National Technical University of Athens

School of Electrical and Computer Engineering

Division of Signals, Control and Robotics

Introducing Lines in Dynamic Visual SLAM and Rigid Object Motion Estimation

Diploma Thesis

of

ARGYRIS M. MANETAS

Supervisor: Petros Maragos
Professor NTUA

Athens, July 2024

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

Introducing Lines in Dynamic Visual SLAM and Rigid Object Motion Estimation

Diploma Thesis

of

ARGYRIS M. MANETAS

Supervisor: Petros Maragos
Professor NTUA

Approved by the Examining Committee on 18th July 2024.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Petros Maragos	Ioannis Kordonis	Athanasios Rontogiannis
Professor NTUA	Assistant Professor NTUA	Associate Professor NTUA

Athens, July 2024

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics

(Signature)

.....

Argyris M. Manetas
Graduate Electrical and Computer Engineer NTUA

Περίληψη

Στο συνεχώς αναπτυσσόμενο πεδίο της ρομποτικής, η λύση του προβλήματος της ταυτόχρονης εκτίμησης πόζας και χαρτογράφησης, το οποίο θα αναφέρεται εφεξής ως SLAM, έχει καθιερωθεί ως βασική προϋπόθεση για την επίτευξη αυτόνομων τεχνολογιών. Ως ένα διεπιστημονικό πεδίο, το SLAM συνδέει αρχές από τις επιστήμες της ρομποτικής και της όρασης υπολογιστών, με στόχο την ταυτόχρονη δημιουργία ενός χάρτη του χώρου και την εκτίμηση της πόζας του ρομπότ μέσα σε αυτόν. Αυτή η δυνατότητα αποτελεί πυλώνα μίας πληθώρας εφαρμογών, μεταξύ αυτών της κινητής ρομποτικής, επιτρέποντας σε αυτόνομα συστήματα να αντιλαμβάνονται και να κατευθύνονται μέσα σε περίπλοκα περιβάλλοντα. Αν και ο ορισμός του προβλήματος και οι μέθοδοι που χρησιμοποιούνται για την επίλυση του SLAM έχουν ωριμάσει σε σημαντικό βαθμό, μεγάλες προκλήσεις, όπως η δυναμική φύση των ανθρωποκεντρικών περιβαλλόντων, παραμένουν, με πολλές υπάρχουσες υλοποιήσεις να βασίζονται στην υπόθεση του στατικού κόσμου, που συνήθως παραβιάζεται.
Η ανάγκη για εύρωστα συστήματα SLAM έχει οδηγήσει στην σταδιακή εγκατάλειψη αυτής της υπόθεσης και στην στροφή προς δυναμικούς SLAM αλγόριθμους. Αν και έχουν υπάρξει πολλές προτάσεις δυναμικών SLAM, η συντριπτική πλειοψηφία αυτών βασίζεται μόνο σε σημειακά χαρακτηριστικά. Ωστόσο, η έρευνα σε στατικά SLAM συστήματα έχει δείξει ότι η συμπερίληψη περιπλοκότερων γεωμετρικών σχημάτων όπως οι ευθείες δύναται να βελτιώσει την απόδοση. Παρακινούμενοι από αυτή την παρατήρηση, σε αυτή τη διπλωματική εργασία δημιουργήσαμε ένα νέο δυναμικό SLAM σύστημα το οποίο εκτιμά τις πόζες της κάμερας και την κίνηση άκαμπτων αντικειμένων, αξιοποιώντας τόσο στατικά όσο και δυναμικά σημεία και ευθείες. Τα ευθύγραμμα τμήματα έχουν ενσωματωθεί με καινοτόμους τρόπους σε κάθε πτυχή του αλγορίθμου μας, με την βελτίωση στις αντιστοιχίσεις τους μέσω της βελτιστοποίησης της οπτικής ροής, την εισαγωγή όρων σφάλματος στις κινήσεις της κάμερας και των αντικειμένων, και στην βελτιστοποίηση παρτίδας. Η πρότασή μας δοκιμάστηκε εκτενώς σε σύνολα δεδομένων εσωτερικού και εξωτερικού χώρου και επέτυχε σημαντική βελτίωση σε σύγκριση με άλλα σύγχρονα δυναμικά SLAM συστήματα. Τα αποτελέσματα μας επέδειξαν ότι τα ευθύγραμμα τμήματα βελτίωσαν την ευρωστία, συνεισφέροντας με αυτόν τον τρόπο σε ένα πλήρως λειτουργικό SLAM σύστημα.

Λέξεις Κλειδιά

SLAM, Οπτικό SLAM, Δυναμικό SLAM, SLAM με Ευθείες, Πρόβλημα Ελαχιστοποίησης Μη-Γραμμικών Τετραγώνων

Abstract

In the rapidly evolving field of robotics, the Simultaneous Localization and Mapping (SLAM) problem has garnered significant attention as a cornerstone for autonomous technologies. As a multidisciplinary field, SLAM integrates principles from robotics, computer vision, optimization, and machine learning, with the aim to concurrently map the environment and estimate a robot's location within it. This capability is of great importance for a plethora of applications, including mobile robotics and augmented reality, enabling autonomous systems to navigate and perceive complex environments. Although the formulation of the problem and the approaches employed in SLAM have matured in a large degree, with the emergence of many robust algorithms, major challenges such as the dynamic nature of real-world environments still remain. Existent solutions often assume a static environment, an assumption frequently violated in human-occupied spaces.

The need for a robust SLAM system operating in real scenarios has led to the gradual abandonment of the static world assumption and to the creation of many dynamic SLAM algorithms. Even though there have been many dynamic SLAM proposals, the vast majority of them relied on point features. However, research in static SLAM systems has demonstrated that the use of more complex geometric shapes such as lines can improve performance. Motivated by this, in this thesis we have created a new dynamic SLAM system that estimates the camera poses and the motion of rigid objects, by exploiting both static and dynamic points and lines. Line segments have been incorporated in a novel way in every aspect of our algorithm, by improving their correspondences through optical flow refinement, and by introducing line error terms in both camera and object motion, and in batch optimization. Our proposal has been tested extensively in indoor and outdoor datasets and has achieved significant improvement compared to other state-of-the-art dynamic SLAM systems. Our results demonstrated that line segments enhanced the robustness, thus contributing towards a fully operational SLAM system.

Keywords

SLAM, Visual SLAM, Dynamic SLAM, Line SLAM, Non-Linear Least Squares Problem

Ευχαριστίες

Πρωτίστως θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Πέτρο Μαραγκό, όχι μόνο για την εμπιστοσύνη που μου έδειξε με την ανάθεση της εκπόνησης της παρούσας διπλωματικής εργασίας, αλλά και για το διδακτικό του έργο στην διάρκεια των φοιτητικών μου χρόνων, που με ενέπνευσε και με εισήγαγε στα επιστημονικά πεδία της Όρασης Υπολογιστών και των Σημάτων με τον πλέον άρτιο τρόπο. Ιδιαιτέρως θα ήθελα να ευχαριστήσω τον Υποψήφιο Διδάκτορα Παναγιώτη Μέρμηγκα, για την καίρια συνεισφορά του στο παρόν έργο και για τον χρόνο που αφιέρωσε. Η καθοδήγησή του, οι συζητήσεις μας και οι πάντα επικοδομητικές παρατηρήσεις του ήταν ζωτικής σημασίας για την ολοκλήρωση αυτής της εργασίας.

Με αφορμή την ολοκλήρωση των σπουδών μου, θα ήθελα να δώσω το πιο μεγάλο ευχαριστώ στους γονείς μου, Βασιλική και Μάριο, και στην αδερφή μου Ελένη, που σε όλη τη διάρκεια της ζωής μου βρισκόντουσαν κοντά μου, στηρίζοντας με ανιδιοτελώς με την πηγαία αγάπη τους. Θα ήθελα επίσης να ευχαριστήσω την Ελίνα, που στα χρόνια των σπουδών μου, μου παρείχε αμέριστη υποστήριξη και με ενέπνεε να είμαι η καλύτερη εκδοχή του εαυτού μου. Τέλος, θα ήθελα να ευχαριστήσω όλους μου τους φίλους, παιδικούς και αυτούς που δημιούργησα στην πορεία των σπουδών μου, για τις στιγμές που μοιραστήκαμε, και που ήταν καθημερινά δίπλα μου ως συνοδοιπόροι σε αυτό το ακαδημαϊκό ταξίδι.

Αργύρης Μ. Μανέτας

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

1

Εισαγωγή

1.1 Εισαγωγή στο οπτικό SLAM

Η ρομποτική είναι ένα ταχέως αναπτυσσόμενο επιστημονικό πεδίο, το οποίο επικεντρώνεται στον σχεδιασμό, την κατασκευή και την λειτουργία ρομπότ, τα οποία είναι μηχανές που επιστρατεύονται για να εκτελέσουν εργασίες με έναν καθορισμένο και ελεγχόμενο τρόπο, συνήθως χωρίς την ανάγκη ανθρώπινης παρέμβασης. Ο τελικός στόχος της ρομποτικής είναι να δημιουργήσει πραγματικά αυτόνομα και κινητά ρομπότ τα οποία θα δύνανται να λειτουργήσουν σε ποικίλα περιβάλλοντα, εκτελώντας πολύπλοκες και διαφορετικές εργασίες. Η σημασία των αυτόνομων κινήτων ρομπότ έχει ήδη αρχίσει να γίνεται εμφανής σε μια πληθώρα εφαρμογών, όπως η αυτόνομη οδήγηση, η εξερεύνηση του θαλάσσιου βυθού και του διαστήματος, οι αποστολές διάσωσης, η αποτροπή πυρκαγιών, η γεωργία, η φροντίδα ηλικιωμένων και πολλές άλλες. Είναι λοιπόν ευνόητο, πως η πρόοδος των ρομπότ μπορεί να συνεισφέρει σε σημαντική βελτίωση του βιοτικού επιπέδου των ανθρώπων, στην προστασία του περιβάλλοντος και στην οικονομική άνθηση.

Η αυτονομία των ρομπότ είναι ένα απαιτητικό έργο το οποίο σχετίζεται απευθείας με την ικανότητα τους να κατανοούν και να αλληλεπιδρούν με το περιβάλλον τους, η οποία καθίσταται εφικτή μέσω της αξιοποίησης ενός αριθμού αισθητήρων. Η ανάγκη για βελτίωση αυτών των ικανοτήτων έχει δώσει το έναυσμα για την εμφάνιση διαφόρων περιοχών έρευνας στην ρομποτική, που καθορίζονται από ένα πλήθος προβλημάτων τα οποία απαιτούν την συνεργασία πολλαπλών επιστημονικών κλάδων, όπως της όρασης υπολογιστών, της τεχνητής νοημοσύνης, της θεωρίας ελέγχου, της βελτιστοποίησης και άλλων. Ένα από τα σημαντικότερα προβλήματα της ρομποτικής είναι το πρόβλημα του ταυτόχρονου εντοπισμού και χαρτογράφησης (SLAM), το οποίο είναι ζωτικής σημασίας για την αντίληψη του περιβάλλοντος του ρομπότ και την ικανότητα του να πλοηγείται και να ενεργεί σε αυτό.

Το SLAM είναι μια θεμελιώδης και καλά μελετημένη περιοχή της ρομποτικής και της όρασης υπολογιστών, σημαντική για ένα ευρύ φάσμα εφαρμογών, μεταξύ αυτών της αυτόνομης οδήγησης, της επαυξημένης πραγματικότητας και των οικιακών ρομπότ. Το SLAM στοχεύει στην εύρεση της πιθανότερης τροχιάς ενός ρομπότ, δεδομένων των μετρήσεων των αισθητήρων του, ενώ ταυτόχρονα κατασκευάζει έναν χάρτη του περιβάλλοντος. Η ύπαρξη χάρτη αποτρέπει την συσσώρευση σφαλμάτων στην εκτίμηση της πόζας του ρομπότ, ενώ παρέχει επίσης ουσιώδεις πληροφορίες για την τοπολογία του περιβάλλοντος (βλ. Σχήμα 1.1). Για την επίλυση αυτού του προβλήματος έχουν χρησιμοποιηθεί διάφοροι αισθητήρες όπως κάμερες, στην οποία περίπτωση επιλέγεται η ονομασία οπτικό SLAM (vSLAM), IMUs και LiDAR. Η πρόοδος στην τεχνολογία των καμερών και η εύκολη πρόσβαση σε υψηλής ποιότητας κάμερες, όπως οι RGB-D κάμερες, έχουν οδηγήσει στην ανάπτυξη πολλών εύρωστων vSLAM συστημάτων.

Οι SLAM αλγόριθμοι έχουν διαφοροποιηθεί με πολλούς τρόπους, δημιουργώντας με αυτό τον τρόπο πολλές ανοικτές ερευνητικές περιοχές. Για παράδειγμα, ποικίλα δομικά στοιχεία, όπως αραία σημεία [KM07; MT17], στοιχεία όγκου (voxels) [Ros+21b], surfels [Sco+18] ή άλλα γεωμετρικές οντότητες όπως ευθείες ή επίπεδα, έχουν χρησιμοποιηθεί για την αναπαράσταση χαρτών. Παρόμοια, ο εντοπισμός στο οπτικό SLAM έχει επιτευχθεί είτε άμε-

σα [ESC14] είτε μέσω της ανίχνευσης χαρακτηριστικών, όπως τα ORB [Rub+11] ή περιπλοκότερων γεωμετρικών σχημάτων όπως ευθείες [Gom+19], επιπέδα [Kae15] ή και τα δύο ταυτόχρονα [Zhe+22].

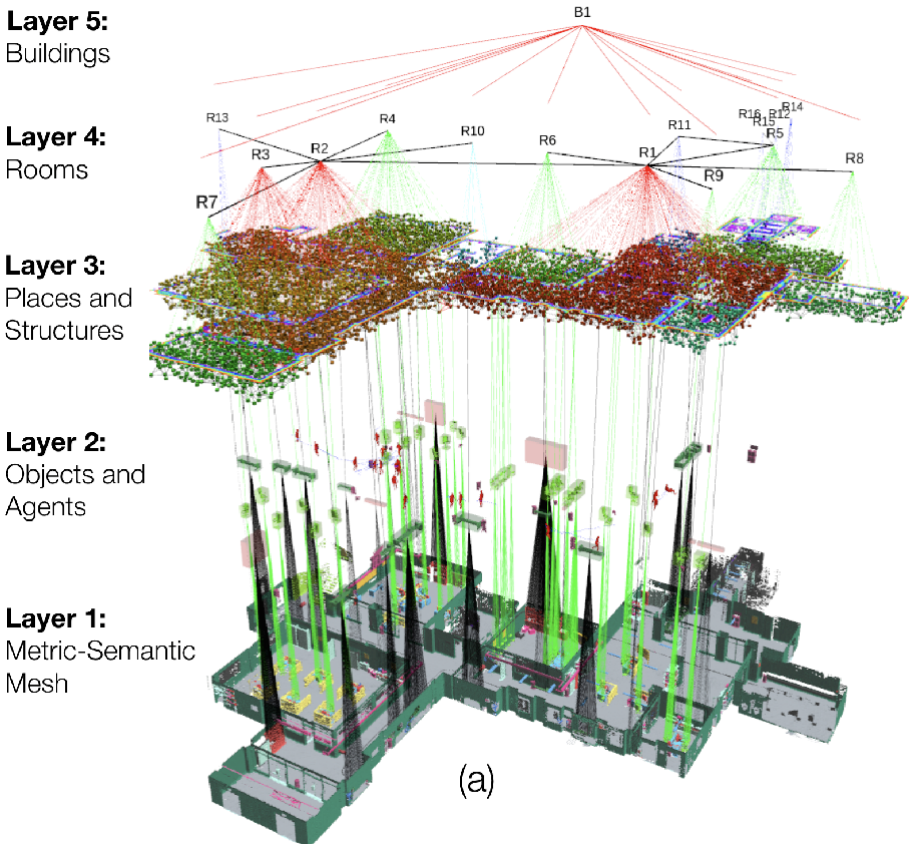
Αν και έχει υπάρξει σημαντική πρόοδος στο οπτικό SLAM και στο SLAM γενικότερα, με την ανάπτυξη πολλών εύρωστων και αποτελεσματικών αλγορίθμων, υπάρχουν ακόμα πολλές προκλήσεις οι οποίες απαιτείται να αντιμετωπιστούν. Ένα θεμελιώδες πρόβλημα είναι η ικανότητα των SLAM συστημάτων να χειρίζονται δυναμικά περιβάλλοντα, στα οποία τα αντικείμενα κινούνται, προστίθενται ή αφαιρούνται. Παραδοσιακά στην έρευνα του SLAM, ο κόσμος θεωρούταν ότι είναι στατικός και οι μετρήσεις σε δυναμικά αντικείμενα αντιμετωπίζονταν με γενικές τεχνικές απόρριψης ακραίων μετρήσεων (outlier), όπως το RANSAC [FB81], και συναρτήσεις σφάλματος, όπως η Huber συνάρτηση σφάλματος. Αυτή η προσέγγιση, ωστόσο, είναι επιρρεπής σε σφάλματα σε εντονα δυναμικά περιβάλλοντα και λόγω της μη-κυρτότητας του προβλήματος ελαχιστοποίησης που χρησιμοποιείται στο SLAM, οι εναπομείναντες παρατηρήσεις ακραιών τιμών, συχνά αποδεικνύονται ιδιαίτερα επιζήμιες στην συνολική ακρίβεια του συστήματος. Ως εκ τούτου, καθίσταται εμφανές ότι η ανάπτυξη εύρωστων δυναμικών συστημάτων είναι ζωτικής σημασίας για την λειτουργία ρομπότ σε πραγματικά περιβάλλοντα, τα οποία κυριαρχούνται από ανθρώπους, αυτοκίνητα και άλλα κινούμενα αντικείμενα.

Παρ΄όλο που έχει αποδειχθεί ότι η αξιοποίηση πιο περίπλοκων γεωμετρικών σχημάτων όπως οι ευθείες βελτιώνουν την ευρωστία του SLAM [Gom+19; Pum+17], ιδιαίτερα σε περιοχές χωρίς υφή και με χαμηλό φωτισμό, ελάχιστη έρευνα έχει γίνει για την χρήση τους σε δυναμικά περιβάλλοντα. Με κίνητρο αυτό και την ανάγκη για ακρίβη συστήματα SLAM σε ανθρωποκεντρικά περιβάλλοντα, προτείνουμε ένα σύστημα SLAM το οποίο εντοπίζει στατικά και δυναμικά σημεία και ευθείες για την εκτίμηση των θέσεων της κάμερας και της κίνησης των δυναμικών αντικειμένων στο χώρο (βλ. Σχήμα 1.2).

1.2 Η προσέγγισή μας, οι συνεισφορές μας και η διάρθρωση της διπλωματικής εργασίας

Στόχος αυτής της διπλωματικής εργασίας ήταν η εξοικείωση με το πεδίο του vSLAM, τις βασικές αρχές του, και η ανάπτυξη ενός αλγορίθμου ο οποίος εισάγει ευθείες ως γεωμετρικό χαρακτηριστικό στην υποκατηγορία των δυναμικών SLAM, προκειμένου να ενισχυθεί η ακρίβεια και η ευρωστία του συστήματος σε δυναμικά σενάρια. Αρχικά, έγινε μια εκτενής ανασκόπηση της υπάρχουσας βιβλιογραφίας, με έμφαση στις διάφορετικές θεωρητικές προσεγγίσεις στο πρόβλημα του SLAM, στις τεχνικές βελτιστοποίησης που χρησιμοποιούνται, στα διαφορετικά χαρακτηριστικά που αξιοποιούνται για τον εντοπισμό και την χαρτογράφηση, στα υπάρχοντα συστήματα SLAM και ιδιαίτερα σε εκείνα τα οποία έχουν αναπτυχθεί για δυναμικά περιβάλλοντα. Λαμβάνοντας υπόψη τα πλεονεκτήματα και τις αδυναμίες των τεχνικών που έχουν χρησιμοποιηθεί, αναπτύξαμε ένα νέο σύστημα SLAM, με συνεισφορές και καινοτομίες σε κάθε πτυχή της υλοποιησής μας, όπως:

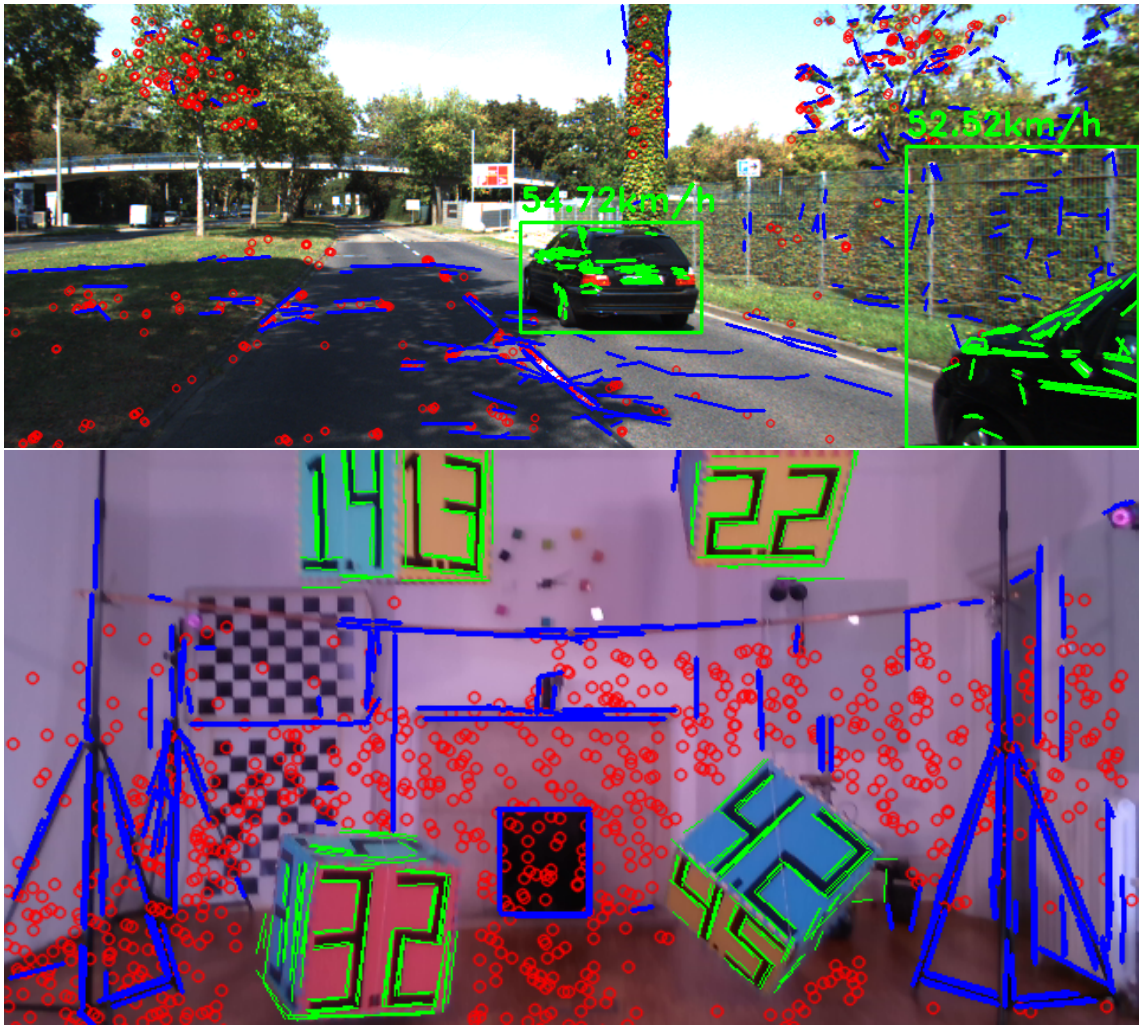
- Την αξιοποίηση οπτικής ροής για μεγαλύτερο αριθμό αντιστοιχίων ευθείων.



Σχήμα 1.1: Παράδειγμα κατασκευασμένου χάρτη από το Kimera [Ros+21b], ένα σύστημα SLAM τελευταίας τεχνολογίας. Ο χάρτης έχει πολλαπλά επίπεδα αφαίρεσης, από το πιο εξειδικευμένο επίπεδο του Μετρικού-Σημασιολογικού Πλέγματος μέχρι το πιο αφαιρετικό επίπεδο των Κτιρίων. Αυτό το σύστημα κάνει εμφανείς τις δυνατότητες που μπορεί να παρέχει στα ρομπότ ένα μοντέρνο οπτικό SLAM σύστημα, επιτρέποντας τα να αντιλαμβάνονται την πραγματική τοπολογία του περιβάλλοντος τους.

- Την εισαγωγή όρων σφάλματος επαναπροβαλλόμενων ευθειών για τον εντοπισμό της θέσης της κάμερας και την εκτίμηση της κίνησης αντικειμένων, με τη ταυτόχρονη βελτιστοποίηση της οπτικής ροής ως διπλή συμβολή.
- Την ένταξη ευθειών στην μερική και ολική βελτιστοποίηση παρτίδας, με την εισαγωγή καινοτόμων συναρτήσεων σφάλματος.
- Την επαλήθευση της μεθόδου μας σε απαιτητικά σύνολα δεδομένων και της συγκριση τους με δυναμικά συστήματα SLAM τελευταίας τεχνολογίας.

Εν κατακλείδι, συνδυάζοντας τα προτερήματα των δυναμικών SLAM και των ευθειών, αναπτύξαμε ένα σύστημα, την απόδοση του οποίου επαληθεύσαμε σε σύνολα δεδομένων τόσο εσωτερικού όσο και εξωτερικού χώρου, και δείξαμε ότι επιτυγχάνει καλύτερα απότελέσματα σε σχέση με άλλα συστήματα τελευταίας τεχνολογίας. Το υπόλοιπο της διατριβής δομείται ως εξής:



Σχήμα 1.2: **Έξοδος του συστήματος μας:** Σημεία και ευθείες εντοπίζονται τόσο σε στατικά όσο και σε δυναμικά αντικείμενα. Χαρακτηριστικά που φαίνονται: στατικά σημεία (Κόκκινο), στατικές ευθείες (Μπλε), και δυναμικές ευθείες (Πράσινο). Στην εικόνα φαίνεται και η ταχύτητα που έχει υπολογιστεί από την εκτιμώμενη κίνηση των αυτοκινήτων.

- Στο δεύτερο κεφάλαιο παρουσιάζουμε το θεωρητικό υπόβαθρο του SLAM, καθώς και σχετική έρευνα.
- Στο τρίτο κεφάλαιο παρουσιάζουμε συνοπτικά το VDO-SLAM, ένα σύστημα τελευταίας τεχνολογίας το οποίο αποτέλεσε την βάση της υλοποίησης μας, και με μεγάλη λεπτομέρεια το SDPL-SLAM, το καινοτόμο μας σύστημα.
- Στο τέταρτο κεφάλαιο παρουσιάζουμε την πειραματική αξιολόγηση του συστηματός μας.
- Στο πέμπτο κεφάλαιο ολοκληρώνουμε την εργασία μας και παρέχουμε κατευθύνσεις για μελλοντική έρευνα.

2

Θεωρητικό Υπόβαθρο και Σχετική Έρευνα

2.1 Αρχικές προσεγγίσεις στο Πρόβλημα του SLAM με Μεθόδους Φιλτραρίσματος

Στα αρχικά στάδια της έρευνας του SLAM, η κυρίαρχη προσέγγιση για την λύση του προβλήματος ήταν μέσω του πιθανοτικού φιλτραρίσματος, σύμφωνα με το οποίο εκτιμάται μόνο η τελευταία κατάσταση του ρομπότ ή της κάμερας. Συγκεκριμένα, δοσμένων μιας σειράς παρατηρήσεων $\mathbf{z}_{1:t}$ και ελέγχων των ενεργοποιητών (actuators) του ρομπότ $\mathbf{u}_{1:t}$, ο στόχος είναι να προσδιοριστεί η κατανομή πιθανότητας του διανύσματος κατάστασης του ρομπότ την χρονική στιγμή t, $\mathbf{x}_t$. Ακολουθώντας τις συμβάσεις του [TBF05], συμβολίζουμε την πεποίθηση ότι ένα ρομπότ βρίσκεται στην κατάσταση $\mathbf{x}_t$ ως $\text{bel}(\mathbf{x}_t)$:

$$\text{bel} = P(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \quad \overline{\text{bel}} = P(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \tag{2.1}$$

Η διαφορά των δύο παραπάνω ορισμών εντοπίζεται στην ενσωμάτωση ή όχι της μέτρησης $\mathbf{z}_t$ στον υπολογισμό της πεποίθησης. Έχοντας ορίσει τα παραπάνω, ο Αλγόριθμος Φίλτρου Bayes είναι ο εξής:

Αλγόριθμος 1: Τροποποιημένος Αλγόριθμος Φίλτρου Bayes [TBF05]

Data: $\text{bel}(\mathbf{x}_{t-1}), \mathbf{u}_t, \mathbf{z}_t$
Result: $\text{bel}(\mathbf{x}_t)$
forall $\mathbf{x}_t$ **do**
  $\overline{\text{bel}}(\mathbf{x}_t) = \int P(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1})\text{bel}(\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1}$
  $\text{bel}(\mathbf{x}_t) = \eta P(\mathbf{z}_t|\mathbf{x}_t)\overline{\text{bel}}(\mathbf{x}_t)$
end
return $bel(\mathbf{x}_t)$

Όπως μπορεί να παρατηρηθεί, ο Αλγόριθμος 1 χωρίζει την εκτέλεσή του επαναληπτικά σε δύο βήματα (i) ένα βήμα εκτίμησης που χρησιμοποιεί την κατανομή πεποιθήσεων της προηγούμενης κατάστασης $\mathbf{x}_t$ και τους ελέγχους $\mathbf{u}_t$ και (ii) ένα διορθωτικό βήμα που ενσωματώνει την τελευταία μέτρηση.

Για να χρησιμοποιηθεί ο Αλγόριθμος Φίλτρου Bayes στην πράξη πρέπει να γίνουν κάποιες απλοποιήσεις:

- Γραμμικότητα του μοντέλου κίνησης με πρόσθετο γκαουσιανό θόρυβο ϵ_t:

$$\mathbf{x}_t = A_t\mathbf{x}_{t-1} + B_t\mathbf{u}_t + \epsilon_t \tag{2.2}$$

- Γραμμικότητα του μοντέλου μετρήσεων με πρόσθετο γκαουσιανό θόρυβο δ_t:

$$\mathbf{z}_t = C_t\mathbf{x}_t + \delta_t \tag{2.3}$$

- Η αρχική πεποίθηση $\text{bel}(\mathbf{x}_0)$ ακολουθεί γκαουσιανή κατανομή, με μέσο $\boldsymbol{\mu}_0$ και πίνακα συνδιακύμανσης Σ_0.

Υπό αυτές τις συνθήκες, εξαιτίας των ιδιοτήτων των γκαουσιανών κατανομών, μπορεί να αποδειχθεί ότι η κατανομή πεποίθησης $\text{bel}(\mathbf{x}_t)$ παραμένει γκαουσιανή κατανομή καθόλη

την διάρκεια της εκτέλεσης του αλγορίθμου. Τροποποιώντας κατάλληλα τον Αλγόριθμο 1 σύμφωνα με τις παραπάνω παραδοχές ορίζεται ο γνωστός Αλγόριθμος Φίλτρου Kalman ως ειδική περίπτωση του Αλγορίθμου Φίλτρου Bayes:

Αλγόριθμος 2: Αλγόριθμος Φίλτρου Kalman [TBF05]

Data: $\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}, \mathbf{u}_t, \mathbf{z}_t$

Result: $\boldsymbol{\mu}_t, \Sigma_t$

1 $\bar{\boldsymbol{\mu}}_t = A_t \boldsymbol{\mu}_{t-1} + B_t \mathbf{u}_t$

2 $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

3 $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$

4 $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\mathbf{z}_t - C_t \bar{\boldsymbol{\mu}}_t)$

5 $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$

6 **return** $\boldsymbol{\mu}_t, \Sigma_t$

Η πεποίθηση $\text{bel}(\mathbf{x}_t)$ αναπαρίσταται μέσω του μέσου όρου $\boldsymbol{\mu}_t$ και του πίνακα συνδιακύμανσης Σ_t της γκαουσιανής κατανομής της, οι R_t, Q_t είναι οι πίνακες συνδιακύμανσης των γκαουσιανών θορύβων ϵ_t και δ_t αντίστοιχα, και το K_t, που ονομάζεται κέρδος Kalman, καθορίζει την επιρροή που θα έχει η διαφορά μεταξύ της εκτιμώμενης μέτρησης και της πραγματικής μέτρησης στη νέα εκτιμώμενη κατάσταση.

Παρόλο που το φιλτράρισμα Kalman είναι πολύ δημοφιλές σε μια πληθώρα εφαρμογών, οι υποθέσεις που απαιτούνται σχετικά με την γραμμικότητα των πιθανοτικών μοντέλων στο SLAM είναι πολύ αυστηρές. Οι μεταβάσεις μπορούν να περιγραφούν ακριβέστερα με έναν πιο γενικευμένο τρόπο ως εξής:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) + \epsilon_t \quad \mathbf{z}_t = h(\mathbf{x}_t) + \delta_t \tag{2.4}$$

όπου g και h είναι μη-γραμμικές συναρτήσεις. Σε μια προσπάθεια να αντιμετωπιστεί αυτό το πρόβλημα, εισήχθη το Εκτεταμένο Φίλτρο Kalman (EKF), το οποίο αξιοποιεί την επέκταση Taylor πρώτου βαθμού για να δημιουργήσει τοπικές γραμμικές προσεγγίσεις των συναρτήσεων g και h, αποφεύγοντας με αυτόν τον τρόπο την υπεργενίκευση της γραμμικότητας σε όλο το πεδίο ορισμού τους, επιτρέποντας ακριβέστερη μοντελοποίηση της δυναμικής του συστήματος υπό μη γραμμικές συνθήκες.

Αλγόριθμος 3: Αλγόριθμος Εκτεταμένου Φίλτρου Kalman [TBF05]

Data: $\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}, \mathbf{u}_t, \mathbf{z}_t$

Result: $\boldsymbol{\mu}_t, \Sigma_t$

1 $\bar{\boldsymbol{\mu}}_t = g(\boldsymbol{\mu}_{t-1}, \mathbf{u}_t)$

2 $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

3 $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

4 $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\mathbf{z}_t - h(\bar{\boldsymbol{\mu}}_t))$

5 $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

6 **return** $\boldsymbol{\mu}_t, \Sigma_t$

Οι G_t, H_t είναι οι Ιακωβιανές των συναρτήσεων g και h αντίστοιχα. Είναι σημαντικό να σημειωθεί ότι τα αποτελέσματα του Αλγορίθμου EKF εξαρτώνται σε μεγάλο βαθμό από την ποιότητα της γραμμικοποίησης και την επιλογή της αρχικής κατάστασης. Ο Αλγόριθμος

EKF εκτελείται επαναληπτικά σε κάθε χρονικό βήμα, και για κάθε μέτρηση οι πίνακες μέσης τιμής και συνδιακύμανσης ενημερώνονται κατάλληλα με την προσθήκη ενός νέου όρου στην περίπτωση που η μέτρηση αυτή αντιστοιχεί σε ένα νέο παρατηρούμενο ορόσημο στο περιβάλλον ή με την ενημέρωση των υφιστάμενων όρων στην περίπτωση ενός γνωστού οροσήμου. Ο Αλγόριθμος EKF έχει πολυπλοκότητα $O(N^3)$, η οποία μπορεί να αποδοθεί στην αντιστροφή του πίνακα συνδιακύμανσης σε κάθε επανάληψη. Αυτή η υψηλή πολυπλοκότητα θέτει έναν σημαντικό περιορισμό στον αριθμό των οροσήμων που μπορεί να χειριστεί ο Αλγόριθμος EKF, καθιστώντας τον ακατάλληλο για προβλήματα SLAM μεγάλης κλίμακας.

2.1.1 Μη-παραμετρικές μέθοδοι

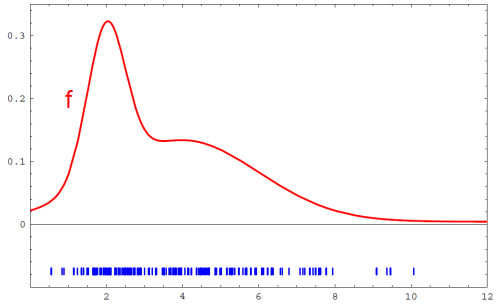
Οι μέθοδοι που αναλύθηκαν στην προηγούμενη ενότητα διέπονται από την υπόθεση ότι η πεποίθηση bel ακολουθεί γκαουσιανή κατανομή, η οποία αν και γενικά αποτελεσματική, επιφέρει ορισμένους περιορισμούς. Συγκεκριμένα, λόγω της μονοτροπικής (unimodal) φύσης των γκαουσιανών κατανομών, οι Αλγόριθμοι που βασίζονται στο φίλτρο Kalman αποτυγχάνουν να αναπαριστούν σωστά πολλαπλές υποθέσεις για την κατάσταση του ρομπότ, μια ικανότητα η οποία είναι ζωτικής σημασίας σε περιβάλλοντα στα οποία παρόμοια αντικείμενα μπορούν να προκαλέσουν διφορούμενες μετρήσεις και συνεπώς πολλαπλά τοπικά μέγιστα στην κατανομή πεποίθησης. Για να αντιμετωπιστεί αυτό το ζήτημα, έχουν διερευνηθεί μη-παραμετρικές μέθοδοι, οι οποίες δύνανται να αναπαραστήσουν πολύπλοκες κατανομές που δεν διαθέτουν απλή αναλυτική μορφή, χρησιμοποιώντας μια τεχνική δειγματοληψίας παρόμοια με τις μεθόδους Monte Carlo για την προσέγγιση της κατανομής. Μία από τις πιο κοινές μη-παραμετρικές μεθόδους είναι το φίλτρο σωματιδίων, το οποίο προσεγγίζει την κατανομή πεποίθησης με ένα σύνολο σωματιδίων $\mathbf{x}_t^{[1]}, \mathbf{x}_t^{[2]}, \ldots, \mathbf{x}_t^{[M]}$, που κατανέμονται σύμφωνα με την πεποίθηση bel. Ο αλγόριθμος μπορεί να περιγραφεί από τα επόμενα βήματα, χρησιμοποιώντας τον ίδιο συμβολισμό με την προηγούμενη ενότητα:

- **Αρχική Δειγματοληψία:** M αρχικά σωματίδια δειγματοληπτούνται από την $\overline{\text{bel}}(\mathbf{x}_t)$ εάν υπάρχει προηγούμενο χρονικό βήμα ή από την αρχική κατανομή πεποίθησης $\overline{\text{bel}}(\mathbf{x}_0)$ εάν t=0.

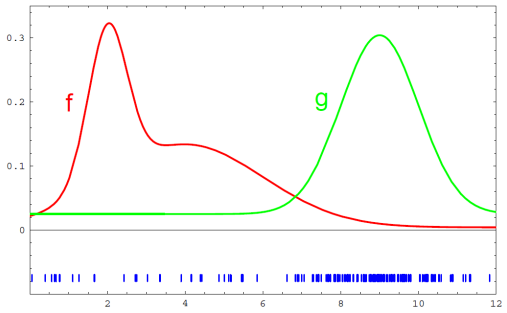
- **Ενημέρωση Βάρους:** Ένα βάρος επισυνάπτεται σε κάθε σωματίδιο του προηγούμενου βήματος βάσει της πιθανοφάνειας της μέτρησης δεδομένης της υπόθεσης της κατάστασης του ρομπότ $\mathbf{x}_t$: $P(\mathbf{z}_t|\mathbf{x}_t)$. Αυτή η διαδικασία απεικονίζεται στο Σχήμα 2.1.

- **Επαναδειγματοληψία:** Τα σωματίδια επαναδειγματοληπτούνται με πιθανότητα ανάλογη των αντίστοιχων βαρών τους, διατηρώντας με αυτόν τον τρόπο τις υποθέσεις σωματιδίων που είναι πιο πιθανές με βάση τις νέες παρατηρήσεις.

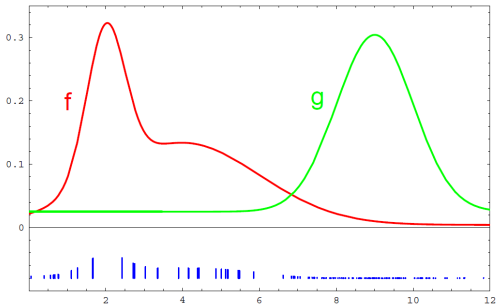
Δεδομένου ότι ο αριθμός των σωματιδίων είναι αρκετά μεγάλος, η παραπάνω διαδικασία μπορεί να προσεγγίσει την κατανομή πεποίθησης, με τους χώρους καταστάσεων που περιέχουν πυκνότερες περιοχές σωματιδίων να έχουν υψηλότερη πιθανότητα. Ένας επιτυχημένος αλγόριθμος SLAM που χρησιμοποιεί το φίλτρο σωματιδίων είναι ο αλγόριθμος FastSLAM [Mon+02], ο οποίος βασίζεται στο Rao-Blackwellized Particle Filter [Dou+13].



Στόχος είναι να προσεγγιστεί η κατανομή f, η οποία αναπαριστά την κατανομή bel, μέσω ενός αριθμού δειγμάτων.



Τα δείγματα μπορούν να παραχθούν μόνο από μια κατανομή g, η οποία αναπαριστά την $\overline{\text{bel}}$.



Βάρη $f(x)/g(x)$ δίνονται σε κάθε δείγμα. Τα δείγματα με μεγαλύτερα βάρη παρουσιάζονται με μακρύτερες μπλε γραμμές από κάτω τους.

Σχήμα 2.1: Απεικόνιση της τοποθέτης βάρους, βασισμένη στην κατανομή στόχο ($f = \text{bel}$) και στην προτεινόμενη κατανομή ($g = \overline{\text{bel}}$) [TBF05].

2.2 Μέθοδος Maximum a posteriori

Τα τελευταία χρόνια, η έρευνα στον τομέα του SLAM έχει συγκλίνει προς τις μεθόδους maximum a posteriori (MAP) έναντι των μεθόδων φιλτραρίσματος [SMD10] που περιγράφηκαν στις προηγούμενες ενότητες. Η MAP μέθοδος στο SLAM βασίστηκε στο πρωτοποριακό έργο των [LM97] και στόχευει στην εύρεση του πιο πιθανού συνδυασμού τροχιάς του ρομπότ και κατάστασης χάρτη δεδομένων ανεξάρτητων μετρήσεων σε κάθε χρονική στιγμή. Δηλαδή, έστω ότι ο συμβολισμός $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ αναφέρεται στη σειρά των κρυφών καταστάσεων του ρομπότ και $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$ η σειρά των παρατηρήσιμων μετρήσεων. Σε αυτή την περίπτωση η εύρεση της πιο πιθανής τροχιάς του ρομπότ είναι ισοδύναμη με την μεγιστοποίηση της ακόλουθης εκ των υστέρων πιθανότητας:

$$\text{argmax}_{\mathbf{x}} P(\mathbf{x}|\mathbf{z}) = \text{argmax}_{\mathbf{x}} \frac{P(\mathbf{z}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{z})} = \text{argmax}_{\mathbf{x}} P(\mathbf{z}|\mathbf{x})P(\mathbf{x}) \tag{2.5}$$

Για την εξαγωγή της τελικής μορφής της παραπάνω σχέσης χρησιμοποιήθηκε ο κανόνας του Bayes. Υποθέτοντας μια ενιαία πιθανότητα για όλες τις καταστάσεις του ρομπότ και την ανεξαρτησία των μετρήσεων, η εξίσωση (2.5) μπορεί να απλοποιηθεί περεταίρω, καταλήγοντας

στο κύριο πρόβλημα βελτιστοποίησης του MAP SLAM:

$$\text{argmax}_{\mathbf{x}} P(\mathbf{x}|\mathbf{z}) = \text{argmax}_{\mathbf{x}} P(\mathbf{z}|\mathbf{x}) = \text{argmax}_{\mathbf{x}} \prod_{k=1}^{K} P(\mathbf{z}_k|\mathcal{X}_k) \tag{2.6}$$

όπου $\mathcal{X}_k$ είναι το υποσύνολο των μεταβλητών από το οποίο εξαρτάται η μέτρηση $\mathbf{z}_k$. Η κατανομή $P(\mathbf{z}_k|\mathcal{X}_k)$ συνηθώς θεωρείται γκαουσιανή, με τον μέσο όρο της ίσο με την προβλεπόμενη τιμή της μέτρησης $\hat{\mathbf{z}}_k$ και πίνακα συνδιακύμανσης Ω_k ίσο με την αβεβαιότητα της μέτρησης. Η προβλεπόμενη μέτρηση υπολογίζεται από ένα μοντέλο μέτρησης h_k ως $\hat{\mathbf{z}}_k = h_k(\mathcal{X}_k)$. Οι όροι στο πρόβλημα βελτιστοποίησης μπορούν ως αποτέλεσμα να γραφτούν:

$$P(\mathbf{z}_k|\mathcal{X}_k) \propto \exp\left(-\frac{1}{2}(\mathbf{z}_k - h_k(\mathcal{X}_k))^T \Omega_k^{-1} (\mathbf{z}_k - h_k(\mathcal{X}_k))\right) \tag{2.7}$$

Όπως συμβαίνει συχνά σε προβλήματα βελτιστοποίησης που περιλαμβάνουν γκαουσιανές κατανομές πιθανοτήτων, η παραπάνω εξίσωση μετατρέπεται σε πρόβλημα ελαχιστοποίησης λαμβάνοντας τον αρνητικό λογάριθμο της κατανομής πιθανοτήτας, καταλήγοντας στην ακόλουθη συνάρτηση κόστους:

$$\text{argmin}_{\mathbf{x}} \sum_{k=1}^{K} (\mathbf{z}_k - h_k(\mathcal{X}_k))^T \Omega_k^{-1} (\mathbf{z}_k - h_k(\mathcal{X}_k)) \tag{2.8}$$

Από την παραπάνω εξίσωση μπορεί να εξαχθεί, ότι το αρχικό maximum a posteriori πρόβλημα είναι ισοδύναμο με ένα πρόβλημα βελτιστοποίησης μη-γραμμικών ελαχίστων τετραγώνων, το οποίο αποτελείται από όρους ενέργειας οι οποίοι είναι ανάλογοι του τετραγωνισμένου σφάλματος $\mathbf{e}_k = \mathbf{z}_k - h_k(\mathcal{X}_k)$ μεταξύ της προβλεπόμενης μέτρησης και της πραγματικής μέτρησης. Επομένως, το πρόβλημα SLAM με τη μέθοδο MAP μπορεί να αναχθεί στην εύρεση των κατάλληλων συναρτήσεων ενέργειας που περιγράφουν με ακρίβεια τους περιορισμούς που δημιουργούνται από την κίνηση του ρομπότ και τις μετρήσεις.

Επειδή αυτό το πρόβλημα βελτιστοποίησης είναι μη-γραμμικό και μη-κυρτό, δεν υπάρχει λύση κλειστής μορφής και, ως εκ τούτου χρησιμοποιούνται επαναληπτικές μέθοδοι βελτιστοποίησης για την επίλυσή του. Ωστόσο, ακόμη και αυτές οι επαναληπτικές μέθοδοι απαιτούν τη γραμμικοποίηση των συναρτήσεων κόστους, η οποία μπορεί να επιτευχθεί μέσω του αναπτύγματος Taylor πρώτης τάξης στη γειτονιά της αρχικής εκτίμησης της κατάστασης του ρομπότ $\check{\mathbf{x}}$:

$$\mathbf{e}_k(\check{\mathbf{x}} + \Delta\mathbf{x}) \approx \mathbf{e}_k(\check{\mathbf{x}}) + \mathbf{J}_k \Delta\mathbf{x} \tag{2.9}$$

όπου $\mathbf{J}_k$ είναι η ιακωβιανή του h_k ως προς $\mathbf{x}$ υπολογισμένη στο $\check{\mathbf{x}}$. Ορίζοντας $\mathrm{F} = \sum_{k=1}^{K} \mathrm{F}_k = \sum_{k=1}^{K} \mathbf{e}_k^T \Omega_k^{-1} \mathbf{e}_k$, και αντικαθιστώντας την (2.9), προκύπτει το εξής:

$$\begin{aligned}
\mathrm{F}_k(\check{\mathbf{x}} + \Delta\mathbf{x}) &= \mathbf{e}_k(\check{\mathbf{x}} + \Delta\mathbf{x})^T \Omega_k^{-1} \mathbf{e}_k(\check{\mathbf{x}} + \Delta\mathbf{x}) \approx (\mathbf{e}_k(\check{\mathbf{x}})^T + \mathbf{J}_k\Delta\mathbf{x})^T \Omega_k^{-1} (\mathbf{e}_k(\check{\mathbf{x}}) + \mathbf{J}_k\Delta\mathbf{x}) \\
&= \underbrace{\mathbf{e}_k(\check{\mathbf{x}})^T \Omega_k^{-1} \mathbf{e}_k(\check{\mathbf{x}})}_{c_k} + 2\underbrace{\mathbf{e}_k(\check{\mathbf{x}})^T \Omega_k^{-1} \mathbf{J}_k}_{\mathbf{b}_k^T} \Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\mathbf{J}_k^T \Omega_k^{-1} \mathbf{J}_k}_{H_k} \Delta\mathbf{x} \\
&= c_k + 2\mathbf{b}_k \cdot \Delta\mathbf{x} + \Delta\mathbf{x}^T H_k \Delta\mathbf{x}
\end{aligned}$$

και επομένως το πρόβλημα βελτιστοποίησης παίρνει την εξής μορφή:

$$\mathrm{F}(\check{\mathbf{x}}+\Delta\mathbf{x}) = \sum_{k=1}^{K} \mathrm{F}_k(\check{\mathbf{x}}+\Delta\mathbf{x}) = \sum_{k=1}^{K} c_k + 2\mathbf{b}_k \cdot \Delta\mathbf{x} + \Delta\mathbf{x}^T H_k \Delta\mathbf{x} = c + 2\mathbf{b}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T H \Delta\mathbf{x} \tag{2.10}$$

όπου $c = \sum_{k=1}^{K} c_k, \mathbf{b} = \sum_{k=1}^{K} \mathbf{b}_k$ και $H = \sum_{k=1}^{K} H_k$. Η βέλτιστη λύση $\Delta\mathbf{x}^*$, η οποία ελαχιστοποιεί τοπικά την συνάρτηση κόστους, μπορεί να υπολογιστεί παραγωγίζοντας την παραπάνω συνάρτηση ως προς $\Delta\mathbf{x}$ και θέτοντας την ίση με μηδέν, καταλήγοντας στην εξής εξίσωση:

$$H\Delta\mathbf{x}^* = -\mathbf{b} \tag{2.11}$$

Μόλις η λύση $\Delta\mathbf{x}^*$ βρεθεί, η νέα εκτίμηση της κατάστασης του ρομπότ, η οποία υπολογίζεται ως $\mathbf{x}^* = \check{\mathbf{x}} + \Delta\mathbf{x}^*$. χρησιμοποιείται ως αρχική εκτίμηση για την επόμενη επανάληψη του προβλήματος βελτιστοποίησης. Αυτή η επαναληπτική διαδικασία περιγράφει τον αλγόριθμο Gauss-Newton. Ένας άλλος επαναληπτικός αλγόριθμος που χρησιμοποιείται ευρέως στη βελτιστοποίηση στο SLAM είναι ο αλγόριθμος Levenberg-Marquadt, ο οποίος αποτελεί μία τροποποίηση του αλγορίθμου Gauss-Newton, περιλαμβάνοντας έναν παράγοντα απόσβεσης μέσω της προσέγγισης του Εσσιανού πίνακα (Hessian Matrix) H με τον ακόλουθο:

$$\mathbf{H} + \lambda\mathbf{I} \tag{2.12}$$

Το λ είναι ο παράγοντας απόσβεσης και $\mathbf{I}$ είναι ο ταυτοτικός πίνακας. Η λύση, παρόμοια με την (2.11) προέρχεται από την ακόλουθη εξίσωση:

$$(\mathbf{H} + \lambda\mathbf{I})\,\Delta\mathbf{x}^* = -\mathbf{b} \tag{2.13}$$

Ο παράγοντας απόσβεσης μεταβάλλεται καθ' όλη τη διάρκεια των επαναλήψεων, επιτρέποντας έτσι στον αλγόριθμο να συμπεριφέρεται όπως ο αλγόριθμος Gauss-Newton όταν η λύση βρίσκεται κοντά στο ελάχιστο, και όπως ο αλγόριθμος απότομης καθόδου (gradient descent) όταν η λύση απέχει πολύ από αυτό.

2.3 Βελτιστοποίηση σε πολλαπλότητες (manifolds)

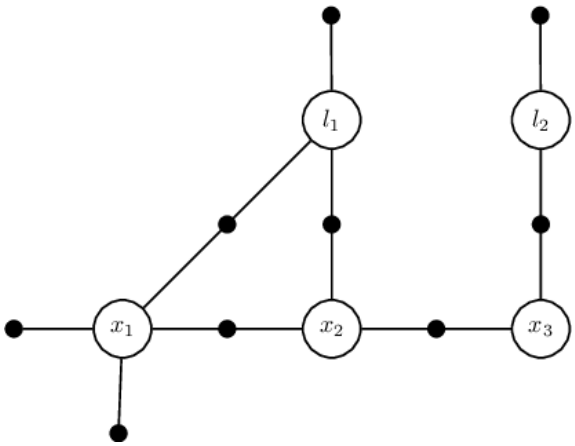
Στην ανάλυση της προηγούμενης ενότητας, θεωρήθηκε ότι η βελτιστοποίηση πραγματοποιείται σε έναν Ευκλείδιο χώρο, γεγονός που παραβλέπει μία κρίσιμη πτυχή του προβλήματος του SLAM: οι πόζες του ρομπότ περιορίζονται σε μια πολλαπλότητα (ή οποία θα αναφέρεται στο εξής με τον αγγλικό της όρο, manifold), λόγω των περιστροφικών της συνιστωσών. Ένα manifold διαφέρει από τον Ευκλείδιο χώρο, καθώς συμπεριφέρεται όπως αυτός μόνο τοπικά, με τον ίδιο τρόπο που η γη φαίνεται επίπεδη σε όσους στέκονται στην επιφάνεια της. Για παράδειγμα, η θέση ενός ρομπότ στον τρισδιάστατο χώρο μπορεί να αναπαρασταθεί από τον πίνακα $T(R, \mathbf{t}) \in SE(3)$, όπου $R \in SO(3)$ και $\mathbf{t} \in \mathbb{R}^3$ είναι η περιστροφή και η μετατόπιση του ρομπότ αντίστοιχα. Το $SO(3)$ σχηματίζει ένα τρισδιάστατο manifold ενσωματομένο μέσα σε ένα 9-διάστατο χώρο, καθώς οι παράμετροι ενός πίνακα περιστροφής είναι 9. Πράξεις όπως η πρόσθεση δεν ορίζονται στο $SO(3)$ όπως στους ευκλείδιους χώρους, και ως

αποτέλεσμα η προσπάθεια πρόσθεσης ποσοτήτων (όπως η πρόσθεση μιας ενημέρωσης στην επαναληπτική εκτέλεση των αλγορίθμων που αναφέρθηκαν στην προηγούμενη υποενότητα) σε έναν πίνακα περιστροφής, μπορεί να παραβιάσει τους εσωτερικούς περιορισμούς που επιβάλλει αυτό το manifold, όπως είναι η ορθογωνιότητα των πινάκων περιστροφής. Κατά συνέπεια, η βελτιστοποίηση μπορεί να αποφέρει λάθος λύσεις, οι οποίες απαιτούν επανακανονικοποίηση ώστε να είναι έγκυρες. Εκτός αυτού του ζητήματος, η επιλογή μιας υπερπαραμετροποιημένης αναπαράστασης για την κατάσταση του ρομπότ, όπως οι πίνακες περιστροφής και τα τετραδόνια (quaternions)—που έχουν περισσότερες παραμέτρους από τους πραγματικούς βαθμούς ελευθερίας—ενδέχεται να προκαλέσει βελτιστοποίηση μη-υπαρκτών βαθμών ελευθερίας που θα απαιτούσε περεταίρω κανονικοποίηση [HP08].

Επομένως, θα μπορούσε να υποστηριχθεί ότι για την αναπαράσταση περιστροφών οι ελάχιστες αναπαραστάσεις (minimal representations), όπως οι γωνίες Euler, είναι πιο κατάλληλες. Ωστόσο, αυτή η επιλογή εξακολουθεί να μην είναι βέλτιστη, δεδομένου ότι οι γωνίες αυτές υποφέρουν από εγγενή προβλήματα, όπως το γνωστό ζήτημα της ευθυγράμμισης δύο αξόνων περιστροφής (gimbal lock). Ως αποτέλεσμα για να αντιμετωπιστούν αυτά τα προβλήματα, έχει προταθεί οι περιστροφές να αναπαρίστανται γενικά με μία υπερπαραμετροποιημένη μορφή—χρησιμοποιώντας πίνακες περιστροφής ή μοναδιαία τετραδόνια—για να αποφεύγονται οι ιδιομορφίες και το gimbal lock, αλλά οι ενημερώσεις να υπολογίζονται με έλαχιστο τρόπο μέσα στον τοπικά Ευκλείδιο χώρο του manifold περιστροφής. Αυτές οι ενημερώσεις μπορούν να αντιστοιχηθούν άμεσα πίσω στο manifold, εξασφαλίζοντας με αυτόν τον τρόπο ότι ο προκύπτων πίνακας περιστροφής παραμένει έγκυρος ικανοποιώντας όλους τους απαραίτητους περιορισμούς, όπως η ορθογωνιότητα και μοναδιαία ορίζουσα, όπως απαιτείται για τις αναπαραστάσεις του $SO(3)$.

2.4 Γράφοι Παραγόντων

Η μέθοδος maximum a posteriori, που αναλύθηκε στην ενότητα 2.2, μπορεί να συνδεθεί με τον υπολογισμό μίας πιθανότητας σε έναν πιθανοτικό γραφικό μοντέλο. Το πρόβλημα SLAM έχει μια πολύ συγκεκριμένη δομή, καθώς οι μετρήσεις βασίζονται μόνο σε ένα συγκεκριμένο υποσύνολο των καταστάσεων του ρομπότ. Αυτή η δομή μπορεί να αναπαρασταθεί αποτελεσματικά με πιθανοτικά γραφικά μοντέλα, τα οποία είναι σε θέση να περιγράψουν πολύπλοκες πυκνότητες πιθανοτήτων, καθώς και τις αλληλοεξαρτήσεις των μεταβλητών. Όπως απεδείχθη στην σχέση (2.6) η εκ των υστέρων πιθανότητα μπορεί να παραγοντοποιηθεί σε ένα γινόμενο όρων, γεγονός που μπορεί να αξιοποιηθεί για την αναπαράσταση του προβλήματος του SLAM ως γράφου παραγόντων (factor graph) [DK+17]. Ένας γράφος παραγόντων είναι ένας διμερής γράφος που αποτελείται από δύο τύπους κόμβων, κόμβους μεταβλητών και κόμβους παραγόντων, με τους κόμβους μεταβλητών να αντιπροσωπεύουν μη-παρατηρήσιμες μεταβλητές, όπως οι καταστάσεις του ρομπότ και των στοιχείων του χάρτη, και τους κόμβους παραγόντων να αντιστοιχούν στους περιορισμούς μεταξύ των μεταβλητών, που δημιουργούνται από τις μετρήσεις. Προκύπτει ότι οι κόμβοι παραγόντων συνδέονται μόνο με τους κόμβους μεταβλητών από τους οποίους εξαρτώνται. Ένα παράδειγμα γράφου παραγόντων φαίνεται στο



Σχήμα 2.2: Αναπαράσταση του προβλήματος του SLAM ως γράφος παραγόντων, που περιέχει τις ρομποτικές πόζες x_1, x_2, x_3 και τα ορόσημα l_1, l_2. Οι κόμβοι μεταβλητών αντιστοιχούν σε ασπρους κύκλους και οι κόμβοι παραγόντων σε μαύρα σημεία. Οι κόμβοι παραγόντων συνδέονται μόνο στους κόμβους μεταβλητών από τις οποίες εξαρτώνται και αναπαριστούν περιορισμούς μεταξύ των μεταβλητών, που έχουν δημιουργηθεί από μετρήσεις [DK+17].

Σχήμα 2.2. Έχουν υπάρξει δύο κύριες βιβλιοθήκες που αξιοποιούν αυτή τη δομή γράφου για την βελτιστοποίηση στο SLAM, το GTSAM [Del12] και η βιβλιοθήκη g2o [Küm+11]. Η τελευταία αναφέρεται στις δομές γράφων της ως υπεργράφους, αλλά είναι στην ουσία ισοδύναμες με τους γράφους παραγόντων.

2.5 Συστήματα SLAM

Στην μεγαλύτερη πλειοψηφία τα χαρακτηριστικά που χρησιμοποιούνται στο πρόβλημα του οπτικού SLAM είναι τα σημεία. Το PTAM [KM07] είναι ένα τέτοιο σύστημα, το οποίο χωρίζει τις διαδικασίες του εντοπισμού και της χαρτογράφησης σε δύο νήματα για να εξασφαλίσει την εκτέλεση σε πραγματικό χρόνο. Το ORB-SLAM2 [MT17] με την αξιοποίηση των αποδοτικών χαρακτηριστικών ORB και τη χρήση ενός αραιού γράφου πόζας για την διόρθωση δέσμης (Bundle Adjustment), πετυχαίνει απόδοση σε πραγματικό χρόνο, ενώ παράλληλα επιτυγχάνει ευρωστία και ακρίβεια με τη δυνατότητα κλεισίματος κύκλων και επανεκτίμησης της θέσης σε περιπτώσεις όπου χάνεται ο εντοπισμός.

Παρ΄όλα αυτά, τα σημεία συχνά ενδέχεται να παρέχουν ανεπαρκή αριθμό αντιστοιχιών σε περιβάλλοντα με χαμηλό φωτισμό και απουσία υφής, ενώ οι αραιοί χάρτες σημείων δεν περιέχουν χρήσιμη πληροφορία για το ρομπότ. Αντίθετα, πιο σύνθετα γεωμετρικά σχήματα, όπως οι γραμμές, συναντώνται συχνά και περικλείουν περισσότερες περιγραφικές πληροφορίες για το περιβάλλον. Αυτή η παρατήρηση οδήγησε στην εμφάνιση πολλών συστημάτων που χρησιμοποιούν γραμμές [Gom+19; Pum+17], επίπεδα [Kae15] ή και τα δύο [Zhe+22]. Για την αποφυγή μη-βέλτιστων λύσεων κατά τη χρήση αυτών των γεωμετρικών οντοτήτων σε μια διαδικασία βελτιστοποίησης, χρησιμοποιούνται ελάχιστες αναπαραστάσεις, όπως η ορθοκανονική αναπαράσταση [BS05] για τις γραμμές στο [Zuo+17].

Τα δυναμικά συστήματα SLAM μπορούν να χωριστούν σε δύο κατηγορίες. Τα συστήματα της πρώτης κατηγορίας ανιχνεύουν δυναμικά αντικείμενα στις εικόνες και τα αφαιρούν από τις διαδικασίες εντοπισμού και βελτιστοποίησης. Το DynaSLAM [Bes+18] αξιοποιεί τις σημασιολογικές μάσκες που παρέχονται από το Mask R-CNN [He+17] και τους ελέγχους σφαλμάτων επαναπροβολής για την απόρριψη των δυναμικών αντικειμένων. Στο DS-SLAM [Yu+18] η απόσταση από τις επιπολικές γραμμές χρησιμοποιείται σε συνδυασμό με τη σημασιολογική κατάτμηση για την απόρριψη δυναμικών αντικειμένων. Το StaticFusion [Sco+18] εκτελεί μια κοινή εκτίμηση της πόζας της κάμερας και της δυναμικότητας της σκηνής, χρησιμοποιώντας μια συνάρτηση ενεργειακού σφάλματος δύο όρων. Ανάλογα με την εκτιμώμενη δυναμικικότητα αποδίδεται ένα βάρος στις παρατηρήσεις, επηρεάζοντας τη συμμετοχή τους στο πρόβλημα βελτιστοποίησης.

Αντίθετα, τα συστήματα της δεύτερης κατηγορίας ανιχνεύουν δυναμικά χαρακτηριστικά και τα εντοπίζουν χωρίς να απορρίπτουν τμήματα των εικόνων, αξιοποιώντας έτσι αποτελεσματικότερα την υπάρχουσα πληροφορία και γεφυρώνοντας το πρόβλημα του SLAM και του εντοπισμού κινούμενων αντικειμένων (Multiple Object Tracking). Το VDO-SLAM [Zha+21] χρησιμοποιεί σημασιολογική πληροφορία για τη διάκριση των δυναμικών αντικειμένων από το στατικό περιβάλλον, ενσωματώνει και τα δύο στο SLAM και υπολογίζει την πορεία της κάμερας και την ανεξάρτητη κίνηση των δυναμικών άκαμπτων αντικειμένων χωρίς προηγούμενη γνώση των γεωμετρικών μοντέλων τους. Το DynaSLAM II [Bes+21] προτείνει ένα πρόβλημα διόρθωσης δέσμης που περιλαμβάνει τόσο στατικά όσο και δυναμικά χαρακτηριστικά, και δημιουργεί και βελτιστοποίει τρισδιάστατα πλαισία που οριοθετούν τα κινούμενα αντικειμένα.

Σε αυτή την διπλωματική εργασία, προτείνουμε ένα καινοτόμο SLAM σύστημα της δεύτερης κατηγορίας, το οποίο συνδυάζει τα προτερήματα του δυναμικού SLAM και την ευρωστία των γραμμικών SLAM συστημάτων, μέσω της παρακολούθησης σημείων και ευθειών τόσο στο στατικό περιβάλλον όσο και σε δυναμικά άκαμπτα αντικείμενα, με αποτέλεσμα μια υλοποίηση υψηλής ακρίβειας.

3

Η προσέγγισή μας

3.1 SDPL-SLAM

Σε αυτή την ενότητα θα παρουσιάσουμε την υλοποίηση του SLAM αλγορίθμου μας [MMM24]. Όπως αναφέρθηκε προηγουμένως, παρόλο που το πρόβλημα της δυναμικότητας στο SLAM έχει μελετηθεί εκτενώς, έχει γίνει ελάχιστη έρευνα σχετικά με τη χρήση πιο σύνθετων χαρακτηριστικών σε αυτό, όπως οι ευθείες ή τα επίπεδα, τα οποία έχει αποδειχθεί ότι βελτιώνουν την ακρίβεια και την ευρωστία των αλγορίθμων στη στατική περίπτωση. Παρακινούμενοι από αυτό η υλοποίησή μας αξιοποιεί εκτός από σημεία, στατικές και δυναμικές ευθείες, επιφέροντας μεγαλύτερο αριθμό και ποικιλία στο σύνολο των χαρακτηριστικών, βελτιώνοντας κατά συνέπεια την ακρίβεια. Η δομή των επόμενων υποενοτήτων είναι η εξής:

- Συμβολισμός
- Επισκόπηση του συστήματος και ανάλυση κάθε συνιστώσας του

3.2 Συμβολισμός

Τα συστήματα συντεταγμένων συμβολίζονται ως C_k και τοποθετούνται ως αριστεροί άνω δείκτες για τα σημεία και τις ευθείες. Εξαιρείται η περίπτωση του παγκόσμιου συστήματος αναφοράς 0 το οποίο παραλείπεται όπου είναι εφικτό.

Σημεία: Οι (μη)ομογενείς τρισδιάστατες συντεταγμένες του i-οστού σημείου στο πλαίσιο k, εκφρασμένες στο σύστημα συντεταγμένων C_k, συμβολίζονται με ${}^{C_k}\mathbf{M}_k^i \in \mathbb{P}^3$ (και ${}^{C_k}\tilde{\mathbf{M}}_k^i \in \mathbb{R}^3$). Ομοίως, οι δισδιάστατες συντεταγμένες ως προς το πλαίσιο συντεταγμένων I_k αναπαρίστανται ως $\mathbf{m}_k^i \in \mathbb{P}^2$ (και $\tilde{\mathbf{m}}_k^i \in \mathbb{R}^2$). Θεωρούμε ότι το τελευταίο στοιχείο των ομογενών συντεταγμένων είναι ίσο με την μονάδα.

Ευθείες: Ένα τρισδιάστατο ευθύγραμμο τμήμα j στο k μπορεί να αναπαρισταθεί από τα άκρα του $\{{}^{C_k}\mathbf{A}_k^j, {}^{C_k}\mathbf{B}_k^j\}$, ενώ μια άπειρη δισδιάστατη ευθεία στο πλαίσιο συντεταγμένων I_k συμβολίζεται με $\mathbf{l}_k^j$. Οι συντεταγμένες ευθείας Πλίκερ (Plücker) μπορούν να υπολογιστούν ως εξής:

$$ {}^{C_k}\mathcal{L}_k^j = \begin{bmatrix} {}^{C_k}\tilde{\mathbf{A}}_k^j \times {}^{C_k}\tilde{\mathbf{D}}_k^j \\ {}^{C_k}\tilde{\mathbf{D}}_k^j \end{bmatrix} = \begin{bmatrix} {}^{C_k}\tilde{\mathbf{N}}_k^j \\ {}^{C_k}\tilde{\mathbf{U}}_k^j \end{bmatrix} \tag{3.1} $$

όπου ${}^{C_k}\tilde{\mathbf{D}}_k^j$ είναι το μοναδιαίο διάνυσμα κατεύθυνσης της ευθείας. Μπορεί να παρατηρηθεί ότι αυτός δεν είναι ο γενικός ορισμός των συντεταγμένων Πλίκερ, καθώς επιβάλλουμε επίσης τους δύο περιορισμούς $||{}^{C_k}\tilde{\mathbf{U}}_k^j|| = 1$ και ${}^{C_k}\tilde{\mathbf{N}}_k^j \cdot {}^{C_k}\tilde{\mathbf{U}}_k^j = 0$. Αυτοί οι δύο περιορισμοί μειώνουν τους βαθμούς ελευθερίας των συντεταγμένων Πλίκερ σε τέσσερις, επιτρέποντας κατά αυτόν τον τρόπο τον ένα προς ένα μετασχηματισμό στην ορθοκανονική αναπαράσταση. Η ορθοκανονική αναπαράσταση της ευθείας $(U, W) \in SO(3) \times SO(2)$ μπορεί να υπολογιστεί από τις συντεταγμένες Πλίκερ ως ακολούθως:

$$ {}^{C_k}U_k^j(\theta) = \begin{bmatrix} \frac{{}^{C_k}\tilde{\mathbf{N}}_k^j}{||{}^{C_k}\tilde{\mathbf{N}}_k^j||} & \frac{{}^{C_k}\tilde{\mathbf{U}}_k^j}{||{}^{C_k}\tilde{\mathbf{U}}_k^j||} & \frac{{}^{C_k}\tilde{\mathbf{N}}_k^j \times {}^{C_k}\tilde{\mathbf{U}}_k^j}{||{}^{C_k}\tilde{\mathbf{N}}_k^j \times {}^{C_k}\tilde{\mathbf{U}}_k^j||} \end{bmatrix} \tag{3.2} $$

$$ {}^{C_k}W_k^j(\theta) = \begin{bmatrix} ||{}^{C_k}\tilde{\mathbf{N}}_k^j|| & -||{}^{C_k}\tilde{\mathbf{U}}_k^j|| \\ ||{}^{C_k}\tilde{\mathbf{U}}_k^j|| & ||{}^{C_k}\tilde{\mathbf{N}}_k^j|| \end{bmatrix} \tag{3.3} $$

Ο πίνακας U ενημερώνεται με το διάνυσμα $\boldsymbol{\theta}$ και ο W με την τιμή θ, όπως στο [BS05].

Οπτική ροή: Ορίζουμε το διάνυσμα το οποίο αντιστοιχεί στην κίνηση ενός πίξελ $\tilde{\mathbf{m}}^i_{k-1}$ από το I_{k-1} στο I_k:

$$\boldsymbol{\phi}^i_k = \tilde{\mathbf{m}}^i_k - \tilde{\mathbf{m}}^i_{k-1} \tag{3.4}$$

Οι οπτικές ροές που αντιστοιχούν στο αρχικό ή το τελικό σημείο του ευθύγραμμου τμήματος j από το I_{k-1} στο I_k είναι τα $\boldsymbol{\phi}^{j,\mathbf{a}}_k$ και $\boldsymbol{\phi}^{j,\mathbf{b}}_k$, αντίστοιχα.

Μετασχηματισμοί: Ένας πίνακας μετασχηματισμού από το πλαίσιο k' στο k συμβολίζεται με ${}^{k'}X_k \in SE(3)$: ${}^{C_{k'}}\mathbf{M}^i_k = {}^{k'}X_k {}^{C_k}\mathbf{M}^i_k$. Ο πίνακας μετασχηματισμού ${}_{k-1}{}^0H_k \in SE(3)$ αναπαριστά την κίνηση σημείων σε δυναμικά άκαμπτα αντικείμενα από το πλαίσιο $k-1$ στο k στο παγκόσμιο σύστημα αναφοράς 0, δηλαδή ${}^0\mathbf{M}^i_k = {}_{k-1}{}^0H_k {}^0\mathbf{M}^i_{k-1}$. Ένας μετασχηματισμός $(R, \mathbf{t})$ μπορεί να εφαρμοστεί σε μια ευθεία ή οποία αναπαρίσταται από Πλίκερ συντεταγμένες ως εξής:

$$T_{line} = \begin{bmatrix} R & [\mathbf{t}]_\times R \\ 0_{3\times 3} & R \end{bmatrix} \tag{3.5}$$

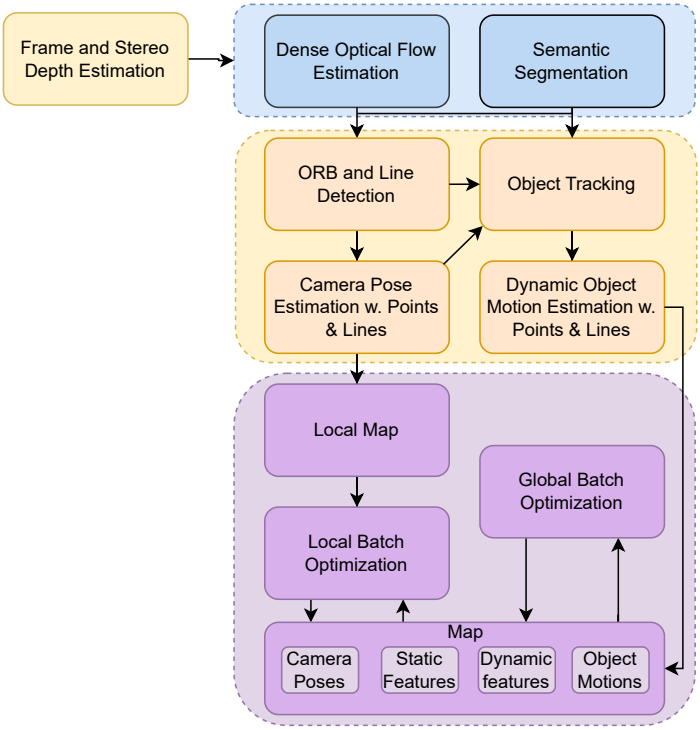
3.3 Επισκόπηση του SDPL-SLAM

Η επισκόπηση του συστήματος μας [MMM24] φαίνεται στο Σχήμα 3.1. Το σύστημα λαμβάνει ως είσοδο εικόνες RGB-D, οι οποίες προεπεξεργάζονται για την ανάκτηση πυκνής οπτικής ροής και για την σημασιολογική τους κατάτμηση. Στο στάδιο του εντοπισμού (tracking), υπολογίζεται η πόζα της κάμερας σε σχέση με το προηγούμενο πλαίσιο αξιοποιώντας παρατηρήσεις σημείων και ευθειών. Αφού ληφθεί η πόζα της κάμερας, εντοπίζονται τα δυναμικά αντικείμενα και ανακτάται η κινησή τους μεταξύ των δύο πλαισίων. Παράλληλα διατηρείται ένας τοπικός και ένας παγκόσμιος χάρτης, οι οποίοι περιέχουν τα στατικά σημεία και ευθείες, και τις ακολουθίες της πόζας της κάμερας και των κινήσεων των αντικειμένων. Ανά ένα καθορισμένο αριθμό χρονικών βημάτων, εκτελείται μια βελτιστοποίηση παρτίδας στον τοπικό χάρτη για βελτίωση της τοπικής τροχιάς—στην περίπτωση μας αυτός ο αριθμός χρονικών βημάτων ισούται με 20—ενώ η παγκόσμια βελτιστοποίηση παρτίδας εκτελείται στον παγκόσμιο χάρτη για την από κοινού βελτίωση ολόκληρης της τροχιάς της κάμερας και του χάρτη.

3.4 Αντιστοίχιση ευθειών και Εκτίμηση Πόζας της Κάμερας

Οι ευθείες στις εικόνες ανιχνεύονται με τη χρήση του Line Segment Detector [Von+08]. Οι ευθείες οι οποίες παρουσιάζουν ασυνέχεια στο βάθος ή των οποίων τα τελικά σημεία ανήκουν σε διαφορετικές σημασιολογικές μάσκες απορρίπτονται.

Για την αντιστοίχιση των ευθειών σε διαδοχικά πλαίσια χρησιμοποιείται η οπτική ροή με τον ίδιο τρόπο που αποκτούνται οι αντιστοιχίσεις σημείων στο [Zha+21]. Η αντιστοίχιση μέσω της οπτικής ροής λύνει ένα μείζον πρόβλημα που υπάρχει συχνά σε συστήματα SLAM που βασίζονται σε ευθείες που χρησιμοποιούν περιγραφητές (descriptors) ευθειών, στα οποία



Σχήμα 3.1: **Επισκόπηση συστήματος SDPL-SLAM (Static-Dynamic Point-Line SLAM):** Αποτελείται από τρεις συνιστώσες: προεπεξεργασία (Μπλε), εντοπισμός (Κίτρινο), και βελτιστοποίηση παρτίδας (Μωβ) [MMM24].

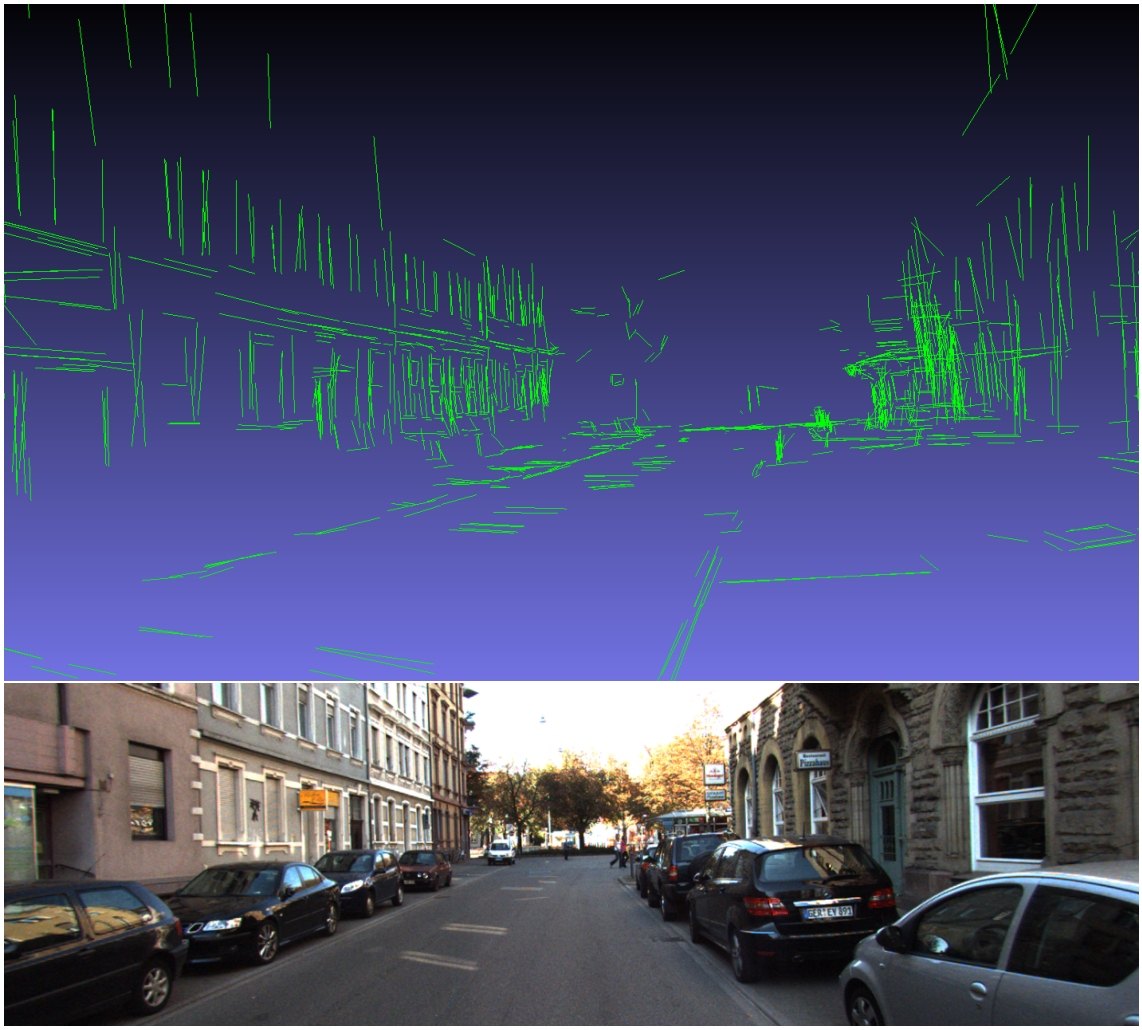
οι ευθείες δεν μπορούν να ανιχνευθούν με συνέπεια μεταξύ των εικόνων ή ανιχνεύονται με διαφορετικά μήκη. Στην πρώτη περίπτωση η αντίστοιχη ευθεία δεν υπάρχει, ενώ στην δεύτερη οι περιγραφητές ενδέχεται να μην ταιριάζουν εξαιτίας της διαφορετικής εμφάνισης της ευθείας. Αξιοποιώντας την οπτική ροή, πετύχαμε μεγαλύτερο πλήθος αντιστοιχιών ευθειών μεταξύ των εικόνων εξασφαλίζοντας μεγαλύτερα παρατηρήματα (tracklets) ευθειών.

Η αρχική θέση της κάμερας εκτιμάται με έναν αλγόριθμο Perspective-n-Point (PnP) [LMF09] σε συνδυασμό με τον αλγόριθμο RANSAC, χρησιμοποιώντας μόνο στατικά σημεία που δεν ανήκουν σε αντικείμενα. Για την βελτιώση αυτής της εκτίμησης, προτείνουμε ένα νέο πρόβλημα ελαχιστοποίησης, το οποίο βελτιστοποιεί ταυτόχρονα τη θέση της κάμερας και την οπτική ροή, βελτιώνοντας τις αρχική αντιστοίχιση σημείων και ευθειών. Συγκεκριμένα προτείνεται ο ακόλουθος όρος σφάλματος:

$$\mathbf{e}_{j,l} = \mathbf{e}_j({}^0X_k, \boldsymbol{\phi}_k^{j,\mathbf{a}}, \boldsymbol{\phi}_k^{j,\mathbf{b}}) = \begin{bmatrix} \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}^0X_k^{-1}\mathbf{A}_{k-1}^j) \\ \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}^0X_k^{-1}\mathbf{B}_{k-1}^j) \end{bmatrix} \tag{3.6}$$

όπου $\mathbf{l}_k^{j,obs}$ είναι η παρατηρηθείσα άπειρη ευθεία δοσμένη από:

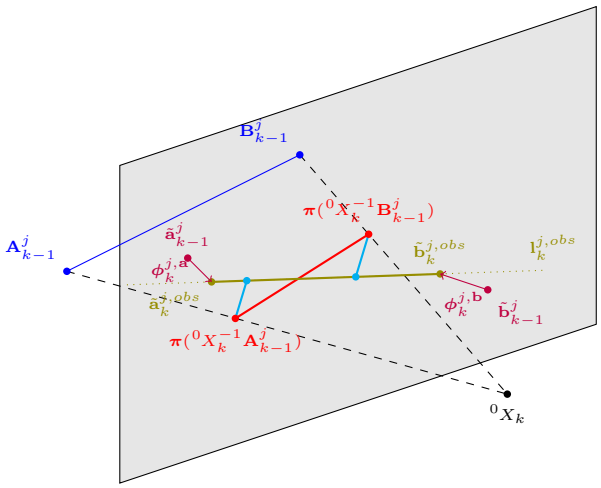
$$\mathbf{l}_k^{j,obs} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \frac{\mathbf{a}_k^{j,obs} \times \mathbf{b}_k^{j,obs}}{||\mathbf{a}_k^{j,obs} \times \mathbf{b}_k^{j,obs}||} \tag{3.7}$$



Σχήμα 3.2: Πάνω: Οπτικοποίηση των ευθειών που περιέχονται στον χάρτη που διατηρεί το SDPL-SLAM για μία σκηνή πόλης. Κάτω: Ένα καρέ από την ακολουθία δεδομένων που χρησιμοποιήθηκε για να παραχθεί ο χάρτης.

όπου $\boldsymbol{\pi}(\cdot)$ είναι η προβολική συνάρτηση που επιστρέφει ένα ομογενές διάνυσμα, $\boldsymbol{\phi}_k^{j,\mathbf{a}}$ και $\boldsymbol{\phi}_k^{j,\mathbf{b}}$ είναι οι οπτικές ροές που αντιστοιχούν στο αρχικό και τελικό σημείο της ευθείας j από το πλαίσιο συντεταγμένων I_{k-1} στο I_k και $\tilde{\mathbf{a}}_k^{j,obs} = \tilde{\mathbf{a}}_{k-1}^{j} + \boldsymbol{\phi}_k^{j,\mathbf{a}}$, $\tilde{\mathbf{b}}_k^{j,obs} = \tilde{\mathbf{b}}_{k-1}^{j} + \boldsymbol{\phi}_k^{j,\mathbf{b}}$ είναι τα τελικά σημεία της παρατηρηθείσας γραμμής στο τρέχον πλαίσιο. Ο όρος σφάλματος (3.6) αποτελείται από τις στοιβαγμένες αποστάσεις των επαναπροβληθέντων ακραίων σημείων της ευθείας j στο πλαίσιο $k-1$ από την ευθεία η οποία ορίζεται από τα αντίστοιχα παρατηρηθέντα ακραία σημεία στο πλαίσιο k (βλ. Σχήμα 3.3). Αν η λύση του προβλήματος ελαχιστοποίησης έχει ως αποτέλεσμα ένας όρος να ξεπερνάει ένα συγκεκριμένο όριο, τότε η αντίστοιχη ευθεία θεωρείται ως ακραία μέτρηση και αφαιρείται. Αυτός ο όρος σφάλματος μοιάζει με αυτό στο [Gom+19; Pum+17], ωστόσο, στην περίπτωση μας εξαρτάται ταυτόχρονα από την οπτική ροή, και είναι αναγκαίος ο υπολογισμός νέας Ιακωβιανής.

Η Ιακωβιανή του όρου σφάλματος υπολογίστηκε αναλυτικά. Η παράγωγος ως προς την



Σχήμα 3.3: **Τρισδιάσταση οπτικοποίηση του όρου σφάλματος της επαναπροβολής ευθείας:** Τα άκρα της ευθείας $\mathbf{A}^j_{k-1}$ και $\mathbf{B}^j_{k-1}$ προβάλλονται στο πλαίσιο συντεταγμένων I_k στα άκρα $\boldsymbol{\pi}({}^0X_k^{-1}\mathbf{A}^j_{k-1})$ και $\boldsymbol{\pi}({}^0X_k^{-1}\mathbf{B}^j_{k-1})$ που ορίζουν το επαναπροβαλλόμενο ευθύγραμμο τμήμα. Οι οπτικές ροές $(\boldsymbol{\phi}^{j,\mathbf{a}}_k, \boldsymbol{\phi}^{j,\mathbf{b}}_k)$ και τα άκρα του ευθύγραμμου τμήματος στο πλαίσιο $k-1$ $(\tilde{\mathbf{a}}^j_{k-1}, \tilde{\mathbf{b}}^j_{k-1})$ προστίθενται μαζί ώστε να ανακτηθούν τα παρατηρούμενα ακραία σημεία του αντίστοιχου ευθύγραμμου τμήματος στο πλαίσιο k. Ο όρος σφάλματος (3.6) αντιστοιχεί στις κυανές γραμμές και αναπαριστά τις αποστάσεις των επαναπροβαλλόμενων άκρων της ευθείας (Κόκκινο) από την αντιστοιχη παρατηρούμενη άπειρη ευθεία (Πράσινο).

οπτική ροή του αρχική σημείου (και αντίστοιχα του τελικού σημείου) υπολογίστηκε ως:

$$\frac{\partial \mathbf{e}_{j,l}}{\partial \boldsymbol{\phi}^{j,\mathbf{a}}_k} = \boldsymbol{\pi}({}^0X_k^{-1}\mathbf{A}^j_{k-1})^\top \frac{\partial \mathbf{l}^{j,obs}_k}{\partial \boldsymbol{\phi}^{j,\mathbf{a}}_k} \tag{3.8}$$

και η Ιακωβιανή ως προς τις παραμέτρους της πόζας $\boldsymbol{\Xi}_k$ ως ακολούθως:

$$\frac{\partial \mathbf{e}_{j,l}}{\partial \boldsymbol{\Xi}_k} = \begin{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix}^\top \frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{A}^j_{k-1})}{\partial \boldsymbol{\Xi}_k} \\ \begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix}^\top \frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{B}^j_{k-1})}{\partial \boldsymbol{\Xi}_k} \end{bmatrix} \tag{3.9}$$

Για τον υπολογισμό της Ιακωβιανής $\frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{A}^j_{k-1})}{\partial \boldsymbol{\Xi}_k}$ αρχικά θέτουμε $\mathbf{g} = [g_x, g_y, g_z]^T = {}^0X_k^{-1}A^j_{k-1}$. Η Ιακωβιανή έπειτα μπορεί να υπολογιστεί ως εξής:

$$\frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{A}^j_{k-1})}{\partial \boldsymbol{\Xi}_k} = \begin{bmatrix} \frac{f_x}{g_z} & 0 & -f_x\frac{g_x}{g_z^2} & -f_x\frac{g_xg_y}{g_z^2} & f_x(1+\frac{g_x^2}{g_z^2}) & -f_x\frac{g_y}{g_z} \\ 0 & \frac{f_y}{g_z} & -f_y\frac{g_y}{g_z^2} & -f_y(1+\frac{g_y^2}{g_z^2}) & f_y\frac{g_xg_y}{g_z^2} & f_y\frac{g_x}{g_z} \end{bmatrix} \tag{3.10}$$

Πληροφορίες για την παραπάνω παραγώγιση μπορούν να βρεθούν στο [Bla22], στο οποίο ο αναγνώστης έχει την ευκαρία να εμβαθύνει στο μαθηματικό υπόβαθρο.

Το πρόβλημα ελαχιστοποίησης που βασίζεται σε σφάλματα επαναπροβολής σημείων και

ευθειών, είναι λοιπόν το εξής:

$$\begin{aligned}\{{}^0X_k^*, \boldsymbol{\Phi}_k^*\} = \text{argmin}_{\{{}^0X_k, \boldsymbol{\Phi}_k\}} \sum_i^{n_p} \{\boldsymbol{\rho}_h(\mathbf{e}_{i,r}^\top \Sigma_{\boldsymbol{\phi}}^{-1} \mathbf{e}_{i.r}) + \\ \boldsymbol{\rho}_h(\mathbf{e}_{i,p}^\top \Sigma_p^{-1} \mathbf{e}_{i,p})\} + \sum_j^{n_l} \{\boldsymbol{\rho}_h(\mathbf{e}_{j,ra}^\top \Sigma_{\boldsymbol{\phi}}^{-1} \mathbf{e}_{j,ra}) + \\ \boldsymbol{\rho}_h(\mathbf{e}_{j,rb}^\top \Sigma_{\boldsymbol{\phi}}^{-1} \mathbf{e}_{j,rb}) + \boldsymbol{\rho}_h(\mathbf{e}_{j,l}^\top \Sigma_l^{-1} \mathbf{e}_{j,l})\},\end{aligned} \tag{3.11}$$

όπου με '*' συμβολίζεται η βέλτιστη λύση, n_p και n_l είναι το πλήθος των στατικών σημείων και ευθειών αντίστοιχα, $\mathbf{e}_{i,p}$ είναι ο γνωστός όρος σφάλματος επαναπροβολής σημείων [MT17; Zha+21; Qiu+22], $\mathbf{e}_{i,r}$, $\mathbf{e}_{j,ra}$ και $\mathbf{e}_{j,rb}$ είναι όροι κανονικοποίησης για την οπτική ροή που αντιστοιχούν στα σημεία [Zha+21], τα αρχικά και τελικά σημεία των ευθειών, αντίστοιχα, Σ_ϕ είναι ο πίνακας συνδιακύμανσης για τους όρους σφάλματος κανονικοποίησης, και Σ_p και Σ_l είναι οι πίνακες συνδιακύμανσης που σχετίζονται με τους όρους σφάλματος επαναπροβολής των σημείων και ευθειών, αντίστοιχα. Το σύνολο $\boldsymbol{\Phi}_k$ περιέχει όλα τα διανύσματα οπτικής ροής από το πλαίσιο συντεταγμένων I_{k-1} στο I_k που αντιστοιχούν στα σημεία και τις ευθείες που συμμετέχουν στο πρόβλημα ελαχιστοποίησης. Το πρόβλημα υλοποιείται με την βιβλιοθήκη g2o [Küm+11], και λύνται μέσω του επαναληπτικού αλγορίθμου Levenberg-Marquardt.

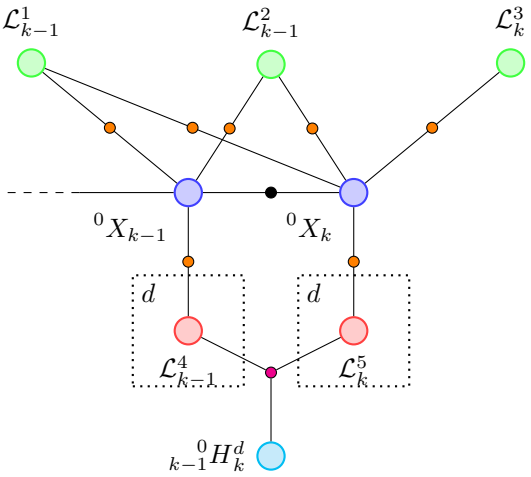
3.5 Εντοπισμός Αντικειμένων και Εκτίμηση κίνησης

Αφού προσδιοριστεί η πόζα της κάμερας, η οπτική ροή χρησιμοποιείται για την συσχέτιση των σημασιολογικών μασκών σε διαδοχικά πλαίσια. Αυτό γίνεται με τον εντοπισμό τον μασκών στα διαδοχικά πλαίσια με τον μεγαλύτερο αριθμό αντιστοιχιών σημείων μεταξύ τους. Ακολούθως, χρησιμοποιείται η ανάλυση ροής σκηνής (scene flow) για να διαχωρίσει τα δυναμικά αντικείμενα από τα στατικά. Συγκεκριμένα, η εκτιμώμενη πόζα χρησιμοποείται για την ευθυγράμμιση των αντίστοιχων παρατηρήσεων σε διαδοχικά πλαίσια, από την οποία προκύπτει μια προσέγγιση των κινήσεων των σημείων. Λαμβάνοντας υπόψιν ότι η ροή σκηνής για στατικά αντικείμενα πρέπει να είναι αμελητέα, εκείνα με μεγάλο αριθμό σημείων που δεν πληρούν αυτήν την προϋπόθεση θεωρούνται δυναμικά.

Μόλις εντοπιστούν τα δυναμικά αντικείμενα, η κίνησή τους εκτιμάται τροποποιώντας ελαφρώς το προβλήμα ελαχιστοποίησης του προηγούμενου υποκεφαλαίου με την εισαγωγή ενός παρόμοιου όρου σφάλματος με (3.6):

$$\mathbf{e}_{j,l} = \mathbf{e}_j({}_{k-1}^0G_k, \boldsymbol{\phi}_k^{j,\mathbf{a}}, \boldsymbol{\phi}_k^{j,\mathbf{b}}) = \begin{bmatrix} \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}_{k-1}^0G_k \mathbf{A}_{k-1}^j) \\ \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}_{k-1}^0G_k \mathbf{B}_{k-1}^j) \end{bmatrix}, \tag{3.12}$$

όπου ή ποσότητα προς εκτίμηση είναι η ${}_{k-1}^0G_k = {}^0X_k^{-1}\,{}_{k-1}^0H_k$ και επομένως, το πρόβλημα



Σχήμα 3.4: **Αναπαράσταση γράφου παραγόντων για ορόσημα ευθειών:** Παρουσιάζονται μόνο τα στατικά και δυναμικά χαρακτηριστικά **ευθειών** και οι περιορισμοί που επιβάλλονται από αυτά. Ημιδιαφανείς Κύκλοι: τρισδιάστατες στατικές ευθείες (Πράσινο), πόζες (Μπλε), τρισδιάστατες δυναμικές ευθείες (Κόκκινο), μετασχηματισμοί κίνησης αντικειμένων (Κυανό). Αδιαφανείς Κύκλοι: περιορισμοί τρισδιάστατων μετρήσεων ευθειών (Πορτοκαλί), περιορισμοί στην κίνηση ευθειών που ανήκουν στο δυναμικό αντικείμενο d (Ροζ), περιορισμοί πόζας (Μαύρο).

ελαχιστοποίησης διατηρεί την μορφή (3.11):

$$\begin{aligned}\{{}^0_{k-1}G^*_k, \mathbf{\Phi}^*_k\} = \text{argmin}_{\{{}^0_{k-1}G_k, \mathbf{\Phi}_k\}} \sum_i^{n_p} \{\boldsymbol{\rho}_h(\mathbf{e}^\top_{i,r}\Sigma^{-1}_{\boldsymbol{\phi}}\mathbf{e}_{i.r}) + \\ \boldsymbol{\rho}_h(\mathbf{e}^\top_{i,p}\Sigma^{-1}_p\mathbf{e}_{i,p})\} + \sum_j^{n_l}\{\boldsymbol{\rho}_h(\mathbf{e}^\top_{j,ra}\Sigma^{-1}_{\boldsymbol{\phi}}\mathbf{e}_{j,ra}) + \\ \boldsymbol{\rho}_h(\mathbf{e}^\top_{j,rb}\Sigma^{-1}_{\phi}\mathbf{e}_{j,rb}) + \boldsymbol{\rho}_h(\mathbf{e}^\top_{j,l}\Sigma^{-1}_l\mathbf{e}_{j,l})\},\end{aligned} \tag{3.13}$$

Αξίζει να τονιστεί ότι ακόμα και αν ένα στατικό αντικείμενο χαρακτηριστεί λαθασμένα αρχικά ως δυναμικό, κατά την διάρκεια αυτού του σταδίου θα βρεθεί ότι δεν παρουσιάζει σχετική κίνηση, λειτουργώντας στην ουσία ως στατικό.

3.6 Χάρτης, Τοπική και Ολική Βελτιστοποίηση Παρτίδας

Κατά τη διάρκεια της εκτέλεσης του SDPL-SLAM [MMM24] διατηρείται ένας χάρτης, ο οποίος περιέχει στατικά και δυναμικά σημεία και ευθείες, πόζες κάμερας και κινήσεις αντικειμένων. Η δομή του χάρτη μπορεί να περιγραφεί ως γράφος, όπως αναλύθηκε στην ενότητα 2.4.

Σε αυτή την διπλωματική εργασία προτείνουμε μια μέθοδο βελτιστοποίησης γράφου για την από κοινού βελτιστοποίηση της τροχιάς της κάμερας, της κίνησης των δυναμικών άκαμπτων αντικειμένων και του χάρτη. Αυτός ο γράφος περικλείει περιορισμούς για τις μεταβλητές

προς εκτίμηση, υπό την μορφή όρων σφάλματος, οι οποίοι συμμετέχουν σε ένα πρόβλημα μη ελαχίστων τετραγώνων , όπως στο [Küm+11]. Συγκεκριμένα, προτείνονται δύο καινοτόμοι περιορισμού ευθειών, (i) ο περιορισμός μέτρησης τρισδιάστατης ευθείας και (ii) ο περιορισμός στην κίνηση ευθειών που ανήκουν σε δυναμικά άκαμπτα αντικείμενα. Οι υπόλοιποι περιορισμοί, που δημιουργούνται από παρατηρήσεις οδομετρίας και σημείων, παραμένουν όπως στο [Zha+21]. Οι καινοτόμοι περιορισμοί (i) και (ii) παρουσιάζονται ως πορτοκαλί και ροζ όροι αντίστοιχα στο Σχήμα 3.4, το οποίο περιέχει μόνο παρατηρήσεις **ευθειών**.

Οι αναπαραστάσεις των ευθειών πρέπει να είναι ελάχιστες ώστε να αποφευχθούν προβλήματα αριθμητικής αστάθειας κατά την βελτιστοποίηση και αυξημένο υπολογιστικό κόστος λόγω επιπρόσθετων βαθμών ελευθερίας. Επιλέγεται η ορθοκανονική αναπαράσταση [BS05] για την ελάχιστη αναπαράσταση των τρισδιάστατων ευθειών.

Το σφάλμα μέτρησης τρισδιάστατης ευθείας ορίζεται ως εξής:

$$\mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j) = \begin{bmatrix} || \, {}^{C_k}\tilde{\mathbf{A}}_k^{j,obs} \times {}^{C_k}\tilde{\mathbf{U}}_k^j - {}^{C_k}\tilde{\mathbf{N}}_k^j || \\ || \, {}^{C_k}\tilde{\mathbf{B}}_k^{j,obs} \times {}^{C_k}\tilde{\mathbf{U}}_k^j - {}^{C_k}\tilde{\mathbf{N}}_k^j || \end{bmatrix}, \tag{3.14}$$

το οποίο αντιστοιχεί στις αποστάσεις [Bro+10] των παρατηρηθέντων τρισδιάστατων άκρων ${}^{C_k}\mathbf{A}_k^{j,obs}$, ${}^{C_k}\mathbf{B}_k^{j,obs}$ από την j ευθεία Πλίκερ στο πλαίσιο k.

Οι ακόλουθες δύο σημειώσεις κρίνονται απαραίτητες. Για τις στατικές ευθείες, ο δείκτης k των στοιχείων ${}^{C_k}\tilde{\mathbf{U}}_k^j$ και ${}^{C_k}\tilde{\mathbf{N}}_k^j$ της ευθείας Πλίκερ, επιλέγεται ως το πρώτο πλαίσιο στο οποίο παρατηρήθηκε η ευθεία j, ενώ στις δυναμικές ευθείες είναι το τρέχον πλαίσιο k. Δεύτερον, οι συντεταγμένες Πλίκερ χρησιμοποιούνται στον υπολογισμό του σφάλματος, ωστόσο οι παράμετροι ενημέρωσης υπολογίζονται για την ορθοκανονική αναπαράσταση των γραμμών.

Ο περιορισμός της κίνησης μιας ευθείας j που ανήκει σε ένα δυναμικό άκαμπτο αντικείμενο d μπορεί να διαιρεθεί σε ένα σφάλμα απόστασης και γωνίας και ορίζεται ως εξής:

$$\mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j) = \begin{bmatrix} \text{dist}(\mathcal{L}_k^j, \mathcal{L}_k^{j,H}) \\ 1 - \frac{\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{U}}_k^{j,H}}{||\tilde{\mathbf{U}}_k^j|| \; ||\tilde{\mathbf{U}}_k^{j,H}||} \end{bmatrix}, \tag{3.15}$$

όπου ${}_{k-1}^{0}H_k^d$ είναι ο μετασχηματισμός κίνησης της ευθείας για το αντικείμενο d. Ο άνω δείκτης 'H' στην ευθεία j στο πλαίσιο k που ανήκει σε ένα αντικείμενο d χρησιμοποίειται για να υποδηλώσει ότι έχει υποβληθεί σε έναν μετασχηματισμό κίνησης ${}_{k-1}^{0}H_k^d$:

$$\mathcal{L}_k^{j,H} = {}_{k-1}^{0}H_k^d \mathcal{L}_{k-1}^j = \begin{bmatrix} \tilde{\mathbf{N}}_k^{j,H} \\ \tilde{\mathbf{U}}_k^{j,H} \end{bmatrix} \tag{3.16}$$

Πρέπει να σημειωθεί ότι για να απλουστευτεί ο συμβολισμός για τις δυναμικές ευθείες, το γεγονός ότι ανήκουν στο αντικείμενο d υποννοείται. Η συνάρτηση dist δίνεται από τον τύπο της απόστασης δύο Πλίκερ ευθειών:

$$\text{dist}(\mathcal{L}_k^j, \mathcal{L}_k^{j,H}) = \begin{cases} \frac{|\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_k^{j,H} + \tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_k^{j,H}|}{||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_k^{j,H}||} & \text{αν } \tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_k^{j,H} \neq 0 \\ \frac{||\tilde{\mathbf{U}}_k^j \times (\tilde{\mathbf{N}}_k^j - \tilde{\mathbf{N}}_k^{j,H}/s)||}{||\tilde{\mathbf{U}}_k^j||^2} & \text{αν } \tilde{\mathbf{U}}_k^{j,H} = s\tilde{\mathbf{U}}_k^j \text{ για κάποιο } s \neq 0 \end{cases} \tag{3.17}$$

4

Πειραματική Αξιολόγηση

Για να υπογραμμίσουμε τη σημασία της ενσωμάτωσης γραμμών στο δυναμικό SLAM, διεξαγάγαμε μια σειρά πειραμάτων σε διάφορα εσωτερικά και εξωτερικά περιβάλλοντα και συγκρίναμε τα αποτελέσματά μας με άλλες σύγχρονες μεθόδους για να δείξουμε την αποτελεσματικότητα της υλοποίησής μας. Συγκεκριμένα χρησιμοποιήθηκαν τα ακόλουθα δύο σύνολα δεδομένων: (i) το KITTI Raw Dataset [Gei+13] και (ii) το Oxford Multimotion Dataset [JG19]. Σε αυτό το κεφάλαιο, παρουσιάζουμε τις διαδικασίες προεπεξεργασίας των συνόλων δεδομένων, τις μετρικές σφάλματος που χρησιμοποιήθηκαν και τα ευρήματά μας σχετικά με την ακρίβεια της εγωκίνησης της κάμερας και των θέσεων των άκαμπτων αντικειμένων.

4.1 Προεπεξεργασία

Για την σημασιολογική κατάτμηση στο KITTI Raw Dataset, χρησιμοποιείται μια υλοποίηση του Mask R-CNN [He+17] με προεκπαιδευμένα βάρη για το σύνολο δεδομένων MS COCO. Για το σύνολο δεδομένων OMD, χρησιμοποιείται μια δικιά μας απλή μέθοδος κατάτμησης βασισμένης στον χρωματικό χώρο HSV, η οποία ακολούθως βελτιώνεται μέσω μορφολογικού φιλτραρίσματος.

Η πυκνή οπτική ροή προκύπτει από την PyTorch εκδοχή του μοντέλου PWC-Net [Sun+18· Nik18] χωρίς αλλαγή των βαρών του.

4.2 Μετρικές Σφάλματος

Για την άμεση σύγκριση των αποτελεσμάτων μας με το VDO-SLAM, χρησιμοποιείται η μετρική που παρουσιάζεται στο άρθρο τους και η υλοποίηση τους [Zha+21]. Για κάθε πλαίσιο, το σφάλμα ορίζεται ως $E = \hat{T}^{-1}T$, όπου $\hat{T}$ είναι ο εκτιμώμενος μετασχηματισμός κίνησης για την κάμερα ή ένα αντικείμενο και T είναι η αντίστοιχη αληθινή κίνηση. Το σφάλμα μετατόπισης E_t είναι η L_2 νόρμα της συνιστώσας μετατόπισης του E, ενώ το E_R είναι η γωνία περιστροφής σε μια αναπαράσταση άξονα-γωνίας της περιστροφικής συνιστώσας του E.

4.3 Αποτελέσματα στο KITTI Raw Dataset και Σχολιασμός

Το KITTI Raw Dataset αποτελείται από πολλές ακολουθίες σε πραγματικά εξωτερικά περιβάλλοντα οδήγησης με δοσμένες τις αληθινές θέσεις της κάμερας και των αντικειμένων. Για να αξιολογίσουμε το σύστημά μας σε μια ποικιλία περιβαλλόντων, επιλέχθηκε ένα σύνολο 13 ακολουθιών με διαφορετικά επίπεδα δυναμικότητας και γεωμετρικής παρουσίας. Τα αποτελέσματα του προτεινόμενου συστήματός μας παρουσιάζονται στον Πίνακα 4.1. Συγκρίνουμε την αποτελεσματικότητα του συστήματός μας με το VDO-SLAM [Zha+21] και τα αναφερόμενα αποτελέσματα του DynaSLAM II [Bes+21], που και τα δύο θεωρούνται σύγχρονα δυναμικά συστήματα SLAM.

Όσον αφορά την εγωκίνηση της κάμερας, η υλοποίησή μας υπερέχει των άλλων δύο συστημάτων σε σχεδόν όλες τις ακολουθίες στο E_t, ενώ είναι ισοδύναμη ή καλύτερη στο E_R.

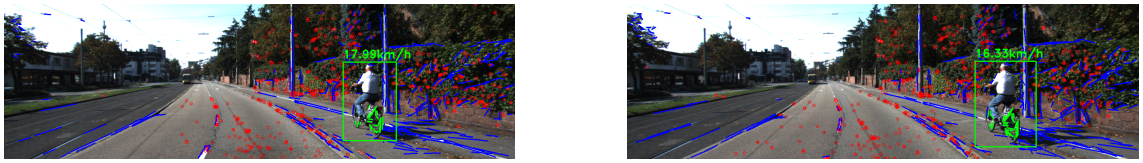
Πίνακας 4.1: Αποτελέσματα KITTI Raw Dataset (E_t[m] και E_R[deg]). *FO = Βελτιστοποίηση Ροής.

Ακολουθία	Μέσο Μήκος Tracklet Στατικών Ευθειών		DynaSLAM II		VDO-SLAM				Υλοποίηση μας (με FO*)				Υλοποίηση μας (χωρίς FO*)			
			Κάμερα		Κάμερα		Αντικείμενα		Κάμερα		Αντικείμενα		Κάμερα		Αντικείμενα	
	με FO*	χωρίς FO*	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R
0926-0001	5.1	3.1	-	-	0.051	**0.056**	0.410	0.439	**0.050**	**0.056**	**0.353**	**0.423**	0.051	0.056	0.450	0.426
0926-0002	5.1	3.0	-	-	0.061	0.067	**0.178**	1.528	**0.055**	**0.066**	0.490	**0.674**	0.055	0.066	0.425	1.142
0926-0005	6.2	3.0	-	-	0.059	0.083	**0.378**	1.988	**0.051**	**0.071**	0.462	**1.799**	0.054	0.071	0.264	1.878
0926-0009	5.6	3.1	1.870	0.573	0.110	**0.065**	0.217	0.188	**0.095**	0.066	**0.211**	**0.165**	0.101	0.060	0.211	0.164
0926-0011	8.0	3.1	-	-	0.043	**0.057**	0.623	1.169	**0.034**	**0.057**	**0.265**	**0.325**	0.037	0.057	0.593	0.816
0926-0013	4.6	3.2	0.930	**0.000**	0.076	0.059	**0.139**	0.390	**0.074**	0.058	1.465	**0.355**	0.079	0.058	1.465	0.369
0926-0014	4.8	3.3	1.350	0.573	**0.108**	0.070	0.988	**2.853**	0.110	**0.069**	**0.811**	3.060	0.110	0.069	0.811	3.229
0926-0051	8.3	3.1	1.140	**0.000**	0.065	0.058	1.067	1.029	**0.061**	0.058	**0.644**	**0.415**	0.072	0.059	0.644	0.416
0926-0091	6.1	3.1	-	-	0.069	0.063	-	-	**0.066**	**0.062**	-	-	0.067	0.062	-	-
0926-0093	6.7	3.0	-	-	2.295	0.085	0.869	1.207	**2.284**	**0.084**	**0.669**	**0.391**	2.285	0.083	0.672	0.393
0926-0101	5.2	3.3	15.020	2.292	**0.570**	**0.072**	-	-	0.585	0.073	-	-	0.647	0.078	-	-
0926-0106	6.9	3.0	-	-	0.047	0.062	-	-	**0.039**	**0.058**	-	-	0.033	0.057	-	-
0929-0004	5.4	3.1	1.410	0.573	0.071	0.058	-	-	**0.065**	**0.057**	-	-	0.062	0.057	-	-

Το DynaSLAM II φαίνεται να επιτυγχάνει χαμηλότερο σφάλμα περιστροφής σε δύο ακολουθίες, ωστόσο, πρέπει να σημειωθεί ότι οι συγγραφείς του παρείχαν αυτά τα αποτελέσματα σε ακτίνια, με αποτέλεσμα την απώλεια δεκαδικής ακρίβειας όταν μετατρέπονται σε μοίρες.

Για να διεξάγουμε μια πιο ολοκληρωμένη ανάλυση των αποτελεσμάτων, έχουμε ενσωματώσει στον Πίνακα 4.1 μια μετρική που αντιστοιχεί στον μέσο αριθμό καρέ στον οποίο εντοπίζονται οι στατικές ευθείες. Θα αναφερόμαστε εφεξής σε ακολουθίες εντοπισμένων χαρακτηριστικών σε πολλές διαδοχικές χρονικές στιγμές ως παρατηρήματα. Το σύστημά μας επιδεικνύει τη σημαντικότερη βελτίωση στις ακολουθίες 0926-(0009, 0011, 0093, 0005, 0106), οι οποίες, εκτός από την 0926-0011, έχουν έντονη παρουσία κοντινών κτιρίων που παρέχουν πολλές ευθείες υψηλής ποιότητας για ανίχνευση. Αυτό μεταφράζεται άμεσα σε υψηλότερες τιμές στην προαναφερθείσα μετρική, τονίζοντας τη σημασία των ευθειών υψηλής ποιότητας που μπορούν να εντοπιστούν με συνέπεια. Ενδιαφέρον παρουσιάζει ότι η σημαντικά βελτιωμένη απόδοση στην ακολουθία 0926-0011 δικαιολογείται, παρά την απουσία κοντινών κτιρίων, μέσω της επίδειξης μιας από τις υψηλότερες τιμές της μετρικής (8.0). Αντιθέτως, το σύστημά μας αποδίδει ελαφρώς χειρότερα στις ακολουθίες 0926-0014 και 0926-0101, οι οποίες χαρακτηρίζονται από ανοιχτούς χώρους και έλλειψη κτιρίων. Οι ευθείες ανιχνεύονται κυρίως στον δρόμο και στα φύλλα δέντρων που βρίσκονται μακριά από την κάμερα, προκαλώντας έτσι υποβάθμιση των αποτελεσμάτων. Αυτό αντικατοπτρίζεται στις τιμές της μετρικής αυτών των ακολουθιών, με το μέσο μήκος των στατικών παρατηρημάτων ευθειών (4.8 και 5.2) να είναι σημαντικά κάτω από τον συνολικό μέσο όρο (6), υπογραμμίζοντας τη συσχέτιση μεταξύ ευθειών χαμηλής ποιότητας και μειωμένης ακρίβειας.

Όσον αφορά την ακρίβεια εντοπισμού των δυναμικών αντικειμένων, η ενσωμάτωση ευθειών βελτιώνει τα αποτελέσματα στην πλειοψηφία των ακολουθιών, κάτι που μπορεί να αποδοθεί στο γεγονός ότι τα περισσότερα δυναμικά αντικείμενα είναι αυτοκίνητα που παρέχουν πολλά ευθύγραμμα τμήματα προς ανίχνευση σε μέρη όπως τα παράθυρα και οι πινακίδες κυκλοφορίας. Οι μόνες ακολουθίες στις οποίες η υλοποίησή μας δεν βελτιώνει τα αποτελέσματα είναι οι 0926-(0002, 0005, 0013). Μια λεπτομερής ποιοτική ανάλυση αποκάλυψε ότι σε δύο από αυτές (0926-0002, 0926-0005), η πλειοψηφία των δυναμικών αντικειμένων που ανιχνεύθηκαν και εντοπίστηκαν είναι κινούμενα ποδήλατα με ανθρώπους (βλ. Σχήμα 4.1 και Σχήμα 4.2), τα



Σχήμα 4.1: Στις ακολουθίες 0926-0002, ένα μεγάλο μέρος της δυναμικότητας της σκηνής οφείλεται σε ποδήλατα, που παρέχουν ευθύγραμμα τμήματα (πράσινες ευθείες στην εικόνα) προς ανίχνευση στις ρόδες τους. Αυτό οδηγεί σε μείωση στην ακρίβεια της εντοπισμού αντικειμένων σε αυτή την συγκεκριμένη ακολουθία.

οποία δεν ακολουθούν την υποκείμενη υπόθεση ακαμψίας. Ως εκ τούτου, οι γραμμές στους τροχούς των ποδηλάτων ή οι γραμμές στα πόδια των ποδηλατών συμβάλλουν στην υποβάθμιση των αποτελεσμάτων σε αυτές τις δύο περιπτώσεις. Ωστόσο, πρέπει να τονιστεί ότι ακόμα και σε αυτές, το E_R των αντικειμένων βελτιώνεται σημαντικά.

Σχήμα 4.2: Σε πολλές εικόνες της ακολουθίας 0926-0005, ανιχνεύονται και εντοπίζονται ευθείες σε ποδηλάτες, οι οποίες παραβιάζουν την υπόθεση ακαμψίας, με αποτέλεσμα μείωση στην ακρίβεια εντοπισμού αντικειμένων.

Τέλος για να αξιολογήσουμε την επίδραση της βελτιστοποίησης της οπτικής ροή στην ακρίβεια του συστήματος, διεξαγάγαμε μια μελέτη αφαίρεσης (βλ. τελευταία στήλη του Πίνακα 4.1), μέσω τροποποιήσεων στις (3.6) και (3.12), αφαιρώντας την εξάρτηση από την οπτική ροή από τους όρους σφάλματος. Αυτό είχε ως αποτέλεσμα λιγότερα συνεπή ταιριάσματα ευθύγραμμων τμημάτων, με σαφή μείωση στο μέσο μήκος των στατικών παρατηρημάτων ευθειών και μια επιδείνωση της απόδοσης τόσο στις μετρικές E_t της κάμερας όσο και στις μετρικές E_R των αντικειμένων. Το γεγονός ότι οι ακολουθίες 0926-0002 και 0926-0005 αποδίδουν χειρότερα στην ακρίβεια της θέσης των αντικειμένων όταν βελτιστοποιείται ταυτόχρονα η οπτική ροή, υποστηρίζει τα ευρήματα της προηγούμενης παραγράφου, καθώς διατηρούνται περισσότερες αντιστοιχίες γραμμών σε μη άκαμπτα αντικείμενα, με αποτέλεσμα να μεγεθύνεται το πρόβλημα.

4.4 Αποτελέσματα στο Oxford Multiomotion Dataset (OMD) και Σχολιασμός

Το Oxforf Multimotion Dataset αποτελείται από ακολουθίες εικόνων που έχουν καταγραφεί σε εσωτερικό περιβάλλον στο οποίο υπάρχουν κινούμενα μικρά αυτοκίνητα ή αιωρούμενοι κύβοι. Αυτό το σύνολο δεδομένων χαρακτηρίζεται από ισχυρή γεωμετρική δομή, καθώς τόσο το στατικό περιβάλλον όσο και οι κινούμενοι κύβοι παρέχουν πολλά ευθύγραμμα τμήματα υψηλής ποιότητας προς ανίχνευση. Αυτό το γεγονός καθιστά το OMD ένα ιδανικό σενάριο

Πίνακας 4.2: Αποτελέσματα OMD (E_t[m] και E_R[deg]).

	VDO-SLAM		Υλοποίηση μας	
	E_t	E_R	E_t	E_R
Ολόκληρη Ακολουθία: Κάμερα	0.038	0.578	**0.022**	**0.507**
Ολόκληρη Ακολουθία: Μέσος Όρος Κύβων	0.032	1.286	**0.029**	**1.231**
500 εικόνες: Κάμερα	0.017	0.466	**0.014**	**0.453**
500 εικόνες: Πάνω Δεξιά	0.033	1.369	**0.032**	**1.367**
500 εικόνες: Κάτω Δεξιά	0.030	1.166	**0.029**	**1.164**
500 εικόνες: Πάνω Αριστερά	0.036	1.494	**0.031**	**1.452**
500 εικόνες: Κάτω Αριστερά	**0.027**	**1.601**	**0.027**	1.605
500 εικόνες: Μέσος Όρος Κύβων	0.032	1.407	**0.030**	**1.397**

για να αναδείξουμε την επίδραση των ευθειών. Η απόδοση του συστήματος αξιολογείται αποκλειστικά στην ακολουθία με τα αιωρούμενα κουτιά και συγκεκριμένα στην περίπτωση της μη-περιορισμένης κίνησης της κάμερας, ένα δύσκολο και ρεαλιστικό σενάριο. Δοκιμάσαμε το σύστημα μας τόσο στις αρχικές 500 εικόνες για σύγκριση με το [Zha+21], όσο και σε όλο το σύνολο των δεδομένων για να αξιολογήσουμε την ευρωστία του συστήματος μας σε μια μακροχρόνια ακολουθία.

Όπως φαίνεται στον Πίνακα 4.2, το σύστημα μας υπερέχει του VDO-SLAM τόσο στην εγωκίνηση όσο και στην ακρίβεια της θέσης των τεσσάρων αιρούμενων κουτιών, το οποίο εξηγείται εύκολα, αν ληφθεί υπόψη ότι η δοκιμή πραγματοποιείται σε εσωτερικό χώρο και τα δυναμικά αντικείμενα στην ακολουθία είναι κύβοι. Συγκεκριμένα, στην πλήρη ακολουθία (και στις πρώτες 500 εικόνες), επιτυγχάνεται βελτίωση κατά περίπου 42% (περίπου 18%) και περίπου 12% (περίπου 2,8%) στις μετρικές E_t και E_R της κάμερας, αντίστοιχα, σε σύγκριση με το VDO-SLAM. Επιπλέον, η ενσωμάτωση ευθειών βελτίωσε την ακρίβεια εκτίμησης της θέσης των κινούμενων κύβων στην πλήρη ακολουθία και είχε οριακές βελτιώσεις στις πρώτες 500 εικόνες, μειώνοντας το μέσο E_t κατά περίπου 9,4% (περίπου 6,3%) και το E_R κατά περίπου 4,3% (περίπου 0,7%).

4.5 Επισκόπηση Αποτελεσμάτων

Δείξαμε ότι η ενσωμάτωση ευθειών είχε ως συνέπεια την βελτίωση της συνολικής απόδοσης, τόσο στην εγωκίνηση όσο και στον εντοπισμό δυναμικών αντικειμένων, σε σενάρια εξωτερικής οδήγησης (Πίνακας 4.1) και εσωτερικού χώρου (Πίνακας 4.2). Επιπρόσθετα, εισάγαμε το μέσο μήκος των στατικών παρατηρημάτων ευθείας, το οποίο ποσοτικοποιεί την ποιότητα και την ευρωστία των ευθύγραμμων τμημάτων. Μια λεπτομερής ανάλυσης επαλήθευσε την υψηλή συσχέτιση αυτής της μετρικής και της βελτίωσης της ακρίβειας της υλοποίησης μας σε σύγκριση με άλλα σύγχρονα συστήματα. Η χρήση της οπτικής ροής για το ταίριασμα ευθειών

παρέχει καλύτερες και περισσότερες αντιστοιχίες ευθειών, με αποτέλεσμα μακροχόνια και συνεπή παρατηρήματα ευθειών, ένα πλεονέκτημα που αποδείχτηκε ότι ενισχύεται περαιτέρω από την ταυτόχρονη βελτιστοποίηση της οπτικής ροής στο στάδιο του εντοπισμού.

5 Επίλογος

5.1 Σύντομη Περίληψη, Συμπεράσματα και Μελλοντικές Ερευνητικές Κατευθύνσεις

Στην παρούσα διπλωματική εργασία παρουσιάσαμε ένα καινοτόμο σύστημα SLAM, το οποίο εκμεταλλεύεται ευθείες που ανιχνεύονται τόσο σε στατικά όσο και σε δυναμικά αντικείμενα, προκειμένου να εκτιμήσει την τροχιά της κάμερας και τις κινήσεις των αντικειμένων. Στο πρώτο μέρος της εργασίας, παρουσιάσαμε και αναλύσαμε βασικές έννοιες του προβλήματος SLAM, καθώς και τη σχετική έρευνα που έχει ήδη διεξαχθεί στον τομέα, προκειμένου να παρέχουμε το απαραίτητο υπόβαθρο για το προτεινόμενο σύστημά μας και να προσδιορίσουμε τους περιορισμούς των υφιστάμενων μεθόδων που οδήγησαν στην ανάπτυξη του αλγορίθμου μας.

Στα επόμενα κεφάλαια, παρουσιάσαμε την προσέγγισή μας σε βάθος, αναλύοντας κάθε συνιστώσα και μεθοδολογία που χρησιμοποιήθηκε. Εξηγήσαμε τη λογική πίσω από τις επιλογές μας και αναλύσαμε τις συνεισφορές μας στα νέα προβλήματα βελτιστοποίησης, που ενσωματώνουν παρατηρήσεις ευθειών για τη βελτίωση του εντοπισμού της κάμερας, των δυναμικών αντικειμένων, και των θέσεων των στατικών και δυναμικών χαρακτηριστικών στον χάρτη. Παρουσιάσαμε σύνολα δεδομένων που θέτουν σημαντικές προκλήσεις στους αλγορίθμους SLAM, επιτρέποντας μας να δοκιμάσουμε το σύστημα μας σε διάφορα σενάρια και να αναδείξουμε τις δυνατότητες του. Η πειραματική μας αξιολόγηση αποκάλυψε ότι η αξιοποίηση της γραμμική δομής του περιβάλλοντος έχει ως αποτέλεσμα την συνολική αύξηση της ακρίβειας και της ευρωστίας του SLAM αλγορίθμου σε σύγκριση με άλλα σύγχρονα συστήματα που βασίζονται σε σημεία.

Παρ΄όλο που η πρόοδος στην περιοχή του δυναμικού SLAM είναι ραγδαία, υπάρχουν ακόμα πολλά ανοιχτά προβλήματα τα οποία πρέπει να αντιμετωπιστούν. Κάποιες από αυτές τις προκλήσεις έγιναν εμφανείς στη διάρκεια αυτής της διπλωματικής, παρέχοντας μας πολλά υποσχόμενες μελλοντικές ερευνητικές κατευθύνσεις. Μελλοντικός μας στόχος είναι η προσπάθεια αντιμετώπισης του προβλήματος της ύπαρξης ανθρώπων, οι οποίοι αποτελούν ένα σημαντικό ποσοστό των καμπτών οντοτήτων στα περιβάλλοντα, επεκτείνοντας την υλοποίηση

μας ώστε να είναι σε θέση να χειριστεί τις ανεξάρτητες κινήσεις των γραμμικών σκελετικών μέρων τους.

6 Introduction

6.1 Introduction to visual SLAM

Robotics is a rapidly evolving field that focuses on the design, construction and operation of robots, which are machines that are employed to perform tasks in a designated and controlled manner, usually without the need for human interaction. The end goal of robotics is to create truly autonomous and mobile robots that can operate in a variety of environments and perform complex and diverse tasks. The importance of autonomous mobile robots has already started to become apparent in a plethora of applications, such as autonomous driving, deep sea discovery, space exploration, rescue missions, fire prevention, agriculture, elderly care, and many more. The advancement of robots can contribute to a significant improvement in the quality of life of humans, the protection of the environment and financial growth.

Robot autonomicity is a challenging task that is directly related to the robot's ability to perceive, understand and interact with its environment, made possible by the existence of a variety of sensors. The enhancement of these abilities has given rise to different areas of research in robotics, posed by problems that require the cooperation of multiple disciplines, such as computer vision, artificial intelligence, control theory, optimization and others. One of the most important problems in robotics is the Simultaneous Localization and Mapping (SLAM) problem, which is crucial for the perception of the robot's environment and its ability to navigate and act in it.

SLAM is a fundamental and well-studied area of Robotics and Computer Vision [Cad+16; Ros+21a], with applications in a wide range of applications, including autonomous driving, augmented reality and house robots. SLAM aims to find the most probable trajectory of a robot, given its sensor measurements, while buidling at the same time a map of the environment. Map existence prevents accumulation of pose drift caused by noisy sensor measurements, while it also provides meaningful information about the topology of the environment (see Figure 6.1). Different sensors have been utilized to solve this problem, such as cameras, in which case it is termed as visual SLAM (vSLAM), IMUs and LiDAR. Advances in camera technology and easier access to high-quality cameras, such as RGB-D cameras, have led to the development of many robust visual SLAM systems.

SLAM algorithms have diversified in many ways, thus creating a lot of areas of open research. For example, various structural elements, such as sparse points [KM07; MT17], voxels [Ros+21b], surfels [Sco+18] or other geometric entities like lines and planes, are used for map representation. Likewise, tracking in visual SLAM is performed either directly [ESC14] or by detecting features, like ORB [Rub+11] or more complex geometric shapes such as lines [Gom+19], planes [Kae15] or both [Zhe+22].

While there have been many advances in visual SLAM and in SLAM in general, with many robust and efficient systems being developed, there are still many challenges that need to be addressed. One fundamental problem is the ability of SLAM systems to handle dynamic environments, where objects move, are added or are removed from the scene. Traditionally in SLAM research, the world was assumed to be static and measure-

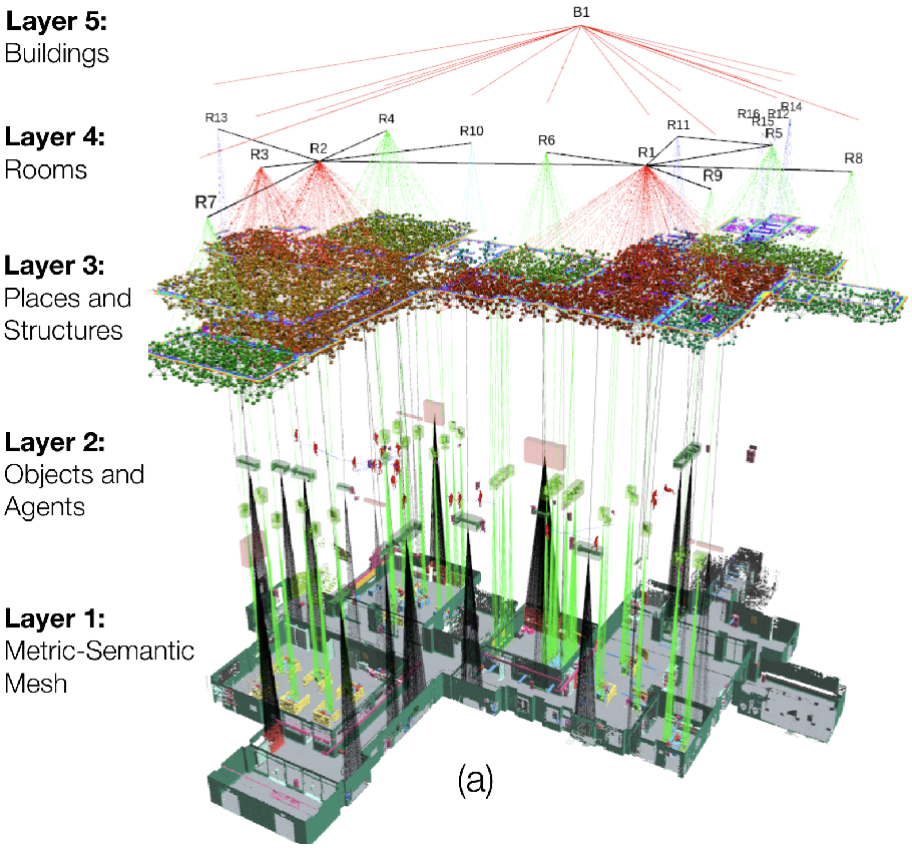


Figure 6.1: Example of a map created by Kimera [Ros+21b], a state-of-the-art visual SLAM system. The map has multiple layers of abstraction, from the low-level Metric-Semantic Mesh to the top-level of Buildings. This system underlines the capabilities provided to robots by modern visual SLAM systems, by enabling them to perceive the true topology of their environment.

ments on dynamic objects were handled with generic outlier rejection techniques, such as RANSAC [FB81], and robust loss functions, like the Huber loss function. This approach, however, is error-prone in highly dynamic environments, and due to the nonconvexity of the minimization problem used in SLAM, persisting outlier observations can prove detrimental to overall system accuracy. Therefore, it becomes apparent that the development of robust dynamic SLAM systems is vital for the operation of robots in real-life environments, which are dominated by humans, cars, and other moving objects.

Even though it has been proven that the use of more complex geometric shapes such as lines increases the robustness of SLAM [Gom+19; Pum+17], especially in textureless and low-lit areas, little research has been done on their use in dynamic environments. Motivated by this and by the need for accurate SLAM systems in human-centered environments, we propose a SLAM system that tracks static and dynamic points and lines to estimate camera positions and motion of dynamic objects in the scene (see Fig. 6.2).

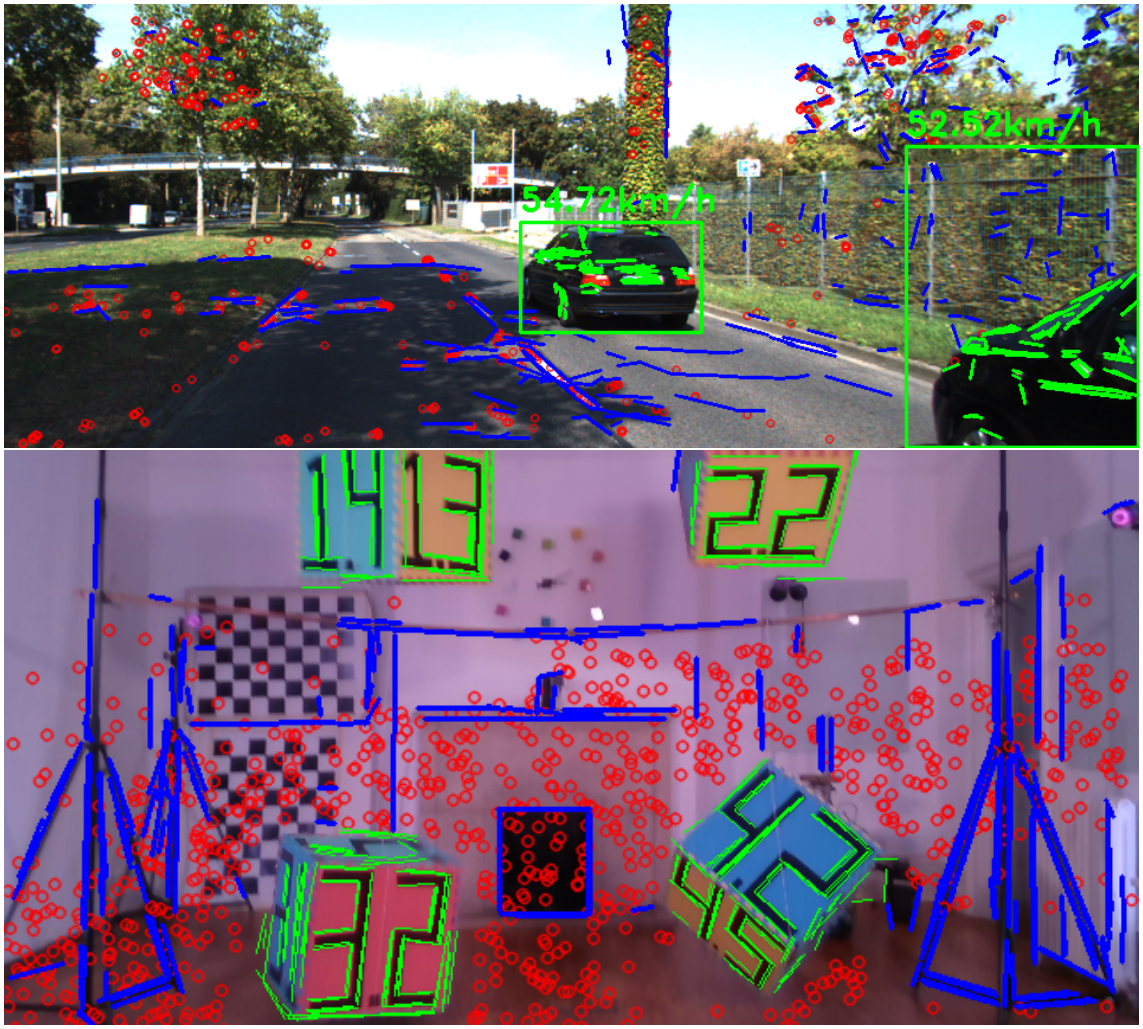


Figure 6.2: **Output of our system:** Points and lines are tracked on both static and dynamic objects. Features presented: static points (Red), static lines (Blue), and dynamic lines (Green). Speed calculated from the estimated motion of cars is shown.

6.2 Our approach, our contributions and structure of the Thesis

The focus of this thesis was to familiriaze with the field of visual SLAM and it's underlying principles, and to develop an algorithm that introduces lines as a geometric primitive in dynamic SLAM, in order to enhance the accuracy and robustness of the system in dynamic scenarios. Initially, there has been an extensive review of the existent literature, with focus on various formulations of the SLAM problem, the optimization techniques leveraged, the different features used for tracking and mapping, the already existing SLAM systems and especially those which have been developed for dynamic environments. Taking into consideration the strengths and weaknesses of the techniques employed, we have developed a novel SLAM system, with contributions and novelties present in every aspect

of our implementation including:

- The usage of optical flow for richer line correspondences.
- Introduction of line reprojection error terms for camera tracking and object motion estimation, with the concurrent optimization of optical flow as a two-fold contribution.
- Inclusion of lines in partial and global batch optimization, with the introduction of novel cost functions.
- Verification of our method on challenging datasets and comparison against state-of-the-art dynamic SLAM systems.

Combining the advantages of dynamic and line SLAMs, we developed a system that surpasses other state-of-the-art systems and verified its performance on both outdoor driving and dynamic indoor datasets. The rest of the thesis is structured as follows:

- In Chapter 7, we present the theoretical background of SLAM, as well as related work.
- In Chapter 8, we present an overview of VDO-SLAM, a state-of-the-art system that has been the basis of our implementation, and in great detail SDPL-SLAM, our novel system.
- In chapter 9, we present the experimental evaluation of our system.
- In Chapter 10, we conclude our work and give directions for future research.
- Lastly, in the Appendix, we provide mathematical derivations and additional information about our system.

7

Background and Related Work

In this chapter we are going to present the core concepts of the area of SLAM, as well as related work, which has already been conducted in the field. As mentioned earlier, since the measurements produced by sensors are noisy, the aim of SLAM is not to recover the true state of the world, but to infer the most fitting estimate based on given observations, and therefore the problem had to be formulated in a probabilistic manner. Specifically we assume that the state $\mathbf{x}_t$ of a robot at time t, knowing the input to its actuators $\mathbf{u}_t$, can be estimated from the state at the previous time step $\mathbf{x}_{t-1}$ through a motion model:

$$P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t). \tag{7.1}$$

The state $\mathbf{x}_t$ produces observations $\mathbf{z}_t$ through an observation model:

$$P(\mathbf{z}_t|\mathbf{x}_t). \tag{7.2}$$

Combining these a priori known models with measurements produced during the robot navigation, an estimate for the state of the world can be infered with various methods that will be analyzed in the following sections.

7.1 Initial Approach to the SLAM problem with filtering methods

In the early stages of SLAM research, the most dominant approach to the solution of SLAM was through probabilistic filtering, under which only the last state of the robot or camera is estimated. Specifically, given a series of observations $\mathbf{z}_{1:t}$ and actuator controls $\mathbf{u}_{1:t}$, the goal is to determine the probability distribution of robot state, $\mathbf{x}_t$. Following the conventions of [TBF05], we denote the belief that a robot is in a state $\mathbf{x}_t$ by $\text{bel}(\mathbf{x}_t)$:

$$\text{bel} = P(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \quad \overline{\text{bel}} = P(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \tag{7.3}$$

The difference between the above two definitions is caused by the incorporation or not of the measurement $\mathbf{z}_t$ in the inference of belief. Having defined these, the Bayes Filter Algorithm is the following:

Algorithm 4: Modified Bayes Filter Algorithm [TBF05]

Data: $\text{bel}(\mathbf{x}_{t-1}), \mathbf{u}_t, \mathbf{z}_t$
Result: $\text{bel}(\mathbf{x}_t)$
forall $\mathbf{x}_t$ **do**
  $\overline{\text{bel}}(\mathbf{x}_t) = \int P(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1})\text{bel}(\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1}$
  $\text{bel}(\mathbf{x}_t) = \eta P(\mathbf{z}_t|\mathbf{x}_t)\overline{\text{bel}}(\mathbf{x}_t)$
end
return $bel(\mathbf{x}_t)$

As can be observed, Algorithm 4 divides its execution iteratively in two steps (i) an estimation step that utilizes the motion model and the belief distribution of the previous state $\mathbf{x}_t$ and (ii) a corrective step which incorporates the last measurement.

In order to use Bayes Filter Algorithm in practice various assumptions are usually made:

- Linearity of motion model with added gaussian noise ϵ_t:

$$\mathbf{x}_t = A_t\mathbf{x}_{t-1} + B_t\mathbf{u}_t + \epsilon_t \tag{7.4}$$

- Linearity of measurement model with added gaussian noise δ_t:

$$\mathbf{z}_t = C_t\mathbf{x}_t + \delta_t \tag{7.5}$$

- The initial belief $\text{bel}(\mathbf{x}_0)$ is a gaussian distribution, with mean $\boldsymbol{\mu}_0$ and covariance matrix Σ_0.

Under these conditions, due to the properties of gaussian distributions, it can be proven that the belief distribution $\text{bel}(\mathbf{x}_t)$ is always a gaussian distribution. By appropriately modifying Algorithm 4 in accordance with the above assumptions, the well-known Kalman Filter Algorithm is defined as a special case of the Bayes Filter Algorithm:

Algorithm 5: Kalman Filter Algorithm [TBF05]

Data: $\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}, \mathbf{u}_t, \mathbf{z}_t$

Result: $\boldsymbol{\mu}_t, \Sigma_t$

1 $\bar{\boldsymbol{\mu}}_t = A_t\boldsymbol{\mu}_{t-1} + B_t\mathbf{u}_t$

2 $\bar{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + R_t$

3 $K_t = \bar{\Sigma}_t C_t^T(C_t\bar{\Sigma}_t C_t^T + Q_t)^{-1}$

4 $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\mathbf{z}_t - C_t\bar{\boldsymbol{\mu}}_t)$

5 $\Sigma_t = (I - K_tC_t)\bar{\Sigma}_t$

6 **return** $\boldsymbol{\mu}_t, \Sigma_t$

The belief $\text{bel}(\mathbf{x}_t)$ is represented through the mean $\boldsymbol{\mu}_t$ and covariance matrix Σ_t of its gaussian distribution, R_t, Q_t are the covariance matrices of gaussian noises ϵ_t and δ_t respectively, and K_t, called Kalman gain, specifies the influence that the discrepancy between the estimated measurement and the actual measurement will have on the new state estimate.

Even though Kalman filtering has been very popular in many applications, in SLAM assumptions about the linearity of the probabilistic models are very strict, and the transitions can be better described as follows:

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) + \epsilon_t \quad \mathbf{z}_t = h(\mathbf{x}_t) + \delta_t \tag{7.6}$$

where g and h are non-linear functions. In an attempt to address this problem Extended Kalman Filter (EKF) was introduced, which utilizes the first-order Taylor expansion to create local linear approximations of functions g and h, thus avoiding the overgeneralization of linearity across their entire domain space, allowing for more accurate modeling of the system's dynamics under non-linear conditions.

Algorithm 6: Extended Kalman Filter Algorithm [TBF05]

Data: $\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}, \mathbf{u}_t, \mathbf{z}_t$

Result: $\boldsymbol{\mu}_t, \Sigma_t$

1 $\bar{\boldsymbol{\mu}}_t = g(\boldsymbol{\mu}_{t-1}, \mathbf{u}_t)$

2 $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

3 $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

4 $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\mathbf{z}_t - h(\bar{\boldsymbol{\mu}}_t))$

5 $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

6 **return** $\boldsymbol{\mu}_t, \Sigma_t$

G_t, H_t are the Jacobians of g and h respectively. It is important to note that the results of EKF algorithm depend heavily on the quality of the linearization and the choice of initial state. The EKF Algorithm is performed iteratively in every time step, and for each measurement the mean and covariance matrices are updated accordingly, by adding new terms in the case this measurement corresponds to a newly observed landmark in the environment, or by updating the existing terms in the case of a known landmark.

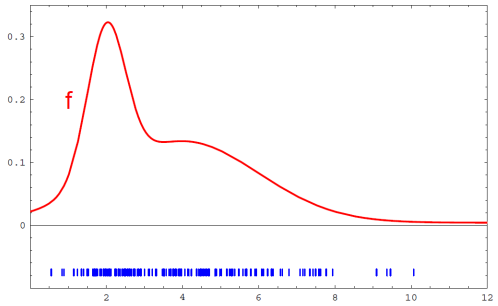
To clarify the implementation details and computational demands of Extended Kalman filtering in SLAM, we will consider a 2D SLAM scenario. In this case robot's state is represented by vector $\mathbf{x} = [x, y, \theta]$, where x, y are the coordinates of the robot in the environment and θ is the orientation of the robot. The i-th's landmark position gaussian distribution is described by its 2-dimensional mean $\mathbf{m}_i = [x_i, y_i]$ and corresponding covariance matrix. Therefore, for N observed landmarks, the state vector in the EKF algorithm would be $\mathbf{x}_t = [x, y, \theta, x_1, y_1, \ldots, x_N, y_N]^T$, having a size of $3 + 2N$ and the covariance matrix Σ_t would have a size of $(3 + 2N) \times (3 + 2N)$. As a result, the EKF algorithm has a complexity of $O(N^3)$, which can be attributed to the inversion of the $(3+2N)\times(3+2N)$ covariance matrix in every iteration. This complexity places a constraint on the number of landmarks that can be handled by the EKF algorithm, thus rendering this approach unsuitable for large-scale SLAM problems.

7.1.1 Non-parametric methods

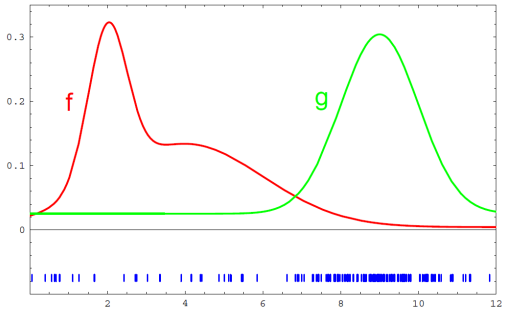
The methods analyzed in the previous sections are governed by the assumption that bel follows a gaussian distribution, which although generally effective, has certain limitations. Specifically, because of the unimodal nature of the gaussian distribution, Kalman filter-based algorithms fail to correctly represent multiple hypotheses about the state of the robot, an ability that is crucial in environments where similar objects or backgrounds can cause ambiguous measurements and thus multiple modes in the belief distribution. In order to address this issue, non-parametric methods have been explored, since they are able to represent complex distributions that lack a straightforward analytical form by utilizing a sampling technique similar to Monte Carlo methods to approximate the distribution. One of the most common non-parametric methods is the Particle Filter, which approximates the belief distribution by a set of particles $\mathbf{x}_t^{[1]}, \mathbf{x}_t^{[2]}, \ldots, \mathbf{x}_t^{[M]}$, distributed in accordance

with the belief. The algorithm can be broken down into the following steps using the terminology of the previous sections:

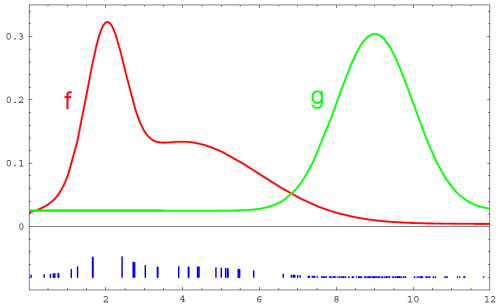
- **Initial Sampling**: M initial particles are sampled from $\overline{\text{bel}}(\mathbf{x}_t)$ if there is a previous time step or from the initial belief distribution $\overline{\text{bel}}(\mathbf{x}_0)$ if t=0.
- **Weight Update**: Weight is attached to each particle of the previous step based on the likelihood of measurement given their robot state hypothesis $\mathbf{x}_t$, ie. $P(\mathbf{z}_t|\mathbf{x}_t)$. A visualization of this process can be seen in figure 7.1.
- **Resampling**: Particles are resampled with a likelihood proportional to their corresponding weights, thus retaining particle hypotheses that are more likely considering the new observations.



The aim is to approximate distribution f, which represents bel distribution through a number of samples



Samples can only be generated from a distribution g, which represents $\overline{\text{bel}}$.



Weights $f(x)/g(x)$ are given to each sample. Samples with bigger weights are illustrated with longer blue lines at the bottom.

Figure 7.1: Illustration of sample weight allocation, based on target ($f = \text{bel}$) and proposal distribution ($g = \overline{\text{bel}}$) [TBF05].

Given that the number of particles is large enough the above procedure can approximate the belief distribution, with state spaces that contain denser regions of particles having higher probability. A successful SLAM algorithm that utilizes the Particle Filter is the FastSLAM algorithm [Mon+02], which is based on the Rao-Blackwellized Particle

Filter [Dou+13]. Another approach to address the unimodality of gaussian distributions is presented in Section 7.5

7.2 Maximum a Posteriori Method

In most recent years, research in SLAM favored the use of maximum a posteriori (MAP) methods instead of the filtering methods [SMD10] described in the previous sections. MAP in SLAM has been based on the seminal work in [LM97] and aims to find the most probable set of robot pose trajectory and map state given independent measurements at each time frame. Namely, let $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote the series of non-observable robot states and $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$ the series of observable measurements. With this notation, finding the most probable robot trajectory would be equivalent to maximizing the following posterior probability:

$$\text{argmax}_{\mathbf{x}}\, P(\mathbf{x}|\mathbf{z}) = \text{argmax}_{\mathbf{x}}\, \frac{P(\mathbf{z}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{z})} = \text{argmax}_{\mathbf{x}}\, P(\mathbf{z}|\mathbf{x})P(\mathbf{x}) \tag{7.7}$$

In the above equation, the Bayes rule was used for the derivation. In addition, assuming a unified probability for all robot states and independency of measurements, equation (7.7) can be further simplified to the main optimization problem of MAP SLAM:

$$\text{argmax}_{\mathbf{x}}\, P(\mathbf{x}|\mathbf{z}) = \text{argmax}_{\mathbf{x}}\, P(\mathbf{z}|\mathbf{x}) = \text{argmax}_{\mathbf{x}} \prod_{k=1}^{K} P(\mathbf{z}_k|\mathcal{X}_k) \tag{7.8}$$

where $\mathcal{X}_k$ is the subset of variable observation $\mathbf{z}_k$ depends upon. The distribution $P(\mathbf{z}_k|\mathcal{X}_k)$ is usually assumed to be gaussian, with its mean being the predicted measurement $\hat{\mathbf{z}}_k$ and its covariance matrix Ω_k being the uncertainty of the measurement. The predicted measurement is calculated from the measurement model h_k as $\hat{\mathbf{z}}_k = h_k(\mathcal{X}_k)$. The factors in the optimization problem can thus be rewritten as:

$$P(\mathbf{z}_k|\mathcal{X}_k) \propto \exp\left(-\frac{1}{2}(\mathbf{z}_k - h_k(\mathcal{X}_k))^T \Omega_k^{-1} (\mathbf{z}_k - h_k(\mathcal{X}_k))\right) \tag{7.9}$$

As is often the case in optimization problems that involve gaussian probability distributions, the above equation is transformed into a minimization problem by taking the negative logarithm of the probability distribution, resulting in the following cost function:

$$\text{argmin}_{\mathbf{x}} \sum_{k=1}^{K} (\mathbf{z}_k - h_k(\mathcal{X}_k))^T \Omega_k^{-1} (\mathbf{z}_k - h_k(\mathcal{X}_k)) \tag{7.10}$$

Paying closer attention to the above equation, it can be observed that the initial MAP problem is equivalent to a non-linear least squares optimization problem, consisting of energy terms that are proportional to the squared error $\mathbf{e}_k = \mathbf{z}_k - h_k(\mathcal{X}_k)$ between the predicted measurement and the actual measurement. Therefore the MAP SLAM problem often reduces to finding the suitable energy functions that accurately describe the constraints induced by the robot's motion and measurements.

Because this optimization problem is non-linear and non-convex, there is no closed-form solution, and as a result, iterative optimization methods are employed to solve it. However, even these iterative methods require the linearization of the cost functions, which can be achieved through their first-order Taylor expansion in the neighborhood of the initial estimate of robot state $\check{\mathbf{x}}$:

$$\mathbf{e}_k(\check{\mathbf{x}} + \Delta\mathbf{x}) \approx \mathbf{e}_k(\check{\mathbf{x}}) + \mathbf{J}_k\Delta\mathbf{x} \tag{7.11}$$

where $\mathbf{J}_k$ is the Jacobian of h_k with respect to $\mathbf{x}$ evaluated at $\check{\mathbf{x}}$. Defining $\mathrm{F} = \sum_{k=1}^{K} \mathrm{F}_k = \sum_{k=1}^{K} \mathbf{e}_k^T \Omega_k^{-1} \mathbf{e}_k$, and subsituting (7.11) the following is obtained:

$$\begin{aligned}
\mathrm{F}_k(\check{\mathbf{x}} + \Delta\mathbf{x}) &= \mathbf{e}_k(\check{\mathbf{x}} + \Delta\mathbf{x})^T \Omega_k^{-1} \mathbf{e}_k(\check{\mathbf{x}} + \Delta\mathbf{x}) \approx (\mathbf{e}_k(\check{\mathbf{x}})^T + \mathbf{J}_k\Delta\mathbf{x})^T \Omega_k^{-1} (\mathbf{e}_k(\check{\mathbf{x}}) + \mathbf{J}_k\Delta\mathbf{x}) \\
&= \underbrace{\mathbf{e}_k(\check{\mathbf{x}})^T \Omega_k^{-1} \mathbf{e}_k(\check{\mathbf{x}})}_{c_k} + 2\underbrace{\mathbf{e}_k(\check{\mathbf{x}})^T \Omega_k^{-1} \mathbf{J}_k}_{\mathbf{b}_k^T} \Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\mathbf{J}_k^T \Omega_k^{-1} \mathbf{J}_k}_{H_k} \Delta\mathbf{x} \\
&= c_k + 2\mathbf{b}_k \cdot \Delta\mathbf{x} + \Delta\mathbf{x}^T H_k \Delta\mathbf{x}
\end{aligned}$$

and thus the optimization problem can be rewritten as:

$$\mathrm{F}(\check{\mathbf{x}} + \Delta\mathbf{x}) = \sum_{k=1}^{K} \mathrm{F}_k(\check{\mathbf{x}} + \Delta\mathbf{x}) = \sum_{k=1}^{K} c_k + 2\mathbf{b}_k \cdot \Delta\mathbf{x} + \Delta\mathbf{x}^T H_k \Delta\mathbf{x} = c + 2\mathbf{b}^T \Delta\mathbf{x} + \Delta\mathbf{x}^T H \Delta\mathbf{x} \tag{7.12}$$

where $c = \sum_{k=1}^{K} c_k, \mathbf{b} = \sum_{k=1}^{K} \mathbf{b}_k$ and $H = \sum_{k=1}^{K} H_k$. The best solution $\Delta\mathbf{x}^*$, that minimizes locally the cost function, can be calculated by differentiating the above equation with respect to $\Delta\mathbf{x}$ and setting it to zero, resulting in the following:

$$H\Delta\mathbf{x}^* = -\mathbf{b} \tag{7.13}$$

Once the solution $\Delta\mathbf{x}^*$ is found, the new estimate of the robot state, which is calculated as $\mathbf{x}^* = \check{\mathbf{x}} + \Delta\mathbf{x}^*$, is used as the initial estimate for the next iteration of the optimization problem. This iterative process describes the Gauss-Newton algorithm. Another iterative algorithm widely used in the optimization in SLAM is Levenberg-Marquardt, which is a modification of the Gauss-Newton algorithm, that includes a damping factor through the approximation of the Hessian matrix H with the following:

$$\mathbf{H} + \lambda\mathbf{I} \tag{7.14}$$

where λ is the damping factor and $\mathbf{I}$ is the identity matrix. The solution, similarly to (7.13) results from the following equation:

$$(\mathbf{H} + \lambda\mathbf{I})\,\Delta\mathbf{x}^* = -\mathbf{b} \tag{7.15}$$

The damping factor is altered throughout the iterations, thus enabling the algorithm to behave like the Gauss-Newton algorithm when the solution is close to the minimum, and like the steepest descent algorithm when the solution is far from it.

7.3 Optimization on manifolds

In the analysis of the previous section, it was assumed that the optimization occurs within a Euclidean space, which overlooks a crucial aspect of the SLAM problem: the robot poses are confined to a manifold, due to the rotational components of the robot's state. A manifold differs from Euclidean space, resembling one only locally, much like the earth seems flat for those standing on its surface. For instance, a robot's 3D pose might be represented by a matrix $T(R, \mathbf{t}) \in SE(3)$, where $R \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ is the rotation and translation of the robot respectively. $SO(3)$ forms a three-dimensional manifold embedded inside a 9-dimensional space, since the parameters of a rotation matrix are 9. Operations like addition are not defined in $SO(3)$ as in Euclidean spaces, and as a result attempting to add quantities (such as an update in the iterative processes mentioned earlier) to a rotation matrix, can violate inner constraints this manifold imposes, such as orthogonality. Consequently, the optimization process may yield invalid solutions, that require renormalization. Furthermore, choosing an overparametrized representation of the robot's state, such as rotation matrices and quaternions—which have more parameters than the actual degrees of freedom—may cause the optimization of non-existent degrees of freedom, that would call for further renormalization [HP08].

Therefore it could be argued that minimal representations, such as Euler angles, should be used for representing rotations; however, this is still a non-optimal choice, since these angles suffer from inherent issues such as the well-known gimbal lock problem due to singularities. As a result, to address these problems, it has been proposed that rotations are represented globally in an overparametrised form—using rotation matrices or unit quaternions—to avoid singularities, while updates are minimally calculated within the local Euclidean space of the rotation manifold. These updates can be directly mapped back onto the manifold, thereby ensuring that the resulting rotation matrix remains valid, satisfying all the necessary constraints such as orthogonality and a determinant of 1, as required for $SO(3)$ representations.

Conveniently, this can be achieved using the concepts of Lie groups and Lie algebras, which are employed to describe certain manifolds, such as the manifold of rotations, and their tangent spaces. These mathematical frameworks provide the capability for the mapping of the updates from their minimal representations back onto the rotation manifold.

Definition 7.3.1: Lie group [HH13]

> A Lie group is a smooth manifold $\mathcal{G}$, the elements of which satisfy the group axioms $(\mathcal{G}, \circ)$.

Lie groups, which are smooth manifolds, exhibit a unique property: locally, they appear identical at every point. Each Lie group has a corresponding Lie algebra, which is a vector space that is tangent to the Lie group at the identity element. Specifically:

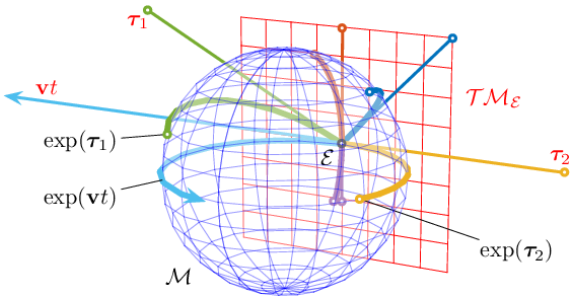


Figure 7.2: Manifold $\mathcal{M}$ and its Lie algebra, which is the tangent space at the identity element. Vectors in Lie algebra (straight green, blue, and yellow lines) are mapped through the exponential map to the manifold (curved green, blue, and yellow lines). The vectors in the Lie algebra, in the case of SLAM, are the updates to the robot's state, while the manifold is the space of the robot's states [SDA18].

Definition 7.3.2: Lie algebra [Bla22]

A Lie algebra is an algebra m with a binary operator $[\cdot,\cdot] : m \times m \to m$ called the Lie bracket, that satisfies the following properties for any elements $a, b, c \in m$:

- **Anti-commutativity**: $[a, b] = -[b, a]$
- **Jacobi identity**: $[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0$

Lie algebras are used to describe the local behaviour of Lie groups, and are the tool that allows the mapping of updates from the minimal representation to the manifold, through the following functions:

- **Exponential map**: $\exp : m \to \mathcal{G}$, which maps an element of the Lie algebra to the Lie group.
- **Logarithmic map**: $\log : \mathcal{G} \to m$, which maps an element of the Lie group to the Lie algebra.

In Figure 7.2 a manifold, its Lie algebra, and the exponential map are illustrated, showing how updates in the Lie algebra are mapped exactly onto the manifold, thus obviating the need for renormalization.

The group of rotation matrices $SO(3)$ is a Lie group with its corresponding Lie algebra $so(3)$ which is the tangent space of $SO(3)$ at the identity element. This tangent space can be determined, by taking the derivative of the orthogonality constraint, yielding the following result [SDA18]:

$$R^T R = I \Rightarrow R^T \dot{R} + \dot{R}^T R = 0 \Rightarrow \dot{R}^T R = -R^T \dot{R} \tag{7.16}$$

This implies that $R^T\dot{R}$ is a skew-symmetric matrix:

$$R^T\dot{R} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} = [\boldsymbol{\omega}]_\times \tag{7.17}$$

where $[\cdot]_\times$ is the skew-symmetric matrix representation of a vector. To get the tangent space at the identity, R is set to the identity matrix, and the above equation can be rewritten as:

$$\dot{R} = [\boldsymbol{\omega}]_\times \tag{7.18}$$

This shows that the Lie algebra of $SO(3)$ is a vector space that can be defined by three generators:

$$E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad E_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.19}$$

as:

$$[\omega]_\times = \omega_x E_1 + \omega_y E_2 + \omega_z E_3 \tag{7.20}$$

Therefore the Lie algebra $so(3)$ can be associated with $\mathbb{R}^3$ via the correspondence $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$. The exponential map of $SO(3)$ is the Rodrigues' formula, which maps elements on the Lie algebra onto the Lie group:

$$\exp([\omega]_\times) = I + \frac{\sin(\theta)}{\theta}[\omega]_\times + \frac{(1-\cos(\theta))}{\theta^2}[\omega]_\times^2 \tag{7.21}$$

where $\theta = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$. To abstract away from the underlying mathematical operations, the "boxplus" operator is defined, which is the equivalent of the addition operator in Lie groups:

$$\mathbf{x} \boxplus \boldsymbol{\delta} = \mathbf{x}\exp(\boldsymbol{\delta}) \tag{7.22}$$

where in the general case $\mathbf{x} \in \mathcal{G}$ and $\boldsymbol{\delta}$ is an element of the corresponding Lie algebra m. Similarly, the "boxminus" operator is defined, as the equivalent of the subtraction operator in Lie groups:

$$\mathbf{x} \boxminus \mathbf{y} = \log(\mathbf{x}^{-1}\mathbf{y}) \tag{7.23}$$

where x, y are both elements of Lie group $\mathcal{G}$. With these definitions (7.11) can be rewritten as:

$$\mathbf{e}_k(\check{\mathbf{x}} \boxplus \Delta\mathbf{x}) \approx \mathbf{e}_k(\check{\mathbf{x}}) + \mathbf{J}_k\Delta\mathbf{x} \tag{7.24}$$

where $\mathbf{J}_k$ becomes:

$$\mathbf{J}_k = \left.\frac{\partial \mathbf{e}_k(\check{\mathbf{x}} \boxplus \Delta\mathbf{x})}{\partial \Delta\mathbf{x}}\right|_{\Delta\mathbf{x}=0} \tag{7.25}$$

With this substitution the optimization problem is solved as described in the previous section, with the difference that now it is executed on the manifold, ensuring that the updated rotation matrices continue to belong on it.

7.3.1 Derivative of Rotated Point

The complexity of the optimization process can be mostly attributed to the participation of rotations. It is very common for an error function to contain a rotated point of the form $R \cdot \mathbf{p}$, where R is a rotation matrix and $\mathbf{p}$ is a point in the environment, that needs to be differentiated with respect to the update parameters of the rotation matrix to calculate the Jacobian (7.25). Therefore in this subsection, we are going to show in greater detail how this derivative can be calculated.

Firstly, we are going to define the axis-angle representation of rotations:

Definition 7.3.3: Axis-angle representation

> The axis-angle representation of rotation R consists of unit vector $\mathbf{n}$ and angle θ, or can just be described by vector $\boldsymbol{\omega} = \theta \cdot \mathbf{n}$. This representation is closely related with the Lie algebra of $SO(3)$, since a small rotation about axis $\mathbf{n}$ by an infinitesimally small angle θ ($\theta\mathbf{n} = \boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$) corresponds to Lie algebra element $[\boldsymbol{\omega}]_\times$ through equation (7.20).

The rotated point after an infinitesimal update to the rotation matrix can be represented as $R_{\text{upd}}(\boldsymbol{\omega}_{\text{upd}}) \cdot R \cdot \mathbf{p}$, where $R_{\text{upd}}(\boldsymbol{\omega}_{\text{upd}})$ is the rotation matrix that results from the exponential map of the update Lie algebra parameters $\boldsymbol{\omega}_{\text{upd}}$. As mentioned in the previous subsection (7.21) in the case of $SO(3)$, the exponential map is equivalent to Rodrigues' formula. Since the update is infinitesimal (7.21) can be simplified as following [Sze22]:

$$R_{\text{upd}}(\boldsymbol{\omega}_{\text{upd}}) \approx I + \sin(\theta_{\text{upd}})[\mathbf{n}_{\text{upd}}]_\times \approx I + [\theta_{\text{upd}}\mathbf{n}_{\text{upd}}]_\times = I + \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{7.26}$$

Therefore the updated rotated point can be rewritten as $R_{\text{upd}}(\boldsymbol{\omega}_{\text{upd}}) \cdot R \cdot \mathbf{p} \approx R \cdot \mathbf{p} + \theta_{\text{upd}}\mathbf{n}_{\text{upd}} \times (R \cdot \mathbf{p}) = R \cdot \mathbf{p} + \boldsymbol{\omega}_{\text{upd}} \times (R \cdot \mathbf{p})$. Consequently, the derivative of the rotated point with respect to the update parameters can now be calculated as:

$$\frac{\partial R_{\text{upd}}(\boldsymbol{\omega}_{\text{upd}}) \cdot R \cdot \mathbf{p}}{\partial \boldsymbol{\omega}_{\text{upd}}} = \frac{\partial R_{\text{upd}}(\boldsymbol{\omega}_{\text{upd}}) \cdot (R \cdot \mathbf{p})}{\partial \boldsymbol{\omega}_{\text{upd}}} = -[R \cdot \mathbf{p}]_\times \tag{7.27}$$

7.4 Factor Graphs

Maximum a posteriori methods can be elegantly connected with the inference of a probability in a probabilistic graphical model. The SLAM problem has a very specific structure since measurements rely only on a specific subset of the robot's states. This structure can be efficiently represented with probabilistic graphical models, which are able to describe complex probability densities, as well as the dependencies between the variables. As it was shown in (7.8) the target posterior probability can be factorized into a product of factors, a fact that can be exploited to represent the SLAM problem as a factor graph [DK+17]. A factor graph is a bipartite graph that consists of two types of

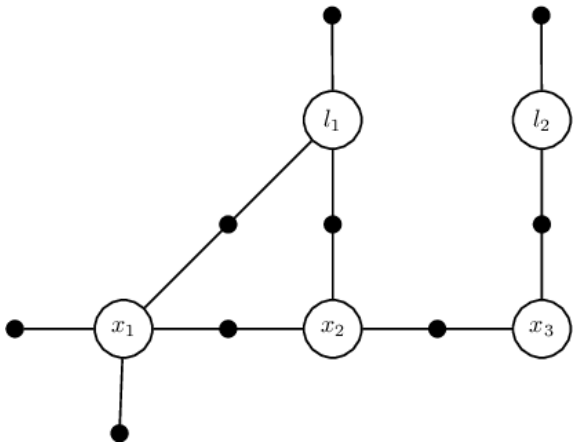


Figure 7.3: Factor graph representation of the SLAM problem, containing robot poses x_1, x_2, x_3 and landmarks l_1, l_2. Variable nodes are represented with white circles and factor nodes with black points. Factor nodes are connected only to the variable nodes they depend upon and represent constraints between variables, created by measurements [DK+17].

nodes, variable nodes and factor nodes, with the variable nodes representing unobservable variables, such as robot and map landmark states, and the factor nodes representing the constraints between the variables, induced by measurements. It follows that factor nodes are connected only to variable nodes upon which they depend. There have been two main frameworks that utilize this graph structure for the SLAM optimization, GTSAM [Del12] and g2o [Küm+11] library. g2o refers to its graph structures as hypergraphs, but they are fundamentally equivalent to factor graphs.

7.5 Advanced Graph-based SLAM Approaches

In this section, we are going to analyze advanced approaches in graph-based SLAM that address a core challenge: noisy data-association. In [SP12], it is assumed that the graphs representing the SLAM problem are not perfect, and thus false loop closures may exist. The authors address this problem by introducing switchable constraints, which are variables attached to the loop closure constraints (edges in the graph) that can "enable" or "disable" them. These variables are continuous and participate in the optimization as inputs to switch functions, such as sigmoids and step functions. Consequently, the topology of the graph representation can be manipulated by the optimization process, resulting in more accurate solutions.

Another significant body of work attempts to address the assumption that graph models are based on unimodal gaussian distributions, which create a network of gaussian potentials [SR14]. As mentioned in previous sections, the unimodality of gaussian distributions fails to address ambiguous measurements or incorrect data associations effectively. Most of the approaches, that will be presented in the rest of this section, assume an approximation of sum-mixtures of gaussians, known as max-mixtures [OA13], which instead of the sum,

rely on the max operator. These models have multiple modalities and are therefore capable of representing more complex distributions. The max operator functions as a selector of the best gaussian in the mixture, returning a single gaussian component on which the known optimization processes are applicable.

In [Lu+21] object poses ambiguities—such as those caused by symmetric objects—are addressed with the introduction of discrete decision variables corresponding to hypotheses on the true pose of objects. Since these variables are discrete, max-mixture models are leveraged for the optimization, enabling efficient tracking of multiple hypotheses. Similarly, [Doh+20] introduces discrete data association variables, thereby allowing the preservation of multiple hypotheses in object associations. These variables are eliminated from the inference procedure again through the max-mixture approach.

In summary, addressing the problem of noisy data association in graph-based SLAM involves various strategies, either with the introduction of continuous variables, such as switchable constraints, or discrete variables, incorporated into the optimization through max-mixture models, which have been shown to be a powerful tool to handle the limitations of unimodal gaussian distributions.

7.6 SLAM systems

The SLAM problem is an area of robotics that has been extensively researched, and as a result, many SLAM systems have been developed. In our work we will focus on a subcategory of the SLAM problem, termed visual SLAM (vSLAM), whose developed algorithms receive their main source of data through a monocular, an RGB-D, or stereo cameras. In this section, we are going to present some of the most well-known vSLAM systems, with a special focus on the ones that handle dynamic environments.

MonoSLAM [Dav+07] was the first system that achieved simultaneous localization and mapping leveraging only a monocular camera. MonoSLAM adopts the EKF approach (Algorithm 6), and represents observed features as small planar 11 x 11 pixel images in 3D space, whose depth is estimated after detecting them from multiple viewpoints. Despite its ground-breaking contributions, MonoSLAM requires some prior knowledge of the environment, through the placement of objects in predetermined locations in front of the camera at its starting position. In contrast to MonoSLAM, traditionally, points were the de facto features used in the vSLAM problem. PTAM [KM07] was such a system that divided tracking and mapping tasks into two threads to ensure real-time execution. ORB-SLAM2 [MT17] with the utilization of ORB features and the use of a sparse pose graph for Bundle Adjustment, achieved real-time performance, while also providing robustness with the capability to close loops and relocalize in cases of lost tracking.

However, points might provide insufficient correspondences in some low-light or low-texture areas and sparse point-based maps lack detailed information. On the contrary, more complex geometry shapes, such as lines, are commonly encountered and encapsulate more descriptive information about the environment. This observation led to the rise of

many systems that utilized lines [Gom+19; Pum+17], planes [Kae15] or both [Zhe+22]. To avoid suboptimal solutions when using these geometric entities in an optimization process, minimal representations are used, such as the orthonormal representation [BS05] for lines in [Zuo+17].

Dynamic SLAM systems can be divided into two categories. Systems in the first category detect dynamic objects in frames and remove them from tracking and optimization procedures. DynaSLAM [Bes+18] leverages semantic masks provided by Mask R-CNN [He+17] and reprojection error checks to discard dynamic objects. In DS-SLAM [Yu+18] the distance from epipolar lines is used in conjunction with semantic segmentation to reject dynamic objects. StaticFusion [Sco+18] performs a joint estimation of camera pose and scene dynamicity, making use of a two-term energy error function. According to the estimated dynamicity a weight is attached to the observations, affecting their participation in the optimization problem. ReFusion [Pal+19] estimates pose directly from the truncated signed distance function (TSDF), and detects dynamicity in the scene by checking residual errors after the first registration, while also by finding occupied voxels that were previously empty.

On the contrary, systems of the second category detect dynamic features and track them without discarding parts of frames, thus exploiting present information more efficiently and bridging the problem of SLAM and Moving Object Tracking. VDO-SLAM [Zha+21] utilizes semantic information to distinguish dynamic objects from the static environment, incorporates both in the SLAM framework, and calculates egomotion and dynamic rigid objects' independent movement without prior knowledge of their geometric models. DynaSLAM II [Bes+21] proposes a bundle adjustment problem that includes both static and dynamic features, in addition to providing and optimizing 3D bounding boxes of moving objects. AirDOS [Qiu+22] addresses nonrigid dynamic objects, by including constraints in the motion of articulated objects. CubeSLAM [YS19] produces cube suggestions, that contain 3D object estimations, from 2D bounding boxes through the utilization of vanishing points. It also contributes a novel Bundle Adjustment framework based on [MT17], which optimizes points, objects and camera poses, while also being capable to handle dynamicity. Objects provide geometric and scalar constraints, which are crucial for the minimization of scale drift.

In this thesis, we propose a novel SLAM system of the second category, which combines the advantages of dynamic SLAM and the robustness of line SLAM systems, by tracking points and lines on both static environment and dynamic rigid objects, resulting in a highly accurate framework.

8

Our Approach

8.1 VDO-SLAM as a Baseline System

In this section, we are going to give a brief overview of VDO-SLAM [Zha+21], the system on which our implementation has been based. VDO-SLAM is a visual SLAM system, which achieves concurrent estimation of the egomotion and accurate tracking of moving rigid objects, by leveraging semantic information present on frames. It has been vaguely based on top of ORB-SLAM, with significant modifications to all of its modules, to achieve the aforementioned functionality. The overview of the system can be seen in Figure 8.1.

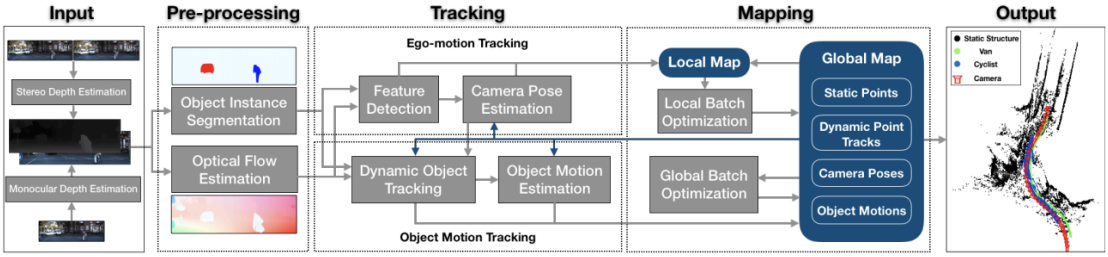


Figure 8.1: Overview of VDO-SLAM system [Zha+21]

The system receives as input a sequence of images, their corresponding depth, their semantic segmentation, and dense optical flow between consecutive frames. In each timestep, ORB features are tracked from the previous frame on the current frame using dense optical flow, while new ORB features are also extracted to supplement the feature pool in cases where the number of correspondences with the previous frame is not enough.

These correspondences are used to solve a PnP problem [LMF09] to retrieve an initial estimate of the camera pose, which is then optimized concurrently with the optical flow, thus enhancing the accuracy of both pose estimation and of the correspondences created by optical flow. Once the camera pose is estimated the execution proceeds with dynamic object tracking. Initially, optical flow is leveraged to match the semantic masks produced by segmentation in consecutive frames. Subsequently, scene flow analysis is utilized to separate dynamic objects from static ones. Specifically, the estimated camera pose is used to align corresponding observations of consecutive frames, hence obtaining an approximation of point motions. Taking into consideration that scene flow should be negligible for static objects, those with a high number of points that do not meet this requirement are deemed as dynamic. The system then proceeds to track the motions of these objects by solving a PnP problem for each of them, resulting in an initial estimate, which is then refined similarly to the camera pose.

In parallel, a map is maintained containing camera poses, dynamic object motions, and, inlier static and dynamic points that were not culled in the stages of camera and dynamic object tracking. The map is optimized both locally, every set number of frames, and globally, to enhance the accuracy of the camera pose, the object motions, and the

mapping.

VDO-SLAM has been evaluated on the KITTI and OMD datasets, where it has shown state-of-the-art performance compared against other dynamic visual SLAM systems. Our implementation, SDPL-SLAM, extended significantly the work of VDO-SLAM, modifying all of its components to add the concurrent utilization of both static and dynamic point and line features, to achieve more accurate and robust tracking of egomotion and dynamic objects. Our system has been compared against VDO-SLAM and DynaSLAM II, showing superior performance in terms of tracking accuracy and robustness.

8.2 SDPL-SLAM

In this section, our implementation is presented. As it has been mentioned earlier, even though the problem of dynamicity in SLAM has been studied extensively, little research has been conducted on the use of more complex features such as lines or planes, which have been shown to enhance the accuracy and the robustness of the algorithms in the static case. Motivated by this, our implementation leverages except points, also static and dynamic lines, with the intuition being, that lines will offer both a higher number and a diversity to the feature set, thus improving the accuracy. The structure of the following subsections is the following:

- Notation used
- Overview of our system and explanation of each component

8.2.1 Notation

Coordinate systems are denoted by C_k and placed as left superscripts for points and lines, excluding the global reference frame 0 which is omitted where possible.

Points: The (in)homogeneous 3D coordinates of the i^{th} point at frame k, with respect to coordinate system C_k, are denoted by ${}^{C_k}\mathbf{M}_k^i \in \mathbb{P}^3$ (and ${}^{C_k}\tilde{\mathbf{M}}_k^i \in \mathbb{R}^3$). Similarly, 2D coordinates with respect to coordinate frame I_k are represented as $\mathbf{m}_k^i \in \mathbb{P}^2$ (and $\tilde{\mathbf{m}}_k^i \in \mathbb{R}^2$). We consider that the last element of homogeneous coordinates is equal to 1.

Lines: A 3D line segment j at frame k can be represented by its endpoints $\{{}^{C_k}\mathbf{A}_k^j, {}^{C_k}\mathbf{B}_k^j\}$, while an infinite 2D line in coordinate frame I_k is denoted by $\mathbf{l}_k^j$. Plücker line coordinates can be constructed as:

$$ {}^{C_k}\mathcal{L}_k^j = \begin{bmatrix} {}^{C_k}\tilde{\mathbf{A}}_k^j \times {}^{C_k}\tilde{\mathbf{D}}_k^j \\ {}^{C_k}\tilde{\mathbf{D}}_k^j \end{bmatrix} = \begin{bmatrix} {}^{C_k}\tilde{\mathbf{N}}_k^j \\ {}^{C_k}\tilde{\mathbf{U}}_k^j \end{bmatrix} \tag{8.1} $$

where ${}^{C_k}\tilde{\mathbf{D}}_k^j$ is the directional unit vector of the line. It can be observed that this is not the general definition of Plücker coordinates, since we also impose the two constraints $||{}^{C_k}\tilde{\mathbf{U}}_k^j|| = 1$ and ${}^{C_k}\tilde{\mathbf{N}}_k^j \cdot {}^{C_k}\tilde{\mathbf{U}}_k^j = 0$. These two constraints reduce the Plücker coordinates degrees of freedom to four, thus enabling a one-to-one transform to the orthonormal representation. The orthonormal representation [BS05; Zuo+17] of the line $(U, W) \in SO(3) \times SO(2)$ can be calculated from the Plücker coordinates as follows:

$$ {}^{C_k}U_k^j(\boldsymbol{\theta}) = \begin{bmatrix} \frac{{}^{C_k}\tilde{\mathbf{N}}_k^j}{||{}^{C_k}\tilde{\mathbf{N}}_k^j||} & \frac{{}^{C_k}\tilde{\mathbf{U}}_k^j}{||{}^{C_k}\tilde{\mathbf{U}}_k^j||} & \frac{{}^{C_k}\tilde{\mathbf{N}}_k^j \times {}^{C_k}\tilde{\mathbf{U}}_k^j}{||{}^{C_k}\tilde{\mathbf{N}}_k^j \times {}^{C_k}\tilde{\mathbf{U}}_k^j||} \end{bmatrix} \tag{8.2} $$

$$ {}^{C_k}W_k^j(\theta) = \begin{bmatrix} ||{}^{C_k}\tilde{\mathbf{N}}_k^j|| & -||{}^{C_k}\tilde{\mathbf{U}}_k^j|| \\ ||{}^{C_k}\tilde{\mathbf{U}}_k^j|| & ||{}^{C_k}\tilde{\mathbf{N}}_k^j|| \end{bmatrix} \tag{8.3} $$

Matrix U is updated with $\boldsymbol{\theta}$ and W with θ, as shown in [BS05].

Optical Flow: We define the vector that corresponds to the movement of a pixel $\tilde{\mathbf{m}}_{k-1}^i$ from I_{k-1} to I_k:

$$ \boldsymbol{\phi}_k^i = \tilde{\mathbf{m}}_k^i - \tilde{\mathbf{m}}_{k-1}^i \tag{8.4} $$

Optical flows that correspond to a start or end point of a line j from I_{k-1} to I_k are $\boldsymbol{\phi}_k^{j,\mathbf{a}}$ and $\boldsymbol{\phi}_k^{j,\mathbf{b}}$, respectively.

Transformations: A transformation matrix from frame k' to k is denoted by ${}^{k'}X_k \in SE(3)$: ${}^{C_{k'}}\mathbf{M}_k^i = {}^{k'}X_k {}^{C_k}\mathbf{M}_k^i$. The transformation matrix ${}_{k-1}^{0}H_k \in SE(3)$ denotes a motion for points on dynamic rigid objects from frame $k-1$ to k with respect to the global reference frame 0, i.e. ${}^0\mathbf{M}_k^i = {}_{k-1}^{0}H_k {}^0\mathbf{M}_{k-1}^i$. A transformation $(R, \mathbf{t})$ can be applied to a line represented in Plücker coordinates with:

$$T_{line} = \begin{bmatrix} R & [\mathbf{t}]_\times R \\ 0_{3\times 3} & R \end{bmatrix} \tag{8.5}$$

8.2.2 Overview of SDPL-SLAM

The overview of our system [MMM24] can be seen in Fig. 8.2. The system receives RGB-D images as input, which are pre-processed to retrieve dense optical flow and semantic segmentation. In the tracking stage, the camera pose is calculated from the last frame using static point and line observations. Once camera pose is obtained, dynamic objects are tracked and their motion between two frames is retrieved. In parallel, a local and a global map are maintained. For every set number of time steps, a local batch optimization is performed on the local map to refine the local trajectory, whereas the global batch optimization is performed on the global map to refine jointly the whole trajectory and map.

8.2.3 Line Correspondences and Camera Pose Estimation

Lines are detected using the Line Segment Detector [Von+08]. Lines that have a depth discontinuity, or whose endpoints belong to different semantic masks are culled.

Optical flow is employed to acquire line correspondences in consecutive frames in the same way point correspondences are found in [Zha+21]. This tackles a big problem often present in line-based SLAM systems that use line descriptors, as lines cannot be detected consistently between frames or are detected with different lengths. In the first case, the correspondent line is not found, while in the second one, descriptors might not match due to different line appearance. Utilizing optical flow, we have achieved a higher number of line matches between frames ensuring long line tracklets.

An initial camera pose is estimated with a Perspective-n-Point algorithm in a RANSAC scheme, using only points that do not belong to objects. To refine this estimate, we propose a novel minimization problem, which optimizes concurrently camera pose and optical flow, improving the initial point and line correspondences. Specifically, the following error term is proposed:

$$\mathbf{e}_{j,l} = \mathbf{e}_j({}^0X_k, \boldsymbol{\phi}_k^{j,\mathbf{a}}, \boldsymbol{\phi}_k^{j,\mathbf{b}}) = \begin{bmatrix} \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}^0X_k^{-1}\mathbf{A}_{k-1}^j) \\ \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}^0X_k^{-1}\mathbf{B}_{k-1}^j) \end{bmatrix} \tag{8.6}$$

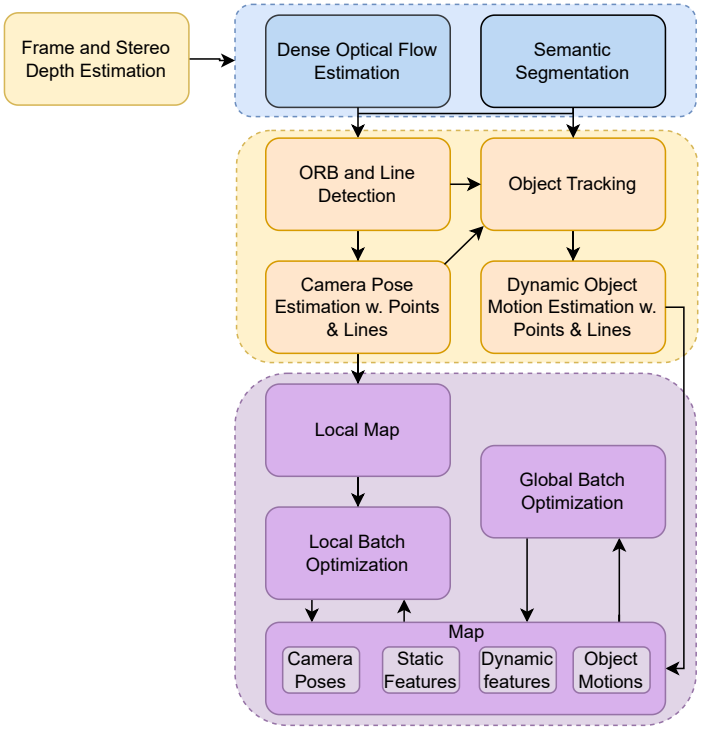


Figure 8.2: **SDPL-SLAM (Static-Dynamic Point-Line SLAM) system overview:** Consists of three main components: pre-processing (Blue), tracking (Yellow), and batch optimization (Purple) [MMM24].

where $\mathbf{l}_k^{j,obs}$ is the observed infinite line given by:

$$\mathbf{l}_k^{j,obs} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \frac{\mathbf{a}_k^{j,obs} \times \mathbf{b}_k^{j,obs}}{||\mathbf{a}_k^{j,obs} \times \mathbf{b}_k^{j,obs}||} \tag{8.7}$$

where $\boldsymbol{\pi}(\cdot)$ is the projective function returning a homogeneous vector, $\boldsymbol{\phi}_k^{j,\mathbf{a}}$ and $\boldsymbol{\phi}_k^{j,\mathbf{b}}$ are the optical flows corresponding to the start and end points of line j from coordinate frame I_{k-1} to I_k and $\tilde{\mathbf{a}}_k^{j,obs} = \tilde{\mathbf{a}}_{k-1}^{j} + \boldsymbol{\phi}_k^{j,\mathbf{a}}$, $\tilde{\mathbf{b}}_k^{j,obs} = \tilde{\mathbf{b}}_{k-1}^{j} + \boldsymbol{\phi}_k^{j,\mathbf{b}}$ the endpoints of the observed line in the current frame. This error term (8.6) consists of the stacked distances of the reprojected endpoints of line j at frame $k-1$ from the line defined by the observed corresponding endpoints at frame k (see Fig. 8.4). If the solution of the minimization problem results in an error term that exceeds a set threshold, the corresponding line is considered an outlier and is removed. This error term resembles that of [Gom+19; Pum+17], however, in our proposal, it is also dependent on the optical flow and new Jacobians had to be calculated.

The Jacobians of this error term were calculated analytically. The derivative with respect to the optical flow of the start point (similar for the endpoint) was calculated as:

$$\frac{\partial \mathbf{e}_{j,l}}{\partial \boldsymbol{\phi}_k^{j,\mathbf{a}}} = \boldsymbol{\pi}({}^0X_k^{-1}\mathbf{A}_{k-1}^{j})^\top \frac{\partial \mathbf{l}_k^{j,obs}}{\partial \boldsymbol{\phi}_k^{j,\mathbf{a}}} \tag{8.8}$$

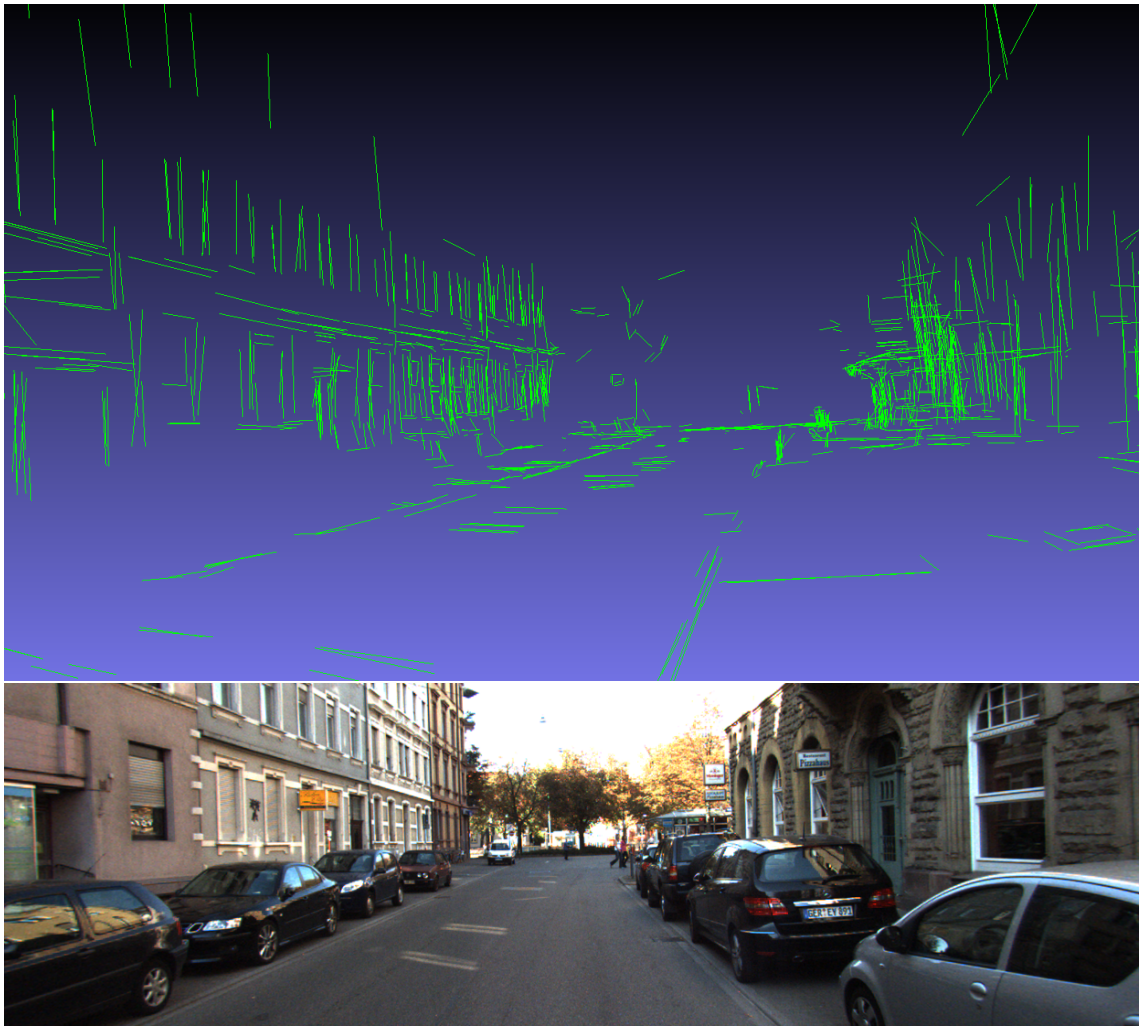


Figure 8.3: Top: Visualization of lines contained in the map maintained by SDPL-SLAM for a city scene. Bottom: A frame from the data sequence that was used to create the map.

and the Jacobian with respect to the pose parameters $\boldsymbol{\Xi}_k$ as:

$$\frac{\partial \mathbf{e}_{j,l}}{\partial \boldsymbol{\Xi}_k} = \begin{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix}^\top \frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{A}^j_{k-1})}{\partial \boldsymbol{\Xi}_k} \\ \begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix}^\top \frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{B}^j_{k-1})}{\partial \boldsymbol{\Xi}_k} \end{bmatrix} \tag{8.9}$$

For the calculation of the Jacobians $\frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{A}^j_{k-1})}{\partial \boldsymbol{\Xi}_k}$ we initially define $\mathbf{g} = [g_x, g_y, g_z]^T = {}^0X_k^{-1} A^j_{k-1}$. The Jacobian then may be given by:

$$\frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{A}^j_{k-1})}{\partial \boldsymbol{\Xi}_k} = \begin{bmatrix} \frac{f_x}{g_z} & 0 & -f_x \frac{g_x}{g_z^2} & -f_x \frac{g_x g_y}{g_z^2} & f_x(1 + \frac{g_x^2}{g_z^2}) & -f_x \frac{g_y}{g_z} \\ 0 & \frac{f_y}{g_z} & -f_y \frac{g_y}{g_z^2} & -f_y(1 + \frac{g_y^2}{g_z^2}) & f_y \frac{g_x g_y}{g_z^2} & f_y \frac{g_x}{g_z} \end{bmatrix} \tag{8.10}$$

The Jacobian $\frac{\partial \boldsymbol{\pi}({}^0X_k, \mathbf{B}^j_{k-1})}{\partial \boldsymbol{\Xi}_k}$ can be calculated in a similar manner. Details about the above

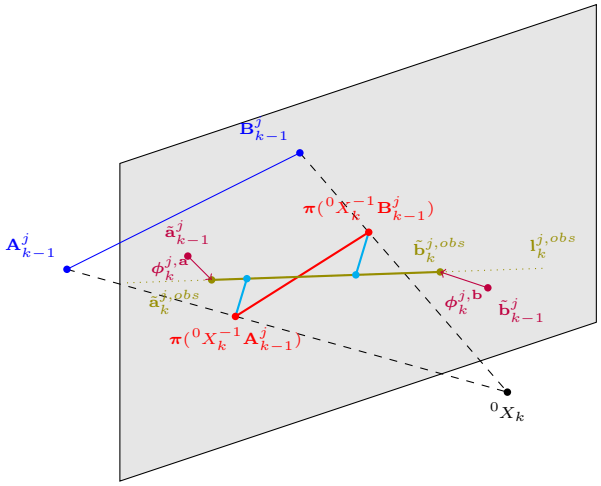


Figure 8.4: **3D illustration of the line reprojection error term:** Line endpoints $\mathbf{A}^j_{k-1}$ and $\mathbf{B}^j_{k-1}$ project onto coordinate frame I_k at the endpoints $\boldsymbol{\pi}(^0X_k^{-1}\mathbf{A}^j_{k-1})$ and $\boldsymbol{\pi}(^0X_k^{-1}\mathbf{B}^j_{k-1})$ that define the reprojected line segment. Optical flows ($\boldsymbol{\phi}^{j,\mathbf{a}}_k$, $\boldsymbol{\phi}^{j,\mathbf{b}}_k$) and the endpoints of the line segment at frame $k-1$ ($\tilde{\mathbf{a}}^j_{k-1}$, $\tilde{\mathbf{b}}^j_{k-1}$) are added together to retrieve the observed endpoints of the corresponding line segment at frame k. Error term (8.6) corresponds to the cyan lines and represents the distances of the reprojected line endpoints (Red) from the corresponding observed infinite line (Olive).

derivation can be found in in [Bla22], in which the reader may gain a deeper insight into the mathematical background.

The minimization problem, based on the reprojection errors of points and lines, is thus the following:

$$\begin{aligned}\{^0X^*_k, \boldsymbol{\Phi}^*_k\} = \text{argmin}_{\{^0X_k, \boldsymbol{\Phi}_k\}} \sum_i^{n_p} \{\boldsymbol{\rho}_h(\mathbf{e}^\top_{i,r}\Sigma^{-1}_{\boldsymbol{\phi}}\mathbf{e}_{i.r}) + \\ \boldsymbol{\rho}_h(\mathbf{e}^\top_{i,p}\Sigma^{-1}_p\mathbf{e}_{i,p})\} + \sum_j^{n_l}\{\boldsymbol{\rho}_h(\mathbf{e}^\top_{j,ra}\Sigma^{-1}_{\boldsymbol{\phi}}\mathbf{e}_{j,ra}) + \\ \boldsymbol{\rho}_h(\mathbf{e}^\top_{j,rb}\Sigma^{-1}_{\boldsymbol{\phi}}\mathbf{e}_{j,rb}) + \boldsymbol{\rho}_h(\mathbf{e}^\top_{j,l}\Sigma^{-1}_l\mathbf{e}_{j,l})\},\end{aligned} \tag{8.11}$$

where "*" denotes the optimal solution, n_p and n_l are the number of static point and line correspondences, $\mathbf{e}_{i,p}$ is the well-known reprojection error term for points [MT17; Zha+21; Qiu+22], $\mathbf{e}_{i,r}$, $\mathbf{e}_{j,ra}$ and $\mathbf{e}_{j,rb}$ are regularization terms for the optical flows that correspond to the points [Zha+21], and the start and end points of lines, respectively, Σ_ϕ is the covariance matrix for the regularization error terms, and Σ_p and Σ_l are the covariance matrices associated with the reprojection error terms of points and lines, respectively. The set $\boldsymbol{\Phi}_k$ contains all optical flow vectors from coordinate frame I_{k-1} to I_k that correspond to the points and line endpoints participating in the minimization problem. This problem is implemented using the g2o library [Küm+11], and is solved with the iterative Levenberg-Marquardt algorithm.

8.2.4 Object Tracking and Motion Estimation

After determining the camera pose, optical flow is employed to correlate semantic masks of the same objects in consecutive frames. This is done by identifying the semantic masks between consecutive frames, with the higher number of point correspondences within them. Subsequently, scene flow analysis is utilized to separate dynamic objects from static ones. Specifically, the estimated camera pose is used to align corresponding observations of consecutive frames, hence obtaining an approximation of point motions as:

$$\mathbf{f}_k^i = \mathbf{M}_{k-1}^i - \mathbf{M}_k^i = \mathbf{M}_{k-1}^i - {}^0X_k \, {}^{C_k}\mathbf{M}_k^i \tag{8.12}$$

where $\mathbf{f}_k^i$ is the scene flow vector. Taking into consideration that scene flow should be negligible for static objects, those with a high number of points that do not meet this requirement are deemed as dynamic.

Once the dynamic objects are identified, their motion is estimated by slightly modifying the minimization problem of the previous subsection with the introduction of a similar error term to (8.6):

$$\mathbf{e}_{j,l} = \mathbf{e}_j({}_{k-1}^0G_k, \boldsymbol{\phi}_k^{j,\mathbf{a}}, \boldsymbol{\phi}_k^{j,\mathbf{b}}) = \begin{bmatrix} \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}_{k-1}^0G_k \mathbf{A}_{k-1}^j) \\ \mathbf{l}_k^{j,obs} \cdot \boldsymbol{\pi}({}_{k-1}^0G_k \mathbf{B}_{k-1}^j) \end{bmatrix}, \tag{8.13}$$

where the variable to be estimated is ${}_{k-1}^0G_k = {}^0X_k^{-1} \, {}_{k-1}^0H_k$ and, thus, the minimization problem maintains the form of (8.11):

$$\begin{aligned} \{{}_{k-1}^0G_k^*, \boldsymbol{\Phi}_k^*\} = \text{argmin}_{\{{}_{k-1}^0G_k, \boldsymbol{\Phi}_k\}} \sum_i^{n_p} \{\boldsymbol{\rho}_h(\mathbf{e}_{i,r}^\top \Sigma_{\boldsymbol{\phi}}^{-1} \mathbf{e}_{i.r}) + \\ \boldsymbol{\rho}_h(\mathbf{e}_{i,p}^\top \Sigma_p^{-1} \mathbf{e}_{i,p})\} + \sum_j^{n_l} \{\boldsymbol{\rho}_h(\mathbf{e}_{j,ra}^\top \Sigma_{\boldsymbol{\phi}}^{-1} \mathbf{e}_{j,ra}) + \\ \boldsymbol{\rho}_h(\mathbf{e}_{j,rb}^\top \Sigma_\phi^{-1} \mathbf{e}_{j,rb}) + \boldsymbol{\rho}_h(\mathbf{e}_{j,l}^\top \Sigma_l^{-1} \mathbf{e}_{j,l})\}, \end{aligned} \tag{8.14}$$

It must be noted that even if a static object is initially incorrectly labeled as dynamic, during this stage, it will be identified to have no relative motion and function as static.

8.2.5 Map, Local and Global Batch Optimization

During the execution of SDPL-SLAM [MMM24] a map is maintained, containing static and dynamic points and lines, camera poses and object motions. The map structure can be represented as a graph, a topic that was covered in section 7.4. A graph optimization formulation is proposed to jointly refine the trajectory of the camera, the motion of dynamic rigid objects, and the map which consists of points and lines on both static and dynamic objects. This graph encapsulates constraints on the variables to be estimated, in the form of error terms, which participate in a nonlinear least square problem, as in [Küm+11]. Specifically, two types of novel line constraints are proposed, (i) 3D line measurement constraints and (ii) constraints on the motion of lines that belong to dynamic

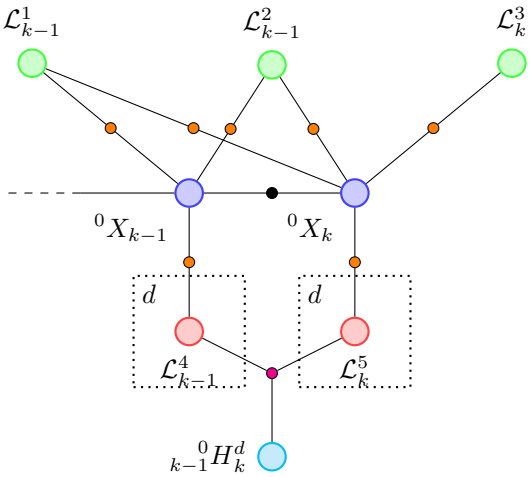


Figure 8.5: **Factor graph representation for line landmarks:** Showcases only static and dynamic **line** features and the constraints imposed by them. Translucent Circles: 3D static lines (Green), poses (Blue), 3D dynamic lines (Red), object motion transform (Cyan). Opaque Circles: 3D line measurement constraints (Orange), constraints on the motion of lines that belong to dynamic objects d (Magenta), pose constraints (Black).

rigid objects. The rest of the constraints, created by point and odometry observations, remain as in [Zha+21]. The novel constraints (i) and (ii) are presented as orange and magenta factors, respectively, in Fig. 8.5, which contains only **line** observations. Line representations have to be minimal in order to avoid numerical instability problems during their optimization and extra computational costs caused by extra degrees of freedom. Orthonormal representation [BS05] is chosen as a minimal representation for 3D lines.

The 3D line measurement error is defined as:

$$\mathbf{e}_{j,k}({}^0X_k, \mathcal{L}^j_k) = \begin{bmatrix} || \, {}^{C_k}\tilde{\mathbf{A}}^{j,obs}_k \times {}^{C_k}\tilde{\mathbf{U}}^j_k - {}^{C_k}\tilde{\mathbf{N}}^j_k || \\ || \, {}^{C_k}\tilde{\mathbf{B}}^{j,obs}_k \times {}^{C_k}\tilde{\mathbf{U}}^j_k - {}^{C_k}\tilde{\mathbf{N}}^j_k || \end{bmatrix}, \tag{8.15}$$

which represents the distances [Bro+10] of the observed 3D endpoints ${}^{C_k}\mathbf{A}^{j,obs}_k$, ${}^{C_k}\mathbf{B}^{j,obs}_k$ from the Plücker line j at frame k.

The following two notes are considered necessary. For static lines, subscript k of Plücker line elements ${}^{C_k}\tilde{\mathbf{U}}^j_k$ and ${}^{C_k}\tilde{\mathbf{N}}^j_k$ is chosen as the first frame line j was observed, whereas for dynamic lines it is actually the current frame k. Secondly, Plücker coordinates are used in the error calculation, however, the update parameters are calculated for the orthonormal representation of the lines, as can be seen in the calculation of the Jacobian with respect to the line parameters $\boldsymbol{\vartheta} = (\boldsymbol{\theta}, \theta)$ (see Appendix).

The constraint of motion of a line j that belongs to a dynamic rigid object d is divided into a distance and angular cost and is defined as:

$$\mathbf{e}_{j,d,k}(\mathcal{L}^j_k, {}^0_{k-1}H^d_k, \mathcal{L}^j_{k-1}) = \begin{bmatrix} \text{dist}(\mathcal{L}^j_k, \mathcal{L}^{j,H}_k) \\ 1 - \frac{\tilde{\mathbf{U}}^j_k \cdot \tilde{\mathbf{U}}^{j,H}_k}{||\tilde{\mathbf{U}}^j_k|| \; ||\tilde{\mathbf{U}}^{j,H}_k||} \end{bmatrix}, \tag{8.16}$$

where ${}_{k-1}^{0}H_k^d$ is the line motion transformation matrix for the object d. The superscript "H" on a line j at frame k belonging to an object d is used to denote that it has undergone a motion transformation ${}_{k-1}^{0}H_k^d$:

$$\mathcal{L}_k^{j,H} = {}_{k-1}^{0}H_k^d \mathcal{L}_{k-1}^{j} = \begin{bmatrix} \tilde{\mathbf{N}}_k^{j,H} \\ \tilde{\mathbf{U}}_k^{j,H} \end{bmatrix} \tag{8.17}$$

Note that to simplify the notation for dynamic lines, we imply that they belong to an object d. The function dist is given by the formula for the distance of two Plücker lines:

$$\begin{aligned} &\text{dist}(\mathcal{L}_k^j, \mathcal{L}_k^{j,H}) = \\ &\begin{cases} \frac{|\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_k^{j,H} + \tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_k^{j,H}|}{||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_k^{j,H}||} & \text{if } \tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_k^{j,H} \neq 0 \\ \frac{||\tilde{\mathbf{U}}_k^j \times (\tilde{\mathbf{N}}_k^j - \tilde{\mathbf{N}}_k^{j,H}/s)||}{||\tilde{\mathbf{U}}_k^j||^2} & \text{if } \tilde{\mathbf{U}}_k^{j,H} = s\tilde{\mathbf{U}}_k^j \text{ for some } s \neq 0 \end{cases} \end{aligned} \tag{8.18}$$

The Jacobians of (8.16) are discussed in the Appendix.

8.2.6 Discussion about the Line Representation in Batch Optimization

In this subsection, we are going to justify our choice of line representation in our proposed batch optimization problem. As was mentioned in the previous subsection, because lines get optimized in our graph proposal, their representation has to be minimal to avoid extra degrees of freedom, thus ensuring a correct optimization. Taking that into consideration, there are two possible choices; either to optimize line segments represented by their endpoints or to transform the line segments into infinite lines using their orthonormal representation.

Even though the endpoint representation of lines in graph optimizations is widely used in the literature [Gom+19; Pum+17], we believe that it is not optimal. To justify that, we analyze the usual case, in which the error to be minimized is the distance of the endpoints of one line from another. Considering that a line is the closest to another nonparallel line at just one point, this error could get smaller by the endpoints just moving closer to that point. Therefore, in this scenario, new local minima could be created in the minimization function, thus leading to nonoptimal solutions. Due to this observation, the orthonormal representation of lines was chosen, which apart from dealing with this problem, also has the advantage, that, because of the smaller number of parameters compared to the endpoint representation, is less computationally demanding. An illustration of this can be seen in Figure 8.6.

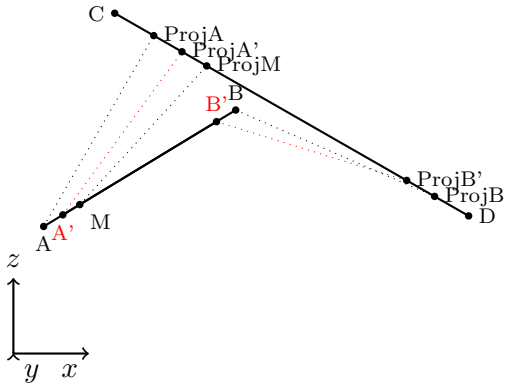


Figure 8.6: **Illustration of the problem of endpoint representation of lines: A** and **B** are the endpoints of a line segment, **C** and **D** are the endpoints of another line segment, and **M** is the point on line segment AB that is closest to segment CD. An error corresponding to the distance of endpoints **A** and **B** from the other line segment, could get smaller by the endpoints just moving closer to the point **M**, as illustrated with $\mathbf{A'}$ and $\mathbf{B'}$. $\mathbf{A'}$ and $\mathbf{B'}$ have a smaller distance from line CD and thus correspond to a smaller error.

9

Experimental Evaluation

To underline the significance of line integration in dynamic SLAM, we conducted a series of experiments in various indoor and outdoor environments and we compared our results against other state-of-the-art methods to demonstrate the effectiveness of our implementation. In this chapter, we present the preprocessing procedures for the datasets, the error metrics used, and our findings on the accuracy of camera egomotion and rigid object poses.

9.1 Datasets

To evaluate the performance of our method in dynamic environments, we tested our algorithm in real-life and challenging datasets. To better assess the accuracy of our method in various conditions we used both indoor and outdoor datasets, namely KITTI Raw Dataset (KITTI) [Gei+13], which consists of extensive urban driving scenarios and Oxford Multimotion Dataset (OMD) [JG19], which includes video sequences of indoor environments with multiple moving objects. In addition, to analyze the robustness of our proposal, sequences with different levels of dynamicity and duration were selected, therefore covering a wide range of possible environments of operation. These datasets provided the necessary information required by our algorithm including:

- Stereo images or RGB-D frames.
- Ground truth camera poses.
- Ground truth poses of dynamic objects.
- Transformations between sensor and camera frames.

This input data was leveraged to extract semantic masks of dynamic objects, dense optical flow images between consecutive frames and depth maps.

9.2 Preprocessing

For the semantic segmentation in KITTI Raw Dataset, an implementation of Mask R-CNN [He+17] is used, which is a deep learning model which detects and segments objects in an image. The model used pre-trained weights on the MS COCO dataset, without further fine-tuning for the specific dataset. The output of the deep network is a binary mask for each image, which is processed to assign ascending values for each object instance. For the OMD dataset, a simple color-based HSV segmentation method implemented by us is used, followed by morphological filtering for refinement.

The dense optical flow is retrieved by the PyTorch version of the PWC-Net model [Sun+18; Nik18] which is a state-of-the-art deep learning model for optical flow estimation. The weights of the model were pre-trained on the FlyingChairs dataset and they are not fine-tuned.

9.2.1 Error Metrics

To compare our results directly with VDO-SLAM, the error metric provided in their paper and implementation is used [Zha+21]. For each frame, the error is defined as $E = \hat{T}^{-1}T$, where $\hat{T}$ is the estimated motion transform for either the camera or an object and T is the corresponding ground truth motion. The translational error E_t is the L_2 norm of the translational part of E, while E_R is the angle of rotation in an axis-angle representation of the rotational component of E.

9.3 KITTI Raw Dataset Results and Discussion

Table 9.1: KITTI Raw Dataset results (E_t[m] and E_R[deg]). *FO = Flow Optimization.

Sequence	Average Length of Static Line Tracklets		DynaSLAM II		VDO-SLAM				Ours (w. FO*)				Ours (wo. FO*)			
			Camera		Camera		Objects		Camera		Objects		Camera		Objects	
	w. FO*	wo. FO*	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R	E_t	E_R
0926-0001	5.1	3.1	-	-	0.051	**0.056**	0.410	0.439	**0.050**	**0.056**	**0.353**	**0.423**	0.051	0.056	0.450	0.426
0926-0002	5.1	3.0	-	-	0.061	0.067	**0.178**	1.528	**0.055**	**0.066**	0.490	**0.674**	0.055	0.066	0.425	1.142
0926-0005	6.2	3.0	-	-	0.059	0.083	**0.378**	1.988	**0.051**	**0.071**	0.462	**1.799**	0.054	0.071	0.264	1.878
0926-0009	5.6	3.1	1.870	0.573	0.110	**0.065**	0.217	0.188	**0.095**	0.066	**0.211**	**0.165**	0.101	0.060	0.211	0.164
0926-0011	8.0	3.1	-	-	0.043	**0.057**	0.623	1.169	**0.034**	**0.057**	**0.265**	**0.325**	0.037	0.057	0.593	0.816
0926-0013	4.6	3.2	0.930	**0.000**	0.076	0.059	**0.139**	0.390	**0.074**	0.058	1.465	**0.355**	0.079	0.058	1.465	0.369
0926-0014	4.8	3.3	1.350	0.573	**0.108**	0.070	0.988	**2.853**	0.110	**0.069**	**0.811**	3.060	0.110	0.069	0.811	3.229
0926-0051	8.3	3.1	1.140	**0.000**	0.065	0.058	1.067	1.029	**0.061**	0.058	**0.644**	**0.415**	0.072	0.059	0.644	0.416
0926-0091	6.1	3.1	-	-	0.069	0.063	-	-	**0.066**	**0.062**	-	-	0.067	0.062	-	-
0926-0093	6.7	3.0	-	-	2.295	0.085	0.869	1.207	**2.284**	**0.084**	**0.669**	**0.391**	2.285	0.083	0.672	0.393
0926-0101	5.2	3.3	15.020	2.292	**0.570**	**0.072**	-	-	0.585	0.073	-	-	0.647	0.078	-	-
0926-0106	6.9	3.0	-	-	0.047	0.062	-	-	**0.039**	**0.058**	-	-	0.033	0.057	-	-
0929-0004	5.4	3.1	1.410	0.573	0.071	0.058	-	-	**0.065**	**0.057**	-	-	0.062	0.057	-	-

The KITTI Raw Dataset consists of many sequences in real outdoor driving environments with given ground truth camera and object poses. To test our system in a variety of environments, a set of 13 sequences with different levels of dynamicity and geometric presence were chosen. The results of our proposed system are presented in Table 9.1. We compare the effectiveness of our system against VDO-SLAM [Zha+21] and the reported results of DynaSLAM II [Bes+21], which are both considered state-of-the-art dynamic SLAM systems.

Regarding the camera's egomotion, our implementation outperforms the other two systems in almost all sequences in E_t, while it is on par or better in E_R. DynaSLAM II seems to achieve a lower rotational error in two sequences, however, it must be noted that its authors provided these results in radians, resulting in a loss of decimal accuracy when they are transformed into degrees.

To conduct a more comprehensive analysis, we have incorporated in our table of results a metric corresponding to the average number of frames static lines are tracked in. Our system demonstrates the most significant improvement in sequences 0926-(0009, 0011, 0093, 0005, 0106), which have, except 0926-0011, a strong presence of nearby buildings providing a lot of high-quality line segments for detection. This translates directly to higher values in the aforementioned metric, underscoring the importance of high-quality

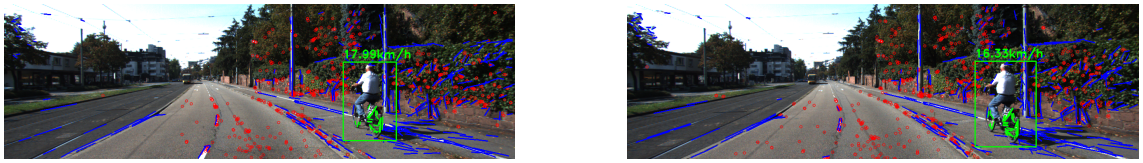


Figure 9.1: In sequence 0926-0002, a big part of the dynamicity of the scene is due to bikes, which provide line segments for detection on their wheels, represented by green lines, leading to a decrease in object tracking accuracy in this specific sequence.

lines that provide consistent tracking. Interestingly, the significantly improved performance in sequence 0926-0011 is justified, despite the absence of nearby buildings, through the exhibition of one of the highest metric values (8.0). Conversely, our system underperforms slightly in sequences 0926-0014 and 0926-0101, which are characterized by open spaces and lack of buildings. Line features are mostly detected on the road and on tree leaves that are located far from the camera, thus causing a degradation in the results. This is reflected in the metric values of these sequences, with their average static line tracklet length (4.8 and 5.2) being significantly below the overall average (6), highlighting the correlation between low-quality line features and reduced accuracy.

Regarding the tracking accuracy of dynamic objects, the inclusion of lines enhances the results in the majority of sequences tested, which may be attributed to most dynamic objects being automobiles that provide a lot of line segments for detection in parts such as windows and license plates. The only sequences in which our implementation does not improve are 0926-(0002, 0005, 0013). A detailed qualitative analysis revealed, that in two of these (0926-0002, 0926-0005), the majority of the dynamic objects detected and tracked are moving bikes with humans (see Figure 9.1 and Figure 9.2), which do not follow the underlying rigidity assumption. Therefore, lines inside the bicycle wheels or lines at the feet of the cyclists contribute to the deterioration of the results in these two cases. However, it must be highlighted that even in these, the object E_R is improved greatly.

Figure 9.2: In many frames of sequence 0926-0005, lines are detected and tracked on a human on a bicycle, which violate the rigidity assumption, leading to a decrease in object tracking accuracy.

Finally, to assess the impact of optical flow optimization on system accuracy, we have conducted an ablation study (see last column of Table 9.1) through modifications in (8.6) and (8.13), by excluding the optical flow dependence from the error terms. This resulted in less consistent line segment matches, with a clear decrease in the average length of static line tracklets and a performance deterioration in both camera E_t and object E_R metrics.

Table 9.2: OMD results (E_t[m] and E_R[deg]).

	VDO-SLAM		Ours	
	E_t	E_R	E_t	E_R
Full Sequence: Camera	0.038	0.578	**0.022**	**0.507**
Full Sequence: Box Mean	0.032	1.286	**0.029**	**1.231**
500 frames: Camera	0.017	0.466	**0.014**	**0.453**
500 frames: Top Right	0.033	1.369	**0.032**	**1.367**
500 frames: Bottom Right	0.030	1.166	**0.029**	**1.164**
500 frames: Top Left	0.036	1.494	**0.031**	**1.452**
500 frames: Bottom Left	**0.027**	**1.601**	**0.027**	1.605
500 frames: Box Mean	0.032	1.407	**0.030**	**1.397**

The fact that sequences 0926-0002 and 0926-0005 perform worse in object pose accuracy when the flow is concurrently optimized, actually supports the findings of the previous paragraph, since more line correspondences in nonrigid objects are retained, therefore magnifying the problem.

9.4 Oxford Multimotion Dataset (OMD) Results and Discussion

The Oxford Multimotion Dataset consists of frame sequences captured in an indoor environment with moving toy cars or levitating cubes. This dataset is characterized by a strong geometric structure, as both the static environment and the moving cubes provide many quality line segments for detection; an ideal scenario to showcase the effect of lines. System performance is evaluated exclusively in the swinging box sequence, and specifically in the unconstrained camera movement case, a challenging realistic scenario. We tested our system both in the initial 500 frames for comparison with [Zha+21], as well as in the entirety of frames to evaluate our system's robustness in a long-running sequence.

As shown in Table 9.2, our system outperforms VDO-SLAM both in egomotion and four moving boxes' pose accuracy, which is a natural outcome considering that the test is run indoors and dynamic objects in the sequence are cubes. Namely, in the full sequence (and in the first 500 frames), a ∼42% (∼18%) and ∼12% (∼2.8%) improvement is achieved in camera E_t and E_R, respectively, compared to VDO-SLAM. Furthermore, the inclusion of lines enhanced the accuracy of the moving boxes' pose estimation in the full sequence and had marginal improvements in the first 500 frames, reducing their average E_t by ∼9.4% (∼6.3%) and E_R by ∼4.3% (∼0.7%).

9.5 Result Summary

We demonstrated that the inclusion of lines resulted in an enhancement to overall performance, on both egomotion and dynamic object tracking, in outdoor driving (Table 9.1) and indoor (Table 9.2) scenarios. Additionally, we introduced the average length of static line tracklets, which quantifies the quality and robustness of line segments. A thorough analysis verified a high correlation between this metric and the improvement in accuracy of our implementation compared to the other state-of-the-art systems. The utilization of optical flow for line matching provides better and more line correspondences, resulting in long-lasting and consistent line tracklets, a benefit that was proved to be further amplified by the concurrent optimization of optical flow in the tracking stage.

10 Conclusions

10.1 Brief Summary and Conclusions

In the current thesis we have presented a novel SLAM system, which exploits line features detected on both static and dynamic objects, in order to estimate camera trajectory and object motions. In the first part of the thesis, we have presented and analyzed core concepts of the SLAM problem, as well as related work, which has already been carried out in the field, in order to provide the necessary background for our proposed system and to identify the limitations of existing methods that motivated the development of our algorithm. In parallel we have introduced complex mathematical tools, such as Lie groups and Lie algebras, which are widely utilised in SLAM research and are essential for the understanding of our proposed system.

In the subsequent chapters, we have presented our approach in depth, analyzing each component and methodology employed. We demonstrated the rationale behind our choices and detailed our contributions in novel optimization formulations, that incorporate line observations to refine camera tracking, dynamic object tracking, and static and dynamic feature positions in the map. We have presented real-life datasets that pose significant challenges to SLAM algorithms, enabling us to test our system across various scenarios and effectively demonstrate its capabilities. Our experimental evaluation revealed that leveraging the line structure of the environment resulted in an overall increase in the accuracy and robustness of the SLAM algorithm compared to other state-of-the-art point-based dynamic systems. Finally, in the Appendix, we delved deeper into the derivations of the Jacobians for our novel error terms, offering insights into the computational process for interested readers.

10.2 Future Research

Even though dynamic Simultaneous Localization and Mapping has garnered considerable attention in SLAM research, numerous challenges remain to be addressed. Several of these became apparent in the course of this thesis, providing valuable directions for future

research. Addressing these issues is an essential step for advancing SLAM algorithms and would significantly enhance applications such as autonomous driving and others, where robustness and accuracy are critical for their operation. Namely, the following areas present opportunities for improvement:

- Initialization of Object Poses: Tracking of features on dynamic objects presents significant challenges, often resulting in an insufficient number of features to reliably initiliaze their poses. The inclusion of lines has expanded the pool of available features and could thus enable more accurate pose initialization. However, currently, only points are included in the Perpective-N-Point (PnP) problem used for pose initialization. Thus, exploring methods to integrate lines in this initialization phase, as suggested in [VFM16] could prove beneficial.

- Humans: Human body's non rigidity poses a significant challenge for the majority of existing dynamic SLAM approaches, which often fail to track the independent movements of human body parts. In future work, we aim to address the presence of humans who contribute significantly to nonrigidity within dynamic environments, by extending our implementation to better handle independent movements of linear skeleton parts of the human body.

- Time Complexity: Incorporating machine learning techniques, particularly deep learning networks, into SLAM systems offers numerous advantages but also contibutes to increased time complexity. A promising direction for future research would be to explore strategies to reduce the time complexity, introduced by these networks. One approach could involve leveraging existing optical flow to predict semantic segmentation masks in conjuction with the reprojection of estimated object poses, thereby potentially eliminating the need to detect these masks in every frame.

11

Appendix

The iterative algorithms discussed in 7.2 necessitate computing the Jacobians of participating error terms to determine the parameters of the optimization problem. Although frameworks like g2o [Küm+11] can compute these Jacobians numerically when not explicitly defined, this approach is both computational intensive and significantly time consuming, thus rendering it unsuitable for real-time applications, such as SLAM. Therefore, in this section, we provide the analytical derivation of the Jacobians of the error terms employed in SDPL-SLAM algorithm. The MATLAB package for symbolic calculations was utilized for complex operations.

11.1 Jacobian of 3D Line Measurement Errors

The error term of 3D Line Measurement is defined in section 8.2.5 as following:

$$\mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j) = \begin{bmatrix} || \, {}^{C_k}\tilde{\mathbf{A}}_k^{j,obs} \times {}^{C_k}\tilde{\mathbf{U}}_k^j - {}^{C_k}\tilde{\mathbf{N}}_k^j || \\ || \, {}^{C_k}\tilde{\mathbf{B}}_k^{j,obs} \times {}^{C_k}\tilde{\mathbf{U}}_k^j - {}^{C_k}\tilde{\mathbf{N}}_k^j || \end{bmatrix}, \tag{11.1}$$

The Jacobian of Eq. (11.1) with respect to orthonormal line parameters $\boldsymbol{\vartheta}_k^j$ of $\mathcal{L}_k^j$ is derived as:

$$\frac{\partial \mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j)}{\partial \boldsymbol{\vartheta}_k} = \frac{\partial \mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j)}{\partial \, {}^{C_k}\mathcal{L}_k^j} \cdot \frac{\partial \, {}^{C_k}\mathcal{L}_k^j}{\partial \mathcal{L}_k^j} \cdot \frac{\partial \mathcal{L}_k^j}{\partial \boldsymbol{\vartheta}_k} \tag{11.2}$$

The first factor is computed straightforwardly using a symbolic math tool, by differentiating the error term (11.1) with respect to the orthonormal line parameters. By defining $[A_x \; A_y \; A_z] = {}^{C_k}\tilde{\mathbf{A}}_k^{j,obs}$, $[B_x \; B_y \; B_z] = {}^{C_k}\tilde{\mathbf{B}}_k^{j,obs}$, $[N_x \; N_y \; N_z] = {}^{C_k}\tilde{\mathbf{N}}_k^j$ and $[U_x \; U_y \; U_z] = {}^{C_k}\tilde{\mathbf{U}}_k^j$, (11.1) can be rewritten as:

$$\mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j) = \begin{bmatrix} \left\| \begin{matrix} -A_zU_y + A_yU_z - N_x \\ A_zU_x - A_xU_z - N_y \\ -A_yU_x + A_xU_y - N_z \end{matrix} \right\| \\ \left\| \begin{matrix} -B_zU_y + B_yU_z - N_x \\ B_zU_x - B_xU_z - N_y \\ -B_yU_x + B_xU_y - N_z \end{matrix} \right\| \end{bmatrix} \tag{11.3}$$

Thus, the calculation of this factor simplifies to differentiating (11.3) with respect to the Plücker line parameters of ${}^{C_k}\mathcal{L}_k^j$:

$$\frac{\partial \mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j)}{\partial \, {}^{C_k}\mathcal{L}_k^j} = \frac{\partial \mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j)}{\partial \left(\mathbf{N}_x, \mathbf{N}_y, \mathbf{N}_z, \mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z\right)} \tag{11.4}$$

The second factor in (11.2) is equal to the transform that converts the Plücker line from the local coordinate system of frame k to the global reference frame:

$$\frac{\partial \, {}^{C_k}\mathcal{L}_k^j}{\partial \mathcal{L}_k^j} = \frac{\partial \, {}^{C_k}T_0\mathcal{L}_k^j}{\partial \mathcal{L}_k^j} = {}^{C_k}T_0 \tag{11.5}$$

The third factor expresses the Jacobian of the Plücker line in the global coordinate system with respect to the orthonormal line parameters as detailed in [Zuo+17]:

$$\frac{\partial \mathcal{L}_k^j}{\partial \boldsymbol{\vartheta}_k} = \begin{bmatrix} -[{}^{C_k}\tilde{\mathbf{N}}_k^j]_\times & -||\,{}^{C_k}\tilde{\mathbf{U}}_k^j||\frac{{}^{C_k}\tilde{\mathbf{N}}_k^j}{||\,{}^{C_k}\tilde{\mathbf{N}}_k^j||} \\ -[{}^{C_k}\tilde{\mathbf{U}}_k^j]_\times & ||\,{}^{C_k}\tilde{\mathbf{N}}_k^j||\frac{{}^{C_k}\tilde{\mathbf{U}}_k^j}{||\,{}^{C_k}\tilde{\mathbf{U}}_k^j||} \end{bmatrix} \tag{11.6}$$

To simplify the computation of the Jacobian of the error term with respect to the pose parameters $\boldsymbol{\Xi}_k$, we divide its derivation in two parts, firstly by considering the translational part $\boldsymbol{\delta}_\rho$ and secondly the rotational part $\boldsymbol{\delta}_\phi$ as in [Zuo+17].

Initially, the Jacobian with respect to the translational part is calculated, with the rotational change $\boldsymbol{\delta}_\phi$ set to zero. The new transform which contains the translation update is denoted as:

$${}^{C_k}T_0^* = \exp(\boldsymbol{\delta}_\rho)\,{}^{C_k}T_0 \approx \begin{bmatrix} I & \boldsymbol{\delta}_\rho \\ \mathbf{0}^T & 1 \end{bmatrix} {}^{C_k}T_0 \tag{11.7}$$

It can be deduced that the rotational part of the transform, R^*, remains the same, while the translational part is updated as $\mathbf{t}^* = \boldsymbol{\delta}_\rho + {}^{C_k}\mathbf{t}_0$. The Plücker line transform is retrieved through Eq (8.5):

$${}^{C_k}T_0^* = \begin{bmatrix} {}^{C_k}R_0 & [\boldsymbol{\delta}_\rho + {}^{C_k}\mathbf{t}_0]_\times\,{}^{C_k}R_0 \\ \mathbf{0} & {}^{C_k}R_0 \end{bmatrix} \tag{11.8}$$

The updated Plücker line in frame k may then be retrieved by transforming the equivalent line expressed in the global reference frame:

$${}^{C_k}\mathcal{L}_k^j = {}^{C_k}T_0^*\mathcal{L}_k^j = \begin{bmatrix} {}^{C_k}R_0\tilde{\mathbf{N}}_k^j + [\boldsymbol{\delta}_\rho + {}^{C_k}\mathbf{t}_0]_\times\,{}^{C_k}R_0\tilde{\mathbf{U}}_k^j \\ {}^{C_k}R_0^T\tilde{\mathbf{U}}_k^j \end{bmatrix} \tag{11.9}$$

In this form (Equation (11.9)), the Jacobian of the Plücker line in the coordinate frame C_k with respect to $\boldsymbol{\delta}_\rho$ can be computed as:

$$\frac{\partial\,{}^{C_k}\mathcal{L}_k^{j,*}}{\partial \boldsymbol{\delta}_\rho} = \begin{bmatrix} \frac{[\partial\delta_\rho + {}^{C_k}\mathbf{t}_0]_\times\,{}^{C_k}R_0\tilde{\mathbf{U}}_k^j}{\partial\delta_\rho} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -[{}^{C_k}R_0\tilde{\mathbf{U}}_k^j]_\times \\ \mathbf{0} \end{bmatrix} \tag{11.10}$$

In the last equation the property:

$$\frac{\partial\,(\mathbf{a}\times\mathbf{b})}{\partial \mathbf{a}} = -[\mathbf{b}]_\times \tag{11.11}$$

is leveraged.

The same procedure is employed to compute the Jacobian $\frac{\partial\,{}^{C_k}\mathcal{L}_k^j}{\partial \boldsymbol{\delta}_\phi}$, incorporating only the rotational update (translational part $\boldsymbol{\delta}_\rho = 0$) to the transform ${}^{C_k}T_0^*$ through the exponential map:

$${}^{C_k}T_0^* = \exp(\boldsymbol{\delta}_\phi)\,{}^{C_k}T_0 \approx \left(I + \begin{bmatrix} [\boldsymbol{\delta}_\phi]_\times & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \right) {}^{C_k}T_0 \tag{11.12}$$

In this case both the rotation part and the translation part of the transform are modified:

$${}^{C_k}R_0^* = (I + [\boldsymbol{\delta}_\phi]_\times)\,{}^{C_k}R_0 \quad \text{and} \quad {}^{C_k}\mathbf{t}_0^* = (I + [\boldsymbol{\delta}_\phi]_\times)\,{}^{C_k}\mathbf{t}_0 \tag{11.13}$$

The line transform is then calculated as:

$$ {}^{C_k}T_0^* = \begin{bmatrix} (I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}R_0 & [(I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}\mathbf{t}_0]_\times\, (I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}R_0 \\ \mathbf{0} & (I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}R_0 \end{bmatrix} \tag{11.14} $$

The top-right block in Equation (11.14) may be simplified by leveraging the property of rotation matrices $(R\mathbf{a}) \times (R\mathbf{b}) = R \cdot (\mathbf{a} \times \mathbf{b}), R \in SO(3)$:

$$ \begin{aligned} [(I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}\mathbf{t}_0]_\times\, (I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}R_0 &= \big((I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}\mathbf{t}_0\big) \times \big((I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}R_0\big) \\ &= (I + [\boldsymbol{\delta}_\phi]_\times)\, ({}^{C_k}\mathbf{t}_0 \times {}^{C_k}R_0) \\ &= (I + [\boldsymbol{\delta}_\phi]_\times)\, [{}^{C_k}\mathbf{t}_0]_\times\, {}^{C_k}R_0 \end{aligned} $$

The updated Plücker line in coordinate frame C_k is therefore as follows:

$$ {}^{C_k}\mathcal{L}_k^{j,*} = {}^{C_k}T_0^*\mathcal{L}_k^j = \begin{bmatrix} (I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j + (I + [\boldsymbol{\delta}_\phi]_\times)\, [{}^{C_k}\mathbf{t}_0]_\times\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j \\ (I + [\boldsymbol{\delta}_\phi]_\times)\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j \end{bmatrix} \tag{11.15} $$

Thus, the derivative of the Plücker line in frame k with respect to the rotational part of the pose parameters, $\boldsymbol{\delta}_\phi$, is:

$$ \begin{aligned} \frac{\partial\, {}^{C_k}\mathcal{L}_k^{j,*}}{\partial \boldsymbol{\delta}_\phi} &= \begin{bmatrix} \frac{\partial \left(I + [\boldsymbol{\delta}_\phi]_\times\right)\, {}^{C_k}R_0\tilde{\mathbf{N}}_k^j}{\partial \boldsymbol{\delta}_\phi} + \frac{\partial \left(I + [\boldsymbol{\delta}_\phi]_\times\right) [{}^{C_k}\mathbf{t}_0]_\times\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j}{\partial \boldsymbol{\delta}_\phi} \\ \frac{\partial \left(I + [\boldsymbol{\delta}_\phi]_\times\right)\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j}{\partial \boldsymbol{\delta}_\phi} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial [\boldsymbol{\delta}_\phi]_\times\, {}^{C_k}R_0\tilde{\mathbf{N}}_k^j}{\partial \boldsymbol{\delta}_\phi} + \frac{\partial [\boldsymbol{\delta}_\phi]_\times [{}^{C_k}\mathbf{t}_0]_\times\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j}{\partial \boldsymbol{\delta}_\phi} \\ \frac{\partial [\boldsymbol{\delta}_\phi]_\times\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j}{\partial \boldsymbol{\delta}_\phi} \end{bmatrix} \\ &= \begin{bmatrix} -[{}^{C_k}R_0\tilde{\mathbf{N}}_k^j]_\times - [[{}^{C_k}\mathbf{t}_0]_\times\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j]_\times \\ -[{}^{C_k}R_0\tilde{\mathbf{U}}_k^j]_\times \end{bmatrix} \end{aligned} $$

As a result, the full Jacobian $\frac{\partial\, {}^{C_k}\mathcal{L}_k^{j,*}}{\partial \boldsymbol{\Xi}_k}$, incorporating both the translational and rotational changes is:

$$ \frac{\partial\, {}^{C_k}\mathcal{L}_k^{j,*}}{\partial \boldsymbol{\Xi}_k} = \begin{bmatrix} -[{}^{C_k}R_0\tilde{\mathbf{U}}_k^j]_\times & -[{}^{C_k}R_0\tilde{\mathbf{N}}_k^j]_\times - [[{}^{C_k}\mathbf{t}_0]_\times\, {}^{C_k}R_0\tilde{\mathbf{U}}_k^j]_\times \\ 0_{3\times 3} & -[{}^{C_k}R_0\tilde{\mathbf{U}}_k^j]_\times \end{bmatrix} \tag{11.16} $$

The Jacobian of the error term with respect to the pose parameters $\boldsymbol{\Xi}_k$ is calculated leveraging the chain rule as:

$$ \frac{\partial \mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j)}{\partial \boldsymbol{\Xi}_k} = \frac{\partial \mathbf{e}_{j,k}({}^0X_k, \mathcal{L}_k^j)}{\partial\, {}^{C_k}\mathcal{L}_k^{j,*}} \cdot \frac{\partial\, {}^{C_k}\mathcal{L}_k^{j,*}}{\partial \boldsymbol{\Xi}_k} \tag{11.17} $$

11.2 Jacobian of Motion of Lines Errors

The error term of Motion of Lines is defined in section 8.2.5 as follows:

$$ \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}{}^0H_k^d, \mathcal{L}_{k-1}^j) = \begin{bmatrix} \text{dist}(\mathcal{L}_k^j, \mathcal{L}_k^{j,H}) \\ 1 - \frac{\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{U}}_k^{j,H}}{||\tilde{\mathbf{U}}_k^j||\ ||\tilde{\mathbf{U}}_k^{j,H}||} \end{bmatrix} \tag{11.18} $$

The Jacobian with respect to the orthonormal line parameters, ϑ_k^j, is derived as:

$$\frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial \boldsymbol{\vartheta}_k^j} = \frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial \mathcal{L}_k^j} \cdot \frac{\partial \mathcal{L}_k^j}{\partial \boldsymbol{\vartheta}_k} \tag{11.19}$$

The first factor of this chain rule is computed as follows:

$$\frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial \mathcal{L}_k^j} = \begin{bmatrix} \frac{\partial \text{dist}(\mathcal{L}_k^j, \mathcal{L}_k^{j,H})}{\partial \mathcal{L}_k^j} \\ \frac{\partial \left(1 - \frac{\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H}}{||\tilde{\mathbf{U}}_k^j|| \; ||\tilde{\mathbf{U}}_{k-1}^{j,H}||}\right)}{\partial \mathcal{L}_k^j} \end{bmatrix} \tag{11.20}$$

This Jacobian consists of two distinct blocks:

- The first block—of size 1×6—is the derivative of the distance between two Plücker lines with respect to the orthonormal line parameters of $\mathcal{L}_k^j$. 'dist' is a piecewise function that represents distance between two lines:

$$\text{dist}(\mathcal{L}_k^j, \mathcal{L}_k^{j,H}) = \begin{cases} \frac{|\tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H} + \tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_{k-1}^{j,H}|}{||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}||} & \text{if } \tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H} \neq 0 \\ \frac{||\tilde{\mathbf{U}}_k^j \times (\tilde{\mathbf{N}}_k^j - \tilde{\mathbf{N}}_{k-1}^{j,H})/s||}{||\tilde{\mathbf{U}}_k^j||^2} & \text{if } \tilde{\mathbf{U}}_{k-1}^{j,H} = s\tilde{\mathbf{U}}_k^j \text{ for some } s \neq 0 \end{cases} \tag{11.21}$$

  Due to the piecewise nature of the function, the derivatives are calculated separately for each case. In the first case, if the content of the norm is positive, the derivative with respect to an element N_i of vector $\tilde{\mathbf{N}}_k^j$ is:

$$\frac{\partial \text{dist}(\mathcal{L}_k^j, \mathcal{L}_{k-1}^{j,H})}{\partial N_i} \tag{11.22}$$

$$= \frac{\frac{\partial (\tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H} + \tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_{k-1}^{j,H})}{\partial N_i} \cdot ||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}|| - (\tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H} + \tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_{k-1}^{j,H}) \cdot \frac{\partial ||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}||}{\partial N_i}}{||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}||^2} \tag{11.23}$$

$$= \frac{\partial (\tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H} + \tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_{k-1}^{j,H})}{\partial N_i} \cdot \frac{1}{||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}||} \tag{11.24}$$

  On the other hand, if the value inside the norm is negative, then the derivative is the opposite of (11.24).

  Similarly, if the content of the norm is positive, the derivative with respect to an element U_i of vector $\tilde{\mathbf{U}}_k^j$ is:

$$\frac{\partial \text{dist}(\mathcal{L}_k^j, \mathcal{L}_{k-1}^{j,H})}{\partial U_i}$$

$$= \frac{\frac{\partial (\tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H} + \tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_{k-1}^{j,H})}{\partial U_i} \cdot ||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}|| - (\tilde{\mathbf{N}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H} + \tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{N}}_{k-1}^{j,H}) \cdot \frac{\partial ||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}||}{\partial U_i}}{||\tilde{\mathbf{U}}_k^j \times \tilde{\mathbf{U}}_{k-1}^{j,H}||^2}$$

  while it is the opposite if its contents are negative.

  The computation of the derivative of the second case, involves the expansion of the cross product and the norm, and differentiation performed with the assistance of symbolic math tools.

- The second block—of size 1×6—corresponds to the derivative of an angle error with respect to the orthonormal line parameters $\mathcal{L}_k^j$. The angle error is calculated as:

$$1 - \frac{\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H}}{||\tilde{\mathbf{U}}_k^j||\ ||\tilde{\mathbf{U}}_{k-1}^{j,H}||} \tag{11.25}$$

Making use of the chain rule, the derivative of the angle error with respect to the orthonormal line parameters of $\mathcal{L}_k^j$ is calculated as:

$$\frac{\partial\left(1 - \left|\frac{\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H}}{||\tilde{\mathbf{U}}_k^j||\ ||\tilde{\mathbf{U}}_{k-1}^{j,H}||}\right|\right)}{\partial \mathcal{L}_k^j} = \frac{\partial\left(1 - \left|\frac{\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H}}{||\tilde{\mathbf{U}}_k^j||\ ||\tilde{\mathbf{U}}_{k-1}^{j,H}||}\right|\right)}{\partial\left(\frac{\tilde{\mathbf{U}}_k^j\ \tilde{\mathbf{U}}_{k-1}^{j,H}}{||\tilde{\mathbf{U}}_k^j||\ ||\tilde{\mathbf{U}}_{k-1}^{j,H}||}\right)} \cdot \frac{\partial\left(\frac{\tilde{\mathbf{U}}_k^j\ \tilde{\mathbf{U}}_{k-1}^{j,H}}{||\tilde{\mathbf{U}}_k^j||\ ||\tilde{\mathbf{U}}_{k-1}^{j,H}||}\right)}{\partial\left[\tilde{\mathbf{N}}_k^j \quad \tilde{\mathbf{U}}_k^j\right]} \tag{11.26}$$

$$= -sgn\left(\frac{\tilde{\mathbf{U}}_k^j \cdot \tilde{\mathbf{U}}_{k-1}^{j,H}}{||\tilde{\mathbf{U}}_k^j||\ ||\tilde{\mathbf{U}}_{k-1}^{j,H}||}\right) \cdot \begin{bmatrix} 0 & 0 & 0 & (1) & (2) & (3) \end{bmatrix} \tag{11.27}$$

where by defining the vector elements of $\tilde{\mathbf{U}}_k^j$ and $\tilde{\mathbf{U}}_{k-1}^{j,H}$ as $[U_{k,x}\ U_{k,y}\ U_{k,z}] = \tilde{\mathbf{U}}_k^j$ and $[U_{k-1,x}\ U_{k-1,y}\ U_{k-1,z}] = \tilde{\mathbf{U}}_{k-1}^{j,H}$, $(1), (2), (3)$ equal:

$$(1) = \frac{U_{k-1,x}(U_{k,y}^2 + U_{k,z}^2) - U_{k,x}U_{k-1,y}U_{k,y} - U_{k,x}U_{k-1,z}U_{k,z}}{||\tilde{\mathbf{U}}_{k-1}^{j,H}||\left(U_{k,x}^2 + U_{k,y}^2 + U_{k,z}^2\right)^{3/2}} \tag{11.28}$$

$$(2) = \frac{U_{k-1,y}(U_{k,z}^2 + U_{k,x}^2) - U_{k,y}U_{k-1,x}U_{k,x} - U_{k,y}U_{k-1,z}U_{k,z}}{||\tilde{\mathbf{U}}_{k-1}^{j,H}||\left(U_{k,x}^2 + U_{k,y}^2 + U_{k,z}^2\right)^{3/2}} \tag{11.29}$$

$$(3) = \frac{U_{k-1,z}(U_{k,x}^2 + U_{k,y}^2) - U_{k,z}U_{k-1,x}U_{k,x} - U_{k,z}U_{k-1,y}U_{k,y}}{||\tilde{\mathbf{U}}_{k-1}^{j,H}||\left(U_{k,x}^2 + U_{k,y}^2 + U_{k,z}^2\right)^{3/2}} \tag{11.30}$$

The intermediate Jacobian $\frac{\partial \mathcal{L}_k^j}{\partial \boldsymbol{\vartheta}_k}$, which participates in (11.19) is detailed in Equation (11.6).

The Jacobian with respect to the orthonormal line parameters $\mathcal{L}_{k-1}^{j,H}$ is computed via the chain rule as follows:

$$\frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial \boldsymbol{\vartheta}_{k-1}^j} = \frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial \mathcal{L}_{k-1}^{j,H}} \cdot \frac{\partial \mathcal{L}_{k-1}^{j,H}}{\partial \mathcal{L}_{k-1}^j} \cdot \frac{\partial \mathcal{L}_{k-1}^j}{\partial \boldsymbol{\vartheta}_{k-1}} \tag{11.31}$$

The derivation of the first factor resembles closely the derivation of $\frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial \mathcal{L}_k^j}$ and is therefore omitted. The third factor is detailed in (11.6) of the previous section. The second factor is equal to the transform ${}_{k-1}^{0}H_k^d$:

$$\frac{\partial \mathcal{L}_{k-1}^{j,H}}{\partial \mathcal{L}_{k-1}^j} = \frac{\partial\ {}_{k-1}^{0}H_k^d \mathcal{L}_{k-1}^j}{\partial \mathcal{L}_{k-1}^j} = {}_{k-1}^{0}H_k^d \tag{11.32}$$

Finally, the Jacobian of the error term with respect to the object motion parameters ${}_{k-1}\boldsymbol{\Xi}_k$ is calculated as follows:

$$\frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial_{k-1}\boldsymbol{\Xi}_k} = \frac{\partial \mathbf{e}_{j,d,k}(\mathcal{L}_k^j, {}_{k-1}^{0}H_k^d, \mathcal{L}_{k-1}^j)}{\partial \mathcal{L}_{k-1}^{j,H}} \cdot \frac{\partial \mathcal{L}_{k-1}^{j,H}}{\partial_{k-1}\boldsymbol{\Xi}_k} \tag{11.33}$$

The computation of these factors has been thoroughly delineated in previous parts of the Appendix.

Βιβλιογραφία

[Bes+18] Berta Bescos, Jose M. Facil, Javier Civera, and Jose Neira. "DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4076–4083.

[Bes+21] Berta Bescos, Carlos Campos, Juan D Tardós, and José Neira. "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5191–5198.

[Bla22] José Luis Blanco-Claraco. *A tutorial on SE(3) transformation parameterizations and on-manifold optimization*. 2022. arXiv: `2103.15980 [cs.RO]`.

[Bro+10] Thomas Brox, Bodo Rosenhahn, Juergen Gall, and Daniel Cremers. "Combined Region and Motion-Based 3D Tracking of Rigid and Articulated Objects". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.3 (2010), pp. 402–415.

[BS05] Adrien Bartoli and Peter Sturm. "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment". In: *Computer Vision and Image Understanding* 100.3 (2005), pp. 416–441.

[Cad+16] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332.

[Dav+07] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1052–1067.

[Del12] Frank Dellaert. "Factor Graphs and GTSAM: A Hands-on Introduction". In: *Georgia Institute of Technology, Tech. Rep* 2 (2012), p. 4.

[DK+17] Frank Dellaert, Michael Kaess, et al. "Factor Graphs for Robot Perception". In: *Foundations and Trends in Robotics* 6.1-2 (2017), pp. 1–139.

[Doh+20] Kevin J Doherty, David P Baxter, Edward Schneeweiss, and John J Leonard. "Probabilistic Data Association via Mixture Models for Robust Semantic SLAM". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 1098–1104.

[Dou+13] Arnaud Doucet, Nando de Freitas, Kevin Murphy και Stuart Russell. *Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks*. 2013. arXiv: 1301.3853 [cs.LG].

[ESC14] Jakob Engel, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 834–849.

[FB81] Martin A Fischler and Robert C Bolles. "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[Gei+13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets Robotics: The KITTI Dataset". In: *The International Journal of Robotics Research* 32 (2013), pp. 1231–1237.

[Gom+19] Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuniga-Noël, Davide Scaramuzza, and Javier Gonzalez-Jimenez. "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments". In: *IEEE Transactions on Robotics* 35.3 (2019), pp. 734–746.

[He+17] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. "Mask R-CNN". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988.

[HH13] Brian C Hall και Brian C Hall. *Lie groups, Lie Algebras, and Representations*. Springer, 2013.

[HP08] C. Hertzberg and Giuseppe Piazzi. "A Framework for Sparse, Non-Linear Least Squares Problems on Manifolds-Ein Rahmen für dünnbesetzte, nichtlineare quadratische Ausgleichsrechnung auf Mannigfaltigkeiten". In: 2008.

[JG19] Kevin Michael Judd and Jonathan D. Gammell. "The Oxford Multimotion Dataset: Multiple SE(3) Motions With Ground Truth". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 800–807.

[Kae15] Michael Kaess. "Simultaneous Localization and Mapping with Infinite Planes". In: *IEEE International Conference on Robotics and Automation (ICRA)* 2015 (June 2015), pp. 4605–4611.

[KM07] Georg Klein and David Murray. "Parallel Tracking and Mapping for Small AR Workspaces". In: *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2007, pp. 225–234.

[Küm+11] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. "G2o: A general framework for graph optimization". In: *IEEE International Conference on Robotics and Automatio (ICRA)*. 2011, pp. 3607–3613.

[LM97] Feng Lu and Evangelos Milios. "Globally Consistent Range Scan Alignment for Environment Mapping". In: *Autonomous Robots* 4 (1997), pp. 333–349.

[LMF09] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. "EPnP: An Accurate O(n) Solution to the PnP Problem". In: *International Journal of Computer Vision* 81 (2009), pp. 155–166.

[Lu+21] Ziqi Lu, Qiangqiang Huang, Kevin Doherty, and John J Leonard. "Consensus-Informed Optimization Over Mixtures for Ambiguity-Aware Object SLAM". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 5432–5439.

[MMM24] Argyris Manetas, Panagiotis Mermigkas, and Petros Maragos. "SDPL-SLAM". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. To be presented. 2024.

[Mon+02] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem". In: *AAAI National Conference on Artificial Intelligence* (2002), pp. 593–598.

[MT17] Raúl Mur-Artal and Juan D. Tardós. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262.

[Nik18] Simon Niklaus. *A Reimplementation of PWC-Net Using PyTorch*. `https://github.com/sniklaus/pytorch-pwc`. 2018.

[OA13] Edwin Olson and Pratik Agarwal. "Inference on Networks of Mixtures for Robust Robot Mapping". In: *The International Journal of Robotics Research* 32.7 (2013), pp. 826–840.

[Pal+19] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. "ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 7855–7862.

[Pum+17] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. "PL-SLAM: Real-time monocular visual SLAM with points and lines". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4503–4508.

[Qiu+22] Yuheng Qiu, Chen Wang, Wenshan Wang, Mina Henein, and Sebastian Scherer. "AirDOS: Dynamic SLAM benefits from Articulated Objects". In: *International Conference on Robotics and Automation (ICRA)*. 2022, pp. 8047–8053.

[Ros+21a] David M Rosen, Kevin J Doherty, Antonio Terán Espinoza, and John J Leonard. "Advances in Inference and Representation for Simultaneous Localization and Mapping". In: *Annual Review of Control, Robotics, and Autonomous Systems* 4.1 (2021), pp. 215–242.

[Ros+21b] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. "Kimera: from SLAM to Spatial Perception with 3D Dynamic Scene Graphs". In: *arXiv preprint arXiv: 2101.06894* (2021).

[Rub+11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *International Conference on Computer Vision (ICCV)*. 2011, pp. 2564–2571.

[Sco+18] Raluca Scona, Mariano Jaimez, Yvan R. Petillot, Maurice Fallon, and Daniel Cremers. "StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 3849–3856.

[SDA18] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. "A micro Lie theory for state estimation in robotics". In: *arXiv preprint arXiv:1812.01537* (2018).

[SMD10] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. "Real-time monocular SLAM: Why filter?" In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2010, pp. 2657–2664.

[SP12] Niko Sünderhauf and Peter Protzel. "Switchable Constraints for Robust Pose Graph SLAM". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 1879–1884.

[SR14] Aleksandr V Segal and Ian D Reid. "Hybrid Inference Optimization for Robust Pose Graph Estimation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2014, pp. 2675–2682.

[Sun+18] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8934–8943.

[Sze22] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[VFM16] Alexander Vakhitov, Jan Funke, and Francesc Moreno-Noguer. "Accurate and Linear Time Pose Estimation from Points and Lines". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 583–599.

[Von+08] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. "LSD: A fast line segment detector with a false detection control". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.4 (2008), pp. 722–732.

[YS19] Shichao Yang and Sebastian Scherer. "CubeSLAM: Monocular 3-D Object SLAM". In: *IEEE Transactions on Robotics* 35.4 (2019), pp. 925–938.

[Yu+18] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. "DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 1168–1174.

[Zha+21] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. *VDO-SLAM: A Visual Dynamic Object-aware SLAM System*. 2021. arXiv: 2005.11052.

[Zhe+22] Weikun Zhen, Huai Yu, Yaoyu Hu, and Sebastian Scherer. "Unified Representation of Geometric Primitives for Graph-SLAM Optimization Using Decomposed Quadrics". In: *International Conference on Robotics and Automation (ICRA)*. 2022, pp. 5636–5642.

[Zuo+17] Xingxing Zuo, Xiaojia Xie, Yong Liu, and Guoquan Huang. "Robust Visual SLAM with Point and Line Features". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1775–1782.