



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ ΚΑΙ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ

Διπλωματική Εργασία

ΜΕΛΕΤΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΣΥΓΧΡΟΝΩΝ ΑΛΓΟΡΙΘΜΩΝ SLAM ΜΕ ΔΕΔΟΜΕΝΑ ΑΠΟ ΑΙΣΘΗΤΗΡΕΣ LIDAR

ΕΜΜΑΝΟΥΗΛ ΤΑΟΥΛΑΙ

ΑΜ: 18391026

Επιβλέποντες Καθηγητές:

Γραμματικόπουλος Λάζαρος, Αναπληρωτής Καθηγητής ΠΑΔΑ
Πέτσα Έλλη, Καθηγήτρια ΠΑΔΑ

ΑΘΗΝΑ, ΜΑΡΤΙΟΣ 2024



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF SURVEYING AND GEOINFORMATICS ENGINEERING

Diploma Thesis

STUDY AND EVALUATION OF MODERN SLAM ALGORITHMS USING LIDAR SENSOR DATA

EMMANUEL TAOLAI

ID: 18391026

Supervisors:

Grammatikopoulos Lazaros, Associate Professor UNIWA
Petsa Elli, Professor UNIWA

ATHENS, MARCH 2024

ΜΕΛΕΤΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΣΥΓΧΡΟΝΩΝ ΑΛΓΟΡΙΘΜΩΝ SLAM ΜΕ ΔΕΔΟΜΕΝΑ ΑΠΟ ΑΙΣΘΗΤΗΡΕΣ LIDAR

Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του Εισηγητή

Η διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

Α/α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1	Γραμματικόπουλος Λάζαρος (Επιβλέπων)	Αναπληρωτής Καθηγητής ΠΑΔΑ	
2	Πέτσα Ελένη	Καθηγήτρια ΠΑΔΑ	
3	Σφήκας Γεώργιος	Επίκουρος Καθηγητής ΠΑΔΑ	

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Εμμανουήλ Ταουλάι του Λαέρντη, με αριθμό μητρώου geo18391026, φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Τοπογραφίας και Γεωπληροφορικής, δηλώνω υπεύθυνα ότι:

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

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

Ο Δηλών

Εμμανουήλ Ταουλάι

Emmanuel.T

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Λάζαρο Γραμματικόπουλο που με στήριξε με την εμπιστοσύνη του και την υπομονή του κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας, αλλά και φυσικά επειδή και αυτός όπως και οι υπόλοιποι καθηγητές του «Εργαστηρίου Φωτογραμμετρίας και Όρασης Υπολογιστών» κ. Έλλη Πέτσα, κ. Γιώργος Σφήκας και κ. Χρηστός Στεντούμης, με έκαναν να αποκτήσω σημαντικό ενδιαφέρον για το αντικείμενο αυτό μέσα από τα μαθήματά τους.

Ύστερα θα ήθελα να ευχαριστήσω όλους τους καθηγητές του τμήματος, καθώς στην πλειοψηφία τους πάντα προσπαθούσαν να μας βοηθήσουν σε κάθε μας δυσκολία και να μας κάνουν να κατανοήσουμε το μάθημα τους καλύτερα.

Θα ήθελα να ευχαριστήσω ακόμη την Μόνικα Κάφκα και τον Ανδρέα Ελ Σαέρ, διδακτορικούς φοιτητές που διαθέσαν και αυτοί χρόνο για τυχόν απορίες μου.

Θα ήθελα να ευχαριστήσω ακόμη τους φοιτητές Παπαδημητρίου Εμμανουήλ και Μαρία Πέτσα για τις επικοινωνητικές συζητήσεις που διεξήγαμε πάνω στα θέματα που αφορούν τις τεχνικές SLAM.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, η οποία πάντα με στηρίζει.

ΠΕΡΙΛΗΨΗ

Η διπλωματική εργασία με τίτλο “ΜΕΛΕΤΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΣΥΓΧΡΟΝΩΝ ΑΛΓΟΡΙΘΜΩΝ SLAM ΜΕ ΔΕΔΟΜΕΝΑ ΑΠΟ ΑΙΣΘΗΤΗΡΕΣ LIDAR” εκπονήθηκε στο πλαίσιο του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Μηχανικών Τοπογραφίας και Γεωπληροφορικής του Πανεπιστημίου Δυτικής Αττικής. Στόχος της εργασίας αποτέλεσε η βιβλιογραφική μελέτη και έρευνα σε θέματα που αφορούν σύγχρονες μεθόδους Ταυτόχρονης Ανίχνευσης και Δημιουργίας Χάρτη (Simultaneous Localization and Mapping - SLAM) οι οποίες αξιοποιούν δεδομένα από αισθητήρες LiDAR, όπως και η διερεύνηση της πρακτικής εφαρμογής και χρηστικότητας των μεθόδων αυτών.

Η τεχνολογία SLAM αναπτύχθηκε αρχικά στο πλαίσιο της επιστήμης της ρομποτικής καθώς υπήρχε ανάγκη αποτύπωσης του περιβάλλοντα χώρου και εντοπισμού θέσης με μεγάλη ακρίβεια αφενός σε πραγματικό χρόνο (real time) και αφετέρου σε απαιτητικά περιβάλλοντα όπως κλειστοί ή υπόγειοι χώροι, αλλά και σε πυκνό αστικό περιβάλλον όπου οι συμβατικές μέθοδοι αποτύπωσης και εντοπισμού θέσης αποτυγχάνουν είτε λόγω χρονικών περιορισμών (απαιτείται χρονοβόρα επεξεργασία), είτε λόγω αδυναμίας εφαρμογής της μεθόδου, όπως για παράδειγμα η χρήση συστημάτων GNSS για υπόγεια πλοήγηση.

Άλλα πλεονεκτήματα της τεχνολογίας SLAM αποτελούν η δυνατότητα για γρήγορη αποτύπωση και εντοπισμό θέσης με υψηλή ακρίβεια και εξοπλισμό χαμηλού κόστους, όπως και η πλήρως αυτοματοποιημένη διαδικασία, εν αντιθέσει με συμβατικές μεθόδους όπου οι διαδικασίες αποτύπωσης πραγματοποιούνται χειροκίνητα. Τα πλεονεκτήματα αυτά σε συνδυασμό με τα διαρκώς αυξανόμενα πεδία εφαρμογής της (ρομποτική, αυτόνομη πλοήγηση οχημάτων και UAV) και την απαίτηση περιορισμού του χρόνου και του κόστους αποτύπωσης, οδήγησαν σε ραγδαία εξέλιξη των συστημάτων που αξιοποιούν τη συγκεκριμένη τεχνολογία και την υλοποιούν σε ποικίλες πλατφόρμες με συνδυασμό διαφορετικών αισθητήρων, και απώτερο σκοπό την βελτιστοποίηση των αποτελεσμάτων. Γίνεται επίσης εύκολα αντιληπτό από την σύγχρονη βιβλιογραφία ότι η πλήρης αυτοματοποίηση της εν λόγω διαδικασίας σε πραγματικό χρόνο και με υψηλή ακρίβεια αποτελεί πρόκληση και ένα ανοικτό πεδίο έρευνας που διευρύνεται συνεχώς σε πρακτικό και θεωρητικό επίπεδο μέσω της εξέλιξης των υπολογιστικών συστημάτων, των αισθητήρων και των αλγορίθμων.

Στο πλαίσιο της παρούσας διπλωματικής εργασίας θα γίνει προσπάθεια αναλυτικής παρουσίασης και θεωρητικής εμβάθυνσης των διαδικασιών που εφαρμόζουν την τεχνολογία SLAM μέσω αισθητήρων LiDAR με έμφαση στους σχετικούς αλγορίθμους που διατίθενται μέσω ανοικτού κώδικα στο διαδίκτυο (υλοποιήσεις σε γλώσσες προγραμματισμού MATLAB, Python, C++) για διαφορετικά λειτουργικά συστήματα (Linux και Windows), καθώς επίσης και σύγκρισης τους με βάση τα πλεονεκτήματα και τα μειονεκτήματα που παρουσιάζουν. Τέλος θα πραγματοποιηθεί εφαρμογή αυτών των μεθόδων σε ανοικτά σύνολα δεδομένων, αλλά σε δεδομένα που λήφθηκαν στο πλαίσιο της διπλωματικής εργασίας μέσω του αισθητήρα LiDAR (LIVOX AVIA) που διαθέτει το εργαστήριο Φωτογραμμετρίας και Όρασης Υπολογιστών.

ABSTRACT

The thesis titled " Study and Evaluation of Modern SLAM Algorithms Using LiDAR Sensor Data" was conducted as part of the Undergraduate Program of the Department of Surveying and Geoinformatics at the University of West Attica. The objective of the thesis was to conduct a literature review and research on topics related to modern Simultaneous Localization and Mapping (SLAM) methods that utilize data from LiDAR sensors, as well as to explore the practical application and usability of these methods.

SLAM technology was initially developed in the field of robotics to address the need for real-time accurate mapping and localization, especially in challenging environments such as closed or underground spaces, as well as dense urban environments where conventional methods fail due to time constraints or due to the inability of using certain techniques for some applications like for example GNSS for underground navigation.

Other advantages of SLAM technology include its ability to rapidly map and localize with high precision and low-cost equipment. Additionally, it performs the entire process automatically, unlike conventional methods that often require manual intervention. These advantages, coupled with the growing demand in various applications (robotics, autonomous vehicle and UAV navigation), led to rapid advancements and experimentation of these systems with different sensor combinations, algorithms, and devices to optimize results. However, achieving fully automated real-time processes with high accuracy remains a challenging and continuously evolving research field. The progress in computational systems, sensors, and algorithms contributes to ongoing improvements in both practical and theoretical aspects.

This thesis provides an analytical presentation and theoretical elaboration of the processes that apply SLAM technology through LiDAR sensors, focusing on the relevant algorithms available through open-source implementations on the internet (implementations in programming languages such as MATLAB, Python, C++) for different operating systems (Linux and Windows), as well as comparing them based on their advantages and disadvantages. Finally, the application of these methods will be carried out on open datasets, as well as on data collected within the framework of the thesis using the LiDAR sensor (LIVOX AVIA) available at the Photogrammetry and Computer Vision Laboratory.

ΑΚΡΩΝΥΜΙΑ

SLAM: Simultaneous Localization and Mapping

EKF: Extended Kalman Filter

ESKF: Error State Kalman Filter

iESKF: Iterated Error State Kalman Filter

GPF: Gaussian Particle Filter

LiDAR: Light Detection and Ranging

IMU: Inertial Measurement Unit

INS: Inertial Navigation System

ToF: Time of Flight

ICP: Iterative Closest Point

G-ICP: Generalised Iterative Closest Point

NDT : Normal Distributions Transform

LOAM: LiDAR Odometry and Mapping

LIO: LiDAR Inertial Odometry

VM: Virtual Machine

RMSE: Root Mean Square Error

BA: Bundle Adjustment

UAV: Unmanned Aerial Vehicle

ROS : Robot Operating System

GNSS : Global Navigation Satellite Systems

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

LiDAR, SLAM, ICP, αλληλοαναφορά, συνταύτιση, χαρτογράφηση 3D, προσδιορισμός θέσης, γράφος θέσης, βελτιστοποίηση, οδομετρία, μετασχηματισμός στερεού σώματος, νέφος σημείων, περιγραφείς

KEYWORDS

LiDAR, SLAM, ICP, Registration, matching, 3D Mapping, localization, pose graph, optimization, odometry, rigid body transformation, point cloud, descriptors

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Αναπαράσταση Γράφου Θέσης (Pose Graph) [33]	17
Σχήμα 2: Συνένωση νεφών σημείων με βάση τον ICP [33]	18
Σχήμα 3: Βασική Δομή Σύγχρονου SLAM [79]	25
Σχήμα 4: Γράφος ενός συστήματος σύγχρονου LiDAR-SLAM [80]	26
Σχήμα 5: Ground truth χάρτης και τροχιά - παράδειγμα 1 MATLAB	32
Σχήμα 7: Κάτοψη χάρτη και τροχιάς - παράδειγμα 1 MATLAB	33
Σχήμα 6: Αποτελέσματα της τροχιάς - παράδειγμα 1 MATLAB	32
Σχήμα 8: Σύγκριση αποτελέσματος με Ground Truth – παράδειγμα 1 MATLAB	33
Σχήμα 9: Σύγκριση αποτελέσματος 2 ^{ης} δοκιμής με Ground Truth (i) – παράδειγμα 1 MATLAB	34
Σχήμα 10: Σύγκριση αποτελέσματος 2 ^{ης} δοκιμής με Ground Truth (ii) – παράδειγμα 1 MATLAB	34
Σχήμα 11: Αποτελέσματα της τροχιάς 2 ^{ης} δοκιμής - παράδειγμα 1 MATLAB	35
Σχήμα 12: Κάτοψη χάρτη και τροχιάς 2 ^{ης} δοκιμής - παράδειγμα 1 MATLAB	35
Σχήμα 13: Αποτέλεσμα χάρτη και τροχιάς (i) - παράδειγμα 2 MATLAB	37
Σχήμα 14: Αποτέλεσμα χάρτη και τροχιάς (ii) - παράδειγμα 2 MATLAB	37
Σχήμα 15: Αποτέλεσμα χάρτη και τροχιάς - παράδειγμα 3 MATLAB	38
Σχήμα 16: Σύγκριση χάρτη με Ground Truth- παράδειγμα 3 MATLAB	39
Σχήμα 17: Αποτέλεσμα εντοπισμού βρόχου 2 ^{ης} δοκιμής- παράδειγμα 3 MATLAB	39
Σχήμα 18: Αποτέλεσμα χάρτη και τροχιάς 2 ^{ης} δοκιμής - παράδειγμα 3 MATLAB	40
Σχήμα 19: Αποτέλεσμα χάρτη και τροχιάς ύστερα από βελτιστοποίηση - παράδειγμα 4 MATLAB	41
Σχήμα 20: Αποτέλεσμα χάρτη και τροχιάς πριν από βελτιστοποίηση - παράδειγμα 4 MATLAB	41
Σχήμα 21: Σχεδιάγραμμα προσομοίωσης - παράδειγμα 5 MATLAB	43
Σχήμα 22: Αποτέλεσμα χάρτη και τροχιάς - παράδειγμα 5 MATLAB	44
Σχήμα 23: Αποτέλεσμα τροχιάς πριν και μετά την βελτιστοποίηση - παράδειγμα 5 MATLAB	44
Σχήμα 24: Αποτέλεσμα τροχιάς και χάρτη πριν και μετά την βελτιστοποίηση - παράδειγμα 6 MATLAB	45
Σχήμα 25: Αποτέλεσμα τροχιάς και χάρτη πριν και μετά την βελτιστοποίηση - παράδειγμα 7 MATLAB	46
Σχήμα 26: Ground Truth χάρτη και τροχιά – παράδειγμα 8 MATLAB	47
Σχήμα 27: Θέσεις εντοπισμού κλεισίματος βρόχων - παράδειγμα 8 MATLAB	47
Σχήμα 28: Σύγκριση τροχιάς πριν και μετά την βελτιστοποίηση - παράδειγμα 8 MATLAB	48
Σχήμα 29: Αποτελέσματα χάρτη και τροχιάς – παράδειγμα 9 MATLAB	49
Σχήμα 30: Παράδειγμα απεικόνισης σαρωμένων σημείων (μωβ χρώμα) και των τελικά εξαγόμενων χαρακτηριστικών σημείων LOAM από την σάρωση αυτή (πράσινο χρώμα).	50
Σχήμα 31: Αποτέλεσμα τροχιάς συνολικά από την μέθοδο LOAM – παράδειγμα 10 MATLAB	51
Σχήμα 32: Αποτέλεσμα τροχιάς οδομετρικά – παράδειγμα 10 MATLAB	51
Σχήμα 33: Γράφημα σύγκρισης τροχιών – παράδειγμα 10 MATLAB	52
Σχήμα 34: Συγκριτικά αποτελέσματα παραδείγματος 10 σε προσομοίωση με λίγα σταθμευμένα οχήματα.	54
Σχήμα 35: Συγκριτικά αποτελέσματα παραδείγματος 8 σε προσομοίωση με πολλά σταθμευμένα οχήματα.	54

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Χρονογραμμή LiDAR-SLAM [23]	15
Εικόνα 2: Επισκόπηση σύγχρονου LiDAR-SLAM [23]	16
Εικόνα 3: Στιγμιότυπο από την σελίδα του benchmarking dataset KITTI [85]	28
Εικόνα 4: Στιγμιότυπο από την σελίδα του benchmarking dataset KITTI [85]	29
Εικόνα 5: Συναρτήσεις της γλώσσας προγραμματισμού MATLAB σχετικές με τα LiDAR-SLAM [88]	30
Εικόνα 6: Σφάλμα μεταβλητής του αλγορίθμου - παράδειγμα 5 MATLAB	42
Εικόνα 7: Στιγμιότυπο από το προσομοιωμένο περιβάλλον – παράδειγμα 5 MATLAB	43
Εικόνα 8: Στιγμιότυπο από το προσομοιωμένο περιβάλλον – παράδειγμα 8 MATLAB	46
Εικόνα 9: Ενσωμάτωση κώδικα για τον υπολογισμό του σφάλματος τροχιάς στο παράδειγμα 8	52
Εικόνα 10: Ο solid state LiDAR σαρωτής με ενσωματωμένο IMU, LIVOX AVIA [95]	59
Εικόνα 11: Κάτοψη της βασικής outdoor περιοχής μελέτης.	61
Εικόνα 12: Κάτοψη της ευρύτερης περιοχής μελέτης των εξωτερικών δοκιμών	62
Εικόνα 13: Κάτοψη αραιού χάρτη της αρχικής περιοχής μελέτης (Άλσους Παγκρατίου)	62
Εικόνα 14: Περιοχή μελέτης εσωτερικού χώρου (indoor)	63
Εικόνα 15: Στιγμιότυπο από το περιβάλλον μιας VM και του περιβάλλοντος του Virtual Box.	64
Εικόνα 16: Στιγμιότυπο από τις απαιτούμενες βιβλιοθήκες του αλγορίθμου LOAM LIVOX [100]	65
Εικόνα 17: Αποτέλεσμα αλγορίθμου LOAM LIVOX με βάση το test dataset CYT_02	67
Εικόνα 19: Στιγμιότυπο από την εκτέλεση του αλγορίθμου LOAM LIVOX (εμφανίζονται ως λευκά τα νέα σημεία σάρωσης)	68
Εικόνα 18: Εικόνα αποτελέσματος αλγορίθμου LOAM LIVOX που παρέχεται από τον δημιουργό για το test dataset CYT_02 [100]	68
Εικόνα 21: Εικόνα αποτελέσματος που παρέχεται από τον δημιουργό για το test dataset CYT_02 [100]	69
Εικόνα 20: Αποτέλεσμα αλγορίθμου LIVOX MAPPING για το dataset CYT_02	69
Εικόνα 22: Αποτέλεσμα αλγορίθμου KISS-ICP για το dataset CYT_02	70
Εικόνα 23: Αποτέλεσμα αλγορίθμου FAST-LIO2 για το dataset outdoor mainbuilding	70
Εικόνα 24: Αποτέλεσμα αλγορίθμου Faster-LIO για το test dataset outdoor mainbuilding	71
Εικόνα 25: Αποτέλεσμα αλγορίθμου LIVOX MAPPING για το dataset outdoor mainbuilding	71
Εικόνα 26: Εικόνα αποτελέσματος από τον δημιουργό για το outdoor mainbuilding [107]	72
Εικόνα 27: Διάταξη εφαρμογής (Σαρωτής LIVOX AVIA με ενσωματωμένο IMU, μπαταριά, φορητός υπολογιστής)	73
Εικόνα 28: Αποτέλεσμα LOAM LIVOX – καταγραφή 1	74
Εικόνα 29: Αποτέλεσμα LOAM LIVOX – καταγραφή 2	74
Εικόνα 30: Αποτέλεσμα LOAM LIVOX – καταγραφή 3	75
Εικόνα 33: Βελτιωμένο αποτέλεσμα LOAM LIVOX – καταγραφή 3	76
Εικόνα 32: Βελτιωμένο αποτέλεσμα LOAM LIVOX – καταγραφή 2	76
Εικόνα 31: Παράμετροι LOAM LIVOX πριν (αριστερά), νέες παράμετροι (δεξιά)	76
Εικόνα 35: Αποτέλεσμα KISS-ICP – καταγραφή 2	77
Εικόνα 34: Αποτέλεσμα KISS-ICP – καταγραφή 1	77
Εικόνα 36: Αποτέλεσμα KISS-ICP – καταγραφή 3	78
Εικόνα 37: Αποτέλεσμα LIVOX MAPPING – καταγραφή 4	79
Εικόνα 38: Αποτέλεσμα LIVOX MAPPING – καταγραφή 5	79
Εικόνα 40: Αποτέλεσμα LIVOX MAPPING – καταγραφή 7	80
Εικόνα 39: Αποτέλεσμα LIVOX MAPPING – καταγραφή 6	80
Εικόνα 41: Αποτέλεσμα FAST-LIO2 – καταγραφή 4	81
Εικόνα 43: Αποτέλεσμα FAST-LIO2 – καταγραφή 6	82
Εικόνα 42: Αποτέλεσμα FAST-LIO2 – καταγραφή 5	82
Εικόνα 44: Αποτέλεσμα FAST-LIO2 – καταγραφή 7	83

Εικόνα 46: Αποτέλεσμα Faster-LIO – καταγραφή 5	84
Εικόνα 45: Αποτέλεσμα Faster-LIO – καταγραφή 4	84
Εικόνα 48: Αποτέλεσμα Faster-LIO – καταγραφή 7	85
Εικόνα 47: Αποτέλεσμα Faster-LIO – καταγραφή 6	85
Εικόνα 49: Αποτέλεσμα LOAM LIVOX – καταγραφή εσωτερικού χώρου	87
Εικόνα 50: Τελικό αποτέλεσμα KISS-ICP – καταγραφή εσωτερικού χώρου	88
Εικόνα 51: Default παράμετροι KISS ICP	89
Εικόνα 52: Βελτιωμένοι παράμετροι KISS ICP για εσωτερικούς χώρους	89
Εικόνα 53: Αρχικό αποτέλεσμα KISS-ICP – καταγραφή εσωτερικού χώρου	89
Εικόνα 54: Αποτέλεσμα LIVOX MAPPING – καταγραφή εσωτερικού χώρου	90
Εικόνα 55: Σφάλματα κατά την εκτέλεση του FAST-LIO2 σε καταγραφή εσωτερικού χώρου	91
Εικόνα 56: FAST-LIO2 – καταγραφή εσωτερικού χώρου (Αποτυχία)	91
Εικόνα 57: Αποτέλεσμα Faster-LIO– καταγραφή εσωτερικού χώρου	92
Εικόνα 58: Παράδειγμα χρόνου εκτέλεσης LOAM LIVOX	96
Εικόνα 59: Στιγμιότυπο αναφοράς χρονών εκτέλεσης του Faster-LIO	97
Εικόνα 60: Στιγμιότυπο αποτελέσματος LOAM LIVOX – CYT_02	98
Εικόνα 62: Σύγκριση των αποτελεσμάτων του LOAM LIVOX και LIVOX MAPPING για το dataset CYT_02	99
Εικόνα 61: Στιγμιότυπο αποτελέσματος LIVOX MAPPING – CYT_02	99
Εικόνα 63: Αποτέλεσμα LOAM LIVOX – καταγραφή 8	100
Εικόνα 64: Αποτέλεσμα LIVOX MAPPING – καταγραφή 8	100
Εικόνα 65: Αποτέλεσμα KISS-ICP - καταγραφή 8	101
Εικόνα 66: Συγκριτικά αποτελέσματα – outdoor mainbuilding (Πάνω μέρος FAST-LIO2, στην μέση Faster-LIO, κάτω LIVOX MAPPING)	102
Εικόνα 67: Συγκριτικά αποτελέσματα - καταγραφή 9 (Πάνω μέρος FAST-LIO2, κάτω Faster-LIO)	103
Εικόνα 68: Συγκριτικά αποτελέσματα - καταγραφή 10 (Πάνω μέρος FAST-LIO2, στην μέση Faster-LIO, κάτω LIVOX MAPPING)	104
Εικόνα 70: Αποτέλεσμα Faster-LIO – καταγραφή 11	105
Εικόνα 69: Αποτέλεσμα FAST-LIO2 – καταγραφή 11	105
Εικόνα 72: Αποτέλεσμα FAST-LIO2 – καταγραφή 12	106
Εικόνα 71: Αποτέλεσμα LIVOX MAPPING – καταγραφή 11	106
Εικόνα 74: Αποτέλεσμα LIVOX MAPPING – καταγραφή 12	107
Εικόνα 73: Αποτέλεσμα Faster-LIO – καταγραφή 12	107
Εικόνα 76: Αποτέλεσμα Faster-LIO – καταγραφή 13	108
Εικόνα 75: Αποτέλεσμα FAST-LIO2 – καταγραφή 13	108
Εικόνα 77: Αποτέλεσμα LIVOX MAPPING – καταγραφή 13	109
Εικόνα 78: Αποτέλεσμα Faster-LIO (CloudCompare) – καταγραφή εσωτερικού χώρου	110
Εικόνα 79: Αποτέλεσμα Faster-LIO (CloudCompare) – καταγραφή εσωτερικού χώρου	111
Εικόνα 80: Αποτέλεσμα LIVOX MAPPING (CloudCompare) – καταγραφή εσωτερικού χώρου	112

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Σύγχρονοι αλγόριθμοι LiDAR SLAM [23]	20
Πίνακας 2: Εξειδικευμένοι αλγόριθμοι LiDAR SLAM για περιστρεφόμενους σαρωτές [23]	24
Πίνακας 3: Τεχνικά χαρακτηριστικά (specs) σαρωτή LIVOX AVIA [95]	60
Πίνακας 4: Τμήμα χρονικής αναφοράς FAST-LIO2	96
Πίνακας 5: Σύγκριση χρόνου επεξεργασίας FAST-LIO2 με Faster-LIO	97
Πίνακας 6: Συγκριτική παρουσίαση των μέσων χρόνων επεξεργασίας των αλγορίθμων	97

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	iii
ABSTRACT.....	iv
1. ΕΙΣΑΓΩΓΗ.....	1
1.1 Αντικείμενο.....	1
1.2 Σκοπός και Στόχοι.....	2
1.2.1 Σκοπός της Διπλωματικής Εργασίας:.....	2
1.2.2 Στόχοι της Διπλωματικής Εργασίας:.....	2
1.3 Μεθοδολογία και Καινοτομία.....	3
1.3.1 Μεθοδολογία.....	3
1.3.2 Καινοτομία.....	3
1.4 Διάρθρωση Εργασίας.....	4
2. ΑΙΣΘΗΤΗΡΕΣ.....	5
2.1 LIDAR.....	6
2.1.1 Τεχνικά Χαρακτηριστικά.....	9
2.1.2 Αλληλοαναφορά νεφών σημείων.....	9
2.1.3 Αναπαράσταση της 3D πληροφορίας.....	11
2.2 Συμπληρωματικά Συστήματα Εντοπισμού Θέσης.....	12
2.2.1 Μονάδα Αδρανειακών Μετρήσεων – IMU/INS.....	12
2.2.2 Συστήματα Δορυφορικού Εντοπισμού GNSS.....	13
2.2.3 Περιστροφικοί κωδικοποιητές (Rotary encoders).....	13
3. ΑΛΓΟΡΙΘΜΟΙ LIDAR SLAM.....	14
3.1 Ιστορική αναδρομή.....	14
3.2 Σύγχρονη μέθοδος LiDAR SLAM.....	16
3.2.1 LiDAR Οδομετρία.....	17
3.2.2 Βελτιστοποίηση Γράφου.....	19
3.2.3 Εντοπισμός Κλειστών Βροχών.....	19
3.3 Τεχνολογίες αιχμής στη μέθοδο SLAM.....	21
3.3.1 Μέθοδοι φιλτραρίσματος (Filter-based).....	21
3.3.2 Μέθοδοι Γράφων (Graph-based).....	22
3.3.3 Μέθοδος SLAM βασισμένη σε περιστροφικά LiDAR.....	23
3.4 Δομή του σύγχρονου LiDAR SLAM.....	25

4. ΠΡΑΚΤΙΚΗ ΕΦΑΡΜΟΓΗ	27
4.1 Παραδείγματα MATLAB	31
4.1.1 Aerial LiDAR SLAM Using FPFH Descriptors	31
4.1.2 Build a map from LiDAR Data using SLAM	36
4.1.3 Implement SLAM with LiDAR Scans	38
4.1.4 Perform SLAM using 3D LiDAR Point Clouds.....	40
4.1.5 Design LiDAR SLAM Algorithm Using 3D Simulation Environment.....	42
4.1.6 Build Map from 2-D LiDAR Scans Using SLAM Example	45
4.1.7 Map Indoor Area Using LiDAR SLAM and Factor Graph Example.....	45
4.1.8 Build an Occupancy Map From LiDAR Data Using Slam Example.....	46
4.1.9 Build A Map From LiDAR Data Example	48
4.1.10 Build a Map with LiDAR Odometry and Mapping (LOAM) Using UE	49
4.2 Πρακτική εφαρμογή σε σύγχρονους αλγορίθμους	56
4.2.1 Εφαρμογή με δεδομένα ελέγχου (Test Data)	67
4.2.2 Πρακτική εφαρμογή σε νέα δεδομένα εξωτερικού χώρου	74
4.2.3 Πρακτική εφαρμογή σε λήψεις δεδομένων εσωτερικού χώρου	86
5. ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ	93
5.1 Χρησιμότητα	94
5.2 Απόδοση	95
5.3 Αποτελέσματα.....	98
6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ	114
6.1 Συμπεράσματα.....	114
6.2 Μελλοντική Έρευνα	116
ΒΙΒΛΙΟΓΡΑΦΙΑ	118

1. ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο

Με τον όρο SLAM (Simultaneous Localization and Mapping) αποδίδεται η διαδικασία μέσω της οποίας πραγματοποιείται ταυτόχρονη χαρτογράφηση του περιβάλλοντα χώρου και προσανατολισμός ενός αισθητήρα στον χώρο αυτό σε πραγματικό χρόνο και με αυτόματο τρόπο.

Η τεχνολογία SLAM αναπτύχθηκε μέσω της ανάγκης γρήγορης και ακριβούς τρισδιάστατης αποτύπωσης του χώρου αλλά και πλοήγησης στον χώρο αυτό για πληθώρα σύγχρονων εφαρμογών σε πραγματικό χρόνο όπως η πλοήγηση των ρομπότ και των αυτόνομων οχημάτων (π.χ. UAV), οι εφαρμογές Εικονικής και Επαυξημένης Πραγματικότητας (VR και AR) και η πλοήγηση και χαρτογράφηση σε απαιτητικά περιβάλλοντα (π.χ. τούνελ), ενθαρρύνοντας έτσι την ανάπτυξη νέων μεθόδων και τεχνικών για τον σκοπό αυτό. Ακόμη, με την εξέλιξη των υπολογιστικών συστημάτων, των αισθητήρων και της επιστημονικής γνώσης, ενισχύονται ακόμη παραπάνω οι δυνατότητες υλοποίησης τεχνικών SLAM.

Οι τεχνικές SLAM δύνανται να ικανοποιήσουν όλες τις παραπάνω ανάγκες, όπως και πιο συνήθεις ανάγκες αποτύπωσης είτε εντοπισμού θέσης όταν αυτές πληρούν τις προδιαγραφές της απαιτούμενης ακρίβειας. Επιπλέον, σε αντίθεση με τις συμβατικές τεχνικές αποτύπωσης, λειτουργούν πλήρως αυτοματοποιημένα χωρίς να χρειάζονται χρονοβόρες διαδικασίες επεξεργασίας των μετρήσεων. Συνήθως τέτοιες τεχνικές χρησιμοποιούν αισθητήρες χαμηλού κόστους, που σε συνδυασμό με τον μικρό χρόνο αποτύπωσης και επεξεργασίας των μετρήσεων, παράγουν αποτελέσματα; μειωμένου κόστους με υψηλή όμως ακρίβεια. Τέλος, ένα ακόμη χαρακτηριστικό των τεχνολογιών SLAM είναι πως μπορούν να εφαρμοστούν ανάλογα με τις ανάγκες και το διαθέσιμο εξοπλισμό, μέσω μετρήσεων από πολλά διαφορετικά είδη αισθητήρων, όπως λήψεις από μια ή δυο κάμερες (Visual SLAM), μετρήσεις RADAR (RADAR SLAM), μετρήσεις από αισθητήρες βάθους RGB-D (RGB-D SLAM) και μετρήσεις LiDAR (LiDAR-SLAM), ενώ μπορούν να ενσωματώσουν και άλλες πληροφορίες από συστήματα μέτρησης όπως είναι τα αδρανειακά συστήματα IMU, οι δορυφορικοί δέκτες GNSS και οι περιστροφικοί κωδικοποιητές (rotary encoders) όταν αυτές είναι διαθέσιμες.

Παρόλο που η κεντρική ιδέα είναι κοινή σε όλες τις επιμέρους μεθόδους SLAM, η υλοποίηση μπορεί να διαφέρει σημαντικά ανάλογα με τον αισθητήρα (ή των συνδυασμό αισθητήρων) που πραγματοποιεί την καταγραφή. Επίσης, ο κάθε τύπος αισθητήρα έχει τα δικά του μειονεκτήματα και πλεονεκτήματα. Κρίθηκε λοιπόν αναγκαίο η παρούσα εργασία να επικεντρωθεί σε αλγορίθμους LiDAR-SLAM,

δηλαδή τεχνικές ταυτόχρονης πλοήγησης και χαρτογράφησης που χρησιμοποιούν δεδομένα αισθητήρων LiDAR.

1.2 Σκοπός και Στόχοι

1.2.1 Σκοπός της Διπλωματικής Εργασίας:

Ο κεντρικός σκοπός της διπλωματικής αυτής εργασίας είναι η εκτενής μελέτη και αξιολόγηση των σύγχρονων αλγορίθμων SLAM, οι οποίοι χρησιμοποιούν δεδομένα από αισθητήρες LiDAR. Αντικείμενο της έρευνας αποτελεί η εξερεύνηση της σχετικής βιβλιογραφίας και η ανάλυση των διαφορετικών μεθόδων LiDAR-SLAM, συγκρίνοντας τα αποτελέσματά τους μέσω εφαρμογής σε πειραματικά δεδομένα τα οποία ελήφθησαν στο πλαίσιο της εργασίας, με σκοπό την περαιτέρω κατανόηση της χρησιμότητάς τους.

1.2.2 Στόχοι της Διπλωματικής Εργασίας:

Οι στόχοι της διπλωματικής εργασίας είναι οι ακόλουθοι:

1. Βιβλιογραφική Μελέτη και Έρευνα: Εκτέλεση λεπτομερούς βιβλιογραφικής μελέτης σχετικά με τους σύγχρονους αλγορίθμους LiDAR-SLAM.
2. Ανάλυση και Αξιολόγηση Μεθόδων SLAM: Εξέταση των διαφόρων μεθόδων SLAM που χρησιμοποιούν δεδομένα LiDAR, με εστίαση στα πλεονεκτήματα και τις προκλήσεις που προκύπτουν στην χρήση τους.
3. Πρακτική Εφαρμογή: Ανάπτυξη και υλοποίηση των μεθόδων SLAM σε πραγματικά δεδομένα, λαμβάνοντας υπόψη τη συμβατότητα τους με τον διαθέσιμο αισθητήρα LiDAR (LIVOX AVIA).
4. Σύγκριση Μεθόδων: Σύγκριση των υλοποιημένων μεθόδων SLAM από διάφορα λειτουργικά συστήματα και γλώσσες προγραμματισμού, αναδεικνύοντας τα πλεονεκτήματα και τα μειονεκτήματά τους.
5. Αξιολόγηση Εφαρμογής: Αξιολόγηση της πρακτικής εφαρμογής και της χρησιμότητας των επιλεγμένων μεθόδων SLAM μέσω δεδομένων από τον διαθέσιμο αισθητήρα LiDAR, σε εξωτερικούς και εσωτερικούς χώρους.

Στο σύνολό της, η διπλωματική εργασία πραγματεύεται προηγμένες τεχνικές SLAM που εκμεταλλεύονται δεδομένα LiDAR και προσφέρουν αξιόπιστες λύσεις για την ταυτόχρονη ανίχνευση και χαρτογράφηση σε πραγματικό χρόνο. Παρουσιάζονται παράλληλα τα αποτελέσματα που παράγουν για test δεδομένα τα οποία παρέχονται ελεύθερα, αλλά και για δεδομένα που αποκτήθηκαν στο πλαίσιο της παρούσας διπλωματικής διατριβής από τον σαρωτή LiDAR (LIVOX AVIA).

1.3 Μεθοδολογία και Καινοτομία

1.3.1 Μεθοδολογία

Για την προσέγγιση του θέματος, αρχικά πραγματοποιήθηκε μια ενδελεχής μελέτη σχετικά με τους αισθητήρες LiDAR και τον τρόπο λειτουργίας τους. Ακολούθησε βιβλιογραφική αναζήτηση για τις πιο πρόσφατες μεθόδους LiDAR-SLAM (μετά το 2014), καθώς και για διαθέσιμο ανοικτό κώδικα με έμφαση σε αλγορίθμους που έχουν υλοποιηθεί σε γλώσσες προγραμματισμού MATLAB και Python όπως και σε λειτουργικό σύστημα Windows. Στη συνέχεια, και αφού δεν εντοπίστηκε επαρκής αριθμός αλγορίθμων LiDAR-SLAM υλοποιημένων σε MATLAB και Python, η αναζήτηση στράφηκε σε υλοποιήσεις που αφορούσαν τον διαθέσιμο αισθητήρα LiDAR (LIVOX AVIA) του εργαστηρίου, ανεξαρτήτως λειτουργικού συστήματος και γλώσσας προγραμματισμού.

Κάθε διαθέσιμος αλγόριθμος εφαρμόστηκε αρχικά στα σύνολα δεδομένων ελέγχου που παρέχουν οι δημιουργοί του, ενώ πραγματοποιήθηκε λήψη και ανάλυση πειραματικών δεδομένων με τον αισθητήρα LiDAR του εργαστηρίου. Αυτά τα δεδομένα χρησιμοποιήθηκαν για την περαιτέρω σύγκριση και αξιολόγηση των διαφόρων σύγχρονων αλγορίθμων SLAM που είχαν επιλεγεί. Η αξιολόγηση των αλγορίθμων πραγματοποιήθηκε με βάση την χρηστικότητα τους, την απόδοσή τους και φυσικά την ποιότητα/ακρίβεια του τελικού αποτελέσματος. Τέλος, τα αποτελέσματα συγκρίθηκαν και αναλύθηκαν προκειμένου να αναδειχθούν τα πλεονεκτήματα και οι περιορισμοί των διαφόρων μεθόδων.

1.3.2 Καινοτομία

Η παρούσα εργασία συμβάλλει σε νέα γνώση καθώς προσφέρει μια ολοκληρωμένη και εμπειριστατωμένη προσέγγιση στην αξιολόγηση της χρηστικότητας και των αποτελεσμάτων διάφορων σύγχρονων αλγορίθμων LiDAR-SLAM. Η ταχεία εξέλιξη αυτών των συστημάτων έχει ως αποτέλεσμα σημαντικές αλλαγές κάθε χρόνο. Μέσω της παρούσας διπλωματικής, παρέχεται ενημέρωση στην κοινότητα σχετικά με τις σύγχρονες μεθόδους, τα προβλήματα που προκύπτουν κατά την εφαρμογή τους και τα αποτελέσματά τους. Ακόμη, η πρακτική εφαρμογή των μεθόδων SLAM σε πραγματικά δεδομένα από τον αισθητήρα LIVOX AVIA προσφέρει πολύτιμη εμπειρία και αποτελέσματα. Η αξιολόγηση της απόδοσης σε πραγματικές συνθήκες βοηθά στην επαλήθευση της αποτελεσματικότητας και της αποδοτικότητας των υπό μελέτη μεθόδων ακόμα και για τους μη εξειδικευμένους χρήστες. Τέλος, προτείνονται βελτιώσεις και μελλοντικές επεκτάσεις των μεθόδων που αναλύονται.

1.4 Διάρθρωση Εργασίας

Παρακάτω ακολουθεί συνοπτική περιγραφή των κεφαλαίων που θα ακολουθήσουν:

Στο Κεφάλαιο 2 πραγματοποιείται μια σύντομη περιγραφή των αισθητήρων οι οποίοι χρησιμοποιούνται στο LiDAR-SLAM με μεγαλύτερη έμφαση στις ιδιότητες και τα χαρακτηριστικά των σαρωτών LiDAR, ενώ γίνεται και μια μικρή αναφορά στις βασικές διαδικασίες που αφορούν τα νέφη σημείων που παράγουν οι σαρωτές.

Στο Κεφάλαιο 3 πραγματοποιείται αρχικά μια ιστορική αναδρομή των αλγορίθμων LiDAR-SLAM, ενώ ακολουθεί βιβλιογραφική ανασκόπηση των σύγχρονων αλγορίθμων LiDAR-SLAM. Τέλος, γίνεται αναφορά στην βασική δομή και τρόπο λειτουργίας των αλγορίθμων αυτών.

Στο Κεφάλαιο 4 γίνεται αναλυτική περιγραφή της πρακτικής εφαρμογής που πραγματοποιήθηκε ξεκινώντας από την μεθοδολογία ενώ ακολουθεί περιγραφή του αισθητήρα και των δεδομένων που χρησιμοποιήθηκαν. Στην συνέχεια, παρατίθενται πιο εξειδικευμένες πληροφορίες για τους αλγορίθμους οι οποίοι αξιοποιήθηκαν κατά την πρακτική εφαρμογή. Τέλος, παρουσιάζονται τα τελικά αποτελέσματα και ένας αρχικός σχολιασμός τους.

Στο Κεφάλαιο 5 πραγματοποιείται η βασική σύγκριση και ανάλυση των αποτελεσμάτων της πρακτικής εφαρμογής που υλοποιήθηκε στο πλαίσιο της παρούσας διπλωματικής.

Στο Κεφάλαιο 6 παρουσιάζονται τα συμπεράσματα που προέκυψαν από την πρακτική εφαρμογή και την συνολική έρευνα η οποία πραγματοποιήθηκε ενώ στο τέλος γίνονται προτάσεις για μελλοντική έρευνα.

2. ΑΙΣΘΗΤΗΡΕΣ

Σε αυτό το κεφάλαιο θα πραγματοποιηθεί μια σύντομη αναφορά στους αισθητήρες που χρησιμοποιούνται από τις σύγχρονες τεχνικές LiDAR-SLAM. Μεταξύ των συστημάτων αυτών κυρίαρχη θέση κατέχουν οι αισθητήρες LiDAR, η ύπαρξη των οποίων επαρκεί για την εφαρμογή της μεθόδου. Ωστόσο σήμερα, έχουν επικρατήσει μέθοδοι οι οποίες συνδυάζουν επιπρόσθετες μετρήσεις κυρίως από αδρανειακά συστήματα IMU/INS, από περιστροφικούς κωδικοποιητές (rotary encoders) και από συστήματα δορυφορικών δεκτών GNSS, με στόχο την βελτίωση των τελικών αποτελεσμάτων (πλοήγηση και χαρτογράφηση).

Γενικά, οι αισθητήρες αποτελούν όργανα τα οποία αντιλαμβάνονται και μετατρέπουν φυσικά φαινόμενα ή μεγέθη σε ηλεκτρικά ή ψηφιακά σήματα. Μπορούν να χρησιμοποιηθούν για παρακολούθηση φαινομένων, μέτρηση μεγεθών ή άλλες εφαρμογές ενώ είναι χρήσιμοι σε διαφορετικούς τομείς της τεχνολογίας και της επιστήμης, καθώς παρέχουν πληροφορίες και μετρήσεις που αφορούν το φυσικό περιβάλλον. Κατηγοριοποιούνται σε παθητικούς και ενεργητικούς ανάλογα με το εάν παράγουν οι ίδιοι το σήμα που καταγράφουν (ενεργητικοί) ή εάν καταγράφουν μόνο εξωτερικά σήματα (παθητικοί).

Μια ακόμη κατηγοριοποίηση που πραγματοποιείται διακρίνει τους αισθητήρες σε ιδιοδεκτικούς (proprioceptive) και αφορά τους αισθητήρες που καταγράφουν μεταβολές της δική τους κατάστασης, όπως για παράδειγμα η μέτρηση της ταχύτητας, της επιτάχυνσης, της διεύθυνσης του σώματος σε σχέση με το μαγνητικό πεδίο της γης και η μεταβολή κάποιου μηχανικού μέρους, π.χ. των τροχών ενός ρομπότ. Παραδείγματα ιδιοδεκτικών αισθητήρων αποτελούν οι περιστροφικοί κωδικοποιητές (rotary encoders), τα επιταχυνσιόμετρα και τα γυροσκόπια. Η άλλη κατηγορία αποτελείται από τους εξωτερικούς (exteroceptive) αισθητήρες οι οποίοι καταγράφουν μεγέθη που σχετίζονται με το εξωτερικό περιβάλλον και τις ιδιότητες άλλων αντικειμένων, π.χ. ο ήχος και η ηλεκτρομαγνητική ακτινοβολία. Βασικά παραδείγματα εξωτερικών αισθητήρων αποτελούν τα συστήματα RADAR, τα LiDAR και οι κάμερες RGB.

2.1 LIDAR

Τα συστήματα σαρωτή laser αναπτύχθηκαν για πρώτη φορά τη δεκαετία του 1960 με σκοπό την ακριβή καταγραφή της γεωμετρικής πληροφορίας [1], μειώνοντας το χρόνο τόσο της αποτύπωσης όσο και της επεξεργασίας των δεδομένων σε σχέση με τις συμβατικές μετρήσεις. Πρόκειται για ένα σύστημα 3D οπτικής μέτρησης μέσω ανέπαφης λήψης και καταγραφής 3D πληροφορίας σχήματος, γεωμετρίας, υψής και χρώματος. Οι σαρωτές laser εντάσσονται στα ενεργά συστήματα οπτικής μέτρησης καθώς εκπέμπουν και στη συνέχεια καταγράφουν ηλεκτρομαγνητική ακτινοβολία laser μέσω της οποίας, σε συνδυασμό με μεθόδους όπως οι τεχνικές time-of-flight και phase-shift, μπορούν να υπολογίσουν την απόσταση περιοχής/σημείου του χώρου σε δεδομένη διεύθυνση και άρα την θέση του σημείου αυτού στο σύστημα του σαρωτή [2]. Πρακτικά λοιπόν, τα συστήματα αυτά ακολουθούν παρόμοια λογική με τους γεωδαιτικούς σταθμούς Total Station (αφού υπολογίζουν διεύθυνση, απόσταση και ορίζουν την θέση ενός σημείου στον χώρο), όμως μετράνε με πολύ μεγαλύτερη πυκνότητα και ταχύτητα προς διαφορετικές διευθύνσεις.

Ο όρος LiDAR (Light Detection and Ranging) προκύπτει σε αντιστοιχία με τον όρο Radar (Radio Detection and Ranging) καθώς συνήθως εκπέμπουν ακτινοβολία στο ορατό ή το εγγύς υπέρυθρο, σε αντίθεση με τα Radar που εκπέμπουν στα μικροκύματα.

Όσον αναφορά τις μεθόδους time-of-flight και phase-shift, η πρώτη λειτουργεί μέσω εκπομπής επαναλαμβανόμενων σύντομων παλμών και εντοπισμού των επιστροφών τους, με την απόσταση d από την πηγή στην επιφάνεια ανάκλασης να υπολογίζεται, μετρώντας τον χρόνο ταξιδιού του σήματος [3], βάσει της σχέσης:

$$d = c/n*t/2 \quad (1)$$

Οπού c η ταχύτητα του φωτός, t ο χρόνος ταξιδιού του σήματος και n ο δείκτης διάθλασης του μέσου. Πρακτικά δηλαδή εκπέμπεται σήμα σε συγκεκριμένη διεύθυνση το οποίο διανύει την απόσταση σημείου αντικειμένου δυο φορές και άρα η απόσταση αυτή ισούται με την ταχύτητα φωτός επί τον χρόνο επιστροφής σήματος δια δύο (καθώς ο δείκτης διάθλασης του αέρα είναι ίσος με την μονάδα).

Η δεύτερη μέθοδος λειτουργεί μέσω υπολογισμού της διαφοράς φάσης ($\Delta\Phi$) μεταξύ εκπεμπόμενου και λαμβανόμενου σήματος, ώστε να υπολογιστεί ο χρόνος διάδοσης t από την εξίσωση :

$$t = \Delta\Phi/2\pi*\lambda m/c \quad (2)$$

Με αυτήν την μέθοδο υπολογίζεται ο χρόνος ταξιδιού t με πολύ καλύτερη ακρίβεια από αυτή που θα υπολόγιζε ένα συμβατικό χρονόμετρο κατά την εφαρμογή της μεθόδου time-of-flight. Έτσι, η τελική απόσταση υπολογίζεται τελικά από την σχέση 1 αλλά αυτήν την φορά, με την νέα ακριβέστερη μέτρηση χρόνου που προκύπτει από την εξίσωση 2 μέσω της μεθόδου phase-shift.

Γενικά η μέθοδος phase-shift αποτυπώνει σημεία με πολύ μεγαλύτερη ταχύτητα (1 εκατ. σημεία ανά δευτερόλεπτο) έναντι της μεθόδου time-of-flight που συνήθως αποτυπώνει 50.000 σημεία το δευτερόλεπτο. Επίσης η phase-shift παρέχει αρκετά μεγαλύτερη ακρίβεια στην μέτρηση της απόστασης (ακρίβεια 0.1-3mm έναντι 5-10 mm). Τέλος, μπορεί να λειτουργήσει μόνο για κοντινές αποστάσεις (80-100m) σε αντίθεση με την time-of-flight που μπορεί να λειτουργήσει και για πολύ μεγαλύτερες αποστάσεις (μέχρι και μερικές χιλιάδες μέτρα) [4].

Τέλος, μέρος της υφής του αντικειμένου μπορεί να αποδοθεί μέσω καταγραφής της έντασης του λαμβανόμενου σήματος, αν και ο συνδυασμός κάμερας - σαρωτή ενδείκνυται για την λήψη των χρωμάτων του αντικειμένου με μεγαλύτερη ακρίβεια. Μέσω της διαδικασίας αυτής λαμβάνονται τελικά οι 3D θέσεις μεγάλου πλήθους σημείων (χρωματισμένων η μη) για τον περιβάλλοντα χώρο σε μορφή νέφους σημείων (point cloud).

Το LiDAR υπερτερεί έναντι άλλων αισθητήρων ιδίως σε περιπτώσεις όπου συνήθως οι οπτικοί δέκτες, αν και χαμηλότερου κόστους, αντιμετωπίζουν προβλήματα όπως π.χ. σε περιπτώσεις εδάφους με έντονη βλάστηση (πχ δάση), σε περιπτώσεις μετρήσεων σε συνθήκες χαμηλής φωτεινότητας, σε περιβάλλοντα χωρίς υφή ή με επαναλαμβανόμενα μοτίβα όπως και σε περιπτώσεις αποτύπωσης καλωδίων ή δυσδιάκριτων λεπτομερειών. Ένα ακόμη βασικό χαρακτηριστικό, αποτελεί η απευθείας μέτρηση της 3D πληροφορίας και το απλό μαθηματικό μοντέλο που περιγράφει τις γεωμετρικές σχέσεις μεταξύ των νεφών σημείων που παράγονται.

Αυτά αποτελούν τα βασικά χαρακτηριστικά και πλεονεκτήματα των αισθητήρων LiDAR, οι οποίοι μπορούν να ταξινομηθούν σε κατηγορίες με βάση διάφορα κριτήρια. Ένα από τα κριτήρια αυτά αποτελεί η φέρουσα πλατφόρμα μέσω της οποίας οι σαρωτές διαχωρίζονται σε τρεις κατηγορίες: τους επίγειους (terrestrial LiDAR), τους αερομεταφερόμενους (airborne LiDAR) και στα κινητά συστήματα LiDAR (mobile LiDAR) [5]. Τα στατικά επίγεια LiDAR χρησιμοποιούνται πιο συχνά σε εφαρμογές αποτύπωσης ακριβείας όπως είναι η αποτύπωση μνημείων. Ωστόσο, λόγω της σταθερής διαμόρφωσης και της υψηλής ανάλυσης, τα στατικά LiDAR χρειάζονται περισσότερο χρόνο για τη λήψη δεδομένων και συνεπώς δεν μπορούν να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου όπως απαιτεί η μέθοδος SLAM. Αντίθετα προτιμώνται σε στατικές αποτυπώσεις όπου απαιτείται υψηλή γεωμετρική ακρίβεια σε συνδυασμό με άλλες ακριβείς αλλά και σχετικά χρονοβόρες μεθόδους όπως η φωτογραμμετρική αποτύπωση και οι γεωδαιτικοί σταθμοί. Αντίθετα, τα κινητά συστήματα LiDAR προορίζονται για χρήση σε κινούμενες πλατφόρμες, όπως αυτόνομα οχήματα, ρομπότ ή ακόμη και συστήματα χειρός (handheld devices). Βέβαια, τα συστήματα αυτά παρέχουν μετρήσεις χαμηλότερης ανάλυσης από τα στατικά LiDAR, αλλά έχουν το πλεονέκτημα της πολύ ταχύτερης απόκτησης δεδομένων, με αποτέλεσμα να επιτρέπουν την παράλληλη κίνηση και καταγραφή μετρήσεων. Το γεγονός αυτό είχε ως συνέπεια τα κινητά συστήματα LiDAR να γίνουν αναπόσπαστο κομμάτι των τεχνικών SLAM και αντίστοιχα των εφαρμογών αυτόνομης πλοήγησης οχημάτων, ρομπότ και της χαρτογράφησης σε

πραγματικό χρόνο. Τέλος, τα αερομεταφερόμενα συστήματα LiDAR έχουν σχεδιαστεί με έμφαση την χρήση τους σε αεροπλάνα ή μη επανδρωμένα εναέρια οχήματα (UAV) και είναι ιδανικά για μεγάλης κλίμακας αναγνώριση εδάφους, παρακολούθηση καλλιεργειών και υδάτινων περιοχών και φυσικά για τοπογραφικές αποτυπώσεις.

Μια επιπρόσθετη ταξινόμηση των αισθητήρων αυτών προκύπτει με βάση την εσωτερική κατασκευή τους, αρχικά, σε συστήματα LiDAR που κατασκευάζονται με μηχανικά περιστρεφόμενα μέρη για τη λήψη τρισδιάστατων δεδομένων (mechanical LiDAR) [6]. Οι αισθητήρες αυτοί ωστόσο είναι πιο επιρρεπείς σε σφάλματα λόγω ακριβώς της ανάγκης για ταυτόχρονη μηχανική κίνηση του αισθητήρα, με αποτέλεσμα την εμφάνιση φθορών στον μηχανισμό τους, γεγονός που μπορεί να προκαλέσει μείωση της ακρίβειας. Τέλος, συνήθως παρουσιάζουν πολύ μικρό οπτικό πεδίο στον κατακόρυφο άξονα. Η άλλη κατηγορία αφορά τα LiDAR στερεάς κατάστασης (solid state). Αυτού του είδους οι αισθητήρες βασίζονται σε αποκλειστικά ηλεκτροοπτικές μεθόδους για την καταγραφή της 3D πληροφορίας. Οι αισθητήρες αυτοί έχουν πολύ μεγαλύτερο οπτικό πεδίο (κατακόρυφα και οριζόντια) κατά την πραγματοποίηση μιας μεμονωμένης λήψης, ενώ επίσης είναι μικρότεροι σε μέγεθος και παρέχουν υψηλότερη γεωμετρική ακρίβεια από τους αντίστοιχους μηχανικούς. Λόγω των παραπάνω, στην πλειοψηφία των εφαρμογών LiDAR-SLAM, χρησιμοποιούνται πλέον φορητά συστήματα LiDAR στέρεας καταστάσεις. Τα συστήματα LiDAR στερεάς κατάστασης χωρίζονται με την σειρά τους και αυτά σε δύο τύπους, τα LiDAR που αναβοσβήνουν (flashing) και τα LiDAR σάρωσης. Τα LiDAR που αναβοσβήνουν σε αντίθεση με αυτά της τεχνολογίας σάρωσης, αποτελούνται από συστοιχία αρκετών αισθητήρων οι οποίοι καταγράφουν την τρισδιάστατη πληροφορία, αυτή η διάταξη μπορεί να πετύχει σημαντικά υψηλότερους ρυθμούς λήψης δεδομένων και υψηλή αξιοπιστία, ωστόσο καταναλώνει μεγαλύτερη ισχύ και είναι ακριβότερη [7].

Μια άλλη κατηγοριοποίηση προκύπτει με βάση το εάν καταγράφουν σημεία σε ένα επίπεδο (2D σάρωση) ή σε περισσότερα (3D σάρωση) [8]. Τα συστήματα 2D LiDAR μπορούν να αποτυπώσουν 3D πληροφορία μέσω μηχανικής περιστροφής του αισθητήρα και προτιμούνταν κατά το παρελθόν, όπου οι 3D LiDAR αισθητήρες ήταν ακριβότεροι και παρήγαγαν πολύ μεγάλο όγκο δεδομένων για τις παρούσες δυνατότητες των υπολογιστικών συστημάτων. Τέτοιες εφαρμογές περιλαμβάνουν κάποιες πρώτες προσπάθειες αυτόματης πλοήγησης και κυρίως ρομπότ που κινούνταν σε επίπεδο έδαφος εσωτερικών χώρων όπως πχ τα ρομπότ σκούπες. Αυτές οι συνθήκες οδήγησαν αντίστοιχα και τους πρώτους αλγορίθμους LiDAR-SLAM να χρησιμοποιήσουν αυτούς τους αισθητήρες, όμως αργότερα με την μείωση του κόστους των αισθητήρων, την αύξηση της υπολογιστικής ισχύος και τα σημαντικά μειονεκτήματα αποτύπωσης έναντι των 3D LiDAR, εγκαταλείφθηκαν σχεδόν πλήρως από τις νεότερες εφαρμογές SLAM και γενικότερα αποτύπωσης.

2.1.1 Τεχνικά Χαρακτηριστικά

Τα τεχνικά χαρακτηριστικά των συστημάτων LiDAR μέσω των οποίων διαμορφώνονται επιπρόσθετες κατηγορίες αισθητήρων αυτού του είδους είναι η μέγιστη εμβέλεια, το οπτικό πεδίο, η χωρική ανάλυση, η ανάλυση εύρους, η ακρίβεια και ο ρυθμός λήψης δεδομένων. Όλοι οι προηγούμενοι παράγοντες συνδέονται στενά με την εσωτερική δομή στην οποία έγινε αναφορά παραπάνω. Η μέγιστη εμβέλεια, καθορίζει τη μέγιστη απόσταση από τον αισθητήρα στην οποία μπορούν να ανιχνευθούν αντικείμενα και μπορεί να κυμαίνεται από λίγα εκατοστά έως εκατοντάδες μετρά. Το οπτικό πεδίο καθορίζει το γωνιακό εύρος στο οποίο μπορούν να συλλεχθούν οι παρατηρήσεις, με το οριζόντιο οπτικό πεδίο να φτάνει έως και τις 360°, ενώ το κατακόρυφο οπτικό πεδίο είναι συνήθως πιο περιορισμένο και κυμαίνεται από μερικές μοίρες έως 90° [8]. Η χωρική ανάλυση είναι επίσης ένα σημαντικό χαρακτηριστικό και σχετίζεται με την ελάχιστη γωνιακή μετατόπιση της ακτίνας laser στο οριζόντιο και κάθετο επίπεδο. Ακολουθως, η ανάλυση εύρους αφορά τη ελάχιστη απόσταση μεταξύ αντικειμένων που μπορούν να ανιχνευθούν ανεξάρτητα και καθορίζεται από τη διάρκεια του παλμού της δέσμης, την αναλογία σήματος προς τον θόρυβο και την φύση της επιφάνειας [9]. Επίσης, ένα από τα βασικότερα χαρακτηριστικά είναι η τελική ακρίβεια και αναφέρεται στην 3D γεωμετρική ακρίβεια των σημείων κατά την μέτρηση ενός νέφους. Τέλος, ο ρυθμός λήψης δεδομένων αφορά την ταχύτητα καταγραφής των μετρήσεων και συνήθως κυμαίνεται από 5 έως 100 Hz (άρα ο αισθητήρας στην μια περίπτωση μπορεί να καταγράψει πέντε νέφη ανά δευτερόλεπτο και στην άλλη εκατό νέφη το δευτερόλεπτο) [10].

2.1.2 Αλληλοαναφορά νεφών σημείων

Σε αυτό το τμήμα θα ακολουθήσει μια μικρή περιγραφή της βασική σχέσης η οποία διέπει τις διαδοχικές σαρώσεις που πραγματοποιεί ένας αισθητήρας LiDAR. Αρχικά σε κάθε σάρωση καταγράφεται η 3D θέση (X,Y,Z ή γωνίες και απόστασή) στο σύστημα του σαρωτή όλων των σημείων που είναι εντός του πεδίου ορατότητας και του εύρους σάρωσης του αισθητήρα. Ως αρχή των αξόνων ορίζεται το σημείο λήψης της σάρωσης (κέντρο σάρωσης). Το σύστημα λοιπόν κάθε σάρωσης, δεδομένης κινούμενης πλατφόρμας, θα διαφέρει από το σύστημα της προηγούμενης κατά έναν μετασχηματισμό που θα αναλύεται σε μετάθεση και στροφή. Προφανώς η 3D μετατόπιση και οι στροφές ισούνται με αυτές που πραγματοποίησε ο αισθητήρας μεταξύ των δυο σαρώσεων. Για να έχουμε ωστόσο έναν ενιαίο χάρτη με όλα τα σημεία της αλληλουχίας σαρώσεων που καταγράφηκαν, πρέπει τα σημεία όλων των σαρώσεων να είναι στο ίδιο σύστημα, οπού συνήθως επιλέγεται να είναι το σύστημα της πρώτης σάρωσης [12]. Η διαδικασία κατά την οποία μεταφέρεται ένα νέφος σημείων στο σύστημα ενός άλλου ονομάζεται αλληλοαναφορά (alignment) των νεφών σημείων ή συνένωση (registration) ή συνταύτιση (scan matching). Για να πραγματοποιηθεί λοιπόν η αλληλοαναφορά μεταξύ των δυο νεφών σημείων πρέπει να είναι γνωστές με ακρίβεια οι τρεις μεταθέσεις και οι τρεις στροφές στον χώρο που αντιστοιχούν σε έναν 3D μετασχηματισμό στερεού σώματος. Φυσικά οι

παράμετροι του μετασχηματισμού αυτού δεν είναι ποτέ γνωστοί με ακρίβεια και πρέπει να υπολογιστούν από μεθόδους που προσδιορίζουν με ακρίβεια τον μετασχηματισμό αυτό όπως είναι για παράδειγμα ο αλγόριθμος ICP (Iterative Closest Point) [36]. Προφανώς, για να λειτουργήσει ο αλγόριθμος ICP πρέπει τα νέφη σημείων να είναι αλληλεπικαλυπτόμενα σε κάποιο βαθμό, ενώ τα δυο νέφη να βρίσκονται ήδη αρκετά κοντά μεταξύ τους (γνώση καλών προσεγγιστικών τιμών). Για να επιτευχθεί αυτό, συνήθως πραγματοποιείται αρχικά ένας προσεγγιστικός μετασχηματισμός στερεού σώματος ο οποίος απαιτεί τουλάχιστον τρία σημεία γνωστά και στα δυο συστήματα για να προσδιοριστούν όλες οι παράμετροι του προσεγγιστικού μετασχηματισμού (τρεις παράμετροι στροφών και τρεις μετάθεσης ($\Delta X, \Delta Y, \Delta Z, \Omega, \Phi, \kappa$) [4]. Συνήθως στη γενική περίπτωση επιλέγονται χοντρικά 4 ομόλογα σημεία σε κάθε σάρωση (υποχρεωτική χειρωνακτική μέτρηση τουλάχιστον τριών σημείων) ώστε να μπορέσει να γίνει ένας αρχικός μετασχηματισμός περιορισμένης ακρίβειας ο οποίος όμως αποτελεί καλή προσέγγιση για τον ICP.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad \begin{array}{l} \bullet \text{ R: Πίνακας στροφής (3 στροφές)} \\ \bullet \text{ T: Διάνυσμα μετάθεσης (3 μεταθέσεις)} \end{array} \quad (3)$$

Εφόσον έχει πραγματοποιηθεί η εύρεση του αρχικού μετασχηματισμού μπορεί να πραγματοποιηθεί αυτόματα η βέλτιστη αλληλοαναφορά των δυο νεφών μέσω του ICP [11]. Ο αλγόριθμος αυτός υπολογίζει επαναληπτικά τον βέλτιστο μετασχηματισμό μέσω διαδοχικής ελαχιστοποίησης της ασυμβατότητας των δύο νεφών (απόστασης μεταξύ των ομολόγων σημείων). Αρχικά, για κάθε σημείο του τρέχοντος νέφους σημείων εντοπίζεται το ομόλογό του από το νέφος αναφοράς ως το πλησιέστερο σημείο. Ελαχιστοποιώντας τη μέση τετραγωνική απόσταση (error) των δύο ομολόγων σημειοσυνόλων υπολογίζεται ο 3D μετασχηματισμός στερεού σώματος (σε συνδυασμό με χρήση βαρών και απαλοιφή εσφαλμένων σημείων). Ακολούθως, γίνεται εφαρμογή του μετασχηματισμού στο τρέχον νέφος σημείων και έτσι ολοκληρώνεται μια πλήρης επανάληψη του αλγορίθμου. Η διαδικασία επαναλαμβάνεται μέχρι την επιθυμητή σύγκλιση (σφάλμα μικρότερο από κάποιο όριο).

Στην περίπτωση της μεθόδου LiDAR-SLAM οι αρχικές προσεγγίσεις του μετασχηματισμού υπολογίζονται συνήθως αυτόματα και ανάλογα με την περίπτωση, με πιο συνηθισμένες προσεγγίσεις να αφορούν τον αυτόματο εντοπισμό ομολογιών μεταξύ των δυο νεφών σημείων έχοντας ως βάση αυτόματα υπολογισμένους 3D περιγραφείς [32] [41]. Μια άλλη προσέγγιση χρησιμοποιεί τις μετρήσεις αδρανειακού συστήματος IMU/INS ως αρχικές προσεγγίσεις καθώς αυτό το σύστημα μπορεί να υπολογίσει τον προσεγγιστικό μετασχηματισμό μεταξύ των δυο θέσεων που βρέθηκε ο σαρωτής. Τέλος, μπορεί να χρησιμοποιηθεί ένας συνδυασμός και των δυο μεθόδων για ακόμη ακριβέστερα αποτέλεσμα. Φυσικά το τελικό αποτέλεσμα μετά τον ICP, πέρα από τον χάρτη των συνενωμένων σημείων παρέχει και όλες τις θέσεις του σαρωτή στο σύστημα της πρώτης σάρωσης. Εφόσον

ο σαρωτής στο σύστημα της πρώτης σάρωσης βρίσκεται στην αρχή των αξόνων το οι θέσεις του σαρωτή μεταξύ των καταγράφων υπολογίζονται από την διαδοχική εφαρμογή των μετασχηματισμών στερεού σώματος που υπολογίστηκαν από τον ICP στο σημείο αυτό. Βέβαια χωρίς κάποια μέθοδο βελτιστοποίησης ή εντοπισμού βρόχων η μέθοδος αυτή συγκεντρώνει διαδοχικά σφάλματα και δεν αποτελεί από μόνη της ολοκληρωμένη μέθοδο SLAM και καλείται συνήθως LiDAR Odometry.

2.1.3 Αναπαράσταση της 3D πληροφορίας

Όπως αναφέρθηκε και παραπάνω οι μετρήσεις που παρέχουν οι αισθητήρες LiDAR αφορούν νέφη σημείων. Ωστόσο, το πλήθος των σημείων αυτών μπορεί να είναι αρκετά μεγάλο γεγονός που τα καθιστά δύσκολο να τα διαχειριστεί κανείς υπολογιστικά. Για αυτόν τον λόγο, σε πολλές περιπτώσεις προτιμώνται εναλλακτικές αναπαραστάσεις οι οποίες είναι πιο "ελαφριές" υπολογιστικά. Μια τέτοια περίπτωση είναι το voxel (συντομογραφία του "volume pixel"), το οποίο αποτελεί μια θεμελιώδη μονάδα τρισδιάστατου χώρου που χρησιμοποιείται ευρέως σε διάφορους τομείς όπως είναι τα Γραφικά Υπολογιστών [13]. Μπορεί, όπως μαρτυρά και η ονομασία, να θεωρηθεί ως το αντίστοιχο τρισδιάστατο ισοδύναμο του εικονοσημείο (pixel), το οποίο αντιπροσωπεύει ένα μόνο σημείο σε μια δισδιάστατη εικόνα. Αντίθετα, ένα voxel αντιπροσωπεύει έναν όγκο συγκεκριμένων διαστάσεων σε έναν τρισδιάστατο χώρο και υλοποιείται συνήθως σε μια κανονική δομή πλέγματος που χωρίζει τον χώρο σε πλήθος από voxel, διαμορφώνοντας έτσι μια δειγματοληπτική μονάδα του 3D χώρου.

Τα voxel χρησιμοποιούνται ευρέως σε τρισδιάστατα μοντέλα και παιχνίδια λόγω της ικανότητάς τους να αναπαριστούν και να χειρίζονται ογκομετρικά δεδομένα αλλά και στην τρισδιάστατη μοντελοποίηση καθώς τα voxel μπορούν να χρησιμοποιηθούν για την κατασκευή πολύπλοκων αντικειμένων και περιβαλλόντων [14]. Κάθε voxel μπορεί να αποθηκεύσει διάφορα χαρακτηριστικά, όπως χρώμα, διαφάνεια και ιδιότητες υλικού, επιτρέποντας τη δημιουργία λεπτομερών και ρεαλιστικών σκηνών. Με βάση αυτήν την ιδιότητα, οι τεχνικές μοντελοποίησης προσφέρουν πολλά πλεονεκτήματα όπως είναι η απλή αναπαράσταση στερεών αντικειμένων, καθώς τα voxel κωδικοποιούν άμεσα την κατάληψη του χώρου. Αυτό τα καθιστά κατάλληλα για εφαρμογές τεχνικών SLAM όπου τα σημεία του περιβάλλοντος πρέπει να αναπαρασταθούν με ακρίβεια. Επιπλέον, τα μοντέλα αυτά υποστηρίζουν εύκολη και γρήγορη τροποποίηση δεδομένων αφού τα μεμονωμένα voxel μπορούν να προστεθούν, να αφαιρεθούν ή να μετασχηματιστούν ανεξάρτητα.

Τα παραπάνω πλεονεκτήματα είναι ένας βασικός λόγος που τα voxel και άλλες αντίστοιχες μορφές αναπαράστασης αποτελούν μια από τις κύριες μορφές δομής δεδομένων για τα 3D δεδομένα που χρησιμοποιούν οι τεχνικές SLAM καθώς συχνά υπάρχει ανάγκη για δειγματοληψία και ελάττωση των μετρήσεων ώστε να αποφευχθεί ο τεράστιος όγκος δεδομένων ο οποίος θα δημιουργούσε προβλήματα στην εφαρμογή των τεχνικών σε πραγματικό χρόνο.

2.2 Συμπληρωματικά Συστήματα Εντοπισμού Θέσης

2.2.1 Μονάδα Αδρανειακών Μετρήσεων – IMU/INS

Οι μονάδες αδρανειακών μετρήσεων έκαναν την εμφάνισή τους το 1930 σε εφαρμογές πλοήγησης αεροσκαφών [15], ενώ σήμερα, χρησιμοποιούνται ευρέως σε σχεδόν κάθε κινητό σύστημα μετρήσεων ως συμπληρωματικός αισθητήρας. Μερικοί τομείς που αξιοποιούν τα συστήματα αυτά αποτελούν η πλοήγηση οχημάτων και UAV και η ρομποτική, ενώ παράλληλα αποτελούν πλέον αναπόσπαστο κομμάτι της πλειονότητας των συστημάτων SLAM ως συμπληρωματικές πηγές δεδομένων για τον υπολογισμό της θέσης του σαρωτή. Τα IMU αποτελούν σύνολα ιδιοδεκτικών αισθητήρων οι οποίοι μπορούν να καταγράψουν πληροφορίες για την ταχύτητα και τον προσανατολισμό τους όπως είναι για παράδειγμα τα γυροσκόπια και τα επιταχυνσιόμετρα [15]. Οι διατάξεις αυτές προσδιορίζουν την θέση, την ταχύτητα και την κινηματική συμπεριφορά ενός κινητού σώματος κάνοντας χρήση των αρχών της αδράνειας και της Νευτώνειας μηχανικής. Το γυροσκόπιο, το οποίο λειτουργεί σύμφωνα με την αρχή της διατήρησης της αδράνειας και της στροφορμής [77], διατηρεί σταθερό τον προσανατολισμό του έχοντας καθορισμένη την αρχική του διεύθυνση. Αυτό σημαίνει ότι μετρώντας την γωνιακή περιστροφή και προσδιορίζοντας τις διορθώσεις που πρέπει να επιβληθούν στους φορείς τους μπορούν να διατηρούν τη διεύθυνσή τους σταθερή. Ύστερα, τα επιταχυνσιόμετρα βασίζονται τη λειτουργία τους στον 2^ο νόμο του Νεύτωνα μέσω ενός αδρανειακού συστήματος μάζας, το οποίο βρίσκεται στερεωμένο στο εσωτερικό τους ανάμεσα σε δύο ελατήρια. Η κίνηση του φορέα προκαλεί τη δημιουργία δυνάμεων που μετατοπίζουν τη μάζα αυτή, συσπειρώνοντας ή επιμηκύνοντας ανάλογα τα ελατήρια, ύστερα μέσα από την σχέση επιτάχυνσης-παραμόρφωσης των ελατήριων μπορεί τελικά να προσδιοριστεί η επιτάχυνση του σώματος [16]. Τέλος, συχνά γίνεται χρήση και μαγνητόμετρων για την ενίσχυση της ακρίβειας του συστήματος.

Μέσω των παραπάνω παρατηρήσεων τελικά παρατηρεί κανείς πως μπορούν να υπολογιστούν πλήρως οι παράμετροι κίνησης (3D στροφές και μεταθέσεις) ενός σώματος χωρίς καμία εξωτερική πληροφορία. Με αυτόν τον τρόπο παρέχεται η δυνατότητα ενός είδους τυφλής πλοήγησης (dead reckoning) [17] από σημείο σε σημείο προσδιορίζοντας κάθε φορά την 3D θέση του αισθητήρα σε σχέση με το αδρανειακό σύστημα που υλοποιείται με τη βοήθεια των IMU. Ωστόσο στην πράξη αυτή η μέθοδος αποφεύγεται καθώς οι μετρήσεις αυτές συσσωρεύουν προοδευτικά σφάλματα, ειδικά γωνιών, χωρίς την δυνατότητα διόρθωσης εάν δεν παρέχεται κάποια εξωτερική πληροφορία για να δεσμεύσει το σύστημα. Για αυτόν τον λόγο οι αισθητήρες αυτού του είδους χρησιμοποιούνται πάντα συμπληρωματικά καθώς οι παρατηρήσεις τους μπορούν να αξιοποιηθούν ώστε να βελτιώσουν την ακρίβεια των εξωτερικών μετρήσεων, να δώσουν καλές προσεγγιστικές τιμές σε μη γραμμικά προβλήματα, ενώ παράλληλα οι μετρήσεις τους είναι πάντα διαθέσιμες ανεξάρτητα από τις εξωτερικές συνθήκες. Χάρη στα παραπάνω προτερήματα ενσωματώνονται πλέον στις περισσότερες εφαρμογές σύγχρονων τεχνικών SLAM.

2.2.2 Συστήματα Δορυφορικού Εντοπισμού GNSS

Τα παγκόσμια δορυφορικά συστήματα πλοήγησης GNSS (Global Navigation Satellite Systems) αφορούν την ικανότητα προσδιορισμού της θέσης (συντεταγμένων) ενός σημείου στον χώρο ανεξαρτήτως καιρικών συνθηκών, φωτισμού και απόστασης με μόνη προϋπόθεση για τον εντοπισμό της 3D θέσης (όχι στροφών) να είναι η επικοινωνία μεταξύ του αισθητήρα και τουλάχιστον τεσσάρων δορυφόρων [18]. Ο προσδιορισμός αυτός επιτυγχάνεται μέσω των δορυφόρων οι οποίοι βρίσκονται σε γνωστές θέσεις κάθε στιγμή έτσι όταν ένας αισθητήρας μπορεί να καταγράψει το ηλεκτρομαγνητικό σήμα τεσσάρων τουλάχιστον δορυφόρων, δύναται να προσδιορίσει την θέση του στον χώρο σαν τομή σφαιρών, καθώς η απόστασή είναι γνωστή και ίση με τον χρόνο ταξιδιού του σήματος, ενώ παράλληλα είναι γνωστή και η θέση των δορυφόρων .

Έχουν ακρίβεια που μπορεί να κυμαίνεται από μερικά μετρά έως μερικά χιλιοστά ανάλογα με το είδος του δέκτη και την τεχνική που χρησιμοποιείται. Αυτή η ακρίβεια οφείλεται συνήθως σε σφάλματα που οφείλονται στην ακρίβεια μέτρησης του χρόνου από τους δέκτες και τους δορυφόρους με αποτέλεσμα η ακρίβεια να είναι μειωμένη , ωστόσο είναι δυνατή η λήψη διορθώσεων από σφάλματα μέσω μέτρησης ενός κοντινού γνωστού σημείου. Σε εφαρμογές πραγματικού χρόνου όπου η τεχνική GNSS που χρησιμοποιείται είναι η RTK (Real Time Kinematic), μπορούν να επιτευχθούν ακρίβειες μερικών χιλιοστών μόνο υπό συγκεκριμένες περιπτώσεις ενώ συνήθως επιτυγχάνουν ακρίβεια μερικών εκατοστών ή μέτρων [19]. Επίσης αυτή η τεχνική μέτρησης παρουσιάζει και κάποια αλλά πολύ σημαντικά μειονεκτήματα όπως είναι η αδυναμία εντοπισμού θέσης σε περιπτώσεις όπου δεν μπορεί να επικοινωνήσει ο δέκτης με τους δορυφόρους (δεν έχει ανοικτό ορατό πεδίο προς το διάστημα) και άρα η χρήση τους είναι αδύνατη σε κλειστούς χώρους αλλά και σε έντονα αστικά ή δασώδη περιβάλλοντα. Μια άλλη σημαντική αδυναμία είναι η πιθανότητα ότι, ακόμη και αν ο δέκτης καταφέρει εν τέλει να εντοπίσει την θέση, το σφάλμα ενδέχεται να είναι απαγορευτικό για εφαρμογές πλοήγησης ακρίβειας όπως είναι η αυτόνομη οδήγηση και η πλοήγηση των ρομπότ, ενώ επίσης δεν μας δίνει καθόλου πληροφορίες για την διεύθυνση και τις στροφές, πάρα μόνο για την θέση του δέκτη στον χώρο. Κατανοεί κανείς φυσικά ότι για τους παραπάνω λόγους η χρήση GNSS δεν θα μπορούσε ποτέ να αποτελέσει βασικό πυλώνα σε τεχνικές SLAM και πλοήγησης, ωστόσο μπορεί να λειτουργήσει συνδυαστικά σε περιπτώσεις όπου είναι διαθέσιμο με καλή ακρίβεια, ειδικά σε εφαρμογές μεγάλης κλίμακας, ώστε να ελαττώσει το συσσωρευμένο σφάλμα που συνήθως δημιουργείται [20].

2.2.3 Περιστροφικοί κωδικοποιητές (Rotary encoders)

Πρόκειται για αισθητήρες, οι οποίοι μέσω ηλεκτρομηχανικών μικροσυσκευών που τοποθετούνται σε τροχούς, μπορούν να καταγράψουν το μήκος που έχει διανθεί συνολικά αλλά και τυχόν μεταβολές στις διευθύνσεις των τροχών [21]. Οι παρατηρήσεις των αισθητήρων αυτών δεν είναι μεγάλης ακρίβειας, ωστόσο μπορούν να ληφθούν υπόψιν ως επιπλέον δέσμευση σε εφαρμογές LiDAR-SLAM για ρομπότ με τροχούς ή οχήματα [22].

3. ΑΛΓΟΡΙΘΜΟΙ LIDAR SLAM

Με τις ραγδαίες εξελίξεις στον τομέα της Τεχνητής Νοημοσύνης, τα ρομποτικά συστήματα αναλαμβάνουν ολοένα και περισσότερο τον ρόλο του ανθρώπου σε πληθώρα πρακτικών εφαρμογών, αυξάνοντας την αποτελεσματικότητα και την ασφάλεια. Ως αποτέλεσμα, αυτά τα ρομποτικά συστήματα, συμπεριλαμβανόμενων των αισθητήρων που τα αποτελούν, τείνουν να γίνουν αναπόσπαστο κομμάτι της καθημερινής ζωής.

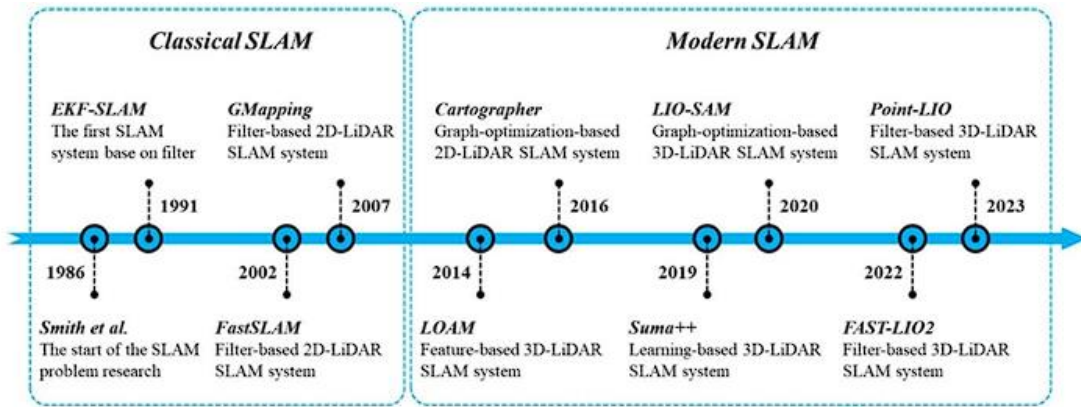
Ο εντοπισμός θέσης μέσα σε άγνωστα περιβάλλοντα αποτελεί βασικό τομέα έρευνας στην ρομποτική, με την μέθοδο SLAM να κατέχει κεντρική θέση. Σε σύγκριση με τις κάμερες, η χρήση της τεχνολογίας LiDAR προσφέρει σημαντικά πλεονεκτήματα, καθώς δεν επηρεάζεται από την υφή των αντικειμένων αλλά και τις συνθήκες φωτισμού επιτρέποντας έτσι μετρήσεις υψηλής ακρίβειας μετρήσεις αποστάσεων σε κάθε περίπτωση [23].

Αυτά οδήγησαν τα συστήματα LiDAR-SLAM στο να αναπτυχθούν εκτενώς σε διάφορους τομείς όπως είναι η αυτόνομη οδήγηση [24], η πλοήγηση ρομποτικών συστημάτων, η αποτύπωση/χαρτογράφηση αστικών περιοχών [25] και η αποτύπωση δασικών περιοχών [26]. Επιπλέον, αναμένεται όλο και μεγαλύτερη αύξηση του πλήθους εφαρμογών που μπορεί να αξιοποιήσουν τις τεχνικές LiDAR-SLAM στο κοντινό μέλλον με την αύξηση της αυτοματοποίησης.

3.1 Ιστορική αναδρομή

Για πρώτη φορά οι Smith και Cheeseman (1986) ήταν αυτοί που συνδύασαν τα προβλήματα εντοπισμού και χαρτογράφησης των ρομπότ κάτω από ένα θεωρητικό πλαίσιο βασισμένο στη θεωρία πιθανοτήτων. Αυτός ο συνδυασμός των δυο προβλημάτων σε ένα μπορεί να θεωρηθεί η αρχή της μελέτης προβλημάτων SLAM [23][27].

Ως ορός στην βιβλιογραφία το SLAM εμφανίζεται το 2006, όπου οι Durrant-Whyte και Bailey τον χρησιμοποίησαν για πρώτη φορά στις ερευνητικές τους εργασίες [23][28]. Η περίοδος μεταξύ 1986 και 2007 αναφέρεται στην βιβλιογραφία ως κλασική περίοδος της διαδικασίας SLAM περιλαμβάνοντας αλγορίθμους με LiDAR όπως ο EKF SLAM [29], ο FastSLAM [30] και ο GMapping [31]. Υστέρα ακολουθεί η περίοδος από το 2014 έως και σήμερα η οποία αναφέρεται στην βιβλιογραφία ως περίοδος σύγχρονων μεθόδων SLAM με βασικό εναρκτήριο ορόσημο στο χώρο του LiDAR-SLAM να ορίζεται ο LOAM [32]. Αυτή η χρονολογική προσέγγιση και κατηγοριοποίηση με βάση το έτος κυκλοφορίας της εκάστοτε μεθόδου τους προτείνεται από το [23] και παρουσιάζεται υπό την μορφή χρονογραμμής στην εικόνα 1 η οποία παρουσιάζεται παρακάτω .



Εικόνα 1: Χρονογραμμή LiDAR-SLAM [23]

Κατά την διάρκεια της κλασσικής περιόδου η κύρια προσέγγιση για την επίλυση προβλημάτων SLAM ήταν μέσω εφαρμογής τεχνικών Filtering, δηλαδή μέσω φίλτρων κατάστασης τα οποία βασίζονται στο Θεώρημα Bayes (Kalman, Extended Kalman, Particle). Τα φίλτρα αυτά αποτελούν μαθηματικά μοντέλα που στοχεύουν στον εκτιμητικό-πιθανοθεωρητικό υπολογισμό της τρέχουσας κατάστασης ενός συστήματος, όπου στην περίπτωση της μεθόδου SLAM είναι η θέση και ο προσανατολισμός με βάση ένα μοντέλο παρατήρησης. Για να λυθεί το SLAM ως ένα δίκτυο Bayes, ο αλγόριθμος φιλτραρίσματος πρέπει να συλλέγει πληροφορίες κάθε στιγμή σε πραγματικό χρόνο και να τις λαμβάνει υπόψιν στην κατανομή πιθανότητας του δικτύου. Αυτή η μέθοδος φιλτραρίσματος αντιπροσωπεύει ένα συνεχώς εναλλασσόμενο δίκτυο παρατηρήσεων, το οποίο όπως είναι προφανές, επιφέρει σημαντική υπολογιστική επιβάρυνση και μπορεί να δημιουργήσει χάρτες μόνο σε μικρή κλίμακα.

Για χαρτογράφηση μεγάλης κλίμακας σκηνών, έχουν πλέον επικρατήσει στρατηγικές βελτιστοποίησης μέσω μεθόδων Γράφου Παραγόντων (Factor Graph). Η μέθοδος βελτιστοποίησης είναι το αντίστροφο της μεθόδου φιλτραρίσματος και λειτουργεί συσσωρεύοντας τις πληροφορίες που αποκτήθηκαν ενώ υπολογίζει εκτός σύνδεσης την τροχιά της οντότητας, χρησιμοποιώντας τις συνολικές πληροφορίες που έχουν συλλεχθεί έως εκείνη την στιγμή (Global optimization). Με την τεράστια αύξηση της υπολογιστικής ισχύος, οι προσεγγίσεις που βασίζονται στη βελτιστοποίηση έχουν κυριαρχήσει στις σύγχρονες μεθόδους SLAM [23].

Την τελευταία δεκαετία, το πρόβλημα του LiDAR-SLAM είθισται να διαχωρίζεται στις εξής υποκατηγορίες: στην προσέγγιση της βελτιστοποίησης γράφου (pose graph optimization/factor graph optimization) και στην προσέγγιση βελτιστοποίησης μέσω πιθανοθεωρητικών τεχνικών που βασίζονται στο θεώρημα Bayes, μέσω χρήσης βελτιωμένων φίλτρων κατάστασης [33]. Τέλος, αξίζει να σημειωθεί ότι μια ειδική υποκατηγορία που προέκυψε την τελευταία πενταετία αποτελούν οι Learning Based αλγόριθμοι οι οποίοι βασίζονται σε τεχνικές μηχανικής εκμάθησης για την εύρεση της συνάρτησης κόστους της βελτιστοποίησης.

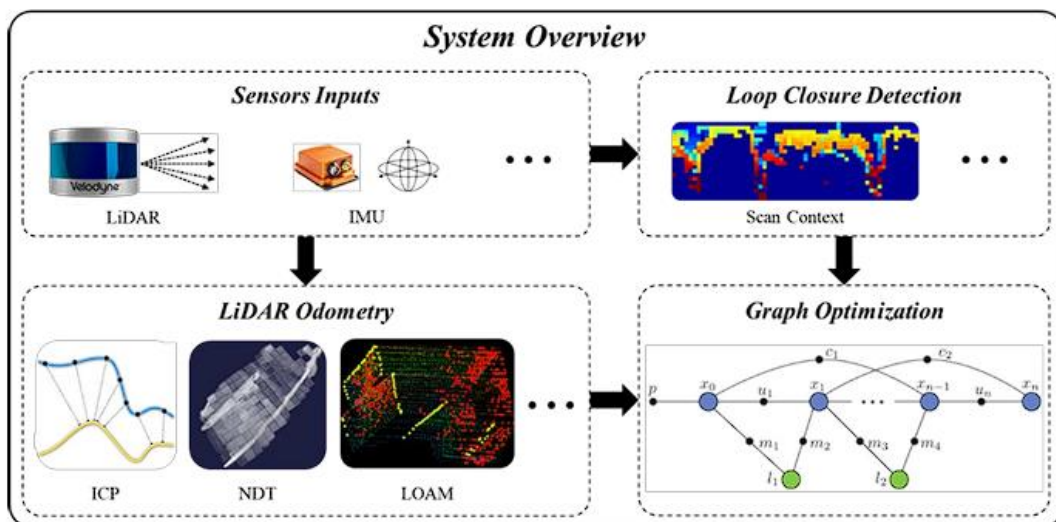
3.2 Σύγχρονη μέθοδος LiDAR SLAM

Το βασικό μοντέλο παρατήρησης στην περίπτωση του LiDAR-SLAM είναι σχετικά απλό καθώς η παρατήρηση συνδέεται άμεσα με την κατάσταση του συστήματος μέσω ενός 3D μετασχηματισμού στερεού σώματος και προκύπτει ως εξής [33] :

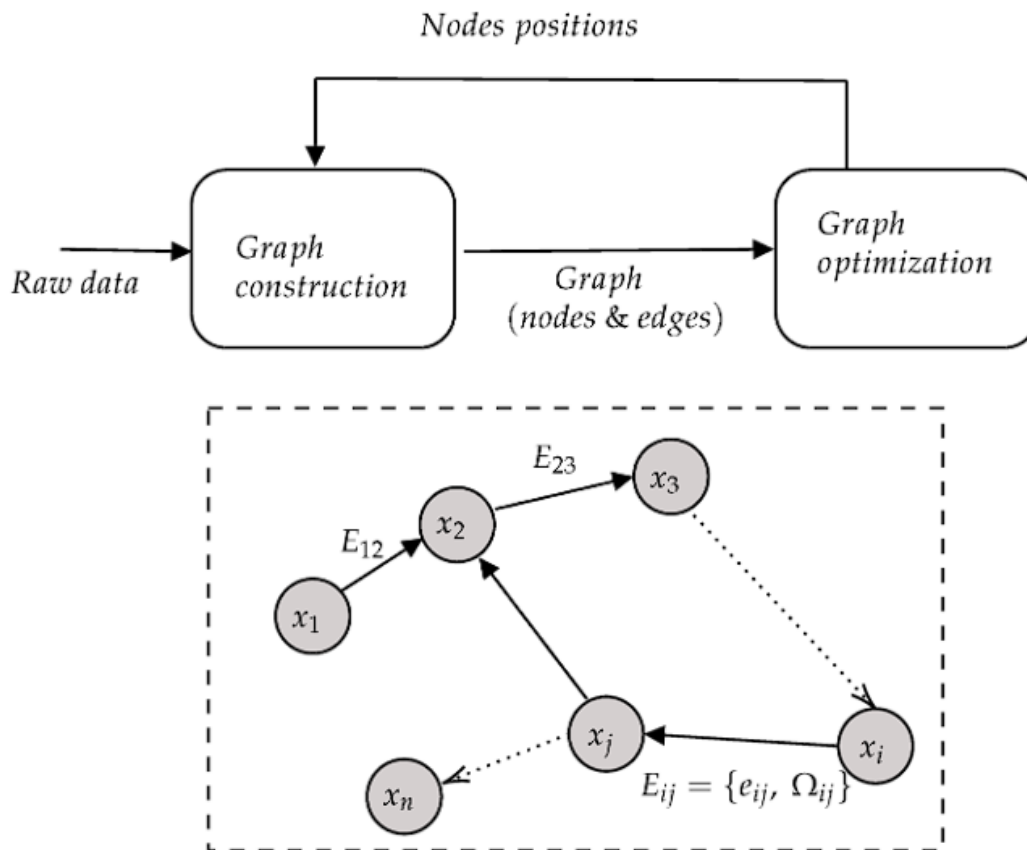
$$P3Dmap = T(P3DLiDAR) \quad (4)$$

Όπου $T = [R, t]$ ο μετασχηματισμός στερεού σώματος, με R και t να είναι οι στροφές και οι μεταθέσεις που χαρακτηρίζουν τον μετασχηματισμό. Σύμφωνα με το μοντέλο αυτό, μπορεί να εφαρμοστεί είτε μια πιθανοθεωρητική προσέγγιση μέσω φιλτραρίσματος είτε η προσέγγιση της βελτιστοποίησης γράφου. Ωστόσο, τις περισσότερες φορές στις σύγχρονες υλοποιήσεις της μεθόδου SLAM προτιμάται η δεύτερη προσέγγιση, όπως φαίνεται στην Εικόνα 2. Στο τελικό στάδιο αυτών των συστημάτων βελτιστοποιείται το σύνολο των σχετικών μετασχηματισμών των νεφών που έχουν προκύψει Οδομετρικά, μέσω επιπλέον δεσμεύσεων οι οποίες εισάγονται στο σύστημα μέσω εντοπισμού κλειστών βρόχων (loop closure detection) ή από παρατηρήσεις άλλων αισθητήρων [34].

Η βελτιστοποίηση του γράφου δημιουργεί ένα πιο απλό πρόβλημα εκτίμησης το οποίο περιγράφει ένα δίκτυο θέσεων και περιορισμών [35]. Οι παρατηρήσεις αντικαθίστανται από τις ακμές του γράφου, τις οποίες μπορούμε να θεωρήσουμε ως "εικονικές παρατηρήσεις". Πράγματι, μια τέτοια ακμή σημειώνεται με μια κατανομή πιθανότητας που βασίζεται στις σχετικές θέσεις των δύο θέσεων/κόμβων που ενώνει η ακμή αυτή, υπό την δέσμευση των αμοιβαίων τους μετρήσεων. Στο σχήμα 1, εμφανίζεται η διαδικασία που περιγράφηκε και η οποία αποτελείται από δύο κύρια μπλοκ: την κατασκευή γράφου και την βελτιστοποίηση γράφου με την βελτιστοποίηση να επικεντρώνεται στον υπολογισμό των βέλτιστων κόμβων (θέσεων) με βάση τους περιορισμούς που επιβάλλουν οι ακμές.



Εικόνα 2: Επισκόπηση σύγχρονου LiDAR-SLAM [23]



Σχήμα 1: Αναπαράσταση Γράφου Θέσης (Pose Graph) [33]

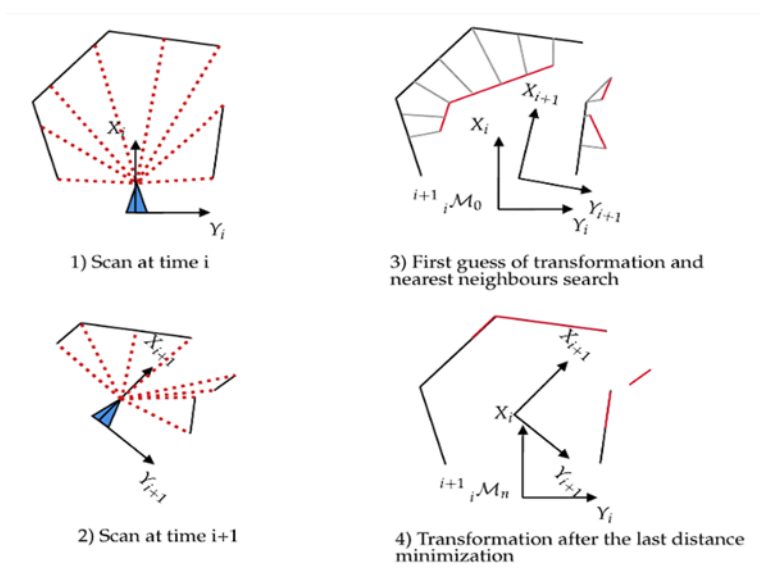
Κάθε κόμβος x_i του γράφου που απεικονίζεται στο σχήμα 1 αντιστοιχεί σε μια θέση του συστήματος. Οι κοντινές θέσεις συνδέονται με ακμές $E_{ij} = \{e_{ij}, \Omega_{ij}\}$ που αντιπροσωπεύουν τους χωρικούς περιορισμούς μεταξύ των θέσεων i και j του ρομπότ (από τις παρατηρήσεις). Σημειώνεται ότι οι ακμές e_{ij} μοντελοποιούν τις εσωτερικές μετρήσεις IMU, ενώ οι Ω_{ij} τις εξωτερικές [33].

Σύμφωνα με τα παραπάνω, προκύπτει πως οι σύγχρονοι αλγόριθμοι LiDAR-SLAM στην πλειοψηφία τους βασίζονται στην βελτιστοποίηση γράφου και αποτελούνται από τα εξής τρία τμήματα τα οποία θα αναλυθούν περαιτέρω παρακάτω: LiDAR Οδομετρία, Εντοπισμός Κλειστών Βροχών και Βελτιστοποίηση Γράφου.

3.2.1 LiDAR Οδομετρία

Αφορά αποκλειστικά την διαδικασία υπολογισμού θέσης και χαρτογράφησης του χώρου χωρίς την χρήση κάποιας μέθοδου βελτιστοποίησης. Αποτελεί μια απλουστευμένη υποκατηγορία του LiDAR-SLAM που επιλύει το πρόβλημα μόνο μέσω των άμεσων παρατηρήσεων του αισθητήρα LiDAR. Το πρόβλημα λοιπόν δύναται να επιλυθεί μέσω της αλληλοαναφοράς νέφους, δηλαδή μέσω του προσδιορισμού των διαδοχικών μετασχηματισμών στερεού σώματος T μεταξύ των νεφών σημείων του συστήματος.

Η διαδικασία της αλληλοαναφοράς νέφους είναι η βασική διαδικασία που χρησιμοποιείται για τη δημιουργία τρισδιάστατων χαρτών με χρήση συστημάτων LiDAR και παρέχει ακριβείς πληροφορίες σχετικά με την κίνηση. Στην βιβλιογραφία αναφέρονται διάφορες προσεγγίσεις για την συνένωση των νεφών σημείων με κυριότερες να αποτελούν οι αλγόριθμοι Iterative Closest Point (ICP) [36], Generalized Iterative Closest Point (G-ICP) [37], Normal Distribution Transform (NDT) [38], Polar Scan Matching (PSM) [39] και η μέθοδος matching LOAM μέσα από LOAM features που περιγράφεται στο [32]. Συχνότερα γίνεται χρήση του ICP (περιγράφεται στο σχήμα 2) του οποίου κύριο μειονέκτημα είναι η δαπανηρή αναζήτηση των αντιστοιχιών μεταξύ των σημείων. Ακόμη ένα μειονέκτημα αποτελεί η μεγάλη ευαισθησία στο σημείο εκκίνησης για την ελαχιστοποίηση (απαιτεί προσεγγιστικές τιμές). Στο ίδιο πλαίσιο κινείται και ο αλγόριθμος (G-ICP), ο οποίος αποτελεί παραλλαγή του ICP που λαμβάνει καλύτερα υπόψιν τις επίπεδες επιφάνειες στις σαρώσεις. Ανάλογα με τον αλγόριθμο που θα επιλεγθεί για την αλληλοαναφορά των σαρώσεων ενδέχεται να συμπεριλαμβάνονται και βήματα προεπεξεργασίας για τις μεθόδους που απαιτούν προσεγγιστικές τιμές, όπως είναι οι αλγόριθμοι ICP και NDT. Σε αυτές τις περιπτώσεις, οι προσεγγιστικές τιμές υπολογίζονται συχνότερα μέσω συνδυασμού μετρήσεων από IMU/INS αλλά και μέσω εύρεσης και αντιστοίχισης ομολογιών μέσω 3D περιγραφών (descriptors) για νέφη σημείων [32] [40] [41]. Τέλος, άλλες όχι τόσο ακριβείς προσεγγίσεις αφορούν παραδοχές για το μοντέλο κίνησης μεταξύ δυο διαδοχικών μετρήσεων, όπως για παράδειγμα το μοντέλο σταθερής ταχύτητας (Constant Velocity Model), και συνήθως συνδυάζονται με μεθόδους φιλτραρίσματος. Εντούτοις, απαιτούν προσοχή καθώς το μοντέλο δεν περιγράφει την πραγματικότητα. Γενικά η βελτιστοποίηση της ακρίβειας και της ταχύτητας της συνένωσης των νεφών σημείων αποτελεί από μόνη της ένα ξεχωριστό πεδίο έρευνας με αποτέλεσμα τα τελευταία χρόνια να βρίσκονται νέες προσεγγίσεις στο πεδίο της αλληλοαναφοράς νεφών .



Σχήμα 2: Συνένωση νεφών σημείων με βάση τον ICP [33]

3.2.2 Βελτιστοποίηση Γράφου

Προκειμένου να μειωθούν τα τοπικά σφάλματα που προκύπτουν από την Οδομετρία =LiDAR, μπορούν να χρησιμοποιηθούν μέθοδοι βελτιστοποίησης βασισμένες σε Γράφους [42]. Με αυτήν την μέθοδο, το σύνολο των θέσεων (poses) της οντότητας αναπαρίσταται από ένα γράφο, στον οποίο κάθε κόμβος αντιπροσωπεύει μια θέση του αισθητήρα, ενώ οι ακμές αντιπροσωπεύουν τον περιορισμό που δημιουργείται από τις παρατηρήσεις (μετασχηματισμός στερεού σώματος) που προέρχονται από την αλληλοαναφορά σαρώσεων. Στη συνέχεια, η βελτιστοποίηση του γράφου στοχεύει στην συνολική βελτιστοποίηση (global optimization) του συστήματος συνδυάζοντας τους περιορισμούς θέσης και κίνησης του αισθητήρα. Τέλος αυτοί οι γράφοι επιλύονται μέσω αλγόριθμων επίλυσης μη γραμμικών συστημάτων όπως είναι ο Levenberg–Marquardt [43].

3.2.3 Εντοπισμός Κλειστών Βροχών

Τα προηγούμενα δυο βήματα παρέχουν ως αποτέλεσμα την πλήρη ταυτόχρονη εύρεση της θέσης και την χαρτογράφηση κυρίως όμως μέσω Οδομετρίας. Γενικά, για να ανταποκριθούμε πλήρως στο πρόβλημα του LiDAR-SLAM, οφείλει να προστεθεί κάποια μέθοδος κλεισίματος βρόχου (loop closing). Το κλείσιμο βρόχου αφορά την δυνατότητα του αλγόριθμου να αναγνωρίζει περιοχές που έχει ξανά επισκεφτεί το σύστημα. Με αυτόν τον τρόπο μπορεί να διορθώνει το σφάλμα που έχει συσσωρευθεί κατά την διάρκεια των διαδοχικών μετασχηματισμών, που οδηγεί σε λανθασμένες αποκλίσεις, ειδικά των στροφών (drift). Αυτό επιτυγχάνεται μέσω βελτιστοποίησης του μετασχηματισμού στον οποίο εντοπίστηκε το κλείσιμο, ανάγοντας τον με βάση το αρχικό πέρασμα στο οποίο είχε συσσωρευθεί μικρότερο σφάλμα λόγω των λιγότερων μετασχηματισμών που είχαν προηγηθεί. Έτσι μέσω αυτής της βελτιστοποίησης των μετρήσεων μεταξύ του βρόχου, βελτιώνεται σημαντικά ο συνολικός χάρτης και η τροχιά που ακολουθήθηκε.

Ο εντοπισμός βρόχων επιτυγχάνεται στην πλειοψηφία των περιπτώσεων μέσω 3D γεωμετρικών περιγραφών (descriptors) για νέφη σημείων [44]. Αυτοί οι περιγραφείς χρησιμοποιούνται για να πραγματοποιηθεί συνένωση (matching) μεταξύ της τρέχουσας σάρωσης με τις προηγούμενες, για την ανίχνευση πιθανού βρόχου. Η ανίχνευση πιθανού υποψήφιου βρόχου γίνεται κατά βάση μέσω σύγκρισης περιγραφών σε συνδυασμό με έλεγχο του σφάλματος ενδεχομένης συνένωσης (αλληλοαναφοράς). Γίνεται εύκολα αντιληπτό βέβαια ότι μια τέτοια διαδικασία επαναλαμβανόμενης συγκεντρωτικής σύγκρισης δύσκολα θα μπορούσε να υλοποιηθεί σε πραγματικό χρόνο. Έχει επικρατήσει, λοιπόν, για πρακτικούς λόγους η χρήση υποχάρτων (submaps), δηλαδή μικρών συστάδων που αποτελούνται από περισσότερες της μίας σαρώσεις, για την αναζήτηση υποψήφιων κλεισίματων βρόχου (loop closures) [45]. Τελικά ο αλγόριθμος εκτελεί την ανίχνευση βρόχου σε υποχάρτες (submaps) που βρίσκονται σε ένα συγκεκριμένο εύρος γύρω από την τρέχουσα θέση ελέγχου.

Τέλος, μια ακόμη αρκετά διαδεδομένη διαδικασία ανίχνευσης βρόχου βασίζεται σε αναπαράσταση των 3D νεφών μέσω του μετασχηματισμού NDT (Normalized Distributions Transform) καθώς και σε ιστογράμματα χαρακτηριστικών που περιγράφουν επιφάνειες [46].

Στην προσέγγιση της διαδικασίας LiDAR-SLAM μέσω φίλτρων (filter-based LiDAR-SLAM), εξακολουθεί να υφίσταται η λογική της Οδομετρίας, ωστόσο αντί για βελτιστοποίηση γράφου χρησιμοποιούνται τεχνικές φιλτραρίσματος. Οι τεχνικές αυτές αποτελούν αλγορίθμους βέλτιστης εκτίμησης με σκοπό τον καθαρισμό των μετρήσεων από θόρυβο, και τελικά, τη δημιουργία μιας νέας βέλτιστης κατάστασης του συστήματος η οποία συνδυάζει όλες τις γνωστές πληροφορίες. Η βέλτιστη κατάσταση αντιστοιχεί σε αυτήν που εμφανίζει την μεγαλύτερη πιθανότητα και την μικρότερη μεταβλητότητα με βάση τις παρατηρήσεις για το συγκεκριμένο μαθηματικό μοντέλο. Οι μέθοδοι αυτές συνήθως είναι υπολογιστικά πιο απαιτητικές σε 3D εφαρμογές όπου συγκεντρώνεται μεγάλος αριθμός σημείων, ενώ παράλληλα σημαντικό μειονέκτημα αποτελεί η δυσκολία υλοποίησης αναζήτησης κλειστών βρόχων με αποτέλεσμα σε μεγαλύτερα σύνολα δεδομένων να υστερούν σε ακρίβεια και ταχύτητα. Ωστόσο, εμφανίζεται ξανά η τάση χρήσης φίλτρων σε συνδυασμό με τεχνικές κλεισίματος βρόχων καθώς αυτά διευκολύνουν την αυτοματοποιημένη συγχώνευση δεδομένων από διαφορετικούς αισθητήρες. Μια τέτοια περίπτωση αποτελεί η μέθοδος EKF (Extended Kalman Filter), η οποία είναι ιδανική για την αντιμετώπιση προβλημάτων συγχώνευσης δεδομένων από πολλαπλούς αισθητήρες, χάρη στα χαρακτηριστικά που προσφέρει [47]. Επιπλέον, η χρήση στατιστικών μέτρων αβεβαιότητας επιτρέπει την ποσοτική αξιολόγηση του ρόλου κάθε αισθητήρα στη συνολική απόδοση του συστήματος.

Ένα ακόμη θετικό χαρακτηριστικό αποτελεί το γεγονός ότι ο συνδυασμός των αισθητήρων βελτιώνει το αποτέλεσμα σε δυσχερείς καταστάσεις όπου τα δεδομένα ενός εκ των αισθητήρων οι οποίοι συμμετέχουν στην μέθοδο δεν συγκεντρώνουν επαρκή δεδομένα, με αποτέλεσμα να αποτρέπει τυχόν σημαντική απώλεια στην ακρίβεια ή ακόμη και αστοχία της μεθόδου σε δύσκολες συνθήκες [23]. Παρακάτω, στον Πίνακα 1 εμφανίζονται κάποιοι από τους πιο διαδεδομένους σύγχρονους αλγορίθμους 3D LiDAR-SLAM ανάλογα με την βασική λογική που ακολουθεί η μέθοδος (Filter/Graph based).

Πίνακας 1: Σύγχρονοι αλγόριθμοι LiDAR-SLAM [23]

System	Sensor	Framework	Source
LeGO-LOAM	3D LiDAR and IMU	Graph optimization	(T. Shan and B. Englot, 2018)
SuMa	3D LiDAR	Graph optimization	(J. Behley and C. Stachniss, 2018)
hdl-graph-slam	3D LiDAR	Graph optimization	(K. Koide, <i>et al.</i> , 2019)
LINS	3D LiDAR and IMU	Filter	(C. Qin, <i>et al.</i> , 2020)
LIO-SAM	3D LiDAR and IMU	Graph optimization	(T. Shan, <i>et al.</i> , 2020)
FAST-LIO	3D LiDAR and IMU	Filter	(W. Xu and F. Zhang, 2021)
BALM	3D LiDAR	Graph optimization	(Z. Liu and F. Zhang, 2021)

F-LOAM	3D LiDAR	Graph optimization	(H. Wang, <i>et al.</i> , 2021)
E-LOAM	3D LiDAR	Graph optimization	(H. Guo, <i>et al.</i> , 2022)
D-LIOM	3D LiDAR and IMU	Graph optimization	(Z. Wang, <i>et al.</i> , 2022)
ART-SLAM	3D LiDAR and IMU	Graph optimization	(M. Frosi and M. Matteucci, 2022)
LOCUS 2.0	3D LiDAR and IMU	Graph optimization	(A. Reinke, <i>et al.</i> , 2022)
FAST-LIO2	3D LiDAR and IMU	Filter	(W. Xu, <i>et al.</i> , 2022)
Faster-LIO	3D LiDAR and IMU	Filter	(C. Bai, <i>et al.</i> , 2022)
EKF-LOAM	3D LiDAR and IMU	Filter	(G.P.C. Junior, <i>et al.</i> , 2022)
VoxelMap	3D LiDAR and IMU	Filter	(C. Yuan, <i>et al.</i> , 2022)
Point-LIO	3D LiDAR and IMU	Filter	(D. He, <i>et al.</i> , 2023)
Inv-LIO1	3D LiDAR and IMU	Filter	(P. Shi, <i>et al.</i> , 2023)

3.3 Τεχνολογίες αιχμής στη μέθοδο SLAM

3.3.1 Μέθοδοι φιλτραρίσματος (Filter-based)

Ένας από τους πρώτους filter-based σύγχρονους αλγόριθμους LiDAR-SLAM αποτελεί ο LINS [48] ο οποίος πραγματοποιεί εκτίμηση κατάστασης σε πραγματικό χρόνο μέσω συγχώνευσης μετρήσεων LiDAR και IMU/INS. Το κύριο πλεονέκτημα που παρουσιάζει ο συγκεκριμένος αλγόριθμος είναι η δυνατότητα σταθερής και αποτελεσματικής πλοήγησης σε δύσκολα περιβάλλοντα, στα οποία δεν μπορούν να εξαχθούν με ευκολία 3D γεωμετρικά χαρακτηριστικά από τις σαρώσεις. Σε αυτό το σύστημα αναπτύχθηκε μια παραλλαγή του φίλτρου Kalman, το iterated error-state Kalman Filter (iESKF) για την επαναλαμβανόμενη διόρθωση της εκτίμησης κατάστασης μέσω της δημιουργίας νέων συνταυτίσεων χαρακτηριστικών σε κάθε επανάληψη, διατηρώντας παράλληλα χαμηλό υπολογιστικό φόρτο. Από την άλλη πλευρά, παρόλο που αυτή η μέθοδος έχει υψηλή ακρίβεια και υψηλή αποδοτικότητα, αντιμετωπίζει προβλήματα στροφών (drift) κατά τη διάρκεια πιο μακροσκελών εφαρμογών. Πάνω σε αντίστοιχο θεωρητικό πλαίσιο στηρίχθηκαν οι σύγχρονοι αλγόριθμοι Fast-LIO [49] και η εξέλιξή του Fast-LIO2 [50] με κύρια διαφορά να αποτελεί μια παραλλαγή στο φίλτρο iESKF η οποία βελτιώνει περαιτέρω το αποτέλεσμα. Λίγο αργότερα, παρουσιάστηκε ο αλγόριθμος Faster-LIO [51] βασισμένος στο FAST-LIO2 που βελτίωσε την αποδοτικότητα και διατήρησε την ακρίβεια. Αυτό επιτεύχθηκε μέσω χρήσης μιας νέας δομής δεδομένων, της iVox (incremental Voxels) για τη διατήρηση του τοπικού χάρτη, μειώνοντας αποτελεσματικά τον χρόνο συνένωσης (registration) των νεφών σημείων χωρίς όμως να επηρεάζεται η ακρίβεια. Επιπλέον, αν και δεν αναφέρεται στην βιβλιογραφία, κατά την διάρκεια των δοκιμών που πραγματοποιήθηκαν στο πλαίσιο της παρούσας διπλωματικής εργασίας, ο αλγόριθμος Faster LIO φάνηκε να αποδίδει καλύτερα σε κλειστούς χώρους με λιγότερα χαρακτηριστικά. Την ίδια περίοδο κυκλοφόρησε και ο αλγόριθμος EKF LOAM [47]. Βασική του καινοτομία αποτελεί η χρήση παρατηρήσεων από rotary encoders πέρα από τον αισθητήρα LiDAR και το IMU. Βέβαια ο συγκεκριμένος αλγόριθμος παρουσίασε αδυναμία επιλογής παραμέτρων αυτοματοποιημένα με βάση τα δεδομένα και το περιβάλλον εφαρμογής, γεγονός

που περιορίζει αρκετά την πρακτικότητα χρήσης, ενώ και η ανάγκη rotary encoder περιορίζει ακόμη παραπάνω την χρησιμότητα σε κάποιες εφαρμογές. Ακόμη την ίδια περίοδο κυκλοφόρησε ο VoxelMap [52] ακόμη ένας 3D LiDAR-SLAM αλγόριθμος που βασίζεται σε μια αποδοτική, με εξαιρετική προσαρμοστικότητα σε ρομποτικές εφαρμογές, πιθανοτική μέθοδο χαρτογράφησης μέσω voxels. Ο χάρτης αποτελείται από voxels, καθένα από τα οποία περιέχει ένα μοναδικό χαρακτηριστικό επίπεδο που επιτρέπει την πιθανοτική αναπαράσταση του περιβάλλοντος και οδηγεί σε ακριβή συνένωση νεφών σημείων σε μικρό χρόνο. Στη συνέχεια, ακολουθήσε προσθήκη υποστήριξης νέων 3D γεωμετρικών χαρακτηριστικών όπως οι ακμές, για τη διεύρυνση της αποτελεσματικότητας της προτεινόμενης μεθόδου αναπαράστασης χάρτη σε περισσότερα σενάρια. Ένα χρόνο μετά (2023) κυκλοφόρησε ο Point-LIO [53] μια προσέγγιση που βασίζεται σε LiDAR/IMU και παρουσίασε ως καινοτομία την ικανότητα να ανταπεξέρχεται σε περιπτώσεις πολύ έντονων και απότομων μετακινήσεων και στροφών που θα μπορούσαν να προκύψουν σε handheld και ρομποτικές εφαρμογές. Τέλος, ο πιο σύγχρονος filter-based αλγόριθμος Inv-LIO [54], προορίζεται κυρίως για ρομποτικές εφαρμογές που ενώ συνδυάζει μετρήσεις LiDAR και IMU χρησιμοποιώντας τη θεωρία των ομάδων Lie [55].

3.3.2 Μέθοδοι Γράφων (Graph-based)

Κατά αντιστοιχία με τους αλγόριθμους filter-based, οι πιο σύγχρονοι graph-based αλγόριθμοι περιλαμβάνουν τον LeGO-LOAM [56] ο οποίος χρησιμοποιεί εξαγωγή χαρακτηριστικών ιδανικών για να επιτευχθεί με ακρίβεια ο G-ICP που χρησιμοποιεί η μέθοδος SUMA [57]. Συγκεκριμένα, υλοποιεί χαρτογράφηση με βάση τα surfel (αντίστοιχα των voxel) για τον υπολογισμό κατάστασης (θέσης και στροφών) εκμεταλλευόμενος την προβολική συσχέτιση μεταξύ της παρούσας σάρωσης και μιας ανακατασκευασμένης προβολής από τον surfel-based χάρτη. Δυο χρόνια μετά, (2020), έχοντας ως βάση τον LeGO-LOAM, αναπτύχθηκε ο αλγόριθμος LIO-SAM [58] που συνδυάζει αναπόσπαστα μετρήσεις LiDAR με IMU ως δεσμεύσεις μεταξύ των κόμβων. Στη συνέχεια προστέθηκε η δυνατότητα εισαγωγής παραπάνω περιορισμών στον γράφο από διαθέσιμες παρατηρήσεις GNSS, οδόμετρου και φυσικά περιορισμών που προκύπτουν από τυχόν εντοπισμούς βρόχων για την τελική συνδυαστική βελτιστοποίηση συνολικά για το σύστημα χάρτη - θέσεων. Ακόμη ένας αλγόριθμος που έκανε την εμφάνισή του την ίδια περίοδο αποτελεί ο BALM [59] ο οποίος προτείνει μια νέα ευρηματική προσέγγιση εφαρμογής δεσμεύσεων πριν την συνένωση των νεφών. Αφορά πρακτικά την υλοποίηση μιας LiDAR μεθόδου δέσμης (LiDAR-BA) η οποία προσπαθεί να θέσει περιορισμούς στα χαρακτηριστικά που θα χρησιμοποιηθούν για την επίλυση. Στη συνέχεια παρουσιάστηκε ο F-LOAM [60] με κύρια καινοτομία την εισαγωγή μιας περαιτέρω διαδικασίας απλοποίησης και αποσφαλμάτωσης ως προεπεξεργασία, γεγονός το οποίο είχε θετική επίδραση στο υπολογιστικό κόστος της βελτιστοποίησης. Σε παρόμοια λογική παρουσιάστηκε ο αλγόριθμος E-LOAM (LOAM with Expanded Local Structural Information) [61] όπου, όπως δηλώνει και η ονομασία του, προσπάθησε να ενισχύσει τα εξαγόμενα χαρακτηριστικά που θα χρησιμοποιηθούν στην οδομετρία και την εύρεση πιθανών

κλειστών βρόχων. Ωστόσο αυτό επέφερε μεικτά αποτελέσματα καθώς σε αραιά περιβάλλοντα βελτίωσε την διαδικασία αλλά στην γενική περίπτωση προκύπτουν επικαλύψεις μεταξύ των ενισχυμένων αυτών χαρακτηριστικών, με αποτέλεσμα τελικά να μειώνεται η ακρίβεια και η υπολογιστική αποδοτικότητα της μεθόδου. Την ίδια χρονιά (2022), σε λίγο διαφορετική λογική, κινήθηκε ο D-LIOM [62] ο οποίος προσπάθησε να απλοποιήσει την διαδικασία με χρήση μετρήσεων LiDAR και IMU κατά τις οποίες δεν λάμβανε υπόψιν τις μετρήσεις κατά την αρχική οδομετρία αλλά μόνο στην συνολική τελική βελτιστοποίηση. Αυτό είχε ως αποτέλεσμα την αύξηση της ταχύτητας αλλά και την σχετικά μικρή μείωση της ακρίβειας λόγω της χρονικής διαφοράς που εμφάνιζαν οι μετρήσεις IMU και LiDAR. Ακόμη την ίδια χρονιά κυκλοφόρησαν οι αλγόριθμοι ART-SLAM [63] και LOCUS 2.0 [64] με τον ART-SLAM να βελτιώνει την ταχύτητα της διαδικασίας μέσω εφαρμογής προεπεξεργασίας στις μετρήσεις για την αφαίρεση κινουμένων αντικειμένων από τις σαρώσεις πριν την κύρια διαδικασία. Αντίστοιχα, ο LOCUS 2.0 σχεδιάστηκε κυρίως με πιο προσαρμοσμένες παραμέτρους σε Registration και Loop closing με στόχο την καλύτερη επίδοση σε υπόγεια ή και στενά περιβάλλοντα. Ένας ακόμα καινοτόμος αλγόριθμος που βασίζεται στην βελτιστοποίηση γράφου αποτελεί ο Suma++ [65], μια εξέλιξη του αρχικού SUMA που εφάρμοσε την χρήση ολοκληρωμένων συνελκτικών δικτύων για την σημασιολογική κατάτμηση του νέφους σημείων, οδηγώντας έτσι σε βελτίωση της γεωμετρίας και ακρίβειας μέσα από την απομάκρυνση κινουμένων ή προβληματικών οντοτήτων της σάρωσης. Ο συγκεκριμένος αλγόριθμος εντάσσεται στην κατηγορία Learning-based SLAM με χρήση μηχανικής μάθησης.

Η κατηγορία γίνεται ολοένα και πιο δημοφιλής, καθώς η χρήση μηχανικής μάθησης και τα νευρωνικά δίκτυα βελτιώνουν σημαντικά την ακρίβεια και την αξιοπιστία της οδομετρίας στο υπάρχον πλαίσιο του graph-based SLAM. Γεγονός που αν και δεν έχει αποτυπωθεί ακόμη πλήρως στην βιβλιογραφία, όλο και περισσότεροι αλγόριθμοι [66] τείνουν προς αυτή την κατεύθυνση (PIN-SLAM [67], Nerf LOAM [68] και Loner [69]).

3.3.3 Μέθοδος SLAM βασισμένη σε περιστροφικά LiDAR

Τέλος, άλλες αναφορές σε state-of-the-art αλγορίθμους LiDAR-SLAM αφορούν παραλλαγές που εξειδικεύονται ιδιαιτέρως σε περιστροφικά μηχανικά LiDAR λόγω των ιδιοτήτων του αισθητήρα (360 μοίρες οριζόντιο οπτικό πεδίο). Παρακάτω θα γίνει αναφορά στους αλγορίθμους αυτούς χωρίς διάκριση σε filter ή graph based.

Ένας βασικός αλγόριθμος βασισμένος σε περιστροφικά συστήματα LiDAR αποτελεί ο UMF [70], ο οποίος σχεδιάστηκε ώστε να μπορεί να λειτουργεί με κάθε τύπο αισθητήρα LiDAR (2D, 3D, 360°) βασιζόμενος στα φίλτρα Error State Kalman filter (ESKF) και Gaussian Particle Filter (GPF). Στην ίδια filtered-based λογική κινήθηκε την ίδια περίοδο και ο LOLA SLAM [71] που αποτελεί όμως και τον μόνο αλγόριθμο για περιστροφικά LiDAR που δεν χρησιμοποιεί συνδυασμό με παρατηρήσεις IMU. Λίγο αργότερα έκαναν την εμφάνιση τους και κάποιες αντίστοιχες graph based τεχνικές, ο R-LIO [72] ο Wildcat [73] και ο Map centric SLAM [74]. Στη συνέχεια, κυκλοφόρησε ο OMC-SLIO [75] ένας filter-based αλγόριθμος με κύρια καινοτομία την προσπάθεια για παράλληλη online βαθμονόμηση του μηχανισμού περιστροφής του LiDAR με βάση ταυτόχρονες παρατηρήσεις από IMU, μια χρονιά μετά, το 2023, υλοποιήθηκε ο SPIN-LOAM [76] που αποτελεί graph based υλοποίηση η οποία βασίζεται στον LOAM [32] και το 2D/3D LiDAR-SLAM Cartographer [45] ώστε να πετύχει καλύτερη ακρίβεια μέσω μιας προσαρμοστικής συνένωσης νεφών ανάλογα με την γεωμετρία των χαρακτηριστικών και τέλος, ο PULSAR [78] αποτέλεσε έναν εξειδικευμένο αλγόριθμο SLAM για την υλοποίηση σε περιστρεφόμενα LiDAR-UAV που θα μπορούσε να δώσει ακριβή αποτελέσματα ειδικά σε UAV εφαρμογές. Στον πίνακα 2 παρουσιάζονται συνοπτικά οι αλγόριθμοι περιστρεφόμενων συστημάτων LiDAR στους οποίους έγινε αναφορά παραπάνω.

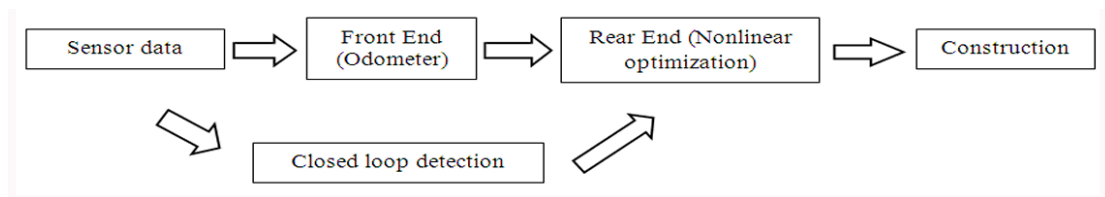
Πίνακας 2: Εξειδικευμένοι αλγόριθμοι LiDAR-SLAM για περιστρεφόμενους σαρωτές [23]

System	Sensor	Framework	Source
UMF	3D LiDAR and IMU	Filter	(W. Zhen and S. Scherer, 2020)
LoLa-SLAM	3D LiDAR	Filter	(M. Karimi, M. Oelsch, O. Stengel, E. Babaian and E. Steinbach, 2021)
R-LIO	3D LiDAR and IMU	Graph optimization	(K. Chen, K. Zhan, F. Pang, X. Yang and D. Zhang, 2022)
Wildcat	3D LiDAR and IMU	Graph optimization	(L. C. de Lima, M. Ramezani, P. Borges and M. Brunig, 2023)
Map-centric SLAM	3D LiDAR and IMU	Graph optimization	(C. Park, P. Moghadam, J. Williams, S. Kim, S. Sridharan and C. Fookes, 2022)
OMC-SLIO	3D LiDAR and IMU	Filter	(S. Wang, H. Zhang and G. Wang, 2022)
Spin-LOAM	3D LiDAR and IMU	Graph optimization	(L. Yan, J. Dai, Y. Zhao and C. Chen, 2023)
PULSAR	3D LiDAR and IMU	Filter	(N. Chen, F. Kong, W. Xu, Y. Cai, H. Li, D. He, Y. Qin and F. Zhang, 2023)

Ωστόσο αξίζει να σημειωθεί ότι οι κλασικοί σύγχρονοι αλγόριθμοι λειτουργούν και με περιστροφικά συστήματα LiDAR και για αυτόν τον λόγο αποτελούν και τις πιο δημοφιλείς μεθόδους, οι παραπάνω αλγόριθμοι γενικά βρίσκουν πιο εξειδικευμένες χρήσεις και δεν είναι τόσο ευρέως διαδεδομένοι. Επιπλέον και οι δυο κατηγορίες αλγόριθμων που αναφέρθηκαν ακολουθούν παρόμοιες αρχές λειτουργίας, συγκεκριμένα αυτές της συνένωσης νεφών σημείων και του εντοπισμού βρόχων. Η κύρια διαφορά κείται στην βελτιστοποίηση του συστήματος (filter based ή graph based), με επιπλέον διαφοροποιήσεις να προκύπτουν στους 3D περιγραφείς, τις προσεγγίσεις για τις συνενώσεις και τον εντοπισμό κλειστών βρόχων.

3.4 Δομή του σύγχρονου LiDAR SLAM

Στο προηγούμενο κεφάλαιο πραγματοποιήθηκε μια αναφορά στις βασικές αρχές λειτουργίας του σύγχρονου LiDAR-SLAM και στην δομή του, καθώς επίσης και μια κατηγοριοποίηση με βάση τον αισθητήρα και την μέθοδο βελτιστοποίησης. Σε αυτό την ενότητα θα παρουσιαστεί μία σύνοψη της διαδικασίας με σκοπό την αποσαφήνιση του τρόπου λειτουργίας των τεχνικών αυτών. Στο Σχήμα 3 παρουσιάζεται η πιο απλή αναπαράσταση μεθόδου SLAM.

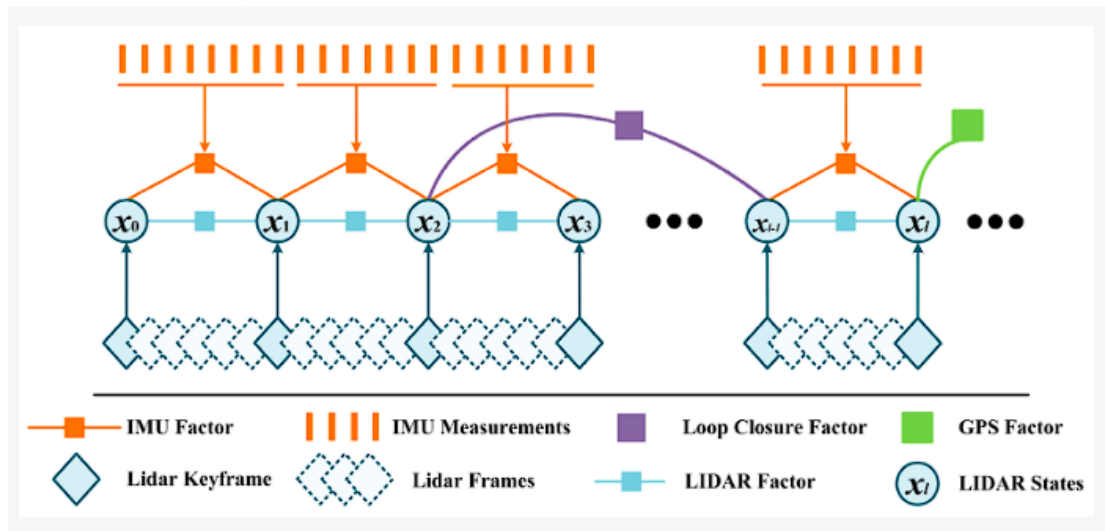


Σχήμα 3: Βασική Δομή Σύγχρονου SLAM [79]

Σύμφωνα με αυτή, παρατηρούμε ότι συνήθως η βιβλιογραφία διαχωρίζει την λειτουργία του αλγορίθμου σε front-end στην οποία τοποθετεί την οδομετρία, που στην περίπτωση του LiDAR-SLAM αφορά την συνένωση νεφών (scan registration/matching) και σε διαδικασία rear-end/back-end που περιλαμβάνει την τεχνική βελτιστοποίησης και την αναζήτηση κλεισίματος βρόχων. Πρακτικά η συγκεκριμένη δομή ακολουθείται στην πλειοψηφία των μεθόδων LiDAR-SLAM, με τις διαφοροποιήσεις να αφορούν είτε τις front-end ανάλογα με την προεπεξεργασία των μετρήσεων (εφαρμογή καθαρισμού ή δειγματοληψίας), και τις προσεγγίσεις για την συνένωση (μέσω παρατηρήσεων IMU ή εξαγωγής χαρακτηριστικών). Τέλος, το κύριο τμήμα της διαδικασίας front-end αποτελεί η μέθοδος που θα χρησιμοποιηθεί για την αλληλοαναφορά (εφαρμογή ICP, NDT, ή κάποια παραλλαγή του ICP).

Ακολουθως, οι διαφοροποιήσεις αυτές περνάνε στην rear-end διαδικασία και την εύρεση βρόχων, (η αναζήτηση θα γίνεται σε υποχάρτες ή στον συνολικό χάρτη ή σε κάποια βάση από features, ή θα γίνεται αναζήτηση μόνο γύρω από την τρέχουσα εκτίμηση) αλλά και στην μη γραμμική βελτιστοποίηση του συστήματος (θα χρησιμοποιηθεί κάποιο φίλτρο και αν ναι ποιο ή θα χρησιμοποιηθεί βελτιστοποίηση γράφου), πριν περάσουμε τελικά στην κατασκευή του χάρτη οπότε και εκεί μπορούν να υπάρχουν διαφοροποιήσεις στον τρόπο αποθήκευσης ή αναπαράστασης.

Απώτερος στόχος της σύγχρονης έρευνας είναι να βρεθεί ο βέλτιστος συνδυασμός αυτών των επιμέρους τμημάτων, ώστε να βελτιωθεί η ακρίβεια ελαχιστοποιώντας παράλληλα τον χρόνο και το υπολογιστικό κόστος που απαιτείται κατά την εφαρμογή. Τέλος, για την παραπάνω αποσαφήνιση της διαδικασίας ακολουθεί και μια καλύτερη επεξήγηση της μεθόδου του γράφου που υλοποιούν οι σύγχρονες εφαρμογές SLAM, με ένα παράδειγμα να απεικονίζεται στο σχήμα 4.



Σχήμα 4: Γράφος ενός συστήματος σύγχρονου LiDAR-SLAM [80]

Στο συγκεκριμένο παράδειγμα παρατηρούμε ότι οι κόμβοι αναπαριστούν τις θέσεις του αισθητήρα και ενώνονται από τις ακμές οι οποίες εκφράζουν τις δεσμεύσεις. Η συγκεκριμένη βελτιστοποίηση βλέπουμε πως λαμβάνει πάντα υπόψη τις διαθέσιμες παρατηρήσεις IMU και LiDAR και κάθε κόμβος ενώνεται με τον επόμενο με πορτοκαλί και γαλάζιες γραμμές οι οποίες αντιστοιχούν στις παρατηρήσεις αυτές. Υπάρχει επίσης δυνατότητα να προστεθούν παραπάνω περιορισμοί όταν το σύστημα έχει διαθέσιμες μετρήσεις GNSS ή έχει εντοπιστεί κλείσιμο κάποιου βρόχου. Τελικά το σύνολο των θέσεων βελτιστοποιείται με βάση το σύνολο των περιορισμών αυτών, ώστε να βρεθεί η βέλτιστη λύση σύμφωνα με το σύνολο των πληροφοριών το οποίο ήταν διαθέσιμο.

4. ΠΡΑΚΤΙΚΗ ΕΦΑΡΜΟΓΗ

Η πρακτική εφαρμογή αφορά την ανάλυση και αξιολόγηση σύγχρονων μεθόδων LiDAR-SLAM εστιάζοντας ιδιαίτερα στα πλεονεκτήματα και τις προκλήσεις που προκύπτουν στην χρήση τους. Ακόμη αφορά την υλοποίηση μεθόδων SLAM σε πραγματικά δεδομένα που λήφθηκαν στο πλαίσιο της παρούσας διπλωματικής, λαμβάνοντας υπόψη τη συμβατότητα τους με τον διαθέσιμο αισθητήρα LiDAR (LIVOX AVIA). Τέλος, πραγματοποιείται μια διερεύνηση της χρησιμότητας των επιλεγμένων τεχνικών.

Τα τελευταία χρόνια έχουν υλοποιηθεί δεκάδες τεχνικές LiDAR-SLAM. Στα σχετικά άρθρα υπάρχει κάθε θεωρητική λεπτομέρεια που αφορά την υλοποίηση της συγκεκριμένης μεθόδου, ενώ πάντα παρέχονται μαζί και συγκριτικά αποτελέσματα σε σχέση με άλλες αντίστοιχες τεχνικές. Για τα συγκριτικά αποτελέσματα, συνήθως αναφέρονται οι χρόνοι εκτέλεσης και τα σφάλματα που προέκυψαν σε σχέση με τα αληθή δεδομένα (ground-truth data) γνωστών συνόλων δεδομένων αξιολόγησης (evaluation dataset) που έχουν επικρατήσει στην επιστημονική κοινότητα του LiDAR-SLAM, όπως το KITTI [81], το KITTI 360 [82], το NCLT [83] και το EuRoC MAC [84]. Τα συγκεκριμένα datasets προσφέρουν πληθώρα διαφορετικών σεναρίων που μπορούν να αξιολογήσουν ικανοποιητικά τα αποτελέσματα των αλγορίθμων. Αντίστοιχα παρουσιάζεται και ένας ικανοποιητικός αριθμός από δημοσιεύσεις ανασκόπησης (review paper) που αναλύουν τα αποτελέσματα της ακρίβειας και ταχύτητας μεταξύ των διαφορετικών αλγορίθμων που κυκλοφορούν. Ωστόσο, παρατηρείται αρκετά συχνά να παραλείπονται οι τεχνικές λεπτομέρειες εφαρμογής των αλγορίθμων αυτών και κατά αντιστοιχίας των πειραμάτων που πραγματοποιήθηκαν.

Τέτοιες τεχνικές λεπτομέρειες για παράδειγμα αποτελούν οι ρυθμίσεις του αισθητήρα, η μορφή των δεδομένων που χρησιμοποιήθηκαν, η γλώσσα προγραμματισμού στην οποία υλοποιήθηκε ο κώδικας, οι απαραίτητες βιβλιοθήκες, η έκδοση του λειτουργικού συστήματος και οι αναλυτικές οδηγίες εγκατάστασης. Επιπλέον παρατηρήθηκε πολύ συχνά το φαινόμενο της πολύ περιορισμένης έως και μηδενικής υποστήριξης από τον προγραμματιστή, με αποτέλεσμα τυχόν σφάλματα (bugs) που εμφανίζονται στην πορεία να μην μπορούν εύκολα να επιλυθούν και ο κώδικας να καθίσταται πρακτικά ανεφάρμοστος. Έτσι, αρκετά συχνά, μη εξειδικευμένοι χρήστες ή χρήστες που δεν έχουν μεγάλη ευχέρεια στον προγραμματισμό να μην μπορούν εύκολα να καταπιαστούν με την διαδικασία του SLAM. Στο πλαίσιο της διπλωματικής λοιπόν πραγματοποιήθηκε η διερεύνηση της χρησιμότητας αυτής.

Σε αρχικό στάδιο έγινε αναζήτηση της γλώσσας προγραμματισμού και του λειτουργικού συστήματος στα οποία έχουν υλοποιηθεί οι περισσότερες μέθοδοι που αναφέρθηκαν κατά την βιβλιογραφική ανασκόπηση. Ιδιαίτερα χρήσιμη ήταν η ιστοσελίδα της KITTI [85] η οποία παρέχει διάφορες πληροφορίες για τους αλγορίθμους που έχουν χρησιμοποιήσει το dataset KITTI για αξιολόγηση. Ένα στιγμιότυπο της ιστοσελίδας αυτής εμφανίζεται στην εικόνα 3.











	Method	Setting	Code	Translation	Rotation	Runtime	Environment	Compare
1	SOFT2			0.53 %	0.0009 [deg/m]	0.1 s	4 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
<small>I. Cvijić, I. Marković and I. Petrović: SOFT2: Stereo Visual Odometry for Road Vehicles Based on a Point-to-Epipolar-Line Metric. IEEE Transactions on Robotics 2022. I. Cvijić, I. Marković and I. Petrović: Enhanced calibration of camera setups for high-performance visual odometry. Robotics and Autonomous Systems 2022. I. Cvijić, I. Marković and I. Petrović: Recalibrating the KITTI Dataset Camera Setup for Improved Odometry Accuracy. European Conference on Mobile Robots (ECMR) 2021.</small>								
2	V-LOAM			0.54 %	0.0013 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
<small>J. Zhang and S. Singh: Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast. IEEE International Conference on Robotics and Automation (ICRA) 2015.</small>								
3	LOAM			0.55 %	0.0013 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
<small>J. Zhang and S. Singh: LOAM: Lidar Odometry and Mapping in Real-time. Robotics: Science and Systems Conference (RSS) 2014.</small>								
4	TVL-SLAM+			0.56 %	0.0015 [deg/m]	0.3 s	1 core @ 3.0 Ghz (C/C++)	<input type="checkbox"/>
<small>C. Chou and C. Chou: Efficient and Accurate Tightly-Coupled Visual-Lidar SLAM. IEEE Transactions on Intelligent Transportation Systems 2021.</small>								
5	Traj-LIO			0.57 %	0.0015 [deg/m]	0.1 s	4 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
<small>X. Zheng and J. Zhu: Traj-LIO: A Resilient Multi-LiDAR Multi-IMU State Estimator Through Sparse Gaussian Process. 2024.</small>								
6	CT-ICP2		code	0.58 %	0.0012 [deg/m]	0.06 s	1 core @ 3.5 Ghz (C/C++)	<input type="checkbox"/>
<small>P. Dellenbach, J. Deschaud, B. Jacquet and F. Goulette: CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure. 2022 International Conference on Robotics and Automation (ICRA) 2022.</small>								
7	TBD			0.58 %	0.0014 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
8	Traj-LO		code	0.58 %	0.0014 [deg/m]	0.1 s	4 cores @ 3.5 Ghz (C/C++)	<input type="checkbox"/>
<small>X. Zheng and J. Zhu: Traj-LO: In Defense of LiDAR-Only Odometry Using an Effective Continuous-Time Trajectory. IEEE Robotics and Automation Letters 2024.</small>								
9	TBDM			0.59 %	0.0014 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
10	GLIM			0.59 %	0.0015 [deg/m]	0.1 s	GPU @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
<small>K. Koide, M. Yokozuka, S. Oishi and A. Banno: Globally Consistent 3D LiDAR Mapping with GPU-accelerated GICP Matching Cost Factors. IEEE Robotics and Automation Letters 2021.</small>								
11	ZRB-SLAM			0.59 %	0.0015 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
12	CT-ICP		code	0.59 %	0.0014 [deg/m]	0.06 s	1 core @ 3.5 Ghz (C/C++)	<input type="checkbox"/>
<small>P. Dellenbach, J. Deschaud, B. Jacquet and F. Goulette: CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure. 2022 International Conference on Robotics and Automation (ICRA) 2022.</small>								
13	W-ICP			0.59 %	0.0015 [deg/m]	0.00 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
14	SDV-LOAM		code	0.60 %	0.0015 [deg/m]	0.06 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
<small>Z. Yuan, Q. Wang, K. Cheng, T. Hao and X. Yang: SDV-LOAM: Semi-Direct Visual-LiDAR Odometry and Mapping. IEEE Transactions on Pattern Analysis and Machine Intelligence 2023.</small>								
15	KISS-ICP		code	0.61 %	0.0017 [deg/m]	0.05 s	1 core @ 4.5 Ghz (Python/C++)	<input type="checkbox"/>
<small>I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley and C. Stachniss: KISS-ICP: In Defense of Point-to-Point ICP -- Simple, Accurate, and Robust Registration If Done the Right Way. IEEE Robotics and Automation Letters (RA-L) 2023.</small>								

Εικόνα 3: Στιγμιότυπο από την σελίδα του benchmarking dataset KITTI [85]

Αρχικά καταγράφεται η σειρά κατάταξης του αλγορίθμου, ο αισθητήρας που χρησιμοποιήθηκε, δηλαδή αν πρόκειται για LiDAR, κάμερα ή στέρεο κάμερα, αλλά και περαιτέρω πληροφορίες όπως το link για τον κώδικα, το σφάλμα της μετάθεσης και των στροφών, ο χρόνος εκτέλεσης, οι προδιαγραφές του υπολογιστή στον οποίο πραγματοποιήθηκαν οι δοκιμές και η γλώσσα προγραμματισμού στην οποία υλοποιήθηκαν. Τέλος επισυνάπτονται τα σχετικά άρθρα.

Παρατηρήθηκε λοιπόν ότι από τους 15 καλύτερους αλγορίθμους σύμφωνα με το dataset KITTI οι 14 έχουν υλοποιηθεί σε C/C++, ενώ μόνο ένας, ο KISS-ICP [86] έχει υλοποιηθεί σε Python και C++.

Σε συνέχεια της αναζήτησης, προέκυψε πάλι όπως απεικονίζεται στην εικόνα 4 από την ίδια ιστοσελίδα ότι και οι υπόλοιποι αλγόριθμοι έχουν υλοποιηθεί σε C++ με εξαίρεση τον PIN-SLAM [67], ο οποίος δημοσιεύθηκε μόλις πρόσφατα και δυστυχώς αφού είχαν ολοκληρωθεί τα πειράματα της διπλωματικής εργασίας.

16	V-Slam			0.61 %	0.0017 [deg/m]	0.05 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
17	KISS test			0.62 %	0.0018 [deg/m]	0.05 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
18	MOLA-LO			0.62 %	0.0018 [deg/m]	0.05 s	4 cores @ 3.0 Ghz (C/C++)	<input type="checkbox"/>	
19	MOLA (default)			0.62 %	0.0019 [deg/m]	0.05 s	4 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
20	V-Test			0.63 %	0.0018 [deg/m]	0.1 s	GPU @ >3.5 Ghz (C/C++)	<input type="checkbox"/>	
21	p2mesh			0.64 %	0.0019 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
22	WICRFZ			0.64 %	0.0019 [deg/m]	0.07 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
23	PIN-SLAM		code	0.64 %	0.0015 [deg/m]	0.1 s	GPU @ >3.5 Ghz (Python)	<input type="checkbox"/>	
24	filter-reg			0.65 %	0.0016 [deg/m]	0.01 s	GPU @ 2.6 Ghz (C/C++)	<input type="checkbox"/>	
<small>X. Zheng and J. Zhu: ECTLO: Effective Continuous-Time Odometry Using Range Image for LiDAR with Small FoV. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023.</small>									
25	SOFT-SLAM			0.65 %	0.0014 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
<small>I. Cvbišić, J. Česić, I. Marković and I. Petrović: SOFT-SLAM: Computationally Efficient Stereo Visual SLAM for Autonomous UAVs. Journal of Field Robotics 2017.</small>									
26	V-VO-Test			0.65 %	0.0019 [deg/m]	0.05 s	GPU @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
27	MULLS		code	0.65 %	0.0019 [deg/m]	0.08 s	4 cores @ 2.2 Ghz (C/C++)	<input type="checkbox"/>	
<small>Y. Pan, P. Xiao, Y. He, Z. Shao and Z. Li: MULLS: Versatile LiDAR SLAM via Multi-metric Linear Least Square. IEEE International Conference on Robotics and Automation (ICRA) 2021.</small>									
28	ELO			0.68 %	0.0021 [deg/m]	0.005 s	GPU @ 2.6 Ghz (C/C++)(0.027s Jetson AGX)	<input type="checkbox"/>	
<small>X. Zheng and J. Zhu: Efficient LiDAR Odometry for Autonomous Driving. IEEE Robotics and Automation Letters(RA-L) 2021.</small>									
29	IMLS-SLAM			0.69 %	0.0018 [deg/m]	1.25 s	1 core @ >3.5 Ghz (C/C++)	<input type="checkbox"/>	
<small>J. Deschaud: IMLS-SLAM: Scan-to-Model Matching Based on 3D Data. 2018 IEEE International Conference on Robotics and Automation (ICRA) 2018.</small>									
30	MC2SLAM			0.69 %	0.0016 [deg/m]	0.1 s	4 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
<small>F. Neuhaus, T. Koss, R. Kohnen and D. Paulus: MC2SLAM: Real-Time Inertial LiDAR Odometry using Two-Scan Motion Compensation. German Conference on Pattern Recognition 2018.</small>									
31	TBD PGO			0.72 %	0.0017 [deg/m]	0.15 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	
32	ISC-LOAM		code	0.72 %	0.0022 [deg/m]	0.1 s	4 cores @ 3.0 Ghz (C/C++)	<input type="checkbox"/>	
<small>H. Wang, C. Wang and L. Xie: Intensity scan context: Coding intensity and geometry relations for loop closure detection. 2020 IEEE International Conference on Robotics and Automation (ICRA) 2020.</small>									
33	FLOAM		code	0.72 %	0.0022 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>	

Εικόνα 4: Στιγμιότυπο από την σελίδα του benchmarking dataset KITTI [85]

Γενικά η μελέτη των σύγχρονων αλγορίθμων LiDAR-SLAM της βιβλιογραφίας έδειξε ότι, με εξαίρεση τους KISS-ICP και PIN-SLAM οι οποίοι έχουν υλοποιηθεί στην γλώσσα προγραμματισμού Python, όλοι οι υπόλοιποι αλγόριθμοι έχουν αναπτυχθεί σε C/C++ γεγονός που περιορίζει την χρήση τους από το ευρύτερο επιστημονικό κοινό.

Σε επόμενο στάδιο πραγματοποιήθηκε αναζήτηση αλγορίθμων ανοικτού κώδικα μέσω της πλατφόρμας GitHub αλλά και στο ευρύτερο διαδίκτυο. Το GitHub [87] αποτελεί την μεγαλύτερη πλατφόρμα διαχείρισης κώδικα και παρέχει δωρεάν χώρο στον οποίο προγραμματιστές μπορούν να ανεβάζουν, να αποθηκεύουν και να διαχειρίζονται τους κώδικες τους με την κοινότητα. Η πλατφόρμα επιτρέπει επίσης στους προγραμματιστές να παρακολουθούν τις αλλαγές στον κώδικα τους, να συνεργάζονται με άλλους προγραμματιστές και να διαχειρίζονται τις διακλαδώσεις του κώδικα. Επιπλέον στην πλατφόρμα αυτή οι χρήστες δημιουργούν αποθετήρια (repositories) για τα έργα τους ώστε να τα μοιράζονται με την υπόλοιπη κοινότητα. Οι λοιποί χρήστες μπορούν με την σειρά τους να αναφέρουν προβλήματα (issues) που εντόπισαν κατά την εφαρμογή του κώδικα στον δημιουργό, στον οποίο δίνεται η δυνατότητα να τους απαντήσει.

Συγκεκριμένα η δυνατότητα συνεργατικής διαχείρισης του κώδικα αποτελεί ένα από τα βασικότερα χαρακτηριστικά του GitHub.

Ύστερα από την προσπάθεια αυτή εκτενούς αναζήτησης, δεν βρέθηκε κάποιος ολοκληρωμένος και δημοσιευμένος σύγχρονος αλγόριθμος LiDAR-SLAM σε άλλη γλώσσα πέρα της C++ (πέρα των λίγων εξαιρέσεων που αναφέρθηκαν σε Python). Ωστόσο εντοπίστηκαν διάφορες υλοποιήσεις αλγορίθμων και εργαλείων LiDAR-SLAM από την MathWorks [88]. Από το 2019 και μετά, η MathWorks προσπάθησε να υποστηρίξει διάφορες εφαρμογές πλοήγησης, αυτόνομης οδήγησης και ρομποτικής μέσω δημιουργίας των αντίστοιχων εργαλειοθηκών (toolboxes). Συγκεκριμένα κυκλοφόρησαν σταδιακά το LiDAR toolbox, το Navigation toolbox, το Automated driving toolbox, το Computer Vision toolbox και το ROS (Robot Operating System) toolbox. Οι συγκεκριμένες εργαλειοθήκες πρόσφεραν διάφορα εργαλεία και συναρτήσεις ώστε να μπορεί κανείς να υλοποιήσει τεχνικές SLAM μέσω του MATLAB. Στην εικόνα 5 εμφανίζονται κάποιες από τις διαθέσιμες αυτές συναρτήσεις.

LiDAR SLAM	
lidarSLAM	Perform localization and mapping using lidar scans (Since R2019b)
buildMap	Build occupancy map from lidar scans (Since R2019b)

Pose Graphs	
poseGraph	Create 2-D pose graph (Since R2019b)
poseGraph3D	Create 3-D pose graph (Since R2019b)
poseGraphSolverOptions	Solver options for pose graph optimization (Since R2020b)
optimizePoseGraph	Optimize nodes in pose graph (Since R2019b)
trimLoopClosures	Optimize pose graph and remove bad loop closures (Since R2020b)
poseplot	3-D pose plot (Since R2021b)

Factor Graph	
factorGraph	Bipartite graph of factors and nodes (Since R2022a)
factorGraphSolverOptions	Solver options for factor graph (Since R2022a)
importFactorGraph	Import factor graph from g2o log file (Since R2022a)
factorIMU	Convert IMU readings to factor (Since R2022a)

Εικόνα 5: Συναρτήσεις της γλώσσας προγραμματισμού MATLAB σχετικές με τα LiDAR-SLAM [88]

Εκτός από τα εργαλεία αυτά, η MathWorks υλοποίησε μερικά παραδείγματα μεθόδων SLAM (tutorials) χρησιμοποιώντας αυτές τις συναρτήσεις. Στο πλαίσιο της παρούσας εργασίας εξετάστηκαν και υλοποιήθηκαν τα συγκεκριμένα παραδείγματα ώστε να δημιουργηθεί μια σαφής εικόνα της διαδικασίας LiDAR-SLAM μέσω της γλώσσας προγραμματισμού MATLAB. Τα παραδείγματα αυτά αφορούν χρήση δεδομένων από σαρωτές LiDAR (2D και 3D) [89] που παρέχονται από την MathWorks.

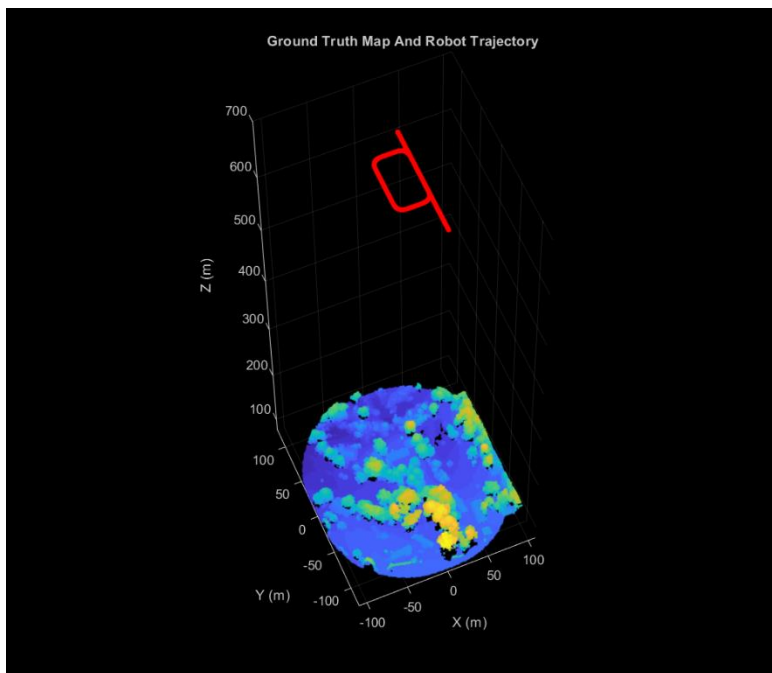
4.1 Παραδείγματα MATLAB

4.1.1 Aerial LiDAR SLAM Using FPFH Descriptors

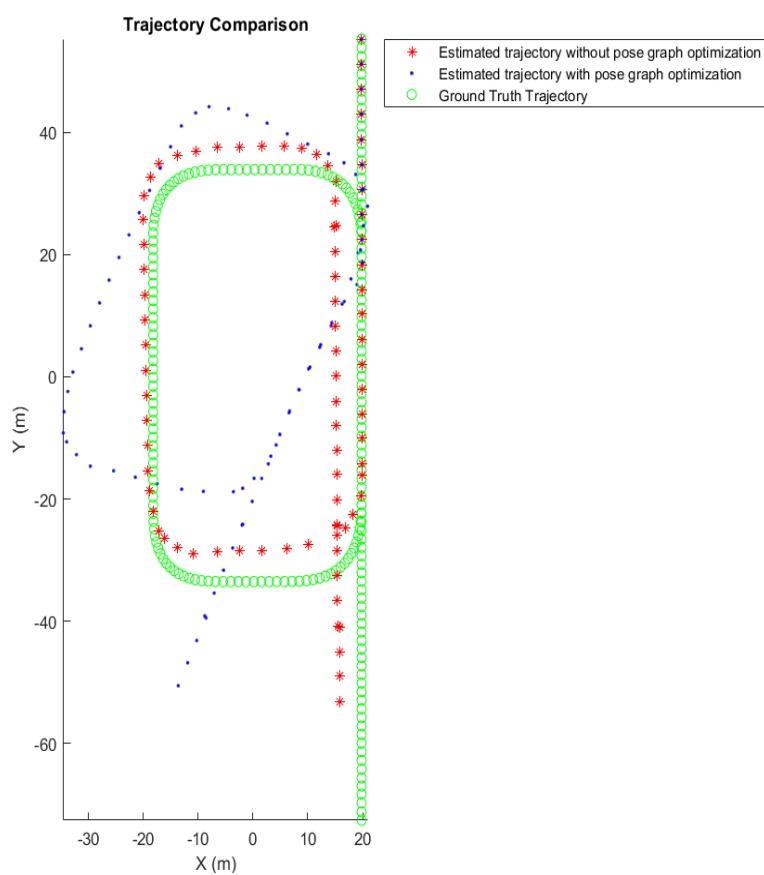
Το συγκεκριμένο παράδειγμα χρησιμοποιεί ως βασική είσοδο PointCloud objects, τα οποία αποτελούν δομή του MATLAB για την αποθήκευση νεφών σημείων διαφορετικών μορφών και τα οποία περιλαμβάνουν την θέση (X,Y,Z) των σημείων της κάθε σάρωσης. Τέλος, οι δομές αυτές μπορούν να αποθηκεύουν και άλλες πληροφορίες όταν αυτές είναι διαθέσιμες όπως είναι η ένταση του ανακλώντος σήματος (Intensity).

Το συγκεκριμένο παράδειγμα αφορά σαρώσεις από αισθητήρα τοποθετημένο σε UAV, ενώ αποτελεί graph-based υλοποίηση η οποία χρησιμοποιεί descriptors ως προσεγγίσεις για την αλληλοαναφορά και την αναζήτηση κλειστών βρόχων. Αναλυτικότερα, η λογική της επίλυσης χρησιμοποιεί εύρεση ομολογιών από κάθε νέφος σημείων μέσω FPHP descriptors [40] και εφαρμογή προσεγγιστικών μετασχηματισμών στερεού σώματος με βάση τις ομολογίες αυτές. Αφού πλέον οι διαδοχικές σαρώσεις έρθουν κοντά, το αποτέλεσμα βελτιώνεται μέσω χρήσης του αλγορίθμου Generalized ICP [37] (G-ICP-pcregistericp). Ακόμη περιλαμβάνει βελτιστοποίηση Γράφου Θέσης (Pose Graph Optimization) που βασίζεται στην συνάρτηση του MATLAB optimizePoseGraph, ενώ υποστηρίζει εύρεση κλειστών βρόχων μέσω σύγκρισης των σαρώσεων με προηγούμενα submaps μέσω του ICP. Τα submaps ελέγχονται διαδοχικά και αυτό το οποίο δίνει τελικά το μικρότερο σφάλμα, θεωρείται υποψήφιο για κλείσιμο βρόχου (loop closure). Εφόσον κάποια από τις σαρώσεις που το αποτελούν δώσουν σφάλμα κάτω από το κατώφλι που έχει οριστεί, ο βρόχος γίνεται αποδεκτός.

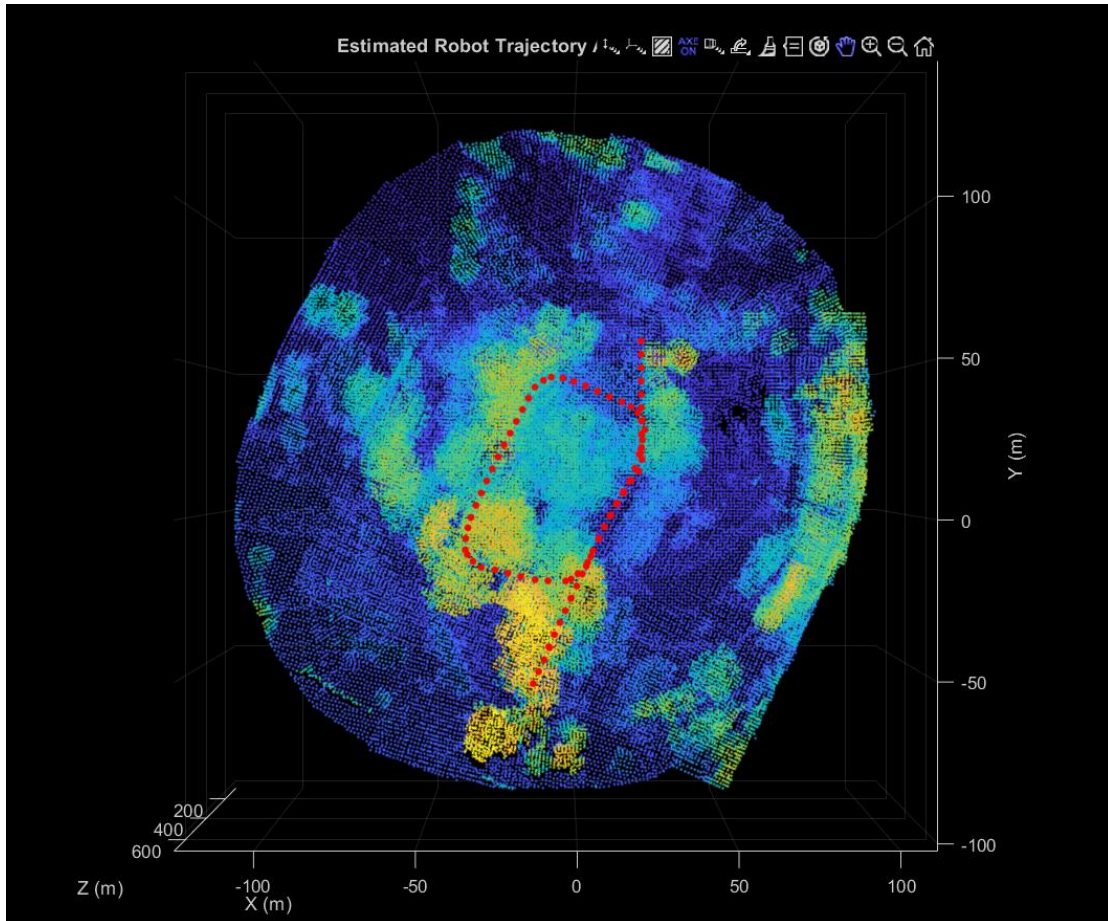
Αξίζει επίσης να αναφερθεί ότι τόσο στο συγκεκριμένο όσο και στα υπόλοιπα παραδείγματα που παρέχονται από την MathWorks, οι παράμετροι δεν προσαρμόζονται στα δεδομένα αλλά είναι σταθερές με αποτέλεσμα, ανάλογα με το dataset, μικρές αλλαγές στις παραμέτρους να δίνουν τελείως διαφορετικά αποτελέσματα. Στα Σχήματα 5-8 παρουσιάζονται συνοπτικά τα αποτελέσματα της μεθόδου.



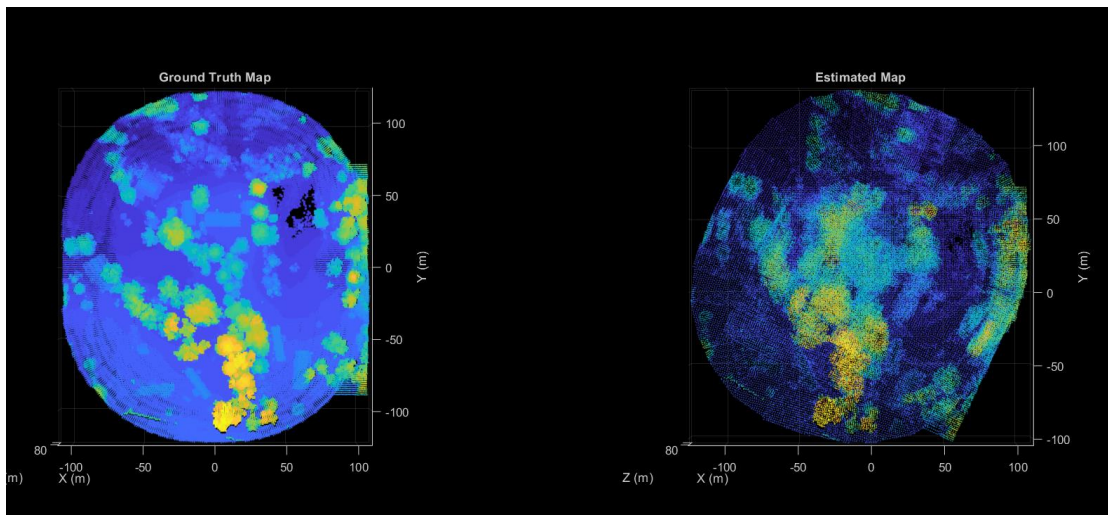
Σχήμα 5: Ground truth χάρτης και τροχιά - παράδειγμα 1 MATLAB



Σχήμα 6: Αποτελέσματα της τροχιάς - παράδειγμα 1 MATLAB



Σχήμα 7: Κάτοψη χάρτη και τροχιάς - παράδειγμα 1 MATLAB



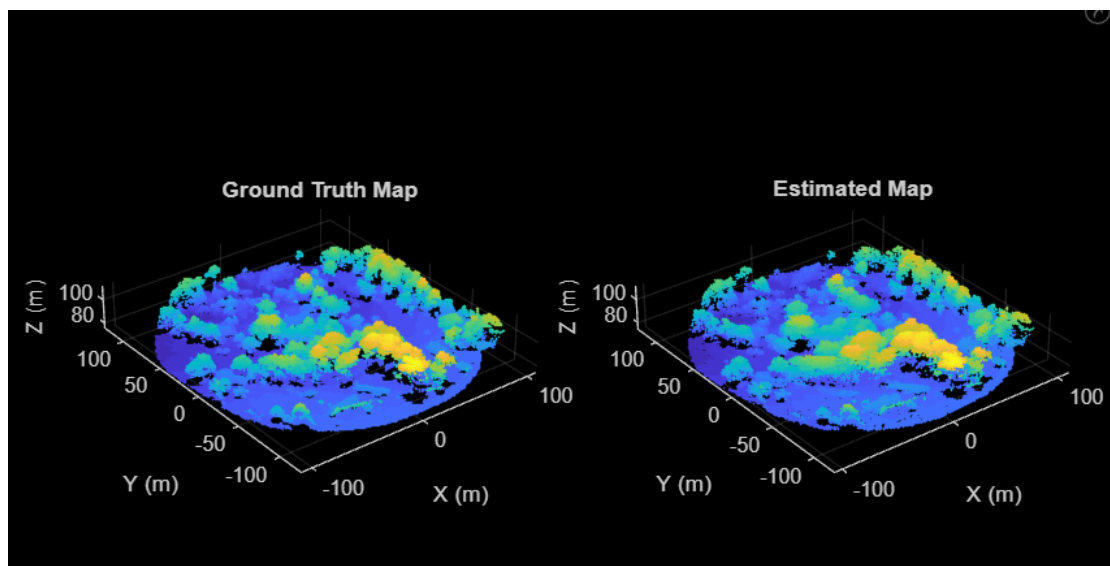
Σχήμα 8: Σύγκριση αποτελέσματος με Ground Truth – παράδειγμα 1 MATLAB

Από τα αποτελέσματα παρατηρείται ότι υπάρχει εμφανής απόκλιση (ολίσθηση) κατά την οδομετρία μέσω των descriptors η οποία αυξάνεται με την πάροδο του χρόνου. Ο αλγόριθμος εντόπισε κάποια κλεισίματα βρόχων καθώς πράγματι κατά την σάρωση το UAV επανέλαβε μια ευθύγραμμη πορεία, ενώ τέλος πραγματοποιήθηκε η βελτιστοποίηση του γράφου με το αποτέλεσμα ωστόσο να είναι αρκετά μακριά από την αληθή πορεία. Το αποτέλεσμα αυτό πιθανώς να

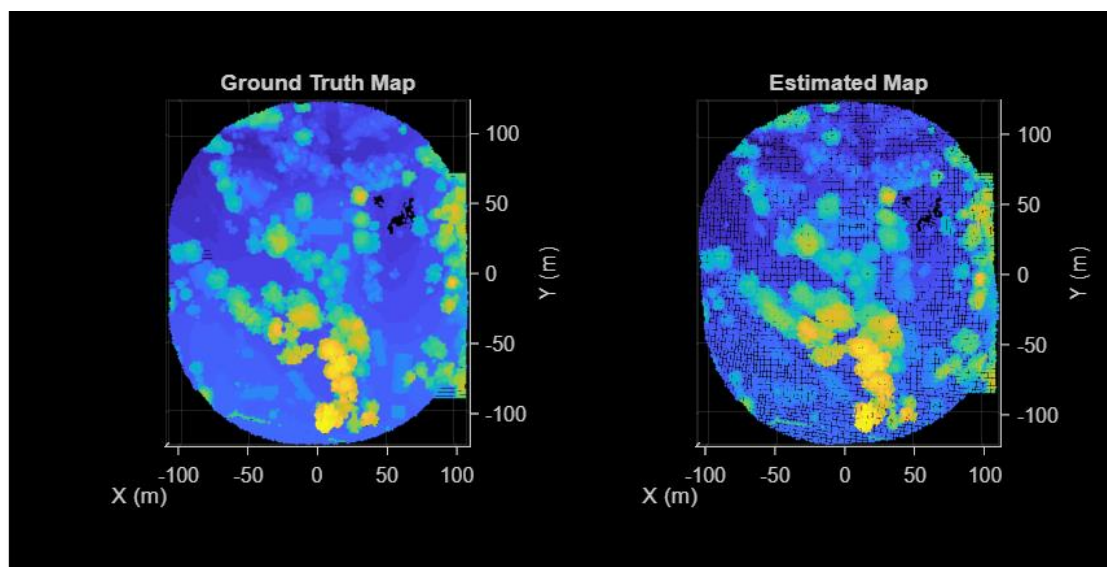
οφείλεται στις τιμές των παραμέτρων. Γι' αυτό το λόγο πραγματοποιήθηκαν κάποιες επιπρόσθετες δοκιμές με διαφοροποιημένες τιμές παραμέτρων. Επιπλέον σημειώνεται πως ο τελικός χάρτης είναι πιο αραιός από τον αληθή χάρτη (ground-truth) καθώς έχει πραγματοποιηθεί μείωση της ανάλυσης των νεφών σημείων ενώ παράλληλα έχει χρησιμοποιηθεί μόνο το 1/4 των διαθέσιμων σαρώσεων. Πραγματοποιήθηκε λοιπόν η δεύτερη δοκιμή ύστερα από αλλαγή της παραμέτρου skipFrames που καθορίζει το πλήθος των σαρώσεων που λαμβάνονται υπόψιν στην επίλυση. Τα αποτελέσματα της 2^{ης} δοκιμής παρουσιάζονται παρακάτω στα Σχήματα 9-12.

Κατά την δεύτερη δοκιμή έγινε αλλαγή της παραμέτρου skipFrames από την τιμή 3 σε 1.

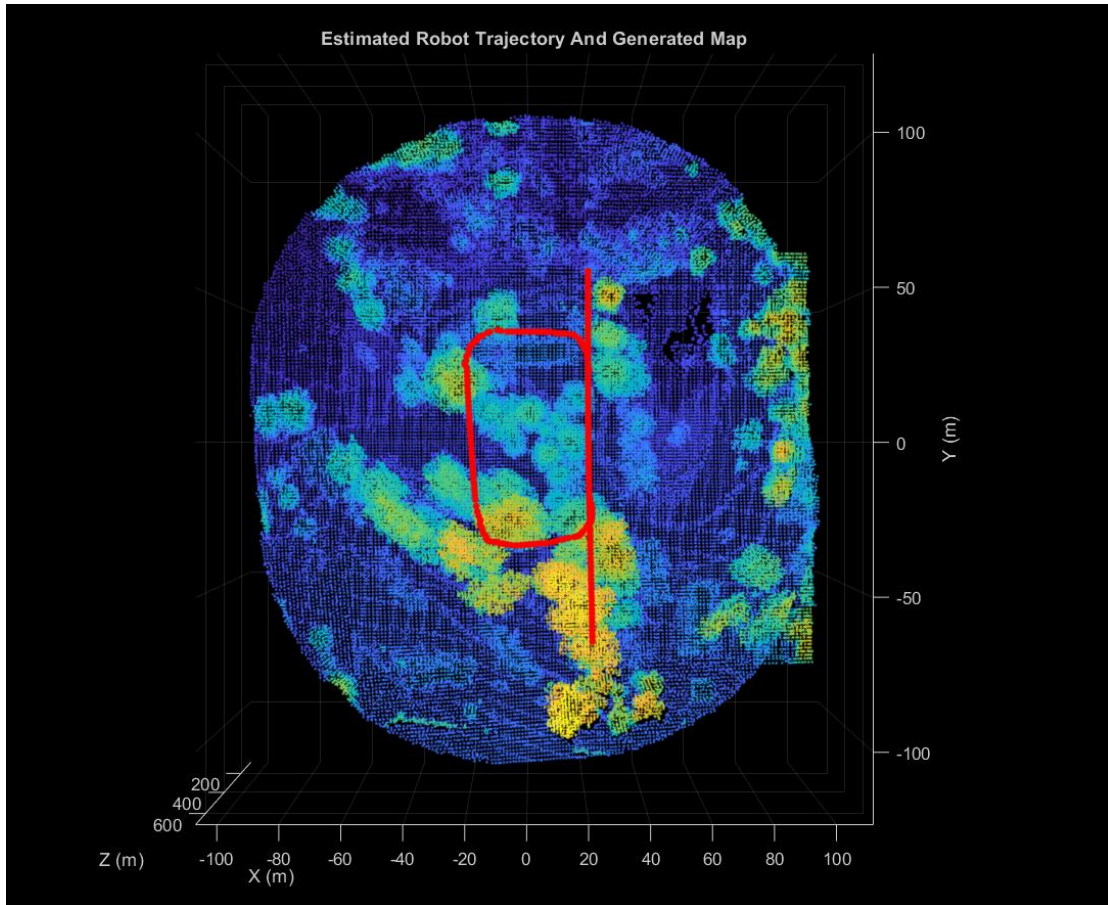
Με αυτήν την αλλαγή λαμβάνονται υπόψιν οι μισές σαρώσεις κατά την επίλυση.



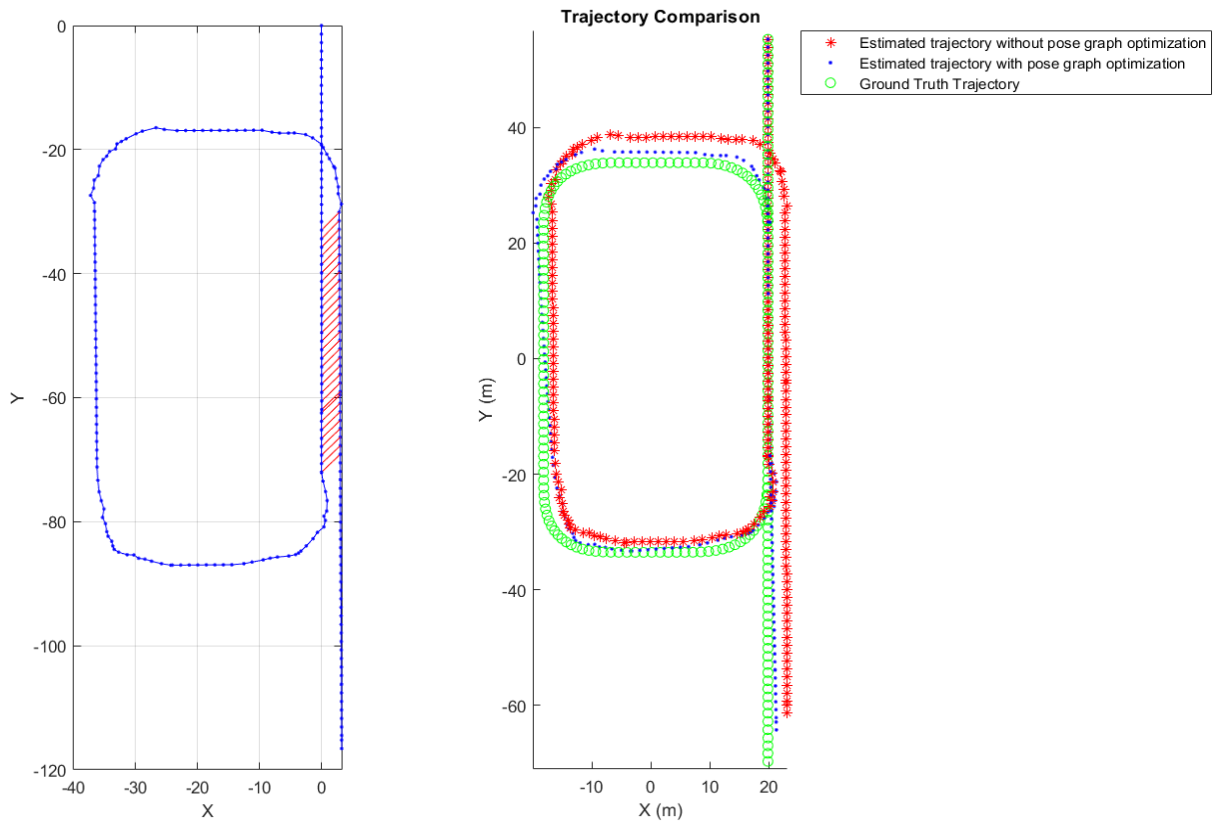
Σχήμα 9: Σύγκριση αποτελέσματος 2^{ης} δοκιμής με Ground Truth (i) – παράδειγμα 1 MATLAB



Σχήμα 10: Σύγκριση αποτελέσματος 2^{ης} δοκιμής με Ground Truth (ii) – παράδειγμα 1 MATLAB



Σχήμα 12: Κάτοψη χάρτη και τροχιάς 2^{ης} δοκιμής - παράδειγμα 1 MATLAB



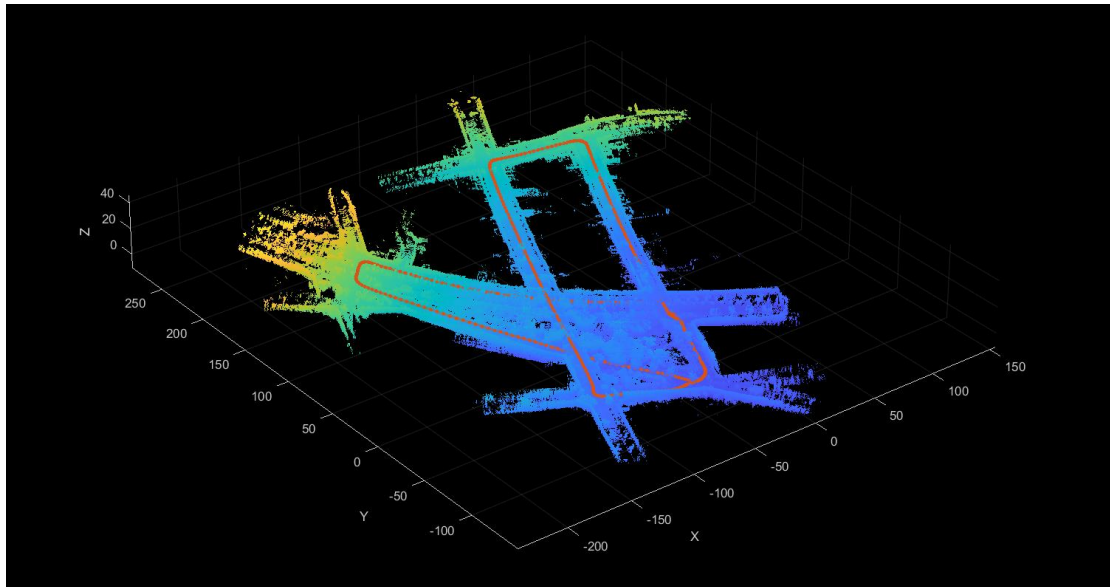
Σχήμα 11: Αποτελέσματα της τροχιάς 2^{ης} δοκιμής - παράδειγμα 1 MATLAB

Κατά την δεύτερη επίλυση παρατηρούμε ξανά πως από την οδομετρία και μόνο, προκύπτουν σταδιακά σφάλματα λόγω στροφών. Ωστόσο, η κύρια διαφορά που προέκυψε στην νέα δοκιμή με τις περισσότερες κοντινές σαρώσεις, είναι πως έχει προσδιοριστεί πολύ καλύτερα ο βρόχος που πραγματοποίησε το UAV, με την βελτιστοποίηση αυτή την φορά πράγματι να διορθώνει το αποτέλεσμα με βάση αυτήν την πληροφορία, πλησιάζοντας έτσι αισθητά τα αληθή (ground truth) αποτέλεσμα.

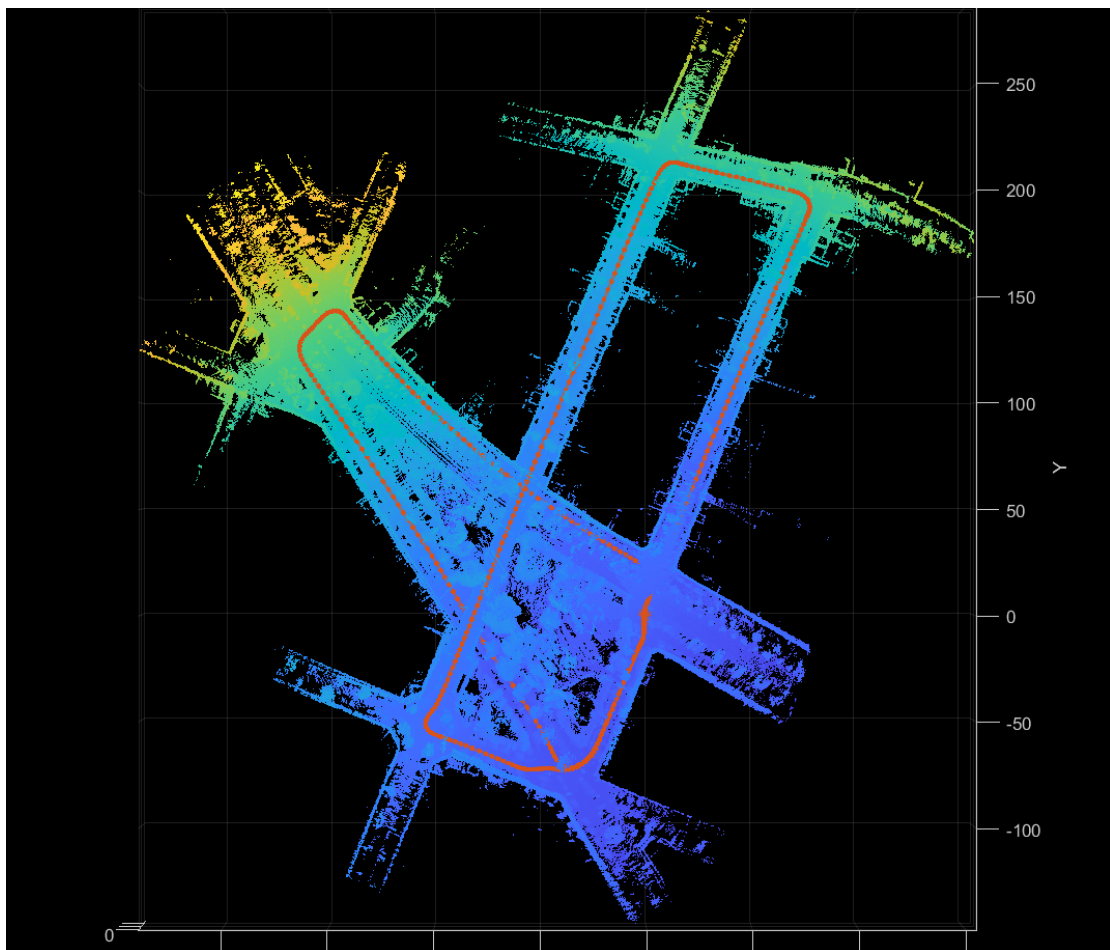
4.1.2 Build a map from LiDAR Data using SLAM

Η συγκεκριμένη υλοποίηση χρησιμοποιεί ως μεταβλητή εισόδου τύπου Point Cloud Objects και IMU μετρήσεις $|\text{Heading}|$, $|\text{Pitch}|$, $|\text{Roll}|$, $|X|$, $|Y|$, και $|Z|$ για κάθε νέφος σημείων ανάλογα με την χρονική στιγμή καταγραφής (Timestamp). Η μέθοδος χρησιμοποιεί τον αλγόριθμο NDT [38] για τη συνένωση των διαδοχικών νεφών σημείων αφού πρώτα έχει υπολογιστεί ένας προσεγγιστικός μετασχηματισμός μέσω των διαδοχικών μετρήσεων του IMU. Κατά την προεπεξεργασία των δεδομένων, αφαιρούνται τα σημεία του επιπέδου του εδάφους καθώς και του οχήματος που φέρει το Lidar. Το τελικό αποτέλεσμα παρέχει ένα χάρτη από τα συνενωμένα νέφη σημείων καθώς και τις αντίστοιχες θέσεις και στροφές μέσω οδομετρίας. Στη συνέχεια χρησιμοποιείται βελτιστοποίηση γράφου θέσης (Pose Graph Optimization) όπου οι κόμβοι αντιστοιχίζονται σε θέσεις οχήματος και οι ακμές αντιπροσωπεύουν μετρήσεις αισθητήρων LiDAR και IMU. Ο γράφος θέσης περιέχει ακμές που περιλαμβάνουν συχνά αντιφατικές πληροφορίες, λόγω των τυχαίων σφαλμάτων των μετρήσεων. Στη συνέχεια, οι κόμβοι του γράφου βελτιστοποιούνται για να βρεθεί το σύνολο των θέσεων που περιγράφουν βέλτιστα τις μετρήσεις ελαχιστοποιώντας τις αντιφάσεις. Ακόμη, όπως έχει αναφερθεί, σημαντικό ρόλο στην βελτιστοποίηση του γράφου θέσης παίζει και η δυνατότητα ανίχνευσης κλεισίματος βρόχων δηλαδή η δυνατότητα αναγνώρισης περιοχής που έχει ξαναανιχνεύσει ο αισθητήρας, με σκοπό την διόρθωση του σφάλματος που έχει συγκεντρωθεί έως την συγκεκριμένη χρονική στιγμή (σφάλμα ολίσθησης). Αυτή η ανίχνευση όπως έχει αναφερθεί επιτυγχάνεται συνήθως μέσω περιγραφέων. Στην συγκεκριμένη περίπτωση χρησιμοποιείται ο Scan Context [41]. Ακολούθως, η βελτιστοποίηση του γράφου πραγματοποιείται μέσω του αλγορίθμου g2o-levenberg-marquardt, που αποτελεί και την συχνότερη προσέγγιση.

Το συγκεκριμένο παράδειγμα ωστόσο δεν παρέχει αληθή δεδομένα για την αξιολόγηση του αποτελέσματος. Αποτελεί επίσης παράδειγμα το οποίο υλοποιεί μια σχετικά απλή εκδοχή μεθόδου graph-based SLAM, αντίστοιχης με το πρώτο παράδειγμα με μόνη διαφορά τον τρόπο υπολογισμού των αρχικών προσεγγίσεων. Τα αποτελέσματα του παραδείγματος παρουσιάζονται στα Σχήματα 13 και 14.



Σχήμα 13: Αποτέλεσμα χάρτη και τροχιάς (i) - παράδειγμα 2 MATLAB



Σχήμα 14: Αποτέλεσμα χάρτη και τροχιάς (ii) - παράδειγμα 2 MATLAB

4.1.3 Implement SLAM with LiDAR Scans

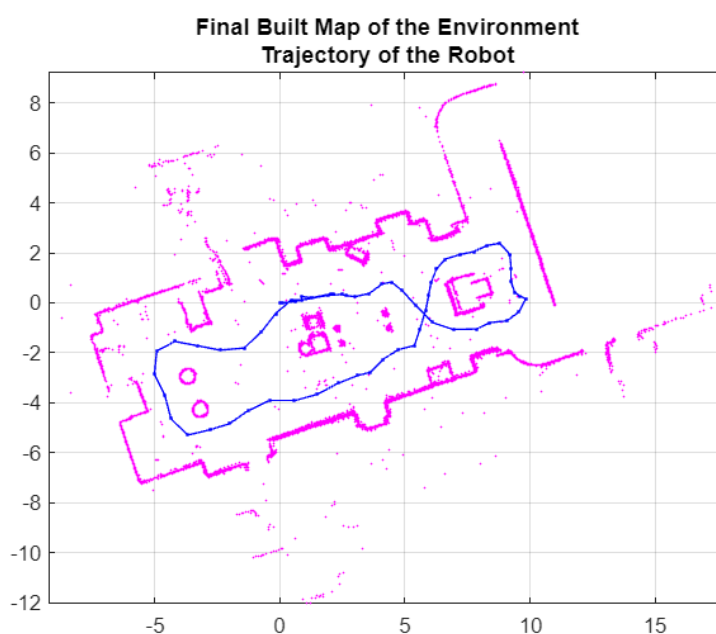
Η συγκεκριμένη υλοποίηση χρησιμοποιεί ως δεδομένα εισόδου LiDARScan objects διπλής ακρίβειας (8 byte ανά τιμή). Τα LiDARScans αποτελούν δομές του MATLAB που περιγράφουν δεδομένα σαρώσεων από 2D LiDAR, τα οποία αποτελούν απαραίτητα δεδομένα εισόδου για την δημιουργία LiDARSLAM objects μέσω της έτοιμης συνάρτησης LiDARSLAM. Ακόμη, χρησιμοποιούνται οι συναρτήσεις:

`addscan` για την συνένωση νεφών σημείων όπου χρησιμοποιείται μια 'BranchAndBound' [90] βελτιστοποίηση των μετασχηματισμών Fourier [91] των δυο «εικόνων» που προκύπτουν πρακτικά από τις 2D σαρώσεις [92]

`detectLoopClosure` (δεν αποσαφηνίζεται ποια μέθοδος matching χρησιμοποιείται)

`optimizePoseGraph` μέσω του "built-in-trust-region" (χρησιμοποιεί τον αλγόριθμο που περιγράφεται στο [93] για την βελτιστοποίηση του γράφου ως προεπιλεγμένη μέθοδο, ενώ υπάρχει επιλογή και για χρήση του "g2o-Levenberg-Marquardt" [43])

Με βάση τα παραπάνω δημιουργείται τελικά ένας post process SLAM αλγόριθμος με οδομετρία μέσω συνενώσεων νεφών, βελτιστοποίησης γράφου και εντοπισμού κλειστών βρόχων που βασίζεται σε 2D LiDAR δεδομένα και ακολουθεί μια προσέγγιση εικόνας για τα 2D νέφη σημείων. Εν τέλει, η συγκεκριμένη υλοποίηση διαφέρει σημαντικά από τις σύγχρονες μεθόδους LiDAR-SLAM καθώς αυτές δεν αξιοποιούν γενικά 2D δεδομένα. Τέλος, δεν παρέχονται ακριβή Ground truth δεδομένα, συγκεκριμένα παρέχεται μόνο ως χοντρική οπτική αναπαράσταση σύγκριση του αποτελέσματος με ένα υποθετικό Ground Truth σχέδιο που αντιστοιχεί στον πραγματικό χάρτη (σχήμα 16).

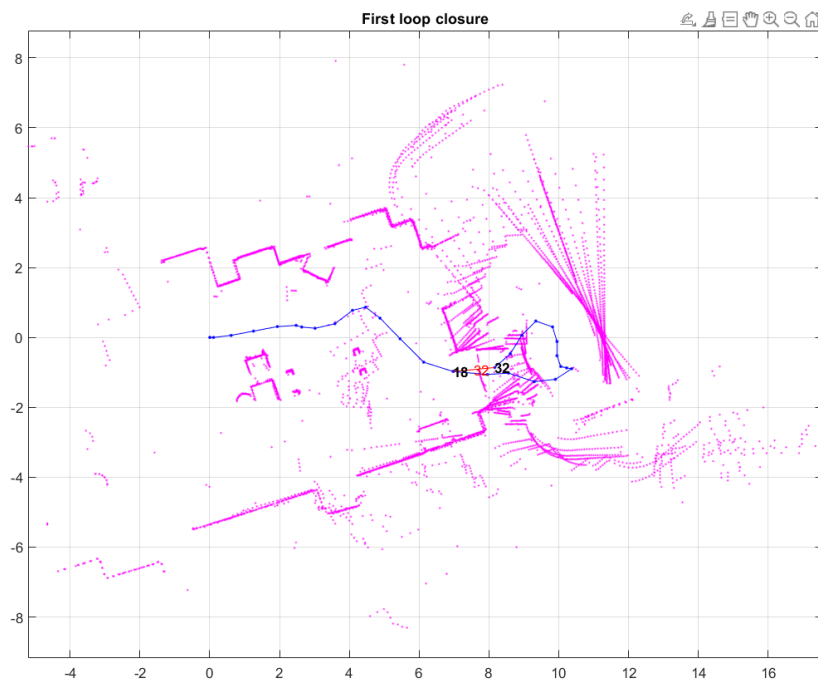


Σχήμα 15: Αποτέλεσμα χάρτη και τροχιάς - παράδειγμα 3 MATLAB

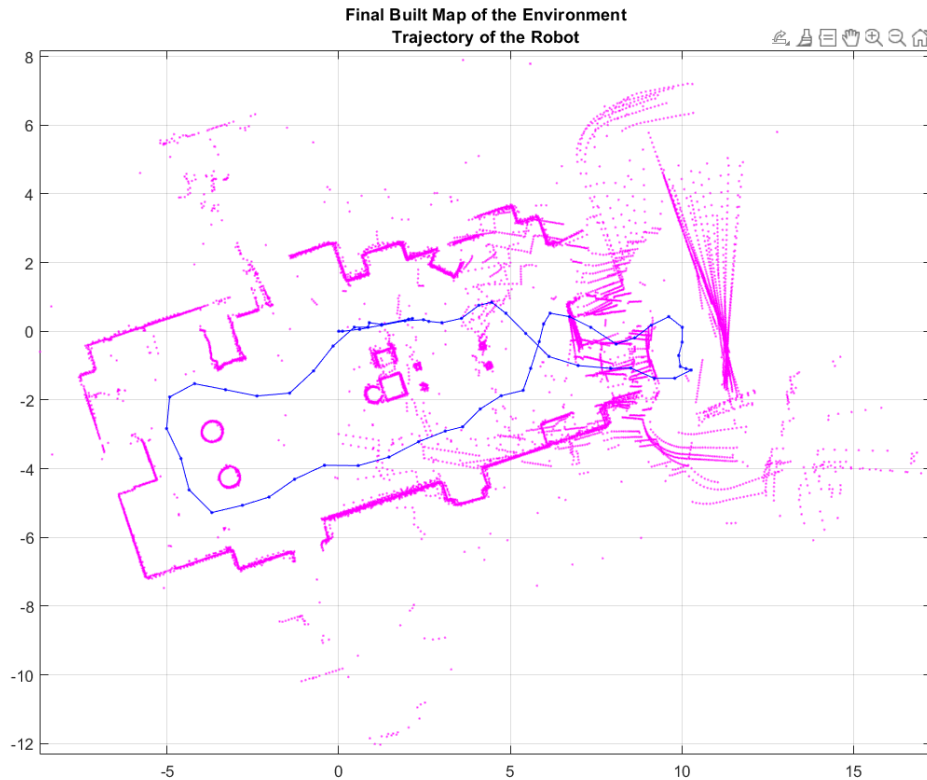


Σχήμα 16: Σύγκριση χάρτη με Ground Truth- παράδειγμα 3 MATLAB

Το αποτέλεσμα αποτελεί καλή προσέγγιση του ορόφου, ωστόσο πάλι οι παράμετροι που έχουν χρησιμοποιηθεί για τους βρόχους δεν προσαρμόζονται στα δεδομένα και έχουν επιλεγεί εκ των προτέρων εμπειρικά όπως επισημαίνεται. Ύστερα από δοκιμή με πολύ μικρή μεταβολή της τιμής της παραμέτρου LoopClosureThreshold (από 210 σε 200) οδηγηθήκαμε σε λανθασμένο εντοπισμούς βρόχου και συνεπώς σε εσφαλμένο αποτέλεσμα. Τα αποτελέσματα της δεύτερης δοκιμής παρουσιάζονται παρακάτω στα Σχήματα 17 και 18.



Σχήμα 17: Αποτέλεσμα εντοπισμού βρόχου 2^{ης} δοκιμής- παράδειγμα 3 MATLAB

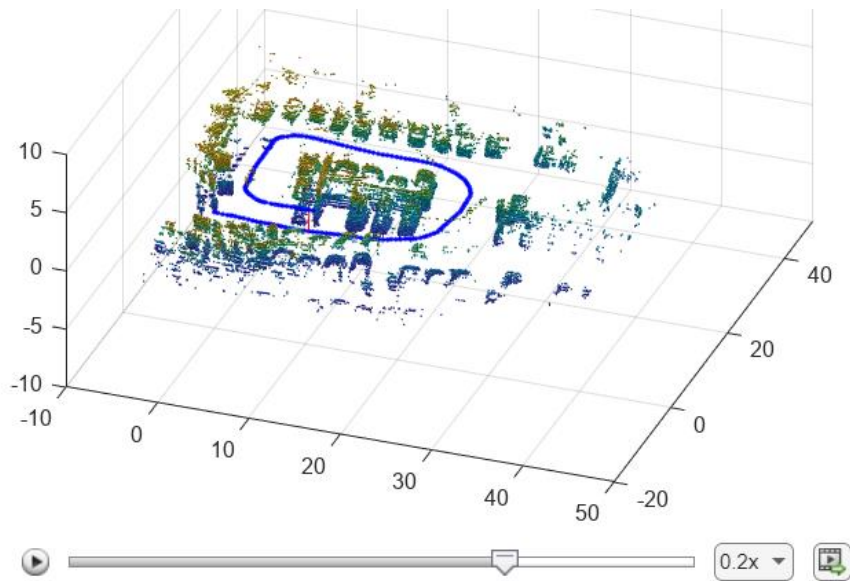


Σχήμα 18: Αποτέλεσμα χάρτη και τροχιάς 2^{ης} δοκιμής - παράδειγμα 3 MATLAB

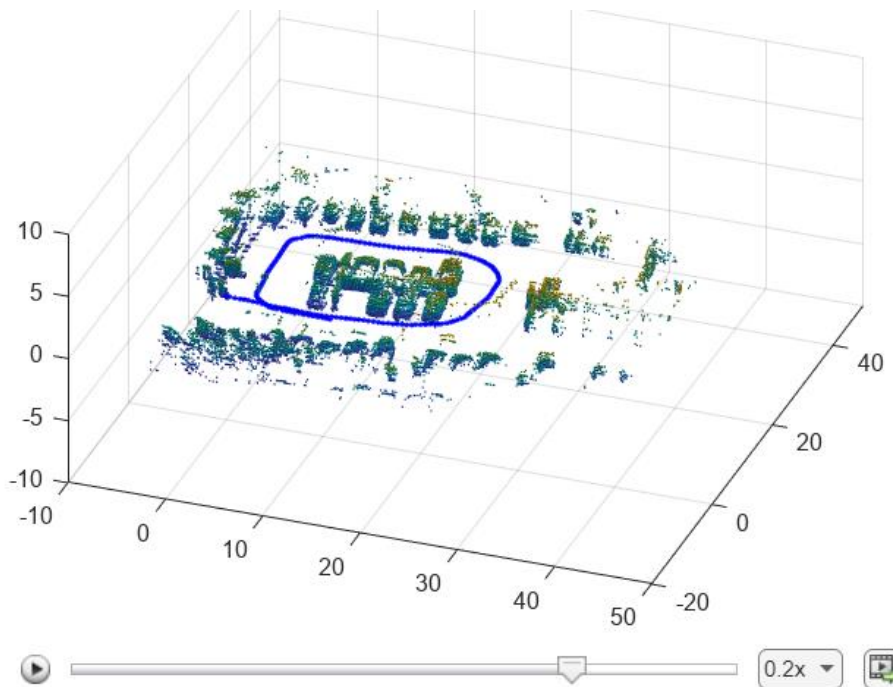
Παρατηρείται λοιπόν πως το συγκεκριμένο παράδειγμα πρόκειται για ασταθή υλοποίηση στην οποία μικρή αλλαγή των παραμέτρων, οι οποίες έχουν ούτως ή άλλως τοποθετηθεί εμπειρικά για το συγκεκριμένο παράδειγμα, μπορεί να επιφέρουν τεράστια σφάλματα.

4.1.4 Perform SLAM using 3D LiDAR Point Clouds

Ως βασικό στοιχείο εισόδου χρησιμοποιεί νέφη σημείων όμως υπό την μορφή πινάκων N (σημείων) \times 3 (X,Y,Z διαστάσεις) του κάθε σημείου απλής ακρίβειας τιμών. Ακόμη πρόκειται για μέθοδο που χρησιμοποιεί 3D LiDAR δεδομένα τα οποία παρέχονται ως δοκιμαστικά χωρίς Ground Truth, ενώ επιπλέον βασίζεται σε βελτιστοποίηση γράφου. Γίνεται αρχικά προεπεξεργασία για αφαίρεση σημείων που αντιστοιχούν στο όχημα, στο έδαφος και σε τυχόν οροφή και τελικά μετατρέπονται μετά το φιλτράρισμα σε LiDARScan objects (δομές MATLAB που περιγράφουν σαρώσεις) με μετρήσεις διπλής ακρίβειας, ενώ χρησιμοποιείται ο αλγόριθμος NDT για την συνένωση των διαδοχικών νεφών σημείων. Τέλος, πραγματοποιείται η βελτιστοποίηση μέσω της έτοιμης συνάρτησης optimizePoseGraph με τις προεπιλεγμένες παραμέτρους. Τα αποτελέσματα της μεθόδου παρουσιάζονται παρακάτω στα σχήματα 19 και 20.



Σχήμα 20: Αποτέλεσμα χάρτη και τροχιάς πριν από βελτιστοποίηση - παράδειγμα 4 MATLAB



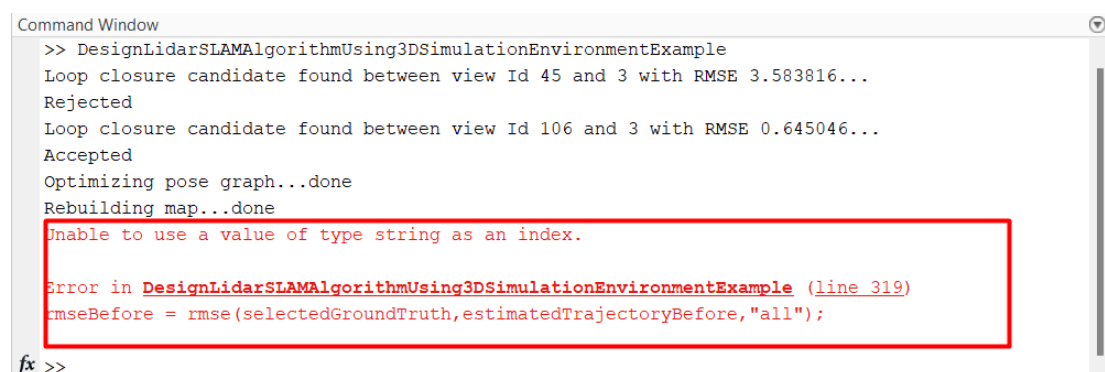
Σχήμα 19: Αποτέλεσμα χάρτη και τροχιάς ύστερα από βελτιστοποίηση - παράδειγμα 4 MATLAB

Από τα αποτελέσματα παρατηρούμε πως εντοπίζεται και διορθώνεται σωστά ο χάρτης και οι θέσεις του οχήματος που προέκυψαν από την οδομετρία, μέσω εντοπισμού βρόχου και βελτιστοποίησης. Ο συγκεκριμένος αλγόριθμος γενικά πλησιάζει πολύ την δομή ενός σύγχρονου αλγορίθμου SLAM που βασίζεται σε Γράφους και αξιοποιεί 3D LiDAR δεδομένα, ωστόσο χρησιμοποιεί μια ασυνήθιστη δομή εισόδου τύπου πίνακα η οποία δεν θα μπορούσε να χρησιμοποιηθεί σε πραγματικές real time εφαρμογές με χρήση κάποιου σαρωτή.

4.1.5 Design LiDAR SLAM Algorithm Using 3D Simulation Environment

Γενικά ο αλγόριθμος του συγκεκριμένου παραδείγματος χρησιμοποιεί το Unreal Engine της Epic Games [94], δηλαδή μια πλατφόρμα ανάπτυξης 3D γραφικών, για να δημιουργήσει προσομοίωση ενός 3D περιβάλλοντος, ενώ παράλληλα δύναται να παράγει προσομοιωμένες μετρήσεις IMU/INS και LiDARScans δομές συγκεκριμένου αισθητήρα από το περιβάλλον αυτό. Η συγκεκριμένη δυνατότητα κρίνεται εξαιρετικά χρήσιμη καθώς έτσι μπορούν να πραγματοποιηθούν έλεγχοι κάθε είδους σε έναν αλγόριθμο SLAM, για διαφορετικές συνθήκες οι οποίες θα ήταν δύσκολο και κοστοβόρο για να υλοποιηθούν στην πραγματικότητα, ενώ μπορούν να υπολογιστούν με ακρίβεια τα σφάλματα της εκάστοτε μεθόδου SLAM, καθώς οι προσομοιώσεις αυτές μας παρέχουν 100% ακριβές Ground Truth.

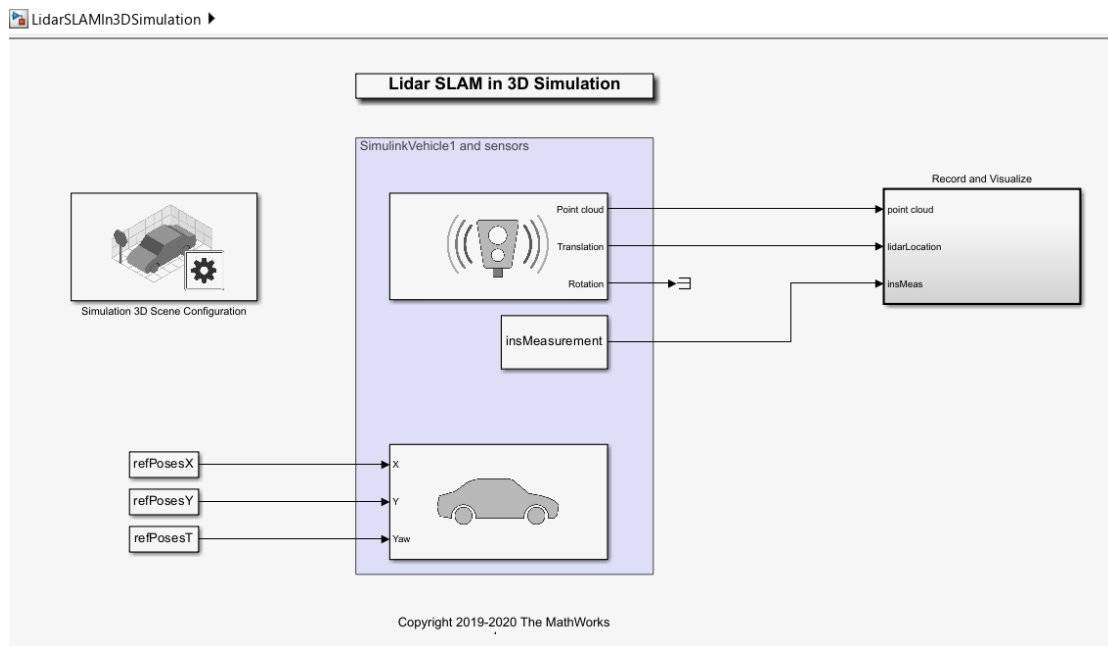
Συγκεκριμένα, δημιουργούν PointCloudObjects με θέσεις (X,Y,Z) απλής ακρίβειας και IMU/INS δεδομένα. Χρησιμοποιεί αρχικά προσεγγιστικούς μετασχηματισμούς μέσω των μετρήσεων IMU/INS ώστε μετά να μπορεί να χρησιμοποιήσει τον Generalized Iterative Closest Point (G-ICP) για την συνένωση των νεφών σημείων, ενώ ύστερα χρησιμοποιεί και Pose Graph Optimization μέσω της έτοιμης συνάρτησης optimizePoseGraph, με την προεπιλεγμένη μέθοδο και έλεγχο κλειστών βρόχων μέσω Scan Context Descriptors. Γενικά λοιπόν πρόκειται για μια αντίστοιχη υλοποίηση και με τα υπόλοιπα παραδείγματα που χρησιμοποιεί λιγότερες όμως ασαφείς έτοιμες συναρτήσεις ενώ παράλληλα χρησιμοποιεί προσομοιωμένα δεδομένα και άρα μας παρέχει βέβαια Ground truth. Ωστόσο, το συγκεκριμένο παράδειγμα δεν λειτουργούσε εξ αρχής καθώς αντιμετώπιζε πρόβλημα σφάλματος (εικόνα 6). Αυτό οφείλεται στην χρήση της έτοιμης συνάρτησης rmse για τον υπολογισμό του σφάλματος πριν και μετά το Pose Graph Optimization, ενώ ο κώδικας χρησιμοποιεί παράλληλα μεταβλητή με το ίδιο όνομα. Τελικά διορθώθηκε στα πλαίσια της διπλωματικής μέσω εκκαθάρισης της μεταβλητής που δεν είναι πια αναγκαία για να μην υπάρχει conflict με την έτοιμη συνάρτηση υπολογισμού σφάλματος rmse (clear στην γραμμή 318).



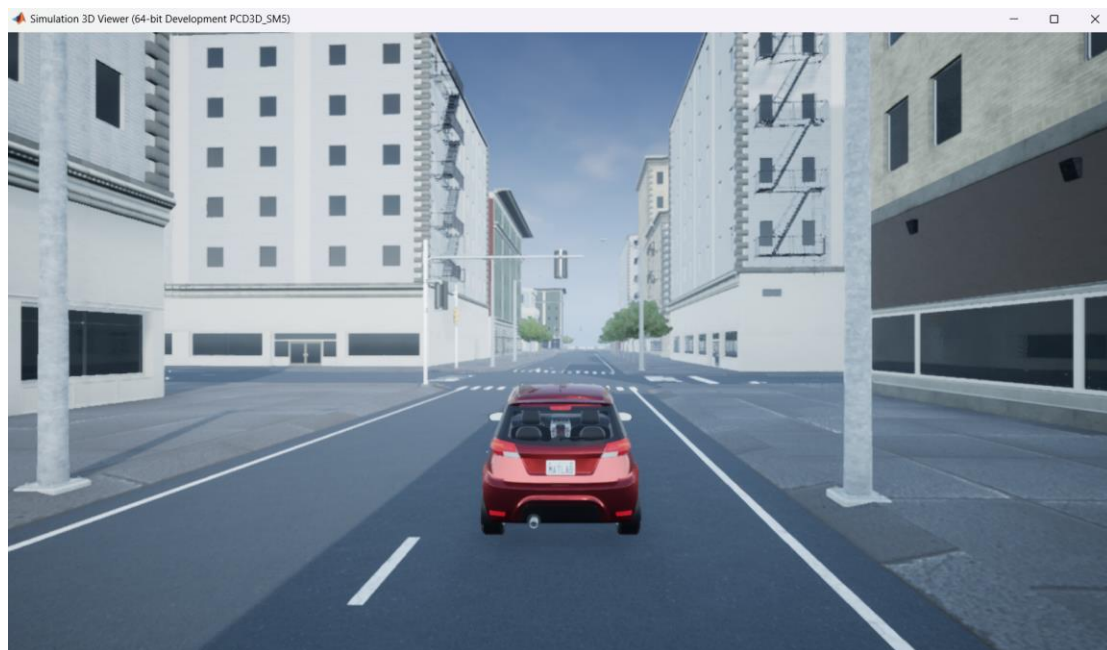
```
Command Window
>> DesignLidarSLAMAlgorithmUsing3DSimulationEnvironmentExample
Loop closure candidate found between view Id 45 and 3 with RMSE 3.583816...
Rejected
Loop closure candidate found between view Id 106 and 3 with RMSE 0.645046...
Accepted
Optimizing pose graph...done
Rebuilding map...done
Unable to use a value of type string as an index.
Error in DesignLidarSLAMAlgorithmUsing3DSimulationEnvironmentExample (line 319)
rmseBefore = rmse(selectedGroundTruth,estimatedTrajectoryBefore,"all");
fx >>
```

Εικόνα 6: Σφάλμα μεταβλητής του αλγορίθμου - παράδειγμα 5 MATLAB

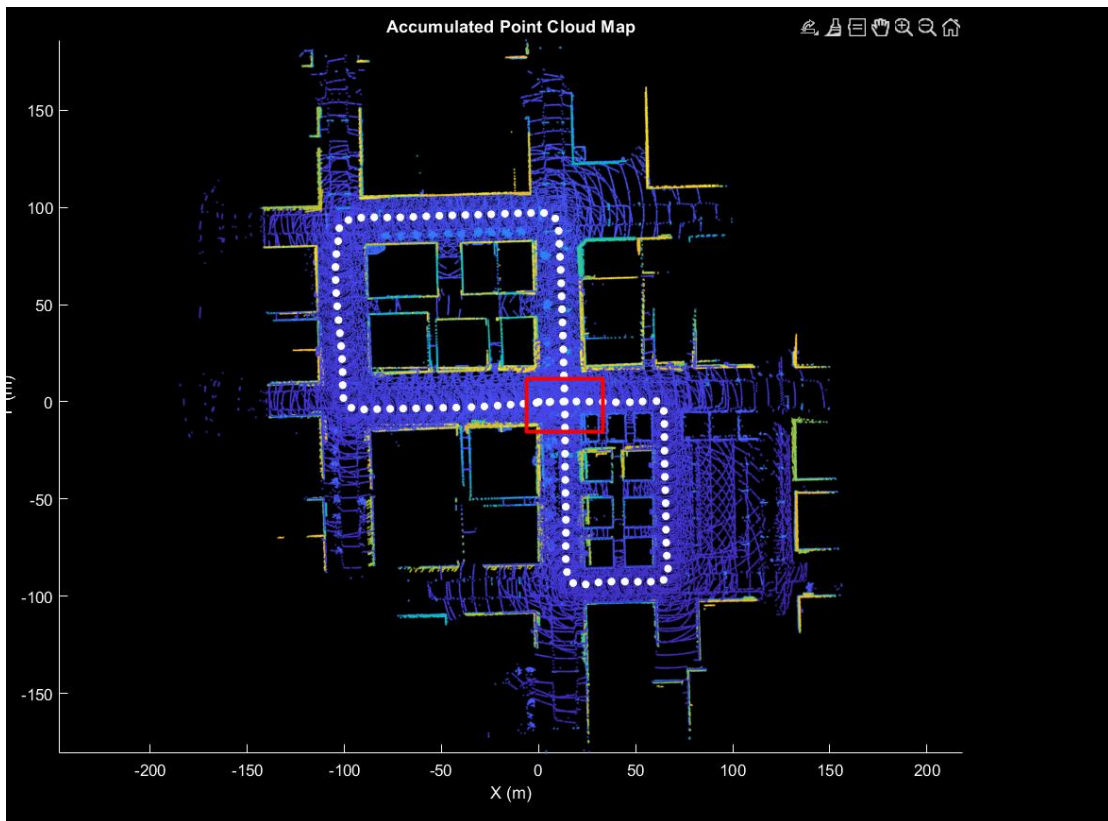
Τα αποτελέσματα της μεθόδου παρουσιάζονται παρακάτω στα σχήματα 22, 23 ενώ στο σχήμα 21 απεικονίζεται ένα διάγραμμα περιγραφής της προσομοίωσης. Τέλος, στην εικόνα 7 απεικονίζεται το προσομοιωμένο περιβάλλον (δρόμος με κτίρια).



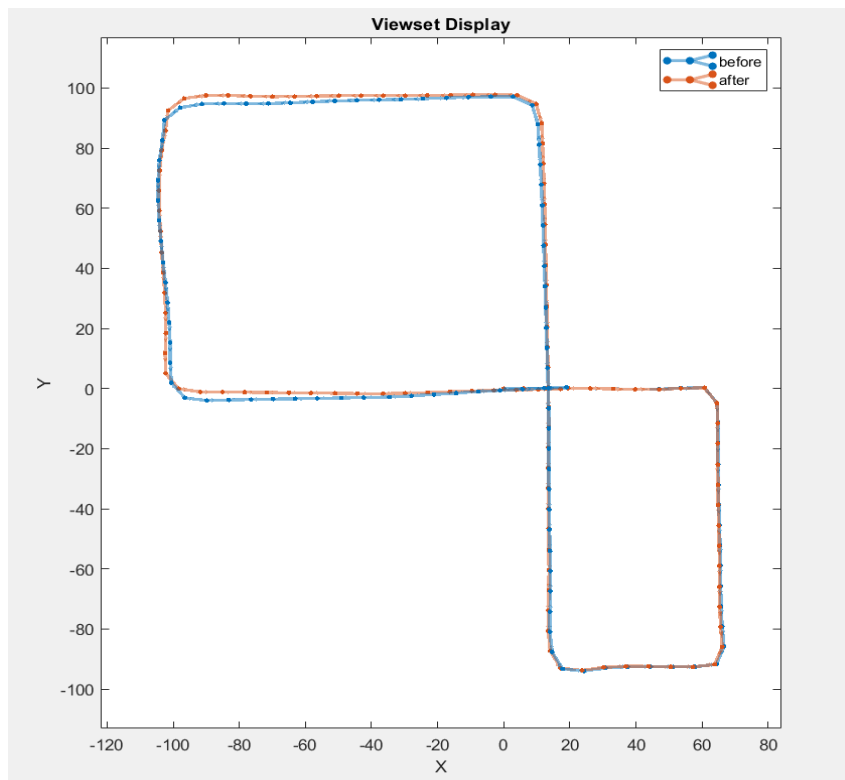
Σχήμα 21: Σχεδιάγραμμα προσομοίωσης - παράδειγμα 5 MATLAB



Εικόνα 7: Στιγμιότυπο από το προσομοιωμένο περιβάλλον – παράδειγμα 5 MATLAB



Σχήμα 22: Αποτέλεσμα χάρτη και τροχιάς - παράδειγμα 5 MATLAB



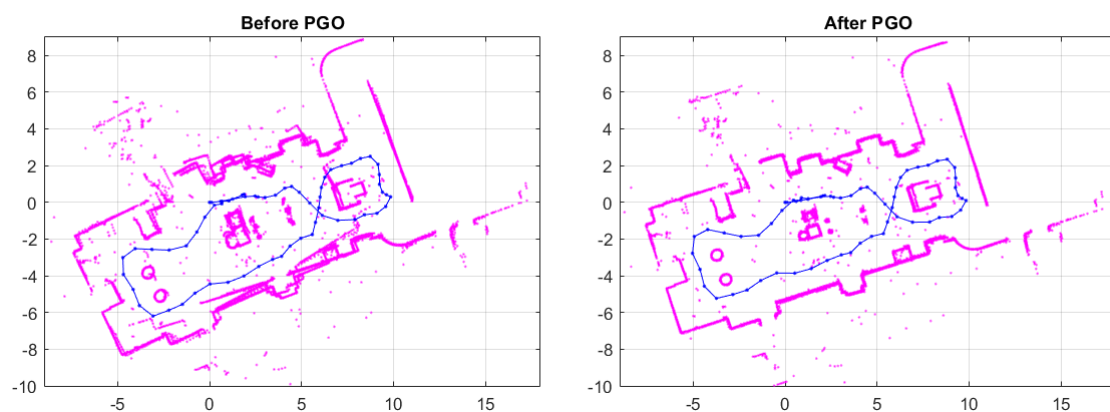
Σχήμα 23: Αποτέλεσμα τροχιάς πριν και μετά την βελτιστοποίηση - παράδειγμα 5 MATLAB

Στο παράδειγμα αυτό πραγματοποιήθηκε πρακτικά δυο φορές κλείσιμο βρόχου στο ίδιο σημείο όπου τέμνονται τα δυο οικοδομικά τετράγωνα (επισημαίνεται στο σχήμα 22), με τον αλγόριθμο να αναγνωρίζει και τα δυο περάσματα ωστόσο να απορρίπτει το πρώτο πριν την βελτιστοποίηση, ενώ το δεύτερο το οποίο πρακτικά ήταν το σημείο της τελευταίας σάρωσης λήφθηκε κανονικά υπόψιν και βελτίωσε το αποτέλεσμα καθώς το rmse μεταξύ ground truth τροχιάς και ήταν 1.4707 μέτρα πριν την βελτιστοποίηση και 0.9088 εκ των υστέρων.

4.1.6 Build Map from 2-D LiDAR Scans Using SLAM Example

Ο συγκεκριμένος αλγόριθμος αφορά μια υλοποίηση που βασίζεται σε 2D LiDAR δεδομένα, ακριβώς αντίστοιχη με το τρίτο παράδειγμα, ενώ χρησιμοποιεί επίσης τα ίδια δεδομένα εισόδου. Βασικός του σκοπός είναι απλώς η εκτύπωση γραφημάτων που επικεντρώνονται στο να παρουσιάσουν το αποτέλεσμα που έχει η βελτιστοποίηση γράφου στο τελικό αποτέλεσμα. Για αυτόν τον λόγο δεν θα πραγματοποιηθεί ιδιαίτερη ανάλυση του αποτελέσματος του.

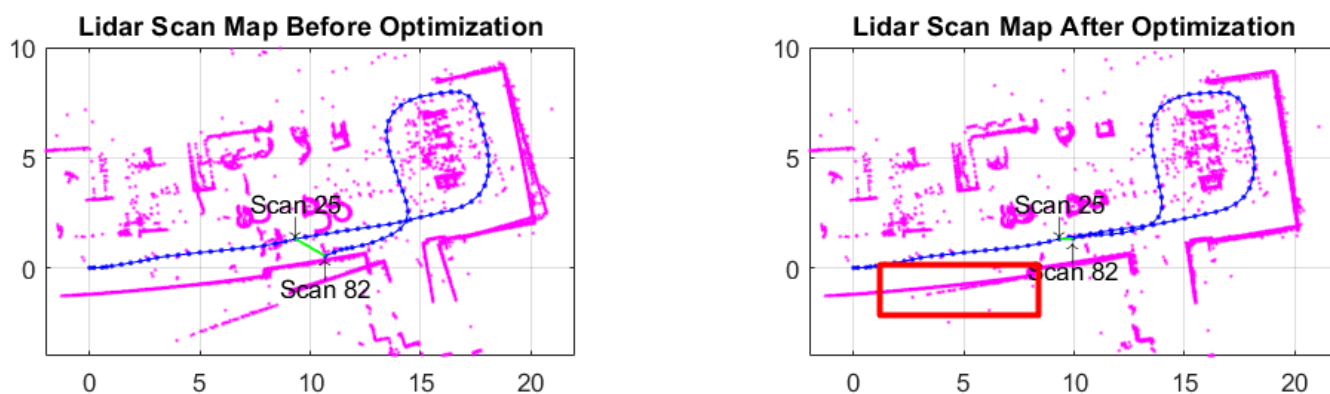
Map of the Environment and Robot Trajectory



Σχήμα 24: Αποτέλεσμα τροχιάς και χάρτη πριν και μετά την βελτιστοποίηση - παράδειγμα 6 MATLAB

4.1.7 Map Indoor Area Using LiDAR SLAM and Factor Graph Example

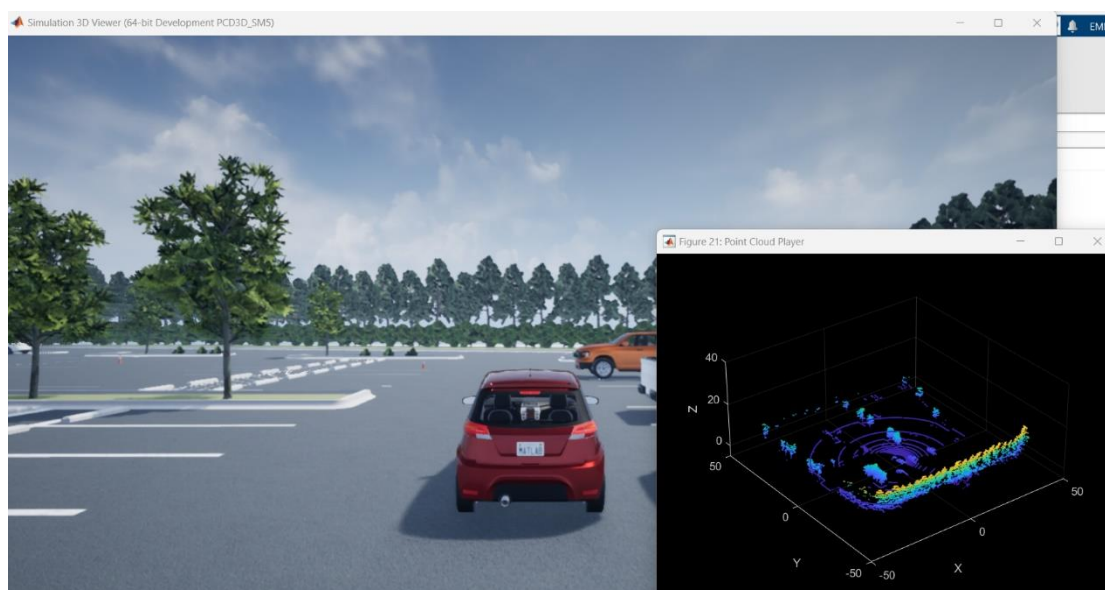
Ο αλγόριθμος του παραδείγματος αποτελεί ακόμη μια σχετικά απλοϊκή υλοποίηση που απαιτεί ως δεδομένα εισόδου LiDARScan objects από 2D LiDAR σαρώσεις εσωτερικού χώρου. Επιπλέον, χρησιμοποιεί αποκλειστικά τις έτοιμες συναρτήσεις για συνένωση νεφών, εύρεση βρόχων και βελτιστοποίηση και συνεπώς δεν διαφοροποιείται από άλλα αντίστοιχα παραδείγματα (παραδείγματα 3 και 6), ενώ εμφανίζει χειρότερο αποτέλεσμα από αυτά όπως φαίνεται στο σχήμα 25, αφού ακόμη και μετά από την βελτιστοποίηση γίνεται εμφανής η στροφή ενός τοίχου του χάρτη. Τέλος, εμφανίζεται και θόρυβος υπό την μορφή κουκίδων.



Σχήμα 25: Αποτέλεσμα τροχιάς και χάρτη πριν και μετά την βελτιστοποίηση - παράδειγμα 7 MATLAB

4.1.8 Build an Occupancy Map From LiDAR Data Using Slam Example

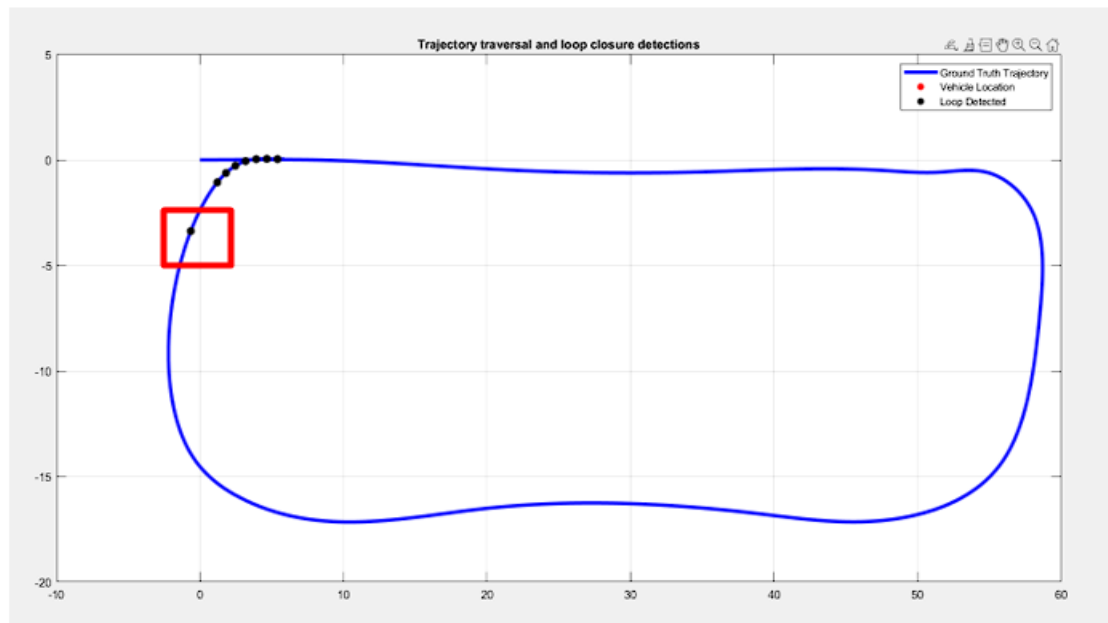
Αποτελεί ακόμη ένα παράδειγμα της MathWorks το οποίο χρησιμοποιεί το Unreal Engine της Epic Games για να κάνει simulate ένα 3D περιβάλλον και LiDARScans από το περιβάλλον αυτό. Η δυνατότητα προσομοίωσης περιβάλλοντος και μετρήσεων αποτελεί από μόνη της ένα πολύ χρήσιμο εργαλείο ανεξάρτητα από την μέθοδο SLAM που υλοποιεί στην συνέχεια το παράδειγμα. Στο συγκεκριμένο παράδειγμα δημιουργούνται PointCloudObjects τα οποία προσομοιώνουν μετρήσεις 3D νεφών σημείων με θέσεις (X,Y,Z) απλής ακρίβειας αρχικά, ενώ ύστερα από την προεπεξεργασία, μετατρέπονται σε LiDARScans για την χρήση του LiDARSlam function. Τέλος, πραγματοποιούνται οι συνενώσεις νεφών σημείων μέσω του αλγορίθμου phase correlation (έτοιμη συνάρτηση pcregistercorr), ενώ για τις υπόλοιπες διαδικασίες χρησιμοποιούνται και πάλι κυρίως οι έτοιμες συναρτήσεις addscan, detectLoopClosure και optimizePoseGraph. Παρακάτω παρουσιάζονται συνοπτικά τα αποτελέσματα του αλγορίθμου στα σχήματα 26-28.



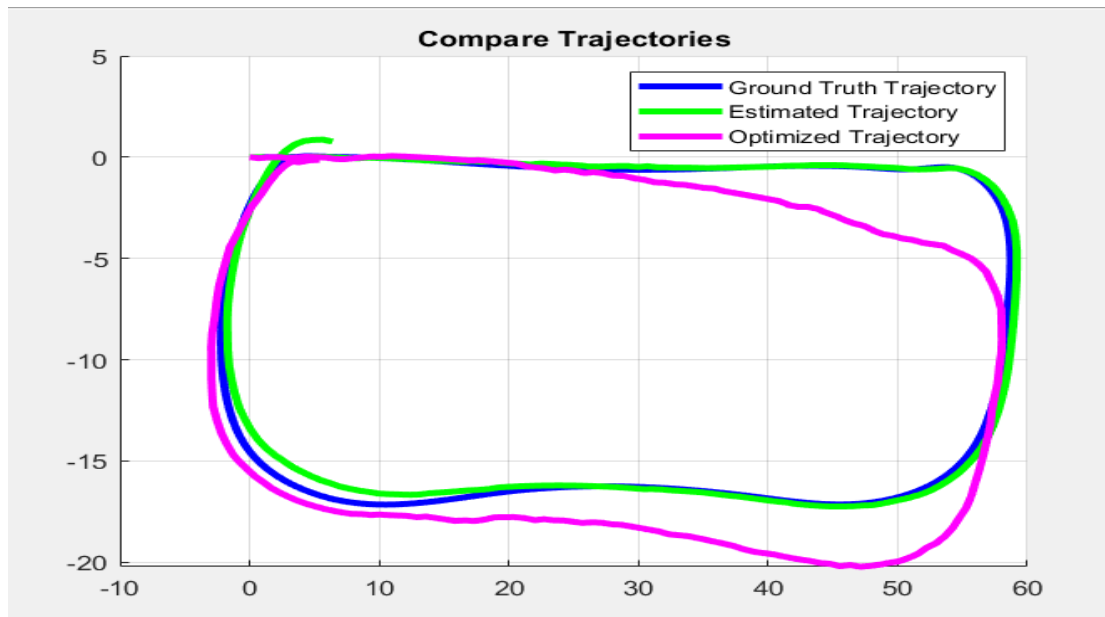
Εικόνα 8: Στιγμιότυπο από το προσομοιωμένο περιβάλλον – παράδειγμα 8 MATLAB



Σχήμα 26: Ground Truth χάρτη και τροχιά – παράδειγμα 8 MATLAB



Σχήμα 27: Θέσεις εντοπισμού κλεισίματος βρόχων - παράδειγμα 8 MATLAB



Σχήμα 28: Σύγκριση τροχιάς πριν και μετά την βελτιστοποίηση - παράδειγμα 8 MATLAB

Γενικά, από τον συγκεκριμένο αλγόριθμο είναι άξιο παρατήρησης ότι παρόλο που η οδομετρία έχει δώσει ένα αποτέλεσμα αρκετά κοντά στην αληθή τροχιά του οχήματος, μετά την βελτιστοποίηση απομακρύνεται σε απαγορευτικό βαθμό από αυτήν (σχήμα 28). Αυτό φαίνεται να συμβαίνει λόγω εσφαλμένου κλεισίματος βρόχου (επισημαίνεται στο σχήμα 27), που εντοπίζεται πριν από την συνολική βελτιστοποίηση με αποτέλεσμα να δεσμεύεται η τροχιά σε δυο εσφαλμένα σημεία και να δημιουργείται στροφή. Το γεγονός αυτό φανερώνει ότι οι παράμετροι που έχουν χρησιμοποιηθεί δεν είναι οι βέλτιστες για το συγκεκριμένο περιβάλλον, πράγμα λογικό καθώς δεν υπάρχει δυνατότητα δυναμικής προσαρμογής των παραμέτρων.

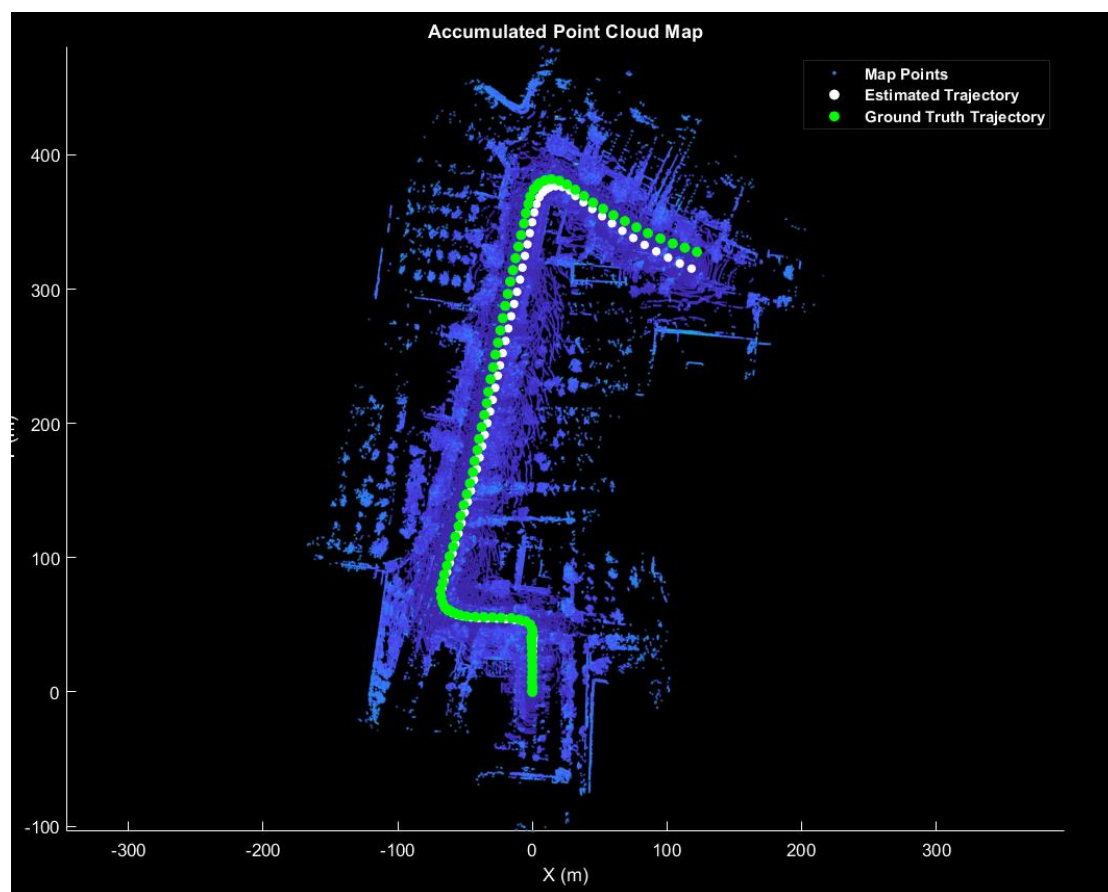
4.1.9 Build A Map From LiDAR Data Example

Το συγκεκριμένο παράδειγμα αποτελεί υλοποίηση 3D LiDAR-SLAM, απαιτεί ως δεδομένα εισόδου Timestamped Pointcloud Objects (X,Y,Z,intensity) απλής ακρίβειας και μετρήσεων IMU, ενώ παρέχονται και μετρήσεις GNSS ως ground truth.

Ακολουθεί την λογική του κλασσικού SLAM ή απλής οδομετρίας και η οποία δεν έχει επιφέρει ικανοποιητικά αποτελέσματα. Συγκεκριμένα, πραγματοποιεί τις συνενώσεις των νεφών μέσω του Generalized-ICP (συνάρτηση pcregistericp), ενώ οι αρχικοί μετασχηματισμοί υπολογίζονται στο ένα σκέλος σύμφωνα με το constant velocity model χρησιμοποιώντας τον προηγούμενο μετασχηματισμό ως νέα προσέγγιση. Επίσης για βελτίωση των αποτελεσμάτων χρησιμοποιεί τις μετρήσεις INS/IMU ως αρχικούς μετασχηματισμούς για τον ICP. Τέλος, συγκρίνει το αποτέλεσμα με βάση το Ground Truth όμως δεν χρησιμοποιεί κάποια μέθοδο βελτιστοποίησης, ούτε κλεισίματος βρόχου, (αν και πρακτικά δεν υλοποιείται βρόχος στο συγκεκριμένο παράδειγμα καθώς ο αισθητήρας περνάει από κάθε σημείο μόνο μια φορά) υλοποιώντας έτσι ουσιαστικά απλά μια οδομετρία που συσσωρεύει προοδευτικά σφάλματα, γεγονός που αποτυπώνεται και από το τελικό

αποτέλεσμα. Στην πρώτη περίπτωση, χωρίς τις μετρήσεις IMU, προκύπτει σφάλμα συνολικής τροχιάς ίσο με 9.773 m, ενώ με χρήση των μετρήσεων IMU ως προσεγγίσεις για τον ICP το σφάλμα πέφτει στα 3.461 μέτρα. Τέτοια σφάλματα κρίνονται απαγορευτικά και ειδικά για διαδρομές με μήκος μόλις 500 m. Τα αποτελέσματα του παραδείγματος παρουσιάζονται στο σχήμα 29.

4.1.10 Build a Map with LiDAR Odometry and Mapping (LOAM) Using UE



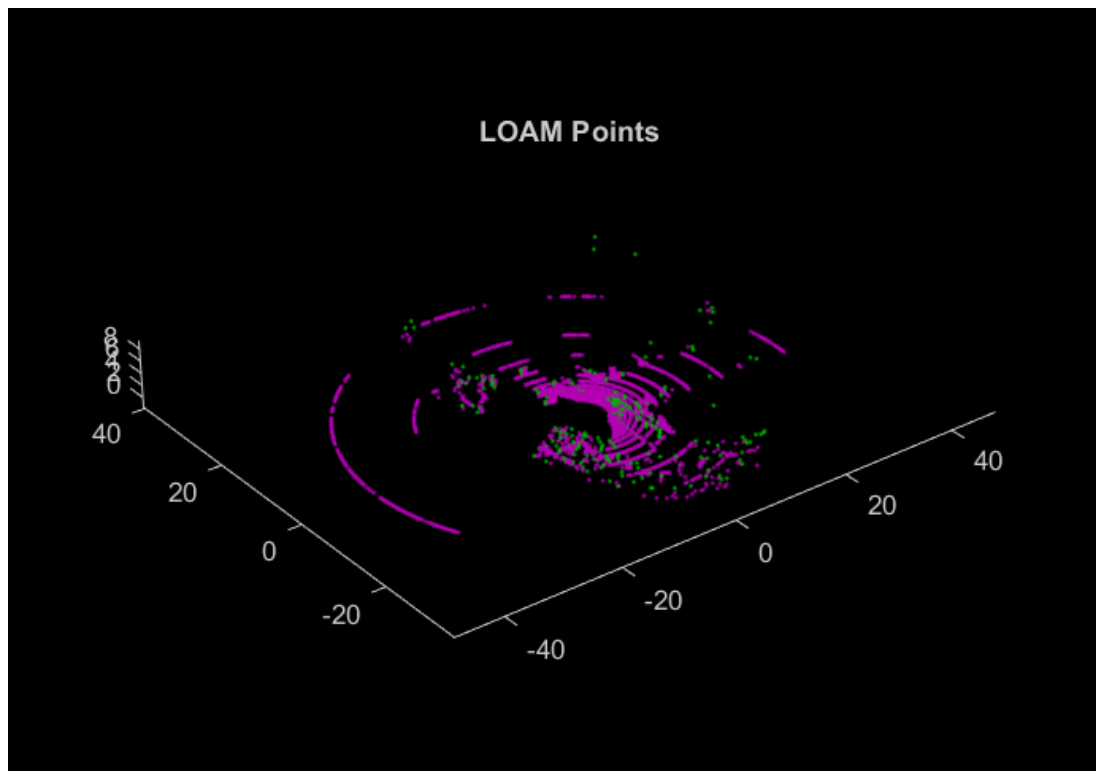
Σχήμα 29: Αποτελέσματα χάρτη και τροχιάς – παράδειγμα 9 MATLAB

Το τελευταίο και πιο σύγχρονο παράδειγμα LiDAR-SLAM αποτελεί μια προσπάθεια υλοποίησης του LOAM [32] σε περιβάλλον MATLAB πάνω σε δεδομένα που έχουν προκύψει από προσομοίωση μέσω της Unreal Engine της εταιρείας Epic Games. Τα δεδομένα εισόδου που προκύπτουν από την προσομοίωση αφορούν PointCloud objects (X,Y,Z) απλής ακρίβειας. Επισημαίνεται επίσης ότι το προσομοιωμένο περιβάλλον του παραδείγματος είναι ίδιο με αυτό του παραδείγματος 8 με μόνη διαφορά στον αριθμό των σταθμευμένων οχημάτων στον χώρο.

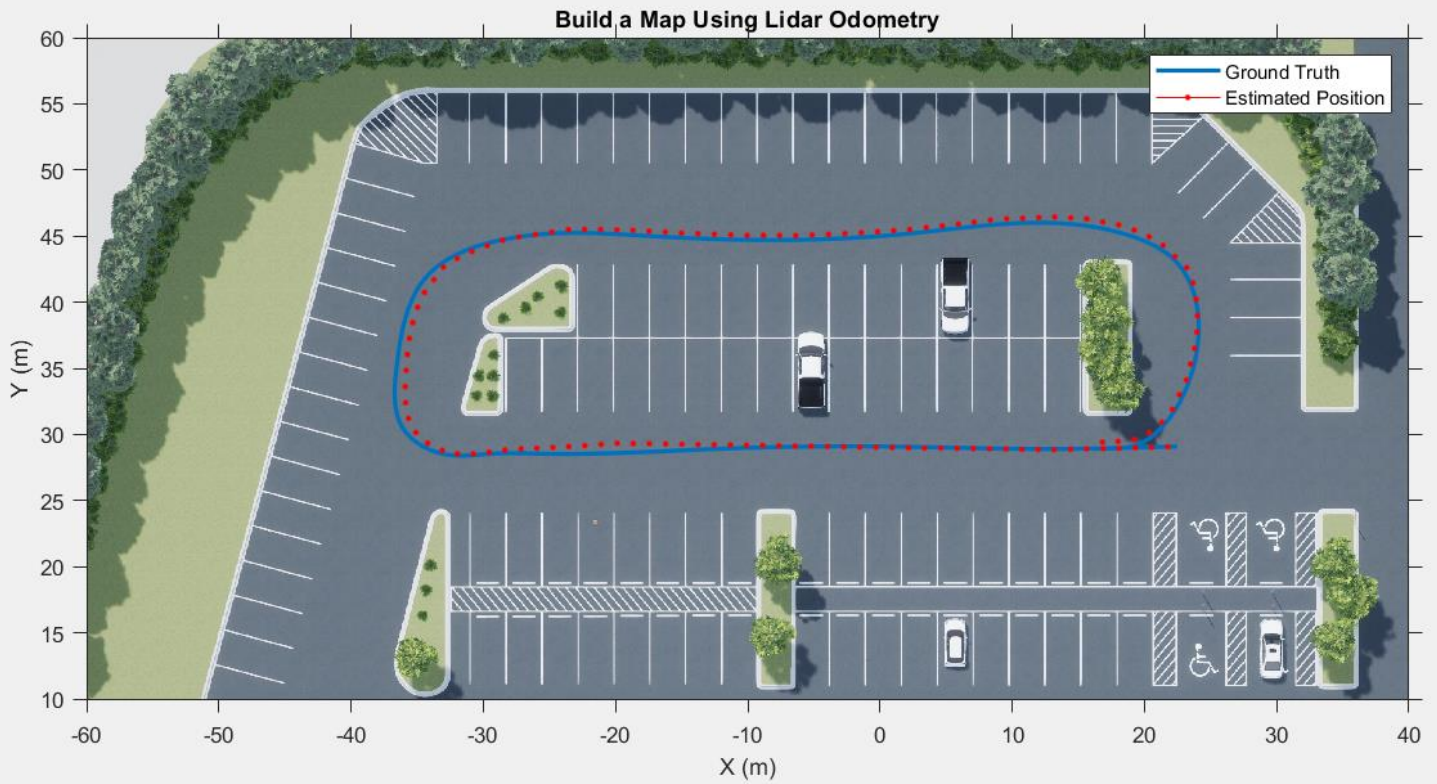
Ο αλγόριθμος πραγματοποιεί τις συνενώσεις νεφών με βάση τα LOAM Features δηλαδή 3D σημεία που χαρακτηρίζονται από την γεωμετρία στην οποία ανήκουν ως sharp-edge, less-sharp-edge, planar-surface, or less-planar-surface (ακμή, επίπεδη επιφάνεια, όχι τόσο έντονη ακμή και όχι τόσο έντονη επιφάνεια). Τα χαρακτηριστικά

αυτά υπολογίζονται μέσω του έτοιμου αλγόριθμου detectLOAMFeatures (Σχήμα 30). Ύστερα ο αλγόριθμος χρησιμοποιεί τα LOAM Points για να κάνει τις συνενώσεις, διαδικασία που το MATLAB επιτρέπει να γίνει σε ένα μόνο βήμα, ωστόσο στην προκειμένη περίπτωση δεν παρέχεται η δυνατότητα ελέγχου και επεξεργασίας των χαρακτηριστικών. Έτσι λοιπόν, κρίθηκε βέλτιστο να πραγματοποιηθεί η διαδικασία σε δυο βήματα, ώστε να γίνει μείωση της ανάλυσης (downsampling) σε κάποιες κατηγορίες σημείων που ανήκουν σε γεωμετρίες που ενδέχεται να επηρεάζουν αρνητικά τις συνενώσεις των νεφών σημείων (όπως είναι τα σημεία με ετικέτα less planar-surface). Παράλληλα προστέθηκαν και περαιτέρω έλεγχοι όπως ο δειγματοληπτικός έλεγχος για απόρριψή του δείγματος αν το σφάλμα της συνένωσης ξεπερνάει ένα συγκεκριμένο όριο. Η παραπάνω διαδικασία υλοποιείται από την συνάρτηση pregisterloam.

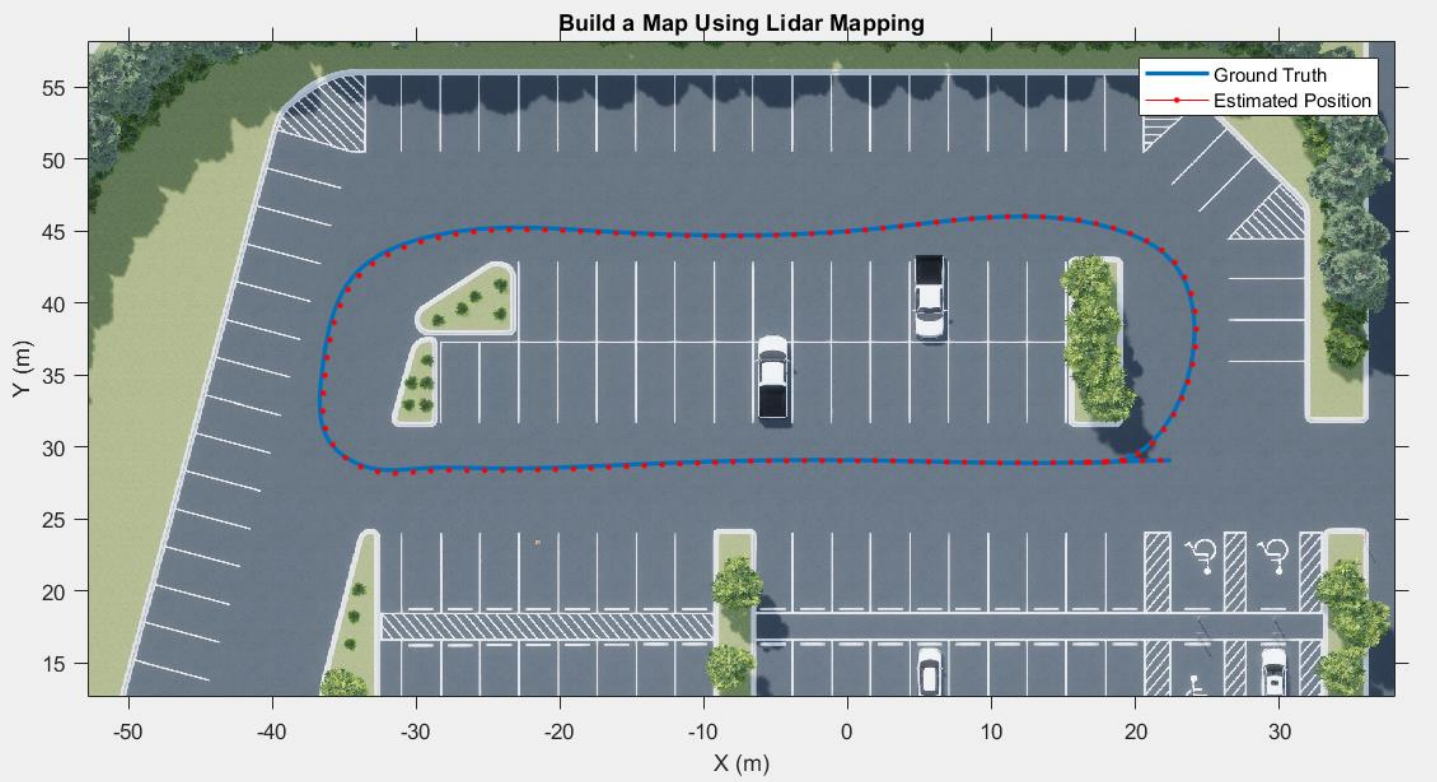
Στη συνέχεια το παράδειγμα επιλύεται αρχικά ως μια LiDAR Οδομετρία κάνοντας τη διαδοχική συνένωση των νεφών σημείων για την δημιουργία του χάρτη και την εκτίμηση τροχιάς. Το αποτέλεσμα της συγκεκριμένης διαδικασίας απεικονίζεται στο Σχήμα 32. Τέλος, πραγματοποιείται βελτιστοποίηση της θέσης κάθε φορά με βάση ένα τοπικό χάρτη (local map) από σαρώσεις, πραγματοποιώντας βελτιστοποίηση για την εύρεση της θέσης που ελαχιστοποιεί το σφάλμα της συνένωσης μέσω της έτοιμης συνάρτησης findPose πριν το συνολικό optimization. Από την δεύτερη διαδικασία προκύπτει το αποτέλεσμα το οποίο απεικονίζεται στο Σχήμα 31.



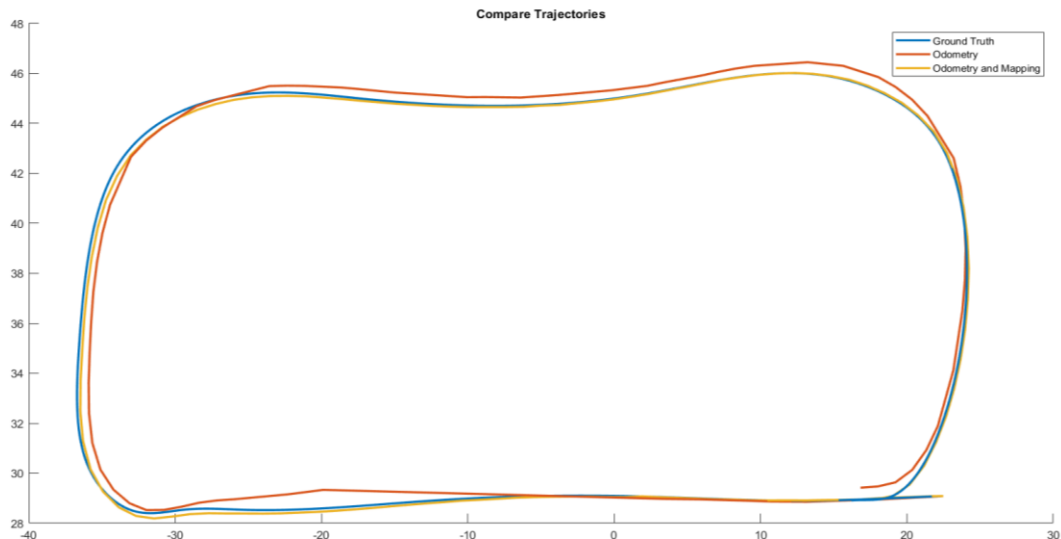
Σχήμα 30: Παράδειγμα απεικόνισης σαρωμένων σημείων (κόκκινο χρώμα) και των τελικά εξαγόμενων χαρακτηριστικών σημείων LOAM από την σάρωση αυτή (πράσινο χρώμα).



Σχήμα 32: Αποτέλεσμα τροχιάς οδομετρικά – παράδειγμα 10 MATLAB



Σχήμα 31: Αποτέλεσμα τροχιάς συνολικά από την μέθοδο LOAM – παράδειγμα 10 MATLAB



Σχήμα 33: Γράφημα σύγκρισης τροχιών – παράδειγμα 10 MATLAB

Το τελικό σφάλμα της τροχιάς με χρήση οδομετρίας στα LOAM χαρακτηριστικά προκύπτει στα 0.87m ενώ μειώνεται σε 0.21m μετά την βελτιστοποίηση. Σημειώνεται επίσης πως δεν υποστηρίζεται η αναζήτηση κλειστών βρόχων στην συγκεκριμένη υλοποίηση SLAM (αντίστοιχα με τον LOAM στον οποίο βασίζεται).

Τέλος πραγματοποιείται σύγκριση του αποτελέσματος του παραδείγματος 10 με αυτό του παραδείγματος 8 με βάση τα σφάλματα τροχιάς τους σε σχέση με το κοινό Ground truth, καθώς χρησιμοποιούν το ίδιο κοινό προσομοιωμένο περιβάλλον. Ωστόσο, καθώς δεν προβλέπεται από τον κώδικα του παραδείγματος 8, δημιουργήθηκε επιπρόσθετος κώδικας (Εικόνα 9) που υπολογίζει τα σφάλματα ώστε να πραγματοποιηθεί η σύγκριση.

To numerically compare the estimated trajectories with the ground truth, compute the root-mean-square error (RMSE) between the ground truth trajectory and the estimated trajectory also compute the root-mean-square error (RMSE) between the ground truth trajectory and the optimized trajectory.

```
% Select the poses to compare
selectedGroundTruth = gTNodes(1:skipFrames:end,:);
% Compute the RMSE
rmseOdometry = rmse(selectedGroundTruth,estNodes,"all");
rmseWithMapping = rmse(selectedGroundTruth,slamAlgPoses,"all");
disp(['RMSE of the trajectory estimated with Odometry: ' num2str(rmseOdometry)])
```

RMSE of the trajectory estimated with Odometry: 0.77463

```
disp(['RMSE of the trajectory estimated after Pose Graph Optimization: ' num2str(rmseWithMapping)])
```

RMSE of the trajectory estimated with Odometry and Mapping: 1.961

Εικόνα 9: Ενσωμάτωση κώδικα για τον υπολογισμό του σφάλματος τροχιάς στο παράδειγμα 8

Τελικά αποτελέσματα των παραδειγμάτων 8 και 10 έχουν ως εξής:

Παράδειγμα 8, Odometry rmse : 0.77 m (συνάρτηση addscan)

Παράδειγμα 10, Odometry rmse : 0.88 m (pcregisterloam)

Παράδειγμα 8, Poses rmse: ύστερα από την βελτιστοποίηση που συμπεριέλαβε τον λανθασμένο βρόχο 1.96 m (συναρτήσεις optimizeposegraph και detectloopclosure)

Παράδειγμα 10, Odometry and Mapping rmse : 0.21 m (Βελτιστοποίηση με βάση τις γεωμετρικές δεσμεύσεις των LOAM Features, συνάρτηση find pose)

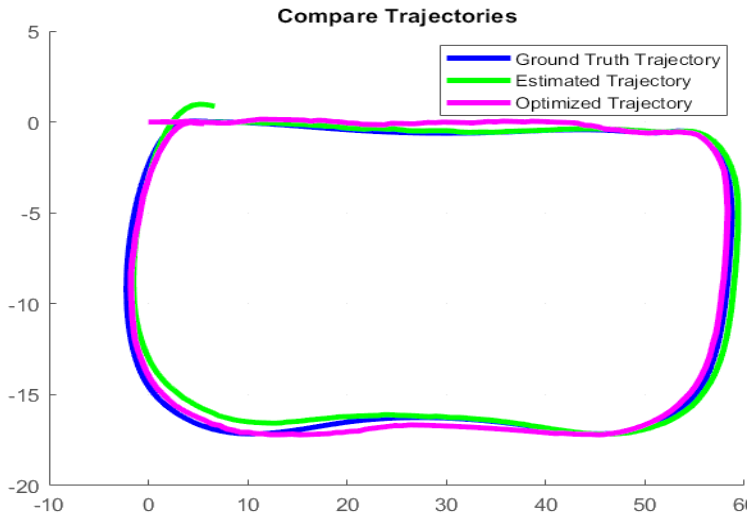
Παρατηρείται πως μόνο με βάση τις διαδοχικές συνενώσεις νεφών (συνάρτηση addScan), η οδομετρία που προέκυψε έδωσε μικρότερο σφάλμα από αυτήν που προέκυψε μέσω της pcregisterloam. Όμως, το τελικό αποτέλεσμα της συνολικής εφαρμογής SLAM του παραδείγματος 8 εμφανίζεται να έχει μεγαλύτερα σφάλματα ακόμη και από την απλή οδομετρία του ίδιου παραδείγματος (1.96m έναντι 0.77m) λόγω του εσφαλμένου βρόχου που επηρέασε τελικά αρνητικά και την βελτιστοποίηση του γράφου θέσης. Αντίθετα, οι βελτιστοποιημένες θέσεις (optimized poses) του LOAM έδωσαν πολύ μικρότερα σφάλμα (0.21m) σε σχέση με την απλή οδομετρία που εφαρμόστηκε αρχικά στο ίδιο παράδειγμα (0.21m έναντι 0.88m), ακόμη και χωρίς την εφαρμογή κάποιας τεχνικής εντοπισμού βρόχων η οποία χρησιμοποιείται στα περισσότερα παραδείγματα του MATLAB και πάρα το γεγονός της ύπαρξης κλειστού βρόχου στα συγκεκριμένα παραδείγματα.

Αξίζει ακόμη να σημειωθεί ότι μεταξύ των παραδειγμάτων 8 και 10, παρόλο που εξετάζεται ο ίδιος προσομοιωμένος χώρος, παρατηρείται διαφορά στον αριθμό των ακίνητων οχημάτων που υπάρχουν στον χώρο. Έτσι πραγματοποιήθηκε πρόσθετος έλεγχος για την επιρροή που ενδεχομένως να προκαλεί η μεταξύ τους διαφορά στο αποτέλεσμα, δοκιμάζοντας τα δυο παραδείγματα και στα δυο σενάρια σταθμευμένων οχημάτων στον προσομοιωμένο χώρο που πραγματοποιείται η μέθοδος SLAM.

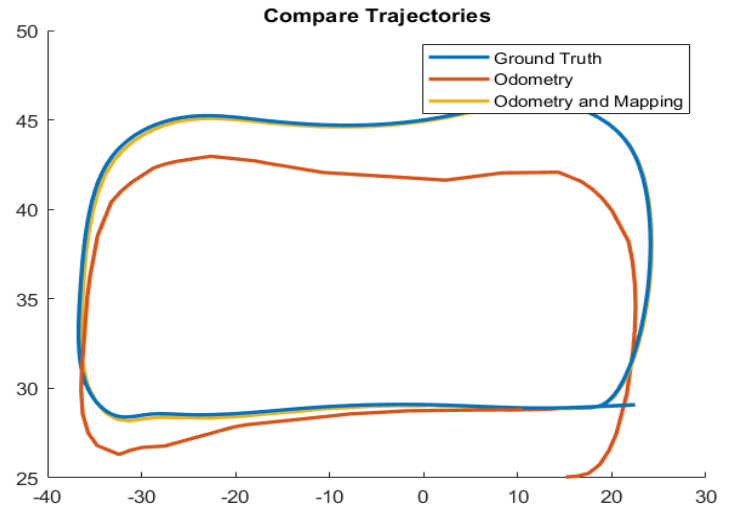
Για την πραγματοποίηση της σύγκρισης έγιναν οι ακόλουθες αλλαγές στον κώδικα, ενώ τα σχετικά αποτελέσματα της τροχιάς που προέκυψαν εμφανίζονται στα Σχήματα 34 και 35.

```
% Set poses of the parked vehicles  
%parkedPoses = data.parkedPoses([18 21],:);  
parkedPoses = data.parkedPoses;
```

```
% Set poses of the parked vehicles  
%parkedPoses = data.parkedPoses;  
parkedPoses = data.parkedPoses([18 21],:);
```



Σχήμα 35: Συγκριτικά αποτελέσματα παραδείγματος 8 σε προσομοίωση με πολλά σταθμευμένα οχήματα.



Σχήμα 34: Συγκριτικά αποτελέσματα παραδείγματος 10 σε προσομοίωση με λίγα σταθμευμένα οχήματα.

Συνολικά σφάλματα 1^{ης} και 2^{ης} δοκιμής παραδειγμάτων 8 και 10 (σε μέτρα):

Παράδειγμα 8, πολλά σταθμευμένα.

RMSE of the trajectory estimated with Odometry: 0.89

RMSE of the trajectory estimated after Pose Graph Optimization: 0.76

Παράδειγμα 10, πολλά σταθμευμένα.

RMSE of the trajectory estimated with Odometry: 0.88

RMSE of the trajectory estimated with Odometry and Mapping: 0.21

Παράδειγμα 8, λίγα σταθμευμένα.

RMSE of the trajectory estimated with Odometry: 0.77

RMSE of the trajectory estimated after Pose Graph Optimization: 1.96

Παράδειγμα 10, λίγα σταθμευμένα.

RMSE of the trajectory estimated with Odometry: 1.90

RMSE of the trajectory estimated with Odometry and Mapping: 0.22

Γενικά παρατηρήθηκε ότι η αύξηση του αριθμού των οχημάτων παρέχει περισσότερα αξιόπιστα σημεία για συνταύτιση βελτιώνοντας έτσι τα αποτελέσματα και στις δυο μεθόδους. Συγκεκριμένα στο παράδειγμα 8 ο αριθμός των οχημάτων δεν φάνηκε να επηρεάζει την απλή οδομετρία, όμως είχε θετική επίδραση στο σφάλμα του αποτελέσματος μετά την βελτιστοποίηση γράφου, καθώς αυτή την

φορά δεν προέκυψε εσφαλμένο κλείσιμο βρόχου, γεγονός το οποίο είχε συντελέσει στη μεγάλη στροφή της τροχιάς. Στο παράδειγμα 10, από την άλλη, παρατηρήθηκε πως ο αυξημένος αριθμός οχημάτων έδωσε αρκετά βελτιωμένο αποτέλεσμα για την απλή οδομετρία, γεγονός που ερμηνεύεται από την εύρεση μεγαλύτερου αριθμού χαρακτηριστικών σημείων LOAM, ενώ το αποτέλεσμα ήταν εξίσου ικανοποιητικό και για τις δυο περιπτώσεις. Τέλος, και στις δυο περιπτώσεις, το παράδειγμα 10 (LOAM) έδωσε καλύτερα αποτελέσματα από αυτά του 8 στο σύνολο της μεθόδου SLAM, με εξαίρεση τις συνενώσεις της απλής οδομετρίας όπου η addScan του παραδείγματος 8 επέφερε καλύτερα αποτελέσματα από την pcregisterloam για το ίδιο περιβάλλον.

Ολοκληρώνοντας λοιπόν τον έλεγχο όλων των παραδειγμάτων LiDAR-SLAM σε περιβάλλον MATLAB, προκύπτουν τα ακόλουθα συμπεράσματα:

Οι προαναφερθείσες μέθοδοι είναι γενικά ανεπαρκείς αφού δεν έχουν την δυνατότητα να λάβουν δεδομένα από σαρωτές σε πραγματικό χρόνο, παράμετρος βασική κατά την εφαρμογή της διαδικασίας SLAM. Επίσης, τα σχετικά παραδείγματα είχαν μη συγκρίσιμους χρόνους εκτέλεσης σε σχέση με τους σύγχρονους αλγόριθμους της βιβλιογραφίας οπότε δεν θεωρήθηκε χρήσιμη η χρονομέτρηση τους. Επιπλέον χρησιμοποιούν ως δεδομένα εισόδου εξειδικευμένες δομές δεδομένων του MATLAB και δεν μπορούν να χρησιμοποιήσουν άμεσα τα δεδομένα άλλων μορφών, στις οποίες εξάγουν τις μετρήσεις τους οι περισσότεροι σαρωτές της αγοράς. Ένα ακόμη μειονέκτημα αποτελεί το γεγονός πως κανένα από τα παραδείγματα δεν χρησιμοποιεί προσαρμοσμένες στα δεδομένα παραμέτρους, με αποτέλεσμα να μην εφαρμόζονται ικανοποιητικά με άμεση εφαρμογή σε άλλα δεδομένα. Η πιο σύγχρονη υλοποίηση που δοκιμάστηκε, προσομοιάζει τον δημοσιευμένο αλγόριθμο LOAM (2014), ο οποίος, όπως έχει αναφερθεί, είναι ο πρώτος σύγχρονος αλγόριθμος LiDAR-SLAM ενώ, εντωμεταξύ, έχουν παρουσιαστεί πολλές άλλες βελτιώσεις του στην βιβλιογραφία. Παρόλ' αυτά, οι νέοι αυτοί αλγόριθμοι δεν έχουν ενσωματωθεί ακόμα στο MATLAB. Τέλος, όσον αφορά την ακρίβεια των επιλύσεων κατά την σύγκριση των παραδειγμάτων 8 και 10, φάνηκε πως οι συμβατικοί αλγόριθμοι του MATLAB (έτοιμες συναρτήσεις συνένωσης νεφών, βελτιστοποίησης γράφων και εύρεσης βρόχων) οι οποίοι εφαρμόστηκαν στο παράδειγμα 8, έδωσε χειρότερα αποτελέσματα σε σύγκριση με τον LOAM από το παράδειγμα 10, με RMS 0.76m έναντι 0.21m (περίπου 3,5 φορές μεγαλύτερο και χρόνο εκτέλεσης 257 sec έναντι 115 sec για καταγραφή δεδομένων 60 sec).

Γίνεται λοιπόν σαφές πως η ερευνητική κοινότητα δεν έχει υλοποιήσει ακόμη αλγόριθμους SLAM στο MATLAB επειδή και το ίδιο έχει ακόμη πολύ δρόμο στο να υποστηρίξει κατάλληλα σύγχρονες τεχνικές SLAM για υλοποίηση εφαρμογών σε πραγματικό χρόνο και ειδικά αυτών που αφορούν 3D δεδομένα από συστήματα LiDAR.

4.2 Πρακτική εφαρμογή σε σύγχρονους αλγορίθμους

Εφόσον δεν βρέθηκε κάποιος διαθέσιμος αλγόριθμος σε γλώσσα προγραμματισμού MATLAB που να επιτρέπει την άμεση σύνδεση με τον διαθέσιμο σαρωτή LIVOX AVIA, έγινε αρχικά αναζήτηση αλγορίθμων LiDAR-SLAM με ανοικτό κώδικα που να υποστηρίζονται από το λειτουργικό σύστημα Windows. Γενικά, τα δημοσιευμένα άρθρα δεν έκαναν αναφορά στο λειτουργικό σύστημα υλοποίησης, οπότε έγινε ο αντίστοιχος έλεγχος των αλγορίθμων στο GitHub. Τελικά διαπιστώθηκε, πως επί του παρόντος, μόνο ο αλγόριθμος KISS-ICP αποτελεί σύγχρονο και αξιόπιστο αλγόριθμο SLAM που μπορεί να λειτουργήσει στα Windows μιας και είναι υλοποιημένος σε Python και σε περιβάλλον ROS (ROBOT OPERATING SYSTEM) [97]. Αξίζει να σημειωθεί ότι η συντριπτική πλειοψηφία των σχετικών τεχνικών υποστηρίζονται μόνο από συγκεκριμένες εκδόσεις των Linux-Ubuntu. Το ROS αποτελεί ανοιχτό λογισμικό το οποίο παρέχει βιβλιοθήκες/εργαλεία για τη δημιουργία και τον έλεγχο κάθε είδους ρομποτικών συστημάτων και αισθητήρων. Επιπλέον, διευκολύνει την ανάπτυξη λογισμικού για ρομπότ, παρέχοντας μια δομημένη προσέγγιση για τη διαχείριση των λογισμικών εφαρμογών, την επικοινωνία μεταξύ των διάφορων λογισμικών ενοτήτων και την επικοινωνία με το υλικό (hardware) διάφορων συσκευών. Το ROS έχει ενσωματώσει πληθώρα βιβλιοθηκών και εργαλείων για την ανάπτυξη σύγχρονων αλγορίθμων SLAM, ενώ μπορεί να συνδεθεί, να καταγράψει και να διαβάσει δεδομένα σε πραγματικό χρόνο από κάθε σαρωτή LiDAR υπό την μορφή των αρχείων rosbag. Επιπλέον μπορεί να διαβάσει και να ενσωματώσει συγχρονισμένες μετρήσεις και από άλλους αισθητήρες όπως τα IMU και τα GNSS, ενώ αποτελεί και την βασική πλατφόρμα υλοποίησης λογισμικού ελέγχου ρομπότ, UAV και αυτόνομων οχημάτων. Σήμερα, το σύνολο σχεδόν των διαθέσιμων αλγορίθμων SLAM βασίζονται στην πλατφόρμα ROS σε συνδυασμό με το λειτουργικό σύστημα Linux Ubuntu, ενώ ως γλώσσα προγραμματισμού χρησιμοποιούν σχεδόν αποκλειστικά την C++.

Για αυτούς τους λόγους έγινε διερεύνηση αλγορίθμων SLAM υλοποιημένων σε γλώσσα προγραμματισμού C++ που βασίζονται στο ROS και είναι συμβατοί με αισθητήρες της εταιρείας LIVOX. Έτσι, πέραν του KISS ICP [86], επιλέχθηκαν οι αλγόριθμοι LOAM LIVOX [98], LIVOX MAPPING [99] (πρακτικά υλοποίηση του LOAM), FAST-LIO2 [50] και Faster-LIO [51]. Οι συγκεκριμένες υλοποιήσεις δήλωναν ρητά την συμβατότητα με τους σαρωτές της LIVOX καθώς ο τρόπος σάρωσης και το είδος του σαρωτή (solid state ή mechanical) παίζει καθοριστικό ρόλο σε διαδικασίες όπως η διόρθωση των σφαλμάτων (θόλωση) λόγω κίνησης (motion blur). Επίσης το εύρος του σαρωτή αλλά και το οπτικό του πεδίο αποτελούν βασικούς παράγοντες για τον τρόπο εξαγωγής των χαρακτηριστικών σημείων. Τέλος, πολύ μεγάλο ρόλο παίζει φυσικά η μορφή των δεδομένων LiDAR που χρησιμοποιεί ως είσοδο (input) ο αλγόριθμος.

Όπως αναφέρεται και στην βιβλιογραφία, σύμφωνα με τις παρούσες συνθήκες, καμία μέθοδος 3D LiDAR-SLAM δεν είναι απαλλαγμένη από την ανάγκη ρύθμισης παραμέτρων ώστε να προσαρμόζεται αυτόματα σε διαφορετικά σενάρια, χρησιμοποιώντας διαφορετικούς αισθητήρες LiDAR, υποστηρίζοντας διαφορετικά προφίλ κίνησης και συνεπώς, διαφορετικούς τύπους πλατφορμών μεταφοράς όπως τα UAV, τα αυτοκίνητα και τα ρομπότ [51].

Παρακάτω, ακολουθεί μια σύντομη περιγραφή των βασικών χαρακτηριστικών των αλγορίθμων αυτών.

Αρχικά ο KISS-ICP αποτελεί έναν σύγχρονο αλγόριθμο που πραγματοποιεί ταυτόχρονη χαρτογράφηση και εύρεση της θέσης πολύ αποτελεσματικά. Ωστόσο, αποτελεί κυρίως μέθοδο Οδομετρίας LiDAR, επειδή δεν χρησιμοποιεί κάποια τεχνική βελτιστοποίησης γράφου, ούτε μέθοδο βέλτιστης εκτίμησης μέσω φιλτραρίσματος αλλά ούτε και μέθοδο εντοπισμού κλειστών βρόχων.

Ο αλγόριθμος KISS-ICP αποτελείται από τέσσερα βασικά βήματα. Στο πρώτο βήμα γίνεται αρχικά η διόρθωση της σάρωσης από θόλωση λόγω κίνησης (motion blur) ή αλλιώς deskewing μέσω μιας υπόθεσης μοντέλου σταθερής κίνησης και μίας παρεμβολής μεταξύ των θέσεων σύμφωνα με την συχνότητα καταγραφής των σαρώσεων. Ύστερα ακολουθεί μια διπλή δειγματοληπτική μείωση του νέφους (subsampling) με βάση τα voxel. Η πρώτη χρησιμοποιείται για να ενημερώσει τον τοπικό χάρτη που έχει προκύψει από την συνένωση προηγούμενων νεφών σημείων, ενώ η δεύτερη χρησιμοποιείται ακολούθως για τον υπολογισμό των παραμέτρων στερεού σώματος μέσω του ICP. Στο επόμενο βήμα χρησιμοποιείται το αραιό νέφος και μετασχηματίζεται μέσω ενός αντίστροφου μοντέλου σταθερής κίνησης και παρεμβολής. Τέλος, στο τελικό βήμα, πραγματοποιείται ο αλγόριθμος ICP για τον εντοπισμό του μετασχηματισμού στερεού σώματος, όχι μεταξύ δυο διαδοχικών σαρώσεων αλλά μεταξύ του νέου νέφους και του τοπικού χάρτη. Στην τελική φάση χρησιμοποιείται παράλληλα ένα είδος αποκλεισμού εσφαλμένων τιμών (outlier) με γεωμετρικό τρόπο. Μέσω αυτής της διαδικασίας ολοκληρώνεται πρακτικά ο αλγόριθμος του ICP για μια σάρωση, ενώ η ίδια διαδικασία επαναλαμβάνεται για όλες τις επόμενες σαρώσεις.

Η μέθοδος LOAM LIVOX αποτελεί αλγόριθμο graph-based ο οποίος πραγματοποιεί αρχικά μια εξαγωγή χαρακτηριστικών ακμών και επιπέδων περιοχών (LOAM features), με βάση την ένταση, το βάθος, και την γωνία πρόσπτωσης. Προηγούμενως για την διόρθωση της σάρωσης από το motion blur, ο αλγόριθμος χωρίζει κάθε σάρωση σε τρία μέρη λαμβάνοντας υπόψιν τον τρόπο σάρωσης του αισθητήρα και το μοντέλο σταθερής κίνησης, πραγματοποιώντας διαφορετική παρεμβολή για κάθε τμήμα της σάρωσης. Επίσης γίνεται χρήση μιας επαναληπτικής βελτιστοποίησης για την αφαίρεση τυχόν κινουμένων αντικειμένων αφαιρώντας το 20% των σημείων που εμφάνισαν μεγαλύτερο σφάλμα κατά την τοπική βελτιστοποίηση.

Τέλος, πραγματοποιείται η συνένωση των νεφών μέσω μιας τεχνικής που προσπαθεί να συνδέσει τα δυο νέφη μέσω της ελαχιστοποίησης του σφάλματος μεταξύ της ταύτισης των LOAM features (παρόμοια με τον G-ICP), ενώ η διαδικασία ολοκληρώνεται με εφαρμογή συνολικής βελτιστοποίησης των θέσεων μέσω Γράφου. Παρόλο που δεν αναφέρεται στο σχετικό άρθρο [100], ο κώδικας παρέχει την δυνατότητα εντοπισμού κλειστών βρόχων η οποία βασίζεται στο άρθρο [101] και η μπορεί να λειτουργήσει σε όλα τα LiDAR-SLAM τα οποία βασίζονται σε LOAM features.

Ο αλγόριθμος LIVOX MAPPING αποτελεί ουσιαστικά μια παραλλαγή του LOAM με παραμέτρους προσαρμοσμένες στους αισθητήρες LIVOX. Ουσιαστικά, ακολουθεί πλήρως την λογική των αλγορίθμων LOAM και LOAM LIVOX, αν εξαιρέσουμε τις παραλλαγές που αναφέρθηκαν παραπάνω και αφορούν την απαλοιφή κινουμένων αντικειμένων, την ενσωμάτωση κλεισίματος βρόχου και το ενισχυμένο μοντέλο κίνησης για την διόρθωση του θολώματος λόγω κίνησης (motion blur).

Ο FAST-LIO2 αποτελεί αλγόριθμο filter-based ο οποίος εισάγει μια νέα δομή αποθήκευσης δεδομένων για επίτρεψη των επιμέρους διαδικασιών, την ikd tree. Όσον αφορά την αλγοριθμική διαδικασία που υλοποιείται, ο συγκεκριμένος αλγόριθμος χρησιμοποιεί μετρήσεις IMU αυστηρά συνδεδεμένες με τις μετρήσεις LiDAR. Συγκεκριμένα, οι μετρήσεις IMU χρησιμοποιούνται αρχικά για την εκτίμηση της κίνησης και της θέσης ώστε να διορθωθεί το motion blur της εκάστοτε σάρωσης. Ύστερα η μέτρηση του IMU και της σάρωσης συνδυάζονται μέσω ενός επαναληπτικού φίλτρου Kalman (iEKF) ώστε να δώσουν την βέλτιστη εκτίμηση θέσης, ενώ έπειτα χρησιμοποιείται ο ICP για την συνένωση της σάρωσης με τον συνολικό χάρτη. Τέλος, η μέθοδος δεν υποστηρίζει έλεγχο κλειστών βρόχων.

Ο Faster-LIO αποτελεί αλγόριθμο filter-based ο οποίος βασίζεται στον FAST-LIO2. Διαφοροποιείται από τον πρώτο, καθώς αντί για την δομή ikd tree αναπτύσσει μια νέα δομή δεδομένων, την iVOX, που αποτελεί μια εκδοχή των voxel με ενισχυμένες ιδιότητες και η οποία μπορεί να επισπεύσει ακόμη περισσότερο τις διαδικασίες αναζήτησης και ενημέρωσης που επιτελούνται. Ακόμη, εκτός από την δομή αποθήκευσης, διαφέρει στην λογική διαχείρισης και δημιουργίας του χάρτη, χρησιμοποιώντας διαρκώς τον συνολικό χάρτη, σε αντίθεση με τον FAST-LIO2 ο οποίος χρησιμοποιεί μια μέθοδο αποκοπής απομακρυσμένων σημείων σε σχέση με την θέση της τρέχουσας σάρωσης. Ωστόσο και οι δυο αλγόριθμοι διατηρούν την ίδια βασική λογική των αλγορίθμων LIO που περιλαμβάνει δειγματοληψία του νέφους, διόρθωση του motion blur και χρήση φίλτρου iEKF για τον συνδυασμό των μετρήσεων από το IMU με τον αλγόριθμο ICP.

Ύστερα από την επιλογή των αλγορίθμων LiDAR-SLAM οι οποίοι αναλύθηκαν παραπάνω, πραγματοποιήθηκε η πειραματική δοκιμή τους με δεδομένα από τον φορητό 3D LiDAR αισθητήρα LIVOX AVIA.



Εικόνα 10: Ο solid state LiDAR σαρωτής με ενσωματωμένο IMU, LIVOX AVIA [95]

Ο LIVOX AVIA είναι ένας μικρός φορητός αισθητήρας Lidar του οποίου το μέγεθος και βάρος επιτρέπει είτε την εύκολη χρήση του με το χέρι, είτε την ενσωμάτωσή του σε μικρά UAV και ρομπότ, καθώς και σε μεγαλύτερες πλατφόρμες όπως ένα όχημα. Καμία του διάσταση δεν ξεπερνάει τα 10cm και ζυγίζει λιγότερο από 500gr. Αποτελεί αισθητήρα LiDAR στέρεας κατάστασης (solid state) και δεν χρησιμοποιεί μηχανικά μέρη για την περιστροφή του αισθητήρα κατά την σάρωση.

Το εύρος σάρωσης κυμαίνεται από το 1m έως τα 450m, ενώ το οπτικό του πεδίο είναι 70.4° οριζόντια και 77.2° κάθετα στην μη επαναληπτική λειτουργία σάρωσης και 70.4° οριζόντια και 4.5° οριζόντια σε περιπτώσεις επαναλαμβανομένων σαρώσεων σε μια γραμμή.

Επίσης, διαθέτει ενσωματωμένο αισθητήρα IMU ο οποίος μπορεί να υπολογίζει τις στροφές και τις μετατοπίσεις του αισθητήρα. Η ονομαστική ακρίβεια μέτρησης απόστασης του αισθητήρα είναι 2cm, ενώ η ονομαστική ακρίβεια μετρήσεις γωνιών είναι μικρότερη από 0.05°. Επιπλέον, υποστηρίζει δυνατότητα καταγραφής πολλαπλών επιστροφών (έως 3 επιστροφές), ενώ ανά επιστροφή μπορεί να καταγραφεί έως και 240.000 σημεία το δευτερόλεπτο. Στον Πίνακα 3 παρουσιάζονται αναλυτικά όλα τα χαρακτηριστικά του αισθητήρα σύμφωνα με την εταιρεία Livox [95].

Πίνακας 3: Τεχνικά χαρακτηριστικά (specs) σαρωτή LIVOX AVIA [95]

Model	AVIA
Laser Wavelength	905 nm
Laser Safety	Class 1 (IEC60825-1:2014)(Eye Safety)
Detection Range (@ 100 klx)	190 m @ 10% reflectivity 230 m @ 20% reflectivity 320 m @ 80% reflectivity
Detection Range (@ 0 klx)	190 m @ 10% reflectivity 260 m @ 20% reflectivity 450 m @ 80% reflectivity
FOV	Non-repetitive scanning pattern: 70.4° (Horizontal) ×77.2° (Vertical) Repetitive line scanning: 70.4° (Horizontal) ×4.5° (Vertical)
Range Precision (1σ @ 20m)	2 cm ¹
Angular Precision (1σ)	< 0.05°
Beam Divergence	0.28° (Vertical) × 0.03° (Horizontal)
Point Rate	240,000 points/s (first or strongest return) 480,000 points/s (dual return) 720,000 points/s (triple return)
Data Latency	≤ 2 ms
Data Port	100 Mbps Ethernet
Data synchronization:	IEEE 1588-2008 (PTPv2), PPS (Pulse Per Second), GPS (PPS+UTC)
False Alarm Rate (@ 100 klx) ²	< 0.0003%
IMU	Built-in model: BMI088
Operating Temperature	-4°F to 149°F (-20°C to 65°C)
IP Rating ³	IP67
Power ⁴	Repetitive scanning pattern: 9 W (Startup: 16W) Non-repetitive: 8 W (Startup: 16W)
Power Supply Voltage Range ⁵	10 ~ 15 V DC (with Converter 2.0: 9~30V DC)
Noise	40cm omnidirectional <45 dBA
Dimensions	91×61.2×64.8 mm
Weight	498 g (without cables)

Ο αισθητήρας LIVOX συνδέεται με τα υπολογιστικά συστήματα μέσω κάρτας δικτύου και μπορεί να αποθηκεύει δεδομένα 3D σε διαφορετικού τύπου αρχεία όπως *rosbag*, *lx*, *las* και *csv*, ενώ δεν υποστηρίζει τα αρχεία *ply* και *pcd*.

Για τους σκοπούς της εργασίας πραγματοποιήθηκε εφαρμογή σε πραγματικό χρόνο των αλγορίθμων οι οποίοι αναφέρθηκαν προηγουμένως, με την επικοινωνία αλγορίθμου και σαρωτή να πραγματοποιείται μέσω του συστήματος ROS με καταγραφή των δεδομένων σε μορφή *rosbag*. Επιπλέον, πραγματοποιήθηκαν καταγραφές για επεξεργασία εκ των υστέρων, σε όλες τις μορφές αποθήκευσης που επιτρέπει ο αισθητήρας.

Τα πρώτα πειράματα πραγματοποιήθηκαν σε μικρό εσωτερικό χώρο (δωμάτιο κατοικίας) διαστάσεων περίπου 5m x 3m. Ωστόσο, επειδή ο αισθητήρας δεν μπορεί να καταγράψει αποστάσεις μικρότερες του ενός μέτρου και επειδή από το 1m έως τα 5m το πεδίο σάρωσης του είναι αρκετά περιορισμένο, οι σαρώσεις επικεντρώθηκαν στο μισό δωμάτιο (μιας και ο αισθητήρας αυτός προορίζεται κυρίως για εφαρμογές σε εξωτερικούς χώρους ή μεγάλους εσωτερικούς χώρους).

Τα εξωτερικά πειράματα πραγματοποιήθηκαν κυρίως στον χώρο του άλσους σκοπευτηρίου Καισαριανής, αν και αρχικά είχε επιλεγεί το Άλσος Παγκρατίου. Το τελευταίο απορρίφθηκε μετά από δοκιμές, καθώς λόγω της πυκνής βλάστησης, οι λεπτομέρειες των συνενωμένων 3D νεφών σημείων που προέκυψαν ήταν αρκετά δυσδιάκριτες. Τέλος, όλες οι λήψεις των σαρώσεων πραγματοποιήθηκαν σε συχνότητα 10 Hz, ενώ οι μετρήσεις IMU σε 200 Hz (τυπική συχνότητα καταγραφής). Επισημαίνεται επίσης πως σε όλες τις δοκιμές η μετακίνηση του αισθητήρα γινόταν με το χέρι (*handheld*). Ως μέθοδος καταγραφής σημείων χρησιμοποιήθηκε αυτή της ισχυρότερης επιστροφής.

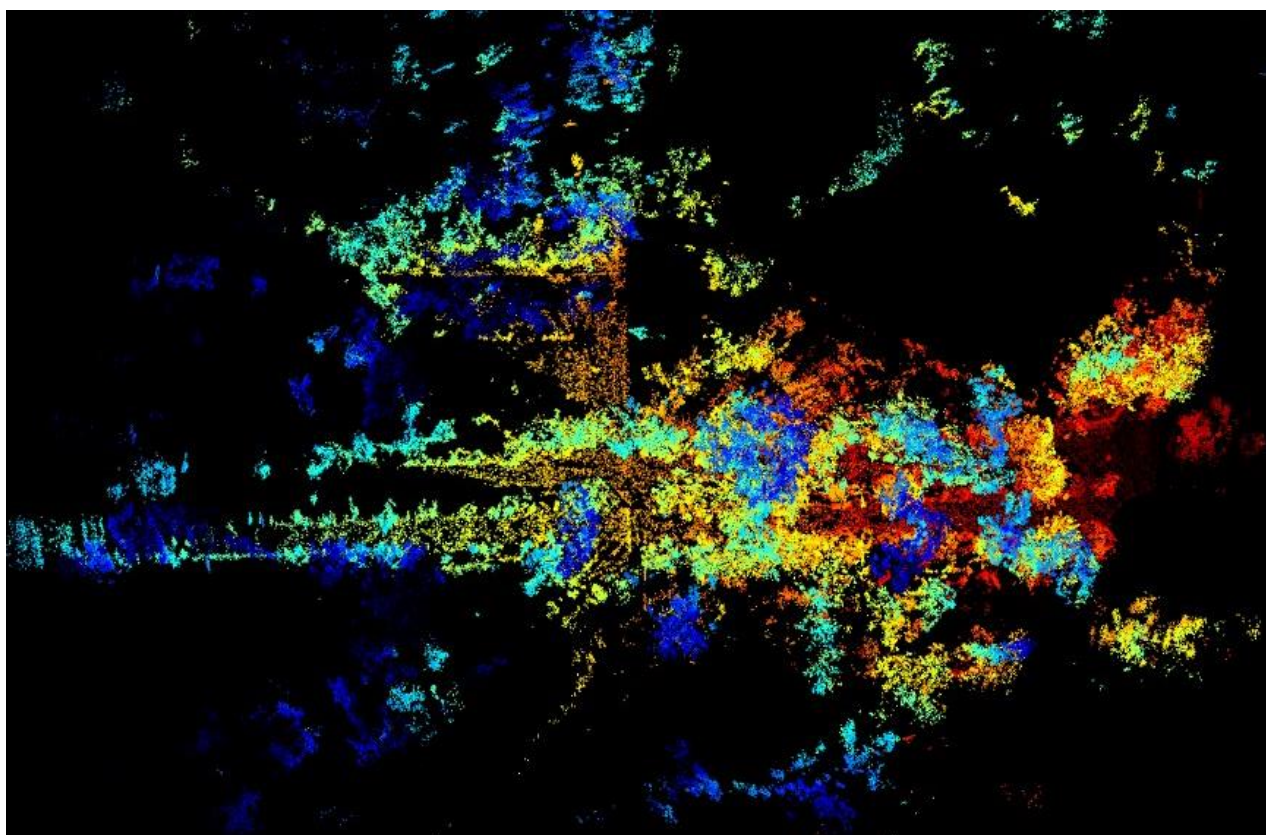
Οι εξωτερικές περιοχές μελέτης απεικονίζονται παρακάτω στις Εικόνες 11-13 ενώ η εσωτερική περιοχή μελέτης των δοκίμων απεικονίζεται στην εικόνα 14.



Εικόνα 11: Κάτοψη της βασικής outdoor περιοχής μελέτης.



Εικόνα 12: Κάτοψη της ευρύτερης περιοχής μελέτης των εξωτερικών δοκίμων



Εικόνα 13: Κάτοψη αραιού χάρτη της αρχικής περιοχής μελέτης (Άλσους Παγκρατίου)



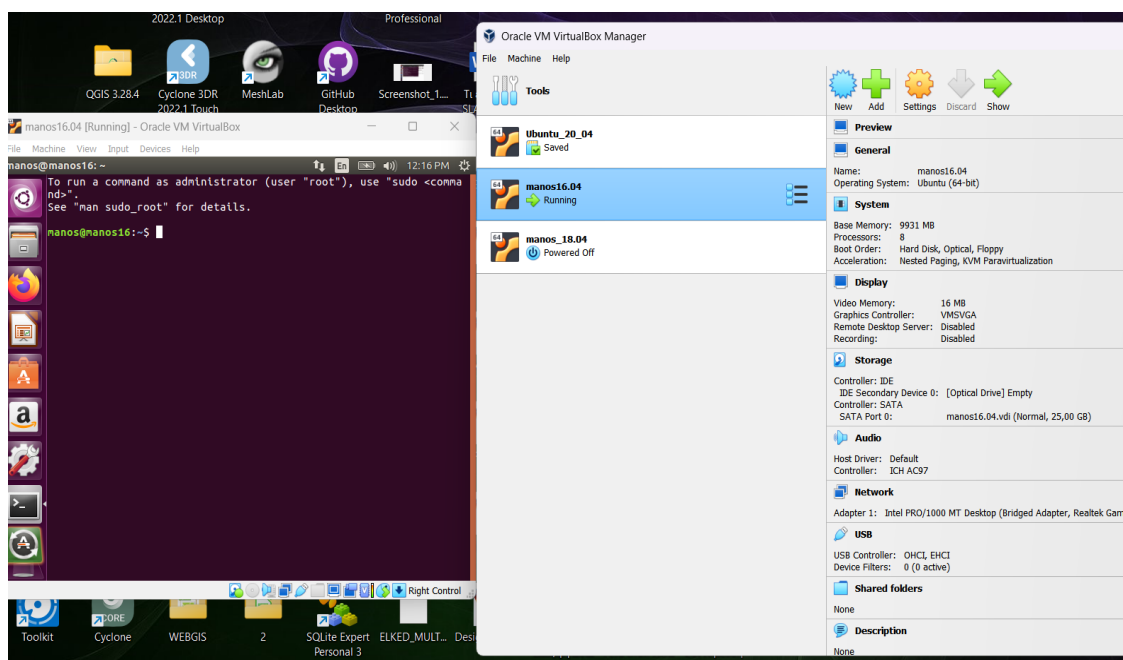
Εικόνα 14: Περιοχή μελέτης εσωτερικού χώρου (indoor)

Όσο για το υλικό και λογισμικό που χρησιμοποιήθηκε για την υλοποίηση των δοκιμών, καθώς όλοι οι αλγόριθμοι (με εξαίρεση τον KISS ICP) απαιτούσαν λειτουργικό σύστημα Linux Ubuntu και χρήση του λογισμικού ROS, έγινε χρήση του λογισμικού Virtual Box [102].

Το Virtual Box αποτελεί μια ελεύθερη και ανοικτού κώδικα εφαρμογή εικονικού περιβάλλοντος, η οποία παρέχει την δυνατότητα να εκτελεστούν εναλλακτικά λειτουργικά συστήματα (εντός ενός εικονικού περιβάλλοντος), πάνω στο ήδη υπάρχον λειτουργικό σύστημα του υπολογιστή. Αυτό καθίσταται δυνατό μέσω της δημιουργίας εικονικών μηχανών (VMs), δηλαδή ενός εικονικού περιβάλλοντος που προσομοιώνει το λογισμικό, τον υπολογιστικό χώρο και τους πόρους ενός φυσικού υπολογιστή. Μέσω αυτής της τεχνολογίας καθίσταται δυνατή η ανεξάρτητη εγκατάσταση λειτουργικού συστήματος, γεγονός που δίνει την δυνατότητα στους χρήστες να εκτελούν διαφορετικά λειτουργικά συστήματα σε έναν μόνο φυσικό υπολογιστή. Εξαιτίας των θετικών αυτών χαρακτηριστικών, οι εικονικές μηχανές χρησιμοποιούνται ευρέως σε δοκιμές λογισμικού καθώς προσφέρουν ένα ασφαλές και απομονωμένο σύστημα ξεχωριστό από το βασικό σύστημα του υπολογιστή. Μπορούν ακόμη να διαμορφωθούν με επιλεγμένους υπολογιστικούς πόρους κάθε φορά, ενώ μπορούν να δοκιμαστούν σε αυτές διαφορετικά περιβάλλοντα χωρίς την ανάγκη επιβάρυνσης του βασικού συστήματος ή την χρήση πολλαπλών φυσικών υπολογιστών. Τέλος, μπορούν πρακτικά να υλοποιηθούν και να αποθηκευτούν σε οποιονδήποτε χώρο ως απλός φάκελος, με αποτέλεσμα η ίδια εικονική μηχανή με τα δεδομένα της να μπορεί να υλοποιηθεί ακόμη και μέσα σε ένα USB stick ή έναν σκληρό δίσκο όπως και να εκτελεστεί από διαφορετικούς φυσικούς υπολογιστές.

Για τους παραπάνω λόγους κρίνεται ως ένα πολύ χρήσιμο εργαλείο για την δοκιμή αλγορίθμων LiDAR-SLAM, καθώς αρχικά οι περισσότεροι χρήστες χρησιμοποιούν το λειτουργικό σύστημα Windows ενώ οι αλγόριθμοι απαιτούν υποχρεωτικά χρήση Linux Ubuntu. Επιπλέον, στην πλειοψηφία των περιπτώσεων οι αλγόριθμοι είναι σχεδιασμένοι να εκτελούνται με συγκεκριμένη έκδοση Ubuntu, ROS και βιβλιοθηκών, με αποτέλεσμα να απαιτείται η φυσική εγκατάσταση πολλαπλών λειτουργικών για την χρήση διαφορετικών τεχνικών, η οποία καθίσταται αρκετά περιπλοκή. Αντιθέτως μέσα από την χρήση των VM, μπορεί κάθε φορά να υλοποιηθεί το προτεινόμενο λογισμικό υπόβαθρο για την εκτέλεση του επιθυμητού αλγορίθμου. Έτσι, στα πλαίσια αυτής της διπλωματικής εργασίας υλοποιήθηκαν διάφορες εικονικές μηχανές Linux Ubuntu για την εκτέλεση και δοκιμή των αλγορίθμων, ωστόσο οι δοκιμές αυτές επικεντρώθηκαν στην έκδοση Ubuntu 20.04 καθώς οι παλιότερες εκδόσεις σταδιακά υποστηρίζονται λιγότερο και εμφανίζουν σημαντικά προβλήματα με αποτέλεσμα οι κώδικες να μην μπορούν να εκτελεστούν.

Οι μηχανές αυτές υλοποιήθηκαν σε σκληρό δίσκο HDD με δυνατότητα χρήσης 10 Gigabyte μνήμης RAM και 8 πυρήνων επεξεργαστή AMD RYZEN 5800H του φυσικού υπολογιστή, ενώ δόθηκε και η άδεια στην μηχανή να επικοινωνεί με την κάρτα δικτύου Ethernet του υπολογιστή ώστε να μπορεί να συνδεθεί ο αισθητήρας με την εικονική μηχανή. Τέλος, δόθηκε και η δυνατότητα επικοινωνίας μεταξύ αρχείων του φυσικού υπολογιστή και της εκάστοτε εικονικής μηχανής για την εύκολη ανταλλαγή δεδομένων όπως για παράδειγμα των παραγομένων νεφών, ολοκληρώνοντας έτσι την απαραίτητη διαμόρφωση λογισμικού για την εκτέλεση των αλγορίθμων.



Εικόνα 15: Στιγμιότυπο από το περιβάλλον μιας VM και του περιβάλλοντος του Virtual Box.

Αφού επιλέξει κατάλληλο λογισμικό, ξεκίνησε η διαδικασία της υλοποίησης. Αρχικά εγκαταστάθηκε η έκδοση ROS η οποία υποστηρίζεται από την έκδοση των Ubuntu (στην έκδοση 20.04 συγκεκριμένα αντιστοιχεί η Noetic), ενώ πραγματοποιήθηκε η εγκατάσταση των οδηγών (drivers) για την σύνδεση του αισθητήρα LIVOX σε ROS [103]. Ακολούθησε το βήμα που αφορά την εγκατάσταση των απαραίτητων βιβλιοθηκών και του κώδικα. Ωστόσο, όσον αφορά τον αλγόριθμο LOAM LIVOX, παρατηρήθηκαν τα εξής:

Αρχικά ο κώδικας έχει δημοσιευτεί το έτος 2019 και από τότε δεν έχει ανεβεί κάποια ανανεωμένη έκδοση. Πέντε χρόνια αποτελούν τεράστιο χρονικό διάστημα στον χώρο των υπολογιστών και της ρομποτικής με αποτέλεσμα να έχουν μεσολαβήσει πολλές αλλαγές στα λειτουργικά συστήματα και τα εν λόγω λογισμικά.

1. Prerequisites

1.1 Ubuntu and ROS

Ubuntu 64-bit 16.04 or 18.04. ROS Kinetic or Melodic. [ROS Installation](#) and its additional ROS package:

```
sudo apt-get install ros-XXX-cv-bridge ros-XXX-tf ros-XXX-message-filters ros-XXX-image-transport
```

NOTICE: remember to replace "XXX" on above command as your ROS distributions, for example, if your use ROS-kinetic, the command should be:

```
sudo apt-get install ros-kinetic-cv-bridge ros-kinetic-tf ros-kinetic-message-filters ros-kinetic-i
```

1.2. Ceres Solver

Follow [Ceres Installation](#).

1.3. PCL

Follow [PCL Installation](#).

NOTICE: Recently, we find that the point cloud output from the voxelgrid filter vary from PCL 1.7 and 1.9, and PCL 1.7 leads some failure in some of our examples ([issue #28](#)). By this, we strongly recommend you to use update your PCL as version 1.9 if you are using the lower version.

Εικόνα 16: Στιγμιότυπο από τις απαιτούμενες βιβλιοθήκες του αλγορίθμου LOAM LIVOX [100]

Παρατηρήθηκε λοιπόν ότι ο κώδικας απαιτούσε έκδοση Ubuntu 16.04 ή 18.04 με τις αντίστοιχα συνδεδεμένες εκδόσεις ROS Kinetic ή Melodic ενώ την παρούσα χρονική στιγμή της συγγραφής είχαν ήδη κυκλοφορήσει οι εκδόσεις 20.04, 22.04 και 23.10 [104]. Επιπλέον θέτει ως προαπαιτούμενες τις βιβλιοθήκες Ceres Solver [105] (βιβλιοθήκη που προσφέρει εργαλεία βελτιστοποίησης και επίλυσης εξισώσεων) και PCL [106] (βιβλιοθήκη που προσφέρει εργαλεία για την διαχείριση νεφών σημείων, όπως ανάγνωση, οπτικοποίηση αλλά και συνένωση). Ωστόσο δεν γίνεται αναφορά στην ακριβή έκδοση της Ceres Solver (εικόνα 16), ενώ και η ίδια για να λειτουργήσει χρειάζεται άλλες προαπαιτούμενες βιβλιοθήκες.

Θα μπορούσε κανείς να εξάγει το συμπέρασμα ότι οι δημιουργοί του αλγορίθμου εννοούν τις τελευταίες εκδόσεις ή ότι λειτουργούν σε όλες, ωστόσο κάτι τέτοιο δεν ίσχυε καθώς οι βιβλιοθήκες αυτές απαιτούν με την σειρά τους επιπρόσθετες βιβλιοθήκες οι οποίες ενδέχεται να έχουν αλλάξουν σημαντικά. Ως αποτέλεσμα, οι βιβλιοθήκες αυτές πολλές φορές δεν υποστηρίζονται ή εμφανίζουν σφάλματα, οπότε οι κώδικες SLAM υλοποιημένοι σε C++ καταλήγουν να αποτελούν ένα δίκτυο από προαπαιτούμενες βιβλιοθήκες το οποίο σε περίπτωση που δεν ενημερώνεται συχνά, δεν μπορεί να λειτουργήσει.

Γίνεται έτσι σαφές πως χρηστικός κώδικας LiDAR-SLAM υλοποιημένος σε τέτοια μορφή χωρίς τακτή ενημέρωση από τους προγραμματιστές δεν μπορεί να υπάρξει. Αντίστοιχη εικόνα με τον αλγόριθμο LOAM LIVOX παρατηρήθηκε και από τον LIVOX MAPPING ο οποίος κυκλοφόρησε το 2020 [99] και ύστερα από εκείνη την χρονιά δεν έγιναν άλλες αλλαγές.

Σε αντίθεση, ο κώδικας FAST-LIO2 [107], μετά την του κυκλοφορία το 2021, ενημερώνεται συχνά (τελευταία ενημέρωση μέσα στο 2023), ενώ παράλληλα σχεδόν όλα τα αιτήματα προβλήματος (issues) που είχαν αναρτηθεί από τους χρήστες απαντήθηκαν από τους προγραμματιστές (4 ανοιχτά, 285 κλειστά). Τέλος, παρέχονται πολύ αναλυτικές οδηγίες για την εγκατάσταση των βιβλιοθηκών και των εκδόσεων της. Αντίστοιχη εικόνα με του FAST-LIO2 παρουσιάζει και η σελίδα του αλγορίθμου Faster-LIO [108], ο οποίος κυκλοφόρησε το 2022 με την τελευταία ενημέρωση να πραγματοποιείται το 2023. Παράλληλα, παρέχει και αυτή αναλυτικές και ενημερωμένες οδηγίες σχετικά με την σωστή εγκατάσταση.

Όσον αναφορά τον κώδικα του KISS-ICP [109], (κυκλοφόρησε το 2022), αναφέρεται η συμβατότητα με τα λειτουργικά συστήματα Windows, Linux και MacOS ενώ η χρήση της Python αντί της C++, κάνει τον αλγόριθμο πιο προσιτό σε μη εξειδικευμένους χρήστες και δεν απαιτεί αλληπάλληλες εγκαταστάσεις βιβλιοθηκών της C++ , επίσης αποτελεί τον αλγόριθμο με τις περισσότερες ενημερώσεις με την τελευταία ενημέρωση να έχει πραγματοποιηθεί το 2024. Τέλος, μπορεί να λειτουργήσει είτε μέσω χρήσης του λογισμικού ROS, είτε όχι καθιστώντας τον έτσι τον πιο εύκολα υλοποιήσιμο αλγόριθμο εκ των 5.

Τα παραπάνω είχαν ως αποτέλεσμα οι αλγόριθμοι Faster-LIO, FAST-LIO2 και KISS ICP να μπορέσουν να εγκατασταθούν και να χρησιμοποιηθούν πιο εύκολα για τα πειράματα. Αντίθετα, για την εγκατάσταση και την εκτέλεση των LOAM LIVOX και LIVOX MAPPING, πραγματοποιήθηκε εκτενής αναζήτηση διάφορων βιβλιοθηκών και ανάγνωση πολλών προβλημάτων και συζητήσεων άλλων προγραμματιστών οι οποίοι ασχολούνται με το LiDAR-SLAM και την C++, με αποτέλεσμα ύστερα από την πτησπάνω έρευνα και τις γνώσεις που αποκτήθηκαν να γίνουν οι κατάλληλες αλλαγές στον κώδικα ώστε να γίνουν συμβατές με την έκδοση Ubuntu 20.04, την αντίστοιχη έκδοση ROS που υποστηρίζει αυτήν την έκδοση (Noetic) και τις πρόσφατες σχετικές βιβλιοθήκες. Ωστόσο δεν κρίνεται χρήσιμη η περαιτέρω ανάλυση της συγκεκριμένης τεχνικής διαδικασίας και έτσι θα ακολουθήσει η παρουσίαση των δοκίμων που πραγματοποιήθηκαν.

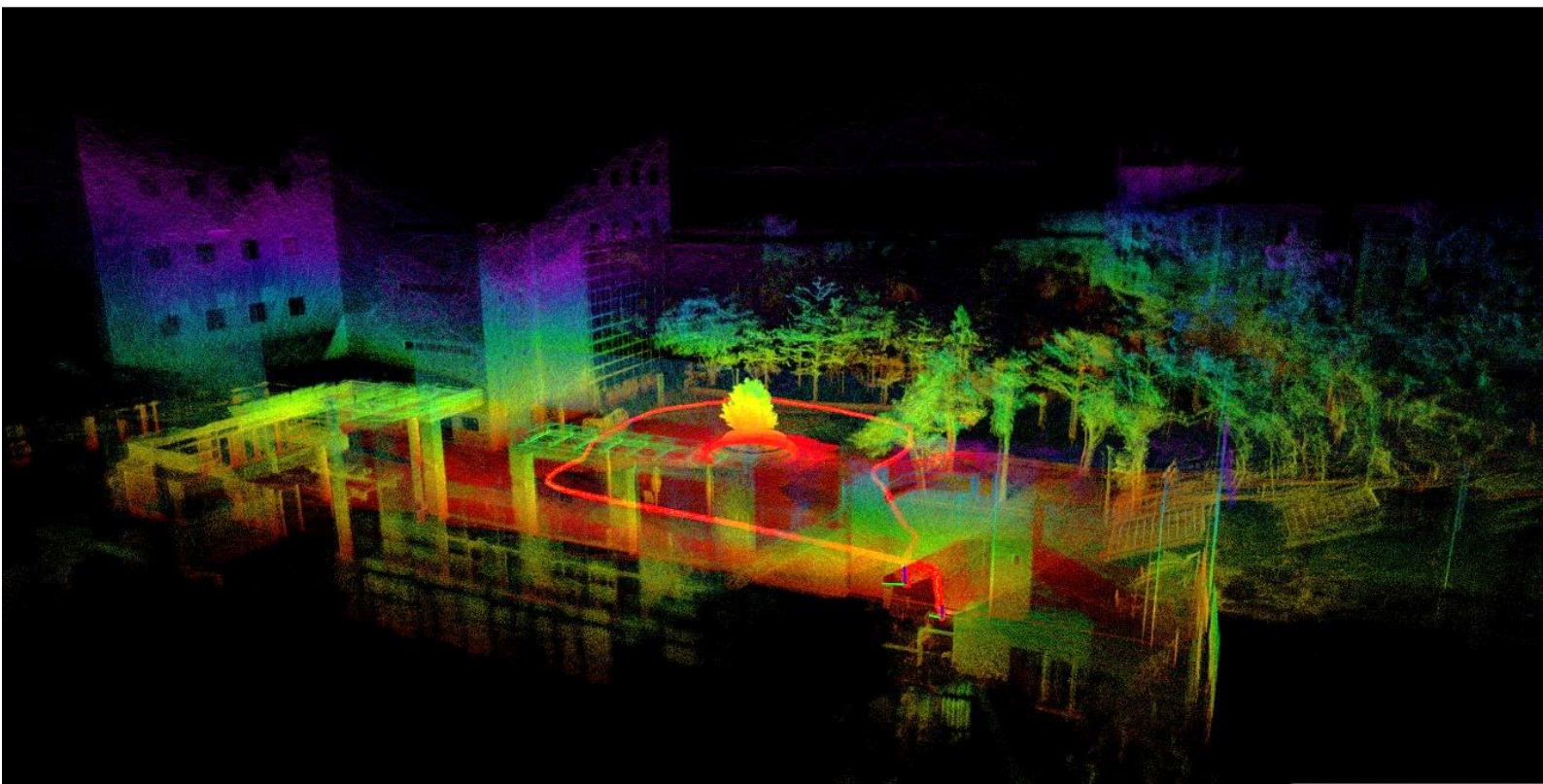
4.2.1 Εφαρμογή με δεδομένα ελέγχου (Test Data)

Ύστερα από την ολοκλήρωση της εγκατάστασης του λογισμικού και των αλγορίθμων, πραγματοποιήθηκαν οι πρώτες δοκιμές για να διαπιστωθεί η ορθή λειτουργία τους. Για τον λόγο αυτό, έγινε χρήση κάποιων ανοιχτών δεδομένων ελέγχου (test data) από σαρωτές LIVOX που παρέχονται από τους δημιουργούς των LOAM LIVOX και FAST-LIO2, ωστόσο για τα δεδομένα αυτά δεν παρέχονται αληθή δεδομένα (ground truth).

Τα δεδομένα που χρησιμοποιήθηκαν κατά τον έλεγχο είναι το CYT_02 σε μορφή rosbag και τύπου Pointcloud2, το οποίο αποτελεί κύριο δεδομένο εισόδου στους αλγορίθμους LOAM LIVOX, LIVOX MAPPING και KISS ICP. Αφορά καταγραφή 229s και δεν περιλαμβάνει μετρήσεις IMU. Επιπλέον, χρησιμοποιήθηκε το dataset outdoor mainbuilding 10 Hz το οποίο είναι τύπου rosbag με σαρώσεις σε μορφή CustomMsg, το οποίο αφορά καταγραφή 141s και περιλαμβάνει μετρήσεις IMU για χρήση στις μεθόδους FAST-LIO2 και Faster-LIO.

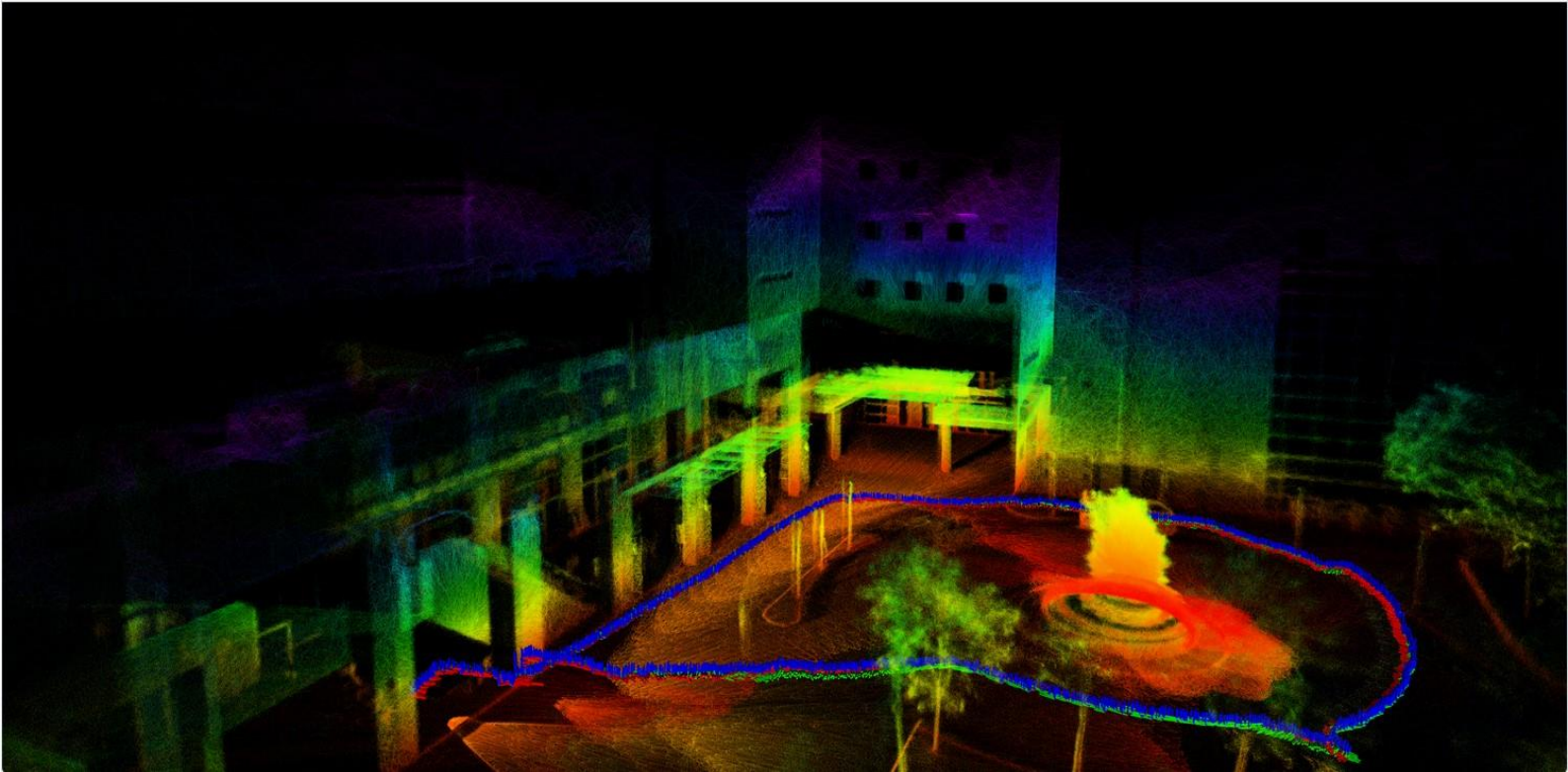
Έτσι πραγματοποιήθηκε η πρώτη δοκιμή πάνω στο CYT_02 με τον αλγόριθμο LOAM LIVOX και το αποτέλεσμα να παρουσιάζεται στην Εικόνα 17. Ωστόσο χρειάστηκαν 310sec για να ολοκληρωθεί ο αλγόριθμος και συνεπώς η συγκεκριμένη εφαρμογή δεν έτρεξε σε πραγματικό χρόνο.

rosbag.rviz* - RViz

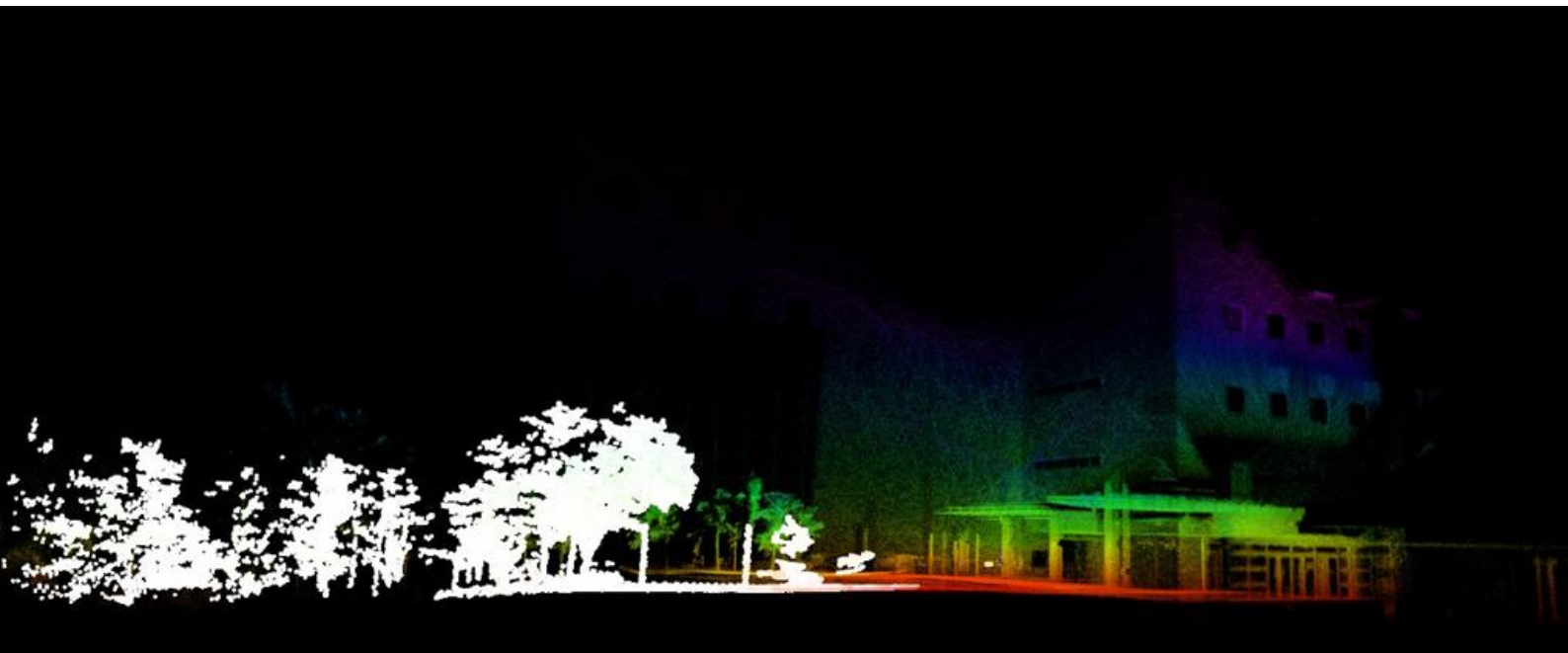


Εικόνα 17: Αποτέλεσμα αλγορίθμου LOAM LIVOX με βάση το test dataset CYT_02

Το αποτέλεσμα ωστόσο δεν φάνηκε να έχει διαφορές από αυτό που παρουσιάζουν οι προγραμματιστές του αλγορίθμου (εικόνα 18) εξετάζοντάς το οπτικά, καθώς δεν παρατηρήθηκαν παραμορφώσεις ή διπλοπροβολές (συνήθη αποτελέσματα όταν υπάρχουν μεγάλες στροφές).

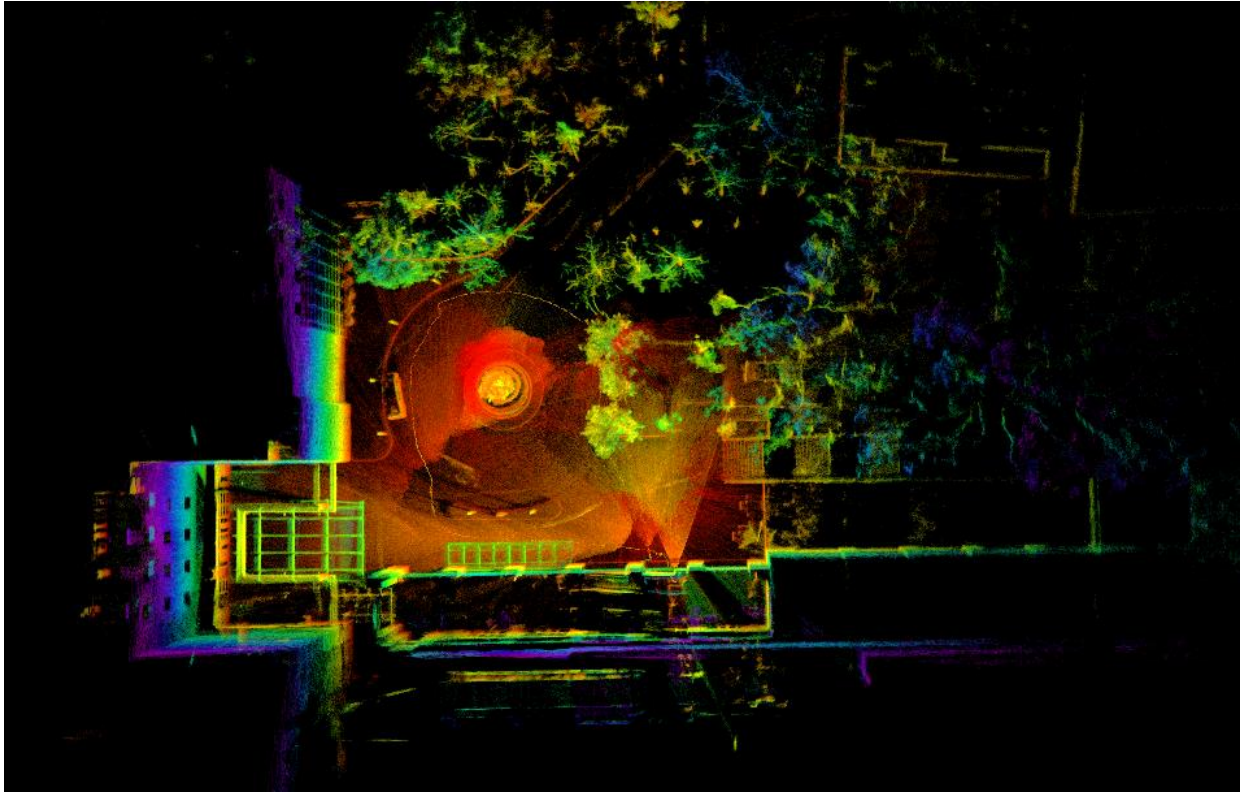


Εικόνα 18: Εικόνα αποτελέσματος αλγορίθμου LOAM LIVOX που παρέχεται από τον δημιουργό για το test dataset CYT_02 [100]

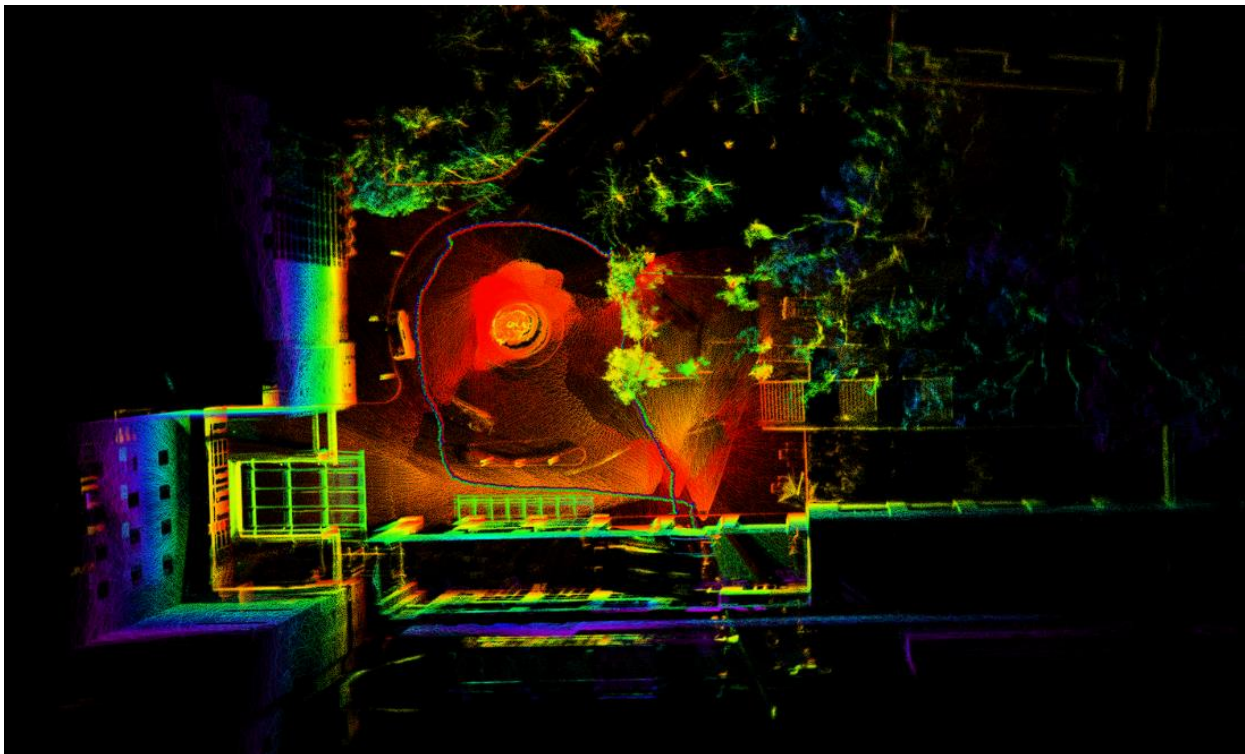


Εικόνα 19: Στιγμιότυπο από την εκτέλεση του αλγορίθμου LOAM LIVOX (εμφανίζονται ως λεύκα τα νέα σημεία σάρωσης)

Στη συνέχεια πραγματοποιήθηκε δοκιμή με τον LIVOX MAPPING με τις βασικές του ρυθμίσεις. Για τα αποτελέσματα απαιτήθηκαν 249sec γεγονός που σημαίνει ότι ο αλγόριθμος έτρεξε σχεδόν σε πραγματικό χρόνο. Επίσης το τελικό αποτέλεσμα από την εφαρμογή του (Εικόνα 20), φαίνεται πανομοιότυπο με το αντίστοιχο δημοσιευμένο από την ίδια την ομάδα του LOAM LIVOX (Εικόνα 21).

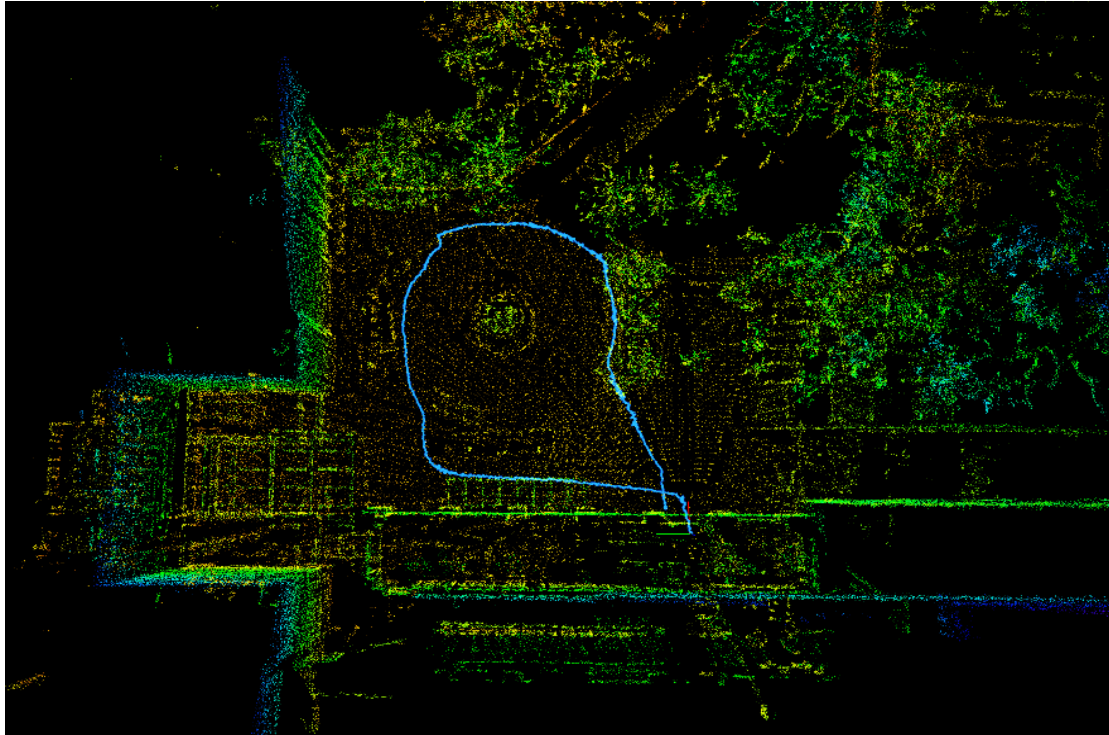


Εικόνα 20: Αποτέλεσμα αλγορίθμου LIVOX MAPPING για το dataset CYT_02



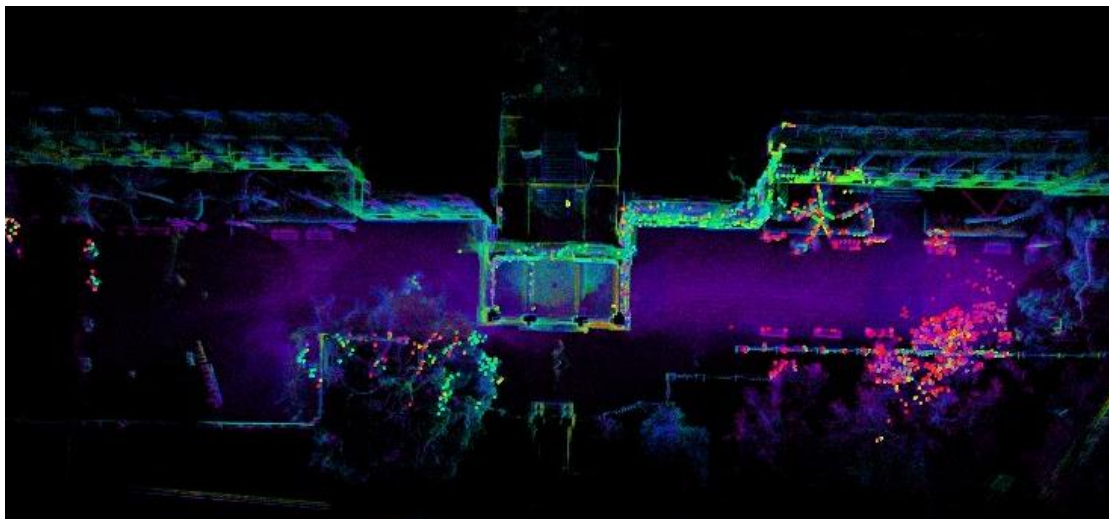
Εικόνα 21: Εικόνα αποτελέσματος που παρέχεται από τον δημιουργό για το test dataset CYT_02 [100]

Τέλος πραγματοποιήθηκε δοκιμή με τον αλγόριθμο KISS ICP στις βασικές του ρυθμίσεις, με την εκτέλεση να διαρκεί 229sec (πραγματικό χρόνο). Το αποτέλεσμα (Εικόνα 22) είναι πολύ αραιότερο από αυτό των LOAM LIVOX και LIVOX MAPPING, ενώ παρέχεται η δυνατότητα για εξαγωγή μόνο των θέσεων και όχι του χάρτη.



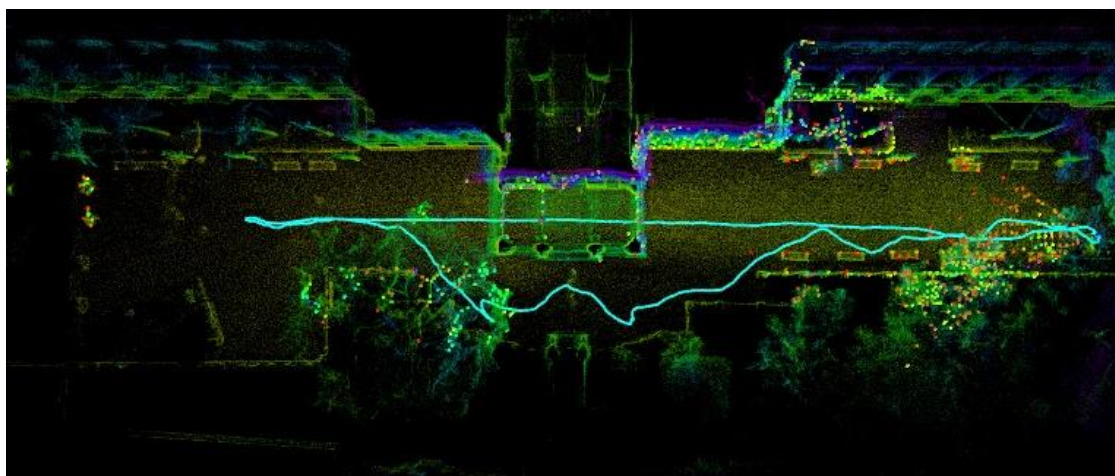
Εικόνα 22: Αποτέλεσμα αλγορίθμου KISS-ICP για το dataset CYT_02

Το σετ δεδομένων CYT_02 δεν μπορούσε να δοκιμαστεί με τους αλγόριθμους Faster-LIO και FAST-LIO2, καθώς οι συγκεκριμένοι χρησιμοποιούν rosbag μορφής CustomMsg και απαιτούν και μετρήσεις IMU. Έτσι πραγματοποιήθηκε έλεγχος με το δεύτερο σύνολο δεδομένων (outdoor mainbuilding), το οποίο πληρούσε τις απαιτούμενες προϋποθέσεις. Αρχικά λοιπόν πραγματοποιήθηκε δοκιμή του FAST-LIO2 στις προεπιλεγμένες ρυθμίσεις, με τον αλγόριθμο να απαιτεί για την ολοκλήρωσή του χρόνο ίσο με 141sec (πραγματικός χρόνος).



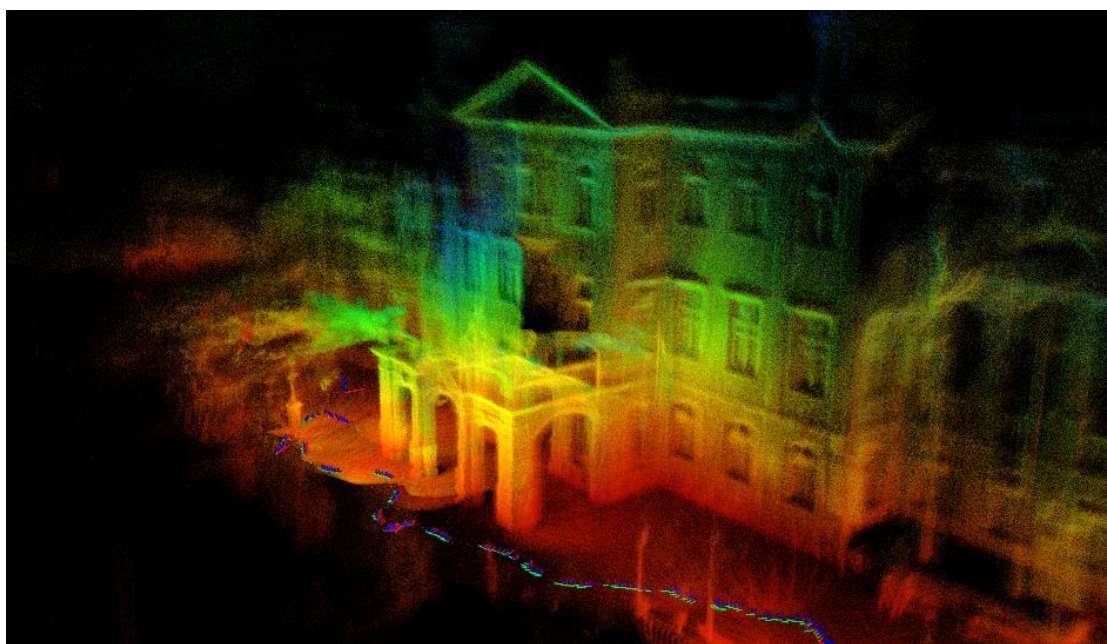
Εικόνα 23: Αποτέλεσμα αλγορίθμου FAST-LIO2 για το dataset outdoor mainbuilding

Ακολούθως, πραγματοποιήθηκε δοκιμή με τον Faster-LIO στις προεπιλεγμένες ρυθμίσεις. Ο αλγόριθμος έτρεξε σε χρόνο ίσο με αυτόν της καταγραφής, ενώ το αποτέλεσμα (Εικόνα 24) εμφανίζεται αρκετά καθαρό χωρίς ιδιαίτερο θόρυβο, πολλαπλές προβολές ή παραμορφώσεις.

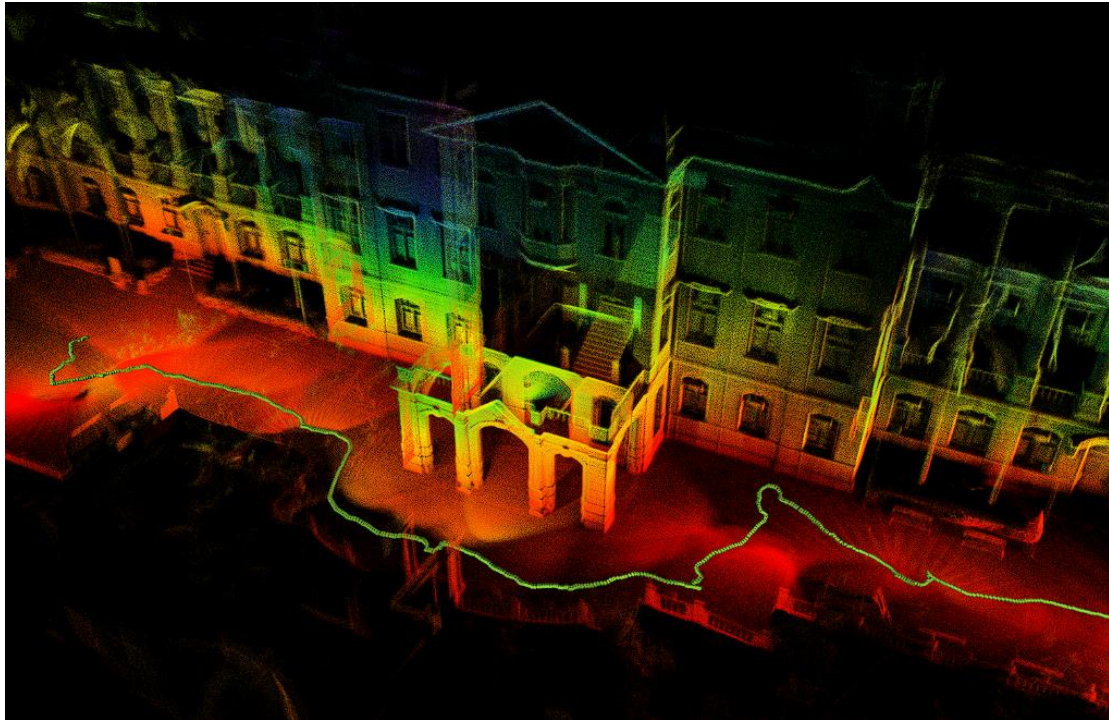


Εικόνα 24: Αποτέλεσμα αλγορίθμου Faster-LIO για το test dataset outdoor mainbuilding

Επίσης, το συγκεκριμένο σετ δοκιμάστηκε και για τον LIVOX MAPPING ο οποίος μπορεί να χρησιμοποιήσει και τις δυο μορφές αρχείων bags (CustomMsg και Pointcloud2). Σε αυτήν την περίπτωση, το αποτέλεσμα (Εικόνα 25), αν και πιο πυκνό, εμφανίζει σαφή θόρυβο και στροφές με αποτέλεσμα να παρατηρούνται θολές περιοχές. Αξίζει να σημειωθεί ότι ο αλγόριθμος δεν κατάφερε να ολοκληρωθεί σε πραγματικό χρόνο καθώς απαιτήθηκαν 183 sec. Ωστόσο, αν και ο συγκεκριμένος αλγόριθμος λειτούργησε κανονικά για το αρχείο CYT_02, τα προβλήματα της συγκεκριμένης δοκιμής πρέπει να οφείλονται στη χρήση διαφορετικού σαρωτή και στην μεγαλύτερη δυσκολία αποτύπωσης του συγκεκριμένου χώρου (ακανόνιστη κίνηση κατά την σάρωση).



Εικόνα 25: Αποτέλεσμα αλγορίθμου LIVOX MAPPING για το dataset outdoor mainbuilding



Εικόνα 26: Εικόνα αποτελέσματος από τον δημιουργό για το outdoor mainbuilding [107]

Αφού λοιπόν ολοκληρωθήκαν τα πρωταρχικά τεστ ελέγχου λειτουργίας των αλγορίθμων, ξεκίνησαν να πραγματοποιούνται τα πειράματα στα δεδομένα που αποκτήθηκαν στο πλαίσιο της διπλωματικής από τον αισθητήρα LIVOX AVIA.

Στη συνέχεια ακολούθησε η λήψη των πειραματικών δεδομένων μέσω του αισθητήρα LIVOX AVIA του εργαστηρίου. Συγκεκριμένα, όπως έχει αναφερθεί και προηγουμένως, έγινε μια πρώτη προσπάθεια αποτύπωσης στο Άλσος Παγκρατίου, ωστόσο παρατηρήθηκε ότι λόγω της αρκετά πυκνής βλάστησης οι λεπτομέρειες των νεφών σημείων δεν ήταν άμεσα ευκρινείς (εικόνα 13). Για αυτό το λόγο επιλέχτηκε ως περιοχή μελέτης το Σκοπευτήριο Καισαριανής.

Καθώς δεν είναι δυνατό όλοι οι αλγόριθμοι να τρέξουν με την ίδια μορφή δεδομένων, οι καταγραφές που πραγματοποιήθηκαν σε πραγματικό χρόνο χωρίστηκαν σε datasets μορφής rosbag CustomMsg με μετρήσεις IMU για τον έλεγχο των (Faster-LIO, FAST-LIO2, LIVOX MAPPING) και σε Pointcloud2 χωρίς IMU (LOAM LIVOX, KISS ICP).

Για τις λήψεις αυτές πραγματοποιήθηκε σύνδεση του αισθητήρα με μπαταρία 12V (Sunlight 12V 2.3Ah) και μικρό φορητό υπολογιστή (Εικόνας 27) ώστε να μπορούν γίνουν μετρήσεις με το χέρι σχετικά εύκολα (το συνολικό βάρος του υπολογιστή, της μπαταρίας και του αισθητήρα δεν ξεπερνούσε τα τρία κιλά). Τέλος, επισημαίνεται πως η αυτονομία της πλήρους φορτισμένης μπαταρίας έφτανε την μια ώρα.

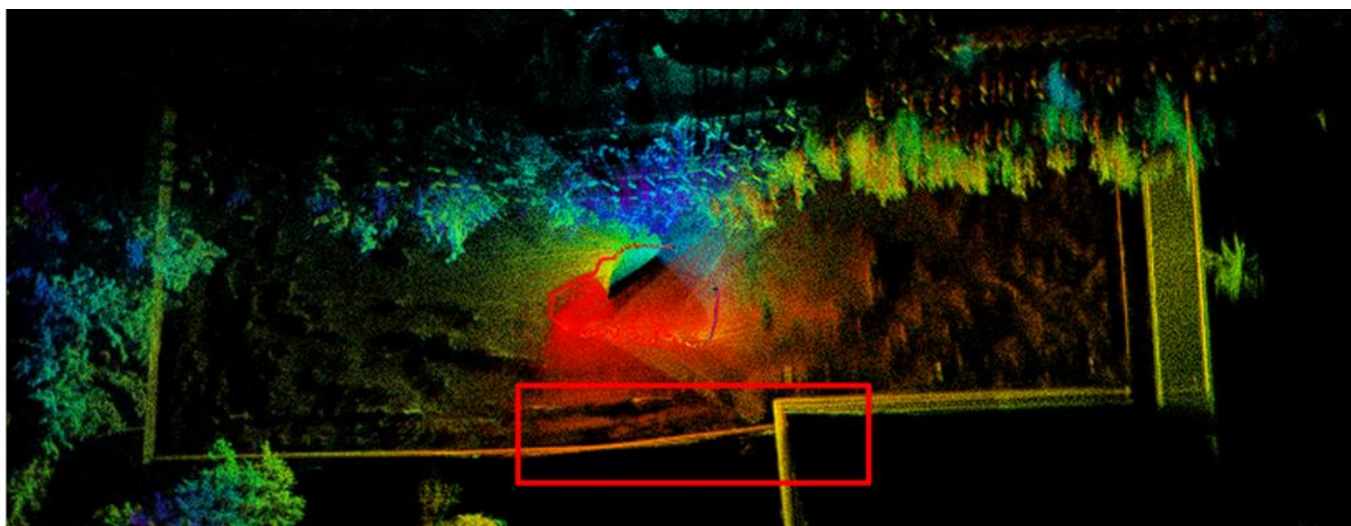


Εικόνα 27: Διάταξη εφαρμογής (Σαρωτής LIVOX AVIA με ενσωματωμένο IMU, μπαταριά, φορητός υπολογιστής)

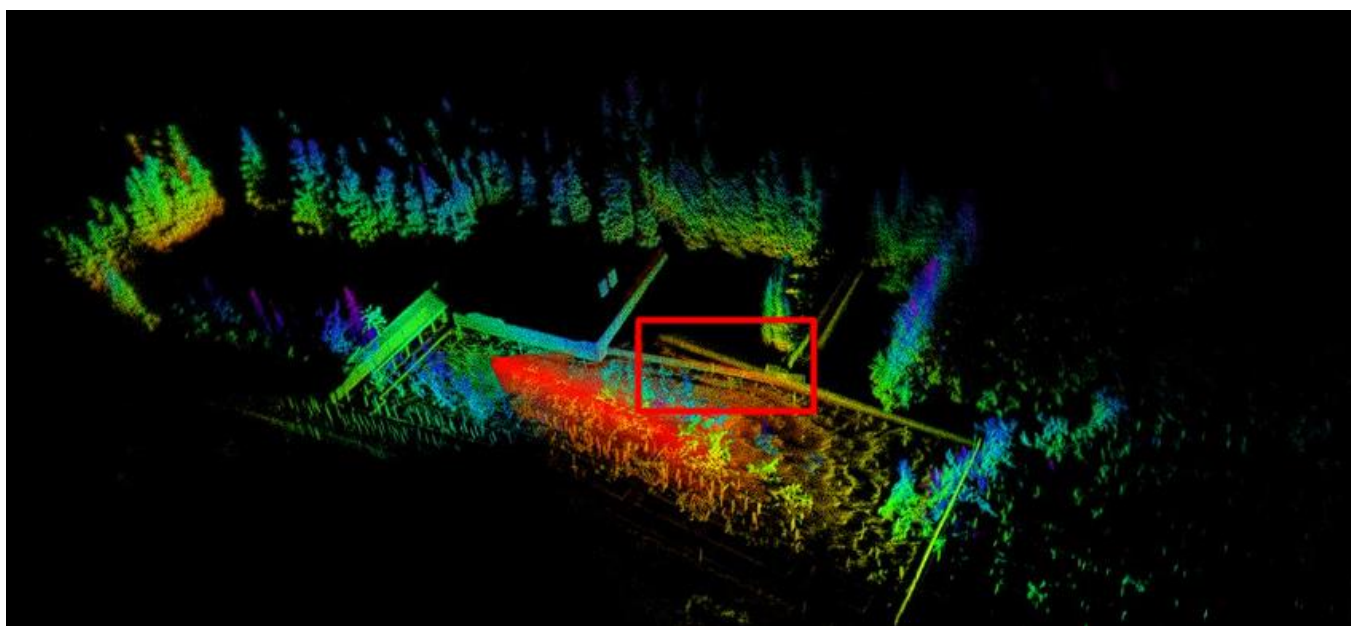
4.2.2 Πρακτική εφαρμογή σε νέα δεδομένα εξωτερικού χώρου

4.2.2.1 Εφαρμογή LOAM LIVOX

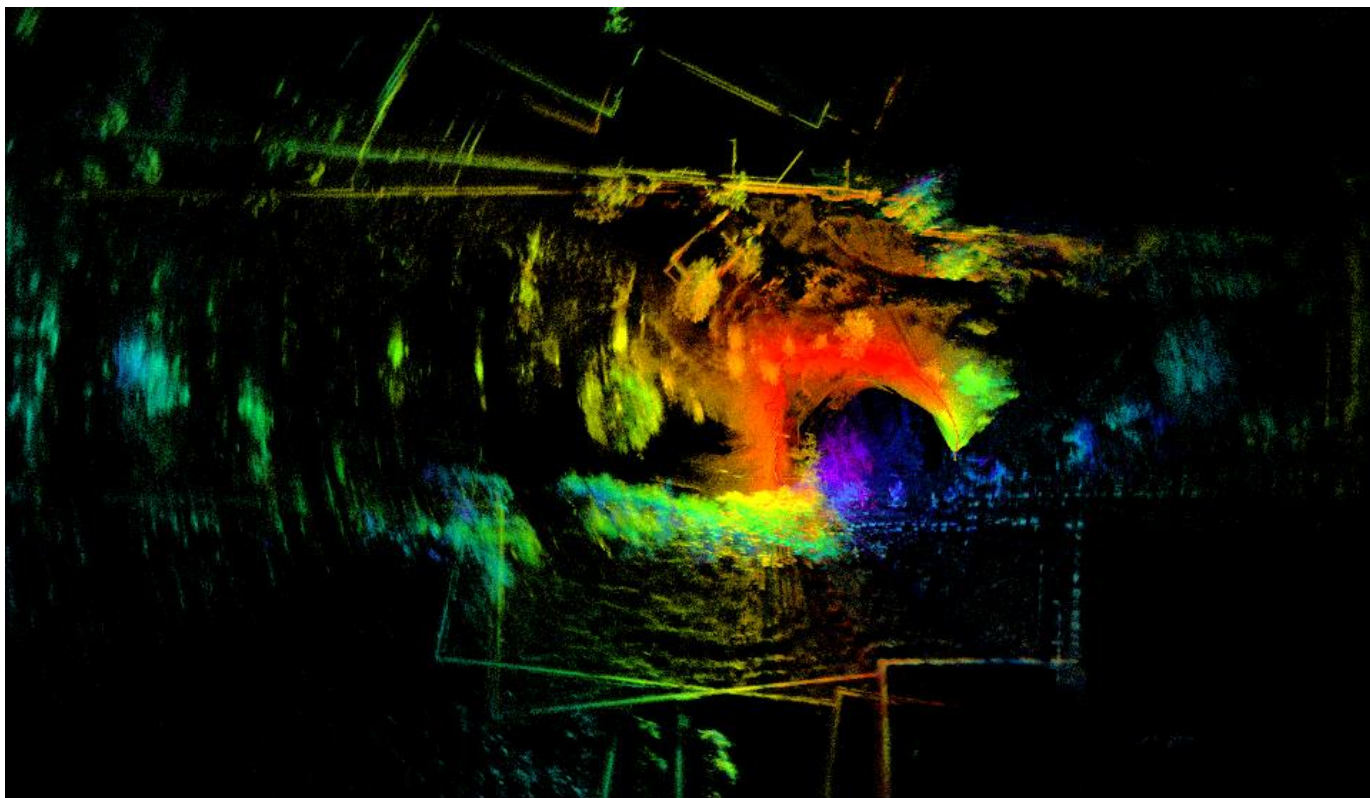
Γενικά, παρόλο που η μέθοδος εφαρμόστηκε με επιτυχία στα δεδομένα CYT_02, οι περισσότερες δοκιμές που πραγματοποιήθηκαν με τον αλγόριθμο LOAM LIVOX, δεν παρουσίασαν αντίστοιχο αποτέλεσμα καθώς είχαν παραμορφώσεις, σαρώσεις που συνενωθήκαν λανθασμένα και περιπτώσεις πλήρους αποτυχίας όπως απεικονίζεται στις εικόνες των αποτελεσμάτων 28-30. Τα σφάλματα αυτά, εμφανίζονταν κυρίως με την παρουσία έντονων στροφών, ενώ παράλληλα, λόγω της μετακίνησης με το χέρι, υπήρχαν έντονοι κραδασμοί του σαρωτή. Τέλος, παρατηρήθηκαν καθυστερήσεις στην εκτέλεση του αλγορίθμου με αποτέλεσμα να μην εκτελείται σε πραγματικό χρόνο υπό τις συνθήκες των δοκιμών, παρουσιάζοντας μια καθυστέρηση της τάξης του 20-30% του χρόνου καταγραφής, για παράδειγμα, κατά την πρώτη δοκιμή ο αλγόριθμος χρειάστηκε περίπου 74sec για καταγραφή διάρκειας 58sec.



Εικόνα 28: Αποτέλεσμα LOAM LIVOX – καταγραφή 1



Εικόνα 29: Αποτέλεσμα LOAM LIVOX – καταγραφή 2



Εικόνα 30: Αποτέλεσμα LOAM LIVOX – καταγραφή 3

Αυτά τα σφάλματα, ενδεχομένως να οφείλονται στην ασυμβατότητα του σαρωτή ως προς την μέθοδο. Όπως αναφέρθηκε παραπάνω, ο συγκεκριμένος αλγόριθμος χρησιμοποιεί το μοντέλο καταγραφής μετρήσεων από τον σαρωτή για την διαδικασία αφαίρεσης του motion blur, ενώ το μοντέλο σάρωσης και αφαίρεσης motion blur επηρεάζουν την εξαγωγή χαρακτηριστικών σημείων LOAM και συνεπώς την ακρίβεια της συνένωσης (τελικός χάρτης). Το πρόβλημα αυτό εντείνεται σε στροφές οπού, λόγω της αλλαγής πορείας, εντοπίζονται λιγότερα κοινά σημεία ενώ παρατηρείται έντονη κίνηση. Ωστόσο, κάτι τέτοιο δεν θα έπρεπε να συμβαίνει καθώς ο αλγόριθμος είναι σχεδιασμένος για σαρωτή παρόμοιας τεχνολογίας της ίδιας εταιρείας (LIVOX MID-40) ο οποίος και χαρακτηρίζεται από ελαφρώς μικρότερο οπτικό πεδίο και απαιτεί περισσότερο χρόνο για να ολοκληρώσει την σάρωση. Συμπεραίνουμε λοιπόν πως, ενδεχομένως, η συγκεκριμένη εφαρμογή, αφορά μια αρκετά “σφιχτή” υλοποίηση γύρω από έναν συγκεκριμένο σαρωτή με τις παραμέτρους να μην δουλεύουν τόσο ικανοποιητικά για άλλους σαρωτές ακόμα και αν αυτοί εμφανίζουν αντίστοιχα χαρακτηριστικά.

Για τους παραπάνω λόγους, σε επόμενο στάδιο πραγματοποιήθηκαν ξανά οι ίδιες δοκιμές με διαφοροποίηση των παραμέτρων (Εικόνα 31) όπως μικρότερο σφάλμα στον ICP, μεγαλύτερο πεδίο αναζήτησης χαρακτηριστικών και μεγαλύτερο βάρος των χαρακτηριστικών στις ακμές. Με τις νέες αυτές παραμέτρους τα αποτελέσματα βελτιώθηκαν σημαντικά και στις τρεις καταγραφές (Εικόνες 32 και 33). Ωστόσο οι αλλαγές αυτές δεν είχαν σαφές θεωρητικό υπόβαθρο και πραγματοποιήθηκαν κυρίως εμπειρικά.


```

10 feature_extraction:
11   scan_line: 64
12   mapping_line_resolution: 0.1
13   mapping_plane_resolution: 0.4
14   livox_min_sigma: 7e-4
15   livox_min_dis: 0.1
16   corner_curvature: 0.1
17   surface_curvature: 0.005
18   minimum_view_angle: 5
19
20 optimization:
21   minimum_icp_R_diff: 0.01
22   minimum_icp_T_diff: 0.01
23   maximum_residual_blocks: 200
24   max_allow_final_cost: 2.0
25   icp_maximum_iteration: 15
26   ceres_maximum_iteration: 50

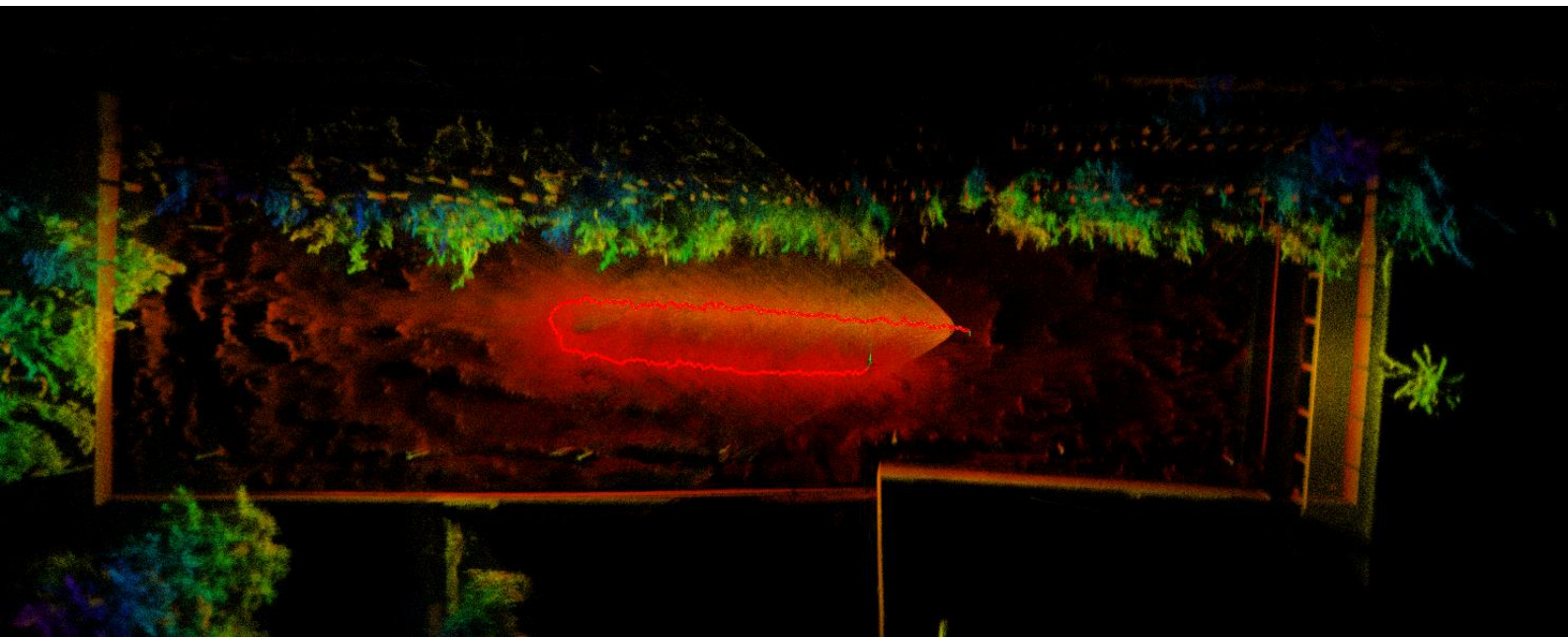
```

```

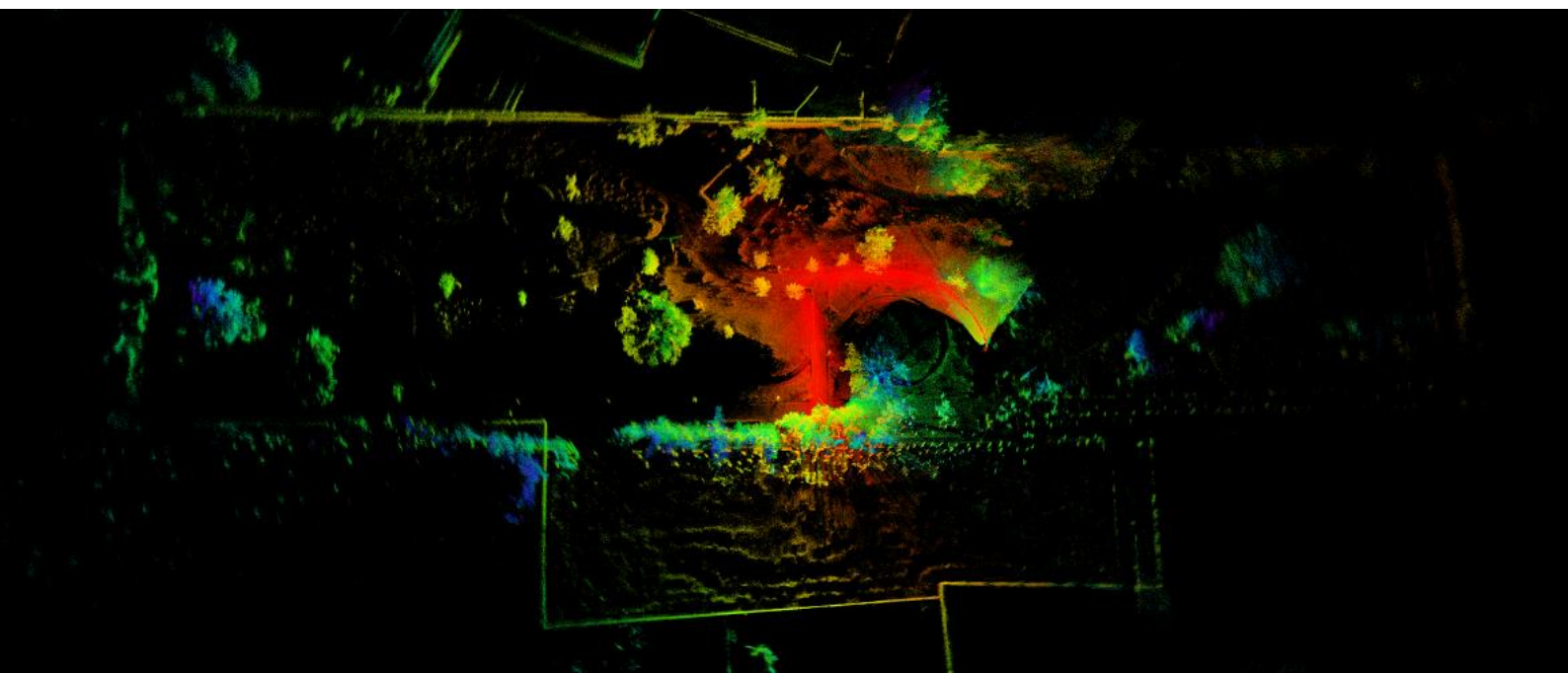
10 feature_extraction:
11   scan_line: 64
12   mapping_line_resolution: 0.4
13   mapping_plane_resolution: 0.2
14   livox_min_sigma: 7e-4
15   livox_min_dis: 0.1
16   corner_curvature: 0.1
17   surface_curvature: 0.005
18   minimum_view_angle: 10
19
20 optimization:
21   minimum_icp_R_diff: 0.005
22   minimum_icp_T_diff: 0.005
23   maximum_residual_blocks: 80
24   max_allow_final_cost: 0.8
25   icp_maximum_iteration: 30
26   ceres_maximum_iteration: 60

```

Εικόνα 31: Παράμετροι LOAM LIVOX πριν (αριστερά), νέες παράμετροι (δεξιά)



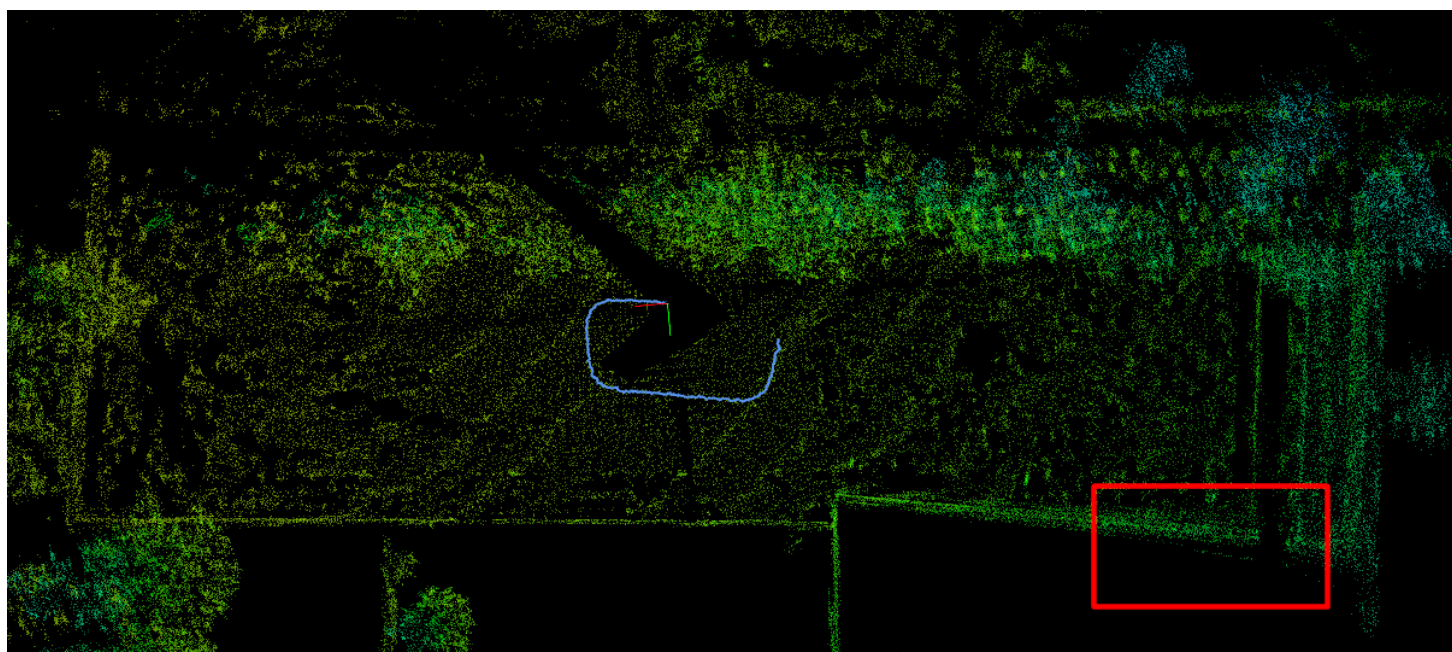
Εικόνα 32: Βελτιωμένο αποτέλεσμα LOAM LIVOX – καταγραφή 2



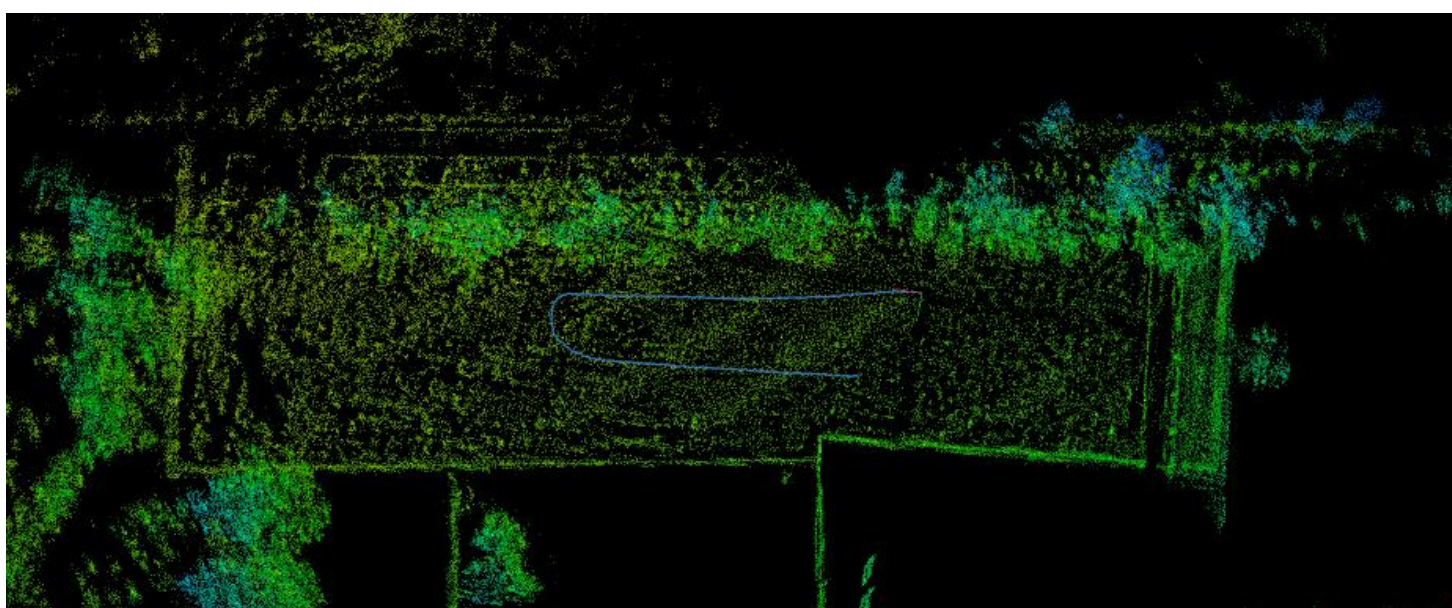
Εικόνα 33: Βελτιωμένο αποτέλεσμα LOAM LIVOX – καταγραφή 3

4.2.2.2 Εφαρμογή KISS-ICP

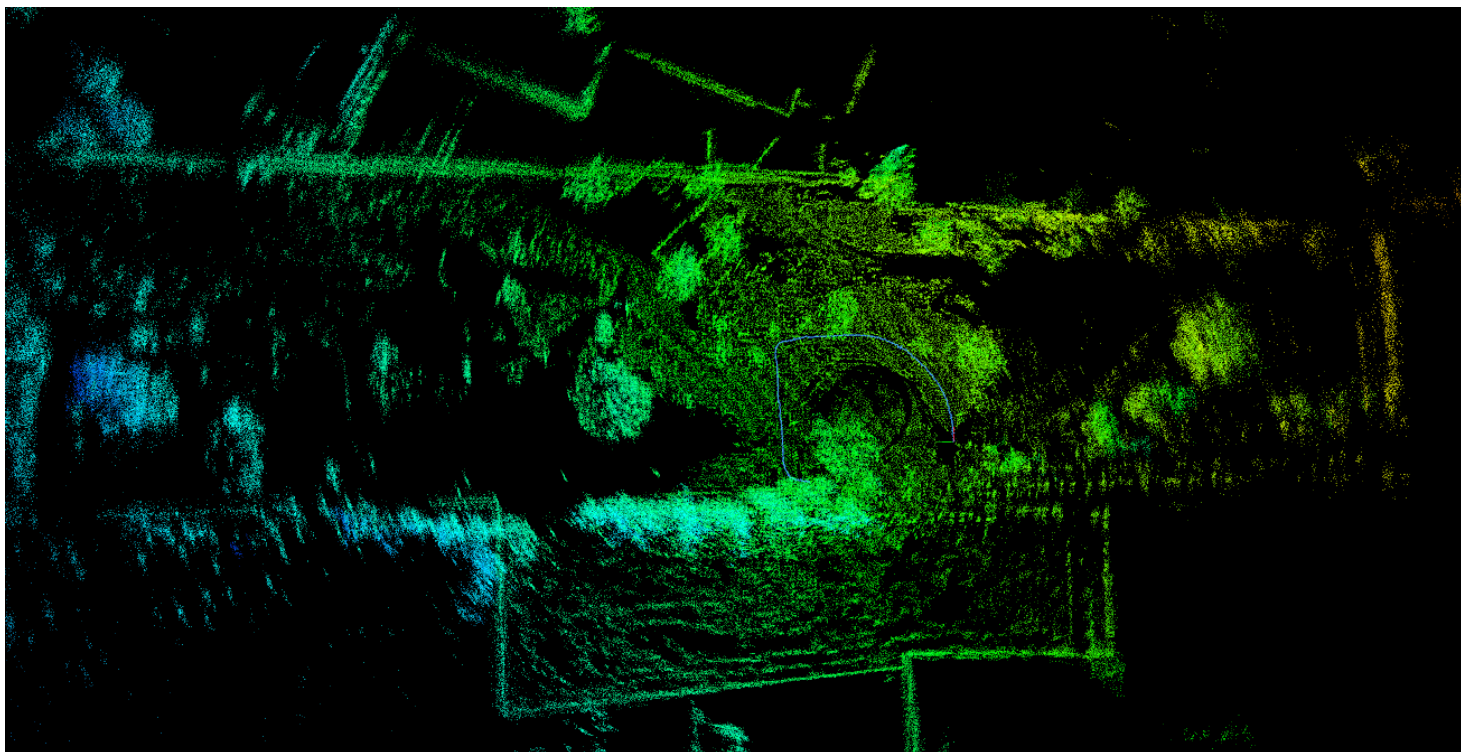
Γενικά σε όλες τις δοκιμές ο αλγόριθμος KISS-ICP τα πήγε αρκετά καλά, παρόλο που υποστηρίζει πολύ απλό μοντέλο που δεν συμπεριλαμβάνει ούτε εντοπισμό κλειστών βρόχων αλλά ούτε και βελτιστοποίηση γράφου. Έτσι σταδιακά, προς το τέλος των σαρώσεων, αρχίζει να γίνεται εμφανής στα αποτελέσματα η συγκέντρωση σφαλμάτων (Εικόνες 34-36), με τα αντικείμενα να εμφανίζονται ελαφρώς μεγεθυμένα (π.χ. οι τοίχοι). Επιπρόσθετα αρνητικά χαρακτηριστικά των αποτελεσμάτων αποτελούν τόσο το πολύ αραιό νέφος σημείων όσο και η αδυναμία αποθήκευσης του χάρτη. Τέλος, επισημαίνεται πως όλες οι δοκιμές εκτελέστηκαν σε πραγματικό χρόνο.



Εικόνα 34: Αποτέλεσμα KISS-ICP – καταγραφή 1



Εικόνα 35: Αποτέλεσμα KISS-ICP – καταγραφή 2

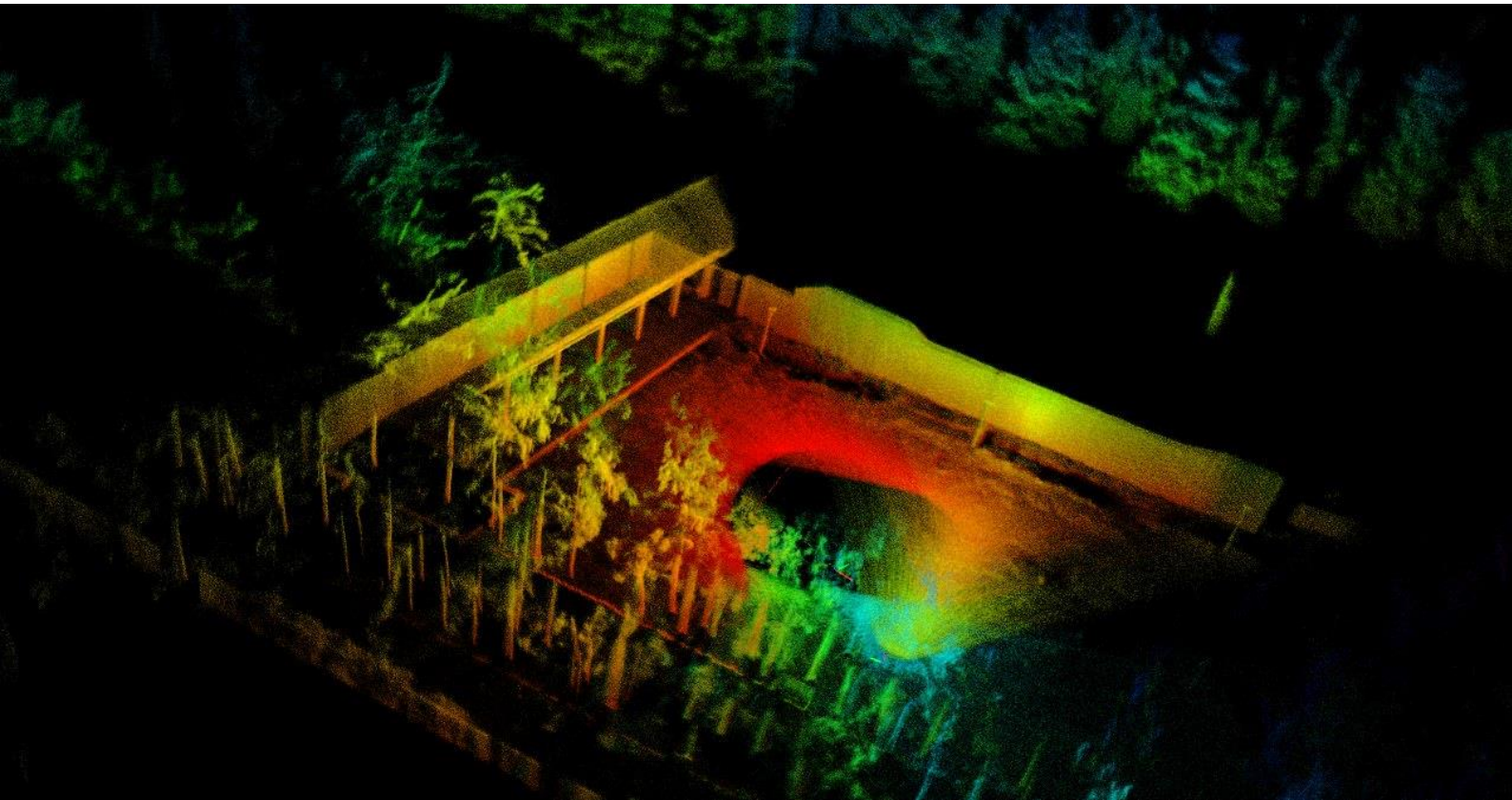


Εικόνα 36: Αποτέλεσμα KISS-ICP – καταγραφή 3

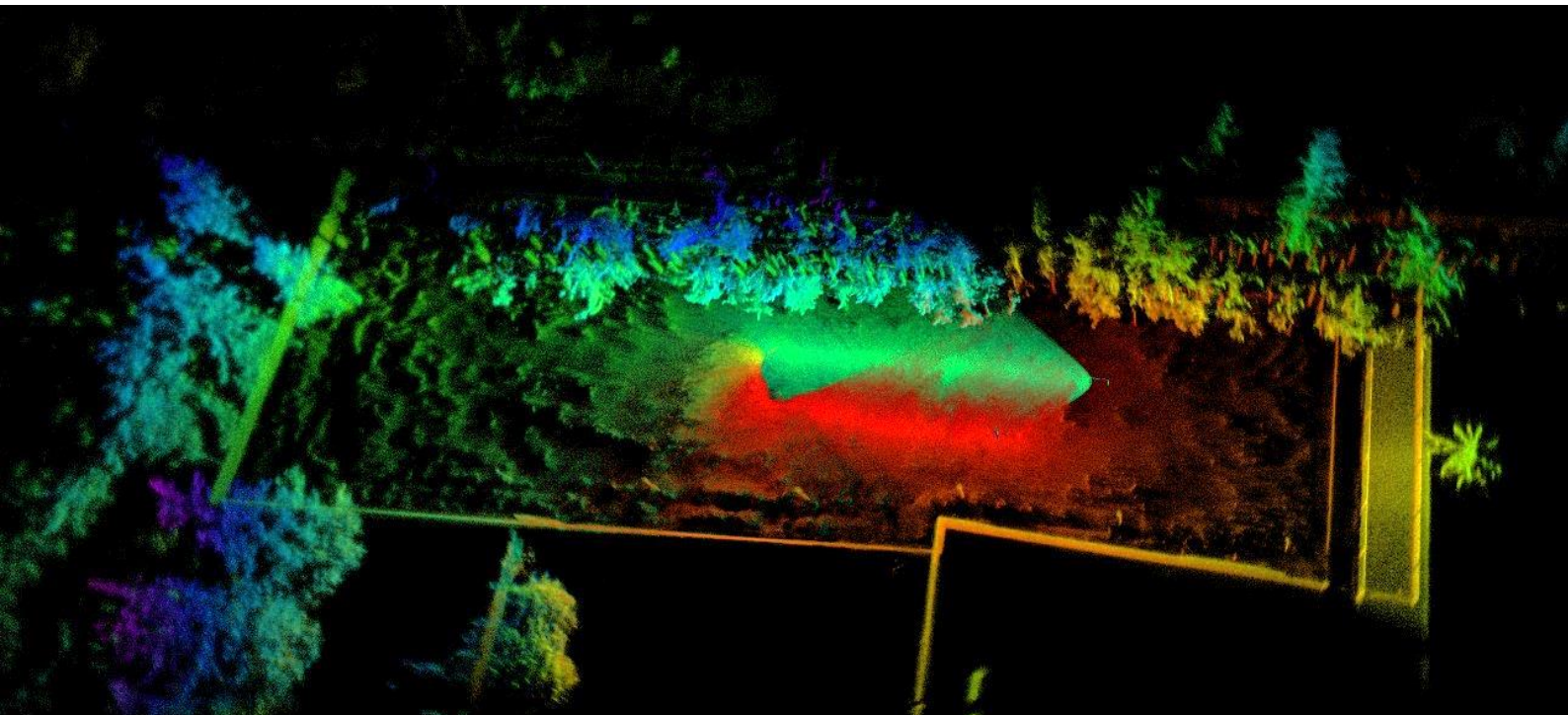
4.2.2.3 Εφαρμογή LIVOX MAPPING

Ο αλγόριθμος LIVOX MAPPING παρουσίασε πολλές ομοιότητες με τον LOAM LIVOX, κάτι λογικό καθώς πρακτικά και οι δυο έχουν ως βάση τον LOAM. Ο συγκεκριμένος αλγόριθμος φέρει την δυνατότητα επιλογής διαφορετικών παραμέτρων για τους σαρωτές LIVOX MID-40 και LIVOX HORIZON, αλλά όχι για τον LIVOX AVIA. Έτσι επιλέχτηκε να επικεντρωθούν οι δοκιμές στο σετ παραμέτρων του LIVOX HORIZON, καθώς ο σαρωτής αυτός παρουσιάζει περισσότερα κοινά με αυτόν των δοκιμών.

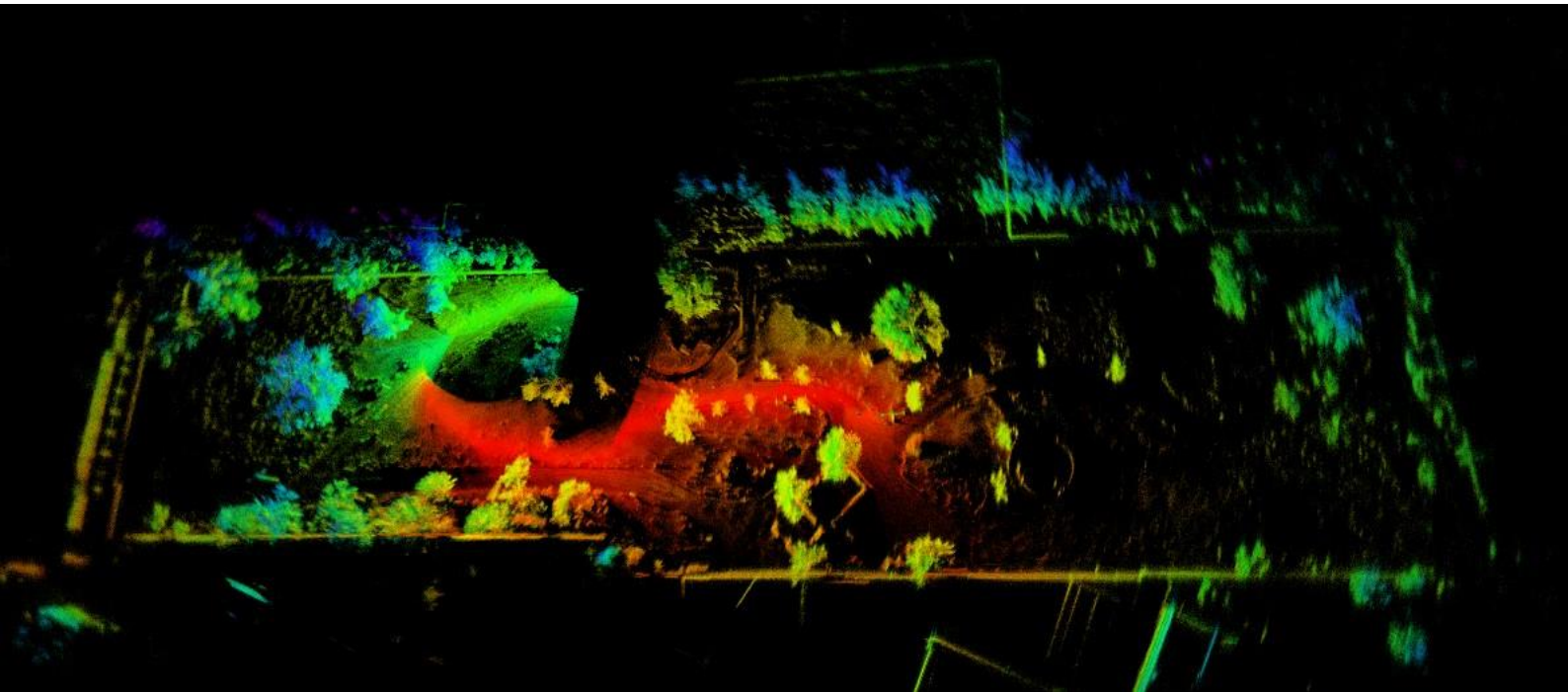
Ο συγκεκριμένος αλγόριθμος παρουσίασε μεικτά αποτελέσματα ενώ στις περισσότερες δοκιμές δεν εντοπίζονται προβλήματα (Εικόνες 37-39). Εντούτοις, σε κάποιες περιπτώσεις προέκυψαν σφάλματα συνενώσεων και αρκετά έντονες στροφές (Εικόνα 40). Κάτι τέτοιο προφανώς είναι αναμενόμενο αφού, ως αλγόριθμος ο οποίος βασίζεται σε εξαγωγή χαρακτηριστικών, μπορεί να εμφανίσει ευκολότερα προβλήματα σε περιπτώσεις έντονων στροφών, ειδικά όταν ο αισθητήρας μετακινείται με το χέρι και παρατηρούνται έντονοι κραδασμοί. Την κατάσταση δυσχεραίνει περαιτέρω το σχετικά μικρό οριζόντιο οπτικό πεδίο των σαρωτών στέρεας κατάστασης (σε αντίθεση με πολλά είδη μηχανικών σαρωτών με οπτικό πεδίο 360°), γεγονός το οποίο μπορεί να οδηγήσει σε εύρεση μειωμένου αριθμού επικαλυπτόμενων χαρακτηριστικών μεταξύ δυο σαρώσεων. Τέλος, ο αλγόριθμος δεν πέτυχε πραγματικούς χρόνους εκτέλεσης αφού υπήρχαν μικρές καθυστερήσεις του χρόνου καταγραφής (της τάξης του 10-20%). όμως ο τελικός χάρτης που εξάγεται είναι αρκετά πυκνός. Παρακάτω παρουσιάζονται τα αποτελέσματα των δοκιμών αυτών (Εικόνες 37-40).



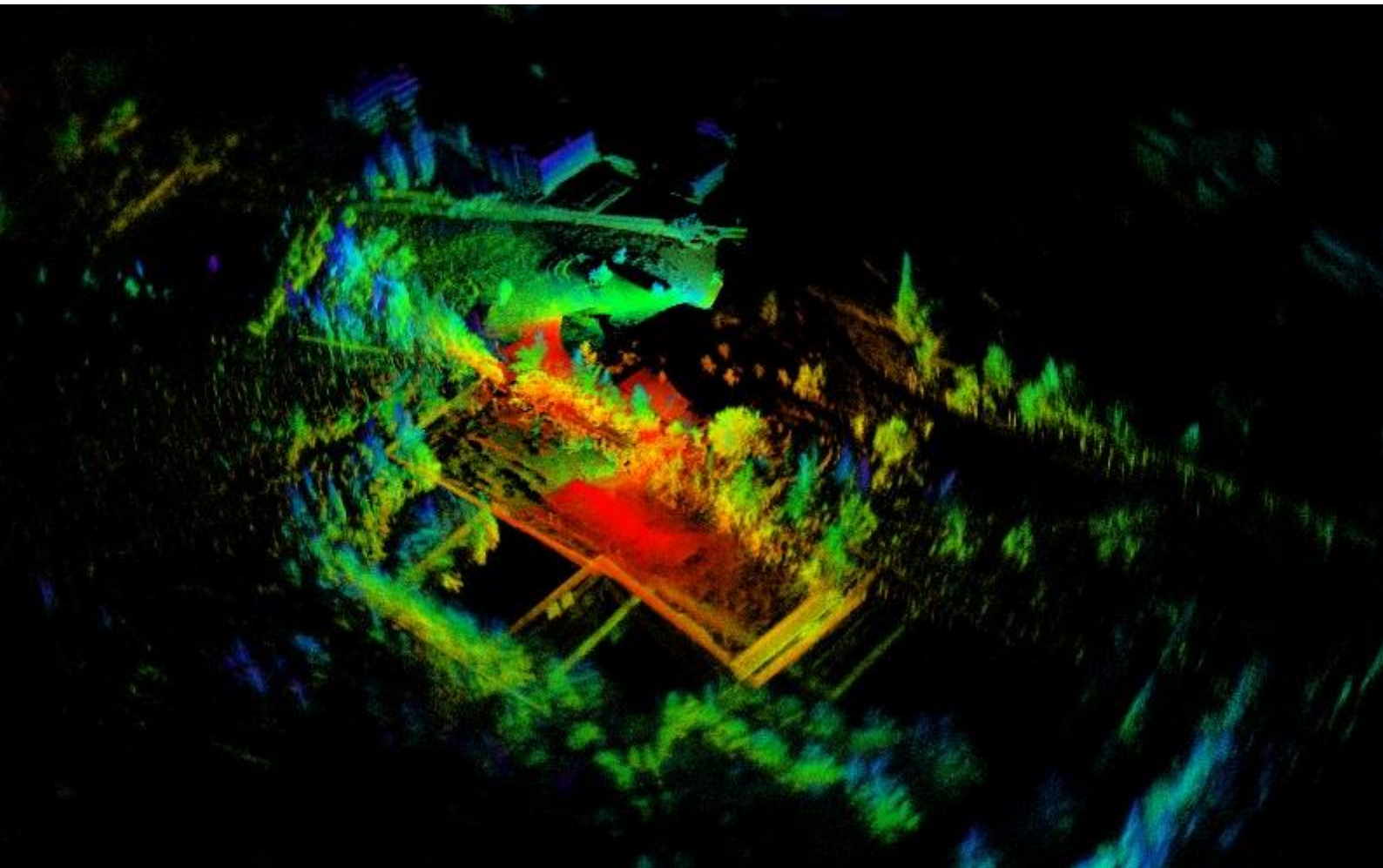
Εικόνα 37: Αποτέλεσμα LIDAR MAPPING – καταγραφή 4



Εικόνα 38: Αποτέλεσμα LIDAR MAPPING – καταγραφή 5



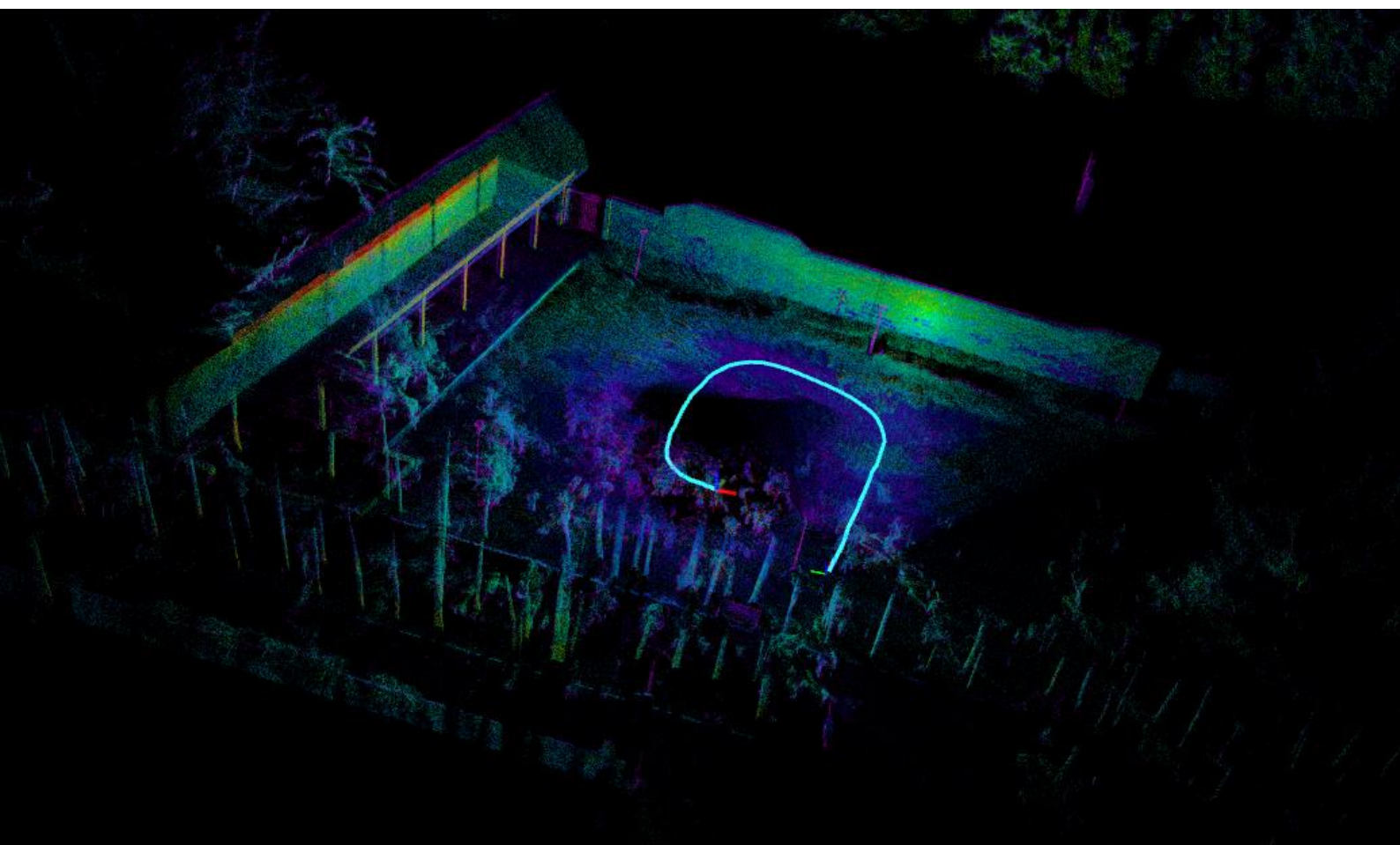
Εικόνα 39: Αποτέλεσμα LIVOX MAPPING – καταγραφή 6



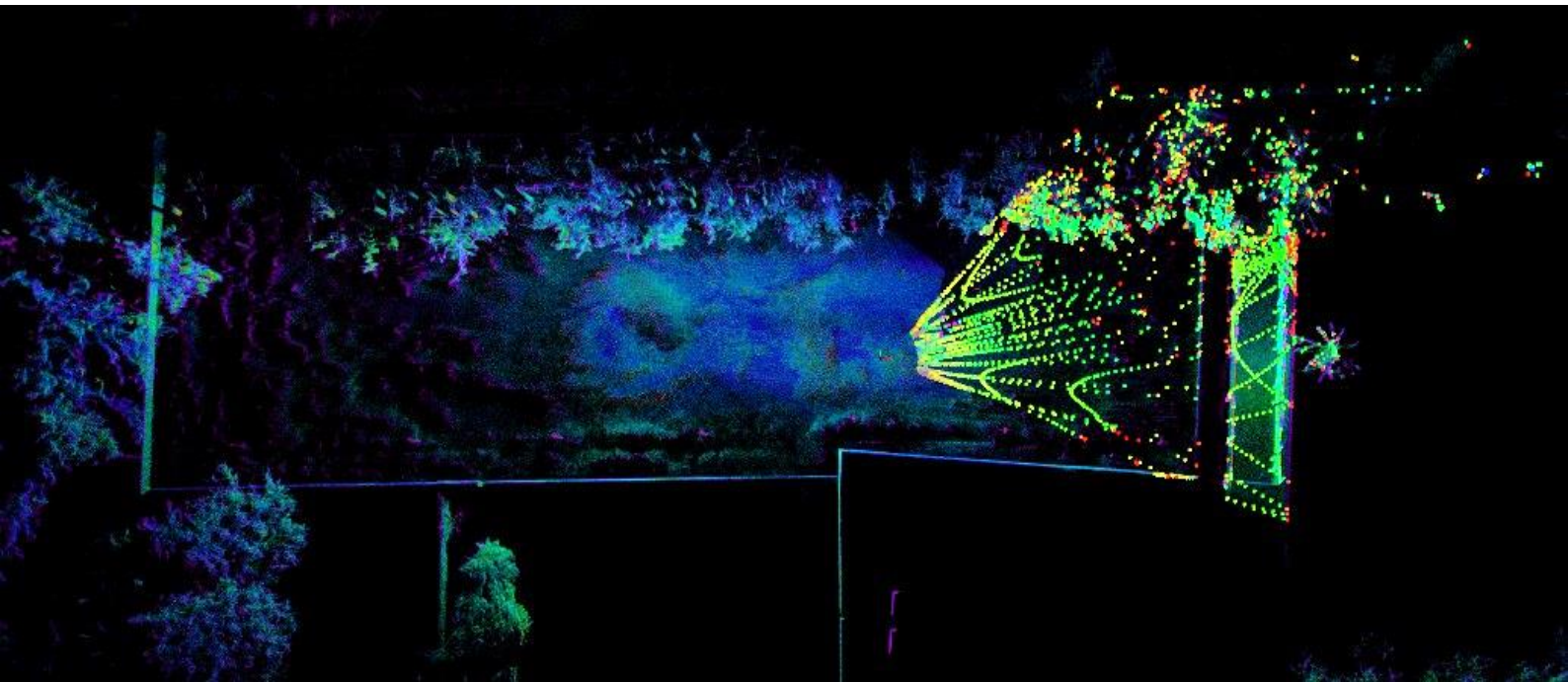
Εικόνα 40: Αποτέλεσμα LIVOX MAPPING – καταγραφή 7

4.2.2.4 Εξωτερικές δοκιμές με τον FAST-LIO2

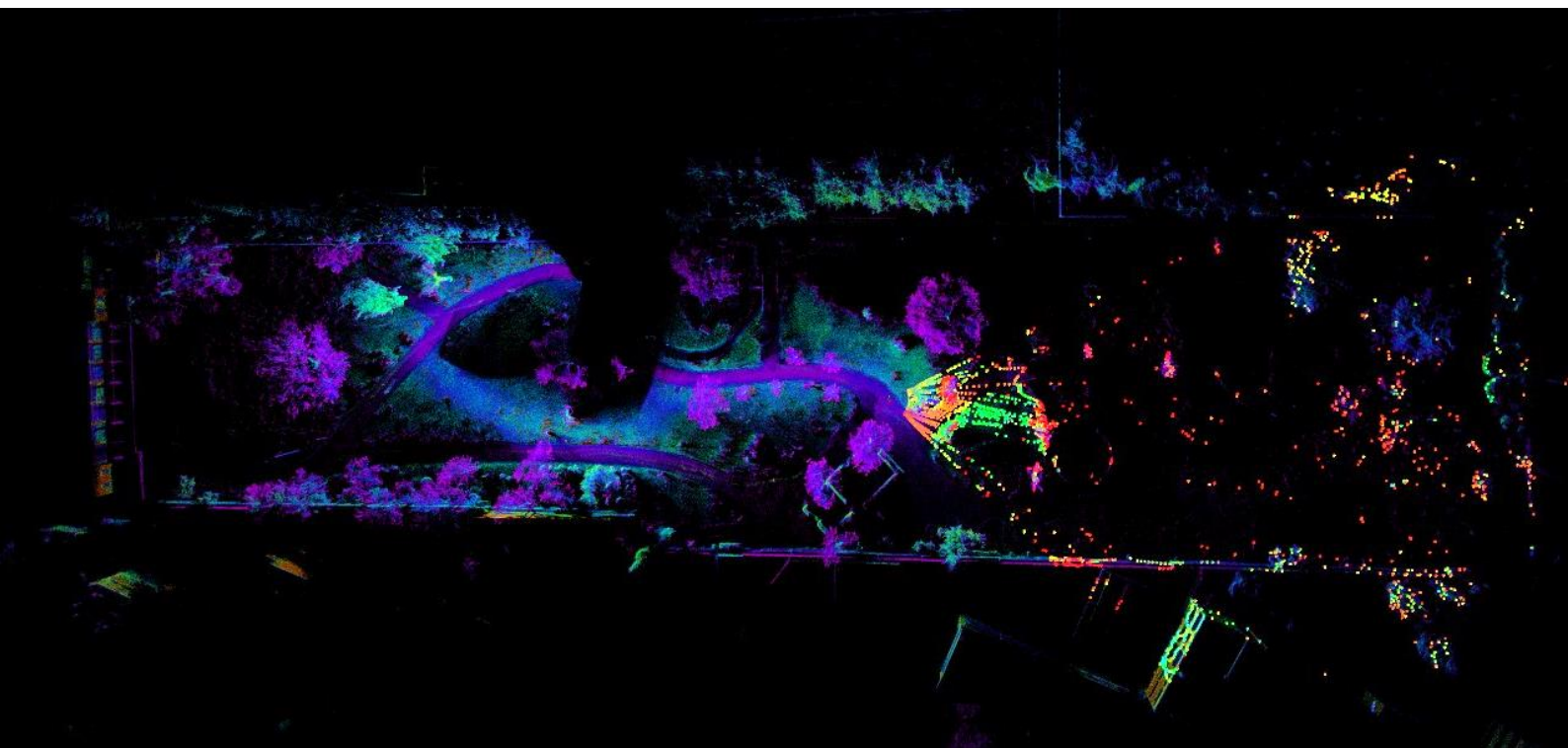
Ο FAST-LIO2 αφορά τον πιο πρόσφατο αλγόριθμο από τους LOAM LIVOX και LIVOX MAPPING ο οποίος διαφέρει αρκετά στην υλοποίηση αφού αποτελεί αλγόριθμο filter-based ο οποίος κάνει χρήση μετρήσεων IMU. Ο FAST-LIO2 περιλαμβάνει προφίλ χαρακτηριστικών για διαφορετικούς σαρωτές, συμπεριλαμβανομένου και αυτού των δοκιμών, οπότε δεν ήταν απαραίτητη η ανάγκη κάποιας ιδιαίτερης ρύθμισης των παραμέτρων. Ο συγκεκριμένος αλγόριθμος είχε αρκετά ικανοποιητικά αποτελέσματα στα δεδομένα ελέγχου, αλλά και σε όλες τις δοκιμές που παρουσιάζονται παρακάτω (Εικόνες 41-44) με εξαίρεση μια στην οποία παρατηρήθηκε μια διπλοπροβολή (Εικόνα 67). Σε γενικές γραμμές, τα νέφη σημείων δεν παρουσίασαν εμφανή προβλήματα λόγω στροφών ή θορύβου σε καμία από τις δοκιμές, όσο δυσμενείς και αν ήταν οι συνθήκες της σάρωσης (κραδασμοί, απότομες στροφές, σάρωση κινούμενων αντικειμένων). Τα νέφη σημείων που εξάγονται είναι αραιά αλλά με καλή ακρίβεια ώστε να μπορεί κανείς να διακρίνει με ευκολία λεπτομέρειες στο εικονιζόμενο περιβάλλον. Τέλος, σε κάθε περίπτωση ο αλγόριθμος εκτελέστηκε σε πραγματικό χρόνο χωρίς καθυστερήσεις (lag).



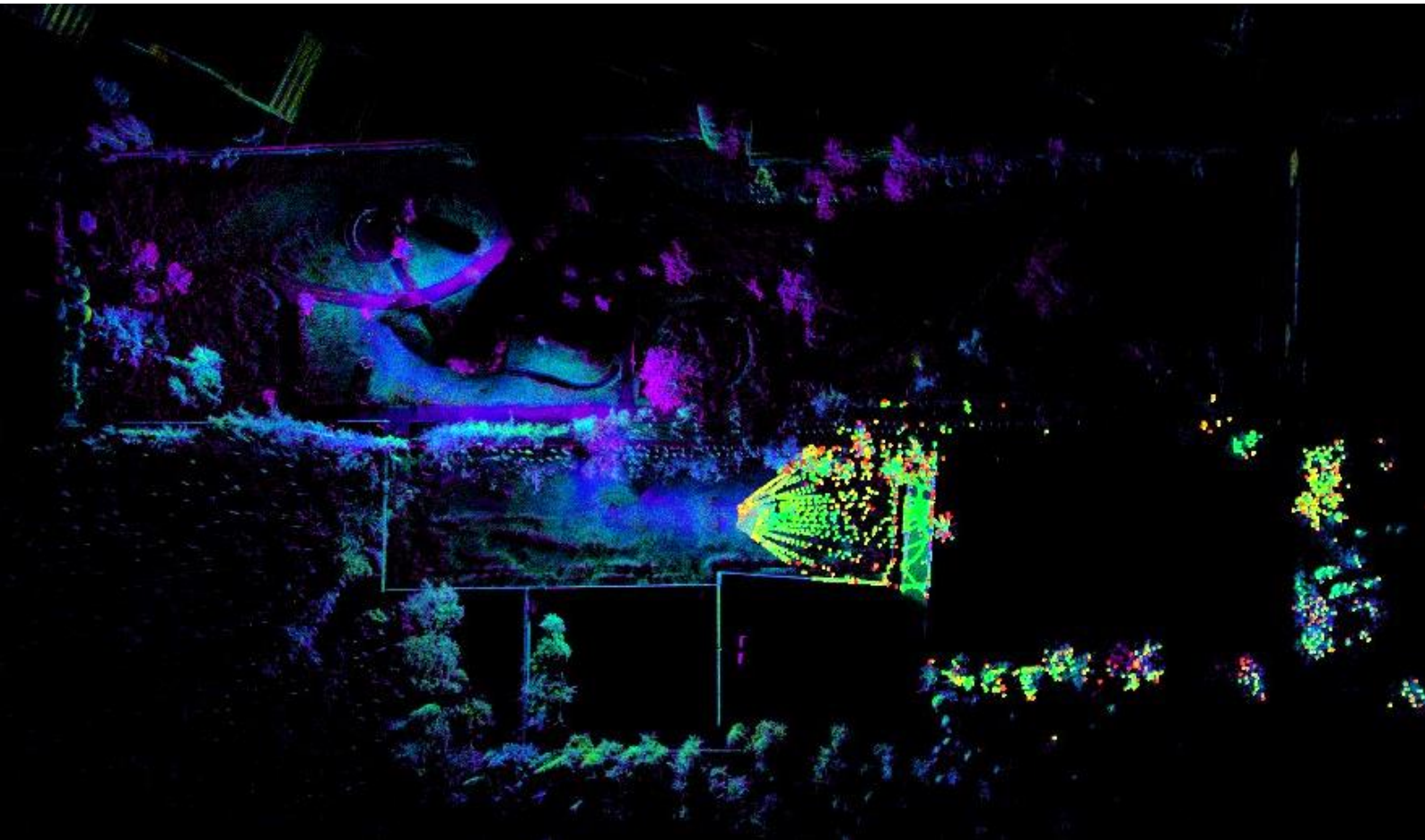
Εικόνα 41: Αποτέλεσμα FAST-LIO2 – καταγραφή 4



Εικόνα 42: Αποτέλεσμα FAST-LIO2 – καταγραφή 5



Εικόνα 43: Αποτέλεσμα FAST-LIO2 – καταγραφή 6

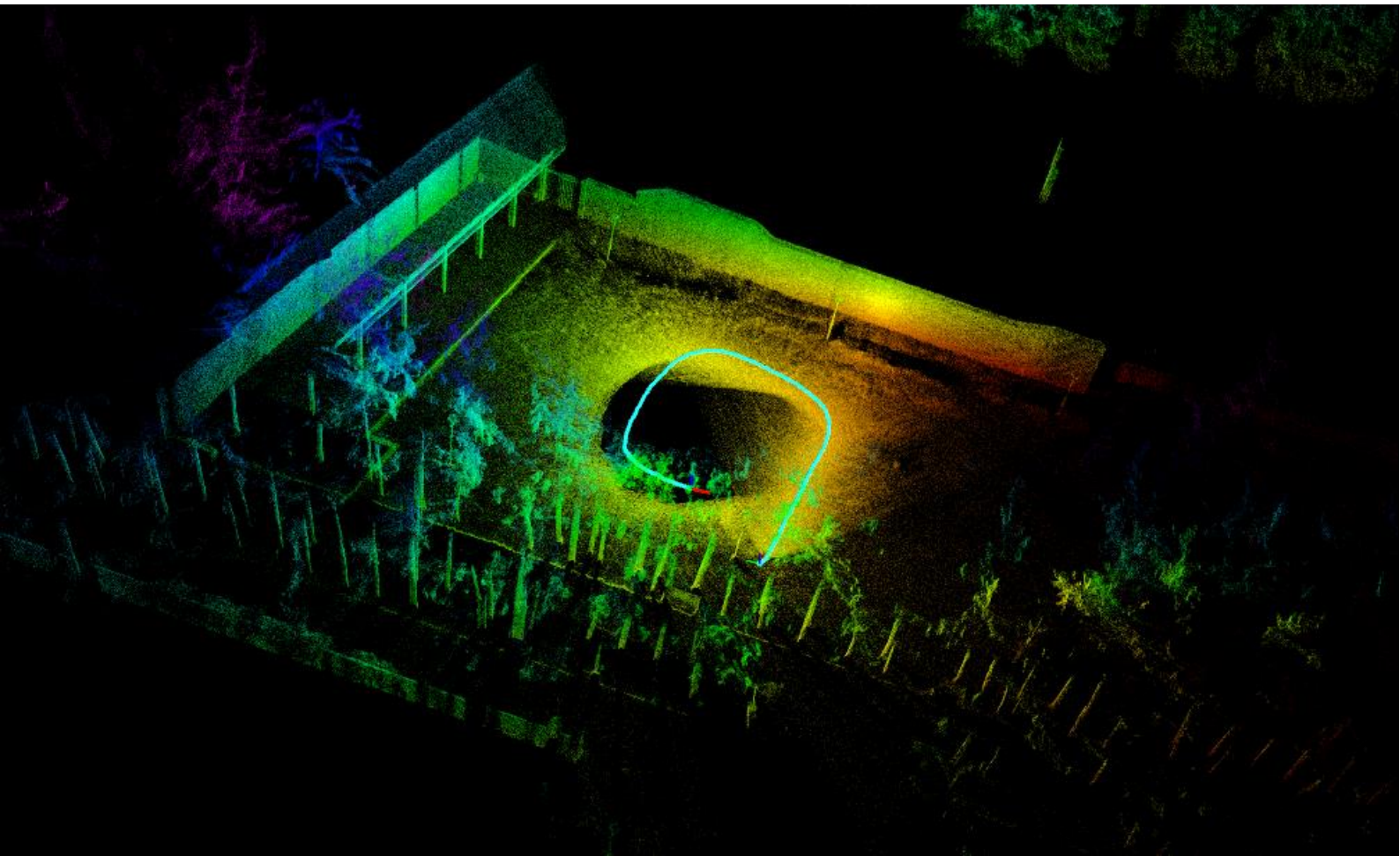


Εικόνα 44: Αποτέλεσμα FAST-LIO2 – καταγραφή 7

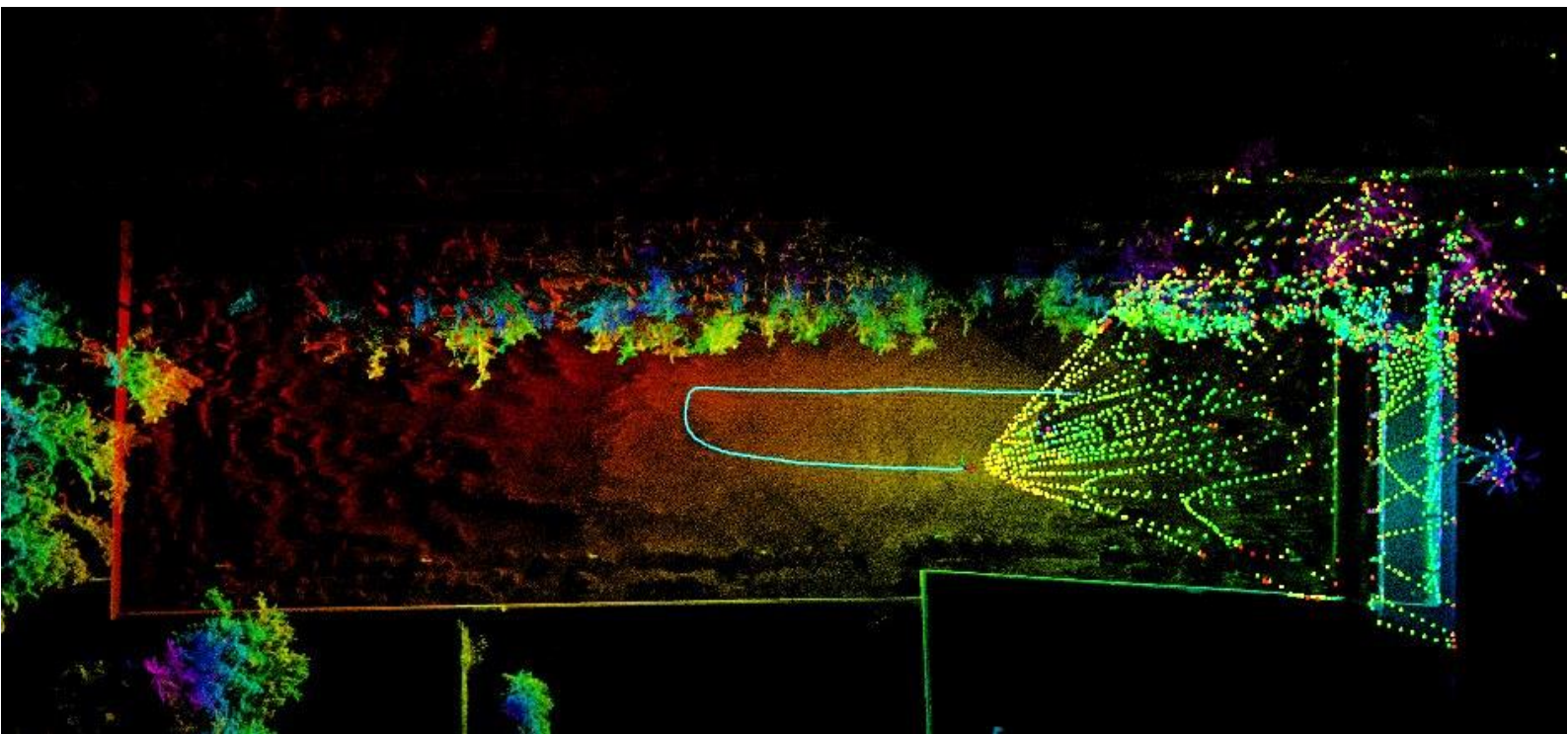
4.2.2.5 Εξωτερικές δοκιμές με τον FASTER-LIO

Ο FASTER-LIO είναι ο πιο πρόσφατος αλγόριθμος που εφαρμόστηκε στην παρούσα διπλωματική, και ο οποίος αποτελεί βελτίωση του FAST-LIO2. Όπως ήταν λοιπόν αναμενόμενο είχε τα καλύτερα αποτελέσματα από όλους τους αλγορίθμους, τόσο στα δεδομένα ελέγχου (test set), όσο και στις υπόλοιπες δοκιμές ανεξάρτητα από τις συνθήκες που επικρατούσαν.

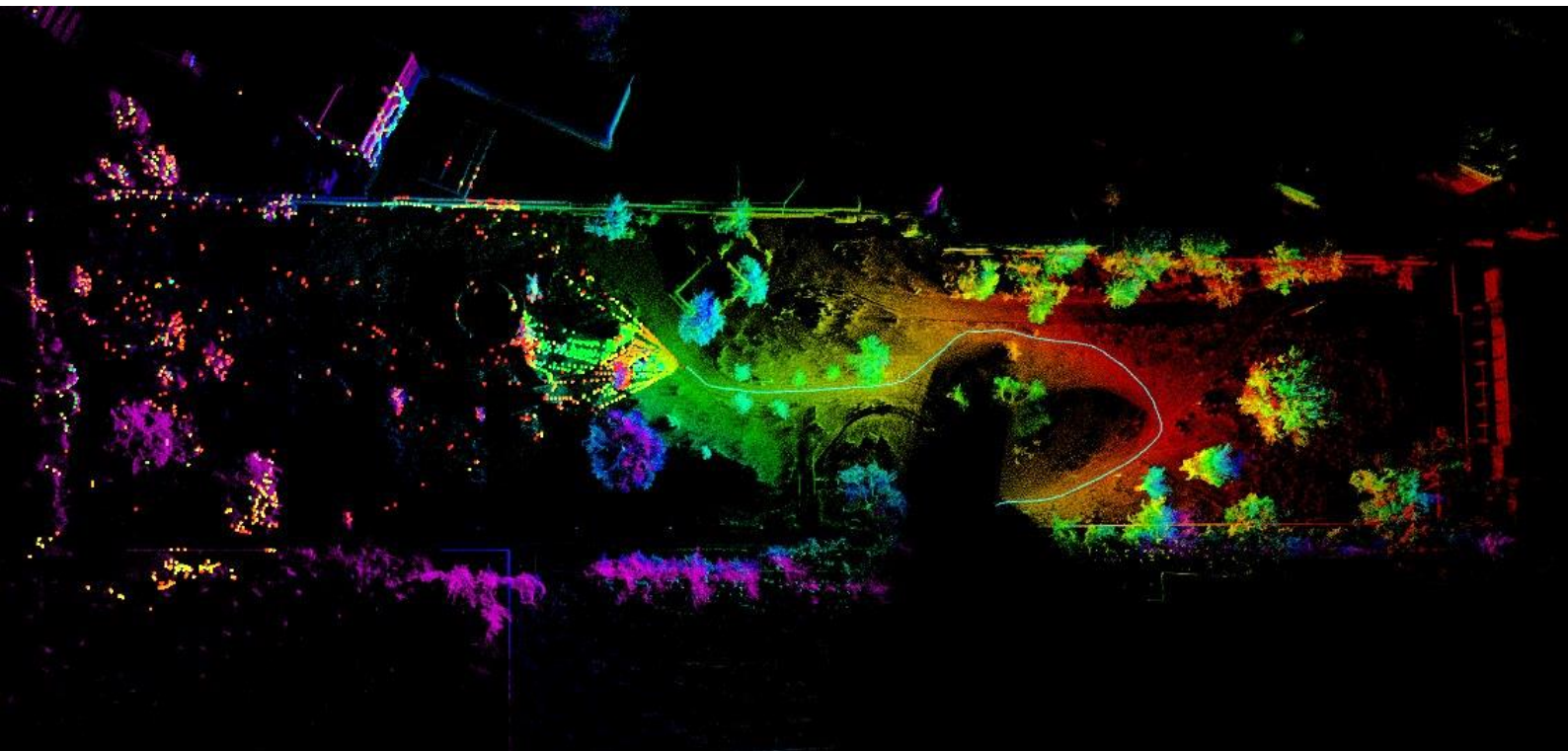
Τα τελικά συνενωμένα νέφη σημείων δεν υποδεικνύουν στροφές ή θόρυβο που μπορεί να εντοπιστεί με οπτική παρατήρηση, και όπως και στην περίπτωση του FAST-LIO2, είναι σχετικώς αραιά αλλά μπορεί κανείς να διακρίνει με ευκολία λεπτομέρειες (υψηλή ακρίβεια). Τέλος, και αυτός ο αλγόριθμος έτρεξε σε πραγματικό χρόνο για όλες τις περιπτώσεις, με τον viewer να μπορεί να αποκριθεί στιγμιαία όσο ενημερώνεται ο χάρτης. Παρακάτω παρουσιάζονται τα αποτελέσματα του αλγορίθμου (Εικόνες 45-48) για τις ίδιες καταγραφές στις οποίες εξετάστηκαν οι LIVOX MAPPING και FAST-LIO2.



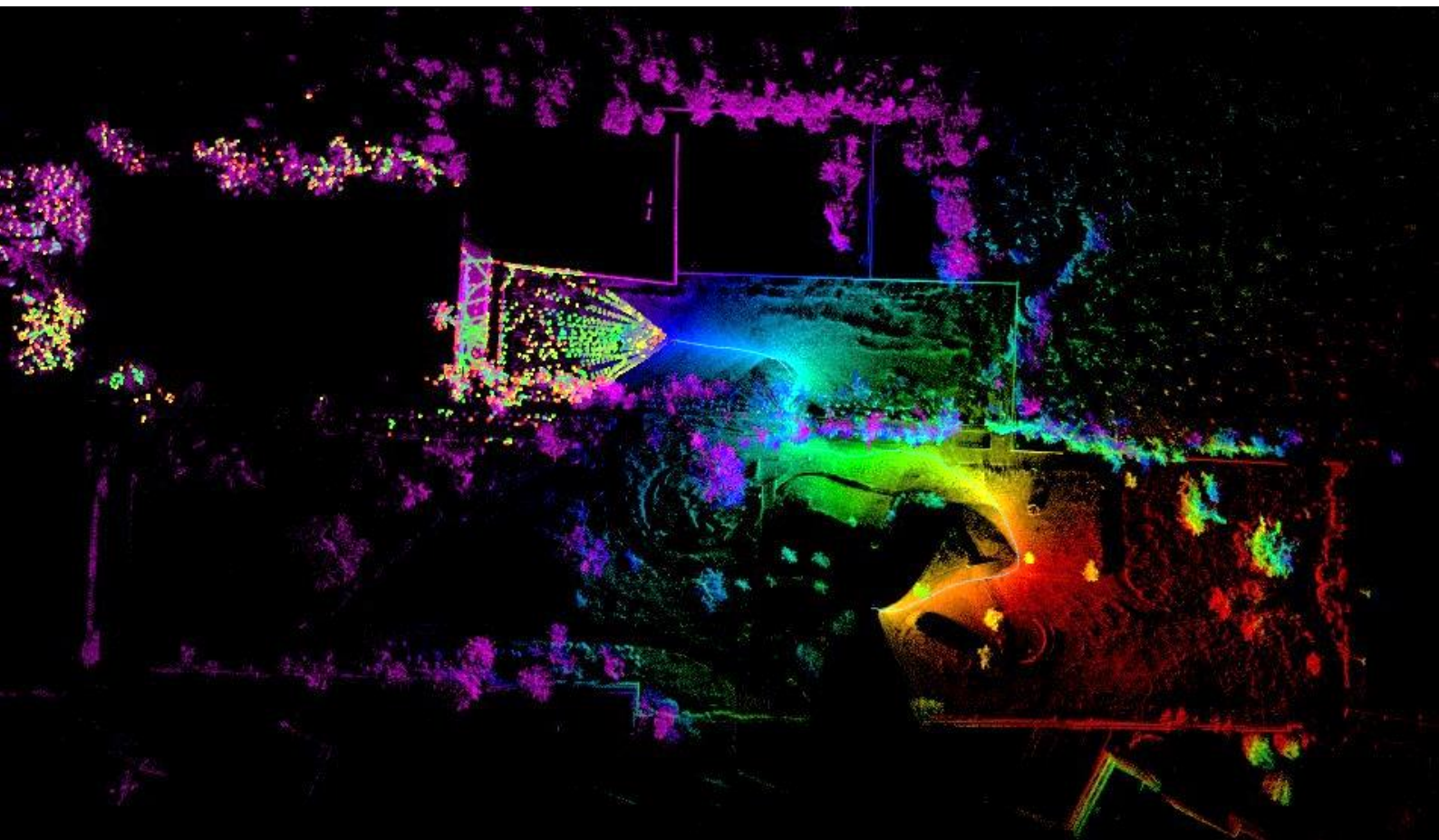
Εικόνα 45: Αποτέλεσμα Faster-LIO – καταγραφή 4



Εικόνα 46: Αποτέλεσμα Faster-LIO – καταγραφή 5



Εικόνα 47: Αποτέλεσμα Faster-LIO – καταγραφή 6



Εικόνα 48: Αποτέλεσμα Faster-LIO – καταγραφή 7

4.2.3 Πρακτική εφαρμογή σε λήψεις δεδομένων εσωτερικού χώρου

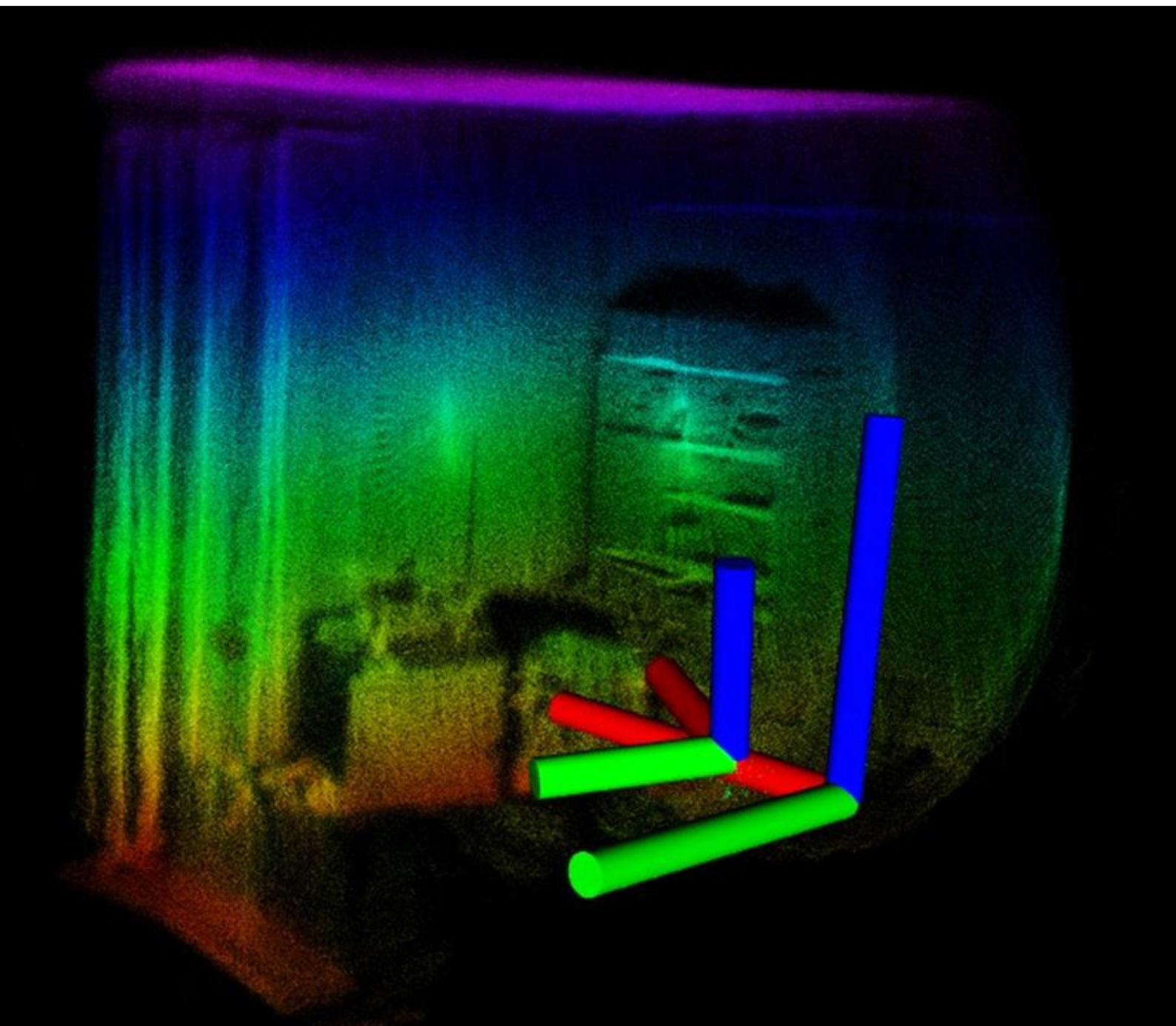
Αφού ολοκληρωθήκαν στο σύνολο τους οι εξωτερικές δοκιμές με τον LIVOX AVIA, πραγματοποιήθηκαν και κάποιες δοκιμές σε κλειστό εσωτερικό χώρο. Ωστόσο, αξίζει να σημειωθεί πως ο συγκεκριμένος σαρωτής δεν ενδείκνυται για τέτοιες εφαρμογές καθώς όπως αναφέρει ο κατασκευαστής, αρχικά δεν μπορεί να καταγράψει δεδομένα σε λιγότερο από ένα μέτρο, ενώ για αποστάσεις 1-5 μέτρα παρουσιάζει σημαντικό θόρυβο. Τέλος, το πεδίο ορατότητας που έχει δεν επαρκεί για να διαμορφώσει μια ικανή περιοχή σάρωσης σε έναν πολύ περιορισμένο χώρο, με αποτέλεσμα ο σαρωτής να καταγραφεί με πολύ μεγαλύτερη δυσκολία μεγάλο αριθμό σημείων και κατά συνέπεια χαρακτηριστικών ανά σάρωση.

Πάρα τους περιορισμούς, η μελέτη της λειτουργίας των αλγορίθμων αυτών σε τόσο αντίξοες συνθήκες μπορεί να παρέχει σημαντικά στοιχεία για την χρησιμότητα των αλγορίθμων αυτών τόσο σε εσωτερικές εφαρμογές αλλά και σε αντίστοιχες εξωτερικές (π.χ. στενοί δρόμοι). Επιπλέον, οι προγραμματιστές των αλγορίθμων γενικά δεν παρέχουν δοκιμαστικά δεδομένα που να αφορούν σκηνές σε περιορισμένο εσωτερικό χώρο για τον έλεγχο και την αξιολόγηση των αλγορίθμων κάτω από αυτές τις συνθήκες. Κρίθηκε λοιπόν σκόπιμο στο πλαίσιο της παρούσας διπλωματικής να πραγματοποιηθεί αξιολόγηση των αλγορίθμων στα περιβάλλοντα αυτά. Τέλος, όπως έχει αναφερθεί και παραπάνω, αυτές οι δοκιμές πραγματοποιήθηκαν σε εσωτερικό χώρο που αφορά δωμάτιο μικρών διαστάσεων (περίπου 5m x 3m), από σχετικά σταθερή θέση, με περιστροφικές κυρίως κινήσεις. Παρακάτω παρουσιάζονται οι δοκιμές στον χώρο αυτό και τα αποτελέσματα ανά αλγόριθμο.

4.2.3.1 Εσωτερικές δοκιμές με τον LOAM LIVOX

Γενικά ο LOAM LIVOX, μετά από διορθώσεις που πραγματοποιήθηκαν στις παραμέτρους, είχε αρκετά ικανοποιητικά αποτελέσματα και στις δοκιμές εσωτερικού χώρου. Κατάφερε να εντοπίσει αρκετά χαρακτηριστικά (features) απαραίτητα για τις συνενώσεις, παρά τις μεγάλες και απότομες στροφές του αισθητήρα, και την έντονη γεωμετρία της σκηνής. Έτσι, στα αποτελέσματα, πάρα τις μικρές αποστάσεις σάρωσης, δεν εμφανίζεται έντονος θόρυβος.

Παρακάτω παρουσιάζεται ένα στιγμιότυπο των αποτελεσμάτων (Εικόνα 49) στο οποίο δεν παρατηρούνται παραμορφωμένες γεωμετρίες σε μεγάλες ακμές ή πολλαπλές προβολές, ούτε έντονος θόρυβος, ενώ μπορούν να παρατηρηθούν μέχρι και σχετικά μικρότερα αντικείμενα (αν και κάπως αμυδρά), κάτι που είναι λογικό δεδομένης της χωρικής ανάλυσης του σαρωτή.

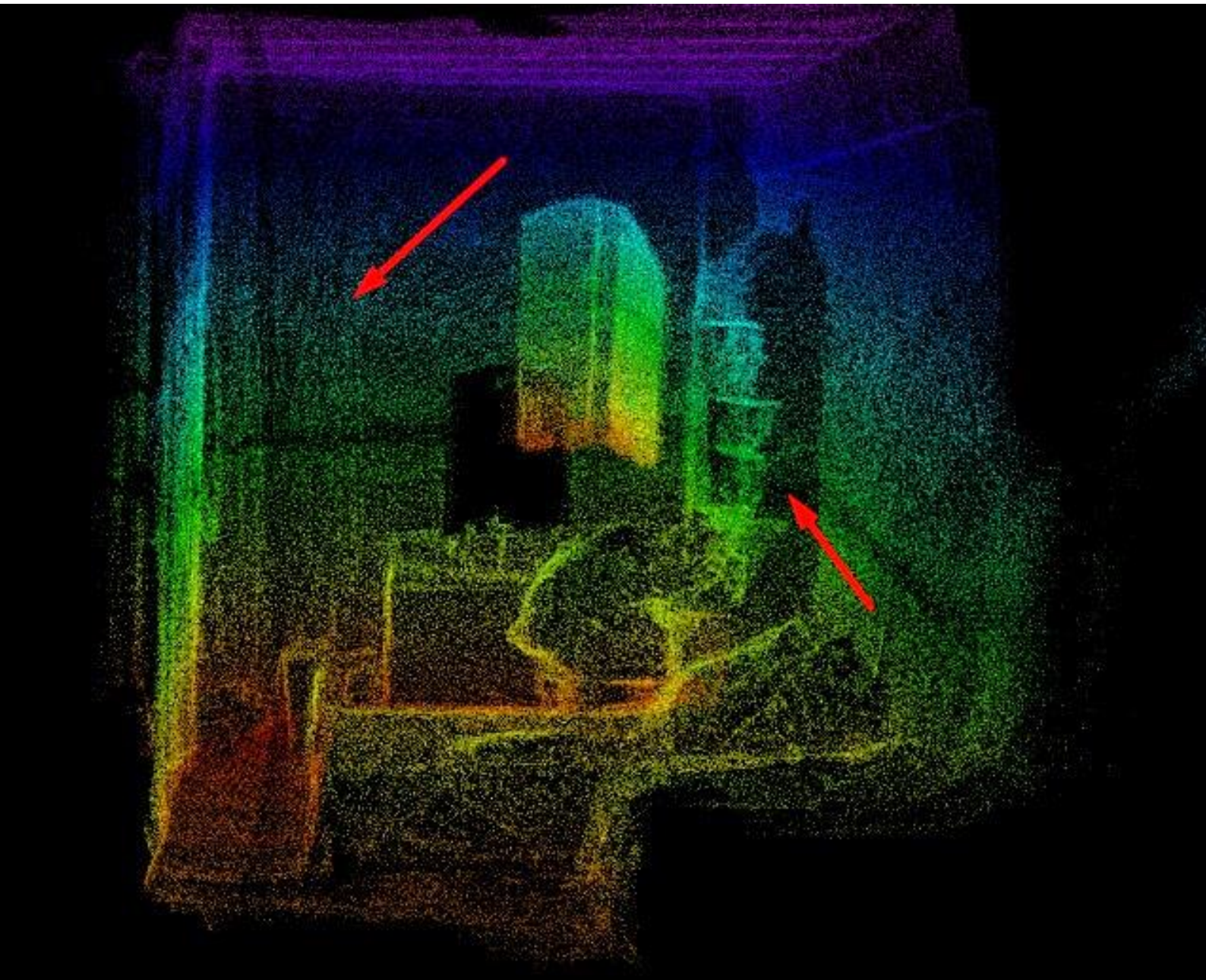


Εικόνα 49: Αποτέλεσμα LOAM LIVOX – καταγραφή εσωτερικού χώρου

4.2.3.2 Εσωτερικές δοκιμές με τον KISS ICP

Ο KISS ICP αρχικά απέτυχε πλήρως στο να λειτουργήσει ορθά σε εσωτερικό χώρο (Εικόνα 53). Για την αντιμετώπιση του προβλήματος, πραγματοποιήθηκε έλεγχος των παραμέτρων που χρησιμοποιεί η μέθοδος. Από τον έλεγχο αυτό διαπιστώθηκε ότι ο αλγόριθμος δεν αξιοποιούσε σημεία σε απόστασή μικρότερη των 5m (Εικόνα 51) και συνεπώς είναι προφανές ότι σε έναν χώρο μικρών διαστάσεων η μέθοδος θα αποτύγχανε. Ωστόσο, πέρα από την συγκεκριμένη παράμετρο, η αρχή λειτουργίας της μεθόδου προϋποθέτει πυκνή δειγματοληψία επί της σκληρής σάρωσης, με αποτέλεσμα η συγκεκριμένη μέθοδος να μην ενδείκνυται για μικρούς κλειστούς

χώρους σε καμία περίπτωση. Ύστερα από προσεκτική επιλογή των σχετικών παραμέτρων όπως μέγεθος voxel, πλήθος σημείων που διατηρούνται σε κάθε voxel και ελάχιστη απόστασή σάρωσης (Εικόνα 52), η μέθοδος κατάφερε να λειτουργήσει, αλλά το τελικό νέφος (Εικόνα 50) ήταν αρκετά αραιό. Τέλος, παρατηρούνται και κάποιες μικρές παραμορφώσεις στο αποτέλεσμα λόγω θορύβου (π.χ. η ακμή του επίπλου δεν εμφανίζεται καθαρά ως ακμή και έχουν εξαφανιστεί οι κυματισμοί της κουρτίνας). Γενικά λοιπόν, προκύπτει ως συμπέρασμά ότι ο KISS ICP είναι μια μέθοδος που δεν ενδείκνυται για εσωτερικούς χώρους ιδιαίτερα πολύ μικρών διαστάσεων.



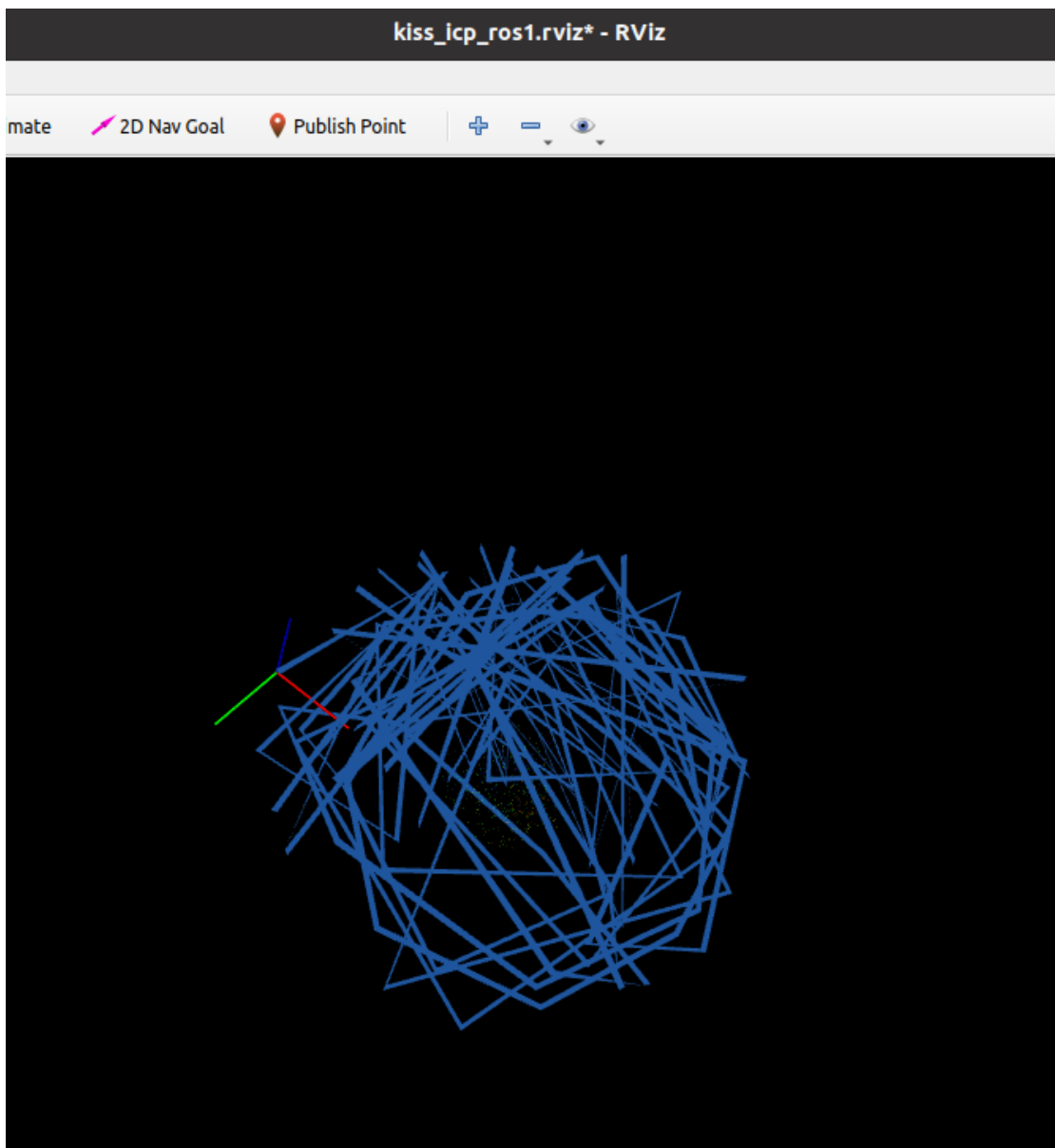
Εικόνα 50: Τελικό αποτέλεσμα KISS-ICP – καταγραφή εσωτερικού χώρου

```
<!-- KISS-ICP paramaters -->
<arg name="deskew" default="false"/>
<arg name="max_range" default="300.0" />
<arg name="min_range" default="5.0" />
<arg name="voxel_size" default="1" />
<arg name="max_points_per_voxel" default="20" />
```

Εικόνα 51: Default παράμετροι KISS ICP

```
<!-- KISS-ICP paramaters -->
<arg name="deskew" default="false"/>
<arg name="max_range" default="300.0" />
<arg name="min_range" default="1.0" />
<arg name="voxel_size" default="0.1" />
<arg name="max_points_per_voxel" default="40" />
```

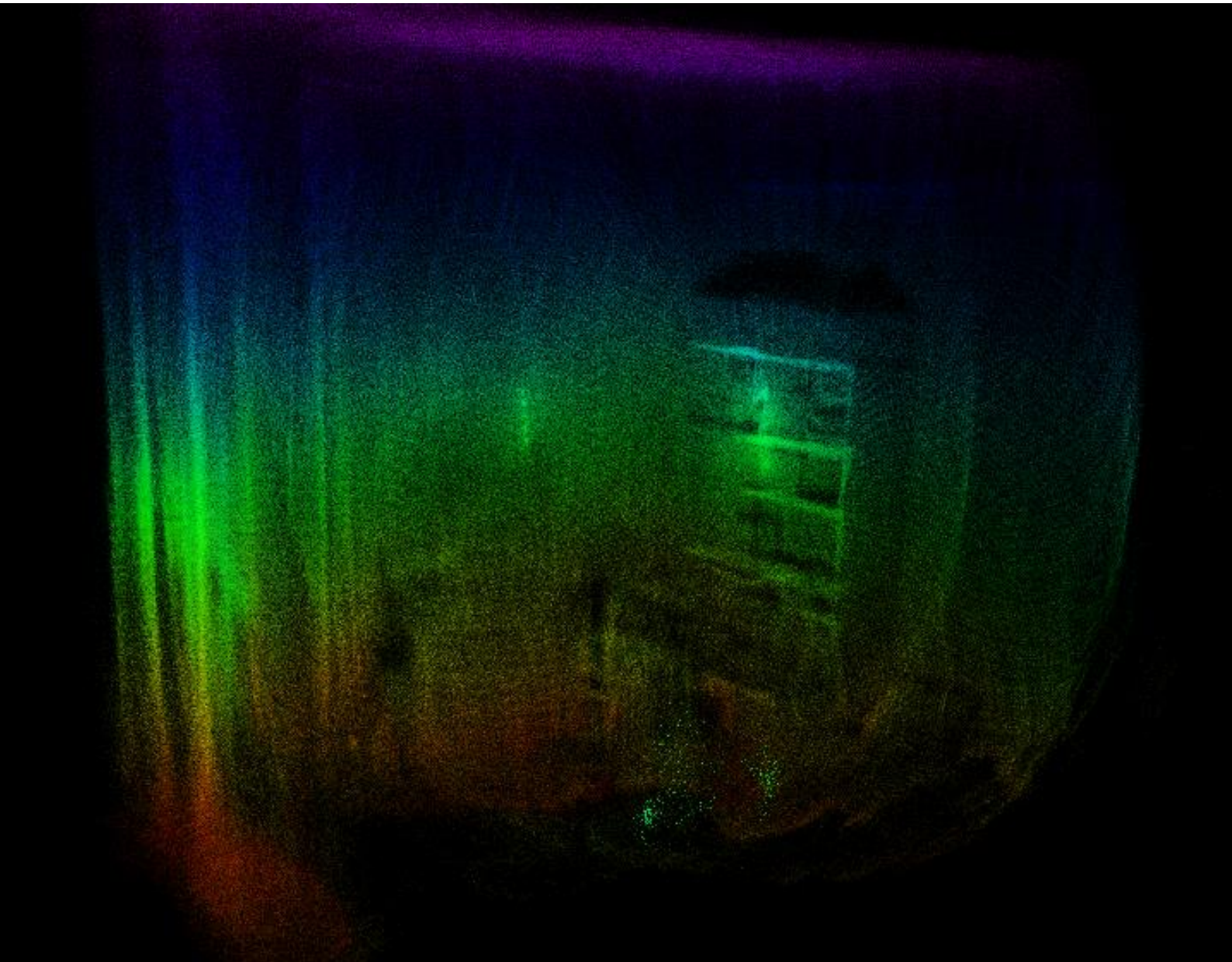
Εικόνα 52: Βελτιωμένοι παράμετροι KISS ICP για εσωτερικούς χώρους



Εικόνα 53: Αρχικό αποτέλεσμα KISS-ICP – καταγραφή εσωτερικού χώρου

4.2.3.3 Εσωτερικές δοκιμές με τον LIVOX MAPPING

Ο LIVOX MAPPING δεν έδωσε τόσο ικανοποιητικό αποτέλεσμα (Εικόνα 54), πάρα την κοινή αρχή λειτουργίας με τον LOAM LIVOX. Το αποτέλεσμα του LIVOX MAPPING στην συγκεκριμένη περίπτωση εμφανίζεται να έχει πολύ περισσότερο θόρυβο, πιθανώς λόγω των παραμέτρων αλλά και της έλλειψης αντίστοιχου μηχανισμού με αυτόν του LOAM LIVOX για την εξομάλυνση θορύβου, με αποτέλεσμα οι λεπτομέρειες του αντικειμένου να μην είναι εύκολα διακριτές. Ενδέχεται ακόμη σε αυτό να συντέλεσε και η παρουσία καθρέφτη στον χώρο που ενδεχομένως να αύξησε τα εσφαλμένα σημεία δυσκολεύοντας έτσι την εύρεση αξιόπιστων χαρακτηριστικών για συνταύτιση.



Εικόνα 54: Αποτέλεσμα LIVOX MAPPING – καταγραφή εσωτερικού χώρου

4.2.3.4 Εσωτερικές δοκιμές με τον FAST-LIO2

Ο FAST-LIO2 απέτυχε σε κάθε περίπτωση να λειτουργήσει σωστά στον εσωτερικό χώρο εμφανίζοντας σφάλματα εκτέλεσης (Εικόνα 55) που αφορούσαν την απουσία επαρκών σημείων με συνέπεια ο αλγόριθμος να αποτυγχάνει διαρκώς.

Σε αυτήν την περίπτωση τα σφάλματα δεν οφείλονται τόσο στις παραμέτρους της μεθόδου, καθώς αυτές είναι ήδη ορισμένες από τον ίδιο τον αλγόριθμο και αφορούσαν τον συγκεκριμένο σαρωτή. Πιθανότατα, τα σφάλματα να οφείλονται στην μεγάλη μείωση του αριθμού των σημείων που εφαρμόζεται από τον αλγόριθμο προκειμένου να διατηρήσει λιγότερα σημεία στην μνήμη κατά την δημιουργία του χάρτη. Προκύπτει λοιπόν πως η συγκεκριμένη μέθοδος υστερεί αρκετά σε εφαρμογές αποτύπωσης εσωτερικών χώρων.

```
[ WARN] [1709124113.772907675]: No point, skip this scan!  
[ WARN] [1709124113.868596723]: Too few input point cloud!  
[ WARN] [1709124113.868853487]: No point, skip this scan!  
[ WARN] [1709124113.965349656]: Too few input point cloud!  
[ WARN] [1709124113.965550543]: No point, skip this scan!  
[ WARN] [1709124114.065758375]: Too few input point cloud!  
[ WARN] [1709124114.167386091]: No point, skip this scan!
```

Εικόνα 55: Σφάλματα κατά την εκτέλεση του FAST-LIO2 σε καταγραφή εσωτερικού χώρου



Εικόνα 56: FAST-LIO2 – καταγραφή εσωτερικού χώρου (Αποτυχία)

4.2.3.5 Εσωτερικές δοκιμές με τον Faster-LIO

Ο Faster-LIO γενικά έτρεξε χωρίς προβλήματα κατά τις δοκιμές εσωτερικού χώρου. Στον τελικό 3D χάρτη (Εικόνα 57), δεν εμφανίζονται παραμορφωμένα σχήματα ή διπλοπροβολές, ωστόσο φαίνεται να υπάρχει αρκετός θόρυβος. Ο θόρυβος αυτός οφείλεται κυρίως στα χαρακτηριστικά του σαρωτή και όχι τόσο στην μέθοδο που χρησιμοποιήθηκε.



Εικόνα 57: Αποτέλεσμα Faster-LIO– καταγραφή εσωτερικού χώρου

Τέλος, επισημαίνεται πως στο πλαίσιο των δοκιμών, έγινε προσπάθεια μετατροπής των δεδομένων από τον LIVOX AVIA στην μορφή που χρησιμοποιεί το MATLAB. Κανένας όμως από τους συγκεκριμένους αλγορίθμους δεν κατάφερε να λειτουργήσει κανονικά και να δημιουργήσει ένα συνενωμένο νέφος σημείων.

5. ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ

Στο κεφάλαιο αυτό γίνεται σύγκριση μεταξύ των σύγχρονων μεθόδων LiDAR-SLAM που αναλύθηκαν και εφαρμόστηκαν στο πλαίσιο της διπλωματικής εργασίας, με γνώμονα την χρηστικότητα τους, την απόδοσή τους και φυσικά την ποιότητα του τελικού αποτελέσματος.

Η χρηστικότητα αφορά κυρίως την ευκολία εφαρμογής της μεθόδου με εξωτερικά δεδομένα από κάποιον τρίτο χρήστη με διαφορετικό σαρωτή και πραγματεύεται κυρίως το τεχνικό μέρος των υλοποιήσεων. Η συγκεκριμένη ανάλυση κρίνεται αρκετά σημαντική η οποία συνήθως παραλείπεται στην βιβλιογραφία.

Η αξιολόγηση της απόδοσης αφορά κυρίως τον χρόνο που χρειάστηκαν οι αλγόριθμοι για να εκτελεστούν και αν αυτό έγινε σε πραγματικό χρόνο. Η γεωμετρική ακρίβεια των αποτελεσμάτων δεν ήταν δυνατό να εξεταστεί ποσοτικά καθώς θα η συλλογή αληθών δεδομένων (ground-truth) δεν ήταν σκοπός της συγκεκριμένης εργασίας. Επίσης, δεν κρίθηκε απαραίτητη η αξιολόγηση της ακρίβειας των αποτελεσμάτων μέσω γνωστών σετ δεδομένων που διατίθενται στο διαδίκτυο, καθώς αυτό έχει ήδη πραγματοποιηθεί από τους δημιουργούς των αλγορίθμων τα συγκριτικά αποτελέσματα είναι δημοσιευμένα στα σχετικά άρθρα. Τέλος, συχνά οι αλγόριθμοι αυτοί έχουν καθορίσει τις τιμές των παραμέτρων έτσι ώστε τα αποτελέσματα να είναι βέλτιστα για τα συγκεκριμένα δεδομένα, με αποτέλεσμα να μην γενικεύονται εύκολα και για αλλά περιβάλλοντα, συνθήκες ή σαρωτές.

Όσον αφορά τα τελικά αποτελέσματα, πραγματοποιήθηκε ενδελεχής παρατήρηση και οπτική σύγκριση αυτών που προέκυψαν σε εξωτερικό και σε εσωτερικό χώρο, καταγράφοντας ενδεχόμενες παραμορφώσεις, θόρυβο, σφάλματα στροφών ή και περιπτώσεις πλήρους αποτυχίας της μεθόδου. Στην σύγκριση αυτή δεν συμμετείχαν οι αλγόριθμοι SLAM σε MATLAB, καθώς εμφανίζουν σημαντικά προβλήματα γενίκευσης σε διαφορετικές καταστάσεις λόγω της ειδικής μορφής δεδομένων που απαιτούν. Επίσης παρουσιάζονται έντονα εξαρτημένες από τις παραμέτρους τους και από τις έτοιμες συναρτήσεις συνένωσης, βελτιστοποίησης και εύρεσης βρόχων. Τέλος, δεν υποστηρίζουν την άμεση σύνδεση του αισθητήρα για εφαρμογές σε πραγματικό χρόνο. Για την σύγκριση των παραγόμενων χαρτών χρησιμοποιήθηκε το ανοιχτό λογισμικό επεξεργασίας και προβολής νεφών σημείων Cloud Compare [110].

5.1 Χρηστικότητα

Την μεγαλύτερη χρηστικότητα μεταξύ των μεθόδων παρουσίασε ο αλγόριθμος KISS ICP ο οποίος εκτελείται σε γλώσσα Python ή C++ ενώ ταυτόχρονα υποστηρίζει και διαφορετικά λειτουργικά συστήματα, σε αντίθεση με την συνηθισμένη ROS-based λογική υλοποίησης που επιβάλλει γλώσσες C++ σε περιβάλλον Linux. Λόγω των χαρακτηριστικών αυτών, πρακτικά απαιτούνται ελάχιστες βιβλιοθήκες λογισμικού τις οποίες φορτώνει από μόνος του ο αλγόριθμος. Επιπλέον, περιλαμβάνονται σαφείς οδηγίες για την λειτουργία του αλγορίθμου ενώ ο κώδικας ανανεώνεται συχνά διορθώνοντας τυχόν σφαλμάτων που προκύπτουν. Ένα ακόμα θετικό χαρακτηριστικό είναι το γεγονός πως οι προγραμματιστές του είναι ενεργοί και απαντούν στα ερωτήματα της κοινότητας μέσω της γνωστής πλατφόρμας GitHub όπου διατίθεται ο κώδικας. Επίσης, σε θεωρητική βάση, ο κώδικας χαρακτηρίζεται από πολύ λιγότερες παραμέτρους (μόλις 7) σε αντίθεση με τους υπολοίπους, τις οποίες μπορεί κανείς εύκολα να παραμετροποιήσει σε ξεχωριστό αρχείο χωρίς να χρειαστεί να επέμβει στον κώδικα. Τέλος, λόγω της σχετικά απλής μεθοδολογία που εφαρμόζει μέσω του αλγορίθμου ICP, μπορεί να λειτουργήσει εύκολα για πολλούς διαφορετικούς σαρωτές, ανεξάρτητα από το οπτικό τους πεδίο και τον τρόπο σάρωσης χωρίς να είναι και απαραίτητη η υποστήριξη από άλλους αισθητήρες όπως τα IMU.

Στη συνέχεια, οι παρεμφερείς αλγόριθμοι FAST-LIO2 και Faster-LIO έχουν μεν υλοποιηθεί σε γλώσσα C++ για λειτουργικό Linux-Ubuntu μέσω χρήσης της βιβλιοθήκης ROS, ωστόσο υποστηρίζουν και νεότερες εκδόσεις του λογισμικού (20.04 και άνω) χωρίς πρόβλημα. Ακόμη, απαιτούν σχετικά μικρό αριθμό βιβλιοθηκών και περιλαμβάνουν σετ προκαθορισμένων παραμέτρων για αρκετούς σαρωτές, παρόλο που και οι ίδιοι παράμετροι μπορούν να λειτουργήσουν αρκετά καλά και σε άλλους σαρωτές παρόμοιας τεχνολογίας. Επιπλέον, οι δημιουργοί του παρέχουν υποστήριξη, απαντάνε σε ερωτήματα και διορθώνουν τυχόν σφάλματα, με την τελευταία ενημέρωση να έχει πραγματοποιηθεί πριν από πέντε μήνες και για τους δυο αλγορίθμους (Οκτώβριος 2023).

Στα αρνητικά των δύο μεθόδων είναι η ανάγκη μετρήσεων από IMU, γεγονός το οποίο αυξάνει την δυσκολία στην σύνδεση και την καταγραφή παρατηρήσεων, ειδικά σε σαρωτές που δεν έχουν ενσωματωμένο IMU. Σε γενικές γραμμές ωστόσο περιλαμβάνονται σχετικές οδηγίες, με αποτέλεσμα χρήστες ακόμα και με βασικές γνώσεις C++ και Linux να μπορούν να χρησιμοποιήσουν τους συγκεκριμένους αλγορίθμους. Από θεωρητικής άποψης αποτελούν εύχρηστους αλγορίθμους καθώς δεν απαιτείται η εξαγωγή χαρακτηριστικών, η διόρθωση του θολώματος από κίνηση (motion blur) και ο υπολογισμός των προσεγγιστικών τιμών γίνονται με βάση το IMU, και οι παράμετροι τους γίνονται ευκολότερα κατανοητοί αφού αφορούν κυρίως βασικά χαρακτηριστικά του σαρωτή όπως το οπτικό πεδίο και το εύρος σάρωσης.

Ως λιγότερο χρηστικοί αλγόριθμοι εμφανίζονται οι LIVOX MAPPING και LOAM LIVOX οι οποίοι είναι και οι παλιότεροι μεταξύ των πέντε αφού παρουσιάστηκαν την περίοδο 2019-2020. Οι ενημερώσεις στον κώδικα και οι απαντήσεις στις ερωτήσεις των χρηστών γίνονταν έως και το 2021. Επιπλέον, οι αλγόριθμοι έχουν σχεδιαστεί και δοκιμαστεί στις παλιότερες εκδόσεις Ubuntu 16.04 και 18.04. Παρόλ' αυτά δεν αναφέρουν τις εκδόσεις βιβλιοθηκών που έχουν χρησιμοποιηθεί, ενώ δεν λειτουργούν με τις πιο πρόσφατες εκδόσεις. Την μεγαλύτερη δυσκολία υλοποίησης εμφάνισε ο αλγόριθμος LOAM LIVOX καθώς μια βασική βιβλιοθήκη του (Ceres Solver) έχει αλλάξει σημαντικά, με αποτέλεσμα να πρέπει να γίνουν και αλλαγές στον κώδικα για να λειτουργήσει.

Σε γενικές γραμμές αφορούν αλγορίθμους οι οποίοι είναι σχετικά δύσκολο να υλοποιηθούν ως έχουν λόγω προβλημάτων από αναβαθμίσεις σε λειτουργικά συστήματα και βιβλιοθήκες, ενώ όταν τελικά εκτελεστούν (ύστερα από τις απαραίτητες διορθώσεις), υποστηρίζουν συγκεκριμένους σαρωτές και παρέχουν τιμές παραμέτρων μόνο για αυτούς. Σε θεωρητική βάση, όντας αλγόριθμοι που χρησιμοποιούν χαρακτηριστικά, έχουν περισσότερες και αρκετά πιο πολύπλοκες παραμέτρους οι οποίες αφορούν την εξαγωγή των χαρακτηριστικών, ενώ παρουσιάζουν μεγαλύτερη εξάρτηση από τον τύπο του σαρωτή. Συνοπτικά, οι αλγόριθμοι LOAM LIVOX και LIVOX MAPPING για να εκτελεστούν χρειάζονται αρκετά πιο εξειδικευμένες γνώσεις σε γλώσσες C++ και ROS, ενώ απαιτούν καλή γνώση της τεχνολογίας των συστημάτων LiDAR αλλά και των μεθόδων SLAM.

5.2 Απόδοση

Όσον αφορά την απόδοση των αλγορίθμων ως προς τον χρόνο εκτέλεσης, καθώς οι καταγραφές έγιναν με συχνότητα 10 Hz, αυτό συνεπάγεται ότι κάθε σάρωση (ή αλλιώς frame της καταγραφής), θα πρέπει να έχει επεξεργαστεί σε μέγιστο χρόνο 100 ms ώστε να χαρακτηρίζεται εφαρμογή πραγματικού χρόνου.

Από τους αλγορίθμους που μελετήθηκαν στο σύστημα VM (Virtual Machine) καθορισμένων προδιαγραφών, οι αλγόριθμοι LOAM LIVOX και LIVOX MAPPING δεν κατάφεραν να ολοκληρωθούν ταυτόχρονα με την καταγραφή και εμφάνισαν καθυστερήσεις. Το γεγονός αυτό υποδεικνύει ότι δεν μπόρεσαν να ολοκληρώσουν την επεξεργασία των νεφών σημείων εγκαίρως. Σε αυτή την περίπτωση ο μέσος χρόνος επεξεργασίας ενός νέφους μπορεί να υπολογιστεί ως ο χρόνος που έκανε να εκτελεστεί ο αλγόριθμος, δια τον συνολικό αριθμό νεφών σημείων που καταγράφηκαν. Στην περίπτωση του LOAM LIVOX, υπήρχε μια καθυστέρηση της τάξης του 20-30% και συνεπώς κάθε επεξεργασία διήρκεσε περίπου 120-130 ms. Πράγματι, από τον χρόνο επεξεργασίας που αποθηκεύονται αυτόματα για κάθε βήμα σάρωσης από τον αλγόριθμο, (Εικόνα 58), επιβεβαιώνεται πως ο συγκεκριμένος αφορά τον τυπικό χρόνο ολοκλήρωσης ενός frame (βήματος).

```

425 [Timer](140612907632384): Wait sync cost time = 19.91 ms
426 [Timer](140612907632384): Query points for match cost time = 20.82 ms
427 [Timer](140612907632384): Build kdtree cost time = 0.00 ms
428 [Timer](140612907632384): Pose optimization cost time = 3.77 ms
429 [Timer](140612907632384): Add new frame cost time = 6.40 ms
430 [Timer](140612907632384): Frame process cost time = 31.15 ms
431 [Timer](140613158766336): Prepare to enter thread cost time = 31.33 ms
432 -----
433 [Timer](140612907632384): Wait sync cost time = 14.58 ms
434 [Timer](140612907632384): Query points for match cost time = 14.86 ms
435 [Timer](140612907632384): Build kdtree cost time = 0.00 ms
436 [Timer](140612907632384): Pose optimization cost time = 11.65 ms
437 [Timer](140612907632384): Add new frame cost time = 7.61 ms
438 [Timer](140612907632384): Frame process cost time = 34.26 ms
439 [Timer](140613158766336): Prepare to enter thread cost time = 35.54 ms

```

Εικόνα 58: Παράδειγμα χρόνου εκτέλεσης LOAM LIVOX

Αντίστοιχα αποτελέσματα έδωσε και ο αλγόριθμος LIVOX MAPPING καθώς παρουσίασε καθυστερήσεις της τάξης του 10-20% και συνεπώς είχε συνολικό χρόνο βήματος περίπου 110-120 ms, (το οποίο είναι και λογικό αφού η μέθοδος είναι αντίστοιχη του LOAM LIVOX με κάποιες λιγότερες λειτουργίες). Η αποτυχία των αλγορίθμων ενδεχομένως να οφείλεται στον διαφορετικού τύπου σαρωτή σε σχέση με αυτόν για τον οποίο σχεδιάστηκε η μέθοδος ή και σε προβλήματα λόγω της χρήσης παλαιότερων βιβλιοθηκών. Αντίθετα, οι αλγόριθμοι KISS-ICP, FAST-LIO2 και Faster-LIO κατάφεραν σε όλες τις περιπτώσεις να λειτουργήσουν σε πραγματικό χρόνο. Συγκεκριμένα, ο αλγόριθμος KISS-ICP, ο οποίος μετράει την ταχύτητα εκτέλεσης σε frames/sec, κατέγραψε μέσο ορό ταχύτητας τα 13 frames/sec, άρα ο συγκεκριμένος αλγόριθμος επεξεργάζεται περίπου ένα νέφος σημείων ανά 77 ms. Ο FAST-LIO2 μπορεί επίσης να καταγράψει τον χρόνο που χρειάστηκε να εκτελεστεί υπό μορφή χρονικής αναφοράς (Πίνακας 4). Έτσι, ο μέσος συνολικός χρόνος (total time) μεταξύ όλων των δοκιμών για την επεξεργασία ενός frame της μεθόδου αυτής, υπολογίστηκε σε 9.85 ms και συνεπώς η συγκεκριμένη μέθοδος είχε ρυθμό εκτέλεσης 102 frames/s .

Πίνακας 4: Τμήμα χρονικής αναφοράς FAST-LIO2

	A	B	C	D	E	F	G	H	I	J	K
1	time_stamp	total time	scan point size	incremental time	search time	delete size	delete time	tree size st	tree size end	add point size	preprocess time
2	296.49960406	0.011211	4100	0.00403913	0	0	9E-07	2316	3923	1911	0
3	296.5994405	0.008381	4016	0.00243606	0	0	2.9E-07	3923	4985	1502	0.0005163
4	296.69928916	0.0088595	4056	0.00296642	0	0	2.3E-07	4985	5991	1418	0.00019444
5	296.79992658	0.008906	4127	0.00213708	0	0	5.7E-07	5991	6950	1253	0.00028484
6	296.89976586	0.0097201	4128	0.00288628	0	0	6.4E-07	6950	7768	1227	0.00021434
7	296.99960786	0.0093984	4113	0.00226022	0	0	5.4E-07	7548	8609	1085	0.00015338
8	297.09945444	0.0091777	3930	0.00180384	0	0	6.3E-07	8261	9133	1058	0.00012527
9	297.1992906	0.0080504	3856	0.00191547	0	0	9.1E-07	9133	9936	1074	0.00133808
10	297.29992154	0.0099166	3899	0.00261811	0	0	7.3E-07	9936	10677	1048	0.00014375
11	297.3997639	0.0088727	3728	0.00224669	0	0	1.52E-06	10677	11315	971	0.00014652
12	297.49960394	0.0094945	3655	0.00251019	0	0	7.1E-07	11315	11911	956	0.00013939
13	297.59944562	0.0084849	3484	0.00220799	0	0	6.4E-07	11911	12460	943	0.00021757
14	297.69928558	0.008295	3301	0.0021882	0	0	1.03E-06	12460	12998	951	0.0001608
15	297.79993182	0.0096559	3247	0.00284793	0	0	6.9E-07	12857	13493	919	0.00017912
16	297.89978242	0.0105712	3528	0.00304298	0	0	8.9E-07	13493	14143	1009	0.00015552

Τέλος, ο Faster-LIO αποτέλεσε τον ταχύτερο αλγόριθμο, καθώς από τις καταγραφές χρόνου που πραγματοποίησε ο αλγόριθμος ανά επιμέρους τμήμα (Εικόνα 59), προέκυψε πως για ένα ολόκληρο βήμα ο αλγόριθμος είχε έναν μέσο συνολικό χρόνο εκτέλεσης 7.65 ms ανά νέφος σημείων, το οποίο αντιστοιχεί σε ρυθμό εκτέλεσης 131 frames/sec. Τέλος, στον Πίνακα 5 γίνεται η σύγκριση χρόνου εκτέλεσης μεταξύ των FAST-LIO2 και Faster-LIO που αποτέλεσαν και τις δυο ταχύτερες μεθόδους.

```

6348 utils.h:52] >>> ===== Printing run time =====
6348 utils.h:54] > [ Ivox Add Points ] average time usage: 0.142001 ms , called times: 1549
6348 utils.h:54] > [ Incremental Mapping ] average time usage: 0.398642 ms , called times: 1549
6348 utils.h:54] > [ ObsModel (IEKF Build Jacobian) ] average time usage: 0.0602908 ms , called times: 6195
6348 utils.h:54] > [ ObsModel (Lidar Match) ] average time usage: 0.889637 ms , called times: 6195
6348 utils.h:54] > [ Downsample PointCloud ] average time usage: 0.267768 ms , called times: 1549
6348 utils.h:54] > [ IEKF Solve and Update ] average time usage: 4.72774 ms , called times: 1549
6348 utils.h:54] > [ Preprocess (Livox) ] average time usage: 0.277737 ms , called times: 1552
6348 utils.h:54] > [ Undistort Pcl ] average time usage: 0.9103 ms , called times: 1550
6348 utils.h:59] >>> ===== Printing run time end =====

```

Εικόνα 59: Στιγμιότυπο αναφοράς χρόνων εκτέλεσης του Faster-LIO

Πίνακας 5: Σύγκριση χρόνου επεξεργασίας FAST-LIO2 με Faster-LIO

FAST-LIO2	0.009848	s	102	frames/s
TOTAL	9.85	ms		
Faster-LIO	0.007647	s	131	frames/s
TOTAL	7.65	ms		
% faster	0.2875817			

Συνολικά λοιπόν, από τα παραπάνω προκύπτει πως ο πιο αργός αλγόριθμος είναι ο LOAM LIVOX, με τον χρόνο εκτέλεσης ανά νέφος σημείων να κυμαίνεται μεταξύ 120-130 ms. Ακολουθεί ο παρόμοιας λογικής αλγόριθμος LIVOX MAPPING ο οποίος ήταν ελαφρώς ταχύτερος με χρόνο εκτέλεσης τα 110-120 ms. Άρα λοιπόν κανένας από αυτούς τους δυο αλγορίθμους δεν κατάφερε να λειτουργήσει σε πραγματικό χρόνο χωρίς καθυστερήσεις καθώς για να συμβεί αυτό για καταγραφές 10Hz, θα έπρεπε να επιτευχθεί χρόνος ίσος ή μικρότερος από 100 ms.

Αντιθέτως, οι υπόλοιποι τρεις αλγόριθμοι οι οποίοι μελετήθηκαν κατάφεραν να ολοκληρωθούν σε πραγματικό χρόνο με τον KISS ICP να χρειάζεται για την επεξεργασία 77 ms ανά σάρωση, τον αλγόριθμο FAST-LIO2 με μόλις 9.85 ms, και τέλος τον Faster-LIO, ο οποίος ήταν ο ταχύτερος και αποδοτικότερος όλων με χρόνο επεξεργασίας ανά νέφος ίσο με 7.65 ms. Τα αποτελέσματα αυτά παρουσιάζονται συνοπτικά κατά φθίνουσα σειρά στον Πίνακα 6.

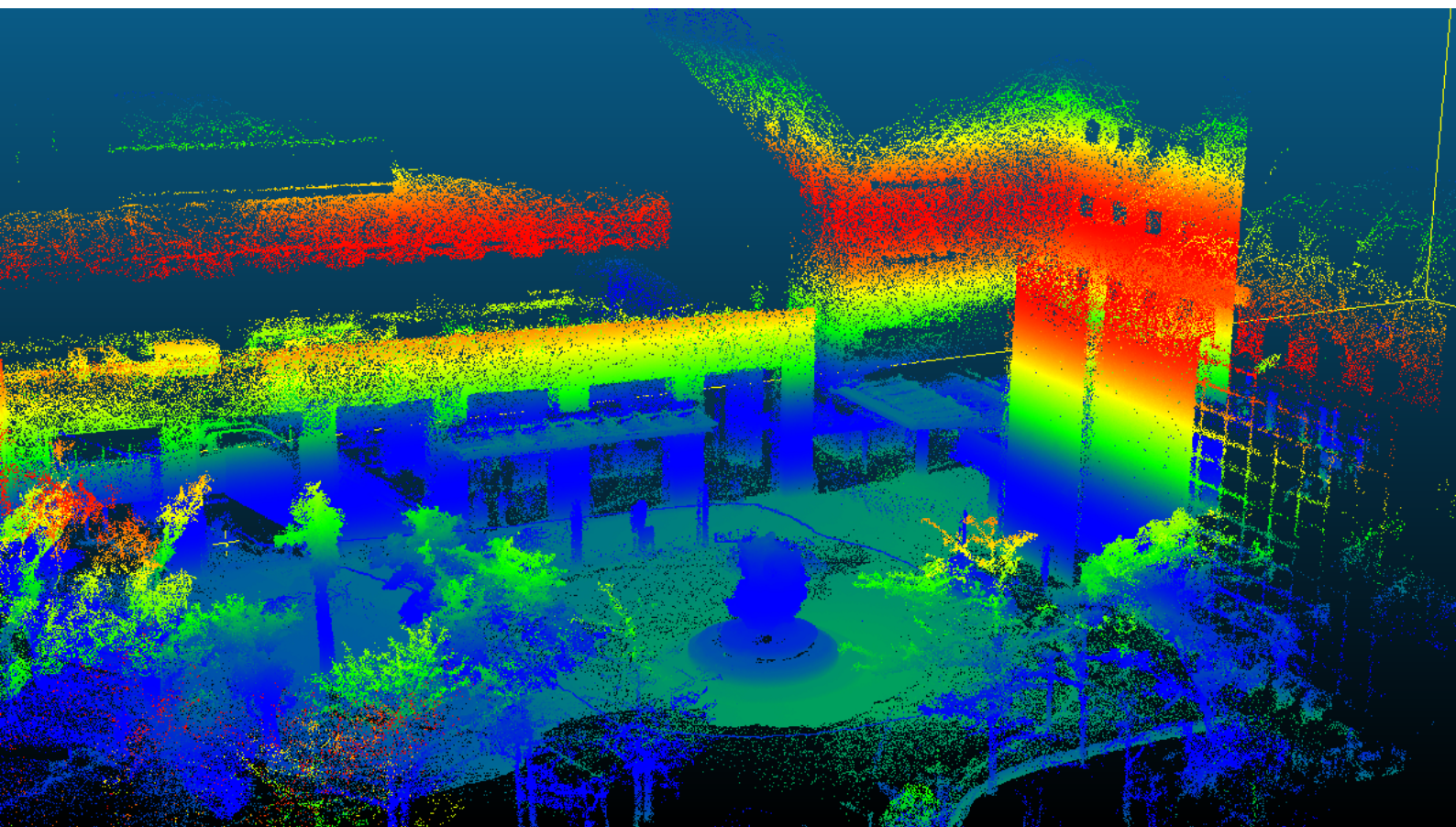
Πίνακας 6: Συγκριτική παρουσίαση των μέσων χρόνων επεξεργασίας των αλγορίθμων

	Average time per frame	Unit
LOAM LIVOX	130	ms
LIVOX MAPPING	120	ms
KISS-ICP	77	ms
FAST-LIO2	9.85	ms
Faster-LIO	7.65	ms

5.3 Αποτελέσματα

Σε αυτήν την υποενότητα γίνεται μια επισκόπηση και αξιολόγηση των αποτελεσμάτων σε περιβάλλον Cloud Compare και στο rviz το οποίο είναι ένα εργαλείο οπτικοποίηση νεφών σημείων του ROS. Στόχος της επισκόπησης αυτής ήταν η ποιοτική αξιολόγηση και ο εντοπισμός διαφορών (ενδεχόμενο θόρυβο, διπλοπροβολές, στροφές ή τυχόν παραμορφώσεις) στα αποτελέσματα των διαφορετικών μεθόδων που εφαρμοστήκαν πάνω στην ίδια καταγραφή.

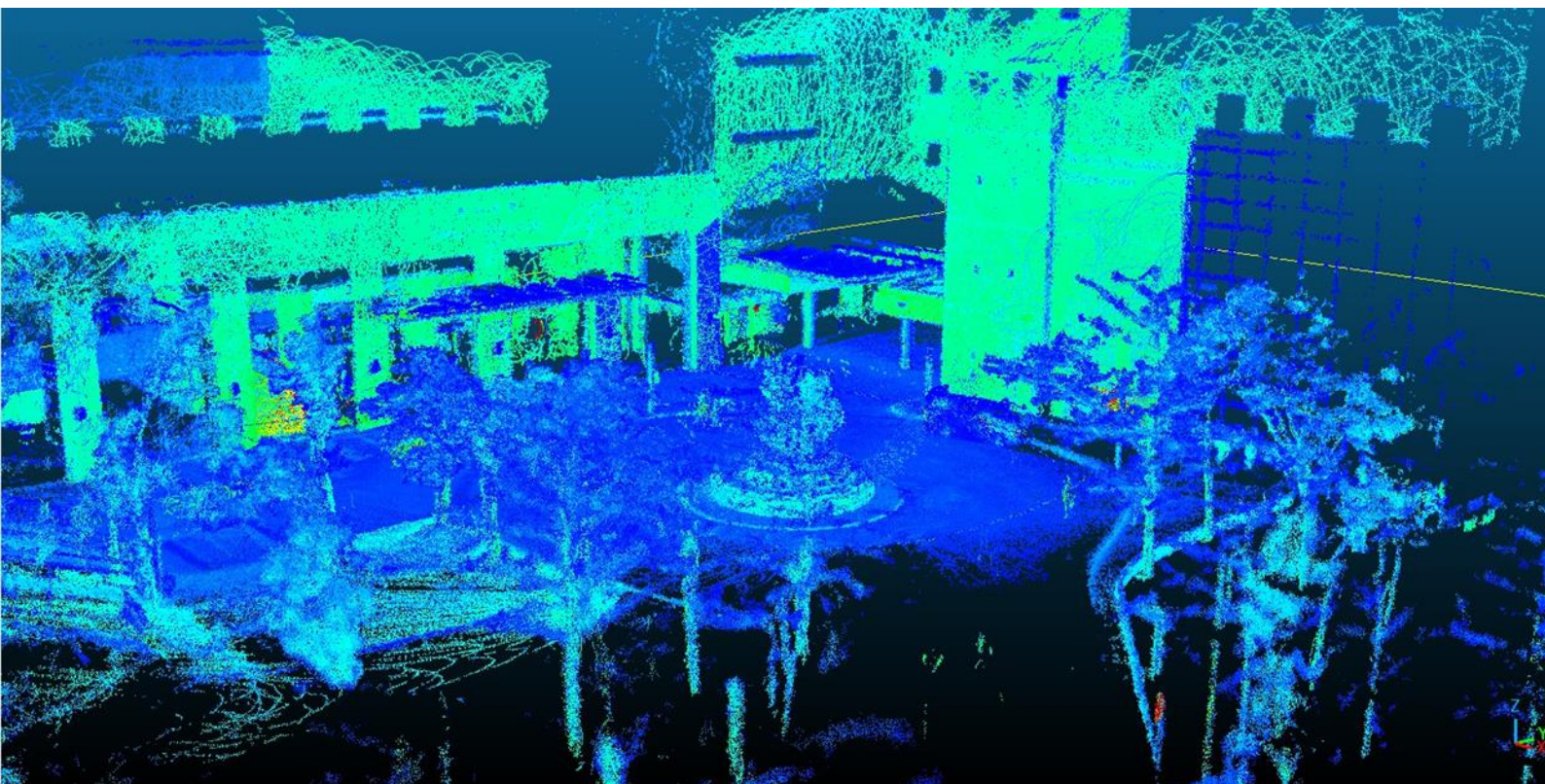
Συγκεκριμένα ο αλγόριθμος LOAM LIVOX, όπως αναφέρθηκε και προηγουμένως, έδωσε αρκετά ικανοποιητικά αποτέλεσμα για τα δοκιμαστικά δεδομένα, γεγονός το οποίο διαπιστώθηκε και μετά την εισαγωγή του χάρτη στο Cloud Compare (Εικόνα 60). Εντούτοις, το νέφος σημείων είναι αρκετά πυκνό (20 εκατομμύρια σημεία) ενώ παράλληλα παρατηρείται σχετικός θόρυβος από τις σαρώσεις.



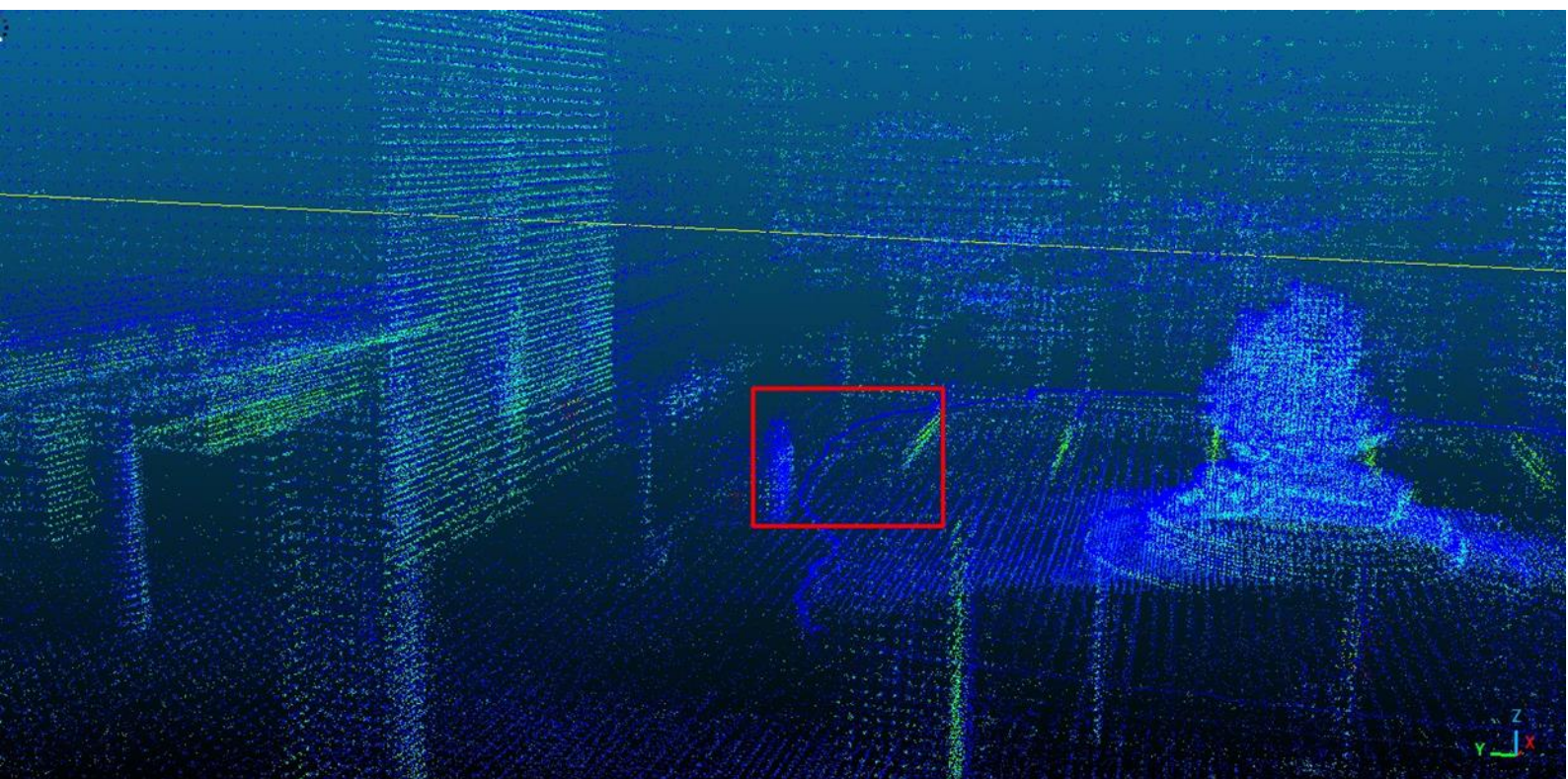
Εικόνα 60: Στιγμιότυπο αποτελέσματος LOAM LIVOX – CYT_02

Έχοντας ως αναφορά τα ίδια δεδομένα, το αποτέλεσμα του αλγορίθμου LIVOX MAPPING παρουσίασε εντονότερα φαινόμενα θορύβου όπως φαίνεται από την Εικόνα 61. Κατά την σύγκριση μεταξύ των δυο παρατηρήθηκε ότι υπάρχει επίσης μια μικρή απόκλιση της τροχιάς του σαρωτή με την μεγαλύτερη διαφορά να παρατηρείται στον άξονα των υψομέτρων (Εικόνα 62). Αντίθετα, στις δοκιμές των αλγορίθμων που πραγματοποιήθηκαν με τον σαρωτή LIVOX AVIA παρόλο που και πάλι τα αποτελέσματα ήταν παρόμοια, ο αλγόριθμος LIVOX MAPPING εμφάνισε

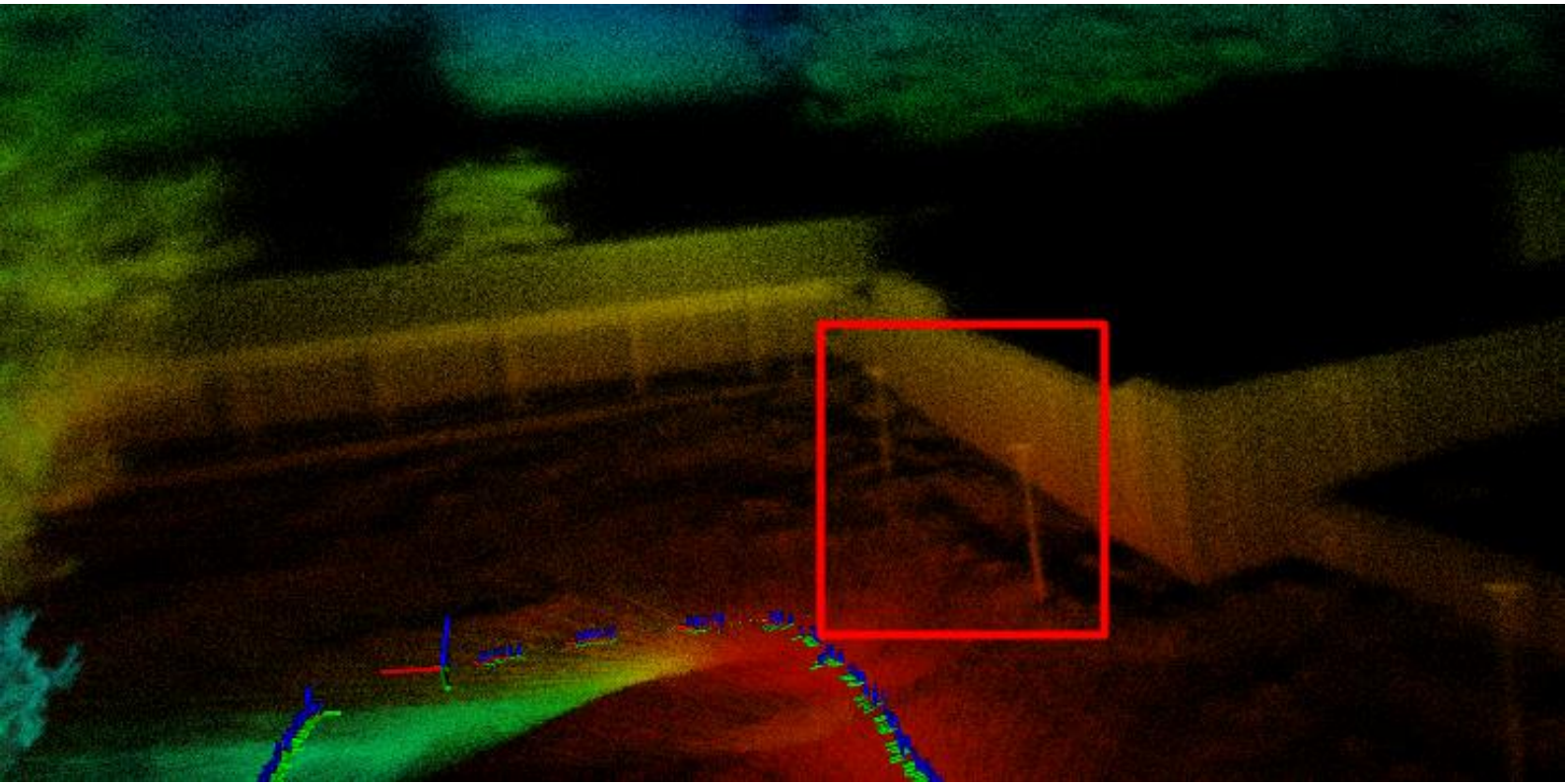
ελαφρώς λιγότερο θόρυβο, γεγονός που εκδηλώνεται κυρίως σε λεπτά γραμμικά στοιχεία όπως στύλοι φωτισμού, όπου απεικονίζονται σχετικά καθαρότερα στον LIVOX MAPPING (Εικόνα 64) από ότι στον LOAM LIVOX (Εικόνα 63). Ωστόσο, σε γενικές γραμμές τα αποτελέσματα είναι πανομοιότυπα.



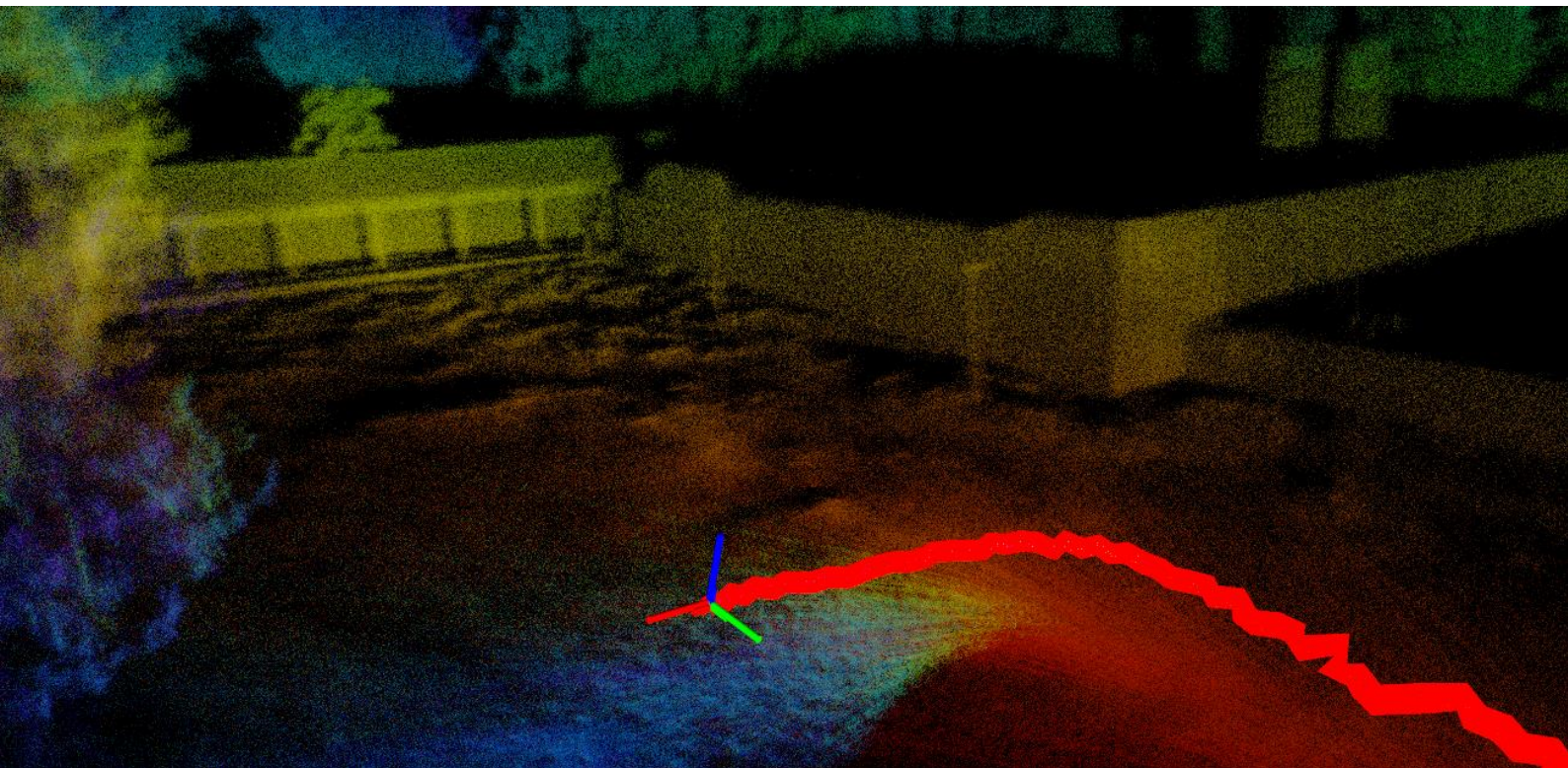
Εικόνα 61: Στιγμιότυπο αποτελέσματος LIVOX MAPPING – CYT_02



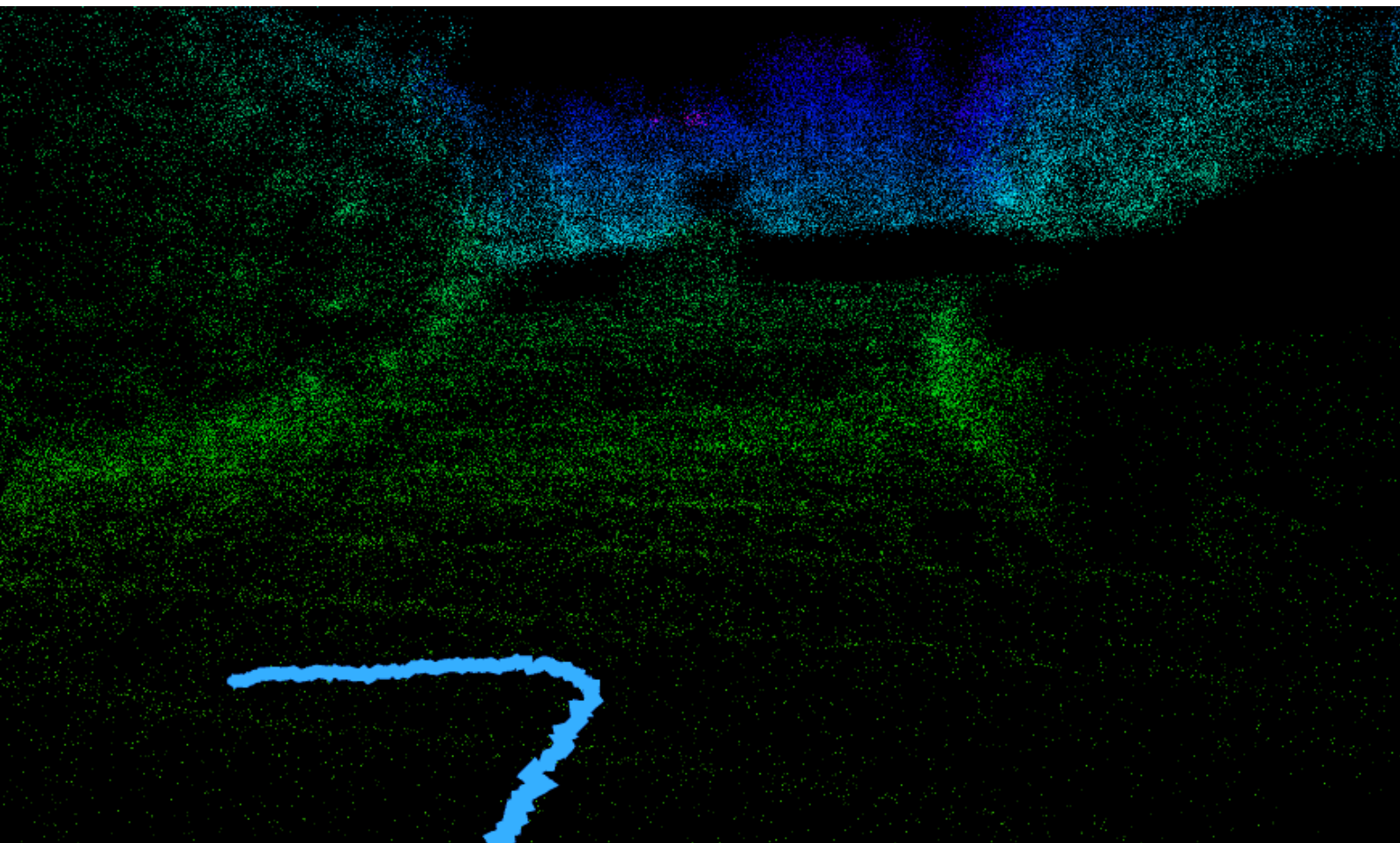
Εικόνα 62: Σύγκριση των αποτελεσμάτων του LOAM LIVOX και LIVOX MAPPING για το dataset CYT_02



Εικόνα 63: Αποτέλεσμα LOAM LIVOX – καταγραφή 8



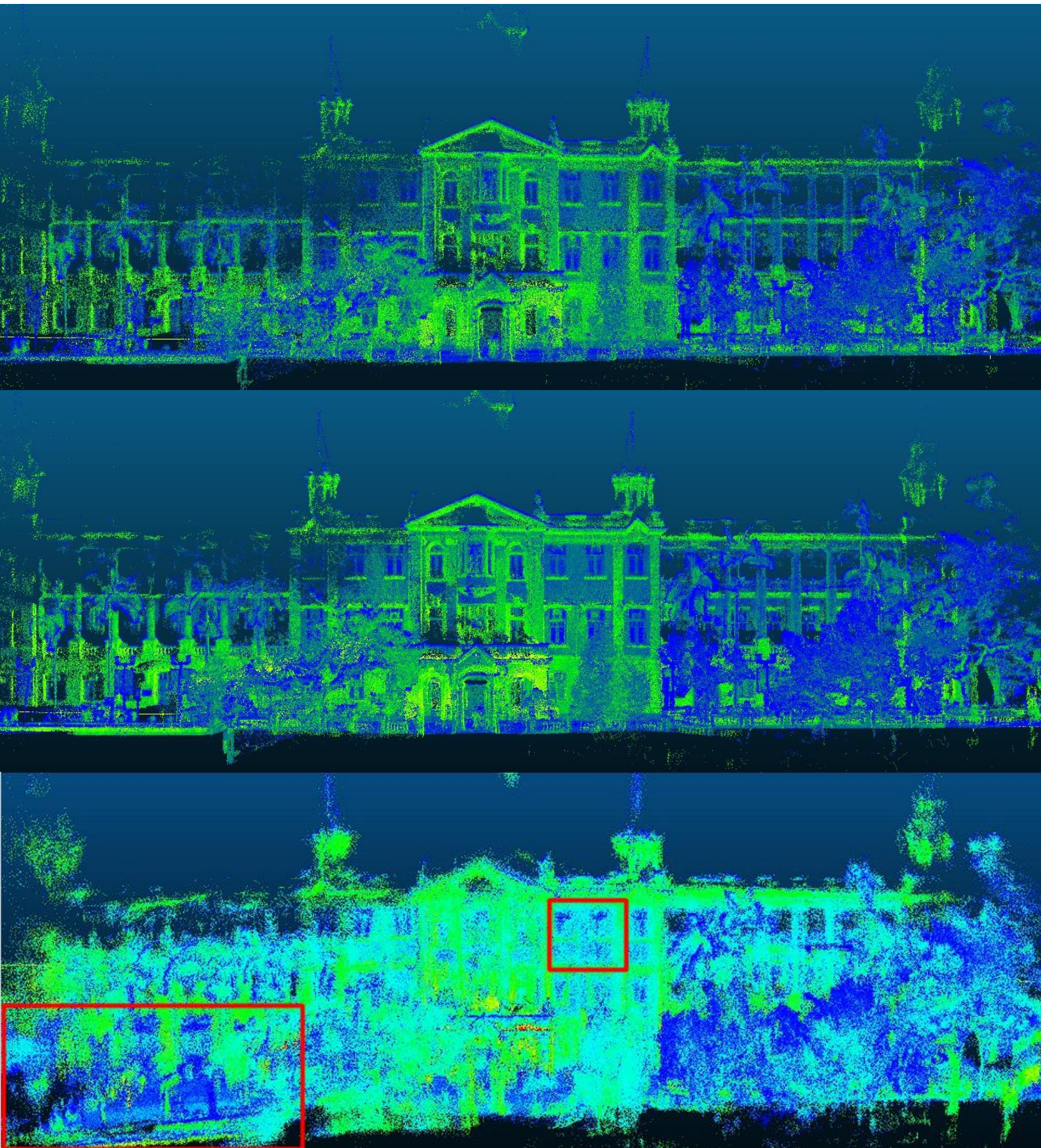
Εικόνα 64: Αποτέλεσμα LIVOX MAPPING – καταγραφή 8



Εικόνα 65: Αποτέλεσμα KISS-ICP - καταγραφή 8

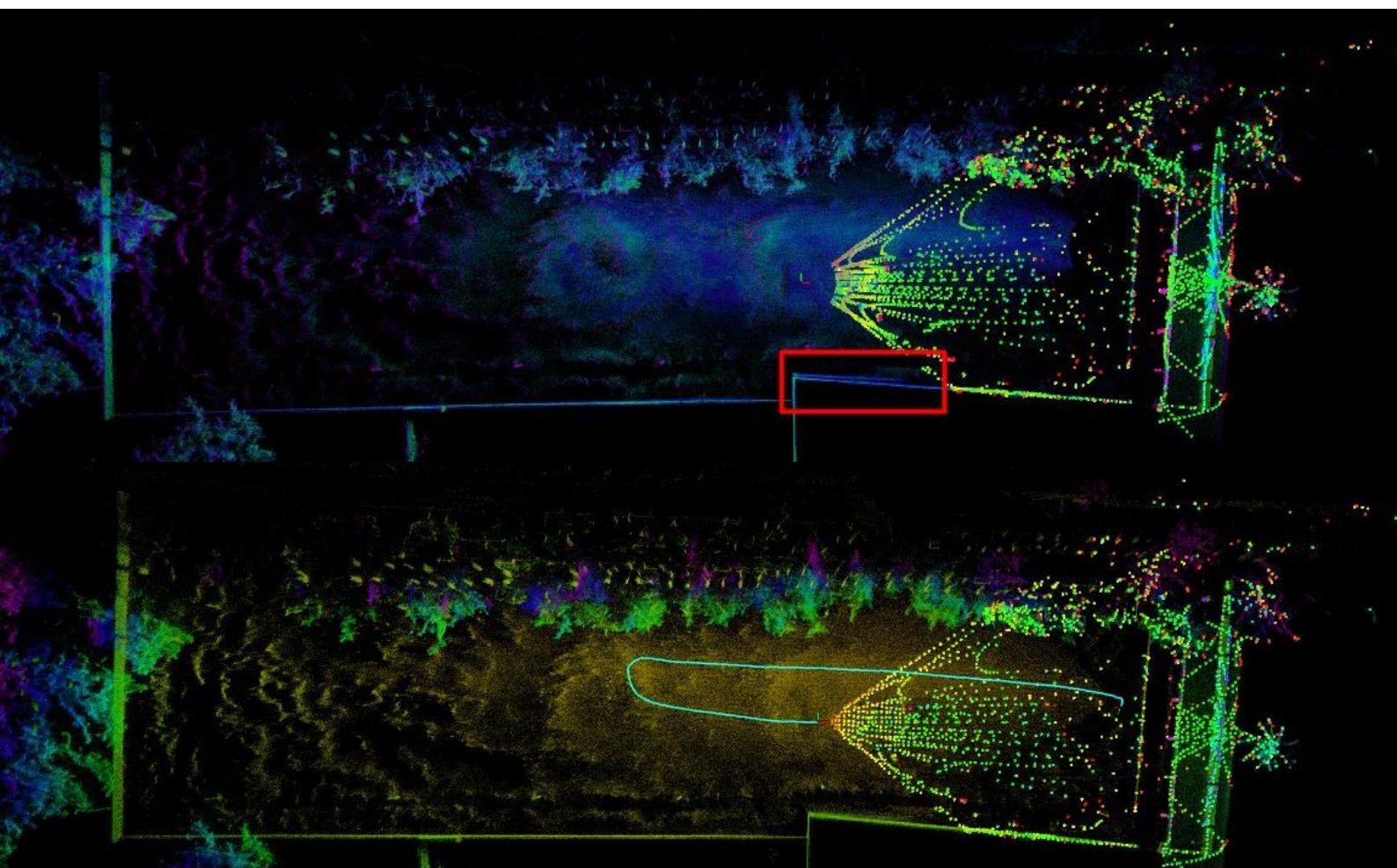
Όσον αφορά τον αλγόριθμο KISS ICP, αυτός έδωσε τα πιο αραιά αποτέλεσμα (Εικόνα 65) και έτσι είναι σχετικά δύσκολο να διακρίνει κανείς οπτικά τις λεπτομέρειες. Επιπλέον, ο KISS-ICP δεν παρέχει ούτε η άλλως δυνατότητα αποθήκευσης του χάρτη και έτσι δεν μπορεί να συγκριθεί το αποτέλεσμα με τις υπόλοιπες μεθόδους. Ωστόσο σε γενικές γραμμές δεν απέτυχε σε καμία μέθοδο. Οι κύριες συγκρίσεις που προκύπτουν τελικά είναι μεταξύ του αλγορίθμου LIVOX MAPPING (καθώς έχει πανομοιότυπο αποτέλεσμα με ελαφρώς λιγότερο θόρυβο από τον LOAM LIVOX στις δοκιμές με τον AVIA) και των FAST-LIO2 και Faster-LIO καθώς μπορούσαν να χρησιμοποιήσουν και τα ίδια δεδομένα εισόδου.

Μεταξύ τους, οι FAST-LIO2 και Faster-LIO είχαν επίσης παρόμοια αποτελέσματα, με τα νέφη σημείων να είναι αρκετά αραιότερα συγκρινόμενα με τα αντίστοιχα των LOAM LIVOX και LIVOX MAPPING (περίπου κατά 50% αραιότερα), όμως ήταν πολύ πιο λεπτομερή, είχαν πολύ λιγότερο εμφανή θόρυβο και δεν παρουσίαζαν σφάλματα από στροφές ή διπλοπροβολές σχεδόν σε καμία περίπτωση. Για παράδειγμα, στα αποτελέσματα από τα δοκιμαστικά δεδομένα Outdoor mainbuilding (Εικόνα 66) δεν μπορούν να παρατηρηθούν διαφορές μεταξύ FAST-LIO2 και Faster-LIO, ενώ το αποτέλεσμα του LIVOX MAPPING εμφανίζεται πολύ πιο θορυβώδες.

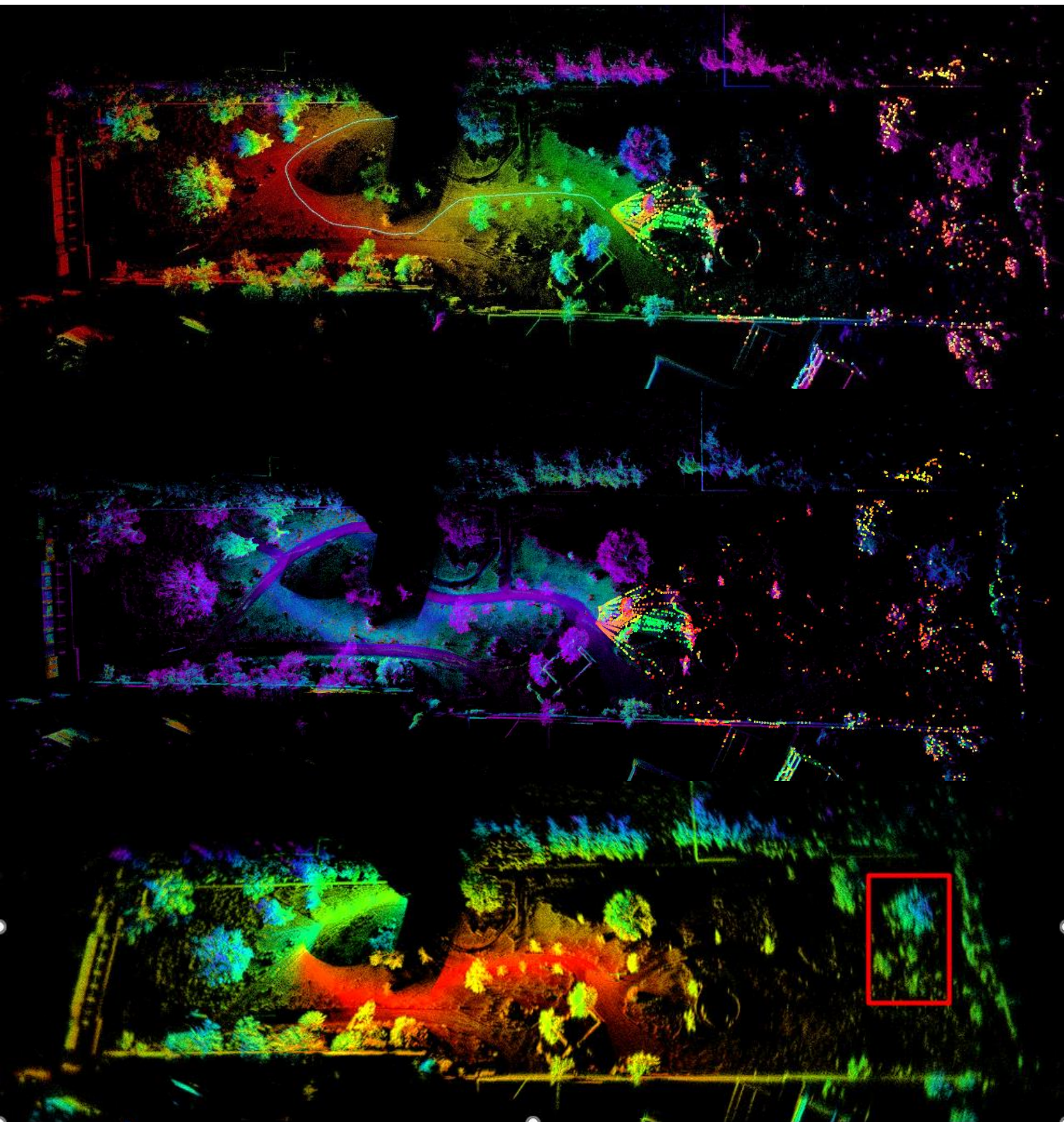


Εικόνα 66: Συγκριτικά αποτελέσματα – outdoor mainbuilding (Πάνω μέρος FAST-LIO2, στην μέση Faster-LIO, κάτω LIVOX MAPPING)

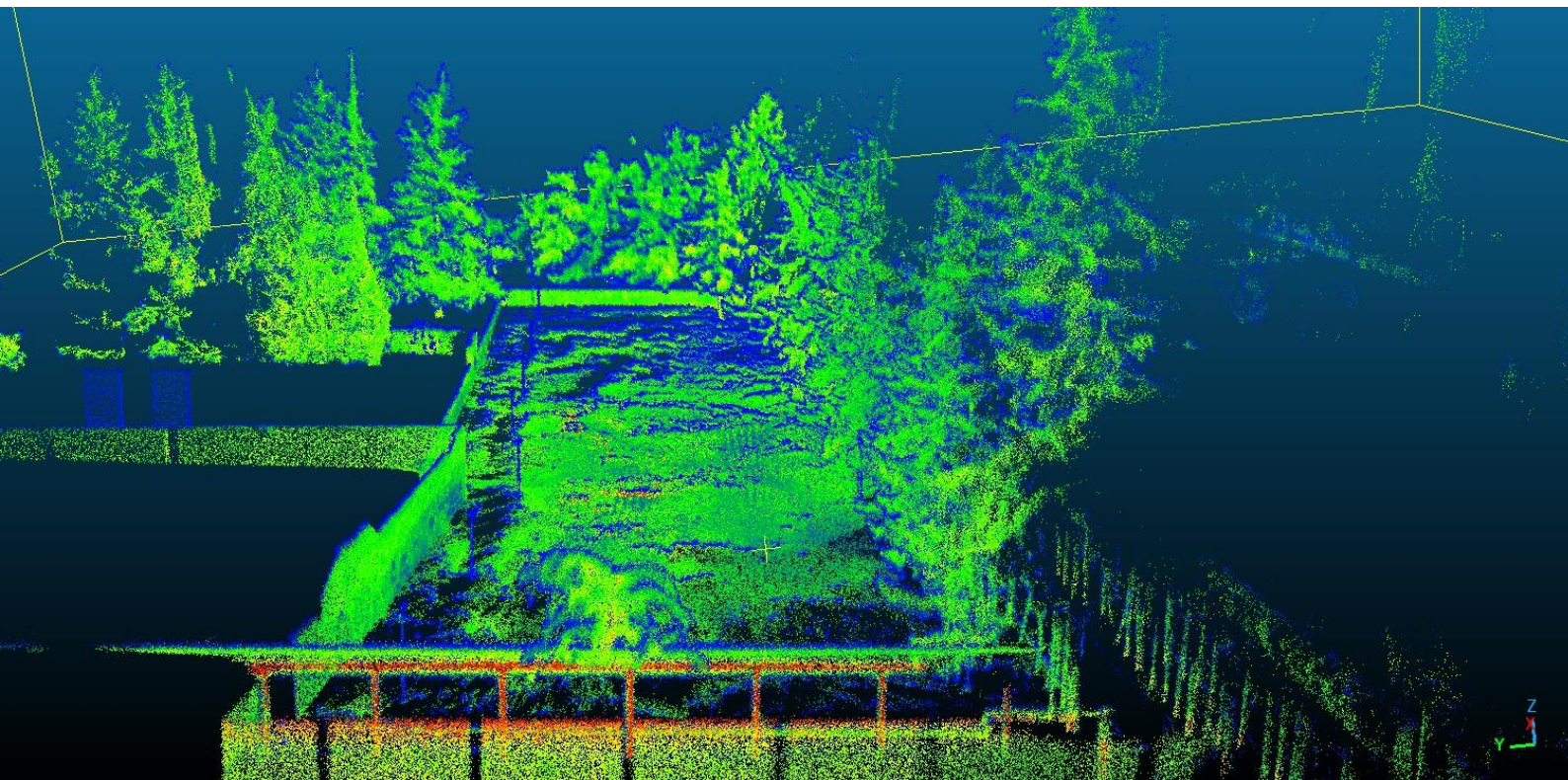
Το συγκεκριμένο σετ δεδομένων ήταν αρκετά απαιτητικό, καθώς ο σαρωτής είχε κινηθεί γρηγορά ενώ υπήρχαν ταυτόχρονα πολλές και απότομες στροφές. Σε γενικές γραμμές ο αλγόριθμος LIVOX MAPPING τα πήγε αρκετά καλά αν και είχε κάποιες αποτυχίες κυρίως στις μεγαλύτερες και δυσκολότερες καταγραφές οι οποίες πραγματοποιήθηκαν,. Από τους αλγορίθμους FAST-LIO2 και Faster-LIO, μόνο ο FAST-LIO2 αντιμετώπισε πρόβλημα διπλοπροβολής στην οποία ο Faster-LIO εμφάνισε μόνο κάποιο θόρυβο. Σε όλες τις υπόλοιπες εξωτερικές δοκιμές οι συγκεκριμένοι δυο αλγόριθμοι ήταν επιτυχείς. Στην Εικόνα 67 παρουσιάζεται η δοκιμή στην οποία ο Faster-LIO είχε καλύτερα αποτελέσματα σε αντίθεση με τον FAST-LIO2 όπου εμφάνισε σφάλμα. Στο παράδειγμα της Εικόνα 68 ο αλγόριθμος LIVOX MAPPING, παρότι εκτελέστηκε κανονικά, είχε αποτέλεσμα κατώτερης ποιότητας σε σχέση με τους άλλους δυο αλγορίθμους. Χαρακτηριστικό είναι ένα θόλωμα που εμφανίζεται στα μακρινά δέντρα τα οποία σαρώθηκαν στο τέλος της καταγραφής και πιθανότατα αφορά συσσώρευση σφαλμάτων από στροφή. Τέλος, παρουσιάζονται ακόμα τρεις δοκιμές (Εικόνες 69-77) στις οποίες οι αλγόριθμοι FAST-LIO2 και Faster-LIO έχουν δώσει αντίστοιχα ικανοποιητικά αποτελέσματα, ενώ ο αλγόριθμος LIVOX MAPPING (και αντίστοιχα ο LOAM LIVOX) έδωσε αποτελέσματα με θόρυβο όπου δεν διακρίνονται τόσο εύκολα οι λεπτομέρειες.



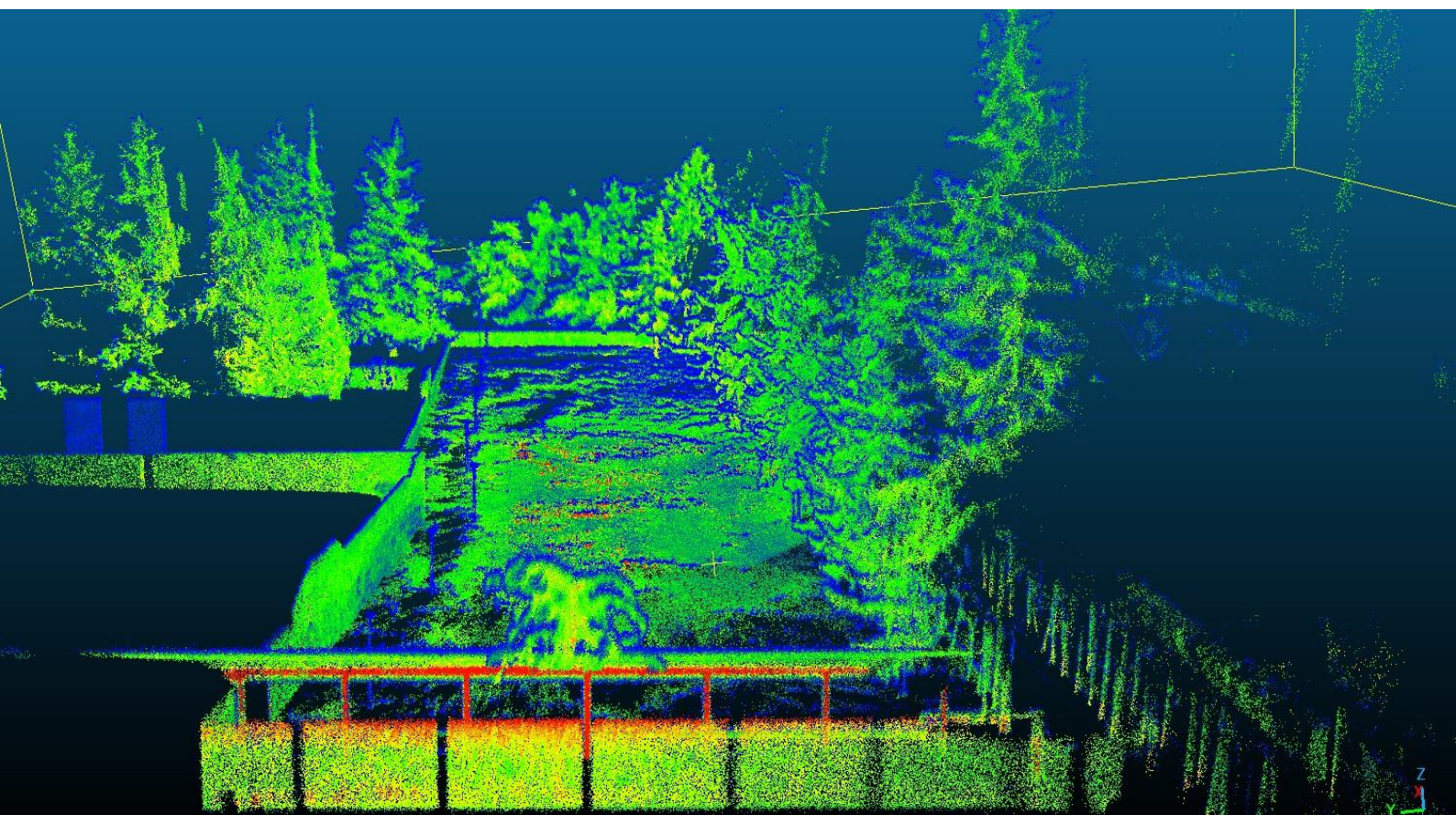
Εικόνα 67: Συγκριτικά αποτελέσματα - καταγραφή 9 (Πάνω μέρος FAST-LIO2, κάτω Faster-LIO)



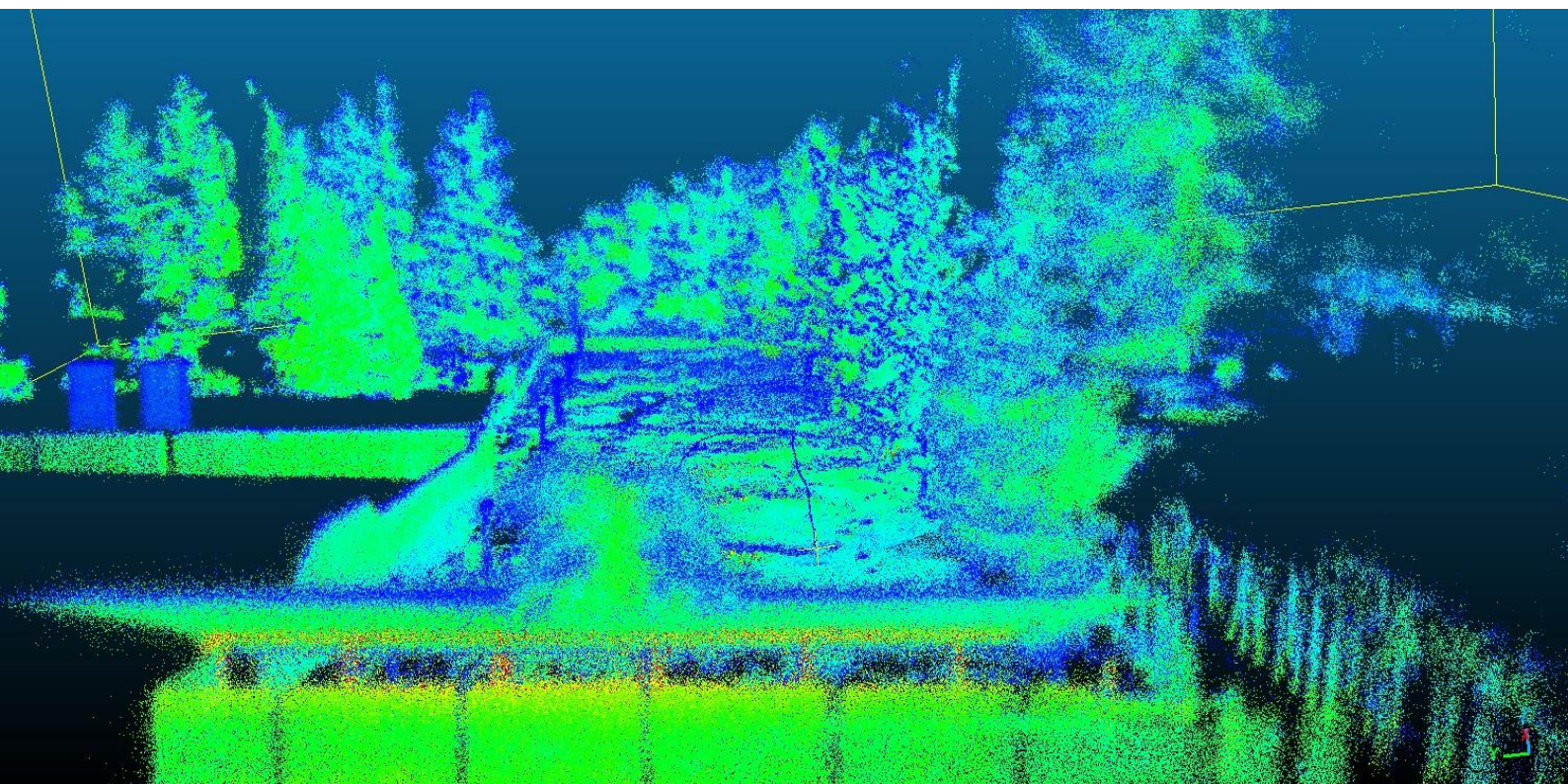
Εικόνα 68: Συγκριτικά αποτελέσματα - καταγραφή 10 (Πάνω μέρος FAST-LIO2, στην μέση Faster-LIO, κάτω LIVOX MAPPING)



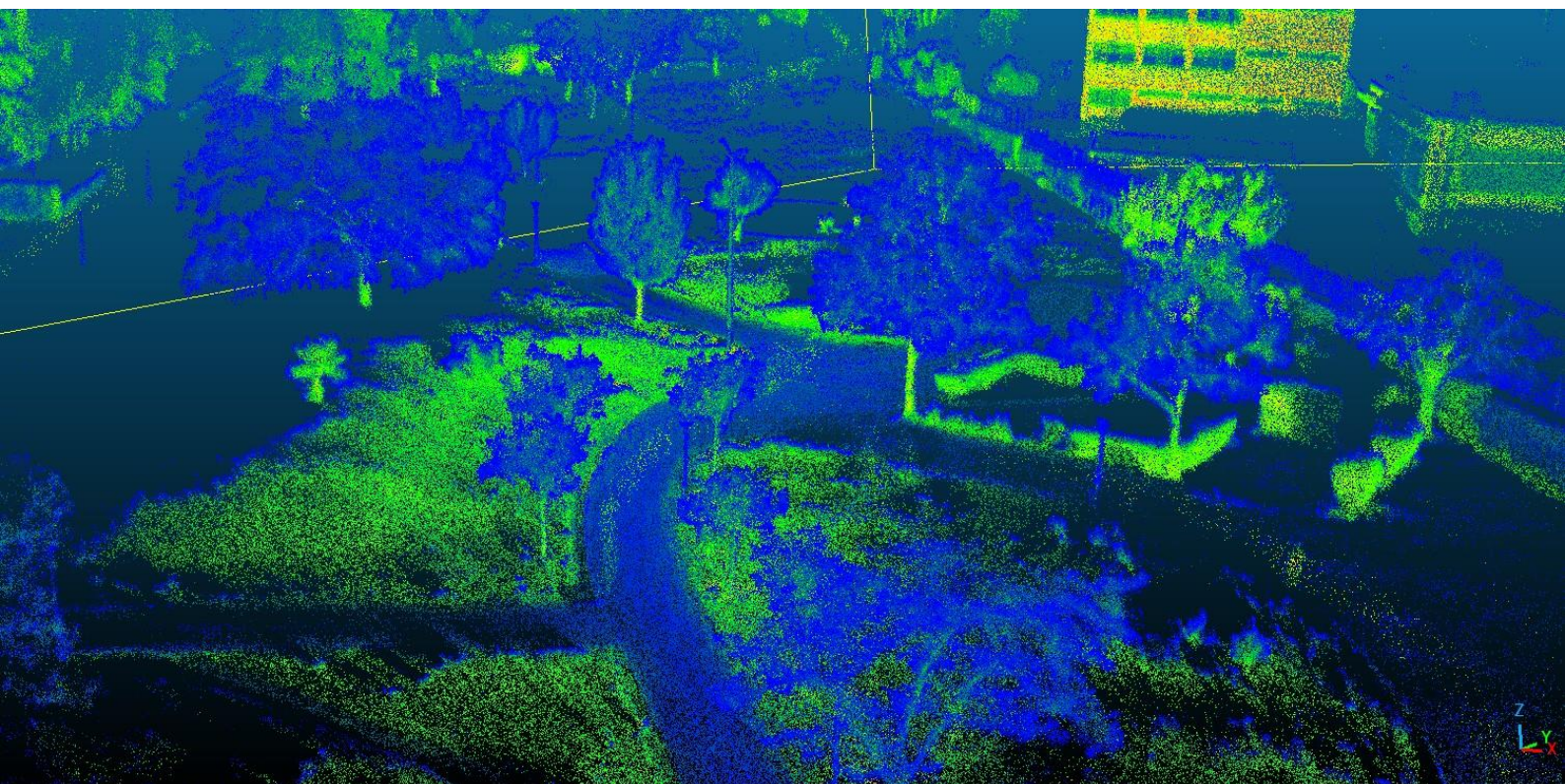
Εικόνα 69: Αποτέλεσμα FAST-LIO2 – καταγραφή 11



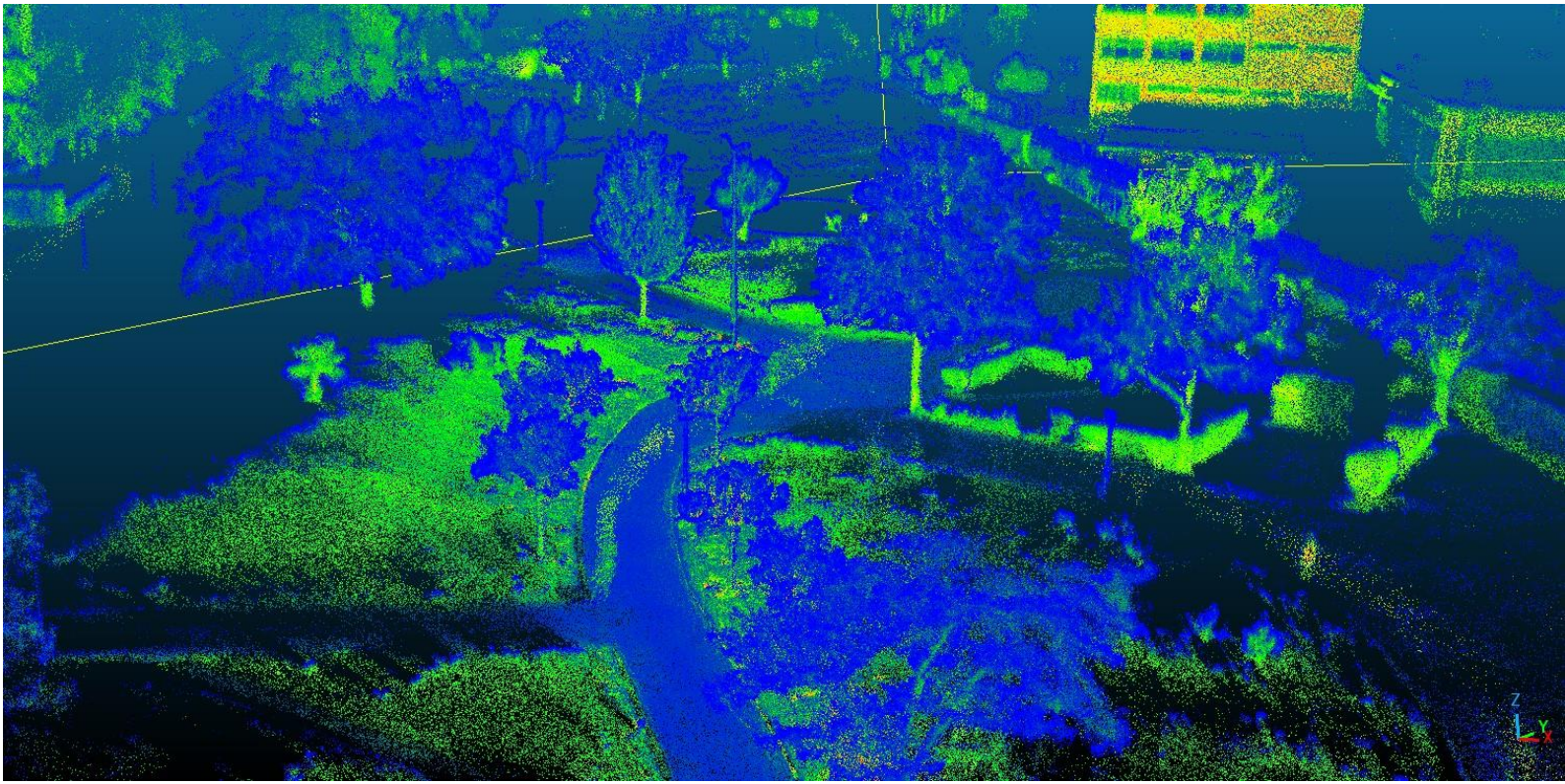
Εικόνα 70: Αποτέλεσμα Faster-LIO – καταγραφή 11



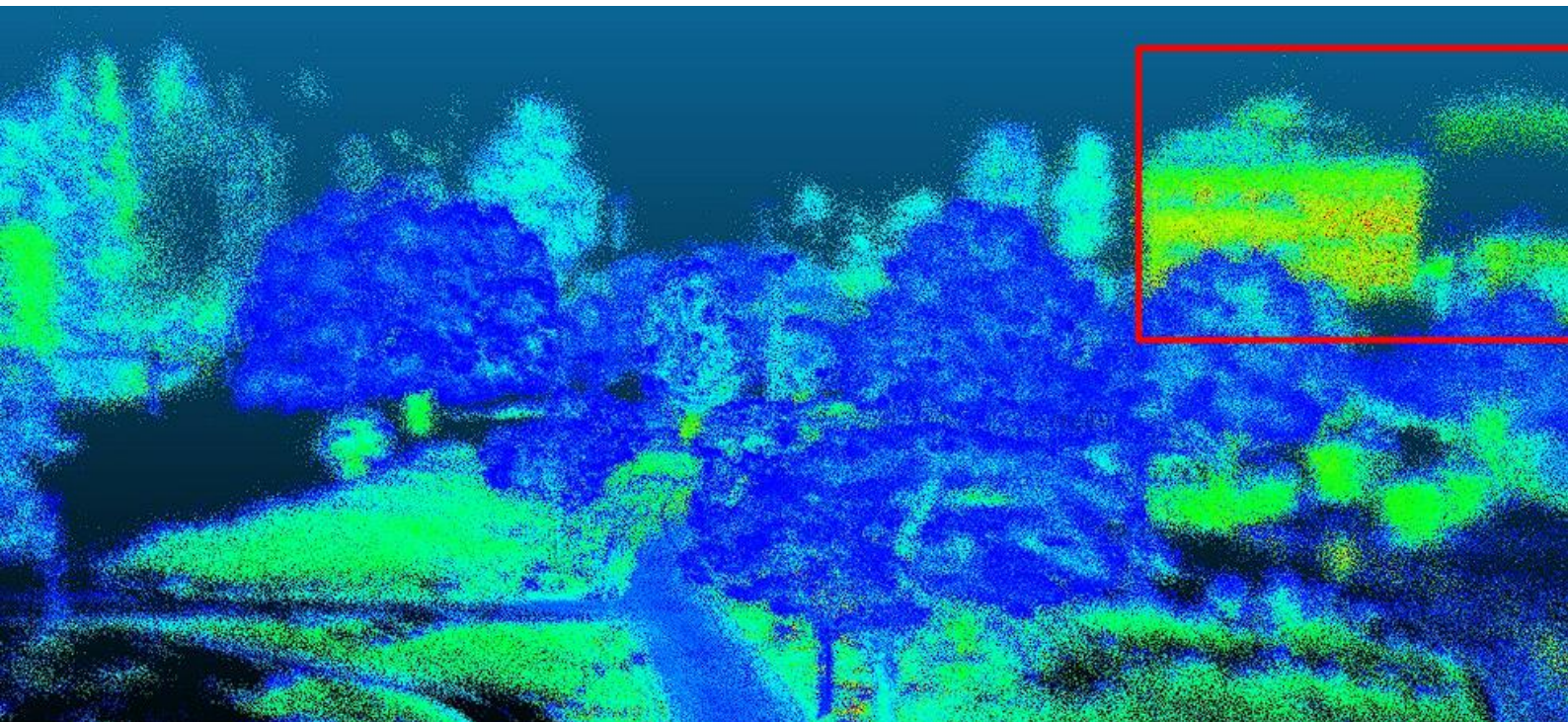
Εικόνα 71: Αποτέλεσμα LIVOX MAPPING – καταγραφή 11



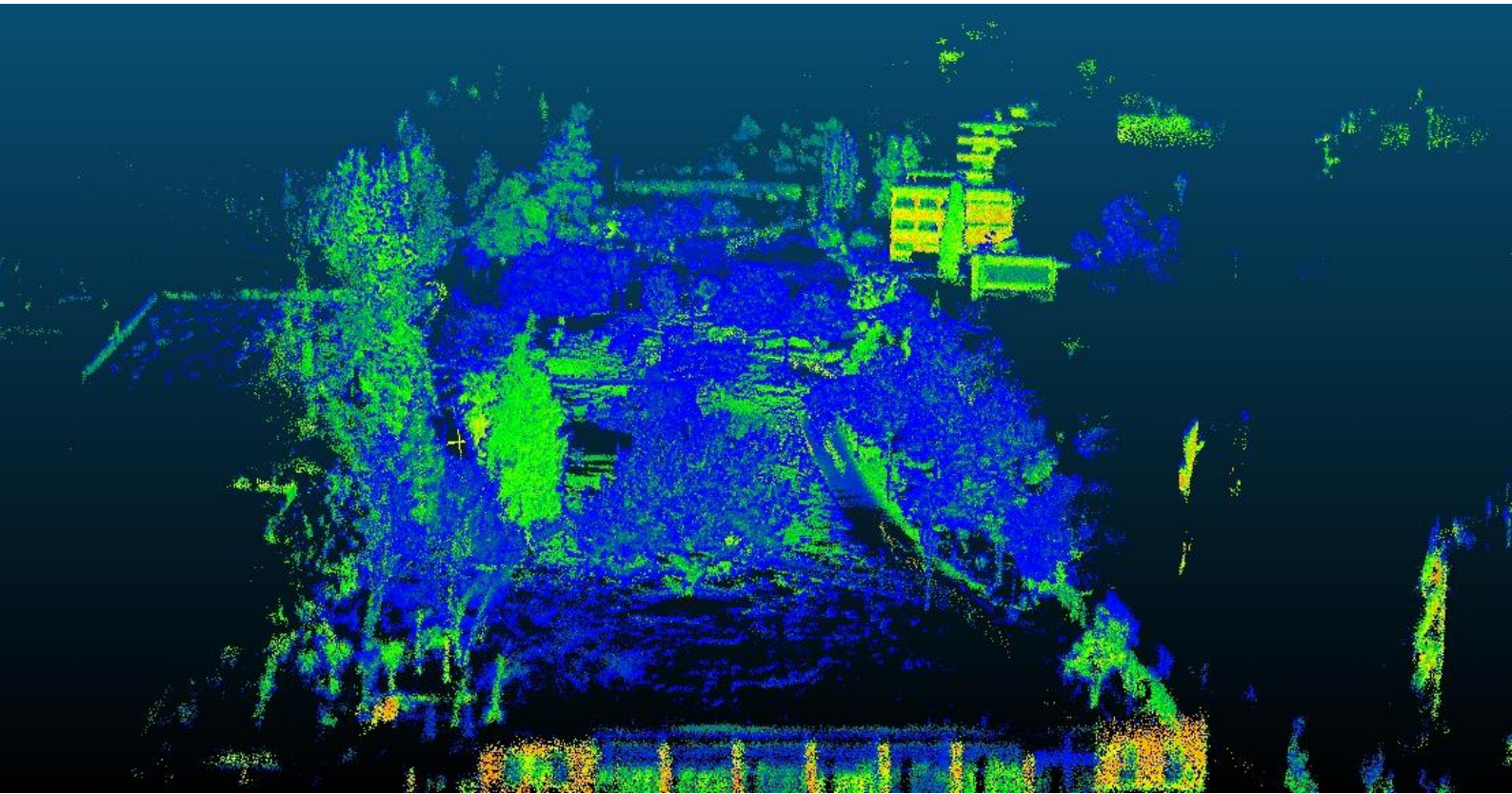
Εικόνα 72: Αποτέλεσμα FAST-LIO2 – καταγραφή 12



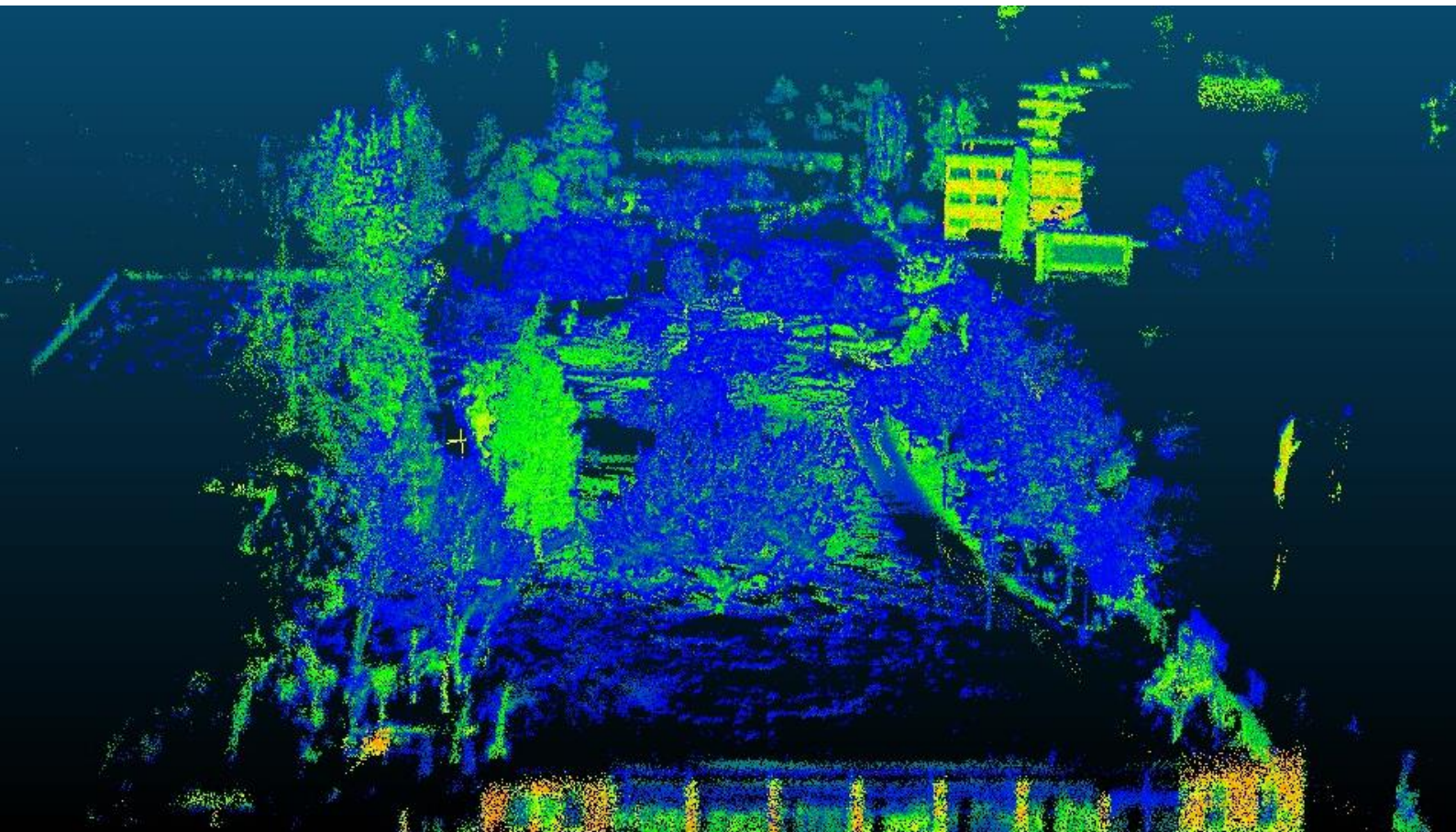
Εικόνα 73: Αποτέλεσμα Faster-LIO – καταγραφή 12



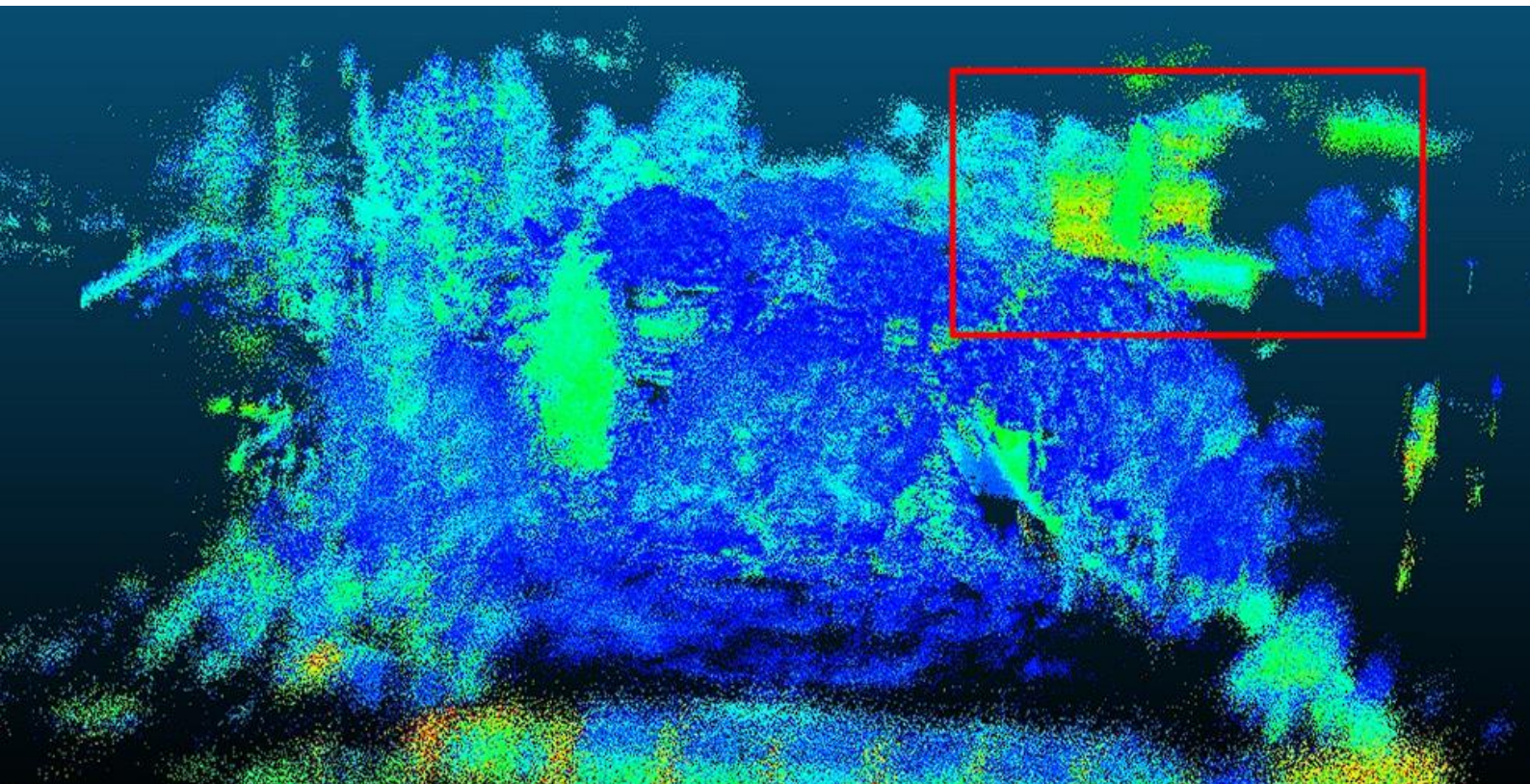
Εικόνα 74: Αποτέλεσμα LIVOX MAPPING – καταγραφή 12



Εικόνα 75: Αποτέλεσμα FAST-LIO2 – καταγραφή 13



Εικόνα 76: Αποτέλεσμα Faster-LIO – καταγραφή 13



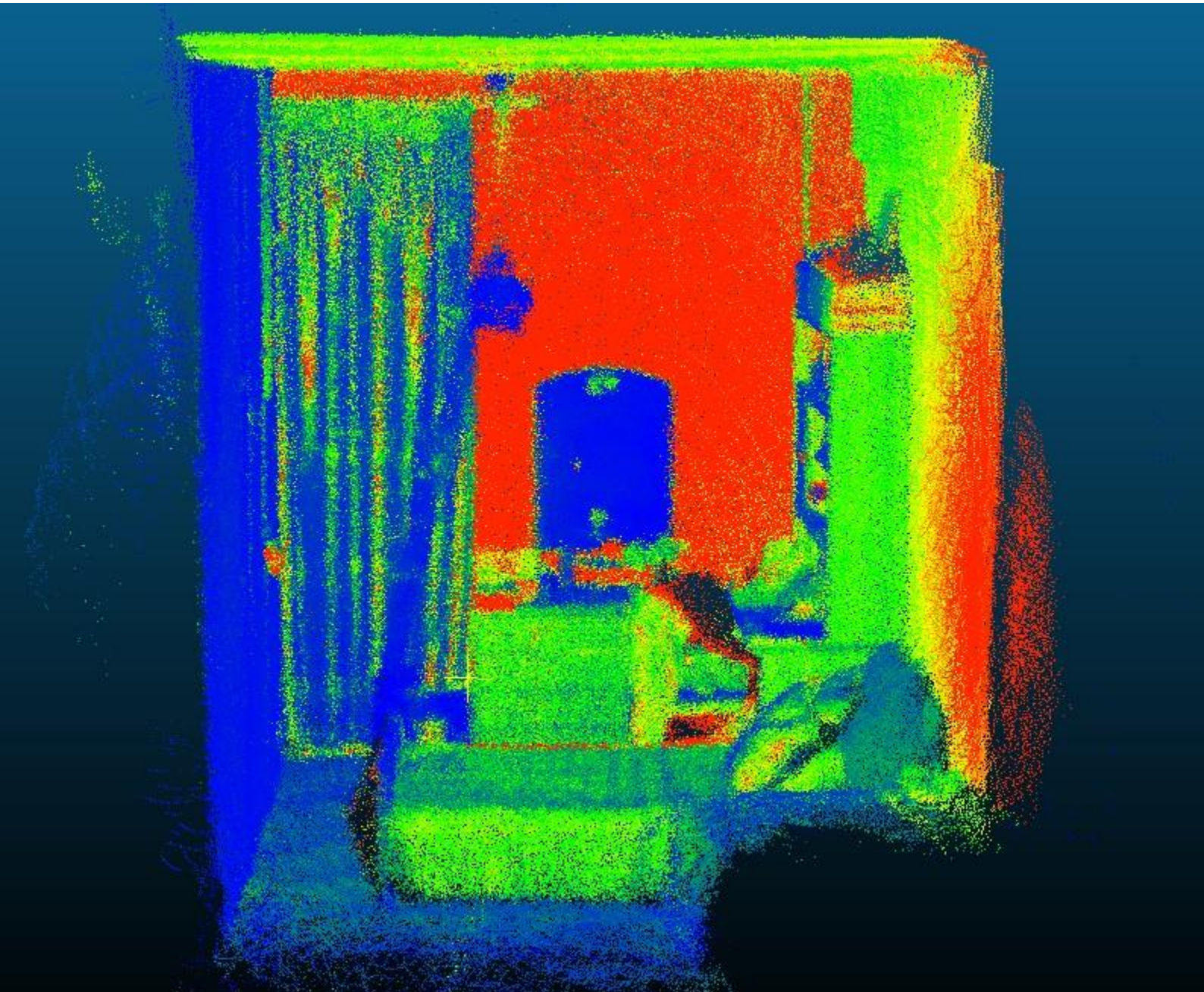
Εικόνα 77: Αποτέλεσμα LIVOX MAPPING – καταγραφή 13

Από τα παραπάνω προκύπτει ότι για εξωτερικούς χώρους, το καλύτερο αποτέλεσμα είχε ο αλγόριθμος Faster-LIO, ενώ μετά ακολούθησε με παρόμοιας ακρίβειας αποτελέσματα ο FAST-LIO2. Εξάιρεση αποτελεί η δοκιμή 9 στην οποία προέκυψε σφάλμα διπλοπροβολής. Ακολουθούν οι αλγόριθμοι LIVOX MAPPING και LOAM LIVOX οι οποίοι εμφάνισαν προβλήματα σε κάποιες πιο απαιτητικές δοκιμές, ενώ και στις πιο απλές περιπτώσεις παρουσίασαν περισσότερο θόρυβο από τα αντίστοιχα αποτελέσματα των πρώτων δύο. Από την μεταξύ τους σύγκριση προέκυψε ότι ο LIVOX MAPPING εμφανίζει λιγότερο θόρυβο, ενώ έχει επίσης και το πλεονέκτημα ότι μπορεί να αναγνωρίσει τον ίδιο τύπο δεδομένων εισόδου με τους δυο πρώτους (rosbag CustomMsg, καθώς ο LOAM LIVOX και ο KISS ICP χρησιμοποιούν rosbag Pointcloud2). Τέλος, ο KISS-ICP δεν απέτυχε σε καμία δοκιμή, όμως τα αραιά νέφη σημείων που παρήγαγε δεν μπορούν να καταγράψουν ιδιαίτερες λεπτομέρειες, ενώ σε μεγαλύτερη κλίμακα εμφανίζει κάποιες μικρές αστοχίες από στροφές και αυξημένο θόρυβο.

Διαφορετική εικόνα παρουσιάζεται στις εσωτερικές δοκιμές, οπότε, όπως αναφέρθηκε και προηγουμένως, ο αλγόριθμος FAST-LIO2 απέτυχε σε κάθε περίπτωση ενώ επίσης, ο LIVOX MAPPING αντιμετώπισε αρκετά σημαντικά προβλήματα θορύβου. Ο KISS-ICP λειτούργησε ύστερα από κατάλληλη ρύθμιση των παραμέτρων αλλά στο τελικό αποτέλεσμα δεν αποδίδονται ικανοποιητικά οι λεπτομέρειες του χώρου. Το βέλτιστο αποτέλεσμα λοιπόν επετεύχθη από τον Faster-LIO ο οποίος δεν απέτυχε σε καμία δοκιμή, και με μόνο πρόβλημα την ύπαρξη θορύβου. Αντίστοιχα ο αλγόριθμος LOAM LIVOX λειτούργησε σε όλες τις

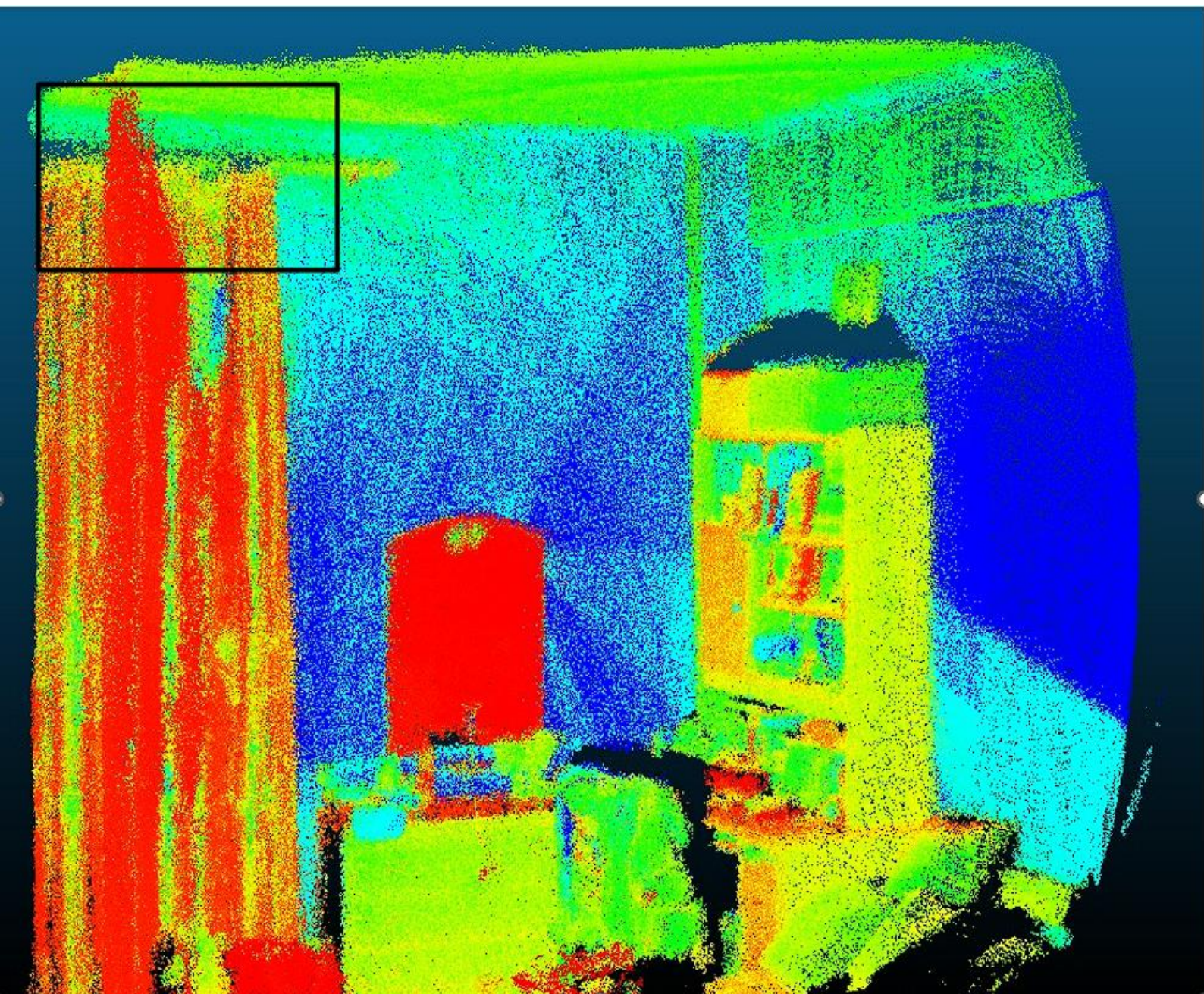
περιπτώσεις, ο οποίος όμως παρουσίασε μεγαλύτερα προβλήματα θορύβου από τον Faster-LIO.

Παρακάτω παρουσιάζονται (Εικόνες 78-80) τα συγκριτικά αποτελέσματα των αλγορίθμων στις δοκιμές εσωτερικού χώρου, με εξαίρεση των KISS-ICP και FAST-LIO2, οι οποίοι απέτυχαν να χαρτογραφήσουν ικανοποιητικά την περιοχή ενδιαφέροντος.



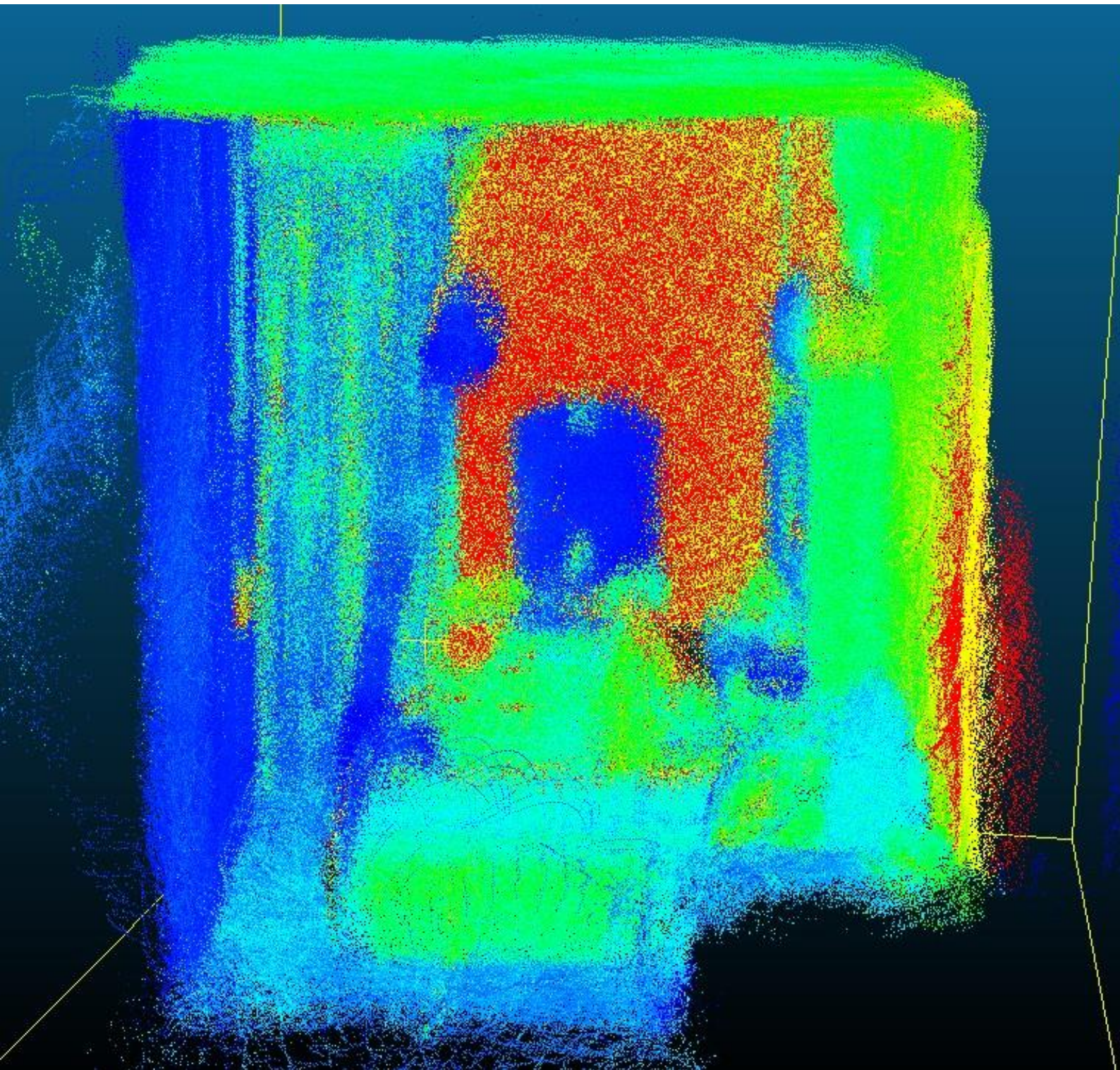
Εικόνα 78: Αποτέλεσμα Faster-LIO (CloudCompare) – καταγραφή εσωτερικού χώρου

Το αποτέλεσμα του Faster-LIO (Εικόνα 78) παρουσιάζει αυξημένο θόρυβο, όμως δεν παρατηρούνται γεωμετρικές παραμορφώσεις ή διπλοπροβολές και συνεπώς, σε γενικές γραμμές, ο εσωτερικός χώρος έχει αποτυπωθεί ικανοποιητικά.



Εικόνα 79: Αποτέλεσμα Faster-LIO (CloudCompare) – καταγραφή εσωτερικού χώρου

Το αποτέλεσμα του LOAM LIVOX (Εικόνα 79) επίσης παρουσιάζει σημαντικό θόρυβο, ενώ ειδικά στην περιοχή της κουρτίνας εμφανίζεται εντονότερος θόρυβος λόγω εσφαλμένης συνένωσης νέφους με αποτέλεσμα η περιοχή να ενώνεται με το ταβάνι. Τα υπόλοιπα στοιχεία εμφανίζονται αρκετά ικανοποιητικά λόγω και της πυκνότερης ανακατασκευής του χάρτη σε σύγκριση με τον Faster-LIO.



Εικόνα 80: Αποτέλεσμα LIVOX MAPPING (CloudCompare) – καταγραφή εσωτερικού χώρου

Ο αλγόριθμος LivoX Mapping (Εικόνα 80) εμφάνισε μεγάλα προβλήματα θορύβου, με αποτέλεσμα τα περισσότερα αντικείμενα του χώρου να μην γίνονται καθόλου αντιληπτά.

Συνολικά, ολοκληρώνοντας την σύγκριση, προκύπτει πως ο βέλτιστος αλγόριθμος ήταν ο Faster-LIO. Όσον αφορά την χρηστικότητα, μπορεί να μην διακρίνεται από αυτήν του KISS ICP (χρήση σε διαφορετικά λειτουργικά συστήματα και γλώσσες προγραμματισμού) ωστόσο μπορεί κανείς να τρέξει τον αλγόριθμο με βασικές γνώσεις Linux Ubuntu, C++ και ROS χωρίς να απαιτούνται αντίστοιχες ρυθμίσεις στον κώδικα. Στους υπολοίπους τομείς πέραν της χρηστικότητας, ο αλγόριθμος Faster-LIO είχε τον μικρότερο χρόνο εκτέλεσης ενώ ταυτόχρονα έδωσε ποιοτικότερα αποτελέσματα στις δοκιμές εσωτερικού και εξωτερικού χώρου.

Ο δεύτερος καλύτερος αλγόριθμος συνολικά ήταν ο FAST-LIO2 καθώς αποτελεί εξίσου χρηστικό αλγόριθμο με τον Faster-LIO ο οποίος όμως ήταν ελαφρώς πιο αργός. Επίσης, δεν κατάφερε να λειτουργήσει σε στενό εσωτερικό χώρο, ενώ επίσης εμφάνισε σφάλμα σε μια εκ των δοκιμών σε εξωτερικό χώρο.

Στη σειρά κατάταξης, ακολουθεί ο αλγόριθμος KISS-ICP λόγω της μεγάλης του χρηστικότητας αλλά και της δυνατότητας λειτουργίας σε πραγματικό χρόνο. Ταυτόχρονα κατάφερε στις εξωτερικές σαρώσεις να δώσει σχετικά αραιά νέφη σημείων χωρίς όμως μεγάλα σφάλματα. Βέβαια αξίζει να σημειωθεί πως ο αλγόριθμος KISS-ICP δεν αφορά επί της ουσίας μέθοδο LiDAR-SLAM αλλά πρακτικά μέθοδο Οδομετρίας καθώς δεν να κάνει χρήση τεχνικών βελτιστοποίησης ή εύρεσης κλειστών βρόχων. Γι' αυτό το λόγο αναμένεται σε μεγαλύτερες καταγραφές να εμφανίζει χειρότερα αποτελέσματα.

Τέλος, οι μέθοδοι LOAM LIVOX και LIVOX MAPPING αποτελούν αρκετά δύσχρηστους αλγόριθμους που απαιτούν πολύ καλή γνώση γλώσσας C++ και ROS για να εκτελεστούν, όπως επίσης και καλές γνώσεις τεχνικών SLAM για να παραμετροποιηθούν σε περίπτωση που δεν χρησιμοποιηθεί κάποιος από τους τυπικούς σαρωτές. Εκτός από την έλλειψη υποστήριξης από τους δημιουργούς, την ανάγκη αλλαγής του κώδικα και εύρεσης των κατάλληλων εκδόσεων βιβλιοθηκών για την σωστή λειτουργία τους, οι αλγόριθμοι αυτοί δεν κατάφεραν να εκτελεστούν σε πραγματικό χρόνο, γεγονός το οποίο αποτελεί σημαντική προϋπόθεση για τις εφαρμογές SLAM. Τέλος, ακόμα και μετά την προσπάθεια εύρεσης παραμέτρων κατάλληλων για τον σαρωτή LIVOX AVIA, τα αποτελέσματα τους σε εσωτερικό και εξωτερικό χώρο ήταν κατώτερα από αυτά του Faster-LIO σε κάθε περίπτωση. Ωστόσο, επισημαίνεται ότι τα αποτελέσματα των δυο τελευταίων μεθόδων ενδέχεται να επηρεάζονται άμεσα από την χρήση διαφορετικού σαρωτή από αυτούς για τους οποίους έχουν σχεδιαστεί οι μέθοδοι, καθώς κατά την εφαρμογή τους στα δοκιμαστικά δεδομένα του LOAM LIVOX πέτυχαν και οι δυο εξαιρετικό αποτέλεσμα, σε αντίθεση με τα αποτελέσματα που προέκυψαν από τα δεδομένα που συλλέχθηκαν στο πλαίσιο της εργασίας.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

6.1 Συμπεράσματα

Αναμφίβολα στα ερχόμενα χρόνια, οι τεχνικές SLAM πρόκειται να πρωταγωνιστήσουν στην επιστημονική κοινότητα λόγω της ολοένα αυξανόμενης ανάγκης για αυτοματοποιημένα οχήματα και ρομποτικά συστήματα. Βασικό τομέα των τεχνικών SLAM αποτελούν οι μέθοδοι που αξιοποιούν δεδομένα αισθητήρων LiDAR καθώς έχουν σημαντικά πλεονεκτήματα όπως η δυνατότητα λειτουργίας σε απαιτητικά περιβάλλοντα χωρίς φωτισμό ή έντονη υφή. Ωστόσο, μέχρι πρόσφατα, κάτι τέτοιο ήταν αρκετά δύσκολο να υλοποιηθεί καθώς οι υπολογιστικοί πόροι ήταν περιορισμένοι και οι σαρωτές που απαιτούνταν αρκετά ακριβοί. Σταδιακά, μέσω της εξέλιξης των υπολογιστικών συστημάτων και της τεχνολογίας των σαρωτών, η χρήση των αλγορίθμων LiDAR-SLAM είναι πλέον αρκετά διαδεδομένη. Σήμερα υπάρχουν σχετικά φθηνοί φορητοί αισθητήρες που έχουν τις προδιαγραφές να πέτυχουν υψηλές ακρίβειες (μερικά εκατοστά), ενώ ταυτόχρονα κάθε χρόνο παρουσιάζεται ένας ικανοποιητικός αριθμός νέων τεχνικών και μεθόδων. Στο πλαίσιο της παρούσας διπλωματική εργασίας εντοπίστηκαν πάνω από είκοσι νέοι αλγόριθμοι οι οποίοι έχουν κάνει την εμφάνισή τους την τελευταία πενταετία, ωστόσο υπάρχουν ακόμα μεγάλα περιθώρια βελτίωσης.

Από την έρευνα και τα πειράματα που πραγματοποιήθηκαν, προκύπτει πως οι αλγόριθμοι SLAM παρουσιάζουν ακόμα μεγάλα προβλήματα κατά την χρήση τους και είναι σχετικά δύσκολο να εφαρμοστούν άμεσα από την επιστημονική κοινότητα. Οι αλγόριθμοι αυτοί είναι σχεδόν αποκλειστικά γραμμένοι σε γλώσσα προγραμματισμού C++ και απαιτούν ως λειτουργικό σύστημα το Linux Ubuntu μαζί με χρήση της βιβλιοθήκης ROS. Παράλληλα εντοπίστηκαν μόλις δυο καταξιωμένοι αλγόριθμοι υλοποιημένοι σε Python και μόνο κάποια λίγα παραδείγματα υλοποιημένα σε MATLAB, τα οποία, αφού μελετήθηκαν, δεν κρίθηκαν ικανά να υποστηρίξουν στην πράξη τέτοιου είδους εφαρμογές. Επίσης εντοπίστηκαν μόνο δυο αλγόριθμοι που μπορούν να λειτουργήσουν σε Windows ή άλλα λειτουργικά συστήματα (πέρα του Linux) με αποτέλεσμα να απαιτούνται πολλές φορές γνώσεις εξειδικευμένων λειτουργικών συστημάτων. Επιπλέον, πολλές μέθοδοι δεν αναφέρουν βασικά χαρακτηριστικά για την πρακτική εφαρμογή τους από τρίτους, όπως είναι ποιες εκδόσεις βιβλιοθηκών χρησιμοποιούν ή με ποιες παραμέτρους πραγματοποίησαν τις δοκιμές που περιγράφουν στις δημοσιεύσεις τους. Ακόμη, οι περισσότεροι από αυτούς τους αλγορίθμους έχουν αναπτυχθεί σε παλιότερες εκδόσεις λειτουργικών ή αφορούν παλιότερες εκδόσεις βιβλιοθηκών ενώ παράλληλα δεν παρέχεται υποστήριξη ή ενημερώσεις από τους δημιουργούς, Αυτό έχει ως αποτέλεσμα να μην λειτουργούν απρόσκοπτα, ύστερα από μια απλή εγκατάσταση. Τέλος, συχνά δεν δίνονται οδηγίες για την εγκατάσταση και εφαρμογή των μεθόδων, αλλά ούτε αναλυτική περιγραφή όλων των παραμέτρων που χρησιμοποιούν, με συνέπεια ο κώδικας να μην μπορεί εύκολα να παραμετροποιηθεί ανάλογα με την περίπτωση.

Όσον αφορά το θεωρητικό μέρος, και βάσει της αναζήτησης που έγινε, προέκυψε ότι υπάρχουν κυρίως δυο είδη σύγχρονων αλγορίθμων LiDAR-SLAM. Οι αλγόριθμοι που βασίζονται σε βελτιστοποίηση μέσω φίλτρων (μέθοδοι LIO) και συνδυάζουν άμεσα μετρήσεις ακόμα και από συστήματα IMU/INS και αυτοί που βασίζονται στην βελτιστοποίηση γράφου και οι οποίοι συνήθως αφορούν μεθόδους που απαιτούν την εξαγωγή γεωμετρικών χαρακτηριστικών και έχουν συνήθως ως βάση τον αλγόριθμο LOAM. Ακόμη προέκυψε πως γενικότερα οι αλγόριθμοι LiDAR-SLAM έχουν πολύ μεγάλη εξάρτηση από τον σαρωτή και το περιβάλλον σάρωσης, καθώς χρειάζονται διαφορετική διαχείριση ειδικά όσον αναφορά την διαδικασία της δειγματοληψίας, της διόρθωσης της θόλωσης από κίνηση (motion blur) και της εξαγωγής των χαρακτηριστικών.

Τα παραπάνω επιβεβαιώθηκαν και από τις σχετικές δοκιμές μέσω του σαρωτή LIVOX AVIA, όπου από τους πέντε αλγορίθμους οι οποίοι εξετάστηκαν, οι αλγόριθμοι filter-based είχαν καλύτερα αποτελέσματα από τους αντίστοιχους graph-based. Οι πρώτοι μάλιστα (Faster-LIO και FAST-LIO2) κατάφεραν να δώσουν ένα πολύ ικανοποιητικό αποτέλεσμα καθώς οπτικά δεν παρατηρήθηκαν χονδροειδή σφάλματα σε τροχιές και χάρτη, ενώ επίσης κατάφεραν να τρέξουν σε πραγματικό χρόνο και μάλιστα πολύ γρηγορότερα από τον συνήθη ρυθμό καταγραφής νεφών σημείων και μάλιστα χωρίς την απαίτηση ιδιαίτερα ισχυρού υπολογιστικού συστήματος. Τέλος, ακόμη και ο αλγόριθμος KISS ICP, που βέβαια δεν αποτελεί κατεξοχήν αλγόριθμο SLAM, έδωσε αρκετά ικανοποιητικά αποτελέσματα και πέτυχε την δυνατότητα λειτουργίας σε πραγματικό χρόνο χωρίς να είναι υλοποιημένος σε C++. Αντίθετα, οι τελευταίοι αλγόριθμοι που εξετάστηκαν (LOAM LIVOX και LIVOX MAPPING) αποτέλεσαν βασικά παραδείγματα που φανερώνουν το πρόβλημα χρηστικότητας και λειτουργικότητας των τεχνικών LiDAR-SLAM. Έτσι, παρά την επιστημονική προσέγγιση που εφαρμόζουν, δεν παρουσίασαν καλά αποτελέσματα λόγω κυρίως του ότι είναι μη ενημερωμένοι και δεν παρέχουν ικανοποιητικές πληροφορίες για τις βιβλιοθήκες ή τις επιμέρους ρυθμίσεις των παραμέτρων σε άλλους σαρωτές. Συγκεκριμένα, η εκτέλεση των συγκεκριμένων αλγορίθμων ήταν αρκετά δύσκολη αφού χρειάστηκε πληθώρα αλλαγών στον κώδικα, ενώ όταν αυτό επιτεύχθηκε, τα αποτελέσματα παρουσίαζαν σημαντικά σφάλματα κυρίως λόγω της χρήσης διαφορετικού σαρωτή. Τελικά, ακόμα και ύστερα από τις κατάλληλες αλλαγές των παραμέτρων, η ακρίβεια δεν ήταν αντίστοιχη των δοκιμών που πραγματοποιήθηκαν στα δεδομένα που συνοδεύουν τους αλγορίθμους. Επίσης, χρονικά καθυστέρησαν και στις δυο περιπτώσεις και δεν κατάφεραν να εκτελεστούν σε πραγματικό χρόνο.

Συνοπτικά λοιπόν από την έρευνα και τα πειράματα που πραγματοποιήθηκαν προκύπτει πως οι σύγχρονοι αλγόριθμοι LiDAR-SLAM ήδη έχουν φτάσει σε ένα ικανοποιητικό σημείο και μπορούν να υποστηρίξουν εφαρμογές πραγματικού χρόνου σε δύσκολες συνθήκες όπου υπάρχει έλλειψη φωτισμού, γρήγορες μετακινήσεις και στροφές, κραδασμοί και κινούμενες οντότητες. Προσφέρουν καλή ακρίβεια χαρτογράφησης και πλοήγησης σε πολύ σύντομο χρόνο, με φθινό σχετικά εξοπλισμό (σαρωτής LIVOX AVIA κόστους περίπου 1600 ευρώ) και στοιχειώδη

υπολογιστικά συστήματα (κοινός υπολογιστής κόστους περίπου 800 ευρώ). Χαρακτηριστικό παράδειγμα αποτελεί η παραγωγή ικανοποιητικών αποτελεσμάτων χαρτογράφησης χώρου έκτασης μερικών στρεμμάτων σε λιγότερο από πέντε λεπτά με τον χάρτη και την τοποθεσία του σαρωτή να ενημερώνονται σε πραγματικό χρόνο. Τέλος, προκύπτει πως πλέον οι αλγόριθμοι LiDAR-SLAM μπορούν να λειτουργήσουν παράγοντας ικανοποιητικά αποτελέσματα και σε περιπτώσεις που δεν εντοπίζονται κλειστοί βρόχοι κατά την πορεία του σαρωτή. Προκύπτει ακόμη πως υπάρχουν σημαντικά ζητήματα διαλειτουργικότητας και χρηστικότητας των αλγορίθμων αυτών, καθώς έχουν αναπτυχθεί κυρίως σε γλώσσα προγραμματισμού C++ και σε συνδυασμό με το λογισμικό ROS, παρόλο που, από τα πειράματα που πραγματοποιήθηκαν, παρατηρήθηκε ότι ακόμα και η Python είναι σε θέση να υποστηρίξει τεχνικές LiDAR-SLAM σε πραγματικό χρόνο. Ακόμη προκύπτει πως και το MATLAB, αν και ακόμη δεν είναι σε θέση να υποστηρίξει πλήρως τέτοιες εφαρμογές, έχει αρχίσει τα τελευταία χρόνια να κάνει σταδιακές βελτιώσεις και να παρέχει σχετικά εργαλεία που μπορούν να χρησιμεύσουν στον έλεγχο ή την εκμάθηση των τεχνικών αυτών. ένα ακόμα συμπέρασμα που προκύπτει είναι ότι οι αλγόριθμοι LIDAR SLAM, και ιδιαίτερα αυτοί που βασίζονται στην εξαγωγή χαρακτηριστικών, επηρεάζονται σημαντικά από τις παραμέτρους του σαρωτή και του περιβάλλοντος χώρου, με τους αλγορίθμους οι οποίοι συνδυάζουν και μετρήσεις IMU/INS ωστόσο να αντιμετωπίζουν πολύ καλύτερα τα προβλήματα αυτά. Όσον αφορά τον σαρωτή των πειραμάτων, προκύπτει ότι αποτελεί έναν ελαφρύ φορητό αισθητήρα ο οποίος μπορεί να υποστηρίξει πλήρως εφαρμογές LiDAR-SLAM σε εξωτερικούς ή μεγάλους εσωτερικούς χώρους παρέχοντας παράλληλα και ενσωματωμένο σύστημα IMU/INS. Ωστόσο, εξειδικεύεται κυρίως στους εξωτερικούς χώρους καθώς δεν μπορεί να καταγράψει δεδομένα σε απόσταση μικρότερη του 1 μέτρου, ενώ σε απόσταση μεταξύ 1-5 μέτρων εμφανίζει σημαντικό θόρυβο. Γενικώς συμπεραίνεται ότι οι τεχνικές SLAM έχουν ήδη φτάσει σε ένα πολύ ικανοποιητικό επίπεδο, ενώ από την έρευνα της βιβλιογραφίας προκύπτει ότι συνεχίζουν να εξελίσσονται συνεχώς.

6.2 Μελλοντική Έρευνα

Αρχικά κρίνεται σημαντικό να υλοποιηθεί μελλοντικά ένα ενιαίο πλαίσιο ανάπτυξης των τεχνικών LIDAR SLAM, το οποίο θα ορίζει συγκεκριμένους κανόνες, όπως προτεινόμενες ονομασίες παραμέτρων, την υποχρεωτική αναφορά των εκδόσεων βιβλιοθηκών και των τιμών των παραμέτρων οι οποίες χρησιμοποιήθηκαν στις δημοσιευμένες δοκιμές αλλά και θα παρέχει βασικές οδηγίες για την εκτέλεση τους. Επιπλέον ένα αντίστοιχο τμήμα έρευνας θα ήταν η υλοποίηση κάποιων εκ των αλγορίθμων αυτών σε διαφορετικές γλώσσες προγραμματισμού και λειτουργικά συστήματα, ώστε να μπορεί να γίνει ευκολότερη η χρήση τους από ένα ευρύτερο κοινό λύνοντας παράλληλα με αυτόν τον τρόπο πολλά προβλήματα που προκύπτουν από την έλλειψη χρηστικότητας και λειτουργικότητας που παρουσιάζονται επί του παρόντος.

Ένα επιπλέον πεδίο που θα μπορούσε να διερευνηθεί μελλοντικά είναι η δημιουργία ενός κατάλληλου συνόλου δεδομένων (dataset), το οποίο θα αποτελείται από πολλές διαφορετικές σκηνές εξωτερικών και εσωτερικών χώρων και θα συνοδεύεται από αληθή δεδομένα. Με αυτόν τον τρόπο θα μπορούσε να γίνει καλύτερη αξιολόγηση των τεχνικών LiDAR-SLAM ακόμα και για διαφορετικού τύπου σαρωτών.

Σε αλγοριθμικό επίπεδο, ιδιαίτερο ενδιαφέρον παρουσιάζουν οι μέθοδοι που χρησιμοποιούν νευρωνικά δίκτυα και τεχνικές μηχανικής μάθησης και οι οποίες έχουν κάνει έντονη την εμφάνιση τους στον χώρο του LiDAR-SLAM όπως οι αλγόριθμοι PIN-SLAM [67], NERF-LOAM [68] και LONER [69]. Οι μέθοδοι αυτές ενδέχεται να αναβαθμίσουν αρκετά τις ικανότητες και τα αποτελέσματα των αλγορίθμων βελτιώνοντας πολλούς τομείς όπως ο εντοπισμός χαρακτηριστικών, η επιλογή κατάλληλων παραμέτρων, η δυναμική χαρτογράφηση, η εύρεση κλειστών βρόχων και η αφαίρεση θορύβου. Τέλος, θα μπορούσε να υλοποιηθεί μια νέα μέθοδος SLAM η οποία θα προσπαθεί να συνδυάσει όλα τα αποτελέσματα της παρούσας και μελλοντικής έρευνας, αναπτύσσοντας έτσι έναν χρηστικό αλγόριθμο ο οποίος θα χρησιμοποιεί τεχνικές μηχανικής μάθησης και συνδυασμό μετρήσεων LiDAR με IMU/INS ώστε να μπορεί να προσαρμόζει αυτοματοποιημένα της παραμέτρους του ώστε να λειτουργεί βέλτιστα με κάθε σαρωτή σε κάθε είδους περιβάλλον.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Maiman, T. H. (1960). Stimulated optical radiation in ruby.
2. Mautz, R. (2012). Indoor positioning technologies.
3. Li, Y., & Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4), 50-61.
4. Γραμματικόπουλος Λάζαρος (2023), σημειώσεις μαθήματος “ΕΠΕΞΕΡΓΑΣΙΑ & ΟΠΤΙΚΟΠΟΙΗΣΗ 3D ΔΕΔΟΜΕΝΩΝ”, Πανεπιστήμιο Δυτικής Αττικής.
5. Di Stefano, F., Chiappini, S., Gorreja, A., Balestra, M., & Pierdicca, R. (2021). Mobile 3D scan LiDAR: A literature review. *Geomatics, Natural Hazards and Risk*, 12(1), 2387-2429.
6. Liu, J., Sun, Q., Fan, Z., & Jia, Y. (2018, September). TOF lidar development in autonomous vehicle. In *2018 IEEE 3rd Optoelectronics Global Conference (OGC)* (pp. 185-190). IEEE.
7. Li, N., Ho, C. P., Xue, J., Lim, L. W., Chen, G., Fu, Y. H., & Lee, L. Y. T. (2022). A progress review on solid-state LiDAR and nanophotonics-based LiDAR sensors. *Laser & Photonics Reviews*, 16(11), 2100511.
8. Raj, T., Hanim Hashim, F., Baseri Huddin, A., Ibrahim, M. F., & Hussain, A. (2020). A survey on LiDAR scanning mechanisms. *Electronics*, 9(5), 741.
9. Bastos, D., Monteiro, P. P., Oliveira, A. S., & Drummond, M. V. (2021, February). An overview of LiDAR requirements and techniques for autonomous driving. In *2021 Telecoms Conference (ConfTELE)* (pp. 1-6). IEEE.
10. Φωτίου Δ., 2023. Ψηφιακή Ανακατασκευή Εσωτερικού Χώρου και Ανίχνευση Ανθρώπων με Τεχνολογία LiDAR. Διπλωματική εργασία, ΗΜΜΥ ΑΠΘ.
11. Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., & Chen, Y. (2018). Registration of laser scanning point clouds: A review. *Sensors*, 18(5), 1641.
12. Nüchter, A., Elseberg, J., Schneider, P., & Paulus, D. (2010). Study of parameterizations for the rigid body transformations of the scan registration problem. *Computer Vision and Image Understanding*, 114(8), 963-980.
13. Laine, S., & Karras, T. (2010, February). Efficient sparse voxel octrees. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (pp. 55-63).

14. Dzik, S., & Ezrielev, J. (1992). Representing surfaces with voxels. *Computers & graphics*, 16(3), 295-301.
15. Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., & Kasi, V. (2013). Reviews on various inertial measurement unit (IMU) sensor applications. *International Journal of Signal Processing Systems*, 1(2), 256-262.
16. Faisal, I. A., Purboyo, T. W., & Ansori, A. S. R. (2019). A review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture. *J. Eng. Appl. Sci*, 15(3), 826-829.
17. Skog, I., Nilsson, J. O., & Händel, P. (2014, January). Pedestrian tracking using an IMU array. In *2014 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)* (pp. 1-4). IEEE.
18. Kaplan, E. D., & Hegarty, C. (Eds.). (2017). *Understanding GPS/GNSS: principles and applications*. Artech house.
19. Capuano, V., Shehaj, E., Blunt, P., Botteron, C., & Farine, P. A. (2017). High accuracy GNSS based navigation in GEO. *Acta Astronautica*, 136, 332-341.
20. Chang, L., Niu, X., Liu, T., Tang, J., & Qian, C. (2019). GNSS/INS/LiDAR-SLAM integrated navigation system based on graph optimization. *Remote Sensing*, 11(9), 1009.
21. Ellin, A., & Dolsak, G. (2008). The design and application of rotary encoders. *Sensor Review*, 28(2), 150-158.
22. Wang, J. X., & Cui, X. (2018, July). Rotary Encoder Based Self-Positioning Method for Mobile Robot. In *2018 5th International Conference on Information Science and Control Engineering (ICISCE)* (pp. 500-504). IEEE.
23. Yue, X., Zhang, Y., Chen, J., Chen, J., Zhou, X., & He, M. (2024). LiDAR-based SLAM for robotic mapping: state of the art and new frontiers. *Industrial Robot: the international journal of robotics research and application*, 51(2), 196-205.
24. Zhang, Y., Wang, L., Jiang, X., Zeng, Y., & Dai, Y. (2022). An efficient LiDAR-based localization method for self-driving cars in dynamic environments. *Robotica*, 40(1), 38-55.
25. Liu, L., Coops, N. C., Aven, N. W., & Pang, Y. (2017). Mapping urban tree species using integrated airborne hyperspectral and LiDAR remote sensing data. *Remote Sensing of Environment*, 200, 170-182.
26. Tao, S., Labrière, N., Calders, K., Fischer, F. J., Rau, E. P., Plaisance, L., & Chave, J. (2021). Mapping tropical forest trees across large areas with lightweight cost-effective terrestrial laser scanning. *Annals of Forest Science*, 78(4), 103.

27. Smith, R. C., & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4), 56-68.
28. Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2), 99-110.
29. Leonard, J. J., & Durrant-Whyte, H. F. (1991, November). Simultaneous map building and localization for an autonomous mobile robot. In *IROS* (Vol. 3, pp. 1442-1447).
30. Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598.
31. Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1), 34-46.
32. Zhang, J., & Singh, S. (2014, July). LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and systems* (Vol. 2, No. 9, pp. 1-9).
33. Debeunne, C., & Vivet, D. (2020). A review of visual-LiDAR fusion based simultaneous localization and mapping. *Sensors*, 20(7), 2068.
34. Koide, K., Miura, J., & Menegatti, E. (2019). A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2), 1729881419841532.
35. Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4, 333-349.
36. Besl, P. J., & McKay, N. D. (1992, April). Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures* (Vol. 1611, pp. 586-606). Spie.
37. Segal, A., Haehnel, D., & Thrun, S. (2009, June). Generalized-icp. In *Robotics: science and systems* (Vol. 2, No. 4, p. 435).
38. Biber, P., & Straßer, W. (2003, October). The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)* (Vol. 3, pp. 2743-2748). IEEE.
39. Diosi, A., & Kleeman, L. (2007). Fast laser scan matching using polar coordinates. *The International Journal of Robotics Research*, 26(10), 1125-1153.

40. Rusu, R. B., Blodow, N., & Beetz, M. (2009, May). Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation* (pp. 3212-3217). IEEE.
41. Kim, G., & Kim, A. (2018, October). Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4802-4809). IEEE.
42. Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., & Vincent, R. (2010, October). Efficient sparse pose adjustment for 2D mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 22-29). IEEE.
43. Ranganathan, A. (2004). The levenberg-marquardt algorithm. *Tutorial on LM algorithm*, 11(1), 101-110.
44. Martín, F., Triebel, R., Moreno, L., & Siegwart, R. (2014). Two different tools for three-dimensional mapping: DE-based scan matching and feature-based loop detection. *Robotica*, 32(1), 19-41.
45. Hess, W., Kohler, D., Rapp, H., & Andor, D. (2016, May). Real-time loop closure in 2D LIDAR SLAM. In *2016 IEEE international conference on robotics and automation (ICRA)* (pp. 1271-1278). IEEE.
46. Magnusson, M., Andreasson, H., Nuchter, A., & Lilienthal, A. J. (2009, May). Appearance-based loop detection from 3D laser data using the normal distributions transform. In *2009 IEEE International Conference on Robotics and Automation* (pp. 23-28). IEEE.
47. Júnior, G. P. C., Rezende, A. M., Miranda, V. R., Fernandes, R., Azpúrua, H., Neto, A. A., ... & Freitas, G. M. (2022). EKF-LOAM: An adaptive fusion of LiDAR SLAM with wheel odometry and inertial data for confined spaces with few geometric features. *IEEE Transactions on Automation Science and Engineering*, 19(3), 1458-1471.
48. Qin, C., Ye, H., Pranata, C. E., Han, J., Zhang, S., & Liu, M. (2020, May). Lins: A lidar-inertial state estimator for robust and efficient navigation. In *2020 IEEE international conference on robotics and automation (ICRA)* (pp. 8899-8906). IEEE.
49. Xu, W., & Zhang, F. (2021). Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2), 3317-3324.
50. Xu, W., Cai, Y., He, D., Lin, J., & Zhang, F. (2022). Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4), 2053-2073.
51. Bai, C., Xiao, T., Chen, Y., Wang, H., Zhang, F., & Gao, X. (2022). Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels. *IEEE Robotics and Automation Letters*, 7(2), 4861-4868.

52. Yuan, C., Xu, W., Liu, X., Hong, X., & Zhang, F. (2022). Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry. *IEEE Robotics and Automation Letters*, 7(3), 8518-8525.
53. He, D., Xu, W., Chen, N., Kong, F., Yuan, C., & Zhang, F. (2023). Point-LIO: Robust High-Bandwidth Light Detection and Ranging Inertial Odometry. *Advanced Intelligent Systems*, 5(7), 2200459.
54. Shi, P., Zhu, Z., Sun, S., Zhao, X., & Tan, M. (2023). Invariant extended Kalman filtering for tightly coupled LiDAR-inertial odometry and mapping. *IEEE/ASME Transactions on Mechatronics*.
55. Khosravian Hemami, A., Trumppf, J., Mahony, R., & Lageman, C. (2015). Observers for invariant systems on Lie groups with biased input measurements and homogeneous outputs.
56. Shan, T., & Englot, B. (2018, October). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4758-4765). IEEE.
57. Behley, J., & Stachniss, C. (2018, June). Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Robotics: Science and Systems* (Vol. 2018, p. 59).
58. Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D. (2020, October). Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5135-5142). IEEE.
59. Liu, Z., & Zhang, F. (2021). Balm: Bundle adjustment for lidar mapping. *IEEE Robotics and Automation Letters*, 6(2), 3184-3191.
60. Wang, H., Wang, C., Chen, C. L., & Xie, L. (2021, September). F-loam: Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4390-4396). IEEE.
61. Guo, H., Zhu, J., & Chen, Y. (2022). E-LOAM: LiDAR odometry and mapping with expanded local structural information. *IEEE Transactions on Intelligent Vehicles*, 8(2), 1911-1921.
62. Wang, Z., Zhang, L., Shen, Y., & Zhou, Y. (2022). D-liom: Tightly-coupled direct lidar-inertial odometry and mapping. *IEEE Transactions on Multimedia*.
63. Frosi, M., & Matteucci, M. (2022). Art-slam: Accurate real-time 6dof lidar slam. *IEEE Robotics and Automation Letters*, 7(2), 2692-2699.
64. Reinke, A., Palieri, M., Morrell, B., Chang, Y., Ebadi, K., Carlone, L., & Agha-Mohammadi, A. A. (2022). Locus 2.0: Robust and computationally efficient lidar

- odometry for real-time 3d mapping. *IEEE Robotics and Automation Letters*, 7(4), 9043-9050.
65. Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., & Stachniss, C. (2019, November). Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4530-4537). IEEE.
 66. Tosi, F., Zhang, Y., Gong, Z., Sandström, E., Mattoccia, S., Oswald, M. R., & Poggi, M. (2024). How NeRFs and 3D Gaussian Splatting are Reshaping SLAM: a Survey. *arXiv preprint arXiv:2402.13255*.
 67. Pan, Y., Zhong, X., Wiesmann, L., Posewsky, T., Behley, J., & Stachniss, C. (2024). PIN-SLAM: LiDAR SLAM Using a Point-Based Implicit Neural Representation for Achieving Global Map Consistency. *arXiv preprint arXiv:2401.09101*.
 68. Deng, J., Wu, Q., Chen, X., Xia, S., Sun, Z., Liu, G., ... & Pei, L. (2023). Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 8218-8227).
 69. Isaacson, S., Kung, P. C., Ramanagopal, M., Vasudevan, R., & Skinner, K. A. (2023). Loner: Lidar only neural representations for real-time slam. *IEEE Robotics and Automation Letters*.
 70. Zhen, W., & Scherer, S. (2020). A unified 3D mapping framework using a 3D or 2D LiDAR. In *Proceedings of the 2018 International Symposium on Experimental Robotics* (pp. 702-711). Springer International Publishing.
 71. Karimi, M., Oelsch, M., Stengel, O., Babaian, E., & Steinbach, E. (2021). LoLa-SLAM: Low-latency LiDAR SLAM using continuous scan slicing. *IEEE Robotics and Automation Letters*, 6(2), 2248-2255.
 72. Chen, K., Zhan, K., Pang, F., Yang, X., & Zhang, D. (2022). R-LIO: rotating lidar inertial odometry and mapping. *Sustainability*, 14(17), 10833.
 73. de Lima, L. C., Ramezani, M., Borges, P., & Brünig, M. (2023). Air-ground collaborative localisation in forests using lidar canopy maps. *IEEE Robotics and Automation Letters*, 8(3), 1818-1825.
 74. Park, C., Moghadam, P., Williams, J. L., Kim, S., Sridharan, S., & Fookes, C. (2021). Elasticity meets continuous-time: Map-centric dense 3D LiDAR SLAM. *IEEE Transactions on Robotics*, 38(2), 978-997.
 75. Wang, S., Zhang, H., & Wang, G. (2022). OMC-SLIO: online multiple calibrations spinning LiDAR inertial odometry. *Sensors*, 23(1), 248.

76. Yan, L., Dai, J., Zhao, Y., & Chen, C. (2023). Real-time 3D mapping in complex environments using a spinning actuated LiDAR system. *Remote Sensing*, 15(4), 963.
77. Τζωρτζάκου Α., 2022. Φωτογραμμετρικές Τεχνικές SfM/SLAM με Έμφαση στις Εφαρμογές στη Σπηλαιολογία. Μεταπτυχιακή εργασία, Τμήμα Μηχανικών Τοπογραφίας και Γεωπληροφορικής Πανεπιστημίου Δυτικής Αττικής.
78. Chen, N., Kong, F., Xu, W., Cai, Y., Li, H., He, D., ... & Zhang, F. (2023). A self-rotating, single-actuated UAV with extended sensor field of view for autonomous navigation. *Science Robotics*, 8(76), eade4538.
79. Yang, J., Li, Y., Cao, L., Jiang, Y., Sun, L., & Xie, Q. (2019). A survey of SLAM research based on LiDAR sensors. *International Journal of Sensors*, 1(1), 1003.
80. Xu, X., Zhang, L., Yang, J., Cao, C., Wang, W., Ran, Y., ... & Luo, M. (2022). A review of multi-sensor fusion slam systems based on 3D LIDAR. *Remote Sensing*, 14(12), 2835.
81. Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3354-3361). IEEE.
82. Xie, J., Kiefel, M., Sun, M. T., & Geiger, A. (2016). Semantic instance annotation of street scenes by 3d to 2d label transfer. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 3688-3697)
83. Carlevaris-Bianco, N., Ushani, A. K., & Eustice, R. M. (2016). University of Michigan North Campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9), 1023-1035.
84. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., ... & Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10), 1157-1163.
85. KITTI Dataset Webpage. Διαθέσιμο στο: [The KITTI Vision Benchmark Suite \(cvlibs.net\)](https://www.cvlibs.net/) (Ανακτήθηκε 23/03/2024).
86. Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., & Stachniss, C. (2023). Kiss-icp: In defense of point-to-point icp—simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, 8(2), 1029-1036.
87. Github Official Webpage. Διαθέσιμο στο: [GitHub: Let's build from here · GitHub](https://github.com) (Ανακτήθηκε 23/03/2024).
88. MathWorks MATLAB Webpage. Διαθέσιμο στο: [MathWorks - MATLAB and Simulink Conferences - MATLAB & Simulink](https://www.mathworks.com) (Ανακτήθηκε 23/03/2024).

89. MathWorks MATLAB SLAM Documentation. Διαθέσιμο στο: [Choose SLAM Workflow Based on Sensor Data - MATLAB & Simulink - MathWorks Switzerland](#) (Ανακτήθηκε 23/03/2024).
90. C. Olsson, F. Kahl and M. Oskarsson, "Branch-and-Bound Methods for Euclidean Registration Problems," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 783-794, May 2009, doi: 10.1109/TPAMI.2008.131
91. Reddy, B. S., & Chatterji, B. N. (1996). An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing*, 5(8), 1266-1271.
92. Dimitrievski, M., Van Hamme, D., Veelaert, P., & Philips, W. (2016). Robust matching of occupancy maps for odometry in autonomous vehicles. In *11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016)* (Vol. 3, pp. 626-633).
93. Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In *International conference on machine learning* (pp. 1889-1897). PMLR.
94. Unreal Engine Official Website. Διαθέσιμο στο: [The most powerful real-time 3D creation tool - Unreal Engine](#) (Ανακτήθηκε 23/03/2024).
95. Livox Avia Sensor - Official Livox Webpage. Διαθέσιμο στο: [Avia LiDAR sensor - Livox \(livoxtech.com\)](#) (Ανακτήθηκε 23/03/2024).
96. Τόπαλη Χ., 2023. Συγκριτική Αξιολόγηση Σαρωτών SLAM. Διπλωματική εργασία, ΣΑΤΜ ΕΜΠ.
97. ROS Official Webpage. Διαθέσιμο στο: [ROS: Home](#) (Ανακτήθηκε 23/03/2024).
98. Lin, J., & Zhang, F. (2020, May). Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3126-3131). IEEE.
99. Livox Mapping GitHub Repository (19/06/2020). Διαθέσιμο στο: [GitHub - Livox-SDK/livox_mapping: A mapping package for Livox LiDARs](#) (Ανακτήθηκε 23/03/2024).
100. Loam Livox GitHub Repository (07/06/2019). Διαθέσιμο στο: [GitHub - hku-mars/loam_livox: A robust LiDAR Odometry and Mapping \(LOAM\) package for Livox-LiDAR](#) (Ανακτήθηκε 23/03/2024).
101. Lin, J., & Zhang, F. (2019). A fast, complete, point cloud based loop closure for LiDAR odometry and mapping. *arXiv preprint arXiv:1909.11811*.
102. Oracle Virtual Box Official Webpage. Διαθέσιμο στο: [Oracle VM VirtualBox](#) (Ανακτήθηκε 23/03/2024).

103. Livox ROS Driver GitHub Repository (08/04/2018). Διαθέσιμο στο: [GitHub - Livox-SDK/livox_ros_driver: Livox device driver under ros, support Lidar Mid-40, Mid-70, Tele-15, Horizon, Avia.](#) (Ανακτήθηκε 23/03/2024).
104. Ubuntu Official Website. Διαθέσιμο στο: [Download Ubuntu Desktop | Download | Ubuntu](#) (Ανακτήθηκε 23/03/2024).
105. Ceres Solver Official Website. Διαθέσιμο στο: [Ceres Solver — A Large Scale Non-linear Optimization Library \(ceres-solver.org\)](#) (Ανακτήθηκε 23/03/2024).
106. R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 1-4, doi: 10.1109/ICRA.2011.5980567.
107. FAST-LIO, FAST-LIO2 GitHub Repository (12/10/2020). Διαθέσιμο στο: [GitHub - hku-mars/FAST_LIO: A computationally efficient and robust LiDAR-inertial odometry \(LIO\) package](#) (Ανακτήθηκε 23/03/2024).
108. Faster-LIO GitHub Repository (17/02/2022). Διαθέσιμο στο: [GitHub - gaoxiang12/faster-lio: Faster-LIO: Lightweight Tightly Coupled Lidar-inertial Odometry using Parallel Sparse Incremental Voxels](#) (Ανακτήθηκε 23/03/2024).
109. Kiss-ICP GitHub Repository (13/09/2022). Διαθέσιμο στο: [GitHub - PRBonn/kiss-icp: A LiDAR odometry pipeline that just works](#) (Ανακτήθηκε 23/03/2024).
110. CloudCompare Official Webpage. Διαθέσιμο στο: [CloudCompare - Open Source project](#) (Ανακτήθηκε 23/03/2024).
111. Αϊβαλιώτης Α., 2021. Η Χρήση της Τεχνολογίας SLAM σε Υπόγειες Εκμεταλλεύσεις. Μεταπτυχιακή εργασία, ΣΑΤΜ ΕΜΠ.