ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

An Efficient Track Detection and Mapping System for Autonomous Driving Race car
Αποτελεσματικό σύστημα ανίχνευσης και χαρτογράφησης πίστας για αυτό-οδηγούμενο αγωνιστικό μονοθέσιο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευφημία Ε. Παναγιωτάκη

Επιβλέπων : Ευάγγελος Β. Χριστοφόρου
Διδακτικό και Ερευνητικό Προσωπικό (Δ.Ε.Π.)

Αθήνα, Οκτώβρης 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

An Efficient Track Detection and Mapping System for Autonomous Driving Race car
Αποτελεσματικό σύστημα ανίχνευσης και χαρτογράφησης πίστας για αυτό-οδηγούμενο αγωνιστικό μονοθέσιο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευφημία Ε. Παναγιωτάκη

Επιβλέπων : Ευάγγελος Β. Χριστοφόρου
Διδακτικό και Ερευνητικό Προσωπικό (Δ.Ε.Π.)

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9η Νοεμβρίου 2017.

.....	.....	.....
Ευάγγελος Χριστοφόρου Διδακτικό και Ερευνητικό Προσωπικό (Δ.Ε.Π.)	Σταύρος Παπαθανασίου Διδακτικό και Ερευνητικό Προσωπικό (Δ.Ε.Π.)	Luc Van Gool Computer Vision Lab – ETH Zürich

Αθήνα, Οκτώβρης 2017

.....

Ευφημία Ε. Παναγιωτάκη

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η συγκεκριμένη διπλωματική εργασία διεκπεραιώθηκε το εαρινό εξάμηνο του 2017 στο ETH Zürich (Swiss Federal Institute of Technology) στα πλαίσια του προγράμματος “Formula Student”, υπό την επίβλεψη του καθηγητή prof. Luc Van Gool από το Computer Vision Lab του D - ITET.

Τον Αύγουστο του 2017, η ομάδα Akademischer Motorsportverein Zürich (AMZ) Driverless, συμμετείχε στο διαγωνισμό Formula Student Driverless, στο Hockenheimring, στη Γερμανία, κερδίζοντας την πρώτη ($1^{η}$) θέση ανάμεσα σε 15 άλλες driverless ομάδες. Η AMZ Driverless είναι η μόνη ομάδα που τερμάτισε επιτυχώς όλα τα αγωνίσματα του διαγωνισμού, λαμβάνοντας την υψηλότερη βαθμολογία.

Η ομάδα ανέπτυξε ένα πλήρες αυτόνομο σύστημα, χρησιμοποιώντας σαν πλατφόρμα το πρωτότυπο μονοθέσιο που είχε κατασκευάσει το 2015. Η AMZ Driverless, αποτελείται από μεταπτυχιακούς φοιτητές του ETH Zürich, παλαιότερα μέλη της ομάδας AMZ και τη συγγραφέα, που συμμετείχε στο πρόγραμμα ως Invited Visiting student του ETH Zürich. Ο συνολικός και επιμέρους σχεδιασμός του αυτόνομου συστήματος είναι προσαρμοσμένος στους κανονισμούς του διαγωνισμού Formula Student Driverless.

Το σύστημα έχει βασιστεί στην ανεξάρτητη λειτουργία των αισθητήρων και των αλγορίθμων συλλογής δεδομένων, εξασφαλίζοντας την επιτυχία και αξιοπιστία του. Ο στόχος του perception pipeline είναι να εξάγει έναν ακριβή σημασιολογικό χάρτη της πίστας, να παρέχει ακριβές εκτιμήσεις ταχύτητας και θέσης του αυτοκινήτου και να εξασφαλίσει την ανεξαρτησία των αισθητήρων σε περίπτωση αποτυχίας κάποιου συστήματος. Οι αισθητήρες που αφορούν την αντίληψη του μονοθεσίου είναι ένας LiDAR αισθητήρας και ένας Visual Inertial αισθητήρας, ανεπτυγμένος από τη συγγραφέα, για την ανίχνευση των ορίων της πίστας και τη χαρτογράφηση της. Η διπλωματική αυτή είναι βασισμένη σε πραγματικά δεδομένα του visual pipeline από το αυτοκίνητο, κατά τη διάρκεια των δοκιμών.

Το visual pipeline παρέχει έναν ανεξάρτητο τρόπο ανίχνευσης κώνων, εξασφαλίζοντας επιτυχία ακόμη και αν ο LiDAR παρέχει εσφαλμένα δεδομένα. Για τον οπτικό αισθητήρα που κατασκευάσαμε, συγχρονίσαμε δύο monocular καμερες και ένα Inertial Navigation System (INS) προκειμένου να παράξουμε ένα ακριβές και ανεξάρτητο visual SLAM συστημα, από την αρχή. Το σύστημα έχει βελτιστοποιηθεί για large scale εφαρμογές και συνθήκες εξωτερικού περιβάλλοντος, πάντα σε συμφωνία με το συνολικό αυτόνομο σύστημα του αυτοκινήτου.

Προκειμένου να ανιχνεύσουμε τη θέση των κώνων, αναπτύξαμε έναν machine learning αλγόριθμο ανίχνευσης κώνων, που διακρίνει SLAM αναγνωρισμένα landmarks σε «κώνοι» και «άλλα». Ο συνδυασμός του αλγόριθμου, του stereo SLAM και του VI-odometry έχει αποδειχτεί ότι είναι αποτελεσματικός σε ταχεις γωνιακές τιμές και επιτυχημένο tracking κατορθώθηκε για πάνω από 180˚/s. Η εργασία αξιολογήθηκε με βάση την ακρίβεια της αναγνώρισης κώνων, του loop closing και της χάραξης τροχιάς.

Abstract

In August 2017, Akademischer Motorsportverein Zurich (AMZ) Driverless, participated in the Formula Student Driverless Competition in Hockenheimring, Germany, winning the 1st place overall, among 15 other driverless teams. AMZ Driverless is the only team that managed to successfully complete all the disciplines in the competition.

The team developed a fully autonomous driving vehicle, using as a platform the prototype that was built by AMZ Racing in 2015. Fluela is a lightweight, 4 wheel drive electric race car. AMZ Driverless consists of master students from ETH Zurich, former team members of the ETH Formula Student Project (AMZ Racing) and the author, an Invited Visiting student at ETH Z{\"u}rich, from the National Technical University of Athens, Greece. The design of the autonomous system is adapted to the rules and regulations of the Formula Student Driverless Competition.

The autonomous concept is focused on redundancy of the sensors and robustness of the autonomous system, in order to reassure success and reliability. The goal of the perception pipeline is to generate an accurate semantic map of the track, to provide accurate velocity and pose estimates and to assure redundancy in case of single mode failures. The perception sensors include a LiDAR and a self-developed Visual Inertial Sensor for boundaries detection and path planning. This Master Thesis, is based on real-time data of the Visual pipeline, from the actual vehicle during testing conducted throughout the season before and during the competition.

The visual pipeline provides a redundant way to detect cones, reassuring robustness in case of LiDAR failures. For the sensor, we synchronize two monocular cameras and an Inertial Navigation System (INS) in order to build an accurate and secluded visual SLAM system completely from scratch. The system is optimized for large scale applications and outdoors conditions, with respect to the vehicle's autonomous system.

To detect cones' positions, we implemented a state-of-the-art machine learning cone detection algorithm which classifies SLAM tracked landmarks to "cones" and "other". The combination of cone detection, stereo SLAM and VI-odometry has proven to be robust to fast angular rates and successful tracking was achieved for more than 180°/s. Our work is evaluated based on the precision and accuracy of: the cone detection, the loop closing, the generated map and the trajectory accuracy.

Ευχαριστίες

Η διπλωματική μου εργασία διεκπεραιώθηκε στο Formula Student Driverless project του πανεπιστημίου ETH Zürich. Πρωτού προχωρήσω περαιτέρω θα ήθελα να ευχαριστήσω τους ανθρώπους που συνετέλεσαν στην πραγματοποίησή της.

Κατ'αρχάς θα ήθελα να ευχαριστήσω θερμά τους επιβλέποντες καθηγητές μου, τον prof. Luc Van Gool και τον dr. Dengxin Dai από το Computer Vision Lab του ETH Zürich, για τη μεγάλη τους υποστήριξη στο εγχείρημα και στη διπλωματική αυτήν, σε όλη τη διάρκειά της αλλά και πριν καν ξεκινήσει. Θα ήθελα επίσης να ευχαριστήσω τον επιβλέποντα καθηγητή μου από το Εθνικό Μετσόβιο Πολυτεχνείο, τον κύριο Ευάγγελο Χριστοφόρου, για τη στήριξή του από την αρχή του εγχειρήματος.

Ένα ιδιαίτερο ευχαριστώ πηγαίνει στην ομάδα μου, Akademischer Motorsportverein Zürich (AMZ) Driverless. Κάθε μέλος της ομάδας του 2017, συνετέλεσε καθοριστικά στη διεξαγωγή της διπλωματικής και στην επιτυχία της ομάδας. Η AMZ Racing παρείχε όλα τα εργαλεία, τα υλικά και το μονοθέσιο που χρειαζόμασταν ώστε να μπορέσει το έργο μας να πραγματοποιηθεί.

Τέλος, είμαι ευγνώμων για την υποστήριξη που είχα από τον Αλέξανδρο Αθανασιάδη και την οικογένειά μου, που κατέστησαν δυνατό να μεταφερθώ στην Ελβετία, να γίνω μέλος της AMZ Driverless και να πραγματοποιήσω τη διπλωματική μου στο ETH Zürich, σαν Invited Visiting Student.

Acknowledgements

This thesis has been conducted at the Formula Student Driverless project of ETH Zürich, during the spring semester of 2017. Before I elaborate any further on this master thesis, I would like to express my gratitude to the people who have helped in realizing it.

First of all, I would like to sincerely thank my supervisors, prof. Luc Van Gool and dr. Dengxin Dai from the Computer Vision Lab, for their great support in the driverless project and this master thesis, during and before the thesis even started. I would also like thank my supervisor from my home University, the National Technical University of Athens (Greece), prof. Evangelos Christoforou, for his support at the beginning of the project.

A special thanks goes to my team, Akademischer Motorsportverein Zürich (AMZ) Driverless, Zürich, the Formula Student team of ETH Zürich. Each and every member of the 2017 project played a crucial role in the conduction of the thesis and the success of the season. AMZ Racing enabled this work, providing all the tools, materials and the vehicle required.

Finally, I am grateful for the support I had from Alex Athanasiadis and my family who made it possible to relocate to Switzerland, be part of AMZ Driverless and conduct my master thesis at ETH Zürich as an Invited Visiting Student.

Σ.Σ.:
Το παρόν αποτελεί την εκτενή περίληψη στα Ελληνικά, της διπλωματικής μου εργασίας που διεκπεραιώθηκε στο Πολυτεχνείο της Ζυρίχης, και δε θεωρείται πλήρης.
Το πρωτότυπο πλήρες κείμενο παρατίθεται στο τέλος στα Αγγλικά.

Ακρωνύμια και Συντομογραφίες

AMZ	Akademischer Motorsportverein Zürich, ομάδα Formula Student του πανεπιστημίου ETH Zürich
ASMS	Autonomous System Master Switch: Γενικός διακόπτης αυτονομου συστήματος
BPP	Business Plan presentation: Παρουσίαση επιχειρηματικού σχεδίου
DV	Driverless vehicles: αυτοκίνητα χωρίς οδηγό
DRS	Drag Reduction System: συστημα μείωσης δύναμης drag
EV	Electric Vehicles: ηλεκτρικά αυτοκίνητα
EBS	Emergency Brake System: Σύστημα φρεναρίσματος έκτακτης ανάγκης
FS	Formula Student
FSD	Formula Student Driverless
GPS	Global Positioning System: Σύστημα πλοήγησης
IMU	Inertial Measuring Unit: Σύσημα μέτρησης αδρανειακών ταχυτήτων
INS	Inertial Navigation System: IMU and GPS
IR	Infra – Red
LBP	Local Binary Patterns
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping: Ταυτόχρονος εντοπισμός και χαρτογράφηση
SM	Semantic Map: Σημασιολογικός Χάρτης
TOF	Time of Flight
VI	Visual Inertial
Computer Vision	Οπτική υπολογιστών
Detection	Εντοπισμός και εξαγωγή πιθανών features εικόνας που μας ενδιαφέρουν
Feature	Συγκεκριμένο στοιχείο που μας ενδιαφέρει σε μία εικόνα (2D)
Landmarks	Μοναδικά 3D στοιχεία που αναπαριστούν μια ξεχωριστή δομή
Master	Βασικός υπολογιστής αυτόνομου συστήματος
OpenCV	Open Source Computer Vision Library
Perception Pipeline	Αναφέρεται στην αντίληψη του μονοθεσίου, από τους αισθηρήρες μέχρι και τους αλγορίθμους επεξεργασίας δεδομένων
Slave	Δευτερος υπολογιστής αυτόνομου συστήματος, που χρησιμοποιείται μόνο από τους Computer Vision Αλγορίθμους
Smearing	Κάθετη έντονα φωτεινή γραμμή διαμήκους μίας εικόνας, προκαλούμενη από πολύ έντονη πηγή φωτός
Tracking	Χωρικός και χρονικός προσδιορισμός ενός landmark

Περιεχόμενα:

Κεφάλαιο 1: Εισαγωγή 19

1.1 Formula Student 19
1.2 Σκιαγράφιση πίστας 20
1.3 Περιγραφή Αυτονομου συστήματος 22
1.4 Βασικός σχεδιασμός 23
1.5 Το αυτοκίνητο: flüela 24

Κεφάλαιο 2: Visual Inertial Sensor System 25

2.1 Stereo Triangulation 25
2.2 Hardware 26
2.3 Wiring 27
2.4 Σχεδιασμός 28
2.5 Θέση 29
2.6 Συγχρονισμός 29
2.7 Πειραματικά αποτελέσματα 31
2.8 Επίλυση 32

Κεφάλαιο 3: Visual Cone Detection 34

3.1 Software – Hardware Αρχιτεκτονική 34
3.2 Detection Concept 34
3.3 Local Binary Patterns 35
3.4 Μεθοδολογία και δομή 35

Κεφάλαιο 4: Συμπεράσματα 36

4.1 Αποτελέσματα 36

Κεφάλαιο 1

Εισαγωγή

. Η χάραξη διαδρομής, ο εντοπισμός και η χαρτογράφηση στο autonomous driving απαιτούν ακριβή αντίληψη του περιβάλλοντος. Οι οπτικοί αισθητήρες αποτελούν τους βασικούς αισθητήρες αντίληψης για υπολογισμό απόστασης και εντοπισμό αντικειμένων. Σε σύγκριση με τα συστήματα δυναμικού εύρους, οι οπτικοί αισθητήρες παρέχουν αναγνώριση στοιχείων εικόνας, υπολογισμό απόστασης μεγάλου εύρους και εντοπισμό στο περιβάλλον.

Στόχος του εγχειρήματος ήταν η ανάπτυξη και τοποθέτηση ενός συγχρονισμένου, ακριβή και έμπιστου συστήματος που παρέχει σε πραγματικό χρόνο δυνατότητες SLAM. Το οπτικό SLAM σύστημα είναι ενσωματωμένο στο αυτό-οδηγούμενο μονοθέσιο και έχει αναπτυχθεί σε συμφωνία με το αυτόνομο σύστημα και τους κανονισμούς του διαγωνισμού FSD. Το σύστημά μας παρέχει ακριβής εκτιμήσεις θέσης και ταχύτητας και χαρτογράφηση. Προκειμένου να ικανοποιεί τις απαιτήσεις του περιβάλλοντος, το οπτικο αδρανειακό σύστημα βελτιστοποιήθηκε για τις διαφορετικές συνθήκες του περιβάλλοντος, όπως είναι οι καιρικές συνθήκες και η θέση του ηλίου σε σχέση με το μονοθέσιο. Το σύστημα είχε ως προαπαιτούμενο να λειτουργεί έμπιστα και με ασφάλεια σε υψηλές ταχύτητες, καταναλώνοντας περιορισμένη υπολογιστική ισχύ.

Στην αρχή του project, δοκιμάσαμε με επιτυχία την επίδοση των Visual SLAM αλγορίθμων [14],[12] χρησιμοποιώντας τον Visual – Inertial Skybotix VI Sensor [8] που ανέπτυξε το Autonomous Systems Lab (ASL) του ETH Zürich. Επιτύχαμε ακριβή χαρτογράφηση της πίστας και loop closure, ενώ οδηγούσαμε το μονοθέσιο στην πίστα. Ύστερα από αυτό το τεστ και επειδή ο αισθητήρας δεν ήταν πλέον διαθέσιμος αποφασίσαμε να κατασκευάσουμε έναν καινούριο βασισμένο στις δικές μας προδιαγραφές, ικανό να τρέξει τους οπτικούς SLAM αλγορίθμους και να ικανοποιεί τις απαιτήσεις του project. Για να εντοπίζουμε τις θέσεις των κώνων και να διακρίνουμε αριστερούς και δεξιούς για ακριβή χάραξη τροχιάς, αναπτύξαμε έναν Machine Learning αλγόριθμο αναγώρισης κώνων που διακρίνει και ταξινομεί SLAM tracked landmarks.

1.1 Formula Student

Πανεπιστημιακές ομάδες καλούνται να μελετήσουν, σχεδιάσουν, κατασκευάσουν και τέλος να οδηγήσουν και να αγωνιστούν με πρωτότυπα μονοθέσια τύπου formula [25]. Στα πλαίσια του διαγωνισμού αυτού, οι φοιτητές υποθέτουν ότι μία κατασκευαστική εταιρεία τους ανέθεσε την κατασκευή ενός

πρωτότυπου open – wheel αγωνιστικού αυτοκινήτου, για τους ερασιτέχνες οδηγούς αγώνων. Τα μονοθέσια θεωρούνται προιόντα παραγωγής και πρέπει να ικανοποιούν τις απαιτήσουν της πιθανής αγοράς. Κάθε ομάδα κρίνεται, βαθμολογείται και συγκρίνεται με τις υπόλοιπες για να καθοριστεί ο τελικός νικητής κάθε κατηγορίας από το σύνολο των αγωνισμάτων.

Ο διαγωνισμός χωρίζεται σε τρεις κατηγορίες: Θερμικά αυτοκίνητα (CV), Ηλεκτρικά αυτοκίνητα (EV) και Αυτόνομα αυτοκίνητα χωρίς οδηγό (DV). Όλα τα μονοθέσια οφείλουν να πληρούν συγκεκριμένους τεχνικούς περιορισμούς καθορισμένους από τους κανονισμούς κάθε κατηγορίας προκειμένου να επιτραπεί στις ομάδες να αγωνιστούν. Ο διαγωνισμός ξεκινά με μια σειρά τεχνικών ελέγχων προκειμένου να ελεγχθουν τα αυτοκίνητα ως προς την ασφάλειά τους και τη συμφωνία τους με τους κανονισμούς.

Το Formula Student Driverless, εισήχθει στο Formula Student τη σεζόν 2016 – 2017, με τον πρώτο διαγωνισμό να πραγματοποιείται στο FS Germany τον Αύγουστο του 2017. Οι ομάδες συμμετέχουν με μονοθέσια που είχαν κατασκευάσει πριν από δύο ή τρία χρόνια, ηλεκτρικά ή θερμικά. Τα μονοθέσια πρέπει να ακολουθούν τους τεχνικούς κανονισμούς του DV αλλά και των EV και CV, ανάλογα με την επιμέρους κατηγορία που ανήκουν.

1.2 Σκιαγράφηση πίστας

Όπως περιγράφεται στο "FSG DV Technical Specification 2017"
Στα δυναμικα αγωνίσματα η πίστα σκιαγραφείται με κώνους, όπως φαίνεται στα Figure 1.1, 1.2, 1.3, 1.4. Τα αριστερά όρια αποτελούνται από μπλε κώνους, ενώ τα δεξιά όρια από κίτρινους κώνους. Η μέγιστη απόσταση μεταξύ δύο κώνων είναι 5m. Στις στροφές η απόστασή τους μειώνεται για καλύτερη ένδειξη χάραξης.

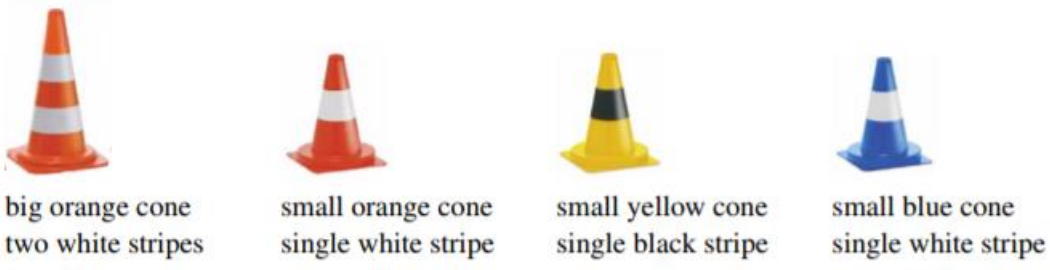


Figure 1.1: FSD Cones

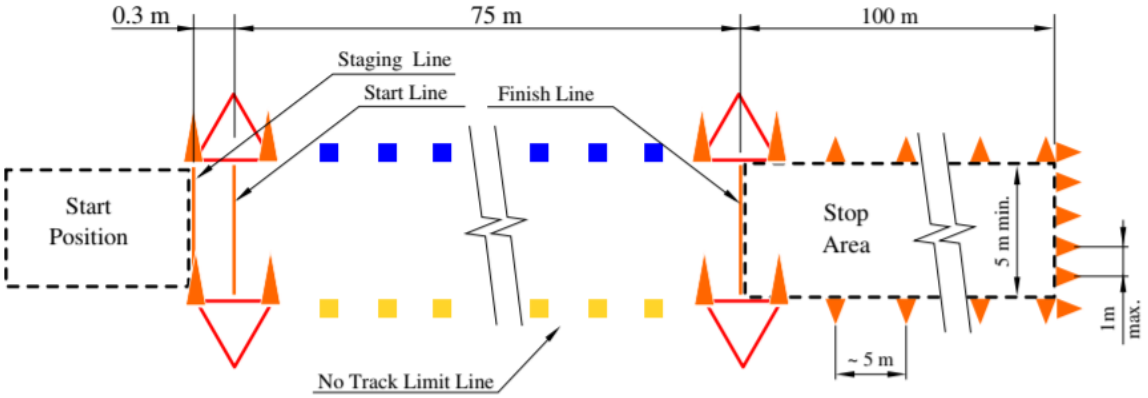


Figure 1.2: FSD Αγωνισμα Επιταχυνσης

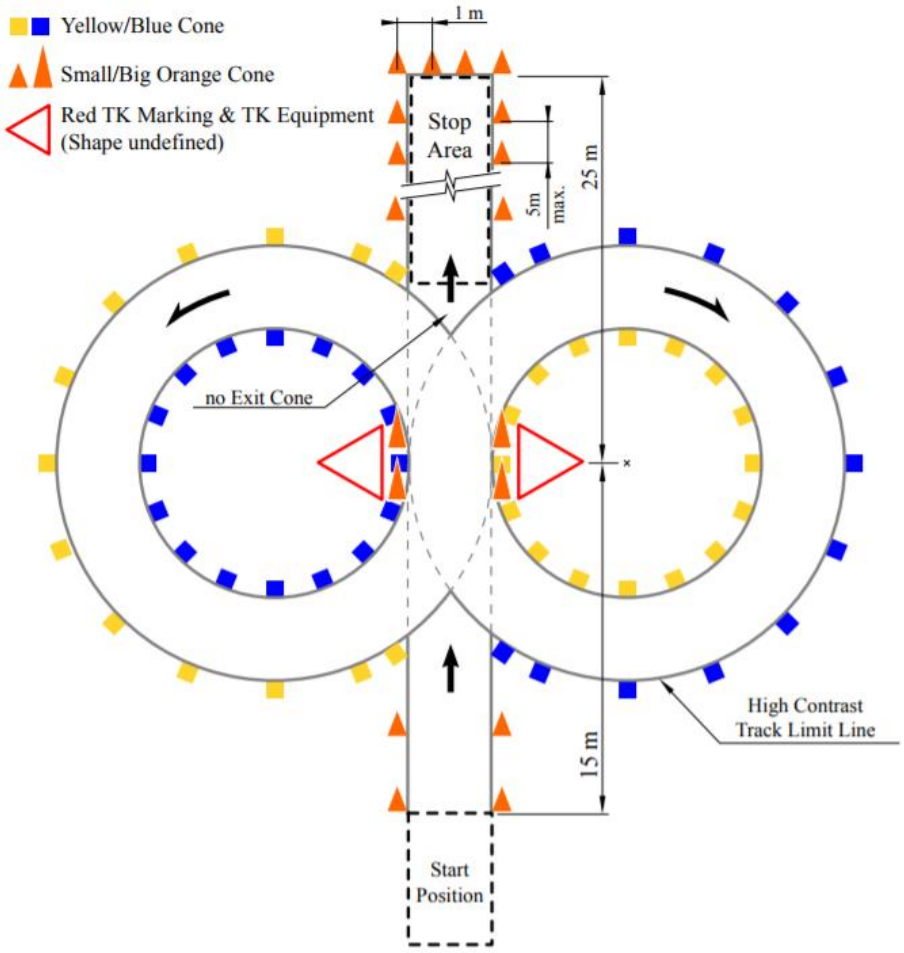


Figure 1.3: FSD SkidPad

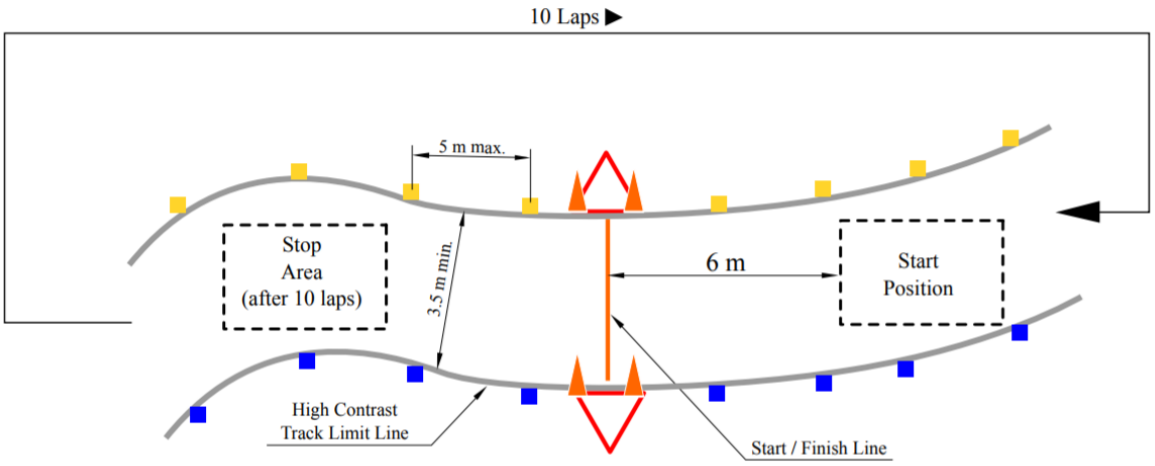
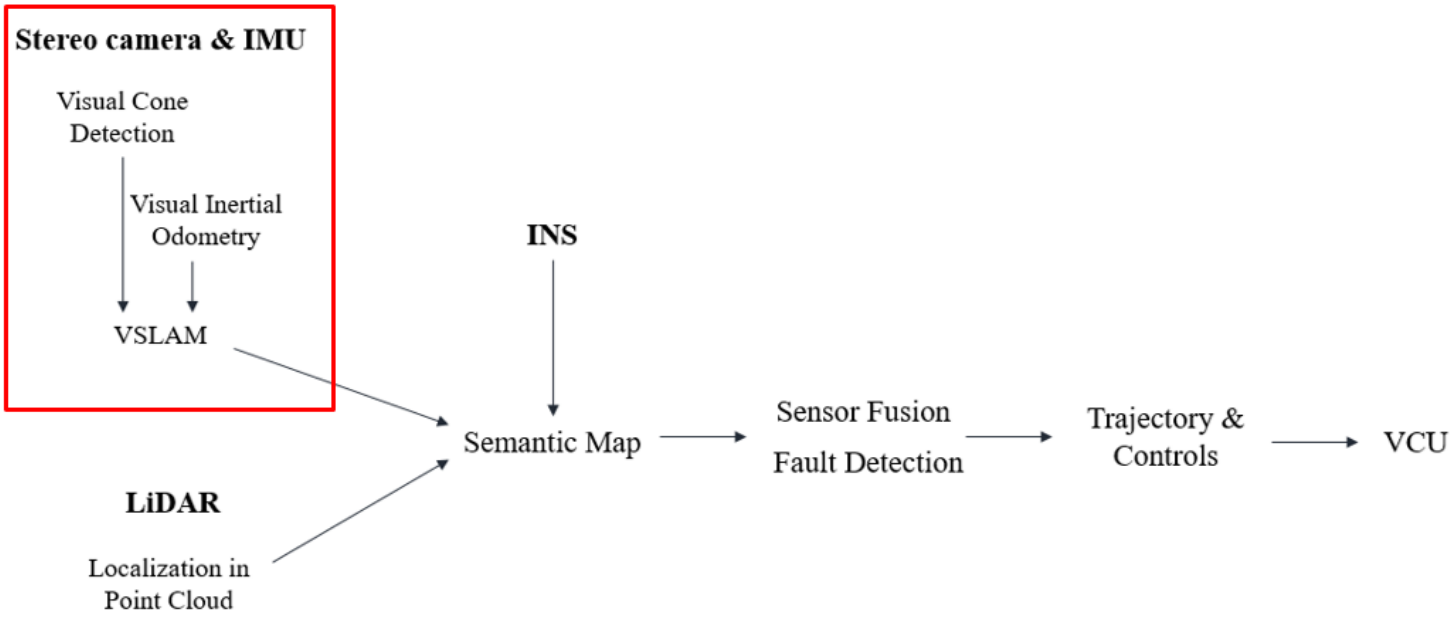


Figure 1.4: FSD Trackdrive

1.3 Περιγραφή Αυτόνομου Συστήματος

Στόχος του Αυτόνομου Συστήματος είναι η κατασκευή ενός αξιόπιστου και ανταγωνιστικού συστήματος προκειμένου να αυξήσουμε τους συνολικούς βαθμούς στο FSD. Αναπτύξαμε ένα αυτόνομο αγωνιστικό μονοθέσιο, δίνοντας ιδιαίτερη προσοχή στους αλγορίθμους, στους αισθητήρες, στην υπολογιστική ισχυ και στους απαιτούμενους actuators. Για να εξασφαλίσουμε την αξιοπιστία του αυτοκινήτου και του αυτόνομου συστήματος, ξεκινήσαμε να κάνουμε δοκιμές τέσσερεις μήνες πριν το διαγωνισμό. Το μονοθέσιο έχει ένα LiDAR, ένα stereo visual inertial system, ένα INS και έναν αισθητήρα μέτρησης της απόλυτης ταχύτητας του μονοθεσίου [10]. Το σύστημα τρέχει σε δύο ενσωματωμένες υπολογιστικές μονάδες σε διάταξη Master – Slave, που τρέχουν ROS (Robot Operating System) Indigo και Ubuntu 14.04. Ο υπολογιστής Slave χρησιμοποιείται αποκλειστικά από τους αλγορίθμους και τους αισθητήρες οπτικής.



Προκειμένου να επιτύχουμε υψηλές ταχύτητες όταν το μονοθέσιο βρίσκεται σε αυτόνομη λειτουργία, πρέπει να έχουμε προηγούμενη γνώση πίστας τουλάχιστον 2 δευτερολέπτων και ο ορίζοντας αντίληψης να ξεπερνά το εύρος των αισθητήρων αντίδρασης. Το αυτοκίνητο οδηγεί προσεκτικά στους πρώτους γύρους για να εξάγει και να δημιουργήσει τον σημασιολογικό χάρτη της πίστας («Discovery Mode») και προκειμένου στη συνέχεια να αυξήσει ταχύτητα και να αγωνιστεί μέσα σε αυτήν («Race Mode»).

Ο σχεδιασμός του συστήματος αντίληψης έγινε με βάση τις προαναφερόμενες απαιτήσεις του real – time racing και με βάση την περιορισμένη υπολογιστική ισχύ που είχαμε στη διάθεσή μας.

Στη διπλωματική αυτή εστιάζουμε στο Visual Pipeline του αυτονομου συστηματος

1.4 Βασικός Σχεδιασμός

Πιο λεπτομερώς, ο στόχος μας είναι να δημιουργήσουμε ένα αξιόπιστο σύστημα οπτικής το οποίο να χρησιμοποιεί μία συγχρονισμένη stereo camera με ένα IMU και τις μετρήσεις αυτών σαν είσοδο για να εξάγει διακεκριμένα τους δεξιούς και τους αριστερούς κώνους, εκτιμήσεις ταχύτητας και έναν 3D χάρτη από landmarks. Η έξοδος χρησιμοποιείται από το LiDAR ως σημείο αναφοράς για εξακρίβωση ή ως ένας ξεχωριστός σημασιολογικός χάρτης.

ORBSLAM [14] είναι ένας stereo SLAM αλγοριθμος ο οποίος χρησιμοποιείται σε συνδυασμό με ένα μονο οπτικό VI συστημα, ROVIO [12], με πρόσβαση σε εξειδικευμένα επιταχυνσιόμετρα και γυροσκόπια. Σαν προηγούμενο του ORBSLAM, χρησιμοποιούμε έναν machine learning cone detection αλγόριθμο, προκειμένου να εξάγουμε τις θέσεις των κώνων. Ο αλγόριθμος είναι βασισμένος σε boosted cascade of weak classifiers [19] και παρέχει εξαιρετικό συμβιβασμό μεταξύ ακρίβειας και χρόνου εκτέλεσης.

Η δημιουργία του αισθητήρα από την αρχή, μας επέτρεψε να επιλέξουμε το hardware της κάμερας, το baseline της stereo διάταξης, το συγχρονισμό του αισθητήρα και τη χρονοτοποθέτηση των μετρήσεων. Ο Visual Inertial Αισθητήρας αποτελείται από δύο monochrome and monocular καμερες και ένα Inertial Navigation System (IMU and GPS). Το GPS χρησιμοποιείται για το συγχρονισμό και το time stamp των μετρήσεων.

Η διάκριση δεξιών και αριστερών κώνων επιτυγχάνεται λόγω του χαρακτηριστικού pattern των κώνων σε μονόχρωμη κλίμακα και με την προηγούμενη γνώση των χαρακτηριστικών τους.

1.5 Το αυτοκίνητο: flüela

Η flüela κατασκευάστηκε το 2015 από την AMZ Racing. Ενσωματώνει πλήρες αεροδυναμικό πακέτο με Drag Reduction System (DRS), υψηλή ροπή σε κάθε τροχό και ελαφρύ σχεδιασμό (188.2 kg). Το μονοθέσιο είναι σχεδιασμένο να τρέχει σε κλειστές πίστες σε διαφορετικές ταχύτητες.

Kerb weight	188.2 kg
Length	2870 mm
Width	1438 mm
Height	1139 mm
Maximal Speed	119 kph
Max Wheel Torque	374 Nm
Peak Power	148 kW

Κεφάλαιο 2

Visual Inertial Sensor System

2.1 Stereo Triangulation

Αναλύοντας και εξετάζοντας τις διάφορες οπτικές τεχνικές υπολογισμού απόστασης, καταλήξαμε στις: Stereo Triangulation, Structured Light και Time of Flight [5]. Για την εφαρμογή μας, πιο κατάλληλη κρίθηκε η τεχνική Stereo Triangulation.

Μπορούμε να εξάγουμε ένα σημείο στο 3D χώρο, δεδομένης της προβολής του στις εικόνες, χρησιμοποιώντας σαν προηγούμενο τα: disparity map, focal length και cameras' calibration parameters. Υποθέτοντας ότι έχουμε μια διάταξη Stereo camera, η απόσταση του αντικειμένου υπολογίζεται από την σχετική απόσταση των δύο φακών. Η stereo camera δημιουργεί μία ψευδαίσθηση βάθους χρησιμοποιώντας δύο εικόνες που έχουν παρθεί από λίγο διαφορετικές θέσεις [21]. Η τεχνική αυτή είναι αξιόπιστη για εξωτερικά περιβάλλοντα και ο ORBSLAM [20], πραγματοποιεί landmark tracking χρησιμοποιώντας αυτή τη μέθοδο.

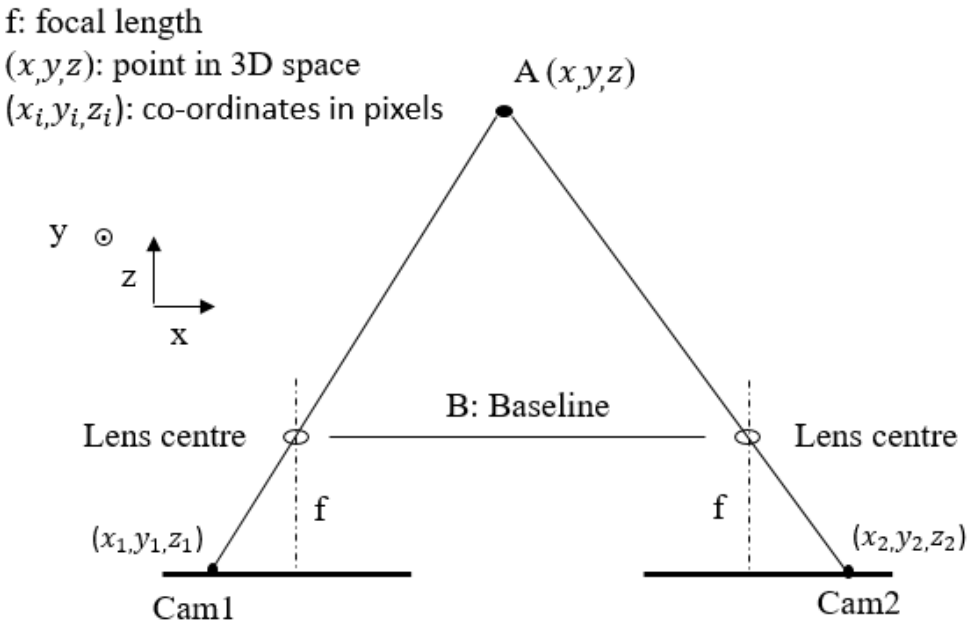


Figure 2.1: Γεωμετρικό μοντέλο stereo triangulation

Η απόσταση εστίασης υπολογίζεται ύστερα από τη βαθμονόμηση των καμερών. Πριν τον υπολογισμό, τα x1, x2 πρέπει να μετατραπούν στη κλίμακα μέτρησης, πολλαπλασιάζοντάς τα με τον scaling factor του αισθητήρα [21] .

2.2 Hardware

Ύστερα από διάφορες δοκιμές, καταλήξαμε ότι ένα stereo setup με δύο monocular καμερες είναι η καλύτερη επιλογή για την εφαρμογή μας. Τα πλεονεκτήματα ενός αυτοσχέδιου visual inertial sensor είναι κυρίως η σχεδιαστική ελευθερία που μας επιτρέπει να έχουμε:

- Stereo Camera setup: μεγάλο baseline, με βάση τους κανονισμούς του διαγωνισμού
- Έλεγχο στο hardware των καμερών και τις απαιτήσεις ισχύος
- Ανεξαρτησία στην επιλογή των: Ανάλυση καμερών, Μέθοδος ανάκτησης εικόνων, Frame Rate, Χρώμα
- Επιλογές triggering
- Η επιλογή του hardware του IMU έγινε με βάση την ακρίβεια του, την αξιοπιστία του και τις απαιτήσεις ισχύος
- Έλεγχο στις επιλογές των frame capturing και exposure
- Time stamping των μετρήσεων και συγχρονισμός συστημάτων

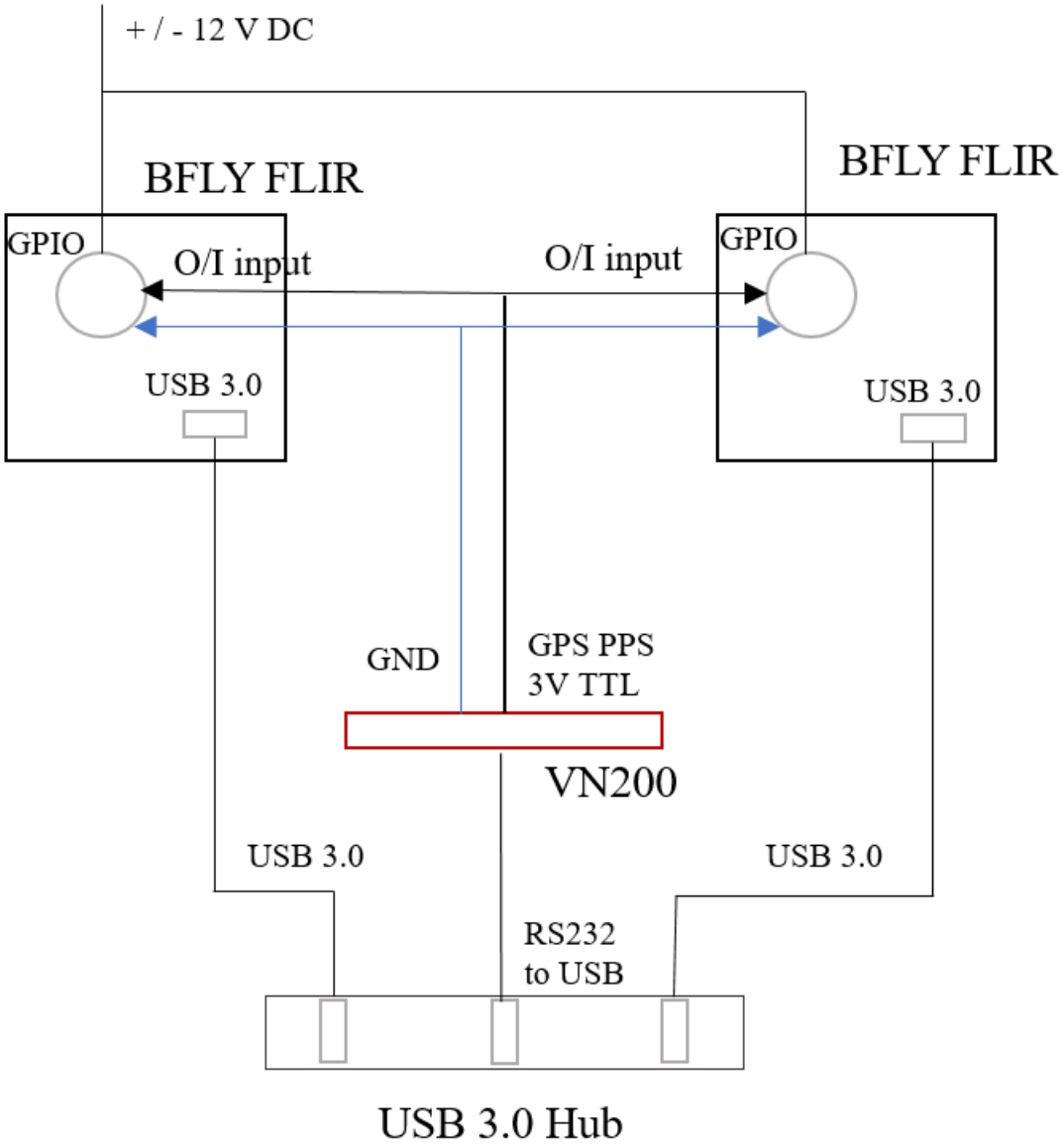
Οι κάμερες που επιλέγχθηκαν είναι δύο μονόχρωμες, monocular Blackfly PointGrey κάμερες. Προκειμένου να επιτύχουμε triggering, συγχρονισμό και timestamp με σημείο αναφοράς τις μετρήσεις του GPS, αποφασίσαμε να χρησιμοποιήσουμε το VN 200 Vectornav INS, ένα 6 – αξονικό INS υψηλής ακρίβειας.

Resolution	648x488
Interface	USB 3.1
Frame Rate	84 fps
Chroma	Mono
Sensor Type	CCD
Readout Method	Global Shutter
Triggering	Hardware and Software
Triggering Modes	Standard, bulb, low smear, overlapped, multishot

Blackfly PointGrey 0.3 MP Mono USB3

VectorNav VN200

2.3 Wiring



Vectornav VN200 :
Inertial Navigation System (IMU and GPS)

BFLY FLIR:
648x488, 84 fps, CCD, Monochrome, Global Shutter

2.4 Σχεδιασμός

Το τελικό σχέδιο της θήκης του αισθητήρα έχει βελτιωθεί προκειμένου να χρειάζεται τον ελάχιστο δυνατό χώρο, ώστε να είναι σε συμφωνία με τους κανονισμούς του FSD. Η θήκη είναι 3D printed, από πλαστικό και ιδιαίτερα προσεκτικά καλυμμένη ωστε να είναι αδιάβροχη.
Η θήκη σχεδιάστηκε από την AMZ Driverless για το συγκεκριμένο project.
Baseline: 102 mm

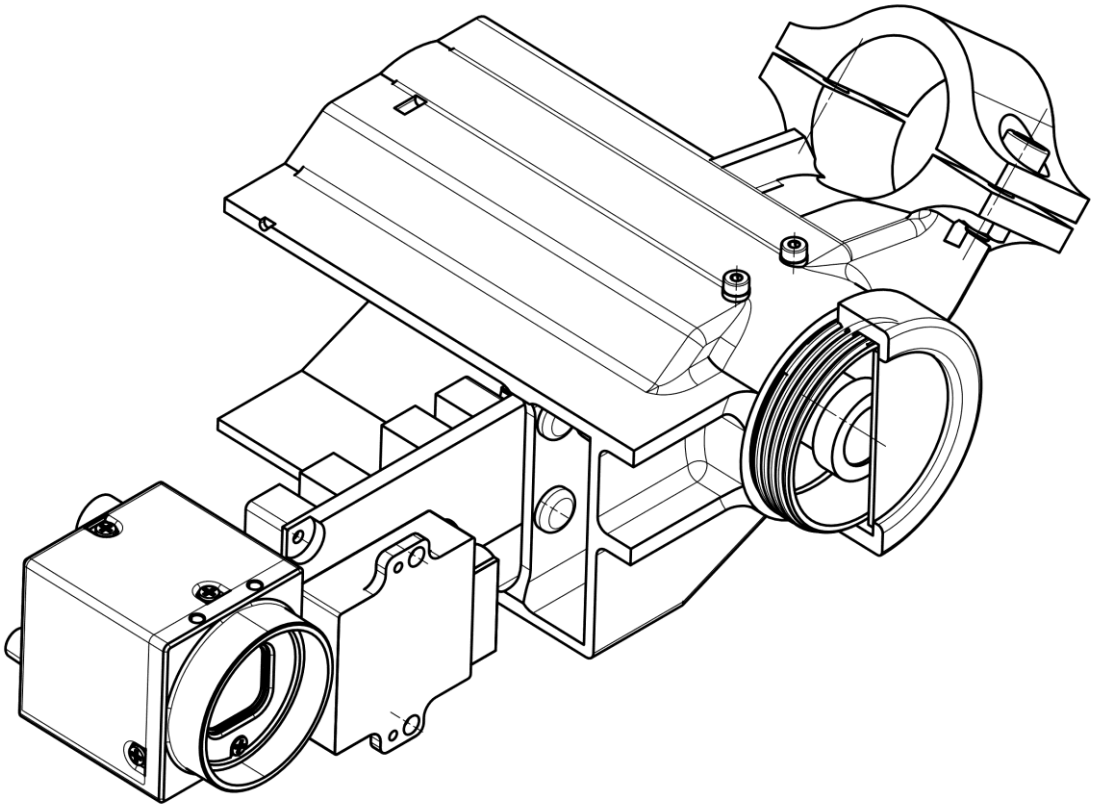


Figure 2.2: 3D printed water-resistant vision case, designed by AMZ Racing

2.5 Θέση

Το σύστημα είναι τοποθετημένο πάνω από το κράνος του οδηγού, στο υψηλότερο δυνατό σημείο, στο main roll hoop. Στη θέση αυτή η επικάλυψη κώνων είναι ελάχιστη, διευκολύνοντας τον αλγόριθμο εντοπισμού. Το αρνητικό της θέσης αυτής είναι ότι το main roll hoop απορροφά όλους τους κραδασμούς, αυξάνοντας το θόρυβο στις μετρήσεις του IMU. Για να αντιμετωπίσουμε το πρόβλημα αυτό, εφαρμόσαμε χαμηλοπερατό φίλτρο στις μετρήσεις και απομονώσαμε κατάλληλα τις πηγές εξωτερικού θορύβου (αντλίες).

2.6 Συγχρονισμός

Για το συγχρονισμό του stereo συστήματος και τη χρονική σήμανση των μετρήσεων είναι απαραίτητο το triggering των αισθητήρων. Το πρόβλημα που δημιουργείται είναι ότι προσθέτει επιπλέον καθυστερήσεις στις μετρήσεις. Οι κάμερες ενεργοποιούνται κάθε ένα δευτερόλεπτο, ανανεώνοντας το χρονικό σημείο αναφοράς του GPS για τη χρονική σήμανση των μετρήσεων. Ύστερα από το triggering, ξεκινούν να λαμβάνουν έναν προκαθορισμένο αριθμό εικόνων σε automatic exposure.

Για το triggering των καμερών και των συγχρονισμό των αισθητήρων, επιλέγουμε hardware triggering από το GPS's Pulse per Second hardware line [28]. Πρόκειται για σήμα εξόδου σε TTL τάση (3V), συνδεδεμένο απευθείας με το PPS Pin στο GPS receiver.

Οι κάμερες μέσω της εισόδου GPIO opto-isolated input, δέχονται το σήμα PPS και ξεκινούν να λαμβάνουν και να μεταδίδουν έναν προκαθορισμένο αριθμό εικόνων [3]. Κάθε κάμερα έχει το δικό της εσωτερικό ρολόι και για να επιτύχουμε τέλειο συγχρονισμό χρειάζεται να μετρήσουμε τις καθυστερήσεις: propagation and clearing delays.

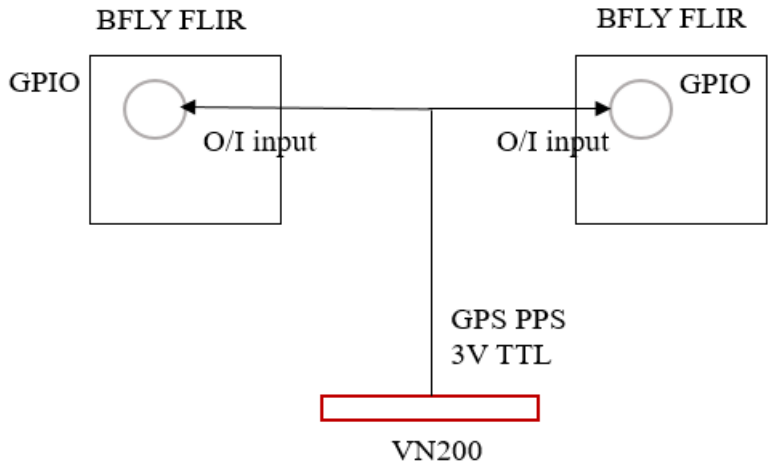


Figure 2.3: Triggering Wiring Diagram

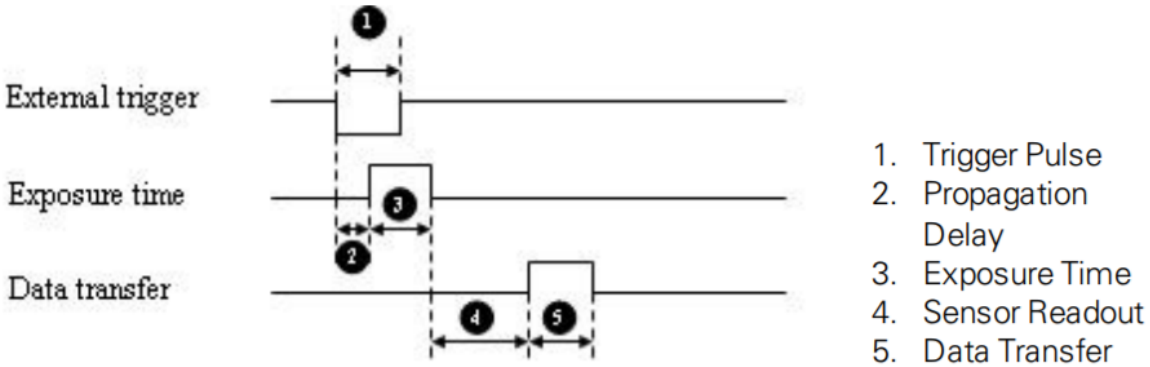


Figure 2.4: Triggering Delays (Source: [3])

Timestamp = GPStime + Propagation Delay

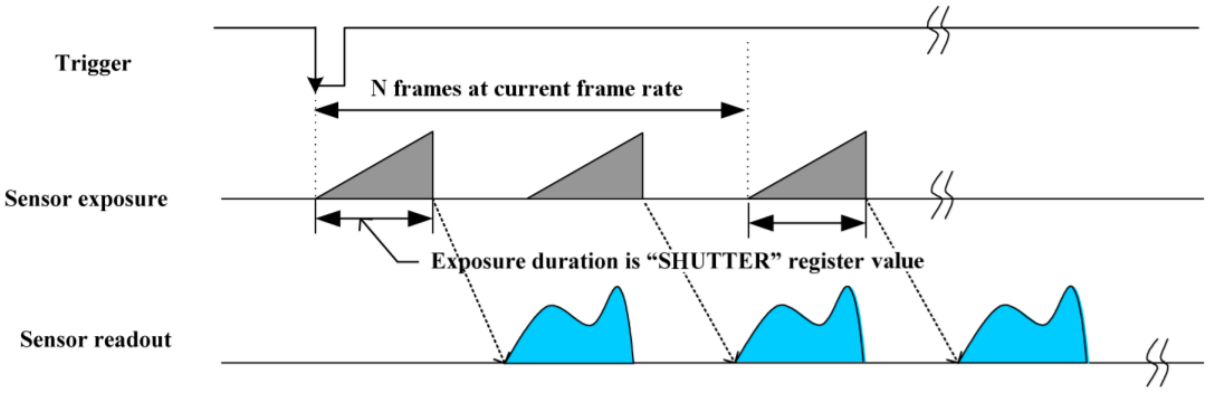
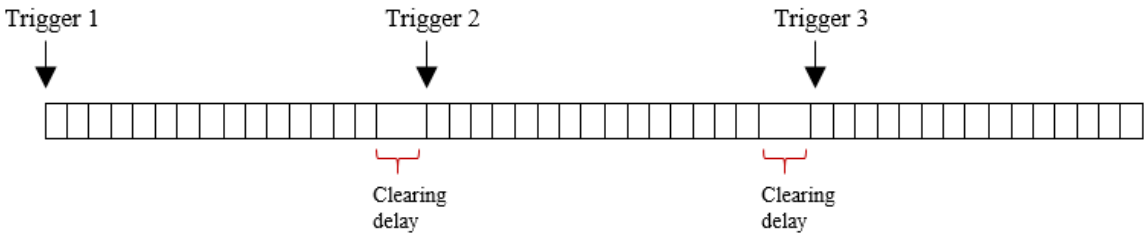


Figure 2.5: Multi shot triggering (Source: [3])

Συμπεριφορά καμερών ανάμεσα στα triggers:

Ανάλογα το frame rate, η κάμερα καθαρίζει τις τροφοδοσίες από τον αισθητήρα σε ρυθμό ρολογιού οριζόντιου εικονοστοιχείου. Αυτό συμβαίνει μετά την τελευταία λήψη εικόνας (shutter integration) και μέχρι να ληφθεί το επόμενο trigger [3]. Αυτό προκαλεί μία χρονική καθυστέρηση αφιερωμένη σε αυτήν την λειτουργία, η οποία προκαλεί την καθυστέρηση μεταξύ της τελευταίας λήψης και της επόμενης.



2.7 Πειραματικά αποτελέσματα

Σε multi - shot triggering, ανάλογα τις καιρικές συνθήκες, η ταχύτητα κλείστρου δεν παραμένει σταθερή σε μία τιμή και δημιουργείται ολίσθηση χρόνου, παρά την μετατροπή ShutterSpeed = 1/N (Appendix B).

Στα παρακάτω διαγράμματα, οι κάθετες μπλε γραμμές αναπαριστούν τα καρέ που λήφθηκαν σε μία καταγραφή. Η πρώτη οριζόντια καρέ είναι οι εικόνες που λήφθηκαν από την αριστερή κάμερα ενώ η δεύτερη αντιστοιχεί στις εικόνες που λήφθηκαν από τη δεξιά κάμερα. Η τρίτη γραμμή αντιστοιχεί στις μετρήσεις του INS. Οι κάθετες γραμμές τοποθετήθηκαν αναλόγα με το χρόνο που λήφθηκαν οι μετρήσεις. Μη σταθερά κενά μεταφράζονται ως χρονικές καθυστερήσεις μεταξύ των καρέ, ενώ κενά για πάνω από 0.9 δευτερόλεπτα μεταφράζονται ως χαμένα triggers.

Σε χαμηλό φωτισμό, παρατηρείται ότι αυξάνεται η ταχύτητα κλείστρου με αποτέλεσμα να χάνεται το επόμενο trigger. Σε πολύ φωτεινές συνθήκες, παρατηρείται μείωση της ταχύτητας κλείστρου προκαλώντας μεγάλα κενά μέχρι να ληφθεί το επόμενο trigger. Για την εφαρμογή μας, πρέπει να έχουμε ακριβή γνώση των καθυστερήσεων ώστε να μπορούμε να συγχρονίσουμε τις μετρήσεις.

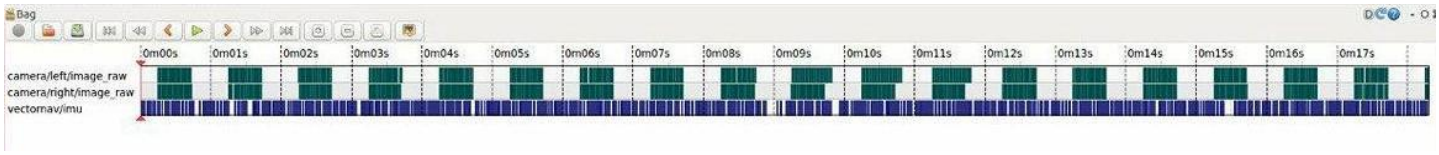


Figure 2.6: 84 fps σε συνθήκες χαμηλού φωτισμού

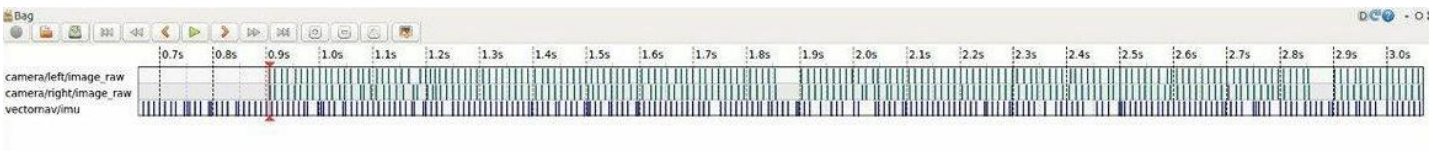


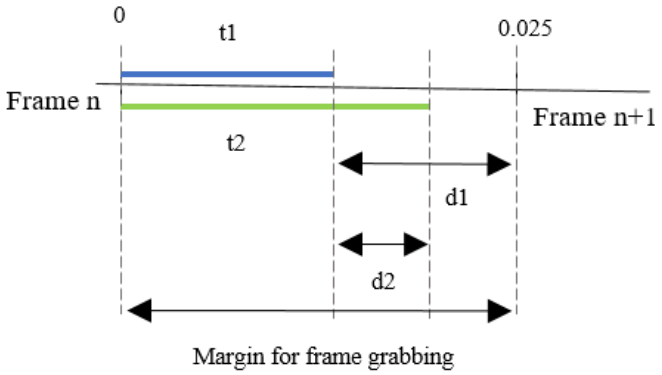
Figure 2.7: 80 fps σε συνθήκες έντονου φωτισμού

2.8 Επίλυση

Στόχος μας είναι να αναπτύξουμε ένα αξιόπιστο σύστημα ανεξάρτητο από τις περιβαλλοντικές συνθήκες φωτισμού. Ασταθείς και ασυγχρόνιστες μετρήσεις αισθητήρων, υποδεικνύουν απρόβρεπτη συμπεριφορά αισθητήρων και χρονική ολίσθηση. Στο project μας είναι απαραίτητο να γνωρίζουμε τις καθυστερήσεις και το εσωτερικό ρολόι των καμερών για ακριβές timestamping.

Για να επιλύσουμε το ζήτημα αφιερώσαμε αρκετές ημέρες δοκιμών σε διαφορετικές συνθήκες περιβάλλοντος, αλλάζοντας τις εσωτερικές παραμέτρους και τους καταχωρητές των καμερών μέχρις ότου να βρούμε το κατάλληλο exposure time και frame rate που ελαχιστοποίησαν τις καθυστερήσεις μεταξύ των καρέ για κάθε διαφορετικό σενάριο.

Η λύση που εφαρμόστηκε ήταν η σταθεροποίηση του χρονικού περιθωρίου ανοίγματος του shutter κρατώντας το exposure time σε αυτόματο mode. Μετρήθηκαν σε συνθήκες ακραίου φωτισμού (χαμηλού και υψηλού), τα κατάλληλα fps, shutter speed και exposure time ώστε να λαμβάνουμε χρήσιμες εικόνες. Παίρνοντας το μέσο όρο των μετρήσεων, διαμορφώσαμε καταλληλα τα όρια.



Automatic exposure depending on the lighting conditions
t1: shutter speed in case of bright lighting (e.g.)
t2: shutter speed in case of dim lighting (e.g.)
d1,d2: delays until next frame which don't affect synchronization or timestamping
Frames are timestamped at the beginning of exposure

Stereo Configuration:

Trigger	ON
Trigger Source	GPIO 0
Trigger Mode	15 (Multishot)
Exposure min	0
Exposure max	2.5
Shutter Speed	0 – 0.025
Mode	Automatic exposure
Parameter	81
Frame Rate	40

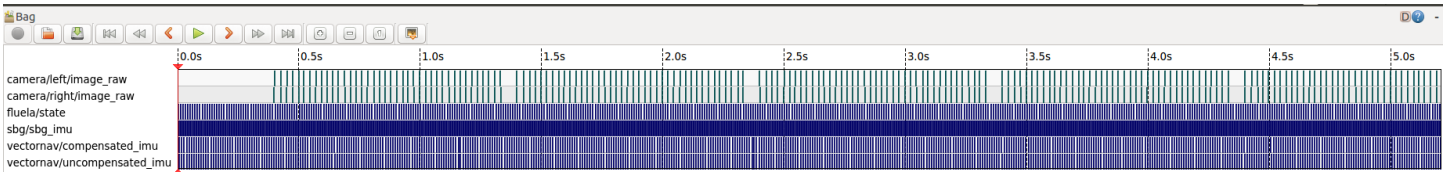


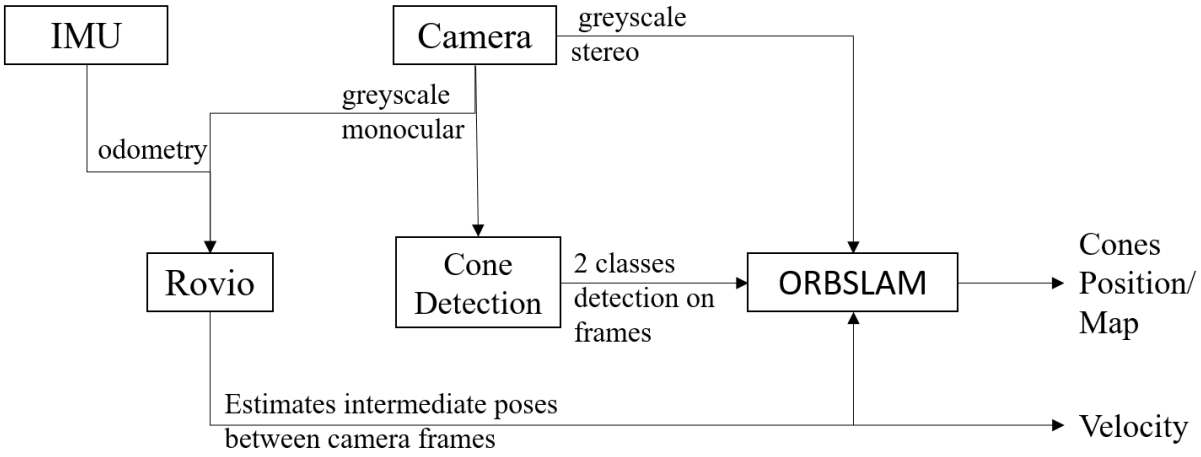
Figure 2.8: Αποτελέσματα συγχρονισμού

Κεφάλαιο 3

Visual Cone Detection

3.1 Software – Hardware αρχιτεκτονική

Οι βελτιστοποιήσεις στο hardware, όπως περιγράφηκαν στο προηγούμενο κεφάλαιο, αποσκοπούν στην αύξηση της απόδοσης του αλγορίθμου αναγνώρισης κώνων. Στη διάταξη Stereo κάμερα – IMU χρησιμοποιούμε μόνο την αριστερή mono camera για την εύρεση κώνων. Ο αλγόριθμος ORBSLAM [20], χρησιμοποιεί το stereo setup για να εντοπίζει landmarks χρησιμοποιώντας την τεχνική stereo triangulation. Ο αλγόριθμος Cone Detection διαχωρίζει τα landmarks σε «κώνοι» και «άλλα». Ο τρίτος αλγόριθμος του vision pipeline, ROVIO, χρησιμοποιεί τη mono camera και τα επιταχυνσιόμετρα και τα γυροσκόπια του IMU. Προσθέτοντας οδομετρικές πληροφορίες, γνωρίζουμε την απόλυτη γωνιακή και γραμμική ταχύτητα, επιτυγχάνοντας ακρίβεια στις μετρήσεις. Η αναγνώριση είναι απαραίτητη για τη χαρτογράφηση πίστας.



3.2 Detection Concept

Για την αναγνώριση κώνων, ο στόχος μας είναι να διακρίνουμε αριστερούς και δεξιούς κώνους στην πίστα. Όπως περιγράφεται στο πρώτο κεφάλαιο χρησιμοποιούμε μπλέ κώνους με μία άσπρη οριζόντια γραμμή στη μέση και κίτρινους κώνους με μαύρη οριζόντια γραμμή στη μέση.

Χρησιμοποιούμε την πληροφορία από εικόνες στη γκρι κλιμακα λόγω των απαιτήσεων του συνολικού μας συστήματος. Παρόλο που δεν έχουμε τη χρωματική πληροφορία, ο εντοπισμός επιτυγχάνεται λόγω του χαρακτηριστικού πατρόν της χρωματικής έντασης του αντικειμένου. Το αποτέλεσμα του ασπρόμαυρου πατρόν είναι ένας ανοιχτόχρωμος κώνος με μία μαύρη γραμμή και ένας σκούρος κωνως με μία λευκή γραμμή. Το πατρόν αυτό ειναι αρκετά χαρακτηριστικό για τον αλγόριθμο και ανεξάρτητο από τις αλλαγές φωτεινότητας
.

3.3 Local Binary Patterns

Τα Local Binary Patterns, από Ojala et al. [15], είναι ένας περιγραφέας χαμηλού κόστους που χρησιμοποιείται για ταξινόμηση. Περιγράφει τη σχέση μεταξύ ενός εικονοστοιχείου με τα γειτονικά του σε συγκεκριμένη ακτινική περιοχή. Είναι ιδανικό για εικόνες γκρι κλιμακας και αναγνώριση πατρόν.

Τα αντικείμενα που θέλουμε να εντοπίσουμε είναι ακέραια και αξονικά συμμετρικά. Τα features εξάγονται από τα LBP και έχουν χαμηλή υπολογιστική πολυπλοκότητα. Επιλέξαμε LBP επειδή:

- Απαιτούν χαμηλή υπολογιστική ισχύ και ταχύτητα
- Η εικονική αναπαράσταση των LBP δεν επηρεάζεται από τις διαφορές στη φωτεινότητα, καθώς χρησιμοποιεί τη σχετική διαφορά της χρωματικής έντασης των εικονοστοιχείων

3.4 Μεθοδολογία και δομή

A. Data generation:

Τα training data, εξήχθησαν από 80 χιλιόμετρα καταγεγραμμένων δεδομένων κατά τη διάρκεια των δοκιμών. Για κάθε σετ θετικών και αρνητικών εικόνων, δοκιμάσαμε την απόδοση του εντοπισμού προκειμένου να μειώσουμε τις λανθασμένες προβλέψεις. Στο σύνολο είχαμε:

- Positive data: 4000 εικόνες για κάθε ξεχωριστό είδος κώνου που θέλαμε να αναγνωρίσουμε
- Negative data: 10000 εικόνες για κάθε είδος με στοιχεία που μπορεί να προκύψουν και να αναγνωριστούν λανθασμένα ως κώνοι (πχ. ουρανός, οριζόντια πατρόν, αφίσες, ανθρωποι κτλπ)

B. Annotation:

Χρησιμοποιώντας το εργαλείο για εξαγωγή στοιχείων εικόνας του OpenCV, επιλέξαμε με το χέρι σε όλες τις εικόνες (σύνολο 8.000) τους κώνους μέσα σε αυτές, σχεδιάζοντας ένα bounding box γύρω τους. Οι κώνοι που εξάγαμε ήταν σε διαφορετικό περιβάλλον, γωνίες φωτισμού, θέσεις και μέγεθος.

Γ. Scaling:

Ο στόχος μας ήταν να επιτύχουμε εντοπισμό σε μεγάλο εύρος, πάντα σύμφωνα με την υπολογιστική ισχύ του slave υπολογιστή. Επιτύχαμε ελάχιστο μέγεθος αντικειμένου 5x5 px και μέγιστο εύρος 25x25 px (20 m απόσταση αντικειμένου από το φακό). Ύστερα από διάφορες καταγραφές των υπολογιστικών απαιτήσεων του αλγορίθμου, η βέλτιστη απόσταση που εντοπίσαμε αντικείμενα ήταν τα 10m στα 20x20 px, για real time data detection.

Δ. Training:

Τα τελικά training του αλγορίθμου έγιναν στον Euler (Erweiterbarer, Umweltfreundlicher, Leistungsfhiger ETH-Rechner) [32]. Χρησιμοποιήσαμε high speed GAB – Gentle AdaBoost και κάναμε training 45 σταδίων μέχρι να ολοκληρωθεί το βέλτιστο.

Κεφάλαιο 4

Συμπεράσματα

4.1 Αποτελέσματα

Προκειμένου να αξιολογήσουμε τα αποτελέσματα του trained cascade, του αισθητήρα και των αλγορίθμων, αξιολογούμε την έξοδο του vision pipeline, ως προς:

1. Loop Closure: Για το συγχρονισμό και την ακρίβεια των οδομετρικών μετρήσεων του VI συστήματος
2. Trajectory Precision: Από τον εξαγώμενο χάρτη και την οδομετρία, συγκρίνουμε τον σημασιολογικό χάρτη που εξήγαγε η οπτική με αυτόν του FAST – Slam.
3. Features discrimination: Η αξιολόγηση του εντοπισμού βασίζεται στη σωστή διάκριση των visual SLAM features, σε «αριστερός κώνος», «δεξής κώνος» και «άλλο».
4. Features detection: Ακόμα μία αξιολόγηση εντοπισμού με βάση τη μέθοδο Intersection over Union (IoU) [24] μεταξύ δύο bounding boxes A και B. Συγκρίνουμε αυτό που εντόπισε ο αλγόριθμος με το “ground truth” που θέσαμε εμείς.

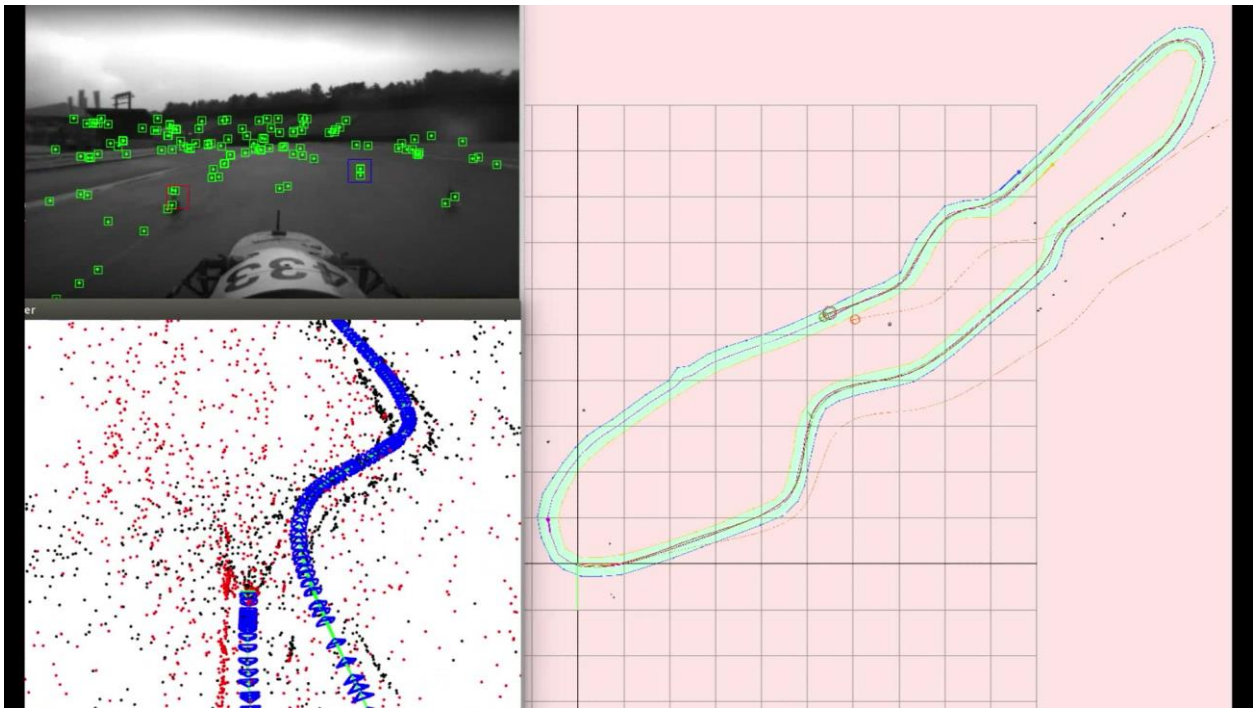


Figure 4.1: Πάνω αριστερά. Τα πράσινα τετράγωνα είναι τα landmarks που εντόπισε ο ORBSLAM, τα κόκκινα και μπλε είναι οι κώνοι (αριστερά και δεξιά) από το Cone Detection. Κάτω αριστερά, ο σημασιολογικός χάρτης που εξάγει το Vision Pipeline. Δεξια ο σημασιολογικός χάρτης που εξάγει ο FAST – SLAM

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BIWI Computer Vision

An Efficient Track Detection and Mapping System for Autonomous Driving Race Car

Formula Student Driverless

Master's Thesis

Efimia Panagiotaki
Department of Information Technology and Electrical Engineering

Advisor: Dr. Dengxin Dai
Supervisor: Prof. Luc van Gool

October 2, 2017

Abstract

In August 2017, Akademischer Motorsportverein Zürich (AMZ) Driverless, participated in the Formula Student Driverless Competition in Hockenheimring, Germany, winning the 1st place overall, among 15 other driverless teams. AMZ Driverless is the only team that managed to successfully complete all the disciplines in the competition.

The team developed a fully autonomous driving vehicle, using as a platform the prototype that was built by AMZ Racing in 2015. Flüela is a lightweight, 4 wheel drive electric race car. AMZ Driverless consists of master students from ETH Zürich, former team members of the ETH Formula Student Project (AMZ Racing) and the author, an Invited Visiting student at ETH Zürich, from the National Technical University of Athens, Greece. The design of the autonomous system is adapted to the rules and regulations of the Formula Student Driverless Competition.

The autonomous concept is focused on redundancy of the sensors and robustness of the autonomous system, in order to reassure success and reliability. The goal of the perception pipeline is to generate an accurate semantic map of the track, to provide accurate velocity and pose estimates and to assure redundancy in case of single mode failures. The perception sensors include a LiDAR and a self-developed Visual Inertial Sensor for boundaries detection and path planning. This Master Thesis, is based on real-time data of the Visual pipeline, from the actual vehicle during testing conducted throughout the season before and during the competition.

The visual pipeline provides a redundant way to detect cones, reassuring robustness in case of LiDAR failures. For the sensor, we synchronize two monocular cameras and an Inertial Navigation System (INS) in order to build an accurate and secluded visual SLAM system completely from scratch. The system is optimized for large scale applications and outdoors conditions, with respect to the vehicle's autonomous system.

To detect cones' positions, we implemented a state-of-the-art machine learning cone detection algorithm which classifies SLAM tracked landmarks to "cones" and "other". The combination of cone detection, stereo SLAM and VI-odometry has proven to be robust to fast angular rates and successful tracking was achieved for more than 180°/s. Our work is evaluated based on the precision and accuracy of: the cone detection, the loop closing, the generated map and the trajectory accuracy.

Acknowledgements

This thesis has been conducted at the Formula Student Driverless project of ETH Zürich, during the spring semester of 2017. Before I elaborate any further on this master thesis, I would like to express my gratitude to the people who have helped in realizing it.

First of all, I would like to sincerely thank my supervisors, prof. Luc Van Gool and dr. Dengxin Dai from the Computer Vision Lab, for their great support in the driverless project and this master thesis, during and before the thesis even started. I would also like thank my supervisor from my home University, the National Technical University of Athens (Greece), prof. Evangelos Christoforou, for his support at the beginning of the project.

A special thanks goes to my team, Akademischer Motorsportverein Zürich (AMZ) Driverless, Zürich, the Formula Student team of ETH Zürich. Each and every member of the 2017 project played a crucial role in the conduction of the thesis and the success of the season. AMZ Racing enabled this work, providing all the tools, materials and the vehicle required.

Finally, I am grateful for the support I had from Alex Athanasiadis and my family who made it possible to relocate to Switzerland, be part of AMZ Driverless and conduct my master thesis at ETH Zürich as an Invited Visiting Student.

Acronyms and Abbreviations

AMZ	Akademischer Motorsportverein Zürich, the Formula Student Team of ETH Zürich
ASMS	Autonomous System Master Switch
BPP	Business Plan Presentation
DV	Driverless Vehicles
DRS	Drag Reduction System
EV	Electric Vehicles
EBS	Electronic Brake System
flüela	Formula Student Electric Vehicle built in 2015 by AMZ Racing
FS	Formula Student
FSD	Formula Student Driverless
FSG	Formula Student Germany
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System (IMU and GPS)
IR	Infra-red
LBP	Local Binary Patterns
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
SM	Semantic Map
TOF	Time of Flight
VI	Visual Inertial
Detection	Localization and extraction of potential image features
Features	Specific structural part of interest in an image (2D)
Landmarks	Unique 3D points representing a distinct secular structure
Master	Main computer unit of the autonomous system
OpenCV	Open Source Computer Vision Library
Slave	Second computer unit of the autonomous system, used by the computer vision algorithms.
Smearing	Bright Vertical line from top to bottom in an image emanating from an extremely bright light source
Tracking	Spatial and temporal re-identification of a landmark

Contents

1 Introduction **1**
- 1.1 Formula Student . . . 1
  - 1.1.1 Events and Judging . . . 2
  - 1.1.2 Track Marking . . . 3
  - 1.1.3 Rules Relevant to Vision . . . 5
- 1.2 Autonomous System Overview . . . 6
  - 1.2.1 Basic Concept . . . 6
  - 1.2.2 Implementation Requirements . . . 7
- 1.3 Thesis Organization . . . 7

2 Related Work **9**
- 2.1 Skybotix VI Sensor . . . 9
- 2.2 TriggerSync: A Time Synchronization Tool . . . 9
- 2.3 Viola Jones Face Detection Algorithm . . . 9
- 2.4 Local Binary Pattern (LBP) . . . 10
- 2.5 You Only Look Once (YOLO) . . . 10
- 2.6 Faster R-CNN . . . 10
- 2.7 Rovio . . . 10
- 2.8 ORB-SLAM . . . 11

3 Materials and Methods **13**
- 3.1 The vehicle: flüela . . . 13
- 3.2 Software Tools . . . 14
  - 3.2.1 Robot Operating System (ROS) . . . 14
  - 3.2.2 Git . . . 14
  - 3.2.3 OpenCV . . . 14
- 3.3 Methodology . . . 14

4 Visual Inertial Sensor System **17**
- 4.1 Depth Perception Techniques . . . 17
  - 4.1.1 Stereo Triangulation . . . 17
  - 4.1.2 Structured Light . . . 18
  - 4.1.3 Time of Flight (TOF) . . . 18
  - 4.1.4 Comparison . . . 19
- 4.2 Sensor Design . . . 19

4.2.1 Hardware selection . . . 19
4.2.2 Experiments and Results . . . 21
4.2.3 Conclusions . . . 21
4.2.4 Wiring . . . 23
4.2.5 Design . . . 24
4.2.6 Position . . . 24
4.3 Calibration . . . 25
4.4 Synchronization . . . 26
4.4.1 Triggering . . . 26
4.4.2 Experiments and Results . . . 27
4.4.3 Solution . . . 28
4.4.4 Timestamping . . . 30
4.5 Outdoors behavior . . . 30
4.5.1 Solution . . . 31

5 Visual Cone Detection **33**
5.1 Perception Concept . . . 33
5.1.1 Perception Pipeline . . . 33
5.1.2 Software - Hardware architecture of Vision pipeline . . . 34
5.1.3 Detection Concept . . . 35
5.2 Local Binary Patterns . . . 35
5.3 Methodology and Structure . . . 36
5.3.1 Training Data generation . . . 36
5.3.2 Annotation . . . 37
5.3.3 Scaling . . . 38
5.3.4 Training . . . 38
5.4 Results . . . 38

6 Conclusion **39**
6.1 Evaluation . . . 39
6.2 Discussion . . . 39
6.3 Conclusions . . . 40

A Calibration parameters **41**

B Stereo camera synchronization **43**

C Datasheets **45**

List of Figures

1.1 FSD Cones . . . 3
1.2 FSD Acceleration . . . 3
1.3 FSD Trackdrive . . . 4
1.4 FSD SkidPad . . . 4
1.5 Minimum helmet clearance . . . 5
1.6 Envelope to mount sensors . . . 5
1.7 Autonomous System overview . . . 6

3.1 flüela . . . 13
3.2 First indoors ORBSLAM Test . . . 15

4.1 Stereo Triangulation geometric model . . . 17
4.2 Structured Light . . . 18
4.3 Time of Flight . . . 18
4.4 DUO MLX . . . 21
4.5 ZED Stereo Camera . . . 21
4.6 Vectornav VN200 . . . 22
4.7 Blackfly PointGrey . . . 22
4.8 Sensor's wiring diagram . . . 23
4.9 Sensors mounting . . . 24
4.10 Sensors position . . . 24
4.11 Triggering wiring diagram . . . 26
4.12 Triggering Delays (Source:[3]) . . . 26
4.13 Multi-shot triggering (Source:[3]) . . . 27
4.14 Clearing delays . . . 27
4.15 84fps in dim lighting . . . 28
4.16 80 fps in bright lighting . . . 28
4.17 Stereo Synchronization . . . 29
4.18 Synchronization Result: A picture of a rosbag from the slave computer during testing. The vertical lines are the frames grabbed from each camera, spaced equally on the time line. . . . 30
4.19 Driving opposite and towards the sun (1:00 pm) . . . 31
4.20 Driving opposite and towards the sun (5:00 pm) . . . 31
4.21 Image captured without filtering (Facing the sun) . . . 32
4.22 Image captured with filtering (Facing the sun) . . . 32
4.23 Image captured without filtering (Opposite direction) . . . 32

4.24 Image captured with filtering (Opposite direction) 32

5.1 Fast-SLAM: boundaries discrimination 34
5.2 Visual Sensor flow 34
5.3 ORBSLAM and Rovio right before (left) and right after (right) performing loop-closure . . 35
5.4 Annotated images 37
5.5 Intersection over Union. 38

6.1 On top left, the green rectangles are the landmarks detected by ORBSLAM and the red and blue rectangles are the cone detection (left and right). Bottom left, is the SM while being generated from the vision pipeline. On the right, is the SM generated by FAST-Slam 40

B.1 Bright lighting: 78 fps 43
B.2 Bright lighting: 80 fps 43
B.3 Indoors lighting: 83 fps. Result: trigger missing 43
B.4 Indoors lighting 83 fps. Result: trigger missing 43

List of Tables

1.1 Maximum points awarded in DV 2

3.1 flüela driverless: Technical specifications 14

4.1 3D measuring Techniques 19
4.2 Cameras Selection 20
4.3 Blackfly 0.3 MP Mono USB3 22
4.4 Stereo Configuration 29

A.1 left frame: camera matrix 41
A.2 left frame: distortion 41
A.3 left frame: rectification 41
A.4 left frame: projection 42
A.5 right frame: camera matrix 42
A.6 right frame: distortion 42
A.7 right frame: rectification 42
A.8 right frame: projection 42

Chapter 1

Introduction

Path planning, localization and mapping in autonomous driving demand robust perception of the environment. Visual sensors have been the leading perception sensor for depth calculation and object detection. In comparison to active ranging and proximity systems, visual sensors provide features recognition, long range depth accuracy and localization in the environment.

The goal of this project was the development and implementation of a synchronized, compact, accurate and reliable system which provides real-time Simultaneous Localization and Mapping (SLAM) capabilities. The Visual SLAM system is integrated in an autonomous driving race car, and the system is developed with respect to the vehicle's autonomous system and the rules of the Formula Student Driverless Competition. Our system provides accurate real-time pose and velocity estimation and mapping. To fill the requirements of the environment, the visual inertial system had to be optimized for different illuminations and weather conditions. The system had to be able to work at high speeds, requiring limited computational power.

At the beginning of the project, we successfully tested the performance of the Visual SLAM algorithms [14], [12] using the visual-inertial Skybotix VI Sensor[8] developed by the Autonomous Systems Lab (ASL) of ETH Zürich. We achieved precise track mapping and accurate loop closure, while driving the race car on track. The purpose of the test was to reassure that the visual pipeline was capable of providing accurate position of the vehicle and the map of the track. Due to the fact that the sensor was no longer available, we decided to build our own Visual Inertial sensor to run the visual SLAM algorithms and to better fit the requirements of our project. To detect cone's positions and to track and distinguish the left and right cones for accurate trajectory, we implemented a machine learning cone detection algorithm which classifies SLAM tracked landmarks.

1.1 Formula Student

The competition challenges teams of university students to conceive, design, fabricate, develop and compete with small formula style race cars.[25] For the purpose of the competition, the students assume that a manufacturing firm has engaged them to produce a prototype open-wheel race car, for the non-professional weekend race car driver. The vehicles are considered as production items and need to fit the demands of the potential market. Each team is compared and judged with the other competing teams, to determine the best overall winner.

The competition is split into three classes: Internal Combustion Engine Vehicles (CV), Electric Vehicles (EV) and Driverless Vehicles (DV). All vehicles must meet the technical requirements defined by the rules of each class, in order to be eligible to compete. The competition starts with a series of technical inspections,

to check the vehicle for safety and compliance with the rules.

Formula Student Driverless was introduced in Formula Student at the 2016-2017 season, with the first ever competition to be held in FS Germany 2017. The teams participate with second or third year, EV or CV vehicles. The vehicles need to be compliant with the technical rules of the DV Class and the EV or CV Class, depending on the vehicle. [4]

1.1.1 Events and Judging

The competition is divided into a series of static and dynamic events, which form the final score of the teams. Maximum points for the DV class are awarded as described on Table 1.1 [4]. The DV static events include Business Plan, Cost and Manufacturing, Engineering Design and Autonomous Design. Based on the hypothetical scenario described, the teams need to hold a series of presentations for the fictional manufacturing firm.

Table 1.1: Maximum points awarded in DV

	Driverless
Static Events	
Business Plan Presentation	75 points
Cost and Manufacturing	100 points
Engineering Design	150 points
Autonomous Design	175 points
Dynamic Events	
Skid Pad	75 points
Acceleration	75 points
Efficiency	100 points
Trackdrive	250 points

The objective of the BPP is to evaluate the teams ability to develop and deliver a comprehensive business model and explain how their product, a prototype race car, could become a rewarding business opportunity. The objective of the cost and manufacturing event is to evaluate the teams understanding of the manufacturing processes and costs associated with the build of a prototype race car. This includes trade off decisions between performance and cost, make or buy decisions and understanding the difference between prototype and mass production.
The concept of the design event is to evaluate the students engineering process and effort that went into the design of a vehicle, meeting the intent of the competition.[4]
The purpose of this event is to evaluate the capability of the vehicle to drive autonomously. Therefore, all systems that are required to drive autonomously will be investigated. This also includes a discussion about the hardware and the software used in the autonomous system. Since the autonomous design event takes place after the dynamic disciplines, the vehicles performance on the trackdrive is also taken into account. The autonomous design consists of three parts: the Autonomous Design Report (ADR), the autonomous design presentation and discussion and the race review.

The dynamics events include: Skid Pad, Acceleration, Efficiency and Trackdrive. The skidpad course consists of two pairs of concentric circles in a figure eight pattern. The acceleration course is a straight course with a length of 75m from the starting line to the finish line. Energy efficiency is measured during the trackdrive event. The trackdrive layout is a closed loop circuit following the below mentioned guidelines:

- Straights: No longer than 80 m
- Constant Turns: up to 50 m diameter
- Hairpin Turns: Minimum of 9 m outside diameter (of the turn)
- Miscellaneous: Chicanes, multiple turns, decreasing radius turns, etc.
- The minimum track width is 3.5 m

The length of one lap is approximately 200 m to 500 m [4]

1.1.2 Track Marking

As described at the "FSG DV Technical Specification 2017"
On the dynamic events the track is marked with cones, as shown in Figure 1.1, 1.2, 1.3, 1.4. The left borders are marked with small blue cones and the right borders are marked with small yellow cones. Exit and entry lanes are marked with small orange cones. Big orange cones are placed before and after start, finish and timekeeping lines (Figure 1.4). The maximum distance between two cones in driving direction is 5 m. In corners, the distance between the cones is smaller for a better indication.

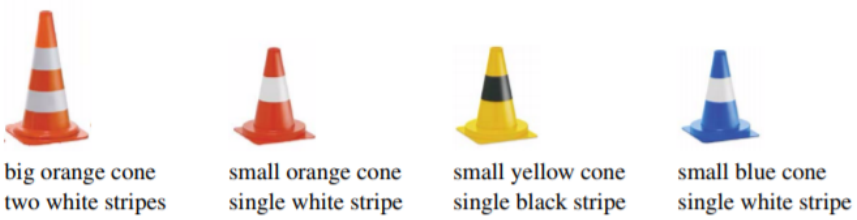


Figure 1.1: FSD Cones

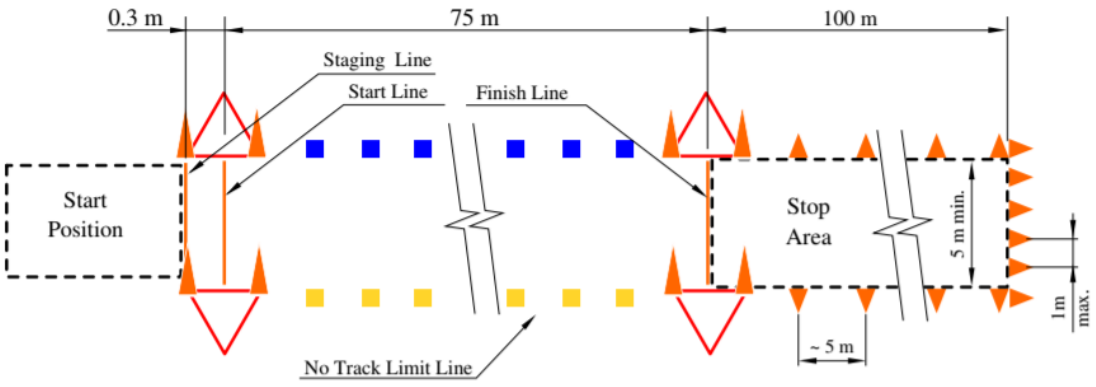


Figure 1.2: FSD Acceleration

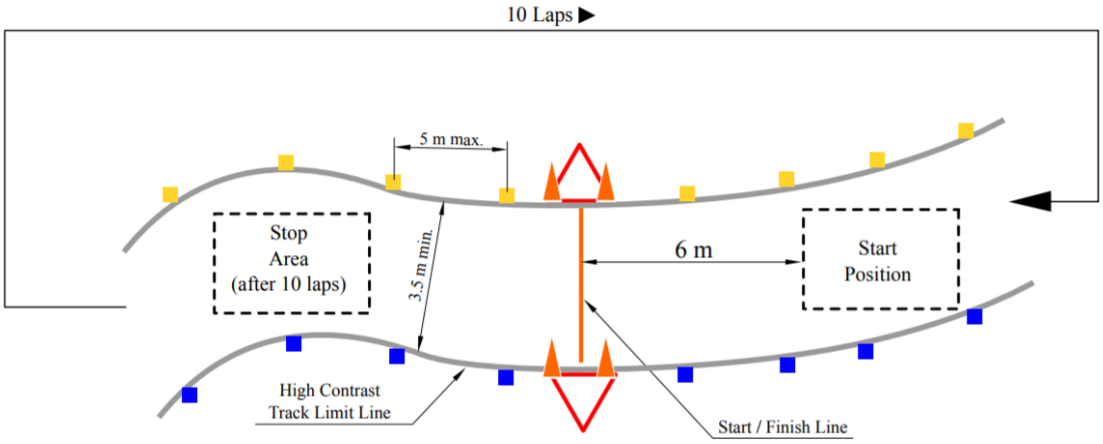


Figure 1.3: FSD Trackdrive

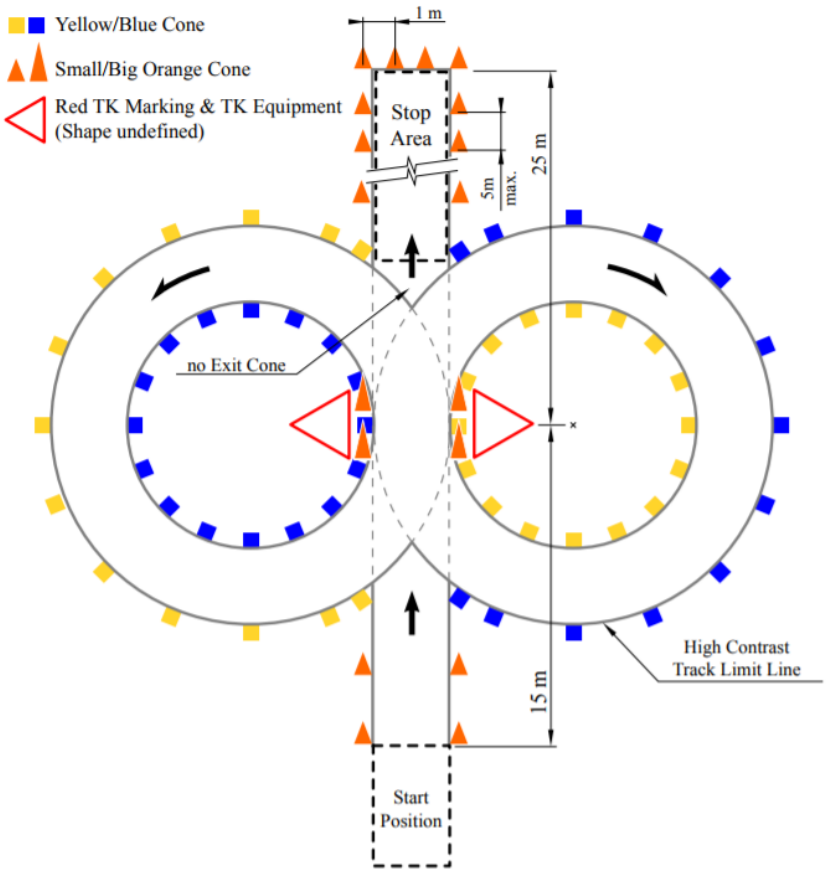


Figure 1.4: FSD SkidPad

1.1.3 Rules Relevant to Vision

DV 2.2.5 When the ASMS is in Off position (i.e. manual mode), It must be possible to operate the vehicle manually as a normal CV or EV.
DV 4.1.2 Sensors may not come into contact with the drivers helmet in any circumstances.
T 3.3.1 When seated normally and restrained by the drivers restraint system, the helmet of a 95th percentile male and all of the teams drivers must: (a) Be a minimum of 50 mm away from the straight line drawn from the top of the main hoop to the top of the front hoop. (b) Be a minimum of 50 mm away from the straight line drawn from the top of the main hoop to the lower end of the main hoop bracing if the bracing extends rearwards. (c) Be no further rearwards than the rear surface of the main hoop if the main hoop bracing extends forwards.
T 3.10.1 The driver must have adequate visibility to the front and sides of the vehicle. Seated in a normal

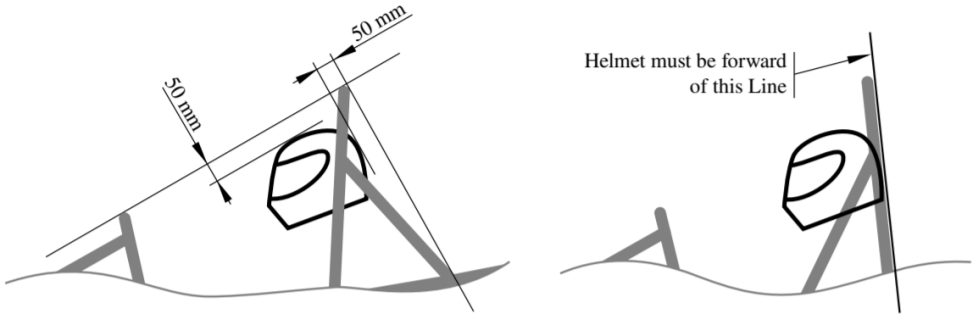


Figure 1.5: Minimum helmet clearance

driving position, the driver must have a minimum field of vision of 200 (a minimum 100 to either side). The required visibility may be obtained by the driver turning their head and/or the use of mirrors.
DV 4.1.3 All sensors must be positioned within the surface defined by the top of the roll bar and the outside edge of the four tires (Figure 1.6).
DV 4.1.4 Additionally, sensors may be mounted with a maximum distance of 500 mm above the ground and not further forward than 700 mm forward of the front of the front tires. They must not exceed the width of the front axle (measured at the height of the hubs)

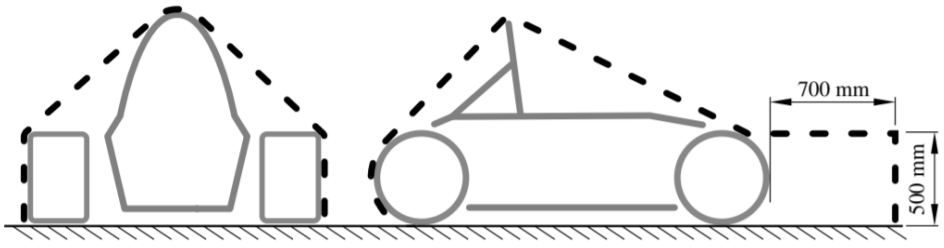


Figure 1.6: Envelope to mount sensors

1.2 Autonomous System Overview

The goal of AMZ Driverless' autonomous system is to build a reliable and competitive system in order to maximize the overall points at FSD. We developed the autonomous race car focusing on the algorithms, the sensors, the computation power and the actuators needed. To reassure reliability of the vehicle and the autonomous system, we started testing online our systems on the vehicle four months before the competition. The vehicle is using a LiDAR, a self - developed stereo visual-inertial system, an Inertial Navigation System (INS) and an absolute velocity sensor[10]. The system runs on two computing units in Master - Slave setup. The development of the project is done in ROS(Robot Operating System) Indigo. The Slave Computer is needed and used only for the Vision algorithms. The rest of the autonomous system including the LiDAR pipeline, run on the Master Computer. The vehicle has been modified to include the controlled actuators: an electric power steering system, four self-developed wheel hub motors and the Emergency Brake System (EBS).

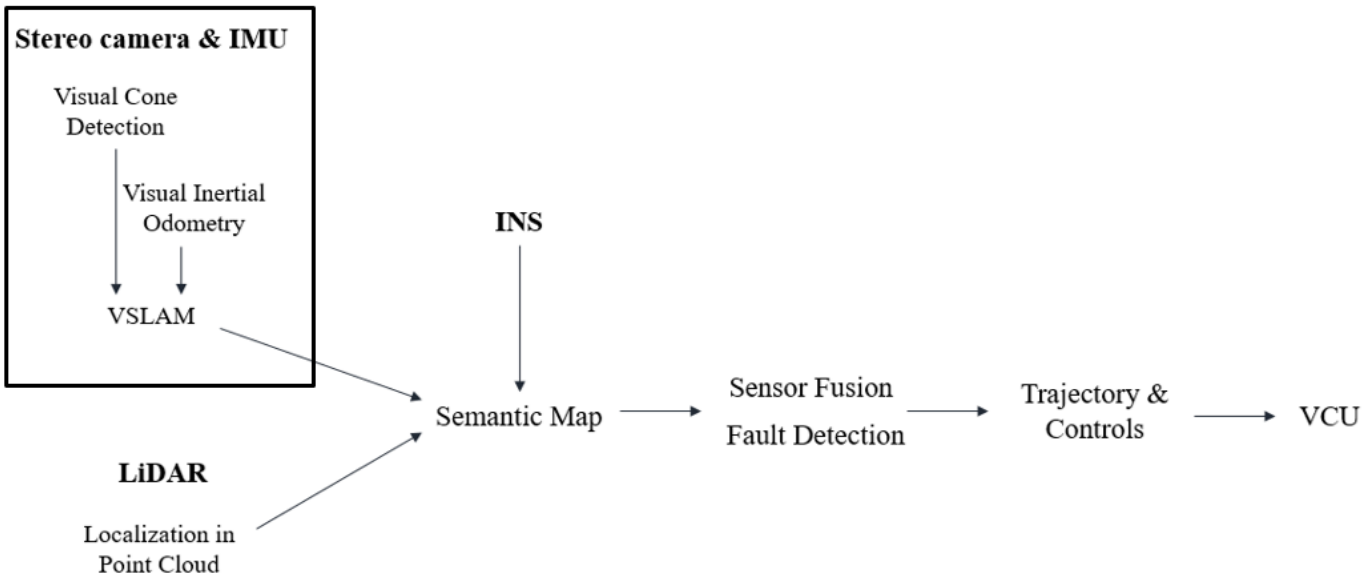


Figure 1.7: Autonomous System overview

To achieve high speeds when racing autonomously, the track must be known at least 2s ahead and the perception horizon needs to exceed the sensors' range. The car must drive carefully on the first laps to discover and build the Semantic Map (SM) of the track ("Discovery Mode") in which it will race on the next laps ("Race Mode").
The perception design needs to be with respect to the above mentioned requirements of real-time racing, while the visual design is also limited by the Computational Power of the system.
In this thesis we are focusing on the visual pipeline of the autonomous system.

1.2.1 Basic Concept

In more detail, the goal is to build a robust visual system, which uses a synchronized stereo camera, and IMU measurements as input, and outputs left and right cones distinctive detection, velocity estimates and

a 3D map of cone landmarks. The output is then being used as LiDAR output validation reference or even directly as SM in case of LiDAR failure.
ORBSLAM[14] is a state-of-the-art stereo SLAM algorithm, which is being used in combination with a monocular visual-inertial (VI) system, ROVIO[12], with access to dedicated accelerometers and gyroscopes. As a prior to ORBSLAM, we implemented a machine learning cone detection algorithm, in order to detect cone's positions. The algorithm is based on a boosted cascade of weak classifiers [19] and provides excellent trade-off between accuracy and run time.
Building the sensor from the ground-up allowed us to customize the cameras hardware selection, the stereo baseline, the sensors' synchronization and timestamping. The Visual Inertial Sensor consists of two monochrome monocular cameras and an Inertial Navigation System (IMU and GPS). The GPS is being used for synchronizing and time stamping of the measurements.
Left and right cones distinction is achieved due to the characteristic pattern of the cones in gray scale and the prior knowledge we have of the cones characteristics as described in Section 1.1.2.

1.2.2 Implementation Requirements

- The Slave computer runs the computer vision algorithms and is a powerful Quad-Core i7, 4.2 GHz. The algorithms and hardware selection had to be adjusted and optimized in terms of computational and time complexity, in order to run in real time.
- The Visual SLAM algorithms we implemented, are optimized for gray scale images and in addition to our computing power limit, we had to use monochrome cameras for real time image processing, excluding color detection techniques.
- The detection algorithm has to be robust against different illuminations and weather conditions. (Section 5.3)
- Visual SLAM requires reliable synchronization of the sensors and prior knowledge of delays for timestamping the measurements. The stereo frames need to be timestamped accordingly for ORB-SLAM[20] to be able to use the stereo setup. (Section 4.4)
- Measurements' timestamps needed to be with respect to the global reference clock used by the vehicle: GPS Clock (Section 4.4.4)
- Synchronization needs to be invariant of the environment and the different lighting conditions. The behavior of the cameras varies depending on the shutter speed required, which can cause inconsistent frame grabbing and not synchronized sensors.
- The system needs to be compatible with the vehicle. Any external "noise" from flüela had to be handled accordingly. We mainly faced Electromagnetic interference and extreme vibrations from the water pumps.
- The cameras position needs to be compliant with the FSD rules, as described in Section 1.1.3

1.3 Thesis Organization

This thesis consists of six chapters and can easily be divided in two different parts: the development and implementation of a visual inertial sensor and the cone detection algorithm.

In the first Chapter, we provide an overview of the project, the competition, our goals and motivations. In the following Chapter, we describe related work in the fields of visual inertial sensors, visual SLAM, synchronization tools and object detection. In Chapter 3, we present the platform in which the system is implemented and we describe the software tools used from the driverless project. We also provide details of the development strategy and methodology we followed. In Chapter 4, we describe the visual inertial system's development process, the synchronization and integration with the autonomous system. We also describe the experiments conducted and our observations. In the following Chapter 5, we describe the visual cone detection and the perception pipeline focusing on vision. In the final chapter, we are giving a resume of the systems' performance during the competition, which can initiate a series of discussions.

Chapter 2

Related Work

2.1 Skybotix VI Sensor

The Autonomous Systems Lab (ASL) from ETH Zrich, built a time-synchronized, calibrated sensor which is targeted at mobile robotic applications[8]. It provides accurate, robust and real-time pose estimations and mapping in controlled environments. The sensor integrates a stereo camera set up along with an Inertial Measurement Unit (IMU), interfaced through an ARM-FPGA micro-controller. It provides high-quality rate gyro and accelerometer measurements, calibrated and hardware-synchronized with the camera frames. Tight fusion of virtual and inertial measurements is achieved. In the programmable hardware, image processing tasks are implemented, aiming to speed up the processing and CPU demand. The measurements are the inputs of a tightly coupled real-time visual inertial SLAM framework, whose output demonstrates the sensor's capabilities.

2.2 TriggerSync: A Time Synchronization Tool

Andrew English et al.[6] presented a framework for synchronizing multiple triggered sensors with respect to a local clock. They demonstrated a method of synchronizing measurements timestamps from different sensors, to each other. The sensors are synchronized with hardware lines to local computer time using standard computing hardware. It uses one way synchronization algorithms combined with sharing of triggered timestamps between clock estimators to gain many of the benefits of two way synchronization algorithms, without requiring support from the sensors. Results demonstrate the effectiveness of this framework by synchronizing offsets between a camera and an IMU to within approximately 100μs.

2.3 Viola Jones Face Detection Algorithm

Paul Viola and Michael Jones [19] proposed a machine learning approach for real-time visual object detection, in 2001. Their algorithm was capable of processing images extremely rapidly and achieving high detection rates. They introduced the "Integral Image" for the first time. The features used in detection were calculated very fast. From the Adaboost learning algorithm[26], they developed a learning algorithm, which was able to collect a small number of visual features from a large set of data. Finally, using cascade features and combining more complex classifier, they managed to ignore the background of an image and to extract the important image features.

2.4 Local Binary Pattern (LBP)

T. Ojala et al.[15], first introduced a LBP operator for texture description and face recognition. They focused on gray scale and rotation invariant texture classification, based on local binary patterns. For face detection, the face is segmented into small regions to extract the LBP histogram. The LBP descriptor captured very fine grained features in an image and is based on ”uniform” patterns.

2.5 You Only Look Once (YOLO)

J.Redmon et al.[9], suggested a new and fast real-time approach for object detection. They re-framed object detection as a single regression problem to spatially separate bounding boxes and associated class probabilities. With their method, object presence prediction only requires to look in the image once. They relied on the fact that single neural networks predict the bounding boxes in one evaluation. The detection pipeline is a single network and can be optimized end-to-end directly on detection performance.
YOLO is trained on a loss function, which directly corresponds to detection performance, unlike classifier based object detection. The entire model is trained jointly and runs on a GPU. Compared to state-of-the-art detection systems, YOLO detects less false positives but makes more localization errors. It is considered to outperform other detection methods including DPM and R-CNN.

2.6 Faster R-CNN

Shaoqing Ren et al.[27] introduced a Region Proposal Network (RPN), allowing cost-free region proposals. Faster R-CNN can be divided in two main modules, a deep fully convolutional network that proposes regions, and a Fast R-CNN[7] detector. RPN shares full-image convolutional features with the detection network and makes accurate predictions of object bounds. It is trained end-to-end to generate high-quality region proposals. They afterwards merged RPN and Fast R-CNN into a single network by sharing their convolutional features. The system requires a GPU for the processing and the training. The detection in real time appears robust and accurate.

2.7 Rovio

Bloesch et al.[12] presented a monocular visual inertial algorithm, achieving accurate tracking performance and robust monocular visual inertial odometry. They directly used pixel intensity errors of image patches, for tracking of the multilevel patch features. After detection the tracking is coupled to the underlying Extended Kalman filter (EKF). The location of 3D landmarks is always estimated with respect to the current camera pose. Landmark representations are decomposed into bearing vectors, to improve the computational performance of the algorithm. Rovio is tested and evaluated in a great number of small and large scale experiments.

2.8 ORB-SLAM

Mur-Artal et al.[20] presented a real - time feature-based monocular simultaneous localization and mapping (SLAM) system. It appears robust to severe motion clutter, allows loop closing and relocalization for wide baseline and includes full automatic initialization. The system operates in small and large indoor and outdoor environments. It achieves excellent robustness and generates a compact and trackable map. The map grows if the scene content change, allowing lifelong operations. ORB-SLAM achieves unprecedented performance with respect to other state-of-the-art monocular SLAM approaches.

Chapter 3

Materials and Methods

3.1 The vehicle: flüela

Flüela was built in 2015 by AMZ Racing. It is a 4 wheel drive electric vehicle. It integrates a full aerodynamic package with a Drag Reduction System (DRS), high wheel torque and a lightweight design (188.2kg). The vehicle runs in closed tracks at different velocities.

Figure 3.1: flüela

Table 3.1: flüela driverless: Technical specifications

	flüela driverless
Kerb weight	188,2 kg
Length	2870 mm
Width	1438 mm
Height	1139 mm
Maximal speed	119 kph
Max wheel torque	374 Nm
Peak power	148 kW

3.2 Software Tools

3.2.1 Robot Operating System (ROS)

The vehicle's computing units run on ROS and our autonomous system is implemented in ROS Indigo. ROS is an open source robotics framework, which provides libraries and tools for robotics applications[23]. Features used widely for our system are device drivers, libraries, visualizers, message-passing and package management. For our system it enabled visualization and low level hardware interference using the device's drivers, which facilitated synchronization and evaluation of the measurements, as we will describe in Chapter 4.

3.2.2 Git

Git is a version control system which allows us to keep a record of the project history[1]. We primarily use it for source code management and distributed revision control, to reassure quality control. Using the branching features of Git, we were able to manage the project's code from development stage to testing on the vehicle to the final version implemented at Formula Student Germany.

3.2.3 OpenCV

We use Open Source Computer Vision library for real-time computer vision[18]. It is a cross-platform library with integration to ROS. Visual Cone Detection uses OpenCV's applications including the facial recognition system (annotation and training), implementing statical machine learning libraries (eg. boosting).

3.3 Methodology

Our system was built incrementally. The development process was divided in stages, adding the needed functionalities until we fulfilled the requirements of our application. In order to reassure robustness in different environments for both the visual inertial sensor and the visual cone detection algorithm, we considered different environments of increasing complexity.

1. Visual Inertial Sensor

   - *Hand held sensor and indoor application:* We primarily focused on evaluating the synchronization and timestamping matching of the two monocular cameras, via testing loop closure of the ORBSLAM[14] algorithm, on an indoors environment. This is an easy environment because: it is a static environment without dynamic components (eg. moving people) and it allows close range elements detection. As a first test, we controlled the movements of the sensor to be as smooth as possible. Indoors surroundings have a great number of features to detect. (Figure 3.2)

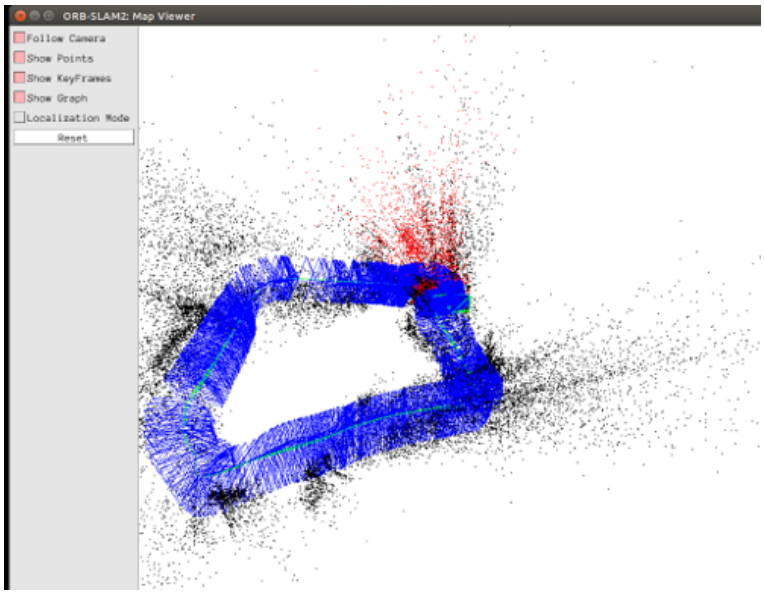


Figure 3.2: First indoors ORBSLAM Test

   - *Hand held sensor and outdoor application with medium lighting:* Testing the synchronization between the IMU and the Stereo camera. Outdoors environment offered distant features and more complex odometry computation.
   - *Sensor mounted on vehicle in outdoors applications:* Long distances at high speed, evaluating the performance of the system on large scale applications. More dynamic elements and sharp sensor movements, while turning the vehicle. We tested the behavior of the sensor mounted on the vehicle and its reaction with the entire system. (Section 4.2.6)
     The performance criteria of our system were sensors' synchronization accuracy, landmarks detection, loop closure and interference from the vehicle. After reassuring the autonomous system's accordance with the visual pipeline and the capability of our system to provide an accurate semantic map, we started testing on different lighting and weather conditions to estimate the sensor's behavior and the differences on the images received. (Section 4.5)

2. Visual Cone Detection

   - *Indoors testing - Data collection:* The environment was chosen because it encloses a great number of features, steady lighting and patterns that could affect the detection. Depending on the performance of the algorithm we increased the negative set of training data. Since the algorithm is based on left and right discrimination (Chapter 5), we adjusted the training set to satisfy the performance requirements of the detection (Section 5.3.1).
   - *Outdoors testing - Data collection:* Our goal was the collection of training data, both positive and negative sets of images. Having the camera mounted on the vehicle, we trained the algorithm on images from the actual perspective that the camera would have (Section 5.3.1) and on different hue and intensity of the images.
   - *Outdoors testing - System testing:* Testing the implementation of the algorithm, with the autonomous driving system. Our focus was to test the required computing and time complexity of the vision pipeline, while running the Slave computer. The vehicle was at that point driving fully autonomously with LiDAR and the output data from the cameras were being recorded for later comparison of the semantic maps and accuracy evaluation.
   - *Outdoors testing - Mapping testing:* The vehicle was running using only visual SLAM in different environments to reassure robustness and performance of the Visual Pipeline. We mainly focused on testing in different lighting conditions and on tracks with different coloring on the tarmac, that could affect the detection.
   - *Outdoors testing - Mapping testing:* The final test was a LiDAR - Camera switch, in which while the vehicle was driving, we switched from LiDAR to Cameras mapping, in order to evaluate the systems' capabilities in case of single mode failures, which was our initial goal.

Chapter 4

Visual Inertial Sensor System

4.1 Depth Perception Techniques

Examining the depth perception requirements of our vision system, we concluded in the following techniques: Stereo Triangulation, Structured Light and Time of Flight [5].

4.1.1 Stereo Triangulation

Using stereo triangulation, we can determine a point in 3D space given its projection in the images, using as a prior the disparity map, focal length and cameras calibration parameters. Assuming we have a stereo camera setup, the depth of an object is calculated from the relative distance between the two lenses. Stereo cameras, create the illusion of depth using two pictures taken at slightly different positions. The advantage is that the measurement is more or less passive[21]. It doesn't require special conditions in terms of scene illumination. We chose stereo triangulation, because it is the most reliable and robust method we could use for an outdoor environment and ORBSLAM[20] performs landmark tracking using stereo triangulation.

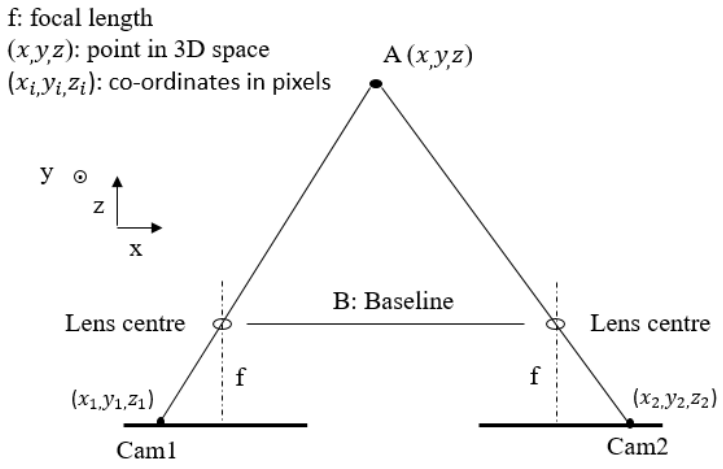


Figure 4.1: Stereo Triangulation geometric model

$z = f - \frac{fB}{x_2 - x_1}$

The focal distance is calculated after calibration of the cameras. Before calculation, x_2 and x_1 need to be converted to metric scale by multiplying them with a scaling factor, relative to the sensor. [21]

4.1.2 Structured Light

Cameras using structured light illuminate a scene with a light designed pattern (dots, lines etc). We focused on Infra Red cameras[31]. The IR projector, projects a pattern of IR light. It falls on objects, spreading its dots. The dots aren't visible to the human light, due to their color range. The images captured by the IR camera are in the Infra-Red color range. On nearer objects the dot pattern gets spread out and for the farther objects the dot pattern becomes denser. Using this displacement the depth is calculated. The depth camera's range usually is around 0.5m to 4m, beyond that there would be just holes and "noise" in the depth streams. The problems here was the fact that IR cameras are mainly for indoors use and have a very limited sensing range. They are quite inadequate for our project.

Figure 4.2: Structured Light

4.1.3 Time of Flight (TOF)

Time of Flight cameras are range imaging systems.
They calculate distance on an entire scene from the phase shift of the light pulse they send and receive[31]. They capture scenes in 3D with a dedicated image sensor. The distance is calculated from the phase difference between the emitted light and the reflective light.c is the speed of light and q_o, q_e are the accumulated charge in the pixel when light is emitted and when it is not [11]. Cameras that implement this technique are relatively new devices, with moderate outdoors performance, requiring light that has been reflected just once. Our system needs to be able to detect left and right cones and to provide redundancy. For this reason a sensor similar to LiDAR, like the TOF Cameras, doesn't fulfill our requirements.

$$d = \frac{ct}{2}\frac{q_o}{q_e+q_o}$$
$$d = \frac{c}{2}\frac{\Delta\phi}{2\pi f}$$

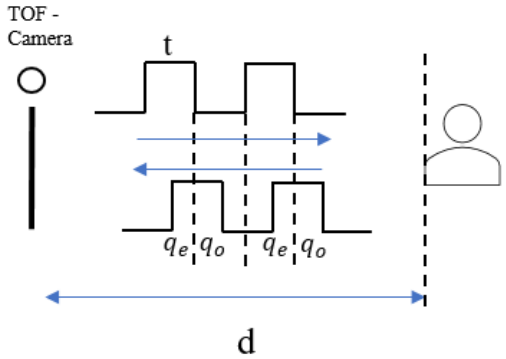


Figure 4.3: Time of Flight

4.1.4 Comparison

The results of our research are demonstrated on Table 4.1. Our initial conclusion after examining the performance trade - off between accuracy and outdoor performance, is to use the Stereo Triangulation Technique.

Table 4.1: 3D measuring Techniques

Technique	X-Y Resolution	Accuracy	Cost	Software Complexity	Outdoor performance
TOF	High	Medium	Medium	Low	Medium
Stereo	Scene dependent	Low	Low	High	Good
IR	Medium	High	High	Medium	Weak

4.2 Sensor Design

Our goal is to build a visual inertial system, able to provide accurate depth and odometry information.

4.2.1 Hardware selection

Examining the market's availability and adequacy with respect to our system's requirements and the available ROS drivers, we concluded on a wide list of cameras suitable for our project. (Table 4.2)

Table 4.2: Cameras Selection

	FPS	Res	Shutter	Mono/ RGB	Type	IMU	Depth Range
Intel RealSense R200	60	640x480	global	RGB	IR, Stereo	Yes	Depth range: 1.2m
ZED stereo camera	60	2560x720	rolling	RGB	Stereo	Yes	Depth range: 20m
DUO3D stereo camera	62	640x480	global	mono	Stereo	Yes	Depth range: 2.5m
SR4000 Swiss Ranger	54	176x144			TOF	Yes	Depth range: 10m
Kinetic v2	30	512x424		RGB	IR	No	Depth range: 4.2m
PMD Camcube 3.0	40	200x200			TOF	No	Depth range: 7m
Argos 3D P100 ToF Camera	160	160x120			TOF	No	Depth range: 3m
Sentis ToF M100 camera	40	161x120			TOF	No	Depth range: 2m
Nerian vision technologies	40	640x480	global	Mono	Stereo	No	Baseline: 10,15,25 cm
Orbec Astra	30	640x480, 1280x720	global	Mono	Stereo	No	Depth range: 8m
PointGrey Mono Cameras	80	640x480	global/ rolling	Mono/RGB	Monocular	No	Blackfly, Chameleon, Flea3 Grasshoper3
Videre Stereo Cam	30	640x480, 1280x720		RGB/mono	Stereo	No	Baseline: 8-24cm
Omni Stereo Cam	60	752x480		RGB/mono	Stereo	No	Baseline: 12cm
Multisense s7	30	808x608	semi-global	RGB/Mono	Stereo	No	
Tara USB 3.0 Stereo Vision	60	752x480	global	Mono	Stereo	Yes	Depth range: 3m
Bumblebee2 0.3 MP	48	648x488	global	RGB/Mono	Stereo	No	
Bumblebee XB3	16	1280x960	global	Mono	3 lenses, Stereo	No	
Ensenso N30	30	1280x1024	global	Mono	IR, Stereo	No	
Basler Ace acA1920-155um	164	1920x1200	global	Mono	Monocular	No	Depth range: 1.2m

4.2.2 Experiments and Results

1. Off-the-self Stereo Camera with integrated IMU:

   From the available Stereo Visual Inertial Camera systems, we tested the DUO MLX (Figure 4.4) which implemented stereo vision with odometry information. After testing visual SLAM and attempting loop closure with unsuccessful results, we concluded that the camera is not suitable for large scale applications. To be more precise:

   - Very small baseline (157mm)
   - Very narrow Field of View (FOV)
   - Sensors not properly synchronized and timestamped
   - IMU measurements' observational error prevented visual SLAM from achieving loop closure in indoors environment

Figure 4.4: DUO MLX

Figure 4.5: ZED Stereo Camera

2. Off-the-self Stereo Camera with 6-axis positional tracking: The second concept we examined was a wide base, Stereo Camera with integrated Real-time depth-based visual odometry and SLAM. We used the Zed Stereo Camera, with 20 meters depth range and wide baseline (12cm). Our conclusions were:

   - The camera is an RGB color camera and requires a GPU with very large compute capability for image processing, due to the amount of data that generates. Our system is based on the computational capability of a 2.4 GHz Quad Core CPU.
   - The built-in visual Odometry informations were neither sufficient, nor accurate for our application. The camera doesn't have an integrated IMU hardware. ROVIO requires as input the information from dedicated gyroscopes and accelerometers, in order to achieve loop closure.

4.2.3 Conclusions

We concluded that a stereo setup using two monocular cameras, is the best option for our application. The advantages of a self-developed visual inertial sensor is mainly the design freedom that allows us to have:

- Stereo Camera setup: Wide baseline in compliance with the rules of Formula Student
- Cameras' hardware and powering demands (Section 4.2.4)
- Cameras' hardware selection: Resolution, Read out Method, Frame Rate, Chrome
- Triggering options (Section 4.4.1)

- IMU Hardware selection based on accuracy, reliability and powering demands
- Frames capturing and exposure options
- Measurements time stamping and components Synchronization (Section 4.4)

The cameras chosen are two monochrome, monocular Blackfly PointGrey Cameras. In order to achieve triggering, synchronization and GPS time stamping, we decided on using the VN200 Vectornav INS, a 6-axis high precision Inertial Navigation System (IMU and GPS).

Table 4.3: Blackfly 0.3 MP Mono USB3

Resolution	648x488
Interface	USB3.1
Frame Rate	84 fps
Chroma	Mono
Sensor Type	CCD
Readout Method	Global Shutter
Triggering	Hardware and Software Triggering
Triggering Modes	Standard, bulb, low smear, overlapped, multi-shot

Figure 4.6: Vectornav VN200

Figure 4.7: Blackfly PointGrey

4.2.4 Wiring

Vectornav VN200:
Inertial Navigation System (IMU and GPS)
BFLY FLIR:
648x488, 84 fps, CCD, Monochrome, Global Shutter

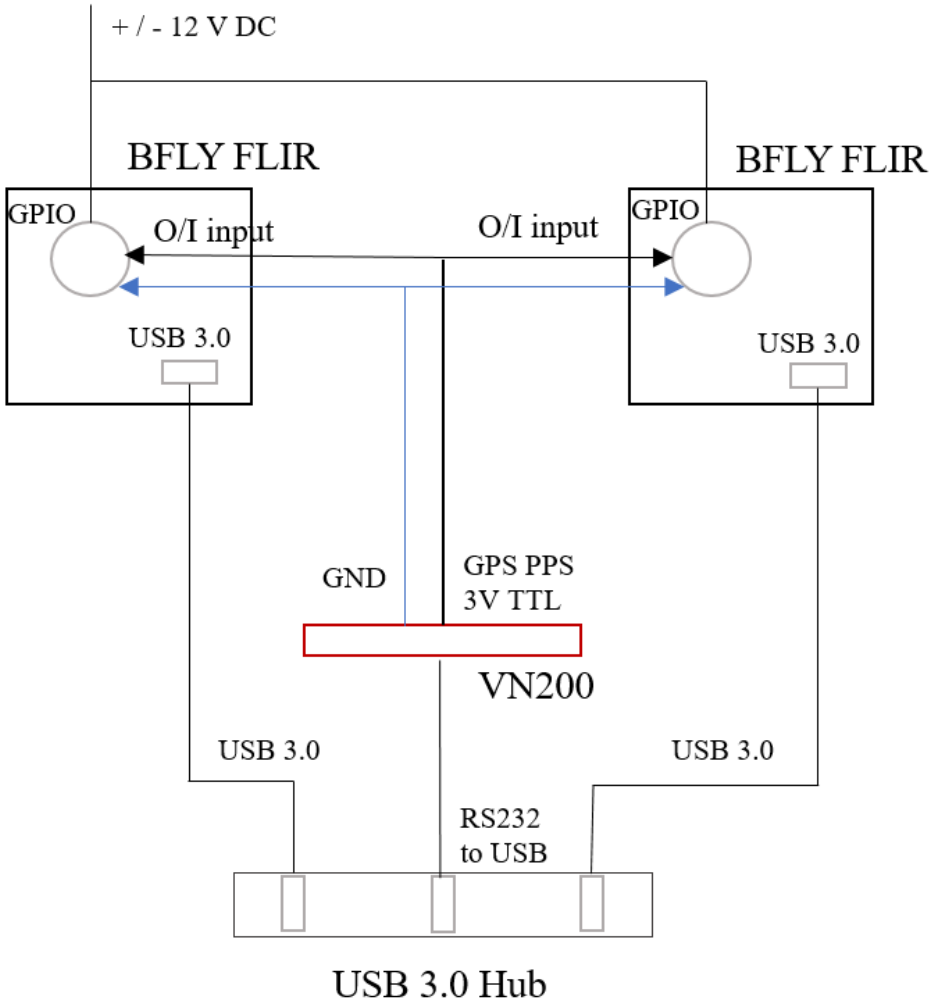


Figure 4.8: Sensor's wiring diagram

4.2.5 Design

The final mounting design of the sensor was optimized for minimum space occupancy, in order to be compliant to the rules of FSD. The case is 3D printed, plastic and tightly shielded for water resistance. It is designed by AMZ Racing for the shake of this project.
Baseline: 10.2 cm

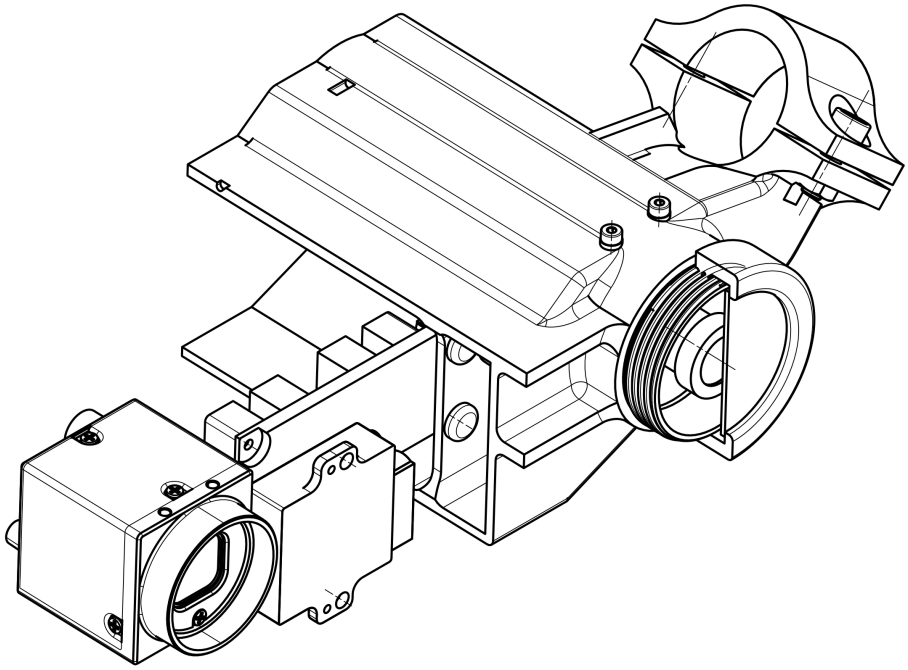


Figure 4.9: Sensors mounting

4.2.6 Position

The system is chosen to be placed above the driver's helmet, at the highest possible position, at the main roll hoop. At this position, cones' overlapping was significantly reduced which facilitated detection. The downside of this position appeared to be the fact that the main roll hoop was absorbing every vibration of the vehicle, which affected the IMU measurements with external "noise". To resolve the issue we filtered the measurements using Low Pass Filtering and we isolated the external vibration sources (water pumps).

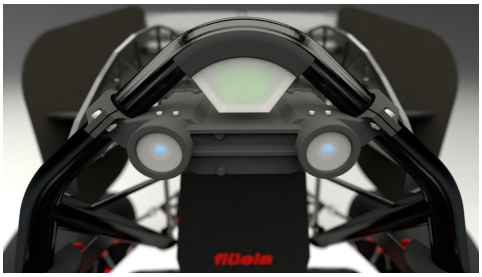


Figure 4.10: Sensors position

4.3 Calibration

To reassure accurate motion estimates and successful detection, we calibrated the sensor setup. Calibration parameters of the stereo camera setup, are essential for depth perception using the stereo triangulation technique. The IMU is already factory calibrated. We need to estimate each camera's intrinsic and stereo setup external parameters. These are determined from a set of stills of a checkerboard, using a dedicated ROS Calibration package[22]. The output is shown on the Appendix 1. The camera calibration is using the pinhole camera model to perform 3D reconstruction[16].

$$s \begin{bmatrix} u \\ v \\ l \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r11 & r12 & r13 & t1 \\ r21 & r22 & r23 & t2 \\ r31 & r32 & r33 & t3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ l \end{bmatrix}$$

(x, y, z): the coordinates of a 3D point in the world coordinate space
(u, v) : the coordinates of the projection point in pixels
(c_x, c_y) : principal point that is usually at the images center
f : focal length in pixel
$\begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$: camera intrinsic parameters matrix
$\begin{bmatrix} r11 & r12 & r13 & t1 \\ r21 & r22 & r23 & t2 \\ r31 & r32 & r33 & t3 \end{bmatrix}$: extrinsic parameters matrix

Extrinsic parameters matrix, on our case, is used to describe the rigid motion of the chessboard in front of our still camera setup. It translates coordinates of a point (X, Y, Z) to a coordinate system, fixed with respect to the camera[16].
Calibrating the cameras, we calculated the radial distortion of our system and we were able to measure depth information, accurately.
Calibration results are shown on Appendix A.

4.4 Synchronization

Triggering is essential for stereo synchronization and timestamping. However, it adds additional delays to the measurements. The cameras are triggered every second, updating the GPS Time reference point for timestamping. They acquire a predetermined number of images in one trigger and are configured in automatic exposure.

4.4.1 Triggering

For triggering the cameras and synchronizing the sensors, we chose hardware triggering from the GPS's Pulse per Second hardware line[28]. It is an output signal at TTL Voltage (3V), directly connected to the PPS pin on the GPS receiver.
The cameras through the GPIO's opto-isolated input, receive the PPS signal and start acquiring and streaming a predetermined number of images[3]. Each camera has its own internal clock and to achieve perfect stereo synchronization we need to measure the delays: propagation and clearing delays.

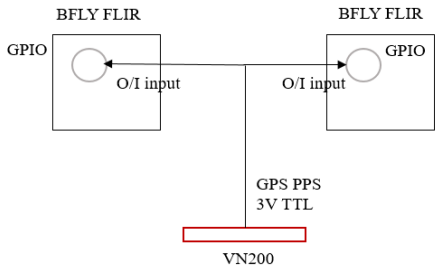


Figure 4.11: Triggering wiring diagram

1. *Propagation delays:* We measured the propagation delay, by configuring one of the camera's GPIO pins to output a strobe pulse and then we connected an oscilloscope up to the input trigger pin and the output strobe[3]. The delays were found to be a few nanoseconds and didn't affect ORBSLAM's performance[30].

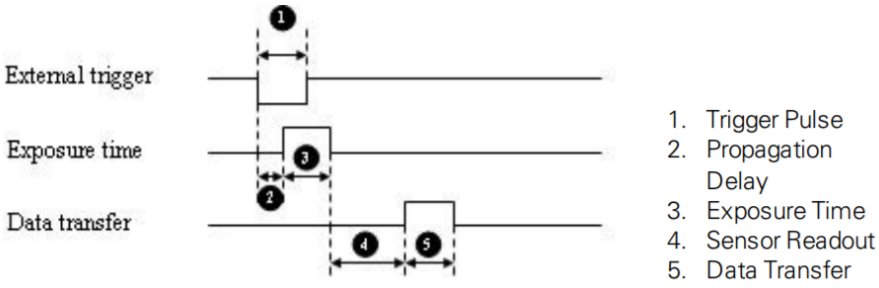


Figure 4.12: Triggering Delays (Source:[3])

$$Timestamp = GPSTime + PropagationDelay$$

2. *Multi-shot Triggering:* After receiving the trigger, the cameras are configured to multi - shot triggering. The number of images acquired is determined by us, from the configuration file of the camera. Once the trigger (PPS) is received, the camera will acquire N number of images, with an exposure time equal to the shutter speed

$$Shutterspeed = 1/N$$

and will stream images to the slave computer at the current frame rate.
We tested multi-shot triggering in different environments and lighting conditions and we observed the problem of frame grabbing inconsistence and drifting, which led to a number of experiments until we found the optimal frame rate and shutter speed to minimize drifting and delays. (Section 4.4)

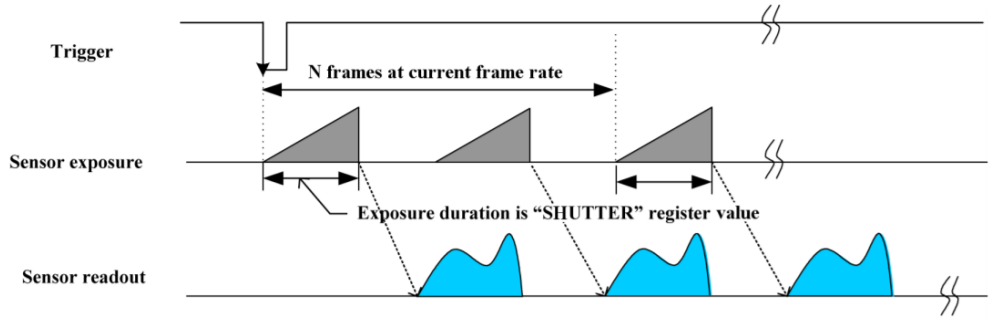


Figure 4.13: Multi-shot triggering (Source:[3])

3. *Camera Behavior between Triggers:* Depending on the frame rate, the camera clears charges from the sensor at the horizontal pixel clock rate. This action takes place after the last shutter integration and until the next trigger is received[3]. This results to a time margin dedicated to this function, which causes a delay between the last shutter integration and the first frame grabbed, of the following trigger.

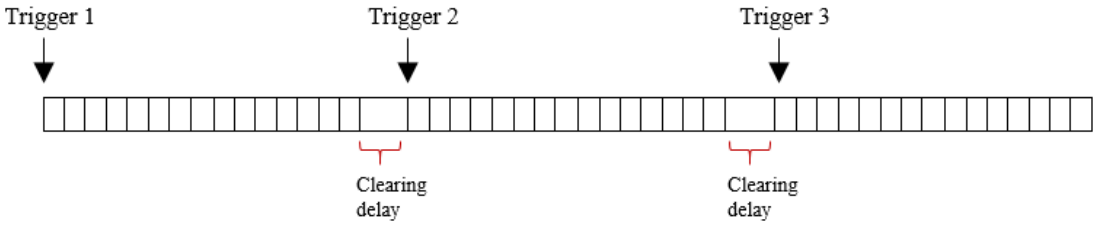


Figure 4.14: Clearing delays

4.4.2 Experiments and Results

On multishot triggering, depending on the weather conditions, cameras shutter speed appeared non consistent and the clearing time was drifting, regardless the convention $Shutterspeed = 1/N$ (Appendix B).
On the following figures, the vertical light blue lines represent the frames grabbed in one recording. The first horizontal line of frames are the images captured from the left camera and the second line are the images

captured from the right camera. The third line are the INS measurements. The vertical lines are placed according their timestamps. Each frame is timestamped and the time it was captured is shown on the diagram. Inconsistent gaps are translated as empty space between frames capturing from the cameras. Gaps more than 0.9 sec are translated as missing trigger.

1. In case of very dim lighting, shutter speed was increasing with respect to the hardware automatic settings of the cameras. This resulted to the cameras missing the next trigger until the N parameter which determined the number of frames grabbed in one trigger, was integrated. When we set the automatic exposure time settings to manual, the images received appeared shaded and underexposed.

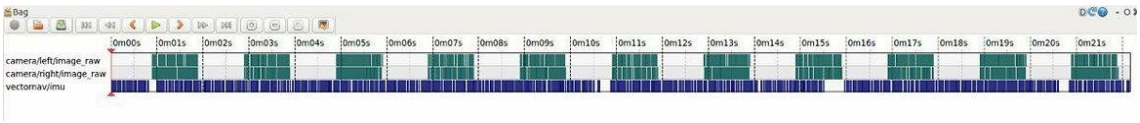


Figure 4.15: 84fps in dim lighting

2. In case of very bright lighting, shutter speed was decreasing, which resulted to an increase of the clearing gap and time drifting. Same as before, we set the automatic exposure time to manual, in order to have the frames following the convention $Shutterspeed = 1/N$, the images received were overexposed.

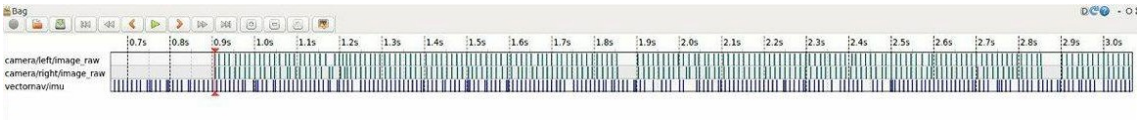


Figure 4.16: 80 fps in bright lighting

4.4.3 Solution

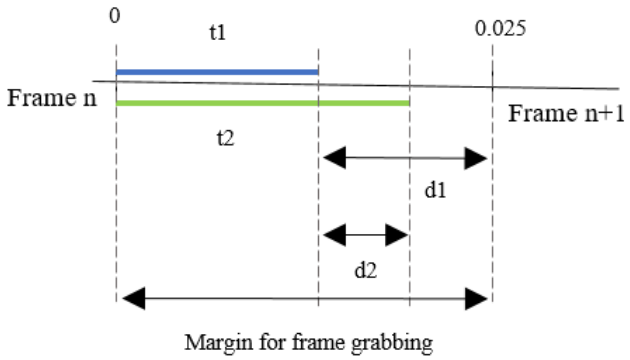
Our goal was to build a reliable system, regardless environment's lighting conditions. Inconsistent and unsynchronized sensor measurements, indicates unpredictable sensor behavior and time drifting. On our project it is essential to know the delays and sensors' internal frame clock, for accurate timestamping.
To resolve the issue, we spent a great number of testing days in different weather conditions, changing the internal parameters and registers of the cameras until we found the optimal exposure time and frame rate which minimized delays between the frames, for every different lighting scenario. The purpose of the experiment was to find the trade-off between properly exposed images and minimizing clearing delay, reassuring trigger reception on every PPS of the GPS. Results:

1. *Extremely Dim Lighting*
   Optimal frame rate: 6 fps
   Exposure time: 0 - 0.14 on automatic mode

2. *Extremely Bright Lighting*
   Optimal frame rate: 84 fps (maximum)
   Exposure time: 0 - 0.011 on automatic mode

3. *Normal Outdoors Lighting*
   Optimal frame rate: 60 fps
   Exposure time: 0-0.016 on automatic mode

Taking these into account, we introduced a method of reassuring frames grabbing consistency, by interfering on the cameras internal settings and parameters. Alternating the configuration file of the ROS Drivers, we set a standard window, in which the cameras were allowed to grab frames. The window size is predetermined and it is the same for every frame. Inside this time window, the cameras depending on the lighting conditions and their internal hardware exposure setting, they adjust their shutter speed, to fit the window.
The margins of this time window are the average time needed for exposure in the two extreme conditions. They also set the min and max margins of the exposure time. We managed to reassure frame grabbing delays consistency, proper exposure, trigger receiving and finally synchronization on the stereo setup.



Automatic exposure depending on the lighting conditions
t1: shutter speed in case of bright lighting (e.g.)
t2: shutter speed in case of dim lighting (e.g.)
d1,d2: delays until next frame which don't affect synchronization or timestamping
Frames are timestamped at the beginning of exposure

Figure 4.17: Stereo Synchronization

Table 4.4: Stereo Configuration

Trigger	ON
Trigger Source	GPIO 0
Trigger Mode	15 (Multishot)
Exposure min	0
Exposure max	2.5
Shutter Speed	[0-0.025]
Mode	Automatic exposure
Parameter	81
Frame Rate	40

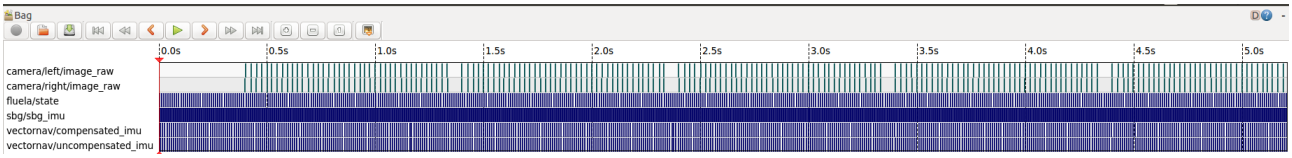


Figure 4.18: Synchronization Result: A picture of a rosbag from the slave computer during testing. The vertical lines are the frames grabbed from each camera, spaced equally on the time line.

4.4.4 Timestamping

The frames are timestamped with respect to the GPS's clock, having the PPS as reference point. On our initial tests, the frames were timestamped on the Computers' clock[6], based on the time the frames were received on the computer, using the embedded ROS time function:
header.timestamp = ros::Time::now()
In order to reduce transportation delays and to be consistent to our system's reference point, we timestamp our measurements to GPS's time. Using the GPS PPS as reference point, the propagation delay and the dropped frames calculator, we were able to time stamp the measurements. The PPS reference point is being updated on every pulse.
$frame_i = PPS + PropagationDelay + 0.025i$
i is equal to the frame counter. In case of a dropped frame, there is an embedded counter c which increases. Before every time stamp, we need to check whether there was a frame drop, by saving the previous value to c_B and comparing it to the next counter value c.
$if\ (c_B = c)$
$frame_i = PPS + PropagationDelay + 0.025 * i;$
$else\ \{$
$i = i + c;\ c = 0;$
$frame_i = PPS + PropagationDelay + 0.025 * i;\ \}$

4.5 Outdoors behavior

Testing the cameras on outdoors conditions, we noticed a significant behavior difference depending on the position of the sun, the clouds and the lighting conditions. The main observations were the intensity and hue changes when driving towards the sun and on opposite direction, which affected the detection drastically.

Figure 4.19: Driving opposite and towards the sun (1:00 pm)

Figure 4.20: Driving opposite and towards the sun (5:00 pm)

4.5.1 Solution

To resolve the issue we attached polarizing filters on the lenses and lenses caps, covering a part of the sky, which appeared overexposed. The result was that the cameras were focusing on the cones and the colors appeared much brighter, increasing intensity and improving detection. We used polarizing filters, in order to darken the sky and decrease reflections from the surface of the vehicle. The most significant contribution, of the filters and the caps, is the reduction of the *sun smearing*. Sun smearing refers to a bright vertical line in the center of the image from top to bottom, emanated from a bright light source, which on our case is the sun[29]. Polarizing filters handle extreme illuminations and lens caps cut out the part of the sky which causes the direct contact of the lens with the light source. Polarizing filters and caps, increased color contrast on the cones and reduced monotonic gray scale transformations.

Figure 4.21: Image captured without filtering (Facing the sun)

Figure 4.22: Image captured with filtering (Facing the sun)

Figure 4.23: Image captured without filtering (Opposite direction)

Figure 4.24: Image captured with filtering (Opposite direction)

Chapter 5

Visual Cone Detection

5.1 Perception Concept

5.1.1 Perception Pipeline

The LiDAR pipeline is capable of performing localization in point cloud and is designed to be accurate and invariant to external conditions. Both the vision and the LiDAR pipeline are independent from each other.[2] The position of each cone's centre is extracted from the cone return point cloud and is being used as a measurement to perform Fast-SLAM[13]. Fast-SLAM is a SLAM technique, which combines Extended Kalman Filters (EKF) with Particle Filters (PF). To reduce computation time and increase robustness, FAST-SLAM uses the odometry estimate from sensor fusion as prior.
The vision pipeline provides a redundant way of detecting cones and generating the track map. ORBSLAM[30] is a state-of-the-art stereo SLAM algorithm which is combined with a monocular visual inertial system. Our goal is to prevent potential failure over non - smooth trajectory. ROVIO[30] is incorporated in the pose graph optimization, improving significantly robustness and diminishing feature association solving time. As prior to ORBSLAM and LiDAR validation reference, we implemented a machine learning cone detection algorithm. The algorithm is based on a boosted cascade of weak classifiers, providing excellent trade-off between accuracy and run time. As opposed to Convolutional-Neural-Networks, the algorithm allows fast evaluation due to its cascade structure, which rejects a patch without cone at a very early stage and proceeds the evaluation of the next patches. It is trained to detect left and right cones independently, using Local Binary Patterns (LBP). When combined, these visual algorithms, build a 3D map of cone landmarks. It is being used for LiDAR's validation reference or as a separate semantic map (SM) in case of LiDAR failure[2]. The cones' positions are given to Fast-SLAM for distinguishing left and right cones in the LiDAR point cloud and for building the final map of the track.

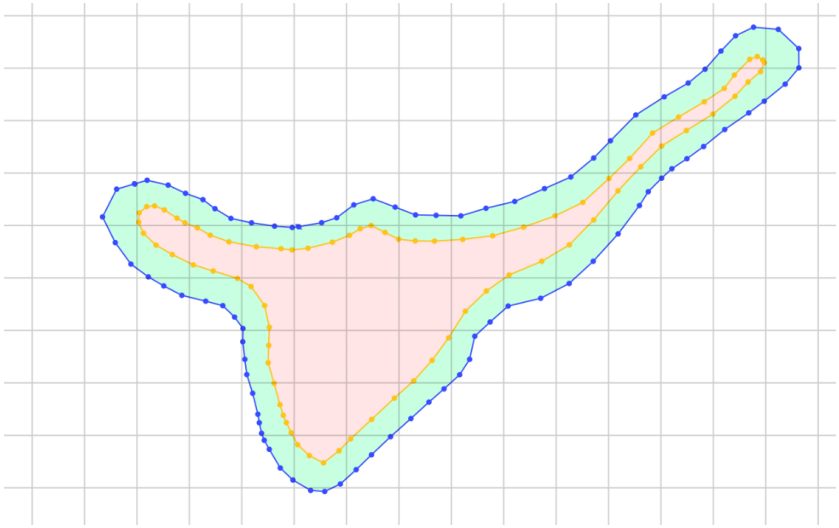


Figure 5.1: Fast-SLAM: boundaries discrimination

5.1.2 Software - Hardware architecture of Vision pipeline

The hardware optimizations made, as described in Chapter 4, were mainly focused on increasing the performance of the visual cone detection algorithm. On the Stereo camera - IMU setup we are only using the left mono - camera, for cone detection. ORBSLAM[20], is using the stereo setup to detect landmarks using stereo triangulation. It only requires the detection from one camera in order to distinguish landmarks as "cones" and "other". The third algorithm of the vision pipeline, ROVIO, is using a mono-camera and the accelerometers and gyroscopes from the IMU. Adding odometry information, we increased the absolute angular and linear velocity, providing robustness. Accurate and robust detection is essential for track mapping, since our system is optimized for reliability.

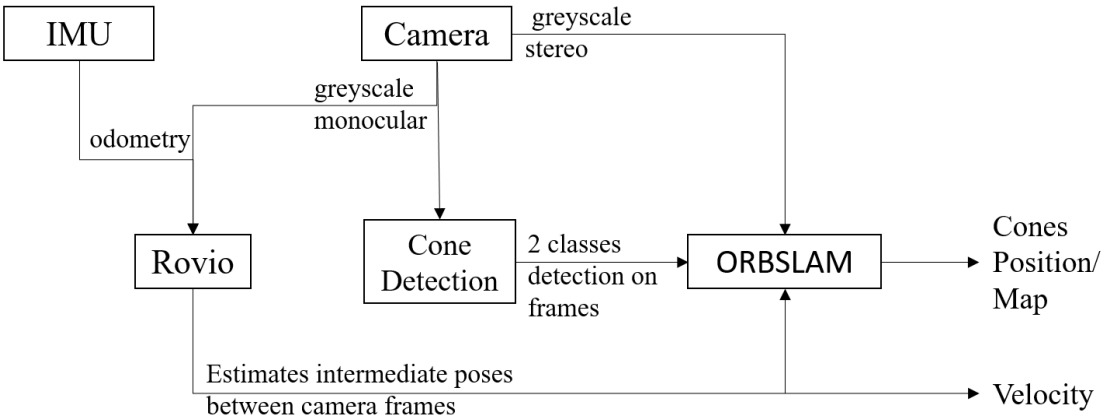


Figure 5.2: Visual Sensor flow

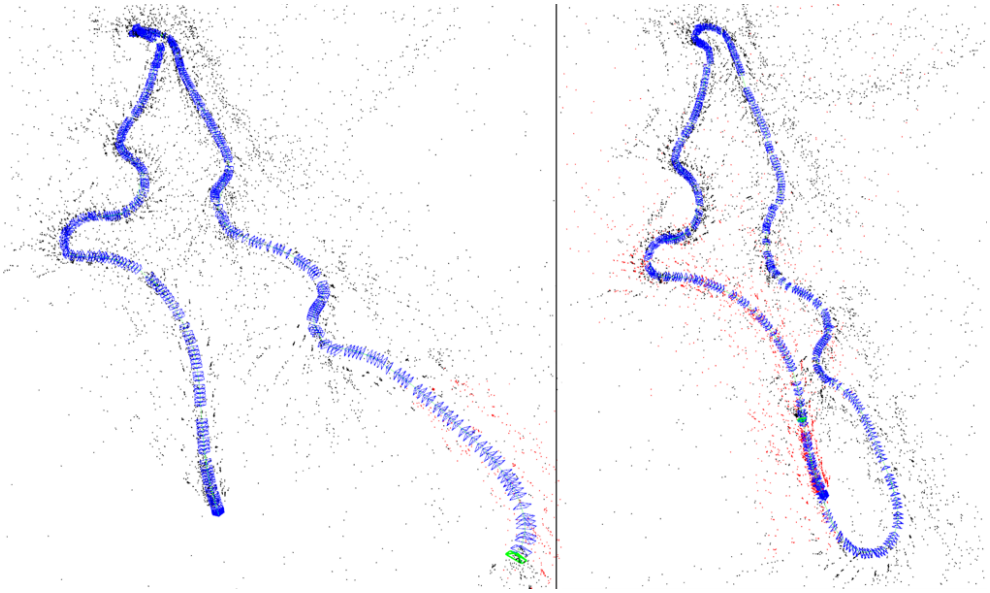


Figure 5.3: ORBSLAM and Rovio right before (left) and right after (right) performing loop-closure

5.1.3 Detection Concept

For cones detection, our goal is to accurately distinguish left and right cones in the track. As described in Chapter 1, during track drive the marking is: blue cones with white stripe are placed on the left side and yellow cones with black stripe (Figure 1.1) are placed on the right side.
We are using gray scale images due to the implementation requirements of our project (Section 1.2.2) Despite not having color information, the detection is achieved due to the characteristic intensity pattern of solid color with a horizontal stripe of another color. This results in a light cone with a dark stripe and a dark cone with a light stripe. The pattern is perfectly distinguishable for the algorithm and quite robust on illumination changes.

5.2 Local Binary Patterns

Local Binary Patterns, introduced by Ojala et al.[15], is a low cost texture descriptor, used for classification. It describes the relationship between a pixel and its neighbors within a specific radius. It is ideal for gray scale images and pattern detection.
In order to build an LBP descriptor, we select for each pixel a neighborhood of size R. LBP uses the central pixel as threshold, measuring the intensity level of its neighbors. Depending on their gray level, the neighbor pixels get the values 1 if it is not less than the threshold and 0, otherwise. From the process we extract a binary sequence which is used to generate a new value for the central pixel through the weighting block. The new values refer to intensity levels which produce an LBP 2^P pixels histogram. The $LBP(x_c, y_c)$ operator of the central pixel, is defined as:

$$LBP(x_c, y_c) = \sum_{P=0}^{P-1} s(g_p - g_c)2^P$$

g_c, g_p: gray level of the central pixel and the p_th neighbor
P : the binary sequence is a p-bit binary number, depending on the radius R of the neighborhood

$$s(g_p - gc) = \left\{ \begin{array}{ll} 1 & , g_p >= g_c \\ 0 & , g_p <= g_c \end{array} \right\}$$

The operator is extended to use a variable neighborhood and performs accurate multi-scale detection. For the central pixel x_c, x_p we determined the position of the neighbor, through calculating:

$$x_p = x_c + Rcos(2\pi p/P)$$

$$y_p = y_c - Rsin(2\pi p/P)$$

The radius R refers to the radius of a circle built from the neighbor blocks. The center of the circle is the center of the central pixel. The points x_p, y_p, are parts of the circle generated.

The objects we want to detect are rigid and axially symmetric. The view point is fixed. The features extracted from LBP have a low-dimensional complexity. We chose LBP because:

- They require less computational power and speed
- The image representation doesn't suffer from illumination variations: it appears robust against monotonic grayscale transformations

5.3 Methodology and Structure

Our Computer Vision algorithms use the Open Computer Vision Library (Section 3.2.3). For object detection, we are using OpenCV's functions and applications, for the software modules: annotation and training. In order to evaluate the performance of our detector, training and testing were emphasized.

5.3.1 Training Data generation

With the camera mounted on the vehicle, we recorded approximately 80 kilometers of vision data during our testing season. For every additional positive and negative set of images, we tested the detection performance in order to reduce false positives and poor detection. Since the detection includes two different classes (left and right cones), the following training steps were performed separately for the two classes.

- *Positive data generation:*
  Using on board camera recordings, we extracted the frames grabbed that contained the features we wanted to detect. Each set was recorded in different environmental and weather conditions. At the end of the season we had generated a file which included images from five different tracks and tarmac, different weather conditions which included rainy and sunny days and different lighting conditions with various sun positions. This file includes 4.000 different images for each class, with the objects appearing in different lighting, scaling, angle and background.

- *Negative data generation:*
  Our goal was to reduce outliers and false positives on our detection. To achieve that we developed a very large set of negatives from images captured from on board recordings and with the sensor hand-held. These include:

  1. The parts of the tracks we didn't want to detect (sky, grass, drown lines on the ground etc).
  2. Different stripe patters similar to the ones of the cones. Images of eyes, faces, windows, shelfs, desks, books, buildings. Anything that had a horizontal lines pattern and could mislead the detection.
  3. Each class includes the other class as negatives. From on board recordings, we chose the frames that only contained the class we didn't want to detect. It is very crucial for our classifiers to be trained against other class detection.
  4. Formula Student tracks have a great number of posters in the walls and lines on the ground. We extracted the frames from the onboard GoPro cameras of the previous years and included those that had the posters and the other cones kind.

  The final negative sets include 10.000 images each.

5.3.2 Annotation

In order to train the cascade classifier, we annotate the positive set of images, to distinguish the objects in the images that we are looking to detect. We used OpenCV annotation tool and we cropped manually the objects, in order to generate a .vec set of data[17]. On every image, we drew rectangles around the cones and stored the data on a text file. This file was later also used to evaluate the detection. In total 4000 images were annotated for each class, building a quite large training set of data. The annotated images included cones in different background, illumination angles, positions and size.

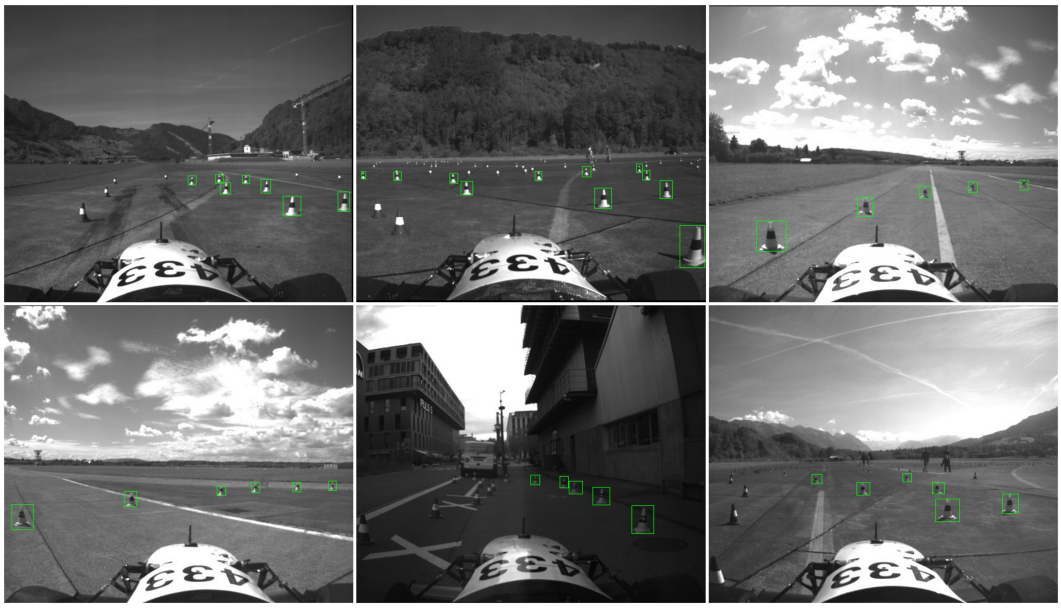


Figure 5.4: Annotated images

5.3.3 Scaling

Our goal was to achieve long distance detection, with respect to the slave computer's computational power. As we were broadening the scaling window, the computational time and complexity was increasing significantly. The slave computer can't process large amount of vision data and to perform detection while running the visual SLAM algorithms in parallel, for detection distances greater than 10 meters. We concluded that the optimal detection in real-time was performed on objects bigger than 20x20 px for 4 different scales. To test the algorithmic complexity and the optimal scaling, we trained on minimum object size 5x5 px and increased until the 25x25 px. Even though we achieved detection at a distance of 20 meters, the slave computer was unable to process in real-time the data received.

5.3.4 Training

After files generation and annotation, we trained the boosted cascade of weak classifiers, on Euler (Erweiterbarer, Umweltfreundlicher, Leistungsfhiger ETH-Rechner) [32]. For training we used the high speed GAB-Gentle AdaBoost, type of boosted classifiers. The final set of images was trained on 45 Stages.
AdaBoost algorithm: A machine learning based approach, with high detection rate and reduced number of outliers[26]. It is based on boosting which combines weak and incorrect information for object detection. AdaBoost generates a strong classifier by extracting the important features from a large data set of visual features. On every stage of training, the weak learners are added to a group and a weighting vector is adjusted depending on the value. This results to higher accuracy classifier than the other known weak classifiers.

5.4 Results

After optimizing the hardware and the training sets for detection, we managed to detect and discriminate the two different classes of cones, in real time. Using Intersection Over Union [24] we annotated a set of positive images and compared the bounding boxes detected from the cone detection algorithm.

$$Detection \quad accuracy : 58\%$$

Figure 5.5: Intersection over Union.

Chapter 6

Conclusion

6.1 Evaluation

In order to evaluate the performance of the trained cascade and the algorithmic implementation, we evaluated the output of the vision pipeline. This includes the evaluation of the following features:

1. *Loop closure*, indicates the synchronization and the odometry measurements accuracy of the visual inertial system.

2. *Trajectory precision*, from the output track map and odometry we compare the semantic map generated with the one from FAST-Slam

3. *Features discrimination*, detection evaluation based on proper classification of visual SLAM features, into "cone left", "cone right" and "other".

4. *Features detection*, another detection evaluation based on the Intersection over Union (IoU) between two boxes A and B. We keep the annotation file of a set of images and extract the width, height, x and y parameters of the detected bounding box. Using Intersection Over Union (IoU)[24], we were able to evaluate our detection, based on:
$IOU(A, B) = \frac{area(A \cap B)}{area(A \cup B)}$

6.2 Discussion

During the testing season, we didn't have the opportunity to test the vision pipeline on extreme weather conditions. To reassure accuracy and robustness in case of single mode failures, the visual SLAM system is optimized to achieve track mapping and loop closure, regardless the false positives or the detection's inefficiency on detecting either left or right. During Formula Student Germany, we experienced extreme weather conditions which resulted to very dim lighting and extreme rainfall, which we hadn't experienced before.
This resulted to unexpected behavior of the detection algorithm. This was first observed during testing, before the race. On-site and with limited time available, we used our algorithm's main strong points and increased the detection window by reducing the neighbors of the LBP cascade, in order for the algorithm to output every patch detected, including a large number of false positives. Even though this increased radically the number of outliers, ORBSLAM was able to handle it by performing outliers rejection when there weren't

any landmarks detected in the area determined by the bounding box. It was feeding information to FAST-Slam and building an accurate semantic map regardless the conditions, as long as it received left cone and/or right cone detection input.
During track drive we achieved loop closure and the map was very similar to the one that LiDAR pipeline built, which proves that the vehicle would have been able to drive with the vision's SM in case of single mode failures. On the downside thought, detection was not accurate in these lighting conditions. When driving towards the sun, it was detecting the left cones perfectly, while when driving on opposite direction, it was detecting the right cones. Due to our system's focus on robustness, this didn't affect the produced SM, but requires further optimization.

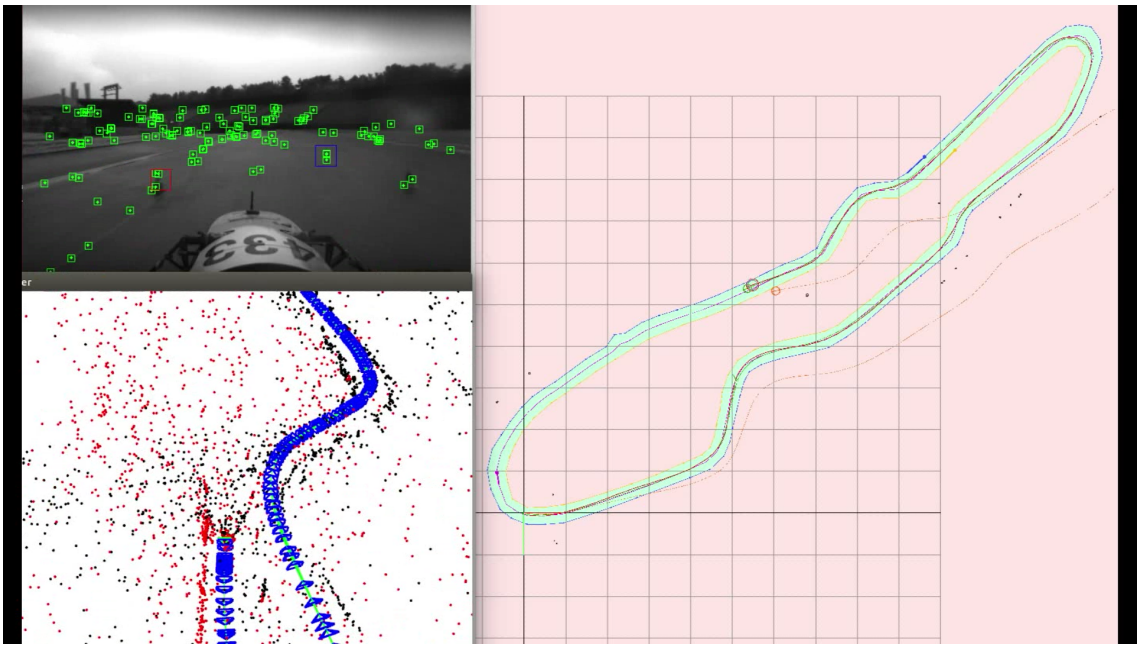


Figure 6.1: On top left, the green rectangles are the landmarks detected by ORBSLAM and the red and blue rectangles are the cone detection (left and right). Bottom left, is the SM while being generated from the vision pipeline. On the right, is the SM generated by FAST-Slam

6.3 Conclusions

Every large scale application needs to be optimized for the environment and purpose of its development. Our computer vision pipeline had to be optimized, to fulfill our system's requirements. At the beginning, we set our system's goals to be reliability and robustness, while building a competitive vehicle. Due to the extreme weather conditions, during the competition, we experienced a great number of failures from our sensors, but our algorithms and concept were able to handle the unpredictable behaviors. The overall project was a team effort from a quite small group of software engineers.
Regarding the vision pipeline, for future applications, the author would suggest the use of a powerful GPU for the vision pipeline and the purchase of a higher-resolution camera or an event based camera.

Appendix A

Calibration parameters

width: 648
height: 488

503.407456	0.000000	307.996893
0.000000	503.139436	260.805020
0.000000	0.000000	1.000000

Table A.1: left frame: camera matrix

-0.404556	0.141767	0.000202	0.003240	0.000000

Table A.2: left frame: distortion

0.999497	0.008791	-0.030463
-0.008759	0.999961	0.001212
0.030472	-0.000945	0.999535

Table A.3: left frame: rectification

463.736586	0.000000	319.193577	0.000000
0.000000	463.736586	257.227137	0.000000
0.000000	0.000000	1.000000	0.000000

Table A.4: left frame: projection

503.923572	0.000000	316.239024
0.000000	504.254784	255.514433
0.000000	0.000000	1.000000

Table A.5: right frame: camera matrix

-0.422678	0.156832	-0.003953	-0.006842	0.000000

Table A.6: right frame: distortion

0.999910	0.013052	-0.003200
-0.013056	0.999914	-0.001058
0.003185	0.001099	0.999994

Table A.7: right frame: rectification

463.736586	0.000000	319.193577	-51.118513
0.000000	463.736586	257.227137	0.000000
0.000000	0.000000	1.000000	0.000000

Table A.8: right frame: projection

Appendix B

Stereo camera synchronization

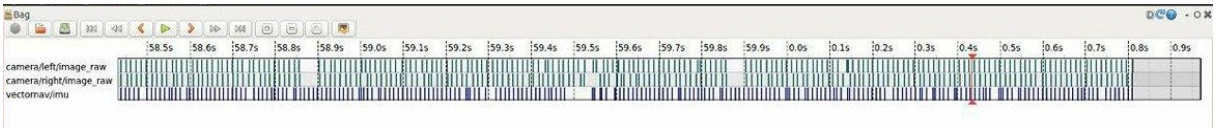


Figure B.1: Bright lighting: 78 fps

Figure B.2: Bright lighting: 80 fps

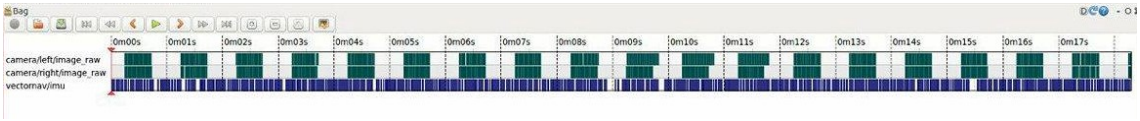


Figure B.3: Indoors lighting: 83 fps. Result: trigger missing

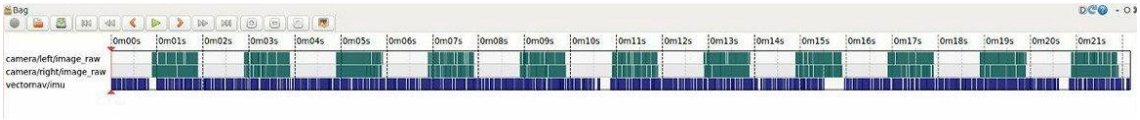


Figure B.4: Indoors lighting 83 fps. Result: trigger missing

Appendix C

Datasheets

TECHNICAL SPECIFICATIONS

Navigation

Horizontal Position Accuracy:	2.5 m RMS
Horizontal Position Accuracy (w/SBAS):	2.0 m RMS
Vertical Position Accuracy:	5.0 m RMS
Vertical Position Accuracy (w/Barometer):	2.5 m RMS
Velocity Accuracy:	±0.05 m/s
Dynamic Accuracy (Heading, True Inertial):	0.3 ° RMS
Dynamic Accuracy (Pitch/Roll):	0.1 ° RMS
Static Accuracy (Heading, Magnetic)[1]:	2.0 ° RMS
Static Accuracy (Pitch/Roll):	0.5 ° RMS
Angular Resolution:	< 0.05 °
Repeatability:	< 0.1 °
Max Output Rate (IMU Data)[2]:	1 kHz
Max Output Rate (Navigation Data):	400 Hz

Gyro

Range:	±2000 °/s
In-Run Bias Stability:	< 10 °/hr
Linearity:	< 0.1 % FS
Noise Density:	0.0035 °/s /√Hz
Bandwidth:	256 Hz
Alignment Error:	±0.05 °

Accelerometer

Range:	±16 g
In-Run Bias Stability:	<0.04 mg
Linearity:	< 0.5 % FS
Noise Density:	0.14 mg/√Hz
Bandwidth:	260 Hz
Alignment Error:	±0.05 °

Magnetometer

Range:	±2.5 Gauss
Linearity:	< 0.1 %
Noise Density:	140 µGauss/√Hz
Bandwidth:	200 Hz
Alignment Error:	±0.05 °

GPS

Receiver Type:	50 Channels, L1 GPS C/A Code
Solution Update Rate:	5 Hz
Time-to-First-Fix (Cold/Warm Start):	36 s
Time-to-First-Fix (Hot Start):	< 1 s
Altitude Limit:	50,000 m
Velocity Limit:	500 m/s

Pressure Sensor

Range:	10 to 1200 mbar
Resolution:	0.042 mbar
Accuracy:	±1.5 mbar
Error Band:	±2.5 mbar
Bandwidth:	200 Hz

Environment

Operating Temp:	-40°C to +85°C
Storage Temp:	-40°C to +85°C

Electrical:	SMD	Rugged
Input Voltage:	3.2 V to 5.5 V	3.3 V to 17 V
Current Draw[3]:	105 mA @ 3.3 V	80 mA @ 5 V
Max Power Consumption[3]:	445 mW	500 mW
Digital Interface:	Serial TTL, SPI	Serial TTL, RS-232

Physical:	SMD	Rugged
Size:	24 x 22 x 3 mm	36 x 33 x 9.5 mm
Weight:	4 g	16 g
Connector:	30-pin LGA	10-pin Harwin
GPS Antenna Connector:	U.FL	MMCX

[1] With proper magnetic declination, suitable magnetic environment and valid hard/soft iron calibration.
[2] Default 800 Hz.
[3] Not including active antenna power consumption.

Specifications

Model	Version	MP	Imaging Sensor
BFLY-U3-03S2C-CS **BFLY-U3-03S2M-CS**	Color Mono	0.3 MP	■ Sony ICX424, 1/3", 7.4 µm ■ Global shutter ■ 84 FPS at 648 x 488
BFLY-U3-05S2C-CS **BFLY-U3-05S2M-CS**	Color Mono	0.5 MP	■ Sony ICX693, 1/3", 6.0 µm ■ Global shutter ■ 50 FPS at 808 x 608
BFLY-U3-13S2C-CS **BFLY-U3-13S2M-CS**	Color Mono	1.3 MP	■ Sony ICX445, 1/3", 3.75 µm ■ Global shutter ■ 30 FPS at 1288 x 964
BFLY-U3-20S4C-CS **BFLY-U3-20S4M-CS**	Color Mono	2.0 MP	■ Sony ICX274, 1/1.8", 4.4 µm ■ Global shutter ■ 15 FPS at 1624 x 1224
BFLY-U3-23S6C-C **BFLY-U3-23S6M-C**	Color Mono	2.3 MP	■ Sony IMX249, 1/1.2", 5.86 µm ■ Global shutter ■ 41 FPS at 1920 x 1200
BFLY-U3-50H5C-C **BFLY-U3-50H5M-C**	Color Mono	5.0 MP	■ Sharp RJ32S3AA0DT, 2/3", 3.45 µm ■ Global shutter ■ 7.5 FPS at 2448 x 2048

Imaging Performance (EMVA 1288)	See the Imaging Performance Specification, which includes quantum efficiency, saturation capacity (full well depth), read noise, dynamic range and signal to noise ratio.
A/D Converter	12-bit; 10- and 12-bit (BFLY-U3-23S6)
Video Data Output	8, 12, 16 and 24-bit digital data
Image Data Formats	Mono8, Mono12, Mono16, Raw8, Raw12, Raw16 (all models); RGB, YUV411, YUV422, YUV 444 (color models)
Partial Image Modes	Pixel binning, decimation, and region of interest (ROI) modes
Image Processing	Gamma, lookup table, hue, saturation, and sharpness
Shutter	Global shutter; Automatic/manual/one-push See product webpage for specific model's range
Gain	Automatic/manual/one-push See product webpage for specific model's range
Gamma	0.50 to 3.99, programmable lookup table
White Balance	Automatic/manual/one-push
Color Processing	On-camera in YUV or RGB format, or on-PC in Raw format
Digital Interface	USB 3.0 interface with screw locks for camera control, data, and power
Transfer Rates	5 Gbit/s
GPIO	6-pin Hirose HR10A-7R-6PB GPIO connector for trigger, strobe, and power
External Trigger Modes	Standard, bulb, low smear, overlapped, multi shot
Image Buffer	16 MB frame buffer
Memory Channels	2 user configuration sets for custom camera settings
Flash Memory	1 MB non-volatile memory
Dimensions	29 mm x 29 mm x 30 mm excluding lens holder, without optics (metal case)
Mass	36 g (without optics)
Power Consumption	5-24 V via GPIO or 5 V via USB 3.0 interface, maximum <3 W
Machine Vision Standard	USB3 Vision
Camera Control	Via FlyCapture SDK or USB Vision third party software
Camera Updates	In-field firmware updates
Lens Mount	CS-mount; C-mount (BFLY-U3-23S6 and BFLY-U3-50H5)
Temperature	Operating: 0° to 45°C; Storage: -30° to 60°C
Humidity	Operating: 20 to 80% (no condensation); Storage: 20 to 95% (no condensation)
Compliance	CE, FCC, KCC, RoHS
Operating System	Windows, Linux (32- and 64-bit)
Warranty	3 years

FLIR Integrated Imaging Solutions

CANADA
12051 Riverside Way
Richmond, BC, Canada
V6W 1K7
T: +1 866.765.0827 (toll free)
T: +1 604.242.9937
F: +1 604.242.9938
E: mv-sales@flir.com
www.flir.com/iis

USA
T: +1 866.765.0827 (toll free)
E: mv-na-sales@ptgrey.com

EUROPE
T: +49 7141 488817-0
F: +49 7141 488817-99
E: mv-eusales@flir.com

CHINA
T: +86 10 8215 9938
F: +86 10 8215 9936
E: mv-chinasales@flir.com

ASIA
E: mv-asiasales@ptgrey.com

For a full list of international distributors and offices visit www.flir.com/contact-us

Bibliography

[1] Git – local branching on a cheap. https://git-scm.com/.

[2] Autonomous Design Report. AMZ Racing, ETH Zürich, 2017.

[3] *FLIR Blackfly USB3 Vision, Technical Reference*, 2017.

[4] *Formula Student Rules 2017*. Formula Student Germany, 2017.

[5] L. G. Trabasso A. M. Kabayama1. Performance evaluation of 3D computer vision techniques. Instituto Tecnolgico de Aeronutica Diviso de Engenharia Mecnica-Aeronutica, 2002.

[6] David Ball Ben Upcroft Andrew English, Patrick Ross and Peter Corke. TriggerSync: A Time Synchronisation Tool . Australian Research Council Linkage Project LP110200375 Robotics for Zero Tillage Agriculture".

[7] Ross Girshick. Fast R-CNN. 2016.

[8] Michael Burri Pascal Gohl Stefan Leutenegger Paul T. Furgale Janosch Nikolic, Joern Rehder and Roland Siegwart. A Synchronized Visual-Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM. IEEE International Conference on Robotics and Automation (ICRA), 2014.

[9] Ross Girshick Ali Farhadi Joseph Redmon, Santosh Divvala. You Only Look Once: Unified, Real-Time Object Detection. 2016.

[10] Youji Okawa Kajiro Watanabe, Kazuyuki Kobayashi and Akira Kikuchi. Absolute speed measurement of vehicles. Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON '91., 1991 International Conference on, 1991.

[11] Larry Li. Time-of-Flight Camera An Introduction. Technical report, 2014.

[12] M. Hutter M. Bloesch, S. Omari and R. Siegwart. Robust visual inertial odometry using a direct EKF-based approach. IEEE International Conference on Intelligent Robots and Systems, vol. 2015-Decem, pp. 298304, 2015.

[13] Daphne Koller Ben Wegbreit Michael Montemerlo, Sebastian Thrun. Fast: SLAM, A Factory Solution to the Simultaneous Localization and Mapping Problem.

[14] R. Mur-Artal and J. D. Tardos. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. 2016.

[15] T. Ojala and M. Pietikainen. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns . IEEE Trans on Pattern Analysis and Machine Intelligence, 2002.

[16] OpenCV. Camera calibration and 3D Reconstruction. http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.

[17] OpenCV. Cascade Classifier Training. http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html.

[18] OpenCV. Open Source Computer Vision Library. http://opencv.org/.

[19] Michael Jones Paul Viola. Rapid Object Detection using a Boosted Cascade of Simple Features. Accepted conference on computer vision and pattern recognition, 2001.

[20] J. M. M. Montiel R. Mur-Artal and J. D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics Vol. 31, no. 5, 2015.

[21] Z. Richard, Hartley; Andrew. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2004.

[22] Robot Operating System (ROS). Camera calibration. http://wiki.ros.org/camera_calibration.

[23] Robot Operating System (ROS). Powering the word's robotics. http://www.ros.org/.

[24] Adrian Rosebrock. Intersection Over Union (IOU) for object detection. http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[25] SAE. *2017-18 Formula SAE Rules*. SAE International, 2017.

[26] Robert E. Schapire. Explaining AdaBoost. Princeton University, Dept. of Computer Science.

[27] Ross Girshick Shaoqing Ren, Kaiming He and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks . 2016.

[28] VectorNav Embedded Navigation Solutions. *VN-200 User Manual*.

[29] Taylor and Francis group. *Image Sensors and Signal Processing for Digital Still Cameras* . CRC Press, 2005.

[30] Graf Ueli. SLAM Integration of Robust Visual-Inertial Odometry . ETH Zürich, 2017.

[31] Marin G. Dal Mutto C. Dominio F. Minto L. Cortelazzo G.M. Zanuttigh, P. *Time-of-Flight and Structured Light Depth Cameras Technology and Applications*. 2016.

[32] ETH Zürich. Euler ScientificComputing the scientific computing wiki. https://scicomp.ethz.ch/wiki/Euler.