



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Βελτιστοποίηση της πόζας χρησιμοποιώντας ευθείες και σημεία φυγής σε σύστημα Visual SLAM μιας κάμερας

Μελέτη και υλοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΓΕΩΡΓΗ ΑΝΔΡΕΑ

Επιβλέπων: Πέτρος Μαραγκός
Καθ. ΕΜΠ

Αθήνα, Οκτώβριος 2022



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Βελτιστοποίηση της πόζας χρησιμοποιώντας ευθείες και σημεία φυγής σε σύστημα Visual SLAM μιας κάμερας

Μελέτη και υλοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΓΕΩΡΓΗ ΑΝΔΡΕΑ

Επιβλέπων: Πέτρος Μαραγκός
Καθ. ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 31 Οκτωβρίου 2022.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Πέτρος Μαραγκός
Καθ. ΕΜΠ

.....
Κων/νος Τζαφέστας
Αν. Καθ. ΕΜΠ

.....
Γεράσιμος Ποταμιάνος
Αν. Καθ. Παν/μιο Θεσσαλίας

Αθήνα, Οκτώβριος 2022



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

.....
Ανδρέας Γεωργής

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Copyright © - All rights reserved. Με την επιφύλαξη παντός δικαιώματος.
2022.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

31 Οκτωβρίου 2022

Περίληψη

Τις τελευταίες δεκαετίες ένα μεγάλο μέρος της επιστημονικής κοινότητας που ασχολούνται με τη ρομποτική έχει εστιάσει στην επίλυση του *Simultaneous Localisation And Mapping (SLAM)* προβλήματος, δηλαδή της ταυτόχρονης εκτίμησης της τροχιάς του ρομπο-τ/κάμερας και της 3D χαρτογράφησης του χώρου. Αυτό το είδος του προβλήματος, σε πολλά ρομπότ τα οποία φέρουν πολλαπλούς αισθητήρες και ο στόχος τους είναι, να εκτιμούν τη θέση τους στο χώρο και τη δυνατότητα αντίληψης του χώρου και ανακατασκευής του. Το πρόβλημα αυτό, αρχικά προσεγγίστηκε με στατιστικές μεθόδους και μεθόδους οπτικής οδο-μετρίας χωρίς όμως να δίνεται βάση στην ανακατασκευή και διατήρηση του χάρτη. Σήμερα, που η τεχνολογία αναπτύσσεται με γρήγορους ρυθμούς, η ερευνητική κοινότητα έχει στρέψει το ενδιαφέρον της σε λύσεις, που χρησιμοποιούν πολλαπλούς αισθητήρες για την αντίληψη του περιβάλλοντος. Επιπλέον, προτάσσονται λύσεις που εκτιμούν καλύτερα τη τροχιά του ρομπότ και είναι σε θέση, να διατηρήσουν και να ανακατασκευάσουν τον χάρτη και να παρέχουν λειτουργίες όπως αναγνώριση βρόχων και χώρων, αξιοποιώντας τις με συνέπεια την καλύτερη εκτίμηση της τροχιάς και ανακατασκευής του χώρου. Επίσης, παρουσιάζουν λύσεις, που αξιοποιούν αλγορίθμους μηχανικής μάθησης συνδυαστικά με τις υπάρχουσες λύσεις, προσφέροντας ακόμα παραπάνω λειτουργίες, όπως σημασιολογική κατάτμηση του χάρτη, προσφέροντας ακόμα παραπάνω δυνατότητες σε ένα ρομπότ. Επηρεασμένοι από όλα αυτά, διαλέξαμε να ασχοληθούμε με αλγορίθμους που αξιοποιούν την περιττή πληροφορία του περιβάλλοντος όπως η ανίχνευση ευθειών και σημείων φυγής για καλύτερη εκτίμηση και βελτίωση της τροχιάς του ρομπότ. Ακόμη, έχουμε επικεντρωθεί σε περιβάλλοντα εσωτερικών χώρων σπιτιών, που έχουν γραφεία, επίπεδες επιφάνειες όπως τοίχοι, υπολογιστές, καρέκλες κ.α. Επομένως, με βάση το *state of the art* προτείνουμε μια λύση που επεκτείνει το πρώτο και χρησιμοποιεί μια κάμερα, εκμεταλλεύοντας τα σημεία φυγής για εκτίμηση και βελτιστοποίηση της πόζας. Τέλος, παρουσιάζουμε την απαραίτητη θεωρία για τη κατανόηση αυτών των ιδεών και δείχνουμε ότι η μέθοδος μας παρουσιάζει το ίδιο καλά και καλύτερα αποτελέσματα με το *state of the art* σε πραγματικά δεδομένα όπως το TUM-RGBD, σε αυτό του είδους τα περιβάλλοντα.

Λέξεις Κλειδιά

VP-SLAM, Μία κάμερα, Σημεία Φυγής, 3D Ευθείες, Βελτιστοποίηση Πόζας.

Abstract

In the last decades a large part of the scientific community dealing with robotics has focused on solving the Simultaneous Localisation And Mapping (SLAM) problem, i.e. the simultaneous estimation of the robot/camera trajectory and the 3D mapping of the space. This kind of problem is in many robots that carry sensors and their goal is to estimate their position in space and the possibility of perceiving space and reconstructing it. This problem was initially approached with static methods and visual odometry methods without giving a basis to its reconstruction and maintenance. Today, when the community is developing at a rapid pace, the research has turned its interest to solutions, which multiple sensors for the perception of the environment. Additionally, solutions are proposed that better estimate the robot's trajectory and are able to maintain and reconstruct the map and provide functions such as loop and country recognition, consistently exploiting the better trajectory estimation and space reconstruction. solutions, which utilize machine learning algorithms in combination with the above existing solutions, offering even more functions, such as semantic segmentation of the map, offering even more possibilities to a robot. the detection lines and vanishing points for better estimation and improvement of the robot's trajectory. Also, we have focused on interior spaces of houses, which have desks, and flat surfaces such as walls, computers, chairs, etc. Therefore, based on state of the art we propose a solution that extends the first and looks at a camera, exploiting the points escape to estimate and optimize the pose. Finally, we present the necessary theory to understand these ideas and show that our method performs as well and better than state-of-the-art to real data such as TUM-RGBD, in these kinds of environments.

Keywords

VP-SLAM, Monocular, Vanishing Points, 3D Lines, Pose optimization.

στην οικογένεια μου

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Πέτρο Μαραγκό για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Σημάτων Ελέγχου και Ρομποτικής και να εντρυφήσω σε αυτό το ερευνητικό πεδίο. Επίσης, ευχαριστώ ιδιαίτερα τον Υ.Δ.Παναγιώτη Μέρμιγκα για την καθοδήγησή του, την τεχνική του υποστήριξη και την καλή συνεργασία που είχαμε και τις εποικοδομητικές συζητήσεις που είχαμε και με βοήθησαν να κατανοήσω το πρόβλημα του SLAM. Τέλος, θα ήθελα να εκφράσω την αγάπη μου για την οικογένεια μου, την κοπέλα μου και τους φίλους μου για την υποστήριξη τους και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Περιεχόμενα

Περίληψη	1
Abstract	3
Ευχαριστίες	7
1 Εισαγωγή	15
1.1 Σχετικές εργασίες	16
1.2 Συμβολή μας	17
1.3 Οργάνωση της εργασίας	18
2 Θεωρητικό υπόβαθρο	19
2.1 Εισαγωγή	19
2.2 Αισθητήρες συλλογής δεδομένων	20
2.2.1 Γεωμετρία Κάμερας	20
2.2.2 Μονοσκοπική κάμερα (Monocular VSLAM)	22
2.2.3 Στερεοπτική κάμερα (Stereo VSLAM)	22
2.2.4 RGB-D κάμερα (RGB-D VSLAM)	22
2.3 Visual Odometry (VO)	22
2.4 Βελτιστοποίηση (Backend) συστήματος	23
2.5 Κλείσιμο βρόχου (Loop Closure)	24
2.6 Ανακατασκευή (Reconstruction-Mapping)	24
2.6.1 Μετρικοί Χάρτες (Metric Maps)	24
2.6.2 Τοπολογικοί Χάρτες (Topological Maps)	25
2.7 Πρώτες Υλοποιήσεις Visual SLAM συστημάτων	25
2.7.1 Φίλτρο Bayes	25
2.7.2 Kalman Φίλτρο	27
2.7.3 Εκτεταμένο Kalman Φίλτρο (Extended Kalman Filter-EKF)	29
2.7.4 Μη παραμετρικό Φιλτράρισμα, Φίλτρο Σωματιδίων (Non-parametric filtering and Particle Filter-PF)	30
2.7.5 Γράφος Πόζας (Pose Graph)	30
2.8 Επαναληπτική Μη γραμμική Βελτιστοποίηση (Iterative Non-linear Optimization)	33
2.8.1 Προσέγγιση στο Manifold χώρο	36

3	3D Ευθείες και Σημεία Φυγής	41
3.1	Μαθηματική Περιγραφή ευθειών	41
3.1.1	Plücker Πίνακας	42
3.2	Σημεία Φυγής	43
3.2.1	Εκτίμηση Σημείων Φυγής	45
4	Εκτίμηση Τροχιάς Κάμερας και Βελτιστοποιήσεις	49
4.1	3D Κίνηση Άκαμπτου Σώματος	49
4.2	Εκτίμηση πόζας (Pose Estimation)	50
4.2.1	Μέσω 2D/2D αντιστοιχίσεων	50
4.2.2	Μέσω 2D/3D αντιστοιχίσεων	53
4.2.3	Μέσω 3D/3D αντιστοιχίσεων	53
4.2.4	Μέθοδοι Φωτεινότητας (Direct methods)	54
4.2.5	Μέσω διαχωρισμού εκτίμησης πόζας	54
4.3	Βελτιστοποιήσεις (Back end) του συστήματος	55
4.3.1	Bundle Adjustment	55
5	Υλοποίηση	59
5.1	Παρουσίαση Μεθοδολογίας	59
5.2	Υλοποιήσεις καλύτερου συστήματος (State of the Art Implementation)	60
5.3	Υλοποίηση	61
5.3.1	Εξαγωγή Σημείων Φυγής (VP extraction)	63
5.3.2	Εκτίμηση Περιστροφής (Rotation Estimation)	63
5.3.3	Βελτιστοποίηση Περιστροφής (Rotation Refinement)	63
5.3.4	Εκτίμηση Μετατόπισης (Translation Estimation)	66
6	Πειράματα	69
6.1	Μέθοδος εκτίμησης Σφάλματος Τροχιάς	69
6.2	Σύνολα Δεδομένων	71
6.3	Αποτελέσματα	71
6.3.1	Σχολιασμός Αποτελεσμάτων	74
7	Συμπεράσματα	77
7.1	Επίλογος	77
7.2	Μελλοντική Εργασία	78
	Βιβλιογραφία	87

Κατάλογος Σχημάτων

2.1	Κλασική αρχιτεκτονική SLAM συστήματος [1].	19
2.2	Μοντέλο Κάμερας μικρής οπής [2].	21
2.3	Κίνηση ρομπότ με οδομετρία [3].	23
2.4	Είδη χαρτών: 2Δ χάρτης σημείων, 2Δ πλέγμα χάρτη πληρότητας, 3Δ χάρτης σημείων.	26
2.5	Αναπαράσταση SLAM πρόβληματος, ως γράφο [4].	27
2.6	Διαδικασία Kalman Φίλτρου [5].	29
2.7	Αρχικοποίηση Φίλτρου Σωματιδίων	31
2.8	Ανανέωση Φίλτρου Σωματιδίων	31
2.9	Μέτρηση Φίλτρου Σωματιδίων	31
2.10	Απόδοση βαρών Φίλτρου Σωματιδίων	31
2.11	Δειγματοληψία Φίλτρου Σωματιδίων	31
2.12	Ανανέωση κίνησης Φίλτρου Σωματιδίων	31
2.13	Μέτρηση Φίλτρου Σωματιδίων	31
2.14	Διαδικασία Φίλτρου Σωματιδίων [6].	31
2.15	Αναπαράσταση SLAM προβλήματος, ως γράφο πόζας με κόμβους πόζας και κόμβους αντικειμένων χάρτη [7].	32
2.16	Λεπτομερής αναπαράσταση του γράφου πόζας, όπου U, L, X, Z έχουν οριστεί στην ενότητα.2.7.1 [8].	33
2.17	Gauss-Newton εξέλιξη αλγορίθμου προς σύγκλιση [9].	34
2.18	Αναπαράσταση της αντικειμενικής συνάρτησης ως γράφος σχέσεων [10].	35
2.19	Αντιστοίχιση μεταξύ Ευκλείδειου χώρου και εφαιπτομενικού [11].	37
3.1	3Δ ευθεία [12].	42
3.2	Σημείο Φυγής [13].	43
3.3	Δημιουργία σημείου φυγής [14].	44
3.4	Το σημείο φυγής, v , μιας ευθείας με κατεύθυνση d είναι η τομή του επιπέδου της εικόνας με μια ακτίνα παράλληλη στο d έως το C [15].	44
3.5	Προβολή παράλληλων ευθειών σε δακτύλιους στη Gaussian σφαίρα [16].	45
3.6	Σημείο φυγής ως τομή δύο δακτυλιών στην Gaussian σφαίρα [17].	46
3.7	Παραγωγή υποθέσεων σημείων φυγής [18].	48
4.1	3Δ Άκαμπος Μετασχηματισμός T μεταξύ γειτονικών συστημάτων συντεταγμένων [19].	50
4.2	2Δ/2Δ αντιστοιχίσεις μεταξύ δύο καμερών (C_1, C_2) [20].	51

4.3	Εύρεση σχετικής περιστροφής από την εύρεση του πίνακα ομογραφίας H [21].	52
4.4	Απόσταση σημείου από επίπεδο μεταξύ δύο καμπυλών [22].	54
5.1	Επισκόπηση ORB-SLAM συστήματος, παρουσιάζοντας όλα τα βήματα από την εκτίμηση τροχιάς μέχρι της τοπικής χαρτογράφησης και κλεισίματος βρόχου. Εμφανίζονται, επίσης τα κύρια στοιχεία της αναγνώρισης περιοχής (Place Recognition) και του χάρτη (Map) [23].	60
5.2	Ανίχνευση ORB χαρακτηριστικών από το ORB-SLAM2 συστήματα [23].	61
5.3	Επισκόπηση PL-SLAM συστήματος [24].	62
5.4	Επισκόπηση Tracking module του PL-SLAM συστήματος [24].	64
5.5	Επισκόπηση Αρχιτεκτονικής μας. Με πράσινο είναι τα τμήματα που έχουμε εισάγει.	65
6.1	Σφάλμα Απόλυτης τροχιάς (ATE) μεταξύ GT τροχιάς και εκτιμώμενης [25].	70
6.2	Αλγόριθμος εκτίμησης ευθυγράμμισης τροχιών [25].	70
6.3	Στιγμιότυπο εικόνας από το TUM fr3-large-cabinet.	72
6.4	Με κόκκινο, πράσινο και μπλε χρώμα είναι οι εικόνες που αντιστοιχούν στα σημεία φυγής που έχει εξάγει ο αλγόριθμος που χρησιμοποιούμε στην αρχιτεκτονική μας στο TUM fr3-structure-texture-far.	72
6.5	Αποτυχία ταξινόμησης ευθείας.	73
6.6	Κατευθύνσεις σημείων φυγής με μπλε, κόκκινο και πράσινο χρώμα και ταξινόμηση αντίστοιχων ευθειών	73
6.7	Παραπάνω φαίνονται οι τροχιές που παρήγαγε η αρχιτεκτονική μας στο TUM dataset. Με μαύρο είναι η αληθινή τροχιά ενώ με κόκκινο είναι η διαφορά της πραγματικής με την εκτιμώμενη.	74

Κατάλογος Πινάκων

1.1	Με κάποια παραδείγματα εφαρμογών για <i>SLAM</i> . Με μαύρο είναι οι εφαρμογές που διαλέξαμε να συγκρίνουμε την διπλωματική μας εργασία.	18
2.1	Διαδικασία και εξισώσεις Kalman Φίλτρου [5].	29
6.1	Αποτελέσματα στο TUM-RGB-D για το RMSE ATE (m)	75
6.2	Χρόνος Εκτέλεσης των νημάτων Παρακολούθησης και Τοπικής Χαρτογράφησης	75

Κεφάλαιο **1**

Εισαγωγή

Ζούμε σε μια εποχή, που η τεχνολογία καλπάζει με ραγδαίους ρυθμούς και καθημερινά δημιουργούνται προϊόντα και αγαθά με σκοπό την κάλυψη και εξυπηρέτηση των αναγκών των ανθρώπων. Ολοένα και περισσότερα προϊόντα ρομποτικής, αναπτύσσονται και εισέρχονται στη ζωή μας και την καθημερινότητά μας. Συγκεκριμένα, ο κλάδος της ρομποτικής, σημειώνει μεγάλη ανάπτυξη σε προϊόντα που αφορούν αυτοκινούμενα ρομπότ, που είναι σε θέση να παρατηρούν και να αντιλαμβάνονται το περιβάλλοντα χώρο και να παίρνουν δράσεις. Προϊόντα, όπως το ρομπότ σκούπα, η αυτόνομη οδήγηση, ή τα αυτόνομα αεροσκάφη, αποτελούνται από αλγόριθμους που έχουν αναπτυχθεί στο κλάδο της ρομποτικής μαζί με τον κλάδο της όρασης υπολογιστών. Αυτά, τα προϊόντα έχουν κάποια κοινά στοιχεία. Στοιχεία, όπως η αντίληψη του περιβάλλοντος στον οποίο κινούνται. Η θέση στην οποία βρίσκονται μέσα στο χώρο, με βάση ένα σύστημα αναφοράς. Επίσης, τα περισσότερα προϊόντα, για να μπορέσουν να εκτελέσουν αυτές τις διεργασίες σε συγκριτικό βαθμό με τον άνθρωπο θα πρέπει να αντιλαμβάνονται και να καταγράφουν το περιβάλλον με τρόπο κοντινό με αυτό του ανθρώπου. Παρατηρούμε, δηλαδή ότι η έννοια της αντίληψης του χώρου είναι ουσιαστική για την ανάπτυξη ρομπότ, που θα εξυπηρετούν τον άνθρωπο και πόσο μάλλον χρησιμοποιώντας κάμερες. Αντικατοπτρίζοντας, έτσι τη στενή σχέση που έχουν η Όραση Υπολογιστών με τη Ρομποτική. Τα βασικά ερωτήματα λοιπόν, που αφορά ένα ρομπότ είναι: ποιά είναι η θέση μου στο χώρο και πώς και ποιά είναι το περιβάλλον που αντιλαμβάνομαι.

Αυτά, τα ερωτήματα εμπίπτουν στο πρόβλημα της ταυτόχρονης εκτίμησης της θέσης και της ανακατασκευής ενός ρομπότ, κινούμενο σε ένα άγνωστο περιβάλλον (γνωστό και ως *Simultaneous Localization and Mapping SLAM*). Για το πρόβλημα αυτό, έχουν ασχοληθεί πάρα πολλοί ερευνητές και έχουν προταθεί διάφοροι αλγόριθμοι για την επίλυση αυτού του προβλήματος. Η ανάπτυξη αυτών των αλγορίθμων, εξαρτιόταν σε μεγάλο βαθμό, από το είδος των αισθητήρων για την καταγραφή του περιβάλλοντος αλλά και η ακρίβεια των μετρήσεων τους. Ξεκινώντας, με απλές υλοποιήσεις που χρησιμοποιούσαν κάμερα ή κάποιον αισθητήρα *laser (LIDAR)* και με μεθοδολογίες που έκαναν χρήση στατιστικών μοντέλων και βασιζόντουσαν κυρίως στην εκτίμηση της πόζας, μη δίνοντας τόσο σημασία στην ανακατασκευή του χάρτη. Στη συνέχεια, που η τεχνολογία άνθιζε, δημιουργήθηκαν αλγόριθμοι που βασιζόνταν στην έννοια της οδομετρίας για την εκτίμησης της πόζας από αισθητήρες *laser* ή από πολλαπλούς αισθητήρες, αδιαφορώντας και αυτοί για τη διατήρηση ενός χάρτη και εστιάζοντας στην εκτίμηση της πόζας. Έπειτα, κατασκευάστηκαν μεθοδολογίες ικανές για να μπορέσουν να λύσουν το πρόβλημα του *SLAM* και να υπερνικήσουν τα προβλήματα των

προηγούμενων μεθόδων, οδηγώντας έτσι σε αλγόριθμους που χρησιμοποιούνται και επεκτείνονται έως και σήμερα και αποτελούν τη βάση για τη δημιουργία καινούριων μεθοδολογιών.

1.1 Σχετικές εργασίες

Πλέον οι περισσότεροι αλγόριθμοι που δημιουργούνται για το πρόβλημα του *SLAM*, στηρίζονται σε αισθητήρες κάμερας ή σε συνδυασμό κάμερας με γυροσκόπια (*IMU*) και διακρίνονται σε τρεις κατηγορίες: Αλγόριθμοι βασισμένοι στα χαρακτηριστικά για την εκτίμηση και ανακατασκευή του περιβάλλοντος, αλγόριθμοι βασισμένοι στην ένταση της εικόνας και υβριδικοί αλγόριθμοι που συνδυάζουν τους προηγούμενους δυο αλγόριθμους. Επίσης, παρατηρούμε και μοντέρνους αλγόριθμους που συνδυάζουν, νευρωνικά δίκτυα, είτε για να προσφέρουν μια εκτίμηση του βάθους, ή της πόζας της κάμερας ή της ανίχνευσης κινητών αντικειμένων, είτε για να προσφέρουν μια σημασιολογική κατάτμηση του χώρου ώστε να αυξηθούν οι λειτουργίες ενός κινούμενου ρομπότ.

Οι εφαρμογές που στηρίζονται στα χαρακτηριστικά της εικόνας, αποτελούν μια καλή μέθοδο για την απόκτηση γεωμετρικής πληροφορίας του περιβάλλοντος. Βασικό πλεονέκτημα αυτής της μεθόδου είναι η ευκολία εξαγωγής τέτοιων χαρακτηριστικών, η αμεταβλητότητα τους σε αλλαγές της γωνίας θέασης και η διακριτοποίησή τους. Ωστόσο, απαιτούνται χρονοβόροι αλγόριθμοι ταιριάσματος χαρακτηριστικών και είναι ευαίσθητες στη φωτεινότητα και σε φαινόμενα θολώματος. Μερικά παραδείγματα εφαρμογών, που στηρίζονται στα χαρακτηριστικά είναι η *MonoSLAM* [26] που αποτελεί την πρώτη *real-time* εφαρμογή, που χρησιμοποιεί μια κάμερα και ένα πιθανοτικό μοντέλο για την εκτίμηση της πόζας και την εκτίμηση των 3D θέσεων των *landmarks*. Η *PTAM* [27] ικανή για αναπαράσταση μικρών χώρων, που στηρίζεται στην εξαγωγή γωνιών χρησιμοποιώντας τον *FAST* [28] ανιχνευτή. Η *ORB-SLAM* [23] που στηρίζεται στην εξαγωγή *FAST* [28] χαρακτηριστικών, χρησιμοποιώντας τον *ORB* [29] περιγραφή, δημιουργεί μια αραιή ανακατασκευή του χώρου, παρέχει λειτουργίες αναγνώρισης βρόχου και περιβάλλοντος και για την βελτιστοποίηση της τροχιάς και του χάρτη, διενεργεί τοπικά και ολικά *Bundle Adjustment* [30] χρησιμοποιώντας έναν γράφο πόζας. Αποτελεί μια από τις καλύτερες *monocular Visual SLAM* εφαρμογές και έχει επεκταθεί σε εφαρμογές για *RGB-D* και *stereo* δεδομένα. Τέλος, το *OKVIS* [31] που συνδυάζει πληροφορία από κάμερα μαζί με πληροφορία από *IMU* και βασίζεται στην εξαγωγή γωνιών, περιγράφοντας τις με τον *BRISK* [32] περιγραφητή.

Στην περίπτωση των αλγόριθμων που χρησιμοποιούν την ίδια την εικόνα, γνωστοί και ως *direct* αλγόριθμοι, βασίζονται σε φωτομετρικά σφάλματα για την εκτίμηση της τροχιάς και κατασκευάζουν τον περιβάλλον χρησιμοποιώντας μόνο τις φωτεινότητες των εικόνων, προσφέροντας μεγαλύτερη ευρωστία σε περιβάλλοντα με χαμηλότερη υφή (όπως εσωτερικοί χώροι) και αξιοποιούν όλη την πληροφορία της εικόνας, χωρίς να απαιτούνται αλγόριθμοι εξαγωγής και ταιριάσματος χαρακτηριστικών. Σε αυτού του είδους τις εφαρμογές η εκτίμηση της πόζας και η ανακατασκευή του χώρου, επιτυγχάνεται μέσα από την ελαχιστοποίηση ενός φωτομετρικού σφάλματος, σε συνδυασμό με τον αλγόριθμο βελτιστοποίησης γράφου πόζας 7 βαθμών ελευθερίας, για γεωμετρική συνοχή (*LSD – SLAM* [33]). Ωστόσο, υπάρχουν και παραλλαγές αυτών, που εκτιμούν την πόζα και την ανακατασκευή του χώρου, βελτιστοποιώντας μόνο το φωτομετρικό σφάλμα σε ένα σύνολο από *frames* ([33], [34], [35], [36]).

Παρόλα αυτά, ένα μειονέκτημα που παρουσιάζουν αυτές οι μέθοδοι, είναι η αδυναμία τους να εκτιμήσουν με ακρίβεια την πόζα, λόγω αλλαγών στη φωτεινότητα του περιβάλλοντος και στα φαινόμενα έκθεσης του φακού.

Έτσι λοιπόν, πολλοί ερευνητές έχουν στραφεί σε μεθόδους που αξιοποιούν ολοένα και περισσότερη πληροφορία από το περιβάλλον, δημιουργώντας εφαρμογές που στηρίζονται στην ανίχνευση περισσότερο γεωμετρικών σχημάτων, όπως, ευθείες, επίπεδα και ελλείψεις. Αλγόριθμοι που έχουν δημιουργηθεί προς αυτήν την κατεύθυνση είναι οι ([37],[38],[39],[40]). Χρησιμοποιώντας τέτοιου είδους γεωμετρικά σχήματα, η εκτίμηση της πόζας, μπορεί να υπολογιστεί βρίσκοντας πρώτα αντιστοιχίσεις μεταξύ διαδοχικών *keyframes* ή εικόνας και χάρτη και ορίζοντας, ένα μικτό σφάλμα, που περιέχει όρους σφαλμάτων μεταξύ σημείων-προσσημείων, σημείων και ευθειών, σημείων και επιπέδων, ευθειών-προσ-ευθειών και ευθειών με επίπεδα. Έπειτα χρησιμοποιούνται γνωστοί αλγόριθμοι βελτιστοποίησης ελαχίστων τετραγώνων. Διαφορετικά χρησιμοποιούνται σφάλματα τύπου *ICP* και οι παραλλαγές του τόσο για άκαμπτες σκηνές [39] όσο και για εύκαμπτες παραμορφωμένες σκηνές [38].

Τέλος, έχουν προταθεί και καινούριες μέθοδοι που χρησιμοποιούν ως βοηθητικά στοιχεία νευρωνικά δίκτυα, σε περιπτώσεις μιας κάμερας ή ενός αισθητήρα *RGB - D*, με στόχο να εκτιμήσουν είτε τη σχετική πόζα από *frame* σε *frame* [41], είτε το βάθος της εικόνας ([42],[43]), είτε τα κάθετα διανύσματα των επιπέδων, που σε συνδυασμό με τις ευθείες μπορούν να εκτιμήσουν την πόζα [37]. Ακόμη, έχουν χρησιμοποιηθεί και για την αύξηση της σημασιολογικής περιγραφής του χάρτη, υλοποιώντας ένα είδος κατάτμησης του περιβάλλοντος ή απόδοσης ετικετών [43].

Στον πίνακα 1.1 παραθέτουμε επιγραμματικά, ορισμένες εφαρμογές για Visual SLAM και στο οποίο φαίνονται το είδος του αισθητήρα που χρησιμοποιεί με *mono* να αποτελεί τη μια κάμερα και με *stereo* τις δύο κάμερες. Τι είδος χαρακτηριστικών χρησιμοποιεί για την εκτίμηση της πόζας και την ανακατασκευή του χάρτη στη δεύτερη στήλη. Και στις δύο τελευταίες παρατίθενται τα είδη του περιβάλλοντος στα οποία έχουν δοκιμαστεί και την πυκνότητα του χάρτη που παράγουν.

1.2 Συμβολή μας

Με βάση τα παραπάνω, η προσέγγιση μας βασίζεται σε αλγόριθμους που χρησιμοποιούν τα χαρακτηριστικά για την εκτίμηση της πόζας και την ανακατασκευή του χώρου. Συγκεκριμένα, επηρεασμένοι από το γεγονός ότι το περιβάλλον περιέχει πληθώρα πληροφορία και γεωμετρικών σχημάτων, προτείνουμε μια μέθοδο που εκμεταλλεύεται την παρουσία των ευθειών και των σημείων φυγής σε εσωτερικούς χώρους σπιτιών και σε περιβάλλοντα που έχουν γραφεία, τοίχοι και επίπεδες επιφάνειες. Σκοπός μας είναι η χρήση αυτών, για την επέκταση υπάρχοντα αλγορίθμου, για την εισαγωγή μεθοδολογιών που θα εκτιμούν και βελτιώνουν την πόζα. Στη συνέχεια, θα αναλύσουμε τη μεθοδολογία μας και τους τρόπους που μπορούμε να ανιχνεύσουμε ευθείες και σημεία φυγής και με ποιόν τρόπο να προσθέτουμε στο σύστημα μας. Επίσης, θα ελέγξουμε τη μεθοδολογία μας, σε πραγματικά δεδομένα και θα δείξουμε ότι το σύστημα όταν εκτελείται σε περιβάλλον που μπορεί να ανιχνεύσει την περιττή πληροφορία θα την αξιοποιήσει και θα βελτιώσει την τροχιά του συστήματος.

Σύστημα	Αισθητήρας	Χαρακτηριστικά	Περιβάλλον	Χάρτης
PTAM [27]	mono	points	any	Αραιός
LSD-SLAM [33]	mono	image	any	Ημιπυκνός
ORB-SLAM2 [23]	mono/stereo,IMU,RGB-D	points	any	Αραιός
D3VO [41]	mono	image/CNN	any	Ημιπυκνός
DeepFusion [42]	mono	image/CNN	any	Πυκνός
CNN-SLAM [43]	mono	image/CNN	Indoor	Πυκνός
Structural-SLAM[37]	mono	points,lines,planes	Indoor	Αραιός
Nardi et.al.[39]	RGB-D	points,lines,planes	Indoor	Πυκνός
PL-SLAM stereo [44]	stereo	points,lines	any	Αραιός
PL-SLAM [24]	mono	points,lines	any	Αραιός
Haoang Li et.al.[45]	mono	points,lines,VPS	indoor	Αραιός
Elastic Fusion [38]	RGB-D	Hybrid	Indoor	Πυκνός
Jiacheng Liu et.al [46]	mono	points,lines,VPS	indoor	Αραιός
Sudipta N. et.al [47]	mono	points,lines,VPS	any	Αραιός

Πίνακας 1.1: Με κάποια παραδείγματα εφαρμογών για SLAM. Με μαύρο είναι οι εφαρμογές που διαλέξαμε να συγκρίνουμε την διπλωματική μας εργασία.

1.3 Οργάνωση της εργασίας

Η δομή αυτής της διπλωματικής εργασίας είναι ως εξής:

- Στο Κεφάλαιο 2 παρουσιάζουμε τη βασική αρχιτεκτονική ενός συστήματος που λύνει το πρόβλημα του SLAM, μαζί με θεωρία βελτιστοποίησης όπως και τις πρώιμες υλοποιήσεις για το πρόβλημα αυτό.
- Στο Κεφάλαιο 3 εισάγουμε τη μαθηματική διατύπωση για τις ευθείες και τα σημεία φυγής και τους τρόπους με τους οποίους μπορούμε να τα εκτιμήσουμε.
- Στο Κεφάλαιο 4 παρουσιάζουμε τις μεθόδους εκτίμησης της τροχιάς/πόζας.
- Στο Κεφάλαιο 5 παρουσιάζουμε τις αρχιτεκτονικές με τις οποίες συγκριθήκαμε όπως επίσης και την υλοποίηση της αρχιτεκτονικής μας.
- Στο Κεφάλαιο 6 παρουσιάζουμε τις μετρικές που χρησιμοποιούμε για την μέτρηση της απώδωσης του συστήματος μαζί με τα διάφορα σύνολα δεδομένων καθώς και τα αποτελέσματα που πέτυχε η αρχιτεκτονικής μας, όπως επίσης και το χρόνο εκτέλεσης των τριών μεθοδολογιών.
- Στο Κεφάλαιο 7 παρουσιάζουμε τα συμπεράσματα μας από την έρευνα μας καθώς και τη μελλοντική εργασία.

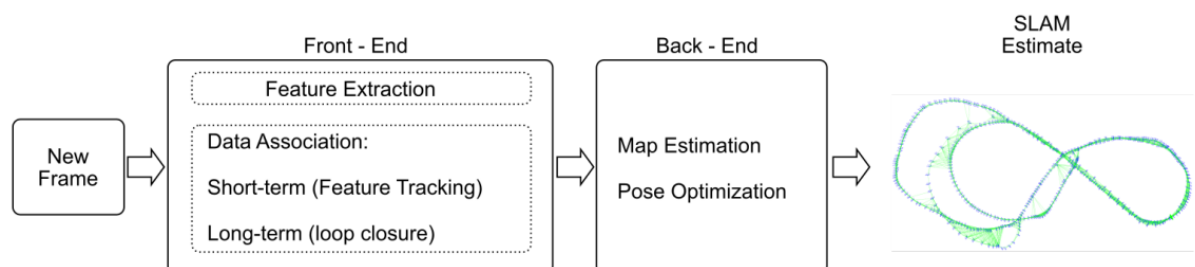
Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Αυτό το κεφάλαιο είναι αφιερωμένο στην παρουσίαση μιας επισκόπησης υφιστάμενων εργασιών σχετικά με την κίνηση ενός κινητού ρομπότ σε ένα άγνωστο περιβάλλον γνωστό και ως *SLAM*. Τα συστήματα που μας απασχολούν καλύπτουν ένα ευρύ φάσμα θεωρίας, ξεκινώντας από διάφορες ιδέες όπως η οδομετρία (*odometry*) και επεκτείνοντας μέχρι τις σύγχρονες έννοιες του *SLAM*.

2.1 Εισαγωγή

Τις τελευταίες δεκαετίες, έχει δοθεί μεγάλη προσοχή στη λύση του προβλήματος της ταυτόχρονης τοπικής προσαρμογής και χαρτογράφησης (*SLAM*) από την κοινότητα της κινητής ρομποτικής. Είναι το πρόβλημα που «ρωτάει εάν είναι δυνατόν ένα κινητό ρομπότ να τοποθετηθεί σε μια άγνωστη τοποθεσία σε ένα άγνωστο περιβάλλον και για το ρομπότ να δημιουργήσει σταδιακά έναν συνεπή χάρτη αυτού του περιβάλλοντος ενώ ταυτόχρονα προσδιορίζει τη θέση του μέσα σε αυτόν τον χάρτη». Ανά τα χρόνια έχουν δημιουργηθεί διάφοροι αλγόριθμοι που λύνουν το πρόβλημα σε πραγματικό χρόνο σε στατικά και κίνουμπε περιβάλλοντα και θα μπορούσαμε να καταλήξουμε σε ένα χονδρικό συμπέρασμα, ότι εάν το περιβάλλον εργασίας περιοριστεί σε σταθερό και άκαμπτο με σταθερές συνθήκες φωτισμού και χωρίς ανθρώπινη παρέμβαση, το πρόβλημα του οπτικού *SLAM* ουσιαστικά λύνεται [βαλε κάποιο ρεφερενς]. Ωστόσο, όλοι οι αλγόριθμοι που έχουν αναπτυχθεί έχουν βασιστεί στην παρακάτω αρχιτεκτονική του συστήματος *SLAM* (βλ.σχήμα.2.1). Ένα τυπικό *SLAM* σύστημα



Σχήμα 2.1: Κλασική αρχιτεκτονική *SLAM* συστήματος [1].

περιλαμβάνει τα ακόλουθα βήματα :

- **Sensor data acquisition.** Αυτό αναφέρεται κυρίως στη λήψη και προ επεξεργασία εικόνων από κάμερες. Για ένα κινητό ρομπότ, αυτό θα περιλαμβάνει επίσης την α-

πόκτηση και τον συγχρονισμό με κωδικοποιητές κινητήρα, αισθητήρες *IMU* κ.λ.π.

- **Visual Odometry (VO).** Το καθήκον του VO είναι να εκτιμήσει την κίνηση της κάμερας μεταξύ γειτονικών καρέ (ego-motion) και να δημιουργήσει έναν πρόχειρο τοπικό χάρτη. Το VO είναι επίσης γνωστό και ως frontend.
- **Backend filtering/optimization.** Το backend λαμβάνει πόζες κάμερας σε διαφορετικές χρονικές στιγμές από το VO και από το κλείσιμο βρόχου (Loop Closing) και στη συνέχεια, εφαρμόζει βελτιστοποίηση για να δημιουργήσει μια πλήρως βελτιστοποιημένη τροχιά και χάρτη. Επειδή συνδέεται μετά το VO, είναι επίσης γνωστό ως backend.
- **Loop Closing.** Το κλείσιμο βρόχου (Loop Closing) καθορίζει εάν το ρομπότ έχει επιστρέψει στην προηγούμενη θέση του, προκειμένου να μειωθεί το συσσωρευμένο σφάλμα λόγω κακής εκτίμησης της περιστροφής. Εάν εντοπιστεί βρόχος, θα παρέχει πληροφορίες στο backend για περαιτέρω βελτιστοποίηση.
- **Reconstruction.** Κατασκευάζει έναν χάρτη για συγκεκριμένη εργασία με βάση την εκτιμώμενη τροχιά της κάμερας.

2.2 Αισθητήρες συλλογής δεδομένων

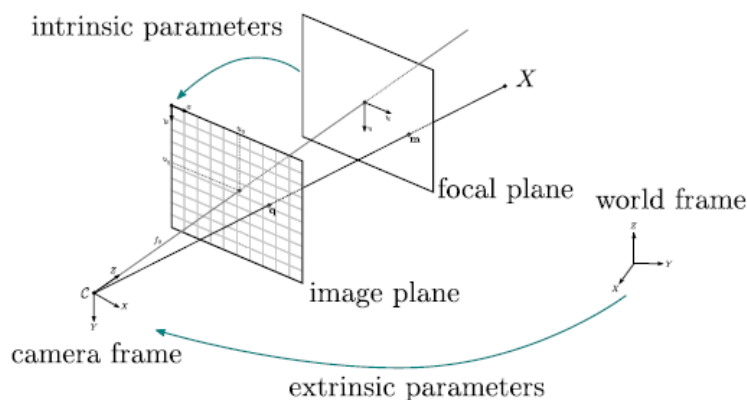
Εξετάζοντας πιο βαθιά τα στοιχεία ενός συστήματος *SLAM* παρατηρούμε ότι πέραν των διαφόρων αλγορίθμων και αρχιτεκτονικών που έχουν αναπτυχθεί για να λύσουν αυτό το πρόβλημα, το είδος του αισθητήρα που φέρει για να αντιλαμβάνεται το περιβάλλον του αποτελεί, ιδιαίτερο χαρακτηριστικό στην αρχιτεκτονική του εκάστοτε συστήματος. Είναι δηλαδή, πολύ σημαντικός ο τρόπος με τον οποίο ένα *SLAM* σύστημα συλλέγει τα δεδομένα από το περιβάλλον καθώς και η χρήση του εκάστοτε αισθητήρα. Έτσι λοιπόν, για το πρόβλημα του *SLAM* οι αισθητήρες που συνήθως χρησιμοποιούνται είναι είτε η χρήση μια κάμερας (ή αλλιώς μονοσκοπική κάμερα), είτε η χρήση στερεοοπτικής κάμερας είτε η χρήση *RGB – D* κάμερας.

2.2.1 Γεωμετρία Κάμερας

Κάθε κάμερα, χαρακτηρίζεται από έναν πίνακα βαθμονόμησης (*Calibration Matrix*) K που περιγράφει τις εσωτερικές παραμέτρους της κάμερας και μια σχέση προβολής των 3D σημείων, στο επίπεδο της εικόνας της, σε 2D σημεία (βλ.σχήμα.2.2).

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Όπου p_x , p_y είναι το κέντρο της κάμερας στο επίπεδο της εικόνας και f είναι το *focal length* της κάμερας [15]. Αυτός ο K αποτελεί την πιο απλή περίπτωση πίνακα βαθμονόμησης κάμερας, που χρησιμοποιείται συχνά στις περισσότερες εφαρμογές *SLAM*, χαρακτηριζόμενο ως μοντέλο μικροσκοπικής οπής *pinhole camera*. Από την προβολική γεωμετρία [15] έχουμε



Σχήμα 2.2: Μοντέλο Κάμερας μικρής οπής [2].

ότι ένα σημείο στο χώρο $X = [x, y, z]^T$ προβάλλεται στο επίπεδο της εικόνας στα *pixel* $[u, v]$ ως:

$$\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2 \quad (2.2)$$

$$(u, v, 1)^T \mapsto K[x, y, z]^T = \left(f \cdot \frac{x}{z} + p_x, f \cdot \frac{y}{z} + p_y, 1 \right) \quad (2.3)$$

Ωστόσο, η παραπάνω σχέση, έχει ληφθεί, μην έχοντας λάβει τις εξωτερικές παράμετροι της κάμερας. Δηλαδή την περιστροφή και τη μετατόπιση που έχει το σύστημα συντεταγμένων της κάμερας ως προς το παγκόσμιο σύστημα συντεταγμένων. Συγκεκριμένα, έστω ότι R , t η περιστροφή και η μετατόπιση του συστήματος συντεταγμένων της κάμερας ως προς το παγκόσμιο σύστημα συντεταγμένων. Τότε, λαμβάνοντας υπόψιν και τις εξωτερικές παραμέτρους της κάμερας, καταλήγουμε ότι ένα 3D σημείο προβάλλεται στο επίπεδο της εικόνας, στα αντίστοιχα *pixel* (u, v) από τον τύπο [15]:

$$(u, v) = K[R | -R^T t] \cdot [x, y, z]^T \quad (2.4)$$

Ενώ σε ομογενείς συντεταγμένες θα έχουμε [14]:

$$(u, v, 1)^T = \begin{bmatrix} KR^T & -KR^T t \\ 0 & 1 \end{bmatrix} (x, y, z, 1)^T \quad (2.5)$$

όπου ο P :

$$P = K \cdot [R | -R^T \cdot t] \quad (2.6)$$

Ο πίνακας P είναι ο προβολικός πίνακας, έχει βαθμό 3 και 11 βαθμούς ελευθερίας. Επίσης, είναι άξιο να σημειωθεί, ότι το προβολικό μοντέλο δεν ισχύει για εικόνες που έχουν διαστρεβλωθεί κατά μήκος του άξονα περιστροφής. Ωστόσο, έχουν προταθεί στη βιβλιογραφία διάφορες μεθόδους επίλυσής του [15]. Από τη σχέση προβολής 2.5 αντιλαμβανόμαστε ότι στην περίπτωση μια κάμερας, από την προβολή ενός σημείου σε αυτήν, μας είναι δύσκολο να εκτιμήσουμε το βάθος του σημείου στις 3 διαστάσεις. Επομένως, για να λυθεί το πρόβλημα αυτό, έχουν προταθεί διάφοροι αλγόριθμοι στη βιβλιογραφία και συνήθως, σε εφαρμογές που χρησιμοποιούν μια κάμερα [23], το βάθος μπορεί να υπολογιστεί χρησιμοποιώντας τη

μέθοδο της τριγωνοποίησης (αξιοποιώντας τη θεωρία της γεωμετρίας πολλαπλών όψεων [15]) ή μπορεί να εκτιμηθεί από ένα συνελκτικό νευρωνικό δίκτυο όπως στο [43].

2.2.2 Μονοσκοπική κάμερα (Monocular VSLAM)

Συγκεκριμένα στην περίπτωση αυτή χρησιμοποιείται μια κάμερα, για την αντίληψη του περιβάλλοντος και την εκτίμηση της θέσης του ρομπότ. Παρόλο, που αποτελεί μια εύκολη και προσιτή λύση σε τέτοια συστήματα, παρουσιάζει προβλήματα όπως η αδυναμία αντίληψης του βάθους του περιβάλλοντα χώρου από μια εικόνα, οδηγώντας έτσι σε συσσώρευση σφάλματος κατά την κίνηση του ρομπότ. Ωστόσο, ανά τα χρόνια έχει γίνει μεγάλη προσπάθεια από πολλούς ερευνητές για να λυθούν αυτά τα προβλήματα, χρησιμοποιώντας μόνο μια κάμερα ([23], [26], [27], [33], [34], [35], [41], [42], [43], [48]).

2.2.3 Στερεοπτική κάμερα (Stereo VSLAM)

Στην περίπτωση, αυτή χρησιμοποιείται ένα σύστημα με δυο συγχρονισμένες κάμερες, για τις οποίες γνωρίζουμε την απόσταση μεταξύ των κέντρων τους ([23], [44], [49], [50], [51]). Αυτό δίνει τη δυνατότητα για άμεσο υπολογισμό της 3D πόζα ενός σημείου, όπως κάνει και ο άνθρωπος χρησιμοποιώντας τα δύο του μάτια, παρακάμπτοντας έτσι τα προβλήματα της αδυναμίας του βάθους και της κλίμακας που παρουσιάζοντας στη χρήση μια κάμερας. Ωστόσο, όμως η ακρίβεια του βάθους από δυο κάμερες εξαρτάται από την απόσταση μεταξύ τους, δηλαδή όσο μεγαλύτερη είναι η απόσταση τους τόσο μεγαλύτερο είναι και το βάθος που μπορούν να μετρήσουν.

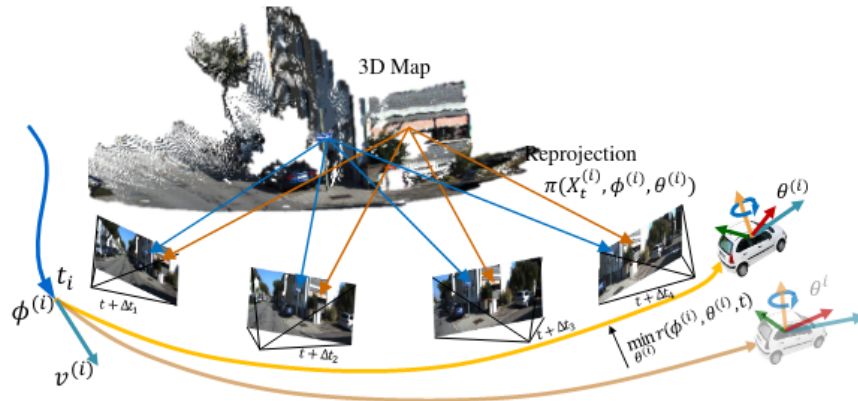
2.2.4 RGB-D κάμερα (RGB-D VSLAM)

Τέλος, μια κάμερα RGB-D είναι ένας αισθητήρας που προσφέρει τόσο οπτικές εικόνες RGB όσο και εικόνες βάθους ([23], [39]). Η εικόνα βάθους, κωδικοποιείται ως εικόνα έντασης σε κλίμακα του γκρι, με κάθε pixel να αντιστοιχεί σε μια τιμή βάθους. Το βάθος συλλαμβάνεται συνήθως χρησιμοποιώντας υπέρυθρες ακτίνες, μετρώντας το χρόνο πτήσης. Σημειώστε ότι τα δεδομένα RGB και βάθους δε συγχρονίζονται απόλυτα από αισθητήρες πραγματικού κόσμου, για αυτό είναι πάντα απαραίτητο ένα βήμα συσχέτισης για την εύρεση των καλύτερων αντιστοιχιών μεταξύ των δύο ροών για να δημιουργηθούν πραγματικά τα τελικά καρτέ που θα περάσουν σε ένα σύστημα προ επεξεργασίας.

2.3 Visual Odometry (VO)

Όταν εξετάζουμε το πρόβλημα VSLAM, είναι φυσικό να αναρωτιόμαστε, εάν ένα σύστημα μπορεί να εκτελέσει μόνο εντοπισμό, δηλαδή να εκτιμήσει τη θέση του στο χώρο χρησιμοποιώντας μόνο την πληροφορία της εικόνας. Η απάντηση βρίσκεται στην οπτική οδομετρία (Visual Odometry) που αφορά την εκτίμηση της κίνησης (θέσης) μιας κάμερας μεταξύ γειτονικών πλαισίων εικόνας και η απλούστερη περίπτωση είναι φυσικά η κίνηση μεταξύ δύο διαδοχικών εικόνων. Με αυτόν τον τρόπο, ο συνολικός μετασχηματισμός μεταξύ της αρχικής πόζας και της τρέχουσας πόζας του αισθητήρα μπορεί να εκτιμηθεί ως το γινόμενο όλων των

ενδιάμεσων ποζών που υπολόγισε το σύστημα. Ωστόσο, η χρήση μόνο ενός VO για την εκτίμηση της τροχιάς θα προκαλέσει αναπόφευκτα αθροιστικό σφάλμα. Αυτό συμβαίνει, γιατί η οπτική οδομετρία (στην απλούστερη περίπτωση) υπολογίζει μόνο την κίνηση μεταξύ δύο πλαισίων. Γνωρίζουμε ότι κάθε εκτίμηση συνοδεύεται από ένα συγκεκριμένο σφάλμα. Λόγω του τρόπου που λειτουργεί η οδομετρία, τα σφάλματα από προηγούμενες στιγμές θα μεταφερθούν στις επόμενες στιγμές, με αποτέλεσμα την ανακριβή εκτίμηση μετά από κάποιο χρονικό διάστημα (πχ. βλ.σχήμα.2.6). Για να λύσουμε αυτό το πρόβλημα, χρειαζόμαστε επίσης άλλα δύο στοιχεία: το backend optimization και το loop closing.



Σχήμα 2.3: Κίνηση ρομπότι με οδομετρία [3].

2.4 Βελτιστοποίηση (Backend) συστήματος

Σε γενικές γραμμές, η βελτιστοποίηση που γίνεται στο τέλος της διαδικασίας του συστήματος Visual SLAM ή αλλιώς backend optimization αναφέρεται κυρίως στη διαδικασία αντιμετώπισης του θορύβου στα συστήματα Visual SLAM. Ελπίζουμε ότι όλα τα δεδομένα των αισθητήρων είναι ακριβή, αλλά στην πραγματικότητα, ακόμη και οι πιο ακριβοί αισθητήρες εξακολουθούν να έχουν ένα συγκεκριμένο ποσό θορύβου. Οι φθηνοί αισθητήρες έχουν συνήθως μεγαλύτερα σφάλματα μέτρησης, ενώ των ακριβών μπορεί να είναι μικρό. Επιπλέον, η απόδοση πολλών αισθητήρων επηρεάζεται από αλλαγές στο μαγνητικό πεδίο, τη θερμοκρασία κ.λ.π. Επομένως, εκτός από την επίλυση του προβλήματος της εκτίμησης της τροχιάς της κάμερας από τις εικόνες, μας ενδιαφέρει επίσης πόσο θόρυβο περιέχει αυτή η εκτίμηση, πώς μεταφέρεται από την προηγούμενη εικόνα στην επόμενη και πόσο σίγουροι είμαστε για την τρέχουσα εκτίμηση τροχιάς. Η backend βελτιστοποίηση λύνει το πρόβλημα της εκτίμησης της κατάστασης ολόκληρου του συστήματος από θορυβώδη δεδομένα εισόδου και τον υπολογισμό της αβεβαιότητάς τους. Ο τρόπος, που επιτυγχάνεται αυτό είναι μέσα από την παραγωγή ενός γράφου πόζας, όπου οι κόμβοι μπορούν να αντιστοιχούν σε σημεία ή στην πόζα του frame και οι συνδέσεις μεταξύ σημείων που ανήκουν σε διαφορετικές ομάδες κόμβων, περιγράφουν το σφάλμα επαναπροβολής. Ενώ οι σχέσεις μεταξύ κόμβων που περιγράφουν την πόζα του frame, αναπαριστούν το πλήθος των κοινών τους στοιχείων. Έτσι λοιπόν, μέσα από την ελαχιστοποίηση όλων των σφαλμάτων επαναπροβολής στο γράφο, βελτιστοποιείται η τροχιά και ο χάρτης.

2.5 Κλείσιμο βρόχου (Loop Closure)

Το κλείσιμο βρόχου (Loop Closure), επίσης γνωστό ως ανίχνευση κλεισίματος βρόχου (loop closure detection), αντιμετωπίζει κυρίως το πρόβλημα της συσσώρευσης του σφάλματος της τροχιάς λόγω κακής εκτίμησης της περιστροφής στο Visual SLAM. Συγκεκριμένα, όταν το ρομπότ κινείται σε ένα περιβάλλον που έχει κύκλο, όταν περάσει από την αρχική του θέση, λόγω του συσσωρευμένου σφάλματος της τροχιάς σε όλη την κίνηση του θα οδηγήσει σε διαφορετική εκτίμηση της θέσης, συγκριτικά με την αρχική του. Επομένως, παρατηρούμε ότι το πρόβλημα του (Loop Closure) διαιρείται σε δυο υπό προβλήματα. Το πρώτο αφορά την αναγνώριση του ίδιου χώρου, που περιλαμβάνει την ανίχνευση ενός γεωμετρικού περιορισμού μεταξύ δύο όψεων κάμερας που τυπικά ανήκουν σε δύο τοπολογικά απομακρυσμένα τμήματα του χάρτη. Και το δεύτερο, αφορά τη διόρθωση του συσσωρευμένου σφάλματος στην τελική εκτίμηση πόζας κατανέμοντας το σφάλμα κατά μήκος της αλυσίδας όλων των θέσεων της κάμερας στην κίνηση της.

2.6 Ανακατασκευή (Reconstruction-Mapping)

Χαρτογράφηση (Mapping) ή ανακατασκευή (Reconstruction) σημαίνει τη διαδικασία κατασκευής ενός χάρτη, όποιου είδους και αν είναι. Ένας χάρτης (βλ. σχήμα.2.4) είναι μια περιγραφή του περιβάλλοντος, αλλά ο τρόπος περιγραφής δεν είναι σταθερός και εξαρτάται από την πραγματική εφαρμογή. Επίσης, αφού η κίνηση του ρομπότ είναι 6 βαθμών ελευθερίας είναι απαραίτητη η χρήση ενός 3D χάρτη για την περιγραφή του περιβάλλοντος. Έτσι λοιπόν, ανάλογα με το είδος της εφαρμογής Visual SLAM οι χάρτες μπορούν να ταξινομηθούν σε δυο είδη: τους τοπολογικούς χάρτες (Topological maps) και τους μετρικούς χάρτες (Metric maps).

2.6.1 Μετρικοί Χάρτες (Metric Maps)

Οι μετρικοί χάρτες τονίζουν τις ακριβείς μετρικές θέσεις των αντικειμένων στους χάρτες. Συνήθως ταξινομούνται είτε ως αραιοί είτε ως πυκνοί. Οι αραιοί μετρικοί χάρτες αναπαριστούν τη σκηνή σε συμπαγή μορφή και δεν εκφράζουν όλα τα αντικείμενα. Για παράδειγμα, μπορούμε να κατασκευάσουμε έναν αραιό χάρτη επιλέγοντας αντιπροσωπευτικά σημεία όπως οι λωρίδες κυκλοφορίας και οι γωνίες των αντικειμένων. Αντίθετα, οι πυκνοί μετρικοί χάρτες επικεντρώνονται στη μοντελοποίηση όλων των πραγμάτων που φαίνονται. Ένας αραιός χάρτης θα ήταν αρκετός για εντοπισμό, ενώ για πλοήγηση χρειάζεται συνήθως ένας πυκνός χάρτης. Ένας πυκνός χάρτης συνήθως αποτελείται από έναν αριθμό μικρών πλεγμάτων σε μια συγκεκριμένη ανάλυση. Μπορεί να είναι μικρά πλέγματα χωρητικότητας για 2D μετρικούς χάρτες ή μικρά κυβικά πλέγματα voxel για τρισδιάστατους χάρτες. Για παράδειγμα, σε έναν δισδιάστατο χάρτη που χωρισμένο σε μικρά τετράγωνα ή αλλιώς πλέγματα, ένα τετράγωνο (πλέγμα) μπορεί να έχει τρεις καταστάσεις: κατειλημμένο, μην κατειλημμένο και άγνωστο, για να εκφράσει εάν υπάρχει ένα αντικείμενο. Όταν τίθεται ερώτημα για μια χωρική τοποθεσία, ο χάρτης μπορεί να παρέχει πληροφορίες σχετικά με το εάν η τοποθεσία είναι προσβάσιμη. Αυτός ο τύπος χαρτών μπορεί να χρησιμοποιηθεί για διάφορους αλγόριθ-

μους πλοήγησης, όπως A^* κ.λ.π., και έτσι να προσελκύσει την προσοχή των ερευνητών της ρομποτικής. Αλλά μπορούμε επίσης να δούμε ότι όλη η κατάσταση του πλέγματος είναι αποθηκευμένη στον χάρτη, με αποτέλεσμα να είναι ακριβή η αποθήκευση.

2.6.2 Τοπολογικοί Χάρτες (Topological Maps)

Σε σύγκριση με τους ακριβείς μετρικούς χάρτες, οι τοπολογικοί χάρτες δίνουν έμφαση στις σχέσεις μεταξύ των στοιχείων του χάρτη. Ένας τοπολογικός χάρτης είναι ένα γράφημα που αποτελείται από κόμβους και ακμές, λαμβάνοντας υπόψη μόνο τη συνδεσιμότητα μεταξύ των κόμβων. Για παράδειγμα, μας ενδιαφέρει μόνο ότι το σημείο A και το σημείο B είναι συνδεδεμένα, ανεξάρτητα από το πώς θα μπορούσαμε να ταξιδέψουμε από το σημείο A στο σημείο B. Χαλαρώνει τις απαιτήσεις για ακριβείς τοποθεσίες ενός χάρτη αφαιρώντας τις λεπτομέρειες του χάρτη και, ως εκ τούτου, είναι περισσότερο συμπαγής έκφραση. Ωστόσο, οι τοπολογικοί χάρτες δεν είναι καλοί στην αναπαράσταση χαρτών με πολύπλοκες δομές. Ερωτήματα όπως ο τρόπος διαχωρισμού ενός χάρτη για να σχηματιστούν κόμβοι και ακμές και ο τρόπος χρήσης ενός τοπολογικού χάρτη για πλοήγηση και σχεδιασμό διαδρομής εξακολουθούν να είναι ανοιχτά προβλήματα προς μελέτη.

2.7 Πρώιμες Υλοποιήσεις Visual SLAM συστημάτων

Στο σχήμα.2.5 απεικονίζουμε ένα τυπικό πρόβλημα (Visual SLAM), όπου ένα κινητό όχημα (τρίγωνο) κινείται σε ένα άγνωστο περιβάλλον και κατά την κίνηση αυτή ανιχνεύει διάφορα χαρακτηριστικά (αντικείμενα) από το περιβάλλον του (κύκλοι). Με στόχο, την καλύτερη εκτίμηση της τροχιάς του και της θέσης αυτών των χαρακτηριστικών, άρα μια βελτιωμένη ανακατασκευή του χάρτη που παρατηρεί. Έτσι, λοιπόν, αντιλαμβανόμαστε ότι υπάρχει μια σχέση εξάρτησης μεταξύ των χαρακτηριστικών του περιβάλλοντος και την πόζας του ρομπότ, που μπορεί να εκφραστεί με πιθανοτικό τρόπο κάνοντας χρήση της θεωρίας του Bayes ή ως ένα δίκτυο σχέσεων. Πιο αναλυτικά, αναλύουμε τη μοντελοποίηση του προβλήματος με πιθανοτικό τρόπο, αξιοποιώντας τη θεωρία του Bayes και στη συνέχεια ως δίκτυο σχέσεων.

2.7.1 Φίλτρο Bayes

Το πρόβλημα SLAM προσεγγίστηκε αρχικά με πιθανοτικές προσεγγίσεις βασισμένες στο κανόνα του Bayes που βασίζεται στη θεμελιώδη αρχή:

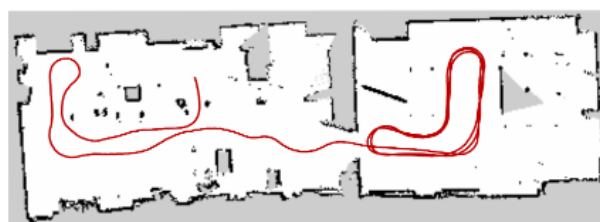
$$p(A|B) = w \cdot p(B|A) \cdot p(A) \quad (2.7)$$

Όπου το $p(A|B)$ συμβολίζει τη δεσμευμένη πιθανότητα να αληθεύει το γεγονός A δεδομένου ότι αληθεύει το γεγονός B. Επίσης, ο κανονικοποιημένος παράγοντας w διασφαλίζει την εγκυρότητα της μορφής του αριστερού μέλους της σχέσης ως συνάρτησης κατανομής πιθανότητας.

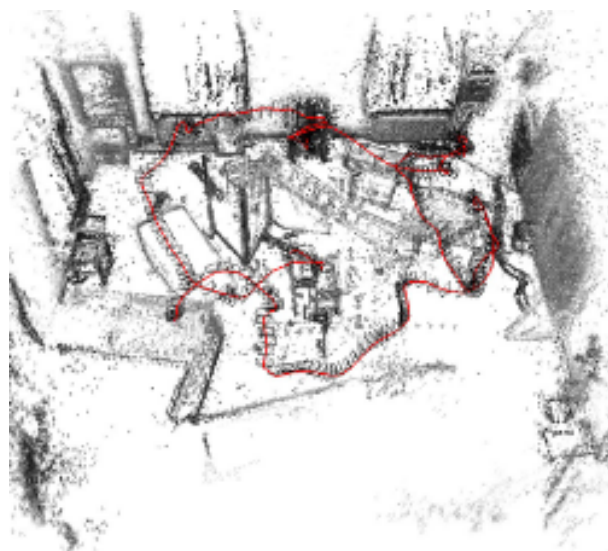
Έτσι λοιπόν, το φίλτρο Bayes επεκτείνει τον ομώνυμο κανόνα 2.7 σε προβλήματα χρονικής εκτίμησης και σχετίζεται άμεσα με τα Φίλτρα Kalman, τα Κρυφά Μαρκοβιανά Μοντέλα, τα Δυναμικά Δίκτυα Bayes (Dynamic Bayes Networks – DBN) και τις Μερικώς Παρατη-



(α) 2Δ χάρτης σημείων [23].

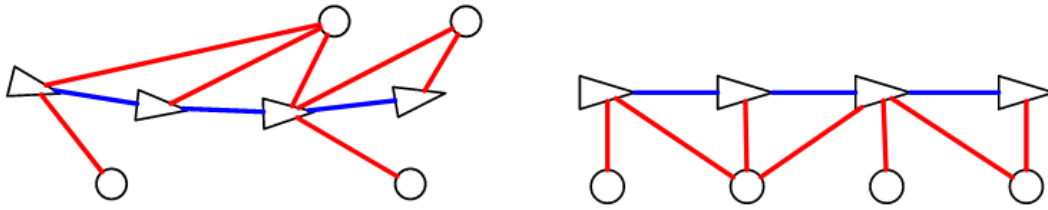


(β) 2Δ πλέγμα χάρτη πληρότητας [52].



(γ) 3Δ χάρτης σημείων [36].

Σχήμα 2.4: Είδη χαρτών: 2Δ χάρτης σημείων, 2Δ πλέγμα χάρτη πληρότητας, 3Δ χάρτης σημείων.



Σχήμα 2.5: Αναπαράσταση SLAM πρόβλεψης, ως γράφο [4].

ρήσιμες Διεργασίες Απόφασης Markov (Partially Observable Markov Decision Processes – POMDPs). Το φίλτρο Bayes είναι ένας επαναληπτικός εκτιμητής για τον υπολογισμό της ακολουθίας «εκ των υστέρων» κατανομών πιθανότητας επί μεγεθών που δεν είναι δυνατό να παρατηρηθούν με άμεσο τρόπο – όπως επί παραδείγματι ένας χάρτης του χώρου. Συγκεκριμένα, το κύριο θέμα στις διάφορες μαθηματικές συνθέσεις που έχουν προταθεί είναι ότι το ρομπότ βρίσκεται σε μια κατάσταση που μπορεί να περιγραφεί χρησιμοποιώντας ένα διάνυσμα κατάστασης \mathbf{x} , οι αισθητήρες του λαμβάνουν μετρήσεις \mathbf{z} και οι ενεργοποιητές τροφοδοτούνται με σήματα ελέγχου \mathbf{u} . Όλες οι παραπάνω ποσότητες δειγματοληπτούνται σε διακριτές χρονικές στιγμές και όλες επηρεάζονται από κάποιο είδος θορύβου, που συνήθως έχει θεωρηθεί ως Gaussian. Δηλαδή, δοθέντος μιας ακολουθίας από μετρήσεις αισθητήρων $\mathbf{z}_{1:t}$ και τα σήματα ελέγχου $\mathbf{u}_{1:t}$ ενός ρομπότ, απαιτούμε να υπολογίσουμε την κατανομή πιθανότητας της τρέχουσας θέσης \mathbf{x}_t . Ακολουθώντας και το [53] έχουμε ότι εάν:

Όλες οι καταστάσεις του ρομπότ $X = \{x_i \mid i = 1, \dots, M\}$

Μεταβλητές Όλα τα σήματα ελέγχου του ρομπότ $U = \{u_i \mid i = 1, \dots, M\}$

Όλες οι μετρήσεις $Z = \{z_i \mid i = 1, \dots, K\}$

η εκ των υστέρων κατανομή πιθανότητας επί των καταστάσεων υπολογίζεται μέσω του φίλτρου Bayes από την ακόλουθη επαναληπτική εξίσωση:

$$p(x_t | z_{1:t}, u_{1:t}) = w \cdot p(z_t | x_t) \cdot \int p(x_t | u_t, x_{t-1}) \cdot p(x_{t-1} | z_{1:t-1}, u_{1:t-1}) dx_{t-1} \quad (2.8)$$

2.7.2 Kalman Φίλτρο

Το Kalman φίλτρο είναι ένα γραμμικό, πεπερασμένων διαστάσεων, χρονολογούμενο σύστημα που αξιολογεί την εκτιμώμενη κατάσταση που ελαχιστοποιεί το σφάλμα μέσου τετραγώνου (MSE) [5] και αποτελεί μια μέθοδο φιλτραρίσματος και πρόβλεψης σε γραμμικά συστήματα.

Για να χρησιμοποιήσει κανείς το Kalman φίλτρο, πρέπει να ορίσει ένα μοντέλο παρατήρησης και ένα μοντέλο συστήματος. Στη συνέχεια, το Kalman φίλτρο λειτουργεί σε δύο βήματα: Πρώτον, το βήμα πρόβλεψης χρησιμοποιεί το μοντέλο συστήματος και τις εισόδους ελέγχου για να εκτιμήσει την επόμενη κατάσταση του συστήματος. Μετά την εφαρμογή της εισόδου ελέγχου, το βήμα ανανέωσης (φιλτράρισμα) χρησιμοποιεί τη νέα παρατήρηση (πέρασμα των μετρήσεων του αισθητήρα μέσω του μοντέλου παρατήρησης) για να βελτιώσει την εκτίμηση της κατάστασης του συστήματος.

Το Kalman φίλτρο προκύπτει με την υπόθεση ότι οι μεταγενέστερες κατανομές πιθανοτήτων (posterior probability distributions) που εμπλέκονται στο Bayesian φιλτράρισμα είναι

γραμμικές σε σχέση με τα ορίσματά τους και έχουν μόνο Gaussian πρόσθετο θόρυβο που εφαρμόζεται σε αυτές. Αν και η παραγωγή είναι κάπως εμπλεκόμενη, ο τελικός αλγόριθμος εξακολουθεί να ακολουθεί μια λογική ενημέρωσης κατάστασης που απηχεί τη δομή του Bayes φίλτρου. Επίσης, μπορεί ναδειχθεί ότι εάν ισχύουν οι παραπάνω παραδοχές, τότε το Kalman φίλτρο συγκλίνει σε μια σταθερή κατάσταση, ελαχιστοποιώντας το σφάλμα μέσου τετραγώνου μεταξύ της πραγματικής κατάστασης του συστήματος και της εκτιμώμενης κατάστασης. Η κατάσταση του συστήματος αντιπροσωπεύεται από μια συνάρτηση πυκνότητας πιθανότητας που, στην περίπτωση που ισχύουν όλες οι παραδοχές που περιγράφηκαν προηγουμένως, είναι επίσης Gaussian. Αυτό σημαίνει ότι μπορεί να περιγραφεί μόνο από τον μέσο όρο και τη συνδιακύμανσή του. Με βάση το [5] έχουμε:

Σύστημα Μοντέλου :

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k, k \geq 0. \quad (2.9)$$

Όπου $x_k \in R^n$ το διάνυσμα μεταβλητών κατάστασης του συστήματος τη χρονική στιγμή $t = kT$, A , B οι πίνακες μετάβασης κατάστασης και εισόδου, αντίστοιχα, u_k το διάνυσμα εισόδου και w_k ένα διάνυσμα γκαουσιανού λευκού θορύβου μηδενικής μέσης τιμής και συνδιακύμανσης Q .

Σύστημα Παρατήρησης :

$$z_k = C_k x_k + v_k \quad (2.10)$$

Όπου $z_k \in R^m$ διάνυσμα μέτρησης, C_k ο πίνακας συσχέτισης μέτρησης-κατάστασης και v_k ένα διάνυσμα γκαουσιανού λευκού θορύβου μηδενικής μέσης τιμής και συνδιακύμανσης R .

Ερμηνεύοντας τις παραπάνω εξισώσεις μπορεί κανείς να δει ότι αντιπροσωπεύουν ένα κλασικό γραμμικό διακριτό σύστημα χρόνου. Τη χρονική στιγμή $k + 1$, το διάνυσμα κατάστασης x_{k+1} εξαρτάται γραμμικά από την κατάσταση της προηγούμενης χρονικής στιγμής x_k και από ένα σήμα ελέγχου u_k . Το λευκό, Gaussian, μηδενικό διάνυσμα θορύβου w_k είναι υπεύθυνο για τη μοντελοποίηση των πηγών σφάλματος του συστήματος. Συγκεκριμένα, η αβεβαιότητα του διανύσματος ελέγχου, η ανακρίβεια του μοντέλου (A_k) και άλλοι εξωτερικοί παράγοντες όπως η ολίσθηση του τροχού κ.λ.π.

Έστω με \widehat{x}_k^- συμβολίσουμε την εκ των προτέρων (a priori) εκτίμηση του διανύσματος κατάστασης x_k τη χρονική στιγμή k , δεδομένου του ότι γνωρίζουμε επακριβώς τη διαδικασία μέχρι και τη στιγμή $k - 1$ και με \widehat{x}_k συμβολίσουμε την εκ των υστέρων (a posteriori) εκτίμηση του x_k έχοντας στη διάθεσή μας τη μέτρηση z_k τότε μπορούμε να ορίσουμε αντίστοιχα το εκ των προτέρων και εκ των υστέρων σφάλμα εκτίμησης:

$$e_k^- = x_k - \widehat{x}_k^- \quad (2.11)$$

και

$$e_k = x_k - \widehat{x}_k \quad (2.12)$$

με αντίστοιχους πίνακες συνδιακύμανσης:

$$P_k^- = E(e_k e_k^{-T}) \quad (2.13)$$

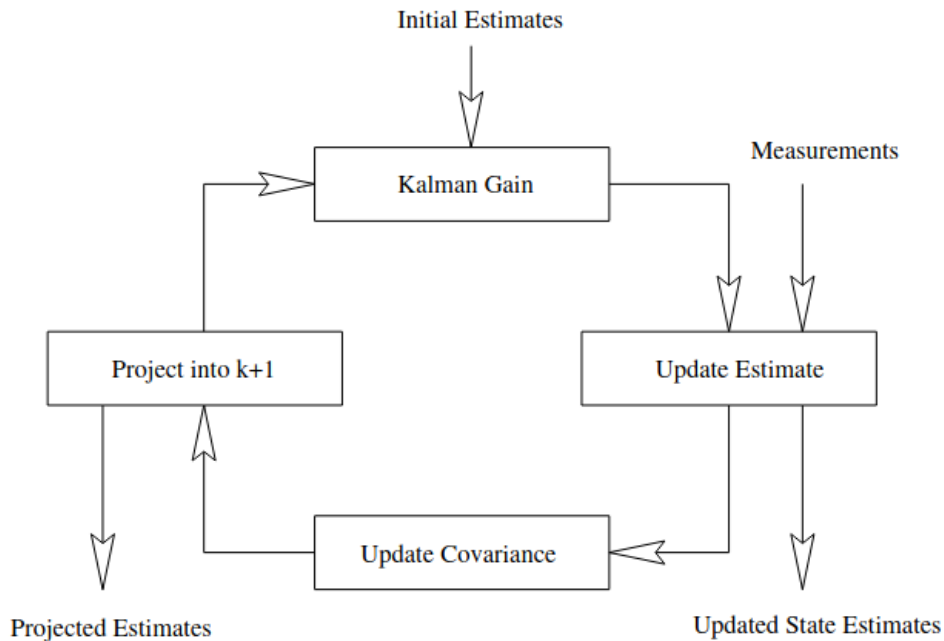
Έτσι λοιπόν, εκφράζοντας την εκ των υστέρων εκτίμηση \widehat{x}_k ως γραμμικό συνδυασμό της εκ των προτέρων εκτίμησης \widehat{x}_k^- και της σταθμισμένης διαφοράς της πραγματικής μέτρησης z_k από την αναμενόμενη μέτρηση \widehat{z}_k , δηλαδή ως:

$$\widehat{x}_k = \widehat{x}_k^- - K_k(z_k - H\widehat{x}_k^-) \quad (2.14)$$

Όπου ο όρος $(z_k - H\widehat{x}_k^-)$ αποτελεί το σφάλμα πρόβλεψης, ο βασικός στόχος του *Kalman* φίλτρου είναι ο υπολογισμός του πίνακα ώστε ο πίνακας P_k να μηδενιστεί. Ο πίνακας υπολογίζεται από τη σχέση:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (2.15)$$

Συγκεντρωντικά ο αλγόριθμος του *Kalman* φίλτρου φαίνεται στο παρακάτω σχήμα.2.6 και πίνακα 2.1:



Σχήμα 2.6: Διαδικασία Kalman Φίλτρου [5].

Description	Equation
Kalman Gain	$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$
Update Estimate	$\widehat{x}_k = \widehat{x}_k^- - K_k(z_k - H\widehat{x}_k^-)$
Update Covariance	$P_k = (I - K_k H) P_k^-$
Project into k + 1	$\widehat{x}_{k+1}^- = A_k \widehat{x}_k, P_{k+1} = A_k P_k A_k^T + Q$

Πίνακας 2.1: Διαδικασία και εξισώσεις Kalman Φίλτρου [5].

2.7.3 Εκτεταμένο Kalman Φίλτρο (Extended Kalman Filter-EKF)

Οι πολύ αυστηρές παραδοχές γραμμικότητας του Kalman φίλτρου όπως αναμένεται δεν μπορούν να ισχύουν σε κάθε περίπτωση, και αυτό διότι το περιβάλλον είναι μη γραμμικό και συνήθως η συμπεριφορά του θορύβου είναι μη γραμμική. Για να βελτιωθεί αυτό, το

εκτεταμένο Kalman φίλτρο (EKF) έχει διαμορφωθεί για να εισάγει ένα ακριβές μη γραμμικό σύστημα και μοντέλο παρατήρησης. Το EKF [53] λαμβάνει αυτό το μη γραμμικό μοντέλο και το γραμμικοποιεί γύρω από τον μέσο όρο της κατάστασης, χρησιμοποιώντας ανάπτυξη κατά Taylor. Αυτό σημαίνει ότι, αντί να μεταφέρει μια κατάσταση Gauss μέσω ενός σταθερού γραμμικού μοντέλου, το EKF στην πραγματικότητα υπολογίζει ένα νέο τοπικά γραμμικοποιημένο μοντέλο, για κάθε σημείο της διαδικασίας επανάληψης. Δεδομένου ότι το μοντέλο θα επεξεργάζεται εκτιμήσεις κατάστασης Gauss, η παραπάνω γραμμικοποίηση έχει ένα μειονέκτημα. Το μειονέκτημα είναι ότι γραμμικοποιώντας γύρω από το διάνυσμα κατάστασης, αμελούμε πληροφορία της τοπικής περιοχής γύρω από το σημείο, με συνέπεια να οδηγήσει το σύστημα σε μικρότερη ακρίβεια.

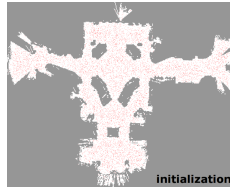
2.7.4 Μη παραμετρικό Φιλτράρισμα, Φίλτρο Σωματιδίων (Non-parametric filtering and Particle Filter-PF)

Μέχρι στιγμής, τα φίλτρα που έχουν εισαχθεί βασίζονται όλα σε κάποια μορφή Γκαουσιανής υπόθεσης. Μια εναλλακτική προσέγγιση στο φιλτράρισμα είναι και το μην παραμετρικό φιλτράρισμα ή φιλτράρισμα σωματιδίων. Η κύρια ιδέα του φίλτρου σωματιδίων είναι να εκτιμήσει τη μεταβλητή που μελετάμε με μια παραμετρική μην Γκαουσιανή πυκνότητα πιθανότητας, καθώς αυτή εξελίσσεται στο χρόνο. Δηλαδή, τα φίλτρα σωματιδίων παράγουν δείγματα για τον εκ των υστέρων χώρο πιθανοτήτων και εξάγουν μια διακριτή προσέγγιση επιλέγοντας και σταθμίζοντας σωματίδια για να προχωρήσουν στην επόμενη επανάληψη με μια διαδικασία Monte Carlo [54]. Έτσι λοιπόν, απαιτείται η προδιαγραφή μια συνάρτησης δειγματοληψίας, ώστε να λαμβάνονται τα δείγματα από το γενικό πληθυσμό, με σκοπό τη διακριτή προσέγγιση της εκ των υστέρων πυκνότητας πιθανότητας. Σημειώστε ότι, εφόσον κάνουμε δειγματοληψία απευθείας από την εκ των υστέρων πυκνότητα πιθανότητας, δεν περιορίζουμε να είναι μια κατανομή Gauss. Τέλος, ο αλγόριθμος είναι αναδρομικός και αποτελείται από το τμήμα της πρόβλεψης και αυτό της ανανέωσης, όπως φαίνεται και στο σχήμα.2.14.

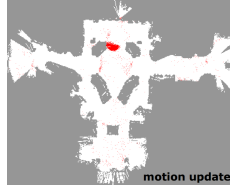
Ενώ είναι προφανές ότι μια τέτοια προσέγγιση βάσει δεδομένων προσδίδει ευελιξία, είναι άξιο να εξεταστεί γιατί η εγκατάλειψη των μοντέλων Gauss είναι εξαρχής επιθυμητή. Μια πτυχή του SLAM που οι KF δεν έχουν κανένα γενικό τρόπο αντιμετώπισης είναι η πολυτροπικότητα. Η πολυτροπικότητα προκύπτει όταν μια κατανομή έχει περισσότερα από ένα σημαντικά τοπικά μέγιστα, π.χ. στις κατανομές εντοπισμού, όταν ένα ρομπότ επισκέπτεται μια περιοχή που δεν φαίνεται να διακρίνεται από μια άλλη (τουλάχιστον εντός των ορίων σφάλματος των αισθητήρων του). Αυτή είναι μια ισχυρά μη-Γκαουσιανή ιδιότητα, η οποία δεν μπορεί να αντιμετωπιστεί αποκλειστικά με μαθηματικές τροποποιήσεις στο παράδειγμα Kalman, κάτι που είναι μέρος του λόγου για τον οποίο τα φίλτρα σωματιδίων θεωρούνται μια αποτελεσματική εναλλακτική λύση.

2.7.5 Γράφος Πόζας (Pose Graph)

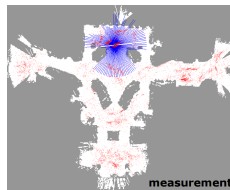
Όλες οι προσεγγίσεις φιλτραρίσματος που έχουμε δει έχουν έναν εγγενή περιορισμό, προσεγγίζουν το SLAM σε μεμονωμένες αυξήσεις, σύμφωνα με την υπόθεση Markov. Αυτό αποτυγχάνει να αντιμετωπίσει το πλήρες πρόβλημα SLAM, καθώς οι πιθανοτικές προσεγγί-



Σχήμα 2.7: Αρχικοποίηση Φίλτρου Σωματιδίων



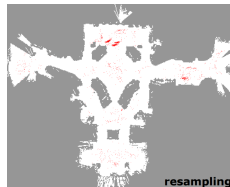
Σχήμα 2.8: Ανανέωση Φίλτρου Σωματιδίων



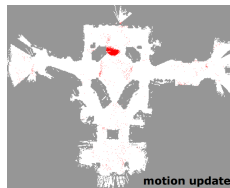
Σχήμα 2.9: Μέτρηση Φίλτρου Σωματιδίων



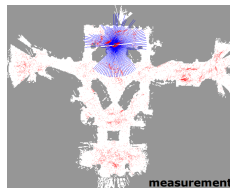
Σχήμα 2.10: Απόδοση βαρών Φίλτρου Σωματιδίων



Σχήμα 2.11: Δειγματοληψία Φίλτρου Σωματιδίων



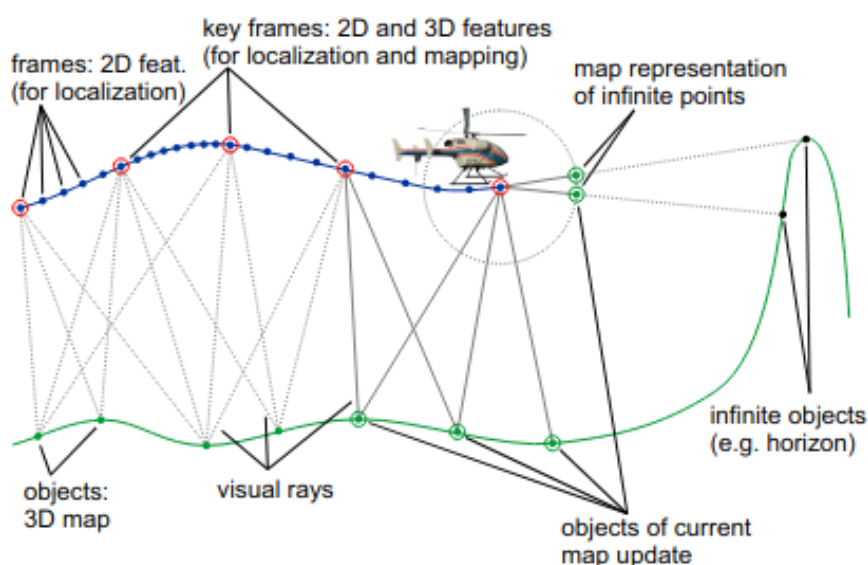
Σχήμα 2.12: Ανανέωση κίνησης Φίλτρου Σωματιδίων



Σχήμα 2.13: Μέτρηση Φίλτρου Σωματιδίων

Σχήμα 2.14: Διαδικασία Φίλτρου Σωματιδίων [6].

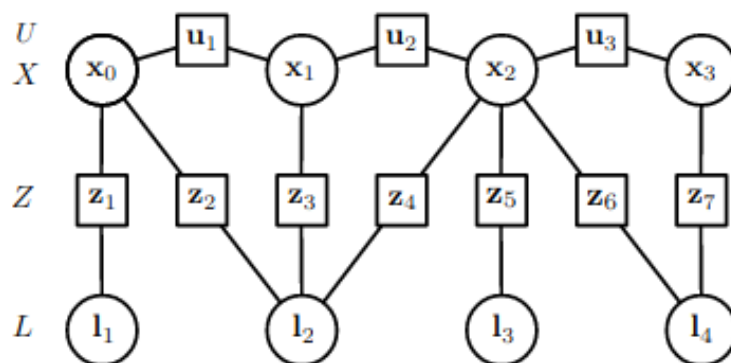
γίσεις που γίνονται εξαρτώνται μόνο από τρέχουσες μετρήσεις και ελέγχους. Στην πραγματικότητα, το πλήρες πρόβλημα SLAM είναι μια μεγιστοποίηση σε ολόκληρη τη σειρά των στάσεων, που εξετάζονται όλα ταυτόχρονα. Έτσι λοιπόν, το SLAM μπορεί να μοντελοποιηθεί σαν ένα γράφο (βλ. σχήμα.2.15) που οι κόμβοι του είναι οι πόζες του ρομπότ στον 3D χώρο και οι συνδέσεις μεταξύ τους αντιπροσωπεύουν τα κοινά χαρακτηριστικά που βλέπουν στο 3D χώρο και το βάρος αυτής της ακμή κωδικοποιεί το πλήθος των κοινών χαρακτηριστικών που έχουν ανιχνεύσει στο περιβάλλον. Αυτή η δομή μια αραιή αναπαράσταση της τροχιάς ενός ρομπότ και χρησιμοποιείται για την κωδικοποίηση όλων των πληροφοριών που είναι απαραίτητες ώστε ένα πλαίσιο βελτιστοποίησης να λειτουργεί με όλες τις παρατηρήσεις και να θέτει εκτιμήσεις ταυτόχρονα, με στόχο τη σύγκλιση σε μια παγκόσμια βέλτιστη λύση στο πρόβλημα SLAM. Επίσης, οι κόμβοι ενός γραφήματος πόζας χωρίζονται σε δύο τύπους: κόμβους στάσης και κόμβους παρατήρησης, ενώ οι ακμές αντιπροσωπεύουν μετασχηματισμούς μεταξύ δύο κόμβων που συνδέουν. Μια άκρη που συνδέει δύο κόμβους στάσης αντιστοιχεί σε ένα βήμα στην τροχιά του ρομπότ, ενώ μια άκρη μεταξύ ενός κόμβου στάσης και ενός κόμβου παρατήρησης κωδικοποιεί τον μετασχηματισμό του παρατηρούμενου χαρακτηριστικού στο πλαίσιο αναφοράς της στάσης.



Σχήμα 2.15: Αναπαράσταση SLAM προβλήματος, ως γράφο πόζας με κόμβους πόζας και κόμβους αντικειμένων χάρτη [7].

Μια λεπτομερειακή εικόνα του παραπάνω σχήματος, που έχει χρησιμοποιηθεί με επιτυχία στο [55] μεταξύ άλλων, είναι το γράφημα παραγόντων (factor graph) (βλ.σχήμα.2.16). Ένα γράφημα παραγόντων αντικαθιστά τις ακμές με παράγοντες, οι οποίοι συνδέουν πολλούς κόμβους ταυτόχρονα, αλλάζοντας τις μαθηματικές λεπτομέρειες της βελτιστοποίησης.

Γενικές μέθοδοι για τη βελτιστοποίηση γραφημάτων αυτού του τύπου έχουν ήδη αναπτυχθεί, όπως το [56], αλλά η εργασία μας βασίζεται στο [10], το οποίο χρησιμοποιείται επίσης από το πρότυπο ORB-SLAM2 [23].



Σχήμα 2.16: Λεπτομερής αναπαράσταση του γράφου πόζας, όπου U , L , X , Z έχουν οριστεί στην ενότητα 2.7.1 [8].

2.8 Επαναληπτική Μη γραμμική Βελτιστοποίηση (Iterative Non-linear Optimization)

Την τελευταία δεκαετία οι περισσότερες εφαρμογές SLAM, αξιοποιούν το γράφο πόζας με σκοπό να μοντελοποιήσουν την τροχιά σαν ένα γράφημα από πόζες και να τη βελτιστοποιήσουν χρησιμοποιώντας επαναληπτικούς αλγόριθμους μη γραμμικής βελτιστοποίησης (όπως στα [10], [23], [43]). Πρωτού χρησιμοποιηθεί ο γράφος πόζας θα πρέπει να ορίσουμε μια αντικειμενική συνάρτηση προς βελτιστοποίηση έστω $\mathbf{F}(\mathbf{x})$ όπου \mathbf{x} είναι το διάνυσμα κατάστασης που επιθυμούμε να βελτιστοποιήσουμε, ελαχιστοποιώντας την αντικειμενική συνάρτηση $\mathbf{F}(\mathbf{x})$. Με άλλα λόγια:

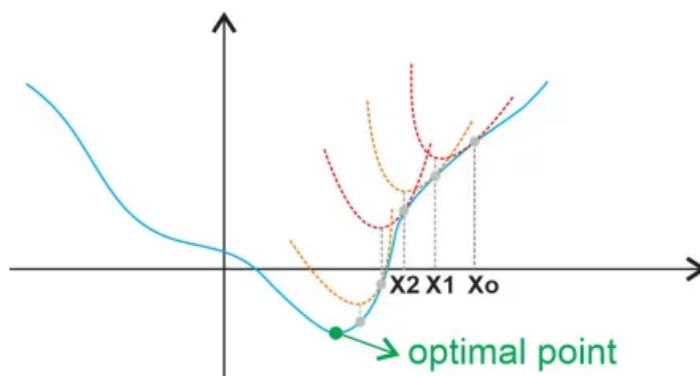
$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \quad (2.16)$$

Έχοντας ορίσει την αντικειμενική συνάρτηση προς βελτιστοποίησης, χρειαζόμαστε τώρα μια διαδικασία για την πραγματική εκτέλεση της βελτιστοποίησης. Εδώ γίνονται σημαντικές οι μη γραμμικές διαδικασίες βελτιστοποίησης ελαχίστων τετραγώνων. Η βασική ιδέα είναι να ελαχιστοποιηθεί αριθμητικά η αντικειμενική συνάρτηση, επαναλαμβάνοντας ορισμένα βήματα επαναληπτικά $\Delta\mathbf{x}$ ώστε η σειρά $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta\mathbf{x}$ να συγκλίνει στο \mathbf{x}^* . Πιο αναλυτικά:

- Προσεγγίση της $\mathbf{F}(\mathbf{x})$ γύρω από μια εκτίμηση κατάστασης $\widehat{\mathbf{x}}$, με μια αναλυτικά μορφή.
- Επίλυση για ένα καλό βήμα $\Delta\mathbf{x}$ σε αυτή τη φόρμα.
- Ενημέρωση της εκτίμησης κατάστασης με το υπολογισμένο βήμα $\mathbf{x}_{n+1} \leftarrow \mathbf{x}_n + \Delta\mathbf{x}$.
- Επανάληψη μέχρι τη σύγκλιση.

Γενική περίπτωση Βελτιστοποίησης στον Ευκλείδειο χώρο Gauss-Newton

Η Newton μέθοδος προσεγγίζει τη συνάρτηση κόστους σε κάθε επανάληψη με ένα παραβολοειδές. Στη συνέχεια, υπολογίζει ακριβώς το ελάχιστο αυτού του παραβολοειδούς και επαναλαμβάνει. Η διαδικασία σκιαγραφείται στο σχήμα. 2.17. Κάθε βήμα Newton $\Delta\mathbf{x}$ καθορίζεται ως εξής. Αρχικά, γράφουμε την τοπική παραβολική προσέγγιση του κόστους γύρω από την τρέχουσα εκτίμηση, $\widehat{\mathbf{x}} = \mathbf{x}_{n1}$, χρησιμοποιώντας την επέκταση Taylor 2ης τάξης:



Σχήμα 2.17: Gauss-Newton εξέλιξη αλγορίθμου προς σύγκλιση [9].

$$\mathbf{F}(\hat{\mathbf{x}} + \Delta \mathbf{x}) \approx \mathbf{F}(\hat{\mathbf{x}}) + \mathbf{J}_F \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{H}_F \Delta \mathbf{x} \quad (2.17)$$

Όπου \mathbf{J}_F και \mathbf{H}_F είναι αντίστοιχα η πρώτη και η δεύτερη παράγωγος αντίστοιχα της συνάρτησης \mathbf{F} .

$$\mathbf{J}_F \triangleq \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \quad (2.18)$$

$$\mathbf{H}_F \triangleq \frac{\partial^2 \mathbf{F}}{\partial \mathbf{x}^2} \quad (2.19)$$

Στη συνέχεια, το βέλτιστο βήμα $\Delta \mathbf{x}^*$ βρίσκεται διαφοροποιώντας την 2.17 και θέτοντας την με μηδέν, λαμβάνουμε:

$$\mathbf{J}_F^T + \mathbf{H}_F \Delta \mathbf{x}^* = 0 \quad (2.20)$$

που αποδίδει το βήμα Newton

$$\Delta \mathbf{x}_N^* = -\mathbf{H}_F^{-1} \mathbf{J}_F^T \quad (2.21)$$

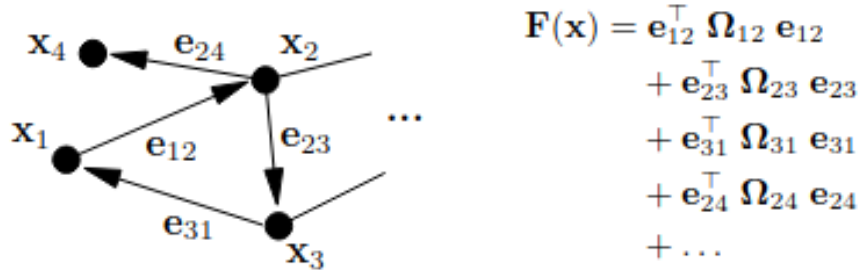
Η μέθοδος Newton συγκλίνει πολύ γρήγορα στη λύση εάν κάποιος δεν απέχει πολύ από αυτή τη λύση. Διαφορετικά, το βήμα Newton, όπως υπολογίζεται στο (2.21), έχει δύο σημαντικά μειονεκτήματα: το ένα είναι ότι το μήκος του βήματος μπορεί να είναι πολύ μεγάλο ώστε να ξεφεύγει από το ελάχιστο, έλκοντας έτσι την ακολουθία σε ένα δευτερεύον τοπικό ελάχιστο. Αυτό συμβαίνει όταν το Hessian \mathbf{H}_F είναι πολύ μικρό, το οποίο αντιστοιχεί σε περιοχές με χαμηλή καμπυλότητα του κόστους $\mathbf{F}(\mathbf{x})$. Το άλλο μειονέκτημα είναι ότι, σε κοίλες ζώνες, δηλαδή όταν το \mathbf{H}_F είναι αρνητικό, το βήμα φεύγει προς την αντίθετη κατεύθυνση, ξεφεύγοντας έτσι από το ελάχιστο και αυξάνοντας το κόστος. Στην πραγματικότητα, επιβάλλοντας μηδενικά παράγωγα του κόστους, η μέθοδος Newton μπορεί να συγκλίνει τέλεια στο μέγιστο (αν και αυτό είναι πράγματι μια ασυνήθιστη κατάσταση, καθώς τα τοπικά μέγιστα είναι απίθανα). Και οι δύο καταστάσεις μπορούν να συνδυαστούν καταστροφικά.

Περίπτωση των ελαχίστων τετραγώνων και η μέθοδος Gauss-Newton

Σε πολλές περιπτώσεις, η συνάρτηση κόστους εκφράζεται ως η συνάρτηση τετραγώνων των σφαλμάτων (2.18):

$$\mathbf{F}(\mathbf{x}) = \frac{1}{2} \mathbf{e}(\mathbf{x})^T \Omega \mathbf{e}(\mathbf{x}) \quad (2.22)$$

όπου στην περίπτωση που χρησιμοποιούμε ένα γράφο πόζας προς βελτιστοποίηση η συνάρτηση κόστους αποτελείται από πολλούς όρους τετραγωνικών σφαλμάτων, δηλαδή:



Σχήμα 2.18: Αναπαράσταση της αντικειμενικής συνάρτησης ως γράφος σχέσεων [10].

Όπου ο πίνακας $\boldsymbol{\Omega}$ είναι ο αντίστροφος πίνακας συνδιακύμανσης των μετρήσεων του ρομπότ από το περιβάλλον του και είναι συμμετρικός και ημι-θετικός. Στη συνέχεια θα εισάγουμε ορισμούς και μεθοδολογίες που λύνουν το πρόβλημα 2.22, ωστόσο η 2.30 λύνεται με τον ίδιους αλγορίθμους που θα παραθέσουμε παρακάτω απλά δημιουργώντας αντίστοιχους πίνακες που περικλείουν όλους τους όρους $e_k(\mathbf{x})$. Έτσι λοιπόν έχουμε:

$$\mathbf{F}(\widehat{\mathbf{x}} + \Delta\mathbf{x}) \approx \frac{1}{2} e^T \boldsymbol{\Omega} e + e^T \boldsymbol{\Omega} \mathbf{J} \Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}^T (\mathbf{J}^T \boldsymbol{\Omega} \mathbf{J} + e^T \boldsymbol{\Omega} \mathbf{H}) \Delta\mathbf{x} \quad (2.23)$$

όπου

$$\mathbf{J} \triangleq \frac{\partial e}{\partial \mathbf{x}} \quad (2.24)$$

$$\mathbf{H} \triangleq \frac{\partial^2 e}{\partial \mathbf{x}^2} \quad (2.25)$$

Το βήμα Gauss-Newton είναι μια προσέγγιση του βήματος του Newton που συνίσταται στην παραμέληση του δεύτερου όρου, $e^T \boldsymbol{\Omega} \mathbf{H}$. Αυτό είναι ενδιαφέρον, γιατί μας απαλλάσσει από τον υπολογισμό και τον χειρισμό του \mathbf{H} . Είναι επίσης σχετικό, επειδή ο όρος $e^T \boldsymbol{\Omega} \mathbf{H}$ είναι στην πλειονότητα των περιπτώσεων διπλά μικρός, καθώς το e είναι μικρό και το ίδιο είναι και το \mathbf{H} . Έτσι λοιπόν καταλήγουμε:

$$\mathbf{F}(\widehat{\mathbf{x}} + \Delta\mathbf{x}) \approx \frac{1}{2} e^T \boldsymbol{\Omega} e + e^T \boldsymbol{\Omega} \mathbf{J} \Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}^T \mathbf{J}^T \boldsymbol{\Omega} \mathbf{J} \Delta\mathbf{x} \quad (2.26)$$

Αυτή η έκφραση συνήθως λαμβάνεται απευθείας, γράφοντας τη γραμμική προσέγγιση του σφάλματος, $e(\widehat{\mathbf{x}} + \Delta\mathbf{x}) \approx e(\mathbf{x}) + \mathbf{J} \Delta\mathbf{x}$, και αντικαθιστώντας στο (2.22). Κατόπιν, ορίζοντας το κατά προσέγγιση το καινούριο \mathbf{H} :

$$\mathbf{H} \triangleq \mathbf{J}^T \boldsymbol{\Omega} \mathbf{J} \quad (2.27)$$

και το Gauss-Newton βήμα γίνεται:

$$\Delta\mathbf{x}_{GN}^* = \mathbf{H} \mathbf{J}^T \boldsymbol{\Omega} e \quad (2.28)$$

Η μέθοδος Gauss-Newton πάσχει από τα ίδια μειονεκτήματα με τη μέθοδο Newton, με την εξαίρεση ότι δεν μπορεί να συγκλίνει στο μέγιστο, επειδή η τροποποιημένη \mathbf{H} είναι θετική

από την κατασκευή.

Βελτίωση της σύγκλισης με τον αλγόριθμο Levenberg-Marquardt

Αποτελεί έναν υβριδικό αλγόριθμο, μεταξύ του Gauss-Newton και του Gradient Descent εισάγοντας στο βήμα $\Delta \mathbf{x}$ του Gauss-Newton αλγορίθμου δύο παραμέτρους τη λ και την α , δηλαδή:

$$\Delta \mathbf{x}_{LM}^* = -\alpha(\mathbf{H} + \lambda \cdot \text{diag}(\mathbf{H}))^{-1} \mathbf{J}^T \Omega e \quad (2.29)$$

Όπου $\text{diag}(\cdot)$ είναι η διαγώνιος του πίνακα και οι παράμετροι α , λ προσαρμόζονται σε κάθε επανάληψη και έτσι για μεγάλες μεταβολές του *Hessian* πίνακα, η λ λαμβάνει μικρή τιμή, προσεγγίζοντας τον Gauss-Newton αλγόριθμο διαφορετικά για μικρές μεταβολές του *Hessian* πίνακα διαλέγουμε μεγάλη τιμή της παραμέτρου λ και άρα προσεγγίζουμε τη Gradient-Descent μέθοδο. Η μέθοδος αυτή, με καλές αρχικές συνθήκες βρίσκει το τοπικό ελάχιστο και είναι πιο εύρωστη από τη Gauss-Newton μέθοδο, αλλά λίγο πιο αργή από την τελευταία.

Όλα τα παραπάνω έγιναν προς την κατεύθυνση που η εξ.2.22 ισχύει για έναν κόμβο του γράφου. Στην περίπτωση του SLAM προβλήματος, η εξ.2.22 μετατρέπεται στην παρακάτω σχέση:

$$\mathbf{F}(\mathbf{x}) = \sum_{k=1}^N e_k(\mathbf{x})^T \Omega_k e_k(\mathbf{x}) \quad (2.30)$$

Συμπεριλαμβάνοντας τώρα όλους τους κόμβους του γράφου (βλ.σχήμα.2.18). Αυτή η εξίσωση είναι ισοδύναμη με την εξ.2.22, όπου με e να είναι ο πίνακας:

$$e = \begin{bmatrix} e_1(\mathbf{x}) \\ e_2(\mathbf{x}) \\ \cdot \\ \cdot \\ \cdot \\ e_N(\mathbf{x}) \end{bmatrix} \quad (2.31)$$

και

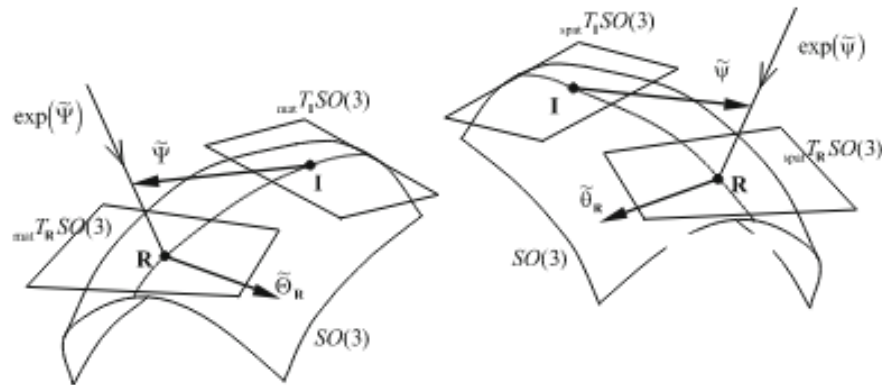
$$\Omega = \begin{bmatrix} \Omega_1 & & & \\ & \Omega_2 & & \\ & & \cdot & \\ & & & \Omega_N \end{bmatrix} \quad (2.32)$$

Καταλήγουμε ότι οι αλγόριθμοι προς βελτιστοποίηση παραμένουν ίδιοι απλά, η συνάρτηση προς βελτιστοποίηση είναι πολυδιάστατη.

2.8.1 Προσέγγιση στο Manifold χώρο

Μας ενδιαφέρει να λύσουμε το πρόβλημα βελτιστοποίησης σε περιπτώσεις όπου η κατάσταση (ή ένα μέρος της) δεν αναπαρίσταται στον Ευκλείδειο χώρο. Για παράδειγμα, οι προσανατολισμοί στις 3D δεν είναι Ευκλείδειοι, καθώς χρησιμοποιήσαμε πρόσθεση διανύσματος για διαταραχές καταστάσεων, χωρίς να λάβουμε υπόψη ότι ανήκουν στο χώρο $SO(3)$ (*Special Orthogonal Group*). Πριν λύσουμε το πρόβλημα αυτό, ας εξετάσουμε πρώτα έναν χώρο που

είναι μόνο τοπικά ευκλείδειος, αλλά όχι απαραίτητα σφαιρικά ευκλείδειος (Σχ.2.19) και ως εισάγουμε κάποιους ορισμούς σχετικά με την *Lie Theory* και τα *Lie Group* και *Lie Algebra*.



Σχήμα 2.19: Αντιστοιχισμός μεταξύ Ευκλείδειου χώρου και εφαπτομενικού [11].

Ορισμός: Αρχικά ένα *Lie Group* αναφέρεται σε γκρουπ με συνεχείς (ομαλές) ιδιότητες και είναι ένα διαφοροποιήσιμο *manifold*.

Εάν ο αναγνώστης ενδιαφέρεται για τη θεωρητική φύση των *Lie Groups*, παρακαλούμε ανατρέξτε στο [57].

Ορισμός: Κάθε *Lie Group* έχει μια *Lie Algebra* που της αντιστοιχεί. Η *Lie Algebra* περιγράφει την τοπική δομή του *Lie Group* γύρω από το σημείο προέλευσής της. Με άλλα λόγια, είναι ο διανυσματικός χώρος (εφαπτομενικός χώρος), \mathfrak{g} πάνω σε ένα group G εφοδιασμένο με έναν δυαδικό τελεστή $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$ γνωστά και ως *Lie brackets*. Αυτή η άλγεβρα (*Lie Algebra*) πρέπει να ικανοποιεί τα ακόλουθα:

- **Κλειστότητα:** $\forall X, Y \in \mathfrak{g}, [X, Y] \in \mathfrak{g}$.
- **Διγραμμικότητα:** $\forall X, Y, Z \in \mathfrak{g}, \forall a, b, c \in G : [aX + bY, Z] = a[X, Z] + b[Y, Z]$.
- **Αντανακλαστικότητα:** $\forall X \in \mathfrak{g}, [X, X] = 0$.
- **Jacobi ταυτότητα:** $\forall X, Y, Z \in \mathfrak{g} : [X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0$.

Μερικά πολύ σημαντικά παραδείγματα των παραπάνω ορισμών αφορούν ομάδες που έχουμε ήδη συναντήσει. Για παράδειγμα, η ομάδα 3D περιστροφών, $\text{mathcal{SO}(3)}$, είναι στην πραγματικότητα ένα *Lie Group*, με τη συσχετισμένη *Lie Algebra* $\text{so}(3)$. Οι μετασχηματισμοί σχηματίζουν επίσης ένα *Lie Group*, γνωστή ως $\text{SE}(3)$, με την *Lie Algebra* να συμβολίζεται με $\text{se}(3)$. Για παράδειγμα, στην περίπτωση του $\text{so}(3)$, η βάση σχηματίζεται από τις ακόλουθες τρεις βάσεις:

$$G1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, G2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, G3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.33)$$

Θεωρώντας αναπαράσταση για τον πίνακα περιστροφής, άξονα-γωνίας (*axis - angle*) $\omega =$

$[\omega_1, \omega_2, \omega_3]$ έχουμε το αντίστοιχο *skew – symmetric* $so(3)$ πίνακα :

$$[\omega]_x = \sum_{i=1}^3 \omega_i G_i \in so(3) \quad (2.34)$$

Τότε η εκθετική συνάρτηση είναι το εργαλείο για την αντιστοίχιση των στοιχείων από τον εφαπτομενικό χώρο (την άλγεβρα *Lie*) στη *Lie Group*, για παράδειγμα :

$$\exp : so(3) \mapsto SO(3), \omega \mapsto R_{3 \times 3} \quad (2.35)$$

έχει την κλειστή μορφή χρησιμοποιώντας τον *Rodrigues* τύπο :

$$\exp([\omega]_x) = I_3 + \frac{\sin \vartheta}{\vartheta} [\omega]_x + \frac{1 - \cos \vartheta}{\vartheta^2} [\omega]_x^2 \quad (2.36)$$

Όπου $\vartheta = \|\omega\|$. Η λογαριθμική συνάρτηση που είναι το αντίστροφο της εκθετικής. Αντιστοιχίζει τον πίνακα περιστροφής R στο εφαπτομενικό διάνυσμα ω :

$$\vartheta = \arccos \frac{\text{tr}(\mathbf{R}) - 1}{2} \quad (2.37)$$

$$\log(\mathbf{R}) = \frac{\vartheta}{2 \sin \vartheta} (\mathbf{R} - \mathbf{R}^T) \quad (2.38)$$

$$\log : SO(3) \mapsto \mathfrak{K}^3 \quad (2.39)$$

Όπου $\text{tr}(\cdot)$ είναι το ίχνος του πίνακα. Η σχέση μεταξύ των δύο χώρων ($so(3), SO(3)$) φαίνεται και στη βλ. σχήμα. 2.19. Ο λόγος που καταλήγουμε στη βελτιστοποίηση στον εφαπτομενικό χώρο είναι ότι, στη βελτιστοποίηση, θέλουμε τα σφάλματα μας $\Delta \mathbf{x}$ να έχουν την ελάχιστη δυνατή διάσταση, χωρίς να χάνεται ακρίβεια στη λύση και χωρίς μεγάλο υπολογιστικό κόστος. Επίσης, δεν έχει ληφθεί υπόψιν η μην Ευκλείδεια δομή του χώρου $SO(3)$. Έτσι, λοιπόν ορίζουμε το διάνυσμα σφάλματος $\Delta \mathbf{x}$ σε έναν Ευκλείδειο χώρο που εφάπτεται στην πολλαπλότητα στο σημείο που ορίζεται από την εκτίμηση της τρέχουσας κατάστασης. Αυτό το σφάλμα είναι ελάχιστης διάστασης. Παράλληλα με το ελάχιστο σφάλμα, ορίζουμε επίσης έναν τελεστή σύνθεσης \oplus που αντιστοιχίζει παραλλαγές στον Ευκλείδειο χώρο σε μια τοπική παραλλαγή στην πολλαπλότητα, $\Delta \mathbf{x} \rightarrow \widehat{\mathbf{x}} \oplus \Delta \mathbf{x}$, έτσι ώστε :

$$\mathbf{x} = \widehat{\mathbf{x}} \oplus \Delta \mathbf{x} \quad (2.40)$$

Η ελαχιστοποίηση *Gauss – Newton* του 2.30 εκτελείται γραμμικοποιώντας τα σφάλματα e_k , λύνοντας ένα γραμμικό πρόβλημα ελαχίστων τετραγώνων και επαναλαμβάνοντας μέχρι τη σύγκλιση, δηλαδή :

$$e_k(\widehat{\mathbf{x}} + \Delta \mathbf{x}) = \widehat{e_k(\mathbf{x})} + \mathbf{J}_k \Delta \mathbf{x} \quad (2.41)$$

Αυτό επηρεάζει τη γραμμικοποίηση των σφαλμάτων 2.41, η οποία γίνεται :

$$e_k(\mathbf{x}) = e_k(\widehat{\mathbf{x}} \oplus \Delta \mathbf{x}) \approx \widehat{e_k} + \mathbf{J}'_k \Delta \mathbf{x} \quad (2.42)$$

που το \mathbf{J}'_k είναι ο καινούριος Ιακωβιανός πίνακας σε σχέση με τη κατάσταση σφάλματος:

$$\mathbf{J}'_k \triangleq \left. \frac{\partial e_k(\mathbf{x})}{\partial \Delta \mathbf{x}} \right|_{\widehat{\mathbf{x}}} = [\cdots \mathbf{J}'_{ki} \cdots \mathbf{J}'_{kj} \cdots] \quad (2.43)$$

όπου

$$\mathbf{J}'_{ki} = \left. \frac{\partial e_k(\mathbf{x})}{\partial \Delta \mathbf{x}_i} \right|_{\widehat{\mathbf{x}}}, \mathbf{J}'_{kj} = \left. \frac{\partial e_k(\mathbf{x})}{\partial \Delta \mathbf{x}_j} \right|_{\widehat{\mathbf{x}}} \quad (2.44)$$

Εφαρμόζοντας τον κανόνα αλυσίδας έχουμε:

$$\mathbf{J}'_{ki} = \left. \frac{\partial e_k(\mathbf{x})}{\partial \Delta \mathbf{x}_i} \right|_{\widehat{\mathbf{x}}} = \left. \frac{\partial e_k(\mathbf{x})}{\partial \mathbf{x}_i} \right|_{\widehat{\mathbf{x}}} \left. \frac{\partial \mathbf{x}_i}{\partial \Delta \mathbf{x}_i} \right|_{\widehat{\mathbf{x}}, \Delta \mathbf{x}_i=0} = \mathbf{J}_{ki} \mathbf{M}_i \quad (2.45)$$

Με αντίστοιχη λογική και το \mathbf{J}_{kj} . Οι δεύτεροι όροι είναι αυτοί που μας ενδιαφέρουν εδώ, οι παράγωγοι του τελεστή \oplus σε σχέση με την κατάσταση σφάλματος κάθε μπλοκ.

$$\mathbf{M}_i = \left. \frac{\partial \widehat{\mathbf{x}}_i \oplus \Delta \mathbf{x}_i}{\partial \Delta \mathbf{x}_i} \right|_{\widehat{\mathbf{x}}_i, \Delta \mathbf{x}_i=0} \quad (2.46)$$

Και καταλήγουμε στο βήμα της ανανέωσης που είναι $\widehat{\mathbf{x}} \leftarrow \widehat{\mathbf{x}} \oplus \Delta \mathbf{x}^*$.

Κεφάλαιο 3

3Δ Ευθείες και Σημεία Φυγής

Τα τελευταία χρόνια πολλοί ερευνητές που ασχολούνται με το πρόβλημα του *Visual SLAM* έχουν στρέψει την προσοχή τους σε ανάπτυξη αλγορίθμων που αντλούν περισσότερη πληροφορία από το περιβάλλον τους για να λύσουν το πρόβλημα του *Visual SLAM*. Η πληροφορία αυτή, αφορά την ανίχνευση περισσότερων γεωμετρικών σχημάτων όπως ευθειών και σημείων φυγής και που έχουν χρησιμοποιηθεί επιτυχώς και αναλύονται λεπτομερώς στα [15], [44], [24], [45] και αποτελούν σημαντικό τμήμα στη δουλειά μας.

Πολλά περιβάλλοντα όπως εσωτερικοί χώροι σπιτιών, γραφεία ή διάδρομοι προσφέρουν σε ένα ρομπότι περισσότερα γεωμετρικά σχήματα από ότι σημεία και γωνίες, όπως ευθείες και σημεία φυγής, οδηγώντας έτσι σε μια καλύτερη εκτίμησης της θέσης του ρομπότι και πιο πληθωρικής αναπαράστασης του χάρτη που αντιλαμβάνονται. Έτσι λοιπόν, ας ορίσουμε τους μαθηματικούς φορμαλισμούς για τις ευθείες και τα σημεία φυγής τόσο στο 3Δ χώρο όσο και στο χώρο της εικόνας.

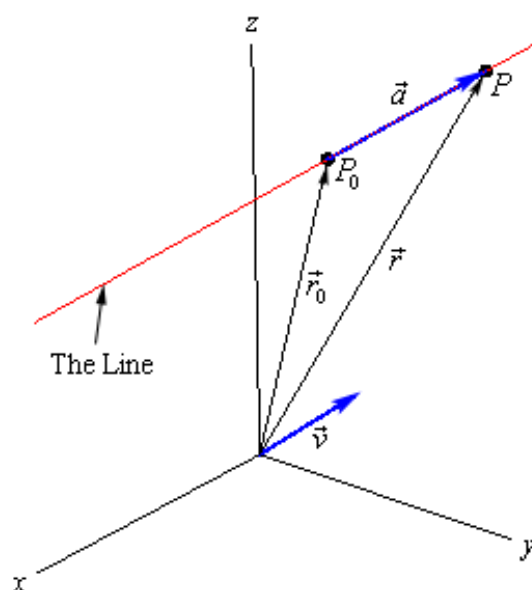
3.1 Μαθηματική Περιγραφή ευθειών

Στο κεφάλαιο αυτό θα παρουσιάσουμε το μαθηματικό υπόβαθρο για τις ευθείες και τα σημεία φυγής.

Ορισμός (3Δ Ευθείας): Έστω L μια ευθεία στο χώρο \mathcal{R}^3 (εικ.3.1) και έστω $\mathbf{P}_o(x_o, y_o, z_o)$ και $\mathbf{P}(x, y, z)$ δύο σημεία πάνω στην ευθεία με $\vec{\mathbf{r}}_o$ και $\vec{\mathbf{r}}$ τα αντίστοιχα διανύσματα τους από την αρχή των αξόνων και έστω $\vec{\mathbf{v}} = (a, b, c)$ ένα οποιοδήποτε διάνυσμα παράλληλο στην ευθεία L . Τότε η διανυσματική μορφή της ευθείας στο 3Δ χώρο θα δίνεται από τον τύπο:

$$\vec{\mathbf{r}} = \vec{\mathbf{r}}_o + \hat{\lambda} \vec{\mathbf{v}}, -\infty < \hat{\lambda} < \infty. \quad (3.1)$$

Ένα μειονέκτημα της ευθείας μέσω των τελικών σημείων είναι ότι έχουν πάρα πολλούς βαθμούς ελευθερίας, δηλαδή έξι (τρεις για κάθε τελικό σημείο) αντί για τις τέσσερις που έχει μια 3Δ ευθεία. Για να ξεπεραστεί αυτό το πρόβλημα έχουν προταθεί στη βιβλιογραφία [15] διάφοροι τρόποι αναπαράστασης 3Δ ευθειών, οι οποίες διαφέρουν ως προς τη μαθηματική πολυπλοκότητά τους. Στην παρούσα εργασία θα αναφερθούμε σε μια. Σε κάθε περίπτωση η αναπαράσταση παρέχει μηχανισμούς για τον καθορισμό μιας ευθείας από: την ένωση δύο σημείων, από την τομή δύο επιπέδων, καθώς και ενός μηχανισμού αντιστοίχιση μεταξύ των δύο ορισμών. Οι αναπαραστάσεις επιτρέπουν επίσης τον υπολογισμό της τομής μεταξύ



Σχήμα 3.1: 3D ευθεία [12].

ευθείας και επιπέδου.

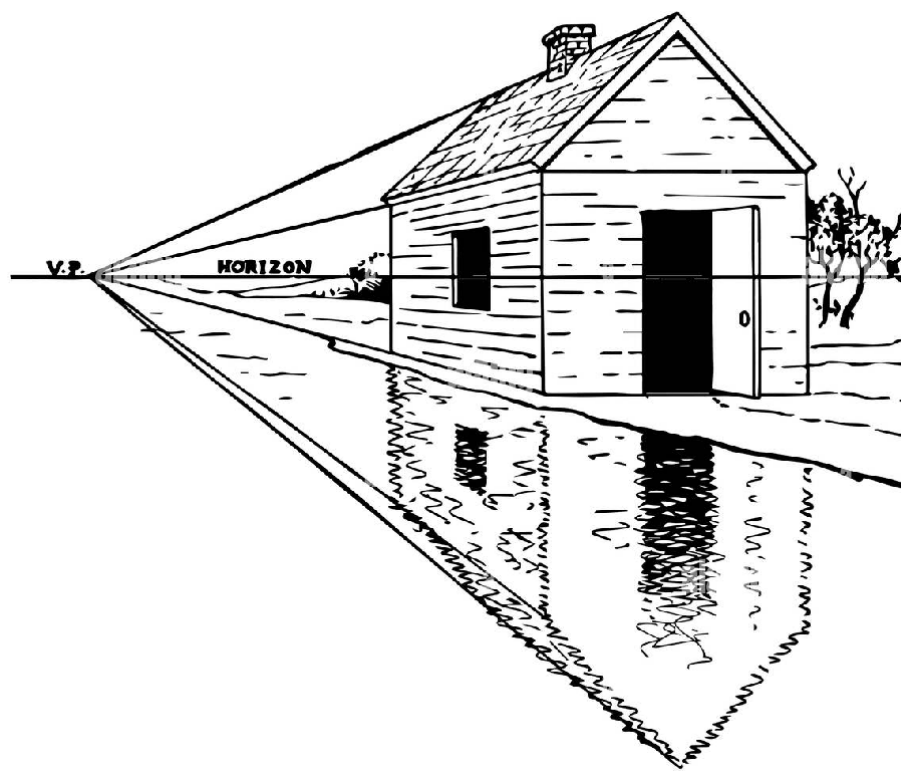
3.1.1 Plücker Πίνακας

Μια 3D ευθεία αντιπροσωπεύεται από έναν 4×4 *skew – symmetric* ομοιογενή πίνακα. Συγκεκριμένα, η ευθεία που ενώνει τα δύο σημεία A, B παριστάνεται από τον πίνακα L με στοιχεία [15]:

$$L = AB^T - BA^T \quad (3.2)$$

Αυτή η αναπαράσταση έχει 4 βαθμούς ελευθερίας, καθώς ο πίνακας L είναι ομογενής και ικανοποιεί τη συνθήκη $\det(L) = 0$, που οδηγεί σε τετραγωνικό περιορισμό στις συντεταγμένες Plücker. Οι ορισμοί που δόθηκαν χρησιμοποιούνται κυρίως κατά τη διαδικασία της ανακατασκευής και βελτιστοποίησης της πόζας [44],[24], [45] καθώς σε ένα πρόβλημα SLAM η αρχική πληροφορία που διαθέτουμε είναι η εικόνα. Ωστόσο, για να μπορέσουμε να χρησιμοποιήσουμε αυτές τις αναπαραστάσεις, πρέπει πρώτα αν ανιχνεύσουμε τις ευθείες στην εικόνα, αφού αυτή είναι η πληροφορία που έχουμε. Και έπειτα, υπολογίζοντας την πόζα και μπορούμε να χρησιμοποιήσουμε αυτές τις μοντελοποιήσεις για να ανακατασκευάσουμε το χώρο.

Η ανίχνευση των ευθειών από μια εικόνα, δεν είναι καινούρια ιδέα και στη βιβλιογραφία έχουν προταθεί αρκετοί αλγόριθμοι που τις ανιχνεύουν. Μερικοί από αυτούς είναι ο LSD [58], ο EDLines [59] και ο SOLD²[60]. Ο LSD [58] είναι ένας πολύ καλό αλγόριθμος ανίχνευσης ευθειών από εικόνες, με ακρίβεια υπό-pixel στην ανίχνευση του και με χρόνο εκτέλεσης $O(n)$. Απεναντίας, ο EDLines [59] έχει τον ίδιο χρόνο εκτέλεσης με τον LSD [58] αλλά λιγότερη ακρίβεια από τον πρώτο. Ενώ, ο SOLD²[60] αποτελεί μια καινούρια μορφή ανιχνευτή ευθειών που βασίζεται σε νευρωνικό δίκτυο και δεν έχει χρησιμοποιηθεί πρόσφατα σε κάποια εφαρμογή SLAM. Έτσι, έχοντας ανιχνεύσει τις ευθείες στην εικόνα μπορούμε να



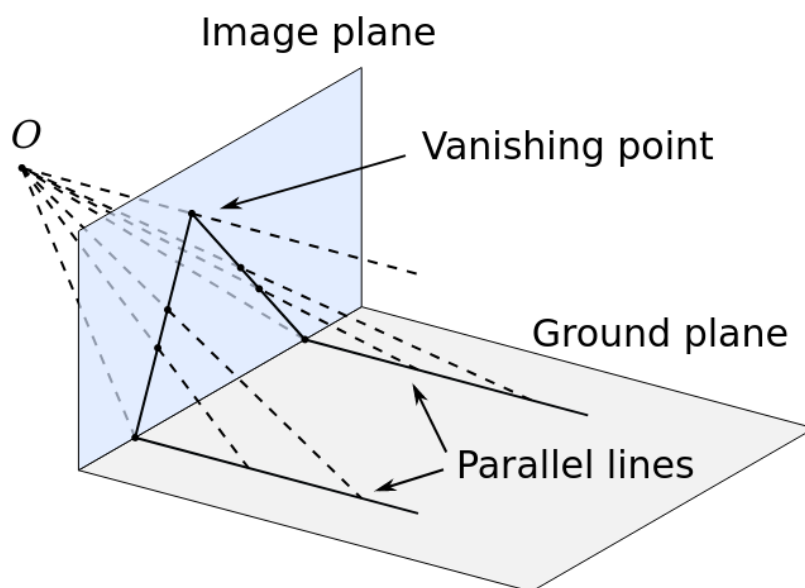
Σχήμα 3.2: Σημείο Φυγής [13].

τις χρησιμοποιήσουμε για να ανιχνεύσουμε τα σημεία φυγής. Ωστόσο, ας παρουσιάσουμε τη μαθηματική μοντελοποίηση αυτών και με ποιους τρόπους μπορούμε να τα ανιχνεύσουμε.

3.2 Σημεία Φυγής

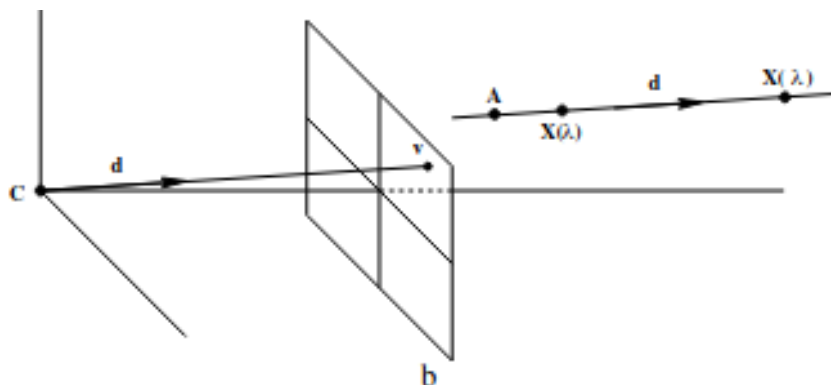
Ένα από τα πιο ενδιαφέροντα και οικεία φαινόμενα στην προοπτική γεωμετρία είναι ότι οι παράλληλες γραμμές στον τρισδιάστατο κόσμο προβάλλονται συνήθως σε μη παράλληλες γραμμές στην εικόνα και συγκεκριμένα αυτές οι προβαλλόμενες γραμμές τέμνονται σε ένα σημείο που ονομάζεται σημείο φυγής σχήμα.3.3. Οι παράλληλες γραμμές είναι αρκετά κοινές σε εσωτερικούς χώρους σπιτιών, σε γραφεία, διαδρόμους σε ουρανοξύστες ακόμα και σε τοίχους και δάπεδα. Τα όρια των δαπέδων, των οροφών και των θυρών συνήθως ευθυγραμμίζονται με ένα φυσικό σύστημα ορθογώνιων συντεταγμένων XYZ και λέγεται ότι τέτοιες σκηνές ακολουθούν έναν *Manhattan* κόσμο [61]. Υπάρχουν τυπικά πολλές άλλες γραμμές στη σκηνή που είναι παράλληλες σε κάθε έναν από αυτούς τους άξονες επίσης. Για παράδειγμα, έπιπλα όπως γραφεία και ράφια έχουν αυτού του είδους τη γεωμετρία κύβου και συχνά τοποθετούνται έτσι ώστε το σύστημα συντεταγμένων τους να είναι παράλληλο με το σύστημα συντεταγμένων της σκηνής. Αυτό σημαίνει ότι υπάρχουν ομάδες γραμμών και ακμών στις εικόνες που είναι οι προβολές παράλληλων γραμμών στον κόσμο και οι γραμμές σε κάθε ομάδα δείχνουν όλες σε ένα σημείο φυγής.

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τους τρόπους που μπορούμε να ανιχνεύσουμε τα σημεία φυγής σε μια εικόνα. Είναι προφανές ότι γεωμετρικά το σημείο φυγής μιας ευθείας λαμβάνεται τέμνοντας το επίπεδο της εικόνας με μια ακτίνα παράλληλη προς την



Σχήμα 3.3: Δημιουργία σημείου φυγής [14].

παγκόσμια γραμμή και περνώντας από το κέντρο της κάμερας. Έτσι, ένα σημείο φυγής εξαρτάται μόνο από την κατεύθυνση μιας γραμμής, όχι από τη θέση της (βλ.σχήμα.3.4) [15]. Αλγεβρικά το σημείο φυγής μπορεί να ληφθεί ως εξής: Από τον ορισμό της ευθείας



Σχήμα 3.4: Το σημείο φυγής, v , μιας ευθείας με κατεύθυνση d είναι η τομή του επιπέδου της εικόνας με μια ακτίνα παράλληλη στο d έως το C [15].

ένα οποιοδήποτε σημείο $\mathbf{X}(\lambda)$ πάνω σε αυτή μπορεί να εκφραστεί ως $\mathbf{X}(\lambda) = \mathbf{P}_o + \lambda D$ [15] όπου \mathbf{P}_o είναι ένα σημείο της ευθείας και $D = (d^T, 0)^T$ είναι η κατεύθυνση της ευθείας σε ομογενείς συντεταγμένες. Καθώς η παράμετρος λ τείνει στο άπειρο το σημείο $\mathbf{X}(\lambda)$ θα τείνει στο σημείο D στο άπειρο. Χρησιμοποιώντας τον πίνακα προβολής $P = K[I|0]$ το σημείο $\mathbf{X}(\lambda)$ προβάλλεται στην εικόνα στο σημείο [15]:

$$\mathbf{x}(\lambda) = P\mathbf{X}(\lambda) = PA + \lambda PD = a + \lambda Kd \quad (3.3)$$

Όπου το a είναι η προβολή του σημείου A στο επίπεδο της εικόνας. Έτσι λοιπόν το σημείο

φυγής [15] λαμβάνεται ως:

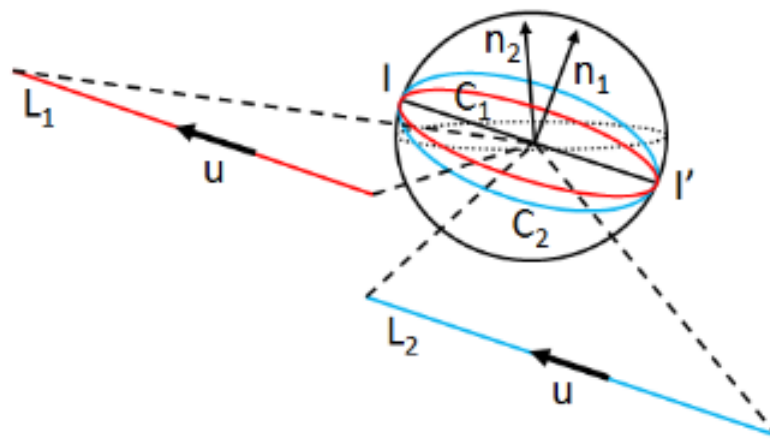
$$\mathbf{v} = \lim_{\beta \rightarrow \infty} \mathbf{x}(\beta) = \lim_{\beta \rightarrow \infty} (a + \beta K \mathbf{d}) = K \mathbf{d} \quad (3.4)$$

καταλήγωντας έτσι στο συμπέρασμα ότι το σημείο φυγής εξαρτάται από την κατεύθυνση της ευθείας. Επίσης, ένα σύνολο από 3D παράλληλες ευθείες αντιστοιχούν στο ίδιο σημείο φυγής. Τέλος, παρατηρούμε από την εξ.3.4 ότι η κατεύθυνση του σημείου φυγής δίνεται από την εξίσωση:

$$\mathbf{d} = K^{-1} \mathbf{v} \quad (3.5)$$

3.2.1 Εκτίμηση Σημείων Φυγής

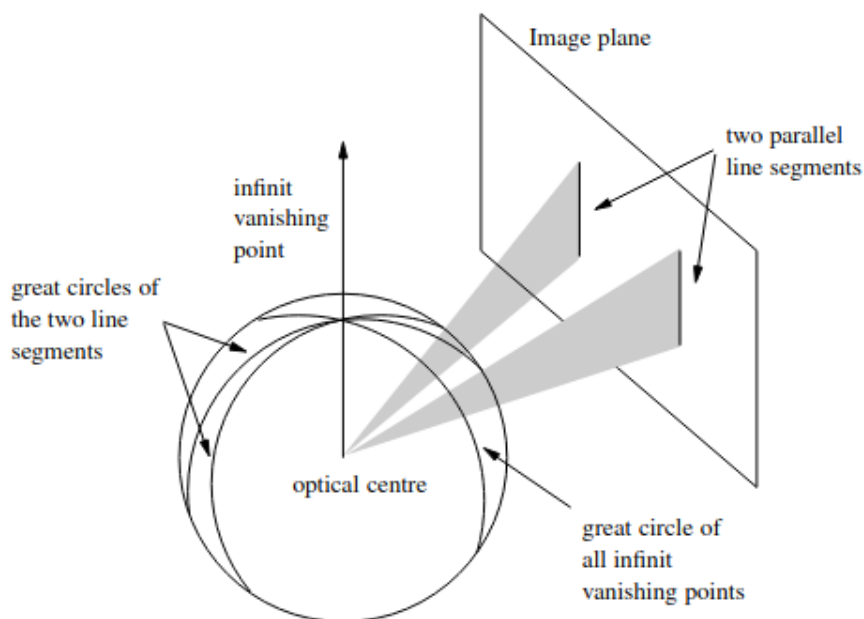
Πριν προχωρήσουμε στις μεθόδους εκτίμησης των σημείων φυγής ας εισάγουμε τον ορισμό της *Gaussian* σφαίρας, καθώς χρησιμοποιείται σε πολλούς αλγόριθμους εκτίμησης σημείων φυγής. Μια *Gaussian* σφαίρα [62] είναι μια προβολική σφαίρα με κέντρο το κέντρο της κάμερας και ακτίνα μονάδα. Στην περίπτωση που χρησιμοποιούμε τη *Gaussian* σφαίρα για να προβάσουμε τις ευθείες σε αυτήν αντί του επιπέδου της εικόνας, τότε η προβολή της θα αντιστοιχεί σε έναν κύκλο πάνω στη σφαίρα (βλ.σχήμα.3.5). Με αυτό τον τρόπο οποιοδήποτε σημείο της 3D ευθείας θα αντιστοιχεί σε ένα σημείο στην περιφέρεια του κύκλου στη *Gaussian* σφαίρα [62]. Επομένως, ένα σύνολο από παράλληλες 3D ευθείες αντιστοιχούν σε



Σχήμα 3.5: Προβολή παράλληλων ευθειών σε δακτύλιους στη *Gaussian* σφαίρα [16].

δακτύλιους επί της *Gaussian* σφαίρας, που θα τέμνονται σε ένα κοινό σημείο, το σημείο φυγής σχήμα.3.6. Κοινό στόχος όλων των αλγορίθμων εκτίμησης σημείων φυγής είναι η ανίχνευση τους που υποστηρίζονται από όσο το δυνατόν περισσότερες παράλληλες 3D ευθείες. Έτσι λοιπόν παρατηρούμε στη βιβλιογραφία ότι οι αλγόριθμοι εκτίμησης σημείων φυγής μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες [18]:

- Αλγόριθμοι εξαντλητικής αναζήτησης.
- Αλγόριθμοι βασισμένοι στον *Expectation – Maximization*(EM) αλγόριθμο.



Σχήμα 3.6: Σημείο φυγής ως τομή δύο δακτυλίων στην Gaussian σφαίρα [17].

- Αλγόριθμοι βασισμένοι στον RANSAC αλγόριθμο.
- Αλγόριθμοι βασισμένοι στην ελαχιστοποίηση αντικειμενικής συνάρτησης.

Επιγραμματικά, οι αλγόριθμοι εξαντλητικής αναζήτησης [17], [63] είναι σχετικά εύκολοι καθώς παράγουν πολλά υποθετικά σημεία φυγής και ελέγχουν κάθε σημείο διεξοδικά εάν είναι ολικά το καλύτερο σημείο φυγής. Για παράδειγμα, η [17] λαμβάνει δείγματα του χώρου αναζήτησης περιστροφής για να καθορίσει την περιστροφή μεγιστοποιώντας τον αριθμό των ευθειών που είναι παράλληλες και έχουν κατεύθυνση έναν άξονα. Αυτή η μέθοδος μπορεί να επιτύχει πολύ ικανοποιητικά αποτελέσματα και είναι σε θέση να επιβάλλει την ορθογωνιότητα των σημείων φυγής. Ωστόσο, εξαρτάται από τον ρυθμό δειγματοληψίας και πρέπει να επεξεργαστεί πολλά δείγματα για λεπτή δειγματοληψία ή μεγάλο χώρο αναζήτησης.

Οι αλγόριθμοι που βασίζονται στον EM αλγόριθμο [64], [65] όπου εκτελούν ταξινόμηση ευθειών που αντιστοιχούν σε ένα σημείο φυγής και αντιπροσωπεύουν ένα άξονα του συστήματος και στη συνέχεια εκτίμηση σημείων φυγής. Πιο αναλυτικά στο E στάδιο εκτελούν ταξινόμηση ευθειών με βάση τα τρέχοντα σημεία φυγής και στο M στάδιο εκτιμούν ξανά τα καινούρια σημεία φυγής μέχρι τη σύγκλιση. Ωστόσο, ένα μειονέκτημα αυτών των μεθόδων είναι ότι η αρχική εκτίμηση των σημείων φυγής θα πρέπει να είναι τουλάχιστον ακριβής για να μπορέσει ο αλγόριθμος να βρει ολικά βέλτιστα σημεία φυγής [18].

Στη συνέχεια οι αλγόριθμοι που βασίζονται στον RANSAC αλγόριθμο [16], [66], [67] ορίζουν ένα ελάχιστο σύνολο λύσεων (*Minimum Solution Set-MSS*) για την εκτίμηση των σημείων φυγής και στη συνέχεια εφαρμόζουν τον RANSAC αλγόριθμο για να δημιουργήσουν τις υποθέσεις για το σημείο φυγής επαναληπτικά και να επιλέξουν την καλύτερη ως τελική λύση [18]. Επί της ουσίας αυτό το ελάχιστο σύνολο λύσεων αφορά το ελάχιστο πλήθος ευθειών που μπορούν να χρησιμοποιηθούν ώστε να εκτιμηθεί με πιθανοτική ακρίβεια το σημείο

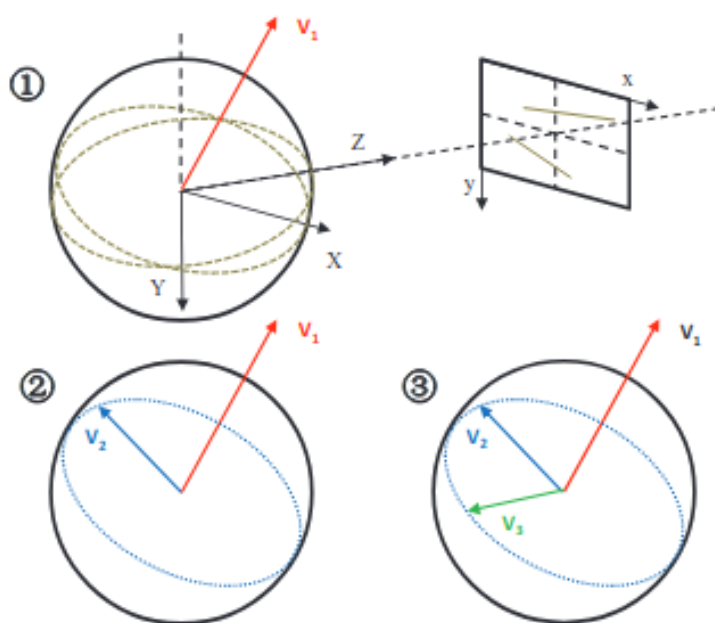
φυγής. Για παράδειγμα, το [16] χρησιμοποιεί σαν ελάχιστο πλήθος ευθειών για την εκτίμηση του σημείου φυγής 3 ευθείες. Επίσης, στην περίπτωση που στην εικόνα παρουσιάζεται ένα σημείο φυγής αυτοί οι αλγόριθμοι θα αποτύχουν να εκτιμήσουν ένα σημείο φυγής.

Τέλος, η τελευταία κατηγορία αλγορίθμων προσπαθούν να επιλύσουν το πρόβλημα της εκτίμησης των σημείων φυγής βρίσκοντας μια ολικά βέλτιστη λύση, βελτιστοποιώντας μια αντικειμενική συνάρτηση. Συγκεκριμένα, στο [68] το πρόβλημα της εκτίμησης των σημείων φυγής μετατρέπεται σε ένα πρόβλημα μεγιστοποίησης συναινετικών συνόλων (*Consensus set maximization*) στον χώρο αναζήτησης περιστροφής και στη συνέχεια επίλυση του με τον αλγόριθμο *Branch and Bound* για την εύρεση του καλύτερου συνόλου σημείων φυγής με περιστροφής, βασιζόμενο δε στη θεωρία της Ανάλυσης Διαστήματος (*Interval Analysis theory*).

Στην παρούσα εργασία, κάνουμε χρήση του αλγορίθμου [18] για την εκτίμηση των ορθοκανονικών σημείων φυγής. Ο αλγόριθμος αυτός είναι ένας συνδυαστικός αλγόριθμος, που χρησιμοποιεί τα χαρακτηριστικά των αλγορίθμων βασιζόμενοι στο *RANSAC* και των πρώτων αλγορίθμων της εξαντλητικής αναζήτησης με σκοπό, την ολικά βέλτιστη εκτίμηση τόσο σε θέση όσο και σε υπολογιστικό χρόνο, τριών ορθοκανονικών σημείων φυγής. Στον αλγόριθμο αυτό γίνεται η χρήση της *Gaussian* σφαίρα για την παραμετροποίηση του χώρου των περιστροφών, με το κέντρο της σφαίρας να αποτελεί το κέντρο της κάμερας και ορίζεται ένα ελάχιστο σύνολο λύσεων δυο ευθειών, πάνω στο οποίο θα εκτελεστεί ο *RANSAC* αλγόριθμος για την εύρεση των σημείων φυγής. Όπως είδαμε και νωρίτερα, δύο παράλληλες 3D ευθείες και προβάλλονται στη *Gaussian* σφαίρα ως δύο μεγάλοι κύκλοι που τέμνονται σε ένα σημείο. Η κατεύθυνση αυτού του σημείου από την αρχή της σφαίρας θεωρείται ως υποψήφια κατεύθυνση σημείου φυγής (\mathbf{v}_1). Για να επιτευχθεί απόδοση σε πραγματικό χρόνο, δημιουργείται ένα πολικό πλέγμα που τέμνει το επίπεδο της εικόνας και εκτείνεται στο γεωγραφικό πλάτος και μήκος στο μισό της *Gaussian* σφαίρας με μέγεθος 90×360 και ακρίβεια 1° . Έτσι, ένα ζεύγος γραμμών που τέμνονται σε ένα σημείο του επιπέδου της εικόνας συνεισφέρει στο αντίστοιχο κελί του πολικού πλέγματος κατά βάρος:

$$score = score + \|l_0\| \times \|l_1\| \times \sin(2\theta) \quad (3.6)$$

Όπου $\|l_0\|$ και $\|l_1\|$ αντιπροσωπεύουν το μήκος δύο τμημάτων ευθειών σε *pixel* και θ είναι η γωνία μεταξύ τους και *score* είναι η συσσωρευμένη βαθμολογία σε κάθε κελί του πολικού πλέγματος. Αφού δημιουργηθούν οι πρώτες υποψήφιας κατευθύνσεις των σημείων φυγής (\mathbf{v}_1) και λάβουμε υπόψη τον ορθογώνιο περιορισμό, το δεύτερο σημείο φυγής (\mathbf{v}_2) πρέπει να βρίσκεται στον μεγάλο κύκλο που είναι κάθετος στον μεγάλο κύκλο του (\mathbf{v}_1) δημιουργώντας 360 ομοιόμορφα καταναμημένα υποψήφια σημεία φυγής (\mathbf{v}_i , $i=1$ to 360) (βλ. σχήμα.3.7). Δεδομένου των δύο σημείων φυγής το τρίτο βρίσκεται με απλό εξωτερικό γινόμενο. Τέλος, οι βέλτιστες εκτιμήσεις σημείων φυγής, λαμβάνονται από το σύνολο των υποψήφιας σημείων φυγής με την υψηλότερη βαθμολογία, όπου η βαθμολογία ενός συνόλου υποθέσεων σημείων φυγής, είναι το άθροισμα των βαθμολογιών τριών κυψελών πολικού πλέγματος που ανήκουν στις τρεις σχετικές κάθετες κατευθύνσεις. Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στο [18] για να εξερευνήσει λεπτομερειακά τον τρόπο λειτουργίας του αλγορίθμου.



Σχήμα 3.7: Παραγωγή υποθέσεων σημείων φυγής [18].

Κεφάλαιο 4

Εκτίμηση Τροχιάς Κάμερας και Βελτιστοποιήσεις

Υποθέτοντας ότι έχουμε στη διάθεση μας, μια συλλογή από εικόνες (frames) επιδιώκουμε, να εκτιμήσουμε την κίνηση της κάμερας στο χώρο των επιτρεπτών κινήσεων του σώματος $SE(3)$, αναπαριστώντας το μονοπάτι ως μια αλληλουχία από πόζες T_i, T_{i+1}, T_{i+2} κ.ο.κ, ενώ, παράλληλα επιθυμούμε να ανακατασκευάσουμε το περιβάλλον.

Στο κεφάλαιο αυτό θα συζητήσουμε για τους τρόπους εκτίμησης της τροχιάς της κάμερας, καθώς και μεθόδους που τη βελτιστοποιούν. Πριν όμως προχωρήσουμε ας ορίσουμε ορισμένα μαθηματικά μοντέλα που θα μας βοηθήσουν να μοντελοποιήσουμε τις μεθόδους για να εκτιμήσουμε την τροχιά.

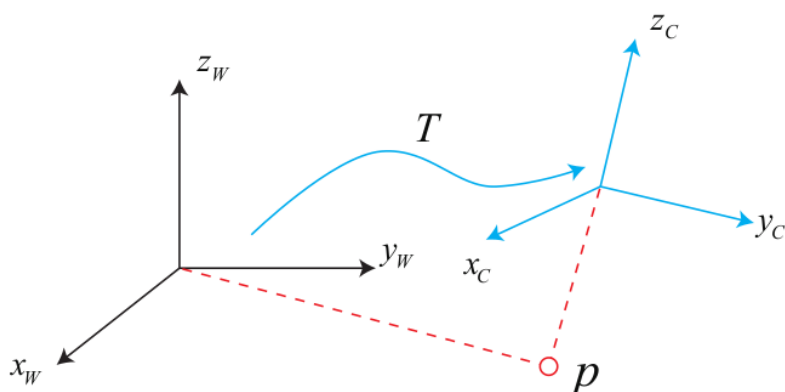
4.1 3D Κίνηση Άκαμπτου Σώματος

Στα τελευταία κεφάλαια, αναλύσαμε την αρχιτεκτονική ενός κλασικού μοντέλου *SLAM* και αρχικές του υλοποιήσεις. Αυτή η ενότητα θα εισαγάγει ένα από τα θεμελιώδη προβλήματα του *SLAM*: Πώς να περιγράψουμε την κίνηση ενός άκαμπτου σώματος στο τρισδιάστατο χώρο. Διαισθητικά, γνωρίζουμε ότι αυτό αποτελείται από μία περιστροφή συν μία μετατόπιση και καλείται ως **3D Κίνηση Άκαμπτου Σώματος**. Αυτή η κίνηση, περιγράφεται από τον μετασχηματισμό $T_{cw} \in SE(3)$ (βλ. σχήμα.4.1) με $\mathbf{t} \in \mathcal{R}^3$ να είναι το διάνυσμα μετατόπισης και ένα $\mathbf{R} \in SO(3)$ τον πίνακα περιστροφής (4.2). Ο χώρος *Special Orthogonal* ($SO(3)$) καλείται και ως χώρος περιστροφών ή *Lie group*, που καθορίζει τη συμπεριφορά του πίνακα στροφής.

$$T_{wc} = \begin{bmatrix} \mathbf{R}_{wc} & \mathbf{t}_{wc} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \in \mathcal{R}^{4 \times 4} \quad (4.1)$$

Επομένως ο αντίστροφος πίνακας μετασχηματισμού θα μας δώσει:

$$T_{cw} = T_{wc}^{-1} = \begin{bmatrix} \mathbf{R}_{wc}^T & -\mathbf{R}_{wc}^T \cdot \mathbf{t}_{wc} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \in \mathcal{R}^{4 \times 4} \quad (4.2)$$



Σχήμα 4.1: 3D Άκαμπτος Μετασχηματισμός T μεταξύ γειτονικών συστημάτων συντεταγμένων [19].

4.2 Εκτίμηση πόζας (Pose Estimation)

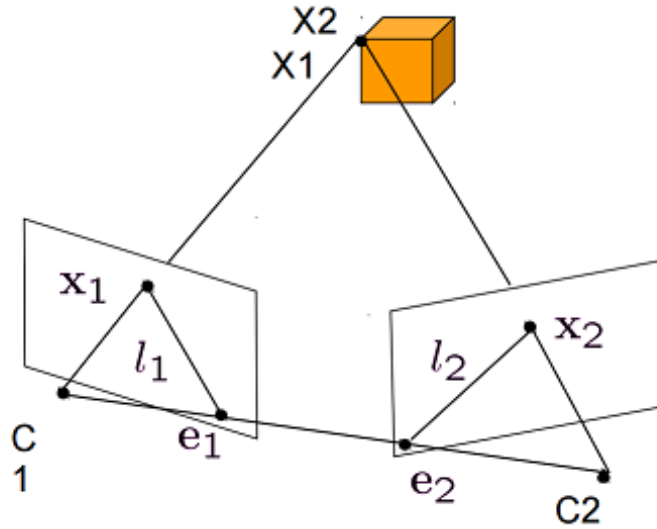
Η εκτίμηση της πόζας μπορεί να υπολογιστεί κλασσικά χρησιμοποιώντας αντιστοιχίες 2Δ/2Δ χαρακτηριστικών της εικόνας όπως *ORB* γωνίες, αξιοποιώντας τη γεωμετρία πολλών όψεων [15] (ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στο [15] για βαθύτερη κατανόηση) ή μέσω 2Δ/3Δ αντιστοιχίες, ή 3Δ/3Δ αντιστοιχίες, είτε *direct* χρησιμοποιώντας τη φωτεινότητα της εικόνας είτε μέσω σημείων φυγής όπως χρησιμοποιούμε στην παρούσα εργασία.

4.2.1 Μέσω 2Δ/2Δ αντιστοιχίσεων

Στην περίπτωση που έχουμε ένα σύνολο από 2Δ ταιριάσματα μεταξύ δύο εικόνων, μπορούμε να υπολογίσουμε τη σχετική κίνηση των δύο εικόνων και θεωρώντας ότι η πρώτη εικόνα έχει μηδενική περιστροφή και μετατόπιση, μπορεί να υπολογιστεί και η πόζα της εκάστοτε κάμερας. Συγκεκριμένα, έχοντας το σύνολο των 2Δ/2Δ ταιριασμάτων, ο υπολογισμός της σχετικής κίνησης μπορεί να γίνει μέσα από τον υπολογισμό και την αποσύνθεση είτε του *Essential* πίνακα, είτε του *Fundamental* πίνακα, είτε του *Homography* πίνακα, ανάλογα με το είδος του περιβάλλοντος. Δηλαδή, για τις περιπτώσεις που έχουμε μη επίπεδες σκηνές, εάν γνωρίζουμε τον πίνακα βαθμονόμησης της κάμερας K τότε μπορούμε να υπολογίσουμε τον *Essential* πίνακα και αποσυνθέτοντας τον υπολογίζουμε τη σχετική κίνηση, αλλιώς υπολογίζουμε τον *Fundamental* πίνακα και αποσυνθέτοντας τον υπολογίζουμε τη σχετική κίνηση. Ειδικά, στην περίπτωση που έχουμε επίπεδη σκηνή υπολογίζουμε τον *Homography* πίνακα και με ίδια βήματα με τους άλλους δύο πίνακες υπολογίζουμε τη σχετική περιστροφή. Πιο αναλυτικά έχουμε:

Essential πίνακας [15]:

Έστω ότι έχουμε δυο διαδοχικές εικόνες, και ένα σημείο \mathbf{X} στο χώρο που προβάλλεται στα πλαίσια των δυο εικόνων, στα σημεία X_1 και X_2 με C_1 και C_2 τα αντίστοιχα κέντρα (βλ.σχήμα.4.2. Η ευθεία που (C_1, C_2) τέμνει τα δυο επίπεδα εικόνας στα επιπολικά σημεία e_1, e_2 αντίστοιχα. Οι επιπολικές ευθείες l_1, l_2 προκύπτουν από την τομή του επιπολικού



Σχήμα 4.2: 2Δ/2Δ αντιστοιχίσεις μεταξύ δύο καμερών (C_1, C_2) [20].

επιπέδου που δημιουργείται από τα σημεία (C_1, C_2, \mathbf{X}), με τα επίπεδα εικόνας. Γνωρίζουμε ότι το \mathbf{x}_1 (που είναι σε ομοιογενείς συντεταγμένες) αποτελεί την προβολή του 3D σημείου \mathbf{X} στο επίπεδο εικόνας 1. Δεδομένου ότι η κάμερα είναι βαθμονομημένη, άρα ο πίνακας K θεωρείται γνωστός και υποθέτοντας ότι η πρώτη κάμερα είναι στάσιμη, έχουμε απλώς μια προβολή με άγνωστο βάθος l_1 . Από την πρώτη έως το δεύτερη εικόνα έχουμε μια περιστροφή κατά \mathbf{R} και μια μετατόπιση κατά \mathbf{t} ακολουθούμενη από προβολή. Έτσι έχουμε [15]:

$$\mathbf{x}_2 = \mathbf{R}(\mathbf{x}_1 - \mathbf{t}) \quad (4.3)$$

Επειδή όμως το \mathbf{x}_1 και το \mathbf{x}_2 ανήκουν στο επιπολικό επίπεδο θα ισχυεί:

$$(\mathbf{x}_1 - \mathbf{t})^T (\mathbf{t} \times \mathbf{x}_1) = 0 \quad (4.4)$$

$$(\mathbf{x}_2^T \mathbf{R})(\mathbf{t} \times \mathbf{x}_1) = 0 \quad (4.5)$$

$$\mathbf{x}_2^T (\mathbf{R}[\mathbf{t}]_x) \mathbf{x}_1 = 0 \quad (4.6)$$

Επομένως, ο πίνακας $E = \mathbf{R}[\mathbf{t}]_x \in \mathcal{R}^{3 \times 3}$ καλείται Essential πίνακας [15]. Για να θεωρηθεί ένας πίνακας E ως Essential πίνακας θα πρέπει να ικανοποιεί το θεώρημα του [69] το οποίο αναφέρει ότι ένας μη μηδενικός πίνακας καλείται ως Essential πίνακας εάν και μόνο εάν μπορεί να διαγωνοποιηθεί με βάση τη μέθοδο SVD με $E = USV^T$ και $S = \text{diag}(s, s, 0)$ να είναι ο πίνακας ιδιοτιμών για κάποιο $s > 0$ και $U, V \in SO(3)$. Χρησιμοποιώντας 5 σημεία μπορούμε να υπολογίσουμε τον Essential και άρα να υπολογίσουμε τη σχετική κίνηση ως εξής [15]:

$$\mathbf{R} \in UWV^T, UW^T V^T \quad (4.7)$$

$$\mathbf{t} = \pm \hat{\rho} u_3, \hat{\rho} \in \mathcal{R}^+ \quad (4.8)$$

και

$$W = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

Όπου λ είναι ένας παράγοντας κλίμακας που συνήθως τίθεται με τη μονάδα. Από τις παραπάνω σχέσεις προκύπτουν 4 λύσεις και δεχόμαστε αυτή τη λύση που δίνει θετικό βάθος στα σημεία του χώρου. Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στο [15] για μια εκτενέστερη περιγραφή του αλγορίθμου 5 σημείων για την εύρεση του Essential πίνακα.

Fundamental πίνακας [15]: Στην περίπτωση που έχουμε μη βαθμονομημένη κάμερα ο πίνακας που χρησιμοποιούμε για την εύρεση της σχετικής κίνησης είναι ο Fundamental πίνακας που σχετίζεται με τον Essential πίνακα με τη σχέση [15]:

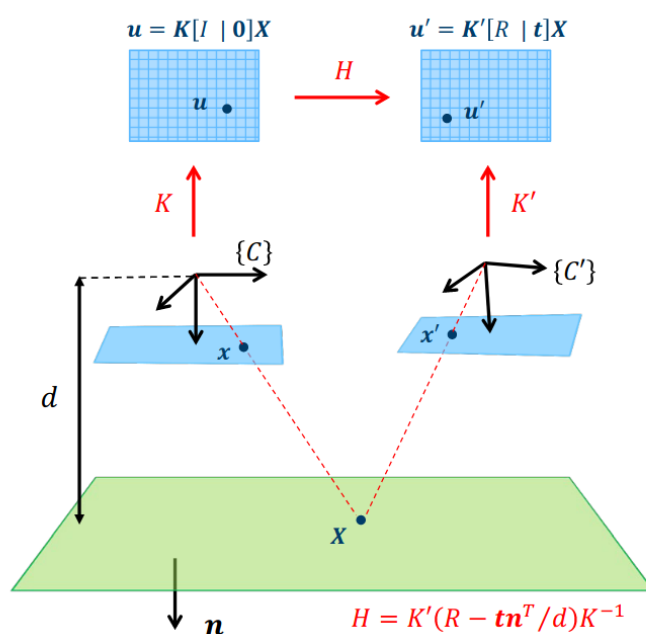
$$F = K^{-T}EK^{-1} \quad (4.10)$$

όπου η εξ. 4.6 γίνεται:

$$\mathbf{x}_2^T(K^T FK)\mathbf{x}_1 = 0 \quad (4.11)$$

Άρα εκτελώντας τα ίδια βήματα με τον Essential πίνακα, μόνο που εδώ χρειαζόμαστε 7 σημεία μπορούμε να υπολογίσουμε τη σχετική περιστροφή.

Homography πίνακας [15]: Στην περίπτωση που έχουμε επίπεδη σκηνή (βλ. σχήμα. 4.3), δεν μπορούμε να χρησιμοποιήσουμε τους παραπάνω πίνακες για εκτιμήσουμε τη σχετική περιστροφή και ύστερα την τροχιά της κάμερας, διότι τα σημεία κείτονται στο ίδιο επίπεδο και αποτελεί μια εκφυλισμένη περίπτωση για τους ανωτέρω πίνακες. Έτσι λοιπόν, έχο-



Σχήμα 4.3: Εύρεση σχετικής περιστροφής από την εύρεση του πίνακα ομογραφίας H [21].

νας 2 frames η σχετική περιστροφή τους μπορεί να υπολογιστεί από την αποσύνθεση της

ομογραφίας (Homography):

$$H = K(\mathbf{R} - \mathbf{t} \frac{\mathbf{n}}{\mathbf{d}})K^{-1} \quad (4.12)$$

Όπου \mathbf{n} είναι το κάθετο διάνυσμα του επιπέδου στο οποίο ανήκουν τα σημεία και \mathbf{d} είναι η απόσταση του επιπέδου από τις κάμερες. Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στο [15].

4.2.2 Μέσω 2D/3D αντιστοιχίσεων

Σε εφαρμογές VSLAM που χρησιμοποιούν χαρακτηριστικά το πρόβλημα της εκτίμησης της πόζας, αφορά την εύρεση του πίνακα προβολής P , χρησιμοποιώντας ένα σύνολο από γνωστά σημεία στο 3D χώρο και τα αντίστοιχα τους στο επίπεδο της εικόνας. Γνωρίζοντας ότι τα σημεία χώρου συνδέονται με τα σημεία της εικόνας με τον τύπο:

$$l_i \cdot \mathbf{x}_c = K \cdot [\mathbf{R}|\mathbf{t}] \cdot \mathbf{X}_w \quad (4.13)$$

Όπου l_i είναι ένας συντελεστής κλίμακας και τα \mathbf{x}_c και \mathbf{X}_w είναι σε ομογενείς συντεταγμένες, θέλουμε να υπολογίσουμε το \mathbf{R} και το \mathbf{t} . Αλγόριθμοι, όπως ο PnP (*Efficient Perspective from n Points*) [70], λύνουν το πρόβλημα χρησιμοποιώντας n σημεία του χώρου και χρησιμοποιείται στο [23]. Όπως και ο αλγόριθμος *Lambda Twist P3P* [71] που λύνει το πρόβλημα, αξιοποιώντας μόνο 3 σημεία, που είναι και το ελάχιστο πλήθος σημείων που μπορεί να αξιοποιηθεί.

4.2.3 Μέσω 3D/3D αντιστοιχίσεων

Δοθέντος ενός συνόλου 3D/3D αντιστοιχίσεων, μπορούμε να εκτιμήσουμε την πόζα της κάμερας, χρησιμοποιώντας τον αλγόριθμο ICP [34], [38], [39] ή τις παραλλαγές του, ορίζοντας ένα γεωμετρικό σφάλμα μεταξύ των αντιστοιχίσεων και ελαχιστοποιώντας το υπολογίζεται ο πίνακας περιστροφής \mathbf{R} και ο \mathbf{t} . Το γεωμετρικό σφάλμα μπορεί να περιγράψει είτε τη διαφορά μεταξύ σημείων του ενός συνόλου με το άλλο, είτε μεταξύ σημείων του ενός συνόλου και ενός επιπέδου του άλλου συνόλου. Στην πρώτη περίπτωση που έχουμε σφάλμα μεταξύ σημείου προς σημείου, το κόστος που ορίζεται δίνεται από τον τύπο:

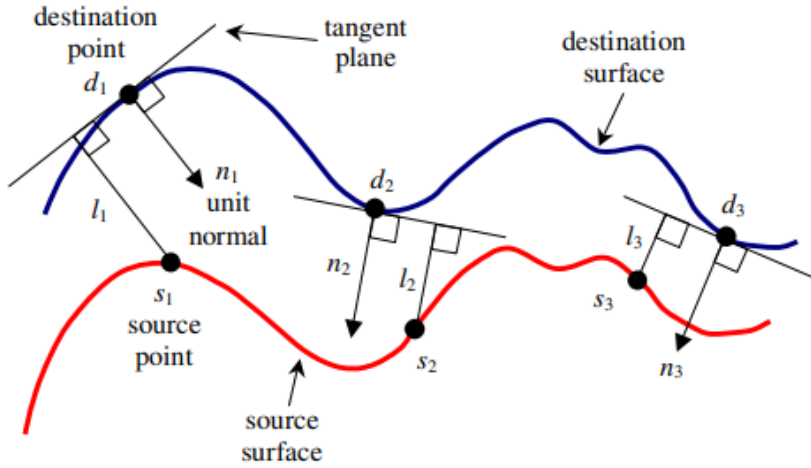
$$E = \sum |\mathbf{R} \cdot \mathbf{p}_i + \mathbf{t} - \mathbf{q}_i|^2 \quad (4.14)$$

Με p_i το σημείο που ανήκει στη μια εικόνα, q_i το σημείο στη δεύτερη εικόνα, ενώ \mathbf{R} και \mathbf{t} οι ζητούμενοι πίνακες μετασχηματισμού.

Στη δεύτερη περίπτωση η συνάρτηση κόστους προς ελαχιστοποίηση έχει τη μορφή:

$$E = \sum [(\mathbf{R} \cdot \mathbf{p}_i + \mathbf{t} - \mathbf{q}_i) \cdot \mathbf{n}_i]^2 \quad (4.15)$$

Δηλαδή για κάθε σημείο της αρχικής εικόνας φέρουμε, το εφαπτόμενο επίπεδο του με κάθετο διάνυσμα το \mathbf{n}_i και προβάλλοντας το σημείο της δεύτερης εικόνας πάνω στο εφαπτόμενο επίπεδο (βλ.σχήμα.4.4) [22], υπολογίζουμε την απόσταση τους.



Σχήμα 4.4: Απόσταση σημείου από επίπεδο μεταξύ δύο καμπυλών [22].

4.2.4 Μέθοδοι Φωτεινότητας (Direct methods)

Σε direct εφαρμογές που δεν εξάγουν χαρακτηριστικά, αλλά χρησιμοποιούν την ένταση της εικόνας, η πόζα μπορεί να εκτιμηθεί μέσα από την ελαχιστοποίηση του φωτομετρικού σφάλματος διαδοχικών εικόνων, δηλαδή μέσα από την ελαχιστοποίηση του σφάλματος [38]:

$$\min_{\xi \in \text{set}(3)} \sum_i \rho(I_1(p_i) - I_2(\pi(\mathbf{R}_{cw}^2(\mathbf{R}_{wc}^1 \pi^{-1}(p_i, d) + \mathbf{t}_{wc}^1) + \mathbf{t}_{cw}^2))) \quad (4.16)$$

Όπου d είναι το βάθος του pixel, $\pi(\cdot)$ είναι ο προβολικός πίνακας και $\rho(\cdot)$ είναι ένας πυρήνας ομαλοποίησης (όπως ο *Huber* ή ο *Cauchy*), σε περιπτώσεις ακραίων σφαλμάτων. Ελαχιστοποιώντας για όλα τα pixel της εικόνας, λαμβάνουμε την εκτίμηση της πόζας. Αυτή, η ελαχιστοποίηση του σφάλματος μπορεί να γίνει, εκτελώντας είτε κάποιον από τους αλγορίθμους μη γραμμικής βελτιστοποίησης ελαχίστων τετραγώνων ή μπορεί να χρησιμοποιηθεί και ένα συνελκτικό νευρωνικό δίκτυο (CNN [43]) όπως το *PoseNet* στο [41].

4.2.5 Μέσω διαχωρισμού εκτίμησης πόζας

Ένας άλλος τρόπος για την εκτίμηση της πόζας είναι η εκτίμηση της σε δύο ή άλλος ο διαχωρισμός της. Στο πρώτο στάδιο εκτιμάται η περιστροφή με βάση κάποια χαρακτηριστικά και στη συνέχεια εκτιμάται η μετατόπιση, με μικρότερο υπολογιστικό κόστος χρησιμοποιώντας λιγότερη πληροφορία, αφού γνωστή η περιστροφή. Πιο αναλυτικά, έχοντας υπολογίσει τρία ορθοκανονικά σημεία φυγής για το j -οστό frame, η περιστροφή του μπορεί να υπολογιστεί ως εξής [47]:

$$\mathbf{v}\mathbf{p}_i = \lambda K^{-1} \mathbf{R}_j \mathbf{e}_i, i = 0, 1, 2 \quad (4.17)$$

$$\mathbf{R}_j = \mathbf{R}_j \mathbf{R} \mathbf{M}^T \quad (4.18)$$

όπου τα \mathbf{e}_i είναι τα μοναδιαία διανύσματα βάσης και το $\mathbf{v}\mathbf{p}_i$ είναι το αντίστοιχο σημείο φυγής. Ωστόσο, έχοντας θα πρέπει την περιστροφή που υπολογίσαμε από την εξ. 4.17 να την πολλαπλασιάσουμε με την περιστροφή $\mathbf{R}\mathbf{M}$ του *Manhattan* πλαισίου του αρχικού *frame* που υπολογίζεται από την ίδια εξίσωση. Επίσης, επειδή η αντίστοιχη στήλη του πίνακα περιστρο-

φής πρέπει να έχει νόρμα μονάδα, υπολογίζουμε την παράμετρο λ .

Διαφορετικά, γνωρίζοντας ποιο σύνολο από τις ευθείες παράγει το ορθοκανονικό σημείο φυγής μπορούμε να υπολογίσουμε την περιστροφή με το εξής συλλογισμό. Τα κάθετα διανύσματα του συνόλου των ευθειών που έχουν χρησιμοποιηθεί για την παραγωγή του σημείου φυγής, θα πρέπει να είναι κάθετα με την κατεύθυνση του σημείου φυγής. Δηλαδή αν:

$$\mathbf{s}_i = K^T \mathbf{l}_i, \mathbf{s}_i \in R^3, [15] \quad (4.19)$$

και

$$S = \begin{bmatrix} \mathbf{s}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{s}_n \end{bmatrix} \quad (4.20)$$

Όπου n είναι το πλήθος των ευθειών του συνόλου και $S \in R^{3 \times n}$ τότε θα πρέπει να ισχύει:

$$S^T \mathbf{d}_j = 0, j = 1, 2, 3 \quad (4.21)$$

Όπου \mathbf{d}_j είναι η κατεύθυνση του σημείου φυγής. Άρα στο k -οστο frame υπολογίζοντας και τα τρία \mathbf{d}_j και άρα κατασκευάζοντας την περιστροφή $MF_k = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$ [46], η απόλυτη περιστροφή του μπορεί να υπολογιστεί ως εξής [46]:

$$\mathbf{R}_k = MF_k MF_w^T \quad (4.22)$$

Όπου MF_w είναι η περιστροφή του Manhattan Frame στο πρώτο frame έχοντας υπολογιστεί από τον τύπο εξ.4.21.

Έχοντας υπολογίσει τις περιστροφές στο πρώτο βήμα, η εκτίμηση της μετατόπισης μπορεί να γίνει από την ελαχιστοποίηση του σφάλματος επαναπροβολής των σημείων ή και των σημείων μαζί με τις ευθείες [37].

4.3 Βελτιστοποιήσεις (Back end) του συστήματος

4.3.1 Bundle Adjustment

Ολοκληρώνοντας με τις διαδικασίες του *front – end* τμήματος, οδηγούμαστε στο *back – end* τμήμα. Το τμήμα αυτό είναι υπεύθυνο για τη βελτιστοποίηση, τόσο της πόζας όσο και της δομής του χάρτη, που έχει δημιουργηθεί μέχρι πρότινος. Και αυτό διότι, τα δεδομένα που λαμβάνουμε είναι θορυβώδη και δεν είμαστε σίγουροι ότι η εκτίμηση της πόζας είναι η αληθινή ή ότι έχουμε δημιουργήσει μια συνεπή αναπαράσταση του χώρου. Έτσι έχει προταθεί στη βιβλιογραφία ο αλγόριθμος *Bundle Adjustment (BA)* [30] που προσπαθεί να βελτιστοποιήσει από κοινού τις συντεταγμένες ενός συνόλου 3D σημείων και τις πόζες ενός συνόλου καμερών, ελαχιστοποιώντας μια αντικειμενική συνάρτηση. Συγκεκριμένα έστω ότι έχουμε στη διάθεση μας ένα σύνολο X που αποτελείται από τα 2D σημεία που παρατηρούμε σε μια εικόνα και ένα σύνολο m εικόνες. Ορίζουμε ως αντικειμενική συνάρτηση προς

βελτιστοποίηση τη συνάρτηση :

$$E(\{\mathbf{R}_i, \mathbf{t}_i\}_{i=1,\dots,m}, \{\mathbf{X}_j\}_{j=1,\dots,N}) = \sum_{i=1}^m \sum_{j=1}^N |\mathbf{x}_i^j - \pi(\mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)|^2 \quad (4.23)$$

Όπου \mathbf{x}_i^j αποτελεί το 2Δ σημείο στην εικόνα i και \mathbf{X}_j το αντίστοιχο 3Δ σημείο, ενώ οι πίνακες περιστροφής \mathbf{R}_i και \mathbf{t}_i συνθέτουν την πόζα της κάμερας. Συχνά αυτή η μέθοδος χρησιμοποιείται στο τέλος, καθώς η παραπάνω αντικειμενική συνάρτηση είναι μη κυρτή και η βελτιστοποίηση είναι χρονοβόρα διαδικασία.

Μια παραλλαγή αυτού του αλγορίθμου, που χρησιμοποιούμε και στην παρούσα εργασία είναι η ενσωμάτωση στην εξ.4.23 πληροφορία γεωμετρικών σχημάτων παραπάνω των σημείων. Δηλαδή στην περίπτωση μας η ενσωμάτωση πληροφορίας εξαγόμενων ευθειών. Πιο αναλυτικά, όπως και στο [24] έστω ότι $\mathbf{P}, \mathbf{Q} \in \mathcal{R}^3$ είναι τα δύο άκρα μιας 3Δ ευθείας και $\mathbf{p}_d, \mathbf{q}_d \in \mathcal{R}^2$ και $\mathbf{p}_d^h, \mathbf{q}_d^h$ είναι η προβολή των 3Δ σημείων της ευθείας στο επίπεδο της εικόνας και σε ομογενείς συντεταγμένες αντίστοιχα. Έτσι λοιπόν μπορούμε να ορίσουμε το σφάλμα επαναπροβολής ως η ελαχιστοποίηση του αθροίσματος του σφάλματος μεταξύ επαναπροβαλλόμενου σημείου και ευθείας. Δηλαδή :

$$E_{line}(\mathbf{P}, \mathbf{Q}, l, \mathbf{R}_i, \mathbf{t}_i, K) = E_{pl}^2(\mathbf{P}, l, \mathbf{R}_i, \mathbf{t}_i, K) + E_{pl}^2(\mathbf{Q}, l, \mathbf{R}_i, \mathbf{t}_i, K) \quad (4.24)$$

με l να είναι η ευθεία στο επίπεδο της εικόνας :

$$l = \frac{\mathbf{p}_d^h \times \mathbf{q}_d^h}{\|\mathbf{p}_d^h \times \mathbf{q}_d^h\|} \quad (4.25)$$

και :

$$E_{pl}^2(\mathbf{P}, l, \mathbf{R}_i, \mathbf{t}_i, K) = l^T \pi(\mathbf{P}, \mathbf{R}_i, \mathbf{t}_i, K) \quad (4.26)$$

Έτσι λοιπόν έχοντας έναν χάρτη με n 3Δ σημεία ευθειών $\mathbf{P}_j, \mathbf{Q}_j, j=1, \dots, n$ [24] προβάλλοντας αυτά τα σημεία στην εικόνα λαμβάνουμε σε ομογενείς συντεταγμένες :

$$\widehat{\mathbf{p}}_{ij}^h = \pi(\mathbf{P}_j, \mathbf{R}_i, \mathbf{t}_i) \quad (4.27)$$

$$\widehat{\mathbf{q}}_{ij}^h = \pi(\mathbf{Q}_j, \mathbf{R}_i, \mathbf{t}_i) \quad (4.28)$$

Χρησιμοποιώντας τα παραπάνω σημεία μαζί με την εξ.4.25 λαμβάνουμε τη 2Δ ευθεία \widehat{l}_{ij} για το i -οστό $frame$. Ενώ, δοθέντος των 2Δ παρατηρήσεων των άκρων της j -οστής ευθείας p_{ij}, q_{ij} που έχουν υπολογιστεί από τον αλγόριθμο ανίχνευσης ευθειών, ορίζουμε το σφάλμα επαναπροβολής μεταξύ σημείου και ευθείας ως :

$$e'_{ij} = \widehat{l}_{ij}^T (K^{-1} \mathbf{p}_{ij}^h) \quad (4.29)$$

$$e''_{ij} = \widehat{l}_{ij}^T (K^{-1} \mathbf{q}_{ij}^h) \quad (4.30)$$

Τελικά, συνδυάζοντας τις εξ.4.29, 4.30 με την εξ.4.23 καταλήγουμε στη συνδυασμένη εξίσω-

ση προς βελτιστοποίηση της πόζας [24]:

$$C = \sum_{\tilde{y}} \rho(e_{\tilde{y}}^T \omega_{\tilde{y}}^{-1} e_{\tilde{y}} + e_{\tilde{y}}'^T \Omega_{\tilde{y}}'^{-1} e_{\tilde{y}}' + e_{\tilde{y}}''^T \Omega_{\tilde{y}}''^{-1} e_{\tilde{y}}'') \quad (4.31)$$

Όπου ρ είναι η *Huber* συνάρτηση κόστους και το σφάλμα $e_{\tilde{y}}$ είναι το σφάλμα επαναπροβολής της εξ. 4.23. Το παραπάνω σφάλμα μας χρησιμοποιείται και σε αυτήν τη διπλωματική εργασία για βελτιστοποίηση της πόζας.

Κεφάλαιο 5

Υλοποίηση

5.1 Παρουσίαση Μεθοδολογίας

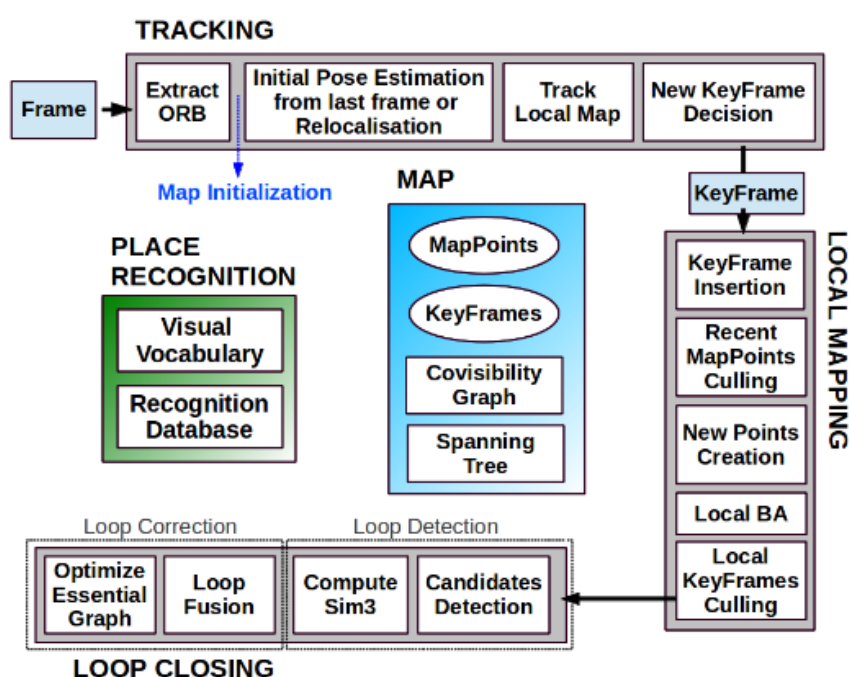
Σε αυτήν την ενότητα θα παρουσιάσουμε τη λύση στο πρόβλημα του *SLAM*. Επειδή τα περισσότερα περιβάλλοντα περιέχουν πληθώρα γεωμετρικά σχήματα όπως ευθείες, επίπεδα και σημεία φυγής, σκοπός μας είναι να δημιουργήσουμε ένα μοντέλο συστήματος *SLAM* που θα είναι σε θέση να ανιχνεύει αυτά τα γεωμετρικά χαρακτηριστικά πέραν των σημείων, ώστε να μπορεί να εκτιμήσει καλύτερα την πόζα του ρομπότ μέσα στο χώρο. Επίσης, αυτή η παραπάνω πληροφορία θα οδηγήσει τόσο στον εμπλουτισμό του χάρτη ανακατασκευής όσο και στην καλύτερη βελτίωση της ίδιας της πόζας. Επιπλέον, θα ωφελήσει το σύστημα να συνεχίσει να εκτιμάει τη θέση του στο χώρο και την ανακατασκευή του περιβάλλοντός του, σε καταστάσεις όπως εσωτερικοί διάδρομοι που συγκεκριμένα γεωμετρικά σχήματα όπως ευθείες παρουσιάζονται σε μεγαλύτερο όγκο και είναι πιο εύκολα να ανιχνευθούν από ότι σημεία, όπως γωνίες. Επομένως, σε αυτήν την εργασία στοχεύουμε στην αξιοποίηση των ευθειών και των σημείων φυγής του περιβάλλοντος, ώστε να χρησιμοποιήσουμε μεθοδολογίες για καλύτερη εκτίμηση της θέσης του ρομπότ, καθώς και διαμόρφωσης καινούριων σχημάτων βελτιστοποίησης που στηρίζονται στα σημεία φυγής και σε συνδυασμό με τα κλασικά σχήματα βελτιστοποίησης. Η βασική μας διαφορά με το [23] είναι ότι χρησιμοποιούμε τον *LSD*[58] αλγόριθμο για να ανιχνεύσουμε σε $O(n)$ χρόνο ευθείες και τον αλγόριθμο [18] για να ανιχνεύσουμε σημεία φυγής και χρησιμοποιούμε αυτές τις πληροφορίες για να εκτιμήσουμε και να βελτιώσουμε τη 3D πόζα της κάμερας. Επίσης, η βασική μας διαφορά με το [24] είναι η προέκταση του και στο να ανιχνεύει σημεία φυγής και να εκτιμάει όποτε είναι δυνατόν την πόζα του και να τη βελτιστοποιεί χρησιμοποιώντας σημεία φυγής. Επιπλέον, με αυτόν τον τρόπο μας δίνεται η δυνατότητα να εκτιμήσουμε τμηματικά την πόζα, πρώτα την περιστροφή αξιοποιώντας ορισμένα γεωμετρικά σχήματα του περιβάλλοντος και έπειτα τη χρήση της περιστροφής για την κατασκευή μεθόδου εκτίμησης της μετατόπισης, με μεγαλύτερη ακρίβεια. Έτσι, λοιπόν είναι πολύ σημαντικό για ένα *SLAM* σύστημα να είναι σε θέση να αντλήσει όσο τον δυνατόν περισσότερη πληροφορία από το περιβάλλον του με γνώμονα την καλύτερη εκτίμηση της τροχιάς του και της ακριβούς ανακατασκευής του χώρου. Επομένως, η δουλειά που παράχθηκε ήταν στην κατεύθυνση της χρήσης γρήγορων αλγορίθμων εκτίμησης ευθειών και σημείων φυγής και η αξιοποίηση αυτών ώστε να δημιουργηθεί ένα σχήμα εκτίμησης και βελτίωσης της πόζας.

5.2 Υλοποιήσεις καλύτερου συστήματος (State of the Art Implementation)

Σε αυτήν την ενότητα θα αναφέρουμε επιγραμματικά την αρχιτεκτονική του *state of art SLAM* συστήματος [23] που στηριχθήκαμε, καθώς και την αρχιτεκτονική του δεύτερου *SLAM* συστήματος [24] που επεκτείνει το πρώτο και από το οποίο λάβαμε κάποιες ιδέες.

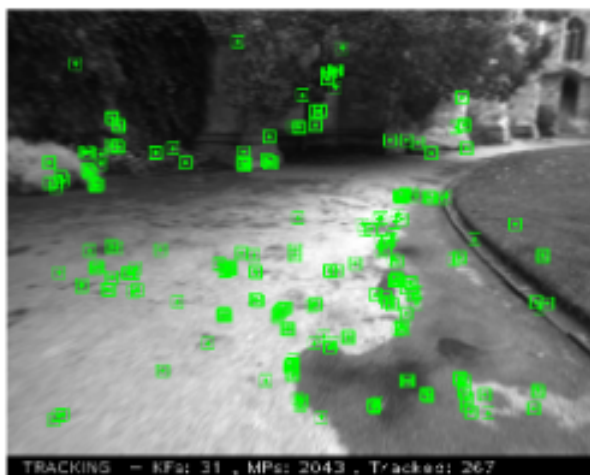
ORB-SLAM2 [23].

Το *ORB – SLAM2* [23] αποτελεί ένα από τα *state of art SLAM* συστήματα στο κλάδο του και αφορά ένα *SLAM* σύστημα που χρησιμοποιεί τα *FAST* [28] χαρακτηριστικά για να εκτιμήσει, να βελτιώσει την τροχιά του αλλά και να ανακατασκευάσει το περιβάλλον που παρατηρεί. Η αρχιτεκτονική του συστήματος φαίνεται στην εικ.5.1. Το σύστημα υποστηρίζει



Σχήμα 5.1: Επισκόπηση ORB-SLAM2 συστήματος, παρουσιάζοντας όλα τα βήματα από την εκτίμηση τροχιάς μέχρι της τοπικής χαρτογράφησης και κλεισίματος βρόχου. Εμφανίζονται, επίσης τα κύρια στοιχεία της αναγνώρισης περιοχής (Place Recognition) και του χάρτη (Map) [23].

πολλά είδη αισθητήρων όπως μια κάμερα, δυο κάμερες *RGB – D* κάμερα και *IMU*, από τους οποίους μας ενδιαφέρει η επιλογή της μιας κάμερας. Το σύστημα χωρίζεται σε τρία νήματα το νήμα της παρακολούθησης, το νήμα του τοπικού χάρτη και το νήμα του χάρτη. Επίσης, τα *FAST* [28] χαρακτηριστικά που χρησιμοποιεί είναι στην ουσία γωνίες, περιγραφόμενες από τον *ORB* [29] περιγραφητή. Το νήμα, της παρακολούθησης είναι υπεύθυνο για το ταίριασμα αυτών των χαρακτηριστικών και την εκτίμηση, παρακολούθηση και τοπική βελτιστοποίηση της τροχιάς του ρομπότ, καθώς και της απόφασης για εισαγωγή καινούριων *keyframe* στο σύστημα. Ενώ, το νήμα του τοπικού χάρτη είναι υπεύθυνο για τη βελτιστοποίηση της τροχιάς και τη διαχείριση εισαγωγής ή διαγραφής καινούριας πληροφορίας στο σύστημα καθώς και την εύρεση κλεισίματος βρόχου. Τέλος, το νήμα του χάρτη είναι υπεύθυνο για τη διαχείριση



Σχήμα 5.2: Ανίχνευση ORB χαρακτηριστικών από το ORB-SLAM2 συστήματα [23].

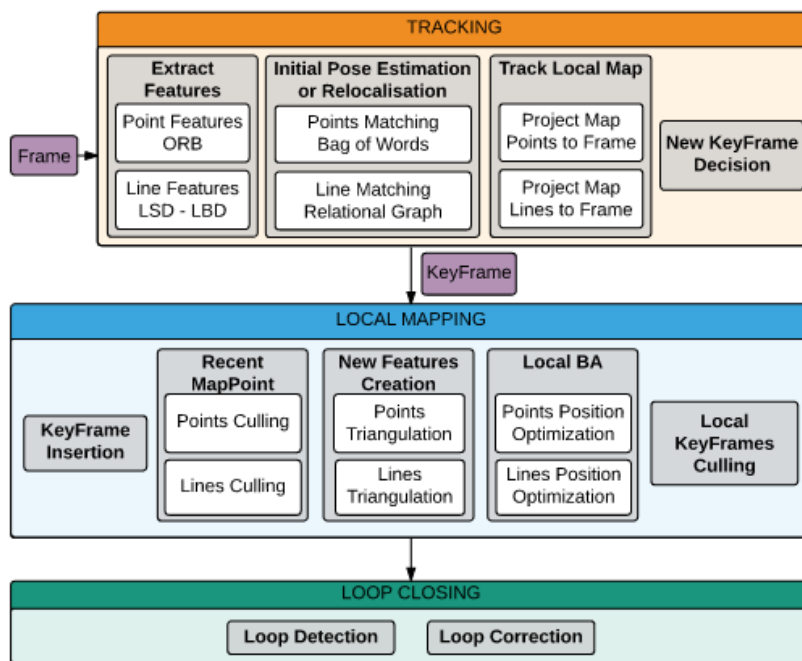
του χάρτη ανακατασκευής και τη δημιουργία του δυο γράφων πόζας τον *Covisibility Graph* που αποτελεί έναν γράφο που συνδέει όλα τα *frames* με εκείνα τα *frames* που μοιράζονται κοινά χαρακτηριστικά (ORB [29] χαρακτηριστικά) και του *Essential Graph* που είναι το ελάχιστο συνδετικό δένδρο του *Covisibility Graph* παρέχοντας συνδέσεις και πληροφορία που είναι πιο χρήσιμη και ελάχιστη σε αριθμό ώστε το σύστημα να παραμένει αποδοτικό, τόσο σε χρόνο όσο και σε ακρίβεια. Επίσης, το σύστημα είναι εφοδιασμένο με το *Place Detection* σύστημα, που παρέχει τα *visual bag of words* που υλοποιούνται στον *DBoW3* [72] αλγόριθμο, σε κάθε *frame* ώστε να μπορεί το σύστημα να αναγνωρίσει την ύπαρξη βρόχου αλλά και την εκτίμηση της πόζας μέσα από ταίριασμα *visual bag of words* στην περίπτωση που το σύστημα, αδυνατίσει να παρακολουθήσει την τροχιά.

PL-SLAM [24].

Σε αντίθεση με το *ORB-SLAM* [23] το *PL-SLAM* [24] βασίζεται στο πρώτο μόνο όμως για μια κάμερα και το επεκτείνει χρησιμοποιώντας τον αλγόριθμο *LSD* [58] για να εξάγει σε $O(n)$ χρόνο, ευθείες, καθώς και το *LBD* [73] περιγραφητή για την περιγραφή τους. Επίσης, για τη βελτιστοποίηση της πόζας στο γράφο εισάγει το σφάλμα (εξ.4.31) και χρησιμοποιεί θεωρία τριών όψεων από το [15] και την πληροφορία των ευθειών για τη δημιουργία και εκτέλεση διαφορετικής μεθόδου αρχικοποίησης του συστήματος. Τέλος, γίνεται η χρήση του αλγορίθμου *EPnPL* [74] που αποτελεί μια παραλλαγή του [70] αλγορίθμου, που χρησιμοποιεί την πληροφορία των ευθειών για να εκτιμήσει την πόζα του ρομπότ στις περιπτώσεις που χάνεται η παρακολούθηση της τροχιάς και θέλουμε γρήγορα να εκτιμήσει το σύστημα την πόζα του ώστε να συνεχίσει να κινείται αποτελεσματικά.

5.3 Υλοποίηση

Στην ενότητα αυτή θα αναφερθούμε στην υλοποίηση του συστήματος μας. Το *ORB-SLAM2* [23] σύστημα αποτελεί ένα από τα καλύτερα *real-time* συστήματα που λύνουν το πρόβλημα του *SLAM* χρησιμοποιώντας γωνίες (ORB [29] χαρακτηριστικά), αποδοτικά και με καλή ακρίβεια σε διάφορα περιβάλλοντα. Ωστόσο, επειδή χρησιμοποιεί μόνο σημεία για την εκτίμηση και βελτιστοποίηση της πόζας όπως και την ανακατασκευή του χώρου, δεν



Σχήμα 5.3: Επισκόπηση PL-SLAM συστήματος [24].

αξιοποιεί όλη την πληροφορία του περιβάλλοντος οδηγώντας το κάποιες φορές σε μειωμένη ακρίβεια στην εκτίμηση της πόζας του ρομπότ. Ακόμη, ο χάρτης που ανακατασκευάζεται είναι αραιός και δεν προσφέρει πολύ πληροφορία ώστε να μπορέσει να αξιοποιηθεί από συστήματα που εκτελούν σημασιολογική ταξινόμηση του χάρτη, προσφέροντας έναν πιο πληθωρικό χάρτη. Σε αντίθεση, το *PL-SLAM* [24] αξιοποιεί την παραπάνω πληροφορία που παρουσιάζεται στο περιβάλλον, ανιχνεύοντας ευθείες και χρησιμοποιεί αυτή την πληροφορία για να βελτιστοποιήσει την τροχιά και να αρχικοποιήσει το σύστημα. Ωστόσο, δε χρησιμοποιεί τις ευθείες ώστε να εκτιμήσει την τροχιά του ρομπότ. Έτσι λοιπόν, στην παρούσα εργασία στηριχθήκαμε στο *ORB-SLAM2* [23] για την περίπτωση μιας κάμερας και ασχοληθήκαμε με ιδέες που χρησιμοποιούν την πληροφορία των ευθειών για την ανίχνευση σημείων φυγής σε $O(n)$ χρόνο, ώστε να εκτιμήσουμε και να βελτιώσουμε την τροχιά του ρομπότ. Συγκεκριμένα ασχοληθήκαμε με τα εξής:

- Εισαγωγή του [18] αλγόριθμου για την εξαγωγή και ανίχνευση σημείων φυγής από μια εικόνα, χρησιμοποιώντας τις εξαγόμενες ευθείες από τον *LSD* [58].
- Εκτίμηση της περιστροφής του ρομπότ χρησιμοποιώντας τα σημεία φυγής.
- Επέκταση του [24] εισάγοντας τμήμα βελτιστοποίησης της περιστροφής του ρομπότ χρησιμοποιώντας τα σημεία φυγής.
- Επέκταση του [24] εισάγοντας τμήμα εκτίμησης της μετατόπισης του ρομπότ βασισμένο σε γραμμικό σύστημα.

Το *ORB-SLAM2* [23] είναι υλοποιημένο στη *C++* γλώσσα και χρησιμοποιεί τη *OpenCV* [75] βιβλιοθήκη για χρήση των *ORB* [29] περιγραφητή και για την αξιοποίηση δομές δεδομένων για την αναπαράσταση της πόζας. Επίσης, χρησιμοποιεί τη *Eigen* [76] βιβλιοθήκη για τη

χρήση μεθόδων γραμμικής άλγεβρας και τη *Pangolin* [77] βιβλιοθήκη για την οπτικοποίηση του χάρτη και της πόζας. Στο διαδίκτυο είναι διαθέσιμος μόνο ο κώδικας για το *ORB-SLAM2* [23]¹. Ενώ για το *PL-SLAM* [24] δεν υπάρχει διαθέσιμος κώδικας. Ο κώδικας μας έχει βασιστεί στον διαθέσιμο κώδικα² που είναι μια καλή προσέγγιση του *PL-SLAM* [24].

Η δουλειά που έχει γίνει σε αυτή τη διπλωματική είναι στο τμήμα της παρακολούθησης (Tracking module) του *PL-SLAM* [24]. Το *PL-SLAM* [24] λειτουργεί ως εξής, αφού λάβει την εικόνα και εξάγει τα απαραίτητα χαρακτηριστικά (ORB [29], LSD [58]), εκτελεί ταίριασμα χαρακτηριστικών και έπειτα χρησιμοποιεί την υπόθεση, ότι η κίνηση του ρομποτή είναι, ομαλή έχοντας σταθερή ταχύτητα με συνέπεια την επόμενη στιγμή η εκτίμηση της πόζας να προέρχεται από το μοντέλο σταθερής ταχύτητας. Στη συνέχεια, βελτιστοποιεί την πόζα στο εκάστοτε *frame* και συνεχίζει με τις λειτουργίες της ανανέωσης και παρακολούθησης του χάρτη. Λεπτομερειακά, το κομμάτι της παρακολούθησης (*Tracking*) του *PL-SLAM* σκιαγραφείται ως εξής 5.4: Ενώ η αρχιτεκτονική που προτείνουμε είναι 5.5:

Παρακάτω αναλύονται όλα τα τμήματα που έχουν χρησιμοποιηθεί σε αυτήν την εργασία.

5.3.1 Εξαγωγή Σημείων Φυγής (VP extraction)

Ο αλγόριθμος που χρησιμοποιούμε για την εξαγωγή των σημείων φυγής είναι ο [18] που έχουμε αναφερθεί εκτενώς στο κεφάλαιο 3, με $O(n)$ χρόνο εκτέλεσης (βλ.πίνακα 6.2), επιθυμητό για *SLAM* συστήματα που θέλουμε να εκτελούνται σε *real time*. Η είσοδος του είναι οι εξαγόμενες ευθείες από τον *LSD* [58] ανιχνευτή και υπολογίζει βέλτιστα 3 ορθοκανονικά σημεία φυγής στην εικόνα.

5.3.2 Εκτίμηση Περιστροφής (Rotation Estimation)

Στην περίπτωση που μπορούμε να υπολογίσουμε 3 ορθοκανονικά σημεία φυγής, μπορούμε να αξιοποιήσουμε αυτήν την πληροφορία για να υπολογίσουμε την περιστροφή του τωρινού *frame* με βάση τον τύπο 4.18. Έχοντας εκτιμήσει την περιστροφή, εκτιμούμε τη μετατόπιση του τωρινού *frame* με βάση το μοντέλο σταθερής ταχύτητας.

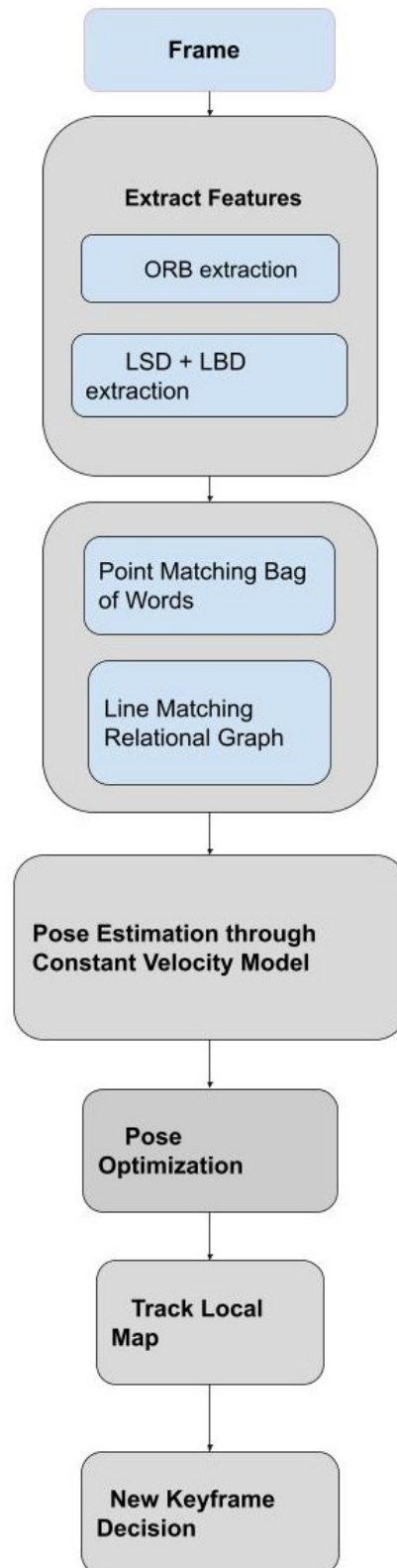
5.3.3 Βελτιστοποίηση Περιστροφής (Rotation Refinement)

Έχοντας εκτιμήσει την περιστροφή στο τωρινό *frame* και έχοντας στη διάθεση μας 3 κάθετα σημεία φυγής, μπορούμε να κατασκευάσουμε ένα σχήμα βελτιστοποίησης της περιστροφής, εκμεταλλευόμενοι αυτές τις πληροφορίες, που θα οδηγήσει το σύστημα σε μια καλύτερη βελτίωση της τροχιάς του συστήματος. Έτσι, λοιπόν, δημιουργήσαμε από την αρχή τον παρακάτω αλγόριθμο [45] βελτιστοποίησης της περιστροφής αξιοποιώντας τα σημεία φυγής. Προτού, συνεχίσουμε εισάγουμε τα απαραίτητα μαθηματικά σύμβολα και ερμηνείες για την καλύτερη κατανόηση του αλγορίθμου.

Έστω i το i -οστό *frame* και έστω $\mathbf{vp}^i = [\mathbf{vp}_1^i | \mathbf{vp}_2^i | \mathbf{vp}_3^i]$ τα 3 σημεία φυγής του και έστω $\mathbf{d}^i = [\mathbf{d}_1^i | \mathbf{d}_2^i | \mathbf{d}_3^i]$ οι αντίστοιχες κατευθύνσεις των σημείων φυγής (βλ.εξ.3.5). Από το [15] έχουμε ότι οι κατευθύνσεις των σημείων φυγής από *frame* σε *frame* εξαρτώνται μόνο από τη

¹Διαθέσιμος κώδικας: https://github.com/raulmur/ORB_SLAM2

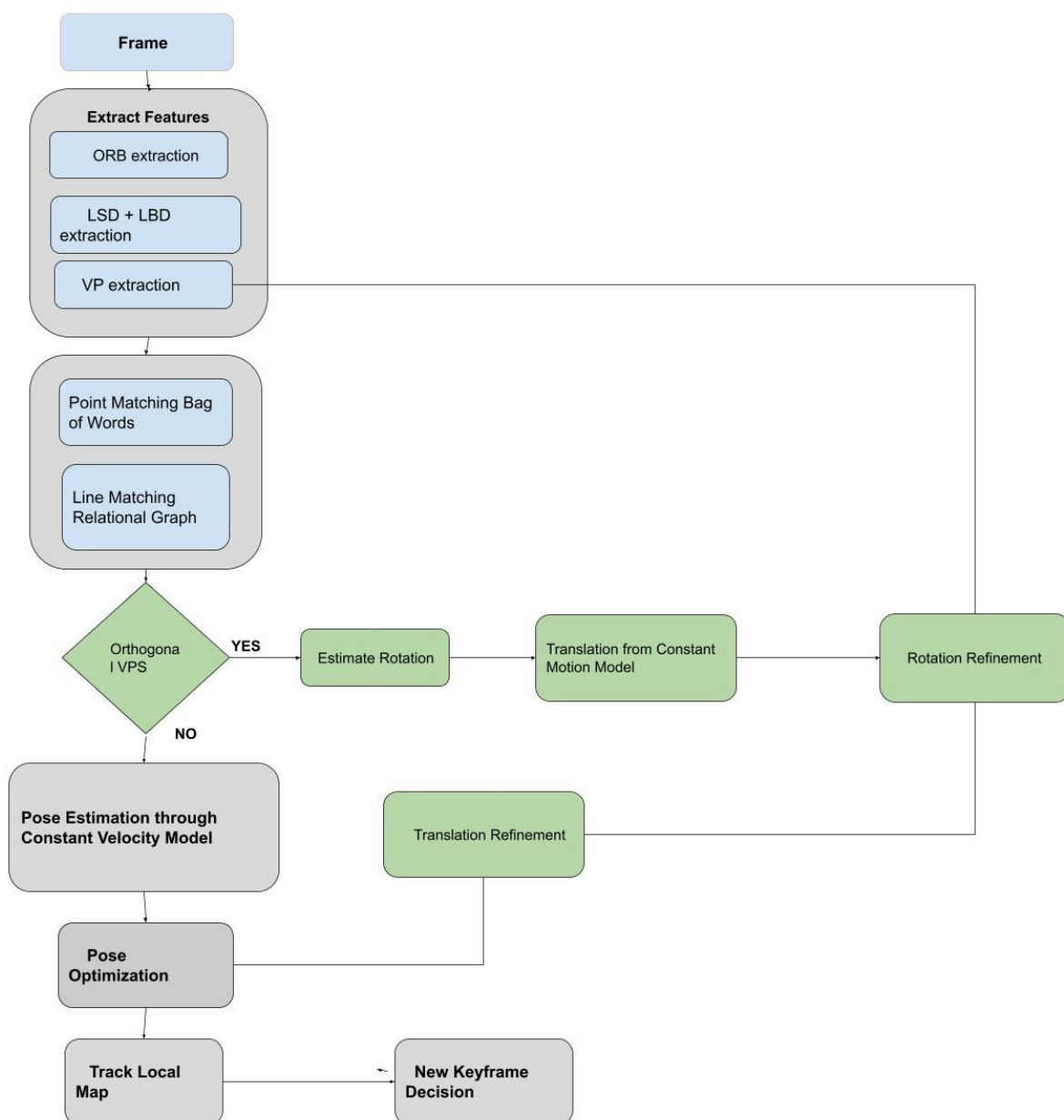
²Διαθέσιμος κώδικας: <https://github.com/HarborC/PL-SLAM>



Σχήμα 5.4: Επισκόπηση Tracking module του PL-SLAM συστήματος [24].

σχετική περιστροφή \mathbf{R} τους, δηλαδή [15]:

$$\mathbf{d}_k^{i+1} = \mathbf{R}d_k^i, k = 1, 2, 3 \quad (5.1)$$



Σχήμα 5.5: Επισκόπηση Αρχιτεκτονικής μας. Με πράσινο είναι τα τμήματα που έχουμε εισάγει.

Έστω ότι \mathbf{d}^0 είναι οι κύριες κατευθύνσεις των σημείων φυγής του αρχικού *frame*. Με βάση τον παραπάνω τύπο 5.1 τα σημεία φυγής στο i -οστό *frame* σχετίζονται με τις κατευθύνσεις του αρχικού *frame* με τη σχέση:

$$\mathbf{vp}^i \sim \mathbf{K}\mathbf{R}_{i_w}\mathbf{d}^0 \quad (5.2)$$

και άρα οι αντίστοιχες κατευθύνσεις τους θα δίνεται:

$$\mathbf{d}_k^i \sim \mathbf{R}_{i_w}\mathbf{d}_k^0, k = 1, 2, 3 \quad (5.3)$$

Ωστόσο, έχοντας υπολογίσει τα σημεία φυγής με τον [45] αλγόριθμο, οι κατευθύνσεις τους θα

δίνονται από την εξ.3.5 δηλαδή:

$$\delta_{\mathbf{k}}^i = \mathbf{d}_{\mathbf{k}}^i = K^{-1} \mathbf{v} \mathbf{p}_{\mathbf{k}}^i, k = 1, 2, 3 \quad (5.4)$$

Έτσι λοιπόν, παρατηρούμε ότι οι δυο τύποι εξ.5.2, εξ.5.4, δεν πρέπει να διαφέρουν, ωστόσο, λόγω θορύβου θα παρατηρείται απόκλιση μεταξύ αυτών των δύο σχέσεων Λαμβάνοντας υπόψιν αυτό μπορούμε να κατασκευάσουμε τον παρακάτω αλγόριθμο βελτιστοποίησης της περιστροφής. Πιο αναλυτικά, η συνάρτηση προς βελτιστοποίηση [45] είναι:

$$E(\omega_i) = \sum_{k=1}^3 E_k(\omega_i) = \sum_{k=1}^3 \arccos(\delta_{\mathbf{k}}^i \cdot \mathbf{R}_{i\mathbf{w}} \mathbf{d}_{\mathbf{k}0}) \quad (5.5)$$

Με αντίστοιχο Ιακωβιανό πίνακα:

$$\begin{aligned} \mathbf{J}_{\mathbf{k}} &= -\frac{1}{\sqrt{1 - (\delta_{\mathbf{k}}^i \cdot \mathbf{d}_{\mathbf{k}})^2}} \cdot \delta_{\mathbf{k}}^i \frac{\partial \mathbf{d}_{\mathbf{k}}}{\partial \omega_i} \\ &= \frac{1}{\sqrt{1 - (\delta_{\mathbf{k}}^i \cdot \mathbf{d}_{\mathbf{k}})^2}} \cdot \delta_{\mathbf{k}}^i \cdot ([\mathbf{d}_{\mathbf{k}}]_x), \end{aligned} \quad (5.6)$$

Ο αλγόριθμος που δημιουργήσαμε και χρησιμοποιήσαμε για τη βελτιστοποίηση της εξ.5.5 ήταν ο *LM* αλγόριθμος 2.29. Διαισθητικά, αυτό που αντιλαμβανόμαστε από τον ανωτέρω αλγόριθμο είναι ότι, εάν λάβουμε τις κατευθύνσεις των σημείων φυγής της αρχικής κάμερας και τις περιστρέψουμε με βάση τον προσανατολισμό της κ-οστής κάμερας θα παράξουν κάποιες κατευθύνσεις σημείων φυγής. Αυτές, οι κατευθύνσεις των σημείων, με τις αντίστοιχες κατευθύνσεις των σημείων φυγής, που έχουν παραχθεί από τον αλγόριθμο θα πρέπει να έχουν πολύ μικρή έως μηδενική διαφορά γωνίας μεταξύ τους. Επομένως, ελαχιστοποιώντας αυτή τη μετρική, βελτιστοποιούμε την περιστροφή.

5.3.4 Εκτίμηση Μετατόπισης (Translation Estimation)

Έχοντας βελτιστοποιήσει την ολική περιστροφή $\mathbf{R}_{i\mathbf{w}}$ του *i*-οστού frame από το προηγούμενο βήμα, μπορούμε να εκμεταλλευτούμε αυτή την πληροφορία για να εκτιμήσουμε καλύτερα τη μετατόπιση της κάμερα λύνοντας το γραμμικό σύστημα [46], χρησιμοποιώντας τον *RANSAC* αλγόριθμο [78]. Συγκεκριμένα, έχοντας γνωστά τα σημεία του χάρτη $\mathbf{X}_{\mathbf{j}}$ που αντιστοιχίζονται στο *i*-οστού frame, λαμβάνουμε το σφάλμα επαναπροβολής:

$$E_p = \sum_{j=1}^N \rho(\|\mathbf{x}_{\mathbf{j}} - \pi(\mathbf{X}_{\mathbf{j}}, \mathbf{R}_{i\mathbf{w}}, \mathbf{t}_{i\mathbf{w}}, K)\|^2) \quad (5.7)$$

Από αυτή τη σχέση έχοντας γνωστή την περιστροφή $\mathbf{R}_{i\mathbf{w}}$ και γράφοντας αναλυτικά την εξ.5.7, έχουμε:

$$\mathbf{t}_{i\mathbf{w}} = ((\mathbf{R}_{i\mathbf{w}} \cdot \mathbf{X}_{\mathbf{j}})^{(3)} + t^{(3)}) \cdot \begin{bmatrix} \frac{u_j - c_x}{f_x} \\ \frac{v_j - c_y}{f_y} \end{bmatrix} - \begin{bmatrix} (\mathbf{R}_{i\mathbf{w}} \cdot \mathbf{X}_{\mathbf{j}})^{(1)} + t^{(1)} \\ (\mathbf{R}_{i\mathbf{w}} \cdot \mathbf{X}_{\mathbf{j}})^{(2)} + t^{(2)} \end{bmatrix}, \quad (5.8)$$

Όπου $[\]^{(j)}$ είναι η j -οστη γραμμή του πίνακα. Αναδιατάσσοντας τη (5.8) θεωρώντας ως άγνωστο το διάνυσμα μετατόπισης \mathbf{t}_i οδηγήμασε στην ελαχιστοποίηση του συστήματος:

$$\mathbf{t}_{iw} = \min(\|\mathbf{A}\mathbf{t} - \mathbf{b}\|^2) \quad (5.9)$$

όπου :

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & \frac{u_i - c_x}{f_x} \\ 0 & -1 & \frac{v_i - c_y}{f_y} \end{bmatrix} \quad (5.10)$$

και

$$\mathbf{b} = \begin{bmatrix} (\mathbf{R}_{iw}\mathbf{X}_j)^{(1)} - (\mathbf{R}_{iw}\mathbf{X}_j)^{(3)} \frac{u_j - c_x}{f_x} \\ (\mathbf{R}_{iw}\mathbf{X}_j)^{(2)} - (\mathbf{R}_{iw}\mathbf{X}_j)^{(3)} \frac{v_j - c_y}{f_y} \end{bmatrix} \quad (5.11)$$

Παρατηρούμε ότι η εξ.5.9 αποτελεί ένα πρόβλημα ελαχιστοποίησης τετραγώνων που μπορεί να λυθεί με τον RANSAC αλγόριθμο, ώστε να διαχειριστούμε καλύτερα τα *outliers* που μπορεί να έχει το σύστημα. Η λύση του συστήματος δίνεται από την εξίσωση:

$$\mathbf{A}^T \mathbf{A} \mathbf{t} = \mathbf{A} \mathbf{b} \quad (5.12)$$

Κεφάλαιο 6

Πειράματα

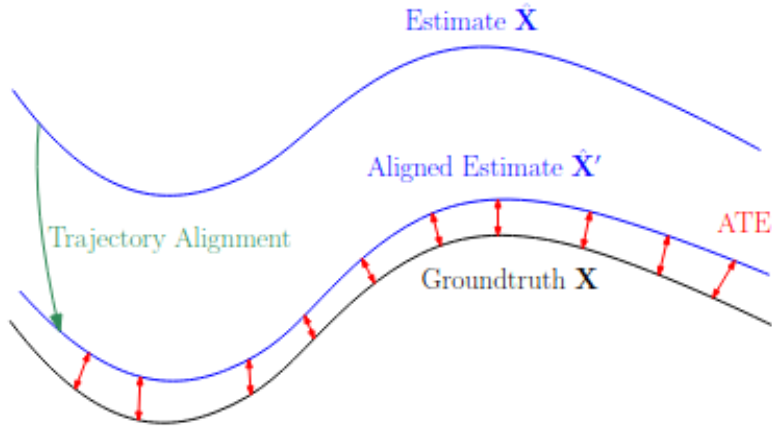
Στο κεφάλαιο αυτό θα ασχοληθούμε και θα σχολιάσουμε τα αποτελέσματα, που πέτυχε η αρχιτεκτονική του συστήματος μας τόσο ως προς το *ORB-SLAM2* [23], όσο και ως προς το *PL-SLAM* [24]. Έχοντας δημιουργήσει ένα μοντέλο συστήματος που λύνει το πρόβλημα του *SLAM*, τα ερωτήματα που ανακύπτουν είναι: Ποια είναι η μετρική στην εστιάζουμε να βελτιστοποιήσουμε; Η μετρική αυτή είναι ο χρόνος εκτέλεσης του μοντέλου μας; Η ακρίβεια της τροχιάς σε μέτρα και αν ναι υπάρχει η αληθινή τροχιά ως προς την οποία θέλουμε να συγκριθούμε; Ή μας ενδιαφέρουν σημασιολογικά ερωτήματα όπως αν μπορεί το μοντέλο μας να αναγνωρίσει τα αντικείμενα του ανακατασκευασμένου χάρτη. Έτσι, λοιπόν όπως διαφαίνεται θα πρέπει να έχουμε αποφασίσει ως προς ποιες μετρικές θέλουμε να ελέγξουμε το σύστημα μας, καθώς δεν είναι μια εύκολη διαδικασία και ανάλογα την περίπτωση να μην υπάρχουν πολλά διαθέσιμα πακέτα για τον έλεγχο του μοντέλου μας. Σε αυτήν, τη διπλωματική, το ερώτημα που προσπαθούμε να λύσουμε είναι κατά πόσο η μέθοδος μας μπορεί να πετύχει καλά αποτελέσματα σε επίπεδο σφάλματος τροχιάς ως προς κάποια αληθινή (*ground-truth*) τροχιά, διατηρώντας το υπολογιστικό κόστος κοντά σε *real-time*.

6.1 Μέθοδος εκτίμησης Σφάλματος Τροχιάς

Όπως έχουμε δηλώσει και στα προηγούμενα κεφάλαια, σε αυτή τη διπλωματική έχουμε επικεντρωθεί στο πρόβλημα της εκτίμησης και βελτιστοποίησης της πόζας. Έτσι λοιπόν το είδος της μετρικής που μας απασχολεί είναι η απόσταση που έχει η εκτιμώμενη τροχιά από την αληθινή τροχιά (*ground-truth GT*). Δηλαδή το σφάλμα που μας ενδιαφέρει είναι το *Absolute Trajectory Error (ATE)* (βλ.σχήμα.6.1). Όπως παρατηρούμε και από το σχήμα.6.1 για να υπολογιστεί το (*ATE*) πρέπει πρώτα να ευθυγραμμιστεί βέλτιστα η εκτιμώμενη τροχιά ως προς την αληθινή (*GT*) ώστε να μπορούν να υπολογιστούν με μοναδικό τρόπο τα σφάλματα. Στη συνέχεια υπολογίζεται το μέσο τετραγωνικό σφάλμα (*Root Mean Square Error RMSE*) των μετατοπίσεων, ώστε να μετατραπεί το σφάλμα σε αριθμό. Δηλαδή έστω ότι $E_{1:n}$ είναι η εκτιμώμενη τροχιά και $GT_{1:n}$ είναι η αληθινή τροχιά και S είναι ο πίνακας μετασχηματισμού που ευθυγραμμίζει βέλτιστα τις δυο τροχιές [79], άρα το *ATE* για κάποια χρονική στιγμή i θα δίνεται από τη σχέση [80]:

$$ATE_i = GT_i^{-1}SE_i \quad (6.1)$$

$$RMSE(ATE_{1:n}) = \left(\frac{1}{n} \sum_{i=1}^n \|\text{trans}(ATE_i)\|^2\right)^{1/2} \quad (6.2)$$



Σχήμα 6.1: Σφάλμα Απόλυτης τροχιάς (ATE) μεταξύ GT τροχιάς και εκτιμώμενης [25].

Στην περίπτωση που έχουμε μονοσκοπική κάμερα θα πρέπει να υπολογιστεί και η κλίμακα στην οποία πρέπει ευθυγραμμίσουμε τη μια τροχιά ως προς την άλλη ώστε να λάβουμε το σωστό σφάλμα. Στο σχήμα.6.2 παρατηρούμε τον αλγόριθμο που υπολογίζει τον μετασχηματισμό για ευθυγράμμιση των δύο τροχιών, υπολογίζοντας τα στατιστικά χαρακτηριστικά των δύο τροχιών και βελτιστοποιώντας ένα τετραγωνικό σφάλμα, υπολογίζει την επιθυμητή ευθυγράμμιση μαζί με την κλίμακα την οποία πρέπει να πολλαπλασιαστεί η μια τροχιά ώστε να υπολογιστεί βέλτιστα το σφάλμα (ATE).

Algorithm 1: Closed-form solution to (13)

Data: estimation $\{\hat{\mathbf{p}}_i\}_{i=0}^{N-1}$, groundtruth $\{\mathbf{p}_i\}_{i=0}^{N-1}$

Result: $s, \mathbf{R}, \mathbf{t}$ that minimize $\sum_{i=0}^{N-1} \|\mathbf{p}_i - s\mathbf{R}\hat{\mathbf{p}}_i - \mathbf{t}\|^2$

- 1 Calculate: $\boldsymbol{\mu}_{\mathbf{p}} = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{p}_i$ $\boldsymbol{\mu}_{\hat{\mathbf{p}}} = \frac{1}{N} \sum_{i=0}^{N-1} \hat{\mathbf{p}}_i$
 $\boldsymbol{\sigma}_{\mathbf{p}}^2 = \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{p}_i - \boldsymbol{\mu}_{\mathbf{p}}\|^2$ $\boldsymbol{\sigma}_{\hat{\mathbf{p}}}^2 = \frac{1}{N} \sum_{i=0}^{N-1} \|\hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}}\|^2$
 $\boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=0}^{N-1} (\mathbf{p}_i - \boldsymbol{\mu}_{\mathbf{p}})(\hat{\mathbf{p}}_i - \boldsymbol{\mu}_{\hat{\mathbf{p}}})^\top$
 - 2 Singular value decomposition: $\boldsymbol{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$
 - 3 **if** $\det(\mathbf{U})\det(\mathbf{V}) < 0$ **then**
 - 4 | $\mathbf{W} = \text{diag}(1, 1, -1)$
 - 5 **else**
 - 6 | $\mathbf{W} = \mathbf{I}_{3 \times 3}$
 - 7 **end**
 - 8 $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^\top$
 - 9 $s = \frac{1}{\boldsymbol{\sigma}_{\hat{\mathbf{p}}}^2} \text{trace}(\mathbf{D}\mathbf{W})$ or $s = 1$ if the **scale** is known
 - 10 $\mathbf{t} = \boldsymbol{\mu}_{\mathbf{p}} - s\mathbf{R}\boldsymbol{\mu}_{\hat{\mathbf{p}}}$
-

Σχήμα 6.2: Αλγόριθμος εκτίμησης ευθυγράμμισης τροχιών [25].

6.2 Σύνολα Δεδομένων

Στην ενότητα αυτή θα παρουσιάσουμε, χρήσιμα σύνολα δεδομένων, που υπάρχουν στην ερευνητική κοινότητα, για την εκτέλεση και έλεγχο των μοντέλων που λύνουν το πρόβλημα του *SLAM*. Πιο αναλυτικά:

Το **TUM RGB-D**[80] είναι ένα μεγάλο σύνολο δεδομένων που περιέχει *RGB-D* δεδομένα μαζί με τις *ground-truth* πόζες κάμερας. Το σύνολο δεδομένων *TUM* περιέχει εικόνες *RGB* και εικόνες βάθους από έναν αισθητήρα *Microsoft Kinect*. Τα δεδομένα έχουν καταγραφεί σε ρυθμό (30 *fps*) και ανάλυση αισθητήρα 640×480 . Η αληθινή τροχιά αποκτήθηκε από ένα σύστημα λήψης κίνησης υψηλής ακρίβειας με οκτώ κάμερες παρακολούθησης υψηλής ταχύτητας (100 *Hz*). Επίσης, περιέχει περιβάλλοντα από εσωτερικούς χώρους γραφείων, βιομηχανικών εγκαταστάσεων αλλά και περιβάλλοντα με κινούμενα αντικείμενα. Εμείς ασχοληθήκαμε με δεδομένα εσωτερικών χώρων που παρουσιάζουν γεωμετρικά χαρακτηριστικά όπως γραφεία, οθόνες υπολογιστές και βιβλιοθήκες. Σε αυτή τη διπλωματική, χρησιμοποιήσαμε αυτό το σύνολο δεδομένων γιατί ικανοποιεί τα εξής κριτήρια:

- Παρέχει βίντεο τόσο για μια κάμερα όσο και για άλλους αισθητήρες.
- Εμπεριέχει σκηνές από πραγματικά και ποικίλα περιβάλλοντα.
- Περιέχει τις αληθινές τροχιές (*ground-truth*) για κάθε βίντεο.
- Περιέχει το, πίνακα *K* κάθε κάμερας για *Calibration*.
- Προσφέρει αρχεία για εύκολη σύγκριση των αποτελεσμάτων [81].

Το **EuRoC**[82] αποτελείται από σκηνές που έχουν ληφθεί από ένα *Micro Aerial Vehicle (MAV)* που φέρει δύο κάμερες και έναν γυροσκόπιο (*IMU*) και κινείται σε βιομηχανικές εγκαταστάσεις. Παρέχει και αυτό τις παραμέτρους της κάμερας, όπως και τις αληθινές τροχιές και μπορεί να χρησιμοποιηθεί σε μοντέλα μιας ή δυο καμερών. Η χρήση του είναι πολύ απλή, αρκεί να δοθούν οι σχετικές διευθύνσεις που είναι αποθηκευμένα τα δεδομένα.

Το **ICL-NUIM**[83], είναι σχετικά απλό στη δομή και περιέχει τεχνητές σκηνές από εσωτερικούς χώρους, με ή χωρίς θόρυβο. Παρέχει επίσης τις αληθινές τροχιές και είναι παρέχει και τις αντίστοιχες εικόνες βάθους, δίνοντας τη δυνατότητα να χρησιμοποιηθούν σε αλγόριθμους με μια ή με *RGB-D* κάμερα. Επίσης, παρέχει ακολουθίες εικόνων που περιέχουν βρόχους, για την καλύτερη αξιολόγηση του συστήματος σε καταστάσεις συσσωρευμένου σφάλματος.

Το **KITTI** [84] περιέχει σκηνές που έχουν ληφθεί με δύο κάμερες, και έχουν καταγραφεί από ένα αυτοκίνητο σε αστικά περιβάλλοντα και σε αυτοκινητοδρόμους. Η απόσταση των καμερών είναι ~ 54 *cm* και λειτουργεί στα 10 *Hz* με ανάλυση μετά από διόρθωση στα 1240×376 *pixel* και περιέχει και αυτό βρόχους για την καλύτερη αξιολόγηση.

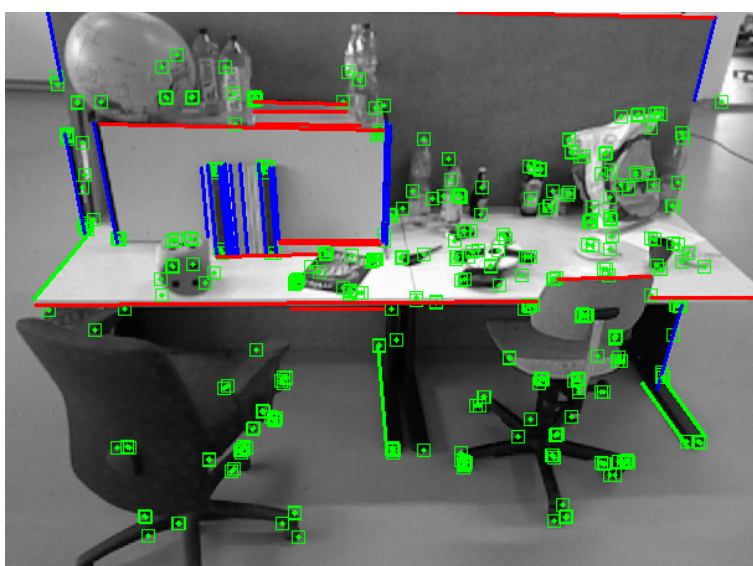
6.3 Αποτελέσματα

Σε αυτή την υποενότητα, αναφέρουμε τα αποτελέσματα που έφερε η αρχιτεκτονική μας σε σχέση με το *ORB-SLAM2* [23] και το *PL-SLAM* [24], στο *ATE RMSE* εκφρασμένο σε μέτρα. Επίσης, παρουσιάζουμε και τα αποτελέσματα των τροχιών που παρήχθησαν από το σύνολο

των δεδομένων 6.7. Η εκτέλεση των πειραμάτων έγινε στο *TUM – RGBD dataset* [80] και για την εξαγωγή των σφαλμάτων χρησιμοποιήθηκαν οι κώδικες που παρέχει το *dataset* [81].

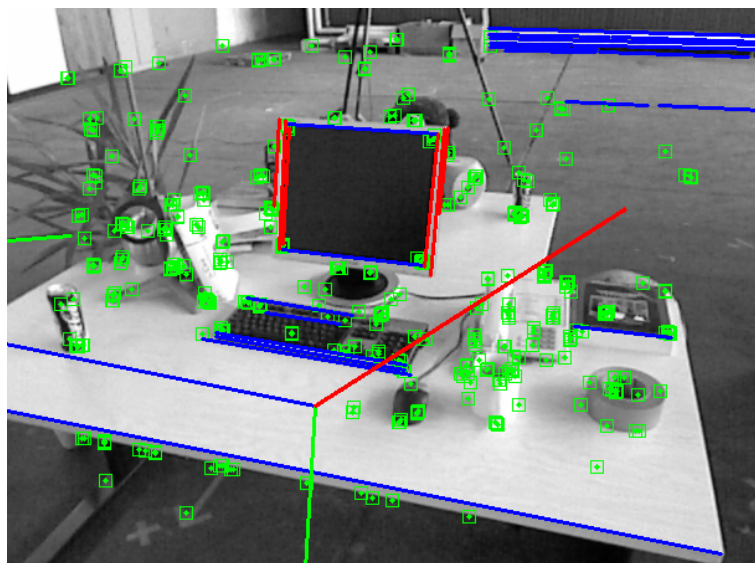


Σχήμα 6.3: Στιγμιότυπο εικόνας από το *TUM fr3-large-cabinet*.

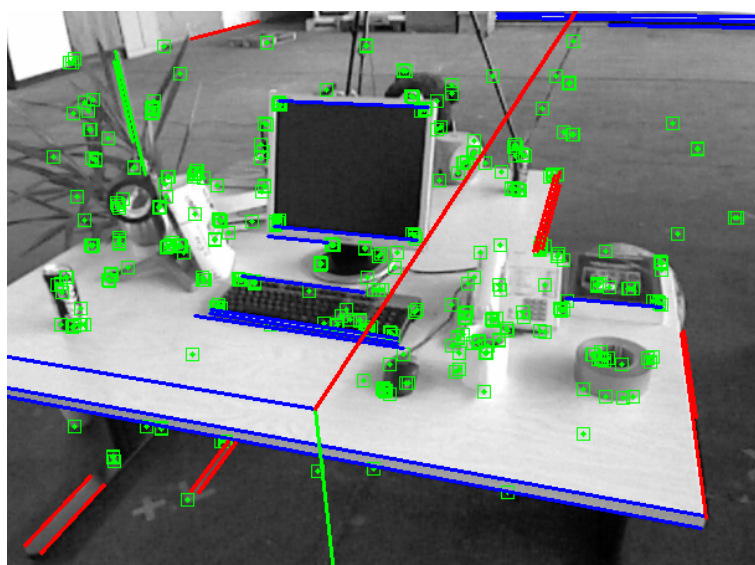


Σχήμα 6.4: Με κόκκινο, πράσινο και μπλε χρώμα είναι οι εικόνες που αντιστοιχούν στα σημεία φυγής που έχει εξαγάγει ο αλγόριθμος που χρησιμοποιούμε στην αρχιτεκτονική μας στο *TUM fr3-structure-texture-far*.

Επίσης, εκτός από τα αποτελέσματα παρουσιάζουμε και τις ευθείες που έχει ταξινομήσει ο αλγόριθμος εξαγωγής των σημείων φυγής και αντιστοιχούν σε αυτά (βλ.σχήμα.6.4) με τα διαφορετικά χρώματα να σηματοδοτούν τις ευθείες που έχουν χρησιμοποιηθεί για την εκτίμηση των σημείων φυγής και έχουν ταξινομηθεί στους αντίστοιχους άξονες (XYZ). Αυτή η ταξινόμηση των ευθειών γίνεται, αφού έχουν υπολογιστεί τα σημεία φυγής με τον αλγόριθμο [18]. Ωστόσο, αυτή η ταξινόμηση δεν είναι πάντα σωστή όπως παρατηρούμε και στο σχήμα 6.5. Όπως, βλέπουμε η πράσινη ευθεία που ανήκει στο φύλλο, έχει ταξινομηθεί λανθασμένα, με βάση τις κατευθύνσεις των σημείων φυγής που έχουν υπολογιστεί, με αφετηρία το κέντρο

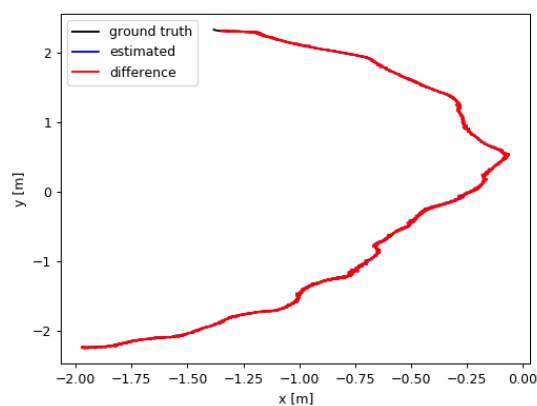


Σχήμα 6.5: Αποτυχία ταξινόμησης ευθείας.

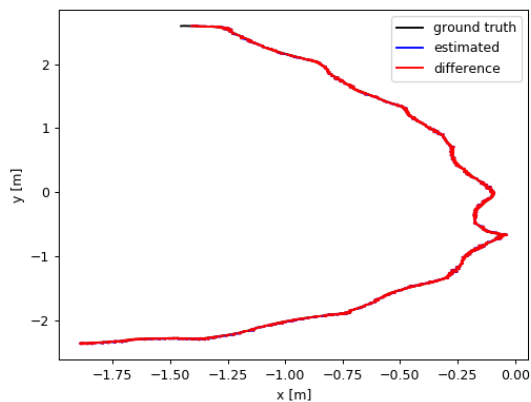


Σχήμα 6.6: Κατευθύνσεις σημείων φυγής με μπλε, κόκκινο και πράσινο χρώμα και ταξινόμηση αντίστοιχων ευθειών

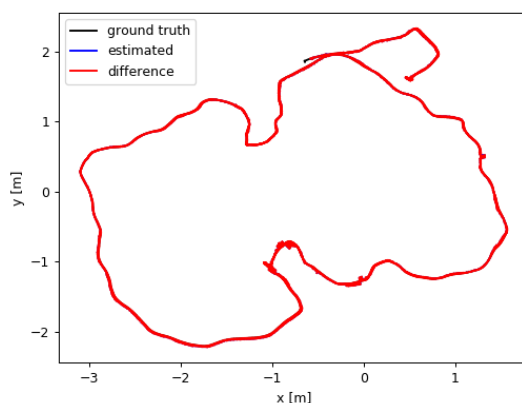
της εικόνας φαίνονται με τα αντίστοιχα χρώματα τους. Αυτό συμβαίνει διότι ο αλγόριθμος για να ταξινομήσει μια ευθεία στο αντίστοιχο σημείο φυγής λαμβάνει το διάνυσμα που ενώνει το κέντρο της ευθείας με το σημείο φυγής και αν αυτό το διάνυσμα έχει γωνία μικρότερη από ένα κατώφλι ως προς την ευθεία τότε την αντιστοιχεί σε αυτό το σημείο φυγής. Επομένως, εξαρτάται πολύ από το κατώφλι και από το εγγενή θόρυβο που έχουν οι μετρήσεις και από το γεγονός ότι αυτή η γωνία υπολογίζεται στο επίπεδο της εικόνας, άρα αυτή η ευθεία στις 3D θα έχει διαφορετική γωνία από ότι στις 2D. Ωστόσο, αυτό το φαινόμενο δεν είναι συχνό. Τέλος, μια καλή περίπτωση ταξινόμησης ευθειών μαζί με τα αντίστοιχα σημεία φυγής σχήμα.6.6.



(α) *TUM fr3/structure-texture-far*



(β) *TUM fr3/structure-texture-far-validation*



(γ) *TUM fr3/long-office-house-validation*

Σχήμα 6.7: Παραπάνω φαίνονται οι τροχιές που παρήγαγε η αρχιτεκτονική μας στο TUM dataset. Με μαύρο είναι η αληθινή τροχιά ενώ με κόκκινο είναι η διαφορά της πραγματικής με την εκτιμώμενη.

6.3.1 Σχολιασμός Αποτελεσμάτων

Όπως αναφέραμε σε περιβάλλοντα που έχουν γεωμετρική πληροφορία πέραν των σημείων η μέθοδος μας έφερε πολύ καλά αποτελέσματα. Για παράδειγμα στις ακολουθίες

Πίνακας 6.1: Αποτελέσματα στο TUM-RGB-D για το RMSE ATE (m)

TUM-RGB-D Sequence	Ours	ORB-SLAM2 [23]	PL-SLAM [24]
fr1/desk	0.032	0.020	0.032
fr1/plant	0.020	0.020	0.097
fr2/desk	0.012	0.009	0.012
fr3/large-cabinet	0.100	1.420	0.100
fr3/long-office-house-validation	0.019	0.010	0.019
fr3/structure-texture-far	0.010	0.012	0.008
fr3/structure-texture-far-near	0.014	0.015	0.014
fr3/nostr-text-near-withLoop	0.177	0.015	0.177
fr3/nostr-text-far	0.029	0.048	0.029

Πίνακας 6.2: Χρόνος Εκτέλεσης των νημάτων Παρακολούθησης και Τοπικής Χαρτογράφησης

Μέσος χρόνος εκτέλεσης σε (ms)				
Thread	Operation	Ours	ORB-SLAM2	PL-SLAM
Tracking	Feature Extraction	96,94	15,79	96,64
	VPs Extraction	46,56	--	--
	Initial Pose Estimation	71,1	49,06	71,1
	Track Local Map	29,93	26,61	29,93
	Total	244,53	91,46	197,97
Local Mapping	Keyframe Insertion	47,90	45,82	47,90
	Map Feature Culling	2,48	1,89	2,48
	Map feature Creation	135,93	21,93	135,93
	Keyframe Culling	72,52	70,07	72,52
	Local BA	1499,37	1434,57	1499,37
	Total	1758,2	1574,28	1711,64

fr1/plant, fr3/large-cabinet, fr3/structure-texture-far, fr3/nostructure-texture-far τα οποία αποτελούνται από αντικείμενα, όπως γραφεία, βιβλιοθήκες, οθόνες υπολογιστών, χαρτιά και καρέκλες, το σύστημα μας αναγνωρίζει με μεγάλη ακρίβεια ευθείες και σημεία φυγής, με αποτέλεσμα να βελτιώνεται η τροχιά αυτών των ακολουθιών. Στην περίπτωση της ακολουθίας *fr3/large – cabinet* (σχήμα.6.3) το συστήμα μας έφερε πολύ πιο καλά αποτελέσματα από το *state of the art*. Αυτό συνέβη γιατί το σύστημα μπορούσε να ανίχνευσει περισσότερες ευθείες από ότι γωνίες καθώς κυριαρχούσαν στην εικόνα και μπορούσε να εκτιμήσει με μεγάλη ακρίβεια τα σημεία φυγής ώστε να αξιοποιήσει αυτή την πληροφορία για βελτιστοποίηση της τροχιάς. Στα σχήματα 6.7 παρουσιάζουμε τις τροχιές που παρήγαγε η αρχιτεκτονική μας στις αντίστοιχες ακολουθίες του TUM. Επίσης, η γνώση τριών ορθοκανονικών σημείων φυγής, που έχουν υπολογιστεί και βέλτιστοι, προσφέρει μια καλή εκτίμηση του προσανατολισμού της κάμερας, με αποτέλεσμα, επειδή η βελτιστοποίηση συναρτήσεων που εμπεριέχουν προσανατολισμούς είναι μη γραμμική και μη κυρτή είναι πολύ εύκολο

να καταλήξει ο αλγόριθμος σε τοπικό μέγιστο. Επίσης, επειδή ο αλγόριθμος που χρησιμοποιείται για την βελτιστοποίηση είναι μια παραλλαγή του Gauss-Newton, αντιλαμβάνομαστε τη βαρύτητα που φέρουν οι αρχικές συνθήκες για την καλύτερη σύγκλιση του αλγορίθμου και την επίτευξη ελαχίστου. Έτσι λοιπόν στις περιπτώσεις που αλγόριθμος έχει υπολογίσει πολύ σωστά τα σημεία φυγής, ο αλγόριθμος βελτιστοποίησης που έχουμε προσθέσει, συγκλίνει με αποτέλεσμα να παράγουμε μια πολύ καλή τροχιά. Ενώ στις περιπτώσεις που δεν υπολογίζονται σωστά τα σημεία φυγής, το σύστημα μας, δεν καταφέρνει να υπερνικήσει το ORB-SLAM2 [23] όπως φαίνεται και στις ακολουθίες **fr1/desk,fr2/desk,fr3/structure-text-far,fr3/nostr=text-far**.

Επίσης, στον πίνακα 6.2 έχουμε καταγράψει και συγκρίνει το χρόνο εκτέλεσης των τριών μοντέλων για τα νήματα της παρακολούθησης (Tracking) και της τοπικής χαρτογράφησης (Local Mapping), καθώς αυτά τα δύο νήματα, παρουσιάζουν το μεγαλύτερο χρόνο εκτέλεσης. Μάλιστα, όπως παρατηρούμε οι ποιοί κοστοβόρες λειτουργίες και στα τρία συστήματα είναι η τοπική βελτιστοποίηση (Local Bundle Adjustment-(BA)) και η δημιουργία χαρακτηριστικών στον χάρτη. Ακόμη, παρατηρούμε ότι ο αλγόριθμος που χρησιμοποιούμε για την εξαγωγή των σημείων φυγής, εκτελείται σε 46,56 ms, που επιβεβαιώνει το ισχυρισμό μας ότι αποτελεί έναν γρήγορο αλγόριθμο για την εξαγωγή βέλτιστων σημείων φυγής, χωρίς να επιβαρύνει πολύ το σύστημα. Επιπλέον, το σύστημα εξακολουθεί με μια επιβάρυνση χρόνου λόγω των επιπρόσθετων γεωμετρικών χαρακτηριστικών, να εκτελείται σε real time, κανόντας αποδεκτή για εφαρμογές που ο σκοπός τους είναι να εκτελούνται σε real time.

Κεφάλαιο **7**

Συμπεράσματα

7.1 Επίλογος

Σε αυτή τη διπλωματική εργασία ασχοληθήκαμε με μεθόδους που αξιοποιούν την πληροφορία του περιβάλλοντος για να βελτιστοποιήσουν την τροχιά του συστήματος. Βελτιστοποιώντας την τροχιά, βελτιστοποιούμε έμμεσα και την ανακατασκευή του χώρου. Η μέθοδος που προτείναμε παρουσιάζει πλεονεκτήματα και αξιοποιεί βέλτιστα την πληροφορία σε περιβάλλοντα εσωτερικών χώρων που έχουν γραφεία, επίπεδες επιφάνειες, καρέκλες, τοίχους κ.α. Ξεκινήσαμε, αναλύοντας το πρόβλημα του SLAM και συνεχίσαμε στην κατεύθυνση στην οποία δουλέψαμε και παρουσιάσαμε μια ιστορία από εφαρμογές σε αυτό το ερευνητικό πεδίο.

Στη συνέχεια, παρουσιάσαμε την κλασική αρχιτεκτονική ενός τέτοιου συστήματος και προχωρήσαμε στην επεξήγηση αυτών των συστημάτων. Συνεχίσαμε, με τη λεπτομερή παρουσίαση της αρχικής περιγραφή του προβλήματος σαν ένα πρόβλημα βελτιστοποίησης μιας στατιστικής κατανομής και καταλήξαμε στη σημερινή εκδοχή αναπαράστασης τους προ βλήματος με γράφους ποζών. Έπειτα παρουσιάσαμε τα απαραίτητα εργαλεία για τη βελτιστοποίηση αυτού του γράφου μέσα από την οποία θα επέλθει η βελτιστοποίηση της πόζας. Τέλος, παρουσιάσαμε ορισμένα σύνολα δεδομένων που είναι χρήσιμα για την εκτέλεση και αξιολόγηση των μοντέλων για το πρόβλημα του *SLAM*.

Έπειτα, παρουσιάσαμε τη μαθηματική περιγραφή και μοντελοποίηση των ευθειών και των σημείων φυγής καθώς και τις μεθόδους που μπορούμε να ανιχνεύσουμε αυτά τα γεωμετρικά σχήματα. Και συνεχίσαμε με τους τρόπους με τους οποίους μπορούμε να εκτιμήσουμε την πόζα της κάμερας χρησιμοποιώντας είτε 2Δ/3Δ αντιστοιχίσεις, είτε την ίδια την εικόνα μέσα από τη βελτιστοποίηση του φωτομετρικού σφάλματος, είτε μέσω των σημείων φυγής. Καταλήγοντας στους τρόπους με τους οποίους μπορούμε να τη βελτιστοποιήσουμε χρησιμοποιώντας σημεία ή και ευθείες.

Τέλος, παρουσιάσαμε τις αρχιτεκτονικές των *state of the art* συστημάτων, πάνω στα οποία βασιστήκαμε και παρουσιάσαμε τη μεθοδολογία μας. Παρουσιάσαμε, πως μπορούμε να χρησιμοποιήσουμε τους αλγορίθμους για την εξαγωγή των σημείων φυγής και ευθειών. Επίσης, παρουσιάσαμε τον τρόπο με τον οποίο μπορούμε να χρησιμοποιήσουμε τα σημεία φυγής για να εκτιμήσουμε την περιστροφή της κάμερας. Στη συνέχεια, αναφέραμε τη μεθοδολογία μας για τη βελτιστοποίηση της περιστροφής, χρησιμοποιώντας τα σημεία φυγής και την αξιοποίηση τη γνώση της περιστροφής για τη βελτιστοποίηση της μετατόπισης, μέσα από

την επίλυση ενός γραμμικού συστήματος. Καταλήγοντας, δείξαμε ότι αυτή η αρχιτεκτονική είναι ικανή να παράγει πολύ ακριβείς τροχιές και ανταγωνιστικές με αυτές των καλύτερων εφαρμογών, σε περιβάλλοντα που παρουσιάζουν γεωμετρικά σχήματα όπως ευθείες. Στηρίζοντας τα αποτελέσματα με εικόνες και τα αντίστοιχα νούμερα σφαλμάτων καθώς και σε το είδος του συνόλου δεδομένων που χρησιμοποιήσαμε αλλά και τον λόγο που το χρησιμοποιήσαμε. Από αυτή την έρευνα, συμπεράναμε ότι όταν ένα σύστημα αξιοποιήσει την πληροφορία του περιβάλλοντος, εισάγοντας αλγορίθμους που ανιχνεύουν γεωμετρικά σχήματα, παραπάνω των σημείων, είναι σε θέση να εκτίμηση καλύτερα την τροχιά του. Επίσης, συμπεράναμε ότι η ακριβής εκτίμηση των σημείων φυγής μπορεί να κατευθύνει τους αλγορίθμους βελτιστοποίησης σε καλύτερα αποτελέσματα αλλά και το αντίθετο μπορεί να τους εγκλωβίσει σε τοπικά μέγιστα, παρουσιάζοντας κατώτερα αποτελέσματα. Επιπλέον, παρατηρήσαμε ότι είναι πολύ ουσιαστική η ύπαρξη συνόλων δεδομένων από πραγματικές σκηνές, για την καλύτερη αξιολόγηση και εποπτεία των αλγορίθμων.

7.2 Μελλοντική Εργασία

Όπως είδαμε, το πρόβλημα του *SLAM* παρουσιάζει πολλές προκλήσεις. Και αυτό, διότι ανάλογα με ποιο πρόβλημα θέλει κανείς να λύσει και τι χαρακτηριστικά θέλει να εισάγει το σύστημα, μπορεί να οδηγήσει σε υλοποιήσεις που φέρουν δυσκολίες αλλά ανοίγουν και καινούριους δρόμους προς εξερεύνηση. Στην κατεύθυνση της εκτίμησης και βελτίωσης της πόζας, που είναι και αυτή η διπλωματική εργασία, υπάρχουν πολλές ιδέες και αρχιτεκτονικές που θα μπορούσε κανείς να εισάγει, να επεκτείνει και να συνδυάσει με σκοπό ένα καλύτερο αποτέλεσμα. Και ένας καθοριστικός παράγοντας που αναδεικνύει τα προβλήματα που παρουσιάζονται σε ένα τέτοιο σύστημα είναι το είδος του περιβάλλοντος στο οποίο λειτουργεί το μοντέλο. Έτσι λοιπόν, λαμβάνοντας την έρευνα που έγινε σε αυτήν τη διπλωματική, ορισμένες μελλοντικές εργασίες που μπορούν να γίνουν είναι:

- **Local Bundle Adjustment:** Τα περισσότερα μοντέλα που λύνουν το πρόβλημα του *SLAM* όπως και το *ORB-SLAM2* [23] για την τοπική βελτιστοποίηση της πόζας και της θέσης των σημείων, χρησιμοποιούν ένα γράφο πόζας που βασίζεται στη βελτιστοποίηση του σφάλματος επαναπροβολής (*reprojection error*). Μια ιδέα που προτείνουμε και δεν έχει δοκιμαστεί τόσο ευρέως και αποτελεί μια διαφορετική προσέγγιση είναι ο υπολογισμός των σχετικών περιστροφών με βάση τα σημεία φυγής, για ένα σύνολο από *frames* και στη συνέχεια η χρήση του μαζί με τις απόλυτες περιστροφές τους, ώστε να δημιουργηθεί ένα γραμμικό σύστημα προς βελτιστοποίηση, η λύση του οποίου θα οδηγήσει σε βελτιστοποιημένες περιστροφές για τα αντίστοιχα *frames*. Επίσης, με αυτόν τον τρόπο θα οδηγήσει στη διαχωρίσιμη βελτιστοποίηση της πόζας της κάθε κάμερας, δηλαδή τη βελτιστοποίηση σε πρώτο βήμα όλων των περιστροφών, κάτω από ένα διαφορετικό σχήμα βελτιστοποίησης και στη συνέχεια την αξιοποίηση αυτής της πληροφορίας για τη βελτιστοποίηση της μετατόπισης από ένα γραμμικό σύστημα και στη συνέχεια της βελτιστοποίησης των θέσεων των σημείων.
- **Ανιχνευτές ευθειών:** Σε αυτήν τη διπλωματική εστίασαμε και χρησιμοποιήσαμε τον *LSD* [58] ανιχνευτή για την ανίχνευση των ευθειών. Ένας άλλος ανιχνευτής που θα

μπορούσαμε να πειραματιστούμε είναι ο *SOLD*² [60] που βασίζεται στη χρήση νευρωνικού δικτύου. Η χρήση αυτού του ανιχνευτή θα οδηγούσε σε ένα μοντέλο που βασίζεται σε καινούριες έρευνες και θα μπορούσε να οδηγήσει το σύστημα σε ακόμα καλύτερη εκτίμηση της πόζας.

- **Geometric Primitives:** Η χρήση περισσότερων και άλλων γεωμετρικών σχημάτων όπως επίπεδα, αξιοποιώντας ακόμα περισσότερο την πληροφορία του περιβάλλοντος και παράγοντας έναν ακόμα πιο πληθωρικό χάρτη.
- **Pose Estimation:** Η αξιοποίηση εργασιών που έχουν γίνει στην εκτίμηση της πόζας με νευρωνικά τόσο για στατικά περιβάλλοντα όπως στη δική μας εργασία, αλλά και σε δυναμικά περιβάλλοντα με κινούμενα αντικείμενα. Δίνοντας έτσι τη δυνατότητα για παραγωγή ενός συστήματος που θα λειτουργεί σε ακόμα περισσότερα περιβάλλοντα όπως δρόμοι.

Βιβλιογραφία

- [1] Franco Hidalgo και Thomas Bräunl. *Evaluation of Several Feature Detectors/Extractors on Underwater Images towards vSLAM*. *Sensors*, 20(15), 2020.
- [2] *Camera Model*. <https://alicevision.readthedocs.io/en/latest/openMVG/cameras/cameras.html>.
- [3] Xin Wang, Fei Xue, Zike Yan, Wei Dong, Qiuyuan Wang και Hongbin Zha. *Continuous-Time Stereo Visual Odometry Based on Dynamics Model*. *Computer Vision - ACCV 2018C*. V. Jawahar, Hongdong Li, Greg Mori και Konrad Schindler, editors, pages 388–403, Cham, 2019. Springer International Publishing.
- [4] *Course on SLAM ,Graph SLAM*. <https://www.iri.upc.edu/people/jsola/JoanSola/objectes/toolbox/courseSLAM.pdf>.
- [5] *Kalman Filter*. <https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalmanfilter.pdf>.
- [6] *Particle Filters*. http://jinyongjeong.github.io/2017/02/22/lec11_Particle_filter/.
- [7] Franz Andert και Stefan Krause. *Optical Aircraft Navigation with Multi-Sensor SLAM and Infinite Depth Features*. 2017.
- [8] *Course on SLAM, Factor Graph*. <https://www.iri.upc.edu/people/jsola/JoanSola/objectes/toolbox/courseSLAM.pdf>.
- [9] *Gauss Newton Method*. <https://ardianumam.wordpress.com/2017/09/27/newtons-method-optimization-derivation-and-how-it-works/>.
- [10] Rainer Kümmmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige και Wolfram Burgard. *G2o: A general framework for graph optimization*. *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.
- [11] Jari Mäkinen. *Rotation manifold $SO(3)$ and its tangential vectors*. *Computational Mechanics*, 42, pages 907–919, 2008.
- [12] *Line Representation*. <https://tutorial.math.lamar.edu/classes/calci/EqnsOfLines.aspx>.
- [13] *Vanishing Point Image*. <https://www.alamy.com/single-point-perspective-is-the-vanishing-point-and-the-horizon-line-it-is-an-approximate-representation-or-generally-on-a-flat-surface-vintage-line-image348648461.html>.

- [14] *Vanishing Point SVG*. https://www.wikiwand.com/en/Vanishing_point#Media/File:Vanishing_point.svg.
- [15] R. I. Hartley και A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2th Edition, 2004.
- [16] Jean Charles Bazin και Marc Pollefeys. *3-line RANSAC for orthogonal vanishing point detection*. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4282–4287, 2012.
- [17] Carsten Rother. *A New Approach for Vanishing Point Detection in Architectural Environments*. *BMVC*, 2000.
- [18] Xiaohu Lu, Jian Yaoy, Haoang Li, Yahui Liu και Xiaofeng Zhang. *2-Line Exhaustive Searching for Real-Time Vanishing Point Estimation in Manhattan World*. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 345–353, 2017.
- [19] Xiang Gao και Tao Zhang. *3D Rigid Body Motion*, pages 33–61. Springer Singapore, Singapore, 2021.
- [20] *Pose Estimation from epipolar geometry*. <http://fanxiaochen.github.io/2016/07/08/epipolar-geometry/>.
- [21] *Multiview Geometry slides*. https://www.uio.no/studier/emner/matnat/its/nedlagte-emner/UNIK4690/v16/forelesninger/lecture_7_3-pose-from-epipolar-geometry.pdf.
- [22] Kok Lim Low. *Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration*. 2004.
- [23] Raul Mur-Artal και Juan D. Tardós. *ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras*. *IEEE Transactions on Robotics*, 33, pages 1255–1262, 2017.
- [24] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu και Francese Moreno-Noguer. *PL-SLAM: Real-time monocular visual SLAM with points and lines*. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4503–4508, 2017.
- [25] Zichao Zhang και Davide Scaramuzza. *A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry*. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7244–7251, 2018.
- [26] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton και Olivier Stasse. *MonoSLAM: Real-Time Single Camera SLAM*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), pages 1052–1067, 2007.
- [27] Georg S. W. Klein και David William Murray. *Parallel Tracking and Mapping for Small AR Workspaces*. *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

- [28] Edward Rosten και Tom Drummond. *Machine Learning for High-Speed Corner Detection*. *Computer Vision - ECCV 2006* Aleš Leonardis, Horst Bischof και Axel Pinz, editors, pages 430–443. Springer Berlin Heidelberg, 2006.
- [29] Ethan Rublee, Vincent Rabaud, Kurt Konolige και Gary Bradski. *ORB: An efficient alternative to SIFT or SURF*. *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.
- [30] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley και Andrew W. Fitzgibbon. *Bundle Adjustment – A Modern Synthesis*. *Vision Algorithms: Theory and Practice* Bill Triggs, Andrew Zisserman και Richard Szeliski, editors, pages 298–372. Springer Berlin Heidelberg, 2000.
- [31] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart και Paul Furgale. *Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization*. *The International Journal of Robotics Research*, 34(3), pages 314–334, 2015.
- [32] Stefan Leutenegger, Margarita Chli και Roland Y. Siegwart. *BRISK: Binary Robust invariant scalable keypoints*. *2011 International Conference on Computer Vision*, pages 2548–2555, 2011.
- [33] Jakob J. Engel, Thomas Schöps και Daniel Cremers. *LSD-SLAM: Large-Scale Direct Monocular SLAM*. *ECCV*, 2014.
- [34] Christian Kerl, Jürgen Sturm και Daniel Cremers. *Dense visual SLAM for RGB-D cameras*. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106, 2013.
- [35] Christian Forster, Matia Pizzoli και Davide Scaramuzza. *SVO: Fast semi-direct monocular visual odometry*. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014.
- [36] Jakob Engel, Vladlen Koltun και Daniel Cremers. *Direct Sparse Odometry*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3), pages 611–625, 2018.
- [37] Yanyan Li, Nikolas Brasch, Yida Wang, Nassir Navab και Federico Tombari. *Structure-SLAM: Low-Drift Monocular SLAM in Indoor Environments*. *IEEE Robotics and Automation Letters*, 5, pages 6583–6590, 2020.
- [38] Thomas Whelan, Stefan Leutenegger, Renato F. Salas-Moreno, Ben Glocker και Andrew J. Davison. *ElasticFusion: Dense SLAM Without A Pose Graph*. *Robotics: Science and Systems*, 2015.
- [39] Federico Nardi, Bartolomeo Della Corte και Giorgio Grisetti. *Unified representation and registration of heterogeneous sets of geometric primitives*. *IEEE Robotics and Automation Letters*, 4(2), pages 625–632, 2019.

- [40] Lachlan Nicholson, Michael Milford και Niko Sünderhauf. *QuadricSLAM: Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM*. *IEEE Robotics and Automation Letters*, 4(1),pages 1–8, 2019.
- [41] N. Yang, L.von Stumberg, R. Wang και D. Cremers. *D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [42] Tristan Laidlow, Jan Czarnowski και Stefan Leutenegger. *DeepFusion: Real-Time Dense 3D Reconstruction for Monocular SLAM using Single-View Depth and Gradient Predictions*. *2019 International Conference on Robotics and Automation (ICRA)*, pages 4068–4074, 2019.
- [43] Keisuke Tateno, Federico Tombari, Iro Laina και Nassir Navab. *CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction*. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6565–6574, 2017.
- [44] Ruben Gomez-Ojeda, Francisco Angel Moreno, David Zuñiga-Noël, Davide Scaramuzza και Javier Gonzalez-Jimenez. *PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments*. *IEEE Transactions on Robotics*, 35,pages 734–746, 2019.
- [45] Haoang Li, Jian Yao, Jean Charles Bazin, Xiaohu Lu, Yazhou Xing και Kang Liu. *A Monocular SLAM System Leveraging Structural Regularity in Manhattan World*. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2518–2525, 2018.
- [46] Jiacheng Liu και Ziyang Meng. *Visual SLAM With Drift-Free Rotation Estimation in Manhattan World*. *IEEE Robotics and Automation Letters*, 5(4),pages 6512–6519, 2020.
- [47] Sudipta N. Sinha, Drew Steedly και Richard Szeliski. *A Multi-stage Linear Approach to Structure from Motion*. *ECCV Workshops*, 2010.
- [48] Chaoqiang Zhao, Yang Tang, Qiyu Sun και Athanasios V. Vasilakos. *Deep Direct Visual Odometry*. *IEEE Transactions on Intelligent Transportation Systems*, 23,pages 7733–7742, 2022.
- [49] Antoni Rosinol, Marcus Abate, Yun Chang και Luca Carlone. *Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping*. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020.
- [50] Jakob Engel, Jörg Stückler και Daniel Cremers. *Large-scale direct SLAM with stereo cameras*. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1935–1942, 2015.
- [51] Dominik Schlegel, Mirco Colosi και Giorgio Grisetti. *ProSLAM: Graph SLAM from a Programmer’s Perspective*. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3833–3840, 2018.

- [52] Ben Bellekens, Vincent Spruyt, Rafael Berkvens, Rudi Penne και Maarten Weyn. *A Benchmark Survey of Rigid 3D Point Cloud Registration Algorithms*. 2015.
- [53] Sebastian Thrun, Wolfram Burgard και Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [54] Sebastian Thrun. *Particle Filters in Robotics. Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI' 02, page 511–518, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [55] Kevin Doherty, Dehann Fourie και John Leonard. *Multimodal Semantic SLAM with Probabilistic Data Association*. pages 2419–2425, 2019.
- [56] Sameer Agarwal, Keir Mierle και The Ceres Solver Team. *Ceres Solver*, 2022.
- [57] V. S. Varadarajan. *Lie Groups and Lie Algebras*, pages 41–148. Springer New York, New York, NY, 1984.
- [58] RafaelGrompone von Gioi, Jeremie Jakubowicz, Jean Michel Morel και Gregory Randall. *LSD: A Fast Line Segment Detector with a False Detection Control*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4),pages 722–732, 2010.
- [59] Cuneyt Akinlar και Cihan Topal. *EDLines: A real-time line segment detector with a false detection control*. *Pattern Recognition Letters*, 32,pages 1633–1642, 2011.
- [60] Rémi Pautrat, Juan Ting Lin, Viktor Larsson, Martin Oswald και Marc Pollefeys. *SOLD 2 : Self-supervised Occlusion-aware Line Description and Detection*. pages 11363–11373, 2021.
- [61] James M. Coughlan και A. L. Yuille. *Manhattan World: Orientation and Outlier Detection by Bayesian Inference*. *Neural Computation*, 15(5),pages 1063–1088, 2003.
- [62] Stephen T. Barnard. *Interpreting perspective images*. *Artificial Intelligence*, 21(4),pages 435–462, 1983.
- [63] Jean Charles Bazin, Cédric Demonceaux, Pascal Vasseur και In So Kweon. *Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment*. *The International Journal of Robotics Research*, 31,pages 63 – 81, 2012.
- [64] M.E. Antone και S. Teller. *Automatic recovery of relative camera rotations for urban scenes*. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, vol 2, pages 282–289, 2000.
- [65] Patrick Denis, James H. Elder και Francisco J. Estrada. *Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery*. *Computer Vision - ECCV 2008*David Forsyth, Philip Torr και Andrew Zisserman, editors, pages 197–210. Springer Berlin Heidelberg, 2008.
- [66] Javier Finatb D. G. Aguilera A, Javier Gómez Lahoza. *A new method for vanishing points detection in 3D reconstruction from a single view*. 2012.

- [67] Faraz M. Mirzaei και Stergios I. Roumeliotis. *Optimal estimation of vanishing points in a Manhattan world*. 2011 International Conference on Computer Vision, pages 2454–2461, 2011.
- [68] Jean Charles Bazin, Yongduek Seo και Marc Pollefeys. *Globally Optimal Consensus Set Maximization through Rotation Search*. Computer Vision - ACCV 2012, Young Mu Lee, Yasuyuki Matsushita, James M. Rehg και Zhanyi Hu, editors, pages 539–551. Springer Berlin Heidelberg, 2013.
- [69] Olivier Faugeras. *Some Properties of the E Matrix in Two-View Motion Estimation*. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11, pages 1310–1312, 1989.
- [70] Vincent Lepetit, Francesc Moreno-Noguer και Pascal Fua. *EPnP: An Accurate $O(n)$ Solution to the PnP Problem*. *International Journal of Computer Vision*, 81(2), pages 155–166, 2009.
- [71] Mikael Persson και Klas Nordberg. *Lambda Twist: An Accurate Fast Robust Perspective Three Point (P3P) Solver*. *ECCV*, 2018.
- [72] Dorian Galvez-López και Juan D. Tardos. *Bags of Binary Words for Fast Place Recognition in Image Sequences*. *IEEE Transactions on Robotics*, 28(5), pages 1188–1197, 2012.
- [73] Lilian Zhang και Reinhard Koch. *An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency*. *Journal of Visual Communication and Image Representation*, 24, page 794–805, 2013.
- [74] Alexander Vakhitov, Jan Funke και Francesc Moreno-Noguer. *Accurate and Linear Time Pose Estimation from Points and Lines*. vol 9911, pages 583–599, 2016.
- [75] OpenCV. <https://opencv.org/>.
- [76] Eigen. https://eigen.tuxfamily.org/index.php?title=Main_Page.
- [77] Pangolin. <https://github.com/stevenlovegrove/Pangolin>.
- [78] Konstantinos G. Derpanis. *Overview of the RANSAC Algorithm*. http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf.
- [79] Berthold K. P. Horn. *Closed-form solution of absolute orientation using unit quaternions*. *J. Opt. Soc. Am. A*, 4(4), pages 629–642, 1987.
- [80] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard και Daniel Cremers. *A benchmark for the evaluation of RGB-D SLAM systems*. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 573–580, 2012.
- [81] TUM tools. <https://vision.in.tum.de/data/datasets/rgbd-dataset/tools>.
- [82] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik και Roland Siegwart. *The EuRoC micro aerial vehicle datasets*. *The International Journal of Robotics Research*, 2016.

- [83] Ankur Handa, Thomas Whelan, John B. McDonald και Andrew J. Davison. *A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM*. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, 2014.
- [84] Andreas Geiger, Philip Lenz, Christoph Stiller και Raquel Urtasun. *Vision meets robotics: The KITTI dataset*. *The International Journal of Robotics Research*, 32, pages 1231 – 1237, 2013.