



**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ &
ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ**
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«Διερεύνηση και συγκριτική αξιολόγηση αλγορίθμων ταυτόχρονου
εντοπισμού και χαρτογράφησης (SLAM)**

**Πρακτική εφαρμογή με χρήση αλγορίθμου ORB SLAM2 και
Microsoft Kinect σε περιβάλλον Linux»**

Την παρούσα εργασία συνέταξε ο μεταπτυχιακός φοιτητής:

Τσαντεκίδης Ιωάννης

Αριθμός Μητρώου: 1708

Εξάμηνο Σπουδών: Δ΄

Επιβλέποντες καθηγητές: Λ. Γραμματικόπουλος – Ε. Πέτσα

3^ο μέλος εξεταστικής επιτροπής: Η. Καλησπεράκης

ΑΘΗΝΑ 2019

Περιεχόμενα

1. Πρόλογος.....	4
2. Εισαγωγή.....	5
3. SLAM και κριτήρια σύγκρισης αλγορίθμων SLAM.....	6
3.1 Τι σημαίνει η τεχνολογία SLAM.....	6
3.2 Γιατί χρειάζεται η τεχνολογία SLAM	6
3.3 “How it works”	7
3.4 Κατηγορίες SLAM	8
3.5 Κριτήρια σύγκρισης αλγορίθμων SLAM	22
Στα επόμενα θα παρουσιαστούν παραδείγματα αλγορίθμων SLAM.....	23
Libviso2 (Library for Visual Odometry2):	23
ORB – SLAM2.....	25
Rovio (Robust Visual Inertial Odometry):.....	26
Cartographer	26
OKVIS (Open Keyframe-based Visual-Inertial SLAM).....	28
VINS-Mono (Versatile Monocular Visual-Inertial State Estimator):	28
SVO+MSF (Semi – Direct Visual Odometry + Multi-Sensor Fusion):.....	29
4. ORB SLAM 2 και συνοπτική επεξήγηση λειτουργίας αλγορίθμου.....	30
4.1 ORB SLAM2.....	30
4.2 Τρόπος λειτουργίας ORB SLAM2	31
4.3 Close stereo keypoints, Far stereo keypoints, Monocular keypoints.....	32
4.4 System bootstrapping (Εκκίνηση του συστήματος)	33
4.5 Συνόρθωση δέσμης σε μονοεικονικά και διεικονικά συστήματα (με περιορισμούς) 33	
4.6 Datasets για έλεγχο του ORB SLAM2	36
5. UcoSLAM (Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers)	38
5.1 UcoSLAM	38
5.2 Τρόπος λειτουργίας UcoSLAM.....	39
6. Εφαρμογή ORB SLAM2 σε indoor σενάριο	40
6.1 Επεξεργασία βίντεο με χρήση ORB SLAM2, ανάλυση λειτουργίας αλγορίθμου ..	40
6.1.1 ORB Extractor	41
6.1.2 Αρχικοποίηση (Initialization)	43
6.1.3 Tracking.....	45
6.1.4 Επανεντοπισμός.....	46
6.1.5 Local mapping	47
6.1.6 Loop closing	49
7. Εφαρμογή αλγορίθμου με δεδομένα εισόδου από το Microsoft Kinect V1.....	53

7.1	Χρήση Microsoft Kinect, ORB SLAM2 και ROS (Robot Operating System)	53
7.2	Χρήση Raspberry Pi car σε συνδυασμό με ORB SLAM2 και ROS	56
8.	Συμπεράσματα.....	59
	BIBΛΙΟΓΡΑΦΙΑ.....	62

1. Πρόλογος

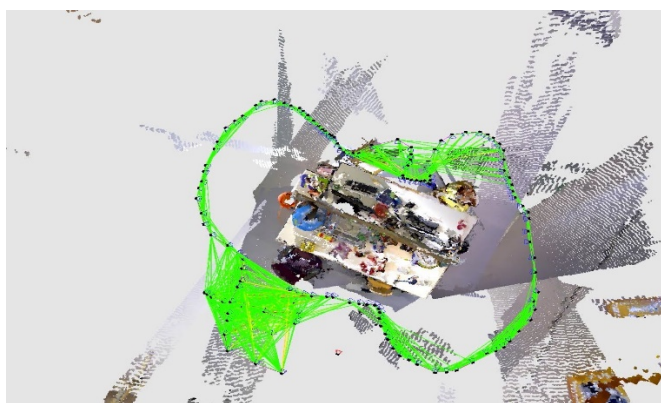
Η 3D ανακατασκευή (3D reconstruction) μέσω εικόνων μπορεί να χρησιμοποιηθεί, μεταξύ άλλων, ως ένα είδος «πλατφόρμας» για την παρακολούθηση της απόδοσης δραστηριοτήτων, π.χ. σε εργοτάξιο, όπως η παρακολούθηση προόδου στις κατασκευές (construction progress monitoring), ο έλεγχος (inspection) και διάσωση μετά από κάποια καταστροφή (post disaster rescue). Συγκριτικά με το προϊόν άλλων αισθητήρων, οι εικόνες RGB έχουν τα πλεονεκτήματα του χαμηλού κόστους, της πλούσιας υφής και της ευκολίας στην εφαρμογή τους, αποτελώντας την κύρια μέθοδο της 3D ανακατασκευής, μεταξύ άλλων στη βιομηχανία. Ωστόσο η συμβατική 3D ανακατασκευή η βασιζόμενη σε εικόνες απαιτεί σχετικά μεγάλο χρονικό διάστημα για την απόκτηση και επεξεργασία των δεδομένων των εικόνων. Αυτό φυσικά συνιστά ένα πρόβλημα, διότι π.χ. πολλές φορές υπάρχουν αυστηρά χρονοδιαγράμματα τα οποία πρέπει να τηρηθούν. Έτσι, η σχετική τεχνολογία και τεχνογνωσία αναπτύσσονται συνεχώς. Η πρόσφατη πρόοδος στον Οπτικό Ταυτόχρονο Εντοπισμό και Χαρτογράφηση (Visual Simultaneous Localization and Mapping – Visual SLAM) έκανε δυνατή την ανακατασκευή 3D χαρτών σε πραγματικό χρόνο. Επίσης, με την χρήση των UAVs (μη επανδρωμένα εναέρια οχήματα) και της μεθόδου SLAM έγινε δυνατή η χαρτογράφηση περιοχών οι οποίες είναι μη προσβάσιμες από εξοπλισμούς εδάφους. Παρά τα πλεονεκτήματα της Visual SLAM και των UAVs, η μέθοδος δεν έχει ερευνηθεί πλήρως έως τώρα στον κατασκευαστικό κλάδο. Σε αυτό το πλαίσιο, στόχος της παρούσας εργασίας είναι να παρουσιάσει μια μελέτη για την χρήση της Visual SLAM στην 3D ανακατασκευή σε πραγματικό χρόνο (real time 3D reconstruction).



Εικόνα 1. Visual SLAM

2. Εισαγωγή

Τα τελευταία χρόνια, η 3D ανακατασκευή και χαρτογράφηση με τη χρήση μη επανδρωμένων οχημάτων (UAV) έχει αποκτήσει σοβαρό ενδιαφέρον για πολλές εφαρμογές (από τον στρατιωτικό τομέα έως την κατασκευαστική βιομηχανία). Αντίθετα με τα οχήματα εδάφους, τα UAVs είναι ταχύτερα, ασφαλέστερα και πιο αποδοτικά. Μπορούν εύκολα να φτάσουν σε περιοχές απρόσιτες για επανδρωμένα οχήματα και να αναλάβουν εργασίες επικίνδυνες για τον άνθρωπο. Επί του παρόντος, η κύρια τεχνική για την οποία είναι εξοπλισμένα τα UAVs με αισθητήρες είναι η φωτογραμμετρία. Σε σύγκριση με άλλους αισθητήρες απομακρυσμένους από το αντικείμενο (remote sensors), όπως συσκευές LiDAR, ραντάρ και υπερήχου, η τεχνική της φωτογραμμετρίας βασίζεται σε οπτικό αισθητήρα που έχει τα πλεονεκτήματα της πλούσιας υφής, του χαμηλού κόστους και βάρους, και μπορεί να ικανοποιήσει το απαιτούμενο επίπεδο ακρίβειας για τις περισσότερες εφαρμογές. Με κάμερες επί του σκάφους, το UAV μπορεί να λαμβάνει εναέριες εικόνες κατά τη διάρκεια της πτήσης και να παράγει 3D νέφος σημείων (point cloud) με την χρήση αυτόματων τεχνικών στην μετέπειτα επεξεργασία των εικόνων. Πρόσφατες εφαρμογές έδειξαν τις δυνατότητες της χρήσης UAVs με σκοπό την παρακολούθηση της προόδου των κατασκευών (survey of civil infrastructures) καθώς και για ιστορικούς-τουριστικούς σκοπούς. Ωστόσο, λόγω του μεγάλου υπολογιστικού φόρτου της επεξεργασίας εικόνων και της χρήσης σύνθετων αλγορίθμων από την όραση υπολογιστών, η επεξεργασία τέτοιου είδους μοντέλων παρουσιάζει δυσκολίες. Αυτό μπορεί σε σημαντικό βαθμό να ξεπεραστεί με την τεχνολογία SLAM. Γενικά, ως SLAM ορίζεται η τεχνολογία στην οποία ένα ρομπότ ή μια συσκευή μπορεί να δημιουργήσει έναν χάρτη του περιβάλλοντός του και να προσανατολισθεί σωστά στον χάρτη σε πραγματικό χρόνο (Εικόνα 2). Η συγκεκριμένη τεχνολογία είναι ιδιαίτερα σημαντική για την εικονική (VR) και την επαυξημένη πραγματικότητα (AR) επιστήμη.



Εικόνα 2. 3D νέφος σημείων με χρήση Visual SLAM

3. SLAM και κριτήρια σύγκρισης αλγορίθμων SLAM

3.1 Τι σημαίνει η τεχνολογία SLAM

Προκειμένου να κατανοηθεί πώς λειτουργεί η τεχνολογία SLAM, μπορεί κανείς να αναλύσει το ακρωνύμιο και να εστιάσει στους όρους ‘Localization’ και ‘Mapping’. Δηλαδή το σύστημα προσπαθεί ταυτόχρονα να εντοπίσει την θέση κάποιου αντικειμένου/αισθητήρα με σημείο αναφοράς το περιβάλλον του και να χαρτογραφήσει τη δομή (layout) και το γενικότερο πλαίσιο (framework) του περιβάλλοντος στο οποίο βρίσκεται η συσκευή. Αυτό μπορεί να πραγματοποιηθεί με μια σειρά αλγόριθμους, οι οποίοι ταυτόχρονα εντοπίζουν και χαρτογραφούν αντικείμενα [1, 2].



Εικόνα 3. SLAM και Virtual Reality

Εφαρμογές αυτών των αλγορίθμων βρίσκονται σε «αυτο-οδηγούμενα» οχήματα, μη επανδρωμένα εναέρια οχήματα, αυτόνομα υποβρύχια οχήματα, ρομπότ, ακόμη και μέσα στο ανθρώπινο σώμα (Body-SLAM) [19].

3.2 Γιατί χρειάζεται η τεχνολογία SLAM

Όπως είναι γνωστό, είναι δεδομένη πλέον η δυνατότητα της περιήγησης σε ‘γνωστούς’ χώρους. Τι συμβαίνει όμως όταν αυτοί είναι ‘άγνωστοι’; Η SLAM εντοπίζει ένα άγνωστο περιβάλλον και βοηθά στην πλοήγηση σε χώρους για τους οποίους δεν διατίθεται κάποιος χάρτη ή σήμα GPS. Η SLAM προσφέρεται ιδιαίτερα για καταστάσεις όπου δεν προϋπάρχουν σημεία αναφοράς.

Η SLAM είναι άλλωστε η τεχνολογία που κρύβεται πίσω από τα αυτο-οδηγούμενα οχήματα της Google. Για να επιτευχθεί αυτό, τα εν λόγω οχήματα είναι εξοπλισμένα με έναν αισθητήρα LIDAR στην οροφή τους, ο οποίος δημιουργεί έναν 3D χάρτη του περιβάλλοντος (σε περίπου 10 δευτερόλεπτα!). Η γρήγορη απόκριση σε αυτού του είδους τα οχήματα είναι επιτακτική διότι μπορεί να κινούνται με υψηλές ταχύτητες και επιταχύνσεις. Αυτές οι χαρτογραφήσεις έχουν προστεθεί πάνω στους ήδη υπάρχοντες χάρτες της Google. Τα αυτόνομα συστήματα παίρνουν αποφάσεις οδήγησης χρησιμοποιώντας στατιστικούς αλγορίθμους, όπως είναι τα φίλτρα Bayes και τα μοντέλα Monte Carlo. Τα τελευταία είναι απαραίτητα στην ανάλυση κινδύνων και στην εύρεση διαφορετικών πιθανοτήτων διαφορετικών αποτελεσμάτων. Έπειτα από τέτοια κριτική ανάλυση το σύστημα μπορεί να παίρνει ακριβείς μετρητικές αποφάσεις.

Η πλοήγηση στον Άρη [20] χρησιμοποιεί επίσης την τεχνολογία SLAM. Η Wikitude, εταιρία η οποία δραστηριοποιείται στη δημιουργία λογισμικού για επαυξημένη πραγματικότητα, χρησιμοποιεί επίσης την συγκεκριμένη τεχνολογία για τις εφαρμογές της.

Τα αυτόνομα συστήματα αποτελούν τις κατ' εξοχήν εφαρμογές που χρησιμοποιούν την τεχνολογία SLAM, και εταιρίες όπως η Cortica [21] έχουν ήδη κάνει χρήση της στον εντοπισμό και χαρτογράφηση για αυτο-οδηγούμενα οχήματα στο Ισραήλ. Όλοι οι 'γίγαντες' της τεχνολογίας (Google, Apple, Facebook κ.λ.π) χρησιμοποιούν την SLAM. Ειδικότερα η Apple, στην προσπάθειά της να 'εισχωρήσει' στον χώρο της τεχνολογίας AR/VR (Augmented Reality/Virtual Reality) έχει δημιουργήσει το ARKit, το οποίο εξαρτάται σε μεγάλο βαθμό από τη SLAM.

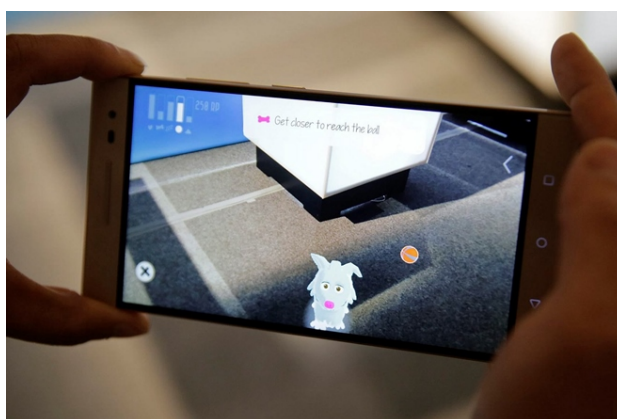
Αρα συμπεραίνεται ότι η SLAM είναι πολύ σημαντική για τις εφαρμογές AR. Όταν το εικονικό περιεχόμενο αντιδρά σωστά με τα πραγματικά αντικείμενα στο περιβάλλον τους, τότε η επαυξημένη πραγματικότητα είναι επιτυχής (Εικόνα 4) [1, 2].

3.3 “How it works”

Η συγκεκριμένη τεχνολογία είναι στην ουσία ένα πρόβλημα βελτιστοποίησης. Αρχικά οι αισθητήρες της μηχανής συλλέγουν οπτικά δεδομένα από τον φυσικό κόσμο ως σημεία αναφοράς. Τα σημεία αυτά βοηθούν την μηχανή να διακρίνει π.χ. τοίχους, πατώματα ή άλλα 'εμπόδια'. Η πλατφόρμα AR *Tango* της Google, χρησιμοποιεί προηγμένη SLAM ώστε να αλληλεπιδρά με το περιβάλλον (Εικόνα 5).



Εικόνα 4. SLAM και Augmented Reality



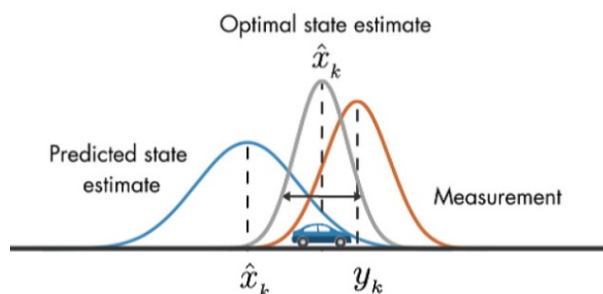
Εικόνα 5. TANGO platform (Google)

Στη συνέχεια, οι μετρήσεις λαμβάνονται συνεχώς καθώς η συσκευή κινείται μέσα στο περιβάλλον και η SLAM φροντίζει για τις ανακρίβειές τους υπολογίζοντας τον 'θόρυβο'. Διαφορετικοί αισθητήρες χρησιμοποιούν διαφορετικούς αλγορίθμους. Η SLAM κάνει φυσικά εκτεταμένη χρήση μαθηματικών και στατιστικών αλγορίθμων. Ένας από αυτούς είναι το φίλτρο Kalman. Το συγκεκριμένο φίλτρο λαμβάνει υπόψη μια σειρά από μετρήσεις με την πάροδο του χρόνου, αντί για μόνο μια. Τότε μπορεί να 'προβλέψει' την θέση των άγνωστων μεταβλητών, στην προκειμένη περίπτωση άγνωστα σημεία σε 3D αντικείμενα στο πεδίο της μηχανής (Εικ. 6) [1, 2].

3.4 Κατηγορίες SLAM

Η τεχνολογία SLAM βασίζεται σε διάφορες θεωρίες ανάλογα με τον σκοπό. Οι τυπικές θεωρίες περιλαμβάνουν αυτόνομη κίνηση, ρομποτική και όραση υπολογιστών. Οι διάφοροι αισθητήρες και εργαλεία μπορούν να χρησιμοποιηθούν για διαφορετικούς σκοπούς της SLAM. Το αποτέλεσμα

των παραπάνω είναι ότι η SLAM μπορεί να κατηγοριοποιηθεί σε LIDAR SLAM, RADAR SLAM, AR SLAM και VISUAL SLAM.



Εικόνα 6. Φίλτρο Kalman

Ας θεωρηθεί ως παράδειγμα ένα αυτόνομο όχημα και η επαυξημένη πραγματικότητα. Κατ' αρχάς, η SLAM εφαρμόζεται σε αυτόνομα οχήματα με δύο λέιζερ, LIDAR (Light Imaging Detection and Ranging) και RADAR (Radio Detection and Ranging).

LIDAR SLAM

Δεδομένου ότι το LIDAR χρησιμοποιεί οπτικά κύματα και το RADAR ραδιοκύματα, εμφανίζουν και διαφορετικά πλεονεκτήματα και μειονεκτήματα. Το LIDAR μπορεί να εντοπίσει μικρά αντικείμενα με μεγάλη συχνότητα και να δει την ακριβή απόσταση, αλλά δεν μπορεί να χρησιμοποιηθεί σε συννεφιασμένες ή βροχερές ημέρες [12]. Πιο αναλυτικά, ο αισθητήρας λέιζερ είναι απαραίτητος για ένα πλήρως αυτοματοποιημένο αυτοκίνητο. Το Lidar είναι απαραίτητο συστατικό για τους κατασκευαστές αυτόνομων οχημάτων. Ωστόσο, σε σύγκριση με την αυτόνομη οδήγηση, το Lidar μπορεί να φαίνεται πιο κατάλληλο για γενικές εφαρμογές ρομποτικής, όπως: λιγότερο ακριβείς απαιτήσεις, μικρότερο κόστος και μικρότερα λάθη.

Το Lidar μπορεί να βοηθήσει το ρομπότ να 'αντιληφθεί' την τοπογραφία ενός άγνωστου περιβάλλοντος, να παρέχει καλή περιβαλλοντική γνώση για την επακόλουθη θέση, την πλοήγηση και ακόμα και την αλληλεπίδραση ανθρώπου-υπολογιστή και γενικά να αυξήσει την ευφυΐα των ρομπότ.

Η ενσωμάτωση ενός σαρωτή λέιζερ και του Lidar λύνει ένα σημαντικό πρόβλημα στη ρομποτική: παρέχει στο ρομπότ ένα ζευγάρι 'μάτια'. Μέσω του Lidar, το ρομπότ μπορεί να εκπέμπει υπέρυθρο φως στον έξω κόσμο, να χαρτογραφεί τη θέση του μέσω τριγωνισμού ή Time of Flight (ToF), και στη συνέχεια να ανακατασκευάσει το περιβάλλον με υψηλή ακρίβεια

με σκοπό την πλοήγηση του robot στο περιβάλλον [13].

Παρατηρώντας τον συνδυασμό Lidar και ρομποτικής, μπορεί να διαπιστώσει κανείς ότι το μέγεθος, το κόστος και η απόδοση του Lidar είναι ιδανικά για ρομπότ. Όχι μόνο τα ρομπότ μπορούν να ολοκληρώσουν εργασίες που απαιτούν ακρίβεια, αλλά και να ικανοποιούν τις ανάγκες των χρηστών σε σύνθετα σενάρια.

Εύρος (range): Ο κύριος σκοπός του laser είναι η μέτρηση απόστασης. Στο παρελθόν, τα laser που χρησιμοποίησαν τεχνικές τριγωνισμού είχαν δυσκολία να φτάσουν σε απόσταση > 20 m. Ωστόσο, τώρα τα laser μπορούν εύκολα να φτάσουν σε απόσταση έως 18 m. Επιπλέον με βελτίωση των αλγορίθμων η απόσταση μπορεί να αυξηθεί στα 25 m [13].

Συχνότητα μέτρησης (measurement frequency): Οι υψηλότερες συχνότητες σάρωσης επιτρέπουν στο ρομπότ να επιτύχει ταχύτερη κίνηση και να αυξήσει την ποιότητα της κατασκευής του χάρτη. Τώρα, τα laser στην αγορά είναι σε θέση να επιτύχουν 16.000 μετρήσεις / sec και μπορούν να επιτευχθούν ακόμη υψηλότερες συχνότητες.

Ανίχνευση μαύρων και λευκών αντικειμένων (detection of black and white objects): Επί του παρόντος, η ονομαστική απόσταση του λέιζερ βασίζεται κυρίως στο φως από ένα αντικείμενο διάχυτης ανάκλασης (όπως λευκό χαρτί) με ανακλαστικότητα 90%. Ωστόσο το Lidar κατάφερε να επιτύχει σταθερή ανίχνευση μαύρων αντικειμένων περί μια ακτίνα 10 m.

Χρήση σε εσωτερικούς και εξωτερικούς χώρους (indoor and outdoor scenarios): Εκτός από την ικανότητα ανίχνευσης σκοτεινών αντικειμένων, είναι ιδιαίτερα σημαντικό ότι το Lidar και οι σαρωτές λέιζερ μπορούν να λειτουργούν υπό διαφορές συνθήκες φωτεινότητας του περιβάλλοντος. Αναβαθμίζοντας τον αλγόριθμο επεξεργασίας και με ρύθμιση του αισθητήρα, το παραδοσιακό ραντάρ λέιζερ μπορεί εύκολα να επιτύχει σταθερή λειτουργία υπό τυπικές συνθήκες φυσικού φωτισμού (70KLux).

Όσο υψηλότερη είναι η απαίτηση ακρίβειας, τόσο μεγαλύτερο είναι το κόστος του σαρωτή λέιζερ και του Lidar. Αλλά το Lidar που χρησιμοποιείται στα ρομπότ δεν χρειάζεται να επιτύχει την ίδια ακρίβεια με τα ραντάρ που χρησιμοποιούνται για τα αυτοκινούμενα οχήματα.

Αλλά μετά την επίλυση των παραπάνω τεχνικών προβλημάτων και προβλημάτων κόστους, σημαίνει ότι το ραντάρ λέιζερ μπορεί να χρησιμοποιηθεί μόνο σε ρομπότ; Φυσικά και όχι!

Εκτός από το πεδίο της ρομποτικής, το Lidar παίζει σημαντικό ρόλο σε πολλούς άλλους τομείς όπως εύρεση θέσης και χαρτογράφηση, 3D μοντελοποίηση και multimedia interaction.

Ως συμπέρασμα όλων των παραπάνω είναι ότι η καλύτερη επιλογή του

Lidar για το ρομπότ εξαρτάται από τις εκάστοτε συγκεκριμένες ανάγκες. Εκτός από τη σάρωση με λέιζερ 360 μοιρών, το γωνιακό άνοιγμα (field of view - FoV) και η αποφυγή σύγκρουσης είναι επίσης απαραίτητα χαρακτηριστικά για τη ρομποτική έρευνα. Τέλος ένα solid-state LiDAR με μεγάλο γωνιακό άνοιγμα μπορεί ταυτόχρονα να παράγει πληροφορίες βάθους και grayscale. Εν τω μεταξύ, δεν περιέχονται μηχανικά περιστρεφόμενα τμήματα, με συνέπεια να μπορεί να εξασφαλιστεί μεγαλύτερη αξιοπιστία και σταθερότητα [13] .

RADAR SLAM

Το RADAR, από την άλλη πλευρά, μπορεί να χρησιμοποιηθεί σε μεγαλύτερες αποστάσεις από το LIDAR και σε συννεφιασμένες ή βροχερές ημέρες, ωστόσο είναι δύσκολο να ανιχνευθούν μικρά αντικείμενα και να οριστεί η ακριβής απόσταση. Επομένως, ανάλογα με το σκοπό, μπορούν να εφαρμοστούν και οι δύο μέθοδοι ταυτόχρονα ή επίσης να επιλεγεί μόνο η μία, ανάλογα με την περίπτωση.

Ειδικότερα, για τα χερσαία οχήματα ή τα σκάφη επιφάνειας υπάρχουν τρεις κύριοι τύποι αισθητήρων που μπορούν να μετρήσουν το περιβάλλον: κάμερες, αισθητήρες λέιζερ και αισθητήρες ραντάρ. Οι κάμερες είναι πολύ πλούσιες σε πληροφορία και μπορούν να καλύψουν μεγάλες αποστάσεις αλλά είναι ευαίσθητες στο φως και τις καιρικές συνθήκες. Οι αισθητήρες λέιζερ παρέχουν εύρωστες και ακριβείς μετρήσεις εμβέλειας, αλλά επίσης είναι πολύ ευαίσθητες στις καιρικές συνθήκες. Το σήμα ραντάρ είναι συνήθως εκείνο που παρέχει λιγότερες πληροφορίες σε σχέση με τα προηγούμενα δύο και είναι επίσης πολύ ευαίσθητο όσον αφορά την αντανάκλαση του σήματος (π.χ. αντικείμενα στα οποία αντανακλάται το σήμα). Από την άλλη πλευρά, ο αισθητήρας ραντάρ λειτουργεί σχεδόν εξίσου καλά σε όλες τις καιρικές συνθήκες.

Το ραντάρ χρησιμοποιείται ως αισθητήρας καταγραφής εικόνων και με την εφαρμογή αλγορίθμων όρασης υπολογιστών εντοπίζονται τα σημεία αναφοράς. Τα σημεία αναφοράς που εμφανίζονται κατά τη διάρκεια των διαδοχικών σαρώσεων ραντάρ χρησιμοποιούνται στη συνέχεια για οπτική οδομετρία, που δίνει ταχύτητα, σχετική θέση και σχετική πορεία. Το κύριο κίνητρο για χρήση αυτών των χαρακτηριστικών σημείων (visual features) για να γίνει συντάυτιση (matching) των σαρώσεων radar αντί να γίνει προσπάθεια ένωσής (alignment) τους είναι ότι τα χαρακτηριστικά σημεία συνταυτίζονται πιο εύκολα παρά τις μεγάλες μετατοπίσεις και στροφές. Τα σημεία αυτά μπορούν προαιρετικά να αποθηκευτούν σε μορφή χάρτη που μπορεί να χρησιμοποιηθεί για την αναγνώριση περιοχών που έχει επισκεπτεί κανείς προηγουμένως.

Τα πλεονεκτήματα του ραντάρ, όπως η ικανότητά του να λειτουργεί σε

όλες τις καιρικές συνθήκες, έχουν οδηγήσει σε επαναχρησιμοποίησή του σε τεχνολογίες όπως είναι η SLAM. Τα ραντάρ μικροκυμάτων χρησιμοποιήθηκαν σε πειράματα SLAM, χωρίς την χρήση σημείων αναφοράς. Η SLAM χρησιμοποιείται τόσο σε αστικές όσο και σε αγροτικές περιοχές, ενώνοντας την τελευταία σάρωση radar με τον χάρτη που παράχθηκε χρησιμοποιώντας 3D συσχετίσεις (matching) για να εκτιμηθούν οι σχετικές κινήσεις του οχήματος. Ο χάρτης ραντάρ κατασκευάστηκε με διαδοχική πρόσθεση της τελευταίας ‘ευθυγραμμισμένης’ (aligned) σάρωσης ραντάρ στις προηγούμενες.

Τα millimeter wave radars χρησιμοποιούνται σε τμήματα της αυτοκινητοβιομηχανίας. Για παράδειγμα, η καμπυλότητα του δρόμου εκτιμάται με χρήση των αντανάκλασεων του ραντάρ που θα χρησιμοποιηθούν σε συστήματα για την προειδοποίηση και την αποφυγή σύγκρουσης. Το πρόβλημα της ένωσης των σαρώσεων ραντάρ υπάρχει επίσης στη μετεωρολογία, όπου οι παρατηρήσεις του ραντάρ διαστήματος και του ραντάρ εδάφους ενώνονται ώστε να αποκτηθεί μια πιο ολοκληρωμένη εικόνα του καιρού. Επίσης, η radar SLAM χρησιμοποιείται στα σκάφη. Όλες οι κινήσεις των σκαφών υπολογίζονται σε σχέση με την παγκόσμια θέση. Οι θέσεις των εντοπισμένων σημείων αναφοράς δεν μετρούνται σε παγκόσμιο σύστημα αναφοράς αλλά σε σχέση με το σκάφος. Επομένως, χρησιμοποιούνται δύο συστήματα συντεταγμένων, ένα παγκόσμιο για την θέση του σκάφους και όλων των σημείων αναφοράς, και ένα τοπικό που παρουσιάζει τις θέσεις των χαρακτηριστικών σημείων του σκάφους [14].

Στη συνέχεια περιγράφεται η SLAM που εφαρμόζεται στην επαυξημένη πραγματικότητα. Επειδή αυτή βασίζεται κυρίως σε κινητές συσκευές, η τεχνολογία SLAM συνεργάζεται με κάμερες σε κινητές συσκευές. Προκειμένου να ελαχιστοποιηθεί η κατανάλωση ενέργειας και η χρήση της CPU, μπορεί να γίνει χρήση μονοσκοπικών καμερών. Επίσης, προκειμένου να βελτιωθεί η ακρίβεια, γίνεται χρήση αισθητήρων IMUs (Inertial Measurement Units – αδρανειακά συστήματα) ή καμερών βάθους (depth cameras) [12]. Για την επαυξημένη πραγματικότητα, η συσκευή πρέπει να γνωρίζει τη 3D θέση της στο περιβάλλον. Η διαδικασία για να επιτευχθεί αυτό είναι η SLAM.

Αισθητήρες

Όταν εκκινείται για πρώτη φορά μια εφαρμογή επαυξημένης πραγματικότητας με το Google ARCore [23], το ARKit της Apple ή το Microsoft Mixed Reality, το σύστημα δεν γνωρίζει πολλά για το περιβάλλον. Αρχίζει την επεξεργασία δεδομένων από διάφορες πηγές – κυρίως από την κάμερα. Για τη βελτίωση της ακρίβειας, η συσκευή συνδυάζει δεδομένα από άλλους χρήσιμους αισθητήρες όπως το

επιταχυνσιόμετρο και το γυροσκόπιο. Με βάση αυτά τα δεδομένα, ο αλγόριθμος έχει δύο στόχους:

- Δημιουργία ενός χάρτη του περιβάλλοντος
- Εντοπισμός της συσκευής σε αυτό το περιβάλλον

Για ορισμένα σενάρια, αυτό είναι εύκολο. Αν υπάρχει η δυνατότητα τοποθέτησης στόχων σε γνωστές θέσεις, πρέπει απλώς να πραγματοποιηθεί ο τριγωνισμός των αποστάσεων και έτσι γνωρίζει κανείς πού ακριβώς βρίσκεται. Για άλλες εφαρμογές, το GPS μπορεί να είναι κατάλληλο.

Ωστόσο, η Επαυξημένη Πραγματικότητα (Augmented Reality) συνήθως δεν έχει την πολυτέλεια τοποθέτησης στόχων σε σημεία με γνωστές συντεταγμένες (εξαιτίας της συνεχούς κίνησης). Ούτε το GPS είναι αρκετά ακριβές – ειδικά σε εσωτερικούς χώρους. Σε ιδεατές περιπτώσεις, θα υπήρχαν τέλειες πληροφορίες σχετικά με την ακριβή τοποθεσία όλων. Αυτό περιλαμβάνει τη θέση ενός στόχου, καθώς και τη θέση ενός ρομπότ. Άρα, θα ήταν εύκολο να υπολογιστεί η ακριβής σχέση μεταξύ του χώρου και της τρέχουσας θέσης. Εάν θα έπρεπε να μετακινηθεί το ρομπότ σε νέα θέση, θα μπορούσαν να εξαχθούν πληροφορίες σχετικά με το πού και πώς έπρεπε να μετακινηθεί. Δυστυχώς, στο πραγματικό σενάριο της AR SLAM υπάρχουν βέβαια ατελείς γνώσεις οι οποίες έχουν ως αποτέλεσμα αβεβαιότητες.

Τα σημεία έχουν χωρικές σχέσεις μεταξύ τους. Ως αποτέλεσμα αυτού, λαμβάνεται μια πιθανότητα κατανομής των θέσεων σχετικά με το πού θα μπορούσαν να βρίσκονται. Για ορισμένα σημεία, ενδέχεται να υπάρχει μεγαλύτερη ακρίβεια. Για άλλα, η αβεβαιότητα μπορεί να είναι μεγάλη. Συχνά χρησιμοποιούμενοι αλγόριθμοι για τον υπολογισμό των θέσεων με βάση τις αβεβαιότητες είναι η εκτίμηση του Extended Kalman Filter ή η Συνόρθωση Δέσμης – Bundle Adjustment (BA).

Λόγω των σχέσεων μεταξύ των σημείων, κάθε νέα ενημέρωση αισθητήρων επηρεάζει όλες τις θέσεις και ενημερώνει ολόκληρο τον χάρτη. Για να είναι επιτυχημένη η Επαυξημένη Πραγματικότητα, ο αλγόριθμος SLAM πρέπει να ανταποκριθεί στις ακόλουθες προκλήσεις:

- Άγνωστος χώρος.
- Μη ελεγχόμενη κάμερα. Για το τρέχον AR με κινητό τηλέφωνο, αυτή είναι συνήθως μόνο μια μονοσκοπική κάμερα.
- Πραγματικός χρόνος (real-time).
- Μηδενική ολίσθηση (drift-free).

Αλλά το ερώτημα που παραμένει είναι ποια είναι τα δεδομένα που χρησιμοποιούνται στην AR SLAM; Η απάντηση είναι όλα τα δεδομένα αδρανειακών αισθητήρων μαζί με τα δεδομένα της κάμερας. Ακολουθεί ο τρόπος με τον οποίο χρησιμοποιούνται από τις μεγάλες εταιρίες πληροφορικής (Google, Microsoft, Apple).

Google: Σε δημοσιοποιημένα έγγραφα της [22], η Google περιγράφει ότι το ARCore χρησιμοποιεί μια διαδικασία που ονομάζεται παράλληλη οδομετρία και χαρτογράφηση – η οποία ουσιαστικά είναι απλώς ένα άλλο όνομα του ευρύτερου όρου SLAM. Το όνομα δείχνει επίσης ότι ενσωματώνονται αδρανειακοί αισθητήρες για οδομετρία. Το άρθρο της Wikipedia σχετικά με το SLAM αναφέρει ότι η Google χρησιμοποιεί συνόρθωση δεσμής (bundle adjustment).

Microsoft: Πολλά από αυτά που συμβαίνουν στο HoloLens [24] και το Windows Mixed Reality βασίζονται σε προηγούμενη έρευνα για το Kinect. Ακόμα, η νέα ερευνητική λειτουργία του HoloLens επιτρέπει την πρόσβαση στα αποτελέσματα των αλγορίθμων SLAM που εκτελούνται. Η Microsoft παρέχει ορισμένα παραδείγματα για να ξεκινήσει κάποιος στο GitHub, συμπεριλαμβανομένης της ενσωμάτωσης του OpenCV.

Apple: Πριν από αρκετά χρόνια, η Apple απέκτησε το Metaio [25] και το FlyBy [26], που ανέπτυξαν αλγορίθμους SLAM. Η Apple χρησιμοποιεί Οπτική Αδρανειακή Οδομετρία (Visual Inertial Odometry) για να συνδυάσει επίσης την κάμερα με άλλους αισθητήρες τηλεφώνου ώστε να βελτιώσει την ακρίβεια.

Το μέλλον της Επαυξημένης Πραγματικότητας βασισμένης σε κινητά τηλέφωνα

Οι αλγόριθμοι SIFT είναι πολύτιμοι για πολλά ερευνητικά θέματα, όπως η αυτόνομη πλοήγηση αυτοκινήτων. Ως εκ τούτου η βελτίωση και η επέκταση της ποιότητας και της απόδοσής του είναι ένα πολύ σημαντικό ερευνητικό θέμα. Στο βιβλίο τους [*A Comparative Analysis of Tightly-coupled Monocular, Binocular, and Stereo VINS*](#) (2017) οι M.K. Paul, K. Wu, J.A. Hesch, E.D. Nerurkar και S.I. Roumeliotis συγκρίνουν αλγορίθμους που συνδυάζουν δεδομένα από αδρανειακά συστήματα (IMUs) με οπτικούς αισθητήρες όπως επίσης και από εμπορικές πλατφόρμες AR. Μεταξύ άλλων, συγκρίνουν και τον νέο αλγόριθμό τους με τους πλέον σύγχρονους αλγορίθμους ORB-SLAM2 και OKVIS, διαπιστώνοντας ότι κατάφεραν να βελτιώσουν ακόμα περισσότερο την αξιοπιστία και την ταχύτητα.

Φυσικά, τα συστήματα επαυξημένης πραγματικότητας επιχειρούν γενικά να κατανοήσουν όλο και περισσότερο το περιβάλλον. Ενώ τα ARKit και ARCore ξεκίνησαν με την παρακολούθηση απλών αεροπλάνων, η HoloLens προσπαθεί ήδη να συλλέξει περισσότερες γνώσεις μέσω της

Χωρικής Κατανόησης. Η Apple εισήγαγε την παρακολούθηση 3D αντικειμένων με το ARKit 2. Αποτέλεσμα αυτού είναι η ύπαρξη μιας πολύ βελτιωμένης ενοποίησης των εικονικών αντικειμένων και του πραγματικού περιβάλλοντος [15].

Visual SLAM

Τι είναι;

Η Visual SLAM είναι μια καινοτομία που αντιμετωπίζει τις αδυναμίες άλλων συστημάτων όρασης και πλοήγησης και έχει μεγάλες εμπορικές δυνατότητες. Δεν αναφέρεται σε κανένα συγκεκριμένο αλγόριθμο ή κομμάτι λογισμικού, αλλά στην διαδικασία προσδιορισμού της θέσης και του προσανατολισμού ενός αισθητήρα σε σχέση με το περιβάλλον του, ενώ παράλληλα απεικονίζει το περιβάλλον γύρω από αυτόν τον αισθητήρα. Υπάρχουν διάφοροι διαφορετικοί τύποι τεχνολογίας SLAM, ορισμένοι από τους οποίους δεν περιλαμβάνουν διόλου κάμερα. Η Visual SLAM είναι ένας συγκεκριμένος τύπος συστήματος SLAM που εκμεταλλεύεται την 3D όραση για να πραγματοποιεί λειτουργίες εντοπισμού θέσης και χαρτογράφησης όταν δεν είναι γνωστό ούτε το περιβάλλον ούτε η θέση του αισθητήρα. Η τεχνολογία Visual SLAM διατίθεται σε διάφορες μορφές, αλλά η γενική ιδέα λειτουργεί κατά τον ίδιο τρόπο σε όλα τα οπτικά συστήματα SLAM [16].

Πώς δουλεύει η τεχνολογία Visual SLAM

Τα περισσότερα συστήματα Visual SLAM λειτουργούν με την παρακολούθηση των καθορισμένων σημείων μέσω διαδοχικών πλαισίων της κάμερας για τον τριγωνισμό της 3D θέσης τους, ενώ ταυτόχρονα χρησιμοποιούν αυτές τις πληροφορίες για να προσεγγίσουν τη θέση της κάμερας. Βασικά, ο στόχος αυτών των συστημάτων είναι να χαρτογραφήσουν το περιβάλλον τους σε σχέση με τη δική τους θέση για τους σκοπούς της πλοήγησης. Αυτό είναι δυνατό με μια 3D κάμερα, σε αντίθεση με άλλες μορφές τεχνολογίας SLAM. Όσο υπάρχει επαρκής αριθμός σημείων που παρακολουθούνται σε κάθε πλαίσιο, τόσο ο προσανατολισμός του αισθητήρα όσο και η δομή του περιβάλλοντος μπορούν να γίνουν γρήγορα κατανοητά.

Όλα τα συστήματα Visual SLAM λειτουργούν συνεχώς για να ελαχιστοποιήσουν το σφάλμα επαναπροβολής ή τη διαφορά μεταξύ των προβαλλόμενων και πραγματικών σημείων, συνήθως μέσω της συνόρθωσης δέσμης. Τα συστήματα Visual SLAM πρέπει να λειτουργούν σε πραγματικό χρόνο, έτσι ώστε συχνά τα δεδομένα θέσης και τα δεδομένα χαρτογράφησης να υποβάλλονται χωριστά σε συνόρθωση δέσμης, αλλά ταυτόχρονα, για να

διευκολύνουν τις επεξεργασίες – δηλ. να γίνονται ταχύτατα πριν τελικά συγχωνευθούν [16].

Τι είδους εφαρμογές χρησιμοποιούν Visual SLAM;

Η Visual SLAM έχει τεράστιες δυνατότητες σε ένα ευρύ φάσμα εφαρμογών, όπως π.χ. οι εφαρμογές επαυξημένης πραγματικότητας. Η ακριβής προβολή εικονικών εικόνων στον φυσικό κόσμο απαιτεί ακριβή χαρτογράφηση του φυσικού περιβάλλοντος και μόνο η τεχνολογία Visual SLAM είναι σε θέση να παρέχει αυτό το επίπεδο ακρίβειας. Τα συστήματα Visual SLAM χρησιμοποιούνται επίσης στην ρομποτική. Για παράδειγμα, τα rovers και τα landers για την εξερεύνηση του Άρη χρησιμοποιούν συστήματα Visual SLAM για αυτόνομη πλοήγηση. Τα ρομπότ πεδίου στη γεωργία, καθώς και τα αεροσκάφη, μπορούν να χρησιμοποιούν την ίδια τεχνολογία για να ταξιδεύουν ανεξάρτητα γύρω από τα χωράφια. Τα αυτόνομα οχήματα χρησιμοποιούν τέτοια συστήματα για χαρτογράφηση και κατανόηση του κόσμου γύρω τους.

Μια μεγάλη δυνατότητα των συστημάτων Visual SLAM είναι ότι μπορούν να αντικαταστήσουν τον εντοπισμό θέσης και την πλοήγηση μέσω GPS σε ορισμένες εφαρμογές. Τα συστήματα GPS δεν είναι χρήσιμα σε εσωτερικούς χώρους ή σε μεγάλες πόλεις όπου η θέα του ουρανού παρεμποδίζεται. Τα συστήματα Visual SLAM επιλύουν τέτοια προβλήματα, καθώς δεν εξαρτώνται από τις δορυφορικές πληροφορίες και λαμβάνουν ακριβείς μετρήσεις του φυσικού κόσμου γύρω τους.

Η τεχνολογία Visual SLAM έχει πολλές πιθανές εφαρμογές και η ζήτηση για αυτήν την τεχνολογία πιθανότατα θα αυξηθεί καθώς βοηθά στην επαυξημένη πραγματικότητα, στα αυτόνομα οχήματα και άλλα προϊόντα, τα οποία καθίστανται εμπορικά βιώσιμα. Η ικανότητα ενός συστήματος να ‘αισθάνεται’ την θέση μιας κάμερας, καθώς και το περιβάλλον γύρω από αυτήν, χωρίς να γνωρίζει εκ των προτέρων τα σημεία δεδομένων είναι ιδιαίτερα δύσκολη. Η τεχνολογία Visual SLAM αποδεικνύεται εξαιρετικά αποτελεσματική για την αντιμετώπιση αυτής της πρόκλησης και επίσης αναδύεται ως μια από τις πιο εξελιγμένες τεχνολογίες όρασης που είναι διαθέσιμες. Η Visual SLAM είναι μια μόνο από τις πολλές καινοτόμες τεχνολογίες στον τομέα της ‘embedded vision’ [16].

Γεωμετρία της Visual SLAM

Extended Kalman Filter (EKF)

Το EKF είναι αναμφισβήτητη η πιο συνηθισμένη τεχνική για την εκτίμηση κατάστασης και βασίζεται στο φίλτρο Bayesian για το φιλτράρισμα και την πρόβλεψη μη γραμμικών λειτουργιών στις οποίες οι γραμμικές

προσεγγίσεις τους λαμβάνονται με χρήση μιας επέκτασης της σειράς Taylor.

Το EKF-SLAM χωρίζεται σε δύο βασικά στάδια: πρόβλεψη και ενημέρωση (διόρθωση). Στο βήμα της πρόβλεψης η μελλοντική θέση του ρομπότ εκτιμάται (προβλέπεται) βάσει της τρέχουσας θέσης του ρομπότ και των δεδομένων εισόδου που εφαρμόζονται για να αλλάξει η θέση του ρομπότ από το χρονικό βήμα k στο $k + 1$ (που συνήθως λαμβάνονται με τεχνικές dead reckoning). Dead reckoning σημαίνει ότι, ξεκινώντας από ένα σημείο με γνωστό διάνυσμα θέσης, προστίθεται η μεταβολή θέσης για να βρεθεί το διάνυσμα της νέας θέσης ($x_2 = x_1 + \Delta x_{12}$). Με αυτό τον τρόπο κάθε νέα θέση που προσδιορίζεται στηρίζεται στις προηγούμενες. Επίσης λαμβάνεται υπόψη ο θόρυβος της διαδικασίας. Η εξίσωση του γενικού μοντέλου κίνησης μπορεί να διατυπωθεί ως εξής:

$$X_{k+1} = f(X_k, U_k) + W_k$$

όπου X_{k+1} είναι μια εκτίμηση της μελλοντικής θέσης του ρομπότ, $f(X_k, U_k)$ είναι μια συνάρτηση της τρέχουσας εκτίμησης της θέσης του ρομπότ X_k και των δεδομένων εισόδου που εφαρμόζονται για να αλλάξει η θέση του ρομπότ από το χρονικό βήμα k στο $k + 1$, και W_k είναι η αβεβαιότητα της διαδικασίας που θεωρείται ότι είναι μη συσχετισμένος μηδενικός μέσος θόρυβος Gauss με μεταβλητότητα Q . Το γενικό μοντέλο μέτρησης (παρατήρησης) μπορεί να διαμορφωθεί ως εξής:

$$z = h(X_k) + V_k$$

όπου z είναι η μέτρηση με θόρυβο, $h(X_k)$ είναι η συνάρτηση παρατήρησης και V_k είναι η μέτρηση με θόρυβο, και θεωρείται ότι είναι μη-συσχετισμένος μηδενικός μέσος θόρυβος Gauss [18].

Συγχώνευση δεδομένων με χρήση μεθόδων φιλτραρίσματος

Η συγχώνευση δεδομένων είναι η διαδικασία συνδυασμού πληροφοριών από διάφορες πηγές/αισθητήρες προκειμένου να παρέχεται μια ισχυρή και πλήρης περιγραφή ενός περιβάλλοντος ώστε η πληροφορία που θα προκύψει να έχει λιγότερη αβεβαιότητα από ό,τι αν αυτές οι πηγές χρησιμοποιούνταν μεμονωμένα. Οι μέθοδοι συγχώνευσης δεδομένων διαδραματίζουν πολύ σημαντικό ρόλο σε αυτόνομα συστήματα και στη ρομποτική. Κατ' αρχήν, οι διαδικασίες αυτοματοποιημένης συγχώνευσης δεδομένων επιτρέπουν τη συγκέντρωση πληροφοριών για την παροχή επαρκούς γνώσης ώστε οι αποφάσεις να μπορούν να διαμορφώνονται και να εκτελούνται αυτόνομα. Το φίλτρο Kalman διαθέτει μια σειρά χαρακτηριστικών τα οποία το καθιστούν ιδανικό για την αντιμετώπιση σύνθετων προβλημάτων πολλαπλών αισθητήρων και συγχώνευσης δεδομένων. Συγκεκριμένα, η ρητή περιγραφή της διαδικασίας και των παρατηρήσεων επιτρέπει την ενσωμά-

τωση μιας ευρείας ποικιλίας διαφορετικών μοντέλων αισθητήρων στη βασική μέθοδο. Επιπλέον, η συνεπής χρήση στατιστικών μέτρων αβεβαιότητας καθιστά δυνατή την ποσοτική αξιολόγηση του ρόλου κάθε αισθητήρα ως προς τη συνολική απόδοση του συστήματος. Έτσι, τα φίλτρα Kalman (και τα EKF για μη γραμμικές εφαρμογές) έχουν γίνει πολύ κοινά εργαλεία για την επίλυση διαφόρων προβλημάτων συγχώνευσης δεδομένων. Ως εκ τούτου, αποτελεί μια λύση για τη συγχώνευση δεδομένων από πολλαπλές πηγές, προκειμένου να αντισταθμιστεί η απώλεια πληροφοριών που παρέχονται από το GPS [18].

Εξαγωγή και συνταύτιση χαρακτηριστικών σημείων (Feature extraction and matching)

Οι αισθητήρες όρασης έχουν προσελκύσει μεγάλο ενδιαφέρον των ερευνητών καθώς παρέχουν εικόνες που είναι πλούσιες σε πληροφορίες. Στις περισσότερες περιπτώσεις, οι πρώτες εικόνες πρέπει να υποβληθούν σε επεξεργασία για να εξαχθούν οι χρήσιμες πληροφορίες. Χαρακτηριστικά που παρουσιάζουν ενδιαφέρον κυμαίνονται από απλά χαρακτηριστικά σημείων, όπως γωνίες, έως πιο σύνθετα χαρακτηριστικά, όπως ακμές και κηλίδες, ακόμα και πολύπλοκα αντικείμενα όπως πόρτες και παράθυρα. Η παρακολούθηση χαρακτηριστικών (feature tracking) είναι η διαδικασία εύρεσης των αντιστοιχιών μεταξύ τέτοιων χαρακτηριστικών σε παρακείμενες εικόνες και είναι χρήσιμη όταν παρατηρούνται μικρές μεταβολές στις κινήσεις μεταξύ δύο πλαισίων (frames). Αντίστροφα, η συνταύτιση χαρακτηριστικών (feature matching) είναι η διαδικασία της ανεξάρτητης εξαγωγής χαρακτηριστικών σημείων και της αντιστοίχισής τους σε πολλαπλά πλαίσια.

Σημειακός τελεστής Harris (Harris corner detector)

Η μέθοδος ανίχνευσης γωνιών (corner), βασίζεται στην διάκριση περιοχών ομοιόμορφων ώστε να μην έχουν μεταβολή στις εντάσεις εικόνας μεταξύ γειτονικών περιοχών προς όλες τις κατευθύνσεις, περιοχών με σημαντική διακύμανση στην διεύθυνση που είναι κάθετη προς την ακμή και, τέλος, περιοχής με γωνία/σημείο (corner) όπου οι εντάσεις εικόνας έχουν μεγάλη διαφορά μεταξύ γειτονικών περιοχών σε όλες τις κατευθύνσεις. Αναλυτικότερα, μια γωνία μπορεί να ερμηνευτεί ως η διασταύρωση δύο ακμών, όπου ακμή είναι η απότομη αλλαγή στη φωτεινότητα της εικόνας. Οι γωνίες αποτελούν σημαντικά χαρακτηριστικά της εικόνας και γενικά ονομάζονται σημεία ενδιαφέροντος (interest point) τα οποία πρέπει να είναι αμετάβλητα σε μετατοπίσεις, στροφές και μεταβολές φωτισμού. Αν και οι γωνίες είναι μόνο ένα μικρό ποσοστό της εικόνας, περιέχουν τα πιο σημαντικά ίσως χαρακτηριστικά για την αποκατάσταση των ζητούμενων πληρο-

φοριών και μπορούν να χρησιμοποιηθούν για την ελαχιστοποίηση αποκλίσεων στα δεδομένα στην παρακολούθηση κίνησης και πολλές άλλες σχετικές εφαρμογές.

Ο τελεστής Harris παρέχει καλή επαναληψιμότητα υπό μεταβαλλόμενο φωτισμό και στροφή, και επομένως χρησιμοποιείται πολύ συχνά στη στερεοσκοπική αντιστοίχιση και ανάκτηση δεδομένων εικόνων. Παρ' όλο που εξακολουθούν να υπάρχουν μειονεκτήματα και περιορισμοί, ο τελεστής Harris εξακολουθεί λοιπόν να αποτελεί σημαντική και θεμελιώδη τεχνική για πολλές εφαρμογές.

FAST (Features from Accelerated Segment Test)

Πρόκειται για ανιχνευτή γωνιών κατά τον οποίο ένας κύκλος με περιφέρεια 16 pixel τοποθετείται περί την κεντρική εικονοψηφίδα. Η τιμή έντασης κάθε εικονοψηφίδας σε αυτόν τον κύκλο συγκρίνεται με την κεντρική εικονοψηφίδα. Μια περιοχή ορίζεται ως ομοιόμορφη, ακμή ή γωνία με βάση το ποσοστό των γειτονικών pixel με εντάσεις παρόμοιες με του κεντρικού. Οι τελεστές FAST θεωρούνται ως μια από τις πιο αποδοτικές μεθόδους εξαγωγής χαρακτηριστικών σημείων. Ωστόσο, είναι πολύ ευαίσθητοι στον θόρυβο.

SIFT (Scale Invariant Feature Transform) features

Ο SIFT (Scale Invariant Feature Transform) αποτελεί ίσως την πιο ισχυρή τεχνική εξαγωγής χαρακτηριστικών. Πρόκειται για έναν ανιχνευτή κατά τον οποίο ορίζεται μια περιοχή εικόνας που διαφέρει από την άμεση γειτονιά της ως προς την ένταση, το χρώμα και την υφή. Ο κύριος λόγος για την επιτυχία του SIFT είναι ότι είναι αμετάβλητος στις στροφές, την κλίμακα, τον φωτισμό και, εν μέρει, στην αλλαγή της οπτικής γωνίας. Ο SIFT όχι μόνο είναι ανιχνευτής χαρακτηριστικών σημείων αλλά και περιγραφέας (descriptor) χαρακτηριστικών βελτιώνοντας την ακρίβεια και την επαναληψιμότητα της αντιστοίχισης των χαρακτηριστικών σημείων σε διαφορετικές εικόνες. Το κύριο μειονέκτημα αυτής της μεθόδου είναι ότι είναι υπολογιστικά δαπανηρή. Τα χαρακτηριστικά SIFT εξάγονται αναλύοντας την Difference of Gaussians (DoG) μεταξύ εικόνων σε διαφορετικές κλίμακες εικόνας. Ένας περιγραφέας χαρακτηριστικών υπολογίζεται με διαίρεση της περιοχής περί το βασικό σημείο σε 4×4 υποπεριοχές. Σε κάθε μια κατασκευάζεται ιστόγραμμα προσανατολισμού 8 διευθύνσεων. Αυτές οι πληροφορίες στη συνέχεια αποθηκεύονται σε διάνυσμα περιγραφής με διάσταση $4 \times 4 \times 8 = 128$. Ο συνδυασμός της θέσης βασικών σημείων, της επιλογής κλίμακας και του περιγραφέα χαρακτηριστικών καθιστά το SIFT μια πολύ ισχυρή τεχνική για την αντιστοίχιση χαρακτηριστικών σημείων σε διαφορετικές εικόνες ακόμα και κάτω από αλλαγές στην κλίμακα, τις στροφές και τον φωτισμό [18].

SURF (Speeded Up Robust Features) descriptor

Πρόκειται για ανιχνευτή και περιγραφέα χαρακτηριστικών σημείων παρόμοιο σε μεγάλο βαθμό με τον SIFT. Η διαφορά τους έγκειται στο ότι ο SIFT χρησιμοποιεί διαφορές φίλτρων Gauss (DoG, Difference of Gaussians) για να προσεγγίσει την Λαπλασιανή φίλτρων Gauss (LoG, Laplacian of Gaussians), ενώ ο SURF χρησιμοποιεί μια βασική προσέγγιση του Εσσιανού πίνακα (Hessian matrix) ο οποίος είναι ένας τετραγωνικός πίνακας που περιλαμβάνει τις συνελίξεις της εικόνας με τα φίλτρα Gauss δευτέρας παραγωγού και περιγράφει την τοπική καμπυλότητα κάθε σημείου της εικόνας (Viola, P. and Jones, M., 2001. Robust real-time object detection. *International journal of computer vision*, 4(34-47), p.4.)[28]. Επίσης πρόκειται για ένα ισχυρό περιγραφέα που έχει παρόμοιες ιδιότητες με τον περιγραφέα SIFT αλλά με λιγότερη υπολογιστική πολυπλοκότητα λόγω της αξιοποίησης των αθροιστικών εικόνων (integral images) και της χρήσης φίλτρων τύπου ‘κουτιού’. Για την υπολογισμό προσανατολισμού του σημείου ενδιαφέροντος χρησιμοποιούνται φίλτρα Haar-wavelet σε μια κυκλική περιοχή γύρω από το σημείο προκειμένου να υπολογιστούν οι τοπικές κλίσεις της εικόνας. Ο SURF είναι ταχύτερος από το SIFT και έχει υψηλότερη υπολογιστική απόδοση. Ωστόσο, ο SIFT ξεπερνά ελαφρώς το SURF όσον αφορά την «ευρωστία» (robustness) διότι είναι αμετάβλητος στις αλλαγές κλίμακας, προσανατολισμού, στις αλλαγές φωτισμού και μερικώς αμετάβλητος στις αφινικές παραμορφώσεις.

BRIEF (Binary Robust Independent Elementary Features) descriptor

Βασίζεται στην αποτελεσματική ταξινόμηση περιοχών των εικόνων (image patches) με βάση έναν σχετικά μικρό αριθμό συγκρίσεων έντασης ανά ζεύγη. Βάσει αυτής της σύγκρισης δημιουργούνται δυαδικές συμβολοσειρές που ορίζουν μια περιοχή που περιβάλλει ένα βασικό σημείο (Calonder, M., Lepetit, V., Strecha, C. and Fua, P., 2010, September. Brief: Binary robust independent elementary features. In *European conference on computer vision* (pp. 778-792). Springer, Berlin, Heidelberg.) [29]. Ο BRIEF είναι μόνο περιγραφέας χαρακτηριστικών και πρέπει να χρησιμοποιείται σε συνδυασμό με έναν ανιχνευτή χαρακτηριστικών όπως οι Harris, FAST, SIFT ή SURF. Η αντιστοίχιση μεταξύ περιγραφικών

δεικτών μπορεί να επιτευχθεί με την εξεύρεση του κοντινότερου γείτονα με βάση την απόσταση Hamming. Ο ανιχνευτής και περιγραφέας χαρακτηριστικών που ονομάζεται ORB, ο οποίος αναλύεται στην ενότητα 6.1, βασίζεται στον ανιχνευτή χαρακτηριστικών FAST και τον περιγραφέα BRIEF.

Σύνδεση δεδομένων (Data Association)

Ως σύνδεση δεδομένων (ΣΔ) ορίζεται η διαδικασία σύνδεσης μίας μέτρησης (ή ενός χαρακτηριστικού) με το αντίστοιχο προηγούμενο εξαχθέν χαρακτηριστικό. Η ΣΔ είναι μια από τις πιο σημαντικές πτυχές της SLAM και της οπτικής οδομετρίας. Στη SLAM, η σωστή ΣΔ μεταξύ των τρεχουσών μετρήσεων και των προηγούμενων αποθηκευμένων χαρακτηριστικών είναι ζωτικής σημασίας για τη SLAM και την παραγωγή ακριβούς χάρτη. Μια λανθασμένη ΣΔ θα κάνει το ρομπότ να 'σκέφτεται' ότι βρίσκεται σε μια θέση όπου δεν βρίσκεται, επομένως θα δημιουργήσει πρόβλημα στη λύση SLAM.

Αλγόριθμος RANSAC (Random Sampling Consensus)

Τα παραπάνω βήματα για αντιστοίχιση χαρακτηριστικών και ΣΔ συνήθως οδηγούν σε ψευδείς αντιστοιχίσεις, κυρίως λόγω θορύβου και πολλών σημείων παρατήρησης. Ο RANSAC αποτελεί το πιο κοινό εργαλείο που χρησιμοποιείται για την εκτίμηση ενός συνόλου παραμέτρων ενός μαθηματικού μοντέλου από ένα σύνολο δεδομένων παρατήρησης που περιέχει χονδροειδή σφάλματα (outliers) (Fischler, M.A. and Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), pp.381-395.) [30]. Η μέθοδος RANSAC, προκειμένου να υλοποιήσει την διαγραφή των outliers αντιστοιχιών μεταξύ δύο πλαισίων, αποτελείται γενικά από τα ακόλουθα βήματα:

- Τυχαία επιλογή του ελάχιστου αριθμού των αντιστοιχισμένων χαρακτηριστικών (π.χ. σημεία σε ένα πλαίσιο και οι αντιστοιχίσεις τους στο επόμενο) που απαιτούνται για την εκτίμηση του μετασχηματισμού, δηλαδή πχ. 3 σημεία για ένα σύστημα 6 βαθμών ελευθερίας εάν χρησιμοποιηθεί μέθοδος εκτίμησης σφαλμάτων βάση επαναπροβολής (3D to 2D).
- Εκτίμηση του μετασχηματισμού (παραμέτροι στροφής και μετατόπισης) με βάση a επιλεγμένα σημεία.
- Εφαρμογή του μετασχηματισμού στα υπόλοιπα σημεία.
- Υπολογισμός της απόστασης D μεταξύ των μετασχηματισμένων σημείων και των αντιστοιχιών τους.
- Ζεύγη με απόσταση D μικρότερη από ένα προκαθορισμένο όριο γίνονται αποδεκτά (inliers).

- Μέτρηση του συνολικού αριθμού inliers που προκύπτουν από τον μετασχηματισμό.
- Επανάληψη των προηγούμενων βημάτων N φορές.
- Ο μετασχηματισμός με τον υψηλότερο αριθμό inliers θεωρείται ότι είναι ο σωστός μετασχηματισμός.

Ο RANSAC είναι ένας μη-ντετερμινιστικός αλγόριθμος υπό την έννοια ότι παράγει ένα λογικό αποτέλεσμα με μια ορισμένη πιθανότητα, με αυτή την πιθανότητα να αυξάνεται όσο επιτρέπονται περισσότερες επαναλήψεις.

ICP (Iterative Closest Point)

Ο ICP είναι μια τεχνική για την αλληλοαναφορά (alignment) 3D σημείων που προκύπτουν από διαφορετικά μέσα, εν προκειμένω εικόνες. Ο ICP αλληλοαναφέρει τα κατά προσέγγιση πλησιέστερα σημεία (αντιστοιχίες που βρέθηκαν με προσέγγιση πλησιέστερης γειτνίασης) σε δύο νέφη σημείων.

Ο ICP έχει αποδειχθεί αποτελεσματικός όταν τα δύο νέφη είναι ήδη σχεδόν ευθυγραμμισμένα. Ως εκ τούτου, μια καλή αρχική εκτίμηση θα συνέβαλε στη παραγωγή καλύτερων αποτελεσμάτων του ICP. Σε καταστάσεις όπου χρησιμοποιείται κακή εκτίμηση, ενδέχεται το αποτέλεσμα είναι κακή εκτίμηση του πραγματικού μετασχηματισμού. Στην RGB-D SLAM, η εκτίμηση μετασχηματισμού RANSAC μπορεί να χρησιμοποιηθεί ως η αρχική εκτίμηση στον αλγόριθμο ICP [18].

3.5 Κριτήρια σύγκρισης αλγορίθμων SLAM

Οι αλγόριθμοι SLAM μπορούν να κατηγοριοποιηθούν ανάλογα με το είδος του αισθητήρα που χρησιμοποιούν, τον τύπο του χάρτη που εξάγουν και ενίοτε από την μαθηματική μέθοδο που εφαρμόζουν (π.χ. Kalman filter ή bundle adjustment).

Ωστόσο υπάρχουν διάφορες περιπτώσεις στη SLAM, οι οποίες δεν εξαρτώνται από το είδος του αισθητήρα ή του εξαγόμενου χάρτη. Αυτό σημαίνει ότι τα αποτελέσματα της SLAM εξαρτώνται από την ποιότητα του αισθητήρα ή από την εκτέλεση του αλγορίθμου, και με αυτό τον τρόπο μπορεί να συμπερανθεί αν ο συγκεκριμένος αλγόριθμος είναι ο καταλληλότερος για τις εκάστοτε ανάγκες, με βάση την απόδοση και στην ποιότητά του. Άρα για να επιλέξει κανείς τον ιδανικότερο αλγόριθμο, ο οποίος θα ικανοποιεί στο μέγιστο τις ανάγκες μας, μπορούν να χρησιμοποιηθούν τα ακόλουθα κριτήρια:

- Η ακρίβεια εντοπισμού θέσης είναι το κύριο κριτήριο για την επιλογή

ενός αλγορίθμου. Συνήθως βασίζεται στη μέση τετραγωνική διαφορά ανάμεσα στις τιμές που έχουν οι θέσεις που έχουν προβλεφθεί από τον αλγόριθμο πχ. ενός ρομπότ και στις αντίστοιχες τιμές που έχουν τα δεδομένα αναφοράς (ground-truth).

- *Ο χρόνος επεξεργασίας των δεδομένων ή ο μέσος όρος χρήσης της CPU (χρόνος σε sec)* είναι ένα άλλο κριτήριο που δείχνει την αποτελεσματικότητα ενός αλγορίθμου.
- Επίσης, μπορεί να τεθεί ως κριτήριο η *χρήση RAM* που απαιτεί / καταναλώνει ένας αλγόριθμος σε σχέση με κάποιον άλλο εφαρμοζόμενοι στο ίδιο dataset. Αυτό είναι πολύ χρήσιμο κυρίως σε κινούμενα ρομπότ με περιορισμένες δυνατότητες όσον αφορά το hardware.
- *FPS (frames per second)* ή ο *χρόνος επεξεργασίας ενός frame της κάμερας* αποτελεί κύριο χαρακτηριστικό για τη Visual SLAM. Αυτό είναι πολύ σημαντικό για τα ιπτάμενα ρομπότ (πχ. drones) καθώς είναι επιθυμητό η επεξεργασία του video να γίνεται σε πραγματικό χρόνο ώστε να γίνονται εύκολα αντιληπτές τυχόν αλλαγές του περιβάλλοντος.
- *Η ποιότητα του αποτελέσματος (χάρτη)* πχ. με σύγκρισή του με τα δεδομένα αναφοράς και εύρεση τυχόν διαφορών του.

Τέλος, υπάρχουν σημαντικές παράμετροι στους αλγόριθμους SLAM, οι οποίες είναι δύσκολο να μετρηθούν όπως είναι η ‘ευρωστία’ (robustness) και ο χρόνος σύγκλισης (convergence).

- *Ευρωστία (Robustness)*: Είναι η ικανότητα του αλγορίθμου να μην ‘χάνει’ την ακρίβεια εντοπισμού της θέσης με την πάροδο του χρόνου ή να μπορεί να εφαρμόζεται σε κάθε είδους περιβάλλον, πχ. εσωτερικό ή εξωτερικό: (indoor, outdoor). Επίσης μπορεί να θεωρήσει κανείς ως ευρωστία την ικανότητα του αλγορίθμου να παρέχει παρόμοια αποτελέσματα σε πολλαπλές εκτελέσεις (runs) στο ίδιο dataset.
- *Σύγκλιση (Convergence)*: Μπορεί να οριστεί ως ο απαραίτητος για τον αλγόριθμο χρόνος για να ελαχιστοποιήσει το σφάλμα εντοπισμού θέσης έπειτα από την τελευταία διαδικασία εντοπισμού θέσης [3].

Στα επόμενα θα παρουσιαστούν παραδείγματα αλγορίθμων SLAM.

Libviso2 (Library for Visual Odometry2): Πρόκειται για μια εφαρμογή που βασίζεται στην ιδέα της στερεο-οδομετρίας. Ως εκ τούτου, χρειάζεται ένα ζευγάρι συγχρονισμένων καμερών και το μόνο που υπολογίζει είναι η κίνηση των καμερών μέσω αντιστοιχιών σημείων (point-to-point) μεταξύ γειτονικών πλαισίων. Αυτό σημαίνει ότι το υπολογιστικό φορτίο θα παραμείνει σταθερό με την πάροδο του χρόνου, καθώς δεν χρειάζεται να

δημιουργηθεί χάρτης του περιβάλλοντος. Όσο η σκηνή έχει ‘σαφή’ χαρακτηριστικά, η libviso2 θα παρέχει λογική ακύτητα δεδομένων, αλλά η εκτιμώμενη θέση δεν θα είναι επαρκώς ακριβής, καθώς μικρά λάθη θα συσσωρεύονται με την πάροδο του χρόνου.

Η πλατφόρμα Libviso2 μπορεί να χρησιμοποιηθεί σε Linux όσο και σε Windows. Στηρίζεται σε C++ και MATLAB για τον υπολογισμό των 6 βαθμών ελευθερίας μιας mono/stereo κάμερας. Η stereo έκδοση βασίζεται στην ελαχιστοποίηση του σφάλματος επαναπροβολής των αραιών κοινών χαρακτηριστικών σημείων (sparse feature matching) και είναι αρκετά γενική (δεν υπάρχουν περιορισμοί στην κίνηση του μοντέλου ή στο σύστημα εκτός του ότι οι εισαγόμενες εικόνες πρέπει να έχουν γνωστές παραμέτρους βαθμονόμησης). Η μονοσκοπική έκδοση (monocular) είναι ακόμα σε πειραματικό στάδιο και χρησιμοποιεί τον αλγόριθμο 8-σημείων για την εκτίμηση του επιπολικού πίνακα. Επιπλέον, υποθέτει ότι η κάμερα κινείται σε γνωστό και σταθερό ύψος από το έδαφος (για την εκτίμηση της κλίμακας). Λόγω των 8 αντιστοιχιών που απαιτούνται για τον συγκεκριμένο αλγόριθμο (8-point) πρέπει να δημιουργούνται πολλά δείγματα RANSAC, κάτι που κάνει τον μονοσκοπικό αλγόριθμο πιο αργό από τον stereo, για τον οποίο 3 αντιστοιχίες είναι ικανοποιητικές για την εκτίμηση των παραμέτρων. Οι πιο σημαντικές διαφορές από την LIBVISO1 είναι οι εξής:

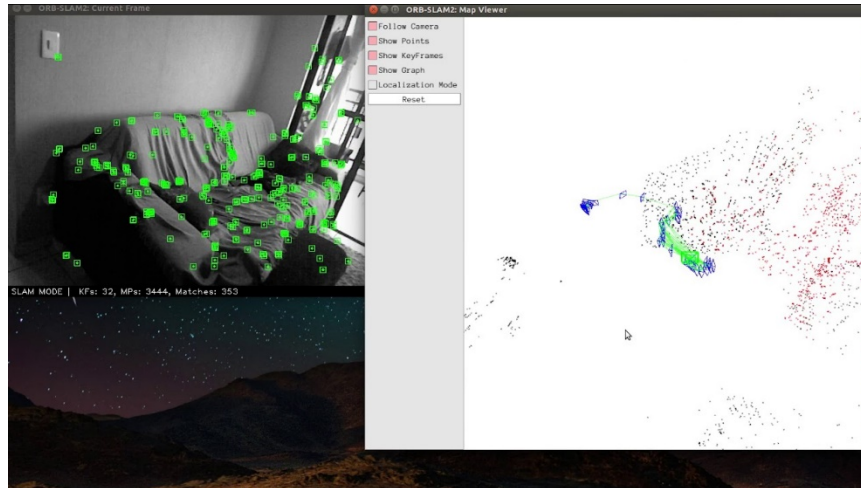
1. Δεν εξαρτάται από εξωτερικές βιβλιοθήκες
2. Μεγαλύτερη πυκνότητα ομόλογων σημείων (μέχρι 15.000 ομόλογα σημεία)
3. Υψηλότερη ταχύτητα στην εύρεση ομόλογων σημείων (1.000 σημεία σε περίπου 35ms)
4. Υποστηρίζει εκτός από stereo και μονοσκοπικό σύστημα
5. Η κλίμακα στο μονοσκοπικό σύστημα υπολογίζεται με την υπόθεση ότι το ύψος της κάμερας από το έδαφος είναι σταθερό
6. Τα ομόλογα σημεία μπορούν να εντοπιστούν σε σύντομο χρονικό διάστημα
7. 3D ανακατασκευή SfM (Structure-from-Motion)



Εικόνα 7.Οπτική οδομετρία με χρήση Libviso2

ORB – SLAM2: Πρόκειται για ένα state-of-the-art λογισμικό SLAM. Καθώς η κάμερα διερευνά τη σκηνή, το λογισμικό δημιουργεί έναν εκτεταμένο χάρτη με τα χαρακτηριστικά που χρησιμοποιούνται για την παρακολούθηση (tracking) και για το βρόχο κλεισίματος (loop closure) αναγνωρίζοντας ποιες περιοχές του χάρτη επισκέφθηκε προηγουμένως). Τα συγκεκριμένα συστήματα SLAM είναι πιο ακριβή από τις προσεγγίσεις οδομετρίας, αλλά υπολογιστικά είναι γενικά πιο απαιτητικά.

Η ORB-SLAM είναι μια ευέλικτη και ακριβής λύση Monocular SLAM. Μπορεί να υπολογίζει σε πραγματικό χρόνο την τροχιά της κάμερας και να πραγματοποιεί μια αραιή 3D ανακατασκευή της σκηνής σε πληθώρα περιβαλλόντων, που κυμαίνονται από μικρές ακολουθίες χειρός (hand-held) μέχρι αυτοκίνητο που κινείται γύρω από αρκετά οικοδομικά τετράγωνα. Είναι σε θέση να πραγματοποιήσει παγκόσμιο εντοπισμό σε πραγματικό χρόνο. Μπορεί να χρησιμοποιηθεί τόσο σε επίπεδες όσο και σε μη επίπεδες σκηνές .



Εικόνα 8.ORB-SLAM2

Rovio (Robust Visual Inertial Odometry): Πρόκειται (χονδρικά) για ένα υβρίδιο μεταξύ των δύο προηγούμενων λύσεων. Μπορεί να θεωρηθεί ως βραχυπρόθεσμη SLAM, διότι ο χάρτης περιορίζεται στα 20-30 πιο πρόσφατα χαρακτηριστικά. Το πιο σημαντικό είναι ότι η Rovio είναι ένα μονοσκοπικό σύστημα και το μόνο που χρειάζεται για να τρέξει είναι μια κάμερα, το οποίο είναι πολύ σημαντικό. Ωστόσο, βασίζεται σε ένα IMU (επιταχυνσιόμετρο + αισθητήρας γυροσκοπίου) για να παρέχει την αίσθηση της κλίμακας στη σκηνή η οποία λείπει από μια μόνο κάμερα. Στην πράξη, αυτό σημαίνει ότι θα μπορούσε ενδεχομένως να προστεθεί ένας απλός αισθητήρας (των 10 ευρώ περίπου) σε ένα ρομπότ.

Με λίγα λόγια το Rovio μπορεί να θεωρηθεί σαν ένας ‘εκτιμητή οπτικής αδρανειακής κατάστασης’ που βασίζεται σε ένα εκτεταμένο Φίλτρο Kalman (EKF). Εκτός από τη γρήγορη αναγνώριση γωνιακών χαρακτηριστικών (corner features), των οποίων η 3D θέση στον χώρο παραμετροποιείται με τη βοήθεια των διανυσμάτων και των αποστάσεων του ρομπότ, πολυεπίπεδα τμήματα εξάγονται από τις εικόνες γύρω από τα χαρακτηριστικά αυτά. Τα τμήματα αυτά παρακολουθούνται (tracking) με χρήση IMU και τα ραδιομετρικά σφάλματα χρησιμοποιούνται για την περαιτέρω βελτίωση. Αναπτύχθηκε ως μια μονοσκοπική εφαρμογή (Visual Inertial Odometry - VIO).

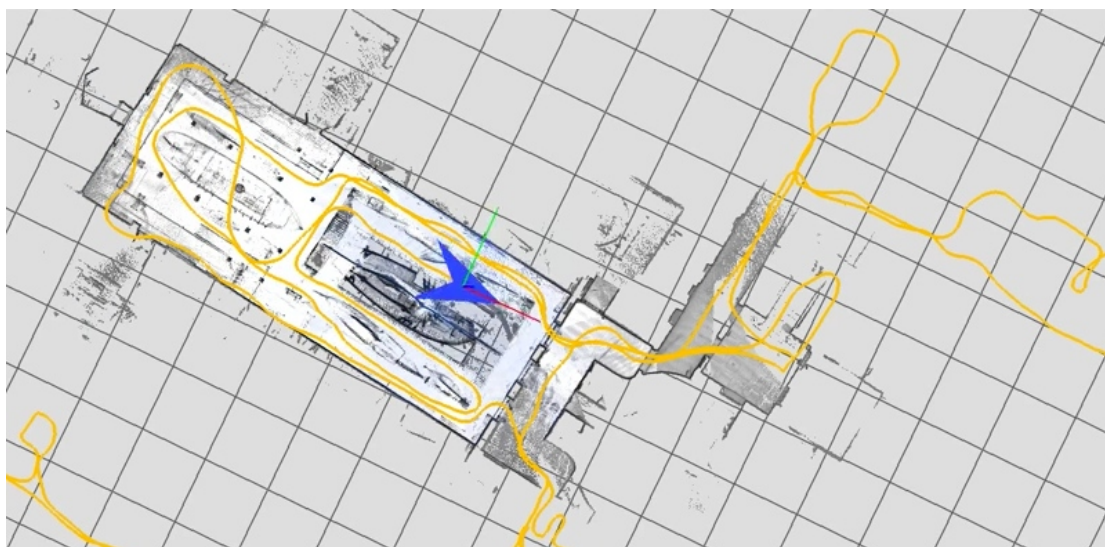
Cartographer: Πρόκειται για έκδοση open-source, την οποία ανακοίνωσε η Google στις βιβλιοθήκες χαρτογράφησης της. Το εργαλείο έχει σχεδιαστεί για να επιτρέπει σε πραγματικό χρόνο τον ταυτόχρονο εντοπισμό και χαρτογράφηση (SLAM), το οποίο έχει την ικανότητα να χτίζει έναν 2D ή 3D χάρτη, ενώ παράλληλα παρακολουθεί την θέση του ρομπότ στο χάρτη.

Το εργαλείο αυτό, η Google, το παρουσίασε για πρώτη φορά το 2014 ως μια τεχνολογία SLAM. Επρόκειτο για ένα εργαλείο χαρτογράφησης που θα επέτρεπε στο χρήστη να κάνει πράγματα όπως το να περπατά μέσα σε ένα κτίριο και να δημιουργεί ένα σχέδιο για τη δομή σε πραγματικό χρόνο. Το σύνθημα της εταιρίας ήταν ‘όσο πιο γρήγορα μπορείτε να περπατήσετε, τόσο πιο πολύ μπορείτε να χαρτογραφήσετε με το Cartographer’. Το συγκεκριμένο εργαλείο θεωρείται σαν μια indoor έκδοση της τεχνολογίας Street View για τη δημιουργία λεπτομερέστατων εικόνων των χώρων μέσα σε ένα κτίριο.

Το Cartographer από τότε έχει ενσωματωθεί στα ROS (Robot Operating Systems), κάτι το οποίο το καθιστά έτοιμο για χρήση σε ρομποτικές πλατφόρμες. Οι μηχανικοί της Google αναφέρουν ότι οι προσπάθειες με το Cartographer, σήμερα σε μεγάλο βαθμό, είναι επικεντρωμένη στην LiDAR SLAM, το οποίο δεν αποτελεί έκπληξη, δεδομένου ότι η τεχνολογία των αυτο-οδηγούμενων οχημάτων εξαρτάται σε μεγάλο βαθμό από τα συστήματα LiDAR.

Επιπλέον, με στόχο την προώθηση της SLAM ως τεχνολογίας, η Google κυκλοφόρησε επίσης για τους προγραμματιστές 2D και 3D δεδομένα LIDAR και IMU που συλλέχθηκαν με το Cartographer. Μέσω της συνεχούς ανάπτυξης, η εταιρεία ελπίζει να προσθέσει υποστήριξη για περισσότερους αισθητήρες και πλατφόρμες, καθώς και νέα χαρακτηριστικά, όπως η ‘δια βίου’ χαρτογράφηση ή ο εντοπισμός σε προϋπάρχοντα χάρτη.

Το Cartographer δεν είναι η μόνη open source βιβλιοθήκη SLAM. Υπάρχουν και άλλες, όπως η hector_slam, η οποία περιέχει πακέτα ROS που σχετίζονται με την εκτέλεση SLAM σε περιβάλλοντα χωρίς γνωστή δομή (unstructured), όπως αυτά που συναντώνται στην ‘αστική διάσωση’ (USAR) και ιδιόκτητες εναλλακτικές λύσεις, όπως η Flyby Media που πρόσφατα αποκτήθηκε από την Apple.



Εικόνα 7. Google Cartographer

OKVIS (Open Keyframe-based Visual-Inertial SLAM): Ο αλγόριθμος OKVIS χρησιμοποιεί μη γραμμική βελτιστοποίηση σε ένα παράθυρο ολίσθησης. Η λειτουργία του είναι αποτέλεσμα του συνδυασμού σταθμισμένων σφαλμάτων επαναπροβολής των σημείων του χώρου και σταθμισμένων αδρανειακών σφαλμάτων. Χρησιμοποιεί έναν πολλών κλιμάκων (multi-scale) ανιχνευτή γωνιών Harris ώστε να βρει χαρακτηριστικά σημεία, και στη συνέχεια εφαρμόζει σε αυτά περιγραφείς BRISK με σκοπό να βρει σύνδεση μεταξύ των καρτέ. Ο OKVIS χρησιμοποιεί την Google's ceres solver (open source βιβλιοθήκη C++ για την μοντελοποίηση και την επίλυση μεγάλων και πολύπλοκων προβλημάτων βελτιστοποίησης) ώστε να επιτύχει τη μη γραμμική βελτιστοποίηση που προαναφέρθηκε. Αξίζει να σημειωθεί πως ο συγκεκριμένος αλγόριθμος δεν έχει αναπτυχθεί για Monocular SLAM, αντίθετα 'δείχνει' την ανώτερη απόδοσή του χρησιμοποιώντας διαμόρφωση stereo.

VINS-Mono (Versatile Monocular Visual-Inertial State Estimator): Πρόκειται για μια μη γραμμική βελτιστοποίηση-παράθυρο εκτίμησης, βασισμένο στην ολίσθηση, το οποίο εντοπίζει ισχυρά χαρακτηριστικά γωνιών, παρόμοια με το OKVIS. Ωστόσο, η VINS-Mono εισάγει πολλά νέα χαρακτηριστικά σε αυτή την κατηγορία εκτίμησης πλαισίου. Οι μετρήσεις IMU είναι προενσωματωμένες (πριν χρησιμοποιηθεί για την βελτιστοποίηση) και προτείνεται μια στενά συνδεδεμένη διαδικασία για επαναπροσανατολισμό (relocalization). Η VINS-Mono διαθέτει επιπλέον χαρακτηριστικά τα οποία μπορούν να εκτελέσουν βελτιστοποίηση του γραφήματος

για 4 βαθμούς ελευθερίας (4DoF) και κλείσιμο βρόχων. Το λογισμικό είναι διαθέσιμο τόσο σε ROS έκδοση συμβατή για PC όσο και σε εφαρμογή iOS για την εκτίμηση κατάστασης σε κινητές συσκευές.

SVO+MSF (Semi – Direct Visual Odometry + Multi-Sensor Fusion):

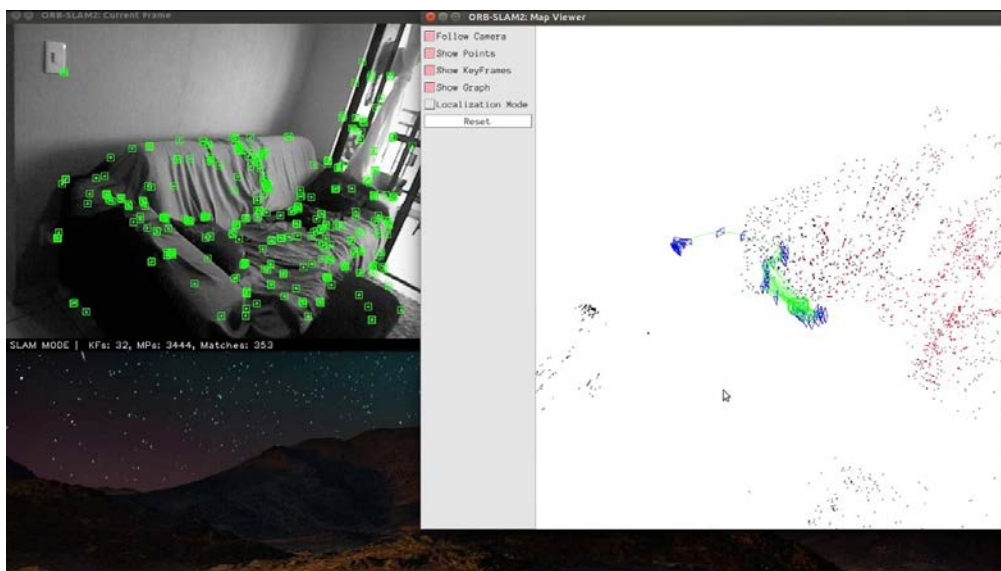
Πρόκειται για ένα EKF (Extended Kalman Filter) για συγχώνευση δεδομένων από διαφορετικούς αισθητήρες με σκοπό την εκτίμηση της κατάστασης. Η SVO (Semi - Direct Visual Odometry) είναι ένας σχετικά ελαφρύς αλγόριθμος οπτικής οδομετρίας, ο οποίος συνδέει εικόνες (alignment) εντοπίζοντας χαρακτηριστικά γωνιών ταχύτατα και ελαχιστοποιεί τα ραδιομετρικά σφάλματα. Πραγματοποιείται αραιή συνταύτιση, η οποία στη συνέχεια βοηθά στην βελτιστοποίηση της δομής της σκηνής (scene structure), ελαχιστοποιώντας τα σφάλματα επαναπροβολής των ομόλογων σημείων με μη γραμμική επίλυση της μεθόδου των ελαχίστων τετραγώνων. Η κίνηση του αντικειμένου εκτιμάται από το σύστημα SVO (vision - only) και στη συνέχεια παρέχεται στο σύστημα MSF όπου συνδυάζεται με τα δεδομένα του IMU. Επίσης η κλίμακα πρέπει να είναι σωστή, το οποίο επιτυγχάνεται με χειροκίνητη αρχικοποίηση ή με χρήση αισθητήρα απόστασης laser. Τόσο ο αλγόριθμος SVO όσο και ο MSF είναι δημόσια διαθέσιμοι στο ευρύ κοινό και επικοινωνούν μέσω της πλατφόρμας ROS. Αυτό το σύστημα αναφέρεται συχνά ως svomsf.

4. ORB SLAM 2 και συνοπτική επεξήγηση λειτουργίας αλγορίθμου

4.1 ORB SLAM2

Πρόκειται για ένα state-of-the-art λογισμικό SLAM. Καθώς η κάμερα διερευνά τη σκηνή, το λογισμικό δημιουργεί έναν εκτεταμένο χάρτη με τα χαρακτηριστικά που χρησιμοποιούνται για την παρακολούθηση (tracking) και για το βρόχο κλεισίματος (loop closure) αναγνωρίζοντας ποιες περιοχές του χάρτη επισκέφθηκε προηγουμένως. Τα συγκεκριμένα συστήματα SLAM είναι πιο ακριβή από τις προσεγγίσεις οδομετρίας, αλλά υπολογιστικά είναι γενικά πιο απαιτητικά.

Η ORB-SLAM2 είναι μια ευέλικτη και ακριβής λύση που μπορεί να λειτουργήσει μονοεικονικά, διεικονικά ή με μηχανές βάθους (RGB-D). Μπορεί να υπολογίζει σε πραγματικό χρόνο την τροχιά της κάμερας και να πραγματοποιεί μια αραιή 3D ανακατασκευή της σκηνής σε πληθώρα περιβαλλόντων (από μικρές ακολουθίες χειρός μέχρι εκείνες από αυτοκίνητο κινούμενο σε αρκετά οικοδομικά τετράγωνα). Είναι σε θέση να πραγματοποιήσει παγκόσμιο εντοπισμό σε πραγματικό χρόνο. Μπορεί να χρησιμοποιηθεί τόσο σε επίπεδες όσο και σε μη επίπεδες σκηνές.



Εικόνα 8. Visual SLAM με χρήση αλγορίθμου ORB SLAM2

Το back-end (επίπεδο λογισμικού όπου υπάρχουν τα δεδομένα τα οποία έχουν καταγραφεί), το οποίο βασίζεται στην συνόρθωση δέσμης (bundle adjustment), τόσο για μονοσκοπική όσο και για στερεοσκοπική καταγραφή της σκηνής, επιτρέπει την ακριβή εκτίμηση της τροχιάς με μετρητική

κλίμακα. Επίσης δημιουργείται μια αραιή 3D ανακατασκευή της σκηνής. Για αυτό το λόγο, το συγκεκριμένο λογισμικό είναι υπολογιστικά ‘ελαφρύ’ και μπορεί κάποιος να το χρησιμοποιήσει με τη βοήθεια όχι ιδιαίτερα δυνατών υπολογιστικών συστημάτων.

4.2 Τρόπος λειτουργίας ORB SLAM2

Η ORB SLAM2, τόσο για stereo όσο και για κάμερες RGB-D, στηρίζεται στην λογική της μονοσκοπικής ORB SLAM. Το σύστημα βασίζεται στην παράλληλη λειτουργία τριών κύριων «τομέων» (threads).

- *Tracking* ➡ ανίχνευση με σκοπό τον εντοπισμό της κάμερας σε κάθε εικόνα με την εύρεση σημείων ομόλογων με τον χάρτη (local map) και χρήση συνόρθωσης δέσμης (motion-only bundle adjustment) με σκοπό την ελαχιστοποίηση των σφαλμάτων επαναπροβολής.
- *Local mapping* ➡ Διαχείριση του χάρτη και βελτίωσή του με χρήση local bundle adjustment.
- *Loop closing* ➡ Ανίχνευση σημείων τα οποία έχει ‘ξανα-επισκεφτεί’ η κάμερα και διόρθωση της ‘ολίσθησης’ (drift) – τα σφάλματα ολίσθησης είναι σωρευτικά – με την πραγματοποίηση pose graph optimization (θέση και προσανατολισμός – στροφές της κάμερας). Ακολουθεί τελικά, ένας τέταρτος ‘τομέας’, η πλήρης συνόρθωση δέσμης, με σκοπό τη δημιουργία της βέλτιστης δομής του χώρου και της κίνησης της κάμερας.

Η ORB SLAM2 έχει ενσωματωμένο ένα Place Recognition module το οποίο βασίζεται στη βιβλιοθήκη DBoW2 (η οποία μετατρέπει τις εικόνες σε ‘λέξεις’ με σκοπό την προσεγγιστική εύρεση ‘γειτόνων’ στο επίπεδο της εικόνας και μετατροπή τους σε ‘λεξιλόγιο’), το οποίο βοηθά στον ‘επανεντοπισμό’ (relocalization) σε περίπτωση όπου υπάρχει σφάλμα εντοπισμού. Με τη βοήθεια του covisibility graph (γράφημα στο οποίο κάθε κόμβος αφορά τη θέση του κάθε σημείου και κάθε ακμή την ‘οπτική’ ένωση μεταξύ των σημείων) ενώνονται ανά δύο τα keyframes, τα οποία στη συνέχεια συνδέονται με τη μορφή δέντρου. Με αυτό τον τρόπο μπορεί να εργάζεται κανείς σε εκτεταμένα περιβάλλοντα εφόσον ο εντοπισμός (tracking) και το local mapping πραγματοποιούνται τοπικά (στα keyframes).

Το σύστημα χρησιμοποιεί την εξαγωγή χαρακτηριστικών ORB (Oriented FAST and Rotated BRIEF) για τις λειτουργίες tracking, mapping και place recognition. Τα συγκεκριμένα χαρακτηριστικά βασίζονται στους FAST

ανιχνευτές χαρακτηριστικών σημείων και στους οπτικούς περιγραφείς BRIEF. Δεν παρουσιάζουν σφάλματα στην κλίμακα και τις στροφές, και είναι αμετάβλητα στις λειτουργίες auto-gain και auto-exposure της κάμερας καθώς και στις αλλαγές φωτεινότητας. Επίσης εξάγονται και συνδέονται γρήγορα (γρηγορότερα από τα SIFT και SURF), κάτι που τα καθιστά χρήσιμα σε εφαρμογές πραγματικού χρόνου.

4.3 Close stereo keypoints, Far stereo keypoints, Monocular keypoints

Ως *stereo keypoints* ορίζονται τα keypoints τα οποία περιγράφονται από 3 συντεταγμένες $X_s = (u_L, v_L, u_R)$, όπου u_L, v_L οι συντεταγμένες της αριστερής εικόνας και u_R η οριζόντια συντεταγμένη της δεξιάς εικόνας. Στις στερεοσκοπικές κάμερες γίνεται εξαγωγή χαρακτηριστικών ORB και στις δύο εικόνες. Για κάθε ORB της αριστερής εικόνας αναζητείται το ομόλογό του στη δεξιά. Αυτό μπορεί να πραγματοποιηθεί με μεγάλη αποτελεσματικότητα με την θεώρηση ότι οι εικόνες stereo είναι ‘κανονικοποιημένες’, ότι δηλαδή οι επιπολικές ευθείες είναι οριζόντιες. Για τις κάμερες RGB-D πραγματοποιείται εξαγωγή χαρακτηριστικών ORB στην εικόνα RGB, και για κάθε χαρακτηριστικό με συντεταγμένες u_L, v_L μετατρέπεται η τιμή βάθους d σε εικονική συντεταγμένη (δεξιά). Άρα:

$$u_R = u_L - (f_x * b) / d$$

όπου f_x → εστιακή απόσταση κατά x

b → baseline (μήκος βάσης) ανάμεσα στον προβολέα (structured light projector, πχ. Davis SLS-2) και την κάμερα υπερύθρων. Ενημερωτικά, στο Kinect το b είναι περίπου 8cm. Κάποιο σημείο (keypoint) θεωρείται ως *close stereo keypoint* αν το βάθος του είναι μικρότερο του 40πλάσιου του μήκους της RGB-D/stereo βάσης (b). Στην αντίθετη περίπτωση θεωρείται *far stereo keypoint*. Ο τριγωνισμός των *close stereo keypoints* μπορεί να πραγματοποιηθεί με ακρίβεια από κάθε frame και μπορεί να δώσει πληροφορίες κλίμακας, μετάθεσης και στροφής. Αντίθετα, τα *far stereo keypoints* παρέχουν ακριβείς πληροφορίες για τις στροφές αλλά λιγότερο ακριβείς για την μετάθεση και την κλίμακα. Ο σωστός τριγωνισμός των σημείων αυτών πραγματοποιείται όταν ‘εντοπίζονται’ από πολλές οπτικές γωνίες (multiple views).

Τα *monocular keypoints* προσδιορίζονται από δύο συντεταγμένες στην αριστερή εικόνα $X_m = (u_L, v_L)$ και αναφέρονται σε όλα τα χαρακτηριστικά ORB για τα οποία δεν υπάρχει διεικονική αντιστοίχιση (matching) ή έχουν μη έγκυρη τιμή βάθους στην περίπτωση των καμερών RGB-D. Αυτά τα

σημεία τριγωνίζονται μόνο από πολλαπλές εικόνες και δεν παρέχουν καμία πληροφορία κλίμακας παρά μόνο εκτίμηση μετάθεσης και στροφής.

4.4 System bootstrapping (Εκκίνηση του συστήματος)

Ένα από τα πλεονεκτήματα της χρήσης καμερών stereo ή RGB-D είναι ότι κατά την εκκίνηση του συστήματος δημιουργείται keyframe από το 1^ο πλαίσιο, τίθεται η θέση και ο προσανατολισμός του (pose) ως αρχική κατάσταση (κατάσταση 0) και δημιουργείται κατά αυτόν τον τρόπο ένας αρχικός χάρτης από όλα τα stereo keyframes.

4.5 Συνόρθωση δέσμης σε μονοεικονικά και διεικονικά συστήματα (με περιορισμούς)

Το ORB SLAM2 εκτελεί συνόρθωση δέσμης για να βελτιστοποιήσει την θέση και τον προσανατολισμό της κάμερας (pose) – motion-only Bundle Adjustment, βελτιστοποίηση των keyframes (local BA) και μετά την περάτωση βρόχων (loop closing) βελτιστοποίηση όλων των keyframes και των σημείων (full BA).

Motion-only BA ➡ Χρησιμοποιείται για τη βελτιστοποίηση της θέσης της κάμερας t και του προσανατολισμού της R με την ελαχιστοποίηση του σφάλματος επαναπροβολής (reprojection error) ανάμεσα στα συνταυτισμένα 3D σημεία σε παγκόσμιες συντεταγμένες και σε keypoints, είτε αναφέρεται κανείς σε μονοσκοπική είτε σε στερεοσκοπική:

$$\{R, t\} = \underset{R, t}{\operatorname{argmin}} \sum_{i \in \mathcal{X}} \rho \left(\left\| x_{(\cdot)}^i - \pi_{(\cdot)}(RX^i + t) \right\|_{\Sigma}^2 \right)$$

Για την επίλυση του προβλήματος της βελτιστοποίησης, γίνεται χρήση της παραπάνω εξίσωσης, όπου:

- ρ ➡ Huber cost function
- Σ ➡ Covariance matrix (δίνει την κλίμακα των keypoints)
- $\pi(\cdot)$, π_m (για monocular), π_s (για stereo) ορίζονται ως προβολικοί συντελεστές

$$\pi_m \left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \end{bmatrix}, \pi_s \left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right) = \begin{bmatrix} f_x \frac{X}{Z} + c_x \\ f_y \frac{Y}{Z} + c_y \\ f_x \frac{X-b}{Z} + c_x \end{bmatrix}$$

όπου

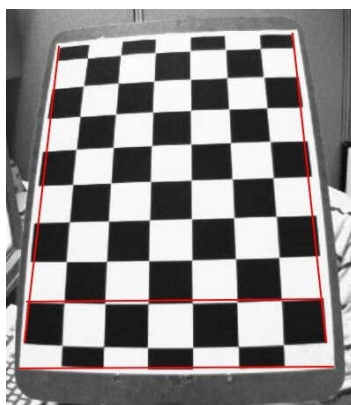
- f_x, f_y → εστιακές αποστάσεις
- c_x, c_y → συντεταγμένες του πρωτεύοντος σημείου
- b → το μήκος της βάσης (baseline)

Όλοι οι παραπάνω συντελεστές είναι γνωστοί από την βαθμονόμηση της κάμερας (calibration).

Camera calibration → Η βαθμονόμηση κάμερας και η εύρεση της παραμόρφωσης της εικόνας αφορά το τι συμβαίνει όταν μια κάμερα ‘κοιτάζει’ τα τρισδιάστατα αντικείμενα στον πραγματικό κόσμο και τα μετατρέπει σε εικόνα 2D. Αυτός ο μετασχηματισμός δεν είναι βέβαια τέλειος. Σε παραμορφωμένες εικόνες μπορεί να διακριθεί ότι οι ακμές τους είναι καμπυλωμένες.

Ο μετασχηματισμός από τα σημεία 3D αντικειμένων P των X, Y και Z σε x και y γίνεται από ένα πίνακα μετασχηματισμού που ονομάζεται πίνακας μηχανής (camera matrix C), ο οποίος χρησιμοποιείται για τη βαθμονόμηση της κάμερας. Ωστόσο, οι πραγματικές κάμερες δεν χρησιμοποιούν σημειακές οπές (pinholes) αλλά φακούς (για να συγκεντρώσουν πολλαπλές ακτίνες φωτός) με σκοπό τη διαμόρφωση εικόνων. Αλλά οι φακοί επίσης μπορούν να προκαλέσουν παραμόρφωση. Δύο σημαντικές παραμορφώσεις είναι η ακτινική διαστρόφη (radial distortion) και η εφαπτομενική διαστρόφη (tangential distortion).

Λόγω της ακτινικής παραμόρφωσης, οι ευθείες γραμμές θα φαίνονται καμπύλες. Η επίδρασή της είναι περισσότερο εμφανής προς τα άκρα της εικόνας. Η εφαπτομενική παραμόρφωση συμβαίνει κυρίως επειδή ο φακός δεν είναι παράλληλος με το επίπεδο απεικόνισης. Παράδειγμα ακτινικής παραμόρφωσης (Εικ. 11): μια από τις δύο άκρες ενός σκακιού σημειώνεται με κόκκινες γραμμές. Αλλά μπορεί να διακριθεί ότι το περίγραμμα δεν είναι ευθύ και δεν ταιριάζει με την κόκκινη γραμμή.



Εικόνα 9. Ακτινική παραμόρφωση

Η παραμόρφωση αυτή εκφράζεται ως εξής:

$$x_{\text{corrected}} = x * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$

$$y_{\text{corrected}} = y * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$

Η εφαπτομενική παραμόρφωση έχει αποτέλεσμα ορισμένες περιοχές στην εικόνα να φαίνονται πιο κοντά από το αναμενόμενο. Εκφράζεται ως εξής:

$$x_{\text{corrected}} = x + [2 * p_1 * x * y + p_2 * (r^2 + 2 * x^2)]$$

$$y_{\text{corrected}} = y + [p_1 * (r^2 + 2 * y^2) + 2 * p_2 * x * y]$$

Εν ολίγοις, πρέπει να βρεθούν οι πέντε παράμετροι, γνωστές ως συντελεστές παραμόρφωσης ή διαστρωφής (distortion coefficients), δηλαδή οι:

$$(k_1 \ k_2 \ k_3 \ p_1 \ p_2)$$

Εκτός από αυτό, πρέπει βέβαια να υπολογιστούν κάποιες άλλες πληροφορίες, όπως οι εσωτερικές και εξωτερικές παράμετροι μιας φωτογραφικής μηχανής (εσωτερικός και εξωτερικός προσανατολισμός). Οι εσωτερικές παράμετροι είναι συγκεκριμένες για μια κάμερα. Περιλαμβάνουν πληροφορίες όπως η εστιακή απόσταση (focal length - f_x , f_y) και η θέση του πρωτεύοντος σημείου (c_x , c_y). Εξαρτώνται μόνο από τη φωτογραφική μηχανή, έτσι όταν υπολογιστεί ο 3x3 πίνακας μηχανής μπορεί πλέον να αποθηκευτεί:

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

Οι εξωτερικές παράμετροι αντιστοιχούν σε διανύσματα στροφής και μετάθεσης που συνδέουν ένα τρισδιάστατο σημείο με ένα σύστημα εικονοσυντεταγμένων.

Για στερεοσκοπικές εφαρμογές, αυτές οι παραμορφώσεις πρέπει να διορθωθούν πρώτα. Για να βρεθούν όλες αυτές οι παράμετροι, πρέπει πχ. να ληφθούν ορισμένες δειγματοληπτικές εικόνες ενός καλά ορισμένου σχηματισμού (όπως η σκακιέρα). Εντοπίζονται κάποια συγκεκριμένα σημεία σε αυτό (γωνίες σκακιέρας, των οποίων οι συντεταγμένες στον πραγματικό κόσμο και στην εικόνα είναι γνωστές). Με αυτά τα δεδομένα, επιλύεται το μαθηματικό πρόβλημα για να βαθμονομηθεί η μηχανή [11].

Local BA ➡ Χρησιμοποιείται για τη βελτιστοποίηση ευδιάκριτων key-frames K_L καθώς και των σημείων P_L τα οποία βρίσκονται σε αυτά.

Full BA (ειδική περίπτωση Local BA) ➡ Χρησιμοποιείται για τη βελτιστοποίηση όλων των keyframes και των σημείων του χάρτη εκτός του αρχικού keyframe.

Keyframe insertion ➡ Η ORB SLAM2 ακολουθεί την λογική της μονοσκοπικής ORB SLAM όπου εισάγονται keyframes συχνά και στη συνέχεια απαλείφονται τα περιττά. Η διάκριση σε *close stereo* και *far stereo points* επιτρέπει να δημιουργηθεί μια νέα συνθήκη στην εισαγωγή των keyframes, η οποία είναι πολύ χρήσιμη όταν μεγάλο μέρος της σκηνής είναι μακριά από τον αισθητήρα. Σε αυτές τις περιπτώσεις πρέπει να υπάρχει ικανοποιητικός αριθμός από *close stereo points* ώστε να εκτιμηθεί με ακρίβεια η μετάθεση [4].

4.6 Datasets για έλεγχο του ORB SLAM2

Για την πρακτική χρήση της ORB SLAM2 λύσης θα γίνει εφαρμογή σε datasets τα οποία βρίσκονται στο Διαδίκτυο [6]. Συγκεκριμένα το [fr1/desk](#), στο tum dataset, κατηγορία handheld SLAM [7]. Επίσης θα χρησιμοποιηθεί laptop με CPU Intel i7 (7th Gen), RAM 8Gb και GPU Nvidia Geforce GTX 1050 (4 Gb). Το λογισμικό εφαρμόστηκε σε περιβάλλον Linux, το οποίο εγκαταστάθηκε εικονικά στον υπολογιστή (με τη βοήθεια του προγράμματος VMWare).

- **KITTI dataset**

Περιέχει στερεοσκοπικές ακολουθίες, η εγγραφή των οποίων έχει πραγματοποιηθεί από αυτοκίνητο σε αστικούς (urban) και υπεραστικούς δρόμους (highways). Ο στερεοσκοπικός αισθητήρας έχει μήκος βάσης (baseline) περίπου 54cm ($b = 54\text{cm}$), συχνότητα λειτουργίας 10Hz και ανάλυση 1240×376 pixels έπειτα από την βαθμονόμηση. Οι ακολουθίες 00, 05, 06, 07, 09 περιέχουν βρόχους (loops).

- **EuRoC Dataset**

Περιέχει ακολουθίες stereo, οι οποίες έχουν εγγραφεί από drone (MAV – Micro aerial vehicle), το οποίο έχει πραγματοποιήσει πτήση σε δύο διαφορετικά δωμάτια και σε βιομηχανική περιοχή. Ο στερεοσκοπικός αισθητήρας έχει μήκος βάσης περίπου 11cm ($b = 11\text{cm}$) και παρέχει εικόνες WVGA σε συχνότητα λειτουργίας 20Hz. Οι ακολουθίες κατηγοριοποιούνται σε *easy*, *medium* και *difficult* ανάλογα με την *ταχύτητα του drone*, τον *φωτισμό (illumination)* και την *υφή της σκηνής (scene texture)*. Σε όλες τις διαθέσιμες ακολουθίες το drone περνά από τα ίδια σημεία της κάθε σκηνής.

- **TUM RGB-D Dataset**

Το συγκεκριμένο dataset περιέχει ακολουθίες εσωτερικού χώρου (indoor) από αισθητήρες RGB-D για την εκτίμηση της ακρίβειας ανακατασκευής αντικειμένων (object reconstruction) και μεθόδων οδομετρίας / SLAM υπό διαφορετικές συνθήκες φωτισμού, υψής και δομής του περιβάλλοντος.

ΣΥΜΠΕΡΑΣΜΑ

Η ORB SLAM2 αποτελεί μια ολοκληρωμένη λύση για συστήματα SLAM με χρήση αισθητήρων μονοσκοπικών, στερεοσκοπικών και RGB-D. Είναι ικανή να πραγματοποιεί περάτωση βρόχων, επανεντοπισμό και επαναχρησιμοποίηση του χάρτη σε πραγματικό χρόνο σε όχι και τόσο ισχυρά υπολογιστικά συστήματα. Μπορεί να χρησιμοποιηθεί σε διαφορετικά περιβάλλοντα (από περιορισμένους χώρους μέχρι μεγάλες βιομηχανικές ζώνες και αυτοκινητόδρομους). Επίσης αυτή η λύση είναι πολύ χρήσιμη σε εφαρμογές εικονικής πραγματικότητας και επαυξημένης πραγματικότητας.

5. UcoSLAM (Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers)

5.1 UcoSLAM

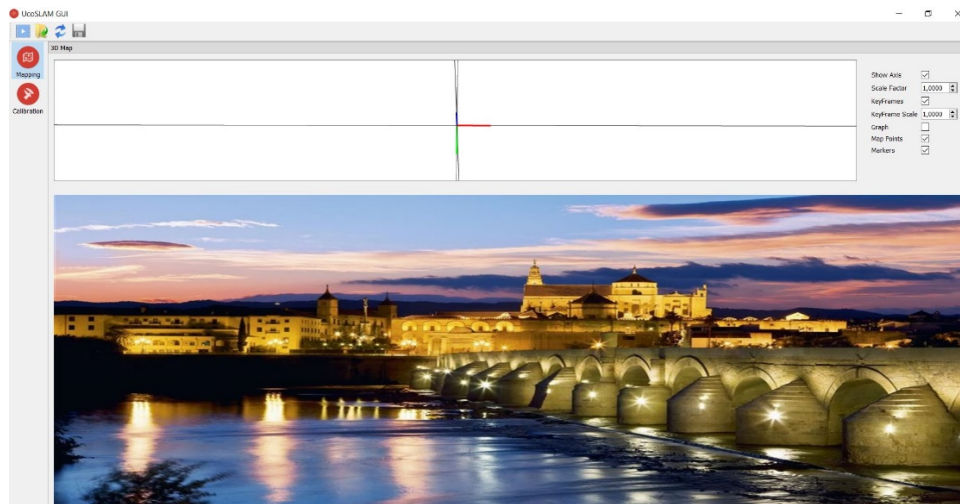
Πολλοί αλγόριθμοι SLAM στηρίζονται στην αναγνώριση φυσικών “στόχων” (keypoints). Το αρνητικό όμως των “στόχων” αυτών είναι ότι δεν είναι κατ’ ανάγκην σταθεροί με την πάροδο του χρόνου και σε περιπτώσεις εντοπισμού και χαρτογράφησης σε εσωτερικούς χώρους δεν προσφέρουν καλή ακρίβεια.

Για να αντιμετωπιστεί αυτό χρησιμοποιούνται τεχνητοί στόχοι, με σκοπό τον εντοπισμό και την χαρτογράφηση σε ποικιλία περιβαλλόντων.

Η UcoSLAM στηρίζεται στον εντοπισμό και την χαρτογράφηση με χρήση τόσο φυσικών χαρακτηριστικών σημείων όσο και τεχνητών στόχων (π.χ επίπεδοι και τετράγωνοι στόχοι (markers) που επιτρέπουν μοναδική ταυτοποίηση με τη μορφή δισδιάστατων barcodes, LEDs, retroreflective spheres, κ.λπ.). Δημοφιλέστεροι είναι οι επίπεδοι τετράγωνοι στόχοι. Περιλαμβάνουν ένα μαύρο εξωτερικό όριο και έναν εσωτερικό (συνήθως δυαδικό – binary) κώδικα ο οποίος κάνει τον κάθε στόχο μοναδικό. Πλεονέκτημά τους είναι ότι παρέχουν τέσσερις γωνίες (correspondence points), οι οποίες επαρκούν ώστε να πραγματοποιηθεί η εύρεση του προσανατολισμού και των στροφών της κάμερας (camera pose estimation).

Λαμβάνοντας υπ’ όψιν ότι η συγκεκριμένη λύση SLAM χρησιμοποιεί και τα δύο είδη χαρακτηριστικών σημείων/στόχων που προαναφέρθηκαν, μπορεί να βγάλει κανείς το συμπέρασμα ότι:

1. Η κλίμακα του χάρτη υπολογίζεται και προσδιορίζεται αυτόματα έπειτα από την ανίχνευση ενός στόχου στο περιβάλλον.
2. Η λύση UcoSLAM μπορεί να εντοπίσει φυσικά χαρακτηριστικά σημεία ή τεχνητούς στόχους ή συνδυασμό τους με αποτέλεσμα να μπορεί να εφαρμοστεί σε περιβάλλοντα τα οποία δεν περιέχουν χαρακτηριστικά σημεία.
3. Οι τεχνητοί στόχοι (markers) βοηθούν στην επίλυση του προβλήματος επανεντοπισμού (relocalization) εφόσον απαλείφουν τις οπτικές ασάφειες σε περιβάλλοντα στα οποία τα περισσότερα σημεία επαναλαμβάνονται. Επίσης, σε εκτεταμένες σκηνές, οι στόχοι μπορούν να τοποθετηθούν σε στρατηγικά σημεία με σκοπό την υπερκάλυψή τους και τη μείωση της ολίσθησης. Επιπλέον, χαρακτηριστικό της UcoSLAM είναι ότι υπάρχει η δυνατότητα αποθήκευσης του παραγόμενου χάρτη και μελλοντικής χρήσης του (Εικ. 13).



Εικόνα 10. UcoSLAM GUI

5.2 Τρόπος λειτουργίας UcoSLAM

Αρχικά, ο χάρτης δημιουργείται με τη βοήθεια των στόχων και των keypoints με την σωστή κλίμακα. Έπειτα, οι στόχοι μπορούν να χρησιμοποιηθούν ως σημεία αναφοράς σε βάθος χρόνου, σε αντίθεση με τα keypoints, τα οποία μπορούν να μεταβληθούν δραστικά ακόμα και σε σύντομο χρονικό διάστημα. Στη συνέχεια, με την χρήση και των δύο ειδών χαρακτηριστικών σημείων (φυσικά και τεχνητά) μπορεί να πραγματοποιηθεί εντοπισμός και χαρτογράφηση σε πιο απαιτητικά περιβάλλοντα, όπως διάδρομοι, γραφεία κτιρίων κ.λπ. Τέλος, δεν είναι απαραίτητο να γίνει χρήση των στόχων, αντίθετα μπορούν απλώς να βελτιώσουν τον εντοπισμό [7].

6. Εφαρμογή ORB SLAM2 σε indoor σενάριο

6.1 Επεξεργασία βίντεο με χρήση ORB SLAM2, ανάλυση λειτουργίας αλγορίθμου

Για τις ανάγκες της Διπλωματικής Εργασίας (λήψη βίντεο, μετατροπή σε ακολουθία εικόνων και επεξεργασία τους) χρησιμοποιήθηκε το κινητό τηλέφωνο Samsung Galaxy A5 (2016) σε σενάριο εσωτερικού χώρου με μονοσκοπική κάμερα (άγνωστη κλίμακα). Για τη μετατροπή του βίντεο (.mp4) σε εικόνες (.png) χρησιμοποιήθηκε κώδικας από το Internet σε γλώσσα Python. Με την εκτέλεση του κώδικα χωρίστηκε το βίντεο σε πλαίσια (πχ. 20), ακολούθησε δε μετατροπή των χρονικών πληροφοριών τους σε timestamp και τελικά αποθήκευση σε αρχείο .txt. Οι εικόνες είναι διαστάσεων 320×240 μορφής .png. Επίσης δημιουργήθηκε κώδικας για την περίπτωση χρήσης του ORB SLAM2 σε εικόνες, δηλαδή μετατροπή των χρονικών πληροφοριών τους σε timestamp και αποθήκευση σε αρχείο .txt

Οι παράμετροι για τον προσανατολισμό των εικόνων εξήχθησαν με την χρήση της εφαρμογής Camera Calibrator του λογισμικού MATLAB (παράμετροι: f_x , f_y , c_x , c_y , k_1 , k_2 , k_3 , p_1 , p_2).

- f_x , f_y ➡ Εστιακή απόσταση (Focal length)
- c_x , c_y ➡ Συντεταγμένες πρωτεύοντος σημείου (Principal point)
- k_1 , k_2 , k_3 ➡ Ακτινική διαστρόφη (Radial distortion φακού)
- p_1 , p_2 ➡ Εφαπτομενική διαστρόφη (Tangential distortion φακού)



Εικόνα 11. Τοποθέτηση στόχων για εκτέλεση αλγορίθμου σε σενάριο εσωτερικού χώρου

Οι στόχοι είναι Aruco DICT_4×4, διαστάσεων 8×11 με μέγεθος τετραγώνου 15mm, και τοποθετήθηκαν με διαφορετικό προσανατολισμό σε διαφορετικά σημεία για να αναγνωριστούν από το λογισμικό σαν διαφορετικοί ο ένας από τον άλλο (Εικ. 14). Επίσης, μέσα στον φάκελο ORB SLAM2/Master/Examples/Monocular δημιουργήθηκε αρχείο Settings.yaml στο οποίο περιέχονται όλες οι παράμετροι «βαθμονόμησης» της κάμερας που εξήχθησαν από το MATLAB καθώς και ο αριθμός των πλαισίων.

Για την εκτέλεση του ORB SLAM2¹ δηλώθηκε το «μονοπάτι» όπου βρίσκεται ο φάκελος ORB SLAM2 Master, το αρχείο ORBvoc.txt (το vocabulary του ORB), το αρχείο ρυθμίσεων που δημιουργήθηκε και ο φάκελος στον οποίο περιέχονται οι εικόνες. Όλα τα keyframes, τα οποία δημιούργησαν τον χάρτη αποθηκεύονται σε αρχείο .txt (Keyframe trajectory.txt). Το νέφος σημείων αποθηκεύεται σε αρχείο μορφής .pcd.

Στο βίντεο *video_orbslam2_phone* παρατηρείται η επεξεργασία βίντεο (ακολουθία εικόνων) με το ORB SLAM2. Ο αισθητήρας έχει οριστεί σε Monocular. Στη συνέχεια διαβάζονται οι παράμετροι προσανατολισμού της κάμερας που ορίστηκαν στο Settings_file.yaml καθώς και τα πλαίσια. Ακολουθούν οι παράμετροι για τα χαρακτηριστικά ORB. Χρησιμοποιήθηκαν οι προεπιλεγμένες παράμετροι.

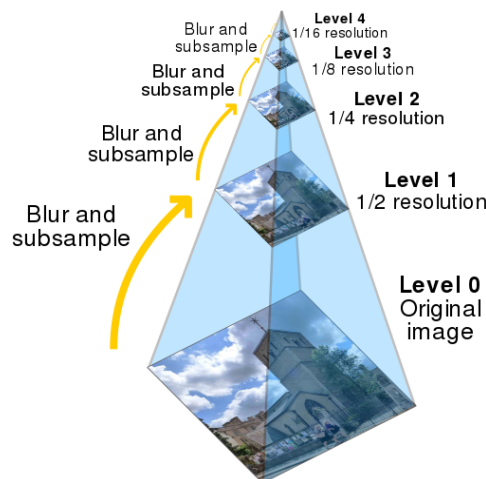
6.1.1 ORB Extractor

- *Number of features: 1000* ➔ Αριθμός χαρακτηριστικών ORB ανά εικόνα
- *Scale levels: 8* ➔ Αριθμός επιπέδων (levels) στην πυραμίδα
- *Scale factor: 1.2* ➔ Συντελεστής κλίμακας μεταξύ των επιπέδων
- *Initial FAST threshold: 20* ➔ Η εικόνα χωρίζεται σε πλέγμα και σε κάθε κελί εξάγονται FAST χαρακτηριστικά σημεία.
- *Minimum FAST threshold: 7* ➔ Σε περίπτωση όπου δεν υπάρχουν γωνίες χρησιμοποιείται αυτή η τιμή. Επίσης, αν οι εικόνες έχουν χαμηλή αντίθεση μπορούν να μειωθούν αυτές οι τιμές.

1. Αρχικά το ORB SLAM2 δημιουργεί μία πυραμίδα εικόνων (Εικόνα 15). Στην πυραμίδα αυτή η αρχική εικόνα χωρίζεται σε εξομαλυσμένες υπο-εικόνες, οι οποίες προστίθενται σαν επίπεδα πάνω από την αρχική. Αυτό σημαίνει ότι οι υπο-εικόνες είναι το αποτέλεσμα της αρχικής με μειωμένη ανάλυση. Συνέπεια αυτού είναι ότι μπορούν να εξαχθούν FAST

¹ Για τη σωστή λειτουργία του ORB SLAM2 σε περιβάλλον Linux πρέπει να ορίζονται σωστά όλα τα path.

περιγραφείς χαρακτηριστικών σημείων (Features from Accelerated Segment Test) σε διαφορετικές κλίμακες της αρχικής εικόνας. Με αυτό τον τρόπο αντιμετωπίζεται το πρόβλημα του να χαθούν κάποια σημαντικά στοιχεία από τις εικόνες αν γινόταν εφαρμογή σε ένα μόνο επίπεδο. Στην παρούσα εφαρμογή χρησιμοποιήθηκαν 8 επίπεδα με συντελεστή κλίμακας 1.2.



Εικόνα 12. Πυραμίδα εικόνων που αποτελείται από 4 επίπεδα (levels) και έχει συντελεστή κλίμακας 2

2. Ακολουθεί ο εντοπισμός των γωνιών με χρήση των χαρακτηριστικών FAST των εικόνων. Η διαδικασία εντοπισμού είναι αναλυτικότερα η εξής: Για κάθε pixel p της εικόνας, δημιουργείται γύρω του κύκλος ακτίνας 3 pixel. Για να θεωρηθεί ότι εντοπίστηκε μια γωνία θα πρέπει τουλάχιστον 9 συνεχόμενα pixel του κύκλου να είναι φωτεινότερα ή σκοτεινότερα του pixel p και μιας τιμής κατωφλίσωσης (threshold) t προστιθέμενης στο p . Προκειμένου να αποφευχθεί ο εντοπισμός των ίδιων γωνιών, επιλέγεται το pixel με την υψηλότερη τιμή του αποτελέσματος της πρόσθεσης των απόλυτων διαφορών του p με καθένα από τα pixel γειτνίασης.

Πλεονέκτημα των FAST corner detectors σε σχέση με άλλους σημειακούς τελεστές, πχ. Harris, είναι ότι είναι αρκετά πιο γρήγοροι. Μεγάλο μειονέκτημα είναι ότι δεν είναι αρκετά σταθεροί όσον αφορά το «θόρυβο». Το ORB SLAM2 διαιρεί την κάθε εικόνα της πυραμίδας σε κελιά 30×30 και προσπαθεί να υπολογίσει τους ανιχνευτές γωνιών FAST σε κάθε ένα από αυτά. Σε περίπτωση όπου δεν μπορέσει να εντοπίσει έστω μια γωνία με το αρχικό κατώφλι (στην προκειμένη περίπτωση είναι $t_{init} = 20$), θα προσπαθήσει εκ νέου με $t_{min} = 7$. Με αυτό τον τρόπο ο αλγόριθμος θα εντοπίσει όλες τις γωνίες σε κάθε εικόνα στην πυραμίδα. Αυτό επιτυγχάνεται (με την προϋπόθεση ότι δεν έχει εντοπιστεί καμία γωνία),

με διαίρεση της εικόνας σε 4 ίδια κελιά, κάθε ένα από τα οποία διαιρείται ξανά σε 4 κελιά. Η διαδικασία επαναλαμβάνεται έως ότου όλα τα υπολειπόμενα κελιά περιέχουν ακριβώς μια γωνία ή θα επιλεγεί η γωνία με την υψηλότερη τιμή για κάθε κελί που περιέχει παραπάνω από μια γωνίες (όπως προαναφέρθηκε).

3. Επόμενο βήμα του ORB SLAM2 είναι ο υπολογισμός των περιγραφών ORB (Oriented Fast and Rotated Brief). Οι συγκεκριμένοι περιγραφείς είναι ένας συνδυασμός των FAST και των BRIEF (Binary Robust Independent Elementary Features). Θετικό χαρακτηριστικό των περιγραφών χαρακτηριστικών ORB είναι ότι περιέχουν πληροφορίες προσανατολισμού σε αντίθεση με τους ανιχνευτές γωνιών. Επίσης οι BRIEF είναι δυαδικοί περιγραφείς string οι οποίοι συγκρίνουν μέσω της απόστασης Hamming τα strings (η Hamming distance μεταξύ 2 strings υπολογίζεται μέσω της πράξης XOR αυτών των strings).

Ωστόσο μειονέκτημα των descriptors BRIEF είναι ότι δεν μπορούν να χειριστούν τις στροφές επιπέδων. Για την αντιμετώπιση αυτού του προβλήματος, χρησιμοποιούνται οι rBRIEF (rotated BRIEF) – πράσινες γραμμές κατά την εκκίνηση του ORB SLAM2 στο παράθυρο Current frame όπως διακρίνεται στο βίντεο που επισυνάπτεται (video_orbslam2_phone). Σκοπός τους είναι η αύξηση της διακύμανσης και η μείωση της συσχέτισης. Αφού χρησιμοποιηθούν οι rBRIEF γίνεται ταύτιση με τον προσανατολισμό που πραγματοποιείται από τους περιγραφείς γωνιών FAST και έτσι οι περιγραφείς ORB δεν επηρεάζονται από αλλαγές στις στροφές. Αφού εξαχθούν όλα τα χαρακτηριστικά των εικόνων, πραγματοποιείται η εξάλειψη της παραμόρφωσής τους με τη βοήθεια των παραμέτρων του προσανατολισμού της κάμερας (που εξήχθησαν από το MATLAB στην συγκεκριμένη περίπτωση).

6.1.2 Αρχικοποίηση (Initialization)

Κατά την εκκίνηση της διαδικασίας, το λογισμικό προσπαθεί να εντοπίσει τουλάχιστον 100 χαρακτηριστικά (features) στο τρέχον και στο τελευταίο πλαίσιο (trying to initialize – video_orbslam2_phone). Όταν βρει αυτά τα χαρακτηριστικά, το συγκεκριμένο frame χρησιμοποιείται ως αρχικό και όλα τα υπόλοιπα θα συγκρίνονται με αυτό.

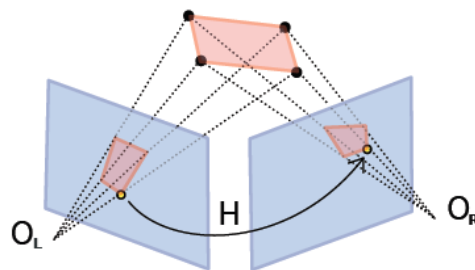
Προκειμένου να βρεθεί μια σύνδεση μεταξύ των χαρακτηριστικών του αρχικού πλαισίου και του τρέχοντος γίνεται σύγκριση της απόστασης και του προσανατολισμού του κάθε χαρακτηριστικού σημείου.

Αρχικά για κάθε χαρακτηριστικό σημείο στο αρχικό πλαίσιο, συλλέγονται όλα τα χαρακτηριστικά σημεία σε μια περιοχή 200×200 pixels γύρω από

το ίδιο χαρακτηριστικό στο τρέχον πλαίσιο. Έπειτα για κάθε χαρακτηριστικό σε αυτή την περιοχή υπολογίζεται η απόσταση του περιγραφέα στο αρχικό πλαίσιο. Δύο χαρακτηριστικά συνδέονται, όσον αφορά την απόσταση, όταν η απόσταση του περιγραφέα είναι μικρότερη του 50 και η μικρότερη απόσταση που υπολογίστηκε είναι μικρότερη κατά 0.9 φορές της δεύτερης μικρότερης απόστασης.

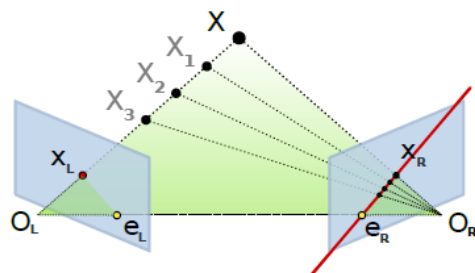
Από την στιγμή που ο αλγόριθμος εγκρίνει όλες τις αποστάσεις, ακολουθεί η σύγκριση του προσανατολισμού του κάθε ζεύγους των χαρακτηριστικών, τα οποία έχουν προκύψει από το προηγούμενο βήμα. Αφού εξεταστούν και οι προσανατολισμοί και εγκριθούν, τότε ξεκινά η διαδικασία της αρχικοποίησης – initialization (εντοπισμός τουλάχιστον 100 matches).

Σε αυτό το σημείο, ο αλγόριθμος υπολογίζει τον πίνακα $\mathbf{H}$ ομογραφίας (Homography matrix – ομογενείς συντεταγμένες) και τον επιπολικό πίνακα $\mathbf{F}$ (Fundamental matrix). Άρα πρέπει να έχει χρησιμοποιηθεί η ίδια κάμερα σε δύο διαφορετικές γωνίες λήψης.



Εικόνα 13. Homography matrix

Ο $\mathbf{H}$ μεταφέρει απευθείας ένα σημείο, το οποίο ανήκει στο σύστημα συντεταγμένων της αριστερής εικόνας, στο σύστημα συντεταγμένων της δεξιάς με την προϋπόθεση ότι ο εσωτερικός προσανατολισμός της κάμερας είναι ίδιος. Επίσης ο $\mathbf{F}$ αντιστοιχίζει σημεία σε στερεοσκοπικές εικόνες. Χρησιμοποιείται σε μη επίπεδες επιφάνειες και περιγράφει την επιπολική γεωμετρία του χώρου.



Εικόνα 14. Fundamental matrix

Τόσο ο **H** όσο και ο **F** εκτιμώνται με τον την βοήθεια του αλγορίθμου RANSAC (Random Sample Consensus). Στη συνέχεια προβάλλονται τα κοινά σημεία από το ένα πλαίσιο στο άλλο με βάση τους υπολογισμένους μετασχηματισμούς. Προκειμένου να χρησιμοποιηθεί ο **H** ή ο **F**, το ORB SLAM2 χρησιμοποιεί μια τιμή η οποία προκύπτει από την σχέση

$$R_H = S_H / (S_H + S_F)$$

R_H : Decision

S_H : Homography score

S_F : Fundamental matrix score

Σε περίπτωση που το $R_H > 0$, επιλέγεται ο **H**, σε αντίθετη περίπτωση ο **F**. Και στις δύο περιπτώσεις πραγματοποιείται ο τριγωνισμός όλων των αντιστοιχισμένων χαρακτηριστικών σημείων προκειμένου να ελεγχθεί αν βρίσκονται μπροστά από τις κάμερες, η παράλλαξη είναι μεγάλη και το σφάλμα επαναπροβολής είναι μικρό. Ο τριγωνισμός επιτυγχάνεται με χρήση του αλγόριθμου DLT (Direct Linear Transformation). Η ελάχιστη παράλλαξη έχει οριστεί σε 1° και τουλάχιστον 90% των αντιστοιχιών πρέπει να τριγωνιστούν σωστά με ελάχιστο αριθμό τα 50. Ο συγκεκριμένος αλγόριθμος κάνει χρήση των ομογενών συντεταγμένων δύο 2D σημείων καθώς και των πινάκων προσανατολισμού της κάμερας.

Από τη στιγμή που πραγματοποιηθεί το βήμα της αρχικοποίησης, δημιουργείται ένας αρχικός χάρτης, ο οποίος περιέχει το αρχικό και το τρέχον keyframe, τα οποία είναι τα πρώτα δύο keyframes, και όλες τις τριγωνισμένες αντιστοιχίες σαν αρχικά σημεία του χάρτη. Σε αυτό το σημείο εκτελείται μια συνολική συνόρθωση δέσμης (global adjustment) στον αρχικό χάρτη. Σε περίπτωση όπου δεν έχουν εντοπιστεί τουλάχιστον 100 σημεία στον χάρτη, το βήμα της αρχικοποίησης θα απορριφθεί και η διαδικασία θα εκτελεστεί ξανά.

6.1.3 Tracking

Εφ' όσον το βήμα της αρχικοποίησης είναι επιτυχές, ο αλγόριθμος θα εκκινήσει το module του εντοπισμού, το οποίο εκτιμά την κίνηση της κάμερας και χαρτογραφεί τον περιβάλλοντα χώρο. Ο εντοπισμός μπορεί να πραγματοποιηθεί είτε με την ύπαρξη ενός μοντέλου κίνησης είτε χωρίς. Στην προκειμένη περίπτωση, όπως παρουσιάστηκε στο επισυναπτόμενο βίντεο, ισχύει η 1^η περίπτωση.

Tracking with a motion model

Ο αλγόριθμος μπορεί να συσχετίσει τα δεδομένα, υποθέτοντας ότι τα χαρακτηριστικά σημεία, τα οποία έχουν χαρτογραφηθεί σε κάποιο σημείο του χάρτη σε προηγούμενη εικόνα, μπορούν να εντοπιστούν εντός μιας περιοχής σε μια νέα εικόνα. Αυτό καθορίζεται με βάση την κίνηση της κάμερας. Το μέγεθος της περιοχής εξαρτάται από το σε ποιο επίπεδο της πυραμίδας της εικόνας εξήχθη το χαρακτηριστικό σημείο. Για παράδειγμα, αν εξαχθεί από τη βάση της πυραμίδας, η περιοχή εντοπισμού του συγκεκριμένου χαρακτηριστικού σημείου στη νέα εικόνα θα είναι 14×14 pixel. Αν το αποτέλεσμα εντοπισμού με το προηγούμενο πλαίσιο είναι μικρότερο από 20 αντιστοιχίες, θα γίνει προσπάθεια εντοπισμού ξανά σε περιοχή μεγαλύτερη, πχ. 28×28 pixel. Κάθε χαρακτηριστικό του επιλεγμένου πλαισίου θα χαρτογραφηθεί με το σημείο του χάρτη του ίδιου χαρακτηριστικού στο προηγούμενο πλαίσιο. Σε περίπτωση εντοπισμού το πολύ 20 αντιστοιχιών, το οποίο δεν είναι αποδεκτό στην επόμενη έρευνα, ο αλγόριθμος θα συνεχίσει τη λειτουργία του με το βήμα *tracking without a motion model*.

6.1.4 Επανεκτοπισμός

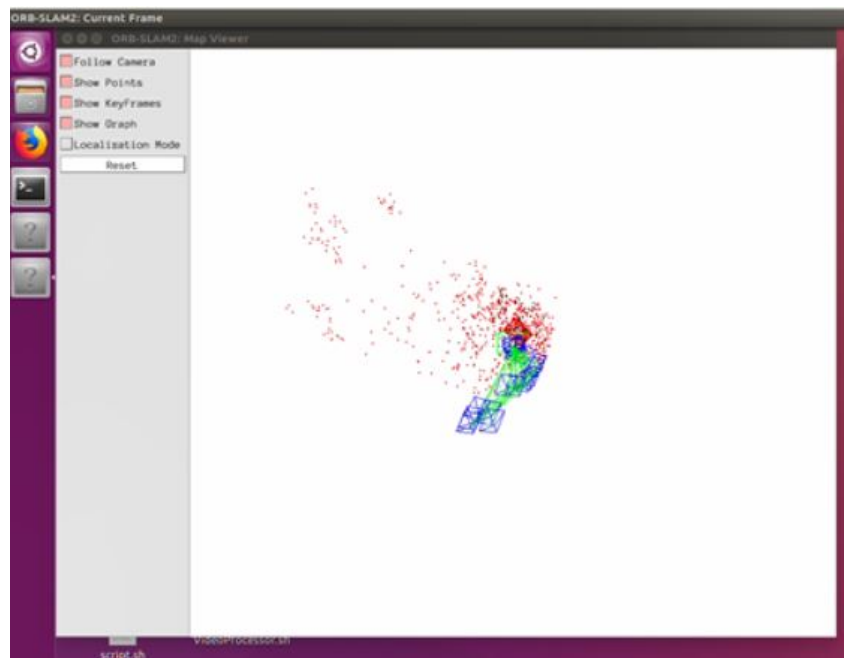
Όταν ο εκτοπισμός αποτύχει, το ORB SLAM2 εκτελεί τον επανεκτοπισμό. Όπως παρατηρείται στο βίντεο, από την στιγμή που χαθεί ο εκτοπισμός, στο παράθυρο *current frame*, ο αλγόριθμος προσπαθεί να κάνει επανεκτοπισμό.

Αναλυτικότερα, προσπαθεί να εντοπίσει ένα *keyframe*, το οποίο ‘βλέπει’ την ίδια σκηνή με το τρέχον πλαίσιο. Αυτό σημαίνει ότι δεν θα πραγματοποιηθεί περαιτέρω χαρτογράφηση ή επανεκτοπισμός μέχρι να βρεθεί ένα *keyframe* το οποίο είναι ‘γνωστό’.

Σε αυτό το σημείο χρησιμοποιείται η βιβλιοθήκη DBoW2 (τεχνική *bag of visual words*, κατά την οποία μια εικόνα μετατρέπεται σε αριθμητικό διάνυσμα με σκοπό τη δημιουργία βάσης δεδομένων με τα διανύσματα αυτά τοποθετημένα ιεραρχικά). Έπειτα, με τη βοήθεια της βιβλιοθήκης, γίνεται προσπάθεια εντοπισμού των *keyframes*, τα οποία είναι παρόμοια με το τρέχον. Σε περίπτωση όπου εντοπίζονται το πολύ 15 αντιστοιχίες, τα αντίστοιχα *keyframes* απορρίπτονται. Ακολουθεί η εκτέλεση του αλγορίθμου EPnP με σκοπό την εκτίμηση της θέσης και του προσανατολισμού (*pose*) όλων των αντιστοιχιών στα *keyframes* τα οποία

είναι αποδεκτά. Για να επιτευχθεί αυτό πρέπει η κάμερα να είναι προσανατολισμένη ώστε να βρεθούν οι n ομολογίες από 3D σε 2D σημεία. Από τη στιγμή που βρεθούν οι ομολογίες αυτές, ο αλγόριθμος τρέχει 5 επαναλήψεις με $n = 4$ σε τυχαία σημεία και προσπαθεί να υπολογίσει τη θέση αυτών. Μία θέση θεωρείται ότι βρέθηκε όταν τουλάχιστον 10 σημεία έχουν αρκετά μικρό σφάλμα επαναπροβολής. Το βήμα του επανεντοπισμού θεωρείται επιτυχές όταν βρεθούν τουλάχιστον 50 αντιστοιχίες ανάμεσα στα υποψήφια εντοπισμένα σημεία του χάρτη και στα χαρακτηριστικά του τρέχοντος keyframe.

6.1.5 Local mapping



Εικόνα 15.3D ανακατασκευή με χρήση ORB SLAM2

Στην Εικ. 12 παρατηρεί κανείς τον χάρτη που δημιουργήθηκε με το ORB SLAM2. Τα μπλε σχήματα αφορούν τις εκτιμώμενες θέσεις της κάμερας για τα keyframes. Οι κόκκινες και μαύρες κουκίδες είναι τα σημεία του χάρτη. Συγκεκριμένα, οι κόκκινες αφορούν τον τρέχοντα χάρτη. Ένας χάρτης ορίζεται από δύο σύνολα keyframes, $K1$ και $K2$. Το $K1$ περιέχει όλα τα keyframes τα οποία έχουν κοινά σημεία με το τρέχον. Το $K2$ περιέχει όλα τα keyframes τα οποία γειτονεύουν με κάποιο keyframe στο $K1$ μέσω του covisibility graph (γράφημα στο οποίο, όπως προαναφέρθηκε, κάθε κόμβος αντιπροσωπεύει τη θέση κάθε σημείου και

κάθε ακμή την μεταξύ τους ένωση).

Το ORB SLAM2 δημιουργεί χάρτη από τον περιβάλλοντα χώρο. Το local mapping module είναι αρμόδιο να κρατά τον χάρτη ενημερωμένο εισάγοντας ή διαγράφοντας keyframes και σημεία του χάρτη, και κάθε φορά που συμβαίνει αυτό κάνει βελτιστοποίηση στον χάρτη.

Για να επιτευχθεί αυτό, το module τρέχει με επαναλήψεις και κάθε 3ms ελέγχει αν υπάρχουν νέα keyframes, τα οποία πρέπει να ενσωματωθούν στον χάρτη. Από την στιγμή που εντοπιστεί ένα νέο (keyframe) ακολουθούνται τα παρακάτω βήματα:

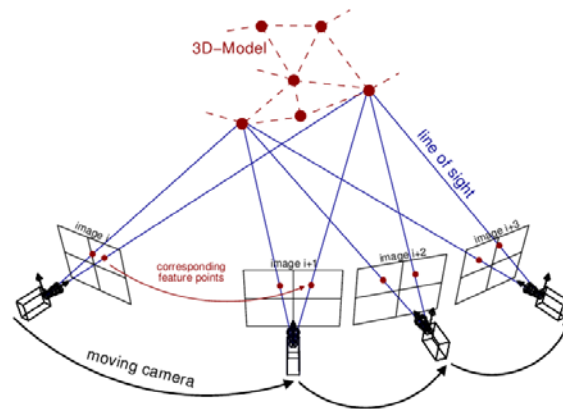
- Εισαγωγή του keyframe στον χάρτη, με ενημέρωση όλων των απαραίτητων συνδέσεων και αντικειμένων.
- Έλεγχος αν μπορούν να διαγραφούν κάποια σημεία του χάρτη. Για να διαγραφεί κάποιο σημείο θα πρέπει:
 - Ο λόγος του πόσο συχνά είναι ορατό από κάποιο πλαίσιο και του πόσο συχνά είναι συνταυτισμένο (matched) να είναι μικρότερος του 0.25.
 - Να έχει γίνει συνταύτιση λιγότερο από 3 φορές και να έχουν εισαχθεί περισσότερα από 1 keyframe στον χάρτη από την πρώτη φορά που εισήχθη το ίδιο.

Σε περίπτωση που ένα σημείο δεν διαγραφεί στα πρώτα 2 keyframes, θα παραμείνει στον χάρτη και δε θα υπάρχει περίπτωση να διαγραφεί στο μέλλον².

- Τα νέα σημεία του χάρτη δημιουργούνται από τον τριγωνισμό των feature matches ανάμεσα στο τρέχον keyframe και στα 20 keyframes, τα οποία διαμοιράζονται τα περισσότερα σημεία του χάρτη με το τρέχον. Τριγωνίζονται τα σημεία του χάρτη, τα οποία δεν σχετίζονται με κάποιο άλλο σημείο.
- Αν δεν υπάρχουν νέα keyframes για εισαγωγή, γίνεται έλεγχος για «διπλά σημεία» (duplicate map points) και ένωσή τους.
- Αν παρ' όλα αυτά ακόμα δεν υπάρχει κάποιο keyframe για εισαγωγή, ο αλγόριθμος πραγματοποιεί μια τοπική συνόρθωση δέσμης (Local Bundle Adjustment) και έπειτα απομακρύνει τα keyframes, τα οποία μοιράζονται τουλάχιστον το 90% των map points με τουλάχιστον 3 διαφορετικά keyframes.

² Το ORB SLAM2 στην πραγματικότητα δεν διαγράφει τα σημεία του χάρτη από την μνήμη. Απλώς τα δηλώνει ως 'unused'.

Η συνόρθωση δέσμης χρησιμοποιείται ευρέως σε συστήματα SLAM. Με τη συνόρθωση δέσμης επιτυγχάνεται η προβολή ενός 3D σημείου X στο ομόλογό του 2D σημείο x στην εικόνα μέσω του P πίνακα της κάμερας (εικόνα n). Εφόσον όμως υπάρχει το πρόβλημα του θορύβου σκοπός της ΒΑ είναι να ελαχιστοποιήσει το σφάλμα επαναπροβολής και να επιλύσει σωστά την σχέση $x = P * X$. Σε αυτό το σημείο γίνεται χρήση της βιβλιοθήκης g^2o του ORB SLAM2.



Εικόνα 16. Bundle adjustment

6.1.6 Loop closing

Όπως το local mapping module, έτσι και το loop closing module εκτελείται ανεξάρτητα από τις υπόλοιπες διεργασίες. Αυτό οφείλεται στο ότι πρέπει να παρατηρούνται τα δεδομένα συνεχώς ώστε να εντοπιστούν οι βρόχοι (loops). Η εκτέλεση του συγκεκριμένου module πραγματοποιείται σε 3 βήματα.

- Εντοπισμός βρόχου (loop detection)
- Επιβεβαίωση βρόχου (loop confirmation)
- Περάτωση (κλείσιμο) βρόχου (loop refinement)

Loop detection

Το ORB SLAM2 ελέγχει για την ύπαρξη βρόχων μόνο στην περίπτωση που εντοπιστεί ένα νέο keyframe. Αυτό γίνεται όταν το local mapping module τελειώσει την επεξεργασία του συγκεκριμένου keyframe. Από την στιγμή που εισαχθεί το νέο keyframe, το module προσπαθεί να εντοπίσει άλλα keyframes, τα οποία συνδέονται με το νέο μέσω του covisibility graph. Δεδομένου ότι υπάρχουν τουλάχιστον 10 keyframes στον χάρτη και η προηγούμενη διεργασία περάτωσης βρόχων πραγματοποιήθηκε 10 keyframes νωρίτερα, θα ζητηθεί από τη βάση δεδομένων των keyframes να επιστραφούν όλα τα keyframes που μοιράζονται τουλάχιστον μια

‘λέξη’ (bag of words) με το τρέχον keyframe. Αυτά τα keyframes θα χρησιμοποιηθούν ως υποψήφια (candidates) για την ολοκλήρωση της περάτωσης βρόχων.

Στη συνέχεια, για κάθε ένα από τα υποψήφια keyframes δημιουργείται ένα σύνολο υποψήφιων, στο οποίο περιέχονται όλα τα keyframes τα οποία συνδέονται με το υποψήφιο. Έπειτα ελέγχεται το κάθε group ως προς την συνοχή του (consistency). Ένα σύνολο θεωρείται ότι έχει συνοχή όταν έχει επικάλυψη με κάποιο σύνολο, το οποίο δημιουργήθηκε για το τελευταίο keyframe που «πέρασε» από το loop detection module. Λέγοντας επικάλυψη, εννοεί κανείς ότι δύο σύνολα υποψήφιων έχουν τουλάχιστον ένα κοινό keyframe. Από τη στιγμή που κάποιο σύνολο υποψήφιων πετύχει μέτρηση συνοχής μεγαλύτερη του 2, περνά στο δεύτερο βήμα του loop closing module, δηλαδή στο loop confirmation.

Loop confirmation

Σε περίπτωση που κάποιο σύνολο υποψήφιων έχει μεγάλη τιμή συνοχής, το τελευταίο keyframe του συνόλου αυτού θα αξιολογηθεί από τον αλγόριθμο αν θα μπορεί να θεωρηθεί ως βρόχος. Για να επιτευχθεί ο συγκεκριμένος έλεγχος, το ORB SLAM2 προσπαθεί να εντοπίσει αντιστοιχίες μεταξύ του τρέχοντος keyframe και του υποψήφιου keyframe (ο έλεγχος γίνεται απευθείας ανάμεσα στα σημεία του χάρτη του τρέχοντος keyframe και στα σημεία του υποψήφιου keyframe με χρήση των αποθηκευμένων περιγραφών ORB, και όχι ανάμεσα σε 3D και 2D σημεία του χάρτη).

Εφ’ όσον βρεθούν τουλάχιστον 20 αντιστοιχίες, ο αλγόριθμος έχει επιτύχει να εντοπίσει ζευγάρια από σημεία του χάρτη, τα οποία αντιστοιχούν σε 3D σημεία του πραγματικού περιβάλλοντος. Το πρόβλημα όμως που υπάρχει είναι η συσσωρευτική κλίμακα ολίσθησης, το οποίο σημαίνει ότι σημεία του χάρτη δεν είναι τόσο κοντά το ένα με το άλλο αν και αντιπροσωπεύουν το ίδιο πραγματικό σημείο.

Στην περίπτωση του εντοπισμού, η λύση της ολίσθησης έχει 6 βαθμούς ελευθερίας: 3 για την μετατόπιση και 3 για τις στροφές. Ωστόσο στον μονοσκοπικό εντοπισμό υπάρχει και μια επιπλέον παράμετρος η οποία είναι η κλίμακα. Αυτό μπορεί να γίνει εύκολα αντιληπτό θεωρηθεί, κατά κάποιον τρόπο σαν παράδειγμα, ο άνθρωπος. Σε περίπτωση που κοιτά με το ένα μάτι χάνει την αίσθηση του βάθους. Με τον ίδιο τρόπο, οι αλγόριθμοι SLAM χρησιμοποιώντας μια απλή κάμερα χάνουν την

αίσθηση της κλίμακας του περιβάλλοντος διότι δεν υπάρχουν σημεία αναφοράς. Αυτό γίνεται διότι γίνεται εκτίμηση του περιβάλλοντος με ένα συντελεστή κλίμακας μεταξύ του τοπικού συστήματος συντεταγμένων σε units και πραγματικών συντεταγμένων σε μέτρα. Έτσι προστίθεται, όπως προαναφέρθηκε, ένας νέος βαθμός ελευθερίας, η αθροιστική κλίμακα ολίσθησης.

Για την επίλυση, ο αλγόριθμος χρησιμοποιεί μετασχηματισμό ομοιότητας μεταξύ του τρέχοντος keyframe και του υποψήφιου loop closing keyframe.

$$S = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}$$

$S \in Sim(3)$	Similarity transformation
$s \in \mathbb{R}$	Scale factor
$R \in SO(3)$	Rotation matrix
$t \in \mathbb{R}^3$	Translation vector

$SO(3) \rightarrow$ στροφές στον 3D χώρο

$Sim(3) \rightarrow$ μεταθέσεις στον 3D χώρο

Στη συνέχεια ο αλγόριθμος χρησιμοποιεί αυτή την τιμή (S), ώστε να ελέγξει αν το σφάλμα είναι μεγάλο όταν προβάλλει τα αντίστοιχα σημεία του χάρτη. Πρέπει να βρεθούν τουλάχιστον 20 αντιστοιχίες ώστε ο μετασχηματισμός να είναι αποδεκτός.

Στα επόμενα βήματα, το ORB SLAM2 προσπαθεί να εντοπίσει περισσότερες 3D - 3D αντιστοιχίες με την χρήση του παραπάνω υπολογισμένου μετασχηματισμού για περαιτέρω βελτιστοποίησή του. Τελικά, γίνεται προσπάθεια εντοπισμού επιπλέον αντιστοιχισμένων σημείων του χάρτη με την προβολή των σημείων αυτών που βρέθηκαν στα συνδεδεμένα keyframes των υποψήφιων στο βήμα της περάτωσης βρόχων. Σε περίπτωση που βρεθούν τουλάχιστον 40 ομολογίες, το υποψήφιο keyframe είναι αποδεκτό και ο αλγόριθμος εκτελεί το τελευταίο βήμα, το loop refinement.

Loop refinement

Στο τελευταίο βήμα ο αλγόριθμος έχει ως σκοπό το κλείσιμο του βρόχου. Σε αυτό το σημείο το local mapping module διακόπτεται ώστε να μην αλλάξει ο χάρτης που έχει ήδη δημιουργηθεί κατά τη διάρκεια της βελτιστοποίησής του. Το τελευταίο αφορά τις συνδέσεις του covisibility

graph, έτσι ώστε το τρέχον keyframe να συνδεθεί με εκείνα που βρίσκονται γύρω από το υποψήφιο. Έπειτα συλλέγονται όλα τα keyframes που συνδέονται με το τρέχον και χρησιμοποιείται ο μετασχηματισμός ομοιότητας ώστε να διορθωθούν οι στροφές και οι μεταθέσεις όλων των συνδεδεμένων keyframes και των σημείων του χάρτη. Αυτό έχει ως αποτέλεσμα να μετακινηθούν όλα τα παραπάνω στο σημείο στο οποίο θα βρίσκονται αφού υλοποιηθεί το loop closing module. Ακόμα, πρέπει να διορθωθούν όλα τα υπόλοιπα keyframes που βρίσκονται στο module και να διαγραφούν τα «διπλά σημεία του χάρτη». Για τη διαγραφή, ο αλγόριθμος προβάλλει όλα τα σημεία που ανήκουν σε συνδεδεμένα keyframes στο σύστημα συντεταγμένων της εικόνας και προσπαθεί να τα συνδέσει με χαρακτηριστικά σημεία. Αν βρεθεί αντιστοιχία για κάποιο χαρακτηριστικό το οποίο είναι συνδεδεμένο με διαφορετικό σημείο στον χάρτη, τότε θεωρείται «διπλό» και διαγράφεται. Αυτό είναι και το 1^ο βήμα του loop refinement. Με βάση αυτές τις αντιστοιχίες οι συνδέσεις στο convisibility graph ενημερώνονται, και συνδέονται keyframes και από τις 2 πλευρές του χάρτη.

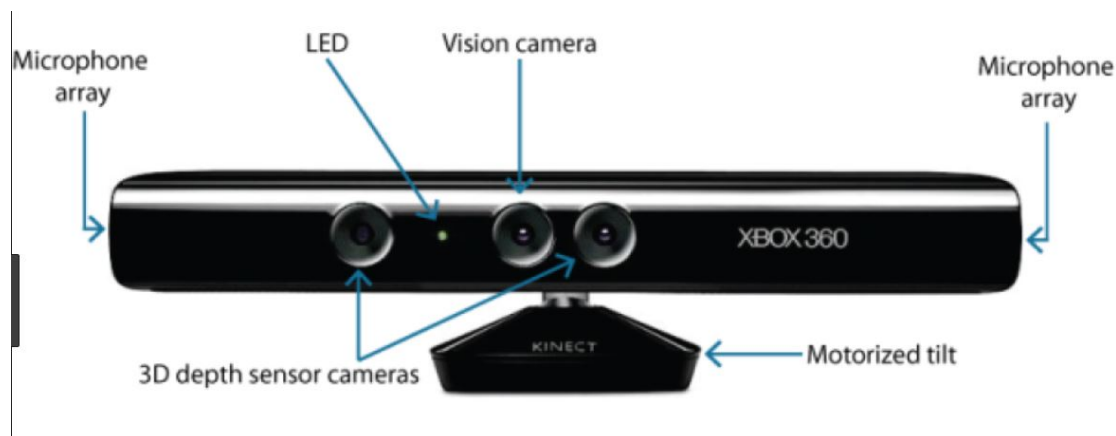
Ακολουθεί η διόρθωση του δεσμευμένου επιπολικού πίνακα (Essential matrix), ένας πίνακας 3×3 που συνδέει σημεία σε βαθμονομημένα ζεύγη εικόνων. Σε αυτό το σημείο η διόρθωση πραγματοποιείται μόνο στις πιο σημαντικές ακμές, με αποτέλεσμα να είναι πολύ πιο γρήγορη διαδικασία από την υλοποίηση στο convisibility graph.

Τελευταίο βήμα είναι η global συνόρθωση δέσμης, η οποία εκτελείται σε ξεχωριστό βήμα και βελτιστοποιεί όλα τα keyframes και σημεία του χάρτη, ενώ παράλληλα πρόκειται να εκτελεστεί και το local mapping module. Το πρόβλημα που προκύπτει είναι ότι αυτή η διαδικασία είναι πολύ χρονοβόρα και το αποτέλεσμα της συνόρθωσης δέσμης πρέπει να ενωθεί με τον τρέχοντα χάρτη [9].

7. Εφαρμογή αλγορίθμου με δεδομένα εισόδου από το Microsoft Kinect V1

7.1 Χρήση Microsoft Kinect, ORB SLAM2 και ROS (Robot Operating System)

Μετά την χρήση του κινητού τηλεφώνου και των δυσκολιών που αντιμετωπίστηκαν όσον αφορά την εκτέλεση του αλγορίθμου, έγινε χρήση του Microsoft Kinect για να ελεγχθεί αν βοηθά περισσότερο η χρήση κάμερας RGB-D (αντί της μονοσκοπικής) στην εκτέλεση του ORB SLAM2. Το συγκεκριμένο hardware παρουσιάζεται στην Εικόνα 20 και αφορά την έκδοση Microsoft Kinect V1 για το Microsoft Xbox 360 (απαραίτητες οι πληροφορίες για την εγκατάσταση σε υπολογιστή, καθώς υπάρχουν και άλλες εκδόσεις με διαφορετικούς οδηγούς και διαφορετικό κώδικα για την εγκατάστασή του). Όπως φαίνεται στην Εικ. 13 υπάρχουν δύο 3D κάμερες με αισθητήρες βάθους (IR - Infrared projector και IR camera) που βοήθησαν να υπολογιστεί η απόσταση από το αντικείμενο (βάθος), η κάμερα οπτικοποίησης η οποία είναι η RGB camera, δύο μικρόφωνα για εγγραφή ήχου ή αναγνώριση φωνής (δεν χρησιμοποιήθηκαν εδώ) και το Motorized tilt το οποίο χρησιμοποιήθηκε για την κίνηση – πάνω ή κάτω – του Kinect.

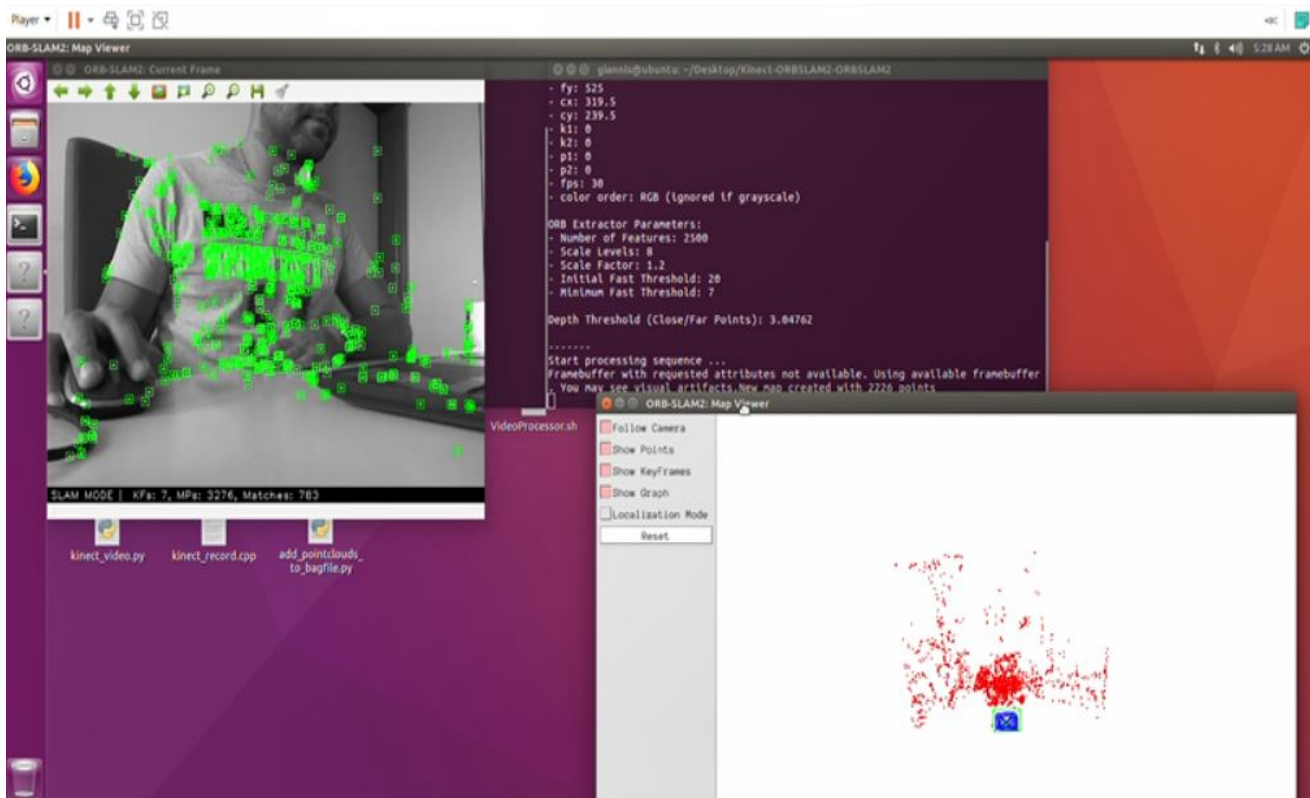


Εικόνα 17. Microsoft Kinect V1 για το Microsoft XBOX 360

Αρχικά εγκαταστάθηκε το Kinect στο VMware Workstation σε περιβάλλον Ubuntu. Οι παράμετροι ‘βαθμονόμησης’ που χρησιμοποιήθηκαν είναι οι προεπιλεγμένες τιμές που παρέχονται από την Microsoft. Επίσης εγκαταστάθηκαν οι οδηγοί Open Kinect, OpenNi, η βιβλιοθήκη freenect καθώς και το προαιρετικό προαπαιτούμενο ROS. Ο driver OpenNI χρησιμοποιεί τις τιμές της ‘βαθμονόμησης’ για να διορθώσει τις παραμορφώσεις των εικόνων και να συνδέσει τις εικόνες

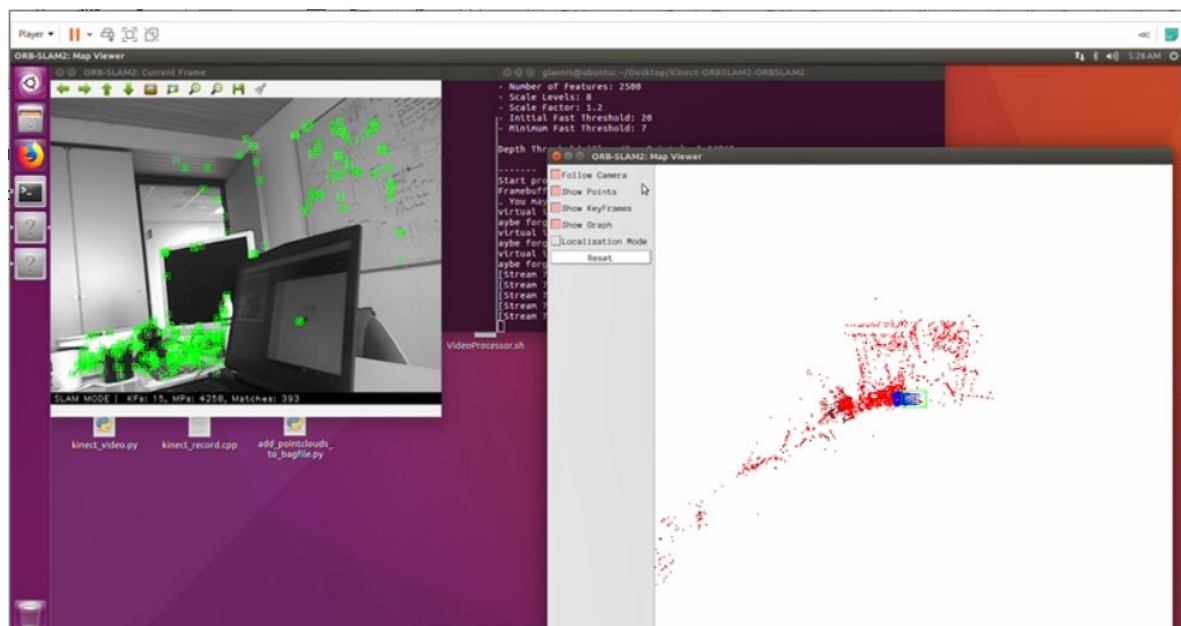
βάθους της κάμερας υπερύθρων (IR) με τις εικόνες RGB. Στην προκειμένη περίπτωση υπάρχει αντιστοίχιση 1:1 μεταξύ των pixel των εικόνων βάθους και των RGB εικόνων. Το ROS επιτρέπει να επεξεργάζεται κανείς live τα δεδομένα εισόδου μονοσκοπικής, στερεοσκοπικής ή RGB-D κάμερας στο ORB SLAM2, τα οποία στη συγκεκριμένη περίπτωση ελήφθησαν από το Microsoft Kinect (RGB-D). Επίσης χρησιμοποιήθηκε Raspberry Pi car με σκοπό να δοθεί μια απόσταση μέχρι ένα αντικείμενο και να σταματήσει πριν το χτυπήσει. Όλη η διαδικασία που εκτελεί το ORB SLAM2 με χρήση του Kinect είναι ίδια με τη διαδικασία που περιγράφηκε παραπάνω για τη μονοσκοπική κάμερα του κινητού, με τη διαφορά ότι τώρα ο αισθητήρας ορίστηκε σε RGB-D ώστε να υπάρχει και η πληροφορία του βάθους (απόστασης). Έτσι είναι γνωστή και η κλίμακα σε αντίθεση με τη μονοσκοπική λύση. Όπως φαίνεται στην Εικ. 18 ο αριθμός των χαρακτηριστικών ORB είναι 2500 και έχει οριστεί μια επιπλέον παράμετρος Depth threshold (Close / Far points) = 3.04762. Αναλυτικότερα, εκτός των παραμέτρων βαθμονόμησης του Kinect (όπως προαναφέρθηκε είναι οι προεπιλεγμένες και παρέχονται από την Microsoft), στο αρχείο Settings.yaml³ χρησιμοποιήθηκε η τιμή βάσης του Kinect η οποία είναι 7.5 cm = 0.07m (το μήκος της βάσης μεταξύ του IR projector και της κάμερας υπερύθρων). Επίσης ως Depth factor ορίζεται ένας συντελεστής κλίμακας του χάρτη βάθους που υπάρχει ως δεδομένο εισόδου στην περίπτωση της RGB-D κάμερας και ως camera.bf ορίζεται το οριζόντιο εστιακό βάθος σε pixels πολλαπλασιαζόμενο με την τιμή της βάσης σε μέτρα.

³ Σημαντικό είναι να γνωρίζει κανείς ότι στο αρχείο Settings.yaml, στην τιμή Depth map factor = 5000, ορίζεται ότι η τιμή 5000 ενός pixel στην εικόνα βάθους, η οποία αντιστοιχεί σε απόσταση 1 m από την κάμερα.



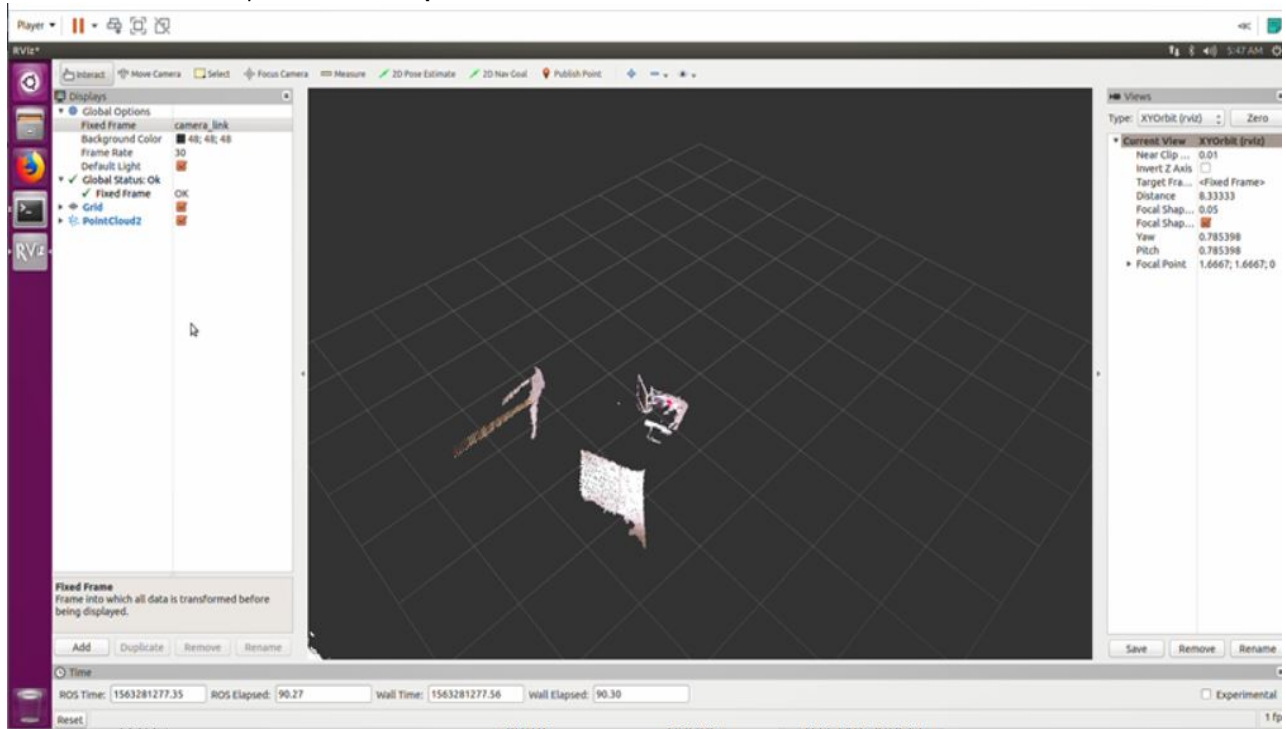
Εικόνα 18. Παράμετροι ORB SLAM2 με χρήση Microsoft Kinect V1

Το αποτέλεσμα όλων των παραπάνω παρουσιάζεται στην Εικ. 19 και αναλυτικότερα στο βίντεο (video_kinect_orbslam2).



Εικόνα 19. Εκτέλεση ORB SLAM2 με χρήση δεδομένων από Microsoft Kinect σε indoor σενάριο

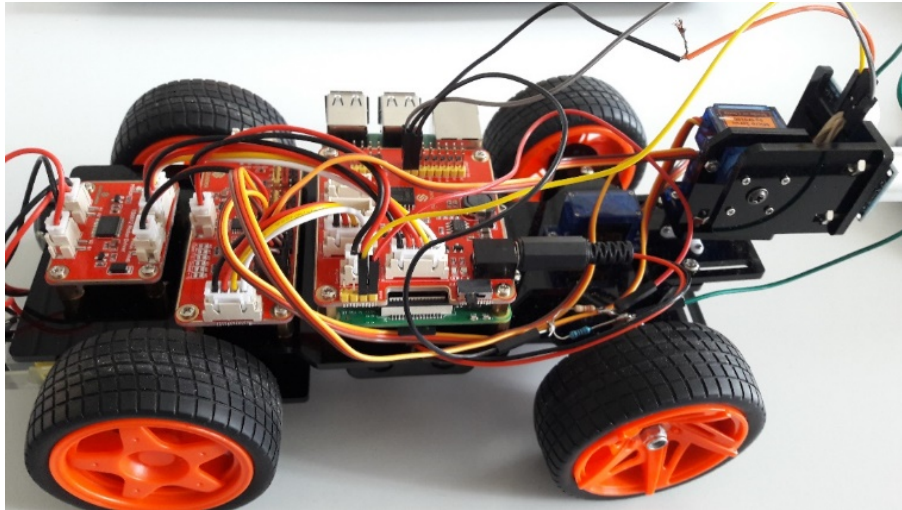
Στο τελευταίο βίντεο (visual_pointcloud_ros_rviz_kinect) παρουσιάζεται, με χρήση του ROS και του προγράμματος Rviz, το νέφος σημείων του γραφείου από το Kinect σε real time. Για να προκύψει το επιθυμητό αποτέλεσμα εκτελέστηκε η εντολή με παράμετρο `depth registration = true`, ορίστηκε στο Rviz η παράμετρος camera link και δημιουργήθηκε topic με το νέφος το οποίο παράγεται από τα συνταυτισμένα σημεία βάθους. Ενδεικτικά φαίνεται στην Εικ.20.



Εικόνα 20. Οπτικοποίηση 3D νέφους σημείων με χρήση Microsoft Kinect, ORB SLAM2 και RViz

7.2 Χρήση Raspberry Pi car σε συνδυασμό με ORB SLAM2 και ROS

Στο τελευταίο πείραμα χρησιμοποιήθηκε ένα Raspberry Pi car, με σκοπό την αποφυγή ενός αντικειμένου (στην προκειμένη περίπτωση να σταματήσει πριν χτυπήσει ένα κουτί). Το αυτοκίνητο φαίνεται στην Εικ. 21.



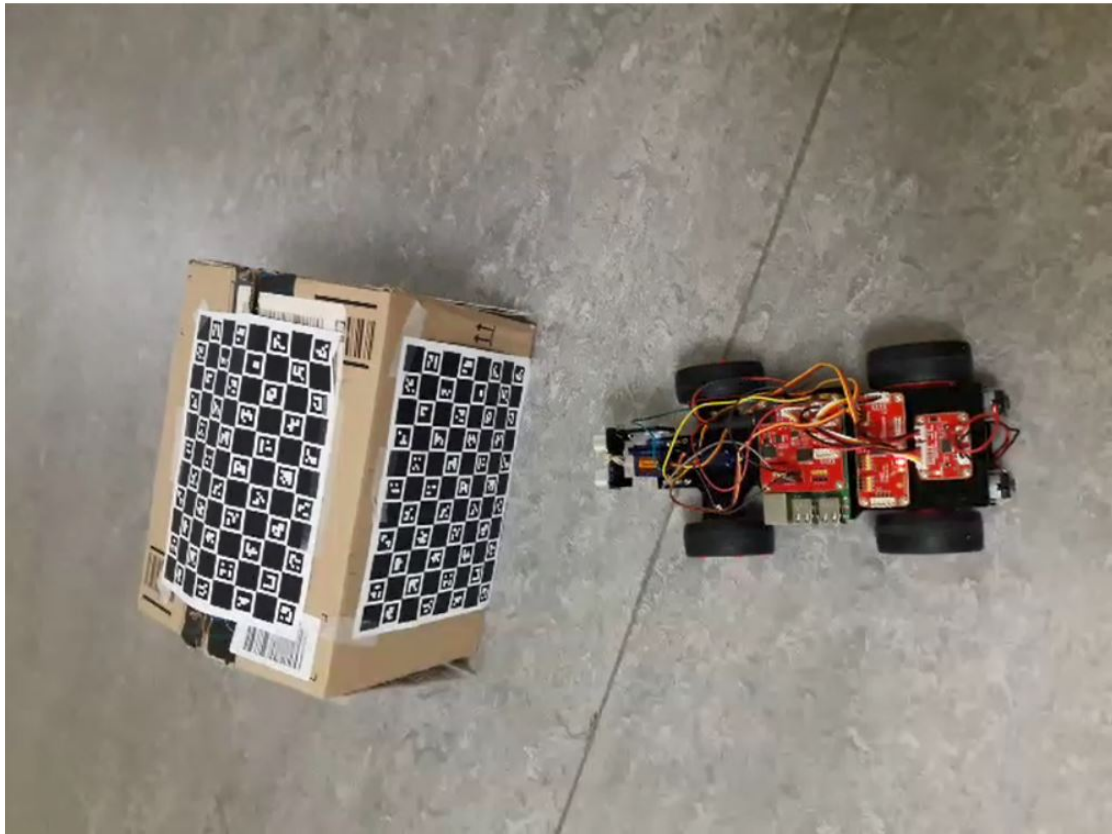
Εικόνα 21. Raspberry Pi car (Sunfounder Smart Video Car kit)

Το συγκεκριμένο μοντέλο είναι το Sunfounder Smart Video Car kit και αποτελείται από το Raspberry Pi model B+, DC – DC converter module, Doppler radar (το οποίο αποσυνδέθηκε για τις ανάγκες της Διπλωματικής Εργασίας), DC motor driver και PCA9685-based servo controller καθώς και τα μηχανικά του μέρη. Όσον αφορά το software είναι της λογικής client–server. Ο VNC client ‘τρέχει’ στο Raspberry Pi για απευθείας έλεγχο του αυτοκινήτου. Ο VNC server ‘τρέχει’ στο laptop με σκοπό να στείλει τις εντολές ελέγχου.

Αρχικά μετετράπη το laptop σε wifi hotspot και συνδέθηκε το αυτοκίνητο μέσω αυτού με το laptop. Στη συνέχεια μέσω του VNC viewer και από τη στιγμή που μπορεί κανείς να διαχειριστεί το λειτουργικό σύστημα του Raspberry Pi από μακριά, έγινε compile στον κώδικα (C code) κατευθείαν πάνω στο αυτοκινητάκι και έπειτα έγινε η εκτέλεσή του.

Αναλυτικότερα, δόθηκε μια τυχαία αρχική ταχύτητα στο αυτοκινητάκι από 0 – 3000. Ωστόσο επειδή έπρεπε να είναι γνωστό πόσα δευτερόλεπτα πρέπει να λειτουργήσει το αυτοκίνητο μέχρι το εμπόδιο, έπρεπε να μετατραπεί αυτή η ταχύτητα σε m/sec. Για τον λόγο αυτό έγιναν χειροκίνητα κάποιες μετρήσεις ταχύτητας. Δηλαδή το αυτοκίνητο διένυσε απόσταση 1 μέτρου και υπολογίστηκε χειροκίνητα (με τη βοήθεια του χρονομέτρου του κινητού τηλεφώνου) πόσο χρόνο χρειάζεται να διανύσει αυτή την απόσταση. Οι μετρήσεις εκτελέστηκαν 5 φορές για να προκύψει μια πιο ακριβής μέση τιμή και υπολογίστηκε η μέση ταχύτητα 0.36 m/sec. Στη συνέχεια με την χρήση του πακέτου ROS και του ORB SLAM2 μετρήθηκε η απόσταση μεταξύ ενός σημείου αναφοράς και του εμποδίου. Ο υπολογισμός έγινε με την οπτικοποίηση του 3D νέφους σημείων που

δημιουργήθηκε από το Kinect, και η μέτρηση έγινε με τη βοήθεια του Rviz με επιλογή των δύο σημείων. Η απόσταση μετρήθηκε ως 2.10m. Τέλος, με τη διαίρεση απόσταση/ταχύτητα υπολογίστηκε ότι το αυτοκίνητο πρέπει να λειτουργήσει για 6 sec και να σταματήσει πριν από το εμπόδιο χωρίς τη βοήθεια ραντάρ ή GPS [10]. Το επιτυχές αποτέλεσμα παρουσιάζεται στο βίντεο raspicar_with_orbslam2 (Εικόνα 22).



Εικόνα 22.Εφαρμογή αλγόριθμου ORB SLAM2 με δεδομένα εισόδου από το Microsoft Kinect

8. Συμπεράσματα

Στο πλαίσιο της παρούσας εργασίας μελετήθηκε η μέθοδος του Οπτικού ταυτόχρονου Εντοπισμού και Χαρτογράφησης (Visual SLAM) ενώ επίσης πραγματοποιήθηκε εφαρμογή δύο αλγοριθμικών υλοποιήσεων V-SLAM ανοικτού κώδικα σε δεδομένα που συλλέχθηκαν από διαφορετικούς αισθητήρες (μηχανή RGB και αισθητήρας βάθους)

Κατά την εκτέλεση των εφαρμογών, παρατηρήθηκε ότι είναι αρκετά χρονοβόρα η υλοποίηση του ORB SLAM2 σε εσωτερικό περιβάλλον με απλή κάμερα, πχ. κινητό τηλέφωνο. Αυτό οφείλεται στο ότι υπάρχει το πρόβλημα της κλίμακας (όπως αναλύθηκε προηγουμένως) με αποτέλεσμα πολλές φορές ο αλγόριθμος να μην εκτελείται σωστά. Για αυτό το λόγο χρειάστηκε να γίνουν πολλές δοκιμές με διαφορετικά βίντεο και παραμέτρους, πχ. αλλαγή αριθμού των πλαισίων. Επίσης στο χώρο όπου έγινε η λήψη προέκυψε πρόβλημα με τους τοίχους, το πάτωμα και γενικά με επιφάνειες οι οποίες ήταν ομογενείς σε υφή με αποτέλεσμα ο αλγόριθμος να μην εντοπίζει σημεία. Αντίθετα, σε βίντεο όπου η λήψη έγινε εντός του γραφείου και υπήρχαν διαφορετικές επιφάνειες, πχ ποντίκι, πληκτρολόγιο, οθόνη κ.λπ., το ORB SLAM2 ανταποκρίθηκε πιο αποτελεσματικά με πολύ λιγότερες δοκιμές. Μεγάλο ρόλο έπαιζε και ο φωτισμός καθώς και ο τρόπος λήψης του βίντεο. Για βέλτιστο αποτέλεσμα έπρεπε να υπάρχει συνεχής μετακίνηση (μπροστά, πίσω, αριστερά και δεξιά) με αργές κινήσεις στο κινητό τηλέφωνο, το οποίο έπρεπε να στρέφεται ώστε να αλλάζουν οι στροφές της κάμερας. Κάτι το οποίο πολλές φορές δεν είχε αποτέλεσμα λόγω των παρόμοιων επιφανειών. Επίσης χρειάζονται γνώσεις προγραμματισμού (κυρίως Python) και χρήσης περιβάλλοντος Linux / Ubuntu.

Τελικά έγινε δυνατόν να υλοποιηθεί ο αλγόριθμος (Εικ. 23), ενώ ο χρόνος που χρειάστηκε για να επεξεργαστεί την ακολουθία των 2357 εικόνων είναι:

Median tracking time: 0.0207236 sec

Mean tracking time: 0.0184151 sec

```
giannis@ubuntu: ~/Desktop/ORB_SLAM2-master
- Scale Levels: 8
- Scale Factor: 1.2
- Initial Fast Threshold: 20
- Minimum Fast Threshold: 7
-----
Start processing sequence ...
Images in the sequence: 2357

Framebuffer with requested attributes not available. Using available framebuffer
. You may see visual artifacts.New Map created with 123 points
-----

median tracking time: 0.0184151
mean tracking time: 0.0207236

Saving keyframe trajectory to KeyFrameTrajectory.txt ...

trajectory saved!

Saving map points to pointcloud.pcd ...

Map Points saved!
giannis@ubuntu:~/Desktop/ORB_SLAM2-master$
```

Εικόνα 23. Παράμετροι εκτέλεσης ακολουθίας εικόνων (βίντεο) από κινητό τηλέφωνο και χρόνοι επεξεργασίας

Όσον αφορά την εκτέλεση του αλγορίθμου με χρήση του Microsoft Kinect, το αποτέλεσμα ήταν επιτυχές με την πρώτη προσπάθεια και βρέθηκαν περισσότερα σημεία (ομολογίες – ORB χαρακτηριστικά), ενώ και το νέφος ήταν πιο πυκνό. Η λήψη έγινε στους ίδιους εσωτερικούς χώρους όπως και με το κινητό τηλέφωνο. Αυτό φαίνεται στα δύο συνημμένα βίντεο (video_kinect_orbslam2 και visual_pointcloud_ros_rviz_kinect) αλλά και στον ίδιο διάδρομο (Τμήμα Πληροφορικής του Braunschweig) που πραγματοποιήθηκε η λήψη μέσω του κινητού τηλεφώνου (λόγω όγκου δεδομένων δεν αποθηκεύτηκε το συγκεκριμένο βίντεο). Η ακρίβεια ήταν μεγαλύτερη και σε αυτό έπαιξε ρόλο ότι το Kinect διαθέτει εκτός από την RGB κάμερα και κάμερα βάθους, με αποτέλεσμα να υπάρχει και κλίμακα γνωστή, ώστε να γίνει τελικά δυνατή η χρήση του πακέτου ROS με το Raspberry Pi car. Η επεξεργασία των δεδομένων σε πραγματικό χρόνο ήταν πολύ ικανοποιητική, και η κίνηση της κάμερας ήταν πιο γρήγορη σε σχέση με την κίνηση το κινητού τηλεφώνου, χωρίς ωστόσο να χάνεται ο εντοπισμός με αποτέλεσμα να εκκινεί ξανά το thread του επανεντοπισμού. Ωστόσο συναντήθηκαν κάποιες δυσκολίες όσον αφορά την εγκατάσταση του Kinect στα Ubuntu καθώς και την σύνδεσή του με τον αλγόριθμο ORB SLAM2 και το ROS. Επίσης κατά την εκτέλεσή του, και όπως φαίνεται στο βίντεο video_kinect_orbslam2, δεν λαμβάνονται όλα τα δεδομένα–πακέτα τα οποία στέλνονται από το Kinect, λόγω της θύρας USB (σύνδεση Kinect μέσω θύρας USB 3.0 με το laptop). Αυτό έχει να κάνει μάλλον με το ότι το Kinect συνδέθηκε με το laptop εικονικά μέσω του VMware workstation. Εν κατακλείδι, η χρήση RGB-D κάμερας ενδείκνυται για

σενάρια εσωτερικού χώρου καθώς είναι πιο λειτουργική και παρέχει όλα τα δεδομένα (μετατοπίσεις, στροφές και κλίμακα) που χρειάζεται ο αλγόριθμος για να λειτουργήσει σωστά.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Shweta Mayekar, Ιούλιος 2018, <https://www.analyticsinsight.net/author/shweta/>
- [2] <https://www.analyticsinsight.net/future-ar-slam-technology-slam/>
- [3] Arthur Huletski, Dmitriy Kartashov, Kirill Krinkin, Νοέμβριος 2015, Evaluation of the modern Visual SLAM methods
- [4] Raul Mur-Artal, Juan D. Tardos, Ιανουάριος 2017, ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras
- [5] Rafael Munoz-Salinas, R. Medina-Carnicer, Φεβρουάριος 2019, UcoSLAM: Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers
- [6] https://github.com/raulmur/ORB_SLAM2
- [7] <https://vision.in.tum.de/data/datasets/rgbd-dataset/download>
- [8] <https://www.analyticsinsight.net/future-ar-slam-technology-slam/>
- [9] https://github.com/kafendt/ORB_SLAM2_Accessible
- [10] www.sunfounder.com
- [11] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html
- [12] <https://www.seeedstudio.com/blog/2019/01/25/add-a-laser-scanner-to-your-robot-the-key-to-enabling-slam/https://medium.com/maxst/types-of-slam-and-application-examples-e056bb8ed366>
- [13] <https://www.seeedstudio.com/blog/2019/01/25/add-a-laser-scanner-to-your-robot-the-key-to-enabling-slam/>
- [14] <https://link.springer.com/article/10.1186/1687-6180-2011-71>
- [15] <https://www.andreasjakl.com/basics-of-ar-slam-simultaneous-localization-and-mapping/>
- [16] <https://www.visiononline.org/blog-article.cfm/What-is-Visual-SLAM-Technology-and-What-is-it-Used-For/99>
- [17] https://en.wikipedia.org/wiki/Hessian_matrix
- [18] <https://link.springer.com/article/10.1007/s40903-015-0032-7>
- [19] Guanqun Bao, Απρίλιος 2014, On Simultaneous Localization and Mapping inside the Human Body (Body-SLAM)
- [20] Johanna Matthaei, Thomas Kruger, Stefan Nowak and Ulf Bestmann, Σεπτέμβριος 2013, Swarm Exploration of Unknown Areas on Mars Using SLAM
- [21] <https://www.cortica.com/>

- [22] <https://developers.google.com/ar>
- [23] <https://developers.google.com/ar/discover>
- [24] <https://www.microsoft.com/en-us/hololens>
- [25] https://techcrunch.com/2015/05/28/apple-metaio/?guccounter=1&guce_referrer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_cs=QwJsGie9UmZJbKUzSk90uQ
- [26] <https://techcrunch.com/2016/01/29/apple-further-its-vr-plans-with-acquisition-of-flyby-media-makers-of-tech-that-sees-the-world-around-you/>
- [27] https://www.web-formulas.com/Math_Formulas/Linear_Algebra/Definition_of_Identity_Matrix.aspx
- [28] Viola, P. and Jones, M., 2001. Robust real-time object detection. *International journal of computer vision*, 4(34-47), p.4.
- [29] Calonder, M., Lepetit, V., Strecha, C. and Fua, P., 2010, September. Brief: Binary robust independent elementary features. In *European conference on computer vision* (pp. 778-792)
- [30] Fischler, M.A. and Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), pp.381-395