυτό χρησιμοποιούμε την μορφολογική συνάρτηση ***imopen*** όπου μηδενίζει την τιμή των pixels που δεν πληρούν την συνθήκη strel(‘disk’,4). Η συνθήκη αυτή υποδεικνύει τα pixels που έχουν μορφή δίσκου. Ο αριθμός μας δείχνει το μέγεθος του δίσκου που θέλουμε να κρατήσουμε από την εικόνα μας.

```
%Έλεγχος αν το background_checkbox είναι τσεκαρισμένο
    if(get(handles.background_checkbox,'Value') == get(handles.background_checkbox,'Min'))
        errordlg('Πρέπει πρώτα να αποκοπούν τα αντικείμενα που ακουμπούν στο background');
        %Ξεμαρκάρει το checkbox
         minVal = get(handles.imopen_checkbox,'Min');
         set(handles.imopen_checkbox,'Value',minVal);
    else
        %Παίρνουμε την τιμή από το frame_input και την εισάγουμε στην μεταβλητή a
        a = get(handles.frame_input,'String');
        %Μετατρέπουμε την τιμή a από string σε num και την τοποθετούμε στην a_num
        a_num = str2num(a);
        %Διάβασμα του βίντεο
        video1 = mmreader('road3.avi');
```

```
    nframes1 = get(video1, 'NumberOfFrames');
    choice = get(handles.choice_text,'String');
    choice_num = str2num(choice);
    if choice_num == nframes1
       video = mmreader('road3.avi');
    else
       video = mmreader('road4.avi');
    end
    %Με την read παίρνουμε το frame που τοποθετείτε στην selected_frame
    selected_frame=read(video,a_num);
    %Μετατροπή σε Grayscale εικόνα
    graySelected_frame = rgb2gray(selected_frame);
    %Threshold
    darkValue = 50;
    %Σκοτείνιασμα frame με την χρήση της imextendedmax
    darkFrame = imextendedmax(graySelected_frame,darkValue);
    %Αποκοπή αντικειμένων από το background
    cutBoard=imclearborder(darkFrame,26);
    %Μορφή δίσκου
    if choice_num == nframes1
       sedisk = strel('disk',5);
    else
       sedisk = strel('disk',8);
    end
    %Μορφολογική συνάρτηση imopen
    noSmallStructures = imopen(cutBoard, sedisk);
    %Παρουσίαση του αποτελέσματος στους άξονες
    imshow(noSmallStructures);
end
```



Σχήμα 14

Οι ενέργειες που γίνονται επιλέγοντας το checkbox είναι οι εξής:

- Γίνεται έλεγχος εάν είναι επιλεγμένο το προηγούμενο checkbox για να μπορέσει να συνεχίσει η διαδικασία.
- Ελέγχει το frame που επιλέχθηκε από ποιο βίντεο προήλθε.
- Γίνεται χρήση της μορφολογικής συνάρτησης ***imopen.***

Πρέπει να σημειώσουμε ότι επειδή τα 2 βίντεο έχουν διαφορές μεταξύ τους όπως η απόσταση που είναι τραβηγμένα τα βίντεο, ο αριθμός που προαναφέραμε στην προηγούμενη

παράγραφο ο οποίος δείχνει το μέγεθος του δίσκου διαφέρει στα 2 βίντεο. Για τον λόγο αυτό πρέπει να γίνει έλεγχος του frame που επιλέχθηκε από ποιο βίντεο προήλθε για να δοθεί και ο κατάλληλος αριθμός για να συνεχιστεί η επεξεργασία. Στην εφαρμογή αυτή εάν το frame προέρχεται από το βίντεο 1 ο αριθμός παίρνει την τιμή 5 αλλιώς παίρνει την τιμή 8, όπως φαίνεται και παρακάτω

```
if choice_num == nframes1
        sedisk = strel('disk',5);
    else
        sedisk = strel('disk',8);
    end
```

Τέλος στο σχήμα 14 απεικονίζεται το αποτέλεσμα της χρήσης της ***imopen*** για το βίντεο 1

9. Pushbutton: Εισαγωγή αλγορίθμου στο βίντεο

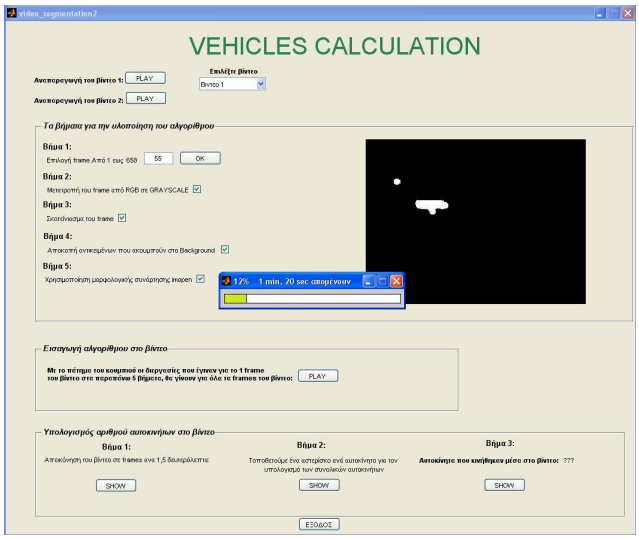
Αφού πλέον ολοκληρώσαμε τον αλγόριθμο για το frame που επιλέξαμε επόμενο βήμα είναι να εισάγουμε τον αλγόριθμο στο βίντεο. Δηλαδή να κάνουμε την διαδικασία που κάναμε όχι μόνο σε ένα frame αλλά σε όλα τα frames του βίντεο. Ο κώδικας είναι ο ίδιος με τον κώδικα όλων των checkboxes μαζί αλλά με την διαφορά ότι τον τοποθετούμε σε ένα for για να εκτελεστεί ο κώδικας τόσες φορές όσο είναι και τα frame του βίντεο που έχουμε επιλέξει εκείνη την στιγμή. Για να φανεί στο μάτι ότι έχουμε πετύχει την διαδικασία αυτό που κάνουμε είναι να αλλάξουμε το κεντρικό pixel των αντικειμένων που αναγνωρίσαμε σε κόκκινο.

```
% Παίρνουμε την περιοχή και το κέντρο των αντικειμένων που έμειναν
   % στο frame. Τα αντικείμενα με την μεγαλύτερη περιοχή είναι
   % τα αυτοκίνητα. Δημιουργούμε ένα αντίγραφο του αρχικού μας frame και
   % στην ουσία μαρκάρουμε το αυτοκίνητο αλλάζοντας το χρώμα του
   % κεντρικού pixel σε κόκκινο.
   taggedCars(:,:,:,k) = singleFrame;

   stats = regionprops(noSmallStructures, {'Centroid','Area'});
   if ~isempty([stats.Area])
      areaArray = [stats.Area];
      [junk,idx] = max(areaArray);
      c = stats(idx).Centroid;
      c = floor(fliplr(c));
      width = 2;
      row = c(1)-width:c(1)+width;
      col = c(2)-width:c(2)+width;
      taggedCars(row,col,1,k) = 255;
      taggedCars(row,col,2,k) = 0;
      taggedCars(row,col,3,k) = 0;
   end
```

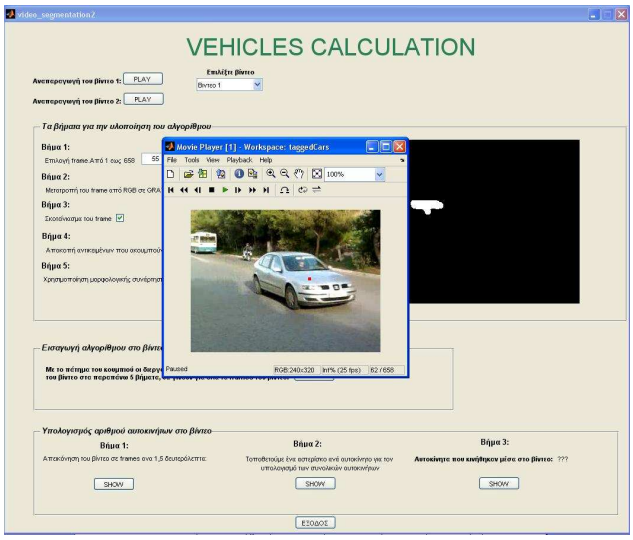
Επίσης επειδή τα βίντεο έχουν πολλά frames μέχρι να εκτελεστεί το for για όλα τα frames απαιτεί κάποια ώρα. Για να μην περιμένει λοιπόν ο χρήστης να τελειώσει μην βλέποντας τίποτα στην οθόνη έχει προσαρμοστεί μία μπάρα όπου γεμίζει με την πάροδο του χρόνου δίνοντας στον χρήστη να καταλάβει ότι η διαδικασία θα τελειώσει μόλις η μπάρα έχει φτάσει στο 100%.

```
for k = 1 : nframes
    pause(0.01)
    progressbar(k/nframes);
```

Η συνάρτηση ***progressbar*** είναι μια συνάρτηση έτοιμη η οποία εισάγεται σε αυτό το σημείο του κώδικα για να εκτεζλεστεί και αυτή.



Σχήμα 15



Σχήμα 16

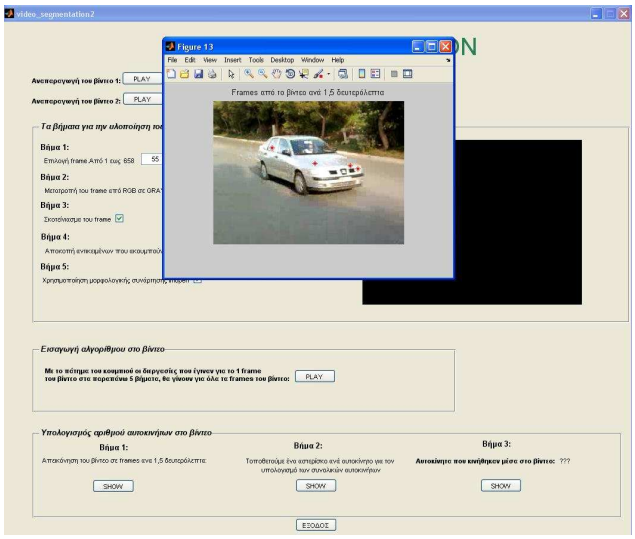
Στο σχήμα 15 απεικονίζεται η στιγμή στην οποία η διεργασία εκτελείται και έχει εμφανιστεί η μπάρα progressbar ενώ στο σχήμα 16 η διεργασία έχει τελειώσει και απεικονίζεται ένα καρέ από το βίντεο όπου έχει γίνει η αναγνώριση των αυτοκινήτων

10. Pushbutton: Απεικόνιση του βίντεο σε frames ανά 1, 5 δευτερόλεπτα

Μέχρι στιγμής έχουμε δημιουργήσει δύο βίντεο στα οποία έχουν εντοπιστεί τα αυτοκίνητα που υπάρχουν μέσα σε αυτά. Επόμενο βήμα μας είναι να υπολογίζουμε την κίνηση των αυτοκινήτων που υπάρχουν μέσα στα δύο βίντεο. Για να γίνει αυτό πρέπει να πάρουμε μερικά frames ανά δευτερόλεπτο έτσι ώστε να μπορέσουμε να βγάλουμε κάποιο πόρισμα για την κίνηση των αυτοκινήτων. Εφόσον γνωρίζουμε τον αριθμό των frames των βίντεο μας και το framerate που έχει το κάθε βίντεο μπορούμε με μια απλή διαίρεση μέσα σε μία for να αναπαραστήσουμε τα βίντεο. Παρακάτω φαίνεται το κομμάτι του κώδικα που βρίσκεται μέσα στο for

```
% Μετατροπή σε grayscale για την μορφολογική επεξεργασία.
    I = rgb2gray(singleFrame);

    % Σκοτείνιασμα των frames.
    %Threshold
    darkValue = 50;
    noDarkFrames = imextendedmax(I, darkValue);

    % Αποκοπή περιοχών που ακουμπούν στο backround
    cutBoard = imclearborder(noDarkFrames,26);

    % Αποκοπή γραμμών και άλλων αντικειμένων που δεν έχουν μορφή δίσκου.
    %Μορφή δίσκου
    sedisk = strel('disk',5);
    noSmallStructures = imopen(cutBoard, sedisk);


    s = regionprops(noSmallStructures,I, {'Centroid'});
    % Εμφανίζει τα figures για την αναπαράσταση των frames
    figure,imshow(singleFrame)
    title('Frames από το βίντεο ανά 1,5 δευτερόλεπτα');
    hold on
    numObj = numel(s);
    for c = 1 : numObj
       %Βάζει κόκκινους σταυρούς στα αυτοκίνητα
       plot(s(c).Centroid(1), s(c).Centroid(2), 'r*');
    end
    hold off
```

Όπως παρατηρούμε εφαρμόζεται πάλι ο αλγόριθμος για την αναγνώριση των αυτοκινήτων στο βίντεο με την διαφορά ότι αυτήν την φορά δεν αλλάζουμε το κεντρικό pixel σε κόκκινο αλλά τοποθετούμε κόκκινους αστερίσκους κάθε φορά που ο αλγόριθμος εντοπίζει σημεία που αντιστοιχούν στα αυτοκίνητα.



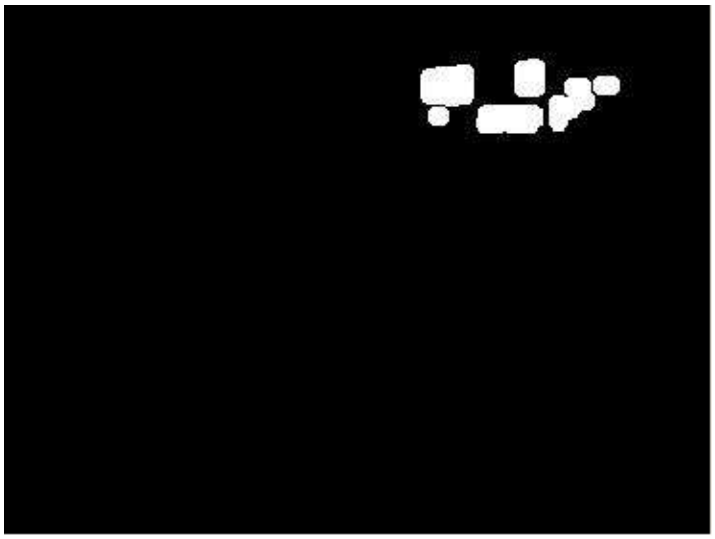
Σχήμα 17

Στο σχήμα 17 απεικονίζεται ένα από τα frames που δημιουργούνται όπου έχουν τοποθετηθεί κόκκινοι αστερίσκοι πάνω του.

11. Pushbutton: Τοποθέτηση ενός αστερίσκου ανά 1, 5 δευτερόλεπτα

Στο προηγούμενο pushbutton που χρησιμοποιήσαμε παρατηρούμε ότι για κάθε αυτοκίνητο που εντοπίζει το πρόγραμμα τοποθετούνται παραπάνω από ένας αστερίσκους. Αυτό το φαινόμενο συμβαίνει διότι η μορφολογική συνάρτηση ***imopen*** εντοπίζει σε κάθε αυτοκίνητο παραπάνω από μία περιοχές μορφής δίσκου. Για να γίνει πιο κατανοητό στα παρακάτω σχήματα απεικονίζεται το ένατο frame του video 1 που έχει υποστεί την μορφολογική συνάρτηση ***imopen***.

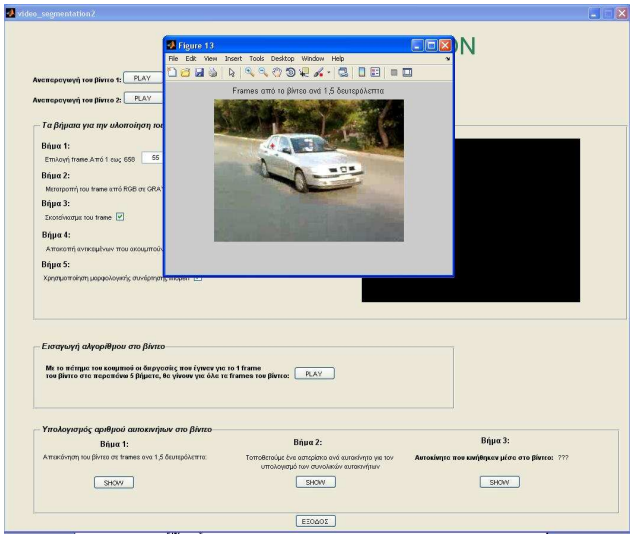
Το ένατο frame από το video 1. Σχήμα 18



Το ένατο frame μετά την χρήση της μορφολογικής συνάρτησης imopen. Σχήμα 19

Όπως παρατηρούμε λοιπόν για αυτό το frame η ***imopen*** δημιουργεί 6 περιοχές με αποτέλεσμα να τοποθετηθούν και 6 αστερίσκοι. Με την χρήση αυτού του pushbutton λοιπόν θα τοποθετούμε ένα αστερίσκο ανά αυτοκίνητο για να έχουμε την δυνατότητα να μετράμε τους αστερίσκους και να βρίσκουμε τα συνολικά αυτοκίνητα που κινήθηκαν μέσα στο βίντεο. Ο κώδικας που θα χρησιμοποιήσουμε είναι ίδιος με αυτόν του προηγούμενου pushbutton με την διαφορά όπως φαίνεται παρακάτω.

```
% Εμφανίζει τα figures για την αναπαράσταση των frames
    figure,imshow(singleFrame)
    title('Frames από το βίντεο ανά 1,5 δευτερόλεπτα');
    hold on
    numObj = numel(s);
    for c = 1 : numObj
       %Βάζει ένα κόκκινο σταυρό σε κάθε αυτοκίνητο
       plot(s(1).Centroid(1), s(1).Centroid(2), 'r*');
    end
    hold off
```



Σχήμα 20

Στις εντολές του σχήματος 20 παρατηρούμε ότι για να μας εμφανιστεί ένας αστερίσκος ανά αυτοκίνητο τοποθετούμε στην plot 1 αντί για την μεταβλητή c. Στο σχήμα 20 απεικονίζεται το αποτέλεσμα αυτής της διαδικασίας.

12. Pushbutton: Αυτοκίνητα που κινήθηκαν μέσα στο βίντεο

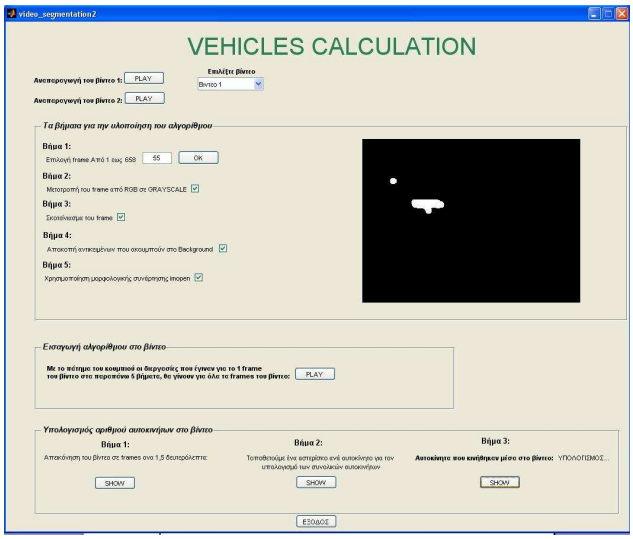
Σε αυτό το pushbutton αυτό που πρέπει να κάνουμε είναι να μετρήσουμε τους αστερίσκους από το βήμα που βάλαμε ένα αστερίσκο ανά αυτοκίνητο για να βρούμε τα συνολικά αυτοκίνητα που κινήθηκαν μέσα στο βίντεο. Αυτό γίνεται εύκολα βάζοντας ένα μετρητή να αυξάνει κατά ένα κάθε φορά που μπαίνει στο loop για να τοποθετήσει τον αστερίσκο, όπως φαίνεται εδώ:

```
    numObj = numel(s);
    for c = 1 : numObj
       %Βάζει ένα κόκκινο σταυρό σε κάθε αυτοκίνητο
       plot(s(1).Centroid(1), s(1).Centroid(2), 'r*');
       %Αυξάνει κατά 1 ο μετρητής
……sinolo = sinolo + 1;
    end
    %Μετατροπή του μετρητή από String σε Αριθμό
    show = num2str(sinolo);
    %Τοποθέτηση του αριθμού στο static text
    set(handles.traffic_text,'String',show);
```

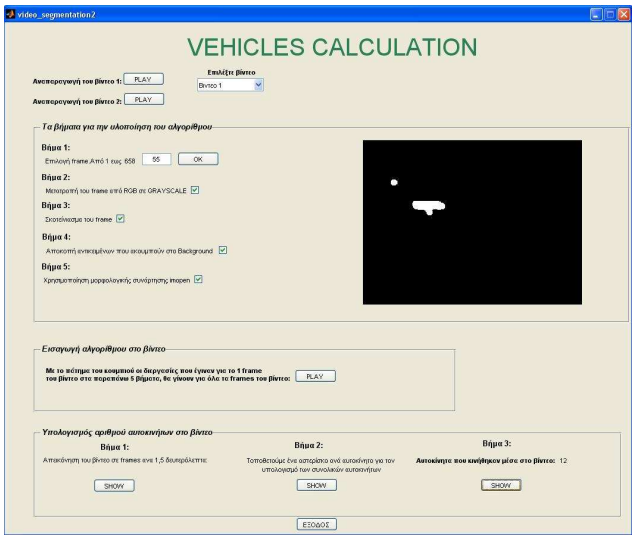
Επειδή ο χρόνος που απαιτείται για να σαρώσει τα frames και να μετρήσει τους αστερίσκους είναι μεγάλος βάζουμε ένα μήνυμα που λέει <<ΥΠΟΛΟΓΙΣΜΟΣ>> μέχρι να

τελειώσει η διαδικασία και να μας δώσει το αποτέλεσμα Ο τρόπος με τον οποίο γίνεται αυτό φαίνεται εδώ:

message = 'ΥΠΟΛΟΓΙΣΜΟΣ...';
set(handles.traffic_text,'String',message);



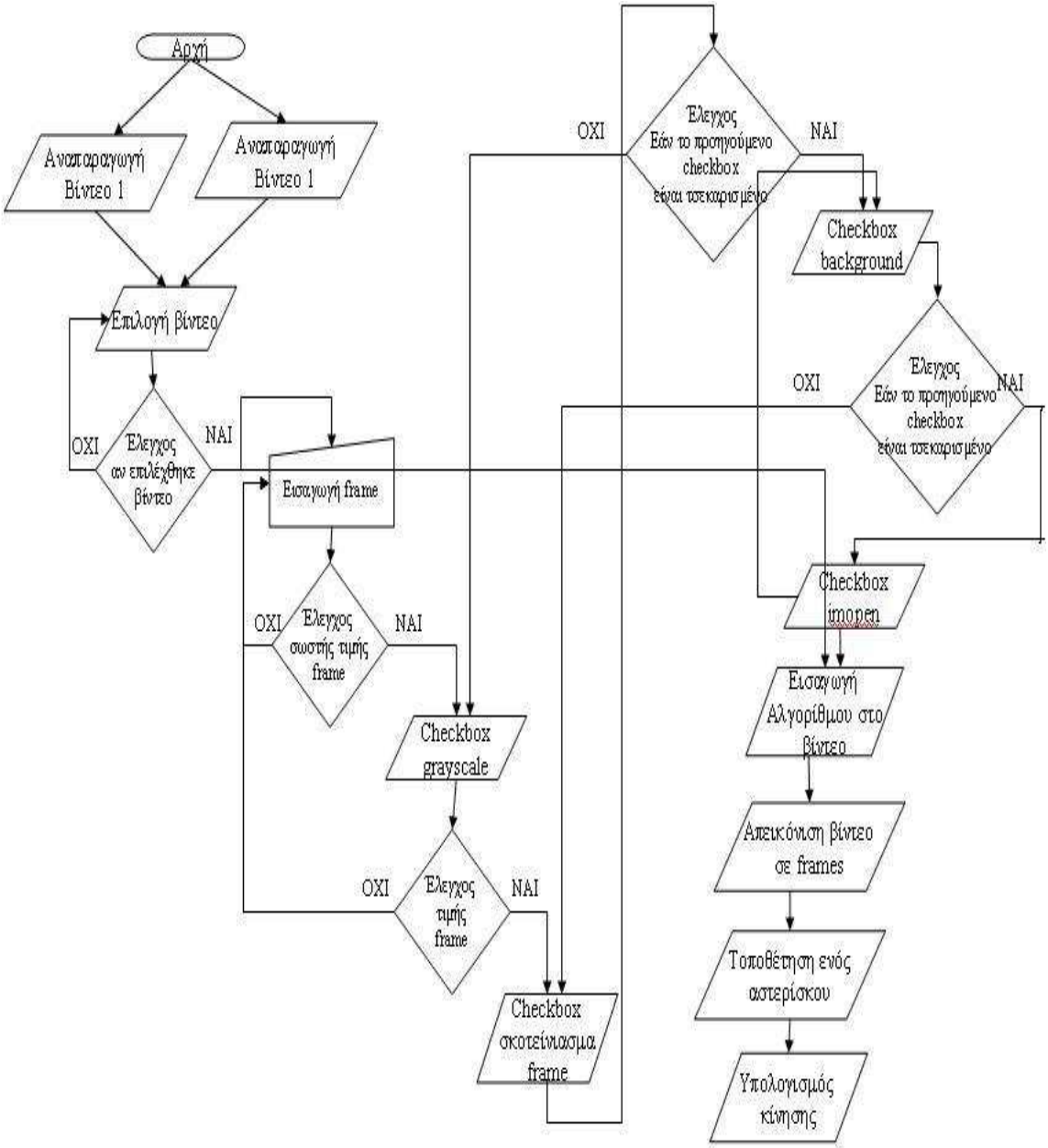
Σχήμα 21



Σχήμα 22

Στο σχήμα 21 απεικονίζεται η εκτέλεση του pushbutton για τον συνολικό αριθμό των αυτοκινήτων την ώρα που γίνεται ο υπολογισμός ενώ στο σχήμα 22 την στιγμή που μας έχει παρουσιάσει το αποτέλεσμα.

Στο παρακάτω διάγραμμα ροής απεικονίζεται ο τρόπος λειτουργίας του GUI.



Κεφάλαιο 9

Παρατηρήσεις-Συμπεράσματα

Πρέπει να σημειωθεί ότι τα βίντεο που εισάγονται για να υποστούν επεξεργασία από την εφαρμογή GUI πρέπει να πληρούν κάποιες προϋποθέσεις για να αποφύγουμε τυχόν αστοχίες που θα εμφανιστούν.

1. Επειδή ακολουθούμε την τεχνική <<σκοτείνιασμα>> του frame θα πρέπει η φωτεινότητα του βίντεο να μην είναι μεγάλη γιατί αλλιώς κατά το σκοτείνιασμα του frame υπάρχει το ενδεχόμενο εκτός από τα αυτοκίνητα να παραμείνουν και άλλα σημεία όπως για παράδειγμα φωτεινές επιγραφές, ο ήλιος και γενικά σημεία ανοιχτού χρώματος κοντά στο άσπρο.

Σχήμα 1. Το άσπρο χρώμα του ουρανού δημιουργεί πρόβλημα στον εντοπισμό του αυτοκινήτου

2. Ένα άλλο πρόβλημα που δημιουργείτε κατά την τεχνική σκοτεινιάσματος του frame είναι ότι μαύρα ή ακόμα και σκουρόχρωμα αυτοκίνητα μπορεί να απαλειφθούν κατά την διαδικασία με αποτέλεσμα να μην εντοπιστούν καθόλου από την εφαρμογή. Για την επίλυση αυτού του προβλήματος θα πρέπει το βίντεο να έχει τραβηχτεί από κοντινή λήψη έτσι ώστε η εφαρμογή να μπορέσει να εντοπίσει τα αυτοκίνητα από μερικά φωτεινά σημεία τους όπως είναι τα φανάρια, τα τζάμια, οι πινακίδες τους κ.α.

Σχήμα 2. Σφάλμα εντοπισμού μαύρου αυτοκινήτου

3. Ένα άλλο σημείο που πρέπει να προσεχθεί κατά την λήψη του βίντεο είναι η λήψη να γίνεται από κάποιο ύψος ή κάποια γωνία με σκοπό να αποφύγουμε το σφάλμα να υπάρχουν δύο αυτοκίνητα ταυτόχρονα και στις δύο λωρίδες του δρόμου και να μην γίνει ο εντοπισμός του ενός από τα δύο.

Σχήμα 3. Σφάλμα εντοπισμού και των δύο αυτοκινήτων λόγω κακής λήψης της κάμερας

Επιπροσθέτως πρέπει να επισημανθεί ότι εφόσον τα βίντεο πληρούν τις προϋποθέσεις που αναφέρθηκαν παραπάνω δεν σημαίνει ότι δεν υπάρχουν περιπτώσεις σφάλματος. Κάθε βίντεο περιέχει κάποιους αστάθμητους παράγοντες οι οποίοι δεν γίνεται να προβλεφθούν. Για παράδειγμα δύο βίντεο μπορεί να έχουν περίπου την ίδια φωτεινότητα αλλά να χρειάζεται να χρησιμοποιήσουμε άλλο κατώφλι(threshold) για το ένα και άλλο για το άλλο για να γίνει σωστά το σκοτείνιασμα του frame. Επίσης δύο βίντεο μπορεί να έχουν τα ίδια χαρακτηριστικά αλλά να χρειάζεται διαφορετικός αριθμός μεγέθους μορφής δίσκου για να χρησιμοποιήσουμε την μορφολογική συνάρτηση ***imopen*** η οποία θα μας απομονώσει τα αυτοκίνητα από το frame.

Πρέπει να γίνει η εισαγωγή του αλγορίθμου στο βίντεο και παρατηρώντας τις αστοχίες που εμφανίζονται να τοποθετούνται οι σωστές τιμές σε διάφορα σημεία της εφαρμογής όπως είναι το κατώφλι και η συνάρτηση ***imopen***.

Εν κατακλείδι η εφαρμογή δουλεύει σωστά για τα βίντεο που πληρούν τις προϋποθέσεις που αναφέρθηκαν παραπάνω απλώς στην συνέχεια θα πρέπει να τροποποιηθούν κάποιες τιμές για να έχουμε το καλύτερο αποτέλεσμα. Για τον λόγο αυτό τοποθετήθηκαν δύο βίντεο στην εφαρμογή και όχι ένα δείχνοντας έτσι ότι για τα δύο βίντεο αυτά που έχουν κάποιες διαφορές μεταξύ τους η εφαρμογή δουλεύει σωστά.

Βιβλιογραφία

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

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

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

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

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

"Deblurring Images: Matrices, Spectra and Filtering". Hansen, Nagy O'Leary.

Society for Industrial and Apllied Mathematics. ISBN 0848716187

Image processing Toolbox από το Matlab

"Image Processing the Fundamentals". Πέτρου Μαρία, Μποσδογιάννη Παναγιώτα

Wiley 1 edition (October 5, 1999). ISBN 0471998834

"Ψηφιακή Επεξεργασία Εικόνας". Πίττας Ιωάννης.
Ιδιωτική Έκδοση 2001