



Πολυτεχνείο Κρήτης
Τμήμα Ηλεκτρονικών Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Πληροφορικής

«Αναγνώριση Ανθρωπίνων Προσώπων και Ενεργή Παρακολούθηση
με τα Ρομποτικά Τετράποδα AIBO»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Κωνσταντίνος Δ. Μακαντάσης

Επιβλέπων Καθηγητής : Μιχαήλ Γ. Λαγουδάκης

Εξεταστική Επιτροπή : Επ. Καθ. Μιχαήλ Γ. Λαγουδάκης
Καθ. Μιχάλης Ζερβάκης
Επ. Καθ. Νικόλαος Βλάσσης

Χανιά 2008

Στην οικογένεια μου

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Πολυτεχνείου Κρήτης. Η εκκίνηση της διπλωματικής εργασίας τοποθετείται χρονικά τον Φεβρουάριο του 2007.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον καθηγητή μου κύριο Μ. Λαγουδάκη για την καθοδήγησή του κατά την εκπόνηση της παρούσας διπλωματικής εργασίας, τις πολύτιμες γνώσεις και τη στήριξη που μου προσέφερε. Τον Λάζαρο Κιτσάκη για την τεχνική βοήθεια, τον Δημήτριο Στρεβίνα και το Στέλιο Κόκκαλη για τις πολύτιμες συμβουλές τους και την Μερόπη Μανατάκη για την ψυχολογική υποστήριξη.

Φεβρουάριος 2008

Κωνσταντίνος Δ. Μακαντάσης

Περιεχόμενα

1. Εισαγωγή.....	10
1.1 Ορισμός και οργάνωση της εραγσίας.....	12
1.2 Σύνοψη.....	13
2. Το ρομπότ AIBO.....	15
2.1 Τεχνικά χαρακτηριστικά του AIBO.....	16
2.2 Προγραμματιστικά περιβάλλοντα.....	17
2.2.1 OPENR-SDK.....	18
2.2.2 Tekkotsu.....	19
2.2.3 URBI.....	19
2.2.4 PYRO.....	19
2.2.5 YART/RCodePlus.....	20
2.2.6 R-Code.....	20
2.3 OPENR-SDK.....	20
2.3.1 OPENR Object.....	22
2.3.2 Δι-αντικειμενική επικοινωνία.....	24
2.3.3 Βασική κλάση.....	25
2.3.4 Ροή της ανάπτυξης του προγράμματος.....	27
2.3.5 Ροή της εκτέλεσης.....	29
2.4 Remote Processing Control (RPC).....	29
2.5 Μηχανή κίνησης του AIBO.....	31
3. Η βιβλιοθήκη OpenCV.....	33
3.1 OpenCV και ανίχνευση αντικειμένων.....	34
3.2 Διαδικασία μάθησης.....	36
3.3 Χαρακτηριστικά και αναγνώριση προσώπων.....	37
3.4 Διαδικασία ανίχνευσης.....	38
4. Σχετικές εργασίες.....	40

4.1	Όραση και κίνηση ως μέσα αλληλεπίδρασης ρομπότ-ανθρώπου....	41
5.	Προσέγγιση του προβλήματος.....	47
5.1	Ορισμός του προβλήματος.....	48
5.2	Όραση του AIBO.....	48
5.2.1	Λήψη εικόνας με χρήση OPEN-R.....	48
5.3	Μετατροπή της εικόνας του AIBO σε τύπο BMP.....	50
5.4	Ανίχνευση προσώπων.....	51
5.5	Χρήση της βιβλιοθήκης OpenCV.....	52
5.6	OpenCV και AIBO.....	54
5.6.1	Χρήση της OpenCV σε πλήρως αυτόνομη λειτουργία του ρομπότ.....	55
5.6.2	Ανίχνευση προσώπων σε πλήρως αυτόνομη λειτουργία του ρομπότ.....	57
5.7	Αρχιτεκτονική του προγράμματος και τα αντικείμενα OPEN-R....	59
5.7.1	Ανάλυση του αντικειμένου FaceDetect.....	61
5.7.2	Ανάλυση του αντικειμένου MovingHead.....	67
5.7.3	Ανάλυση του αντικειμένου MovingLegs.....	70
5.8	Σύνοψη του προγράμματος.....	74
6.	Αποτελέσματα και μελλοντικές προεκτάσεις.....	78
6.1	Στόχος και υλοποίηση της εφαρμογής.....	79
6.2	Αποτελέσματα.....	79
6.3	Μελλοντικές προεκτάσεις και βελτιώσεις.....	82
6.4	Συμπεράσματα.....	83
	Αναφορές.....	84

Περίληψη

Η παρούσα διπλωματική εργασία εστιάζει στην αλληλεπίδραση ανθρώπου με ρομπότ. Κοινό χαρακτηριστικό της αλληλεπίδρασης μεταξύ ανθρώπων είναι η πρόσωπο προς πρόσωπο αντικρυστή επικοινωνία. Στόχος της εργασίας είναι η ανάπτυξη λογισμικού για το ρομποτικό τετράποδο AIBO ώστε να επιδεικνύει ανάλογη συμπεριφορά στην αλληλεπίδρασή του με ανθρώπους. Συγκεκριμένα, το ρομπότ προσπαθεί να ανιχνεύσει κάποιο ανθρώπινο πρόσωπο στο οπτικό του πεδίο και να εστιάσει το βλέμμα του σ' αυτό ώστε να το βλέπει συνεχώς enface. Στην προσπάθειά του αυτή καλείται να χρησιμοποιήσει όχι μόνο τις κινήσεις του κεφαλιού, αλλά και τη δυνατότητα μετακίνησής του στο χώρο, έτσι ώστε, εάν το πρόσωπο φαίνεται σε προφίλ, να αναγνωρίζει την κατάλληλη κατεύθυνση μετακίνησης και να κινείται ανάλογα με τελικό στόχο να βλέπει το πρόσωπο enface. Για την υλοποίηση της εν λόγω συμπεριφοράς χρησιμοποιήθηκαν πολλά εργαλεία (OpenCV, Open-R, Remote Processing Control, Walk Engine), τα οποία προσαρμόστηκαν στις δυνατότητες του υλικού και του λογισμικού του ρομπότ και ενοποιήθηκαν κατάλληλα ώστε να προκύψει ένα ολοκληρωμένο σύστημα με δυνατότητες για περαιτέρω αξιοποίηση σε εφαρμογές περίπλοκης οπτικής αντίληψης και μηχανικής δράσης. Πιστεύουμε ότι η παρούσα εργασία είναι ένα βήμα προς την κατεύθυνση της γεφύρωσης των διαφορών στους τρόπους επικοινωνίας μεταξύ ανθρώπων και μηχανών.

Κεφάλαιο 1

Εισαγωγή

Από το ξεκίνημα της επιστήμης των υπολογιστών, η επικοινωνία και η αλληλεπίδραση μεταξύ ανθρώπου και υπολογιστή ήταν και παραμένει μια από τις βασικές επιδιώξεις του ανθρώπου. Η τεχνητή νοημοσύνη και η μηχανική μάθηση αποτελούν τα σημαντικότερα εργαλεία στην προσπάθεια για την επίτευξη αυτού του στόχου.

Ο άνθρωπος για να επικοινωνήσει με τον υπολογιστή χρησιμοποιεί τις αισθήσεις του. Ανάλογα αισθητήρια, που θα παρέχουν εξωτερικά ερεθίσματα, θα πρέπει να διαθέτει και ο υπολογιστής για την επίτευξη αμφίδρομης επικοινωνίας. Μία κάμερα σε συνδυασμό με την ανάπτυξη κατάλληλων εφαρμογών μηχανικής όρασης, αποτελούν ίσως το σημαντικότερο αισθητήριο για έναν υπολογιστή, κυρίως λόγω του μεγάλου όγκου δεδομένων που μπορεί να περιέχεται σε μια εικόνα.

Η συνεχής βελτίωση των υπολογιστικών συστημάτων σε συνδυασμό με την βελτίωση των μέσων λήψης εικόνων και την ανάπτυξη της μηχανικής όρασης, έχει ως συνέπεια τη δημιουργία μιας πληθώρας εφαρμογών που σκοπό έχουν την προσομοίωση της λειτουργίας του ανθρώπινου ματιού μέσα από την επεξεργασία εικόνας. Αν και η πλήρης προσομοίωση της λειτουργίας του ανθρώπινου ματιού δεν έχει επιτευχθεί, μέσω των σύγχρονων συστημάτων μηχανικής όρασης και των ανάλογων εφαρμογών παρέχεται σημαντική εξωτερική πληροφορία στον υπολογιστή. Παραδείγματα τέτοιων εφαρμογών είναι η αναγνώριση ανθρωπίνων προσώπων, η αναγνώριση χειρονομιών και μορφασμών, η κατανόηση κίνησης στον περιβάλλοντα χώρο κ.λ.π.

Η ανάγκη για δημιουργία ικανοποιητικής αλληλεπίδρασης μεταξύ ανθρώπου και υπολογιστή έγινε ακόμη μεγαλύτερη με την ανάπτυξη της ρομποτικής και την κατασκευή και χρήση οικιακών ρομπότ. Οι χρήστες που θα αλληλεπιδρούν με τα

οικιακά ρομπότ δεν είναι απαραίτητο να έχουν γνώσεις ρομποτικής, ίσως μάλιστα να μην έχουν καθόλου τέτοιου είδους γνώσεις, γι' αυτό το λόγο ο τρόπος επικοινωνίας και αλληλεπίδρασης θα πρέπει να είναι εξαιρετικά διαισθητικός.

1.1 Ορισμός και οργάνωση της εργασίας

Ένα μεγάλο μέρος της επικοινωνίας που πραγματοποιείται μεταξύ των ανθρώπων οφείλεται στις εκφράσεις του προσώπου. Η επικοινωνία, που στηρίζεται μόνο σε λόγια, χωρίς οπτική επαφή και χωρίς εκφράσεις είναι δύσκολη. Υπάρχουν εκφράσεις που αν ειπωθούν με διαφορετικό τρόπο αποκτούν τελείως διαφορετική σημασία. Η πληροφορία που περιέχεται στο πρόσωπο ενός ανθρώπου, κυρίως σε θέση enface, είναι πολύ σημαντική για την ταυτοποίηση του και ίσως για την εξαγωγή στοιχείων του χαρακτήρα του. Γι' αυτούς τους λόγους η επικοινωνία ανθρώπων πρόσωπο προς πρόσωπο είναι πολύ σημαντική. Η πληροφορία του ανθρώπινου προσώπου μπορεί να θεωρηθεί το ίδιο σημαντική και για την αλληλεπίδραση μεταξύ ανθρώπου και ρομπότ ή υπολογιστή.

Ο σκοπός της διπλωματικής μου εργασίας ήταν ο ακόλουθος:

“Σχεδιασμός και υλοποίηση κατάλληλου λογισμικού για το τετράποδο ρομπότ AIBO, μέσω του οποίου θα επιτυγχάνεται ανίχνευση και παρακολούθηση ανθρώπινου προσώπου, σε θέση προφίλ και enface. Αν το πρόσωπο που ανιχνεύτηκε είναι σε θέση προφίλ, απαιτείται ο σχεδιασμός και η υλοποίηση κίνησης, ώστε το ρομπότ να βρεθεί σε θέση που να μπορεί να δει το πρόσωπο enface.”

Μια εφαρμογή που θα χρησιμοποιεί κινούμενη κάμερα για την ανίχνευση προσώπων μπορεί να αποδειχθεί πολύ χρήσιμη. Η εγκατάσταση κινούμενων

καμερών μπορεί να διευκολύνει την επικοινωνία απομακρυσμένων συνομιλητών. Επίσης, μια τέτοια εφαρμογή μπορεί να χρησιμοποιηθεί για φύλαξη ιδιωτικών και δημόσιων χώρων καθώς και για ανίχνευση παραβατικής συμπεριφοράς.

Η ολοκλήρωση της εργασίας ακολούθησε τα παρακάτω στάδια:

- i. Συλλογή και μελέτη δημοσιεύσεων σχετικές με το AIBO, ώστε να κατανοηθεί η λειτουργία του, και εξέταση των τεχνικών προγραμματισμού που παρέχει.
- ii. Μελέτη εφαρμογών μηχανικής όρασης που αφορούν στην ανίχνευση αντικειμένων με συγκεκριμένα χαρακτηριστικά.
- iii. Απόφαση για χρήση συγκεκριμένων προγραμματιστικών τεχνικών και σχεδιασμός ενός μοντέλου σύμφωνα με τις απαιτήσεις της εργασίας.
- iv. Σταδιακή υλοποίηση και πρακτική χρήση του μοντέλου που σχεδιάστηκε, ώστε να πληρεί τις απαιτήσεις της εργασίας.
- v. Εκτίμηση των αποτελεσμάτων και ανάλυση του βαθμού επιτυχίας της εφαρμογής.

1.2 Σύνοψη

Η παρούσα εργασία έχει χωριστεί σε κεφάλαια. Στο παρόν κεφάλαιο υπάρχουν γενικές πληροφορίες για την αλληλεπίδραση ανθρώπου με μηχανή. Αναφέρεται στην σημασία της πρόσωπο προς πρόσωπο επικοινωνίας και παρουσιάζει τον στόχο της συγκεκριμένης εργασίας, καθώς και την δομή του παρόντος κειμένου. Στο δεύτερο και στο τρίτο κεφάλαιο παρουσιάζονται τα τεχνικά χαρακτηριστικά

του ρομπότ και τα εργαλεία τα οποία χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Στο τέταρτο κεφάλαιο γίνεται μια αναφορά σε προηγούμενες εργασίες που συνδυάζουν την χρήση του AIBO με την επεξεργασία εικόνας. Στο πέμπτο κεφάλαιο παρουσιάζεται αναλυτικά ο τρόπος με τον οποίο προσεγγίστηκε το πρόβλημα και ο τρόπος υλοποίησης της εφαρμογής. Τέλος, στο έκτο κεφάλαιο γίνεται παρουσίαση των αποτελεσμάτων που προκύπτουν από την εκτέλεση της εφαρμογής και κριτική τους. Επίσης, δίνονται προτάσεις για πιθανές μελλοντικές επεκτάσεις και βελτιώσεις.

Κεφάλαιο 2

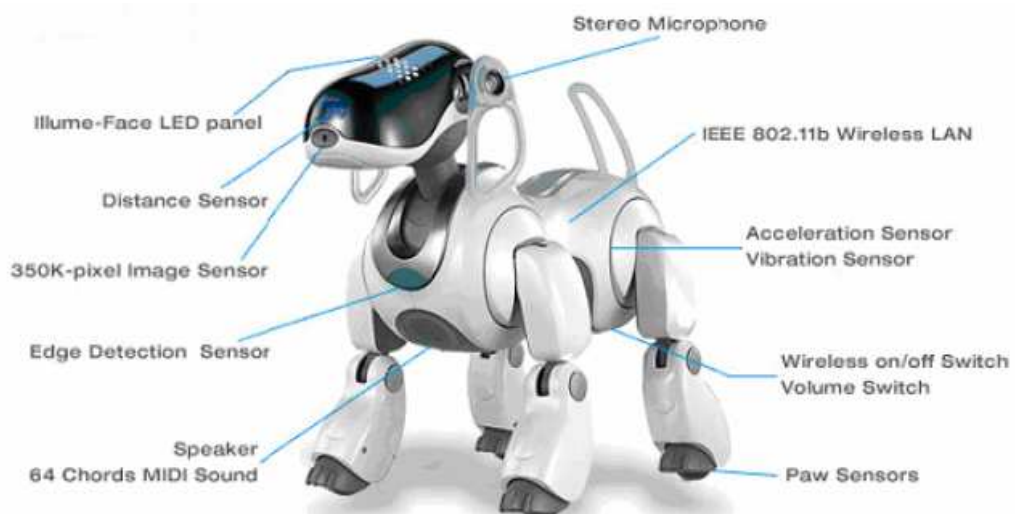
Το ρομπότ AIBO

Το AIBO είναι ένα αυτόνομο ρομπότ. Την παραγωγή του ξεκίνησε η SONY και εισήχθη στην αγορά το 1999 ως κατοικίδιο ρομπότ (robot pet). Το όνομα του πηγάζει από τις λέξεις τεχνητή νοημοσύνη AI (artificial intelligence) και ρομπότ (roBOt) και στα γιαπωνέζικα σημαίνει φίλος. Στα επόμενα χρόνια το AIBO εξελίχθηκε και άλλαξε μορφή αρκετές φορές. Αρχικά, η SONY κρατούσε κρυφή την απαραίτητη διασύνδεση με το hardware με αποτέλεσμα να είναι αδύνατον ανεξάρτητοι προγραμματιστές να γράψουν λογισμικό για το συγκεκριμένο ρομπότ. Το 2002 η SONY άλλαξε την πολιτική της και εξέδωσε το AIBO OPEN-R SDK, ένα πακέτο για ανάπτυξη λογισμικού για το AIBO. Από εκείνη την στιγμή το AIBO έγινε ιδιαίτερα δημοφιλές και κίνησε το ενδιαφέρον των πανεπιστημίων για ερευνητικούς ακαδημαϊκούς σκοπούς λόγω του ότι παρείχε μια ολοκληρωμένη ρομποτική πλατφόρμα πάνω στην οποία μπορούσε να γίνει ανάπτυξη λογισμικού και διεξαγωγή πειραμάτων με σχετικά μικρό κόστος.

2.1 Τεχνικά χαρακτηριστικά του AIBO

Η επικοινωνία του AIBO με το εξωτερικό περιβάλλον διασφαλίζεται με την χρήση αισθητήρων και επενεργητών. Ειδικότερα, το AIBO είναι εφοδιασμένο με έγχρωμη κάμερα που του παρέχει πληροφορία για τον περιβάλλοντα χώρο. Ακόμη, υπάρχουν αισθητήρες αφής στο κεφάλι, στην πλάτη, στο πιγούνι και στο κάτω μέρος των ποδιών. Επίσης διαθέτει τρεις αισθητήρες υπερύθρων για ανίχνευση απόστασης, μικρόφωνα στα αυτιά του και τρία επιταχυνσιόμετρα. Οι αντιδράσεις του AIBO γίνονται αντιληπτές μέσω ενός μεγαφώνου που βρίσκεται στο στήθος του, μέσω διαφόρων LEDs και μέσω μιας σειράς από κινούμενα μέρη, όπως κεφάλι, στόμα, πόδια, αυτιά και ουρά. Συνολικά το ρομπότ AIBO έχει είκοσι βαθμούς ελευθερίας, ή αρθρώσεις που μπορεί να κινήσει ανεξάρτητα. Ο

ενσωματωμένος υπολογιστής περιέχει επεξεργαστή 64-bit RISC στα 576MHz και RAM 64Mb. Τέλος, το AIBO είναι εφοδιασμένο με ασύρματη κάρτα δικτύου, η οποία επιτρέπει έλεγχο της κατάστασης του AIBO και σύνδεση με άλλες ηλεκτρονικές συσκευές, ώστε να επιτρέπεται η μεταφορά εντολών ελέγχου, εικόνων, βίντεο, αρχείων ήχου και άλλων μηνυμάτων.[1]



Σχήμα 2.1 : Ανατομία του AIBO

2.2 Προγραμματιστικά περιβάλλοντα

Τα προγραμματιστικά περιβάλλοντα τα οποία είναι διαθέσιμα για τον προγραμματισμό της συμπεριφοράς του AIBO είναι τα ακόλουθα:

- OPEN-R SDK
- Tekkotsu
- URBI

- Pyro
- Yart/RCodePlus
- R-Code

2.2.1 OPEN-R SDK

Το συγκεκριμένο περιβάλλον προγραμματισμού είναι βασισμένο στον μεταγλωττιστή gcc. Η ανάπτυξη λογισμικού γίνεται με χρήση της γλώσσας C++. Τα προγράμματα τα οποία δημιουργούνται με χρήση του OPEN-R SDK μπορούν να εκτελούνται είτε στο ίδιο το AIBO, είτε σε κάποιον απομακρυσμένο υπολογιστή.

Ένα πρόγραμμα το οποίο έχει αναπτυχθεί με χρήση του OPEN-R SDK αποτελείται από αντικείμενα τα οποία τρέχουν ταυτόχρονα. Η ανάπτυξη ενός προγράμματος απαιτεί την υλοποίηση όλων των απαραίτητων αντικειμένων που βοηθούν στον έλεγχο του προγράμματος. Αυτά τα αντικείμενα τρέχουν ανεξάρτητα μεταξύ τους, αλλά ανταλλάσσουν μηνύματα προκειμένου να συμπεριφερθούν συντονισμένα.

Το OPEN-R SDK προσφέρει συναρτήσεις διασύνδεσης με το υλικό του ρομπότ οι οποίες χρησιμοποιούνται για την κίνηση των αρθρώσεων, την λήψη πληροφοριών από τους αισθητήρες, την λήψη εικόνων από την κάμερα και την χρήση της ασύρματης επικοινωνίας. Τέλος, προσφέρει στον χρήστη την δυνατότητα υλοποίησης αντικειμένων τα οποία τρέχουν σε απομακρυσμένο ηλεκτρονικό υπολογιστή, ώστε να ελαττώνεται ο υπολογιστικός φόρτος του επεξεργαστή του AIBO [4].

2.2.2 Tekkotsu

Το περιβάλλον προγραμματισμού Tekkotsu δημιουργήθηκε στο Carnegie Mellon University, είναι υψηλότερου επιπέδου προγραμματιστικό περιβάλλον και έχει αναπτυχθεί πάνω στο περιβάλλον OPEN-R SDK [3]. Παρέχει στον προγραμματιστή εύκολη πρόσβαση στους αισθητήρες και στους μηχανισμούς του ρομπότ, όπως επίσης και επιλογές για συνεργασία με άλλα προγραμματιστικά εργαλεία, π.χ. Matlab. Το βασικό λειτουργικό σύστημα στο οποίο τρέχει είναι το Linux, αλλά μπορεί να τρέξει και κάτω από Windows με χρήση του Cygwin. [2][6]

2.2.3 URBI

Το περιβάλλον URBI (Universal Real-time Behaviour Interface) προσφέρει μια scripted command language η οποία χρησιμοποιείται για τον έλεγχο του ρομπότ[3]. Η λειτουργικότητα που προσφέρει το περιβάλλον URBI στηρίζεται στην αρχιτεκτονική client/server. Το URBI server είναι ουσιαστικά ένα OPEN-R αντικείμενο το οποίο μπορεί να χρησιμοποιηθεί είτε σε έναν ηλεκτρονικό υπολογιστή με χρήση ασύρματης επικοινωνίας, είτε κατευθείαν πάνω στο AIBO. Το συγκεκριμένο αντικείμενο επικοινωνεί με άλλα αντικείμενα του OPEN-R προκειμένου να εκτελεστεί μια εντολή. Τέλος, το περιβάλλον URBI παρέχει μια βιβλιοθήκη με την χρήση της οποίας επιτρέπεται η ανάπτυξη προγραμμάτων σε C++ [4][5].

2.2.4 Pyro

Το Pyro είναι ένα προγραμματιστικό περιβάλλον το οποίο επιτρέπει έλεγχο της

συμπεριφοράς του AIBO με χρήση της προγραμματιστικής γλώσσας Python. Η ανάπτυξη του συγκεκριμένου περιβάλλοντος έχει γίνει πάνω στο προγραμματιστικό περιβάλλον Tekkotsu, κατά συνέπεια απαραίτητη προϋπόθεση για την λειτουργία του είναι η εγκατάσταση του OPEN-R SDK και του Tekkotsu.

2.2.5 Yart/RCodePlus

Το περιβάλλον Yart/RCodePlus προσφέρει ένα γραφικό εργαλείο το οποίο παρέχει εύκολη δημιουργία συμπεριφορών για τον έλεγχο του ρομπότ. Η λειτουργικότητα του συγκεκριμένου περιβάλλοντος περιορίζεται από την απλότητα που προσφέρει. Ωστόσο, προσφέρει κάποια ενδιαφέροντα εργαλεία, όπως ασύρματη επικοινωνία και απομακρυσμένος έλεγχος του ρομπότ.

2.2.6 R-Code

Η R-Code είναι μια scripting γλώσσα προγραμματισμού, η οποία επιτρέπει προγραμματισμό πολύπλοκων συμπεριφορών του ρομπότ με λίγες εντολές. Δεν προσφέρει δυνατότητα για λεπτομερή και ακριβή έλεγχο του AIBO, αλλά μόνο για εκτέλεση προκαθορισμένων ενεργειών.

2.3 OPEN-R SDK

Το OPEN-R είναι η διεπαφή που παρέχει η SONY για τον προγραμματισμό συμπεριφοράς των ρομπότ προκειμένου να αυξηθούν οι δυνατότητές τους. Αυτή η διεπαφή είναι το μέσο για να γίνει αποδοτικότερη η χρήση του υλικού του ρομπότ μέσω ανάπτυξης λογισμικού.

Χαρακτηριστικά του OPEN-R λογισμικού

- Το OPEN-R λογισμικό είναι αντικειμενοστραφές και αρθρωτό. Οι μονάδες λογισμικού καλούνται αντικείμενα (τα αντικείμενα της C++ είναι διαφορετικά από τα αντικείμενα του OPEN-R. Με τον όρο αντικείμενο, στο εξής, θα αναφέρομαι σε OPEN-R αντικείμενο).

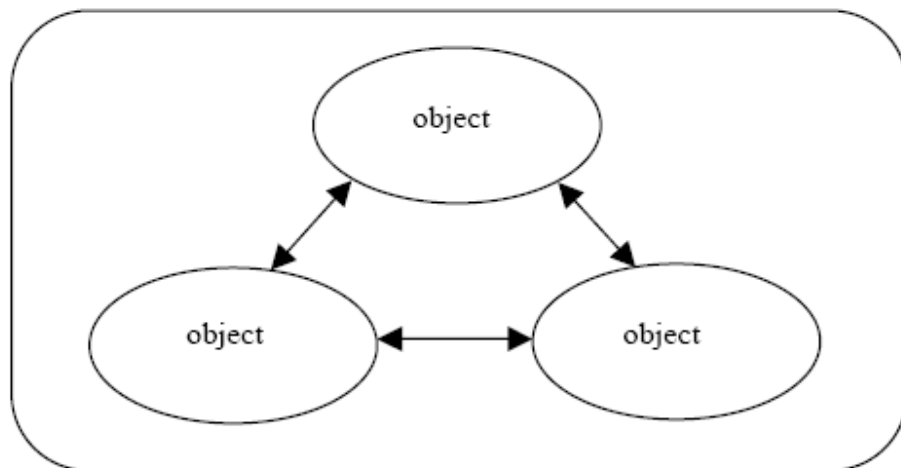
Το λογισμικό του ρομπότ αναπτύσσεται έτσι ώστε η διαδικασία της εκτέλεσης να πραγματοποιείται από πολλά αντικείμενα τα οποία τρέχουν ταυτόχρονα και επικοινωνούν μεταξύ τους με ανταλλαγή μηνυμάτων.

Ο τρόπος επικοινωνίας των αντικειμένων περιγράφεται σε ένα αρχείο. Όταν φορτώνεται το λογισμικό, φορτώνεται και το αρχείο που περιγράφει την επικοινωνία των αντικειμένων και βοηθά στον προσδιορισμό και στον κατάλληλο μετασχηματισμό των δρόμων επικοινωνίας μεταξύ των αντικειμένων. Οι πόρτες επικοινωνίας των αντικειμένων ορίζονται από το όνομα τους. Αυτό καθιστά τα αντικείμενα ως μονάδες λογισμικού εύκολα αντικαταστήσιμες.

- Το OPEN-R παρέχει ένα σύνολο υπηρεσιών ως διασύνδεση μεταξύ συστήματος και εφαρμογών. Αυτές οι υπηρεσίες επιτρέπουν υλοποίηση αντικειμένων και εφαρμογών χωρίς εξειδικευμένη γνώση του υλικού του ρομπότ. Μια από αυτές τις υπηρεσίες προσφέρει τη διασύνδεση για τη χρήση του TCP/IP πρωτοκόλλου, το οποίο προσφέρει στον χρήστη την δυνατότητα ανάπτυξης δικτυακών εφαρμογών με χρήση του ασυρμάτου δικτύου [9][10].

2.3.1 OPEN-R Object

Όπως προαναφέρθηκε, οι εφαρμογές λογισμικού σε OPEN-R αποτελούνται από διαφορετικά αντικείμενα. Η έννοια του αντικειμένου είναι ίδια με την έννοια των διεργασιών του λειτουργικού συστήματος.



Σχήμα 2.2: Εφαρμογή λογισμικού OPEN-R

Η ιδέα του αντικειμένου υφίσταται μόνο κατά τη διάρκεια της εκτέλεσης του προγράμματος. Κάθε αντικείμενο αντιστοιχεί σε ένα ισοδύναμο εκτελέσιμο αρχείο το οποίο δημιουργείται από τη μεταγλώττιση και τη σύνδεση του πηγαίου κώδικα. Στην συνέχεια το εκτελέσιμο αρχείο μεταφέρεται στο memory stick του AIBO, φορτώνεται όταν ξεκινά η λειτουργία του ρομπότ και εκτελείται αυτόνομα (τα εκτελέσιμα προγράμματα έχουν την κατάληξη .bin).

Ιδιαίτερα χαρακτηριστικά των αντικειμένων

- Τα αντικείμενα ανταλλάσσουν πληροφορία με ανταλλαγή μηνυμάτων

Ένα αντικείμενο μπορεί να στέλνει μηνύματα σε άλλα αντικείμενα. Κάθε μήνυμα περιλαμβάνει κάποια δεδομένα και έναν ακέραιο (selector), ο οποίος καθορίζει την εργασία που πρέπει να γίνει από τον παραλήπτη του μηνύματος. Όταν ένα αντικείμενο παραλάβει ένα μήνυμα ενεργοποιεί τη συνάρτηση που αντιστοιχεί στον selector και ως όρισμα της συνάρτησης λειτουργούν τα δεδομένα του μηνύματος. Μια συνάρτηση η οποία αντιστοιχεί σε κάποιον selector ονομάζεται μέθοδος.

Ένα άλλο ιδιαίτερο χαρακτηριστικό των αντικειμένων είναι ότι μπορούν να επεξεργάζονται μόνο ένα μήνυμα κάθε φορά. Εάν ένα αντικείμενο λάβει ένα μήνυμα τη στιγμή που επεξεργάζεται κάποιο άλλο, τότε το δεύτερο μήνυμα μπαίνει σε ουρά αναμονής και εκτελείται αργότερα μετά το πέρας του πρώτου.

Τυπικός κύκλος ζωής ενός αντικειμένου

1. Το αντικείμενο φορτώνεται από το σύστημα
2. Ενεργοποιείται
3. Περιμένει για κάποιο μήνυμα, είτε από κάποιον αισθητήρα είτε από άλλο αντικείμενο, ή στέλνει κάποιο μήνυμα
4. Όταν παραλάβει κάποιο μήνυμα εκτελεί την κατάλληλη μέθοδο
5. Όταν τελειώσει η εκτέλεση της μεθόδου το αντικείμενο επιστρέφει στο δεύτερο βήμα.

Το παραπάνω είναι ένας άπειρος βρόγχος. Το αντικείμενο δεν μπορεί να

τερματίσει από μόνο του, υπάρχει όσο το σύστημα είναι ενεργοποιημένο.

- Ένα αντικείμενο έχει πολλαπλά entry points

Σε αντίθεση με τα συνηθισμένα προγράμματα τα οποία έχουν ένα entry point, την main(), το OPEN-R επιτρέπει σε ένα αντικείμενο να έχει πολλαπλά entry points. Κάθε entry point αντιστοιχεί σε μια μέθοδο ενός selector. Κάποια entry points καθορίζονται από το σύστημα (αρχικοποίηση, τερματισμός) και κάποια άλλα καθορίζονται από τις λειτουργίες του αντικειμένου.

Κάθε αντικείμενο έχει και ένα ξεχωριστό entry point το Prologue(), το οποίο ξεκινά όταν φορτώνεται το αντικείμενο στο σύστημα και σκοπό έχει να κάνει κάποιες αρχικοποιήσεις και να καλέσει τους constructors των καθολικών μεταβλητών [9][10].

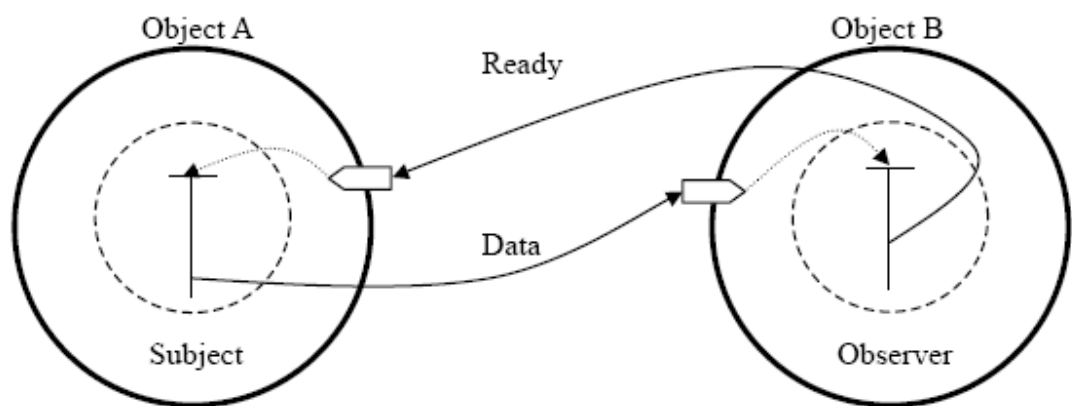
2.3.2 Δι-αντικειμενική επικοινωνία

Τα αντικείμενα παράλληλα με την εκτέλεση των λειτουργιών τους επικοινωνούν μεταξύ τους. Αυτή η επικοινωνία ονομάζεται δι-αντικειμενική επικοινωνία (inter-object communication).

- Επικοινωνία μεταξύ subjects και observers

Η χρήση της δι-αντικειμενικής επικοινωνίας δίνει την δυνατότητα σε κάθε αντικείμενο να δημιουργηθεί ξεχωριστά και αργότερα να συνδεθεί με τα

υπόλοιπα. Όταν δύο αντικείμενα επικοινωνούν, το αντικείμενο το οποίο στέλνει δεδομένα ονομάζεται subject, ενώ το αντικείμενο που τα παραλαμβάνει ονομάζεται observer. Το subject στέλνει ένα μήνυμα τύπου 'NotifyEvent' στο observer, το οποίο περιέχει τα δεδομένα τα οποία θέλει να στείλει. Από την άλλη μεριά το observer στέλνει ένα μήνυμα τύπου 'ReadyEvent', το οποίο μπορεί να φέρει την τιμή ASSERT_READY αν το αντικείμενο είναι έτοιμο να δεχτεί μήνυμα ή DEASSERT_READY αν δεν είναι έτοιμο, στο subject. Σκοπός του 'ReadyEvent' είναι να πληροφορηθεί το subject αν το observer είναι έτοιμο να δεχτεί δεδομένα, αν δεν είναι τότε δεν στέλνονται δεδομένα[9][10].

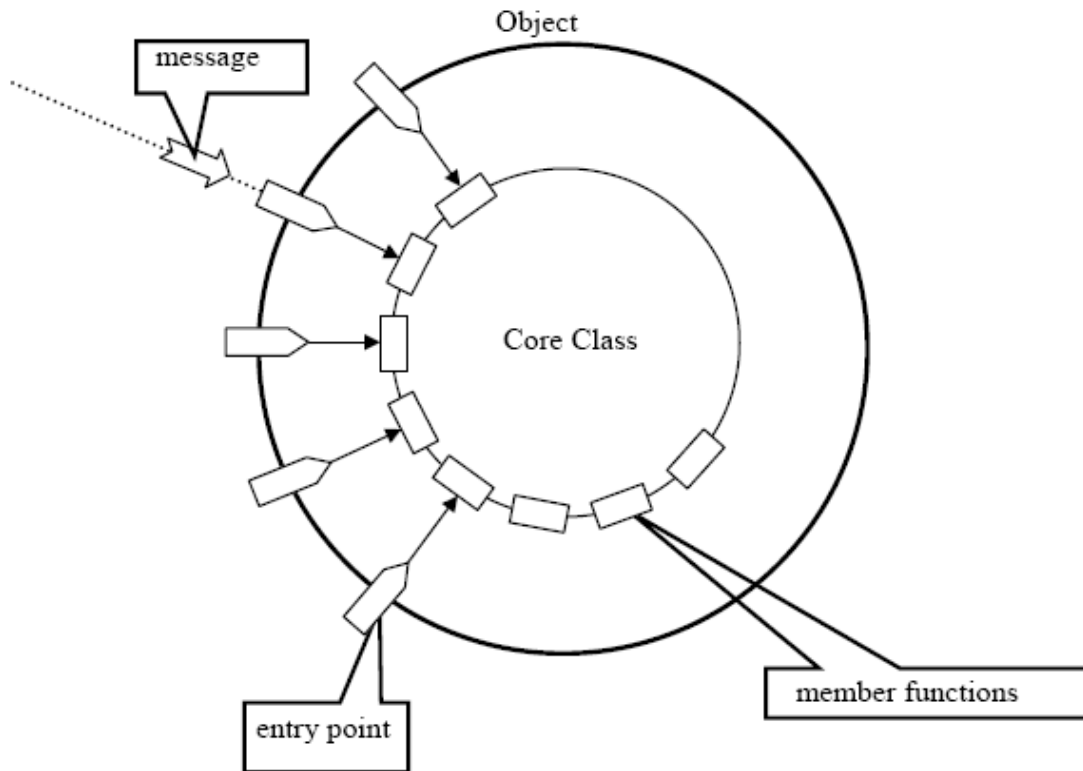


Σχήμα 2.3: Δι-αντικειμενική επικοινωνία

2.3.3 Βασική κλάση (Core class)

Η βασική κλάση είναι μια κλάση της C++, η οποία αναπαριστά ένα αντικείμενο. Κάθε αντικείμενο μπορεί να αναπαρίσταται μόνο από μια βασική κλάση. Τα αντικείμενα έχουν entry points για τη λήψη μηνυμάτων. Κάθε entry point αντιστοιχεί σε κάποια μέθοδο του αντικειμένου και κάθε μέθοδος αντιστοιχεί σε

μία συνάρτηση της βασικής κλάσης. Στο σχήμα που ακολουθεί φαίνεται η δομή μιας βασικής κλάσης.



Σχήμα 2.4: Βασική κλάση αντικειμένου

Χαρακτηριστικά μιας βασικής κλάσης

- Μια βασική κλάση κληρονομεί από την κλάση Oobject.
- Σε μια βασική κλάση πρέπει να υλοποιούνται οι συναρτήσεις DoInit(), DoStart(), DoStop() και DoDestroy().
- Στην κύρια κλάση περιέχονται οι μέθοδοι που είναι υπεύθυνες για την ανταλλαγή μηνυμάτων.

Μέθοδοι που χρησιμοποιούνται από το subject

- Control Method : λαμβάνει τα αποτελέσματα της σύνδεσης μεταξύ του subject και του observer.
- Ready Method : μέσω αυτής της μεθόδου το subject λαμβάνει ASSERT_READY ή DEASSERT_READY από τα observer αντικείμενα.

Μέθοδοι που χρησιμοποιούνται από το observer

- Connect Method : λαμβάνει τα αποτελέσματα της σύνδεσης μεταξύ του observer και του subject.
- Notify Method : λαμβάνει τα μηνύματα από τα subject αντικείμενα [9][10].

2.3.4 Ροή της ανάπτυξης ενός προγράμματος

1. Σχεδιασμός των αντικειμένων

Αφορά στο σχεδιασμό των συναρτήσεων των αντικειμένων και τη ροή των δεδομένων μεταξύ των αντικειμένων.

2. Σχεδιασμός του τύπου των δεδομένων των μηνυμάτων

Η επιλογή κατάλληλου τύπου δεδομένων στη μεταφορά μηνυμάτων, είναι ουσιαστικής σημασίας, αφού τα δεδομένα των μηνυμάτων παίζουν τον ρόλο των ορισμάτων των μεθόδων.

3. Περιγραφή του stub.cfg

Το αρχείο stub.cfg έχει συγκεκριμένη δομή και στόχος του είναι η αντιστοίχιση των entry points των αντικειμένων προκειμένου να γίνεται σωστά η ανταλλαγή μηνυμάτων.

4. Υλοποίηση της βασικής κλάσης

Η βασική κλάση εκτός από τις συναρτήσεις DoInit(), DoStart(), DoDestroy() και DoStop() περιέχει και υλοποιήσεις των μεθόδων που θα εκτελεί το αντικείμενο.

5. Διαμόρφωση του αρχείου .ocf

Το αρχείο αυτό βοηθά στην διαμόρφωση των αντικειμένων κατά την ώρα της εκτέλεσης.

6. Build

Αφορά στην μεταγλώττιση του πηγαίου κώδικα και στην σύνδεσή του με τις απαραίτητες βιβλιοθήκες για την δημιουργία ενός εκτελέσιμου αρχείου.

7. Σύνταξη των παρακάτω αρχείων

OBJECT.CFG : περιγράφει ποια αντικείμενα θα εκτελεστούν.

CONNECT.CFG : περιγράφει τις συνδέσεις μεταξύ των αντικειμένων.

DESIGNDB.CFG : περιγράφει τα αρχεία τα οποία θα προσπελαστούν από τα αντικείμενα την ώρα της εκτέλεσης.

8. Εκτέλεση στο AIBO

Μεταφέρονται τα απαραίτητα αρχεία στο memory stick του AIBO. Αφού το ρομπότ ενεργοποιηθεί, φορτώνει τα αρχεία και ξεκινά την εκτέλεση των αντικειμένων.

9. Αποσφαλμάτωση κώδικα

Η αποσφαλμάτωση του κώδικα γίνεται με χρήση των μακροεντολών OSYSRINT και OSYSLOG1, οι οποίες τυπώνουν μηνύματα σε τερματικό μέσω του ασυρμάτου δικτύου [9][10].

2.3.5 Ροή εκτέλεσης

1. Εισαγωγή των αντικειμένων στο memory stick

Τα αντικείμενα που έχουν δημιουργηθεί εισάγονται σε ένα memory stick, το οποίο έχει προετοιμαστεί και περιέχει τα αρχεία που απαιτεί το OPEN-R.

2. Ενεργοποίηση του AIBO

Το AIBO ενεργοποιείται, γίνεται φόρτωση του λειτουργικού συστήματος και των αντικειμένων που βρίσκονται στο memory stick.

3. Ενεργοποίηση των αντικειμένων

Τα αντικείμενα ενεργοποιούνται και ξεκινάνε την εκτέλεση των λειτουργιών τους [9][10].

2.4 Remote Processing Control (RPC)

Στο πρώτο κεφάλαιο έχω αναφέρει ότι τα προγράμματα βασισμένα σε OPEN-R αποτελούνται από διαφορετικά αντικείμενα τα οποία τρέχουν παράλληλα. Με την χρήση RPC κάποια αντικείμενα εκτελούνται σε ηλεκτρονικό υπολογιστή, ο οποίος συνδέεται με το ρομπότ μέσω ασυρμάτου δικτύου, και κάποια άλλα κατευθείαν πάνω στο AIBO. Όλα τα αντικείμενα τρέχουν ως ένα πρόγραμμα, αλλά είναι κατανεμημένα σε διαφορετικά μηχανήματα.

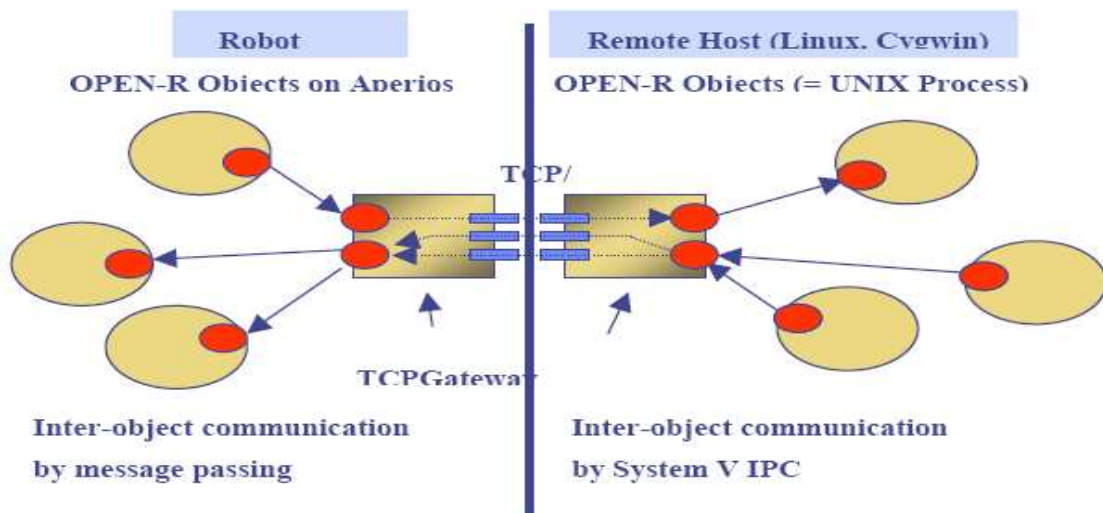
Οι αρχές που ακολουθούνται για τη συγγραφή αντικειμένων, είτε αυτά τρέχουν στο AIBO, είτε τρέχουν στον υπολογιστή, είναι οι ίδιες. Αυτό σημαίνει ότι υπάρχει συμβατότητα πηγαίου κώδικα μεταξύ των αντικειμένων, δεν υπάρχει όμως συμβατότητα μεταξύ των δυαδικών αρχείων που παράγονται από την μεταγλώττιση, γιατί χρησιμοποιούνται διαφορετικοί μεταγλωττιστές.

Το σημαντικότερο πλεονέκτημα της χρήσης RPC είναι η κατανομή υπολογιστικού φόρτου σε πολλά μηχανήματα. Μ' αυτό τον τρόπο υπολογιστικά απαιτητικές διαδικασίες τρέχουν στον υπολογιστή και λιγότερο απαιτητικές διαδικασίες στο ρομπότ. Ένα άλλο σημαντικό πλεονέκτημα είναι η χρήση βιβλιοθηκών και προγραμμάτων, που έχουν εγκατασταθεί στον υπολογιστή, χωρίς να είναι απαραίτητη η μεταγλώττιση τους ώστε να γίνουν συμβατά με το λειτουργικό σύστημα του AIBO.

Κατά τη διάρκεια λειτουργίας του RPC χρησιμοποιούνται τα αντικείμενα *TCPGateway* τόσο στο ρομπότ όσο και στον υπολογιστή. Τα *TCPGateway* αντικείμενα είναι υπεύθυνα για την επικοινωνία των αντικειμένων που εκτελούνται στο ρομπότ με τα αντικείμενα που εκτελούνται στον υπολογιστή. Η επικοινωνία γίνεται μέσω του ασυρμάτου δικτύου και χρησιμοποιείται το πρωτόκολλο TCP.

Τα αντικείμενα τα οποία εκτελούνται στον υπολογιστή, ουσιαστικά έχουν τα ίδια χαρακτηριστικά με τις διεργασίες του UNIX. Στο Σχήμα 2.5 φαίνεται η

επικοινωνία των αντικειμένων με χρήση RPC [10].



Σχήμα 2.5: Δι-αντικειμενική επικοινωνία σε RPC

2.5 Μηχανή μετακίνησης του AIBO

Η μηχανή μετακίνησης του AIBO που χρησιμοποιήθηκε είναι μια εφαρμογή ανοιχτού λογισμικού που στόχο έχει την μετακίνηση του ρομπότ προς κάθε κατεύθυνση. Η συγκεκριμένη εφαρμογή αναπτύχθηκε σε OPENR-SDK και λειτουργεί ως αυτόνομο αντικείμενο, το οποίο εκτελείται στο ρομπότ.

Η εφαρμογή παρέχει διεπαφή γραμμής εντολών και για την εκτέλεσή της απαιτείται η παρουσία χρήστη, ο οποίος από ένα απομακρυσμένο τερματικό μέσω ασύρματης επικοινωνίας στέλνει εντολές κίνησης στο ρομπότ.

Αρχικά, ο χρήστης στέλνει εντολές, ώστε το ρομπότ να βρεθεί σε μια αρχική θέση, π.χ όρθια στάση ή στάση ύπνου. Στη συνέχεια μπορεί να στείλει εντολές

μετακίνησης προς κάθε κατεύθυνση, να καθορίσει την απόσταση και την ταχύτητα κίνησης και το μήκος του κάθε βήματος. Τέλος, η εφαρμογή παρέχει βοήθεια για την χρήση της κάθε εντολής [11].

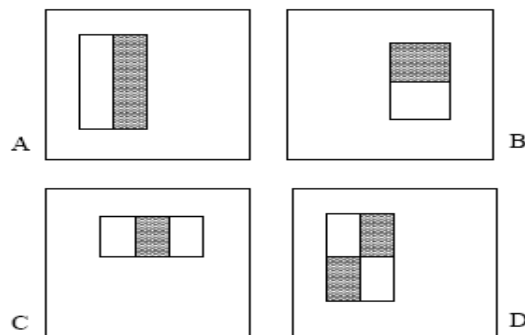
Κεφάλαιο 3

Η βιβλιοθήκη OpenCV

Η OpenCV είναι μια βιβλιοθήκη ελεύθερου λογισμικού η οποία αναπτύχθηκε από την Intel και αφορά στην επεξεργασία εικόνας. Η OpenCV παρέχει μία μεσαίου έως υψηλού επιπέδου διασύνδεση εφαρμογών με περίπου τριακόσιες συναρτήσεις γραμμένες σε C και μερικές κλάσεις C++. Η ανάπτυξη επικοινωνίας ανθρώπου με υπολογιστή, η ανίχνευση, απομόνωση και αναγνώριση αντικειμένων, η ανίχνευση και αναγνώριση προσώπων, η κατανόηση και παρακολούθηση κίνησης είναι μερικά από τα πεδία μηχανικής όρασης που καλύπτει. Οι αλγόριθμοι της OpenCV είναι βελτιστοποιημένοι για επεξεργαστές αρχιτεκτονικής Intel Pentium (MMX, Pro, 3, 4). Τέλος, η δημιουργία της OpenCV αποσκοπεί και στην δημιουργία μιας κοινότητας ανοιχτού λογισμικού, σχετική με την μηχανική όραση, η οποία θα αναπτύσσει σύγχρονες μεθόδους επεξεργασίας εικόνας σε ένα συνεχώς αναπτυσσόμενο τεχνολογικό περιβάλλον.

3.1 OpenCV και ανίχνευση αντικειμένων

Η ανίχνευση αντικειμένων στην OpenCV στηρίζεται σε ένα σύστημα μηχανικής μάθησης το οποίο είναι ικανό να επεξεργάζεται πολύ γρήγορα μια εικόνα και να πετυχαίνει πολύ υψηλά ποσοστά επιτυχούς ανίχνευσης. Η διαδικασία της ανίχνευσης είναι πολύ πιο αποδοτική όταν βασίζεται στην ανίχνευση χαρακτηριστικών (features), τα οποία κωδικοποιούν πληροφορία σχετικά με το αντικείμενο που πρόκειται να ανιχνευτεί. Τα χαρακτηριστικά αυτά χρησιμοποιούνται σε σύνολα και κωδικοποιούν πληροφορία που αφορά την αντίθεση φωτεινότητας ανάμεσα σε περιοχές της εικόνας, τον προσανατολισμό της αντίθεσης και την χωρική σχέση αυτών των περιοχών.



*Σχήμα 3.1 : Παραδείγματα χαρακτηριστικών
Τα A και B είναι δι-πολυγωνικά, το C τρι-πολυγωνικό
και το D τετρα-πολυγωνικό*

Στην διαδικασία της ανίχνευσης υπάρχουν τρία είδη χαρακτηριστικών, τα δι-πολυγωνικά, τα τρι-πολυγωνικά και τα τετρα-πολυγωνικά χαρακτηριστικά. Η τιμή των δι-πολυγωνικών χαρακτηριστικών είναι η διαφορά ανάμεσα στο σύνολο των pixels που βρίσκονται μέσα στις δύο πολυγωνικές περιοχές. Η τιμή ενός τρι-πολυγωνικού χαρακτηριστικού υπολογίζεται αφαιρώντας από το άθροισμα των pixels των εξωτερικών πολυγώνων, το άθροισμα των pixels του εσωτερικού πολυγώνου. Τέλος η τιμή των τετρα-πολυγωνικών χαρακτηριστικών υπολογίζεται από την διαφορά των pixels ανάμεσα στα διαγώνια μέρη του πολυγώνου. Στο Σχήμα 3.1 φαίνεται ένα παράδειγμα τέτοιων χαρακτηριστικών.

Για τον γρήγορο υπολογισμό των χαρακτηριστικών, σε οποιαδήποτε κλίμακα ή περιοχή, το σύστημα δεν χρησιμοποιεί κατευθείαν τις τιμές φωτεινότητας της εικόνας, αλλά μια ενδιάμεση αναπαράστασή της, που ονομάζεται πλήρη εικόνα. Η πλήρης εικόνα μπορεί να υπολογιστεί κατευθείαν από την αρχική εικόνα με χρήση απλών πράξεων στις τιμές των pixels της. Συγκεκριμένα, ο τρόπος υπολογισμού της πλήρους εικόνας γίνεται με την χρήση της σχέσης :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

όπου $ii(x, y)$ η τιμή της πλήρους εικόνας στο σημείο (x, y) και $i(x, y)$ η τιμή της πραγματικής εικόνας στο ίδιο σημείο. Η πλήρης εικόνα μπορεί να υπολογιστεί με ένα μόνο πέρασμα πάνω από την πραγματική εικόνα.

3.2 Διαδικασία μάθησης

Η διαδικασία ανίχνευσης αντικειμένων που χρησιμοποιεί η OpenCV αρχικά προτάθηκε από τον Paul Viola και υλοποιήθηκε από τον Rainer Lienhart. Αρχικά ένας ταξινομητής εκπαιδεύεται με ένα μεγάλο αριθμό θετικών δειγμάτων (εικόνες που περιέχουν το αντικείμενο προς ανίχνευση και έχουν όλες το ίδιο μέγεθος) και με αρνητικά δείγματα (αυθαίρετες εικόνες που δεν περιέχουν το αντικείμενο και είναι ίδιου μεγέθους με τα θετικά δείγματα).

Από κάθε θετικό δείγμα εξάγεται ένας πολύ μεγάλος αριθμός χαρακτηριστικών. Πειραματικά έχει αποδειχθεί ότι ένα πολύ μικρό σύνολο αυτών των χαρακτηριστικών είναι αρκετό για την δημιουργία ενός πολύ αποδοτικού ταξινομητή. Για την εξαγωγή αυτών των χαρακτηριστικών χρησιμοποιείται ένας αλγόριθμος μάθησης ο οποίος έχει σχεδιαστεί να επιλέγει μόνο τα πολυγωνικά χαρακτηριστικά τα οποία ανταποκρίνονται καλύτερα στα θετικά και στα αρνητικά δείγματα.

Ο αλγόριθμος μάθησης που χρησιμοποιήθηκε είναι ο AdaBoost σε συνδυασμό με naive Bayes. Ο AdaBoost είναι ένας προσαρμοστικός αλγόριθμος με την έννοια ότι οι μεταγενέστεροι ταξινομητές που δημιουργεί ανταποκρίνονται καλύτερα στα

θετικά και αρνητικά δείγματα σε σχέση με τους προγενέστερους. Ο αλγόριθμος καλείται επαναληπτικά για μια σειρά T κύκλων. Σε κάθε κύκλο η κατανομή βαρών των χαρακτηριστικών ανανεώνεται έτσι ώστε να ξεχωρίζουν τα χαρακτηριστικά με βάση την σημαντικότητα τους στην διαδικασία της ταξινόμησης. Η περιγραφή του αλγορίθμου φαίνεται στο Σχήμα 3.2.

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:
 1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$
 so that w_t is a probability distribution.
 2. For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
 3. Choose the classifier, h_t , with the lowest error ϵ_t .
 4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
 where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
 where $\alpha_t = \log \frac{1}{\beta_t}$

Σχήμα 3.2: Περιγραφή του αλγορίθμου AdaBoost για τη εκπαίδευση ενός ταξινομητή

3.3 Χαρακτηριστικά για αναγνώριση προσώπων

Στο Σχήμα 3.3 φαίνεται ένα παράδειγμα εξαγωγής δύο χαρακτηριστικών από ένα θετικό δείγμα. Το πρώτο χαρακτηριστικό προκύπτει από την διαφορά φωτεινότητας ανάμεσα στα μάτια και στα ζυγωματικά ενός ανθρώπινου

προσώπου. Το δεύτερο χαρακτηριστικό προκύπτει από την διαφορά φωτεινότητας ανάμεσα στα μάτια και στην περιοχή ανάμεσα στα μάτια, από το μέτωπο ως την μύτη. Άλλα χαρακτηριστικά μπορούν να προκύψουν από γειτονικές περιοχές της εικόνας, οι οποίες έχουν διαφορά φωτεινότητας μεγαλύτερη από ένα προκαθορισμένο κατώφλι.

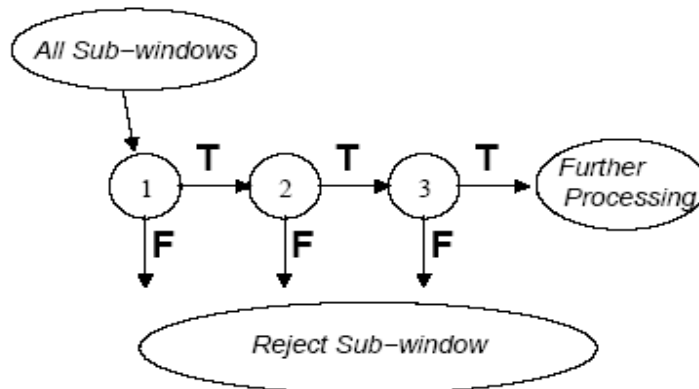


Σχήμα 3.3: Παράδειγμα εξαγωγής χαρακτηριστικών

3.4 Διαδικασία ανίχνευσης

Από τη στιγμή που έχουν δημιουργηθεί οι ταξινομητές μπορεί να ξεκινήσει η διαδικασία ανίχνευσης αντικειμένων σε μια εικόνα. Η ανίχνευση αντικειμένων γίνεται με εφαρμογή ταξινομητών στην εικόνα. Η εφαρμογή των ταξινομητών έχει έξοδο 1 αν η περιοχή στην οποία εφαρμόστηκε μοιάζει να περιέχει το αντικείμενο προς αναγνώριση, αλλιώς η έξοδος είναι 0. Για την ανίχνευση σε όλη την εικόνα ο ταξινομητής εξετάζει την εικόνα σειριακά με ένα κυλιόμενο παράθυρο. Επίσης, ο ταξινομητής έχει σχεδιαστεί έτσι ώστε εύκολα να μπορεί να μεταβάλλει το μέγεθος του παραθύρου που εξετάζει προκειμένου να μπορεί να ανιχνεύει αντικείμενα διαφορετικών μεγεθών. Το μέγεθος του υποπαραθύρου μεταβάλλεται κάθε φορά που εξετάζεται όλη η εικόνα και με συγκεκριμένο ρυθμό. Ο ρυθμός μεταβολής του μεγέθους του υποπαραθύρου ορίζεται από τον προγραμματιστή ή από τον χρήστη.

Η διαδικασία της ανίχνευσης σε μια εικόνα χωρίζεται σε στάδια. Αρχικά στην εικόνα εφαρμόζονται οι πιο απλοί ταξινομητές ώστε να απορριφθεί η πλειοψηφία των υποπαραθύρων της εικόνας τα οποία δεν περιέχουν το αντικείμενο προς ανίχνευση και στη συνέχεια εφαρμόζονται ταξινομητές πιο περίπλοκοι που σκοπό έχουν την μείωση του ποσοστού λάθους ανίχνευσης. Αν κάποιο υποπάρθυρο της εικόνας απορριφθεί κατά την διαδικασία της ανίχνευσης δεν επανεξετάζεται από πιο περίπλοκους ταξινομητές. Στο Σχήμα 3.4 φαίνεται η διαδικασία ανίχνευσης αντικειμένου [7][8].



Σχήμα 3.4: Απεικόνιση της διαδικασίας ανίχνευσης αντικειμένου

Κεφάλαιο 4

Σχετικές Εργασίες

4.1 Όραση και κίνηση ως μέσα αλληλεπίδρασης ρομπότ-ανθρώπου

Ο τομέας της μηχανικής όρασης είναι ένας από τους ταχύτατα αναπτυσσόμενους τομείς της επιστήμης των υπολογιστών. Η ανάπτυξη εφαρμογών, οι οποίες έχουν σκοπό την ανάλυση, επεξεργασία και κατανόηση της εικόνας, αυξάνεται συνεχώς. Αυτό οφείλεται, κυρίως, στο πολύ μεγάλο μέγεθος πληροφορίας που μπορεί να περιέχει μια εικόνα σε σχέση με άλλες μορφές δεδομένων, όπως ήχος ή κείμενο. Αυτή η ιδιότητα της εικόνας, την καθιστά σημαντικό αντικείμενο εμπορικών εφαρμογών, αλλά και απαραίτητο στοιχείο για την ουσιαστική αλληλεπίδραση και επικοινωνία ανάμεσα σε άνθρωπο και υπολογιστή και ανάμεσα σε άνθρωπο και ρομπότ.

Το ενδιαφέρον της αγοράς για δημιουργία οικιακών ρομπότ συνεχώς μεγαλώνει. Προκειμένου τα ρομπότ να μπορέσουν να λειτουργήσουν ικανοποιητικά σε σπίτια και σε μη βιομηχανικά περιβάλλοντα, ο τρόπος με τον οποίο θα αλληλεπιδρούν με τον άνθρωπο και θα εκπαιδεύονται ώστε να φέρνουν σε πέρας μια συγκεκριμένη εργασία είναι αποφασιστικής σημασίας. Οι άνθρωποι οι οποίοι θα αλληλεπιδρούν με τα ρομπότ, ίσως να μην έχουν καμία σχέση με ρομποτική, γι' αυτόν το λόγο ο τρόπος αλληλεπίδρασης θα πρέπει να είναι εξαιρετικά διαισθητικός. Ένας τέτοιος τρόπος αλληλεπίδρασης μπορεί να επιτευχθεί με ανάπτυξη της όρασης των ρομπότ, ώστε να είναι σε θέση να αναγνωρίζουν πρόσωπα, χειρονομίες και αντικείμενα και να διαμορφώνουν την συμπεριφορά τους με βάση αυτό που βλέπουν.

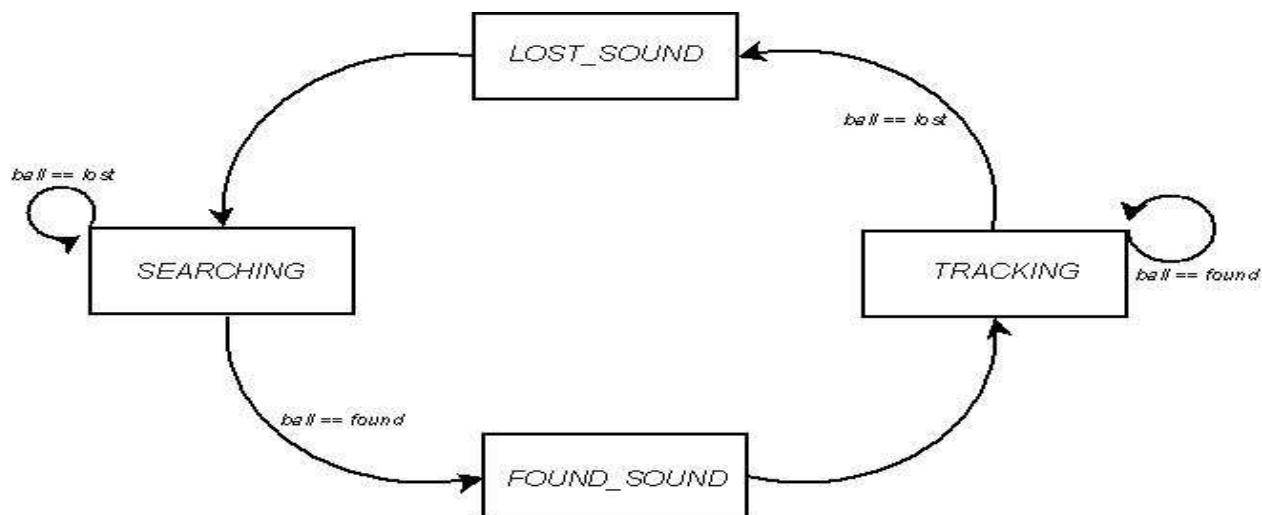
Έρευνες σχετικά με την όραση και την κίνηση των ρομπότ έχουν γίνει πάρα πολλές τα τελευταία χρόνια. Στην συνέχεια του κεφαλαίου θα παρουσιάσω κάποιες εφαρμογές που αφορούν στο παραπάνω θέμα και οι οποίες αναπτύχθηκαν για να εκτελούνται στο οικιακό ρομπότ AIBO ERS-7 της Sony.

- BallTracking

Είναι μια εφαρμογή, η οποία αναπτύχθηκε από την Sony για εκπαιδευτικούς σκοπούς και στόχο έχει την ανίχνευση και παρακολούθηση από το AIBO, με περιστροφή του κεφαλιού, μιας μπάλας συγκεκριμένου χρώματος. Η ανάπτυξη της εφαρμογής έγινε στο προγραμματιστικό περιβάλλον OPENR-SDK.

Το ρομπότ λαμβάνει εικόνα χρησιμοποιώντας την κάμερα με την οποία είναι εφοδιασμένο. Η εικόνα που λαμβάνει είναι εικόνα ανίχνευσης χρώματος. Τέτοιου είδους εικόνες ουσιαστικά είναι κατάλογοι με την χρωματική τιμή του κάθε pixel, αυτός είναι και ο λόγος για τον οποίο η μπάλα πρέπει να έχει συγκεκριμένο χρώμα, παρέχονται κατευθείαν από την κάμερα του AIBO και το OPENR-SDK παρέχει συναρτήσεις που επιτρέπουν την πρόσβαση και την επεξεργασία τους. Ο αλγόριθμος ανίχνευσης που χρησιμοποιείται αναζητά ομάδες γειτονικών pixels με το ίδιο χρώμα. Η ανίχνευση θεωρείται επιτυχής αν ο αριθμός των pixels μιας ομάδας είναι μεγαλύτερος από ένα συγκεκριμένο κατώφλι.

Το AIBO, κατά την εκτέλεση της εφαρμογής, αρχικά κουνά τα πόδια και το κεφάλι του σε μια αρχική θέση. Στην συνέχεια κουνά το κεφάλι του κυκλικά αναζητώντας τη συγκεκριμένη μπάλα και όταν την ανιχνεύσει την παρακολουθεί. Επίσης, παίζει έναν συγκεκριμένο ήχο κάθε φορά που ανιχνεύει ή χάνει την μπάλα από το οπτικό του πεδίο. Οι καταστάσεις στις οποίες μπορεί να βρεθεί το ρομπότ κατά την εκτέλεση της εφαρμογής φαίνονται στο Σχήμα 4.1.



Σχήμα 4.1 : Καταστάσεις του AIBO κατά την εκτέλεση της εφαρμογής *Balltracking*

Όπως προανέφερα, η εφαρμογή *BallTracking* δημιουργήθηκε για εκπαιδευτικούς σκοπούς και γι' αυτό το λόγο είναι και σχετικά απλή. Για την ανάπτυξή της υλοποιήθηκε ένα αντικείμενο για την κίνηση του κεφαλιού, ένα για την κίνηση των ποδιών και ένα για την διαδικασία της αναζήτησης και ανίχνευσης. Ο αλγόριθμος για την ανίχνευση της μπάλας, απλά, αναζητά την ύπαρξη ενός συγκεκριμένου χρώματος μέσα στην εικόνα. Αν μόνο η μπάλα έχει το συγκεκριμένο χρώμα η ανίχνευση γίνεται σωστά, αν όμως υπάρχουν και άλλα αντικείμενα του ίδιου χρώματος ο αλγόριθμος μπορεί να αποτύχει, εφόσον δεν αναγνωρίζει σχήμα. Η κίνηση του ρομπότ είναι εξίσου απλή. Οι μόνες κινήσεις που πραγματοποιεί είναι κίνηση του κεφαλιού δεξιά-αριστερά και πάνω-κάτω και κίνηση των ποδιών σε μια αρχική θέση.

Η συγκεκριμένη εφαρμογή δεν μπορεί να αποτελεί εργαλείο αλληλεπίδρασης μεταξύ ανθρώπου και υπολογιστή, όμως η μελέτη και η κατανόησή της είναι απαραίτητες για τον σχεδιασμό και την υλοποίηση πολυπλοκότερων εφαρμογών.

- Αναγνώριση προσώπου και ομιλίας

Η παρακάτω εφαρμογή υλοποιήθηκε στο πανεπιστήμιο TUDelft τον Οκτώβριο του 2005 από την ερευνητική ομάδα του καθηγητή L.Rothkrantz που ασχολείται με την αλληλεπίδραση ανάμεσα σε άνθρωπο και μηχανή. Στόχος της εφαρμογής ήταν η ανάπτυξη κατάλληλου λογισμικού για το ρομπότ AIBO, το οποίο θα επιτυγχάνει ανίχνευση και αναγνώριση προσώπων και αναγνώριση φωνής [6].

Η διαδικασία αναγνώρισης προσώπου όπως και η διαδικασία αναγνώρισης φωνής μπορούν να μοντελοποιηθούν σαν διαδικασίες αρχικά εξαγωγής και στην συνέχεια ταύτισης χαρακτηριστικών.

Η διαδικασία αναγνώρισης προσώπου χωρίζεται σε δύο μέρη. Αρχικά πρέπει να ανιχνευτεί ανθρώπινο πρόσωπο και στη συνέχεια να γίνει έλεγχος αν το πρόσωπο που ανιχνεύθηκε ανήκει στην βάση δεδομένων που περιέχει τα “γνωστά” πρόσωπα. Η ανίχνευση προσώπου γίνεται με χρήση της βιβλιοθήκης OpenCV. Αν ανίχνευση είναι επιτυχής, τότε η περιοχή του προσώπου απομονώνεται με τέτοιο τρόπο ώστε τα μάτια να βρίσκονται σε συγκεκριμένα σημεία μέσα στην εικόνα. Στην συνέχεια ο έλεγχος για ταύτιση του προσώπου με ένα από τα γνωστά πρόσωπα γίνεται με την τεχνική των *Eigenfaces*, αφού πρώτα μεταβληθεί το μέγεθος της εικόνας σύμφωνα με τις απαιτήσεις του αλγορίθμου.

Η αναγνώριση ανθρώπινης φωνής έγινε στο περιβάλλον Matlab με αρχικά απομόνωση και στην συνέχεια ανάλυση του σήματος της φωνής. Τόσο η διαδικασία αναγνώρισης φωνής όσο και η διαδικασία αναγνώρισης προσώπου απαιτούν εκπαίδευση του συστήματος και για την ανίχνευση και απομόνωση φωνής και πρόσωπο, αλλά και για την ταυτοποίηση τους.

Η παραπάνω εφαρμογή υλοποιήθηκε στο προγραμματιστικό περιβάλλον Tekkotsu,

έγινε χρήση του Matlab και της C++ και οι διαδικασίες ανίχνευσης και ταυτοποίησης εκτελούνται σε απομακρυσμένο ηλεκτρονικό υπολογιστή, ο οποίος επικοινωνεί με το ρομπότ μέσω ασυρμάτου δικτύου.

Σε αντίθεση με το σύστημα το οποίο αναπτύχθηκε στην παρούσα εργασία, η παραπάνω εφαρμογή δεν ασχολείται καθόλου με τις δυνατότητες κίνησης του ρομπότ.

- Ανίχνευση κίνησης

Η συγκεκριμένη εργασία έχει αναπτυχθεί για το ρομπότ AIBO στο περιβάλλον OPENR-SDK. Αρχικά το ρομπότ κινείται ώστε να βρεθεί σε όρθια στάση με το κεφάλι στη θέση μηδέν. Η ανίχνευση κίνησης γίνεται λαμβάνοντας διαδοχικές εικόνες από την κάμερα και αφαιρώντας τις τιμές των pixels της κάθε μίας από τις τιμές των αντίστοιχων pixels της προηγούμενης. Αν η απόλυτη τιμή του αποτελέσματος της αφαίρεσης για κάποια pixels είναι μεγαλύτερη από κάποιο προκαθορισμένο κατώφλι, τότε στην περιοχή που βρίσκονται αυτά τα pixels έχει παρουσιαστεί κίνηση.

Κατά την εκτέλεση της συγκεκριμένης εφαρμογής το ρομπότ δεν εκτελεί κάποια κίνηση, εκτός από αυτή που απαιτείται ώστε να βρεθεί σε όρθια στάση. Όσον αφορά την όραση του AIBO η διαδικασία επεξεργασίας εικόνας που υλοποιείται δεν αποτελεί κάτι πολύπλοκο.

Αν και απλή η παραπάνω εφαρμογή, εκτός από εκπαιδευτικό χαρακτήρα, μπορεί να έχει και ρόλο δομικού στοιχείου για μια πολυπλοκότερη εφαρμογή, που στόχο έχει τη διαμόρφωση συμπεριφοράς του ρομπότ με τέτοιο τρόπο ώστε να αναγνωρίζει συγκεκριμένα πρόσωπα, να ανιχνεύει τυχόν κίνησή τους και να τα

ακολουθεί.

Κεφάλαιο 5

Προσέγγιση του προβλήματος

5.1 Ορισμός του προβλήματος

Στόχος της εργασίας ήταν η ανάπτυξη λογισμικού για το AIBO, το οποίο θα κάνει χρήση των δυνατοτήτων κίνησης και της κάμερας του ρομπότ, έτσι ώστε να επιτυγχάνεται η ανίχνευση και η παρακολούθηση ανθρώπινου προσώπου με κατάλληλη κίνηση του ρομπότ ώστε το πρόσωπο να φαίνεται enface.

5.2 Όραση του AIBO

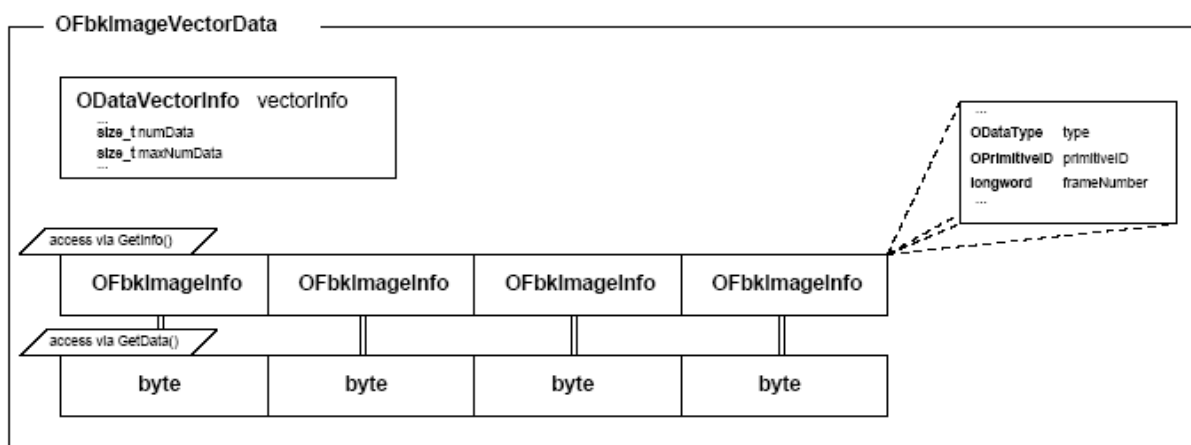
Το AIBO είναι εξοπλισμένο με μία έγχρωμη κάμερα, η οποία μπορεί να λαμβάνει 30 εικόνες το δευτερόλεπτο διαστάσεων 208x160. Η κάμερα χρησιμοποιεί αισθητήρα τεχνολογίας CMOS μεγέθους 0.25in και ευρυγώνιο φακό διαφράγματος $f/2,8$ που της επιτρέπει να καταγράφει πεδίο 56,9 μοιρών στον οριζόντιο άξονα και 45,2 μοιρών στον κατακόρυφο. Οι προκαθορισμένες τιμές της εξισορρόπησης λευκού και της ταχύτητας φωτοφράχτη είναι 5000 βαθμοί Kelvin και 1/100 του δευτερολέπτου αντίστοιχα. Ανάλογα με τις συνθήκες φωτισμού, η εξισορρόπηση λευκού μπορεί να ρυθμιστεί στους 2856 ή στους 6500 βαθμούς Kelvin. Αντίστοιχα, ανάλογα με την ένταση του φωτός, η ταχύτητα φωτοφράχτη μπορεί να ρυθμιστεί στο 1/50 ή 1/200 του δευτερολέπτου [1].

5.2.1 Λήψη εικόνας με χρήση OPEN-R

Το AIBO λαμβάνει πληροφορία από την κάμερα σε τέσσερις διαφορετικές εικόνες. Οι τρεις είναι έγχρωμες εικόνες διαφορετικής ανάλυσης και η τέταρτη μια εικόνα για ανίχνευση χρώματος. Οι έγχρωμες εικόνες χρησιμοποιούν την χρωματική παλέτα YCrCb, το οποίο σημαίνει ότι υπάρχουν τρία κανάλια για την

αναπαράσταση της εικόνας: το Y (φωτεινότητα), το Cr (κόκκινη συνιστώσα – φωτεινότητα) και το Cb (μπλε συνιστώσα – φωτεινότητα). Η εικόνα για ανίχνευση χρώματος είναι απλά ένας δισδιάστατος πίνακας με την χρωματική τιμή του κάθε pixel.

Η εικόνα μεταφέρεται από την κάμερα στο αντικείμενο OPEN-R το οποίο είναι αρμόδιο για την επεξεργασία της εικόνας μέσω μηνυμάτων. Τα δεδομένα που μεταφέρουν τα μηνύματα είναι τύπου *OFbkImageVectorData*. Τα δεδομένα τύπου *OFbkImageVectorData* είναι μια δομή με τρία πεδία. Το πρώτο πεδίο είναι τύπου *ODataVectorInfo*, το δεύτερο πεδίο είναι ένας πίνακας *OFbkImageInfo* και το τρίτο ένας πίνακας *bytes*. Οι πίνακες των *OFbkImageInfo* και *bytes* χωρίζονται σε τέσσερα μέρη και κάθε μέρος του πίνακα *OFbkImageInfo* αντιστοιχίζεται με ένα μέρος του πίνακα *bytes*. Τα τέσσερα μέρη του πίνακα *OFbkImageInfo* αντιστοιχούν στα τέσσερα είδη εικόνων (τρεις εικόνες διαφορετικής ανάλυσης και μια εικόνα για ανίχνευση χρώματος), ενώ τα τέσσερα μέρη του πίνακα *bytes*, είναι δείκτες που αντιστοιχούν στα δεδομένα των παραπάνω εικόνων. Στο Σχήμα 5.1 φαίνεται η οργάνωση της δομής τύπου *OFbkImageVectorData*.



Σχήμα 5.1: Η δομή *OFbkImageVectorData*

Το περιβάλλον OPEN-R παρέχει μια κλάση για διαχείριση εικόνων. Με χρήση των συναρτήσεων της κλάσης επιτυγχάνεται πρόσβαση στα δεδομένα της εικόνας. Συγκεκριμένα, για τη λήψη έγχρωμης εικόνας απαιτείται λήψη τριών μονοχρωματικών εικόνων (Y , Cr και Cb εικόνα) και στην συνέχεια η συνένωσή τους. Η τελική έγχρωμη εικόνα δεν ανήκει σε κάποιον επίσημο τύπο εικόνων (π.χ. BMP ή JPEG) και το περιβάλλον OPEN-R δεν παρέχει συναρτήσεις ή κλάσεις για την μετατροπή της εικόνας σε εικόνα κάποιου επίσημου τύπου ούτε και συναρτήσεις ή κλάσεις για την διαχείριση τους [9].

5.3 Μετατροπή της εικόνας του AIBO σε τύπου BMP

Για να είναι δυνατή η πρόσβαση και η επεξεργασία των δεδομένων των εικόνων, που λαμβάνονται από την κάμερα του AIBO, από εξωτερικές βιβλιοθήκες ή εξωτερικά προγράμματα είναι απαραίτητη η μετατροπή τους σε εικόνες κάποιου επίσημου τύπου. Η μετατροπή των εικόνων έγινε σε τύπο *Bitmap* (BMP). Επιλέχθηκε ο συγκεκριμένος τύπος γιατί η μετατροπή είναι σχετικά εύκολη και δεν υπάρχει απώλεια πληροφορίας. Για την μετατροπή της εικόνας σε τύπο BMP υλοποιήθηκαν συναρτήσεις που ολοκληρώνουν την διαδικασία σε δύο βήματα.

Το πρώτο βήμα περιλαμβάνει την μετατροπή της χρωματικής παλέτας YCrCb της εικόνας στην χρωματική παλέτα RGB. Αρχικά για την μετατροπή αυτή έγινε κανονικοποίηση των τιμών του καναλιού Y της εικόνας από 0 έως 1 και των τιμών των καναλιών Cr και Cb από -1 έως 1. Στην συνέχεια με χρήση των κανονικοποιημένων τιμών των καναλιών Y, Cr και Cb υπολογίστηκαν οι τιμές για τα κανάλια R, G, B. Οι σχέσεις που χρησιμοποιήθηκαν για τους υπολογισμούς είναι οι παρακάτω:

- ◆ Κόκκινο κανάλι $R = 255 * (Y + Cr)$
- ◆ Πράσινο κανάλι $G = 255 * (Y - 0.51 * Cr - 0.19 * Cb)$
- ◆ Μπλε κανάλι $B = 255 * (Y + Cb)$

Το δεύτερο βήμα αρχικά περιλαμβάνει την δημιουργία κεφαλίδας τύπου BMP, η οποία περιέχει πληροφορία σχετικά με την εικόνα που πρόκειται να αποθηκευτεί, και την εγγραφή της σε ένα αρχείο. Στην συνέχεια την εγγραφή στο αρχείο των δεδομένων των καναλιών R, G και B και τέλος την αποθήκευση του αρχείου με κατάληξη BMP.

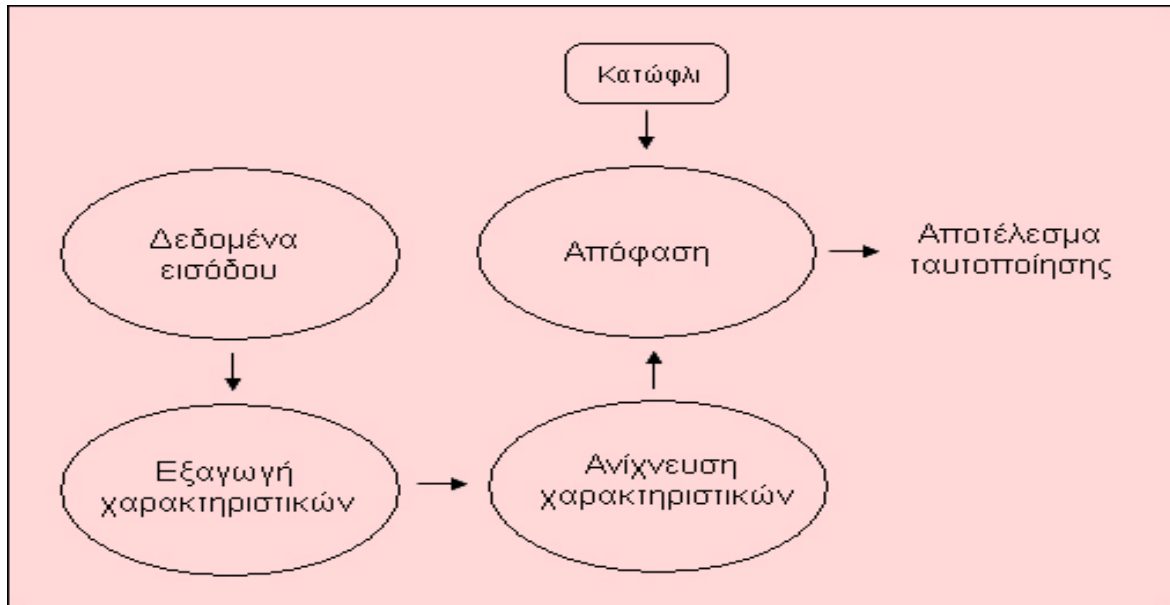
Μετά το πέρας της διαδικασίας μετατροπής της εικόνας που λαμβάνει η κάμερα του AIBO σε τύπο BMP μπορεί να ξεκινήσει η διαδικασία για ανίχνευση ανθρώπινου προσώπου.

5.4 Ανίχνευση προσώπου

Η διαδικασία της ανίχνευσης προσώπου μπορεί να οργανωθεί σε δύο στάδια. Το πρώτο είναι το στάδιο της *εξαγωγής χαρακτηριστικών* και το δεύτερο το στάδιο της *ανίχνευσης χαρακτηριστικών*.

Η *εξαγωγή χαρακτηριστικών* είναι η διαδικασία εξαγωγής ενός μικρού αριθμού δεδομένων από μία εικόνα προσώπου με τέτοιο τρόπο ώστε αυτά τα χαρακτηριστικά να μπορούν να αναπαραστήσουν το πρόσωπο.

Η *ανίχνευση χαρακτηριστικών* είναι η διαδικασία ανίχνευσης και ταυτοποίησης, με χρήση κατάλληλου κατωφλιού, των χαρακτηριστικών που έχουν εξαχθεί από εικόνες προσώπων και των χαρακτηριστικών της άγνωστης εικόνας. Ολοκληρωμένη η διαδικασία ανίχνευσης προσώπου φαίνεται στο Σχήμα 5.2[6].



Σχήμα 5.2: Μοντέλο αναγνώρισης προσώπου

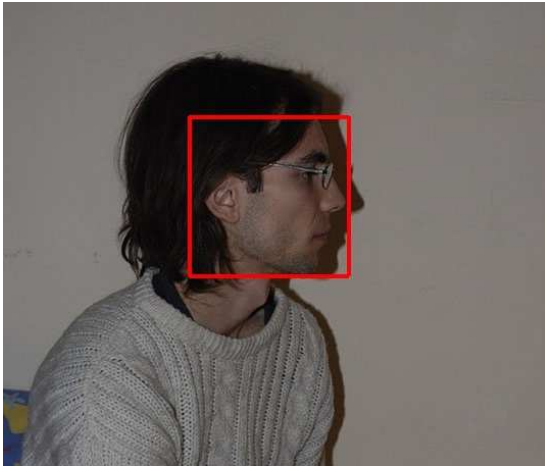
5.5 Χρήση της βιβλιοθήκης OpenCV

Το πρώτο πράγμα που έπρεπε να υλοποιηθεί ήταν ένα πρόγραμμα βασισμένο στην βιβλιοθήκη OpenCV το οποίο θα μπορούσε να κάνει ανίχνευση προσώπου. Το πρόγραμμα, όπως έχω προαναφέρει, πρέπει να μπορεί να ανιχνεύει πρόσωπα και προφίλ και enface, άρα για την υλοποίηση του απαραίτητη είναι η δημιουργία δύο ταξινομητών. Ο ένας θα είναι υπεύθυνος για την ανίχνευση προσώπων προφίλ και ο άλλος για την ανίχνευση προσώπων enface. Η OpenCV προσφέρει για ακριβώς αυτό το σκοπό δύο έτοιμους ταξινομητές, τους οποίους και χρησιμοποίησα για την υλοποίηση του προγράμματος. Ο ένας ταξινομητής είναι υπεύθυνος για την ανίχνευση προσώπου σε θέση enface και ο δεύτερος για ανίχνευση προσώπου σε θέση αριστερού προφίλ. Για την ανίχνευση προσώπων σε θέση δεξιού προφίλ χρησιμοποίησα τον δεύτερο ταξινομητή κατοπτρίζοντας την εικόνα στον

κατακόρυφο άξονα.

Το πρόγραμμα αρχικά φορτώνει τους δύο ταξινομητές από δύο αρχεία XML στα οποία βρίσκονται αποθηκευμένοι και διαβάζει την εικόνα στην οποία θα γίνει η διαδικασία της ανίχνευσης. Στην συνέχεια εφαρμόζεται ο αλγόριθμος ανίχνευσης αντικειμένων, στην συγκεκριμένη περίπτωση ανθρώπινα πρόσωπα προφίλ και enface. Ο αλγόριθμος αποθηκεύει σε ένα πίνακα πληροφορία για την χωρική θέση των πετυχημένων ανιχνεύσεων, αν υπάρχουν, μέσα στην εικόνα. Τέλος, χρησιμοποιείται η πληροφορία που αποθηκεύτηκε προκειμένου να σημειωθούν τα πρόσωπα που ανιχνεύθηκαν. Μερικά αποτελέσματα από την χρήση του προγράμματος φαίνονται παρακάτω [8].





Παραδείγματα ανίχνευσης προσώπων σε θέσης προφίλ και enface

5.6 OpenCV και AIBO

Όπως φαίνεται και παραπάνω το πρόγραμμα ανίχνευσης προσώπων προφίλ και enface λειτουργεί πολύ ικανοποιητικά. Το επόμενο βήμα είναι η συνεργασία του παραπάνω προγράμματος με το ρομπότ. Η συνεργασία αυτή μπορεί να επιτευχθεί με δύο τρόπους, είτε με εκτέλεση του προγράμματος πάνω στο ρομπότ, είτε στέλνοντας εικόνες από το ρομπότ σε έναν υπολογιστή και στέλνοντας ο υπολογιστής πίσω στο ρομπότ τα αποτελέσματα του προγράμματος ανίχνευσης (remote processing control).

Αρχικά θεωρήσαμε ότι το πρόγραμμα έπρεπε να εκτελείται πάνω στο ρομπότ, ώστε το AIBO να λειτουργεί πλήρως αυτόνομα. Οι πιθανότητες να μην είναι εφικτό κάτι τέτοιο ήταν πολύ μεγάλες λόγω των περιορισμένων υπολογιστικών δυνατοτήτων του ρομπότ (ταχύτητα επεξεργαστή 576MHz και 64MB ram) και λόγω του ότι η OpenCV είναι βελτιστοποιημένη για επεξεργαστές αρχιτεκτονικής Intel Pentium, ενώ το AIBO χρησιμοποιεί επεξεργαστή 64-bit αρχιτεκτονικής RISC. Επειδή όμως υπήρχαν, έστω και λίγες, πιθανότητες το AIBO να μπορεί να λειτουργεί αυτόνομα, χρησιμοποιώντας το παραπάνω πρόγραμμα, ξεκινήσαμε διαδικασίες που σκοπό είχαν να κάνουν συμβατή την OpenCV με το υλικό και το λειτουργικό σύστημα του AIBO.

5.6.1 Χρήση της OpenCV σε πλήρως αυτόνομη λειτουργία του ρομπότ

Το πρώτο βήμα που έπρεπε να γίνει, προκειμένου να γίνει συμβατή η OpenCV με το λειτουργικό και υλικό σύστημα του AIBO, ήταν να μεταγλωττιστεί η βιβλιοθήκη με τον δια-μεταγλωττιστή (*mipsel-linux-g++*) που χρησιμοποιεί το ρομπότ.

Για να μπορέσει να ολοκληρωθεί χωρίς λάθη η διαδικασία της μεταγλώττισης, έπρεπε να ενημερωθεί ο κατάλογος των βιβλιοθηκών (*include directory*), που χρησιμοποιεί ο μεταγλωττιστής, προσθέτοντας σε αυτόν τις βιβλιοθήκες που χρειάζεται η OpenCV για να μεταγλωττιστεί. Μετά την πρόσθεση των βιβλιοθηκών στον κατάλογο βιβλιοθηκών του μεταγλωττιστή η διαδικασία της μεταγλώττισης μπορεί να ξεκινήσει. Κάθε αρχείο πηγαίου κώδικα της βιβλιοθήκης OpenCV μεταγλωττίζεται και γίνεται αρχείο αντικειμένου (*object file, filename.o*). Τέλος, όλα τα μεταγλωττισμένα αρχεία συγκεντρώνονται σε αρχεία τύπου *archive* (*.a), μέσα στα οποία βρίσκονται οι υλοποιήσεις όλων των συναρτήσεων που

χρησιμοποιούνται από την OpenCV.

Όπως έχω ήδη προαναφέρει, οι εικόνες οι οποίες λαμβάνονται από την κάμερα του ρομπότ μετατρέπονται σε τύπο BMP. Για αυτό τον λόγο αφαίρεσα από την OpenCV αρχεία στα οποία υλοποιούνται συναρτήσεις για χρήση και επεξεργασία εικόνων διαφορετικού τύπου (*jpeg, jpeg-2000, png, tiff*).

Για να ελεγχθεί, η συμβατότητα της OpenCV με το AIBO υλοποίησα απλά προγράμματα επεξεργασίας εικόνας. Συγκεκριμένα υλοποίησα ένα πρόγραμμα το οποίο κάνει ανίχνευση ακμών με την μέθοδο Sobel, ένα πρόγραμμα που κάνει smoothing σε μία εικόνα και ένα πρόγραμμα το οποίο μετατρέπει μία έγχρωμη εικόνα σε μαυρόασπρη. Τα αποτελέσματα των προγραμμάτων φαίνονται παρακάτω.



Original image



Η εικόνα μετά την μετατροπή της

από RGB σε Grayscale



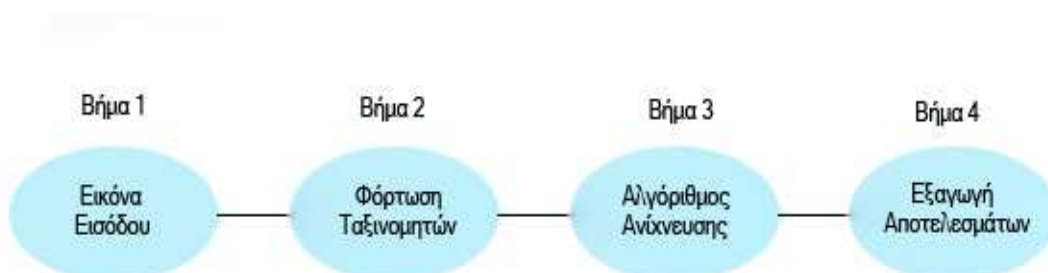
*Η εικόνα μετά την εφαρμογή του
αλγορίθμου για Smoothing*



*Η εικόνα μετά την εφαρμογή του
αλγορίθμου για εύρεση ακμών με
την μέθοδο Sobel*

5.6.2 Ανίχνευση προσώπου σε πλήρως αυτόνομη λειτουργία του ρομπότ

Έχω ήδη αναφέρει πως λειτουργεί ένα πρόγραμμα ανίχνευσης αντικειμένων υλοποιημένο με χρήση της OpenCV, εδώ θα παραθέσω σχηματικά τα στάδια από τα οποία περιέρχεται.



Μετά την μετατροπή της εικόνας που δέχεται η κάμερα του AIBO από YCrCb σε RGB και τύπο BMP η εικόνα μπορεί να επεξεργαστεί από την βιβλιοθήκη OpenCV και με αυτόν τον τρόπο το πρώτο βήμα διεκπεραιώνεται.

Οι ταξινομητές είναι δύο δενδρικές δομές δεδομένων, οι οποίες, όπως έχω προαναφέρει, είναι αποθηκευμένες σε δύο αρχεία τύπου XML. Η OpenCV για να φορτώσει τους ταξινομητές χρησιμοποιεί συναρτήσεις που αφορούν στην επεξεργασία συμβολοσειρών, προκειμένου να μπορέσει να πάρει από τα αρχεία XML τις πληροφορίες που χρειάζεται, και συναρτήσεις δημιουργίας και επεξεργασίας δενδρικών δομών, για να φορτώσει τις πληροφορίες. Η χρήση των παραπάνω συναρτήσεων γίνεται αναδρομικά.

Η διαδικασία φόρτωσης των ταξινομητών σε πλήρως αυτόνομη λειτουργία του ρομπότ αποτύγχανε, εξαιτίας της περιορισμένης μνήμης του AIBO, και το ρομπότ οδηγούνταν σε απενεργοποίηση. Για να μπορέσει να λυθεί το παραπάνω πρόβλημα έπρεπε να προχωρήσω σε αποσφαλμάτωση του προγράμματος και βελτιστοποίηση του κώδικα. Αυτό αποτελεί ιδιαίτερα χρονοβόρα και κουραστική διαδικασία λόγω του ότι το προγραμματιστικό περιβάλλον OPEN-R SDK δεν παρέχει κάποιον αξιόλογο αποσφαλματωτή. Ο μόνος τρόπος για να γίνει αποσφαλματοποίηση κάποιου προγράμματος είναι με χρήση εντολών του συστήματος, όπως η *fprintf()*, σε συγκεκριμένα σημεία του κώδικα. Αυτή η διαδικασία έχει αποτελέσματα, όταν το πρόγραμμα δεν χρησιμοποιεί αναδρομικές συναρτήσεις. Οι συναρτήσεις, όμως, που χρησιμοποιούνται για την φόρτωση των ταξινομητών καλούνται αναδρομικά μεταξύ τους, πράγμα που κάνει την αποσφαλμάτωση αδύνατη.

Μια λύση για το παραπάνω πρόβλημα, η οποία όμως ξεφεύγει από τα όρια της εργασίας μου, είναι η σειριοποίηση του αναδρομικού μέρους του προγράμματος και η βελτιστοποίηση του για τον επεξεργαστή του AIBO. Δεύτερη λύση, η οποία

και επιλέχθηκε, είναι η εγκατάλειψη της ιδέας για πλήρως αυτόνομη λειτουργία του ρομπότ και η δημιουργία ενός αντικειμένου OPEN-R το οποίο θα τρέχει σε υπολογιστή, θα συνεργάζεται ασύρματα με το ρομπότ και θα είναι υπεύθυνο για την διαδικασία της ανίχνευσης προσώπου.

5.7 Αρχιτεκτονική του προγράμματος και τα αντικείμενα OPEN-R

Όπως έχω αναφέρει στο Κεφάλαιο 2 τα προγράμματα που υλοποιούνται στο περιβάλλον OPEN-R αποτελούνται από μια συλλογή αντικειμένων. Τα αντικείμενα αυτά εκτελούνται παράλληλα και έχουν την δυνατότητα να επικοινωνούν μεταξύ τους με ανταλλαγή μηνυμάτων.

Το πρόγραμμα που υλοποίησα, ώστε να καλύπτει τις απαιτήσεις της εργασίας, έχει την ακόλουθη συμπεριφορά: αρχικά το AIBO κουνά το κεφάλι και τα πόδια του σε μια αρχική θέση, το κεφάλι στην θέση (0,0,0) και τα πόδια με τέτοιο τρόπο ώστε να βρεθεί σε όρθια στάση, και στη συνέχεια κουνά το κεφάλι του ελέγχοντας αν στην περιοχή γύρω του υπάρχει ανθρώπινο πρόσωπο. Αν το ρομπότ ανιχνεύσει ανθρώπινο πρόσωπο κινεί το κεφάλι του με τέτοιο τρόπο, ώστε να τοποθετήσει το πρόσωπο στο κέντρο της εικόνας, που λαμβάνει με την κάμερά του. Αν το πρόσωπο που ανίχνευσε είναι enface προσπαθεί να το διατηρεί τοποθετημένο στο κέντρο της εικόνας, ενώ εάν είναι προφίλ ξεκινά μια διαδικασία κίνησης, χωρίς να χάσει οπτική επαφή με το πρόσωπο, ώστε να το δει enface.

Τα αντικείμενα τα οποία υλοποιήθηκαν για την ανάπτυξη αυτού του προγράμματος είναι το παρακάτω:

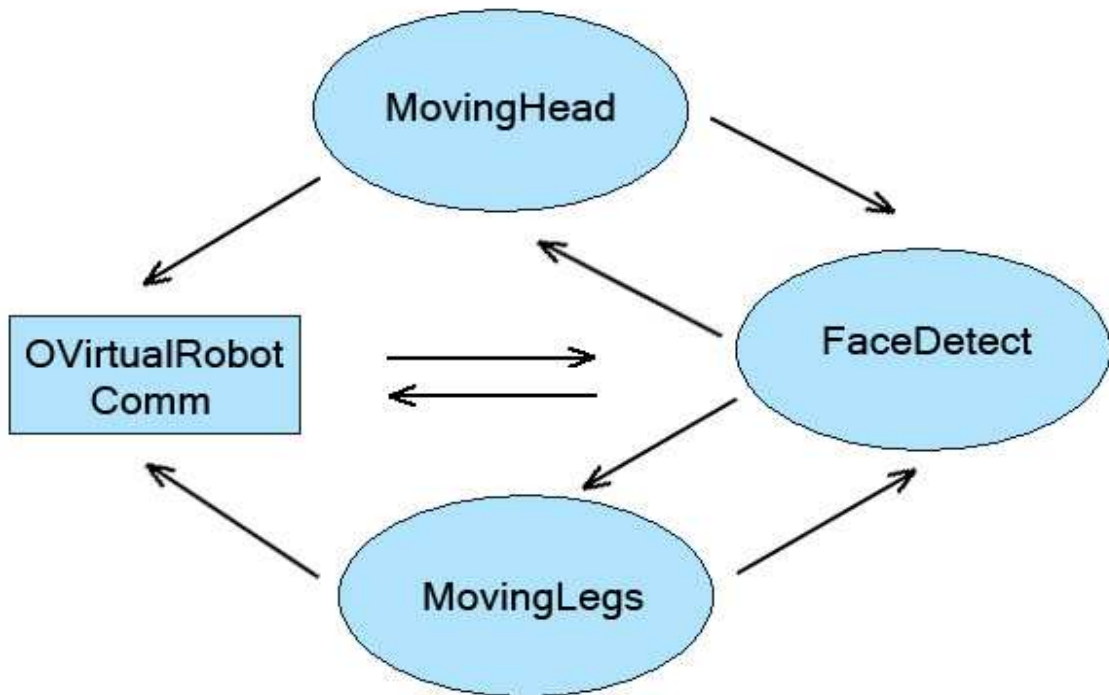
- ◆ *MovingHead* : είναι υπεύθυνο για όλες τις κινήσεις του κεφαλιού,

- ◆ *MovingLegs* : είναι υπεύθυνο για την κίνηση του ρομπότ, δεξιά ή αριστερά, ώστε να έρθει σε θέση που βλέπει το πρόσωπο enface,
- ◆ *FaceDetect* : είναι υπεύθυνο για την διαδικασία ανίχνευσης προσώπου και εκτελείται σε απομακρυσμένο υπολογιστή.

Παράλληλα με τα αντικείμενα τα οποία υλοποίησα, το πρόγραμμα χρησιμοποιεί και κάποια άλλα αντικείμενα τα οποία παρέχει έτοιμα το περιβάλλον OPEN-R και χρησιμοποιούνται για πρόσβαση στο υλικό του ρομπότ. Αυτό σημαίνει ότι οι αισθητήρες και οι μηχανισμοί κίνησης του AIBO ελέγχονται μέσα από τα έτοιμα αντικείμενα που παρέχει το σύστημα. Τα αντικείμενα αυτά είναι τα:

- ◆ *OVirtualRobotComm* : αποτελεί την διασύνδεση με τους αισθητήρες, τους μηχανισμούς κίνησης, την κάμερα και τα LEDs του ρομπότ,
- ◆ *TCPGateway* : είναι το ενδιάμεσο αντικείμενο για την ασύρματη επικοινωνία των αντικειμένων που εκτελούνται στο ρομπότ με αυτά τα οποία εκτελούνται σε απομακρυσμένο υπολογιστή.

Η αρχιτεκτονική του προγράμματος φαίνεται στο Σχήμα 5.3. Τα βέλη συμβολίζουν την επικοινωνία μεταξύ των αντικειμένων, τα αντικείμενα που παριστάνονται με ελλείψεις είναι τα αντικείμενα που υλοποίησα και το αντικείμενο που παριστάνεται με ορθογώνιο είναι αντικείμενο του συστήματος. Το αντικείμενο *FaceDetect* εκτελείται σε απομακρυσμένο υπολογιστή, άρα σε κάθε επικοινωνία του με τα υπόλοιπα αντικείμενα μεσολαβεί το αντικείμενο *TCPGateway*.



Σχήμα 5.3 : Αρχιτεκτονική του προγράμματος και δι-αντικειμενική επικοινωνία

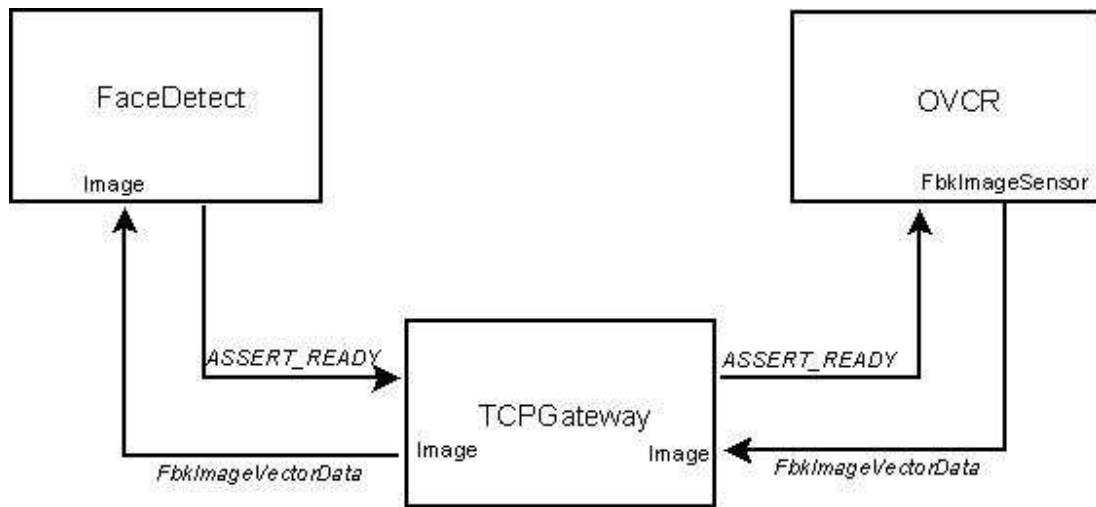
5.7.1 Ανάλυση του αντικειμένου FaceDetect

Το αντικείμενο *FaceDetect* είναι υπεύθυνο για την ανίχνευση προσώπου. Για να μπορέσει να ξεκινήσει η διαδικασία ανίχνευσης προσώπου το αντικείμενο *FaceDetect* πρέπει να δεχτεί εικόνα από το ρομπότ. Το αντικείμενο που αναλαμβάνει να στείλει την εικόνα από το ρομπότ είναι το *OVirtualRobotComm* το οποίο είναι αντικείμενο του συστήματος του OPEN-R. Το *OVirtualRobotComm* εκτελείται πάνω στο AIBO, ενώ το *FaceDetect* εκτελείται στον υπολογιστή, επομένως χρειάζεται ένα επιπλέον αντικείμενο που να μπορεί να μεταφέρει το μήνυμα που περιέχει τα δεδομένα της εικόνας από το ρομπότ στον υπολογιστή. Αυτό το ρόλο έχει αναλάβει το αντικείμενο *TCPGateway* το οποίο επίσης αποτελεί αντικείμενο του συστήματος.

Η ακολουθία ανταλλαγής μηνυμάτων μεταξύ των παραπάνω αντικειμένων φαίνεται στα παρακάτω βήματα:

- Βήμα 1ο : Αποστολή μηνύματος από το *FaceDetect* στο *OVirtualRobotComm*, με το οποίο επιβεβαιώνεται ότι το *FaceDetect* είναι έτοιμο να δεχτεί εικόνα.
- Βήμα 2ο : Αν το *OVirtualRobotComm* έχει λάβει επιβεβαίωση από το *FaceDetect* ότι μπορεί να δεχτεί μήνυμα, τότε αποστέλλει μήνυμα με τα δεδομένα της εικόνας. Αν δεν έχει ληφθεί επιβεβαίωση, τότε το πρόγραμμα επιστρέφει στο πρώτο βήμα.
- Βήμα 3ο : Το *FaceDetect* δέχεται την εικόνα και την επεξεργάζεται. Όταν τελειώσει την επεξεργασία επιστρέφει το πρόγραμμα στο βήμα 1, όπου το *FaceDetect* επιβεβαιώνει ότι μπορεί να δεχτεί επόμενη εικόνα.

Όπως έχω ήδη αναφέρει, όλα τα παραπάνω μηνύματα ανάμεσα στον υπολογιστή και στο ρομπότ μεταφέρονται μέσω του αντικειμένου *TCPGateway*. Η πύλη, η οποία δημιουργήθηκε στο αντικείμενο *FaceDetect* για την λήψη των μηνυμάτων που περιέχουν τα δεδομένα των εικόνων, ονομάστηκε *Image*. Τα δεδομένα που λαμβάνονται, προφανώς, είναι τύπου *OFbkImageVectorData*. Με το ίδιο όνομα ονομάστηκε και η πύλη του αντικειμένου *TCPGateway* μέσω της οποίας μεταβιβάζονται τα μηνύματα. Η επικοινωνία των δύο αντικειμένων σχηματικά φαίνεται παρακάτω.



Σχήμα 5.4 : Σχηματική απεικόνιση της δι-αντικειμενικής επικοινωνίας και της ανταλλαγής μηνυμάτων για λήψη εικόνας

Η λειτουργία του αντικειμένου *FaceDetect* δεν σταματά απλά στην λήψη και επεξεργασία εικόνας. Όταν το αντικείμενο επεξεργαστεί μια εικόνα πρέπει να είναι σε θέση να μπορεί να δώσει σαφείς οδηγίες στο αντικείμενο που είναι υπεύθυνο για την κίνηση του κεφαλιού (*MovingHead*) για το πως ακριβώς θα κινηθεί.

Οι περιπτώσεις στις οποίες μπορεί να καταλήξει η διαδικασία επεξεργασίας εικόνας είναι οι ακόλουθες:

- ◆ Περίπτωση 1η : Στην εικόνα δεν υπάρχει πρόσωπο,
- ◆ Περίπτωση 2η : Στην εικόνα υπάρχει ένα πρόσωπο και
- ◆ Περίπτωση 3η : Στην εικόνα υπάρχουν περισσότερα από ένα πρόσωπα.

Στην πρώτη περίπτωση η οδηγία που πρέπει να σταλεί στο αντικείμενο *MovingHead* είναι να κουνήσει το κεφάλι με τέτοιο τρόπο ώστε να ελέγξει το

χώρο γύρω του για πιθανή ύπαρξη προσώπων. Στην δεύτερη περίπτωση δίνεται εντολή στο *MovingHead* να κουνήσει το κεφάλι με τέτοιο τρόπο ώστε το πρόσωπο που ανιχνεύθηκε να τοποθετηθεί στο κέντρο της εικόνας. Στην τρίτη περίπτωση δεν αποστέλλεται κάποια εντολή στο *MovingHead* οπότε το ρομπότ παραμένει ακίνητο.

Τα αποτελέσματα της επεξεργασίας της εικόνας κωδικοποιούνται σε έναν πίνακα τριών θέσεων. Η τρίτη θέση του πίνακα παίρνει τις τιμές μηδέν και ένα, για τις περιπτώσεις ένα και δύο αντίστοιχα. Οι δύο πρώτες θέσεις του πίνακα, εφόσον βρισκόμαστε στην περίπτωση δύο, περιέχουν τιμές των γωνιών που πρέπει να κινηθεί το κεφάλι του ρομπότ, τόσο στον οριζόντιο, όσο και στον κατακόρυφο άξονα, ώστε το κέντρο του προσώπου να βρεθεί στο κέντρο της εικόνας. Ο πίνακας αυτός αποστέλλεται από το *FaceDetect* στο *MovingHead* και ανάλογα με τις τιμές που περιέχει εκτελούνται οι κατάλληλες εντολές.

Για τον υπολογισμό της απόστασης του κέντρου του προσώπου από το κέντρο της εικόνας υλοποίησα μια συνάρτηση η οποία εκτελείται από το αντικείμενο *FaceDetect*. Όπως παρατηρούμε από την εκτέλεση του προγράμματος ανίχνευσης προσώπου, όταν ένα πρόσωπο ανιχνεύεται το πρόγραμμα συγκρατεί πληροφορία για τις μέγιστες και ελάχιστες τεταγμένες και τετμημένες του χώρου μες στον οποίο βρίσκεται το πρόσωπο, ώστε να μπορέσει να δημιουργήσει ένα ορθογώνιο μέσα στο οποίο θα το συμπεριλάβει. Γνωρίζοντας, λοιπόν, τα δύο αυτά σημεία του ορθογωνίου το κέντρο του μπορεί να υπολογιστεί από τις παρακάτω σχέσεις:

$$X_{centroid} = (X_{max} + X_{min}) / 2$$

$$Y_{centroid} = (Y_{max} + Y_{min}) / 2$$

Στην συνέχεια, εφόσον το κέντρο της εικόνας είναι γνωστό, μπορούν εύκολα να

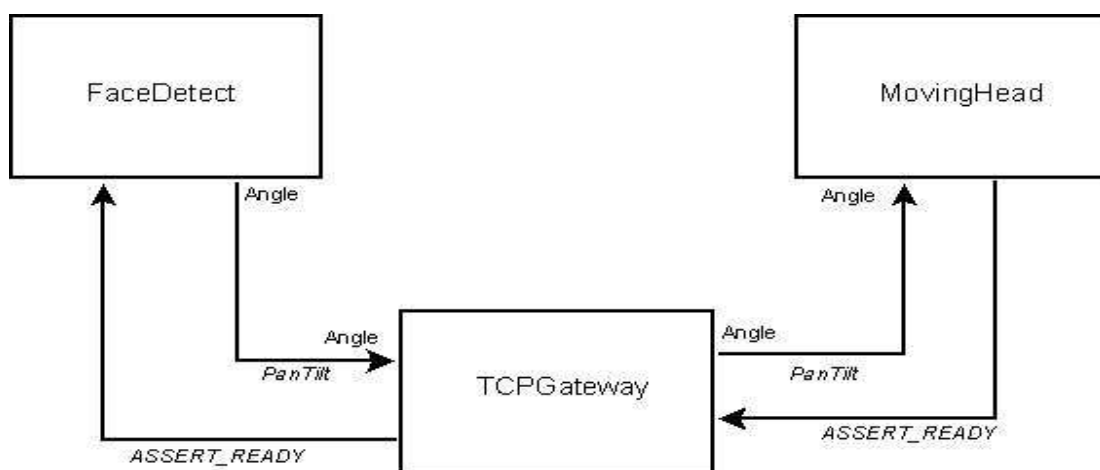
υπολογιστούν οι γωνίες κίνησης του κεφαλιού, στον οριζόντιο και στον κατακόρυφο άξονα, χρησιμοποιώντας τις παρακάτω σχέσεις, δεδομένου ότι οι διαστάσεις της εικόνας είναι 208x160 και το εύρος του οπτικού πεδίου είναι 57,6 μοίρες στον οριζόντιο άξονα και 47,8 μοίρες στον κατακόρυφο:

$$Pan = -1 * 57,6 * (X_{centroid} - 104) / 208$$

$$Tilt = -1 * 47,8 * (Y_{centroid} - 80) / 160$$

Η πύλη που δημιουργήθηκε για την αποστολή του πίνακα που περιέχει τα παραπάνω δεδομένα, ονομάστηκε *Angle* και στο *FaceDetect* που αποστέλλει το μήνυμα και στο *TCPGateway* που το μεταβιβάζει και στο *MovingHead* που είναι ο τελικός αποδέκτης. Τα δεδομένα που μεταφέρονται είναι τύπου *Pantilt* το οποίο έχει ορισθεί ότι είναι ένας πίνακας τριών θέσεων τύπου *double*. Η επικοινωνία των αντικειμένων παρουσιάζεται στο Σχήμα 5.5.

Τέλος, το αντικείμενο *FaceDetect* είναι υπεύθυνο και για την αποστολή οδηγιών και στο αντικείμενο *MovingLegs*, η εκτέλεση των οποίων θα οδηγήσει το ρομπότ σε θέση ικανή ώστε να μπορεί να δει το πρόσωπο enface.



Σχήμα 5.5 : Δι-αντικειμενική επικοινωνία για την αποστολή οδηγιών από το *FaceDetect* στο *MovingHead*

Από τη στιγμή που το αντικείμενο *FaceDetect* έχει ανιχνεύσει πρόσωπο, μπορεί να καταλάβει αν το πρόσωπο αυτό είναι προφίλ ή enface ανάλογα με τον ταξινομητή που χρησιμοποίησε για την επιτυχημένη ανίχνευση. Αν το πρόσωπο είναι enface, δεν αποστέλλονται πληροφορίες στο *MovingLegs*. Σε αυτή την περίπτωση το ρομπότ προσπαθεί να διατηρεί, κουνώντας το κεφάλι του, το πρόσωπο στο κέντρο της εικόνας. Αν το πρόσωπο που ανιχνεύτηκε είναι προφίλ, τότε το αντικείμενο *FaceDetect* στέλνει εντολή στο *MovingLegs* να κινηθεί το ρομπότ σταδιακά με τέτοιο τρόπο ώστε να μπορεί να δει το πρόσωπο enface. Στην περίπτωση που ανιχνεύθηκε δεξί προφίλ το ρομπότ κινείται δεξιά και counterclockwise, ενώ αν ανιχνεύτηκε αριστερό προφίλ, κινείται αριστερά και clockwise όπως φαίνεται στο Σχήμα 5.6.



Σχήμα 5.6: Κίνηση του ρομπότ σε σχέση με το πρόσωπο

Οι εντολές αποστέλλονται μέσω της πύλης *SendString0* και τα δεδομένα που περιέχουν είναι τύπου *char*. Οι εντολές κίνησης αποτελούνται από απλές συμβολοσειρές και είναι πολύ υψηλού επιπέδου. Για παράδειγμα, αν θέλουμε το ρομπότ να κινηθεί 50mm αριστερά απλά στέλνουμε την εντολή *walk 0 50*

(αναλυτικότερα θα δούμε τις εντολές κίνησης στην ανάλυση του αντικείμενου *MovingLegs*).

Επαναλαμβάνω, ότι όλα τα μηνύματα μεταφέρονται από το ρομπότ στον υπολογιστή και αντίστροφα με την διαμεσολάβηση του αντικείμενου *TCPGateway*.

5.7.2 Ανάλυση του αντικείμενου *MovingHead*

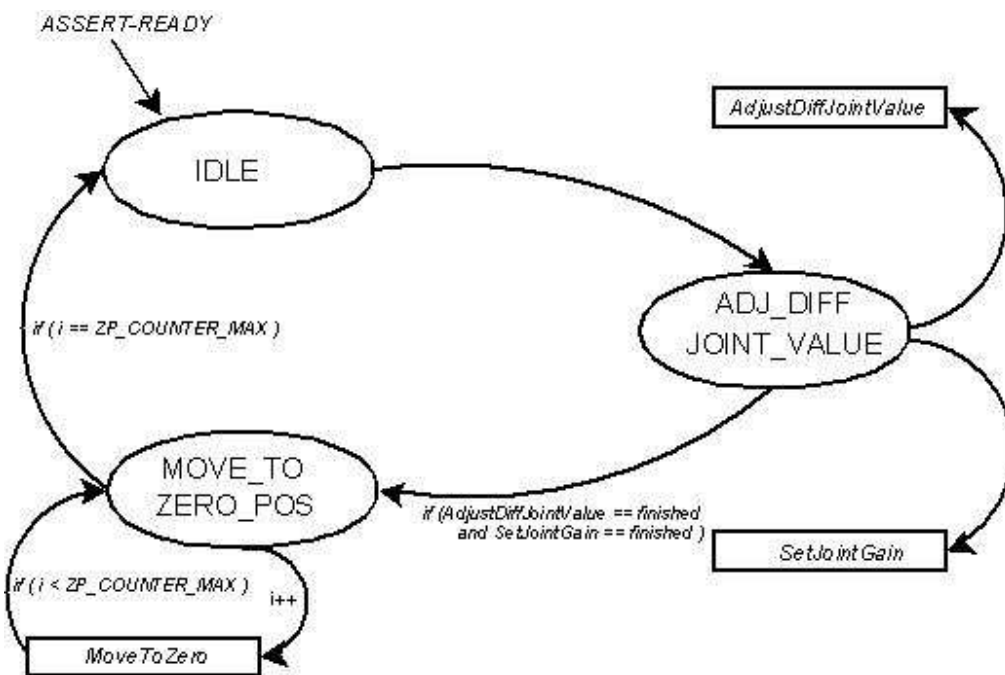
Κατά την αρχικοποίηση το αντικείμενο *MovingHead* μετακινεί το κεφάλι του ρομπότ στην αρχική θέση. Το AIBO αρχικά βρίσκεται στην κατάσταση *IDLE* και παραμένει σ' αυτή μέχρι να λάβει επιβεβαίωση ότι το αντικείμενο *OVirtualRobotComm* είναι έτοιμο να χειριστεί τους μηχανισμούς κίνησης του ρομπότ. Όταν το *MovingHead* λάβει την παραπάνω επιβεβαίωση, τότε μετακινείται στην κατάσταση *ADJ DIFF JOINT VALUE* και καλείται η αντίστοιχη συνάρτηση (*AdjustDiffJointValue*).

Στόχος της *AdjustDiffJointValue* είναι η ρύθμιση της διαφοράς των τιμών ανάμεσα στους αισθητήρες και στους κωδικοποιητές των αρθρώσεων του ρομπότ. Αρχικά, η συνάρτηση διαβάζει τη θέση της κάθε άρθρωσης. Αυτή η θέση μπορεί να διαφέρει από την πραγματική θέση που βρίσκεται η άρθρωση, εξαιτίας μη ρυθμισμένων τιμών των αισθητήρων. Για να λυθεί αυτό το πρόβλημα, στέλνεται εντολή σε κάθε άρθρωση να μετακινηθεί στην θέση που διαβάστηκε αρχικά. Με αυτόν τον τρόπο είμαστε σίγουροι ότι η τιμή της θέσης της άρθρωσης που διαβάζεται από τους αισθητήρες ταυτίζεται με την πραγματική θέση της άρθρωσης.

Όταν η παραπάνω διαδικασία τελειώσει πρέπει να ορισθούν τα *gains* και *shifts* των μηχανισμών κίνησης των αρθρώσεων του ρομπότ, σύμφωνα με τις

προκαθορισμένες τιμές που παρέχονται από την Sony. Η συνάρτηση που καλείται να το κάνει αυτό είναι η *SetJointGain*.

Στην συνέχεια το AIBO μεταφέρεται στην κατάσταση *MOVING TO ZERO POS* και το κεφάλι μεταφέρεται στην θέση μηδέν (κοιτά ευθεία μπροστά). Η μετακίνηση του κεφαλιού στην θέση μηδέν γίνεται σε *ZP_COUNTER_MAX* βήματα για να μπορεί να ελεγχθεί η ταχύτητα της κίνησης. Αφού ολοκληρωθεί η μετακίνηση του κεφαλιού στην θέση μηδέν, το αντικείμενο επιστρέφει στην κατάσταση *IDLE*. Στο Σχήμα 5.7 φαίνονται οι καταστάσεις από τις οποίες πρέπει να περιέλθει και οι λειτουργίες που πρέπει να εκτελέσει το αντικείμενο *MovingHead* πριν την λήψη εντολών από το αντικείμενο *FaceDetect*.



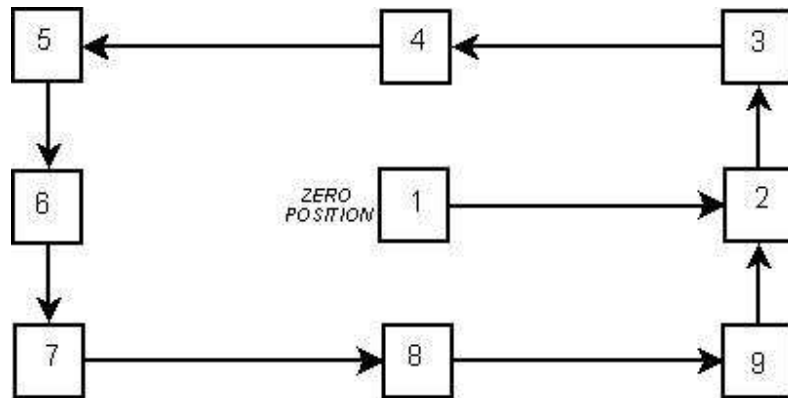
Σχήμα 5.7 : Καταστάσεις και λειτουργίες του αντικειμένου *MovingHead*

Οι επόμενες ενέργειες του αντικειμένου εξαρτώνται από τις εντολές που δέχεται

από το αντικείμενο *FaceDetect*. Αν το αντικείμενο *FaceDetect* ανιχνεύσει ένα πρόσωπο, τότε το αντικείμενο *MovingHead* πρέπει να το τοποθετήσει στο κέντρο της εικόνας και εάν κινείται να το παρακολουθεί. Δεύτερη περίπτωση είναι το αντικείμενο *FaceDetect* να μην ανιχνεύσει πρόσωπο. Σ' αυτή την περίπτωση, το αντικείμενο *MovingHead* πρέπει να κινήσει το κεφάλι του ώστε να ελέγξει τον χώρο γύρω του για πιθανή ύπαρξη προσώπου. Τελευταία περίπτωση είναι το αντικείμενο *FaceDetect* να ανιχνεύσει περισσότερα από ένα πρόσωπα, οπότε το ρομπότ παραμένει στην θέση του και το αντικείμενο *MovingHead* δεν εκτελεί καμία εντολή.

Στην πρώτη περίπτωση που έχει ανιχνευτεί ένα πρόσωπο το αντικείμενο *MovingHead* λαμβάνει πληροφορία για τις συντεταγμένες του κέντρου του προσώπου μέσα στην εικόνα και για τις γωνίες κίνησης του κεφαλιού που πρέπει να πραγματοποιήσει. Με βάση αυτές τις πληροφορίες και τις θέσεις των αρθρώσεων του κεφαλιού το αντικείμενο *MovingHead* μπορεί να κινήσει το κεφάλι με τέτοιο τρόπο ώστε να τοποθετηθεί το πρόσωπο στο κέντρο της εικόνας.

Στην δεύτερη περίπτωση το αντικείμενο *MovingHead* δίνει, αρχικά, εντολή στο κεφάλι να κινηθεί στην θέση μηδέν και στην συνέχεια να κινηθεί κατά μήκος των πλευρών ενός ορθογωνίου ώστε να ελέγξει τον χώρο γύρω του. Η γωνία κάθε κίνησης στον οριζόντιο άξονα είναι 20 μοίρες, ενώ στον κατακόρυφο άξονα είναι 10 μοίρες. Στο Σχήμα 5.8 φαίνεται η ακολουθία των θέσεων του κεφαλιού του ρομπότ.



Σχήμα 5.8 : Ακολουθία θέσεων του κεφαλιού κατά την διαδικασία ελέγχου του χώρου για πιθανή ύπαρξη προσώπου

Αν σε κάποια θέση ανιχνευτεί πρόσωπο τότε η λειτουργία του ρομπότ συνεχίζεται όπως περιγράψαμε στην πρώτη περίπτωση.

5.7.3 Ανάλυση του αντικειμένου MovingLegs

Το αντικείμενο *MovingLegs*, όπως περιγράφεται και από το όνομα του, είναι υπεύθυνο για την κίνηση των ποδιών και για το περπάτημα του AIBO. Για την επίτευξη σωστής κίνησης των ποδιών που θα οδηγήσει και σε ευσταθές περπάτημα είναι απαραίτητη:

- ◆ η γνώση της γεωμετρίας του χώρου που θα εκτελείται η κίνηση,
- ◆ η γνώση της γεωμετρίας του ίδιου του ρομπότ και
- ◆ η σχεδίαση βημάτων για κάθε πόδι λαμβάνοντας υπόψη τις δύο παραπάνω παραμέτρους.

Οι ακριβείς διαστάσεις των μελών του ρομπότ, καθώς και οι επιτρεπτές τιμές και η ταχύτητα κίνησης κάθε άρθρωσης, όπως αυτές αναφέρονται στο εγχειρίδιο χρήσης που παρέχεται από την Sony, είναι αρκετά στοιχεία για την γνώση της γεωμετρίας του ρομπότ. Η γνώση της γεωμετρίας του ρομπότ είναι απαραίτητη, τόσο για τον υπολογισμό του κέντρου βάρους του ρομπότ, για την επίτευξη ευσταθούς κίνησης, όσο και για τον έλεγχο του εύρους των κινήσεων, οι οποίες πρέπει να βρίσκονται μέσα στα επιτρεπτά όρια, όπως αυτά έχουν ορισθεί από την κατασκευάστρια εταιρία, προκειμένου να μην δημιουργηθούν προβλήματα και φθορές στις αρθρώσεις του AIBO.

Για την σχέση του ρομπότ με τον χώρο επισυνάπτεται στο σώμα του ρομπότ ένα σύστημα τριών συντεταγμένων (x,y,z). Η μετατόπιση και οι αποστάσεις στους τρεις άξονες μετριοούνται σε mm. Το σύστημα συντεταγμένων είναι διαφορετικό για κάθε πόδι, δηλαδή η μετατόπιση στους άξονες αντιστοιχίζεται σε κάθε πόδι διαφορετικά. Ο άξονας y είναι κάθετος στο επίπεδο κίνησης του AIBO με φορά προς τα κάτω για κάθε πόδι. Ο άξονας x είναι παράλληλος με την κατεύθυνση του σώματος του ρομπότ (από την ουρά στο κεφάλι). Η διεύθυνση του διαφέρει για τα μπροστά και τα πίσω πόδια. Η διεύθυνση του για τα μπροστά πόδια είναι από την ουρά προς το κεφάλι του ρομπότ, ενώ για τα πίσω πόδια η διεύθυνση είναι αντίθετη. Ο άξονας z είναι κάθετος στο επίπεδο $x-y$ και ουσιαστικά δείχνει την απόσταση των ποδιών από τον κορμό του σώματος του ρομπότ. Η διεύθυνσή του είναι προς τα αριστερά για τα αριστερά πόδια, κοιτώντας το ρομπότ από την ουρά προς το κεφάλι, ενώ για τα δεξιά πόδια η διεύθυνσή του είναι αντίθετη. Σαν σημείο αναφοράς χρησιμοποιείται ο ώμος κάθε ποδιού με τιμές συντεταγμένων (0,0,0). Η απεικόνιση των αξόνων για κάθε πόδι του ρομπότ φαίνεται στο παρακάτω Σχήμα 5.9.



Σχήμα 5.9 : Απεικόνιση της κατεύθυνσης των αξόνων για κάθε πόδι του ρομπότ

Για τον σχεδιασμό κίνησης των ποδιών του ρομπότ κατά μήκος των αξόνων απαιτείται η μετατροπή των συντεταγμένων του συστήματος (x, y, z) σε γωνίες κίνησης (j_1, j_2, j_3) των αρθρώσεων των ποδιών. Οι γωνίες κίνησης ορίζονται ως εξής:

- ◆ j_1 είναι η γωνία που δημιουργείται μεταξύ της προβολής του ανωτέρου μέρους του ποδιού στο x - y επίπεδο και την διεύθυνση του y άξονα.
- ◆ j_2 είναι η γωνία ανάμεσα στο x - y επίπεδο και στο ανώτερο μέρος του ποδιού του ρομπότ.
- ◆ j_3 είναι η γωνία ανάμεσα στο ανώτερο και στο κατώτερο μέρος του ποδιού.

Είναι, δηλαδή η γωνία που σχηματίζεται από το γόνατο του ποδιού.

Όπως έχω προαναφέρει, για την μετατροπή των συντεταγμένων σε γωνίες κίνησης είναι απαραίτητη η γνώση των φυσικών διαστάσεων των αρθρώσεων του ρομπότ, οι οποίες απαιτείται να είναι στην ίδια τάξη μεγέθους με τις μετατοπίσεις κατά μήκος των αξόνων (mm).

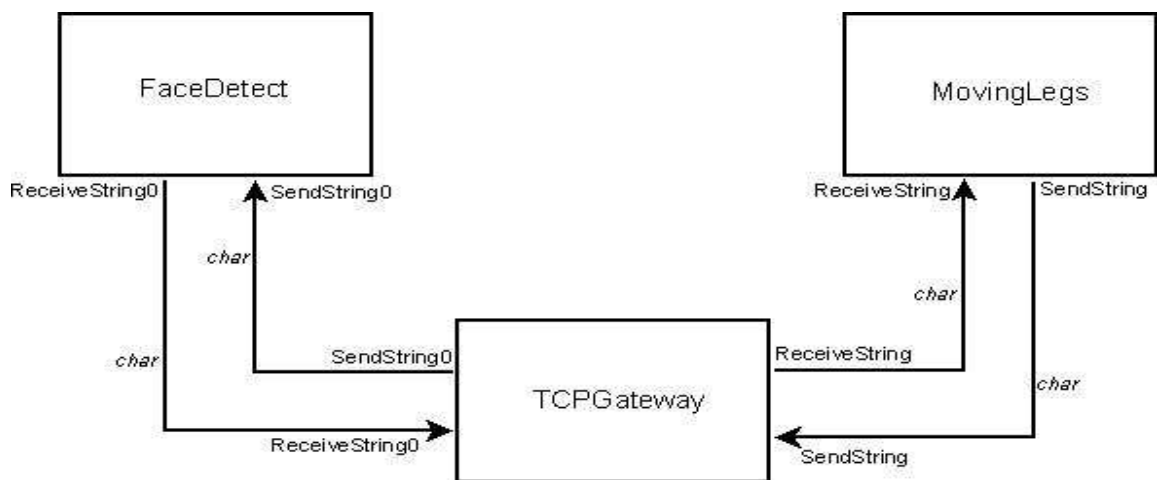
Το αντικείμενο *MovingLegs* είναι σχεδιασμένο να επικοινωνεί ασύρματα με τερματική κονσόλα, μέσω της οποίας ο χρήστης θα μπορεί να στέλνει εντολές κίνησης στο ρομπότ. Για την διευκόλυνση του χρήστη, η επικοινωνία επιτυγχάνεται με την αποστολή συγκεκριμένων εντολών, πολύ υψηλού επιπέδου (walk 0 0 50), για εκτέλεση συγκεκριμένων κινήσεων. Είναι προφανές, λοιπόν, ότι τα μηνύματα που ανταλλάσσονται μεταξύ χρήστη και αντικειμένου, φέρουν δεδομένα τύπου *string*.

Για να μπορέσει να χρησιμοποιηθεί το αντικείμενο *MovingLegs* στην δική μου εργασία, έπρεπε να τροποποιηθεί με τέτοιο τρόπο, ώστε την θέση του χρήστη να λάβει ένα άλλο αντικείμενο του προγράμματος, προκειμένου το ρομπότ να μπορεί να λειτουργεί χωρίς ανθρώπινη παρέμβαση.

Το αντικείμενο *MovingLegs* επικοινωνεί μέσω ασυρμάτου δικτύου με το αντικείμενο *TinyTelnet* μέσω του οποίου στέλνει εντολές κίνησης ο χρήστης στο ρομπότ. Αν, λοιπόν, η θέση του αντικειμένου *TinyTelnet* καθώς και η συμμετοχή του χρήστη στην διαδικασία εκτέλεσης του προγράμματος, αντικατασταθούν από το αντικείμενο *FaceDetect* τότε το αντικείμενο *MovingLegs* μπορεί να χρησιμοποιηθεί στην υλοποίηση του προγράμματος ικανοποιώντας τις απαιτήσεις της εργασίας μου.

Το αντικείμενο *FaceDetect* είναι υπεύθυνο για τη διαδικασία ανίχνευσης

προσώπου, και συνεπώς είναι σε θέση να γνωρίζει αν το πρόσωπο που ανιχνεύθηκε είναι προφίλ ή enface. Αν το πρόσωπο που ανιχνεύτηκε είναι προφίλ, τότε τα αντικείμενο *FaceDetect* στέλνει εντολές κίνησης προς το *MovingLegs* με την μορφή συμβολοσειρών, αφού τέτοιου τύπου είναι τα δεδομένα των μηνυμάτων που μεταφέρονται προς το *MovingLegs* και περιέχουν τις εντολές. Αν το πρόσωπο που ανιχνεύτηκε είναι enface τότε το αντικείμενο *FaceDetect* στέλνει εντολές διακοπής της κίνησης. Κατά την κίνηση του ρομπότ το πρόσωπο που παρακολουθείται ελέγχεται αν συνεχίζει να είναι προφίλ. Αν παραμένει σε θέση προφίλ η κίνηση συνεχίζεται, αλλιώς το ρομπότ σταματάει και απλά συνεχίζει να παρακολουθεί το πρόσωπο. Η επικοινωνία του αντικειμένου *MovingLegs* με το *FaceDetect* φαίνεται στο Σχήμα 5.10.



Σχήμα 5.10 : Επικοινωνία μεταξύ *FaceDetect* – *MovingLegs*

5.8 Σύνοψη του προγράμματος

Παραπάνω έγινε αναλυτική περιγραφή των αντικειμένων που χρησιμοποιεί το

πρόγραμμα και της δι-αντικειμενικής επικοινωνίας, προκειμένου να εκτελεστεί σωστά μια διαδικασία του προγράμματος. Στο τέλος του κεφαλαίου παρατίθεται στο Σχήμα 5.11 η επικοινωνία όλων των αντικειμένων του προγράμματος.

Για την καλύτερη κατανόηση της διαδικασίας ανταλλαγής μηνυμάτων μεταξύ των αντικειμένων του προγράμματος, παρατίθενται τα αρχεία τα οποία χρησιμοποιήθηκαν για την επικοινωνία τους.

Robot connect file

```
OVirtualRobotComm.FbkImageSensor.OFbkImageVectorData.S
TCPGateway.Image.OFbkImageVectorData.O
TCPGateway.Angle.Pantilt.S MovingHead.Angle.Pantilt.O
MovingHead.Move.OCommandVectorData.S
OVirtualRobotComm.Effector.OCommandVectorData.O
WalkingLegs7.Move.OCommandVectorData.S
OVirtualRobotComm.Effector.OCommandVectorData.O
TCPGateway.SendString0.char.S WalkingLegs7.ReceiveString.char.O
WalkingLegs7.SendString.char.S TCPGateway.ReceiveString0.char.O
```

Host connect file

```
TCPGateway.Image.OFbkImageVectorData.S
FaceDetect.Image.OFbkImageVectorData.O
FaceDetect.Angle.Pantilt.S TCPGateway.Angle.Pantilt.O
TCPGateway.SendString0.char.S FaceDetect.SendString0.char.O
FaceDetect.ReceiveString0.char.S TCPGateway.ReceiveString0.char.O
```

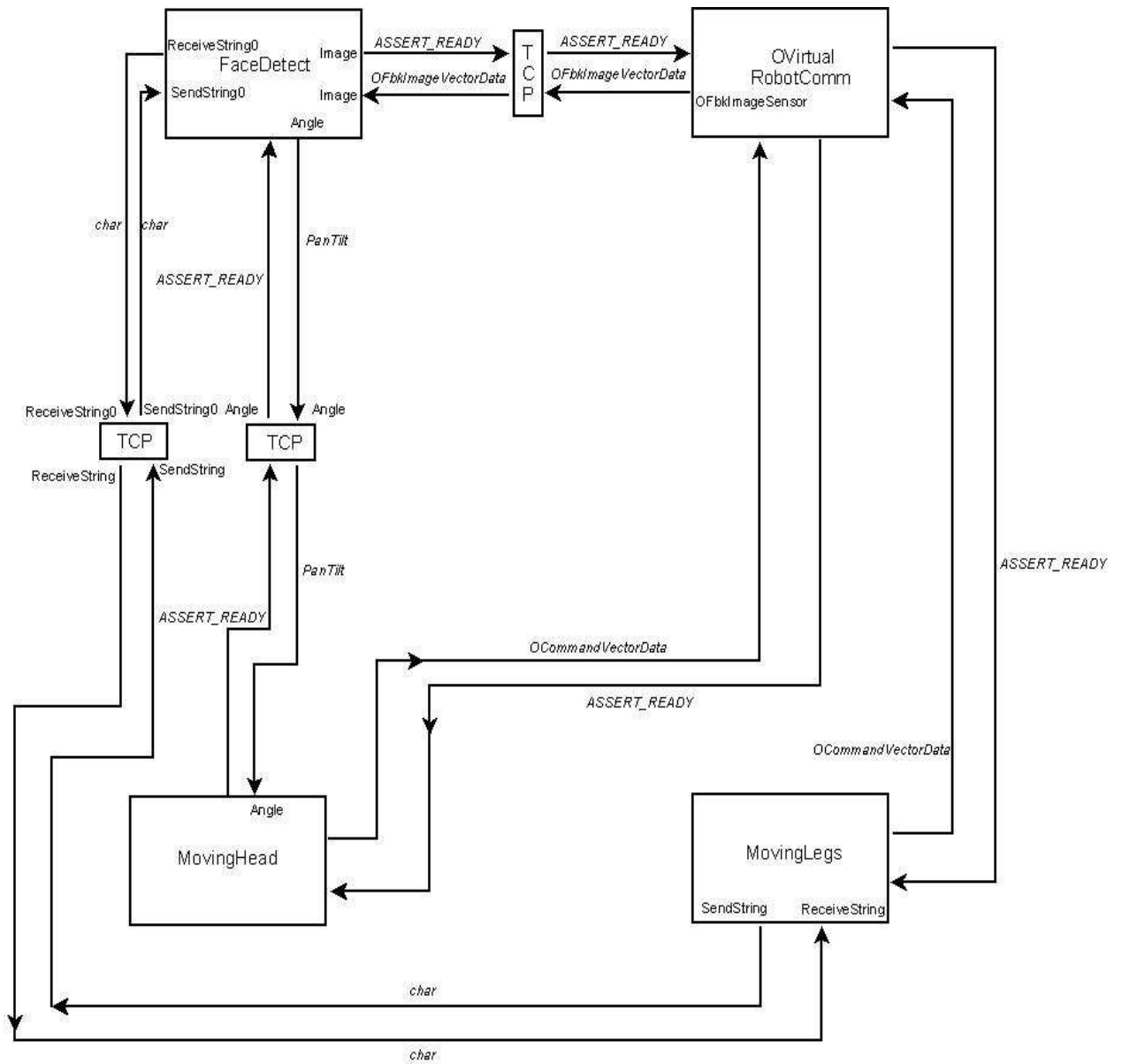
Το πρώτο αρχείο χρησιμοποιείται για την επικοινωνία των αντικειμένων που εκτελούνται στο ρομπότ με το αντικείμενο *TCPGateway*, το οποίο έχω αναφέρει είναι το αντικείμενο που μεταφέρει τα μηνύματα από ρομπότ στον υπολογιστή και αντίστροφα. Το δεύτερο αρχείο εξασφαλίζει την επικοινωνία του αντικειμένου

FaceDetect, που εκτελείται στον υπολογιστή, με το αντικείμενο *TCPGateway*.

Κάθε γραμμή των παραπάνω αρχείων έχει την εξής δομή:

Αντικείμενο.Πύλη.ΤύποςΔεδομένων.S Αντικείμενο.Πύλη.ΤύποςΔεδομένων.O

Το *S* και το *O* αντιστοιχούν στο *Subject* και στο *Observer*, δηλαδή στον αποστολέα και στον παραλήπτη του μηνύματος. Για παράδειγμα η πρώτη γραμμή του αρχείου *Host connect*, δηλώνει ότι το αντικείμενο *TCPGateway* μέσω της πύλης *Image* θα στείλει δεδομένα τύπου *OFbkImageVectorData* και το αντικείμενο *FaceDetect* μέσω της πύλης *Image* θα δεχτεί αυτά τα δεδομένα.



Σχήμα 5.11 : Επικοινωνία όλων των αντικειμένων του προγράμματος

Κεφάλαιο 6

Αποτελέσματα και μελλοντικές προεκτάσεις

Το παρόν κεφάλαιο περιέχει μια σύντομη αναφορά των όσων έχουν υλοποιηθεί στα πλαίσια της διπλωματικής μου εργασίας. Επίσης περιλαμβάνει αναφορά των αποτελεσμάτων που εξήχθησαν και κριτική τους ώστε να φανεί ο βαθμός επίτευξης των αρχικών στόχων. Τέλος, αναφέρονται βελτιώσεις και πιθανές μελλοντικές προεκτάσεις της συγκεκριμένης εφαρμογής.

6.1 Στόχος και υλοποίηση της εφαρμογής

Στόχος της παρούσας εργασίας, όπως έχει αναφερθεί, είναι η ανίχνευση και η παρακολούθηση προσώπου, από το τετράποδο ρομπότ AIBO, και η μετακίνηση του ρομπότ, εάν αυτή απαιτείται, ώστε να βρεθεί σε θέση που θα μπορεί να δει το πρόσωπο enface. Η υλοποίηση της εφαρμογής έγινε στο προγραμματιστικό περιβάλλον OPENR-SDK. Η διαδικασία ανίχνευσης προσώπου έγινε με την βοήθεια της βιβλιοθήκης OpenCV. Το μέρος της εφαρμογής, το οποίο είναι υπεύθυνο για την κίνηση εκτελείται πάνω στο ρομπότ, ενώ το μέρος, το οποίο είναι υπεύθυνο για την διαδικασία ανίχνευσης προσώπου εκτελείται σε υπολογιστή, ο οποίος επικοινωνεί ασύρματα με το ρομπότ.

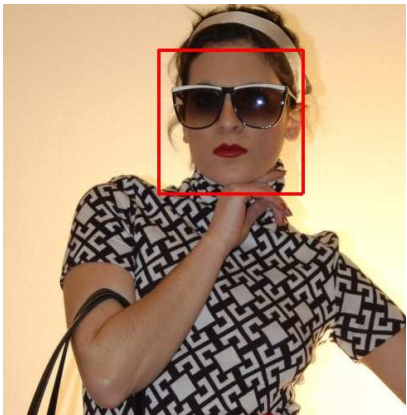
6.2 Αποτελέσματα

Τα αποτελέσματα της συγκεκριμένης εφαρμογής εξαρτώνται σε μεγάλο βαθμό από το ποσοστό επιτυχίας της διαδικασίας ανίχνευσης. Η διαδικασία ανίχνευσης δοκιμάστηκε με ανδρικά και γυναικεία πρόσωπα ενηλίκων σε φωτισμό δωματίου. Στην περίπτωση που το πρόσωπο βρίσκεται σε θέση enface η διαδικασία ανίχνευσης είναι επιτυχημένη εφόσον ολόκληρο το πρόσωπο βρίσκεται μες στην εικόνα. Αυτός ο περιορισμός ισχύει και για την περίπτωση που το πρόσωπο είναι

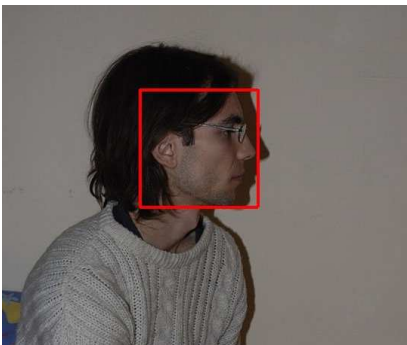
προφίλ. Όπως διαπιστώθηκε, η διαδικασία ανίχνευσης, στην περίπτωση που το πρόσωπο είναι προφίλ, επιτυγχάνει μόνο για το αριστερό προφίλ. Αυτό δεν αποτελεί πρόβλημα για την επίτευξη του αρχικού στόχου, που απαιτεί ανίχνευση και των δύο προφίλ, γιατί και το δεξί προφίλ μπορεί να ανιχνευτεί με κατοπτρισμό της εικόνας ως προς τον κατακόρυφο άξονα. Μερικά αποτελέσματα της διαδικασίας ανίχνευσης παρουσιάζονται παρακάτω.



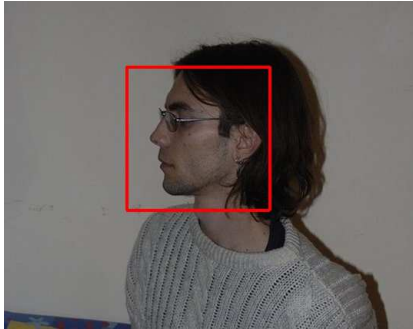
Ανδρικό πρόσωπο enface



Γυναικείο πρόσωπο enface



Αριστερό προφίλ



Δεξί προφίλ

Η διαδικασία ανίχνευσης προσώπου παρουσιάζει κάποιες ιδιαιτερότητες με τις εικόνες που λαμβάνει η κάμερα του ρομπότ, λόγω χαμηλής ανάλυσης. Το πρόσωπο προς ανίχνευση πρέπει να έχει διαστάσεις τουλάχιστον 20x20 pixels, πράγμα που σημαίνει ότι το πρόσωπο πρέπει να βρίσκεται σε απόσταση όχι μεγαλύτερη από 1,5m από την κάμερα του AIBO. Όπως έχει αναφερθεί στα τεχνικά χαρακτηριστικά του AIBO, η κάμερα έχει την δυνατότητα λήψης 30 εικόνων το δευτερόλεπτο. Λόγω των καθυστερήσεων της διαδικασίας ανίχνευσης και της ανταλλαγής μηνυμάτων μεταξύ ρομπότ και υπολογιστή γίνονται 15 ανιχνεύσεις το δευτερόλεπτο. Αυτό επηρεάζει την διαδικασία παρακολούθησης προσώπου. Επειδή γίνονται δώδεκα και όχι εικοσιπέντε ανιχνεύσεις το δευτερόλεπτο ο άνθρωπος πρέπει να κινείται με μικρή ταχύτητα, έτσι ώστε να μην βγαίνει κάποιο μέρος του προσώπου του έξω από το οπτικό πεδίο του ρομπότ.

Η ταχύτητα κίνησης του ρομπότ επηρεάζει την διαδικασία ανίχνευσης και παρακολούθησης, λόγω του ότι απότομες κινήσεις τείνουν να βγάλουν το πρόσωπο έξω από το οπτικό πεδίο του ρομπότ. Η μεγαλύτερη ταχύτητα για την οποία η διαδικασία ανίχνευσης και παρακολούθησης ήταν επιτυχημένες είναι η κίνηση με βήμα 1cm μπροστά και πίσω και 2cm στα πλάγια. Αναφέρεται ότι το συγκεκριμένο μέγεθος του πλάγιου βήματος είναι και το μέγιστο που μπορεί να εκτελέσει το ρομπότ.

6.3 Μελλοντικές προεκτάσεις και βελτιώσεις

Κάποιες πιθανές μελλοντικές προεκτάσεις και βελτιώσεις της συγκεκριμένης εφαρμογής μπορεί να είναι οι ακόλουθες:

- *Ταυτοποίηση προσώπου*

Η θέση προσώπου enface είναι αυτή που παρέχει την περισσότερη πληροφορία για ένα πρόσωπο. Από την στιγμή που το ρομπότ είναι σε θέση να ανιχνεύει πρόσωπο και να κινείται ώστε να βρεθεί σε θέση που θα μπορεί να το δει enface, μπορεί να εκπαιδευτεί με πρόσωπα συγκεκριμένων ατόμων και να προχωρά σε αναγνώριση και ταυτοποίηση προσώπου.

- *Ανάπτυξη πολυπλοκότερης συμπεριφοράς του ρομπότ*

Παράλληλα με την ανίχνευση και αναγνώριση προσώπου θα ήταν ενδιαφέρον να αναπτυχθεί και πολυπλοκότερη συμπεριφορά για το AIBO, ώστε να αλληλεπιδρά αποδοτικότερα με το περιβάλλον του. Εκφράσεις με τα LED's του κεφαλιού, οι ήχοι και ομιλία από το μεγάφωνο και η εκτέλεση πολυπλοκότερων κινήσεων είναι πιθανές επιλογές για την επίτευξη αυτού του στόχου.

- *Ανίχνευση μεμονωμένων χαρακτηριστικών προσώπου*

Για να είναι επιτυχής η ανίχνευση προσώπων όταν κάποιο μέρος τους βρίσκεται έξω από το οπτικό πεδίο του AIBO, μπορούν να δημιουργηθούν ταξινομητές για μεμονωμένα χαρακτηριστικά ανθρώπινων προσώπων. Τέτοια χαρακτηριστικά μπορούν να αποτελέσουν τα μάτια, το στόμα και η μύτη.

- *Πλήρης συμβατότητα της OpenCV με το AIBO*

Στα πλαίσια της διπλωματικής μου εργασίας ένα μέρος της OpenCV έχει γίνει συμβατό με το AIBO, και το ρομπότ είναι σε θέση να εκτελεί απλές λειτουργίες επεξεργασίας εικόνας πλήρως αυτόνομα. Λειτουργίες, όμως, που

έχουν μεγάλες απαιτήσεις μνήμης και μεγάλο υπολογιστικό κόστος δεν μπορούν να εκτελεστούν στο ρομπότ. Βελτιστοποίηση της βιβλιοθήκης για την αρχιτεκτονική του επεξεργαστή του AIBO, ίσως δώσει λύση σε αυτό το πρόβλημα.

6.4 Συμπεράσματα

Με την παραπάνω εργασία μου δόθηκε η ευκαιρία να δω τον τρόπο με τον οποίο μπορούν να αναπτυχθούν περιβάλλοντα αλληλεπίδρασης ανάμεσα σε άνθρωπο και ρομπότ, καθώς επίσης, και το πώς μπορούν διαφορετικές εφαρμογές να συνδυαστούν για την επίτευξη ενός συγκεκριμένου στόχου. Ακόμη, μου δόθηκε η ευκαιρία να ασχοληθώ με αντικείμενα, με τα οποία δεν είχα ασχοληθεί πιο πριν, όπως event-based προγραμματισμός και ρομποτική.

Το πρόβλημα της αλληλεπίδρασης ανθρώπου – υπολογιστή έχει κινήσει το ενδιαφέρον των επιστημόνων και των ερευνητών, με αποτέλεσμα συνεχώς να αναπτύσσονται εφαρμογές που στόχο έχουν την φυσική αλληλεπίδραση ανάμεσα σε άνθρωπο και μηχανή. Η λύση του προβλήματος παραμένει, όμως, πολύ δύσκολη, αν στόχος είναι η πλήρως φυσική αλληλεπίδραση ανάμεσα σε άνθρωπο και υπολογιστή, όπως αυτή ορίζεται από τα ανθρώπινα δεδομένα.

Τα σύγχρονα ανθρωποειδή ρομπότ απέχουν από τα ανθρωποειδή ρομπότ που έπλασε η φαντασία του Philip K. Dick, όπως επίσης, και η εποχή μας μοιάζει πολύ διαφορετική από την εποχή που ανθρωποειδή ρομπότ θα κινούνται στους δρόμους και θα βρίσκονται σε κάθε σπίτι, για να εξυπηρετούν τις ανάγκες του ανθρώπου. Η ιστορία της επιστήμης, όμως, έχει δείξει ότι ο δρόμος για κάθε επιστημονικό επίτευγμα ξεκινά όταν το πλάσει η φαντασία κάποιου.

Αναφορές

Βιβλιογραφία

- [1] OPEN-R SDK Documentation – Model Information for ERS-7, Sony Corporation, 2002
- [2] <http://www-2.cs.cmu.edu/~tekkotsu/> - Official Tekkotsu Website
- [3] <http://www.urbiforge.com/eng/> - URBI official web page
- [4] Jean-Christophe Baillie, “URBI Language Specification”, 2005
- [5] <http://www.mmi.tudelft.nl/~siska/AIBO/> - MMI AIBO Team
- [6] “Fusing speech and face recognition on the AIBO ERS-7”, Delft University of Technology, Project Report, M. Thouzery, L. Rothkrantz, Z. Yang, 2005
- [7] “Rapid Object Detection using a Boosted Cascade of Simple Features” Paul Viola and Michael Jones, Conference on Computer Vision and Pattern Recognition (CVPR), 2001
- [8] <http://www.sourceforge.net/projects/opencvlibrary/>
- [9] http://www.ensta.fr/~baillie/openr_tutorial.html
- [10] OPEN-R SDK Documentation – Programmer’s Guide, Sony Corporation, 2002
- [11] <http://www.users.bigpond.com/hengst/OmniLoco.zip>

Πηγές σχημάτων και εικόνων

- Σχήμα 2.1 (<http://openr.aibo.com>)
- Σχήμα 2.2 , 2.3 , 2.4 (http://www.ensta.fr/~baillie/openr_tutorial.html)
- Σχήμα 2.5 (<http://openr.aibo.com>)

- Σχήμα 3.1 , 3.2 , 3.3 , .3.4 “Rapid Object Detection using a Boosted Cascade of Simple Features” Paul Viola and Michael Jones, 2001
- Σχήμα 5.1 (http://www.ensta.fr/~baillie/openr_tutorial.html)
- Σχήμα 5.9 (<http://openr.aibo.com>)

Χρήσιμοι δικτυακοί τόποι

http://www.ensta.fr/~baillie/openr_tutorial.html

<http://www.ouroboros.org/~rt71592>

<http://www.aibohack.com>

<http://www.aibo-life.org/>