

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ



Σύστημα Τηλεχειρισμού και Επικοινωνίας
Αυτοκινούμενων Ρομπότ

ΘΕΟΔΩΡΟΠΟΥΛΟΣ ΘΕΟΦΑΝΗΣ - ΜΕ/0460

Εξάμηνο: Δ΄

Επιβλέπων: Καθ. Σγούρος Ν.

Περιεχόμενα

1	Περίληψη	3
2	Εισαγωγή.....	4
2.1	Εργαλεία Υλοποίησης	5
3	Το Πρόβλημα που καλούμαστε να επιλύσουμε	6
3.1	Σύντομο Ιστορικό.....	6
3.2	Τα προβλήματα στην διαδικασία του Τηλεχειρισμού	7
3.3	Προβλήματα στον τηλεχειρισμό αυτοκινούμενων ρομπότ και μέσα Επίλυσης τους 8	
3.4	Σημαντικές εφαρμογές πλοήγησης αυτοκινούμενων ρομπότ	16
3.4.1	Το πρόγραμμα Pathfinder της NASA.....	16
3.4.2	Το πρόγραμμα AGV στο Πανεπιστήμιο της Οχφόρδης.....	18
3.4.3	MIT: Νοημοσύνη χωρίς Απεικόνιση.....	19
3.4.4	Το πρόγραμμα NavLab του πανεπιστημίου Carnegie Mellon.....	20
4	Παρουσίαση και τεχνικά Χαρακτηριστικά του AmigoBot.....	23
4.1	Φυσικά Χαρακτηριστικά και Υποσυστήματα	23
4.2	Λογισμικό Ελέγχου και Αρχιτεκτονική του AmigoBot	25
4.3	Τύποι Επικοινωνίας του AmigoBot με τον Η/Υ	26
4.4	Σύστημα Σόναρ	27
4.5	Σύστημα Κίνησης του AmigoBot	28
5	Το προτεινόμενο Σύστημα Τηλεχειρισμού Ρομπότ μέσω Δικτυακής Κάμερας.....	29
5.1	Στόχος του προτεινόμενου Συστήματος.....	29
5.2	Η Μεθοδολογία που ακολουθήθηκε	30
5.3	Μέθοδος Εντοπισμού του AmigoBot και Επεξεργασία Εικόνας	31
5.3.1	Η τεχνική Επεξεργασίας Εικόνας που ακολουθήθηκε	33
5.3.2	Μέθοδος εντοπισμού του AmigoBot.....	35
5.3.3	Χάραξη Τροχιάς για το AmigoBot και Αναγωγή των αποστάσεων σε Διαστάσεις του Πραγματικού Χώρου	39
5.4	Υλοποίηση του προτεινόμενου Συστήματος	43
5.4.1	Το περιβάλλον προγραμματιστικών Διεπαφών ARIA	44
5.4.2	Αρχιτεκτονική του Συστήματος και κλάσεις που χρησιμοποιήθηκαν από το API της ARIA.....	45
6	Περιορισμοί και Προτεινόμενες Βελτιώσεις	48
7	Συμπεράσματα	49
8	Αναφορές.....	50

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον κύριο Νικήτα Σγούρο, καθηγητή του Πανεπιστημίου Πειραιώς με την καθοδήγηση του οποίου κατέστη δυνατή η ολοκλήρωση της συγκεκριμένης εργασίας. Η συμβολή του υπήρξε καθοριστική σε όλα τα στάδια υλοποίησης, από την επιλογή του υλικού και την ανάλυση μέχρι και τον τρόπο παρουσίασης. Θα ήθελα να τον ευχαριστήσω ιδιαίτερα για τις εύστοχες και παραγωγικές επισημάνσεις του και για την επιμονή του στην τελειότητα και στην πληρότητα. Τέλος δεν μπορώ να παραβλέψω την υπομονή που έδειξε από την αρχή μέχρι το τέλος της συνεργασίας μας καθώς για μένα η ενασχόληση με ένα αντικείμενο τόσο δύσκολο και πρωτοποριακό υπήρξε πρωτόγνωρη εμπειρία.

1 Περίληψη

Η ανάπτυξη εξελιγμένων διεπαφών χρήστη (user interface) για τον τηλεχειρισμό ρομπότ έχει αρχίσει να γίνεται στις μέρες μας επιτακτική ανάγκη [1]. Βέβαια για ορισμένου τύπου εφαρμογές, ο τηλεχειρισμός (ο απομακρυσμένος έλεγχος της συμπεριφοράς ενός ρομπότ από κάποιον άνθρωπο) αποτελεί προς το παρόν μια συμβιβαστική λύση έως ότου βελτιωθούν οι αυτόνομες ικανότητες των ρομπωτικών μηχανισμών. Ένα από τα σημαντικότερα όμως πεδία χρησιμοποίησης των ρομπότ είναι η εξερεύνηση χώρων όπου η αλληλεπίδραση τους με τον άνθρωπο παίζει πρωτεύοντα ρόλο. Γι' αυτό είναι κρίσιμο στις μέρες μας το να μάθουμε να σχεδιάζουμε καλύτερες διεπαφές χρήστη ώστε να καταλήξουμε σε αληθινά ολοκληρωμένα και αποτελεσματικά συστήματα ανθρώπου – ρομπότ.

Στην παρούσα εργασία αναπτύσσουμε ένα εξελιγμένο σύστημα τηλεχειρισμού ρομπότ μέσω δικτυακής κάμερας, το οποίο βασίζεται σε μία ολοκληρωμένη διεπαφή χρήστη και παράλληλα προτείνουμε λύσεις για μερικά από τα σημαντικότερα προβλήματα που παρουσιάζει ένα τέτοιο εγχείρημα, όπως είναι ο εντοπισμός του ρομπότ στον χώρο και η ρεαλιστική πλοήγηση του με βάση τις πληροφορίες που λαμβάνουμε από την κάμερα.

2 Εισαγωγή

Κατά καιρούς έχουν δοθεί διάφοροι ορισμοί για την λέξη «ρομπότ» άλλοι λιγότερο και άλλοι περισσότερο επιτυχημένοι. Σύμφωνα με το Αμερικανικό Ινστιτούτο Ρομποτικής (RIA – Robotic Institute of Robotics) «ρομπότ είναι ένας επαναπρογραμματιζόμενος, πολυλειτουργικός χειριστής σχεδιασμένος να μεταφέρει υλικά, αντικείμενα, εργαλεία ή ειδικές συσκευές μέσω μεταβλητών πολυπρογραμματιζόμενων κινήσεων, προκειμένου να εκτελέσει διάφορα έργα». Μάλιστα η ίδια ή λέξη «ρομπότ» προέρχεται από την τσέχικη λέξη «ρομπότα – robota» που σημαίνει καταναγκαστική εργασία [2].

Ο παραπάνω ορισμός καλύπτει ένα ευρύτατο φάσμα ρομποτικών χειριστών μέσα στο οποίο μπορούν να συμπεριληφθούν διάφοροι τύποι ρομπότ που εξυπηρετούν εφαρμογές όπως:

- Αυτόματη ή Ευέλικτη Κατασκευή
- Απομακρυσμένη Εξερεύνηση
- Προσθετική Και Βιοϊατρική
- Διαχείριση Επικίνδυνων Υλικών
- Άλλες Υπηρεσίες

Στην παρούσα εργασία μας απασχολεί ο τομέας της απομακρυσμένης εξερεύνησης, δηλαδή η τάξη των ρομποτικών χειριστών που έχουν σχεδιαστεί έτσι ώστε να εξερευνούν απομακρυσμένους χώρους και μάλιστα συχνά σε συνθήκες που δεν μπορεί να ανεχθεί ο ανθρώπινος οργανισμός. Εξελιγμένα ρομπότ αυτής της κατηγορίας χρησιμοποιούνται στις μέρες μας για την εξερεύνηση σε πλανήτες του ηλιακού μας συστήματος ή στα βάθη των ωκεανών.

Μέσα από την μελέτη της σχετικής βιβλιογραφίας [1] φάνηκε ξεκάθαρα ότι ο τηλεχειρισμός ενός αυτοκινούμενου ρομπότ συχνά μπορεί να γίνει προβληματικός ειδικότερα για αρχάριους ή ανεκπαιδευτους χειριστές. Τα σημαντικότερα **προβλήματα** ενός τέτοιου εγχειρήματος έχουν να κάνουν με:

- Αδυναμία εντοπισμού του ρομπότ στον χώρο
- Ανακριβή δεδομένα για την θέση του ρομπότ
- Δυσχέρειες στην χάραξη της τροχιάς κίνησης του
- Αποτυχία της διαδικασίας ανίχνευσης εμποδίων

Ακόμα και αν το όχημα διαθέτει ενσωματωμένες ικανότητες αυτοελέγχου (π.χ. διατήρησης πορείας – route following) και ελέγχεται από εξειδικευμένο χειριστή, αστάθμητοι παράγοντες όπως προβλήματα επικοινωνίας ή υπερφορτωμένο δίκτυο, μπορούν να υπονομεύσουν τα επιθυμητά αποτελέσματα.

Για την αντιμετώπιση των παραπάνω προβλημάτων και προκειμένου ο τηλεχειρισμός ενός ρομπότ να γίνει αποτελεσματικότερος και παραγωγικότερος ώστε να ικανοποιήσει πραγματικές ανάγκες της ζωής του ανθρώπου, χρειαζόμαστε πλατφόρμες χειρισμού του ρομπότ που θα μας βοηθούν να αντιληφθούμε πληρέστερα το περιβάλλον στο οποίο κινείται, να κάνουμε ακριβείς εκτιμήσεις και να λάβουμε τις σωστές αποφάσεις για τον έλεγχο του.

Σκοπός της παρούσας μελέτης είναι να αναπτύξουμε ένα σύστημα τηλεχειρισμού ρομπότ μέσω δικτυακής κάμερας, το οποίο βασίζεται σε μία ολοκληρωμένη διεπαφή χρήστη και παράλληλα να προτείνουμε λύσεις για μερικά από τα σημαντικότερα προβλήματα που παρουσιάζει ένα τέτοιο εγχείρημα, όπως είναι ο εντοπισμός του ρομπότ στον χώρο και η ρεαλιστική πλοήγηση του με βάση τις πληροφορίες που λαμβάνουμε από την κάμερα.

2.1 Εργαλεία Υλοποίησης

Για την εκπόνηση της συγκεκριμένης εργασίας χρησιμοποιήθηκαν τα παρακάτω εργαλεία Λογισμικού (Software) και Υλικού (Hardware) . Τα επιλέξαμε με γνώμονα την εμπειρία που έχουμε αποκτήσει όσον αφορά την χρήση τους, καθώς επίσης και το γεγονός της ευρείας αποδοχής που έχουν. Όσον αφορά την ρομπωτική πλατφόρμα (AmigoBot) πάνω στην οποία εργαστήκαμε, θεωρούμε ότι είναι αναγκαία μια εκτενέστερη περιγραφή των χαρακτηριστικών της, την οποία και παραθέτουμε στο κεφάλαιο 4.

Σχεδίαση	Εργαλείο
Σχεδίαση Αρχιτεκτονικής	Microsoft Visio Professional 2000
Περιβάλλον Ανάπτυξης	Εργαλείο
Λειτουργικό Σύστημα	Windows XP Pro SP2
Ανάπτυξη	Borland JBuilder 2005 Enterprise
Java	JDK 1.5.0_01
Media Framework	JMF 2.1.1e
Υλικό	Εργαλείο
Ρομπωτική Πλατφόρμα	Active Media Robotics - AmigoBot
Camera	Logitech WebCam 2.0 MP
Wireless Router	Cisco Aironet 1100

Πίνακας 2.1: Τα εργαλεία Λογισμικού που χρησιμοποιήθηκαν

Ακολουθεί το κύριο μέρος της εργασίας ξεκινώντας από την περιγραφή του προβλήματος για το οποίο καλούμαστε να προτείνουμε λύσεις και την παράθεση των σημαντικότερων προσπαθειών που έχουν γίνει προς αυτή την κατεύθυνση και παρουσιάζονται στην διεθνή βιβλιογραφία.

Στην συνέχεια εξηγούμε την προσέγγιση που ακολουθήσαμε στην συγκεκριμένη μελέτη και εισχωρούμε σε λεπτομέρειες όπως η αρχιτεκτονική του προτεινόμενου συστήματος και τα τεχνικά χαρακτηριστικά του ρομπότ μας (AmigoBot). Στο σημείο αυτό και όπου κρίνουμε απαραίτητο θα παραθέσουμε και κάποια από τα σημαντικότερα κομμάτια κώδικα που υλοποιήσαμε ώστε να βοηθήσουμε τον αναγνώστη να κατανοήσει πληρέστερα την προτεινόμενη αρχιτεκτονική.

3 Το Πρόβλημα που καλούμαστε να επιλύσουμε

Ο τηλεχειρισμός μηχανών υπήρξε πάντα ένα πεδίο έρευνας για εξειδικευμένο προσωπικό. Ο προσδιορισμός της θέσης του οχήματος, της πορείας που πρέπει να ακολουθήσει και τελικά η απομακρυσμένη καθοδήγηση του αποτελούν περίπλοκα ζητήματα. Οι δυσκολίες εντείνονται ακόμα περισσότερο όταν έχουμε να κάνουμε με αντίζσα περιβάλλοντα, με δυσλειτουργικές επικοινωνίες ή ανεπαρκείς πόρους.

3.1 Σύντομο Ιστορικό

Οι μηχανισμοί τηλεχειρισμού χρησιμοποιούνται για την υποστήριξη των ρομπότ όταν αυτά δεν μπορούν από μόνα τους να υπερβούν τις δυσκολίες που παρουσιάζει μία κατάσταση ή μία δραστηριότητα που οφείλουν να εκτελέσουν. Συνεπώς είναι απαραίτητος ο σχεδιασμός και η κατασκευή ρομποτικών μηχανισμών που θα πραγματώνουν την συνέργια ανθρώπου - μηχανής [3]. Αυτός ο συνδυασμός είναι καθοριστικής σημασίας και αποτελεί την κύρια διαφορά με τον συνήθη σχεδιασμό ρομπότ.

Οι πρώτοι μηχανισμοί τηλεχειρισμού δημιουργήθηκαν μετά τον δεύτερο παγκόσμιο πόλεμο για τις ανάγκες της πυρηνικής έρευνας. Όλοι τους χρησιμοποιούσαν την ιδέα του Καθοδηγητή – Εξαρτημένου (Master – Slave). Αποτελούνταν από δύο συμμετρικούς βραχίονες. Ο πρωτεύοντας ελεγχόταν από τον χειριστή ενώ ο εξαρτημένος αντέγραφε τις κινήσεις του χειριστή στον χώρο όπου η εργασία έπρεπε στην πραγματικότητα να υλοποιηθεί. Στα πρώτα χρόνια, οι δύο βραχίονες συνδέονταν μηχανικά μεταξύ τους. Στην συνέχεια και με την βοήθεια του ηλεκτρισμού δόθηκε η δυνατότητα για χειρισμό από οποιαδήποτε απόσταση. Κατά την δεκαετία του 80' οι ηλεκτρονικοί υπολογιστές χρησιμοποιήθηκαν σαν συστήματα ελέγχου ανοίγοντας τον δρόμο για τα Υποβοηθούμενα από Υπολογιστή Συστήματα Τηλεχειρισμού (Computer Aided Teleoperation - CAT). Στις μέρες μας τα συστήματα αυτά έχουν σε μεγάλο βαθμό ισχυροποιηθεί από τεχνικές Εικονικής Πραγματικότητας.

Στα πρώιμα συστήματα, η απουσία ανεπτυγμένων ηλεκτρονικών εξαρτημάτων και ακόμα περισσότερο η απουσία υπολογιστών, καθιστούσαν υποχρεωτικό έναν συμμετρικό μηχανισμό για την ορθή μεταφορά των κινήσεων του χειριστή στην εξαρτημένη συσκευή. Με την είσοδο των υπολογιστών η εξαρτημένη συσκευή διατήρησε την μηχανική δομή της αλλά πλέον ο πρωτεύοντας χειριστής μπορούσε να μικρύνει σε μέγεθος και να μετασχηματιστεί σε ένα σύνολο από τηλεχειριστήρια εφοδιασμένα με αισθητήρες ή σε ένα σύνολο από ευέλικτα υποσυστήματα.

Παρόλο όμως που οι νέες αυτές διεπαφές ταίριαζαν καλύτερα στον άνθρωπο δεν μείωσαν στον αναμενόμενο βαθμό τις γενικότερες δυσκολίες ενός τέτοιου εγχειρήματος. Επιπροσθέτως, κατά τον απομακρυσμένο έλεγχο, ήταν απαραίτητη η αξιόπιστη παρακολούθηση του περιβάλλοντος δράσης της εξαρτημένης συσκευής. Οι συμβατικές κάμερες ακόμα και οι στερεοσκοπικές φαίνεται ότι απέτυχαν να δώσουν επαρκή πληροφόρηση στον χειριστή.

3.2 Τα προβλήματα στην διαδικασία του Τηλεχειρισμού

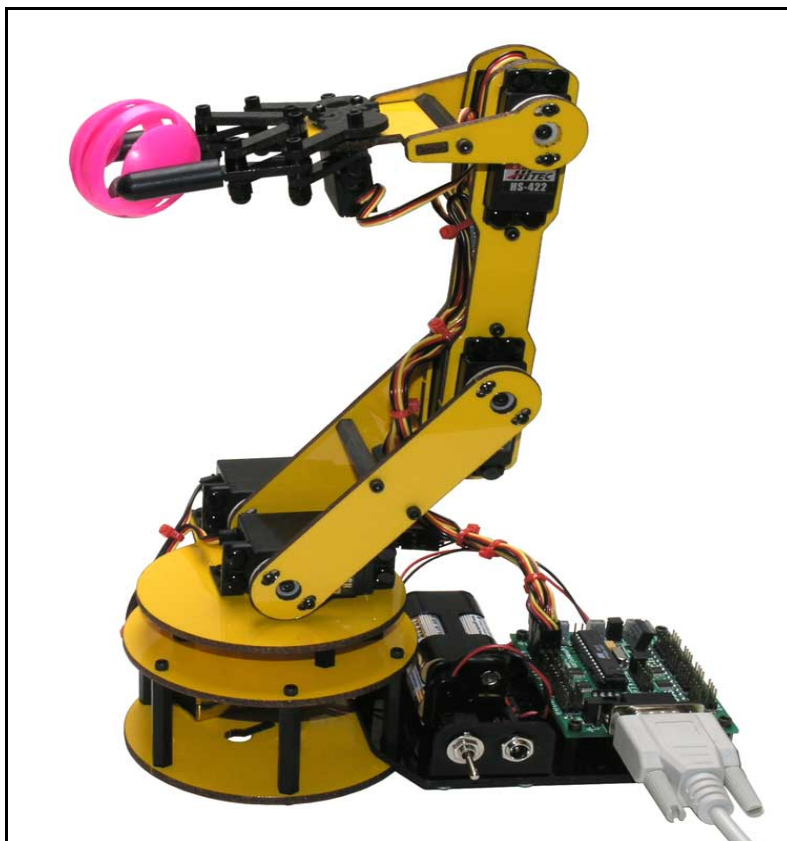
Η διαδικασία του τηλεχειρισμού ρομπότ έχει αναπτυχθεί στις μέρες μας σε σημαντικό βαθμό τόσο από τεχνολογική όσο και από θεωρητική άποψη. Παρολαυτά υπάρχουν προβλήματα για τα οποία δεν έχει δοθεί ακόμη μια ικανοποιητική ή πειστική λύση. Στην συνέχεια θα προσπαθήσουμε να συνοψίσουμε τα σημαντικότερα από αυτά.

Πρώτα απ' όλα δεν έχουμε καταφέρει ακόμα να προσεγγίσουμε ικανοποιητικά αυτό που ονομάζεται **διαφάνεια (transparency)** στα συστήματα τηλεχειρισμού [3]. Η διαφάνεια μπορεί να θεωρηθεί σαν ένα κριτήριο μέτρησης της αίσθησης που επιστρέφεται σε έναν χειριστή αναφορικά με κάποια δράση του πάνω στον ελεγχόμενο μηχανισμό [4]. Η διαφάνεια είναι μία έννοια που θα πρέπει να ισορροπεί με την σταθερότητα του συστήματος και έχει γίνει πλέον κατανοητό, ότι δεν μπορεί να επιτευχθεί στον μέγιστο βαθμό από συμβατικά συστήματα ισότιμου ελέγχου ανάμεσα στον χειριστή και στον εξαρτώμενο μηχανισμό. Θα πρέπει να επαναπροσδιορίσουμε την έννοια της διαφάνειας με βάση διαφορετικά κριτήρια. Η πηγή του προβλήματος εντοπίζεται στο ότι κάθε ενέργεια του χειριστή και η αντίστοιχη ανάδραση (feedback) που συνεπάγεται αυτή, είναι αναγκασμένες να διέρχονται μέσα από κανάλια επικοινωνίας τα οποία χαρακτηρίζονται από συνισταμένες τις οποίες δεν είναι δυνατόν να παρακάμψουμε χωρίς απώλειες στην σταθερότητα ή στην ασφαλή λειτουργία του συστήματος.

Η **χρονική καθυστέρηση (time delay)** κατά τον έλεγχο απομακρυσμένων συστημάτων. Η καθυστέρηση στην επικοινωνία ανάμεσα στον χειριστή και στον ελεγχόμενο μηχανισμό αποτελεί ίσως το κρισιμότερο πρόβλημα στην όλη διαδικασία του τηλεχειρισμού. Δεν επηρεάζει μόνο την διαφάνεια, που περιγράψαμε προηγουμένως αλλά και την σταθερότητα του συστήματος. Πολλές προσπάθειες έχουν επικεντρωθεί στην επίλυση αυτού του προβλήματος, όμως στην παρούσα φάση παραμένει ουτοπική η άμεση απόκριση από το απομακρυσμένο περιβάλλον εφόσον εξακολουθούμε να αντιμετωπίζουμε καθυστερήσεις της τάξεως δευτερολέπτων.

Ο **ανθρώπινος παράγοντας**. Η ευελιξία και η επιτυχία ενός συστήματος τηλεχειρισμού εξαρτάται σε μεγάλο βαθμό από τον βαθμό προσαρμογής του χειριστή – ανθρώπου στο σύστημα. Θα πρέπει δηλαδή ο χειριστής να είναι ένα εκπαιδευμένο και εξειδικευμένο άτομο προκειμένου να φέρει σε πέρας μια δραστηριότητα. **Το περιβάλλον – διεπαφή ελέγχου που έχουμε υλοποιήσει** σκοπεύει στο να εκμηδενίσει αυτές τις παραμέτρους παρέχοντας μία ολοκληρωμένη διεπαφή χρήστη που «κρύβει» τεχνικές λεπτομέρειες και απευθύνεται έτσι και σε μη εξειδικευμένους χειριστές.

Τα προβλήματα που περιγράψαμε μέχρι τώρα αφορούν στο σύνολο τους τα συστήματα τηλεχειρισμού μηχανισμών (π.χ. μηχανικοί βραχίονες εικόνα. 3.1, κινούμενα ρομπότ κ.τλ.). Στην συνέχεια θα επικεντρώσουμε την προσοχή μας στα ιδιαίτερα ζητήματα, που παρουσιάζει ο τομέας της απομακρυσμένης εξερεύνησης χώρων που αποτελεί και το θέμα της μελέτης μας. Παράλληλα θα παραθέσουμε παραδείγματα καθώς και τις σημαντικότερες έρευνες που έχουν γίνει στον τομέα αυτό.



Εικόνα 3.1: Μηχανικός Βραχίονας

3.3 Προβλήματα στον τηλεχειρισμό αυτοκινούμενων ρομπότ και μέσα Επίλυσης τους

Ίσως το σημαντικότερο συμπέρασμα που εξαγάμε από την μελέτη της σχετικής βιβλιογραφίας είναι ότι μέχρι σήμερα δεν έχει βρεθεί μία απόλυτη και βέλτιστη λύση όσον αφορά τον τηλεχειρισμό αυτοκινούμενων ρομπότ [5]. Η ακριβής γνώση της θέσης του οχήματος (mobile robot positioning) που χρησιμοποιείται για την εξερεύνηση απομακρυσμένων χώρων αποτελεί θεμελιώδες πρόβλημα στις εφαρμογές αυτοκινούμενων ρομπότ. Αναζητώντας λύσεις σε αυτό το ζήτημα οι ερευνητές και οι μηχανικοί έχουν αναπτύξει ποικίλα συστήματα, αισθητήρες και τεχνικές. Μπορούμε να διακρίνουμε τα συστήματα αυτά σε δύο κύριες κατηγορίες οι οποίες διασπώνται ακόμα περισσότερο ώστε να καταλήξουμε τελικά σε επτά κατηγορίες. Η κατηγοριοποίηση έχει ως εξής:

- A. **Συστήματα Σχετικής Καταγραφής Θέσης (Relative Position Measurements)**
 - I. Οδομετρία
 - II. Αδρανής Πλοήγηση
- B. **Συστήματα Απόλυτης Καταγραφής Θέσης (Absolute Position Measurements)**
 - III. Μαγνητικές Πυξίδες
 - IV. Ενεργοί Σηματοδότες
 - V. GPS (Global Positioning Systems)
 - VI. Πλοήγηση βασισμένη σε Ορόσημα
 - VII. Παρακολούθηση βασισμένη σε Χάρτες - Μοντέλα

Εξαιτίας της απουσίας μίας και μοναδικής αξιόπιστης λύσης, οι ερευνητές είναι αναγκασμένοι τις περισσότερες φορές να συνδυάζουν τις επιμέρους λύσεις συνήθως χρησιμοποιώντας μία τεχνική από κάθε μία από τις δύο κύριες κατηγορίες.

Στην συνέχεια θα προσπαθήσουμε να δώσουμε τις βασικές αρχές που χαρακτηρίζουν κάθε μία από τις υπάρχουσες λύσεις. Πρέπει να τονίσουμε στο σημείο αυτό ότι στο πεδίο έρευνας με το οποίο έχουμε καταπιαστεί υπάρχει ιδιαίτερη δραστηριότητα με αποτέλεσμα να αναπτύσσονται διαρκώς νέα πρωτοποριακά συστήματα και ιδέες.

Οδομετρία

Η οδομετρία είναι η πιο ευρέως χρησιμοποιούμενη μέθοδος πλοήγησης. Παρέχει αρκετά καλή ακρίβεια, δεν απαιτεί ακριβούς πόρους και επιτρέπει την λήψη δειγμάτων με πολύ μεγάλη συχνότητα. Παρολαυτά η βασική ιδέα στην οποία στηρίζεται η οδομετρία είναι η ερμηνεία της πληροφορίας που συσσωρεύεται από την κίνηση του ρομπότ. Αυτό αναπόφευκτα δημιουργεί συσσώρευση λαθών ιδιαίτερα όσον αφορά τον προσανατολισμό και τελικά οδηγούμαστε σε όλα και πιο λανθασμένη εκτίμηση της θέσης του ρομπότ, ιδιαίτερα όσο μεγαλύτερη είναι η απόσταση την οποία έχει διανύσει. Εντούτοις, οι περισσότεροι ερευνητές συμφωνούν στο ότι η οδομετρία πρέπει να αποτελεί βασικό κομμάτι ενός συστήματος πλοήγησης ρομπότ και μπορεί να συμβάλλει στην απλοποίηση της όλης διαδικασίας όσο θα βελτιώνονται και οι μηχανισμοί μέτρησης της.

Στο σύστημα που προτείνουμε χρησιμοποιούμε την οδομετρία σαν μέσον εντοπισμού της θέσης και της ταχύτητας του ρομπότ μας αλλά και της απόστασης που αυτό διανύει. Αυτό κατέστη δυνατό εξαιτίας και της ακρίβειας που παρέχει η πλατφόρμα μας (AmigoBot) με 123 στιγμές – ticks/χιλιοστόμετρο.

Αδρανής Πλοήγηση

Η αδρανής πλοήγηση χρησιμοποιεί γυροσκόπια και επιταχυνσιόμετρα για την καταμέτρηση αντίστοιχα του ρυθμού περιστροφής και της επιτάχυνσης. Οι μετρήσεις αυτές στην συνέχεια συνδυάζονται για να προσδιοριστεί τελικά η θέση του ρομπότ. Η τεχνική αυτή έχει το πλεονέκτημα ότι εμπειριέχει μεγάλο βαθμό αυτονομίας και δεν απαιτεί εξωτερικά σημεία αναφοράς. Παρολαυτά τα δεδομένα που μεταδίδουν οι αδρανείς αισθητήρες έχουν την τάση να γίνονται όλο και πιο ανακριβή με την πάροδο του χρόνου, κυρίως εξαιτίας της διαδικασίας συνδυασμού τους. Έτσι οι αδρανείς αισθητήρες κρίνονται ακατάλληλα μέσα προσδιορισμού της θέσης του ρομπότ για εκτεταμένα χρονικά διαστήματα.

Πιο συγκεκριμένα όσον αφορά τα **επιταχυνσιόμετρα**, τα αποτελέσματα των δοκιμών τους κατά την πλοήγηση αυτοκινούμενων ρομπότ υπήρξαν αποθαρρυντικά. Σε μία ανεπίσημη μελέτη που πραγματοποιήθηκε στο πανεπιστήμιο του Μίσιγκαν παρατηρήθηκε πολύ χαμηλή σχέση σήματος προς θόρυβο στις χαμηλές επιταχύνσεις. Επιπλέον τα επιταχυνσιόμετρα είναι ιδιαίτερα ευαίσθητα σε ανομοιογενή εδάφη επειδή κάθε παρέκκλιση από μία τέλεια οριζόντια θέση οδηγεί τον αισθητήρα σε ανίχνευση της επιτάχυνσης g της βαρύτητας.

Ένα σύστημα χαμηλού κόστους που στοχεύει στο να εξαλείψει το τελευταίο αυτό πρόβλημα έχει προταθεί από τους Barshan και Durrant [6] [Barshan and Durrant-Whyte, 1993; 1995]. Το σύστημα αυτό περιλαμβάνει έναν αισθητήρα κλίσης. Και πάλι όμως το ποσοστό λάθους κρίθηκε ως μη αποδεκτό για τις περισσότερες από τις εφαρμογές αυτοκινούμενων ρομπότ.

Τα **γυροσκόπια** από την άλλη είναι ιδιαίτερα σημαντικά για την διαδικασία προσδιορισμού της θέσης του ρομπότ αφού μπορούν να δώσουν λύσεις στα σημαντικότερα προβλήματα της οδομετρίας. Κι αυτό διότι σε κάθε τεχνική προσδιορισμού θέσης βασισμένη σε οδομετρία, οποιοδήποτε μικρό στιγμιαίο σφάλμα στον προσανατολισμό οδηγεί σε σημαντικότερο σφάλμα καθώς εξελίσσεται η όλη διαδικασία. Έτσι είναι απαραίτητος ο έγκαιρος εντοπισμός και η εξάλειψη των σφαλμάτων προσανατολισμού.

Μέχρι πριν από μερικά χρόνια η χρήση γυροσκοπίων μεγάλης ακρίβειας στις ρομποτικές εφαρμογές ήταν ασύμφορη από άποψη κόστους. Η πρόσφατη ανακάλυψη όμως των γυροσκοπίων οπτικών ινών (γνωστά και ως λείζερ – γυροσκόπια) έχουν μειώσει το κόστος δραματικά και αποτελούν πλέον δημοφιλέστατη λύση για τις ρομποτικές εφαρμογές εξερεύνησης χώρων.

Μαγνητικές Πυξίδες

Η κατεύθυνση του οχήματος έχει κρίσιμη σημασία για τις παραμέτρους πλοήγησης (x,y,θ) υπό την έννοια της επιρροής που παρουσιάζει στα συσσωρευμένα λάθη παρέκκλισης. Γι' αυτό, οι αισθητήρες που μπορούν να καταγράψουν με ακριβή και απόλυτο τρόπο την κατεύθυνση του οχήματος είναι ζωτικής σημασίας για την επιτυχία της πλοήγησης αυτόνομων συστημάτων. Η μαγνητική πυξίδα είναι ένας τέτοιος αισθητήρας.

Εντούτοις ένα από τα **μειονεκτήματα** της είναι ότι βασίζεται στο μαγνητικό πεδίο της Γης το οποίο επηρεάζεται σημαντικά κοντά σε πεδία ενέργειας ή μεταλλικές κατασκευές [7]. Έτσι η χρήση γεωμαγνητικών αισθητήρων σε εσωτερικούς χώρους είναι αναποτελεσματική.

Με βάση τα διαφορετικά είδη φυσικής επίδρασης στο μαγνητικό πεδίο της Γης έχουν αναπτυχθεί διάφορα είδη αισθητήρων τέτοιου τύπου:

- Μηχανικές Μαγνητικές Πυξίδες
- Ηλεκτρομαγνητικές Πυξίδες
- Μαγνητοελαστικές Πυξίδες
- Πυξίδες αντίστασης μαγνητικού πεδίου κ.α.

Η πυξίδα που φαίνεται να προσαρμόζεται καλύτερα με τις εφαρμογές αυτοκινούμενων ρομπότ είναι η ηλεκτρομαγνητική. Όταν χρησιμοποιείται σε επίπεδες επιφάνειες η πυξίδα αυτή καταγράφει τον οριζόντιο συντελεστή του μαγνητικού πεδίου της Γης προσφέροντας οφέλη όπως χαμηλή κατανάλωση ενέργειας, απουσία μετακινούμενων εξαρτημάτων, αντίσταση σε κλονισμούς και αναταράξεις και άμεση εκκίνηση λειτουργίας. Αν μιλάμε για ανομοιογενείς - ανώμαλες επιφάνειες το πηνίο του αισθητήρα θα πρέπει να στερεωθεί κατάλληλα ώστε να αποφευχθούν σοβαρά σφάλματα εξαιτίας του κατακόρυφου συντελεστή του γεωμαγνητικού πεδίου της Γης.

Επιπλέον οι ηλεκτρομαγνητικές πυξίδες συνδυάζονται αποτελεσματικά με τα γυροσκόπια τα οποία συζητήσαμε προηγουμένως. Συγκεκριμένα η πυξίδα παρέχει αξιόπιστες μακροπρόθεσμες πληροφορίες για την κατεύθυνση του ρομπότ χωρίς να επηρεάζεται από τις γεωμαγνητικές διαταραχές ενώ το γυροσκόπιο είναι ακριβές για μικρά χρονικά διαστήματα χωρίς με την σειρά του να επηρεάζεται από την επιτάχυνση της βαρύτητας ή τις κλίσεις του εδάφους.

Ενεργοί Σηματοδοτές

Τα συστήματα πλοήγησης με την συμμετοχή ενεργών σηματοδοτών είναι τα πιο συνηθισμένα υποστηρικτικά μέσα τόσο για την πλοήγηση πλοίων ή αεροπλάνων όσο και για τα εμπορικά συστήματα αυτοκινούμενων ρομπότ. Οι σηματοδοτές αυτοί που λειτουργούν κατά κάποιον τρόπο σαν φάροι, είναι εύκολο να εντοπιστούν και παρέχουν ακριβή δεδομένα όσον αφορά την θέση του ρομπότ με το μικρότερο δυνατό κόστος επεξεργασίας. Το μειονέκτημα τους είναι ότι απαιτούν υψηλό κόστος εγκατάστασης και συντήρησης. Επιπλέον θα πρέπει να εγκατασταθούν με μεγάλη ακρίβεια σε καθορισμένα σημεία προκειμένου να παρέχουν ακριβή εντοπισμό θέσης.

Για την υλοποίηση των συστημάτων αυτών χρησιμοποιούνται συνήθως τρεις ή περισσότεροι σηματοδοτές – πομποί σε σταθερά και προκαθορισμένα σημεία, ενώ το ρομπότ από την μεριά του είναι εφοδιασμένο με ένα δέκτη ή και το αντίθετο. Μέσω του δέκτη αυτού είμαστε σε θέση να καταγράψουμε την σχετική απόσταση και γωνία του ρομπότ σε σχέση με τα σταθερά σημεία ή με άλλα λόγια να εντοπίσουμε την ακριβή του θέση στον χώρο.

Μία διάσημη εφαρμογή αυτού του τύπου έχει υλοποιηθεί στο πανεπιστήμιο του MIT [8] η οποία ονομάστηκε CONAC (Computerized Opto- electronic Navigation and Control) και χρησιμοποιεί για πομπό μια μονάδα λέιζερ STROAB (STRuctured Opto- electronic Acquisition Beacon), η οποία έχει τοποθετηθεί πάνω στο κινούμενο μηχανισμό (εικόνα 3.2).



Εικόνα 3.2: Ένα και μόνο STROAB διαθέτει ένα λέιζερ κάθετης διασποράς ενώ περιστρέφεται με ταχύτητα 3000 στροφές/λεπτό

Παγκόσμιο Σύστημα Εντοπισμού - GPS (Global Positioning System)

Το GPS (Global Positioning System, Παγκόσμιο Σύστημα Εντοπισμού) ελέγχει 24 δορυφόρους μεγάλης ακριβείας οι οποίοι είναι σε θέση να εκπέμπουν ραδιοσήματα με μεγάλη ακρίβεια. Σύμφωνα με αυτό, η ακριβής θέση ενός σημείου που βρίσκεται επάνω στην επιφάνεια της γης προσδιορίζεται από τη λήψη και σύγκριση των σημάτων τριών τέτοιων δορυφόρων. Η θέση δηλαδή του σημείου μεταφράζεται ως η μοναδική τομή τριών κώνων των οποίων οι κορυφές είναι οι τρεις δορυφόροι.

Η τυπική ακρίβεια μέτρησης του GPS είναι $\pm 100\text{m}$ και μπορεί να φθάσει τα $\pm 10\text{m}$ με τη βοήθεια διαφορικού GPS. Το σύστημα GPS επιτρέπει τον προσδιορισμό της θέσης σημείων σε όλο τον κόσμο 24 ώρες το 24ωρο κάτω από οποιοσδήποτε καιρικές συνθήκες και μπορεί να συνδυαστεί άμεσα με συστήματα GIS (Geographical Information System) καθώς και με συστήματα ηλεκτρονικών χαρτών και πληροφοριακών συστημάτων ECDIS (Electronic Chart Display and Information System).

Όσον αφορά στον τομέα της πλοήγησης αυτοκινούμενων ρομπότ η χρήση του GPS για τον εντοπισμό της θέσης του οχήματος γίνεται όλο και πιο δημοφιλής. Μια σημαντική μελέτη στον τομέα αυτό πραγματοποιήθηκε το 1993 από τον Raymond H. Byrne [Byrne 1993] στο τμήμα ανάπτυξης και εξέλιξης οχημάτων στο Νέο Μεξικό. Ο Byrne διεξήγαγε μια σειρά από λεπτομερείς συγκριτικές δοκιμές με πέντε διαφορετικούς GPS δέκτες. Οι δοκιμές εστιάστηκαν στην ευαισθησία του δέκτη, την στατική και δυναμική ακρίβεια του και το πλήθος των GPS δορυφόρων που εντοπίστηκαν.

Η έρευνα αυτή απέδειξε ότι η χρήση GPS είναι ιδανικό εργαλείο για πολλές εφαρμογές πλοήγησης σε εξωτερικούς χώρους. Στα μειονεκτήματα της τεχνικής αυτής όσον αφορά την πλοήγηση αυτοκινούμενων ρομπότ συγκαταλέγονται:

1. Η περιοδική διακοπή του σήματος εξαιτίας του ανώμαλου εδάφους,
2. Ανακριβή δεδομένα για την θέση του ρομπότ όταν χρησιμοποιηθεί σε πρωτεύοντα – αυτόνομα συστήματα πλοήγησης.

Πλοήγηση βασισμένη σε Ορόσημα (Landmark Navigation)

Τα ορόσημα είναι διακριτά χαρακτηριστικά γνωρίσματα τα οποία μπορεί το ρομπότ να αναγνωρίσει σαν εισερχόμενα σήματα στους αισθητήρες του. Μπορεί να είναι γεωμετρικά σχήματα (π.χ. ορθογώνια, γραμμές ή καμπύλες) και ίσως περιέχουν επιπρόσθετη πληροφορία (π.χ. με την μορφή ραβδωτών κωδίκων – bar codes). Σε γενικές γραμμές, τα ορόσημα έχουν μια προκαθορισμένη και γνωστή θέση την οποία το ρομπότ χρησιμοποιεί σαν σημείο αναφοράς προκειμένου να προσδιορίσει την σχετική θέση του.

Τα ορόσημα πρέπει να επιλέγονται προσεκτικά με σκοπό να είναι εύκολα αναγνωρίσιμα και γι' αυτό πρέπει να διακρίνονται ευκρινώς σε σχέση με το περιβάλλον που θα τοποθετηθούν. Πριν το ρομπότ μπορέσει να χρησιμοποιήσει κάποιο ορόσημο για την πλοήγηση του θα πρέπει να έχει αποθηκεύσει στην μνήμη του τα χαρακτηριστικά αυτού. Έτσι για να είναι επιτυχημένη η πλοήγηση το πρόβλημα εστιάζεται πλέον στο πόσο πιο αξιόπιστα θα είναι σε θέση να το αναγνωρίσει στον χώρο.

Για να απλοποιηθεί το πρόβλημα συχνά θεωρείται ότι η τρέχουσα θέση και ο προσανατολισμός του ρομπότ είναι περίπου γνωστά. Έτσι το ρομπότ θα πρέπει να προσπαθήσει να ανιχνεύσει τα ορόσημα σε μια περιορισμένη περιοχή. Για αυτό τον λόγο απαιτείται το ρομπότ να είναι εφοδιασμένο με κάποιο ακριβές σύστημα οδομετρίας.

Κάποιες προσεγγίσεις που έχουν γίνει μέχρι τώρα προσπαθούν να συνδυάσουν την τεχνική αυτή με την πλοήγηση που βασίζεται σε χάρτες και την οποία συζητάμε στην συνέχεια. Στην περίπτωση αυτή αρχικά χρησιμοποιούνται οι αισθητήρες του ρομπότ προκειμένου να εξαχθεί ένας χάρτης του περιβάλλοντος του και στην συνέχεια εντοπίζονται χαρακτηριστικές δομές στους χάρτες αυτούς που μπορούν στην πορεία να χρησιμοποιηθούν σαν ορόσημα.

Γενικά μπορούμε να διακρίνουμε δύο τύπους ορόσημων: τα τεχνητά και τα φυσικά. Πρέπει εδώ να τονίσουμε ότι τα φυσικά ορόσημα λειτουργούν πολύ καλύτερα σε δομημένα περιβάλλοντα, όπως διάδρομοι, νοσοκομειακές αίθουσες κ.τλ.. Στο σημείο αυτό είναι λογικό να σκεφτεί κάποιος ότι τα φυσικά ορόσημα είναι πιο αποδοτικά όταν έχουν κατασκευαστεί από τον άνθρωπο ειδικά για τους σκοπούς της πλοήγησης.

Για τον λόγο αυτό δίνουμε τους παρακάτω ορισμούς: **φυσικά ορόσημα** ονομάζονται τα αντικείμενα ή χαρακτηριστικά που υπάρχουν ήδη σε ένα περιβάλλον και εξυπηρετούν σκοπό διαφορετικό από αυτόν της πλοήγησης, ενώ από την άλλη **τεχνητά ορόσημα** είναι τα ειδικά σχεδιασμένα αντικείμενα ή χαρακτηριστικά που έχουν τοποθετηθεί στο περιβάλλον ακριβώς για να υποβοηθήσουν την πλοήγηση του ρομπότ.

Όσον αφορά τα πρώτα, το βασικότερο πρόβλημα είναι ο αξιόπιστος εντοπισμός και η αξιολόγηση τους από τους αισθητήρες του ρομπότ. Ο πιο συχνά χρησιμοποιούμενος αισθητήρας για τον σκοπό αυτό είναι η ψηφιακή όραση. Συνήθως σαν φυσικά ορόσημα λαμβάνονται μεγάλοι μήκους κάθετες ακμές όπως πόρτες, ενώσεις τοίχων και φωτιστικά σώματα στο ταβάνι. Η επιλογή των ορόσημων παίζει κρίσιμο ρόλο αφού θα καθορίσει την πολυπλοκότητα του όλου εγχειρήματος.

Μία σημαντική προσπάθεια που χρησιμοποιεί την συγκεκριμένη τεχνική έχει γίνει από την εταιρεία AECL (Atomic Energy of Canada Ltd) η οποία ανέπτυξε το σύστημα ARK Project One [9]. Σκοπός ήταν η δημιουργία ενός εξελιγμένου ρομποτικού συστήματος επονομαζόμενου ARK (Autonomous Robot for a Known Environment). Το σύστημα αυτό χρησιμοποιεί κάμερες και αισθητήρες λέιζερ και το μόνο κριτήριο για την επιλογή κάποιου αντικειμένου σαν φυσικό ορόσημο είναι αυτό να διακρίνεται ξεκάθαρα σε σχέση με το περιβάλλον είτε με βάση το χρώμα είτε με βάση την αντίθεση.

Η ανίχνευση γίνεται πολύ πιο εύκολα με την χρήση **τεχνητών ορόσημων** [10], εξαιτίας του ότι αυτά έχουν σχεδιαστεί με στόχο την βέλτιστη αντίθεση σε σχέση με το περιβάλλον. Επιπλέον είναι εκ των προτέρων γνωστά το ακριβές μέγεθος και σχήμα τους. Οι ερευνητές έχουν χρησιμοποιήσει μέχρι τώρα διάφορα μοτίβα με αποτέλεσμα οι διάφορες μέθοδοι να διαφέρουν τόσο στην γεωμετρία όσο και στην τεχνική προσδιορισμού της θέσης.

Πολλά από τα συστήματα που χρησιμοποιούν τεχνητά ορόσημα βασίζονται στην ψηφιακή όραση. Η ακρίβεια των μεθόδων αυτών εξαρτάται από την ακρίβεια με την οποία τα μοτίβα – ορόσημα είναι δυνατόν να εξαχθούν από τις εικόνες που λαμβάνονται. Η διαδικασία αυτή επηρεάζεται άμεσα από την σχετική απόσταση και την γωνία λήψης. Υπό φυσιολογικές συνθήκες υπάρχει απώλεια όσο μεγαλώνει η απόσταση ανάμεσα στο ορόσημο και το ρομπότ. Επιπλέον παρατηρείται ένα εύρος από γωνίες λήψεως για τις οποίες τα αποτελέσματα είναι καλύτερα ενώ τα πράγματα δυσκολεύουν όσο απομακρυνόμαστε από την «καλή» περιοχή.

Κλασικά παραδείγματα τέτοιων συστημάτων που συναντάμε στην βιβλιογραφία είναι του Fukui με χρήση ορόσημου σε σχήμα διαμαντιού [11], των Lapin, Mesaki και Masuda [12] που χρησιμοποίησαν επιφάνειες αντανάκλασης και τέλος των Fleury και Baron που εφάρμοσαν ενεργά ορόσημα και πιο συγκεκριμένα τεχνολογία LED [13].

Υπάρχουν όμως και συστήματα τεχνητών ορόσημων που δεν χρησιμοποιούν την ψηφιακή όραση. Τα ορόσημα που χρησιμοποιούνται τότε είναι συνήθως ραβδωτοί κώδικες (bar codes) ανιχνεύσιμοι από λέιζερ αναγνώστες. Προσπάθειες προς αυτή την κατεύθυνση έχουν γίνει από τον Everett για το σύστημα MDARS (Mobile Detection Assessment and Response System) [14]. Αντιθέτως μία πρωτοποριακή προσέγγιση έχει προταθεί από τον Feng [15] ο οποίος χρησιμοποιεί ένα κυκλικό τεχνητό ορόσημο ενώ με την χρήση μετασχηματισμού Hough στις λαμβανόμενες εικόνες εξάγει τις παραμέτρους του ελλειψοειδούς σε πραγματικό χρόνο.

Στο σύστημα που προτείνουμε χρησιμοποιούμε κατά κάποιο τρόπο την πλοήγηση βασισμένη σε ορόσημα με χρήση μάλιστα της ψηφιακής όρασης. Μονό που στην δική μας περίπτωση θέτουμε σαν ορόσημο το ίδιο το ρομπότ χωρίς να υπάρχει ανάγκη για επιπρόσθετο εξοπλισμό. Είναι το **έντονο κόκκινο χρώμα του AmigoBot** που το διακρίνει σαφώς από το περιβάλλον του και μας βοηθά στο να εντοπίσουμε την θέση του.

Παρακολούθηση βασισμένη σε Χάρτες - Μοντέλα

Η πλοήγηση που βασίζεται σε χάρτες, (map matching) είναι μία τεχνική σύμφωνα με την οποία το ρομπότ χρησιμοποιεί τους αισθητήρες του για να συνθέσει έναν χάρτη του τοπικού του περιβάλλοντος. Ο τοπικός αυτός χάρτης στην συνέχεια συγκρίνεται με ένα καθολικό χάρτη που έχει σχηματισθεί εκ των προτέρων και βρίσκεται αποθηκευμένος στην μνήμη. Αν η σύγκριση καταλήξει σε ταύτιση το ρομπότ μπορεί πλέον να προσδιορίσει την πραγματική του θέση και τον προσανατολισμό του στον χώρο.

Ο προ-αποθηκευμένος χάρτης μπορεί να είναι ένα μοντέλο του χώρου δημιουργημένο από κάποιο CAD εργαλείο ή μπορεί να εξάγεται από δεδομένα τα οποία έχουν συλλεχθεί προηγουμένως από τους αισθητήρες. Η πλοήγηση που βασίζεται σε χάρτες έχει το μεγάλο **πλεονέκτημα** του να χρησιμοποιεί την φυσική και συνήθη δομή των εσωτερικών χώρων για να πληροφορηθεί την θέση του ρομπότ χωρίς να πρέπει να τροποποιήσουμε το περιβάλλον του. Επιπλέον με κάποιους πιο εξελιγμένους αλγόριθμους το ρομπότ είναι σε θέση να «μάθει» κάποιο νέο περιβάλλον και να βελτιώσει την ακρίβεια της μεθόδου ενώ συνεχίζει να πλοηγείται στον χώρο. Στα **μειονεκτήματα** της μεθόδου συγκαταλέγονται οι αυστηρές απαιτήσεις όσον αφορά την ακρίβεια του χάρτη που συντίθεται με την βοήθεια των αισθητήρων και η απαίτηση για κάποιες σταθερές παραμέτρους, χαρακτηριστικά δηλαδή που διακρίνονται εύκολα και θα χρησιμοποιηθούν για το ταίριασμα των συγκρινόμενων χαρτών.

Όπως έχουμε αναφέρει η τεχνική αυτή χρησιμοποιεί δύο βασικές διαδικασίες:

1. Την δημιουργία χαρτών (είτε κάποιου προκατασκευασμένου, είτε χάρτη που οφείλει να δημιουργήσει το ίδιο το ρομπότ) και
2. Την σύγκριση των χαρτών.

Όσον αφορά την **κατασκευή των χαρτών** ο Rencken [16] καθορίζει το πρόβλημα ως εξής: «με δεδομένη την θέση του ρομπότ και ένα σύνολο παραμέτρων τι είναι αυτό που βλέπει το ρομπότ;». Προφανώς η ικανότητα του ρομπότ στον να δημιουργεί χάρτες εξαρτάται άμεσα από την ικανότητα των αισθητήριων μηχανισμών του. Ένα πρόβλημα που σχετίζεται με την δημιουργία χαρτών είναι αυτό της «αυτόνομης εξερεύνησης» [17].

Προκειμένου να χτίσει τον χάρτη το ρομπότ πρέπει να εξερευνήσει το περιβάλλον του. Συνήθως στην περίπτωση αυτή θεωρείται ότι το ρομπότ ξεκινά την εξερεύνηση χωρίς καμία προγενέστερη γνώση για το περιβάλλον του. Κατά την κίνηση του όμως ακολουθεί μία συγκεκριμένη στρατηγική που σκοπεύει να καλύψει τον μέγιστο δυνατό χώρο στον ελάχιστο χρόνο. Μία τέτοια στρατηγική ονομάζεται «στρατηγική εξερεύνησης» (exploration strategy) και εξαρτάται άμεσα από το είδος των αισθητήρων που χρησιμοποιούνται. Ένα παράδειγμα απλής στρατηγικής εξερεύνησης δίνεται από τους Edlinger και Puttkamer [18].

Οι περισσότεροι ερευνητές πιστεύουν ότι ένας αισθητήρας δεν μπορεί από μόνος του να συλλέξει επαρκή δεδομένα για ένα πραγματικό περιβάλλον. Για να ξεπεραστεί αυτός ο περιορισμός είναι αναγκαίος ο συνδυασμός των δεδομένων που λαμβάνονται από διαφορετικούς αισθητήρες μια διαδικασία που αποκαλείται «σύνθεση αισθητήρων» (sensor fusion). Για παράδειγμα ο Buchberger [19] και ο Jorg [20, 21] ανέπτυξαν έναν μηχανισμό που αξιοποιεί ετερογενή δεδομένα τα οποία έχουν ληφθεί από ένα λέιζερ ραντάρ και από ένα σόναρ προκειμένου να δημιουργήσουν αξιόπιστα και πλήρη μοντέλα του περιβάλλοντος.

Όσον αφορά την δεύτερη διαδικασία δηλαδή τη **σύγκριση – ταίριασμα των χαρτών**, αυτή αφορά την εξαγωγή της σχέσης που μπορεί να υπάρχει ανάμεσα σε κάποιον τοπικό χάρτη και έναν προ-αποθηκευμένο καθολικό χάρτη [22]. Το πρόβλημα αυτό συχνά θεωρείται σαν μέρος του γενικότερου προβλήματος ταιριάσματος μιας εικόνας που έχει ληφθεί από συγκεκριμένη θέση και γωνία, με τις εικόνες που συνιστούν κάποιο μοντέλο (π.χ. [23]).

Σε γενικές γραμμές το ταίριασμα επιτυγχάνεται αφού σε πρώτη φάση εξαχθούν κάποια συγκεκριμένα χαρακτηριστικά, και στην συνέχεια τα χαρακτηριστικά αυτά αντιστοιχισθούν ανάμεσα στους δύο χάρτες. Οι αλγόριθμοι που χρησιμοποιούνται για το ταίριασμα μπορούν να κατηγοριοποιηθούν ως εξής:

- βασισμένοι σε εικονίδια (icon-based) και
- βασισμένοι σε χαρακτηριστικά (feature-based).

Παρουσιάσαμε στο κεφάλαιο αυτό τα προβλήματα που συναντάει κανείς κατά τον τηλεχειρισμό ρομπότ που χρησιμοποιούνται στην διαδικασία εξερεύνησης απομακρυσμένων χώρων.

Συμπερασματικά μπορούμε να πούμε ότι τα δύο σημαντικότερα είναι:

- a. η καθυστέρηση επικοινωνίας ανάμεσα στον χειριστή και το ίδιο το ρομπότ που μειώνει την διαφάνεια της όλης διαδικασίας και απαιτεί εξειδικευμένο χρήστη και μία ολοκληρωμένη διεπαφή χρήστη, και
- b. ο εντοπισμός της ακριβούς θέσης του ρομπότ στο πραγματικό περιβάλλον.

Γι' αυτό και οι περισσότερες προσπάθειες που εντοπίσαμε στην βιβλιογραφία προσπαθούν να αντιμετωπίσουν αποτελεσματικά τις δύο αυτές δυσκολίες. Στο κεφάλαιο 5 θα παρουσιάσουμε την μέθοδο που προτείνουμε με την παρούσα μελέτη αφού προηγουμένως όμως περιγράψουμε επαρκώς το περιβάλλον με το οποίο εργαστήκαμε και κυρίως τα τεχνικά χαρακτηριστικά του ρομπότ μας (AmigoBot).

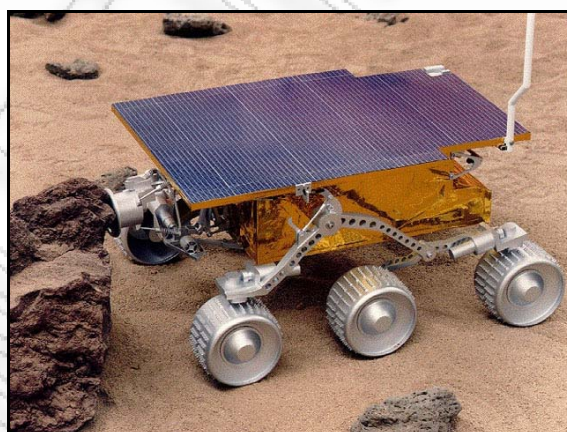
3.4 Σημαντικές εφαρμογές πλοήγησης αυτοκινούμενων ρομπότ

Για λόγου πληρότητας της μελέτης μας περιγράφουμε στην συνέχεια μερικές από τις σημαντικότερες και πιο γνωστές εφαρμογές πλοήγησης αυτοκινούμενων ρομπότ.

3.4.1 Το πρόγραμμα Pathfinder της NASA

Όταν αναφερόμαστε στην αιχμή της τεχνολογίας πλοήγησης ρομπότ, οι περισσότεροι άνθρωποι φέρνουν στο μυαλό τους το όχημα “Sojourner” [24] που προσεδάφιστηκε στον Άρη στις 4 Ιουλίου του 1997 στα πλαίσια του προγράμματος “Pathfinder” της NASA. Γι’ αυτό είναι χρήσιμο να εξετάσουμε τον τρόπο λειτουργίας του, τις δυνατότητες που διέθετε και τα προβλήματα που ήταν σε θέση να επιλύει. Η κάψουλα που προσεδάφιστηκε στον Άρη αποτελείτο από ένα διαστημόπλοιο προσεδάφισης το οποίο περιέκλειε και προστάτευε το όχημα, ενώ μετά την προσεδάφιση χρησίμευε σαν σταθμός βάσης και επικοινωνίας για το Sojourner το οποίο πραγματοποιούσε αποστολές εξερεύνησης στην κοντινή περιοχή. Το όχημα παρέμεινε ενεργό έως τις 27 Σεπτεμβρίου του ίδιου χρόνου οπότε και η επικοινωνία με την Γη χάθηκε και στο διάστημα αυτό εκπλήρωσε τους σημαντικότερους στόχους της αποστολής, συλλέγοντας τεράστιες ποσότητες πληροφοριών για την επιφάνεια του πλανήτη.

Το Sojourner, το οποίο διέθετε 6 ρόδες και είχε μέγεθος 68cm x 48cm x 28cm έχει περιγραφεί από πολλούς επιστήμονες σαν ένα διαστημόπλοιο από μόνο του, εξαιτίας του ότι ήταν εξοπλισμένο με μία εντυπωσιακή σειρά από συσκευές, αισθητήρες και εξαρτήματα προκειμένου να επιβιώσει στις αντίξοες συνθήκες του Άρη (με ημερήσιες θερμοκρασίες που κυμαίνονται από τους -100 °C ως τους -20 °C).



Εικόνα 3.3: Το όχημα Sojourner του προγράμματος Pathfinder

Ο σχεδιασμός του καθορίστηκε σε μεγάλο βαθμό από τους ενεργειακούς περιορισμούς του περιβάλλοντος στον οποίο καλείτο να χρησιμοποιηθεί. Συγκεκριμένα όλη η ενέργεια έπρεπε να παραχθεί από τα σχετικά μικρά ηλιακά πλέγματα που ήταν προσαρμοσμένα στο αμάξωμα του (ενώ διέθετε και κάποιες ελαφριές μη επαναφορτίσιμες μπαταρίες που θα χρησιμοποιούνταν όμως μόνο εφεδρικά). Αυτό σήμαινε ότι δεν ήταν δυνατή η παραγωγή επαρκούς ενέργειας για την ταυτόχρονη λειτουργία των διαφόρων συσκευών του. Για παράδειγμα δεν μπορούσε να κινείται καθώς χρησιμοποιούσε τους αισθητήρες του ή ενώ μετέδιδε πληροφορίες και γι’ αυτό ακολουθούσε μία στρατηγική «σταμάτα – ξεκίνα».

Κατά την περιπλάνηση του το Sojourner παρακολουθήτο από στερεοσκοπικές κάμερες υψηλής ανάλυσης τοποθετημένες στον σταθμό βάσης. Το ρομπότ επιπλέον αναμετέδιδε συνεχώς προς τον σταθμό βάσης πληροφορίες τηλεμετρίας και άλλες πληροφορίες για την κατάσταση των αισθητήρων του και των άλλων συστημάτων του.

Δύο φορές κατά την διάρκεια της ημέρας οι εικόνες και τα δεδομένα προωθούνταν από τον σταθμό βάσης προς την Γη. Στο κέντρο ελέγχου πίσω στην Γη τα δεδομένα αυτά και οι εικόνες συνδυάζονταν προκειμένου να κατασκευαστούν τρισδιάστατες απεικονίσεις του ρομπότ και του περιβάλλοντος του. Έτσι οι επιστήμονες μπορούσαν να σχηματίσουν μία ρεαλιστική εκτίμηση της κατάστασης.

Η πλοήγηση του οχήματος ήταν μία διαδικασία δύο βημάτων.

1. Στην αρχή κάθε ημέρας ο άνθρωπος – χειριστής της επίγειας βάσης εξέταζε τις τρισδιάστατες εικόνες και σχεδίαζε μία διαδρομή σε μορφή ακολουθίας σημείων από τα οποία το ρομπότ θα έπρεπε υποχρεωτικά να περάσει και
2. Στην συνέχεια κατά την διάρκεια της ημέρας το ρομπότ, κατευθυνόταν αυτόνομα κατά μήκος της σχεδιασμένης τροχιάς, ενώ παρακολουθούσε την πρόοδο του μέσω των αισθητήρων οδομετρίας που διέθετε.

Στο σύστημα που προτείνουμε χρησιμοποιούμε παρόμοια τεχνική πλοήγησης και χάραξης πορείας βασισμένης σε σημεία. Λεπτομερής ανάλυση της τεχνικής που χρησιμοποιήσαμε παρουσιάζεται στο κεφάλαιο 5.

Καθώς το όχημα κατευθυνόταν προς κάποιο σημείο της τροχιάς πραγματοποιούσε συχνές στάσεις προκειμένου να ελέγξει τον χώρο για πιθανά εμπόδια ή άλλους παράγοντες επικινδυνότητας. Για τον εντοπισμό των εμποδίων χρησιμοποιούσε δύο εμπρόσθιες κάμερες και πέντε λέιζερ πομπούς που σάρωναν τον χώρο. Τα δεδομένα αυτά στην συνέχεια συνδυάζονταν για την κατασκευή μιας τρισδιάστατης απεικόνισης του περιβάλλοντος μέσα από την οποία το ρομπότ ήταν σε θέση να εντοπίσει τα εμπόδια. Αν κάποιο εμπόδιο χαρακτηριζόταν σαν ανυπέβλητο, τότε έμπαινε σε λειτουργία ένας προσωρινός ελιγμός για την αποφυγή του. Αν για οποιοδήποτε λόγο το ρομπότ δεν κατάφερε να επανέλθει στην προσχεδιασμένη τροχιά μέσα σε ένα εύλογο χρονικό διάστημα τότε ήταν αναγκασμένο να σταματήσει και να περιμένει νέες εντολές από την Γη.

Συμπερασματικά μπορούμε να πούμε ότι τελικά ότι ο βαθμός αυτονομίας πλοήγησης του Sojourner ήταν σχετικά μικρός. Ο ανθρώπινος παράγοντας ήταν και πάλι αυτός που έφερνε εις πέρας το μεγαλύτερο μέρος της διαδικασίας αποφυγής εμποδίων, της επιλογής των στόχων και της χάραξης της πορείας του ρομπότ. Μάλιστα οι αποστάσεις ανάμεσα στα υποχρεωτικά σημεία της τροχιάς έπρεπε να είναι ιδιαίτερα μικρές ειδικά όταν το ρομπότ έπρεπε να πλοηγηθεί σε ανώμαλες επιφάνειες. Εξαιτίας των περιορισμών στην επικοινωνία ήταν απαραίτητη ή ύπαρξη κάποιου βαθμού αυτονομίας για το Sojourner μιας και ήταν αδύνατος ο συνεχής έλεγχός του από τον επίγειο σταθμό. Παρολαυτά, η μικρή ταχύτητα κίνησης του σήμαινε ότι οι αποστάσεις που θα διένυε ανάμεσα στα «κενά» χρονικά διαστήματα, όπου ο έλεγχος του από την Γη ήταν αδύνατος, ήταν αρκετά μικρές ώστε να επιτρέπουν σημαντική χειροκίνητη βοήθεια. Η προσέγγιση αυτή αποδείχθηκε ιδιαίτερα επιτυχημένη για το πρόγραμμα Pathfinder και αυτό καταδεικνύεται από την διάρκεια της αποστολής και τις τεράστιες ποσότητες πληροφοριών που κατάφερε να συλλέξει.

3.4.2 Το πρόγραμμα AGV στο Πανεπιστήμιο της Οξφόρδης

Το πρόγραμμα «Όχημα Αυτόνομης Καθοδήγησης» (AGV - Autonomous Guided Vehicle Project) [25] που αναπτύχθηκε στο πανεπιστήμιο της Οξφόρδης το 1994, ήταν μια προσπάθεια να συνδυαστούν διάφορες τεχνικές πλοήγησης ρομπότ χρησιμοποιώντας ποικίλους αισθητήρες και μεθόδους. Τελικός σκοπός ήταν η παραγωγή ρομπότ για εμπορική χρήση που θα μπορούσαν να χρησιμοποιηθούν σε βιομηχανικές δραστηριότητες. Βασικό χαρακτηριστικό της προσπάθειας αυτής ήταν η ολοκλήρωση των αισθητήρων δηλαδή ο συνδυασμός των δεδομένων που λαμβάνονται από διαφορετικές συσκευές (σόναρ, κάμερες, λέιζερ κ.τ.λ.) ώστε να εξαλειφθούν οι αδυναμίες που έχουν οι συσκευές αυτές ξεχωριστά η κάθε μία.

Η μεγαλύτερη επιτυχία σημειώθηκε ίσως από τους Durrant – Whyte στον τομέα της σύνθεσης χαρτών (map building) και τον εντοπισμό του ρομπότ με χρήση σόναρ [26,27,28,29,30]. Αυτοί κατασκεύασαν σόναρ ανίχνευσης τα οποία ακολουθούν δυναμικά την σχετική κίνηση των χαρακτηριστικών του χώρου καθώς το ρομπότ κινείται. Τα σόναρ αυτά είναι συσκευές μικρού κόστους ενώ περισσότερα του ενός μπορούν να τοποθετηθούν σε ένα ρομπότ. Κύρια αποστολή τους είναι να επιστρέφουν πληροφορίες για το βάθος και την σχετική θέση των αντικειμένων. Στην συνέχεια οι πληροφορίες αυτές χρησιμοποιούνται για την αποφυγή αντικειμένων. Τα σόναρ επιπλέον χρησιμοποιούνται για την ανίχνευση ορόσημων, τον εντοπισμό της θέσης του ρομπότ και την σύνθεση χαρτών. Τα χαρακτηριστικά που σηματοδοτούνται σαν ορόσημα είναι περιοχές σταθερού βάθους στο σήμα του σόναρ και αντιστοιχούν σε γωνίες αντικειμένων ή άλλες παρόμοιες δομές του χώρου.

Μεγάλο μέρος της προσπάθειας εστιάστηκε στην διαδικασία εντοπισμού στα πλαίσια ενός γνωστού περιβάλλοντος. Το σύστημα έχει εφοδιαστεί εκ των προτέρων με έναν χάρτη και εντοπίζει με επιτυχία το ρομπότ πραγματοποιώντας συγκρίσεις με τον χάρτη αυτό. Η τεχνική αυτή έχει πολλά κοινά με την πλοήγηση βασισμένη σε μοντέλα με χρήση ψηφιακής όρασης που περιγράψαμε στο κεφάλαιο 3 η οποία απαιτεί απλά τον εντοπισμό ενός γνωστού αντικειμένου σε κάποια από τις ληφθείσες εικόνες.

Στα πλαίσια του προγράμματος AGV μεγάλη προσοχή δόθηκε στην σύνθεση χαρτών και τον εντοπισμό της θέσης του ρομπότ και σε άγνωστο περιβάλλον. Στον τομέα αυτό, ο Leonard [31] παρατηρεί ότι η σχετική έρευνα που έγινε σε ένα άλλο πρόγραμμα το INRIA () στην Γαλλία και χαρακτηρίστηκε σαν «σύνθεση χαρτών» (map fusion), η έμφαση δόθηκε στην δημιουργία των χαρτών παρά στην χρήση τους. Αυτή η παρατήρηση είναι πολύ σημαντική οι πλούσιοι σε πληροφορία και χαρακτηριστικά χάρτες που δημιουργούνται στο INRIA δεν είναι κατανάλωση αυτό που χρειάζεται η διαδικασία πλοήγησης ενός ρομπότ. Στο πρόγραμμα AGV με την χρήση των σόναρ, οι χάρτες που δημιουργούνται είναι πολύ πιο «φτωχοί» όσον αφορά την πυκνότητα των ορόσημων αλλά δίνεται πολύ μεγαλύτερη προσοχή στον τρόπο με τον οποίο δημιουργούνται και αξιοποιούνται. Οι Lee και Recce [32] έχουν εργαστεί στο ζήτημα της εκτίμησης της χρησιμότητας των χαρτών που δημιουργούνται αυτόματα από τα σόναρ ενός ρομπότ. Συγκεκριμένα όταν σχηματιστεί ένας χάρτης επιλέγονται πιθανές τροχιές από τυχαία επιλεγμένα σημεία και στην συνέχεια οι τροχιές αυτές ελέγχονται για την ασφάλεια και την καταλληλότητα τους.

Μία άλλη παρατήρηση από τον Leonard αναφέρει ότι καθώς κινείται ένα ρομπότ, μερικές φορές είναι σημαντικότερο να ακολουθήσει μία τροχιά που θα το διατηρεί σε αντιστοιχία και καλή επαφή με τον χάρτη παρά να ακολουθήσει την συντομότερη ή πιο γρήγορη διαδρομή. Μία αναλογία στην διαπίστωση αυτή συναντάμε στους αγώνες μεγάλων αποστάσεων σε

ανώμαλα εδάφη είναι προτιμότερο στην περίπτωση αυτή για τους αθλητές να επιλέξουν μία μεγαλύτερη διαδρομή με πολλά όμως σημεία ορόσημα προκειμένου να ελαχιστοποιήσουν την πιθανότητα να χάσουν τον δρόμο τους. Ακόμα και στα ζώα παρατηρείται ανάλογη συμπεριφορά όπου για παράδειγμα οι μέλισσες [33] θα επιλέξουν να πετάξουν στην μεγαλύτερη διαδρομή αν σε αυτή υπάρχει κάποιο εύκολα διακριτό χαρακτηριστικό, π.χ. ένα μεγάλο δέντρο.

Ένα άλλο σημαντικό συμπέρασμα που προκύπτει από το πρόγραμμα AGV είναι το ότι η χρήση των σόναρ εντείνει το πρόβλημα του συσχετισμού των δεδομένων σε σχέση με την ψηφιακή όραση. Δηλαδή όσο μικρότερη πληροφορία συλλέγεται από τα σόναρ τόσο πιο ανασφαλής είναι τα συμπεράσματα που εξάγονται για το αν ένα παρατηρούμενο χαρακτηριστικό είναι σημαντικό ή όχι για την διαδικασία της πλοήγησης.

Άλλοι τομείς έρευνας με τους οποίους καταπιάστηκε το πρόγραμμα AGV έχουν να κάνουν με την δημιουργία ενός συστήματος ενεργού σηματοδότη με χρήση λέιζερ για ακριβή εντοπισμό θέσης σε ελεγχόμενο περιβάλλον, το οποίο και είχε εξαιρετικά αποτελέσματα, καθώς και μία παράλληλη υλοποίηση του συστήματος DROID του Harris [34] που κάνει χρήση παθητικής ψηφιακής όρασης.

3.4.3 MIT: Νοημοσύνη χωρίς Απεικόνιση

Μία τελείως διαφορετική προσέγγιση στο πρόβλημα της πλοήγησης αλλά και γενικότερα στην ρομποτική έχει ακολουθηθεί από τον Brooks και τους συνεργάτες του στο πανεπιστήμιο MIT [35,36,37]. Η μέθοδος τους μπορεί να συνοψιστεί στην φράση «νοημοσύνη χωρίς απεικόνιση». Τα ρομποτικά συστήματα που έχουν αναπτύξει συντίθενται από μία συλλογή απλών συμπεριφορών που αλληλεπιδρούν μεταξύ τους. Δεν υπάρχει κεντρικός έλεγχος, κεντρική επεξεργασία για την λήψη αποφάσεων, ούτε γίνεται προσπάθεια για τον σχηματισμό μιας σφαιρικής άποψης των τεκταινομένων. Οι απλές συμπεριφορές ενός τέτοιου συστήματος χειρίζονται η κάθε μία ένα μικρό ποσοστό απεικόνισης του εξωτερικού κόσμου. Πολλές μάλιστα από αυτές είναι απλές αντανάκλαστικές αντιδράσεις που μετασχηματίζουν τα δεδομένα που λαμβάνονται από κάποιον αισθητήρα σε κάποια ενέργεια του ρομπότ.

Βασικό χαρακτηριστικό της φιλοσοφίας αυτής στην ρομποτική είναι η άποψη ότι τα μυστικά του σχεδιασμού πραγματικά ευφυών ρομπότ μπορούν να αποκαλυφθούν μόνο μέσα από την υλοποίηση και δοκιμή τους στον πραγματικό κόσμο. Τα αποτελεσματικά ρομποτικά συστήματα πρέπει να χτίζονται σε επίπεδα ακολουθώντας προσέγγιση από κάτω προς τα πάνω (bottom-up). Αρχικά δηλαδή θα πρέπει να υλοποιείται και να δοκιμάζεται διεξοδικά η βασική λειτουργικότητα που θα πρέπει να διαθέτει το σύστημα, ενώ τα υπόλοιπα επίπεδα συμπεριφοράς θα πρέπει να προστίθενται διαδοχικά στην συνέχεια. Κάθε υψηλότερο επίπεδο θεωρείται ικανό να λειτουργήσει όταν δεν απαιτεί τον έλεγχο ή την υποστήριξη κάποιου από τα χαμηλότερα. Η προσέγγιση αυτή αποκαλείται «αρχιτεκτονική υπαγωγής».

Έτσι όσον αφορά το θέμα της πλοήγησης και εξερεύνησης χώρων που μας ενδιαφέρει, το βασικότερο επίπεδο κατά την υλοποίηση θα πρέπει να αφορά την ικανότητα αποφυγής συγκρούσεων με αντικείμενα του χώρου. Το επίπεδο αυτό θα μπορούσε να υλοποιηθεί με ρουτίνες άμεσης απόκρισης του συστήματος μετάδοσης κίνησης του ρομπότ στα δεδομένα που λαμβάνονται από ένα σόναρ. Το αμέσως επόμενο επίπεδο θα μπορούσε να αφορά την επιθυμία για τυχαία περιπλάνηση στον χώρο. Αν όμως το ρομπότ είναι ασφαλές από

συγκρούσεις πλέον είναι εύκολο να επιλέξει μία τυχαία κατεύθυνση και να την ακολουθήσει. Σε ακόμα πιο υψηλό επίπεδο θα μπορούσε να υλοποιηθεί η επιθυμία για εξερεύνηση άγνωστων περιοχών. Στην περίπτωση αυτή η τυχαία περιπλάνηση του προηγούμενου επιπέδου θα μπορούσε να αποκαλύψει κάποια μεγάλη ανεξερεύνητη περιοχή και στην συνέχεια το τρίτο επίπεδο να στείλει το ρομπότ προς την κατεύθυνση της .

Το αποτέλεσμα της προσέγγισης αυτής στο σύνολο της, είναι η δημιουργία ρομπότ που εμφανίζουν συμπεριφορές ανάλογες με της πραγματικής ζωής. Μάλιστα με τον τρόπο αυτό εμφανίζονται ακολουθίες και πρότυπα συμπεριφορών που δεν έχουν υλοποιηθεί μέσω κώδικα. Αυτό προκύπτει σαν αποτέλεσμα της αλληλεπίδρασης των διαφορετικών υποσυστημάτων ελέγχου. Εντυπωσιακά αποτελέσματα της δουλειάς αυτής παρουσιάστηκαν με χρήση ρομπότ που διέθεταν «πόδια» παρόμοια με των εντόμων, όπου τα αρχικά ασυγχρόνιστα πόδια μπορούν να μάθουν να περπατούν συγχρονισμένα επιτρέποντας στο ρομπότ να κινείται προς τα εμπρός.

Η ερευνητική ομάδα του Brook έχει τώρα στρέψει την προσοχή της στην δημιουργία ενός ανθρωπόμορφου ρομπότ που αποκαλείται “COG” [38]. Αρκετές από τις αρχές ανάπτυξης αυτοκινούμενων ρομπότ έχουν διατηρηθεί και στην ανάπτυξη του COG, μαζί με την κεντρική αντίληψη ότι για να αποκτήσουμε μία βαθιά γνώση για την λειτουργία του ανθρωπίνου εγκεφάλου θα πρέπει να δημιουργήσουμε ανθρωπόμορφα ρομπότ πραγματικών διαστάσεων. Παρολαυτά φαίνεται ότι ο αρχικός ενθουσιασμός της ομάδας του MIT για τα αυτοκινούμενα ρομπότ, έχει αρχίσει να ξεθωριάζει αφού δεν έχει καταφέρει να παράξει πολλά ρομπότ που να μπορούν να εκτελέσουν χρήσιμες δραστηριότητες. Εντούτοις ορισμένες από τις αρχές της φιλοσοφίας του Brook, όπως η ιεράρχηση των δεξιοτήτων σε επίπεδα μοιάζουν να είναι ότι ακριβώς απαιτεί ένα αποτελεσματικό ρομποτικό σύστημα.

3.4.4 Το πρόγραμμα NavLab του πανεπιστημίου Carnegie Mellon

Μία άλλη σημαντική προσπάθεια στον τομέα της πλοήγησης αυτοκινούμενων ρομπότ έχει πραγματοποιηθεί στον πανεπιστήμιο Carnegie Mellon με το μεγάλης κλίμακας πρόγραμμα NavLab που ξεκίνησε στις αρχές της δεκαετίας του 80. Το πρόγραμμα αυτό είχε δύο κύριους τομείς έρευνας, την ομάδα UGV (Unmanned Ground Vehicles) που ασχολήθηκε με την αυτόνομη λειτουργία οχημάτων εκτός δρόμου και την ομάδα AHS (Automated Highways Systems) που προσπάθησε να περιορίσει ή και να εξαλείψει την ανθρώπινη συμμετοχή στα πλαίσια της οδήγησης οχημάτων σε αυτοκινητόδρομους.

Η ομάδα UGV [39,40] ακολούθησε μια προσέγγιση με χρήση πολλών αισθητήριων οργάνων, και την εφήρμοσε για το δύσκολο πρόβλημα της πλοήγησης εκτός δρόμου χρησιμοποιώντας μάλιστα οχήματα πραγματικού μεγέθους. Οι αισθητήρες περιελάμβαναν ψηφιακή όραση και ανιχνευτές λέιζερ. Σημαντικό στοιχείο αυτής της δουλειάς είναι η ανάπτυξη αλγορίθμων για την κατηγοριοποίηση του εδάφους και στην συνέχεια η δημιουργία χαρτών συντεταγμένων του χώρου οι οποίοι θεωρούνταν ασφαλείς για την διέλευση του οχήματος.

Η χάραξη της πορείας του οχήματος στην περίπτωση αυτή, είναι αποτέλεσμα μιας διαδικασίας ψηφοφορίας και επίλυσης διαφορών. Διαφορετικά υποσυστήματα αίσθησης και συμπεριφοράς ψηφίζουν κάθε φορά για την επιθυμητή αντίδραση του οχήματος με

διαφορετικό συντελεστή βαρύτητας ανάλογα με την σπουδαιότητα τους. Για παράδειγμα οι συμπεριφορές που αφορούν την ασφάλεια του οχήματος (π.χ. αποφυγή συγκρούσεων) παίρνουν τον υψηλότερο συντελεστή.

Το σύστημα αυτό είχε ενθαρρυντικά αποτελέσματα, αφού πέτυχε την αυτόνομη πλοήγηση οχημάτων σε άγνωστο έδαφος και σε απόσταση μεγαλύτερη του ενός χιλιομέτρου. Η ενσωμάτωση και ενός συστήματος GPS θα μπορούσε να επιφέρει ακόμα καλύτερα αποτελέσματα με την δημιουργία ενός σφαιρικού χάρτη (Global map) της υπό εξερεύνηση περιοχής.

Από την άλλη η ομάδα AHS [41,42,43], ασχολήθηκε με το πώς θα μπορέσουν τα οχήματα αυτοκινητοδρόμων να ελέγξουν τα ίδια τις κινήσεις τους συμβάλλοντας στην ασφάλεια των οδηγών. Το πρόβλημα αυτό είναι σαφώς πιο περιορισμένης κλίμακας από εκείνο της πλοήγησης εκτός δρόμου. Διάφοροι παράγοντες συντελούν σε αυτό, όπως η απαραίτητη σε κάθε αυτοκινητόδρομο ύπαρξη της λωρίδας κυκλοφορίας ή πολύ πιθανή θέαση του πίσω μέρους κάποιου προπορευόμενου οχήματος.

Η ομάδα AHS κατάφερε να αναπτύξει διάφορες τεχνικές που επέτρεπαν στα αυτοκίνητα να ακολουθούν τις στροφές ενός δρόμου, να διατηρούν αποστάσεις ασφαλείας από τα άλλα οχήματα ή και να εκτελούν πιο πολύπλοκους ελιγμούς όπως η αλλαγή λωρίδας.

Για να το πετύχουν αυτό συχνά χρησιμοποίησαν νευρωνικά δίκτυα και διάφορους αισθητήρες όπως ψηφιακή όραση και ραντάρ. Παρολαυτά αν λάβουμε υπόψιν μας τις τεράστιες αποστάσεις που πρέπει να διανύει καθημερινά ένα αυτοκίνητο στην πραγματική ζωή, τα συστήματα αυτά θα πρέπει να είναι εξαιρετικά αξιόπιστα προκειμένου να θεωρηθούν ασφαλέστερα από την ανθρώπινη οδήγηση.

Το πιο φιλόδοξο πείραμα που επιχειρήθηκε από την ομάδα AHS ονομάστηκε «Χωρίς Χέρια Διάσχιση της Αμερικής» (No Hands Across America) και μπήκε σε εφαρμογή το 1995. Στα πλαίσια αυτού του πειράματος ένα ημιαυτόνομο αυτοκίνητο καθοδηγήθηκε για περίπου 3000 μίλια από την μία ακτή των Η.Π.Α. στην άλλη. Κατά την διάρκεια του ταξιδιού, οι επιστήμονες που επέβαιναν στο όχημα ήταν σε θέση να πάρουν τον έλεγχο όπου κρινόταν απαραίτητο.

Τελικά το όχημα μπόρεσα να περάσει αυτόνομα από τις στροφές στο 98.2 % των περιπτώσεων ενώ η επιτάχυνση και τα φρένα ελέγχονταν καθόλη την διάρκεια από τους ανθρώπους. Τα αποτελέσματα αυτά είναι εντυπωσιακά αν αναλογιστεί κανείς την ποικιλία των δρόμων και των συνθηκών που συνάντησε το όχημα κατά την πορεία του. Εντούτοις ήταν απαραίτητη η ανθρώπινη καθοδήγηση για περίπου 50 μίλια στα οποία οι συνθήκες ήταν ιδιαίτερα περίπλοκες για τον υπολογιστή, συνήθως εξαιτίας των πολλών στάσεων που έπρεπε να πραγματοποιήσει το όχημα.

Βέβαια οι αυστηρά προκαθορισμένες συνθήκες και οι υψηλές ταχύτητες που συναντά ένα όχημα κινούμενο σε κάποιον αυτοκινητόδρομο κάνουν το συγκεκριμένο πρόβλημα να μοιάζει αρκετά διαφορετικό από αυτό της πλοήγησης σε άγνωστο περιβάλλον όπου συχνά απαιτείται η κίνηση σε πολλαπλές κατευθύνσεις.

Επιπλέον, σύμφωνα με την ανθρώπινη εμπειρία η οδήγηση σε κάποιο δρόμο δεν παρέχει απαραίτητως πολλές πληροφορίες για την περιοχή, που θα μπορούσαν να αποδειχθούν χρήσιμες αν την επισκεπτόμασταν αργότερα ως πεζοί. Τέλος η γνώση της θέσης του

οχήματος σε σχέση με τον κόσμο δεν κρίνεται απαραίτητη κατά την οδήγηση σε αυτοκινητοδρόμους.

Στα κεφάλαια που ακολουθούν περιγράφουμε το σύστημα πλοήγησης που αναπτύξαμε για τις ανάγκες της παρούσας μελέτης, θεωρώντας ότι έχουμε παρουσιάσει επαρκώς το απαιτούμενο θεωρητικό υπόβαθρο και ανάλογες σημαντικές προσπάθειες που έχουν πραγματοποιηθεί στον τομέα του ενδιαφέροντος μας.

4 Παρουσίαση και τεχνικά Χαρακτηριστικά του AmigoBot

Το AmigoBot είναι το μικρότερο μέλος της οικογένειας αυτοκινούμενων ρομπότ Pioneer η οποία περιλαμβάνει το Pioneer 1 και Pioneer AT, το Pioneer 2 και τα νεότερα Pioneer 3-DX και AT [44,45]. Αυτές οι μικρές πλατφόρμες είναι ιδανικές για έρευνα και ανάπτυξη ενώ μοιράζονται μία κοινή αρχιτεκτονική και κοινό λογισμικό ελέγχου με όλες τις άλλες πλατφόρμες της εταιρείας MOBILEROBOTS όπως το PeopleBot, PatrolBot και το PowerBot.

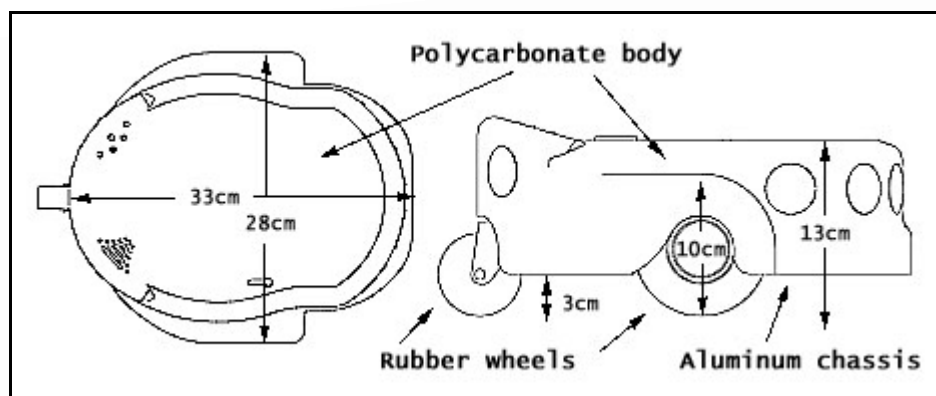


Εικόνα 4.1: Η βασική έκδοση του AmigoBot με την κεραία ασύρματης επικοινωνίας προσαρμοσμένη στο επάνω μέρος του.

4.1 Φυσικά Χαρακτηριστικά και Υποσυστήματα

Το AmigoBot είναι ένα έξυπνο αυτοκινούμενο ρομπότ, δεν είναι παιχνίδι αλλά μια πανίσχυρη κινούμενη μηχανή. Είναι έξυπνο γιατί διαθέτει έναν ενσωματωμένο μικροεπεξεργαστή με το αντίστοιχο λογισμικό ελέγχου (AmigOS) του καθώς επίσης και αισθητήρες που του επιτρέπουν να διαισθάνεται το περιβάλλον του και να περιπλανιέται ασφαλώς μέσα σε αυτό. Συνδυαζόμενο με το εξελιγμένο περιβάλλον ανάπτυξης λογισμικού ARIA (ActivMedia Robotics' Interface for Applications) της ActivMedia Robotics και το Saphira τα οποία εκτελούνται στην πλευρά του εξυπηρετούμενου (δηλ. στο pc) το AmigoBot μπορεί να προσδιορίσει την θέση του και να κινηθεί από σημείο σε σημείο ασφαλώς, είτε με την βοήθεια κάποιου χειριστή (τηλεχειρισμός) είτε αυτόνομα, αποφεύγοντας εμπόδια, περνώντας από πόρτες και κάτω από έπιπλα κ.ο.κ..

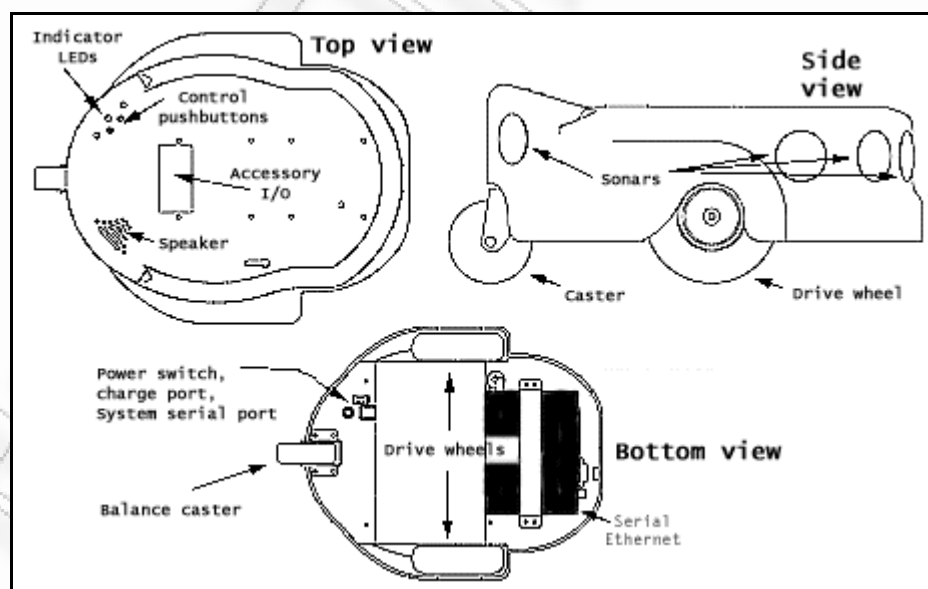
Στο σύστημα που προτείνουμε χρησιμοποιούμε ακριβώς τις κλάσεις του συγκεκριμένου API (Application Programming Interface) της ARIA προκειμένου να υλοποιήσουμε σε υψηλότερο επίπεδο αφάιρησης, μία διεπαφή έλεγχου του AmigoBot. Τελικός στόχος είναι να δημιουργήσουμε ένα περιβάλλον ελέγχου για το AmigoBot το οποίο να είναι λειτουργικό και εύχρηστο ακόμη και για τον μη εξειδικευμένο χρήστη, αποκρύπτοντας από αυτόν τις λειτουργίες χαμηλότερου επιπέδου.



Εικόνα 4.2: Βασικά χαρακτηριστικά και διαστάσεις του AmigoBot.

Το AmigoBot προορίζεται για να χρησιμοποιηθεί σε εσωτερικούς χώρους, όπως σπίτια, σχολεία, νοσοκομεία, γραφεία και εργαστήρια. Το μικρό, ελαφρύ και ευέλικτο AmigoBot διαθέτει 2 ρόδες από σκληρό καουτσούκ διαμέτρου 4 ιντσών, κάθε μία από τις οποίες καθοδηγείται από ένα αντιστρέψιμο μοτέρ συνεχούς ρεύματος (DC – Direct Current). Η ηλεκτρική ενέργεια κατανέμεται ξεχωριστά και ισορροπημένα ανάμεσα στα δύο μοτέρ ώστε να επιτευχθεί αρμονική και ανεξάρτητη κίνηση (τόσο προς τα εμπρός και προς τα πίσω όσο και σύμφωνα με τους δείκτες του ρολογιού και αντίστροφα).

Τέλος το AmigoBot διαθέτει και μία παθητική ρόδα στο μπροστινό μέρος η οποία χρησιμοποιείται για να ισορροπεί το ρομπότ. Σημαντικό είναι το ότι με την βοήθεια αυτής της μικρής ρόδας το AmigoBot, σε αντίθεση με άλλα ρομπότ, είναι σε θέση να περιστρέφεται χωρίς να μετακινηθεί από την θέση του, με αποτέλεσμα να μην αδρανοποιείται σε γωνίες και άλλες δύσκολες περιοχές.



Εικόνα 4.3: Οι αισθητήρες και τα εξαρτήματα ελέγχου και ειδοποιήσεων του AmigoBot.

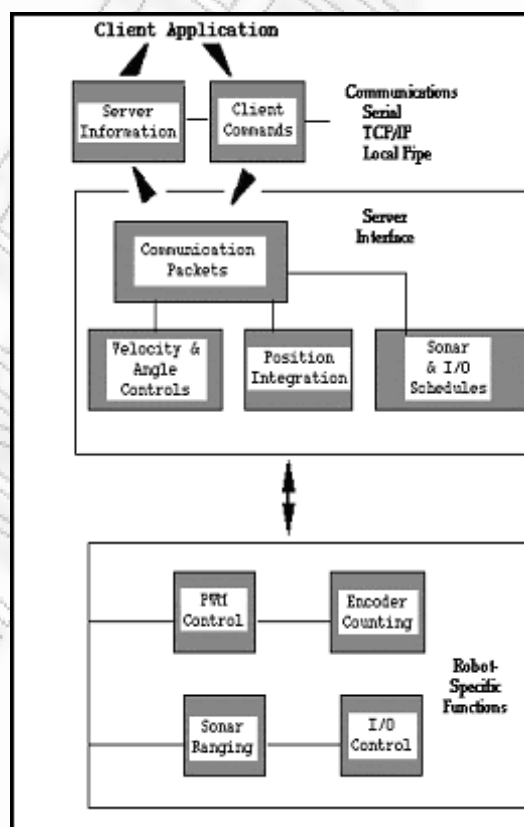
Περισσότερες λεπτομέρειες για το σύστημα ελέγχου της κίνησης του AmigoBot θα παραθέσουμε και στην συνέχεια.

4.2 Λογισμικό Ελέγχου και Αρχιτεκτονική του AmigoBot

Εγκέφαλος για κάθε AmigoBot είναι ο μικροεπεξεργαστής H8 της Hitachi των 20Mhz. Το λειτουργικό σύστημα του ρομπότ βρίσκεται μόνιμα αποθηκευμένο σε μία μόνο για ανάγνωση μνήμη FLASH. Το λειτουργικό αυτό το οποίο αποκαλείται AmigoOS, διαχειρίζεται όλες τις λειτουργίες χαμηλού επιπέδου των διαφόρων υποσυστημάτων του AmigoBot. Είναι υπεύθυνο δηλαδή για λειτουργίες όπως η χρήση του σόναρ η ανάγνωση των αποκωδικοποιητών που βρίσκονται στις ρόδες, η ρύθμιση της ταχύτητας για την εκτέλεση ελιγμών, η αναπαραγωγή προαποθηκευμένων ήχων κ.ο.κ..

Όπως τα περισσότερα λειτουργικά συστήματα το AmigoOS παραμένει κρυφό για τους χρήστες και λειτουργεί στο παρασκήνιο. Παρολαυτά παρέχει ελεύθερα προγραμματιστικές διεπαφές για την ανάπτυξη εφαρμογών από εξωτερικούς χρήστες. Το AmigoOS επιπλέον είναι εξοπλισμένο με ενσωματωμένα προγράμματα αυτοελέγχου τα οποία για παράδειγμα ελέγχουν την λειτουργία των μοτέρ ή των σόναρ και μπορούν να οδηγήσουν το ρομπότ σε μια στοιχειώδη αλλά ασφαλή περιπλάνηση στο χώρο ενώ ταυτόχρονα εκτελεί μια απλή ακολουθία κινήσεων και ήχων (Self Test Mode).

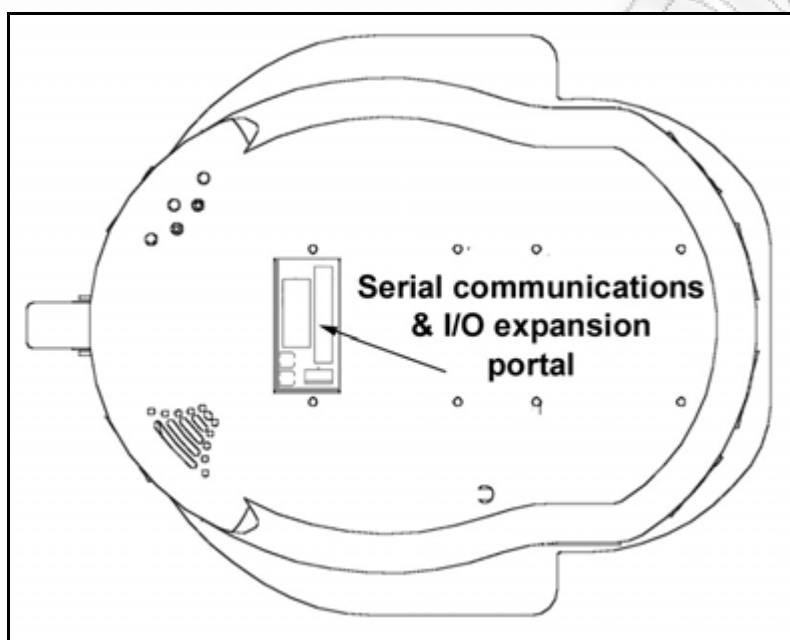
Όλες οι ρομποτικές πλατφόρμες της MobileRobots χρησιμοποιούν αρχιτεκτονική Πελάτη-Εξυπηρετητή (Client-Server) για τον έλεγχο τους. Ο ελεγκτής (controller) του ρομπότ έχει την ευθύνη της διαχείρισης όλων των λεπτομερειών χαμηλού επιπέδου των διαφόρων υποσυστημάτων του AmigoBot τα οποία τελικά πρέπει να καταστούν διαθέσιμα στις εφαρμογές – πελάτες. Μια γραφική απεικόνιση της αρχιτεκτονικής αυτής φαίνεται στην παρακάτω εικόνα.



Εικόνα 4.4: Η αρχιτεκτονική ελέγχου (Πελάτη – Εξυπηρετητή) της ActivMedia Robotics

4.3 Τύποι Επικοινωνίας του AmigoBot με τον Η/Υ

Κάθε AmigoBot είναι δυνατόν να συνδεθεί και να επικοινωνήσει με ηλεκτρονικό υπολογιστή είτε ενσύρματα είτε ασύρματα. Για ενσύρματη σύνδεση υπάρχει μία σειριακή υποδοχή των 9 pins που είναι τοποθετημένη στο πάνω μέρος του ρομπότ. Η χρήση της συγκεκριμένης σύνδεσης είναι χρήσιμη για λόγους συντήρησης του AmigoBot και για την απευθείας επικοινωνία με το λειτουργικό του σύστημα που βρίσκεται αποθηκευμένο στην μνήμη Flash.



Εικόνα 4.5: Η θύρα σειριακής επικοινωνίας του AmigoBot

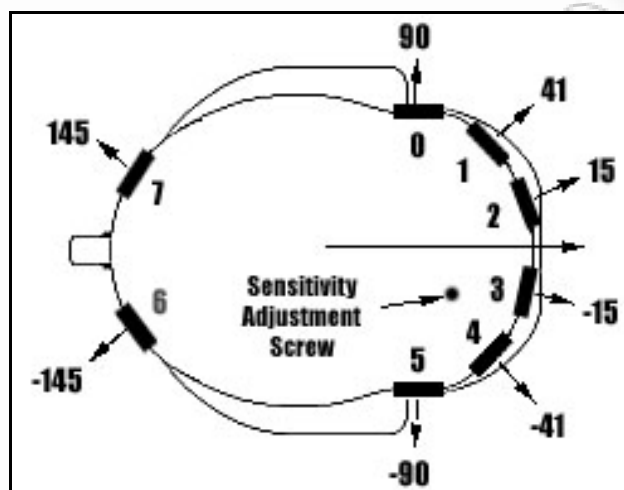
Για την **ασύρματη επικοινωνία** του με το PC, κάθε AmigoBot είναι εξοπλισμένο με μία κεραία Amigo Wirefree η οποία έρχεται σαν αποσπώμενο κομμάτι και προσαρμόζεται στο πάνω μέρος του. Το σύστημα ασύρματης επικοινωνίας έχει ισχύ πάνω από 100mW και χρησιμοποιεί σήμα 900MHz με εύρος συχνοτήτων μεταξύ 902 και 928 MHz.

Η εμβέλεια του σήματος σε εσωτερικούς χώρους τοποθετείται στα 100μ και παρόλο που δεν προορίζεται για χρήση σε εξωτερικούς χώρους η εμβέλεια του εκεί φτάνει τα 250μ. Τέλος ο ρυθμός μετάδοσης των δεδομένων κυμαίνεται ανάμεσα στα 9600kb/sec και 38400kb/sec ανάλογα με τον φόρτο του δικτύου που μπορεί να ενταθεί με την παρουσία και άλλων ρομπότ στον χώρο.

Στο σύστημα που προτείνουμε επιλέξαμε φυσικά να χρησιμοποιήσουμε την ασύρματη επικοινωνία αφού μας επιτρέπει πολλή μεγαλύτερη ελευθερία κινήσεων. Τέλος όπως έχουμε αναφέρει η ασύρματη επικοινωνία με το AmigoBot επιτυγχάνεται μέσω ενός ασύρματου δρομολογητή (Cisco Aironet 1100).

4.4 Σύστημα Σόναρ

Κάθε AmigoBot είναι εξοπλισμένο με μία απλή διάταξη από 8 σόναρ. Οι θέσεις τους είναι προκαθορισμένες και επιλεγμένες έτσι ώστε να επιτευχθεί κάλυψη 360 μοιρών. Έτσι έχουμε ένα σε κάθε πλευρά, τέσσερις μπροστά και δύο στο πίσω μέρος. Η αρίθμηση των σόναρ γίνεται σύμφωνα με τους δείκτες του ρολογιού ξεκινώντας από την αριστερή πλευρά (εικ. 4.6).



Εικόνα 4.6: Η διάταξη των σόναρ του AmigoBot

Η μήτρα μετάδοσης των σόναρ είναι πολυπλεγμένη (multiplexed), που σημαίνει ότι μόνο ένας δίσκος ενεργοποιείται ανά πάσα στιγμή. Η διάρκεια εκπομπής κάθε σόναρ είναι δυνατόν να προσαρμοστεί ενώ η προκαθορισμένη της τιμή είναι στα 25 Hz (40 milliseconds για κάθε δίσκο). Η ευαισθησία τους κυμαίνεται από τα 10εκ. έως τα 5μ.. Προσαρμόσιμη είναι και η ακολουθία εκπομπής των δίσκων με την προκαθορισμένη να είναι από αριστερά προς τα δεξιά σύμφωνα και με την αρίθμηση τους.

Το σύνολο των οχτώ σόναρ βρίσκεται κάτω από τον ίδιο κεντρικό έλεγχο. Παρόλο που η ευαισθησία και η εμβέλεια τους έχουν ρυθμιστεί εργοστασιακά, είναι δυνατόν να αλλαχθούν από τους χρήστες προκειμένου να είναι αποτελεσματικά σε διαφορετικά περιβάλλοντα λειτουργίας.

Για παράδειγμα αν ελαττώσουμε την ευαισθησία τους τότε το AmigoBot δεν θα είναι σε θέση να «βλέπει» μικρά αντικείμενα. Κάτι τέτοιο μπορεί να είναι επιθυμητό όταν για παράδειγμα σκοπεύουμε να χρησιμοποιήσουμε το ρομπότ μας σε ανώμαλες επιφάνειες ή πάνω σε παχύ χαλί. Αν η ευαισθησία είναι πολύ υψηλή το ρομπότ θα θεωρούσε άμεσα το χαλί σαν εμπόδιο.

Στο σύστημα που έχουμε αναπτύξει έχουμε αφήσει ανεπηρέαστες τις ρυθμίσεις του συστήματος των σόναρ αφού όλες μας οι δοκιμές έγιναν σε εργαστηριακούς εσωτερικούς χώρους και σε λείες επιφάνειες. Για τις συνθήκες αυτές διαπιστώσαμε ότι οι εργοστασιακές ρυθμίσεις λειτουργούν ικανοποιητικά .

4.5 Σύστημα Κίνησης του AmigoBot

Τα αισθητήρια συστήματα και το σύστημα κίνησης του AmigoBot ελέγχεται και ενεργοποιείται από έναν ελεγκτή ο οποίος καθοδηγείται από έναν πανίσχυρο μικροεπεξεργαστή Renesas SH2. Το λειτουργικό σύστημα AmigoSH διαθέτει 16 Kb μνήμη RAM και 128 Kb μνήμη FLASH.

Οι ρόδες του AmigoBot έχουν διάμετρο 4 ίντσες (10.2 εκ.) και είναι κατασκευασμένες από σκληρό καουτσούκ προσαρμοσμένο όμως για άριστη πρόσφυση. Το σύστημα κίνησης χρησιμοποιεί 2 μοτέρ συνεχούς ρεύματος (DC) που χαρακτηρίζονται από μεγάλη ταχύτητα και ροπή. Κάθε ένα από τα μοτέρ διαθέτει έναν οπτικό κωδικοποιητή υψηλής ακρίβειας που παράγει 123 κινήσεις (τικ) ανά χιλιοστόμετρο περιστροφής, για ακριβή ανάλυση της θέσης και της ταχύτητας. Επιπλέον τα δεδομένα που λαμβάνονται από τους κωδικοποιητές μας βοηθούν να εξάγουμε την απόσταση που το ρομπότ έχει διανύσει καθώς και τον προσανατολισμό του.

Οδομετρία

Οδομετρία είναι η διαδικασία εκτίμησης της θέσης ενός κινούμενου οχήματος κατά την πλοήγηση του. Ο όρος συχνά χρησιμοποιείται για να περιγράψει την απόσταση που έχει διανύσει ένα όχημα. Αφορά την αξιοποίηση των δεδομένων που παράγονται καθώς περιστρέφονται οι ρόδες με σκοπό την εκτίμηση της αλλαγής της θέσης του οχήματος σε σχέση με τον χρόνο. Συχνά αυτή η μέθοδος είναι πολύ ευαίσθητη σε λάθη και παρεκκλίσεις και γ' αυτό απαιτεί μεγάλη ακρίβεια και υψηλό ρυθμό δειγματοληψίας.

Στο σύστημα που προτείνουμε, όπως θα δούμε και στην συνέχεια, η οδομετρία παίζει καθοριστικό ρόλο αφού όχι μόνο μας επιτρέπει να εξάγουμε την θέση του ρομπότ αλλά και να συλλέξουμε ένα σωρό άλλες χρήσιμες πληροφορίες τις οποίες εμφανίζουμε σε πραγματικό χρόνο στον χρήστη, γεγονός που συμβάλλει κατά πολύ στην δημιουργία μιας φιλικής διεπαφής χρήστη.

Όσον αφορά το ρομπότ μας, AmigoBot, τρεις είναι οι μεταβλητές που έχουν αποθηκευθεί στην μνήμη FLASH και επηρεάζουν την οδομετρία του:

- Η **RevCount**, καθορίζει το πλήθος των τικ που θα μεταδίδει κάθε κωδικοποιητής ξεχωριστά σε κάθε ρόδα (διαφορικό σύστημα) στα πλαίσια μιας περιστροφής 180 μοιρών. Έτσι το AmigoSH είναι σε θέση να υπολογίσει τον προσανατολισμό και την ταχύτητα περιστροφής.
- Η **TicksMM** που περιγράφει το πλήθος των τικ ανά χιλιοστόμετρο περιστροφής και το AmigoSH την χρησιμοποιεί για να υπολογίσει την ταχύτητα του AmigoBot και την απόσταση που έχει διανύσει και τέλος
- Η **DriftFactor**, η οποία είναι ένας προσημασμένος αέριος που μετράται σε μονάδες του 1/8192 και προστίθεται ή αφαιρείται στην τιμή του κωδικοποιητή της αριστερής ρόδας, ώστε να αντισταθμίσει τυχόν διαφορές ανάμεσα στο μέγεθος των δύο ροδών.

Ακολουθεί η περιγραφή του συστήματος που έχουμε αναπτύξει για τις ανάγκες της συγκεκριμένης μελέτης.

5 Το προτεινόμενο Σύστημα Τηλεχειρισμού Ρομπότ μέσω Δικτυακής Κάμερας

Με βάση την ανάλυση που έχουμε παραθέσει μέχρι τώρα για τα ζητήματα που έχουν να κάνουν με τον τηλεχειρισμό ρομπότ φαίνεται ξεκάθαρα ότι είναι μία διαδικασία η οποία συχνά μπορεί να γίνει προβληματική ειδικότερα για τους αρχάριους ή ανεκπαιδευτους χειριστές.

5.1 Στόχος του προτεινόμενου Συστήματος

Συνοψίζοντας την παρούσα κατάσταση μπορούμε να συμπεράνουμε ότι τα σημαντικότερα **προβλήματα** που καλείται να επιλύσει ο δημιουργός μιας εφαρμογής τηλεχειρισμού και ελέγχου αυτοκινούμενων ρομπότ έχουν να κάνουν με:

1. Αδυναμία εντοπισμού του ρομπότ στον χώρο
2. Ανακριβή δεδομένα για την θέση του ρομπότ
3. Δυσχέρειες στην χάραξη της τροχιάς κίνησης του
4. Αποτυχία της διαδικασίας ανίχνευσης εμποδίων

Όπως έχουμε τονίσει ξανά, ακόμα και αν το όχημα διαθέτει ενσωματωμένες ικανότητες αυτοελέγχου (π.χ. διατήρησης πορείας – route following) όπως η ρομποτική πλατφόρμα που χρησιμοποιούμε και εμείς (AmigoBot) και ελέγχεται από εξειδικευμένο χειριστή, αστάθμητοι παράγοντες όπως προβλήματα επικοινωνίας ή υπερφορτωμένο δίκτυο, μπορούν να υπονομεύσουν τα επιθυμητά αποτελέσματα.

Για την αντιμετώπιση των παραπάνω προβλημάτων και προκειμένου ο τηλεχειρισμός ενός ρομπότ να γίνει αποτελεσματικότερος και παραγωγικότερος ώστε να ικανοποιήσει πραγματικές ανάγκες της ζωής του ανθρώπου, χρειαζόμαστε πλατφόρμες χειρισμού του ρομπότ που θα μας βοηθούν να αντιληφθούμε πληρέστερα το περιβάλλον στο οποίο κινείται, να κάνουμε ακριβείς εκτιμήσεις και να λάβουμε τις σωστές αποφάσεις για τον έλεγχο του.

Σκοπός της παρούσας μελέτης είναι να αναπτύξουμε είναι σύστημα τηλεχειρισμού ρομπότ μέσω δικτυακής κάμερας, το οποίο βασίζεται σε μία ολοκληρωμένη διεπαφή χρήστη και παράλληλα να προτείνουμε λύσεις για μερικά από τα σημαντικότερα προβλήματα που παρουσιάζει ένα τέτοιο εγχείρημα, όπως είναι ο εντοπισμός του ρομπότ στον χώρο και η ρεαλιστική πλοήγηση του με βάση τις πληροφορίες που λαμβάνουμε από την κάμερα

Τα σημεία στα οποία κυρίως επικεντρώσαμε την προσοχή μας είναι:

- η ανίχνευση του ρομπότ στον χώρο με βάση τις πληροφορίες που λαμβάνουμε από την κάμερα,
- η χάραξη ασφαλούς τροχιάς για το ρομπότ και τέλος
- η σωστή αναγωγή των διαστάσεων που προβάλλονται στην οθόνη σε διαστάσεις του πραγματικού χώρου.

Εκτός από αυτά βέβαια η εφαρμογή μας δεν παύει να αποτελεί μια ολοκληρωμένη διεπαφή χρήστη για τον έλεγχο του ρομπότ φροντίζοντας για παράδειγμα, για την σύνδεση και αποσύνδεση σε αυτό, για τον χειρισμό της κάμερας κ.α., παρέχοντας ταυτόχρονα δυναμικά, ένα πλήθος από πληροφορίες σχετικά με την όλη διαδικασία.

5.2 Η Μεθοδολογία που ακολουθήθηκε

Ξεκινώντας την μελέτη μας και την δημιουργία της εφαρμογής μας ήταν απαραίτητο αρχικά να εξασφαλίσουμε κάποια βασική λειτουργικότητα όσον αφορά την επικοινωνία με το ρομπότ ώστε να αφοσιωθούμε στην συνέχεια στην επίλυση δυσκολότερων ζητημάτων πλοήγησης του. Έτσι κάνοντας χρήση των προγραμματιστικών διεπαφών του περιβάλλοντος της ARIA βεβαιωθήκαμε ότι είμαστε σε θέση να πραγματοποιούμε επιτυχημένη σύνδεση και αποσύνδεση με την ρομποτική μας πλατφόρμα μέσω του ασύρματου ρούτερ. Μάλιστα η διαδικασία αυτή είναι παραμετροποιήσιμη με την έννοια ότι ο χρήστης έχει την δυνατότητα να επιλέγει κάθε φορά με ποιο ρομπότ επιθυμεί να συνδεθεί αν υπάρχουν στον χώρο περισσότερα του ενός.

Έπειτα μέσω της βιβλιοθήκης χειρισμού πολυμέσων **JMF** (Java Media Framework) ήμασταν σε θέση να εμφανίσουμε τις εικόνες που λαμβάνονται από την κάμερα σε πραγματικό χρόνο. Τα δεδομένα αυτά είναι τα πιο σημαντικά για το σύστημα πλοήγησης που προτείνουμε, αφού όλη η διαδικασία εντοπισμού και χάραξης τροχιάς του AmigoBot βασίζεται στην επεξεργασία τους.

Όπως έχουμε αναφέρει δύο είναι τα κύρια ζητήματα που μας απασχόλησαν και για τα οποία προσπαθήσαμε να δώσουμε μία βιώσιμη λύση.

1. **Ο ακριβής εντοπισμός του ρομπότ στον χώρο**, χωρίς να είναι απαραίτητη η γνώση προηγούμενων θέσεων του και
2. **Η ασφαλής καθοδήγηση του** στον χώρο.

Όσον αφορά τον εντοπισμό του AmigoBot ακολουθήσαμε την τεχνική των **τεχνητών ορόσημων**. Η μοναδικότητα της λύσης που προτείνουμε σε σχέση με άλλες υλοποιήσεις που περιγράψαμε στο κεφάλαιο 3, έχει να κάνει με το ότι σαν ορόσημο δεν χρησιμοποιούμε εξωτερικά αντικείμενα του χώρου αλλά το ίδιο το ρομπότ. Έχουμε βασιστεί δηλαδή στο έντονο κόκκινο χρώμα του αμαξώματος του το οποίο και το διακρίνει ξεκάθαρα από το περιβάλλον.

Αυτό μας δίνει μεγάλη ευελιξία αφού σε θεωρητικό τουλάχιστον επίπεδο η λύση που προτείνουμε λειτουργεί ανεξαρτήτως περιβάλλοντος. Αρκεί να υπάρχει μια κάμερα προσαρμοσμένη είτε επάνω στο ίδιο το ρομπότ είτε σε κάποιο σταθερό σημείο του χώρου η οποία θα το εποπτεύει. Ο μόνος περιορισμός έγκειται στο ότι το ρομπότ θα πρέπει να βρίσκεται διαρκώς μέσα στο οπτικό πεδίο της κάμερας.

Τέλος όσον αφορά την **καθοδήγηση και την χάραξη της τροχιάς** του AmigoBot ακολουθούμε τη τεχνική των **υποχρεωτικών σημείων**. Η πορεία δηλαδή που θα ακολουθηθεί από το ρομπότ, του μεταφέρεται σαν μία ακολουθία σημείων από τα οποία θα πρέπει υποχρεωτικά να διέλθει. Ανάλογη τεχνική χρησιμοποιήθηκε για την καθοδήγηση του ρομπότ Sojourner του προγράμματος Pathfinder της NASA, όπως έχουμε περιγράψει στην παράγραφο 3.4.1.

Μέσω των δεδομένων οδομετρίας τα σημεία αυτά μπορούν να χρησιμοποιηθούν και σαν μέσο ελέγχου και αξιολόγησης της συνέπειας με την οποία το ρομπότ ακολουθεί την επιθυμητή τροχιά.

Με βάση τα παραπάνω καταλήξαμε στο συμπέρασμα ότι για να υλοποιηθεί η παραπάνω λύση το πρόβλημα πρέπει να διαιρεθεί σε τρία επιμέρους κομμάτια. Αυτά είναι:

1. **Ο έλεγχος της κάμερας** και η εμφάνιση πληροφοριών σχετικών με την πλοήγηση στην οθόνη
2. **Ο εντοπισμός του ρομπότ** έπειτα από επεξεργασία των λαμβανόμενων εικόνων και τέλος
3. **Ο έλεγχος του ίδιου του ρομπότ** και η λήψη επιπρόσθετων πληροφοριών για την κίνηση του στον χώρο

Η διαπίστωση αυτή κατά την υλοποίηση μεταφράστηκε σε αντίστοιχες κλάσεις Java και μας επέτρεψε να διατηρήσουμε μια ξεκάθαρη αρχιτεκτονική στο σύστημα μας. Στην συνέχεια θα υπεισέλθουμε σε περισσότερες λεπτομέρειες σχετικά με την αρχιτεκτονική της προτεινόμενης λύσης όποτε τα παραπάνω θα γίνουν περισσότερο κατανοητά.

Σημαντικότερο ρόλο όπως έχουμε ξαναπεί, παίζει στο σύστημα μας η τεχνική της ψηφιακής όρασης, και η επεξεργασία των εικόνων που λαμβάνουμε από την δικτυακή κάμερα. Στην επόμενη παράγραφο αναλύουμε τον όρο «ψηφιακή όραση» και περιγράφουμε τον τρόπο με τον οποίο επεξεργαζόμαστε τις εικόνες προκειμένου να πετύχουμε τον εντοπισμό του ρομπότ και την ασφαλή καθοδήγηση του.

5.3 Μέθοδος Εντοπισμού του AmigoBot και Επεξεργασία Εικόνας

Παρόλο που η ψηφιακή όραση δεν έχει αποδειχθεί μέχρι τώρα σαν η πιο επιτυχημένη τεχνική αίσθησης του περιβάλλοντος για τα αυτοκινούμενα ρομπότ (για παράδειγμα τα σόναρ ή οι αισθητήρες υπέρυθρων ακτινών θεωρούνται προτιμότερες) σίγουρα είναι η πιο ελπιδοφόρα όσον αφορά το μακροπρόθεσμο μέλλον [46]. Η διεθνής έρευνα στον τομέα της ψηφιακής όρασης εντείνεται συνεχώς σε διεθνές επίπεδο αλλά φαίνεται να στρέφεται προς το παρόν σε άλλες εφαρμογές εκτός από την ρομποτική, από την αναγνώριση προσώπου και τα συστήματα παρακολούθησης για λόγους ασφαλείας έως τις τρισδιάστατες απεικονίσεις για την εικονική πραγματικότητα. Αυτό οφείλεται στην μεγαλύτερη ζήτηση που έχουν αυτές οι εφαρμογές και στην επιτυχία που έχουν σημειώσει τέτοιες υλοποιήσεις. Παρολαυτά είναι σίγουρο ότι μακροπρόθεσμα η χρήση της ψηφιακής όρασης στην ρομποτική θα αποδώσει καρπούς και είναι κάτι με το οποίο αξίζει να ασχοληθεί η επιστημονική κοινότητα.

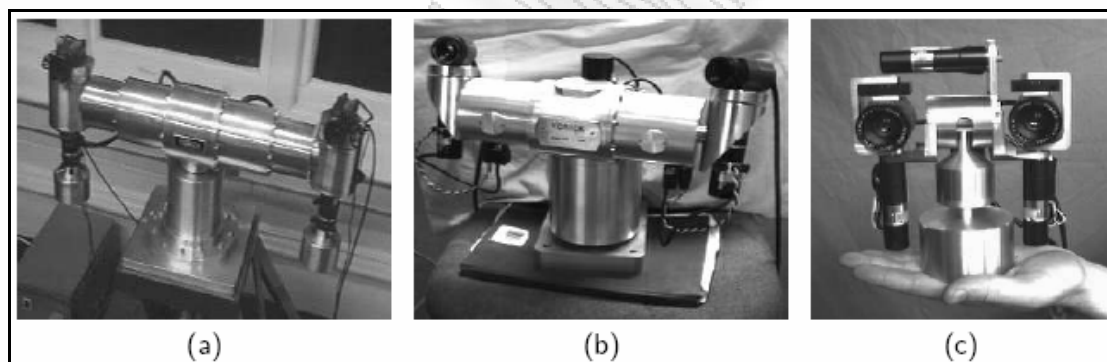
Η όραση είναι η αίσθηση που μπορεί να αποκαλύψει πληρέστερα από κάθε άλλη το «τι» και το «που» για τα αντικείμενα που είναι πιθανόν να συναντήσει στο περιβάλλον του ένα ρομπότ. Μάλιστα όσο κι αν θα πρέπει να είμαστε προσεκτικοί στην εξαγωγή αντιστοιχιών ανάμεσα στα ρομποτικά συστήματα και στους βιολογικούς οργανισμούς [69], είναι ξεκάθαρο ότι η όραση είναι το κύριο όργανο που χρησιμοποιούν για την πλοήγηση τους και τα περισσότερα ζώα.

Όσον αφορά τον άνθρωπο συγκαταλέγεται σίγουρα στα όντα που χρησιμοποιούν ένα σύστημα **ενεργής όρασης**. Αυτό σημαίνει ότι είμαστε σε θέση να επικεντρώνουμε την προσοχή μας σε κάποια συγκεκριμένη περιοχή του οπτικού μας πεδίου, μετακινώντας τα μάτια ή το κεφάλι μας ή απλά δίνοντας μεγαλύτερη προσοχή σε κάποιο συγκεκριμένο σημείο. Ποια είναι όμως τα πλεονεκτήματα της ενεργής όρασης σε σχέση με την παθητική όπου οι

οπτικοί αισθητήρες βρίσκονται σε προκαθορισμένη θέση και είναι αναγκασμένοι να εποπτεύουν με την ίδια βαρύτητα κάθε περιοχή μιας εικόνας;

1. Περιοχές μιας σκηνής που δεν είναι εύκολα ανιχνεύσιμες από έναν απλό παθητικό αισθητήρα είναι δυνατόν να ανιχνευθούν από μία κινούμενη συσκευή, όπως για παράδειγμα στους ανθρώπους όπου η κίνηση των ματιών και του κεφαλιού μας δίνει μια πανοραμική άποψη του χώρου.
2. Στρέφοντας την προσοχή σε κάποιες συγκεκριμένες μικρές περιοχές που είναι πιο σημαντικές από τις υπόλοιπες μπορούμε να αποφύγουμε την περιττή προσπάθεια του να αναλύουμε κάθε φορά την συνολική εικόνα, όπως για παράδειγμα όταν ο άνθρωπος επιχειρεί να πιάσει ένα αντικείμενο επικεντρώνεται σ' αυτό και πολύ δύσκολα αποπροσανατολίζεται έως ότου πετύχει τον σκοπό του.

Συμπερασματικά μπορούμε να πούμε ότι η ενεργός όραση είναι μια διαδικασία καθοδηγούμενη από στόχους (task-driven) σε αντίθεση από την παθητική. Το ρομπότ έχοντας αποθηκεύσει έναν συγκεκριμένο στόχο μπορεί μέσω ενός ενεργού οπτικού αισθητήρα, να επιλέξει άμεσα μέσα από όλη την διαθέσιμη πληροφορία μόνο τα κομμάτια εκείνα που είναι σε απευθείας συνάρτηση με τον στόχο ή την λύση ενός προβλήματος. Αντιθέτως ένα παθητικό σύστημα όρασης θα προσπαθήσει να κατασκευάσει μια σφαιρική εικόνα πριν προχωρήσει στην λήψη αποφάσεων, και για αυτό μπορεί να θεωρηθεί ότι είναι καθοδηγούμενο από τα δεδομένα (data - driven). Στην εικόνα 5.1 εμφανίζεται ένα παράδειγμα ενεργών οπτικών αισθητήρων που αναπτύχθηκαν στο Πανεπιστήμιο της Οχφόρδης.



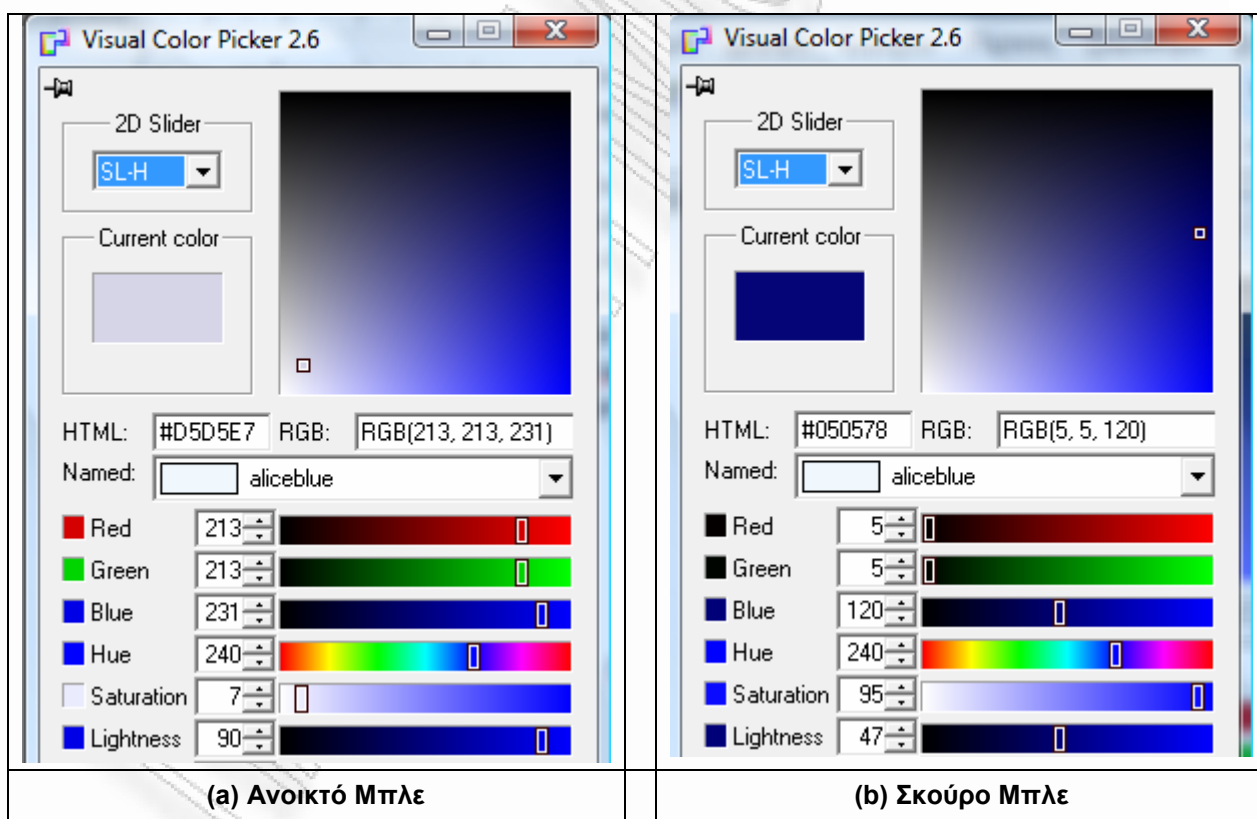
Εικόνα 5.1: Η σειρά ενεργών ρομποτικών κεφαλών Yorick: (a) Το αρχικό μοντέλο που χρησιμοποιείται για παρακολούθηση, (b) Το Yorick 8-11 που τοποθετήθηκε σε ένα όχημα και χρησιμοποιήθηκε για την πλοήγηση του, (c) Το Yorick 55C που είναι αρκετά μικρό ώστε να προσαρμοσθεί σε έναν ρομποτικό βραχίονα και χρησιμοποιείται για την απομακρυσμένη εκτέλεση δραστηριοτήτων από το ρομπότ.

Στο σύστημα πλοήγησης που έχουμε αναπτύξει, χρησιμοποιούμε σε κάποιο βαθμό τα χαρακτηριστικά της ενεργούς ψηφιακής όρασης με την έννοια ότι από την συνολική εικόνα που λαμβάνουμε από τον οπτικό αισθητήρα (κάμερα) μας ενδιαφέρει μόνο η περιοχή όπου βρίσκεται το ίδιο το ρομπότ. Αφού καταφέρουμε δηλαδή να το εντοπίσουμε, επεξεργαζόμαστε τις πληροφορίες της συγκεκριμένης περιοχής της εικόνας και στην συνέχεια του επιστρέφουμε εντολές πλοήγησης.

5.3.1 Η τεχνική Επεξεργασίας Εικόνας που ακολουθήθηκε

Βασικό χαρακτηριστικό της μεθόδου μας είναι όπως έχουμε πει ο εντοπισμός του AmigoBot στον χώρο με βάση τις πληροφορίες που λαμβάνουμε από την δικτυακή κάμερα. Η διαδικασία αυτή ενεργοποιείται όταν ο χρήστης επιλέξει μέσω της διεπαφής χρήστη την εντολή “Draw Path” προκειμένου να χαράξει μία νέα πορεία για το ρομπότ. Τότε λαμβάνεται μία εικόνα (frame) από αυτές που μεταδίδει η κάμερα και αποθηκεύεται προσωρινά στον δίσκο σε μορφή αρχείου PNG ώστε να μπορέσουμε να την επεξεργαστούμε.

Για την επεξεργασία της εικόνας αναλύουμε την αποθηκευμένη εικόνα εικονοστοιχείο (pixel) προς εικονοστοιχείο, με βάση τον **χρωματικό χώρο HSV** (Hue – απόχρωση, Saturation – κορεσμός, Value - τιμή). Προτιμήσαμε να χρησιμοποιήσουμε αυτή την πληροφορία παρά το συνηθισμένο **πρότυπο RGB** (Red – Green – Blue), καθώς το πρώτο διακρίνει ξεκάθαρα την απόχρωση από τις άλλες μεταβλητές κάτι που δεν γίνεται με το RGB. Για παράδειγμα με το RGB είναι πολύ δύσκολο να περιγράψει κανείς όλες τις αποχρώσεις ενός χρώματος. Αντιθέτως με το πρότυπο HSV μπορούμε πιο εύκολα να χαρακτηρίσουμε ένα pixel κόκκινο άσχετα από τον κορεσμό ή την φωτεινότητα. Στην εικόνα 5.2 φαίνεται η σταθερότητα και μεγαλύτερη αξιοπιστία του προτύπου HSV σε σχέση με το RGB. Στην συνέχεια παρουσιάζουμε τα βασικά χαρακτηριστικά του χρωματικού χώρου HSV.



Εικόνα 5.2: Οι τιμές RGB και HSV για (a) Ανοικτό Μπλε και (b) Σκούρο Μπλε. Είναι ξεκάθαρο ότι η τιμή Hue του HSV μπορεί να χρησιμοποιηθεί πολύ πιο ασφαλώς για τον χαρακτηρισμό ενός χρώματος αφού όπως βλέπουμε είναι η μόνη παράμετρος που παραμένει σταθερή άσχετα από τον κορεσμό ή την φωτεινότητα.

Ο χρωματικός χώρος HSV

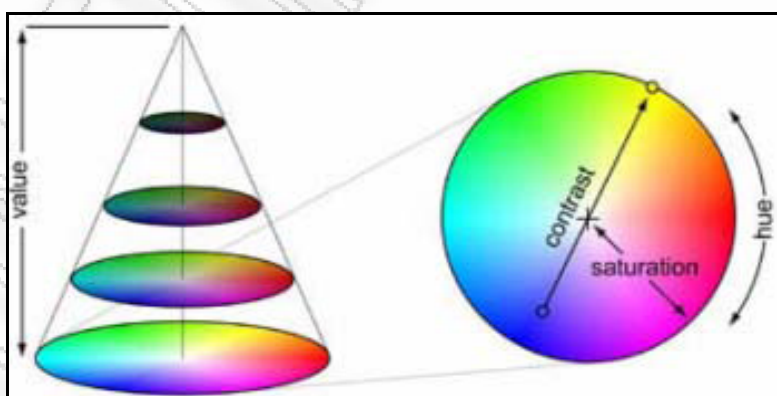
Στα πλαίσια του χρωματικού χώρου HSV, όλα τα χρώματα χαρακτηρίζονται από τρεις παραμέτρους:

1. **Την Απόχρωση** (Hue)
2. **Τον Κορεσμό** (Saturation) και
3. **Την Τιμή** (Value) που έχει σχέση με την φωτεινότητα

Η απόχρωση είναι το σημαντικότερο στοιχείο και κατά κάποιο τρόπο χαρακτηρίζει την βαφική ύλη του χρώματος. Για παράδειγμα το κόκκινο, πράσινο, μπλε και κίτρινο είναι αποχρώσεις, ενώ το ανοικτό μπλε ή το σκούρο μπλε έχουν παρόμοια τιμή απόχρωσης. Ο κορεσμός περιγράφει την ποσότητα του χρώματος ή με άλλα λόγια το πόσο έντονο είναι. Όταν ο κορεσμός έχει τιμή μηδέν το χρώμα είναι μία απόχρωση του γκρι. Τέλος υπάρχει και μία τιμή ή οποία πολλές φορές αναφέρεται και σαν φωτεινότητα. Αν έχει τιμή μηδέν το χρώμα είναι μαύρο ανεξάρτητα από την απόχρωση ή τον κορεσμό. Αν πάλι πάρει την μέγιστη τιμή της το χρώμα θα είναι σε κάθε περίπτωση το λευκό.

Το γεγονός αυτό οδηγεί σε μερική ανακρίβεια του προτύπου HSV όταν αναφερόμαστε σε οριακές τιμές. Για παράδειγμα όταν η φωτεινότητα είναι οριακή, η απόχρωση και ο κορεσμός είναι απροσδιόριστα, ενώ όταν ο κορεσμός είναι μηδέν η απόχρωση είναι και πάλι απροσδιόριστη. Αυτό σημαίνει ότι το μαύρο και το λευκό μπορούν να έχουν οποιαδήποτε τιμή απόχρωσης και κορεσμού, ενώ το γκρι μπορεί να έχει οποιαδήποτε τιμή απόχρωσης.

Κατά την επεξεργασία των HSV χαρακτηριστικών ενός χρώματος στον υπολογιστή, οι τρεις επιμέρους παράμετροι (Hue, Saturation, Value) αναπαριστώνται η κάθε μία από ένα byte με εύρος τιμών από 0 έως 255. Οι ζωνρές αποχρώσεις του κόκκινου, κίτρινου, πράσινου και μπλε έχουν όλες τιμή κορεσμού 255 και τιμή απόχρωσης αντιστοίχως: 0 ή 255, 43, 85 και 170. Εδώ παρατηρείται άλλη μία ανακρίβεια καθώς το κόκκινο μπορεί να έχει τιμή απόχρωσης είτε 0 είτε 255. Αυτό συμβαίνει διότι το εύρος των τιμών της απόχρωσης είναι κυκλικό και γύρω από το κόκκινο. Οι διακυμάνσεις του χρωματικού χώρου HSV φαίνονται καλύτερα στην εικόνα 5.3.



Εικόνα 5.3: Ο χρωματικός χώρος HSV και οι διακυμάνσεις του ανάλογα με την μεταβολή των παραμέτρων του. Βλέπουμε ότι το κόκκινο μπορεί να έχει τιμή απόχρωσης είτε 0 είτε 255 εξαιτίας του κυκλικού εύρους τιμών της απόχρωσης (Hue)

5.3.2 Μέθοδος εντοπισμού του AmigoBot

Αφού αναλύσαμε στην προηγούμενη παράγραφο την τεχνική επεξεργασίας εικόνας που χρησιμοποιήσαμε θα περιγράψουμε στην παράγραφο αυτή τον τρόπο με τον οποίο πετυχαίνουμε τον εντοπισμό του AmigoBot.

Όπως αναφέραμε και στην προηγούμενη παράγραφο η διαδικασία εντοπισμού του AmigoBot είναι το πρώτο βήμα για την διαδικασία της χάραξης νέας πορείας και ενεργοποιείται όταν ο χρήστης επιλέξει μέσω της διεπαφής χρήστη την εντολή “Draw Path”. Τότε συλλαμβάνεται το τρέχον καρέ από αυτά που μεταδίδονται από την κάμερα και εμφανίζονται στην διεπαφή χρήστη, και αποθηκεύεται στον δίσκο σε μορφή PNG (Portable Network Graphics), προκειμένου να γίνει περαιτέρω επεξεργασία του.

Παρόλο που υπάρχουν πολλά διαφορετικά πρότυπα για αρχεία εικόνων, π.χ. bitmap, ή jpg κ.τ.λ., επιλέξαμε το PNG γιατί είναι πρότυπο που δεν επιφέρει απώλειες. Αντιθέτως μία προτυποποίηση όπως του JPG που επιφέρει απώλειες πληροφορίας, μπορεί να μην παρουσιάζει αισθητά αποτελέσματα στο ανθρώπινο μάτι αλλά είναι πολύ πιθανό να οδηγήσει τον αλγόριθμο εντοπισμού του ρομπότ σε λανθασμένα αποτελέσματα.

Εφόσον έχουμε αποθηκεύσει την εικόνα στο δίσκο τα βήματα που ακολουθούν είναι τα εξής:

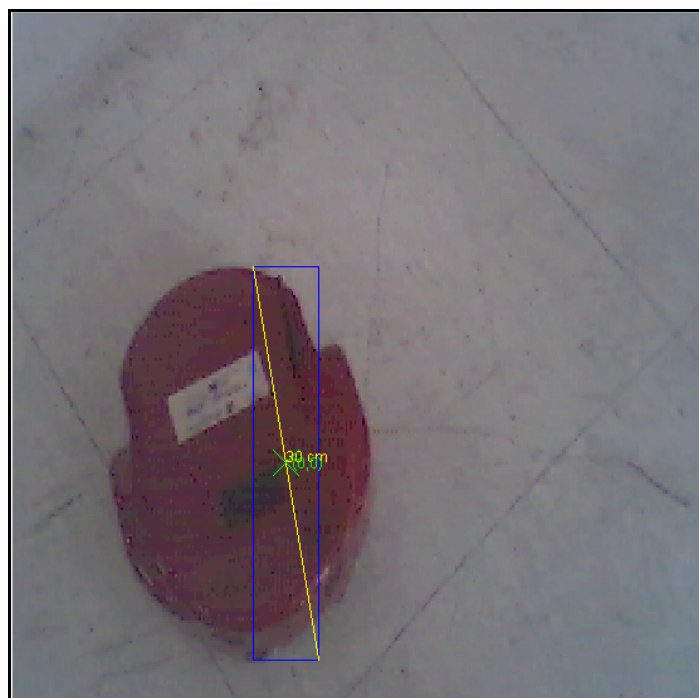
1. Ανάγνωση της εικόνας από τον δίσκο
2. Σάρωση της εικόνας pixel προς pixel και υπολογισμός στατιστικών μεμονωμένα αλλά και συγκεντρωτικά για περιοχές από Pixel
3. Ενημέρωση (απεικόνιση) της εικόνας με πληροφορίες όπως είναι η θέση του ρομπότ και η εκτιμώμενη σχετική κλίμακα των διαστάσεων του πραγματικού χώρου σε σχέση με τις διαστάσεις που απεικονίζονται στην εικόνα.
4. Στην συνέχεια, ο χρήστης εφόσον αποδέχεται την κλίμακα που υπολόγισε η εφαρμογή μπορεί να προχωρήσει στην χάραξη τροχιάς για το AmigoBot. Σε αντίθετη περίπτωση έχει την δυνατότητα να προσπαθήσει να ορίσει χειροκίνητα την κλίμακα.

Ένα παράδειγμα της εικόνας που εμφανίζεται στην οθόνη του χρήστη στην περίπτωση επιτυχούς εντοπισμού του AmigoBot φαίνεται στην εικόνα 5.4

1. Ανάγνωση της εικόνας από τον δίσκο

Η κλάση Java που υλοποιεί την επεξεργασία και γενικότερα τον χειρισμό της αποθηκευμένης εικόνας είναι η robotDetection. Για να μπορέσουμε λοιπόν να φορτώσουμε το αρχείο στην μνήμη καλούμε την έτοιμη βιβλιοθήκη της Java, javax.imageio.ImageIO η οποία αντιλαμβάνεται το είδος και την μορφή κωδικοποίησης του αρχείου και επιστρέφει ένα java.awt.BufferedImage.

Παρόλο που η Java υποστηρίζει πολλά διαφορετικές υλοποιήσεις για την κλάση *Image*, στην εφαρμογή μας χρησιμοποιούμε αποκλειστικά τα BufferedImages διότι μας εγγυούνται ότι έτσι φορτώνεται και είναι διαθέσιμη όλη η πληροφορία της εικόνας. Άλλες υλοποιήσεις είναι πιο αργές και δύσχρηστες και ακατάλληλες για τις ανάγκες της συγκεκριμένης εφαρμογής.



Εικόνα 5.4: Παράδειγμα επιτυχούς εντοπισμού του AmigoBot. Στο συγκεκριμένο παράδειγμα έχει επιτευχθεί και ο ορθός ορισμός της κλίμακας, αφού η απόσταση των 30 cm του πραγματικού χώρου έχει αποδοθεί σωστά στο μήκος του AmigoBot.τ

Η κλάση `robotDetection`, δημιουργεί λοιπόν το κατάλληλο αντικείμενο `file` και το φορτώνει στην μνήμη μέσω της `ImageIO.read()`. Σε περίπτωση που η τελευταία ρουτίνα δεν μπορεί να εντοπίσει το αρχείο ή να συμπεράνει την μορφή κωδικοποίησης, αποτυγχάνει και παράλληλα ρίχνει μία εξαίρεση (`exception`). Τέλος η `robotDetection` πραγματοποιεί άλλη μια πολύ σημαντική λειτουργία μέσω της ρουτίνας `maslab.util.ImageUtil.RGBtoHSV()`. Επειδή το `image` που δημιουργείται με την `ImageIO.read()` περιγράφεται από τον χρωματικό χώρο RGB, η ρουτίνα `maslab.util.ImageUtil.RGBtoHSV()` είναι αυτή που πραγματοποιεί τον μετασχηματισμό του στον χρωματικό χώρο HSV τοποθετώντας αντίστοιχα:

1. την απόχρωση (Hue) στο κόκκινο κανάλι,
2. τον κορεσμό (Saturation) στο πράσινο κανάλι και
3. την τιμή (Value) ή φωτεινότητα στο μπλε κανάλι

2. Σάρωση της εικόνας pixel προς pixel και υπολογισμός στατιστικών

Κατά την φάση αυτή πραγματοποιείται ο εντοπισμός του ρομπότ στον χώρο με βάση το έντονο κόκκινο χρώμα του όπως έχουμε ξαναπεί, που το διακρίνει ξεκάθαρα από το περιβάλλον του. Ο υπολογισμός αυτός καθώς και η ενημέρωση της εικόνας με πληροφορίες που θα χρησιμοποιήσει στην συνέχεια ο χρήστης για την χάραξη νέας τροχιάς για το AmigoBot, πραγματοποιείται μέσω της ρουτίνας `paintDetectedRobot()`. Αυτή σε πρώτη φάση καλεί την `find_red_Robot()` η οποία πραγματοποιεί μία γραμμική σάρωση της εικόνας pixel προς pixel προσπαθώντας να εξακριβώσει ποιο από αυτά είναι κόκκινο. Κάθε φορά που ένα pixel αναγνωρίζεται ως κόκκινο ενημερώνονται τα στατιστικά.

Ο χαρακτηρισμός ενός pixel ως κόκκινου πραγματοποιείται μέσω της ρουτίνας **isRed(pixel)**. Στην ρουτίνα αυτή εισάγεται ένας ακέραιος μήκους 32 bit ο οποίος αναλύεται ως εξής:

8 bit	8 bit	8 bit	8 bit
xxxxxxx	hhhhhhh	sssssss	vvvvvvv
Αχρησιμοποίητα	Hue	Saturation	Value
-	Δεξιά Ολίσθηση κατά 16 bit	Δεξιά Ολίσθηση κατά 8 bit	-
Αφου γίνει ολίσθηση πραγματοποιούμε μια ψηφιακή πράξη bitwise AND ώστε να συλλέξουμε τα 8 τελευταία bit		xxxxxxxxhhhhhhssssssvvvvvv &0000000000000000000000000000000011111111	
val = 00000000000000000000000000000000vvvvvvvv			

Πίνακας 5.5: Η διαδικασία εξαγωγής των τιμών HSV για ένα pixel

Όπως βλέπουμε τα 8 τελευταία bit μας δίνουν την τιμή της Φωτεινότητας για το συγκεκριμένο pixel, τα αμέσως 8 προηγούμενα την τιμή του Κορεσμού και τα 8 προηγούμενα του κορεσμού, την τιμή της Απόχρωσης. ΜΕ ΔΙΑΔΟΧΙΚΕΣ ΔΕΞΙΕΣ ΟΛΙΣΘΗΣΕΙΣ ΤΟΥ BYTE ΑΝΤΙΣΤΟΙΧΑ ΚΑΤΑ 8 ΚΑΙ 16 ΚΑΙ ΕΚΤΕΛΕΣΗ ΤΗΣ ΛΟΓΙΚΗΣ ΠΡΑΞΗΣ AND ΜΕ 0xFF ΛΑΜΒΑΝΟΥΜΕ ΤΙΣ ΤΙΜΕΣ ΤΟΥ ΚΟΡΕΣΜΟΥ ΚΑΙ ΤΗΣ ΑΠΟΧΡΩΣΗΣ.

Στην συνέχεια έχοντας ορίσει μετά από πειραματισμό με διάφορες τιμές, ένα εύρος τιμών HSV για τις οποίες ένα pixel χαρακτηρίζεται ως κόκκινο αποφασίζουμε αν τελικά θα χαρακτηρίσουμε το συγκεκριμένο pixel κόκκινο ή όχι. Το εύρος των αποδεκτών τιμών φαίνεται παρακάτω.

Εύρος αποδεκτών τιμών HSV για τον χαρακτηρισμό ενός pixel σαν κόκκινο	
RED_HUE_MAX	0
RED_HUE_MIN	220
RED_SAT_MAX	220
RED_SAT_MIN	60
RED_VAL_MAX	170
RED_VAL_MIN	30

Πίνακας 5.6: Εύρος των αποδεκτών τιμών HSV που οφείλει να διαθέτει ένα pixel προκειμένου να χαρακτηριστεί σαν κόκκινο.

Θεωρούμε ότι έχουμε πετύχει εντοπισμό του AmigoBot όταν βρούμε περιοχή 200 pixel στην εικόνα που έχουμε χαρακτηρίσει ως κόκκινα. Στο σημείο αυτό η αναζήτηση και σάρωση της εικόνας σταματάει και προχωρούμε στην ενημέρωση της εικόνας με τις κατάλληλες πληροφορίες (εικόνα 5.4). Το όριο των της περιοχής 200 pixel το καθορίσαμε μετά από πειραματισμό και με βάση τις δυνατότητες της κάμερας που χρησιμοποιήσαμε κατά την ανάπτυξη της εφαρμογής. Τόσο το όριο αυτό όσο και το εύρος των αποδεκτών τιμών HSV είναι δυνατόν να τροποποιηθούν προκειμένου να πετύχουμε μεγαλύτερη ακρίβεια στον εντοπισμό του AmigoBot.

3. Ενημέρωση (απεικόνιση) των Πληροφοριών στην Εικόνα

Εφόσον μπορέσαμε να εντοπίσουμε την θέση του AmigoBot στην αποθηκευμένη εικόνα θα πρέπει στην συνέχεια να την ενημερώσουμε με τα αποτελέσματα της επεξεργασίας, ώστε να καταστεί χρήσιμη στον χρήστη για τους σκοπούς της πλοήγησης του ρομπότ.

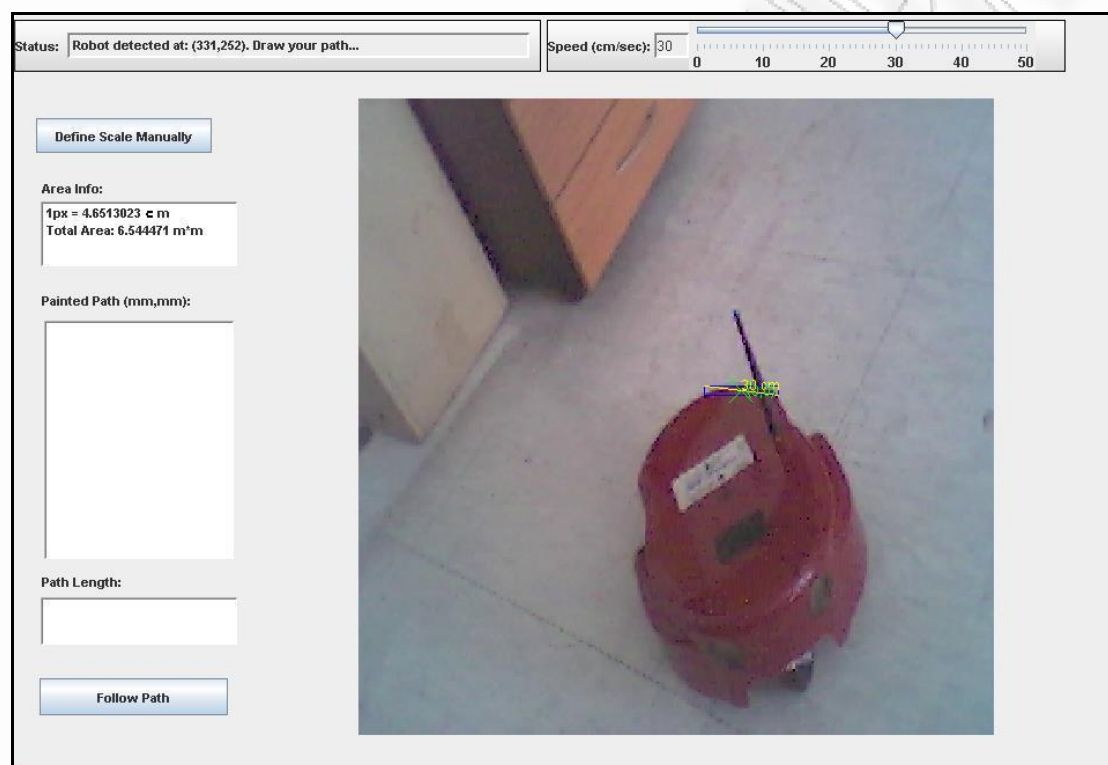
Η απεικόνιση των στατιστικών στην εικόνα πραγματοποιείται μέσα στην ρουτίνα `paintDetectedRobot`. Αρχικά δημιουργούμε ένα αντικείμενο `Graphics2D` και το συσχετίζουμε με το `JPanel` πάνω στο οποίο τοποθετείται το `BufferedImage` στο οποίο επιθυμούμε να σχεδιάσουμε. Το τελικό αποτέλεσμα του εντοπισμού του AmigoBot παρουσιάζεται στον χρήστη μέσω μιας εικόνας που θα μοιάζει με την εικόνα 5.4. Ο επόμενος πίνακας συνοψίζει την σημασία των στοιχείων που απεικονίζονται στην εικόνα αυτή και τις ρουτίνες που καλούμε για να τα σχεδιάσουμε.

Χαρακτηριστικό	Σημασία	Ρουτίνες
Μπλε Παραλληλόγραμμα	Σκοπεύει στο να καταδείξει την περιοχή που εντοπίστηκε ως κόκκινη. Κανονικά πρέπει να περικλείει το AmigoBot	<ul style="list-style-type: none"> ▪ <code>s.setColor(Color.blue);</code> ▪ <code>s.drawRect(it.x_min,it.y_min, it.x_max - it.x_min, it.y_max - it.y_min);</code>
Κίτρινη Διαγώνιος	Το μήκος της πρέπει κανονικά να ταυτίζεται με το μήκος του AmigoBot δηλαδή 30cm πραγματικών διαστάσεων. Χρησιμοποιείται για την εξαγωγή της κλίμακας των αποστάσεων που πρέπει να χρησιμοποιηθεί.	<ul style="list-style-type: none"> ▪ <code>s.setColor(Color.YELLOW);</code> ▪ <code>s.drawLine(it.x_min, it.y_min, it.x_max, it.y_max);</code> ▪ <code>s.drawString("30cm", it.x_min + (it.x_max - it.x_min) / 2, it.y_min + (it.y_max - it.y_min) / 2);</code>
Πράσινος Σταυρός	Καταδεικνύει το κέντρο της κόκκινης περιοχής και κανονικά πρέπει να απεικονίζεται στο κέντρο της επάνω επιφάνειας του AmigoBot	<ul style="list-style-type: none"> ▪ <code>s.setColor(Color.green);</code> ▪ <code>s.drawLine(it.x_position - 10, it.y_position - 10, it.x_position + 10, it.y_position + 10);</code> ▪ <code>s.drawLine(it.x_position - 10, it.y_position + 10, it.x_position + 10, it.y_position - 10);</code>
<code>it.x_min - t.x_max</code>	Η Java χρησιμοποιεί σύστημα συντεταγμένων (x,y) με την συντεταγμένη αρχής (0,0) στην πάνω αριστερή γωνία. Οι μεταβλητές αυτές αντιστοιχούν στην ελάχιστη και μέγιστη τετμημένη των κορυφών που ορίζουν το μπλε παραλληλόγραμμα.	
<code>it.y_min - it.y_max</code>	Ορίζουν αντίστοιχα την ελάχιστη και μέγιστη τεταγμένη των σημείων που ορίζουν το μπλε παραλληλόγραμμα	
<code>it.x_position</code> <code>it.y_position</code>	- Τετμημένη - Τεταγμένη του σημείου που αντιστοιχεί στο κέντρο του μπλε παραλληλογράμμου και στο μέσο της κίτρινης διαγώνιου -	

Πίνακας 5.7: Τα στοιχεία που απεικονίζονται στην επεξεργασμένη εικόνα μετά τον εντοπισμό του AmigoBot

5.3.3 Χάραξη Τροχιάς για το AmigoBot και Αναγωγή των αποστάσεων σε Διαστάσεις του Πραγματικού Χώρου

Προκειμένου να περιγράψουμε τον τρόπο με τον οποίο ο χρήστης προσδιορίζει μία νέα πορεία για το AmigoBot θα παρουσιάσουμε στην συνέχεια ένα ολοκληρωμένο παράδειγμα πλοήγησης του. Όπως είπαμε και προηγουμένως ο επιτυχής εντοπισμός του ρομπότ έχει σαν αποτέλεσμα την εμφάνιση στην οθόνη του χρήστη μιας εικόνας παρόμοιας με την εικόνα 5.4. Θεωρούμε λοιπόν αρχικά ότι σε κάποια άλλη εκτέλεση της ρουτίνας εντοπισμού, προέκυψε η εικόνα 5.8.



Εικόνα 5.8: Παράδειγμα ανεπιτυχούς εντοπισμού του AmigoBot. Στο συγκεκριμένο παράδειγμα παρόλο που το AmigoBot έχει εντοπιστεί (πράσινος σταυρός) δεν έχει επιτευχθεί ορθός ορισμός της κλίμακας, αφού η απόσταση των 30 cm του πραγματικού χώρου δεν έχει αποδοθεί σωστά στο μήκος του ρομπότ.

Στην περίπτωση αυτή το AmigoBot έχει και πάλι εντοπιστεί, μιας και ο πράσινος σταυρός έχει σχεδιαστεί εντός της επιφανείας του, όμως η κλίμακα που προσδιορίστηκε αυτόματα είναι λανθασμένη σε μεγάλο βαθμό. Αυτό το αντιλαμβανόμαστε καταρχήν οπτικά αφού το μπλε παραλληλόγραμμα καλύπτει ένα πολύ μικρό μέρος του AmigoBot ενώ το ίδιο συμβαίνει και με την κίτρινη διαγώνιο που κανονικά θα έπρεπε να διατρέχει όλο το μήκος του ρομπότ.

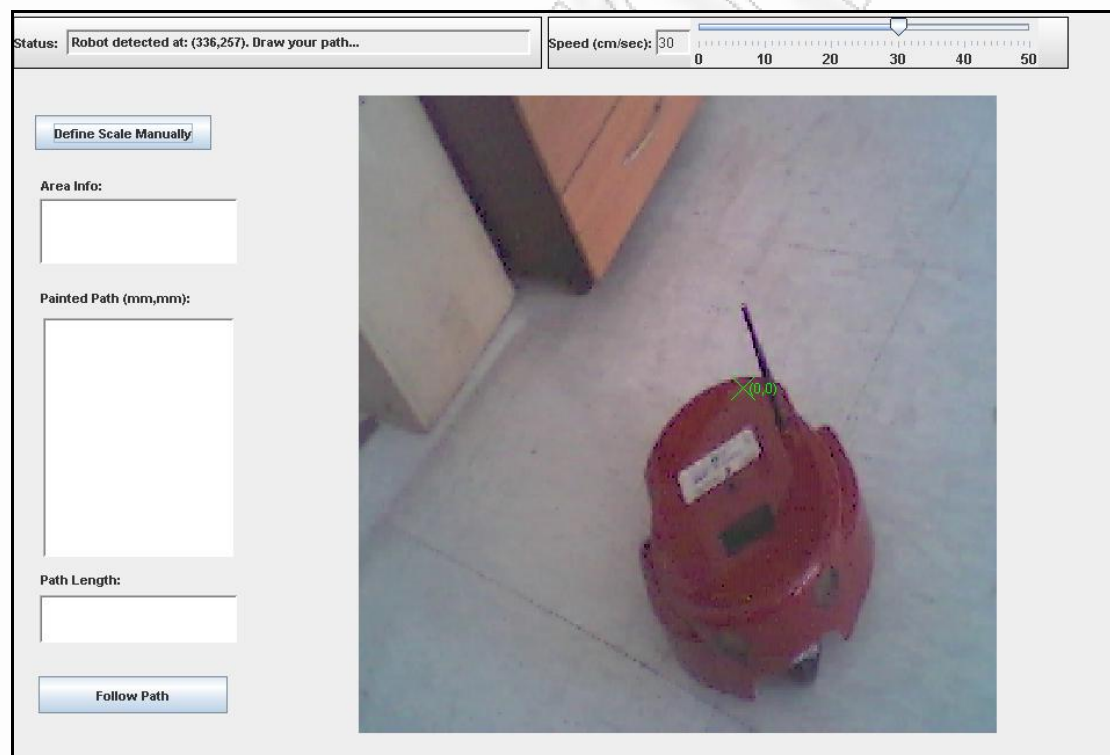
Επιπλέον το αντιλαμβανόμαστε και από τις πληροφορίες που αναγράφονται στην περιοχή "Area Info". Εδώ πληροφορούμαστε ότι μετά την διαδικασία του αυτόματου εντοπισμού βρέθηκε ότι κάθε εικονοστοιχείο (pixel) της εικόνας αντιστοιχεί σε 4.651cm απόστασης του πραγματικού χώρου και ότι όλη η περιοχή που φαίνεται στην εικόνα αντιστοιχεί σε 6.5444mm² κάτι το οποίο απέχει πολύ από την πραγματικότητα.

Στο σημείο αυτό οφείλουμε να περιγράψουμε τον **τρόπο με τον οποίο το σύστημα μας υπολογίζει την κλίμακα** των αποστάσεων της εικόνας σε σχέση με αυτές του πραγματικού

χώρου. Όπως αναφέρουμε και στον πίνακα 5.7 η κίτρινη διαγώνιος στην ιδανική περίπτωση θα πρέπει να καλύπτει απόσταση ίση ακριβώς με το μήκος του AmigoBot. Η εφαρμογή μας σε κάθε περίπτωση θεωρεί ότι αντιστοιχεί σε 30cm διαστάσεων πραγματικού χώρου.

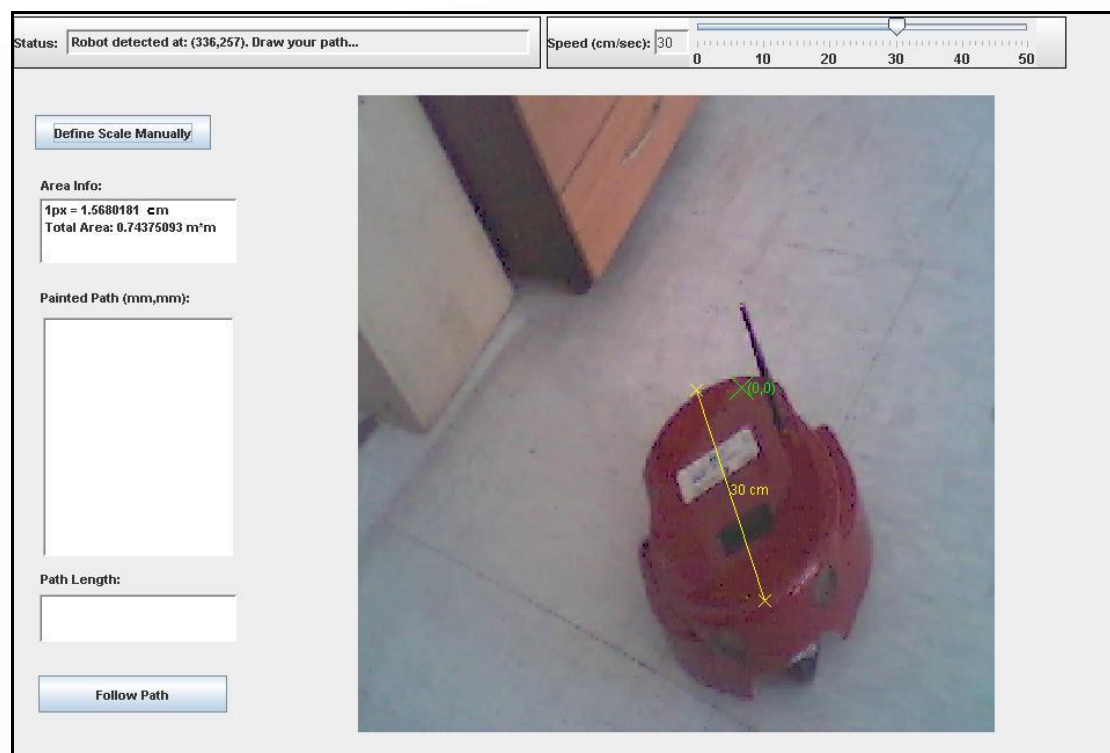
Διαιρώντας λοιπόν την απόσταση αυτή (30cm) με το πλήθος των pixels που υπολογίζουμε κάθε φορά ότι συνθέτουν την κίτρινη διαγώνιο, εξάγουμε την απόσταση πραγματικού χώρου που αντιστοιχεί σε κάθε pixel της εικόνας. Στην συνέχεια είναι εύκολο να υπολογίσουμε και το συνολικό εμβαδόν πραγματικού χώρου που απεικονίζεται στην οθόνη του χρήστη. Οι πληροφορίες αυτές εμφανίζονται στην περιοχή “Area Info” του γραφικού περιβάλλοντος της εφαρμογής.

Προκειμένου η διαδικασία πλοήγησης στην περίπτωση αυτή να μην διακοπεί, δίνουμε την δυνατότητα στον χρήστη να επέμβει και να ορίσει χειροκίνητα την σωστή κλίμακα. Αυτό που οφείλει να κάνει ο χρήστης είναι να σχεδιάσει μόνος του την κίτρινη διαγώνιο ώστε να ισούται με το μήκος του AmigoBot για το οποίο γνωρίζουμε εκ των προτέρων ότι ισούται με 30cm. Επιλέγοντας λοιπόν “Define Scale Manually” η εικόνα καθαρίζει από τα προηγούμενα στατιστικά και παραμένει μόνο ο πράσινος σταυρός (εικόνα 5.9)



Εικόνα 5.9: Ο χρήστης έχει αποφασίσει να καθορίσει χειροκίνητα την κλίμακα. Η εικόνα καθαρίζει από τις αυτόματα υπολογιζόμενες πληροφορίες για την κλίμακα και παραμένει μόνο ο πράσινο σταυρός.

Στην συνέχεια ο χρήστης πρέπει να σχεδιάσει καταλλήλως την κίτρινη διαγώνιο. Κάτι τέτοιο έχουμε πραγματοποιήσει και παρουσιάζουμε στην εικόνα 5.10.



Εικόνα 5.10: Ο χρήστης έχει ορίσει χειροκίνητα την κλίμακα. Πλέον η κίτρινη διαγώνιος αντιστοιχίζεται σωστά στο μήκος του AmigoBot (30 cm).

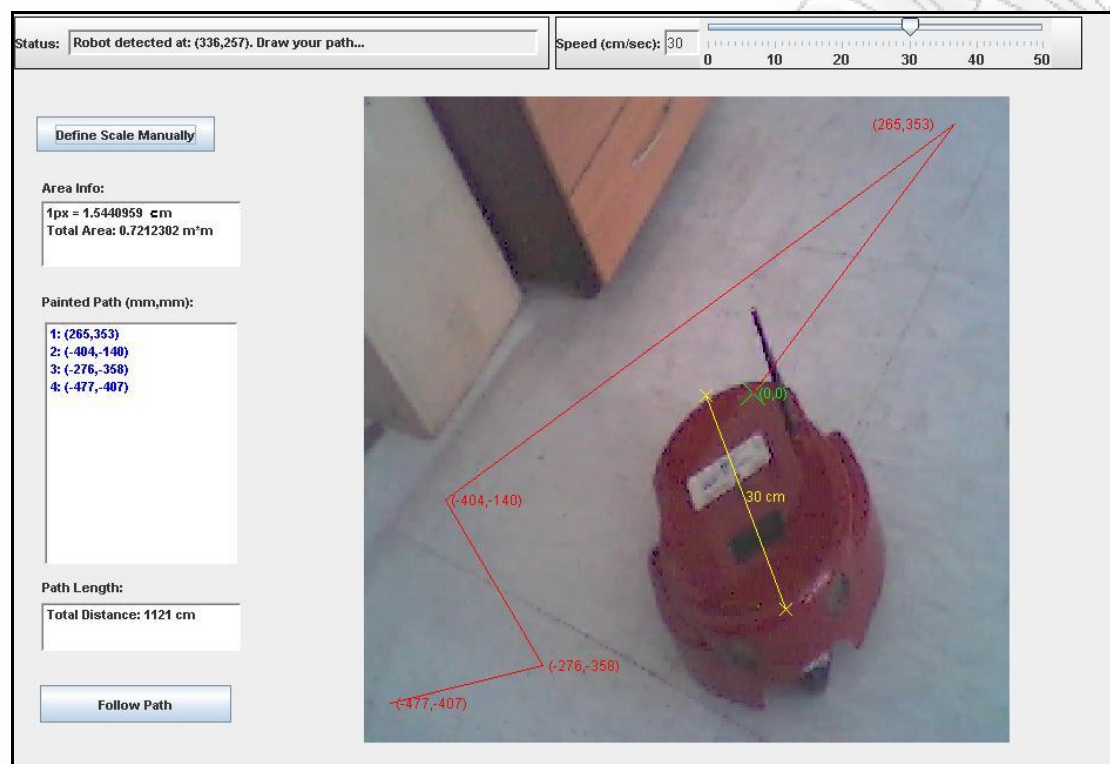
Τα στατιστικά έχουν ανανεωθεί και βλέπουμε ότι πλέον ανταποκρίνονται στην πραγματικότητα. Το κάθε pixel δηλαδή έχει αντιστοιχισθεί σε περίπου 1,57cm διαστάσεων πραγματικού χώρου ενώ η περιοχή του δαπέδου που εμφανίζεται στην οθόνη υπολογίζεται σωστά ότι καταλαμβάνει εμβαδόν ίσο με περίπου 0.74 m². Το σύστημα μας είναι εφοδιασμένο πλέον με σωστές πληροφορίες για τον χώρο και ο χρήστης μπορεί να χαράξει αξιόπιστα μία νέα πορεία για το AmigoBot (εικόνα 5.12).

Στην εικόνα 5.12 φαίνεται με τον καλύτερο τρόπο η **τεχνική των υποχρεωτικών σημείων** που ακολουθούμε για την καθοδήγηση του AmigoBot και για την οποία μιλήσαμε στην παράγραφο 5.2 όπου αναλύσαμε την μεθοδολογία μας. Στο συγκεκριμένο παράδειγμα έχουμε χαράξει μία νέα τροχιά για το ρομπότ μας. Οι πληροφορίες που συνθέτουν την πορεία αυτή παρουσιάζονται στον πίνακα 5.11.

Σειρά Διέλευσης	Συντεταγμένες Υποχρεωτικών Σημείων	
1	(265,353)	
2	(-404,-140)	
3	(-276,-358)	
4	(-477,-407)	
Αρχή των Αξόνων	(0,0)	Πράσινος Σταυρός
Μονάδα Μέτρησης	Χιλιοστόμετρα	

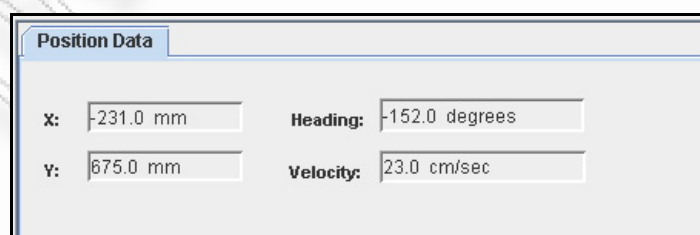
Πίνακας 5.11: Οι συντεταγμένες των υποχρεωτικών σημείων της νέας τροχιάς κίνησης του AmigoBot.

Το AmigoBot θα μεταβεί αρχικά στο σημείο (266,353) και αφού περάσει και από τα δύο ενδιάμεσα σημεία θα σταματήσει ξανά στο σημείο (- 477,- 407). Οι συντεταγμένες των σημείων αντιστοιχούν σε χιλιοστόμετρα απόστασης πραγματικού χώρου. **Σημαντικό πλεονέκτημα της μεθόδου χάραξης πορείας** που χρησιμοποιούμε στο σύστημα μας, είναι το ότι δεν είναι απαραίτητη η οποιαδήποτε γνώση της προηγούμενης θέσης του. Αυτό συμβαίνει διότι σε κάθε νέα χάραξη πορείας τίθεται σαν αρχή (0,0) των συντεταγμένων που θα χρησιμοποιηθούν για τον καθορισμό των υποχρεωτικών σημείων της τροχιάς το ίδιο το AmigoBot.



Εικόνα 5.12: Απεικόνιση της νέας τροχιάς που οφείλει να ακολουθήσει το AmigoBot. Ο χρήστης μπορεί να επιλέξει την ασφαλέστερη διαδρομή για την μετακίνηση του ρομπότ από ένα σημείο σε κάποιο άλλο φροντίζοντας να αποφύγει τα όποια εμπόδια.

Προκειμένου η διεπαφή χρήστη να είναι ολοκληρωμένη και φιλική, κατά την κίνηση του AmigoBot εμφανίζονται ποικίλες πληροφορίες όπως η ταχύτητα, η τρέχουσα θέση και ο προσανατολισμός του AmigoBot (εικόνα 5.13) καθώς και πληροφορίες για την τρέχουσα κατάσταση του ρομπότ (περιοχή Status, εικόνα 5.12). Τέλος τα υποχρεωτικά σημεία που έχει χαράξει ο χρήστης εμφανίζονται στην περιοχή “Painted Path” στα αριστερά της εικόνας.



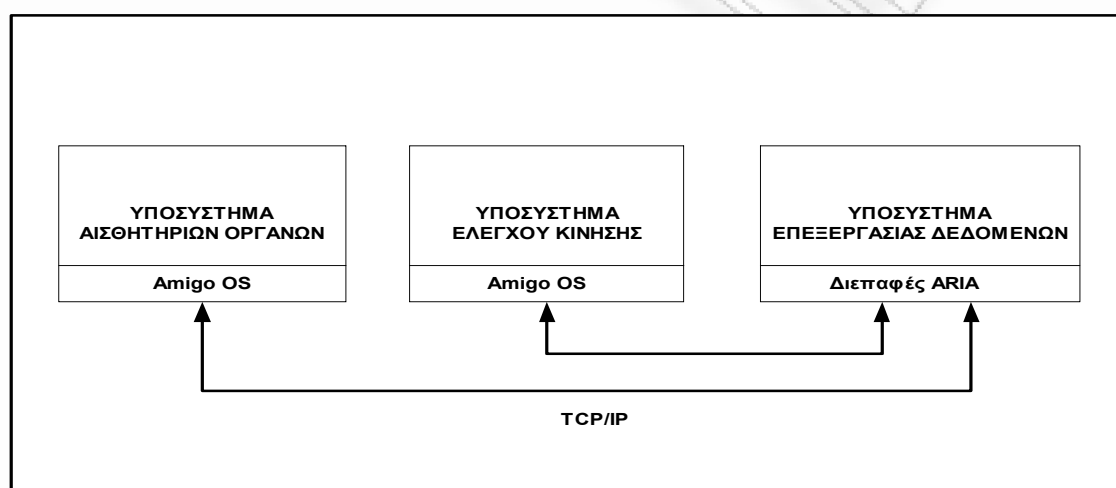
Εικόνα 5.13: Οι πληροφορίες για την κίνηση του AmigoBot εμφανίζονται δυναμικά στο γραφικό περιβάλλον διασύνδεσης χρήστη.

5.4 Υλοποίηση του προτεινόμενου Συστήματος

Σε γενικές γραμμές για να μπορέσει να λειτουργήσει αποτελεσματικά κάθε σύστημα πλοήγησης αυτοκινούμενων ρομπότ θα πρέπει να διαιρείται και να ολοκληρώνει τρία ξεχωριστά υποσυστήματα. Αυτά είναι:

1. Υποσύστημα Αισθητήριων Οργάνων
2. Υποσύστημα Επεξεργασίας των δεδομένων που λαμβάνονται από τα αισθητήρια όργανα
3. Υποσύστημα ελέγχου Κίνησης

Τα τρία αυτά υποσυστήματα θα πρέπει ασφαλώς να επικοινωνούν μεταξύ τους. Για έναν μηχανικό λογισμικού θεωρείται αυτόνοτο ότι θα πρέπει να υλοποιηθούν σαν ξεχωριστές υπομονάδες κατά την ανάπτυξη της εφαρμογής. Μία εικόνα της αρχιτεκτονικής του προτεινόμενου συστήματος σε υψηλό βαθμό αφαίρεσης εμφανίζεται στο Σχήμα 5.14.



Σχήμα 5.14: Η Αρχιτεκτονική του προτεινόμενου Συστήματος σε υψηλό επίπεδο αφαίρεσης

Όπως έχουμε αναφέρει στο κεφάλαιο 4, το λειτουργικό σύστημα του AmigoBot είναι αυτό το που διαχειρίζεται όλες τις λειτουργίες χαμηλού επιπέδου των διαφόρων υποσυστημάτων του ρομπότ. Είναι υπεύθυνο δηλαδή για λειτουργίες όπως η χρήση του σόναρ η ανάγνωση των αποκωδικοποιητών που βρίσκονται στις ρόδες, η ρύθμιση της ταχύτητας για την εκτέλεση ελιγμών, η αναπαραγωγή προαποθηκευμένων ήχων κ.ο.κ.. Παρολαυτά παρέχει ελεύθερα προγραμματιστικές διεπαφές για την ανάπτυξη εφαρμογών από εξωτερικούς χρήστες.

Όπως φαίνεται και στο σχήμα 5.1, η επικοινωνία της εφαρμογής μας με το λειτουργικό σύστημα του AmigoBot πραγματοποιείται μέσω TCP/IP. Το σύστημα μας λαμβάνει πληροφορίες από τα αισθητήρια όργανα του ρομπότ (σόναρ και οδομετρία) μέσω του AmigoOS, στην συνέχεια τις επεξεργάζεται μαζί με τις πληροφορίες από τις εικόνες που μεταδίδει η κάμερα και τελικά επιστρέφει τις επιθυμητές εντολές για την πλοήγηση του ρομπότ στο υποσύστημα ελέγχου κίνησης και πάλι μιλώντας με το AmigoOS. Ουσιαστικά δηλαδή η εφαρμογή μας λειτουργεί σαν εφαρμογή - πελάτης (client) της ρομποτικής πλατφόρμας που παίζει τον ρόλο του εξυπηρετητή (server) και δεν είναι αναγκαίο να γνωρίζει λεπτομέρειες χαμηλού επιπέδου όσον αφορά αυτήν.

5.4.1 Το περιβάλλον προγραμματιστικών Διεπαφών ARIA

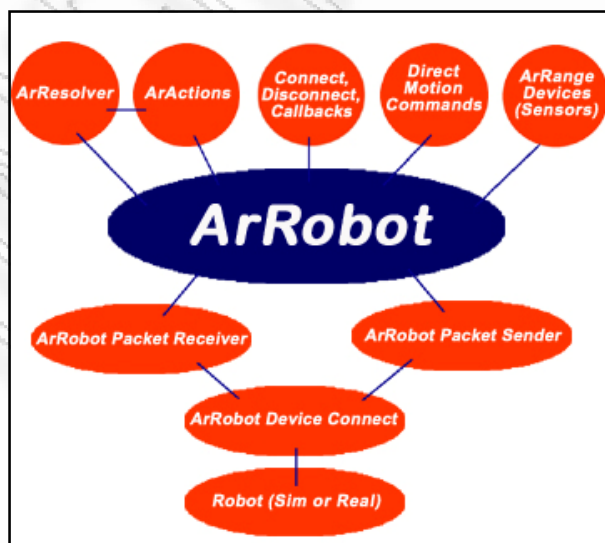
Τα παραμένω καθίστανται δυνατά με την χρήση των έτοιμων προγραμματιστικών διεπαφών του περιβάλλοντος της ARIA τις οποίες και καλούμε από την εφαρμογή μας. Εξαιτίας της σπουδαιότητας του συγκεκριμένου API θα το περιγράψουμε στην παράγραφο αυτή με μεγαλύτερη λεπτομέρεια [44]. Τέλος θα παρουσιάσουμε τις κλάσεις και ρουτίνες της ARIA που χρησιμοποιήσαμε στο σύστημα μας και ποιο σκοπό εξυπηρετεί κάθε μία από αυτές.

Το περιβάλλον της ARIA (Advanced Robotics Interface Application) έχει σχεδιαστεί για επαγγελματίες μηχανικούς λογισμικού. Είναι ένα ισχυρό αντικειμενοστραφές (Object Oriented) σύνολο διεπαφών για τα ρομπότ της οικογένειας MobileRobots, όπως είναι και το AmigoBot, και μπορεί να χρησιμοποιηθεί τόσο σε περιβάλλον Windows όσο και σε Linux. Προσφέρεται τόσο σε C++ όσο και σε Java.

Η ARIA έχει υλοποιηθεί εξολοκλήρου σύμφωνα με τα πρότυπα του αντικειμενοστραφούς προγραμματισμού. Επικοινωνεί με την εκάστοτε ρομποτική πλατφόρμα μέσω μίας σχέσης πελάτη εξυπηρετητή (client – server), είτε μέσω σειριακής σύνδεσης είτε μέσω σύνδεσης TCP/IP. Επιπλέον η ARIA είναι ευέλικτη. Για παράδειγμα είναι δυνατόν να υλοποιηθεί σαν διαδικασία μονού νήματος (single-threaded) ή και πολλαπλών νημάτων (multi-threaded).

Στο σύστημα που προτείνουμε υλοποιούμε διαδικασίες πολλαπλών νημάτων (multi-threaded). Τοποθετούμε δηλαδή σε ένα ξεχωριστό νήμα τον εξυπηρετητή (δλδ τις διαδικασίες που τρέχουν στην ρομποτική πλατφόρμα) και σε άλλον το περιβάλλον διεπαφής χρήστη, προκειμένου να αποφύγουμε φαινόμενα «παγώματος» της εφαρμογής μας και να την καταστήσουμε φιλικότερη προς τον χρήστη. Η δομή της ARIA φαίνεται στο σχήμα 5.2.

Συνήθως οι μηχανικοί λογισμικού χρησιμοποιούν τις διεπαφές υψηλού επιπέδου της ARIA προκειμένου να υλοποιήσουν επιχειρησιακή λογική και συμπεριφορά ή αλλιώς «ενέργειες» (actions). Επιπλέον οι βιβλιοθήκες της ARIA είναι δυνατόν να χρησιμοποιηθούν για την λήψη πληροφοριών από τα αισθητήρια όργανα και για την καθοδήγηση του AmigoBot στο να εκτελέσει συγκεκριμένες ενέργειες, για την αποστολή και λήψη εντολών ή απλά για τον έλεγχο της σειριακής θύρας επικοινωνίας. Τέλος το API της ARIA είναι σε θέση να υλοποιεί νηματικό προγραμματισμό τόσο μέσω της διαχείρισης νημάτων που υποστηρίζει το Linux όσο και τα Windows.



Σχήμα 5.15: Η αρχιτεκτονική του API της ARIA και οι σημαντικότερες κλάσεις του

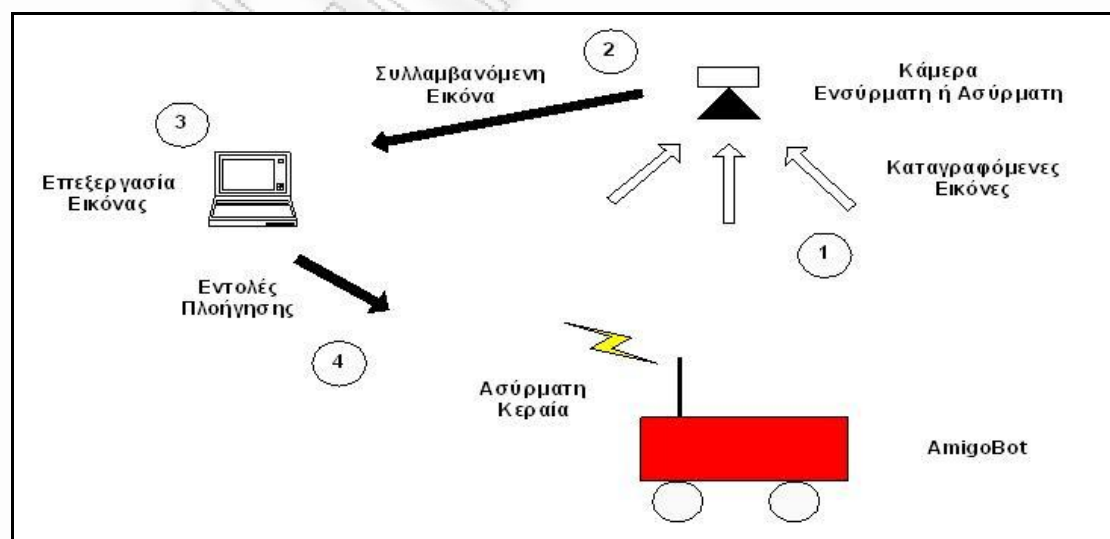
Τα βασικά χαρακτηριστικά του περιβάλλοντος προγραμματιστικών διεπαφών της ARIA, πολλά από τα οποία χρησιμοποιήσαμε και στην εφαρμογή μας, είναι τα παρακάτω:

- Δυναμικός έλεγχος της ταχύτητας και της κατεύθυνσης του ρομπότ
- Ολοκλήρωση του διαύλου Εισόδου/Εξόδου (I/O bus) με τον βραχίονα, τις ρόδες και τα άλλα εξαρτήματα του AmigoBot, γεγονός που επιτρέπει στις εφαρμογές πελάτες που θα χρησιμοποιήσουν την ARIA να ελέγξουν με ευκολία τα εν λόγω εξαρτήματα
- Ελέγχει όλες τις συσκευές που βρίσκονται εντός εμβέλειας μέσω μίας και μοναδικής ρουτίνας
- Διαθέτει υψηλό βαθμό ευελιξίας και επεκτασιμότητας
- Τρέχει σε διαφορετικές πλατφόρμες (Linux και Windows 32)
- Διαθέτει ενσωματωμένη υποστήριξη για την ανάπτυξη πολυνηματικών (multithreaded) εφαρμογών
- Διαθέτει ενσωματωμένες ρουτίνες αυτοελέγχου του ρομπότ και αποφυγής εμποδίων
- Απλοποιεί την διαδικασία ανάπτυξης προσαρμοσμένων ενεργειών
- Συγκρατεί ιστορικές πληροφορίες από τα δεδομένα που μεταδίδει το σόναρ
- Έχει σχεδιαστεί με στόχο να υποστηρίξει την ανάπτυξη εφαρμογών μεγάλης κλίμακας

Από τα παραπάνω γίνεται εύκολα αντιληπτή η χρησιμότητα του συγκεκριμένου API για κάποιον που θέλει να αναπτύξει τη δική του εφαρμογή πλοήγησης για τον συγκεκριμένο τύπο ρομπότ. Στην επόμενη παράγραφο εισχωρούμε σε λεπτομερέστερη ανάλυση του τρόπου υλοποίησης της εφαρμογής μας, παρουσιάζοντας τις συγκεκριμένες ρουτίνες του API της ARIA που χρησιμοποιήσαμε.

5.4.2 Αρχιτεκτονική του Συστήματος και κλάσεις που χρησιμοποιήθηκαν από το API της ARIA

Τα παραπάνω καθίστανται δυνατά με την χρήση των έτοιμων προγραμματιστικών διεπαφών του περιβάλλοντος της ARIA τις οποίες και καλούμε από την εφαρμογή μας. Εξαιτίας της σπουδαιότητας τους για την υλοποίηση του συστήματος μας, παραθέτουμε τις σημαντικότερες από αυτές τις κλάσεις στον πίνακα 5.17. Η γενική αρχιτεκτονική του συστήματος φαίνεται στην εικόνα 5.16.



Εικόνα 5.16: Αρχιτεκτονική της Διαδικασίας Πλοήγησης του AmigoBot

Κλάση “myRobot”		
Ρουτίνα Aria	Σκοπός	Μέθοδοι
ArRobot	Είναι η πιο σημαντική κλάση και ίσως η μόνη που θα πρέπει να χρησιμοποιηθεί οπωσδήποτε σε ανάλογες εφαρμογές. Ουσιαστικά προσομοιώνει το AmigoBot και μας επιτρέπει τόσο να λάβουμε πληροφορίες από τους αισθητήρες του όσο και να του μεταδώσουμε εντολές πλοήγησης και χειρισμού των εξαρτημάτων του.	<ul style="list-style-type: none"> ▪ .setDeviceConnection ▪ .runAsync ▪ .lock ▪ .disconnect ▪ .unlock ▪ .getVel ▪ .getTh ▪ .enableMotors ▪ .disableMotors ▪ .addRangeDevice ▪ .getX ▪ .getY ▪ .addAction ▪ .areMotorsEnabled ▪ .isConnected
ArTcpConnection	Δημιουργεί μία νέα σύνδεση με το AmigoBot μέσω TCP/IP. Παίρνει σαν παραμέτρους την IP διεύθυνση του ρομπότ και την θύρα (port) στην οποία το ρομπότ αναμένει νέες συνδέσεις από εφαρμογές – πελάτη. Τελικά εισάγεται σαν παράμετρος στην “setDeviceConnection” της κλάσης “ArRobot”.	<ul style="list-style-type: none"> ▪ .setPort ▪ .openSimple
ArSonarDevice	Παρέχει μεθόδους για την παρακολούθηση και τον έλεγχο του συστήματος σόναρ του AmigoBot. Συνδέουμε την κλάση αυτή με κάποιο στιγμιότυπο της κλάσης “ArRobot” μέσω της μεθόδου “addRangeDevice” της “ArRobot”	
ArPose	Αντιστοιχεί σε ένα σημείο του χώρου. Περιγράφεται από το σύστημα συντεταγμένων (X,Y) όπου X: τετμημένη και Y: τεταγμένη. Χρησιμοποιούμε την κλάση αυτή ώστε να μεταβιβάσουμε τα «υποχρεωτικά σημεία» τροχιάς στο AmigoBot	<ul style="list-style-type: none"> ▪ .setPose
ArActionGoto	Η κλάση αυτή δέχεται σαν όρισμα κάποιο σημείο που έχουμε προσδιορίσει μέσω της “ArPose” και οδηγεί το AmigoBot στο σημείο αυτό. Παρέχει μεθόδους ελέγχου της διαδικασίας αυτής (για παράδειγμα αν όντως το ρομπότ έφτασε στο επιθυμητό σημείο)	<ul style="list-style-type: none"> ▪ .setGoal ▪ .haveAchievedGoal ▪ .cancelGoal ▪ .setSpeed

Πίνακας 5.17: Οι κλάσεις από το API της ARIA που χρησιμοποιήσαμε για την εφαρμογή μας

Στην εικόνα 5.16 βλέπουμε την αρχιτεκτονική του προτεινόμενου συστήματος. Ουσιαστικά είναι μία συνοπτική ματιά της διαδικασίας πλοήγησης που περιγράψαμε στην παράγραφο 5.3.3. Έχουμε σημειώσει τις ενέργειες που απαιτούνται για την επιτυχημένη πλοήγηση. Οι ενέργειες αυτές είναι τέσσερις και αναλύονται στην συνέχεια.

Βήμα 1: Καταγραφή Εικόνων του περιβάλλοντος

Η κάμερα μας (ενσύρματη ή ασύρματη για μεγαλύτερη αυτονομία) καταγράφει συνεχώς τον χώρο που κινείται το ρομπότ και μέσω της βιβλιοθήκης χειρισμού πολυμέσων **JMF** (Java Media Framework) μεταδίδει τις εικόνες στο γραφικό περιβάλλον διασύνδεσης χρήστη (GUI) της εφαρμογής μας. Εδώ πρέπει να σημειώσουμε πάλι, ότι το AmigoBot θα πρέπει να βρίσκεται εντός του οπτικού πεδίου της κάμερας.

Βήμα 2: Σύλληψη Συγκεκριμένης Εικόνας και Αποστολής της προς Επεξεργασία

Όπως είπαμε και στην παράγραφο 5.3.3 όταν ο χρήστης επιλέξει από τα μενού επιλογών “Draw Path” ή τρέχουσα εικόνα που βλέπει ο χρήστης «παγώνει» και αποθηκεύεται προσωρινά στον σκληρό δίσκο του υπολογιστή.

Βήμα 3: Επεξεργασία Εικόνας

Σχεδόν ταυτόχρονα με το βήμα 2, η εφαρμογή μας ξεκινά την επεξεργασία της αποθηκευμένης εικόνας. Ο σκοπός είναι όπως έχουμε ξαναπεί αρχικά να εντοπίσει το AmigoBot και στην συνέχεια να προσδιορίσει την σωστή κλίμακα των διαστάσεων της εικόνας σε σχέση με τις διαστάσεις του πραγματικού χώρου. Τελικά το αποτέλεσμα της επεξεργασίας αυτής εμφανίζεται στην οθόνη του χρήστη ο οποίος είναι σε θέση πλέον να καθοδηγήσει το AmigoBot.

Βήμα 4: Μεταβίβαση Εντολών Πλοήγησης

Ο χρήστης παίρνει τώρα τον έλεγχο των κινήσεων. Αν αποφασίσει ότι η κλίμακα που προσδιόρισε αυτόματα η εφαρμογή δεν είναι η σωστή μπορεί όπως έχουμε ξαναπεί να την ορίσει χειροκίνητα (παρ. 5.3.3). Σε κάθε περίπτωση τελικά θα χαράξει μία νέα τροχιά για το AmigoBot σύμφωνα με την τεχνική των «υποχρεωτικών σημείων». Μέσω των βιβλιοθηκών του API της Aria οι νέες αυτές εντολές καθοδήγησης του AmigoBot θα μεταδοθούν στο AmigoOS και το ρομπότ μας θα ξεκινήσει την πορεία του.

Ακολουθούν οι παρατηρήσεις μας όσον αφορά τους περιορισμούς και τα μειονεκτήματα του προτεινόμενου συστήματος καθώς και οι πιθανές βελτιώσεις που θα μπορούσαν να εφαρμοσθούν σε αυτό προκειμένου να καταστεί αποτελεσματικότερο.

6 Περιορισμοί και Προτεινόμενες Βελτιώσεις

Μέσα από όλη την πορεία της μελέτης μας και συγκρίνοντας το σύστημα που αναπτύξαμε με παρόμοιες υλοποιήσεις μπορούμε να πούμε ότι το σύστημα μας επιλύει ικανοποιητικά πολλά από τα προβλήματα της διαδικασίας πλοήγησης αυτοκινούμενων ρομπότ. Παρολαυτά για να είναι αξιοποιήσιμο σε πραγματικές συνθήκες ή ακόμα και για εμπορικούς σκοπούς θα έπρεπε προηγουμένως να τελειοποιηθούν διάφορες παράμετροι του.

Η σημαντικότερη αδυναμία του, όπως έχουμε τονίσει και στην παράγραφο 5.2 είναι το γεγονός ότι η ρομποτική μας πλατφόρμα (AmigoBot) θα πρέπει να βρίσκεται διαρκώς μέσα στο οπτικό πεδίο της κάμερας που το παρακολουθεί. Σε αντίθετη περίπτωση η επεξεργασία της εικόνας που συλλαμβάνεται θα καταλήξει σε ανεπαρκή αποτελέσματα και δεν θα είμαστε σε θέση όχι μόνο να το πλοηγήσουμε επιτυχώς αλλά ούτε καν να το εντοπίσουμε στον χώρο.

Για την επίλυση του παραπάνω περιορισμού προτείνουμε τις παρακάτω τρεις λύσεις:

1. την χρησιμοποίηση είτε περισσότερων συσκευών παρακολούθησης ώστε να καλύπτεται όλος ο χώρος
2. την κατάλληλη προσαρμογή μίας κάμερας επάνω στο αμάξιμο του AmigoBot, με τέτοιο τρόπο όμως ώστε να καταγράφει τόσο το ρομπότ όσο και μεγάλο τμήμα του περιβάλλοντος χώρου (π.χ. μια κάμερα με **ευρυγώνιο φακό**)
3. η καταλληλότερη ίσως λύση είναι η πλήρης εκμετάλλευση των δυνατοτήτων της **ενεργούς οράσεως** (active vision) [46] που περιγράψαμε στην παράγραφο 5.3. Δηλαδή η χρήση μίας κάμερας που θα είναι σε θέση να παρακολουθεί με ενεργό τρόπο το AmigoBot.

Ένας ακόμα περιορισμός έχει να κάνει με την πιθανότητα να παρουσιαστεί η ανάγκη για τον ταυτόχρονο έλεγχο περισσότερων από ένα ρομπότ. Η εφαρμογή μας αυτή την στιγμή δεν μπορεί να διαχειριστεί μία τέτοια κατάσταση αν και επιτρέπει την σύνδεση σε διαφορετικά ρομπότ αλλά μόνο με ένα από αυτά την φορά. Η υλοποίηση βασισμένη σε νήματα (**threads**) που χρησιμοποιούμε μπορεί να τροποποιηθεί ώστε να καλύψει τον ταυτόχρονο χειρισμό περισσότερων ρομπότ, αναθέτοντας για παράδειγμα τον έλεγχο καθενός από αυτά σε ξεχωριστό νήμα.

Τέλος επισημαίνουμε το γεγονός ότι κατά την διάρκεια της ανάπτυξης χρησιμοποιήσαμε μία ενσύρματη δικτυακή κάμερα συνδεδεμένη με τον υπολογιστή στον οποίο εκτελείται η εφαρμογή. Για σκοπούς πέρα από ερευνητικούς θεωρούμε αυτονόητη την χρησιμοποίηση μίας ασύρματης δικτυακής κάμερας που θα δώσει πολύ μεγαλύτερη ευελιξία όσον αφορά την έκταση του χώρου στον οποίο μπορεί το σύστημα μας να πλοηγήσει το AmigoBot. Για την ενσωμάτωση μίας κάμερας αυτού του τύπου δεν απαιτείται καμία τροποποίηση στο κώδικα αφού η βιβλιοθήκη πολυμέσων της JMF (Java Media Framework) που χρησιμοποιούμε θα την διαχειριστεί με παρόμοιο τρόπο με την ενσύρματη.

Στο σημείο έχουμε ολοκληρώσει την μελέτη μας την οποία και κλείνουμε στο επόμενο κεφάλαιο με την παράθεση των συμπερασμάτων μας.

7 Συμπεράσματα

Στην παρούσα εργασία αναπτύξαμε ένα εξελιγμένο σύστημα τηλεχειρισμού ρομπότ μέσω δικτυακής κάμερας, το οποίο βασίζεται σε μία ολοκληρωμένη διεπαφή χρήστη και παράλληλα προτείναμε λύσεις για μερικά από τα σημαντικότερα προβλήματα που παρουσιάζει ένα τέτοιο εγχείρημα, όπως είναι ο εντοπισμός του ρομπότ στον χώρο και η ρεαλιστική πλοήγηση του με βάση τις πληροφορίες που λαμβάνουμε από την κάμερα

Τα σημεία στα οποία κυρίως επικεντρώσαμε την προσοχή μας είναι:

1. η ανίχνευση του ρομπότ στον χώρο με βάση τις πληροφορίες που λαμβάνουμε από την κάμερα,
2. η χάραξη ασφαλούς τροχιάς για το ρομπότ και τέλος
3. η σωστή αναγωγή των διαστάσεων που προβάλλονται στην οθόνη σε διαστάσεις του πραγματικού χώρου.

Εκτός από αυτά βέβαια η εφαρμογή μας δεν παύει να αποτελεί μια ολοκληρωμένη διεπαφή χρήστη για τον έλεγχο του ρομπότ φροντίζοντας για παράδειγμα, για την σύνδεση και αποσύνδεση σε αυτό, για τον χειρισμό της κάμερας κ.α., παρέχοντας ταυτόχρονα δυναμικά, ένα πλήθος από πληροφορίες σχετικά με την όλη διαδικασία.

Το βασικότερο πλεονέκτημα που παρέχει σε σχέση με άλλες υλοποιήσεις πλοήγησης αυτοκινούμενων ρομπότ εντοπίζεται στην διαδικασία εντοπισμού της ρομποτικής πλατφόρμας στον χώρο. Συγκεκριμένα στο προτεινόμενο σύστημα η διαδικασία αυτή δεν εξαρτάται καθόλου από εξωτερικούς υποστηρικτικούς παράγοντες ανίχνευσης (όπως GPS, πυξίδες, τεχνητά ορόσημα κ.τ.λ.). Αντιθέτως χρησιμοποιούμε σαν ορόσημο το ίδιο το ρομπότ και στηρίζομαστε στο έντονο κόκκινο χρώμα του που το διακρίνει ξεκάθαρα από το περιβάλλον του. Αρκεί να υπάρχει μια κάμερα προσαρμοσμένη είτε επάνω στο ίδιο το ρομπότ είτε σε κάποιο σταθερό σημείο του χώρου η οποία θα το εμποπτεύει.

Η μελλοντική εξέλιξη του συγκεκριμένου συστήματος, όπως αναφέρουμε και στο κεφάλαιο 7 θα μπορούσε να περιλαμβάνει την ενσωμάτωση τεχνικών ενεργούς οράσεως που θα προσφέρει ακόμα μεγαλύτερη ευελιξία, καθώς και την ταυτόχρονη υποστήριξη περισσότερων του ενός ρομπότ μέσω μοναδικής διεπαφής χρήστη.

8 Αναφορές

- [1] **Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Web-based Tools:** Terrence Fong, Charles Thorpe and Charles Baur
- [2] **Ρομποτική:** Δ. Εμίρης, Β' Έκδοση 2003, Εκδόσεις: Τεκδοτική, ISBN: 960-8257-20-4
- [3] **Chapter 56. Virtual Reality Assisted Teleoperation:** Abderrahmane KHEDDAR PhD, University of Evry Val d'Essonne, France
- [4] **IEEE TRA. (1993). IEEE Transactions on Robotics and Automation, Special issue on space robotics:** Laurence
- [5] **Mobile Robot Positioning & Sensors and Techniques:** J. Borenstein¹, H.R. Everett, L. Feng, and D. Wehe - Invited paper for the Journal of Robotic Systems, Special Issue on Mobile Robots. Vol. 14 No. 4, pp. 231 – 249.
- [6] **An Inertial Navigation System for a Mobile Robot:** Barshan, B. and Durrant-Whyte, H.F., 1993 - Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robotics and Systems, Yokohama, Japan, July 26-30, pp. 2243-2248
- [7] **Techniques for Autonomous Navigation:** Byrne, R.H., Klarer, P.R., and Pletta, J.B., 1992 - Sandia Report SAND92-0457, Sandia National Laboratories, Albuquerque, NM, March
- [8] **Navigating Mobile Robots: Systems and Techniques:** Borenstein, J., Everett, B., and Feng, L., 1996b - CD-ROM Edition, A. K. Peters, Ltd., Wellesley, MA, ISBN 1-56881-058-X
- [9] **Global Navigation for ARK:** Jenkin, M., Miliotis, E., Jasiobedzki, P., Bains, N., and Tran, K., 1993 - Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robotics and Systems, Yokohama, Japan, July 26-30, pp. 2165-2171
- [10] **Real-time Vision-based Robot Localization:** Atiya, S. and Hager, G., 1993 - IEEE Transactions on Robotics and Automation, Vol. 9, No. 6, pp. 785-800
- [11] **TV Image Processing to Determine the Position of a Robot Vehicle:** Fukui, I., 1981 – Pattern Recognition, Vol. 14, pp. 101-109
- [12] **Adaptive Position Estimation for an Automated Guided Vehicle:** Lapin, B., 1992 - Proc. Of the 1992 SPIE Conference on Mobile Robots, Boston, MA, Nov. 18-20 pp. 82-94.
- [13] **Absolute External Mobile Robot Localization Using a Single Image:** Fleury, S. and Baron, T., 1992 - Proceedings of the 1992 SPIE Conference on Mobile Robots, Boston, MA, Nov. 18- 20, pp. 131-143

- [14] **Sensors for Mobile Robots: Theory and Application:** Everett, H. R., 1995 -A K Peters, Ltd., Wellesley, MA, ISBN 1-56881-048-2
- [15] **Estimate of Absolute Position of Mobile Systems by Opto-electronic Processor:** Feng, L., Fainman, Y., and Koren, Y., 1992 - IEEE Transactions on Man, Machine and Cybernetics, Vol. 22, No. 5, pp. 954-963
- [16] **Concurrent Localization and Map Building for Mobile Robots Using Ultrasonic Sensors:** Rencken, W.D., 1993 - Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robotics and Systems, Yokohama, Japan, July 26-30, pp. 2192-2197
- [17] **Autonomous Sonar Navigation in Indoor, Unknown, and Unstructured Environments:** Rencken, W.D., 1994 - 1994 International Conference on Intelligent Robots and Systems (IROS '94). Munich, Germany, Sept. 12-16, pp. 127-134
- [18] **Exploration of an Indoor Environment by an Autonomous Mobile Robot:** Edlinger, T. and Puttkamer, E., 1994 - International Conference on Intelligent Robots and Systems (IROS '94). Munich, Germany, Sept. 12-16, pp. 1278-1284
- [19] **Laserradar and Sonar Based World Modeling and Motion Control for Fast Obstacle Avoidance of the Autonomous Mobile Robot MOBOT-IV:** Buchberger, M., JCrg, K., and Puttkamer, E., 1993 - Proceedings of IEEE International Conference on Robotics and Automation, Atlanta, GA, May 10-15, pp. 534-540
- [20] **Echtzeitfähige Multisensorintegration für autonome mobile Roboter:** Jorg, K.W., 1994 - ISBN 3-411-16951-6, B.I. Wissenschaftsverlag, Mannheim, Leipzig, Wien, ZKrich
- [21] **World Modeling for an Autonomous Mobile Robot Using Heterogenous Sensor Information:** Jorg, K.W., 1995 - Robotics and Autonomous Systems, Vol. 14, pp. 159-170
- [22] **Hierarchical Evidence Accumulation in the PSEIKI System and Experiments in Model-driven Mobile Robot Navigation:** Kak, A., Andress, K., Lopez-Abadia, and Carroll, M., 1990 - Uncertainty in Artificial Intelligence, Vol. 5, Elsevier Science Publishers B. V., North-Holland, pp. 353-369
- [23] **Position Estimation Techniques for an Autonomous Mobile Robot - a Review:** Talluri, R., and Aggarwal, J., 1993 - in Handbook of Pattern Recognition and Computer Vision, World Scientific: Singapore, Chapter 4.4, pp. 769-801
- [24] **Mars pathfinder microrover: A low-cost, low-power spacecraft:** H. W. Stone – In Proceedings of the 1996 AIAA Forum on Advanced Developments in Space Robotics, Madison, WI., 1996.
- [25] **Advanced Guided Vehicles - Aspects of the Oxford AGV Project.:** S. A. Cameron and P. J. Probert - World Scientific, Singapore, 1994

- [26] **Directed Sonar Sensing for Mobile Robot Navigation:** J. J. Leonard - PhD thesis, University of Oxford, 1990
- [27] **Dynamic map building for an autonomous mobile robot.:** J. J. Leonard, H. Durrant-Whyte, and I. J. Cox - International Journal of Robotics Research, 11(4):286-298, 1992
- [28] **Directed Sonar Navigation:** J. J. Leonard and H. F. Durrant-Whyte - Kluwer Academic Press, 1992.
- [29] **An Information-Theoretic Approach to Data Fusion and Sensor Management.:** J. Manyika - PhD thesis, University of Oxford, 1993.
- [30] **Where am I? A tutorial on mobile vehicle localization:** H. F. Durrant-Whyte - Industrial Robot, 21(2):11{16, 1994.
- [31] **Directed Sonar Sensing for Mobile Robot Navigation:** J. J. Leonard - PhD thesis, University of Oxford, 1990.
- [32] **Quantitative evaluation of the exploration strategy of a mobile robot:** D. Lee and M. Recce - In AISB 94 Workshop: Models or behaviours | which way forward for robotics?, University of Leeds, April 1994.
- [33] **Animal and robot navigation:** U. Nehmzow - In The Biology and Technology of Intelligent Autonomous Agents. Springer Verlag, 1993.
- [34] **Architectures and algorithms for 3D vision: the parallel Droid system:** M. Brady H. Wang and C. Bowman - In S. Cameron and P. Probert, editors, Advanced Guided Vehicles, pages 125{139. World Scienti_c, Singapore, 1994
- [35] **A robust, layered control system for a mobile robot:** R. A. Brooks - Technical Report A.I. Memo 864, Massachusetts Institute of Technology, September 1985.
- [36] **Achieving arti_cial intelligence through building robots:** R. A. Brooks – Technical Report A.I. Memo 899, Massachusetts Institute of Technology, May 1986
- [37] **Intelligence without representation:** R. A. Brooks - In Workshop on the Foundations of Artificial Intelligence, Dedham, MA, 1987
- [38] **Building brains for bodies:** R. A. Brooks and L. A. Stein - Technical Report A.I. Memo 1439, Massachusetts Institute of Technology, August 1993
- [39] **A behaviour-based system for off-road navigation:** D. Langer, J. K. Rosenblatt, and M. Hebert - IEEE Trans. Robotics and Automation, 10(6):776{782, 1994
- [40] **Stereo perception and dead-reckoning for a prototype lunar rover:** E. Krotkov, M. Hebert, and R. Simmons - Autonomous Robots, 2(4):313 - 331, 1995.
- [41] **Neural Network Perception for Mobile Robot Guidance:** D. Pomerleau - Kluwer Academic Publishers, 1993.

- [42] **Vision guided lane transition:** T. M. Jochem, D. A. Pomerleau, and C. E. Thorpe – In IEEE Symposium on Intelligent Road Vehicles, Detroit, MI, 1995
- [43] **Overtaking vehicle detection using implicit optical flow.:** P. H. Batavia, D. A. Pomerleau, and C. E. Thorpe - In Proceedings of the IEEE Intelligent Transportation Systems Conference, Boston, MA, 1997
- [44] **AmigoBot User's Guide:** Copyright 2003, ActivMedia Robotics, LLC. All rights reserved - Version 1.9, August, 2003
- [45] **Team AmigoBot™ Operations Manual:** Copyright 2005, ActivMedia Robotics, LLC. All rights reserved - version 2, September 2005
- [46] **Mobile Robot Navigation Using Active Vision:** Andrew John Davison Keble College - Robotics Research Group Department of Engineering Science University of Oxford Submitted February 14 1998; Examined June 9th 1998