Extrinsic and temporal calibration of heterogeneous mobile robot exteroceptive sensor systems

Peršić, Juraj

Doctoral thesis / Disertacija

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:186549

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Juraj Peršić

EXTRINSIC AND TEMPORAL CALIBRATION OF HETEROGENEOUS EXTEROCEPTIVE MOBILE ROBOT SENSOR SYSTEMS

DOCTORAL THESIS

Zagreb, 2021

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Juraj Peršić

EXTRINSIC AND TEMPORAL CALIBRATION OF HETEROGENEOUS EXTEROCEPTIVE MOBILE ROBOT SENSOR SYSTEMS

DOCTORAL THESIS

Supervisor: Professor Ivan Petrović, PhD

Zagreb, 2021

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Juraj Peršić

EKSTRINZIČNO I VREMENSKO UMJERAVANJE HETEROGENIH EKSTEROCEPCIJSKIH SENZORSKIH SUSTAVA MOBILNIH ROBOTA

DOKTORSKI RAD

Mentor: prof. dr. sc. Ivan Petrović

Zagreb, 2021.

Doctoral thesis was written at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering.

Supervisor: Professor Ivan Petrović, PhD

Thesis contains 121 pages

Thesis no.:

ABOUT THE SUPERVISOR

IVAN PETROVIĆ received B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 1983, 1989 and 1998, respectively. For the first ten years after graduation, he was with the Institute of Electrical Engineering of Končar Corporation in Zagreb, where he had been working as a research and development engineer for control and automation systems of electrical drives and industrial plants. From 1994 he has been working at the Department of Control and Computer Engineering at FER, where he is currently a Full Professor with tenure. He has actively participated as a collaborator or principal investigator on 40 national and 25 international scientific projects, where from them eight are funded form FP7 and Horizon 2020 framework programmes. He is also co-director of the Centre of Research Excellence for Data Science and Advanced Cooperative Systems. He published more than 60 papers in scientific journals and more than 200 papers in proceedings of international conferences in control engineering and automation applied to control mobile robots and vehicles, power systems, electromechanical systems and other technical systems.

Prof. Petrović is a member of IEEE, Croatian Academy of Engineering (HATZ), chair of the Technical committee on Robotics of the International Federation of Automatic Control (IFAC), a permanent board member of the European Conference on Mobile Robots, an executive committee member of the Federation of International Robot-soccer Association (FIRA), and a founding member of the iSpace Laboratory Network. He is also a member of the Croatian Society for Communications, Computing, Electronics, Measurements and Control (KoREMA) and Editor-in-Chief of the Automatika journal. He received the award "Professor Vratislav Bedjanič" in Ljubljana for outstanding M.Sc. thesis in 1990 and silver medal "Josip Lončar" from FER for outstanding Ph.D. thesis in 1998. For scientific achievements he received the award "Rikard Podhorsky" from the Croatian Academy of Engineering (2008), "National Science Award of the Republic of Croatia" (2011), the gold plaque "Josip Lončar" (2013), "Science Award" from FER (2015) and award "Nikola Tesla" from IEEE Croatia Section (2019).

O MENTORU

IVAN PETROVIĆ diplomirao je, magistrirao i doktorirao u polju elektrotehnike na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER), 1983., 1989. odnosno 1998. godine. Prvih deset godina po završetku studija radio je na poslovima istraživanja i razvoja sustava upravljanja i automatizacije elektromotornih pogona i industrijskih postrojenja u Končar - Institutu za elektrotehniku. Od svibnja 1994. radi u Zavodu za automatiku i računalno inženjerstvo FER-a, gdje je sada redoviti profesor u trajnome zvanju. Sudjelovao je ili sudjeluje kao suradnik ili voditelj na 40 domaćih i 25 međunarodnih znanstvenih projekata, od čega osam projekata iz programa FP7 i Obzor 2020. Nadalje, suvoditelj je *Znanstvenog centra izvrsnosti za znanost o podatcima i kooperativne sustave*. Objavio je više od 60 znanstvenih radova u časopisima i više od 200 znanstvenih radova u zbornicima skupova u području automatskog upravljanja i estimacije s primjenom u upravljanju mobilnim robotima i vozilima te energetskim, elektromehaničkim i drugim tehničkim sustavima.

Prof. Petrović član je stručne udruge IEEE, Akademije tehničkih znanosti Hrvatske (HATZ), predsjednik tehničkog odbora za robotiku međunarodne udruge IFAC, stalni član upravnog tijela European Conference of Mobile Robots, član izvršnog odbora međunarodne udruge FIRA, suutemeljitelj međunarodne udruge „The iSpace Laboratory Network". Član je i upravnog odbora Hrvatskog društva za komunikacije, računarstvo, elektroniku, mjerenja i automatiku (KoREMA) te glavni i odgovorni urednik časopisa Automatika. Godine 1990. primio je u Ljubljani nagradu „Prof. dr. Vratislav Bedjanič" za posebno istaknuti magistarski rad, 1998. srebrnu plaketu "Josip Lončar" FER-a za posebno istaknutu doktorsku disertaciju, a za znanstvena je postignuća dobio 2008. godine nagradu „Rikard Podhorsky" Akademije tehničkih znanosti Hrvatske, 2011. godine „Državnu nagradu za znanost", 2013. godine zlatnu plaketu "Josip Lončar" FER-a, 2015. godine nagradu za znanost FER-a te 2019. godine nagradu „Nikola Tesla" Hrvatske sekcije IEEE.

ACKNOWLEDGEMENTS

I have entered the world of science without hesitation. I knew very well that I was drawn to it, but I did not know what awaited me there. In hindsight, I realise how little I knew about research, but thanks to all the support I kept receiving, it was not difficult to find my way. I am deeply convinced that for any success, the people you surround yourself with are the most important factor. Therefore, first and foremost, I am extremely grateful to my PhD supervisor Prof. Ivan Petrović for founding and leading the *LAMOR* group, an environment where many exceptional people have been doing great research for many years. I would also like to thank him for all the insightful comments and suggestions, friendly atmosphere and complete freedom in my research. Furthermore, I would like to extend my sincere gratitude to my unofficial mentor Assoc. Prof. Ivan Marković, who has always guided me with his valuable advice.

I am deeply grateful to my friends and colleagues in the *LAMOR* group for a friendly atmosphere, fruitful discussions and support in challenging times. From a long list of dear people, I would like to offer my special thanks to Antea, Filip, Luka P., Tomislav, Josip, Kruno and Luka F. In addition, I would also like to thank all my friends outside the work environment who supported me during my studies.

I would like to express my sincere gratitude to Ivan Hrvoić and Prof. Dr. Sc. Jasna Šimuić-Hrvoić Foundation for the opportunity to meet and collaborate with people from *STARS* laboratory in Toronto, Canada. I would also like to extend my gratitude to Prof. Jonathan Kelly and Emmett Wise from the *STARS* laboratory for interesting, pleasant and successful collaboration. Besides my friends from academia, I would like to thank my friends from industry who gave me a different perspective on our research field. I would like to express my greatest appreciation to colleagues from *Končar - Institut za elektrotehniku d.d.* for successful collaboration on the SafeTram project. Lastly, I owe a very important debt to my teammates at *Motional*, especially to my mentor Francisco Suárez-Ruiz whose insightful advice helped me to find my way after the PhD studies.

My deep appreciation goes to my mother Marica, father Zlatko and brother Ivan who were a constant source of support and love.

Zagreb, 16th July 2021.

ABSTRACT

Robust environment perception of a mobile robot strongly relies on fusion of multiple heterogeneous sensors. Sensor fusion algorithms aim to harness all the valuable information from different sensor modalities, while circumventing their weaknesses. However, to achieve that goal, proper sensor calibration is essential. It can be achieved with offline (target-based) methods or online by relying on information from the environment. Regardless of the approach, it should provide internal sensor parameters, i.e. intrinsic calibration, accompanied with spatial and temporal relations between the sensors, i.e. extrinsic and temporal calibration. This thesis aims to solve extrinsic and temporal calibration of radar – camera – lidar systems in both offline and online manner.

Extrinsic calibration tries to find transform between coordinate frames of two or more sensors. This problem is more complicated with heterogeneous sensors, due to different operating principles of the sensors and subsequently different types of data they produce. It is essential to find correct correspondences which are then used in the next step, estimation of extrinsic parameters. One of the strategies that enables robust correspondence registration is calibration based on a special target. This thesis proposes a novel calibration target suitable for accurate 6 degrees of freedom calibration of radar – camera – lidar systems. Furthermore, measurements of the target enable two-step optimization which leads to accurate extrinsic calibration. While the first step is rather standard reprojection error optimization, a novel second step based on radar cross section (RCS) is proposed. It exploits newly discovered effect of radar's RCS estimation error related to the elevation angle.

Temporal calibration tries to align timestamps of multiple sensors based on comparison of their measurements. It requires motion, either of the sensor systems or an object that the sensor system perceives. This thesis proposes a method for temporal calibration based on moving target tracking thus enabling temporal calibration of radars with other sensors such as cameras and lidars. The backbone of the proposed approach are Gaussian Processes used for continuous-time trajectory representation. It is shown that continuous-time representation is essential for accurate temporal calibration since it enables theoretically grounded temporal correspondence registration between asynchronous sensors with different frame rates. Furthermore, a novel joint spatiotemporal calibration is proposed that owes its efficiency to the Exactly Sparse Gaussian Process Regression and on-manifold optimization. Developed method enables efficient and accurate multisensor calibration that is applicable to a wide range of sensors.

Online calibration uses information from the environment to generate correspondences

between the sensors, thus avoiding specialized targets. This thesis proposes a novel method for online calibration based on moving object tracking applied to radar – camera – lidar systems. The method builds upon the standard detection and tracking pipeline of any autonomous stack that is performed for each sensor separately. It adds a calibration-agnostic track-to-track association scheme that works well under miscalibration. Furthermore, lightweight online decalibration detection scheme is proposed based on analytical pairwise calibration solution. Lastly, complete recalibration of the system is achieved through graph-based multisensor calibration. Combination of the proposed target-based and targetless methods enables a complete solution to calibration of radar – camera – lidar sensor systems.

KEYWORDS: sensor calibration, extrinsic calibration, temporal calibration, moving object tracking, calibration target, radar, lidar, camera, identifiability, Fisher Information Matrix, Gaussian Processes, Lie groups, on-manifold optimization

SAŽETAK

EKSTRINZIČNO I VREMENSKO UMJERAVANJE HETEROGENIH EKSTEROCEPCIJSKIH SENZORSKIH SUSTAVA MOBILNIH ROBOTA

Robusna percepcija okoline jedan je od preduvjeta koje autonomni mobilni robot ili vozilo mora ispuniti. Kako bi se postigao taj cilj, koriste se razni senzori poput kamera, radara, lidara i inercijalnih mjernih jedinica, a njihove se informacije često integriraju. Temeljni zadaci kao što su istodobna lokalizacija i kartiranje (SLAM), otkrivanje i praćenje gibajućih objekata te odometrija često se unaprjeđuju fuzijom više senzora. Temeljni korak u procesu fuzije jest umjeravanje senzora, intrinzično, ekstrinzično i vremensko. Intrinzičnim umjeravanjem pronalaze se unutarnji parametri pojedinog senzora (npr. žarišna udaljenost kamere ili pomak u lidarovim mjerenjima udaljenosti), dok ekstrinzično umjeravanje daje relativnu transformaciju iz koordinatnog sustava jednog senzora u drugi. Vremensko umjeravanje senzora podrazumijeva usklađivanje satova senzora pri čemu se određuje pomak između satova kao i njihova različita frekvencija.

Metode umjeravanja zahtijevaju povezivanje podudarajućih značajki u mjerenjima senzora, što je jedan od glavnih izazova u ekstrinzičnom i vremenskom umjeravanju heterogenih senzora jer različiti senzori koriste različita fizikalna načela pri mjerenju. Povezane značajke u mjerenjima mogu potjecati iz dvaju izvora: (i) predodređene mete za umjeravanje ili (ii) značajki iz okoline. Nakon pronalaska podudarajućih značajki provode se optimizacijski koraci za estimaciju parametara umjeravanja. Dok neke metode zahtijevaju intrinzično umjerene senzore za pronalaženje ekstrinzičnog i vremenskog umjeravanja, druge metode obavljaju optimizaciju nad objema skupinama parametara istodobno. Metode za umjeravanje obično pokušavaju zadovoljiti neka geometrijska ograničenja minimiziranjem reprojekcijske pogreške specifične za problem. Uspjeh optimizacije uvelike ovisi o prikupljenim podacima. Važan korak prije prikupljanja podataka jest određivanje minimalnih zahtjeva na skup podataka koji osiguravaju identifikabilnost problema (ili osmotrivost u slučaju dinamičkih sustava). Neke metode pristupaju problemu identifikabilnosti s geometrijskog stajališta, dok druge to čine iz okvira nelinearne osmotrivosti ili pomoću statističkih alata kao što je Fisherova informacijska matrica. Kroz disertaciju je razvijeno nekoliko metoda za ekstrinzično i vremensko umjeravanje eksterocepcijskih senzora koji se uobičajeno koriste na mobilnim robotima: radari, lidari i kamere. Razvijena je nova univerzalna meta za umjeravanje koja omogućuje umjeravanje spomenutih senzora u svih 6 stupnjeva slobode. Pored toga, ostvareno je i vremensko umjeravanje zasnovano na gibanju spomenute

mete. Na kraju, razvijeni matematički okvir primijenjen je u *online* okruženju gdje je sustav umjeren koristeći informacije o gibajućim objektima u radnom prostoru robota.

Svojstva dobro osmišljene mete su (i) lakoća otkrivanja i (ii) visoka točnost lokalizacije za sve senzore koji se umjeravaju. Prvo svojstvo osigurava uspjeh pronalaženja podudarajućih značajki, dok drugo svojstvo ima veliki utjecaj na kvalitetu rezultata dobivenih optimizacijskim postupkom. Nadalje, ako je dostupno apriorno znanje o meti, metoda umjeravanja ga može iskoristiti kako bi poboljšala preciznost. Eksteroceptivni senzori koji se koriste u robotici koriste raznolike fizikalne pojave kako bi dobili informacije o okolini. Zbog raznih tipova podataka dobivenih heterogenim senzora, postoji mnogo različitih meta za umjeravanje koje moraju zadovoljiti sve potrebe sustava koji se umjerava.

Vremensko umjeravanje podrazumijeva usklađivanje satova različitih senzora kako bi se njihova mjerenja mogla ispravno upariti. Senzori mogu koristiti vlastite satove ili zajednički sat na centralnom računalu. Kada senzori koriste razdvojene satove, potrebno je odrediti vremenski pomak između njih, kao i razliku između njihovih frekvencija. Problem različitih frekvencija moguće je riješiti korištenjem centralnog sata, ali taj pristup dovodi do nepreciznosti u vremenskim oznakama uzrokovanih smetnjama u mreži. Vremensko umjeravanje je ponekad moguće izbjeći usklađenim okidanjem senzora. Međutim, takav pristup nije uvijek moguće implementirati, kao što i on nužno ne garantira nulti pomak satova. Stoga, najsigurniji pristup je odrediti odnose među satovima koristeći stvarna mjerenja senzora. Kako bi se sustav vremenski umjerio, potrebno je gibanje koje može potjecati iz gibanja senzorskog sustava ili gibanja objekta kojeg senzorski sustav promatra. Kroz disertaciju je razvijena metoda zasnovana na potonjem pristupu koji ima prednost da je primjenjiv i kod statičnih senzorskih sustava. Nadalje, kontinuirana reprezentacija je ključni aspekt predložene metode jer omogućuje jednoznačno uparivanje mjerenja asinkronih senzora s različitim frekvencijama. U disertaciji se koristi regresija Gaussovim procesima (GP), dok druge metode često koriste *B-spline* interpolaciju ili jednostavniju linearnu i sferičnu interpolaciju.

Online umjeravanje zasniva se na korištenju mjerenja iz radnog prostora robota, tj. bez korištenja mete za umjeravanje. Pri rješavanju tog problema, javljaju se dodatni problemi koje je potrebno razmotriti. Naime, okolina pruža veliku količinu podataka među kojima je potrebno pronaći najkorisnije u svrhu umjeravanja. Nadalje, umjeravanje robota je povremeno potrebno ponovno provesti jer razni utjecaji mogu narušiti kvalitetu umjeravanja te time naštetiti drugim zadaćama koje robot mora obavljati. Stoga je potrebno razviti metode koje mogu pravovremeno otkriti takve situacije uz mali utrošak računalnih resursa.

Disertacija je organizirana u sedam poglavlja. Prvo poglavlje disertacije daje uvod u temu, formalno opisuje problem te daje ilustrativni primjer kojim motivira potrebu za umjeravanjem. Drugo poglavlje daje pregled područja, dok treće poglavlje pružna osnovni uvod u korištene matematičke alate. Četvrto poglavlje opisuje glavne doprinose i rezultate disertacije. Peto poglavlje iznosi zaključke donesene kroz disertaciju te nudi pregled mogućeg budućeg rada. Šesto poglavlje daje popis objavljenih radova koji čine disertaciju, dok sedmo poglavlje opisuje doprinos autora na svakome od njih. Potom je izložen popis literature korištene u disertaciji te su priloženi radovi na kojima se disertacija zasniva. Disertacija je izrađena po skandinavskom modelu te ju čine četiri časopisna i tri konferencijska članka. Glavni doprinosi disertacije su izloženi i opisani u nastavku poglavlja.

#1 *Metoda ekstrinzičnog umjeravanja senzorskog sustava radar – kamera – laser u šest stupnjeva slobode poboljšana evaluacijom mjerenja radarskog presjeka.*

Ekstrinzično umjeravanje heterogenih senzora je zahtjevan zadatak jer takvi senzori mjere različite fizikalne pojave te daju raznovrsne podatke. Kako bi se taj izazov prebrodio, često se koriste mete za umjeravanje koje omogućuju precizne i efikasne metode. Oslanjajući se na mete, potraga za korespondencijama između senzora je pojednostavljena, dok apriorna znanja o meti mogu poboljšati preciznost estimacije. Nadalje, analiza identifikabilnosti osigurava dohvatljivost rješenja danog problema, a može dati preporuke za dizajn procedure prikupljanja podataka.

Prvi doprinos disertacije bavi se ekstrinzičnim umjeravanje sustava radar – lidar – kamera u 6 stupnjeva slobode (DOF). Metoda uključuje dizajn univerzalne mete prikladne za radar, lidar i kameru predstavljene u [Pub1]. Meta za umjeravanje se sastoji od stiropornog trokuta nevidljivog radaru s dobrim svojstvima detekcije i lokalizacije u oblaku točaka i slici. Radar prima refleksiju od trihedralnog kutnog reflektora koji ima visoki radarski presjek (RCS) i nisku osjetljivost na orijentaciju. Nova dvo-koračna optimizacija koja omogućuje potpuno i precizno umjeravanje u svih 6 DOF je predstavljena u [Pub1]. Prvi korak optimizacije zasnovan je na minimizaciji reprojekcijske pogreške, dok drugi korak, optimizacija RCS-a, koristi prostornu distribuciju RCS-a kako bi estimirao varijable koje nisu identifikabilne kriterijem reprojekcijske pogreške zbog nedostatka radarove vertikalne rezolucije. Podskup ekstrinzičnih parametara koje drugi korak optimizacije popravlja uključuje translaciju u vertikalnom smjeru te kuteve valjanja i poniranja. Posebno odabrana parametrizacija ekstrinzičnog umjeravanja konzistentno omogućuje najveću razdvojenost nesigurnosti između parametara čime potvrđuje odluku o zaključavanju parametara u dvo-koračnoj optimizaciji.

Prva inačica RCS optimizacije [Pub1] zasnovana je na predodređenom nominalnom vidnom polju (FOV) radara i pragu RCS-a. Optimizacijskim kriterijem se pokušava obuhvatiti sva mjerenja s visokim RCS-om unutar nominalnog FOV-a. Druga inačica RCS optimizacije [Pub2] uvodi novi kriterij koji vodi do preciznijih rezultata, pri čemu izostavlja potrebu za podešavanjem početnih parametara nominalnog FOV-a i praga RCS-a. Optimizacijom se estimiraju parametri krivulje elevacija – RCS koji najbolje objašnjavaju mjerenja. Iako sporedni parametri krivulje nemaju praktičnu vrijednost, oni vode do poboljšanog ekstrinzičnog umjeravanja povezujući elevaciju mjerenu lidarom s RCS-om mjerenim radarom. Nadalje, [Pub2] proširuje metodu iz [Pub1] uključujući kameru u proces umjeravanja. Uz to, detaljna analiza identifikabilnosti optimizacije reprojekcijske pogreške je provedena u [Pub2]. Ona objašnjava hipotezu o nejednakoj raspodjeli nesigurnosti za različite konfiguracije senzora, potvrđuje potrebu za drugim korakom optimizacije te pruža naputke za proces prikupljanja podataka koji omogućuje pouzdano umjeravanje. Metoda je testirana na simuliranim i stvarnim podatcima koristeći dva različita radara. Rezultati su pokazali da je moguće konzistentno i precizno estimirati svih 6 DOF-e ekstrinzičnog umjeravanja. Na kraju, metoda je iskorištena kako bi se provela procjena radarova vertikalnog pozicioniranja uz pomoć ravnine poda estimirane lidarom.

#2 *Metoda za ekstrinzično i vremensko umjeravanje heterogenih eksterocepcijskih senzorskih sustava mobilnih robota zasnovana na praćenju objekata pomoću regresije Gaussovim procesima.*

Vremensko umjeravanje zahtjeva gibanje, bilo vlastito gibanje senzorskog sustava ili gibanje objekta kojeg sustav promatra. Iako su metode zasnovane na vlastitom gibanju pouzdan izvor informacija za umjeravanje, svi senzori ne mogu dovoljno dobro estimirati vlastito gibanje, npr. radari. Uz to, statični senzorski sustavi su lišeni vlastitog gibanja. Stoga se drugi doprinos disertacije zasniva na korištenju gibajućih meta za ekstrinzično i vremensko umjeravanje heterogenih senzora. Jedini preduvjet je da svi senzori mogu estimirati 3D poziciju objekta što je moguće koristeći mnogo raznih senzora, npr. kamerama, lidarima, radarima, sustavima za praćenje gibanja (MOCAP), itd.

Srž metode za umjeravanje predstavljene u [Pub6] čini regresija GP-ima. Trajektorije mete su opisane koristeći regresiju GP-ima kako bi se dobile izglađenje reprezentacije u vremenski kontinuiranoj domeni koje omogućuju preciznu vremensku registraciju korespondencija i umjeravanje. Apstrahirajući mjerenja senzora s kontinuiranim trajektorijama, registracija korespondencija između asinkronih senzora s različitim vremenima uzorkovanja postaje jednostavna. Nadalje, preklapanje trajektorija kojime se ostvaruje umjeravanje, odvija se kroz optimizaciju na mnogostrukosti. Metoda optimizacije zahtjeva izglađene trajektorije dobivene GP-ima, ali omogućuje precizno i računski učinkovito umjeravanje. Uz to, metoda omogućuje estimaciju vremenskog pomaka, ali i razlike u frekvencijama satova.

Predložena metoda iscrpno je testirana u simulacijama te stvarnim eksperimentima koji uključuju četiri različita senzora: kameru, lidar, radar te MOCAP sustav. Pokazano je kako metoda može estimirati vremenski pomak s greškom znatno manjoj od najkraćeg vremena uzorkovanja, npr. greška od 0.8 ms za kamere s vremenom uzorkovanja od 50 ms. Nadalje, precizno vremensko umjeravanje omogućava i jednostavno ekstrinzično umjeravanje, čak i pri visoko dinamičnom gibanju mete. Važnost vremenskog umjeravanja je prikazana kroz primjer kamera – MOCAP fuzije gdje je metoda uspjela smanjiti prosječnu reprojekcijsku pogrešku s 1.9 cm na 0.5 cm. Na kraju, javno dostupna implementacija koda za regresiju GP-ima i metodu umjeravanje je omogućila računski jednostavno i skalabilno rješenje. Naime, za minutu mjerenja frekvencije 20 Hz, metoda zahtjeva samo 49 ms za regresiju GP-om te 41 ms za optimizaciju.

#3 *Nenadzirana metoda ekstrinzičnog i vremenskog umjeravanja heterogenih eksterocepcijskih senzorskih sustava mobilnih robota tijekom rada zasnovana na grafovima.*

Cjeloživotna operacija robotskog sustava je iznimno ovisna o pouzdanoj umjerenosti sustava koja može degradirati s vremenom. Kako bi se nadišao taj izazov, metode za online umjeravanje koriste informacije iz okoline kao korespondencije među senzorima. Otkrivanje i praćenje gibajućih objekata poput vozila i pješaka se često obavlja pomoću svih senzora na robotskoj platformi, što pruža veliku količinu informacija za umjeravanje. Treći doprinos disertacije predstavljen u [Pub5] proširuje umjeravanje zasnovano na gibajućoj meti otpuštanjem preduvjeta za poznatom metom te dodajući nekoliko značajki koje omogućuju efikasno online otkrivanje pogreške umjeravanje te ponovno umjeravanje.

Metoda se sastoji od standardnog pristupa otkrivanju i praćenju gibajućih objekata koristeći radare, kamere i lidare. Uvedena je nova tehnika za uparivanje praćenih traka koja je otporna na pogreške umjeravanja. Uz to, metoda koristi jednostavnu metodu za umjeravanje parova senzora čime je omogućena računarski pristupačna metoda za otkrivanje pogreške umjeravanja te inicijalizaciju ponovnog umjeravanja. Na kraju, metoda koristi globalno umjeravanje svih senzora zasnovano na grafovima kako bi omogućila konzistentno ponovno umjeravanje cijelog sustava.

Predložena metoda je testirana na javno dostupnom skupu podataka namijenjenom razvoju autonomnih vozila koji sadrži radar, kameru i lidar. Od uobičajenih sudionika u prometu, metoda koristi samo podatke o okolnim vozilima jer samo njih svi senzori mogu pouzdano pratiti. Rezultati su pokazali da je metoda sposobna otkriti male pogreške u umjeravanju rotacije unutar nekoliko sekundi, kao i ponovno umjeriti cijeli sustav u normalnom režimu rada. Nadalje, pokazano je kako ovaj pristup radi bolje od uobičajenih pristupa zasnovanih na vlastitom gibanju prilikom neinformativnih segmenata vožnje.

KLJUČNE RIJEČI: umjeravanje senzora, ekstrinzično umjeravanje, vremensko umjeravanje, praćenje gibajućih objekata, meta za umjeravanje, radar, lidar, kamera, identifikabilnost, Fisherova informacijska matrica, Gaussovi procesi, Lieve groupe, optimizacija na mnogostrukosti

CONTENTS

1

Introduction

Robust environment perception is one of the essential tasks which an autonomous mobile robot or vehicle has to accomplish. To achieve this goal, various sensors such as cameras, radars, lidars, and inertial navigation units (IMU) are used and information thereof is often fused. Essential tasks such as simultaneous localization and mapping (SLAM), detection and tracking of moving objects (DATMO), and odometry are often improved by sensor fusion. A fundamental step in the fusion process is sensor calibration, commonly divided into extrinsic, intrinsic and temporal. In this chapter, a formal problem statement and motivation are given in Section 1.1, followed by a list of original contributions in Section 1.2 and outline of the thesis in Section 1.3.

1.1 Problem statement and motivation

1.1.1 *Problem statement*

Sensor calibration aims to find necessary parameters for (i) interpretation of data from a single sensor and (ii) fusing data between multiple sensors. Former set of parameters are found with intrinsic calibration, while the later ones with extrinsic and temporal calibration. Intrinsic calibration provides internal parameters of each sensor, e.g. focal length of a camera or bias in lidar range measurements. Extrinsic calibration, also known as spatial calibration, provides relative transformation between coordinate frames of two sensors. Lastly, temporal calibration, also known as synchronization, of the sensors aligns the clocks of different sensors which includes a constant temporal offset, i.e. time delay, as well as drift between the clocks caused by different rates. A calibration method can estimate all parameter groups at the same time or a procedure can be devised that decouples the parameter groups and estimates them individually.

Intrinsic parameters are related to the working principle of the sensor. Therefore, methods for finding intrinsic parameters do not share many similarities for different types of sensors. On the other hand, parametrization of extrinsic calibration, i.e. homogeneous transform, can always be expressed in the same manner, regardless of the sensors involved. Despite that, solving the extrinsic calibration is challenging because it requires finding correspondences between the data acquired by sensors that can measure different physical quantities. Lastly, temporal calibration can be performed using external hardware systems, e.g. flashing diodes for cameras, or by using motion cues, either the system itself or the target that the system observes. The former approach is often restricted to specific sensor

configuration, while the later can be limited by feasible motion. Nevertheless, extrinsic and temporal calibration require correspondence registration between two or more sensors which can stem from designated targets, environment or observed motion.

After correspondence registration, calibration parameters are estimated using an appropriate method. Calibration methods typically try to satisfy some geometric constraints through minimization of a problem-specific reprojection error. The geometric constraints involve nonlinearities which often cannot be solved analytically. To resolve this challenge, estimators use iterative linearization techniques to find the appropriate solution. A common approach is to perform iterative non-linear batch optimization, e.g. using Levenberg–Marquardt algorithm. This approach often leads to the most accurate results that are more robust to modelled errors, while requiring longer computation time. Due to the nonconvexity of the problem caused by the nonlinearities, these methods have a risk of converging to a local minimum. To avoid this risk, some methods divide optimization in initial rough estimates that guarantee near-optimal solutions followed by nonlinear iterative refinement step. Alternatively, calibration can be solved within a filtering framework where real-time performance and continuous calibration are required. However, these approaches often have stronger requirements on initial calibration parameters and can diverge due to uninformative data.

Calibration approaches can be target-based or targetless, sometimes referred to offline and online approaches, respectively. In the case of target-based calibration, correspondences originate from a specially designed target, while targetless methods utilize environment features perceived by the sensors. Former has the advantage of the freedom of design that maximizes the chance of all involved sensors perceiving the calibration target, but requires the development of such a target and execution of an appropriate offline calibration procedure. The latter has the advantage of using the environment itself as the calibration target and can operate online by registering structural correspondences in the environment, but requires all involved sensors to be able to extract the same environment features. Registration of structural correspondences can be avoided by motion-based methods, which use the system's motion estimated by the individual sensors to calibrate them. These methods have two main advantages: (i) they rely less on the sensors' operating principles and can be applied to different sensors, assuming that a sensor can estimate its motion, (ii) unlike other methods, they are able to extrinsically calibrate sensors with non-overlapping fields of view. Finally, calibration can be performed using trajectories of moving objects around the sensor system. While these methods require overlapping sensor field of view (FOV), they abstract the sensor measurements by using object trajectories and work even with static sensor systems.

Regardless of the chosen approach, the success of the calibration is highly dependent on the provided data. An important step before the data acquisition is determination of minimal requirements on the dataset for which the problem becomes identifiable (or observable in the case of dynamic systems). The identifiability question is often addressed through (i) the geometric viewpoint, (ii) framework of nonlinear observability or (iii) statistical tools such as Fisher Information Matrix. This analysis provides guidelines on data collection which would yield reliable results. Furthermore, it can provide a measure of calibration estimate uncertainty. Namely, datasets used for sensor calibration can often times result

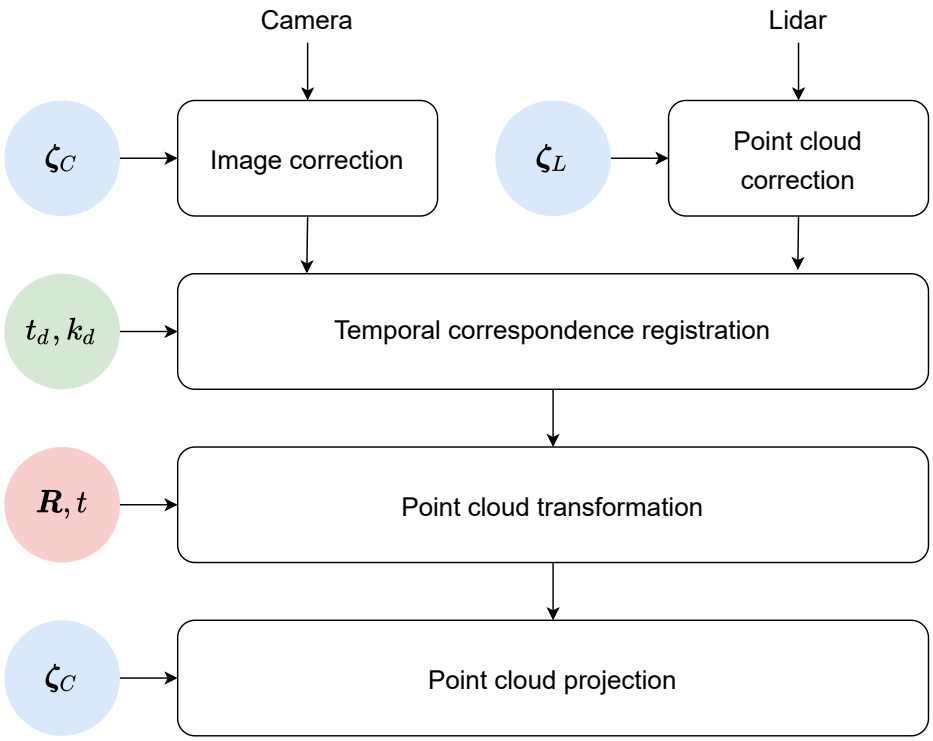


Figure 1.1: Illustration of lidar – camera fusion pipeline emphasizing influence of intrinsic, temporal and extrinsic calibration parameters.

in uneven certainty among parameters. By levering on that insight, we can often direct the data collection process or use the estimated parameters with more caution. Besides determining minimal requirements on the dataset, calibration method can benefit from wisely choosing a subset of measurements that enter the estimation step. Using only recent data enables a system to detect miscalibration quickly, while it often sacrifices accuracy of the results. Furthermore, avoiding dataset subsets which lead to unobservability can improve the overall calibration accuracy.

Due to a myriad of approaches, significant differences among the involved sensors and all the aforementioned challenges, solving sensor calibration is not a straightforward task. An approach that could handle all the sensors, impose low requirements on the dataset and provide the most accurate results is still a holy grail of calibration. Thus, it is still common practice to choose an approach that best accommodates the sensors at hand. In this thesis, the main goal is to enable proper calibration between radars and their most common companions, cameras and lidars. Namely, radars have become ubiquitous in modern robotic systems, e.g. autonomous vehicles (AVs), while the methods for their calibration remained limited. To overcome this issue, several methods for extrinsic and temporal calibration between radars, cameras and lidars have been developed and are presented in the sequel.

1.1.2 *Motivation*

In this section, a canonical example of sensor fusion, the camera–lidar case, is dissected to illustrate the crucial role of the sensor calibration. These sensors are often used in robotics for many tasks, while complementary nature of their measurements makes the fusion compelling. Cameras provide dense colored information about the environment which

enables extracting fine structural details. However, they lack the depth information which is readily available from lidars that usually provide sparse data. In addition, lidars measure intensity of the reflected ray which can be used to infer additional object properties, e.g. material of the reflecting surface. Thus, proper fusion of this data enhances various tasks which a robot has to perform such as mapping and localization or semantic segmentation. However, properly fusing the data requires accurate temporal, intrinsic and extrinsic calibration between the sensors. In a standard fusion pipeline, presented in Figure 1.1, the initial step is to correct individual sensor measurements using their intrinsic calibrations. It is followed by establishing a correct temporal correspondence between the measurements. The last two steps are used to project the lidar data into the camera image. By using extrinsic calibration, measured lidar points are transformed from lidar to camera coordinate frame. Lastly, intrinsic camera calibration is used to project these points into the image that yields final data correspondence. In the sequel, each step is further examined.

The first step in the pipeline corrects the sensor readings based on their intrinsic calibration parameters. Lidar intrinsics $\boldsymbol{\zeta}_L$ typically include range bias and intensity correction, while more complex models consider angular and position offsets for each beam [1]. Cameras usually require lens distortion removal, while photometric calibration can enhance the fusion further by removing undesired effects such as vignetting [2]. All intrinsic camera parameters, including projection parameters introduced in the sequel, are given with $\boldsymbol{\zeta}_C$.

Establishing temporal correspondence can be performed in several ways: (i) through hardware synchronization using external triggers, (ii) routing all the data through a single computer that assigns the timestamps and (iii) using local timestamps assigned by each sensor clock. Hardware synchronization often results with the most accurate correspondence, but is not always possible to implement. Using a central computer is the simplest approach, but can degrade correspondence accuracy due to network jitter. On the other hand, local timestamps are not affected by the jitter, but suffer from time delay drift k_d due to different clock rates. Regardless of the approach, it is wise to use the actual sensor measurements in temporal calibration to estimate the time delay t_d between the sensors as well as clock drift in case of locally generated timestamps. Besides estimating temporal calibration parameters, specifics of sensor data acquisition need to be taken into account. For example, lidars often perform *sweeping* of the environment resulting in different timestamps for each azimuth angle. On the other hands, cameras can either have a global or rolling shutter requiring the knowledge of shutter and readout time. Lastly, knowing the temporal correspondence, the pipeline often has to perform motion distortion compensation and interpolation.

The last two steps enable final data correspondence by projecting lidar points in the camera image. Firstly, extrinsic parameters $(\boldsymbol{R}, \boldsymbol{t})$ are used to transform all lidar points ${}^l\boldsymbol{p}_l \in {}^lP_l$ from the lidar reference frame to the camera reference frame, ${}^c\boldsymbol{p}_l \in {}^cP_l$. Lastly, transformed points ${}^cP_l$ are projected on the image using an appropriate projection model $\pi(\boldsymbol{\zeta}_C, {}^c\boldsymbol{p}_l)$, given the camera intrinsic calibration parameters $\boldsymbol{\zeta}_C$. In case of a simple pinhole camera model, camera focal lengths and principal point are required at this step.

In this illustrative example, it is shown how sensor calibration estimates enter the system at multiple stages. Calibration inaccuracies can deteriorate any step of the pipeline. Therefore, it is necessary to provide the system with accurate sensor calibration to enable desired performance of other essential tasks that a robot has to solve.

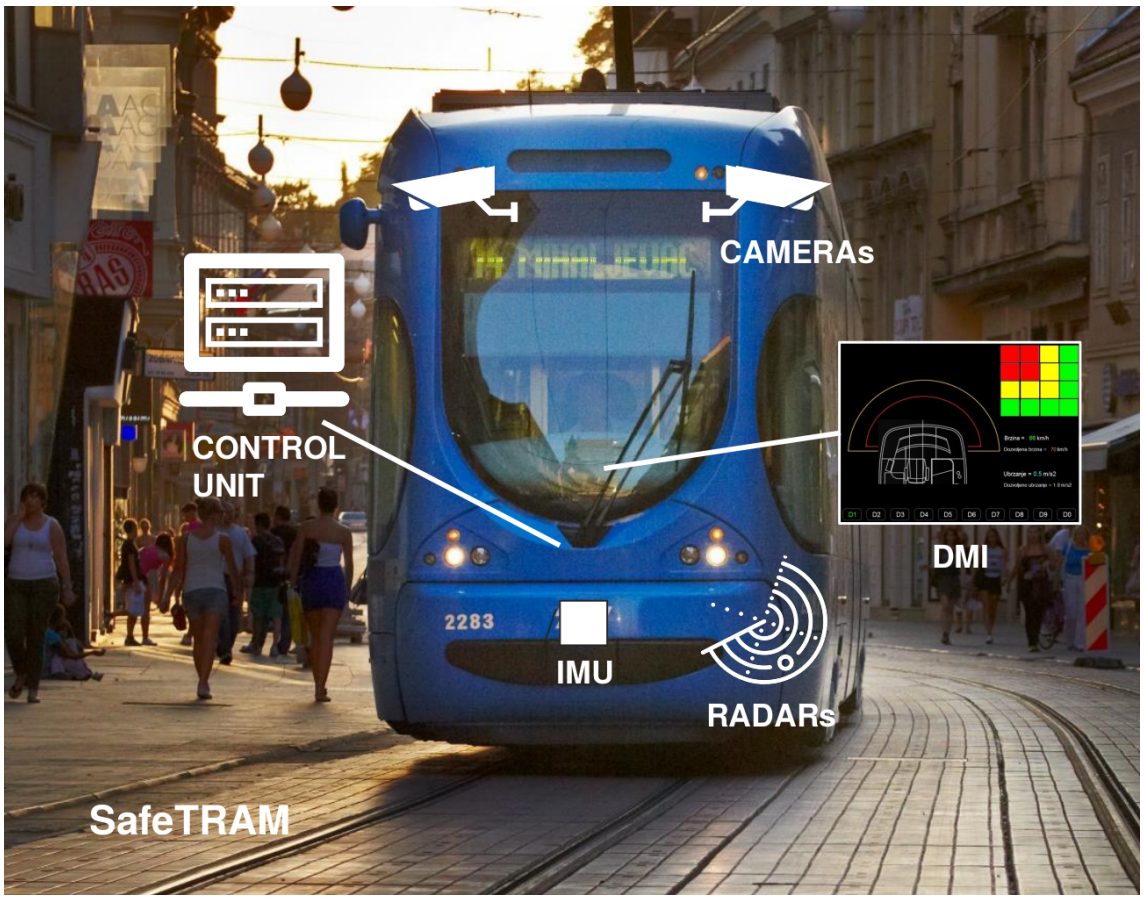


Figure 1.2: Illustration of system concept developed within the SafeTram project.

Furthermore, the research conducted within the thesis was motivated by the requirements of the SafeTram project. The proposed system illustrated in Fig. 1.2 consisted of multiple cameras and radars. Since radar calibration was not well studied, the conducted research explored the possibilities of calibrating radar – camera – lidar systems through various approaches.

1.2 Original contributions

The original contributions of the thesis enabled both extrinsic and temporal calibration of sensor systems involving radars. Due to common fusion of cameras and lidars with the radars, each developed method enables complete calibration of such systems. This thesis introduced three scientific contributions as a result of the conducted research:

- #1 A method for six degrees of freedom extrinsic calibration of radar – camera – lidar sensor system enhanced by radar cross section measurement evaluation.
- #2 A method for extrinsic and temporal calibration of heterogeneous exteroceptive mobile robot sensor systems based on object tracking using Gaussian process regression.
- #3 An online unsupervised graph-based method for extrinsic and temporal calibration of heterogeneous exteroceptive mobile robot sensor systems.

A more elaborate presentation of the aforementioned scientific contributions is given in Section 4.

1.3 OUTLINE OF THE THESIS

The thesis is divided into seven chapters. The main chapters discuss the current state of the art in sensor calibration and provide required theoretical background on mathematical frameworks used within the thesis. In addition, main contributions and results from the thesis are presented, along with the concluding remarks and directions for future research. In the sequel, a short summary of each remaining chapter is presented.

Ch 2 This chapter provides a broad overview of calibration approaches and sensor specifics that are relevant for this thesis. The main operating principles necessary for understanding of the contributions are briefly presented for each involved sensor: radar, camera and lidar. The state of the art in sensor calibration is classified into several categories: (i) target based calibration, (ii) targetless calibration, (iii) ego-motion based calibration and (iv) moving object based calibration. Finally, additional considerations that emerge in solving sensor calibration are addressed, including multi-sensor calibration, online calibration and the issue of observability.

Ch 3 This chapter provides a brief introduction into the mathematical frameworks used within this thesis. Application of the Fisher Information Matrix for the purpose of identifiability analysis is elaborated. Batch continuous-time estimation is addressed by introducing the Gaussian Process regression. Lastly, Lie Group theory is introduced as a prerequisite for on-manifold optimization used within the thesis.

Ch 4 This chapter describes the main scientific contributions of the thesis.

Ch 5 This chapter gives concluding remarks of the thesis and discusses some directions for the future work.

Ch 6 This chapter lists all the publications contributing to the main results of the thesis.

Ch 7 This chapter gives a statement on the author's contribution to each of the included publications.

The main part of the thesis is followed by a list of referenced bibliography. Afterwards, all the publications related to the main results of the thesis that were previously published in proceedings of international scientific conferences or peer-reviewed journals are attached.

2

Overview of calibration methods

Sensor calibration has to solve numerous challenging tasks, depending on the sensors involved, surrounding environment and various other circumstances. That has led to a variety of different approaches in solving it. In this chapter, Sec. 2.1 provides essential information about the involved sensors, Sec. 2.2 gives an overview of existing calibration approaches, while Sec. 2.3 discusses some additional considerations that arise in sensor calibration.

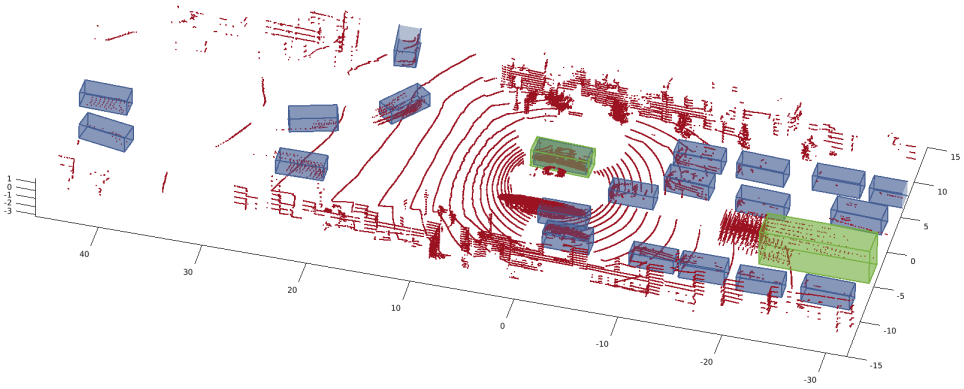
2.1 sensors

Understanding the operating principles of the sensors is essential in designing a suitable sensor calibration solution. Thus, this section provides essential information on radar, camera and lidar operating principles. To illustrate the challenges in heterogeneous sensor calibration, Fig. 2.1 shows how different sensors observe the same scene and detect objects on an autonomous vehicles (AV) dataset nuScenes [3], used within [Pub3].

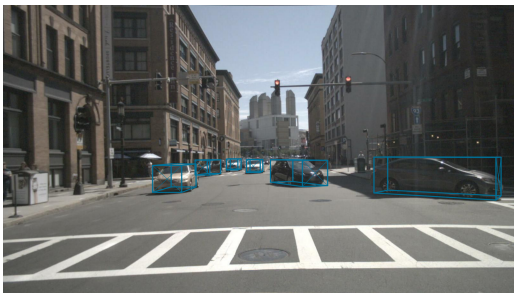
2.1.1 *Radar*

Radars are active sensors that emit electromagnetic waves to detect objects around them. By processing data from multiple emitting and receiving antennas, they produce a list of detected objects. Every object is described with azimuth angle, range, range-rate and reflectivity measure called radar cross section (RCS). The only structural information about an object is encoded in RCS, which depends on the object's shape, material, orientation, etc. Such radars are still the most common in robotics and AV applications, while the newer generations also measure elevation angle. In addition, novel mechanically scanning radars have been recently applied to AVs leading to increased resolution and informativeness at the cost of lower frame rate and bulkiness [4].

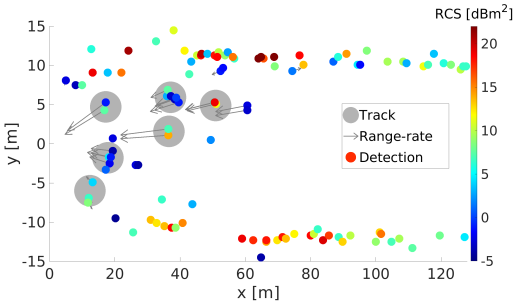
Figure 2.1c illustrates data that a radar outputs per single frame. It can be seen that radar produces a limited number of detections with a high outlier rate and low precision. Furthermore, it shows that RCS can help in initial filtering stage, but it is unreliable to infer on the environment structure. Angular and range accuracy are typically much lower than with the competing sensor modalities, while radar's strength lies in direct measurement of velocity. Furthermore, radars are more reliable at inclement weather, and they perform best when detecting metal objects. To overcome the aforementioned challenges and produce usable information in real-world scenarios, radar detections have to be filtered and tracked



(a) Lidar



(b) Camera



(c) Radar

Figure 2.1: Vehicle detection using lidar, camera and radar. Lidar pointcloud consists of 10 consecutive sweeps, while blue and green boxes represent car and truck detections, respectively. Radar detections are colored with radar cross-section and show range rate, while gray circles represent confirmed tracks.

over time. Thus, radars are most commonly used for detection and tracking of moving objects (DATMO), and there exist numerous tracking algorithms developed specifically for the them [5, 6]. In addition, radars have also been used to reliably estimate translational velocity of the ego-vehicle [7, 8].

2.1.2 *Lidar*

Lidars are active sensors that emit highly focused light pulses using lasers to measure distance to surrounding objects. Commonly, one or multiple lasers are attached to a mechanically spinning head, forming 2D or 3D lidars, respectively. Besides measuring distance, lidars also provide intensity measurements, while they recover angles using internal encoders and intrinsic calibration. To reduce production cost and increase reliability, solid state lidars have been recently developed. Despite producing a similar output as their mechanical counterparts, they require slight modifications in data interpretation and intrinsic calibration [9].

Perception algorithms usually process lidar data in a pointcloud form, illustrated in Fig. 2.1a. It can be seen that lidars often produce sparse data with significantly different horizontal and vertical angular resolution. Thus, extraction of fine details in the structure of the environment is often limited, especially at longer distances. Furthermore, due to

continuous sweeping of the mechanical head, each point is obtained at a different time instant. It complicates temporal calibration and registration with other sensors, while pointclouds get distorted due to the sensor system motion while taking the scan [10].

2.1.3 *Camera*

Camera is a passive sensor that converts light waves into an image. Compared to the other sensors, cameras provide dense information about the environment that enables the extraction of fine structural details (c.f. Fig. 2.1b). However, unlike the aforementioned sensors, their performance highly depends on the environment lightning conditions and they cannot measure distance. In the context of robotics, it is essential to understand image formation process which is determined by two main building blocks: an imaging chip, i.e. imaging sensor, and a lens.

Imaging chip is responsible for converting light waves into electrical signal. Two most dominant technologies used are charge couple devices (CCD) and active-pixel sensor (CMOS), differing in how they convert light into signal. The performance of a camera highly depends on the used technology, imaging chip size and other design factors. It is often measured in terms of signal-to-noise ratio, dynamic range, quantum efficiency, etc. To obtain color images, several color separation techniques are used, where Bayes-filter is the most common. While it does provide colored images, it suffers from lower quantum efficiency and image artefacts stemming from demosaicing. Another important aspect of the camera performance is exposure time control. While analog cameras use a conventional mechanical shutter, digital cameras use an electrical shutter that can be global or rolling. Global shutter enables conversion of light into signal for all the pixels simultaneously, while the rolling shutter performs the conversion one row at a time. While rolling shutter cameras have several advantages in terms of performance, they suffer from motion distortion.

Before the light reaches the image sensor, it is directed towards it by the camera lens. Two main properties of a lens are aperture and focal length. While aperture determines the amount of light that reaches the image chip, focal length determines camera's a FOV. Based on the FOV, lenses can be divided into normal, wide-angle and long-focus. Due to their ability to bend light, lenses inevitably introduce distortion, i.e. deviation from rectilinear projection.

To reliably interpret camera data, it is essential to properly model all the aforementioned properties. Simple cameras with moderate FOV can be modelled as pinhole cameras, defined with focal lengths, principal point and skew parameter [11]. Moderate distortion is usually described using a radial-tangential distortion model, while more complex lenses (e.g. omnidirectional or fisheye lenses) require other appropriate models [12].

2.2 CALIBRATION APPROACHES

Sensor calibration is a wide research area and approaches in solving it vary significantly. Design of an appropriate calibration solution is highly driven by the sensors involved, requirements on the environment and performance. In this section, the most common calibration approaches are presented including (i) target based approaches in Sec. 2.2.1, (ii)

targetless approaches in Sec. 2.2.2, (iii) ego-motion approaches in Sec. 2.2.3 and (iv) moving object based approaches in Sec. 2.2.4. Furthermore, additional considerations in solving sensor calibration are addressed with Sec. 2.3.1 on multi-sensor calibration, Sec. 2.3.2 on online calibration and Sec. 2.3.3 on identifiability analysis.

2.2.1 *Target based calibration*

Calibration targets are frequently used due to numerous advantages. They simplify the correspondence registration step since the number and type of correspondences is known in advance which virtually eliminates the problems associated with outliers. Additionally, target-based methods can use a priori knowledge about the target which can enhance the calibration results. Therefore, target-based methods are generally more precise than the targetless. Finally, there are no requirements on the environment which can be uninformative and prevent targetless methods from success. However, these methods are the least practical since they require design and construction of the target and may not always be suitable (e.g. end-user applications like smartphones). Furthermore, they have to be performed offline before any other application for which the calibration is important. Therefore, it is impossible to make any runtime adjustments and the process has to be repeated in case of decalibration.

The properties of a well-designed target are (i) ease of detection and (ii) high localization accuracy for all the sensors in the calibration. The former ensures the success of the correspondence registration, while the latter has strong influence on the quality of the results given by the optimization step. Furthermore, if the a priori knowledge about the target is used, construction imprecision may degrade calibration results. Perception sensors used in robotics utilize a wide range of physical phenomena to extract information about the environment. Due to different types of data provided by heterogeneous sensors, there exist many diverse target designs. In the sequel we will present some of the designs grouped by the sensor types.

Camera

Cameras are passive sensors that utilize the light which goes through the lens and is detected at the imaging sensor. They are a rich source of information with an affordable price what makes them commonly used in robotics and other fields. Due to their long presence and frequent usage, intrinsic camera calibration has been given a lot of research attention which resulted with many camera description models and calibration techniques. While cameras with high distortion such as fisheye and omnidirectional cameras require more complex models [13], commonly used cameras with slight distortion are usually modelled as a pinhole cameras with a previously undistorted image [11]. This intrinsic parametrization consists of distortion coefficients (e.g. radial-tangential distortion) and camera matrix formed by focal lengths, principal point and skew parameter between the axes.

Commonly used camera calibration targets are planar checkerboard patterns. They are suitable because they can be easily detected in the image and enable sub-pixel resolution using interpolation based on a known target dimensions. These calibration methods are based on pioneering work by Tsai [14] and Zhang [15]. Checkerboard pattern was also used

in calibration of rolling shutter camera parameters [16]. Besides checkerboard pattern, a grid of circles is also frequently used [17] with a comparison study of different patterns given in [18]. Moreover, a grid of triangles, named Deltille Grid [19], was introduced showing an increase in calibration accuracy of high-resolution cameras. Novel calibration target is presented in [20] where authors use a noise-like pattern with many features of varying scales. It is suitable for both intrinsic and extrinsic calibration of multiple cameras with no or little FOV overlap. The only requirement is that the neighbouring cameras observe parts of the target which may not overlap at all. Additionally, it can simultaneously handle both close-range and far-range cameras. To increase target detection rate and enable calibration of non-overlapping cameras, grids with encoded fields are often used. A grid of fiducial markers, AprilTags [21], is used in [22], while [23] extends the common checkerboard with various letters and signs.

Lidar

Lidars are active sensors that use light pulses to determine the range of the objects in the environment. Considering sensor calibration, 2D and 3D lidars have received extensive attention due to the application requirements and possibility to recover structure from the environment. Intrinsic parameters of interest are range measurement offsets and pose of the individual rays to the common reference frame. Unlike cameras, precision of intrinsic factory calibration parameters is usually considered sufficient. However, for the applications that require higher precision, authors in [24] propose a method for intrinsic calibration of rotating 3D lidar using a box with known dimensions, while the authors in [25] use a planar wall as a calibration target. Similarly, method presented in [26] uses a single plane to calibrate extrinsics of a push broom lidar, i.e. transform between rotating platform and a 2D lidar. In [9], the authors focused on spatial geometry of pointcloud formation and proposed a unified model for mechanical and solid-state lidars. For sensor configurations in which multiple 2D lidars share the same parts of FOV, Fernandez-Moral et al. [27] presented a solution which uses corner structures to perform extrinsic calibration. Additionally, using the rank of Fisher Information Matrix (FIM) they show that problem becomes identifiable when at least three perpendicular planes are observed. Contrary to aforementioned planar structures, a moving sphere is used as a target in [28] to calibrate multiple 2D and 3D lidars on a vehicle.

Lidar – Camera

Pointclouds from 2D/3D lidars are often fused with camera images. Both are rich information sources and precise extrinsic calibration is crucial for tasks such as 3D reconstruction what led to the development of numerous calibration methods. A common approach in target-based lidar – camera calibration is using planar targets which are easily detected and localized in the pointcloud covered by a fiducial pattern (e.g. checkerboard) which allows estimation of the plane position and orientation in the image.

Widely adopted and extended method presented by Zhang and Pless [29] introduced point-plane geometric constraint initially designed for 2D lidar – camera calibration. Lidar points originating from the target plane are transformed into the camera frame. After

that, the method tries to minimize point to plane distances based on the estimated plane parameters in the image. Pandey et al. [30] showed that the method is also applicable in case of 3D lidar – camera calibration. Zhou and Deng [31] improved the method by introducing additional constraints which decoupled rotation from translation. They achieved better results than other methods because their method is less affected by errors in the plane parameters estimation from the checkerboard image. Additionally, they showed that for a 2D lidar, at least five correspondences should be made with different target orientations, while a 3D lidar required minimum of 3 different views. Geiger et al. [32] tried to reduce the time of the calibration procedure by extending the method with global correspondence registration that allows for multiple plane observations in a single shot. The same constraint was used by Mirzaei et al. in [1] where instead of the checkerboard pattern, AprilTag fiducial markers were used [21]. Additionally, they extended the extrinsic calibration with estimation of intrinsic lidar parameters. AprilTag markers and the same geometric constraint were also used in [33] as a part of multi-sensor graph based calibration.

Besides commonly used point-plane constraint, 3D lidar-camera pairs were calibrated based on the point-point correspondences. Velas et al. [34] used a target with circular holes which allows a single-shot calibration and does not require observation of the plane in multiple orientations. Similar geometric constraints where used by Kwak et al. [35] for 2D lidar-camera calibration, where the improvements were made by extracting centreline and edge features of a V-shaped planar target. Furthermore, an interesting target adaptation to the working principle of different sensors was presented by Bormann et al. [36], where the authors proposed a method for extrinsic calibration of a 3D lidar and a thermal camera by expanding a planar checkerboard surface with a grid consisting of light bulbs. To extend the extrinsic calibration with the temporal one using a checkerboard as a target, Norwicki [37] proposed continuous representation of the plane equations.

Radar – Camera/Lidar

Radars are active sensors which, similarly to the lidar, emit an electromagnetic signal and determine the range of objects in the vicinity based on the returned echo. Although being frequently used in automotive applications due to their low price and robustness, extrinsic radar calibration has not gained as much research attention as lidar and camera calibrations. Majority of existing methods are target-based since, for all practical means and purposes, the targetless methods are hardly feasible due to limited resolution of current automotive radar systems, as the radar is virtually unable to infer the structure of the detected object and extract features such as lines or corners. As explained in Sec. 2.1.1, many automotive radars had no ability to measure elevation angle until the recent development. Although having no elevation resolution, radars have substantial elevation FOV which makes the extrinsic calibration challenging due to the uncertainty of the measurements.

Concerning automotive radars, common operating frequencies (24 GHz and 77 GHz) result with reliable detections of conductive objects, such as plates, cylinders and corner reflectors, which are then used in intrinsic and extrinsic calibration methods [38]. Wang et al. [39] used a metal panel as the target for radar – camera calibration. They assume that all radar measurements originate from a single ground plane, thereby neglecting the 3D

nature of the problem. The calibration is found by optimizing homography transformation between the ground and image plane. Contrary to [39], Sugimoto et al. [40] take into account the 3D nature of the problem. Therein, they manually search for detection intensity maximums by moving a corner reflector within the FOV. They assume that detections lie on the radar plane (zero elevation plane in the radar coordinate frame). Using these points a homography transformation is optimized between the radar and the camera. The drawback of this method is that the maximum intensity search is prone to errors, since the return intensity depends on a number of factors. For example, target orientation and radar antenna radiation pattern which is usually designed to be as constant as possible across the nominal FOV. Another solution that takes into account the 3D nature of the problem is presented in [41], where the authors form a novel criterion based on a priori known distances among multiple reflectors. Therein, the authors focus on radar – camera calibration and also tackle the issue of unknown scale of target detections from the images. The work presented in [Pub1, Pub2] was recently extended in [42] with iterative optimization scheme and temporal calibration based on target azimuth trajectory alignment. Lastly, a comparative study of multiple radar – camera calibration methods is available in [43], while [44, 45] provides a convenient tool for extrinsic calibration of radar – camera – lidar systems.

2.2.2 *Targetless calibration*

In order to maintain the reliability of a perception system, sensor calibration has to be performed occasionally. Sensors displacement due to mechanical vibrations, changes of intrinsic parameters due to the variable environment conditions such as temperature and pressure, are some of the effects that can cause sensor decalibration. In such cases target-based methods are impractical and can restrict usability of the system which led to development of the targetless methods. They eliminate the need for artificial targets by using environment features to match correspondences in the sensor data. This problem is especially challenging in the heterogeneous sensor systems. It is feasible when sensors provide enough information about the environment to extract its structure. Therefore, the techniques described in the sequel are mainly used in camera and lidar calibration.

Camera

Cameras are well suited for targetless calibration because they provide rich information about their environment. In targetless calibration of monocular cameras, primary concern are intrinsic parameters. Barazzetti et al. [46] proposed an approach for intrinsic camera calibration using only natural scenes. Their method uses feature extraction methods and robust estimation techniques to create correspondences between different views of the same scene. It is suitable for scenes with many features that can be uniquely described. However, repetitive textures (e.g. building facades, tiles) result in image features with similar descriptors which can be easily mismatched and thus compromise the calibration results. Although showing valuable results, authors conclude that high precision and industry applications still require target-based methods for desired calibration accuracy. Similar approach was adopted by Fraser and Stamatopoulos [47] where they showed comparable results to the target-based methods. The issue of online photometric camera calibration is

tackled in [2], where authors propose a method that works alongside SLAM systems in a realtime.

To recover camera poses efficiently without knowing the calibration, Sattler et al. [48] proposed a focal length sampling scheme to improve solver speed. Deep learning has been applied in calibration of pan-tilt-zoom cameras to recover their variable intrinsics and extrinsics [49]. In another deep learning approach, Cramariuc et al. [50] focused solely on detecting camera miscalibration. Neural networks have also been used to detect informative features in the environment, which are used as an input to calibration. Kocur et al. relied on vehicle vanishing point detector to calibrate traffic cameras [51], while Han et al. [52] used detections of stop sign to calibrate cameras on a vehicle. In [53], the authors detect horizon line and vertical vanishing point to obtain bird's eye view from a single image, while they simultaneously estimate camera focal length.

In order to retrieve depth information about the environment, two cameras are often rigidly connected to form a stereo vision system. Besides the intrinsic calibration of individual cameras, high precision of extrinsic calibration between the cameras is crucial for successful stereo reconstruction. Ling and Shen [54] have presented an approach which minimizes epipolar errors between the image pairs based on the sparse natural features to obtain 5 degrees of freedom (DOF) transformation between the cameras. They show comparable results to the target-based methods, with the unobservable scale of translation vector. To achieve continuous calibration, Hansen et al. [55] rely on linear Kalman filter, while Dang et al. [56] use iterated extended Kalman filter. Rehder et al. [57] propose a solution to estimate both intrinsic and extrinsic calibration from scratch by restructuring bundle adjustment into an incremental process.

In multi-camera systems with little or no FOV overlap, SLAM based solutions are often used [58, 59, 60]. Within their CamOdoCal framework, Heng et al. [61, 62] proposed a solution that estimates both intrinsics and extrinsics, while they recover scale ambiguity using calibrated odometry system. Lin et al. [63] divide the problem in two stages, where they first estimate camera extrinsics up to a scale, followed by intrinsic and extrinsic refinement. Keivan and Sibley [64], tackle the issue of change detection, delayed observability, informative segments and constant-time computation enabling online SLAM-based calibration. OpenVINS framework [65, 66] formulates the problem through visual-inertial multi-state constraint Kalman Filter [67] to obtain temporal, extrinsic and intrinsic calibration of multiple-camera multiple-IMU systems.

Lidar

Despite having lower data density than cameras, lidars can also extract environment features suitable for calibration. In [27], the authors used perpendicular planes, e.g. building corners, to enable extrinsic calibration of multiple 2D lidars in any geometric configuration. Maddern et al. [68] calibrate multiple 2D and 3D lidars mounted on a moving vehicle by optimizing the quality of generated 3D pointcloud. Levinson and Thrun [69] perform intrinsic calibration of a single 3D lidar (orientation and remittance response for each beam) based on the assumption that points originate from contiguous surfaces. In configurations with small or none FOV overlap between multiple lidars, extrinsic calibration is commonly performed by

relying on motion. Liu and Zhang [70] formulate their problem through graph optimization to enable co-visible features of non-overlapping lidars with small FOVs. Heng [71] proposed a solution to calibrate multiple 3D lidars with radars on a moving vehicle, introducing the first feature-based targetless radar calibration method. Firstly, 3D map and lidar extrinsics with respect to the vehicle are estimated, followed by the radar calibration that relies on the estimated map.

Camera – lidar

Informativeness of these sensors enables inference on the structure of the environment that can be used in generating correspondences. For example, Levinson and Thrun [72] based their calibration of a 3D lidar and a camera on line features detected as intensity edges in the image and depth discontinuities in the pointcloud. Their method is able to detect decalibration on-the-fly and track the gradual drift of the sensor pose over time. Similar approach was adopted by Moghdam et al. [73] where they increased the robustness of their method by handling one-to-many correspondence registration by re-weighting the error metric. Gong et al. [74] proposed an approach in which they use arbitrary trihedrons commonly found in urban and indoor environments (e.g. corners of the buildings).

In addition to range measurements, lidars also provide information about returned signal's intensity. Pandey et al. [75] find extrinsic calibration by maximizing the mutual information between the cameras grayscale pixel intensities and projected surface reflectivity values measured by the lidar. For success of their method it is important to first perform intrinsic inter-beam calibration of the surface reflectivity values. The concept of mutual information was also used by Taylor and Nieto [76]. Instead of using dense information from the pointcloud, they only project selected features into the 2D lidar image. Additionally, they complement returned intensity information with estimated surface normal as there exist strong statistical dependence between these quantities. They show that the method is applicable to variety of lidars. Furthermore, mutual information between camera image and lidar generated reflectance image was used by Napier et al. [77] to calibrate a push-broom 2D lidar with camera in natural scenes. The method allows calibration of sensors without overlapping FOVs, but it requires ego-motion information. Park et al. [78] proposed a SLAM-based solution that enables both extrinsic and temporal calibration of camera and 3D lidar by aligning 3D features. To circumvent hand-engineering features, Yuan et al. [79] proposed a deep learning approach with proper consideration of the underneath geometry.

Generation of 2D image from the lidars pointcloud was also done by Scaramuzza et al. [80]. Instead of intensity, they introduced bearing angle images which are constructed from angle difference of the surface normals in the pointcloud. This metric highlights environment plane intersections arising from wall corners and other similar discontinuities. However, their method requires manual registration of the correspondences. Lastly, extracting features from the environment can lead to a high number of correspondences. Scott et al. [81] claimed that not all correspondences are equally informative and that appropriate choice of scenes can improve calibration. They use normalized information distance as a criteria for scene selection scheme which provided more effective and precise calibration results using fewer scenes.

2.2.3 *Ego-motion based calibration*

Motion-based calibration techniques compare ego-motion estimates from individual sensors to perform calibration. These methods can be classified as targetless methods because they also use environment features indirectly. However, due to feature abstraction using ego-motion, developed methods are applicable to a wider range of sensor configurations. The only requirement is that the sensor can estimate its motion. Moreover, for the proprioceptive sensors such as IMU or encoder odometry, ego-motion based methods are the only viable solution. Additionally, they are often the only option for calibration of sensors whose FOVs do not overlap. Many of these methods are agnostic in terms of sensor choice. In the sequel, some of the general methods will be addressed, followed by methods with focus on proprioceptive sensors.

Hand – Eye calibration

Ego-motion based estimation is often referred to as hand – eye calibration due to its origins. Namely, many robotic applications involve a manipulator equipped with a wrist-mounted sensor such as a camera [82]. Calibration between the end of the manipulator and the perceptive sensor is crucial in these applications. This problem is often referred to as an $AX = XB$ problem due to the emerging equation that needs to be solved (A and B represent manipulator and sensor movement, respectively, while X represents the extrinsic calibration). It has been studied for more than three decades [83] and many solutions exist.

Some of the recent advances in the field have dealt with the problem of unknown correspondences caused by asynchronous sensors or missing detections [84, 85, 86]. In their work [87], Schneider et al. have proposed a solution for extrinsic calibration of sensors based on Unscented Kalman Filter. The method is generic and can be used with sensors which provide both 3 DOF and 6 DOF motion estimates. However, they require time-synchronized delta poses. Furthermore, a general solution for motion-based extrinsic and temporal calibration was given by Taylor and Nieto [88]. The solution is based upon the framework of $AX = XA$ problem which is further enhanced by targetless methods if the sensor types and overlaps allow such refinements. It was evaluated through calibration of several vehicle-mounted sensor configurations. Brookshire and Teller [89] proposed an approach in which they explicitly model the noise via the Lie algebra yielding a constrained FIM from which they analyze motion degeneracy and proceed to singularity-free optimization procedure. Furthermore, Huang and Stachniss [90] have addressed the problem of high measurement noise which compromises the results of the commonly used least square optimization techniques. They improved the calibration results by adopting Gauss-Helmert optimization paradigm which jointly optimizes calibration parameters and pose observation errors. Della Corte et al. [91] proposed a framework for unified motion-based multi-sensor calibration with time delay estimation. The problem of certifiable globally optimal solution was tackled in [92] using a dual quaternion based approach, while Giamou et al. [93] formulate calibration as a quadratically constrained quadratic program. Wise et al. [94] adapted the approach to handle sensors with unknown scale such as cameras. Calibration of cameras with no overlapping FOV on a vehicle is often solved using hand-eye approaches [95] with explicit handling of the unknown scale [96]. Due to the issue of poorly observable parameters

caused by planar motion, several methods include ground plane constraints to the optimization [97, 98]. To improve results of the hand-eye calibration, Schmidt and Niemann [99] proposed a solution for data selection based on vector quantization.

Proprioceptive sensors

Proprioceptive sensors such as IMUs and wheel encoders do not observe environment features and can only rely on ego-motion in calibration. However, IMUs and encoders are often fused with other sensors due to their reliability and convenience.

Visual-inertial navigation is able to accurately estimate 6 DOF motion and it is well suited for many robotic tasks. However, it requires precise extrinsic calibration which led to numerous calibration methods. Mirzaei and Roumeliotis [100] proposed a Kalman filter based approach for IMU – camera calibration. They based their method on estimating the camera motion using checkerboard pattern. Through the observability analysis based on the Lie derivatives rank criterion they showed that it is necessary to excite at least two rotational axis of the system to make the calibration parameters observable. Kelly and Sukhatme [101] have continued on the previous research by discarding the checkerboard and using environment features for the visual odometry. They showed that additional two translational axes need to be excited in order to resolve camera scale issue and make the calibration parameters observable. Keivan and Sibley [64] proposed a SLAM solution which is able to detect system decalibration and perform calibration online. Furgale et al. [102, 103] proposed a method which relaxes the synchronization constraint by using continuous-time batch estimation while simultaneously estimating both spatial and temporal calibration parameters. Their approach was further developed by Sommer et al. [104] where efficient B-Splines derivative computation was proposed.

Similarly to visual-inertial systems, IMU is often fused with lidar for enhanced performance. Kim et al. [105] proposed a calibration method that combines feature matching, motion and rigid body constraints for multiple lidars and IMU on a vehicle. Le Gentil et al. [106] proposed a framework for lidar-inertial SLAM and autocalibration for a hand-held device. In [107], Lv et al. used B-Splines as continuous-time representation in lidar – IMU calibration to tackle to problem of asynchronous measurements.

Wheel encoders are often used in robotics for local trajectory estimation as they are often readily available on vehicles and require simple integration. However, they need to be calibrated with respect to other sensors on the platform for proper fusion. Kellner et al. [108] proposed a radar to odometer calibration approach based on estimated vehicle velocity using radar range-rate Doppler measurements. Guo et al. [109] presented a two-step analytical solution to extrinsic odometer – camera calibration. On the other hand, Kümmerle et al. [110] solved the problem of extrinsic calibration within the SLAM framework, while also enabling estimation of kinematic robot parameters. In [91], a solution for hand-eye calibration was extended to enable both extrinsic and kinematic calibration of a robot with respect to other sensors. Deray et al. [111] formalized the problem of encoder preintegration, similarly to IMU preintegration [112], to enable accurate kinematic robot calibration.

2.2.4 *Moving object based calibration*

To avoid operating directly on features and designing a sensor-specific measures, trajectories of moving objects can be used to abstract sensor readings. Unlike ego-motion based methods, these methods usually require FOV overlap. However, they enable calibration of static sensor systems, often referred to as sensor networks. The concept of aligning trajectories of moving calibration targets has been applied in several methods for homogeneous sensors. A solution to calibration of depth camera network was proposed by Su et al. [113] where they used a spherical calibration object to obtain extrinsic calibration. Fornaser et al. [114] adopted a similar approach, while also estimating time delays among the sensors. On the other hand, Faion et al. [115] relied on a cube as calibration target, while also enabling extrinsic calibration of sensors without FOV overlap by using a known global position of the target.

In solutions without calibration target, human motion is often used to obtain extrinsic calibration [116, 117, 118]. Glas et al. [119, 120] adopted this approach in 2D lidar calibration where they proposed a novel matching strategy. Specifically, to match trajectories between the sensors, the authors observe a similarity measure of the net velocity history profiles; however, in the optimization step, they rely only on the detected positions of the tracked people. They have further extended their approach to calibrate 2D lidars with depth cameras [121]. Rowekamper et al. [122] propose an extrinsic calibration method for a network of 2D lidars by formulating a graph-based optimization problem. In [123], authors also rely on pose graphs in 2D lidar calibration, wherein rotation is decoupled from translation by using a rotation averaging approach. Schöller et al. [124] proposed a method for stationary camera – radar calibration in traffic surveillance environment. Huber et al. [125] focused on temporal calibration using moving target tracking, while calibrating cameras, infrared tracker and coordinate measurement machine.

2.3 ADDITIONAL CONSIDERATIONS

Sensor calibration is a complex problem that can be solved in numerous ways, as previously presented. Regardless of the approach, several important questions often arise in the process. In this sections, a brief overview of most common additional considerations is given. Namely, Sec. 2.3.1 presents the problem of calibrating more than two sensors, Sec. 2.3.2 provides details on important considerations in online calibration, and Sec. 2.3.3 addresses the issue of calibration identifiability.

2.3.1 *Multi-sensor calibration*

Modern robotic systems often employ a large number of sensors. A system can be equipped with multiple non-overlapping cameras to provide a 360° surround view or might fuse forward facing radar, lidar and camera to obtain a rich environment interpretation. Nevertheless, all these sensors have to be calibrated. Classic approach is a pairwise calibration which can handle two sensors at a time. However, it raises questions about solution consistency, while the choice of sensor pairs that enter the pairwise calibration is ambiguous. To overcome these challenges, multi-sensor calibration methods explicitly calibrate systems with more than two sensors.

Inspired by graph-based SLAM approaches [126, 127], multi-sensor calibration can be formed as a graph optimization problem where nodes represent sensors poses, while the edges encode measurement correspondences between them [114, 128, 129, 33, 130]. Le and Ng [128] tackled the problem of multi-sensor calibration by (i) grouping sensors to produce 3D data, (ii) using a variety of geometric constraints and (iii) sharing sensors between groups for increased robustness. Thus, they constraint the problem in multiple ways, i.e. their nodes in a graph have many edges. Wagner et al. [129] proposed a framework for graph-based multi-sensor calibration, by respecting the geometry of the problem through formulating a manifold-based optimization scheme. Owens et al. [33] rely on a planar target in their graph-based calibration approach, while requiring FOV overlap between only two sensors to produce a global solution. Kühner and Kümmerle [130] handle uncertainties of each sensor through several models to handle different types of sensors. To incorporate their approach with graph-based optimization, they extend the graph nodes with positions of the target, yielding a more complex, but accurate solution.

A global solution to multi-sensor calibration can be obtained by constraining the optimization with a single trajectory of a sensor system. Rehder et al. [103] formulated the motion of sensor system via continuous-time B-spline representation, while constraining lidar, camera and IMU with their individual cost functions to produce temporal and extrinsic calibration among them. Della Corte et al. [91] formulate the problem of multi-sensor hand-eye calibration by choosing one sensor, e.g. wheel odometry, as a reference. While they consider odometry constraints only between the reference and any other sensor, they enable additional constraints between other perceptive sensors such as Iterative Closest Point (ICP).

2.3.2 *Online calibration*

Online calibration is a vague term that has been used to describe various features of sensor calibration approaches. In a general robotics context, the term *online* is often a synonym for *real-time*. While some methods indeed enable continuous, real-time calibration, it is not the only trait that makes a calibration approach online. The term is often used to emphasize that artificial calibration objects are not needed. Furthermore, online calibration has to solve some additional challenges that do not appear in offline, target-based calibration.

One of the challenges in online calibration is proper data selection. Namely, when a calibration system uses data from the environment, the amount of data can grow quickly. Furthermore, adding uninformative data to the optimization might even degrade the overall calibration accuracy. Della Corte et al. [91] propose a metric that combines Hessian determinant and ratio of eigenvalues, to evaluate both the complete amount of data and uneven uncertainty among the parameters. Keivan and Sibley [64] propose a priority queue of trajectory segments encoding calibration mean and covariance. After choosing a fixed number of most informative segments, they perform calibration using all the segments in the queue. Scott et al. [81] propose a diligent scene selection scheme for online lidar – camera calibration based on normalized information distance between lidar point reflectance and image intensity. Using the proposed approach, they are able to produce more accurate estimates using only the selected scenes. Schmidt and Niemann [99] propose a vector quantization

approach for data selection in hand – eye calibration. Schneider et al. [131] rely on various tests of Fisher Information Matrix to determine the informativeness of individual trajectory segments in visual-inertial calibration. Maye et al. [132] propose a general calibration approach using batch optimization that relies on evaluation of information gain. By choosing only the new measurements that increase informativeness, they reduce the total number of correspondences and thus create framework for feasible online calibration.

Another important aspect in online calibration is decalibration detection, because in real-world systems it is as crucial as calibration itself. A parameter might drift slowly due to external influences such as temperature or change abruptly due to sudden shocks. In scenarios where calibration is costly, a simple decalibration detection schemes are added to enable quick response of the system. In [64], authors detect a possible change in a new trajectory segment by comparing distributions of newly estimated parameters with the old ones using the Multivariate Behrens-Fisher hypothesis testing. Della Corte et al [91] follow the same approach with extension to kinematic parameters. Deray et al. [111] propose a window based approach with adaptive length based on a trend of overall cost. Thus, they are able to use more data in normal operating condition to achieve higher accuracy, while also enable sudden miscalibration to shorten the window. To detect miscalibration in lidar – camera calibration, Levinson and Thrun [72] evaluate the percentage of current calibration perturbations that increase the overall cost. They show that their method can even work on a single frame, while adding more frames significantly lowers false positives. Cramariuc et al. [50] proposed a camera miscalibration detection approach using a deep convolutional neural network.

To enable online calibration, several methods extend common navigation frameworks. Visual-inertial odometry is often based on variants of nonlinear Kalman filters in which extrinsic camera – IMU calibration can be seamlessly integrated [133, 65, 64, 134, 135, 136]. Several methods extend that approach to estimate temporal [133, 65, 134] and intrinsic camera calibration [65, 135] as well. On the other hand, Kümmerle et al. [110] extend a graph-based SLAM solution to simultaneously estimate sensor extrinsics and robot kinematic parameters.

2.3.3 *Identifiability*

Sensor calibration aims to find a set of usually stationary parameters from available data. However, parameter estimation can fail partially, or even fully, due to uninformative data. Thus, it is essential to perform identifiability, i.e. observability, analysis that can tell which parameters are recoverable from the available data. Term *observability* stems from control theory, where it refers to ability to estimate non-stationary states of a system with current measurements. On the other hand, *identifiability* is a more precise term as it refers to estimation of stationary parameters. However, we use both terms interchangeably as it is a common practice in sensor calibration literature.

In their seminal work, Hermann and Krener [137] proposed a method for analyzing controllability and observability of nonlinear systems that has been used by numerous calibration methods. Their differential geometric approach based on Lie derivatives creates an observability matrix. It is based on system dynamics and measurement models upon

which observability can be proven by examining rank of the matrix. Several methods have used this approach in camera – IMU filtering based calibration [135]. Mirzaei and Roumeliotis [100] show that camera – IMU extrinsics are observable using a calibration target. Kelly and Sukhatme [101] prove the observability of camera – IMU extrinsics, IMU biases, gravity vector, scene structure and IMU pose for both target-based and targetless calibration. Tsao and Jan [135] use a similar approach to additionally prove observability of camera intrinsics in visual-inertial odometry, while Li and Mourikis [138] examine observability of time offset estimation in visual-inertial systems. Martinelli et al. [139] proved the observability of kinematic robot parameters and extrinsics between robot odometry system and a camera. Wu et al. [140] analyze observability of IMU – odometer calibration, while Censi et al. [141] examined the observability of 2D extrinsics and kinematic parameters on a differential drive robot.

Another approach to tackle identifiability is through examining geometric constraints of the problem. It often leads to an intuitive solution, but it is not always easy to formulate the problem in a suitable way. Minimal requirements on pointcloud registration, i.e. *Orthogonal Procrustes problem*, have been studied by several authors through examining the closed-form solutions of the problem [142, 143]. Mirzaei et al. [1] tackled the problem of extrinsics identifiability between 3D lidar and camera using a point – plane geometric constraint. They examined the influence of perturbing selected extrinsics directions on the geometric constraints for observation of one, two and three linearly independent planes.

FIM is a tool often used in identifiability analysis due to its wide applicability and relation to Cramér–Rao bound, a lower bound on the variance of unbiased estimators. Maye et al. [132] used FIM to detect unobservable directions in parameter space from the available data. They enhanced their online calibration approach by locking the calibration estimation of selected unobservable parameters during problematic trajectory segments. Schneider et al. [131] used FIM to detect most informative trajectory segments for self-calibration of visual and inertial sensors. Brookshire and Teller [144] showed the minimal condition for calibration of 2D sensors based on egomotion, while they further extended it do a 3D case in their later work [89]. Use of FIM goes beyond calibration observability analysis. Wang and Dissanayake [145] used it to verify observability of several 2D SLAM solutions. Furthermore, FIM has been used in several papers for active calibration, i.e. finding and executing robot trajectory that improves calibration accuracy [146, 147, 148]. Alternatively, Preiss et al. [149] optimize trajectory using a novel criterion related to the local observability Gramian following the differential geometric observability approach.

3

Theoretical background

Calibration methods introduced throughout this thesis leverage several mathematical frameworks. In this chapter, a brief theoretical background on the used tools is given. Section 3.1 introduces the concepts of identifiability analysis using FIM, Sec. 3.2 elaborates the main concepts of Gaussian Process (GP) regression, while Sec. 3.3 lays out fundamentals of Lie Group theory essential for on-manifold optimization.

3.1 FISHER INFORMATION MATRIX

Identifiability analysis is an important aspect of estimation methods which aims to answer whether the parameters of interest can be properly estimated with the available data and the chosen criterion. Furthermore, it can be used to determine minimal requirements on the dataset and provide guidelines on experiment design which ensures proper calibration. As presented in Sec. 2.3.3, the main approaches applied in robotics can be divided into (i) geometric approaches based on qualitative analysis of the constraints, (ii) differential geometric approach based on analysis of observability matrix and (iii) statistical approaches. In this section, a short introduction into the most common statistical approach based on FIM is given. Although the other approaches enable analytical solutions, they are often impractical due to heavy nonlinearities associated with the problem at hand. On the other hand, FIM provides a principled approach for determining local identifiability of the system that can be applied in a wide range of problems. Informally, Fisher Information measures the amount of information that an observable random variable X carries about an unknown parameter θ of a distribution that models X. When θ is N-dimensional vector, it takes $N \times N$ matrix form, i.e. FIM. Formally, it is the covariance of the score, which is a gradient of log-likelihood function, further elaborated in the sequel.

Let us start by formally defining the estimation problem as a nonlinear regression

$$\mathbf{Y} = \mathbf{H}(\theta, \mathbf{X}) + \varepsilon, \tag{3.1}$$

with a response variable $\mathbf{Y} \in \mathbf{R}^{M \cdot N \times 1}$ and a predictor variable $\mathbf{X} \in \mathbb{R}^{S \cdot N \times 1}$, where N, M and S denote number of measurements, measurement variable size and predictor variable size, respectively. Parameters of size D, e.g. extrinsic calibration, are given with vector $\theta \in \mathbb{R}^D$, while $\varepsilon \sim \mathcal{N}(0, \mathbf{Q}) \in \mathbb{R}^{2N \times 1}$ is additive zero-mean white noise. Finally, $\mathbf{H}(\cdot)$ represents nonlinear transformation, e.g. calibration criterion. For simplicity, measurements of one sensor are often treated as the predictor variable, thus being modelled as noise-free. This

may lead to slight imprecision in the simulation of the error. However, we are not necessarily concerned with precise estimation of the error and covariance, but with the impact of the proposed nonlinear transformation on the identifiability of the problem.

Identifiability can be a global or a local concept for a specific $\theta_{\mathbf{0}}$ [150]. For linear regression, local identifiability coincides with global identifiability, while for the nonlinear regression, this claim does not hold. Since FIM cannot provide insights into global identifiability, we restrict our analysis to local identifiability only. However, it is sufficient in most situations if we can assume a known rough initial estimate of the parameters. Let us proceed with formally defining the local identifiability.

Definition 3.1. *Local identifiability*

The noise-free system is locally identifiable at $\theta_{\mathbf{0}}$ *if*

$$\exists U_{\theta_{\mathbf{0}}} \subset \mathbb{R}^d \text{ (open subset containing } \theta_{\mathbf{0}})$$

$$\forall \theta \in U_{\theta_{\mathbf{0}}}, \{\theta \neq \theta_{\mathbf{0}}\} \Rightarrow \{\mathbf{H}(\theta, \mathbf{X}) \neq \mathbf{H}(\theta_{\mathbf{0}}, \mathbf{X})\}.$$

To interpret it intuitively, the nonlinear function must not provide the same output for different parameter sets. In other words, a change in the response variable must be observed given the change in the parameter values. Two other important theoretical concepts are the *likelihood function* and the *score*.

Definition 3.2. *Likelihood function*

The likelihood function $\mathcal{L}_\theta$ *is defined as*

$$\mathcal{L}_\theta = \mathcal{L}(\theta; \mathbf{Y}, \mathbf{X}) = p(\mathbf{Y}; \theta, \mathbf{X}),$$

where $p(\mathbf{Y}; \theta, \mathbf{X})$ is probability density function of the random variable $\mathbf{Y}$ given the parameters θ and predictor variable $\mathbf{X}$. On the other hand, $\mathcal{L}$ is a function of parameters θ given the actual observed outcome of random variable $\mathbf{Y}$ and predictor variable $\mathbf{X}$.

Definition 3.3. *Score function*

The score function $\dot{\mathcal{L}}_\theta$ *is the gradient of the log-likelihood function* $\mathcal{L}(\theta; \mathbf{Y}, \mathbf{X})$ *at* θ

$$\dot{\mathcal{L}}_\theta = \nabla_\theta \log \mathcal{L}(\theta; \mathbf{Y}, \mathbf{X}) = \frac{1}{\mathcal{L}(\theta; \mathbf{Y}, \mathbf{X})} \frac{\partial \mathcal{L}(\theta; \mathbf{Y}, \mathbf{X})}{\partial \theta}.$$

Informally, score function indicates how sensitive a likelihood function $\mathcal{L}_\theta$ is to the change in its parameters θ. Intuitively, this would mean that with higher sensitivity, it should be easier to estimate the parameter. Under certain differentiability conditions, expected

value of the score is 0:

$$\begin{aligned} \mathrm{E}\left[\frac{\partial}{\partial\theta}\log p(\mathbf{Y};\theta,\mathbf{X}) \mid \theta\right] &= \int \frac{\frac{\partial}{\partial\theta}p(\mathbf{Y};\theta,\mathbf{X})}{p(\mathbf{Y};\theta,\mathbf{X})} p(\mathbf{Y};\theta,\mathbf{X})dy \\ &= \frac{\partial}{\partial\theta}\int p(\mathbf{Y};\theta,\mathbf{X})dy \\ &= \frac{\partial}{\partial\theta}1 = 0. \end{aligned} \tag{3.2}$$

An interesting notion that is used in defining FIM as the covariance matrix of the score. Since the expected value of the score is zero, FIM is a positive semi-definite matrix of size $D \times D$ whose elements can be computed as

$$\left[\mathcal{I}(\theta)\right]_{i,j} = \mathrm{E}_\theta\left[\left(\frac{\partial}{\partial\theta_i}\log\mathcal{L}(\theta;\mathbf{Y},\mathbf{X})\right)\left(\frac{\partial}{\partial\theta_j}\log\mathcal{L}(\theta;\mathbf{Y},\mathbf{X})\right)\right]. \tag{3.3}$$

If the FIM is twice differentiable with respect to θ and under certain regularity conditions [151], it can be rewritten as negative expectation of the log-likelihood Hessian.

$$\left[\mathcal{I}(\theta)\right]_{i,j} = -\mathrm{E}_\theta\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\mathcal{L}(\theta;\mathbf{Y},\mathbf{X})\right] \tag{3.4}$$

Informally, FIM tells how much information about the parameters is available in any direction of the parameter space from observing the sample. Furthermore, due to log-likelihood, FIM is additive for independent and identically distributed random variable (i.i.d.) samples. Thus for K i.i.d. observations, FIM is simply:

$$\mathcal{I}_K(\theta) = K\mathcal{I}(\theta) \tag{3.5}$$

It is thus often computationally easier to sum multiple samples. Furthermore, omitting expectation operator leads to the sample based version of FIM, often referred to as observed FIM. While it is often easier to compute observed FIM, when $N \to \inf$, observed FIM converges to the expected FIM [152]. Its assessment was conducted by Efron and Hinkley [153] where they argued that the observed FIM should be preferable choice when normal approximation for the distributions of maximum-likelihood estimates are employed. Observed Fisher information is defined as

$$I(\hat{\theta})_{i,j} = -\sum_{k=1}^{K}\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\mathcal{L}(\theta;\mathbf{Y}_k,\mathbf{X}_k)\right]\Bigg|_{\theta=\hat{\theta}} \tag{3.6}$$

where $\hat{\theta}$ is the final estimate of the parameters, while $\mathbf{Y}_k$ and $\mathbf{X}_k$ represent individual i.i.d. samples of previously stacked vectors $\mathbf{Y}_k$ and $\mathbf{X}_k$, respectively.

Since we defined our problem as a nonlinear regression with additive white noise, our likelihood function is simply a well-known probability density function of a multivariate normal distribution. For such cases, it can be shown that calculation of the FIM elements simplifies to [150]

$$\left[\mathcal{I}(\theta)\right]_{i,j} = \frac{\partial\mathbf{H}(\theta,\mathbf{X})}{\partial\theta_i}\mathbf{Q}^{-1}\frac{\partial\mathbf{H}(\theta,\mathbf{X})^{\mathrm{T}}}{\partial\theta_j}. \tag{3.7}$$

After the FIM is constructed, it is analyzed to assess identifiability of the problem. The sufficient condition for identifiability is that FIM is of full rank [153, 154]. Due to the numerical imprecision and noise, it is often necessary to use numeric rank of a matrix. Initial step in obtaining numeric rank is to perform Singular value decomposition composition of a matrix [155]. Numerical rank is then determined as a number of singular values greater than a "small" value ε which is often found empirically. Alternatively, the matrix conditional number can be observed. It is the ratio of the biggest and the smallest singular value, where high values indicate degeneracy of the matrix.

Besides identifiability test, FIM is used for many other purposes. The Cramér–Rao bound states that the inverse of the Fisher information matrix is a lower bound on the covariance of any unbiased estimator of θ [151]. Thus, it can be used to determine confidence intervals for the parameter estimates [156]. Furthermore, it is often used in optimal experiment design where several summary statistics of FIM are of particular interest: (i) D-optimality - maximization of FIM determinant; (ii) T-optimality - maximization of FIM trace and (iii) A-optimality: minimization of FIM inverse trace [157].

3.2 BATCH CONTINUOUS-TIME ESTIMATION

Estimation techniques can be formulated in continuous or discrete time manner. While processes we try to model usually occur in continuous-time domain, discretizing them can lead to computationally simpler models. Furthermore, implementing an estimation algorithm on a physical computer requires discretization. However, discretized form has certain limitations due to the made assumptions, e.g. fixed frame rate. Nevertheless, certain estimation problems can benefit greatly if addressed in continuous time.

One of the main contributions of this thesis is the use of continuous-time trajectory representation for sensor calibration. While measurement processes of some sensors can be well approximated with discrete-time interpretation (e.g. camera, radar), others are inherently continuous-time, e.g. a sweeping lidar. Furthermore, sensor calibration has to handle asynchronous sensors that operate at different frequencies. Obtaining a mathematically correct temporal correspondence between them is virtually impossible without continuous-time representations. In the robotics community, there are several common ways of representing trajectories in continuous-time domain. The simplest approach is to interpolate discrete values of two consecutive states. For example, position at any time instant can be obtained using linear interpolation, while rotation relies on spherical linear interpolation (SLERP) [158]. This approach was used in fusion of visual odometry with inertial navigation system [159] and calibration-free removal of rolling shutter effect [160]. SLERP was also used by several ego-motion based calibration methods [91, 161]. Nowicki [37] used a similar approach and applied it to plane parameters interpolation for the camera – lidar calibration. While this approach offers simple solution, it does not provide smoothness that is especially needed in temporal calibration. To overcome this issue, several approaches have adopted B-splines as a tool for continuous-time trajectory representation [162, 163]. They were applied in rolling shutter camera calibration [16, 164, 165], as well as lidar – camera – IMU spatiotemporal calibration [103]. Processing of event based cameras in visual-inertial odometry also relied on splines [166]. They were also used in asynchronous multi-camera SLAM [167], online

lidar SLAM [168] and structure from motion using rolling shutter cameras [169]. Rigid body motion interpolation is often required in animation as well, while a comprehensive survey of several spline techniques can be found in [170].

This thesis focuses on continuous-time representation using GPs, which enable a theoretically grounded batch state estimation and interpolation. They have been a well recognized tool in machine learning [171] both for regression and classification problems. Recently, they entered the robotics community following their efficient implementation for state estimation [172, 173, 174, 175], with proposed use for a variety of robotics tasks [175]. Several methods relied on GPs to enable efficient motion planning [176, 177, 178, 179, 180, 181], and some of them used the implementation presented in [Pub4]. In [173, 174] they were used for mobile robot localization, while in [182] they were used for tracking of extended objects. Yan et al. [183] proposed an extension to batch state GP based estimation that enables incremental updates, reduces computational costs and thus enables efficient online continuous-time SLAM.

The most common alternative to GP regression are aforementioned B-splines, often used for their computational efficiency. However, recent development of the GP regression [174] enabled comparable computational efficiency, while GPs provide several advantages. They are configured using a standard state estimation framework, i.e. by choosing a physical motion model and tuning process and measurement noise. Furthermore, GPs include covariance estimation which opens possibilities to use uncertainty of the interpolated states. In the sequel, a brief overview of the used GP regression will be given, while a more detailed derivation can be found in [174, 175].

3.2.1 *Gaussian Process Regression*

Gaussian Process are stochastic processes, in our case indexed by time, for which any finite collection has a multivariate normal distribution. They are distributions over functions defined by mean and covariance, while marginalizing to a particular single time of interest yields a Gaussian random variable. In this thesis, systems with a continuous-time GP model priors and discrete-time measurements are considered. While it is required to interpolate trajectory states at any time of interest, sensor measurements are available only at discrete-time instances. Let the continuous-time GP model prior be defined with

$$\boldsymbol{x}(t) \sim \mathcal{GP}(\check{\boldsymbol{x}}(t), \check{\boldsymbol{P}}(t, t')), \tag{3.8}$$

and a discrete-time, linear measurement model:

$$\boldsymbol{y}_k(t) = \boldsymbol{C}_k \boldsymbol{x}_k(t_k) + \boldsymbol{n}_k, \tag{3.9}$$

where $\boldsymbol{x}(t)$ is the state, $\check{\boldsymbol{x}}(t)$ is the mean function, $\check{\boldsymbol{P}}(t, t')$ is the covariance function, $\boldsymbol{y}_k$ are the measurements, $\boldsymbol{n}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ is Gaussian measurement noise, and $\boldsymbol{C}_k$ is the measurement model matrix. For now, we assume that the state is queried at the measurement times ($t_0 < t_1 < \ldots < t_N$), while there is no limitation in estimating states at arbitrary times of interest, i.e. interpolation times. However, there is no significant improvement in estimating states at interpolation times compared to estimating them at measurement times and performing querying at interpolation times as described later in (3.27) and (3.28). Interested

reader is referred to [175] (cf. Sec. 3.4.1) for more details. Joint distribution of both the state and the measurement is given with

$$p\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \check{\mathbf{x}} \\ \mathbf{C}\check{\mathbf{x}} \end{bmatrix}, \begin{bmatrix} \check{\mathbf{P}} & \check{\mathbf{P}}\mathbf{C}^T \\ \mathbf{C}\check{\mathbf{P}} & \mathbf{R} + \mathbf{C}\check{\mathbf{P}}\mathbf{C}^T \end{bmatrix}\right) \tag{3.10}$$

where $\check{\boldsymbol{P}}$, $\boldsymbol{C}$, and $\boldsymbol{R}$ are batch matrices defined as

$$\check{\boldsymbol{P}} = [\check{\boldsymbol{P}}(t_i, t_j)]_{ij}, \tag{3.11}$$

$$\boldsymbol{C} = \operatorname{diag}(\boldsymbol{C}_0, \ldots, \boldsymbol{C}_N), \tag{3.12}$$

$$\boldsymbol{R} = \operatorname{diag}(\boldsymbol{R}_0, \ldots, \boldsymbol{R}_N), \tag{3.13}$$

while stacked vectors of states $\boldsymbol{x}$, state priors $\check{\boldsymbol{x}}$ and actual sensor measurements $\boldsymbol{y}$ at measurement times are given with

$$\boldsymbol{x} = [\boldsymbol{x}(t_0), \ldots, \boldsymbol{x}(t_N)]^T, \tag{3.14}$$

$$\check{\boldsymbol{x}} = [\check{\boldsymbol{x}}(t_0), \ldots, \check{\boldsymbol{x}}(t_N)]^T, \tag{3.15}$$

$$\boldsymbol{y} = [\boldsymbol{y}_0, \ldots, \boldsymbol{y}_N]^T, \tag{3.16}$$

with $N+1$ being the number of measurements. After the factoring of Eq. (3.10) (cf. [175] Sec. 2.2.3), the Gaussian posterior of the states given the measurements evaluates to

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\Big(\underbrace{(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C})^{-1}(\check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{y})}_{\hat{\boldsymbol{x}}\text{, posterior mean}}, \underbrace{(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C})^{-1}}_{\hat{\boldsymbol{P}}\text{, posterior covariance}} \Big). \tag{3.17}$$

After rearranging the posterior mean expression, a linear system for $\hat{\boldsymbol{x}}$ is obtained

$$(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C})\hat{\boldsymbol{x}} = (\check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{y}), \tag{3.18}$$

In general, time complexity for solving (3.18), as currently presented, is $\mathcal{O}(N^3)$ [174]. To improve the computational efficiency, a special class of GP priors is introduced, whose sparsely structured matrices can be exploited.

3.2.2 *Exactly Sparse GP Priors*

The special class of GP priors is based on the following linear time-varying stochastic differential equation (LTV-SDE)

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{v}(t) + \boldsymbol{L}(t)\boldsymbol{w}(t), \tag{3.19}$$

where $\boldsymbol{F}$ and $\boldsymbol{L}$ are system matrices, $\boldsymbol{v}$ is a known control input, and $\boldsymbol{w}(t)$ is generated by a white noise process. The white noise process is itself a GP with zero mean value

$$\boldsymbol{w}(t) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{Q}_c\delta(t - t')), \tag{3.20}$$

where $\boldsymbol{Q}_c$ is a power spectral density matrix.

The mean and the covariance of the GP are generated from the solution of the LTV-SDE given in (3.19)

$$\check{\boldsymbol{x}}(t) = \boldsymbol{\Phi}(t, t_0)\check{\boldsymbol{x}}_0 + \int_{t_0}^{t} \boldsymbol{\Phi}(t, s)\boldsymbol{\nu}(s)\mathrm{d}s, \tag{3.21}$$

$$\check{\boldsymbol{P}}(t, t') = \boldsymbol{\Phi}(t, t_0)\check{\boldsymbol{P}}_0\boldsymbol{\Phi}(t', t_0)^T + \int_{t_0}^{\min(t,t')} \boldsymbol{\Phi}(t, s)\boldsymbol{L}(s)\boldsymbol{Q}_c\boldsymbol{L}(s)^T\boldsymbol{\Phi}(t', s)^T\mathrm{d}s, \tag{3.22}$$

where $\check{\boldsymbol{x}}_0$ and $\check{\boldsymbol{P}}_0$ are the initial mean and covariance of the first state, and $\boldsymbol{\Phi}(t, s)$ is the state transition matrix [173].

Due to the Markov property of the LTV-SDE in (3.19), the inverse kernel matrix $\check{\boldsymbol{P}}^{-1}$ of the prior, which is required for solving the linear system in (3.18), is exactly sparse block tridiagonal [173]:

$$\check{\boldsymbol{P}}^{-1} = \boldsymbol{F}^{-T}\boldsymbol{Q}^{-1}\boldsymbol{F}^{-1}, \tag{3.23}$$

where

$$\boldsymbol{F}^{-1} = \begin{bmatrix} \mathbf{1} & 0 & \dots & 0 & 0 \\ -\boldsymbol{\Phi}(t_1, t_0) & \mathbf{1} & \dots & 0 & 0 \\ 0 & -\boldsymbol{\Phi}(t_2, t_1) & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{1} & 0 \\ 0 & 0 & \dots & -\boldsymbol{\Phi}(t_N, t_{N-1}) & \mathbf{1} \end{bmatrix} \tag{3.24}$$

and

$$\boldsymbol{Q}^{-1} = \mathrm{diag}(\check{\boldsymbol{P}}_0^{-1}, \boldsymbol{Q}_{0,1}^{-1}, \dots, \boldsymbol{Q}_{N-1,N}^{-1}) \tag{3.25}$$

with

$$\boldsymbol{Q}_{a,b} = \int_{t_a}^{t_b} \boldsymbol{\Phi}(t_b, s)\boldsymbol{L}(s)\boldsymbol{Q}_c\boldsymbol{L}(s)^T\boldsymbol{\Phi}(t_b, s)^T\mathrm{d}s. \tag{3.26}$$

This kernel allows for computationally efficient, structure-exploiting inference with $\mathcal{O}(N)$ complexity. This is the main advantage of the proposed exactly sparse GP priors based on a LTV-SDE in (3.19).

As we previously stated, the key benefit of using GPs for the continuous-time trajectory representation is the possibility to query the state $\hat{\boldsymbol{x}}(\tau)$ at any time of interest τ, and not only at measurement times. For multisensor calibration, this proves to be extremely useful, since many sensors operate at different frequencies; thus, the GP approach enables us to temporally align the states. If the prior proposed in (3.21) is used, GP interpolation can be performed efficiently due to the aforementioned Markovian property of the LTV-SDE in (3.19). State $\hat{\boldsymbol{x}}(\tau)$ at $\tau \in [t_i, t_{i+1}]$ is a function of only its neighbouring states [174],

$$\hat{\boldsymbol{x}}(\tau) = \check{\boldsymbol{x}}(\tau) + \boldsymbol{\Lambda}(\tau)(\hat{\boldsymbol{x}}_i - \check{\boldsymbol{x}}_i) + \boldsymbol{\Psi}(\tau)(\hat{\boldsymbol{x}}_{i+1} - \check{\boldsymbol{x}}_{i+1}), \tag{3.27}$$

$$\boldsymbol{\Lambda}(\tau) = \boldsymbol{\Phi}(\tau, t_i) - \boldsymbol{\Psi}(\tau)\boldsymbol{\Phi}(t_{i+1}, t_i), \tag{3.28}$$

$$\boldsymbol{\Psi}(\tau) = \boldsymbol{Q}_{i,\tau}\boldsymbol{\Phi}(t_{i+1}, \tau)^T\boldsymbol{Q}_{i,i+1}^{-1}, \tag{3.29}$$

where $\boldsymbol{Q}_{a,b}$ is given in (3.26). The fact that any state $\check{\boldsymbol{x}}(\tau)$ can be computed in $O(1)$ complexity can be exploited for efficient matching of trajectory states between multiple asynchronous sensors.

3.3 LIE GROUPS

Vast majority of problems in robotics handle rigid-body motion which is nonlinear in a general case. However, mathematical machinery used in robotics is often designed for linear problems, while handling ubiquitous nonlinearities requires linearisation or some other techniques. Considering rigid-body motion, roboticists struggle most with handling rotation. There are numerous ways to represent it: rotation matrices, Euler angles, quaternions, angle-axis, etc. Some of them are minimal representations that can suffer from singularities, while others avoid them by overparametrization which in return requires normalization. Neither case is ideal for mathematically consistent solutions as they require explicit handling of special cases or introduce errors into the pipeline. Recent decade has brought a rapid introduction of Lie group theory into robotics for handling rigid-body motion. Lie group theory is a broad area of research in mathematics, dating back to 19th century, while it is still an active field today. It has been widely used in various areas of physics and mathematics, while roboticists rely on a small subset of the theory. However, it was particularly useful in describing 3D pose uncertainty [184, 185], efficient on-manifold optimization [186], improving consistency of the filtering solutions [187, 188], measurement preintegration [112, 189, 111], etc. In this section, a short introduction into required concepts is given in Sec. 3.3.1, while an interested reader is highly encouraged to read an excellent introductory tutorial for roboticists by Sola et al. [190] or a more comprehensive introduction by Stillwell [191]. Section 3.3.2 describes a particular application of Lie groups used in this thesis, on-manifold optimization.

3.3.1 *Concepts*

To begin with, Lie groups embody the concepts of groups and smooth manifolds. Moreover, terms *Lie group* and *manifold* are often treated as synonyms and used interchangeably in robotics community. A Lie group $(\mathcal{G}, \circ)$ is a set $\mathcal{G}$ with group operation $\circ$ that satisfies closure, identity, inverse and associativity axioms:

$$\text{Closure} \ : \mathcal{X} \circ \mathcal{Y} \in \mathcal{G} \tag{3.30}$$

$$\text{Identity } \mathcal{E} : \mathcal{E} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{E} = \mathcal{X} \tag{3.31}$$

$$\text{Inverse } \mathcal{X}^{-1} : \mathcal{X}^{-1} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{X}^{-1} = \mathcal{E} \tag{3.32}$$

$$\text{Associativity } (\mathcal{X} \circ \mathcal{Y}) \circ \mathcal{Z} = \mathcal{X} \circ (\mathcal{Y} \circ \mathcal{Z}). \tag{3.33}$$

where $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}$. On the other hand, Lie group is a differentiable smooth manifold because its topological space locally resembles linear space. This property ensures existence of a unique tangent space at each point $\mathcal{X}$ on the manifold $\mathcal{M}$, often noted as $\mathcal{T}_{\mathcal{X}}\mathcal{M}$. Intuitively, if a point $\mathcal{X} \in \mathcal{M}$ is moving over the manifold, its velocity is expressed in the tangent space $\mathcal{T}_{\mathcal{X}}\mathcal{M}$. Furthermore, the manifold in a Lie group looks the same at every point, resulting with the same tangent spaces at any point. While a tangent space can be defined for any point of the manifold, a special one is defined at the identity element $\mathcal{E}$ called *Lie algebra* of the Lie group, noted as $\mathcal{T}_{\mathcal{E}}\mathcal{M} \triangleq \mathfrak{m}$.

While there exists numerous Lie Groups, roboticists are primarily working with Special Orthogonal groups $\mathrm{SO}(2)$ and $\mathrm{SO}(3)$ representing 2D and 3D rotation matrices, respectively,

Special Euclidean groups $SE(2)$ and $SE(3)$ extended with translation of the rigid-body motion and Similarity transform group $Sim(3)$ group further extended with scaling. In the sequel, some of the common properties of the Lie groups used within this thesis are given. In addition, $SO(3)$ group is chosen as an illustrative example to provide intuitive explanation of the theoretical concepts.

Lie groups are often used to transform elements of other sets, e.g. rotation of a vector for the $SO(3)$. Such transformations are called *group actions* where $\mathcal{X} \cdot \mathbf{v}$ represents action of manifold element $\mathcal{X} \in \mathcal{M}$ on another set's element $\mathbf{v} \in \mathcal{V}$. For example, $\mathbf{R} \cdot \mathbf{v}$ where $\mathbf{R} \in SO(3)$ is a rotation matrix and $\mathbf{v} \in \mathbb{R}^3$ is a vector. A group action has to satisfy the following axioms:

$$\text{Identity} : \mathcal{E} \cdot v = v \tag{3.34}$$

$$\text{Compatibility} : \quad (\mathcal{X} \circ \mathcal{Y}) \cdot v = \mathcal{X} \cdot (\mathcal{Y} \cdot v). \tag{3.35}$$

Besides group action on other sets, group composition defined in (3.30) can also be interpreted as an action of group on itself.

Relationship between a Lie group $\mathcal{M}$ and its corresponding Lie algebra $\mathfrak{m}$ is the most fundamental concept upon which all the other tools are built. Lie algebra has one particularly desirable property, it is a *vector space*. There exist two mutually inverse linear maps, i.e. isomorphisms, that convert an element from $\mathfrak{m}$ to $\mathbb{R}^m$ and vice versa, where m represents Lie algebra's degrees of freedom. These linear maps are commonly called *hat* and *vee*:

$$\text{Hat} : \mathbb{R}^m \mapsto \mathfrak{m}; \quad \tau \mapsto \tau^\wedge = \sum_{i=1}^{m} \tau_i E_i \tag{3.36}$$

$$\text{Vee:} \quad \mathfrak{m} \mapsto \mathbb{R}^m; \quad \tau^\wedge \mapsto \left(\tau^\wedge\right)^\vee = \tau = \sum_{i=1}^{m} \tau_i \mathbf{e}_i. \tag{3.37}$$

where $\mathbf{e}_i$ are base vectors of $\mathbb{R}^m$ and E_i their Lie algebra counterparts, i.e. $\mathbf{e}_i^\wedge = E_i$. Operations with vectors are much easier and they enable a myriad of tools used within robotics that rely on linear algebra. The structure of the Lie algebra can be found by taking the time derivative of the group constraint given in (3.32). In the case of $SO(3)$, the constraint is $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, which leads to the following structure of Lie algebra $\mathfrak{so}(3)$

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3.38}$$

with vector $\omega \in \mathbb{R}^3$.

The next key ingredient are transforms between $\mathcal{M}$ and $\mathfrak{m}$ defined with exponential and logarithmic map

$$\exp : \quad \mathfrak{m} \mapsto \mathcal{M} \quad ; \quad \tau^\wedge \mapsto \mathcal{X} = \exp\left(\tau^\wedge\right) \tag{3.39}$$

$$\log : \quad \mathcal{M} \mapsto \mathfrak{m} \quad ; \quad \mathcal{X} \mapsto \tau^\wedge = \log(\mathcal{X}). \tag{3.40}$$

For convenience, capitalized exponential and logarithm maps can be defined as

$$\text{Exp} : \quad \mathbb{R}^m \mapsto \mathcal{M} \quad ; \quad \tau \mapsto \mathcal{X} = \text{Exp}(\tau) \tag{3.41}$$

$$\text{Log} : \quad \mathcal{M} \mapsto \mathbb{R}^m \quad ; \quad \mathcal{X} \mapsto \tau = \text{Log}(\mathcal{X}). \tag{3.42}$$

In the case of the SO(3), these maps have closed form solutions given with

$$\mathbf{R} = \mathrm{Exp}(\theta\mathbf{u}) \triangleq \mathbf{I} + \sin\theta[\mathbf{u}]_\times + (1-\cos\theta)[\mathbf{u}]_\times^2 \in \mathbb{R}^{3\times 3} \tag{3.43}$$

$$\theta\mathbf{u} = \log(\mathbf{R}) \triangleq \frac{\theta(\mathbf{R}-\mathbf{R}^\mathsf{T})^\vee}{2\sin\theta} \in \mathbb{R}^3 \tag{3.44}$$

$$\theta = \cos^{-1}\left(\frac{\mathrm{trace}(\mathbf{R})-1}{2}\right). \tag{3.45}$$

In the derivations, it is often necessary to increment an element of a manifold with an element in its tangent vector space. These operation can be easily performed using the introduced concepts. However, to enable intuitive and concise notation, new operators *boxplus* and *boxminus* are introduced

$$\oplus: \quad \mathcal{Y} = \mathcal{X} \oplus {}^{\mathcal{X}}\tau \triangleq \mathcal{X} \circ \mathrm{Exp}\left({}^{\mathcal{X}}\tau\right) \in \mathcal{M} \tag{3.46}$$

$$\ominus: \quad {}^{\mathcal{X}}\tau = \mathcal{Y} \ominus \mathcal{X} \triangleq \mathrm{Log}\left(\mathcal{X}^{-1} \circ \mathcal{Y}\right) \in T_{\mathcal{X}}\mathcal{M}. \tag{3.47}$$

It should be noted that these operator are right versions because term $\mathrm{Exp}\left({}^{\mathcal{X}}\tau\right)$ appears at the right hand side of composition, while an alternative left version is also available [190].

The hitherto introduced concepts enable elegant introduction of derivatives in the context of Lie groups. It will be examined through derivation of a vector-valued multivariate function $\mathbf{f}(\mathbf{x})$. As a reminder, derivative of such function on a vector space, called *Jacobian* matrix, is defined with

$$\mathbf{J} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n\times m}. \tag{3.48}$$

When the function takes element of Lie group as an argument, this definition has to be adapted. Derivatives are taken with respect to the tangential vector space, while all the required operations are already available

$$\begin{aligned} \frac{{}^{\mathcal{X}}Df(\mathcal{X})}{D\mathcal{X}} &\triangleq \lim_{\boldsymbol{\tau}\to 0} \frac{f(\mathcal{X}\oplus\boldsymbol{\tau}) \ominus f(\mathcal{X})}{\boldsymbol{\tau}} \quad \in \mathbb{R}^{n\times m} \\ &= \lim_{\boldsymbol{\tau}\to 0} \frac{\mathrm{Exp}\left(f(\mathcal{X})^{-1} \circ f(\mathcal{X}\circ\mathrm{Exp}(\boldsymbol{\tau}))\right)}{\boldsymbol{\tau}} \\ &= \left.\frac{\partial\,\mathrm{Exp}\left(f(\mathcal{X})^{-1} \circ f(\mathcal{X}\circ\mathrm{Exp}(\boldsymbol{\tau}))\right)}{\partial\boldsymbol{\tau}}\right|_{\boldsymbol{\tau}=0}. \end{aligned} \tag{3.49}$$

Similarly to the box operators, Lie group Jacobian also have their left and right form. In practice, they are both used for the same purpose, but they yield different expression and depending on the context one can be preferable over another, e.g. easier derivation or shorter final expressions. To illustrate the elegance of such approach, consider a simple Jacobian derivation for the group action $f(\mathbf{R}) = \mathbf{R}\cdot\mathbf{p}$ of the SO(3) group:

$$\begin{aligned} \frac{{}^{\mathbf{R}}Df(\mathbf{R})}{D\mathbf{R}} &= \lim_{\boldsymbol{\theta}\to 0} \frac{(\mathbf{R}\oplus\boldsymbol{\theta})\mathbf{p} \ominus \mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} = \lim_{\boldsymbol{\theta}\to 0} \frac{\mathbf{R}\,\mathrm{Exp}(\boldsymbol{\theta})\mathbf{p} - \mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} = \lim_{\boldsymbol{\theta}\to 0} \frac{\mathbf{R}\left(\mathbf{I}+[\boldsymbol{\theta}]_\times\right)\mathbf{p} - \mathbf{R}\mathbf{p}}{\boldsymbol{\theta}} \\ &= \lim_{\boldsymbol{\theta}\to 0} \frac{\mathbf{R}[\boldsymbol{\theta}]_\times\mathbf{p}}{\boldsymbol{\theta}} = \lim_{\boldsymbol{\theta}\to 0} \frac{-\mathbf{R}[\mathbf{p}]_\times\boldsymbol{\theta}}{\boldsymbol{\theta}} = -\mathbf{R}[\mathbf{p}]_\times \end{aligned} \tag{3.50}$$

Applications of Lie groups in robotics depend on many other important concepts such as *adjoint* and manifold integration [190]. However, the concepts laid out hitherto equip us to introduce a relevant Lie group application used within this thesis, on-manifold optimization.

3.3.2 *On-manifold optimization*

Non-linear optimization is a backbone of calibration as described in Sec. 1.1, while extrinsic calibration is parametrized with the SE(3) group. Thus, on-manifold calibration is particularly useful within the context of sensor calibration. For instance, usage of analytical Jacobians obtained using the technique provided in Sec. 3.3.1 enables significant computational improvements, especially with multi-sensor calibration problems. In this section, a brief introduction of on-manifold nonlinear least squares optimization is given, as it set a base for contributions of this thesis.

To follow the standard Maximum Likelihood Estimation (MLE) framework [192], the goal of optimization is to find a distribution of parameters $\mathcal{X}$, given the measurements $\mathbf{z}_{1:N}$ using the Bayes rule

$$p\left(\boldsymbol{x}|\boldsymbol{z}_{1:N}\right) = \frac{p\left(\boldsymbol{z}_{1:N}|\mathbf{x}\right) \cdot p\left(\boldsymbol{x}\right)}{p\left(\boldsymbol{z}_{1:N}\right)}. \tag{3.51}$$

With the assumption of no prior knowledge about the states, i.e. uniform prior, it reduces $p\left(\boldsymbol{x}\right)$ to a constant c_x. Similarly, since the measurement distribution $p\left(\boldsymbol{z}_{1:N}\right)$ is not changing and does not depend on the states, it also reduces to a constant c_z. By dropping the constant terms and treating the measurements as independent, the following expression emerges:

$$p\left(\boldsymbol{x}|\boldsymbol{z}_{1:N}\right) \propto \prod_{k=1}^{N} p\left(\boldsymbol{z}_k|\boldsymbol{x}\right). \tag{3.52}$$

where we treat the observation model as a Gaussian random variable

$$p(\mathbf{z}_k|\mathbf{x}) = \mathcal{N}(\mathbf{h}_k(\mathbf{x}) \ , \ \Omega_k^{-1}) \tag{3.53}$$

or alternatively

$$p\left(\boldsymbol{z}_k|\boldsymbol{x}\right) \propto \exp\left(-\left(\mathbf{h}_k(\mathbf{x}) - \boldsymbol{z}_k\right)^T \boldsymbol{\Omega}_k \left(\mathbf{h}_k(\mathbf{x}) - \boldsymbol{z}_k\right)\right). \tag{3.54}$$

where a mean of predicted measurement $\mathbf{h}_k(\mathbf{x})$ is a generic non-linear function of the state $\mathbf{x}$, commonly called *measurement function*.

Since the observation model is available, it is easier to maximize the right-hand side likelihood of the Eq. (3.52), which is equivalent to finding the optimal solution $\boldsymbol{x}^*$ by minimizing the following expression:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \left(F(\boldsymbol{x}) \right) \tag{3.55}$$

$$F(\boldsymbol{x}) = \sum_{k=1}^{N} \boldsymbol{e}_k^T(\boldsymbol{x}) \boldsymbol{\Omega}_k \boldsymbol{e}_k(\boldsymbol{x}) \tag{3.56}$$

$$\boldsymbol{e}_k(\boldsymbol{x}) = \boldsymbol{h}_k(\boldsymbol{x}) - \boldsymbol{z}_k. \tag{3.57}$$

When the state is a Lie group, i.e. $\mathbf{x} \in \mathcal{X}$, the problem can be solved by following on-manifold Gauss-Newton optimization framework [186]. Starting with an initial guess of the states $\boldsymbol{x}_0 \in \mathcal{X}$, it is used as current estimate $\hat{\boldsymbol{x}} \in \mathcal{X}$ to evaluate the errors (3.57). Next, the optimal state perturbation $\Delta\boldsymbol{x} \in \mathbb{R}^m$ is found by linearizing the error term (3.57) around $\hat{\boldsymbol{x}} \boxplus \Delta\boldsymbol{x}$ using the first-order Taylor approximation:

$$\boldsymbol{e}_k(\hat{\boldsymbol{x}} \boxplus \Delta\boldsymbol{x}) \approx \mathbf{e}_k(\hat{\boldsymbol{x}}) + \underbrace{\left.\frac{\partial \boldsymbol{e}_k(\hat{\boldsymbol{x}} \boxplus \Delta\boldsymbol{x})}{\partial \Delta\boldsymbol{x}}\right|_{\Delta\boldsymbol{x}=0}}_{\boldsymbol{J}_i} \Delta\boldsymbol{x}. \tag{3.58}$$

After substituting linearized error (3.58) into (3.56) to obtain linearized criterion, the following quadratic form appears

$$F(\hat{\boldsymbol{x}} \boxplus \Delta\boldsymbol{x}) \approx \Delta\boldsymbol{x}^T \boldsymbol{H} \Delta\boldsymbol{x} + 2\boldsymbol{b}^T \Delta\boldsymbol{x} + \sum_{k=1}^{N} \boldsymbol{e}_k^T(\hat{\boldsymbol{x}}) \boldsymbol{\Omega}_k \boldsymbol{e}_k(\hat{\boldsymbol{x}}) \tag{3.59}$$

where

$$\boldsymbol{H} = \sum_{k=1}^{N} \boldsymbol{J}_k^\top \boldsymbol{\Omega}_{z,k} \boldsymbol{J}_k \tag{3.60}$$

$$\boldsymbol{b} = \sum_{k=1}^{N} \boldsymbol{J}_k^\top \boldsymbol{\Omega}_{z,k} \boldsymbol{e}_k. \tag{3.61}$$

Furthermore, optimal perturbation vector at each iteration is found by equating the derivative of the (3.59) with zero:

$$\Delta\boldsymbol{x} = -\boldsymbol{H}^{-1}\boldsymbol{b}. \tag{3.62}$$

It is then used to update the current state estimate using with $\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}} \boxplus \Delta\boldsymbol{x}$ and the process is repeated until convergence. Final ingredient, the analytical Jacobian, is found by following the procedure presented in Sec. 3.3.1.

4

The main scientific contributions of the thesis

The goal of thesis research was to develop extrinsic and temporal calibration algorithms for heterogeneous sensor systems. It was accomplished through three main contributions. The first contribution introduced a target and a calibration method for an accurate extrinsic calibration of a radar-camera-lidar sensor system [Pub1, Pub2]. The second contribution involved development of a general extrinsic and temporal calibration method based on target tracking represented with continuous-time GP regression [Pub4]. Furthermore, it led to application of continuous-time estimation paradigm in targetless camera – radar calibration [Pub5]. The third contribution proposed a method for unsupervised online graph-based calibration based on moving object tracking [Pub3]. Further discussion on individual contributions is given in the sequel.

#1 *A method for six degrees of freedom extrinsic calibration of radar – camera – lidar sensor system enhanced by radar cross section measurement evaluation*

Extrinsic calibration of heterogeneous sensors is a challenging task because such sensors measure different physical phenomena and provide diverse data. To overcome this challenge and design an accurate and efficient method, calibration targets are often used. By relying on targets, the search for correspondences between the sensor measurements is simplified, while a priori information about the target can improve the estimate accuracy. Furthermore, identifiability analysis ensures that the solution to the problem is attainable, while it provides guidelines on how to properly design a data collection procedure.

The first contribution deals with 6 DOF extrinsic calibration of a 3D lidar – radar – camera system. The method includes a universal target design suitable for the lidar, radar and camera introduced in [Pub1]. The calibration target consists of a styrofoam triangle which is invisible to the radar while it has good properties for detection and localization in the point cloud and image. Radar receives the echo from the trihedral corner reflector which has high radar cross section (RCS) and low orientation sensitivity. A novel two-step optimization procedure that enables full and accurate 6 DOF calibration is presented in [Pub1]. The first step is based on the reprojection error minimization while the second step, i.e. RCS optimization, uses space distribution of RCS to estimate variables which are not identifiable from the reprojection error due to the lack of radar's vertical resolution. The subset of parameters refined by the second step includes translation in vertical axis, roll and pitch angles. A particular parametrization of extrinsic calibration is chosen to enable

the highest spread of uncertainty among the parameters consistently, thus reinforcing the two-step procedure and the locking of the parameters.

The first version of RCS optimization [Pub1] is based on predefined nominal radar FOV and RCS threshold. The optimization criterion tries to encompass all the measurements with high RCS value within the nominal radar FOV. In the second version of the RCS optimization proposed in [Pub2], criterion is changed which lead to more accurate results, while it removed nominal FOV and RCS threshold as requirements. Instead, the optimization estimates parameters of the elevation – RCS curve that best explain the data. While the nuisance curve parameters are of no practical interest, they lead to an improved extrinsic calibration since they relate lidar measured elevation with radar measured RCS. Furthermore, [Pub2] extends the approach in [Pub1] by adding a camera to the lidar – radar calibration framework. Additionally, a thorough identifiability analysis of the reprojection error optimization is carried out in [Pub2]. It clarifies the uncertainty spread hypothesis for different sensor configurations, confirms the need for the second optimization step and provides guidelines on data acquisition for reliable calibration. The method was tested in both simulations and in real world experiments with two different radar systems. The results showed that it is possible to consistently and accurately estimate all the 6 DOF of extrinsic calibration. Lastly, the method was used to carry out radar vertical alignment assessment with the aid of lidar estimated ground plane.

#2 *A method for extrinsic and temporal calibration of heterogeneous exteroceptive mobile robot sensor systems based on object tracking using Gaussian process regression*

Temporal calibration requires motion, either the sensor system's ego-motion or the motion of the objects it perceives. While the ego-motion estimation is a reliable source of information for calibration, not all the sensors can perform it well, e.g. radar. Furthermore, static sensors systems are deprived of any ego-motion. Thus, the second contribution relies on using moving targets for extrinsic and temporal, i.e. spatiotemporal, calibration of heterogeneous sensors. The only requirement on the sensors is the ability to estimate 3D position of the object which is feasible for a number of different sensors, e.g. camera, lidar, radar, motion capture (MOCAP) system, etc.

The backbone of the spatiotemporal calibration method presented in [Pub4] is the GP regression. The target trajectories were described using the GP regression to provide smoothed continuous-time representations which allow precise temporal correspondence registration and calibration. By abstracting the sensor measurements with continuous-time trajectories, correspondence registration between asynchronous sensors with different frame-rates becomes seamless. Furthermore, the trajectory alignment that achieves the calibration is formulated through on-manifold optimization framework. Although smooth trajectory estimates provided by the GPs are required, this approach enabled computationally inexpensive, but accurate spatiotemporal calibration. Furthermore, the method enabled estimation of both the time delay and sensor clock drift between the sensors.

The proposed method was extensively tested in simulations and real world experiments with 4 different types of sensors: camera, lidar, radar and MOCAP system. It was shown that the method is able to estimate time delay with the error up to a fraction of the shortest

sampling time, e.g. error less than 0.8 ms for cameras operating with 50 ms sampling time. Furthermore, accurate temporal calibration enabled seamless extrinsic calibration, even with highly dynamic target motion. The importance of temporal calibration was shown on camera – MOCAP fusion where the method enabled reduction of average reprojection error from 1.9 cm downto 0.5 cm. Lastly, the open-sourced implementation of GP regression and calibration method enabled computationally inexpensive and scalable solution. Namely, one minute interval of measurements at 20 Hz required only 49 ms for GP regression and 41 ms for optimization.

#3 *An online unsupervised graph-based method for extrinsic and temporal calibration of heterogeneous exteroceptive mobile robot sensor systems*

Lifelong operation of a robotic system is highly dependant on reliable calibration which can degrade over time. To overcome this challenge, online calibration method aim at using information from the environment as correspondences between the sensors. Detection and tracking of moving objects such as vehicles and pedestrians is often performed with every sensor on a robotic platform, while it provides abundant source of information for the calibration. The third contribution of the thesis presented in [Pub3] extends the moving target based calibration by releasing the requirement for a known target and adding several features that enable efficient online decalibration detection and recalibration.

The method consists of a standard pipeline for detection and tracking of moving objects using radars, cameras and lidars. A novel technique for the track to track association between the sensors resistant to decalibration is introduced. Furthermore, a lightweight pairwise calibration is presented enabling inexpensive miscalibration detection and initialization for the recalibration. As a final ingredient, the method enables graph-based global calibration of all the sensor simultaneously to provide a consistent recalibration of the whole system.

The proposed method was tested on publicly available dataset for autonomous vehicle development involving radar, camera and a lidar. From the common traffic participants, the method processed the data from only the surrounding vehicles as they could be reliably tracked by all the sensor modalities. The results showed that the method is capable of detecting small rotational miscalibration within a few seconds, as well as recalibrating the whole system on the fly. Furthermore, it was shown how this approach outperformed the more common ego-motion based calibration during uninformative driving segments.

5

Conclusions and future work

Mobile robotics is increasingly entering our daily lives through technologies such as autonomous vehicles, warehouse robots and drones. The basis of their operation involves interaction with a dynamic environment. To enable safe operation for both the robots as well as all the other agents in the environment, a robust environment perception is essential. The standard and the most promising approach is sensor fusion of heterogeneous sensors. However, quality of the sensor calibration predetermines whether the fusion will succeed or fail miserably. While online sensor calibration enables long-term autonomy of mobile robots, its complexity and absence of information in the environment make it a challenging task. On the other hand, offline calibration provides reliable results in controlled environments. They are still widely used and will serve as a baseline for development of the online methods. With increasing market interest in the environment perception, novel sensors and sensor configuration are emerging frequently. Thus, it is essential to develop novel calibration techniques for them to pave the way to the ultimate goal, long-term autonomous navigation in highly dynamic environments.

5.1 The main conclusions of the thesis

This thesis deals with both offline and online calibration of sensor systems involving radars, lidars and cameras. While lidars and cameras are rich sources of information, radars proved to be a more challenging sensing modality. While their resistance to inclement weather is a particularly desirable trait, their low informativeness highly limits the possibilities for sensor calibration. Nevertheless, several novel methods for calibration of radars with other sensors, both offline and online, were introduced. Three main contribution of the thesis revolve around different techniques for extrinsic and temporal calibration of radars with other exteroceptive sensors, primarily lidars and cameras. Through development of these methods, novel target design, correspondence registration and optimization frameworks were introduced. In the sequel, conclusions drawn from each contribution are elaborated.

The first contribution of the thesis encompassed target design, two-step optimization procedure and identifiability analysis of the extrinsic target-based radar – lidar – camera calibration. The developed target enabled seamless detection and accurate unambiguous localization for all the sensor modalities. Furthermore, its consistent RCS enabled discovery of RCS – elevation effect in radar data that led to significant improvements in 6 DOF extrinsic calibration. The proposed two-step optimization with a particular extrinsic parametriza-

tion led to subdegree and centimetre level accuracy. The extensive statistical analysis of the FIM confirmed the intuitive reasoning for the particular choice of parametrization. It provided theoretical evidence of uneven uncertainty among the extrinsic calibration parameters. Furthermore, it showed that the spread of uncertainty in different directions is consistent regardless of the sensor configuration. In addition, FIM test showed that 4 non-coplanar points can theoretically lead to identifiability of only the first step, reprojection error optimization. However, the shown uneven uncertainty among parameters that was observed through the experiments confirmed the need for additional refinement step, the RCS optimization. Extensive experiments involving radars from two different manufacturers confirmed the existence of the RCS – elevation effect in the radar measurements. Finally, accurate 6DOF extrinsic calibration enables convenient method for the detection of radar's vertical misalignment.

The main goal of the second contribution was to the extend the developed method in the first contribution with temporal calibration using the movement of the target. The question of how to formalize the handling of asynchronous sensors with different frame rates emerged. It has led the research towards the state of the art in the continuous-time trajectory representations. The recent development of GP regression for the robotics problems and the goal of the thesis led to a fruitful combination of temporal calibration with the GPs. They have emerged as a particularly convenient tool for solving the temporal correspondence registration, while imposing low computational requirements and adaptation. Besides the theory of GP regression, recent progress in robotic applications of Lie Groups and particularly on-manifold optimization further improved the calibration pipeline. The proposed on-manifold optimization framework enabled an accurate and fast solution to joint estimation of extrinsic and temporal parameters, including both time delay and clock drift between the sensors. Furthermore, the solution proved scalable to multisensor calibration as well as high frame rate sensors due to GP regression's $\mathcal{O}(n)$ computational complexity. Namely, combination of estimated velocities using the GPs and the analytical Jacobian in the on-manifold optimization depending on velocities enabled development of a quick solver. Through extensive tests in simulations and real world experiments using 4 different sensor modalities, wide applicability of the method was proven, accompanied by accurate temporal and extrinsic calibration estimates. Lastly, comparison with the competing approaches further confirmed the need for smooth continuous-time representations in temporal calibration.

The third contribution developed on the previous one by relaxing the requirement for a known target. Instead, a novel method for online calibration of radar – lidar – camera system based on moving object detection and tracking was introduced. The method was designed to reuse the information of other perception systems on a mobile robot. Namely, while ego-motion, commonly used in calibration, is often estimated by only a subset of sensors, detection and tracking of moving objects is often performed using all the available sensors. In addition to reusing readily available information, the proposed method consisted of a lightweight miscalibration detection scheme to enable system reliability in long-term autonomy. A novel track-to-track association independent of calibration was introduced to mitigate the effects of the miscalibration. It was shown that using this approach, it is possible to detect moderate miscalibration within a few seconds. On the other hand, graph-based

optimization enabled recalibration of the whole system providing a globally consistent solution. Through extensive tests on real world data, it was shown that the most reliable objects in the environment for the purposes of calibration are the vehicles, since they are most reliably detected in all the sensor modalities. Furthermore, due to a high degree of bias in the detection stage, the calibration was limited to rotational calibration that was less affected by the bias. However, for all practical means and purposes, rotational miscalibration is far more hazardous and should be quickly detected upon its occurrence. Lastly, compared to a more common approach of using ego-motion, relying on moving objects proved more informative in certain situations. Namely, the proposed method was able to calibrate the sensors even during static periods as well as provide full rotational observability during straight line segments, which are particularly challenging for ego-motion methods.

5.2 FURTHER RESEARCH DIRECTIONS

Development of calibration methods throughout this thesis tackled various open research problems. While many of them were solved efficiently, it opened interesting research avenues within the sensor calibration, as well as other tasks in robotics. The first contribution introduced a universal target for radar – lidar – camera calibration. While it enabled efficient calibration of the considered system, its application to target-based calibration of different sensor modalities such as thermal or event based cameras poses an interesting challenge. Furthermore, application of FIM in the context of calibration can be further applied in active calibration with the aim of minimizing the calibration uncertainty.

The second contribution solved the temporal calibration with strong reliance on GP regression. It was shown how GPs proved to be a very useful tool in sensor calibration, while their use can be further developed. Namely, the application of GPs in ego-motion-based calibration has the potential to significantly improve the temporal calibration capabilities of current methods. Moreover, readily available estimate of trajectory uncertainty within the GPs could be used to enhance calibration results and enable more robust methods. Nevertheless, continuous-time methods have a huge potential in online sensor fusion approaches since they enable seamless temporal correspondence between asynchronous sensors. Developed on-manifold optimization could be used outside the calibration framework. For example, it can be considered as an ICP algorithm version that handles unknown temporal correspondences between the points. The proposed method's efficiency enables its online applications that could yield improved point cloud registration techniques.

The third contribution tackled the issue of online sensor calibration. While it relied on object detection within the pipeline, it opened questions about influence of miscalibration on detection stage. Namely, an interesting research avenue is exploration of miscalibration influence on algorithms for monocular 3D object detection and depth estimation. With proper treatment of the calibration, it might be possible to improve generalization capabilities of learning based methods that provide state of the art results in these fields. Lastly, each calibration approach has its strengths and weaknesses. Thus, an ideal online system should combine several methods to achieve the most reliable results. For instance, moving object-based methods could be fused with ego-motion-based methods to enable calibration in various situations, while relaxing the need for overlapping FOV. Real world systems have

increasing need for online calibration to enable long-term autonomy and scalability. In order to achieve that, every bit of information should be used in calibration and it should be fused in the optimal way.

6

List of publications

Pub1 J. Peršić, I. Marković and I. Petrović. Extrinsic 6DoF calibration of 3D lidar and radar. *IEEE European Conference on Mobile Robots (ECMR)*. Paris, France, 1–6, 2017.

Pub2 J. Peršić, I. Marković and I. Petrović. Extrinsic 6DoF calibration of a radar – LiDAR – camera system enhanced by radar cross section estimates evaluation. *Robotics and Autonomous Systems*, 114:217–230, 2019, IF: 2.825 (Q2).

Pub3 J. Peršić, L. Petrović, I. Marković and I. Petrović. Online multi-sensor calibration based on moving object tracking. *Advanced Robotics*, 35(3-4):130-140, 2021, IF: 1.247 (Q4).

Pub4 J. Peršić, L. Petrović, I. Marković and I. Petrović. Spatiotemporal Multisensor Calibration via Gaussian Processes Moving Target Tracking. *IEEE Transactions on Robotics*, Early Access, 2021, IF: 6.123 (Q1).

Pub5 E. Wise, J. Peršić, C. Grebe, I. Petrović and J. Kelly. A Continuous-Time Approach for 3D Radar-to-Camera Extrinsic Calibration. *IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China, 2021 (accepted).

7

Author's contribution to publications

The results presented in this thesis are based on the research carried out in the Laboratory for autonomous systems and mobile robotics (LAMOR) headed by Professor Ivan Petrović, at the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia during the period of 2016 – 2021 as a part of a research project:

- [2016 – 2020] *SafeTram - System for increased driving safety in public urban rail traffic* (KK.01.2.1.01.0022) which was financially supported by the European Union from the European Regional Development Fund.

A part of the thesis also includes the research carried out at the Space and Terrestrial Autonomous Robotic Systems (STARS) laboratory headed by Professor Jonathan Kelly, at the University of Toronto Institute for Aerospace Studies (UTIAS), Canada. The collaboration was financially supported by Prof. Dr. SC. Jasna Simunic-Hrvoic Foundation.

The thesis includes five publications written in collaboration with co-authors of the published papers. The author's contribution to each paper consists of the method design, software implementation, testing in simulations and real world experiments, result analysis and written presentation.

Pub1 In the paper entitled *Extrinsic 6DoF calibration of 3D lidar and radar* the author proposed a novel target design and two-step optimization framework for extrinsic calibration of 3D lidar and radar. The connection between radar's RCS estimation error and elevation angle was discovered and used to enhance calibration results. The author constructed a calibration target and implemented the proposed method in Matlab. The author conducted a real world experiment involving a mobile robot, radar and 3D lidar to confirm the applicability of the method. Results from the experiment were thoroughly analyzed confirming the ability of the method to estimate 6 DOF extrinsic calibration between the radar and 3D lidar.

Pub2 In the paper entitled *Extrinsic 6DoF calibration of a radar – LiDAR – camera system enhanced by radar cross section estimates evaluation* the author extended the method presented in [Pub1] by enabling additional calibration of the camera with the other sensors. It involved target modification and implementation of algorithms for image processing. The author modified the second step of the proposed optimization, i.e. RCS optimization, by modelling the RCS – elevation effect as a parabolic curve. It

led to improved results with fewer tuning parameters. The author conducted an identifiability analysis using FIM which confirmed uneven uncertainty among the estimated parameters, proved the correct choice of parametrization and provided guidelines on data acquisition process. The author conducted a thorough analysis of results in simulations and real world experiments, where camera, 3D lidar and radars from two different manufactures were used to confirm the discovered effect. Finally, the author applied the estimated 6 DOF calibration in assessment of radar's vertical misalignment by implementing ground plane detection in lidar data and transforming it into the radar frame.

Pub3 In the paper entitled *Online multi-sensor calibration based on moving object tracking* the author proposed a novel framework for online calibration of radar – lidar – camera system based on detection and tracking of moving objects. The framework included calibration-agnostic track-to-track association, lightweight decalibration detection and pairwise calibration and complete rotational calibration of the system based on graph optimization. The author implemented detection, tracking, decalibration detection and graph-based recalibration algorithms in Matlab. Furthermore, the method was extensively tested on a publicly available dataset for development of AVs, and compared to a state of the art ego-motion based method.

Pub4 In the paper entitled *Spatiotemporal Multisensor Calibration via Gaussian Processes Moving Target Tracking* the author proposed a novel spatiotemporal calibration method based on moving target tracking which relied on GPs for continuous-time trajectory representation. Furthermore, the author proposed an on-manifold optimization framework that jointly estimates extrinsic and temporal calibration parameters, including time delay and clock drift. The method was further extended with multi-sensor capabilities, enabling joint calibration of arbitrary number of sensors. The author implemented the method in both Matlab and C++. The efficient implementation of Exactly Sparse GP regression and ROS package for calibration were open-sourced. The author verified the accuracy, robustness and applicability of the method in both simulations and real world experiments including four different sensor modalities.

Pub5 In the paper entitled *A Continuous-Time Approach for 3D Radar-to-Camera Extrinsic Calibration*, in collaboration with co-authors, the author proposed a novel method for targetless calibration of 3D radar and camera based on continuous-time trajectory representation. The author contributed in conducting the real world experiment, including sensor rig design and data collection. Furthermore, the author conducted a comparison of the method with the competing state of the art approach.

BIBLIOGRAPHY

[1] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis. 3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *International Journal of Robotics Research*, 31(4):452–467, 2012.

[2] Paul Bergmann, Rui Wang, and Daniel Cremers. Online photometric calibration of auto exposure video for realtime visual odometry and SLAM. *IEEE Robotics and Automation Letters*, 3(2):627–634, 2017.

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint*, (March), 2019.

[4] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford radar RobotCar Dataset: A radar extension to the Oxford RobotCar Dataset. In *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, pages 6433–6438, 2020.

[5] Josip Ćesić, Ivan Marković, Igor Cvišić, and Ivan Petrović. Radar and stereo vision fusion for multitarget tracking on the special Euclidean group. *Robotics and Autonomous Systems*, 83:338–348, 2016.

[6] Tokihiko Akita and Seiichi Mita. Object tracking and classification using millimeter-wave radar based on LSTM. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1110–1115, 2019.

[7] Dominik Kellner, Michael Barjenbruch, Jens Klappstein, Jürgen Dickmann, and Klaus Dietmayer. Instantaneous ego-motion estimation using Doppler radar. In *Intelligent Transportation Systems (ITSC)*, pages 869–874, 2013.

[8] Andrew Kramer, Carl Stahoviak, A. Santamaria-Navarro, Ali Akbar Agha-Mohammadi, and Christoffer Heckman. Radar-inertial ego-velocity estimation for visually degraded environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5739–5746, 2020.

[9] Jiunn-Kai Huang, Chenxi Feng, Madhav Achar, Maani Ghaffari, and Jessy W. Grizzle. Global unifying intrinsic calibration for spinning and solid-state LiDARs. *arXiv preprint*, pages 1–14, 2020.

[10] Biao Zhang, Xiaoyuan Zhang, Baochen Wei, and Chenkun Qi. A point cloud distortion removing and mapping algorithm based on lidar and IMU UKF fusion. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 966–971. IEEE, 2019.

[11] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2 edition, 2003.

[12] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5695–5701, 2006.

[13] Davide Scaramuzza. Omnidirectional vision: from calibration to robot motion estimation. *ETH Zurich, PhD Thesis*, (17635):189, 2008.

[14] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[15] Zhengyou Zhang. A flexible new technique for camera calibration (technical report). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2002.

[16] Luc Oth, Paul Furgale, Laurent Kneip, and Roland Siegwart. Rolling shutter camera calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1360–1367. IEEE, 2013.

[17] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1112, 1997.

[18] John Mallon and Paul F. Whelan. Which pattern? Biasing aspects of planar calibration patterns and detection methods. *Pattern Recognition Letters*, 28(8):921–930, 2007.

[19] Hyowon Ha, Michal Perdoch, Hatem Alismail, In So Kweon, and Yaser Sheikh. Deltille grids for geometric camera calibration. In *IEEE International Conference on Computer Vision (ICCV)*, number 2, pages 5354–5362, 2017.

[20] Bo Li, Lionel Heng, Kevin Koser, and Marc Pollefeys. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1301–1307, 2013.

[21] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. *International Conference on Robotics and Automation (ICRA)*, pages 3400–3407, 2011.

[22] Andrew Richardson, Johannes Strom, and Edwin Olson. AprilCal: Assisted and repeatable camera calibration. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1814–1821, 2013.

[23] Ziran Xing, Jingyi Yu, and Yi Ma. A new calibration technique for multi-camera systems of limited overlapping field-of-views. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5892–5899, 2017.

[24] Gerardo Atanacio-Jiménez, José-Joel González-Barbosa, Juan B. Hurtado-Ramos, Francisco J. Ornelas-Rodríguez, Hugo Jiménez-Hernández, Teresa García-Ramirez, and Ricardo González-Barbosa. LIDAR Velodyne HDL-64E calibration using pattern planes. *International Journal of Advanced Robotic Systems*, 8(5):70–82, 2011.

[25] Naveed Muhammad and Simon Lacroix. Calibration of a rotating multi-beam Lidar. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5648–5653, 2010.

[26] Jaehyeon Kang and Nakju Lett Doh. Full-DOF calibration of a rotating 2-D LIDAR with a simple plane measurement. *IEEE Transactions on Robotics*, 32(5):1245–1263, 2016.

[27] Eduardo Fernández-Moral, Javier González-Jiménez, and Vicente Arévalo. Extrinsic calibration of 2D laser rangefinders from perpendicular plane observations. *International Journal of Robotics Research*, 34(11):1401–1417, 2015.

[28] Marcelo Pereira, Vitor Santos, and Paulo Dias. Automatic calibration of multiple LIDAR sensors using a moving sphere as target. *Advances in Intelligent Systems and Computing*, 417:477–489, 2016.

[29] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2301–2306, 2004.

[30] Gaurav Pandey, James McBride, Silvio Savarese, and Ryan Eustice. Extrinsic calibration of a 3D laser scanner and an omnidirectional camera. In *IFAC Symposium on Intelligent Autonomous Vehicles*, pages 336–341, 2010.

[31] Lipu Zhou and Zhidong Deng. Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation. *IEEE Intelligent Vehicles Symposium (IV)*, pages 642–648, 2012.

[32] Andreas Geiger, Frank Moosmann, Omer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *IEEE Conference on Robotics and Automation (ICRA)*, pages 3936–3943, 2012.

[33] Jason L. Owens, Philip R. Osteen, and Kostas Daniilidis. MSG-cal: Multi-sensor graph-based calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3660–3667, 2015.

[34] Martin Velas, Michal Spanel, Zdeněk Materna, and Adam Herout. Calibration of RGB camera with Velodyne LiDAR. *WSCG 2014 Communication Papers*, pages 135–144, 2014.

[35] Kiho Kwak, Daniel F. Huber, Hernan Badino, and Takeo Kanade. Extrinsic calibration of a single line scanning lidar and a camera. In *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, pages 3283–3289, 2011.

[36] Dorit Borrmann, Hassan Afzal, Jan Elseberg, and Andreas Nüchter. Mutual calibration for 3D thermal mapping. *IFAC Proceedings Volumes*, 45(22):605–610, 2012.

[37] Michal R. Nowicki. Spatiotemporal calibration of camera and 3D laser scanner. *IEEE Robotics and Automation Letters*, 5(4):6451–6458, 2020.

[38] Eugene F. Knott. *Radar cross section measurements.* ITP Van Nostrand Reinhold, 1993.

[39] Tao Wang, Nanning Zheng, Jingmin Xin, and Zheng Ma. Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications. *Sensors*, 11(9):8992–9008, 2011.

[40] Shigeki Sugimoto, Hayato Tateda, Hidekazu Takahashi, and Masatoshi Okutomi. Obstacle detection using millimeter-wave radar and its visualization on image sequence. In *International Conference on Pattern Recognition (ICPR)*, pages 342–345, 2004.

[41] Ghina El Natour, Omar Ait Aider, Raphael Rouveure, Francois Berry, and Patrice Faure. Radar and vision sensors calibration for outdoor 3D reconstruction. In *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, pages 2084–2089, 2015.

[42] Chia-Le Lee, Yu-Han Hsueh, Chieh-Chih Wang, and Wen-Chieh Lin. Extrinsic and temporal calibration of automotive radar and 3D LiDAR. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9976–9983, 2021.

[43] Jiyong Oh, Ki-seok Kim, Miryong Park, and Sungho Kim. A comparative study on camera-radar calibration methods. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1057–1062. IEEE, 2018.

[44] Joris Domhof, Julian F P Kooij, and Dariu M Gavrila. A multi-sensor extrinsic calibration tool for lidar , camera and radar. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2019.

[45] Joris Domhof, Julian F.P. Kooij, and Dariu M. Gavrila. A joint extrinsic calibration tool for radar, camera and lidar. *IEEE Transactions on Intelligent Vehicles*, 2019.

[46] L. Barazzetti, L. Mussio, F. Remondino, and M. Scaioni. Targetless camera calibration. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/:335–342, 2011.

[47] Clive Fraser and Christos Stamatopoulos. Automated target-free camera calibration. In *Annual American Society for Photogrammetry and Remote Sensing (ASPRS) Conference*, 2014.

[48] Torsten Sattler, Chris Sweeney, and Marc Pollefeys. On sampling focal length values to solve the absolute pose problem. In *European Conference on Computer Vision (ECCV)*, pages 828–843, 2014.

[49] Chaoning Zhang, Francois Rameau, Junsik Kim, Dawit Mureja Argaw, Jean Charles Bazin, and In So Kweon. DeepPTZ: Deep self-calibration for PTZ cameras. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1030–1038, 2020.

[50] Andrei Cramariuc, Aleksandar Petrov, Rohit Suri, Mayank Mittal, Roland Siegwart, and Cesar Cadena. Learning camera miscalibration detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4997–5003, 2020.

[51] Viktor Kocur and Milan Ftáčnik. Traffic camera calibration via vehicle vanishing point detection. In *arXiv preprint*, pages 1–12, 2021.

[52] Yunhai Han, Yuhan Liu, David Paz, and Henrik Christensen. Auto-calibration method using stop signs for urban autonomous driving applications. In *arXiv preprint*, pages 1–7, 2020.

[53] Syed Ammar Abbas and Andrew Zisserman. A geometric approach to obtain a bird's eye view from an image. In *International Conference on Computer Vision Workshop (ICCVW)*, pages 4095–4104, 2019.

[54] Yonggen Ling and Shaojie Shen. High-precision online markerless stereo extrinsic calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1771–1778, 2016.

[55] Peter Hansen, Hatem Alismail, Peter Rander, and Brett Browning. Online continuous stereo extrinsic parameter estimation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1059–1066, 2012.

[56] Thao Dang, Christian Hoffmann, and Christopher Stiller. Continuous stereo self-calibration by camera parameter tracking. *IEEE Transactions on Image Processing*, 18(7):1536–1550, 2009.

[57] Eike Rehder, Christian Kinzig, Philipp Bender, and Martin Lauer. Online stereo camera calibration from scratch. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1694–1699, 2017.

[58] Steffen Urban, Sven Wursthorn, Jens Leitloff, and Stefan Hinz. MultiCol bundle adjustment: A generic method for pose estimation, simultaneous self-calibration and reconstruction for arbitrary multi-camera systems. *International Journal of Computer Vision*, 121(2):234–252, 2017.

[59] Gerardo Carrera, Adrien Angeli, and Andrew J. Davison. SLAM-based automatic extrinsic calibration of a multi-camera rig. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2652–2659, 2011.

[60] Omer Faruk Ince and Jun-Sik Kim. Accurate on-line extrinsic calibration for a multi-camera SLAM system. In *International Conference on Ubiquitous Robots (UR)*, pages 540–545, 2020.

[61] Lionel Heng, Bo Li, and Marc Pollefeys. CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1793–1800, 2013.

[62] Lionel Heng, Paul Furgale, and Marc Pollefeys. Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig. *Journal of Field Robotics*, 32(5):775–802, 2015.

[63] Yukai Lin, Viktor Larsson, Marcel Geppert, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Infrastructure-based multi-camera calibration using radial projections. *arXiv preprint*, pages 1–17, 2020.

[64] Nima Keivan and Gabe Sibley. Online SLAM with any-time self-calibration and automatic change detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5775–5782, 2015.

[65] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A research platform for visual-inertial estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, number May, pages 4666–4672, 2019.

[66] Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang. MIMC-VINS: A versatile and resilient multi-IMU multi-camera visual-inertial navigation system. *arXiv preprint*, pages 1–20, 2020.

[67] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3565–3572, 2007.

[68] Will Maddern, Alastair Harrison, and Paul Newman. Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LIDARs. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3096–3102, 2012.

[69] Jesse Levinson and Sebastian Thrun. Unsupervised calibration for multi-beam lasers. *Springer Tracts in Advanced Robotics*, 79:179–193, 2014.

[70] Xiyuan Liu and Fu Zhang. Extrinsic alibration of multiple LiDARs of small FoV in targetless environments. *IEEE Robotics and Automation Letters*, 6(2):2036–2043, 2021.

[71] Lionel Heng. Automatic targetless extrinsic calibration of multiple 3D LiDARs and radars. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10669–10675, 2020.

[72] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems (RSS)*, 2013.

[73] Peyman Moghadam, Michael Bosse, and Robert Zlot. Line-based extrinsic calibration of range and image sensors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3685–3691, 2013.

[74] Xiaojin Gong, Ying Lin, and Jilin Liu. 3D LIDAR-camera extrinsic calibration using an arbitrary trihedron. *Sensors (Switzerland)*, 13(2):1902–1918, 2013.

[75] Gaurav Pandey, James R. McBride, Silvio Savarese, and Ryan M. Eustice. Automatic extrinsic calibration of vision and lidar by maximizing mutual information. *Journal of Field Robotics*, 32(5):696–722, 2015.

[76] Zachary Taylor and Juan Nieto. Automatic calibration of lidar and camera images using normalized mutual information. *IEEE Conference on Robotics and Automation (ICRA)*, 2013.

[77] Ashley Napier, Peter Corke, and Paul Newman. Cross-calibration of push-broom 2D LIDARs and cameras in natural scenes. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3679–3684, 2013.

[78] Chanoh Park, Peyman Moghadam, Soohwan Kim, Sridha Sridharan, and Clinton Fookes. Spatiotemporal camera-LiDAR calibration: A targetless and structureless approach. *IEEE Robotics and Automation Letters*, 5(2):1556 – 1563, 2020.

[79] Kaiwen Yuan, Zhenyu Guo, and Z. Jane Wang. RGGNet: Tolerance aware LiDAR-camera online calibration with geometric deep learning and generative model. *IEEE Robotics and Automation Letters*, 5(4):6956–6963, 2020.

[80] Davide Scaramuzza, Ahad Harati, and Roland Siegwart. Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4164–4169, 2007.

[81] Terry Scott, Akshay A. Morye, Pedro Pinies, Lina M. Paz, Ingmar Posner, and Paul Newman. Choosing a time and place for calibration of lidar-camera systems. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4349–4356, 2016.

[82] Hanqi Zhuang, Kuanchih Wang, and Zvi S. Roth. Simultaneous calibration of a robot and a hand-nounted camera. *IEEE Transactions on Robotics and Automation*, 11(5):649–660, 1995.

[83] Yiu Cheung Shiu and Shaheen Ahmad. Finding the mounting position of a sensor by solving a homogeneous transform equation of the form AX=XB. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1666–1671, 1987.

[84] Martin Kendal Ackerman, Alexis Cheng, Bernard Shiffman, Emad Boctor, and Gregory Chirikjian. Sensor calibration with unknown correspondence: Solving AX=XB using Euclidean-group invariants. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1308–1313, 2013.

[85] Qianli Ma, Haiyuan Li, and Gregory S. Chirikjian. New probabilistic approaches to the AX = XB hand-eye calibration without correspondence. In *IEEE Conference on Robotics and Automation (ICRA)*, pages 4365–4371, 2016.

[86] H Li, Q Ma, T Wang, and G S Chirikjian. Simultaneous hand-eye and robot-world calibration by solving the AX=YB problem without correspondence. *IEEE Robotics and Automation Letters*, 1(1):145–152, 2016.

[87] Sebastian Schneider, Thorsten Luettel, and Hans Joachim Wuensche. Odometry-based online extrinsic sensor calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1287–1292, 2013.

[88] Zachary Taylor and Juan Nieto. Motion-nased calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Transactions on Robotics*, 32(5):1215–1229, 2016.

[89] Jonathan Brookshire and Seth Teller. Extrinsic calibration from per-sensor egomotion. In *Robotics: Science and Systems (RSS)*, 2012.

[90] Kaihong Huang and Cyrill Stachniss. Extrinsic multi-sensor calibration for mobile robots using the Gauss-Helmert model. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1490–1496, 2017.

[91] Bartolomeo Della Corte, Henrik Andreasson, Todor Stoyanov, and Giorgio Grisetti. Unified motion-based calibration of mobile multi-sensor platforms with time delay estimation. *IEEE Robotics and Automation Letters*, 4(2):902–909, 2019.

[92] Markus Horn, Thomas Wodtko, Michael Buchholz, and Klaus Dietmayer. Online extrinsic calibration based on per-sensor ego-motion using dual quaternions. *IEEE Robotics and Automation Letters*, 6(2):982–989, 2021.

[93] Matthew Giamou, Ziye Ma, Valentin Peretroukhin, and Jonathan Kelly. Certifiably globally optimal extrinsic calibration from per-sensor egomotion. *IEEE Robotics and Automation Letters*, 4(2):367–374, 2019.

[94] Emmett Wise, Matthew Giamou, Soroush Khoubyarian, Abhinav Grover, and Jonathan Kelly. Certifiably optimal monocular hand-eye calibration. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2020-Septe:271–278, 2020.

[95] Frank Pagel and Dieter Willersinn. Motion-based online calibration for non-overlapping camera views. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 843–848, 2010.

[96] Celyn Walters, Oscar Mendez, Simon Hadfield, and Richard Bowden. A robust extrinsic calibration framework for vehicles with unscaled sensors. In *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, pages 36–42, 2019.

[97] Moritz Knorr, Wolfgang Niehsen, and Christoph Stiller. Online extrinsic multi-camera calibration using ground plane induced homographies. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 236–241. IEEE, 2013.

[98] David Zuñiga-Noël, Jose Raul Ruiz-Sarmiento, Ruben Gomez-Ojeda, and Javier Gonzalez-Jimenez. Automatic multi-sensor extrinsic calibration for mobile robots. *IEEE Robotics and Automation Letters*, 4(3):2862–2869, 2019.

[99] Jochen Schmidt and Heinrich Niemann. Data selection for hand-eye calibration: A vector quantization approach. *International Journal of Robotics Research*, 27(9):1027–1053, 2008.

[100] Faraz M Mirzaei and Stergios I Roumeliotis. A Kalman-filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5):1143–1156, 2008.

[101] Jonathan Kelly and Gaurav S Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.

[102] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1280–1286, 2013.

[103] Joern Rehder, Roland Siegwart, and Paul Furgale. A general approach to spatiotemporal calibration in multisensor systems. *IEEE Transactions on Robotics*, 32(2):383–398, 2016.

[104] Christiane Sommer, Vladyslav Usenko, David Schubert, Nikolaus Demmel, and Daniel Cremers. Efficient derivative computation for cumulative B-splines on lie groups. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[105] Hyungjin Kim, Sathya Narayanan, Kasturi Rangan, Shishir Pagad, and Veera Ganesh Yalla. Motion-based calibration between multiple LiDARs and INS with rigid body constraint on vehicle platform. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1787–1793, 2020.

[106] Cedric Le Gentil, Teresa Vidal-calleja, and Shoudong Huang. IN2LAAMA: Inertial lidar localization autocalibration and mapping. *IEEE Transactions on Robotics*, 37(1):275–290, 2021.

[107] Jiajun Lv, Jinhong Xu, Kewei Hu, Yong Liu, and Xingxing Zuo. Targetless calibration of lidar-imu system based on continuous-time batch estimation. In *arXiv preprint*, pages 1–8, 2020.

[108] Dominik Kellner, Michael Barjenbruch, Klaus Dietmayer, Jens Klappstein, and Juergen Dickmann. Joint radar alignment and odometry calibration. In *International Conference on Information Fusion*, pages 366–374, 2015.

[109] Chao X. Guo and Stergios I. Roumeliotis. An analytical least-squares solution to the odometer-camera extrinsic calibration problem. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2943–2948. IEEE, 2013.

[110] Rainer Kümmerle, Giorgio Grisetti, and Wolfram Burgard. Simultaneous parameter calibration, localization, and mapping. *Advanced Robotics*, 26(17):2021–2041, 2012.

[111] Jeremie Deray, Joan Sola, and Juan Andrade-Cetto. Joint on-manifold self-calibration of odometry model and sensor extrinsics using pre-integration. In *European Conference on Mobile Robotics (ECMR)*, pages 1–6, 2019.

[112] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

[113] Po Chang Su, Ju Shen, Wanxin Xu, Sen Ching S. Cheung, and Ying Luo. A fast and robust extrinsic calibration for RGB-D camera networks. *Sensors (Switzerland)*, 18(1):1–23, 2018.

[114] A. Fornaser, P. Tomasin, M. De Cecco, M. Tavernini, and M. Zanetti. Automatic graph based spatiotemporal extrinsic calibration of multiple Kinect V2 ToF cameras. *Robotics and Autonomous Systems*, 98:105–125, 2017.

[115] Florian Faion, Marcus Baum, Antonio Zea, and Uwe D. Hanebeck. Depth sensor calibration by tracking an extended object. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 19–24, 2015.

[116] Fengjun Lv, Tao Zhao, and Ramakant Nevatia. Camera calibration from video of a walking human. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1513–1518, 2006.

[117] Jaehoon Jung, Inhye Yoon, Sangkeun Lee, and Joonki Paik. Object detection and tracking-based camera calibration for normalized human height estimation. *Journal of Sensors*, 2016:1–9, 2016.

[118] Zheng Tang, Yen Shuo Lin, Kuan Hui Lee, Jenq Neng Hwang, Jen Hui Chuang, and Zhijun Fang. Camera self-calibration from tracking of moving persons. In *International Conference on Pattern Recognition (ICPR)*, pages 265–270, 2016.

[119] Dylan F. Glas, Takahiro Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. Automatic position calibration and sensor displacement detection for networks of laser range finders for human tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2938–2945, 2010.

[120] Dylan F Glas, Florent Ferreri, Takahiro Miyashita, and Hiroshi Ishiguro. Automatic calibration of laser range finder positions for pedestrian tracking based on social group detections. *Advanced Robotics*, 28(9):573–588, 2014.

[121] Dylan F. Glas, Drazen Brščić, Takahiro Miyashita, and Norihiro Hagita. SNAPCAT-3D: Calibrating networks of 3D range sensors for pedestrian tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 712–719, 2015.

[122] Jörg Röwekämper, Michael Ruhnke, Bastian Steder, Wolfram Burgard, and Gian Diego Tipaldi. Automatic extrinsic calibration of multiple laser range sensors with little overlap. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2072–2077, 2015.

[123] Jan Quenzel, Nils Papenberg, and Sven Behnke. Robust extrinsic calibration of multiple stationary laser range finders. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1332–1339, 2016.

[124] Christoph Schöller, Maximilian Schnettler, Annkathrin Krämmer, Gereon Hinz, Maida Bakovic, Müge Güzet, and Alois Knoll. Targetless rotational auto-calibration of radar and camera for intelligent transportation systems. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3934–3941, 2019.

[125] Manuel Huber, Michael Schlegel, and Gudrun Klinker. Application of time-delay estimation to mixed reality multisensor tracking. *Journal of Virtual Reality and Broadcasting*, 11(3), 2014.

[126] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

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

[128] Quoc V. Le and Andrew Y. Ng. Joint calibration of multiple sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3651–3658, 2009.

[129] René Wagner, Oliver Birbach, and Udo Frese. Rapid development of manifold-based graph optimization systems for multi-sensor calibration and SLAM. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3305–3312, 2011.

[130] Tilman Kühner and Julius Kümmerle. Extrinsic multi sensor calibration under uncertainties. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 3921–3927, 2019.

[131] Thomas Schneider, Mingyang Li, Cesar Cadena, Juan Nieto, and Roland Siegwart. Observability-aware self-calibration of visual and inertial sensors for ego-motion estimation. *IEEE Sensors Journal*, 19(10):3846–3860, 2019.

[132] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale. Online self-calibration for robotic systems. *The International Journal of Robotics Research*, 35(4):357–380, 2015.

[133] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[134] Mingyang Li, Hongsheng Yu, Xing Zheng, and Anastasios I. Mourikis. High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 409–416. IEEE, 2014.

[135] Shu Hua Tsao and Shau Shiun Jan. Observability analysis and performance evaluation of EKF-based visual-inertial odometry with online intrinsic camera parameter calibration. *IEEE Sensors Journal*, 19(7):2695–2703, 2019.

[136] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *International Journal of Robotics Research*, 36(10):1053–1072, 2017.

[137] Robert Hermann and Arthur J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, 1977.

[138] Mingyang Li and Anastasios I. Mourikis. Online temporal calibration for camera-IMU systems: Theory and algorithms. *International Journal of Robotics Research*, 33(7):947–964, 2014.

[139] Agostino Martinelli, Davide Scaramuzza, and Roland Siegwart. Automatic self-calibration of a vision system during robot motion. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 43–48, 2006.

[140] Yuanxin Wu, Meiping Wu, Xiaoping Hu, and Dewen Hu. Self-calibration for land navigation using inertial sensors and odometer: Observability analysis. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–10, 2009.

[141] Andrea Censi, Antonio Franchi, Luca Marchionni, and Giuseppe Oriolo. Simultaneous calibration of odometry and sensor parameters for mobile Robots. *IEEE Transactions on Robotics*, 29(2):475–492, 2013.

[142] Berthold K P Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629, 1987.

[143] K S Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[144] Jonathan Brookshire and Seth Teller. Automatic calibration of multiple coplanar sensors. In *Robotics: Science and Systems (RSS)*, pages 1–8, 2011.

[145] Zhan Wang and Gamini Dissanayake. Observability analysis of SLAM using fisher information matrix. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1242–1247, 2008.

[146] Andrew D. Wilson, Jarvis A. Schultz, and Todd D. Murphey. Trajectory synthesis for fisher information maximization. *IEEE Transactions on Robotics*, 30(6):1358–1370, 2014.

[147] Karol Hausman, James Preiss, Gaurav S Sukhatme, and Stephan Weiss. Observability-aware trajectory optimization for self-calibration with application to UAVs. *IEEE Robotics and Automation Letters*, 2(3):1770–1777, 2017.

[148] Keenan Albee, Monica Ekal, Rodrigo Ventura, and Richard Linares. Combining parameter identification and trajectory optimization: Real-time planning for information gain. In *arXiv preprint*, 2019.

[149] James A Preiss, Karol Hausman, Gaurav S Sukhatme, and Stephan Weiss. Simultaneous self-calibration and navigation using trajectory optimization. *International Journal of Robotics Research*, 37(13-14):1573–1594, 2018.

[150] Claude Jauffret. Observability and Fisher information matrix in nonlinear regression. *IEEE Transactions on Aerospace and Electronic Systems*, 43(2):756–759, 2007.

[151] Erich Leo Lehmann and George Casella. *Theory of point estimation*. Springer Texts in Statistics, 1998.

[152] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning Second Edition*. Springer Series in Statistics Trevor, 2008.

[153] Bradley Efron and David V. Hinkley. Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information. *Biometrika*, 65(3):457–487, 1978.

[154] Thomas J. Rothenberg. Identification in parametric models. *Econometrica*, 39(3):577–591, 1971.

[155] Gene H Golub and Charles F Van Loan. *Matrix computations*. Johns Hopkins University Press, 2013.

[156] Dan Coe. Fisher matrices and confidence ellipses: A quick-start guide and software. In *arXiv preprint*, pages 1–4, 2009.

[157] Javier Schloemann and R. Michael Buehrer. Using Fisher information matrix summary statistics to assess the value of collaborative positioning opportunities. In *IEEE Military Communications Conference (MILCOM)*, pages 1316–1321, 2013.

[158] Ken Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH Computer Graphics*, 19(3):245–254, 1985.

[159] Simon Tomažič and Igor Škrjanc. Fusion of visual odometry and inertial navigation system on a smartphone. *Computers in Industry*, 74:119–134, 2015.

[160] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2012.

[161] Zachary Taylor and Juan Nieto. Automatic markerless calibration of multi-modal sensor arrays. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4843–4850, 2015.

[162] Paul Furgale, Chi Hay Tong, Timothy D. Barfoot, and Gabe Sibley. Continuous-time batch trajectory estimation using temporal basis functions. *The International Journal of Robotics Research*, 34(14):1688–1710, 2015.

[163] Hannes Sommer, James Richard Forbes, Roland Siegwart, and Paul Furgale. Continuous-time estimation of attitude using B-splines on Lie groups. *Journal of Guidance, Control, and Dynamics*, 39(2):242–261, 2016.

[164] Alonso Patron-Perez, Steven Lovegrove, and Gabe Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *International Journal of Computer Vision*, 113(3):208–219, 2015.

[165] Steven Lovegrove, Alonso Patron-Perez, and Gabe Sibley. Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In *British Machine Vision Conference (BMVC)*, pages 1–12, 2013.

[166] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6):1425–1440, 2018.

[167] Anqi Joyce Yang, Can Cui, Ioan Andrei Bârsan, Raquel Urtasun, and Shenlong Wang. Asynchronous multi-view SLAM. *arXiv preprint*, 2021.

[168] David Droeschel and Sven Behnke. Efficient continuous-time SLAM for 3D lidar-based online mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5000–5007. IEEE, 2018.

[169] Hannes Ovrén and Per-Erik Forssén. Stochastic modeling for hysteretic bit–rock interaction of a drill string under torsional vibrations. *International Journal of Robotics Research*, 38(6):686–701, 2019.

[170] Adrian Haarbach, Tolga Birdal, and Slobodan Ilic. Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation. In *International Conference on 3D Vision (3DV)*, pages 381–389. IEEE, 2018.

[171] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[172] Chi Hay Tong, Paul Furgale, and Timothy D. Barfoot. Gaussian Process Gauss-Newton for non-parametric simultaneous localization and mapping. *International Journal of Robotics Research*, 32(5):507–525, 2013.

[173] Tim Barfoot, Chi Hay Tong, and Simo Sarkka. Batch continuous-time trajectory estimation as exactly sparse Gaussian Process regression. In *Robotics: Science and Systems*, 2014.

[174] Sean Anderson, Timothy D. Barfoot, Chi Hay Tong, and Simo Särkkä. Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression. *Autonomous Robots*, 39(3):221–238, 2015.

[175] Timothy D. Barfoot. *State estimation for robotics*. Cambridge University Press, 1 edition, 2017.

[176] Mustafa Mukadam, Xinyan Yan, and Byron Boots. Gaussian Process motion planning. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9–15, 2016.

[177] Luka Petrović, Ivan Marković, and Marija Seder. Multi-agent Gaussian Process motion planning via probabilistic inference. In *IFAC World Congress*, pages 160–165, 2018.

[178] Luka Petrović, Juraj Peršić, Marija Seder, and Ivan Marković. Stochastic optimization for trajectory planning with heteroscedastic Gaussian processes. In *European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2019.

[179] Filip Marić, Oliver Limoyo, Luka Petrović, Trevor Ablett, Ivan Petrović, and Jonathan Kelly. Fast manipulability maximization using continuous-time trajectory optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8258–8264, 2019.

[180] Luka Petrović, Juraj Peršić, Marija Seder, and Ivan Marković. Cross-entropy based stochastic optimization of robot trajectories using heteroscedastic continuous-time Gaussian processes. *Robotics and Autonomous Systems*, 133:103618, 2020.

[181] Luka Petrović, Filip Marić, Ivan Marković, and Ivan Petrović. Gaussian Processes Incremental Inference for Mobile Robots Dynamic Planning. In *IFAC World Congress*, pages 1–6, 2020.

[182] Niklas Wahlström and Emre Özkan. Extended target tracking using Gaussian Processes. *IEEE Transactions on Signal Processing*, 63(63):4165–4178, 2015.

[183] Xinyan Yan, Vadim Indelman, and Byron Boots. Incremental sparse GP regression for continuous-time trajectory estimation and mapping. *Robotics and Autonomous Systems*, 87:120–132, 2017.

[184] Timothy D. Barfoot and Paul T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[185] Joshua G. Mangelson, Maani Ghaffari, Ram Vasudevan, and Ryan M. Eustice. Characterizing the uncertainty of jointly distributed poses in the Lie algebra. *IEEE Transactions on Robotics*, pages 1–18, 2020.

[186] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

[187] Josip Ćesić, Ivan Marković, Mario Bukal, and Ivan Petrović. Extended information filter on matrix Lie groups. *Automatica*, 82:226–234, 2017.

[188] Kruno Lenac, Josip Ćesić, Ivan Marković, and Ivan Petrović. Exactly sparse delayed state filter on Lie groups for long-term pose graph SLAM. *International Journal of Robotics Research*, 37(6):585–610, 2018.

[189] Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang. Closed-form preintegration methods for graph-based visual–inertial navigation. *International Journal of Robotics Research*, 38(5):563–586, 2019.

[190] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. *arXiv preprint*, 2020.

[191] John Stillwell. *Naive Lie Theory*. Springer, 2008.

[192] Giorgio Grisetti, Tiziano Guadagnino, Irvin Aloise, Mirco Colosi, Bartolomeo Della Corte, and Dominik Schlegel. Least squares optimization: From theory to practice. *Robotics*, 9(3):1–46, 2020.

PUBLICATIONS

PUBLICATION 1

J. Peršić, I. Marković and I. Petrović. Extrinsic 6DoF calibration of 3D lidar and radar. *IEEE European Conference on Mobile Robots (ECMR)*. Paris, France, 1–6, 2017.

Extrinsic 6DoF Calibration of 3D LiDAR and Radar

Juraj Peršić, Ivan Marković, Ivan Petrović

Abstract— **Environment perception is a key component of any autonomous system and is often based on a heterogeneous set of sensors and fusion thereof, for which extrinsic sensor calibration plays fundamental role. In this paper, we tackle the problem of 3D LiDAR–radar calibration which is challenging due to low accuracy and sparse informativeness of the radar measurements. We propose a complementary calibration target design suitable for both sensors, thus enabling a simple, yet reliable calibration procedure. The calibration method is composed of correspondence registration and a two-step optimization. The first step, reprojection error based optimization, provides initial estimate of the calibration parameters, while the second step, field of view optimization, uses additional information from the radar cross section measurements and the nominal field of view to refine the parameters. In the end, results of the experiments validated the proposed method and demonstrated how the two steps combined provide an improved estimate of extrinsic calibration parameters.**

I. Introduction

Robust environment perception is one of the essential tasks which an autonomous mobile robot or vehicle has to accomplish. To achieve this goal, various sensors such as cameras, radars, LiDAR-s, and inertial navigation units are used and information thereof is often fused. A fundamental step in the fusion process is sensor calibration, both intrinsic and extrinsic. Former provides internal parameters of each sensor, while latter provides relative transformation from one sensor coordinate frame to the other. The calibration can tackle both parameter groups at the same time or assume that sensors are already intrinsically calibrated and proceed with the extrinsic calibration, which is the approach we take in the present paper.

Solving the extrinsic calibration problem requires finding correspondences in the data acquired by intrinsically calibrated sensors, which can be challenging since different sensors can measure different physical quantities. The calibration approaches can be target-based or targetless. In the case of target-based calibration, correspondences originate from a specially designed target, while targetless methods utilize environment features perceived by both sensors. Former has the advantage of the freedom of design which maximizes the chance of both sensors perceiving the calibration target, but requires the development of such a target and execution of an appropriate offline calibration procedure. The latter has the advantage of using the environment itself as the calibration target and can operate online by registering structural correspondences in the environment, but requires both sensors to be able to extract the same environment features. For example, calibration of a 3D-LiDAR and a camera can be based on line features detected as intensity edges in the image and depth discontinuities in the point cloud [1]. In addition, registration of structural correspondences can be avoided by odometry-based methods, which use the system's motion estimated by individual sensors to calibrate them [2], [3]. However, for all practical means and purposes, the targetless methods are hardly feasible due to limited resolution of current automotive radar systems, as the radar is virtually unable to infer the structure of the detected object and extract features such as lines or corners. Therefore, we focus our research on target-based methods.

Target-based 3D LiDAR calibration commonly uses flat rectangles which are easily detected and localized in the point cloud. For example, extensive research exists on 3D LiDAR-camera calibration with a planar surface covered by a chequerboard [4]–[7] or a set of QR codes [8], [9]. Extrinsic calibration of a 2D LiDAR-camera pair was also calibrated with the same target [10], while improvements were made by extracting centerline and edge features of a V-shaped planar target [11]. Furthermore, an interesting target adaptation to the working principle of different sensors was presented in [12], where the authors proposed a method for extrinsic calibration of a 3D LiDAR and a thermal camera by expanding a planar chequerboard surface with a grid consisting of light bulbs. Concerning automotive radars, common operating frequencies (24 GHz and 77 GHz) result with reliable detections of conductive objects, such as plates, cylinders, and corner reflectors, which are then used in calibration methods [13]. In [14] authors used a metal panel as the target for radar-camera calibration. They assume that all radar measurements originate from a single ground plane, thereby neglecting the 3D nature of the problem. The calibration is found by optimizing homography transformation between the ground and image plane. Contrary to [14], in [15] authors take into account the 3D nature of the problem. Therein, they manually search for detection intensity maximums by moving a corner reflector within the field of view (FoV). They assume that detections lie on the radar plane (zero elevation plane in the radar coordinate frame). Using these points a homography transformation is optimized between the radar and the camera. The drawback of this method is that the maximum intensity search is prone to errors, since the return intensity depends on a number of factors, e.g., target orientation and radar antenna radiation pattern which is usually designed to be as constant as possible in the FoV. In [16] radar performance is evaluated using a 2D LiDAR as a ground truth with a target composed of radar tube reflector

Authors are with University of Zagreb Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Unska 3, HR-10000, Zagreb, Croatia, juraj.persic@fer.hr, ivan.markovic@fer.hr, ivan.petrovic@fer.hr

and a square cardboard. The cardboard is practically invisible to the radar, while enabling better detection and localization in the LiDAR point cloud. These complementary properties were taken as an inspiration for our target design.

While the above described radar calibration methods provide sufficiently good results for the targeted applications, they lack the possibility to fully assess the placement of the radar with respect to other sensors.Therefore, we propose a novel method which utilizes a 6 degrees of freedom (DoF) extrinsic calibration of a 3D LiDAR-radar pair. The proposed method involves special calibration target design, correspondence registration, and two-step optimization. The first step is based on reprojection error optimization, while the second step uses additional information from the radar cross section (RCS), a measure of detection intensity. RCS distribution across the radar's FoV is used to refine a subset of calibration parameters that were noticed to have higher uncertainty.

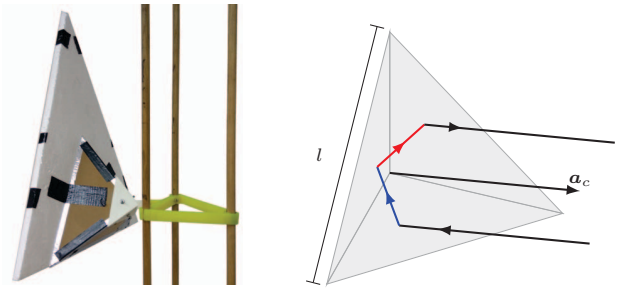
The paper is organized as follows. Section II elaborates the calibration method including calibration target design II-A and data correspondence registration II-B. Section III explains two steps of optimization: reprojection error optimization III-A and FoV optimization III-B. Section IV-A provides details on the setup of experiment conducted to test the method, while the results are given in IV-B. We give final remarks and propose future work in section V.

II. Extrinsic Radar-LiDAR Calibration Method

The proposed method is based on observing the calibration target placed at a range of different heights, both within and outside of the nominal radar FoV. It requires the 3D LiDAR's FoV to exceed the radar's vertical FoV, which is the case in most applications. In addition, due to the problems associated with radars such as ghost measurements from multipath propagation, low angular resolution etc., data collection has to be performed outdoor at a set of ranges $(2 - 10\,\mathrm{m})$ with enough clear space around the target.

A. Calibration Target Design

Properties of a well-designed target are (i) ease of detection and (ii) high localization accuracy for both sensors. In terms of the radar, a target with a high RCS provides good detection rates. Formally, RCS of an object is defined as the area of a perfectly conducting sphere whose echo strength would be equal to the object strength [13]. Consequently, it is a function of object size, material, shape and orientation. While any metal will suffice for the material, choosing other properties is not trivial. Radars typically estimate range and angle of an object as a centroid in the echo signal. Therefore, in order to accurately localize the source of detection, the target should be as small as possible, but which implies a small RCS. Thus, a compromise between the target size and a high enough RCS has to be considered. Radar reflectors, objects that are highly visible to radars, are used not only in intrinsic calibration, but also as marine safety equipment resulting in numerous designs. Given the previous discussion, we assert that one of these designs can be considered as a good compromise and we chose the triangular trihedral corner reflector which consists of three orthogonal flat metal triangles.



(a) Calibration Target (b) Corner reflector

Fig. 1: Constructed calibration target and the illustration of the working principle of the triangular trihedral corner reflector

The constructed radar calibration target and an illustration of the working principle is shown in Fig. 1a. It has an interesting property that any ray reflected from all three sides is returned in the same direction as illustrated in Fig. 1b. The reason behind this is that normals of the three sides form an orthonormal basis. Namely, reflection causes direction reverse of incident ray's component parallel to the surface normal, while the component parallel to the surface tangent plane remains the same. After three reflections, which form an orthonormal basis, the ray's direction is reversed. Due to this property, regardless of the incident angle, many rays are returned to their source, i.e., the radar. Unlike a single flat plate, which has a high RCS but is highly sensitive to orientation changes, trihedral corner reflector provides a high and stable RCS. When the axis of the corner reflector, $\mathbf{a_c}$, points directly to the radar, it reaches its maximum RCS value:

$$\sigma_c = \frac{\pi l^4}{3\lambda^2}, \tag{1}$$

where l is a hypotenuse length of a corner reflector's side and λ is radar's operating wavelength.

Analytical description of the reflector RCS as a function of the orientation is nontrivial. However, from experiments presented in [13], it can be seen that orientation changes of $\pm 20^\circ$ result in a slight decrease of RCS, which can be approximated as a constant, while $\pm 40^\circ$ causes a decrease of $-3\mathrm{dBm}^2$. Furthermore, authors in [17] show that all the rays which go through multiple reflections travel the same length as the ray which is reflected directly from the corner centre. This results in a high localization accuracy.

Corner reflector is visible to the LiDAR, but is difficult to accurately localize it at greater distances due to its small size and complex shape. This problem is solved by placing a flat styrofoam triangle board in front of the reflector. Styrofoam is made of approximately 98% air resulting with low permittivity (around 1.10) and nonconductiveness. These properties make it virtually invisible to the radar, but still visible to the LiDAR. However, instead of a common rectangular shape, we choose a triangular shape with which we can solve localization ambiguity issues caused by finite LiDAR

resolution. Namely, LiDAR azimuth resolution is commonly larger than the elevation resolution, which results with the 'slicing' effect of an object; thus, translating the rectangle along the vertical axis would yield identical measurements until it becomes visible to the next LiDAR layer (which is not the case for the triangle shape). This effect has a stronger impact on localization at greater distances which are required by our method.

Finally, target stand should be able to hold the target at a range of different heights (0–2 m). Additionally, it must have a low RCS not to interfere with the target detection and localization. We propose a stand made of three thin wooden rods which are fixed to a ground wooden plane and connected with a plastic bridge (Fig. 1a). Target attached to the bridge can be slided and tilted to adjust its height and orientation.

B. Correspondence Registration

Correspondence registration in the data starts with the detection of the triangle in the point cloud. The initial step is to segment plane candidates from which edge points are extracted. Afterwards, we try to fit these points to the triangle model. Levenberg–Marquardt (LM) algorithm optimizes the pose of the triangle by minimizing the distance from edge points to the border of the triangle model. A final threshold is defined based on which we accept or discard the estimate. Position of the corner reflector ${}^{l}\boldsymbol{x}_l$[1] origin is calculated based on the triangle pose estimate and the known target configuration.

Radar data of interest is a list of detected objects described by the detection angle ${}^{r}\phi_{r,i}$, range ${}^{r}r_{r,i}$ and RCS $\sigma_{r,i}$. The i-th object from the list is described by the vector ${}^{r}\boldsymbol{m}_i = [{}^{r}\phi_{r,i}\ {}^{r}r_{r,i}\ {}^{r}\sigma_{r,i}]$ in the radar coordinate frame, $\mathcal{F}_r : ({}^{r}x, {}^{r}y, {}^{r}z)$. The only structural property of detected objects is contained within the RCS, which is influenced by many other factors; hence, it is impossible to classify a detection as the corner reflector based solely on the radar measurements. To find the matching object, a rough initial calibration is required, e.g., with a measurement tape, which is used to transform the estimated corner position from the LiDAR coordinate frame, $\mathcal{F}_l : ({}^{l}x, {}^{l}y, {}^{l}z)$, into the $\mathcal{F}_r$, and eliminate all other objects that fall outside of a predefined threshold. The correspondence is accepted only if a single object is left.

The radar correspondence groups are obtained as follows. The target is observed at rest for a short period while the registered correspondences fill a correspondence group with pairs of vectors ${}^{r}\boldsymbol{m}_i$ and ${}^{l}\boldsymbol{x}_l$. Variances of the radar data $({}^{r}\phi_{r,i}, {}^{r}r_{r,i}, {}^{r}\sigma_{r,i})$ within the group are used to determine the stability of the target. If any of the variances surpasses a preset threshold, the correspondence is discarded, since it is likely that the target detection was obstructed. Otherwise, the values are averaged. In addition, we create unregistered groups where radar detections are missing. These groups are used in the second optimization step where we refine the FoV. Hereafter, we will refer to the mean values of the groups as radar and LiDAR measurements.

[1]In the article, we use left superscript r and l to denote that the value belongs to the $\mathcal{F}_r$ and $\mathcal{F}_l$, respectively

III. Two-Step Optimization

A. Reprojection Error Optimization

Once the paired measurements are found, alignment of sensor coordinate frames is performed. To ensure that the optimization is performed on the radar measurements originating from the calibration target, we perform RCS threshold filtering. We choose the threshold ζ_{RCS} close to the σ_c so that we encompass as many strong and reliable radar measurements while leaving out the possible outliers.

The optimization parameter vector includes the translation and rotation part, i.e., $\boldsymbol{c}_r = [{}^{r}\boldsymbol{p}_l\ \boldsymbol{\Theta}]$. For translation, we choose position of the LiDAR in the $\mathcal{F}_r$, ${}^{r}\boldsymbol{p}_l = [{}^{r}p_{x,l}\ {}^{r}p_{y,l}\ {}^{r}p_{z,l}]^T$. For rotation, we choose Euler angles representation $\boldsymbol{\Theta} = [\theta_z\ \theta_y\ \theta_x]$ where rotation from $\mathcal{F}_r$ to $\mathcal{F}_l$ is given by:

$$ {}^{l}_{r}R(\boldsymbol{\Theta}) = {}^{l}_{2}R_x(\theta_x){}^{2}_{1}R_y(\theta_y){}^{1}_{r}R_z(\theta_z). \tag{2} $$

Figure 2 illustrates the calculation of the reprojection error for the i-th paired measurement. As discussed previously, radar provides measurements in spherical coordinates lacking elevation ${}^{r}\boldsymbol{s}_{r,i} = [{}^{r}r_{r,i}\ {}^{r}\phi_{r,i}\ \sim]$, i.e., it provides an arc ${}^{r}a_{r,i}$ upon which the object potentially resides. On the other hand, LiDAR provides a point in Euclidean coordinates ${}^{l}\boldsymbol{x}_{l,i}$. Using the current transformation estimate, LiDAR measurement ${}^{l}\boldsymbol{x}_{l,i}$ is transformed into the radar coordinate frame:

$$ {}^{r}\boldsymbol{x}_{l,i}(\boldsymbol{c}_r) = {}^{l}_{r}R^T(\boldsymbol{\Theta}) \cdot {}^{l}\boldsymbol{x}_{l,i} + {}^{r}\boldsymbol{p}_l, \tag{3} $$

and then ${}^{r}\boldsymbol{x}_{l,i}$ is converted to spherical coordinates ${}^{r}\boldsymbol{s}_{l,i} = [{}^{r}r_{l,i}\ {}^{r}\phi_{l,i}\ {}^{r}\psi_{l,i}]$. By neglecting the elevation angle ${}^{r}\psi_{l,i}$, we obtain the arc ${}^{r}a_{l,i}$ upon which LiDAR measurement resides and can be compared to the radar's. Reprojection error $\epsilon_{r,i}$ is then defined as the Euclidean distance of points on the arc for which ${}^{r}\psi_{r,i} = {}^{r}\psi_{l,i} = 0°$:

$$ \epsilon_{r,i}(\boldsymbol{c}_r) = \left\| \begin{bmatrix} {}^{r}r_{r,i}\cos({}^{r}\phi_{r,i}) \\ {}^{r}r_{r,i}\sin({}^{r}\phi_{r,i}) \end{bmatrix} - \begin{bmatrix} {}^{r}r_{l,i}\cos({}^{r}\phi_{l,i}) \\ {}^{r}r_{l,i}\sin({}^{r}\phi_{l,i}) \end{bmatrix} \right\|. \tag{4} $$

Using the LM algorithm, we obtain the estimate of the calibration parameters $\hat{\boldsymbol{c}}_r$ by minimizing the sum of squared reprojection errors from N measurements:

$$ \hat{\boldsymbol{c}}_{\mathbf{r}} = \arg\min_{\boldsymbol{c}_r} \left(\sum_{i=1}^{N} \epsilon_{r,i}^2(\boldsymbol{c}_r) \right). \tag{5} $$

Optimization of described reprojection error yields unequal estimation uncertainty among the calibration parameters. Namely, translation in the radar plane and rotation around it's normal causes significant changes in the radar measurements. Therefore, parameters ${}^{r}p_{x,l}$,${}^{r}p_{y,l}$ and θ_z can be properly estimated. In contrast, the change in the remaining parameters ${}^{r}p_{z,l}$,θ_y and θ_x causes smaller changes in the radar measurements, e.g. translation of radar along ${}^{r}z$ introduces only a small change in the range measurement. Therefore, these parameters are refined in the second step.

Due to the filtering in the correspondence registration, not many outliers are present in the data. The remaining outliers are removed from the dataset by inspection of the reprojection error after the optimization. Measurements

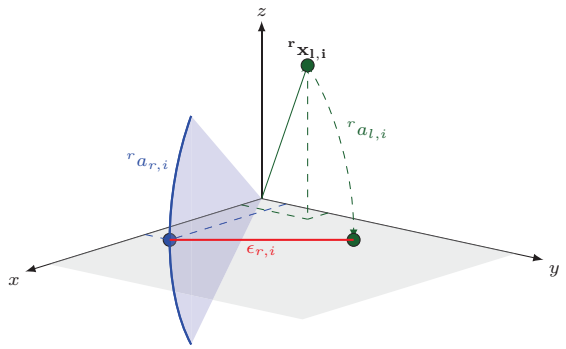
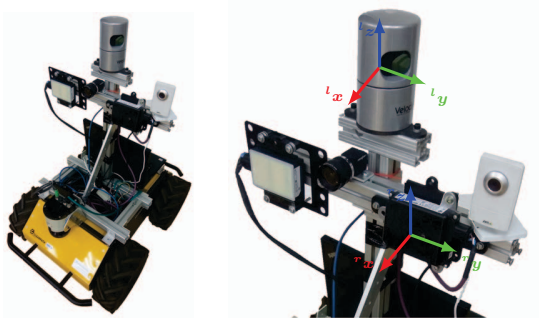


Fig. 2: Illustration of reprojection error calculation. Green: LiDAR's measurement; blue: radar's; red: reprojection error.



(a) Mobile robot (b) Sensor placement

Fig. 3: Mobile robot and sensors used in the experiment.

that surpass the radar's accuracy are excluded from the dataset and optimization is performed again on the remaining measurements.

B. FoV optimization

To refine the parameters with higher uncertainty we propose a second optimization step which uses additional information from RCS. We try to fit the radar's nominal FoV in the LiDAR data by encompassing as many measurements with high RCS as possible. Definition of RCS is such that it is independent of the radar's radiation. However, radar estimates the object RCS based on the intensity of the echo which is dependent on the radiated energy. Intrinsic calibration of a radar ensures that RCS is correctly estimated only within the nominal FoV where it is fairly constant. As the object leaves the nominal FoV, less energy is radiated in its direction, which then results in decrease of RCS until the object becomes undetectable. This effect is used to estimate the pose of the nominal FoV based on the RCS distribution across the LiDAR's data.

Vertical FoV of width $2\psi_f$ is defined with two planes that go through the origin of $\mathcal{F}_r$, $\mathcal{P}_U$ and $\mathcal{P}_D$, with elevation angles $\pm\psi_f$. We propose an optimization in which we position radar's nominal FoV, so that as many as possible strong reflections fall within it, while leaving the weak ones out. The optimization parameter vector consist of a subset of transformation parameters and an RCS threshold, $\boldsymbol{c}_f = [{}^r p_{z,l}\ \theta_y\ \theta_x\ \zeta_{RCS}]$, whereas other parameters are kept fixed.

After transforming a LiDAR measurement ${}^l\boldsymbol{x}_{l,i}$ to $\mathcal{F}_r$, the FoV error of i-th measurement $\epsilon_{f,i}$ is defined as:

$$\epsilon_{f,i}(\boldsymbol{c}_f) = \begin{cases} 0 & \text{if inside FoV and } \sigma_i > \zeta_{RCS} \\ d & \text{if inside FoV and } \sigma_i > \zeta_{RCS} \\ 0 & \text{if outside FoV and } \sigma_i < \zeta_{RCS} \\ d & \text{if outside FoV and } \sigma_i < \zeta_{RCS}, \end{cases} \tag{6}$$

where

$$d = \min\{\text{dist}(\mathcal{P}_U, {}^r\boldsymbol{x}_{l,i}), \text{dist}(\mathcal{P}_D, {}^r\boldsymbol{x}_{l,i})\}. \tag{7}$$

Error is greater than zero only if the LiDAR measurement falls inside the FoV when it should not according to the threshold, and vice versa. Function $\text{dist}(\mathcal{P}, x)$ is defined as an unsigned distance from plane $\mathcal{P}$ to point x.

An estimate of calibration parameters is obtained by minimizing the following cost function:

$$\hat{\boldsymbol{c}}_\mathbf{f} = \arg\min_{\boldsymbol{c}_f} \left(\sum_{i=1}^{N} \epsilon_{f,i}^2(\boldsymbol{c}_f) \right). \tag{8}$$

Dependence of the cost function is discrete with respect to the RCS threshold, since change of the threshold does not affect the cost function until at least one measurement falls in or out of the FoV. This results in many local minima and the interior points method was used for optimization, since it was found to be able to converge in majority of analysed cases.

IV. Experiment

A. Experiment Setup

An outdoor experiment was conducted to test the proposed method. A mobile robot Husky UGV, shown in Fig. 3, was equipped with a Velodyne HDL-32E 3D LiDAR and two short range radars from different manufacturers, namely the Continental SRR 20X and Delphi SRR2.

Commercially available radars are sensors which provide high level information in the form of detected object list. Raw data, i.e., the return echo, is processed by proprietary signal processing techniques and is unavailable to the user. However, from the experiments conducted with both radars, we noticed that they follow the behaviour as expected from our calibration method. The only noticed difference is that the target stand without the target was completely invisible to Continental, while the Delphi was able to detect it at closer ranges (${}^r r_{r,i} < 5$ m). This effect was present because the Delphi radar accepts detections with lower RCS. However, this did not present an issue, because the stand has a significantly lower RCS than the target and it was easily filtered out. Since the purpose of the experiment is evaluation of the method and not radar performance, in the sequel we only present results for the Continental radar.

Continental radar technical data of interest is given in Table I. Based on the analysis of the reprojection error, radar measurements outside of the azimuth angle range of $\pm 45°$ were excluded from the reprojection error optimization, because they exhibited significantly higher reprojection errors

TABLE I: Continental SRR 20X specifications

Continental SRR 20X	Value
HFoV × VFoV	150° × 12°
Range Accuracy	0.2m
Azimut Accuracy @ HFoV	±2°@±20°; ±4°@±60°; ±5°@±°75

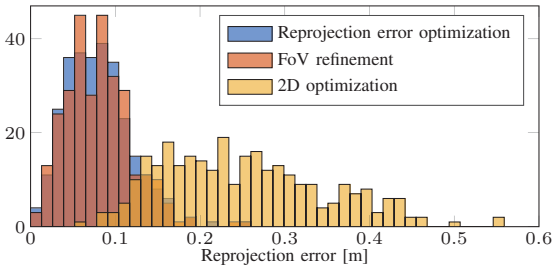


Fig. 4: Histogram of reprojection errors for the two steps of the calibration and the 2D calibration

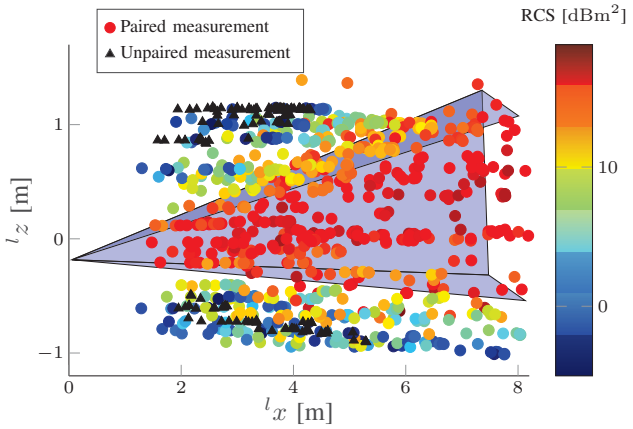


Fig. 5: RCS distribution across LiDAR 3D data and placement of the radar's FoV.

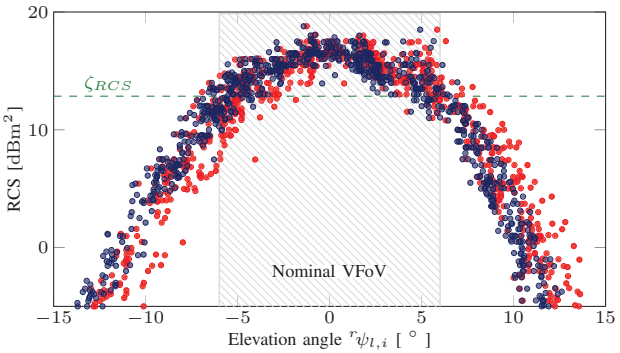


Fig. 6: RCS distribution across radar's VFoV. Red: reprojection error optimization; blue: FoV optimization.

than those inside the range. Considering FoV optimization, we noticed that outside of the azimuth angle range ±60° radar detections were occasionally missing. Therefore, they were excluded from the FoV optimization.

The calibration target was composed of a corner reflector with side length $l = 0.32\,\text{m}$ with a maximum RCS of $\sigma_c = 18.75\,\text{dBm}^2$. Based on vertical resolution of the Velodyne HDL-32E LiDAR (1.33°) we used styrofoam triangle of height $h = 0.65$ m. It ensured extraction of at least two lines from the target, which is a prerequisite to unambiguously determine the pose. Data acquisition was done by driving a robot in the area up to 10 m of range with target placed at 17 different heights ranging from ground up to 2 m height. In total, 880 registered radar-LiDAR measurements were collected, together with 150 LiDAR measurements unregistered by the radar.

B. Results

To assess the quality of calibration results we conducted four experiments. First, we examined the distribution of the reprojection error after both optimization steps and compared it to a 2D optimization, which minimizes reprojection error by optimizing only the calibration parameters with lower uncertainty, i.e., translation parameters $^rp_{x,l}$ and $^rp_{y,l}$, and rotation θ_z. Secondly, we inspect FoV placement with respect to the distribution of RCS over the LiDAR's data. Afterwards, we examine the correlation between RCS and the elevation angle. Lastly, we run Monte Carlo simulations by randomly subsampling the dataset to examine reliability of the estimated parameters and potential overfitting of data.

Parameters estimated by reprojection error optimization are $\hat{\boldsymbol{c}}_r = [-0.047, -0.132, 0.079\text{m}; -2.07, 3.58, -0.02°]$, while FoV optimization estimates $\hat{\boldsymbol{c}}_f = [0.191, \text{m}; 4.19, -0.84°; 12.85\text{dBm}^2]$. Carefully measured translation by hand between the sensors $^r\tilde{\boldsymbol{p}}_l = [-0.08, -0.14, 0.18]^T$ m is given as a reference.

Figure 4 shows distribution of the reprojection error and is composed of three histograms, where we can see how the reprojection error of both steps of calibration is compared to the case of 2D calibration. We notice that neglecting the 3D nature of the problem causes higher mean and greater variance of the reprojection error which implies poor calibration. Furthermore, the FoV optimization is bound to degrade the overall reprojection error because it is not a part of the optimization criterium. However, resemblance between the distributions after the first and the second optimization steps implies low degradation of reprojection error.

In Fig. 5, distribution of the RCS across LiDAR's data is shown. LiDAR's measurements are color-coded with the RCS of the paired radar measurement, accompanied with the black-dyed markers which indicate the lack of registered radar measurements. We can see that within the nominal FoV, target produces a strong, fairly constant reflections. As the elevation angle of the target leaves the radars FoV, the RCS decreases until the point where it is no longer detectable.

To examine the effect of decrease in the target's RCS as a function of the elevation angle after both optimizations, we use Fig. 6. It shows elevation $^r\psi_{l,i}$ of each LiDAR measurement transformed into the $\mathcal{F}_r$ and RCS of the paired radar measurement. In the ideal case, i.e. if the transformation was correct and the axis of corner reflector always pointed directly to the radar, the data would lay on the curve which describes radar's radiation pattern in respect to the elevation angle. The dispersion from the curve is present in the both steps due to the imperfect directivity of the target in the

TABLE II: Monte Carlo Analysis Results

	Reprojection Error Optimization	FoV optimization
${}^{r}p_{x,l}$	$\mathcal{N}(-0.047\text{m}, 1.53 \times 10^{-5})$	
${}^{r}p_{y,l}$	$\mathcal{N}(-0.132\text{m}, 6.12 \times 10^{-5})$	
${}^{r}p_{z,l}$	$\mathcal{N}(0.078\text{m}, 2.53 \times 10^{-3})$	$\mathcal{N}(0.174\text{m}, 9.10 \times 10^{-4})$
θ_z	$\mathcal{N}(-2.08°, 1.12 \times 10^{-2})$	
θ_y	$\mathcal{N}(3.59°, 9.50 \times 10^{-1})$	$\mathcal{N}(4.00°, 9.93 \times 10^{-2})$
θ_x	$\mathcal{N}(-0.03°, 8.08 \times 10^{-1})$	$\mathcal{N}(-0.93°, 1.44 \times 10^{-1})$

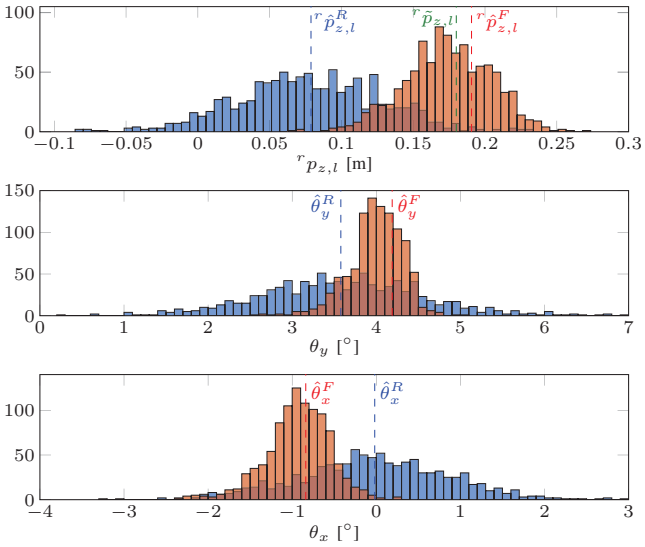


Fig. 7: Monte Carlo Analysis histograms. Red: calibration after reprojection error optimization; blue: with FoV optimization

measurements. In addition, we notice a higher dispersion using only reprojection error optimization which indicates miscalibration.

Lastly, Monte Carlo analysis is done by randomly subsampling our dataset to half of the original size and performing 1000 runs of optimization on different subsampled datasets. The results follow a Gaussian distribution whose estimated parameters are given by the Table II. As expected, distributions of parameters ${}^{r}p_{x,l}$, ${}^{r}p_{y,l}$ and θ_z obtained by the reprojection error optimization have a significantly lower variance than the rest. Figure 7 illustrates how the FoV optimization refines parameters ${}^{r}p_{z,l}$, θ_y and θ_x. We can see overall decrease in variance, as well as the shift in the mean. Estimation of parameter ${}^{r}p_{z,l}$ using reprojection error optimization is clearly further away from the measured value, unlike the FoV optimization's estimate.

V. Conclusion

In this paper we have proposed an extrinsic calibration method for a 3D-LiDAR-radar pair. A calibration target was designed in a way which enabled both sensors to detect and localize the target within their operating principles. The extrinsic calibration was found by a two-step optimization: (i) reprojection error optimization, which was the followed by (ii) FoV optimization which used additional information from RCS to refine the estimate of the calibration parameters. Results of the experiments validated the proposed method and demonstrated how the two steps combined provide an improved estimate of extrinsic calibration parameters. In the future work, we plan to include a camera in the extrinsic calibration. In addition, we plan to improve the results of the calibration by introducing the sensor uncertainty models as radars typically have variable accuracy across the FoV.

Acknowledgment

This work has been supported from the Unity Through Knowledge Fund (no. 24/15) under the project Cooperative Cloud based Simultaneous Localization and Mapping in Dynamic Environments (cloudSLAM). This research has also been carried out within the activities of the Centre of Research Excellence for Data Science and Cooperative Systems supported by the Ministry of Science and Education of the Republic of Croatia.

References

[1] J. Levinson and S. Thrun, "Automatic Online Calibration of Cameras and Lasers," in *Robotics: Science and Systems (RSS)*, 2013.

[2] S. Schneider, T. Luettel, and H. J. Wuensche, "Odometry-based online extrinsic sensor calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1287–1292.

[3] N. Keivan and G. Sibley, "Online SLAM with any-time self-calibration and automatic change detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5775–5782.

[4] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3D laser scanner and an omnidirectional camera," in *IFAC Symposium on Intelligent Autonomous Vehicles*, 2010, pp. 336–341.

[5] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot." in *IEEE Conference on Robotics and Automation (ICRA)*, 2012, pp. 3936–3943.

[6] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of RGB Camera With Velodyne LiDAR," *WSCG 2014 Communication Papers*, pp. 135–144, 2014.

[7] L. Zhou and Z. Deng, "Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 642–648, 2012.

[8] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," *Int. Journal of Robotics Research*, vol. 31, no. 4, pp. 452–467, 2012.

[9] J. L. Owens, P. R. Osteen, and K. Daniilidis, "MSG-cal: Multi-sensor graph-based calibration," in *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, 2015, pp. 3660–3667.

[10] Q. Z. Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 2301–2306.

[11] K. Kwak, D. F. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning lidar and a camera," in *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, 2011, pp. 3283–3289.

[12] D. Borrmann, H. Afzal, J. Elseberg, and A. Nüchter, "Mutual calibration for 3D thermal mapping," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 605–610, 2012.

[13] E. F. Knott, *Radar Cross Section Measurements*. ITP Van Nostrand Reinhold, 1993.

[14] T. Wang, N. Zheng, J. Xin, and Z. Ma, "Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications," *Sensors*, vol. 11, no. 9, pp. 8992–9008, 2011.

[15] S. Sugimoto, H. Tateda, H. Takahashi, and M. Okutomi, "Obstacle detection using millimeter-wave radar and its visualization on image sequence," in *International Conference on Pattern Recognition (ICPR)*, 2004, pp. 342–345.

[16] L. Stanislas and T. Peynot, "Characterisation of the Delphi Electronically Scanning Radar for Robotics Applications," in *Australasian Conference on Robotics and Automation (ARAA)*, 2015.

[17] C. G. Stephanis and D. E. Mourmouras, "Trihedral rectangular ultrasonic reflector for distance measurements," *NDT&E international*, vol. 28, no. 2, pp. 95–96, 1995.

PUBLICATION 2

J. Peršić, I. Marković and I. Petrović. Extrinsic 6DoF calibration of a radar – LiDAR – camera system enhanced by radar cross section estimates evaluation. *Robotics and Autonomous Systems*, 114:217–230, 2019.

Robotics and Autonomous Systems 114 (2019) 217–230

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Extrinsic 6DoF calibration of a radar–LiDAR–camera system enhanced by radar cross section estimates evaluation

Juraj Peršić *, Ivan Marković, Ivan Petrović

University of Zagreb Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Laboratory for Autonomous Systems and Mobile Robotics, Unska 3, HR-10000, Zagreb, Croatia

HIGHLIGHTS

- Extrinsic radar–camera–LiDAR calibration estimated accurately in all 6DoF.
- Radar's missing elevation angle compensated with radar cross section measurements.
- Method is suitable for radar vertical misalignment detection.
- Identifiability analysis confirms chosen transform parametrization.
- Identifiability analysis provides minimal requirements on the dataset.

ARTICLE INFO

Article history:
Available online 6 December 2018

Keywords:
Sensor calibration
Radar
LiDAR
Camera
Radar cross section

ABSTRACT

Autonomous navigation of mobile robots is often based on information from a variety of heterogeneous sensors; hence, extrinsic sensor calibration is a fundamental step in the fusion of such information. In this paper, we address the problem of extrinsic calibration of a radar–LiDAR–camera sensor system. This problem is primarily challenging due to sparse informativeness of radar measurements. Namely, radars cannot extract rich structural information about the environment, while their lack of elevation resolution, that is nevertheless accompanied by substantial elevation field of view, introduces uncertainty in the origin of the measurements. We propose a novel calibration method which involves a special target design and two-step optimization procedure to solve the aforementioned challenges. First step of the optimization is minimization of a reprojection error based on an introduced point–circle geometric constraint. Since the first step is not able to provide reliable estimates of all the six extrinsic parameters, we introduce a second step to refine the subset of parameters with high uncertainty. We exploit a pattern discovered in the radar cross section estimation that is correlated to the missing elevation angle. Additionally, we carry out identifiability analysis based on the Fisher Information Matrix to show minimal requirements on the dataset and to verify the method through simulations. We test the calibration method on a variety of sensor configurations and address the problem of radar vertical misalignment. In the end, we show via extensive experiment analysis that the proposed method is able to reliably estimate all the six parameters of the extrinsic calibration.

1. Introduction

Autonomous systems navigate through the environment based on the information they gather from sensors. They have to solve many task such as simultaneous localization and mapping, detection and tracking of moving objects, etc., based on the available information from a variety of sensors. Commonly used proprioceptive sensors in robotics can include global positioning system, inertial measurement units, and wheel encoders, while extroceptive sensors include LiDARs, cameras, sonars, and radars. Appropriateness of a sensor is dependent on the application, because these sensors utilize different physical phenomena, leading to different sets of advantages and disadvantages. Therefore, to achieve a robust, full-stack autonomy, information from the aforementioned sensors is often fused.

The fundamental step in sensor fusion is sensor calibration, commonly divided to intrinsic and extrinsic calibration. The former provides internal parameters of an individual sensor related to its working principle, while the latter represent spatial displacement between a pair of sensors. The calibration can tackle both parameter groups at the same time or assume that sensors are already intrinsically calibrated and proceed with the extrinsic calibration.

* Corresponding author.
E-mail addresses: juraj.persic@fer.hr (J. Peršić), ivan.markovic@fer.hr (I. Marković), ivan.petrovic@fer.hr (I. Petrović).

https://doi.org/10.1016/j.robot.2018.11.023
0921-8890/

On the one hand, methods for finding intrinsic parameters do not share much similarities for different types of sensors since they are related to the working principle of the sensor. On the other hand, parametrization of extrinsic calibration, i.e., homogeneous transform, can always be expressed in the same manner, regardless of the sensors involved in it. Nevertheless, solving the extrinsic calibration requires finding correspondences in the data acquired by the sensors which can be challenging since different types of sensors measure different physical quantities. The calibration approaches can be target-based or targetless. In the case of target-based calibration, correspondences originate from a specially designed target, while targetless methods utilize environment features perceived by both sensors. Registration of structural correspondences can be avoided by motion-based methods, which leverage motion estimated by individual sensors for calibration.

Cameras and LiDARs are rich sources of information, commonly used in robotics, which often require precise calibration. Therefore, extensive research has been devoted to calibration of these sensors within all aforementioned calibration approaches. Target-based camera calibration approaches, based on pioneering work [1,2], typically involve planar targets with known patterns such as checkerboard [3] or a grids of circles [4]. Novel calibration target is presented in [5] where authors use a noise-like pattern with many features of varying scales. It is suitable for both intrinsic and extrinsic calibration of multiple cameras with no or little field of view (FoV) overlap. LiDAR calibration also uses flat surfaces as calibration targets. For instance, intrinsic calibration of LiDARs is achieved by placing the LiDAR inside a box [6] or by observing planar wall [7], while extrinsic calibration of multiple 2D LiDARs was found by the aid of a corner structure [8]. Extrinsic target-based calibration between LiDARs and cameras has also received significant research attention, while the common targets are planes covered with a pattern suited for camera detection. Widely adopted and extended method presented in [9] introduced point–plane geometric constraint initially designed for 2D LiDAR–camera calibration. Proposed approach was also applied in the calibration of a 3D LiDAR and a camera [10]. Further improvements were made by decoupling rotation from translation in the optimization procedure [11]. To reduce the labour requirements, authors in [12] extended the method with global correspondence registration which allows for multiple plane observations in a single shot. The same constraint was used in [13] where instead of checkerboard pattern, AprilTag fiducial markers were used [14]. Additionally, they extended the extrinsic calibration with estimation of intrinsic LiDAR parameters. AprilTag markers and the same geometric constraint were also used in [15] as a part of multi-sensor graph based calibration. Besides commonly used point–plane constraint, 3D LiDAR-camera pair was calibrated based on the point–point correspondences. In [16] authors used a target with circular holes for localization, while in [17] authors extracted centreline and edge features of a V-shaped planar target to improve 2D LiDAR-camera calibration.

Radars are frequently used in automotive applications for detection and tracking of multiple objects due to their low price and robustness. Since radars cannot provide rich information about the detections, automotive systems often fuse radars with cameras [18,19] or LiDARs [20,21] to perform advanced tasks, e.g., object classification [22,23]. Although sensor fusion requires precise calibration, extrinsic radar calibration has not gained much research attention. Existing calibration methods are all target-based since, for all practical means and purposes, the targetless methods are hardly feasible due to limited resolution of current automotive radar systems, as the radar is virtually unable to infer the structure of the detected objects and extract features such as lines or corners. Current radars have no elevation resolution while the information about the detected objects they provide contains range, azimuth angle, radar cross section (RCS) and range-rate based on the Doppler effect. Although having no elevation resolution, radars have substantial elevation FoV which makes the extrinsic calibration challenging due to the uncertainty in the origin of the measurements. Concerning automotive radars, common operating frequencies (24 GHz and 77 GHz) result with reliable detections of conductive objects, such as plates, cylinders and corner retroreflectors, which are then used in intrinsic and extrinsic calibration methods [24]. In [25] authors used a metal plate as the target for radar – camera calibration assuming that all radar measurements originate from a single ground plane, thereby neglecting the 3D nature of the problem. The calibration is then found by optimizing a homography transformation between the ground and image plane. Later, a similar approach was adopted by using thin metal poles as calibration targets [18]. Contrary to previous examples, 3D nature of the problem was taken into account by moving a corner retroreflector within the FoV and manually searching for detection intensity maximums [26]. Authors assumed that detections lie on the radar plane (zero elevation plane in the radar coordinate frame) and used the points to optimize a homography transform between the radar and camera. The drawback of this method is that the maximum intensity search is prone to errors, since the returned intensity depends on a number of factors, e.g., target orientation and radar antenna radiation pattern, which is usually designed to be as constant as possible in the nominal FoV.

Even though current automotive radars cannot provide 3D information about the targets (the missing elevation angle), accurate 6DoF extrinsic calibration involving a more informative sensor, e.g., LiDAR or camera, can also be especially useful for detecting *vertical misalignment*. Namely, radars should be mounted on the vehicle so that the radar and the ground plane are aligned. Vertical misalignment is loosely defined as an angular deviation between these two planes, while typical commercial radars allow the misalignment for up to a few degrees (e.g. Delphi ESR allows $\pm 1°$). With greater misalignment, radar range and detection probability are decreased, as less energy is radiated in the direction of interest. To the best of the authors' knowledge, existing related work does not address the vertical misalignment problem nor are the existing calibration methods accurate enough to provide reliable misalignment assessment. However, several misalignment detection procedures are patented [27–29], thus confirming the importance of the aforementioned issue.

Sensor calibration approaches should ideally address the aspects of identifiability, i.e., give answers if and to what extent in terms of uncertainty, one can estimate the parameters of the addressed calibration problem. Furthermore, minimal requirements on the dataset can also give practical advice on the experiment design and are also useful for robust estimation techniques (e.g. RANSAC), where the time cost of the estimation depends on the minimal size of the dataset. Some methods approach the identifiability question from the geometric viewpoint, while others from the framework of nonlinear observability or through statistical tools such as Fisher Information Matrix (FIM). In [13] authors calibrated a 3D LiDAR–camera pair by examining how the geometric point–plane constraints react in the scenarios in which they observe one, two, or three planes with linearly independent normals. Nonlinear observability analysis developed in [30] is a convenient tool for cases where system dynamics are exploited in the calibration, such as visual–inertial odometry combined with extrinsic calibration, as demonstrated in [31] and [32]. Authors in [8] presented a solution which uses corner structures to perform extrinsic calibration of multiple 2D LiDARs. To show identifiability requirements, they relied on the FIM rank to show that the problem becomes identifiable when at least three perpendicular planes are observed. FIM was also used in motion-based calibration [33] to detect unobservable directions in parameter space from the available data.

In this paper we present a novel target-based calibration method for extrinsic 6DoF calibration of 3D LiDAR–radar and camera–radar sensor pairs. By using FIM based statistical analysis, we also address the questions of parameter identifiability, estimation uncertainty, and the choice of transform parametrization. The proposed method involves a special calibration target design whose properties enable accurate cross-sensor correspondence localization and registration. Afterwards, these correspondences are used in two consecutive optimization steps: reprojection error based optimization and RCS enhanced optimization. When combined, the steps are able to accurately estimate all the 6DoF of the extrinsic calibration. The current paper draws upon our earlier work [34], where the target design and preliminary results of 3D LiDAR–radar calibration were presented. We extend this work with novel contributions by adding camera in the optimization framework, performing FIM based identifiability and estimation uncertainty analysis, introducing improved RCS enhanced optimization step, and correspondingly reporting extended experimental analysis for both sensor pairs with two radars from different manufacturers to demonstrate the validity of the proposed method.

The paper is organized as follows. Section 2 elaborates the calibration method including calibration target design and data correspondence registration. Section 3 explains two steps of the optimization: reprojection error optimization and RCS optimization. Section 4 gives insight on the theoretical background used in the identifiability analysis and the tools used in the FIM analysis. Section 5 provides details on the results of the identifiability analysis, the setup and the results of the real-world experiments. In the end, Section 6 concludes the paper.

2. Target based correspondence registration

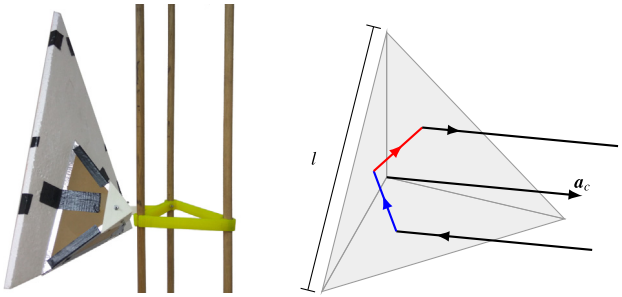
The proposed method is based on observing a calibration target placed at a range of different heights and positions, both within and outside of the nominal radar FoV. The final goal of the calibration is to estimate relative displacements between the radar, LiDAR and camera coordinate frames, i.e., $\mathcal{F}_r$, $\mathcal{F}_l$, and $\mathcal{F}_c$, respectively. In the present paper, we will designate both the 3D LiDAR and camera as 3D sensors, in the sense that they can both infer the 3D position of a known target from measurements. The method further assumes that the 3D sensor's FoV exceeds radar's vertical FoV, which is the case in most applications. Given that, when it is not necessary to differentiate between the two 3D sensor coordinate frames, we will designate the 3D sensor frame as $\mathcal{F}_s$. Additionally, due to challenges associated with radars, such as ghost measurements from multipath propagation and low angular resolution, data collection has to be performed outdoors at a set of distances ranging from 2–10 m with enough clear space around the target.

2.1. Calibration target design

Calibration target design for radar–LiDAR calibration was developed within our previous work [34], where we gave detailed remarks considering the design. However, for completeness, in this section we provide essential information necessary for the rest of the paper.

Properties of a well-designed target are (i) ease of detection and (ii) high localization accuracy for all the three sensors. For the radar, a target with a high RCS provides good detection rates. Formally, RCS of an object is defined as the area of a perfectly conducting sphere whose echo strength would be equal to the object strength [24]. Consequently, it is a function of object size, material, shape, and orientation.

We proposed a complementary target design which consist of a styrofoam triangle covered by a checkerboard-like pattern and a triangular corner retroreflector. Since the styrofoam is mostly made out of air (98%), it is virtually invisible to the radar, while its flat shape enables precise localization within the point cloud. Furthermore, its triangular shape solves localization ambiguity issues existing with common rectangular targets caused by the finite LiDAR resolution, as shown in [35] and [36]. On the other hand, the triangular corner retroreflector, which consists of three orthogonal flat metal triangles, has good detection and localization properties with the radar. It has an interesting property that any ray reflected from all three sides is returned in the same direction as illustrated in Fig. 1b. Due to this property, regardless of the incident angle, many rays are returned to their source, i.e., the radar, which leads to a high and orientation-insensitive RCS. When the retroreflector axis, $\mathbf{a}_c$, points directly to the radar, it reaches its maximum RCS value:



(a) Calibration target (b) Triangular corner retroreflector

Fig. 1. Constructed calibration target and the illustration of the working principle of the triangular trihedral corner retroreflector.

$$\sigma_c = \frac{\pi l^4}{3\lambda^2}, \tag{1}$$

where l is the hypotenuse of the retroreflector's side and λ is radar's operating wavelength. Furthermore, authors in [37] show that all the rays which go through multiple reflections travel the same length as the ray which is reflected directly from the corner centre, thus providing good localization accuracy. Lastly, target stand is designed to have RCS as small as possible, while it allows adjusting of target's height and orientation. The constructed radar calibration target and an illustration of the working principle is shown in Fig. 1a.

2.2. Correspondence registration

Correspondence registration procedure from our previous work [34] is expanded with checkerboard detection in the images. It starts with the detection and localization of a target in the LiDAR point cloud or camera image. Once we obtain the 3D location of the retroreflector origin, the rest of the method is equal for the camera–radar and LiDAR–radar calibration. Method for the target localization within the point cloud is explained in [34], while the image procedure is given in the sequel.

The intrinsic calibration of a camera, modelled as a pinhole camera with radial distortion, is found using the Kalibr toolbox [38]. In the sequel we perform all the steps on the rectified images. To estimate the position of a corner origin in the image, we use the toolbox developed in [12], which was able to effectively find the corners in our cluttered environment shown in Fig. 2. The size of the checkerboard corners was selected to present a compromise between the number of points on the target and the ability to be detected at larger distances. In the end, we opted for the size of 0.1 m. However, since our target did not have a rectangular form, we had to adapt the toolbox to accept non-square patterns. After the corners of the pattern are found, to recover the pose

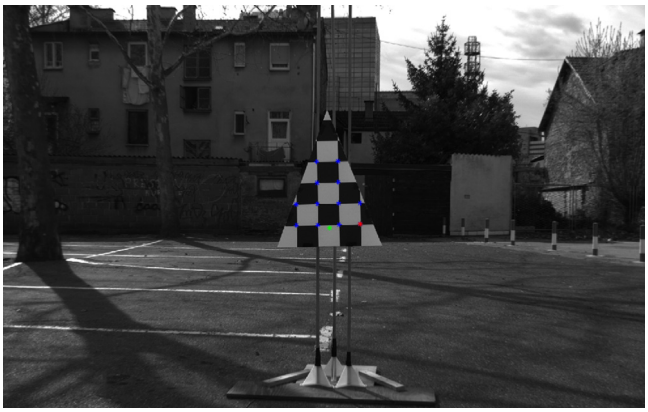


Fig. 2. Front view of the calibration target at the experiment site with detected corners (blue and origin red) and estimated position of the retroreflector origin (green).

of the triangle based on a known checkerboard configuration, we used the built-in Matlab function *extrinsics* , which is based on a closed form solution resulting with sufficient accuracy. Finally, as in the LiDAR's case, the position of the retroreflector origin ${}^{c}\mathbf{x}_c$ is calculated based on the pose of the checkerboard and known target configuration.

Radar reports data as a list of detected objects described by the measured azimuth ${}^{r}\phi_{r,i}$, range ${}^{r}r_{r,i}$ and RCS $\sigma_{r,i}$. The ith object from the list is described by the vector ${}^{r}\boldsymbol{m}_i = [{}^{r}\phi_{r,i}\ {}^{r}r_{r,i}\ {}^{r}\sigma_{r,i}]$ in the radar coordinate frame, $\mathcal{F}_r : ({}^{r}x, {}^{r}y, {}^{r}z)$. The only structural property of detected objects is contained within the RCS, which is influenced by many other factors; hence, it is impossible to classify a detection as the retroreflector based solely on radar measurements. To find the matching object, a rough initial calibration is required, e.g., with a measurement tape, which is used to transform the estimated corner position from the 3D sensors coordinate frame, $\mathcal{F}_s : ({}^{s}x, {}^{s}y, {}^{s}z)$, to the radar frame $\mathcal{F}_r : ({}^{r}x, {}^{r}y, {}^{r}z)$, and eliminate all other objects that fall outside of a predefined distance threshold. The correspondence is accepted only if a single object is left.

Lastly, we form correspondence groups by observing the target at rest for a short period while the registered correspondences fill a correspondence group with pairs of vectors ${}^{r}\boldsymbol{m}_i$ and ${}^{s}\boldsymbol{x}_s$. Variances of the radar data $({}^{r}\phi_{r,i}, {}^{r}r_{r,i}, {}^{r}\sigma_{r,i})$ within the group are used to determine the stability of the target. If any of the variances surpasses a preset threshold, the correspondence is discarded, since it is likely that the target detection was obstructed. Otherwise, the values are averaged. Hereafter, we will refer to the mean values of the groups as radar and 3D sensor measurements.

3. Two-step optimization

In this section we provide insight on how the optimization is performed to obtain the 6DoF transformation between the radar and the 3D sensor. The optimization is divided in two steps which are based on different information provided by the radar. Namely, first step, i.e., reprojection error optimization, optimizes all six transformation parameters based on the comparison of 3D corner positions estimated by the 3D sensor, and range and azimuth information provided by the radar. On the other hand, second step, i.e., RCS optimization, uses information from the 3D sensor combined with the radar RCS estimate to refine only a subset of transformation parameters which could not be estimated reliably in the first step.

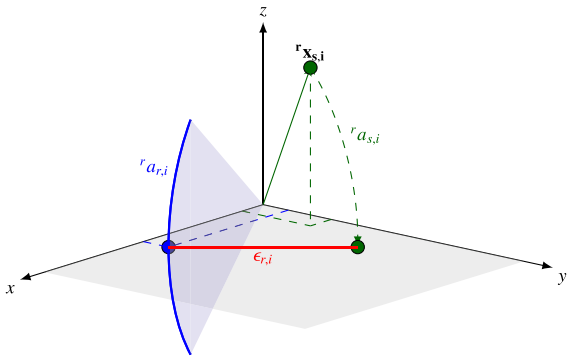


Fig. 3. Illustration of the reprojection error calculation. Green: 3D sensor's measurement; blue: radar's; red: reprojection error.

3.1. Reprojection error optimization

Reprojection error optimization is based on a point–circle geometric constraint, while the optimization parameter vector includes the translation and rotation parameters, i.e., $\boldsymbol{c}_r = [{}^{r}\boldsymbol{p}_s\ {}^{s}_{r}\boldsymbol{\Theta}]$. For translation, we choose position of the 3D sensor in the $\mathcal{F}_r$, ${}^{r}\boldsymbol{p}_s = [{}^{r}p_{s,x}\ {}^{r}p_{s,y}\ {}^{r}p_{s,z}]^T$. For rotation, we choose Euler angles parametrization ${}^{s}_{r}\boldsymbol{\Theta} = [{}^{s}_{r}\theta_z\ {}^{s}_{r}\theta_y\ {}^{s}_{r}\theta_x]$ where rotation from $\mathcal{F}_r$ to $\mathcal{F}_s$ is given by:

$$ {}^{s}_{r}R({}^{s}_{r}\boldsymbol{\Theta}) = {}^{s}_{2}R_x({}^{s}_{r}\theta_x)\,{}^{2}_{1}R_y({}^{s}_{r}\theta_y)\,{}^{1}_{r}R_z({}^{s}_{r}\theta_z). \tag{2} $$

Although transformation can be expressed in multiple ways, the proposed choice is preferable due to its distribution of uncertainty caused by radar's inability to measure elevation angle. Further elaboration of the parametrization choice will be given in Section 4.2 with results in Section 5.1 which further confirm this assertion.

Fig. 3 illustrates the calculation of the reprojection error for the ith paired measurement. As discussed previously, radar provides measurements in spherical coordinates lacking elevation ${}^{r}\boldsymbol{s}_{r,i} = [{}^{r}r_{r,i}\ {}^{r}\phi_{r,i}\ \sim]$, i.e., it provides an arc ${}^{r}a_{r,i}$ upon which the object potentially resides. On the other hand, 3D sensor provides a point in Euclidean coordinates ${}^{s}\boldsymbol{x}_{s,i}$. Using the current transformation estimate, 3D sensor measurement ${}^{s}\boldsymbol{x}_{s,i}$ is transformed into the radar coordinate frame:

$$ {}^{r}\boldsymbol{x}_{s,i}(\boldsymbol{c}_r) = {}^{s}_{r}R^T({}^{s}_{r}\boldsymbol{\Theta})\,{}^{s}\boldsymbol{x}_{s,i} + {}^{r}\boldsymbol{p}_s, \tag{3} $$

and then ${}^{r}\boldsymbol{x}_{s,i}$ is converted to spherical coordinates ${}^{r}\boldsymbol{s}_{s,i} = [{}^{r}r_{s,i}\ {}^{r}\phi_{s,i}\ {}^{r}\psi_{s,i}]$. By neglecting the elevation angle ${}^{r}\psi_{s,i}$, we obtain the arc ${}^{r}a_{s,i}$ upon which 3D sensor measurement resides and can be compared to the radar's. Reprojection error $\epsilon_{r,i}$ is then defined as the Euclidean distance of points on the arc for which ${}^{r}\psi_{r,i} = {}^{r}\psi_{s,i} = 0°$:

$$ \epsilon_{r,i}(\boldsymbol{c}_r) = \left\| \begin{bmatrix} {}^{r}r_{r,i}\cos({}^{r}\phi_{r,i}) \\ {}^{r}r_{r,i}\sin({}^{r}\phi_{r,i}) \end{bmatrix} - \begin{bmatrix} {}^{r}r_{s,i}\cos({}^{r}\phi_{s,i}) \\ {}^{r}r_{s,i}\sin({}^{r}\phi_{s,i}) \end{bmatrix} \right\|. \tag{4} $$

The estimate of the calibration parameters $\hat{\boldsymbol{c}}_r$ is obtained using the Levenberg–Marquardt (LM) algorithm, which minimizes the sum of squared reprojection errors from N measurements:

$$ \hat{\boldsymbol{c}}_{\mathbf{r}} = \arg\min_{\boldsymbol{c}_r} \left(\sum_{i=1}^{N} \epsilon_{r,i}^2(\boldsymbol{c}_r) \right). \tag{5} $$

Reprojection error optimization yields unequally uncertain calibration parameters, in other words, some parameters are easier to estimate than the others. The lack of radar's elevation angle measurement leads to poor estimation of ${}^{r}p_{s,z}$, ${}^{s}_{r}\theta_y$ and ${}^{s}_{r}\theta_x$. A formal analysis of these properties based on FIM is carried out in Section 4.

3.2. RCS optimization

For the second optimization step, i.e., the RCS optimization, we propose a method that is based on the distribution of RCS across the 3D measurements. The idea of this step is to exploit patterns discovered in radar's RCS estimation; namely, RCS depends on the object properties and relative orientation with respect to the radar. The reason behind these patterns is that radars can only estimate the RCS based on the amplitude difference between the radiated and received electromagnetic energy. Ideally, radars would radiate with constant strength within the nominal FoV and zero outside of it; however, this is infeasible and leads to errors in RCS estimation. Using the retroreflector as a calibration target, we can assume that the RCS estimate is constant with respect to the object properties, since we use the same target in all the experiments, and with respect to the relative orientation, due to the retroreflector properties. However, radars emit the highest amount of radiation at the zero elevation angle, while the dependence between elevation angle and radiated energy, and thus RCS estimation, can be modelled as a curve. Since radars cannot distinguish objects at different elevation angles, they can neither compensate for the error in the RCS estimation. For the usual application, such as object tracking, this might not seem like an exploitable property, but for our case of calibration with a target of a stable RCS, we can exploit this pattern of varying RCS with respect to elevation and enhance calibration results.

The results from the reprojection error optimization exhibit varying uncertainty among the calibration parameters, which was examined in the identifiability analysis (cf. Section 4). In the RCS optimization step, only the parameters with the highest uncertainty from the previous optimization step are refined. Given that, the RCS optimization parameter vector consist of a subset of transformation parameters and curve parameters

$$\boldsymbol{c}_\sigma = [{}^r p_{s,z}\ {}^s_r\theta_y\ {}^s_r\theta_x\ c_0, c_2],$$

while other extrinsic parameters are kept fixed. Through the empirical evaluation of the used radars, we have noticed that the RCS – elevation dependence follows a quadratic form; hence, we have modelled it as a second order polynomial without the linear term. In the experiments (cf. Section 5.2), the proposed model gave accurate and stable results for two automotive radars from different manufacturers. However, other radars might exhibit different patterns and the procedure could require a revision of the curve parametrization. To initialize curve parameters, a fair assumption is to assume that at the elevation angle zero, RCS is equal to the target maximum value defined in (1), while at the edge of the nominal FoV it reduces −3 dBm. Due to the sufficiently good initialization of transformation parameters provided by the reprojection error optimization, the proposed curve initialization showed sufficient for converging. The proposed step can be seen as a combination of extrinsic and intrinsic radar calibration, where the estimated curve is merely a nuisance variable used to obtain an enhanced extrinsic calibration (since it is of no relevance to other radar applications). Another perspective on the idea behind the RCS optimization concept is to provide a replacement for the radar's lack of elevation measurements. The prerequisite for this method is a target with reliable and stable RCS with respect to its orientation, which in our case is ensured by the retroreflector properties.

The cost function for optimization is formed as follows. After transforming a 3D sensor measurement ${}^s\boldsymbol{x}_{s,i}$ to $\mathcal{F}_r$, the elevation angle ${}^r\psi_{s,i}$ in $\mathcal{F}_r$ is calculated. Afterwards, the expected RCS is obtained using

$$\hat{\sigma}_{s,i} = c_2\,{}^r\psi_{s,i}^2 + c_0. \tag{6}$$

Cost function is then given by the sum of squared distances between the expected and measured RCS, $\hat{\sigma}_{s,i}$ and $\sigma_{s,i}$, respectively:

$$\hat{\boldsymbol{c}}_\sigma = \arg\min_{\boldsymbol{c}_\sigma} \left(\sum_{i=1}^{N} \left(\sigma_{s,i} - \hat{\sigma}_{s,i}(\boldsymbol{c}_\sigma) \right)^2 \right). \tag{7}$$

In our previous work [34], we referred to the second optimization step as the FoV optimization. Although the presently proposed and previous approach exploit the same effect, the present one shows better results and has several advantages. First, for the FoV optimization, we have noticed that it works well with many measurements, while it becomes unstable with only few measurements. The problem with the FoV optimization is that the cost function focuses only on the measurements near the nominal FoV border and ignores all the other measurements. Therefore, the proposed RCS optimization was designed so that it takes into account all the measurements. Second, FoV optimization requires predetermination of the nominal FoV, which can also affect calibration results. The nuisance parameter in the FoV optimization, i.e., the RCS threshold, requires more precise initialization than the nuisance parameters, i.e., curve parameters, in the RCS optimization.

4. Identifiability analysis

Extrinsic calibration methods typically involve minimization of a specific reprojection error depending on the type of the data provided by the sensors. This minimization will yield an estimate of the calibration parameters, but it would also be desirable if it could provide an assessment of the whole process – for example, by answering the following questions. What are the minimal conditions on the dataset to ensure identifiability of the parameters? How should the dataset be constructed to maximize the quality of the estimation? Does the chosen parametrization fit well with the optimization problem? In the sequel, we present theoretical background and experimental results that address the aforementioned questions for the calibration problem investigated in the present paper.

For dynamical systems the term *observability* is used within the context of a procedure assessing if system states can be estimated given a sequence of measurements. The term *identifiability* is used in conjunction with a procedure for estimating system parameters that are constant over time. However, the term observability is also often used within the context of estimating constant system parameters, due to commonly used tools in control theory and robotics. Nevertheless, in the present paper we use the term *identifiability*, since we believe that it more precisely describes the problem at hand. Given that, the objective of the identifiability analysis is to determine whether it is possible to correctly estimate parameters of a model based on the chosen criterion, e.g., the reprojection error, and available data. In some cases, it is possible to derive analytical solutions for such problems. However, when nonlinear transformations in the criterion grow in complexity, using methods such as those developed in [30] becomes impractical, if not infeasible. Since our reprojection error design, described in Section 3.1, involves heavy nonlinearities, we decided to adopt the statistical concept of FIM through which local identifiability of the system can shown. In the sequel, we provide the theoretical background on the FIM, followed by the description of the performed experiments that can be used to address identifiability, assess the parametrization and give general advice on the experiment setup.

4.1. Theoretical background

Before approaching any identification problem, it is important to know if it is even possible to correctly estimate the parameters

in a noise-free system. That is the intuitive purpose of the identifiability analysis. To approach it more formally, we first define our system as a nonlinear regression

$$\mathbf{Y} = \mathbf{H}(\mathbf{\Theta}, \mathbf{X}) + \boldsymbol{\epsilon}, \tag{8}$$

where the response variable $\mathbf{Y} \in \mathbb{R}^{2N\times 1}$ represents radar measurements, the predictor variable $\mathbf{X} \in \mathbb{R}^{3N\times 1}$ represents LiDAR measurements, $\mathbf{\Theta} \in \mathbb{R}^d$ are parameters of the extrinsic calibration, $\epsilon \sim \mathcal{N}(0, \mathbf{Q}) \in \mathbb{R}^{2N\times 1}$ is additive zero-mean white noise, N is the number of measurements, and the nonlinear transformation $\mathbf{H}(\cdot)$ represents the reprojection function. We can notice that LiDAR measurements are modelled as noise-free. This may lead to slight imprecision in the simulation of the error; however, we are not here concerned with precise estimation of the error and covariance, but with the impact of the proposed reprojection error on the identifiability of the calibration parameters.

Identifiability can be a global or a local concept for a specific $\mathbf{\Theta_0}$ [39]. Since FIM cannot provide insights into global identifiability, we restrict our analysis to local identifiability. This is sufficient for our method, since we assume to have a rough initial estimate of the parameters, e.g., by hand measuring the displacements or from the project design. Now, we move on to more formally defining the local identifiability.

Definition 4.1 (*Local Identifiability*)**.** The noise-free system is locally identifiable at $\mathbf{\Theta_0}$ if

$$\exists U_{\mathbf{\Theta_0}} \subset \mathbb{R}^d \text{ (open subset containing } \mathbf{\Theta_0})$$

$$\forall \mathbf{\Theta} \in U_{\mathbf{\Theta_0}}, \{\mathbf{\Theta} \neq \mathbf{\Theta_0}\} \Rightarrow \{\mathbf{H}(\mathbf{\Theta}, \mathbf{X}) \neq \mathbf{H}(\mathbf{\Theta_0}, X)\}.$$

In other words, for a different parameter set the nonlinear function cannot yield the same output. This is intuitively clear, since we would like to see a change in the response variable given the change in the parameter values. Another theoretical concept that we require for the present problem is the *score*.

Definition 4.2 (*Score Function*)**.** The score function $\dot{\mathcal{L}}_{\mathbf{\Theta}}$ is the gradient of the log-likelihood function $\mathcal{L}(\mathbf{Y}; \mathbf{\Theta}, \mathbf{X})$ at $\mathbf{\Theta}$

$$\dot{\mathcal{L}}_{\mathbf{\Theta}} = \nabla_{\mathbf{\Theta}} \log \mathcal{L}(\mathbf{Y}; \mathbf{\Theta}, \mathbf{X}).$$

The score function can be seen as an indicator of how sensitive the likelihood functions is to the change in its parameters. Intuitively, this would mean that higher the sensitivity, the more easy it should be to estimate the parameter. An interesting notion that we will use is that FIM is defined as the covariance matrix of the score.

Informally, FIM tells how much information about the parameters is available in any direction of the parameter space from observing the sample. Since the expected value of the score is zero, FIM is a positive semi-definite matrix of size $d \times d$ whose elements can be computed as

$$\left[\mathcal{I}(\theta)\right]_{i,j} = \mathrm{E}_\theta \left[\left(\frac{\partial}{\partial \theta_i} \log \mathcal{L}(Y; \theta, X) \right) \left(\frac{\partial}{\partial \theta_j} \log \mathcal{L}(Y; \theta, X) \right) \right]. \tag{9}$$

Since we defined our problem as a nonlinear regression with additive white noise, our likelihood function is simply a well-known probability density function of a multivariate normal distribution. For such cases, it can be shown that calculation of the FIM elements simplifies to [39]

$$\left[\mathcal{I}(\theta)\right]_{i,j} = \frac{\partial \mathbf{H}(\mathbf{\Theta}, \mathbf{X})}{\partial \theta_i} \mathbf{Q}^{-1} \frac{\partial \mathbf{H}(\mathbf{\Theta}, \mathbf{X})^{\mathrm{T}}}{\partial \theta_j}. \tag{10}$$

As discussed in [40], such simplification is beneficial, especially for numerical accuracy, which can cause problems in complex nonlinear problems.

It is also worth mentioning some additional properties of FIM that are commonly used. First, the Cramér–Rao lower bound (CRLB), calculated as an inverse of the FIM, is used to express the lower bound on the variance of the estimated parameters. Second, if we draw independent identically distributed samples, likelihood function is simply the product of individual likelihoods, whereas log-likelihood turns into summation of the individual log-likelihoods. Due to linearity, this property also holds for FIM. Therefore, if we draw two data samples of the same random variable, maximum information expressed with FIM is doubled. Finally, in [41] it was shown that the local identifiability as defined in Definition 4.1 is equivalent to the regularity of the FIM. Therefore, if FIM is not of full rank, we conclude that the problem is not identifiable.

4.2. FIM tests

After the FIM has been evaluated at the estimated maximum likelihood estimate, we proceed with test which will give us insight into: (i) minimal requirements on the dataset which ensures identifiability of our problem, (ii) appropriateness of the parametrization, and (iii) general advice on the dataset collection. To evaluate our reprojection function, we will create synthetic datasets and test FIM behaviour.

To show the minimal requirements on the dataset, we will apply the rank test of FIM. For the case of 3D point – point correspondences, at least three non-colinear points are required to estimate the 6D transformation between two coordinate frames [42]. However, the problem that we face is more complex, and our reprojection error is less informative, because we use point–circle correspondences. Therefore, it is a fair assumption to take three non-colinear, but coplanar points, as a starting dataset and expand it to find the minimal requirements. The FIM is computed for each dataset and based on its regularity, we infer on the identifiability. Furthermore, numerical inaccuracies and noise can result in an illusory full rank of FIM; therefore, it is advisable to examine the numerical rank of the matrix [43]. A convenient summary statistic is given by the matrix conditional number, i.e., the ratio of the biggest and the smallest singular value, where high values indicate degeneracy of the matrix.

In order to evaluate the choice of parametrization and to provide some practical advice on the dataset collection, we will also rely on the theory of optimal experiment design. The optimal experiment is the experiment that allows estimation of parameters without bias and with minimum variance with equal or less experiment data than any other non-optimal experiment. There exist many optimality criteria which a single experiment can satisfy; however, we will use only the T-optimality criterion, which tries to maximize the trace of the FIM. It is convenient as it tells us that we can observe only the diagonal elements of the FIM, which actually represent informativeness of individual parameters. With this tool at our disposal, we are able to infer on how different datasets affect estimation of individual parameters.

Furthermore, extrinsic calibration seeks for a homogeneous transformation which can be parametrized in a number of ways. Translation can be expressed in any of the two coordinate frames, while orientation can expressed through multiple Euler angle parametrizations. Generally, it may seem counterintuitive that a certain parametrization of the transformation can be preferable to others. However, for our calibration method it is important due to the second optimization step – the RCS optimization. Namely, in that step we do not refine all the parameters estimated in the first step, the reprojection error optimization, but only the poorly estimated parameters (which we will be able to identify with our FIM tests). However, we justify locking the parameters that were well estimated by concentrating the information in them. Our aim

is to show, through the FIM tests, that our parametrization has highest concentration of information in the locked parameters for a variety of sensor configurations. This result is a direct consequence of the radar's inability to measure the elevation angle.

5. Experiment

To test the proposed calibration method, we conducted both simulated and real-world data experiments. Through the simulations described in Section 5.1, based on the framework of FIM described in Section 4, we have tested the properties of designed reprojection error. Afterwards, in Sections 5.2 and 5.3 we describe the setup of the conducted experiment and the final results, respectively. Finally, in Section 5.4, we present a real world application where our calibration method is used to find radar vertical misalignment.

5.1. Simulations

To test our method in simulations under various conditions, we have created a number of different synthetic datasets described with the labelled tuple $\mathcal{D}_{label} = ({}^{\mathbf{S}}\mathbf{X}, {}^{\mathbf{R}}\mathbf{Y}, \mathcal{T}, N, S)$ where:

- ${}^{\mathbf{S}}\mathbf{X}$ represents the measurement set originating from a 3D sensor in the sensor coordinate frame $\mathcal{F}_{\mathcal{S}}$,
- ${}^{\mathbf{R}}\mathbf{Y}$ represents the planar measurement set originating from the radar in the radar coordinate frame $\mathcal{F}_{\mathcal{R}}$,
- $\mathcal{T}$ represents the transformation between $\mathcal{F}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{R}}$ which can be parametrized in different forms,
- N represents number of unique measurement points in the dataset,
- S represents number of samples of each unique point in the dataset.

For simulation purposes we have assumed a diagonal covariance matrix $\mathbf{Q} = \text{diag}(\sigma^2)$, where $\sigma^2 = 6.25 \times 10^{-4}\,\text{m}^2$. Furthermore, as datasets are comprised of a different number of unique measurements N, S compensates for the total number of used points. It allows a fair comparison of FIMs since amount of information is proportional to the number of points. The measurements are first given in radar's spherical coordinates ${}^{r}\boldsymbol{s}_{r,i} = [{}^{r}r_{r,i}\ {}^{r}\phi_{r,i}\ {}^{r}\psi_{r,i}]$, with ϕ and ψ being azimuth and elevation, respectively. Afterwards, they are transformed into the 3D sensor frame ${}^{\mathbf{S}}\mathbf{X}$, and in radar's planar measurements in the zero-elevation plane ${}^{\mathbf{R}}\mathbf{Y}$ (cf. Section 3.1).

Minimal requirements on the number of measurements is found by examining FIM singular values for the following marginal datasets: $\mathcal{D}_{3CP}$ consists of three ($N = 3$) coplanar, non-colinear points at ${}^{r}r_{r,i} = 5\,\text{m}$, ${}^{r}\phi_{r,i} = [-45, 0, 45]^\circ$, ${}^{r}\psi_{r,i} = 0^\circ$; $\mathcal{D}_{4CP}$ consists of four ($N = 4$) coplanar, non-colinear points at ${}^{r}r_{r,i} = 5\,\text{m}$, ${}^{r}\phi_{r,i} = [-45, -15, 15, 45]^\circ$, ${}^{r}\psi_{r,i} = 0^\circ$; $\mathcal{D}_{4nCP}$ consists of four ($N = 4$) non-coplanar, non-colinear points at ${}^{r}r_{r,i} = 5\,\text{m}$, ${}^{r}\phi_{r,i} = [-45, -45, 45, 45]^\circ$, ${}^{r}\psi_{r,i} = [-5, 5, -5, 5]^\circ$. Additionally, dataset $\mathcal{D}_{FoV}$ consists of $N = 300$ uniformly spread points through the FoV within the following range, azimuth and elevation intervals: ${}^{r}r_{r,i} = [4, 5]\,\text{m}$, ${}^{r}\phi_{r,i} = [-45, 45]^\circ$, ${}^{r}\psi_{r,i} = [-5, 5]^\circ$. It illustrates the upper bound on the achievable parameter informativeness. FIM analysis results for the four datasets are shown in Table 1, where we are striving to have the singular values as large as possible, since it suggests identifiability of the parameters. Note that at this point we are not concerning ourselves which exact parameters are identifiable, but only with if all the 6 parameters of the relative transformation between the coordinate frames are identifiable. By examining smallest singular values, we can see an evident increase ($\sim 10^4$) in the conditional number κ, i.e., the ratio between the largest and smallest singular value, when the non-coplanar point is added to the dataset (note the increase of the smallest singular value σ_6). Difference in the order of magnitude between the largest and smallest singular value for $\mathcal{D}_{4nCP}$ still exists, but unlike the other two datasets, this is not caused by the degeneracy of FIM, i.e., non-identifiability. It is caused by different scales of the parameters, i.e., Euler angles and translation, and uneven sensitivity in the parameters, which is further elaborated in the justification of parametrization choice. This conclusion was also confirmed by the optimization results, since regardless of how big of an S we chose, the reprojection error optimization was unable to converge close to parameter ground truth values for $\mathcal{D}_{3CP}$ and $\mathcal{D}_{4CP}$, while for $\mathcal{D}_{4nCP}$ and $\mathcal{D}_{FoV}$ it always converged successfully. Finally, dataset $\mathcal{D}_{FoV}$ shows that adding more unique points to the dataset does not present a significant impact on the singular values in terms of the identifiability. This brings us to the first important result of the identifiability analysis. To calibrate a radar and a 3D sensor, the previous analysis suggests that to have all the 6 parameters identifiable, the best course of action would be to have at least 4 non-coplanar non-colinear points in the dataset.

The second important result of the identifiability analysis is the justification of the parameter locking in the second optimization step, which, as we will see, is related to parametrization of the relative transformation between the two sensor frames. We have conducted four experiments which differ only in the poses between the sensor coordinate frames and the parametrization of the pertaining transformation. The dataset $\mathcal{D}_{rPs\_0}$ consists of a transformation $\mathcal{T}_{rPs\_0}$ that assumes the simplest case of no rotation and translation between the sensors, while the parametrization is the same as the one defined in Section 3.1 – translation defined as the position of the 3D sensor in $\mathcal{F}_{\mathcal{R}}$, i.e., radar's coordinate system. The dataset $\mathcal{D}_{sPr\_0}$ differs in the parametrization of the translation. Namely, it is defined as the position of the radar in $\mathcal{F}_{\mathcal{S}}$. The other two datasets, $\mathcal{D}_{rPs\_45}$ and $\mathcal{D}_{sPr\_45}$, share the same differences in the translation parametrization, but they also assume that there exists a difference in the pitch angle ${}^{s}_{r}\theta_y = 45^\circ$ between the radar and 3D sensor. All the datasets use $N = 300$ unique ($S = 1$) uniformly distributed measurements within the following range, azimuth and elevation intervals: ${}^{r}r_{r,i} = [2, 8]\,\text{m}$, ${}^{r}\phi_{r,i} = [-75, 75]^\circ$, ${}^{r}\psi_{r,i} = [-10, 10]^\circ$.

By analysing the results for this experiment, which are shown in Table 2, we can notice that the datasets with the same sensor poses, $\mathcal{D}_{rPs\_0}$ and $\mathcal{D}_{sPr\_0}$, but different translation parametrization, exhibit the same FIM results, which confirms uneven uncertainty, or equivalently, uneven informativeness in estimating each parameter. We can see that the yaw angle ${}^{s}_{r}\theta_z$ is significantly more informative than the other Euler angles. Similarly, translations in directions ${}^{r}p_{s,x}$ and ${}^{r}p_{s,y}$ are more informative compared to the direction ${}^{r}p_{s,z}$. For $\mathcal{D}_{rPs\_0}$ and $\mathcal{D}_{sPr\_0}$, the uncertainty is equivalent for directions ${}^{s}p_{r,x}$, ${}^{s}p_{r,y}$, and ${}^{s}p_{r,z}$ since the axes coincide due to the lack of rotation between the sensor frames. However, if we observe datasets $\mathcal{D}_{rPs\_45}$ and $\mathcal{D}_{sPr\_45}$, which include displacement in rotation, we can notice significant differences in FIM diagonal elements for the two translation parametrizations. Namely, when the translation is defined in $\mathcal{F}_{\mathcal{R}}$, informativeness remains the same as in the previous two cases. However, if the translation is expressed in $\mathcal{F}_{\mathcal{S}}$, we can notice that informativeness somewhat decreases in the ${}^{s}p_{r,x}$ direction, while it increases in the ${}^{s}p_{r,z}$ direction, leading to the same informativeness of the two directions.

The main cause for this uneven informativeness of the parameters is radar's inability to measure the elevation angle. To illustrate the assertion, we refer to Fig. 4. We observe the effect on a single measurement, $\mathbf{X} = [2\,\text{m}, 0^\circ, 0^\circ]$, for two cases: when the radar is translated along its ${}^{r}x$ and along its ${}^{r}z$ axis, yielding new measurements $\mathbf{X_x}$ and $\mathbf{X_z}$, respectively. The 3D sensor and the target from which the measurement originates are kept fixed. Measurement $\mathbf{X_x} = [1.8\,\text{m}, 0^\circ, 0^\circ]$ is acquired by translating the radar along the direction of ${}^{r}x$ for $\Delta{}^{r}p_{s,x} = 0.2\,\text{m}$, while the measurement

Table 1
FIM singular values for the three datasets used for analysing the minimum number and the distribution of points in the dataset.

	ς_1	ς_2	ς_3	ς_4	ς_5	ς_6	κ
$\mathcal{D}_{3CP}$	1.17×10^7	5.07×10^5	1.84×10^5	8.83×10^3	4.38×10^3	1.58×10^{-1}	7.41×10^7
$\mathcal{D}_{4CP}$	1.17×10^7	5.05×10^5	1.57×10^5	8.98×10^3	3.69×10^3	6.47×10^{-1}	1.81×10^7
$\mathcal{D}_{4nCP}$	1.18×10^7	5.18×10^5	2.59×10^5	5.09×10^4	4.33×10^4	3.70×10^3	3.19×10^3
$\mathcal{D}_{FoV}$	1.01×10^7	4.79×10^5	8.83×10^5	2.15×10^4	4.86×10^3	1.29×10^3	7.83×10^3

Table 2
FIM's diagonal elements corresponding to the informativeness of individual parameters.

	${}^s_r\theta_z$	${}^s_r\theta_y$	${}^s_r\theta_x$	p_x	p_y	p_z
$\mathcal{D}_{rPs\_0}$	1.37×10^7	5.81×10^4	8.28×10^4	4.79×10^5	4.80×10^5	4.81×10^3
$\mathcal{D}_{sPr\_0}$	1.37×10^7	5.81×10^4	8.28×10^4	4.79×10^5	4.80×10^5	4.81×10^3
$\mathcal{D}_{rPs\_45}$	1.37×10^7	5.28×10^4	6.87×10^6	4.78×10^5	4.81×10^5	4.74×10^3
$\mathcal{D}_{sPr\_45}$	1.37×10^7	5.28×10^4	6.87×10^6	2.41×10^5	4.81×10^5	2.41×10^5

$\mathbf{X_z} = [2.01\,\text{m}, 0°, 5.7°]$ is acquired by translating the radar along the direction of ${}^r z$ for $\Delta^r p_{s,z} = 0.2\,\text{m}$. The only difference that radar detects, in this case, is the change in the range measurement which is significantly smaller in the case of $\mathbf{X_z}$. To generalize, if the radar is displaced along its xy-plane, or rotates around its ${}^r z$ axis, it would produce significant changes in range or azimuth or both. Meanwhile, the elevation, which is unavailable, would not take away the information about the translation or rotation, which is a case for the changes in the other parameters.

Furthermore, Fig. 4 explains why parametrization $\mathcal{T}_{rPs\_45}$ is preferred to $\mathcal{T}_{sPr\_45}$. Namely, in $\mathcal{T}_{rPs\_45}$, where ${}^r x$ coincides with the range, previously elaborated uncertainty has the most spread form. On the other hand, in $\mathcal{D}_{sPr\_45}$, this uncertainty is equally spread between ${}^s x$ and ${}^s z$, which is confirmed by the FIM analysis in Table 2. Different distribution of uncertainty, due to different parametrization, would merely be a preference if we performed only reprojection error optimization, since it does not provide any more information or lead to better calibration. However, the second step tries to compensate for the lack of radar's elevation angle measurements based on the RCS estimation. Since RCS measurements are less reliable, we do not want to refine parameters which can be properly estimated through reprojection error. Therefore, it is desirable to separate reliable from unreliable parameters, as good as possible, which is the case when translation is given in $\mathcal{F}_\mathcal{R}$ as ${}^r\mathbf{p}_s$, while there is a rotation which coincides with the rotation around the ${}^r z$ axis.

Finally, based on the simulation results, we can give some general advice on the dataset collection. Despite the omnipresent rule, the more the merrier, we would like to emphasize the requirement on the observation of the target at a wide range of radar elevation angle. If we observe points only in the radar plane (zero elevation), we would obviously provide a degenerate case as shown in the minimal requirements tests. As we observe the target at a wider range of elevation angles, we are further away from the singularity. However, when we observe the target at greater elevation angles, there is a risk we might misinterpret the target stand for the actual target. This is best avoided by determining the target stand RCS and performing RCS thresholding. Better results of reprojection error optimization will provide better initial values for the RCS optimization.

5.2. Experiment setup

An outdoor experiment was conducted to test the proposed calibration method. A mobile robot Husky UGV, shown in Fig. 5 was used as the platform for data collection. It was equipped with a Velodyne HDL-32E 3D LiDAR, two short range radars from different manufacturers, namely the Continental SRR 20X and Delphi SRR2, and PointGrey camera sensor combined with Kowa lens with resolution 1920 × 1080 and HFoV × VFoV = 60° × 40°.

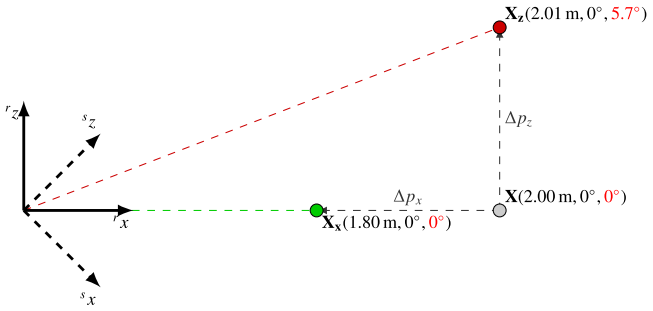
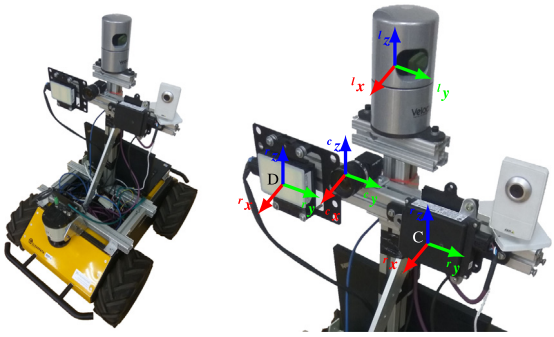


Fig. 4. Illustration of unequal uncertainty in parametrization caused by radar's inability to measure elevation angle (indicated in red for reference). $\mathbf{X_x}$ and $\mathbf{X_z}$ show how a single radar measurement in spherical coordinates $\mathbf{X}$ changes when the radar is translated for 0.2 m along ${}^r x$ and ${}^r z$, respectively.



(a) Mobile robot (b) Sensor placement

Fig. 5. Mobile robot and sensors used in the experiment. Marks D and C stand for Delphi and Continental radar, respectively.

Commercially available radars are sensors which provide high level information in the form of detected object list. Raw data, i.e., the return echo, is processed by proprietary signal processing techniques and is unavailable to the user. However, from the experiments conducted with both radars, we noticed that they follow the behaviour as expected from our calibration method. The only noticed difference is that the target stand without the target was completely invisible to the Continental radar, while the Delphi radar was able to detect it at closer ranges (${}^r r_{r,i} < 5\,\text{m}$).

Although the purpose of the experiment is evaluation of the proposed calibration method and not radar performance, we believe it is important to present results for two different radars since they exhibit slightly different behaviour as previously elaborated. Furthermore, RCS optimization uses a novel metric based on a pattern that may not be equal for all the radars. Therefore, success

Table 3
Continental SRR 20X specifications.

Continental SRR 20X	Value
HFoV × VFoV 150° × 12°	
Range Accuracy	0.2 m
Azimuth Accuracy @ HFoV	±2°@±20°; ±4°@±60°; ±5°@±75°

Table 4
Delphi SRR2 specifications.

Delphi SRR2	Value
HFoV × VFoV	150° × 10°
Range accuracy	0.5 m (noise error); 2.5% (bias error)
Azimuth accuracy @ HFoV	±1°@±75°

of the calibration using radars from two different manufacturers further confirms the validity of the proposed method.

Technical data of interest for the Continental and Delphi radars is given in Tables 3 and 4, respectively. Based on the analysis of the reprojection error for the Continental radar, radar measurements outside of the azimuth angle range of ±45° were excluded from the optimization, because they exhibited significantly higher reprojection error than those inside the range. For the Delphi radar, measurements outside of the azimuth angle range of ±60° were also excluded due to the observed increase in the reprojection error. The calibration target was composed of a retroreflector with side length $l = 0.32$ m with a maximum RCS of $\sigma_c = 18.75$ dBm^2. Based on the vertical resolution of Velodyne HDL-32E LiDAR (1.33°), we used a styrofoam triangle of height $h = 0.65$ m. This ensured extraction of at least two lines from the target in the experimental data, which is a prerequisite to unambiguously determine the pose. Data acquisition was done by driving a robot in the area of up to 7 m of distance from the target which was placed at 17 different heights ranging from ground level up to a 2 m height. For the Continental radar, 334 registered radar–LiDAR and 227 radar–camera corresponding measurements were collected. For the Delphi radar, 322 registered radar–LiDAR and 193 radar–camera corresponding measurements were collected.

5.3. Experimental results

In this section we present calibration results of four sensor combinations, i.e., two radars combined with the camera and 3D LiDAR. Since the method for the LiDAR–radar and camera–radar differs only in the step of target 3D localization, results of all the experiments are shown simultaneously for both pairs. We have noticed larger difference in calibration results when using different radars, as opposed to calibrating different sensor types with the same radar. Therefore, we first present calibration results for the Continental radar combined with both 3D sensors, and then we show results of the calibration involving the Delphi radar.

To assess the quality of calibration results we conducted four experiments. First, we examined the distribution of the reprojection error after both optimization steps and compared it to a 2D optimization that minimizes reprojection error by optimizing only the calibration parameters with lower uncertainty, i.e., translation parameters ${}^r p_{l,x}$ and ${}^r p_{l,y}$, and rotation ${}^s_r\theta_z$. Second, we inspect FoV placement with respect to the distribution of RCS over the 3D sensor's data. Afterwards, we examine the correlation between RCS and the elevation angle. In the end, we run Monte Carlo simulations by random bootstrap resampling with replacement of the dataset, to examine reliability of the estimated parameters and potential overfitting of data.

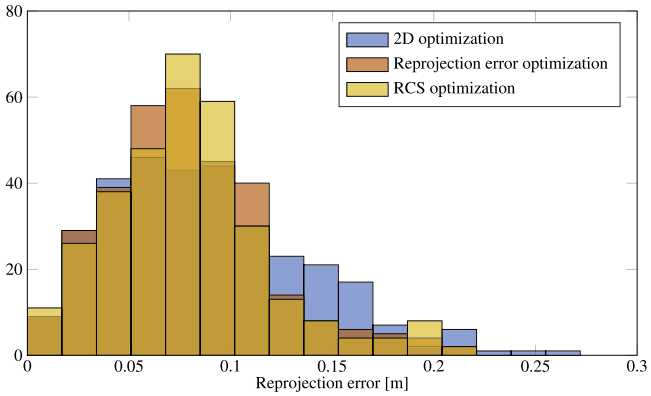


Fig. 6. Histogram of reprojection errors for the two steps of the calibration and the 2D calibration for Continental radar–LiDAR calibration.

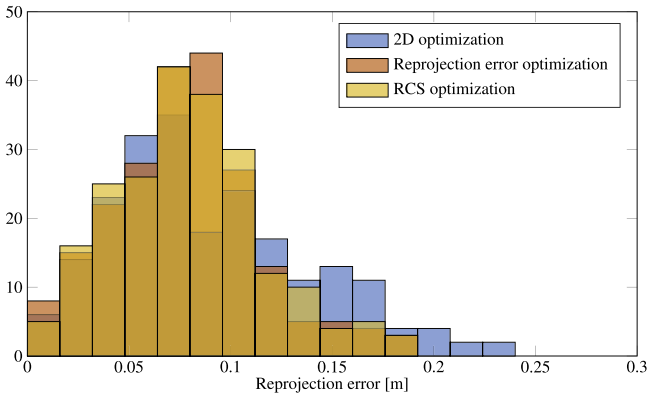


Fig. 7. Histogram of reprojection errors for the two steps of the calibration and the 2D calibration for Continental radar–camera calibration.

5.3.1. Continental radar

We obtained the following results for the reprojection error optimization, ${}^l\hat{\boldsymbol{c}}_r$, RCS optimization ${}^l\hat{\boldsymbol{c}}_\sigma$, and the carefully hand measured translation, ${}^r\tilde{\boldsymbol{p}}_l$, for the Continental radar–LiDAR pair:

- ${}^l\hat{\boldsymbol{c}}_r = [-0.05\,\text{m}, -0.14\,\text{m}, 0.11\,\text{m}, -2.2°, 5.1°, -1.7°]$
- ${}^l\hat{\boldsymbol{c}}_\sigma = [0.20\,\text{m}, 4.8°, -0.8°, -0.13\,\text{dBm}^2\,\text{deg}^{-2}, 16.2\,\text{dBm}^2]$
- ${}^r\tilde{\boldsymbol{p}}_l = [-0.08\,\text{m}, -0.12\,\text{m}, 0.19\,\text{m}]^T$.

Furthermore, for the Continental radar – camera pair, we obtained the following results:

- ${}^c\hat{\boldsymbol{c}}_r = [0.04\,\text{m}, -0.15\,\text{m}, -0.08\,\text{m}, 0.1°, 5.9°, -2.3°]$
- ${}^c\hat{\boldsymbol{c}}_\sigma = [0.04\,\text{m}, 5.6°, -1.6°, -0.15\,\text{dBm}^2\,\text{deg}^{-2}, 16.2\,\text{dBm}^2]$
- ${}^r\tilde{\boldsymbol{p}}_c = [0.00\,\text{m}, -0.15\,\text{m}, 0.04\,\text{m}]^T$.

Figs. 6 and 7 show distribution of the reprojection errors for LiDAR–radar and camera–radar calibrations, respectively. They are composed of three histograms, where we can see how the case of 2D calibration compares to the reprojection error of both steps of the proposed calibration. Besides neglecting three additional DoF, 2D reprojection error assumes that all the measurements reside in the same plane, thus reducing the original circle–point relationship to point – point distance. Although this 2D reprojection metric is not the same as the one used for our two steps of optimization, it is the only fair comparison since 2D optimization is based on minimizing it. We can observe that the 2D reprojection error has a larger number of point correspondences with higher reprojection error. These originate from the measurements that are further away from the radar plane because the circle–point relationship has a greater impact than the 2D optimization can

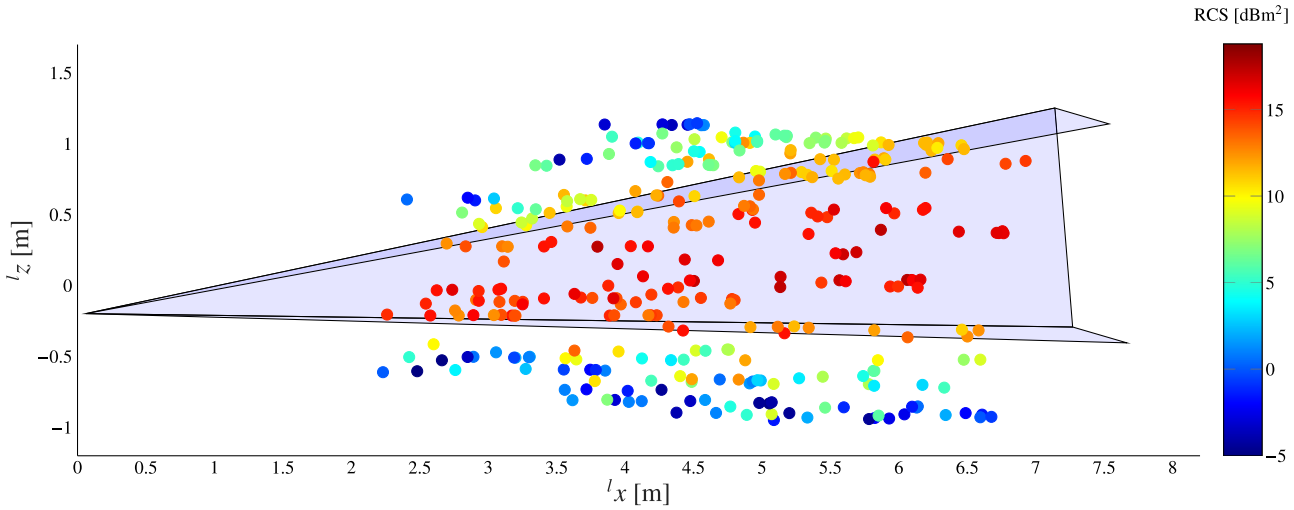


Fig. 8. RCS distribution across LiDAR 3D data and placement of the Continental radar's FoV.

explain. Therefore, we conclude that neglecting the 3D nature of the problem causes higher mean of the reprojection error which implies poor calibration. Furthermore, the RCS optimization is bound to degrade the overall reprojection error because it is not a part of the optimization criterion. However, resemblance between the distributions after the first and the second optimization steps implies low degradation. Finally, it can be seen from both the first and second optimization step, that the reprojection error is below the nominal range accuracy of the radar.

In Fig. 8, distribution of the RCS across LiDAR's data is shown, while we omit results for camera since they do not exhibit any significant difference. Measurements from 3D sensors are colour-coded with the RCS of the paired radar measurement, while the pose of the radar's nominal FoV is illustrated with blue bounding pyramid. We can see that within the nominal FoV, target produces a strong, fairly constant reflections. As the elevation angle of the target leaves the radar's nominal FoV, the RCS decreases and this effect is the basis of the RCS optimization step.

To examine the effect of decrease in the target's RCS as a function of the elevation angle after both optimizations, we use Figs. 9 and 10 for LiDAR–radar and camera–radar results, respectively. Each figure shows elevation $^{r}\psi_{s,i}$ of each 3D sensor measurement transformed into the $\mathcal{F}_r$ and RCS of the paired radar measurement. Furthermore, intrinsic radar curve estimated by the RCS optimization is plotted. In the ideal case, i.e., if the transformation was correct and the axis of retroreflector always pointed directly to the radar, the data would lay on the curve which describes radar's radiation pattern with respect to the elevation angle. The dispersion around the curve is present in both steps due to imperfect directivity of the target and measurement noise. We have evaluated directivity of the target towards the radar after the calibration and noticed that all the measurements differed less than 18° from the ideal directivity in the context of maximum response. According to experimental results in [24], such small angles do not reduce retroreflector's RCS significantly, which, combined with errors in directivity estimation, prevents us from performing directivity compensation. We assert that this is not crucial in our case. However, if the experiment is performed in such way that corner retroreflector orientation differs significantly from the ideal, we believe that RCS directivity compensation would be necessary. From the plots, we can notice that dispersion of the measurements after the reprojection error optimization is higher compared to the case of RCS optimization. This effect is caused due to the poor

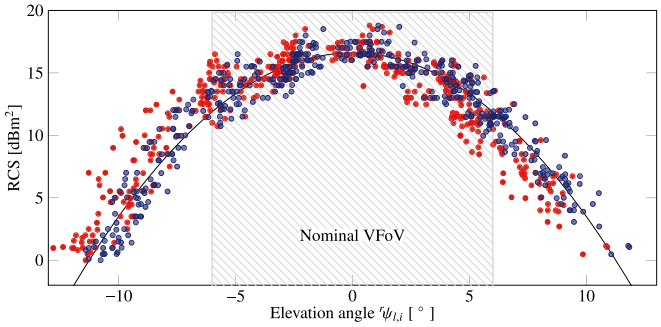


Fig. 9. RCS distribution across radar's VFoV for Continental radar–LiDAR calibration. Red: reprojection error optimization; blue: RCS optimization.

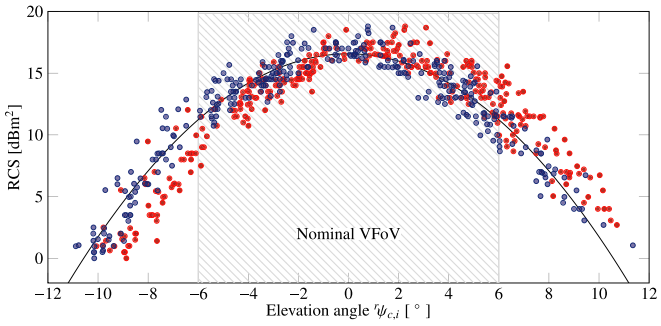


Fig. 10. RCS distribution across radar's VFoV for Continental radar–camera calibration. Red: reprojection error optimization; blue: RCS optimization.

estimation of the parameters with higher uncertainty which are corrected by the RCS optimization.

In the end, we performed Monte Carlo analysis to test how sensitive our parameter estimates are to the available dataset. We performed random bootstrap resampling with replacement of our dataset. Optimization was performed in 1000 runs on different randomly sampled datasets from which we observe the estimated extrinsic calibration parameters. The results follow a Gaussian distribution whose estimated parameters are given in Table 5 for the LiDAR–radar calibration and Table 6 for the camera – radar calibration. As expected, distributions of parameters $^{r}p_{s,x}$, $^{r}p_{s,y}$ and

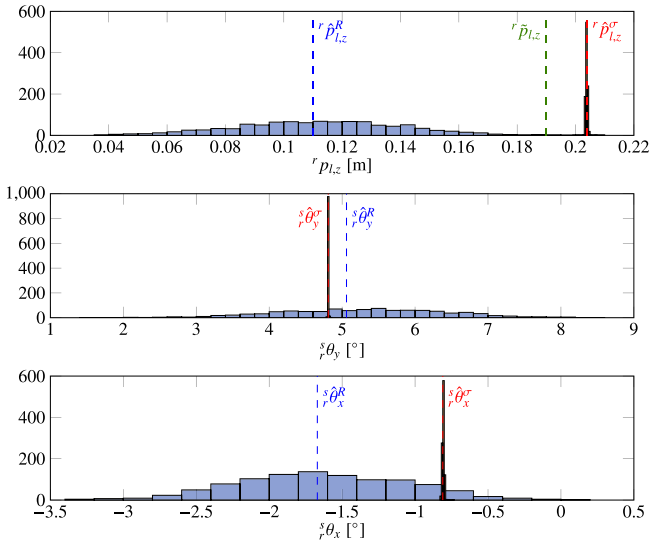


Fig. 11. Monte Carlo analysis results for Continental radar–LiDAR calibration. Blue: calibration after reprojection error optimization; red: with RCS optimization.

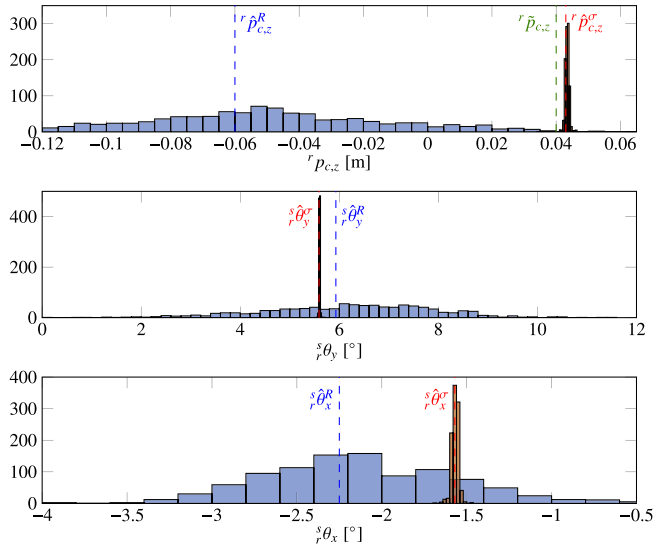


Fig. 12. Monte Carlo analysis results for Continental radar–camera calibration. Blue: calibration after reprojection error optimization; red: with RCS optimization.

${}^{s}_{r}\theta_z$ obtained by the reprojection error optimization have significantly lower variance than the other parameters. Figs. 11 and 12 illustrate how the RCS optimization refines parameters ${}^{r}p_{s,z}$, ${}^{s}_{r}\theta_y$ and ${}^{s}_{r}\theta_x$. We can see significant decrease in variance, as well as the shift in the mean. For the purposes of distribution visualisation, we have reduced the bin size for RCS optimization to 10% compared to the results for the reprojection error optimization. Otherwise, all the results would fall within one bin due to the significantly lower variance. Estimation of the mean of ${}^{r}p_{s,z}$ using the reprojection error optimization is clearly further away from the measured value than the RCS optimization's estimate. Estimation of ${}^{r}p_{c,z}$ is fairly close to the measured value, while ${}^{r}p_{l,z}$ exhibits a slight bias of 2 cm from the measured value. The cause of the bias could be imprecise hand-measurement or the systematic errors in the LiDAR's estimates of the retroreflector's position. Furthermore, a bias compared to the hand-measured values is also visible in the estimation of both ${}^{r}p_{l,x}$ and ${}^{r}p_{c,x}$. We believe that it could originate from the bias in the radar's range measurements or the imprecision of the target design. However, when introduced to the reprojection error optimization as a parameter, it could not be distinguished between the translation parameters.

5.3.2. Delphi radar

From the experimental results for the Delphi radar, we noticed that estimation of the ${}^{r}p_{s,x}$ exhibited a noticeable offset from the measured value. Although the exact origin of this bias is uncertain, we hypothesize that most likely it is an effect caused by the target stand. Namely, since the Delphi radar is able to detect the stand, proprietary algorithms that determine the range of the observed object could infer that a target is at the greater range. To address this issue, the reprojection error estimation was expanded with estimation of the range offset that is subtracted from the radar range measurements, $\boldsymbol{c}_r = [{}^{r}\boldsymbol{p}_s\ \boldsymbol{\Theta_s}, \Delta^{r}r_r]$.

We obtained the following results for the reprojection error optimization, ${}^{l}\hat{\boldsymbol{c}}_r$, RCS optimization ${}^{l}\hat{\boldsymbol{c}}_\sigma$, and the carefully hand measured translation, ${}^{r}\tilde{\boldsymbol{p}}_l$, for the Delphi radar–LiDAR pair:

- ${}^{l}\hat{\boldsymbol{c}}_r = [-0.07\,\text{m}, 0.13\,\text{m}, 0.11\,\text{m}, -2.9°, 5.0°, 7.6°, 0.10\,\text{m}]$
- ${}^{l}\hat{\boldsymbol{c}}_\sigma = [0.21\,\text{m}, 2.0°, -0.2°, -0.25\,\text{dBm}^2\,\text{deg}^{-2}, 17.9\,\text{dBm}^2]$
- ${}^{r}\tilde{\boldsymbol{p}}_l = [-0.08\,\text{m}, 0.15\,\text{m}, 0.20\,\text{m}]^T$.

Furthermore, for the Delphi radar – camera pair, we obtained the following results:

Table 5
Monte Carlo analysis results for Continental radar–LiDAR calibration.

	Reprojection error optimization	RCS optimization
${}^{r}p_{l,x}$ [m]	$\mathcal{N}(-0.050,\ 2.36 \times 10^{-5})$	
${}^{r}p_{l,y}$ [m]	$\mathcal{N}(-0.134,\ 8.04 \times 10^{-5})$	
${}^{r}p_{l,z}$ [m]	$\mathcal{N}(0.113,\ 8.46 \times 10^{-4})$	$\mathcal{N}(0.204,\ 1.48 \times 10^{-7})$
${}^{l}_{r}\theta_z$ [°]	$\mathcal{N}(-2.21,\ 1.61 \times 10^{-2})$	
${}^{l}_{r}\theta_y$ [°]	$\mathcal{N}(5.29,\ 1.29)$	$\mathcal{N}(4.81,\ 2.32 \times 10^{-5})$
${}^{l}_{r}\theta_x$ [°]	$\mathcal{N}(-1.63,\ 3.39 \times 10^{-1})$	$\mathcal{N}(-0.81,\ 4.29 \times 10^{-5})$

Table 6
Monte Carlo analysis results for Continental radar – camera calibration.

	Reprojection error optimization	RCS optimization
${}^{r}p_{c,x}$ [m]	$\mathcal{N}(0.039,\ 2.37 \times 10^{-5})$	
${}^{r}p_{c,y}$ [m]	$\mathcal{N}(-0.148,\ 1.96 \times 10^{-4})$	
${}^{r}p_{c,z}$ [m]	$\mathcal{N}(-0.051,\ 1.48 \times 10^{-3})$	$\mathcal{N}(0.043,\ 7.48 \times 10^{-7})$
${}^{c}_{r}\theta_z$ [°]	$\mathcal{N}(0.12,\ 4.44 \times 10^{-2})$	
${}^{c}_{r}\theta_y$ [°]	$\mathcal{N}(6.21,\ 2.91)$	$\mathcal{N}(5.60,\ 1.65 \times 10^{-4})$
${}^{c}_{r}\theta_x$ [°]	$\mathcal{N}(-2.11,\ 3.26 \times 10^{-1})$	$\mathcal{N}(-1.57,\ 4.07 \times 10^{-4})$

- ${}^{c}\hat{\boldsymbol{c}}_r = [0.02\,\text{m}, 0.11\,\text{m}, -0.01\,\text{m}, -0.1°, 4.7°, 7.3°, 0.11\,\text{m}]$
- ${}^{c}\hat{\boldsymbol{c}}_\sigma = [0.02\,\text{m}; 2.6°, -0.8°, -0.27\,\text{dBm}^2\,\text{deg}^{-2}, 17.4\,\text{dBm}^2]$
- ${}^{r}\tilde{\boldsymbol{p}}_c = [0.00\,\text{m}, 0.12\,\text{m}, 0.05\,\text{m}]^T$.

Results for the Continental radar showed that there is no significant difference between calibration of LiDAR–radar pair and camera–radar pair; therefore, for brevity, we focus on the results of LiDAR – radar calibration. Reprojection error histograms exhibited similar results to the case of the Continental radar when comparing reprojection error optimization, RCS optimization, and 2D reprojection optimization results; hence, they are not repeated here. However, an interesting effect was noticed when comparing the reprojection error optimization estimating the range bias with the reprojection error optimization omitting the bias. Fig. 13 shows that when the bias is included, average reprojection error per correspondence is reduced from 0.056 m to 0.045 m. On the other hand, for the case of the Continental radar, estimation of the bias compromised the results when calibrating the radar with a 3D sensor and was not able to reduce the average reprojection error. It can be concluded that with the Delphi radar, an actual bias is

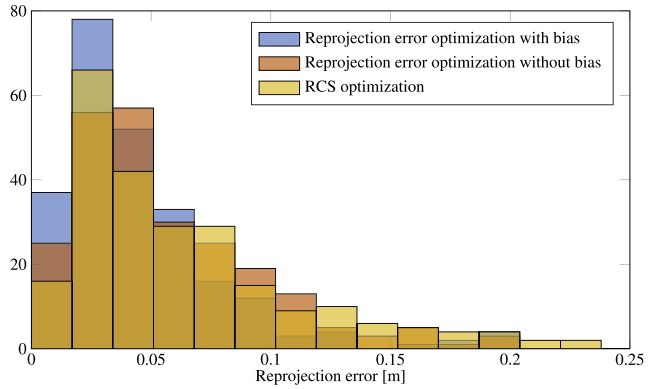


Fig. 13. Histogram of reprojection errors for two types of reprojection error optimization and RCS optimization for Delphi radar–LiDAR calibration.

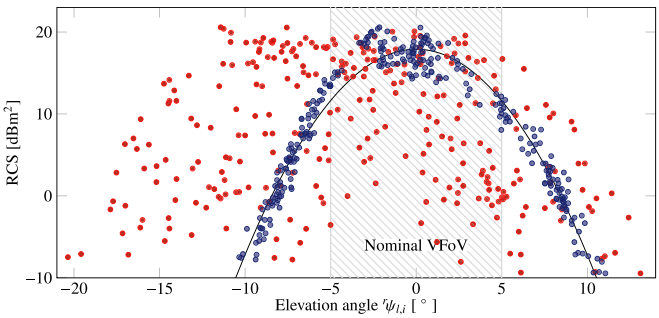


Fig. 14. RCS distribution across radar's VFoV for Delphi radar–LiDAR calibration. Red: reprojection error optimization; blue: RCS optimization.

present, most likely due to the target design, and the method is able to converge to a local minimum of a significantly lower cost.

Success of the RCS optimization is most evident in Fig. 14, where we can see a significant difference in the RCS distribution after two steps of optimization. The origin of this mismatch is convergence to poor values of the less certain parameters in reprojection error optimization. From the figure, one could conclude that there is no pattern in the data after the reprojection error optimization. However, the RCS optimization is able to find the same quadratic pattern as with the Continental radar without significantly degrading the reprojection error, as seen in Fig. 13.

Finally, Fig. 15 and Table 7 present results for the Monte Carlo analysis. The results are similar to those of the Continental radar, although a slight increase in variance can be seen in the estimation of ${}^{r}p_{l,x}$. The cause for the increase could be the performance of the radar or the coupling of the range and bias estimation.

5.4. Radar vertical alignment

In Section 1 we outlined the importance of proper radar vertical alignment, and in this section we present a simple, yet reliable method for its assessment. The proposed method requires precise 6DoF extrinsic calibration. Thus, we compared our method, labelled *RCS*, to the other 6DoF calibration method [26], labelled *MAN*, which manually searches for RCS maximums and artificially assigns zero elevation angle to these radar measurements. From Fig. 9, we can see that the measured target reports RCS in the range of $[16, 19]$ dBm^2 at the zero elevation angle. Therefore, for the *MAN* method, we used only the correspondences that surpass the RCS threshold of $\sigma_{th} = 16$ dBm^2, resulting in 32 correspondence groups total. Even though the low number of correspondence groups

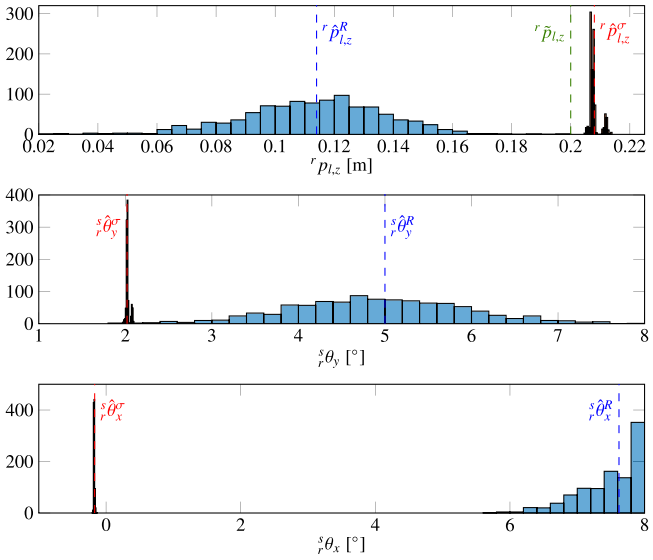


Fig. 15. Monte Carlo analysis results for Delphi radar–LiDAR calibration. Blue: calibration after reprojection error optimization; red: with RCS optimization.

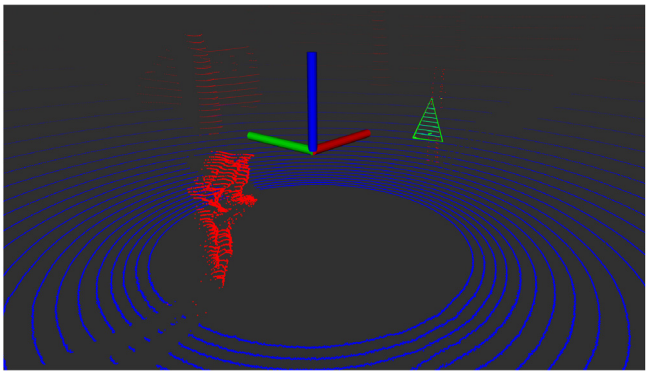


Fig. 16. Detected ground plane (blue), target(green) and environment clutter (red) with LiDAR in the purposes of vertical misalignment test.

Table 7
Monte Carlo analysis results for Delphi radar–LiDAR calibration.

	Reprojection error optimization	RCS optimization
${}^{r}p_{l,x}$ [m]	$\mathcal{N}(-0.064,\ 8.75 \times 10^{-5})$	
${}^{r}p_{l,y}$ [m]	$\mathcal{N}(0.132,\ 2.70 \times 10^{-5})$	
${}^{r}p_{l,z}$ [m]	$\mathcal{N}(0.113,\ 5.79 \times 10^{-4})$	$\mathcal{N}(0.208,\ 3.08 \times 10^{-6})$
${}^{l}_{r}\theta_z$ [°]	$\mathcal{N}(-2.93,\ 5.98 \times 10^{-3})$	
${}^{l}_{r}\theta_y$ [°]	$\mathcal{N}(4.92,\ 1.01)$	$\mathcal{N}(2.02,\ 5.09 \times 10^{-4})$
${}^{l}_{r}\theta_x$ [°]	$\mathcal{N}(7.50,\ 2.22 \times 10^{-1})$	$\mathcal{N}(-0.18,\ 4.85 \times 10^{-5})$
$\Delta^{r}r_r$ [m]	$\mathcal{N}(0.101,\ 5.79 \times 10^{-5})$	

available from the experiment could affect the accuracy of the *MAN* method, we can see that only a small fraction of measurements could be used in the optimization, thus leading to an expensive calibration data collection.

To find the vertical misalignment we drove the robot and detected the ground plane using a LiDAR, as illustrated by Fig. 16. We used LiDAR for simplicity, but the ground plane can also be found using a single camera as well [44]. The estimated ground plane normals ${}^{l}\boldsymbol{n}_{gp}$ from a 2-minute drive were averaged to remove the effects of uneven ground and robot rotation. The averaged normal ${}^{l}\boldsymbol{n}_{gp}$ was transformed to the radar ${}^{r}\boldsymbol{n}_{gp}$ coordinate frame using the estimated LiDAR to radar extrinsic calibration (for both the *RCS* and *MAN* method). Finally, we expressed the rotation between

Table 8
Monte Carlo analysis of vertical misalignment assessment for the Continental radar using LiDAR-s ground plane estimation with extrinsic calibration results.

	MAN	RCS
${}^r_g\theta_y[^\circ]$	$\mathcal{N}(-0.91,\ 8.86)$	$\mathcal{N}(-4.64,\ 3.40 \times 10^{-5})$
${}^r_g\theta_x[^\circ]$	$\mathcal{N}(6.40,\ 3.12)$	$\mathcal{N}(0.70,\ 3.69 \times 10^{-5})$

the ground and radar plane in the radar's coordinate frame. We set the arbitrary yaw angle around the ground plane normal to ${}^r_g\theta_z = 0^\circ$ and determined the pitch and roll angles ${}^r_g\theta_y$ and ${}^r_g\theta_x$, respectively. The method was tested using the same Monte Carlo analysis described in Section 5.3 through $N = 1000$ runs.

From Table 8, we can see that *MAN* method produced results with high uncertainty, which is inadequate for vertical misalignment assessment. Namely, the *MAN* method estimated pitch angles in the interval ${}^r_g\theta_y = [-11.12, 8.19]^\circ$ which surpasses common allowable vertical misalignments, thus providing unreliable misalignment correction guidelines. On the other hand, our method produced stable estimation of ground to radar plane angles, e.g. pitch angles in the interval ${}^r_g\theta_y = [-4.68, -4.62]^\circ$.

From the results, we can see the *RCS* method estimated vertical misalignment which surpasses allowable tolerance. Namely, Continental SRR20X user manual specifies allowable mounting pitch angle of $\pm 1^\circ$. To elaborate, vertical misalignment causes reduction in range and thus probability of detection. For instance, pitch misalignment of ${}^r_g\theta_y = 4.5^\circ$ causes a 25% decrease in range for the Delphi SRR2, while such misalignment causes decrease of 80% for the long range radar Delphi ESR. Therefore, radar mounting on a vehicle is a crucial step where our method can provide helpful guidelines. Given that, we conclude that for our sensor setup we should correct the orientation of the Continental radar according to results because the misalignment would impair the performance.

6. Conclusion

In this paper we have presented a method for extrinsic calibration of a LiDAR–camera–radar sensor system. A special calibration target design was developed to enable all the sensors to detect and accurately localize the target. The extrinsic calibration is based on the proposed two-step optimization procedure which involved: (i) optimization of a reprojection error based on the point–circle constraint which captures radar's lack of elevation angle measurements, and (ii) RCS optimization based on a pattern found in the radar's RCS estimation – again caused by the lack of the elevation angle resolution across substantial FoV thereof. Throughout the identifiability analysis, we have shown that the proposed point–circle geometric constraint requires minimum of 4 non-coplanar points to become identifiable, while the experimentally discovered effect of uneven uncertainty in the extrinsic parameters was confirmed by the FIM analysis. We presented the experimental results for LiDAR and camera sensors in combination with two radars from different manufacturers and have also addressed the radar vertical misalignment problem. In the end, through extensive experimental analysis, we have shown that the proposed method is able to accurately estimate all the six DoF of the extrinsic calibration.

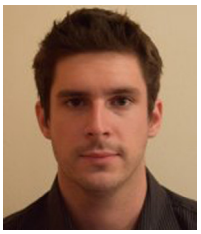
Acknowledgements

This work has been supported by the European Regional Development Fund under the project "System for increased driving safety in public urban rail traffic (SafeTRAM)". The research has been carried out within the activities of the Centre of Research Excellence for Data Science and Cooperative Systems supported by the Ministry of Science and Education of the Republic of Croatia.

References

[1] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, IEEE J. Robot. Autom. 3 (4) (1987) 323–344.
[2] Z. Zhang, A flexible new technique for camera calibration (technical report), IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2002) 1330–1334.
[3] D. Scaramuzza, A. Martinelli, R. Siegwart, A toolbox for easily calibrating omnidirectional cameras, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2006, pp. 5695–5701.
[4] J. Heikkila, O. Silven, A four-step camera calibration procedure with implicit image correction, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (1997) 1106–1112.
[5] B. Li, L. Heng, K. Koser, M. Pollefeys, A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 1301–1307.
[6] G. Atanacio-Jiménez, J.-J. González-Barbosa, J.B. Hurtado-Ramos, F.J. Ornelas-Rodríguez, H. Jiménez-Hernández, T. García-Ramirez, R. González-Barbosa, LIDAR velodyne HDL-64E calibration using pattern planes, Int. J. Adv. Robot. Syst. 8 (5) (2011) 70–82.
[7] N. Muhammad, S. Lacroix, Calibration of a rotating multi-beam Lidar, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, pp. 5648–5653.
[8] E. Fernández-Moral, J. González-Jiménez, V. Arévalo, Extrinsic calibration of 2D laser rangefinders from perpendicular plane observations, Int. J. Robot. Res. 34 (11) (2015) 1401–1417.
[9] Q.Z.Q. Zhang, R. Pless, Extrinsic calibration of a camera and laser range finder (improves camera calibration), in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2004, pp. 2301–2306.
[10] G. Pandey, J. McBride, S. Savarese, R. Eustice, Extrinsic calibration of a 3D laser scanner and an omnidirectional camera, in: IFAC Symposium on Intelligent Autonomous Vehicles, 2010, pp. 336–341.
[11] L. Zhou, Z. Deng, Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation, in: IEEE Intelligent Vehicles Symposium (IV), 2012, pp. 642–648.
[12] A. Geiger, F. Moosmann, O. Car, B. Schuster, Automatic camera and range sensor calibration using a single shot, in: IEEE Conference on Robotics and Automation (ICRA), 2012, pp. 3936–3943.
[13] F.M. Mirzaei, D.G. Kottas, S.I. Roumeliotis, 3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization, Int. J. Robot. Res. 31 (4) (2012) 452–467.
[14] E. Olson, AprilTag: A robust and flexible visual fiducial system, in: International Conference on Robotics and Automation (ICRA), 2011, pp. 3400–3407.
[15] J.L. Owens, P.R. Osteen, K. Daniilidis, MSG-cal: Multi-sensor graph-based calibration, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 3660–3667.
[16] M. Velas, M. Spanel, Z. Materna, A. Herout, Calibration of RGB Camera With Velodyne LiDAR, in: WSCG 2014 Communication Papers, 2014, pp. 135–144.
[17] K. Kwak, D.F. Huber, H. Badino, T. Kanade, Extrinsic calibration of a single line scanning lidar and a camera, in: IEEE International Conference on Intelligent Robots and Systems (ICRA), 2011, pp. 3283–3289.
[18] X. Wang, L. Xu, H. Sun, J. Xin, N. Zheng, On-road vehicle detection and tracking using MMW radar and monovision fusion, IEEE Trans. Intell. Transp. Syst. 17 (7) (2016) 2075–2084.
[19] J. Ćesić, I. Marković, I. Cvišić, I. Petrović, Radar and stereo vision fusion for multitarget tracking on the special Euclidean group, Robot. Auton. Syst. 83 (2016) 338–348.
[20] H. Cho, Y.-w. Seo, B.V.K.V. Kumar, R.R. Rajkumar, A multi-sensor fusion system for moving object detection and tracking in urban driving environments, in: IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1836–1843.
[21] D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter, K. Dietmayer, Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter, in: IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 1074–1081.
[22] R.O. Chavez-Garcia, O. Aycard, Multiple sensor fusion and classification for moving object detection and tracking, IEEE Trans. Intell. Transp. Syst. 17 (2) (2016) 525–534.
[23] T.D. Vu, O. Aycard, F. Tango, Object perception for intelligent vehicle applications: A multi-sensor fusion approach, in: IEEE Intelligent Vehicles Symposium (IV), 2014, pp. 774–780.
[24] E.F. Knott, Radar Cross Section Measurements, ITP Van Nostrand Reinhold, 1993.
[25] T. Wang, N. Zheng, J. Xin, Z. Ma, Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications, Sensors 11 (9) (2011) 8992–9008.

[26] S. Sugimoto, H. Tateda, H. Takahashi, M. Okutomi, Obstacle detection using millimeter-wave radar and its visualization on image sequence, in: International Conference on Pattern Recognition (ICPR), 2004, pp. 342–345.
[27] O. Schwindt, K. Buckner, B. Chakraborty, Systems and Methods for Radar Vertical Misalignment Detection, US 2016/0223649 A1, (2016).
[28] R. Hellinger, O.F. Schwindt, Automotive Radar Alignment, US 2017/0212215 A1, (2017).
[29] B.K. Park, K.K. Im, H. Chang Ahn, Alignment Method And System for Radar of Vehicle, US 9, 523, 769 B2, (2016).
[30] R. Hermann, A.J. Krener, Nonlinear controllability and observability, IEEE Trans. Autom. Control 22 (5) (1977) 728–740.
[31] F.M. Mirzaei, S.I. Roumeliotis, A Kalman-filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation., IEEE Trans. Robot. 24 (5) (2008) 1143–1156.
[32] J. Kelly, G.S. Sukhatme, Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration, Int. J. Robot. Res. 30 (1) (2011) 56–79.
[33] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, P. Furgale, Online self-calibration for robotic systems, Int. J. Robot. Res. 35 (4) (2015) 357–380.
[34] J. Peršić, I. Marković, I. Petrović, Extrinsic 6Dof Calibration of 3D LiDAR and Radar, in: European Conference on Mobile Robotics (ECMR), 2017, pp. 165–170.
[35] Y. Park, S. Yun, C.S. Won, K. Cho, K. Um, S. Sim, Calibration between color camera and 3D LIDAR instruments with a polygonal planar board, Sensors 14 (3) (2014) 5333–5353.
[36] S. Debattisti, L. Mazzei, M. Panciroli, Automated extrinsic laser and camera inter-calibration using triangular targets, in: IEEE Intelligent Vehicles Symposium (IV), 2013, pp. 696–701.
[37] C.G. Stephanis, D.E. Mourmouras, Trihedral rectangular ultrasonic reflector for distance measurements, NDT & E Int 28 (2) (1995) 95–96.
[38] P. Furgale, J. Rehder, R. Siegwart, Unified temporal and spatial calibration for multi-sensor systems, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 1280–1286.
[39] C. Jauffret, Observability and Fisher information matrix in nonlinear regression, IEEE Trans. Aerosp. Electron. Syst. 43 (2) (2007) 756–759.
[40] J.D. Stigter, J. Molenaar, A fast algorithm to assess local structural identifiability, Automatica 58 (2015) 118–124.
[41] T.J. Rothenberg, Identification in parametric models, Econometrica 39 (3) (1971) 577–591.
[42] K.S. Arun, T.S. Huang, S.D. Blostein, Least-squares fitting of two 3-D point sets, IEEE Trans. Pattern Anal. Mach. Intell. 9 (5) (1987) 698–700.
[43] G.H. Golub, C.F. Van Loan, Matrix Computations, Johns Hopkins University Press, 2013.
[44] J. Arrospide, L. Salgado, M. Nieto, R. Mohedano, Homography-based ground plane detection using a single on-board camera, IET Intell. Transp. Syst. 4 (2) (2010) 149.



Juraj Peršić, mag. ing was born on October 15th 1992. He received his B.Sc. degree in 2014 and M.Sc. degree in 2016, both in Electrical Engineering from Faculty of Electrical Engineering and Computing (FER Zagreb), University of Zagreb, Croatia. He has been employed as a researcher on the SafeTRAM project since September 2016 at FER. At the same time he became a Ph.D. student at FER under mentorship of prof. dr. sc. Ivan Petrović. His main areas of interest is mobile robotics with focus on sensor calibration and autonomous localization.

Ivan Marković is an Assistant Professor at the University of Zagreb Faculty of Electrical Engineering and Computing, Croatia (UNIZG-FER). He received the M.Sc. and Ph.D. degree in Electrical Engineering from the UNIZG-FER in 2008 and 2014, respectively. During his undergraduate and graduate studies he was awarded with the "INETEC" award (2007), "Josip Lončar" faculty award (2008), and with scholarship from the Croatian Ministry of Science and Education for the best students (2003–2008). In 2014 for his Ph.D. thesis he was awarded with the Silver Plaque "Josip Lončar" faculty award for outstanding doctoral dissertation and particularly successful scientific research. He is a member of the Institute of Electrical and Electronics Engineers (IEEE). He was a visiting researcher at INRIA Rennes-Bretagne Atlantique, Rennes, France, Lagadic group (Prof. François Chaumette). His research interests are mobile robotics, especially detection and tracking of moving objects and speaker localization.

Prof. Ivan Petrović (www.unizg.fer.hr/ivan.petrovic) is the Head of the Laboratory for Autonomous Systems and Mobile Robotics (http://lamor.fer.hr) and the Centre of Research Excellence for Advanced Cooperative Systems — ACROSS (http://across.fer.unizg.hr). He has more than 30 years of professional experience in R&D of automatic control theory and its applications. In the last fifteen years his research is focused on the advanced control and estimation techniques and their application in control and navigation of autonomous mobile robots and vehicles. He published about 50 journal papers and more than 180 conference papers. Results of his research effort have been implemented in several industrial products. He is a member of IEEE, IFAC – Vice Chair of TC on Robotics and FIRA – Executive committee. He is a member of the Croatian Academy of Engineering.

PUBLICATION 3

J. Peršić, L. Petrović, I. Marković and I. Petrović. Online multi-sensor calibration based on moving object tracking. *Advanced Robotics*, 35(3-4):130-140, 2021.

ADVANCED ROBOTICS
https://doi.org/10.1080/01691864.2020.1819874

FULL PAPER

Online multi-sensor calibration based on moving object tracking

J. Peršić, L. Petrović, I. Marković and I. Petrović

Faculty of Electrical Engineering and Computing, Laboratory for Autonomous Systems and Mobile Robotics (LAMOR), University of Zagreb, Zagreb, Croatia

ABSTRACT

Modern autonomous systems often fuse information from many different sensors to enhance their perception capabilities. For successful fusion, sensor calibration is necessary, while performing it online is crucial for long-term reliability. Contrary to currently common online approach of using ego-motion estimation, we propose an online calibration method based on detection and tracking of moving objects. Our motivation comes from the practical perspective that many perception sensors of an autonomous system are part of the pipeline for detection and tracking of moving objects. Thus, by using information already present in the system, our method provides resource inexpensive solution for the long-term reliability of the system. The method consists of a calibration-agnostic track to track association, computationally lightweight decalibration detection, and a graph-based rotation calibration. We tested the proposed method on a real-world dataset involving radar, lidar and camera sensors where it was able to detect decalibration after several seconds, while estimating rotation with 0.2° error from a 20 s long scenario.

ARTICLE HISTORY
Received 30 April 2020
Revised 15 July 2020
Accepted 19 August 2020

KEYWORDS
Online calibration; moving object tracking; radar; lidar; camera

1. Introduction

Modern robotic systems such as autonomous vehicles (AV) usually operate in highly dynamic scenarios where the actions they take significantly impact the surrounding environment. In order to achieve autonomy, they have to reliably solve many complex tasks, such as environment perception, motion prediction, motion planning and control. Environment perception, as the first building block of the autonomy pipeline, provides input data for many complex components, such as simultaneous localization and mapping (SLAM), detection and tracking of moving objects (DATMO) and semantic scene understanding. To increase the accuracy and robustness of an autonomous system, environment perception is often based on fusion of information from multiple heterogeneous sensors, such as lidar, camera, radar, GNSS and IMU. Accurate sensor calibration is a prerequisite for successful sensor fusion.

The sensor calibration consists of finding the intrinsic, extrinsic and temporal parameters, i.e. parameters of individual sensor models, transformations between sensor coordinate frames and alignment of sensor clocks, respectively. There are numerous offline and online approaches to sensor calibration and they vary significantly based on the sensors involved. While the offline approaches rely on controlled environments or calibration targets to achieve accurate calibration, the online approaches use information from the environment during the regular system operation, thus enabling long term robustness of the autonomous system. In this paper, we focus on the online calibration methods which are applicable for lidar–camera–radar sensor systems.

The online calibration methods can be roughly divided into feature-based and motion-based methods. Feature-based methods rely on extracting informative structure from the environment to generate correspondences between the sensors. These methods are limited to camera–lidar calibration, since other existing sensors do not provide enough structural information. For instance, extrinsic camera–lidar calibration can be based on line features detected as intensity edges in the image and depth discontinuities in the point cloud [1, 2]. Alternatively, the intensity of signal returned by lidar was used in [3] to find extrinsic calibration by maximizing the mutual information between images from camera and projected intensity values measured by the lidar. Recently, Park et al. [4] proposed a method for extrinsic and temporal camera–lidar calibration based on 3D point features in the environment. When the sensors do not provide enough structural information (e.g. radar), online calibration can be solved by depending on either the ego-motion or motion of objects in the environment. The

CONTACT J. Peršić juraj.persic@fer.hr

former come with the advantage that the sensors do not have to share a common field of view (FOV), while the latter also work with a static sensor systems. In [5] authors proposed an ego-motion based calibration suitable for camera–lidar calibration, while Kellner et al. [6] proposed a solution for radar odometry and alignment with the thrust axis of the vehicle. Furthermore, Kummerle et al. [7] proposed simultaneous calibration, localization and mapping framework which enables both extrinsic calibration and estimation of the robot kinematic parameters. Recently, Giamou et al. [8] proposed a solution for globally optimal ego-motion based calibration. Tracking-based methods have mostly been employed in static homogeneous sensor systems. To calibrate multiple stationary lidars, Quenzel et al. [9] relied on tracking of moving objects, while Glas et al. [10, 11] used human motion tracking. Human motion was also used for a stationary camera calibration [12, 13]. Considering tracking-based calibration of stationary heterogeneous sensors, Glas et al. [14] proposed a method for calibration of multiple 2D lidars and RGB-D cameras, while Schöller et al. [15] proposed a method for stationary camera–radar calibration.

Within the context of an AV, a sensor system consists of multiple lidars, cameras, radars and other sensors. While it is sufficient to use only a subset of sensors for accurate ego-motion estimation, DATMO is often performed using all the available exteroceptive sensors to provide a greater FOV coverage, robustness to adverse conditions and to increase the accuracy [16–18]. Several datasets have been recently made public by both the industry and the academia to emphasize importance and accelerate research on DATMO [17, 19–21]. In this paper, we leverage current state of the art in DATMO and propose an online calibration method based on it. Our motivation is to enable decalibration detection and recalibration based on the information which is already present in an autonomous system pipeline without adding significant computational overhead. To the best of the authors' knowledge, this is a first online calibration method that is based on heterogeneous sensor DATMO on a moving platform. In addition, while several target-based methods for calibration of radar–lidar–camera systems exists [22, 23], this is the first attempt to calibrate these sensors simultaneously in an online setting.

Our method provides a full pipeline which includes: (i) DATMO algorithm for each sensor modality, (ii) track-to-track association based on a calibration invariant measure, (iii) efficient decalibration detection and (iv) a graph-based calibration handling multiple heterogeneous sensors simultaneously. We point out that our method estimates only rotational component of the extrinsic calibration, because translation is unobservable due to limited sensor accuracy and a bias in detections (e.g. radar might measure a metal rear axle, while lidar detections report center of a bounding box. Even the methods based on ego-motion would struggle estimating translation on an AV, because they require motion which excites at least two rotational axis [24]. However, in contrast to rotational decalibration, feasible translational decalibration would not have a significant impact on the system performance. For instance, if rotational decalibration existed, object detection fusion would experience a growing error in the position with the increase in object distance, while translational decalibration would only introduce a position error of equal value. Our method assumes that translational calibration is obtained using either target-based or sensor-specific methods. The presented approach was evaluated on the nuScenes dataset [17], but is not in any way limited to this specific sensor setup. However, for testing our method, it is currently the only dataset containing appropriate data from cameras, radars and a lidar, which are sensors in the focus of the proposed calibration method.

2. Proposed method

In this section, we present each element of the method pipeline illustrated in Figure 1. The pipeline starts with the object detection which is specific for each sensor. Afterwards, the detections are tracked with separate trackers for each sensor which slightly differ among sensor modalities to accommodate their specifics. The confirmed tracks of different sensors are then mutually associated using calibration invariant measures. Each aforementioned stage has built-in outlier filtering mechanisms to prevent degradation of the results of subsequent steps. With the associated tracks, we proceed to a computationally lightweight decalibration detection. Finally, if decalibration is detected, we proceed to the graph-based sensor calibration. The method handles asynchronous sensors by assuming temporal correspondence between sensor clocks is known and performing linear interpolation of the object positions. Throughout the paper, we use the following notation: world frame $\mathcal{F}_w$, ego-vehicle frame $\mathcal{F}_e$ and *i-th* sensor frame $\mathcal{F}_i$. For convenience, we choose one sensor to be aligned with $\mathcal{F}_e$. In the case of the nuScenes sensor setup, we chose the top lidar as it shares FOV segments with all the other sensors.

2.1. Object detection

The proposed pipeline starts with object detection performed for the each sensor individually. Automotive radars usually provide object detections obtained from

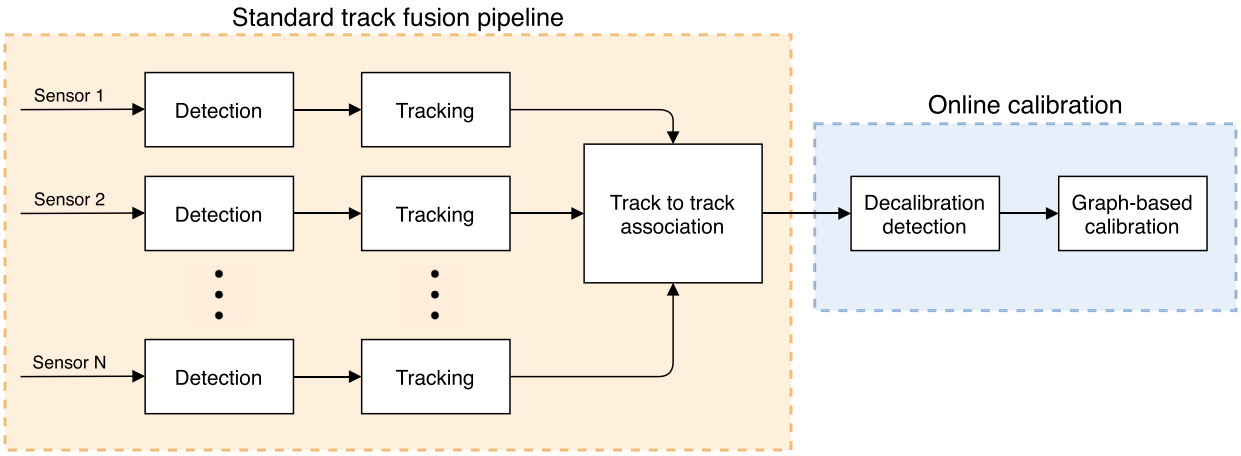


Figure 1. Illustration of the proposed pipeline for calibration based on DATMO. Detection, tracking and track to track association are commonly parts of a track fusion pipelines. However, our association criterion is oriented towards being calibration agnostic. Thereafter, we propose two new modules: decalibration detection and graph-based calibration.

proprietary algorithms performed locally on the sensor, while most can also provide tracked measurements. Obtaining the raw data is not possible due to low communication bandwidth of the CAN bus, typically used by these sensors. Nevertheless, radars provide a list of detected objects consisting of the following measured information: range, azimuth angle, range-rate, and radar cross-section (RCS). We use these detections and classify them as moving or stationary based on the range-rate. Furthermore, to avoid the need for extended target tracking where one target can generate multiple measurements, we perform clustering of close detections. These clusters are forwarded to the radar tracking module.

Contrary to the radar, lidar's and camera's raw data provides substantial information from which object detection is required. To extract detections from the lidar's point cloud, we used the *MEGVII* network based on sparse 3D convolution proposed by Zhu et al. [25] which is currently the best performing method for object detection on the nuScenes challenge. The method works by accumulating 10 lidar sweeps into a single one to form a dense point cloud input, thus reducing the effective frame rate of the sensor by a factor of 10. As the output, the network provides 3D position of objects as well as their size, orientation, velocity, class and detection score. Finally, for the object detection from images, we rely on a state-of-the-art 3D object detection approach dubbed *CenterNet* [26]. The output of *CenterNet* is similar to the lidar detections output, except that the velocity information is not provided since detections are based on a single image. We used the network weights trained on the KITTI dataset and determined the range scale factor by comparing *CenterNet* detections to the *MEGVII* detections. At this stage, outlier filtering was based on the detection score threshold.

2.2. Tracking of moving objects

Tracking modules for individual sensors take detections from the previous step as inputs, associate them between different time frames and provide estimates of their states, which are later used as inputs for subsequent steps. Since tracking is sensor specific, we perform it in each respective coordinate frame $\mathcal{F}_i$. We adopt a similar single-hypothesis tracking strategy for all the sensors, following the nuScenes baseline approach [27]. Assigning detections to tracks is done by using a global nearest neighbor approach and the Hungarian algorithm which provides efficient assignment solution [28]. The assignment is tuned by setting a threshold which controls the likelihood of a detection being assigned to a track. The state estimation of individual tracks is provided by an Extended Kalman filter which uses a constant turn-rate and velocity motion model [29]. Thus, the state vector in the lidar and camera tracker is

$$\boldsymbol{x}_k = [x_k\, y_k\, z_k\, \dot{x}_k\, \dot{y}_k\, \dot{z}_k\, \omega_k]^T, \tag{1}$$

with the state transition defined as

$$\boldsymbol{x}_{k+1} = \begin{pmatrix} 1 & 0 & 0 & \frac{\sin(\omega_k T)}{\omega_k} & -\frac{1-\cos(\omega_k T)}{\omega_k} & 0 & 0 \\ 0 & 1 & 0 & \frac{1-\cos(\omega_k T)}{\omega_k} & \frac{\sin(\omega_k T)}{\omega_k} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & \cos(\omega_k T) & -\sin(\omega_k T) & 0 & 0 \\ 0 & 0 & 0 & \sin(\omega_k T) & \cos(\omega_k T) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\times \boldsymbol{x}_k + \begin{pmatrix} \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 & 0 \\ 0 & 0 & \frac{T^2}{2} & 0 \\ T & 0 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{w}, \tag{2}$$

where $\boldsymbol{w} = [w_x\, w_y\, w_z\, w_\omega]^T$ is white noise on acceleration and turn-rate, while T is sensor sampling time.

Using object position measurements forms the measurement model defined as

$$\boldsymbol{y}_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \boldsymbol{x}_k + \boldsymbol{v}, \tag{3}$$

where $\boldsymbol{v} = [v_x\, v_y\, v_z]^T$ is white measurement noise.

Due to the lack of radar's elevation angle measurement, we drop the position and velocity in the z-direction, thus reducing the state vector for the radar tracker to

$$\boldsymbol{x}_k = [x_k\, y_k\, \dot{x}_k\, \dot{y}_k\, \omega_k]^T \tag{4}$$

with adjusted state transition function and measurement model.

Track management is based on the track history, i.e. the track is confirmed after N_{birth} consecutive reliable detections and removed after N_{coast} missing detections. All parameters are tuned for each sensor separately as they have significantly different frame rates and accuracies. Lastly, subparts of individual tracks that exhibit sudden changes in velocity are marked as unreliable and these time instants are excluded from the subsequent steps.

2.3. Track-to-track association

Track to track association has been previously studied and a common approach is based on the history and distance of track positions [30]. Contrary to the traditional approaches, we do not assume a perfect calibration, as decalibration could degrade the association. Thus, we observe two criteria for each track pair candidates through their common history: (i) mean of the velocity norm difference and (ii) mean of the position norm difference. The track pair has to satisfy both criteria and not surpass predefined thresholds. If multiple associations are possible, none of them are associated. This conservative approach helps in eliminating wrong association which would compromise the following calibration steps. However, the remaining tracks can be associated with more common association metrics (e.g. Euclidean or Mahalanobis distance) and used within a track fusion module. In our method we can use such a conservative approach and discard some track associations, since our goal is sensor calibration and not safety critical online DATMO for vehicle navigation.

The position norm is not truly calibration independent, as it is affected by both the measurement bias in the individual sensor and the translation between the sensors. Thus we use it in a loose way solely to distinguish between clearly distant tracks, i.e. we rely on the previously calibrated translational parameters and use a high threshold. On the other hand, velocity norm has already been used in a stationary system calibration for track association [14] as well as for frame-invariant temporal calibration of the sensors [31]. In a stationary scenario, it is trivial that velocity norm measured from different reference frames is equal. However, with a moving sensor platform which experiences both translational and rotational movement, this insight may not be that trivial. Namely, if a rigid body has non-zero angular velocity, different points on it will experience different translational velocities due to the lever arm. To state this more formally, we present the following proposition

Proposition 2.1. *Translational velocity norm of moving objects estimated from two reference frames $\mathcal{F}_1$ and $\mathcal{F}_2$ on the same rigid body is invariant to the transform between the frames and the motion of the rigid body.*

Proof: Let ${}^w\boldsymbol{p}_k$ be the position of the observed object at time k in the $\mathcal{F}_w$. Then, let ${}^1\boldsymbol{p}_k$ and ${}^2\boldsymbol{p}_k$ be the same position expressed in the sensor reference frames $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively:

$${}^1\boldsymbol{p}_k = {}^1_w\boldsymbol{R}_k \cdot {}^w\boldsymbol{p}_k + {}^1_w\boldsymbol{t}_k, \tag{5}$$

$${}^2\boldsymbol{p}_k = {}^2_1\boldsymbol{R} \cdot {}^1\boldsymbol{p}_k + {}^2_1\boldsymbol{t}, \tag{6}$$

where we express the motion of the rigid body $({}^1_w\boldsymbol{R}_k, {}^1_w\boldsymbol{t}_k)$ as time-varying $SE(3)$ transform, while the transform between sensors frames (i.e. calibration) is constant in time $({}^2_1\boldsymbol{R}, {}^2_1\boldsymbol{t})$. Let us now observe displacement of the moving object in the two sensor frames, ${}^1\delta p$ and ${}^2\delta p$, between two discrete time instances k and l:

$${}^1\delta\boldsymbol{p} = {}^1\boldsymbol{p}_k - {}^1\boldsymbol{p}_l, \tag{7}$$

$${}^2\delta\boldsymbol{p} = {}^2\boldsymbol{p}_k - {}^2\boldsymbol{p}_l = {}^2_1\boldsymbol{R}({}^1\boldsymbol{p}_k - {}^1\boldsymbol{p}_l) = {}^2_1\boldsymbol{R}^1\delta\boldsymbol{p}. \tag{8}$$

Since the rotation matrix is orthogonal, the norm of displacement is equal, i.e. $||{}^2\delta\boldsymbol{p}|| = ||{}^2_1\boldsymbol{R}^1\delta\boldsymbol{p}|| = ||{}^1\delta\boldsymbol{p}||$. Thus, the translational velocity norm is also equal because it is simply the ratio of the above displacements over the time difference $k-l$. ■

2.4. Decalibration detection

In a standard track fusion pipeline, track associations from the previous step are commonly used in object state estimate fusion. However, fusion depends on the accuracy of sensor calibration which can change over time due to disturbances. Thus, we propose a computationally inexpensive decalibration detection method, which is based on the data already present in the system. Similarly to the strategy presented by Deray et al. [32], we adopt a window-based approach for decalibration detection, but tailor the criterion we observe to accommodate the tracking-based scenario.

At the time instant t_k we form sets of corresponding track positions ${}^{i,j}\mathcal{S}_w = ({}^e\boldsymbol{x}_i, {}^e\boldsymbol{x}_j)$ that fall within the time window of length T_w ($t \in (t_k - T_w, t_k)$) for each sensor pair, where ${}^e\boldsymbol{x}_i$ and ${}^e\boldsymbol{x}_j$ represent stacked object positions obtained by i-th and j-th sensor, respectively. The positions are transformed from individual sensor frames $\mathcal{F}_i$ and $\mathcal{F}_j$ into the common reference frame $\mathcal{F}_e$ using the current calibration parameters. In the ideal case, the position should coincide, but due to the inevitable bias in the sensor measurements and the decalibration, in practice the error is always non-zero. To distinguish the error caused by bias from the decalibration error, we use an efficient closed-form solution for *orthogonal Procrustes problem* to obtain pairwise sensor calibrations [33]. Based on the ${}^{i,j}\mathcal{S}_w$, we form a 3×3 data matrix:

$$\boldsymbol{H} = ({}^e\boldsymbol{x}_i - {}^e\bar{\boldsymbol{x}}_i)({}^e\boldsymbol{x}_j - {}^e\bar{\boldsymbol{x}}_j)^T, \tag{9}$$

where ${}^e\bar{\boldsymbol{x}}_i$ and ${}^e\bar{\boldsymbol{x}}_j$ are means of corresponding sets.

The rotation ${}^j_i\boldsymbol{R}$ can be found using the singular-value decomposition (SVD):

$$[U, S, V] = \mathrm{SVD}(H), \tag{10}$$

$${}^j_i\boldsymbol{R} = VU^T. \tag{11}$$

Since the ${}^j_i\boldsymbol{R}$ should be an identity matrix in the ideal case, we define the decalibration criterion for the time instant t_k as an angle of rotation in the angle-axis representation by

$$J_k = \arccos\left(\frac{\mathrm{Tr}({}^j_i\boldsymbol{R}) - 1}{2}\right). \tag{12}$$

When the criterion (12) surpasses a predefined threshold, the system proceeds to the complete graph-based sensor calibration. The magnitude of the minimal decalibration that can be detected is limited by the predefined threshold and the horizon defined with the T_w. Longer horizon enables detection of smaller calibration changes, but with slower convergence.

2.5. Graph-based extrinsic calibration

The last step of the pipeline estimates the extrinsic parameters when the system detects decalibration. As previously mentioned, we handle only rotational decalibration due to the limited accuracy and the bias in the measurements. Since we are dealing with more than two sensors, pairwise calibration would produce inconsistent transformations among the sensors. Thus, we rely on the graph-based optimization presented in [34]. However, to ensure and speed up the convergence, we use the results of the previous step as an initialization. In the graph-based multi-sensor calibration paradigm, one sensor is chosen as an anchor and aligned with the $\mathcal{F}_e$ for convenience. We then search for the poses of other sensors with respect to the anchor sensor by minimizing the following criterion:

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \sum_{i,j}^{i\neq j} \sum_{k=1}^{N_{ij}} \boldsymbol{e}_{i,j,k}^T \cdot \boldsymbol{\Omega}_{i,j,k} \cdot \boldsymbol{e}_{i,j,k} \tag{13}$$

$$\boldsymbol{e}_{i,j,k} = {}^i\boldsymbol{p}_{i,k} - ({}^i_j\boldsymbol{R}(\boldsymbol{\phi}){}^j\boldsymbol{p}_{j,k} + {}^i\boldsymbol{t}_j) \tag{14}$$

where ϕ is a set of non-anchor sensor rotation parametrizations and N_{ij} is the number of corresponding measurements between the $i{-}th$ and $j{-}th$ sensor. To enable integration of the noise from both sensors, we follow the total least squares approach presented in [35] and define the noise model as:

$$\boldsymbol{\Omega}_{i,j,k} = ({}^i_j\boldsymbol{R}(\boldsymbol{\phi})V[{}^j\boldsymbol{p}_{j,k}]{}^i_j\boldsymbol{R}^T(\boldsymbol{\phi}) + V[{}^i\boldsymbol{p}_{i,k}])^{-1} \tag{15}$$

where $V[\cdot]$ is an observation covariance matrix of the zero-mean Gaussian noise.

Additionally, if a sensor does not have a direct link with the anchor sensor, we obtain ${}^i_j\boldsymbol{R}$ by multiplying the corresponding series of rotation matrices to obtain the final rotation between the *i-th* and *j-th* sensor. This approach enables the estimation of all parameters with a single optimization, while ensuring consistency between sensor transforms.

3. Experimental results

To validate the proposed method we used real world data provided with the nuScenes dataset [17]. Important details on the dataset, sensor setup and the scenario are given in Section 3.1, while Section 3.2 presents the results for each step of the calibration pipeline with greater attention on the introduced novelties related to calibration (Sec. 2.3–2.5).

3.1. Experimental setup

The nuScenes dataset consists of 1000 scenes that are 20 s long and collected with a vehicle driven through Boston and Singapore. The vehicle is equipped with a roof-mounted 3D lidar, 5 radars and 6 cameras. Each sensor modality has 360° coverage with small overlap of the sensors within the same modality. For clarity, in this experiment we focus only on measurements from the top lidar, front radar and front camera which all share a common FOV. The radar works at 13 Hz, camera 12 Hz, while the lidar provides point clouds with 20 Hz with the effective frame rate reduced to 2 Hz due to the object detector. The intrinsic and extrinsic calibration of all the sensors is obtained using several methods and calibration targets and is provided with the dataset. We considered it as a ground truth in the assessment of our method. Furthermore, we used isotropic and identical noise models for all the sensors involved.

In the following section, we present the results for *scene 343* from the dataset, because it contains variety of motions (cf. Figure 2). The scene is from the test subpart of the dataset for which the data annotations are not provided. The ego vehicle is stationary during the first 5 s, while afterward it accelerates and reaches a speed of around 40 km/h. Through the scene, total of 17 moving vehicles are driving in both the same and the opposite direction, while some of them make turns. In addition, the scene contains 8 stationary vehicles in the detectable area for all the sensors.

3.2. Results

We present the results sequentially for all the steps of the method as they progress through the pipeline illustrated in Figure 1. The starting point of the pipeline is object detection using individual sensors illustrated by Figure 3. Object detection using lidar and camera provided reliable results for the range of up to 50 m, both for moving and stationary vehicles. Rare false negatives did not cause significant challenges for the subsequent steps. In comparison to the camera, lidar provided significantly more detections with frequent false positives which we successfully filtered by setting a threshold on their detection scores. In addition, Figure 3(a) illustrates how the *MEGVII* network occasionally detects and classifies the same object as both car and truck, visible as blue and green box next to the ego-vehicle. However, these ambiguities were easily handled by the filtering within the tracking algorithm. In contrast to the other sensors, the radar provided many false positives and multiple detections of the same vehicles. We were able to extract only the moving vehicles based on the range rate, because it was difficult to differentiate stationary vehicles from the close-by surrounding buildings as they had the same range rate. Figure 3(c) illustrates radar detections colored with RCS, measured range-rate and confirmed radar tracks. It is clear that RCS is not a reliable measure for vehicle classification as it varies significantly across different vehicles due to their orientation, construction and other factors. On the other hand, the strongest reflections belong to the infrastructure and buildings, but the limited resolution prevents extracting fine structural information. Vehicles at closer range are usually detected as multiple objects, which was handled by the clustering algorithm. The range-rate provided useful information for classification of moving object, but the Figure 3(c) illustrates how cross-traffic vehicles impose greater challenge because their range-rate is closer to zero.

The previously described detections were used in the subsequent tracking step for each sensor. Counting only tracks longer than 2 s, radar extracted 18 (105 s), lidar 25 (177 s) and camera 18 (84 s) tracks with total duration given in the parentheses. A visual of both detections and tracks for each sensor is available in the accompanying video[1].

To test the track association, which is the first step of the pipeline, we hand-labeled the ground truth track

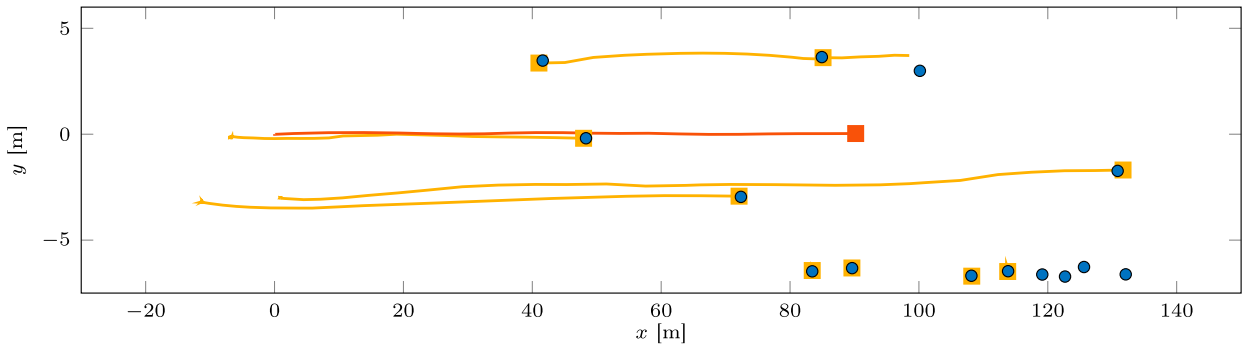
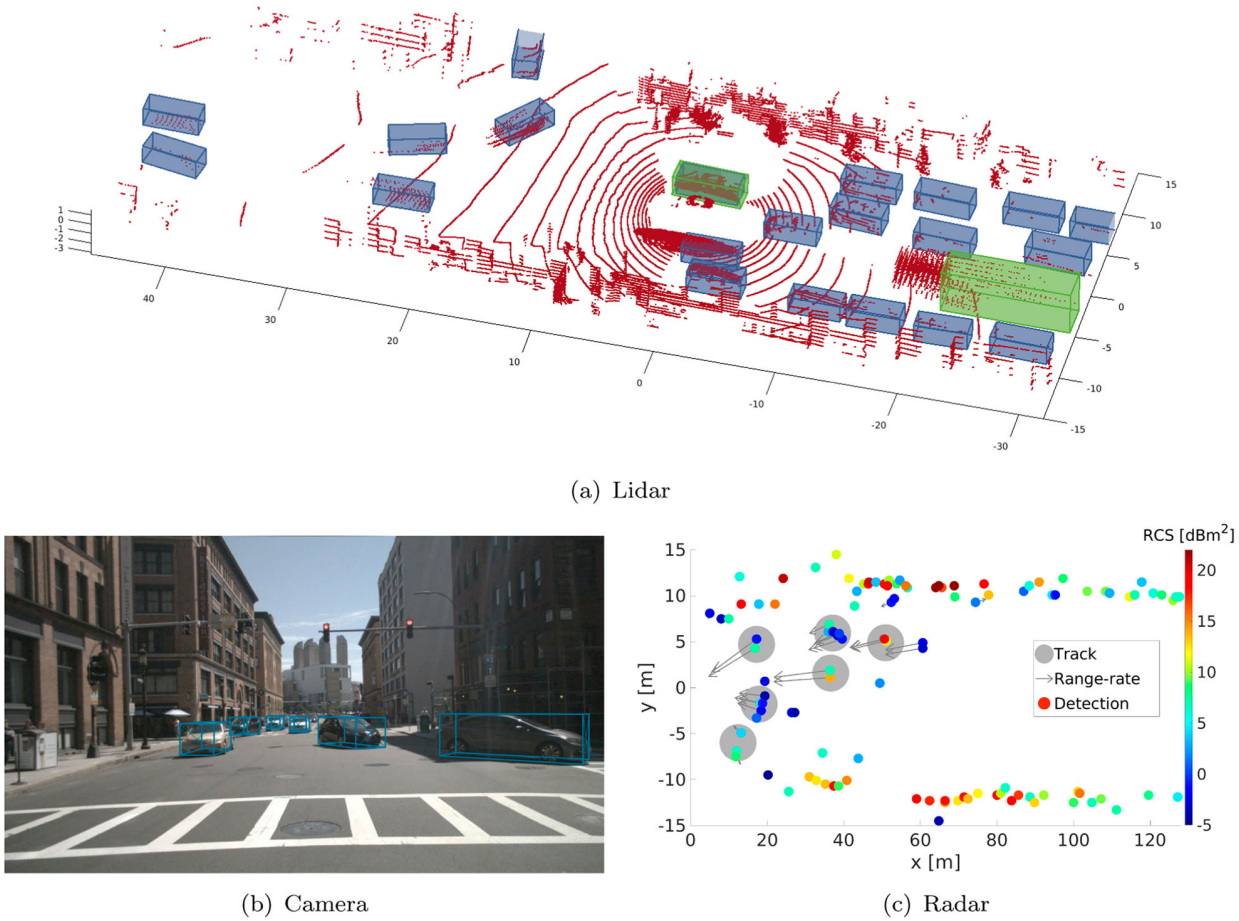


Figure 2. Illustration of the used scene showing ego-vehicle trajectory (red), lidar detections at $t = 19s$ (blue) and history of lidar tracks (yellow) (color in online).



(a) Lidar

(b) Camera

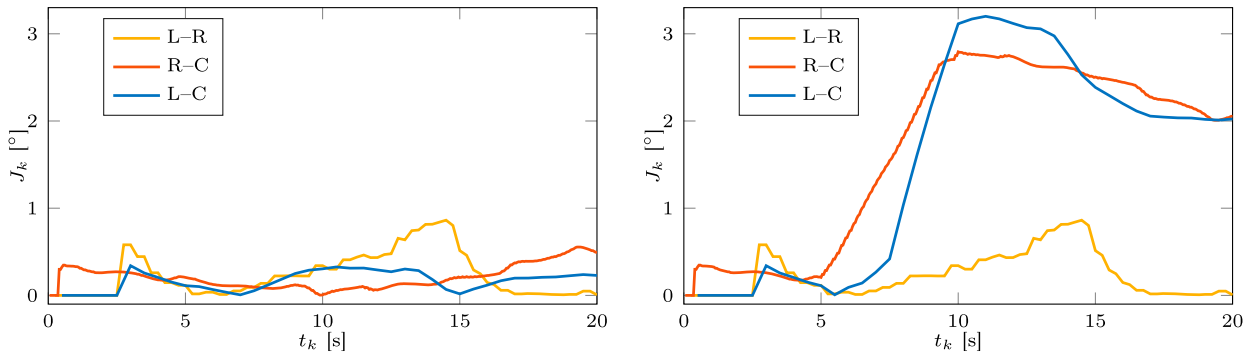
(c) Radar

Figure 3. Vehicle detection using lidar, camera and radar. Lidar pointcloud consists of 10 consecutive sweeps, while blue and green boxes represent car and truck detections, respectively. Radar detections are colored with radar cross-section and show range rate, while gray circles represent confirmed tracks. (a) Lidar (b) Camera. (c) Radar (color in online).

associations for each sensor pair by carefully observing the measurement data. The proposed method did not produce any false positive associations, while the success rate for each sensor pair was as follows: lidar–radar 93%; lidar–camera 94%; radar–camera 94%. An average time for two tracks to be associated after the tracking has started with both sensors was 1.5 s for every sensor combination. In addition, we note that introducing decalibration did not lead to any noticeable difference in results.

The following step, the decalibration detection, was tested in two scenarios as illustrated in Figure 4. The two scenarios examine the temporal evolution of the change detection criterion J_k using the same horizon $T_w = 5s$. The scenario with correct calibration throughout the scene (Figure 4(a)) shows that the criterion for each sensor pair is below 1° throughout the scene. The criterion varies mostly due to the number of correspondences between the sensors where the first part of the scene contains more moving objects than the second. In the second scenario (Figure 4(b)), we introduced an artificial decalibration of 3° in the yaw angle of the camera frame with respect to ego frame at the time instant $t_d = 5$ s. We can notice a significant increase in the criterion for the sensor pairs involving the camera, while the criterion for lidar–radar remained the same. Thus, besides detecting system decalibration, we were also able to assess which sensor changed its orientation by simply comparing the sensor-pairwise criteria.

Finally, we tested the extrinsic calibration by iteratively adding the available correspondences through the scene and running the calibration. Temporal evolution of the estimated calibration error is shown in the Figure 5. The results are presented as a difference of ground truth and estimated pairwise rotation expressed in Euler angles for intuitive assessment. For the calibration involving radar, we observe only the yaw angle. Namely, even for the target-based methods [23], the accuracy of the remaining angles is limited due to the lack of elevation measurements. We can notice a consistent convergence of the error towards zero as more correspondences are added with the following final



(a) Without decalibration (b) Decalibration at $t_k = 5$

Figure 4. Test of the decalibration criterion J_k for each sensor pair through the scene with horizon $T_w = 5$s. Figure 4(a) show the case with correct calibration, while the Figure 4(b) shows an example of introducing 3° error in camera yaw angle. Significant increase in the criteria for pairs involving camera clearly indicates its decalibration. (a) Without decalibration. (b) Decalibration at $t_k = 5$.

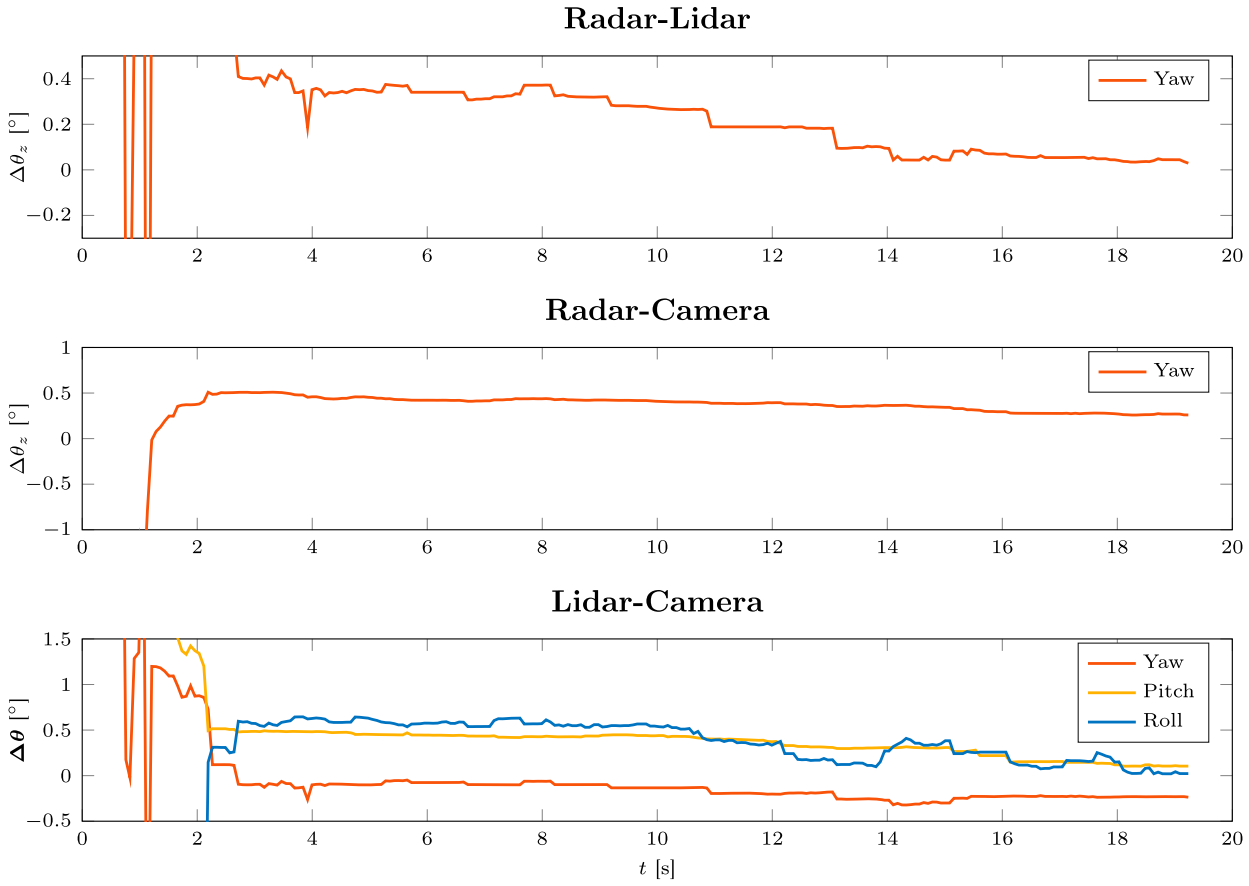


Figure 5. The plots show temporal evolution of the rotation calibration results by using available correspondences until the time $t = 20$ s. The errors are reported in Euler angles as the difference between the estimated and ground truth parameters obtained by target-based methods before the experiment. The results for sensor combination involving radar do not show pitch and roll as radar's missing elevation angle measurements prevent their accurate calibration.

errors: lidar–radar yaw $\Delta\theta_z = 0.03°$; radar–camera yaw $\Delta\theta_z = -0.26°$; lidar–camera yaw, pitch and roll $\Delta\Theta = (-0.24, 0.10, 0.02)°$. Additionally, the analysis provided good guidelines for determining the minimal horizon in the previous step. For this particular scene, at least 2 s are necessary to reduce the error down to 0.5°. Furthermore, to test the influence of graph-based optimization, we performed pairwise calibration for all three sensor combinations using the data from the whole scene. Compared to the ground truth calibration, errors for sensor

Table 1. Calibration results for three sensor combinations using pairwise (P) and graph (G) approaches.

	L–C		L–R		R–C	
	P	G	P	G	P	G
$\Delta\theta_z$ [°]	−0.20	−0.24	0.02	0.03	−0.42	−0.26
$\Delta\theta_y$ [°]	0.22	0.10	–	–	–	–
$\Delta\theta_x$ [°]	−0.28	0.02	–	–	–	–

pairs were: lidar–radar yaw $\Delta\theta_z = 0.02°$; radar–camera yaw $\Delta\theta_z = -0.42°$; lidar–camera yaw, pitch and roll $\Delta\Theta = (-0.20, 0.22, -0.28)°$. Comparison between pairwise and graph approaches is summarized in Table 1. While the magnitude of the error is similar to the joint graph optimization, pairwise calibration violates the consistency of the solution. Namely, rotational error of closing the loop, i.e. evaluating $\Delta R = {}^{l}_{c}R \cdot {}^{c}_{r}R \cdot {}^{r}_{l}R$, resulted with an error expressed in yaw, pitch and roll angles $\Delta\Theta = (-0.19, -0.18, -0.34)°$, while the problem of rotational error due to loop closing does not exist in the graph-based approaches.

3.3. Comparison with odometry-based calibration

To compare our method with an online calibration approach based on ego-motion, we tested the *SRRG* method proposed in [36]. The *SRRG* method is based on odometry constraints and can estimate vehicle odometry and extrinsic and temporal parameters of multiple sensors. However, we limit the comparison to lidar–camera calibration as these sensors can provide reliable 6DoF estimates of the vehicle ego-motion. For the lidar ego-motion, we used results of the map-based localization provided with the nuScenes dataset [17]. In the nuScenes dataset, only images from monocular cameras are available which prevents ego-motion estimation with correct scale, which is needed by *SRRG*. Therefore, we coupled the front camera images with the on-board IMU to obtain 6DoF odometry using the *Rovio* toolbox [37].

The chosen test scene presented significant challenges for the ego-motion methods because it included forward-only vehicle motion, which is usually the most common driving mode of vehicles. Namely, to achieve full observability, such methods usually require non-planar movement and excitation of at least two rotational axes [24]. This conclusion is also confirmed by our results, where translation in all the three axes and roll could not be estimated. For example, with a small perturbation in the initial calibration guess, the error in translation parameters reached 18 m, while the roll angle error reached 173°. This is not surprising, since these parameters are unobservable, but nevertheless they should not be estimated because they affect estimation accuracy of the observable parameters. Specifically, when estimating all 6DoF, error distributions of the yaw and pitch angles were $\mathcal{N}(-0.24°, 0.14°)$ and $\mathcal{N}(0.33°, 0.16°)$, respectively. On the other hand, when we locked the estimation of translational parameters by setting a prior to the ground truth values, we noticed a significant decrease in the standard deviation of the yaw and pitch angle error distributions $\mathcal{N}(-0.31°, 0.014°)$ and $\mathcal{N}(0.11°, 0.04°)$, respectively. However, the roll angle error was still significant with distribution $\mathcal{N}(24.11°, 0.961°)$. These results show that the proposed method yilded similar accuracy in the yaw and pitch angles as the odometry-based method, with an additional benefit of being able to estimate the roll angle on a dataset with forward-only motion.

4. Conclusion

In this paper we have proposed an online multi-sensor calibration method based on detection and tracking of moving objects. To the best of the authors' knowledge, this is the first method which calibrates radar–camera–lidar sensor system on a moving platform without relying on a known target. We proposed a complete pipeline for track based fusion which does not assume a constant and known sensor calibration. Proposed track to track association is based on a criterion resistant to decalibration, which is then followed by a decalibration detection relying on the information already present in the system without imposing significant computational burden. Finally, pairwise calibration provided by the decalibration detection module is used as an initialization for the final graph-based optimization which refines the results and provides consistent transformation across multiple frames. We validated the method on real world data from the nuScenes dataset which provides radar, lidar and camera measurements collected with a vehicle driving through an urban environment. The method was able to perform track association for calibration with high success rate and without wrong associations. Furthermore, it was able to detect decalibration within several seconds. Due to limited accuracy in position measurements, the method is currently limited to rotation calibration only. Nevertheless, it was able to estimate rotation parameters with an approximate error of 0.2° from a 20 s long scene.

For future work, we pan to explore the possibility of using the motion of moving objects for temporal calibration as well. We believe it could improve fusion since not all sensors, e.g. radar, can always be hardware synchronized. Additionally, a statistical analysis of individual sensor noises using the whole nuScenes and other datasets could further improve the results of the proposed method.

Note

1. https://youtu.be/MgqIs-d6hRM

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work has been supported by the European Regional Development Fund under the grants KK.01.1.1.01.0009 (DATACROSS),KK.01.2.1.01.0022 (SafeTRAM).

Notes on contributors

Juraj Peršić received his B.Sc. degree in 2014. and M.Sc. degree in 2016., both in electrical engineering from Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Croatia. He has been employed as a researcher on the SafeTRAM project since September 2016 at FER. At the same time he became a Ph.D. student at FER under mentorship of prof. dr. sc. Ivan Petrović. His main areas of interest is mobile robotics with focus on sensor calibration and fusion.

Luka Petrović received his B.Sc. and M.Sc. Degrees in electrical engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2015 and 2017, respectively. During his graduate studies, he was awarded with the Rector's Award (2016) for a practical application in the field of robotics and the Bronze Plaque 'Josip Lončar' faculty award (2017) for outstanding academic achievement. His main research interests are in the areas of autonomous systems and robotics with focus on high-dimensional motion planning.

Ivan Marković received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Zagreb, Croatia, in 2008 and 2014, respectively. He is an Associate Professor with University of Zagreb Faculty of Electrical Engineering and Computing, Croatia. He was a visiting researcher at INRIA RennesBretagne Atlantique, Rennes, France under the supervision of Prof. Franasois Chaumette. In 2018 he received the Croatian Academy of Engineering Young Scientist Award Vera Johanides. He also a member of the IFAC Technical Committee on Robotics. His research interests include estimation theory with applications to autonomous mobile robotic systems.

Ivan Petrović is a professor and the head of the Laboratory for Autonomous Systems and Mobile Robotics at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. His research interests include advanced control and estimation techniques and their application in autonomous systems and robotics. He published more than 60 journal and 200 conference papers. Among others, he is a full member of the Croatian Academy of Engineering, and the Chair of the IFAC Technical Committee on Robotics.

ORCID

J. Peršić http://orcid.org/0000-0001-8127-5433

References

[1] Levinson J, Thrun S. Automatic online calibration of cameras and lasers. Robotics: Science and Systems (RSS); 2013.
[2] Moghadam P, Bosse M, Zlot R. Line-based extrinsic calibration of range and image sensors. IEEE International Conference on Robotics and Automation (ICRA); 2013. p. 3685–3691.
[3] Pandey G, McBride JR, Savarese S, et al. Automatic extrinsic calibration of vision and lidar by maximizing mutual information. J Field Robotics. 2015;32(5):696–722.
[4] Park C, Moghadam P, Kim S, et al. Spatiotemporal camera-LiDAR calibration: a targetless and structureless approach. IEEE Robot Automat Lett. 2020;5(2):1556–1563. 2001.06175.
[5] Taylor Z, Nieto J. Motion-Based calibration of multimodal sensor extrinsics and timing offset estimation. IEEE Trans Robot. 2016;32(5):1215–1229.
[6] Kellner D, Barjenbruch M, Dietmayer K, et al. Joint radar alignment and odometry calibration. International Conference on Information Fusion; 2015. p. 366–374.
[7] Kümmerle R, Grisetti G, Burgard W. Simultaneous parameter calibration, localization, and mapping. Adv Robot. 2012;26(17):2021–2041.
[8] Giamou M, Ma Z, Peretroukhin V, et al. Certifiably globally optimal extrinsic calibration from per-Sensor egomotion. IEEE Robotics Automation Lett. 2019;4(2):367–374. 1809.03554.
[9] Quenzel J, Papenberg N, Behnke S. Robust extrinsic calibration of multiple stationary laser range finders. IEEE International Conference on Automation Science and Engineering (CASE); 2016. p. 1332–1339.
[10] Glas DF, Miyashita T, Ishiguro H, et al. Automatic position calibration and sensor displacement detection for networks of laser range finders for human tracking. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2010. p. 2938–2945.
[11] Glas DF, Ferreri F, Miyashita T, et al. Automatic calibration of laser range finder positions for pedestrian tracking based on social group detections. Adv Robot. 2014;28(9):573–588.
[12] Tang Z, Lin YS, Lee KH, et al. Camera self-calibration from tracking of moving persons. International Conference on Pattern Recognition (ICPR); 2017. p. 265–270.
[13] Jung J, Yoon I, Lee S, Paik J. Object detection and tracking-Based camera calibration for normalized human height estimation. J Sensors. 2016;0:00–99.
[14] Glas DF, Brscic D, Miyashita T, et al. SNAPCAT-3D: Calibrating networks of 3D range sensors for pedestrian tracking. IEEE International Conference on Robotics and Automation (ICRA); 2015. p. 712–719.
[15] Schöller C, Schnettler M, Krämmer A, et al. Targetless Rotational Auto-Calibration of Radar and Camera for Intelligent Transportation Systems. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC); 2019. p. 3934–3941. 1904.08743.
[16] Ćesić J, Marković I, Cvišić I, et al. Radar and stereo vision fusion for multitarget tracking on the special Euclidean group. Rob Auton Syst. 2016;83:338–348.

[17] Caesar H, Bankiti V, Lang AH, et al. nuScenes: a multimodal dataset for autonomous driving. 2019 March. arXiv preprint. 1903.11027.
[18] Wang X, Xu L, Sun H, et al. On-road vehicle detection and tracking using MMW radar and monovision fusion. IEEE Trans Intell Transp Syst. 2016;17(7):2075–2084.
[19] Pitropov M, Garcia D, Rebello J, et al. Canadian Adverse Driving Conditions Dataset. 2020. 2001.10117.
[20] Sun P, Kretzschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: an open dataset benchmark. 2019. 1912.04838.
[21] Chang MF, Lambert J, Sangkloy P, et al. Argoverse: 3D tracking and forecasting with rich maps. 2019. p. 8748–8757. 1911.02620.
[22] Domhof J, Kooij JFP, Gavrila DM. A multi-sensor extrinsic calibration tool for lidar, camera and radar. IEEE International Conference on Robotics and Automation (ICRA); 2019. p. 1–7.
[23] Peršić J, Marković I, Petrović I. Extrinsic 6DoF calibration of a radar-LiDAR-camera system enhanced by radar cross section estimates evaluation. Rob Auton Syst. 2019;114:00–00.
[24] Brookshire J, Teller S. Extrinsic calibration from per-sensor egomotion. Robotics: Science and systems (rss). 2012.
[25] Zhu B, Jiang Z, Zhou X, et al. Class-balanced grouping and sampling for point cloud 3D object detection. 2019. arXiv preprint. p. 1–8. 1908.09492.
[26] Zhou X, Wang D, Krähenbühl P. Objects as points. 2019. arXiv preprint. 1904.07850.
[27] Weng X, Kitani K. A baseline for 3D multi-object tracking. 2019. arXiv preprint. 1907.03961.
[28] Kuhn HW. The Hungarian method for the assignment problem. Naval Res Logistics Q. 1955;2(5):83–97.
[29] Rong Li X, Jilkov V. Survey of maneuvering target tracking. part I. dynamic models. IEEE Trans Aerospace Electron Syst. 2003;39(4):1333–1364.
[30] Houenou A, Bonnifait P, Cherfaoui V. A track-to-track association method for automotive perception systems. Intelligent Vehicles Symposium (iv); 2012. p. 704–710.
[31] Peršić J, Petrović L, Marković I, et al. Spatio-temporal multisensor calibration based on Gaussian processes moving object tracking. 2019. arXiv preprint. 1904.04187.
[32] Deray J, Sola J, Andrade-Cetto J. Joint on-manifold self-calibration of odometry model and sensor extrinsics using pre-integration. European Conference on Mobile Robots (ECMR); 2019. p. 1–6.
[33] Larusso A, Eggert D, Fisher R. A comparison of four algorithms for estimating 3-D rigid transformations. The British Machine Vision Conference; 1995. p. 237–246.
[34] Owens JL, Osteen PR, Daniilidis K. MSG-cal: Multi-sensor graph-based calibration. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2015. p. 3660–3667.
[35] Estépar RSJ, Brun A, Westin CF. Robust generalized total least squares. International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI); 2004. p. 234–241.
[36] Corte BD, Andreasson H, Stoyanov T, et al. Unified motion-based calibration of mobile multi-sensor platforms with time delay estimation. IEEE Robotics Automat Lett. 2019;4(2):902–909.
[37] Bloesch M, Burri M, Omari S, et al. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. Int J Robotics Res. 2017;36(10):1053–1072.

PUBLICATION 4

J. Peršić, L. Petrović, I. Marković and I. Petrović. Spatiotemporal Multisensor Calibration via Gaussian Processes Moving Target Tracking. *IEEE Transactions on Robotics*, Early Access, 2021.

Spatiotemporal Multisensor Calibration via Gaussian Processes Moving Target Tracking

Juraj Peršić, *Student Member, IEEE*, Luka Petrović, *Student Member, IEEE*, Ivan Marković, *Member, IEEE*, and Ivan Petrović, *Member, IEEE*

***Abstract*—Robust and reliable perception of autonomous systems often relies on fusion of heterogeneous sensors, which poses great challenges for multisensor calibration. In this article, we propose a method for multisensor calibration based on Gaussian processes (GPs) estimated moving target trajectories, resulting with spatiotemporal calibration. Unlike competing approaches, the proposed method is characterized by the following: first, joint multisensor on-manifold spatiotemporal optimization framework, second, batch state estimation and interpolation using GPs, and, third, computational efficiency with O(n) complexity. It only requires that all sensors can track the same target. The method is validated in simulation and real-world experiments on the following five different multisensor setups: first, hardware triggered stereo camera, second, camera and motion capture system, third, camera and automotive radar, fourth, camera and rotating 3-D lidar, and, fifth, camera, 3-D lidar, and the motion capture system. The method estimates time delays with the accuracy up to a fraction of the fastest sensor sampling time, outperforming a state-of-the-art ego-motion method. Furthermore, this article is complemented by an open-source toolbox implementing the calibration method available at bitbucket.org/unizg-fer-lamor/calirad.**

***Index Terms*—Gaussian processes (GPs), multisensor calibration, temporal calibration.**

I. Introduction

MODERN autonomous robotic systems navigate through the environment using information gathered by various sensors. To process the gathered information, robots must rely on accurate sensor models and often fuse information from multiple sensors to improve the performance. For sensor fusion, appropriate knowledge of both temporal and spatial relations between the sensors is required, which can be challenging when working with heterogeneous sensor systems, since sensors can operate based on various physical phenomena, while providing measurements asynchronously with different frame rates. The described challenges are addressed by sensor calibration, which can be divided into intrinsic, extrinsic also referred as spatial, and temporal calibration.

The intrinsic calibration is related to individual sensors as it provides parameters for sensor models. The task of the extrinsic calibration is to find homogeneous transforms relating multiple sensors, while temporal calibration aims to find relation between the individual sensor clocks.

The sensor calibration approach for a particular problem depends on multiple factors, e.g., the type of involved sensors, overlapping field of view, required degree of calibration accuracy, nevertheless, to calibrate multiple sensors extrinsically and temporally, we need to perform correspondence registration in the sensor data, which is later used to form an optimization criterion. The correspondences can originate from a designed target, yielding the target-based methods [1], [2], or from the environment itself, as in the case of the so-called targetless methods [3], [4]. For example, odometry-based methods are a special class of targetless methods suitable for online application and are based on leveraging the environment to estimate ego-motion and calibrate the multisensor system [5], [6]. The concept of sensor calibration by aligning trajectories of moving targets received most attention in the target-based calibration of depth sensors [7]–[9], and calibration of cameras, depth sensors, and lidars by exploiting human motion [10]–[13].

Specifically, to match trajectories between the sensors, the authors observe a similarity measure of the net velocity history profiles; however, in the optimization step, they rely only on the detected positions of the tracked people. In [14], authors propose to calibrate multiple 2-D lidars by tracking moving targets using a pose graph, wherein rotation is decoupled from translation by using a rotation averaging approach.

Temporal calibration of a sensor system requires motion, either of the observed target [7], [15] or the system itself [16]–[22].

Furthermore, some research advocates a unified approach to spatiotemporal calibration [17], while others claim that estimating uncorrelated quantities, such as time delay and homogeneous transforms, might degrade the final result [23]. Additional challenge in temporal calibration is computational complexity; namely, at each optimization step new correspondences need to be computed due to the new time-delay perturbation. Therefore, the common approach is to reduce the dimensionality of the problem and preferably remove correlation with the extrinsic calibration. In [7], authors tracked a colored sphere to perform spatiotemporal calibration of multiple Kinect v2 sensors. By performing principal component analysis on the trajectories,

Manuscript received July 30, 2020; revised November 19, 2020 and February 1, 2021; accepted February 15, 2021. This work was supported by the European Regional Development Fund under Grant KK.01.1.1.01.0009 (DATACROSS) and Grant KK.01.2.1.01.0022 (SafeTRAM). This article was recommended for publication by Associate Editor L. Carlone and Editor F. Chaumette upon evaluation of the reviewers' comments. *(Corresponding author: Juraj Peršić.)*

The authors are with the Faculty of Electrical Engineering and Computing, Laboratory for Autonomous Systems and Mobile Robotics, University of Zagreb, Zagreb 10000, Croatia (e-mail: juraj.persic@fer.hr; luka.petrovic@fer.hr; ivan.markovic@fer.hr; ivan.petrovic@fer.hr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2021.3061364.

Digital Object Identifier 10.1109/TRO.2021.3061364

they obtained field of view invariant one-dimensional kernels used in temporal calibration. Even though this method is applicable to other sensors, it assumes the same frame rate of the sensors and its resolution is limited to the sampling time. In [15], temporal calibration based on target tracking is presented where the authors use linear interpolation for continuous-time representation and position norm for the dimensionality reduction. The $AX = XB$ sensor calibration problem with unknown temporal correspondences was tackled in [19]. To perform dimensionality reduction, authors used one-dimensional invariants—displacement and angle of rotation—defined by Plücker coordinates of the screw motion. In [20], authors proposed an algorithm based on system motion by aligning curves in the 3-D orientation space. The temporal calibration problem was formulated as a registration task, which can be considered as a variant of the iterative closest point (ICP) algorithm. Temporal camera–lidar calibration using a hardware system based on LED display and photo diode is performed in [24].

The approach in [17] and [25] is similar to ours as it uses B-splines for continuous-time representation. Camera-IMU spatiotemporal calibration relies on estimator that

1) represents system's motion as a single continuous-time trajectory;
2) incorporates raw IMU measurements;
3) minimizes projection error of detected checkerboard corners with a camera.

In addition, method from [17] can include lidar in the calibration if environment has enough planar surfaces. In this article, we focus on the target tracking-based spatiotemporal calibration relying on continuous-time representation using Gaussian processes (GPs). Leveraging GPs enables a theoretically grounded batch state estimation and interpolation, while it has been a well-recognized tool in machine learning [26] both for regression and classification problems, and have been proposed for a variety of robotics challenges as well [27]. For example, in [28] and [29], mobile robot localization was a motivation for an efficient batch state estimation using GP regression, in [30], GPs have been used for efficient motion planning, being especially valuable in high-dimensional configuration spaces, while in [31], they were used for tracking of extended objects. Common alternative to GP regression are B-splines, often used for their computational efficiency. However, recent development of the GP regression [29] enabled comparable efficiency, while GPs provide several advantages. They are configured using a standard state estimation framework, i.e., by choosing a physical motion model and tuning process and measurement noise. On the other hand, B-splines require tuning the polynomial degree and the spacing between the knots, which can be a nontrivial task [32]. Furthermore, unlike B-splines, GPs estimate trajectory covariance.

The advantages of the proposed calibration method are as follows:

1) *joint spatiotemporal calibration* based on efficient on-manifold optimization;
2) *theoretically grounded batch state estimation and interpolation*, based on the theory of GPs, which enables both the *time delay and clock drift estimation*;
3) graph-based extension enabling *multisensor calibration*;
4) *computational efficiency*, thanks to the exactly sparse GP priors resulting with $\mathcal{O}(N)$ complexity with respect to the number of measurements.

Furthermore, the GP interpolation provides an *exact temporal registration* between the sensors, which is necessary for the extrinsic calibration. We evaluate the proposed method in extensive simulation and real-world experiments with five different multisensor setups and compare the method to state-of-the-art. Note that the proposed method *requires only* that sensors can track position of the same moving target. Thus, we can use variety of different targets, while specific knowledge about the target can be used in the preprocessing step, e.g., target size for monocular camera scale recovery. Furthermore, this article is complemented by an open-source ROS toolbox *Calirad* implementing the proposed method and a C++ library *ESGPR* implementing the GP regression.

The rest of this article is organized as follows. Section II formulates the problem, provides theoretical insights on the used exactly sparse GP regression, and elaborates the proposed multisensor spatiotemporal calibration method. Section III shows the results of the method on the simulated data where ground truth calibration is available and compares it to a state-of-the-art ego-motion based method. Experimental results with four different multisensor setups, combined with discussion on implementation details, are given in Section IV. Finally, Section V concludes this article.

II. Proposed Calibration Method

In this section, we formulate the spatiotemporal calibration problem, present necessary theoretical insights, and describe individual steps of the proposed method. The novel calibration method can be separated in the following two consecutive steps: 1) representing the trajectories of moving targets captured by each sensor with a separate GP and 2) joint spatiotemporal calibration based on GP interpolation and efficient on-manifold optimization. Furthermore, the method can be seamlessly extended to graph representation enabling multisensor calibration. Given that, in Section II-A, we first formulate our problem and then present the necessary background on GPs in Section II-B. The following Section II-C describes the proposed on-manifold pairwise calibration, while Section II-D introduces adjustments for seamless multisensor calibration.

A. Problem Formulation

The goal of our method is to enable extrinsic and temporal calibration of heterogeneous exteroceptive sensors, e.g., cameras, lidars, radars, sonars, etc. The method relies on tracking the calibration target whose 3-D position can be determined by all sensors. To formalize the approach, we start with defining a target reference frame $\mathcal{F}_t$, described by the target's position ${}^s\boldsymbol{p}(k)$ and orientation ${}^s\boldsymbol{R}(k)$ at discrete time instants. When target reference frames between sensors do not align (e.g., different sensor modalities measure different points on the target), target orientation from one of the sensors and known target configuration are used to express the positions in a unified target reference frame. After this step, we continue to use only target positions because some sensors cannot estimate the target

orientation, e.g., the radar. In addition, it also allows us to use a linear motion model yielding faster GP regression; thus, for each sensor, the GP regression takes in ${}^s\boldsymbol{p}(k)$ and outputs continuous-time target trajectories ${}^s\boldsymbol{x}(t)$. The method itself is not limited to any target design as it abstracts the sensor readings with the estimated trajectories using GPs.

One of the advantages of our method is that it does not require motion of the sensor system. By relying on target motion, we can perform highly dynamic motions and obtain informative data for precise temporal calibration regardless of the system. While hand-held device can rely on motion-based methods for temporal calibration, sensor systems such as vehicles can greatly benefit from this approach. However, we point out that our method is not limited to static sensor systems, i.e., we can either move the sensor platform or the target itself. Lastly, to achieve accurate temporal calibration it is crucial to avoid any source of clock jitter. Clock jitter can be avoided by using local sensors' clocks even though clock drift might be present. If the clock drift is ignored, time delay becomes nonstationary and the system performance degrades over time. Thus, our temporal calibration approach is extended to estimate clock drift together with the time delay. By relying solely on the sensor measurements, our method is not affected by the clock jitter.

B. GP Trajectory Representation

The proposed method is based on the GP regression approach to target trajectory estimation, leveraging the work in [27]–[29]. It enables an efficient continuous-time trajectory estimation based on discrete-time position measurements, i.e., we are able to query the state at any time of interest. Thus, continuous-time GP representation enables elegant temporal correspondence registration between asynchronous sensors with different frame rates. In this section, we give a brief overview of the GP regression necessary for our method, while we refer the reader to [27] for more details.

We consider systems with a continuous-time GP model prior

$$\boldsymbol{x}(t) \sim \mathcal{GP}(\check{\boldsymbol{x}}(t), \check{\boldsymbol{P}}(t,t')) \tag{1}$$

and a discrete time, linear measurement model

$$\boldsymbol{y}_k(t) = \boldsymbol{C}_k \boldsymbol{x}_k(t_k) + \boldsymbol{n}_k \tag{2}$$

where $\boldsymbol{x}(t)$ is the state, $\check{\boldsymbol{x}}(t)$ is the mean function, $\check{\boldsymbol{P}}(t,t')$ is the covariance function, $\boldsymbol{y}_k$ are the measurements, $\boldsymbol{n}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ is Gaussian measurement noise, and $\boldsymbol{C}_k$ is the measurement model matrix. For now, we assume that the state is queried at the measurement times, and we will describe querying at other times in (13) and (14). Following the approach presented in [27], the Gaussian posterior evaluates to

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\left(\underbrace{(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C})^{-1}(\check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{y})}_{\hat{\boldsymbol{x}},\ \text{posterior mean}}, \times \underbrace{(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C})^{-1}}_{\hat{\boldsymbol{P}},\ \text{posterior covariance}} \right). \tag{3}$$

After rearranging the posterior mean expression, a linear system for stacked vector of posterior states $\hat{\boldsymbol{x}}$ is obtained

$$(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C})\hat{\boldsymbol{x}} = (\check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}} + \boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{y}) \tag{4}$$

where $\check{\boldsymbol{P}}$, $\boldsymbol{C}$, and $\boldsymbol{R}$ are batch matrices defined as $\check{\boldsymbol{P}} = [\check{\boldsymbol{P}}(t_i,t_j)]_{ij}$, $\boldsymbol{C} = \operatorname{diag}(\boldsymbol{C}_0,\ldots,\boldsymbol{C}_N)$, and $\boldsymbol{R} = \operatorname{diag}(\boldsymbol{R}_0,\ldots,\boldsymbol{R}_N)$, while $\check{\boldsymbol{x}}$ and $\boldsymbol{y}$ are stacked vectors of prior states at measurement times and actual sensor measurements, $\check{\boldsymbol{x}} = [\check{\boldsymbol{x}}_0,\ldots,\check{\boldsymbol{x}}_N]^T$ and $\boldsymbol{y} = [\boldsymbol{y}_0,\ldots,\boldsymbol{y}_N]^T$, with N being the number of measurements. In general, time complexity for solving (4), as currently presented, is $\mathcal{O}(N^3)$ [29]. To improve the computational efficiency, a special class of GP priors is introduced, whose sparsely structured matrices can be exploited.

The special class of GP priors is based on the following linear time-varying stochastic differential equation (LTV-SDE)

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{v}(t) + \boldsymbol{L}(t)\boldsymbol{w}(t) \tag{5}$$

where $\boldsymbol{F}$ and $\boldsymbol{L}$ are system matrices, $\boldsymbol{v}$ is a known control input, and $\boldsymbol{w}(t)$ is generated by a white noise process. The white noise process is itself a GP with zero mean value

$$\boldsymbol{w}(t) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{Q}_c\delta(t-t')) \tag{6}$$

where $\boldsymbol{Q}_c$ is a power spectral density matrix.

The mean and the covariance of the GP are generated from the solution of the LTV-SDE given in (5)

$$\check{\boldsymbol{x}}(t) = \boldsymbol{\Phi}(t,t_0)\check{\boldsymbol{x}}_0 + \int_{t_0}^{t} \boldsymbol{\Phi}(t,s)\boldsymbol{v}(s)\,\mathrm{d}s \tag{7}$$

$$\check{\boldsymbol{P}}(t,t') = \boldsymbol{\Phi}(t,t_0)\check{\boldsymbol{P}}_0\boldsymbol{\Phi}(t',t_0)^T + \int_{t_0}^{\min(t,t')} \boldsymbol{\Phi}(t,s)\boldsymbol{L}(s)\boldsymbol{Q}_c\boldsymbol{L}(s)^T\boldsymbol{\Phi}(t',s)^T\,\mathrm{d}s \tag{8}$$

where $\check{\boldsymbol{x}}_0$ and $\check{\boldsymbol{P}}_0$ are the initial mean and covariance of the first state, and $\boldsymbol{\Phi}(t,s)$ is the state transition matrix [28].

Due to the Markov property of the LTV-SDE in (5), the inverse kernel matrix $\check{\boldsymbol{P}}^{-1}$ of the prior, which is required for solving the linear system in (4), is exactly sparse block tridiagonal [28]

$$\check{\boldsymbol{P}}^{-1} = \boldsymbol{F}^{-T}\boldsymbol{Q}^{-1}\boldsymbol{F}^{-1} \tag{9}$$

where

$$\boldsymbol{F}^{-1} = \begin{bmatrix} \boldsymbol{1} & 0 & \ldots & 0 & 0 \\ -\boldsymbol{\Phi}(t_1,t_0) & \boldsymbol{1} & \ldots & 0 & 0 \\ 0 & -\boldsymbol{\Phi}(t_2,t_1) & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \boldsymbol{1} & 0 \\ 0 & 0 & \ldots & -\boldsymbol{\Phi}(t_N,t_{N-1}) & \boldsymbol{1} \end{bmatrix} \tag{10}$$

and

$$\boldsymbol{Q}^{-1} = \operatorname{diag}(\check{\boldsymbol{P}}_0^{-1}, \boldsymbol{Q}_{0,1}^{-1},\ldots,\boldsymbol{Q}_{N-1,N}^{-1}) \tag{11}$$

with

$$\boldsymbol{Q}_{a,b} = \int_{t_a}^{t_b} \boldsymbol{\Phi}(t_b,s)\boldsymbol{L}(s)\boldsymbol{Q}_c\boldsymbol{L}(s)^T\boldsymbol{\Phi}(t_b,s)^T\,\mathrm{d}s. \tag{12}$$

This kernel allows for computationally efficient, structure-exploiting inference with $\mathcal{O}(N)$ complexity. This is the main advantage of the proposed exactly sparse GP priors based on an LTV-SDE in (5).

As we previously stated, the key benefit of using GPs for the continuous-time target trajectory estimation is the possibility to query the state $\hat{\boldsymbol{x}}(\tau)$ at any time of interest τ, and not only at measurement times. For multisensor calibration, this proves to be extremely useful, since many sensors operate at different frequencies; thus, the GP approach enables us to temporally align the measurements. If the prior proposed in (7) is used, GP interpolation can be performed efficiently due to the aforementioned Markovian property of the LTV-SDE in (5). State $\hat{\boldsymbol{x}}(\tau)$ at $\tau \in [t_i, t_{i+1}]$ is a function of only its neighboring states [29]

$$\hat{\boldsymbol{x}}(\tau) = \check{\boldsymbol{x}}(\tau) + \boldsymbol{\Lambda}(\tau)(\hat{\boldsymbol{x}}_i - \check{\boldsymbol{x}}_i) + \boldsymbol{\Psi}(\tau)(\hat{\boldsymbol{x}}_{i+1} - \check{\boldsymbol{x}}_{i+1}) \quad (13)$$

$$\boldsymbol{\Lambda}(\tau) = \boldsymbol{\Phi}(\tau, t_i) - \boldsymbol{\Psi}(\tau)\boldsymbol{\Phi}(t_{i+1}, t_i) \quad (14)$$

$$\boldsymbol{\Psi}(\tau) = \boldsymbol{Q}_{i,\tau}\boldsymbol{\Phi}(t_{i+1}, \tau)^T \boldsymbol{Q}_{i,i+1}^{-1} \quad (15)$$

where $\boldsymbol{Q}_{a,b}$ is given in (12). The fact that any state $\check{\boldsymbol{x}}(\tau)$ can be computed in $O(1)$ complexity can be exploited for efficient matching of trajectories of a target detected by multiple sensors.

For the calibration purposes, measurements from individual sensors are used to create separate GPs, where $s \in S$ represents a particular sensor. As we will see in Section II-C, temporal calibration requires velocity estimates in the analytical Jacobians. While the simplest applicable motion model is the constant velocity (CV) model, we opt for the constant acceleration (CA) model. From our experience, the CV model cannot capture the necessary maneuvering dynamics of the target and provides slightly lower precision. However, it can be applied if further decrease in computation time is needed. The model for the sensor s trajectory ${}^s\boldsymbol{x}(t) \in \mathbb{R}^{9\times 1}$ consists of position ${}^s\boldsymbol{p}(t) \in \mathbb{R}^{3\times 1}$, velocity ${}^s\boldsymbol{v}(t) \in \mathbb{R}^{3\times 1}$, and acceleration ${}^s\boldsymbol{a}(t) \in \mathbb{R}^{3\times 1}$

$${}^s\boldsymbol{x}(t) = \begin{bmatrix} {}^s\boldsymbol{p}(t) \\ {}^s\boldsymbol{v}(t) \\ {}^s\boldsymbol{a}(t) \end{bmatrix} \sim \mathcal{GP}({}^s\check{\boldsymbol{x}}(t), {}^s\check{\boldsymbol{P}}(t, t')). \quad (16)$$

To employ the CA motion prior, the LTV-SDE matrices in (5) have the following form:

$$\boldsymbol{F}(t) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \boldsymbol{L}(t) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{1} \end{bmatrix}, \boldsymbol{C}(t) = \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}^T \quad (17)$$

while the matrices $\boldsymbol{\Phi}(t, s)$ and $\boldsymbol{Q}_{a,b}$ are defined as

$$\boldsymbol{\Phi}(t, s) = \begin{bmatrix} \boldsymbol{1} & (t-s)\boldsymbol{1} & \frac{(t-s)^2}{2}\boldsymbol{1} \\ \boldsymbol{0} & \boldsymbol{1} & (t-s)\boldsymbol{1} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \quad (18)$$

$$\boldsymbol{Q}_{a,b} = \begin{bmatrix} \frac{\Delta t^5}{20}\boldsymbol{Q}_c & \frac{\Delta t^4}{8}\boldsymbol{Q}_c & \frac{\Delta t^3}{6}\boldsymbol{Q}_c \\ \frac{\Delta t^4}{8}\boldsymbol{Q}_c & \frac{\Delta t^3}{3}\boldsymbol{Q}_c & \frac{\Delta t^2}{2}\boldsymbol{Q}_c \\ \frac{\Delta t^3}{6}\boldsymbol{Q}_c & \frac{\Delta t^2}{2}\boldsymbol{Q}_c & \Delta t\boldsymbol{Q}_c \end{bmatrix} \quad (19)$$

with $\Delta t = t_b - t_a$. We would also like to emphasize that using motion prior with proper covariances can help mitigate the effects of occasional outliers, which can occur in the context of the sensor calibration.

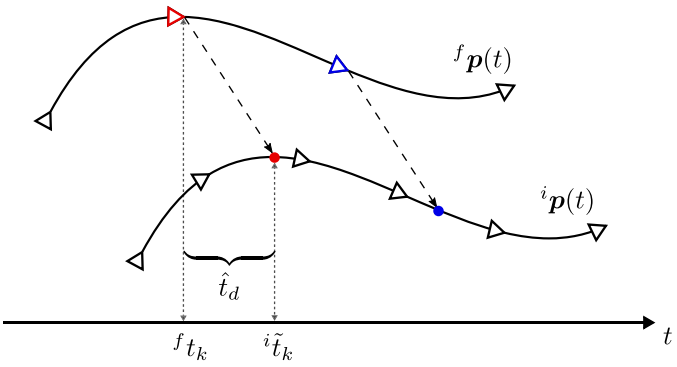


Fig. 1. Continuous-time trajectory representation using GPs provides an elegant temporal registration of asynchronous measurements. Illustration shows the time-delay estimation by aligning two target trajectories, ${}^f\boldsymbol{p}(t)$ and ${}^i\boldsymbol{p}(t)$. States of the fixed sensor at measurement times (triangles) and states at interpolated times (circle) are used to generate correspondences (blue and red pairs).

C. Joint On-Manifold Optimization

Lets consider a sensor setup consisting of two sensors. Once a GP target trajectory for each of them is estimated, we proceed to joint spatiotemporal calibration. Our goal is to find temporal and extrinsic parameters between the sensors, which best align the target trajectories in terms of their positions. This task can be treated as an ICP problem with known point correspondence, but unknown temporal correspondence. We propose an iterative least-square solver that leverages previous work on efficient on-manifold optimization for ICP presented by Grisetti *et al.* [33]. By relying on continuous-time trajectory estimates using the GPs, we are able to extend the solver to estimate temporal calibration between the sensors as well.

We start by defining one sensor as fixed (label *f*) whose states are evaluated at its respective measurement time instances ${}^f t_k$, $k \in (1, N)$. The other sensor we define as the interpolated one (label *i*), because we interpolate its states at each optimization step using (13)–(15) at corresponding time instances based on the current temporal parameters. Fig. 1 illustrates temporal correspondence registration and target position trajectories observed by a fixed and an interpolated sensor, labeled ${}^f\boldsymbol{p}(t)$ and ${}^i\boldsymbol{p}(t)$, respectively. It is worth noting that in the case of different sensor frame rates, the slower sensor should be chosen as the fixed one to reduce interpolation errors [34] and computational costs.

To derive our method, we start by defining the optimization problem as a search for extrinsic and temporal calibration parameters defined on the manifold that is a direct product of the SE(3) and $\mathbb{R}^2$ Lie groups, representing extrinsic and temporal calibration parameters, respectively, i.e., $\boldsymbol{X} \in \mathrm{SE}(3) \times \mathbb{R}^2 = \mathcal{M}$. Given that, we write our state $\boldsymbol{X}$ as the following composite matrix (all other elements are zero):

$$\boldsymbol{X} = \left\{ \begin{bmatrix} {}^f_i\boldsymbol{R} & {}^f_i\boldsymbol{t} \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & t_d \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & k_d \\ 0 & 1 \end{bmatrix} \right\} \in \mathbb{R}^{8\times 8} \quad (20)$$

where ${}_i^f\boldsymbol{R}$, ${}_i^f\boldsymbol{t}$, t_d, and k_d are the rotation matrix, the translation vector, the time delay, and the clock drift coefficient, respectively. At each step of the iterative optimization, using the current estimate of t_d and k_d, we obtain corresponding timestamps using

$$ {}^f t_k = (1 + k_d)^i t_k + t_d. \tag{21}$$

We note that when clock drift estimation is unnecessary, e.g., sensors use a central clock, we simply drop out the terms related to k_d. To follow the standard maximum likelihood estimation (MLE) framework [33], we treat the fixed sensor position estimates obtained by the GP as measurements corrupted by the Gaussian noise

$$ \boldsymbol{z}_k = {}^f\boldsymbol{p}\left({}^f t_k\right) + \nu_k, \quad \nu_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}_k^{-1}) \tag{22}$$

where $\boldsymbol{\Omega}_k \in \mathbb{R}^{3\times 3}$ defines the inverse of the position covariance matrix.

On the other hand, positions of the interpolated sensor ${}^i\boldsymbol{p}(t)$ are nonstationary since they are interpolated at each iteration of the optimization process. Thus, we treat them as a part of the observation model $\boldsymbol{h}_k(\boldsymbol{X}) : \mathcal{M} \to \mathbb{R}^3$

$$ \boldsymbol{h}_k(\boldsymbol{X}) = {}_i^f\boldsymbol{R} \cdot {}^i\boldsymbol{p}\left(\frac{{}^f t_k - t_d}{1 + k_d}\right) + {}_i^f\boldsymbol{t}. \tag{23}$$

To find the optimal solution $\boldsymbol{X}^*$ given all the measurements, the MLE approach suggests to minimize the following expression:

$$ \boldsymbol{X}^* = \arg\min_{\boldsymbol{X}} F(\boldsymbol{X}) \tag{24}$$

$$ F(\boldsymbol{X}) = \sum_{k=1}^{N} \boldsymbol{e}_k^T(\boldsymbol{X}) \boldsymbol{\Omega}_k \boldsymbol{e}_k(\boldsymbol{X}) \tag{25}$$

$$ \boldsymbol{e}_k(\boldsymbol{X}) = \boldsymbol{h}_k(\boldsymbol{X}) - \boldsymbol{z}_k. \tag{26}$$

To solve this optimization problem, we follow the on-manifold Gauss–Newton (GN) optimization framework [35].

Our solver builds upon the formulation of the ICP problem [33] with additional estimation of temporal calibration parameters.

Computationally the most demanding part is the state interpolation that occurs at each iteration.

Therefore, our goal is to minimize the number of cost function evaluations by obtaining the parameter perturbations on the manifold and by using analytical Jacobians that will be derived in the sequel.

As previously stated, the manifold $\mathcal{M}$ on which we perform the optimization is a direct product of the SE(3) and $\mathbb{R}^2$ Lie groups, thus, the perturbation vector $\Delta\boldsymbol{x} \in \mathbb{R}^8$ is the corresponding Lie algebra element

$$ \Delta\boldsymbol{x} = \mathrm{Log}(\boldsymbol{X}) = [\Delta\boldsymbol{r}\ \Delta\boldsymbol{t}\ \Delta t_d\ \Delta k_d]. \tag{27}$$

In order to avoid cluttering, the section with additional mathematical notation, we do not introduce here explicitly Lie group operators. We would just like to point out that our perturbation vector is actually the Euclidean vector of the space isomorphic to the Lie algebra of $\mathrm{SE}(3) \times \mathbb{R}^2$, while the corresponding matrix exponential and logarithm that map the vector space elements to the group, and vice-versa, are denoted as Exp and Log. We believe that this lack of mathematical accuracy does not impact the correctness, but brings clarity in presenting this article method. For more details on Lie groups, Lie algebra, and pertaining operators, we refer the reader to [36], [37].

To perform the on-manifold GN optimization note that our perturbed observation model is as follows (we use perturbation on the left in this article):

$$ \boldsymbol{h}_k(\mathrm{Exp}(\Delta\boldsymbol{x})\boldsymbol{X}) = \Delta\boldsymbol{R}\, {}_i^f\boldsymbol{R}^i\boldsymbol{p}\left(\tilde{t}_k\right) + \Delta\boldsymbol{R}\, {}_i^f\boldsymbol{t} + \Delta\boldsymbol{t} \tag{28}$$

$$ \tilde{t}_k = \frac{{}^f t_k - (t_d + \Delta t_d)}{1 + k_d + \Delta k_d}. \tag{29}$$

We start with an initial guess of the state $\boldsymbol{X}_0 \in \mathcal{M}$ and use it as the current estimate $\hat{\boldsymbol{X}} \in \mathcal{M}$ to evaluate the errors (26). Next, we find the optimal state perturbation $\Delta\boldsymbol{x}$ by linearizing the error term (26) at $\mathrm{Exp}(\Delta\boldsymbol{x})\hat{\boldsymbol{X}}$ using the first-order Taylor approximation

$$ \boldsymbol{e}_k(\mathrm{Exp}(\Delta\boldsymbol{x})\hat{\boldsymbol{X}}) \approx \mathbf{e}_k(\hat{\boldsymbol{X}}) + \underbrace{\left.\frac{\partial \boldsymbol{e}_k(\mathrm{Exp}(\Delta\boldsymbol{x})\hat{\boldsymbol{X}})}{\partial \Delta\boldsymbol{x}}\right|_{\Delta\boldsymbol{x}=0}}_{\boldsymbol{J_k}} \Delta\boldsymbol{x}. \tag{30}$$

After substituting the linearized error (30) into (25) to obtain a linearized criterion, we get the following quadratic form:

$$ F(\mathrm{Exp}(\Delta\boldsymbol{x})\hat{\boldsymbol{X}}) \approx \Delta\boldsymbol{x}^T\boldsymbol{H}\Delta\boldsymbol{x} + 2\boldsymbol{b}^T\Delta\boldsymbol{x} + \sum_{k=1}^{N} \boldsymbol{e}_k^T(\hat{\boldsymbol{X}})\boldsymbol{\Omega}_k\boldsymbol{e}_k(\hat{\boldsymbol{X}}) \tag{31}$$

where

$$ \boldsymbol{H} = \sum_{k=1}^{N} \boldsymbol{J}_k^\top \boldsymbol{\Omega}_k \boldsymbol{J}_k, \quad \boldsymbol{b} = \sum_{k=1}^{N} \boldsymbol{J}_k^\top \boldsymbol{\Omega}_k \boldsymbol{e}_k. \tag{32}$$

The optimal perturbation vector at each iteration is found by equating the derivative of (31) with zero

$$ \Delta\boldsymbol{x} = -\boldsymbol{H}^{-1}\boldsymbol{b}. \tag{33}$$

We then update the current state estimate using $\hat{\boldsymbol{X}} \leftarrow \mathrm{Exp}(\Delta\boldsymbol{x})\hat{\boldsymbol{X}}$ and the process is repeated until convergence.

As the final ingredient, we derive the analytical (left) Jacobians as they are essential for reducing the computational complexity. We start by separating the complete kth Jacobian to subparts for convenience

$$ \boldsymbol{J}_k = [\boldsymbol{J}_k^{\Delta\boldsymbol{r}}\ \boldsymbol{J}_k^{\Delta\boldsymbol{t}}\ \boldsymbol{J}_k^{\Delta t_d}\ \boldsymbol{J}_k^{\Delta k_d}]. \tag{34}$$

We approximate the perturbation rotation matrix by $\Delta R = \boldsymbol{I} + [\Delta r]_\times$ [36], where the $[\cdot]_\times$ operator constructs a skew-symmetric matrix from the vector. Leveraging this approximation and neglecting the constant terms, which disappear via derivation, we obtain the following Jacobians:

$$ \boldsymbol{J}_k^{\Delta\boldsymbol{t}} = \left.\frac{\partial(\Delta\boldsymbol{t})}{\partial(\Delta\boldsymbol{t})}\right|_{\Delta\boldsymbol{x}=0} = \boldsymbol{I} \tag{35}$$

$$ \boldsymbol{J}_k^{\Delta\boldsymbol{r}} = \left.\frac{\partial(\Delta\boldsymbol{R} \cdot {}^i\boldsymbol{p}_k')}{\partial(\Delta\boldsymbol{r})}\right|_{\Delta\boldsymbol{x}=0} = \left[-{}^i\boldsymbol{p}_k'\right]_\times \tag{36}$$

$${}^i\boldsymbol{p}_k' = {}_i^f\boldsymbol{R} \cdot {}^i\boldsymbol{p}\left(\tilde{t}_k\right) + {}_i^f\boldsymbol{t}. \tag{37}$$

And regarding the temporal calibration parameters, Jacobians evaluate to the following expressions:

$$\boldsymbol{J}_k^{\Delta t_d} = \left.\frac{\partial(\Delta \boldsymbol{R}_i^f \boldsymbol{R}^i \boldsymbol{p}\left(\tilde{t}_k\right))}{\partial(\Delta t_d)}\right|_{\Delta \boldsymbol{x}=0} = {}_i^f\boldsymbol{R}^i \boldsymbol{v}\left(\tilde{t}_k\right) \frac{-1}{1+k_d} \quad (38)$$

$$\boldsymbol{J}_k^{\Delta k_d} = \left.\frac{\partial(\Delta \boldsymbol{R}_i^f \boldsymbol{R}^i \boldsymbol{p}\left(\tilde{t}_k\right))}{\partial(\Delta k_d)}\right|_{\Delta \boldsymbol{x}=0} = {}_i^f\boldsymbol{R}^i \boldsymbol{v}\left(\tilde{t}_k\right) \frac{t_d - {}^f t_k}{(1+k_d)^2} \quad (39)$$

where ${}^i\boldsymbol{v}(\tilde{t}_k)$ is the interpolated sensor's velocity estimate of the target at time instant $\tilde{t}_k$. As shown in Section II-B, it is readily available since the used GPs provide smooth continuous-time velocity estimates.

Finally, it is crucial to keep the number of correspondences constant to ensure convergence; otherwise, the cost function loses its smoothness, because adding or removing a correspondence inevitably introduces a discontinuity and prevents convergence. This situation occurs when the method seeks correspondence between the fixed sensor and the interpolated state of the second sensor, which is outside of the trajectory lifetime. Even though the GP framework allows for extrapolation into the future or the past, thus enabling the necessary correspondences, we avoid this approach as it does not convey any additional information and could possibly degrade the calibration results. Instead, we set a lower and upper bound on the time delay. Given that, we align two GP trajectories and discard fixed measurements at the beginning and the end in accordance to the bounds, thus ensuring constant number of correspondences.

D. Multisensor Extension

The proposed method can be easily modified and applied in multisensor scenarios (with more than two sensors) by relying on extrinsic graph-based calibration [38] and extending it to perform temporal calibration as well. For the multisensor case, in addition to previously defined fixed and interpolated sensors, we also need to declare one sensor as the global reference sensor, labeled *r*, since fixed and interpolated sensors now relate a pairwise relation within the graph. Then, we search for extrinsic and temporal parameters relating sensors $(2, \ldots, S)$ to the reference sensor. When there are only two sensors, the reference and fixed sensor are the same, while here we choose one fixed sensor for each edge of the graph, preferably the one with the lower frame rate. To start, we need to modify the state vector from (20) to

$$\boldsymbol{X} = \{X_1 \times X_2 \times \cdots \times X_S\} \in \mathbb{R}^{8\cdot S \times 8 \cdot S}. \quad (40)$$

Each node in the graph represents sensor's extrinsic and temporal parameters, while edges represent correspondences between sensors. Due to multiple edges in a general graph, we need to redefine the observation model for the multisensor approach as follows:

$$\boldsymbol{z}_k^{f,i} = \nu_k^{f,i} \quad (41)$$

$$\nu_k^{f,i} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}_{f,i,k}^{-1}) \quad (42)$$

$$\boldsymbol{h}_k^{f,i}(\boldsymbol{X}) = \boldsymbol{h}_k^i(\boldsymbol{X}) - \boldsymbol{h}_k^f(\boldsymbol{X}) \quad (43)$$

$$\boldsymbol{h}_k^f(\boldsymbol{X}) = {}_f^r\boldsymbol{R}^f \boldsymbol{p}({}^f t_k^i) + {}_f^r\boldsymbol{t} \quad (44)$$

$$\boldsymbol{h}_k^i(\boldsymbol{X}) = {}_i^r\boldsymbol{R}^i \boldsymbol{p}({}^i\tilde{t}_k^f) + {}_i^r\boldsymbol{t} \quad (45)$$

$${}^i\tilde{t}_k^f = \frac{(1+k_{d,f})^f t_k^i + t_{d,f} - t_{d,i}}{1+k_{d,i}}. \quad (46)$$

In the multisensor approach, the target positions from all sensors depend on the estimated temporal parameters (except for the reference sensor); thus, target positions from both sensors within a graph edge are part of the observation model $\boldsymbol{h}_k^{f,i}(\boldsymbol{X})$. To avoid interpolation of both sensor trajectories, we have decided to keep time instances ${}^f t_k^i$ fixed, where they represent measurement times of the fixed sensor that have correspondence with the interpolated sensor. As such, $\boldsymbol{h}_k^f(\boldsymbol{X})$ depends only on the extrinsic parameters of the fixed sensor. On the other hand, we combine temporal parameters of both the fixed and the interpolated sensor into $\boldsymbol{h}_k^i(\boldsymbol{X})$ by first transforming ${}^f t_k^i$ into the reference clock and then into the interpolated sensors clock via (47).

Following these extensions, we need to modify the objective function (25) to sum over all edges defined with set $\mathcal{E}$

$$F(\boldsymbol{X}) = \sum_{(f,i)\in\mathcal{E}} \sum_{k=1}^{N_{f,i}} (\boldsymbol{e}_k^{f,i}(\boldsymbol{X}))^T \boldsymbol{\Omega}_k \boldsymbol{e}_k^{f,i}(\boldsymbol{X}) \quad (47)$$

$$\boldsymbol{e}_k^{f,i}(\boldsymbol{X}) = \boldsymbol{h}_k^{f,i}(\boldsymbol{X}) - \boldsymbol{z}_k^{f,i}. \quad (48)$$

While the remaining expressions are trivially adjusted and omitted here for brevity, we state the Jacobians with respect to the temporal parameters, since they are slightly more complex due to the novel formulation (47)

$$\boldsymbol{J}_k^{\Delta t_{d,f}} = {}_i^r\boldsymbol{R}^i \boldsymbol{v}\left({}^i\tilde{t}_k^f\right) \frac{1}{1+k_{d,i}} \quad (49)$$

$$\boldsymbol{J}_k^{\Delta t_{d,i}} = {}_i^r\boldsymbol{R}^i \boldsymbol{v}\left({}^i\tilde{t}_k^f\right) \frac{-1}{1+k_{d,i}} \quad (50)$$

$$\boldsymbol{J}_k^{\Delta k_{d,f}} = {}_i^r\boldsymbol{R}^i \boldsymbol{v}\left({}^i\tilde{t}_k^f\right) {}^f t_k \frac{{}^f t_k^i}{1+k_{d,i}} \quad (51)$$

$$\boldsymbol{J}_k^{\Delta k_{d,i}} = {}_i^r\boldsymbol{R}^i \boldsymbol{v}\left({}^i\tilde{t}_k^f\right) {}^f t_k \frac{(1+k_{d,f})^f t_k^i + t_{d,f} - t_{d,i}}{-(1+k_{d,i})^2}. \quad (52)$$

III. Simulation Results

Sensor calibration is a task for which ground truth is virtually impossible to obtain in real world experiments. Given that, we use synthetic datasets with known ground truth to assess accuracy of our method and compare it to a state-of-the-art motion-based method [39].

A. Method Analysis

To analyze the results of our method in a controlled environment, we simulated an experiment which would mimic a real world experiment. We simulated 1000 sinusoidal trajectories that lasted for 60 s (20 s in each direction) with an amplitude of 1 m and sine period of 4 s. All the sensors operate at 20 Hz and we add white noise with standard deviation of $\sigma_p = 0.01$ m to position measurements.

To test the graph based multisensor calibration, we simulated a graph of four sensors with edges $\mathcal{E} =$

TABLE I
MEAN ABSOLUTE ERROR OVER GRAPH

f – i	$\Delta_i^f \boldsymbol{R}$ [°]	$\Delta^f \boldsymbol{t_i}$ [mm]	$\lvert\Delta t_{d,f,i}\rvert$ [ms]
1 – 2	0.065	1.81	0.30
1 – 3	0.066	1.76	0.30
1 – 4	0.065	1.73	0.29
2 – 3	0.066	1.75	0.30
3 – 4	0.065	1.73	0.29

$((1,2),(1,3),(2,3),(3,4))$. The first sensor is chosen as the reference, while we set various relative transformations and delays between different sensors up to 400 ms, 40 cm, and 70° in Euler angles. Table I shows mean absolute errors between the estimated parameters and the ground truth for five sensor pairs of interest. Rotational error $\Delta_i^f \boldsymbol{R}$ is defined as angle in the angle-axis representation of the rotation matrix ${}_i^f\boldsymbol{R}^T\,{}_i^f\boldsymbol{R}_{gt}$, while translational error $\Delta^f \boldsymbol{t}_i$ is the standard Euclidean norm of the difference ${}^f\boldsymbol{t}_i - {}^f\boldsymbol{t}_{i,gt}$. From Table I, we can see that error is essentially equal for all the sensor combinations, whether they share a connection or not. Furthermore, we also obtained very similar results by using a pairwise approach which, however, does not preserve the global consistency. To assess the consistency error present with the pairwise approach, we have "closed the loop" by combining the following pairwise transformations: 1–2, 2–3, 3–1. The resulting mean/maximum errors after the loop closing were $0.007°/0.02°$, 0.07 mm/0.23 mm, and 0.06 ms/0.27 ms for rotation, translation, and time delay, respectively. Note that for the graph based approach these errors are zero.

To gain further insights about the influence of the experimental setup and modeling, we compared CV and CA motion models for the GP, tested different dynamics of the moving target, added clock drift estimation when it did not exist, and varied the measurement noise. We noted that using the simpler CV motion model resulted with an increase in the mean absolute time-delay error from 0.30 to 0.44 ms, showing that using a more complex CA motion model is justified. When we doubled the sine frequency, it lowered the mean absolute time-delay error from 0.30 to 0.15 ms, indicating that the precision of our method is mostly limited by the experiment design. Furthermore, when we included the clock drift in the optimization, we noticed an increase of the mean absolute time-delay error from 0.30 to 0.62 ms. This effect is most likely due to overfitting and suggest that clock drift should not be estimated if it does not exist, e.g., if sensors use a central clock. Finally, we examined the influence of the measurement error by simulating severe noise $\sigma_p = 0.05$ m. It increased the mean absolute errors to $\Delta_i^f \boldsymbol{R} = 0.37°$, $\Delta^f \boldsymbol{t_i} = 10.2$ mm, and $|\Delta t_{d,f,i}| = 2.1$ ms.

B. Comparison With an Ego-Motion Based Method

In this section, we compared our method to a state-of-the-art ego-motion-based method named SRRG by Della Corte *et al.* [39]. We generated synthetic data using a *Bernoulli-Lemniscate* 3-D trajectory simulator provided with the accompanying SRRG toolbox. The generated trajectory resembled a figure eight and excited all rotational axes leading to full observabilty for the ego-motion based methods. We simulated 1000 1-min-long datasets with two sensors operating at 20 Hz ($T = 50$ ms) and we added white noise with standard deviation of $\sigma_p = 0.01$ m to positions of the sensors and $\sigma_\theta = 0.1°$ to each Euler angle representing the sensor orientations. Ground truth time delay was set to 0 ms, translation to ${}^1\boldsymbol{t}_2 = [0.2\ 0.2\ 0.2]^T$ m and rotation expressed as quaternion to ${}_2^1\boldsymbol{q} = [0.85\ 0.30\ 0.30\ 0.30]$. Besides odometry constraints, the SRRG method allows addition of a generic ICP constraints using raw sensor data to improve results of the extrinsic calibration. We tested both approaches and refer to them as SRRG-ODO and SRRG-ICP. To enable SRRG-ICP, the dataset was expanded with 300 points (added white noise with standard deviation of $\sigma_p = 0.01$ m) that were observed throughout the whole trajectory. The input to our GP method was only positions of sensor reference frame origins. To obtain continuous-time trajectories, the SRRG-ODO approach uses linear interpolation for translation and spherical linear interpolation for rotation. On the other hand, there is no continuous time representation for the SRRG-ICP constraint, but they select two closest measurements between sensors based on current time-delay estimate. Thus, the SRRG-ICP constraint mostly helps correct the extrinsic calibration, which can be unobservable for odometry-based constraints (e.g., planar motion). We briefly note that all three methods produced unbiased estimates of extrinsic parameters. They produced mean absolute translational and rotational errors $\boldsymbol{e}_{\text{GP}} = (0.2\,\text{cm},\ 0.42°)$, $\boldsymbol{e}_{\text{SRRG-ODO}} = (2.4\,\text{cm},\ 0.31°)$ and $\boldsymbol{e}_{\text{SRRG-ICP}} = (0.8\,\text{cm},\ 0.24°)$. Furthermore, we tested the SRRG-ICP method with disabled temporal calibration using the ground truth delay. It reduced the errors to $\boldsymbol{e}_{\text{SRRG-ICP}} = (0.1\,\text{cm},\ 0.02°)$ showing the influence of incorrect temporal calibration described in the sequel.

In this scenario, our method produced an accurate unbiased estimate of the time delay with normal distribution $t_d = \mathcal{N}(0.0004, 0.54)$ ms. On the other hand, SRRG-ODO and SRRG-ICP provided estimates of the time-delay spread across the interval $(-54, 2.7)$ ms, with two modes, one at $-T$ and one at 0 ms. Furthermore, lowering the sensor frequency to 10 Hz caused the stronger separation of the modes with most of the estimates being spread ± 8 ms around the $-T$ and 0 ms modes. After thorough testing of the SRRG method, we concluded that the lack of smoothness in the cost function might cause incorrect convergence. Namely, the method relies on numerical calculation of the Jacobian with respect to the time delay, where parameter ϵ_{time} is used to differentiate the cost function. While the SRRG toolbox suggests setting it to $\epsilon_{\text{time}} = T$, we noticed that with $\epsilon_{\text{time}} < T/2$, the method completely diverges. Thus, it is not trivial to choose a proper ϵ_{time} when sensors have significantly different frequencies as in our multisensor real-world experiment that will be presented in the sequel.

IV. EXPERIMENTAL RESULTS

To validate the proposed calibration method, we conducted thorough real-world experiments on the following five different sensor setups.

1) Hardware synchronized stereo camera—testing the method on a setup with accurate ground truth.

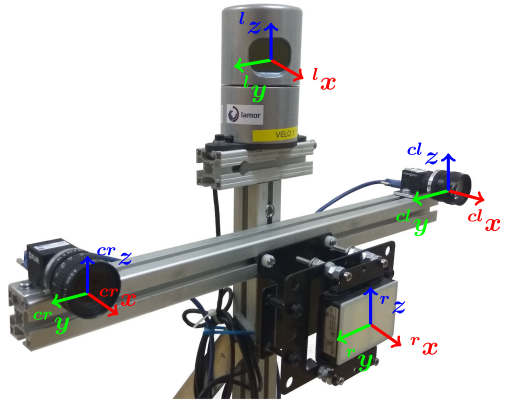


Fig. 2. Illustration of the used multisensor system with corresponding sensor coordinate systems.

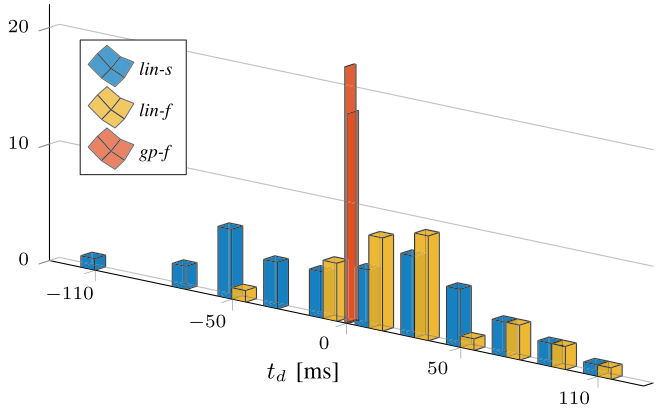


Fig. 3. Histograms compare the estimated time delays using our method (*gp-f*) applied on the whole dataset, to the linear interpolation method [15], where *lin-f* uses the whole dataset, while *lin-s* uses sequences with target moving along the optical axes, thus not introducing a bias in the estimation. Ground truth time delay was $t_d = 0$ ms.

2) Camera and motion capture system—testing heterogeneous sensors operating at significantly different frame rates with separate clocks exhibiting substantial drift.
3) 3-D lidar and camera—calibration accounting for the lidar's sweeping data acquisition process.
4) Radar and camera—calibration tackling automotive radar's lack of 3-D position measurement.
5) Camera, 3-D lidar, and motion capture system—testing the multisensor graph-based calibration.

In all the experiments, we used a single known target for convenience, even though the method does not rely on a special target. We covered a planar triangular cardboard with motion capture markers and an AprilTag [40], a square fiducial marker of side length $a = 16$ cm that removed camera's scale ambiguity, thus enabling 3-D target position estimation. Additionally, to get reliable radar detections, we have adopted the target design from [41] and placed a metal corner reflector behind the cardboard. Fig. 2 shows the used multisensor system that was mounted on the Husky A200 mobile robot platform. Intrinsic camera calibration was obtained using the Kalibr toolbox [42], while we used factory calibration for the remaining sensors. In the end, we analyze the proposed method's computational complexity and influence of the hyperparameters.

A. Hardware Synchronized Stereo Camera

In this experiment, we used two PointGrey BFLY-U3-23S6M-C global shutter cameras with Kowa C-Mount 6 mm f/1.8-16 1" HC fixed lens with $96.8° \times 79.4°$ field of view. The cameras were synchronized by an external trigger with the sampling rate set to 0.05 s. Note that this setup does not require temporal calibration; however, we leverage this fact to have an experiment with a ground truth time delay ($t_d = 0$ s). We have recorded 40 1-min-long sequences and compared the performance of the proposed approach to two recent temporal calibration frameworks based on target tracking [7], [15]. The first method [7], correctly estimated the zero time delay for all the 40 recorded sequence, but the approach is limited to estimating the time delay as a multiple of the sampling rate, thus requiring all the sensors to operate at equal sampling rates. Given that, although accurate, due to this limitation the method is not considered further in this article.

The second method [15], is capable of temporal calibration of asynchronous sensors in the continuous-time domain by relying on linear interpolation of the position norm, thus mitigating the limitation of the previous method. In Fig. 3, we compare calibration results of our method and that of linear interpolation. Note that we applied the linear interpolation method first on the whole 40-min-long sequence (*lin-f*), and second on a subset (*lin-s*) for reasons that will be explained in the sequel. Specifically, we can see that linear interpolation exhibits a bias when applied on the full sequence, and larger variance for the subset of the sequence, in comparison to our method (*gp-f*).

The explanation lies in the fact that this method relies on the position norm for the dimensionality reduction, which can cause error with the displacements of sensors. Concretely, in the first third of the dataset, the target was moved along the optical axis of both cameras, and the linear interpolation method provided estimated time delay with fitted Gaussian distribution $\hat{t}_d \sim \mathcal{N}(6.03, 50.99)$ ms[1]. In the remaining parts of the dataset, motion was not aligned with the optical axis and the position norm measurements differed for the two cameras due to large enough displacement, and when applied on the whole dataset the method resulted with the following fitted Gaussian distribution $\hat{t}_d \sim \mathcal{N}(32.20, 33.16)$ ms. Therefore, we believe that the position norm is not the most appropriate dimensionality reduction technique as it is not frame-invariant. As can be seen from Fig. 3, our method was able to produce an unbiased time-delay estimate with the fitted Gaussian distribution $\hat{t}_d \sim \mathcal{N}(0.11, 0.39)$ ms. Furthermore, all the estimates were within the range $(-0.82, 0.78)$ ms, which corresponds to a $\pm 1.6\%$ range of the sampling interval. We can see that the proposed method supports temporal calibration of asynchronous sensors in the continuous-time domain, and that it significantly outperforms the linear interpolation method.

[1]In this article, we represent the Gaussian distribution with the mean and the standard deviation, i.e., $\mathcal{N}(\mu, \sigma)$.

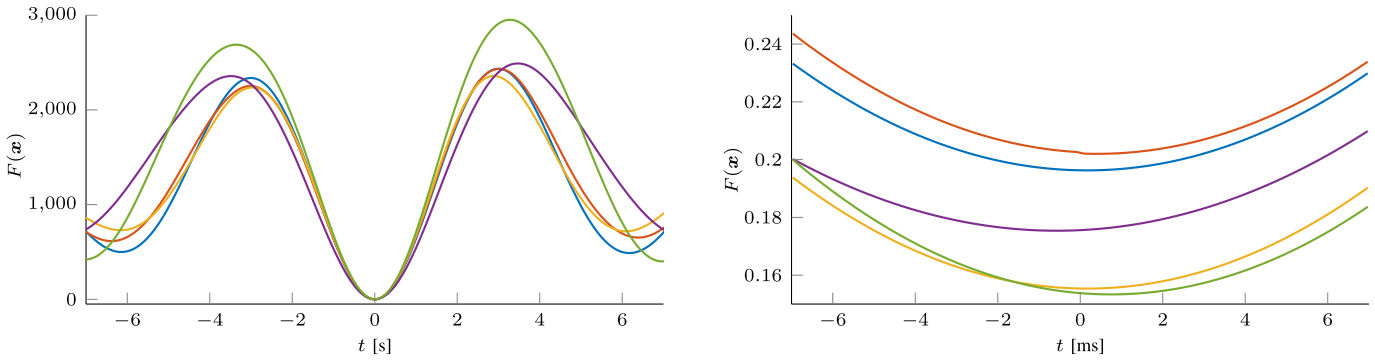


Fig. 4. Cost function of the proposed calibration method for five different stereo camera experiments. Wide preview shows that initialization within ±3 s interval is sufficient for convergence to the global minimum, while the closer preview around the ground truth time delay confirms cost function smoothness. (a) Wide preview illustrating local minima and global minimum. (b) Closer preview around the ground truth.

To further gain insight in the proposed temporal calibration method, we examined the cost function defined by (25) (which is smooth compared to linear interpolation). Fig. 4(a) shows the value of the cost function with respect to the time delay using estimated extrinsics at the interval $t_d \in (-7, 7)$ s, while Fig. 4(b) provides a closer look around the global optimum, $t_d \in (-7, 7)$ ms. For clarity, only five out of 40 experiments are shown, while the remaining ones follow the same pattern. From Fig. 4(a), we can see that the cost function has local minima, while the global minimum always resides near the ground truth. Since our method uses an iterative solver, proper initialization is necessary. By initializing the time delay to a starting point in the interval $(-3, 3)$ s, the method would be able to converge to the global minimum for all the experiments [see Fig. 5(a)]. The local minima are tightly coupled with the executed target motion and can be further spread from the global minimum by avoiding repetitive motion or increasing its period. Fig. 4(b) shows that our cost function is smooth with a minimum around the ground truth value, thus enabling stable and accurate results using an iterative optimization.

Furthermore, we use this experiment to evaluate the SRRG method on real data due to available target orientation estimates and time delay ground truth. Pose of the camera with respect to the target was used for the SRRG-ODO method, while the target position in the camera reference frames was added as a single-point input for the SRRG-ICP extension. Both methods suffered the same convergence issue described in Section III-B with estimated time delay distributions $\hat{t}_d \sim \mathcal{N}(-14.42,\ 24.32)$ ms and $\hat{t}_d \sim \mathcal{N}(-16.20, 23.20)$ ms for SRRG-ODO and SRRG-ICP, respectively. Considering the extrinsic calibration, the SRRG-ODO method was not able to estimate translation parameters due to the lack of rotational target movement. This can be seen by comparing estimated means of the translation parameters, e.g., the ${}^1t_{2,x} = 56.9$ cm, ${}^1t_{2,x} = 53.5$ cm, and ${}^1t_{2,x} = 0.9$ cm for *GP*, SRRG-ICP, and SRRG-ODO, respectively. Furthermore, we noticed that our method had significantly greater extrinsic parameter repeatability, as shown with Table II. We attribute this result to wrong temporal calibration, confirming the conclusion by Zuñiga-Noël *et al.* [43] on the SRRG method.

TABLE II
STANDARD DEVIATION OF THE ESTIMATED EXTRINSIC CALIBRATION PARAMETERS

	GP	*SRRG–ODO*	*SRRG–ICP*
${}^1t_{2,x}$ [m]	2.79×10^{-3}	6.30×10^{-2}	2.12×10^{-2}
${}^1t_{2,y}$ [m]	4.79×10^{-3}	3.89×10^{-2}	2.22×10^{-2}
${}^1t_{2,z}$ [m]	2.11×10^{-3}	3.66×10^{-2}	5.50×10^{-3}
${}^1_2\theta_z$ [°]	1.52×10^{-1}	4.21×10^{-1}	4.23×10^{-1}
${}^1_2\theta_y$ [°]	5.79×10^{-2}	6.55×10^{-1}	5.26×10^{-1}
${}^1_2\theta_x$ [°]	3.88×10^{-2}	1.02	6.93×10^{-1}

To conclude, with this experiment, we confirmed that the proposed method provides an unbiased estimate of the time delay, which is precise up to a fraction of the sampling interval, and we also showed convergence to a global solution from a wide set of initial values.

B. Camera and Motion Capture System

In this experiment, we used a single PointGrey camera and the OptiTrack motion capture system (MOCAP). MOCAP provides 6-D pose measurements at 120 Hz by processing measurements on a dedicated computer and assigns local timestamps using the computer's clock. Poses are transmitted over the wireless network to the central computer. The camera provides images at 20 Hz and has an internal clock according to which local timestamps are assigned. Images are transmitted over USB to the central computer. Given that, this setup gives us the following two options for handling data timestamps: 1) to use the time-of-arrival of measurements at the central computer or 2) to use local timestamps provided by each sensor. The first approach eliminates the timestamp drift caused by separate local clocks, but suffers from the network jitter (since MOCAP data are transferred over the wireless network). The second approach is resilient to the jitter, but separate local clocks introduce a timestamp drift. We analyzed both options in a 34-min-long experiment recording a moving calibration target.

For the time-of-arrival approach, the estimated time-delay results, between the MOCAP and camera measurements,

followed the Gaussian distribution $\hat{t}_d \sim \mathcal{N}(17.14, 1.59)$ ms. Note that the obtained mean value can be interpreted as the average time delay due to network jitter and we noticed that most of the deviations were in the ± 2.9 ms range. However, the time delay can differ significantly during the experiments, because of the changing intensity of the network traffic or other protocol induced stochastic effects. Notably, analysis of the MOCAP time-of-arrival jitter showed that 2.7% of the measurements fell in the range of (8.3418) ms, indicating that on some occasions, delay can be much greater than the MOCAP sampling time. Given that, the jitter caused by the network delay can act as a strong limiting factor for the temporal calibration accuracy.

In the local timestamps approach, we used sensor internal clocks, which eliminates the stochastic effects associated with the communication over a wireless network. However, separate local clocks introduce a drift in the time-delay estimation, which has to be addressed. To estimate this drift, we compared the following three approaches:

1) joint drift and delay estimation using the proposed GP method on the full 34 min sequence (*gp-f*);
2) drift estimation on the full 34 min sequence using convex hull approach [44] (*ch-f*);
3) drift estimation using only one minute subsets (*gp-s*/*ch-s*).[2]

In the *gp-f* approach, we performed a joint drift–delay optimization using the proposed GP method on the full sequence. The estimated drift and delay were, $\hat{k}_d = 49.1$ μs/s and $\hat{t}_d = 23$ ms, respectively. In the *ch-f* approach, the authors observe the temporal evolution of the clock skew, i.e., difference between the arrival times and the local timestamps. They estimate a lower convex hull where the slope of the lower boundary represents the clock drift. With this approach, we can obtain each sensor clock drift with respect to the central computer. However, since we are interested in the relative drift, as was estimated in the *gp-f* approach, we report the difference between the two line slopes. Thus, the *ch-f* approach resulted with an estimated relative drift of 49.3 μs/s (at this point, unlike *gp-f*, there is no time-delay estimate). To compare the accuracy of estimated drifts, we tested their impact on the time-delay estimation. The estimated drifts were used to correct local timestamps, which was followed by the proposed GP time-delay estimation on individual one-minute intervals (30 in total). The *ch-f* approach resulted with time-delay estimates in the range (22.80,23.56) ms with estimated distribution $\hat{t}_d \sim \mathcal{N}(23.22, 0.17)$ ms, while the *gp-f* approach resulted with estimates in the range (22.79,23.29) ms with estimated distribution $\mathcal{N}(23.02, 0.12)$ ms. The results depicted in Fig. 5 show the estimated time delays throughout the whole experiment for both the *ch-f* and *gp-f* approach. We can notice that the drift estimate error by the *ch-f* approach introduced a slope of 0.17 μs/s in the time-delay estimate, whereas the *gp-f* approach correctly estimated the drift and provided a consistent time-delay estimate throughout the whole 34 min experiment (the more horizontal line, the better: resulting slope of the time-delay estimate was -0.01 μs/s).

[2]Waving a calibration target for a 34 min stretch requires good stamina, which is why *gp-f* is not a practical approach and serves as the ground truth.

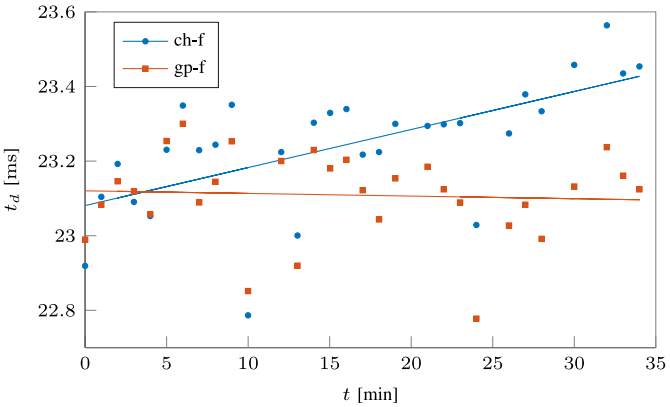


Fig. 5. Estimated time delay for each one-minute interval over the whole experiment for the camera and MOCAP temporal calibration. Time delays were obtained from data with compensated drift using the *ch-f* and *gp-f* drift estimates. Steeper slope of the *ch-f* method indicates larger error in the drift estimate used for timestamp compensation.

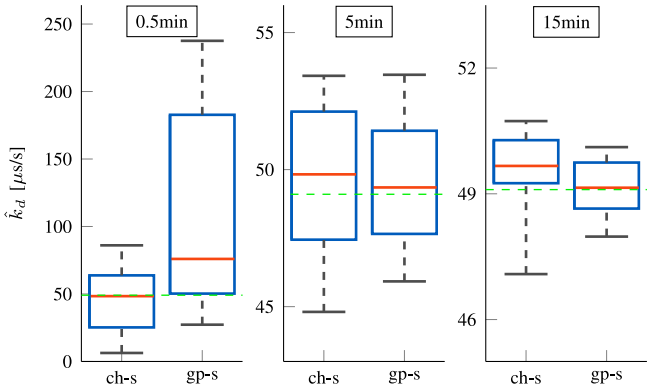


Fig. 6. Uncertainty of the drift estimates for pairs of 30 s intervals separated by 0.5, 5, and 15 min. Ground truth (*gp-f*) is illustrated by the horizontal green dashed line. Interval separation of 0.5 min did not produce a reliable drift estimate for both approaches. Longer separation is necessary and proposed *gp-s* outperformed the *ch-s* (notice the difference in the y-axis scale).

With the *gp-s* and *ch-s* approaches, the goal was to test if we can obtain accurate clock drift estimation with the proposed method by relying just on one minute long sequences (instead of 30-min-long sequences). Furthermore, we also wanted to see what are the requirements on the dataset to produce a reliable drift estimate. Given that, we rearranged the whole 34 min experiment by dividing it into 30 s intervals. Afterwards, the 30 s intervals were paired so that the separation between them was $\Delta t \in (0.5, 5, 15)$ min. We compared the proposed *GP* framework approach (labeled *gp-s*) to the convex hull approach (labeled *ch-s*). The results of the drift estimation are shown in Fig. 6, illustrating drift estimate uncertainty for different interval separations and methods. The standard deviations of the drift estimates using the *gp-s* approach were $(87.74, 2.93, 0.70)$ μs/s, while *ch-s* approach yielded $(34.65, 3.61, 1.45)$ μs/s, for the $\Delta t \in (0.5, 5, 15)$ min separation, respectively. It is clear that the case $\Delta t = 0.5$ min does not provide enough information to estimate the drift, while extending the time separation between the intervals yielded significantly better results, with *gp-s* outperforming *ch-s*.

Finally, to validate the proposed method on MOCAP and camera data sensor fusion, we conducted an experiment in which we observed the reprojection error of the target position. Namely, the target position centroid computed by MOCAP is interpolated to the closest camera frame, using temporal calibration parameters, and then projected in the image using the estimated extrinsic calibration parameters, and compared to the target image centroid. In the experiment,[3] the target exhibited static and dynamic periods. From the experiment, we can see that during the dynamic periods, the reprojection error rises significantly when using just the time-of-arrivals without any delay compensation, yielding an average reprojection error of 1.9 cm. When we compensated for the network jitter caused time delay of 17.1 ms that was obtained by the GP method, the average reprojection error was reduced to 1.0 cm. Furthermore, by using the local timestamps approach and GP, i.e., the *gp-f* method, the average reprojection error was further reduced to 0.5 cm.

From all the aforementioned results, we can conclude that using time-of-arrival strongly limits the accuracy of the temporal calibration method, while it does provide an estimate of the network delay. On the other hand, the local timestamps approach provides a time-delay estimate that is more accurate by an order of magnitude, but requires drift estimation.

C. Radar and Camera Calibration

In this experiment, we used a single PointGrey camera and a Delphi short range radar—a sensor combination commonly applied in automotive applications for tracking of moving targets, since radars are known to be robust to diverse weather conditions and offer long range with wide field of view. However, current radars have a substantial field-of-view in the elevation, but no elevation angle measurements, which makes the extrinsic calibration challenging [41]. Given that, the inability to recover a 3-D position of the target violates our main assumption of the proposed calibration.

In the sequel, we describe how we adapt our method to tackle this scenario. To the best of the authors' knowledge, this is a first attempt of temporal calibration involving an automotive radar sensor.

To address the lack of 3-D position measurements in the radar data, we propose a two-step approach. The first step, labeled *gp-3 d*, neglects the 2-D nature of the radar, and assigns a fictive ${}^{r}p_z = 0$ position measurement to the radar data, i.e., we assume that all the measurements have zero height, thus, this step does not require extrinsic calibration *a priori*. Then, the second step, labeled *gp-2 d*, builds upon the results of the *gp-3 d* calibration by projecting the 3-D camera measurements onto the 2-D radar plane. The projected 2-D camera measurements are then used to generate a new 2-D GP from which refine the calibration. To verify the accuracy of the proposed method, we conducted experiments consisting of 30 1-min intervals with a moving calibration target. During the experiment, we moved the target in the area where camera and radar field of view overlap, while trying to avoid motions unobservable to the radar (an example is given in the accompanying video). Since quality of radar detection degrades in confined spaces, these experiments were conducted in a large open hall.

The estimated time delay using the *gp-3 d* step followed the Gaussian distribution $\mathcal{N}(21.81, 1.23)$ ms, while the *gp-2 d* refinement step produces results with the distribution $\mathcal{N}(21.89, 1.07)$ ms. The results show that the first step, even though neglecting the 3-D nature, was able to produce a good time-delay estimate without the prior knowledge of extrinsic calibration parameters, while accounting for the 2-D nature of the radar produced results with slightly lower standard deviation. Additionally, we note that in these experiments, we used arrival times, since using the local timestamps did not provide better results. Probable explanation is that the radar's accuracy of position measurements introduces more uncertainty than does the communication channel jitter. Thus, the estimated time delay shows that the relative latency between the sensors was approximately 22 ms.

Since radar can introduce a higher rate of outliers than other sensors analyzed in this article, we also studied their effect on the calibration (see Fig. 7). Radar outliers mostly occurred when the azimuth measurements were around 0° and are probably caused by limited radar resolution and internal data processing. Fig. 7 depicts radar measurements, camera, and radar GP posterior means in the y-direction after the calibration. Fig. 7(a) shows the case with low outlier rate, where the GP posterior mean was not affected by corrupted measurements due to relying on the motion prior. On the other hand, Fig. 7(b) illustrates the effect of high outlier rate, where we can notice strong corruption of the radar posterior mean. Given that, during calibration we analyze the deviations of measurements from the estimated posterior and discard those above a certain threshold; thereafter, the GP regression on the radar data is recomputed.

With this experiment, we showed that our method can be easily adapted to a sensor calibration scenario, which violates the main assumption: availability of the target's 3-D position measurements. Furthermore, we examined the influence of the outliers and showed how we can leverage the GP motion prior to mitigate their influence.

D. 3-D Lidar and Camera Calibration

In this experiment, we used a single PointGrey camera and a 3-D lidar Velodyne 32E. While the camera, with the global shutter imaging sensor, takes images at discrete-time instances, the lidar head sweeps the environment in a continuous manner. Conventionally, despite the continuous nature of the lidar motion, a single sweep of data is most commonly packed in one point cloud, with the timestamp corresponding to the beginning or the end of the sweep. However, to obtain accurate temporal calibration with a moving target, we need to be able to interpolate timestamps between the beginning and end of the sweep.

Therefore, continuous-time representations such as GPs are necessary.

[3]The experiments are shown in the accompanying video are [online]. Available: https://youtu.be/vqTR6zMIKJs

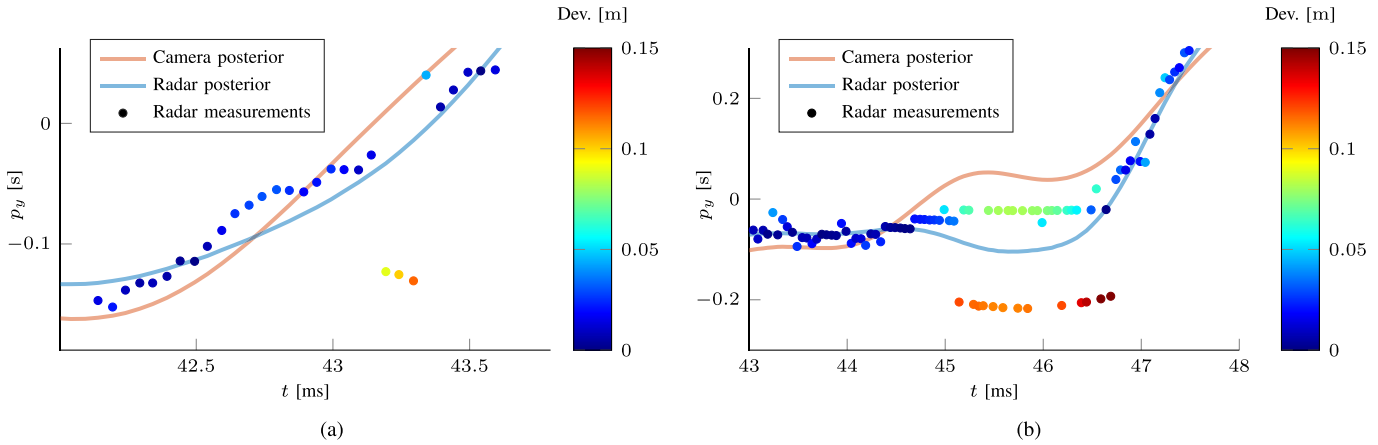


Fig. 7. Effect of outliers on GP regression showing raw radar measurements, camera, and radar GP mean posteriors (with applied extrinsic calibration) in the y-direction. Deviation between the radar posterior and measurements is color coded. Low outlier rate is virtually ignored by the posterior due to relying on the GP motion prior and as such could be used directly for temporal calibration. On the other hand, high outlier rate introduces significant discrepancy between camera and radar posterior (radar posterior is "pulled down" by the outliers), thus, a measurement validation process needs to be introduced where large deviations from the mean are ignored and the mean is recomputed. (a) Low outlier rate. (b) High outlier rate.

Since our method requires an exact 3-D position of a target, we used an isosceles triangle [side lengths (38,54,54) cm], which enables unambiguous target localization in a sparse point cloud [45]. Furthermore, we have developed a real-time triangle detection and tracking algorithm, which is also available as part of the provided toolbox. Briefly, once the algorithm segments planes in the point cloud, it fits the lines to the edges of the planes. Intersections of the lines are then used as vertex hypotheses, which are compared to the triangle model vertices. The solution is accepted if the error does not surpass a predefined threshold.

To address lidar's continuous sweep, we compensate target's timestamps by the azimuth angle of the target detection. We form the point cloud by using a fixed cut angle $\theta_{\text{cut}} = \pi$, i.e., all new data are packed into a point cloud when the driver receives a new measurement at the azimuth angle θ_{cut}, at which point the latest timestamp is assigned to the point cloud. We assume a constant rotational velocity of the lidar with frequency 10 Hz and subtract the point cloud timestamps proportionally to the angular distance between the cut angle and the current target azimuth angle. Finally, we used the local timestamps, which introduced a slight drift; thus, using the *ch-f* approach as in Section IV-B, we estimated the relative drift of 1.33 μs/s and compensated the timestamps accordingly. The results of the time-delay estimation were in the range of ± 0.85 ms around the mean value with estimated Gaussian distribution $\mathcal{N}(78.32, 0.42)$ ms. The results are comparable to the calibration of synchronized cameras in Section IV-A, despite the challenging factors such as sensor asynchronicity, lidar's continuous sweep, and twice lower sampling rate. Thus, we can assert that the method is precise up to the fraction of the fastest sensor.

To evaluate calibration qualitatively, we conducted a data fusion experiment and tested it on a validation dataset not used in calibration. We overlaid camera images with the segmented triangle points from the point clouds. To synchronize the images and the point clouds, we compensated for the estimated time delay and chose the point cloud closest in time to the current image. Finally, to align the point cloud with the image, we translate the point cloud points using the linear interpolation based on the difference vector of the two consecutive triangle positions estimated by the tracker that surround the current image. A preview of the results is shown in Fig. 8, while the accompanying video also shows this experiment. From Fig. 8 and video, we can notice an excellent performance of the triangle tracker and accuracy of temporal and extrinsic calibration. Static periods corroborate the accuracy of the extrinsic calibration, as they show consistent overlap of the triangle in the image and segmented lidar points, as illustrated in Fig. 8(a). Dynamic periods also exhibit proper alignment, corroborating temporal calibration accuracy, and in Fig. 8(b) and (c), we illustrate typical worst cases that appear during vertical and sideways motions, respectively. In addition, Fig. 8(d) shows that the developed tracker works properly even when the triangle is only partially visible by the 3-D lidar.

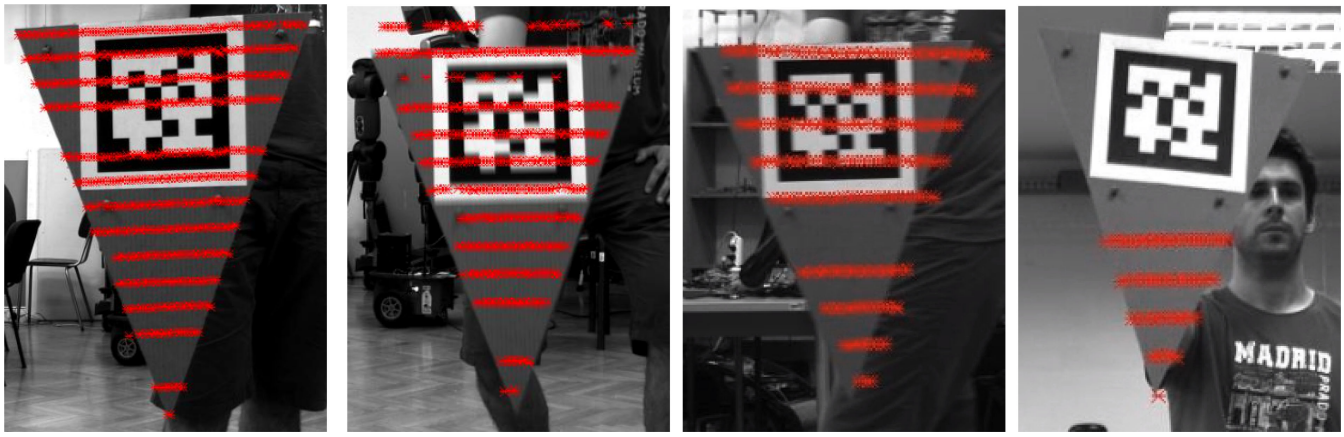
This experiment showed that the proposed method is well-suited for handling sensors with continuous motion affecting data acquisition and combining them with discrete acquisition sensors, such as cameras. Furthermore, the data fusion experiment showed a robust performance of the developed target tracker and further confirmed the calibration results.

TABLE III
STANDARD DEVIATION OF THE ESTIMATED EXTRINSIC CALIBRATION PARAMETERS

	Stereo camera	MOCAP–Camera	3D Lidar–Camera	Radar–Camera
${}^{1}t_{2,x}$ [m]	2.79×10^{-3}	4.00×10^{-4}	6.41×10^{-4}	9.26×10^{-3}
${}^{1}t_{2,y}$ [m]	4.79×10^{-3}	2.43×10^{-4}	2.06×10^{-3}	1.94×10^{-2}
${}^{1}t_{2,z}$ [m]	2.11×10^{-3}	6.24×10^{-4}	3.64×10^{-3}	5.44×10^{-2}
${}^{1}_{2}\theta_{z}$ [°]	1.52×10^{-1}	9.75×10^{-3}	6.26×10^{-2}	2.40×10^{-1}
${}^{1}_{2}\theta_{y}$ [°]	5.79×10^{-2}	3.22×10^{-2}	1.65×10^{-1}	7.88×10^{-1}
${}^{1}_{2}\theta_{x}$ [°]	3.88×10^{-2}	1.67×10^{-2}	8.74×10^{-2}	6.84×10^{-1}

E. Extrinsic Calibration

In this section, we provide extrinsic calibration results for the four previously described experiments. Table III shows estimated standard deviations of the individual extrinsic calibration



(a) (b) (c) (d)

Fig. 8. Images showing 3-D lidar and camera sensor fusion results after proposed spatiotemporal calibration. Static case confirms the validity of extrinsic calibration results, while the dynamic cases illustrate worst case motion introduced error. Detections are present even when the triangle is partially visible. (a) Static period. (b) Vertical motion. (c) Sideways motion. (d) Partially visible.

parameters obtained by analyzing the results of one-minute intervals.[4] Uneven uncertainty among different sensor combinations is primarily caused by the involved sensor precision, e.g., MOCAP and camera calibration produces an order of magnitude lower uncertainty due to the high precision of the MOCAP system. Furthermore, some variations in the uncertainty of the extrinsic parameters within each sensor combination is most likely caused by an uneven excitation of different calibration directions in the datasets. Finally, extrinsic calibration of the radar and the camera resulted with a 6-D transform whose translation in the z-axis, and Euler angles about the y- and x-axis, had higher variance than the other counterparts. This effect is caused by the lack of the radar's elevation measurements and the interested reader is directed to [40] for a detailed analysis.

F. Multisensor Experiment

To test the graph-based approach presented in Section II-D, we evaluated it on the lidar–camera–MOCAP setup where all three sensors shared the same field of view, i.e., we had a fully connected three-node graph. Here, we focused on the time-delay estimation, and thus, we preprocessed the timestamps to remove the drift using the *gp-f* approach, while we analyzed results using 30 1-min intervals. We did not notice any significant differences between the pairwise and graph-based results in terms of delay precision. Namely, standard deviations of time-delay estimates for the joint/pairwise sensor combinations were lidar–camera 0.41/0.42 ms; lidar–MOCAP 0.37/0.38 ms; and camera–MOCAP 0.12/0.12 ms. However, we did notice a significant impact on the consistency of the solution when using the pairwise approach, which is inherently solved using the graph-based approach. Therefore, we used the same consistency test as in Section III-A by *closing the loop* in the graph. The experiment yielded the following results:

1) average rotational error of 0.03° with the maximum of 0.45°;
2) average translational error of 0.5 mm with the maximum of 5.9 mm;
3) average absolute time-delay error of 0.1 ms with the maximum of 1.2 ms.

Note again that for the graph-based approach these errors are zero.

G. Implementation Details

The computation performance of the proposed method was tested on 40 datasets from Section IV-A problem of the stereo pair spatiotemporal calibration. The calibration starts with two separate GP regressions for each sensor that are completely decoupled and performed in separate threads. On average, one-minute intervals consisted of 1138 measurements requiring $t_{\text{GP}} = 49$ ms for a complete GP regression. After the GP regression, we performed the GN optimization to obtain extrinsic and temporal calibration parameters. For the stereo pair problem, which had hardware ensured zero time delay, when the optimization was initialized at $t_d = 0.5$ s, ${}^f_i\boldsymbol{R} = \mathbb{I}^{3\times3}$, and ${}^f_i\boldsymbol{t} = [0\ 0\ 0]^T$ m, it took around six iterations to converge, which translated to the average optimization time of $t_{\text{opt}} = 41$ ms. Finally, the total time required for the delay estimation was on average $t_{\text{total}} = t_{\text{GP}} + t_{\text{opt}} = 90$ ms.[5] In general, we handle missing measurements and varying sample times; however, under the assumption of constant sample rates and absence of missing measurements, further improvements on the GP regression performance are possible through offline construction of the required batch matrices. It is also important to point out that the algorithm time complexity is $\mathcal{O}(n)$, which makes the method well scalable, especially for sensors with high frame rates or longer experiments.

[4]Indices 1 and 2 denote first and second sensor for a specific experiment.

[5]Machine used for testing had i7-6700HQ CPU at 2.6 GHz × 8 and 16 GB of 2133 MHz DDR4 RAM.

Considering the effect of the process noise $\boldsymbol{Q}_c$ on the performance of the method, we found that it is fairly resilient. In scenarios with higher outlier rate (e.g., radar experiment), an optimal $\boldsymbol{Q}_c$ can be found which mitigates the influence of the outliers. However, in scenarios with low or zero outlier rate (e.g., simulations or the camera and MOCAP experiment), choosing any reasonable $\boldsymbol{Q}_c$ that does not suppress the measurements in favor of the motion model leads to the same results.

V. Conclusion

In this article, we have proposed a spatiotemporal multisensor calibration method based on GPs moving target tracking. The proposed method relies on the target positions in joint spatiotemporal calibration, while it can also estimate clock drift and the time delay. Method efficiency is achieved by relying on exactly sparse GP regression for target trajectory representation and on-manifold optimization framework. Furthermore, the method is applicable to any multisensor setup with arbitrary number of sensors, as long as sensors can estimate the 3-D position of a moving target.

We have validated the proposed calibration method in extensive simulation and real-world experiments on four multisensor setups. The first setup consisted of two externally triggered cameras, demonstrating the validity of our method on vision sensors with a readily available ground truth. The second setup consisted of a single camera and a motion capture system, demonstrating the proposed method on a heterogeneous sensor setup with significant difference in frame rates and communication over a wireless network. The third setup analyzed a common automotive heterogeneous sensor fusion setup of a single camera and radar—a challenging calibration setup due to radar's lack of elevation measurement. The fourth setup incorporated a rotating 3-D lidar with a single camera, demonstrating the validity of the method on the fusion of a continuous sweeping sensor and a discrete-time acquisition sensor. Where applicable, we compared the proposed method to the state-of-the-art approaches and the results showed that the proposed method outperformed other approaches and that it reliably estimated the time delay up to a fraction of the sampling rate of the faster sensor. In the end, we discussed the computational complexity of the proposed method and the influence of hyperparameters, mainly the process noise used in the GP regression.

The subject of future research and the potential of the proposed method is to serve as the base for online calibration of autonomous vehicle or robot heterogeneous sensors by tracking multiple moving targets in the environment—an information that is potentially already available in most autonomous systems navigating in dynamic environments.

References

[1] A. Richardson, J. Strom, and E. Olson, "AprilCal: Assisted and repeatable camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013, pp. 1814–1821.

[2] J. K. Huang, M. Ghaffari, R. Hartley, L. Gan, R. M. Eustice, and J. W. Grizzle, "LiDARTag: A real-time fiducial tag using point clouds," 2020. [Online]. Available: https://github.com/UMich-BipedLab/LiDARTag

[3] T. Scott, A. A. Morye, P. Pinies, L. M. Paz, I. Posner, and P. Newman, "Choosing a time and place for calibration of lidar-camera systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 4349–4356.

[4] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Proc. Robot.: Sci. Syst.*, pp. 1–8 2013, pp. 1–8. [Online]. Available: https://scholar.google.com/scholar?cluster=8341079850197594800&hl=hr&as_sdt=0,5

[5] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale, "Online self-calibration for robotic systems," *Int. J. Robot. Res.*, vol. 35, no. 4, pp. 357–380, 2015.

[6] N. Keivan and G. Sibley, "Online SLAM with any-time self-calibration and automatic change detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5775–5782.

[7] A. Fornaser, P. Tomasin, M. De Cecco, M. Tavernini, and M. Zanetti, "Automatic graph based spatiotemporal extrinsic calibration of multiple kinect V2 ToF cameras," *Robot. Auton. Syst.*, vol. 98, pp. 105–125, 2017.

[8] F. Faion, M. Baum, A. Zea, and U. D. Hanebeck, "Depth sensor calibration by tracking an extended object," in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2015, pp. 19–24.

[9] P. C. Su, J. Shen, W. Xu, S. C. S. Cheung, and Y. Luo, "A fast and robust extrinsic calibration for RGB-D camera networks," *Sensors*, vol. 18, no. 1, pp. 1–23, 2018.

[10] F. Lv, T. Zhao, and R. Nevatia, "Camera calibration from video of a walking human," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1513–1518, Sep. 2006.

[11] D. F. Glas, T. Miyashita, H. Ishiguro, and N. Hagita, "Automatic position calibration and sensor displacement detection for networks of laser range finders for human tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2010, pp. 2938–2945.

[12] D. F. Glas, D. Brscic, T. Miyashita, and N. Hagita, "SNAPCAT-3D: Calibrating networks of 3D range sensors for pedestrian tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 712–719.

[13] Z. Tang, Y. S. Lin, K. H. Lee, J. N. Hwang, J. H. Chuang, and Z. Fang, "Camera self-calibration from tracking of moving persons," in *Proc. Int. Conf. Pattern Recognit.*, 2017, pp. 265–270.

[14] J. Quenzel, N. Papenberg, and S. Behnke, "Robust extrinsic calibration of multiple stationary laser range finders," in *Proc. Int. Conf. Autom. Sci. Eng.*, 2016, pp. 1332–1339.

[15] M. Huber, M. Schlegel, and G. Klinker, "Application of time-delay estimation to mixed reality multisensor tracking," *J. Virtual Reality Broadcast.*, vol. 11, no. 3, pp. 1–22, 2014.

[16] J. Kelly and G. S. Sukhatme, "A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors," *Springer Tracts Adv. Robot.*, vol. 79, pp. 195–209, 2014.

[17] J. Rehder, R. Siegwart, and P. Furgale, "A general approach to spatiotemporal calibration in multisensor systems," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 383–398, Apr. 2016.

[18] M. Li and A. I. Mourikis, "Online temporal calibration for camera-IMU systems: Theory and algorithms," *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 947–964, 2014.

[19] M. K. Ackerman, A. Cheng, B. Shiffman, E. Boctor, and G. Chirikjian, "Sensor calibration with unknown correspondence: Solving AX=XB using euclidean-group invariants," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013, pp. 1308–1313.

[20] J. Kelly, N. Roy, and G. S. Sukhatme, "Determining the time delay between inertial and visual sensor measurements," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1514–1523, Dec. 2014.

[21] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1215–1229, Oct. 2016.

[22] T. Qin and S. Shen, "Temporal calibration for monocular visual-inertial systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 3662–3669.

[23] E. Mair, M. Fleps, M. Suppa, and D. Burschka, "Spatio-temporal initialization for IMU to camera registration," in *Proc. Int. Conf. Robot. Biomimetics*, 2011, pp. 557–564.

[24] H. Sommer, R. Khanna, I. Gilitschenski, Z. Taylor, R. Siegwart, and J. Nieto, "A low-cost system for high-rate, high-accuracy temporal calibration for LIDARs and cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2017, pp. 2219–2226.

[25] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative B-splines on lie groups," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11145–11153.

[26] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[27] T. D. Barfoot, *State Estimation for Robotics*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[28] T. Barfoot, C. H. Tong, and S. Sarkka, "Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression," in *Proc. Robot.: Sci. Syst.*, 2014, pp. 1–9.
[29] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression," *Auton. Robot.*, vol. 39, no. 3, pp. 221–238, 2015.
[30] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 9–15.
[31] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Trans. Signal Process.*, vol. 63, no. 63, pp. 4165–4178, Aug. 2015.
[32] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1360–1367.
[33] G. Grisetti, T. Guadagnino, I. Aloise, M. Colosi, B. Della Corte, and D. Schlegel, "Least squares optimization: From theory to practice," *Robot.*, vol. 9, no. 3, 2020, Art. no. 51.
[34] M. Huber, M. Schlegel, and G. Klinker, "Temporal calibration in multisensor tracking setups," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2009, pp. 195–196.
[35] C. Herzberg, R. Wanger, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Inf. Fusion*, vol. 14, no. 1, pp. 57–77, 2013.
[36] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," Institut de Robótica i Informática Industrial, Tech. Rep. IRI-TR-18-01, 2018. [Online]. Available: https://github.com/artivis/manif/blob/devel/docs/pages/publication.md
[37] R. Kruno, C. Josip, M. Ivan, and P. Ivan, "Exactly sparse delayed state filter on Lie groups for long-term pose graph SLAM," *Int. J. Robot. Res.*, vol. 37, no. 6, pp. 585–610, 2018, doi: 10.1177/0278364918767756.
[38] R. Wagner, O. Birbach, and U. Frese, "Rapid development of manifold-based graph optimization systems for multi-sensor calibration and SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2011, pp. 3305–3312.
[39] B. D. Corte, H. Andreasson, T. Stoyanov, and G. Grisetti, "Unified motion-based calibration of mobile multi-sensor platforms with time delay estimation," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 902–909, Apr. 2019.
[40] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2016, pp. 4193–4198.
[41] J. Peršić, I. Marković, and I. Petrović, "Extrinsic 6DoF calibration of a radar-LiDAR-camera system enhanced by radar cross section estimates evaluation," *Robot. Auton. Syst.*, vol. 114, pp. 217–230 2019.
[42] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2013, pp. 473–480.
[43] D. Zuñiga-Noël, J. R. Ruiz-Sarmiento, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, "Automatic multi-sensor extrinsic calibration for mobile robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2862–2869, Jul. 2019.
[44] L. Zhang, Z. Liu, and C. H. Xia, "Clock synchronization algorithms for network measurements," in *Proc. IEEE INFOCOM*, vol. 1, no. c, pp. 160–169, 2002.
[45] S. Debattisti, L. Mazzei, and M. Panciroli, "Automated extrinsic laser and camera inter-calibration using triangular targets," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2013, pp. 696–701.

Juraj Peršić (Student Member, IEEE) received the B.Sc. degree in electrical engineering and the M.Sc. degree in electrical engineering in 2014 and 2016, respectively, from the Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Zagreb, Croatia, where he is currently working toward the Ph.D. degree in robotics under Prof. Ivan Petrović.

He has been a Researcher on the SafeTRAM project since September 2016 with FER. His main areas of interest are mobile robotics with focus on sensor calibration and fusion.

Luka Petrović (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 2015 and 2017, respectively.

His main research interests are in the areas of autonomous systems and robotics with focus on high-dimensional motion planning.

Mr. Petrović, during his graduate studies, was awarded with the Rector's Award for a practical application in the field of robotics, 2016 and the Bronze Plaque "Josip Lončar" faculty award for outstanding academic achievement, in 2017.

Ivan Marković (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Zagreb, Zagreb, Croatia, in 2008 and 2014, respectively.

He is currently an Associate Professor with the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia. He was a Visiting Researcher with INRIA RennesBretagne Atlantique, Rennes, France, under the supervision of Prof. François Chaumette. His research interests include estimation theory with applications to autonomous mobile robotic systems.

Dr. Marković was the recipient of the Croatian Academy of Engineering Young Scientist Award "Vera Johanides," in 2018. He is also a member of the IFAC Technical Committee on Robotics and is the Vice-President of the Croatian IEEE RAS chapter.

Ivan Petrović (Member, IEEE) received B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 1983, 1989 and 1998, respectively.

He is currently a Professor and the Head of the Laboratory for Autonomous Systems and Mobile Robotics, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia. He has authored or coauthored more than 60 journal and 200 conference papers. His research interests include advanced control and estimation techniques and their application in autonomous systems and robotics.

Prof. Petrović is a Full Member of the Croatian Academy of Engineering, and the Chair of the IFAC Technical Committee on Robotics.

PUBLICATION 5

E. Wise, J. Peršić, C. Grebe, I. Petrović and J. Kelly. A Continuous-Time Approach for 3D Radar-to-Camera Extrinsic Calibration. *IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China, 2021 (accepted).

A Continuous-Time Approach for 3D Radar-to-Camera Extrinsic Calibration

Emmett Wise[1], Juraj Peršić[2], Christopher Grebe[1], Ivan Petrović[2], and Jonathan Kelly[1,†]

***Abstract*— Reliable operation in inclement weather is essential to the deployment of safe autonomous vehicles (AVs). Robustness and reliability can be achieved by fusing data from the standard AV sensor suite (i.e., lidars, cameras) with *weather robust* sensors, such as millimetre-wavelength radar. Critically, accurate sensor data fusion requires knowledge of the rigid-body transform between sensor pairs, which can be determined through the process of extrinsic calibration. A number of extrinsic calibration algorithms have been designed for 2D (planar) radar sensors—however, recently-developed, low-cost 3D millimetre-wavelength radars are set to displace their 2D counterparts in many applications. In this paper, we present a continuous-time 3D radar-to-camera extrinsic calibration algorithm that utilizes radar velocity measurements and, unlike the majority of existing techniques, does not require specialized radar retroreflectors to be present in the environment. We derive the observability properties of our formulation and demonstrate the efficacy of our algorithm through synthetic and real-world experiments.**

I. Introduction

Safety is a paramount concern for autonomous vehicles (AVs) operating in human-centric environments (e.g., self-driving cars travelling on city streets). To reduce the risk of failure and improve robustness, most AVs fuse data from multiple sensors on board. The standard AV sensor suite typically includes cameras and lidar units; while these sensors are able to provide a high degree of situational awareness, they may fail to work reliably in inclement weather (e.g., heavy rain or snowfall). In turn, many AV sensor platforms incorporate 2D (planar) millimetre-wavelength radar units that are *weather robust*—radar measurements are relatively immune to interference caused by precipitation, for example.

All radar sensors operate on the same basic principle: a low-frequency electromagnetic (EM) pulse is emitted from the radar antenna, reflects off of radar-opaque targets in the environment, and returns to the sensor. By measuring the time of flight and phase of the return pulse, the radar is able to determine the azimuth, range, range-rate (velocity in the radial direction), and cross-section (reflectivity) of targets. Low-frequency EM waves are able to pass through rain, snow, and other obscurants [1]. Although 2D radar has proven useful for many AV applications, the lack of complete 3D information limits its utility in many cases.

[1]Emmett Wise, Christopher Grebe, and Jonathan Kelly are with the Space & Terrestrial Autonomous Robotics Systems (STARS) Laboratory at the University of Toronto Institute for Aerospace Studies, Toronto, Canada. <firstname>.<lastname>@robotics.utias.utoronto.ca

[2]Juraj Peršić and Ivan Petrović are with the Laboratory for Autonomous Systems and Mobile Robotics, University of Zagreb Faculty of Electrical Engineering and Computing, Croatia. <firstname>.<lastname>@fer.hr

[†]Jonathan Kelly is a Vector Institute Faculty Affiliate. This research was supported in part by the Canada Research Chairs program.

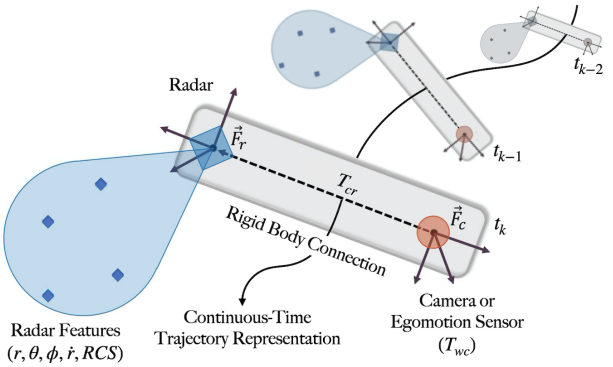


Fig. 1. Depiction of the calibration problem. The radar measures the range, azimuth, elevation, range-rate, and reflectivity of objects in the environment. The camera (or egomotion sensor) measures its own pose change relative to a fixed reference frame. Our goal is to recover the rigid-body transform $\mathbf{T}_{cr}$ between the radar unit and the camera.

More recently, low-cost 3D radar sensors, such as the Texas Instruments AWR1843BOOST, have become available. Because of the additional information contained in 3D radar measurements (i.e., elevation), 3D radars are poised to replace 2D sensors in AV systems and in other applications. To properly fuse 3D radar data with measurements from other AV sensors, however, knowledge of the rigid-body transform between the radar and the other sensors is required. The process of determining the transform is known as extrinsic calibration. Often, extrinsic calibration is performed prior to deployment, in a laboratory or factory setting; the transform parameters are prone to change, however, due to material fatigue or user modifications. Consequently, there is a need for methods to estimate the extrinsic calibration in the field.

Radar extrinsic calibration is challenging for several reasons. First, most radar measurement models assume that the EM pulse is reflected by one surface only. In reality, there are often multipath reflections from several different surfaces. These multipath reflections create measurement outliers that can obscure or 'drown out' the true reflection from a target. Second, raw radar measurements have substantial jitter, which reduces measurement precision. Finally, a radar pulse is a wave, and hence the exact point of reflection from a target can be ambiguous and/or inconsistent [2]. The low precision and high outlier rate of radar measurements can degrade estimates of the extrinsic calibration. To mitigate some of these issues, many existing calibration algorithms rely on specialized radar retroreflectors that are placed strategically in the environment. Although this approach improves calibration, specialized retroreflectors are rarely available in the field during regular operation.

We overcome the challenges of radar extrinsic calibration by relying on the *motion* of the sensor platform rather than on specific scene structure (see Fig. 1). Work by Stahoviak has shown that the velocity of a 3D millimetre-wavelength (hereafter, mm-wave) radar sensor can be determined directly and without knowledge of the environment [3]. By relying on velocity information provided by the 3D radar, instead of attempting to localize and track specific targets, we avoid many of the issues caused by noise, outliers, and jitter. We focus on radar-to-camera extrinsic calibration—however, the method we describe is applicable to any complementary sensor that is able to estimate its egomotion (e.g., 3D lidar, GNSS/INS sensors, etc.). We require only enough information for egomotion estimation and sufficient excitation of the system (see Section IV-B). In this paper we:

1) prove that extrinsic calibration for a 3D radar-camera pair is observable given sufficient excitation of the system;
2) describe the required motions necessary for proper calibration;
3) develop a continuous-time batch radar-to-monocular camera extrinsic calibration algorithm; and
4) verify the performance of our algorithm on synthetic data and through extensive real-world experiments.

We provide one of the first methods for estimating the extrinsic calibration parameters between a 3D mm-wave radar and monocular camera without the use of radar retroreflectors. Although our goal is to build weather-robust navigation platforms, we focus on calibration under nominal conditions in the field (i.e., without adverse weather), since this is already a very difficult problem.

II. Related Work

A variety of mm-wave radar extrinsic calibration algorithms exist, which can roughly be grouped according to the sensor pair involved and the specific degrees of freedom that are calibrated. Early extrinsic calibration algorithms for radar-camera sensor pairs considered 2D radar units only, either ignoring the 3D nature of radar measurements or constraining the positions of any retroreflectors to the radar measurement plane [4]–[7]. These algorithms operate by estimating the homography between the camera image plane and the radar measurement plane. Sugimoto et al. note in [4] that 2D radar units typically measure a maximum return when a retroreflector lies on the plane of zero elevation in the radar reference frame; the return intensity decreases for reflectors that lie above or below this plane. The approach in [4] filters returns by intensity to ensure that only targets in the plane at zero elevation (relative to the radar frame) are used as part of the calibration process.

More recent algorithms estimate the rigid sensor-to-sensor transform by minimizing a 'reprojection error': this is the error in the alignment of identifiable environmental structures or objects that appear within the fields of view of both sensors. Kim et al. [8] align hybrid visual-radar targets that can be easily identified in the camera and radar data, but assume that the radar measurements are constrained to the zero-elevation plane.

The zero-elevation plane constraint is relaxed for certain 'reprojection error' algorithms. El Natour et al. estimate the radar-to-camera transform by intersecting backprojected camera rays with the 'arcs' in 3D along which radar measurements must lie [9]. Domhof et al. rely on a known visual target structure to convert camera measurements into 'pseudo-radar' measurements. The transform that best aligns the radar and pseudo-radar measurements then defines the extrinsic calibration [10]. Peršić et al. [11] improve upon these methods by resolving the elevation ambiguity using target reflection intensity as a pseudo-measurement of the elevation angle. Peršić et al. [11] also extend their approach to include 2D radar-to-lidar calibration. The reprojection and homography methods are summarized and compared by Oh et al. in [12], where the authors conclude that the homography and reprojection methods have similar accuracy.

All of the algorithms described above require specialized retroreflective radar targets, but a small number of 'targetless' or target-free extrinsic calibration algorithms for 2D mm-wave radar also exist. Schöller et al. [13] use end-to-end deep learning to estimate the extrinsic rotation parameters that align vehicles (i.e., automobiles) detected in radar measurements and camera images. However, the algorithm requires an external measurement of the translation parameters. Peršić et al. [14] perform target-free, online pairwise extrinsic calibration of 2D radars, cameras, and lidar sensors by estimating the transform that aligns moving object trajectories. This method assumes a priori knowledge of the translation parameters and only estimates yaw between the radar-camera and radar-lidar pairs.

Similar to our approach, Kellner et al. [15] use radar velocity measurements to estimate the yaw angle between a 2D radar sensor and a vehicle-mounted gyroscope, by relating the angular velocity of the gyroscope to the lateral velocity of the radar. This technique also requires a priori knowledge of the translation between the sensors.

In summary, the mm-wave radar calibration algorithms developed to date are generally limited by hardware constraints (i.e., an inability to resolve elevation reliably) or the need for specialized retroreflective targets, or suffer from high calibration parameter uncertainty due to a lack of true 3D information. We take advantage of the available elevation data in 3D radar measurements to estimate the instantaneous (3D) velocity of the radar unit. These data, in combination with pose estimates from a camera (or other egomotion sensors), allow us to determine the full sensor-to-sensor rigid-body transform without the need for specialized targets.

III. Problem Formulation

A. Notation

Latin and Greek letters (e.g., a and α) represent scalar variables, while boldface lower and upper case letters (e.g., $\mathbf{x}$ and $\mathbf{\Theta}$) represent vectors and matrices, respectively. A parenthesized superscript pair, for example, $\mathbf{A}^{(i,j)}$, indicates

the ith row and the jth column of the matrix $\mathbf{A}$. A three-dimensional reference frame is designated by $\underrightarrow{\mathcal{F}}$. The translation vector from point a (often a reference frame origin) to b, expressed in $\underrightarrow{\mathcal{F}}_a$, is denoted by $\mathbf{r}_a^{ba}$. The translational velocity of point b relative to point a, expressed in $\underrightarrow{\mathcal{F}}_c$, is denoted by $\mathbf{v}_c^{ba}$. The angular velocity of frame $\underrightarrow{\mathcal{F}}_a$ relative to an inertial frame, expressed in $\underrightarrow{\mathcal{F}}_a$, is denoted by $\boldsymbol{\omega}_a$.

We denote rotation matrices by $\mathbf{R}$; for example, $\mathbf{R}_{ab} \in \mathrm{SO}(3)$ defines the rotation from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$. We reserve $\mathbf{T}$ for $\mathrm{SE}(3)$ transform matrices; for example, $\mathbf{T}_{ab}$ is the homogeneous matrix that defines the rigid-body transform from frame $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$. These transforms are constructed using the split representation of $\mathrm{SE}(3)$. For example, the transform from frame $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$ at time t is,

$$\mathbf{T}_{ab}(t) = \begin{bmatrix} \mathbf{R}_{ab}(t) & \mathbf{r}_a^{ba}(t) \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{1}$$

where the transform is split into a rotation matrix, $\mathbf{R}_{ab}(t) \in \mathrm{SO}(3)$, and translation vector, $\mathbf{r}_a^{ba}(t) \in \mathbb{R}^3$. The unary operator $^\wedge$ acts on $\mathbf{r} \in \mathbb{R}^3$ to produce a skew-symmetric matrix such that $\mathbf{r}^\wedge\mathbf{s}$ is equivalent to the cross product $\mathbf{r} \times \mathbf{s}$.

B. Sensor Measurements

We consider three reference frames: frame $\underrightarrow{\mathcal{F}}_w$ is an (approximate) inertial frame attached to the surface of the Earth, while $\underrightarrow{\mathcal{F}}_r$ is the reference frame of the radar sensor, and $\underrightarrow{\mathcal{F}}_c$ is the reference frame of the camera (or other egomotion sensor). The radar unit measures the velocity of the sensor in $\underrightarrow{\mathcal{F}}_r$ relative to $\underrightarrow{\mathcal{F}}_w$, expressed in $\underrightarrow{\mathcal{F}}_r$ at an instant in time, t,

$$\mathbf{v}_r^{rw}(t) = \mathbf{R}_{wr}(t)^T \frac{\partial \mathbf{r}_w^{rw}(t)}{\partial t}, \tag{2}$$

where we use the partial derivative notation to indicate that the radar position also depends upon the parameters of our B-spline trajectory representation (see Section III-C).

Assuming that a series of three or more (known) 3D landmarks are visible in frame $\underrightarrow{\mathcal{F}}_w$, the camera is able to measure its pose at time t relative to $\underrightarrow{\mathcal{F}}_w$,

$$\mathbf{T}_{cw}(t) = \mathbf{T}_{cr}\mathbf{T}_{wr}^{-1}(t), \tag{3}$$

where $\mathbf{T}_{wr}(t)$ is the homogeneous pose matrix of the radar in the inertial frame at time t and $\mathbf{T}_{cr}$ is the homogeneous matrix that defines the (constant but unknown) radar-to-camera transform. If the metric positions of the landmarks are not known, the camera translation can only be determined up to an unknown scale factor.

C. Continuous-Time Trajectory Representation

We use a continuous-time representation of the sensor platform trajectory in our problem formulation. The continuous-time representation is advantageous because it allows measurements to be made at arbitrary time instants; since the radar and the camera operate at different rates and are not hardware synchronized, the relationship between their measurement times is not fixed. There are multiple possible ways to parameterize trajectories in continuous time [16]–[18]. We choose the cumulative B-spline representation on Lie groups developed by Sommer et al. in [16]. Below, we very briefly review this representation, and refer the reader to [16] for more details.

B-splines are functions of one continuous parameter (e.g time) and a finite set of control points (or *knots*); for brevity, we restrict our example here to points $\{\mathbf{p}_0, \ldots, \mathbf{p}_N \mid \mathbf{p}_i \in \mathbb{R}^d\}$. The order k of the spline determines the number of control points that are required to evaluate the spline at time t. In a uniformly spaced B-spline, each control point is assigned a time $t_i = t_0 + i\Delta t$, where t_0 is the start of the spline and Δt is the time between control points. Given a B-spline of length N and order k, the end of the spline is t_{N-k+1}.

Given a time t, a normalized time $u = \frac{t-t_i}{t_{i+1}-t_i}$ can be defined, where t_i is the time assigned to control point $\mathbf{p}_i$ and $t_i \leq t < t_{i+1}$. The B-spline function evaluated at normalized time u is

$$\mathbf{p}(u) = \begin{bmatrix} \mathbf{p}_i & \mathbf{d}_1^i & \ldots & \mathbf{d}_{k-1}^i \end{bmatrix} \tilde{\mathbf{M}}_k \mathbf{u}, \tag{4}$$

where $\mathbf{u}^T = [1 \; u \; u^2 \; \ldots \; u^{k-1}]$ and $\mathbf{d}_j^i = \mathbf{p}_{i+j} - \mathbf{p}_{i+j-1}$. The matrix $\tilde{\mathbf{M}}_k$ is a $k \times k$ *mixing matrix*. The elements of the mixing matrix are a function of the spline order k and are defined by

$$\tilde{m}_k^{(a,n)} = \sum_{s=a}^{k-1} m_k^{(s,n)}, \tag{5}$$

$$m_k^{(s,n)} = \frac{C_{k-1}^n}{(k-1)!} \sum_{l=s}^{k-1} (-1)^{l-s} C_k^{l-s} (k-1-l)^{k-1-n} \quad a, s, n \in \{0, \ldots, k-1\}. \tag{6}$$

The scalar $C_j^i = \frac{j!}{i!(j-i)!}$ is a binomial coefficient. This B-splines definition can be simplified by defining $\lambda_j(u) = \tilde{\mathbf{M}}_k \mathbf{u}$, which results in

$$\mathbf{p}(u) = \mathbf{p}_i + \sum_{j=1}^{k-1} \lambda_j(u) \mathbf{d}_j^i. \tag{7}$$

This B-spline representation is a convenient way to describe smooth rigid-body trajectories in continuous time. Our development above is for splines on a vector space, but B-splines can also be defined over Lie groups, including the group $\mathrm{SO}(3)$ of rotations,

$$\mathbf{R}(u) = \mathbf{R}_i \prod_{j=1}^{k-1} \exp(\lambda_j(u) \boldsymbol{\phi}_j^i), \tag{8}$$

where $\mathbf{R}_i$ is a control point of the rotation spline and $\boldsymbol{\phi}_j^i = \log(\mathbf{R}_{i+j-1}^T \mathbf{R}_{i+j})$. The operators exp and log map from the Lie algebra $\mathfrak{so}(3)$ to $\mathrm{SO}(3)$ and vice versa, respectively [18].

D. Optimization Problem

The error equation for the radar velocity is

$$\mathbf{e}_v(t) = \mathbf{v}_r^{rw}(t) - \mathbf{R}_{wr}(t)^T \frac{\partial \mathbf{r}_w^{rw}(t)}{\partial t} + \mathbf{n}_v, \quad \mathbf{n}_v \sim \mathcal{N}(0, \boldsymbol{\Sigma}_v(t)), \tag{9}$$

where $\mathbf{R}_{wr}(t)$ and $\mathbf{r}_w^{rw}(t)$ are the split spline representation of $\mathbf{T}_{wr}(t)$ with control points $\{\mathbf{R}_0, \ldots, \mathbf{R}_N \mid \mathbf{R}_i \in \mathrm{SO}(3)\}$ and $\{\mathbf{p}_0, \ldots, \mathbf{p}_N \mid \mathbf{p}_i \in \mathbb{R}^3\}$. The vector $\mathbf{v}_r^{rw}(t)$ is the measured radar velocity at time t. The error equation for the camera measurements is

$$\mathbf{T}_{err}(t) = \mathbf{T}_{cw}(t)\mathbf{T}_{wr}(t)\mathbf{T}_{cr}^{-1} \tag{10}$$

$$\mathbf{e}_p(t) = \begin{bmatrix}\mathbf{r}_{err}(t)\\ \boldsymbol{\phi}_{err}(t)\end{bmatrix} + \mathbf{n}_p,\ \mathbf{n}_p \sim \mathcal{N}(0, \boldsymbol{\Sigma}_p(t)) \tag{11}$$

$$\boldsymbol{\phi}_{err}(t) = \log(\mathbf{R}_{err}(t)), \tag{12}$$

where $\mathbf{r}_{err}(t)$ and $\mathbf{R}_{err}(t)$ are the $\mathbb{R}^3$ and $\mathrm{SO}(3)$ elements of $\mathbf{T}_{err}(t)$. The set of parameters, $\mathbf{x}$, that we wish to estimate are the control points of the split representation of $\mathbf{T}_{wr}(t)$ and the extrinsic calibration parameters in $\mathbf{T}_{cr}$,

$$\mathbf{x} = \{\mathbf{p}_0, \quad \ldots, \quad \mathbf{p}_N, \quad \mathbf{R}_0, \quad \ldots, \quad \mathbf{R}_N, \quad \mathbf{R}_{cr}, \quad \mathbf{r}_c^{rc}\}. \tag{13}$$

Our optimization problem is then to find $\mathbf{x}^*$ that minimizes the following cost function:

$$\begin{aligned}\mathcal{J}(\mathbf{x}) = &\sum_{i=1}^{l} \mathbf{e}_v^T(t_i)\boldsymbol{\Sigma}_v^{-1}(t_i)\mathbf{e}_v(t_i)\\ &+ \sum_{j=1}^{m} \mathbf{e}_p^T(t_j)\boldsymbol{\Sigma}_p^{-1}(t_j)\mathbf{e}_p(t_j),\end{aligned} \tag{14}$$

where l and m are, respectively, the number of radar velocity measurements and camera pose measurements.

E. Implementation Details

Our approach to estimate the velocity of the radar unit involves finding the velocity vector that best fits a series of measured range-rate vectors. To do so, we use an algorithm and software package developed by Stahoviak et al. called 'Goggles' [3].[1] The Goggles algorithm applies MLESAC to find an inlier set of radar velocity measurements. The final velocity estimate is calculated using orthogonal distance regression on this inlier set of velocities.

We solve the full batch nonlinear optimization problem to determine the extrinsic parameters using the Levenberg-Marquardt implementation available in the Ceres solver [19]. Ceres' auto-differentiation capability is applied to calculate the Jacobians of the error equations. To manipulate the B-splines, we rely on the library from Sommer et al. [16].[2] Our translation and rotation splines have a spline order of $k = 4$.

IV. Observability Analysis

In order to estimate the calibration parameters, the system must be observable (or, equivalently for our batch formulation, identifiable). In Section IV-A, we make use of the observability rank condition criterion defined by Hermann and Krener [20] to prove that the calibration and scale estimation problem is observable. It is well known that, in the absence of metric distance information, absolute scale cannot be recovered from monocular camera measurements alone [21]. We show below that, given radar velocity data, it is possible to identify both the calibration parameters and the visual scale factor *without* knowledge of the (metric) distances between visual landmarks. It follows that radar-to-camera calibration, in the general case, does not require a specialized camera calibration target (or any other external source of scale information). We are concerned with the following set of parameters:

$$\mathbf{x} = \{\mathbf{r}_c^{rc}, \quad \mathbf{R}_{cr}, \quad \alpha\}, \tag{15}$$

where α is the unknown scale factor that appears in the camera pose measurement. A brief degeneracy analysis of the calibration problem, which identifies conditions that result in a loss of observability, is provided in Section IV-B.

A. Observability of Radar-to-Camera Extrinsic Calibration

We follow an approach similar to that in [22] and note that the (scaled) linear and angular velocities of the camera can be determined by taking the time derivatives of the camera pose measurements. Also, Stahoviak has shown that the 3D velocity of the radar (in the radar frame) can be recovered from three non-coplanar range-rate measurements [3]. These quantities can be related through rigid-body kinematics,

$$\mathbf{h}_i = \alpha\mathbf{v}_c^{cw} = \alpha(\mathbf{R}_{cr}\mathbf{v}_r^{rw} - \boldsymbol{\omega}_c^{\wedge}\mathbf{r}_c^{rc}), \tag{16}$$

where $\mathbf{h}_i$ is the scaled linear velocity of the camera and $\boldsymbol{\omega}_c$ is the angular velocity of the camera, both relative to the camera frame. To decrease the notational burden going forward, we drop the superscripts and subscripts defining the velocities and extrinsic transform parameters. The gradient of the zeroth-order Lie derivative of the ith measurement is

$$\nabla_{\mathbf{x}} L_0\mathbf{h}_i = \begin{bmatrix}-\alpha\boldsymbol{\omega}_i^{\wedge} & -\alpha(\mathbf{R}\mathbf{v}_i)^{\wedge}\mathbf{J} & \mathbf{R}\mathbf{v}_i - \boldsymbol{\omega}_i^{\wedge}\mathbf{r}\end{bmatrix}, \tag{17}$$

where $\mathbf{J}$ is the Lie algebra left Jacobian of $\mathbf{R}_{cr}$ [18]. Since the parameters of interest are constant with respect to time, we are able to stack the gradients of several Lie derivatives (at different points times) to form the observability matrix,

$$\mathbf{O} = \begin{bmatrix}\nabla_{\mathbf{x}} L_0\mathbf{h}_1\\ \nabla_{\mathbf{x}} L_0\mathbf{h}_2\\ \nabla_{\mathbf{x}} L_0\mathbf{h}_3\end{bmatrix}, \tag{18}$$

which has full column rank when three or more sets of measurements are available (we omit the full proof for brevity). We note that the analysis is simplified by considering the measurement equation only, and at different points in time. However, it is also possible to show that the system is instantaneously locally weakly observable when the sensor platform undergoes both linear and angular accelerations (again, we omit this proof due to space).

B. Degeneracy Analysis

The conditions under which a loss of observability (identifiability) may occur can be determined by examining the nullspace of the observability matrix. In this section, we consider the scale parameter to be known, which removes the last column of the matrix defined by Eq. 17—in turn,

[1] Available at https://github.com/cstahoviak/goggles

[2] Available at https://gitlab.com/VladyslavUsenko/basalt-headers.git

only two sets of measurements are required. The nullspace of $\nabla_{\mathbf{x}} L_0 \mathbf{h}_i$ contains the vectors

$$\mathbf{U}_i = \begin{bmatrix} \boldsymbol{\omega}_i & \mathbf{0} & (\mathbf{I} - \frac{\boldsymbol{\omega}_i \boldsymbol{\omega}_i^T}{\|\boldsymbol{\omega}_i\|^2})\mathbf{R}\mathbf{v}_i \\ \mathbf{0} & \mathbf{J}^{-1}\mathbf{R}\mathbf{v}_i & (\mathbf{I} - \frac{\mathbf{J}^{-1}\mathbf{R}\mathbf{v}_i(\mathbf{J}^{-1}\mathbf{R}\mathbf{v}_i)^T}{\|\mathbf{J}^{-1}\mathbf{R}\mathbf{v}_i\|^2})\mathbf{J}^{-1}\boldsymbol{\omega}_i \end{bmatrix}, \tag{19}$$

where each column of $\mathbf{U}_i$ defines one null vector. To ensure that the stacked observability matrix formed from $\nabla_{\mathbf{x}} L_0 \mathbf{h}_1$ and $\nabla_{\mathbf{x}} L_0 \mathbf{h}_2$ has full column rank (i.e., that the nullspace contains the zero vector only), the following constraints must be satisfied, at minimum:

$$\begin{aligned} \boldsymbol{\omega}_2 \times \boldsymbol{\omega}_1 &\neq 0, \\ \mathbf{v}_2 \times \mathbf{v}_1 &\neq 0. \end{aligned} \tag{20}$$

The constraints defined by Eq. 20 show that the system must rotate about and translate along two non-collinear axes at different points in time. The rotation constraint is expected because our problem is similar to the one defined by Brookshire and Teller in [23]. However, the angular velocity of the radar unit cannot be measured directly, which leads to the second excitation requirement. Additional constraints can be generated from the third column of Eq. 19, but these motions are more difficult to characterize; we posit, based on our experiments, that these constraints are less likely to be violated in practice.

V. Experiments and Results

In general, our algorithm can be applied to any 3D radar and egomotion sensor pair, but our experimental focus is on 3D radar-to-monocular camera extrinsic calibration. For convenience, in this work, we estimate the camera pose relative to a 12×10 planar checkerboard calibration target of known size. However, as shown in Section IV, knowledge of metric scale is not required—the camera must simply view a sufficient number of features (three or more) that lie in a general configuration in the environment.

Below, we present a series of synthetic and real world calibration experiments to evaluate the performance of our algorithm. In Section V-A, we empirically analyze the sensitivity of the algorithm to measurement noise when applied to synthetic data. In Section V-B, we demonstrate that our approach improves upon hand-measured calibration and compares favourably with the algorithm of Peršić et al. [24], although our approach does not require specialized radar retroreflectors.

A. Synthetic Data

Our simulation environment is shown in Fig. 2. In order to ensure sufficient excitation of the system, the sensor platform trajectory has non-zero linear and angular acceleration about all three axes in the radar sensor frame; see the bottom of Fig. 2. We added zero-mean Gaussian noise to each radar and camera measurement, with magnitudes similar to the noise levels identified in our real-world experiments.

Simulation results show that our algorithm is accurate in the low-noise regime, but that the performance degrades as the amount of noise in the radar velocity measurements increases (see Figure 3). We found that the average standard deviations of our real-world radar velocity estimates were 0.03, 0.06, and 0.1 m/s in the x, y, and z directions, respectively. As a result, our noisiest simulation experiment represents a worst-case calibration scenario, because the experiment uses twice the amount of noise as found in our true radar velocity data. Overall, the proposed calibration algorithm shows robustness to significant noise—we are able to successfully calibrate in all of our trials despite very large worst-case noise levels.

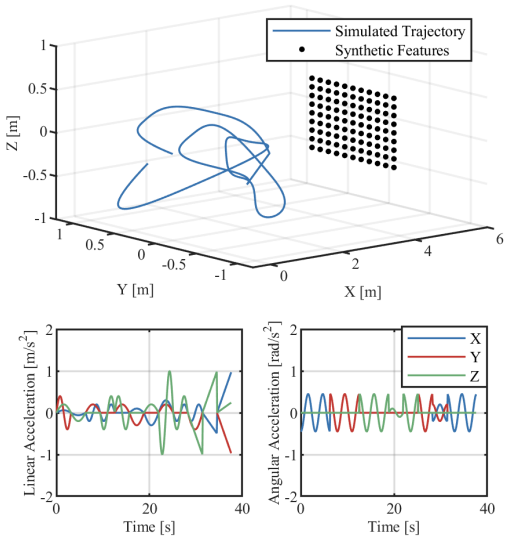


Fig. 2. Experimental setup for our simulation studies. The calibration rig rotates while moving along the blue trajectory. The black dots represent the internal corners of a 12-by-10 checkerboard with squares that are 9.9 cm by 9.9 cm in size, the same as those of our physical checkerboard.

B. Real-World Experiments

We collected a real-world dataset that allowed us to compare the performance of our algorithm to the 3D reprojection-based algorithm of Peršić et al. [24]. Our data collection rig (shown in Figure 4) carried: (i) a PointGrey BFLY-U3-23S6M-C global shutter camera with a Kowa C-Mount 6 mm fixed-focus lens ($96.8° \times 79.4°$ field of view) and (ii) a Texas Instruments AWR1843BOOST 3D radar unit. Both sensors operated at approximately 10 Hz. Data were captured and stored by an on-board Raspberry Pi 4 Model B. The camera intrinsic and lens distortion parameters were obtained using the Kalibr toolbox [25] prior to conducting the experiments. We performed a rough, ad hoc temporal alignment of the radar and camera data before running our optimization algorithm. Additionally, the extrinsic calibration (translation and rotation) parameters were carefully measured by hand for comparison.

Experiments were conducted outdoors to mitigate (to some extent) radar multipath reflections and other detrimental effects. We placed five specialized hybrid radar-camera targets [11] in the environment for validation purposes and for comparison with the calibration method in [24]. However, we emphasize that our algorithm does not specifically make

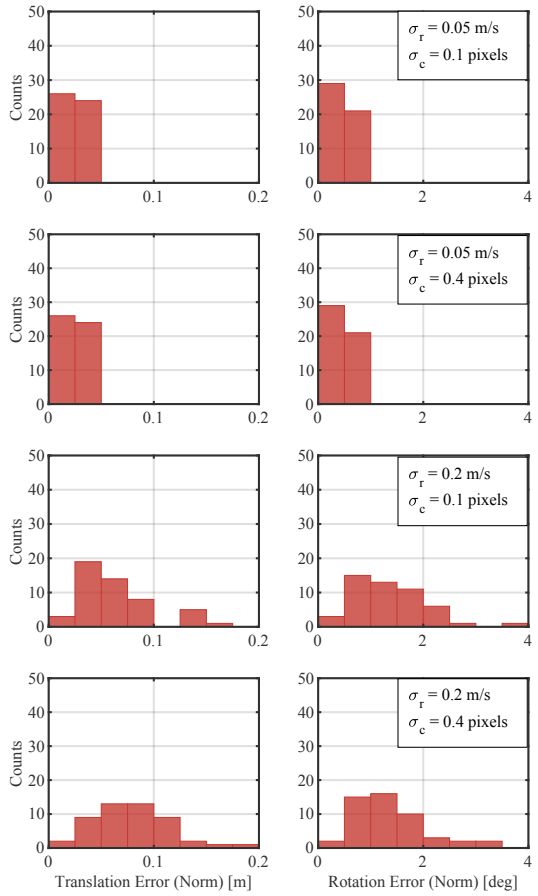


Fig. 3. Left: histograms of translation error norm between estimated and ground truth calibration parameters for different amounts of simulated radar velocity and image pixel noise. Right: histograms of rotation errors. The rotation error is the magnitude of the angle that aligns the estimated and true radar frames. For each noise combination, 50 test cases were run.

use of the retroreflective radar targets; the velocity of the radar can be determined independently.

We evaluated the performance of the calibration algorithm by measuring target reprojection error. We placed an AprilTag [26] on each radar-camera target in the environment, enabling us to estimate the 3D positions of the targets. Using the extrinsic transform obtained via a given calibration method, the radar measurement of the target can be projected into the camera reference frame. The distance between the observed 3D position of the target (from image data) and the projected radar estimate of the target position is the target reprojection error. Figure 5 shows the radar-to-camera reprojection error determined using three different calibration methods: hand-measurement, the 3D reprojection-based method of Peršić et al. [24], and our proposed method. Since the transform estimated by the 3D reprojection method in [24] optimally aligns the AprilTag positions with the projected radar measurements of the targets, this approach outperforms our algorithm according to this metric, as expected. However, the difference in the median reprojection error between our proposed method and that in [24] is less than 4 mm. In contrast to [24], our algorithm does not require any specialized radar targets in the general case.

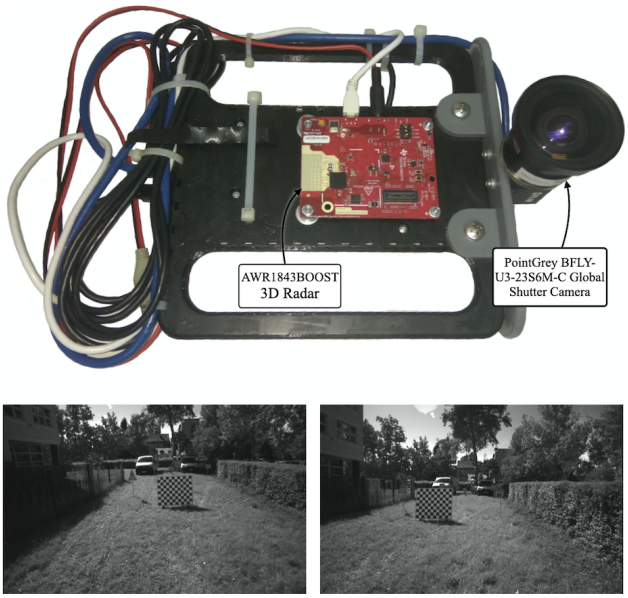


Fig. 4. The top image is a picture of the handheld data collection rig. The bottom two images show different perspectives of our data collection environment.

VI. Conclusion

In this paper, we described a novel continuous-time 3D millimetre-wavelength radar-to-camera extrinsic calibration algorithm. We showed that the problem is observable and derived the necessary conditions for calibration from radar velocity and camera pose measurements only. On synthetic data, our algorithm was shown to be accurate and reliable, but our sensitivity analysis indicated that performance depends on the amount of noise in the radar velocity measurements. Using data from a handheld sensor rig, we demonstrated that we are able to calibrate the extrinsic transform with an accuracy comparable to the method in [24] but without the need for retroreflectors. One future research direction is to investigate alternative cost functions that explicitly consider alignment errors (similar to [24]). Finally, joint spatiotemporal calibration [27] and monocular camera trajectory scale estimation, similar to [28], would be valuable extensions to our algorithm.

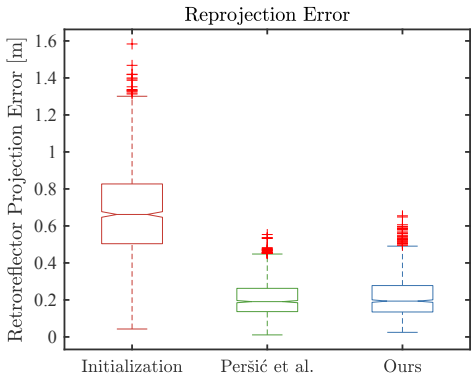


Fig. 5. The target reprojection error is shown for the following calibration methods: hand-measured, Peršić et al. [24], and our proposed method. All algorithms used the same dataset and all calibration results were obtained from a held-out dataset.

REFERENCES

[1] R. Gourova, O. Krasnov, and A. Yarovoy, "Analysis of rain clutter detections in commercial 77 GHz automotive radar," in *2017 European Radar Conference (EURAD)*, 2017, pp. 25–28.

[2] M. A. Richards, J. A. Scheer, and W. A. Holm, Eds., *Principles of Modern Radar: Basic principles*, ser. Radar, Sonar & Navigation. Institution of Engineering and Technology, 2010.

[3] C. C. Stahoviak, "An instantaneous 3D ego-velocity measurement algorithm for frequency modulated continuous wave (FMCW) doppler radar data," Master's thesis, University of Colorado at Boulder, 2019.

[4] S. Sugimoto, H. Tateda, H. Takahashi, and M. Okutomi, "Obstacle detection using millimeter-wave radar and its visualization on image sequence," in *International Conference on Pattern Recognition (ICPR)*, 2004, pp. 342–345.

[5] T. Wang, N. Zheng, J. Xin, and Z. Ma, "Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications," *Sensors*, vol. 11, no. 9, pp. 8992–9008, 2011.

[6] D. Y. Kim and M. Jeon, "Data fusion of radar and image measurements for multi-object tracking via Kalman filtering," *Information Sciences*, vol. 278, pp. 641–652, 2014.

[7] J. Kim, D. S. Han, and B. Senouci, "Radar and vision sensor fusion for object detection in autonomous vehicle surroundings," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 76–78.

[8] T. Kim, S. Kim, E. Lee, and M. Park, "Comparative analysis of RADAR-IR sensor fusion methods for object detection," in *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, 2017, pp. 1576–1580.

[9] G. El Natour, O. Ait Aider, R. Rouveure, F. Berry, and P. Faure, "Radar and vision sensors calibration for outdoor 3D reconstruction," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2084–2089.

[10] J. Domhof, J. F. P. Kooij, and D. M. Gavrila, "An extrinsic calibration tool for radar, camera and lidar," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8107–8113.

[11] J. Peršić, I. Marković, and I. Petrović, "Extrinsic 6DoF calibration of a radar–lidar–camera system enhanced by radar cross section estimates evaluation," *Robotics and Autonomous Systems*, vol. 114, pp. 217 – 230, 2019.

[12] J. Oh, K. Kim, M. Park, and S. Kim, "A comparative study on camera-radar calibration methods," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 1057–1062.

[13] C. Schöller, M. Schnettler, A. Krämmer, G. Hinz, M. Bakovic, M. Güzet, and A. Knoll, "Targetless rotational auto-calibration of radar and camera for intelligent transportation systems," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3934–3941.

[14] J. Peršić, L. Petrović, I. Marković, and I. Petrović, "Online multi-sensor calibration based on moving object tracking," *Advanced Robotics*, vol. 35, no. 3-4, pp. 130–140, 2021.

[15] D. Kellner, M. Barjenbruch, K. Dietmayer, J. Klappstein, and J. Dickmann, "Joint radar alignment and odometry calibration," in *2015 18th International Conference on Information Fusion (Fusion)*, 2015, pp. 366–374.

[16] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on Lie groups," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 145–11 153.

[17] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley, "Continuous-time batch trajectory estimation using temporal basis functions," *The International Journal of Robotics Research*, vol. 34, no. 14, pp. 1688–1710, 2015.

[18] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[19] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[20] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control (TAC)*, vol. 22, no. 5, pp. 728–740, 1977.

[21] A. Chiuso, P. Favaro, Hailin Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 523–535, 2002.

[22] M. Li and A. I. Mourikis, "Online temporal calibration for camera-imu systems: Theory and algorithms," *International Journal of Robotics Research*, vol. 33, no. 7, pp. 947–964, 2014.

[23] J. Brookshire and S. Teller, "Extrinsic calibration from per-sensor egomotion," *Robotics: Science and Systems VIII*, pp. 504–512, 2013.

[24] J. Peršić, L. Petrović, I. Marković, and I. Petrović, "Spatio-temporal multisensor calibration based on gaussian processes moving object tracking," To appear in: *IEEE Transactions on Robotics (TRO)*.

[25] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 473–480.

[26] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3400–3407.

[27] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1280–1286.

[28] E. Wise, M. Giamou, S. Khoubyarian, A. Grover, and J. Kelly, "Certifiably optimal monocular hand-eye calibration," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2020, pp. 271–278.

CURRICULUM VITAE

Juraj Peršić was born in Zagreb, Croatia in 1992. He received his BSc and MSc degree in electrical engineering and information technology from the University of Zagreb, Faculty of Electrical Engineering and Computing (UNIZG-FER) in 2014 and 2016, respectively. As an undergraduate student, he received Dean's Award Josip Lončar for outstanding achievements in the first year of studies, while he was continuously awarded with Scholarship of the City of Zagreb (2009-2016) for excellence. During his master program, he was awarded Erasmus+ scholarship and finished the third semester at Aalborg University, Denmark, as an exchange student. His work on the master thesis was done in collaboration with the company Rimac Automobili d.o.o.

Upon finishing his studies, he was employed as a research assistant at the Department of Control and Computer Engineering (ZARI) at FER, Zagreb, since September 2016. He has worked on several international and domestic scientific projects. In a collaboration with the industry partner KONČAR – Institut za elektrotehniku d.d., he worked on the SafeTram project: System for increased driving safety in public urban rail traffic (2016-2020). His primary roles were radar data processing and calibration of the sensor system. Furthermore, he participated in L4MS project : Logistics for Manufacturing SMEs, where his role was robot localization (2018-2019). During his PhD studies, he spent several months in collaboration at foreign universities. His stay at Julius-Maximilians-University Würzburg, Germany (2016) and Karlsruhe Institute of Technology, Germany (2017) were sponsored by DAAD, while he was awarded Prof. Dr. Sc. Jasna Šimunić-Hrvoić Foundation's scholarship for his stay with the STARS lab at University of Toronto Institute for Aerospace Studies (UTIAS), Canada in 2019. At the end of his PhD studies, he interned at Motional (2020-2021) (former MIT spin-off nuTonomy) - an autonomous driving joint venture between Aptiv and Hyundai Motor Group.

His main research interests within the area of autonomous systems and mobile robotics revolve around sensor calibration and fusion, localization and moving object tracking with focus on radar, lidar and camera systems. He is an author or co-author of 4 papers published in peer-reviewed journals and 5 papers presented at international conferences. The full list of publications is given below.

FULL LIST OF PUBLICATIONS

JOURNAL PUBLICATIONS:

1. J. Peršić, I. Marković and I. Petrović. Extrinsic 6DoF calibration of a radar – LiDAR – camera system enhanced by radar cross section estimates evaluation. *Robotics and Autonomous Systems*, 114:217–230, 2019, IF: 2.825 (Q2).

2. L. Petrović, J. Peršić, M. Seder and I. Marković. Cross-entropy based stochastic optimization of robot trajectories using heteroscedastic continuous-time Gaussian processes. *Robotics and Autonomous Systems*, 133:103618, 2020, IF: 2.825 (Q2).

3. J. Peršić, L. Petrović, I. Marković and I. Petrović. Online multi-sensor calibration based on moving object tracking. *Advanced Robotics*, 35(3-4):130-140, 2021, IF: 1.247 (Q4).

4. J. Peršić, L. Petrović, I. Marković and I. Petrović. Spatiotemporal Multisensor Calibration via Gaussian Processes Moving Target Tracking. *IEEE Transactions on Robotics*, Early Access, 2021, IF: 6.123 (Q1).

CONFERENCE PUBLICATIONS:

1. J. Peršić, I. Marković and I. Petrović. Extrinsic 6DoF calibration of 3D lidar and radar. *IEEE European Conference on Mobile Robots (ECMR)*. Paris, France, 1–6, 2017.

2. L. Petrović, J. Peršić, M. Seder and I. Marković. Stochastic optimization for trajectory planning with heteroscedastic Gaussian processes. *IEEE European Conference on Mobile Robots (ECMR)*. Prague, Czech Republic, 1–6, 2019.

3. M. Seder, L. Petrović, J. Peršić, G. Popović, T. Petković, A. Šelek, B. Bićanić, I. Cvišić, D. Josić, I. Marković, I. Petrović and A. Muhammad. Open Platform Based Mobile Robot Control for Automation in Manufacturing Logistics. *IFAC-PapersOnLine, 52(22):95-100, 2019*.

4. E. Wise, J. Peršić, C. Grebe, I. Petrović and J. Kelly. A Continuous-Time Approach for 3D Radar-to-Camera Extrinsic Calibration. *IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China, 2021 (accepted).

5. Z. Gršković, J. Peršić, I. Marković and I. Petrović. Depth from Mono Accuracy Analysis by Changing Camera Parameters in the CARLA simulator. *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Opatija, Croatia, 2021 (accepted).

ŽIVOTOPIS

Juraj Peršić rođen je u Zagrebu, Hrvatska, 1992. godine. Zvanje prvostupnika, odnosno magistra elektrotehnike i informacijske tehnologije Sveučilišta u Zagrebu, Fakulteta elektrotehnike i računarstva (UNIZG-FER) stekao je 2014. te 2016. godine. Tijekom preddiplomskog studija, nagrađen je Dekanovom nagradom Josip Lončar za izvanredan uspjeh u prvoj godini studija. Kroz svoje obrazovanje, više puta je nagrađen Stipendijom Grada Zagreba (2009-2016) za izvrsnost. Tijekom diplomskog studija, nagrađen je stipendijom Erasmus+ programa te je završio treći semestar na Sveučilištu u Aalborgu, Danska, kao student na razmjeni. Njegov diplomski rad odrađen je u suradnji s kompanijom Rimac Automobili d.o.o.

Po završetku svojih studija, zaposlen je kao znanstveni suradnik na Zavodu za automatiku i računalno inženjerstvo (ZARI) pri UNIZG-FER-u, Zagreb, od rujna 2016. Radio je na nekoliko domaćih i međunarodnih znanstvenih projekata. U suradnji s partnerom iz industrije KONČAR – Institut za elektrotehniku d.d., radio je na projektu SafeTram: Sustav za povećanje sigurnosti vožnje javnog urbanog tračničkog prometa (2016-2020). Glavne uloge na tom projektu su uključivale obradu radarovih podataka te umjeravanje senzorskog sustava. Uz to, sudjelovao je i u projektu L4MS: Logistics for Manufacturing SMEs, gdje se bavio lokalizacijom robota (2018-2019). Tijekom doktorskog studija, proveo je nekoliko mjeseci surađujući na stranim sveučilištima. Njegov boravak na Julius-Maximilians Sveučilištu u Würzburgu, Njemačka (2016) i Karlsruhe institutu za tehnologiju, Njemačka (2017) financirala je zaklada DAAD, dok je stipendijom zaklade Prof. Dr. SC. Jasna Šimunić-Hrvoić nagrađen za boravak u laboratoriju STARS Sveučilišta u Torontu, Instituta za zrakoplovne studije (UTIAS), Kanada u 2019. Na kraju svog doktorskog studija, odradio je praksu u kompaniji Motional (2020-2021) (bivši MIT spin-off nuTonomy) - suradnja na autonomnoj vožnji kompanija Aptiv i Hyundai Motor Group.

Njegovi glavni istraživački interesi u području autonomnih sustava i mobilne robotike nalaze se u području umjeravanja i fuzije senzora, lokalizacije i praćenja gibajućih objekata s naglaskom na sustave radar, lidar, kamera. Autor je ili suautor 4 znanstvena rada u časopisima i 5 radova prezentiranih na međunarodnim konferencijama.

COLOPHON

This document was typeset and inspired by the typographical look-and-feel `classicthesis` developed by André Miede, which was based on Robert Bringhurst's book on typography *The Elements of Typographic Style*, and by the `FERElemental` developed by Ivan Marković whose design was based on `FERBook` developed by Jadranko Matuško.