Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πτυχιακή εργασία

Τίτλος: Έλεγχος και πλοήγηση ρομποτικού οχήματος με χρήση του action pro live

Νικόλαος-Εμμανουήλ Σκορδίλης (ΑΜ:3177)

Επιβλέπων καθηγητής : Βιδάκης Νικόλαος

Επιτροπή Αξιολόγησης : Βιδάκης Νικόλαος, Κοσμόπουλος Δημήτρης, Μηλολιδάκης Ιωάννης

Ημερομηνία παρουσίασης:

Περιεχόμενα

Περιεχόμενα

Σύνοψη

Την τελευταία δεκαετία η ρομποτική σημείωσε σημαντική πρόοδο. Παρόλα αυτά τα ρομπότ αυτά αφορούν αποκλειστικά την βιομηχανία και λειτουργούν σε καλά ελεγχόμενα περιβάλλοντα. Επίσης το προσωπικό που έρχεται σε επαφή με αυτά πρέπει να έχει μάλλον ειδική εκπαίδευση. Η είσοδος των οικιακών ρομπότ που θα αποτελούν μέρος της καθημερινότητας μας φαίνεται να απέχει ακόμα από το να γίνει πραγματικότητα.

Προϋπόθεση για την λειτουργιά τέτοιων φιλικών προς τον χρήστη ρομπότ είναι η δυνατότητα του εκάστοτε ρομπότ να αντιλαμβάνεται την ύπαρξη των ανθρώπων στο χώρο καθώς επίσης και επιπλέων πληροφοριών σχετικές με αυτούς. Θα πρέπει να είναι δυνατή η ταυτοποίηση ώστε το ρομπότ να γνωρίζει ποιος είναι καθένας, την θέση τους στο χώρο και την στάση του σώματος τους. Επίσης το ρομπότ θα πρέπει να μπορεί να ακολουθεί κάποιον όταν πρέπει. Αυτά με την σειρά τους έχουν ανάγκη ένα σύστημα ρομποτικής όρασης. Παλιότερα περίπλοκοι αλγόριθμοι θα ήταν απαραίτητο να υλοποιηθούν προκειμένου να παρέχουν τις απαραίτητες πληροφορίες ρομποτικής όρασης. Τώρα πλέον όμως με την εμπορική εμφάνιση συσκευών όπως το Kinect και asus action pro και την ελεύθερη πρόσβαση σε βιβλιοθήκες όπως το open-ni ,ειδικά σχεδιασμένες για τις παραπάνω συσκευές ,η απόκτηση των περισσότερων από τις πληροφορίες που ανέφερα αρχικά είναι εύκολη υπόθεση.

Σαν παράδειγμα εφαρμογής που περιέχει όλα τα παραπάνω επιλέχθηκε η κατασκευή ενός ρομποτικού οχήματος που θα αναγνωρίζει τον χρήστη του και μετά θα τον ακολουθεί. Βέβαια κάποιες από τις απαιτήσεις που έχει ένα τέτοιο ρομποτικό όχημα δεν μπορούν να καλυφτούν από το open-ni μόνο, είτε γιατί δεν προβλέπονται ούτως ή άλλως από το open-ni όπως η ταυτοποίηση του χρήστη ,είτε γιατί προκύπτουν κάποια ιδιαίτερα προβλήματα που προκύπτουν από τις παραδοχές που έχουν γίνει κατά την κατασκευή του open-ni. Για παράδειγμα όταν γίνεται προσπάθεια εντοπισμού ενός ανθρώπου στο χώρο υπάρχει η παραδοχή από το open-ni ότι μόνο οι άνθρωποι κινούνται. Αυτό συμβαίνει γιατί κανονικά το action pro είναι ακίνητο και έτσι η παραδοχή είναι λογική. Όμως όταν το όχημα κινείται αυτό δεν ισχύει. Για αυτούς τους λόγους επιπλέον μηχανισμοί βασισμένοι σε νευρωνικά δίκτυα και άλλες τεχνικές αναπτυχτήκαν και έτσι είναι δυνατή και η ταυτοποίηση των χρηστών και η απόρριψη τον εσφαλμένων εντοπισμών. Επιπλέον ένας μηχανισμός αποφυγής εμποδίων και σχεδιασμού της κίνησης υλοποιήθηκε πρέπει να φτιαχτεί πράγμα που διευκολύνεται από τον χάρτη βάθους που προσφέρει το asus action pro. Βάση των παραπάνω, κατασκευάστηκε ένα ρομποτικό όχημα που εντοπίζει, ταυτοποιεί, και αν ο χρήστης είναι κατάλληλος στην συνεχεία τον ακολουθεί χωρίς πρόβλημα.

Abstract

During the last decade, robot technology has advanced significantly. However these robots are used exclusively by industry in fully controlled environments. Additionally, the personnel using them should have adequate training. Home use robots as part of our daily routines still seems a distant reality.

Prerequisite for such user friendly robots is the ability to perceive humans around them and additional information about them. Identification must be available so that the robot can identify, its operator, location and body pose. Additionally the robot must be able to follow its operator if it is desired so. All these have a robotic vision system as prerequisite. In the past, more complex and sophisticated algorithms were necessary for the implementation of a robotic vision system. But nowadays the appearance of relatively low-cost commercial devices such as Kinect and asus action pro, and with free access on frameworks such as open-ni, designed specifically for these devices, gathering most of the aforementioned information becomes easy.

As a use case, a mobile robot have been constructed which is able to recognize and follow its user. Open-ni alone cannot cover some of the requirements of such a robot, either because the open-ni was not designed to provide user recognition, or because some special problems arise. For example when open-ni tries to find a person in the area, the assumption is made that only humans move. This happens because normally the action pro is fixed in place and in that case this assumption is valid. But when the robot moves it is not. For these reasons, additional mechanisms have been developed based on neural networks and other techniques so the user recognition and the rejection of fault detections is possible. Additionally, a mechanism of obstacle avoidance and path planning have been be designed, something that becomes easier with the use of action pros' depth map.

Κεφάλαιο 1: Εισαγωγή

1.1 Εισαγωγή: γενικά για την πτυχιακή εργασία

Κατά την τελευταία δεκαετία η ρομποτική τεχνολογία σημείωσε σημαντική πρόοδο. Παρόλα αυτά τα συμβατικά ρομπότ χρησιμοποιούνται μόνο για βιομηχανική χρήση σε κάποια καλά ελεγχόμενα περιβάλλοντα ,όπως βιομηχανίες και εργαστήρια και τα ρομπότ καθημερινής χρήσης απέχουν αρκετά από την καθημερινότητα. Τα μελλοντικά ρομπότ θα πρέπει να είναι φιλικά προς τον χρήστη και να μπορούν να είναι αποτελεσματικοί προσωπικοί βοηθοί. Ένα από τα πιο σημαντικά θέματα πάνω στην ανάπτυξη φιλικών προς τον χρήστη ρομπότ είναι η κατανόηση του πως θα μπορούν να συνεργάζονται επιτυχώς με τους ανθρώπους. Μπορούμε να παρατηρήσουμε ότι σχεδόν σε όλες τις εφαρμογές που υπάρχει συνεργασία ενός ρομπότ με ανθρώπους υπάρχει η ανάγκη του εντοπισμού και ακολούθησης του χρήστη. Για παράδειγμα ένας ρομποτικός βοηθός έχει ανάγκη να είναι κοντά και να παρακολουθεί αυτόν που τον χρησιμοποιεί προκειμένου να λάβει τις επόμενες εντολές. Σε κάποιες άλλες εφαρμογές η δυνατότητα να ακολουθάει το ρομπότ τον χρήστη του είναι από μόνη της αρκετή όπως για παράδειγμα στην μεταφορά φορτίων. Τέλος τέτοια ρομπότ μπορούν να χρησιμοποιηθούν σαν παιχνίδια ή ρομποτικά κατοικίδια. Με βάση τα παραπάνω αποφάσισα να φτιάξω ένα ρομπότ που να μπορεί να ταχτοποιήσει τον χρήστη του και στην συνέχεια να τον ακολουθάει αποφεύγοντας τυχών εμπόδια και όντας σε θέση να ξαναβρεί τον χρήστη του μετά από προσωρινή απώλεια επαφής με αυτόν.

1.2 Παρόμοιες εργασίες

Αρκετές ερευνητικές ομάδες έχουν παρουσιάσει δουλειά γύρω από το θέμα αυτό. Για παράδειγμα στη δουλεία των S.O. Lee et al[2] παρουσιάζεται ένα κινούμενο ρομπότ ,που πάντοτε βρίσκεται απέναντι σε κάποιον και λειτουργεί σαν ρομποτικός βοηθός. Το ρομπότ που παρουσιάζεται στο [3] είναι ένα ρομπότ σχεδιασμένο να ακολουθεί. Αυτό το ρομπότ οδηγεί ένα καροτσάκι μέσα σε ένα νοσοκομείο ή ένα σταθμό και είναι ικανό να εκτιμά την ταχύτητα και την θέση των ανθρώπων στο χώρο όπως και να αποφεύγει εμπόδια. Στο [4] ,τετράποδα ρομπότ με σκοπό να ακολουθούν ανθρώπους παρουσιάζονται. Σε όλες τις προαναφερθείσες εργασίες η ανάλυση γίνεται πάνω στο πώς να ακολουθάς κάποιον και όχι πώς να τον εντοπίζεις. Οι αναλύσεις τους βασίζονται στην υπόθεση ότι ο εντοπισμός έχει ήδη γίνει.

Γενικά διάφορες προσεγγίσεις έχουν γίνει πάνω στο πως είναι δυνατός ο εντοπισμός των παρευρισκόμενων και της σχετικής τους θέσης με το όχημα, όπως υπερηχητικοί αισθητήρες, συσκευές αναγνώρισης φωνής,laser scanner,υπέρυθρες κάμερες, κάμερες ccd,και άλλα[5]. Ο Gockley et al [6] περιγράφουν μια βασισμένη σε laser ανίχνευση ατόμων και δυο για να ακολουθούνται τα άτομα, μια βασισμένη στην κατεύθυνση και μια σε μονοπάτι.

1.3 Οι λόγοι που οδήγησαν στην χρήση του action pro

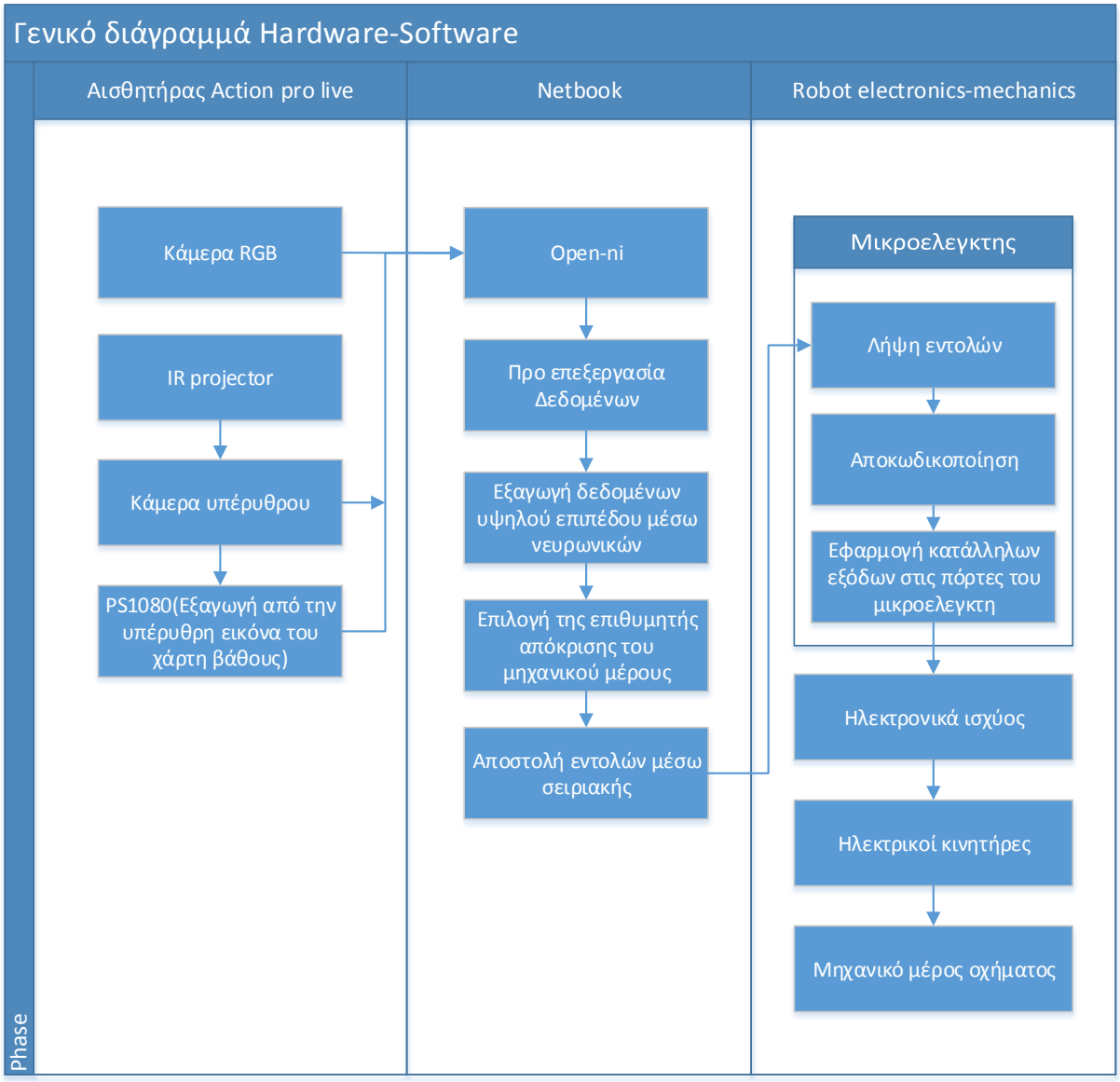
Παίρνοντας στα υπόψη το κόστος που είχαν διάφορες εναλλακτικές όπως αυτές που χρησιμοποιήθηκαν σε παρόμοιες εργασίες μόνο χρήση μιας κάμερας rgb (η δυο για στερεοσκοπική όραση) ήταν κατάλληλη για ένα φτηνό σύστημα ρομποτικής όρασης. Όμως σε αυτήν την περίπτωση ο φωτισμός της περιοχής, οι σκιές, η ασάφεια των εικόνων(πως μπορεί ο υπολογιστής να ξεχωρίσει μια φωτογραφία από ένα αληθινό πρόσωπο?)δημιουργούν επιπλέον προβλήματα στον εντοπισμό και την αναγνώριση ανθρώπων και αντικειμένων. Επίσης η απόσταση είναι δύσκολο να υπολογιστεί με τις μεθόδους της επεξεργασίας εικόνας, γνώση της οποίας είναι απαραίτητη για κίνηση του οχήματος και για τον υπολογισμό της κλίμακας, που με την σειρά της είναι σημαντική για την αναγνώριση των αντικειμένων. Όλα αυτά τα προβλήματα μπορούν να λυθούν αν έχουμε στην διάθεση μας ένα χάρτη βάθους. Τα Laser Scanner που μπορούν να παρέχουν τέτοιου είδους χάρτες είναι ακριβά αλλά με τις πρόσφατα εμφανισμένες στην αγορά κάμερες βάθους όπως το MS Kinect[7] ή το Asus Actıon pro live[8],υπάρχει πλέων ένας φτηνός και αποδοτικός τρόπος να έχουμε χάρτες βάθους. Με βάση αυτές τις συσκευές έχουν σχεδιαστεί εφαρμογές και middleware που μπορούν να μας δώσουν υψηλού επιπέδου πληροφορίες σαν την θέση του χρήστη στο οπτικό πεδίο της κάμερας. Οι παραπάνω δυνατότητες αν και έχουν σχεδιαστεί με βάση την υπόθεση ότι μόνον οι άνθρωποι κινούνται στο οπτικό πεδίο της κάμερας μπορούν να διευκολύνουν σημαντικά δημιουργία του λογισμικού του οχήματος. Έτσι με δεδομένο ότι το Asus Actıon pro live σε αντίθεση με το MS Kinect δεν απαιτεί επιπλέον τροφοδοσία το επέλεξα σαν τον αισθητήρα που θα βασιστεί η ρομποτική όραση του οχήματος. Θετικά επίσης επέδρασε στην επιλογή του action pro το γεγονός ότι έχει μικρότερο βάρος αλλά και μικρότερο μέγεθος.

1.3Γενική περιγραφή του τρόπου λειτουργιάς του οχήματος

Η λειτουργιά του οχήματος σε γενικές γραμμές μπορεί να περιγραφεί ως εξής:

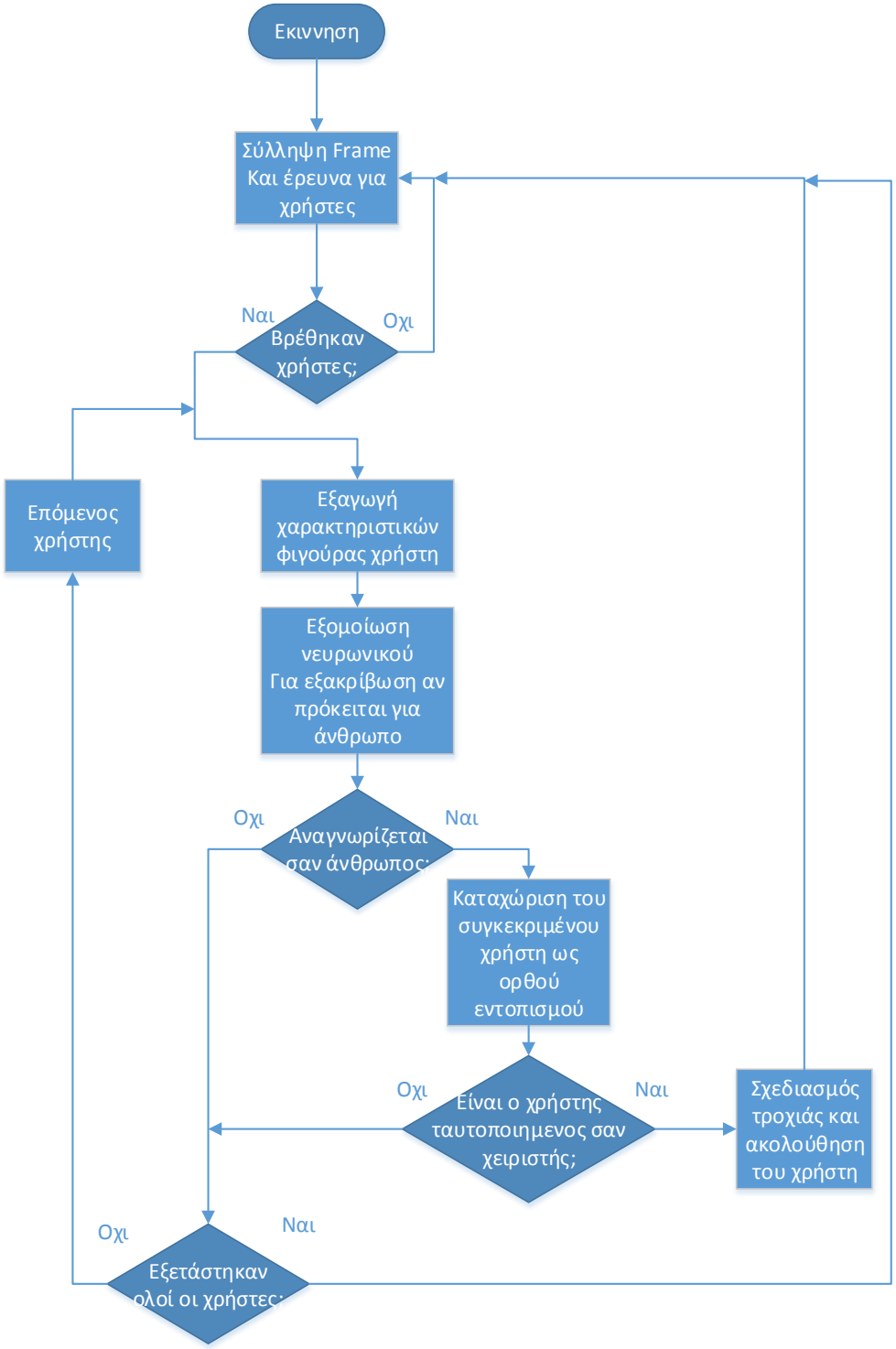
Αρχικά μια εικόνα rgb και ένας χάρτης βάθους λαμβάνεται από το action pro.Μετά το net book που βρίσκεται πάνω στο όχημα λαμβάνει τα δεδομένα αυτά και με βάση αυτά το πρόγραμμα που εκτελείται

σε αυτό προσπαθεί να αναγνωρίσει την ύπαρξη ανθρώπου. Εάν εντοπιστεί άνθρωπος γίνεται εντοπισμός του κεφαλιού και προσπάθεια ταυτοποίησης αυτού με βάση το πρόσωπο. Εάν μετά την ταυτοποίηση ο άνθρωπος ανήκει στην λίστα χρηστών το πρόγραμμα με βάση τις συντεταγμένες του και τα τυχών εμπόδια θα υπολογίσει την τροχιά του οχήματος. Με βάση την τροχιά θα υπολογιστεί η ισχύς στον κάθε κινητήρα του οχήματος. Οι εντολές που περιέχουν την επιθυμητή ισχύ των κινητήρων αποστέλλονται σειριακά στον μικροελενκτη που ελέγχει τους κινητήρες και ξεκινά η κίνηση του ρομπότ. Όσο το ρομπότ ακολουθεί τοβ χρήστη συλλέγει επιπλέων πληροφορίες που ενώ δεν είναι κατάλληλες για μακρόχρονη ταυτοποίηση είναι κατάλληλες για ταυτοποίηση μετά από προσωρινή απώλεια οπτικής επαφής, ακόμα και αν δεν είναι ορατό το πρόσωπο. Σε περίπτωση που το ρομπότ χάσει την επαφή με τον χρήστη του κινείται μέχρι το τελευταίο σημείο που υπήρχε οπτική επαφή και περιστρέφεται με σκοπό να τον βρει κάπου στο οπτικό του πεδίο. Εάν η απόσταση από τον χρήστη πάρει αρκετά μικρή τιμή το ρομπότ ακινητοποιείται μέχρι αυτός να απομακρυνθεί πάλι. Παρακάτω παρουσιάζεται το γενικό διάγραμμα της δομής του ρομπότ:

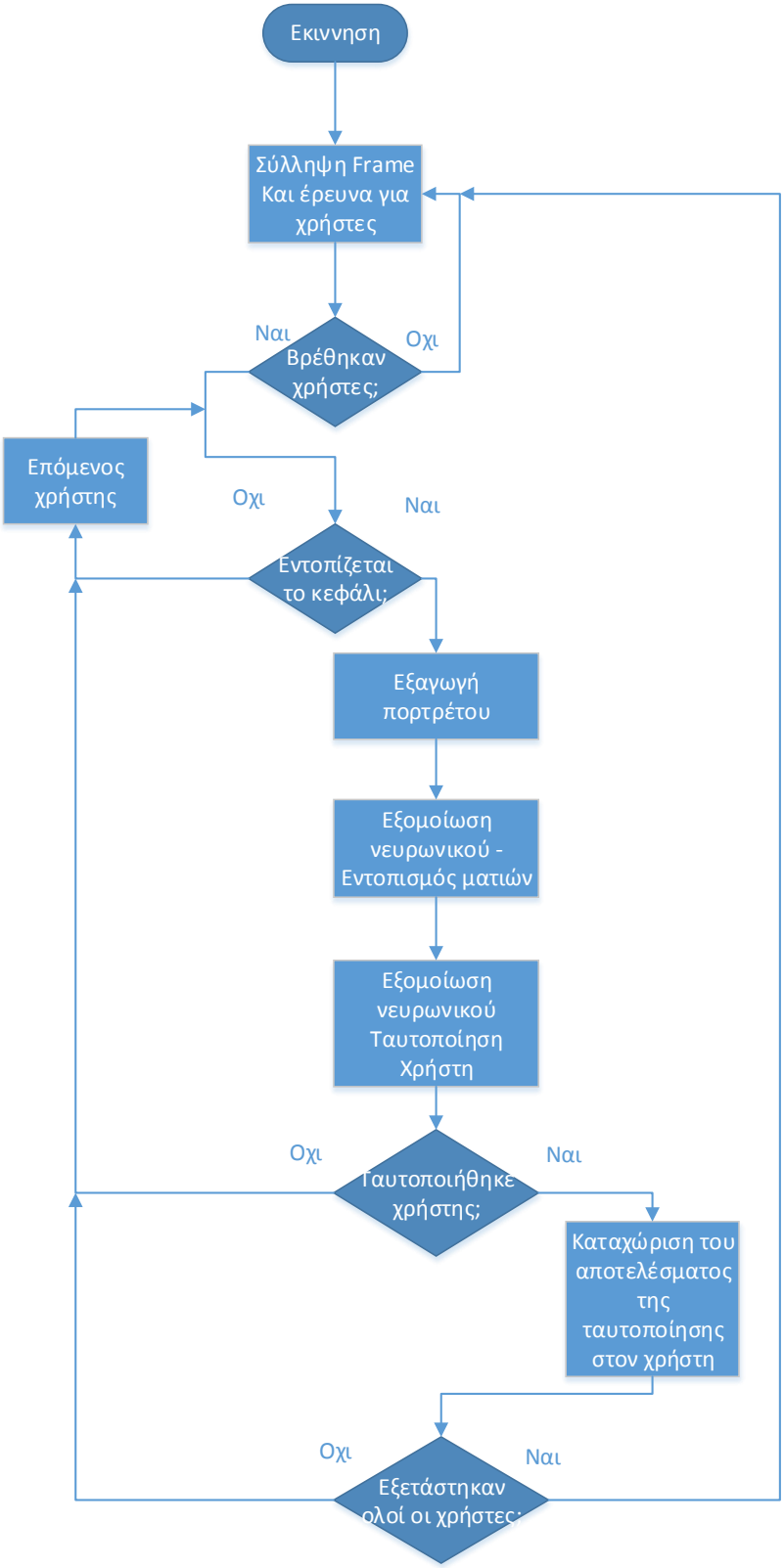


Διάγραμμα 1

Παρακάτω παρουσιάζονται σε μορφή μπλοκ διαγράμματος διάφορες υπομονάδες του software , στο διάγραμμα 2 ο τρόπος με τον οποίο αποφασίζεται αν το ρομπότ θα ακολουθήσει κάποιον και στο 3 η διαδικασία ταυτοποίησης.



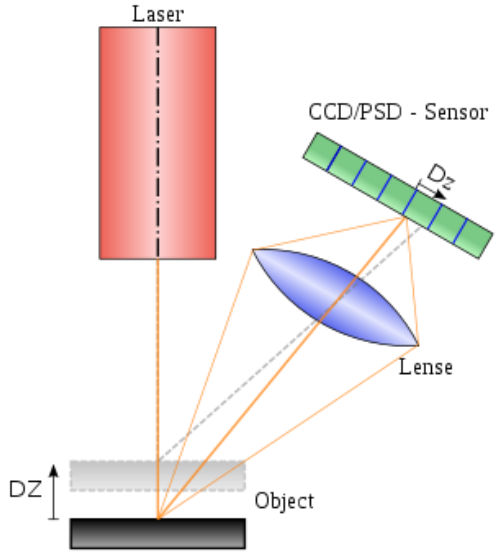
Διάγραμμα 2: Robot selection


Εκιννηση
Σύλληψη Frame Και έρευνα για χρήστες
Βρέθηκαν χρήστες;
Ναι
Οχι
Επόμενος χρήστης
Οχι
Ναι
Εντοπίζεται το κεφάλι;
Εξαγωγή πορτρέτου
Εξομοίωση νευρωνικού - Εντοπισμός ματιών
Εξομοίωση νευρωνικού Ταυτοποίηση Χρήστη
Οχι
Ταυτοποιήθηκε χρήστης;
Ναι
Καταχώριση του αποτελέσματος της ταυτοποίησης στον χρήστη
Οχι
Εξετάστηκαν ολοί οι χρήστες;
Ναι

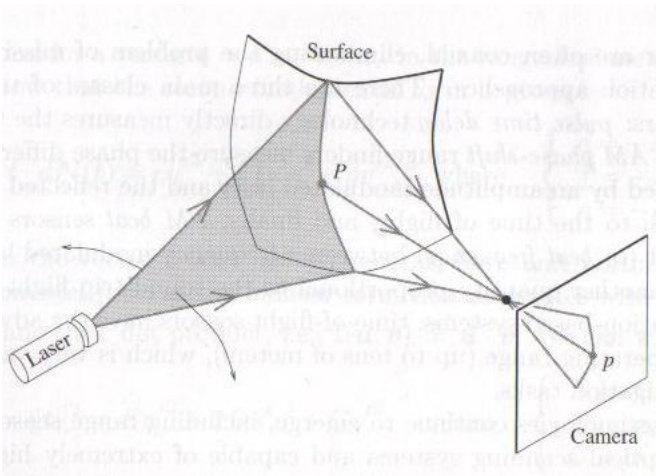
Διάγραμμα 3:

2.1 Γενικά για τις depth κάμερες

Η πλέων κλασική συσκευή που μπορεί να μας παρέχει έναν χάρτη βάθους είναι οι σαρωτές laser. Ο τρόπος λειτουργιάς τους έχει ως εξής: Μια δέσμη laser σαρώνει τον χώρο και η ανάκλαση της από τα αντικείμενα στα οποία προσπίπτει συλλαμβάνεται από μια κάμερα. Μετά με χρήση τριγωνισμού υπολογίζεται η απόσταση του αντικειμένου στο οποίο ανακλάστηκε. Στην εικόνα 1 φαίνεται πως από την ανάκλαση τις δέσμης σε διαφορετική απόσταση αυτή προσπίπτει σε άλλο pixel της κάμερας:



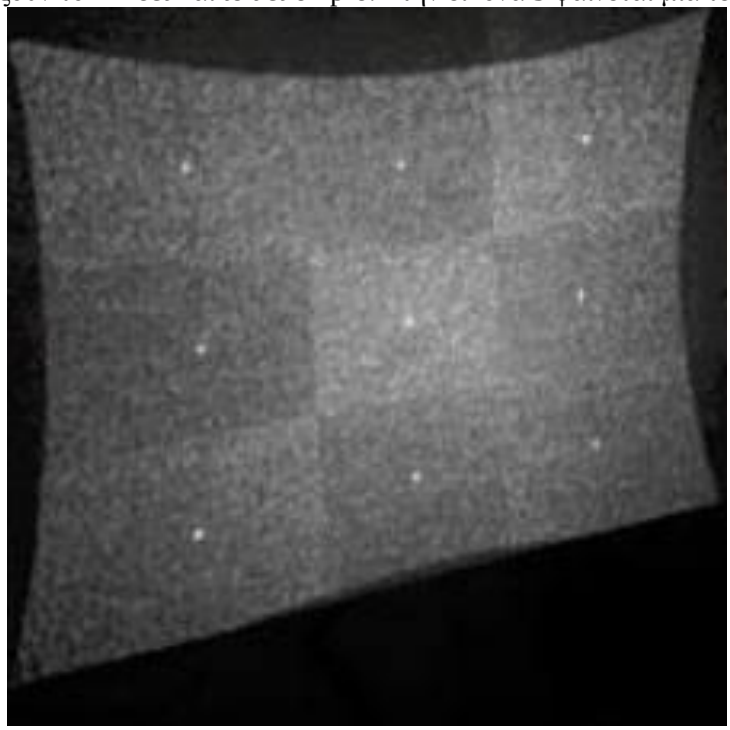
Εικόνα 1 : Υπολογισμός απόστασης μέσω τριγωνισμού



Εικόνα 2 : Σάρωση μέσω επιπέδου φωτός

Η σάρωση του χώρου από την δέσμη laser μπορεί να γίνει με δυο καθρέπτες περιστρεφόμενους με κατάλληλο τρόπο. Αυτό βέβαια έχει σαν αποτέλεσμα η ταχύτητα με την οποία γίνεται η σάρωση του χώρου να είναι αρκετά μικρή. Μια εναλλακτική είναι να μετατραπεί με χρήση ενός κυλινδρικού φακού η δέσμη laser σε ένα επίπεδο φωτός. Σε αυτήν την περίπτωση χρειάζεται μόνον ένας κινούμενος καθρέπτης και μάλλον το πιο σημαντικό είναι ότι επιταχύνεται η σάρωση του χώρου μιας και σε κάθε frame μια ολόκληρη στήλη από την τελική εικόνα βάθους συλλαμβάνεται και όχι ένα μόνον σημείο. Στην εικόνα 2 (πηγή [15]) φαίνεται τέτοιος τύπος σάρωσης. Σε κάποιες περιπτώσεις περισσότερες από μια κάμερες χρησιμοποιούνται για να βελτιώσουν την ακρίβεια. Γενικά τα συστήματα σάρωσης με χρήση τριγωνισμού και laser έχουν σαν ελαττώματα την χαμηλή ταχύτητα ,χαμένα σημεία από τη εικόνα που οφείλονται στην κάλυψη της δέσμης laser από κάποιο αντικείμενο και την παραμόρφωση ή επιπλέων απώλεια δεδομένων λόγο κακής ανάκλασης της δέσμης. Επίσης δευτερεύουσες ανακλάσεις μπορούν να δώσουν εσφαλμένες μετρήσεις. Μια άλλη επιλογή πιο πρόσφατα εμφανισμένη είναι η

προβολή ενός συγκεκριμένου πατερν φωτός(υπέρυθρο κατά προτίμηση) και η εκτίμηση της απόστασης από την παραμόρφωση αυτού καθώς η απόσταση αλλάζει. Σε αυτήν την περίπτωση δεν υπάρχουν περιστρεφόμενα μηχανικά μέρη και όλη η εικόνα βάθους προκύπτει από ένα frame.Αυτή είναι και η μέθοδος που εμφανίζουν το Kinect και το action pro. Στην εικόνα 3 φαίνεται μια τέτοια προβολή.



Εικόνα 3 : Δομημένο φως

2.2Περιγραφή του action pro live, προδιαγραφές

Στην εικόνα 4 βλέπουμε το action pro live

Εικόνα 4 : Action pro live

Και στην εικόνα 5 την εσωτερική διάταξη των αισθητήρων (πυγή [16])

Εικόνα 5 : Εσωτερικό action pro live

Το action pro live όπως αναφέραμε και στην προηγούμενη ενότητα εφαρμόζει για την δημιουργία του χάρτη βάθους την μέθοδο της προβολής ενός συγκεκριμένου πατερν υπέρυθρου φωτός και έτσι παράγει έναν ολόκληρο χάρτη βάθους ανά frame. Η διαδικασία εξαγωγής αυτής της πληροφορίας είναι αρκετά απαιτητική και για αυτόν το λόγο υπάρχει ένα ολοκληρωμένο που κάνει αποκλειστικά αυτήν την δουλεία το ps1080.
Όσον αφορά τις προδιαγραφές του αυτές φαίνονται όπως δίνονται από την εταιρία στον πινάκα 1.

Πίνακας 1 : Προδιαγραφές action pro live

Power Consumption	below 2.5W
Distance of Use	Between 0.8m and 3.5m
Field of View	58° H, 45° V, 70° D (Horizontal, Vertical, Diagonal)
Sensor	RGB& Depth& Microphone*2
Depth Image Size	VGA (640x480) : 30 fps QVGA (320x240): 60 fps
Resolution	SXGA (1280*1024)
Platform	Intel X86 & AMD
OS Support	Win 32/64 : XP , Vista, 7 Linux Ubuntu 10.10: X86,32/64 bit Android(by request)
Interface	USB2.0
Software	software development kits(OPEN NI SDK bundled)
Programming Language	C++/C# (Windows) C++(Linux) JAVA
Operation Environment	Indoor
Dimensions	18 x 3.5 x 5 cm

Όσον αφορά τα API που το υποστηρίζουν υπάρξει το open-ni και nite που προσφέρουν μια πλήρη συλλογή από βιβλιοθήκες και παραδείγματα κώδικα. Στην εφαρμογή μου έχω χρησιμοποιήσει το open-ni.

2.2.1To open-ni

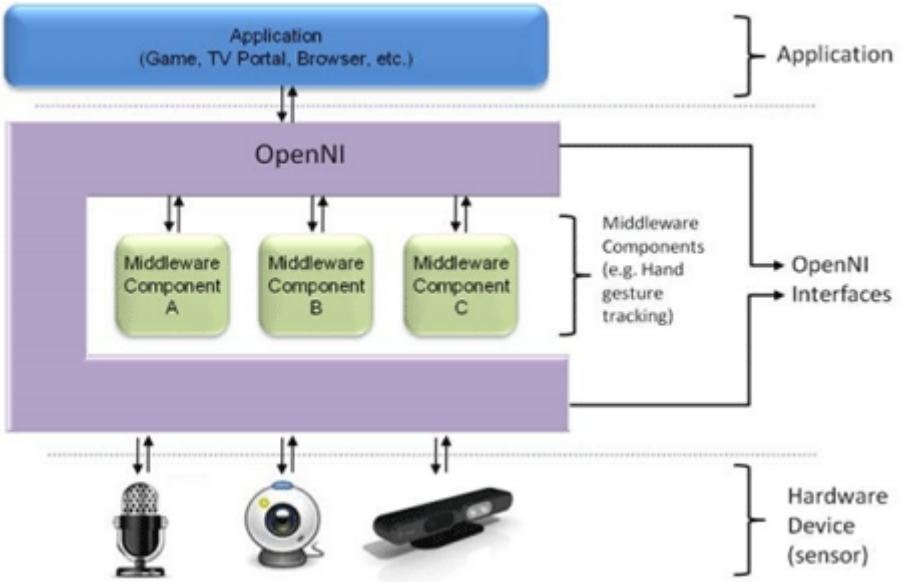
Το open-ni είναι μια μη κερδοσκοπική κοινοπραξία που ξεκίνησε τον Νοέμβριο του 2010. Σαν σκοπό έχει την προώθηση και την δημιουργία στάνταρ για την συμβατότητα και την διαλειτουργικότητα συσκευών, εφαρμογών και ενδιάμεσου λογισμικού σχετικά με το Natural Interaction (NI). Natural Interaction είναι η πιο φυσική αλληλεπίδραση του χρήστη με τον υπολογιστή. Σε αντίθεση με την επικοινωνία ανθρώπου υπολογιστή που γίνεται με πληκτρολόγιο, ποντίκι και άλλες κλασικές συσκευές επικοινωνίας με τον υπολογιστή , η επικοινωνία μέσο προφορικού λογού ή νοημάτων θεωρείται πιο φυσική γιατί είναι ένας τρόπος επικοινωνίας που θα ήταν φυσικό να γίνει μεταξύ ανθρώπων. Επίσης πληροφορίες που λαμβάνονται από το προγράμματα με πιο φυσικό τρόπο όπως για παράδειγμα οι εκφράσεις του προσώπου ,η στάση και η θέση του σώματος εμπίπτουν στην ίδια κατηγορία. Το NI δηλαδή είναι μια διεπαφή με τον χρήστη πιο φυσική.

Το API του open-ni είναι διαθέσιμο για τις παρακάτω πλατφόρμες:

- Windows XP ή μεταγενέστερο 32bit(μόνο)
- Linux Ubuntu 10.10 ή μεταγενέστερο για x86

Το open-ni παρέχει ένα σύνολο από api για να ενσωματωθούν από τις συσκευές. Υποστηρίζει και το action pro και το Kinect. Επίσης έχει ένα σύνολο από api για το middleware.Με το να μπαίνει ανάμεσα και στην συσκευή και το middleware αλλά και ανάμεσα στην εφαρμογή και το middleware δίνει την

δυνατότητα οι εφαρμογές να γράφονται ανεξάρτητα και από την συσκευή και από το middleware όπως επίσης middleware από διαφορετικούς κατασκευαστές να συνεργαστούν με διαφορετικές συσκευές. Στο διάγραμμα 4 φαίνεται σχηματικά η διασύνδεση.



Διάγραμμα 4 : Διασύνδεση εφαρμογής ,open-ni , συσκευών

Όσον αφορά τα αντικείμενα που παρέχει σαν μεσάζοντες για την συσκευή αυτά είναι:

- 3D sensor
- RGB κάμερα
- IR κάμερα
- Συσκευή ήχου(ένα μικρόφωνο ή μια ομάδα από μικρόφωνα)

Τα αντικείμενα που παρέχει από την πλευρά του ενδιάμεσου λογισμικού είναι:

- Πλήρης ανάλυση σώματος: μετά από επεξεργασία τον δεδομένων από τον αισθητήρα είναι διαθέσιμα διάφορα δεδομένα σχετικά με το σώμα όπως το κέντρο βάρους, η στάση κ.α.
- Πληροφορίες σχετικά με το χέρι: Δίνει σαν πληροφορία την θέση της παλάμης.
- Ανίχνευση χειρονομιών: Ανιχνεύει προκαθορισμένες χειρονομίες και στέλνει το ανάλογο σήμα στην εφαρμογή
- Ανάλυση σκηνής: Ένα αντικείμενο που βρίσκει πληροφορίες σχετικές με την σκηνή όπως διαχωρισμός του background και των φιγούρων, το επίπεδο του πατώματος και τον διαχωρισμό των διαφόρων φιγούρων που υπάρχουν στην σκηνή.

ΚΕΦΑΛΑΙΟ 3 : ΤΟ ΡΟΜΠΟΤΙΚΟ ΟΧΗΜΑ

3.1 Το ρομποτικό όχημα

Στης εικόνες 6,7 βλέπουμε το ρομποτικό όχημα.

Εικόνα 6 : Το ρομποτικό όχημα

Εικόνα 7 : Το ρομποτικό όχημα πλάγια όψη

Η σχεδίαση του περιλαμβάνει ένα notebook 10’’, ένα ερπυστριοφόρο όχημα και ένα κύκλωμα που ελέγχεται από το notebook μέσω σειριακής. Επίσης ένας μετατροπέας usb σε σειριακή και ένα action pro live συμμετέχουν στην σύνθεση του οχήματος. Το notebook έχει 2GB ram και επεξεργαστή Intel atom N570 στα 1.66Ghz.

3.2 Περιγραφή του ηλεκτρονικού μέρους του ρομποτικού οχήματος

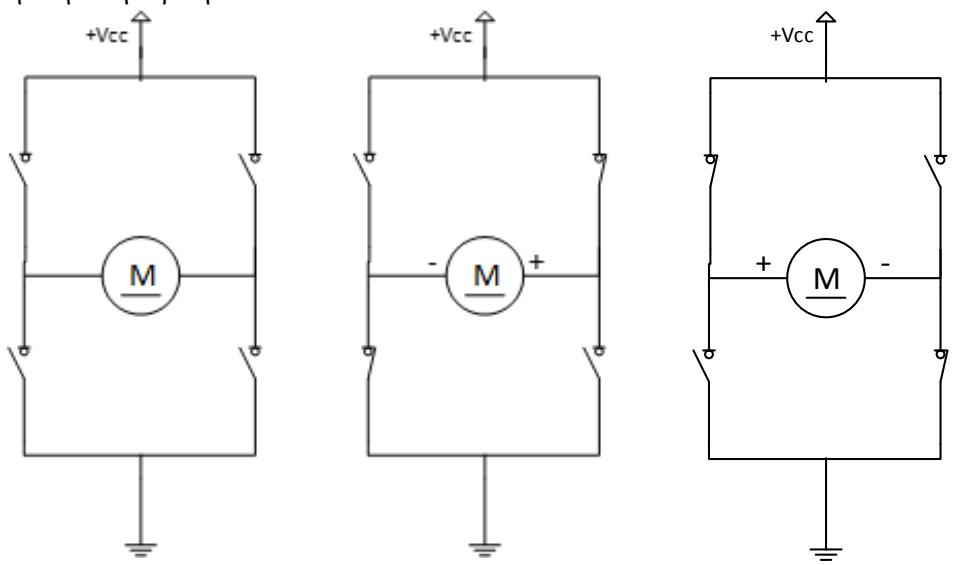
Η τροφοδοσία του κυκλώματος γίνεται από 8 μπαταριές AA συνδεμένες σε σειρά, επαναφορτιζόμενες ή μη δίνοντας στο κύκλωμα 9.6V ή 12V στην δεύτερη περίπτωση. Ο πυρήνας του ηλεκτρονικού μέρους του οχήματος είναι ο μικροελενκτής atmega8515.Οι προδιαγραφές του φαίνονται στον πίνακα 2.

Πίνακας 2 : Χαρακτηριστικά του ATMEGA8515

Flash (Kbytes):	8 Kbytes
EEPROM:	512 bytes
SRAM:	512 bytes
Pin Count:	44
Max. Operating Frequency:	16 MHz
CPU:	8-bit AVR
Max I/O Pins:	35
Ext Interrupts:	3
USB Interface:	No
Serial Interface:	Yes

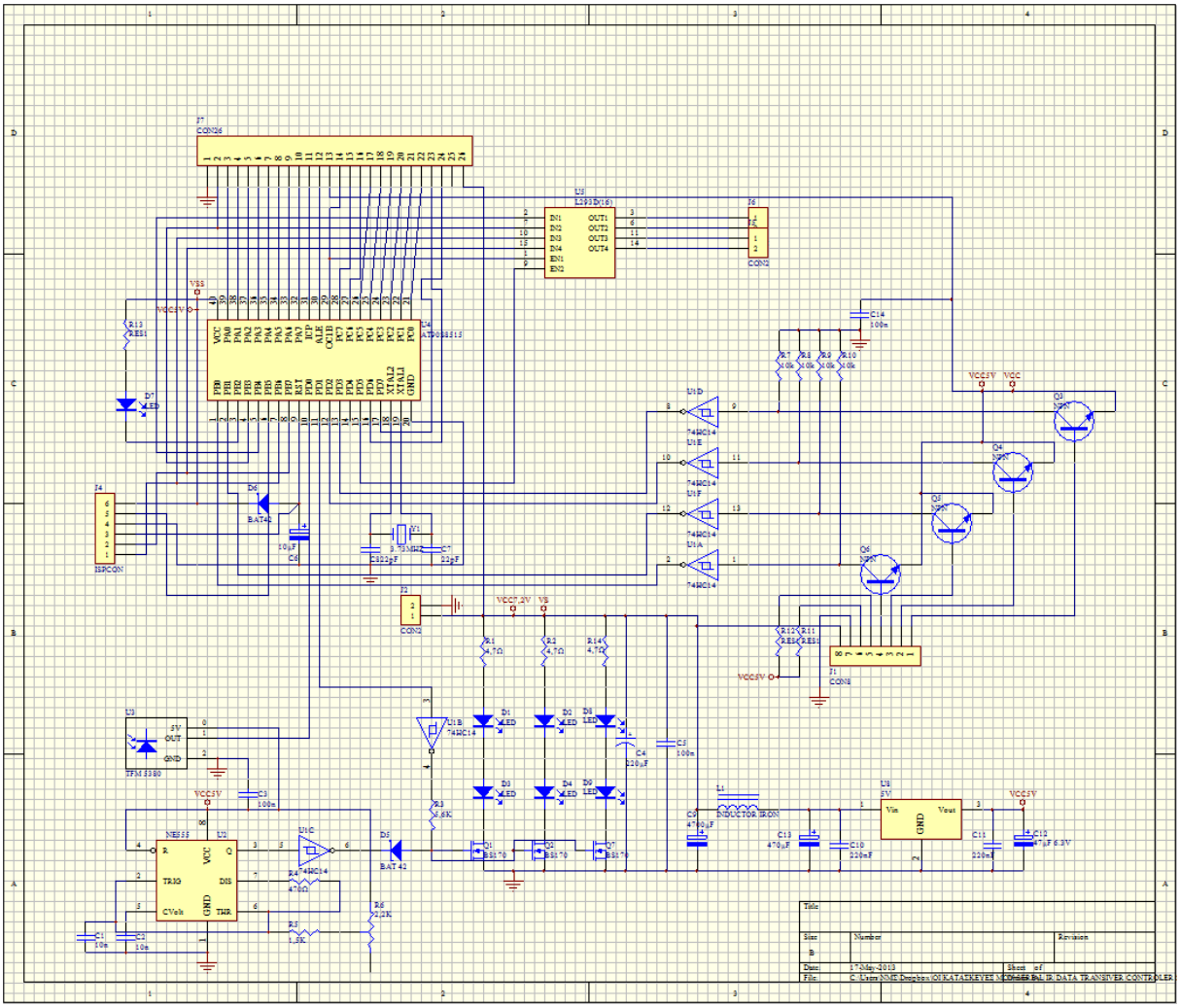
Αυτός αναλαμβάνει την διασύνδεση του ηλεκτρονικού μέρους του οχήματος με τον υπολογιστή. Λαμβάνοντας εντολές από τον υπολογιστή μέσω σειριακής τις ερμηνεύει στην κατάλληλη ενεργοποίηση των διαφόρων υποσυστημάτων της ηλεκτρονικής πλακέτας. Οι εντολές που έχουν να κάνουν με την κίνηση του οχήματος εκτελούνται από τον μικροελεγκτή με την κατάλληλη διέγερση του ολοκληρωμένου L293D το οποίο έχει μέσα του δυο γέφυρες Η. Χάρη στης γέφυρες Η είναι δυνατή η αντιστροφή πολικότητας στην τάση που δέχεται το μοτέρ και έτσι η αλλαγή στην φορά περιστροφής του. Στο διάγραμμα 5 φαίνεται η αρχή λειτουργιάς μιας γέφυρας Η. Στην πρώτη περίπτωση όλοι οι διακόπτες είναι ανοικτοί και δεν πηγαίνει καθόλου τάση στο μοτέρ. Στην δεύτερη ο πάνω δεξιά και ο κάτω

αριστερά είναι κλειστοί και έτσι το μοτέρ έχει στα δεξιά την θετική τροφοδοσία και στα αριστερά την αρνητική. Στην τρίτη το αντίθετο.

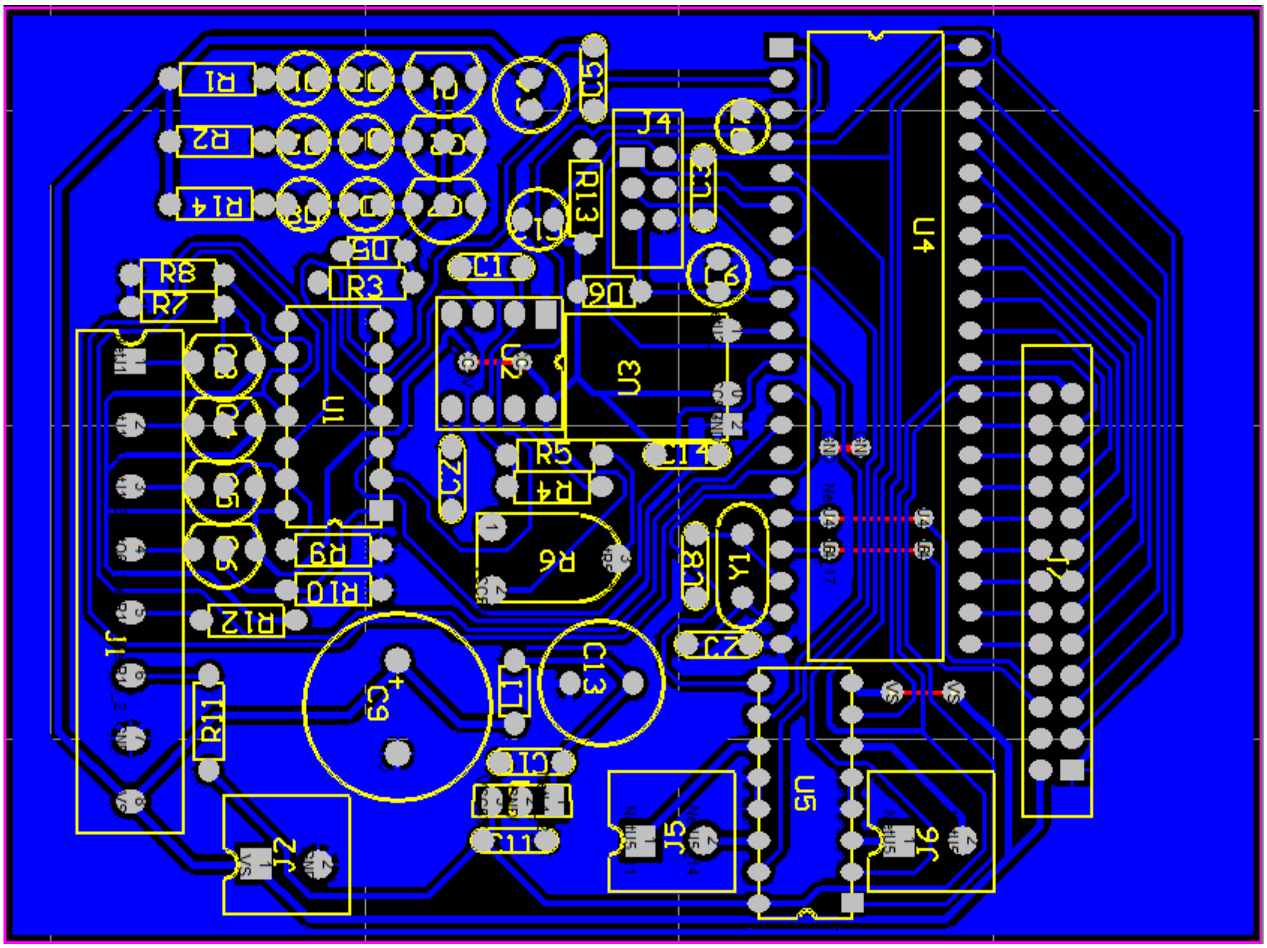


Διάγραμμα 5

Η ισχύς που δέχεται το κάθε μοτέρ μπορεί να ρυθμιστεί με χρήση της τεχνικής PWM. Η τεχνική PWM(Pulse Width Modulation) μπορεί να εφαρμοστεί εάν οι διακόπτες αντί να είναι διαρκώς κλειστοί ανοιγοκλείνουν με τον λόγο του χρόνου που είναι ανοικτοί ως προς τον χρόνο που είναι κλειστοί να καθορίζει το ποσοστό της ισχύος που θα δέχεται το μοτέρ σε σχέση με αυτό που θα δεχόταν αν ήταν διαρκώς κλειστοί. Εάν στην έξοδο της γέφυρας τοποθετηθεί το κατάλληλο φίλτρο και η συχνότητα που ανοιγοκλείνουν οι διακόπτες είναι αρκετά υψηλή το ρεύμα που θα περνά από το μοτέρ θα είναι σχετικά σταθερό και όχι διακοπτόμενο.
Στο διάγραμμα 6 φαίνεται το σχηματικό διάγραμμα του κυκλώματος. Το σχηματικό αυτό διάγραμμα σχεδιάστηκε στο protel. Στο σχηματικό διάγραμμα φαίνονται και κάποια επιπλέων εξαρτήματα που δίνουν την δυνατότητα το κύκλωμα να επικοινωνεί με τον υπολογιστή μέσω υπερύθρων τα οποία δεν χρησιμοποιούνται αλλά παρόλα αυτά υπάρχουν στο κύκλωμα. Τέτοια είναι για παράδειγμα το NE555 και τα led D1 έως D9. Γύρο από το U8 είναι φτιαγμένο το σύστημα τροφοδοσίας . Το U4 είναι ο μικροελενκτής μας και γύρο του έχει ορισμένα παθητικά εξαρτήματα απαραίτητα για την λειτουργία του. Τα 4 τρανζίστορ όπως και τα γύρο παθητικά εξαρτήματα σε συνδυασμό με το U1 που έχει 6 πύλες not τύπου smith trigger φροντίζουν κάποια σήματα από εξωτερικούς αισθητήρες να πάρουν της ανάλογες στάθμες ώστε να αποτελούν ευδιάκριτα 0 και 1. Το U5 έχει δυο γέφυρες Η που τον τρόπο λειτουργιάς τους εξήγησα αναλυτικά στην προηγούμενη παράγραφο και στο διάγραμμα 5. Στους κονέκτορες J5,6 συνδέονται τα μοτέρ. Στον J2 η τροφοδοσία. Ο κονέκτορας J7 βγάζει προς τα έξω διάφορα σήματα του μικροελενεκτή όπως επίσης και τροφοδοσίες και επιτρέπει την επέκταση του κυκλώματος. Μέσο αυτού γίνεται και η διασύνδεση μέσω σειριακής με τον υπολογιστή. Ο J4 επιτρέπει τον μέσα στο σύστημα προγραμματισμό του μικροελενκτή. Στην εικόνα 8 φαίνεται το pcb. Με μπλε οι αγωγοί, κίτρινο τα περιγράμματα των εξαρτημάτων, με γκρι τα pad και κόκκινο τα jumper.



Διάγραμμα 6 : Το σχηματικό διάγραμμα του κυκλώματος .



Εικόνα 8 : Το pcb του κυκλώματος

3.3 Το όχημα

Το όχημα πάνω στο οποίο προστέθηκε επιπλέων εξοπλισμός ώστε να μετατραπεί σε ρομπότ πρόεκυψε από ένα παιχνίδι το οποίο ήταν δυνατόν να τηλεκατευθύνθει μέσω καλωδίου. Από αυτό αφαιρεθήκαν τα ηλεκτρονικά του μέρη για να τοποθετηθούν τα κατάλληλα για την χρήση του ως ρομπότ, προσαρμόστηκε μια ξύλινη βάση ώστε να μπορεί να τοποθετηθεί το notebook, και ένα ξύλινο στήριγμα ώστε να μπορεί να τοποθετηθεί το action pro. Το συνολικό βάρος του οχήματος με όλα τα υποσυστήματα τοποθετημένα πάνω του είναι 4 κιλά. Το μήκος 32 εκατοστά ,το ύψος μέχρι το notebook 20 εκατοστά και μαζί με το action pro 28. Το πλάτος είναι 22 εκατοστά. Η κάθε ερπύστρια κινείται με δικό της μοτέρ και έτσι είναι δυνατή η στροφή του οχήματος αν η κάθε μια κινείται με άλλη ταχύτητα.

ΚΕΦΑΛΑΙΟ 4 : ΤΑΥΤΟΠΟΙΗΣΗ ΧΡΗΣΤΗ

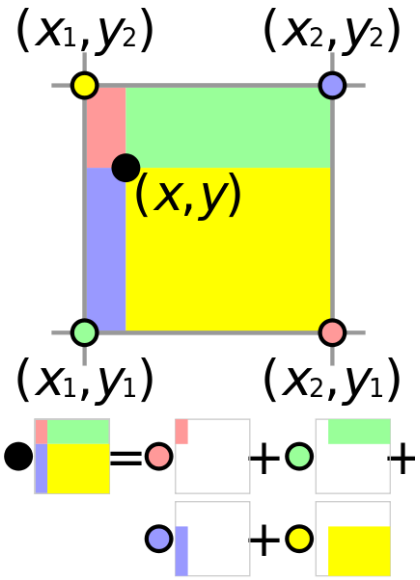
4.1 Ταυτοποίηση χρηστή

Απαραίτητη προϋπόθεση για την χρηστικότητα του ρομπότ είναι να μπορεί να αναγνωρίζει τον χρήστη του. Αυτό είναι απαραίτητο τόσο γιατί δεν είναι δυνατόν κατά την διάρκεια που ακολουθεί κάποιον να αρχίσει να ακολουθεί κάποιον άλλο όσο και επειδή δεν πρέπει να αρχίσει να ακολουθά κάποιον που δεν είναι ιδιοκτήτης του. Σαν χαρακτηριστικό που επιτρέπει την ταυτοποίηση με την μεγαλύτερη σιγουριά επιλέχτηκε η αναγνώριση προσώπου. Σε κανένα άλλο χαρακτηριστικό από αυτά που είναι κατάλληλα για οπτική αναγνώριση δεν μπορούμε να στηριχτούμε όσο στην αναγνώριση προσώπου γιατί όλα τα υπόλοιπα είναι εύκολο να αλλάξουν (όπως για παράδειγμα χρώμα ρούχων, μαλλιών κ.τ.λ.). Επίσης τα άλλα χαρακτηριστικά που δεν είναι δυνατόν να αλλάξουν όπως στοιχεία του σωματότυπου δεν είναι επαρκή για αναγνώριση. Επομένως για να ξεκινήσει να ακόλουθα το ρομπότ τον χρήστη και να εκτελέσει τυχών άλλες εντολές του θα πρέπει να αναγνωρίσει το πρόσωπο του. Από εκεί και έπειτα, με εξαίρεση το ενδεχόμενο να χαθεί η οπτική επαφή με τον χρήστη σε αρκετό ποσοστό ώστε να χαθεί το user tracking,δεν χρειάζεται κάποιο άλλο χαρακτηριστικό ταυτοποίησης ώστε να διασφαλιστεί η εκτέλεση εντολών από το σωστό άτομο καθώς επίσης δεν είναι πλέων απαραίτητη η οπτική επαφή με το πρόσωπο. Παρόλα αυτά επειδή υπάρχει το ενδεχόμενο προσωρινής απώλειας της οπτικής επαφής ή του user tracking χρησιμοποιούνται και άλλα στοιχεία αναγνώρισης. Τα στοιχεία αυτά αν και μπορούν να αλλάξουν από συνάντηση σε συνάντηση με το ρομπότ είναι κατάλληλα για αναγνώριση μετά την αρχική ταυτοποίηση. Για αυτόν τον λόγο το κεφάλαιο αυτό χωρίζεται σε δυο μέρη την ,αναγνώριση μέσω προσώπου (που είναι και η αρχική αλλά και πιο προσεγμένη προσέγγιση) και την αναγνώριση μέσω άλλων χαρακτηριστικών.

Μέρος πρώτο: αναγνώριση μέσω προσώπου

4.2 Εξαγωγή του πορτρέτου του προσώπου μέσο skeleton tracking, προεπεξεργασία

Προκειμένου να γίνει η ταυτοποίηση μέσω προσώπου θα πρέπει πρώτα να εντοπιστεί το πρόσωπο και αφού γίνει η απαραίτητη προεπεξεργασία να γίνει η ταυτοποίηση προσώπου. Η προεπεξεργασία περιλαμβάνει την αλλαγή κλίμακας και την αλλαγή της περιοχής τιμών των pixel της εικόνας έτσι ώστε να βρίσκονται στην περιοχή -1,1 αντί για 0,255,περιοχη τιμών πιο κατάλληλη για νευρωνικά δίκτυα. Το action pro προσφέρει μέσω των βιβλιοθηκών του την δυνατότητα user tracking που μεταξύ άλλων δίνει έναν πίνακα που χαρακτηρίζει τα pixel της εικόνας με βάση το αν ανήκουν στον χρήστη η όχι. Προκειμένου να αφαιρεθούν οι άσχετες με την αναγνώριση του χρήστη πληροφορίες η rgb εικόνα περνιέται με μάσκα τον προηγούμενο πίνακα και κάθε pixel που δεν ανήκει στον χρήστη παίρνει την τιμή μηδέν. Το action pro προσφέρει επίσης την δυνατότητα του skeleton tracking,δηλαδή του εντοπισμού σημείων στο σώμα ενός ανθρώπου όπως αγκώνες, γόνατα κορμός, κεφάλι καθώς επίσης και μέσω της διασύνδεσης αυτών μιας εικόνας για την στάση του σώματος. Προκειμένου να βρούμε το πρόσωπο χρησιμοποιούμε την πληροφορία για το που εντοπίζεται το κεφάλι. Μετά ένα τετράγωνο σταθερού μεγέθους αποκόπτεται με κέντρο του τετραγώνου τις συντεταγμένες που δίνει ο μηχανισμός του skeleton tracking.Επειδή η εικόνα που περιέχεται μέσα στο τετράγωνο έχει περάσει από την μάσκα του user tracking δηλαδή κάθε pixel που δεν ανήκει στον χρήστη έχει τη τιμή μηδέν εύκολα αφαιρούνται από το τετράγωνο αυτό οι περιοχές που αποτελούνται αποκλειστικά από μηδενικά pixel.Μετά στο μικρότερο(εάν έχει αφαιρεθεί κάτι) τετράγωνο που έχει απομείνει εκτελείται διαδικασία αλλαγής κλίμακας ώστε το νευρωνικό να τροφοδοτείται με πορτραίτο σταθερού μεγέθους στην είσοδο του. Το μέγεθος αυτό είναι 60X60 pixel. Η τεχνική μεγέθυνσης που χρησιμοποιείται είναι η bilinear interpolation. Η τεχνική αυτή επιλέχτηκε αντί της τεχνικής του πλησιέστερου γειτονικού pixel ή άλλων πιο εκλεπτυσμένων, γιατί δίνει ικανοποιητικά καλά αποτελέσματα χωρίς ο αλγόριθμος να είναι ιδιαίτερα περίπλοκος ‘αυτήν την τεχνική η τιμή του κάθε pixel υπολογίζεται από τον σταθμισμένο μέσο όρο των τεσσάρων πλησιέστερων pixel. Όσο πιο κοντά είναι το pixel στην θέση από την οποία θα έπρεπε να πάρουμε τιμή βάση υποδειγματοληψίας τόσο μεγαλύτερη η συμμετοχή στον μέσο όρο. Στην εικόνα 9 φαίνεται με απεικόνιση στα εμβαδά η συμμετοχή κάθε ενός από τα πλησιέστερα pixel.(πηγή [13])



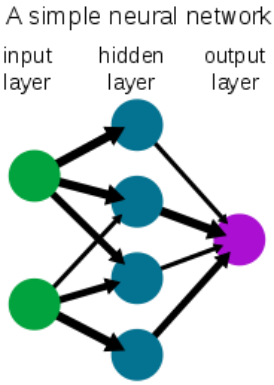
Εικόνα 9 : Σχηματική απεικόνιση των ποσοστών συμμετοχής των γειτονικών pixel στον αλγόριθμο bilinear interpolation.

Μετά προκειμένου να περιοριστούν οι τιμές των pixel στην περιοχή -1,1 αρχικά αφαιρείται από κάθε pixel το 127 και μετά το αποτέλεσμα διαιρείται με το 128.

4.3 Γενικά για τα νευρωνικά δίκτυα

Τα τεχνητά νευρωνικά δίκτυα είναι δομές επεξεργασίας δεδομένων που είναι εμπνευσμένες από το νευρικό σύστημα ανθρώπων και ζώων. Μπορεί να είναι είτε υλοποιημένα με κυκλώματα σε ολοκληρωμένα VLSI ή να εξομοιώνονται μέσω διαφόρων αλγόριθμων. Είναι ουσιαστικά μια μέθοδος αναγνώρισης προτύπων. Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από ‘’νευρώνες’’ που προσπαθούν να μιμηθούν την συμπεριφορά των πραγματικών νευρώνων. Σε ένα φυσικό νευρωνικό δίκτυο οι νευρώνες συνδέονται μεταξύ τους μέσο συνάψεων και κάθε κύτταρο μπορεί να είναι συνδεδεμένο με πλήθος άλλον νευρικών κυττάρων μέσο τον συνάψεων αυτόν. Ένα νευρικό κύτταρο δέχεται τις διεγέρσεις άλλων κυττάρων μέσω συνάψεων που μπορεί να υποβοηθούν την δική του διέγερση ή να την αναστέλλουν. Για να μιμηθούν αυτή την λειτουργιά τα τεχνητά νευρωνικά δίκτυα προσομοιώνουν την λειτουργιά του νευρώνα μέσο του μοντέλου του νευρώνα. Το μοντέλο του νευρώνα έχει ως εξής: κάθε είσοδος του νευρώνα πολλαπλασιάζεται με έναν αριθμό θετικό ή αρνητικό που το μέτρο του αντιστοιχεί στο πόσο ισχυρή είναι η σύναψη δηλαδή πόσο έντονα επηρεάζει η συγκεκριμένη είσοδος τον νευρώνα. Το πρόσημο του αριθμού αντιστοιχεί στο αν η είσοδος είναι διεγερτική ή ανασταλτική. Οι αριθμοί αυτοί ονομάζονται συναπτικά βάρη ή για συντομία απλός βάρη. Επίσης ανάμεσα στις εισόδους υπάρχει και μια που έχει πάντα την τιμή 1 και δεν αντιστοιχεί σε κάποια εξωτερική διέγερση αλλά στην πόλωση του κυττάρου. Και αυτή η είσοδος έχει το δικό της βάρος. Τα αποτελέσματα αυτών τον πολλαπλασιασμών προστίθενται για να προκύψει η συνολική διέγερση του κυττάρου. Το άθροισμα αυτό δίνεται σαν είσοδος στην συνάρτηση ενεργοποίησης που αντιστοιχεί στον τρόπο με τον οποίο ενεργοποιείται το κύτταρο βάση της διέγερσης. Το αποτέλεσμα της συνάρτησης είναι η διέγερση του κυττάρου. Επίσης υπάρχουν οι νευρώνες εισόδου που δεν έχουν σαν πηγή διέγερσης άλλους νευρώνες αλλά κάποιο εξωτερικό ερέθισμα. Σε ένα φυσικό νευρωνικό δίκτυο θα μπορούσε να είναι για παράδειγμα κάποιο κωνικό κύτταρο του οφθαλμού που η διέγερση του εξαρτάτε μόνο από το φως που προσπίπτει πάνω του και όχι από αλλά νευρωνικά κύτταρα. Αυτοί εξομοιώνονται πολύ απλά με έξοδο το επίπεδο της εξωτερικής διέγερσης δηλαδή την εξωτερική είσοδο του νευρωνικού δικτύου. Ένα νευρωνικό δίκτυο με έναν μόνο νευρώνα που οι είσοδοι του συνδέονται μόνο σε νευρώνες εισόδου δεν μπορεί να ταξινομήσει παρά μόνο γραμμικά διαχωριζόμενα δεδομένα. Περισσότεροι νευρώνες που όλοι τους είναι συνδεμένοι αποκλειστικά με νευρώνες εισόδου απλός μπορούν να υλοποιήσουν περισσότερους γραμμικούς ταξινομητές. Ο τρόπος με τον οποίο ξεπερνιέται αυτό το πρόβλημα είναι παρεμβάλλοντας τουλάχιστον ένα ακόμα στρώμα νευρώνων ανάμεσα στους νευρώνες εισόδου και στους νευρώνες εξόδου που η συνάρτηση τους είναι μη γραμμική, εάν είναι γραμμική υπάρχει ένα ισοδύναμο νευρωνικό δίκτυο χωρίς το επιπλέων αυτό ενδιάμεσο στρώμα οπότε δεν προσφέρει και κάτι η προσθήκη του. Συνήθεις μη γραμμικές συναρτήσεις που χρησιμοποιούνται είναι οι logsig και η tansig. Σε όλα τα νευρωνικά δίκτυα που έχω χρησιμοποιήσει η συνάρτηση μεταφοράς είναι η tansig τύπος της οποίας είναι tansig(a) =

$\frac{2}{1+e^{-2*a}} - 1$. Το στρώμα αυτό ονομάζεται κρυφό στρώμα γιατί δεν έχει άμεση επαφή με τον έξω κόσμο ούτε από την πλευρά της εισόδου ούτε από την πλευρά της εξόδου. Στην εικόνα 10 φαίνεται ένας συνηθισμένος τρόπος απεικόνισης ενός νευρωνικού δικτύου. Το συγκεκριμένο έχει ένα κρυφό στρώμα. Ο κάθε κύκλος συμβολίζει έναν νευρώνα και τα βελάκια τις συνάψεις με την αρχή του βέλους να ξεκινά από τον νευρώνα που διεγείρει και να καταλήγει στον νευρώνα που διεγείρεται.



Εικόνα 10 : Συνήθης τρόπος σχηματικής αναπαράστασης ενός νευρωνικού δικτύου (πυγή [13])

Με την χρήση του κρυφού στρώματος ο περιορισμός του γραμμικού διαχωρισμού δεν υπάρχει πλέον. Επίσης ο αριθμός των νευρώνων του κρυφού στρώματος δεν καθορίζεται από τον αριθμό των εισόδων ούτε από τον αριθμό τον εξόδων αλλά από τις επιλογές του σχεδιαστή. Αυξάνοντας τον αριθμό των νευρώνων στο κρυφό στρώμα αυξάνεται και ο αριθμός των συναπτικών βαρών και είναι δυνατή η ενσωμάτωση μεγαλύτερης ποσότητας πληροφορίας προκειμένου να περιγραφή μια πιο περίπλοκη σχέση εισόδου εξόδου. Επίσης δεδομένης της ευελιξίας που υπάρχει στο κρυφό στρώμα μπορεί κάποιοι νευρώνες στο κρυφό στρώμα να μην συνδέονται ούτε με όλους τους νευρώνες εισόδου, ούτε με όλους τους νευρώνες εξόδου επιτρέποντας έτσι περισσότερες γνώσεις σχετικά με την φύση του προβλήματος να ενσωματωθούν στην δομή του νευρωνικού δικτύου. Το βασικό πρόβλημα που υπήρχε με τα νευρωνικά δίκτυα που περιείχαν κρυφά στρώματα ήταν ότι δεν ήταν δυνατόν να εκπαιδευτούν με τους αλγορίθμους που εκπαιδεύονταν οι πιο απλές εκδοχές χωρίς κρυφό στρώμα όπως τα perseptron και τα adaline. Έτσι ουσιαστικά δεν ήταν δυνατή η χρήση τους μέχρι την εύρεση ενός αποτελεσματικού αλγορίθμου μάθησης πράγμα που άλλαξε με την εύρεση του αλγορίθμου back propagation. Επειδή ο αλγόριθμος αυτός έκανε δυνατή την χρήση τους καθώς επίσης και επειδή είναι κατάλληλος μόνο για τέτοιου τύπου νευρωνικά δίκτυα τα δίκτυα αυτά συνήθως αποκαλούνται και backpropagation. Υπάρχουν ακόμα νευρωνικά δίκτυα με πιο περίπλοκη ή διαφορετική δομή όπως για παράδειγμα τα Hopfield που οι έξοδοι των νευρώνων του στρώματος εξόδου ανατροφοδοτούν προηγούμενα στρώματα στο νευρωνικό δίκτυο. Επίσης άλλο ένα ενδιαφέρον παράδειγμα είναι τα kohonen SOMs που βασίζονται στην διαπίστωση ότι στους εγκεφάλους των θηλαστικών οι νευρώνες οργανώνονται σε ένα είδος «χαρτών» όπου γειτονικοί νευρώνες ενεργοποιούνται από παρόμοιες εισόδους. Όμως επειδή δεν χρησιμοποιώ άλλους τύπους νευρωνικών δικτύων πέραν των backpropegation δεν θα προχωρήσω στην ανάλυση τους. Ένα ακόμα σημαντικό θέμα σχετικά με τα νευρωνικά δίκτυα είναι το πώς τα βάρη θα πάρουν τις κατάλληλες τιμές. Υπάρχουν αλγόριθμοι που με βάση ένα σύνολο παραδειγμάτων εισόδου εξόδου μπορούν μέσω μιας διαδικασίας που λέγεται εκπαίδευση μπορούν να δώσουν τις κατάλληλες τιμές στα συνοπτικά βάρη. Το γεγονός ότι τα τεχνητά νευρωνικά δίκτυα μπορούν να ‘’μάθουν’’ την επιθυμητή τους απόκριση στις διάφορες πιθανές εισόδους τους είναι ένα ιδιαίτερα σημαντικό πλεονέκτημα. Είναι δυνατή έτσι η ταξινόμηση προτύπων χωρίς να γνωρίζει ο κατασκευαστής μια αφηρημένη περιγραφή της σχέσης εισόδου-εξόδου των δεδομένων αλλά μόνον παρέχοντας στον αλγόριθμο εκπαίδευσης τα κατάλληλα παραδείγματα. Προκειμένου να γίνει η εκπαίδευση του νευρωνικού δικτύου σε δίκτυα με ένα ή περισσότερα κρυφά στρώματα μια δυνατή επιλογή είναι ο αλγόριθμος οπισθοδιάδοσης σφάλματος(backpropagation).

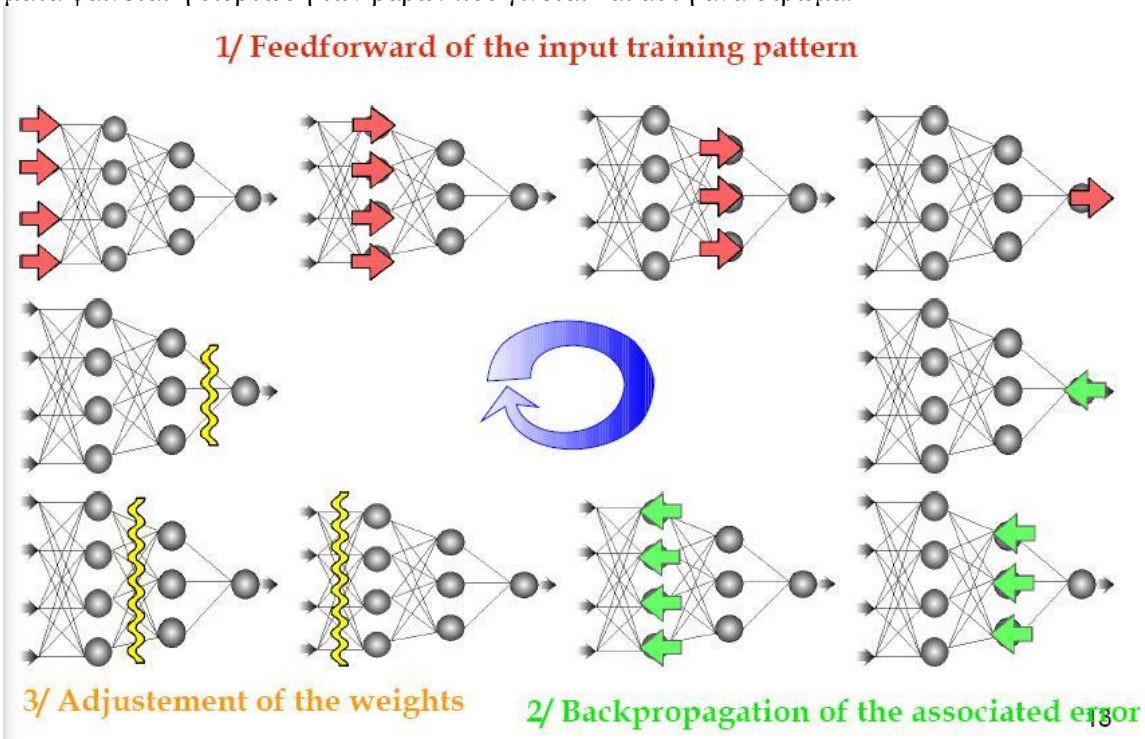
Προκειμένου να εκτελεστεί ο αλγόριθμος αυτός σαν πρώτο βήμα τα βάρη του δικτύου περνούν τυχαίες τιμές.

Έπειτα σαν δεύτερο βήμα εφαρμόζονται τα δεδομένα εισόδου από το πρώτο παράδειγμα στο νευρωνικό

και υπολογίζεται αρχικά η έξοδος του και έπειτα η διάφορα από την επιθυμητή με βάση το παράδειγμα. Η έξοδος του κάθε νευρώνα υπολογίζεται από τον παρακάτω τύπο : $g(h_j)$ οπου για τον υπολογισμό του h_j ισχειει $h_j = \sum_i w_{j,i}\, x_i$. Η $g(h_j)$ είναι η συναρτηση μεταφορας του νευρωνα.

Σαν τρίτο βήμα τα βάρη του δικτύου μεταβάλλονται με βάση τον τύπο:
$w(t+1)_{j,i} = w(t)_{j,i} + a\delta_j out_i$.
Το t είναι χρόνος πριν την εκτέλεση του βήματος και t+1 μετά. Το $w_{j,i}$ είναι το βάρος σύνδεσης του νευρώνα j με την i είσοδο του. Το η είναι ένας συντελεστής που ονομάζεται ρυθμός μάθησης και καθορίζει πόσο πολύ αλλάζουν οι τιμές τον βαρών σε κάθε κύκλο μάθησης. Το out_i είναι εξοδος του νευρονα i δηλαδή η είσοδος i του νευρώνα j. Το δ_j υπολογιζεται διαφορετικα εάν ο νευρώνας είναι νευρώνας εξόδου ή κρυφού στρώματος. Πιο συγκεκριμένα στην περίπτωση των νευρώνων εξόδου ισχύει:
$\delta_j = (Tar_j - out_j)g'(h_j)$.
Στους νευρώνες των κρυφών στρωμάτων ισχύει:
$\delta_j = g'(h_j)\sum_\kappa(\delta_\kappa\, w_{\kappa,j})$.
Εφόσον οι παράμετροι των παραπάνω εξισώσεων είναι γνωστοί όλα τα βάρη του δικτύου ενημερώνονται.

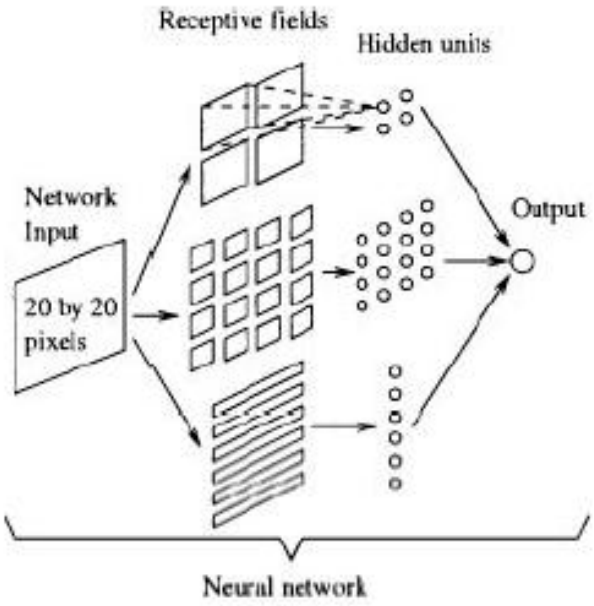
Σαν τέταρτο βήμα περνάμε στο επόμενο παράδειγμα και επαναλαμβάνουμε συνεχίζοντας από το βήμα 2 μέχρι να γίνει αυτό για όλα τα παραδείγματα. Μετά εάν το σφάλμα είναι αρκετά μικρό ο αλγόριθμος τερματίζει. Εάν δεν είναι ξανά επιστρέφουμε στο βήμα 2 από το πρώτο παράδειγμα ώμος. Κάθε πλήρες πέραμα από όλα τα δείγματα ονομάζεται εποχή. Συνήθως χρειάζονται πολλές εποχές για την εκπαίδευση του νευρωνικού δικτύου. Από την άλλη δεν είναι καλό να διαρκεί η εκπαίδευση του δικτύου υπερβολικά πολλές εποχές γιατί συμβαίνει υπερεκπεδευση του νευρωνικού δικτύου και δεν μπορεί να ταξινομήσει σωστά παρά μόνο εισόδους πανομοιότυπες με τα αρχικά παραδείγματα. Στην εικόνα 11 φαίνεται σχηματικά η λειτουργιά του αλγορίθμου. Με κόκκινα βελάκια συμβολίζεται ο από στρώμα σε στρώμα υπολογισμός της εξόδου των νευρώνων δηλαδή το βήμα 2. Με πράσινα ο υπολογισμός η οπισθοδιάδοση του σφάλματος και ο βάση αυτής ο υπολογισμός του δ_j. Παρατηρούμε ότι ο υπολογισμός αυτός γίνεται με αντίθετη φορά από ότι ο προηγούμενος δηλαδή από την έξοδο προς την είσοδο. Τέλος με κίτρινα κύματα φαίνεται η διόρθωση των βαρών που γίνεται και αυτή ανά στρώμα.



Εικόνα 11 : Σχηματική αναπαράσταση του αλγορίθμου back propagation

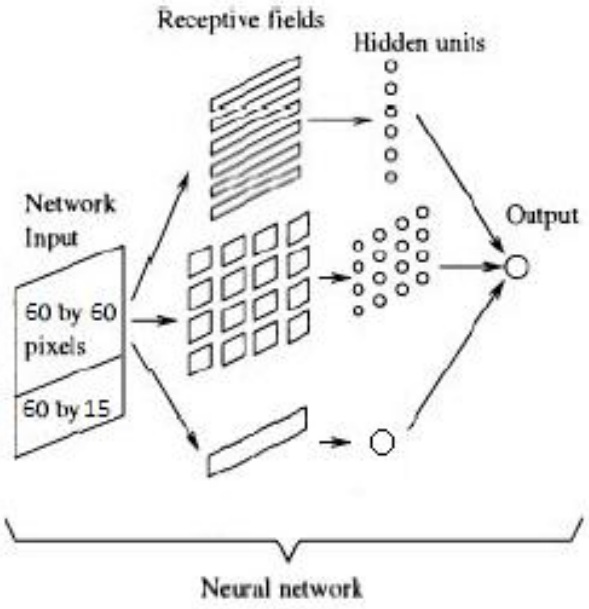
4.4 Δομή νευρωνικού δικτύου

Η δομή του νευρωνικού δικτύου που κάνει την αναγνώριση του προσώπου είναι εμπνευσμένη με αρκετές τροποποιήσεις από την εργασία του Rowley [17] .Το νευρωνικό δίκτυο που παρουσιάζεται στην εργασία του Rowley[17] έχει σαν σκοπό τον εντοπισμό τον προσώπων που υπάρχουν μέσα σε μια εικόνα και όχι την αναγνώριση. Επίσης η εργασία του είναι από τις λίγες που τα δεδομένα από την εικόνα τροφοδοτούν το νευρωνικό δίκτυο με ελάχιστη προεπεξεργασία. Με το σκεπτικό ότι για να είναι δυνατόν να κάνει αναγνώριση προσώπων το νευρωνικό αυτό θα πρέπει αναγκαστικά να είναι σε θέση να εντοπίζει κάποια χαρακτηριστικά του προσώπου θεώρησα ότι η δομή του συγκεκριμένου νευρωνικού δικτύου θα είναι μια καλή επιλογή σαν βάση. Η εικόνα 12 παρουσιάζει την δική του δομή.



Εικόνα 12 : Η δομή του νευρωνικού δικτύου του Rowley

Στην αρχική υλοποίηση σκοπός ήταν απλός ο εντοπισμός ενός προσώπου και έτσι λίγοι σχετικά νευρώνες ήταν αρκετοί στο κρυφό στρώμα, όπως και μια ανάλυση 20X20.Στην δική μου περίπτωση δεν ήταν αρκετό να ενεργοποιείται ένας νευρώνας για παράδειγμα αν υπάρχει κάτι που μοιάζει με στόμα αλλά θα έπρεπε να υπάρχουν αρκετοί νευρώνες ώστε να ενεργοποιούνται από διαφορετικές μορφές ενός στόματος και έτσι να γίνεται διάκριση. Με βάση αυτήν την υπόθεση προσπάθησα απλός αυξάνοντας τον αριθμό των νευρώνων και της ανάλυση της εικόνας εισόδου να κάνω αναγνώριση προσώπου. Παρόλο που το νευρωνικό δίκτυο που προέκυψε είχε κάποιες δυνατότητες αναγνώρισης τα αποτελέσματα δεν ήταν ικανοποιητικά. Έτσι πρόσθεσα στο κρυφό στρώμα μια ακόμα ομάδα νευρώνων που θα τροφοδοτούνταν από την περιοχή των ματιών και το νευρωνικό μου πήρε τη μορφή που φαίνεται στην εικόνα 13.



Εικόνα 13 : Η δομή του νευρωνικού δικτύου αναγνώρισης χρηστών

Αφού δεν ήταν σταθερή και εκ τον προτέρων γνωστή η θέση των ματιών(τόσο η γεωμετρία του προσώπου την επηρεάζει όσο και ο προσανατολισμός του προσώπου),δυο επιπλέων νευρωνικά δίκτυα χρησιμοποιήθηκαν για τον εντοπισμό του ύψους των ματιών. Το πρώτο είχε δυο εξόδους με την μια να ενεργοποιείται από την ύπαρξη ματιών και την άλλη από την ανυπαρξία. Μια σάρωση της εικόνας πραγματοποιούνταν με το νευρωνικό να παίρνει σαν είσοδο διαφορετικές περιοχές της εικόνας και τα αποτελέσματα να καταχωρούνται σε έναν πίνακα. Το δεύτερο είχε 25 εξόδους και ενεργές ήταν οι έξοδοι που αντιστοιχούσαν στο ύψος που γινόταν εντοπισμός ματιών. Η διαφορά των εξόδων του πρώτου νευρωνικού δικτύου προσταζόταν στην έξοδο της αντίστοιχης περιοχής από το δεύτερο δίκτυο πολλαπλασιασμένης με έναν συντελεστή βαρύτητας. Η περιοχή που θα είχε το μεγαλύτερο σκορ εάν αυτό ξεπερνούσε ένα κατώφλι θα αναγνωριζόταν σαν το ύψος που βρίσκονται τα μάτια και η γύρο περιοχή δινόταν σαν είσοδος στην ομάδα νευρώνων εισόδου που προορίζεται για τα μάτια.
Το νευρωνικό δίκτυο σε αυτήν την εκδοχή έδινε αρκετά καλά αποτελέσματα και παρόλο που ο διαχωρισμός του προσώπου σε επιπλέων περιοχές δείχνει μια αρκετά καλή ιδέα δεν έγινε περεταίρω ανάπτυξη του νευρωνικού δικτύου.

4.5 Διαδικασία Εκπαίδευσης

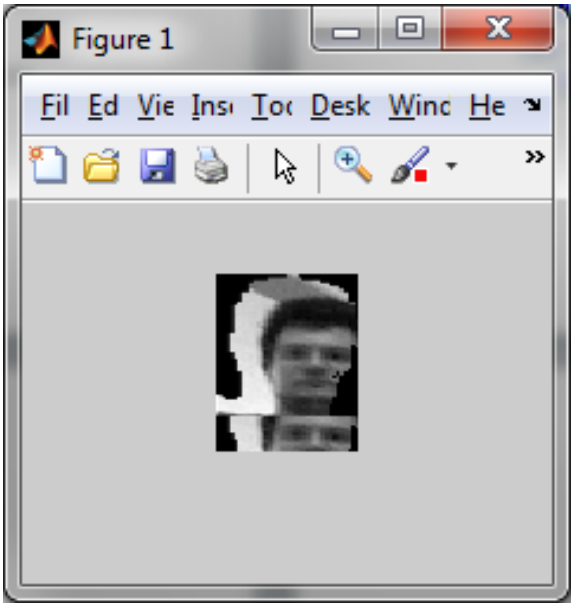
Προκειμένου να γίνει η εκπαίδευση του νευρωνικού δικτύου ένα σύνολο από δείγματα του κάθε ατόμου που θα πρέπει να είναι δυνατόν να ταυτοποιήθει. Τα άτομα αυτά κάθισαν μπροστά από το action pro και η τετράγωνη περιοχή που βρισκόταν γύρω από την περιοχή που εντοπίστηκε το κεφάλι τους αποθηκευόταν σε ένα αρχείο που θα μπορούσε να διαβάσει το matlab.Τα άτομα κάθισαν μπροστά από το action pro αρκετά ώστε να μαζευτούν πλήθος από καρέ και ο προσανατολισμός του κεφαλιού τους δεν ήταν σταθερός ώστε μετά την εκπαίδευση το νευρωνικό δίκτυο να έχει ανοχή στην διαφορετικότητα της κλίσης και της πόζας του κεφαλιού. Τα αποθηκευμένα δεδομένα δεν είχαν υποστεί καμιά προεπεξεργασία γιατί ήθελα να δοκιμάσω αρχικά στο matlab παραλλαγές στην προεπεξεργασία μέχρι να επιλέξω αυτήν που θα λειτουργούσε καλύτερα. Μετά έγινε η κατάλληλη αλλαγή κλίμακας στις εικόνες. Μετά με το κατάλληλο πρόγραμμα από ένα μέρος των εικόνων που τραβήχτηκαν επιλέχτηκαν οι περιοχές που περιελάμβαναν μάτια προκειμένου να εκπαιδευτούν τα νευρωνικά που θα εντοπίζουν τα μάτια. Παρακάτω παρουσιάζεται ο κώδικας που χρησιμοποιήθηκε:

```
for i=1:2400
   [i2 re]=imcrop(faces(:,:,i));
   simk=fix(re(2))-5
   if(simk>10)
      ypsosmation(:,:,mc)= faces (simk:(simk+9),:,i);
```

```
        if mod(mc,10)==1
            save('ypsosmation.mat','ypsosmation');
        end
        mc=mc+1;
        simiakopis=[simiakopis [simk;i]];
        faces (simk:(simk+9),:,i)=ones(10,60)*255;
    end
    imshow(faces (:,:,i));
    pause(0.8)
end
```

Η μεταβλητή simk περιέχει το ύψος από όπου ξεκινά το ορθογώνιο που περιέχει τα μάτια. Τα μάτια ποτέ δεν βρίσκονται σε ύψος μικρότερο του 10 και έτσι επέλεξα σαν εύκολο τρόπο απόρριψης των εικόνων που δεν είναι κατάλληλες για εκπαίδευση την επιλογή ύψους σε αυτή την περιοχή. Η επιλογή του ύψους γινόταν εύκολα χάρης στην συνάρτηση imcrop με ένα κλικ του ποντικιού στο ύψος των ματιών.Ο πινάκας faces περιέχει τα πρόσωπα και στον πίνακα ypsosmation βρίσκονται εικόνες διαστάσεων 10X60 με μάτια που θα χρησιμοποιηθούν για την εκπαίδευση του νευρωνικού που διακρίνει μάτια ή όχι. Στον πίνακα simiakopis βρίσκονται η τιμές ύψους για το που βρέθηκαν μάτια και χρησιμοποιούνται για το νευρωνικό που με είσοδο την συνολική εικόνα, στο οποίο ενεργοποιείται ένας νευρώνας, αυτός που αντιστοιχεί στην περιοχή που βρίσκονται τα μάτια.

Αφού εκπαιδευτούν τα νευρωνικά δίκτυα εντοπισμού ματιών ελέγχεται το σύνολο των εικόνων-προσώπων και ένας νέος πίνακας με εικόνες προσώπων δημιουργείται που έχει στο κάτω μέρος της εικόνας μια περιοχή 60X15 pixel που περιέχει την περιοχή γύρω από τα μάτια της συγκεκριμένης εικόνας. Έτσι ο νέος αυτός πίνακας περιέχει εικόνες 60X75.Η παρακάτω εικόνα αποτελεί ένα παράδειγμα:



Εικόνα 14 : Δείγμα επεξεργασμένης εικόνας

Μετά με ένα κατάλληλο πρόγραμμα σε matlab αφαίρεσα από το σύνολο των εικόνων τις εικόνες εκείνες που δεν ήταν καλές για δείγματα εκπαίδευσης. Στην συνέχεια επειδή τα δεδομένα πρέπει να δοθούν στο νευρωνικό σαν κάθε δείγμα να είναι πίνακας γραμμή και όχι δισδιάστατος, ο κάθε 60X75 υποπινακας μετατρέπεται σε ένα πίνακα 4500 θέσεων με τον παρακάτω κώδικα:

```
for i=1:size(faces,3)
    temp=[];
    for jy=0:3
        for jx=0:4
            for ky=1:15
                temp=[temp faces(ky+(jy*15),(1:15)+(jx*15),i)];
            end
```

```
        end
    end
    facesLined (:,i)=temp;
end
```

Όπως φαίνεται στον παραπάνω κώδικα η μετατροπή του πίνακα δεν γίνεται με μια απλή σάρωση του πίνακα αλλά γίνεται έτσι ώστε αν χωρίσουμε τον γραμμικό πίνακα σε περιοχές που η κάθε μια να έχει 225 στοιχεία αυτή να αντιστοιχεί σε ένα τετράγωνο 15X15 μέσα στην εικόνα. Αυτό γίνεται ώστε να μπορούν εύκολα να τροφοδοτηθούν οι κατάλληλες ομάδες νευρώνων στο κρυφό στρώμα από την σωστή περιοχή της εικόνας. Στο επόμενο στάδιο αφαιρείται από τον πίνακα το 127 και την μετέπειτα διαίρεση του με το 128.Μετα από αυτό με χρήση της παρακάτω εντολής:

networkFaceCl=train(net,tarS,facesLined)

Γίνεται η εκπαίδευση του νευρωνικού δικτύου.

Ο πίνακας tarS είναι ένας πίνακας 4X946 που κάθε μια από τις 946 γραμμές αντιστοιχεί σε μια από τις εικόνες του facesLined. Κάθε μια από τις 4 στήλες του αντιστοιχεί σε διαφορετικό υποψήφιο για κάτοχο του προσώπου, με τιμή -1 εάν δεν απεικονίζεται αυτός και 1 εάν απεικονίζεται.

Το νευρωνικό δίκτυο μπορεί να εκπαιδεύτηκε στο περιβάλλον του matlab αλλά η χρήση του γενικά θα γίνεται στο ρομπότ σε περιβάλλον c++.Έτσι προκειμένου να χρησιμοποιηθεί παράγεται ένα δυαδικό αρχείο που περιέχει τις πληροφορίες του νευρωνικού όπως δομή και βάρη και θα μπορεί να χρησιμοποιηθεί από τον κώδικα που έχω φτιάξει σε c++ . Για αυτόν τον σκοπό έφτιαξα την συνάρτηση SaveNetBin(net,file). Στο αρχείο καταγράφονται με την σειρά που αναφέρονται :το πλήθος των ομαδόν νευρώνων που υπάρχουν, πόσους νευρώνες έχει η κάθε μια, ποιες έχουν είσοδο πόλωσης, μετά ποιες ομάδες συνδέονται με ποιες αριθμός τον ομάδων νευρώνων εισόδου και πόσοι υπάρχουν σε κάθε ομάδα, ποιες ομάδες νευρώνων επεξεργασίας συνδέονται με ποιες εισόδους, τα βάρη πόλωσης, τα βάρη συνάψεων με τους νευρώνες εισόδου και τα βάρη ανάμεσα στους υπόλοιπους νευρώνες.

4.6 Λεπτομέρειες υλοποίησης

4.6.1 Η υλοποίηση του νευρωνικού

Ο κώδικας που υλοποιεί το νευρωνικό δίκτυο στην πλευρά του οχήματος είναι σχεδιασμένος ώστε να μην χρειάζεται τροποποίηση σε περίπτωση που αλλάξει η δομή του νευρωνικού δικτύου που κατασκευάζει το matlab. Επίσης καλό θα ήταν αφού υπάρχουν συναρτήσεις μέσα στο πρόγραμμα με νόημα μονό για το νευρωνικό δίκτυο να εμφωλεύουν όλα αυτά σε μια κλάση. Για να γίνει αυτό μια κλάση νευρωνικών δικτύων δημιουργείται που περιέχει όλες τις απαραίτητες πληροφορίες και μεθόδους για το νευρωνικό δίκτυο. Ο κώδικας για την δημιουργία της παρουσιάζεται παρακάτω:

```
class net_mat
{
 //arithmos layer
 int nl;
 //arithmos isodon
 int ni;
 //layer eksodou
 int outputl;
 //arithmos neyrono ana layer,megethos isodon
 int *nol,*isi;
 //sindeseis metaksi leyer,sinesis isodon me layer
 int **lcl,**icl;
 //bari
 double ***iw,***lw;
 //bias,layer outputs
 double **b,**lout;
public:
 int outSize()
 { return nol[outputl];}
 #include "sim_net.h"
 #include "loadnetfcf.h"
};
```

Επειδή η δομή και το μέγεθος του δικτύου είναι άγνωστα εκ των πρότερων η πλειοψηφία των δεδομένων

είναι προσπελάσιμη μέσω δεικτών. Μια πληροφορία που δεν καταγράφεται σε αυτήν την δομή είναι η συνάρτηση μεταφοράς των νευρώνων του κάθε στρώματος και αυτό γίνεται γιατί σε όλες τις περιπτώσεις χρησιμοποίησα την συνάρτηση tansig.Στην μεταβλητή nl καταχωρείται ο αριθμός των ομάδων από νευρώνες που περιέχονται στο νευρωνικό. Στην outputl καταχωρείται πια από όλες αυτές τις ομάδες είναι η ομάδα εξόδου. Στον τρόπο με τον οποίο το matlab υλοποιεί τα νευρωνικά δίκτυα το σύνολο των δεδομένων εισόδου είναι δυνατό να χωριστεί σε ομάδες, επιτρέποντας έτσι την σύνδεση του κάθε ενός από τους νευρώνες στης κατάλληλες εισόδους. Ο αριθμός αυτών τον ομάδων καταχωρείται στην μεταβλητή ni.Στον πίνακα που δείχνει ο δείκτης nol καταχωρείται ο αριθμός των νευρώνων που υπάρχει σε κάθε μια από τις ομάδες νευρώνων. Στον πίνακα που δείχνει δείκτης isi καταχωρείται ο αριθμός των εισόδων που υπάρχει σε κάθε μια από τις ομάδες που έχει χωριστεί η είσοδος. Στον δισδιάστατο πίνακα που δείχνει δείκτης lcl καταχωρούνται οι συνδέσεις μεταξύ των ομάδων των νευρώνων. Δηλαδή εάν στην γραμμή i και την στήλη j υπάρχει η τιμή 1 τότε η ομάδα i αποτελεί είσοδο για την ομάδα j.Με τον ίδιο τρόπο στον πίνακα του δείκτη icl καταχωρούνται οι συνδέσεις μεταξύ των ομάδων εισόδων και των ομάδων νευρώνων. Στον τρισδιάστατο πίνακα iw καταχωρούνται τα βάρη μεταξύ των ομάδων νευρώνων και στον lw μεταξύ εισόδων και νευρώνων. Στον στο b τα βάρη πόλωσης. Στο lout δεν καταχωρείται κάτι στην φάση της φόρτωσης αλλά στην φάση της εξομοίωσης μπαίνουν οι τιμές εξόδου τον νευρώνων. Η παρακάτω μέθοδος φορτώνει το νευρωνικό δίκτυο:

int loadnetfmcfile(char *fileName)

Η κλήση της μεθόδου επιστρέφει -1 αν κάτι δεν πάει καλά με το άνοιγμα του αρχείου. Η παρακάτω μέθοδος κάνει την εξομοίωση του νευρωνικού δικτύου:

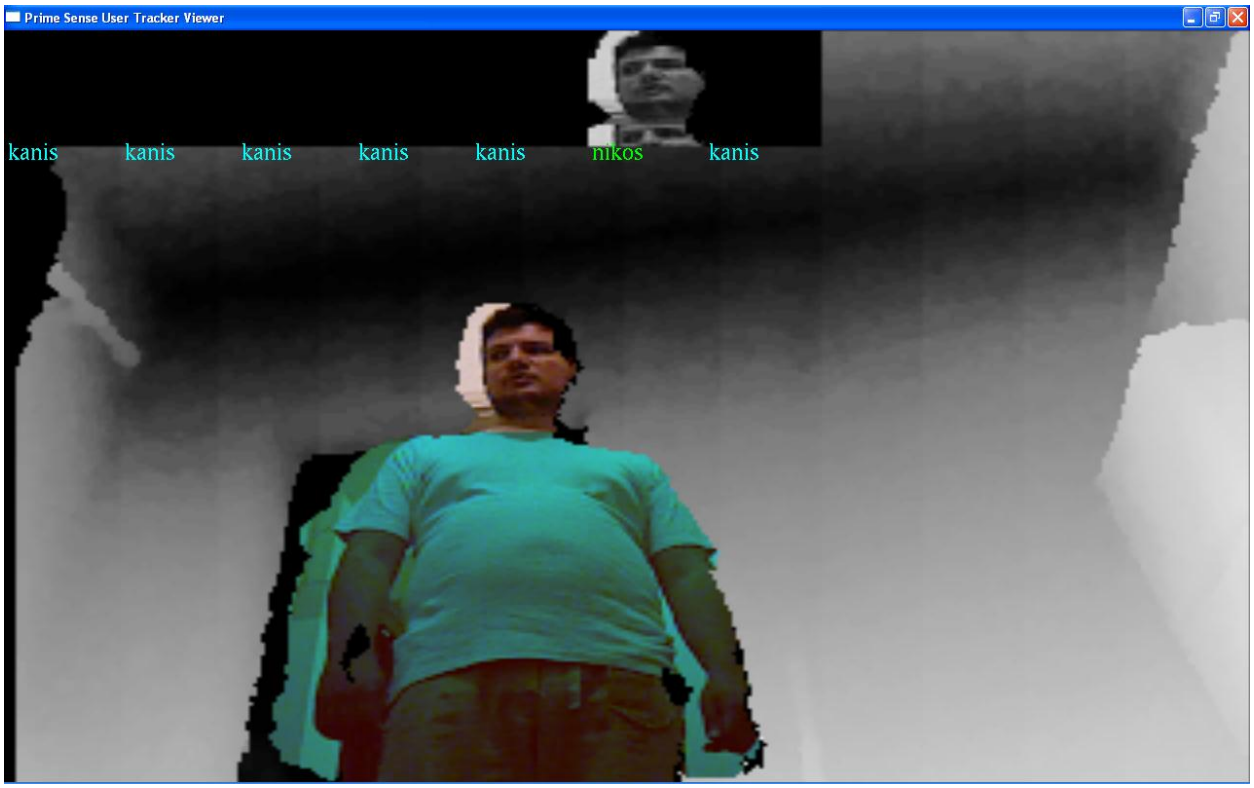
double *netSim(double *eis)

Αφού ο κώδικας είναι γενικός και δεν είναι εκ τον προτέρων γνωστό το μέγεθος της εισόδου, η συνάρτηση έχει πρόσβαση στην είσοδο μέσω του δείκτη eis.Μετά την εκτέλεση της συνάρτησης ένας δείκτης προς τον πίνακα που βρίσκεται η έξοδος του νευρωνικού επιστρέφεται.

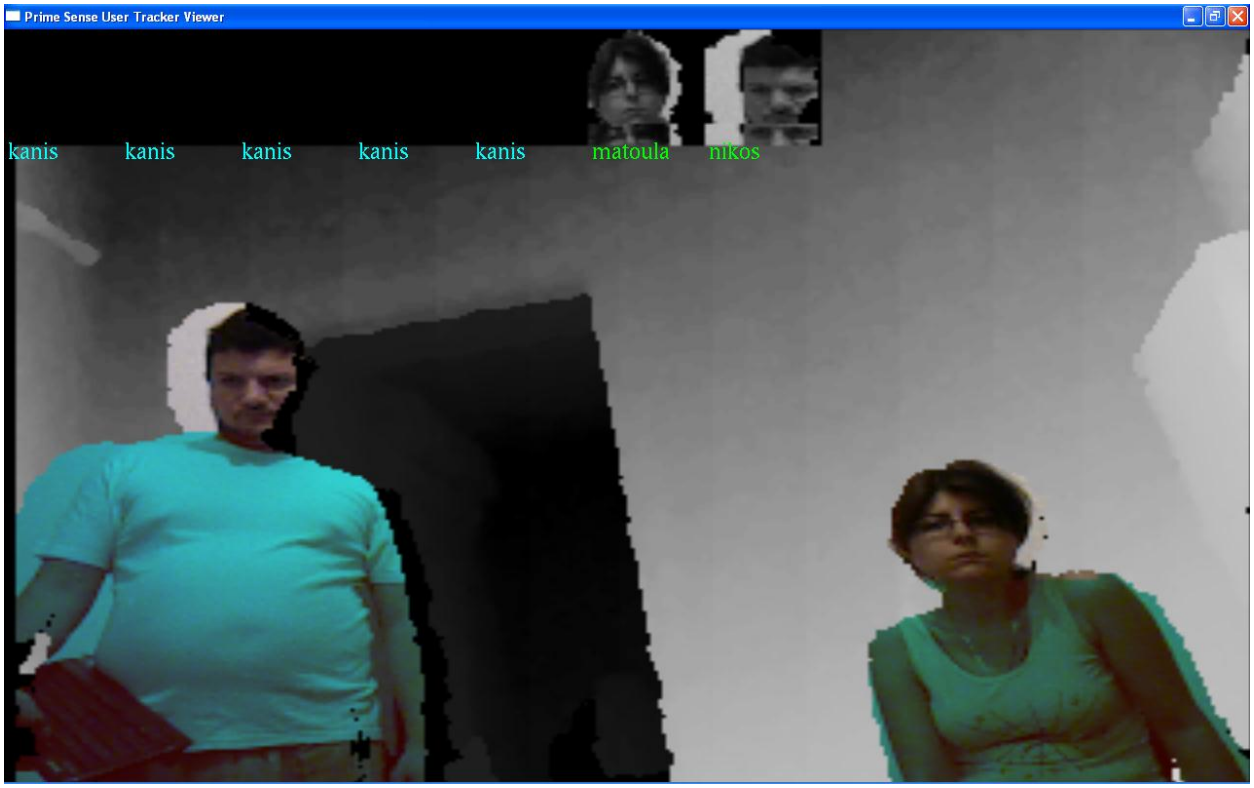
Η συνάρτηση *int outSize()* επιστρέφει το μέγεθος της εξόδου προκειμένου να είναι δυνατή η χρήση του δείκτη.

4.6.2 Η γενικότερη λειτουργιά

Ο μηχανισμός του user tracking θα εντοπίσει έναν ή περισσότερους χρήστες. Σε κάποιους από αυτούς το skeleton tracking θα εντοπίσει το κεφάλι. Στο πρόγραμμα υπάρχει ο πίνακας portret τεσσάρων διαστάσεων, που οι τρεις πρώτες διαστάσεις αφορούν την εικόνα(x,y,χρώμα) και η τέταρτη τον χρήστη. Σε αυτόν θα αντιγράφει ένα τετράγωνο με κέντρο το κεφάλι από κάθε έναν χρήστη που έχει εντοπιστεί το κεφάλι του. Στις υπόλοιπες θέσεις του πίνακα εγγράφονται τα μηδενικά. Με βάση τα πρόσωπα που θα βρεθούν ο αλγόριθμος προεπεξεργασίας θα εγγράψει στον πίνακα scaledPortret προεπεξεργαζμένες εκδοχές τον προσώπων. Μετά θα μετατραπούν τα δεδομένα αυτού του πίνακα για κάθε πρόσωπο σε γραμμικά και θα δοθούν σαν είσοδος στα νευρωνικά. Μετά με βάση τις εξόδους των νευρωνικών και έναν πίνακα αντιστοιχίσεων που υπάρχει στο πρόγραμμα θα αντιστοιχηθεί σε κάθε πρόσωπο και μια ταυτότητα(άγνωστος είναι μια πιθανότητα). Στην εικόνα 15 φαίνεται ένα παράδειγμα από εκτέλεση του προγράμματος με επιτυχή ταυτοποίηση ενός ατομου και στην 16 ταυτοχρονη ταυτοποιηση 2 ατομων.



Εικόνα 15 : Ταυτοποίηση ενός ατόμου



Εικόνα 16 : Ταυτόχρονη ταυτοποίηση δυο ατόμων

Στην υλοποίηση απεικονίζονται οι επτά πρώτες εικόνες που βρίσκονται στον πίνακα scaledPortret έτσι ώστε να είναι δυνατή η εξακρίβωση της ταυτοποίησης πολλών ατόμων ταυτόχρονα. Για κάθε χρήστη υπάρχει μια τιμή που διατηρεί πόσες φορές συνεχόμενα ταυτοποιείται το ίδιο άτομο με βάση το πρόσωπο. Όταν η τιμή αυτή ξεπεράσει ένα κατώφλι θεωρείται ότι ο η αναγνώριση ήταν επιτυχής και όχι ένα παροδικό σφάλμα. Από αυτό το σημείο και μετά ο χρήστης θεωρείται ταυτοποιημένος.

Μέρος δεύτερο: αναγνώριση μέσω ρούχων και άλλων χαρακτηριστικών

4.8 Διατήρηση της ταυτοποίησης χρήστη μετά από τυχών προσωρινή απώλειας οπτικής επαφής

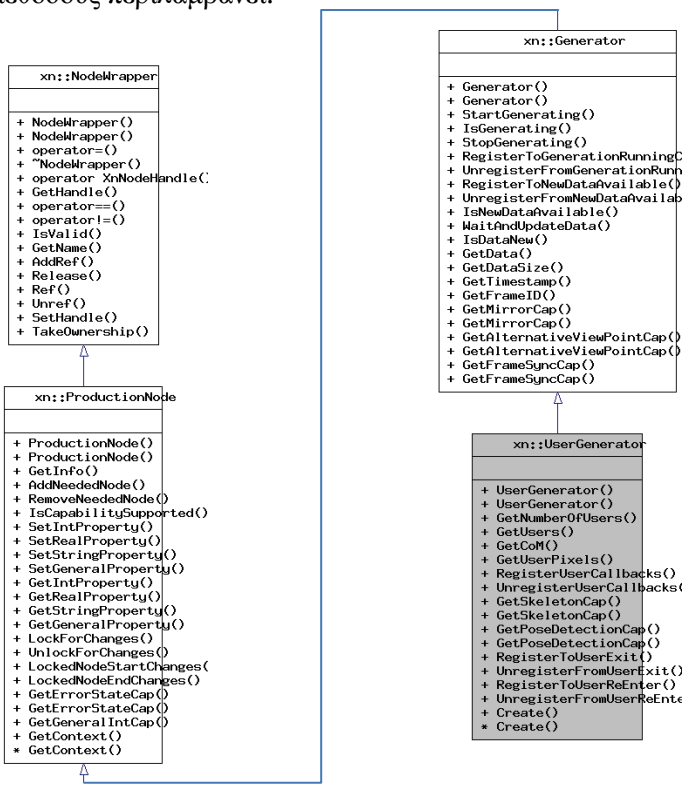
Η ανάγκη για ταυτοποίηση χωρίς την αναγνώριση του προσώπου προκύπτει εάν ενώ το όχημα ακολουθεί τον χρήστη υπάρξει προσωρινή απώλεια του εντοπισμού του χρήστη. Όσο το user tracking κάνει track τον χρήστη δεν υπάρχει ανάγκη για επιπλέον ταυτοποίηση του χρήστη ακόμα και αν αυτός, που το μεγαλύτερο μέρος της πορείας αυτό συμβαίνει, έχει γυρισμένη την πλάτη. Εάν τώρα ενώ ο χρήστης έχει γυρισμένη την πλάτη συμβεί προσωρινή απώλεια οπτικής επαφής θα διακοπεί η πορεία του ρομπότ προς τον χρήστη του και ο χρήστης θα πρέπει να γυρίσει προς το ρομπότ για εκ νέου ταυτοποίηση. Επίσης απώλεια εντοπισμού μπορεί να προκύψει από αστοχία του μηχανισμού του user tracking , που μπορεί να ευνοηθεί από μια απότομη κίνηση του χρήστη εμπρός αλλά δεν παρουσιάζεται μόνον σε αυτήν την περίπτωση. Έτσι θα πρέπει να υπάρξει ένας επιπλέων μηχανισμός που θα εξασφαλίζει την επαναταυτοποίηση μετά την απώλεια του εντοπισμού. Για τον σκοπό αυτό διάφορα χαρακτηριστικά του χρήστη ακατάλληλα για μακροχρόνια ταυτοποίηση αλλά κατάλληλα στα πλαίσια μιας χρήσης του οχήματος θα πρέπει να ληφθούν.

4.9 Τα χαρακτηριστικά ταυτοποίησης και η χρήση αυτών

Για την ταυτοποίηση επέλεξα την απόσταση μεταξύ των ώμων, την απόσταση μεταξύ ώμου και αγκώνα ,το ιστόγραμμα χρώματος σε μια μικρή περιοχή γύρο από τις συντεταγμένες των ώμων και τον αγκώνων. Όσον αφορά το ιστόγραμμα χρώματος αυτό μια δυνατή επιλογή θα ήταν η παράμετρος hue από το μοντέλο HSV. Παρόλο που μια τέτοια επιλογή θα ήταν καλή αν τα ρούχα του χρήστη έχουν έντονα χρώματα , δεν βοηθά καθόλου στην ταυτοποίηση αν το χρώμα των ρούχων βρίσκεται στην κλίμακα του γκρι. Η λύση που επέλεξα είναι να διαχωρίζονται τα ρούχα σε έγχρωμα και άχρωμα. Για τον διαχωρισμό αυτό υπολογίζεται η μέση τιμή του κορεσμού και μετά βάση κατωφλιού χαρακτηρίζεται ανάλογα. Έτσι ο πρώτος έλεγχος που γίνεται είναι αν το ρούχο είναι έγχρωμο ή μη και αν διαφέρει από αυτό της ταυτοποίησης γίνεται άμεση απόρριψη. Αν τα ρούχα ανήκουν στην ίδια κατηγορία με του χρήστη υπολογίζεται το ιστόγραμμα χρώματος ή φωτεινότητας. Εξισορρόπηση ιστογράμματος δεν γίνεται μιας και όσον αφορά το ιστόγραμμα χρώματος αυτό ούτος ή άλλος δεν έχει νόημα. Επίσης όσον αφορά το ιστόγραμμα φωτεινότητας αυτό είναι αρκετά πιθανό να είναι επικεντρωμένο σε μια απόχρωση του γκρι που αυτό είναι μια πληροφορία που δεν πρέπει να χαθεί (μην ξεχνάμε ότι δεν έχουμε να κάνουμε με μια πλήρη εικόνα αλλά για μια περιοχή των ρούχων γύρο από τους ώμους). Το ιστόγραμμα μετά υποδειγματολιπτειται και γίνεται σύγκριση με αυτό. Η υποδειγματολειψια γίνεται ανά 15 και η τιμή δεν προκύπτει μόνο από την 15 τιμή για παράδειγμα αλλά από τον μέσο όρο τον 15 μικρότερων και μεγαλύτερο τιμών. Μετά γίνεται σύγκριση των ιστογραμμάτων. Όσον αφορά την σύγκριση των ιστογραμμάτων υπάρχουν πολλές διαφορετικές επιλογές από τις οποίες δεν είναι ευδιάκριτο πια είναι καλύτερη. Συμφώνα με το [14] μια αρκετά επιτυχημένη μετρική είναι Jeffrey την οποία και χρησιμοποιώ. Εάν έχουμε δυο ιστογράμματα H και K , με h_i και k_i τις τιμές στις διάφορες θέσεις του ιστογραμμάτων τότε η Jeffrey divergence d(H,K) ορίζεται ως εξής: $d(H,K)= \sum_i \left(h_i \log \frac{2h_i}{h_i+k_i} + k_i \log \frac{2k_i}{h_i+k_i} \right)$ (εξ.7.1). Μικρότερη τιμή σημαίνει και μεγαλύτερη ομοιότητα. Επίσης το μέτρο της διαφοράς μεταξύ τον διαστάσεων των άκρων του προτύπου και του χρήστη προς ταυτοποίηση όσο μικρότερη τιμή έχει σημαίνει μεγαλύτερη ομοιότητα, και με βάση αυτό γίνεται η σύγκριση των χαρακτηριστικών αποστάσεων του σώματος. Όλα τα παραπάνω αθροίζονται με τους κατάλληλους συντελεστές βαρύτητας που έχουν επιλεχθεί εμπειρικά. Το άθροισμα αυτό αν έχει τιμή μικρότερη ενός κατωφλιού σημαίνει και επαναταυτοποίηση.

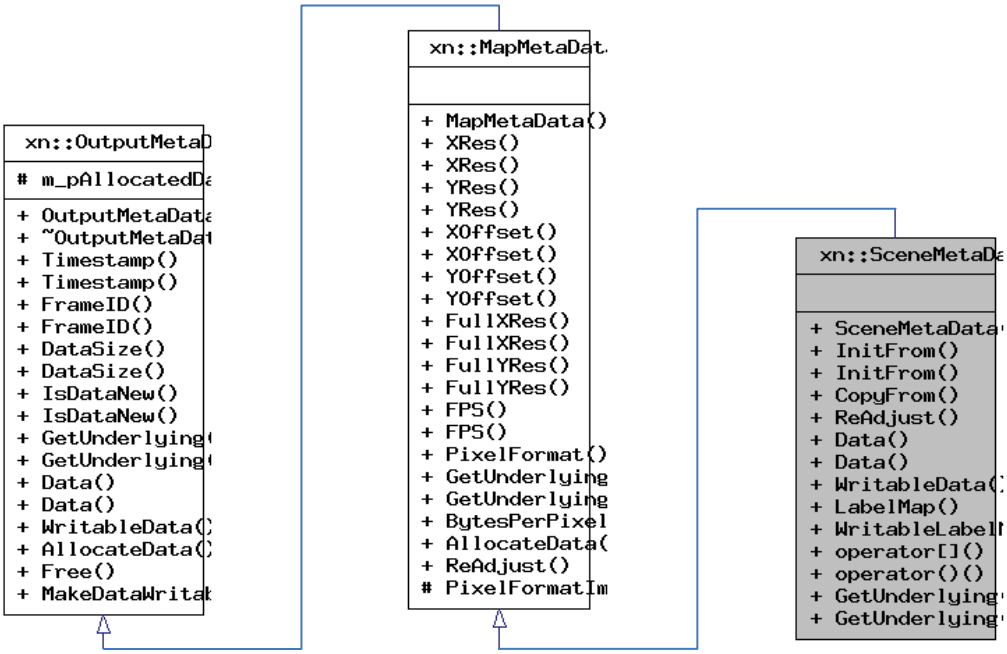
5.1 Εντοπισμός χρηστή μέσω user tracking

Το open-ni παρέχει έναν εύκολο τρόπο για τον εντοπισμό ενός χρήστη στο οπτικό πεδίο του action pro. Ο τρόπος αυτός είναι ο μηχανισμός user tracking ο οπαίος υλοποιείται μέσω της κλάσης UserGenerator. Στο διάγραμμα 7 φαίνεται σε μορφή uml πως συνδέεται μέσω κληρονομικότητας με άλλες κλάσεις , και ποιές μεθόδους περιλαμβάνει.

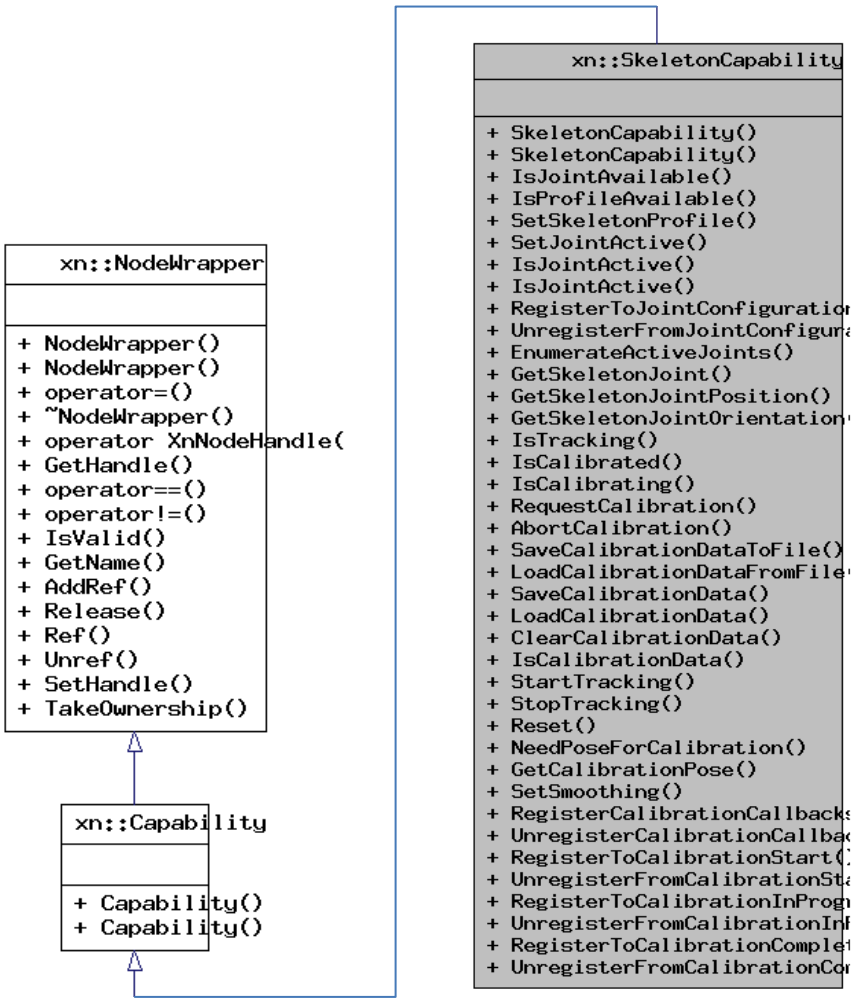


Διάγραμμα 7 : UML Της κλάσης UserGenerator

Στην εφαρμογή μου δημιουργώ και χρησιμοποιώ μόνο ένα αντικείμενο αυτής της κλάσης το g_UserGenerator. Από τις διαθέσιμες μεθόδους χρησιμοποιώ τις παρακάτω: IsCapabilitySupported ,Create ,RegisterUserCallbacks ,GetSkeletonCap ,GetPoseDetectionCap ,GetUserPixels ,GetUsers και GetCoM. Η IsCapabilitySupported έχει κληρονομηθεί από την ProductionNode και κάνει έλεγχο αν είναι δυνατόν να υποστηριχτεί η ζητούμενη δυνατότητα. Η create δημιουργεί το αντικείμενο. Η GetPoseDetectionCap επιστρέφει ένα αντικείμενο με πληροφορίες για την στάση του χρήστη. Η μέθοδος GetUserPixels μπορεί να ενημερώσει τις πληροφορίες ενός αντικειμένου τύπου SceneMetaData έτσι ώστε να περιέχει μια ταξινόμηση των pixel της εικόνας σε αυτά που ανήκουν στον χρήστη και σε αυτά που όχι. Στο διάγραμμα 8 φαίνεται σε μορφή UML η σχέση του αντικειμένου SceneMetaData με άλλα όπως επίσης και οι μέθοδοι που περιλαμβάνει. Η μέθοδος GetUsers καταχωρεί σε έναν πίνακα τους χρήστες που έχουν εντοπιστεί καθώς και τον αριθμό τους. Η μέθοδος GetCoM καταχωρεί σε μια δομή τύπου XnPoind3D τις συντεταγμένες (X,Y,Z) του κέντρου βάρους του χρήστη. Η μέθοδος GetSkeletonCap επιστρέφει ένα αντικείμενο τύπου SkeletonCapability μέσω του οποίου μπορούν να εντοπιστούν τα μέλη του σώματος και να ευρεθούν οι συντεταγμένες αυτών. Στο διάγραμμα 9 φαίνεται η περιγραφή της μέσω UML. Ουσιαστικά εάν δημιουργηθεί ένα αντικείμενο τύπου UserGenerator και μετά εντοπιστεί ο χρήστης μπορούμε εύκολα, με την χρήση της GetCoM να πάρουμε πληροφορίες για την θέση του χρήστη. Αυτές από μόνες τους είναι, εάν δεν υπάρχουν στο ενδιάμεσο εμπόδια και ο εντοπισμός είναι σωστός, αρκετές για την παρακολούθηση του χρήστη.



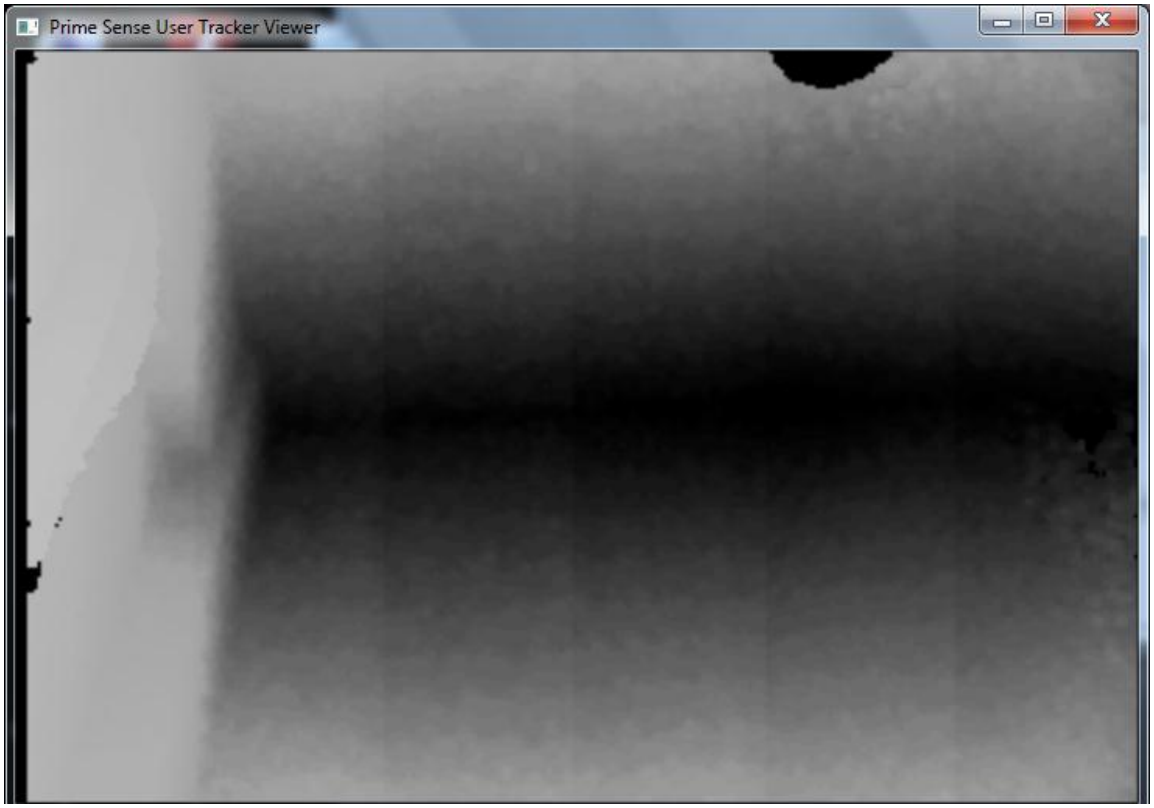
Διάγραμμα 8 : UML Της κλάσης SceneMetaData



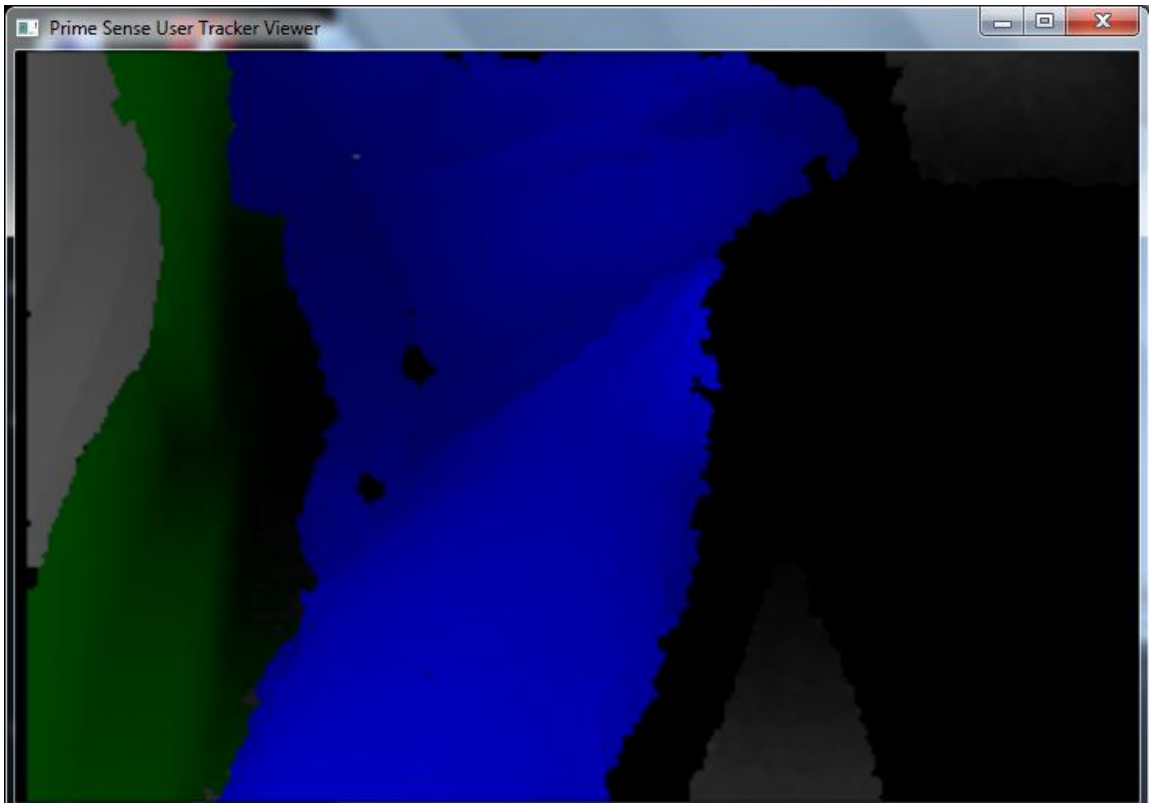
Διάγραφα 9 : UML Της κλάσης SkeletonCapability

5.2 Προβλήματα εντοπισμού κατά την κίνηση του ρομποτικού

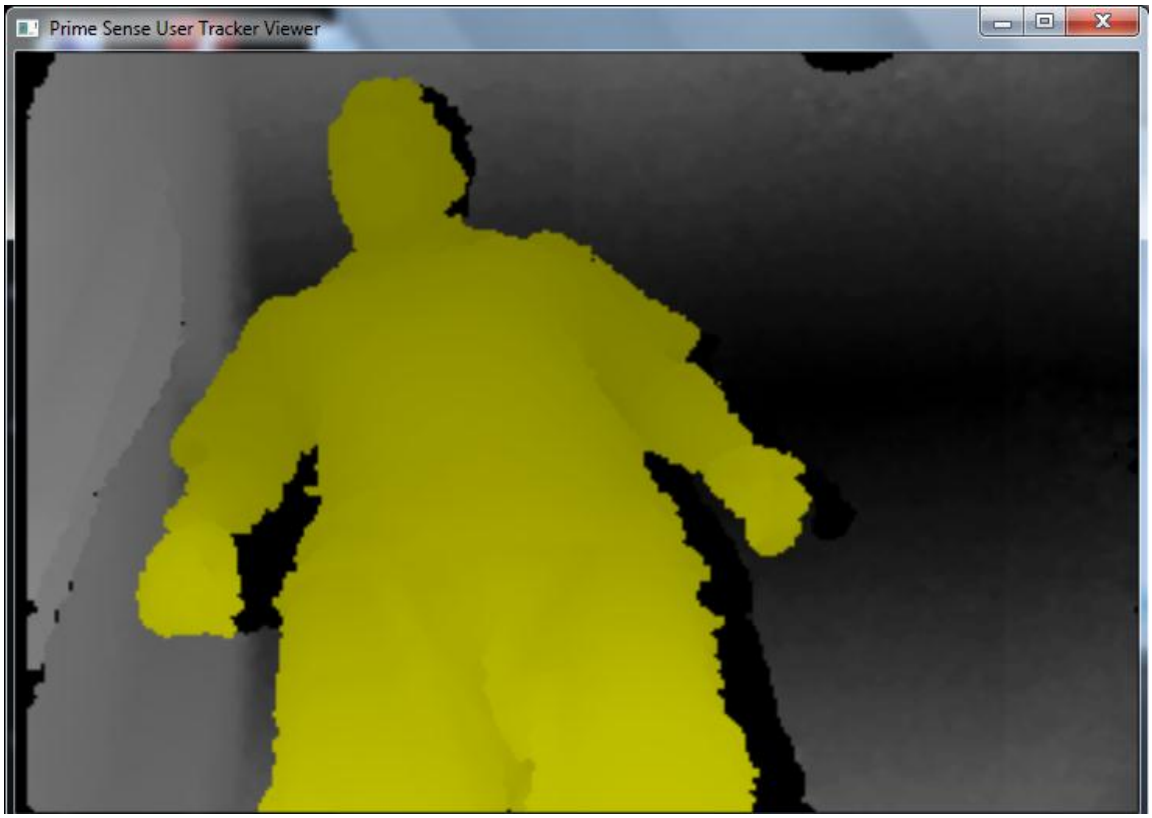
Το user tracking έχει σχεδιαστεί με βάση την υπόθεση ότι οι χρήστες είναι ακίνητοι και τα έπιπλα, το μπακράουντ και όλα τα υπόλοιπα αντικείμενα ακίνητα. Όταν όμως το ρομπότ αρχίσει να κινείται αυτή η υπόθεση παύει να ισχύει. Κατά την κίνηση του οχήματος διάφορα αντικείμενα εντοπίζονται εσφαλμένα σαν άνθρωποι. Συνήθως αντικείμενα που έχουν ψηλό και στενό σχήμα όπως ‘’καλόγεροι’’ είναι αυτά που εντοπίζονται εσφαλμένα αλλά τα σφάλματα δεν περιορίζονται σε αυτά. Διάφορα πιο ογκώδη αντικείμενα τυχαίνει να εντοπιστούν εσφαλμένα αλλά και τοίχοι κάποιες φορές. Στην εικόνα 18 φαίνεται ένα παράδειγμα εσφαλμένου εντοπισμού. Τα δυο χρώματα (πράσινο ,μπλε)της εικόνας 18 αντιστοιχούν σε δυο υποτιθέμενους χρήστες. Επίσης από ότι φαίνεται δεν μοιάζει κανένα από τα δυο ιδιαίτερα με άνθρωπο. Στην εικόνα 19 από την άλλη παρουσιάζεται ένας ορθός εντοπισμός.



Εικόνα 17 : Δείγμα χωρίς εντοπισμό



Εικόνα 18 : Εσφαλμένος εντοπισμός



Εικόνα 19 : Σωστός εντοπισμός

Έτσι ένας επιπλέων μηχανισμός που θα απορρίπτει τους εσφαλμένους εντοπισμούς είναι απαραίτητος. Ευτυχώς δεν γίνεται μια διαρκής παρερμηνεία των αντικειμένων όταν το ρομπότ βρίσκεται σε κίνηση. Έτσι ο μηχανισμός αυτός δεν χρειάζεται να διαχωρίζει από ένα πλήθος εντοπισμών που η μεγάλη πλειοψηφία είναι εσφαλμένη. Ένας απλός τρόπος είναι η απόρριψη των εντοπισμών που δίνουν ύψος

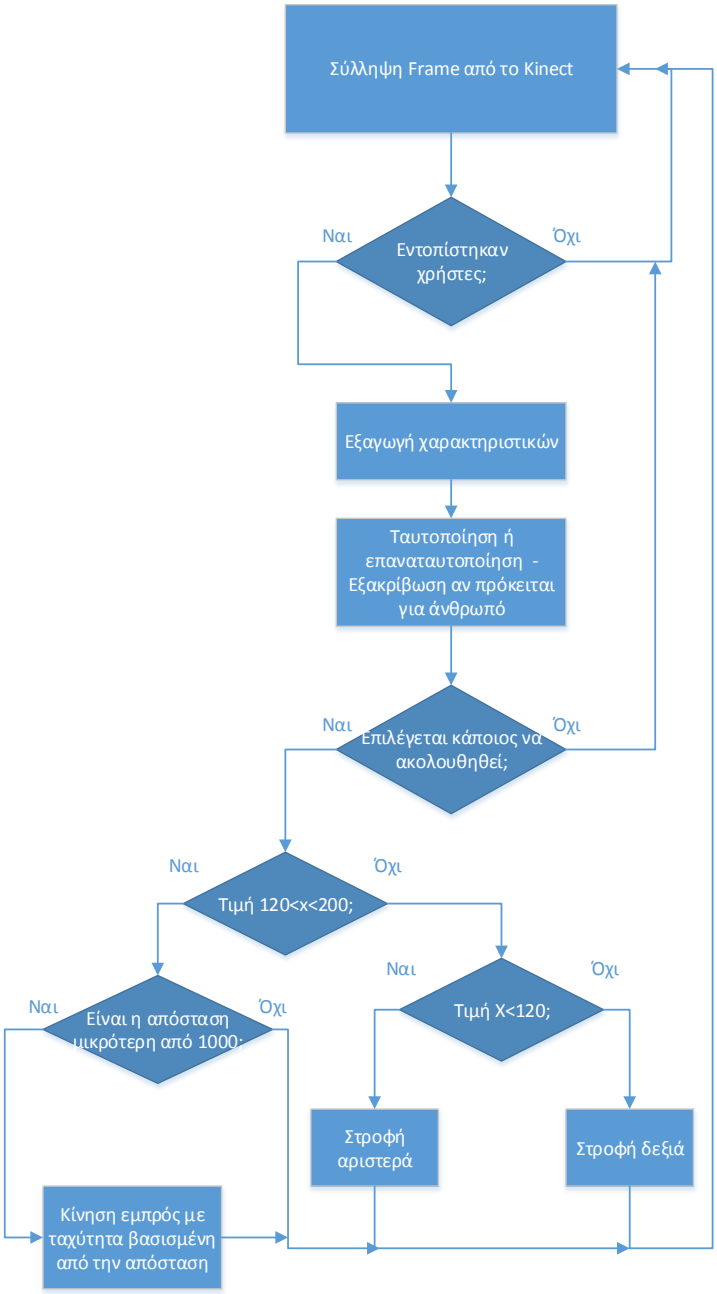
κέντρου βάρους του χρήστη παράλογα ψηλά ή χαμηλά. Αυτό από μόνο του δεν είναι αρκετό και μια πιο σύνθετη μέθοδος απαιτείται.

5.3 Τρόπος αντιμετώπισης αυτών μέσω νευρωνικών δικτύων

Για να επιλύσω αποτελεσματικά το πρόβλημα αυτό επέλεξα την χρήση νευρωνικών δικτύων. Για να γίνει όμως αυτό πρέπει πρώτα να εξαχθούν από τα δεδομένα που δίνει το user tracking κάποιον πιο συνοπτικών δεδομένων. Το user tracking δίνει έναν χάρτη των pixel που η τιμή του κάθε ενός αντιστοιχεί στον χρήστη που αντιστοιχεί στον χρήστη που έχει εντοπιστεί. Σε pixel που δεν αντιστοιχούν σε χρήστη η τιμή είναι μηδέν. Τα δεδομένα που θα εξαχθούν από τον χάρτη αυτόν καλό είναι να μην μεταβάλλονται ιδιαίτερα καθώς ο χρήστης απομακρύνεται από το όχημα ούτε από του που βρίσκετε στον οριζόντιο άξονα ο χρήστης. Ο ένας τύπος δεδομένων που εξάγεται είναι το άθροισμα των pixel που ανήκουν στον χρήστη που διερευνάται αν είναι όντος κάποιος. Ουσιαστικά αυτό είναι το πάχος του χρήστη στην κάθε γραμμή και δεν αλλάζει σημαντικά καθώς το αντικείμενο κινείται οριζόντια. Η δεύτερη πληροφορία που συλλέγεται είναι το πλήθος των εναλλαγών. Σαρώνοντας τον πίνακα γραμμή γραμμή κάθε φορά που μπαίνουμε ή βγαίνουμε από μια περιοχή που ανήκει στον χρήστη αυξάνουμε κατά 1.Αυτη η τιμή δεν αλλάζει για μικρή μετακίνηση του αντικείμενου στον άξονα x ή z.Επίσης οι παραπάνω διαδικασίες πραγματοποιούνται με υποδιγματοληψία. Μόνον 16 γραμμές σαρώνονται και μόνο 16 pixel από την κάθε γραμμή. Έτσι έχοντας δυο τιμές για κάθε γραμμή, το πάχος και τις εναλλαγές προκύπτουν 32 τιμές. Επιπλέων στο νευρωνικό δίνονται οι πληροφορίες της θέσης του αντικειμένου δηλαδή θέση στον άξονα x,y,z. Το νευρωνικό έτσι τροφοδοτείται με 35 τιμές. Μια επιπλέων προεπεξεργασία πραγματοποιείται στις τιμές αυτές προκειμένου να είναι ιδίας τάξης μεγέθους. Το νευρωνικό στο κρυφό του στρώμα έχει 85 νευρώνες και στην έξοδο δυο. Ο ένας είναι ενεργός όταν εκτιμάται ότι αυτό που έχει εντοπιστεί είναι άνθρωπος ο άλλος σε αντίθετη περίπτωση. Οι δυο αυτές τιμές συγκρίνονται και αν η διαφορά είναι αρκετή ο εντοπισμός που έκανε το user tracking θεωρείται ορθός. Η πληροφορία αυτή καταγράφεται και λαμβάνεται υπόψη από το σύστημα πλοήγησης μαζί με την ταυτοποίηση και κάποιες άλλες για να αποφασιστεί αν το ρομπότ θα χαράξει πορεία προς τον χρήστη.

6.1 Η κίνηση του ρομπότ όταν υπάρχει οπτική επαφή χωρίς εμπόδια

Σε περίπτωση που η αναγκαία κίνηση του οχήματος είναι απλός η κίνηση προς τις συντεταγμένες του στόχου ,πράγμα που συμβαίνει όταν δεν υπάρχουν καθόλου εμπόδια ανάμεσα στο όχημα και τον χρήστη, χρησιμοποιείται ο αλγόριθμος που ακολουθεί. Εάν οι συντεταγμένες στον άξονα x είναι μεγαλύτερη του 200 το όχημα περιστρέφεται δεξιά με σκοπό να φέρει στο κέντρο του πάχους 320 pixel frame τον άνθρωπο που ακολουθεί. Εάν η τιμή του x είναι μικρότερη του 120 περιστρέφεται αριστερά. Εάν το x είναι στην περιοχή 120-200 τότε εάν το z (η απόσταση δηλαδή του χρήστη) είναι μικρότερη από 1000 το ρομπότ μένει ακίνητο. Εάν το z είναι μικρότερο από 1400 η ισχύς των μοτέρ μειώνεται ανάλογα με την διαφορά του z από το 1000. Για μεγαλύτερες τιμές του z η ισχύς είναι η μέγιστη. Εάν το x είναι στην περιοχή 120-200 η ισορροπία ισχύος μεταξύ τον μοτέρ επηρεάζεται με βάση τα παρακάτω. Εάν το x είναι 160 η ισχύς είναι ίση, εάν το x είναι μεγαλύτερο του 160 το αριστερό μοτέρ παίρνει περισσότερη ισχύ όσο μεγαλύτερο είναι το x και αν το x είναι μικρότερο το δεξιό μοτέρ παίρνει περισσότερη ισχύ με βάση τον ίδιο κανόνα. Στο παρακάτω διάγραμμα παρουσιάζεται σχηματικά η παραπάνω λειτουργιά:



Διάγραμμα 9

6.2 Κινηματικό μοντέλο του οχήματος

Στην προηγούμενη ενότητα παρουσιάστηκε η κίνηση του ρομπότ όταν δεν υπάρχουν καθόλου εμπόδια στον δρόμο του και ο χρήστης βρίσκεται στο οπτικό του πεδίο. Σε πιο απαιτητικές καταστάσεις όμως τόσο απλές τεχνικές δεν μπορούν να αποδώσουν. Για την εφαρμογή πιο περίπλοκων τεχνικών είναι απαραίτητο ένα μοντέλο της θέσης του οχήματος. Η θέση ενός οχήματος μπορεί να παρασταθεί με το παρακάτω διάνυσμα $p=[x\ y\ \theta]^T$.Στο οποίο το x,y είναι οι συντεταγμένες και θ ο προσανατολισμός του οχήματος. Σε ένα διαφορικά κινούμενο όχημα (όπως το δικό μου) οι μεταβολές των στοιχείων του διανύσματος(Δx,Δy,Δθ) μπορούν να υπολογιστούν, αν υποθέσουμε ότι η πρόσφυση είναι ιδανική, με βάση τους παρακάτω τύπους:

$\Delta x=\Delta s \cos(\theta+\Delta\theta/2)$ (6.1)

$\Delta y=\Delta s \sin(\theta+\Delta\theta/2)$ (6.2)

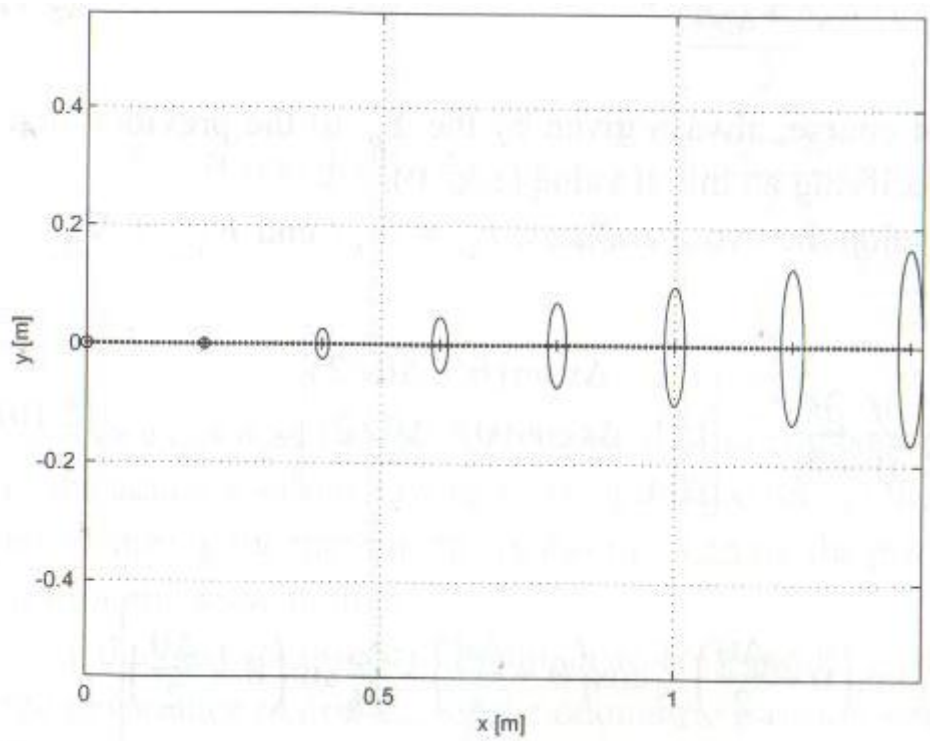
$\Delta\theta=(\Delta s_r-\Delta s_l)/b$ (6.3)

$\Delta s=(\Delta s_r+\Delta s_l)/2$ (6.4)

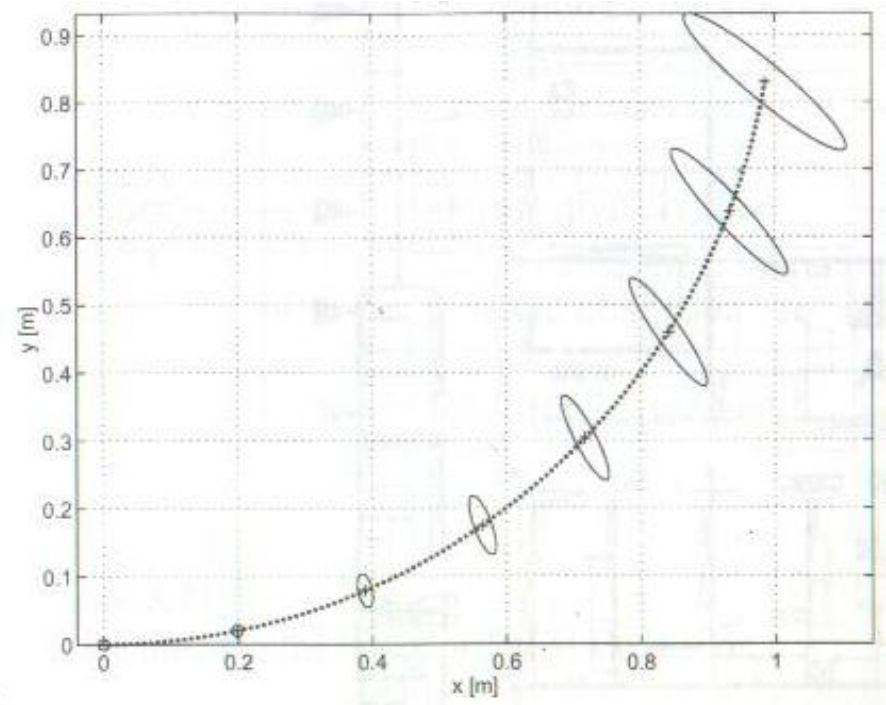
Στους παραπάνω τύπους το b είναι η απόσταση ανάμεσα στους τροχούς ή τους ιμάντες του οχήματος, Δs_l η μετακίνηση του αριστερού τροχού ή ιμάντα και Δs_r του δεξιού. Αν στους τροχούς του οχήματος υπάρχει ενσωματωμένο σύστημα οδομετρίας θα μπορούμε να έχουμε τις τιμές των Δs_l και Δs_r. Έτσι κάνοντας τις κατάλληλες αντικαταστάσεις στους παραπάνω τύπους μπορούμε να υπολογίσουμε την νέα θέση του οχήματος p᾽ με βάση τον τύπο:

$$p'=\begin{bmatrix} x+\frac{\Delta sr+\Delta sl}{2}\cos(\theta+\frac{\Delta sr-\Delta sl}{2b}) \\ y+\frac{\Delta sr+\Delta sl}{2}\sin(\theta+\frac{\Delta sr-\Delta sl}{2b}) \\ \theta+\frac{\Delta sr-\Delta sl}{b} \end{bmatrix}$$

Εάν οι αρχικές συντεταγμένες είναι γνωστές ή είναι δυνατόν να είναι αυθαίρετες (με λογική επιλογή το $[0\ 0\ 0]^T$) και κάθε φορά που κινείται το ρομπότ ανανεώνει την τιμή του p,είναι δυνατόν να γνωρίζει τις συντεταγμένες του με βάση τα παραπάνω. Δυστυχώς επειδή γενικά η μετακίνηση των ιμάντων δεν προκαλεί πάντα την αντίστοιχη μετακίνηση του οχήματος, μιας και πάντα υπάρχει κάποια ολίσθηση, οι παραπάνω τύποι δεν μπορούν να δώσουν την σωστή θέση του οχήματος μετά από κάποια μετατόπιση. Αν εξετάσουμε το σφάλμα που έχει η πραγματική θέση του οχήματος με την υπολογισμένη θα δούμε ότι αυτό οφείλεται κυρίως στην εσφαλμένη εκτίμηση του θ και όχι της απόστασης που κινήθηκε το όχημα. Ένα σφάλμα στην απόσταση που κινήθηκε το όχημα θα δώσει μια απόκλιση στην τελική θέση ίση με το σφάλμα. Ένα σφάλμα στην γωνιά θα δίνει ένα σφάλμα στην θέση διαρκώς αυξανόμενο κατά την μετατόπιση του ρομπότ. Επίσης όταν όχημα προσπαθεί να στρίψει που κυρίως τότε αλλάζει το θ οι ολισθήσεις είναι μεγαλύτερες και κατ᾽επέκταση το σφάλμα της οδομετρίας. Έτσι η θέση του οχήματος με βάση αυτόν τον τρόπο μπορεί μόνον για μικρές διαδρομές να υπολογιστεί. Στην εικόνα 20 φαίνεται η προοδευτική αύξηση του σφάλματος (πυγή [14]) για ευθύγραμμη κίνηση. Και στην εικόνα 21 για κυκλική (πυγή [14]).



Εικόνα 20 : Αύξηση σφάλματος στην ευθύγραμμη κίνηση



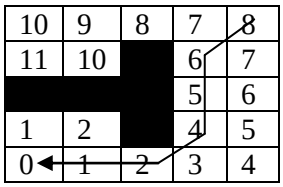
Εικόνα 21 : Αύξηση σφάλματος στην κυκλική κίνηση

Και στα δυο παραπάνω διαγράμματα έγινε η υπόθεση ότι τα σφάλμα στα Δs_r,Δs_l είναι ανεξάρτητα μεταξύ τους και η απόκλιση λόγο σφαλμάτων είναι ανάλογη των απολύτων τιμών των Δs_r,Δs_l. Με βάση λοιπόν ότι κύρια πυγή σφάλματος είναι το θ η πρώτη βελτίωση που θα έπρεπε να κάνει κάποιος σε αυτόν τον απλό τρόπο εύρεσης της θέσης είναι η βελτίωση του υπολογισμού του θ. Με χρήση δεδομένων από τον χάρτη βάθους είναι εύκολο να διατηρηθεί σταθερό το θ όταν το όχημα πρέπει να κινηθεί ευθεία και κάπως πιο δύσκολη η σωστή αλλαγή του θ όταν αυτό πρέπει να στρίψει. Το οχημα μου δεν εχει ενσωματομενο μηχανισμο οδομετριας που να μετρα την περιστροφη των ερπιστριων. Παρόλα αυτά η χρήση της παραπάνω ανάλυσης είναι δυνατή μιας και είναι δυνατό να γίνει εκτίμηση της κίνησης τους από τον ισχύ που δίνεται στους κινητήρες και από τον χρόνο για τον οποίο δίνεται αυτή. Ούτος η άλλος

με την παραπάνω μέθοδο δεν είναι δυνατός ο ακριβής υπολογισμός της κίνησης του οχήματος και η ακριβής μέτρηση της κίνησης τον ερπυστριών δεν θα έδινε δυνατότητα για ακριβή γνώση της θέσης του οχήματος κάθε στιγμή. Απλός θα μείωνε τα σφάλματα. Επισης ο υπολογισμος των συντεταγμενων του οχηματος γινεται με σκοπο αυτές να χρησιμοποιηθουν στα πλεσια ενός μικρου τοπικου χαρτη. Ο χαρτης αυτος εχει σαν σκοπο την αποφυγη των εμποδιον και είναι εφημερος. Μόλις κινηθεί αρκετά το όχημα ώστε να προσπεράσει τα εμπόδια που το έκαναν να καταφύγει σε μια πιο προσεκτική επιλογή της πορείας του, ο χάρτης αυτός ξανασχεδιάζεται και το όχημα παίρνει πάλι συντεταγμένες αφετηρίας. Έτσι δεν υπάρχουν αθροιστικά σφάλματα από πριν για κάθε νέα διαδρομή και η τιμή του σφάλματος περιορίζεται αν η απόσταση που διανύεται είναι περιορισμένη.

6.3 Ο σχεδιασμός του μονοπατιού του οχήματος

Προκειμένου να σχεδιαστεί η διαδρομή του οχήματος η δημιουργία ενός μικρού χάρτη εμποδίων είναι απαραίτητη. Ο χάρτης αυτός δημιουργείται με βάση τα δεδομένα του χάρτη βάθους. Προκειμένου να γίνει αυτό μια συγκεκριμένη περιοχή του χάρτη βάθους σαρώνεται. Σαρώνεται οριζόντια όλη η εικόνα για μια περιοχή ύψους που αντιστοιχεί στο ύψος του οχήματος. Για κάθε θέση στον πινάκα χάρτη βάθους που θα ελεγχθεί δυο πληροφορίες λαμβάνονται υπόψη. Η τιμή του που αντιστοιχεί στην απόσταση (έστω Z) και η θέση που βρέθηκε κατά την οριζόντια σάρωση που αντιστοιχεί στην θέση του στον άξονα x στο οπτικό πεδίο (Εστω X). Ο χάρτης εμποδίων αρχικά έχει σε όλες του τις θέσεις την τιμή 0 και αυτό θεωρείται σαν ελεύθερος χώρος. Για κάθε θέση που ελέγχεται στον χάρτη βάθους εάν αυτή είναι έγκυρη μέτρηση (δηλαδή δεν έχει την τιμή 0 αφού όπου απέτυχε το action pro να μετρήσει υπάρχει η τιμή μηδέν στο χάρτη βάθους) ένα σημείο παίρνει την τιμή -1 στον χάρτη εμποδίων. Για να βρεθεί η θέση του σημείου αυτού υπολογίζεται x=Z*X*c1 και y=Z*c2. Έτσι η καταγραφή γίνεται στην θέση x,y του χάρτη εμποδίων. Οι τιμές c1,c2 είναι συντελεστές κλίμακας . Ο πίνακας του χάρτη εμποδίων είναι 30*30. Στην συνεχεία προκειμένου να βρεθεί το μονοπάτι εκτελείται μια μορφή αναζήτησης κατά πλάτος , γνωστή ως grassfire. Η αναζήτηση ξεκινά από τον στόχο και κάθε ελεύθερο κελί στον πινάκα παίρνει σαν τιμή την απόσταση Manhattan από τον στόχο. Ξεκινώντας από την θέση του ρομπότ προσθέτουμε στο μονοπάτι της θέση από τις γειτονικές με την μικρότερη τιμή. Μετά κάνουμε το ίδιο για αυτήν την θέση μέχρι να φτάσουμε στον στόχο. Στην εικόνα 22 φαίνεται ένα παράδειγμα εύρεσης διαδρομής. Η εκκίνηση είναι δεξιά πάνω γωνιά και ο τερματισμός αριστερά κάτω γωνιά.



Εικόνα 22 : Παράδειγμα εύρεσης μονοπατιού μέσο grassfire

Στην συνέχεα το όχημα κινείται με βάση αυτό το μονοπάτι μέχρι τον στόχο. Όταν το όχημα φτάσει στον στόχο θα έχει ξεπεράσει τα εμπόδια που το έκαναν να σχεδιασει αναλυτικο μονοπατι. Εάν δεν υπάρχουν άλλα εμπόδια το όχημα θα κινηθεί με τον απλό αλγόριθμο που περιγράφτηκε στο 6.1

ΚΕΦΑΛΑΙΟ 7: ΕΠΙΛΟΓΟΣ - ΣΥΜΠΕΡΑΣΜΑΤΑ

7.1 Οι εμπειρίες και οι διαπιστώσεις από την εκπόνηση της πτυχιακής

Η κατασκευή ενός ρομπότ απαιτεί τον σχεδιασμό ενός συστήματος πραγματικού χρόνου που γενικά ο συνδυασμός διαφόρων τύπον δεδομένων εισόδου από διαφορετικούς αισθητήρες και συσκευές εισόδου που θα πρέπει ερμηνευτούν σαν συγκεκριμένα γεγονότα. Αυτά με την σειρά τους θα πρέπει να οδηγήσουν σε μια άμεση ή σε συγκεκριμένο χρόνο ανταπόκριση από τους επενεργητές του ρομπότ.

Η παρούσα πτυχιακή περιελάμβανε πολλές διαφορετικές μεθόδους και τεχνικές οι οποίες έπρεπε να συνδυαστούν μεταξύ του έτσι ώστε να δουλεύουν αρμονικά και συγχρονισμένα. Επίσης ορισμένα μέρη του συστήματος αναπτύχτηκαν και χρησιμοποιήθηκαν σε διαφορετικές πλατφόρμες. Για παράδειγμα τα τεχνητά νευρωνικά δίκτυα εκπαιδευτήκαν και αναπτύχτηκαν σε περιβάλλον matlab επειδή σε αυτό το περιβάλλον είναι εύκολη η αλλαγή της δομής και των παραμέτρων του νευρωνικού δικτύου. Επίσης είναι εύκολη ακόμα και η αλλαγή των παραμέτρων της εκπαίδευσης. Δηλαδή ήταν ένα περιβάλλον πιο κατάλληλο για την ανάπτυξη του νευρωνικού δικτύου. Από τη άλλη η χρήση του νευρωνικού δικτύου έγινε σε περιβάλλον c++ γιατί σε αυτό το περιβάλλον η εξομοίωση γίνεται πιο γρήγορα καθώς επίσης και επειδή η υπόλοιπη εφαρμογή είναι γραμμένη σε c++ και έτσι η δημιουργία μιας ενιαίας εφαρμογής ήταν πιο εύκολη. Έτσι αυτή η επιλογή είχε σαν αποτέλεσμα την εξεύρεση τρόπον με σκοπό την διασύνδεση δυο διαφορετικών περιβαλλόντων προγραμματισμού. Μια άλλη ενδιαφέρουσα παρατήρηση ήταν πως μπορούν τα νευρωνικά δίκτυα να συνεισφέρουν στην διατήρηση μιας λειτουργιάς που είναι σε συγκεκριμένες συνθήκες εύκολη αλλά σε διαφορετικές δεν λειτουργεί. Πιο συγκεκριμένα στην περίπτωση της αναγνώρισης του σώματος του χρήστη το σύστημα του user tracking παρείχε ένα σύνολο από δεδομένα αρκετά συμπυκνωμένα και περιεχτικά με δυνατότητα για πιο μεγάλη συμπύκνωση κατάλληλα για εύκολη ταξινόμηση των προτύπων από ένα νευρωνικό. Το πιο σημαντικό όμως είναι ότι ήταν δυνατή η αυτόματη συγκέντρωση πλήθους παραδειγμάτων κατάλληλα για τόσο για θετικά όσο και για αρνητικά δείγματα. Όλα αυτά τα δείγματα δεν θα ήταν εύκολο να απομονωθούν χειροκίνητα. Επίσης θα ήταν πρακτικά δύσκολη η συγκέντρωση τέτοιου αριθμού δειγμάτων.

7.2 Τα πλεονεκτήματα και οι δυνατότητες του ρομποτικού οχήματος.

Το συγκεκριμένο όχημα είναι φτηνό σε σχέση με άλλες εναλλακτικές. Τα υλικά κατασκευής του δεν είναι δυσεύρετα ή δύσχρηστα ούτε υπάρχει μεγάλη δυσκολία στην συναρμολόγηση του οχήματος. Δεν υπάρχει ανάγκη να φέρει κάποια συγκεκριμένα διακριτικά ή συσκευές ο χρήστης προκειμένου να τον ακολουθήσει το ρομπότ. Ακόμα και κάτι να αλλάξει στον ρουχισμό του χρήστη όπως για παράδειγμα το να φορέσει ένα μπουφάν εάν αυτό δεν γίνει όταν το ρομπότ έχει χάσει τον εντοπισμό, το ρομπότ μετά από λίγα frame θα εντοπίζει τον χρήστη με βάση τα νέα ρούχα. Τα ρούχα χρησιμεύουν για τον εντοπισμό του χρήστη μετά από προσωρινή απώλεια εντοπισμού και όταν δεν είναι ορατό το πρόσωπο. Ο μηχανισμός ταυτοποίησης του χρήστη δεν έχει ιδιαίτερες απαιτήσεις όσον αφορά τον φωτισμό. Επίσης με την προϋπόθεση ότι θα φαίνεται όλο το πρόσωπο η κλίση του κεφαλιού και η απόσταση εάν βρισκόμαστε μέσα στην εμβέλεια του action pro, δεν επηρεάζουν σημαντικά την ταυτοποίηση. Ακόμα ύπαρξη γυαλιών και σημαντικές διαφορές στην κόμμωση πολλές φορές δεν επηρεάζουν την ταυτοποίηση. Το όχημα μπορεί να ξαναβρεί τον χρήστη και να τον επαναταυτοποιησει τις περισσότερες φορές μετά από σύντομη απώλεια του εντοπισμού ακόμα και αν αυτός βρίσκεται εκτός του οπτικού πεδίου του οχήματος. Γενικά δεν υπάρχει η ανάγκη να υπάρχει οπτική επαφή με όλο το σώμα του χρήστη για να διατηρηθεί ο εντοπισμός. Ο μηχανισμός που παρέχει σε μια υλοποίηση τις παρακάτω λειτουργιές ταυτοποίηση-εντοπισμός-ακολούθηση είναι απαραίτητος σχεδόν σε κάθε ρομποτικό βοηθό προκειμένου αυτός να προσφέρει ένα φυσικό τρόπο χρήσης αυτού. Έτσι οι τεχνικές που αναπτύχτηκαν στα πλαίσια αυτής της εργασίας μπορούν να μεταφερθούν σε διαφορετικά ρομπότ και να αποτελέσουν βάση για τα αντίστοιχα υποσυστήματα σε ένα ρομπότ με πολύ περισσότερες δυνατότητες από το δικό μου.

7.3 Οι αδυναμίες του ρομποτικού οχήματος.

Παρόλο που το σύστημα φαίνεται να τα καταφέρνει αρκετά καλά στο να ακολουθεί κάποιον, δεν μπορεί όντος να το καταφέρει αυτό σε κάθε περίπτωση. Για παράδειγμα του είναι αδύνατον να ανέβει και να κατέβει σκάλες πράγμα που περιορίζει σημαντικά την χρησιμότητα του σε πραγματικές συνθήκες. Επίσης το έντονο φως του ηλίου μπορεί να υπερκαλύψει το δομημένο φως που προβάλλεται από το action pro και έτσι η χρήση του σε εξωτερικό περιβάλλον περιορίζεται δραστικά. Επίσης λόγο του μάλλον μικρού μεγέθους του οχήματος παρόλο που αυτό του δίνει μια σχετική ευελιξία όσον αφορά το που μπορεί να χωρέσει, δεν μπορεί να μεταφέρει κάποιο σημαντικό φορτίο πάνω του ούτε έχει κάποιον ειδικά διαμορφωμένο χώρο. Βέβαια σε αυτήν την εργασία δεν ήταν σκοπός η δημιουργία ενός ρομπότ με

τις φυσικές διαστάσεις που θα έπρεπε να έχει για να εκτελεί τέτοιες εργασίες αλλά κυρίως η ανάπτυξη του κατάλληλου λογισμικού για ένα τέτοιο όχημα και σε κάποιο βαθμό του ηλεκτρονικού του μέρους. Ένα άλλο πρόβλημα που υπάρχει είναι το περιορισμένο οπτικό πεδίο του ρομπότ. Ούτος ή άλλος εκτός εάν χρησιμοποιηθεί μια πανκετευθηντική κάμερα το οπτικό πεδίο δεν μπορεί να είναι 360 μοίρες. Επίσης οι υπάρχουσες πανκετευθηντικές κάμερες δεν προσφέρουν έναν χάρτη βάθους όπως κάνει το Kinect για παράδειγμα. Στην περίπτωση του action pro το οπτικό πεδίο είναι 58 H, 45 V, 70 D που δεν είναι και πολύ ευρύ για ένα ρομποτικό όχημα. Έτσι το όχημα αναγκάζεται να κινηθεί ολόκληρο προκειμένου να έχει μια καλύτερη οπτική του χώρου.

7.4 Ιδέες για μελλοντικές αναβαθμισεις ,διόρθωση των αδυναμιών και γενικά για βελτιώσεις

Όσον αφορά το πρόβλημα του μικρού οπτικού πεδίου εκτός από την προσπάθεια για ανεύρεση μιας κάμερας με ευρύτερο οπτικό πεδίο ένας μηχανισμός που θα επιτρέπει την κίνηση του action pro live τόσο οριζόντια όσο και κατακόρυφα θα δώσει την δυνατότητα στο όχημα να αποκτήσει στα γρήγορα μια καλή πληροφόρηση για τον χώρο γύρω του. Το ηλεκτρονικό μέρος του οχήματος με λίγες προσθήκες υποστηρίζει την οδήγηση επιπλέων κινητήρων καθώς επίσης μια τέτοια τροποποίηση δεν είναι ιδιαίτερα δύσκολη μηχανικά. Οπότε είναι μάλλον μια από τις πιο εύκολες βελτιώσεις. Όσον αφορά την αδυναμία του ρομπότ να μεταφέρει ικανοποιητικά μεγάλα φορτία και να ανέβει σκάλες ριζικές αλλαγές θα πρέπει να γίνουν στο μηχανικό μέρος του οχήματος. Το πιο εύκολο που μπορεί να γίνει είναι η αντικατάσταση του οχήματος με ένα μεγαλύτερο που θα έχει και ειδικά διαμορφωμένο χώρο για την τοποθέτηση πραγμάτων ώστε να είναι δυνατή η μεταφορά φορτίων. Όσον αφορά την ικανότητα του να ανεβαίνει σκάλες θα πρέπει να χρησιμοποιηθούν πιθανόν άλλου τύπου ερπύστριες και καλό θα ήταν να είναι δυνατό εκτός από την κίνηση τους να αλλάζει και η κλίση τους ως προς το κυρίως όχημα ώστε να μπορεί να κρατηθεί η ισορροπία. Μια σχετικά εύκολη προσθήκη που απαιτεί αλλαγές μόνο στο λογισμικό είναι να αποκτήσει το ρομπότ την δυνατότητα αφού ακολουθήσει κάποιον να επιστρέφει στην αφετηρία του. Εφόσον η διαδρομή είναι αποθηκευμένη το ρομπότ θα μπορεί να ξαναεπιστρέψει εκεί που άφησε το χρήστης του . Σε αυτήν την περίπτωση όσον αφορά το σενάριο της μεταφοράς φορτίων θα είναι δυνατόν αντί ο χρήστης να οδηγεί το φορτωμένο όχημα πίσω στο σημείο εκκίνησης αυτό να επιστρέφει μόνο του. Μετά κάποιος θα μπορεί να βρίσκεται στην αφετηρία και αφού ξεφορτώσει το όχημα να του δίνει εντολή να επιστρέψει στην θέση του αρχικού χρήστη και μετά να επαναληφτεί ο κύκλος. Επίσης εκτός από τον μηχανισμό user tracking κάποιες φορές και το skeleton tracking εμφανίζει δυσλειτουργίες. Έτσι μια ενίσχυση αυτού με νευρωνικά δίκτυα όπως έγινε στην περίπτωση του user tracking είναι μια πιθανή μελλοντική βελτίωση.

Παράρτημα Α

Παραπομπές

References

[1] H. Sidenbladh, D. Kragic, and H. I. Christensen, “A person following behavior for a mobile robot,” in Proc. 1999 IEEE Int. Conf. Robotics and Automation, 1999, pp. 670–675.

[2] S.-O. Lee, M. Hwang-Bo, B.-J. You, S.-R. Oh, and Y.-J. Cho, “Vision based mobile robot control for target tracking,” in Proc. IFAC Workshop Mobile Robot Technology, 2001, pp. 73–78.

[3] E. Prassler, D. Bank, B. Kluge, and M. Hagele, “Key technologies in robot assistants: Motion coordination between a human and a mobile robot,” in Proc. 32nd Int. Symp. Robotics, 2001, pp. 410–415.

[4] Masahiro Fujita, "AIBO: Toward the Era of Digital Creatures," in the International Journal of Robotics Research, October 2001 vol. 20 no. 10781-794.

[5] Quoc Khanh Dang; Young Soo Suh; , "Human-following robot using infrared camera," Control, Automation and Systems (ICCAS), 2011 11th International Conference on , vol., no., pp.1054-1058, 26-29 Oct. 2011

[6] R. Gockley, J. Forlizzi, and R. Simmons, “Natural person-following behavior for social robots,” in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, 2007

[7] MS Kinect, http://www.microsoft.com/en-us/kinectforwindows/

[8] ASUS Xtion Pro, http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO/

[9] Albrektsen, Sigurd Mørkved, “Using the Kinect Sensor for Social Robotics,” Student thesis, Institutt for teknisk kybernetikk, Norwegian University of Science and Technology, 2011

[10] Hoshino, F.; Morioka, K.; , "Human following robot based on control of particle distribution with integrated range sensors," System Integration (SII), 2011 IEEE/SICE International Symposium on , vol., no., pp.212-217, 20-22 Dec. 2011

[11] OpenNI framework, http://www.openni.org/ and http://openni.org/Documentation/

[12] Robert Bodor and Bennett Jackson and Nikolaos Papanikolopoulos, “Vision-based human tracking and activity recognition,” in Proc. of the 11th Mediterranean Conf. on Control and Automation, 2003, pp. 18-20.

[13] www.wikipedia.org

[14] Intruduction Autonomus Mobile Robots R.Siegwart R.Nourbakhsh isbn 9780262195027

[15] Computer Vision A Modern Approach isbn 0-27376414-4

[16] http://www.iheartrobotics.com

[17] Henry A. Rowley, Shumeet Baluja, Takeo Kanade , “Neural NetworkBased Face Detection” , Appears in *Computer Vision and Pattern Recognition*, 1996.

Παράρτημα Β
Πλήρης πηγαίος κώδικας εκτελέσιμου γραμμένου σε c++

Αρχείο main.cpp

```
/****************************************************************************
*                                                                           *
*  OpenNI 1.0 Alpha                                                         *
*                                                                           *
*  Copyright (C) 2010 PrimeSense Ltd.                                       *
*                                                                           *
*                                                                           *
*  This file is part of OpenNI.                                             *
*                                                                           *
*                                                                           *
*  OpenNI is free software: you can redistribute it and/or modify           *
*  it under the terms of the GNU Lesser General Public License as published *
*  by the Free Software Foundation, either version 3 of the License, or     *
*  (at your option) any later version.                                      *
*                                                                           *
*                                                                           *
*  OpenNI is distributed in the hope that it will be useful,                *
*  but WITHOUT ANY WARRANTY; without even the implied warranty of           *
*  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the             *
*  GNU Lesser General Public License for more details.                      *
*                                                                           *
*                                                                           *
*  You should have received a copy of the GNU Lesser General Public License *
*  along with OpenNI. If not, see <http://www.gnu.org/licenses/>.           *
*                                                                           *
****************************************************************************/



//---------------------------------------------------------------------------
// Includes
//---------------------------------------------------------------------------
#include <XnOpenNI.h>
#include <XnCodecIDs.h>
#include <XnCppWrapper.h>
#include "SceneDrawer.h"
#include <iostream>
#include <math.h>
#include <fstream>
#include <string.h>
#include <mat.h>
using namespace std;

//---------------------------------------------------------------------------
// Globals
//---------------------------------------------------------------------------
xn::Context g_Context;
xn::DepthGenerator g_DepthGenerator;
xn::ImageGenerator g_ImageGenerator;
xn::UserGenerator g_UserGenerator;

ofstream outkef("kefali.m",ios::out | ios::binary);
ofstream outn("neyronikoXarDHKx.m",ios::out | ios::binary);
ofstream outnh("neyronikoh.m",ios::out | ios::binary);
ofstream outnnoth("neyronikonoth.m",ios::out | ios::binary);
ofstream tst1("oikonesEisNeyExs.m",ios::out | ios::binary);
ofstream XarVath("XartisVathous.m",ios::out | ios::binary);
#include "struct_net.h"
#include "struct_net_mat.h"
#include "loadnetfcf.h"
net diktio;
net_mat diktio_mat;
net_mat diktio_mation;
net_mat diktio_mationYp;
XnBool g_bNeedPose = FALSE;
XnChar g_strPose[20] = "";
XnBool g_bDrawBackground = TRUE;
XnBool g_bDrawPixels = TRUE;
XnBool g_bDrawSkeleton = TRUE;
XnBool g_bPrintID = TRUE;
XnBool g_bPrintState = TRUE;

#if (XN_PLATFORM == XN_PLATFORM_MACOSX)
 #include <GLUT/glut.h>
#else
 #include <GL/glut.h>
#endif

#define GL_WIN_SIZE_X 720
#define GL_WIN_SIZE_Y 480

XnBool g_bPause = false;
XnBool g_bRecord = false;
```

```
XnBool g_bQuit = false;

//---------------------------------------------------------------------------
// Code
//---------------------------------------------------------------------------

double logsig(double a)
{
 return (1 / (1 + exp(-a)));
}
double tansig(double a)
{
 return (2/(1+exp(-2*a))-1);
}

void CleanupExit()
{
 g_Context.Shutdown();

 exit (1);
}

// Callback: New user was detected
void XN_CALLBACK_TYPE
User_NewUser(xn::UserGenerator& generator,
XnUserID nId, void* pCookie)
{
 printf("New User %d\n", nId);
 // New user found
 if (g_bNeedPose)
 {

 g_UserGenerator.GetPoseDetectionCap().StartP
oseDetection(g_strPose, nId);
 }
 else
 {

 g_UserGenerator.GetSkeletonCap().RequestCal
ibration(nId, TRUE);
 }
}
// Callback: An existing user was lost
void XN_CALLBACK_TYPE
User_LostUser(xn::UserGenerator& generator,
XnUserID nId, void* pCookie)
{
 printf("Lost user %d\n", nId);
}
// Callback: Detected a pose
void XN_CALLBACK_TYPE
UserPose_PoseDetected(xn::PoseDetectionCapa
bility& capability, const XnChar* strPose,
XnUserID nId, void* pCookie)
{
 printf("Pose %s detected for user %d\n", strPose,
nId);
 g_UserGenerator.GetPoseDetectionCap().StopP
oseDetection(nId);
 g_UserGenerator.GetSkeletonCap().RequestCal
ibration(nId, TRUE);
}
// Callback: Started calibration
void XN_CALLBACK_TYPE
UserCalibration_CalibrationStart(xn::SkeletonCa
pability& capability, XnUserID nId, void*
pCookie)
{
 printf("Calibration started for user %d\n", nId);
}
// Callback: Finished calibration
void XN_CALLBACK_TYPE
UserCalibration_CalibrationEnd(xn::SkeletonCa
pability& capability, XnUserID nId, XnBool
bSuccess, void* pCookie)
{
 if (bSuccess)
 {
  // Calibration succeeded
  printf("Calibration complete, start tracking user
%d\n", nId);

 g_UserGenerator.GetSkeletonCap().StartTracki
ng(nId);
 }
 else
 {
  // Calibration failed
  printf("Calibration failed for user %d\n", nId);
  if (g_bNeedPose)
  {

 g_UserGenerator.GetPoseDetectionCap().StartP
oseDetection(g_strPose, nId);
  }
  else
  {

 g_UserGenerator.GetSkeletonCap().RequestCal
ibration(nId, TRUE);
  }
 }
}
HANDLE hCom;
// this function is called each frame
void glutDisplay (void)
{
```

```
 glClear (GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);

 // Setup the OpenGL viewpoint
 glMatrixMode(GL_PROJECTION);
 glPushMatrix();
 glLoadIdentity();

 xn::SceneMetaData sceneMD;
 xn::DepthMetaData depthMD;
 xn::ImageMetaData imMD;
 //g_DepthGenerator.GetMetaData(depthMD);
 g_ImageGenerator.GetMetaData(imMD);
 glOrtho(0, imMD.XRes(), imMD.YRes(), 0,
-1.0, 1.0);
 //glOrtho(0, depthMD.XRes(),
depthMD.YRes(), 0, -1.0, 1.0);


 glDisable(GL_TEXTURE_2D);

 if (!g_bPause)
 {
  // Read next available data
  g_Context.WaitAndUpdateAll();
 }

  // Process the data
  g_DepthGenerator.GetMetaData(depthMD);
  g_UserGenerator.GetUserPixels(0, sceneMD);
  g_ImageGenerator.GetMetaData(imMD);
  DrawDepthMap(depthMD,
sceneMD,hCom,imMD);

 glutSwapBuffers();
}

void glutIdle (void)
{
 if (g_bQuit) {
  CleanupExit();
 }

 // Display the frame
 glutPostRedisplay();
}

void glInit (int * pargc, char ** argv)
{
 glutInit(pargc, argv);
 glutInitDisplayMode(GLUT_RGB |
GLUT_DOUBLE | GLUT_DEPTH);
 glutInitWindowSize(GL_WIN_SIZE_X,
GL_WIN_SIZE_Y);
 glutCreateWindow ("Prime Sense User Tracker
Viewer");
 //glutFullScreen();
 glutSetCursor(GLUT_CURSOR_NONE);

 //glutKeyboardFunc(glutKeyboard);
 glutDisplayFunc(glutDisplay);
 glutIdleFunc(glutIdle);

 glDisable(GL_DEPTH_TEST);
 glEnable(GL_TEXTURE_2D);

 glEnableClientState(GL_VERTEX_ARRAY);
 glDisableClientState(GL_COLOR_ARRAY);
}

#define SAMPLE_XML_PATH
"../../Data/SamplesConfig.xml"

#define CHECK_RC(nRetVal, what)          \
 if (nRetVal != XN_STATUS_OK)          \
 {                  \
  printf("%s failed: %s\n", what,
xnGetStatusString(nRetVal));\
  return nRetVal;            \
 }
void PrintCommState(DCB dcb)
{
   //  Print some of the DCB structure values
  /* printf( TEXT("\nBaudRate = %d, ByteSize =
%d, Parity = %d, StopBits = %d\n"),
           dcb.BaudRate,
           dcb.ByteSize,
           dcb.Parity,
           dcb.StopBits );*/
}

void fanOn(HANDLE hCom)
{
   DWORD       bytes_written = 0;
   char dedom[6]={1,2,58,1,1,13};
   int bStatus = WriteFile(hCom,      // Handle
          &dedom,      // Outgoing data
          6,          // Number of bytes to write
          &bytes_written,  // Number of bytes
written
          NULL);
}



int main(int argc, char **argv)
{

 tst1<<"lined=[];\nYpsiEntopismou=[];\nApokri
siNetYm=[];\nApokrisiNetAnP=[];\n";
```

```
  outn<<"xarDa=[];\nXromata=[];\n";
  XarVath<<"Vathos=[];\n";
  //{
  ifstream
inn("FaceClassWhithEyeIY2.net",ios::in |
ios::binary);
  if(!inn)
  {
   cout<<"Den mporo na anikso to arxeio
neyronikLW\n";
   return -1;
  }
  loadnetfmcfile(diktio_mat,inn);
  inn.close();
  ifstream inn2("matiaFnn5.net",ios::in |
ios::binary);
  if(!inn2)
  {
   cout<<"Den mporo na anikso to arxeio
neyronikLW\n";
   return -1;
  }
  loadnetfmcfile(diktio_mation,inn2);
  inn2.close();

  ifstream inn3("matiaFnnYp19.net",ios::in |
ios::binary);
  if(!inn3)
  {
   cout<<"Den mporo na anikso to arxeio
neyronikLW\n";
   return -1;
  }
  loadnetfmcfile(diktio_mationYp,inn3);
  inn3.close();
  /*const char *file = "m.mat";
  int i;
  cin>>i;
  MATFile *pmat,*o;
  if(i<2)
  {
  // pmat = matOpen(file, "r");
  }
  cin>>i;

 }//*/
 //#include "sim_net.h"
 //netsim



 //tst1<<"img=[];\r\nline=[];\r\n";
 //tst1<<"p123";
 //tst2.put('a');
 cout<<"me RGB\n";
 outkef<<"ke=[";
 #include "loadnet.h"
 {
  double
nd[]={1,2,3,4,5,6,7,8,9,10,1,2,3,4,5,6,7,8,9,10,1,
2,3,4,5,6,7,8,9,0,0,0,20,200,100};
  for(int m=0;m<35;m++)
  {
   diktio.inlayer[m]=nd[m];
  }
  #include "neural_network_core.h"
 }//*/
 //#include "openCom1.h"
 //fanOn(hCom);
 XnStatus nRetVal = XN_STATUS_OK;

 if (argc > 1)
 {
  nRetVal = g_Context.Init();
  CHECK_RC(nRetVal, "Init");
  nRetVal =
g_Context.OpenFileRecording(argv[1]);
  if (nRetVal != XN_STATUS_OK)
  {
   printf("Can't open recording %s: %s\n",
argv[1], xnGetStatusString(nRetVal));
   return 1;
  }
 }
 else
 {
  xn::EnumerationErrors errors;
  nRetVal =
g_Context.InitFromXmlFile(SAMPLE_XML_P
ATH, &errors);
  if (nRetVal ==
XN_STATUS_NO_NODE_PRESENT)
  {
   XnChar strError[1024];
   errors.ToString(strError, 1024);
   printf("%s(SAMPLE_XML_PATH)\n",
strError);
   return (nRetVal);
  }
  else if (nRetVal != XN_STATUS_OK)
  {
   printf("Open failed:
%s(SAMPLE_XML_PATH)\n",
xnGetStatusString(nRetVal));
   return (nRetVal);
  }
 }

 nRetVal =
```

```
g_Context.FindExistingNode(XN_NODE_TYP
E_DEPTH, g_DepthGenerator);
 CHECK_RC(nRetVal, "Find depth generator");
 nRetVal =
g_Context.FindExistingNode(XN_NODE_TYP
E_USER, g_UserGenerator);
 if (nRetVal != XN_STATUS_OK)
 {
  nRetVal =
g_UserGenerator.Create(g_Context);
  CHECK_RC(nRetVal, "Find user generator");
 }
 nRetVal =
g_Context.FindExistingNode(XN_NODE_TYP
E_IMAGE, g_ImageGenerator);
 CHECK_RC(nRetVal, "Find depth generator");

 XnCallbackHandle hUserCallbacks,
hCalibrationCallbacks, hPoseCallbacks;
 if
(!g_UserGenerator.IsCapabilitySupported(XN_C
APABILITY_SKELETON))
 {
  printf("Supplied user generator doesn't support
skeleton\n");
  return 1;
 }
 g_UserGenerator.RegisterUserCallbacks(User_
NewUser, User_LostUser, NULL,
hUserCallbacks);
 g_UserGenerator.GetSkeletonCap().RegisterCal
ibrationCallbacks(UserCalibration_CalibrationSt
art, UserCalibration_CalibrationEnd, NULL,
hCalibrationCallbacks);

 if
(g_UserGenerator.GetSkeletonCap().NeedPoseF
orCalibration())
 {
  g_bNeedPose = TRUE;
  if
(!g_UserGenerator.IsCapabilitySupported(XN_C
APABILITY_POSE_DETECTION))
  {
   printf("Pose required, but not supported\n");
   return 1;
  }

 g_UserGenerator.GetPoseDetectionCap().Regis
terToPoseCallbacks(UserPose_PoseDetected,
NULL, NULL, hPoseCallbacks);

 g_UserGenerator.GetSkeletonCap().GetCalibrat
ionPose(g_strPose);
 }
 g_UserGenerator.GetSkeletonCap().SetSkeleton
Profile(XN_SKEL_PROFILE_ALL);

 nRetVal = g_Context.StartGeneratingAll();
 CHECK_RC(nRetVal, "StartGenerating");

 glInit(&argc, argv);
 printf("Trexome\n");
 glutMainLoop();

}
```

Αρχειο class_net_mat.h

```
class net_mat
{
 //arithmos layer
 int nl;
 //arithmos isodon
 int ni;
 //sindeseis metaksi leyer,sinesis isodon me layer
 int **lcl,**icl;
 //bari
 double ***iw,***lw;
 //bias
 double **b;
 //layer outputs
 double **lout;
 //layer eksodou
 int outputl;
 //arithmos neyrono ana layer,megethos isodon
 int *nol,*isi;
public:
 int outSize()
 {
  return nol[outputl];
 }
 #include "sim_net.h"
 #include "loadnetfcf.h"
};
```

Αρχειο loadnetfcf.h

```
int loadnetfmcfile(char *fileName)
{
 ifstream inn(fileName,ios::in | ios::binary);
 if(!inn)
 {
  cout<<"Den mporo na anikso to arxeio
neyronikLW\n";
  return -1;
 }
 int temp;
```

```
//for(int mx=0;mx<4;mx++)
inn.read((char*)&nl,sizeof(nl));
cout<<"Arithmos Layer "<<nl<<"\n";

//neyrones ana layer,layer conekted 2 layer
nol=new int[nl];
lcl=new int*[nl];
for(int m=0;m<nl;m++)
{
 inn.read((char*)&temp,sizeof(temp));
 cout<<temp<<" ";
 nol[m]=temp;
 inn.read((char*)&temp,sizeof(temp));
 //cout<<temp<<"\t";
}
cout<<"\n";
for(int m=0;m<nl;m++)
{
 lcl[m]=new int[nl];
 for(int mj=0;mj<nl;mj++)
 {
  inn.read((char*)&temp,sizeof(temp));
  lcl[m][mj]=temp;
  cout<<temp<<" ";
 }
 cout<<"\n";
}
cout<<"\n";



//arithmos isodon
inn.read((char*)&ni,sizeof(ni));
cout<<ni<<"\nisi ";
//isi = megethos isodoy
isi=new int[ni];
icl=new int*[nl];
for(int m=0;m<ni;m++)
{
 inn.read((char*)&temp,sizeof(temp));
 isi[m]=temp;
 cout<<temp<<" ";
}
//cout<<"\n\n";
//sindesi isodoy me layer
for(int m=0;m<nl;m++)
{
 icl[m]=new int[ni];
 for(int mj=0;mj<ni;mj++)
 {
  inn.read((char*)&temp,sizeof(temp));
  icl[m][mj]=temp;
  //cout<<temp<<" ";
 }
 //cout<<"\n";
}
double w;
b=new double*[nl];
lout=new double*[nl];
for(int m=0;m<nl;m++)
{
 b[m]=new double[nol[m]];
 lout[m]=new double[nol[m]];
 for(int jk=0;jk<nol[m];jk++)
 {
  inn.read((char*)&b[m][jk],sizeof(b[m][jk]));
  //cout<<b[m][jk]<<" ";
 }
 //cout<<"\n\n";
}
iw=new double**[nl];
for(int m=0;m<nl;m++)
{
 iw[m]=new double*[ni];
 for(int jk=0;jk<ni;jk++)
 {
  if(icl[m][jk]==1)
  {
   iw[m][jk]=new double[nol[m]*isi[jk]];
   for(int sar=0;sar<nol[m]*isi[jk];sar++)
   {
    inn.read((char*)&w,sizeof(w));
    iw[m][jk][sar]=w;
    //cout<<w<<" ";
   }
  }
  else
  {
   iw[m][jk]=0;
  }
 }
 //cout<<"\n\n";
}
//cout<<iw[6][15][(30*225)-1]<<"\n";
lw=new double**[nl];
for(int m=0;m<nl;m++)
{
 lw[m]=new double*[nl];
 for(int jk=0;jk<nl;jk++)
 {
  if(lcl[m][jk]==1)
  {
   lw[m][jk]=new double[nol[m]*nol[jk]];
   for(int sar=0;sar<nol[m]*nol[jk];sar++)
   {
    inn.read((char*)&w,sizeof(w));
    lw[m][jk][sar]=w;
    //cout<<w<<" ";
   }
  }
```

```
    else
    {
     lw[m][jk]=0;
    }
   }
  //cout<<"\n\n";
 }
 inn.close();
 return 0;
}
```

Αρχειο SceneDrawer.cpp

```
/*****************************************
*****************************************
*                                           *
*  OpenNI 1.0 Alpha
*
*  Copyright (C) 2010 PrimeSense Ltd.
*
*                                           *
*  This file is part of OpenNI.
*
*                                           *
*  OpenNI is free software: you can redistribute it
and/or modify           *
*  it under the terms of the GNU Lesser General
Public License as published  *
*  by the Free Software Foundation, either version
3 of the License, or      *
*  (at your option) any later version.
*
*                                           *
*  OpenNI is distributed in the hope that it will be
useful,                *
*  but WITHOUT ANY WARRANTY; without
even the implied warranty of           *
*  MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the            *
*  GNU Lesser General Public License for more
details.                   *
*                                           *
*  You should have received a copy of the GNU
Lesser General Public License  *
*  along with OpenNI. If not, see
<http://www.gnu.org/licenses/>.           *
*                                           *
*****************************************
*****************************************/



//-------------------------------------------------------
-----------------
```

```
// Includes
//-------------------------------------------------------
-----------------
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <math.h>
#include <atlbase.h>
#include <string.h>
#include <windows.h>
#include <time.h>
using namespace std;

#include "SceneDrawer.h"

#if (XN_PLATFORM ==
XN_PLATFORM_MACOSX)
 #include <GLUT/glut.h>
#else
 #include <GL/glut.h>
#endif

extern xn::UserGenerator g_UserGenerator;
extern xn::DepthGenerator g_DepthGenerator;

extern XnBool g_bDrawBackground;
extern XnBool g_bDrawPixels;
extern XnBool g_bDrawSkeleton;
extern XnBool g_bPrintID;
extern XnBool g_bPrintState;
extern ofstream
outn,outnh,outnnoth,outkef,tst1,XarVath;
extern double logsig(double a);
extern double tansig(double a);
#include "struct_net.h"
#include "struct_net_mat.h"
#include "sim_net.h"
extern net diktio;
extern net_mat diktio_mat;
extern net_mat diktio_mation;
extern net_mat diktio_mationYp;
struct
{
 double lastV[40][5];
}meameans4efns[15];
static int cMetr=0;
int cu=1;
//extern net network;

#define MAX_DEPTH 10000
#define XANAL 5
float g_pDepthHist[MAX_DEPTH];

unsigned int getClosestPowerOfTwo(unsigned
int n)
```

```
{
 unsigned int m = 2;
 while(m < n) m<<=1;

 return m;
}
GLuint initTexture(void** buf, int& width, int&
height)
{
 GLuint texID = 0;
 glGenTextures(1,&texID);

 width = getClosestPowerOfTwo(width);
 height = getClosestPowerOfTwo(height);
 *buf = new unsigned char[width*height*4];
 glBindTexture(GL_TEXTURE_2D,texID);

 glTexParameteri(GL_TEXTURE_2D,
GL_TEXTURE_MIN_FILTER, GL_LINEAR);
 glTexParameteri(GL_TEXTURE_2D,
GL_TEXTURE_MAG_FILTER, GL_LINEAR);

 return texID;
}

GLfloat texcoords[8];
void DrawRectangle(float topLeftX, float
topLeftY, float bottomRightX, float
bottomRightY)
{
 GLfloat verts[8] = { topLeftX, topLeftY,
  topLeftX, bottomRightY,
  bottomRightX, bottomRightY,
  bottomRightX, topLeftY
 };
 glVertexPointer(2, GL_FLOAT, 0, verts);
 glDrawArrays(GL_TRIANGLE_FAN, 0, 4);

 //TODO: Maybe glFinish needed here instead - if
there's some bad graphics crap
 glFlush();
}
void DrawTexture(float topLeftX, float topLeftY,
float bottomRightX, float bottomRightY)
{
 glEnableClientState(GL_TEXTURE_COORD_
ARRAY);
 glTexCoordPointer(2, GL_FLOAT, 0,
texcoords);

 DrawRectangle(topLeftX, topLeftY,
bottomRightX, bottomRightY);

 glDisableClientState(GL_TEXTURE_COORD
_ARRAY);
}

XnFloat Colors[][3] =
{
 {0,1,1},
 {0,0,1},
 {0,1,0},
 {1,1,0},
 {1,0,0},
 {1,.5,0},
 {.5,1,0},
 {0,.5,1},
 {.5,0,1},
 {1,1,.5},
 {1,1,1}
};
XnUInt32 nColors = 10;

void glPrintString(void *font, char *str)
{
 int i,l = strlen(str);

 for(i=0; i<l; i++)
 {
  glutBitmapCharacter(font,*str++);
 }
}

void DrawLimb(XnUserID player,
XnSkeletonJoint eJoint1, XnSkeletonJoint
eJoint2)
{
 if
(!g_UserGenerator.GetSkeletonCap().IsTracking
(player))
 {
  printf("not tracked!\n");
  return;
 }
 //cout<<".";
 XnSkeletonJointPosition joint1, joint2;
 g_UserGenerator.GetSkeletonCap().GetSkeleto
nJointPosition(player, eJoint1, joint1);
 g_UserGenerator.GetSkeletonCap().GetSkeleto
nJointPosition(player, eJoint2, joint2);

 if (joint1.fConfidence < 0.5 || joint2.fConfidence
< 0.5)
 {
  return;
 }

 XnPoint3D pt[2];
 pt[0] = joint1.position;
 pt[1] = joint2.position;
```

```
 g_DepthGenerator.ConvertRealWorldToProject
ive(2, pt, pt);
 glVertex3i(pt[0].X, pt[0].Y, 0);
 glVertex3i(pt[1].X, pt[1].Y, 0);
}

XnPoint3D GetLimbPos(XnUserID player,
XnSkeletonJoint eJoint1)
{
 XnSkeletonJointPosition joint1;
 g_UserGenerator.GetSkeletonCap().GetSkeleto
nJointPosition(player, eJoint1, joint1);
 cout<<"\nX"<<joint1.position.X<<"X\n";
 if
(!g_UserGenerator.GetSkeletonCap().IsTracking
(player))
 {
  printf("not tracked!\n");
 }
 if (joint1.fConfidence > 0.5)
 {
  XnPoint3D ret=joint1.position;

 g_DepthGenerator.ConvertRealWorldToProject
ive(1 , &ret, &ret);
  return ret;
 }
 cout<<"\nlib error\n";
}
void perimene(int ms)
{
 clock_t time;
   time=clock();
 cout<<"i wait for "<<ms<<" ms\n";
 for(;(clock()-time)<ms;)
 {
 }
}
#include "robotManageF.h"
//#include "robotManageFAsfairo.h"
#include "DrawDepthMap.h"
```

Αρχειο DrawDepthMap.h

```
void DrawDepthMap(const
xn::DepthMetaData& imMd, const
xn::SceneMetaData& smd, HANDLE
hCom,const xn::ImageMetaData& dmd)
{
 clock_t time,ttime;
   time=clock();
 double times[5];
 static int metritis=0;
 static int xorisXristi=0;
 static bool bInitialized = false;
 static GLuint depthTexID;
 static unsigned char* pDepthTexBuf;
 static int texWidth, texHeight;

 float topLeftX;
 float topLeftY;
 float bottomRightY;
 float bottomRightX;
 float texXpos;
 float texYpos;

 struct
 {
  XnSkeletonJointPosition uj;
  bool traced;
  bool daded;
 } xristis[15];


 unsigned char portret[64][61][3][15];

 float scaledPortret[75][60][15];

 float linedPortret[4500][15];

 float netPuser[7][15];

 int userd[15];

 //char
string[8][50]={"kostas","nikos","paris","tsampik
os","anestis","petros","mama","agnostos"};
 char
string[5][50]={"nikos","tsampikos","anestis","pe
tros","agnostos"};

 char strLabel[50] = "";
 XnUserID aUsers[15];
 XnUInt16 nUsers = 15;
 struct
 {
  XnPoint3D pos;
  //int W,Wc,W2,W2c;
  int paxos[20];
  int en[20];
  int ek[20];
  int ptimon;
 }usersData[15];
 xn::SceneMetaData usmd;
 const XnLabel *userPM;
```

```
 if(!bInitialized)
 {

  texWidth =
getClosestPowerOfTwo(dmd.XRes());
  texHeight =
getClosestPowerOfTwo(dmd.YRes());

//  printf("Initializing depth texture: width = %d,
height = %d\n", texWidth, texHeight);
  depthTexID =
initTexture((void**)&pDepthTexBuf,texWidth,
texHeight) ;

//  printf("Initialized depth texture: width = %d,
height = %d\n", texWidth, texHeight);
  bInitialized = true;

  topLeftX = dmd.XRes();
  topLeftY = 0;
  bottomRightY = dmd.YRes();
  bottomRightX = 0;
  texXpos =(float)dmd.XRes()/texWidth;
  texYpos  =(float)dmd.YRes()/texHeight;

  memset(texcoords, 0, 8*sizeof(float));
  texcoords[0] = texXpos, texcoords[1] =
texYpos, texcoords[2] = texXpos, texcoords[7] =
texYpos;

 }
 unsigned int nValue = 0;
 unsigned int nHistValue = 0;
 unsigned int nIndex = 0;
 unsigned int nX = 0;
 unsigned int nY = 0;
 unsigned int nNumberOfPoints = 0;
 XnUInt16 g_nXRes = dmd.XRes();
 XnUInt16 g_nYRes = dmd.YRes();

 unsigned char* pDestImage = pDepthTexBuf;

 const XnUInt8* idat = dmd.Data();
 const XnDepthPixel* pDepth = imMd.Data();
 const XnLabel* pLabels = smd.Data();

 // Calculate the accumulative histogram
 memset(g_pDepthHist, 0,
MAX_DEPTH*sizeof(float));
 for (nY=0; nY<g_nYRes; nY++)
 {
  for (nX=0; nX<g_nXRes; nX++)
  {
   nValue = *pDepth;

   if (nValue != 0)
   {
    g_pDepthHist[nValue]++;
    nNumberOfPoints++;
   }

   pDepth++;
  }
 }

 for (nIndex=1; nIndex<MAX_DEPTH;
nIndex++)
 {
  g_pDepthHist[nIndex] +=
g_pDepthHist[nIndex-1];
 }
 if (nNumberOfPoints)
 {
  for (nIndex=1; nIndex<MAX_DEPTH;
nIndex++)
  {
   g_pDepthHist[nIndex] = (unsigned int)(256 *
(1.0f - (g_pDepthHist[nIndex] /
nNumberOfPoints)));
  }
 }

 pDepth = imMd.Data();


 #include "DeclAndSetMX.h"


 unsigned char maxb=0;

 for(int i=0;i<15;i++)
 {
  for(int sy=0;sy<64;sy++)
  {
   for(int sx=0;sx<61;sx++)
   {
    for(int col=0;col<3;col++)
    {
     portret[sy][sx][col][i]=0;
    }
   }
  }
  for(int sy=0;sy<75;sy++)
  {
   for(int sx=0;sx<60;sx++)
   {
    scaledPortret[sy][sx][i]=0;
   }
```

```
 }
}

if (g_bDrawPixels)
{

 bool hasAman=false;

 XnUInt32 nIndex = 0;
 // Prepare the texture map
 unsigned char maxr=0;
 unsigned char maxg=0;
 maxb=0;
 for(int i=0;i<2;i++)
 {
  for(int sar=0;sar<15;sar++)
  {
   for(int sar2=0;sar2<256;sar2++)
   {
    mesoXroma.xromataEp[sar][0][sar2][i]=0;
    mesoXroma.xromataEp[sar][1][sar2][i]=0;
    mesoXroma.xromataEp[sar][2][sar2][i]=0;
   }
   mesoXroma.xromataMes[sar][0][i]=0;
   mesoXroma.xromataMes[sar][1][i]=0;
   mesoXroma.xromataMes[sar][2][i]=0;
   mesoXroma.paron[sar]=1;
   mesoXroma.countx[sar]=0;
   mesoXroma.maxcountx[sar]=0;
  }
 }
 pLabels = smd.Data();
 for(int i=0;i<g_nXRes*g_nYRes;i++)
 {
  if (g_bDrawBackground || *pLabels != 0)
  {
    XnLabel label = *pLabels;
    if (label != 0)
    {
     hasAman=true;
     break;
    }
  }
  pLabels++;
 }
 pLabels = smd.Data();
 if ( hasAman == true)
 {
  #include "userPros.h"
 }
 XarVath<<"b=[";
 #include "KefaliSarosis.h"

       if(maxr<idat[0])
       {
        maxr=idat[0];
       }
       if(maxg<idat[1])
       {
        maxg=idat[1];
       }
       if(maxb<idat[1])
       {
        maxb=idat[1];
       }
       /*pDestImage[0] = maxr;//idat[0];
       pDestImage[1] = maxg;//idat[1];
       pDestImage[2] = idat[2];//*/
       if(mesoXroma.paron[label]>3000 &&
mesoXroma.paron[label]<4500 &&
mesoXroma.countx[label]>30 &&
mesoXroma.countx[label]<90)
       {

 mesoXroma.xromataEp[label][0][idat[0]/XAN
AL][0]++;

 mesoXroma.xromataEp[label][1][idat[1]/XAN
AL][0]++;

 mesoXroma.xromataEp[label][2][idat[2]/XAN
AL][0]++;


 mesoXroma.xromataMes[label][0][0]+=idat[0];

 mesoXroma.xromataMes[label][1][0]+=idat[1];

 mesoXroma.xromataMes[label][2][0]+=idat[2];

        mesoXroma.parong[label]++;
       }

       if(mesoXroma.paron[label]>7000 &&
mesoXroma.paron[label]<8500 &&
mesoXroma.countx[label]>30 &&
mesoXroma.countx[label]<90)
       {

 mesoXroma.xromataEp[label][0][idat[0]/XAN
AL][1]++;

 mesoXroma.xromataEp[label][1][idat[1]/XAN
AL][1]++;

 mesoXroma.xromataEp[label][2][idat[2]/XAN
AL][1]++;


 mesoXroma.xromataMes[label][0][1]+=idat[0];
```

```
 mesoXroma.xromataMes[label][1][1]+=idat[1];

 mesoXroma.xromataMes[label][2][1]+=idat[2];

        }
        if(xristis[label].traced==true)
        {
         XnPoint3D pt=xristis[label].uj.position;

 g_DepthGenerator.ConvertRealWorldToProject
ive(1, &pt, &pt);
          if(abs(pt.X-nX)<30 &&
abs(pt.Y-nY+5)<30)
          {
           for(int col=0;col<3;col++)
           {
            int xk=nX-floor(pt.X)+30;
            int yk=nY-floor(pt.Y)+28;
            portret[yk][xk][col][label]=idat[col];
           }
          }
        }
        mesoXroma.paron[label]++;
        mesoXroma.countx[label]++;

 mesoXroma.maxcountx[label]=max(mesoXrom
a.countx[label],mesoXroma.maxcountx[label]);
       }
      }
     }
     XarVath<<pDepth[0]<<" ";
     //cout<<pDepth[0]<<" ";
     pDepth++;
     pLabels++;
     idat+=3;
     //pDestImage+=3;
    }

    //pDestImage += (texWidth - g_nXRes) *3;
   }
   XarVath<<"];\nVathos=[Vathos;b];\n";

   #include "calcMesokaiEpikr.h"


   #include "scaler.h"
   #include "liner.h"

   for(int xr=0;xr<15;xr++)
   {
    float megA=-1;
    int pos=-1;
    for(int neuA=0;neuA<4;neuA++)
    {
     if(megA<netPuser[neuA][xr])
     {
      megA=netPuser[neuA][xr];
      pos=neuA;
     }
    }
    if(megA>0.3)
    {
     userd[xr]=pos;
     //cout<<"\nxristis "<<xr<<"
"<<string[pos]<<"\n";
    }
    else
    {
     userd[xr]=4;
    }
   }
   #include "dravUser.h"
  }
  else
  {
   xnOSMemSet(pDepthTexBuf, 0,
3*2*g_nXRes*g_nYRes);
  }
  //#include "userPros.h"

  glBindTexture(GL_TEXTURE_2D,
depthTexID);
  glTexImage2D(GL_TEXTURE_2D, 0,
GL_RGB, texWidth, texHeight, 0, GL_RGB,
GL_UNSIGNED_BYTE, pDepthTexBuf);

  // Display the OpenGL texture map
  glColor4f(1,1,1,1);

  glEnable(GL_TEXTURE_2D);
  //glPrintString(GLUT_BITMAP_HELVETICA
_18, "alz");
  DrawTexture(dmd.XRes(),dmd.YRes(),0,0);
  glDisable(GL_TEXTURE_2D);
  glDisable(GL_LIGHTING);
  glColor3f(0.0, 1.0, 0.0);
  for(int i=1;i<8;i++)
  {
   glRasterPos2i((i*30)-29 ,41);
   if(userd[i]>=0)
   {

  glPrintString(GLUT_BITMAP_TIMES_ROM
AN_24, string[userd[i]]);
   }
   else
   {

  glPrintString(GLUT_BITMAP_HELVETICA_
```

```
18, "kanis");
  }
 }
 glDisable(GL_TEXTURE_2D);

 //#include "userPros.h"
 //#include "dravUser.h"
 //*/

 ttime=clock();
 times[1]=((double)(ttime-time))/1000;
 time=ttime;
 {
  int X,Y=-1,Z;
  bool fountet=false;
  for(int i=0;i<nUsers;i++)
  {

 if((usersData[i].pos.Y>100)&&(usersData[i].po
s.Y<155))
    {
     //cout<<"\nx0 " <<usersData[i].pos.X<<" x1
"<<usersData[i].pos.X<<" z
"<<usersData[i].pos.Z<<"\n";

 //if((usersData[i].Wc>65)&&(usersData[i].Wc<
160))
     {
      bool Human=true;//teliosProsorino
      #include "apothikeysiDedomenonGiaNN.h"
      //#include "neural_network.h"
      bool isOk=false;
      //#include "diakrisiMeIf.h";
      isOk=Human;
      if(isOk==true)
      {
       fountet=true;
       int d=abs(usersData[i].pos.Y-115);
       if(d<abs(Y-115))
       {
        X=usersData[i].pos.X;
        Y=usersData[i].pos.Y;
        Z=usersData[i].pos.Z;
       }
      }
     }
    }
   }
   if(fountet==true)
   {
    outn<<"a=[";
    idat = dmd.Data();
    for(int
i=0;i<(g_nYRes*g_nXRes);i++,idat+=3)
    {
     outn<<(int)idat[0]<<" "<<(int)idat[1]<<"
"<<(int)idat[2]<<" ";
    }
    outn<<"];\nXromata=[Xromata;a];\n";
   }
   if((nUsers>0)&&(fountet==true))
   {
    //#include "ploigisiBasiThesis.h"
   }
   else if(xorisXristi>20)
   {
    srl(hCom);
    xorisXristi-=4;
   }
   else
   {
    st(hCom);
    //xorisXristi++;
    //cout<<"xorisXristi"<<xorisXristi<<"\n";prs
   }

  }
  ttime=clock();
  times[2]=((double)(ttime-time))/1000;
  time=ttime;
  /*cout<<"times ";
  for(int t=0;t<3;t++)
  {
   cout<<times[t]<<"\t";
  }
  cout<<"\n";*/
 }
```

Αρχειο userPros.h

```
/*
char strLabel[50] = "";
XnUserID aUsers[15];
XnUInt16 nUsers = 15;
struct
{
 XnPoint3D pos;
 //int W,Wc,W2,W2c;
 int paxos[20];
 int en[20];
 int ek[20];
 int ptimon;
}usersData[15];*/
g_UserGenerator.GetUsers(aUsers, nUsers);
ttime=clock();
times[0]=((double)(ttime-time))/1000;
time=ttime;
//cout<<nUsers<<"\n";prs
if(nUsers<1)
```

```
{
 st(hCom);
 //xorisXristi++;
 if(xorisXristi<20)
 {
  return;
 }
}
//#include "dimiourgiaOnomatos.h"
//ofstream out(onoma.c_str(),ios::out |
ios::binary);
//xn::SceneMetaData usmd;
//const XnLabel *userPM;
for (int i = 0; i < nUsers; ++i)
{
 if (g_bPrintID)
 {
  XnPoint3D com;
  g_UserGenerator.GetCoM(aUsers[i], com);

 g_DepthGenerator.ConvertRealWorldToProject
ive(1, &com, &com);
  xnOSMemSet(strLabel, 0, sizeof(strLabel));
  if (!g_bPrintState)
  {
   // Tracking
   sprintf(strLabel, "%d", aUsers[i]);
  }
  else if
(g_UserGenerator.GetSkeletonCap().IsTracking(
aUsers[i]))
  {
   // Tracking
   cout<<"\nTracking\n";
   sprintf(strLabel, "%d - Tracking", aUsers[i]);
  }
  else if
(g_UserGenerator.GetSkeletonCap().IsCalibratin
g(aUsers[i]))
  {
   // Calibrating
   sprintf(strLabel, "%d - Calibrating...",
aUsers[i]);
  }
  //else
  {

 //g_UserGenerator.GetSkeletonCap().StartTrack
ing(aUsers[i]);
   sprintf(strLabel, "%d - Looking for pose",
aUsers[i]);
   usersData[i].pos=com;
   usersData[i].ptimon=0;
   //xn::SceneMetaData usmd;
 g_UserGenerator.GetUserPixels(aUsers[i],usmd
);
   //const XnLabel *userPM=usmd.Data();
   userPM=usmd.Data();
   #include "paxosEkKaiEn.h"

 //cout<<"ptimon"<<usersData[i].ptimon<<"\n";
   //#include "apothikeysiDedomenonGiaNN.h"
   /*for(int m=0;m<usersData[i].ptimon;m++)
   {
    cout<<usersData[i].en[m]<<" ";
   }*/
   //out<<"\n\n";
   //#include"WW2WcW2c.h"
   //cout<<"i "<<i<<" x "<<com.X<<" y
"<<com.Y<<" z "<<com.Z<<"\n";
  }
  /*glColor4f(1-Colors[i%nColors][0],
1-Colors[i%nColors][1],
1-Colors[i%nColors][2], 1);
  glRasterPos2i(com.X, com.Y);

 glPrintString(GLUT_BITMAP_HELVETICA_
18, strLabel);*/
 }
 #include "DrawTheSkeleton.h"
}
```

Αρχειο DrawTheSkeleton.h

```
if (g_bDrawSkeleton &&
g_UserGenerator.GetSkeletonCap().IsTracking(a
Users[i]))
{
 xristis[aUsers[i]].traced=true;
 g_UserGenerator.GetSkeletonCap().GetSkeleto
nJointPosition(aUsers[i], XN_SKEL_HEAD,
xristis[aUsers[i]].uj);

 glBegin(GL_LINES);
 glColor4f(1-Colors[aUsers[i]%nColors][0],
1-Colors[aUsers[i]%nColors][1],
1-Colors[aUsers[i]%nColors][0], 1);
 DrawLimb(aUsers[i], XN_SKEL_HEAD,
XN_SKEL_NECK);

 DrawLimb(aUsers[i], XN_SKEL_NECK,
XN_SKEL_LEFT_SHOULDER);
 DrawLimb(aUsers[i],
XN_SKEL_LEFT_SHOULDER,
XN_SKEL_LEFT_ELBOW);
 DrawLimb(aUsers[i],
XN_SKEL_LEFT_ELBOW,
XN_SKEL_LEFT_HAND);
```

```
 DrawLimb(aUsers[i], XN_SKEL_NECK,
XN_SKEL_RIGHT_SHOULDER);
 DrawLimb(aUsers[i],
XN_SKEL_RIGHT_SHOULDER,
XN_SKEL_RIGHT_ELBOW);
 DrawLimb(aUsers[i],
XN_SKEL_RIGHT_ELBOW,
XN_SKEL_RIGHT_HAND);

 DrawLimb(aUsers[i],
XN_SKEL_LEFT_SHOULDER,
XN_SKEL_TORSO);
 DrawLimb(aUsers[i],
XN_SKEL_RIGHT_SHOULDER,
XN_SKEL_TORSO);

 DrawLimb(aUsers[i], XN_SKEL_TORSO,
XN_SKEL_LEFT_HIP);
 DrawLimb(aUsers[i], XN_SKEL_LEFT_HIP,
XN_SKEL_LEFT_KNEE);
 DrawLimb(aUsers[i],
XN_SKEL_LEFT_KNEE,
XN_SKEL_LEFT_FOOT);

 DrawLimb(aUsers[i], XN_SKEL_TORSO,
XN_SKEL_RIGHT_HIP);
 DrawLimb(aUsers[i], XN_SKEL_RIGHT_HIP,
XN_SKEL_RIGHT_KNEE);
 DrawLimb(aUsers[i],
XN_SKEL_RIGHT_KNEE,
XN_SKEL_RIGHT_FOOT);

 DrawLimb(aUsers[i], XN_SKEL_LEFT_HIP,
XN_SKEL_RIGHT_HIP);

 glEnd();//*/
}
else
{
 xristis[aUsers[i]].traced=false;
}
```

ΑρχειοKefaliSarosis.h

```
for (nY=0; nY<g_nYRes; nY++)
{
 for(int sar=0;sar<15;sar++)
 {
  mesoXroma.countx[sar]=0;
 }
 for (nX=0; nX < g_nXRes; nX++, nIndex++)
 {
  pDestImage[0] = 0;
  pDestImage[1] = 0;
  pDestImage[2] = 0;
  if (g_bDrawBackground || *pLabels != 0)
  {
     nValue = *pDepth;
     XnLabel label = *pLabels;
     XnUInt32 nColorID = label % nColors;
     if (label == 0)
     {
      nColorID = nColors;
     }
     if (nValue != 0)
     {
      nHistValue = g_pDepthHist[nValue];
      pDestImage[0] = nHistValue *
Colors[nColorID][0];
      pDestImage[1] = nHistValue *
Colors[nColorID][1];
      pDestImage[2] = nHistValue *
Colors[nColorID][2];
      if(nColorID!=10)
      {
```

Αρχειο scaler.h

```
for(int xr=0;xr<15;xr++)
{
  int sky=0;
  int skx1=0;
  int skx2=60;

  for(int say=0;say<64;say++)
  {
   bool noEm=false;
   for(int sax=0;sax<61;sax++)
   {

 if((portret[say][sax][0][xr]+portret[say][sax][1][
xr])!=0)
    {
     noEm=true;
     break;
    }
   }
   if (noEm==true)
   {
    sky=say;
    break;
   }
  }
  for(int sax=0;sax<61;sax++)
  {
   int sum=0;
   for(int say=0;say<64;say++)
   {
```

```
   sum+=portret[say][sax][0][xr];
   sum+=portret[say][sax][1][xr];
  }
  if(skx1==0)
  {
   if(sum!=0)
   {
    skx1=sax;
   }
  }
  else
  {
   if(sum==0)
   {
    skx2=sax;
    break;
   }
  }
 }

 int syt=64-sky;
 int sxt=skx2-skx1;

 if((syt*sxt)<175)
 {
  continue;
 }



 for(int dy=0;dy<60;dy++)
 {
  for(int dx=0;dx<60;dx++)
  {
   double klx=((double)(dx+1)/61)*sxt;
   double kly=(((double)(dy+1)/64)*syt);
   int dxc=max(floor(klx),1);
   //cout<<dxc;
   int dyc=max(floor(kly),1);
   double psX=dxc/klx;
   double psY=dyc/kly;
   dxc+=skx1;
   dyc+=sky;

 scaledPortret[dy][dx][xr]=portret[dyc][dxc][0][
xr]*psX*psY+portret[dyc+1][dxc][0][xr]*(1-ps
X)*psY+portret[dyc][dxc+1][0][xr]*psX*(1-psY
)+portret[dyc+1][dxc+1][0][xr]*(1-psX)*(1-psY)
;

 scaledPortret[dy][dx][xr]+=portret[dyc][dxc][1]
[xr]*psX*psY+portret[dyc+1][dxc][1][xr]*(1-ps
X)*psY+portret[dyc][dxc+1][1][xr]*psX*(1-psY
)+portret[dyc+1][dxc+1][1][xr]*(1-psX)*(1-psY)
;

    scaledPortret[dy][dx][xr]/=2;

 scaledPortret[dy][dx][xr]=max(scaledPortret[dy
][dx][xr],0);
   }
  }
}


Αρχειο  liner.h

cMetr=(cMetr+1)%5;
for(int xr=0;xr<15;xr++)
{
 const int thres=0;
 double megekt=-2;
 {
  int p2l=0;
  unsigned int faousa=0;
  for(int jy=0;jy<5;jy++)
  {
   for(int jx=0;jx<4;jx++)
   {
    for(int ky=0;ky<15;ky++)
    {
     for(int sar=0;sar<15;sar++)
     {

 linedPortret[p2l][xr]=scaledPortret[ky+(jy*16)]
[sar+(jx*16)][xr]-127;
      p2l++;

 faousa+=scaledPortret[ky+(jy*16)][sar+(jx*16)
][xr];
     }
    }
    //cout<<linedPortret[p2l-1][xr]<<" ";
   }
   //cout<<p2l<<" "<<faousa<<".,";
  }
  double eis[3600];
  double maxeis=-1;
  for(int i=0;i<3600;i++)
  {
   eis[i]=linedPortret[i][xr]/127;
   maxeis=max(maxeis,eis[i]);
  }
  netSim(diktio_mationYp,eis);
  //cout<<diktio_mationYp.outputl<<" noul
"<<diktio_mationYp.nol[diktio_mationYp.outpu
tl]<<" ";
  double meg=-1;
  int Ymat=0;
  for(int
mid=0;mid<diktio_mationYp.nol[diktio_mation
```

```
Yp.outputl];mid++)
   {

  //cout<<diktio_mationYp.nol[diktio_mationYp.
outputl]<<" ";

  //cout<<diktio_mationYp.lout[diktio_mationYp
.outputl][mid]<<" ";

  if(meg<diktio_mationYp.lout[diktio_mationYp.
outputl][mid])
    {

  meg=diktio_mationYp.lout[diktio_mationYp.ou
tputl][mid];
      Ymat=mid;
    }
   }
   //cout<<"Ymat:"<<Ymat<<" "<<meg<<"
"<<maxeis<<"\n";
   //cout<<diktio_mationYp.outputl<<" noul
"<<diktio_mationYp.nol[diktio_mationYp.outpu
tl]<<" ";
   //cout<<"ni"<<diktio_mationYp.ni<<" ";
   if(maxeis>-1)
   {
    cout<<"Ymat:"<<Ymat<<" "<<meg<<"
"<<maxeis<<"\n";
    tst1<<"YpsiEntopismou=[YpsiEntopismou
"<<Ymat<<"];\n";
    //cout<<faousa;
   }
   diktio_mat;
   double eyeCantArea[600];
   int ypsosmat=10;
   //cout<<"matia :";
   for(int jy=5;jy<35;jy++)
   {
    for(int sar=0;sar<600;sar++)
    {

  eyeCantArea[sar]=(scaledPortret[(sar%10)+jy][
sar/10][xr]-127)/127;
    }
    netSim(diktio_mation,eyeCantArea);
    double
ektmat=diktio_mation.lout[diktio_mation.outputl
][0]-diktio_mation.lout[diktio_mation.outputl][1]
;

  ektmat+=diktio_mationYp.lout[diktio_mationY
p.outputl][jy/2]*2;

  //meameans4efns[xr].lastV[jy-5][cMetr]=ektmat
;

   /*if(jy>10)
   {
    ektmat=0;
    for(int i=0;i<5;i++)
    {

ektmat+=meameans4efns[xr].lastV[jy-10][i];

ektmat+=meameans4efns[xr].lastV[jy-11][i];
    }
   }//*/
   //cout<<ektmat<<" ";
   if(megekt<ektmat)
   {
    megekt=ektmat;
    ypsosmat=jy;
    //cout<<ektmat<<" ";
   }
  }
  //cout<<"\n";
  if(megekt>thres)
  {
   for(int sar=0;sar<900;sar++)
   {

  scaledPortret[(sar/60)+60][sar%60][xr]=scaledP
ortret[(sar/60)+(ypsosmat)][sar%60][xr];
     //cout<<(sar/60)+60<<" ";
   }
  }
 }

 int p2l=0;
 unsigned int faousa=0;
 for(int jy=0;jy<5;jy++)
 {
  for(int jx=0;jx<4;jx++)
  {
   for(int ky=0;ky<15;ky++)
   {
    for(int sar=0;sar<15;sar++)
    {

  linedPortret[p2l][xr]=scaledPortret[ky+(jy*15)]
[sar+(jx*15)][xr]-127;
      p2l++;

  faousa+=scaledPortret[ky+(jy*15)][sar+(jx*15)
][xr];
    }
   }
   //cout<<linedPortret[p2l-1][xr]<<" ";
  }
  //cout<<p2l<<" "<<faousa<<".,";
 }
```

```
 //cout<<"\n";
 if((faousa>1)&&(megekt>thres))
 {
  double eis[4500];
  for(int i=0;i<4500;i++)
  {
   eis[i]=linedPortret[i][xr]/127;
  }
  netSim(diktio_mat,eis);
  for(int
mid=0;mid<diktio_mat.nol[diktio_mat.outputl];
mid++)
  {

 netPuser[mid][xr]=diktio_mat.lout[diktio_mat.o
utputl][mid];
  }


  tst1<<"a=[";
  for(int i=0;i<4500;i++)
  {
   tst1<<eis[i]<<" ";
  }
  tst1<<"\n];\nlined=[lined;a];\n";
  tst1<<"ApokrisiNetYm=[ApokrisiNetYm;";
  for(int
mid=0;mid<diktio_mationYp.nol[diktio_mation
Yp.outputl];mid++)
  {

 tst1<<diktio_mationYp.lout[diktio_mationYp.o
utputl][mid]<<" ";
  }
  tst1<<"];\n";
  tst1<<"ApokrisiNetAnP=[ApokrisiNetAnP;";
  for(int
mid=0;mid<diktio_mat.nol[diktio_mat.outputl];
mid++)
  {

 tst1<<diktio_mat.lout[diktio_mat.outputl][mid]
<<" ";
  }
  tst1<<"];\n";


 }
}
```

ΑρχειοpaxosEkKaiEn.h

```
for(unsigned int
say=0;say<usmd.YRes();say+=15)
{
 usersData[i].paxos[usersData[i].ptimon]=0;
 usersData[i].en[usersData[i].ptimon]=0;
 usersData[i].ek[usersData[i].ptimon]=0;
 bool einai=false;
 for(unsigned int
sax=0;sax<usmd.XRes();sax+=10)
 {

 if(userPM[(say*usmd.XRes())+sax]==aUsers[i]
)
  {
   usersData[i].paxos[usersData[i].ptimon]++;
   if(usersData[i].ek[usersData[i].ptimon]==0)
   {
    usersData[i].ek[usersData[i].ptimon]=sax;
   }
   if(einai==false)
   {
    usersData[i].en[usersData[i].ptimon]++;
   }
   einai=true;
  }
  else
  {
   if(einai==true)
   {
    usersData[i].en[usersData[i].ptimon]++;
   }
   einai=false;
  }
  //out<<userPM[(say*usmd.XRes())+sax]<<" ";
 }
 usersData[i].ptimon++;
 //out<<"\n";
}
```

Αρχειο dravUser.h

```
nIndex = 0;
pDestImage = pDepthTexBuf;
pDepth = imMd.Data();
pLabels = smd.Data();
idat = dmd.Data();
// Prepare the texture map
/*if(xristis.traced==true &&
xristis.uj.position.Z>0 && xristis.daded==true)
{
 cu++;
 //tst1<<"img"<<cu<<"=[];\r\n";
 xristis.daded=false;
}//*/
  //#include "KefaliSarosis.h"
```

```
for (nY=0; nY<g_nYRes; nY++)
{
 for(int sar=0;sar<15;sar++)
 {
  mesoXroma.countx[sar]=0;
 }
 for (nX=0; nX < g_nXRes; nX++, nIndex++)
 {
  pDestImage[0] = 0;
  pDestImage[1] = 0;
  pDestImage[2] = 0;


  if (g_bDrawBackground || *pLabels != 0)
  {




      nValue = *pDepth;
      XnLabel label = *pLabels;
      XnUInt32 nColorID = label % nColors;
      if (label == 0)
      {
       nColorID = nColors;
      }
      if (nValue != 0)
      {

       nHistValue = g_pDepthHist[nValue];
       pDestImage[0] = nHistValue *
Colors[nColorID][0];
       pDestImage[1] = nHistValue *
Colors[nColorID][1];
       pDestImage[2] = nHistValue *
Colors[nColorID][2];
       if((nX<30)&&(nY<37))
       {

 pDestImage[0]=scaledPortret[nY*2][nX*2][1];/
/aUsers[label-1]];
        //cout<<scaledPortret[nX][nY][label];

 pDestImage[1]=scaledPortret[nY*2][nX*2][1];/
/aUsers[label-1]];

 pDestImage[2]=scaledPortret[nY*2][nX*2][1];/
/aUsers[label-1]];//*/
        /*if(scaledPortret[nY*2][nX*2][1]>64)
        {
         pDestImage[2]=255;
        }//*/
       }
       else if((nX<60)&& (nY<37))
       {

 pDestImage[0]=scaledPortret[(nY%37)*2][(nX
%30)*2][2];//aUsers[label-1]];
        //cout<<scaledPortret[nX][nY][label];

 pDestImage[1]=scaledPortret[(nY%37)*2][(nX
%30)*2][2];//aUsers[label-1]];

 pDestImage[2]=scaledPortret[(nY%37)*2][(nX
%30)*2][2];//aUsers[label-1]];

 /*if(scaledPortret[(nY%37)*2][(nX%30)*2][2]
>255)
        {
         pDestImage[2]=255;
        }//*/
       }
       else if((nX<90)&& (nY<37))
       {

 pDestImage[0]=scaledPortret[(nY%37)*2][(nX
%30)*2][3];//aUsers[label-1]];

 //cout<<scaledPortret[(nX%30)][nY][label];

 pDestImage[1]=scaledPortret[(nY%37)*2][(nX
%30)*2][3];//aUsers[label-1]];

 pDestImage[2]=scaledPortret[(nY%37)*2][(nX
%30)*2][3];//aUsers[label-1]];
       }
       else if((nX<120)&& (nY<37))
       {

 pDestImage[0]=scaledPortret[(nY%37)*2][(nX
%30)*2][4];//aUsers[label-1]];

 //cout<<scaledPortret[(nX%30)][nY][label];

 pDestImage[1]=scaledPortret[(nY%37)*2][(nX
%30)*2][4];//aUsers[label-1]];

 pDestImage[2]=scaledPortret[(nY%37)*2][(nX
%30)*2][4];//aUsers[label-1]];
       }
       else if((nX<150)&& (nY<37))
       {

 pDestImage[0]=scaledPortret[(nY%37)*2][(nX
%30)*2][5];//aUsers[label-1]];

 //cout<<scaledPortret[(nX%30)][nY][label];

 pDestImage[1]=scaledPortret[(nY%37)*2][(nX
%30)*2][5];//aUsers[label-1]];
```

```
 pDestImage[2]=scaledPortret[(nY%37)*2][(nX
%30)*2][5];//aUsers[label-1]];
        }
        else if((nX<180)&& (nY<37))
        {

 pDestImage[0]=scaledPortret[(nY%37)*2][(nX
%30)*2][6];//aUsers[label-1]];

 //cout<<scaledPortret[(nX%30)][nY][label];

 pDestImage[1]=scaledPortret[(nY%37)*2][(nX
%30)*2][6];//aUsers[label-1]];

 pDestImage[2]=scaledPortret[(nY%37)*2][(nX
%30)*2][6];//aUsers[label-1]];
        }
        else if((nX<210)&& (nY<37))
        {

 pDestImage[0]=scaledPortret[(nY%37)*2][(nX
%30)*2][7];//aUsers[label-1]];

 //cout<<scaledPortret[(nX%30)][nY][label];

 pDestImage[1]=scaledPortret[(nY%37)*2][(nX
%30)*2][7];//aUsers[label-1]];

 pDestImage[2]=scaledPortret[(nY%37)*2][(nX
%30)*2][7];//aUsers[label-1]];
        }
        if(nColorID!=10)
        {
       /*if(mesoXroma.paron[label]>3000 &&
mesoXroma.paron[label]<4500 &&
mesoXroma.countx[label]>30 &&
mesoXroma.countx[label]<90)
       {
        pDestImage[0] = (unsigned
char)(mesoXroma.xromataMes[label][0][0]/12)*
12;
        pDestImage[1] = (unsigned
char)(mesoXroma.xromataMes[label][1][0]/12)*
12;
        pDestImage[2] = (unsigned
char)(mesoXroma.xromataMes[label][2][0]/12)*
12;//*/
        /*
        pDestImage[0] =
epx[label][0][0]*XANAL;//(unsigned
char)mesoXroma.xromata[label][0];
        pDestImage[1] =
epx[label][1][0]*XANAL;//(unsigned
char)mesoXroma.xromata[label][1];
        pDestImage[2] =
epx[label][2][0]*XANAL;//(unsigned
char)mesoXroma.xromata[label][2];//*/
      /* }
       else if(mesoXroma.paron[label]>7000 &&
mesoXroma.paron[label]<8500 &&
mesoXroma.countx[label]>30 &&
mesoXroma.countx[label]<90)
       {
        pDestImage[0] = (unsigned
char)(mesoXroma.xromataMes[label][0][1]/12)*
12;
        pDestImage[1] = (unsigned
char)(mesoXroma.xromataMes[label][1][1]/12)*
12;
        pDestImage[2] = (unsigned
char)(mesoXroma.xromataMes[label][2][1]/12)*
12;
        /*
        pDestImage[0] =
epx[label][0]*XANAL;//(unsigned
char)mesoXroma.xromata[label][0];
        pDestImage[1] =
epx[label][1]*XANAL;//(unsigned
char)mesoXroma.xromata[label][1];
        pDestImage[2] =
epx[label][2]*XANAL;//(unsigned
char)mesoXroma.xromata[label][2];//*/
      /* }
       else//*/
       if((nX<60)&&(nY<75))
       {
       }
       if(nY>250)
       {
       }
       else
       {
        if(xristis[label].traced==true)
        {
         XnPoint3D pt=xristis[label].uj.position;

 g_DepthGenerator.ConvertRealWorldToProject
ive(1, &pt, &pt);
         outkef<<pt.X<<","<<pt.Y<<";";
         if(abs(pt.X-nX)<30 &&
abs(pt.Y-nY+5)<30)
         {
          for(int col=0;col<3;col++)
          {
           int xk=nX-floor(pt.X)+30;
           int yk=nY-floor(pt.Y)+28;
           portret[yk][xk][col][label]=idat[col];
          }
          pDestImage[0]=idat[0];
```

```
          pDestImage[1]=idat[1];
          pDestImage[2]=idat[2];
          /*tst1<<"c(:,:,1)="<<(int)idat[0]<<";\r\n";
          tst1<<"c(:,:,2)="<<(int)idat[1]<<";\r\n";
          tst1<<"c(:,:,3)="<<(int)idat[2]<<";\r\n";

 tst1<<"img("<<nY-floor(pt.Y)+34<<","<<nX-fl
oor(pt.X)+31<<",:,"<<cu<<")=uint8(c);\r\n";*/
          xristis[label].daded=true;
          //tst1<<"line=[line,c];\r\n";
         }
         else
         {
          pDestImage[0]=50;
          pDestImage[1]=idat[1];
          pDestImage[2]=idat[2];
         }
        }
        else
        {
         pDestImage[0]=100;
         pDestImage[1]=idat[1];
         pDestImage[2]=idat[2];
        }
       }

      mesoXroma.paron[label]++;
      mesoXroma.countx[label]++;

     }
    }
   }
   pDepth++;
   pLabels++;
   idat+=3;
   pDestImage+=3;
  }
  pDestImage += (texWidth - g_nXRes) *3;
}//*/
```

Αρχειο DeclAndSetMX.h

```
struct
{
 unsigned int xromataEp[15][3][256][2];
 unsigned int xromataMes[15][3][2];
 int paron[15];
 int countx[15];
 int maxcountx[15];
 int parong[15];
}mesoXroma;
for(int i=0;i<2;i++)
{
 for(int sar=0;sar<15;sar++)
 {
  for(int sar2=0;sar2<256;sar2++)
  {
   mesoXroma.xromataEp[sar][0][sar2][i]=0;
   mesoXroma.xromataEp[sar][1][sar2][i]=0;
   mesoXroma.xromataEp[sar][2][sar2][i]=0;
  }
  mesoXroma.xromataMes[sar][0][i]=0;
  mesoXroma.xromataMes[sar][1][i]=0;
  mesoXroma.xromataMes[sar][2][i]=0;
  mesoXroma.paron[sar]=1;
  mesoXroma.parong[sar]=1;
 }
}
```

Παράρτημα Γ
Πλήρης πηγαίος κώδικας matlab

Αρχειο Disdim2lineDiamerismena.m

```
for i=1:size(fk{1},3)
   temp=[];
   for jy=0:3
```

```
    for jx=0:3
        for ky=1:15
            temp=[temp
fk{1}(ky+(jy*15),(1:15)+(jx*15),i)];
        end
    end
  end
  fk{3}(:,i)=temp;
end
```

Αρχειο Cscale.m

```
ShowAll(imgrsug)
imgCSc=uint8([]);
imgCSc2=uint8([]);
tarS=[];
CScounter=1;
for i=1:size(imgrsug,3)
  zeroLine=sum(imgrsug(:,:,i)');
  simKop=1;
  for j=1:size(imgrsug,1)
    if(zeroLine(j)~=0)
      simKop=j;
      break;
    end
  end
  imtemp=imgrsug(simKop:size(imgrsug,2),:,i);
  zeroLine=sum(imtemp);
  simKop=[1 size(imtemp,2)];
  for j=1:size(imtemp,2)
    if(zeroLine(j)~=0)
      simKop(1)=j;
      break;
    end
  end
  zeroLine=zeroLine(end:-1:1);
  for j=1:size(imtemp,2)
    if(zeroLine(j)~=0)
      simKop(2)=simKop(2)-j;
      break;
    end
  end
  imtemp=imtemp(:,simKop(1):simKop(2));
  SUBPLOT(2,1,1);
  imshow(imtemp);
  SUBPLOT(2,1,2);
  imshow(imgrsug(:,:,i));
  pause(0.03);
  [sxt syt]=size(imtemp);
  if((sxt*syt)<180)
    continue
  end
  if(((sum(sum(imtemp))/(sxt*syt))<36)&&
i>800)
    continue
  end
  for dx=1:(size(imgrsug,1)-1)
    for dy=1:(size(imgrsug,2)-1)

dxc=max(fix(dx/(size(imgrsug,1)/sxt)),1);

dyc=max(fix(dy/(size(imgrsug,2)/syt)),1);
      psX=dxc/(dx/(size(imgrsug,1)/sxt));
      psY=dyc/(dy/(size(imgrsug,2)/syt));

imgCSc(dx,dy,CScounter)=imtemp(dxc,dyc)*ps
X*psY+imtemp(dxc+1,dyc)*(1-psX)*psY+imte
mp(dxc,dyc+1)*psX*(1-psY)+imtemp(dxc+1,dy
c+1)*(1-psX)*(1-psY);

%imgCSc2(dx,dy,CScounter)=imtemp(dxc,dyc);
    end
  end
  tarS(:,CScounter)=tar(:,i);
  CScounter=CScounter+1;
end
SUBPLOT(1,1,1);
ShowAll(imgCSc)
```

ΑρχειοcustomNN4myeyenewff.m

```
%euroseis1=[-1 1;-1 1;-1 1;-1 1];
%euroseis2=[-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1
1];
cunet=network;
cunet.numInputs=6;
cunet.numLayers=5;
cunet.biasConnect=ones(5,1);
z6=zeros(1,6);
z5=zeros(1,5);
o6=ones(1,6);
o4=ones(1,4);
cunet.inputConnect=[o4 0 0;0 o4 0;0 0 o4;o6;z6];
cunet.layerConnect=[z5;z5;z5;z5;1 1 1 1 0];
cunet.outputConnect=[0 0 0 0 1];
for dik=1:6
```

```
cunet.inputs{dik}.exampleInput=[(ones(100,1)*-
1) ones(100,1)];
end
%cunet.inputs{1}.processFcns =
{'removeconstantrows'};
%cunet.inputs{2}.processFcns =
{'removeconstantrows'};
%cunet.layers{1}.size=5;
%cunet.layers{2}.size=5;
%cunet.layers{3}.size=3;
for i=1:3
   cunet.layers{i}.size=7;
   cunet.layers{i}.transferFcn='tansig';
   cunet.layers{i}.initFcn='initnw';
end
cunet.layers{4}.size=9;
cunet.layers{4}.transferFcn='tansig';
cunet.layers{4}.initFcn='initnw';
cunet.layers{5}.size=2;
cunet.layers{5}.transferFcn='tansig';
cunet.layers{5}.initFcn='initnw';
cunet.adaptFcn='trains';
cunet.initFcn='initlay';
cunet.performFcn='mse';
cunet.trainFcn='traingd';
%cunet.divideFcn='dividerand';
cunet.plotFcns={'plotperform','plottrainstate'};
cunet=init(cunet);
cunet.trainParam.epochs=18000
```

```
cunet.inputs{dik}.exampleInput=[(ones(100,1)*-
1) ones(100,1)];
end
%cunet.inputs{1}.processFcns =
{'removeconstantrows'};
%cunet.inputs{2}.processFcns =
{'removeconstantrows'};
%cunet.layers{1}.size=5;
%cunet.layers{2}.size=5;
%cunet.layers{3}.size=3;
for i=1:3
   cunet.layers{i}.size=40;
   cunet.layers{i}.transferFcn='tansig';
   cunet.layers{i}.initFcn='initnw';
end
cunet.layers{4}.size=3;
cunet.layers{4}.transferFcn='tansig';
cunet.layers{4}.initFcn='initnw';
for i=5:7
   cunet.layers{i}.size=2;
   cunet.layers{i}.transferFcn='tansig';
   cunet.layers{i}.initFcn='initnw';
end
cunet.adaptFcn='trains';
cunet.initFcn='initlay';
cunet.performFcn='mse';
cunet.trainFcn='traingd';
%cunet.divideFcn='dividerand';
cunet.plotFcns={'plotperform','plottrainstate'};
cunet=init(cunet);
cunet.trainParam.epochs=35000
```

Αρχειοcustom NN4myeyenewffst.m

```
%euroseis1=[-1 1;-1 1;-1 1;-1 1];
%euroseis2=[-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1
1];
cunet=network;
cunet.numInputs=6;
cunet.numLayers=7;
cunet.biasConnect=ones(7,1);
z6=zeros(1,6);
z5=zeros(1,5);
o6=ones(1,6);
o4=ones(1,4);
cunet.inputConnect=[o4 0 0;0 o4 0;0 0
o4;o6;z6;z6;z6];
cunet.layerConnect=[z5 0 0;z5 0 0;z5 0 0;z5 0 0;1
0 0 1 0 0 0;0 1 0 1 0 0 0;0 0 1 1 0 0 0];
cunet.outputConnect=[0 0 0 0 1 1 1];
for dik=1:6
```

Αρχειοmatiafinter.m

```
subplot(1,1,1)
ypsosmation=[];
mc=1;
ikprt=ik;
for i=1:2400
   [i2 re]=imcrop(ikprt(:,:,i));
   simk=fix(re(2))-5
   if(simk>10)

ypsosmation(:,:,mc)=ikprt(simk:(simk+9),:,i);
      mc=mc+1;
      ikprt(simk:(simk+9),:,i)=ones(10,60)*255;
   end
   imshow(ikprt(:,:,i));
   pause(0.8)
end
ikprt=ik;
```

```
subplot(1,1,1)
for i=1:600
   energ=-2;
   point=0;
   for j=5:50
      netin=reshape(ikprt(j:(j+9),:,i),[1 600]);

apot=sim(netraind,(double(netin')-128)/127);
      dif=apot(1)-apot(2);
      if(energ<dif)
         energ=dif;
         point=j;
      end
   end
   if(energ>(1.9))
      point;
      ikprt((point:point+6),:,i)=ones(7,60)*255;
   end
   imshow(ikprt(:,:,i));
   pause(0.01)
end
```

ΑρχειοmatiafinterYpXs.m

```
subplot(1,1,1)
%ypsosmation=[];
%simiakopis=[];
%mc=1;
%ikprt=ik;
%for i=1:2400
%    [i2 re]=imcrop(ikprt(:,:,i));
%    simk=fix(re(2))-5
%    if(simk>10)
%
ypsosmation(:,:,mc)=ikprt(simk:(simk+9),:,i);
%       if mod(mc,10)==1
%          save('ypsosmation.mat','ypsosmation');
%       end
%       mc=mc+1;
%       simiakopis=[simiakopis [simk;i]];
%       ikprt(simk:(simk+9),:,i)=ones(10,60)*255;
%    end
%    imshow(ikprt(:,:,i));
%    pause(0.8)
%end
ikprt=ik;
subplot(1,1,1)
zakpn=1;
zakpo=1;
for
i=1:2200%[280:292,640:664,865:935,1519:1523
,1690:1694,1911:1926]
   energ=-2;
   point=0;
   temp=[];
   for jy=0:3
      for jx=0:3
         for ky=1:15
            temp=[temp
ikprt(ky+(jy*15),(1:15)+(jx*15),i)];
         end
      end
   end
   tmp=((double(temp)-128)/127);
   apot=
tansig((tansig((tmp*network7.IW{1}')+network7
.b{1}')*network7.LW{2,1}')+network7.b{2}')';
   apot=apot(1:2:24)+apot(2:2:25);
   for w=1:12
      if ((apot(w)-(w*0.3))>energ)
         energ=apot(w);
         point=w*4;
      end
   end

if(energ>-0.5)%(energ>(0.5))&&((energ<(1.7))))
      point;
      ikprt((point:point+9),:,i)=ones(10,60)*255;
      i
      %[i2 re]=imcrop(ikprt(:,:,i));
      %if (re(2)<40)
      %    if(re(1)<30)
      %
epipleonN(:,:,zakpn)=ik((point:point+9),:,i);
      %       zakpn=zakpn+1
      %    else
      %
epipleonO(:,:,zakpo)=ik((point:point+9),:,i);
      %       zakpo=zakpo+1
      %    end
      %end

   end

   imshow(ikprt(:,:,i));
   pause(0.02)

end
```

ΑρχειοSaveNetBin.m

```
function  SaveNetBin( net,file )
%UNTITLED Summary of this function goes
here
%   Detailed explanation goes here
fid=fopen(file,'w');
```

```
%layer struct
fwrite(fid,net.numLayers,'int');
for i=1:net.numLayers
   fwrite(fid,net.layers{i}.size,'int');
   fwrite(fid,net.biasConnect(i),'int');
end
for i=1:net.numLayers
   for j=1:net.numLayers
      fwrite(fid,net.layerConnect(i,j),'int');
   end
end
%input struct
fwrite(fid,net.numInputs,'int');
for i=1:net.numInputs
   fwrite(fid,net.inputs{i}.size,'int');
end
for i=1:net.numLayers
   for j=1:net.numInputs
      fwrite(fid,net.inputConnect(i,j),'int');
   end
end
%b
for i=1:net.numLayers
   fwrite(fid,net.b{i},'real*8');
end
%iw
for i=1:net.numLayers
   for j=1:net.numInputs
      %net.iw{i,j}
      fwrite(fid,net.iw{i,j},'real*8');
   end
end
%lw
for i=1:net.numLayers
   for j=1:net.numLayers
      %net.lw{i,j}
      fwrite(fid,net.lw{i,j},'real*8');
   end
end
fclose(fid);

end
```

Αρχειο
customNN4myicnewffmeEisodoGiaMatia.m

```
%euroseis1=[-1 1;-1 1;-1 1;-1 1];
%euroseis2=[-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1 1;-1
1];
cunet=network;
cunet.numInputs=17;
cunet.numLayers=9;
cunet.biasConnect=ones(9,1);
z8=zeros(1,8);
z9=zeros(1,9);
o8=ones(1,8);
z4=zeros(1,4);
cunet.inputConnect=[o8 z8 0;z4 o8 z4 0;z8 o8 0;1
1 0 0 1 1 0 0 z9;0 0 1 1 0 0 1 1 z9;z8 1 1 0 0 1 1 0
0 0;z8 0 0 1 1 0 0 1 1 0;zeros(1,16) 1;zeros(1,17)];
cunet.layerConnect=[z9;z9;z9;z9;z9;z9;z9;z9;o8
0];
cunet.outputConnect=[0 0 0 0 0 0 0 0 1];
for dik=1:16
   cunet.inputs{dik}.exampleInput=[zeros(225,1)
ones(225,1)*255];
end
cunet.inputs{17}.exampleInput=[zeros(900,1)
ones(900,1)*255];
%cunet.inputs{1}.processFcns =
{'removeconstantrows'};
%cunet.inputs{2}.processFcns =
{'removeconstantrows'};
%cunet.layers{1}.size=5;
%cunet.layers{2}.size=5;
%cunet.layers{3}.size=3;
for i=1:3
   cunet.layers{i}.size=10;
   cunet.layers{i}.transferFcn='tansig';
   cunet.layers{i}.initFcn='initnw';
end
for i=4:7
   cunet.layers{i}.size=10;
   cunet.layers{i}.transferFcn='tansig';
   cunet.layers{i}.initFcn='initnw';
end
cunet.layers{8}.size=40;
cunet.layers{8}.transferFcn='tansig';
cunet.layers{8}.initFcn='initnw';

cunet.layers{9}.size=7;
cunet.layers{9}.transferFcn='tansig';
cunet.adaptFcn='trains';
cunet.initFcn='initlay';
cunet.performFcn='mse';
cunet.trainFcn='traingd';
%cunet.divideFcn='dividerand';
cunet.plotFcns={'plotperform','plottrainstate'};
cunet=init(cunet);
cunet.trainParam.epochs=350000
```

Παράρτημα Δ
Assemply μικροελενκτη πλακετας ελενχου

```
.device at90s8515

.def sendareastart =r0 ;ΚΑΤΑΧΩΡΙΤΗΣ ΠΟΥ ΔΕΙΧΝΕΙ ΑΠΟ ΠΙΟ ΣΗΜΕΙΟ ΤΗΣ ΜΝΗΜΗΣ ΘΑ ΞΕΚΙΝΗΣΕΙ Η ΑΠΟΣΤΟΛΗ
.def roadcounter1r =r1 ;Ο ΛΙΓΟΤΕΡΟ ΣΗΜΑΝΤΙΚΟΣ ΑΠΟ ΤΗΝ ΟΜΑΔΑ ΚΑΤΑΧΩΡΙΤΩΝ ΠΟΥ ΔΕΙΧΝΟΥΝ ΤΗΝ ΜΕΤΑΤΟΠΙΣΗ
.def roadcounter2r =r2 ;ΤΗΣ ΔΕΞΙΑΣ ΕΡΠΙΣΤΡΙΑΣ
.def roadcounter3r =r3
.def roadcounter4r =r4
.def roadcounter1l =r5 ;Ο ΛΙΓΟΤΕΡΟ ΣΗΜΑΝΤΙΚΟΣ ΑΠΟ ΤΗΝ ΟΜΑΔΑ ΚΑΤΑΧΩΡΙΤΩΝ ΠΟΥ ΔΕΙΧΝΟΥΝ ΤΗΝ ΜΕΤΑΤΟΠΙΣΗ
.def roadcounter2l =r6 ;ΤΗΣ ΑΡΙΣΤΕΡΗΣ ΕΡΠΙΣΤΡΙΑΣ
.def roadcounter3l =r7
.def roadcounter4l =r8
.def speedcalcrate=r9 ;ΚΑΤΑΧΩΡΙΤΗΣ ΠΟΥ ΕΧΕΙ ΤΟ ΧΡΟΝΙΚΟ ΔΙΑΣΤΗΜΑ ΠΟΥ ΘΑ ΚΑΘΑΡΙΖΕΤΑΙ Ο
        ;ΒΟΗΘΙΤΙΚΟΣ ΚΑΤΑΧΩΡΙΤΗΣ ΜΕΤΡΗΣΗΣ ΤΑΧΥΤΙΤΑΣ
.def timecounter3 =r10
.def timecounter4 =r11
.def portbstate =r12
.def dxregister =r13

.def speedl =r14
.def speedr =r15
.def wandetspeedl=r16
.def wandetspeedr =r17
.def counterl =r18
.def counterr =r19
.def timecounter =r20
.def timecounter2 =r21
.def flagsreg =r22
.def timeoutreg =r23
.def temp =r24
.def temp2 =r25
.def xlow =r26
.def xhigh =r27
.def ylow =r28
.def yhigh =r29
.def zlow =r30
.def zhigh =r31

;ΣΥΜΒΟΛΙΚΑ ΘΕΣΕΩΝ ΜΝΗΜΗΣ
.equ splow =$3d
.equ sphigh =$3e
.equ startofresbuff =$0060

.equ bufferusage =$0066
.equ bufferpointer =$0067
.equ pcreadatfun =$0068
.equ isbufferemp4pc =$0069
.equ lastbit4qb =$006a
.equ resbuffpointl =$006b
.equ resbuffpointh =$006c
.equ arccounterlow =$006e
.equ arccounterhigh=arccounterlow+1
.equ flagsreg2 =$0070
.equ arxipinaka =$0260
.equ MemoryTop =$7fff

;ΣΥΜΒΟΛΙΚΑ ΣΤΑΘΕΡΩΝ

.equ maxofarcounter=11938 ;ΑΡΧΙΚΑ ΥΠΟΛΟΓΙΣΘΗΣΑ ΤΙΜΗ 11936
.equ aristera =$68
.equ empros =$70
.equ deksia =$90
.equ piso =$88

;flagsreg bits
;0 = baddatapack
;1 = ΑΥΤΟΜΑΤΙ ΡΥΘΜΙΣΙ ΤΑΧΥΤΙΤΑΣ
;2 = roadcounterr overflow
;3 = roadcounterr underflow
;4 = roadcounterl overflow
;5 = roadcounterl underflow
;6 = Ο ΥΠΟΛΟΓΙΣΤΗΣ ΘΕΛΕΙ ΑΠΑΝΤΗΣΗ
;7 = QB ΣΥΜΒΑΤΗ ΑΠΟΣΤΟΛΗ ΔΕΔΟΜΕΝΟΝ

.include "macros.mac"
.include "movmacros.mac"
.include "logicmacros.mac"
.include "macros_topic.mac"


.cseg


rjmp main
rjmp interap0
rjmp interap1
rjmp interap2
.org $0006
rjmp time

.org $0009
rjmp Recive
.include "Send_isr.inc"
```

```
.include "Recive_isr.inc"



interap0:
enter2isr
inc  counterl
rcallwhatrotateleft
brlo endrap0
brnepisol
;ΥΠΟΛΟΓΙΣΜΟΣ ΓΩΝΙΑΣ ΟΧΙΜΑΤΟΣ
ldiwzhigh,zlow,arccounterlow
rcalldecarccounter

baddi roadcounter1l,1
ldi  temp,0
adc  roadcounter2l,temp
adc  roadcounter3l,temp
adc  roadcounter4l,temp
brccendrap0
ori   flagsreg,0b00010000
rjmp exitfromisraddres
pisol:
;ΥΠΟΛΟΓΙΣΜΟΣ ΓΩΝΙΑΣ ΟΧΙΜΑΤΟΣ
ldiwzhigh,zlow,arccounterlow
rcallincarccounter

bsubiroadcounter1l,1
ldi  temp,0
sbc  roadcounter2l,temp
sbc  roadcounter3l,temp
sbc  roadcounter4l,temp
brccendrap0
ori   flagsreg,0b00100000
endrap0:
rjmp exitfromisraddres



interap1:
enter2isr
inc  counterr
rcallwhatrotateright
brlo endrap1
brnepisokinisi
;ΥΠΟΛΟΓΙΣΜΟΣ ΓΩΝΙΑΣ ΟΧΙΜΑΤΟΣ
ldiwzhigh,zlow,arccounterlow
rcallincarccounter

baddi roadcounter1r,1
ldi  temp,0
adc  roadcounter2r,temp
adc  roadcounter3r,temp
adc  roadcounter4r,temp
brccendrap1
ori   flagsreg,0b00000100
rjmp exitfromisraddres
pisokinisi:
;ΥΠΟΛΟΓΙΣΜΟΣ ΓΩΝΙΑΣ ΟΧΙΜΑΤΟΣ
ldiwzhigh,zlow,arccounterlow
rcalldecarccounter

bsubiroadcounter1r,1
ldi  temp,0
sbc  roadcounter2r,temp
sbc  roadcounter3r,temp
sbc  roadcounter4r,temp
brccendrap1
ori   flagsreg,0b00001000
endrap1:
rjmp exitfromisraddres


interap2:
enter2isr
rjmp exitfromisraddres


time:
enter2isr

djnztimecounter2,nexttimearea
ldi  timecounter2,101
;ΕΚΤΕΛΕΙΤΑΙ ΚΑΘΕ 13,973ms

sbi   $18,2
in   portbstate,$16
sbrs portbstate,0
cbi   $18,2

sbrs flagsreg,6       ;ΕΛΕΝΧΟΣ ΑΝ Ο
ΥΠΟΛΟΓΙΣΤΗΣ ΔΕΝ ΕΣΤΗΛΕ ΣΗΜΑ
rjmp noresend       ;ΠΑΡΑΛΑΒΗΣ.
djnztimeoutreg,nexttimearea   ;ΑΝ ΣΤΑΛΟΥΝ
ΔΕΔΟΜΕΝΑ ΣΤΟΝ ΥΠΟΛΟΓΙΣΤΗ
bldi timeoutreg,30      ;ΚΑΙ ΑΥΤΟΣ ΔΕΝ
ΑΠΑΝΤΗΣΕΙ ΣΕ 420ms
rcallresend         ;ΕΚΙΝΗΤΑΙ ΔΙΑΔΙΚΑΣΙΑ
ΕΠΑΝΑΠΟΣΤΟΛΗΣ
noresend:

djnztimecounter3,nexttimearea
bldi timecounter3,8
;ΕΚΤΕΛΕΙΤΑΙ ΚΑΘΕ 111,784ms


;LedComander


djnztimecounter4,nexttimearea
bldi timecounter4,5
```

```
;ΕΚΤΕΛΕΙΤΑΙ ΚΑΘΕ 558,92ms


;LedComander2


nexttimearea:
djnztimecounter,exitfromisraddres
mov timecounter,speedcalcrate

mov speedl,counterl
clr counterl

mov speedr,counterr
clr counterr

sbrs flagsreg,1
rjmp notauto
cp speedl,wandetspeedl
breqleftok
in temp,$28
rcallfixpower
out $28,temp
leftok:
cp speedr,wandetspeedr
breqnotauto
in temp,$2a
rcallfixpower
out $2a,temp
notauto:




exitfromisraddres:
exitfromisr



main:
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΣΤΟΙΒΑΣ
;ΜΕΓΙΣΤΟ ΜΕΓΕΘΟΣ ΣΤΟΙΒΑΣ 11 from isr +
13 ΑΠΟ ΒΑΣΙΚΟ ΠΡΟΓΡΑΜΑ 24
outiw sphigh,splow,$025f
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΩΝ 6 ΤΕΛΕΥΤΕΩΝ PIN
ΤΗΣ ΠΟΡΤΑΣ Β ΣΕ ΕΞΟΔΟ
outi $17,$dc
outi $18,$d8
;ΕΛΕΝΧΟΣ EXPANSIONBOARD
outi $11,$c0
outi $12,$80
dnop
in temp,$10
andi temp,$c0
cpi temp,$80
outi $11,$00
brnepc
outi $11,$c0
ldiwzhigh,zlow,$003a
rcallporttest
outi $1a,$00
brnepc
ldiwzhigh,zlow,$0034
rcallporttest
outi $14,$00
brnepc
rcallledFlip
;ΑΝΕΡΧΟΜΕΝΟ ΜΕΤΟΠΟ ΠΡΟΚΑΛΕΙ
ΙΝΤΕΡΑΡΤ ΚΑΙ Η ΕΞΩΤΕΡΗ RAM ΕΝΕΡΓΗ
outi $35,$0f
;MEMORY TEST
rjmp MemTestEnd;ΠΑΡΑΒΛΕΨΗ MEM TEST

ldi temp2,$00
rcallFillMemWith
ldiwzhigh,zlow,$0260
MemtestLoop1:
ld temp,z+
cpi temp,$00
brnepc
cpiw zhigh,zlow,MemoryTop+1
brneMemtestLoop1

rcallledFlip

ldi temp2,$ff
rcallFillMemWith
ldiwzhigh,zlow,$0260
MemtestLoop2:
ld temp,z+
cpi temp,$ff
brnepc
cpiw zhigh,zlow,MemoryTop+1
brneMemtestLoop2

rcallledFlip

ldiwzhigh,zlow,$0260
;ΠΡΟΣΟΡΙΝΗ ΧΡΗΣΗ ΤΟΥ ΚΑΤΧΩΡΙΤΗ Υ
ΓΙΑ ΑΛΛΗ ΧΡΗΣΗ ΑΠΟ ΤΗΝ ΚΑΝΟΝΙΚΗ
ΤΟΥ
ldiwyhigh,ylow,$0005
MemFillWithRandLoop:
rcallPreudoRandon32kNumpers
st z+,ylow
cpiw zhigh,zlow,MemoryTop+1
brneMemFillWithRandLoop

ldiwzhigh,zlow,$0260
ldiwyhigh,ylow,$0005
```

```
MemtestLoopRand:
rcallPreudoRandon32kNumpers
ld  temp,z+
cp  temp,ylow
brnepc
cpiw zhigh,zlow,MemoryTop+1
brneMemtestLoopRand

MemTestEnd:

rcallledFlip

;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ PIN 5 ΤΗΣ ΠΟΡΤΑΣ
D ΣΕ ΕΞΟΔΟ
outi $11,$22
sbi  $12,1
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ ΚΑΤΑΧΟΡΙΤΗ Χ
ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΙΤΑΙ ΓΙΑ ΑΠΟΣΤΟΛΗ
DATA
ldiwxhigh,xlow,$0066
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ ΚΑΤΑΧΟΡΙΤΗ Υ
ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΙΤΑΙ ΣΑΝ ΔΕΙΚΤΗΣ
ΠΡΟΓΡΑΜΑΤΟΣ
ldiwyhigh,ylow,$0100
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ ΔΕΙΚΤΗ BUFFER
ΛΑΜΒΑΝΟΜΕΝΟΝ ΔΕΔΟΜΕΝΟΝ
stsiw resbuffpointh,resbuffpointl,startofresbuff
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ ΚΑΤΑΧΩΡΙΤΗ
ΓΩΝΙΑΣ
stsiw arccounterhigh,arccounterlow,$0000
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΝ ΔΕΙΚΤΟΝ BUFFER
ΕΝΤΟΛΟΝ
stsi bufferusage,0
stsi bufferpointer,0
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ ΚΑΤΑΧΟΡΙΤΗ
ΣΕΙΜΕΩΝ
ldi  flagsreg,$00
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ ΔΕΥΤΕΡΟΥ
ΚΑΤΑΧΟΡΙΤΗ ΣΕΙΜΕΩΝ
stsi flagsreg2,$00
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΝ ΚΑΤΑΧΟΡΙΤΩΝ
ΜΕΤΡΙΣΗΣ ΑΠΟΣΤΑΣΗΣ
bldi roadcounter1r,0
clr  roadcounter2r
clr  roadcounter3r
clr  roadcounter4r
bldi roadcounter1l,0
clr  roadcounter2l
clr  roadcounter3l
clr  roadcounter4l
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΝ ΚΑΤΑΧΟΡΙΤΩΝ
ΜΕΤΡΗΣΗΣ ΧΡΟΝΟΥ
ldi  timecounter,100
mov  speedcalcrate,timecounter
ldi  timecounter2,101
;ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ TIMER1
outi $2e,$01
;ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ PWM
outi $2f,$f1
;ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ INTERAPT ΤΟΥ
TIMER1
outi $39,$80
;ΑΡΧΟΙΚΟΠΟΙΗΣΗ ΤΟΥ BITRATE
outi $09,191
;ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ UART
outi $0a,$98
;ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΩΝ ΕΞΩΤΕΡΙΚΩΝ
INTEPAPTS
outi $3b,$c0
;ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΩΝ INTEPAPTS
sei



rcallLedFlip


BasicLoop:    ;ΒΑΣΙΚΟ LOOP
;rcall readfromarray2d
;rjmppc-1

ldiwzhigh,zlow,PithenesEpiloges
lds  temp,flagsreg2
andi temp,$03
add  zlow,temp
ldi  temp,0
adc  zhigh,temp
ijmp

PithenesEpiloges:
rjmp BasicLoop
rjmp buffermode
readprogramemorymode:

mov  zhigh,yhigh
mov  zlow,ylow
adiw zlow,1
rcallldcontiniousrameeprom
mov  temp2,temp
mov  zhigh,yhigh
mov  zlow,ylow
rcallldcontiniousrameeprom
ldi  zhigh,$07
mov  zlow,temp
mov  temp,temp2
adiw ylow,2
icall
rjmp BasicLoop

buffermode:
```

```
andioi$18,~$04
rcallreadbuffer
breqBasicLoop
ldi zhigh,$07
mov zlow,temp
push zlow
push zhigh
Wait4Operant:
rcallreadbuffer
breqWait4Operant
pop  zhigh
pop  zlow
icall
rjmp BasicLoop




;ΡΟΥΤΙΝΕΣ

;ΠΑΡΑΓΟΓΗ ΨΕΥΔΟΤΥΧΑΙΑΣ
ΑΚΟΛΟΥΘΙΑΣ 2 ΙΣΤΙΝ 15 ΑΡΙΘΜΟΝ
PreudoRandon32kNumpers:
mov temp,yhigh
swap temp
lsl  temp
lsl  temp
swap temp
eor  temp,ylow
ror  temp
rol  ylow
rol  yhigh
ret

;ΑΝΑΒΟΣΒΥΜΑ LED
LedFlip:
andioi$18,~$04
rcallBadDeley100ms
orioi $18,$04
rcallBadDeley100ms
ret

;ΡΟΥΤΙΝΑ ΠΟΥ ΓΕΜΙΖΕΙ ΤΗΝ ΕΩΤΕΡΙΚΗ
ΜΝΗΜΗ ΜΕ ΤΗΝ ΤΙΜΗ ΤΟΥ temp2
FillMemWith:
ldiwzhigh,zlow,$0260
MemFillLoop:
st  z+,temp2
cpiw zhigh,zlow,MemoryTop+1
brneMemFillLoop
ret

;ΡΟΥΤΙΝΑ ΠΟΥ ΕΛΕΝΧΕΙ ΑΝ ΜΙΑ ΠΟΡΤΑ
ΠΕΡΝΕΙ ΤΙΜΕΣ ΕΛΕΥΘΕΡΑ ΕΛΕΥΘΕΡΗ
porttest:
sti  z+,$ff
sti  z,$00
sbiw zlow,2
ld   temp,z
cpi  temp,0
brnebadport
adiw zlow,2
sti  z,$ff
subi zlow,2
ld   temp,z
cpi  temp,$ff
badport:
ret

ldcontiniousrameeprom:
cpiw zhigh,zlow,$0200
brlo ramcase
cli
subi zhigh,$02
out  $1f,zhigh
out  $1e,zlow
sbi  $1c,0
in   temp,$1d
reti
ramcase:
ld   temp,z
ret


incarccounter:
ld   temp,z+
ld   temp2,z
adiw temp,1
push temp2
cpi  temp2,byte2(maxofarcounter+1)
ldi  temp2,byte1(maxofarcounter+1)
cpc  temp,temp2
pop  temp2
brneinccomplete
ldiwtemp2,temp,0
inccomplete:
st  z,temp2
st  -z,temp
ret


decarccounter:
ld   temp,z+
ld   temp2,z
sbiw temp,1
brpl deccomplete
ldiwtemp2,temp,maxofarcounter
deccomplete:
st  z,temp2
st  -z,temp
```

```
ret


readbuffer:
ldi  zhigh,$01
lds  temp,bufferusage
lds  zlow,bufferpointer
cp  temp,zlow
breqbufferempty
ld  temp,z+
sts bufferpointer,zlow
bufferempty:
ret


write2buffer:
push temp
ldi  zhigh,$01
lds  zlow,bufferusage
mov  temp,zlow
inc  temp
lds  temp2,bufferpointer
cp  temp2,temp
breqbufferfull
sts bufferusage,temp
pop  temp
st  z,temp
bufferfull:
ret


fixpower:
brlo argo
cpi  temp,255-100
brshotiekana
inc  temp
rjmp otiekana
argo:
cpi  temp,0
breqotiekana
dec  temp
otiekana:
ret


;ΡΟΥΤΙΝΑ ΠΟΥ ΕΓΡΑΦΕΙ ΣΤΗΝ ΠΟΡΤΑ
ΜΟΝΟ ΤΑ ΒΙΤ ΠΟΥ ΣΧΕΤΙΖΟΝΤΑΙ
;ΜΕ ΤΗΝ ΦΟΡΑ ΠΕΡΙΣΤΡΟΦΗΣ (Η ΤΟ
ΦΡΕΝΑΡΙΣΜΑ) ΤΟΝ ΕΡΠΙΣΤΡΙΩΝ
setrotate:
andi temp,~$27
in  temp2,$18
andi temp2,$27
or  temp2,temp
out  $18,temp2
ret


;ΡΟΥΤΙΝΑ ΠΟΥ ΕΛΕΝΧΕΙ ΑΝ Ο ΑΡΙΣΤΕΡΟΣ
ΤΡΟΧΟΣ ΤΕΙΝΕΙ ΝΑ ΠΕΡΙΣΤΡΕΦΕΤΑΙ
ΚΑΘΟΛΟΥ-ΕΜΠΡΟΣ-ΠΙΣΩ
;ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΙΝΑΙ ΣΑΝ
ΕΓΙΝΕ ΣΙΓΡΙΣΗ ΜΕ ΜΙΚΡΟΤΕΡΟ ΓΙΑ ΤΟ
ΚΑΘΟΛΟΥ,ΙΣΟ
;ΓΙΑ ΤΟ ΜΠΡΟΣΤΑ Κ.Τ.Λ.
whatrotateleft:
in  temp,$18
com  temp
andi temp,$18
cpi  temp,~$18
brneoxi11
clr  temp
oxi11:
cpi  temp,$08
ret


;ΟΠΟΣ ΚΑΙ Η whatrotateleft ΑΛΛΑ ΓΙΑ ΤΟΝ
ΔΕΞΙΟ ΤΡΟΧΟ
whatrotateright:
in  temp,$18
andi temp,$c0
cpi  temp,$c0
brneoxi11r
clr  temp
oxi11r:
cpi  temp,$40
ret


BadDeley1s:
ldi  temp,20
rjmp BadDeleyYourChose
BadDeley100ms:
ldi  temp,2
BadDeleyYourChose:
push temp
ldi  temp2,240
l1:
djnztemp,l1
djnztemp2,l1
pop  temp
djnztemp,BadDeleyYourChose
ret


readfromarray2d:
;ΣΤΗΝ ΠΑΡΟΥΣΑ ΣΥΝΑΡΤΗΣΗ Η ΜΙΑ
ΔΙΑΣΤΑΣΗ ΜΠΕΝΕΙ ΣΤΟΝ temp ΚΑΙ Η
;ΑΛΛΗ ΣΤΟΝ temp2,ΤΟ ΑΠΟΤΕΛΕΣΜΑ
```

```
ΣΤΟΝ temp2
;Ο ΠΙΝΑΚΑΣ ΕΙΝΑΙ 152*152(0 ΕΩΣ 151)
push temp2
ldi  temp2,152
rcallmul8bit
pop  temp
add zlow,temp
ldi  temp2,0
adc  zhigh,temp2
push zlow
andizlow,$fc
clc
ror  zhigh
ror  zlow
ror  zhigh
ror  zlow
baddi zlow,byte1(arxipinaka)
adci zhigh,byte2(arxipinaka)
ld  temp2,z
pop  temp
rcallget2bitofreg
ret


get2bitofreg:
andi temp,$03
notyet2bit_2:
brnenotyet2bit
andi temp2,$03
ret
notyet2bit:
lsr  temp2
lsr  temp2
dec  temp
rjmp notyet2bit_2


gethenbitofreg:
andi temp,$07
notyet_2:
brnenotyet
andi temp2,$01
ret
notyet:
lsr  temp2
dec  temp
rjmp notyet_2


mul8bit:
ldiwzhigh,zlow,0
tst  temp
brnesinexise
ret
sinexise:
com  zlow
com  zhigh
sinexise2:
sub  zlow,temp2
sbci zhigh,0
djnztemp,sinexise2
com  zlow
com  zhigh
ret


;ΔΙΕΡΕΣΗ 16 BIT
;* "zhigh:zlow" (dividend) and "temp2:temp"
(divisor).

div16bit:
push zlow
push zhigh
push temp
push temp2
ldi  temp,$00
ldi  temp2,$00
push temp
push temp2
in   zlow,splow
in   zhigh,sphigh
dierese:
ldd  temp,z+6
ldd  temp2,z+4
cp  temp,temp2
ldd  temp,z+5
ldd  temp2,z+3
cpc  temp,temp2
brlo teliose
ldd  temp,z+2
ldd  temp2,z+1
adiw temp,1
std  z+2,temp
std  z+1,temp2
ldd  temp,z+6
ldd  temp2,z+4
sub  temp,temp2
std  z+6,temp
std  z+4,temp2
ldd  temp,z+5
ldd  temp2,z+3
sbc  temp,temp2
std  z+5,temp
std  z+3,temp2
rjmp dierese
teliose:
pop  temp2
pop  temp
pop  zlow
pop  zlow
```

```
pop  zhigh
pop  zlow
ret


getarc254counter:
ldiwzhigh,zlow,arccounterlow
ld  temp2,z+
ld  zhigh,z
mov zlow,temp2
ldiwtemp2,temp,47
rcalldiv16bit
ret


;ΡΟΥΤΙΝΕΣ icall ΕΣΩΤΕΡΙΚΗΣ ΧΡΗΣΗΣ
ifdxnotisthendonext:
mov temp2,roadcounter2r
sub temp2,dxregister
cp  temp2,temp
breqskipnextinstruction
ret


ifdxisthendonext:
mov temp2,roadcounter2r
sub temp2,dxregister
cpsetemp2,temp
ret
skipnextinstruction:
adiw ylow,2
ret


setdx:
mov dxregister,roadcounter2r
ret


ifempodiothennotgoto:
mov temp2,portbstate
andi temp2,$01
brnegoto0
ret


ifempodiothengoto:
mov temp2,portbstate
andi temp2,$01
breqgoto0
ret


goto3:
mov ylow,temp
ldi yhigh,3
ret

goto2:
mov ylow,temp
ldi yhigh,2
ret

goto1:
mov ylow,temp
ldi yhigh,1
ret

goto0:
mov ylow,temp
ldi yhigh,0
ret

gotoarc:
push temp
rcallgetarc254counter
mov temp2,temp
pop  temp
sub temp2,temp
cpi  temp2,128
brlo dexiasearc
rjmp aristerasearc

dexiasearc:
push temp
ldi temp,deksia
rcallsetrotate
pop  temp
elenxosdexia:
push temp
rcallgetarc254counter
mov temp2,temp
pop  temp
cp  temp,temp2
brneelenxosdexia
ret

aristerasearc:
push temp
ldi temp,aristera
rcallsetrotate
pop  temp
elenxosarist:
push temp
rcallgetarc254counter
mov temp2,temp
pop  temp
cp  temp,temp2
brneelenxosarist
ret
```

```
justsetspeed:
mov wandetspeedl,temp
mov wandetspeedr,temp
ret

stop:
ldi wandetspeedl,$00
ldi wandetspeedr,$00
ldi temp,$00
rcallsetrotate
ret

pisoeosedx:
push temp
ldi temp,piso
rcallsetrotate
pop  temp2
mov dxregister,roadcounter2r
com dxregister
mov temp,roadcounter2r
com temp
sub temp,dxregister
cp temp,temp2
brnepc-4
ret

dexiaeosdx:
push temp
ldi temp,deksia
rcallsetrotate
pop  temp2
mov dxregister,roadcounter2l
mov temp,roadcounter2l
sub temp,dxregister
cp temp,temp2
brnepc-3
ret

mprostaeosdx:
push temp
ldi temp,empros
rcallsetrotate
pop  temp2
mov dxregister,roadcounter2r
mov temp,roadcounter2r
sub temp,dxregister
cp temp,temp2
brnepc-3
ret

aristeraeosdx:
push temp
ldi temp,aristera
rcallsetrotate
pop  temp2
mov dxregister,roadcounter2r
mov temp,roadcounter2r
sub temp,dxregister
cp temp,temp2
brnepc-3
ret

mprostaeosempodio:
mov wandetspeedl,temp
mov wandetspeedr,temp
ldi temp,empros
rcallsetrotate
rcallBadDeley100ms
mov temp,portbstate
andi temp,$01
brnepc-2
ret

aristeraeosempodio:
mov wandetspeedl,temp
mov wandetspeedr,temp
ldi temp,aristera
rcallsetrotate
rcallBadDeley100ms
mov temp,portbstate
andi temp,$01
brnepc-2
ret


;ΠΕΡΙΧΗ ΚΛΗΣΗΣ ΡΟΥΤΙΝΩΝ icall
ΕΣΩΤΕΡΙΚΗΣ ΧΡΗΣΗΣ
.org $0700
rjmp aristeraeosempodio
rjmp mprostaeosempodio
rjmp aristeraeosdx
rjmp mprostaeosdx
rjmp dexiaeosdx     ;4
rjmp pisoeosedx
rjmp stop
rjmp justsetspeed
rjmp aristerasearc
rjmp dexiasearc     ;9
rjmp gotoarc
rjmp goto0
rjmp goto1      ;12
rjmp goto2
rjmp goto3
rjmp ifempodiothengoto
rjmp ifempodiothennotgoto ;16
rjmp setdx
rjmp ifdxisthendonext
rjmp ifdxnotisthendonext
```

```
.org $07ff
ret


;ΡΟΥΤΙΝΕΣ icall
.org $0800


ireset:
wdr
outi $21,$0d
ret


writeto:
lds  zhigh,$0062
lds  zlow,$0063
lds  temp,$0064
st z,temp
ret


writetosomebits:
ldi zhigh,$00
lds  zlow,$0062
;ΚΑΤΑΧΟΡΙΣΗ ΣΤΟΝ ΚΑΤΑΧΟΡΙΤΙ ΤΕΜΠ ΤΗΣ ΜΑΣΚΑΣ
lds  temp,$0063
;ΚΑΤΑΧΟΡΙΣΗ ΣΤΟΝ ΚΑΤΑΧΟΡΙΤΙ ΤΕΜΠ2 ΤΟΝ ΒΙΤ ΠΟΥ ΘΕΛΟΥΜΕ ΝΑ ΑΝΤΙΓΡΑΦΟΥΝ
lds  temp2,$0064
and temp2,temp
push temp2
;ΑΝΑΓΝΟΣΗ ΤΙΣ ΘΕΣΗΣ ΜΝΗΜΗΣ ΠΟΥ ΘΕΛΟΥΜΕ ΝΑ ΕΓΡΑΦΟΥΝ ΤΑ ΒΙΤ
ld  temp2,z
com  temp
and temp2,temp
pop  temp
or  temp2,temp
;ΕΓΡΑΦΗ ΣΤΙΝ ΘΕΣΗ ΜΝΗΜΗΣ ΠΟΥ ΘΕΛΟΥΜΕ ΝΑ ΕΓΡΑΦΟΥΝ ΤΑ ΒΙΤ
st  z,temp2
ret


setmotorspower:
lds  temp,$0062
out  $28,temp
lds  temp,$0063
out  $2a,temp
lds  temp,$0064
rcallsetrotate
ret


setspeed:
lds  wandetspeedl,$0062
lds  wandetspeedr,$0063
lds  temp,$0064
rcallsetrotate
ret


;ΡΟΥΤΙΝΑ ΠΟΥ ΠΡΟΚΑΛΕΙ ΤΗΝ ΕΚΙΝΗΣΗ ΑΠΟΣΤΟΛΗΣ ΕΝΟΣ ΠΑΚΕΤΟΥ ΣΤΕΛΝΟΝΤΑΣ
;ΤΟ ΠΡΟΤΟ ΒΥΤΕ.ΤΑ ΥΠΟΛΥΠΑ ΘΑ ΣΤΑΛΟΥΝ ΧΑΡΙΣ ΤΟ INTERAP ΟΛΟΚΛΗΡΟΣΗΣ
;ΑΠΟΣΤΟΛΗΣ ΕΝΟΣ ΒΥΤΕ
sendmesamthing:
lds  sendareastart,$0062
sendmesamthingforothers:
mov  xlow,sendareastart
resend:
stsi lastbit4qb,$08
bldi timeoutreg,30
ori  flagsreg,0b01000000
outi $0c,$00
sbi  $0a,5
ret


addtobuffer:
lds  temp,$0062
rcallwrite2buffer
ret


addtobuffer2byte:
lds  temp,$0062
rcallwrite2buffer
lds  temp,$0063
rcallwrite2buffer
ret


getfrombuffer:
eorioi $18,$04
rcallreadbuffer
sts  pcreadatfun,temp
in  temp,$3f
sts  isbufferemp4pc,temp
rjmp sendmesamthing
```

```
writetoeeprom:
lds   temp,$0062
andi temp,$01
out   $1f,temp
lds   temp,$0063
out   $1e,temp
lds   temp,$0064
out   $1d,temp
lds   temp2,$0062
andi temp2,$06
outi $1c,$04
out   $1c,temp2
stsi $0062,$00
ret


readeeprom:
lds   temp,$0062
out   $1f,temp
lds   temp,$0063
out   $1e,temp
sbi   $1c,0
in   temp,$1d
sts  pcreadatfun,temp
bldi sendareastart,pcreadatfun
rjmp sendmesamthingforothers


.org $0f00
rjmp ireset
rjmp writeto
rjmp writetosomebits
rjmp setmotorspower
rjmp setspeed
rjmp sendmesamthing
rjmp addtobuffer
rjmp addtobuffer2byte
rjmp getfrombuffer
rjmp writetoeeprom    ;9
rjmp readeeprom


.org $0fff
ihaverecive:
ret
```

Depth camera driven mobile robot for human localization and following

N. Skordilis, N. Vidakis, G. Papadourakis and G. Triantafyllidis
Applied Informatics and Multimedia
Technological Educational Institute of Crete, Heraklion, Greece
Email: gt@teicrete.gr

Abstract— **In this paper we describe the design and the development of a mobile robot able to locate and follow a human target using a depth camera, such as the ASUS Xtion Pro sensor. This sensor may use the PrimeSense's OpenNI library which can detect humans in a scene and also provide the position of the detected human in the 3D space. But this human detection and localization algorithm employs (among other techniques) the movement of the object in the scene, in order to decide if this object is a human or not. Thus, if an object seems to move (because of the mobile robot's movement), it is often falsely detected as a human. So a mechanism to discern the real humans from the objects is needed. In our case a special-tailored feed forward neural network is proposed and used. Experimental results show the efficient performance of the proposed design for a mobile robot with a depth camera for human localization and following.**

Keywords— *Depth cameras, mobile robot, human localization*

Introduction

During the last decade, robot technology has advanced significantly. However, conventional robots are only used for industrial use in some restricted places, including factories, and intelligent robots for our general daily use have yet to be achieved. The robots that will be needed in the near future are human-friendly robots that are able to coexist with humans and support humans effectively. One of the most important aspects in the development of human-friendly robots is to realize cooperation between humans and robots [1], [2].

In this context, there are many applications in robotics science that need the use of mobile robots that incorporate the capability of locating or/and following humans. For example, a robotic assistant [3] need to be close to and aware of the position of the person who commands it. In some other cases the ability to follow a human is enough, for example carrying objects. Also such robots can be used just as a game or a robotic pet [4].

Several research groups have presented work on robots designed to follow humans for over a decade. S.O. Lee et al [2] describe a mobile robot, which always faces humans and acts as an assistant robot. The mobile robot presented in [3] is a human-following robot for assisting humans. This robot is aimed at guiding a wheelchair in a hospital or a station and has the ability to estimate the position and velocity of humans and to avoid obstacles. In [4], four-legged mobile robots for following humans were considered. However, these studies only addressed the problem of how to follow humans, not how to detect the presence of humans. They developed their studies on the premise that human detection is possible.

In general, various approaches have been proposed to detect humans and the relative position between a mobile robot and a human, such as using ultrasonic sensors, voice recognition sensors, laser range sensors, infrared cameras, charge-coupled device (CCD) cameras and so on [5]. Gockley et al [6] discuss a laser-based person-tracking method and two different approaches to person-following: direction-following and path-following.

Taking into account the cost of such approaches, only data from an RGB image (or two images in the case of stereo vision) was the only choice for such a cheap computer vision system. However, the luminance of the area, the shadows, the uncertainty of the images (how a computer can separate a real face from a photograph?) produced extra trouble to the object and human detection and recognition. Also, the image-based extraction of the object distances was a difficult problem to be solved with the RGB image processing techniques - the object distance is vital for estimating scale, which is necessary for the object recognition.

All these problems can now be solved if we use a depth map. Laser scanners which can provide us with depth maps, are expensive, but now with the newly introduced depth cameras (such as the MS Kinect [7] or the Asus Xtion Pro [8]), there is a cheap and efficient way to get a depth map [9], [10]. Based on these devices, applications and middleware have been developed to provide us with high-level information like the position of a human. But it was designed with the constraint that only humans are moved in the scene.

In this context, this paper proposes a complete and efficient design of a mobile robot using a depth camera to locate and follow human by employing a neural network to separate the real from false human detections. The paper is organized as follows. In section 2 we describe the hardware and the technical characteristics of the mobile robot. Section 3 presents the data process, the algorithms and the architecture of the proposed approach. In Section 4 performance analysis is presented and finally Section 5 draws the conclusions and future extensions.

The mobile robot

The proposed design for the mobile robot consists of a 10" notebook, a vehicle on tracks, a circuit that is controlled from the notebook via serial port and drives the motors of the vehicle. Also, a USB to serial converter is part of this robot and an ASUS Xtion pro sensor. The core of the control circuit is an at90s8515 avr microcontroller. The notebook has 2GB ram and an Intel atom N570 processor at 1.66 Ghz.

The total weight of the mobile robot is 4 kgr. The track length is 32 cm, the total length of the mobile robot is 41 cm, the height until the notebook is 20 cm and with the Xtion sensor included the height is 28 cm. The width is 22 cm.

Figure 1 shows two photos of the proposed mobile robot.

Figure 1. The mobile robot

Proposed architecture

To detect humans using a depth camera we employ the standard OpenNI human detect functions [11]. These functions provide us with the coordinates of the human, a depth map of the scene and a map illustrating the pixels of the depth map belonging to the human.

It is not efficient to use all these forms of the data to our neural network, which will be built to discern the real humans from the object falsely considered as humans (due to the depth camera movement), since this provided data contains a lot of redundant information.

In this context, we propose a process which decreases the amount of the data, keeping it to the necessary for the required human detection. Also, it would be more effective if this process provides us with data that doesn't change significantly when the human is in a different position of the scene.

Towards this goal, we count the number of pixels per line belonging to the object detected as human from the OpenNI function. This number (which is actually the object's width) is not significantly changed with object moving in x axis. Then, to decrease the amount of data, sub-sampling is used to gather this information. Only 16 lines are examined and only 16 pixels per line.

But experiments show that only this is not enough to recognize humans effectively and one more feature is added as input in the neural network: Each of the 16 lines is scanned and the number of changes from object pixels to non object pixel or the opposite is counted. These values didn't change for small change of the position of object in x and z axis.

So, the neural network is provided with the input of the central position (xyz) of the object, the pixels per line of object (width) and the number of changes. The suggested neural network has 35 neurons at input layer, 85 neurons at hidden layer and 2 at output. The neurons transfer function is tansig. The one output means human and the other output not.

Figures 2-4 show screen shots of the proposed software.

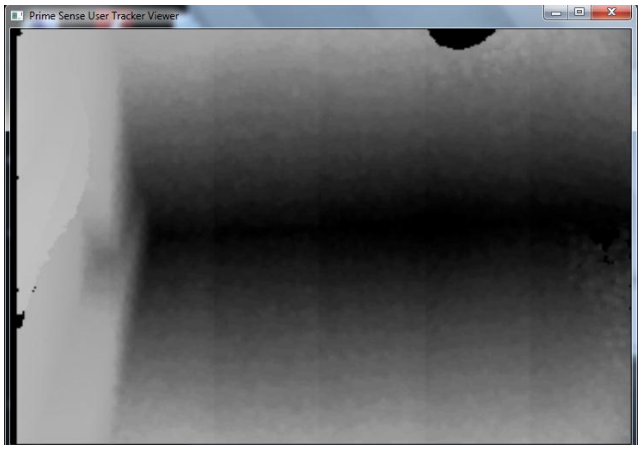


Figure 2. Depth map without any human detection

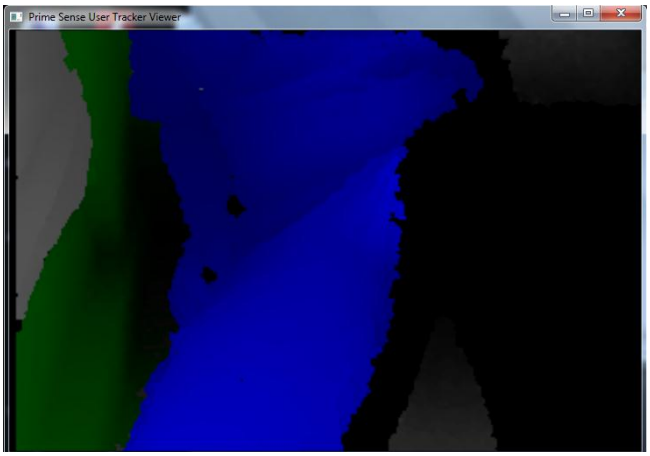


Figure 3. Depth map with a false human detection

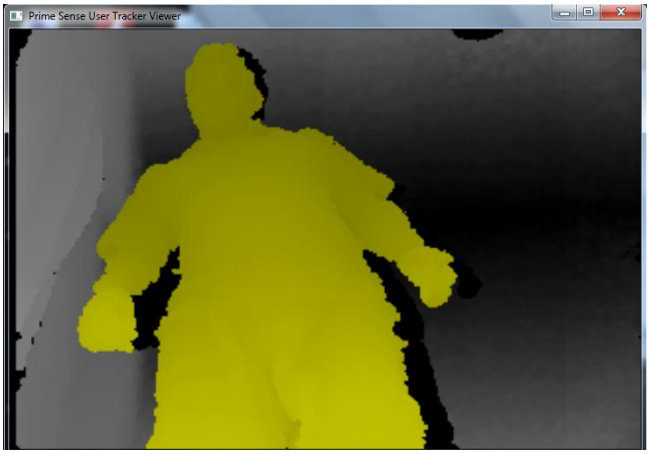


Figure 4. Depth map with a true human detection

If the robot can't detect a human for a significant amount of time, it starts to rotate slowly with the prospect that a human is somewhere out of the field of view.

After the detection of a human we use the coordinates of the detected human to move the mobile robot accordingly. Please notice that the image used in our experiments is of QVGA resolution 320x240. So, if the x coordinate is greater than 200 the mobile robot rotates right in order to centralize the human in its frame of 320 pixels width, else if x is less than 120 rotates left. If x is in the area 120-200, then if z<1000 the mobile robot stops. Else if z<1400 the power of the motors is lowered proportional to the difference of z from 1000. Closer to 1400 the speed decreases. If z is greater than 1400 maximum power is chosen. If x is in the range 120 to 200 the balance of the power between left and right motor is determined as follows: If x=160 the power is equal, if x is greater than 160 left motor takes more power as much is the difference. If x is less than 160 the right motor takes more power according to the same rule. The proposed scheme is illustrated in Figure 5.

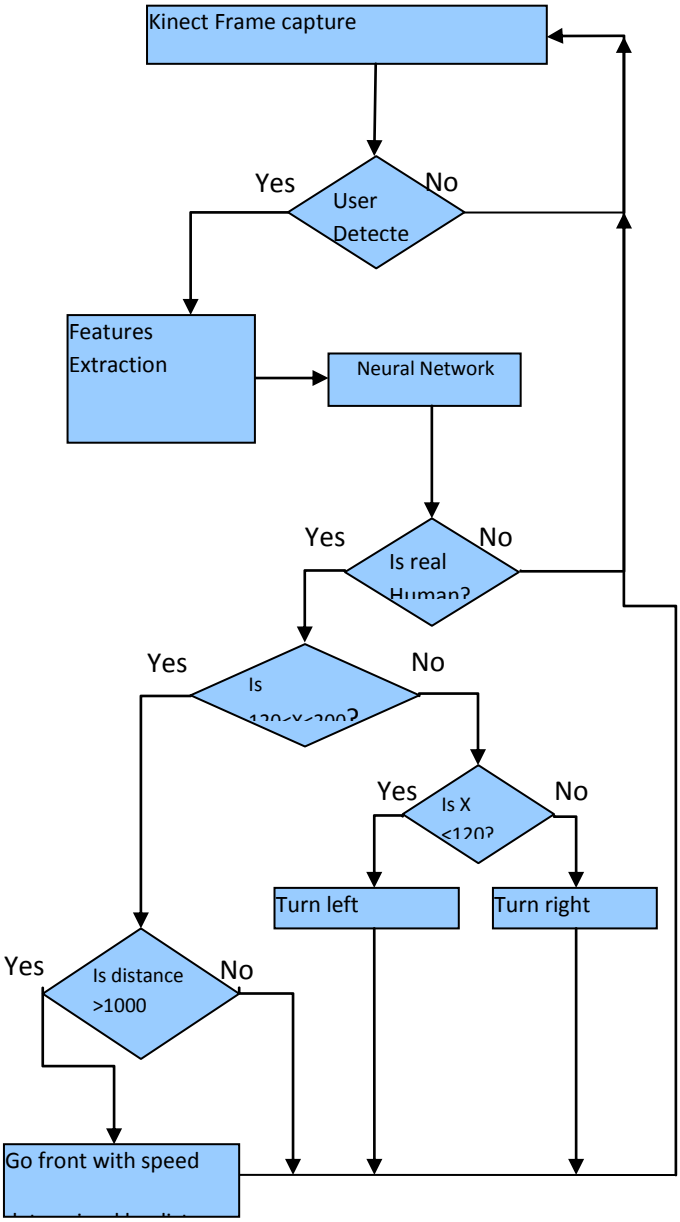


Figure 5. Proposed scheme

Performance

In our experimental results we used two sets of samples for training and testing the neural network. At first 775 samples (depth maps) were used with a detected human as provided from OpenNI function (true or false human detection). Each sample contains the central position of the detected object (human), its width and the number of changes from object pixels to non object pixel or the opposite. In 259 of these samples there is a human in the scene, while in the rest 516 samples there is a false detection of a human.

The second set of samples used for training and testing contains 4196 samples, from which 2909 samples present a true human detection and 3425 samples present a false human detection.

Table I illustrates the false human acceptance and rejection. 70% of the samples used for training, 15% for validation and 15% for testing. When using the set B, the performance of the proposed neural network is very satisfactory.

TABLE I. FALSE HUMAN ACCEPTANCE AND REJECTION

	False human acceptance	***false human rejection***
Set A 775 samples	4.7%	4 %
Set B 4196 samples	2.6 %	1.4 %

Overall, the proposed mobile robot design provides efficient detection and following of a human at maximum distance of about 4 meters. A video showing the mobile robot following a human can be found at http://www.youtube.com/watch?v=-lTHV6kswWA

Conclusions and Future Work

In this paper we presented a human following mobile robot. A non-expensive depth sensor (Asus Xtion Pro) makes the most part of the job for detecting humans in the scene, by using the OpenNI functions. But several false detections occur due to the movement of the scene, caused by the mobile robot's movement. So, we propose a neural network to discard these false detections.

Future work includes the use of RGB information for human recognition using SIFT or SURF. Then our mobile robot will be able to follow a specified person.

References

[18] H. Sidenbladh, D. Kragic, and H. I. Christensen, "A person following behavior for a mobile robot," in Proc. 1999 IEEE Int. Conf. Robotics and Automation, 1999, pp. 670–675.

[19] S.-O. Lee, M. Hwang-Bo, B.-J. You, S.-R. Oh, and Y.-J. Cho, "Vision based mobile robot control for target tracking," in Proc. IFAC Workshop Mobile Robot Technology, 2001, pp. 73–78.

[20] E. Prassler, D. Bank, B. Kluge, and M. Hagele, "Key technologies in robot assistants: Motion coordination between a human and a mobile robot," in Proc. 32nd Int. Symp. Robotics, 2001, pp. 410–415.

[21] Masahiro Fujita, "AIBO: Toward the Era of Digital Creatures," in the International Journal of Robotics Research, October 2001 vol. 20 no. 10781-794.

[22] Quoc Khanh Dang; Young Soo Suh; , "Human-following robot using infrared camera," Control, Automation and Systems (ICCAS), 2011 11th International Conference on , vol., no., pp.1054-1058, 26-29 Oct. 2011

[23] R. Gockley, J. Forlizzi, and R. Simmons, "Natural person-following behavior for social robots," in Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, 2007

[24] MS Kinect, http://www.microsoft.com/en-us/kinectforwindows/

[25] ASUS Xtion Pro, http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO/

[26] Albrektsen, Sigurd Mørkved, "Using the Kinect Sensor for Social Robotics," Student thesis, Institutt for teknisk kybernetikk, Norwegian University of Science and Technology, 2011

[27] Hoshino, F.; Morioka, K.; , "Human following robot based on control of particle distribution with integrated range sensors," System Integration (SII), 2011 IEEE/SICE International Symposium on , vol., no., pp.212-217, 20-22 Dec. 2011

[28] OpenNI framework, http://www.openni.org/ and http://openni.org/Documentation/

[29] Robert Bodor and Bennett Jackson and Nikolaos Papanikolopoulos, "Vision-based human tracking and activity recognition," in Proc. of the 11th Mediterranean Conf. on Control and Automation, 2003, pp. 18-20.