

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση και Υλοποίηση Ενσωματωμένου
Συστήματος Χαμηλού Κόστους για Εύρεση
Θέσης μη Επανδρωμένων Αεροσκαφών που
Πετούν σε Σχηματισμό

Συγγραφέας:

Χρήστος Σπυρίδακης

Εξεταστική επιτροπή:

Καθ. Απόστολος ΔΟΛΛΑΣ (Επιβλέπων)

Αναπ. Καθ. Ευτύχιος ΚΟΥΤΡΟΥΛΗΣ

Αναπ. Καθ. Παναγιώτης ΠΑΡΤΣΙΝΕΒΕΛΟΣ



Η παρούσα διπλωματική εκπονήθηκε στα πλαίσια απόκτησης του
διπλώματος του Ηλεκτρολόγου Μηχανικού και Μηχανικού

Υπολογιστών στην σχολή

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εργαστήριο Μικροεπεξεργαστών και Τηλικού

Φεβρουάριος 25, 2022

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Περίληψη

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Σχεδίαση και Υλοποίηση Ενσωματωμένου Συστήματος Χαμηλού Κόστους για Εύρεση Θέσης μη Επανδρωμένων Αεροσκαφών που Πετούν σε Σχηματισμό

Η ανάγκη προσδιορισμού της θέσης αντικειμένων έχει παρουσιαστεί εδώ και χιλιάδες χρόνια, με τα τελευταία 70 να αξιοποιούμε ακόμα και ψηφιακά μέσα για να την επιτύχουμε. Συνδυάζοντας την ιδέα αυτή με τη πλέον αναπτυσσόμενη τεχνολογία των UAV, στην συγκεκριμένη εργασία γίνεται προσπάθεια υλοποίησης της πρώτης γενιάς ενός οικονομικού συστήματος εκτίμησης του position αντικειμένου από συνεργατικά συμήνη drones. Για να γίνει διαχριτό το πλήθος των διαφορετικών μεθόδων που ήδη χρησιμοποιούνται στην βιβλιογραφία για επίλυση παρόμοιων προβλημάτων διεξήχθη ένα prior art search. Συχνά επιλέγεται η χρήση ραδιοσυχνοτήτων, όπως GPS, WiFi, UWB, κυψελωτά δίκτυα και δίκτυα Lora ώστε να πραγματοποιηθεί το localization, άλλες φορές ηχητικά κύματα, και άλλες οπτικά, όπως με χρήση καμερών και LiDAR. Ακόμη, ήταν σημαντικό να γίνουν κατανοητοί οι μαθηματικοί φορμαλισμοί που ακολουθούνται σε κάθε προσέγγιση, ώστε να καθοριστεί η πολυπλοκότητας τους. Αφού, βάση των αρχών λειτουργίας τους και των πόρων που χρειάζεται ο καθένας, καταλήγουμε στο κατάλληλο hardware του ενσωματωμένου συστήματος. Αυτές αφορούν στον τρόπο υπολογισμού ενδιάμεσων πληροφοριών - όπως εκτίμηση απόστασης και γωνίας - μέσω τεχνικών RSSI, TDoA, AoA ακόμα και Stereo Vision, που χρησιμοποιούνται σε αλγόριθμους όπως τον Trilateration, Triangulation ή Hyperbolic Positioning για την εκτίμηση τελικά της θέσης. Με την γνώση των αναγκών της κάθε προσέγγισης, έγινε η σχεδίαση ενός συστήματος δύο επιπέδων, όπου μέσω monocular vision και structure from reference εκτιμήθηκε η απόσταση καθορισμένου αντικειμένου από worker nodes, και μέσω της αρχής λειτουργίας του Multilateration, επιτεύχθηκε στο master node να υπολογιστεί η θέση του object. Πειράματα πραγματοποιήθηκαν σε εσωτερικό και εξωτερικό χώρο από μεμονωμένο κόμβο, ενώ προσομοιώθηκε και η λειτουργία στο σύνολο του συστήματος. Για λόγους πληρότητας/επαλήθευσης ανίχνευσης, υλοποιήθηκε επίσης μηχανισμός προσδιορισμού του ID του εκτιμώμενου αντικειμένου μέσω ανάλυσης συχνότητας ενός blinking led.

TECHNICAL UNIVERSITY OF CRETE

Abstract

School of Electrical and Computer Engineering

Design and Implementation of a Low Cost Embedded System for Localization of Drones Flying in Swarms

The requirement for object localization has existed for thousands of years, and over the past 70 years, we have even taken advantage of digital methods to achieve it. The following thesis focuses on the attempt to combine the above idea with the most advanced technology of UAVs and implement the first generation of a low-cost system for estimating the position of an object with the help of a swarm of drones. The prosecution of prior art research has distinguished the number of different methods already used in the literature to solve similar problems. In such methods, we often use Radio Frequencies - like GPS, WiFi, UWB, cellular, and Lora networks -, sound waves, or vision-related ones by operating cameras and LiDAR. It was also essential to understand the mathematical background behind these approaches and determine their complexities since the selection of the hardware for this kind of embedded systems often relies on the operating principles and resources that each one needs. These are related to how we calculate distances and angles by applying RSSI, TDoA, AoA, or even Stereo Vision techniques, and how we utilize this information in algorithms like Triangulation, Trilateration, or Hyperbolic Positioning to achieve localization. By comprehending each approach necessity, this work presents a two-tier system, in which worker nodes estimate the distance between the object of interest and camera through monocular vision and structure from reference. Then, through the principle of Multilateration, we finally achieve estimating its position by the system orchestrator node, called master. We first performed experiments in both indoor and outdoor scenarios from single worker nodes. Then, we used the data taken from the above procedure to simulate the operation of the entire proposed system. To complete this work, we also introduce a mechanism to estimate the ID of the detected object through frequency analysis of flashing led.

Ευχαριστίες

Η διεξαγωγή μίας διπλωματικής εργασίας δεν είναι ατομική αρμοδιότητα. Πολλά άτομα υπάρχουν με τον ένα ή άλλο τρόπο πίσω από αυτήν, χωρίς την βοήθεια των οποίων δεν θα είχε ολοκληρωθεί.

Αρχικά, είμαι σίγουρος ότι η συγκεκριμένη εργασία δεν θα ήταν εφικτή χωρίς την βοήθεια του επιβλέπων καθηγητή μου. Θα ήθελα συνεπώς να ευχαριστήσω τον καθηγητή και επιβλέπων της συγκεκριμένης διπλωματικής εργασίας, Καθ. Απόστολο Δόλλα, ο οποίος μου έδωσε την ευκαιρία να πραγματοποιήσω την εργασία μου υπό την καθοδήγηση του, και ήταν δίπλα μου σε όλη την διάρκεια της, όχι μόνο με συμβουλευτική, καθοδηγητική και τεχνική φύση αλλά και με ανθρώπινη. Ένας επίσης σημαντικός - για εμένα - λόγος, για τον οποίο θα ήθελα να ευχαριστήσω τον κύριο Δόλλα, είναι διότι, ήταν από τους καθηγητές που κατά την διάρκεια των σπουδών μου^{*} υπήρξε μία καθοριστική παρουσία καθηγητή και ανθρώπου, που μέσω του τρόπου διδασκαλίας του, και του πάθος προς την επιστήμη του μηχανικού, με έκανε αγαπήσω τον χώρο των ενσωματωμένων συστημάτων τον οποίο μέχρι τότε δεν γνώριζα, και τελικά ωθήμηκα να ασχοληθώ με αυτόν.

Επίσης, θα ήθελα να ευχαριστήσω τους καθηγητές, Αναπ. Καθ. Ευτύχιο Κουτρούλη και Αναπ. Καθ. Παναγιώτη Παρτσινέβελο, ως μέλη της τριμελής επιτροπής, καθώς, και για την βοήθεια τους, στον προσδιορισμό των requirements που έπρεπε να ικανοποιεί το σύστημα.

Συμπληρωματικά, θα ήθελα να ευχαριστήσω το Πολυτεχνείο Κρήτης και όλους τους υπόλοιπους καθηγητές για τις γνώσεις που μου έδωσαν όλα αυτά τα χρόνια, οι οποίες με βοήθησαν κατά την διάρκεια της διπλωματικής μου. Σημαντική ήταν επίσης η βοήθεια των εργαστηρίων MHL και SenseLab, μαζί με το διδακτικό προσωπικό και τους συναδέλφους που βρίσκονται σε αυτά. Θα ήθελα συνεπώς να ευχαριστήσω τον κύριο Κιμιωνή Μάρκο καθώς και τους συναδέλφους που με βοήθησαν να σχεδιάσω το σύστημα και να ξεπεράσω προβλήματα που προέκυψαν στην υλοποίηση του, συγκεκριμένα τους, Αντωνόπουλο Άγγελο, Φελέκη Παναγιώτη και Φωτάκη Τζανή.

Τέλος, θα ήθελα να ευχαριστήσω τους χρυφούς ήρωες μου. Τους ανθρώπους που ήταν εκεί, ήδη από τα πρώτα βήματα, πριν ακόμα ξεκινήσω το ταξίδι στο κόσμο του μηχανικού. Ένα μεγάλο ευχαριστώ συνεπώς ανήκει στους γονείς, συγγενείς και φίλους που ήταν δίπλα μου όλα αυτά τα χρόνια, καθώς και στις Άννα Μαρία Ζαμπετάκη και Αγγελική Σηφάκη για την υποστήριξη τους.

Σπυριδάκης Χρήστος
Χανιά, 2022

Περιεχόμενα

Περίληψη	iii
Abstract	v
Ευχαριστίες	vii
Περιεχόμενα	ix
Λίστα εικόνων	xiii
Λίστα πινάκων	xvii
Λίστα Αλγορίθμων	xix
Φυσικές Σταθερές	xxi
Λίστα συμβόλων	xxiii
Κατάλογος ακρωνυμίων	xxv
1 Εισαγωγή	1
1.1 Μη επανδρωμένα οχήματα	2
1.1.1 Ιστορία και τύποι drone	2
1.1.2 Χαρακτηριστικά	4
1.2 Κίνητρο	5
1.2.1 Φύση	5
1.2.2 Συμήνη και εφαρμογές	6
1.3 Ερευνητικοί στόχοι και συνεισφορά	7
1.4 Περιεχόμενα Διπλωματικής	7
2 Προϋπάρχουσες προσεγγίσεις	9
2.1 Ραδιοσυχνότητες	10
2.1.1 WiFi	10
2.1.2 UWB	11

2.1.3	Κυψελωτά δίκτυα	13
2.1.4	Lora	13
2.1.5	RTK GNSS	15
2.2	Ηχητικά κύματα	16
2.3	Optical Oriented	17
2.3.1	Κάμερα	17
2.3.2	LiDAR	17
2.3.3	VSLAM	19
2.3.4	Motion Capture System	20
3	Θεωρητικό Υπόβαθρο	21
3.1	Εντοπισμός θέσης με βάση την ενέργεια	24
3.1.1	Εκτίμηση Απόστασης/Γωνίας	24
	Received Signal Strength	24
	Χρόνος μετάδοσης σήματος	25
	Εκτίμηση γωνίας	29
	Amplitude response	29
	Phase response	30
3.1.2	Υπολογισμός θέσης	31
	Trilateration	32
	Bounding Box	34
	Triangulation	35
	Multilateration	36
	Πιθανοκρατικοί μέθοδοι	37
3.1.3	Localization Αλγόριθμοι	38
	One-hop vs Multi-hop	39
	Anchor-based vs Anchor-free	39
	Relative vs Absolute Positioning	40
	Indoor vs Outdoor Scenarios	40
	Distributed vs Centralized Position Computation	40
	Range-based vs Range-free	41
3.2	Εκτίμηση θέσης με οπτικούς τρόπους	43
3.2.1	Structure from Reference	43
3.2.2	Structure from Motion	45
3.2.3	Stereo Vision	45
3.3	Σκοποί Διπλωματικής	46
4	Διαδικασία Σχεδίασης και Υλοποίησης	49
4.1	Τεχνολογίες και εργαλεία	50

4.1.1	Embedded Linux System	50
4.1.2	ROS	52
4.1.3	Λειτουργικό σύστημα	52
4.1.4	OpenCV	53
4.1.5	Κάμερα	53
4.1.6	GPS	53
4.1.7	IMU	54
4.1.8	Breakout Board	55
4.1.9	System Overview	55
4.2	Environment	56
4.2.1	Παραμετροποίηση OS	56
4.2.2	Επικοινωνία αισθητήρων	57
	Σειριακή Επικοινωνία	57
	Επικοινωνία I2C	59
4.3	Camera	60
4.3.1	Camera Calibration	62
4.3.2	Χρήση κάμερας	63
4.3.3	Εντοπισμός Αντικειμένου	64
	Με βάση το χρώμα	64
	Με βάση το σχήμα	66
	Εναλλακτικές προσεγγίσεις	68
4.3.4	Εκτίμηση απόστασης	69
4.3.5	Εκτίμηση γωνίας	70
4.4	Διαμοιρασμός μηνυμάτων	71
4.5	Συγχρονισμός μηνυμάτων	72
4.6	Εντοπισμός θέσης αντικειμένου	72
4.7	Τρισδιάστατη απεικόνιση	74
4.8	Επαλήθευση ID του detected object	74
5	Επαλήθευση Λειτουργίας και Αποτελέσματα	77
5.1	Επαλήθευση λειτουργίας μεμονωμένου node	77
5.1.1	Περιβάλλον δοκιμών	77
5.1.2	Απόδοση του συστήματος	80
5.1.3	Λαμβανόμενα δεδομένα και απεικόνιση	84
5.2	Δεδομένα πολλαπλών κόμβων	87
5.2.1	Διαδικασία λήψης δεδομένων	87
5.2.2	Εκτίμηση θέσης και απεικόνιση	90
5.3	Επαλήθευση προσδιορισμού ID	92

6 Τελικά Συμπεράσματα και Μελλοντικές Εξελίξεις	95
6.1 Συμπεράσματα	95
6.2 Μελλοντικές Εξελίξεις	97
References	99

Λίστα εικόνων

1.1 Παραδείγματα των UAV	3
1.2 Κυρίαρχοι άξονες αναφοράς	4
2.1 PBR και PSL προσεγγίσεις εντοπισμού	11
2.2 Αναπαράσταση του spring system	12
2.3 Κυψελωτά δίκτυα	13
2.4 Σύνθεση συστήματος Lora	14
2.5 Αρχή λειτουργίας RTK	15
2.6 Χρήση ακουστικών διατάξεων για εντοπισμό θέσης	16
2.7 LiDAR	18
2.8 Παράδειγμα αντιληπτικής ικανότητας VSLAM σε drone	19
2.9 Χρήση του Optitrack Motion Capture System για εντοπισμό σμήνος από drone	20
3.1 Στοιχεία των Localization Systems	22
3.2 Επισκόπηση των Localization Systems (RF-based approach)	23
3.3 Ιδανική εξάρτηση του RSSI με βάση την απόσταση	25
3.4 Περιπτώσεις Time of Arrival	26
3.5 Περιπτώσεις Time Difference of Arrival	27
3.6 TDoA υπερβολές βασισμένες στο Δt	28
3.7 Pattern ευαισθησίας της anisotropic κεραίας	30
3.8 Παραδείγματα Antenna Arrays	31
3.9 Παραδείγματα του Trilateration	32
3.10 Bounding Box παράδειγμα	35
3.11 Principles of Triangulation	35
3.12 Triangulation παραδείγματα	36
3.13 Multilateration παραδείγματα	37
3.14 Παράδειγμα γράφου	39
3.15 Optical illusion που δημιουργείται λόγο του depth ambiguity	44
3.16 Optical axis του αντικειμένου - κάμερας	44
3.17 Τυπικό παράδειγμα κίνησης εικόνας για ανακατασκευή μέσω Structure from Motion της σκηνής	45

3.18 Stereo Vision	46
3.19 Thesis drone swarm system overview	47
4.1 High level system block diagram	50
4.2 Υποψήφια Embedded Linux Systems για την υλοποίηση	51
4.3 Η κάμερα που χρησιμοποιήθηκε για τον εντοπισμό και ανίχνευση του αντικειμένου	53
4.4 Το GPS που χρησιμοποιήθηκε	54
4.5 Adafruit 10 DoF IMU	55
4.6 Raspberry Pi breakout	55
4.7 Το πρώτυπο σύστημα που σχεδιάστηκε στα πλαίσια της διπλωματικής	56
4.8 Ιδανικό μοντέλο του pinhole	60
4.9 Οι πιο συνηθισμένοι τύποι distortion των φωκών	62
4.10 Matlab camera toolkit	63
4.11 ROS - camera related nodes/topics setup	63
4.12 Color Spaces	64
4.13 Επεξεργασία για ανίχνευση με βάση το χρώμα	65
4.14 Ανίχνευση μπάλας με βάση το χρώμα	65
4.15 Convolution παράδειγμα	66
4.16 Επεξεργασία για ανίχνευση με βάση το σχήμα	67
4.17 Ανίχνευση μπάλας με βάση το σχήμα	67
4.18 Circle Hough Transform	68
4.19 Παράδειγμα από το θετικό dataset που δημιουργήθηκε, για να γίνει train το μοντέλο του haar cascade.	69
4.20 Συσχέτιση AFOV με focal length	70
4.21 Η δομή του μηνύματος που αποστέλλει κάθε node	71
4.22 High level sequence diagram για τον συγχρονισμό των πακέτων . .	72
4.23 Παραδείγματα προσεγγίσεων καθορισμού του ID	75
5.1 Εξοπλισμός που χρησιμοποιήθηκε στην πρώτη εξωτερική πειραματική φάση	78
5.2 Χωρική τοποθέτηση του υπό ελέγχου συστήματος και αντικειμένου εκτίμησης θέσης	79
5.3 Αναπαράσταση των θέσεων στις οποίες έγιναν οι μετρήσεις του πειράματος	79
5.4 Το ψηφιακό λέιζερ μέτρησης απόστασης που χρησιμοποιήθηκε (Bosch GLM 40)	80
5.5 Υπολογισμός των γωνιών με χρήση εργαλείου μέτρησης γωνίας . .	80
5.6 Επεξεργαστική ισχύς του συστήματος	81

5.7	Ανάγκες μνήμης του συστήματος	82
5.8	Ενεργειακές απαιτήσεις συστήματος	82
5.9	Θερμοκρασίες συστήματος έχοντας σε χρήση το fan του breakout board	83
5.10	Χρόνος του object detection μέσα σε κάθε μεμονωμένη χρονική στιγμή χρήσης του συστήματος	83
5.11	Εξάρτηση του μεγέθους του αντικειμένου σε pixel με την απόσταση του από την κάμερα	84
5.12	Αντιληπτική ικανότητα μεμονωμένου κόμβου κατά την διάρκεια των δοκιμών, σε συνδυασμό με τρισδιάστατη απεικόνιση της μπάλας . . .	85
5.13	Εκτίμηση απόστασης για κάθε δευτερόλεπτο χρήσης του συστήματος	86
5.14	Εκτίμηση γωνιών στους άξονες x και y για κάθε δευτερόλεπτο χρήσης του συστήματος	86
5.15	Τρισδιάστατη απεικόνιση των μετρήσεων για 3 από τις 5 ευθείες . . .	87
5.16	Απεικόνιση των θέσεων από τις οποίες πραγματοποιήθηκαν λήψεις, καθώς και οι θέσεις του αντικειμένου	88
5.17	Γραφική αναπαράσταση της εκτίμησης θέσης - πραγματικής απόστασης	89
5.18	Αντιληπτική ικανότητα από κάθε γωνία λήψης για την θέση 2 . . .	90
5.19	Εκτίμηση θέσης του αντικειμένου.	91
5.20	Μέρος των εν λειτουργία nodes/topics, καθώς και απεικόνιση των πληροφοριών που αποστέλλονται στα topics τα 4 nodes, μαζί με την εκτίμηση της θέσης από τον master.	92
5.21	Παράδειγμα δύο εκ των τεσσάρων συχνοτήτων για την λειτουργία του led που επιλέχθηκαν.	93
5.22	Στιγμιότυπο της διαδικασίας πείραματος, κατά την διάρκεια καθορισμού του ID του αντικειμένου με βάση την συχνότητα που αναβοσβήνει το led	93
6.1	Προτεινόμενα drone για τα οποία μπορούν να εκτιμηθούν οι θέσεις .	96

Λίστα πινάκων

1.1	Κατηγοριοποίηση των UAVs με βάση την δομή τους	2
3.1	Ορισμοί ονομάτων των Nodes	22
3.2	Σύνοψη τεχνικών εκτίμησης απόστασης/γωνίας και υπολογισμού θέσης	38
3.3	Σύγχριση μεταξύ Centralized και Distributed	41
3.4	Σύγχριση μεταξύ Range-based και Range-free	42
5.1	Πραγματικές αποστάσεις και εκτιμήσεις για κάθε γωνία λήψης	89

Λίστα Αλγορίθμων

1	Multi Dimensional Scaling-Mobile Assisted Programming	41
2	Approximate Point in Triangle	42
3	Distance Vector-Hop	42
4	HSV (color based) ball detection	65
5	Hough circle (shape based) ball detection	67

Φυσικές Σταθερές

Ταχύτητα του φωτός στο κενό $c_0 = 2.997\,924\,58 \times 10^8 \text{ m s}^{-1}$ (αριθμός)
π, Σταθερά του Αρχιμήδη $\pi = 3.14159\,26535$ (προσέγγιση)

Λίστα συμβόλων

$\ \cdot \ $	ευκλείδεια απόσταση	
$ \cdot $	cardinality	
A	πλάτος	
d	απόσταση	m
f	συχνότητα	Hz
F	focal length	mm
P	ισχύς	$\text{W} (\text{J s}^{-1})$
r	ακτίνα	m
s	ταχύτητα	m s^{-1}
t	χρόνος	s
T	περίοδος	s
x, y, z	συντεταγμένες θέσης	
V	ταχύτητα	m s^{-1}
θ	γωνία	°
λ	μήκος κύματος	m
ϕ	φάση	°
ω	γωνιακή συχνότητα	rad

Κατάλογος ακρωνυμίων

2D	2 Dimensional
3D	3 Dimensional
AFOV	Angular Field Of View
AHD	Average Hop Distance
AoA	Angle of Arrival
APIT	Approximate Point In Triangle
BP	Belief Propagation
BS	Base Station
CIF	Covariance Intersection Filter
CL	Cooperative Localization
CPU	Central Processing Unit
CV	Computer Vision
DEM	Digital Elevation Model
DoA	Direction of Arrival
DoF	Degrees of Freedom
DV-Hop	Distance Vector-Hop
EKF	Extended Kalman Filter
ESC	Electronic Speed Control
FANET	Flying Ad-hoc Network
FOV	Field Of View
FPV	First Person View
FSPL	Free Space Path Loss
GCC-PHAT	Generalized Cross Correlation - Phase Transform
GNSS	Global Navigation Satellite System
GPDF	Gauss priori Probability Density Function
GPS	Global Positioning System
GPU	Graphics Processing Unit
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IMU	Intertial Measurement Unit
IoT	Internet of Things

IR	Infrared
ISR	Intelligence, Surveillance, and Reconnaissance
KF	Kalman Filter
LBSA	Localization Based on Simulated Annealing
LiDAR	Light Detection And Ranging
LoRaWAN	Long Range Wide Area Network
LoS	Line of Sight
LPN	Low Power Network
LPS	Local Positioning System
LPWAN	Low Power Wide Area Network
MCU	Micro Controller Unit
MDS-MAP	Multi Dimensional Scaling-Mobile Assisted Programming
MEMS	Micro-electro-mechanical Systems
ML	Maximum Likelihood
MLAT	Multilateration
MoCap	Motion Capture
MOOC	Massive Open Online Course
MPU	Micro Processor Unit
MRS	Minimal Required number of Sensors
NLoS	Non-Line of Sight
OpenCV	Open Source Computer Vision
OS	Operating System
PBR	Passive Bistatic Radar
PSL	Passive Source Location
RF	Radio Frequency
ROS	Robot Operating System
RSSI	Received Signal Strength Indicator
RTCM	Radio Technical Commission for Maritime Service
RTK	Real Time Kinematic
RTLS	Real Time Location System
RTT	Round-Trip Time
SCIF	Split Covariance Intersection Filter
SDP	Semi Definite Programming
SITL	Software In The Loop
SLAM	Simultaneous Localization and Mapping
SNR	Signal to Noise Ratio
SR	Swarm Robotics
S&R	Search & Rescue

SS-TWR	Single-Sided Two-Way Ranging
TDoA	Time Difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
UART	Universal Asynchronous Receiver-Transmitter
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UCAV	Unmanned Combat Aerial Vehicle
UKF	Unscented Kalman Filter
UWB	Ultra-wideband
VSLAM	Visual Simultaneous Localization and Mapping
VTOL	Vertically Hover, Take-off, and Land
WSN	Wireless Sensor Network

Σε όσους με βοήθησαν να είμαι ο άνθρωπος που είμαι
σήμερα. . .

Κεφάλαιο 1

Εισαγωγή

“Μεμονωμένοι κάνουμε τόσα λίγα, ως σύνολο κάνουμε πολλά”

Hellen Keller

Τα τελευταία χρόνια η ανάπτυξη της επιστήμης έχει επιφέρει την απόκτηση των τεχνολογικών επιτευγμάτων από το ευρύ κοινό^{*} με ένα πολύ οικονομικό αντίτιμο. Αυτό σημαίνει ότι ο καθένας πολύ εύκολα, μπορεί να έχει στην κατοχή του ακόμα και προϊόντα τα οποία θεωρούνται state-of-the-art, χωρίς να χρειάζεται να δαπανήσει μεγάλα ποσά. Το ίδιο φυσικά συμβαίνει και με τον κλάδο των drone και την - κατά επέκταση - χρήση αυτών^{*} ακόμα και για ψυχαγωγικό σκοπό.

Κατά το τέλος του έτους 2019 μόνο στις Ηνωμένες Πολιτείες της Αμερικής υπήρχαν πάνω από 990 χιλιάδες εγγεγραμμένοι χειριστές drone με πάνω από 1.32 εκατομμύρια drone ψυχαγωγικού χαρακτήρα να χρησιμοποιούνται [1]. Ενώ μέχρι το 2025 υπολογίζεται ότι το μέγεθος αγοράς των υπηρεσιών drone θα κοστολογείται στα 63.6 δισεκατομμύρια δολάρια [2].

Φυσικά η χρήση τους δεν περιορίζεται μόνο στην ψυχαγωγία, επαιρίες όπως η Amazon έχουν αποκτήσει ήδη τα απαραίτητα πιστοποιητικά και εγκρίσεις, με σκοπό να αξιοποιήσουν drone για παράδοση των δεμάτων αρκετά σύντομα [3], καθώς προς το παρόν η διαδικασία βρίσκεται σε στάδιο δοκιμών. Συνεπώς, είναι εύκολο να κατανοήσει ότι ο συγκεκριμένος κλάδος πρόκειται να έχει ακόμα μεγαλύτερη άνθιση, με αρκετά μεγάλο ερευνητικό ενδιαφέρον να του αναλογεί.

Με την αύξηση των drone και την αύξηση των εφαρμογών, υπάρχει η ανάγκη συνεργασίας και η δημιουργία drone swarms για την επιτυχή ολοκλήρωση των στόχων που έχουν οριστεί. Όμως για να καταφέρουν τα drone να συνεργαστούν, χρειάζεται πρώτα να μπορούν να ξεπεράσουν τα προβλήματα τα οποία υπάρχουν. Με ένα από τα μεγαλύτερα να είναι η πλήρη γνώση της θέσης τους στον χώρο.

1.1 Μη επανδρωμένα οχήματα

Είναι σημαντικό από τα πρώτα βήματα, να έχει γίνει κατανοητό με τον όρο drone σε τι παραπέμπουμε - όπως επίσης πότε θεωρείται ότι ένα συμήνος από drone πετάει σε σχηματισμό (drone swarm).

1.1.1 Ιστορία και τύποι drone

Όταν αναφερόμαστε στον όρο Unmanned Aerial Vehicle (**UAV**) ή απλούστερα drone κάνουμε αναφορά για ένα μη επανδρωμένο ιπτάμενο αεροσκάφος το οποίο ελέγχεται είτε απομακρυσμένα από έναν άνθρωπο, είτε είναι τελείως αυτόνομο. Τα **UAV** μαζί με ένα Base Station (**BS**) και την από κοινού επικοινωνίας του σταθμού - drone, δημιουργούν αυτό που ονομάζουμε Unmanned Aircraft System (**UAS**) [4] [5].

Η πρώτη εμφάνιση των **UAV** έγινε κατά το 1849 στα πλαίσια μάχης, ενώ οι πρώτες καινοτομίες πάνω σε αυτά ξεκίνησαν ήδη από τις αρχές του 20ου αιώνα. Το 2013 τουλάχιστον 50 χώρες χρησιμοποιούσαν **UAVs** για κάποιον σκοπό, με μερικές από αυτές φυσικά να σχεδιάζουν τα δικά τους [5]. Αυτήν την στιγμή υπάρχουν πάνω από 1000 διαφορετικά μοντέλα **UAV** που χρησιμοποιούνται ανά τον κόσμο, με τα περισσότερα από αυτά να μην έχουν ψυχαγωγικό χαρακτήρα [6].

Είναι λοιπόν ξεκάθαρο ότι το πλήθος των drone είναι τόσο μεγάλο, λόγω των διαφορετικών αναγκών - και ότι κάποια έχουν καλύτερα αποτελέσματα από ότι άλλα σε συγκεκριμένες αποστολές. Για αυτό, έχουν γίνει ήδη προσπάθειες για την κατηγοριοποίηση των **UAVs** σύμφωνα με τα διάφορα χαρακτηριστικά που μπορεί να έχουν. Ενδεικτικά με βάση το μέγεθος, την αυτονομία, το βάρος ή το μηχανολογικό σχεδιασμό των **UAV** να είναι μερικές από τις υπάρχουσες [4] [7] [8]. Στο Πίνακα 1.1 υπάρχει μία απλουστευμένη κατηγοριοποίηση η οποία προτάθηκε από τους συγγραφείς του [8] σύμφωνα με τη βασική μηχανολογική δομή που μπορεί να έχει ένα drone καθώς και τα πλεονεκτήματα της κάθε δομής.

ΠΙΝΑΚΑΣ 1.1: Κατηγοριοποίηση των **UAVs** με βάση την δομή τους [8].

Drones	Main features
Fixed-Wing	μεγάλη αυτονομία και γρήγορες ταχύτητες πτήσης
Fixed-Wing Hybrid	VTOL και μεγάλη αυτονομία πτήσης
Single Rotor	VTOL, hover, και μεγάλη αυτονομία πτήσης
Multirotor	VTOL, hover, και μικρή αυτονομία πτήσης



ΕΙΚΟΝΑ 1.1: Παραδείγματα των UAV

Τυπικά τα Fixed-Wing drones είναι αρκετά ακριβά, χρειάζονται εξειδικευμένους χειριστές για να λειτουργήσουν, όπως επιπλέον και περισσότερο χώρο για την απογείωση και την προσγείωση. Είναι ιδανικά για εφαρμογές που χρειάζεται να καλύψουμε μεγάλες περιοχές και συχνά έχουν αυτονομία τουλάχιστον μερικών ωρών. Για αυτούς τους λόγους χρησιμοποιούνται κυρίως από κυβερνήσεις, στρατιωτικές μονάδες ή επιχειρήσεις για την γρήγορη επίβλεψη μεγάλων εκτάσεων [9].

Τα Fixed-Wing Hybrid προσπαθούν να λύσουν τα μειονεκτήματα που έχουν τα Fixed-Wing drones, την μη ικανότητα δηλαδή για Vertically Hover, Take-off, and Land (VTOL) όμως είναι ακόμα σε αρχικά στάδια [8].

Τα Single Rotor είναι επίσης αρκετά ακριβά, πολύπλοκα μηχανολογικά μηχανήματα, που δέχονται πολλούς κραδασμούς, απαιτούν εξειδικευμένους χειριστές όμως μπορούν να μεταφέρουν αρκετά βαριά payloads, θετικό στην χρήση τους ότι μπορούν να πραγματοποιήσουν VTOL [8].

Τα Multirotor είναι ίσως τα πιο ευρέως διαδεδομένα. Καθώς είναι τα πιο οικονομικά από τα παραπάνω και εύκολο να κατασκευαστούν. Πραγματοποιούν επίσης VTOL, ενώ μπορούν να βρεθούν στο εμπόριο με διάφορο πλήθος από έλικες και είναι το

κύριο είδος που χρησιμοποιείται από ερασιτέχνες ή χομπίστες για λόγους αναψυχής [8].

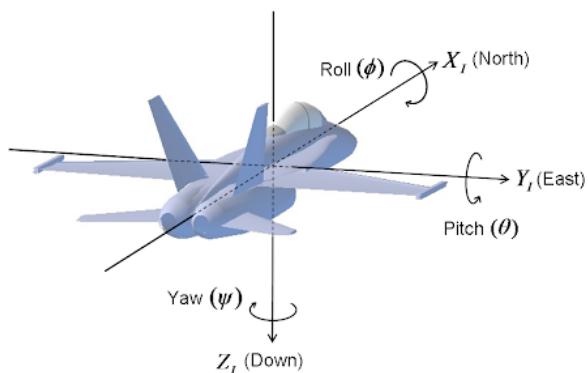
Στην Εικόνα 1.1 δίνονται κάποια ενδεικτικά παραδείγματα UAVs με βάση την κατηγοριοποίηση του Πίνακα 1.1. Φυσικά αυτή η κατηγοριοποίηση δεν περιλαμβάνει όλα τα είδη drone, είναι όμως ικανοποιητική για να γίνουν ζεκάθαρες δύο βασικές ιδέες. Αρχικά ανάλογα με την εφαρμογή που μας ενδιαφέρει, θα πρέπει να επιλέξουμε την χρήση του πλέον κατάλληλου τύπου drone. Όπως επίσης με βάση την επιλογή του συγκεκριμένου τύπου - αυτόματα έχουμε να διαχειριστούμε τα πλεονεκτήματα ή τα μειονεκτήματα που έχει.

Αυτά είναι και τα δύο σημεία στα οποία θα πρέπει να δοθεί ιδιαίτερη προσοχή στον σχεδιασμό του συστήματος που θα αναλυθεί στα επόμενα κεφάλαια.

Σε περίπτωση που μας ενδιαφέρει, οι συγγραφείς του [7] παρουσιάζουν με εκτενέστερο τρόπο, διάφορες κατηγοριοποιήσεις και είδη drone τα οποία δεν εμπίπτουν στα πλαίσια αυτής της διπλωματικής και κυμαίνονται από smart dust, bio-drones, hybrid drones και άλλα πολλά.

1.1.2 Χαρακτηριστικά

Σε όποια από τις κατηγορίες και αν αντιστοιχεί ένα drone από την στιγμή που είναι ένα ιπτάμενο αντικείμενο, θα πρέπει να έχει την δυνατότητα να κινείται - φυσικά - στον αέρα. Στην Εικόνα 1.2 παρουσιάζονται στους 3 άξονες, τα 6 Degrees of Freedom (DoF) - τόσο Transitional όσο και Rotational - ενός UAV [10] καθώς και το όνομα που δίνεται στην κίνηση ανάλογα με τον άξονα που πραγματοποιείται.



ΕΙΚΟΝΑ 1.2: Κυρίαρχοι άξονες αναφοράς (URL)

Ζούμε σε μία ψηφιακή εποχή, στην οποία ένα από τα πιο σημαντικά κατορθώματα είναι η ανάπτυξη των Integrated Circuits ([ICs](#)) & Micro-electro-mechanical Systems ([MEMS](#)) [11], πράγμα το οποίο έχει βοηθήσει - έξυπνες συσκευές γεμάτες με αισθητήρες να βρίσκονται γύρω μας. Τέτοια τεχνολογικά επιτεύγματα όπως τα drones είναι συνεπώς εξοπλισμένα με Micro Controller Units ([MCU](#)) [12] ή Micro Processor Units ([MPU](#)) [13] για τον έλεγχο τους, ενώ δεν θα μπορούσαν να μην περιλαμβάνουν πλέον και ένα μεγάλο πλήθος και εύρος διαφορετικών τύπων sensors on-board. Με δύο από τους πιο σημαντικούς να είναι το Electronic Speed Control ([ESC](#)) [14] και το Inertial Measurement Unit ([IMU](#)) [15] - τα οποία χρησιμοποιούνται ώστε το drone να μπορεί να διατηρεί μία σταθερή και ελεγχόμενη πτήση [8]. Εκτός από αυτούς βέβαια, ένα drone μπορεί να διαθέτει Global Positioning System ([GPS](#)), camera για λήψη οπτικού υλικού ή ελέγχου μέσω First Person View ([FPV](#)), Obstacle avoidance sensors ή ακόμα και άλλους. Με κύριο περιορισμό[•] τα αισθητήρια όργανα να βρίσκονται στο εύρος βάρος του payload, που μπορεί να μεταφέρει το drone.

Σε αυτό το section πραγματοποιήθηκε μία πρώτη οριοθέτηση του όρου drone, παρόλα αυτά δεν αναφέρθηκαν λόγοι ύπαρξης τους, καθώς και εφαρμογές τους. Η ύπαρξης των swarms είναι ουσιαστικά η κάλυψη των αναγκών από των individual drones σε μεγαλύτερη κλίμακα, για αυτό[•] μερικές από τις εφαρμογές των drone - βρίσκονται στη [Παράγραφο 1.2.2](#).

1.2 Κίνητρο

Ξεκινώντας από τα individual drones, αξίζει να σημειωθεί ότι το [Πολυτεχνείο Κρήτης](#) έχει ένα ενεργό ερευνητικό ιστορικό στον τομέα των αεροχημάτων. Το SenseLab στο οποίο υπεύθυνος καθηγητής είναι ο κύριος Παρτσινέβελος της Σχολής Μηχανικών Ορυκτών Πόρων να είναι ένα από αυτά - και μάλιστα με πολλαπλές διαχρίσεις σε διεθνείς διαγωνισμούς [16] [17].

1.2.1 Φύση

Στην φύση είναι αρκετά συχνή η ύπαρξη έμβιων οργανισμός που κινούνται σε ομάδες, με μερικά από αυτά ως παράδειγμα, τα συμήνη από πουλιά και έντομα, κοπάδια ψαριών ή και αγέλες ζώων. Σκοπός της συνεργασίας τους είναι κυρίως η προστασία από θηρευτές ή άλλοι σχετικοί με την επιβίωση. Αρκετές έρευνες έχουν γίνει εμπνευσμένες σε αυτό, ακόμα και στον τομέα των fleet of drones [18] [19] [20].

1.2.2 Συμήνη και εφαρμογές

Από μόνο του ένα UAV σε πολλές περιπτώσεις θα μπορέσει να φέρει εις πέραν την αποστολή που του έχει ανατεθεί. Πολύ εύκολα όμως μπορούν να δημιουργηθούν ζητήματα αξιοπιστίας, όταν ένα μεμονωμένο drone χρειάζεται ως παράδειγμα να χαρτογραφήσει σε μικρό χρονικό διάστημα ένα άγνωστο και μεγάλης έκτασης περιβάλλον. Όπως επίσης, εάν θα θέλαμε να διαθέτει πολλαπλούς αισθητήρες για να έχουμε πιο λεπτομερείς δεδομένα σε μία αποστολή, όμως αυτό να είναι αδύνατον διότι υπερβαίνουν το δυνατόν payload που μπορεί να μεταφέρει το drone. Η ακόμα, και για το ενδεχόμενο αποτυχίας της αποστολής, σε κάποιο πιθανό malfunction που θα πραγματοποιηθεί στο ιπτάμενου όχημα. Μπορούμε να πούμε λοιπόν σε μία abstract σύγκριση με αυτή των ζώων, ότι ωθούμαστε για ανάλογους σκοπούς στην χρήση των swarms ώστε να ξεπεράσουμε αυτά τα προβλήματα [21].

Με τον όρο Swarm Robotics (SR) συχνά αναφερόμαστε στην μεθοδολογία που ακολουθούμε ώστε να συντονίζουμε πολλαπλές ανεξάρτητες ρομποτικές οντότητες να συμπεριφέρονται συνεργατικά* ως ένα ενιαίο σύστημα [22]. Με τον όρο συνεργατικά, αυτό σημαίνει είτε να μπορούν να κινηθούν ως ομάδα [23] [24] [25], είτε να επικοινωνούν ώστε γνώση πληροφορίας την οποία έχει συλλέξει ένα από αυτά να μεταδίδεται και στα υπόλοιπα (π.χ. σε περίπτωση που θέλουμε να κάνουμε 3D reconstruction μίας περιοχής η γνώση της θέσης από την οποία λαμβάνεται - με χρήση της κάμερας - υλικό είναι σημαντική για την δημιουργία του ψηφιακού μοντέλου [26]).

Σήμερα τα drones καθώς και τα fleet of drones χρησιμοποιούνται σε ένα μεγάλος εύρος εφαρμογών. Έχει γίνει αναφορά είδη από την αρχή του chapter ότι χρησιμοποιούνται για λόγους αναψυχής, με μερικά παραδείγματα, την διεκπεραίωση shows [27] [28] [29] και την καταγραφή οπτικού υλικού για δημιουργία ταινιών [30]. Άλλες πιο ζωτικής σημασίας χρήσεις τους, είναι σχετικές με environment mapping [31], police surveillance [32], natural disaster inspection [33], Search & Rescue (S&R) [34], light cargo transportation [35] και πολλές άλλες. Ακόμα και στρατιωτικές επιχειρήσεις χρησιμοποιούν drones. Συνήθως σε αυτές τις περιπτώσεις αναφερόμαστε στην χρήση Unmanned Combat Aerial Vehicle (UCAV) [36] [37] και πολλές από τις εφαρμογές στο συγκεκριμένο τομέα έχουν να κάνουν με Intelligence Surveillance, and Reconnaissance (ISR) [38].

Για περισσότερες εφαρμογές των drones/swarms [39] ή λεπτομέρειες σχετικά με αυτές, μπορούμε να ανατρέξουμε στην υφιστάμενη βιβλιογραφία [40] [41] [42].

1.3 Ερευνητικοί στόχοι και συνεισφορά

Όπως μπορεί να γίνει εύκολα αντιληπτό· η γνώση της σχετικής θέσης ενός UAV σε σχέση με τα υπόλοιπα του σμήνους, είτε σε συνδυασμό αυτής σε απόλυτο τρόπο με βάση ένα καθορισμένο σύστημα αξόνων, είτε ακόμα και ο υπολογισμός της θέσης επιπλέον αντικειμένων - ενδιάμεσα του σμήνους - τα οποία σχετίζονται με την εκάστοτε αποστολή, είναι αρκετά σημαντική για ένα πολύ μεγάλο πλήθος εφαρμογών. Σκοπός της συγκεκριμένης διπλωματικής εργασίας, είναι να γίνει η απαραίτητη ερευνητική αναζήτηση, ώστε να επιτευχθεί ο σχεδιασμός καθώς και η υλοποίηση ενός ενσωματωμένου συστήματος - το οποίο να είναι σε θέση με όσο το δυνατόν πιο οικονομικό τρόπο - να υπολογίζει την θέση αντικειμένων από drone όταν αυτά πετούν σε σχηματισμό, το ακριβές πρόβλημα το οποίο θα γίνει προσπάθεια να επιλυθεί καθώς και η μεθοδολογία, παρουσιάζονται αναλυτικά στο Chapter 3.3.

1.4 Περιεχόμενα Διπλωματικής

- **Κεφάλαιο 2 - Προϋπάρχουσες προσεγγίσεις:** Στην συγκεκριμένη ενότητα γίνεται μία συνοπτική αναφορά σε τεχνικές που χρησιμοποιούνται ήδη, σύμφωνα με την βιβλιογραφία, για την εύρεσης θέσης σχετιζόμενα με drone.
- **Κεφάλαιο 3 - Θεωρητικό Υπόβαθρο:** Σε αυτό το κεφάλαιο παρουσιάζεται το απαραίτητο μαθηματικό υπόβαθρο, καθώς και βασικές μεθοδολογίες εύρεσης τοποθεσία, που προέρχονται από ευρύτερους τομείς, όπως των Wireless Sensor Networks (WSN).
- **Κεφάλαιο 4 - Διαδικασία Σχεδίασης και Υλοποίησης:** Σε αυτό το σημείο παρουσιάζεται η διαδικασία σχεδιασμού του ενσωματωμένου συστήματος που σχετίζεται αυτή η διπλωματική.
- **Κεφάλαιο 5 - Επαλήθευση Λειτουργίας και Αποτελέσματα:** Στο παρόν κεφάλαιο παρουσιάζονται οι απαραίτητοι έλεγχοι που έγιναν για την επαλήθευση της οριζόντιας λειτουργίας του συστήματος.
- **Κεφάλαιο 6 - Τελικά Συμπεράσματα και Μελλοντικές Εξελίξεις:** Στο συγκεκριμένο κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα τα οποία βγήκαν από το σύνολο της διπλωματικής - καθώς και κάποιες από τις πιθανές μελλοντικές εξελίξεις της.

Κεφάλαιο 2

Προϋπάρχουσες προσεγγίσεις

“Αν σήμερα η τεχνολογία
βρίσκεται σε αυτό το επίπεδο,
φαντάσου τι θα μπορούμε να
καταφέρουμε στο μέλλον”

Timothy Chung

Σε αυτό το *Chapter* περιγράφονται τρόποι με τους οποίους υπάρχουσες εφαρμογές UAV swarms, καθώς και μεμονωμένων drone - των οποίων οι βασικές αρχές μπορούν να χρησιμοποιηθούν στα συμήνη - επιλύουν το localization problem. Κάποια από τα συστήματα στις παρακάτω έρευνες έχουν καθαρά θεωρητική πλευρά, ενώ, άλλα έχουν δοκιμαστεί σε real-life scenarios. Τέλος, σε αρκετές από αυτές τις εφαρμογές χρησιμοποιούνται τεχνικές που θα αναλυθούν στο *Κεφάλαιο 3* σε μεγαλύτερο βάθος, καθώς θα αποτελέσουν το προαπαιτούμενο θεωρητικό και μαθηματικό υπόβαθρο για τον σχεδιασμό του συστήματος της παρούσας διπλωματικής.

Το πρόβλημα του Local Positioning System (**LPS**) [43] έχει ερευνηθεί από διάφορες οπτικές· οι οποίες μπορεί να βασίζονται σε Radio Frequency (**RF**), Sound Waves ή ακόμα και να είναι Optical Oriented.

Σχεδόν βέβαιο είναι επίσης, ότι από όποια κατεύθυνση και αν προσεγγίσουμε το **LPS** να έχουμε την ανάγκη να χρησιμοποιήσουμε κάποια τεχνική για να συνενώσουμε όλες τις πληροφορίες που έχουμε από τους διάφορους αισθητήρες - για κάθε μεμονωμένο time-frame - με δεδομένο ότι κάνουμε χρήση πολλαπλών αισθητήρων ώστε να μειώσουμε την εντροπία, συνεπώς και την αβεβαιότητα ως προς την εκτίμηση της θέσης. Την ανάγκη αυτού έρχεται να καλύψει η έννοια του *sensor fusion* [44]. Όπως επίσης, και την ανάγκη φιλτραρίσματος των μετρήσεων - που οδηγεί σε καλύτερα αποτελέσματα - με μερικούς γνωστούς αλγόριθμους που να το επιτυγχάνουν· να είναι ο Extended Kalman Filter (**EKF**), Unscented Kalman Filter (**UKF**), Covariance Intersektion

Filter (**CIF**), Split Covariance Intersection Filter (**SCIF**) και Belief Propagation (**BP**) [45].

2.1 Ραδιοσυχνότητες

Το πιο εύκολο που μπορούμε να φανταστούμε είναι να χρησιμοποιήσουμε **RF** τεχνολογίες όπως **GPS**, WiFi, Zigbee, Bluetooth, Ultra-Wideband (**UWB**) ή ακόμα και άλλα* για την επικοινωνία και την υλοποίηση του **LPS**. Συχνός παράγοντας για το ποιο θα επιλεχθεί, σχετίζεται αν η εφαρμογή μας κυμαίνεται σε short ή long range εμβέλειες.

Άλλοι παράγοντες, ως παράδειγμα, είναι ότι το WiFi είναι εύκολα προσβάσιμο με μικρό κόστος, ενώ το Zigbee προτιμάται για low power consumption εφαρμογές. Όμως προβλήματα, μπορεί να είναι, ότι στο WiFi είναι πολύ εύκολο να υπάρχουν παρεμβολές από τις υπόλοιπες γειτονικές συσκευές¹ άρα δεν το καθιστά καλή λύση για κάθε αποστολή ενός swarm. Το Zigbee επιλύει γενικά κάποιες από τις αδυναμίες του WiFi² όμως ταυτόχρονα πρέπει να αποδεχτούμε το μικρό data rate που μπορούμε να μεταδώσουμε. Συχνή εφαρμογή έχουν επίσης τα **UWB** καθώς επιλύουν θέματα σχετικά με την ακρίβεια των μετρήσεων, το κόστος καθώς και την εμβέλεια* ταυτόχρονα [46].

Τέλος, θετικό στην χρήση **RF** τεχνολογιών, είναι αρχικά ότι δεν χρειαζόμαστε άμεση οπτική επαφή με το ιπτάμενο, ενώ επίσης ότι μπορούμε εύκολα να δημιουργήσουμε ένα Flying Ad-hoc Network (**FANET**) το οποίο βοηθάει στην ευελιξία του συστήματος.

2.1.1 WiFi

Μία αρχική προσέγγιση για τον εντοπισμό ενός drone δίνεται από τους συγγραφείς του [47] οι οποίοι χρησιμοποιούν WiFi προκειμένου να εντοπίσουν την θέση του αεροχήματος. Το ενδιαφέρον - με την χρήση του WiFi - είναι ότι, είναι μία εύκολη λύση όταν αναφερόμαστε για short range localization.

Ο τρόπος με τον οποίο προσέγγιζουν το πρόβλημα, είναι με το να προσπαθήσουν να οξιοποιήσουν - από την πλευρά του WiFi based - τις τεχνικές, Passive Bistatic Radar (**PBR**) και Passive Source Location (**PSL**), να χρησιμοποιήσουν Kalman Filter (**KF**) και τέλος να πραγματοποιήσουν Data Fusion τεχνικές στους δύο αισθητήρες που χρησιμοποιήθηκαν* ώστε να γίνει ο εντοπισμός της θέσης.

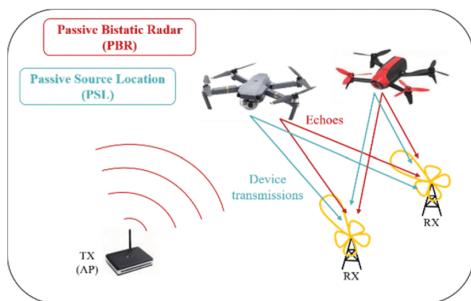
¹Αυτό μπορεί να είναι αρνητικό για κάποιες εφαρμογές σε indoor scenarios

²Σε indoor σενάρια παραμένει η δυσκολία για collision-free formation

Ξεκινώντας με τη τεχνική **PBR**, η οποία αξιοποιεί τα echoes που γίνονται scatter από τα εμπόδια που υπάρχουν στον χώρο^{*} προκειμένου να εντοπίσει το αντικείμενο. Έχει σαν κύρια χαρακτηριστικά ότι χρησιμοποιείται για εντοπισμό κινούμενων αντικειμένων και δεν χρειάζεται το ίδιο το αντικείμενο να εκπέμπει σήματα, πράγμα που το καθιστά ιδανικό για εντοπισμό non-cooperative στόχων. Προκειμένου βέβαια να εκτιμηθεί η θέση, έπειτα από την αρχική επεξεργασία που γίνεται από τα λαμβανόμενα echoes, ο εντοπισμός τελικά της θέσης πραγματοποιείται μέσω ενός συνδυασμού από μετρήσεις με βάση range είτε γωνία.

Αντίθετα, με την τεχνική **PSL** - η οποία προαπαιτεί stationary αντικείμενα - είναι σημαντικό να λαμβάνουμε πακέτα από το αντικείμενο ενδιαφέροντος, καθώς εκτιμάει την θέση του^{*} μέσω των πακέτων που λαμβάνει, σε συνδυασμό με μία μέτρηση **AoA** και μία **TDoA** - όπου στη *Παράγραφο 3.1* γίνεται ανάλυση τους.

Συνοπτικά οι συγγραφείς της συγκεκριμένης έρευνας, παρουσιάζουν στην *Εικόνα 2.1* (a) τον γενικό τρόπο λειτουργίας των δύο μεθόδων, ενώ στην *Εικόνα 2.1* (b) τις διαφορές τους.



(a) PBR και PSL (URL)

PASSIVE BISTATIC RADAR (PBR)	PASSIVE SOURCE LOCATION (PSL)
Device-free localization	Device-based localization
Higher computational cost	Lower computational cost
Effective for moving targets	Potentially inaccurate for moving targets
No detection of stationary targets	Stationary targets can be detected and localized
Closely spaced targets cannot be discriminated	Closely spaced targets can be discriminated thanks to the device MAC Address

(b) Χαρακτηριστικά των PBR και PSL

(URL)

ΕΙΚΟΝΑ 2.1: PBR και PSL προσεγγίσεις εντοπισμού βάση του [47]

Για την υλοποίηση χρησιμοποιήθηκε ένα DJI Mavic Pro, το οποίο ήταν το drone του οποίου η θέση έγινε προσπάθεια να εκτιμηθεί. Σχετικά με το σκέλος του **PBR** χρησιμοποιήθηκε ένα Access Point (AP, D-Link DAP1160), το οποίο ήταν συνδεδεμένο σε μία transmitting directive antenna (TP-LINK TL-ANT2409A). Ενώ για τα receivers χρησιμοποιήθηκαν οι ίδιες κεραίες σε συνδυασμό με ένα URSP-2955 της National Instruments.

2.1.2 UWB

Όπως αναφέρθηκε στην αρχή αυτού του κεφαλαίου, ένας τρόπος προσδιορισμού της θέσης^{*} είναι με χρήση **UWB**. Ένα από τα πλεονεκτήματα της χρήσης **UWB** είναι

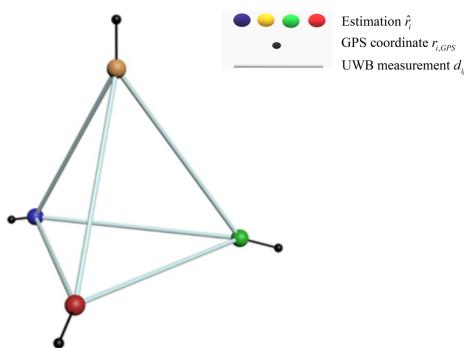
η ικανότητα υπολογισμού της σχετικής απόστασης μεταξύ δύο σημείων με σφάλμα απόκλισης μικρότερο των 0.1m [48].

Συγκεκριμένα, οι συγγραφείς του [49] πραγματοποιούν Sensor Fusion, βασισμένα σε μετρήσεις GPS/IMU/UWB προκειμένου να εκτιμήσουν την θέση 7 Fixed-Wing UAVs σε εξωτερικό χώρο.

Ο τρόπος με τον οποίο το επιτυγχάνουν, συνοπτικά, είναι ο εξής. Κάθε drone είναι σε θέση να συλλέξει για τον εαυτό του - μέσω των αισθητήρων - πληροφορίες όπως τις συντεταγμένες του, το διανύσματα της ταχύτητας του, της επιτάχυνσης του και τέλος μέσω UWB· και χρήση μίας τεχνικής που ονομάζεται Single-sided Two-way Ranging (SS-TWR) - ουσιαστικά μία ToA-based προσέγγιση (η οποία γενικά αναφέρεται στη Παράγραφο 3.1.1) - να υπολογίσει την απόσταση που βρίσκεται σε σχέση με κάθε γειτονικό ιπτάμενο.

Αφού συλλεχθούν οι παραπάνω πληροφορίες, γίνεται το απαραίτητο preprocessing, το οποίο περιλαμβάνει την πραγματοποίηση μετασχηματισμών και φιλτραρίσματος με χρήση EKF, ώστε να δημιουργηθεί μία καλύτερη εκτίμηση της κάθε απόστασης με τα γειτονικά drones.

Στην συγκεκριμένη έρευνα, μοντελοποιούν το σύστημα ως ένα time-varying undirected γράφο τον οποίο τον διαχειρίζονται ως ένα spring system - όπως παρουσιάζεται στην Εικόνα 2.2 - με διαφορετικούς συντελεστές σκληρότητας και προσπαθούν να βρουν το σημείο ισορροπίας του, που τελικά είναι το σημείο με την ελάχιστη συνολική δυναμική ενέργεια. Ουσιαστικά μετατρέπουν το position estimation problem σε ένα non-convex optimization problem.



ΕΙΚΟΝΑ 2.2: Αναπαράσταση του spring system βασισμένο στο [49]
(URL)

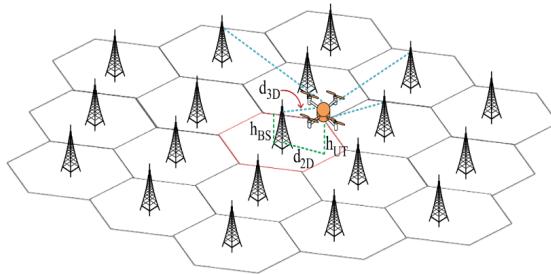
Σχετικά με το υλικό που χρησιμοποιήθηκε, έγινε χρήση ενός NEO-M8N GPS της U-blox με συχνότητα ανανέωσης τα 5Hz. Σε κάθε UAV χρησιμοποιήθηκε ένα Pixhawk

PX4 flight management unit. Για το κομμάτι του **UWB** έγινε χρήση ενός module LinkTrack, για την επικοινωνία με το ground station χρησιμοποιήθηκε ένα UBNT, ενώ η επεξεργασία των αλγορίθμων έγιναν σε ένα Raspberry Pi 3B+.

Τέλος, τα αποτελέσματα που κατάφεραν να έχουν ήταν, **3D** relative position estimation με σφάλμα απόκλισης τα 0.4m.

2.1.3 Κυψελωτά δίκτυα

Ένας ακόμα εφικτός τρόπος προσέγγισης - για την εκτίμηση της θέσης - είναι με χρήση cellular networks. Η εξέλιξη των mobile networks (GSM/LTE/5G) έχει οδηγήσει να γίνουν προσπάθειες εκτίμησης της θέσης των drones με την βοήθεια αυτών των τεχνολογιών, παράδειγμα της μεθόδου βρίσκεται στην *Eikóna 2.3*.



EIKONA 2.3: Κυψελωτά δίκτυα βασισμένο στο [50] ([URL](#))

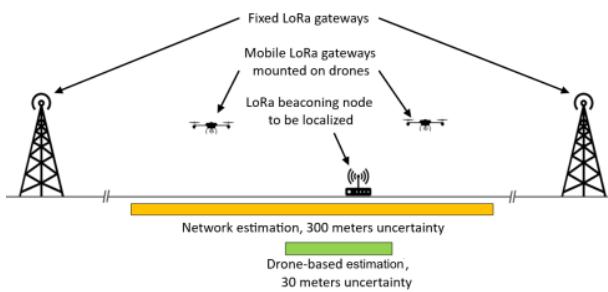
Συγκεκριμένα, οι συγγραφείς του [50] ερευνούν από την θεωρητική οπτική το συγκεκριμένο πρόβλημα, και αφού το μοντελοποιήσουν^{*} χρησιμοποιούν Monte Carlo simulation και το snapshot model, προκειμένου να αναλύσουν την επιρροή των διάφορων παραμέτρων του συστήματος στην τελική απόδοση διάφορων localization τεχνικών (όπως με χρήση **TDoA**). Τέτοιοι παράμετροι - που υπάρχουν σε αυτά τα δίκτυα - είναι το πλήθος των χεραιών με τις οποίες μπορούμε να επικοινωνήσουμε και το πόσο συγχρονισμένες είναι, τα χαρακτηριστικά του air-to-ground καναλιού καιώς και το αν υπάρχει ενδιάμεση ύπαρξη εμποδίων σε καθένα από αυτά. Όπως, και το ύψος του αεροχήματος σε συνδυασμό με τις λαμβανόμενες παρεμβολές.

2.1.4 Lora

Ο τομέας των Internet of Things (IoT) έχει επιφέρει στην έντονη χρήση^{*} για πολλές τεχνολογίες, με τα Long Range Wide Area Network (**LoRaWAN**) - που πρακτικά αποτελούν παρακλάδι του ευρύτερου τομέα των Low Power Wide Area Network (**LPWAN**) - να είναι μία από αυτές. Λόγοι που χρησιμοποιούνται συχνά - για τέτοιες

εφαρμογές - είναι ότι τα Low Power Networks (LPN) είναι ιδανικά όταν μας ενδιαφέρει το power consumption ενός συστήματος, καθώς επίσης και η μεγάλη εμβέλεια των LoRaWAN, που μπορούν να φτάσουν σε ακτίνες επικοινωνίας ακόμα και 2km σε κατοικημένες περιοχές εξυπηρετώντας χιλιάδες κόμβους σε ένα σύστημα [51].

Οι συγγραφείς του [51] αξιοποιούν τα πλεονεκτήματα αυτά, και τα χρησιμοποιούν - παρόλο που δεν είναι σχεδιασμένα για αυτό το λόγο - επεκτείνοντας ένα σύστημα από στατικούς κόμβους, σε ένα και με δυναμικές μετρήσεις* με την βοήθεια drone σαν 3D mobile gateway stations - *Εικόνα 2.4*, όπου καταφέρνουν να βελτιώσουν έως και 10 φορές την ακρίβεια εκτίμησης της θέσης ενός αντικειμένου.



ΕΙΚΟΝΑ 2.4: Σύνθεση συστήματος Lora βασισμένο στο [51] ([URL](#))

Για να το καταφέρουν, αξιοποιούν πληροφορία σχετικά με την ένταση της ισχύς των λαμβανόμενων πακέτων, μοντελοποιούν αυτήν την σχέση για να καταφέρουν να εκτιμήσουν την απόσταση μεταξύ δύο κόμβων και στην συνέχεια με χρήση αυτών των αποστάσεων, καταφέρνουν να βρουν την θέση του αντικειμένου (στο *Κεφάλαιο 3* υπάρχει ανάλυση των μεθόδων που ακολουθήσαν).

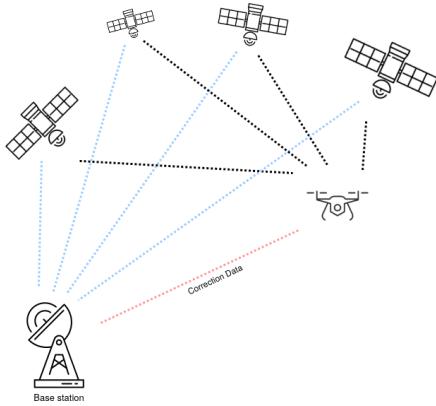
Η έρευνα τους περιστρέφεται στο develop και testing ενός search algorithm, τόσο από την πλευρά των simulation καθώς και ως real system, καταλήγοντας σε διάφορα συμπεράσματα - με βάση την χρήση διαφορετικού πλήθος drone σε συνδυασμό με διαφορετικών συχνοτήτων λήψεις μετρήσεων.

Σχετικά με τις τεχνολογίες που χρησιμοποιήθηκαν, ο search αλγόριθμος αρχικά δοκιμάστηκε σε MATLAB simulations ενώ στην συνέχεια σε Software In The Loop (SITL) simulations χρησιμοποιώντας το Gazebo. Τέλος, στο κομμάτι του hardware, το κάθε drone ήταν εξοπλισμένο με ένα PixHawk flight το οποίο έτρεχε PX4 autopilot, ένα Raspberry Pi 3B+ στο οποίο υπήρχε το Robot Operating System (ROS) και το MAVROS, ένα MultiTech MultiConnect Conduit LoRa gateway και η σύνδεση στο Swisscom LoRaWAN έγινε με την βοήθεια ενός Huawei E3372 USB modem.

2.1.5 RTK GNSS

Ένας εύκολα προσβάσιμος τρόπος - για την εκτίμηση θέσης αντικειμένου - που ο καθένας εύκολα μπορεί να χρησιμοποιήσει, σχετίζεται με τους δορυφόρους και τη χρήση GNSS τεχνικών όπως το GPS.

Παρόλο, την πολλά υποσχόμενη, εξέλιξη της ακρίβειας από ~5m [52] σε ~30cm [53] - για μη στρατιωτική χρήση - η οποία όμως επίσης δεν είναι αρκετή για τις ανάγκες ορισμένων εφαρμογών. Πολλές φορές οδηγούμαστε να κινηθούμε σε αρκετά ακριβές λύσεις διορθώσεις των σφαλμάτων, και να χρησιμοποιήσουμε Real Time Kinematic (RTK) [54] τεχνικές, οι οποίες είναι ικανές να προσφέρουν ακρίβεια 1-2cm σε ακτίνες ~20km.



EIKONA 2.5: Αρχή λειτουργίας RTK

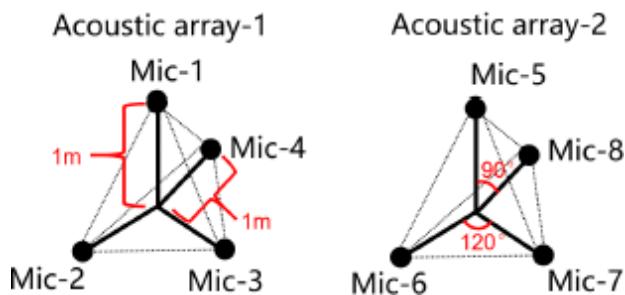
Η γενική αρχή λειτουργίας του RTK-GPS, είναι ότι αποτελείται από ένα επίγειο base station - ουσιαστικά μιλάμε για μία stationary GPS master antenna - του οποίου οι συντεταγμένες είναι γνωστές εξ' αρχής. Αυτός είναι υπεύθυνος να συγχρίνει τα carrier phase των λαμβανόμενων σημάτων από τους δορυφόρους προκειμένου να κάνει τις απαραίτητες διορθώσεις, τις οποίες αποστέλλει στην συνέχεια - μέσω του πρωτοκόλλου Radio Technical Commission for Maritime Service (RTCM) - στα κινούμενα rover, τα οποία τελικά επιθυμούν να υπολογίσουν την θέση τους. Απλοποιημένο παράδειγμα αυτού, βρίσκεται στην Εικόνα 2.5.

Οι συγγραφείς του [55] πραγματοποιούν μεγαλύτερη ανάλυση λειτουργίας του, αναφέρουν λόγους ύπαρξης προβλημάτων, καθώς προτείνουν και μία αρχιτεκτονική η οποία μπορεί να βελτιώσει το availability του RTK-GPS σε ένα multi-vehicle system.

2.2 Ηχητικά χύματα

Μία εναλλακτική εκδοχή προσέγγισης υπολογισμού της θέσης, είναι με την βοήθεια ακουστικών κυμάτων. Παρόλο που αναλογικά με τις άλλες μεθόδους δεν είναι τόσο συχνά χρησιμοποιούμενη, υφίσταται ερευνητικό ενδιαφέρον για προσπάθειας απόκτησης location information με χρήση ήχου.

Της πάρχουν εφαρμογές στις οποίες χρειάζεται να ανιχνευτεί η θέση non cooperative - ακόμα και εχθρικών - drone τα οποία είτε να μην μπορούν να ανιχνευτούν - λόγο μεγέθους - από κάμερα είτε να μην μπορεί να χρησιμοποιηθεί κάποια RF τεχνική. Στην έρευνα [56] οι συγγραφείς εμπνευσμένοι από αυτό, κάνουν προσπάθεια δημιουργίας ενός συστήματος παρακολούθησης, για localization και tracking ενός drone με την χρήση acoustic arrays - παρουσιάζονται στην *Eικόνα 2.6*.



ΕΙΚΟΝΑ 2.6: Χρήση ακουστικών διατάξεων για εντοπισμό θέσης βασισμένο στο [56] ([URL](#))

Συγκεκριμένα, τοποθετούν σε συγκεκριμένες θέσεις τα δύο arrays, από τα οποία λαμβάνουν, με χρήση Generalized Cross Correlation - Phase Transform (GCC-PHAT), ένα πλήθος μετρήσεων TDoA μεταξύ των δύο arrays. Λόγο όμως το ότι από μόνες τους αυτές οι μετρήσεις - εξ' αιτίας του multipath effect - καταλήγουν να περιλαμβάνουν σφάλματα, προτείνουν έναν βασισμένο στο Gauss priori Probability Density Function (GPDF) αλγόριθμο* ώστε τελικά να εντοπίσουν την θέση των drone, ενώ στην συνέχεια χρησιμοποιούν KF για να το κάνουν track.

Ο εξοπλισμός που χρησιμοποιήθηκε ήταν sensors τύπου CHZ-213 1/2-inch metrological microphones, με κάθε sensor εφοδιασμένο με ένα pre-amplifiers τύπου YG-201. Ενώ, το drone του οποίου έγινε προσπάθεια να εκτιμηθεί η θέση ήταν το DJI Phantom 2. Τα αποτελέσματα που είχαν, ήταν να καταφέρουν στο μεγαλύτερο ποσοστό των εκτιμήσεων, τα σφάλματα να είναι μικρότερα των 6m σε μέγιστη απόσταση κάλυψης τα 100m.

2.3 Optical Oriented

Το πρόβλημα της εκτίμησης θέσης ενός αντικειμένου, έχει επίσης επιστημονικό ενδιαφέρον να λυθεί με χρήση οπτικών μέσων. Τέτοιοι τρόποι μπορεί να είναι με χρήση monocular κάμερας, stereo κάμερας ή τεχνικές που σχετίζονται με το φως - με μερικές από αυτές, να αναλύονται συνοπτικά στην συνέχεια του κεφαλαίου.

2.3.1 Κάμερα

Η πιο απλή εκδοχή^{*} είναι να γίνει χρήση μίας κάμερας και τεχνικών image processing για την εκροή συμπερασμάτων σχετικά με τον εντοπισμό θέσης UAV.

Στην έρευνα τους οι [57], προτείνουν την δημιουργία ενός 2-tier heterogeneous UAV swarm συστήματος το οποίο θα είναι σε θέση να αντικαταστήσει την ανάγκη από ανθρώπινο δυναμικό να διεκπεραιώνει επικίνδυνες διαδικασίες (όπως καθάρισμα των παραθύρων σε ουρανοξύστη).

Το σύστημα τους αποτελείται από τους workers και τους spotters, με τους workers να είναι τα να drone τα οποία δεν είναι απαραίτητο να είναι εφοδιασμένα με πλήθος αισθητήρων - καθώς με το να είναι τόσο κοντά σε κτήρια οι μετρήσεις τους μπορεί να μην έχουν ακριβή αποτελέσματα - και είναι αυτά που πραγματοποιούν το κύριο μέρος της αποστολής. Σε αντίθεση, οι spotters τοποθετούνται σε απόσταση από τους workers και είναι αυτοί που είναι υπεύθυνοι για τον εντοπισμό της θέσης των workers, με το μεγαλύτερο μέρος της επεξεργασίας να πραγματοποιείται σε αυτούς.

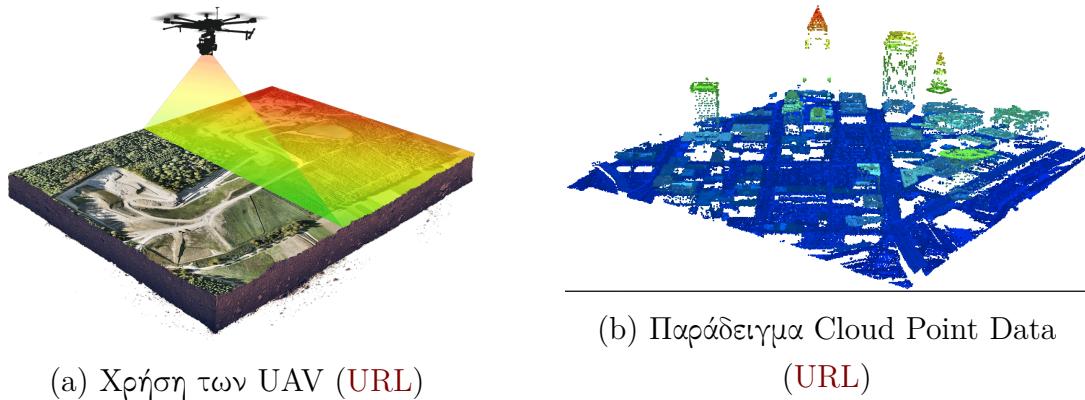
Με την βοήθεια μίας πολύχρωμης ping pong μπάλας - η οποία είναι τοποθετημένη πάνω στους workers και χρησιμοποιείται ως vision marker - χρήση μίας κάμερας και συνδυασμό αυτής με την βιβλιοθήκη OpenCV [58], καταφέρνουν να εντοπίσουν στο 3D την θέση της μπάλας, ενώ σε συνδυασμό με statistical Sensor Fusion, χρήση EKF και dead-reckoning καταφέρνουν να κινήσουν τους workers στον χώρο.

Για τα workers - τα οποία κατάφεραν στα πειράματα τους να κάνουν hover με ακρίβεια $<0.1m$ - χρησιμοποίησαν πολλαπλά CrazyFlie 2.0, ενώ για το spotter έγινε χρήση υπολογιστή και μίας USB κάμερας.

2.3.2 LiDAR

Ένας ακόμα τρόπος εκτίμησης της θέσης, μπορεί να προκύψει με χρήση τεχνικών από τον κλάδο των remote sensing technologies, συγκεκριμένα με την χρήση Light Detection And Ranging (LiDAR) προσέγγισης. Ο τρόπος λειτουργίας αυτής της τεχνικής - απλοποιημένο παράδειγμα στην Εικόνα 2.7 (a) - είναι πρακτικά η εκπομπή

laser pulses προς διάφορες κατευθύνσεις, η εκτίμηση της απόστασης κάθε αντικειμένου στα οποία ανακλά η κάθε στοιχειώδη αυτή ακτινοβολία και ο συνδυασμός αυτών των αποστάσεων σε ένα συνολικό μοντέλο, ώστε τελικά να δημιουργήσουμε 3D reconstructions των σκαναρισμένων χώρων, που συχνά τις ονομάζουμε LiDAR Cloud Point Data - *Εικόνα 2.7* (b) - οι οποίες χρησιμοποιούνται στην συνέχεια για το localization [59].



EIKONA 2.7: LiDAR

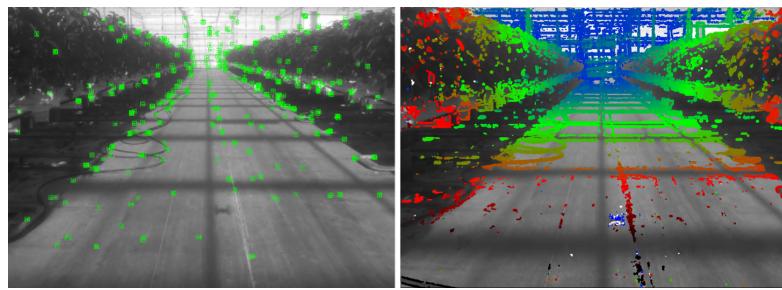
Οι συγγραφείς του [60] κάνουν προσπάθεια χρήσης αυτής της τεχνολογίας, προκειμένου να εξαλείψουν την ανάγκη καμερών - οι οποίες είναι σε άμεση εξάρτηση από τις συνθήκες φωτισμού - ώστε να επιτύχουν ένα long distance drift-free navigation ενός ελικοπτέρου.

Η μεθοδολογία που ακολουθούν είναι να σπάνε το 3D localization problem σε ένα 2D translation problem και ένα 1D altitude problem, τα οποία διαδοχικά προσπαθούν να επιλύσουν. Η γενική αρχή που ακολουθούν είναι ότι προσπαθούν να βρουν την βέλτιστη ευθυγράμμιση μεταξύ των μετρήσεων του LiDAR και ενός geo-referenced Digital Elevation Model ([DEM](#)) ώστε να καταλήξουν στο 3D world position.

Τα αποτελέσματα που είχαν ήταν final position error της τάξης των 27m σε απόσταση ταξιδιού τα 218km. Χρησιμοποιήθηκαν dataset από ένα Bell 206L (Long-Ranger) helicopter, εξοπλισμένο με ένα Near Earth Autonomy m4 sensor suite το οποίο περιλαμβάνει ένα 2D LiDAR και ένα fiber-optic IMU. Ενώ χρησιμοποιήθηκε υπολογιστής εξοπλισμένος με quad-core 2.5GHz i7 processor, στον οποίο σε παράλληλο προγραμματισμό έγινε το filtering και matching.

2.3.3 VSLAM

Μία ειδική κατηγορία - με χρήση κάμερας - είναι αυτή στην οποία ακολουθείται η λογική του Visual Simultaneous Localization and Mapping (VSLAM). Σε αυτήν την περίπτωση, το UAV προσπαθεί να εντοπίσει τις συντεταγμένες του - αφού πρώτα δημιουργήσει έναν map του χώρου στον οποίο βρίσκεται - ώστε τελικά να βρει την θέση που έχει μέσα σε αυτόν. Υπάρχουν διαφορετικοί Simultaneous Localization and Mapping (SLAM) αλγόριθμοι γενικά, με αυτούς που μπορούν να κατηγοριοποιηθούν στους σχετιζόμενος με filtering και αυτούς που σχετίζονται με smoothing τεχνικές [61]. Συνήθως σε αυτούς τους αλγορίθμους, γίνεται εύρεση landmarks στον χώρο, τα οποία χρησιμοποιούνται - *Eikona 2.8* - στην διαδικασία.



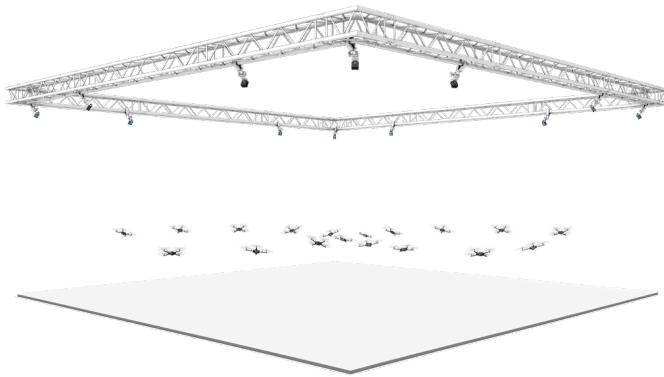
ΕΙΚΟΝΑ 2.8: Παράδειγμα αντιληπτικής ικανότητας VSLAM σε drone
βασισμένο στο [62] (URL)

Στο [63] προτείνουν μία αρχιτεκτονική ενός συνεργατικού key-frame-based Multi-UAV SLAM συστήματος. Σε αυτό το σύστημα το κάθε μεμονωμένο drone έχει την δυνατότητα μέσω μίας monocular κάμερας και χρήση real-time Visual Odometry να εκτιμήσει την θέση του, και να παρέχει ένα - περιορισμένων key-frames, εξαιτίας των on-board computation δυνατοτήτων - Local Map του περιγύρου του. Ταυτόχρονα, το σύστημα αποτελείται και από ένα communication module, ώστε τα drone να μπορούν να αλληλεπιδρούν με ένα server, ο οποίος χρησιμοποιείτε για non time-critical computationally expensive processes.

Βασικές τεχνολογίες που χρησιμοποιήθηκαν ήταν Visual Odometry and Place Recognition βασισμένες στην υλοποίηση του ORB-SLAM2 αλγόριθμου, το ROS framework για τις επικοινωνίες. Επίσης, ο server που χρησιμοποιήθηκε ήταν ένα Thinkpad T460s notebook εξοπλισμένο με τον τετραπύρηνο Core i7-6600U @ 2.60GHz και 20 GB RAM.

2.3.4 Motion Capture System

Συχνή χρήση έχουν επίσης τα Motion Capture (MoCap) Systems [64]. Όταν αναφερόμαστε σε MoCap systems, όπως το Vicon [65] ή το Optitrack [66], μιλάμε κυρίως για στατικά, εσωτερικών χώρων συστήματα τα οποία είναι υπεύθυνα να ανιχνεύουν και να συλλαμβάνουν την κίνηση σωμάτων. Αποτελούνται από ένα πλήθος καμερών - οι οποίες είναι τοποθετημένες σε προκαθορισμένες θέσεις - μέσω των οποίων μπορεί να εκτιμηθεί η θέση αντικειμένων - όπως drones - στο χώρο αναφοράς. Παράδειγμα αυτού υπάρχει στην *Eikóna 2.9*



EIKONA 2.9: Χρήση του Optitrack Motion Capture System για εντοπισμό σμήνος από drone ([URL](#))

Στην προσπάθεια τους οι [67] χρησιμοποιούν ένα τέτοιο MoCap system, προκειμένου να δημιουργήσουν ένα multi-UAV αλγόριθμο ελέγχου για centralized formation flight.

Στα πειράματα τους χρησιμοποιήθηκαν το Vicon, ως drone τα Parrot AR Drone 2.0, ενώ η υλοποίηση έγινε στο περιβάλλον του MATLAB/SIMULINK αξιοποιώντας το AR Drone Simulink Development Kit.

Κεφάλαιο 3

Θεωρητικό Υπόβαθρο

”Μηδείς ἀγεωμέτρητος εἰσίτω
μου τὴν στέγην”

Πλάτωνας

Ο χώρος των **WSN** έχει και αυτός τα τελευταία χρόνια υψηλό ερευνητικό ενδιαφέρον. Θα μπορούσε να πει κανείς - λόγο το ότι αποτελεί ένα πιο γενικό κλάδο - περισσότερο από ότι αυτό των drones. Συνεπώς θα μεταφερθούμε σε πρώτο επίπεδο στο πιο γενικό φάσμα, αυτό των **WSN** για να προσεγγίσουμε το localization problem.

Όταν μιλάμε για **WSN**, αναφερόμαστε σε αυτόνομα ηλεκτρονικά συστήματα, χωρικά διασκορπισμένα σε ένα πεδίο - τα οποία συχνά περιλαμβάνουν αισθητήρες και επικοινωνούν με τα γειτονικά τους ή **BS** για να μεταφέρουν πληροφορία [68] [69].

Το καθένα από αυτά, τα ανεξάρτητα συστήματα, ονομάζεται **Node**. Ενώ, για το κάθε μεμονωμένο node - μπορεί να έχουμε στην διάθεση μας location information ή όχι. Μία πρώτη σκέψη θα ήταν, κάθε node ενός συστήματος αισθητήρων να περιλαμβάνει **GPS** ώστε να γνωρίζουμε την θέση του. Αυτό μπορεί γρήγορα να καταρριφθεί σαν σκέψη, αν αναλογιστούμε αρχικά ότι το Global Navigation Satellite System (**GNSS**) δεν είναι διαθέσιμο σε κάθε περιβάλλον λειτουργίας¹, όπως επίσης μπορεί να μην είναι δυνατή η χρήση του σε όλους τους κόμβους ενός συστήματος, λόγο περιορισμών όπως το κόστος², μέγεθος του node & energy consumption [69], ή ακόμα ενδεχόμενο παρατήρησης και εντοπισμού ενός ετερογενούς - από το σύστημα μας - αντικειμένου στο οποίο δεν είμαστε σε θέση να επέμβουμε.

Στην υπάρχουσα βιβλιογραφία [69] [70] [71] βρίσκουμε ότι nodes των οποίων η θέση είναι γνωστή ή άμεσα υπολογίσιμη, συχνά ονομάζονται **Beacons**. Πληροφορία σχετικά με την θέση αυτών των nodes είναι γνωστή, είτε γιατί έχουν τοποθετηθεί

¹Παράδειγμα ενός να είναι οι εσωτερικοί χώροι

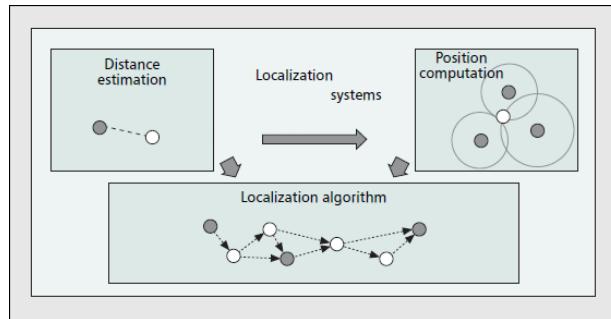
²Ειδικά για low-end-low-cost devices, μπορεί να είναι ακόμα και αποτρεπτικός παράγοντας

από εμάς σε προκαθορισμένες θέσεις, είτε μέσου ενός εξωτερικού συστήματος όπως το GPS [72]. Αντίθετα κόμβοι για τους οποίους δεν έχουμε αρχικά πληροφορία της θέσης τους, ονομάζονται **Non-anchors**. Άλλος ένας σημαντικός ορισμός, που θα πρέπει να αναφερθεί είναι ότι συχνά ονομάζουμε **Settle nodes**, αυτά τα οποία αρχικά δεν γνωρίζαμε την θέση τους αλλά στην συνέχεια την εκτιμήσαμε. Στο *Πίνακα 3.1* παρουσιάζονται συνοπτικά τα διάφορα ονόματα που έχουν δοθεί ανά καιρούς για το κάθε τύπο node.

ΠΙΝΑΚΑΣ 3.1: Ορισμοί ονομάτων των Nodes

Node name	Περιγραφή
Unknown/Free/Dumb/Non-anchors	Κόμβοι με άγνωστη θέση
Beacons/Anchors/Landmarks	Κόμβοι με γνώση της θέσης τους
Settled	Κόμβοι που αρχικά δεν γνωρίζαμε την θέση τους αλλά την εκτιμήσαμε

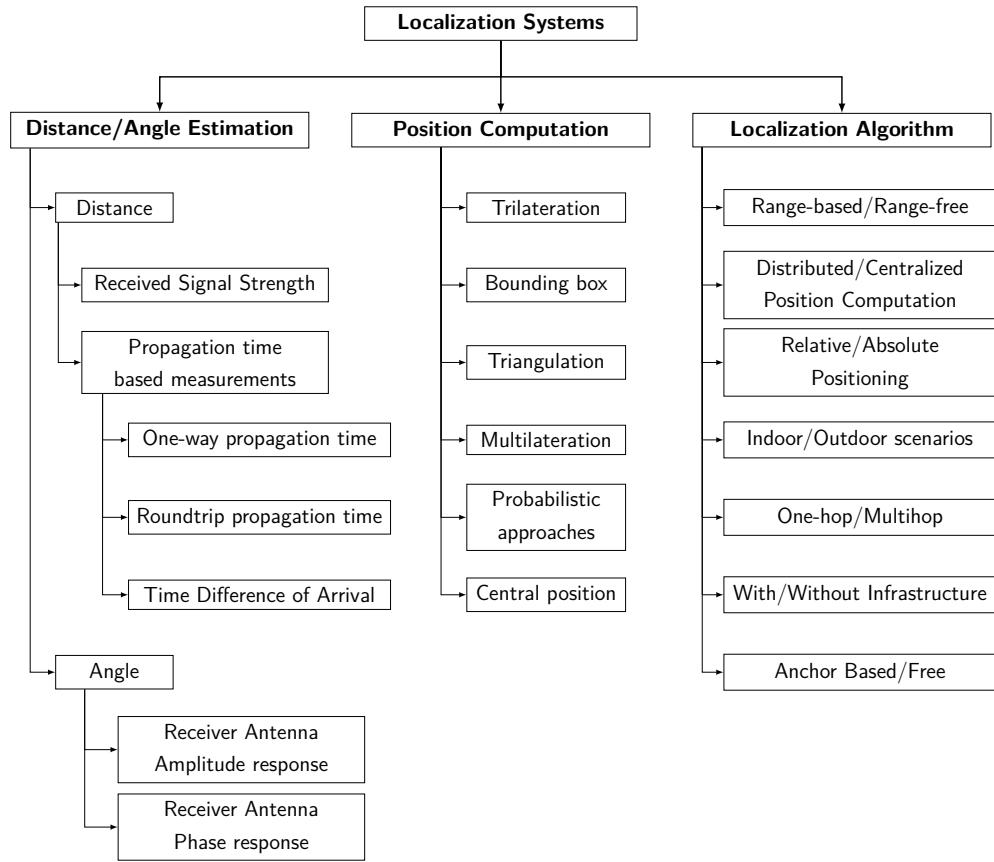
Σκοπός ενός localization system είναι, με χρήση της γνώσης που έχουμε για τα beacon nodes να εκτιμήσουμε την θέση όσο περισσότερων unknown nodes ώστε να τα μετατρέψουμε σε settled nodes και η εκτίμηση της κάθε θέσης να είναι με όσο το δυνατόν μικρότερο error απόκλισης.



ΕΙΚΟΝΑ 3.1: Στοιχεία των Localization Systems βασισμένο στο [70]
(URL)

Οι συγγραφείς του [70] επιχειρούν να χωρίσουν ένα localization system, ώστε αυτό να αποτελείται από τρία διαχριτά components, αυτή τη κατηγοριοποίηση την υιοθετούν κατά την έρευνα τους και οι ερευνητές του [73]. Πρώτο μπορεί να θεωρηθεί αυτό του **Distance/Angle Estimation**, που σκοπό έχει να υπολογίσει την γωνία ή απόσταση που έχουν δύο nodes του συστήματος μεταξύ τους. Η πληροφορία που θα παραχθεί από αυτό το component θα χρησιμοποιηθεί στα άλλα μέρη του συστήματος. Στην συνέχεια υπάρχει το **Position Computation**, δουλειά του οποίου είναι

να υπολογίσει την θέση ενός node με βάση την γνώση που έχουμε για τα beacons και την πληροφορία που λάβαμε από το πρώτο component. Ενώ τέλος είναι το κύριο μέρος του συστήματος, με όνομα **Localization Algorithm** και ουσιαστικά είναι ο προκαθορισμένος τρόπος που θα ακολουθήσει για να υπολογιστεί η θέση των unknown nodes με βάση όλες τις πληροφορίες που έχουμε. Στην *Eikόνα 3.1* δίνεται η απεικόνιση που έδωσαν οι συγγραφείς του [70] για να εξηγήσουν το παραπάνω. Ενώ στην *Eikόνα 3.2* έχει γίνει μία προσπάθεια να κατηγοριοποιηθούν τα κομμάτια καθώς και τεχνικές των Localization Systems, με βάση τα [69] [70] [71] και αναλύονται στην συνέχεια του κεφαλαίου.



ΕΙΚΟΝΑ 3.2: Επισκόπηση των Localization Systems (RF-based approach)

Με βάση τον παραπάνω διαχωρισμό των Localization Systems σε τρία διαχριτά μέρη, μπορούμε να καταλάβουμε ότι η απόκλιση της εκτίμησης του συνολικού συστήματος, εξαρτάται από τα σφάλματα του κάθε μεμονωμένου μέρους.

Τέλος, έχει γίνει ήδη αναφορά ότι το πρόβλημα προσδιορισμού της θέσης ενός αντικειμένου, μπορεί να προσεγγιστεί με χρήση διαφορετικών μεθόδων, με αυτές να μπορούν να κατηγοριοποιηθούν σε τεχνικές που σχετίζονται με ενέργεια (e.g. RF

and sound) και τεχνικές που σχετίζονται με οπτικά μέσα³. Στην συνέχεια του κεφαλαίου, γίνεται διαδοχική αναφορά στην κάθε μία εξ' αυτών.

3.1 Εντοπισμός θέσης με βάση την ενέργεια

3.1.1 Εκτίμηση Απόστασης/Γωνίας

Θα ξεκινήσουμε με μία σύντομη ανάλυση των τεχνικών εκτίμησης της απόστασης - μεταξύ δύο nodes - που χρησιμοποιούνται ήδη, στις προσεγγίσεις RF & sound.

Received Signal Strength

Η πρώτη τεχνική η οποία έχει χρησιμοποιηθεί για τον υπολογισμό απόστασης στα WSN με χρήση RF, είναι αυτή με όνομα Received Signal Strength Indicator (RSSI) και έχει ως αρχή την χρήση της έντασης της ισχύς ενός σήματος που λαμβάνουμε στον δέκτη, ως τρόπο υπολογισμού της απόστασης του πομπό από αυτόν. Path loss ή path attenuation [74] ονομάζεται η μείωση της ισχύς ενός σήματος καθώς αυτό διαδίδεται. Στον ελεύθερο χώρο η λαμβανόμενη ισχύς $P_r(d)$ που ανιχνεύει ο πομπός μπορεί να περιγραφτεί από το μοντέλο του Free Space Path Loss (FSPL) [75], το οποίο προκύπτει μέσω της Friis transmission equation [71] [76] [77] - σχέση (3.1).

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2} \quad (3.1)$$

Όπου P_t είναι η ισχύς που στέλνει ο πομπός, G_t είναι το gain της κεραίας του πομπού, G_r το gain της κεραίας του δέκτη, λ είναι το μήκος κύματος του σήματος το οποίο μεταδίδουμε και d η απόσταση του πομπού από τον δέκτη. Αν θεωρήσουμε ότι τα G_t , G_r και λ είναι μη μεταβλητές τιμές - με $C_f = \frac{G_t G_r \lambda^2}{(4\pi)^2}$ - τότε μπορούμε να καταλήξουμε στην (3.2) [78].

$$P_r(d) = C_f \frac{P_t}{d^2} \quad (3.2)$$

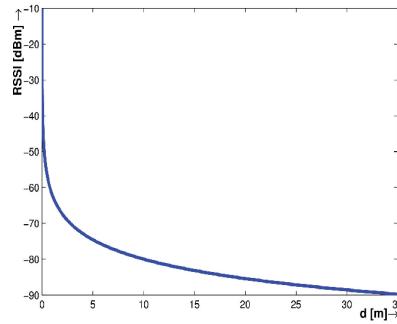
Αυτό που μπορούμε να δούμε από την παραπάνω σχέση είναι ότι, ιδανικά στον ελεύθερο χώρο σε Line of Sight (LoS) μετάδοση - η ισχύς του σήματος που λαμβάνει ο πομπός εξαρτάται από το αντίστροφο τετράγωνο της απόστασης των δύο nodes. Η

³Ενώ κάποιες φορές μπορούν να συνδυαστούν βασικές αρχές της κάθε μία για μία ενοποιημένη υλοποίηση

σχέση (3.2) συχνά αναφέρεται σε watt, όμως όταν μιλάμε για transmitting power αντί για την χρήση των watt, είναι αρκετά βολική η χρήση του dBm [79]. Για αυτό το λόγο, παρακάτω παρουσιάζονται τα conversion equations, από το ένα στο άλλο [76] [79].

$$\begin{cases} P[dBm] = 10 \cdot \log_{10} \left(\frac{P[mW]}{1[mW]} \right) \\ P[mW] = 10^{\frac{(P[dBm])}{10}} \end{cases} \quad (3.3)$$

Ενώ η *Eikόνα 3.3* περιγράφει σχηματικά το ιδανικό μοντέλο της εξάρτηση της ισχύς με την αύξηση της απόστασης όταν χρησιμοποιούμε στον δέκτη μέτρηση σε dBm.



ΕΙΚΟΝΑ 3.3: Ιδανική εξάρτηση του RSSI με βάση την απόσταση
βασισμένο στο [80] ([URL](#))

Αν λοιπόν, μετράμε την ισχύ με την οποία λαμβάνουμε ένα σήμα, μπορούμε να είμαστε σε θέση να υπολογίσουμε την απόσταση που βρίσκεται ο πομπός από εμάς. Αυτή η μέθοδος παρόλο που είναι αρκετά δημοφιλής και οικονομική για τον υπολογισμό της απόστασης - λόγω του ότι δεν απαιτεί επιπλέον αισθητήρες - σε πραγματικές συνθήκες αντιμετωπίζει αρκετά προβλήματα, καθώς οι μετρήσεις μπορούν να επηρεαστούν από θόρυβο, ανακλάσεις του σήματος, διαθλάσεις, δυναμικά περιβάλλοντα ή εμπόδια σε αυτά, Non-Line of Sight (NLoS) μετάδοση, ή ακόμα και errors στο hardware [70] [80]. Σε ένα βαθμό μπορεί να βελτιωθεί η απόδοση με στατικό ή δυναμικό calibration του συστήματος, όμως μέχρι τώρα δεν χρησιμοποιείται για εκτίμηση απόστασης σε εφαρμογές όπου nodes έχουν μεγάλη απόσταση μεταξύ τους ή μας ενδιαφέρει να έχουμε μεγάλη ακρίβεια προσέγγισης της απόστασης [80].

Χρόνος μετάδοσης σήματος

Σε αυτήν την κατηγορία εκτίμησης απόστασης μεταξύ nodes - η οποία βασίζεται σε χρονικές μετρήσεις της διάδοσης του σήματος, Time of Flight (ToF) - κατά κύριο

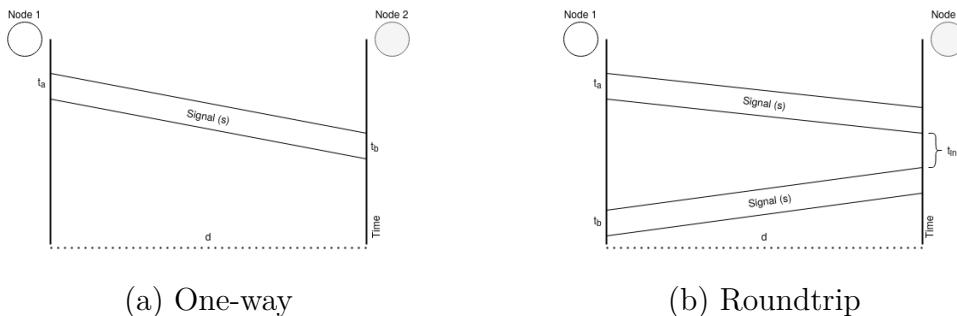
λόγο χρησιμοποιούνται δύο βασικές τεχνικές, η Time of Arrival (ToA) και η Time Difference of Arrival (TDoA) [70].

Θα ξεκινήσουμε με μία σύντομη περιγραφή της ToA. Από την κινηματική γνωρίζουμε την σχέση (3.4), η οποία συσχετίζει την ταχύτητα κίνησης ενός σώματος V ως το πιλίκιο της μεταβολής της θέσης ds - που έκανε το σώμα, προς τον χρόνο dt που χρειάστηκε για να πραγματοποιηθεί η μεταβολή [81].

$$V = \frac{ds}{dt} \quad (3.4)$$

$$d = s(t_b - t_a) \quad (3.5)$$

Αξιοποιώντας την (3.4) ως αρχή, μπορούμε να καταλήξουμε στην (3.5) για να εκτιμήσουμε την απόσταση d που βρίσκονται δύο nodes μεταξύ τους, αν ένα κύμα κινείται με ταχύτητα s και χρειάστηκε χρόνο t για να μεταδοθεί από το ένα node στο άλλο. Στην Εικόνα 3.4 (a) απεικονίζεται σχηματικά αυτό, όπου $t = t_b - t_a$ με t_b η χρονική στιγμή που φτάνει το κύμα στο receiver και t_a η χρονική στιγμή η οποία ξεκινάει από τον transmitter. Σε περίπτωση που μιλάμε για RF η ταχύτητα μετάδοσης του κύματος είναι ίση με την ταχύτητα μετάδοσης του φωτός c_0 , το οποίο σε $0.1\mu s$ διανύει περίπου $30m$. Με βάση αυτό, μπορούμε εύκολα να καταλάβουμε ότι για να έχουμε ακριβή αποτελέσματα είναι αρκετά σημαντικό τα clocks των δύο nodes να είναι απόλυτα συγχρονισμένα για να μην έχουμε error απόκλισης, πράγμα που απαιτεί να κάνουμε το συνολικό σύστημα αρκετά πιο πολύπλοκο σχεδιαστικά ώστε η απόκλιση μας να είναι σε ανεκτά σημεία για την εφαρμογή [70] [71].



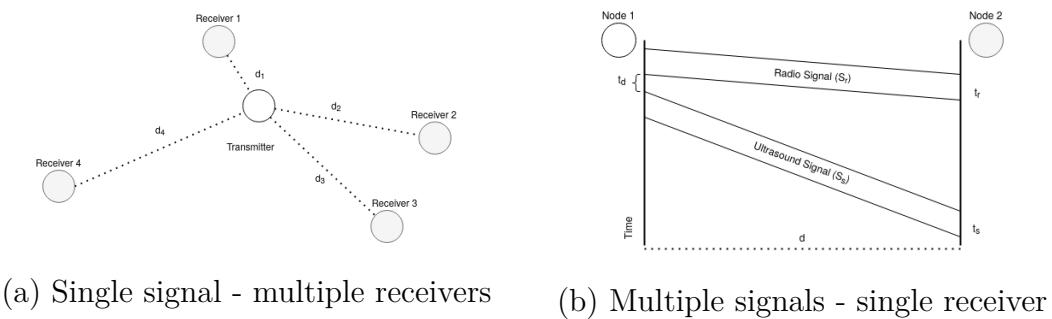
ΕΙΚΟΝΑ 3.4: Περιπτώσεις Time of Arrival

Μπορούμε να το παρακάμψουμε αυτό, με το να γίνει η μέτρηση σε Round-Trip Time (RTT), Εικόνα 3.4 (b). Σε αυτήν την περίπτωση το ένα node στέλνει ένα σήμα, και μόλις το λάβει ένα γειτονικό node, απαντάει πίσω στο πρώτο. Με αυτόν τον τρόπο η

μέτρηση του χρόνου εκκίνησης t_a και άφιξης t_b του σήματος γίνονται στο ίδιο node - άρα δεν χρειάζεται συγχρονισμός, και η πραγματική απόσταση είναι η μισή από αυτή που θα υπολογιστεί. Ο κύριος παράγοντας σφάλματος σε αυτή την μέθοδο, είναι ο υπολογισμός του χρόνου που χρειάστηκε το δεύτερο node για να διαχειριστεί το σήμα που έλαβε και να απαντήσει. Αυτό το internal delay t_{in} μπορεί να είναι είτε γνωστό από ένα a priori calibration, είτε μπορεί να μετριέται και να στέλνεται μαζί με το σήμα απάντησης - ώστε να αφαιρείται από τον χρόνο μετάδοσης του κύματος [71]. Με αυτά τα δεδομένα, η σχέση (3.6) περιγράφει τον τρόπο υπολογισμού της απόστασης d μεταξύ των δύο nodes.

$$d = \frac{s(t_b - t_a - t_{in})}{2} \quad (3.6)$$

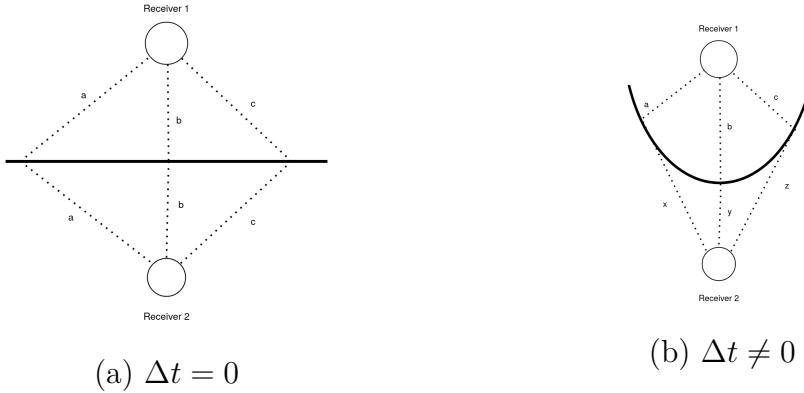
Όσον αφορά την τεχνική **TDoA**, και σε αυτήν την περίπτωση υπάρχουν δύο παραλλαγές της [70] - όπου και οι δύο είναι βασισμένες στην αρχή το ότι δεν μας ενδιαφέρει η χρονική στιγμή που ξεκίνησε η αποστολή ενός σήματος, αλλά μόνο οι χρονική στιγμή που το λάβαμε.



ΕΙΚΟΝΑ 3.5: Περιπτώσεις Time Difference of Arrival

Η πρώτη περίπτωση σχετίζεται με single signal και multiple receivers, παρουσιάζεται στην Εικόνα 3.5 (a), χρησιμοποιείται συνήθως στα cellular network και απαιτεί την ύπαρξη τουλάχιστον 4 beacons για την απόκτηση location πληροφορίας ενός free node στον τρισδιάστατο χώρο. Υπολογίζει την χρονική διαφορά που έφτασε σε καθένα από τα beacons το σήμα που έστειλε το free node, για την εκτίμηση της θέσης του free node από το καθένα από αυτά. Σημαντικό σε αυτήν την περίπτωση είναι και εδώ, να είναι απόλυτα συγχρονισμένα τα beacons. Βασίζεται στις ιδιότητες των υπερβολικών καμπύλων και στο ότι με δεδομένη μία συγκεκριμένη χρονική διαφορά του σήματος, το free node θα πρέπει να βρίσκεται κάπου πάνω σε μία υπερβολική καμπύλη όπως παρουσιάζεται από την Εικόνα 3.6 [82].

Μόνο από αυτόν τον υπολογισμό δεν δύναται να κάνουμε ακριβή εκτίμηση απόστασης, μπορούμε να συμπεράνουμε όμως τον γεωμετρικό τόπο στον οποίο βρίσκεται το free node. Στη *Παράγραφο 3.1.2* - και σε επέκταση αυτού - παρουσιάζεται όμως, ο τρόπος με τον οποίο μπορούμε να συνδυάσουμε τέτοια πληροφορία από πολλαπλά beacon για να εκτιμήσουμε την ακριβή θέση του free node.



ΕΙΚΟΝΑ 3.6: TDoA υπερβολές βασισμένες στο Δt

$$\Delta t = a - a = 0$$

$$\Delta t = x - a$$

$$\Delta t = b - b = 0$$

$$\Delta t = y - b$$

$$\Delta t = c - c = 0$$

$$\Delta t = z - c$$

Η δεύτερη εκδοχή της χρήσης **TDoA** - η οποία είναι πιο συχνή στα **WSN** και μας ενδιαφέρει περισσότερο, χρησιμοποιεί multiple signals με single receiver και παρουσιάζεται στην *Εικόνα 3.5* (b). Αρχή λειτουργίας έχει ότι ο transmitter θα στείλει πολλαπλά - διαφορετικού είδους - σήματα και ο δέκτης θα μετρήσει την χρονική διαφορά που τα έλαβε. Ως παράδειγμα μπορεί το ένα σήμα να είναι σε **RF** και να κινείται με ταχύτητα $s_r = c_o$ και το άλλο ηχητικό με ταχύτητα $s_s \approx 343m/s$ ⁴. Αν t_r η χρονική στιγμή που λαμβάνει το **RF** σήμα, t_s το ηχητικό και t_d το delay που μεσολάβησε από την αποστολή του ενός στο άλλο, τότε μπορούμε να υπολογίσουμε την απόσταση μεταξύ των κόμβων από την εξίσωση (3.7) [70] [84].

$$d = (s_r - s_s)(t_s - t_r - t_d) \quad (3.7)$$

⁴Για μετάδοση σε ξηρό αέρα υπό θερμοκρασία 20°C [83]

Θετικό σε αυτήν την μέθοδο είναι ότι το error μπορεί να είναι της τάξης των μερικών εκατοστών, όμως αρχικά απαιτεί επιπλέον εξοπλισμό στο node - ώστε να μπορεί να στείλει και να λάβει πολλαπλά είδη σήματος - πράγμα που μπορεί να το κάνει αντιοικονομικό ή αρκετά μεγαλύτερο σε διαστάσεις από το επιθυμητό. Όπως επίσης - και μάλιστα σημαντικότερο - η απόσταση η οποία μπορεί να χρησιμοποιηθεί, επηρεάζεται σε μεγάλο βαθμό από τα χαρακτηριστικά του δεύτερου σήματος. Ως παράδειγμα, τα ηχητικά σήματα δεν μπορούν να μεταφερθούν σε μεγάλες αποστάσεις ή το ότι η ταχύτητα τους μπορεί να επηρεαστεί σημαντικά από περιβαλλοντολογικούς παράγοντες [69].⁵

Εκτίμηση γωνίας

Άλλη μία χρήσιμη μέτρηση η οποία μας ενδιαφέρει, είναι η εκτίμηση της γωνίας από την οποία λαμβάνουμε το σήμα ενός γειτονικού node σε σχέση με έναν άξονας αναφοράς. Ο άξονας αυτός μπορεί να είναι είτε κοινώς για όλα τα nodes (π.χ. ως προς το βόρειο γεωγραφικό πόλο), είτε μπορεί να είναι για το κάθε node ξεχωριστός, ως παράδειγμα με βάση τον προσανατολισμό του ίδιου του node ή με βάση την γωνία λήψης ενός επιπλέον σήματος [70]. Την πληροφορία αυτή την βρίσκουμε στην βιβλιογραφία να ονομάζεται Angle of Arrival (AoA) ή Direction of Arrival (DoA).

$$s(t) = A \sin(2\pi f t + \phi) \quad (3.8)$$

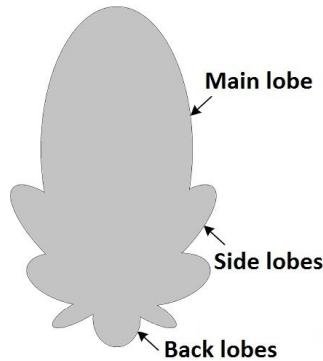
Γνωρίζουμε ότι ένα σήμα καθορίζεται πλήρως από τρεις παραμέτρους, το πλάτος του Α, την συχνότητα του f καθώς επίσης και από την φάση του ϕ - σχέση (3.8). Για τον υπολογισμό της γωνίας AoA μπορούν να χρησιμοποιηθούν τεχνικές με γνώμονα τα παραπάνω.

Amplitude response

Ο όρος *Beamforming* [85] χρησιμοποιείται για να περιγράψει το pattern της ευαισθησίας του σήματος που στέλνει/λαμβάνει μία *directional antenna* [86]. Το pattern αυτού του τύπου κεραίας, παρουσιάζεται στην Εικόνα 3.7 [71] και μπορούμε να χρησιμοποιήσουμε αυτό το χαρακτηριστικό για τον υπολογισμό της AoA. Εάν χρησιμοποιούμε *directional antenna* στον receiver και μπορούμε είτε με μηχανικό είτε με ηλεκτρονικό τρόπο να στρέψουμε την κεραία σε διάφορες κατευθύνσεις, τότε μπορούμε μετρώντας το πλάτος του σήματος που λαμβάνουμε, να εκτιμήσουμε και

⁵Η συγκεκριμένη μέθοδος είναι διαισθητικά παρόμοια, με αυτό που θα κάναμε για να εκτιμήσουμε την απόσταση μίας αστραπής από εμάς, μετρώντας το χρονικό διάστημα μέχρι να ακούσουμε την βροντή.

την κατεύθυνση του - όπου ιδανικά βρίσκεται εκεί από όπου έχουμε την μέγιστη τιμή του amplitude του κύματος. Σημαντικές παράμετροι για την εκτίμηση της γωνίας σε αυτήν την περίπτωση, είναι επίσης η ευαισθησία του δέκτη καθώς και το width του beam της. Λανθασμένη εκτίμηση μπορεί να προκύψει, αν για κάποιο λόγο το πλάτος του σήματος που λαμβάνουμε δεν είναι σταθερό. Σε αυτή την περίπτωση, μία λύση είναι να χρησιμοποιήσουμε μία δεύτερη - μη κινούμενη - *omnidirectional* κεραία και να κανονικοποιήσουμε τις μετρήσεις της *directional* με βάση τις μετρήσεις της δεύτερης [71].



ΕΙΚΟΝΑ 3.7: Pattern ευαισθησίας της anisotropic κεραίας βασισμένο στο [71] ([URL](#))

Phase response

Άλλος τρόπος υπολογισμού του AoA είναι με την αξιοποίηση της πληροφορία για την φάση που έχουμε για ένα σήμα. Σε αυτήν την περίπτωση χρειαζόμαστε ή κεραίες αρκετά μεγαλύτερες από το μήκος κύματος του σήματος που στέλνουμε, ή antenna arrays [71].

Για τα κύματα γνωρίζουμε ότι η διαφορά φάσης $\Delta\phi$ δύο σημείων που απέχουν μεταξύ τους $\Delta\chi$ μπορεί να υπολογιστεί από την σχέση (3.9) [87].

$$\Delta\phi = 2\pi \frac{\Delta\chi}{\lambda} = 2\pi \frac{\Delta t}{T} \quad (3.9)$$

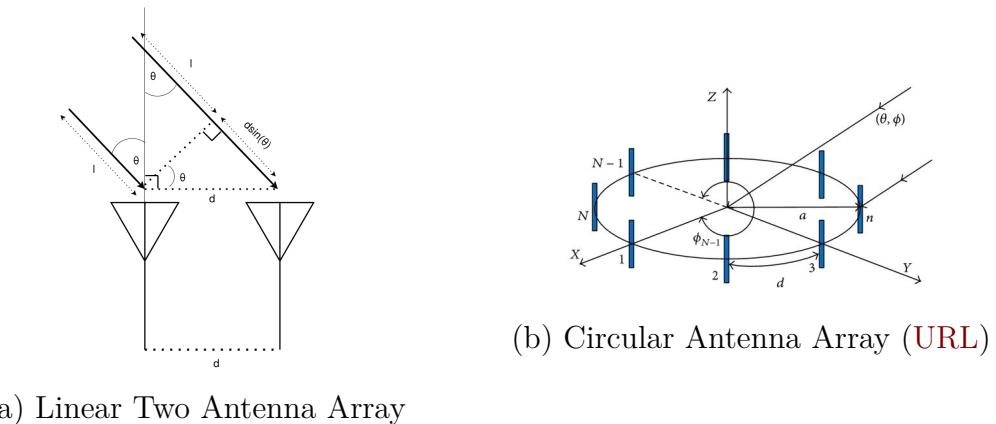
Συνεπώς, αν σχεδιάσουμε ένα antenna array, όπου η κάθε κεραία απέχει από την γειτονική της σε συγκεκριμένη απόσταση d και μετρώντας το phase difference του σήματος που λαμβάνουμε μεταξύ τους, τότε μπορούμε να υπολογίσουμε την γωνία πρόσληψης της ακτινοβολίας μέσω της σχέσης (3.10) [71] [88]. Παράδειγμα αυτού·

βρίσκεται στην *Εικόνα 3.8* (a). Σε αυτήν την περίπτωση χρησιμοποιούμε *omnidirectional* κεραίες και ο υπολογισμός της διαφοράς φάσης μπορεί να γίνει μέσω μετρήσεων **TDoA**.

$$\phi = \frac{2\pi}{\lambda} d \sin(\theta) \Rightarrow$$

$$\theta = \sin^{-1} \left(\frac{\lambda \phi}{2\pi d} \right) \quad (3.10)$$

Δεν υπάρχουν όμως μόνο τα linear antenna arrays, μπορούμε να τα σχεδιάσουμε σε διάφορες διατάξεις, παράδειγμα μίας εξ' αυτών βρίσκεται στην *Εικόνα 3.8* (b).



ΕΙΚΟΝΑ 3.8: Παραδείγματα Antenna Arrays

Η μέθοδος με την αξιοποίηση της σχέσης (3.10) λειτουργεί αρκετά καλά όταν έχουμε υψηλό Signal to Noise Ratio (**SNR**), αλλά μπορεί να έχει λανθασμένα αποτελέσματα όταν έχουμε διασυμβολικές παρεμβολές ή multipath signals. Επειδή στην πράξη εμφανίζονται και τα δύο, υπάρχει αρκετά μεγάλο επιστημονικό ενδιαφέρον να αντιμετωπιστεί αυτός ο περιορισμός. Αρκετές τεχνικές έχουν προταθεί οι οποίες στρέφονται γύρω από την λογική του Maximum Likelihood (**ML**) για τον υπολογισμό τελικά της γωνίας [71].

3.1.2 Υπολογισμός θέσης

Αν έχουμε δύο nodes, οι μετρήσεις που λάβαμε από το *Παράγραφο 3.1.1* είναι αρκετές για να γνωρίζουμε την θέση του καθενός, συνεπώς ενδιαφέρον βρίσκεται στην ύπαρξη τριών ή παραπάνω nodes σε ένα σύστημα. Με βάση τις πληροφορίες που έχουμε συλλέξει - μέσω των τεχνικών που περιγράφτηκαν στο προηγούμενο κεφάλαιο - θα

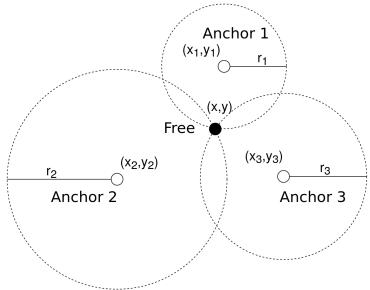
προσπαθήσουμε πλέον να εκτιμήσουμε την θέση ενός node. Στην συνέχεια αυτού του section, περιγράφονται μέθοδοι οι οποίοι μπορούν να χρησιμοποιηθούν για να το επιτύχουν αυτό. Κύρια διαφορά τους είναι η απόδοση που μπορεί να έχουν, η οποία όμως σχετίζεται με την αύξηση της πολυπλοκότητας στους υπολογισμούς που θα χρειαστούν καθώς επίσης και ποιες από τις παραπάνω πληροφορίες θα εκμεταλλευτούν.

Για την εύρεση της θέσης ενός unknown node στο τρισδιάστατο χώρο (\mathbb{R}^3) το Minimal Required number of Sensors (MRS) είναι τέσσερα nodes. Στις παρακάτω περιγραφές όμως, και χωρίς βλάβη της γενικότητας θα χρησιμοποιηθούν τρία για την απλούστευση της περιγραφής - με απότερο σκοπό την εύκολη κατανόηση της, με την παραδοχή ότι μας ενδιαφέρει η δισδιάστατη ανάλυση (\mathbb{R}^2). Ενώ στο Κεφάλαιο 4 στο οποίο γίνεται και ο σχεδιασμός του συστήματος - βάση συγκεκριμένου αλγόριθμου - θα γίνει η ανάλυση στους τρεις άξονες.

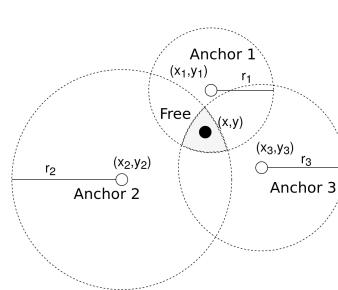
Trilateration

Η συγκεκριμένη μέθοδος είναι ίσως η πιο απλή διαισθητικά, βασίζεται στην γεωμετρία κύκλων, είναι αυτή που χρησιμοποιείται από τα GPS για τον υπολογισμό θέσης [89], ενώ αξιοποιεί μόνο πληροφορία απόστασης και όχι γωνίας [90]. Η εξίσωση ενός κύκλου από την γεωμετρία γνωρίζουμε ότι περιγράφεται από την εξίσωση (3.11) όπου (x_i, y_i) οι συντεταγμένες του κέντρο του κύκλου με ακτίνα r_i .

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 \quad (3.11)$$



(a) Ideal Trilateration



(b) Realistic Trilateration

ΕΙΚΟΝΑ 3.9: Παραδείγματα του Trilateration

Εάν χρησιμοποιούμε *omnidirectional* κεραίες [91] ή στην γενικότερη περίπτωση το μοντέλο της *isotropic antenna* [92] και υπολογίσουμε την απόσταση ενός beacon

από το node στο οποίο θέλουμε να υπολογίσουμε την θέση του - τότε μπορούμε να συμπεράνουμε ότι το free node είναι κάπου πάνω στην περιφέρεια ενός κύκλου, με κέντρο το beacon και ακτίνα την απόσταση μεταξύ του beacon και του free node. Επαναλαμβάνοντας αυτό για ακόμα δύο beacons, τελικά το free node στο δισδιάστατο επίπεδο (\Re^2) θα πρέπει να βρίσκεται στην τομή των τριών κύκλων, πράγμα που γραφικά απεικονίζεται στην *Εικόνα 3.9* (a) [93].

$$x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 = r_1^2 \quad (3.12)$$

$$x^2 - 2x_2x + x_2^2 + y^2 - 2y_2y + y_2^2 = r_2^2 \quad (3.13)$$

$$x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 = r_3^2 \quad (3.14)$$

Οι εξισώσεις (3.12), (3.13) και (3.14) περιγράφουν πλήρως τους κύκλους του κάθε node από το παράδειγμα που έχουμε στην *Εικόνα 3.9* (a), όπου (x_i, y_i) το κέντρο του κύκλου, r_i η ακτίνα του - για κάθε beacon $i = 1, 2, 3$ και τελικά (x, y) οι συντεταγμένες του free nodes τις οποίες και ψάχνουμε. Ένας τρόπος να υπολογίσουμε τις συντεταγμένες αυτές είναι να αφαιρέσουμε από την (3.13) την (3.12) και όμοια από την (3.14) την (3.13) ώστε να καταλήξουμε στις παρακάτω δύο εξισώσεις [94] [95].

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2$$

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y = r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2$$

Ο λόγος που κάναμε το παραπάνω βήμα είναι διότι πλέον έχουμε ένα σύστημα με δύο εξισώσεις και δύο αγνώστους, οπότε μπορούμε εύκολα να θεωρήσουμε τους παρακάτω πίνακες.

$$A = \begin{bmatrix} -2x_1 + 2x_2 & -2y_1 + 2y_2 \\ -2x_2 + 2x_3 & -2y_2 + 2y_3 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \end{bmatrix} \quad B = \begin{bmatrix} r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{bmatrix}$$

Με την χρήση των οποίων καταλήγουμε ότι έχουμε να λύσουμε το γραμμικό σύστημα πινάκων που περιγράφεται από την εξίσωση (3.15) για τον υπολογισμό των συντεταγμένων (x, y) του free node οι οποίες μας ενδιαφέρουν.

$$AX = B \quad (3.15)$$

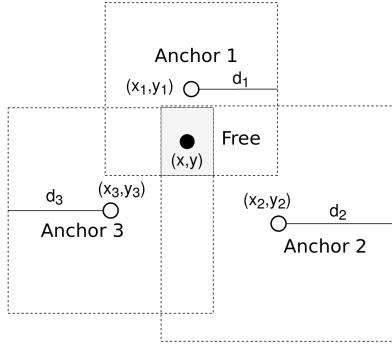
Όπως αναφέρθηκε και στις προηγούμενες ενότητες, ο υπολογισμός της απόστασης πολύ πιθανόν να περιλαμβάνει μια μικρή απόκλιση $\hat{r}_i = r_i - \varepsilon$, με το ε συχνά να θεωρείται μία ανεξάρτητη κανονική τυχαία μεταβλητή με μηδενικό μέσο. Αυτό σημαίνει ότι τότε οι κύκλοι δεν έχουν ένα κοινό σημείο τομής, αλλά το free node βρίσκεται κάπου μέσα στο χωρίο επικάλυψης των τριών κύκλων, σχηματικά αυτό παρουσιάζεται στην *Εικόνα 3.9 (b)* και σε αυτήν την περίπτωση καταλήγουμε σε ένα μη πεπερασμένο πλήθος λύσεων, με τη συνάρτηση του κάθε κύκλου να περιγράφεται από την εξίσωση (3.16) [70].

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 - e \quad (3.16)$$

Το αρνητικό με αυτήν την μέθοδο, είναι η ανάγκη πραγματοποίησης floating point operations για τον υπολογισμό των συντεταγμένων (x, y) σε πραγματικές συνθήκες - όπου το πλήθος των οπίων εξαρτάται από τον τρόπο που θα επιλέξουμε να επιλύσουμε το σύστημα. Μία από τις μεθόδους επίλυσης της γραμμικής εξίσωσης είναι μέσω του least square method, όπου τότε το πλήθος των floating point operations που απαιτούνται είναι $(m + \frac{n}{3}) \cdot n^2$, με m τον αριθμό των αγνώστων και n τον αριθμό των δοθέντων εξισώσεων [70].

Bounding Box

Σε αυτήν την μέθοδο χρησιμοποιούνται τετράγωνα αντί για κύκλους του Trilateration, ενώ και εδώ θεωρούμε (x_i, y_i) τις συντεταγμένες των beacon και d_i η απόσταση που έχουμε υπολογίσει από το free node - για κάθε beacon i .

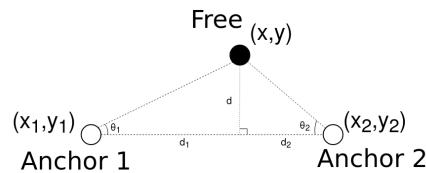


ΕΙΚΟΝΑ 3.10: Bounding Box παράδειγμα

Δημιουργούμε τετράγωνα μήκος πλευράς $2d_i$ με κέντρο το κέντρο του beacon και συντεταγμένες $(x_i - d_i, y_i - d_i)$ & $(x_i + d_i, y_i + d_i)$. Θετικό πλέον είναι ότι δεν χρειάζεται να κάνουμε floating point operations για τον υπολογισμό του χωρίου τομής - αλλά μπορούμε να το υπολογίσουμε με απλή γεωμετρία. Αφού έχουμε υπολογίσει το χωρίο τομής των τετραγώνων μπορούμε να θεωρήσουμε ότι στο κέντρο του βρίσκεται το free node. Παράδειγμα αυτής της μεθόδου βρίσκεται στην Εικόνα 3.10. Η συγχεκριμένη μέθοδος μπορεί να είναι ευκολότερη υπολογιστικά και να απαιτεί λιγότερα processor resources από το Trilateration, όμως ταυτόχρονα προκύπτει και μεγαλύτερο σφάλμα απόκλισης [70].

Triangulation

Σε αντίθεση με τις παραπάνω μεθόδους, η τεχνική Triangulation εκτιμάει την θέση του node που μας ενδιαφέρει, χρησιμοποιώντας γνώση που έχουμε για γωνίες και όχι εκτίμηση της απόστασης. Ένα απλούστερο παράδειγμα παρουσιάζεται στην Εικόνα 3.11, όπου σε αυτό το παράδειγμα από την τριγωνομετρία, μπορούμε εύκολα να συμπεράνουμε ότι για τις γωνίες θ_1 και θ_2 ισχύουν οι σχέσεις (3.17).



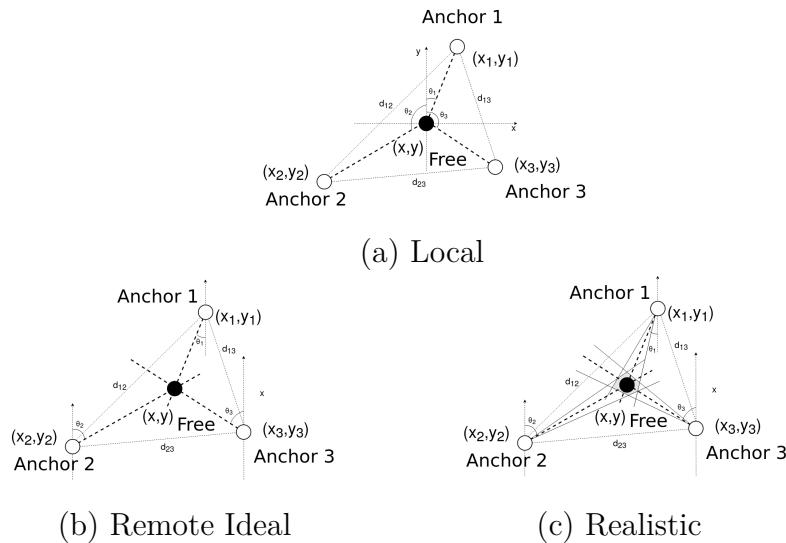
ΕΙΚΟΝΑ 3.11: Principles of Triangulation

$$\theta_1 = \frac{d}{d_1} = \tan^{-1} \left(\frac{y_1 - y}{x_1 - x} \right) \quad \theta_2 = \frac{d}{d_2} = \tan^{-1} \left(\frac{y_2 - y}{x_2 - x} \right) \quad (3.17)$$

Από τις σχέσεις (3.17) μπορούμε να καταλήξουμε στις σχέσεις (3.18) [96], μέσω των οποίων τελικά για δεδομένες θέσεις των beacon - γνωστά δηλαδή τα (x_i, y_i) για κάθε beacon i - και με βάση την μέτρηση των γωνιών θ_1 και θ_2 , να υπολογίσουμε την θέση του free node (x, y) .

$$x = \frac{y_2 - y_1 - x_2 \tan \theta_2 + x_1 \tan \theta_1}{\tan \theta_1 - \tan \theta_2} \quad (3.18)$$

$$y = x \tan \theta_1 + y_1 - x_1 \tan \theta_1 = x \tan \theta_2 + y_2 - x_2 \tan \theta_2$$



ΕΙΚΟΝΑ 3.12: Triangulation παραδείγματα

Αυτή η τεχνική εκτίμησης της θέσης του node, μπορεί να γίνει είτε από τοπικές μετρήσεις του ίδιου του node - *Εικόνα 3.12 (a)* - είτε από μετρήσεις γωνιών που κάνουν τα anchors - *Εικόνα 3.12 (b)* [70]. Φυσικά σε μία πραγματική εφαρμογή, όπως έχει αναφερθεί παραπάνω, οι μετρήσεις παρουσιάζουν αποκλίσεις από τα ιδανικά μοντέλα. Συνεπώς την *Εικόνα 3.12 (c)* παρουσιάζει σε πιο ρεαλιστικά επίπεδα - συνυπολογίζοντας την ύπαρξη των αποκλίσεων - το χωρίο στο οποίο με μεγαλύτερη πιθανότητα βρίσκεται το node.

Multilateration

Η μέθοδος του Multilateration⁶ (**MLAT**), έχει ως βάση την γεωμετρία των υπερβολικών καμπύλων και συχνά μπορεί να βρεθεί επίσης και ως hyperbolic positioning στην

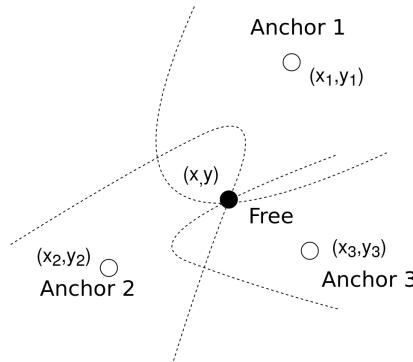
⁶To Multilateration μπορεί να αναφερθεί τόσο σε time-synchronous όσο και σε non-synchronous προσέγγιση [97]. Για την time-synchronous ο όρος αυτός χρησιμοποιείται επίσης σε αναφορά της χρήσης του Trilateration ή του Triangulation με πάνω από 3 beacons [70] [96]. Όμως εδώ συζητάτε η γενικότερη που είναι η non-synchronous.

βιβλιογραφία [98] [99] [100]. Στη Παράγραφο 3.1.1 έγινε αναφορά ότι η χρήση του **TDoA** βασίζεται στις υπερβολικές καμπύλες, συνεπώς μπορούμε εύκολα να χρησιμοποιήσουμε τις μετρήσεις αυτές για τον υπολογισμό της θέσης του free node.

Η γενική ιδέα σε αυτήν την περίπτωση περιγράφεται στην εξίσωση (3.19) [71] [101] - και αυτό που θέλουμε, είναι να υπολογίσουμε την θέση του k_t (free node) γνωρίζοντας για κάθε δύο διαφορετικά k_i και k_j beacon nodes την χρονική στιγμή t_i και t_j που έφτασε το σήμα στο καθένα - εάν αυτό κινούταν με ταχύτητα s και το $\|\cdot\|$ συμβολίζει την ευκλείδεια απόσταση μεταξύ τους.

$$\Delta t_{ij} \triangleq t_i - t_j = \frac{1}{s} (\|k_i - k_t\| - \|k_j - k_t\|), \quad i \neq j \quad (3.19)$$

Αυτό που αναπαριστά ουσιαστικά η σχέση (3.19) δεν είναι τίποτα άλλο από την προσπάθεια εύρεσης του σημείου τομής των καμπύλων. Παράδειγμα αυτής της μεθόδου βρίσκεται στην Εικόνα 3.13.



ΕΙΚΟΝΑ 3.13: Multilateration παραδείγματα

Πιθανοκρατικοί μέθοδοι

Το γεγονός ότι οι μετρήσεις απόστασης και γωνίας σε πραγματικές συνθήκες εμπεριέχουν σφάλματα, έχει ωθήσει στην έρευνα probabilistic μεθόδων εκτίμησης τοποθεσίας των free nodes. Αυτού του τύπου οι προσεγγίσεις λαμβάνουν υπόψιν τα μετρικά σφάλματα και συχνά τα μοντελοποιούν ως κανονικές τυχαίες μεταβλητές. Το μεγάλο μειονέκτημα σε αυτήν την μέθοδο είναι τα μεγάλα υπολογιστικά και χωρικά - για την αποθήκευση των δεδομένων - κόστη [70].

ΠΙΝΑΚΑΣ 3.2: Σύνοψη τεχνικών εκτίμησης απόστασης/γωνίας και υπολογισμού θέσης [102]

Method	Position Computation	Advantages	Disadvantages	Applications
RSSI	Trilateration, Bounding box	Simple, inexpensive, no need for extra hardware, synchronization not needed	Multipath interference, NLoS and environment can affect readings	In sort-distance with LoS applications and low accuracy needed (e.g. indoor)
ToA	Trilateration, Bounding box	High accuracy	Need to precisely measure time, LoS is normally assumed, more sophisticated design	Common in cellular networks or systems assisted by GPS
TDoA	Multilateration	High accuracy	Need for extra hardware or precisely synchronized nodes	Common in WSN or systems assisted by GPS
AoA	Triangulation	MRS is two nodes (Ideal \mathbb{R}^2), synchronization not needed	LoS is normally assumed, smart antenna design, multipath can affect readings	Common in radar scenarios

Στο Πίνακα 3.2 γίνεται σύνοψη των μεθόδων εκτίμησης απόστασης που αναλύθηκαν παραπάνω, μαζί με τις μεθόδους εκτίμησης θέσης που συχνά συνδυάζονται, καθώς και θετικά/αρνητικά όπως και συχνές τους εφαρμογές.

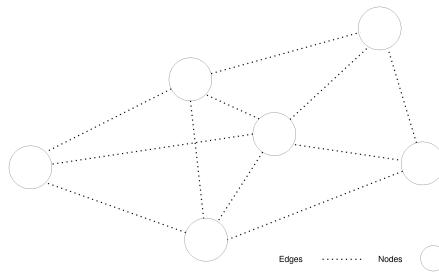
3.1.3 Localization Αλγόριθμοι

Στις προηγούμενες ενότητες αναφέρθηκαν κάποιες από τις βασικές τεχνικές που χρησιμοποιούνται, προκειμένου να εκτιμηθεί η θέση ενός node στον χώρο. Σε ένα πραγματικό σύστημα όμως, πρέπει να συνυπολογιστούν και άλλοι παράγοντες πέρα από - απλά - τον υπολογισμό απόστασης/γωνίας για να καταλήξουμε να αποκτήσουμε location information. Για αυτό το λόγο, στο παρόν Section παρουσιάζονται επιπλέον πιτυχές - του συστήματος - που πρέπει να λάβουμε υπόψιν· καθώς και αναφορά αλγόριθμων από την υπάρχουσα βιβλιογραφία, με τους οποίους μπορούμε να καλύψουμε αυτές τις ανάγκες.

Αφού έχουμε δει τα παραπάνω, αρχικά ας ορίσουμε το localization problem με ένα πιο αυστηρό μαθηματικό φορμαλισμό. Έστω ότι έχουμε στην διάθεση μας n αριθμό από nodes, και για λόγους απλότητας* συμμετρικά και όμοια δίκτυα επικοινωνίας - με εμβέλεια r - για κάθε node. Συνεπώς, ένα node u επικοινωνεί με το v αν και μόνο αν, και το v μπορεί να επικοινωνήσει πίσω στο u , τα οποία είναι διασκορπισμένα σε ένα δισδιάστατο τετραγωνικό πεδίο - και εδώ γίνεται η θεώρηση για \mathbb{R}^2 ανάλυση - $Q = [0, s] \times [0, s]$.

Τότε μπορούμε να μοντελοποιήσουμε το δίκτυο από nodes, ως ένα μη κατευθυνόμενο γράφο $\mathcal{G} = (V, E)$ - παράδειγμα του οποίου υπάρχει στην Εικόνα 3.14 - με τα παρακάτω χαρακτηριστικά [70].

- $V = \{v_1, v_2, \dots, v_n\}$ το σύνολο των nodes, όπου έχει την έννοια των κορυφών του γράφου, με $|V| = n^7$.
- $\langle i, j \rangle \in E$ εάν το u_i μπορεί να επικοινωνήσει με το v_j . Πράγμα που σημαίνει - με βάση τα παραπάνω - ότι η ευχλείδεια απόσταση τους⁷ είναι μικρότερη από r και έχει την έννοια των ακμών του γράφου, με $|E| = m^7$.
- $0 \leq w(e) \leq r$ με $e = \langle i, j \rangle$, να είναι το βάρος της κάθε ακμής και να χρησιμοποιείται για την έννοια της απόστασης d_{ij} μεταξύ του node u_i με το v_j .



ΕΙΚΟΝΑ 3.14: Παράδειγμα γράφου

One-hop vs Multi-hop

Η πρώτη κατηγοριοποίηση στην οποία θα αναφερθούμε - για αυτό το δίκτυο γράφου - προκειμένου να εκτιμήσουμε την θέση των nodes, είναι από ποιους κόμβους τελικά θα αξιοποιήσουμε πληροφορία. Στους αλγόριθμους **One-hop** αξιοποιείται πληροφορία, μόνο από άμεσους γείτονες των nodes^{*} προκειμένου να υπολογίσουμε την θέση του κόμβου στο χώρο. Αντίθετα στις τεχνικές **Multi-hop**, χρησιμοποιούμε και πληροφορία που λαμβάνουμε με έμμεσο τρόπο^{*} από τους γείτονες των γειτόνων μας - προκειμένου να αποφανθούμε για την θέση ενός node.

Anchor-based vs Anchor-free

Ένας ακόμα σημαντικός διαχωρισμός είναι η ύπαρξη ή όχι τελικά στο δίκτυο - από anchor nodes. Καθώς υπάρχουν εφαρμογές στις οποίες μπορεί να μην χρειάζεται ή να μην έχουμε δυνατότητα να τοποθετηθούν beacons - ακόμα και αν αναφερόμαστε σε mobile beacons - οπότε τότε θα χρησιμοποιήσουμε **Anchor-free** αλγορίθμους. Αν υπάρχει η δυνατότητα χρήσης τους, τότε ενδιαφερόμαστε για **Anchor-based**.

⁷Ο παραπάνω συμβολισμός $|\cdot|$ χρησιμοποιείται ως το cardinality του εκάστοτε set

Relative vs Absolute Positioning

Ανάλογα με τις προδιαγραφές της εφαρμογής που θέλουμε να καλύψουμε, θα πρέπει να χρησιμοποιήσουμε ανάλογες εκτιμήσεις θέσεων. Υπάρχουν εφαρμογές, όπως για παράδειγμα - εκτίμηση θέσης των nodes ενός κινούμενου swarm^{*} για χαρτογράφηση σε άγνωστο terrain. Σε αυτού του τύπου τις εφαρμογές μας ενδιαφέρει η απόλυτη θέση στον χώρο^{*} οπότε θέλουμε μεθοδολογίες για **Absolute Positioning**, ενώ υπάρχουν άλλες εφαρμογές στις οποίες μας ενδιαφέρει μόνο πληροφορία θέσης σε σχέση με τα υπόλοιπα στοιχεία του περιβάλλοντος της εφαρμογής, οπότε χρειαζόμαστε **Relative Positioning**.

Indoor vs Outdoor Scenarios

Κάτι ακόμα που πρέπει να προσέξουμε είναι το περιβάλλον στο οποίο θα βρίσκεται η εφαρμογή, καθώς ανάλογα με το περιβάλλον^{*} έχουμε να διαχειριστούμε και τις δυσκολίες που μας παρουσιάζει - εμπόδια, ανακλάσεις σημάτων, κλπ. Με δύο κύριους διαχωρισμούς, τα **Indoor Scenarios** και τα **Outdoor Scenarios**.

Distributed vs Centralized Position Computation

Πρέπει να αναφερθεί επίσης, ότι είναι πολύ σημαντικό να είναι ξεκάθαρο το σημείο^{*} στο οποίο θα γίνεται η επεξεργασία της πληροφορίας. Υπάρχουν δύο ενδεχόμενα. Το πρώτο είναι, ένα BS να αντιλαμβάνεται και να υπολογίζει την θέση των nodes, τα οποία έπειτα θα ενημερώνονται μέσω αυτού^{*} για την θέση τους - οπότε αναφερόμαστε για **Centralized** τεχνικές. Αντίθετα άλλη εναλλακτική είναι να γίνει τοπικά η επεξεργασία. Σε αυτή την περίπτωση, αναφερόμαστε για **Decentralized** ή **Distributed** αλγορίθμους^{*} όπου κάθε node θα επικοινωνεί από μόνο του με τα γειτονικά του για την απόκτηση πληροφορίας^{*} και έπειτα υπολογίζει μεμονωμένα την θέση του. Το μεγάλο πλεονέκτημα των Centralized αλγορίθμων είναι ότι απομακρύνουμε το πρόβλημα του computation από το κάθε node, όμως την ίδια στιγμή προκύπτουν δυσκολίες όπως καθυστερήσεις στις επικοινωνίες, μεγαλύτερο power consumption & bandwidth, ενώ τέλος και προβλήματα με το scalability του συστήματος - καθώς προτείνεται για μικρότερα networks. Το οποίο όμως σε ένα βαθμό^{*} μπορεί να λυθεί με το να χρησιμοποιηθούν πολλαπλά BSs, συνεπώς να έχουμε ένα multi-tier network. Γνωστοί Centralized αλγόριθμοι είναι ο Multi Dimensional Scaling-Mobile Assisted Programming ([MDS-MAP](#)), Semi Definite Programming ([SDP](#)) και Localization Based on Simulated Annealing ([LBSA](#)) [103]. Παρακάτω στον *Άλγοριθμο 1* - ως παράδειγμα χρήσης των *Centralized* αλγορίθμων - παρουσιάζεται ο pseudo-code του [MDS-MAP](#).

Algorithm 1 Multi Dimensional Scaling-Mobile Assisted Programming [84]

-
- procedure** MDS-MAP ($r_{i,j} = w(e) \quad \forall e \in E$)
- 1: Form a sparse matrix R from $r_{i,j}$
 - 2: Run a standard all pairs shortest path algorithm on R to produce inter-node distances D (e.g. Dijkstra's, Floyd's)
 - 3: Run classical metric MDS on D to find estimated positions X
 - 4: Transform the solution X into global coordinates G
-

ΠΙΝΑΚΑΣ 3.3: Σύγχριση μεταξύ Centralized και Distributed [104]

	Centralized	Distributed
Accuracy	Data collection	Position merging
Energy consumption	Signal transition	Position merging
Coverage area	Network topology	Anchor deployment
Costs	Central module	Anchor equipment

Range-based vs Range-free

Στη Παράγραφο 3.1 αναλύθηκαν τεχνικές εκτίμησης απόστασης ή γωνίας μεταξύ των nodes - συγκεκριμένα οι RSSI, ToA, TDoA και AoA. Σε αυτήν την περίπτωση, όταν χρησιμοποιούμε δηλαδή έξτρα hardware ή γενικά εκτεταμένες ranging τεχνικές για να καταλήξουμε στις θέσεις των free nodes στον χώρο, αναφερόμαστε σε **Range-based** μεθόδους. Θετικό σε αυτήν την περίπτωση είναι ότι έχουμε μεγαλύτερη ακρίβεια της θέσης, όμως ταυτόχρονα πιθανόν να χρειαστούμε ακριβότερα nodes - λόγο των έξτρα components ή του πιο *sophisticated hardware*.

Για να το αποφύγουμε αυτό, μπορούμε να χρησιμοποιήσουμε τεχνικές οι οποίες ονομάζονται **Range-free**. Αυτές οι τεχνικές δεν απαιτούν κανένα επιπρόσθετο hardware, διότι δεν χρησιμοποιούν εκτιμήσεις απόστασης/γωνίας, συνεπώς έχουμε μικρότερο κόστος ανά node. Έχουν αρκετό ενδιαφέρον λόγω της απλότητας τους παρά την μικρότερη ακρίβεια που παρέχουν. Γνωστοί Range-free αλγόριθμοι είναι ο Approximate Point in Triangle (APIT), Distance Vector-Hop (DV-Hop), Centroid και Gradient [103]. Όμοια με παραπάνω, στους Αλγόριθμο 2 και 3 παρουσιάζονται οι pseudo-codes των APIT και DV-Hop αντίστοιχα.

Algorithm 2 Approximate Point in Triangle [84]

procedure APIT

- 1: Receive beacon positions from hear-able beacons.
 - 2: Initialize inside-set to be empty
 - 3: **for** each triangle T_i in possible triangles formed over beacons **do**
 - 4: Add T_i to inside-set if node is inside T_i . Go to next Step when accuracy of inside-set is sufficient.
 - 5: Compute position estimate as the center of mass of the intersection of all triangles in inside-set.
-

Algorithm 3 Distance Vector-Hop [105] [106] [107]

procedure DV-Hop

- 1: **for** Each anchor **do**
 - 2: Broadcast a message with it's ID, location and hop count initialized to 0
 - 3: **for** all nodes received all messages from neighbors **do**
 - 4: Neighbor nodes record the node ID, the coordinate values, and the smallest hop. Then forward packet after incrementing hop by one
 - 5: **for** each anchor i to each anchor j **do**
 - 6: Estimate Average Hop Distance ([AHD](#))
$$HopSize_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum hop_{i,j}}, i \neq j$$
 - 7: Broadcast [AHD](#) to network
 - 8: **for** each free node u **do**
 - 9: Save the first [AHD](#) you hear, then broadcast it to your neighbors
 - 10: Calculate your distance from your closest anchor k, $d_{uk} = HopSize_i \times hop_{uk}$
 - 11: Based on known beacons' coordinates and estimated distances, use one of the methods described in *Section 3.1.2* to compute free nodes locations
-

ΠΙΝΑΚΑΣ 3.4: Σύγχριση μεταξύ Range-based και Range-free [104]

	Range-based	Range-free
Accuracy	Ranging algorithm	Geometric algorithm
Energy consumption	Signal transition	Instruction execution
Coverage area	Signal cover	Network topology
Costs	Ranging module	Execution module

Σε ευρύτερη κλίμακα, στην βιβλιογραφία υπάρχουν επιπλέον αλγόριθμοι ή παραλλαγές χρήσης αυτών που αναφέρονται, για την εύρεση τοποθεσίας nodes σε ένα

WSN. Παρόλα αυτά, σε αυτό το σημείο έγινε μία συνοπτική αναφορά τους, και μία πρώτη ανάλυση γύρω από το localization problem. Στην συνέχεια, με βάση αυτές τις αρχές⁸ θα γίνει η μοντελοποίηση του συστήματος που συσχετίζεται αυτή η διπλωματική.

3.2 Εκτίμηση Θέσης με οπτικούς τρόπους

Άλλος ένας τρόπος για να καταλήξουμε να εκτιμήσουμε την θέση ενός αντικειμένου στον χώρο⁹ είναι με την βοήθεια καμερών και οπτικών τεχνολογιών. Κύριος περιοριστικός παράγοντας σε αυτές τις προσεγγίσεις, είναι η ανάγκη για οπτική επαφή με το αντικείμενο ενδιαφέροντος. Ενώ, καταλήγει να είναι και ο τρόπος με τις μεγαλύτερες ανάγκες επεξεργασίας. Βασικοί τρόποι απόκτησης 3D πληροφορίας - με χρήση 2D, που προσφέρει τυπικά μία κάμερα⁸ - είναι μέσω τεχνικών με όνομα structure from reference, structure from motion και με χρήση stereo [108] οι οποίες συνοπτικά αναλύονται στην συνέχεια.

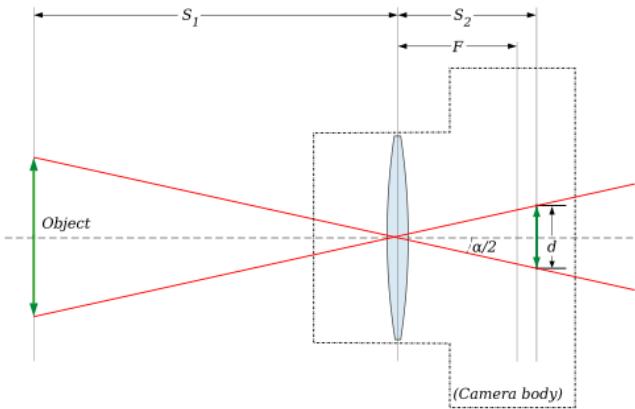
3.2.1 Structure from Reference

Τυπικά, κατά τη μεταφορά των σημείων του πραγματικού χόσμου 3D σε pixel μίας εικόνας 2D χάνουμε την διάσταση του βάθους, πράγμα που μπορεί να δημιουργήσει πλάνες/αυταπάτες - παράδειγμα αυτού στην *Eικόνα 3.15*. Ένας τρόπος για την επανάκτηση της πληροφορίας του depth - προκειμένου να εκτιμήσουμε μέσω εικόνας την θέση των αντικειμένων - είναι με το να υπάρχουν στο scene αντικείμενα γνωστών διαστάσεων, τα οποία θα χρησιμοποιηθούν ως reference objects.

⁸Σε αντίθεση με τις RGB-D, οι οποίες εκτός από color information, παρέχουν και depth information για κάθε pixel τους



EIKONA 3.15: Optical illusion που δημιουργείται λόγο του depth ambiguity ([URL](#))



EIKONA 3.16: Optical axis του αντικειμένου - κάμερας ([URL](#))

Στην *Eikόνα 3.16* παρουσιάζονται οι νοητές ευθείες πορείας του φωτός στο thin lens model, από τα σημεία του τρισδιάστατου κόσμου, προς τον αισθητήρα της κάμερας. Όπου S_1 η απόσταση του αντικειμένου από την κάμερα σε m , S_2 η απόσταση του image plane για το pinhole model (περισσότερες πληροφορίες στη *Παράγραφο 4.3*) σε mm , F το focal length σε mm , d το μέγεθος του αντικειμένου στην εικόνα σε mm και H το μέγεθος του αντικειμένου στον πραγματικό κόσμο σε m . Επίσης, θεωρούμε ότι το αντικείμενο είναι σε focus, συνεπώς $F = S_2$. Τότε, με χρήση τριγωνομετρίας και όμοιων τριγώνων τόσο από την μεριά του αισθητήρα, όσο και του αντικειμένου, καταλήγουμε στην σχέση (3.20) [109] [110] [111].

$$\frac{d}{F} = \frac{H}{S_1} \quad (3.20)$$

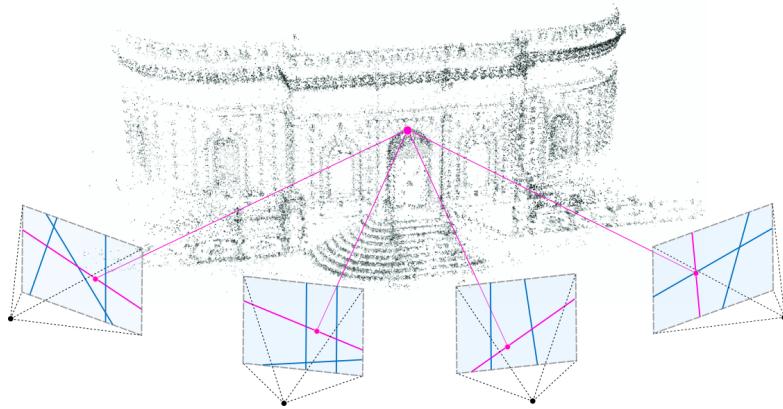
Από την οποία καταλήγουμε στην *Εξίσωση (3.21)*. Όπου στο μέγεθος του αντικειμένου μπορούμε να χρησιμοποιήσουμε οποιαδήποτε διάσταση επιθυμούμε, όπως το ύψος, το πλάτος ή ακόμα και κάποια διαγώνιο του.

$$\text{Distance to object } S_1 \text{ (m)} = \text{Focal length } F \text{ (mm)} \frac{\text{Object real size } H \text{ (m)}}{\text{Object size on sensor } d \text{ (mm)}} \quad (3.21)$$

Σε γενικές γραφικές η αναδόμηση της τρίτης διάστασης από single view ακόμα και με ύπαρξη γνωστών αντικειμένων στο scene, μπορεί να καταλήξει να είναι μία εξαιρετικά ασαφής διαδικασία. Για αυτό τον λόγο προσπαθούμε είτε μέσω monocular vision να πάρουμε πολλαπλά views ενός κινούμενου αντικειμένου και να χρησιμοποιήσουμε τεχνικές με όνομα Structure from Motion (βλ. Παράγραφο 3.2.2), είτε από πολλαπλές κάμερες - stereo vision (βλ. Παράγραφο 3.2.3) - να λάβουμε πολλαπλά views στατικού αντικειμένου για την εκτίμηση τελικά του depth.

3.2.2 Structure from Motion

Αρχή λειτουργίας της μεθόδου Structure from Motion, είναι η εξαγωγή features από ένα image frame και το tracking τους κατά την διάρκεια κίνησης της κάμερας. Με αυτόν τον τρόπο, στο τέλος της διαδικασίας - για το δούθεν sequence of frames - συλλέγεται ένα σύνολο σημείων για κάθε feature $p_f = (x_f, y_f)$ στο image plain της κάμερας (για κάθε frame f) τα οποία αναπαριστούν το ίδιο σημείο στο φυσικό κόσμο $P = (x, y, z)$. Η Εικόνα 3.17 αναπαριστά παράδειγμα αυτής της διαδικασίας. Αφού συλλεχθούν τα παραπάνω δεδομένα, μπορούν να περάσουν από μία πληθώρα γκάμα αλγορίθμων με ομώνυμο όνομα, μέσω των οποίων μπορεί να προσεγγιστεί η τρισδιάστατη μορφή του αντικειμένου, η κίνηση της κάμερας ή άλλες σημαντικές πληροφορίες.

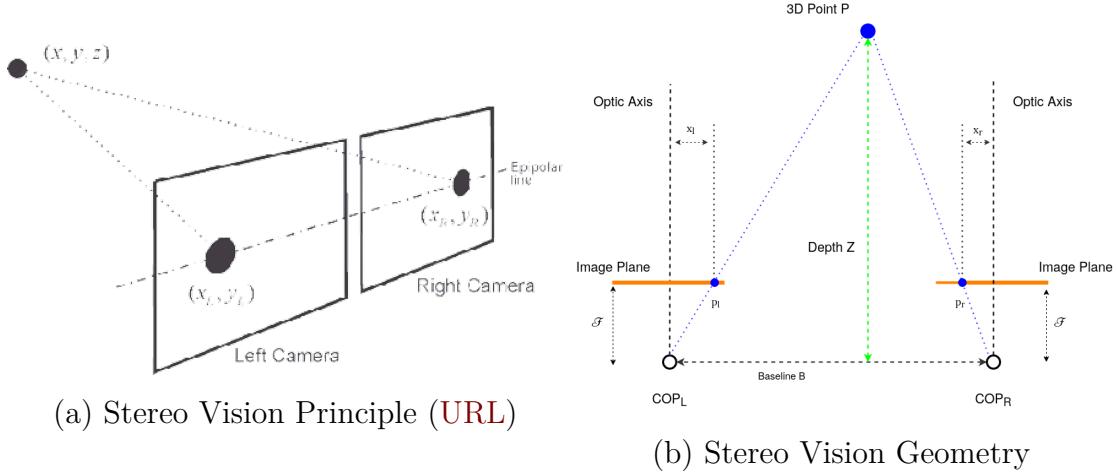


ΕΙΚΟΝΑ 3.17: Τυπικό παράδειγμα κίνησης εικόνας για ανακατασκευή μέσω Structure from Motion της σκηνής ([URL](#))

3.2.3 Stereo Vision

Για τον υπολογισμό του depth με χρήση stereo vision, λαμβάνουμε από δύο κάμερες views ενός scene. Σε γενίκευση τα image planes μπορεί να έχουν οποιαδήποτε προσανατολισμό, ειδική περίπτωση - και συχνά πιο χρησιμοποιούμενη - είναι αυτή

που υπάρχει στην *Eikόνα 3.18* (a) στην οποία είναι παράλληλα. Προκειμένου να εκτιμήσει το depth με χρήση stereo γίνονται υπολογισμοί βασισμένοι στο Epipolar Geometry [112].



EIKONA 3.18: Stereo Vision

Η *Eikόνα 3.18* (b) παρουσιάζει την γεωμετρία στην περίπτωση των parallel optical axis. Μέσω σχέσεων των όμοιων τριγώνων (p_l, P, p_r) και (COP_L, P, COP_R) μπορούμε να δημιουργήσουμε την σχέση (3.22), από την οποία τελικά καταλήγουμε στην *Εξίσωση* (3.23) για τον υπολογισμό του depth ενός Point [113].

$$\frac{B - x_l + x_r}{Z - F} = \frac{B}{Z} \Rightarrow \quad (3.22)$$

$$Z = F \frac{B}{x_l - x_r} \quad (3.23)$$

Η ποσότητα $x_l - x_r$ ονομάζεται Disparity, και ουσιαστικά μετρώντας την διαφορά της θέσης που εμφανίζεται στο κάθε plane το point, υπολογίζουμε την απόσταση του σημείου από το baseline των καμερών [113].

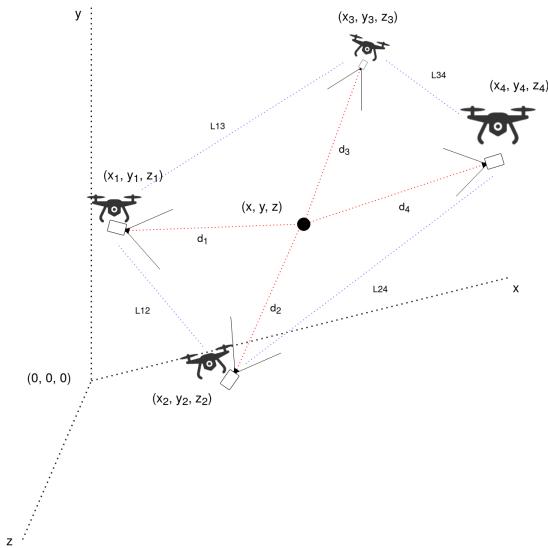
3.3 Σκοποί Διπλωματικής

Σκοπός της παραπάνω βιβλιογραφικής αναζήτησης· ήταν η απόκτηση μίας σφαιρικής γνώσης σχετικά με τους δυνατούς τρόπους επίλυσης του προβλήματος εντοπισμού της θέσης αντικειμένων, ώστε να πραγματοποιηθεί μία αρχική εκτίμηση της πολυπλοκότητας και του κόστους hardware που απαιτεί η κάθε μία, προκειμένου τελικά

να γίνει η επιλογή της μεθοδολογίας με την οποία θα υλοποιηθεί στην συγκεκριμένη εργασία· η επίλυση του προβλήματος ενδιαφέροντος.

Φυσικοί περιορισμοί, προκύπτουν από τις δυνατότητες του διαθέσιμου εξοπλισμού που υπάρχουν στο **Πολυτεχνείο Κρήτης**, βάση των οποίων θα γίνει και ο σχεδιασμός του συστήματος. Πράγμα που καθιστά το σύστημα να ακολουθεί ρεαλιστικά requirements, τον σχεδιασμό με γνώμονα την επίλυση ενός πραγματικού προβλήματος, όπως επίσης και την δυνατότητα - με αυτόν τον τρόπο - να υπάρχουν δυνατότητες δοκιμών από αυτά, σε περίπτωση που χρειαστεί.

Ακολουθώντας την λογική των MoCap system (βλ. *Παράγραφο 2.3.4*). Στην συγκεκριμένη διπλωματική εργασία γίνεται μία πρώτη προσπάθεια επίλυσης του προβλήματος υπολογισμού της θέσης ενός μεμονωμένου - γνωστών διαστάσεων - αντικειμένου με χρήση drones swarm που πραγματοποιεί close formation flight, όπως παρουσιάζεται στην *Εικόνα 3.19* με σκοπό μελλοντικά να χρησιμοποιηθούν ως feature points και να είμαστε πιο κοντά σε ένα Cooperative Localization (**CL**) Real Time Location System (**RTLS**) για εντοπισμό και ανίχνευση της θέσης αντικειμένων σε εξωτερικούς χώρους και δυναμικά περιβάλλοντα.



ΕΙΚΟΝΑ 3.19: Thesis drone swarm system overview

Παρόμοια προβλήματα προσπαθούν να λύσουν στην έρευνα τους [114] οι οποίοι με ένα μεμονωμένο drone προσπαθούν να πραγματοποιήσουν Human MoCap. Επίσης, οι συγγραφείς του [115] χρησιμοποιούν τρία quad-copter για να πραγματοποιήσουν Localization and Tracking ενός αντικειμένου. Στην συγκεκριμένη διπλωματική δίνεται έμφαση όμως στα παρακάτω.

Το σύστημα που σχεδιάζεται στα επόμενα κεφάλαια, είναι προσανατολισμένο να χρησιμοποιείται από Multirotor drones (βλ. *Παράγραφο 1.1.1*), τα οποία θα κινούνται με σχετικά χαμηλές ταχύτητες και μπορούν να πραγματοποιήσουν hover σε αυθαίρετες θέσεις περιφερειακά του αντικειμένου που θέλουμε να εντοπίσουμε στον 3D space, χωρίς την a priori ανάγκη γνώσης της θέσης τους, παρόλα αυτά το rotation τους (βλ. *Παράγραφο 1.1.1*) - για λόγους ευκολία σχεδίασης - θα θεωρηθεί ότι είναι ήδη με κατεύθυνση το αντικείμενο ενδιαφέροντος.

Κύρια σημεία αναφοράς, είναι αρχικά να επιτευχθεί relative positioning του αντικειμένου με real time onboard sensing and computing, με την βοήθεια ενός Embedded Linux System. Άρα πρόκειται να σχεδιαστεί ένα Anchor-based, One-hop και Range-based σύστημα για μελλοντική χρήση σε Outdoor Scenarios το οποίο προσπαθεί να επιτύχει Online Distributed Position Computation (βλ. *Παράγραφο 3.1.3*).

Η εκτίμηση της απόστασης του αντικειμένου από το κάθε drone θα γίνει με χρήση monocular vision - από την στιγμή που αποτελεί ένα ετερογενές κομμάτι του συστήματος - όπου αφού εντοπιστεί, ακολουθούνται τεχνικά με τις γνώσεις των ακριβών του διαστάσεων και τις λαμβανόμενες από την κάμερα μετρήσεις (βλ. *Παράγραφο 3.2.1*) για να υπολογιστούν τα ranges του αντικειμένου από το κάθε drone, τα οποία στην συνέχεια θα χρησιμοποιηθούν στον Range-based αλγόριθμο ώστε να πραγματοποιηθεί το position computation (βλ. *Παράγραφο 3.1.2*).

Κεφάλαιο 4

Διαδικασία Σχεδίασης και Υλοποίησης

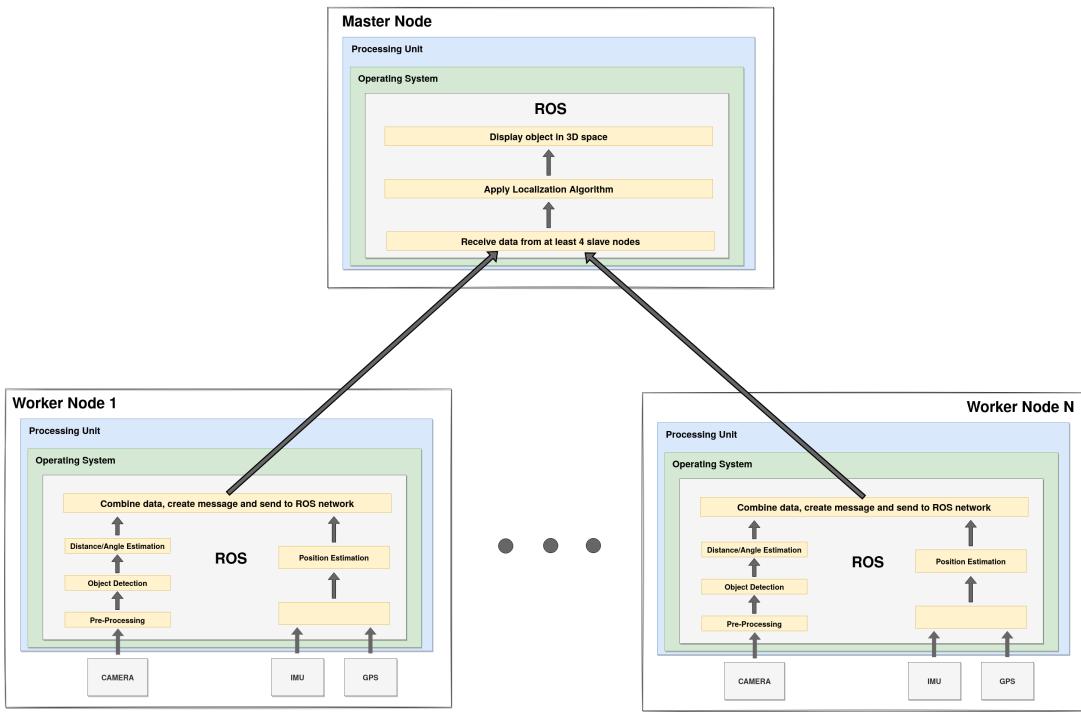
”Οι στόχοι μας μπορούν μόνο να
έρθουν εις πέρας με την
σχεδιάση ενός καλά¹
οργανωμένου πλάνου, στο οποίο
πρέπει επίμονα να πιστεύουμε,
και το οποίο με σθένος πρέπει να
πράτουμε. Δεν υπάρχει άλλος
δρόμος για την επιτυχία.”

Pablo Picasso

Στο παρόν κεφάλαιο περιγράφεται η διαδικασία σχεδιασμού και υλοποίησης του [Mo-Cap](#) με χρήση drone swarm συστήματος, που συσχετίζεται η παρούσα διπλωματική.

Μία high-level προσέγγιση, θα μπορούσε να διαχωρίζει το σύστημα σε τρία διαχριτά υποσυστήματα. Αρχικά το optical, του οποίου αρμοδιότητα είναι το detection, το tracking, καθώς και η εκτίμηση του range ή γωνίας του αντικειμένου από την camera. Δεύτερο, η λήψη των πληροφοριών από τους αισθητήρες ώστε να προσεγγιστεί η θέση του ίδιου του drone. Τέλος, ο συνδυασμός των δύο παραπάνω μερών και η χρήση κατάλληλης localization τεχνικής για να βρεθεί η θέση του αντικειμένου στο [3D](#) χώρο.

Πριν γίνει αναφορά καθενός από τα υποσυστήματα που σχεδιάστηκαν, στην *Eικόνα 4.1* παρουσιάζεται ένα block diagram του γενικού συστήματος, το οποίο αποτελείται από τα worker nodes - δουλειά των οποίων είναι η ανίχνευση του αντικειμένου και εκτίμηση της απόστασης του από αυτά - τα οποία αποστέλλουν στον master του συστήματος όλες τις πληροφορίες που έχουν συλλέξει και είναι αυτός που αναλαμβάνει τον προσδιορισμό της θέσης τελικά του object.



ΕΙΚΟΝΑ 4.1: High level system block diagram

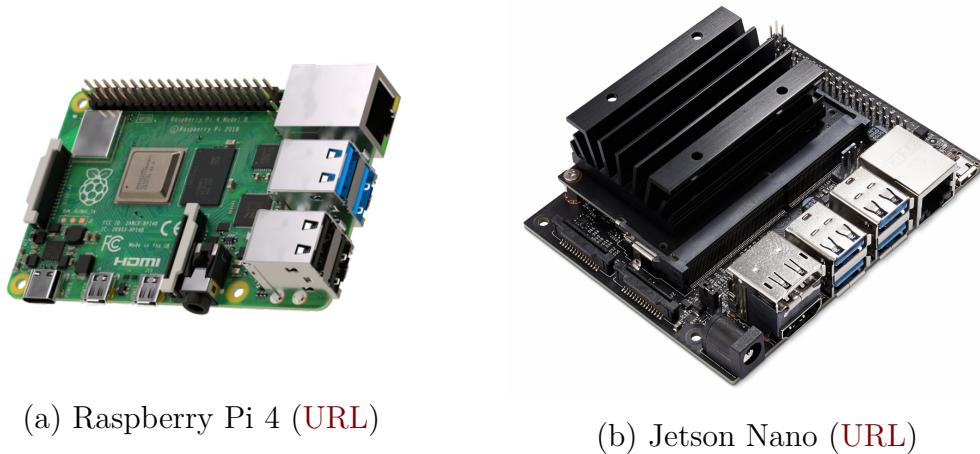
4.1 Τεχνολογίες και εργαλεία

Αρχικά όταν αναφερθούν συνοπτικά οι τεχνολογίες στις οποίες έγινε επιλογή να επιλυθεί το πρόβλημα, όπως επίσης και τα εξαρτήματα/αισθητήρια όργανα καθώς και τα λογισμικά τα οποία χρησιμοποιήθηκαν. Συνοπτικά, μπορεί να αναφερθεί ότι το σύνολο της υλοποίησης¹ αποτελείται από κώδικα C++, Bash scripts, Matlab scripts και ROS configuration files.

4.1.1 Embedded Linux System

Δύο πολύ δημοφιλείς επιλογές στον χώρο των ενσωματωμένων συστημάτων - ως κεντρικές μονάδες επεξεργασίας - είναι οι πλακέτες Raspberry Pi που κατασκευάζονται από την Raspberry Pi Foundation σε συνεργασία με την Broadcom, καθώς και οι πλακέτες Jetson της Nvidia. Στην Εικόνα 4.2 ως παράδειγμα παρουσιάζονται ενδεικτικά μία εκδοχή από την κάθε οικογένεια, ενώ στον Πίνακα 4.1 και Πίνακα 4.2 τα τεχνικά χαρακτηριστικά της εκάστοτε πλακέτας ως αρχικό σημείο αναφοράς.

¹Η οποία μπορεί να βρεθεί ως open source project διαθέσιμη στον εξής σύνδεσμο [116]



ΕΙΚΟΝΑ 4.2: Υποψήφια Embedded Linux Systems για την υλοποίηση

Και οι δύο επιλογές αποτελούνται από έναν ARM αρχιτεκτονικής Central Processing Unit (CPU), ενώ στις Jetson βρίσκεται επιπρόσθετα και ένα Graphics Processing Unit (GPU) που μπορεί να χρησιμοποιηθεί σε ενσωματωμένα με μεγάλες ανάγκες επεξεργασίας (όπως αυτά που σχετίζονται με εικόνα/βίντεο).

ΠΙΝΑΚΑΣ 4.1: Raspberry Pi 4 Model B Specifications

Feature	Value
Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	8GB LPDDR4-3200 SDRAM
Storage	External Micro-SD
Power	5V DC (maximum 3A), 5-15Watt
Cost	~100 €
Weight	46 grams (without case), 99 grams (with case)
Peripherals	GPIO, I2C, SPI, UART

Στην συγκεκριμένη διπλωματική επιλέχθηκε η ανάπτυξη του συστήματος να γίνει σε Raspberry Pi 4 boards - λόγω του ελάχιστα μικρότερου κόστους και μάζας τους - έχοντας μελλοντικά την επιλογή για migration ενός ή περισσότερων κόμβων του συστήματος σε Jetson boards, αν χριθεί αυτή η ανάγκη, για λόγους σχετικούς με την ταχύτερη επεξεργασία των δεδομένων.

ΠΙΝΑΚΑΣ 4.2: Jetson Nano Developer Kit Specifications

Feature	Value
CPU	Quad-core ARM Cortex-A57 MPCore processor
GPU	NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores
Memory	4 GB 64-bit LPDDR4; 25.6 gigabytes/second
Storage	External Micro-SD
Power	5V DC, 5-10Watt
Cost	~120€
Weight	250 grams (without case)
Peripherals	GPIO, I2C, I2S, SPI, UART

4.1.2 ROS

Συνδετικός χρίκος των υποσυστημάτων είναι το open-source middleware Robot Operation System ([ROS](#)) [117] το οποίο περιλαμβάνει ένα εκτενές σύνολο εργαλείων, βιβλιοθηκών και συμβάσεων. Τα πακέτα του οποίου χρησιμοποιούνται για την λήψη και φιλτράρισμα από τους αισθητήρες των πληροφοριών, επικοινωνία μεταξύ των drone όπως τέλος και όποια τρισδιάστατη απεικόνιση χρειάζεται.

Μεγάλο πλεονέκτημα του [ROS](#) είναι η ύπαρξη των packages. Τα packages είναι διακριτά αυτόνομα κομμάτια κώδικα τα οποία περικλείουν μία συχνά επαναλαμβανόμενη λογική, συνεπώς μπορούν να χρησιμοποιηθούν αυτούσια - με πολύ εύκολο τρόπο - σε διαφορετικές εφαρμογές χωρίς να υπάρχει η ανάγκη να κάνουμε *reinvent the wheel* κάθε φορά, πετυχαίνοντας με αυτό τον τρόπο το rapid prototyping and testing ενός συστήματος καθώς και την αποφυγή δημιουργίας boilerplate κώδικα δίνοντας έμφαση περισσότερο στο main logic του εκάστοτε συστήματος.

Το [ROS](#) στην πραγματικότητα είναι ένα meta-operating system, με αποτέλεσμα να χρειάζεται να τρέχει σε ένα πρωτεύον Operating System ([OS](#)), με την μεγαλύτερη συμβατότητά να παρέχεται από διανομές Ubuntu Linux.

Σημαντικές πηγές που χρησιμοποιήθηκαν για την εκμάθηση του [ROS](#) ήταν κυρίως το documentation του [118], καθώς και κάποια από τα external tutorials τα οποία προτείνονται στην παραπάνω σελίδα.

4.1.3 Λειτουργικό σύστημα

Από την στιγμή που επιλέχθηκε να γίνει χρήση του [ROS](#) - που όπως αναφέρθηκε στη [Παράγραφο 4.1.2](#) προτείνεται ο συνδυασμός του με Ubuntu Linux - έγινε εγκατάσταση στο Raspberry Pi η ειδικά σχεδιασμένη για το αυτό έκδοση Ubuntu Linux 20.04.2 64bit version για ARM [119] αρχιτεκτονική. Ουσιαστικά όλες οι λειτουργίες

της εφαρμογής όταν χρησιμοποιούν το Embedded General Purpose Operating System προκειμένου να λειτουργήσουν, το οποίο σημαίνει ότι αυτό όταν είναι κυρίως υπεύθυνο για το scheduling, file-system abstraction, networking, etc.

4.1.4 OpenCV

Για το οπτικό σκέλος χρησιμοποιήθηκε η ανοιχτού κώδικα βιβλιοθήκη OpenCV [58], η οποία αποτελεί την δημοφιλέστερη επιλογή για real-time Computer Vision related εφαρμογές. Ξεκίνησε η ανάπτυξη της στα εργαστήρια της Intel και έχει ήδη διάρκεια ζωής λίγο περισσότερο από δύο δεκαετίες. Για τις ανάγκες της συγκεκριμένης εργασίας χρησιμοποιήθηκε η έκδοση της 4.2.0. Όμοια με το ROS και για την εκμάθηση του OpenCV, καθοριστικό ρόλο συνέβαλε η κατανόηση του επίσημου documentation της βιβλιοθήκης [120].

4.1.5 Κάμερα

Σχετικά με την κάμερα, ήταν ανάγκη - όντας πρώτη γενιά του συστήματος - να επιλεχθεί μία low-cost 1080p camera η οποία όταν παρέχει δυνατότητα επιλογής χαμηλότερου resolution για λόγους δοκιμών. Στο πρωτότυπο σύστημα τελικά γίνεται χρήση μία 1080p web cameras της Creative [121] - Εικόνα 4.3 - με δυνατότητες λήψης βίντεο στα 30fps, η οποία συνδέεται μέσω USB στο Raspberry Pi.



ΕΙΚΟΝΑ 4.3: Η κάμερα που χρησιμοποιήθηκε για τον εντοπισμό και ανίχνευση του αντικειμένου ([URL](#))

4.1.6 GPS

Παρόλο που - όπως αναφέρθηκε στο Κεφάλαιο 3.3 - στο συγκεκριμένο σύστημα μας ενδιαφέρει το relative positioning και όχι το absolute, σκεπτόμενοι ότι στην πραγματικότητα το σύστημα σχεδιάζεται με γνώμονα το να λειτουργεί σε outdoor

scenarios· ένας άμεσος τρόπος προσδιορισμού της θέσης του κάθε drone είναι με χρήση κάποιου εμπορικού αισθητήρα **GPS**. Στην συνέχεια και αφού έχει αποκτηθεί πληροφορία απόλυτης θέσης για το κάθε drone μπορεί - θεωρώντας ένα από αυτά ως αρχή των αξόνων - να γίνει translate των απόλυτων συντεταγμένων ώστε να κρατηθεί μόνο πληροφορία σχετικά με την χωρική τοπολογία του δικτύου.

Συγκεκριμένα, χρησιμοποιείται το **GPS** για commercial χρήση BN-220 [122] - *Εικόνα 4.4* - το οποίο υπόσχεται εμβέλειας ακρίβειας της τάξης των δύο μέτρων. Πρακτικά, παρόλο που για ένα **MoCap** σύστημα αυτή η τιμή είναι απαγορευτική, χρησιμοποιείται στα πρώτα versions, καθαρά αναφορικά με την εξουκείωση του τρόπου επικοινωνίας **ROS - GPS**. Σε επόμενα revisions σκοπός είναι η αντικατάσταση του με **RTK-GPS** που συχνά μπορεί να φέρουν drone για αυτήν την χρήση ώστε να φτάσει η συνολική ακρίβεια εκτίμησης της απόλυτης θέσης στα μερικά εκατοστά.

Το συγκεκριμένο **GPS** συνδέεται με το Raspberry Pi μέσω Universal Asynchronous Receiver-Transmitter (**UART**) [123] σύνδεσης και χρησιμοποιεί NMEA-0183 [124] πακέτα για την επικοινωνία.



ΕΙΚΟΝΑ 4.4: Το GPS που χρησιμοποιήθηκε ([URL](#))

4.1.7 IMU

Ηδη από τη *Παράγραφο 1.1.2* έχει γίνει αναφορά για την σημαντικότητα των **IMU**, καθώς αποτελούν τα κύρια αισθητήρια όργανα καυθορισμού σε πολλαπλούς άξονες της σχετικής θέσης/κίνησης του drone. Επιλέχθηκε να χρησιμοποιηθεί το **IMU** [125] της Adafruit - *Εικόνα 4.5* - το οποίο παρέχει 10-**DoF**, με onboard αισθητήρες ένα τριών αξόνων accelerometer, τριών αξόνων gyroscope, τριών αξόνων magnetometer (compass), ένα barometric pressure/altitude αισθητήρα και δυνατότητα υπολογισμού της θερμοκρασίας.

Θετικό του συγκεκριμένου module είναι ότι όλοι οι παραπάνω αισθητήρες είναι συνδεδεμένοι σε ένα κοινό Inter-Integrated Circuit (**I2C**) [126] bus, μέσω του οποίου μπορεί πολύ εύκολα να γίνει η διασύνδεση με την επεξεργαστική μονάδα που επιθυμούμε και να πραγματοποιηθεί επικοινωνία.

ΕΙΚΟΝΑ 4.5: Adafruit 10 DoF IMU ([URL](#))

4.1.8 Breakout Board

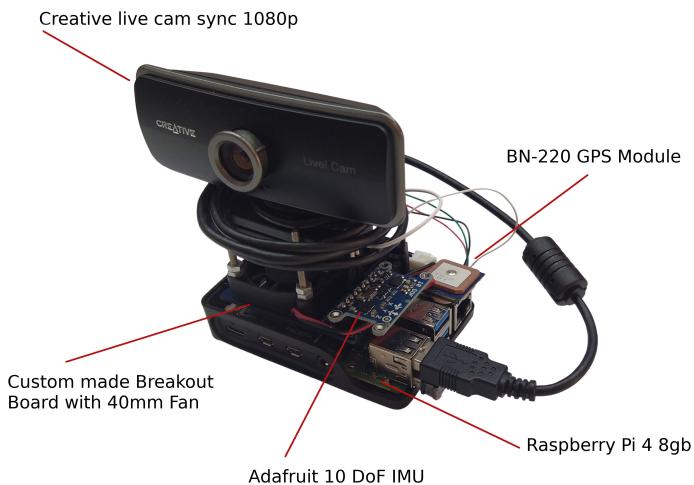
Για να λειτουργήσουν τα παραπάνω υποσυστήματα, χρειαζόταν να πραγματοποιηθούν οι κατάλληλες φυσικές διασυνδέσεις. Η απλούστερη εκδοχή θα ήταν να γίνει αυτό με χρήση breadboard, πράγμα όμως που θα πρόσθετε όγκο και βάρος στο τελικό σύστημα, οι οποίοι σε περίπτωση δοκιμών πάνω σε πραγματικά drone να είναι απαγορευτικοί παράγοντες. Συνεπώς, προκειμένου να μπορεί με ευκολία να γίνει η ανάπτυξη του συστήματος, σχεδιάστηκε (στο cad εργαλείο KiCad [127]) και κατασκευάστηκε ένα custom breakout board - το οποίο είναι διαθέσιμο ως open hardware project στο [128] - όπως φαίνεται στην Εικόνα 4.6. Αυτό παρέχει εύκολη πρόσβαση στα GPIO του Raspberry, έξτρα pins για τροφοδοσία στα 5 και 3.3 Volt, pins για τοποθέτηση αισθητήρων - όπως του IMU - καθώς και mounting holes στα οποία μπορεί να τοποθετηθεί 40x40mm fan για την ψύξη του συστήματος.



ΕΙΚΟΝΑ 4.6: Raspberry Pi breakout

4.1.9 System Overview

Η ολοκληρωμένη μορφή του πρωτότυπου συστήματος παρουσιάζεται στην Εικόνα 4.7, ενώ στο Πίνακα 4.3 μπορούν να βρεθούν συνολικά τα εξαρτήματα που χρησιμοποιήθηκαν μαζί με την κοστολόγηση τους.



ΕΙΚΟΝΑ 4.7: Το πρώτυπο σύστημα που σχεδιάστηκε στα πλαίσια της διπλωματικής

Το σύστημα ζυγίζει ~ 250gr, μία αρχική εκτίμηση κατανάλωσης είναι τα 15 watt και οι εξωτερικές διαστάσεις του είναι 17x7.5x10.5cm.

ΠΙΝΑΚΑΣ 4.3: Bill of Materials

Component	Cost
Raspberry Pi 4 Model B 8GB	~ 100 €
Creative live cam sync 1080p [121]	~ 44 €
Adafruit 10 DoF IMU [125]	~ 30 €
BN-220 GPS Module [122]	~ 15 €
Breakout Board with fan [128]	~ 8 €

4.2 Environment

Έχοντας ήδη αναφερθεί σε όλα τα υποσυστήματα που χρησιμοποιούνται, σε αυτό το section θα δούθει ο τρόπος με τον οποίο διαμορφώθηκαν/προγραμματίστηκαν ώστε να λειτουργούν μεταξύ τους.

4.2.1 Παραμετροποίηση OS

Στο [129] βρίσκονται αναλυτικές οδηγίες εγκατάστασης του Ubuntu στο Raspberry Pi, ανάλογα με το λειτουργικό που ήδη χρησιμοποιούμε. Στην συγκεκριμένη περίπτωση - επειδή η εγκατάσταση έγινε από διανομή Linux - αφού έγινε λήψη

του pre-made image του Ubuntu, βρέθηκε το path της SD card στον υπολογιστή, στην οποία έγινε umount, και στην συνέχεια χρησιμοποιήθηκε η εντολή dd για την δημιουργία του bootable μέσου, με τον εξής τρόπο.

ΠΑΡΑΘΕΣΗ 4.1: Create bootable SD from Linux

```
$ sudo dd bs=4M if=PATH_TO_YOUR_IMAGE_FILE of=PATH_TO_YOUR_SD_CARD  
status=progress
```

Μόλις ολοκληρωθεί η παραπάνω διαδικασία, το λειτουργικό σύστημα είναι έτοιμο προς χρήση. Αφού έγινε update του συστήματος, έγιναν οι εξής παραμετροποιήσεις. Αρχικά απενεργοποιήθηκε το Graphical User Interface κατά την διάρκεια του boot, επίσης απενεργοποιήθηκε το auto-suspend μετά από χρονικό διάστημα μη χρήσης, και τέλος εγκαταστάθηκαν οι εφαρμογές/πακέτα/βιβλιοθήκες - όπως το ROS - που είναι απαραίτητα για την υλοποίηση του συστήματος.

Σε αυτό το σημείο να αναφερθεί ότι χρησιμοποιήθηκε η έκδοση noetic του ROS.

4.2.2 Επικοινωνία αισθητήρων

Αφού ολοκληρώθηκαν οι παραπάνω απαραίτητες ενέργειες, υπήρχε πλέον ένα λειτουργικό περιβάλλον, οπότε και ξεκίνησε η διαδικασία πραγματοποίησης επιτυχημένης επικοινωνίας με το κάθε υποσύστημα.

Σειριακή Επικοινωνία

Πρώτα ωστόσο αναφερθεί η επικοινωνία με το GPS η οποία όπως αναφέρθηκε στη Παράγραφο 4.1.6 γίνεται μέσω UART. Η σειριακή port του Raspberry μπορεί να γίνει access μέσω του αρχείου /dev/ttys0. Ενώ, για να μπορεί να προσπελαστεί από τον χρήστη χρειάστηκε να γίνουν τα βήματα [130] που υπάρχουν στη Παράθεση 4.2.

ΠΑΡΑΘΕΣΗ 4.2: Fix serial communication

```
# 1. Add 'enable_uart=1' to /boot/config.txt file
sudo bash -c 'echo "enable_uart=1" >> /boot/config.txt'

# 2. Remove 'console=serial0,115200' from /boot/firmware/cmdline.txt
sudo sed -e
"s/console=serial0,115200//g" -i /boot/firmware/cmdline.txt

# 3. Disable serial console service
sudo systemctl stop serial-getty@ttyS0.service
sudo systemctl disable serial-getty@ttyS0.service

# 4. Give privileges to user
sudo adduser $USER tty
sudo adduser $USER dialout
sudo chmod g+r /dev/ttyS0
```

Μετά την εκτέλεση τους, συνδέοντας κατάλληλα τα RX - TX του **GPS** στο Raspberry, είναι εφικτό κάνοντας run την εντολή ‘`cat /dev/ttyS0` να έχουμε πρόσβαση στα πακέτα NMEA που στέλνει το **GPS**, που έχουν μορφή παρόμοια με τη *Παράθεση 4.3.*

ΠΑΡΑΘΕΣΗ 4.3: Serial Output, NMEA packets example

```
...
$GNGSA,A,1,,,,,,,99.99,99.99,99.99*2E
$GPGSV,1,1,01,31,,,13*78
$GLGSV,1,1,00*65
$GNGLL,,,,,180928.00,V,N*5E
...
```

Προκειμένου το **ROS** να χρησιμοποιεί το **GPS**, χρειάστηκε να εγκατασταθεί το πακέτο **nmea_notsat_driver** [131] και παράδειγμα χρήσης αυτού με το **ROS** υπάρχει στη *Παράθεση 4.4.*

ΠΑΡΑΘΕΣΗ 4.4: GPS - ROS sample usage

```
rosrun nmea_notsat_driver nmea_serial_driver _port:=/dev/ttyS0
_baud:=9600
```

Σημαντικό είναι να αναφερθεί ότι η σειριακή θύρα χρησιμοποιείται για debug λόγους κατά το boot του Raspberry Pi, συνεπώς το **GPS** πρέπει να μην είναι συνδεδεμένο

αρχικά, και μόνο αφού ολοκληρωθεί το boot να συνδεθεί στο σύστημα.

Επικοινωνία I2C

Σε αντίθεση με την σειριακή επικοινωνία, το IMU module χρησιμοποιεί το I2C πρωτόκολλο (βλ. *Παράγραφο 4.1.7*). Για να μπορέσουμε να χρησιμοποιήσουμε το I2C στο Raspberry, χρειάστηκε να εκτελεστούν οι εντολές που υπάρχουν στη *Παράθεση 4.5*.

ΠΑΡΑΘΕΣΗ 4.5: Fix I2C communication

```
# 1. Install needed library
sudo apt-get install -y libi2c-dev i2c-tools

# 2. Give privileges to user
sudo adduser $USER i2c
sudo chmod g+r /dev/i2c-1
```

Στην συνέχεια μπορούν να γίνουν οι κατάλληλες συνδέσεις και εκτελώντας την εντολή ‘**i2cdetect -y 1**’ έχουμε πρόσβαση στις διευθύνσεις όλων των αισθητήρων που είναι συνδεδεμένες στο I2C bus, παράδειγμα του output από την εντολή υπάρχει στη *Παράθεση 4.6*².

ΠΑΡΑΘΕΣΗ 4.6: I2C addressed output example

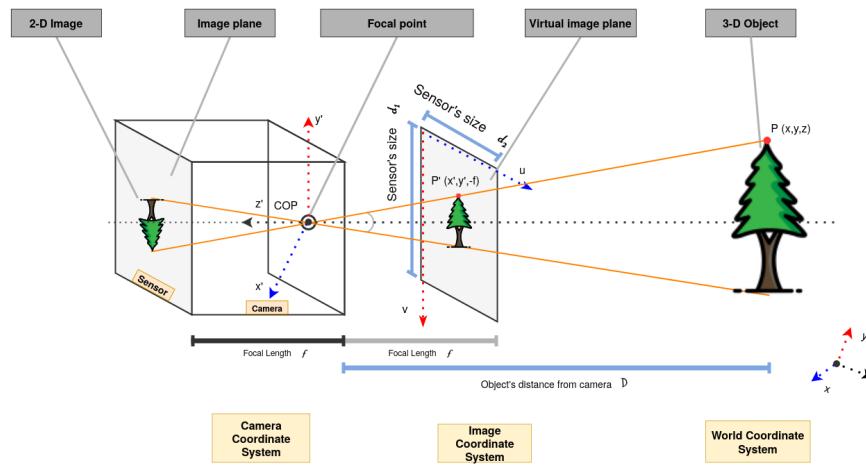
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10:	-	-	-	-	-	-	-	-	-	-	19	-	-	-	1e	--
20:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60:	-	-	-	-	-	-	-	-	-	-	6B	-	-	-	-	-
70:	-	-	-	-	-	-	-	-	-	-	77	-	-	-	-	-

Το μόνο ROS πακέτο που βρέθηκε [132] - που χρησιμοποιεί αυτό το module - περιείχε κάποια compilation errors. Αφού διορθώθηκαν και προστέθηκαν επιπλέον log messages - το updated repo μπορεί να βρεθεί εδώ [133] - ήταν πλέον δυνατό το IMU να επικοινωνήσει επιτυχώς με το ROS.

²Το συγκεκριμένο module έχει διοθεί από την Adafruit ως open hardware project, σε περίπτωση που χρησιμοποιήσει κάποιο replica του, πιθανόν κάποιο address να είναι διαφορετικό

4.3 Camera

Πριν αναφερθεί ο τρόπος με τον οποίο χρησιμοποιήθηκε η κάμερα, χρειάζεται πρώτα να μοντελοποιηθεί η πληροφορία που μας παρέχει μία εικόνα. Στην πραγματικότητα όταν μιλάμε για images, ουσιαστικά αναφερόμαστε σε functions που έχουν ως πεδίο ορισμού τα 3D points του χώρου στον οποίο τα λάβαμε, και ως πεδίο τιμών τα 2D projection points που κατέγραψε ο sensor της κάμερας. Για το projection υπάρχουν διάφορα μοντέλα, όπως το Perspective, Weak και Orthographic. Στα πλαίσια της συγκεκριμένης διπλωματικής βασιζόμαστε στο Perspective projection model.



ΕΙΚΟΝΑ 4.8: Ιδανικό μοντέλο του pinhole

Επίσης, ένα απλό (για την κάμερα) - παρόλα αυτά αρκετά περιγραφικό και κοντά στην πραγματικότητα - μοντέλο, το οποίο χρησιμοποιείται συχνά για CV applications είναι αυτό του Pinhole Model, παράδειγμα στην Εικόνα 4.8, με το COP να είναι το Center of Projection και να χρησιμοποιούμε το Virtual Image Plane για τους υπολογισμούς - μαθηματικά ισοδύναμο με το πραγματικό Image Plane - με τη διαφορά του μη ανεστραμμένου ειδώλου.

Μπορούμε να μεταφερθούμε - από ένα κοινό σε όλους - World Coordinate System (${}^w\vec{p}$) στο Coordinate System της κάμερας (${}^c\vec{p}$) μέσω μίας Rotation (c_wR) και μίας Translation (${}^c_w\vec{t}$) διαδικασίας - Εξίσωση (4.1). Συχνά τα περιεχόμενα των Rotation και Translation ονομάζονται από κοινού Extrinsic Parameters.

Ενώ για τη πραγματική μεταφορά από το Camera Coordinate System σε αυτό της εικόνας θα πρέπει να λάβουμε υπόψιν τις φυσικές παραμέτρους της κάμερας όπως το Focal Length, Skew, etc., οι οποίες συνολικά ονομάζονται Intrinsic Parameters. Η σχέση (4.2) παρουσιάζει αυτόν τον μετασχηματισμό, με (u, v) τα pixel στην κάμερα,

(x, y, z) οι συντεταγμένες του αντικειμένου με βάση το Camera Coordinate System και $(\alpha, \beta, \theta, u_0, v_0)$ τα Intrinsic Parameters.

$${}^c\vec{p} = {}_w^cR {}^w\vec{p} + {}_w^c\vec{t} \quad (4.1)$$

$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0 \quad v = \frac{\beta}{\sin \theta} \frac{y}{z} + v_0 \quad (4.2)$$

Οι παραπάνω υπολογισμοί με τη μορφή που είναι αυτή την στιγμή κάνουν πιο δύσκολο τον τρόπο υπολογισμού σε ψηφιακά συστήματα. Επιπλέον έχουν μη γραμμικά μέρη, με αποτέλεσμα να μην είναι αναστρέψιμοι, για αυτόν τον λόγο γίνεται αναφορά των homogeneous coordinates. Τα homogeneous coordinates χρησιμοποιούνται ώστε να μπορούμε να μετασχηματίσουμε εύκολα, σε μία πλέον τις παραπάνω σχέσεις, με χρήση πινάκων - *Εξίσωση* (4.3). Όπου x οι συντεταγμένες στην εικόνα, X οι συντεταγμένες στον φυσικό κόσμο και M ο πίνακας μετασχηματισμού, τα περιεχόμενα του οποίου αναλύονται στην σχέση (4.4)

$$(x/w, y/w) \Leftrightarrow \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad \text{homogeneous image (2D) coordinates}$$

$$(x/w, y/w, z/w) \Leftrightarrow \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \quad \text{homogeneous scene (3D) coordinates}$$

$$x \simeq \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = MX = M \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.3)$$

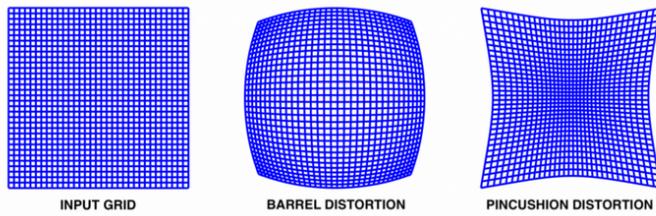
$$M_{3x4} = \begin{bmatrix} f & s & x'_c \\ 0 & af & y'_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3x3} & 0_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \begin{bmatrix} I_{3x3} & T_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \quad (4.4)$$

Intrinsics *Projection* *Rotation* *Translation*

Ένα αρκετά βοηθητικό Massive Open Online Course ([MOOC](#)) το οποίο βοήθησε ώστε να κατανοηθεί περισσότερο το σκέλος του [CV](#) και στο οποίο μπορεί κάποιος να ανατρέξει για περισσότερες λεπτομέρειες σε σχέση με τους παραπάνω φορμαλισμούς είναι το [113]. Επιπρόσθετα, στη [Παράγραφο 4.3.1](#) περιγράφεται ο τρόπος υπολογισμού των στοιχείων του πίνακα μετασχηματισμού M .

4.3.1 Camera Calibration

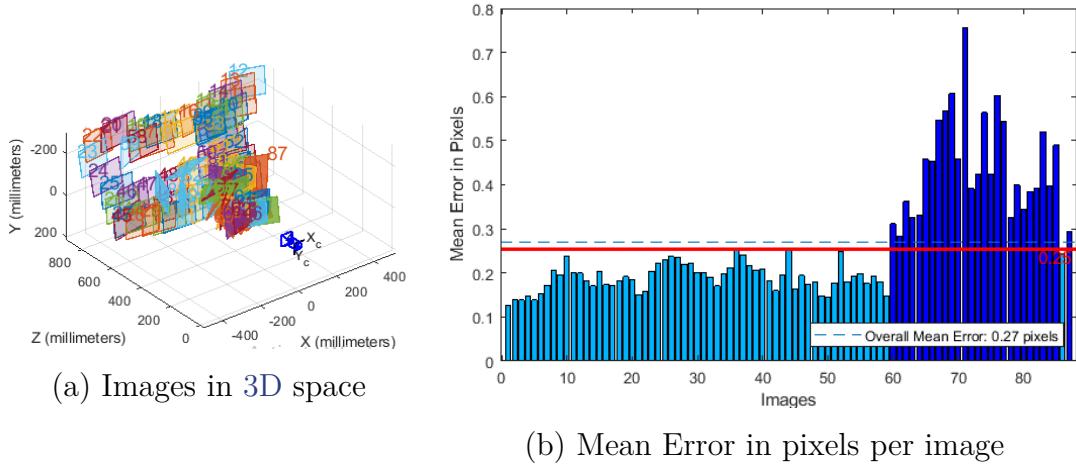
Στη [Παράγραφο 4.3](#) έγινε αναφορά του πίνακα μετασχηματισμού M . Επίσης, τα lens μίας κάμερας δεν είναι ιδανικά, κάνοντας τις λήψεις να φέρουν παραμορφώσεις (βλ. [Εικόνα 4.9](#)). Η συνολική παραμόρφωση ενός lens μπορεί να περιγραφεί μέσω των Distortion coefficients. Η διαδικασία υπολογισμού όλων των παραμέτρων της κάμερας - προκειμένου αξιόπιστα να μπορούμε να κάνουμε υπολογισμούς με βάση τις λήψεις της σε μία [CV](#) εφαρμογή - ονομάζεται camera calibration.



ΕΙΚΟΝΑ 4.9: Οι πιο συνηθισμένοι τύποι distortion των φωκών ([URL](#))

Στην συγκεκριμένη διπλωματική τυπώθηκε ένα checkerboard pattern, και αξιοποιήθηκαν εργαλεία που υλοποιούν το Zhang's method για τον υπολογισμό των παραμέτρων. Αρχικά, έγινε προσπάθεια με native τρόπο, χρησιμοποιώντας άμεσα το framework [OpenCV](#) να πραγματοποιηθεί το calibration, επειδή όμως κατέληγε σε μεγάλο mean error in pixel - περίπου 3.56 στα πειράματα - αποφασίστηκε να χρησιμοποιηθεί εναλλακτική μέθοδος. Παρόλο που και το [ROS](#) παρέχει πακέτο για το calibration, επιλέχθηκε μέσω του Matlab και χρήση του Camera Calibration toolkit να πραγματοποιηθεί τελικά. Κύριος λόγος διότι σε αυτό παρέχονται επιπρόσθετες πληροφορίες - βλ. [Εικόνα 4.10](#) - κατά το calibration, επιτυγχάνοντας τελικά mean error in pixel περίπου 0.25. Αυτή η διαδικασία πραγματοποιήθηκε για αναλύσεις 1920x1080, 1280x720 και 640x480.

Στο τέλος του calibration, είχαν υπολογιστεί και έγιναν exported τα parameters. Ενώ με τα περιεχόμενα των πεδίων FocalLength, PrincipalPoint, RadialDistortion, TangentialDistortion και Skew δημιουργήθηκαν τα yaml αρχεία που χρειάζεται το [ROS](#) [134] [135].

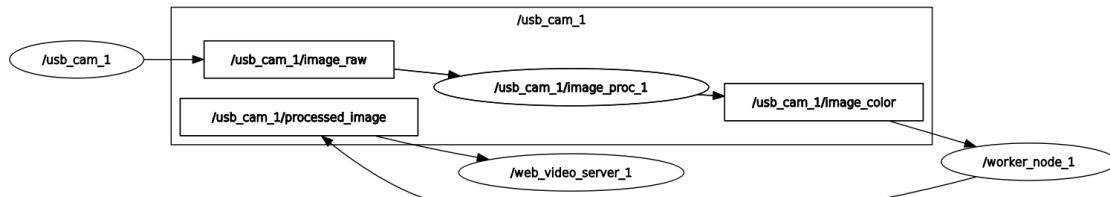


ΕΙΚΟΝΑ 4.10: Matlab camera toolkit

4.3.2 Χρήση κάμερας

Προκειμένου να λειτουργεί η κάμερα με το ROS, χρειάστηκε αρχικά να συνδεθεί μέσω USB με το Raspberry Pi, ενώ χρησιμοποιήθηκε το package `usb_cam` για την διαχείριση της. Στο launch αρχείο που δημιουργήθηκε, κατά την εκτέλεση του `/usb_cam_node`, γίνεται επιλογή της ανάλυσης που πρόκειται να χρησιμοποιηθεί· όπως επίσης μέσω της παραμέτρου `camera_info_url`, επιλέγεται το yaml αρχείο με τα χαρακτηριστικά της κάμερας που δημιουργήθηκε κατά το calibration.

To `/usb_cam_node` κάνει publish στο topic `/usb_cam/image_raw` τα περιεχόμενα της κάμερας. To `image_proc` είναι υπεύθυνο για το undistortion του image ώστε να μπορούν να πραγματοποιηθούν αξιόπιστες μετρήσεις. Η επεξεργασία του stream της κάμερας γίνεται από το node με όνομα `/worker_node` που δημιουργήθηκε, το οποίο κάνει subscribe στο topic της undistorted εικόνας, μετασχηματίζει από πακέτα ROS σε Mat στοιχεία - που χρησιμοποιεί το OpenCV, μέσω του `cv_bridge` - και αναλαμβάνει το κομμάτι της CV επεξεργασίας για της ανάγκες της εφαρμογής.



ΕΙΚΟΝΑ 4.11: ROS - camera related nodes/topics setup

Αρκετές ήταν οι φορές που υπήρχε ανάγκη (για λόγους αποσφαλμάτωσης), σε οπτικό feedback της [CV](#) επεξεργασίας από nodes, που μπορεί να είναι διαφορετικά φυσικά μηχανήματα από αυτά που τρέχει η εφαρμογή. Πράγμα που εύκολα επιλύεται αξιοποιώντας το package [/web_video_server](#). Αυτό είναι subscriber στο topic [/usb_cam/processed_image](#) (topic στο οποίο το [/worker_node](#) κάνει publish τα αποτελέσματα της επεξεργασίας) ώστε μέσω κάποιου browser να μπορούμε εύκολα να δούμε τα αποτελέσματα της. Η [Εικόνα 4.11](#) παρουσιάζει τα topics/nodes που αναφέρθηκαν και τον τρόπο με τον οποίο επικοινωνούν.

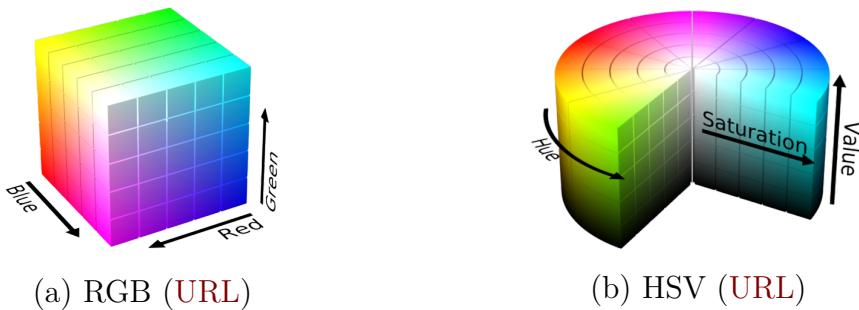
4.3.3 Εντοπισμός Αντικειμένου

Το αντικείμενο του οποίου η θέση επιλέχθηκε να προσδιοριστεί, είναι μία μπάλα - καθώς μπορεί να προσφέρει ανεξαρτήτου γωνία λήψης ίδια πληροφορία για τις διαστάσεις της - μεγέθους ανάλογη με τις μπάλες ποδοσφαίρου. Συνεπώς δεν μας ενδιαφέρει το orientation του αντικειμένου, παρά μόνο το location.

Με βάση το χρώμα

Πρώτη προσπάθεια εκτίμησης της θέσης, ήταν χρησιμοποιώντας ως σημείο αναφοράς το χρώμα της. Έχοντας επιλέξει το χρώμα της μπάλας να είναι διαφορετικό από τα περισσότερα στοιχεία της σκηνής στην οποία την καταγράφουμε, μπορούμε εύκολα να προσδιορίσουμε και την θέση του αντικειμένου στο image plane.

Συχνά οι λήψεις μίας κάμερας αναπαριστάνονται στο RGB color space, αρκετά χρήσιμο όμως είναι η χρήση του HSV για [CV](#) εφαρμογές, με κύριο λόγο τον διαχωρισμό που παρέχεται μεταξύ color information και intensity στο συγκεκριμένο color space. Οπτική αναπαράσταση αυτών βρίσκεται στην [Εικόνα 4.12](#).



ΕΙΚΟΝΑ 4.12: Color Spaces

Στον [Αλγόριθμο 4](#) παρουσιάζεται η διαδικασία που ακολουθήθηκε προκειμένου να εντοπιστεί η μπάλα στην σκηνή με βάση το χρώμα της, στην [Εικόνα 4.13](#) όμοια

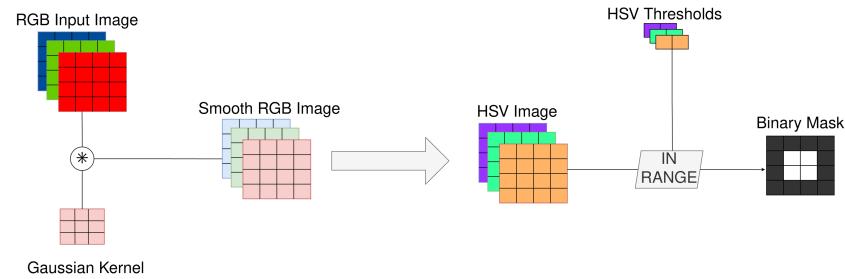
η επεξεργασία, ενώ στην *Eikόνα 4.14* εμφανίζονται τα αποτελέσματα από το κάθε επίπεδο επεξεργασίας.

Algorithm 4 HSV (color based) ball detection

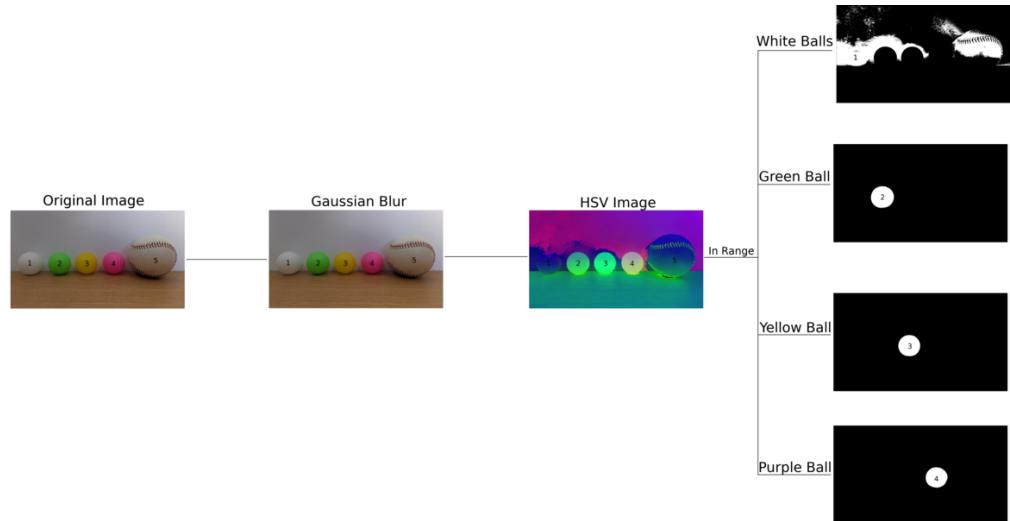
```

procedure detectBall
1: Define preferable color limit HSV values
2: for Each frame of the video/stream do
3:   Get frame
4:   Remove high frequencies from it
5:   Convert frame from RGB to HSV color space
6:   Choose only pixels that are in between preferable color limit values
7:   Find bounding box of mask (threshold image)

```



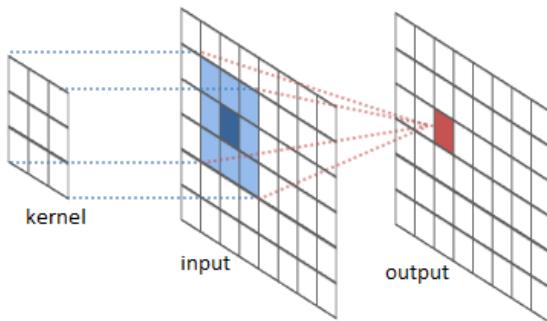
EIKONA 4.13: Επεξεργασία για ανίχνευση με βάση το χρώμα



EIKONA 4.14: Ανίχνευση μπάλων με βάση το χρώμα

Ένα μεγάλο μέρος των **CV** εφαρμογών περιλαμβάνει ως διαδικασίες επεξεργασίας το να γίνεται convolution ή cross-correlation³ ενός kernel με τα pixel της εικόνας. Αυτή η διαδικασία - που για το convolution περιγράφεται μαθηματικά από την εξίσωση (4.5) και οπτικά από την *Εικόνα 4.15* - έχει ποικίλες εφαρμογές οι οποίες διαχωρίζονται με βάση την επιλογή του kernel. Μία εξ' αυτών είναι για λόγους smoothing, που στην συγκεκριμένη περίπτωση περιλαμβάνει σημαντική διαδικασία του preprocessing, καθώς με αυτόν τον τρόπο μειώνουμε την ποσότητα των υψηλών συχνοτήτων συνεπώς και τον θόρυβο στην εικόνα.

$$O[i, j] = k * I = \sum_{u=-k}^k \sum_{v=-k}^k k[u, v]I[i - u, j - v] \quad (4.5)$$



ΕΙΚΟΝΑ 4.15: Convolution παράδειγμα ([URL](#))

Συγκεκριμένα χρησιμοποιήθηκε Gaussian μορφής kernel το οποίο παρέχει πιο φυσικής μορφής blur, και αφού έγινε το transform σε HSV επιλέχθηκαν μόνο τα pixel των οποίων οι τιμές ήταν αποδεκτές. Δύο μειονεκτήματα της συγκεκριμένης μεθόδου είναι ότι τα αποτελέσματα μεταβάλλονται σημαντικά ανάλογα τις συνθήκες φωτισμού, ακόμα και σε επίπεδο που για ίδια thresholds να μην γίνεται ανίχνευση του αντικειμένου. Όπως επίσης, αν το χρώμα του αντικειμένου δεν επιλεχθεί κατάλληλα - παράδειγμα στην *Εικόνα 4.14* για τις άσπρες μπάλες - το αντικείμενο τελικά να μην μπορεί να εντοπιστεί.

Με βάση το σχήμα

Στη δεύτερη προσπάθεια εντοπισμού της μπάλας, χρησιμοποιήθηκε η πληροφορία του σχήματος της, καθώς η προβολή που καταγράφει η κάμερα για αυτή, δεν είναι παρά

³Η διαφορά μεταξύ του convolution και του cross-correlation, είναι ότι στο convolution ο kernel αναστρέφεται

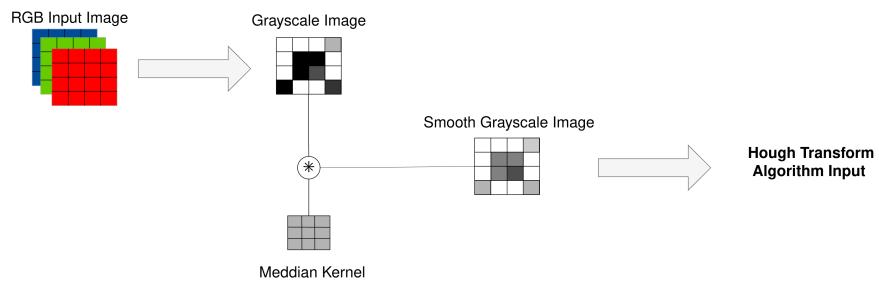
ένας κύκλος. Με όμοιο τρόπο στον *Αλγόριθμο 5* παρουσιάζεται σε ψευδοχώδικα η διαδικασία, στην *Εικόνα 4.13* όμοια η επεξεργασία, ενώ στην *Εικόνα 4.17* τα αποτελέσματα στο κάθε phase.

Algorithm 5 Hough circle (shape based) ball detection

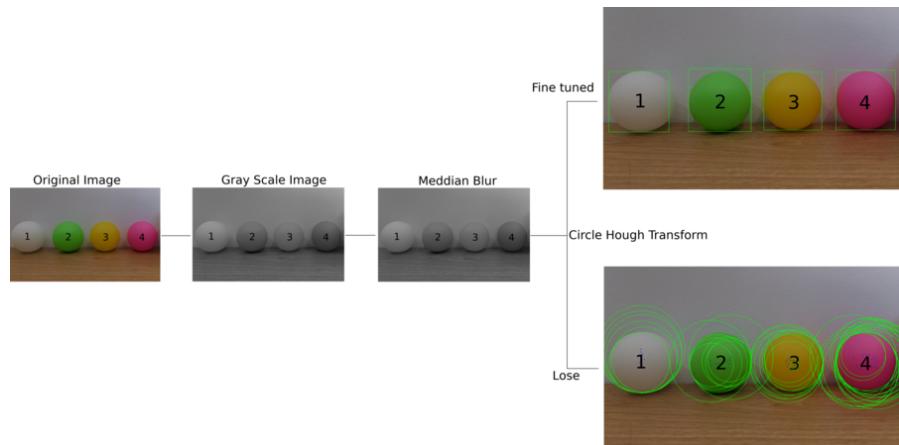
```

procedure HoughdetectBall
1: for Each frame of the video/stream do
2:   Get frame
3:   Convert frame from RGB to Grayscale
4:   Remove high frequencies from the frame
5:   Use Hough Transform principle to detect circles in image

```



EIKONA 4.16: Επεξεργασία για ανίχνευση με βάση το σχήμα

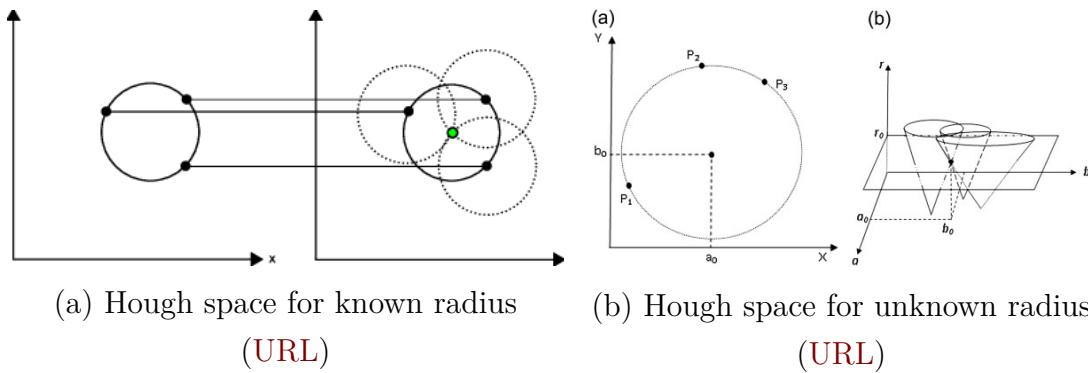


EIKONA 4.17: Ανίχνευση μπάλας με βάση το σχήμα

Για τον εντοπισμό του κυκλικού σχήματος της μπάλας χρησιμοποιήθηκε ο Hough Transform for circles detection. Ο τρόπος με τον οποίο λειτουργεί ο Hough Transform είναι να μετασχηματίζει το circle localization πρόβλημα σε ένα voting πρόβλημα.

Είσοδος στον αλγόριθμο Hough είναι τα edges ενός image, και έξοδος είναι η θέσεις των κύκλων.

Η απλούστερη περίπτωση του Hough Transform είναι σε μία εικόνα να κάνουμε to-localize έναν κύκλο με γνωστό radius r , όπως παρουσιάζεται στην *Eικόνα 4.18 (a)*. Σε αυτήν την περίπτωση για κάθε στοιχειώδες σημείο (x,y) των edges δημιουργούμε στο Hough space έναν κύκλο ακτίνας r με κέντρο (x,y) . Αφού γίνει αυτή η διαδικασία για κάθε σημείο των edges, το σημείο στο hough space (εναλλακτικό όνομα accumulator) με τα περισσότερα votes - σημείο χρώματος πράσινο στην *Eικόνα 4.18 (a)* - είναι τελικά και το κέντρο του κύκλου.



ΕΙΚΟΝΑ 4.18: Circle Hough Transform

Η πιο σύνθετη περίπτωση - *Εικόνα 4.18 (b)* - είναι όταν δεν γνωρίζουμε την ακτίνα του κύκλου που αναζητούμε στην σκηνή, σε αυτήν την περίπτωση επαναλαμβάνουμε την παραπάνω διαδικασία για διαφορετικές τιμές ακτίνων, με αποτέλεσμα στο Hough space να δημιουργήσουμε κώνους αντί για κύκλους. Ενώ, οι συντεταγμένες του σημείου με τα περισσότερα votes στον accumulator, σε αυτήν την περίπτωση μας προσδιορίζουν τις συντεταγμένες του κέντρου του κύκλου μαζί με την ακτίνα του.

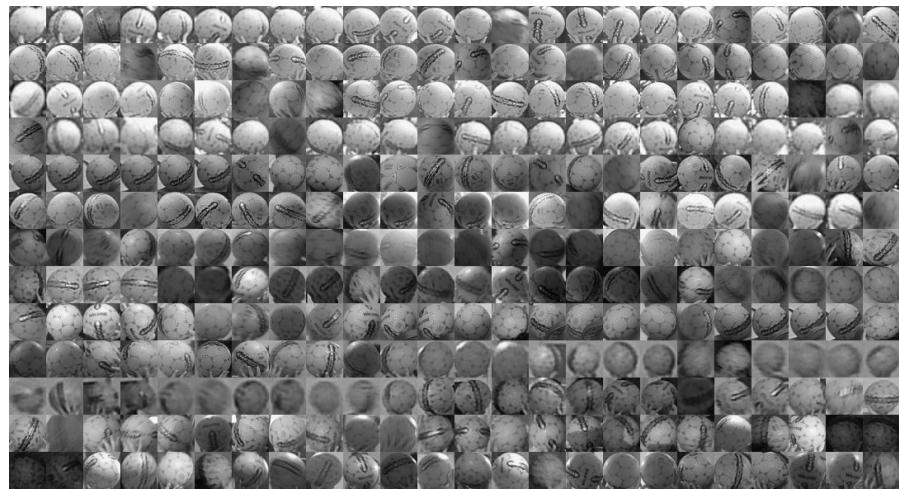
Εναλλακτικές προσεγγίσεις

Τπάρχουν αρκετές εναλλακτικές μεθοδολογίες που θα μπορούσαν επίσης να χρησιμοποιηθούν για το localization του object στο image plane, όπως το background extraction, tracking ενός object που γίνεται defined στην αρχή της λειτουργίας ή ακόμα και machine learning based προσεγγίσεις.

Ένας επιπλέον, σύμφωνα με αυτά, τρόπος που έγινε προσπάθεια να εντοπιστεί - δυστυχώς χωρίς θετικά αποτελέσματα - η μπάλα στην εικόνα, είναι μέσω της τεχνικής haar cascade classifier [136] η οποία αποτελεί μία weak machine learning ταχτική,

που έχει χρησιμοποιηθεί στο παρελθόν σε ανίχνευση της μπάλας σε εφαρμογές όπως για RoboCup [137] [138].

Πρακτικά, σε αυτή χρησιμοποιούνται δύο dataset, ένα θετικό στο οποίο παρουσιάζεται το αντικείμενο ενδιαφέροντος που θέλουμε να εκπαιδεύσουμε το μοντέλο να αναγνωρίζει και ένα αρνητικό στο οποίο δεν βρίσκεται. Για αυτόν τον λόγο, στην προσπάθεια να δημιουργηθεί ένα παρόμοιο μοντέλο για την συγκεκριμένη εφαρμογή πάρθηκαν πάνω από 1000 φωτογραφίες της μπάλας όπως παρουσιάζονται στην *Eikona 4.19* ενώ επίσης υπήρξαν και περίπου 4000 φωτογραφίες για το αρνητικό dataset.



ΕΙΚΟΝΑ 4.19: Παράδειγμα από το θετικό dataset που δημιουργήθηκε, για να γίνει train το μοντέλο του haar cascade.

4.3.4 Εκτίμηση απόστασης

Αφού είχε εντοπιστεί η θέση του αντικείμενο στην εικόνα καθώς και η ακτίνα του σε pixel, μπορούσε να υπολογιστεί και η απόσταση του από την κάμερα. Για τον υπολογισμό της απόστασης της μπάλας, χρησιμοποιείται ως χύρια αρχή ο φορμαλισμός που περιγράφηκε στη *Παράγραφο 3.2.1 - Εξίσωση (3.21)* - η οποία χρειάζεται να μετασχηματιστεί ώστε σαν είσοδο να χρησιμοποιεί πλήθος pixel και όχι mm, συνεπώς μέσω της εξίσωσης (4.6) μπορούμε να υπολογίσουμε την απόσταση που μας ενδιαφέρει, υπολογίζοντας μόνο το πλήθος των pixel που καταλαμβάνει στην εικόνα το αντικείμενο [111] [109].

$$\text{Distance (m)} = \frac{f \text{ (mm)} \times \text{Object real size(m)} \times \text{Image size(pixels)}}{\text{Object size on sensor (pixels)} \times \text{Sensor size(mm)}} \quad (4.6)$$

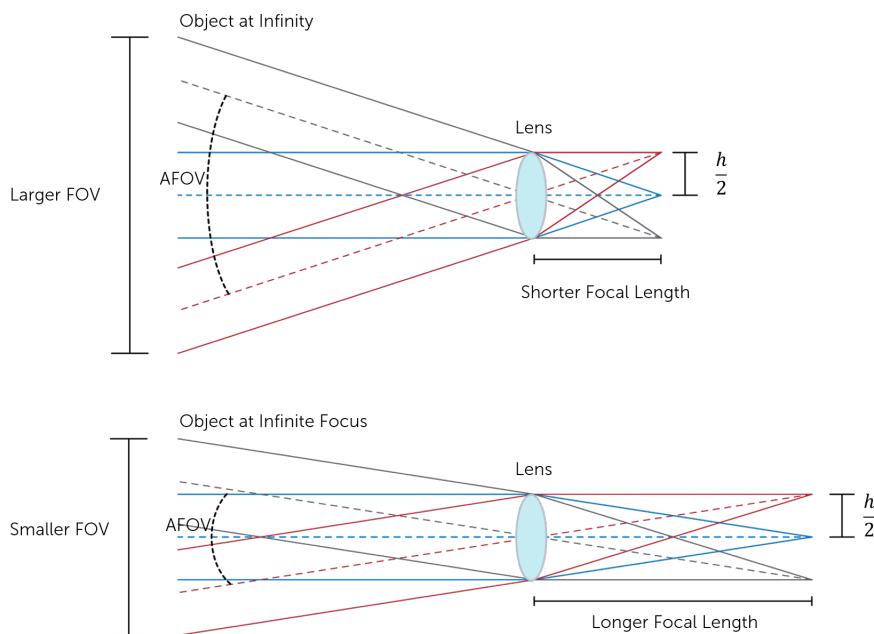
4.3.5 Εκτίμηση γωνίας

Για την εκτίμηση γωνίας ως προς τον x και y άξονα του image plane σε μία calibrated κάμερα μπορεί να χρησιμοποιηθεί η *Εξίσωση (4.7)*, η οποία είναι στην πραγματικότητα υλοποίηση απλής μεθόδου των τριών με βάση το Field of View (FOV) μίας κάμερας. Σε περίπτωση που δεν γνωρίζουμε το FOV μπορεί να υπολογιστεί ως προς την συσχέτιση του Angular Field of View (AFOV) με το Focal length της κάμερας, αυτή η εξάρτηση παρουσιάζεται οπτικά στην *Εικόνα 4.20* και μαθηματικά στην *Εξίσωση (4.8)* [139] και στην συνέχεια μέσω της σχέσης (4.9) μπορούμε να μεταβούμε από το ένα στο άλλο.

$$\text{Angle } (\circ) = \frac{\text{FOV}(\circ) \times \text{distance from origin (pixel)}}{\text{image length(pixel)}} \quad (4.7)$$

$$\text{AFOV(rad)} = 2 \tan^{-1} \frac{\text{image length(mm)}}{2 \times \text{focal length(mm)}} \quad (4.8)$$

$$\text{FOV}(\circ) = \frac{\text{AFOV(rad)} \times 180}{\pi} \quad (4.9)$$



ΕΙΚΟΝΑ 4.20: Συσχέτιση AFOV με focal length ([URL](#))

4.4 Διαμοιρασμός μηνυμάτων

Έχοντας εντοπίσει σε κάθε worker node το αντικείμενο και εκτιμήσει μέσω των παραπάνω εξισώσεων τις πληροφορίες που μας ενδιαφέρουν, επόμενο βήμα είναι ο διαμοιρασμός των μηνυμάτων με κατεύθυνση το master node, για την εκτίμηση της ύστησης. Προκειμένου να περαστεί πληροφορία μέσα στο δίκτυο με node, χρειάστηκε να δημιουργήσουν proprietary πακέτα, στα οποία όμως αποθηκεύονταν αυτές οι πληροφορίες. Συγκεκριμένα κάθε node αφού πραγματοποιήσει όλη την επεξεργασία, αποστέλλει mocap_worker_data πακέτα στο δίκτυο Wi-Fi όπου είναι συνδεδεμένα όλα τα nodes. Αυτά ακολουθούν την δομή που εμφανίζονται στην *Εικόνα 4.21*, η οποία δημιουργήθηκε για τις ανάγκες του συστήματός και μέσα σε αυτή περιλαμβάνονται πληροφορίες σχετικά με τον σειριακό αριθμό του node, την ύσηση του, τα χαρακτηριστικά της κάμερας του, επίσης πληροφορία για κάθε αντικείμενο που ανιχνεύτηκε καθώς και για τον συγχρονισμό των πακέτων.

```

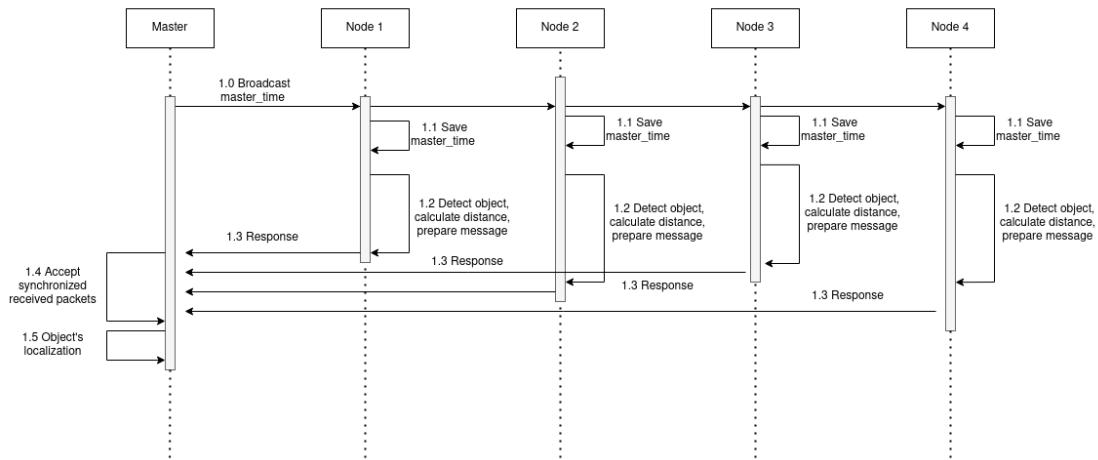
mocap_worker_data
├── header
├── timestamp
├── nodeID
└── master_time
    ├── header
    ├── ros_time
    ├── internal_time
    └── packet_id
└── obj_pose
    ├── x
    ├── y
    ├── z
    ├── roll
    ├── pitch
    └── yaw
└── camera_params
    ├── objectsRealSizeInMeter
    ├── imageHeightInPixels
    ├── imageWidthInPixels
    ├── XFieldOfViewInAngles
    ├── YFieldOfViewInAngles
    ├── XfocalLengthInMillimeters
    ├── YfocalLengthInMillimeters
    ├── XsensorSizeInMillimeters
    └── YsensorSizeInMillimeters
└── detected_ball_data[]
    ├── id
    ├── color
    ├── distance_from_camera
    ├── image_plane_r
    ├── image_plane_x
    ├── image_plane_y
    ├── xangle
    └── ynagle

```

ΕΙΚΟΝΑ 4.21: Η δομή του μηνύματος που αποστέλλει κάθε node

4.5 Συγχρονισμός μηνυμάτων

Σημαντικό είναι η επίτευξη συγχρονισμού μεταξύ των μηνυμάτων που αποστέλλονται. Για αυτό ο master node προγραμματίστηκε να κάνει broadcast, τακτικά, πακέτα τύπου master_time (βλ. μέρος του πακέτου της [Εικόνα 4.21](#)) με το ρολόι το οποίο έχει αυτός. Όταν τα worker nodes λάβουν αυτήν την πληροφορία, την αποθηκεύουν σε εσωτερικό buffer, πραγματοποιούν το detection και localization και στα mocap_worker_data που στέλνουν πίσω, συμπεριλαμβάνουν και αυτήν την πληροφορία. Με αυτόν τον τρόπο ο συγχρονισμός των πακέτων γίνεται με βάση το ρολόι του master, ο οποίος είναι και αυτός που επεξεργάζεται και τα δεδομένα. Η [Εικόνα 4.22](#) προσεγγίζει σε abstract μορφή αυτήν την επικοινωνία. Με βάση την θεώρηση που έγινε για την σχεδίαση του συστήματος, ότι τόσο το αντικείμενο όσο και τα nodes κινούνται με μικρές ταχύτητες - άρα σε μικρό χρονικό διάστημα - δεν υπάρχει μεγάλη μεταβολή της θέσης, επιλέχθηκε ο master να κάνει broadcast τα πακέτα συγχρονισμού με συχνότητα 2Hz, ενώ τα nodes αποστέλλουν πληροφορία στο master με συχνότητα 10Hz. Επίσης οι valid ektimήσεις θέσεις της μπάλας οι οποίες γίνονται broadcast στο σύστημα από το master, γίνονται με συχνότητα 2Hz.



ΕΙΚΟΝΑ 4.22: High level sequence diagram για τον συγχρονισμό των πακέτων

4.6 Εντοπισμός θέσης αντικειμένου

Για να εντοπιστεί τελικά στον τρισδιάστατο χώρο το αντικείμενο, χρησιμοποιείται η αρχή λειτουργίας του Trilateration αλγορίθμου, που αναφέρθηκε στη [Παράγραφο 3.1.2](#). Που όμως επειδή είναι στον 3D χώρο, στον οποίο χρειαζόμαστε τουλάχιστον 4 κόμβους για τον καθορισμό της θέσης, ονομάζεται Multilateration [3.1.2](#). Όπως

αναφέρθηκε και στη *Παράγραφο 3.1.2*, πρακτικά προσπαθούμε για τις εξισώσεις (4.10) με γνωστά τα (x_i, y_i, z_i) και r_i να βρούμε τα (x, y, z) .

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= r_2^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 &= r_3^2 \\ (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 &= r_4^2 \end{aligned} \quad (4.10)$$

Ένας τρόπος προσέγγισης, για επίλυση των εξισώσεων (4.10) είναι να τις αναλύσουμε στις (4.11) και από εκεί να έχουμε να λύσουμε το γραφικό σύστημα πινάκων που περιγράφεται στην (4.12) [140].

$$\begin{aligned} (x^2 + y^2 + z^2) - 2x_1x - 2y_1y - 2z_1z &= r_1^2 - x_1^2 - y_1^2 - z_1^2 \\ (x^2 + y^2 + z^2) - 2x_2x - 2y_2y - 2z_2z &= r_2^2 - x_2^2 - y_2^2 - z_2^2 \\ (x^2 + y^2 + z^2) - 2x_3x - 2y_3y - 2z_3z &= r_3^2 - x_3^2 - y_3^2 - z_3^2 \\ (x^2 + y^2 + z^2) - 2x_4x - 2y_4y - 2z_4z &= r_4^2 - x_4^2 - y_4^2 - z_4^2 \end{aligned} \quad (4.11)$$

$$A = \begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \\ 1 & -2x_4 & -2y_4 & -2z_4 \end{bmatrix} \quad X = \begin{bmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{bmatrix} \quad B = \begin{bmatrix} r_1^2 - x_1^2 - y_1^2 - z_1^2 \\ r_2^2 - x_2^2 - y_2^2 - z_2^2 \\ r_3^2 - x_3^2 - y_3^2 - z_3^2 \\ r_4^2 - x_4^2 - y_4^2 - z_4^2 \end{bmatrix}$$

$$AX = B \quad (4.12)$$

Θετικό με την συγκεκριμένη μέθοδο προσέγγισης - μέσω των εξισώσεων (4.11) - είναι ότι εύκολα το σύστημα μπορεί να γίνει scale καθώς η κάθε στήλη των πινάκων εξαρτάται μόνο από δεδομένα ενός μεμονωμένου node. Άρα παρόλο που στην συγκεκριμένη διπλωματική κάνουμε πειράματα για δεδομένα τεσσάρων θέσεων, με την ίδια ευκολία θα μπορούμε να ήταν και για N όπου $N > 4$.

Η υλοποίηση του συγκεκριμένου αλγορίθμου πραγματοποιήθηκε με την βοήθεια της βιβλιοθήκης Eigen, η οποία αποτελεί και την προτεινόμενη για το ROS βιβλιοθήκη για διαχείριση πινάκων και υπολογισμών σε linear math.

4.7 Τρισδιάστατη απεικόνιση

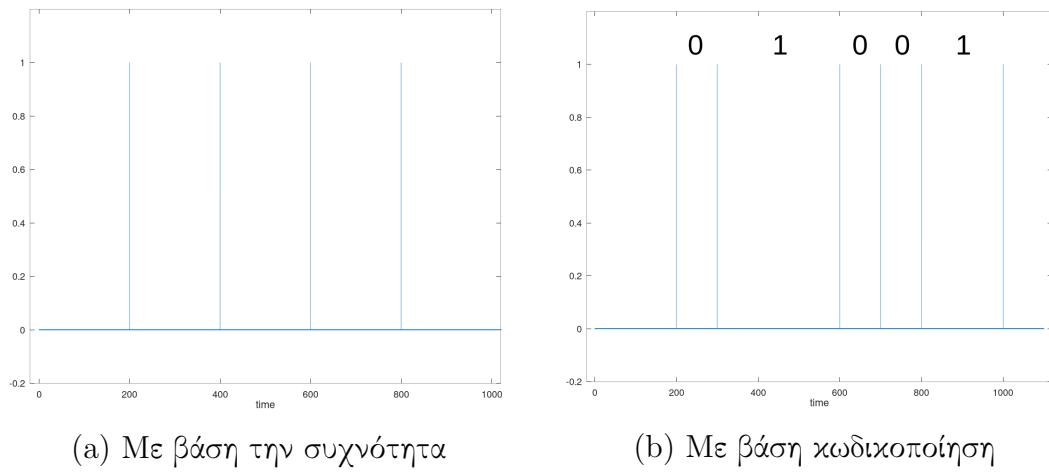
Για την τρισδιάστατη απεικόνιση των δεδομένων που συλλέχθηκαν κατά την διάρκεια των πειραμάτων - που περιγράφονται στο επόμενο κεφάλαιο - χρησιμοποιήθηκε το εργαλείο Matlab, μέσω του οποίου έγιναν οι περισσότερες γραφικές. Σημαντική όμως είναι και η βοήθεια του πακέτου RViz του ROS, το οπόιο σε real time χρόνο μπορεί να οπτικοποιήσει δεδομένα.

4.8 Επαλήθευση ID του detected object

Για λόγους πληρότητας, καθώς και επαλήθευσης του detection, ως τελευταίο σκέλος του συστήματος, προτείνεται ένας τρόπος ώστε να μπορεί να αποδοθεί ID στο αντικείμενο το οποίο ανιχνεύτηκε. Το ID μπορεί να χρησιμοποιηθεί για την πραγματοποίηση localization συγκεκριμένου αντικείμενου - σε περίπτωση που υπάρχουν περισσότερα από ένα στο scene.

Ο τρόπος που προσεγγίστηκε το πρόβλημα είναι με την ανίχνευση ενός γεγονότος στο stream της κάμερας. Συγκεκριμένα, αφού ανιχνευτεί το αντικείμενο όπως επεξηγήθηκε στο παρόν κεφάλαιο, επιλέγεται εσωτερικά των εξωτερικών διαστάσεων του αντικειμένου να φωτοβολεί ένα led για τον προσδιορισμό του γεγονός αναφοράς. Αυτό μπορεί να είναι περιοδικό είτε κωδικοποιημένο - βλ. *Εικόνα 4.23* - ώστε ο προσδιορισμός του ID να μετατραπεί σε ένα χρονικό πρόβλημα.

Ακολουθείται παρόμοια διαδικασία όπως περιγράφηκε στη *Παράγραφο 4.3.3*, όμως για το led που βρίσκεται εσωτερικά του bounding box που περικλείει η μπάλα. Πλέον το εμβαδόν του bounding box για το led, όταν δεν είναι αναμμένο αυτό^{*} είναι μηδενικό, ενώ σε αντίθετη περίπτωση έχει μία μη μηδενική τιμή. Με τον μετασχηματισμό του προβλήματος, πλέον, μετρώντας τον χρόνο ενδιάμεσα μηδενικών τιμών για το εμβαδόν του bounding box που δημιουργείται λόγο του led, είναι εφικτός ο προσδιορισμός μοναδικού ID.



EIKONA 4.23: Παραδείγματα προσεγγίσεων καθορισμού του ID

Στην συγκεκριμένη εργασία επιλέχθηκε ο προσδιορισμός να γίνει με βάση την σταθερή συχνότητα με την οποία κάνει blink το led.

Κεφάλαιο 5

Επαλήθευση Λειτουργίας και Αποτελέσματα

”Η διαδικασία δοκιμών ενός συστήματος είναι μία διαδικασία εκμάθησης”

Brian Mahick

Σε αυτό το σημείο περιγράφονται οι ενέργειες που ακολουθήθηκαν, προκειμένου να επαληθευτεί η απόδοση του συστήματος. Η διαδικασία δοκιμών χωρίστηκε σε τρία στάδια. Το πρώτο, το οποίο γινόταν σε indoor περιβάλλον ταυτόχρονα με την υλοποίηση του συστήματος ώστε να επαληθευτεί η δυνατότητα εντοπισμού της μπάλας σε κάθε frame του feed της κάμερας¹. Την δεύτερη που αποτελεί το κομμάτι δοκιμών ενός μεμονωμένου node σε outdoor scenarios για το localization στο image plane του αντικειμένου με βάση την μέθοδο εντοπισμού που επιλέχθηκε². Τελευταίο κομμάτι ήταν η δοκιμή ολόκληρου του συστήματος.

Στην συνέχεια του κεφαλαίου, δίνονται αναλυτικά στοιχεία για την δεύτερη και τρίτη φάση δοκιμών και όχι για την πρώτη, καθώς περιλαμβάνονται σε αυτές μετρικές της πρώτης. Ενώ, για την ανίχνευση του object στο εκάστοτε καρέ χρησιμοποιείται η διαδικασία με τον HSV μετασχηματισμό (βλ. Παράγραφο 4.3.3).

5.1 Επαλήθευση λειτουργίας μεμονωμένου node

5.1.1 Περιβάλλον δοκιμών

Για την επαλήθευση λειτουργίας του μεμονωμένου κόμβου, η διαδικασία που ακολουθήθηκε είναι η εξής. Χρησιμοποιήθηκαν δύο αντικείμενα, το υπό έλεγχο σύστημα

¹Βίντεο από την διαδικασία μπορεί να βρεθεί [141]

²Βίντεο από την διαδικασία μπορεί να βρεθεί [142]

- *Εικόνα 5.1* (a) - το οποίο κατά όλη την διάρκεια του πειράματος ήταν στατικό σε συγκεκριμένο σημείο, και το κινούμενο αντικείμενου (η κίτρινη μπάλα), η θέση του οποίου έγινε προσπάθεια κάθε χρονική στιγμή να εκτιμηθεί - *Εικόνα 5.1* (b).



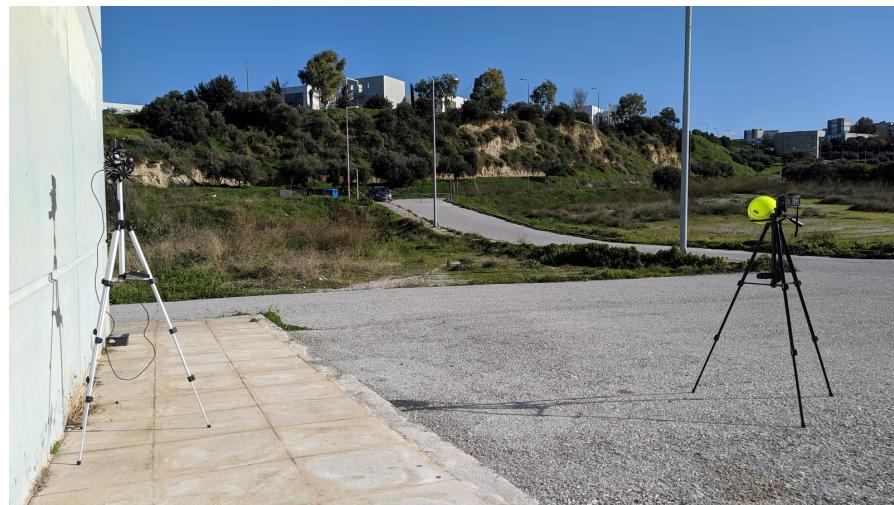
(a) Υπό έλεγχο σύστημα (Node) το οποίο τροφοδοτείται από powerbank κατά την διάρκεια πειραμάτων



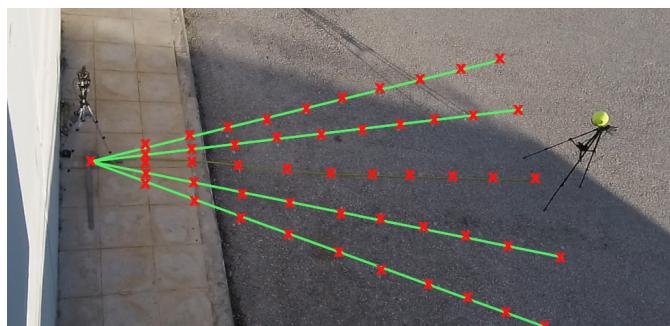
(b) Κινούμενο αντικείμενο εκτίμησης θέσης μαζί με το όργανο μέτρησης απόστασης

ΕΙΚΟΝΑ 5.1: Εξοπλισμός που χρησιμοποιήθηκε στην πρώτη εξωτερική πειραματική φάση

Τα δύο αυτά αντικείμενα τοποθετήθηκαν το ένα απέναντι από το άλλο όπως φαίνεται στην *Εικόνα 5.2*, και πάρθηκαν μετρήσεις απόστασης για διαφορετικές γωνίες όπως φαίνεται στην *Εικόνα 5.3*. Ταυτόχρονα, το σύστημα κατά την διάρκεια της διαδικασίας, ήταν σε πλήρη λειτουργία, και πραγματοποιούσε ανίχνευση και εκτίμηση της απόστασης του αντικειμένου από την κάμερα. Στη *Παράγραφο 5.1.3* παρουσιάζονται τα πειραματικά αποτελέσματα των μετρήσεων αυτής της διαδικασία, ενώ στη *Παράγραφο 5.1.2* παρουσιάζονται οι ανάγκες λειτουργίας από την σκοπιά του ενσωματωμένου συστήματος.



ΕΙΚΟΝΑ 5.2: Χωρική τοποθέτηση του υπό ελέγχου συστήματος και αντικειμένου εκτίμησης θέσης



ΕΙΚΟΝΑ 5.3: Αναπαράσταση των θέσεων στις οποίες έγιναν οι μετρήσεις του πειράματος

Όπως φαίνεται στην Εικόνα 5.1 (b), το object το οποίο πρόκειται να εντοπίσουμε βρίσκεται σε τρίποδο, για μεγαλύτερη ευκολία των μετρήσεων, ενώ στο πλάγιο μέρος γίνεται διακριτό το όργανο που χρησιμοποιήθηκε για τον υπολογισμό της απόστασης μεταξύ των δύο αντικειμένων. Το όργανο αυτό είναι το laser range finder της Bosch GLM 40 - Εικόνα 5.4 - με δυνατότητες μέτρησης 0.15 – 40.00m και απόκλιση μετρήσεων $\pm 1.5\text{mm}$

Για τον υπολογισμό των διάφορων γωνιών από τις οποίες θα γινόντουσαν μετρήσεις, χρησιμοποιήθηκε το όργανο μέτρησης γωνιών όπως φαίνεται στην Εικόνα 5.5. Ενώ οι μετρήσεις που στην συνέχεια θα αναγερθούν αφορούν δεδομένα από γωνίες

$[-20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ]$ ³ και αποστάσεις μέτρησης ανά 50cm μέχρι τα 5m (εκτός από την κεντρική ευθεία στην οποία έγιναν μετρήσεις μέχρι περίπου τα 8m).



ΕΙΚΟΝΑ 5.4: Το ψηφιακό λέιζερ μέτρησης απόστασης που χρησιμοποιήθη (Bosch GLM 40) ([URL](#))



ΕΙΚΟΝΑ 5.5: Υπολογισμός των γωνιών με χρήση εργαλείου μέτρησης γωνίας

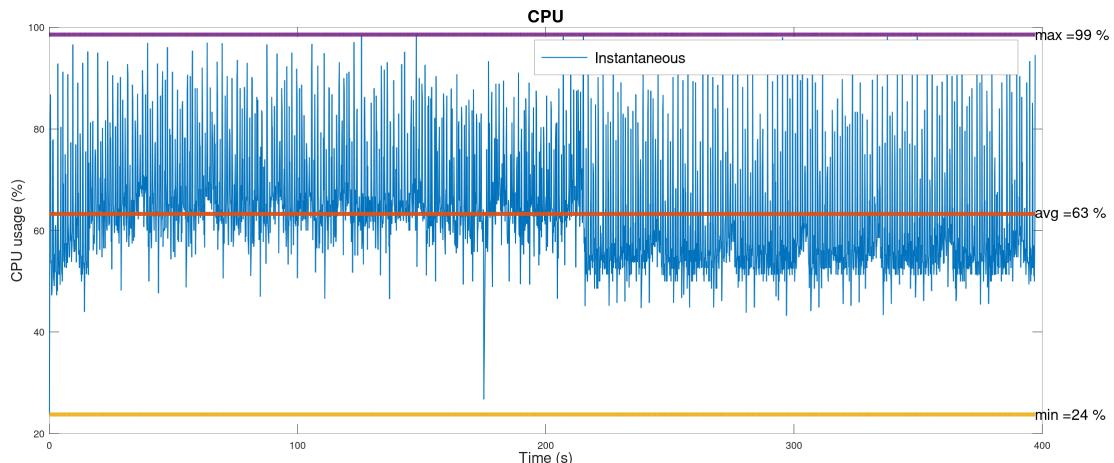
5.1.2 Απόδοση του συστήματος

Πριν αναφερθούν τα δεδομένα που συλλέχθηκαν ως προς την απόδοση του συστήματος, θα αναφερθούν κάποια από τα χαρακτηριστικά λειτουργίας του. Η υλοποίηση

³Λόγω του διαφορετικού ύψους μεταξύ της θέσης που μετρήθηκαν οι γωνίες και της κάμερας, δεν είναι οι ίδιες με αυτές που ανιχνεύονται στο χάρονα της εικόνας

του συστήματος έγινε σε C++. Δημιουργήθηκαν συνεπώς Function-like macros για εξαγωγή και αποθήκευση χρήσιμων πληροφοριών του συστήματος. Οι πληροφορίες που συλλέχθηκαν, αναφέρονται σε αυτήν την ενότητα, και σχετίζονται με τις επεξεργαστικές ανάγκες του συστήματος, η μνήμη μου χρειάζεται για να λειτουργήσει, και άλλες σημαντικές πληροφορίες.

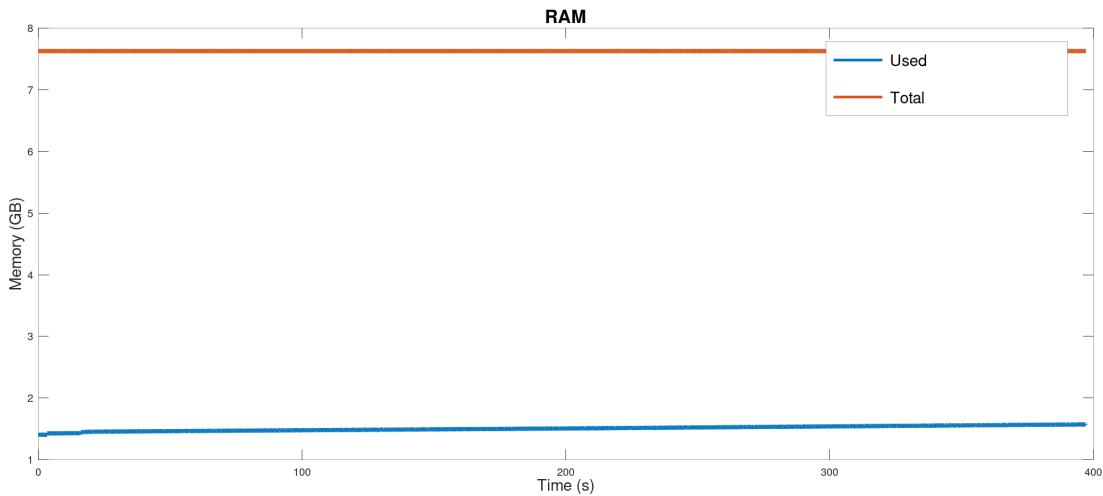
Πρώτη από αυτές τις πληροφορίες είναι οι επεξεργαστικές ανάγκης του, όπου στην *Εικόνα 5.6* φαίνονται τα δεδομένα που συλλέχθηκαν κατά την διάρκεια ενός από τα πειράματα.



ΕΙΚΟΝΑ 5.6: Επεξεργαστική ισχύς του συστήματος

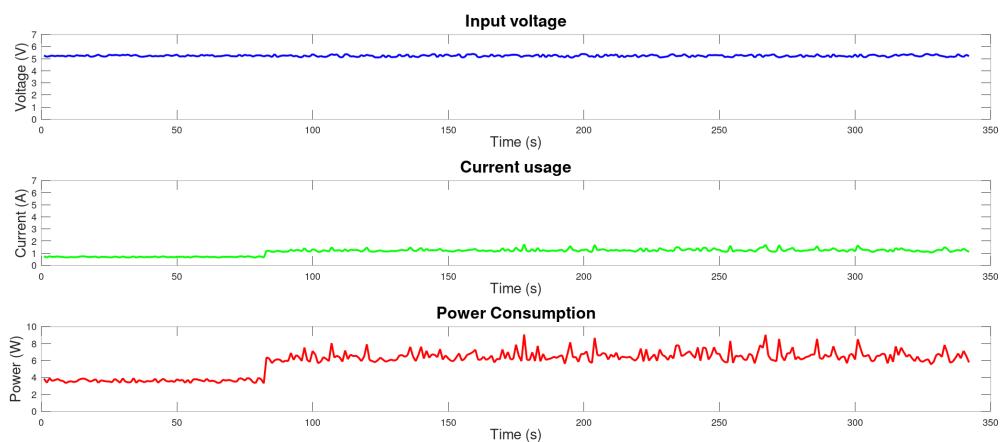
Μία ακόμα χρήσιμη μετρική είναι αυτή της μνήμης που χρησιμοποιεί το σύστημα, στην *Εικόνα 5.7* παρουσιάζεται η συγκεκριμένη πληροφορία. Από αυτό το διάγραμμα μπορούμε να παρατηρήσουμε ότι για την συγκεκριμένη υλοποίηση το σύστημα δεν έχει μεγάλες ανάγκες μνήμης⁴.

⁴Έχοντας αυτήν την πληροφορία, μπορούμε να μεταβούμε σε επιλογή Embedded Linux System με μικρότερη μνήμη, που σημαίνει μικρότερο κόστος ανά node συστήματος



ΕΙΚΟΝΑ 5.7: Ανάγκες μνήμης του συστήματος

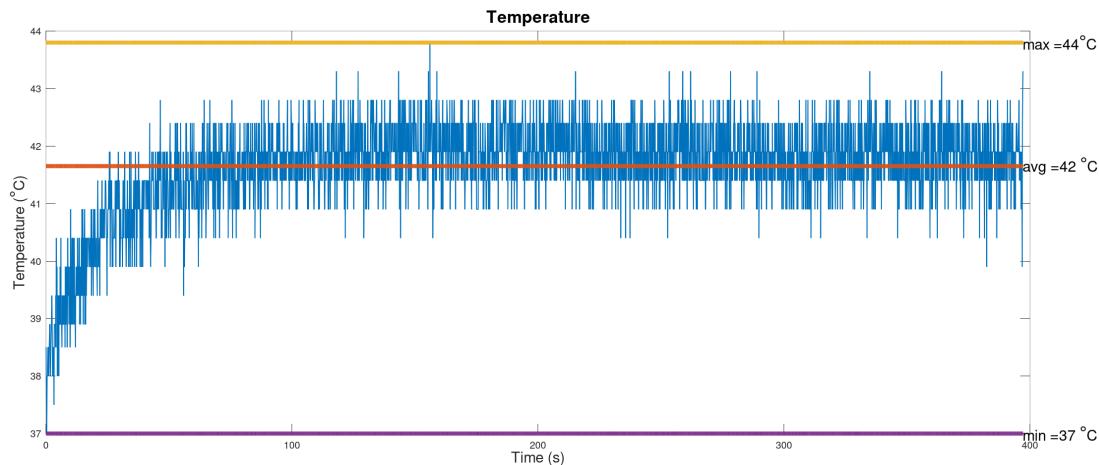
Το σύστημα αυτό σχεδιάζεται με γνώμονα μελλοντικά να χρησιμοποιηθεί σε drones, συνεπώς οι ενεργειακές απαιτήσεις του είναι σημαντικές. Για αυτό τον λόγο εξάχθηκαν και δεδομένα σχετικά με την κατανάλωση του συστήματος⁵ - Εικόνα 5.8. Με την μέγιστη κατανάλωση που παρατηρήθηκε από τροφοδοσία μέσω power bank κατά την διάρκεια δοκιμών είναι τα 10Watt, ενώ τάση τροφοδοσίας είναι τα 5Volt. Τα multitorotor - με βάση τα οποία σχεδιάζεται το σύστημα - τυπικά με τις μπαταρίες που χρησιμοποιούν έχουν αυτονομία περίπου της τάξης των 30 λεπτών. Η ανάγκες ενέργειας του προτεινόμενου συστήματος παρατηρείται να είναι στα πλαίσια που μπορεί να τροφοδοτηθεί από την μπαταρία του UAV χωρίς να επιβαρύνει την αυτονομία του, καθώς, δεν είναι στην ίδια τάξη μεγέθους με τις ανάγκες ισχύς που χρειάζονται οι κινητήρες.



ΕΙΚΟΝΑ 5.8: Ενεργειακές απαιτήσεις συστήματος

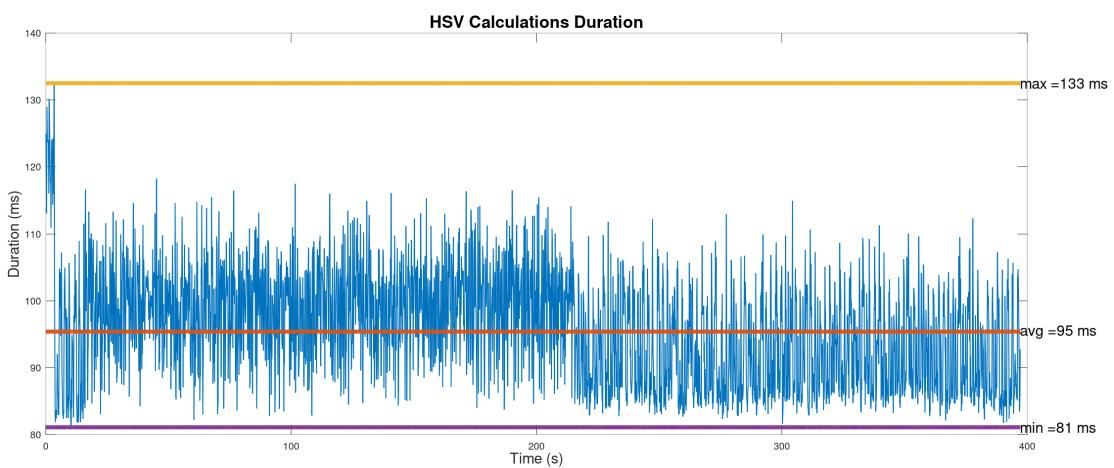
⁵Στις συγκεκριμένες γραφικές, μέχρι το 80s το σύστημα είναι σε idle mode, ενώ από εκεί και έπειτα είναι σε πλήρη λειτουργία

Σημαντικός παράγοντας της απόδοσης ενός επεξεργαστή, είναι η θερμοκρασία του, καθώς αν είναι αυξημένη μπορεί η απόδοση να μειωθεί λόγω thermal throttling. Για αυτό συλλέχθηκε επίσης πληροφορία για την θερμοκρασία του επεξεργαστή - *Εικόνα 5.9.*



ΕΙΚΟΝΑ 5.9: Θερμοκρασίες συστήματος έχοντας σε χρήση το fan του breakout board

Ενώ τελευταία, αλλά εξίσου σημαντική μετρική, είναι ο υπολογισμός του χρόνου που χρειάζεται το σύστημα ανίχνευσης του αντικειμένου, για κάθε μεμονωμένη χρονική στιγμή χρήσης του συστήματος, και παρουσιάζεται στην *Εικόνα 5.10.*

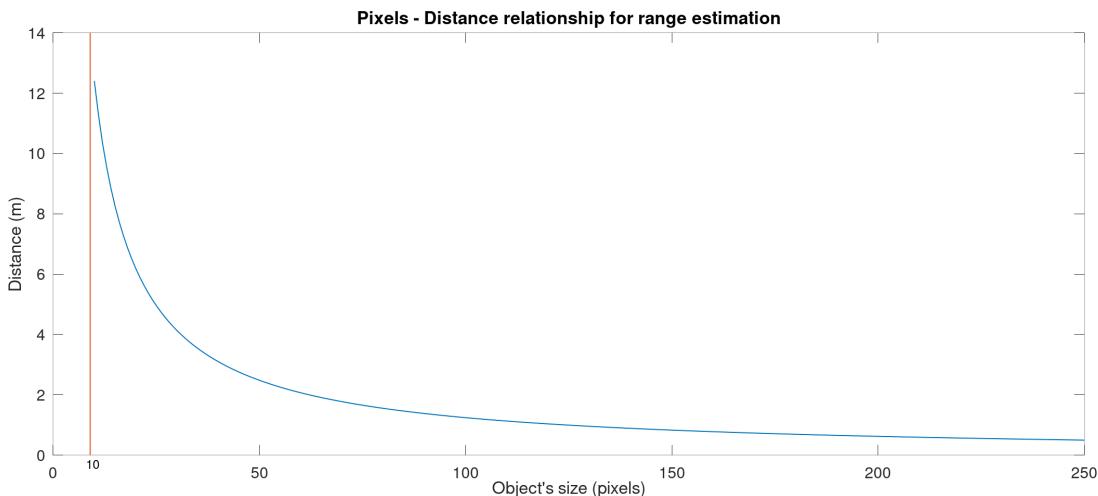


ΕΙΚΟΝΑ 5.10: Χρόνος του object detection μέσα σε κάθε μεμονωμένη χρονική στιγμή χρήσης του συστήματος

5.1.3 Λαμβανόμενα δεδομένα και απεικόνιση

Πριν αναφερθούν τα δεδομένα που λήφθηκαν κατά το πείραμα, είναι σημαντικό να γίνει κατανοητή η εξάρτηση pixel-απόστασης της σχέσης (4.6) που αναλύθηκε στο προηγούμενο κεφάλαιο και χρησιμοποιείται για το range estimation του αντικειμένου.

Η Εικόνα 5.11 παρουσιάζει την παραπάνω αναλογία, για τις παραμέτρους της κάμερας που χρησιμοποιήθηκε, σε αναλύσεις 1280x720 pixels.



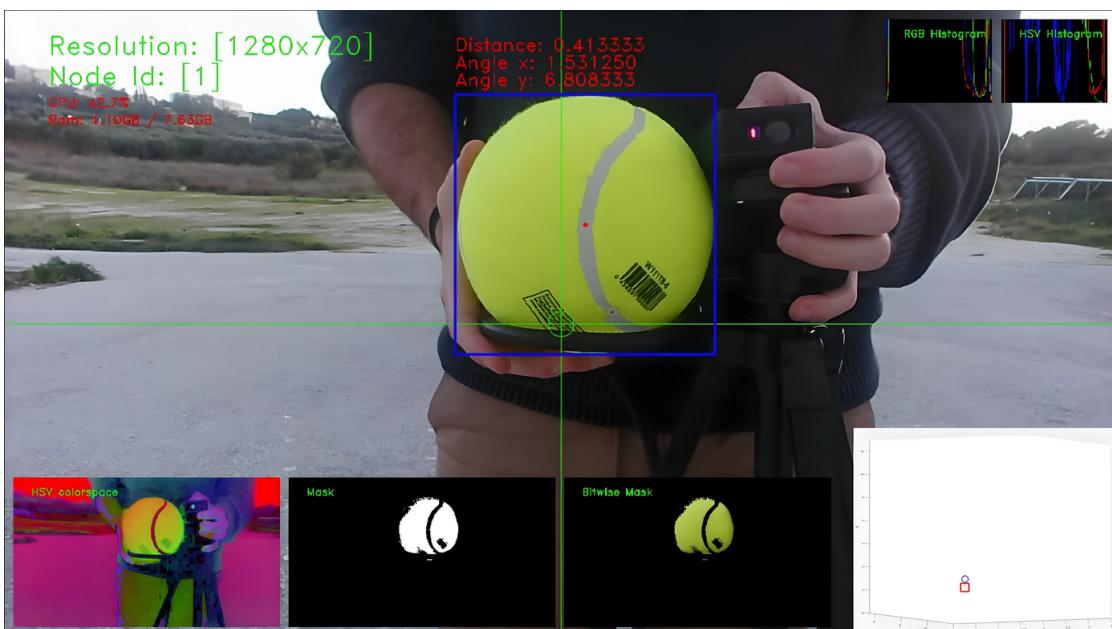
ΕΙΚΟΝΑ 5.11: Εξάρτηση του μεγέθους του αντικειμένου σε pixel με την απόσταση του από την κάμερα

Αυτό που μπορούμε να παρατηρήσουμε είναι, ότι δεν είναι γραμμική. Όμως, μπορεί να προσεγγιστεί σε ένα υποσύνολο του πεδίο ορισμού της, με γραμμικό τρόπο. Αυτό σημαίνει ότι για σχετικά μεγάλες ακτίνες η μεταβολή σε pixel επιφέρει μικρές μεταβολές στην απόσταση, ενώ για μικρές ακτίνες επιφέρει μεγάλες μεταβολές της απόστασης.

Για λόγους πληρότητας θα δούμονταν τρία παραδείγματα. Σε πολύ μικρές ακτίνες - λόγου χάρη για αντικείμενο ακτίνας 10 pixel - η εκτιμώμενη απόσταση είναι 12.4m, ενώ, αν το αντικείμενο εκτιμούσαμε ότι είχε ακτίνα 11 pixel, τότε η απόσταση θα ήταν 11.27m. Ως αποτέλεσμα, σε αυτήν την περίπτωση το 1 pixel σφάλματος του αντικειμένου στο image plane να καταλήγει σε 1.13m σφάλματος της εκτιμώμενης απόστασης στον φυσικό χώρο. Για λίγο μεγαλύτερες ακτίνες, όπως για 50 pixel, η εκτιμώμενη απόσταση είναι 2.48m. Αν όμως για αυτή την ακτίνα του αντικειμένου είχαμε σφάλμα 1 pixel - δηλαδή μέτρηση 51 pixel - τότε η εκτιμώμενη απόσταση θα ήταν τα 2.43m, με τελικό σφάλμα τα 5cm σε αυτήν την περίπτωση. Τέλος, σε αρκετά μεγαλύτερες ακτίνες, με τα 100 pixel ως αναφορά, έχουμε εκτιμώμενη απόσταση τα 1.24m, ενώ, σε περίπτωση που η ακτίνα μας έχει 1 pixel διαφορά από

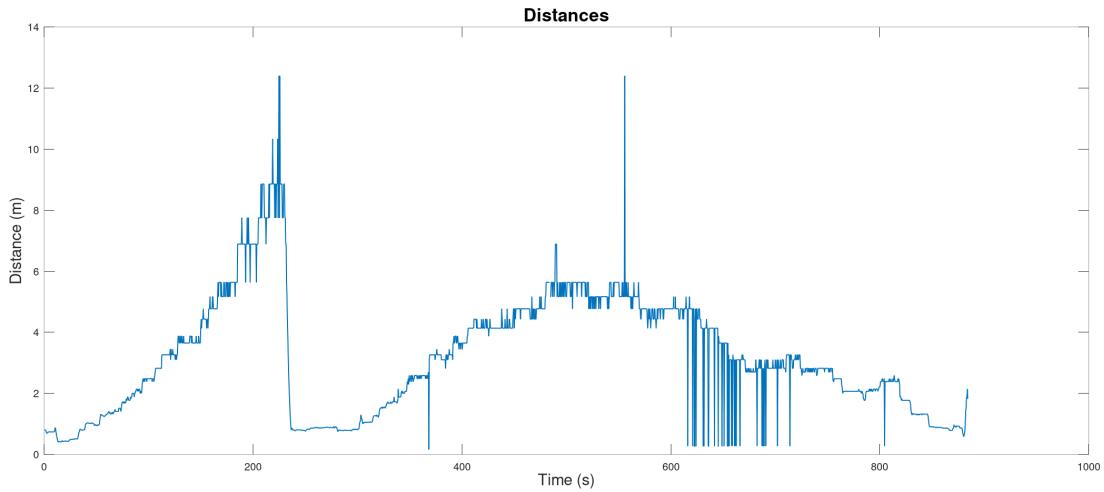
την πραγματικότητα - δηλαδή έστω 101 pixels - και τα 1.22m εκτιμώμενη απόσταση, καταλήγουμε σε 2cm σφάλμα. Αυτό που βγαίνει ως συμπέρασμα από αυτήν την σχέση είναι ότι το τελικό σφάλμα εκτίμησης είναι σε άμεση εξάρτηση με το ποσοστό του image plane που καταλαμβάνει το αντικείμενο σε αυτό.

Αφού έχει γίνει ξεκάθαρος αυτός ο περιορισμός, θα αναφερθούν τα δεδομένα που λήφθηκαν κατά την διάρκεια του πειράματος, από το node. Αρχικά, στην *Eικόνα 5.12* παρουσιάζεται στιγμιότυπο της αντιληπτικής ικανότητας - μεμονωμένης χρονικής στιγμής - του node, μαζί με τα διάφορα στάδια επεξεργασίας, σε συνδυασμό με post-processing τρισδιάστατη απεικόνιση της μπάλας σε σχέση με την κάμερα (κάτω δεξιά στην εικόνα).

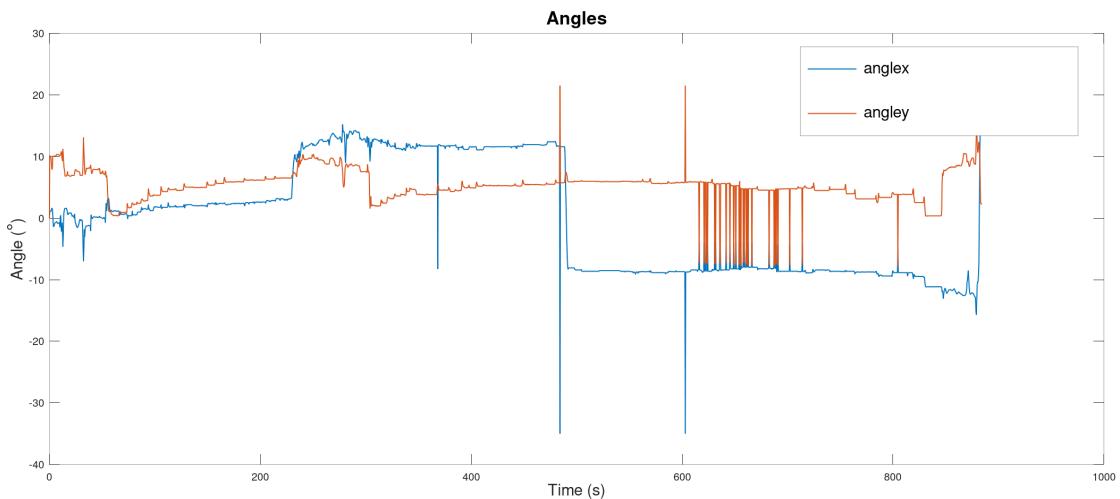


ΕΙΚΟΝΑ 5.12: Αντιληπτική ικανότητα μεμονωμένου κόμβου κατά την διάρκεια των δοκιμών, σε συνδυασμό με τρισδιάστατη απεικόνιση της μπάλας

Οι γραφικές - *Εικόνα 5.13* και *Εικόνα 5.14* - παρουσιάζουν τα raw data για την απόσταση και γωνίες που συλλέχθηκαν κατά την διάρκεια του πειράματος. Με την βοήθεια αυτών, μπορεί ακόμα και να φανεί η διαδρομή που ακολουθήθηκε κατά την διάρκεια του πειράματος. Επίσης, παρατηρείται να είναι σε ένα αρκετά μεγάλο ποσοστό χοντά στην πραγματική διαδρομή, με πολύ καλή προσέγγιση για τον εντοπισμό του αντικειμένου (όπως φανεί στην συνέχεια). Για τις οποίες, είναι σημαντικό όμως να αναφερθεί ότι τα spikes αναπαριστούν σημεία όπου δεν είχε γίνει ακριβές detection του object.

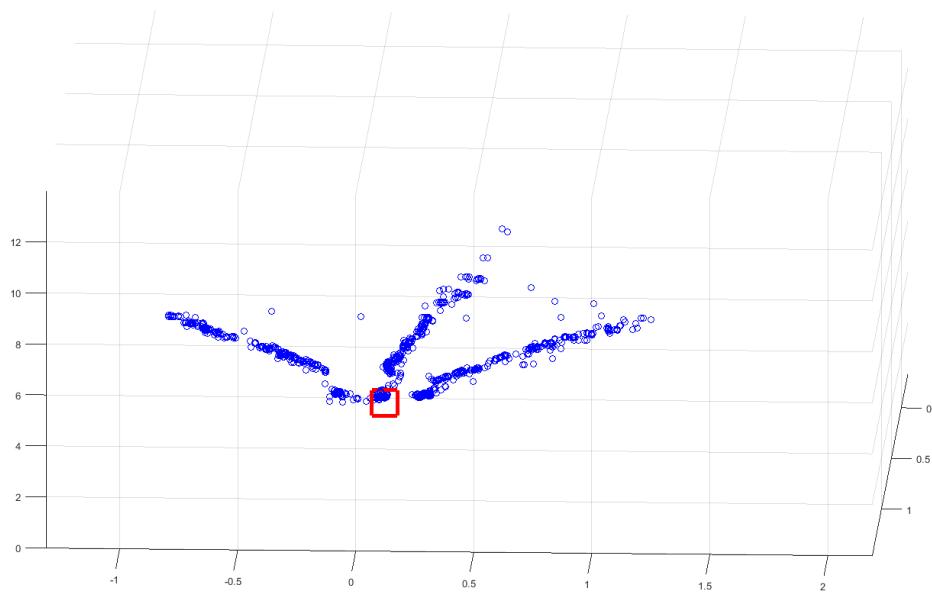


ΕΙΚΟΝΑ 5.13: Εκτίμηση απόστασης για κάθε δευτερόλεπτο χρήσης του συστήματος (για 3 από τις 5 ευθείες)



ΕΙΚΟΝΑ 5.14: Εκτίμηση γωνιών στους άξονες x και y για κάθε δευτερόλεπτο χρήσης του συστήματος (για 3 από τις 5 ευθείες)

Με βάση τις δύο παραπάνω γραφικές, για την εκτίμηση απόστασης και γωνιών οι οποίες αποτελούν scalar τιμές, μπορούμε σε συνδυασμό - με απλή γεωμετρία - να αναπαραστήσουμε στο τρισδιάστατο χώρο όλες τις εκτιμήσεις για τις θέσης του αντικειμένου και να καταλήξουμε σε ένα vector θέσης ως προς την κάμερα. Αυτό γίνεται προσπάθεια να επιτευχθεί στην *Εικόνα 5.15*. Θεωρητικά θα μπορούσαμε μόνο με αυτόν τον τρόπο να εντοπίσουμε το αντικείμενο στον 3D χώρο, για λόγους αξιοπιστίας όμως αυθούμαστε στον προσδιορισμό της θέσης από δεδομένα πολλαπλών κόμβων - από swarm - ώστε σε μία πιθανή αποστολή να μην έχουμε single point of failure.



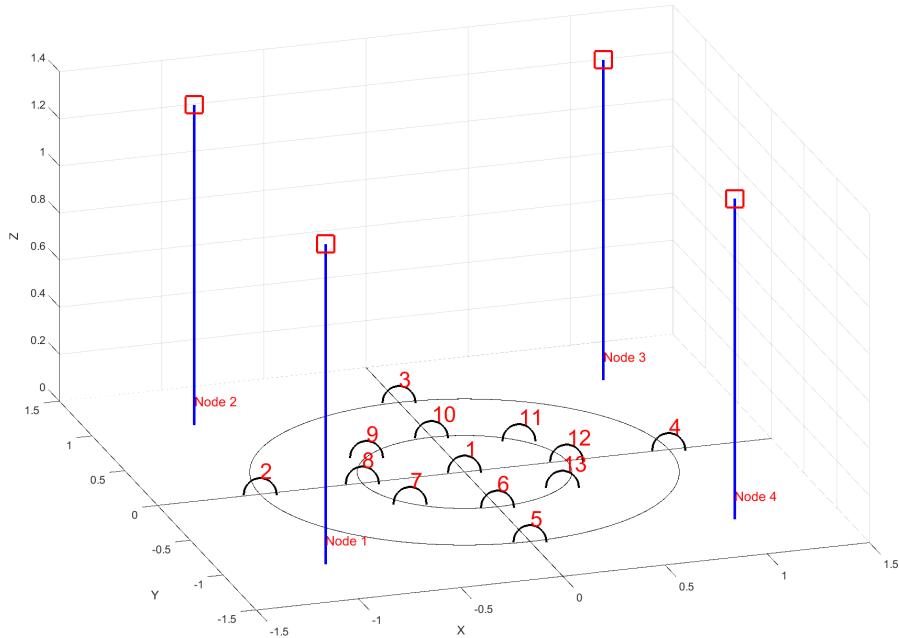
ΕΙΚΟΝΑ 5.15: Τρισδιάστατη απεικόνιση των μετρήσεων για 3 από τις 5 ευθείες

5.2 Δεδομένα πολλαπλών κόμβων

Όπως αναφέρθηκε και στη Παράγραφο 4.1.6, η λειτουργία του GPS - τουλάχιστον σε αυτήν την φάση - ήταν για λόγους πληρότητας της υλοποίησης, ώστε να γίνει εύκολο το integration ενός GPS με μικρότερο σφάλμα - όπως RTK GPS - σε μελλοντική υλοποίηση. Για αυτόν τον λόγο στην συνέχεια αναφερόμαστε σε καθορισμένων συντεταγμένων nodes.

5.2.1 Διαδικασία λήψης δεδομένων

Στην συνέχεια πραγματοποιήθηκαν μετρήσεις από πολλαπλά σημεία (κόκκινα τετράγωνα στο διάγραμμα) για το ίδιο αντικείμενο όπως παρουσιάζεται στην Εικόνα 5.16. Με αυτόν τον τρόπο, για την ίδια ύλη της μπάλας (με μαύρο χρώμα στο διάγραμμα) θα μπορούσε να προσομοιωθούν - σε ένα πρώιμο τουλάχιστον στάδιο - τα δεδομένα που δημιουργούνται από πολλαπλούς κόμβους σε ένα συγκρότημα από drone.



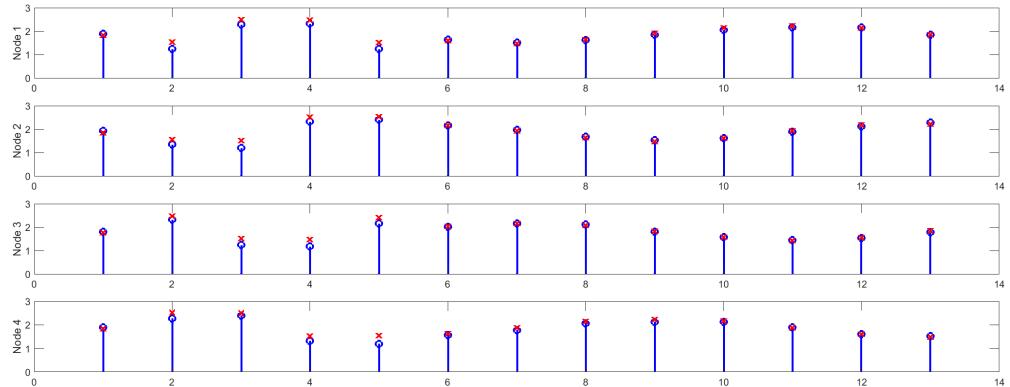
ΕΙΚΟΝΑ 5.16: Απεικόνιση των θέσεων από τις οποίες πραγματοποιήθηκαν λήψεις, καθώς και οι θέσεις του αντικειμένου

Για κάθε μία από αυτές τις θέσεις πάρθηκαν μετρήσεις απόστασης μέσω που laser range finder που αναφέρθηκε παραπάνω (Εικόνα 5.4) καθώς επίσης καταγράφηκαν και οι εκτιμήσεις αποστάσεις που πραγματοποιήθηκαν από το σύστημα. Στο Πίνακα 5.1 παρουσιάζονται αναλυτικά αυτές οι μετρήσεις, ενώ στην Εικόνα 5.17 γίνεται οπτικοποίηση τους.

Επίσης, στην Εικόνα 5.18 δίνεται παράδειγμα της αντιληπτικής ικανότητας του κάθε κόμβου, όταν έχει τοποθετηθεί στην θέση 2 (βλ. Εικόνα 5.16) το αντικείμενο.

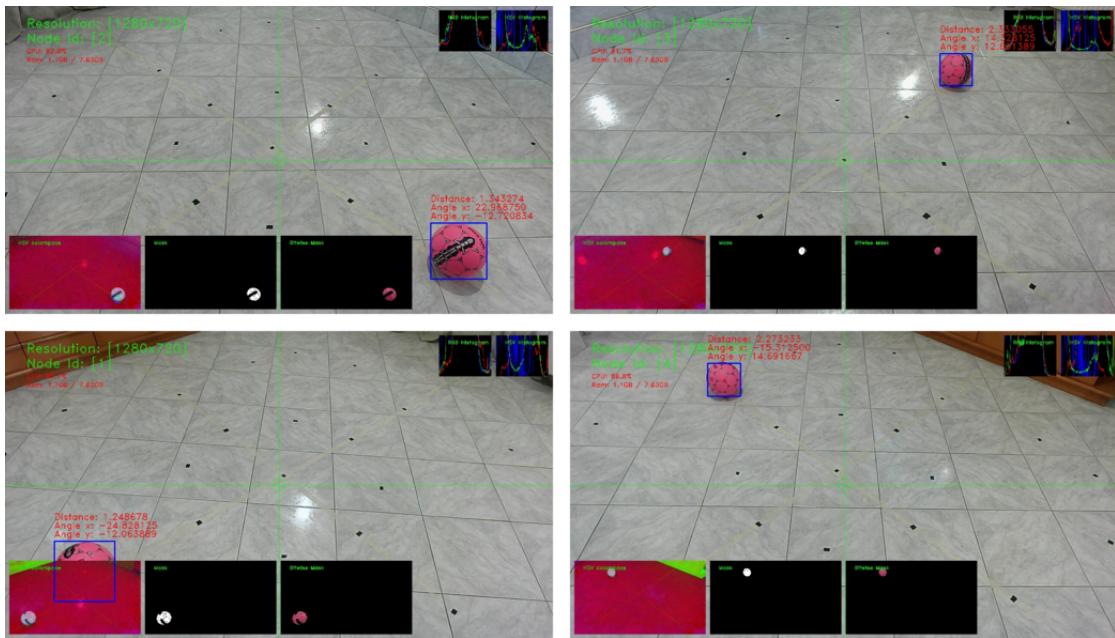
ΠΙΝΑΚΑΣ 5.1: Πραγματικές αποστάσεις και εκτιμήσεις για κάθε γωνία λήψης

	Node 1		Node 2		Node 3		Node 4	
pos	Estimation	Real	Estimation	Real	Estimation	Real	Estimation	Real
1	1.886	1.820	1.927	1.847	1.809	1.765	1.886	1.839
2	1.248	1.539	1.343	1.560	2.333	2.475	2.273	2.510
3	2.273	2.488	1.198	1.507	1.231	1.514	2.396	2.499
4	2.333	2.471	2.333	2.519	1.182	1.469	1.323	1.515
5	1.248	1.510	2.396	2.522	2.162	2.396	1.182	1.532
6	1.641	1.593	2.162	2.146	2.014	2.040	1.555	1.617
7	1.502	1.473	1.970	1.927	2.162	2.157	1.773	1.870
8	1.611	1.627	1.672	1.643	2.110	2.071	2.061	2.139
9	1.847	1.914	1.528	1.478	1.809	1.831	2.110	2.223
10	2.061	2.129	1.611	1.610	1.583	1.568	2.110	2.155
11	2.162	2.209	1.886	1.919	1.453	1.427	1.886	1.877
12	2.162	2.138	2.110	2.168	1.528	1.547	1.611	1.591
13	1.847	1.837	2.273	2.225	1.773	1.836	1.528	1.488



ΕΙΚΟΝΑ 5.17: Γραφική αναπαράσταση της εκτίμησης θέσης - πραγματικής απόστασης (Σύμφωνα με τα δεδομένα του Πίνακα 5.1, όπου οι εκτιμήσεις και οι πραγματικές αποστάσεις)

Από τις παραπάνω μετρήσεις καταλήγουμε σε συνολικό μέσο όρο σφάλματος τα 0.0974 m. Αν διαχωρίσουμε το σφάλμα, με βάση τις εξωτερικές (2-5) και εσωτερικές (1 και 6-13) θέσεις τότε μπορούμε να καταλήξουμε σε δύο επιπλέον σφάλματα. Για τις εξωτερικές ο μέσος όρος των σφαλμάτων είναι τα 0.2230m ενώ για τις εσωτερικές τα 0.0686m. Ο λόγος που συμβαίνει αυτό, σχετίζεται με την παραμόρφωση που προκαλεί ο φακός της κάμερας (με μεγαλύτερες επιδράσεις προς τα άκρα του image plane), και δεν έγινε εφικτό μέσω του calibration να διορθωθεί.



ΕΙΚΟΝΑ 5.18: Αντιληπτική ικανότητα από κάθε γωνία λήψης για την θέση 2

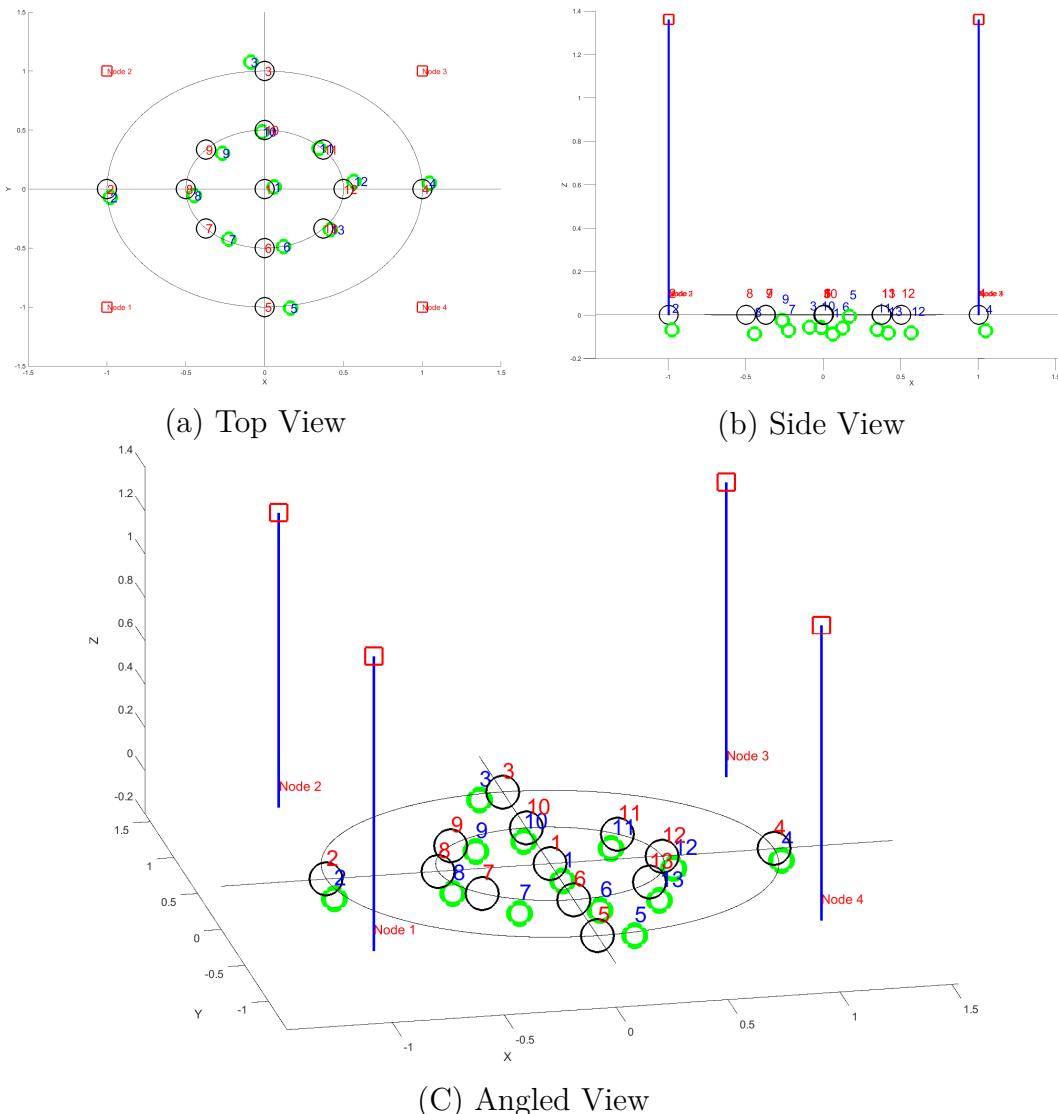
5.2.2 Εκτίμηση θέσης και απεικόνιση

Έχοντας συλλέξει τα δεδομένα που αναφέρθηκαν στη Παράγραφο 5.2.1, μπορούσε να χρησιμοποιηθεί ο φορμαλισμός που περιγράφηκε στη Παράγραφο 4.6 για την εκτίμηση της θέσης στον τρισδιάστατο χώρο.

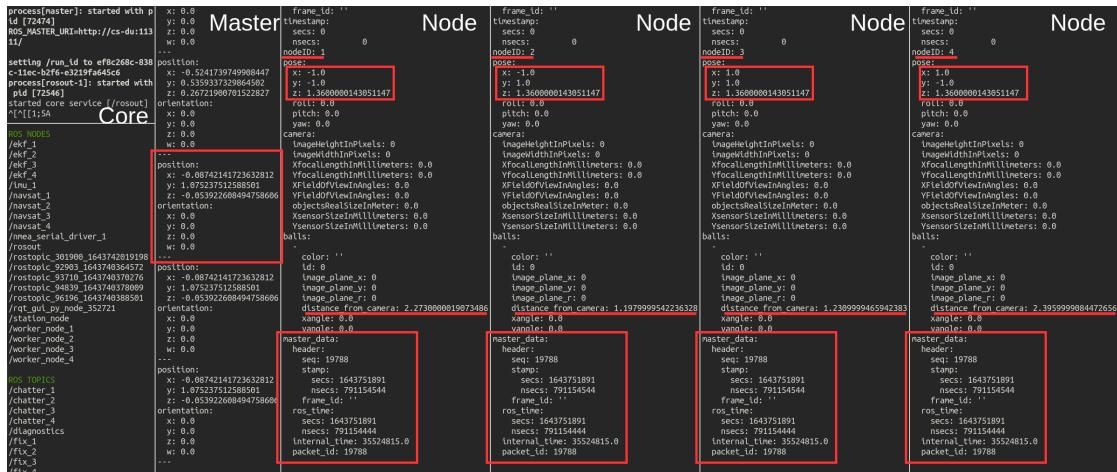
Χρησιμοποιήθηκαν 4 nodes - με ένα από αυτά να αποτελεί ταυτόχρονα χρέη master και worker node - όπου έστελναν στο σύστημα επαναλαμβανόμενα τα positions του κάθε node μαζί με τα ranges και που αναφέρθηκαν προηγουμένως, με τον τρόπο που περιγράφηκε αναλυτικά στο προηγούμενο κεφάλαιο. Αφού το master λάβει τα πακέτα, πραγματοποιεί την επεξεργασία για την εκτίμηση της θέσης, με βάση τον αλγόριθμο του Multilateration, και χρησιμοποιεί geometry_msgs/Pose - αποτελεί μία από τις βασικές μορφές μηνυμάτων για προσδιορισμό της θέσης στο ROS - ώστε να γνωστοποιήσει πίσω στο δίκτυο το position της μπάλας το οποίο και εκτιμήσαμε.

Σύμφωνα με τα παραπάνω, στην Εικόνα 5.20 φαίνονται τα περιεχόμενα των topics κατά την διαδικασία του πειράματος, ενώ στην Εικόνα 5.19 μπορούμε να δούμε συνδυαστικά τις γραφικές, στις οποίες παρουσιάζονται οι θέσεις που βρισκόταν το αντικείμενο, καθώς και οι εκτιμήσεις της θέσης που τελικά προέκυψαν.

Από τις εκτιμήσεις και τις ακριβής θέσεις, μπορούμε τελικά να υπολογίσουμε το συνολικό σφάλμα του συστήματος. Για το οποίο καταλήγουμε να είναι τα 0,1159m.



ΕΙΚΟΝΑ 5.19: Εκτίμηση θέσης του αντικειμένου. Με O είναι οι θέσεις στις οποίες τοποθετήθηκε το αντικείμενο ενώ με O είναι οι θέσεις που έγινε εκτίμηση ότι βρίσκεται



EIKONA 5.20: Μέρος των εν λειτουργία nodes/topics, καθώς και απεικόνιση των πληροφοριών που αποστέλλουν στα topics τα 4 nodes, μαζί με την εκτίμηση της θέσης από τον master.

5.3 Επαλήθευση προσδιορισμού ID

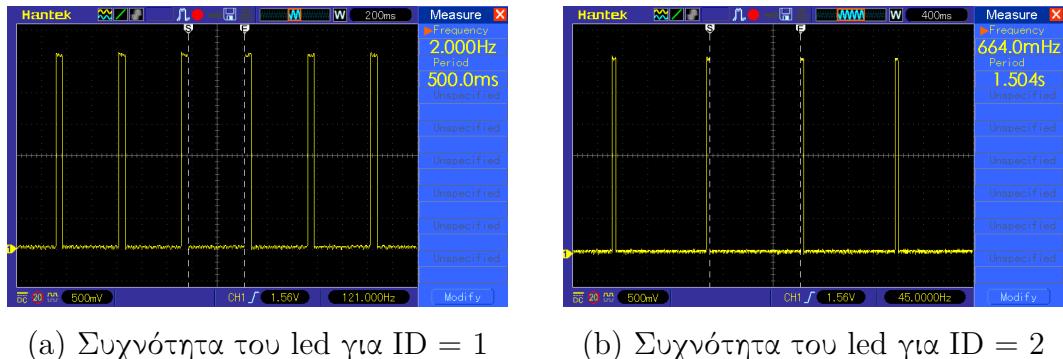
Όπως αναφέρθηκε στη Παράγραφο 4.8 χρησιμοποιούμε την συχνότητα με την οποία αναβοσβήνει το led για τον καθορισμό του ID⁶. Σημαντικός παράγοντας για την επιλογή της συχνότητας είναι το frame-rate της κάμερας με την οποία δειγματοληπτούμε.

Από το Nyquist–Shannon sampling theorem γνωρίζουμε ότι πρέπει να ισχύει $f_s > 2f_{max}$ μεταξύ της συχνότητας δειγματοληψίας και της μέγιστης συχνότητας σε ένα σήμα. Στην πράξη, είναι συχνό φαινόμενο να επιλέγουμε να διαφέρουν ακόμα και μία τάξη μεγέθους αυτά μεταξύ τους στην παραπάνω ανίσωση. Λόγω των 30fps της κάμερας καταλήγουμε να έχουμε ανά περίπου 33.33ms νέο καρέ, συνεπώς επιλέχθηκε ο χρόνος για τον οποίο θα είναι ενεργοποιημένο το led να είναι τα 70ms ώστε να καταγράφεται από τουλάχιστον 2 καρέ η ενεργοποίηση του.

Για τις περιόδους, ανάλογα με το ID, επιλέχθηκαν οι χρόνοι 500ms για ID = 1, 1500ms για ID = 2, 2500ms για ID = 3, κλπ., ώστε να υπάρχει ένα περιθώριο 1000ms για πιθανά σφάλματα μεταξύ των μετρήσεων.

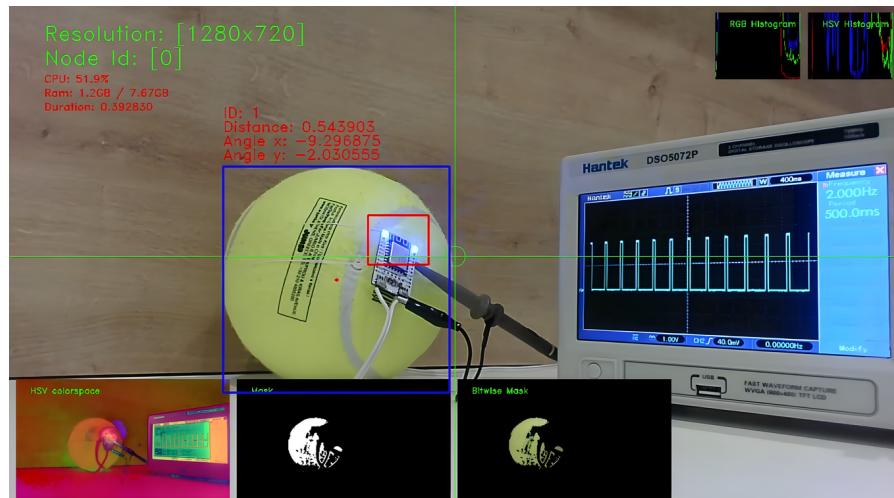
Η Εικόνα 5.21 παρουσιάζει τους παλμούς στους οποίους ενεργοποιείται το led, και κρατάμε αποδεκτό το ID = 1 για durations μεταξύ των παλμών durations ≤ 1000 ms, όμοια για ID = 2 έχουμε 1000ms $<$ duration ≤ 2000 ms, κλπ.

⁶Βίντεο από την διαδικασία του πειράματος μπορεί να βρεθεί στο [143]



ΕΙΚΟΝΑ 5.21: Παράδειγμα δύο εκ των τεσσάρων συχνοτήτων για την λειτουργία του led που επιλέχθηκαν.

Επίσης, η *Εικόνα 5.22* παρουσιάζει στιγμιότυπο κατά την διαδικασία επαλήθευσης της λειτουργίας, στο οποίο παρουσιάζεται τόσο το ID που εκτιμήθηκε όσο και το duration μεταξύ την παλμών.



ΕΙΚΟΝΑ 5.22: Στιγμιότυπο της διαδικασίας πείραματος, κατά την διάρκεια καθορισμού του ID του αντικειμένου με βάση την συχνότητα που αναβοσβήνει το led

Από τα παραπάνω πειράματα, δείχνεται ότι είναι εφικτός ο προσδιορισμός της θέσης της μπάλας από πολλαπλές λήψεις που θα παρέχει ένα σημήνος. Παρόλα αυτά, λόγω των περιορισμών που αναφέρθηκαν, σε αυτήν την μορφή που βρίσκεται ακόμα το σύστημα, συμπεραίνουμε ότι πραγματοποιεί τις βέλτιστες προσεγγίσεις της θέσης, όταν το αντικείμενο ανιχνεύεται περίπου στο μέσο του image plane σε αποστάσεις κάτω των 5m. Ενώ, σε σχέση με τις επεξεργαστικές ανάγκες, το Raspberry Pi

μπορεί να διαχειριστεί αυτού του είδους την επεξεργασία αλλά ίσως δυσκολευτεί αν γίνει προσπάθεια να προσθέσουμε σε αυτό επιπλέον λογική.

Ακόμα, ενώ μπορεί να προσδιοριστεί μοναδικά το ID του αντικειμένου που εντοπίζεται, είναι φανερό ότι ο χρόνος που χρειάζεται ώστε αξιόπιστα να καθοριστεί αυτό^{*} δεν είναι πεπερασμένος και μπορεί να χρειαστεί μερικά δευτερόλεπτα. Ο οποίος εξαρτάται από το πλήθος των μοναδικών ID που θέλουμε να ανιχνεύσουμε, το duration που έχει προεπιλεχθεί για κάθε ID, καθώς και τον αριθμό των επαναλήψεων που επιθυμούμε να επαναλάβουμε πριν τον τελικό προσδιορίσουμε του, ώστε αξιόπιστα να έχουμε μία εκτίμηση.

Κεφάλαιο 6

Τελικά Συμπεράσματα και Μελλοντικές Εξελίξεις

”Μία επαρκής έρευνα τείνει να είναι αρκετή να υποστηρίζει τα συμπεράσματα τα οποία εξάγουμε”

Arthur Bloch

Έπειτα από την ολοκλήρωση των πειραματικών διαδικασιών, είμαστε πλέον σε θέση για καταλήξουμε σε συγκεκριμένα συμπεράσματα για την παρούσα διπλωματική, καθώς και τα σημεία τα οποία μελλοντικά μπορούν να βελτιωθούν ή να εξελιχθούν.

6.1 Συμπεράσματα

Αρχικά, ο λόγος επιλογής της HSV μεθόδου στην συγκεκριμένη εργασία - για την υλοποίηση του object detection - ήταν διότι μπορεί να παρέχει ικανοποιητική απόδοση όντως πρώτη γενιά, με μικρές σχετικά ανάγκες επεξεργασίας. Παρόλα αυτά, κύριο αρνητικό της είναι, η ανάγκη πριν την χρήση του συστήματος να πραγματοποιηθεί calibration με βάση το scene λειτουργίας.

Σχετικά με τα χαρακτηριστικά του Hough Transform. Αυτός είχε αποδεκτά αποτελέσματα σε σκηνές με μικρό πλήθος από edges στο background. Ενώ, σε περιπτώσεις με αρκετά noisy υπόβαθρα θα πρέπει να αποφεύγεται η επιλογή του. Καθώς, ακόμα και στις δοκιμές που έγιναν σε 16-core Workstation με AMD Ryzen 7 2700X ως επεξεργαστή, παρουσιάστηκαν σημαντικά frame drops ώστε να κάνουν την ανίχνευση του αντικειμένου να μην μπορεί να γίνει σε πραγματικό χρόνο¹.

¹Η συγκεκριμένη απόδοση εξαρτάται σε σημαντικό βαθμό από την επιλογή των παραμέτρων που χρησιμοποιούμε για το Hough Transform, οι οποίες καθορίζουν και πόσο σωστά γίνεται το detection

Το σύστημα δοκιμάστηκε τόσο σε indoor και outdoor περιβάλλοντα, κάτω από διαφορετικές συνθήκες φωτισμού, και ο πιο ακριβής εντοπισμός του αντικειμένου στο image plane που προέκυψε - ήταν σε κατάσταση ημέρας. Όμως, παρατηρήθηκε ότι ακόμα και σε συνθήκες χαμηλού φωτισμού ήταν δυνατή η ανίχνευση του αντικειμένου, αρκεί η πηγή φωτός να μην φωτοβολούσε υπό συγκεκριμένη κατεύθυνση, αλλά να ήταν πιο ομαλά diffused στο χώρο.

Το πιο σημαντικό συμπέρασμα από το σύνολο της εργασίας, είναι το proof of concept το οποίο μας παρέχει για την δυνατότητα εντοπισμού αντικειμένων σε real time χρόνο από low-computation low-cost Embedded Linux συστήματα. Αυτός ο εντοπισμός, έγινε με γνώμονα να αφορά drone του συμήνους για relative positioning - τα οποία φέρουν στο πλαίσιο τους παρόμοιες μορφολογίες όπως αυτό της μπάλας και για τα οποία δεν έχουμε την δυνατότητα με άλλο τρόπο να εντοπίσουμε την θέση τους (βλ. Εικόνα 6.1). Παρόλα αυτά, πολύ εύκολα μπορεί να χρησιμοποιηθεί ακόμα και για τον εντοπισμό εχθρικών UAVs, για τα οποία εξ' ορισμού λόγω της ετερογένειας τους δεν γνωρίζουμε την θέση τους, ακολουθώντας παρόμοια λογική ότι φέρουν ένα χαρακτηριστικό στο body τους. Ενώ, μπορεί ακόμα και να χρησιμοποιηθεί για τον προσδιορισμό της θέσης άλλων επίγειων ή εναέριων αντικειμένων από το swarm, όπως για παράδειγμα σε αποστολές S&R.



(a) Μορφολογία μπάλας (URL)



(b) Ημισφαίριο στο πλαίσιο του (URL)

ΕΙΚΟΝΑ 6.1: Προτεινόμενα drone για τα οποία μπορούν να εκτιμηθούν οι θέσεις

6.2 Μελλοντικές Εξελίξεις

Μέσω των παραπάνω συμπερασμάτων, μπορούν να προταθούν και πιθανές εξελίξεις οι οποίες θα βοηθήσουν τελικά, στην πραγματοποίηση ενός συστήματος, το οποίο θα είναι σε θέση, ανεξάρτητα από το περιβάλλον και την δυναμικότητα του, να πραγματοποιήσει εντοπισμό αντικειμένων στον χώρο.

- Για να ξεπεραστεί ο περιορισμός της πραγματοποίησης calibration πριν την λειτουργία του συστήματος - ακολουθώντας την ίδια προσέγγιση υλοποίησης - μελλοντικά σημαντικό είναι η αντικατάσταση της κάμερας που χρησιμοποιήθηκε με μία Infrared (IR).
- Επίσης, ενδιαφέρουσα μελλοντική επέκταση είναι να απομακρυνθούμε από μία αυτή και άλλη αυτή computer vision προσέγγιση εντοπισμού του αντικειμένου, και να χρησιμοποιηθεί κάποια machine learning εναλλακτική. Ως παράδειγμα, χρησιμοποιώντας Tiny-YOLO για το object detection. Με αυτόν τον τρόπο θα είναι δυνατός ο εντοπισμός μεγαλύτερης γκάμας αντικειμένων από το σύστημα.
- Σημαντικό είναι να γίνουν μελλοντικά και συγχρίσεις του προσδιορισμού της θέσης με εναλλακτικές μεθόδους, όπως χρησιμοποιώντας Triangulation αντί για Multilateration και να αξιοποιηθεί πληροφορία σχετικά με την γωνία για το localization.
- Όταν το σύστημα φτάσει σε σημείο να υλοποιηθεί σε πραγματικά drones, είναι πολύ πιθανόν η κάμερα για λόγους σχετικά με τους κραδασμούς, να βρίσκεται πάνω σε Gimbal για σταθεροποίηση της εικόνας. Μία άμεση επέκταση, είναι η εκμετάλλευση του Gimbal, για λήψη μετρήσεων υπό τις βέλτιστες συνθήκες του συστήματος - απλά κινώντας τη κάμερα ώστε το αντικείμενο παρά τα motion που πραγματοποιεί να βρίσκεται στο μέσο της εικόνας.
- Η ύπαρξη ενός μηχανισμού smoothing για τις μετρήσεις απόστασης και γωνίας - ώστε να αποφεύγονται τα spikes - είναι εξίσου κάτι που πρόκειται να βελτιώσει την απόδοση συνολικά του συστήματος. Καθώς έτσι, ακόμα και τις χρονικές στιγμές τις οποίες δεν ανιχνεύεται σωστά ή λόγω θορύβου περιλαμβάνει μεγαλύτερα σφάλματα από τα επιτρεπτά, ο προσδιορισμός της θέσης θα συνεχίζει να γίνεται σε ρεαλιστικά πλαίσια. Σε αυτό μπορεί να βοηθήσουν προσεγγίσεις με βάση το optical flow των αντικειμένων.
- Για να έχει πραγματική χρησιμότητα το σύστημα, ένα ακόμα ζήτημα αφορά τα ranges μέχρι τα οποία αξιόπιστα θα μπορούμε να υπολογίσουμε το αντικείμενο. Είναι σημαντικό λοιπόν να δοκιμαστούν διαφορετικές τεχνικές - πέρα από το

structure from reference - για την εκτίμηση της απόσταση, με μία από αυτές ως προτεινόμενη να είναι το stereo vision.

- Σχοπός είναι να δημιουργηθεί ένα distributed αυτόνομο σύστημα το οποίο δεν θα παρουσιάζει κάποιο single point of failure σε μία πιθανή αποστολή, την οποία θα πρέπει να διεκπεραιώσει με επιτυχία. Για αυτόν τον λόγο πρέπει να βρεθεί τρόπος να ξεπεραστεί ο περιορισμός που υπάρχει μέχρι αυτήν την στιγμή και προέρχεται από την αρχιτεκτονική του ROS, ότι το master node είναι προκαθορισμένο πριν από την έναρξη λειτουργίας του συστήματος. Ενώ μελλοντικά, σε περίπτωση που το node που αναλαμβάνει αυτήν την λειτουργία του προκληθεί κάποιο malfunction θα πρέπει να μπορεί on the fly να αντικατασταθεί και να αναλάβει κάποιο άλλο drone αυτήν την αρμοδιότητα.
- Πρέπει επιπλέον να γίνουν δοκιμές με πραγματικά δεδομένα για τις θέσεις των nodes, τα οποία με μεγάλη ακρίβεια θα προσδιορίζουν την θέση τους. Αυτό μπορεί να γίνει είτε με χρήση RTK GPS είτε με οικονομικότερες εναλλακτικές αν συνεχίζει να μας ενδιαφέρει το relative positioning. Μία εξ' αυτών είναι η χρήση των UWB για το σχετικό localization πρώτα των drones μεταξύ τους - με τρόπους που περιγράφτηκαν στο κεφάλαιο με το θεωρητικό υπόβαθρο - και σε δεύτερο επίπεδο, γνωρίζοντας τις σχετικές θέσεις τελικά να πραγματοποιούμε τον εντοπισμό ετερογενούς object.
- Σχετικά με τον καθορισμό του ID, μπορούν να επιλεχθούν τεχνικές κωδικοποίησης για τον καθορισμό τους, ώστε σε πεπερασμένο χρονικό διάστημα να γνωρίζουμε σε ποιο node αναφερόμαστε. Δύο πιθανές προσεγγίσεις είναι με χρήση Huffman ή Golomb codes.
- Η πιο σημαντική μελλοντική επέκταση, είναι φυσικά οι δοκιμές του συστήματος σε πραγματικά drone, και πιθανόν να βρεθεί τρόπος να αντικατασταθεί η ανάγκη για επικοινωνία μέσω WiFi με μία που να μπορεί να καλύψει αξιόπιστα μεγαλύτερες αποστάσεις.

References

- [4] S. G. Gupta, D. Ghonge, P. M. Jawandhiya, *et al.*, “Review of unmanned aircraft system (uas),” *International Journal of Advanced Research in Computer Engineering & Technology (IJAR CET)* Volume, vol. 2, 2013. [Online]. Available: [http://refhub.elsevier.com/S0376-0421\(16\)30134-8/sbref2](http://refhub.elsevier.com/S0376-0421(16)30134-8/sbref2).
- [7] M. Hassanalian and A. Abdelkefi, “Classifications, applications, and design challenges of drones: A review,” *Progress in Aerospace Sciences*, vol. 91, pp. 99 –131, 2017, ISSN: 0376-0421. DOI: <https://doi.org/10.1016/j.paerosci.2017.04.003>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0376042116301348>.
- [8] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, “Swarms of unmanned aerial vehicles — a survey,” *Journal of Industrial Information Integration*, vol. 16, p. 100106, 2019, ISSN: 2452-414X. DOI: <https://doi.org/10.1016/j.jii.2019.100106>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2452414X18300086>.
- [26] S. Milani and A. Memo, “Impact of drone swarm formations in 3d scene reconstruction,” pp. 2598–2602, 2016. DOI: <10.1109/ICIP.2016.7532829>.
- [40] A. Bürkle, F. Segor, and M. Kollmann, “Towards autonomous micro uav swarms,” *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 339–353, 2011, ISSN: 1573-0409. DOI: <10.1007/s10846-010-9492-x>. [Online]. Available: <https://doi.org/10.1007/s10846-010-9492-x>.
- [41] E. Mitka and S. Mouroutsos, “Classification of drones,” *American Journal of Engineering research*, vol. 6, pp. 36–41, Jul. 2017.
- [45] S. Goel, “A distributed cooperative uav swarm localization system: Development and analysis,” Sep. 2017. DOI: <10.33012/2017.15217>.
- [46] J. Li, Y. Bi, K. Li, K. Wang, F. Lin, and B. M. Chen, *Accurate 3d localization for mav swarms by uwb and imu fusion*, 2018. arXiv: [1807.10913\[cs.RO\]](1807.10913[cs.RO]).
- [47] I. Milani, C. Bongioanni, F. Colone, and P. Lombardo, “Fusing active and passive measurements for drone localization,” in *2020 21st International*

- Radar Symposium (IRS)*, 2020, pp. 245–249. DOI: [10.23919/IRS48640.2020.9253794](https://doi.org/10.23919/IRS48640.2020.9253794).
- [48] M. Malajner, P. Planinšič, and D. Gleich, “Uwb ranging accuracy,” in *2015 International Conference on Systems, Signals and Image Processing (IWS-SIP)*, 2015, pp. 61–64. DOI: [10.1109/IWSSIP.2015.7314177](https://doi.org/10.1109/IWSSIP.2015.7314177).
- [49] Y. Qi, Y. Zhong, and Z. Shi, “Cooperative 3-d relative localization for UAV swarm by fusing UWB with IMU and GPS,” *Journal of Physics: Conference Series*, vol. 1642, p. 012028, Sep. 2020. DOI: [10.1088/1742-6596/1642/1/012028](https://doi.org/10.1088/1742-6596/1642/1/012028). [Online]. Available: <https://doi.org/10.1088/1742-6596/1642/1/012028>.
- [50] I. A. Meer, M. Ozger, and C. Cavdar, “On the localization of unmanned aerial vehicles with cellular networks,” in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6. DOI: [10.1109/WCNC45663.2020.9120588](https://doi.org/10.1109/WCNC45663.2020.9120588).
- [51] V. Delafontaine, F. Schiano, G. Cocco, A. Rusu, and D. Floreano, “Drone-aided localization in lora iot networks,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 286–292. DOI: [10.1109/ICRA40945.2020.9196869](https://doi.org/10.1109/ICRA40945.2020.9196869).
- [53] S. K. Moore, “Superaccurate gps chips coming to smartphones in 2018,” *IEEE Spectrum*, Sep. 2017. [Online]. Available: <https://spectrum.ieee.org/tech-talk/semiconductors/design/superaccurate-gps-chips-coming-to-smartphones-in-2018>.
- [55] I. Um, S. Park, H. T. Kim, and H. Kim, “Configuring rtk-gps architecture for system redundancy in multi-drone operations,” *IEEE Access*, vol. 8, pp. 76 228–76 242, 2020. DOI: [10.1109/ACCESS.2020.2989276](https://doi.org/10.1109/ACCESS.2020.2989276).
- [56] X. Chang, C. Yang, J. Wu, X. Shi, and Z. Shi, “A surveillance system for drone localization and tracking using acoustic arrays,” in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2018, pp. 573–577. DOI: [10.1109/SAM.2018.8448409](https://doi.org/10.1109/SAM.2018.8448409).
- [57] C. Ruiz, X. Chen, L. Zhang, and P. Zhang, “Collaborative localization and navigation in heterogeneous uav swarms: Demo abstract,” Nov. 2016, pp. 324–325. DOI: [10.1145/2994551.2996544](https://doi.org/10.1145/2994551.2996544).
- [60] G. Hemann, S. Singh, and M. Kaess, “Long-range gps-denied aerial inertial navigation with lidar localization,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1659–1666. DOI: [10.1109/IROS.2016.7759267](https://doi.org/10.1109/IROS.2016.7759267).
- [62] S. Krul, C. Pantos, M. Frangulea, and J. Valente, “Visual slam for indoor livestock and farming using a small drone with a monocular camera: A

- feasibility study," *Drones*, vol. 5, no. 2, 2021, ISSN: 2504-446X. DOI: [10.3390/drones5020041](https://doi.org/10.3390/drones5020041). [Online]. Available: <https://www.mdpi.com/2504-446X/5/2/41>.
- [63] M. Guillén, S. García, R. Barea, L. Bergasa, E. Molinos, R. Arroyo, E. Romera, and S. Pardo, "A multi-sensorial simultaneous localization and mapping (slam) system for low-cost micro aerial vehicles in gps-denied environments," *Sensors (Switzerland)*, vol. 17, Apr. 2017. DOI: [10.3390/s17040802](https://doi.org/10.3390/s17040802).
- [67] A. Mashhood, M. Mohammed, M. Abdulwahab, S. Abdulwahab, and H. Noura, "A hardware setup for formation flight of uavs using motion tracking system," in *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, 2015, pp. 1–6. DOI: [10.1109/ISMA.2015.7373474](https://doi.org/10.1109/ISMA.2015.7373474).
- [69] M. F. i Azam and M. N. Ayyaz, *Location and position estimation in wireless sensor networks*, 2016. arXiv: [1611.03420 \[cs.NI\]](https://arxiv.org/abs/1611.03420).
- [70] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007. DOI: [10.1109/MWC.2007.4407221](https://doi.org/10.1109/MWC.2007.4407221).
- [71] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529 –2553, 2007, ISSN: 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2006.11.018>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128606003227>.
- [72] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1, 2006, pp. 374–382. DOI: [10.1109/SAHCN.2006.288442](https://doi.org/10.1109/SAHCN.2006.288442).
- [76] B. Yamamoto, A. Wong, P. J. Agcanas, K. Jones, D. Gaspar, R. Andrade, and A. Z. Trimble, "Received signal strength indication (rss) of 2.4 ghz and 5 ghz wireless local area network systems projected over land and sea for near-shore maritime robot operations," *Journal of Marine Science and Engineering*, vol. 7, no. 9, p. 290, Aug. 2019, ISSN: 2077-1312. DOI: [10.3390/jmse7090290](https://doi.org/10.3390/jmse7090290). [Online]. Available: <http://dx.doi.org/10.3390/jmse7090290>.
- [78] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rss value for distance estimation in wireless sensor networks based on zigbee," pp. 303–306, 2008. DOI: [10.1109/IWSSIP.2008.4604427](https://doi.org/10.1109/IWSSIP.2008.4604427).

- [80] M. Botta and M. Simek, “Adaptive distance estimation based on rssi in 802.15.4 network,” *Radioengineering*, vol. 22, pp. 1162–1168, Dec. 2013.
- [84] J. Bachrach and C. Taylor, *1 localization in sensor networks*, 2004. [Online]. Available: <https://www.semanticscholar.org/paper/1-Localization-in-Sensor-Networks-Bachrach-Taylor/a70a5ac4c092fec8c140fc3850e2e1f6d7b698cf>.
- [96] A. M. H. KHALEL, “Position location techniques in wireless communication systems,” 2010. [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:bth-4796>.
- [98] R. Kaune, C. Steffes, S. Rau, W. Konle, and J. Pagel, “Wide area multilateration using ads-b transponder signals,” in *2012 15th International Conference on Information Fusion*, 2012, pp. 727–734.
- [101] B. O’Keefe, “Finding location with time of arrival and time difference of arrival techniques,” *ECE Senior Capstone Project*, 2017. [Online]. Available: https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/FireBrick_OKeefe_F1.pdf.
- [102] X. Li, Z. D. Deng, L. T. Rauchenstein, and T. J. Carlson, “Contributed review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements,” *Review of Scientific Instruments*, vol. 87, no. 4, p. 041502, 2016. DOI: [10.1063/1.4947001](https://doi.org/10.1063/1.4947001). eprint: <https://doi.org/10.1063/1.4947001>. [Online]. Available: <https://doi.org/10.1063/1.4947001>.
- [103] S. P. Singh and S. Sharma, “Range free localization techniques in wireless sensor networks: A review,” *Procedia Computer Science*, vol. 57, pp. 7 –16, 2015, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2015.07.357>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050915018864>.
- [104] C. Zhou, T. Xu, and H. Dong, “Distributed locating algorithm mds-map (lf) based on low-frequency signal,” *Computer Science and Information Systems*, vol. 12, pp. 55–55, Nov. 2015. DOI: [10.2298/CSIS140801055Z](https://doi.org/10.2298/CSIS140801055Z).
- [105] X. Wang and N. Yunfeng, “An improved distance vector-hop localization algorithm based on coordinate correction,” *International Journal of Distributed Sensor Networks*, vol. 13, p. 155014771774183, Nov. 2017. DOI: [10.1177/1550147717741836](https://doi.org/10.1177/1550147717741836).

-
- [106] X. Yang and W. Zhang, “An improved dv-hop localization algorithm based on hop distance and hops correction,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, pp. 319–328, Jun. 2016. DOI: [10.14257/ijmue.2016.11.6.28](https://doi.org/10.14257/ijmue.2016.11.6.28).
 - [107] L. Yin, “A new distance vector-hop localization algorithm based on half-measure weighted centroid,” *Mobile Information Systems*, vol. 2019, pp. 1–9, Jan. 2019. DOI: [10.1155/2019/9892512](https://doi.org/10.1155/2019/9892512).
 - [108] E. Mair, K. Strobl, M. Suppa, and D. Burschka, “Efficient camera-based pose estimation for real-time applications,” Nov. 2009. DOI: [10.1109/IROS.2009.5354584](https://doi.org/10.1109/IROS.2009.5354584).
 - [114] X. Zhou, S. Liu, G. Pavlakos, V. Kumar, and K. Daniilidis, “Human motion capture using a drone,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2027–2033. DOI: [10.1109/ICRA.2018.8462830](https://doi.org/10.1109/ICRA.2018.8462830).
 - [115] M. A. Ma’sum, G. Jati, M. K. Arrofi, A. Wibowo, P. Mursanto, and W. Jatmiko, “Autonomous quadcopter swarm robots for object localization and tracking,” in *MHS2013*, 2013, pp. 1–6. DOI: [10.1109/MHS.2013.6710447](https://doi.org/10.1109/MHS.2013.6710447).
 - [140] A. Norrdine, *An algebraic solution to the multilateration problem*, Apr. 2015. DOI: [10.13140/RG.2.1.1681.3602](https://doi.org/10.13140/RG.2.1.1681.3602).

External Links

- [1] M. Satell, *Ultimate list of drone stats for 2020*, Jul. 2020. [Online]. Available: <https://www.phillybyair.com/blog/drone-stats/> (visited on 11/2020).
- [2] B. I. Intelligence, *Drone market outlook: Industry growth trends, market stats and forecast*, Mar. 2020. [Online]. Available: <https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts> (visited on 11/2020).
- [3] C. de León, *Drone delivery? amazon moves closer with f.a.a. approval*, Aug. 2020. [Online]. Available: <https://www.nytimes.com/2020/08/31/business/amazon-drone-delivery.html> (visited on 11/2020).
- [5] *Unmanned aerial vehicle*. [Online]. Available: https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle (visited on 11/2020).
- [6] *List of unmanned aerial vehicles*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_unmanned_aerial_vehicles (visited on 11/2020).
- [9] *A guide to fixed wing drones*. [Online]. Available: <https://www.coptrz.com/a-guide-to-fixed-wing-drones/> (visited on 11/2020).
- [10] *Aircraft principal axes*. [Online]. Available: https://en.wikipedia.org/wiki/Aircraft_principal_axes (visited on 11/2020).
- [11] *Microelectromechanical systems*. [Online]. Available: https://en.wikipedia.org/wiki/Microelectromechanical_systems (visited on 01/2021).
- [12] *Microcontroller*. [Online]. Available: <https://en.wikipedia.org/wiki/Microcontroller> (visited on 11/2020).
- [13] *Microprocessor*. [Online]. Available: <https://en.wikipedia.org/wiki/Microprocessor> (visited on 11/2020).
- [14] *Electronic speed control*. [Online]. Available: https://en.wikipedia.org/wiki/Electronic_speed_control (visited on 11/2020).
- [15] *Inertial measurement unit*. [Online]. Available: https://en.wikipedia.org/wiki/Inertial_measurement_unit (visited on 11/2020).
- [16] *Tuc researcher's night 2020: Drones, satellites, and more ... Nov. 2020*. [Online]. Available: <https://www.youtube.com/watch?v=9ZZW-XQKG1w&>

- [feature=youtu.be&fbclid=IwAR2VdxWuFOwiCbEdEGJcBDxapixuPtExsViDCoBEPQLnU8YQLg3hGRBn47w](https://www.youtube.com/watch?v=IwAR2VdxWuFOwiCbEdEGJcBDxapixuPtExsViDCoBEPQLnU8YQLg3hGRBn47w) (visited on 11/2020).
- [17] *Senselab research*. [Online]. Available: <http://senselab.tuc.gr/> (visited on 11/2020).
- [18] *Optimized flocking of autonomous drones in confined environments*, Jul. 2018. [Online]. Available: <https://www.youtube.com/watch?v=E4XpyG4eMKE> (visited on 11/2020).
- [19] *An autonomous swarm of flying robots*, Feb. 2014. [Online]. Available: https://www.youtube.com/watch?v=W7yJx_IY9Mg (visited on 11/2020).
- [20] *Drone swarm performance and applications*, Jun. 2019. [Online]. Available: <https://www.embention.com/news/drone-swarm-performance-and-applications/> (visited on 11/2020).
- [21] M. Murison, *The potential of drone swarms*, Apr. 2019. [Online]. Available: <https://blog.dronebase.com/the-potential-of-drone-swarms> (visited on 11/2020).
- [22] *Swarm robotics*. [Online]. Available: https://en.wikipedia.org/wiki/Swarm_robotics (visited on 11/2020).
- [23] *Controlling robotic swarms*, Dec. 2014. [Online]. Available: <https://www.youtube.com/watch?v=stzQNjtDg0g> (visited on 11/2020).
- [24] *Kargu - the kamikaze drones getting ready for the swarm operation*, Jul. 2019. [Online]. Available: <https://www.youtube.com/watch?v=3d28APIfwSI> (visited on 11/2020).
- [25] *A swarm of nano quadrotors*, Pennsylvania, Jan. 2012. [Online]. Available: <https://www.youtube.com/watch?v=YQIMGV5vtd4> (visited on 11/2020).
- [27] *How intel made its world record-breaking drone show*, Jul. 2018. [Online]. Available: https://www.youtube.com/watch?v=zdLjoqa_oUs (visited on 11/2020).
- [28] *Ford motor company / firefly drone shows*, Aug. 2019. [Online]. Available: <https://www.youtube.com/watch?v=Bdsr1cojSxo> (visited on 11/2020).
- [29] *A swarm of mini drones makes ... magic! / marco tempest*. [Online]. Available: <https://www.youtube.com/watch?v=u2bQSKvZ2qI> (visited on 12/2020).
- [30] *Hollywood welcomes drones to the set*, Jan. 2015. [Online]. Available: <https://www.youtube.com/watch?v=GXhlgNHK14s> (visited on 12/2020).
- [31] *3d mapping with a dji phantom & drone deploy*, Aug. 2016. [Online]. Available: <https://www.youtube.com/watch?v=SATijfXnshg> (visited on 12/2020).

- [32] *Dji – m300 rtk - coordinating police missions from the sky*, May 2020. [Online]. Available: <https://www.youtube.com/watch?v=FpkEsmU1NEo> (visited on 12/2020).
- [33] *Drones for disaster management (nepal earthquake)*, Aug. 2015. [Online]. Available: <https://www.youtube.com/watch?v=xYrhAoX2H8I> (visited on 12/2020).
- [34] *Drone search and rescue - wake forest fire department*, Jun. 2018. [Online]. Available: https://www.youtube.com/watch?v=Ih_ayi318X0 (visited on 12/2020).
- [35] *Amazon gets faa approval to operate a fleet of delivery drones*, Aug. 2020. [Online]. Available: <https://www.youtube.com/watch?v=beZ8iKGuDeo> (visited on 12/2020).
- [36] *Unmanned combat aerial vehicle*. [Online]. Available: https://en.wikipedia.org/wiki/Unmanned_combat_aerial_vehicle (visited on 11/2020).
- [37] *Military robot*. [Online]. Available: https://en.wikipedia.org/wiki/Military_robot (visited on 11/2020).
- [38] *Isr unmanned aerial vehicles*. [Online]. Available: <https://www.militaryfactory.com/aircraft/uav-intelligence-surveillance-reconnaissance.asp> (visited on 12/2020).
- [39] *List of unmanned aerial vehicle applications*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_unmanned_aerial_vehicle_applications (visited on 11/2020).
- [42] *Drone swarms: The buzz of the future*, Nov. 2015. [Online]. Available: <https://www.youtube.com/watch?v=0so5Fc6B7W4> (visited on 11/2020).
- [43] *Positioning system*. [Online]. Available: https://en.wikipedia.org/wiki/Positioning_system (visited on 01/2021).
- [44] *Sensor fusion*. [Online]. Available: https://en.wikipedia.org/wiki/Sensor_fusion (visited on 01/2021).
- [52] *Gps accuracy*. [Online]. Available: <https://www.gps.gov/systems/gps/performance/accuracy/> (visited on 05/2020).
- [54] *Real-time kinematic positioning*. [Online]. Available: https://en.wikipedia.org/wiki/Real-time_kinematic_positioning (visited on 05/2020).
- [58] *About opencv*. [Online]. Available: <https://opencv.org/about/> (visited on 05/2020).
- [59] L. A. Wasser, *The basics of lidar - light detection and ranging - remote sensing*. [Online]. Available: <https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics> (visited on 07/2020).

- [61] *Understanding slam using pose graph optimization / / autonomous navigation, part 3.* [Online]. Available: <https://www.youtube.com/watch?v=savZtgPyyJQ> (visited on 07/2020).
- [64] *Motion capture.* [Online]. Available: https://en.wikipedia.org/wiki/Motion_capture (visited on 05/2020).
- [65] *Vicon.* [Online]. Available: <https://www.vicon.com/about-us/what-is-motion-capture/> (visited on 05/2020).
- [66] *Optitrack.* [Online]. Available: <https://optitrack.com/> (visited on 05/2020).
- [68] *Wireless sensor network.* [Online]. Available: https://en.wikipedia.org/wiki/Wireless_sensor_network (visited on 12/2020).
- [73] *Research on localization algorithms and technologies,* Nov. 2015. [Online]. Available: <https://www.youtube.com/watch?v=XDFVtYEh0tM> (visited on 12/2020).
- [74] *Path loss.* [Online]. Available: https://en.wikipedia.org/wiki/Path_loss (visited on 12/2020).
- [75] *Free-space path loss.* [Online]. Available: https://en.wikipedia.org/wiki/Free-space_path_loss (visited on 12/2020).
- [77] *Friis transmission equation.* [Online]. Available: https://en.wikipedia.org/wiki/Friis_transmission_equation (visited on 12/2020).
- [79] *Dbm.* [Online]. Available: <https://en.wikipedia.org/wiki/DBm> (visited on 12/2020).
- [81] *Kinematics.* [Online]. Available: <https://en.wikipedia.org/wiki/Kinematics> (visited on 12/2020).
- [82] *An introduction to direction findin,* Feb. 2020. [Online]. Available: <https://www.youtube.com/watch?v=N8rZIAHxAH4> (visited on 12/2020).
- [83] *Speed of sound.* [Online]. Available: https://en.wikipedia.org/wiki/Speed_of_sound (visited on 12/2020).
- [85] *Beamforming.* [Online]. Available: <https://en.wikipedia.org/wiki/Beamforming> (visited on 12/2020).
- [86] *Directional antenna.* [Online]. Available: https://en.wikipedia.org/wiki/Directional_antenna (visited on 12/2020).
- [87] *Phase difference,* Dec. 2015. [Online]. Available: <https://www.miniphysics.com/phase-difference.html> (visited on 12/2020).
- [88] *Fmcw radars lec 5: Angle estimation,* May 2018. [Online]. Available: <https://www.youtube.com/watch?v=0k2HarUxS8o> (visited on 12/2020).
- [89] *How gps works? trilateration explained,* Mar. 2014. [Online]. Available: <https://www.youtube.com/watch?v=403ZVHFhes> (visited on 12/2020).

- [90] *Trilateration vs triangulation – how gps receivers work*, Dec. 2020. [Online]. Available: <https://gisgeography.com/trilateration-triangulation-gps/> (visited on 12/2020).
- [91] *Omnidirectional antenna*. [Online]. Available: https://en.wikipedia.org/wiki/Omnidirectional_antenna (visited on 12/2020).
- [92] *Isotropic radiator*. [Online]. Available: https://en.wikipedia.org/wiki/Isotropic_radiator (visited on 12/2020).
- [93] *Rssi-based accurate indoor localization scheme for wireless sensor networks*, Nov. 2015. [Online]. Available: <https://www.youtube.com/watch?v=CWvRJdF7oVE> (visited on 12/2020).
- [94] J. Wedding, *Find x location using 3 known (x,y) location using trilateration*, Jul. 2017. [Online]. Available: <https://math.stackexchange.com/questions/884807/find-x-location-using-3-known-x-y-location-using-trilateration> (visited on 12/2020).
- [95] *Hybrid localization algorithm for wireless sensor networks*, Oct. 2015. [Online]. Available: <https://www.youtube.com/watch?v=kf-hXqZHkAA> (visited on 12/2020).
- [97] *Multilateration*. [Online]. Available: <https://www.radartutorial.eu/02.basics/rp52.en.html> (visited on 01/2022).
- [99] I. Biswas, *Triangulation vs trilateration vs multilateration – for indoor positioning systems*, Jun. 2019. [Online]. Available: <https://www.pathpartnertech.com/triangulation-vs-trilateration-vs-multilateration-for-indoor-positioning-systems/> (visited on 12/2020).
- [100] *Multilateration*. [Online]. Available: <https://en.wikipedia.org/wiki/Multilateration> (visited on 12/2020).
- [109] *Calculate distance or size of an object in a photo image*. [Online]. Available: <https://www.scantips.com/lights/subjectdistance.html> (visited on 09/2021).
- [110] A. Rosebrock, *Find distance from camera to object/marker using python and opencv*, Jul. 2021. [Online]. Available: <https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/> (visited on 09/2021).
- [111] *How do i calculate the distance of an object in a photo?* [Online]. Available: <https://photo.stackexchange.com/questions/12434/how-do-i-calculate-the-distance-of-an-object-in-a-photo> (visited on 09/2021).
- [112] *Epipolar geometry*. [Online]. Available: https://en.wikipedia.org/wiki/Epipolar_geometry (visited on 09/2021).

- [113] *Introduction to computer vision*. [Online]. Available: <https://www.udacity.com/course/introduction-to-computer-vision--ud810> (visited on 09/2021).
- [116] *Ros_drone_swarm_mocap*. [Online]. Available: https://github.com/CSpyridakis/ros_drone_swarm_mocap (visited on 01/2022).
- [117] *About ros*. [Online]. Available: <https://www.ros.org/about-ros/> (visited on 05/2020).
- [118] *Documentation*. [Online]. Available: <http://wiki.ros.org/ROS/Tutorials> (visited on 05/2021).
- [119] *Ubuntu 20.04.2 for raspberry pi*. [Online]. Available: <https://ubuntu.com/download/raspberry-pi/thank-you?version=20.04.2&architecture=server-arm64+raspi> (visited on 08/2021).
- [120] *Opencv 4.2.0 documentation*. [Online]. Available: <https://docs.opencv.org/4.2.0/> (visited on 01/2022).
- [121] *Creative live! cam sync 1080p*. [Online]. Available: <https://en.creative.com/p/peripherals/creative-live-cam-sync-1080p> (visited on 08/2021).
- [122] *Bn-220 gps module + antenna datasheet*. [Online]. Available: <https://files.banggood.com/2016/11/BN-220%20GPS+Antenna%20datasheet.pdf> (visited on 08/2021).
- [123] *Universal asynchronous receiver-transmitter*. [Online]. Available: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter (visited on 09/2021).
- [124] *Nmea 0183*. [Online]. Available: https://en.wikipedia.org/wiki/NMEA_0183 (visited on 09/2021).
- [125] *Adafruit 10-dof imu breakout - l3gd20h + lsm303 + bmp180*. [Online]. Available: <https://www.adafruit.com/product/1604> (visited on 08/2021).
- [126] *I²c*. [Online]. Available: <https://en.wikipedia.org/wiki/I%C2%B2C> (visited on 09/2021).
- [127] *About kicad*. [Online]. Available: <https://www.kicad.org/about/kicad/> (visited on 09/2021).
- [128] *Raspberry-pi-fan-breakout*. [Online]. Available: <https://github.com/CSpyridakis/Raspberry-Pi-Fan-Breakout> (visited on 08/2021).
- [129] *How to install ubuntu desktop on raspberry pi 4*. [Online]. Available: <https://ubuntu.com/tutorials/how-to-install-ubuntu-desktop-on-raspberry-pi-4#1-overview> (visited on 09/2021).
- [130] *Rpi4 serial port not working on either raspberry os or ubuntu*. [Online]. Available: <https://raspberrypi.stackexchange.com/questions/114366/>

- rpi4-serial-port-not-working-on-either-raspberry-os-or-ubuntu
(visited on 09/2021).
- [131] *Nmea_nava sat_driver*. [Online]. Available: http://wiki.ros.org/nmea_nava sat_driver (visited on 09/2021).
- [132] *Ros-adafruit-10dof-imu*. [Online]. Available: <https://github.com/shanta nubhadoria/ros-adafruit-10dof-imu> (visited on 10/2021).
- [133] *Ros-adafruit-10dof-imu*. [Online]. Available: <https://github.com/CSpyri dakis/ros-adafruit-10dof-imu> (visited on 10/2021).
- [134] *Calibration manual*. [Online]. Available: https://reconcell.gitlab.io/reconcell_docs/UserManuals/VisualProcessing/CalibrationManual/calibration_manual.html (visited on 10/2021).
- [135] *Ros/opencv stereo calibration yaml files*. [Online]. Available: https://www.merrysprout.com/posts/camera/ros_stereo_calibration_yaml/ (visited on 10/2021).
- [136] *Cascade classifier*. [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html (visited on 01/2022).
- [137] *How to use opencv for ball detection*. [Online]. Available: <https://dbloisi.github.io/tutorial/balldetection.html> (visited on 01/2022).
- [138] *W-i-b*. [Online]. Available: <https://github.com/tassosblackg/W-I-B> (visited on 01/2022).
- [139] *Understanding focal length and field of view*. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/> (visited on 01/2022).
- [141] *Ball detection - distance/angle estimation - demo*. [Online]. Available: <https://www.youtube.com/watch?v=yE8dfYj0fwA> (visited on 01/2022).
- [142] *Motion capture system - outdoor experiments - phase 1*. [Online]. Available: <https://www.youtube.com/watch?v=twRdn7D5eus> (visited on 01/2022).
- [143] *Motion capture system - calculate node's id from blinking led frequency*. [Online]. Available: https://www.youtube.com/watch?v=-_UyhhGHF5s (visited on 01/2022).

Last Page