



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

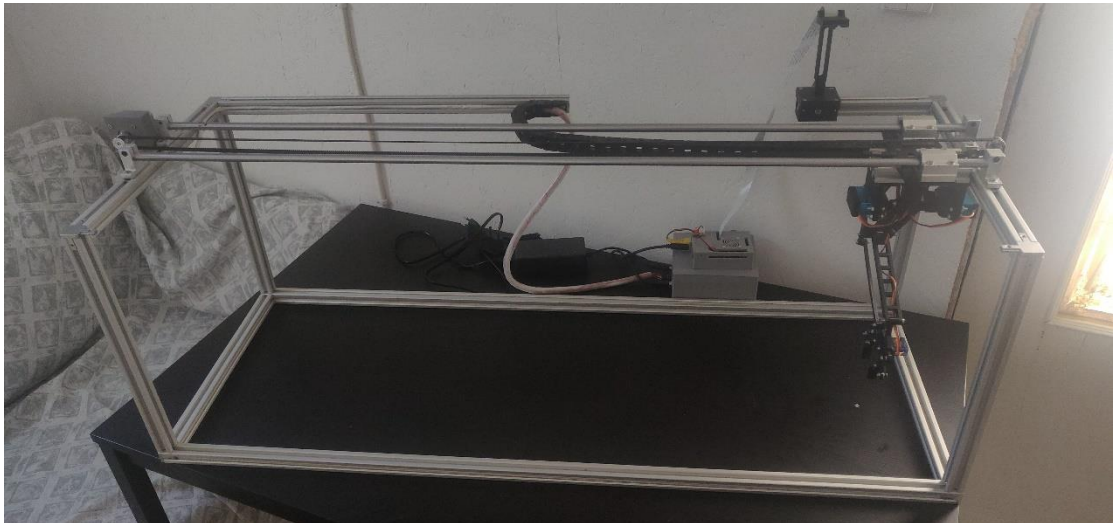
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΠΜΣ “ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ ΚΑΙ ΕΥΦΥΗ ΠΕΡΙΒΑΛΛΟΝΤΑ”

Διπλωματική Εργασία

**Σχεδίαση, Ανάπτυξη και Προγραμματιζόμενος Έλεγχος Ρομποτικού
Συστήματος με Δυνατότητα Οπτικής Αναγνώρισης Αντικειμένων**



Φοιτητής Πολυχρόνης Κουτσίδης

ΑΜ: msciot19004

Επιβλέπων Καθηγητής

Ξενοφών Διονύσιος Κανδρής

Καθηγητής

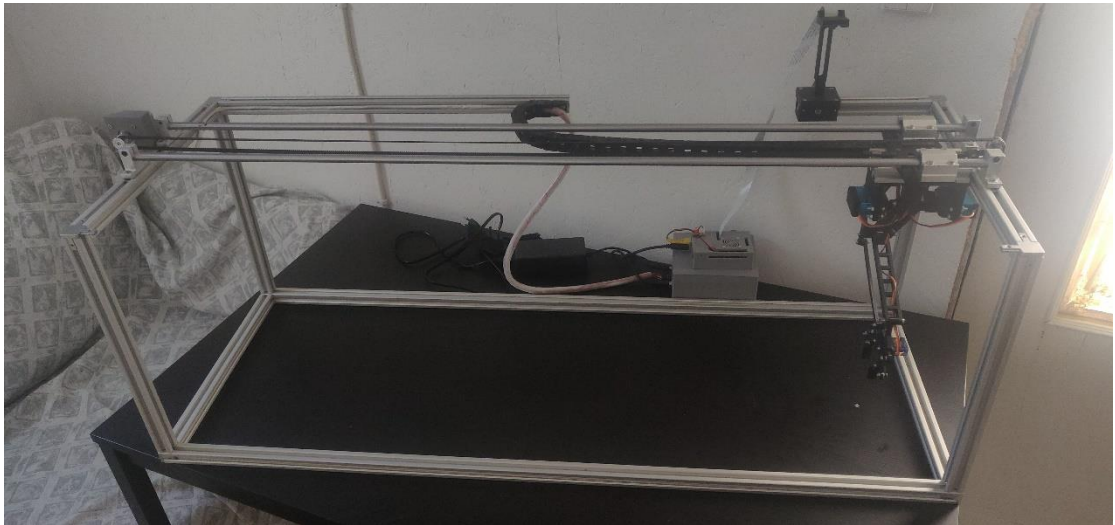
ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Οκτώβριος 2021



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
M.Sc. in INTERNET OF THINGS AND INTELLIGENT ENVIRONMENTS

Diploma Thesis

Design, Development and Programmable Control of a Robotic System with Visual Object Recognition




Student: Polychronis Koutsidis
Registration Number: msciot19004

Supervisor

Xenofon Dionisis Kandris
Professor

ATHENS-EGALEO, October 2021

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Ξενοφών Διονύσιος Κανδρής Καθηγητής	Γρηγόριος Καλτσάς Καθηγητής	Παναγιώτης Παπαγέωργας Καθηγητής
		

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Πολυχρόνης Κουτσίδης,
Ιούνιος, 2021**

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Πολυχρόνης Κουτσίδης του Γεωργίου, με αριθμό μητρώου msciot19004 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Ο Δηλών

Πολυχρόνης Κουτσίδης



Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον εισηγητή και επιβλέποντα της διπλωματική εργασίας μου, κύριο Διονύσιο Κανδρή για την πολύτιμη βοήθειά του, την άψογη συνεργασία μας καθώς και την καθοδήγησή του, καθ' όλη την διάρκεια της εκπόνησης της εργασίας μου.

Θα ήθελα ακόμη, να εκφράσω την ευγνωμοσύνη μου προς την οικογένεια μου για την ηθική υποστήριξή και για την υπομονή που μου έδειξαν σε όλη τη διάρκεια της φοίτησής μου.

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας είναι η σχεδίαση και ανάπτυξη ενός ρομποτικού συστήματος σταθερής βάσης, το οποίο αναγνωρίζει οπτικά τα αντικείμενα που βρίσκονται στο περιβάλλον του, τα ξεχωρίζει ανάλογα το χρώμα τους και στη συνέχεια τα τοποθετεί σε συγκεκριμένες επιθυμητές θέσεις.

Για να επιτευχθεί ο παραπάνω στόχος, είναι σημαντικό να κατανοηθεί ο τρόπος λειτουργίας των ρομποτικών συστημάτων σταθερής βάσης καθώς και τους τρόπους αναγνώρισης ενός αντικειμένου. Έτσι αρχικά γίνεται μια αναφορά στα ρομποτικά συστήματα, στα ρομποτικά συστήματα σταθερής βάσης και ο τρόπος λειτουργίας της κάθε διαμόρφωσης.

Έπειτα, περιγράφονται οι τεχνικές της αναγνώρισης των αντικειμένων, καθώς και τα είδη παραμορφώσεων της κάμερας.

Στη συνέχεια, περιγράφεται συνοπτικά ο τρόπος της κατασκευής του ρομποτικού συστήματος, ο προγραμματισμός του και οι διαδικασίες που χρειάζονται για την αντιμετώπιση των παραμορφώσεων την κάμερας.

Τέλος αξιολογείται η λειτουργία του συστήματος, εξάγονται συμπεράσματα και καταγράφονται προτάσεις για μελλοντικές βελτιώσεις που μπορούν να πραγματοποιηθούν.

Λέξεις – κλειδιά

Ρομποτικός βραχίονας σταθερής βάσης, Οπτική αναγνώριση αντικειμένων, Κάμερα.

Abstract

The aim of this thesis is to design and develop a fixed base robotic system, which visually recognizes ambient objects, separates them according to their color and then places them in certain desired places. To achieve the above goal, it is important to understand how a fixed base robotic system work as well as how to identify an object. Thus, a reference is made first to the fixed base robotic systems and the working principle of each type. Object detection techniques are described, as well as types of camera distortion.

Subsequently, there is a brief description of how the robotic system is built, the programming and the procedures needed to deal with the camera distortion.

Finally, the performance of the system developed is evaluated, concluding remarks are drawn and suggestions for future enhancements of the systems are made.

Keywords

Fixed base robotic arm, Visual object detection, Camera

Περιεχόμενα

Κατάλογος Εικόνων	7
ΕΙΣΑΓΩΓΗ.....	9
1. Ρομποτικά συστήματα	10
1.1 Τύποι ρομποτικών συστημάτων	11
1.1.1 Προ-προγραμματισμένα ρομπότ (Pre-Programmed robots)	11
1.1.2 Ανθρωποειδή ρομπότ (Humanoid robots)	12
1.1.3 Αυτόνομα ρομπότ (Autonomous Robots)	12
1.1.4 Τηλεκατευθυνόμενα ρομπότ (Teleoperated Robots)	13
1.1.5 Φορετά ρομπότ (Wearable Robots)	14
1.2 Ρομποτικά συστήματα σταθερής βάσης	14
1.2.1 Καρτεσιανό ρομποτικό σύστημα (Cartesian)	14
1.2.2 Ρομποτικό σύστημα SCARA	15
1.2.3 Αρθρωτό ή 6 αξόνων ρομποτικό σύστημα (Articulated or 6 Axis)	16
1.2.4 Κυλινδρικό ρομποτικό σύστημα (Cylindrical).....	17
1.2.5 Ρομποτικό σύστημα Δέλτα (Delta)	18
1.2.6 Σφαιρικό ρομποτικό σύστημα (Spherical).....	19
2. Οπτική ανίχνευση αντικειμένων και προβλήματα	20
2.1 Τεχνική Background Subtraction.....	20
2.2 Οπτική αναγνώριση με Deep Learning.....	21
2.2.1 Region-based Convolutional Neural Networks (R-CNN).....	21
2.2.2 Fast R-CNN	22
2.2.3 Faster R-CNN.....	23
2.2.4 Region-based Fully Convolutional Network (R-FCN)	23
2.2.5 Spatial Pyramid Pooling Net (SPPNet)	24
2.2.6 Single Shot Detector (SSD).....	24
2.2.7 You Only Look Once (YOLO).....	25
2.3 Οπτική ανίχνευση με Machine Learning.....	26
2.3.1 Viola Jones	26
2.3.2 Ιστόγραμμα Προσανατολισμένων Κλίσεων (HOG)	29
2.3.3 Μοντέλο με δυνατότητα παραμόρφωσης (DPM)	29
2.4 Παραμόρφωση εικόνας της κάμερας.....	30
3. Ανάπτυξη ρομποτικού συστήματος	32
3.1 Μηχανολογικό μέρος.....	32
3.2 Μετατροπή συντεταγμένων Y, Z σε γωνίες	39
3.3 Ηλεκτρονικό μέρος	44
3.4 Προγραμματισμός	48
3.5 Διόρθωση κάμερας (Camera Calibration & Perspective Calibration)	53
3.6 Τρόπος λειτουργίας	59
4. Επίλογος	61
4.1 Σύνοψη εργασίας.....	61
4.2 Προβλήματα και αντιμετώπιση	61
4.3 Παρατηρήσεις / Συμπεράσματα	62
4.4 Προτάσεις μελλοντικής εξέλιξης.....	62
Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές	64

Κατάλογος Εικόνων

Εικόνα 1.1.1 Προ-προγραμματισμένα ρομπότ. [35]

Εικόνα 1.1.2 Ανθρωποειδή ρομπότ. [36]

Εικόνα 1.1.3 Αυτόνομο ρομπότ. [37]

Εικόνα 1.1.4 Τηλεκατευθυνόμενο ρομπότ. [38]

Εικόνα 1.1.5 Φορετά ρομπότ. [39]

Εικόνα 1.2.1 Καρτεσιανό ρομποτικό σύστημα, όπου (α) απλό καρτεσιανό και (β) τύπου Gantry [10, 11]

Εικόνα 1.2.2 Χώρος εργασίας Καρτεσιανού Συστήματος..[40]

Εικόνα 1.2.3 Ρομποτικό σύστημα SCARA. [41]

Εικόνα 1.2.3 Χώρος εργασίας ρομποτικού συστήματος SCARA. [42]

Εικόνα 1.2.5 Αρθρωτό ρομποτικό σύστημα. [43]

Εικόνα 1.2.6 Χώρος εργασίας αρθρωτού ρομποτικού συστήματος. [44]

Εικόνα 1.2.7 Κυλινδρικό ρομποτικό σύστημα. [45]

Εικόνα 1.2.8 Χώρος εργασίας κυλινδρικού ρομποτικού συστήματος. [46]

Εικόνα 1.2.9 Ρομποτικό σύστημα Δέλτα. [47]

Εικόνα 1.2.10 Χώρος εργασίας ρομποτικού συστήματος Δέλτα. [48]

Εικόνα 1.2.11 Χώρος εργασίας ενός σφαιρικού ρομποτικού συστήματος. [49]

Εικόνα 2.1.1 Παράδειγμα μεθόδου Background Subtraction. [50]

Εικόνα 2.2.1 R-CNN. [23]

Εικόνα 2.2.2: Fast R-CNN. [23]

Εικόνα 2.2.3 Faster R-CNN. [23]

Εικόνα 2.2.4 SPPNet [23]

Εικόνα 2.2.5 SSD. [51]

Εικόνα 2.2.6 YOLO. [23]

Εικόνα 2.3.1 Παράθυρα ανίχνευσης [26]

Εικόνα 2.3.2 Haar-like χαρακτηριστικό. [26]

Εικόνα 2.3.3 Δημιουργία ακέραιης εικόνας. [27]

Εικόνα 2.3.4 Αλγόριθμος AdaBoost. [27]

Εικόνα 2.3.5 ταξινομητής Cascade [27]

Εικόνα 2.4.1 Χωρίς παραμόρφωση (α), παραμόρφωση βαρελιού (β), παραμόρφωση μαξιλαριού (γ), παραμόρφωση μουστάκι (δ). [52]

Εικόνα 3.1.1 Μέρη του ρομποτικού βραχίονα.

Εικόνα 3.1.2 Ενισχυμένες βέργες.

Εικόνα 3.1.3 Τρίγωνο σύνδεσης (α), απλές βέργες (β).

Εικόνα 3.1.4 Εύρος λειτουργίας. [32]

Εικόνα 3.1.5 Άμεση σύνδεση του σέρβο με την ενισχυμένη βέργα B (α) και αντίστοιχα η έμμεση σύνδεση του δεύτερου σέρβο (β).

Εικόνα 3.1.6 Τρίγωνο σύνδεσης.

Εικόνα 3.1.7 Άνω όψη σταθερής βάσης (α) και κάτω όψη σταθερής βάσης.

Εικόνα 3.1.8 Γενικό πλάνο ρομποτικού βραχίονα.

Εικόνα 3.1.9 Γενική όψη του άξονα X (α), σύνδεση του βηματικού, του οδοντωτού τροχαλία και των γραμμικών οδηγών (β) και σύνδεση του απλού τροχαλία και των άκρων των οδηγών (γ).

Εικόνα 3.1.10 Εγκατάσταση κάμερας.

Εικόνα 3.1.11 Γενικό πλάνο κατασκευής στο μηχανολογικό μέρος.

Εικόνα 3.2.1 Περιγραφή ρομποτικού βραχίονα. [32]

Εικόνα 3.2.2 Ενδιαφερόμενα στοιχεία ρομποτικού βραχίονα. [32]

Εικόνα 3.2.3 Υπολογισμός H. [32]

Εικόνα 3.2.4 Υπολογισμός s_1 , s_2 , α και β . [32]

Εικόνα 3.2.5 Υπολογισμός δεύτερου σέρβο κινητήρα. [32]

Εικόνα 3.3.1 Ηλεκτρονικό διάγραμμα.

Εικόνα 3.3.2 Κουτί με τα εξαρτήματα.

Εικόνα 3.3.3 Σύνδεση κάμερας και τελική σύνδεση όλων των εξαρτημάτων.

Εικόνα 3.3.4 Τελικό αποτέλεσμα κατασκευής.

Εικόνα 3.4.1 Αρχεία προγράμματος.

Εικόνα 3.4.2 Έναρξη σειριακής επικοινωνίας για το Arduino (αριστερά) και το Raspberry Pi (δεξιά).

Εικόνα 3.4.3 Κώδικας ανίχνευσης αντικειμένων.

Εικόνα 3.4.4 Οριοθέτηση του περιγράμματος.

Εικόνα 3.4.5 Τελική διαδικασία αναγνώρισης αντικειμένων.

Εικόνα 3.4.6 Καθορισμός ορίων των χρωμάτων σε BGR.

Εικόνα 3.4.7 Καθορισμός ύψους.

Εικόνα 3.4.8 Μεταβλητές διόρθωσης.

Εικόνα 3.4.9 Συντεταγμένες τοποθέτησης αντικειμένων.

Εικόνα 3.5.1 Πίνακας κάμερας. [53]

Εικόνα 3.5.2 Χαρτί με τις αριθμημένες κουκκίδες από το οπτικό πεδίο της κάμερας.

Εικόνα 3.5.3 Πίνακας του `newcam_mtx` της κατασκευής, με $c_x=633$ και $c_y=357$.

Εικόνα 3.5.4 Εντοπισμός του σημείου των pixel c_x, c_y .

Εικόνα 3.5.5 Υπολογισμός του ύψους.

Εικόνα 3.5.6 Pixel κάθε κουκκίδας.

Εικόνα 3.5.7 Τοποθέτηση των δεδομένων.

Εικόνα 3.5.8 S factor.

Εικόνα 3.6.1 Εμφάνιση μηνύματος Background captured.

Εικόνα 3.6.2 Αναγνώριση αντικειμένων.

Εικόνα 3.6.3 Δοκιμές συστήματος.

ΕΙΣΑΓΩΓΗ

Υπάρχουν πολλά είδη ρομποτικών συστημάτων, ανάλογα με τα μηχανικά και ηλεκτρονικά χαρακτηριστικά τους μέρη, κάθε ρομποτικό σύστημα έχει διαφορετικές ιδιότητες. Στην παρούσα διπλωματική εργασία, ο σκοπός είναι η κατασκευή ενός ρομποτικού συστήματος σταθερής βάσης, ο

οποίος αναγνωρίζει και ξεχωρίζει οπτικά τα αντικείμενα με βάση το χρώμα τους και στη συνέχεια τα περισυλλέγει και τα τοποθετεί σε ορισμένα επιθυμητά σημεία. Γι' αυτό το λόγο γίνεται η αναφορά των χαρακτηριστικών που μπορούν να αποτελούνται από ένα τέτοιο σύστημα.

Συγκεκριμένα στο πρώτο κεφάλαιο αναφέρονται οι τύποι των ρομποτικών συστημάτων καθώς και τα ρομποτικά συστήματα σταθερής βάσης.

Στο δεύτερο κεφάλαιο γίνεται μια αναφορά στις μεθόδους της οπτικής αναγνώρισης και στις παραμορφώσεις που παρουσιάζονται στην εικόνα μιας κάμερας.

Στο τρίτο κεφάλαιο αναλύεται η κατασκευή του συστήματος και επεξηγείται η διαδικασία που χρησιμοποιήθηκε για την ελαχιστοποίηση της παραμόρφωσης της εικόνας από την κάμερα.

Τέλος στο τέταρτο κεφάλαιο υπάρχουν ορισμένα συμπεράσματα που παρουσιάστηκαν μετά το πέρας της υλοποίησης και χρήσης της κατασκευής και παραπέμπονται ορισμένες βελτιώσεις που μπορούν να πραγματοποιηθούν.

1. Ρομποτικά συστήματα

Ένα ρομπότ είναι μια μηχανή αυτόματης λειτουργίας όπου μπορεί να βοηθήσει ή να αντικαταστήσει μια εργασία που πραγματοποιεί ο άνθρωπος. Συνήθως τα χαρακτηριστικά του δεν μοιάζουν με τον άνθρωπο καθώς και ο τρόπος εκτέλεσης μιας εργασίας δεν είναι ίδιος.^[1]

Ένα γενικός ορισμός για το τι είναι ένα ρομπότ έχει ως ακολούθως.

Ένα ρομπότ είναι μια αυτόνομη μηχανή ικανή να ανιχνεύει το περιβάλλον του, να εκτελεί υπολογισμούς για τη λήψη αποφάσεων και να εκτελεί ενέργειες στον πραγματικό κόσμο.^[2]

1.1 Τύποι ρομποτικών συστημάτων

Υπάρχουν έξι είδη ρομποτικών συστημάτων. Χρησιμοποιούνται σε διαφορετικά περιβάλλοντα και για διαφορετικές χρήσεις, αν και είναι πολύ διαφορετικά στην εφαρμογή και τη μορφή τους, όλοι μοιράζονται τρεις βασικές ομοιότητες όσον αφορά την κατασκευή τους.

- 1. Μηχανικά μέρη:** Όλα τα ρομποτικά συστήματα αποτελούνται από ορισμένα μηχανικά μέρη όπου είναι σχεδιασμένα για να πετύχουν μια συγκεκριμένη εργασία.
- 2. Ηλεκτρονικά μέρη:** Όλα τα ρομποτικά συστήματα έχουν ορισμένα ηλεκτρονικά εξαρτήματα για την τροφοδοσία και τον έλεγχο τους. Επίσης χρησιμοποιούνται για την κίνηση (μέσω κινητήρων), την ανίχνευση (όπου τα ηλεκτρικά σήματα χρησιμοποιούνται για τη μέτρηση πραγμάτων όπως θερμότητα, ήχος, θέση και ενεργειακή κατάσταση) και την λειτουργία (τα ρομποτικά συστήματα χρειάζονται κάποιο επίπεδο ηλεκτρικής ενέργειας που παρέχεται στους κινητήρες τους και αισθητήρες για ενεργοποίηση και εκτέλεση βασικών λειτουργιών).
- 3. Προγραμματισμός:** Με τον προγραμματισμό, το ρομποτικό σύστημα μπορεί να λαμβάνει αποφάσεις για το πότε ή πως να κάνει κάτι.^[3]

1.1.1 Προ-προγραμματισμένα ρομπότ (Pre-Programmed robots)

Τα προ-προγραμματισμένα ρομπότ λειτουργούν σε ελεγχόμενο περιβάλλον όπου εκτελούν απλές και επαναλαμβανόμενες εργασίες. Ένα παράδειγμα τέτοιου ρομποτικού συστήματος είναι οι ρομποτικοί βραχίονες. Ο ρομποτικός βραχίονας εξυπηρετεί μια λειτουργία (συγκόλληση, μεταφορά αντικειμένων κλπ.) και η εκτέλεση της λειτουργίας του είναι πιο αποτελεσματική και γρήγορη σε σύγκριση με τον άνθρωπο.^[4]



Εικόνα 1.1.1: Προ-προγραμματισμένα ρομπότ.

1.1.2 Ανθρωποειδή ρομπότ (Humanoid robots)

Τα ανθρωποειδή ρομποτικά συστήματα μοιάζουν με τον άνθρωπο και είναι σχεδιασμένα να μιμούνται την ανθρώπινη κίνηση και αλληλεπίδραση. Τα ανθρωποειδή ρομπότ χρησιμοποιούνται στην επιθεώρηση και τη συντήρηση.

Επίσης είναι διατεθειμένοι να αναλάβουν εργασίες με μεγάλη επικινδυνότητα για τον άνθρωπο (πχ αντιμετώπιση καταστροφών σε σταθμούς ηλεκτροπαραγωγής) καθώς και εργασίες ρουτίνας των αστροναυτών στο διάστημα.

Άλλες διαφορετικές εφαρμογές περιλαμβάνουν την παροχή συντροφιάς για ηλικιωμένους και άρρωστους και ενδεχομένως ακόμη και ως ξενιστής για την ανάπτυξη ανθρώπινων οργάνων μεταμόσχευσης.^[5]



Εικόνα 1.1.2: Ανθρωποειδή ρομπότ.

1.1.3 Αυτόνομα ρομπότ (Autonomous Robots)

Το αυτόνομο ρομπότ είναι σχεδιασμένο να λειτουργεί αυτόνομα, δηλαδή χωρίς την παρέμβαση του ανθρώπου. Τα αυτόνομα ρομπότ έχουν συχνά εξελιγμένα χαρακτηριστικά που μπορούν να τους βοηθήσουν να κατανοήσουν το φυσικό τους περιβάλλον.

Συνήθως κάνουν τη δουλειά τους χωρίς καμία ανθρώπινη αλληλεπίδραση, εκτός εάν είναι απαραίτητη ως μέρος της εργασίας. Πολλά από αυτά τα ρομποτικά συστήματα έχουν αισθητήρες και άλλα λειτουργικά εργαλεία που τους βοηθούν να βλέπουν τυχόν εμπόδια στο δρόμο τους, ή να περιηγούνται σε δωμάτια, διαδρόμους ή κάποιο άλλο είδος περιβάλλοντος.^[6]



Εικόνα 1.1.3: Αυτόνομο ρομπότ.

1.1.4 Τηλεκατευθυνόμενα ρομπότ (Teleoperated Robots)

Τα τηλεκατευθυνόμενα ρομποτικά συστήματα είναι ημιαυτόνομα ρομπότ που χρησιμοποιούν ενσύρματο ή ασύρματο δίκτυο ώστε να υπάρχει ανθρώπινος έλεγχος, από μια απόσταση ασφαλείας.

Τα τηλεκατευθυνόμενα ρομπότ χρησιμοποιούνται κυρίως σε ιατρικές χειρουργικές επεμβάσεις και στρατιωτικές επιχειρήσεις. Οι κρίσιμες χειρουργικές επεμβάσεις γίνονται πιο εύκολα με τους τηλεκατευθυνόμενους ρομποτικούς βραχίονες ή εργαλεία, λόγω της ικανότητάς τους να φτάσουν στα πιο στενά μέρη όπου τα ανθρώπινα χέρια δεν μπορούν να λειτουργήσουν. Σε στρατιωτικές επιχειρήσεις, τα τηλεκατευθυνόμενα ρομπότ βοηθούν στη συγκέντρωση πληροφοριών και στην εκτέλεση επικίνδυνων εργασιών, όπως εξουδετέρωση ή μετακίνηση ενός εκρηκτικού.^[7]

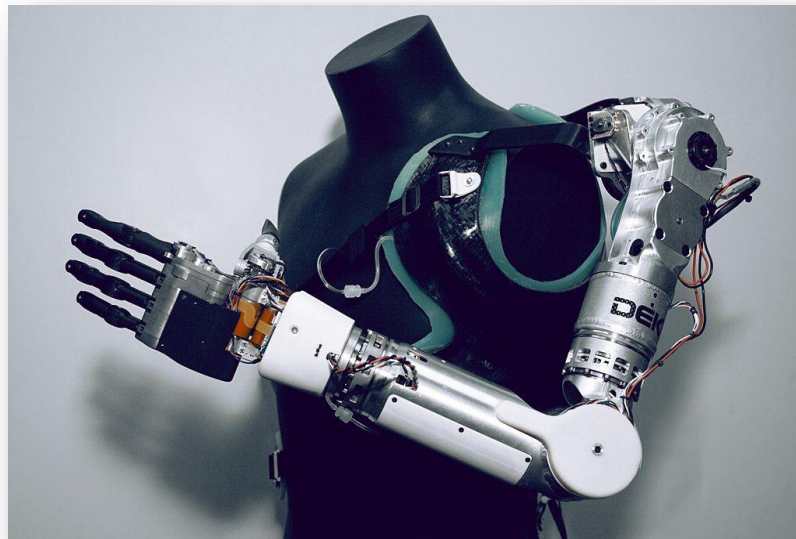


Εικόνα 1.1.4: Τηλεκατευθυνόμενο ρομπότ.

1.1.5 Φορετά ρομπότ (Wearable Robots)

Τα φορετά ρομπότ γενικά βελτιώνουν τις δυνατότητες που έχει ήδη ένα άτομο ή αντικαθιστούν τις δυνατότητες που έχει χάσει ένα άτομο. Ένα παράδειγμα ενός φορετού ρομποτικού συστήματος είναι ένα προσθετικό άκρο. Τα σύγχρονα προσθετικά άκρα μπορεί να είναι πολύπλοκα ρομποτικά συστήματα που μαθαίνουν να ανταποκρίνονται σε νευρικά σήματα που στέλνει το άτομο, παραδείγματος χάρι μπορεί να συνδεθούν με τους μύες στο στήθος, έτσι ώστε όταν το άτομο κινεί αυτούς τους μύες να μπορεί να προκαλέσει την κίνηση του βραχίονα με χαρακτηριστικούς τρόπους.

Ακόμη, υπάρχουν οι εξωσκελετοί (exoskeleton) οι οποίοι είναι σχεδιασμένοι να ενισχύσουν τις υπάρχουσες δυνατότητες ενός ατόμου (σε αντίθεση με τα προσθετικά, τα οποία δίνουν σε ένα άτομο μια ικανότητα που λείπει). Ο στόχος αυτών των φορετών ρομποτικών συστημάτων είναι η αύξηση της δύναμης του ατόμου, χωρίς να θυσιάζεται η ταχύτητα ή η ευελιξία. Επίσης οι εξωσκελετοί έχουν την δυνατότητα να βοηθήσουν ασθενείς που έχουν παραλύσει ή πάσχουν από κάποια νευρομυϊκή πάθηση, να ανακτήσουν ένα πλήρες φάσμα κινητικών ικανοτήτων.^[8]



Εικόνα 1.1.5: Φορετά ρομπότ.

1.2 Ρομποτικά συστήματα σταθερής βάσης

Ένα ρομποτικό σύστημα σταθερής βάσης (ή κοινώς ρομποτικός βραχίονας) είναι ένας προ-προγραμματισμένος τύπος ρομποτικού συστήματος, όπου έχει παρόμοιες λειτουργίες με ένα ανθρώπινο χέρι. Οι σύνδεσμοι (αρθρώσεις) ενός ρομποτικού βραχίονα, συνδέονται με τέτοιο τρόπο ώστε να επιτρέπουν είτε περιστροφική κίνηση είτε γραμμική μετατόπιση. Οι σύνδεσμοι αυτοί μπορούν να θεωρηθούν ότι αποτελούν μια κινηματική αλυσίδα, όπου το άκρο της ονομάζεται τελικός τελεστής και είναι ανάλογος με την ανθρώπινη παλάμη.^[9]

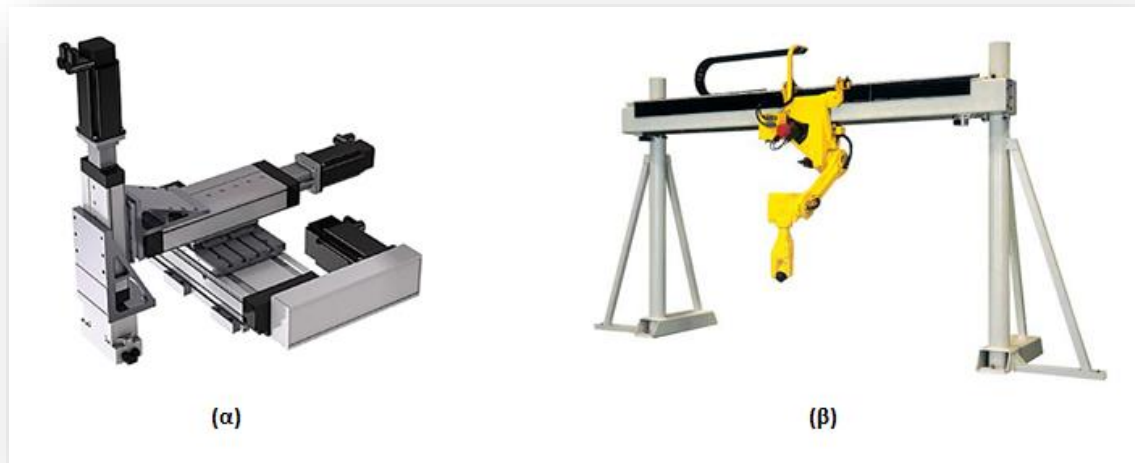
1.2.1 Καρτεσιανό ρομποτικό σύστημα (Cartesian)

Καρτεσιανό ρομποτικό σύστημα ονομάζεται ένα ρομποτικό σύστημα το οποίο κινείται σε τρεις ορθογώνιους άξονες (X,Y,Z) σύμφωνα με τις καρτεσιανές συντεταγμένες. Οι άξονες των καρτεσιανών συστημάτων εκτελούν συντονισμένη κίνηση, μέσω ενός κοινού ελεγκτή κίνησης.^[10]

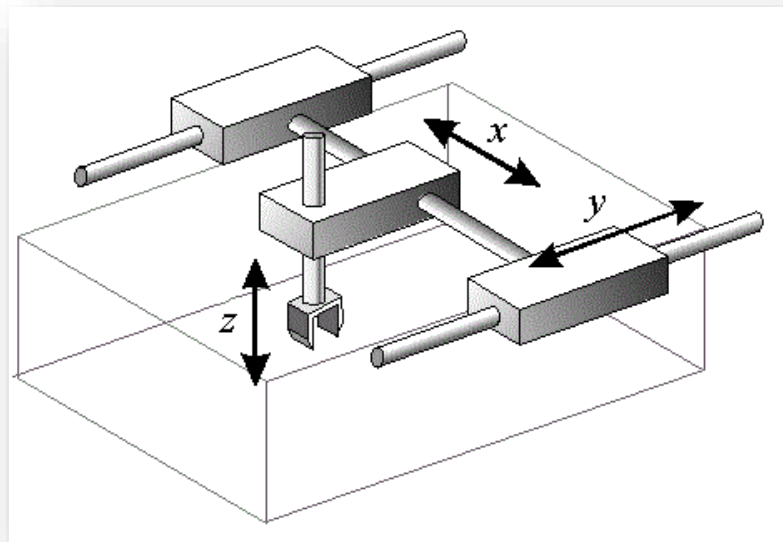
Σε αυτή την κατηγορία υπάρχουν τα gantry ρομπότ, τα οποία έχουν δυο άξονες βάσης (πχ δυο βάσεις για τον άξονα X), όπου τους δίνει το πλεονέκτημα των μεγάλων χώρων εργασίας και της καλύτερης

ακρίβειας τοποθέτησης, καθώς επίσης περιορίζονται λιγότερο από τους περιορισμούς του χώρου στο δάπεδο.^[11]

Τα ρομπότ καρτεσιανού συστήματος είναι ιδανικά για εφαρμογές όπως διαλογή και τοποθέτηση, εφαρμογές συναρμολόγησης, παλετοποίησης, εργαλειομηχανές CNC κ.α.^[12]



Εικόνα 1.2.1: Καρτεσιανό ρομποτικό σύστημα, όπου (α) απλό καρτεσιανό και (β) τύπου Gantry.



Εικόνα 1.2.2: Χώρος εργασίας Καρτεσιανού Συστήματος.

1.2.2 Ρομποτικό σύστημα SCARA

Τα αρχικά του SCARA είναι Selective Compliant Assembly Robot Arm (Ρομποτικός Βραχίονας Επιλεκτικής Συναρμολόγησης). Τα ρομποτικά συστήματα SCARA έχουν ευέλικτο εύρος κίνησης στους X,Y άξονες και μπορούν να κινούνται κατακόρυφα (Z άξονας), όμως η θέση του κάθετου άξονα παραμένει σταθερή. Χάρη στο συνδυασμό άκαμπτων και δυναμικών αξόνων, τα SCARA είναι

κατάλληλα για την αντιμετώπιση μιας ποικιλίας λειτουργιών χειρισμού υλικών, όπως για παράδειγμα χρησιμοποιούνται για την εκτέλεση εφαρμογών συναρμολόγησης ή τον χειρισμό ημιαγωγών.^[13]



Εικόνα 1.2.3: Ρομποτικό σύστημα SCARA.



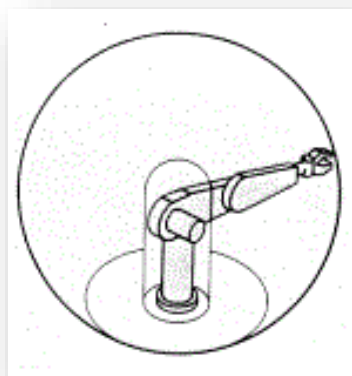
Εικόνα 1.2.4: Χώρος εργασίας ρομποτικού συστήματος SCARA.

1.2.3 Αρθρωτό ή 6 αξόνων ρομποτικό σύστημα (Articulated or 6 Axis)

Το αρθρωτό ρομποτικό σύστημα είναι ο πιο συνηθισμένος τύπος βιομηχανικού ρομποτικού βραχίονα. Η ομοιότητά του με έναν ανθρώπινο χέρι είναι ένας από τους λόγους που ξεχωρίζουν τόσο πολύ στο μυαλό μας. Ωστόσο, τα μηχανικά οφέλη αυτού του σχεδιασμού είναι ο πραγματικός λόγος για τον οποίο χρησιμοποιείται ευρέως. Ο σχεδιασμός του ρομποτικού βραχίονα συνδυάζει ένα ευρύ φάσμα περιστροφικής κίνησης και γραμμικής εμβέλειας με τα πλεονεκτήματα της κίνησης με μεγάλη ακρίβεια. Οι αρθρωτοί βραχίονες είναι ιδανικοί για συγκόλληση, χειρισμό υλικών, εργασίες συλλογής και της διαλογής και τοποθέτησης. Οι πολυάριθμοι άξονες και οι βαθμοί ελευθερίας τα καθιστά πιο ευπροσάρμοστα και ευέλικτα σε σχέση με τα υπόλοιπα ρομποτικά συστήματα.^[14]



Εικόνα 1.2.5: Αρθρωτό ρομποτικό σύστημα.



Εικόνα 1.2.6: Χώρος εργασίας αρθρωτού ρομποτικού συστήματος.

1.2.4 Κυλινδρικό ρομποτικό σύστημα (Cylindrical)

Το κυλινδρικό ρομποτικό σύστημα αποτελείται από έναν κύριο βραχίονα το οποίο μετακινείται πάνω κάτω και υπάρχει ένας ενσωματωμένος κύλινδρος ο οποίος δημιουργεί την κίνηση της επέκτασης και της αναδίπλωσης.

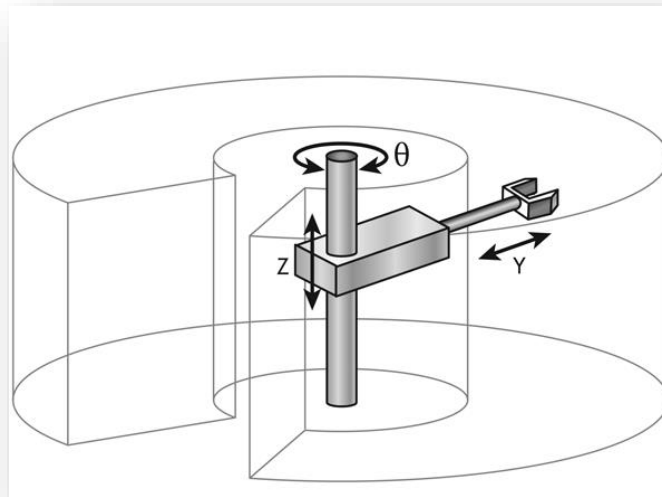
Στα περισσότερα κυλινδρικά ρομποτικά συστήματα, η περιστροφή γίνεται με την βοήθεια των γραναζιών και των κινητήρων, και η κάθετη κίνηση γίνεται με την βοήθεια ενός πνευματικού κυλίνδρου.

Επίσης αυτό το ρομποτικό σύστημα χρησιμοποιεί μια μέθοδο συντεταγμένων με τρεις διαστάσεις έχοντας επιλεγμένο έναν άξονα αναφοράς, με σχετικές αποστάσεις από αυτόν για τον καθορισμό των σημείων. Οι θέσεις των σημείων καθορίζονται χρησιμοποιώντας την απόσταση προς μια επιλεγμένη κατεύθυνση αναφοράς, τη σχετική κατεύθυνση του άξονα και την απόσταση στον κάθετο άξονα από ένα επιλεγμένο επίπεδο αναφοράς.

Οι κυλινδρικοί ρομποτικοί βραχίονες χρησιμοποιούνται κυρίως για συγκολλήσεις, χειρισμό μηχανημάτων χύτευσης, διαδικασίες λείανσης, παλετοποίησης, για γενικά εργαλεία χειρισμού καθώς και για λειτουργίες συναρμολόγησης.^[15]



Εικόνα 1.2.7: Κυλινδρικό ρομποτικό σύστημα.



Εικόνα 1.2.8: Χώρος εργασίας κυλινδρικού ρομποτικού συστήματος.

1.2.5 Ρομποτικό σύστημα Δέλτα (Delta)

Το ρομποτικό σύστημα Δέλτα είναι ένα σύστημα ρομπότ στο οποίο η βάση του είναι συνδεδεμένη σε αρθρωτά παραλληλόγραμμα. Αυτά τα παραλληλόγραμμα εκτελούν κινήσεις σε ένα χώρο εργασίας που έχει σχήμα θόλου και είναι γνωστό στον βιομηχανικό τομέα για την ικανότητα του να εκτελεί λεπτές και ακριβείς κινήσεις.

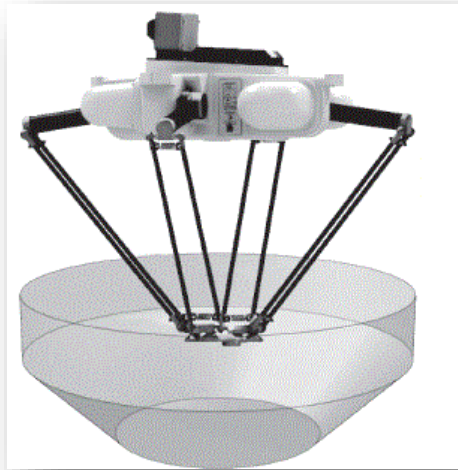
Ο σχεδιασμός τους, τους επιτρέπει να εκτελούν αποτελεσματικά τις επαναλαμβανόμενες κινήσεις με πολύ σταθερό ρυθμό. Ένα από τα βασικά πλεονεκτήματα του ρομποτικού συστήματος είναι ότι οι κινητήρες είναι εγκατεστημένοι στο πλαίσιο του ρομποτικού βραχίονα με αποτέλεσμα τα κινούμενα μέρη να είναι πολύ ελαφριά.

Το ρομποτικό σύστημα αυτό είναι γνωστό και ως ρομποτικό σύστημα αράχνης (Spider) ή παράλληλο ρομποτικό σύστημα (Parallel).

Τα ρομπότ συστήματος Δέλτα χρησιμοποιούνται κυρίως για λειτουργίες συναρμολόγησης υψηλής ακρίβειας, βιομηχανική συσκευασία, ιατρικές ή φαρμακευτικές εργασίες, επεξεργασία τροφής κ.α.^[16]



Εικόνα 1.2.9: Ρομποτικό σύστημα Δέλτα.



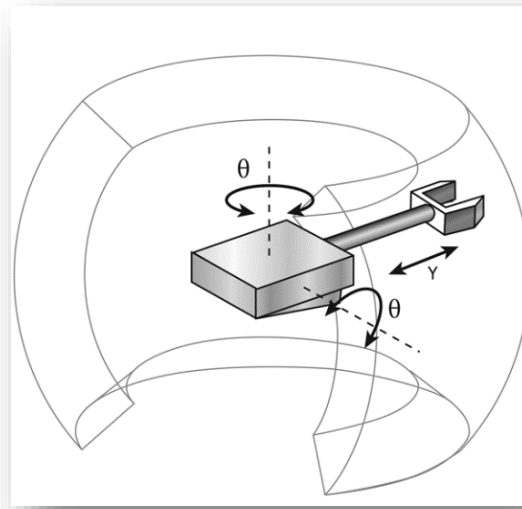
Εικόνα 1.2.10: Χώρος εργασίας ρομποτικού συστήματος Δέλτα.

1.2.6 Σφαιρικό ρομποτικό σύστημα (Spherical)

Το σφαιρικό ρομποτικό σύστημα (ή κυλινδρικό ρομποτικό σύστημα, Polar) αποτελείται από μια συνδεδεμένη γραμμική άρθρωση και δυο περιστροφικές αρθρώσεις, με συνδεδεμένο τον ρομποτικό βραχίονα σε μια βάση όπου ο σύνδεσμος του μπορεί να περιστρέφεται.

Θεωρείται ως ένας από τους πρώτους τύπους βιομηχανικών ρομπότ που έχουν αναπτυχθεί και χάρις αυτό το ρομποτικό σύστημα δημιουργήθηκε το ρομποτικό σύστημα έξι αξόνων, καθώς και πολλές άλλες εξελίξεις πάνω στην βιομηχανική ρομποτική.

Ορισμένες από τις εφαρμογές του ρομποτικού συστήματος είναι ο χειρισμός εργαλειομηχανών, λειτουργίες συναρμολόγησης, χύτευση, συγκόλληση με αέριο ή με τόξο, βαφή, κ.α.^[17]



Εικόνα 1.2.11: Χώρος εργασίας ενός σφαιρικού ρομποτικού συστήματος.

2. Οπτική ανίχνευση αντικειμένων και προβλήματα

Η ανίχνευση αντικειμένων είναι μια τεχνική που επιτρέπει την αναγνώριση και τον εντοπισμό των αντικειμένων σε μια εικόνα ή βίντεο. Με αυτό το είδος ταυτοποίησης και εντοπισμού, η ανίχνευση αντικειμένων μπορεί να χρησιμοποιηθεί για τη μέτρηση αντικειμένων σε ένα σημείο και για τον προσδιορισμό και την παρακολούθηση των ακριβών θέσεων τους, ενώ ταυτόχρονα χαρακτηρίζονται με ακρίβεια.^[18] Η ανίχνευση των αντικειμένων μπορεί να κατατάξει τα αντικείμενα σε τύπους, ενώ ταυτόχρονα εντοπίζει τις παρουσίες τους μέσα στην εικόνα βάζοντας ένα περίγραμμα οριοθέτησης (bounding box).

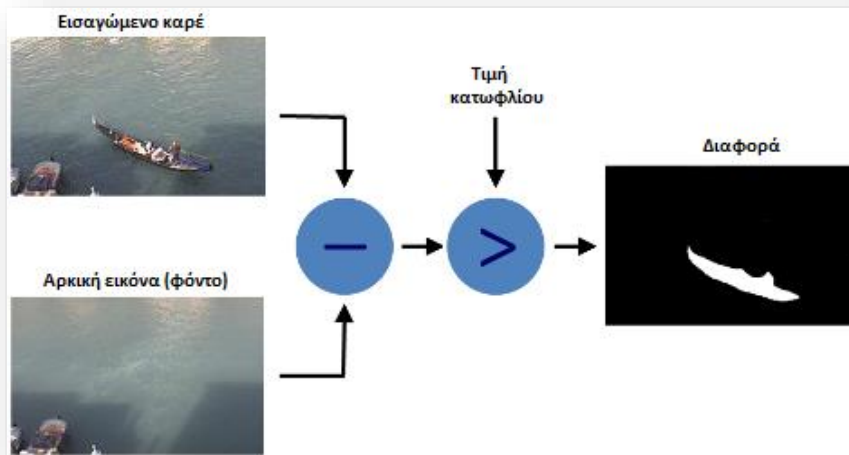
2.1 Τεχνική Background Subtraction

Η Background Subtraction τεχνική χρησιμοποιείται για την ανίχνευση αντικειμένου, συγκρίνοντας δύο διαφορετικά καρέ για να βρει την διαφορά και δημιουργεί έναν πίνακα απόστασης. Στην ουσία συγκρίνει την τιμή της διαφοράς, με την τιμή κατωφλίου (Threshold value).^[19]

Ένας πίνακας απόστασης είναι ένας τετραγωνικός πίνακας που περιέχει τις αποστάσεις, λαμβανόμενες κατά ζεύγη, μεταξύ των στοιχείων ενός συνόλου. Ανάλογα με τη σχετική εφαρμογή, η απόσταση που χρησιμοποιείται για τον καθορισμό αυτού του πίνακα μπορεί να είναι (ή να μην είναι) σε μετρικό σύστημα.^[20]

Η τιμή κατωφλίου δεν είναι προκαθορισμένη, αλλά υπολογίζεται χρησιμοποιώντας μερικά καρέ που θα δοθούν αρχικά στο σύστημα (αρχικά καρέ). Έτσι, το κύριο σενάριο είναι ότι εάν η διαφορά είναι μεγαλύτερη από την τιμή κατωφλίου, τότε επισημαίνεται ως κινούμενο αντικείμενο, διαφορετικά θα το λάβει ως αρχική εικόνα (φόντο).

Υπάρχουν αρκετά μοντέλα για την μοντελοποίηση του φόντου, όμως ο ευκολότερος τρόπος είναι μέσω μιας μεμονωμένης κλίμακας του γκρι / έγχρωμης εικόνας χωρίς κινούμενα αντικείμενα. Αυτή η εικόνα μπορεί να είναι μια εικόνα που τραβήχτηκε σε απουσία κίνησης ή υπολογίστηκε μέσω ενός χρονικού φίλτρου.^[19]



Εικόνα 2.1.1: Παράδειγμα μεθόδου Background Subtraction.

2.2 Οπτική αναγνώριση με Deep Learning

Η οπτική αναγνώριση με Deep Learning χρησιμοποιεί νευρωνικά δίκτυα, ώστε να εκπαιδευτεί αυτόματα για τον εντοπισμό των αντικειμένων μέσα σε εικόνες ή βίντεο. Υπάρχουν δυο βασικές μέθοδοι σε αυτή τη τεχνική:

- **Δημιουργία και εκπαίδευση ενός προσαρμοσμένου ανιχνευτή αντικειμένων:** Για να γίνει η εκπαίδευση ενός προσαρμοσμένου ανιχνευτή αντικειμένων από το μηδέν, πρέπει πρώτα να σχεδιαστεί η αρχιτεκτονική δικτύου, ώστε να γίνει η εκπαίδευση για τα ενδιαφερόμενα αντικείμενα.

Επίσης πρέπει να συνταχθεί ένα πολύ μεγάλο σύνολο ενδεικτικών δεδομένων για να την εκπαίδευση του νευρωνικού δικτύου, καθώς και η ρύθμιση των επιπέδων του νευρωνικού δικτύου γίνονται χειροκίνητα το οποίο απαιτεί πολύ χρόνο και δεδομένα εκπαίδευσης.

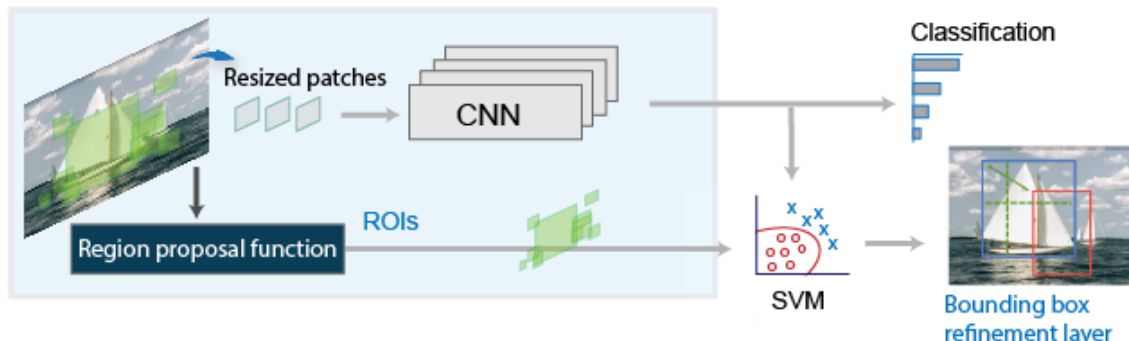
- **Χρήση ενός προ εκπαιδευμένου ανιχνευτή αντικειμένων:** Πολλές ροές εργασιών ανίχνευσης αντικειμένων με Deep Learning βοηθούν στην μεταφορά της μάθησης, και με αυτόν τον τρόπο υπάρχει μια προσέγγιση στη χρήση ενός προ εκπαιδευμένου νευρικού δικτύου, με ορισμένες βελτιώσεις για την επιθυμητή εφαρμογή. Με αυτό το τρόπο επιτυγχάνονται ταχύτερα αποτελέσματα, διότι οι ανιχνευτές αντικειμένων έχουν ήδη εκπαιδευτεί σε χιλιάδες, ή ακόμη και εκατομμύρια, εικόνες.^[21]

Παρακάτω αναφέρονται επτά βασικοί αλγόριθμοι.

2.2.1 Region-based Convolutional Neural Networks (R-CNN)

Με δεδομένη μια εικόνα εισόδου, το R-CNN ξεκινά εφαρμόζοντας έναν μηχανισμό που ονομάζεται «επιλεκτική αναζήτηση» για εξαγωγή περιοχών ενδιαφέροντος (ROI-Region of Interest), όπου κάθε ROI είναι ένα ορθογώνιο που μπορεί να αντιπροσωπεύει το όριο ενός αντικειμένου στην εικόνα. Μετά από αυτό, κάθε ROI διαμορφώνεται σε μια εικόνα σταθερής κλίμακας και τροφοδοτείται μέσω ενός νευρικού δικτύου για την παραγωγή χαρακτηριστικών εξόδου. Για όλες τις δυνατότητες εξόδου κάθε ROI, χρησιμοποιούνται SVM ταξινομητές για τον προσδιορισμό του είδους αντικειμένου (εάν υπάρχει) που περιέχεται στο ROI.^[22]

Ένα μεγάλο μειονέκτημα που έχει το R-CNN είναι ότι οι περιττοί υπολογισμοί χαρακτηριστικών σε μεγάλο αριθμό ROI (πάνω από 2000 ROI από μία εικόνα) οδηγούν σε εξαιρετικά αργή ταχύτητα ανίχνευσης. Επίσης, ο αλγόριθμος επιλεκτικής αναζήτησης είναι ένας σταθερός αλγόριθμος, επομένως, δεν υπάρχει μάθηση σε αυτό το στάδιο. Αυτό θα μπορούσε να οδηγήσει στη δημιουργία προτάσεων για κακές υποψήφιες περιοχές.^[23]

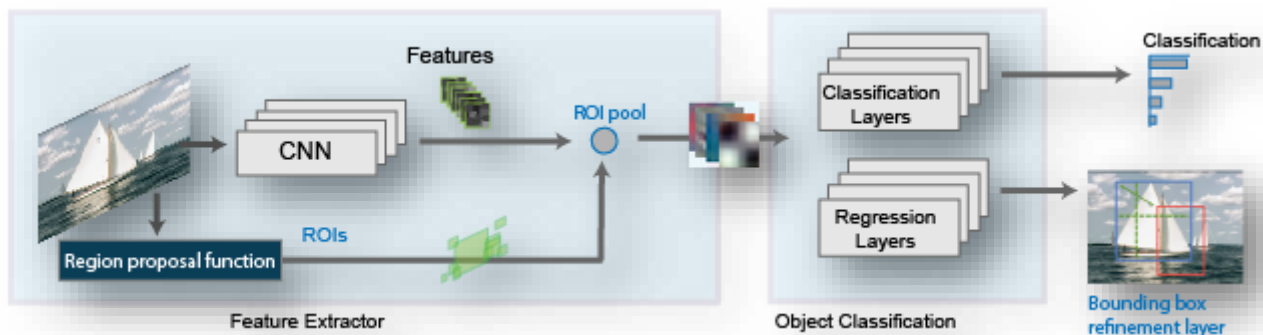


Εικόνα 2.2.1: R-CNN.

2.2.2 Fast R-CNN

Σε σύγκριση με ένα μοντέλο R-CNN, ένα μοντέλο Fast R-CNN χρησιμοποιεί ολόκληρη την εικόνα ως είσοδο CNN για εξαγωγή χαρακτηριστικών, αντί για κάθε προτεινόμενη περιοχή. Η επιλεκτική αναζήτηση εφαρμόζεται στην εικόνα και δημιουργεί n προτεινόμενες περιοχές, τα διαφορετικά σχήματα τους υποδεικνύουν περιοχές ενδιαφέροντος (ROI) διαφορετικών σχημάτων. Το Fast R-CNN εισάγει την ομαδοποίηση ROI, η οποία χρησιμοποιεί την έξοδο CNN και τα ROI ως είσοδο για την παραγωγή μιας συνένωσης των χαρακτηριστικών που εξάγονται από κάθε προτεινόμενη περιοχή και τροφοδοτούνται σε ένα πλήρως συνδεδεμένο στρώμα.

Κατά τη διάρκεια της πρόβλεψης κατηγορίας, το σχήμα της εξόδου του πλήρως συνδεδεμένου επιπέδου μετατρέπεται και πάλι σε $n \times q$ και πραγματοποιείται η χρήση της λειτουργίας softmax (q είναι ο αριθμός κατηγοριών και n είναι ο αριθμός των προτεινόμενων περιοχών). Κατά την πρόβλεψη του πλαισίου οριοθέτησης, το σχήμα της πλήρως συνδεδεμένης εξόδου μετατρέπεται και πάλι σε $n \times 4$. Αυτό σημαίνει ότι πραγματοποιείται η πρόβλεψη της κατηγορίας και του πλαισίου οριοθέτησης για κάθε προτεινόμενη περιοχή.^[23]



Εικόνα 2.2.2: Fast R-CNN.

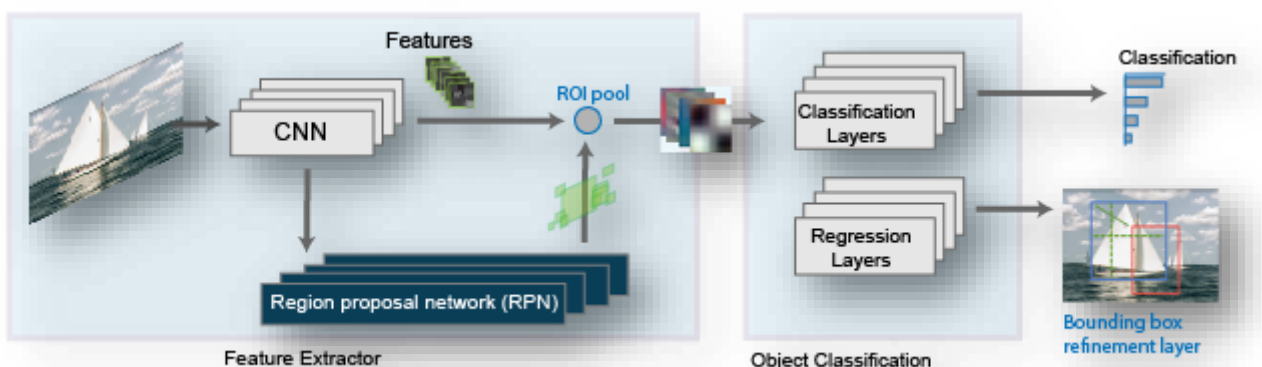
2.2.3 Faster R-CNN

Είναι η πρώτη ανίχνευση αντικειμένων Deep Learning η οποία πραγματοποιείται σχεδόν σε πραγματικό χρόνο, αφού δεν χρησιμοποιεί την επιλεκτική αναζήτηση. Η επιλεκτική αναζήτηση είναι μια αργή και χρονοβόρα διαδικασία που επηρεάζει την απόδοση του δικτύου.

Παρόμοια με το Fast R-CNN, η εικόνα παρέχεται ως είσοδος σε ένα δίκτυο το οποίο παρέχει έναν χάρτη χαρακτηριστικών. Αντί να χρησιμοποιεί αλγόριθμο επιλεκτικής αναζήτησης στο χάρτη δυνατοτήτων για τον προσδιορισμό των προτάσεων περιοχής, χρησιμοποιείται ένα ξεχωριστό δίκτυο για την πρόβλεψη των προτάσεων περιοχής. Οι προβλεπόμενες προτάσεις περιοχής αναδιαμορφώνονται έπειτα, χρησιμοποιώντας ένα στρώμα συγκέντρωσης ROI το οποίο στη συνέχεια χρησιμοποιείται για την ταξινόμηση της εικόνας εντός της προτεινόμενης περιοχής και την πρόβλεψη των τιμών μετατόπισης για τα πλαίσια οριοθέτησης.

Ως μέρος του ταχύτερου μοντέλου R-CNN, το δίκτυο πρότασης περιοχής εκπαιδεύεται μαζί με το υπόλοιπο μοντέλο. Επιπλέον, το Faster R-CNN περιλαμβάνει τις προβλέψεις για την κατηγορία και το πλαίσιο οριοθέτησης στην ανίχνευση αντικειμένων, καθώς και τις προβλέψεις για την κατηγορία και το πλαίσιο οριοθέτησης για τα πλαίσια αγκύρωσης (anchor boxes) στο δίκτυο προτάσεων περιοχής.

Τέλος, το δίκτυο πρότασης περιοχής μπορεί να μάθει πώς να δημιουργεί προτεινόμενες περιοχές υψηλής ποιότητας, γεγονός που μειώνει τον αριθμό των προτεινόμενων περιοχών διατηρώντας παράλληλα την ακρίβεια της ανίχνευσης αντικειμένων.^[23]



Εικόνα 2.2.3: Faster R-CNN.

2.2.4 Region-based Fully Convolutional Network (R-FCN)

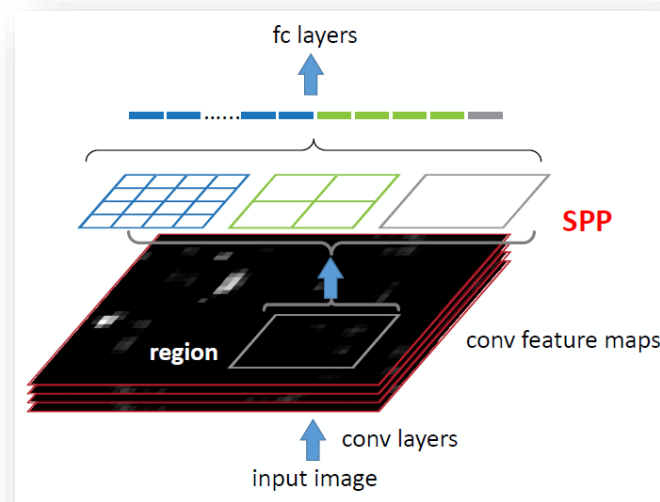
Σε αντίθεση με άλλους region-based ανιχνευτές που εφαρμόζουν ένα δαπανηρό υποδίκτυο ανά περιοχή, όπως το Fast R-CNN ή το Faster R-CNN, αυτός ο συμβατικός, σχεδόν με όλους τους κοινόχρηστους υπολογισμούς σε ολόκληρη την εικόνα.

Το R-FCN αποτελείται από κοινόχρηστες, πλήρως συνελκτικές αρχιτεκτονικές (convolutional architecture) όπως στην περίπτωση του FCN όπου αποδίδει καλύτερο αποτέλεσμα από το Faster R-CNN. Σε αυτόν τον αλγόριθμο, όλα τα επίπεδα μαθημάτων που μπορούν να διδαχθούν είναι ομοιόμορφα και έχουν σχεδιαστεί για να ταξινομούν τις ROI σε κατηγορίες αντικειμένων και υπόβαθρα (background).^[24]

2.2.5 Spatial Pyramid Pooling Net (SPPNet)

Το SPPNet είναι μια δομή δικτύου που μπορεί να δημιουργήσει αναπαράσταση σταθερού μήκους ανεξάρτητα από το μέγεθος ή την κλίμακα της εικόνας. Η συγκεντρωτική πυραμίδα (pyramid pooling) είναι ανθεκτική στις παραμορφώσεις αντικειμένων και το SPPNet βελτιώνει όλες τις μεθόδους ταξινόμησης εικόνας που βασίζονται στο CNN.

Χρησιμοποιώντας το SPPNet, πραγματοποιείται ο υπολογισμός των χαρακτηριστικών χαρτών από ολόκληρη την εικόνα μόνο μία φορά και, στη συνέχεια, συγκεντρώνονται τα χαρακτηριστικά σε αυθαίρετες περιοχές (υπό εικόνες) για την δημιουργία αναπαραστάσεων σταθερού μήκους για την εκπαίδευση των ανιχνευτών. Αυτή η μέθοδος αποφεύγει τον επαναλαμβανόμενο υπολογισμό των συνελκτικών χαρακτηριστικών.^[24]

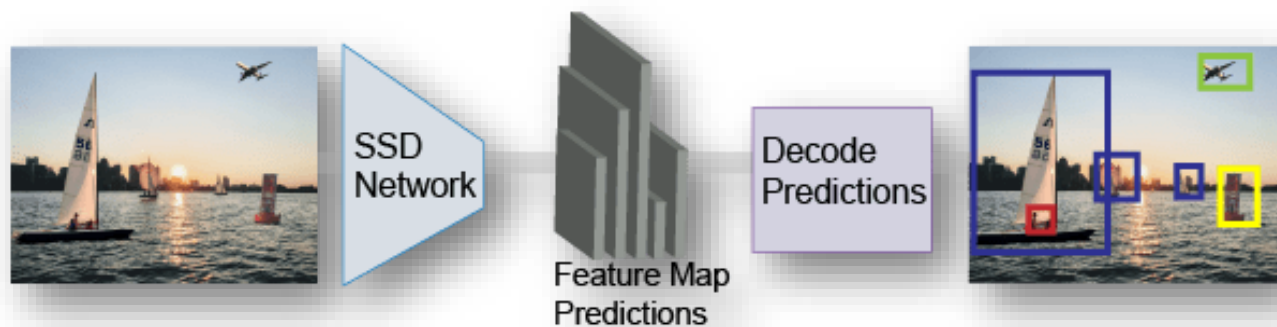


Εικόνα 2.2.4: SPPNet

2.2.6 Single Shot Detector (SSD)

Το (SSD) είναι μια μέθοδος για την ανίχνευση αντικειμένων σε εικόνες χρησιμοποιώντας ένα μόνο νευρικό δίκτυο. Η προσέγγιση SSD διαχωρίζει τον χώρο εξόδου των οριοθετημένων πλαισίων σε ένα σύνολο προεπιλεγμένων πλαισίων σε διαφορετικές κλίμακες διαστάσεων. Μετά το διαχωρισμό, η μέθοδος κλιμακώνεται ανά τοποθεσία χάρτη χαρακτηριστικών. Το δίκτυο SSD συνδυάζει προβλέψεις από πολλαπλούς χάρτες χαρακτηριστικών με διαφορετικές αναλύσεις για τον απλό χειρισμό αντικειμένων διάφορων μεγεθών. Αυτή η τεχνική έχει τρία βασικά πλεονεκτήματα:

- Το SSD εξαλείφει εντελώς τη δημιουργία προτάσεων και των επόμενων σταδίων δειγματοληψίας pixel ή χαρακτηριστικών και ενσωματώνει όλους τους υπολογισμούς σε ένα μόνο δίκτυο.
- Εύκολη εκπαίδευση και απλή ενσωμάτωση σε συστήματα που απαιτούν ένα στοιχείο ανίχνευσης.
- Το SSD έχει ανταγωνιστική ακρίβεια σε μεθόδους που χρησιμοποιούν ένα πρόσθετο βήμα πρότασης αντικειμένου και είναι πολύ πιο γρήγορο παρέχοντας ένα ενοποιημένο πλαίσιο τόσο για εκπαίδευση όσο και για συμπεράσματα.^[24]



Εικόνα 2.2.5: SSD.

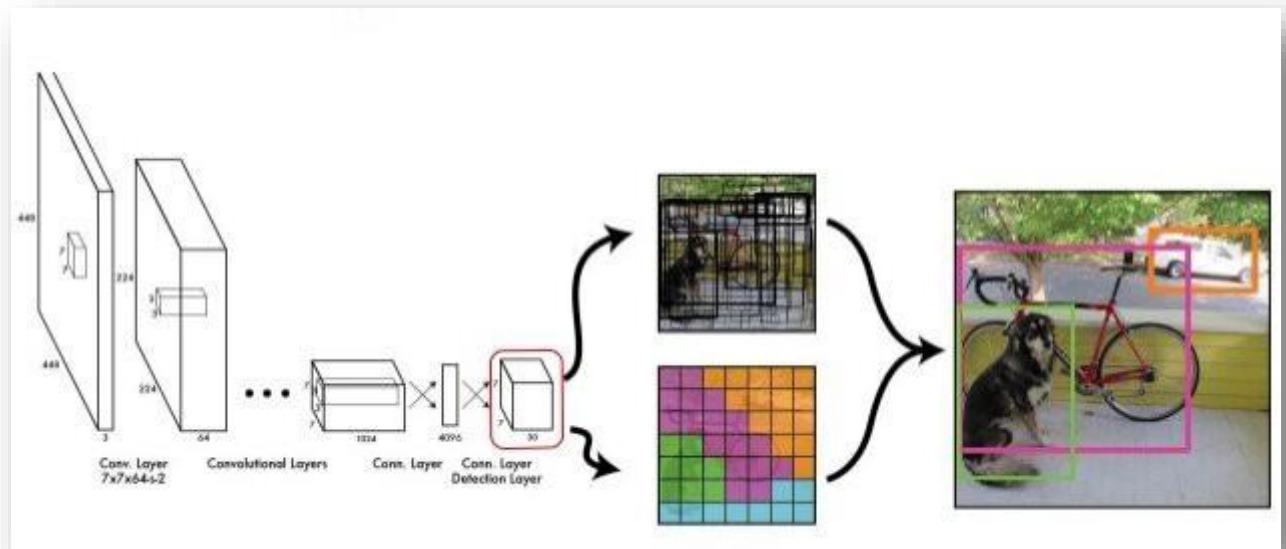
2.2.7 You Only Look Once (YOLO)

Όλοι οι προηγούμενοι αλγόριθμοι ανίχνευσης αντικειμένων χρησιμοποιούν περιοχές για να εντοπίσουν το αντικείμενο μέσα στην εικόνα. Το δίκτυο δεν εξετάζει την πλήρη εικόνα, αλλά εξετάζει τμήματα της εικόνας που έχουν μεγάλες πιθανότητες να περιέχουν το αντικείμενο.

Το YOLO εκπαιδεύεται σε πλήρεις εικόνες και βελτιστοποιεί άμεσα την απόδοση ανίχνευσης. Με το YOLO, ένα μόνο CNN προβλέπει ταυτόχρονα πολλαπλά πλαίσια οριοθέτησης και πιθανότητες κλάσης για αυτά τα πλαίσια. Προβλέπει επίσης όλα τα πλαίσια οριοθέτησης σε όλες τις κατηγορίες για μια εικόνα ταυτόχρονα. Χωρίζει την εικόνα εισόδου σε πλέγμα $S \times S$. Εάν το κέντρο ενός αντικειμένου πέσει σε ένα κελί πλέγματος, αυτό το κελί πλέγματος είναι υπεύθυνο για την ανίχνευση αυτού του αντικειμένου. Κάθε κελί πλέγματος προβλέπει B οριοθετημένα πλαίσια και βαθμούς εμπιστοσύνης για αυτά τα πλαίσια. Οι βαθμοί εμπιστοσύνης αντικατοπτρίζουν το πόσο σίγουρο είναι το μοντέλο, ότι το πλαίσιο περιέχει ένα αντικείμενο και επίσης πόσο ακριβές θεωρεί ότι είναι το πλαίσιο που προέβλεψε.

Κάθε πλαίσιο οριοθέτησης αντιπροσωπεύεται από 6 αριθμούς ($p_c, b_x, b_y, b_h, b_w, c$), όπου το p_c είναι η αξιοπιστία ότι υπάρχει ένα αντικείμενο στο πλαίσιο οριοθέτησης. Τα b_x, b_y, b_h, b_w αντιπροσωπεύουν το ίδιο το πλαίσιο οριοθέτησης. Το c είναι ένας φορέας που περιέχει τις πιθανότητες κλάσης. Επίσης ορίζονται τα πλαίσια αγκύρωσης διερευνώντας τα δεδομένα εκπαίδευσης για την επιλογή των λογικών αναλογιών ύψους και πλάτους που αντιπροσωπεύουν τις διαφορετικές κατηγορίες. Ένα πλαίσιο αγκύρωσης μπορεί να έχει πολλαπλά οριοθετημένα πλαίσια. Για κάθε πλαίσιο αγκύρωσης υπολογίζεται το στοιχείο $p_c * c[i]$ με γνώμονα το στοιχείο και εξάγεται ένας βαθμός πιθανότητας του πλαισίου που περιέχει μια συγκεκριμένη κλάση. Η κλάση που σχετίζεται με το μέγιστο βαθμό ανατίθεται στο πλαίσιο αγκύρωσης μαζί με το ίδιο το αποτέλεσμα.

Στη συνέχεια, απαλλάσσονται τα πλαίσια με το χαμηλό αποτέλεσμα χρησιμοποιώντας ένα κατώφλι πιθανότητας αποτελέσματος. Έπειτα χρησιμοποιείται το NMS (Non Maxima Suppression) στο οποίο πραγματοποιείται η επιλογή μόνο ενός πλαισίου, όταν πολλά πλαίσια αλληλεπικαλύπτονται μεταξύ τους και εντοπίζουν το ίδιο αντικείμενο.^[23]



Εικόνα 2.2.6: YOLO.

2.3 Οπτική ανίχνευση με Machine Learning

Η οπτική ανίχνευση με Machine Learning είναι μια παλιά μέθοδος και θεωρείται ως η «παραδοσιακή οπτική ανίχνευση των αντικειμένων», η οποία έχει άλλη προσέγγιση από αυτή του Deep Learning.

Ακολουθούν τις ίδιες βασικές μεθόδους, δηλαδή μπορεί να γίνει η δημιουργία και εκπαίδευση ενός προσαρμοσμένου ανιχνευτή αντικειμένων ή μπορεί να χρησιμοποιηθεί ένας προ εκπαιδευμένος ανιχνευτής αντικειμένων, αλλά η επιλογή των χαρακτηριστικών αναγνώρισης ενός αντικειμένου πρέπει να γίνει χειροκίνητα σε σχέση με το Deep Learning στο οποίο γίνεται αυτόματα.

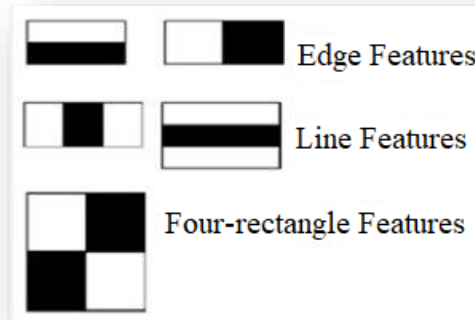
Η οπτική αναγνώριση με Deep Learning για να λειτουργήσει καλύτερα χρειάζεται αρκετές ενδεικτικές εικόνες εκπαίδευσης και αρκετή γραφική ισχύ (GPU) για να μειώσει το χρόνο που απαιτείται για την εκπαίδευση του μοντέλου, κάτι στο οποίο η τεχνική με Machine Learning δεν απαιτεί σε τόσο μεγάλο βαθμό.^[21] Παρακάτω αναφέρονται οι τρεις βασικές τεχνικές.

2.3.1 Viola Jones

Η Viola Jones τεχνική, είναι μια τεχνική ανίχνευσης αντικειμένων και χρησιμοποιείται κυρίως για την ανίχνευση (όχι αναγνώριση) των ανθρώπινων προσώπων. Ο τρόπος ανίχνευσης που χρησιμοποιεί είναι τα παράθυρα ανίχνευσης, δηλαδή περνάει από όλες τις πιθανές τοποθεσίες και κλίμακες σε μια εικόνα για να ελέγξει αν οποιοδήποτε παράθυρο περιέχει ανθρώπινο πρόσωπο.^[29] Η τεχνική Viola Jones αποτελείται από τέσσερα στάδια:

1. **Haar-like χαρακτηριστικό:** Το Haar-like χαρακτηριστικό λαμβάνει υπόψη γειτονικές ορθογώνιες περιοχές σε μια συγκεκριμένη τοποθεσία σε ένα παράθυρο ανίχνευσης, συνοψίζει τις εντάσεις των pixel σε κάθε περιοχή και υπολογίζει τη διαφορά μεταξύ αυτών των αθροισμάτων. Αυτή η διαφορά χρησιμοποιείται στη συνέχεια για την κατηγοριοποίηση των υποδιαιρέσεων μιας εικόνας.^[25] Για παράδειγμα όλα τα ανθρώπινα πρόσωπα έχουν ορισμένα ίδια χαρακτηριστικά σε μία εικόνα, για παράδειγμα τα pixel στη περιοχή των ματιών είναι πιο σκούρα από τα γειτονικά και η περιοχή της μύτης είναι πιο φωτεινή από την περιοχή των ματιών. Ένας τρόπος αναγνώρισης για το ποια περιοχή είναι πιο ανοιχτή ή

πιο σκοτεινή είναι το άθροισμα των τιμών pixel και των δύο περιοχών και η σύγκριση τους. Το άθροισμα των τιμών των pixel στη σκοτεινότερη περιοχή θα είναι μικρότερο από το άθροισμα των pixel στην πιο ανοιχτή περιοχή. Εάν η μία πλευρά είναι πιο φωτεινή από την άλλη, μπορεί να είναι μια άκρη ενός φρυδιού ή μερικές φορές το μεσαίο τμήμα μπορεί να είναι πιο φωτεινό από τα γύρω ορθογώνια περιοχές, τα οποία μπορούν να ερμηνευτούν ως μύτη.^[26]

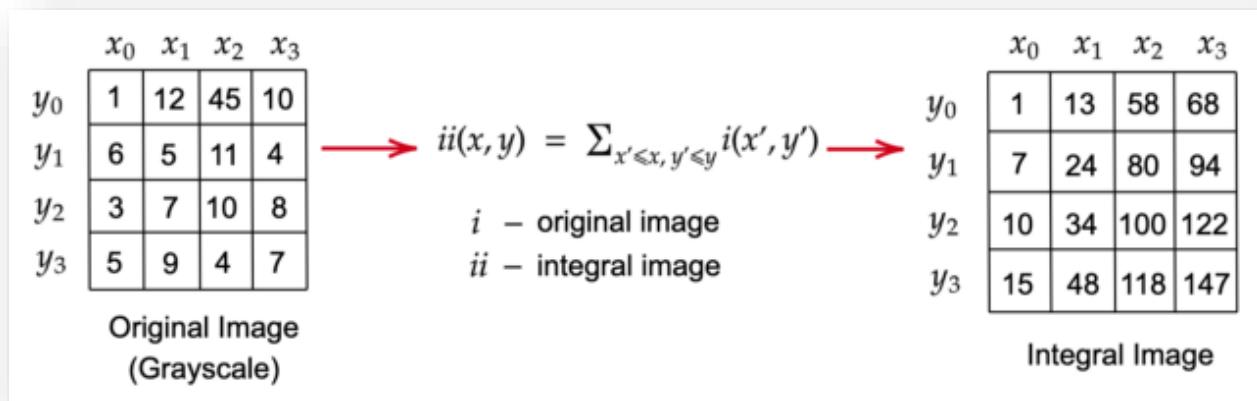


Εικόνα 2.3.1: Παράθρορα αντίγνευσης.



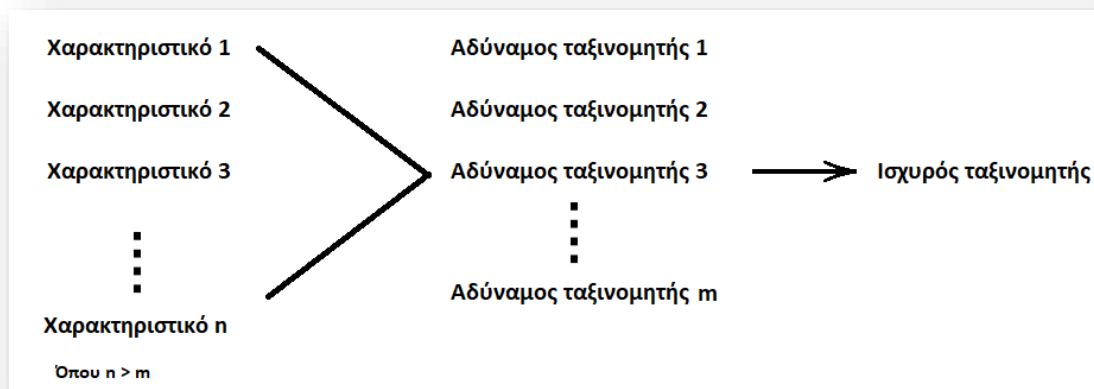
Εικόνα 2.3.2: Haar-like χαρακτηριστικό.

- 2. Δημιουργία ακέραιης εικόνας (Integral image):** Η ακέραιη εικόνα είναι η ενδιάμεση αναπαράσταση μιας εικόνας όπου η τιμή για την θέση (x, y) στην ακέραιη εικόνα ισούται με το άθροισμα των pixel πάνω και αριστερά (συμπεριλαμβανομένου) της θέσης (x, y) στην αρχική εικόνα.^[27] Με αυτό το τρόπο επιταχύνεται ο υπολογισμός των pixel σε μια ορθογώνια περιοχή, καθώς πρέπει πραγματοποιείται ο υπολογισμός μόνο σε τέσσερις άκρες του ορθογώνιου.



Εικόνα 2.3.3: Δημιουργία ακέραιας εικόνας.

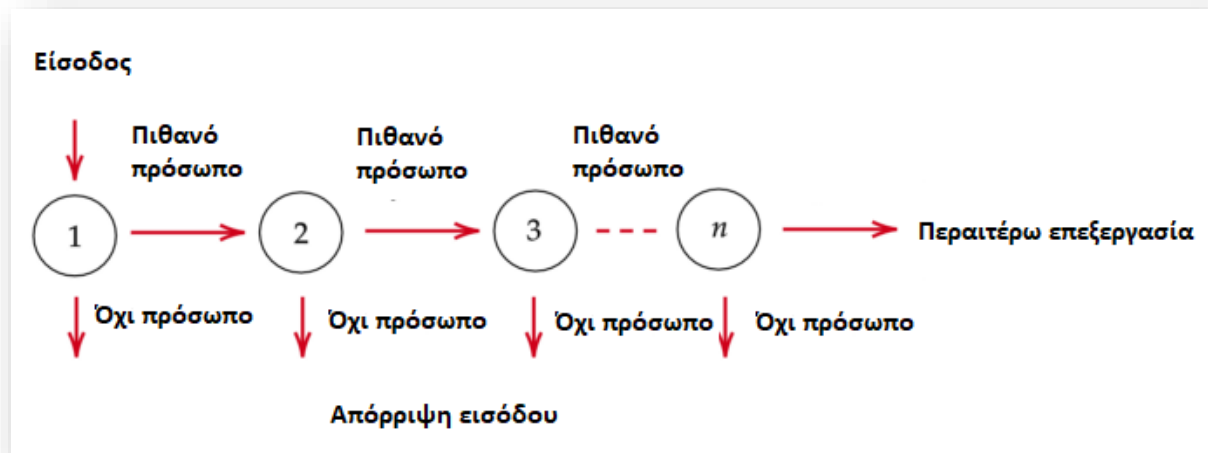
3. Εκπαίδευση με τον αλγόριθμο AdaBoost: Ο αλγόριθμος AdaBoost (Adaptive Boosting) είναι ένας αλγόριθμος μηχανικής εκπαίδευσης (machine learning) για την επιλογή του καλύτερου υποσυνόλου χαρακτηριστικών μεταξύ όλων των διαθέσιμων Haar-like χαρακτηριστικών. Η έξοδος του αλγορίθμου είναι ένας ταξινομητής (γνωστή ως συνάρτηση πρόβλεψης, συνάρτηση υπόθεσης) που ονομάζεται «ισχυρός ταξινομητής». Ένας ισχυρός ταξινομητής αποτελείται από γραμμικούς συνδυασμούς «αδύναμων ταξινομητών». Για την εύρεση των αδύναμων ταξινομητών, ο αλγόριθμος τρέχει για T επαναλήψεις όπου το T είναι ο αριθμός των αδύναμων ταξινομητών που πρόκειται να βρεθούν και ορίζονται από τον χρήστη. Σε κάθε επανάληψη, ο αλγόριθμος βρίσκει το ποσοστό σφάλματος για όλες τις δυνατότητες και, στη συνέχεια, επιλέγει τη λειτουργία με το χαμηλότερο ποσοστό σφάλματος για αυτήν την επανάληψη.^[27]



Εικόνα 2.3.4: Αλγόριθμος AdaBoost.

4. Ταξινομητής Cascade: Ο ταξινομητής Cascade είναι ένας ταξινομητής πολλαπλών σταδίων που μπορεί να εκτελέσει ανίχνευση γρήγορα και με ακρίβεια. Κάθε στάδιο αποτελείται από έναν ισχυρό ταξινομητή που παράγεται από τον αλγόριθμο AdaBoost. Από το ένα στάδιο στο άλλο, αυξάνεται ο αριθμός των ασθενών ταξινομητών σε έναν ισχυρό ταξινομητή. Μια είσοδος αξιολογείται σε διαδοχική βάση (στάδιο προς στάδιο). Εάν ένας ταξινομητής για ένα συγκεκριμένο στάδιο εμφανίζει αρνητικό αποτέλεσμα, η είσοδος απορρίπτεται αμέσως. Σε περίπτωση που η έξοδος είναι θετική, η είσοδος προωθείται στο

επόμενο στάδιο. Σύμφωνα με τους Viola και Jones, αυτή η προσέγγιση πολλαπλών σταδίων επιτρέπει την κατασκευή απλούστερων ταξινομητών, οι οποίοι μπορούν στη συνέχεια να χρησιμοποιηθούν για την απόρριψη των περισσότερων αρνητικών (μη προσώπων) εισόδων γρήγορα, ενώ αφιερώνουν περισσότερο χρόνο στη θετική (πρόσωπα) είσοδο.^[27]



Εικόνα 2.3.5: Ταξινομητής Cascade

2.3.2 Ιστόγραμμα Προσανατολισμένων Κλίσεων (HOG)

Το HOG (Histogram of Oriented Gradients), είναι μια μορφή περιγραφής χαρακτηριστικών που χρησιμοποιείται συχνά για την εξαγωγή χαρακτηριστικών από δεδομένα εικόνας.

Εστιάζει στη δομή ή το σχήμα ενός αντικειμένου. Στην περίπτωση χαρακτηριστικών άκρων, αναγνωρίζει μόνο εάν το στοιχείο της εικόνας (pixel) είναι άκρο ή όχι. Το HOG μπορεί επίσης να παρέχει την κατεύθυνση των άκρων, όπου αυτό πραγματοποιείται με την εξαγωγή της κλίσης και του προσανατολισμού των άκρων.

Επίσης, αυτοί οι προσανατολισμοί υπολογίζονται σε μικρά τμήματα, δηλαδή η πλήρης εικόνα χωρίζεται σε μικρότερες περιοχές και για κάθε περιοχή, υπολογίζονται οι κλίσεις και ο προσανατολισμός.

Τέλος, το HOG δημιουργεί ένα ιστόγραμμα για καθεμία από αυτές τις περιοχές ξεχωριστά. Τα ιστογράμματα δημιουργούνται χρησιμοποιώντας τις κλίσεις και τους προσανατολισμούς των τιμών των pixel.^[28]

2.3.3 Μοντέλο με δυνατότητα παραμόρφωσης (DPM)

Το DPM (Deformable Part-based Model) είναι μια βελτίωση του HOG. Η εκπαιδευτική διαδικασία περιλαμβάνει την εκμάθηση ενός σωστού τρόπου αποσύνθεσης ενός αντικειμένου, και το συμπέρασμα περιλαμβάνει το σύνολο ανιχνεύσεων διαφορετικών τμημάτων αντικειμένων. Για παράδειγμα, το πρόβλημα της ανίχνευσης ενός αυτοκινήτου μπορεί να θεωρηθεί ως η ανίχνευση του παραθύρου, του αμαξώματος και των τροχών του.

Ο ανιχνευτής DPM αποτελείται από ένα φίλτρο ρίζας και έναν αριθμό μερικών φίλτρων. Μια αδύναμη εποπτευόμενη μέθοδος εκμάθησης αναπτύσσεται στο DPM όπου όλες οι διαμορφώσεις

(μέγεθος, τοποθεσία κ.λπ.) των μερικών φίλτρων μπορούν να μαθευτούν αυτόματα ως λανθάνουσες μεταβλητές.^[29]

2.4 Παραμόρφωση εικόνας της κάμερας

Η παραμόρφωση είναι ένας όρος που χρησιμοποιείται για τον ορισμό μιας σειράς εκτροπών που συμβαίνουν με ορισμένες οπτικές επιλογές (όπως τύπος φακού) και προοπτικές λήψης.

Για την οπτική αναγνώριση των αντικειμένων αυτή η παραμόρφωση δημιουργεί προβλήματα, όπως την λανθασμένη αντίληψη του μεγέθους του αντικειμένου.

Παρακάτω αναφέρονται τα δυο είδη παραμορφώσεων.

1. Γεωμετρική παραμόρφωση (Optical Distortion):

Γεωμετρική (ή οπτική) παραμόρφωση ονομάζεται η παραμόρφωση στη γεωμετρία της εικόνας, λόγω της σχεδίασης του φακού (τα οπτικά στοιχεία που βρίσκονται μέσα του, η οργάνωση και «χωροταξία» τους, κλπ.). Υπάρχουν τρεις βασικοί τύποι οπτικής παραμόρφωσης, όμως πρώτα θα γίνει μια αναφορά για έναν φακό χωρίς παραμόρφωση:

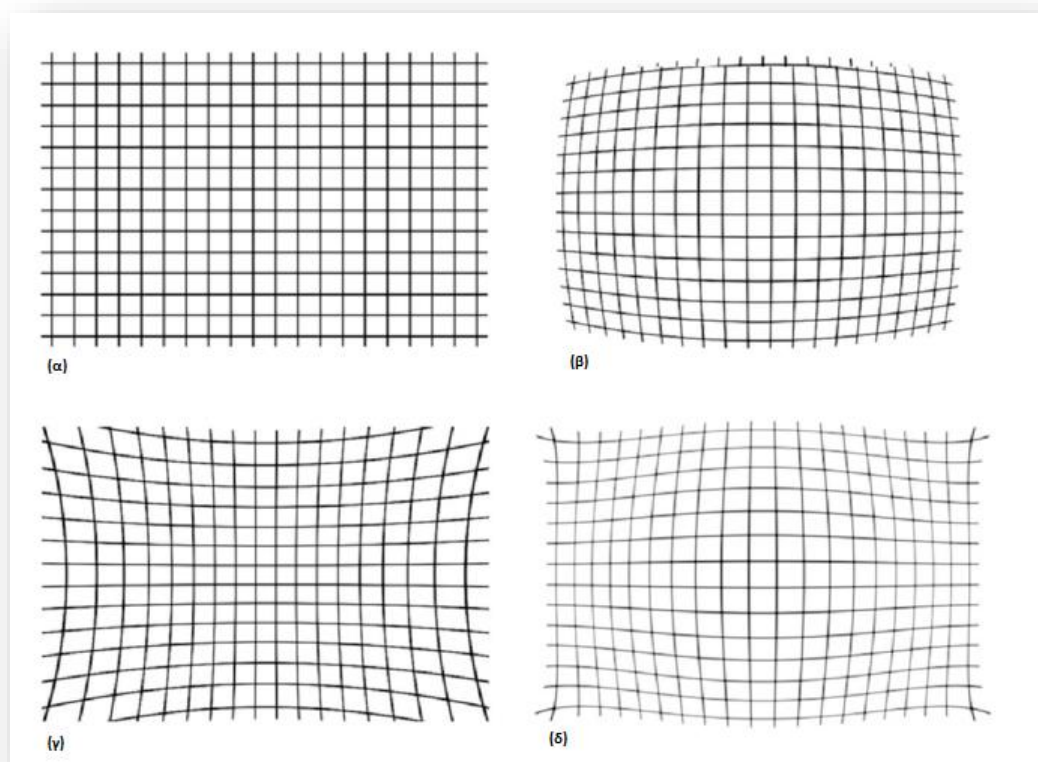
- **Χωρίς παραμόρφωση:** Τέτοιοι φακοί είναι πολύ σπάνιοι. Οι αρκετά καλοί φακοί μπορούν να μειώσουν σημαντικά την παραμόρφωση, σε τέτοιο βαθμό, όπου δεν είναι ορατό στο ανθρώπινο μάτι. Αρκετοί φακοί με ζουμ, και κυρίως φακοί με υψηλή εστίαση υποφέρουν από πολλαπλούς τύπους παραμόρφωσης σε διαφορετικά εστιακά μήκη.
- **Βαρελοειδής παραμόρφωση (Barrel Distortion):** Όταν οι ευθείες γραμμές είναι καμπυλωμένες προς τα μέσα σε σχήμα βαρελιού, αυτός ο τύπος παραμόρφωσης ονομάζεται «βαρελοειδής παραμόρφωση». Συνήθως παρατηρείται σε ευρυγώνιους φακούς, διότι το οπτικό πεδίο του φακού είναι μεγαλύτερο από το μέγεθος του αισθητήρα εικόνας και ως εκ τούτου πρέπει να «συμπιεστεί» για να ταιριάζει. Έτσι, οι ευθείες γραμμές κάμπτονται ορατά προς τα μέσα, ειδικά προς τα άκρα του πλαισίου. Με ένα λογισμικό επεξεργασίας μπορεί εύκολα να διορθωθεί αυτό το πρόβλημα παραμόρφωσης αρκεί ο φακός να έχει ένα υποστηρικτικό προφίλ στη βάση δεδομένων.
- **Κυρτοειδής παραμόρφωση (Pincushion Distortion):** Η κυρτοειδής παραμόρφωση είναι το ακριβώς αντίθετο της βαρελοειδούς παραμόρφωσης καθώς οι ευθείες γραμμές κάμπτονται από το κέντρο προς τα έξω. Αυτός ο τύπος παραμόρφωσης παρατηρείται συνήθως σε τηλεφακούς και συμβαίνει λόγω της μεγέθυνσης της εικόνας που αυξάνεται προς τα άκρα του πλαισίου από τον οπτικό άξονα. Σε αυτή την περίπτωση, το οπτικό πεδίο είναι μικρότερο από το μέγεθος του αισθητήρα εικόνας και συνεπώς πρέπει να «τεντωθεί» για να χωρέσει.

Όπως στην περίπτωση της βαρελοειδής παραμόρφωσης, έτσι και στην περίπτωση της κυκλοειδής παραμόρφωσης το πρόβλημα μπορεί να αντιμετωπιστεί με τη βοήθεια ενός λογισμικού επεξεργασίας.

- **Μυστακοειδής παραμόρφωση ή σύνθετη παραμόρφωση (Mustache Distortion, Complex Distortion):** Είναι βασικά ένας συνδυασμός των παραπάνω

παραμορφώσεων και είναι ο πιο αλλοιωτικός από τους τύπους οπτικής παραμόρφωσης. Οι ευθείες γραμμές εμφανίζονται με καμπύλες προς τα μέσα, προς το κέντρο του πλαισίου και στη συνέχεια με καμπύλες προς τα έξω στις ακραίες γωνίες.

Για να πραγματοποιηθεί η διόρθωση σε αυτό τον τύπο παραμόρφωσης, απαιτείται εξειδικευμένο λογισμικό διότι με απλό λογισμικό επεξεργασίας όταν θα διορθώνεται η βαρελοειδής παραμόρφωση, υπάρχει μεγάλη περίπτωση οι ακραίες γραμμές να κάμπτονται περισσότερο, καθώς και όταν θα διορθώνεται η κυρτοειδής παραμόρφωση, υπάρχει μεγάλη περίπτωση να επιτευχθεί μια βαρελοειδής παραμόρφωση προς το κέντρο. Στην μόνη περίπτωση που μπορεί να χρησιμοποιηθεί ένα απλό λογισμικό επεξεργασίας είναι όταν έχει δημιουργηθεί ένα συγκεκριμένο προφίλ φακού για την καταπολέμηση της συγκεκριμένης παραμόρφωσης.^[30]



Εικόνα 2.4.1: Χωρίς παραμόρφωση (α), παραμόρφωση βαρελιού (β), παραμόρφωση μαξιλαριού (γ), παραμόρφωση μουστάκι (δ).

2. Παραμόρφωση προοπτικής (Perspective Distortion):

Σε αντίθεση με την γεωμετρική παραμόρφωση, η παραμόρφωση προοπτικής δεν έχει καμία σχέση με τα στοιχεία του φακού και ως εκ τούτου, δεν είναι σφάλμα φακού. Η προοπτική παραμόρφωσης αναφέρεται στη διάσταση των αντικειμένων και στη χωρική σχέση μεταξύ τους. Υπάρχουν δυο τύποι παραμόρφωσης προοπτικής:

- **Παραμόρφωση επέκτασης ή ευρείας γωνίας (extension distortion, wide angle distortion):** Μπορεί να φανεί σε εικόνες όπου έγινε χρήση ενός ευρυγώνιου φακού και το αντικείμενο είναι σε αρκετά κοντινή ή μακρινή απόσταση από τον φακό.

Τα αντικείμενα που βρίσκονται κοντά στον φακό φαίνονται ασυνήθιστα μεγάλα σε σχέση με τα αντικείμενα που βρίσκονται πιο μακριά και τα απομακρυσμένα, από τον φακό, αντικείμενα εμφανίζονται ασυνήθιστα μικρά και ως εκ τούτου δίνουν την αίσθηση ότι είναι πιο μακριά.

- **Παραμόρφωση συμπίεσης ή παραμόρφωση τηλεφακού (Compression, Telephoto distortion):** Αυτή η παραμόρφωση εμφανίζεται σε εικόνες που τραβήχτηκαν από απόσταση χρησιμοποιώντας έναν φακό μεγάλης εστίασης.

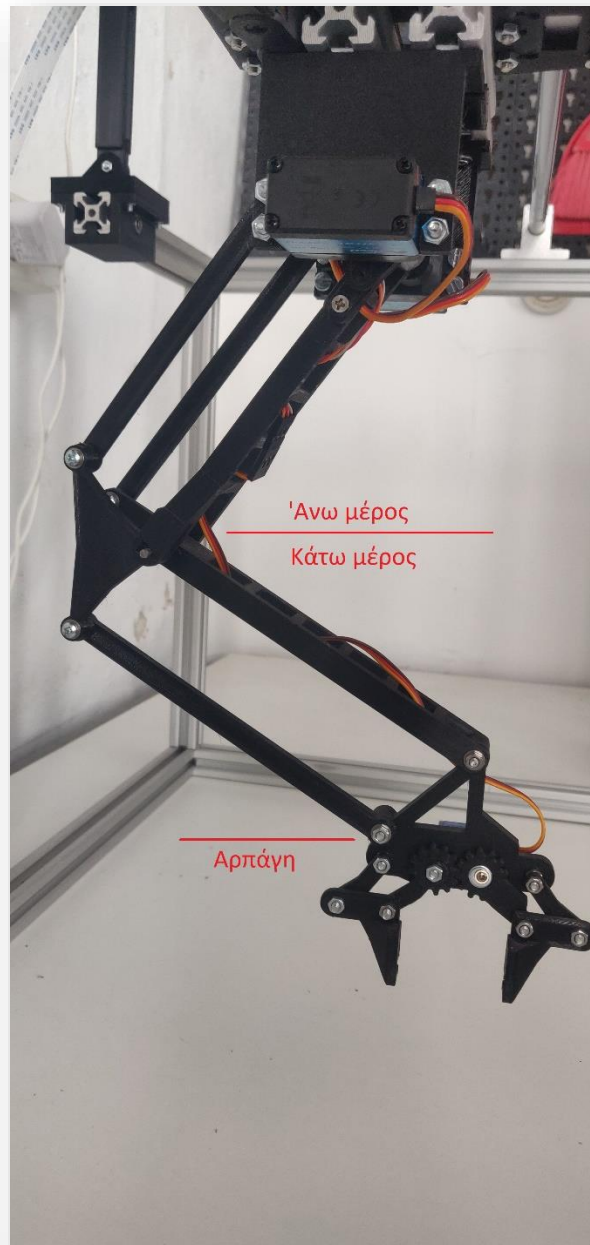
Τα απομακρυσμένα αντικείμενα μοιάζουν περίπου στο ίδιο μέγεθος, τα κοντινότερα αντικείμενα είναι ασυνήθιστα μικρά, και τα πιο απομακρυσμένα αντικείμενα είναι ασυνήθιστα μεγάλα, με αποτέλεσμα ο θεατής να μην μπορεί να διακρίνει τις σχετικές αποστάσεις μεταξύ των αντικειμένων.^[31]

3. Ανάπτυξη ρομποτικού συστήματος

Παρακάτω γίνεται μια αναφορά στα χαρακτηριστικά της κατασκευής καθώς και ορισμένες διαδικασίες οι οποίες χρειάστηκαν, ώστε να λειτουργήσει το ρομποτικό σύστημα.

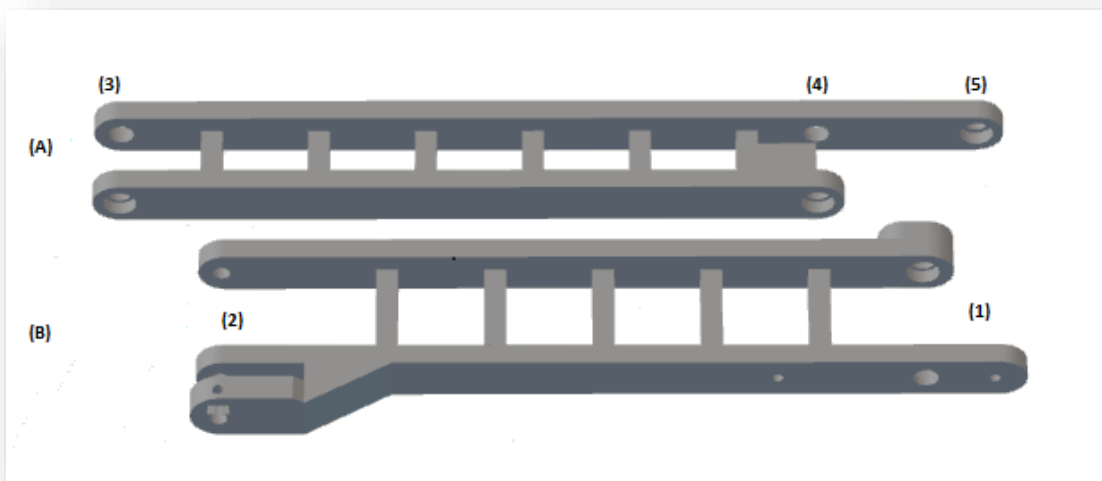
3.1 Μηχανολογικό μέρος

Ο ρομποτικός βραχίονας αποτελείται από δυο σέρβο κινητήρες FS5323M για την μετακίνηση του στον Y και Z άξονα. Ο βραχίονας χωρίζεται σε άνω και κάτω μέρος, όπως φαίνεται στην παρακάτω εικόνα.

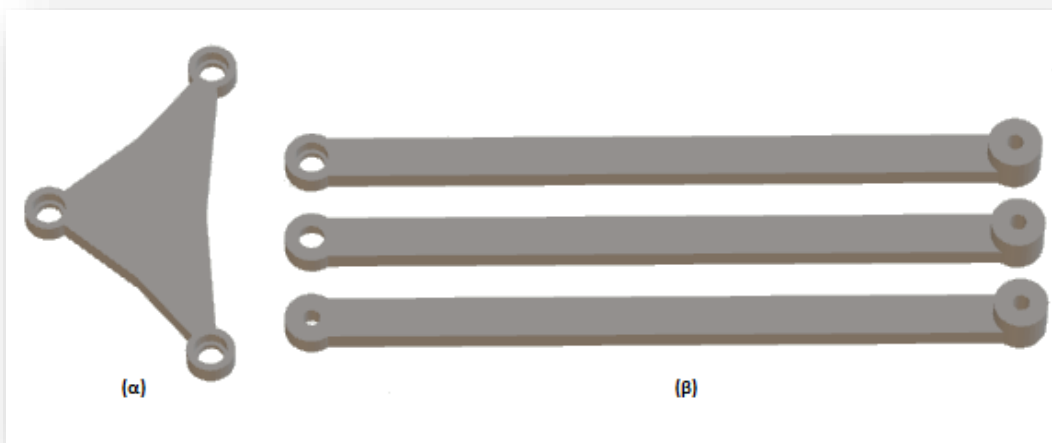


Εικόνα 3.1.1: Μέρη του ρομποτικού βραχίονα.

Στον καρπό του ρομποτικού βραχίονα είναι συνδεδεμένη μια αρπάγη ώστε να μπορεί να λαμβάνει τα αντικείμενα, με την βοήθεια ενός σέρβο κινητήρα SG90. Το κύριο στέλεχος του ρομποτικού βραχίονα αποτελείται από δυο ενισχυμένες βέργες μήκους 13cm (Εικόνα 3.1.2), τρεις απλές με μήκος 13cm και ένα τρίγωνο σύνδεσης (Εικόνα 3.1.3).



Εικόνα 3.1.2: Ενισχυμένες βέργες.



Εικόνα 3.1.3: Τρίγωνο σύνδεσης (α), απλές βέργες (β).

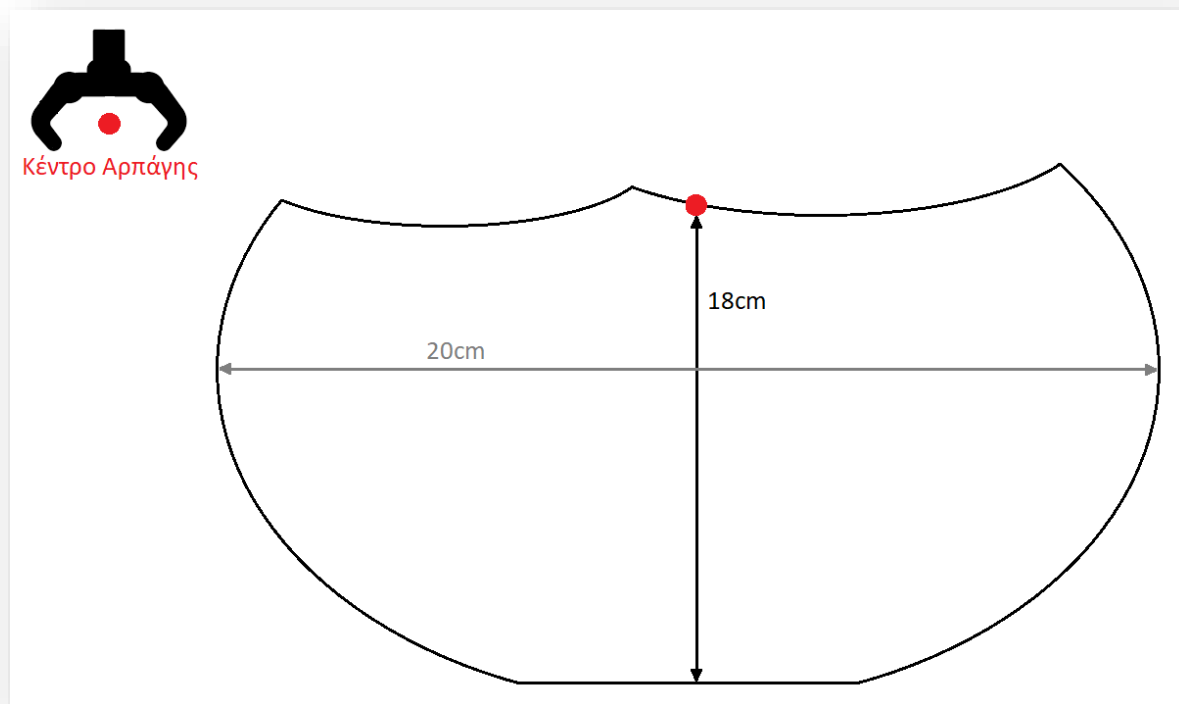
Η ενισχυμένη βέργα Α είναι σχεδιασμένη για τον έλεγχο του κάτω μέρους του ρομποτικού βραχίονα και η ενισχυμένη βέργα Β είναι σχεδιασμένη για τον έλεγχο του άνω μέρους. Στο ένα άκρο της ενισχυμένης βέργας Β (σημείο (1) της Εικόνας 3.1.2) συνδέονται οι δυο σέρβο κινητήρες. Ο σέρβο κινητήρας ο οποίος ελέγχει το άνω μέρος του ρομποτικού βραχίονα είναι συνδεδεμένος άμεσα με την ενισχυμένη ράβδο, ενώ ο δεύτερος σέρβο κινητήρας ο οποίος ελέγχει το κάτω μέρος του ρομποτικού βραχίονα είναι συνδεδεμένος έμμεσα με την ενισχυμένη ράβδο. Στο άλλο άκρο (σημείο (2) της Εικόνας 3.1.2) συνδέονται παράλληλα το τρίγωνο σύνδεσης μαζί με το άκρο της ενισχυμένης βέργας Α (σημείο (4) της Εικόνας 3.1.2).

Στην ενισχυμένη βέργα Α το σημείο (5) της Εικόνας 3.1.2 συνδέεται με μια απλή βέργα, η οποία καταλήγει στην άμεση σύνδεση με τον σέρβο κινητήρα, ο οποίος ελέγχει το κάτω μέρος του βραχίονα. Το σημείο (3) της Εικόνας 3.1.2 είναι το ένα μέρος του καρπού, δηλαδή σε αυτό το σημείο γίνεται η σύνδεση του τελικού στοιχείου σύνδεσης (αρπάγη). Το δεύτερο μέρος του καρπού αποτελείται από μια απλή βέργα η οποία είναι συνδεδεμένη στο τρίγωνο σύνδεσης.

Η τρίτη απλή βέργα συνδέεται στο τρίγωνο σύνδεσης και στην σταθερή βάση του ρομποτικού βραχίονα. Η σύνδεση με τη σταθερή βάση πραγματοποιείται με τέτοιο τρόπο, ώστε, η απλή αυτή βέργα να μπορεί να μετακινείται.

Η σταθερή βάση αποτελείται από μια πλάκα σχεδιασμένη με τέτοιο τρόπο, ώστε στο άνω μέρος της να συνδεθούν τα ρουλεμάν των οδηγών και στο κάτω μέρος δυο προφίλ αλουμινίου V-slot 2020 μήκους 10cm. Στα προφίλ αλουμινίου στηρίζονται ο ρομποτικός βραχίονας, μαζί με τους σέρβο κινητήρες.

Στην παρακάτω εικόνα φαίνεται το εύρος λειτουργίας του ρομποτικού βραχίονα.



Εικόνα 3.1.4: Εύρος λειτουργίας.

Για την μετακίνηση του ρομποτικού βραχίονα κατά τον άξονα X, χρησιμοποιείται ένας βηματικός κινητήρας τύπου NEMA 17, ο οποίος μετακινεί όλο το παραπάνω σύστημα.

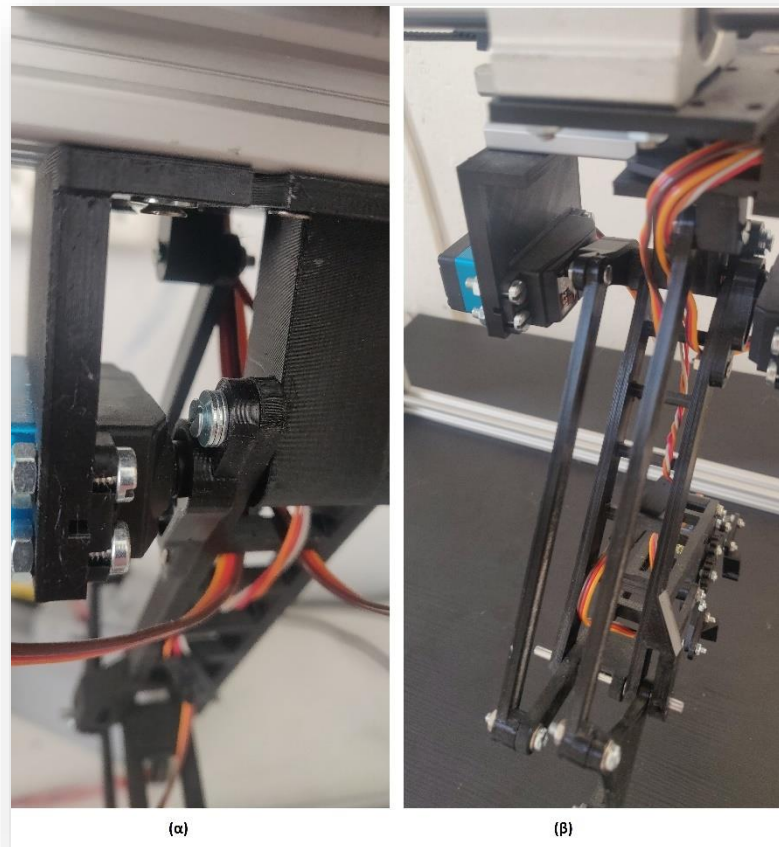
Αρχικά ο βηματικός κινητήρας είναι συνδεδεμένος στο ένα άκρο του πλαισίου της κατασκευής με μια τροχαλία χρονισμού (timing pulley) 20 δοντιών με οπή 5mm. Στο άλλο άκρο του πλαισίου υπάρχει μια απλή τροχαλία, η οποία είναι στην ίδια ευθεία με τη τροχαλία χρονισμού. Η σύνδεση μεταξύ των δυο υλικών γίνεται με έναν ιμάντα μετάδοσης κίνησης, ανοιχτού τύπου, μήκους περίπου 2m και τύπου GT2. Το διπλάσιο μήκος χρειάζεται διότι ο ιμάντας κλείνει στις δυο άκρες της πλάκας της σταθερής βάσης, έτσι ώστε να πραγματοποιείται η μετακίνηση του ρομποτικού βραχίονα κατά τον άξονα X.

Για την υποστήριξη της μετακίνησης κατά τον άξονα X τοποθετούνται δυο γραμμικοί οδηγοί μήκους 1m, οι οποίοι στηρίζονται στα άκρα της κατασκευής. Ενδιάμεσα τους υπάρχει το σύστημα μετάδοσης του βηματικού κινητήρα, όπως αναφέρθηκε παραπάνω. Στα ρουλεμάν που υπάρχουν στη σταθερή βάση του ρομποτικού βραχίονα εισέρχονται οι γραμμικοί οδηγοί, έτσι ώστε να υποστηρίξουν το σύστημα, και να διευκολύνουν την μετακίνηση του.

Η κάμερα που χρησιμοποιείται είναι η Raspberry Pi Camera module v2 και είναι σταθεροποιημένη στο πλαίσιο της κατασκευής, κάθετα από τον άξονα X.

Τέλος είναι σημαντικό να αναφερθεί ότι οι τρεις σέρβο κινητήρες πρέπει να «κεντραριστούν» στις 90 μοίρες, πριν την εγκατάστασή τους, καθώς επίσης ο βραχίονας, η βάση στήριξης της κάμερας και πολλά άλλα πλαστικά μέρη έχουν κατασκευαστεί με τη βοήθεια ενός 3D Printer.

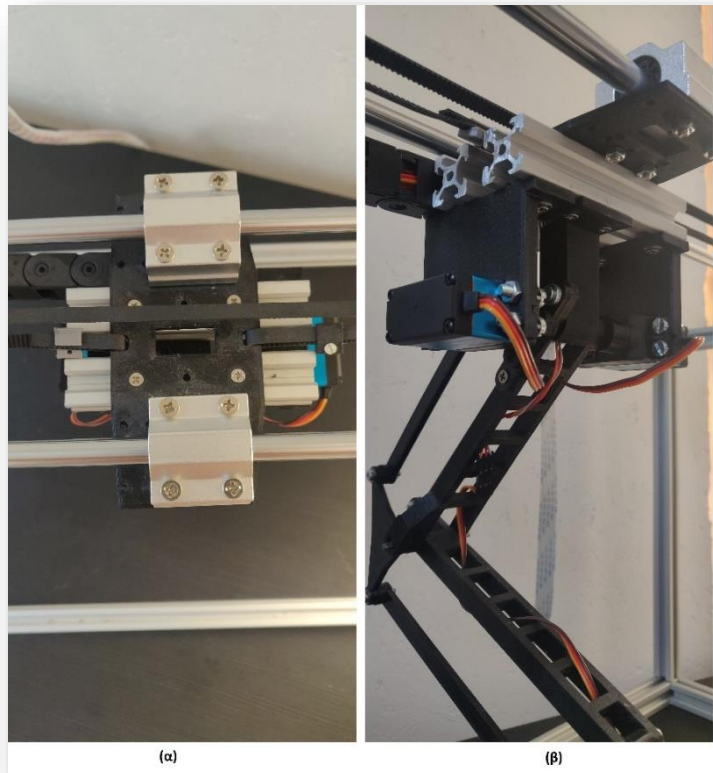
Στις παρακάτω εικόνες φαίνεται το τελικό αποτέλεσμα.



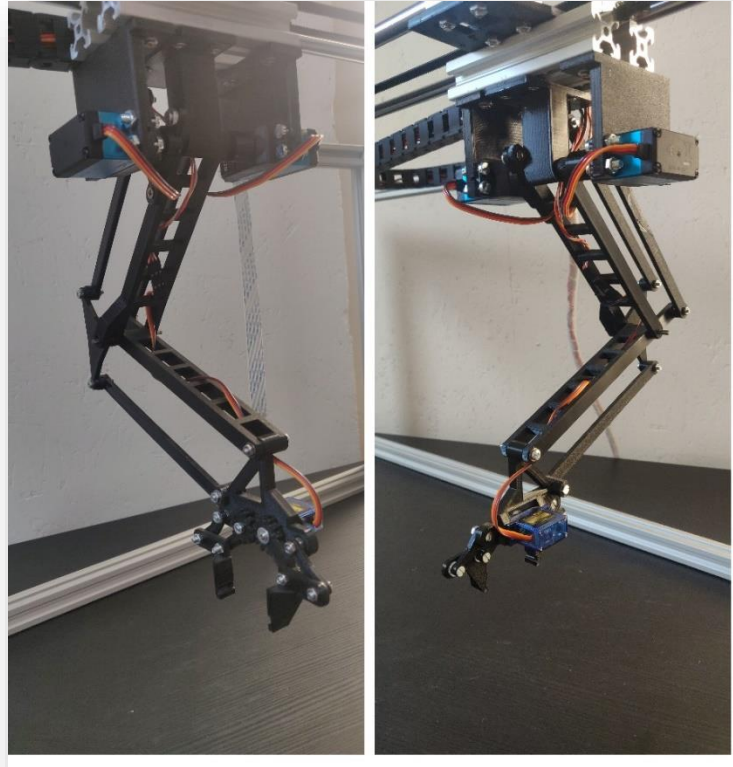
Εικόνα 3.1.5: Άμεση σύνδεση του σέρβο με την ενισχυμένη βέργα Β (α) και αντίστοιχα η έμμεση σύνδεση του δεύτερου σέρβο (β).



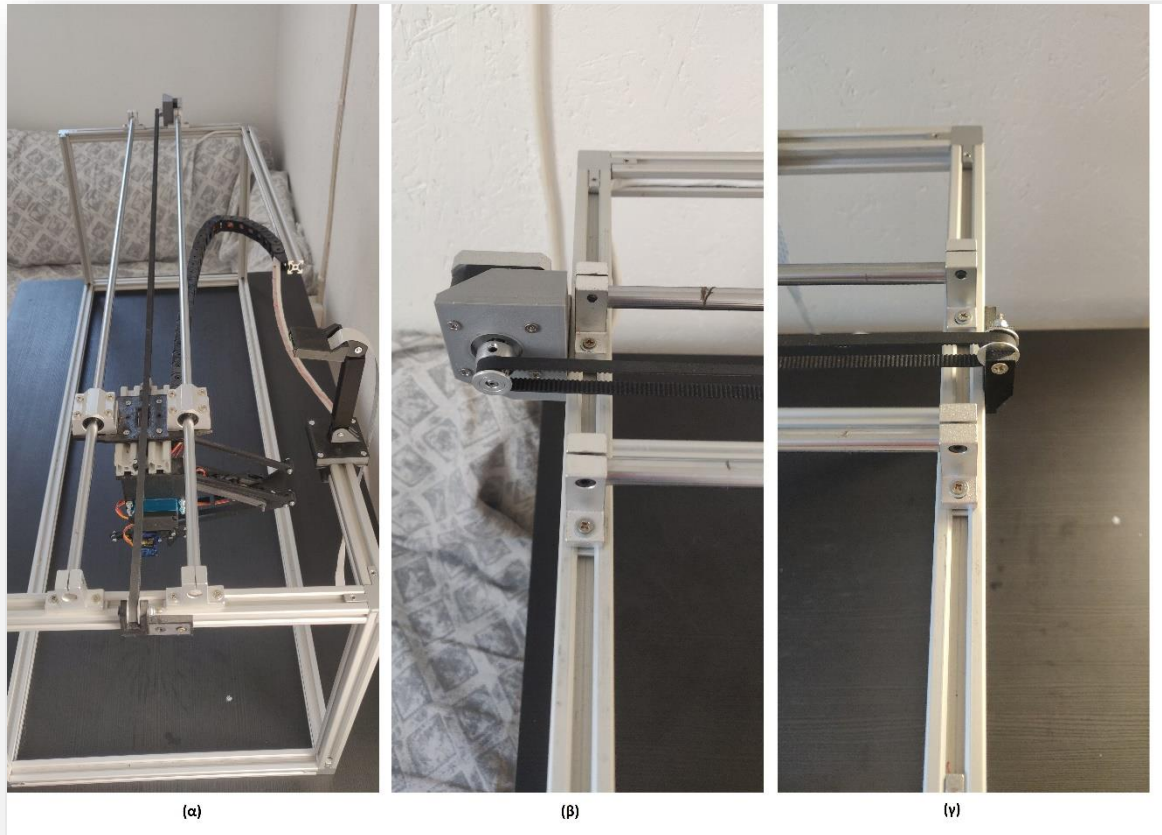
Εικόνα 3.1.6: Τρίγωνο σύνδεσης.



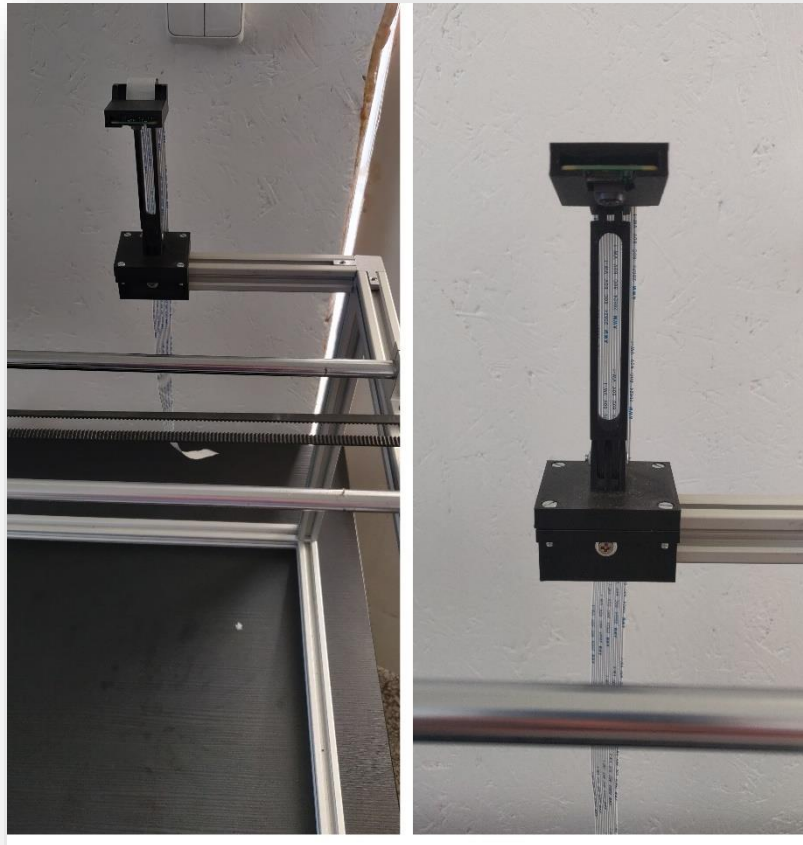
Εικόνα 3.1.7: Άνω όψη σταθερής βάσης (α) και κάτω όψη σταθερής βάσης.



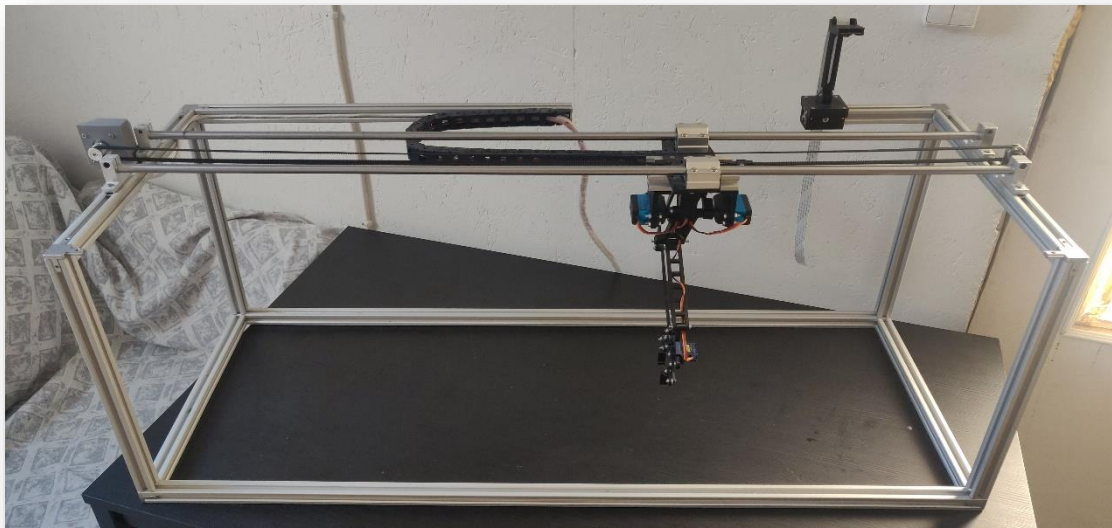
Εικόνα 3.1.8: Γενικό πλάνο ρομποτικού βραχίονα.



Εικόνα 3.1.9: Γενική όψη του άξονα X (α), σύνδεση του βηματικού, του οδοντωτού τροχαλία και των γραμμικών οδηγών (β) και σύνδεση του απλού τροχαλία και των άκρων των οδηγών (γ).



Εικόνα 3.1.10: Εγκατάσταση κάμερας.



Εικόνα 3.1.11: Γενικό πλάνο κατασκευής στο μηχανολογικό μέρος.

3.2 Μετατροπή συντεταγμένων Y, Z σε γωνίες

Στο ρομποτικό βραχίονα παρέχεται ένα σύνολο συντεταγμένων, (Y,Z) και με ορισμένα βήματα (εξισώσεις) το ίδιο το πρόγραμμα καταλάβει τη θέση που απαιτείται από τους σέρβο κινητήρες. Στη

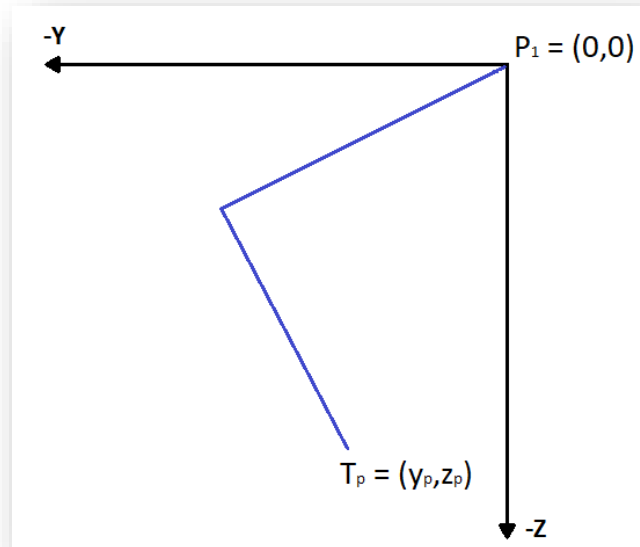
ρομποτική είναι γνωστό ως Inverse Kinematics και ο ρομποτικός βραχίονας της κατασκευής χρησιμοποιεί την πιο απλή μορφή του.

Στην παρακάτω εικόνα φαίνεται η κατεύθυνση των συντεταγμένων Y, Z του ρομποτικού βραχίονα και τα δυο σημεία ενδιαφέροντος (P_1 και T_p). Η αρπάγη θεωρείται ένα σταθερό μέγεθος και δεν λαμβάνεται υπ' όψη στους υπολογισμούς.



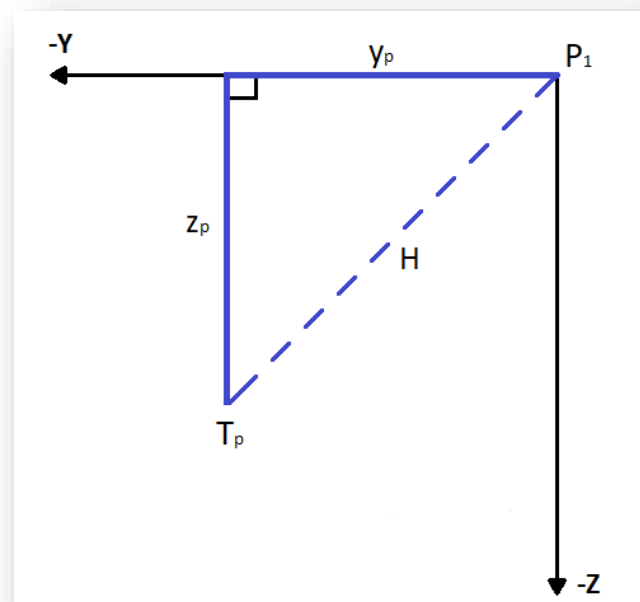
Εικόνα 3.2.1: Περιγραφή ρομποτικού βραχίονα.

Το P_1 είναι στη βάση του ρομποτικού βραχίονα και βρίσκεται στον άξονα του άνω βραχίονα. Το T_p είναι το σημείο στόχου (το σημείο που πρέπει να μετακινηθεί ο βραχίονας) με ζητούμενο τα σημεία y_p και z_p ώστε, ο ρομποτικός βραχίονας να υπολογίζει αυτόματα τις γωνίες για τους σέρβο κινητήρες.



Εικόνα 3.2.2: Ενδιαφερόμενα στοιχεία ρομποτικού βραχίονα.

Το H είναι η υποτείνουσα ενός ορθογωνίου τριγώνου με στοιχεία y_p και z_p και υπολογίζεται με την εξίσωση $H = \sqrt{y_p^2 + z_p^2}$



Εικόνα 3.2.3: Υπολογισμός H .

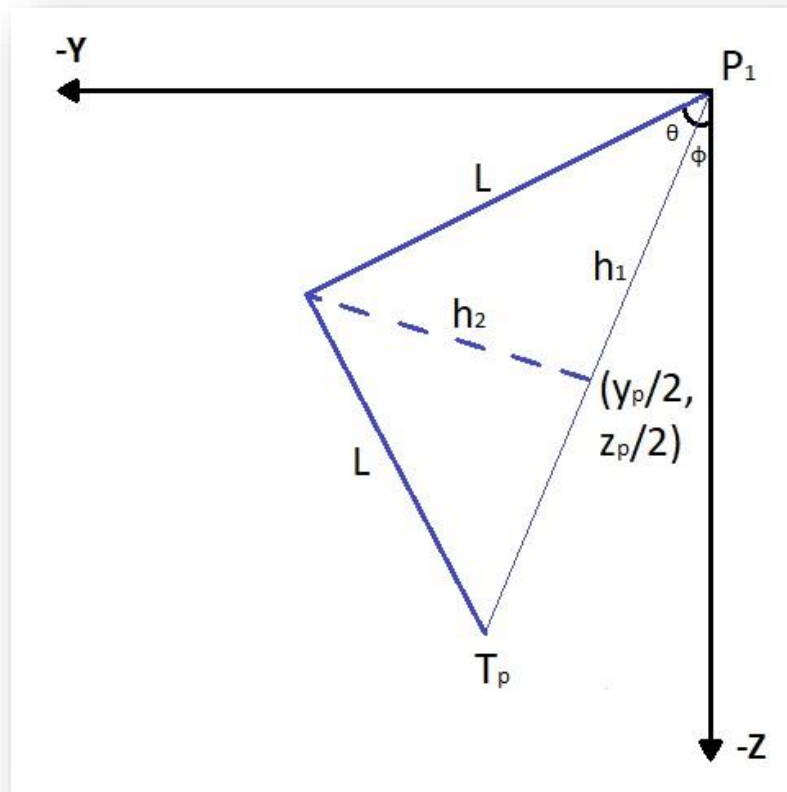
Στη συνέχεια υπολογίζονται τα h_1 και h_2 , με δεδομένα τα μήκη του ρομποτικού βραχίονα (μήκος των ενισχυμένων βεργών) καθώς και τις γωνίες θ και φ , με τις παρακάτω εξισώσεις.

$$h_1 = \frac{H}{2}$$

$$h_2 = \sqrt{L^2 - h_1^2}$$

$$\theta = \tan^{-1} \frac{h_2}{h_1}$$

$$\varphi = \tan^{-1} \frac{y_p/2}{z_p/2}$$

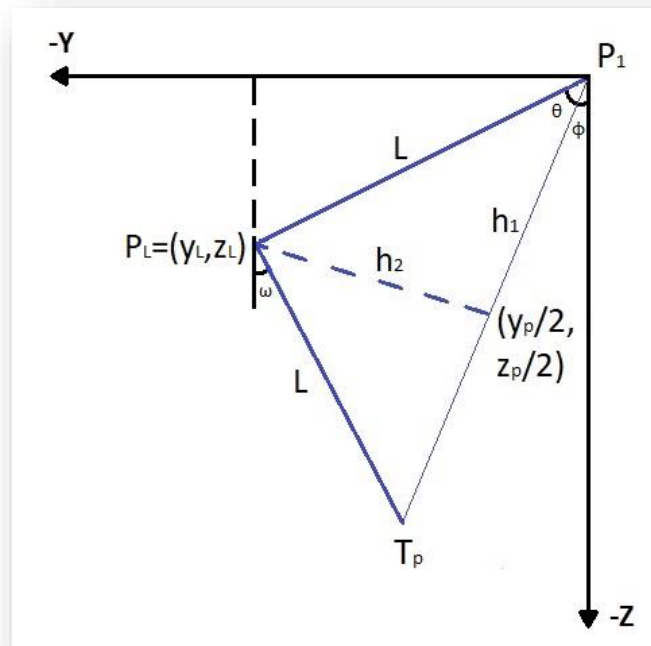


Εικόνα 3.2.4: Υπολογισμός h_1 , h_2 , θ και φ .

Με τα παραπάνω δεδομένα υπολογίζεται η γωνία θέσης του πρώτου σέρβο κινητήρα με την εξίσωση: $\sigma\epsilon\rho\beta o1 = \theta + \varphi$. Γνωρίζοντας τη γωνία του πρώτου σέρβο κινητήρα, πραγματοποιείται ο υπολογισμός της θέσης $P_3 (y_L, z_L)$ και τέλος υπολογίζεται η γωνία του δεύτερου σέρβο κινητήρα.

$$\begin{bmatrix} y_L \\ z_L \end{bmatrix} = \begin{bmatrix} \cos(\theta + \varphi) & \sin(\theta + \varphi) \\ -\sin(\theta + \varphi) & \cos(\theta + \varphi) \end{bmatrix} \begin{bmatrix} 0 \\ -L \end{bmatrix}$$

$$\sigma\epsilon\rho\beta o2 = \omega = \tan^{-1} \left(\frac{z_p - z_L}{y_p - y_L} \right)$$



Εικόνα 3.2.5: Υπολογισμός δεύτερου σέρβο κινητήρα.

Αυτή είναι μια πολύ απλή εξήγηση για τον τρόπο υπολογισμού θέσης των σέρβο κινητήρων και λειτουργούν στην περιοχή από 0 έως 90 μοίρες. Για θέσεις που απαιτούν επίλυση πέρα από αυτό το εύρος γωνιών, πρέπει να γίνουν μερικές προσαρμογές για να ληφθούν υπόψη οι αλλαγές στην τριγωνομετρία. Παρακάτω φαίνεται ο κώδικας για με τις εξισώσεις.^[32]

```

void get_angles_from_yz(double y, double z) {

    double H, s1, s2, aB, aA, aQ, servolangle, servo2angle, y2, z2, y3, z3;

    //arm length in cm
    int L = 13;

    H= sqrt (pow(y,2) + pow(z,2));
    s1=H/2;
    s2=sqrt (pow(L,2) - pow(s1,2));

    aB=atan(s2/s1);
    y2=y/2;
    z2=z/2;
    aA=atan(y2/z2);

    servolangle=aA+aB;
    servolangle= (servolangle/ (2 * M_PI)) * 360;

    //matrix multiplication - counterclockwise rotation
    y3 = -L*sin(aA+aB);
    z3 = -L* cos(aA+aB);

    servo2angle=atan((y-y3)/(z-z3));

    servo2angle= (servo2angle / (2 * M_PI)) * 360;
    //tangent calculation changes when servo2 exceeds 90 degrees, correction below
    if ((z-z3)>0) {
        servo2angle=servo2angle-180;
    }

    //Absolute Top Arm Angle
    //Top Arm moves 0 to +90
    angle_next[0] = servolangle;

    //Absolute Middle Arm Angle
    //Middle Arm moves 0 to +90
    angle_next[1] = -servo2angle;

    //Convert to SERVO Angle
    //in this case, a 90 servo position is equal to 71 degrees for Top arm
    //90 servo position is equal to 65 Middle Arm
    angle_next[0] = angle_next[0] + calibrate_TopArm;
    angle_next[1] = angle_next[1] + calibrate_MiddleArm;
}

```

Εικόνα 3.2.6: Κώδικας με την τριγωνομετρία.

3.3 Ηλεκτρονικό μέρος

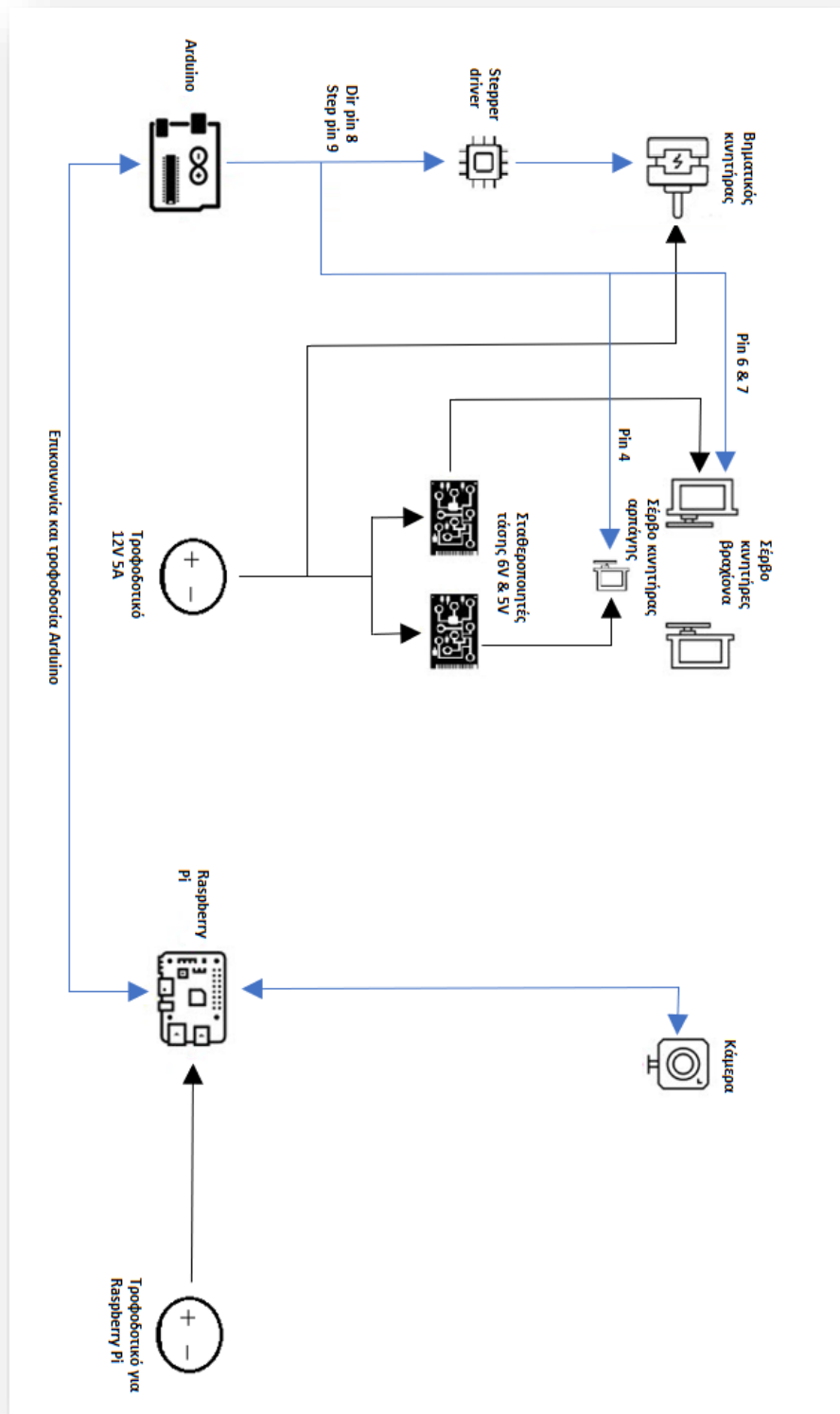
Οι σέρβο κινητήρες, καθώς και ο βηματικός κινητήρας ελέγχονται από ένα Arduino. Το Arduino μετατρέπει τις συντεταγμένες σε μοίρες, για τους σέρβο κινητήρες ή σε βήματα, για τον βηματικό κινητήρα. Για την τροφοδοσία των σέρβο κινητήρων FS5323M χρησιμοποιείται ένας σταθεροποιητής τάσης LM2596 ρυθμιζόμενος στα 6V και για τον σέρβο κινητήρα SG90 χρησιμοποιείται αντίστοιχος σταθεροποιητής τάσης ρυθμιζόμενος στα 5V. Για τον βηματικό κινητήρα χρησιμοποιείται ένας stepper driver A4988, για την οδήγηση του, ο οποίος τροφοδοτείται με 12V τάση. Για την τροφοδοσία όλων των κινητήρων χρησιμοποιείται ένα τροφοδοτικό 12V 5A και με αυτόν τον τρόπο οι σταθεροποιητές τάσης εφαρμόζουν την επιθυμητή τάση όπου χρειάζεται.

Το Arduino τροφοδοτείται και επικοινωνεί με το Raspberry Pi μέσω ενός USB.

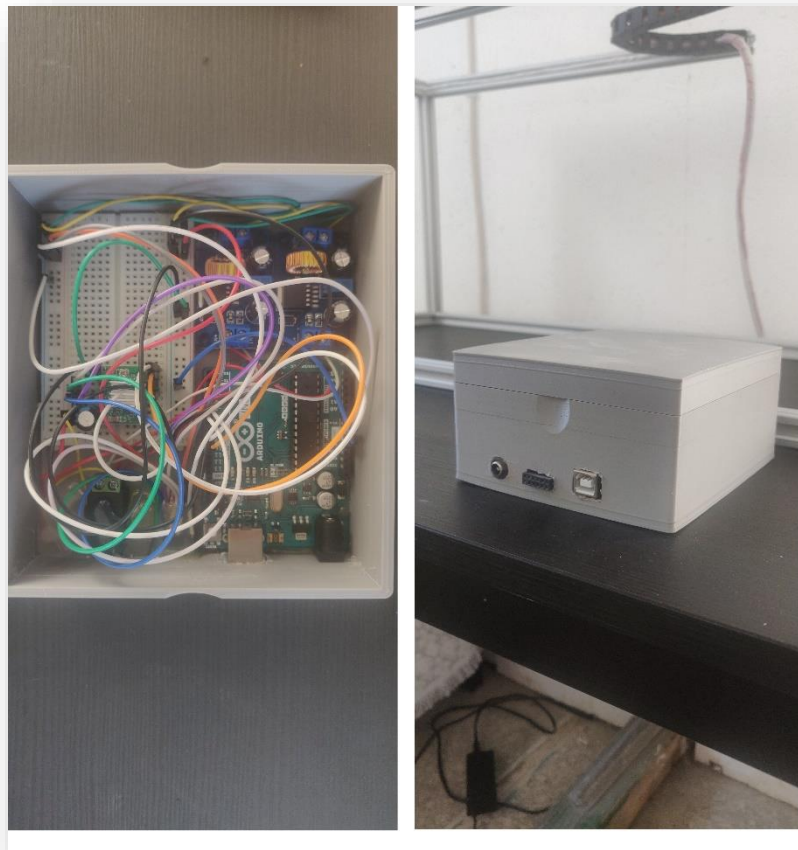
Στο Raspberry Pi συνδέεται επίσης η κάμερα μέσω μιας καλωδιωτικής που προσφέρει ο κατασκευαστής και το Raspberry Pi τροφοδοτείται ξεχωριστά από το υπόλοιπο ρομποτικό σύστημα. Οι προδιαγραφές του τροφοδοτικού είναι προτεινόμενο να ακολουθούν αυτές του κατασκευαστή ($12V \geq 2A$).

Όλα τα εξαρτήματα, εκτός του Raspberry Pi έχουν τοποθετηθεί σε ένα κουτί κατασκευασμένο μέσω ενός 3D printer, με διαμορφωμένες εξόδους για την πραγματοποίηση των συνδέσεων των κινητήρων, της τροφοδοσίας τους, καθώς και την επικοινωνία μεταξύ του Arduino και του Raspberry Pi.

Παρακάτω φαίνεται το ηλεκτρονικό διάγραμμα των εξαρτημάτων (μαζί με τα pin για την σύνδεση των κινητήρων με το Arduino), καθώς και ορισμένες εικόνες της συνδεσμολογίας και της διαχείρισης των καλωδίων.



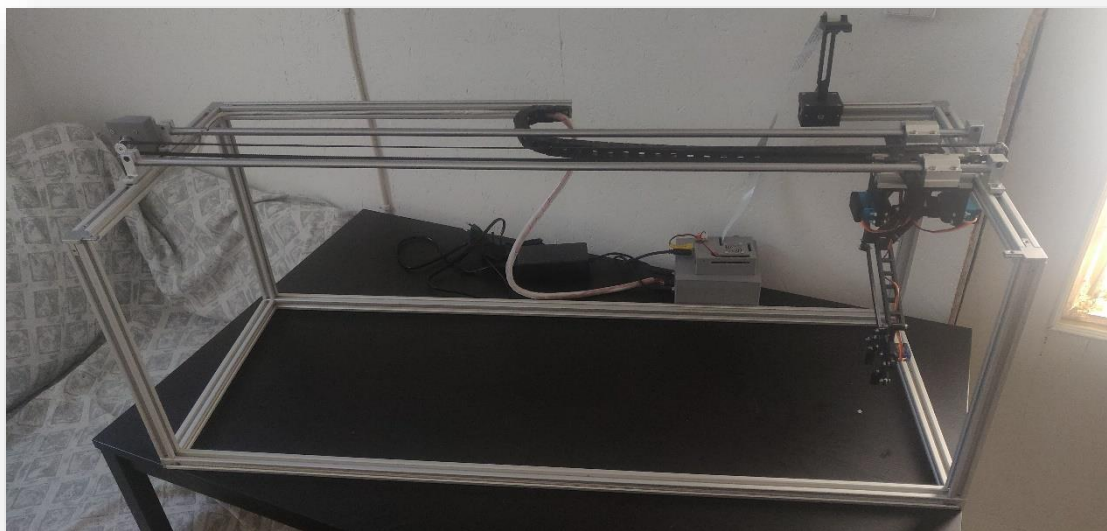
Εικόνα 3.3.1: Ηλεκτρονικό διάγραμμα.



Εικόνα 3.3.2: Κουτί με τα εξαρτήματα.



Εικόνα 3.3.3: Σύνδεση κάμερας και τελική σύνδεση όλων των εξαρτημάτων.



Εικόνα 3.3.4: Τελικό αποτέλεσμα κατασκευής.

3.4 Προγραμματισμός

Για τον προγραμματισμό της κατασκευής θα γίνει αναφορά σε ορισμένα σημεία, ώστε να κατανοηθεί καλύτερα ο τρόπος λειτουργίας του συστήματος.

Αρχικά η ανίχνευση των αντικειμένων και η διόρθωση της κάμερας είναι μια υπολογιστική (ή μηχανική) όραση (Computer Vision). Για τον λόγο αυτό χρειάζεται η εγκατάσταση της OpenCV στη Python. Η OpenCV (Open Source Computer Vision Library) είναι μια υπολογιστική όραση ανοικτού κώδικα και βιβλιοθήκη λογισμικού μηχανικής μάθησης.^[33]

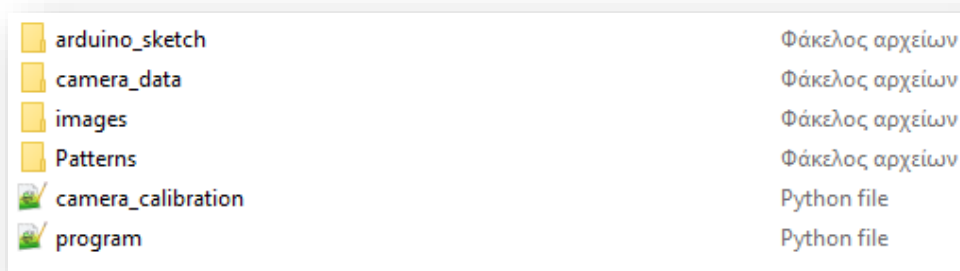
Η OpenCV χρησιμοποιεί επιστημονικούς υπολογισμούς (scientific computing), γι' αυτό τον λόγο απαιτείται η εγκατάσταση μιας βιβλιοθήκης στη Python που ονομάζεται NumPy. Πρόκειται για μια βιβλιοθήκη (Python) που παρέχει ένα πολυδιάστατο αντικείμενο συστοιχίας, διάφορα παράγωγα αντικείμενα (όπως συγκαλυμμένες συστοιχίες και πίνακες) και μια σειρά από ρουτίνες για γρήγορες λειτουργίες σε συστοιχίες, συμπεριλαμβανομένων μαθηματικών, χειρισμού σχήματος, ταξινόμησης, επιλογής, διακριτοί μετασχηματισμοί Fourier, βασική γραμμική άλγεβρα, βασικές στατιστικές λειτουργίες και πολλά άλλα.^[34]

Υπάρχουν δυο διαφορετικά αρχεία κώδικα python.

1. **camera_calibration:** Αυτό το αρχείο χρησιμοποιείται για την διόρθωση της κάμερας ως προς την γεωμετρική παραμόρφωση και τη παραμόρφωση προοπτικής. Επίσης ελέγχει την ακρίβεια της διόρθωσης και βοηθάει στην αρχικοποίηση των αξόνων X και Y (πραγματικές συντεταγμένες) πάνω στο οπτικό πεδίο της κάμερας.
2. **program:** Το συγκεκριμένο αρχείο είναι υπεύθυνο για την λειτουργία του ρομποτικού συστήματος σταθερής βάσης. Συγκεκριμένα πραγματοποιεί την οπτική ανίχνευση των αντικειμένων, την αναγνώριση των χρωμάτων τους και στην συνέχεια τοποθετεί τα αντικείμενα σε ορισμένα επιθυμητά σημεία (ανάλογα το χρώμα τους).

Πέρα από τα αρχεία, υπάρχουν και τέσσερις φάκελοι:

1. Στο φάκελο `arduino_sketch` υπάρχει ο κώδικας για τον προγραμματισμό του Arduino.
2. Ο φάκελος `calibration_images` περιέχει ορισμένες φωτογραφίες που λήφθηκαν μέσω της κάμερας, ώστε να πραγματοποιηθεί η διόρθωση της κάμερας.
3. Στο φάκελο `camera_data` υπάρχουν όλα τα αρχεία που δημιουργούνται κατά την διαδικασία της διόρθωσης της κάμερας, τα οποία χρειάζονται για την ορθή λειτουργία του ρομποτικού βραχίονα.
4. Ο φάκελος `Patterns` περιέχει ορισμένα «εργαλεία» για την διεξαγωγή της διόρθωσης της κάμερας.



Εικόνα 3.4.1: Αρχεία προγράμματος.

Το Arduino λειτουργεί ως οδηγός των κινητήρων και η επικοινωνία του με το Raspberry Pi γίνεται μέσω μιας σειριακής επικοινωνίας που δημιουργείται από τον χρήστη, με ένα επιθυμητό Baud rate.

```
void setup() {  
    // Setup Communication  
    Serial.begin(9600);  
    import time  
    import serial  
    ser = serial.Serial('/dev/ttyACM0', 9600)
```

Εικόνα 3.4.2: Έναρξη σειριακής επικοινωνίας για το Arduino (αριστερά) και το Raspberry Pi (δεξιά).

Η οπτική αναγνώριση των αντικειμένων πραγματοποιείται με την τεχνική Background Subtraction. Τα καρέ που πρόκειται να συγκριθούν αρχικά μετατρέπονται σε κλίμακα του γκρι και στη συνέχεια φιλτράρεται ώστε να μειωθεί ο θόρυβος Gauss (Gaussian Noise). Έπειτα υπολογίζεται η διαφορά που υπάρχει μεταξύ τους και βρίσκονται το περίγραμμα οριοθέτησης του κάθε αντικειμένου.

```

image = frame.array
gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
gray = cv.morphologyEx(gray, cv.MORPH_CLOSE, kernel)
gray = cv.GaussianBlur(gray, (5, 5), 0)

absolute_difference = cv.absdiff(first_frame, gray)
_, absolute_difference = cv.threshold(absolute_difference, 80, 255, cv.THRESH_BINARY)
contours, hierarchy = cv.findContours(absolute_difference, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)[-2:]
    
```

Εικόνα 3.4.3: Κώδικας ανίχνευσης αντικειμένων.

Έπειτα, γίνεται ένας έλεγχος εάν το περίγραμμα είναι εντός των ορίων που έχουν καθοριστεί από τον χρήστη, και στην περίπτωση που το περίγραμμα είναι εκτός των ορίων, το πρόγραμμα παραβλέπει το αντικείμενο αυτό. Τα όρια βοηθούν στην μείωση της λανθασμένης οπτικής ανίχνευσης (θόρυβος) ενός αντικειμένου.

Επίσης με τα όρια ο χρήστης μπορεί να καθορίσει το μέγεθος των αντικειμένων που τον ενδιαφέρει να ανιχνεύσει το πρόγραμμα. Για παράδειγμα εάν τα αντικείμενα δεν ξεπερνούν τις διαστάσεις 5x5 (cm), τότε δεν χρειάζεται το πρόγραμμα να ανιχνεύει μεγαλύτερα αντικείμενα.

```

area = cv.contourArea(cnt)
if area > 600 and area < 10000:
    
```

Εικόνα 3.4.4: Οριοθέτηση του περιγράμματος.

Στην συνέχεια, βρίσκονται τα κέντρα του περιγράμματος των αντικειμένων και πραγματοποιείται έλεγχος εάν τα κέντρα είναι εντός ορισμένων ορίων (σε pixel) στο οπτικό πεδίο της κάμερας. Η οπτική αναγνώριση δεν γίνεται σε όλο το οπτικό πεδίο της κάμερας αλλά μόνο στο μέρος που πρόκειται να τοποθετούνται τα αντικείμενα.

Εάν τα αντικείμενα είναι εντός των οπτικών ορίων που έχουν καθοριστεί, τότε δημιουργούνται το περίγραμμα και το κέντρο κάθε αντικειμένου και στη συνέχεια πραγματοποιείται η αναγνώριση του χρώματος για κάθε αντικείμενο.

Τέλος οι πληροφορίες αποθηκεύονται σε ένα πίνακα. Ο συγκεκριμένος πίνακας περιέχει τις τιμές του κέντρου (X, Y του μετρικού συστήματος) και το χρώμα του αντικειμένου. Για κάθε αναγνωρισμένο αντικείμενο υπάρχει ένας αντίστοιχος πίνακας, ο οποίος δεν εμφανίζεται στον χρήστη αλλά είναι σημαντικός διότι αποτελείται από πληροφορίες οι οποίες χρειάζονται για το πρόγραμμα.

```

x,y,w,h = cv.boundingRect(cnt)
x2 = x + int(w/2)
y2 = y + int(h/2)

if (x2>530 and x2<1100) and (y2>220 and y2<570):

    b,g,r =(image[(y2,x2)])
    clr= [b,g,r]
    # Detect the color of each object
    # BLACK
    if (clr[0]>-1 and clr[0]<40) and (clr[1]>-1 and clr[1]<40) and (clr[2]>-1 and clr[2]<94):
        clr=1
    # RED
    elif (clr[0]>1 and clr[0]<70) and (clr[1]>1 and clr[1]<70) and (clr[2]>90 and clr[2]<255):
        clr=2
    # GREY
    elif (clr[0]>40 and clr[0]<150) and (clr[1]>40 and clr[1]<150) and (clr[2]>40 and clr[2]<150):
        clr=3
    # UNKNOWN
    else:
        clr=4

    nuv=np.array([[x2,y2,1]], dtype=np.float32)
    nuv=nuv.T
    nsuv=sarr * nuv
    xyzc=inverse_newmtx.dot(nsuv)
    xyzc=xyzc-newtvec
    XYZ=inverse_rodvec.dot(xyzc)
    Xx = np.round(XYZ[0],2)
    X = np.float (Xx)
    Yy = np.round(XYZ[1],2)
    Y = np.float (Yy)
    obj = [X,Y,clr]
    objj.append(obj)
    
```

Εικόνα 3.4.5: Τελική διαδικασία αναγνώρισης αντικειμένων.

Στην υπολογιστική όραση η αναγνώριση ή η απεικόνιση των χρωμάτων είναι ένας συνδυασμός κόκκινου, πράσινου και μπλε (γνωστό ως RGB) χρώματος με εύρος 0 έως 255 για το δυαδικό σύστημα και 00 έως FF για το δεκαεξαδικό σύστημα. Η σειρά των χρωμάτων είναι σημαντική για την αναγνώριση ή την απεικόνιση ενός χρώματος. Για παράδειγμα η OpenCV έχει την ακολουθία BGR (Blue Green Red) και στη περίπτωση ορισμένων εφαρμογών που χρησιμοποιούν ακολουθία RGB πρέπει να γίνει διαμόρφωση της ακολουθίας BGR σε RGB, μέσω μιας εντολής που υπάρχει στην OpenCV (cvtColor()) για να αποφευχθεί η σύγχυση στην απεικόνιση ή την αναγνώριση και η εμφάνιση του σφάλματος στον κώδικα.

Παρακάτω υπάρχουν ορισμένα παραδείγματα σε δυαδικό σύστημα:

- **Κόκκινο:** Σε RGB είναι [255,0,0] ενώ σε BGR είναι [0,0,255]
- **Πράσινο:** Σε RGB είναι [0,255,0] ενώ σε BGR είναι [0,255,0]
- **Μπλε:** Σε RGB είναι [0,0,255] ενώ σε BGR είναι [255,0,0]

Για την πραγματοποίηση της αναγνώρισης των χρωμάτων στο πρόγραμμα ακολουθείται η εξής σειρά.

Αρχικά πραγματοποιείται η ανίχνευση των αντικειμένων. Στη συνέχεια τοποθετούνται τα πλαίσια οριοθέτησης για κάθε αναγνωρισμένο αντικείμενο και υπολογίζεται το κέντρο της οριοθέτησης τους. Έπειτα γίνεται η αναγνώριση του χρώματος κάθε αντικειμένου στο κέντρο που υπολογίστηκε

προηγουμένως και τέλος ανάλογα τα προκαθορισμένα από τον χρήστη όρια, το πρόγραμμα, αποθηκεύει μια μεταβλητή σε έναν πίνακα.

```
b,g,r =(image[(y2,x2)])
clrs = [b,g,r]
# Detect the color of each object
# BLACK
if (clrs[0]>-1 and clrs[0]<40) and (clrs[1]>-1 and clrs[1]<40) and (clrs[2]>-1 and clrs[2]<94):
    clr=1
# RED
elif (clrs[0]>1 and clrs[0]<70) and (clrs[1]>1 and clrs[1]<70) and (clrs[2]>90 and clrs[2]<255):
    clr=2
# GREY
elif (clrs[0]>40 and clrs[0]<150) and (clrs[1]>40 and clrs[1]<150) and (clrs[2]>40 and clrs[2]<150):
    clr=3
# UNKNOWN
else:
    clr=4
```

Εικόνα 3.4.6: Καθορισμός ορίων των χρωμάτων σε BGR.

Τα προκαθορισμένα όρια είναι τα όρια των αποχρώσεων των επιθυμητών χρωμάτων. Τα όρια τοποθετούνται μεταξύ δυο τιμών διότι κάθε χρώμα έχει ορισμένες αποχρώσεις. Για παράδειγμα στη περίπτωση του κόκκινου χρώματος, εάν το όριο ήταν μόνο [0,0,255] (σε BGR) τότε το χρώμα θα έπρεπε να είναι αρκετά έντονο και απόλυτα ίδιο με αυτό που «γνωρίζει» ως κόκκινο το πρόγραμμα.

Ακόμη υπάρχουν εξωτερικοί παράγοντες που επηρεάζουν σε μεγάλο βαθμό την αναγνώριση των χρωμάτων. Ο φωτισμός του χώρου, τα χαρακτηριστικά της κάμερας (τύπος αισθητήρα, φακός) και η ποιότητα του χρώματος του αντικειμένου είναι οι πιο σημαντικοί εξωτερικοί παράγοντες.

Για να βρεθούν τα σωστά όρια πραγματοποιείται μια αναγνώριση των επιθυμητών χρωμάτων, σε πραγματικό χρόνο (με την κάμερα του συστήματος), ώστε να ληφθούν οι τιμές με βάση τους εξωτερικούς παράγοντες. Με αυτό το τρόπο μειώνεται δραματικά το σφάλμα λανθασμένης αναγνώρισης.

Η κάμερα μπορεί να δώσει μόνο δυο διαστάσεις για τον πραγματικό κόσμο (συντεταγμένες X και Y). Για να γνωρίζει το σύστημα την τρίτη διάσταση, υπάρχουν ορισμένοι περιορισμοί – καθορισμοί για τον άξονα Z. Βοηθούν στους υπολογισμούς του Arduino ώστε ο βραχίονας να πάρει την σωστή θέση πάνω από το αντικείμενο.

```
LOW_Z = -22.8 # Το πιο χαμηλό σημείο που μπορεί να φτάσει η αρπάγη
HOVER_Z = -16.5 # Ελάχιστο ύψος για να περνάει πάνω απο τα αντικείμενα
DROP_Z = -21.0 # Ύψος για την τοποθέτηση του αντικειμένου
MID_Z = -12.0 # Ενδιάμεσο ύψος
HIGH_Z = -6.0 # Μέγιστο ύψος
DELAY = 0
```

Εικόνα 3.4.7: Καθορισμός ύψους.

Η αρχή των αξόνων, του πραγματικού κόσμου (X,Y), που υπολογίζει το πρόγραμμα (μετά τη διόρθωση της κάμερας), διαφέρει από αυτή που βρίσκεται ο βραχίονας λόγω κατασκευής. Για την

διόρθωση αυτού του σφάλματος υπάρχουν δυο μεταβλητές (μια για κάθε άξονα), στις οποίες δηλώνεται η διαφορά της αρχής των αξόνων της κάμερας, με αυτή του ρομποτικού βραχίονα.

```
calibX = 1.12  
calibY = 11.85
```

Εικόνα 3.4.8: Μεταβλητές διόρθωσης.

Για κάθε αναγνωρισμένο χρώμα, ο χρήστης, έχει την δυνατότητα του ορισμού των σημείων (σε συντεταγμένες X, Y), όπου το αντικείμενο πρόκειται να τοποθετηθεί.

```
if objj[i][2] == 1:  
    placeX = 35  
    placeY = 0  
elif objj[i][2] == 2:  
    placeX = 45  
    placeY = 0  
elif objj[i][2] == 3:  
    placeX = 55  
    placeY = 0  
else:  
    placeX = 65  
    placeY = 0
```

Εικόνα 3.4.9: Συντεταγμένες τοποθέτησης αντικειμένων.

3.5 Διόρθωση κάμερας (Camera Calibration & Perspective Calibration)

Για τη διόρθωση της κάμερας ως προς την γεωμετρική παραμόρφωση πρέπει στο αρχείο camera_calibration να τεθεί η παράμετρος distortion_calibration σε κατάσταση True. Χρησιμοποιείται μια εικόνα σκακιέρας (βρίσκεται στο φάκελο Pattern), η οποία φωτογραφίζεται σε διάφορα σημεία του οπτικού πεδίου της κάμερας. Οι εικόνες που λαμβάνονται αποθηκεύονται στον

φάκελο images ώστε το αρχείο που είναι υπεύθυνο να κάνει αυτή τη διόρθωση, να διαβάσει τις εικόνες. Όταν η διαδικασία ολοκληρωθεί το πρόγραμμα δημιουργεί ορισμένα αρχεία στο φάκελο camera_data.

Από αυτά τα αρχεία το newmtx είναι ο νέος πίνακας της κάμερας (δηλαδή με την διόρθωση) και τα στοιχεία του χρειάζονται για το perspective calibration.

Οι εγγενείς παράμετροι είναι συγκεκριμένες για μια κάμερα και περιλαμβάνει πληροφορίες όπως εστιακό μήκος (f_x , f_y) και τα οπτικά κέντρα (c_x , c_y). Αυτοί οι παράμετροι είναι ο πίνακας της κάμερας (Camera Matrix).

$$\text{camera matrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Εικόνα 3.5.1: Πίνακας κάμερας.

Η κάμερα είναι τοποθετημένη σε ένα σταθερό σημείο, οπότε το perspective calibration γίνεται μια φορά. Για να πραγματοποιηθεί η διόρθωση της κάμερας ως προς την παραμόρφωση προοπτικής πρέπει η παράμετρος distortion_calibration του αρχείου camera_calibration να τεθεί σε κατάσταση False. Το perspective_pattern στο φάκελο Patterns περιέχει δυο εικόνες οι οποίες χρησιμοποιούνται για την συγκεκριμένη διόρθωση. Τοποθετείται ένα χαρτί με την εικόνα του αρχείου (αριθμημένες κουκκίδες) στο οπτικό πεδίο της κάμερας.



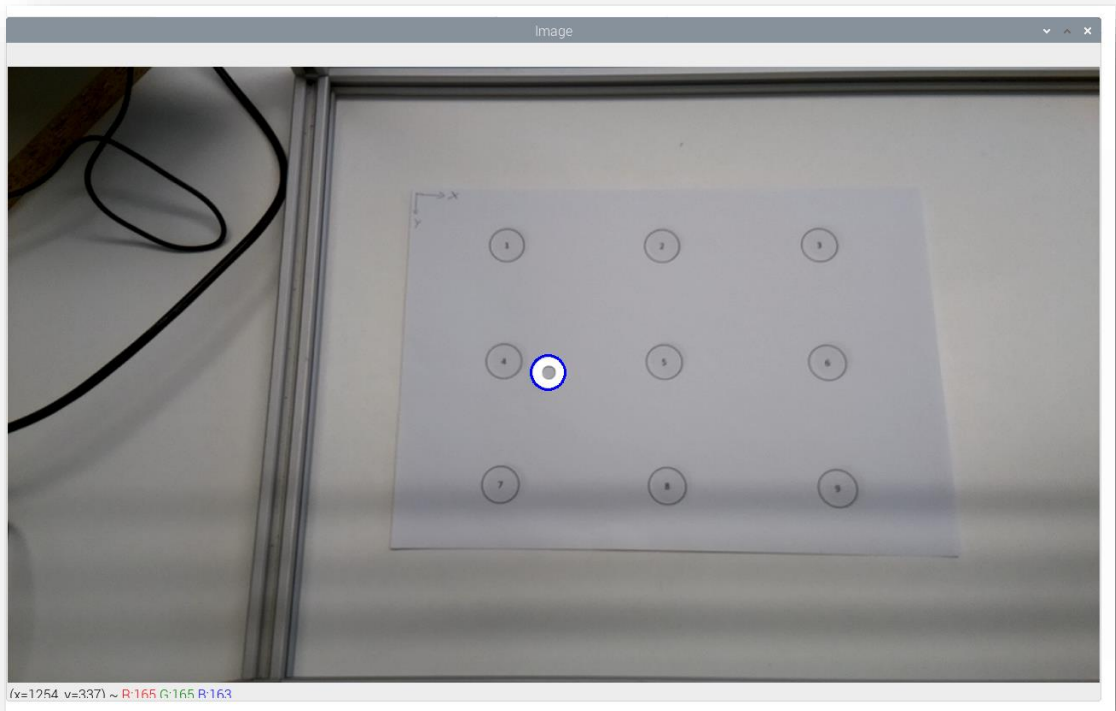
Εικόνα 3.5.2: Χαρτί με τις αριθμημένες κουκκίδες από το οπτικό πεδίο της κάμερας.

Από το πίνακα newmtx λαμβάνονται οι τιμές c_x, c_y και εντοπίζονται, μέσω της κάμερας, πάνω στο τοποθετημένο χαρτί. Έπειτα υπολογίζεται το ύψος από το εμφανιζόμενο σημείο, μέχρι το κέντρο του φακού της κάμερας.

Πρέπει να σημειωθεί ότι η αρχή των αξόνων της κάμερας είναι πάνω αριστερά, οπότε το χαρτί με τις αριθμημένες κουκκίδες πρέπει να μπει με τέτοιο τρόπο ώστε η αρχή των αξόνων και των δυο να είναι παράλληλα (δεν είναι απαραίτητο να ταυτίζονται).

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} [1.07752356e+03 & 0.00000000e+00 & 6.33806008e+02] \\ [0.00000000e+00 & 1.04926025e+03 & 3.57414424e+02] \\ [0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00] \end{bmatrix}$$

Εικόνα 3.5.3: Πίνακας του `newcam_mtx` της κατασκευής, με $c_x=633$ και $c_y=357$.



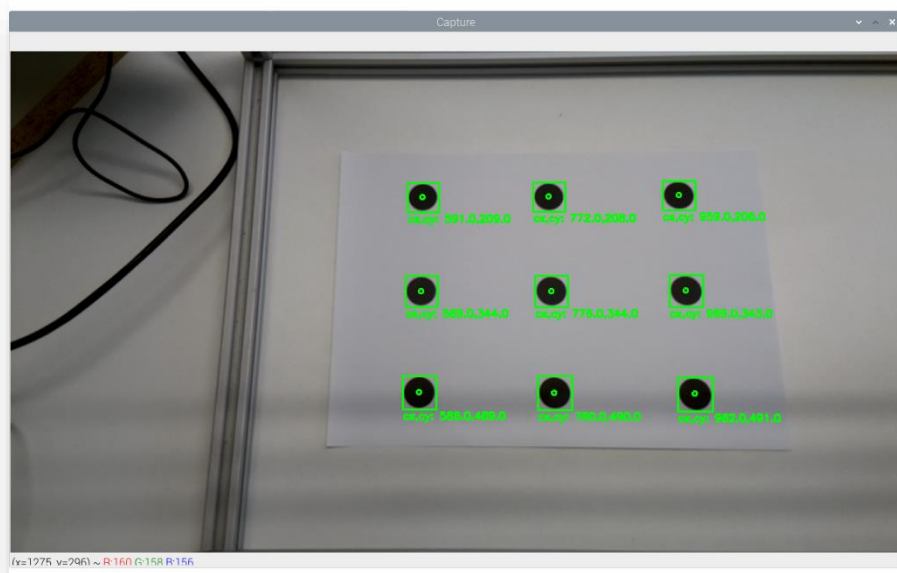
Εικόνα 3.5.4: Εντοπισμός του σημείου των `pixel cx,cy`.



Εικόνα 3.5.5: Υπολογισμός του ύψους.

Η διαδικασία της εύρεσης του ύψους γίνεται για κάθε αριθμημένη κουκκίδα, καθώς και σημειώνονται οι αποστάσεις X και Y των κουκκίδων με αρχή των αξόνων την άκρη του χαρτιού από το τη πρώτη κουκκίδα.

Στη συνέχεια τοποθετείται ένα άλλο χαρτί με την εικόνα των γεμάτων κουκκίδων στο ίδιο ακριβώς σημείο, έτσι ώστε να βρεθούν και να σημειωθούν τα pixel της κάθε κουκκίδας. Αυτό το κομμάτι μπορεί να πραγματοποιηθεί με την οπτική αναγνώριση του συστήματος, προγραμματίζοντας το να εμφανίζει τα pixel από τα κέντρα των αναγνωρισμένων αντικειμένων.



Εικόνα 3.5.6: Pixel κάθε κουκκίδας.

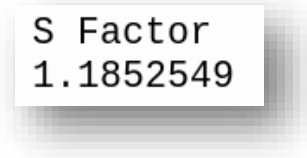
Όλα αυτά τα δεδομένα τοποθετούνται στο κώδικα και κατά την ολοκλήρωση της διαδικασίας, το πρόγραμμα δημιουργεί ορισμένα αρχεία στο φάκελο camera_data.

```
#X,Y,Z of the center
X=7.8
Y=10.9
Z=48.2
# 9 metric center points
metric=np.array([[X,Y,Z],
                 [5.5,3.9,47],
                 [14.2,3.9,51],
                 [22.8,3.9,52],
                 [5.5,10.6,48.5],
                 [14.2,10.6,48.6],
                 [22.8,10.6,50.4],
                 [5.5,17.3,47.4],
                 [14.2,17.3,47.3],
                 [22.8,17.3,49]], dtype=np.float32)

#MANUALLY INPUT THE DETECTED IMAGE COORDINATES HERE
#[u,v] center + 9 uv centers same as the metric center points
uv=np.array([[u,v],
             [590,214],
             [774,212],
             [959,207],
             [591,352],
             [780,349],
             [973,344],
             [591,498],
             [788,495],
             [990,493]], dtype=np.float32)
```

Εικόνα 3.5.7: Τοποθέτηση των δεδομένων.

Το S factor είναι μια πληροφορία που δίνεται στο χρήστη, ώστε να γνωρίζει το ποσοστό επιτυχίας της διόρθωσης της κάμερας που πραγματοποιήθηκε. Όσο πιο κοντά στη μονάδα βρίσκεται το S factor, τόσο πιο ορθή είναι η διόρθωση της κάμερας.

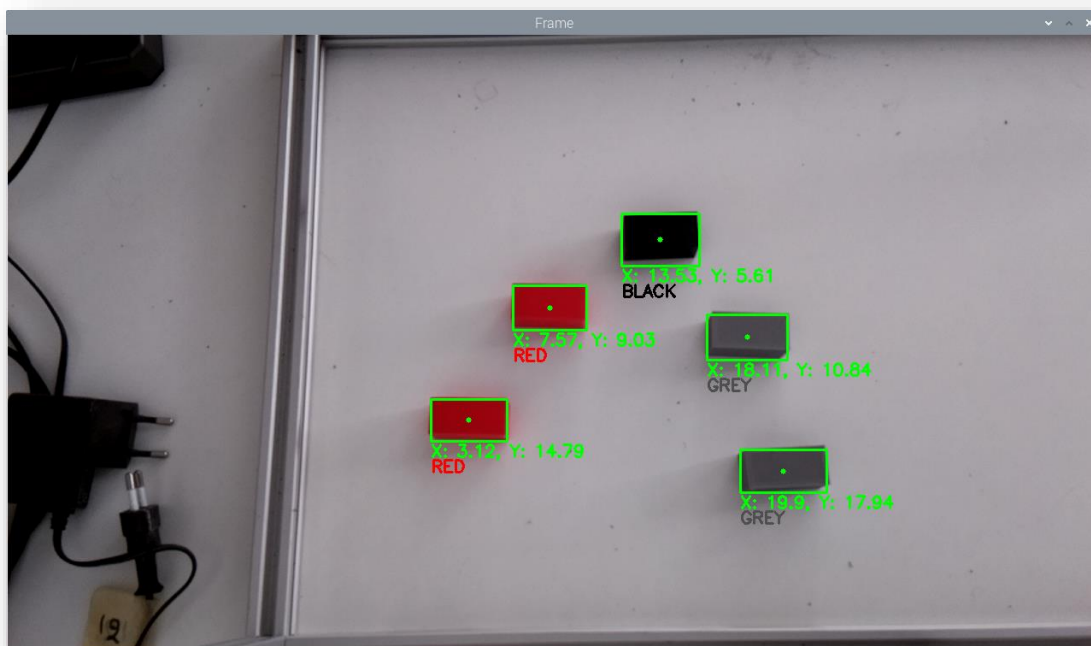


Εικόνα3.5.8: S factor.

3.6 Τρόπος λειτουργίας

Με την επιλογή «enable_robotic_arm = True», μέσα στον κώδικα του program κατά την εκκίνηση του προγράμματος ο ρομποτικός βραχίονας μετακινείται κατά τον άξονα X σε μια τιμή που έχει καθοριστεί (σημείο αναφοράς), ώστε να είναι μακριά από το οπτικό πεδίο της κάμερας. Στη συνέχεια, η κάμερα λαμβάνει ένα στιγμιότυπο, το οποίο δεν εμφανίζεται στον χρήστη, ώστε να ξεκινήσει η διαδικασία της αναγνώρισης των αντικειμένων.

Όταν τοποθετούνται τα αντικείμενα στον χώρο αντίληψης, η κάμερα ανιχνεύει τα αντικείμενα σε πραγματικό χρόνο και τοποθετεί σε ένα Bounding Box το κάθε αντικείμενο ξεχωριστά. Κάτω από το κάθε αναγνωρισμένο αντικείμενο εμφανίζει την πληροφορία της θέσης του αντικειμένου, σε πραγματικές διαστάσεις (X και Y), καθώς και το χρώμα του.



Εικόνα 3.6.2: Αναγνώριση αντικειμένων.

Η διαδικασία της αντίληψης ολοκληρώνεται μετά από 60 δευτερόλεπτα και εκείνη την στιγμή λαμβάνεται ένα στιγμιότυπο, το οποίο παραμένει μέχρι να ολοκληρωθεί η διαδικασία της τοποθέτησης όλων των αντικειμένων.

Μόλις τοποθετηθεί το τελευταίο αναγνωρισμένο αντικείμενο στο καθορισμένο σημείο, ο βραχίονας επανέρχεται στο σημείο αναφοράς και το σύστημα είναι έτοιμο να δεχθεί νέα αντικείμενα προς αναγνώριση και τοποθέτηση με την ίδια ακριβώς διαδικασία.

Πατώντας το πλήκτρο q στο πληκτρολόγιο, ο ρομποτικός βραχίονας μετακινείται στην αρχική του θέση ($X=0$). Ο τερματισμός του συστήματος με αυτό τον τρόπο γίνεται μόνο όταν το σύστημα βρίσκεται σε κατάσταση ηρεμίας, δηλαδή στην περίπτωση που ο βραχίονας έχει ολοκληρώσει την τοποθέτηση όλων των αντικειμένων, μετά από μια διαδικασία αναγνώρισης, ή στην περίπτωση όπου δεν έχει πραγματοποιηθεί καμία αναγνώριση αντικειμένου.

Στην περίπτωση της επιλογής «enable_robotic_arm = False», το ρομποτικό σύστημα ακολουθεί ακριβώς την ίδια διαδικασία, με την μόνη διαφορά ότι η διαδικασία της τοποθέτησης δεν πραγματοποιείται.

Με άλλα λόγια το ρομποτικό σύστημα ανιχνεύει συνέχεια τα αντικείμενα (με την παραπάνω διαδικασία), μέχρις ότου να δοθεί η εντολή του τερματισμού, πατώντας το πλήκτρο q στο πληκτρολόγιο. Σε αυτή την περίπτωση της επιλογής το σύστημα μπορεί να τερματιστεί με αυτόν τον τρόπο σε οποιαδήποτε στιγμή.

4. Επίλογος

4.1 Σύνοψη εργασίας

Ο ρομποτικός βραχίονας που κατασκευάστηκε είναι καρτεσιανού συστήματος τύπου Gantry και η αναγνώριση αντικειμένων πραγματοποιείται με την τεχνική Background Subtraction. Οι διαστάσεις της κατασκευής είναι 100x37x36 (cm) και ο χώρος εργασίας έχει διαστάσεις 80x20x18 (cm). Το σύστημα ελέγχεται από ένα Raspberry Pi το οποίο επικοινωνεί με ένα Arduino για τον έλεγχο των κινητήρων. Σκοπός του ρομποτικού βραχίονα είναι να εντοπίζει τα αντικείμενα που έχουν τοποθετηθεί σε ένα σημείο του χώρου εργασίας, να αναγνωρίζει το χρώμα του αντικειμένου και στη συνέχεια να τα τοποθετεί σε ένα άλλο σημείο του χώρου εργασίας, με βάση το χρώμα του αντικειμένου, το οποίο έχει οριστεί εξ' αρχής.

4.2 Προβλήματα και αντιμετώπιση

Ένα πρόβλημα που παρουσιάστηκε είναι η διόρθωση της κάμερας, διότι το S factor δεν ήταν αρκετά κοντά στο ένα.

Για να επιλυθεί το πρόβλημα χρειάστηκε, αρχικά, να ληφθούν περίπου τριάντα φωτογραφίες (με την εικόνα της σκακιέρας) για την διόρθωση της κάμερας ως προς την οπτική παραμόρφωση. Η σκακιέρα, σε όλες τις φωτογραφίες που λήφθηκαν, χρειάστηκε να βρίσκεται κοντά στο σημείο που θα γίνεται η οπτική αναγνώριση των αντικειμένων και όχι σε όλο το οπτικό πεδίο της κάμερας. Με αυτό τον τρόπο ο πίνακας που δημιουργείται βοηθά αρκετά στη λήψη των σωστών σημείων c_x και c_y τα οποία χρειάζονται στην διόρθωση της κάμερας για την παραμόρφωση προοπτικής και έχει ως αποτέλεσμα το S factor να είναι αρκετά κοντά στο ένα.

Ένα δεύτερο σημαντικό πρόβλημα το οποίο παρουσιάστηκε είναι ότι ο ρομποτικός βραχίονας έχει μηχανικά όρια στον άξονα Y. Συγκεκριμένα δεν μπορεί να ξεπεράσει τους δέκα πόντους (10cm) προς μια κατεύθυνση.

Η οπτική αναγνώριση των αντικειμένων πραγματοποιείται σε όλο το οπτικό πεδίο της κάμερας, με αποτέλεσμα να ξεπερνά τα μηχανικά όρια του ρομποτικού βραχίονα, όχι μόνο στον άξονα Y άλλα και στον άξονα X, αφού «βλέπει» και έξω από τα όρια της κατασκευής. Για την λύση του προβλήματος τέθηκαν όρια στον χώρο οπτικής αναγνώρισης. Συγκεκριμένα μετά την αναγνώριση του αντικειμένου και τον υπολογισμό του κέντρου του, πραγματοποιείται ένας έλεγχος εάν το κέντρο του αντικειμένου βρίσκεται εντός των ορίων των pixel που έχουν οριστεί πάνω στο οπτικό πεδίο της κάμερας. Σε περίπτωση που το κέντρο του αντικειμένου είναι εντός των ορίων, το πρόγραμμα το αναγνωρίζει ως αντικείμενο, εμφανίζοντας τις συντεταγμένες και το είδος του χρώματος. Στην περίπτωση όπου το κέντρο του αντικειμένου δεν είναι στα επιθυμητά όρια, τότε, το πρόγραμμα αγνοεί το αντικείμενο αυτό.

4.3 Παρατηρήσεις / Συμπεράσματα

Το ρομποτικό σύστημα έχει αρκετά καλή επαναληψιμότητα. Το μειονέκτημα είναι ότι η ακρίβεια θέσης δεν είναι καλή στην περίπτωση που πρόκειται να συλλέξει το αναγνωρισμένο αντικείμενο. Αυτό συμβαίνει διότι τα δεδομένα για τον άξονα Z έχουν δοθεί μέσω του προγράμματος και όχι μέσω της κάμερας. Με αυτό τον τρόπο το ύψος είναι σταθερό σε κάθε οπτικό σημείο της κάμερας, με αποτέλεσμα ο αλγόριθμος για τον ρομποτικό βραχίονα να μην υπολογίζει σωστά τις μοίρες που χρειάζονται για την σωστή θέση του βραχίονα. Ο αλγόριθμος υπολογίζει το σημείο με βάση τα δεδομένα Y και Z, αφού κινείται ταυτόχρονα σε αυτούς τους άξονες.

Η αρχή των αξόνων του ρομποτικού βραχίονα δεν ταυτίζεται με την αρχή των αξόνων για τις συντεταγμένες X και Y της κάμερας. Η χειροκίνητη διόρθωση η οποία πραγματοποιείται δεν μπορεί να υπολογιστεί με απόλυτη ακρίβεια.

Ο συνδυασμός αυτών των δυο δημιουργεί ένα μικρό σφάλμα ακρίβειας.

Τα όρια των επιθυμητών χρωμάτων επιλέγονται από τον χρήστη, με αποτέλεσμα ορισμένες φορές να μην πραγματοποιείται σωστή αναγνώριση. Αυτό οφείλεται κυρίως από την ποιότητα του φωτισμού του χώρου. Ακόμα και στη περίπτωση που πραγματοποιήθηκε η αναγνώριση των επιθυμητών χρωμάτων σε πραγματικό χρόνο, όπως αναφέρθηκε παραπάνω για τον καθορισμό των ορίων, ο φωτισμός του χώρου, εκείνη την στιγμή, μπορεί να μην είναι ίδιος κατά τη διάρκεια λειτουργίας της κατασκευής (πχ διαφορετική ποσότητα και ποιότητα φυσικού φωτισμού του χώρου κατά τις πρωινές ή μεσημβρινές ώρες).

4.4 Προτάσεις μελλοντικής εξέλιξης

Το ρομποτικό σύστημα έχει αρκετές ικανότητες εξέλιξης, όμως για να πραγματοποιηθούν χρειάζονται οικονομικοί πόροι και ελεύθερο χρόνο ενασχόλησης. Παρακάτω δίνονται ορισμένες προτάσεις με τις οποίες το ρομποτικό σύστημα μπορεί να εξελιχθεί, μειώνοντας το σφάλμα της ακρίβειας θέσης, πραγματοποιεί καλύτερη ανίχνευση και αναγνώριση των αντικειμένων και προσθέτοντας ορισμένα συστήματα ασφαλείας.

Για την επίλυση του πρώτου προβλήματος μπορεί να τοποθετηθεί μια δεύτερη κάμερα η οποία να υπολογίζει το ύψος μεταξύ του ρομποτικού βραχίονα και του αντικειμένου και συνδυάζοντας τα δεδομένα των καμερών να παρέχεται μια τρισδιάστατη πληροφορία, η οποία θα βοηθήσει στην καλύτερη ακρίβεια θέσης του βραχίονα πάνω στο αντικείμενο.

Για την πιο αποτελεσματική αναγνώριση των χρωμάτων μπορεί να χρησιμοποιηθεί η αναγνώριση με Deep Learning ή Machine Learning. Σε αυτές τις περιπτώσεις το σύστημα εκπαιδεύεται με τέτοιο τρόπο, ώστε η αναγνώριση να είναι ακριβής σε κάθε περίπτωση.

Η ανίχνευση των αντικειμένων και η αναγνώριση των χρωμάτων με Deep Learning ή Machine Learning είναι μια σημαντική βελτίωση που μπορεί να πραγματοποιηθεί στο ρομποτικό σύστημα. Η ανίχνευση των αντικειμένων με αυτούς τους τρόπους, έχει μεγαλύτερη ακρίβεια και ταχύτητα.

Μηχανολογικά μπορούν να τοποθετηθούν στο ρομποτικό σύστημα δυο limit switch στα άκρα του άξονα X, και ένα estop. Τα limit switch προστατεύουν το σύστημα μηχανολογικά, διακόπτοντας την μετακίνηση προς μια κατεύθυνση όταν φτάσει σε κάποιο άκρο. Στην ουσία οριοθετούν προγραμματιστικά το σύστημα.

Το estop είναι ένα σημαντικό μπουτόν. Με το πάτημα του, το σύστημα ακινητοποιείται σε όποια κατάσταση ή θέση βρίσκεται. Με άλλα λόγια μπορεί να προστατέψει το ίδιο το σύστημα σε περίπτωση βλάβης και το πιο σημαντικό είναι μπορεί να προστατέψει τον ίδιο το χειριστή ή το προσωπικό το οποίο βρίσκεται κοντά στο ρομποτικό σύστημα.

Επίσης μπορούν να τοποθετηθούν αισθητήρες για την επιτήρηση του χώρου, έτσι ώστε σε περίπτωση όπου υπάρξει κίνηση ενός ατόμου μέσα στο χώρο, το ρομποτικό σύστημα να σταματά την λειτουργία του και με κάποιου είδους σήμανσης (πχ φάρος, σειρήνα, κλπ.), να ειδοποιεί το άτομο ότι βρίσκεται στο χώρο εργασίας του συστήματος, εν ώρα λειτουργίας.

Τέλος μπορεί να γίνει μια διαμόρφωση της αρπάγης ώστε να περιστρέφεται γύρω από τον άξονα της, όπου με αυτό τον τρόπο θα υπάρχει η δυνατότητα της λήψης των αντικειμένων από όλες τις θέσεις (οριζόντια, κάθετα, διαγώνια).

Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές

- [1] Hans Peter Moravec, “Robot”, Διαθέσιμο: <https://www.britannica.com/technology/robot-technology>
- [2] ieee, “What is robot” Διαθέσιμο: <https://robots.ieee.org/learn/what-is-a-robot/>
- [3] The Investigators, “Robotic Aspects”, Διαθέσιμο: <https://x03478.wixsite.com/xinchaodume/robotic-aspects>
- [4] Built In, “Robotics”, Διαθέσιμο: <https://builtin.com/robotics>
- [5] A3 Robotics, “Humanoid Robots”, Διαθέσιμο: <https://www.automate.org/a3-content/service-robots-humanoid-robots>
- [6] Techopedia, “Autonomous Robots”, Διαθέσιμο: <https://www.techopedia.com/definition/32694/autonomous-robot>
- [7] Greg Conrad, “Teleoperated Robots: The Industrial Future Using AR and VR”, 2018, Διαθέσιμο: <https://blog.robotiq.com/teleoperated-robots-the-industrial-future-using-ar-and-vr>
- [8] Jeremy Gottlieb, David Leech Anderson, “Augmenting Robots”, Διαθέσιμο: https://mind.ilstu.edu/curriculum/medical_robotics/augmenting.html
- [9] Wikipedia, “Robotic arm”, Διαθέσιμο: https://en.wikipedia.org/wiki/Robotic_arm
- [10] Danielle Collins, “What is a Cartesian robot”, 2018, Διαθέσιμο: <https://www.linearmotiontips.com/what-is-a-cartesian-robot/>
- [11] RobotWork, “What Are Gantry Robots”, Διαθέσιμο: <https://www.robots.com/faq/what-are-gantry-robots>
- [12] Anish Devasia, “What are Cartesian Robots”, 2021, Διαθέσιμο: <https://control.com/technical-articles/what-are-cartesian-robots/>
- [13] DIY Robotics, “What is SCARA”, Διαθέσιμο: <https://diy-robotics.com/blog/scara-robots/>
- [14] DIY Robotics, “What’s an Articulated Robot”, Διαθέσιμο: <https://diy-robotics.com/blog/articulated-robots/>
- [15] MWES Engineered Systems, “Cylindrical Robots”, Διαθέσιμο: <https://www.mwes.com/cylindrical-robots/>
- [16] MWES Engineered Systems, “Delta Robots”, Διαθέσιμο: <https://www.mwes.com/delta-robots/>
- [17] MWES Engineered Systems, “Polar/Spherical Robots”, Διαθέσιμο: <https://www.mwes.com/polar-spherical-robots/>
- [18] FRITZ AI, “object detection guide”, Διαθέσιμο: <https://www.fritz.ai/object-detection/>
- [19] Helly M Desai, Vaibhav Gandhi, “A Survey: Background Subtraction Techniques”, International Journal of Scientific & Engineering Research, 2014

- [20] Wikipedia, “Distance matrix”, Διαθέσιμο: https://en.wikipedia.org/wiki/Distance_matrix
- [21] MathWorks, “What Is Object Detection”, Διαθέσιμο: <https://www.mathworks.com/discovery/object-detection.html>
- [22] Rohith Gandhi, “R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms”, 2018, Διαθέσιμο: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>
- [23] Chinmoy Borah, “Evolution of Object Detection”, 2020, Διαθέσιμο: <https://medium.com/analytics-vidhya/evolution-of-object-detection-582259d2aa9b>
- [24] Analytics India Magazine, “Top 8 Algorithms For Object Detection”, 2020, Διαθέσιμο: <https://analyticsindiamag.com/top-8-algorithms-for-object-detection/>
- [25] Wikipedia, “Haar-like feature”, Διαθέσιμο: https://en.wikipedia.org/wiki/Haar-like_feature
- [26] Hussain Mujtaba, “Face Detection using Viola Jones Algorithm”, 2020, Διαθέσιμο: <https://www.mygreatlearning.com/blog/viola-jones-algorithm/>
- [27] Soret Lee, “Understanding Face Detection with the Viola-Jones Object Detection Framework”, 2020, Διαθέσιμο: <https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>
- [28] Aishwarya Singh, “Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor”, 2019, Διαθέσιμο: <https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/>
- [29] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye, “Object Detection in 20 Years: A Survey”, Διαθέσιμο: <https://arxiv.org/pdf/1905.05055.pdf>
- [30] Nasim Mansurov, “What is Lens Distortion”, 2020, Διαθέσιμο: <https://photographylife.com/what-is-distortion>
- [31] Wikipedia, “Perspective distortion”, Διαθέσιμο: [https://en.wikipedia.org/wiki/Perspective_distortion_\(photography\)](https://en.wikipedia.org/wiki/Perspective_distortion_(photography))
- [32] Paco Garcia, “Computer Vision Pick and Place for Toys using Raspberry Pi and Arduino”, 2019, Διαθέσιμο: <https://medium.com/@pacogarcia3/computer-vision-pick-and-place-for-toys-using-raspberry-pi-and-arduino-68a04874c039>
- [33] OpenCV “About OpenCV”, Διαθέσιμο: <https://opencv.org/about/>
- [34] NumPy “What is NumPy”, Διαθέσιμο: <https://numpy.org/doc/stable/user/whatisnumpy.html>
- [35] mint, “Men and machines in India”, Διαθέσιμο: <https://www.livemint.com/Opinion/nmcz8boB5Bn5JgT0TAvtM/Men-and-machines-in-India.html>
- [36] Carlos Gonzalez, “What’s the Future Role for Humanoid Robots”, 2017, Διαθέσιμο: <https://www.machinedesign.com/mechanical-motion-systems/article/21836113/whats-the-future-role-for-humanoid-robots>

- [37] Discovery Home, “Is Xiaomi MIJIA 1C 2-in-1 Robot Vacuum Cleaner Worth Buying”, Διαθέσιμο: <https://m.gearbest.com/community/L4SubH>
- [38] Robot Automate, “Tele-operated robots: keeping the human in the loop”, 2017, Διαθέσιμο: <https://robotauto.wordpress.com/2012/12/10/tele-operated-robots-keeping-the-human-in-the-loop/>
- [39] The TrenThings, “AUGMENTING ROBOTS”, Διαθέσιμο: <https://the-trenthings.jimdofree.com/robots/augmenting-robots/>
- [40] Fabacademy, “Mechanical design and machine design”, 2018, Διαθέσιμο: <http://archive.fabacademy.org/2018/labs/fablabccept/students/adhitya-ss/week-15---16.html>
- [41] RobothalloIs Xiaomi MIJIA 1C 2-in-1 Robot Vacuum Cleaner Worth Buyingffame, “SCARA”, Διαθέσιμο: <http://www.robotalloffame.org/inductees/06inductees/scara.html>
- [42] Omron “Industrial robot solutions” Διαθέσιμο: https://www.airlinehyd.com/literature_catalog/omron/industrial_robot_solutions_bro_en_201603_y82ie01.pdf
- [43] Industry plaza “Articulated robot - KF194”, Διαθέσιμο: <https://www.industry-plaza.com/articulated-robot-kf194-p271541358.html>
- [44] Thnet, “Robotic working envelopes”, Διαθέσιμο: <http://thnet.co.uk/thnet/robots/25.htm>
- [45] Cfp, Cylindrical Robots, Διαθέσιμο: <https://robot.cfp.co.ir/en/robots/cylindrical>
- [46] Robot machines, “What Is Cylindrical Robot”, Διαθέσιμο: <https://www.robot-machines.com/what-are-cylindrical-robots/>
- [47] Servotronix, “DELTA robot - linear model”, Διαθέσιμο: http://softmc.servotronix.com/wiki/DELTA_robot_-_linear_model
- [48] Omron, “Quattro 650H / HS”, Διαθέσιμο: <https://www.omron-ap.co.in/products/family/3513/dimension.html>
- [49] Machine design, “What’s the Difference Between Industrial Robots”, 2016, Διαθέσιμο: <https://www.machinedesign.com/markets/robotics/article/21835000/whats-the-difference-between-industrial-robots>
- [50] OpenCV, “How to Use Background Subtraction Methods”, Διαθέσιμο: https://docs.opencv.org/master/d1/dc5/tutorial_background_subtraction.html
- [51] MathWorks, “Getting Started with SSD Multibox Detection”, Διαθέσιμο: <https://www.mathworks.com/help/vision/ug/getting-started-with-ssd.html>
- [52] Luis R Ramírez-Hernández, Julio C. Rodríguez-Quíñonez, Moises J Castro-Toscano, Daniel Hernandez-Balbuena, “Improve three-dimensional point localization accuracy in stereo vision systems using a novel camera calibration method”, International Journal of Advanced Robotic Systems, 2020
- [53] OpenCV, “Camera Calibration”, Διαθέσιμο: https://docs.opencv.org/3.3.0/dc/dbb/tutorial_py_calibration.html

[54] Κώδικας, Διαθέσιμο: <https://github.com/pacogarcia3/hta0-horizontal-robot-arm>