



Τμήμα  
Μηχανικών  
Πληροφορικής τ.ε.  
Τεχνολογικό Εκπαιδευτικό Ίδρυμα  
Δυτικής Ελλάδας

# **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

## **ΘΕΜΑ:**

***Σύστημα αυτόματης ανίχνευσης και  
ειδοποίησης δραστηριότητας σε  
υποπεριοχή βίντεο***

Σπουδαστές: Τζουγανάτος Γρηγόριος  
Κολλιόπουλος Χρήστος

Εισηγητής : Τσακανίκας Βασίλειος

# Σύστημα αυτόματης ανίχνευσης και ειδοποίησης δραστηριότητας σε υποπεριοχή βίντεο

## **Contents**

Σύστημα αυτόματης ανίχνευσης και ειδοποίησης δραστηριότητας σε υποπεριοχή βίντεο .....	1
Κεφάλαιο 1 .....	4
1.1 Εισαγωγή.....	4
Κεφάλαιο 2 Αφαίρεση Φόντου –Κατάτμηση κίνησης.....	9
2.1 Εισαγωγή.....	9
2.2 Κατάτμηση κίνησης με τεχνικές αφαίρεσης φόντου .....	9
2.2.1 Αφαίρεση φόντου μέσω τεχνικής διαδοχικών καρτέ .....	11
2.2.2 Μορφολογικού μετασχηματισμού – Μετά επεξεργασία .....	15
2.2.3 Κατάτμηση κίνησης με χρήση στατιστικών μεθόδων .....	18
Κεφάλαιο 3 Εντοπισμός κινούμενης Οντότητας .....	21
3.1 Εισαγωγή.....	21
3.2 Point tracking – Deterministic method.....	23
3.3 Point tracking – Statistical method.....	25
3.4 Kernel Tracking .....	26
3.5 Silhouette Tracking .....	26
3.6 Φίλτρο Kalman.....	27
3.6.1 Γενικά για το Kalman .....	27
3.6.2 Μαθηματική Επεξήγηση.....	28
3.6.3 Αλγόριθμος .....	32
Κεφάλαιο 4 Υλοποίηση.....	35
4.1 Εισαγωγή στο Matlab.....	35
4.2 Ανάπτυξη εφαρμογής .....	45
4.2.1 Απαιτήσεις Εφαρμογής.....	45
4.2.2 Ανάλυση Εφαρμογής .....	45
4.2.3 Επεξήγηση Κώδικα.....	51
4.2.4 Εκτέλεση κώδικα.....	55
Παράρτημα .....	64
Κώδικας.....	66
Σύνδεση WebCam με Matlab .....	70

Εικόνα 1 - τρία διαδοχικά frames $\alpha$ , $\beta$ , $\gamma$ όπου τα μαύρα εικονοστοιχεία αντιστοιχούν στα εικονοστοιχεία που δεν μετακινήθηκαν, δηλαδή στο φόντο, ενώ τα άσπρα εικονοστοιχεία αντιστοιχούν στα εικονοστοιχεία που άλλαξαν, δηλαδή στο κινούμενο αντικείμενο. ....	12
Εικόνα 2 - Δύο αντικείμενα που κινούνται. $\alpha$ . Το κατώφλι καθορίστηκε με βάση την ταχύτητα των σωμάτων που κινούνται, με αποτέλεσμα το αντικείμενο που κινείται αργά να μη μπορεί ποτέ να ανιχνευθεί. $\beta$ . Το κατώφλι εξαρτάται από ένα προκαθορισμένο καρέ αναφοράς ανεξάρτητα από την ταχύτητα των αντικειμένων. ....	13
Εικόνα 3 - Αποτελέσματα από την χρήση της τεχνικής των διαδοχικών καρέ για τα δύο βίντεο σε διαφορετικά χρονικά καρέ.....	14
<b>Εικόνα 4 - Αποτελέσματα από την χρήση της τεχνικής των στατιστικών μεθόδων για</b> .....	19
Εικόνα 5 - Τεχνικές παρατήρησης κίνησης αντικειμένου .....	22
Εικόνα 6 - Παραδείγματα εφαρμογής διαφορετικών μεθόδων για την παρακολούθηση κινούμενου αντικειμένου.....	23
Εικόνα 7 - Αντιστοίχιση σημείων δύο διαφορετικών καρέ.....	23
Εικόνα 8 - Επεξηγηματική αναπαράσταση παραδοχών .....	24
Εικόνα 9 - Παραδείγματα εφαρμογών με χρήση του deterministic method .....	25
Εικόνα 10 -Απολογισμός μεθόδου silhouette tracking .....	27
Εικόνα 11 - Αλγόριθμος Kalman .....	33
Εικόνα 12 - Matlab Desktop (Επιφάνεια Εργασίας Matlab).....	36
Εικόνα 13 - Παράθυρο Εντολών (Command Window).....	37
Εικόνα 14 - Ιστορικό Εντολών (Command History) .....	38
Εικόνα 15 - Χώρος εργασίας(Workspace).....	39
Εικόνα 16 - Τρέχων Κατάλογος (Current Directory) .....	40
Εικόνα 17 - Παράθυρο Εισαγωγής Κειμένου (Editor window).....	41
Εικόνα 18 - Παράθυρο Σχημάτων (Figure window) .....	42
Εικόνα 19 - Παράθυρο Βοήθειας (Help Window) .....	43
Εικόνα 20 - Παράδειγμα χρήσης του Matlab στην επεξεργασία εικόνας.....	44
Εικόνα 21 - Ορισμός Υποπεριοχής.....	46
Εικόνα 22 - Διάγραμμα ροής.....	50
Εικόνα 23 -'Όχημα εντός υποπεριοχής.....	56
Εικόνα 24 -'Όχημα εντός υποπεριοχής.....	57
Εικόνα 25 - 'Όχημα εκτός υποπεριοχής.....	58
Εικόνα 26 - 'Όχημα εκτός υποπεριοχής.....	59
Εικόνα 27 - 'Όχημα εντός υποπεριοχής.....	60
Εικόνα 28 - 'Όχημα εντός υποπεριοχής.....	61
Εικόνα 29- 'Όχημα εκτός υποπεριοχής .....	62
Εικόνα 30 - 'Όχημα εκτός υποπεριοχής.....	63

# Κεφάλαιο 1

## 1.1 Εισαγωγή

Η αναγνώριση και παρατήρηση ενός κινούμενου αντικειμένου αποτελεί ένα σημαντικό κεφάλαιο στο πεδίο της υπολογιστικής όρασης. Η αύξηση της υπολογιστικής ισχύος των μηχανών και η διάθεση ολοένα και υψηλότερης ποιότητας και χαμηλού κόστους συσκευών λήψης, σε συνδυασμό με την αυξανόμενη ανάγκη για αυτοματοποιημένη ανάλυση βίντεο έχει δημιουργήσει μεγάλο ενδιαφέρον στην επιστημονική κοινότητα για την ανάπτυξη αλγορίθμων που θα προσομοιώνουν την ανθρώπινη αντίληψη αναγνώρισης της κίνησης. Η προσομοίωση αυτή βέβαια δεν είναι ακόμα ώριμη να γίνει σε ποιοτικά χαρακτηριστικά και σε καμία περίπτωση δεν είναι σε θέση να προσεγγίσει την πολυπλοκότητα των ανθρώπινων νευρώνων στην αντιμετώπιση μιας υπόθεσης απλής αναγνώρισης και παρακολούθησης κίνησης. Παρόλα αυτά γίνεται μεγάλη προσπάθεια προσέγγισης του προβλήματος κάνοντας χρήση άλλων χαρακτηριστικών.

Παρόλο που οι μηχανές έχουν σημαντικά πλεονεκτήματα όσο αναφορά την ταχύτητα της επεξεργασίας πολύπλοκων δεδομένων και την αποθήκευση και γρήγορη ανάκτηση μεγάλου όγκου πληροφοριών αδυνατούν να προσεγγίσουν τα ποιοτικά χαρακτηριστικά που ακολουθεί ο ανθρώπινος εγκέφαλος προκειμένου να επιτελέσει την διαδικασία της αναγνώρισης και παρακολούθησης τροχιάς ενός αντικειμένου ακόμα και κάτω από περιπτώσεις κακής ορατότητας ή πολλών δευτερογενών αντικειμένων που σταδιακά κρύβουν το αντικείμενο που είναι προς παρακολούθηση. Η ανθρώπινη αντίληψη σε τέτοιες περιπτώσεις ενεργοποιείται αυτόματα, χωρίς κάποια προσπάθεια, και η περίπτωση λάθους είναι πολύ μικρή. Η αντίληψη αυτή προέρχεται μέσω ερεθισμάτων που λαμβάνουμε από τις αισθήσεις μας και στην συγκεκριμένη περίπτωση μέσω του αισθητήριου οργάνου της όρασης, δηλαδή του ματιού. Στην περίπτωση των υπολογιστικών συστημάτων η αίσθηση της όρασης εξομοιώνεται καταγράφοντας ακολουθίες εικόνων μέσω μιας ή περισσοτέρων καμερών. Στα πλαίσια αυτά λοιπόν, μια μηχανή συνδεδεμένη με μια απλή κάμερα είναι σε θέση να "βλέπει". Όπως όμως αναφέραμε και παραπάνω η μεγάλη διαφορά έγκειται στο γεγονός ότι ο ανθρώπινος οργανισμός είναι σε θέση να αντιλαμβάνεται άμεσα αυτά που βλέπει, ενώ το υπολογιστικό σύστημα θα πρέπει να εκτελέσει μια

σειρά από σύνθετες λειτουργίες προκειμένου να φτάσει σε χειρότερα ή στην καλύτερη περίπτωση στα ίδια συμπεράσματα.

Για να πάρει, δηλαδή, μια απόφαση θα πρέπει να εξάγει κάποια χαρακτηριστικά από τις εικόνες που δέχεται, βάσει των οποίων θα μπορέσει να διακρίνει, να αναγνωρίσει και να χαρακτηρίσει αυτά τα οποία βλέπει. Στη προσπάθεια λοιπόν ένα σύστημα να κατανοήσει αυτά που βλέπει μέσω της μηχανικής όρασης, δέχεται σε πρώτη φάση τα δεδομένα εισόδου, τα οποία πριν από το στάδιο της επεξεργασίας τους δεν έχουν καμία πρακτική χρησιμότητα στην αντίληψη της μηχανής. Δεν είναι παρά οι συντεταγμένες των εικονοστοιχείων των εικόνων στο χώρο των χρωμάτων. Με βάση μόνο τη πληροφορία αυτή, το σύστημα καλείται να επιτελέσει υψηλού επιπέδου εργασίες που σε κάποιο βαθμό να φανερώνουν την αντίληψή του. Η μεθοδολογία που ακολουθείται στη λύση τέτοιων προβλημάτων (ώστε να επιτελεστούν οι εργασίες αυτές στη γενική περίπτωση) εξαρτάται από ένα σύνολο παραμέτρων που έχουν να κάνουν με τα επιθυμητά αποτελέσματα εξόδου ως προς τη φύση τους, την πολυπλοκότητά τους, και την ακρίβεια με την οποία θέλουμε να λαμβάνονται και με ποικίλους άλλους τρόπους.

Έτσι εντελώς διαφορετική προσέγγιση θα υιοθετήσουμε για παράδειγμα όταν θέλουμε να μετρήσουμε το ποσοστό ενός χρώματος σε μία εικόνα, διαφορετική όταν θέλουμε να ανιχνεύσουμε κινούμενες οντότητες σε μια ακολουθία εικόνων και διαφορετική όταν θέλουμε να υπολογίσουμε την σχετική απόσταση μεταξύ δύο στόχων. Στη πράξη, ένα σύστημα λέμε ότι αντιλαμβάνεται τα περιεχόμενα μιας τέτοιας πληροφορίας όταν είναι σε θέση να μετασχηματίσει τη χρωματική πληροφορία των ψηφιακών εικόνων σε κάποια άλλη μορφή πληροφορίας με περισσότερο σημασιολογικό περιεχόμενο. Για παράδειγμα, ο χωρισμός μιας εικόνας σε χρωματικές περιοχές, η εξαγωγή των ακμών της και η εκτίμηση της κίνησης των εικονοστοιχείων από καρέ σε καρέ, αποτελούν κάποια αρχικά στοιχεία που μπορεί κάποιος να εξάγει και που περιέχουν πληροφορία χρήσιμη για τα επόμενα στάδια επεξεργασίας. Τα στοιχεία αυτά ονομάζονται γνωρίσματα (features).

Γενικά, οι μέθοδοι εξαγωγής γνωρισμάτων χρησιμοποιούν παρεμφερείς μεθόδους. Αρχικά, ένα ή περισσότερα γνωρίσματα χαμηλού επιπέδου εξάγονται από τις εικόνες, όπως το χρώμα, η υφή, το σχήμα, η κίνηση κλπ. με χρήση κάποιας εκ των προτέρων γνώσης, γενικής ή ειδικότερης ανάλογα με την εφαρμογή και στη συνέχεια εξάγεται πληροφορία υψηλότερου επιπέδου υπολογίζοντας νέα χαρακτηριστικά με

μεγαλύτερη σημασιολογική πληροφορία. Τα χαρακτηριστικά αυτά μπορούν με τη σειρά τους να χρησιμοποιηθούν για να χαρακτηρίσουν μια εικόνα, μια ακολουθία βίντεο ή ακόμη και αντικείμενα των εικόνων, όπως την απόσταση ενός στόχου από τη κάμερα, το τρισδιάστατο μοντέλο ενός αντικειμένου ή τον προσδιορισμό της φύσης μιας κινούμενης οντότητας. Κάποια από τα πιο βασικά γνώρισμα χαμηλού επιπέδου που επεξεργάζονται κατά κόρον σε εφαρμογές του ενδιαφέροντος μας είναι τα εξής :

- **Χρώμα** : Αποτελεί ίσως την πιο άμεσα εξαγόμενη πληροφορία μιας εικόνας. Όπως τα υπόλοιπα γνώρισμα, έτσι και το χρώμα μπορεί να μοντελοποιηθεί με διάφορους τρόπους. Στη γενικότερη περίπτωση, προσπαθούμε η μοντελοποίηση που θα κάνουμε να περιέχει όσο το δυνατόν περισσότερη πληροφορία, ανάλογα με την εφαρμογή. Έτσι το χρώμα μοντελοποιείται συνήθως από έναν μικρό αριθμό βασικών χρωμάτων (RGB) ή μέσω του χρωματικού ιστογράμματος.

- **Υφή** : Όμοια με το χρώμα αντιμετωπίζεται και η υφή μιας εικόνας. Διάφορα μοντέλα έχουν προταθεί για τη μοντελοποίηση της καθώς είναι ένα μέγεθος που συνδέεται άμεσα με την εγκυρότητα των υπολογισμών μας, ειδικά σε περιπτώσεις εκτίμησης κίνησης.

- **Σχήμα** : Ενώ μπορεί γενικά να θεωρηθεί ως γνώρισμα υψηλότερου επιπέδου, μοντελοποιείται και αυτό σε χαμηλό επίπεδο με βάση τις ιδιότητες της εικόνας όπως για παράδειγμα τις ακμές, το εμβαδό, τη θέση κλπ. Το περίγραμμα αντικειμένων ενδιαφέροντος μιας εικόνας εξάγεται και μοντελοποιείται σε υψηλότερο επίπεδο.

- **Κίνηση** : Αποτελεί πολύ σημαντικό γνώρισμα για τις περισσότερες εφαρμογές του κλάδου και εξάγεται από δύο ή περισσότερα καρέ μιας ακολουθίας βίντεο. Μέχρι και σήμερα, έχει αναπτυχθεί ένας μεγάλος αριθμός μεθόδων για τον υπολογισμό της κίνησης καθώς η εκτίμηση της αποτελεί βάση πολλών εφαρμογών, όπως για παράδειγμα ο υπολογισμός της τρισδιάστατης κίνησης ενός αντικειμένου και η παρακολούθηση τροχιάς.

- **Πρότυπα** : Μια συνηθισμένη τακτική σε κάποιες από τις κατηγορίες εφαρμογών είναι η εξαγωγή προτύπων αντί χαρακτηριστικών από ακολουθίες εικόνων τα οποία στη συνέχεια χρησιμοποιούνται αυτούσια, μετασχηματισμένα ή και παραμορφωμένα σε διαδικασίες εντοπισμού και αναγνώρισης. Ως παράδειγμα αναφέρουμε την εξαγωγή προτύπων από εικόνες με ανθρώπινα πρόσωπα.

- **Μετασχηματισμοί** : Ενδιαφέροντα γνωρίσματα για εφαρμογές μηχανικής όρασης προκύπτουν και από το μετασχηματισμό των εικόνων ή συγκεκριμένων περιοχών τους. Για παράδειγμα ο μετασχηματισμός Fourier δίνει πληροφορίες για το συχνοτικό περιεχόμενο των εικόνων.

Σε πάρα πολλές εφαρμογές, όπως για παράδειγμα τον εντοπισμό εισβολέων σε κατοικία ή τη παρακολούθηση της τροχιάς κινούμενων αντικειμένων χρησιμοποιείται ως πρώτο στάδιο ο εντοπισμός κύριων κινούμενων αντικειμένων.

Σε τέτοιες περιπτώσεις χρησιμοποιούνται συνήθως γνωρίσματα κίνησης, σχήματος ή και συνδυασμός των δυο (π.χ. κίνηση περιοχών ή ακμών). Όταν είναι επιθυμητή η λεπτομερής εξαγωγή της μορφής κινούμενων αντικειμένων, χαρακτηριστικά κίνησης συνδυάζονται συνήθως με γνωρίσματα χρώματος για πιο έγκυρα αποτελέσματα.

Σε εφαρμογές που στόχο έχουν την αναγνώριση και ταξινόμηση αντικειμένων τα γνωρίσματα σχήματος είναι περισσότερο κατάλληλα και ειδικότερα τα υψηλού επιπέδου χαρακτηριστικά όπως το περίγραμμα. Στη περίπτωση που μελετάται ο διαχωρισμός δύσκαμπτων και εύκαμπτων αντικειμένων, τα χαρακτηριστικά κίνησης είναι επίσης πολύ χρήσιμα και καταλήγουν σε σαφείς διαχωρισμούς με μικρά επίπεδα σφάλματος. Βεβαίως, όταν τα υπό αναγνώριση αντικείμενα έχουν ιδιότητες που τα διαφοροποιούν στο πεδίο του χρώματος, της συχνότητας ή της υφής δεν αποκλείεται και η χρήση των γνωρισμάτων αυτών, αν και η περιοχή εφαρμογών που καλύπτουν είναι στη πραγματικότητα αμελητέα.

## **1.2 Σκοπός**

Σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος αυτόματης ανίχνευσης και ειδοποίησης δραστηριότητας σε υποπεριοχή βίντεο. Το πρόγραμμα είτε θα φορτώνει κάποιο βίντεο είτε θα έχει ως είσοδο ροή δεδομένων από κάμερα με στόχο την ανίχνευση κίνησης σε συγκεκριμένη υποπεριοχή και εμφάνιση ειδοποίησης σε περίπτωση όπου το αντικείμενο είναι έξω από τα όρια της υποπεριοχής.

## **1.3 Δομή**

Στο δεύτερο κεφάλαιο παρουσιάζεται η έννοια και τεχνικές που εφαρμόζονται στην κατάτμηση της κίνησης. Πιο συγκεκριμένα θα παρουσιαστεί η κατάτμηση κίνησης με

την χρήση τεχνικών αφαίρεσης φόντου, τεχνικών διαδοχικών καρτέ και τη χρήση στατιστικών μεθόδων.

Στο τρίτο κεφάλαιο, αναφέρονται οι τεχνικές που χρησιμοποιούνται για τον εντοπισμό και την παρακολούθηση της κίνησης οι οποίες ποικίλουν ανάλογα την εφαρμογή και η χρήση τους εξαρτάται από πολλούς παράγοντες όπως το μέγεθος του αντικειμένου, το περιβάλλον στο οποίο δραστηριοποιείτε, τις απαιτήσεις στην ποιότητα της παρακολούθησης κ.α.

Στο τέταρτο κεφάλαιο παρουσιάζεται η εφαρμογή η οποία αναπτύχθηκε και αναλύονται τα κυριότερα στάδια.



## **Κεφάλαιο 2 Αφαίρεση Φόντου –Κατάτμηση κίνησης**

### **2.1 Εισαγωγή**

Σχεδόν κάθε σύστημα που έχει ως στόχο την ανίχνευση κίνησης ξεκινά με την ανίχνευση ή αλλιώς κατάτμηση της κίνησης σε μια ακολουθία βίντεο. Αυτή η ανίχνευση αποσκοπεί στον διαχωρισμό των περιοχών που περιέχουν κινούμενες οντότητες από την υπόλοιπη εικόνα. Είναι ένα πολύ σημαντικό βήμα για να προχωρήσει κάποιος σε περαιτέρω ανάλυση της ανθρώπινης δραστηριότητας καθώς οι διαδικασίες που ακολουθούν εξαρτώνται σε μεγάλο βαθμό από τα αποτελέσματα του.

Πρόκειται λοιπόν για ένα πολύ σημαντικό και απαιτητικό πρόβλημα, με στόχο την ανίχνευση των περιοχών εκείνων που αντιστοιχούν σε κινούμενα αντικείμενα, όπως οχήματα, άνθρωποι κ.α.. Ο εντοπισμός κινούμενων οντοτήτων επικεντρώνει ουσιαστικά τη προσοχή μας στις περιοχές που μας ενδιαφέρουν για τα επόμενα στάδια επεξεργασίας, όπως η παρακολούθηση της πορείας των αντικειμένων και η ανάλυση των δραστηριοτήτων τους. Παρόλα αυτά, αλλαγές λόγω καιρού, φωτισμού, σκιών και επαναλαμβανόμενων κινήσεων από ανεπιθύμητα αντικείμενα δυσκολεύουν την γρήγορη και αξιόπιστη επεξεργασία των δεδομένων.

Μέχρι σήμερα, οι δημοφιλέστερες μέθοδοι κατάτμησης χρησιμοποιούν είτε χρονική είτε χωρική πληροφορία που εξάγουν από τα διαδοχικά καρέ. Οι μέθοδοι που μπορούν να χρησιμοποιηθούν είναι πολλές κάθε μία εκ των οποίων έχει διαφορετικά πλεονεκτήματα και μειονεκτήματα. Γενικά όσο πιο ελαφριά είναι μία μέθοδος από άποψη υπολογιστικού κόστους τόσο χειρότερα είναι τα εξαγόμενα αποτελέσματα ενώ αντίθετα όσο περισσότερες πράξεις γίνονται και όσο πιο πολλές παράμετροι εξετάζονται τόσο πιο λεπτομερή είναι τα εξαγόμενα αποτελέσματα.

[1][2][3]

### **2.2 Κατάτμηση κίνησης με τεχνικές αφαίρεσης φόντου**

Η τεχνική αυτή είναι μία πολύ δημοφιλής μέθοδος για τον εντοπισμό των κινούμενων περιοχών, ειδικά κάτω από συνθήκες σχετικά σταθερού φόντου. Λειτουργεί

υπολογίζοντας τις διαφορές ανάμεσα στη τρέχουσα εικόνα και το φόντο αναφοράς για κάθε εικονοστοιχείο ξεχωριστά. Το πρόβλημα της είναι ότι παρουσιάζει πολύ μεγάλη ευαισθησία σε δυναμικές αλλαγές του σκηνικού λόγω φωτισμού και άλλων εξωγενών παραγόντων.

Υπάρχουν πολλές διαφορετικές προσεγγίσεις αυτού του προβλήματος οι οποίες διαφοροποιούνται στο τύπο του μοντέλου που χρησιμοποιούν για το φόντο καθώς και στον τρόπο με τον οποίο το ανανεώνουν. Η τρέχουσα εικόνα συγκρίνεται με ένα μοντέλο του φόντου το οποίο δεν περιέχει τα κινούμενα αντικείμενα. Συνήθως, αυτό το μοντέλο του φόντου ανακτάται με την πάροδο του χρόνου. Το πλεονέκτημα τέτοιων προσαρμόσιμων μεθόδων είναι ότι αφαιρούν το φόντο από τις εικόνες. Συνήθως με τον όρο αφαίρεση φόντου εννοούμε την απομάκρυνση των στατικών αντικειμένων έτσι ώστε να παραμείνει μόνο η περιοχή ενδιαφέροντος μας που είναι τα κινούμενα άτομα σε μια ακολουθία εικόνων.

Η βασική ιδέα είναι ο υπολογισμός της απόλυτης διαφοράς των εικόνων δύο διαδοχικών στιγμών, Δηλαδή έστω ότι έχουμε  $N$  καρέ για να βρούμε τα άτομα που κινούνται μεταξύ των καρέ  $N-1$  και  $N$ , δεν έχουμε παρά να υπολογίσουμε την απόλυτη διαφορά τους και να προκύψει σαν στατικό (μαύρο) οτιδήποτε είναι ακίνητο ενώ οτιδήποτε άλλο θα είναι κινούμενο αντικείμενο. Μην ξεχνάμε ότι οι υπολογισμοί αυτοί γίνονται σε εικόνες και στην ουσία αυτό που υπολογίζουμε είναι η διαφορά των τιμών των εικονοστοιχείων σε κάθε καρέ, οπότε με βάση αυτά που είπαμε παραπάνω οτιδήποτε ακίνητο θα μας επιστρέψει την τιμή μηδέν μιας και οι τιμές του εικονοστοιχείου αυτού δεν άλλαξαν κατά τις δύο χρονικές στιγμές. Αυτό σημαίνει ότι τα αντικείμενα που ήταν στην σκηνή κατά την εξαγωγή του φόντου βαθμιαία θα αντικατασταθούν από το φόντο. Αυτό εξαλείφει την ανάγκη της αρχικοποίησης ενός άδειου φόντου. Επίσης, παρέχει ένα μοντέλο φόντου που προσαρμόζεται αυτόματα στις αλλαγές. Για παράδειγμα, όταν αλλάζει ο καιρός ή όταν ένα παρκαρισμένο αυτοκίνητο φεύγει από τη σκηνή.

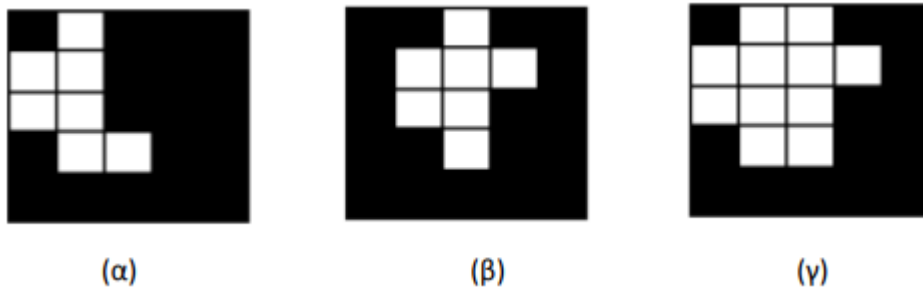
Ένα σημαντικό μειονέκτημα είναι η αλληλεξάρτηση (trade-off) μεταξύ δύο αντιτιθέμενων απαιτήσεων. Από τη μία, η ενημέρωση του φόντου θα πρέπει να γίνεται γρήγορα ώστε να μπορεί να αντιμετωπίσει τις αλλαγές φωτισμού και τις αλλαγές στο φόντο, όπως αλλαγές του φωτισμού λόγω αλλαγής της ώρας ή λόγω ανοίγματος του διακόπτη σε κλειστό χώρο, αλλαγές λόγω ανθρώπων που περπατούν ή λόγω αντικειμένων που φεύγουν από τη σκηνή. Από την άλλη, η ενημέρωση θα

πρέπει να γίνεται αργά, ώστε τα αντικείμενα που κινούνται αργά να μην περιλαμβάνονται στο φόντο, όπως ένας άνθρωπος που κοιμάται. Η ταχύτητα λοιπόν, της ενημέρωσης εξαρτάται από την εφαρμογή. Παρακάτω αναφέρονται πέντε αλγόριθμοι για την εξαγωγή του φόντου και παρατίθενται τα αποτελέσματα της εφαρμογής τους σε ακολουθίες βίντεο. Χαρακτηριστικά παραδείγματα της τεχνικής αφαίρεσης φόντου περιγράφονται παρακάτω.

### **2.2.1 Αφαίρεση φόντου μέσω τεχνικής διαδοχικών καρτέ**

Αποτελεί τον πιο απλός ίσως τρόπο για τον διαχωρισμό του φόντου από το κινούμενο αντικείμενο σε ασπρόμαυρο βίντεο με στατική κάμερα είναι να συγκρίνουμε ανά δύο τα καρτέ του βίντεο παίρνοντας τη διαφορά τους. Αυτό γίνεται ως εξής :

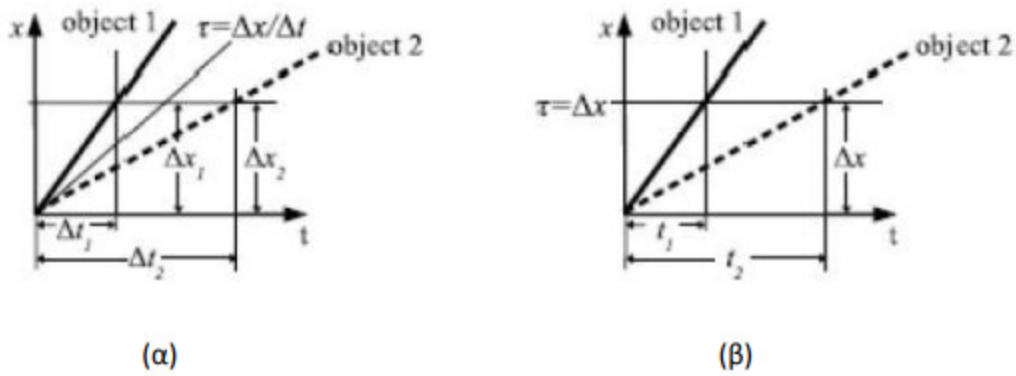
1. Ορίζουμε έναν πίνακα με διαστάσεις ίδιες με αυτές των καρτέ τον οποίο αρχικοποιούμε.
2. Για κάθε δύο διαδοχικά καρτέ παίρνουμε κάθε εικονοστοιχείο και βρίσκουμε τη διαφορά της τιμής του μεταξύ των δύο καρτέ.
3. Ορίζουμε ένα κατώφλι (threshold) ( $0 < \tau < 1$ )
4. Αν η διαφορά των τιμών του εικονοστοιχείου είναι μικρότερη της τιμής του κατωφλίου, τότε βάζουμε στην αντίστοιχη θέση του πίνακα την τιμή 1, ενώ αν η διαφορά είναι μεγαλύτερη του κατωφλίου, τότε στην αντίστοιχη θέση βάζουμε την τιμή 0. Έτσι, από κάθε δύο διαδοχικά καρτέ παίρνουμε μία ασπρόμαυρη εικόνα όπου τα μαύρα εικονοστοιχεία αντιστοιχούν στα εικονοστοιχεία που δεν μετακινήθηκαν, δηλαδή στο φόντο, ενώ τα άσπρα εικονοστοιχεία αντιστοιχούν στα εικονοστοιχεία που άλλαξαν, δηλαδή στο κινούμενο αντικείμενο. Στο παρακάτω σχήμα φαίνεται ένα παράδειγμα, όπως αυτό περιγράφηκε παραπάνω με κάποια υποτιθέμενα διαδοχικά frames, ώστε να οπτικοποιηθεί η περιγραφεί.



Εικόνα 1 - τρία διαδοχικά frames α, β, γ όπου τα μαύρα εικονοστοιχεία αντιστοιχούν στα εικονοστοιχεία που δεν μετακινήθηκαν, δηλαδή στο φόντο, ενώ τα άσπρα εικονοστοιχεία αντιστοιχούν στα εικονοστοιχεία που άλλαξαν, δηλαδή στο κινούμενο αντικείμενο.

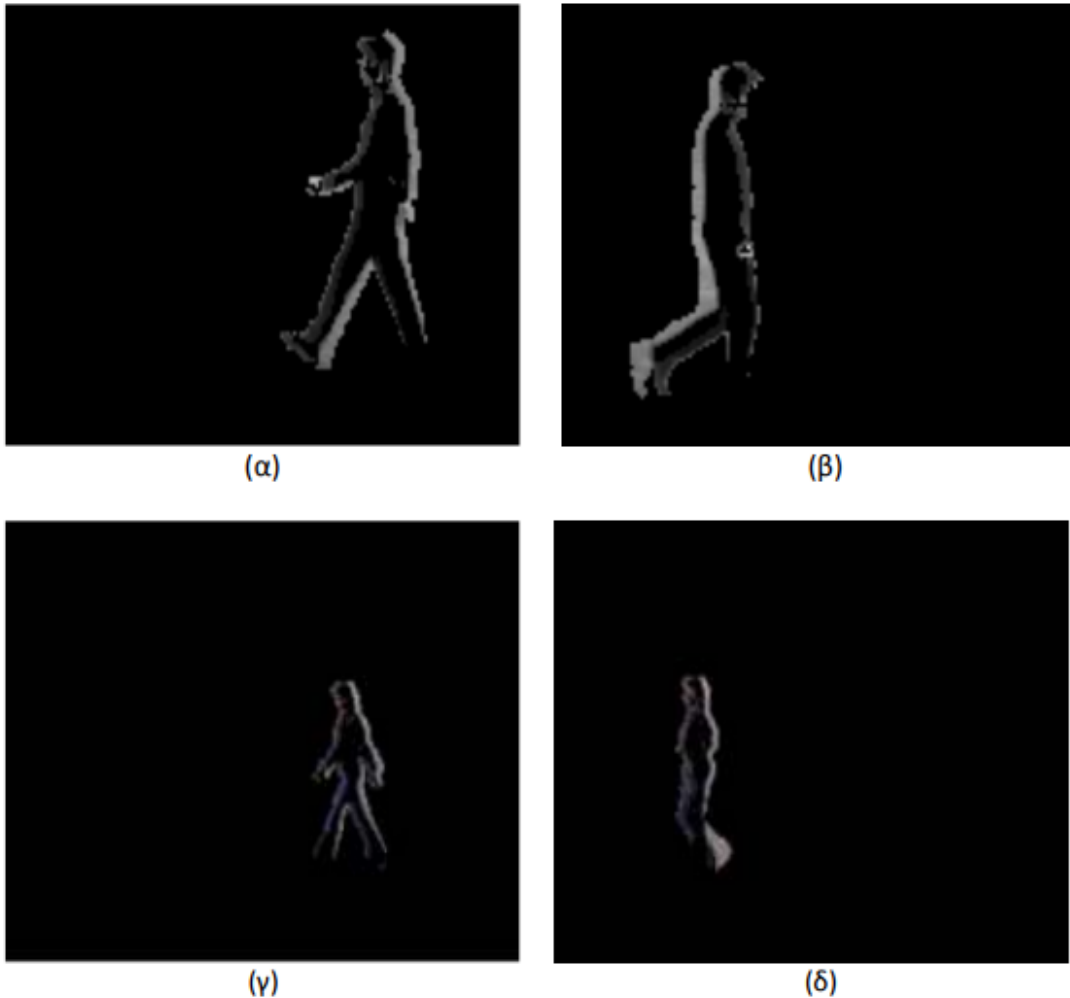
Με αυτόν τον τρόπο όμως, προκύπτουν διάφορα προβλήματα. Καταρχήν, χρειάζεται μεγάλη προσοχή στην τιμή που θα βάλουμε στο κατώφλι. Αν η τιμή είναι πολύ μικρή, τότε η παραμικρή κίνηση θα ανιχνευθεί, πράγμα που μπορεί να οδηγήσει σε εσφαλμένες εκτιμήσεις, διότι μπορεί να έχουμε μικρές αλλαγές στο φόντο, οι οποίες δε θέλουμε να θεωρηθούν ως κινούμενα αντικείμενα.

Τέτοιες αλλαγές μπορεί να οφείλονται σε θόρυβο λόγω της κάμερας, σε αλλαγές φωτισμού, σε σκιές, σε αέρα αν είμαστε σε εξωτερικό περιβάλλον, κ.ά. Απ' την άλλη, αν η τιμή του κατωφλίου είναι πολύ μεγάλη, τότε κάποιες αλλαγές μπορεί να μην ανιχνευθούν καθόλου, όπως για παράδειγμα ένα αντικείμενο που κινείται με πολύ μικρή ταχύτητα. Στα παρακάτω διαγράμματα βλέπουμε ένα αντικείμενο που κινείται γρήγορα (object 1) και ένα που κινείται αργά (object 2). Στο διάγραμμα που βρίσκεται αριστερά, το κατώφλι  $\tau$  που επιλέχτηκε, καθορίστηκε με βάση την ταχύτητα των σωμάτων που κινούνται, με αποτέλεσμα το αντικείμενο που κινείται αργά να μη μπορεί ποτέ να ανιχνευθεί ( $\tau = \Delta x / \Delta t$ ). Αντίθετα, στο διάγραμμα που βρίσκεται στα δεξιά, το κατώφλι  $\tau$  εξαρτάται από ένα προκαθορισμένο καρέ αναφοράς ( $\tau = \Delta x$ ). Αυτό οδηγεί στην ανίχνευση και των δύο αντικειμένων, ανεξάρτητα από την ταχύτητά τους.



Εικόνα 2 - Δύο αντικείμενα που κινούνται. α. Το κατώφλι καθορίστηκε με βάση την ταχύτητα των σωμάτων που κινούνται, με αποτέλεσμα το αντικείμενο που κινείται αργά να μη μπορεί ποτέ να ανιχνευθεί. β. Το κατώφλι εξαρτάται από ένα προκαθορισμένο καρέ αναφοράς ανεξάρτητα από την ταχύτητα των αντικειμένων.

Δυστυχώς η απλότητα της μεθόδου έχει δύο πολύ μεγάλα μειονεκτήματα, πρώτον ότι για αντικείμενα με ομοιόμορφη κατανομή της τιμής της πυκνότητας τους τα εσωτερικά εικονοστοιχεία τους ερμηνεύονται ως μέρη του φόντου και δεύτερον ότι τα αντικείμενα πρέπει να κινούνται συνεχώς ειδιάλλως στην περίπτωση που το αντικείμενο παραμένει ακίνητο για χρονικό διάστημα ενός καρέ ερμηνεύεται και πάλι σαν μέρος του φόντου. Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα:



Εικόνα 3 - Αποτελέσματα από την χρήση της τεχνικής των διαδοχικών καρέ για τα δύο βίντεο σε διαφορετικά χρονικά καρέ

Παρατηρώντας τα παραπάνω αποτελέσματα διαπιστώνουμε ότι η συγκεκριμένη μέθοδος είναι πολύ ευαίσθητη και αυτό λόγω των απλοποιημένων υποθέσεων που κάναμε. Τα αποτελέσματα, αν και αναγνωρίζεται η ανθρώπινη φιγούρα, δεν είναι καλά και σε πολλές περιπτώσεις δεν μπορούν να αναγνωριστούν με σιγουριά οι κινήσεις των άκρων των ανθρώπων. Χαρακτηριστικό είναι ότι τα κινούμενα αντικείμενα αφήνουν πίσω τους ίχνη τα οποία ποικίλουν ανάλογα με την ταχύτητά τους.

Επίσης, μειονέκτημα της μεθόδου αποτελεί το γεγονός ότι το εσωτερικό των κινούμενων αντικειμένων αναγνωρίζεται από την μέθοδο ως φόντο και αφαιρείται και αυτό διότι μεταξύ δύο διαδοχικών καρέ δεν έχει προλάβει το αντικείμενο να απομακρυνθεί αρκετά έτσι ώστε στην ίδια περιοχή, τις δύο διαφορετικές χρονικές

στιγμές , να υπάρχουν εικονοστοιχεία με ίδιο χρωματικό περιεχόμενο. Ένα καλό φυσικά που έχει η μέθοδος είναι ο περιορισμένος θόρυβος του φόντου.

[1][2][3]

### **2.2.2 Μορφολογικού μετασχηματισμού – Μετά επεξεργασία**

Η τεχνική των μορφολογικών μετασχηματισμών δεν αποτελεί ουσιαστικά διαφορετική μέθοδο και μπορεί να χρησιμοποιηθεί για ενσωμάτωση σε κάθε μία από τις προηγούμενες μεθόδους βελτιώνοντας με αυτό τον τρόπο τα αποτελέσματα που παίρνουμε.

Μπορεί να αποτελέσει λοιπόν ένα πολύ σημαντικό κομμάτι στη κατάτμηση κίνησης που βελτιώνει θεαματικά τα αρχικά αποτελέσματα όλων των παραπάνω τεχνικών. Πιο συγκεκριμένα, σε κάθε καρτέ, αφού υπολογίσουμε την δυαδική εικόνα πρώτου πλάνου, που έχει τιμή μονάδα για όλα τα κινούμενα εικονοστοιχεία σύμφωνα με την εκατοστέ μέθοδο, επιδρούμε πάνω της με μια σειρά μετασχηματισμών ώστε αυτή να γίνει πιο ομοιόμορφη. Οι μετασχηματισμοί αυτοί έχουν στόχο να σβήσουν από την εικόνα όσες ομάδες εικονοστοιχείων είναι πολύ μικρές για να παριστάνουν κάποιο αντικείμενο ενδιαφέροντος και παράλληλα γεμίζουν τυχόν τρύπες στις κινούμενες οντότητες που προέρχονται από κομμάτια ομοιόμορφης τιμής της έντασης. Επιπλέον εκτός από την εξάλειψη των μικρών περιοχών, οι οποίες αποτελούν θόρυβο, μπορούμε να χρησιμοποιήσουμε μορφολογικούς μετασχηματισμούς για εξάλειψη και μεγάλων περιοχών που δεν είναι πιθανό να αντιπροσωπεύουν κάποια ανθρώπινη φιγούρα αλλά μπορεί να είναι κάποιο άλλο κινούμενο αντικείμενο όπως για παράδειγμα κάποιο αυτοκίνητο.

Η μείωση των περιοχών θορύβου στηρίζεται στην παρατήρηση ότι οι περιοχές θορύβου τείνουν να είναι μικρότερες από την περιοχή του αντικειμένου. Πρώτα, εφαρμόζεται ο κλασικός αλγόριθμος των συνδεδεμένων συστατικών (connected component algorithm) στην αρχική μάσκα αντικειμένου για να σημειωθούν οι απομονωμένες περιοχές. Στη συνέχεια, υπολογίζεται η έκταση κάθε περιοχής. Περιοχές με έκταση μικρότερη από κάποια τιμή κατωφλίου εξαλείφονται από την μάσκα αντικειμένου. Αντίστοιχα περιοχές μεγαλύτερες από μία προκαθορισμένη τιμή μπορούν να εξαλειφθούν επίσης. Έτσι, διατηρείται το σχήμα του αντικειμένου, ενώ μικρές περιοχές θορύβου ή μεγάλες περιοχές που αντιπροσωπεύουν κάποιο άλλο

κινούμενο αντικείμενο αφαιρούνται. Εφόσον υπάρχουν δύο ήδη θορύβου, θόρυβος στο φόντο και θόρυβος στο προσκήνιο, χρειάζονται δύο περάσματα σε αυτό το στάδιο. Το πρώτο πέρασμα αφαιρεί μικρές μαύρες περιοχές (περιοχές φόντου), οι οποίες είναι περιοχές θορύβου στο προσκήνιο ή ‘τρύπες’ στη μάσκα ανίχνευσης αλλαγών. Το δεύτερο πέρασμα αφαιρεί μικρές άσπρες περιοχές (περιοχές στο προσκήνιο), οι οποίες είναι περιοχές θορύβου στο φόντο ή λανθασμένες περιοχές αλλαγών στη μάσκα ανίχνευσης αλλαγών. Μετά την αφαίρεση των περιοχών θορύβου, εφαρμόζονται στη μάσκα αντικειμένου οι μορφολογικές πράξεις (morphological operations) ανοίγματος και κλεισίματος (open and close operations) με ένα 3x3 δομικό στοιχείο (structural element). Το μικρό δομικό στοιχείο επιλέγεται για τη λείανση του περιγράμματος του αντικειμένου, χωρίς να επηρεάζει τις λεπτομέρειες της πληροφορίας του σχήματος του αντικειμένου. Οι μορφολογικές πράξεις είναι η βάση για την απομάκρυνση του θορύβου μετά τον διαχωρισμό του φόντου και την κατωφλιοποίηση. Πιο συγκεκριμένα, οι μορφολογικές τεχνικές που συνήθως χρησιμοποιούνται, αποτελούνται από δύο βασικές πράξεις : την διεύρυνση (dilation) και τη διάβρωση (erosion). Η διεύρυνση επεκτείνει το προσκήνιο της εικόνας, προσθέτοντας ένα εικονοστοιχείο στο προσκήνιο αν οποιοδήποτε από τα γειτονικά του εικονοστοιχείου εντός μιας συγκεκριμένης ‘γειτονιάς’ ακτίνας  $r$  (η οποία λέγεται δομικό στοιχείο – structuring element) ανήκει ήδη στο προσκήνιο. Η διάβρωση επεκτείνει το φόντο, μετακινώντας ένα εικονοστοιχείο από το προσκήνιο αν ένα από τα γειτονικά του εικονοστοιχείου ανήκει ήδη στο φόντο. Αυτές οι δύο πράξεις μπορούν να συνδυαστούν. Μία διεύρυνση ακολουθούμενη από μία πανομοιότυπη διάβρωση λέγεται ‘κλείσιμο’ (closing), και γεμίζει τις τρύπες του προσκηνίου που είναι μικρότερες από την διάμετρο της γειτονιάς. Παρομοίως, μία διάβρωση ακολουθούμενη από μία πανομοιότυπη διεύρυνση λέγεται ‘άνοιγμα’ (opening), και χρησιμοποιείται για την εξάλειψη απομονωμένων εικονοστοιχείων του προσκηνίου.

Ο θόρυβος στην εικόνα που προκύπτει από την αφαίρεση του φόντου, κάνει ορισμένα εικονοστοιχεία του προσκηνίου να μοιάζουν με αυτά του φόντου και αντίστροφα. Μία πράξη ‘άνοιγματος’ ακολουθούμενη από μία πράξη ‘κλεισίματος’ αντιμετωπίζει τις πηγές του λάθους : το ‘κλείσιμο’ γεμίζει τα εικονοστοιχεία του προσκηνίου που λείπουν (υποθέτοντας ότι αρκετά από τα γειτονικά εικονοστοιχεία έχουν προσδιοριστεί σωστά), και το ‘άνοιγμα’ αφαιρεί άσχετα εικονοστοιχεία του



προσκηνίου που είναι περικυκλωμένα από εικονοστοιχεία του φόντου. Ιδιαίτερη προσοχή πρέπει να δοθεί στην επιλογή της ακτίνας  $r$ . Αν η ακτίνα είναι πολύ μικρή, τότε μεγαλύτερες ομάδες εικονοστοιχείων θορύβου θα μείνουν αδιόρθωτες. Αν απ' την άλλη η ακτίνα είναι πολύ μικρή, τότε κάποια σωστά εικονοστοιχεία των αντικειμένων του προσκηνίου θα χαθούν.

Σε ιδιαίτερα θορυβώδεις εικόνες φόντου, τα λανθασμένα εικονοστοιχεία φόντου μπορεί να είναι τόσο πολλά και τόσο κοντά το ένα στο άλλο, ώστε η πράξη 'κλεισίματος' να γεμίσει τα κενά ανάμεσά τους. Η αύξηση του κατωφλίου για τον αρχικό διαχωρισμό φόντου-προσκηνίου οδηγεί στην αποφυγή αυτού του ανεπιθύμητου αποτελέσματος, διότι μεγαλώνοντας την τιμή του κατωφλίου τα εικονοστοιχεία του προσκηνίου που ταξινομούνται ως εικονοστοιχεία του φόντου είναι περισσότερα από τα εικονοστοιχεία του φόντου που τα ταξινομούνται ως εικονοστοιχεία του προσκηνίου. Έτσι, εφαρμόζοντας υπό αυτές τις συνθήκες την πράξη 'κλεισίματος' θα κλείσουν τα κενά ανάμεσα σε σωστά εκτιμημένα εικονοστοιχεία του προσκηνίου.

Ο θόρυβος στην εικόνα που προκύπτει από την αφαίρεση του φόντου, κάνει ορισμένα εικονοστοιχεία του προσκηνίου να μοιάζουν με αυτά του φόντου και αντίστροφα. Μία πράξη 'ανοίγματος' ακολουθούμενη από μία πράξη 'κλεισίματος' αντιμετωπίζει τις πηγές του λάθους : το 'κλείσιμο' γεμίζει τα εικονοστοιχεία του προσκηνίου που λείπουν (υποθέτοντας ότι αρκετά από τα γειτονικά εικονοστοιχεία έχουν προσδιοριστεί σωστά), και το 'άνοιγμα' αφαιρεί άσχετα εικονοστοιχεία του προσκηνίου που είναι περικυκλωμένα από εικονοστοιχεία του φόντου. Ιδιαίτερη προσοχή πρέπει να δοθεί στην επιλογή της ακτίνας  $r$ . Αν η ακτίνα είναι πολύ μικρή, τότε μεγαλύτερες ομάδες εικονοστοιχείων θορύβου θα μείνουν αδιόρθωτες. Αν απ' την άλλη η ακτίνα είναι πολύ μικρή, τότε κάποια σωστά εικονοστοιχεία των αντικειμένων του προσκηνίου θα χαθούν.

Σε ιδιαίτερα θορυβώδεις εικόνες φόντου, τα λανθασμένα εικονοστοιχεία φόντου μπορεί να είναι τόσο πολλά και τόσο κοντά το ένα στο άλλο, ώστε η πράξη 'κλεισίματος' να γεμίσει τα κενά ανάμεσά τους. Η αύξηση του κατωφλίου για τον αρχικό διαχωρισμό φόντου-προσκηνίου οδηγεί στην αποφυγή αυτού του ανεπιθύμητου αποτελέσματος, διότι μεγαλώνοντας την τιμή του κατωφλίου τα εικονοστοιχεία του προσκηνίου που ταξινομούνται ως εικονοστοιχεία του φόντου είναι περισσότερα από τα εικονοστοιχεία του φόντου που τα ταξινομούνται ως

εικονοστοιχεία του προσκηνίου. Έτσι, εφαρμόζοντας υπό αυτές τις συνθήκες την πράξη ‘κλεισίματος’ θα κλείσουν τα κενά ανάμεσα σε σωστά εκτιμημένα εικονοστοιχεία του προσκηνίου.

[1][2][3]

### 2.2.3 Κατάτμηση κίνησης με χρήση στατιστικών μεθόδων

Σχετικά πρόσφατα, κάποιες στατιστικές μέθοδοι για να εξαχθούν οι περιοχές αλλαγών από το φόντο έχουν εμπνευστεί από τις βασικές μεθόδους αφαίρεσης φόντου που ήδη περιγράψαμε. Οι στατιστικές αυτές προσεγγίσεις χρησιμοποιούν χαρακτηριστικά από ατομικά εικονοστοιχεία ή από ομάδες τους για να κατασκευάσουν πιο σύνθετα μοντέλα για το φόντο. Οι στατιστικές αυτών των μοντέλων μάλιστα, αλλάζουν δυναμική κατά τη διάρκεια της επεξεργασίας. Κάθε εικονοστοιχείο στην τρέχουσα εικόνα χαρακτηρίζεται ως εικονοστοιχείο πρώτου πλάνου ή φόντου αφού συγκριθεί με τις στατιστικές του ενίοτε μοντέλου φόντου. Αυτό η προσέγγιση γίνεται όλο και πιο δημοφιλής λόγω της σθεναρότητας που παρουσιάζει στο θόρυβο, στις σκιές, στις αλλαγές των συνθηκών φωτός κλπ.

Το πιο αντιπροσωπευτικό παράδειγμα της κατηγορίας ονομάζεται μίγμα γκαουσιανών (mixture of Gaussians – MOG). Στη μέθοδο αυτή, το φόντο δεν είναι ένα καρέ με τιμές, όπως είδαμε στη προηγούμενη κατηγορία, αλλά παραμετρικό. Η κάθε θέση εικονοστοιχείο αντιπροσωπεύεται από έναν αριθμό γκαουσιανών συναρτήσεων που αθροίζονται όλες μαζί δημιουργώντας μία συσσωρευτική κατανομή πιθανότητας:

$$F(i_t = \mu) = \sum_{i=1}^k \omega_{i,t} \cdot \eta(\mu, o)$$

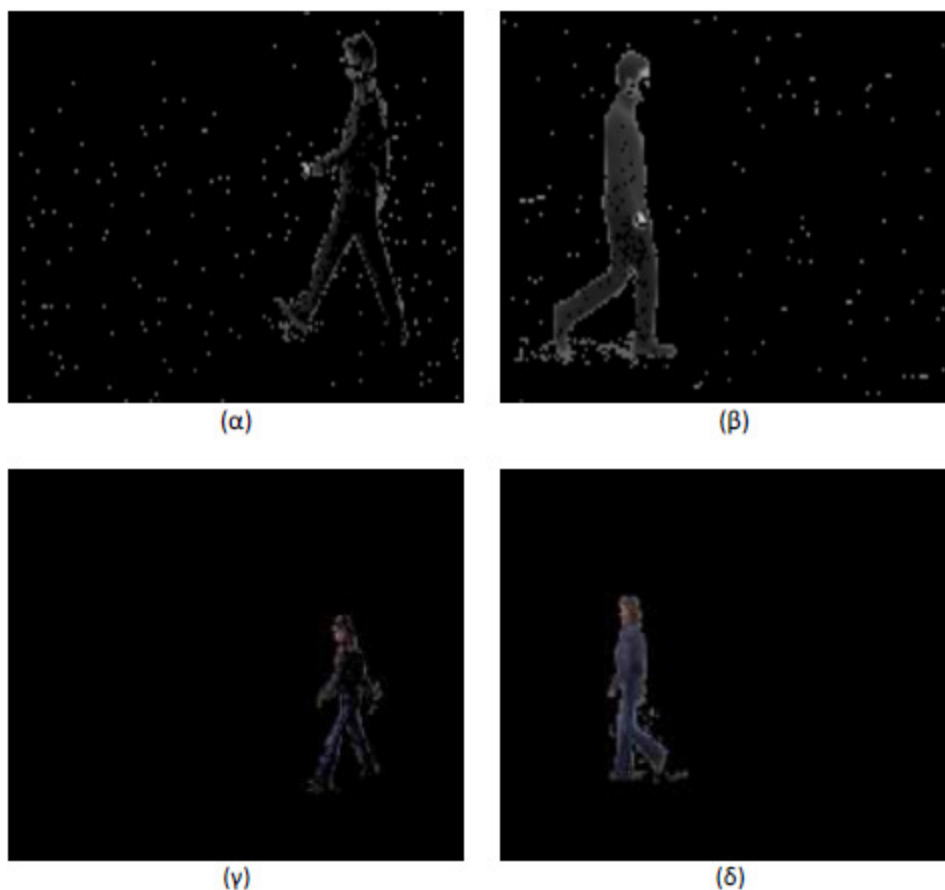
Ο μέσος όρος της κάθε γκαουσιανής συνάρτησης μπορεί να γίνει αντιληπτός ως μια βάσιμη υπόθεση (educated guess) της τιμής του εικονοστοιχείου στην επόμενη εικόνα, θεωρώντας στο σημείο αυτό ότι τα εικονοστοιχεία ανήκουν γενικά στο φόντο. Το βάρος και η τυπική απόκλιση της κάθε συνάρτησης αποτελούν μέτρα της εμπιστοσύνης μας σε αυτή την υπόθεση όπου μεγαλύτερο βάρος και μικρότερο σ

σημαίνουν μεγαλύτερη εμπιστοσύνη. Τυπικά χρησιμοποιούνται από τρεις έως πέντε γκαουσιανές κατανομές για κάθε εικονοστοιχείο, ανάλογα τους περιορισμούς μας στο μέγεθος της μνήμης του συστήματος.

Για να καθορίσουμε αν ένα εικονοστοιχείο είναι μέρος του σκηνικού το συγκρίνουμε με τις αντίστοιχες γκαουσιανές συναρτήσεις. Εάν η τιμή του βρίσκεται κοντά στη τυπική απόκλιση ενός στοιχείου του φόντου, σε επίπεδο μιας τάξης μεγέθους, θεωρείται ως μέρος του. Σε αντίθετη περίπτωση, καταχωρείται ως εικονοστοιχείο πρώτου πλάνου.

Η μέθοδος αυτή είναι πολύ ισχυρή και περιέχει πολλές παραμέτρους με την αλλαγή των οποίων μπορούμε να τον προσαρμόσουμε σε οποιοδήποτε περιβάλλον. Η πολυπλοκότητα του όμως και σε σύγκριση με τα αποτελέσματα του μας κάνει να επιλέξουμε την άλλη μέθοδο, για μείωση υπολογιστικού κόστους.

Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα:



**Εικόνα 4 - Αποτελέσματα από την χρήση της τεχνικής των στατιστικών μεθόδων για τα δύο βίντεο σε διαφορετικά χρονικά καρέ**

Τα αποτελέσματα της τεχνικής αυτής μας δείχνουν ότι πρέπει να περάσει κάποιος χρόνος ώστε ολόκληρο το αντικείμενο να περάσει στο προσκήνιο. Και στα δύο βίντεο στα αρχικά καρέ το αντικείμενο δεν παρουσιάζεται ολόκληρο ενώ όσο περνάει ο χρόνος αυτό φτιάχνει. Ο θόρυβος στο πρώτο βίντεο φαίνεται αυξημένος σε σύγκριση με τις άλλες μεθόδους, ενώ παρατηρούμε θόρυβο και στο δεύτερο βίντεο κάτι το οποίο δεν είχε παρατηρηθεί σε καμία από τις προηγούμενες μεθόδους. Λόγω των πολλών παραμέτρων της μεθόδου, έχουμε πολλές εναλλακτικές για να προσαρμόσουμε τον αλγόριθμο σε μια σκηνή και να πάρουμε πιο καλές εικόνες από τις παραπάνω. Παρόλα αυτά, ο χαρακτηριστικά μεγάλος βαθμός πολυπλοκότητας και η ανάγκη επαναπροσδιορισμού των παραμέτρων σε κάθε αλλαγή του σκηνικού ενδιαφέροντος, καθιστούν τη τεχνική αυτή ακατάλληλη για την εφαρμογή που θέλουμε να σχεδιάσουμε και γι αυτό και δεν θα εμβαθύνουμε περαιτέρω. [4]

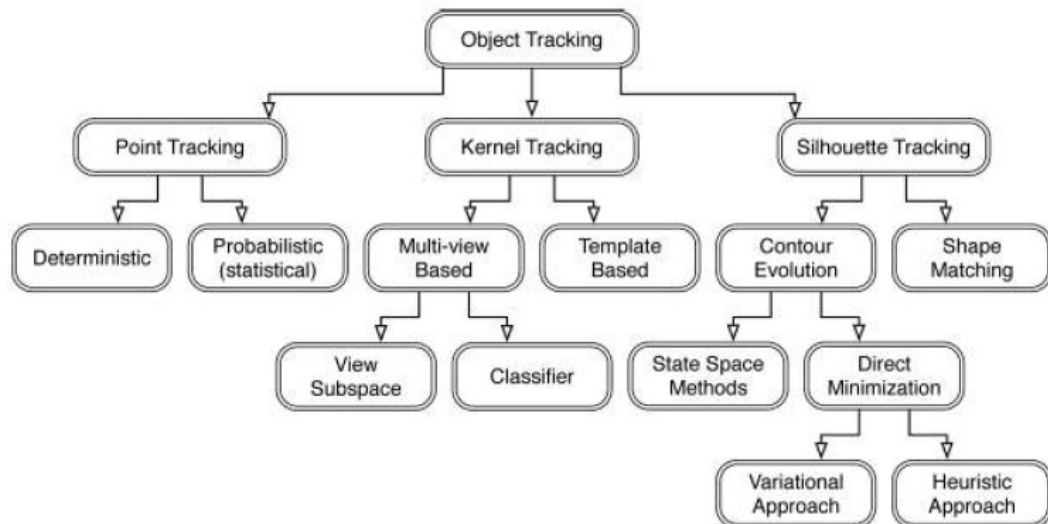
## **Κεφάλαιο 3 Εντοπισμός κινούμενης Οντότητας**

### **3.1 Εισαγωγή**

Μετά την διαδικασία της κατάτμησης η οποία είχε σαν αποτέλεσμα τον εντοπισμό και την αναγνώριση του αντικειμένου, το οποίο βρίσκεται υπό μελέτη, το επόμενο βήμα είναι η παρακολούθηση του αντικειμένου καθ' όλη την διάρκεια της κίνησης του. Δεν θα είχε νόημα απλώς μια αναγνώριση του αντικειμένου ενδιαφέροντος αν αυτή δεν συνδυαζόταν και με μία παρακολούθηση της τροχιάς της κίνησης του, αφού αυτό που πραγματικά μας ενδιαφέρει δεν είναι απλώς ο εντοπισμός του αντικειμένου αλλά περισσότερο η συμπεριφορά του στο χώρο δραστηριοποίησης του και η οποία προφανώς είναι απόρροια της κίνησής του.

Στόχος της διαδικασίας της παρακολούθησης αντικειμένου είναι η καταγραφή της τροχιάς του καθώς αυτό αλλάζει θέση σε κάθε καρέ της ακολουθίας βίντεο. Τυπικά, η παρακολούθηση αυτή περιλαμβάνει ταίριασμα των σημείων του αντικειμένου σε διαδοχικά καρέ χρησιμοποιώντας γνωρίσματα όπως σημεία, γραμμές, ή οντότητες. Πιο συγκεκριμένα, έχει να κάνει με την εύρεση συναφών σχέσεων μεταξύ χαρακτηριστικών της εικόνας σε διαδοχικά καρέ, όπως για παράδειγμα το χρώμα, η υφή, η ταχύτητα και η θέση. Η παρακολούθηση τροχιάς των αντικειμένων είναι σαφώς ένα ιδιαίτερα σημαντικό και βασικό κομμάτι στην ανάλυση ανθρώπινης κίνησης.

Οι τεχνικές που χρησιμοποιούνται για την παρακολούθηση της κίνησης ποικίλουν ανάλογα την εφαρμογή και η χρήση τους εξαρτάται από πολλούς παράγοντες όπως το μέγεθος του αντικειμένου, το περιβάλλον στο οποίο δραστηριοποιείτε, τις απαιτήσεις στην ποιότητα της παρακολούθησης κ.α. . Στις επόμενες παραγράφους θα αναφέρουμε κάποιες από αυτές τις τεχνικές οι οποίες μπορούν να φανούν συγκεντρωμένες και χωρισμένες στις διάφορες κατηγορίες, στην παρακάτω εικόνα.



Εικόνα 5 - Τεχνικές παρατήρησης κίνησης αντικειμένου

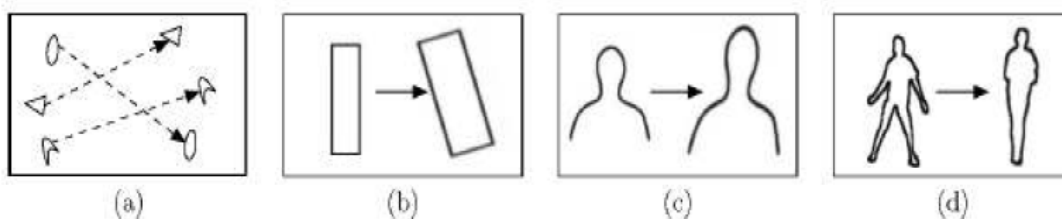
Όπως εύκολα μπορεί να φανεί από το παραπάνω σχήμα τρεις είναι οι βασικές κατηγορίες επίλυσης του προβλήματος, κάθε μία εκ των οποίων, μπορεί να προσεγγιστεί από διαφορετική μέθοδο. Οι κατηγορίες αυτές είναι:

- Point tracking. Σε αυτή την κατηγορία το αντικείμενο που είναι προς παρακολούθηση αντιπροσωπεύεται και αναπαριστάτε από ένα σημείο ή ένα πλήθος σημείων. Η σύνδεση των σημείων αυτών βασίζεται στην προηγούμενη κατάσταση του αντικειμένου η οποία περιέχει την θέση του και την κίνησή του. Ένα παράδειγμα εφαρμογής αυτής της μεθόδου φαίνεται στην παρακάτω εικόνα (a)

- Kernel tracking. Οι τεχνικές Kernel βασίζονται κυρίως στην εμφάνιση και στο σχήμα του αντικειμένου. Το αντικείμενο που εντοπίζεται εισάγεται εντός περιθωρίου το σχήμα του οποίου είναι κάποιο ορθογώνιο ή έλλειψη. Το αντικείμενο παρακολουθείται υπολογίζοντας την κίνηση του περιθωρίου σε κάθε ένα από τα συνεχόμενα καρέ. Αυτή η κίνηση είναι συνήθως σε μορφή κάποιου παραμετρικού μετασχηματισμού όπως π.χ. περιστροφή. Στην παρακάτω εικόνα (b) φαίνεται ένα τέτοιο παράδειγμα.

- Silhouette tracking. Στην κατηγορία αυτή η παρακολούθηση γίνεται υπολογίζοντας την περιοχή που καταλαμβάνει το αντικείμενο σε κάθε ένα καρέ της ακολουθίας βίντεο. Η πληροφορία που χρησιμοποιείται βρίσκεται εντός του αντικειμένου και αντιπροσωπεύει την πυκνότητα ή το σχήμα του αντικειμένου και τα οποία είναι

χαρακτηριστικά που συνήθως χρειάζονται προκειμένου να εξαχθεί ο χάρτης ακμών .  
Παραδείγματα φαίνονται στην παρακάτω εικόνα (c)-(d).

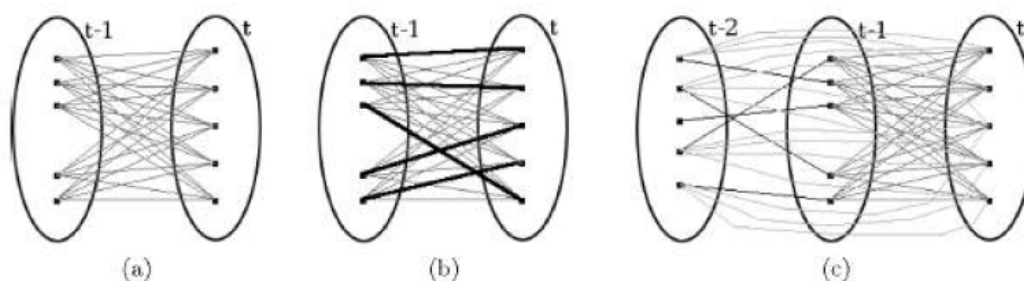


Εικόνα 6 - Παραδείγματα εφαρμογής διαφορετικών μεθόδων για την παρακολούθηση κινούμενου αντικειμένου

[5]

### 3.2 Point tracking - Deterministic method

Η Ντετερμινιστική μέθοδος για την παρακολούθηση αντικειμένου κάνει χρήση εξισώσεων κόστους στις οποίες γίνεται προσπάθεια ελαχιστοποίησής τους καθώς γίνεται σύγκριση του αντικειμένου που παρατηρείται στο καρέ  $t-1$  με το αντικείμενο που παρατηρείται στο καρέ  $t$ . Λύση σε αυτό το πρόβλημα πετυχαίνεται με την ακριβή αντιστοίχιση ένα προς ένα σημείων όπως αυτή φαίνεται στην παρακάτω εικόνα (b)



Εικόνα 7 - Αντιστοίχιση σημείων δύο διαφορετικών καρέ

Η συνάρτηση του κόστους διασύνδεσης συνήθως ορίζεται κάνοντας χρήση συνδυασμού των παρακάτω παραδοχών.

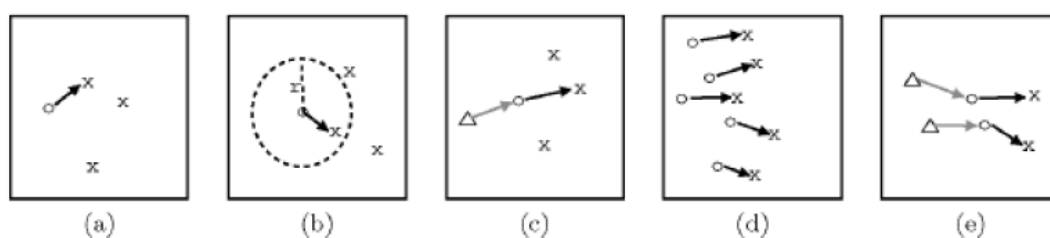
- Εγγύτητα. Ένα αντικείμενο μπορούμε να δεχτούμε ότι σε διαφορά ενός καρέ δεν μπορεί να έχει παραμορφωθεί σε τέτοιο βαθμό ώστε η απόσταση μεταξύ των σημείων του να έχει μεταβληθεί αρκετά. Αυτό σημαίνει ότι γειτονικά σημεία θα πρέπει να παραμένουν γειτονικά με την ίδια απόσταση μεταξύ τους. Αυτή η παραδοχή φαίνεται στην παρακάτω εικόνα (a)

- Μέγιστη ταχύτητα. Η ταχύτητα ενός αντικειμένου είναι περιορισμένη επομένως και η θέση στην οποία θα βρίσκονται τα σημεία του σε επόμενο καρέ θα πρέπει να είναι περιορισμένη και σίγουρα εντός μιας συγκεκριμένης ακτίνας. Αυτό μπορεί να φανεί στην παρακάτω εικόνα (b)

- Μεταβολή κίνησης. Αυτό δηλώνει ότι δεν μπορεί να υπάρξει δραματική μεταβολή στα χαρακτηριστικά της κίνησης, δηλαδή τόσο στην τιμή της ταχύτητάς του αντικειμένου όσο και στην κατεύθυνση της κίνησης του. Χαρακτηριστικό αυτής της παραδοχής φαίνεται στην παρακάτω εικόνα (c)

- Ομοιότητα κίνησης. Σύμφωνα με αυτή την παραδοχή δεχόμαστε ότι γειτονικά σημεία δεν μπορεί να κινούνται διαφορετικά, η κίνηση τους πρέπει να είναι όμοια όπως φαίνεται στην παρακάτω εικόνα(d)

- Ακαμψία. Δεχόμαστε ότι τα αντικείμενα στον τρισδιάστατο κόσμο είναι άκαμπτα επομένως η απόσταση σημείων του αντικειμένου σε διαφορετικά καρέ θα πρέπει να παραμένει η ίδια, κάτι το οποίο φαίνεται στην παρακάτω εικόνα (e)



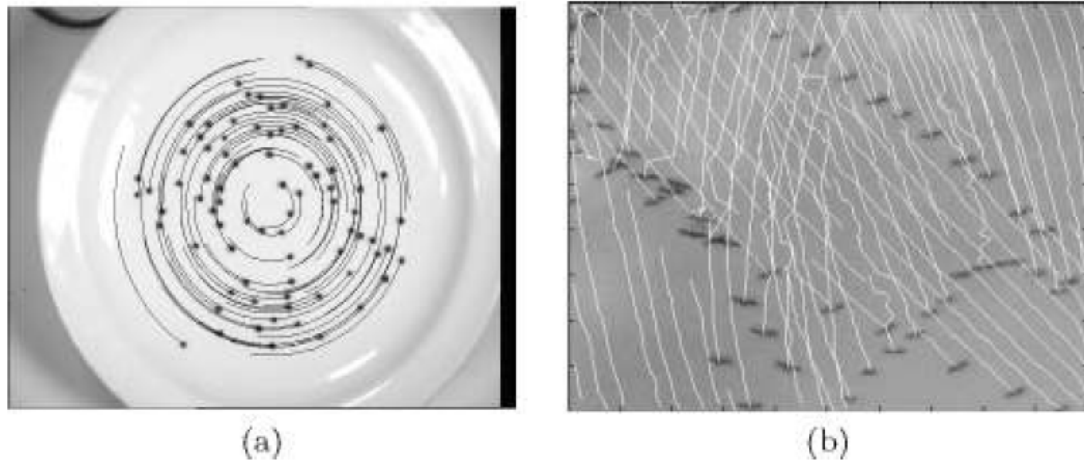
Εικόνα 8 - Επεξηγηματική αναπαράσταση παραδοχών

Από τα παραπάνω αντιλαμβανόμαστε ότι κάνοντας χρήση αυτής της μεθόδου ουσιαστικά εγκαταλείπουμε την ιδέα της παρακολούθησης ολόκληρου του αντικειμένου και επικεντρώνουμε την προσοχή μας σε ορισμένα σημεία αυτού. Τα πλεονεκτήματα της είναι ότι μειώνει σημαντικά την υπολογιστική πολυπλοκότητα, ενώ παράλληλα είναι σθεναρή σε μερική επικάλυψη των αντικειμένων με τη προϋπόθεση ότι κάποια από τα γνωρίσματα παραμένουν ορατά.

Η μέθοδος αυτή χωρίζεται σε δύο μέρη, στην εξαγωγή των γνωρισμάτων και στην αντιστοίχιση τους. Χαμηλότερου επιπέδου χαρακτηριστικά, όπως σημεία, είναι ευκολότερο να εξαχθούν αλλά πολύ πιο δύσκολο να παρακολουθηθούν αφού είναι πιο δύσκολο να προσδιοριστεί μια ένα-προς-ένα αντιστοιχία μεταξύ τους. Αντίστροφα, υψηλότερου επιπέδου χαρακτηριστικά, όπως γραμμές ή οντότητες, είναι ευκολότερο να εντοπιστούν αλλά πιο δύσκολο να εξαχθούν. Φαίνεται λοιπόν πως



υπάρχει μια αλληλεξάρτηση ανάμεσα στη πολυπλοκότητα των χαρακτηριστικών και την αποδοτικότητα της παρακολούθησης. Στα παρακάτω παραδείγματα φαίνονται τα αποτελέσματα κάνοντας χρήση αυτής της μεθόδου.



Εικόνα 9 - Παραδείγματα εφαρμογών με χρήση του *deterministic method*

[5]

### **3.3 Point tracking - Statistical method**

Χαρακτηριστικό παράδειγμα αυτής της κατηγορίας είναι τα φίλτρα Kalman τα οποία συνήθως χρησιμοποιούνται ώστε να υπολογιστεί η κατάσταση του γραμμικού συστήματος υποθέτοντας ότι η κατάσταση αυτή έχει κανονική κατανομή κατά Gauss. Τα φίλτρα Kalman είναι μια τεχνική στοχαστικής πρόγνωσης και εκτίμησης των σωστών παραμέτρων ενός μοντέλου σε ένα δυναμικό περιβάλλον. Σε αντίθεση με την ντετερμινιστική τεχνική, όπου οι παράμετροι του συστήματος θεωρούνται

άγνωστοι μεν αλλά σταθεροί, στα φίλτρα Kalman οι παράμετροι θεωρείται πως εξαρτώνται κάθε στιγμή από την κατάσταση του συστήματος και επομένως αλλάζουν δυναμικά με κάθε μεταβολή της κατάστασής του. Έτσι, η τεχνική αυτή επανεκτιμά με κάθε νέα μέτρηση την κατάσταση του δυναμικού συστήματος, βασιζόμενη σε παρατηρήσεις που έχουν γίνει στις προηγούμενες χρονικές στιγμές και επαναπροσδιορίζει τις παραμέτρους του συστήματος.

Τα φίλτρα Kalman χρησιμοποιούνται σε εφαρμογές όπου απαιτείται η χρήση ενός δυναμικού μοντέλου του οποίου οι βέλτιστοι παράμετροι πρέπει να προσδιορίζονται σε διακριτά χρονικά διαστήματα. Η υλοποίησή τους πραγματοποιείται σε δύο στάδια

, της πρόβλεψης και της διόρθωσης. Το βήμα της πρόβλεψης χρησιμοποιεί μία αρχική κατάσταση ώστε να γίνει πρόβλεψη μιας νέας κατάστασης των μεταβλητών.

[5]

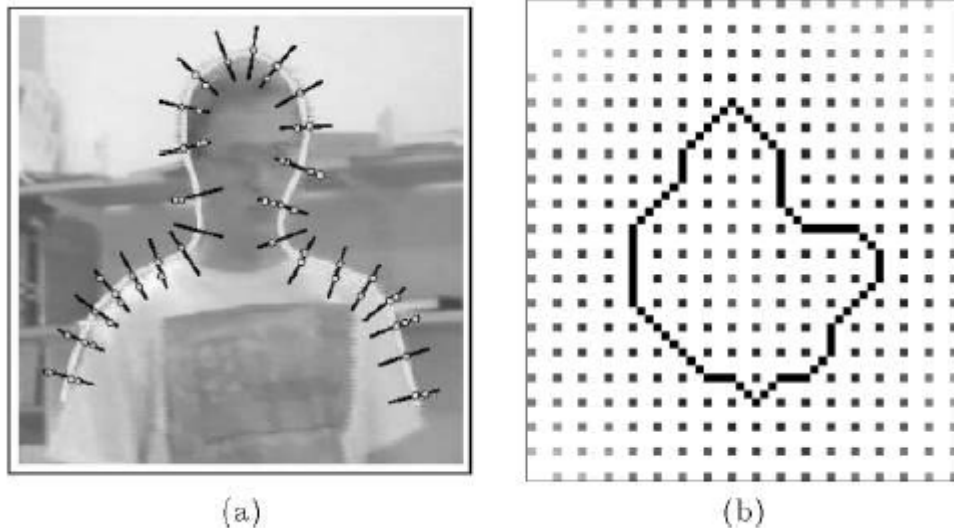
### ***3.4 Kernel Tracking***

Η υπολογιστική πολυπλοκότητα ενός παρακολουθητή είναι ένα πολύ σημαντικό μέγεθος για τις περισσότερες εφαρμογές μηχανικής όρασης καθώς μόνο ένα μικρό ποσοστό των πόρων του συστήματος αφιερώνεται στην παρακολούθηση. Το υπόλοιπο, προορίζεται συνήθως για τα στάδια προεπεξεργασίας και για εργασίες υψηλότερου επιπέδου, όπως αναγνώριση και ερμηνεία τροχιάς. Στη συνέχεια θα παρουσιάσουμε μια προσέγγιση για παρακολούθηση αντικειμένων σε πραγματικό χρόνο που βασίζεται σε οπτικά χαρακτηριστικά, όπως το χρώμα και η υφή, των οποίων η στατιστική κατανομή χαρακτηρίζει το αντικείμενο ενδιαφέροντος. Η προτεινόμενη παρακολούθηση είναι κατάλληλη για πολλές κατηγορίες αντικειμένων με διαφορετικά μοτίβα χρώματος ή/και υφής ενώ παράλληλα είναι σθεναρό σε μερικές επικαλύψεις και κινήσεις της κάμερας. Οι επαναλήψεις ολίσθησης μέσου όρου χρησιμοποιούνται ώστε να βρεθεί η υποψήφια περιοχή που μοιάζει περισσότερο με το δοσμένο μοντέλο. Η ομοιότητα αυτό εκφράζεται από ένα μετρικό που βασίζεται στο συντελεστή Bhattacharyya. [6]

### ***3.5 Silhouette Tracking***

Τα αντικείμενα μπορεί να έχουν πολύπλοκα σχήματα, για παράδειγμα τα χέρια , το κεφάλι και οι ώμοι δεν μπορούν να περιγραφούν καλά από απλά γεωμετρικά σχήματα, κάτι το οποίο φαίνεται από στην εικόνα 10. Γενικά το σχήμα είναι από τους πλέον σημαντικούς παραμέτρους για την αναγνώριση των αντικειμένων. Ο όρος “σχήμα” μπορεί να αναφέρεται είτε στο εξωτερικό δισδιάστατο (ή τρισδιάστατο) περίγραμμα του αντικειμένου, ή στη γενική του όψη και διάταξη. Γεωμετρικά, δύο αντικείμενα έχουν το ίδιο σχήμα αν το ένα μπορεί να μετατραπεί στο άλλο με χρήση κάποιου συνδυασμού από μετατοπίσεις, περιστροφές και ομοιόμορφες αλλαγές κλίμακας, με άλλα λόγια το σχήμα είναι όλη εκείνη η γεωμετρική πληροφορία για το αντικείμενο που παραμένει αμετάβλητη από την θέση ,την κλίμακα και στην

περιστροφή του αντικειμένου. Επιπλέον οι άνθρωποι αναγνωρίζουν δύο ίδια σχήματα ακόμα και αν το ένα αποτελεί κατοπτρισμού του άλλου.



Εικόνα 10 -Απολογισμός μεθόδου silhouette tracking

Όσο αναφορά την όραση των υπολογιστών το πρόβλημα μπορεί να λυθεί είτε υπολογίζοντας το περίγραμμα των αντικειμένων (boundary based) είτε μελετώντας την περιοχή που περικλείεται εντός του περιγράμματος (region based).

[7]

### 3.6 Φίλτρο Kalman

#### 3.6.1 Γενικά για το Kalman

Το φίλτρο Kalman [8] είναι ένα ιδιαίτερα διαδεδομένο και εύχρηστο φίλτρο που χρησιμοποιείται για απαλλαγή από τις θορυβώδεις μετρήσεις. Ο αλγόριθμος αυτός, πραγματοποιεί πρόβλεψη της θέσης του στόχου και χρησιμοποιεί τις μετρήσεις για να διορθώσει την πρόβλεψη αυτή. Επομένως έχει την δυνατότητα να παρέχει εκτιμήσεις με σφάλμα μικρότερο του σφάλματος μέτρησης. Αποτελεί μια επέκταση της μεθόδου ελαχίστων τετραγώνων και πρόκειται ουσιαστικά για ένα σύνολο μαθηματικών εξισώσεων, οι οποίες εισάγουν έναν εκτιμητή πρόβλεψης – διόρθωσης, ο οποίος είναι βέλτιστος από την άποψη, ότι ελαχιστοποιεί την συνδιακύμανση σφάλματος

εκτίμησης, όταν η δυναμική κατάσταση του στόχου και ο θόρυβος μέτρησης έχουν μοντελοποιηθεί με ακρίβεια. Σε γραμμικά περιβάλλοντα ο αλγόριθμος του Kalman αποτελεί τη βέλτιστη λύση.

Η εκτεταμένη χρήση του φίλτρου Kalman οφείλεται στο ότι συγκεντρώνει πολλά πλεονεκτήματα, τα οποία βοηθούν στην καλύτερη εκτίμηση της δυναμικής κατάστασης του συστήματος. Κατά πρώτο λόγο, το κέρδος του φίλτρου ανανεώνεται με κάθε νέα μέτρηση. Το γεγονός αυτό διευκολύνει την ανίχνευση σε συστήματα με χρονομεταβλητές δυναμικές καταστάσεις ή μετρήσεις, οι διαδικασίες των οποίων μεταβάλλονται με τον χρόνο. Επιπλέον, η συνδιακύμανση του θορύβου μέτρησης μπορεί να αυξομειωθεί, έτσι ώστε η κάθε μέτρηση να αποκτήσει διαφορετικό βάρος ανάλογα με την απόσταση από τον στόχο, την ισχύ του σήματος ή οποιαδήποτε άλλη ενδεχόμενη ένδειξη της ποιότητας της μέτρησης. Αυτό είναι ιδιαίτερα σημαντικό για περιπτώσεις όπου το διάνυμα μετρήσεων περιέχει μετρήσεις από διαφορετικού τύπου αισθητήρες γιατί δίνεται η δυνατότητα στον χρήστη να κάνει μία αρχικοποίηση ανάλογα με την κατά περίπτωση αναμενόμενη αξιοπιστία του εκάστοτε αισθητήρα. Άλλο πλεονέκτημα του φίλτρου είναι η αποδοτικότητά του σε υπολογιστικά συστήματα περιορισμένης μνήμης. Όλη η πληροφορία για τις αρχικές καταστάσεις, τις μετρήσεις και την κίνηση περιέχεται στον πίνακα συνδιακύμανσης σφάλματος  $P_k$ . Σημαντικό είναι επίσης το ότι δεν απαιτεί περιοδικές ενημερώσεις χρόνου ή μετρήσεων. Στην γενική περίπτωση, το φίλτρο ενημερώνει τον χρόνο με υψηλό ρυθμό και όταν κάποια μέτρηση παρέχεται από κάποιον αισθητήρα, η ενημέρωση χρόνου διακόπτεται προσωρινά, προκειμένου να πραγματοποιηθεί η ενημέρωση της μέτρησης. Επομένως έχει την δυνατότητα να διαχειρίζεται δεδομένα από ανόμοιους αισθητήρες, οι οποίοι δεν είναι καν συγχρονισμένοι.

### 3.6.2 Μαθηματική Επεξήγηση

Για την εκτίμηση ενός δυναμικού συστήματος γενικά χρησιμοποιούνται 2 μοντέλα :

- 1) ένα μοντέλο που περιγράφει την εξέλιξη της κατάστασης στο χρόνο (κινηματικό μοντέλο) και
- 2) ένα μοντέλο που σχετίζεται με τις θορυβώδεις μετρήσεις (μοντέλο μέτρησης)

Έστω  $x \in \mathcal{R}^n$  το διάνυσμα κατάστασης ενός στόχου, του οποίου η δυναμική διαδικασία περιγράφεται από την γραμμική εξίσωση διαφορών :

$$x_k = Ax_{k-1} + Du_k + w_{k-1} \quad (3.1)$$

όπου

- Ο  $(n \times n)$  πίνακας  $A$  αποτελεί τον πίνακα μετάβασης της κατάστασης από την  $k-1$  σάρωση στην  $k$ , όπου το χρονικό διάστημα σάρωσης είναι  $T$ .
- Ο  $(n \times l)$  πίνακας  $D$  συσχετίζει την κατάσταση με το διάνυσμα εισόδου  $u \in \mathcal{R}^l$ .
- Η τυχαία μεταβλητή  $w$  μοντελοποιεί τον λευκό, γκαουσιανό θόρυβο διαδικασίας μηδενικής μέσης τιμής και με συνδιακύμανση που δίνεται από τον πίνακα  $(n \times n)$   $Q$ . Ο θόρυβος διαδικασίας αναπαριστά τις τυχαίες επιδράσεις, τις οποίες υφίσταται το σύστημα και την ατελή μοντελοποίηση της δυναμικής κατάστασης του στόχου.

Από τη (3.1) είναι σαφές ότι η στατιστική αναπαράσταση της διαδικασίας την παρούσα στιγμή ( $k$ -οστή σάρωση) καθορίζεται αποκλειστικά από την προηγούμενη (σάρωση  $k-1$ ). Η δυναμική κατάσταση του στόχου οδηγείται από την ντετερμινιστική είσοδο και τον τυχαίο θόρυβο διαδικασίας. Η ντετερμινιστική είσοδος του συστήματος είναι δυνατόν να περιέχει κάποιες γνωστές μεταβλητές εισόδου, οι οποίες δεν περιλαμβάνονται στις καταστάσεις του συστήματος.

Έστω ακόμα η μέτρηση  $z \in \mathcal{R}^m$  περιγράφεται από το μοντέλο:

$$z_k = Hx_k + v_k \quad (3.2)$$

όπου

- Ο  $(m \times n)$  πίνακας  $H$  συνδέει την κατάσταση με το διάνυσμα μετρήσεων.
- Αντίστοιχα με το  $w$  η τυχαία μεταβλητή  $v$  παριστάνει τον θόρυβο μέτρησης μηδενικής μέσης τιμής και συνδιακύμανσης  $R$  ( $m \times n$ ).

Οι πίνακες  $A$ ,  $H$ ,  $Q$  και  $R$  είναι γενικά χρονικά μεταβαλλόμενοι, αλλά εφεξής θεωρούνται σταθεροί για λόγους απλότητας

Έστω τώρα  $\hat{x}_k^- \in \mathfrak{R}^n$  το *a priori* εκτιμώμενο διάνυσμα κατάστασης της σάρωσης  $k$ , γνωρίζοντας την διαδικασία μέχρι την σάρωση  $k$  και  $\hat{x}_k \in \mathfrak{R}^n$  το *a posteriori* διάνυσμα εκτίμησης στο βήμα  $k$ , γνωρίζοντας την μέτρηση  $z_k$ . Στην συνέχεια ορίζονται αντίστοιχα το *a priori* και *a posteriori* σφάλμα εκτίμησης :

$$e_k^- \equiv x_k - \hat{x}_k^- \quad (3.3)$$

$$e_k \equiv x_k - \hat{x}_k \quad (3.4)$$

Επίσης αντίστοιχα ορίζονται και οι *a priori* και *a posteriori* πίνακες συνδιακύμανσης σφάλματος εκτίμησης :

$$P_k^- = E[e_k^- e_k^{-T}] \quad (3.5)$$

$$P_k = E[e_k e_k^T] \quad (3.6)$$

Προκειμένου να παραχθούν οι εξισώσεις του φίλτρου χρειάζεται μια σχέση, η οποία να υπολογίζει την *a posteriori* εκτίμηση της κατάστασης ως γραμμικό συνδυασμό της *a priori* εκτίμησης και μιας διαφοράς της πραγματικής μέτρησης και της πρόβλεψης μέτρησης  $H \hat{x}_k^-$  :

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H \hat{x}_k^-) \quad (3.7)$$

Η διαφορά  $(z_k - H\hat{x}_k^-)$  συναντάται στη βιβλιογραφία ως innovation ή residual vector και αναπαριστά την διαφορά ανάμεσα στην πρόβλεψη μέτρησης και την πραγματική μέτρηση και μηδενική τιμή σημαίνει, ότι η πρόβλεψη συμπίπτει με την πραγματική μέτρηση.

Ο  $(n \times m)$  πίνακας  $K$  είναι το κέρδος (gain factor) του φίλτρου.

Το κέρδος  $K$  ελαχιστοποιεί τον *a posteriori* πίνακα συνδιακύμανσης σφάλματος εκτίμησης  $P_k = E[e_k e_k^T]$ . Η ελαχιστοποίηση επιτυγχάνεται ως εξής :

Με αντικατάσταση της (3.7) στην (3.4) :

$$e_k = x_k - \hat{x}_k^- (I + KH) - Kz_k \quad (3.8)$$

Στην συνέχεια η (3.8) αντικαθίσταται στην (3.6) και το ίχνος του αποτελέσματος ως προς  $K$  τίθεται ίσο προς μηδέν.

Το  $K_k$ , το οποίο ελαχιστοποιεί την (3.6) μπορεί να γραφεί και στην μορφή :

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{HP_k^- H^T + R} \quad (3.9)$$

Παρατηρούμε ότι καθώς ο πίνακας συνδιακύμανσης σφάλματος μέτρησης  $R$  τείνει στο μηδέν, το κέρδος  $K$  καθορίζει το υπόλοιπο περισσότερο :

$$\lim_{R_k \rightarrow 0} K_k = H^{-1}$$

Παράλληλα, καθώς ο *a priori* πίνακας συνδιακύμανσης σφάλματος εκτίμησης  $P_k$  τείνει στο μηδέν, το κέρδος καθορίζει το υπόλοιπο λιγότερο :

$$\lim_{P_k \rightarrow 0} K_k = 0$$

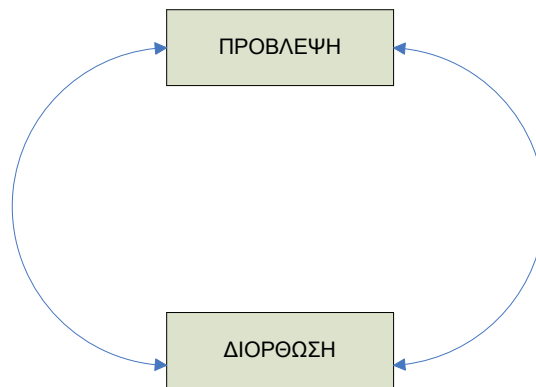
Με άλλα λόγια, μπορεί να ειπωθεί ότι καθώς το σφάλμα μέτρησης τείνει στο μηδέν, η πραγματική μέτρηση γίνεται πιο αξιόπιστη, ενώ η πρόβλεψη μέτρησης γίνεται λιγότερο αξιόπιστη. Αντίστοιχα, καθώς ο *a priori* πίνακας συνδιακύμανσης σφάλματος εκτίμησης τείνει στο μηδέν, η πραγματική μέτρηση γίνεται λιγότερο αξιόπιστη και η πρόβλεψη μέτρησης περισσότερο.

### 3.6.3 Αλγόριθμος

Το Φίλτρο Kalman αποτελεί έναν αναδρομικό αλγόριθμο εκτίμησης της κατάστασης ενός συστήματος (*recursive state-estimation*) [9]. Ο αλγόριθμος του φίλτρου πραγματοποιεί την εκτίμηση, χρησιμοποιώντας μια μορφή ελέγχου ανάδρασης (*feedback control*). Αρχικά το φίλτρο εκτιμά την κατάσταση του συστήματος για κάποια στιγμή και στην συνέχεια αποκτά ανάδραση από τις μετρήσεις. Επομένως οι εξισώσεις του φίλτρου Kalman διακρίνονται σε δύο ομάδες, στις εξισώσεις ενημέρωσης χρόνου (*time update*) και στις εξισώσεις ενημέρωσης μέτρησης (*measurement update*). Οι εξισώσεις της πρώτης ομάδας είναι υπεύθυνες για την χρονική προώθηση της εκτίμησης της παρούσας κατάστασης και την *a priori* εκτίμηση της συνδιακύμανσης σφάλματος για το επόμενο βήμα. Οι εξισώσεις της άλλης ομάδας είναι υπεύθυνες για την ανάδραση, όπως για παράδειγμα την εισαγωγή νέων μετρήσεων στην *a priori* εκτίμηση, ώστε να παραχθεί βελτιωμένη *a posteriori* εκτίμηση.

Συνοπτικά μπορεί να ειπωθεί ότι οι εξισώσεις ενημέρωσης χρόνου είναι οι εξισώσεις πρόβλεψης (*predictor equations*), ενώ οι εξισώσεις ενημέρωσης μέτρησης είναι οι εξισώσεις διόρθωσης (*corrector equations*). Επομένως το φίλτρο Kalman μπορεί να θεωρηθεί ως ένας αλγόριθμος πρόβλεψης – διόρθωσης.





Εικόνα 11 - Αλγόριθμος Kalman

Οι εξισώσεις ενημέρωσης χρόνου του φίλτρου είναι οι εξής :

$$\hat{x}_k^- = A\hat{x}_{k-1} + Du_k \quad (3.10)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3.11)$$

και οι εξισώσεις ενημέρωσης μέτρησης είναι :

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.12)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (3.13)$$

$$P_k = (I - K_k H)P_k^- \quad (3.14)$$

Αρχικά υπολογίζεται η *a priori* εκτίμηση της κατάστασης, με βάση την *a posteriori* εκτίμηση της προηγούμενης σάρωσης, όπως φαίνεται στην σχέση (3.10).

Αντίστοιχα υπολογίζεται και η *a priori* εκτίμηση του πίνακα συνδιακύμανσης σφάλματος από την *a posteriori* εκτίμηση του προηγούμενου βήματος (σχέση 3.11). Αυτό είναι το στάδιο της πρόβλεψης.

Στη συνέχεια ακολουθεί η φάση της διόρθωσης των μεγεθών αυτών, με χρήση των νέων μετρήσεων. Υπολογίζεται η νέα τιμή του κέρδους  $K$  και πραγματοποιείται η *a posteriori* εκτίμηση της κατάστασης με βάση την προβλεπόμενη τιμή, το κέρδος και το υπόλοιπο μέτρησης (σχέση 3.13).

Τέλος λαμβάνει χώρα η εκτίμηση της *a posteriori* τιμής του πίνακα συνδιακύμανσης σφάλματος (σχέση 3.14).

Οι *a posteriori* τιμές χρησιμοποιούνται στο επόμενο βήμα εκ νέου για ενημέρωση των προβλέψεων.

## Κεφάλαιο 4 Υλοποίηση

Σε αυτό το σημείο θα περάσουμε στο πρακτικό μέρος της πτυχιακής παρουσιάζοντας το εργαλείο με το οποίο θα υλοποιήσουμε την πτυχιακή μας. Στη συνέχεια θα δούμε ένα διάγραμμα, σύμφωνα με το οποίο θα πραγματοποιήσουμε το πρόγραμμα, το οποίο θα ανιχνεύει ανθρώπινη κίνηση στο χώρο.

### 4.1 Εισαγωγή στο Matlab

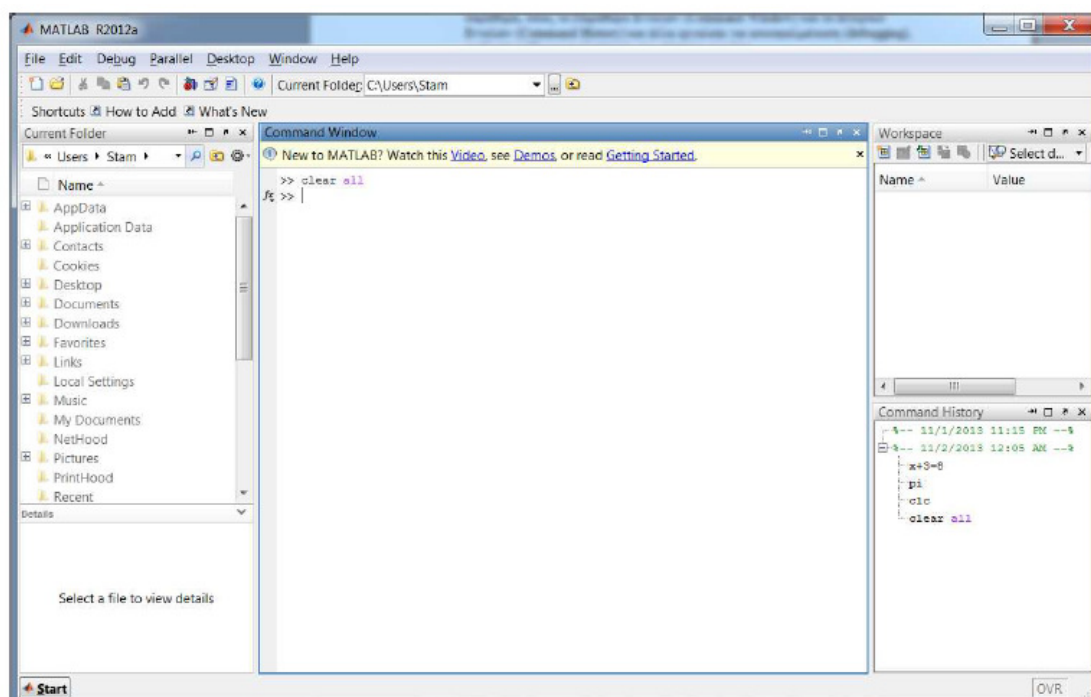
Το Matlab είναι ένα εργαλείο το οποίο προσφέρει ένα διαδραστικό προγραμματιστικό περιβάλλον στον χρήστη και χρησιμοποιείται σε ένα μεγάλο αριθμό εφαρμογών. Στους χρήστες του Matlab προσφέρεται μια υκαι επίλυση σύνθετων μαθηματικών, και όχι μόνο, προβλημάτων.

Το όνομά του προέρχεται από τις λέξεις Matrix Laboratory (εργαστήριο πινάκων) επειδή για τη λειτουργία του χρησιμοποιεί εξολοκλήρου πίνακες. Το Matlab εκτός των άλλων, επιτρέπει τον εύκολο χειρισμό πινάκων, τη γραφική απεικόνιση (plotting) συναρτήσεων και δεδομένων, την υλοποίηση αλγορίθμων και τη συνεργασία και διαλειτουργικότητα με προγράμματα γραμμένα σε άλλες γλώσσες προγραμματισμού. Επιπροσθέτως επειδή το Matlab μπορεί να χρησιμοποιηθεί σε πολλά επιστημονικά πεδία(επεξεργασία σήματος, νευρωνικά δίκτυα, συστήματα ελέγχου κ.τ.λ.), είναι δυνατή η ενσωμάτωση πρόσθετων πακέτων, που ονομάζονται toolboxes, και δίνουν τη δυνατότητα χρήσης εξεζητημένων συναρτήσεων. Τα βασικά συστατικά του Matlab είναι:

- το Περιβάλλον Ανάπτυξης και τα αντίστοιχα εργαλεία του: περιλαμβάνει διάφορα παράθυρα, όπως το Παράθυρο Εντολών (Command Window) και το Ιστορικό Εντολών (Command History) και άλλα εργαλεία για αποσφαλμάτωση (debugging), ανάλυση κώδικα και πλοήγηση στο σύστημα αρχείων
- η βιβλιοθήκη μαθηματικών συναρτήσεων: ένα από τα πιο σημαντικά συστατικά του Matlab, το οποίο προσφέρει μεγάλο εύρος αριθμητικών συναρτήσεων, από τις πιο απλές μέχρι τις πιο περίπλοκες.
- η γλώσσα προγραμματισμού: μια υψηλού επιπέδου προγραμματιστική γλώσσα με δομές δεδομένων, συναρτήσεις, εντολές ελέγχου ροής, εντολές εισόδου/εξόδου και στοιχεία από αντικειμενοστραφείς γλώσσες προγραμματισμού

- τα γραφικά συστατικά: το Matlab παρέχει μια πληθώρα δυνατοτήτων απεικόνισης διανυσμάτων, πινάκων και γραφημάτων στις 2 και 3 διαστάσεις.

Στο σημείο αυτό θα περιγράψουμε το περιβάλλον ανάπτυξης (Development Environment) του Matlab.



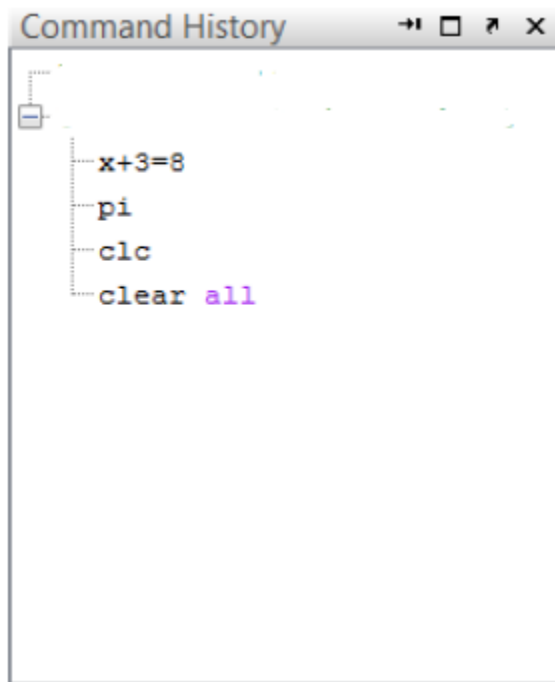
Εικόνα 12 - Matlab Desktop (Επιφάνεια Εργασίας Matlab)

Το MATLAB Desktop (Επιφάνεια Εργασίας MATLAB) αποτελεί τη βάση του Περιβάλλοντος Ανάπτυξης του συγκεκριμένου εργαλείου. Παρέχει στο χρήστη ένα προγραμματιστικό περιβάλλον στο οποίο του δίνεται η δυνατότητα να εκτελέσει εντολές, να αστικοποιήσει δεδομένα και αποτελέσματα, να δημιουργήσει και να τροποποιήσει αρχεία mfiles (απλά αρχεία κειμένου με κατάληξη .m που περιέχουν εντολές MATLAB) και να λάβει στοχευόμενη βοήθεια. Αποτελείται από έναν αριθμό άλλων παραθύρων των οποίων η χρησιμότητα θα αναλυθεί στην συνέχεια.



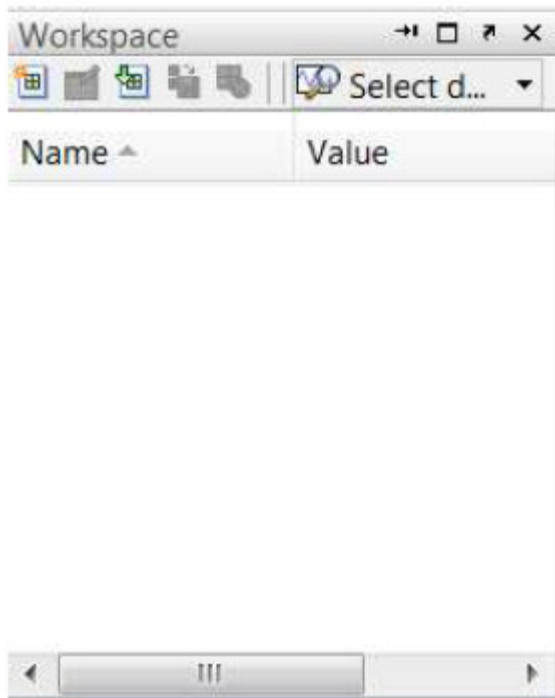
Εικόνα 13 - Παράθυρο Εντολών (Command Window)

Στο παράθυρο εντολών μπορούμε να εκτελέσουμε διάφορες εντολές σε γλώσσα Matlab, προκειμένου να ολοκληρώσουμε έναν αριθμό εργασιών, όπως για παράδειγμα την δημιουργία και επεξεργασία μεταβλητών (variables), τη δημιουργία, τροποποίηση και κλήση αρχείων m-files, τη διαχείριση δεδομένων και πολλά άλλα. Στο παράθυρο αυτό η εκτέλεση της εντολής πραγματοποιείται με τη χρήση του πλήκτρου «Enter» και τα αποτελέσματα της εντολής γίνονται άμεσα εμφανή κάτω από την εντολή μας. Να επισημάνουμε σε αυτό το σημείο ότι με τη χρήση του πλήκτρου «Tab» εμφανίζεται μια λίστα βοήθειας με όλες τις δυνατές εντολές, συναρτήσεις και μεταβλητές που αρχίζουν από τους χαρακτήρες που έχουμε εισάγει. Επίσης, είναι δυνατή η επανάκληση προηγούμενα εκτελεσμένων εντολών απλά χρησιμοποιώντας τα Up & Down Arrow Keys (πάνω και κάτω βελάκι).



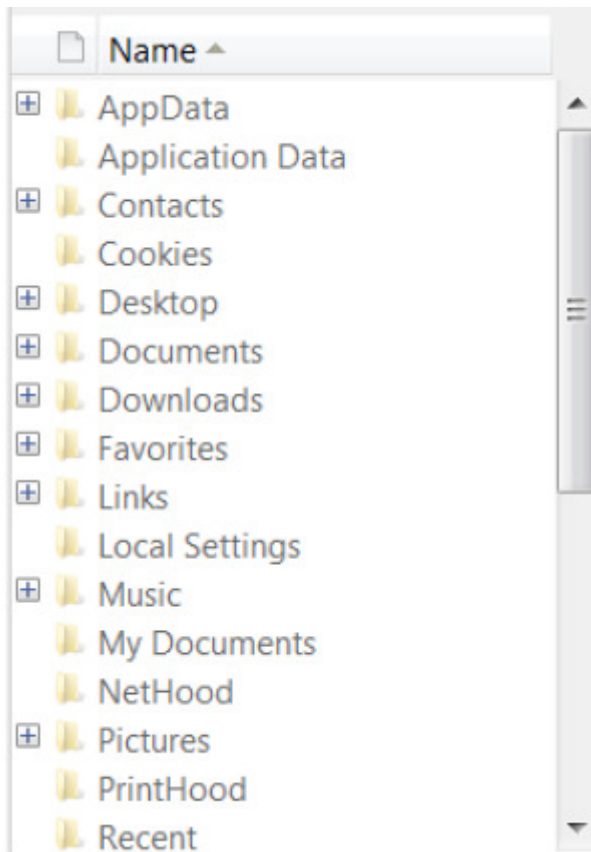
Εικόνα 14 - Ιστορικό Εντολών (Command History)

Όλες οι εντολές που έχουμε εκτελέσει στο παράθυρο εντολών (Command Window) αποθηκεύονται στο παράθυρο αυτό. Οι εγγραφές αυτές, ταξινομούνται σύμφωνα με την ώρα εκτέλεσής τους. Μπορούμε με τη βοήθεια αυτού του παραθύρου να εκτελέσουμε ξανά εντολές που έχουμε ήδη εισάγει, ή να τις επεξεργαστούμε περισσότερο σέρνοντας τις στο παράθυρο εντολών. Οι εγγραφές του Ιστορικού Εντολών, τέλος, μπορούν να διαγραφούν, να ομαδοποιηθούν, να τροποποιηθούν ή να σωθούν σε ένα m-file.



**Εικόνα 15 - Χώρος εργασίας(Workspace)**

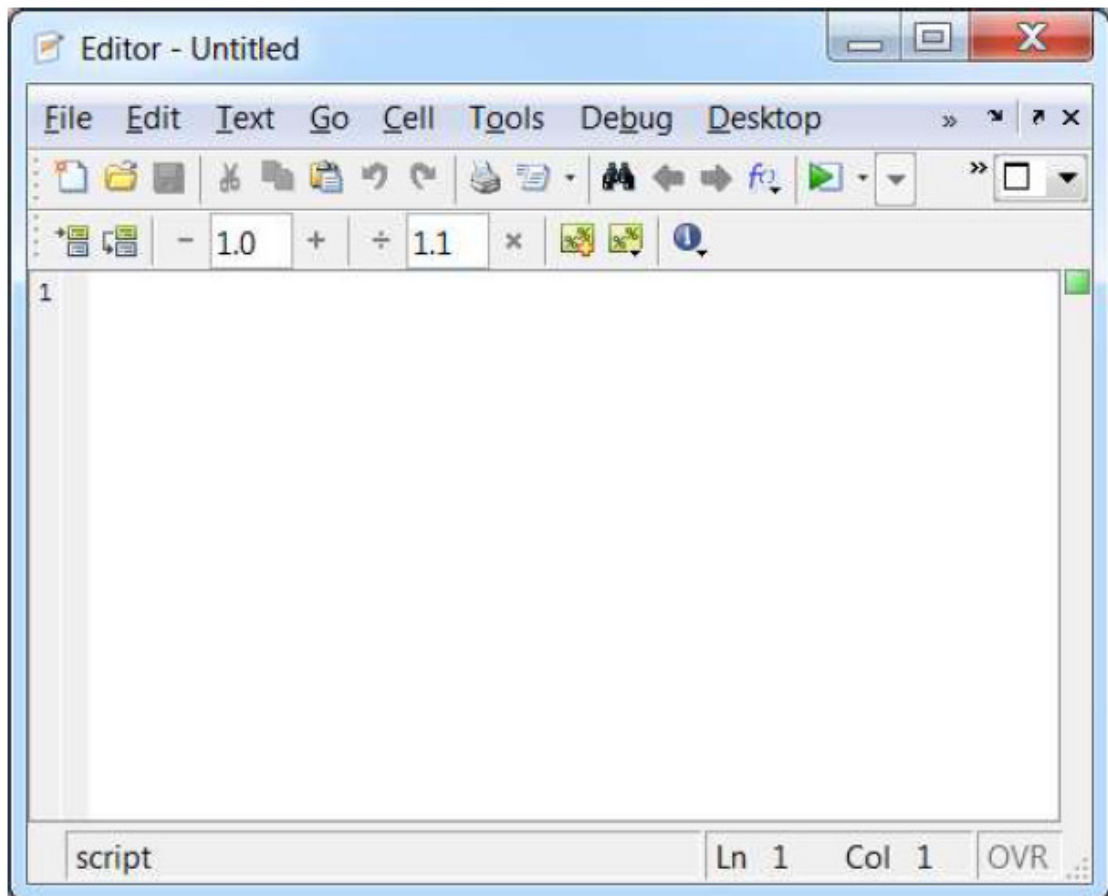
Ο χώρος εργασίας αποτελείται από όλες τις μεταβλητές που έχουν οριστεί στο παράθυρο εντολών και είναι προσβάσιμες από όλα τα προγράμματα. Κάθε μεταβλητή που αποθηκεύεται έχει διάφορες πληροφορίες όπως η τιμή της, το όνομά της, διάφορα στατιστικά στοιχεία και άλλα. Οι πληροφορίες αυτές γίνονται εμφανείς στον χρήστη μέσω του Workspace Browser. Δυνατότητες όπως η διαδραστική ανάγνωση και τροποποίηση των χαρακτηριστικών των αποθηκευμένων μεταβλητών (με διπλό-κλικ επάνω στην επιθυμητή μεταβλητή και χρήση του Array Editor) και η γραφική απεικόνισή τους (plotting) είναι μερικές από τις παρεχόμενες από αυτόν ευκολίες. Η συμβολή του, δε, κατά την διαδικασία της αποσφαλμάτωσης (debugging) προγραμμάτων είναι καταλυτική.



Εικόνα 16 - Τρέχων Κατάλογος (Current Directory)

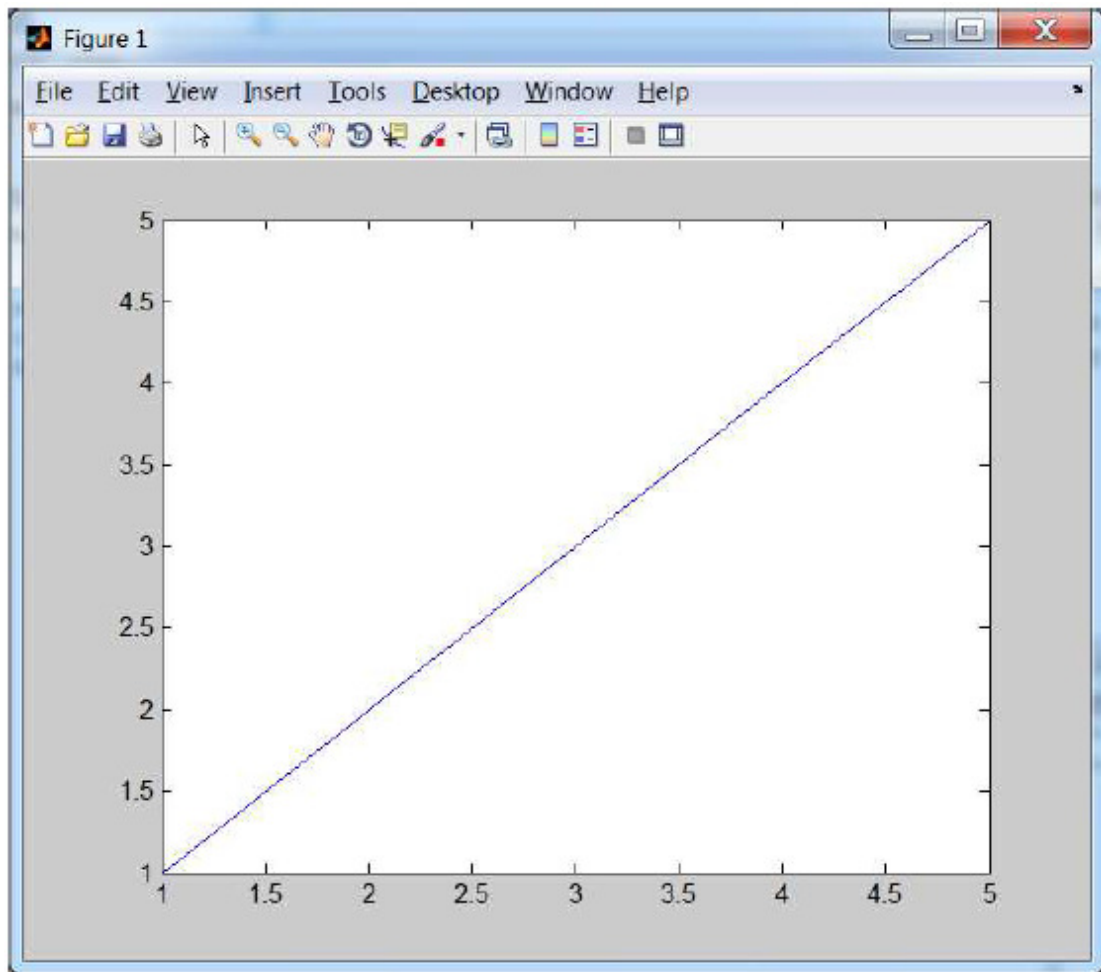
Κάθε λειτουργία η οποία στηρίζεται στην χρήση αρχείων, χρησιμοποιεί τον Τρέχοντα Κατάλογο (Current Directory) ως σημείο αναφοράς. Κάθε αρχείο που επιθυμούμε να εκτελέσουμε ή να χρησιμοποιήσουμε θα πρέπει αναγκαστικά να βρίσκεται μέσα στο συγκεκριμένο φάκελο εκτός και αν εμείς το ορίσουμε διαφορετικά. Στον κατάλογο αυτό παρουσιάζονται τα περιεχόμενα του, επιτρέποντας την περιήγηση του χρήστη στο σύστημα αρχείων του εκάστοτε υπολογιστή, το άνοιγμα (με το κατάλληλο προκαθορισμένο πρόγραμμα), την αναζήτηση και τροποποίηση σχετικών αρχείων ή καταλόγων καθώς και την διαχείριση m-files.





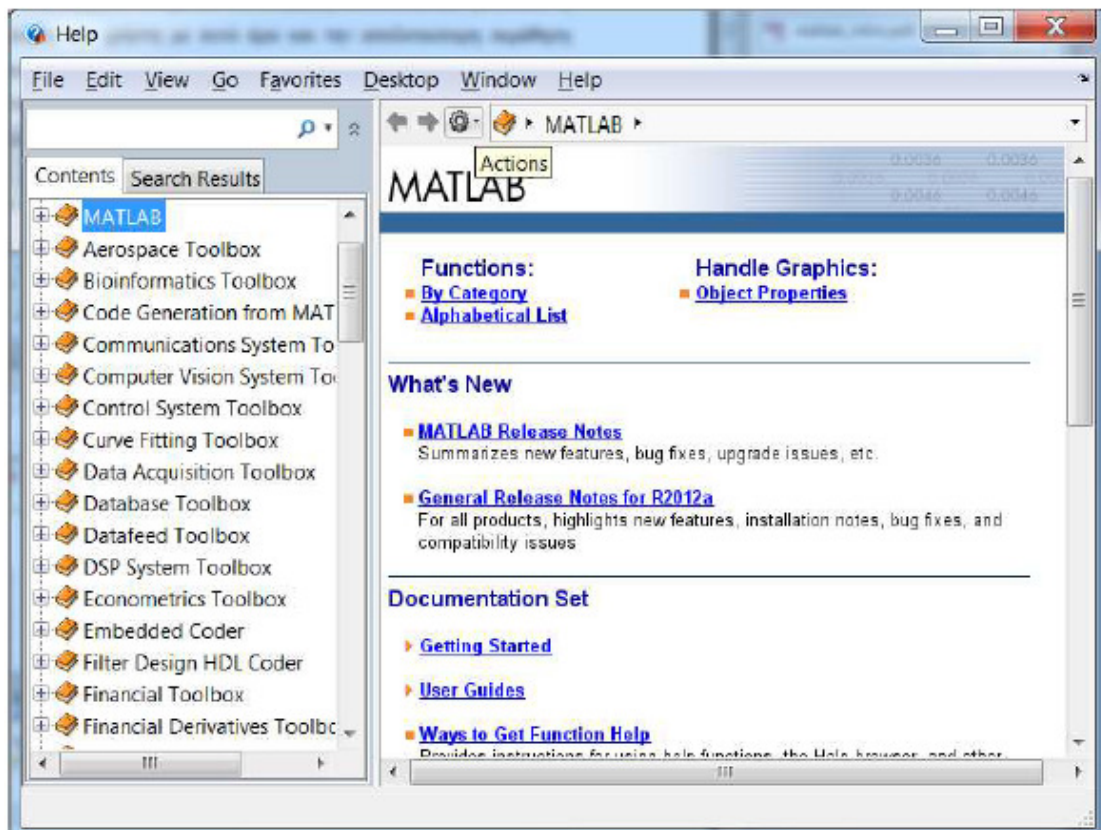
Εικόνα 17 - Παράθυρο Εισαγωγής Κειμένου (Editor window)

Στο παράθυρο αυτό μπορούμε να δημιουργήσουμε μια ακολουθία εντολών και συναρτήσεων και να τα αποθηκεύσουμε ως m-file, έτσι ώστε να μπορέσουμε να τα εκτελέσουμε οποιαδήποτε στιγμή. Το παράθυρο αυτό εμφανίζεται είτε με το πάτημα του κουμπιού «new» είτε πληκτρολογώντας την εντολή «edit» στο παράθυρο εντολών. Αφού δημιουργήσουμε το αρχείο μπορούμε είτε να το σώσουμε στον τρέχων κατάλογο είτε να το εκτελέσουμε άμεσα με τη χρήση του κουμπιού «Run».



Εικόνα 18 - Παράθυρο Σχημάτων (Figure window)

Στο Παράθυρο Σχημάτων σχεδιάζουμε γραφικές παραστάσεις με τη χρήση κατάλληλων σχεδιαστικών εντολών οι οποίες γράφονται είτε στη γραμμή εντολών είτε σε προγράμματα MATLAB.



Εικόνα 19 - Παράθυρο Βοήθειας (Help Window)

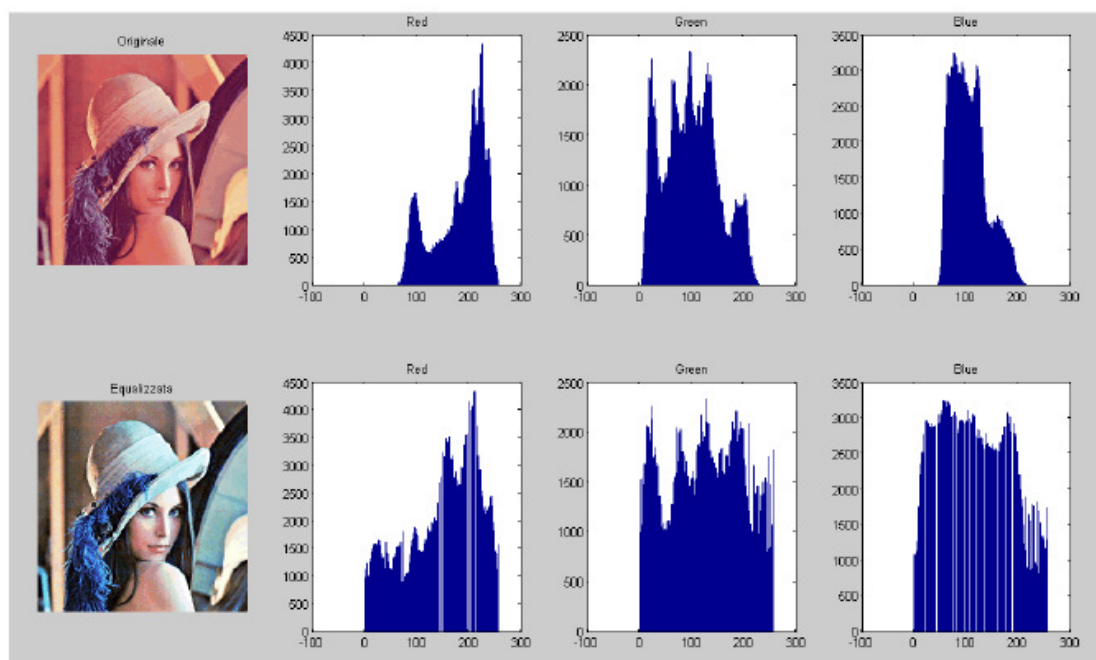
Το παράθυρο αυτό είναι ένα πολύ χρήσιμο εργαλείο τόσο για αρχάριους όσο και για έμπειρους χρήστες καθώς παρέχει όλες τις απαραίτητες πληροφορίες για οποιαδήποτε συνάρτηση. Μπορούμε να το χρησιμοποιήσουμε γράφοντας την εντολή «help» και το όνομα της συνάρτησης για την οποία θα θέλαμε περισσότερες πληροφορίες.

### Αρχεία προγραμμάτων και συναρτήσεων

Το Matlab μας δίνει τη δυνατότητα να εργαστούμε με δύο τρόπους. Μπορούμε να χρησιμοποιήσουμε μεμονωμένες εντολές στο παράθυρο εντολών και να πάρουμε άμεσα αποτελέσματα είτε μέσω προγραμμάτων τα οποία γράφονται από το χρήστη στο παράθυρο εισαγωγής κειμένου. Τα προγράμματα αυτά αποθηκεύονται με την κατάληξη .m σαν αρχεία απλού κειμένου και ονομάζονται m-files. Τα ονόματα των m-file δεν επιτρέπεται να περιλαμβάνουν τελείες και συγκεκριμένους ειδικούς χαρακτήρες. Το πρόγραμμα δεν είναι απαραίτητο να περάσει το στάδιο της μεταγλώττισης(compiling), όπως γίνεται σε άλλες γλώσσες προγραμματισμού αλλά γίνεται αυτόματα πριν το στάδιο της εκτέλεσης.

Υπάρχουν δυο ειδών τύποι m-files. Τα αρχεία προγραμμάτων(scripts) και τα αρχεία συναρτήσεων(functions). Τα αρχεία προγραμμάτων είναι αυτόνομα, δεν δέχονται ορίσματα εισόδου και δεν επιστρέφουν μεταβλητές εξόδου. Χρησιμοποιούν τις υπάρχουσες μεταβλητές που βρίσκονται στο χώρο εργασίας ή δημιουργούν καινούργιες.

Τα αρχεία συναρτήσεων από την άλλη, δέχονται υποχρεωτικά ορίσματα εισόδου και μπορούν να επιστρέψουν μεταβλητές εξόδου. Το όνομα μιας συνάρτησης μαζί με τα ορίσματα εισόδου και τις μεταβλητές εξόδου γράφονται στην πρώτη γραμμή του αντίστοιχου m-file συνοδευόμενα από τη λέξη-κλειδί function, ενώ στην τελευταία γραμμή του αρχείου, γράφεται η λέξη-κλειδί end. Επίσης, το m-file που περιέχει μια συνάρτηση πρέπει να έχει το ίδιο όνομα με αυτή. Τέλος να αναφέρουμε ότι μια συνάρτηση χρησιμοποιεί το δικό της χώρο μεταβλητών, ο οποίος δεν είναι προσπελάσιμος μετά την εκτέλεσή της. Οι συναρτήσεις μπορούν να κληθούν μέσα από scripts ή από άλλες συναρτήσεις για διάφορες τιμές των ορισμάτων εισόδου τους.



Εικόνα 20 - Παράδειγμα χρήσης του Matlab στην επεξεργασία εικόνας

## **4.2 Ανάπτυξη εφαρμογής**

### **4.2.1 Απαιτήσεις Εφαρμογής**

Η εφαρμογή η οποία θα αναπτυχτεί στα πλαίσια της διπλωματικής μου εργασίας με τη χρήση κάμερας ή τη φόρτωση κάποιου video το οποίο θα ορίζει ο χρήστης θα εντοπίζει τα κινούμενα αντικείμενα. Τα κινούμενα αντικείμενα θα περιβάλλονται από ένα πλαίσιο. Στη συνέχεια εάν το κινούμενο αντικείμενο είναι εκτός κάποιας περιοχής που οριστεί τότε το πλαίσιο που το περιβάλλει θα αλλάζει χρώμα. Όταν είναι εντός της υποπεριοχής που έχει οριστεί θα είναι χρώμα κίτρινο και όταν είναι εκτός υποπεριοχής θα είναι χρώμα κόκκινο.

### **4.2.2 Ανάλυση Εφαρμογής**

Η εφαρμογή η οποία αναπτύχθηκε αποτελεί μια ολοκληρωμένη προσέγγιση λύσης στο πρόβλημα της αυτόματης ανίχνευσης και παρακολούθησης κινούμενων αντικειμένων από ένα βίντεο που καταγράφηκε από ακίνητη κάμερα. Ο αλγόριθμος είναι βασισμένος στην κίνηση (motion - based) των αντικειμένων και χωρίζεται σε δύο βασικά μέρη:

- Την ανίχνευση των κινούμενων αντικειμένων σε κάθε frame.
- Την συσχέτιση των ανιχνεύσεων που καταγράφηκαν στα διάφορα frames έτσι ώστε να ανταποκρίνονται στο ίδιο αντικείμενο κατά την πάροδο του χρόνου. (παρακολούθηση)

Για την ανίχνευση των κινούμενων αντικειμένων χρησιμοποιήθηκε ένας αλγόριθμος αφαίρεσης παρασκηνίου (background subtraction algorithm) βασισμένος σε Gaussian mixtures models. Στη συνέχεια εφαρμόζονται στη μάσκα του προσκηνίου (foreground mask) μορφολογικές διαδικασίες με σκοπό την εξάλειψη του θορύβου. Τέλος μέσω ανάλυσης blob γίνεται η ανίχνευση ομάδων από συνδεδεμένα pixels, τα οποία πιθανόν να ανήκουν σε κινούμενα αντικείμενα.

Για τον αλγόριθμο κάθε αντικείμενο αποτελεί ένα ίχνος (track). Ο συσχετισμός των ανιχνεύσεων για το ίδιο track γίνεται αποκλειστικά βάση της κίνησης των αντικειμένων. Η κίνηση κάθε track εκτιμάται μέσω του φίλτρου Kalman που

αναπτύχθηκε σε προηγούμενη ενότητα. Το φίλτρο Kalman χρησιμοποιείται για την εκτίμηση της θέσης που θα έχει το κάθε αντικείμενο σε κάθε frame, και καθορίζει την πιθανότητα κάθε ανίχνευσης να αποτελεί την επόμενη θέση των προϋπαρχόντων tracks.

Η διατήρηση της συνοχής των tracks κατά τη διάρκεια εξέλιξης του βίντεο αποτελεί σημαντικό κομμάτι του αλγορίθμου. Σε κάθε δοσμένο frame, μέρος του συνόλου των ανιχνύσεων, ανατίθενται σε tracks (assigned tracks), ενώ κάποιες ανιχνεύσεις και tracks παραμένουν απροσδιόριστα (unassigned). Βάσει της θέσης του κάθε αντικειμένου καθορίζεται εάν είναι μέσα ή έξω από μία υποπεριοχή του βίντεο και ανάλογα χρωματίζεται η υποπεριοχή και το κινούμενο αντικείμενο.

Στην παραπάνω εικόνα παρουσιάζεται ο χωρισμός της εικόνας του video σε υποπεριοχές. Όπως φαίνεται από την παρακάτω εικόνα, το βίντεο χωρίζεται σε 3 ίσες υποπεριοχές.

33,33% πλάτους	συνολικού	33,33% πλάτους	συνολικού	33,33% πλάτους	συνολικού
Όταν το κινούμενο αντικείμενο είναι μέσα σε αυτήν την υποπεριοχή τότε το πλαίσιο γύρω του είναι κόκκινο		Όταν το κινούμενο αντικείμενο είναι μέσα σε αυτήν την υποπεριοχή τότε το πλαίσιο γύρω του είναι κίτρινο		Όταν το κινούμενο αντικείμενο είναι μέσα σε αυτήν την υποπεριοχή τότε το πλαίσιο γύρω του είναι κόκκινο.	

Εικόνα 21 - Ορισμός Υποπεριοχής

## ❖ Γενική περιγραφή αλγορίθμου σε στάδια

### **Στάδιο 1: Δημιουργία System Objects.**

Στο πρώτο στάδιο γίνεται η προετοιμασία για την επεξεργασία (καθορίζονται input / output) του αρχείου βίντεο. Η συνάρτηση που αποτελεί το πρώτο αυτό στάδιο, δημιουργεί objects, δηλαδή λογικές δομές (logical structures), από την ανάγνωση του αρχείου. Στη συνέχεια εξάγονται τα objects που ανιχνεύονται σε κάθε frame.

### **Στάδιο 2: Προετοιμασία των tracks.**

Κάθε ξεχωριστό αντικείμενο που εντοπίζει ο αλγόριθμος το θεωρεί ως ένα track. Κάθε track αποτελεί ένα διάνυσμα από πληροφορίες για το αντικείμενο. Στο στάδιο αυτό καθορίζονται ποιές είναι αυτές οι πληροφορίες που περιέχει το κάθε track.

### **Στάδιο 3: Καθορισμός δεικτών.**

Καθορίζονται δυο δείκτες να είναι ίσοι με το 1. Ο πρώτος ονομάζεται nextId και αναφέρεται στην ταυτότητα του επόμενου track.

### **Στάδιο 4: Συνθήκη επανάληψης.**

Τα επόμενα στάδια του αλγορίθμου αποτελούν μια επαναληπτική διαδικασία για όλα τα frames του βίντεο. Έτσι το στάδιο αυτό αποτελεί μια συνθήκη της μορφής while object reader is not done then:

- Στην περίπτωση που το αποτέλεσμα είναι TRUE τότε ο αλγόριθμος συνεχίζει στα επόμενα στάδια (στάδια 5-13)
- Στην περίπτωση που το αποτέλεσμα είναι FALSE σημαίνει ότι έχουν επεξεργαστεί όλα τα frame και το επόμενο βήμα του αλγορίθμου είναι το στάδιο 14.

### **Στάδιο 5: Ανάγνωση frame.**

Στο στάδιο αυτό γίνεται η ανάγνωση του εκάστοτε frame του βίντεο. Η διαδικασία αυτή γίνεται μέσω του object reader, step by step.

### **Στάδιο 6: Ανίχνευση αντικειμένων και εξαγωγή πληροφοριών.**

Μέσω της συνάρτησης detectObject γίνεται η ανίχνευση των tracks και η καταγραφή των πληροφοριών : centroids, bboxes, mask.

### **Στάδιο 7: Εκτίμηση καινούριας τοποθεσίας των tracks.**

Εδώ γίνεται η εκτίμηση της τοποθεσίας που θα έχουν τα tracks, μέσω του Kalman Filter

### **Στάδιο 8: Ανάθεση ανίχνευσης στα tracks.**

Η ανάθεση των αντικειμένων που ανιχνεύτηκαν στο παρόν frame στα ήδη υπάρχοντα tracks γίνεται μέσω της ελαχιστοποίησης του κόστους. Το κόστος ορίζεται ως ο **αρνητικός λογάριθμος των ανιχνεύσεων που αντιστοιχούν σε κάθε track.**

### **Στάδιο 9: Εκσυγχρονισμός προσδιορισμένων tracks.**

Στην περίπτωση που καταγεγραμμένα tracks προσδιορίζονται στο frame (δηλαδή υπάρχει αντιστοίχιση ανίχνευσης σε προϋπάρχον track) τότε γίνεται ο εκσυγχρονισμός τους σύμφωνα με την τοποθεσία της ανίχνευσης.

### **Στάδιο 10: Εκσυγχρονισμός μη προσδιορισμένων tracks.**

Στην περίπτωση που σε κάποια tracks δεν βρεθεί αντιστοιχία με κάποια ανίχνευση, αυτά τα tracks μαρκάρονται ως αόρατα. Κάθε φορά που δεν υπάρχει ανίχνευση που να αντιστοιχεί σε αυτά τα tracks, αυξάνεται η "ηλικία" τους κατά 1.

### **Στάδιο 11: Διαγραφή χαμένων tracks.**

Η συνάρτηση που αποτελεί αυτό το στάδιο, διαγράφει τα tracks που είναι αόρατα για πολλά συνεχόμενα frames. Διαγράφει επίσης και νεοεισαχθείσα tracks τα οποία ήταν αόρατα για πάρα πολλά frames στο σύνολο.

### **Στάδιο 12: Δημιουργία καινούριων tracks.**

Οι ανιχνεύσεις που γίνονται και δεν έχουν ανατεθεί σε κάποιο προϋπάρχον track, δημιουργούν καινούρια tracks.

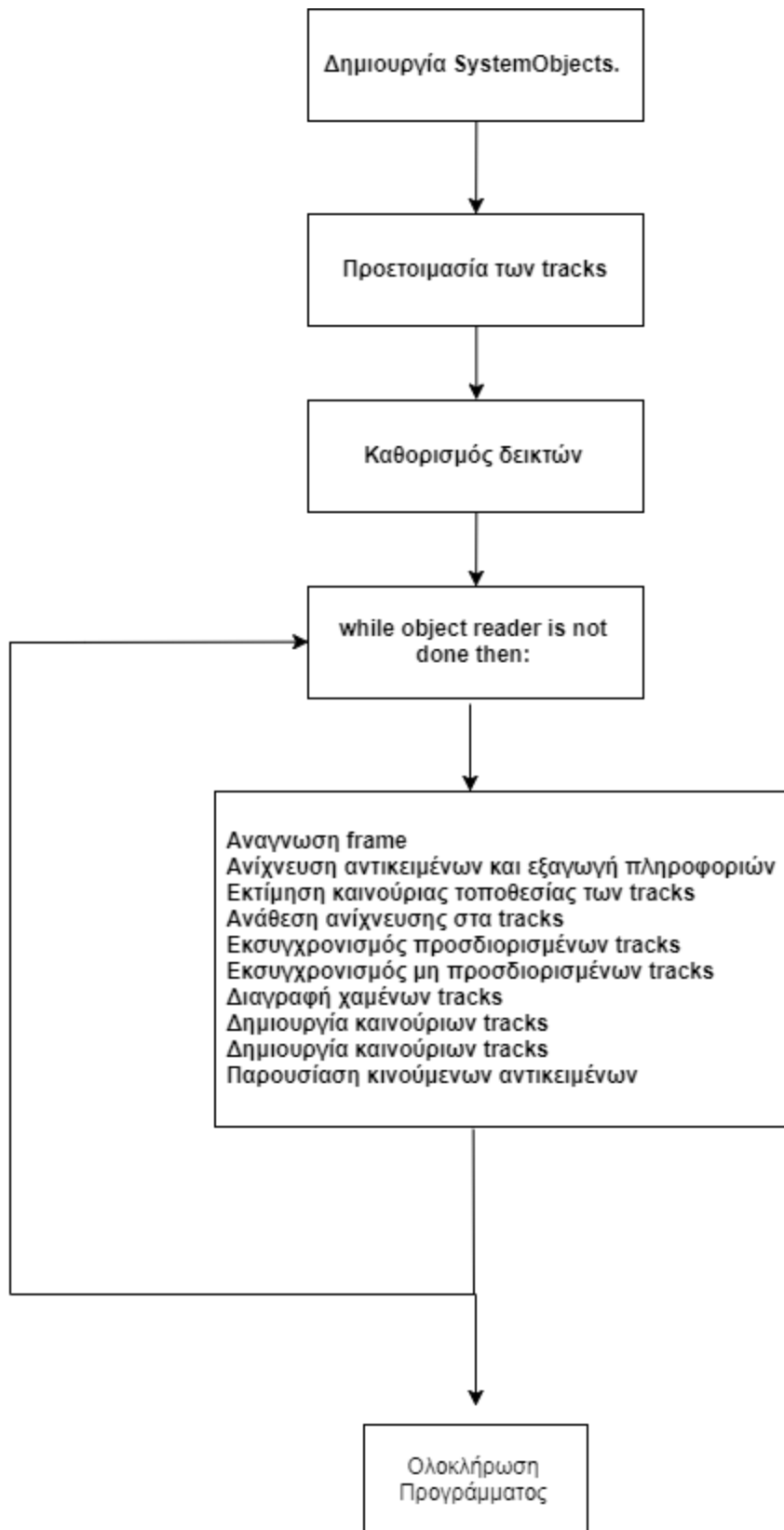
### **Στάδιο 13: Δημιουργία καινούριων tracks.**

Στο στάδιο αυτό καθορίζονται ποιά tracks είναι αξιόπιστα (Reliable tracks). Για να είναι αξιόπιστο ένα track πρέπει να έχει εμφανιστεί σε τουλάχιστον οκτώ συνεχόμενα frames. Στη συνέχεια, εφόσον το track είναι αξιόπιστο, καταγράφονται οι πληροφορίες του στους πίνακες id,x,y. Τέλος γίνεται η προβολή του αρχικού εκάστοτε frame καθώς και της δυαδικής μάσκας, με ζωγραφισμένα πάνω τους τα bbox και το trackId των ανιχνεύσεων. Για καλύτερη κατανόηση του αλγορίθμου, στις περιπτώσεις που το track δεν κινείται προς την τοποθεσία που εκτιμήθηκε πως θα έχει



στο επόμενο frame σύμφωνα με το Kalman filter, ζωγραφίζεται στη εικόνα αυτή η εκτιμώμενη περιοχή με ετικέτα "predicted". Εάν το κινούμενο αντικείμενο είναι στην υποπεριοχή τότε το περίγραμμα είναι κίτρινο, διαφορετικά είναι κόκκινο.

**Στάδιο 14: Ολοκλήρωση προγράμματος.**



Εικόνα 22 - Διάγραμμα ροής

### 4.2.3 Επεξήγηση Κώδικα

Στο σημείο αυτό γίνεται η λεπτομερής περιγραφή των σταδίων που αναφέρθηκαν πιο πάνω. Δηλαδή θα αναλυθούν εις βάθος τα στάδια που χρίζουν περαιτέρω επεξήγησης.

#### Δήλωση & Αρχικοποίηση Μεταβλητών

Δημιουργία object για ανάγνωση του αρχείου.

```
obj.reader = vision.VideoFileReader('file_name.avi')
```

Δημιουργία δυο Objects ως Video players.

Το πρώτο video player παρουσιάζει το αρχικό βίντεο ενώ το δεύτερο τη δυαδική μάσκα του αρχικού βίντεο. Για τα δυο βίντεο player καθορίζονται και οι συντεταγμένες του τετράπλευρου που θα καταλαμβάνουν στην οθόνη.

```
obj.videoPlayer = vision.VideoPlayer('Position', [20, 400, 700, 400]);  
obj.maskPlayer = vision.VideoPlayer('Position', [740, 400, 700, 400]);
```

Δημιουργία object για ανίχνευση προσκηνίου (foreground detection).

Η ανίχνευση του προσκηνίου χρησιμοποιείται για τον διαχωρισμό των κινούμενων αντικειμένων από το παρασκήνιο. Προϊόν της συνάρτησης είναι η εξαγωγή της δυαδικής μάσκας (binary mask) του frame, στην οποία τα pixels, με τιμή 1 αντιπροσωπεύουν το προσκήνιο και αυτά με τιμή 0 αντιπροσωπεύουν το παρασκήνιο.

Καθορίζονται οι παράμετροι της συνάρτησης ForegroundDetector

```
obj.detector = vision.ForegroundDetector('NumGaussians', 3,  
... 'NumTrainingFrames', 10, 'MinimumBackgroundRatio', 0.7);
```

Η παράμετρος 'NumGaussians' αναφέρεται στον αριθμό των Gaussian που θα χρησιμοποιηθούν στο μοντέλο μίξης (Gaussian mixture model). Στη συγκεκριμένη εργασία επιλέγηκε ο αριθμός αυτός να είναι ίσος με 3. Η τιμή αυτή συνήθως κυμαίνεται μεταξύ 3 - 5.

Η παράμετρος 'NumTrainingFrames' αφορά τον αριθμό των frames που θα θεωρηθούν από τον αλγόριθμο ως frames εκπαίδευσης (training frames). Μετά από δοκιμές η τιμή αυτή ορίστηκε να είναι ίση με 10.

Τέλος η παράμετρος 'MinimumBackgroundRatio' είναι η ελάχιστη αναλογία παρασκηνίου. Στην ουσία αποτελεί ένα όριο για τον καθορισμό του background model. Η τιμή του ορίου αυτού επιλέγηκε να είναι ίση με 0,7 στα θερμικά δεδομένα και 0,4 στα οπτικά.

Δημιουργία object για blob ανάλυση.

Συνδεδεμένες ομάδες από pixels προσκηνίου είναι πιθανό να αντιστοιχούν σε κινούμενα αντικείμενα. Για τον εντοπισμό τέτοιων ομάδων γίνεται η χρήση της συνάρτησης blobAnalysis, η οποία υπολογίζει τα χαρακτηριστικά αυτών των ομάδων (pixels). Τα χαρακτηριστικά είναι : το μέγεθος (area), το centroid και το bounding box.

```
obj.blobAnalyser =  
vision.BlobAnalysis('BoundingBoxOutputPort', ... true,  
'AreaOutputPort', true, 'CentroidOutputPort', true,  
...'MinimumBlobArea', 0);
```

### **Προετοιμασία των tracks.**

Η συνάρτηση αυτού του σταδίου δημιουργεί μια παράταξη (array) από tracks, όπου κάθε track είναι μια δομή (structure) που αντιπροσωπεύει ένα κινούμενο αντικείμενο στο βίντεο. Ο σκοπός αυτής της δομής είναι η διατήρηση της κατάστασης (maintain the state) των αντικειμένων που ανιχνεύτηκαν.

Η κατάσταση περιέχει πληροφορίες που χρησιμοποιούνται για την ανάθεση ανιχνεύσεων σε tracks, για τον τερματισμό των tracks και για την παρουσίαση τους.

Η δομή (structure) των tracks περιέχει τα ακόλουθα πεδία:

- id : είναι η ταυτότητα του track (ακέραιος αριθμός).
- bbox: αποτελεί το παρόν πλαίσιο οριοθέτησης (bounding box) του αντικειμένου. Χρησιμοποιείται στο κομμάτι της παρουσίασης.

- Kalman filter: Αναφέρεται στο object Kalman filter που χρησιμοποιείται για την παρακολούθηση των αντικειμένων σε motion based εφαρμογές.
- age : Αντιπροσωπεύει τον αριθμό των frames που έχουν περάσει από τότε που εντοπίστηκε για πρώτη φορά το track.
- total visible count : Είναι το πεδίο που αποθηκεύει τον αριθμό του συνόλου των frames που έχει εντοπιστεί το track.
- consecutive invisible count : Στο πεδίο αυτό αποθηκεύεται ο αριθμός των συνεχόμενων frames για τα οποία το track δεν έχει ανιχνευτεί (αόρατο).

```
tracks = struct(...
    'id', {}, ...
    'bbox', {}, ...
    'kalmanFilter', {}, ...
    'age', {}, ...
    'totalVisibleCount', {}, ...
    'consecutiveInvisibleCount', {});
```

Ανιχνεύσεις που αποτελούν θόρυβο, τείνουν να παρουσιάζουν μικρή "διάρκεια ζωής". Δηλαδή εμφανίζονται σε πολύ λίγα frames.

### **Ανίχνευση αντικειμένων και εξαγωγή πληροφοριών (σε κάθε frame).**

Η συνάρτηση detect objects επιστρέφει τα centroids και τα bounding boxes των αντικειμένων που ανιχνεύτηκαν. Επιστρέφει επίσης τη δυαδική μάσκα η οποία έχει το ίδιο μέγεθος με το frame εισόδου.

Η συνάρτηση εκτελεί κατάτμηση κίνησης (motion segmentation) χρησιμοποιώντας ανιχνευτή προσκηνίου. Τέλος εκτελούνται μορφολογικές διαδικασίες στη δυαδική μάσκα για την αφαίρεση του θορύβου (noisy pixels) και για το "γέμισμα" των καινών στα εναπομείναντα blobs.

**Βήμα 1: Ανίχνευση προσκηνίου.**

```
mask = obj.detector.step(frame);
```

**Βήμα 2: Εφαρμογή μορφολογικών διαδικασιών για αφαίρεση θορύβου και γέμισμα καινών.**

```
mask = imopen(mask, strel('square', 1));
mask = imclose(mask, strel('square', 10));
```

**Βήμα 3: Εκτέλεση blob analysis για την εύρεση συνδεδεμένων pixels.**

```
[areas, centroids, bboxes] = obj.blobAnalyser.step(mask);
```

### **Ανάθεση ανίχνευσης στα tracks.**

Η ανάθεση των αντικειμένων που ανιχνεύτηκαν στο παρόν frame, στα ήδη υπάρχοντα racks, γίνεται μέσω της ελαχιστοποίησης του κόστους. Ως κόστος ορίζεται ο ρητικός λογάριθμος των ανιχνεύσεων που αντιστοιχούν σε κάθε track. \

Στάδιο 1:

Ο υπολογισμός του κόστους ανάθεσης κάθε ανίχνευσης σε κάθε track γίνεται με τη χρήση της μεθόδου <distance> (απόστασης) από τη συνάρτηση <vision.KalmanFilter>. Το κόστος λαμβάνει υπόψη την Ευκλείδεια απόσταση μεταξύ του centroid της εκτίμησης της θέσης του track (που έγινε μέσω του Kalman filter) και της θέσης του centroid της ανίχνευσης του track. Μέσα στη διαδικασία υπολογισμού του κόστους περιέχεται και ο παράγοντας αξιοπιστία της πρόβλεψης, που παρέχεται από το Kalman filter. Τα αποτελέσματα αυτής της διαδικασίας αποθηκεύονται σε ένα πίνακα διαστάσεων  $M \times N$ , όπου  $M$  είναι ο αριθμός των tracks και  $N$  ο αριθμός των ανιχνεύσεων.

Στάδιο 2:

Στο στάδιο αυτό γίνεται η λύση του προβλήματος ανάθεσης του πίνακα κόστους που δημιουργήθηκε πριν, με τη χρήση της συνάρτησης <assignedDetectionToTracks>. Η συνάρτηση παίρνει ως δεδομένα τον πίνακα κόστους και τη τιμή του κόστους που προκύπτει από τη μη ανάθεση καμίας ανίχνευσης στο track.

Η τιμή του κόστους της μη ανάθεσης καμίας ανίχνευσης στο track (costOfNonAssignment) εξαρτάται από το φάσμα των τιμών που επιστρέφει η μέθοδος <distance> της συνάρτησης <vision.KalmanFilter>. Αυτή η τιμή πρέπει να καθοριστεί πειραματικά. Αν τεθεί πολύ χαμηλά, αυξάνεται η πιθανότητα της δημιουργίας καινούριου track, ή και την διάσπαση των tracks. Αν τεθεί πολύ υψηλή μπορεί ο αλγόριθμος να εντοπίζει μόνο ένα track που να περιέχει όλα τα κινούμενα αντικείμενα.

Μετά τη διεξαγωγή αρκετών επαναλήψεων στον αλγόριθμο, τα καλύτερα αποτελέσματα προέκυψαν θέτοντας  $\text{costOfNonAssignment} = 20$ . Αξίζει να σημειωθεί ότι η συνάρτηση AssignDetectiontoTracks χρησιμοποιεί την έκδοση του Hungarian algorithm που πρότεινε ο Munkres. Ο Hungarian algorithm υπολογίζει και

αντιστοιχεί τις αναθέσεις των ανιχνεύσεων στα tracks, με τρόπο που να ελαχιστοποιείται το συνολικό κόστος.

Η συνάρτηση επιστρέφει ένα πίνακα  $M \times 2$  που περιέχει τους δείκτες των προσδιορισμένων ανιχνεύσεων και tracks στις δυο του στήλες. Επιστρέφει επίσης και τους δείκτες των ανιχνεύσεων και tracks για τα οποία προέκυψε κάποια αντιστοίχιση.

```
nTracks = length(tracks);
nDetections = size(centroids, 1);

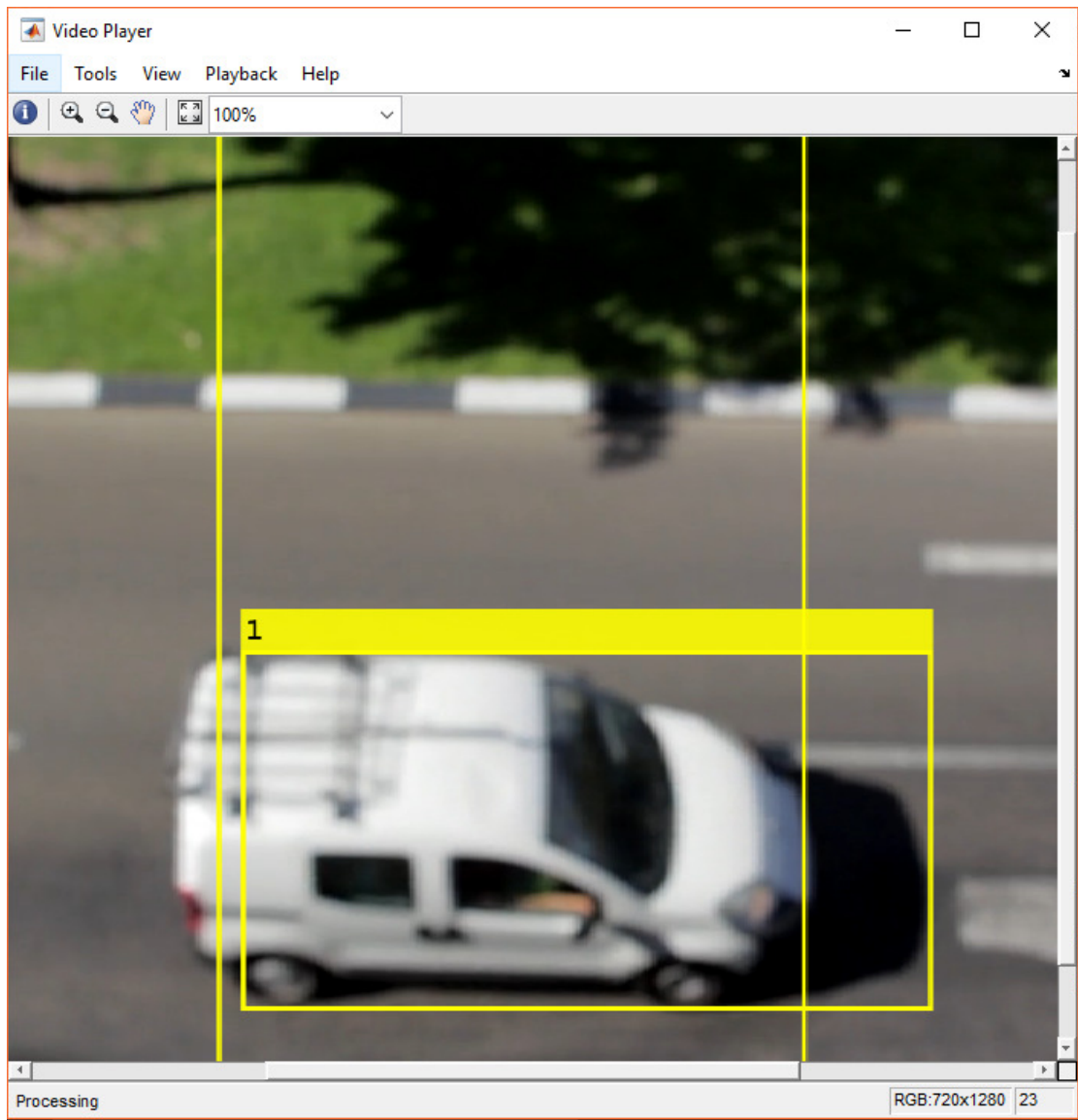
cost = zeros(nTracks, nDetections);
for i = 1:nTracks
    cost(i, :) = distance(tracks(i).kalmanFilter, centroids);
end
costOfNonAssignment = 20;
[assignments, unassignedTracks, unassignedDetections] = ...
    assignDetectionsToTracks(cost, costOfNonAssignment);
```

#### 4.2.4 Εκτέλεση κώδικα

Για την επιβεβαίωση του κώδικα χρησιμοποιήθηκε το βίντεο το οποίο επισυνάπτεται (Cars.mp4).

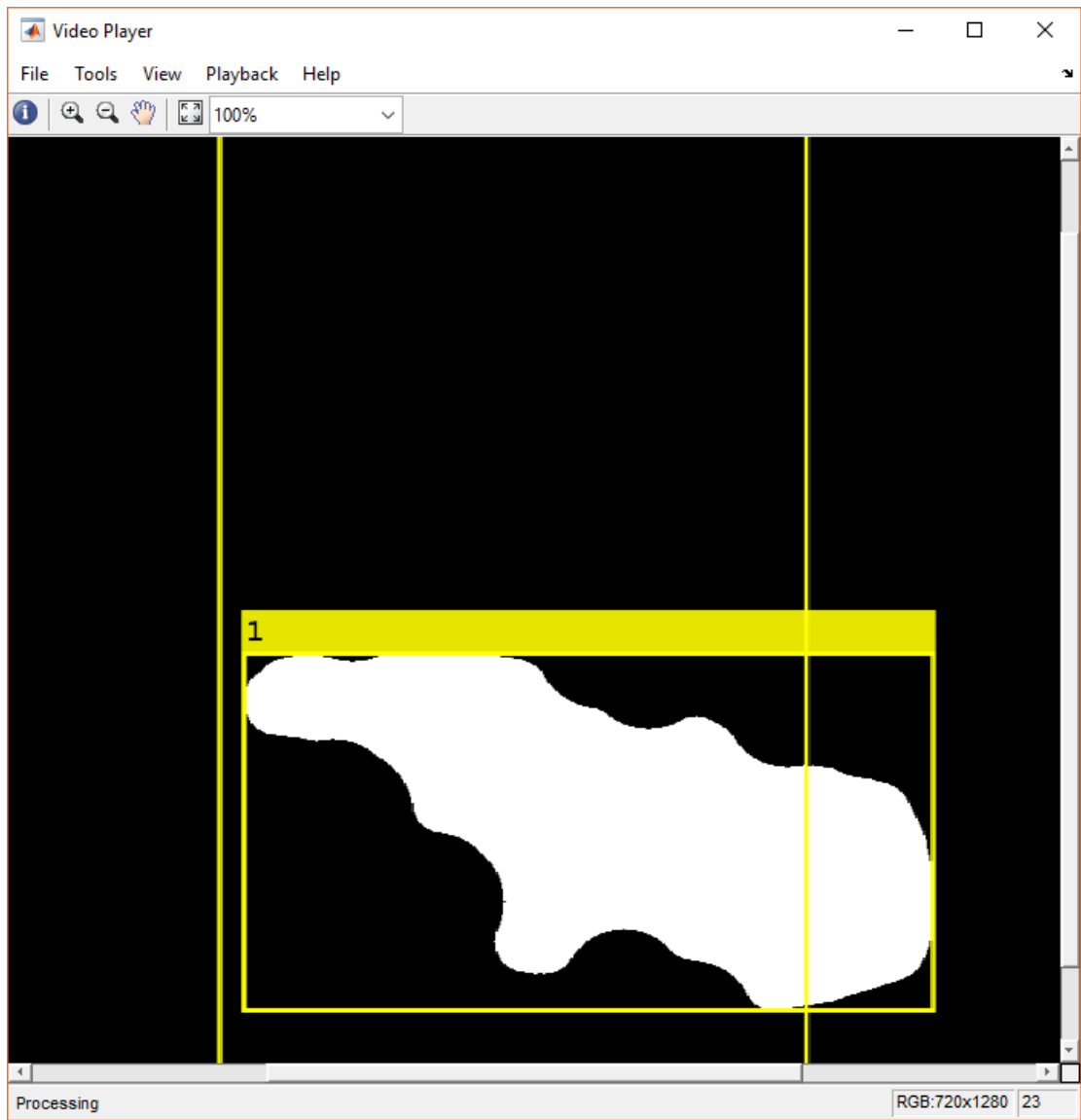
Στις εικόνες 20,21 παρουσιάζεται το όχημα το οποίο είναι στο  $23^\circ$  (καρέ του βίντεο) είναι εντός της ορισμένης υποπεριοχής.

Στις εικόνες 22,23 παρουσιάζεται το όχημα το οποίο είναι στο  $34^\circ$  (καρέ του βίντεο) είναι εντός της ορισμένης υποπεριοχής.

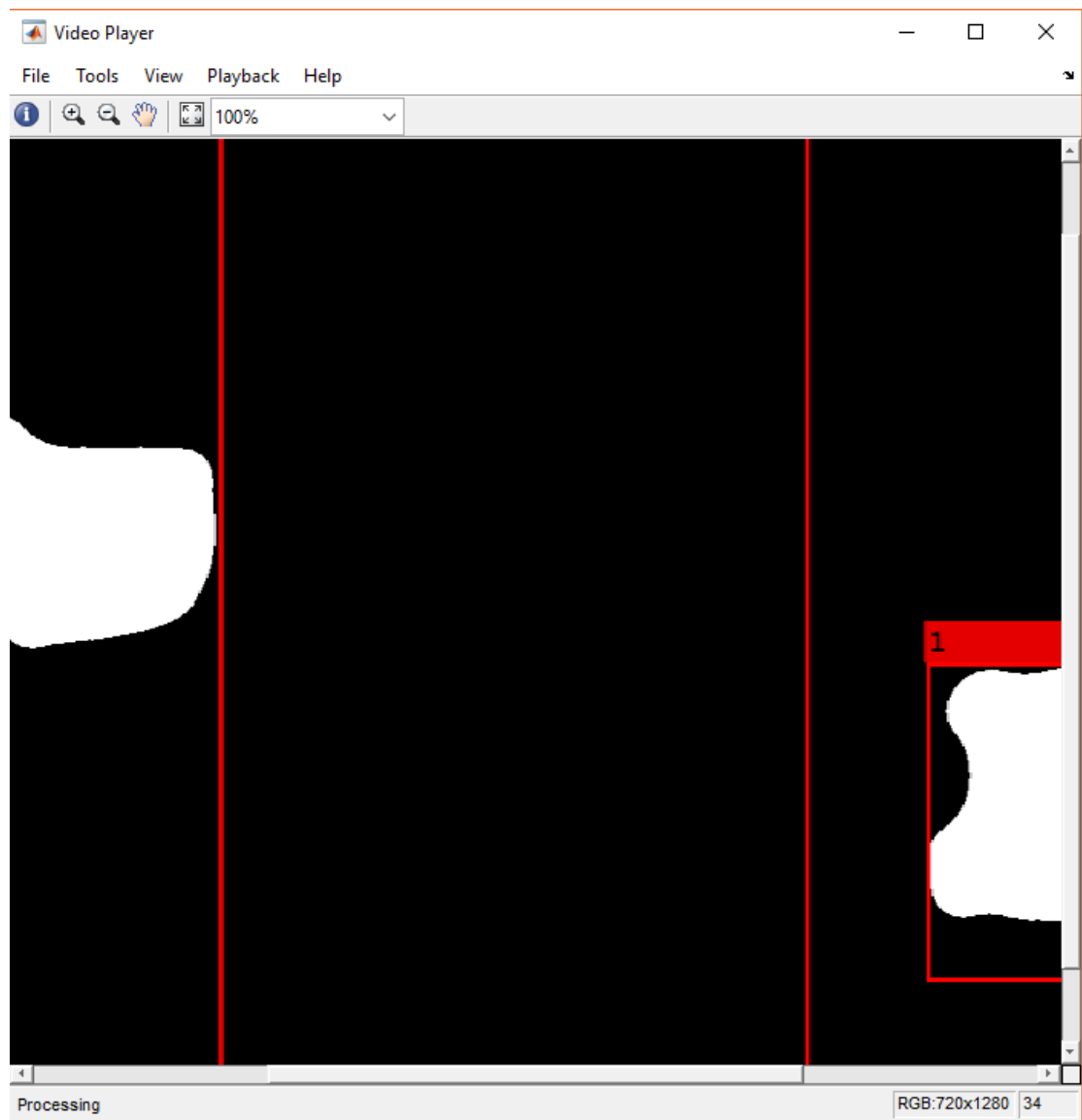


Εικόνα 23 -'Οχημα εντός υποπεριοχής

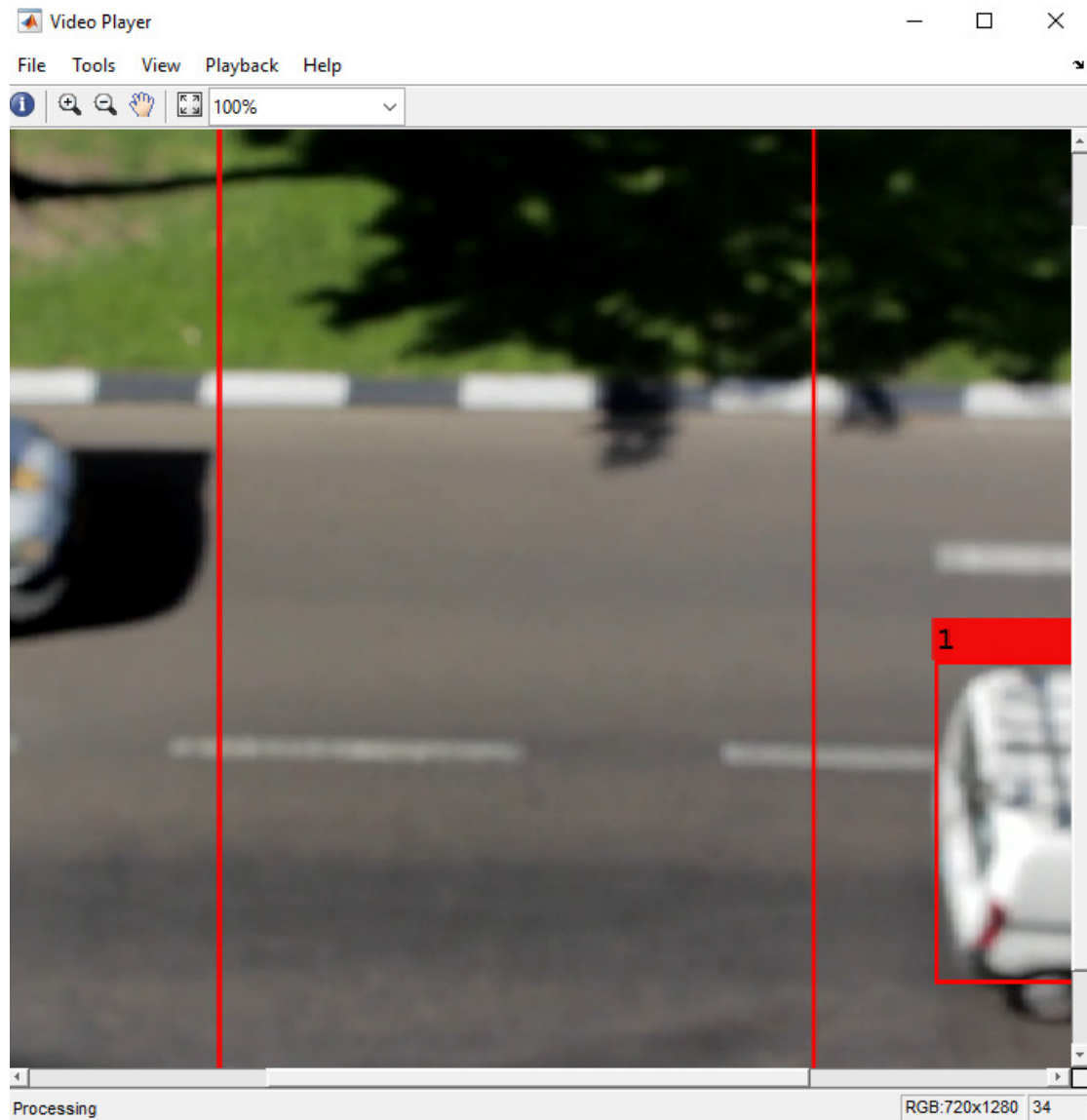




Εικόνα 24 -'Όχημα εντός υποπεριοχής



Εικόνα 25 - Όχημα εκτός υποπεριοχής

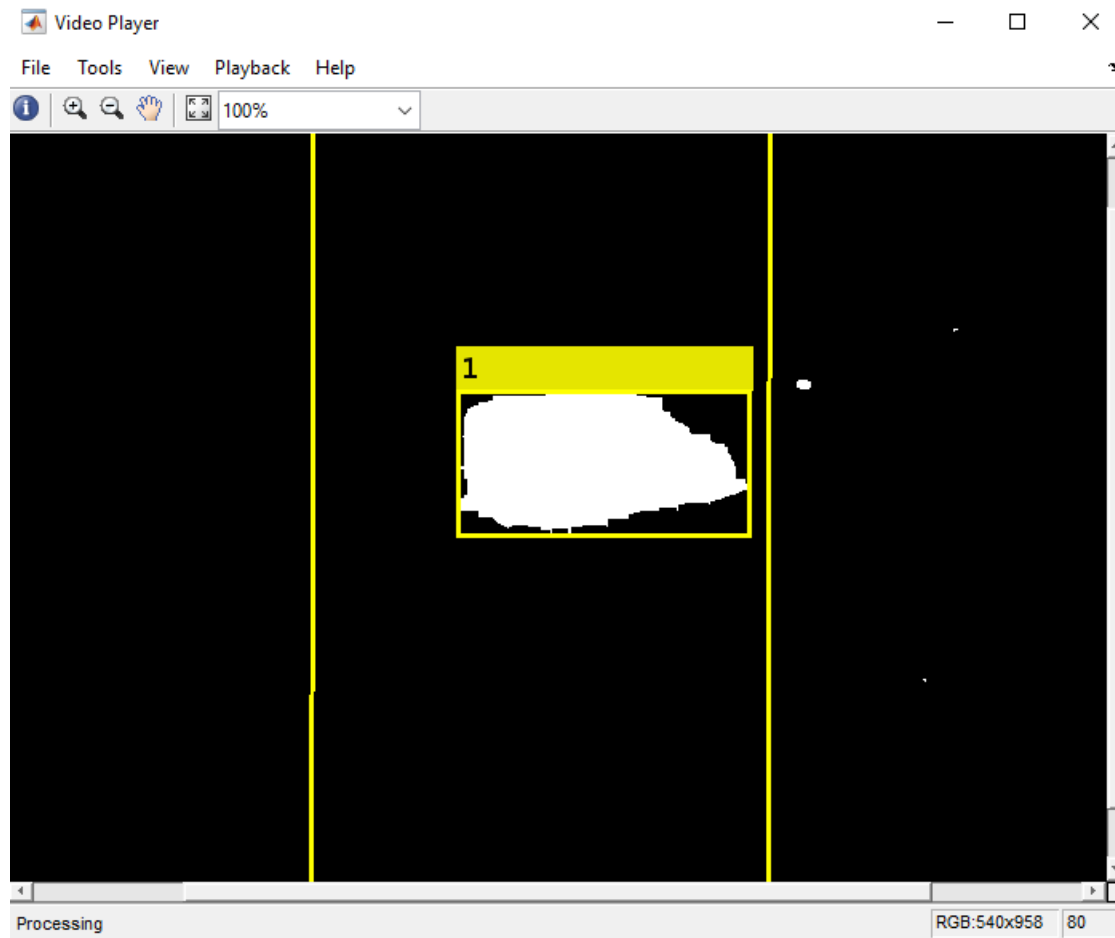


Εικόνα 26 - Όχημα εκτός υποπεριοχής

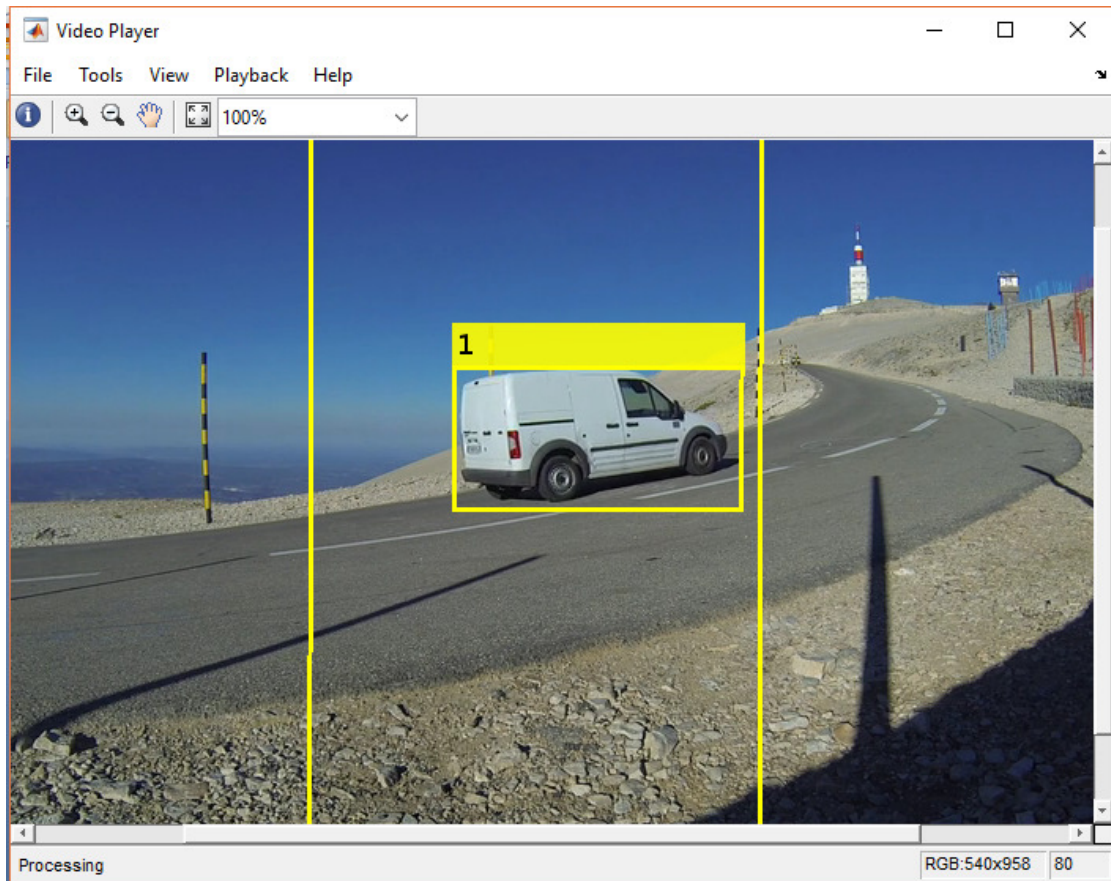
Για την επιβεβαίωση του κώδικα χρησιμοποιήθηκε το βίντεο το οποίο επισυνάπτεται (Ventux.mp4).

Στις εικόνες 24,25 παρουσιάζεται το όχημα το οποίο είναι στο  $80^\circ$  (καρέ του βίντεο) είναι εντός της ορισμένης υποπεριοχής.

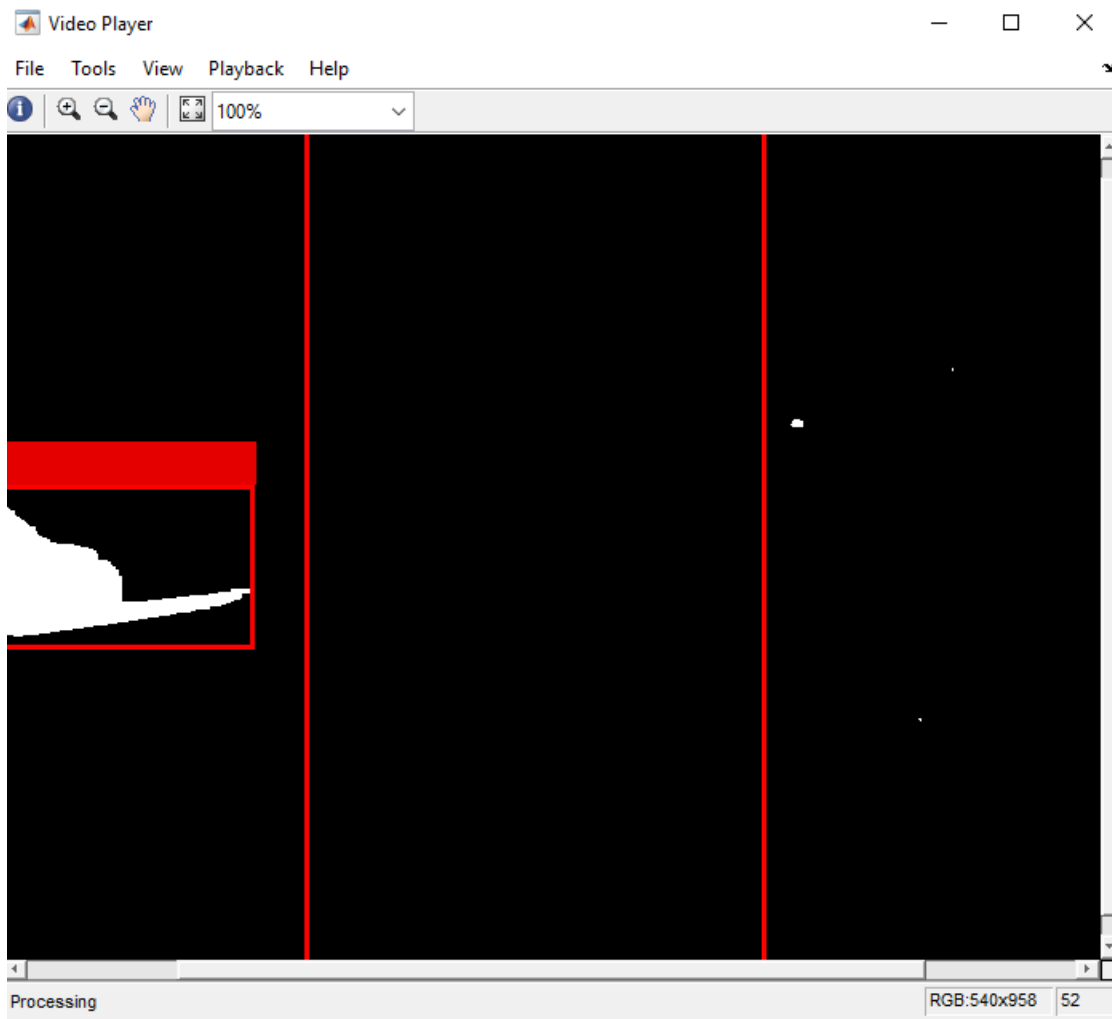
Στις εικόνες 26,27 παρουσιάζεται το όχημα το οποίο είναι στο  $52^\circ$  (καρέ του βίντεο) είναι εντός της ορισμένης υποπεριοχής.



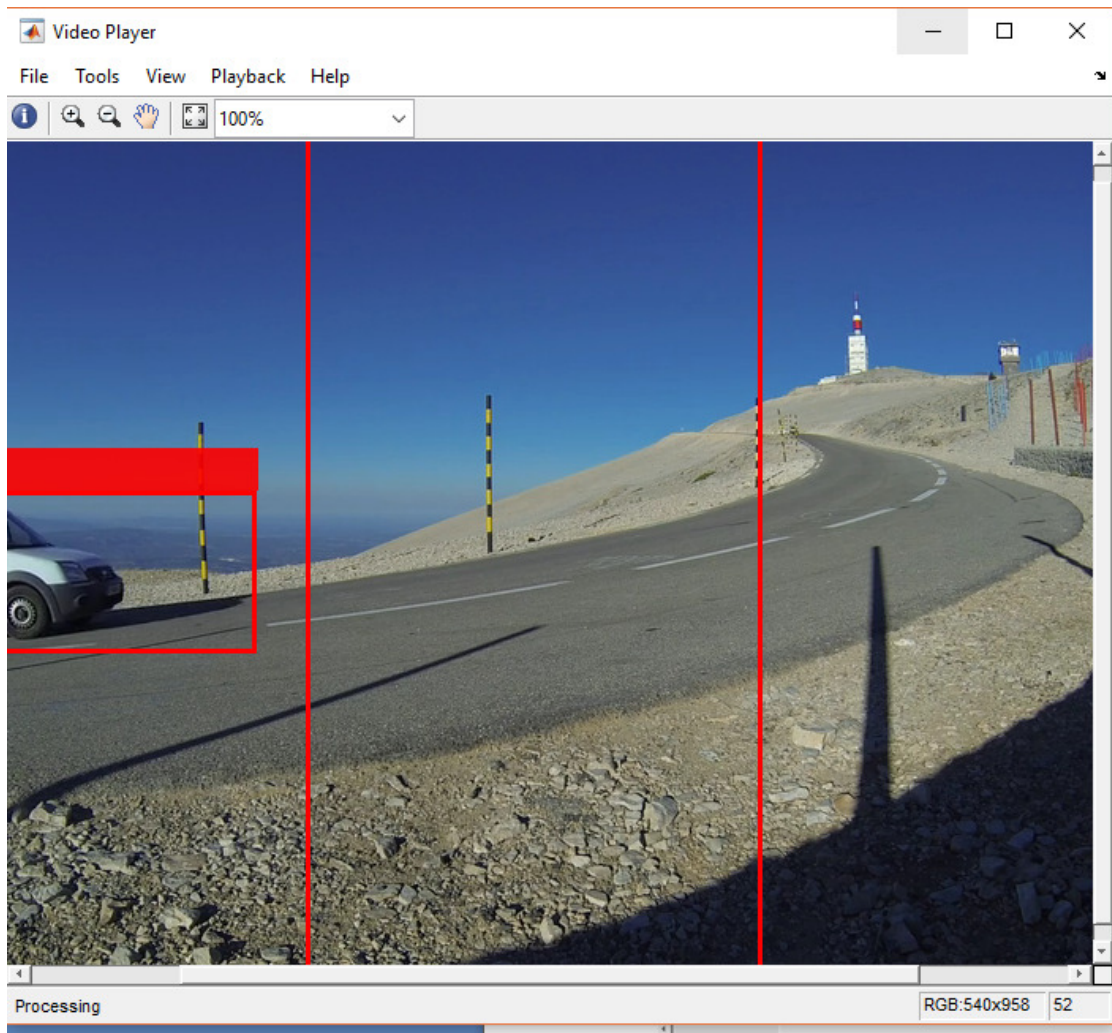
Εικόνα 27 - Όχημα εντός υποπεριοχής



Εικόνα 28 - Όχημα εντός υποπεριοχής



Εικόνα 29- Όχημα εκτός υποπεριοχής



Εικόνα 30 - Όχημα εκτός υποπεριοχής

## Βιβλιογραφία

- [1] J.K. Aggarwal, Q. Cai. Human Motion Analysis: A Review, *Computer Vision and Image understanding*, vol. 73, no. 3, pp. 428-440, March 1999.
- [2] Meghna Singh, “Robust Tracking and Human Activity Recognition”, Master of Science University of Alberta, 2004.
- [3] Yilmaz, A., Javed, O., and Shah, M. 2006. Object tracking: A survey. *ACM Comput. Surv.* 38, 4, Article 13 (Dec. 2006), 45 pages.
- [4] Ζουρνής-Καρούζος Νικόλαος, «Τεχνικές ανίχνευσης κίνησης σε προκαθορισμένες περιοχές της εικόνας σε πραγματικό χρόνο», Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ιούνιος 2007
- [5] Meha J. Patel, Bhumika Bhatt, 2015, A Comparative Study of Object Tracking Techniques, Vol. 4, Issue 3, March 2015, *International Journal of Innovative Research in Science, Engineering and Technology*
- [6] F. Aherne, N. Thacker, and P. Rockett, “The Bhattacharyya metric as an absolute similarity measure for frequency coded data,” *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.
- [7] Bodo Rosenhahn, Uwe G. Kersting, Lei He1, Andrew W. Smith Thomas Brox, Reinhard Klette, Hans-Peter Seidel, A Silhouette Based Human Motion Tracking System, *Communication and Information Technology Research Technical Report*
- [8] Kalman, R. (1960), ‘A new approach to linear filtering and prediction problems’, *Journal of Basic Engineering*, pp. 35–45.
- [9] Negenborn, R. (2003), ‘Robot Localization and Kalman Filters On finding your position in a noisy world’ Diploma Thesis, Utrecht University.
- [10] Mathworks.com. (2017). Motion-Based Multiple Object Tracking - MATLAB & Simulink Example. [online] Available at: <https://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html> [Accessed 3 Aug. 2017].



[11] Mathworks.com. (2017). Kalman Filter Design - MATLAB & Simulink Example. [online] Available at:  
<https://www.mathworks.com/help/control/examples/kalman-filter-design.html>  
[Accessed 3 Aug. 2017].

[12] How to Detect and Track Red, G. (2017). How to Detect and Track Red, Green and Blue Colored Object in LIVE Video - File Exchange - MATLAB Central. [online] Mathworks.com. Available at:  
<https://www.mathworks.com/matlabcentral/fileexchange/40154-how-to-detect-and-track-red--green-and-blue-colored-object-in-live-video> [Accessed 3 Aug. 2017].

# Παράρτημα

## Κώδικας

```
obj.reader = vision.VideoFileReader('cars.mp4');

obj.videoPlayer = vision.VideoPlayer('Position', [20, 400, 700,
400]);
obj.maskPlayer = vision.VideoPlayer('Position', [740, 400, 700,
400]);

obj.detector = vision.ForegroundDetector('NumGaussians', 3, ...
'NumTrainingFrames', 30, 'MinimumBackgroundRatio', 0.7);

obj.blobAnalyser = vision.BlobAnalysis('BoundingBoxOutputPort', true,
...
'AreaOutputPort', true, 'CentroidOutputPort', true, ...
'MinimumBlobArea', 3000);

tracks = struct(...
'id', {}, ...
'bbox', {}, ...
'kalmanFilter', {}, ...
'age', {}, ...
'totalVisibleCount', {}, ...
'consecutiveInvisibleCount', {});
nextId = 1; % ID of the next track

numOfFrames = 1;

while ~isDone(obj.reader)
    numOfFrames = numOfFrames + 1;
    frame = obj.reader.step();

    mask = obj.detector.step(frame);

    mask = imopen(mask, strel('disk',20,0));
    mask = imclose(mask, strel('disk', 40,0));
    %mask = imclose(mask, strel('ball', 80,40));
    %mask = imopen(mask, strel('square', 20));
    %mask = imclose(mask, strel('square', 40));

    %w = frame.width;
    [h w n] = size(frame);
    xmin = w - 0.66*w;
    xmax = w - 0.33*w;

    mask = imfill(mask, 'holes');
    [areas, centroids, bboxes] = obj.blobAnalyser.step(mask);

    for i = 1:length(tracks)
        bbox = tracks(i).bbox;
```

```

        predictedCentroid = predict(tracks(i).kalmanFilter);
        predictedCentroid = int32(predictedCentroid) - bbox(3:4)
/ 2;
        tracks(i).bbox = [predictedCentroid, bbox(3:4)];
    end

nTracks = length(tracks);
nDetections = size(centroids, 1);

    cost = zeros(nTracks, nDetections);
    for i = 1:nTracks
        cost(i, :) = distance(tracks(i).kalmanFilter, centroids);
    end
    costOfNonAssignment = 20;
    [assignments, unassignedTracks, unassignedDetections] = ...
        assignDetectionsToTracks(cost, costOfNonAssignment);

numAssignedTracks = size(assignments, 1);
    for i = 1:numAssignedTracks
        trackIdx = assignments(i, 1);
        detectionIdx = assignments(i, 2);
        centroid = centroids(detectionIdx, :);
        bbox = bboxes(detectionIdx, :);

        correct(tracks(trackIdx).kalmanFilter, centroid);

        tracks(trackIdx).bbox = bbox;

        tracks(trackIdx).age = tracks(trackIdx).age + 1;

        tracks(trackIdx).totalVisibleCount = ...
            tracks(trackIdx).totalVisibleCount + 1;
        tracks(trackIdx).consecutiveInvisibleCount = 0;
    end

    for i = 1:length(unassignedTracks)
        ind = unassignedTracks(i);
        tracks(ind).age = tracks(ind).age + 1;
        tracks(ind).consecutiveInvisibleCount = ...
            tracks(ind).consecutiveInvisibleCount + 1;
    end

    if isempty(tracks) == false
        invisibleForTooLong = 20;
        ageThreshold = 8;

        ages = [tracks(:).age];
        totalVisibleCounts = [tracks(:).totalVisibleCount];
        visibility = totalVisibleCounts ./ ages;

        lostInds = (ages < ageThreshold & visibility < 0.6) | ...
            [tracks(:).consecutiveInvisibleCount] >=
invisibleForTooLong;

        tracks = tracks(~lostInds);

```

```

end

centroids = centroids(unassignedDetections, :);
bboxes = bboxes(unassignedDetections, :);

for i = 1:size(centroids, 1)

    centroid = centroids(i, :);
    bbox = bboxes(i, :);

    % Create a Kalman filter object.
    kalmanFilter = configureKalmanFilter('ConstantVelocity',
...
        centroid, [250, 70], [150, 45], 95);
    %kalmanFilter =
configureKalmanFilter('ConstantAcceleration',...
        % centroid, [1 1 1]*1e5, [10, 10, 10], 10);

    newTrack = struct(...
        'id', nextId, ...
        'bbox', bbox, ...
        'kalmanFilter', kalmanFilter, ...
        'age', 1, ...
        'totalVisibleCount', 1, ...
        'consecutiveInvisibleCount', 0);

    tracks(end + 1) = newTrack;

    % detect color & detect diadromi

    nextId = nextId + 1;
    disp('new track object no');
    disp(nextId);
    %disp('numOfFrames');
    %disp(numOfFrames);
    pixels = impixel(frame, centroid(1), centroid(2));

end

frame = im2uint8(frame);
mask = uint8(repmat(mask, [1, 1, 3])) .* 255;

minVisibleCount = 8;
if ~isempty(tracks)

    reliableTrackInds = ...
        [tracks(:).totalVisibleCount] > minVisibleCount;
    reliableTracks = tracks(reliableTrackInds);

    if ~isempty(reliableTracks)
        bboxes = cat(1, reliableTracks.bbox);
    end
end

```

```

ids = int32([reliableTracks(:).id]);

labels = cellstr(int2str(ids'));
predictedTrackInds = ...
    [reliableTracks(:).consecutiveInvisibleCount] >
0;

isPredicted = cell(size(labels));
isPredicted(predictedTrackInds) = {' predicted'};
labels = strcat(labels, isPredicted);
bboxes(1:1)
xmin
xmax
if (bboxes(1:1) < xmin || bboxes(1:1) > xmax)
    frame = insertObjectAnnotation(frame,
'rectangle', ...
    bboxes, labels, 'LineWidth',3, 'Color', {'red'},
'TextBoxOpacity',0.9, 'FontSize',18);
    frame = insertObjectAnnotation(frame,
'rectangle', ...
    [xmin 0 0.3*w h],
labels, 'LineWidth',3, 'Color', {'red'},
'TextBoxOpacity',0.9, 'FontSize',18);
    mask = insertObjectAnnotation(mask, 'rectangle',
...
    bboxes, labels, 'LineWidth',3, 'Color', {'red'},
'TextBoxOpacity',0.9, 'FontSize',18);
    mask = insertObjectAnnotation(mask, 'rectangle',
...
    [xmin 0 0.3*w h],
labels, 'LineWidth',3, 'Color', {'red'},
'TextBoxOpacity',0.9, 'FontSize',18);
    disp('red');
else
    frame = insertObjectAnnotation(frame,
'rectangle', ...
    bboxes,
labels, 'LineWidth',3, 'Color', {'yellow'},
'TextBoxOpacity',0.9, 'FontSize',18);
    frame = insertObjectAnnotation(frame,
'rectangle', ...
    [xmin 0 0.3*w h],
labels, 'LineWidth',3, 'Color', {'yellow'},
'TextBoxOpacity',0.9, 'FontSize',18);
    mask = insertObjectAnnotation(mask, 'rectangle',
...
    bboxes,
labels, 'LineWidth',3, 'Color', {'yellow'},
'TextBoxOpacity',0.9, 'FontSize',18);
    mask = insertObjectAnnotation(mask, 'rectangle',
...
    [xmin 0 0.3*w h],
labels, 'LineWidth',3, 'Color', {'yellow'},
'TextBoxOpacity',0.9, 'FontSize',18);
    disp('yellow');
end

end
end

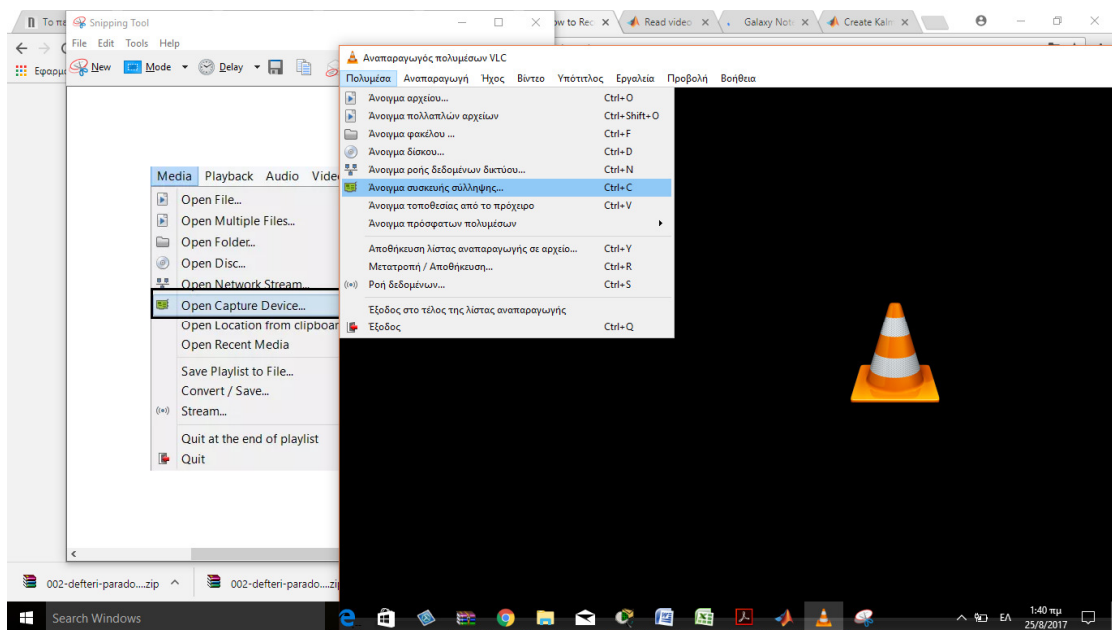
```

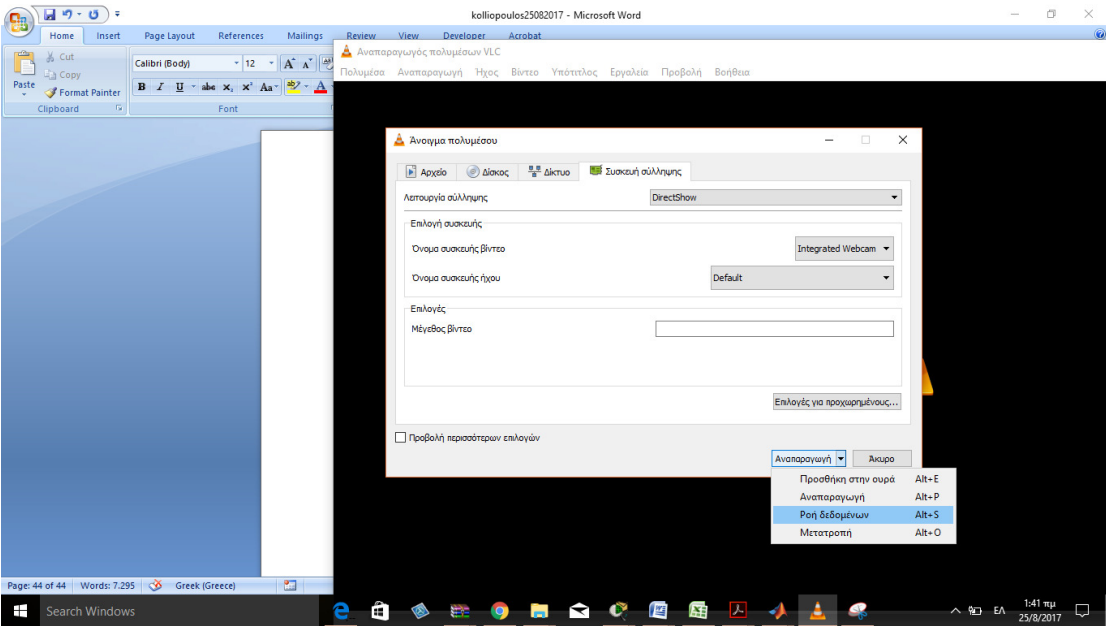
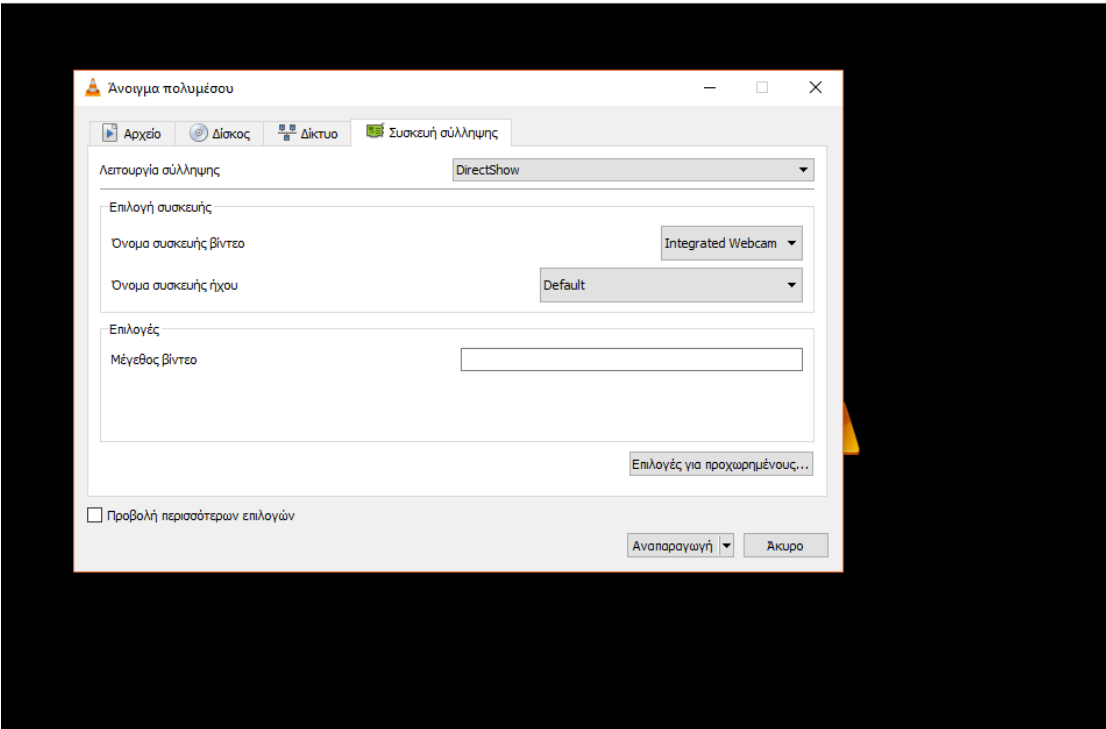
```
obj.maskPlayer.step(mask);  
obj.videoPlayer.step(frame);
```

end

## Σύνδεση WebCam με Matlab

Τα βήματα τα οποία είναι απαραίτητα παρουσιάζονται στις παρακάτω εικόνες





Εξοδος ροής δεδομένων

**Πηγή**  
Εγκατάσταση πηγών πολυμέσων στη ροή δεδομένων

Αυτός ο οδηγός θα σας επιτρέψει να διοχετεύσετε ροές δεδομένων ή να μετατρέψετε τα πολυμέσα σας για τοπική χρήση, στο ιδιωτικό σας δίκτυο ή στο διαδίκτυο. | Θα πρέπει να ξεκινήσετε ελέγχοντας ότι η πηγή ταιριάζει με αυτό που θέλετε να είναι η είσοδός σας και μετά να πατήσετε το κουμπί "Επόμενο" για να συνεχίσετε.

Πηγή:

Τύπος: dshow

Πίσω **Επόμενο** Ακύρο

Εξοδος ροής δεδομένων

**Ρύθμιση προορισμού**  
Επιλέξτε προορισμούς για τη διοχέτευση ροών δεδομένων

+ Αρχείο x

Αυτή η ενότητα γράφει την επανακωδικοποιημένη ροή δεδομένων σε αρχείο.

Όνομα αρχείου  Περίληψη...

Πίσω **Επόμενο** Ακύρο



